# Dissertation

submitted to the
Combined Faculties of the Natural Sciences and Mathematics
of the Ruperto-Carola-University of Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

Presented by
**冷卢子未 Luziwei Leng**
born in: Yifeng, Jiangxi, China

Date of oral examination: July 10, 2019

# Solving Machine Learning Problems with Biological Principles

Referees: Dr. Johannes Schemmel
Prof. Dr. Daniel Durstewitz

**Solving Machine Learning Problems with Biological Principles**

Spiking neural networks (SNNs) have been proposed both as models of cortical computation and as candidates for solving problems in machine learning. While increasing recent works have improved their performances in benchmark discriminative tasks, most of them learn by surrogates of backpropagation where biological features such as spikes are regarded more as defects than merits. In this thesis, we explore the generative abilities of SNNs with built-in biological mechanisms. When sampling from high-dimensional multimodal distributions, models based on general Markov chain Monte Carlo methods often have the mixing problem that the sampler is easy to get trapped in local minima. Inspired from traditional annealing or tempering approaches, we demonstrate that increasing the rate of background Poisson noise in an SNN can flatten the energy landscape and facilitate mixing of the system. In addition, we show that with synaptic short-term plasticity (STP) the SNN can achieve more efficient mixing by local modulation of active attractors and eventually outperforming traditional benchmark models. We reveal diverse sampling statistics of SNNs induced by STP and finally study its implementation on conventional machine learning methods. Our work thereby highlights important computational consequences of biological features that might otherwise appear as artifacts of evolution.

**Probleme des maschinellen lernens mit biologischen Prinzipien lösen**

Spikende neuronale Netze (SNN) sind vielversprechende Modellsysteme für die Untersuchung der Funktionsweise des menschlichen Gehirns. Auch im Bereich des maschinellen Lernens finden sie zunehmend Verwendung. Obwohl die Leistungsfähigkeit solcher Netze in letzter Zeit deutlich verbessert wurde, sind die Trainingsmethoden häufig nur Abwandlungen des traditionellen Fehlerrückführens, wobei Spikes hier eher als Hindernisse als als Vorteil gesehen werden. Die vorliegende Arbeit untersucht die generativen Eigenschaften von SNNs, im besonderen unter Zuhilfenahme biologischer Mechanismen. Ein Beispiel ist die Aufgabe, Stichproben aus hoch-dimensionalen Verteilungen mit vielen lokalen Minima zu ziehen. Traditionelle Sampler, basierend auf Markov-Chain-Monte-Carlo-Verfahren, bleiben dabei häufig in einer dieser Moden gefangen. Inspiriert von traditionellem Simulated Annealing bzw. Tempering, wird in dieser Arbeit gezeigt, dass eine Erhöhung des Hintergrund-Poisson-Rauschens eines SNNs zu einer Verflachung der Energielandschaft führt und somit das Mixing verbessert wird. Weiterhin wird gezeigt, dass mit Hilfe von synaptischer Short-Term-Plasticity (STP) SNNs besseres Mixing zeigen, was auf die lokale Modulation von Attraktoren zurückgeführt wird. Die Ergebnisse übertreffen in der Qualität des Mixings die von traditionellen Methoden. Statistische Mae des Samplings von SNNs mit STP werden entwickelt und ihre Eigenschaften werden auf konventielles maschinelles Lernen übertragen. Das Ergebnis dieser Arbeit macht deutlich, dass biologische Eigenschaften nicht nur nicht als Balast der Evolution zu sehen sind, sondern sogar Vorteile gegenüber traditionellen Herangesehensweisen aufzeigen können.

# Contents

# 1. Introduction

Recently, artificial intelligence has achieved state-of-art or even superhuman performances on benchmark datasets such as natural image classification (*Krizhevsky et al.*, 2012) and language modeling (*Devlin et al.*, 2018), as well as hard practical problems like computer games (*Mnih et al.*, 2015; *Oriol Vinyals*, 2019) and Go (*Silver et al.*, 2017). However, their training is usually task-specific and often requires a huge amount of data and iterations which are computationally expensive.

In contrast, humans are able to learn from small samples and generalize from knowledge with low energy consumption. One potential direction for future intelligence system, therefore, is to take inspiration from underlying biological principles in the brain. Different from conventional artificial neural networks (ANNs) that adopt uniform initial architectures, information processing in the brain involves intricate interactions between diverse structures and mechanisms, whose functions still lack sufficient study. Such explorations can be bottom-up including constructing spiking neural networks (SNNs) (*Maass*, 1997) and local synaptic learning rules - an approach also known as neuromorphic computing (*Mead*, 1990), or top-down by developing high-level mechanisms like reinforcement learning (*Sutton and Barto*, 2018), attention (*Xu et al.*, 2015; *Vaswani et al.*, 2017) or meta-learning (*Schmidhuber*, 1987; *Thrun and Pratt*, 1998), etc.

Compare to the usual task driven top-down approach which is receiving increasing popularities, proofs of the computational advantage of SNNs - which are embedded as core computing frameworks in most existing neuromorphic platforms - over traditional ANNs are still insufficient, making them uninteresting for the mainstream machine learning community. The success of deep learning (*LeCun et al.*, 2015) has motivated efforts to implement similar learning principles in SNNs. While an increasing number of recent works have improved their performances in benchmark discriminative tasks (*Lee et al.*, 2016; *Neftci et al.*, 2017; *Zenke and Ganguli*, 2018), the majority of them adopt supervised learning and use surrogates of backpropagation where a large amount of labeled data are still required and certain biological features such as spikes are more regarded as defects than merits.

Different from discriminative models which usually learn a conditional probability distribution of the label given the input data, generative models are statistical models of the joint probability distribution on data and labels. When labels are not provided, the model can be trained in an unsupervised way where the input can be encoded by latent variables. Afterward, generation can be done by sampling from those latent variables.

One typical example of these models is the Boltzmann machine (*Smolensky*, 1986) whose variants and related deep architectures are among the earliest efficient models in

deep learning (*Hinton et al.*, 2006; *Salakhutdinov and Hinton*, 2009; *Dahl et al.*, 2010; *Srivastava and Salakhutdinov*, 2012). Recent works (*Buesing et al.*, 2011; *Petrovici et al.*, 2013, 2016) have related its dynamics to SNNs. When sampling from high-dimensional multimodal distributions which are cases for many real-world datasets, models based on general Markov chain Monte Carlo (MCMC) methods often have the mixing problem that the sampler is easy to get trapped in local minima. Traditional solutions includes annealing or tempering algorithms *Desjardins et al.* (2010a); *Salakhutdinov* (2010); *Bengio et al.* (2013). More recent models avoid this problem by replacing MCMC with other Bayesian approaches (*Kingma and Welling*, 2013; *Goodfellow et al.*, 2014). However, these models use backpropagation to transmit a global error signal, which, despite many recent efforts (*Lillicrap et al.*, 2016; *Whittington and Bogacz*, 2017; *Scellier and Bengio*, 2017; *Sacramento et al.*, 2017), is still controversial in terms of biological plausibility.

In this thesis, under a sampling context, we explore the generative abilities of SNNs and demonstrate how they solve the mixing problem by leveraging built-in biological mechanisms, i.e. background noise and synaptic short-term plasticity (STP).

## Outline

First, we give a brief introduction of generative models and the corresponding mixing problem. We specifically focus on Boltzmann machines which are used as benchmark models throughout the work and their learning algorithms (section 2.1). Subsequently, we discuss the implementation of stochastic sampling on spiking neurons and introduce the LIF sampling framework (section 2.2).

In chapter 3, we study the influence of noise on the membrane potential distribution and the activation function of the LIF neuron (section 3.1), based on which we establish a mapping relation between the rate of background Poisson noise and the temperature of energy based models (section 3.2). For the functional application of the network, we further develop a spike-based tempering framework with a noise variation scheme inspired by neural oscillations and apply it for generation tasks (section 3.3).

In chapter 4, we introduce the biological basis of STP and the Tsodysk-Markram model. Based on this model, we discuss the functionality of synapses with specific shapes of PSP envelopes (section 4.1). On a range of dimensions, we demonstrate how STP can improve the mixing of LIF networks through local modulation of active population and enable them to outperform conventional approaches in machine learning (section 4.2). Furthermore, we reveal the effect of STP on the probability distribution of network states, providing a theoretical explanation for its functionality (section 4.3).

Finally, motivated by the mixing advantage of STP-endowed sampling, we study the implementation of a similar mechanism on traditional restricted Boltzmann machines (chapter 5). Potential influence induced from a variation of PSP shape on sampling is discussed. With preliminary experiments, we demonstrate similar generative performances can be achieved by these networks using a different range of STP parameters.

# 2. Prerequisites

In this chapter, we first give a brief introduction of generative models and the mixing problem. Specifically, due to their close connections with spiking neural networks in dynamics, we will discuss Boltzmann machines which are used as benchmark models for generative tasks throughout the thesis and their corresponding learning algorithms.

In the second part of this chapter, we introduce the implementation of sampling dynamics on both stochastic spiking neurons and the deterministic leaky-integrate and fire (LIF) neurons. The established LIF sampling network integrated with certain biological mechanisms will be used for generative tasks in the following chapters.

## 2.1. Generative networks and the mixing problem

Recently, machine learning and particularly deep learning related approaches have achieved state-of-art or even superhuman performances on benchmark datasets such as natural image classification (*Krizhevsky et al.*, 2012) and language modeling (*Devlin et al.*, 2018), as well as hard practical problems like computer games (*Mnih et al.*, 2015; *Oriol Vinyals*, 2019) and Go (*Silver et al.*, 2017). Among them, much of the applications involve discriminative models which learn the conditional probability distribution of the target given the input data, and the supervised learning usually requires a huge amount of labeled data which are expensive or sometimes unavailable.

In contrast, humans, however, are able to learn by much smaller datasets with few or no labels and even create new samples based on past knowledge or experience. In machine learning, such functionalities are partially realized by generative models which are able to learn the distribution of data either in an unsupervised way or in a supervised way when provided with targets. These models usually encode the input using latent variables and the generation process involves sampling from the latent probability distribution.

One of these generative models is the Boltzmann machine (*Smolensky*, 1986). And the corresponding deep architectures are among the earliest efficient models in deep learning (*Hinton et al.*, 2006; *Salakhutdinov and Hinton*, 2009; *Dahl et al.*, 2010; *Srivastava and Salakhutdinov*, 2012). The learning of BMs depends on Markov chain Monte Carlo (MCMC) sampling to approximate the model distribution (see section 2.1.2), which often has the so-called mixing problem (see section 2.1.3) particularly in high dimensional space that sampling gets stuck in a local minimum and produces correlated samples.

More recent models avoid this problem by substituting MCMC with other techniques. The variational autoencoder (VAE) (*Kingma and Welling*, 2013; *Rezende et al.*, 2014) uses variational inference to approximate the posterior distribution of the latent variable

and generates through a decoder network. Another model that has recently gained great popularity is the generative adversarial network (GAN) (*Goodfellow et al.*, 2014) where learning of the generative network is driven by a discriminative network in order to cheat it, and generation is performed by sampling from a uniform distribution. However, these approaches use backpropagation to transmit a global error signal, which, despite many recent efforts (*Lillicrap et al.*, 2016; *Whittington and Bogacz*, 2017; *Scellier and Bengio*, 2017; *Sacramento et al.*, 2017), is still controversial in terms of biological plausibility.

In comparison, the learning of BMs only requires local information. The use of binary signals also resembles electrical spikes transmitted between synapses. Recent works (*Buesing et al.*, 2011; *Petrovici et al.*, 2013, 2016) have related its dynamics to spiking neural networks based on biologically plausible neuron models and its training has been proved implementable with spike-timing-dependent plasticity (*Neftci et al.*, 2014). Despite other aspects of it that contradict what has been observed in a biological nerves system such as the violation of Dale's law (*Dale*, 1935) and the requirement of symmetrical reciprocal connections. It can still be considered as a sufficiently good prototype to study neuronal dynamics, especially when related to tasks in high dimensional space.

We will introduce BMs and its learning algorithms in subsequent sections and demonstrate how certain biological principles can solve its mixing problem in the following chapters.

### 2.1.1. Boltzmann machines

A Boltzmann machine (BM) is a type of recurrent neural network which consists of symmetrically connected binary units. It resembles a Hopfield network except its units are stochastic. Different from the unit in a Hopfield network whose state is determined by comparing its input to a hard threshold, a unit $i$ in a BM 'fires' with a probability defined by a logistic function:

$$p(z_i = 1) = \frac{1}{1 + e^{-u_i}} \,, \tag{2.1}$$

$$u_i = \sum_{j=1}^{J} W_{ji} z_j + b_i \,, \tag{2.2}$$

where $z_i$ is the binary state of unit $i$ and $u_i$ is its potential which equals to the sum of input it receives plus its own bias $b_i$, $W_{ji}$ is the symmetric weight connecting units $z_j$ and $z_i$, satisfying $W_{ii} = 0$ and $W_{ij} = W_{ji}$.

A BM further defines an energy function on its state $\{\mathbf{z}\}$ as

$$E(\mathbf{z}) = -\sum_{i,j} \frac{1}{2} W_{ij} z_i z_j - \sum_i b_i z_i \,, \tag{2.3}$$

The network assigns a probability to a state vector via the energy function $E(\mathbf{z})$

$$p(\mathbf{z}) = \frac{1}{Z} \exp[-E(\mathbf{z})] \,, \tag{2.4}$$

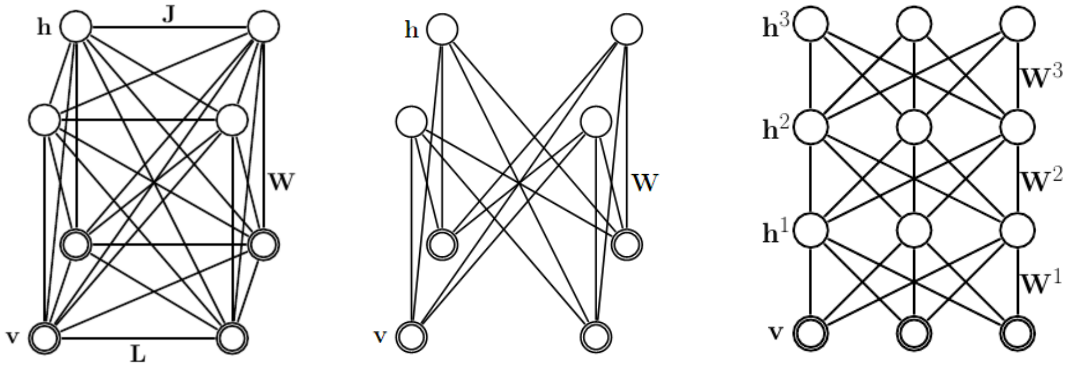


Figure 2.1.: **Left:** A Boltzmann machine with visible units **v** and hidden units **h**. **W**, **L** and **J** are symmetric, zero-diagonal matrices that contain the visible-to-hidden, visible-to-visible and hidden-to-hidden couplings. **Middle:** Setting interaction terms L and J to zero obtains an RBM. **Right:** A three-layer DBM. It can be viewed as three RBMs stacked together. (Images taken from *Salakhutdinov and Hinton*, 2009).

where $Z = \sum_{\mathbf{z}} \exp[-E(\mathbf{z})]$ represents the so-called partition function. The network states thus follow a Boltzmann distribution. With certain sampling algorithms (usually Gibbs sampling) the network can update its state and produce samples sequentially, which will reflect the distribution defined by the model parameters.

The units in a BM can be further subdivided into a set of visible units **v**, whose states can be determined by the input , and a set of hidden units **h**, whose states cannot be directly determined by the input and act as latent variables (see Fig. 2.1). The energy of the state $\{\mathbf{v}, \mathbf{h}\}$ is then defined as:

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i \in \text{visible}} a_i v_i - \sum_{j \in \text{hidden}} b_j h_j - \sum_{i,j} W_{ij} v_i h_j - \sum_{i,i'} \frac{1}{2} L_{ii'} v_i v_{i'} - \sum_{j,j'} \frac{1}{2} J_{jj'} h_j h_{j'} \ , \tag{2.5}$$

where $v_i$ and $h_j$ are the binary states of visible unit $i$ and hidden unit $j$ and $a_i$ and $b_j$ are their biases, $W_{ij}$, $L_{ii'}$ and $J_{jj'}$ represent the visible-to-hidden, visible-to-visible and hidden-to-hidden connection weights. The probability for a particular state of the visible units to occur in a BM is given by summing (marginalizing) over all possible hidden vectors

$$p(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} \exp\left[-E(\mathbf{v}, \mathbf{h})\right] \ . \tag{2.6}$$

When only allowing connections between the visible and hidden units, the network forms a layered structure and becomes a restricted Boltzmann machine (RBM) (see Fig. 2.1). During learning, the visible layer receives training data and the hidden layer learns to model dependencies between the visible units. Hidden units in a way act as feature

detectors are able to capture complex features of a pattern. The connectivity reduction leads to faster states update and allows for more efficient learning algorithms than the ones used for general BMs.

Furthermore, by adding additional layers of hidden units to RBMs, one obtains multilayer deep architectures such as deep belief networks (DBNs) (*Hinton et al.*, 2006) and deep Boltzmann machines (DBMs) (see Fig. 2.1) (*Salakhutdinov and Hinton*, 2009), with improved efficiency of representing complex distributions (*Bengio and LeCun*, 2007; *Srivastava and Salakhutdinov*, 2012).

### 2.1.2. Learning of BMs

In practice, when provided with a training dataset, a BM can be trained to represent the dataset with high probability, corresponding to low energy. Since the stochastic dynamics of a BM favors state vectors that have low energy values, during the learning process, its parameters are updated to lower the energy function of the data in the training set.

By differentiating Eq. 2.4 over the weights and using the fact $\partial E(\mathbf{z})/\partial W_{ij} = -z_i z_j$, it can be shown that

$$\begin{aligned}\frac{\partial p(\mathbf{z})}{\partial W_{ij}} &= \frac{\partial}{\partial W_{ij}}\left[\frac{e^{-E(\mathbf{z})}}{\sum_{\mathbf{z}'} e^{-E(\mathbf{z}')}}\right] \\ &= e^{-E(\mathbf{z})}\cdot z_i\cdot z_j\cdot\left[\sum_{\mathbf{z}'} e^{-E(\mathbf{z}')}\right]^{-1} - e^{-E(\mathbf{z})}\cdot\left[\sum_{\mathbf{z}'} e^{-E(\mathbf{z}')}\right]^{-2}\cdot\sum_{\mathbf{z}'}\left[e^{-E(\mathbf{z}')}\cdot z_i'\cdot z_j'\right] \\ &= p(\mathbf{z})\cdot z_i\cdot z_j - p(\mathbf{z})\left\{\frac{\sum_{\mathbf{z}'}\left[e^{-E(\mathbf{z}')}\cdot z_i'\cdot z_j'\right]}{\sum_{\mathbf{z}'} e^{-E(\mathbf{z}')}}\right\}, \end{aligned} \tag{2.7}$$

where $\sum_{\mathbf{z}'}$ is a sum over all possible states of the model. Moving the $p(\mathbf{z})$ from the LHS of the equation, we get

$$\frac{\partial \log p(\mathbf{z})}{\partial W_{ij}} = z_i\cdot z_j - \left\{\frac{\sum_{\mathbf{z}'}\left[e^{-E(\mathbf{z}')}\cdot z_i'\cdot z_j'\right]}{\sum_{\mathbf{z}'} e^{-E(\mathbf{z}')}}\right\}, \tag{2.8}$$

which can be further transformed to

$$\left\langle \frac{\partial \log p(\mathbf{z})}{\partial W_{ij}} \right\rangle = \langle\, z_i\cdot z_j \,\rangle_{\text{data}} - \langle\, z_i z_j \,\rangle_{\text{model}}, \tag{2.9}$$

where $\langle\cdot\rangle_{\text{data}}$ denotes an average over all the training samples and $\langle\cdot\rangle_{\text{model}}$ denotes the expectation value of the distribution defined by the model. This can lead to a learning rule (*Hinton*, 2010) for performing gradient ascent in the log-probability of the training data

$$\Delta W_{ij} = \eta(\langle\, z_i z_j \,\rangle_{\text{data}} - \langle\, z_i z_j \,\rangle_{\text{model}}), \tag{2.10}$$

where $\eta$ represents the learning rate. The learning rule for the bias $b_i$ is the same as in Eq. 2.9, but with $z_j$ omitted:

$$\Delta b_i = \eta(\langle z_i \rangle_{\text{data}} - \langle z_i \rangle_{\text{model}}) \,. \tag{2.11}$$

Parameter updates for an RBM follow a similar derivation, except that now we model the data distribution only with $p(\mathbf{v})$. Differentiating $p(\mathbf{v})$ over the weights yields

$$\left\langle \frac{\partial \log p(\mathbf{v})}{\partial W_{ij}} \right\rangle = \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}} \,. \tag{2.12}$$

which gives a learning rule similar to Eq. 2.10 and 2.11

$$\Delta W_{ij} = \eta(\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}) \tag{2.13}$$

$$\Delta a_i = \eta(\langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{model}}) \tag{2.14}$$

$$\Delta b_j = \eta(\langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{model}}) \,, \tag{2.15}$$

However, different from a fully visible BM where $\langle z_i z_j \rangle_{\text{data}}$ can be given by the training data, the binary state of a hidden unit $h_j$ in $\langle v_i h_j \rangle_{\text{data}}$ is obtained by evolving the RBM for one sampling step with the visible units clamped to the input.

The computation of $\langle \cdot \rangle_{\text{model}}$ requires the calculation of the partition function of the model, which becomes exponentially expensive as the number of units increases. As an alternative, an approximation can be made by drawing an appropriate sample from the model distribution. Proposed by Hinton (*Hinton*, 2002), contrastive divergence (CD) approximates the expectation value for the model distribution by initializing the model with a training sample and collecting a sample approximating the model distribution after freely evolving the network for $k$ steps of Gibbs sampling. Although it can be shown that CD is not actually following the gradient in Eq. 2.9 (*Sutskever and Tieleman*, 2010), it has been proven to work well enough in many applications (*Hinton*, 2010).

### 2.1.3. The mixing problem

When trained from data, the energy landscape $E(\boldsymbol{z})$ of a BM is shaped in a way that assigns low energy values (modes) to the samples in the training data. If this dataset is composed of very dissimilar classes, training algorithms tend to separate them by high energy barriers (see Fig. 2.2) (*Salakhutdinov*, 2010; *Breuleux et al.*, 2010). As their height grows during training, traditional MCMC sampling methods such as Gibbs sampling become increasingly ineffective at covering the entire relevant state space especially for high-dimensional multimodal distributions, as reflected by a high correlation between consecutive samples (see Fig. 2.2) caused by the component-wise update of states (*Salakhutdinov*, 2010; *Bengio et al.*, 2013; *Breuleux et al.*, 2010; *Desjardins et al.*, 2010b). Consequently, this leads to a poor approximation of the model distribution in the learning process and will increase the time for the network to converge towards its underlying distribution after learning.

The ability of a sampling-based generative model to jump across energy barriers, also known as mixing (see Fig. 2.2), has therefore received significant attention (*Marinari*

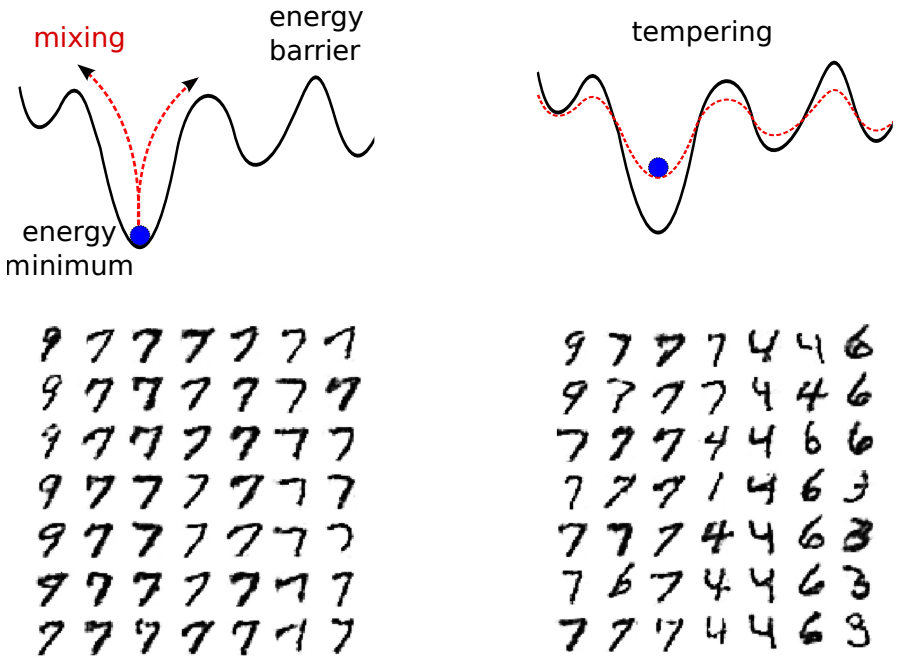


Figure 2.2.: **Upper left:** A conceptual plot of the multimodal energy landscape of the BM formed by training on a high dimensional dataset. **Upper right:** To facilitate mixing, tempering methods globally rescale the energy landscape with a temperature factor and thus make it easier for the network to jump out from a local minimum. **Bottom left:** A sequence (per column) of images generated by an RBM with Gibbs sampling. Due to the large variance in the energy landscape, Gibbs sampling becomes trapped in a local mode, therefore constantly generating similar images. **Bottom right:** A sequence (per column) of images generated by an RBM with adaptive simulated tempering (*Salakhutdinov*, 2010). Within the same amount of samples, the tempering method generates more diverse class of images, significantly better in mixing.

*and Parisi*, 1992; *Wang and Landau*, 2001; *Salakhutdinov*, 2010; *Bengio et al.*, 2013). A plain solution to obtain samples with better mixing property is to run the Markov chain for a longer time (*Tieleman*, 2008). However, it assumes a persistent sampling chain during learning and requires small learning rates, which will eventually reduce the changes made to the energy landscape and prolong the dwell times in local minima, ultimately leading to poor generative models (*Salakhutdinov*, 2010). An alternative approach adopts annealing or tempering techniques (*Marinari and Parisi*, 1992; *Desjardins et al.*, 2010b; *Salakhutdinov*, 2010). The energy landscape is rescaled by a temperature which controls the mixing speed of the sampling process:

$$p(\mathbf{z}) = \frac{1}{Z} \exp[-\beta E(\mathbf{z})] \,, \tag{2.16}$$

where $\beta \in (0, 1]$ is the temperature which changes dynamically during the network evolution. The energy landscape is globally flattened ($\beta < 1$) so that the network is easier to jump out from a local minimum and cooled down ($\beta = 1$) for gathering valid samples (Fig. 2.2). While greatly increasing the mixing capabilities of generative networks, it is important to note that all tempering schedules come with a cost of their own, both because they require additional computations and because they only gather valid samples at low temperatures, thereby effectively slowing down the sampling process.

### 2.1.4. Solutions to the mixing problem

We now introduce a tempering algorithm called adaptive simulated tempering (AST) (*Salakhutdinov*, 2010) which we used in following mixing experiments. It can also be further integrated to form an efficient learning algorithm called coupled adaptive simulated tempering (CAST) (*Salakhutdinov*, 2010), which we use in large size RBMs for high-dimensional learning tasks in subsequent chapters.

AST is a combination of simulated tempering (ST) (*Marinari and Parisi*, 1992) and the Wang-Landau (WL) algorithm (*Wang and Landau*, 2001). The WL algorithm is used to approximate the partition functions in ST, which are originally computationally intractable. In AST, states $z(t+1)$ are updated by Gibbs sampling from the conditional distribution $p(z|\beta_T)$. After each state update, the inverse temperature $\beta$ is itself updated by an adaptive rule that ensures the algorithm spends a roughly equal amount of time at each value. Details of the AST algorithm is described in table 2.1. The CAST algorithm

Table 2.1.: Adaptive simulated tempering

1: Given adaptive weights $\{\mathbf{g}_k\}_{k=1}^K$ and the initial configuration of the state $\boldsymbol{z}^1$ at temperature 1, $k = 1$:
2: **for** $t = 1 : T$ (number of iterations) **do**
3: Given $\boldsymbol{z}^t$, sample a new state $\boldsymbol{z}^{t+1}$ from $p(\boldsymbol{z}|k^t)$ by Gibbs sampling.
4: Given $k^t$, sample $k^{t+1}$ from proposal distribution $q(k^{t+1} \leftarrow k^t)$. Accept with probability: $\min\left(1, \frac{p(\boldsymbol{z}^{t+1},k^{t+1})q(k^t \leftarrow k^{t+1})g_{k^t}}{p(\boldsymbol{z}^{t+1},k^n)q(k^{t+1} \leftarrow k^t)g_{k^{t+1}}}\right)$
5: Update adaptive adjusting factors: $g_i^{t+1} = g_i^t(1 + \gamma_t I(k^{t+1} = i)),\ i = 1, ..., K.$
6: **end for**
7: Collect data: Obtain (dependent) samples from target distribution $p(z)$ by keeping $k = 1$.

improves mixing during the learning process of the RBM. In CAST, two instances of the RBM are simulated in parallel, with one of them staying at a constant inverse temperature $\beta = 1$ for parameter update using persistent contrastive divergence (PCD) (slow chain) (*Tieleman*, 2008) and the other one using adaptive simulated tempering (AST) for mixing

(fast chain). The states of the two RBMs are swapped constantly to help the slow chain jump out of local minima during parameter updating. A draft of the algorithm is plotted in Fig. 2.3.

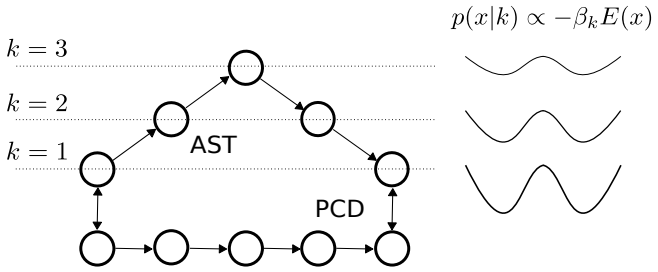


Figure 2.3.: Double-chain system of CAST. The network parameters are updated by PCD, while AST facilitates mixing by adaptively changing the global energy landscape. It is easier for the network to jump out of local minima when energy barriers are lower. The state is swapped between the AST and PCD chain when the former reaches a state with temperature equals 1, i.e. recovers to the original target distribution.

## 2.2. Stochastic sampling with spiking neurons

The sampling dynamics of generative neural networks discussed in the previous section depends largely on the stochastic nature of their units. In this section, we will demonstrate how stochastic sampling is implemented on the seemly incompatible, deterministic biological plausible spiking neurons. As a start, we first create a link between MCMC sampling and the firing activity of simplified, abstract spiking neurons.

### 2.2.1. Sampling with abstract neurons

The work from *Buesing et al.* (2011) proposed a model of abstract spiking neuron (ASNs) with finite time postsynaptic potentials (PSPs) and refractory mechanisms, which is able to implement irreversible MCMC sampling.

In the ASN model, the firing activity of the network at time $t$ is represented by a binary vector $(z_1, ..., z_K)$ as follows:

$$z_k(t) = 1 \Leftrightarrow \text{neuron } k \text{ has fired within the time interval } (t-\tau, t] \ ,$$

meaning that a spike of neuron $k$ sets the value of the associated binary variable $z_k$ to 1 for a duration of $\tau$. Further, the membrane potential $u_k(t)$ of neuron $k$ at time $t$ is defined by the so-called neural computability condition (NCC):

$$u_k(t) = \log \frac{p(z_k(t) = 1 | z_{\setminus k}(t))}{p(z_k(t) = 0 | z_{\setminus k}(t))} \ , \tag{2.17}$$

where$z_{\setminus k}(t)$ are the firing states of all other units with $i \neq k$. For the Boltzmann distribution, plugging its formulation (Eq. 2.4) into the NCC gives a membrane potential of the form:

$$u_k(t) = b_k + \sum_{i=1}^{I} W_{ki} z_i(t) \ , \tag{2.18}$$

where $b_k$ is the bias of neuron $k$ which regulates its excitability, $W_{ki}$ is the synaptic strength between neuron $k$ and $i$. $W_{ki} z_i(t)$ thus represents the time course of the post-synaptic potential in neuron $k$ triggered by the spike of neuron $i$.

In order to specify exactly when the neuron has fired within the time interval $(t-\tau, t]$, additional non-binary variables $(\zeta_1, ..., \zeta_K)$ are introduced. The auxiliary variable record the finite length of the refractory process once the corresponding neuron has fired. In discrete time and for a neuron with an absolute refractory mechanism , the dynamics of $\zeta_k$ is defined in the following way: $\zeta_k$ is set to $\tau$ (the refractory period) when neuron $k$ fires, and decays by 1 in each subsequent discrete time step (see figure 2.4). The neuron can only spike if $\zeta_k \leq 1$ and the spiking probability is defined by

$$p\left[z_k(t) = 1 | \zeta_k(t) \in \{0, 1\}, u_k(t)\right] = \sigma(u_k - \log \tau) \ , \tag{2.19}$$

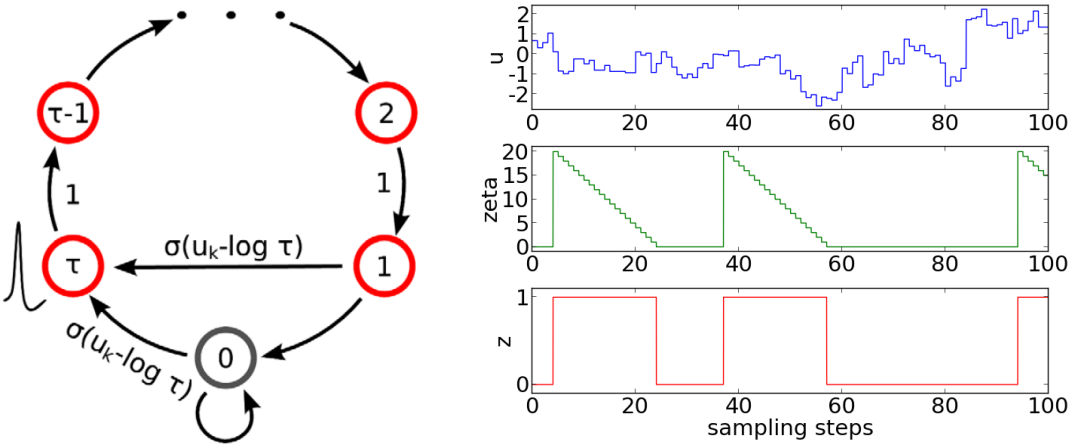


Figure 2.4.: **Left:** A schematic of the internal state variable $\zeta_k$ of a neuron $k$ with an absolute refractory period. During the refractory period, $\zeta_k$ decays by 1 in each subsequent discrete time step and is reset to $\tau$ when the neuron fires again. The neuron can only fire in the resting state $\zeta_k = 0$ and in the last refractory state $\zeta_k = 1$, with a probability defined by a logistic function. (Image is taken from *Buesing et al.*, 2011). **Right:** The figure shows example traces of $z$, $\zeta$ and $u$ of a single neuron for 100 sampling steps in a network of 100 neurons with randomly selected weights and biases. The refractory period $\tau$ is chosen to be 20. While the membrane potential $u$ is updated at every discrete time step, $\zeta$ decreases from $\tau$ to 1 in a fixed manner during the refractory period, and is reset to $\tau$ at $z = 1$ or $z = 0$ with a probability depending on the value of $u$ at that time. The state of the neuron $z$ is set to 1 during the refractory period, otherwise 0.

where $\sigma(x) = (1 + e^{-x})^{-1}$ is the logistic function, and when $\tau = 1$ it resembles the activation function of a unit in BM. Figure 2.4 illustrates how $z_k$, $\zeta_k$ and $u_k$ evolve during a sampling process in discrete time.

Having the three variables defined for the ASN model, *Buesing et al.* (2011) further demonstrated with rigorous mathematical proves that the network dynamics suffices the invariance property of probabilistic inference. And, in the particular case of applying the Boltzmann distribution in the NCC, one can construct a BM based on ASNs.

### 2.2.2. Spiking activity of single LIF neuron in the high-conductance state

In contrast to the ASN model which fires according to a certain probability, neurons in vitro experiments and in biological plausible neural circuits are highly deterministic: the neuron spikes when its membrane potential is above a certain threshold. An often-used model with such properties is the leaky integrate-and-fire (LIF) neuron model, with its

membrane potential $u$ described by the ODE

$$C_{\mathrm{m}} \frac{du}{dt} = g_{\mathrm{l}}(E_{\mathrm{l}} - u) + I(t) \ , \tag{2.20}$$

where $C_{\mathrm{m}}$ is the membrane capacitance, $g_{\mathrm{l}}$ and $E_{\mathrm{l}}$ the leak conductance and potential, and $I$ the input current. When $u$ crosses a threshold $\vartheta$ from below, a spike is emitted, which causes the membrane to be clamped to a reset potential for a duration equal to the refractory period of $\tau_{\mathrm{ref}}$. The synaptic current is modeled as a sum of exponential kernels triggered by presynaptic spikes $s$ with a synaptic time constant $\tau_{\mathrm{syn}}$ and weighted by synaptic efficacy $w_i$ and reversal potentials $E_i^{\mathrm{rev}}$:

$$I^{\mathrm{syn}}(t) = \sum_s \sum_i w_i \left(E_i^{\mathrm{rev}} - u\right) \exp\left[-(t - t^s)/\tau_{\mathrm{syn}}\right] \ . \tag{2.21}$$

In a noisy environment, the total input current $I$ to a neuron can be partitioned into recurrent synaptic input, diffuse synaptic noise and additional external currents: $I = I^{\mathrm{rec}} + I^{\mathrm{noise}} + I^{\mathrm{ext}}$. For the analysis of individual neurons, $I^{\mathrm{rec}}$ and $I^{\mathrm{ext}}$ can be set to zero. When a neuron receives enough synaptic stimulation, it enters a so-called high-conductance state (HCS), in which the neuron is "driven" by synaptic inputs rather than being dominated by its intrinsic dynamics, and usually characterized by accelerated membrane dynamics. In the high input rate circumstance, the equation governing the membrane potential can then be written as

$$\tau_{\mathrm{eff}} \frac{du}{dt} = u_{\mathrm{eff}} - u \tag{2.22}$$

$$u_{\mathrm{eff}} = \frac{g_{\mathrm{l}} E_{\mathrm{l}}}{\langle\, g_{\mathrm{tot}} \,\rangle} + \frac{\sum_i g_i^{\mathrm{noise}} E_i^{\mathrm{rev}}}{\langle\, g_{\mathrm{tot}} \,\rangle} \ , \tag{2.23}$$

with $\langle \cdot \rangle$ denoting the mean and $g_i^{\mathrm{noise}}$ representing the total conductance at the $i$th synapse. The membrane time constant $\tau_m = C_{\mathrm{m}}/g_{\mathrm{l}}$ in Eq. 2.20 is replaced by an effective time constant $\tau_{\mathrm{eff}} = C_{\mathrm{m}}/g_{\mathrm{tot}}$, with the total conductance $g_{\mathrm{tot}}$ subsuming both leakage and synaptic conductance. In a first-order approximation, $\tau_{\mathrm{eff}}$ can be considered very small in the HCS, resulting in $u \approx u_{\mathrm{eff}}$, with the effective potential $u_{\mathrm{eff}}$ simply being a linear transformation of the synaptic noise input.

Derived from the approach in *Ricciardi and Sacerdote* (1979), *Petrovici et al.* (2013) have shown that, if stimulated by a large number of uncorrelated spike sources, the synaptic current $I^{\mathrm{noise}}$ - and therefore, also $u_{\mathrm{eff}}$ - can be described as an Ornstein-Uhlenbeck (OU) process

$$du(t) = \theta \cdot (\mu - u(t)) + \Sigma \cdot dW(t) \ , \tag{2.24}$$

where $W(t)$ represents the Wiener process, with parameters

$$\theta = \frac{1}{\tau_{\mathrm{syn}}} \tag{2.25}$$

$$\mu = \frac{I^{\mathrm{ext}} + g_{\mathrm{l}} E_{\mathrm{l}} + \Sigma_i \nu_i w_i E_i^{\mathrm{rev}} \tau_{\mathrm{syn}}}{\langle\, g_{\mathrm{tot}} \,\rangle} \tag{2.26}$$

$$\Sigma^2 = \sum_i \nu_i w_i^2 (E_i^{\mathrm{rev}} - \mu)^2 \tau_{\mathrm{syn}} / \langle\, g_{\mathrm{tot}}^2 \,\rangle \ , \tag{2.27}$$

where $\nu_i$ represents the input rate at the $i$th noise synapse.

Based on the comprehensive description of the free membrane potential dynamics, an activation function of the LIF neuron in a spiking noisy environment can be derived when a firing threshold is introduced (*Petrovici et al.*, 2013). Analogously to the ASN model, a neuron with a freely evolving membrane potential is said to be in the state $z_k = 0$ and switches to the state $zk = 1$ upon firing, where it stays for the duration of the refractory period (see Fig. 2.5). For neuron membrane dynamics with a reset mechanism, two

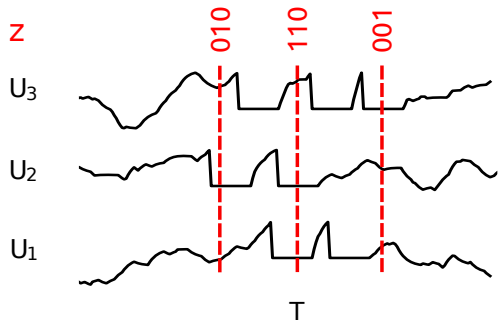


Figure 2.5.: Interpretation of states as samples in a spiking network. The plot shows membrane potential traces of three neurons and their corresponding binary states interpretation. When an LIF neuron is in a refractory period its state is defined as $z_k = 1$, otherwise as $z_k = 0$.

modes can be observed: the "bursting" mode, where the effective membrane potential after the refractory period is still above the threshold, and the freely evolving mode, where the neuron does not spike again immediately after the refractory period. Denoting the relative occurrence of burst lengths $n$ by $P_n$ and the average duration of the freely evolving mode that follows an $n$-spike-burst by $T_n$, the following relation can be derived:

$$p(z = 1) = \frac{\sum_n P_n \cdot n \cdot \tau_{\text{ref}}}{\sum_n P_n \cdot \left( n\tau_{\text{ref}} + \sum_{k=1}^{n-1} \overline{\tau_k^{\text{b}}} + T_n \right)} \quad , \tag{2.28}$$

where $\overline{\tau_k^{\text{b}}}$ represents the average drift time from the reset to the threshold potential following the $k$th refractory period within a burst. The calculation of $P_n$, $T_n$ and $\overline{\tau_k^{\text{b}}}$ depend on all neuron and noise parameters, where $P_n$, $T_n$ can be computed recursively[1]. *Petrovici et al.* (2016) demonstrated that, under HCS created by independent excitatory and inhibitory Poisson input noise with certain frequencies, the firing probability of the LIF neuron as a function of its mean membrane potential[2] resembles a logistic function (see figure 2.6). The LIF neuron can thus closely approximate the spiking activity of the ASN model.

[1] For detailed derivation see *Petrovici* (2016).

[2] The mean membrane potential can be determined by leak potential or external input currents.

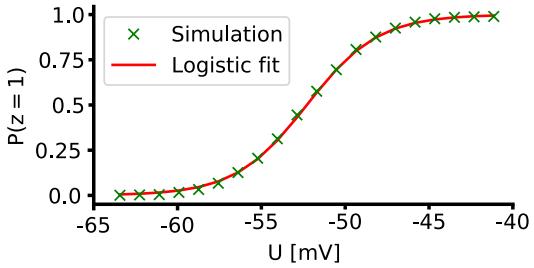


Figure 2.6.: The activation function of a LIF neuron. Theoretical prediction (red) vs. simulation results (green). A logistic function $\sigma(\bar{u})$ (red) is fitted to the prediction.

### 2.2.3. Sampling with LIF networks

Based on the logistic activation function, we can construct a sampling framework based on LIF neurons by mapping the corresponding ASN network parameters to the LIF domain. For a single ASN without recurrent synaptic connections, its firing probability is fully determined by its bias (see Eq. 2.18), which can be translated to the LIF domain as:

$$b_k = (\bar{u}_k^b - \bar{u}_k^0)/\alpha \,, \tag{2.29}$$

where $k$ is the neuron index, $\bar{u}_k^b$ denotes the mean free membrane potential, $\bar{u}_k^0$ is the mean free potential when the neuron has a firing probability of 0.5 and $\alpha$ is a scaling factor.

The synaptic efficacy in the LIF domain is estimated by making the impact of a pre-synaptic spike on the post-synaptic neuron the same as the one in the ASN model. Specifically, the impact is an average interaction and can be calculated as the integrated area of PSPs for a duration of the refractory period (see figure 2.7). The synaptic weights translation can thus be written down as (*Petrovici et al.*, 2013):

$$W_{kj} = \frac{1}{\alpha C_\mathrm{m}} \frac{w_{kj}\left(E_{kj}^\mathrm{rev} - \mu\right)}{1 - \frac{\tau_\mathrm{syn}}{\tau_\mathrm{eff}}} \left[\tau_\mathrm{syn}(e^{-1} - 1) - \tau_\mathrm{eff}\left(e^{-\frac{\tau_\mathrm{syn}}{\tau_\mathrm{eff}}} - 1\right)\right] . \tag{2.30}$$

where $w_{kj}$ is the synaptic weight in the LIF domain projecting from neuron $k$ to $j$, $E_j^\mathrm{rev}$ is the reversal potential for synapse $w_{kj}$. Furthermore, short-term synaptic depression (see section 4.1 for details) was employed to approximate the theoretically optimal rectangular PSP shape for consecutive spikes (bursts). This setup of parameter translations allows an accurate sampling of LIF networks from target Boltzmann distributions, as demonstrated in Fig. 2.7.

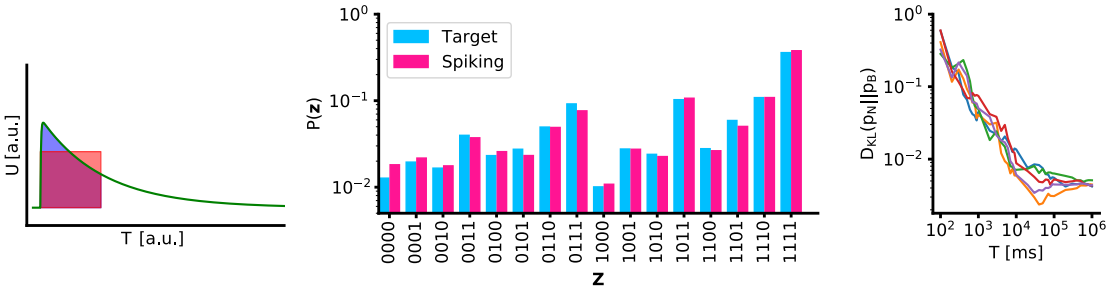


Figure 2.7.: **Left:** Sketch of synaptic weight translation. The synaptic efficacy in the LIF domain is estimated by making the impact of a pre-synaptic spike on the post-synaptic neuron the same as the one in the ASN model which is a rectangle. **Middle:** Sampled distribution of a fully connected 4-neuron LIF network vs. target distribution. **Right:** Evolution of Kullback-Leibler divergence between sampled ($p_{\mathrm{N}}$) and target ($p_{\mathrm{B}}$) distribution for 5 different random seeds.

# 3. LIF networks with temperatures

In the previous chapter, we have shown that a network consisting of LIF neurons can approximate the dynamics of stochastic sampling. Specifically, when the activation function of the LIF neuron is calibrated to well fit a logistic function under Poisson background noise, the network is able to closely match the statistics of Gibbs sampling in a Boltzmann distribution after appropriate parameter translations. However, this close approximation also inherits the mixing problem of traditional Boltzmann machines, particularly when sampling from high dimensional multimodal distributions, as discussed in section 2.1.3.

In this chapter, we demonstrate that instead of using a homogeneous Poisson noise, a variation of noise rate can change the slope of the activation function of individual neurons which leads to a rescaling of the global energy landscape. This enables the network to jump out of local attractors and facilitates mixing. The approach is analogous to principles in traditional annealing or tempering algorithms, therefore, we named it as spike-based tempering.

To start with, in section 3.1, we will first discuss how noise modulates the membrane potential distribution of the LIF neuron and its further influences on the shape of the activation function. Based on this, in section 3.2, through theoretical calculation, we derive a mapping relation between the temperature defined for energy based models and the rate of background Poisson noise in LIF sampling networks. In addition, we demonstrate how to counteract the shift of activation functions induced from the reset mechanism by using imbalanced excitatory and inhibitory noise. In section 3.3, inspired by neural oscillations observed in the cortex, we develop a rate variation scheme for the background noise and apply the LIF network to high dimensional image generation tasks where we test the mixing performance. In the end, with a quantitative measurement of mixing, we search for an optimal configuration of noise parameters.

The work described in this chapter are collaborate works with Agnes Korcsák-Gorzó (*Korcsak-Gorzo*, 2017) and are currently preparing for publication. Mihai A. Petrovici also offered a lot of insightful advice during the discussion. The corresponding simulator module based on PyNN (*Davison et al.*, 2008) is developed by Agnes Korcsák-Gorzó and Oliver Breitwieser. The neuron parameters for all simulations in this chapter are described in appendix A.2.1 unless specifically mentioned.

## 3.1. Dynamics of LIF neurons under varying background noise

To sample accurately from a target distribution, the LIF sampling framework relies on Poisson noise with a constant input rate. In this section, we will investigate how varying background noise modulates the activation function of LIF neurons.

### 3.1.1. Free membrane potential dynamics under Poisson noise

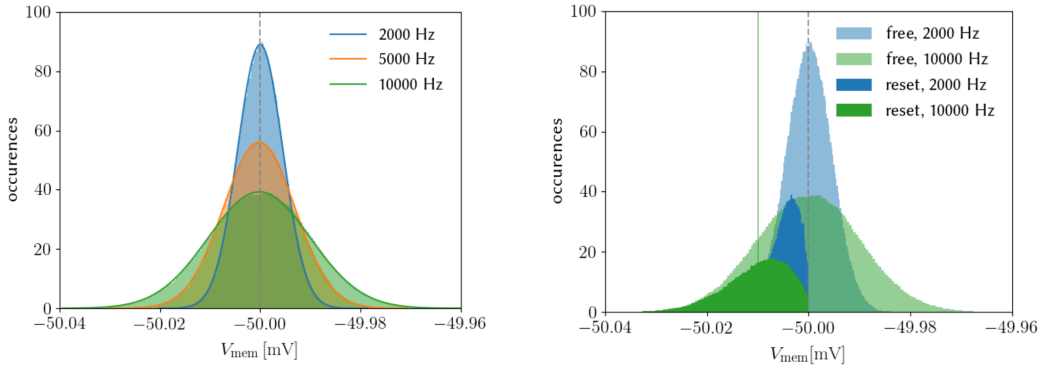


Figure 3.1.: **Left:** Histogram of the free membrane potential of a LIF neuron in a simulation for 10 seconds (transparent colored area) and fitted with a Gaussian (envelope line). The simulation is performed for balanced Poisson noise of 2 kHz (blue), 5 kHz (orange) and 10 kHz (green). With increasing rates the variance of the membrane potential increases. **Right:** The free membrane potential distributions (transparent colored area) opposed to distribution with a threshold potential (colored areas) for balanced input of 2 kHz (blue) and 10 kHz (green). The threshold potential is set identical to the resting potential -50 mV (gray dashed line). The vertical line at the reset membrane potential -50.01 mV (green overlays blue) corresponds to the reset states after the spikes. At the reset potential even more states are accumulated, which are cut off for clarity. Figure is taken from *Korcsak-Gorzo* (2017).

The dynamics and statistics of Poisson-driven LIF neurons were nicely elaborated in *Petrovici* (2016). According to the work, when the Poisson rate is high enough (HCS), the mean and variance of the membrane potential of a current-based (CUBA) LIF neuron can be expressed as

$$E[u] = E_{\mathrm{l}} + \frac{I^{\mathrm{ext}}}{g_{\mathrm{l}}} + \frac{\sum_{k=1}^{n} w_k \nu_k \tau_k^{\mathrm{syn}}}{g_{\mathrm{l}}}\,, \tag{3.1}$$

$$\mathrm{Var}\,[u] = \sum_{k=1}^{n} \left[\frac{\tau_{\mathrm{m}}\tau_k^{\mathrm{syn}}}{C_{\mathrm{m}}(\tau_{\mathrm{m}} - \tau_k^{\mathrm{syn}})}\right]^2 w_k^2 \nu_k \left(\frac{\tau_{\mathrm{m}}}{2} + \frac{\tau_k^{\mathrm{syn}}}{2} - 2\frac{\tau_{\mathrm{m}}\tau_k^{\mathrm{syn}}}{\tau_{\mathrm{m}} + \tau_k^{\mathrm{syn}}}\right). \tag{3.2}$$

where $w_k$ and $\nu_k$ are the synaptic weight and rate of the corresponding input noise. The expressions for a conductance-based (COBA) LIF neuron are:

$$E[u] = \frac{g_\mathrm{l} E_\mathrm{l} + I^\mathrm{ext} + \sum_k w_k \nu_k \tau_k^\mathrm{syn} E_k^\mathrm{rev}}{g_\mathrm{l} + \sum_k w_k \nu_k \tau_k^\mathrm{syn}} \,, \tag{3.3}$$

$$\mathrm{Var}\,[u] = \sum_{k=1}^{n} \left[ \frac{\langle\, \tau_\mathrm{eff} \,\rangle\, \tau_k^\mathrm{syn} (E_k^\mathrm{rev} - \langle\, u_\mathrm{eff} \,\rangle)}{C_\mathrm{m}(\langle\, \tau_\mathrm{eff} \,\rangle - \tau_k^\mathrm{syn})} \right]^2 w_k^2 \nu_k \left( \frac{\langle\, \tau_\mathrm{eff} \,\rangle}{2} + \frac{\tau_k^\mathrm{syn}}{2} - 2 \frac{\langle\, \tau_\mathrm{eff} \,\rangle\, \tau_k^\mathrm{syn}}{\langle\, \tau_\mathrm{eff} \,\rangle + \tau_k^\mathrm{syn}} \right) \tag{3.4}$$

where $\langle\, u_\mathrm{eff} \,\rangle$ equals $E[u]$ and $\langle\, \tau_\mathrm{eff} \,\rangle$ takes the form

$$\langle\, \tau_\mathrm{eff} \,\rangle = \frac{C_\mathrm{m}}{g_\mathrm{l} + \sum_k w_k \nu_k \tau_k^\mathrm{syn}} \,. \tag{3.5}$$

Under HCS, according to the central limit theorem, the free membrane potential of a LIF neuron will follow a Gaussian distribution. For both CUBA and COBA neurons, it can be seen that $\mathrm{Var}\,[u]$ is proportional to the noise input rate $\nu_k$ and the squared input weight $w_k$ . Changing these two terms leads to a change in the width of the Gaussian distribution (Fig. 3.1 left). This results in a slope change in the cumulative function of the Gaussian, which approximates the change of the activation function. This role played by the noise laid the theoretical foundation of spike-based tempering.

### 3.1.2. Activation functions under varying background noise

For a single LIF neuron embedded in a noisy background without external input from other neurons, its activation function can be measured by the proportion of its spiking period (refractory duration multiply the number of spikes) over the total simulation time on a range of mean membrane potentials.

As we know from the previous section, an increasing noise rate or input weight will lead to the broadening of the membrane potential distribution. Its influence on the activation function can be briefly illustrated. For the case when the mean membrane potential is smaller than the threshold, a neuron with a broader membrane potential distribution will have a larger distributed area above the threshold than a neuron with a narrower membrane potential distribution (see Fig. 3.2 left), leading to elevated spiking probabilities, corresponding to a more flattened slope (see Fig. 3.3). Similarly, for the case when the mean membrane potential is larger than the threshold, a neuron with a broader membrane potential distribution will have a larger distributed area below the threshold than a neuron with a narrower membrane potential distribution (see Fig. 3.2 right), leading to a decrease of spiking probabilities, which also corresponds to a more flattened slope (see Fig. 3.3).

However, the above analysis is only an approximation. In practice, the influence of the threshold and reset mechanism distorts the original membrane potential distribution, especially near the threshold (see Fig. 3.1 right). Intuitively, it occurs due to the membrane potential can no longer evolve to the nearby region of the threshold from values above. The reset mechanism allocates all membrane potential above the threshold to the

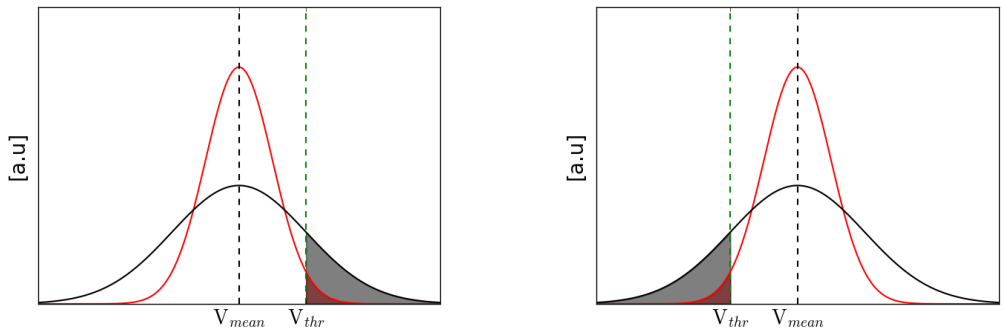


Figure 3.2.: **Left:** Membrane potential distribution when the mean membrane potential is below the threshold. The shaded area indicates the membrane potential above the threshold. It can be clearly seen that the broader the distribution, the larger this area. **Right:** Membrane potential distribution when the mean membrane potential is above the threshold. The shaded area indicates the membrane potential bellow the threshold. The broader the distribution, the larger this area.

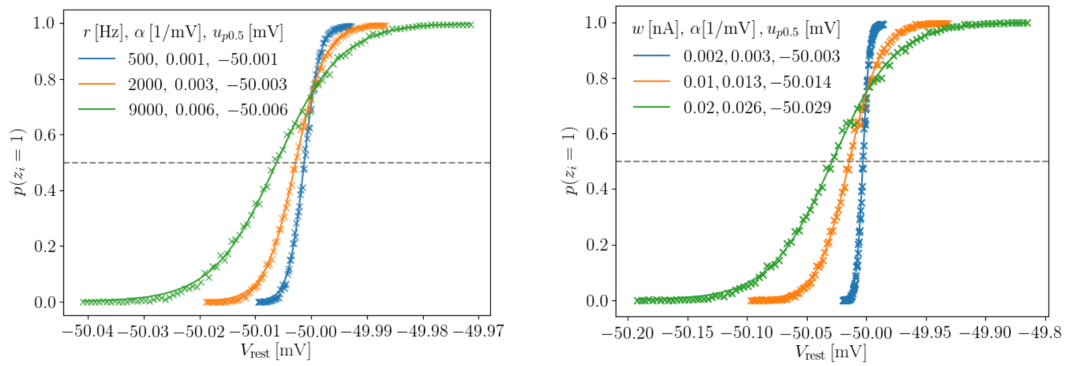


Figure 3.3.: **Left:** Activation functions with balanced excitatory and inhibitory noise of 0.5 (blue), 2 (orange), 9 (green) kHz and 0.002 nA synaptic weights. With increasing noise rate, the slope decreases and the activation function is shifted to the left. **Right:** Activation functions with balanced excitatory and inhibitory noise of 2 kHz and different synaptic weights of 0.002 (blue), 0.01 (orange), 0.02 (green) nA. With increasing weight, the slope decreases and the activation function is shifted to the left. Figure is taken from *Korcsak-Gorzo* (2017).

reset potential immediately. The distortion eventually leads to a shift of the activation function, which can be seen from Fig. 3.3. However, the activation functions seem always cross at a particular point. For further investigation, we swept over a range of noise rates

and found the cross point at a firing probability of approximately 0.8 (see Fig. 3.4)[1]. At this specific mean membrane potential, the firing probability is stable independent

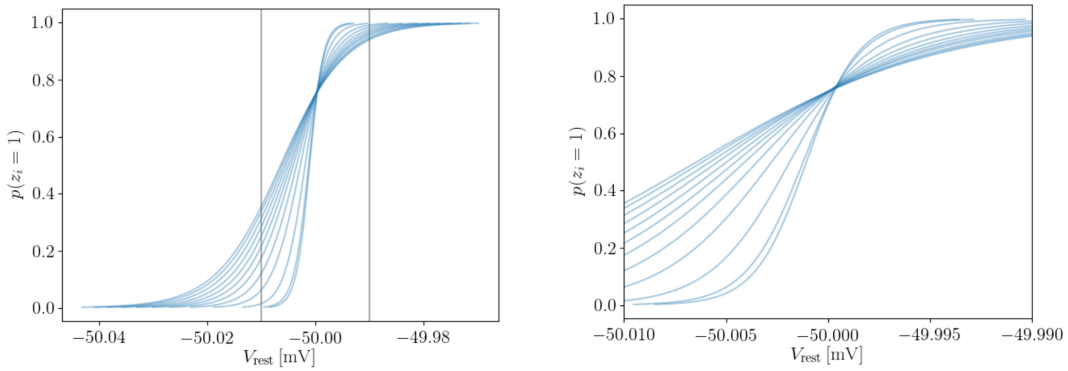


Figure 3.4.: Sigmoid functions fitted to activation functions of a LIF neuron, calibrated on 2 kHz under different balanced noise rates: 0.4, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 kHz. All lines cross at a firing probability of approximately 0.8. A close-up of the area between the gray lines in the left plot is on the right. Figure is taken from *Korcsak-Gorzo* (2017).

of the value of the balanced noise rate pair. The position of the crossing point could be important for potential functional applications. Further investigation is needed to understand this phenomenon.

[1] Value might differ for other neuron parameters

## 3.2. From noise to temperatures

In annealing or tempering approaches, the inverse temperature of the system varies between 0 and 1 in order to modify the attractor strength of the energy landscape. For functional use of the noise, we need to derive a mapping relation between the temperature and the noise.

### 3.2.1. Mapping temperature to the slope of activation function

We first derive the relation between the temperature and the slope of the activation function. The activation function of a unit in a Boltzmann machine with temperature $\beta$ is defined as a logistic function

$$p(z_i = 1) = \frac{1}{1 + e^{-\beta u_i}} \ , \tag{3.6}$$

$$u_i = b_i + \sum_{j=1}^{J} W_{ji} z_j \ , \tag{3.7}$$

The activation function of a LIF neuron is fitted by

$$p(z = 1) = \frac{1}{1 + e^{-(u - u_{p0.5})/\alpha}} \tag{3.8}$$

where $u_{p0.5}$ is the mean membrane potential when the neuron fires at a probability of 0.5, $\alpha$ denotes the slope. Comparison between Eq. 3.6 and Eq. 3.8 reveals the relation between $\alpha$ and $\beta$. Since $\beta$ is unit free and varies between 0 to 1, we need to set a certain $\alpha_0$ for reference as the equivalent of $\beta_0 = 1$, which leads to the following mapping relation

$$\beta_n = \frac{\alpha_0}{\alpha_n} \tag{3.9}$$

### 3.2.2. Mapping noise to the slope of activation function

The shape of the activation function of a LIF neuron can be approximated by the cumulative distribution function (CDF) of the free membrane potential. Since the membrane distribution is modulated by the noise, we can then derive the relation between the noise and the slope of the activation function $\alpha$.

The probability density function of a Gaussian is given by

$$\mathrm{PDF}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left( - \frac{(x - \mu)^2}{2\sigma^2} \right) \tag{3.10}$$

where $x$ is the random variable, $\mu$ the mean and $\sigma^2$ the variance. The CDF is defined as the integral over the probability density function from minus infinity to $x$

$$\mathrm{CDF}(x|\mu, \sigma) = \frac{1}{2} \left[ 1 + \mathrm{erf}\left( \frac{x - \mu}{\sigma\sqrt{2}} \right) \right] \tag{3.11}$$

where the error function is defined as

$$\mathrm{erf}(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2} dt = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2} dt \tag{3.12}$$

Consider the case when the CDF is at the center ($\mu = 0$), the same as with the LIF activation function in Eq. 3.8 ($u_{p0.5} = 0$) and set their slope (derivative) to be equal at $x = 0$:

$$\partial_x \mathrm{CDF}|_{x=0} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}|_{x=0} = \frac{1}{\sqrt{2\pi}\sigma} \tag{3.13}$$

$$\partial_x \sigma(x)|_{x=0} = \frac{1}{\alpha} \frac{e^{-\frac{x}{\alpha}}}{(1+e^{-\frac{x}{\alpha}})^2}|_{x=0} = \frac{1}{4\alpha} \tag{3.14}$$

$$\frac{1}{\sqrt{2\pi}\sigma} = \frac{1}{4\alpha} \tag{3.15}$$

The 0th order approximation yields the dependency of the slope on the variance of the membrane potential distribution

$$\alpha = \frac{1}{4}\sqrt{2\pi}\sigma \ . \tag{3.16}$$

This estimate can be used as an initial value for the fit of the activation function in the LIF sampling framework. The dependence of the variance on the noise rate (Eq. 3.2, 3.4) further yields

$$\frac{1}{\beta} \sim \alpha \propto \sigma \propto \sqrt{\nu} \tag{3.17}$$

This approximation can be further verified from simulation. The resulting temperature (inverse alpha ratio) values of a range of input noise rates are plotted in Fig. 3.5. The logarithmic plot shows a linear dependence based on which we calculate the underlying power law

$$\beta = \frac{\alpha_0}{\alpha} = \mathrm{const} \cdot \nu^m \tag{3.18}$$

Taking the logarithm of both sides and solving for the power $m$ by plugging into two data points

$$m = \frac{\log\left(\frac{\beta_1}{\beta_2}\right)}{\log\left(\frac{\nu_1}{\nu_2}\right)} \tag{3.19}$$

gives a value of $m$ of approximately -0.5, which leads to

$$\beta = \frac{\alpha_0}{\alpha} \sim \frac{1}{\sqrt{\nu}} \ . \tag{3.20}$$

This result matches our previous approximation in Eq. 3.17. Fig. 3.6 further shows the close match between the prediction and simulation.

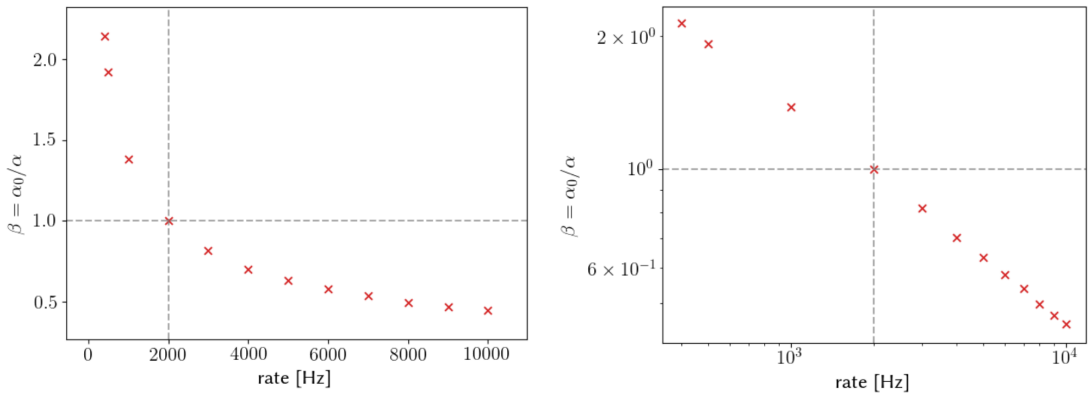


Figure 3.5.: Simulation results of temperature values (inverse alpha ratio) corresponding to input noise rates. The reference noise rate and its corresponding $\beta$ value are marked with gray dashed lines. **Left:** Display with linear scales. **Right:** Display with logarithmic scales. Figure is taken from *Korcsak-Gorzo* (2017).

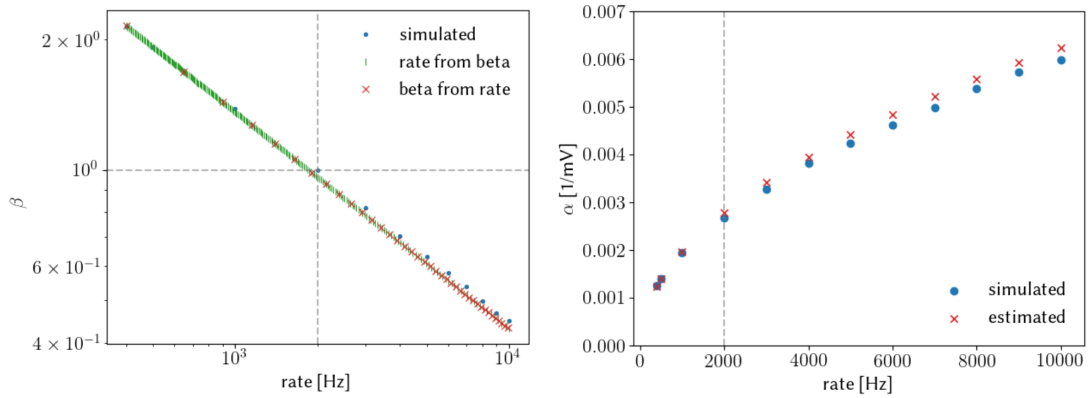


Figure 3.6.: **Left:** Simulated temperature values versus predicted values calculated from Eq. 3.16. **Right:** $\alpha$ from simulation and prediction calculated from Eq. 3.16. Both results show generally good accordance. Figure is taken from *Korcsak-Gorzo* (2017).

### 3.2.3. Calibration of the activation function

After establishing the mapping between noise and temperature, the final step is to adjust the horizontal shift of activation functions caused by the reset mechanism (see Fig. 3.3), which can be achieved by changing the mean membrane potential of the neuron. In the following, we take CUBA type LIF neuron as an example.

For a single CUBA LIF neuron under excitatory and inhibitory Poisson noise input, its mean membrane potential is calculated according to Eq. 3.1 as

$$E[u] = E_\mathrm{l} + \frac{I^\mathrm{ext} + w_e \nu_e \tau_e^\mathrm{syn} + w_i \nu_i \tau_i^\mathrm{syn}}{g_\mathrm{l}} \tag{3.21}$$

where $I^{\text{ext}}$ can be set to zero. In the case of balanced noise input, the inhibitory weight $w_i$ has a negative sign so that the inhibitory noise term cancels the excitatory, and the mean membrane potential is fully determined by $E_{\text{l}}$. To change $E[u]$, one can modify $I^{\text{ext}}$, or break the balance of the input noise by changing $w$ or $\nu$. In practice, considering biological plausibility and the potential need to change the temperature continuously, we take the rate of the noise as the means of modulation. This approach of varying the rate of the background noise also resembles neural oscillations observed in the brain.

We take a benchmark noise rate of 2 kHz, where the shift of $u_{p0.5}$ of other rates is measured from. For other noise rates, we sweep a number of inhibitory rates, below and above the excitatory rate. The resulting shift values for the pairs of excitatory and inhibitory rates are plotted in Fig. 3.7 and encoded by color. The blue line corresponds to the balanced case with $r_{\text{exc}} = r_{\text{inh}}$ . The black circles mark the noise pairs that diminish the shift closest to zero. In this way, we collect a set of noise pairs that can be better mapped to temperatures. The right plot of Fig. 3.7 shows three calibrated activation functions (solid lines) achieved with the noise pairs we found. The dashed lines correspond to the activation function before calibration with balanced input noise.

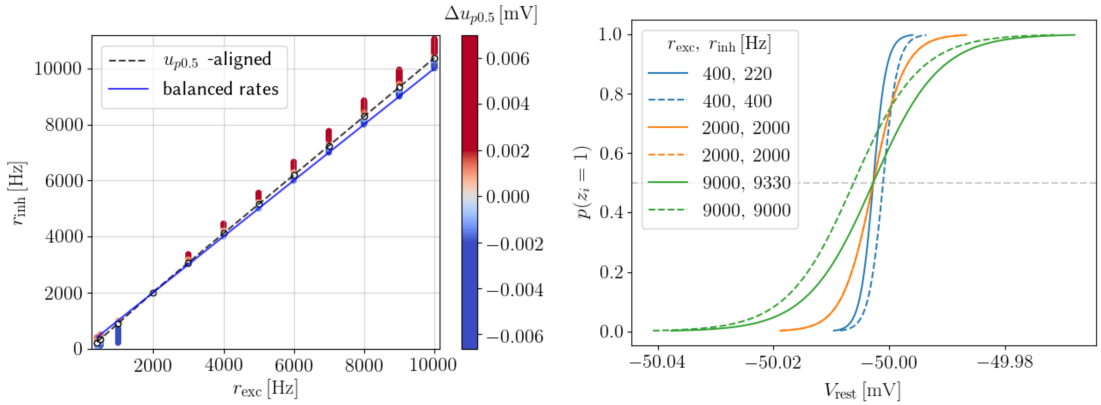


Figure 3.7.: **Left:** Determination of the shift-compensating inhibitory rates. The x-axis corresponds to the excitatory rate and the y-axis to the inhibitory rate. The excitatory rate values are chosen in equal distances between 400 Hz and 10 kHz. A neuron is stimulated by each noise rate pair and the corresponding shift of $u_{p0.5}$ value of the activation function is plotted encoded by color. The blue line corresponds to the case of balanced excitatory and inhibitory. Black circles denote the inhibitory rates that reduce the shift closest to 0. **Right:** Activation functions with shift-compensating inhibitory rates (solid lines) compared to balanced noise input (dashed lines) for different rates: 0.4 (blue), 2 kHz (orange) and 9 kHz (green). The activation function corresponding to the reference rate at 2 kHz, stays unchanged. The other two functions are shifted by adjusting the inhibitory noise rate until the inflection points overlap at a spike probability of 0.5, marked with a dashed gray line. Figure is taken from *Korcsak-Gorzo* (2017).

## 3.3. Spike-based tempering

In this section, based on the noise to temperature mapping, we develop LIF networks with background Poisson noise of varying rates and apply them for generation tasks on the MNIST handwritten digits (*LeCun*, 1998). We first describe the design of the rate variation schemes.

### 3.3.1. Rate variation schemes

As a reference of the rate variation scheme, we take AST (*Salakhutdinov*, 2010) described in section 2.1.4 where the inverse temperature $\beta$ of the system varies between 0 and 1 (see Eq. 2.16). Presumably, the rate variation range needs to cover a certain region of the temperature values after mapping, with higher rates (corresponding to low inverse temperatures) facilitating mixing and smaller rates stabilizing the generated pattern. However, changing the noise rates influences the sampling approximation of the LIF network as the neuron needs high-conductance state for diffusion approximation (see section 2.2). Intuitively, the sampling accuracy decays when the noise rates become too small. To find an appropriate range of noise rates, we need to study the quality of different noise rates in terms of sampling accuracy.

We construct a small LIF-based RBM with 10 neurons and measure the sampling accuracy using the Kullback-Leibler divergence (DKL). The weights and biases of the RBM are randomly drawn from a beta distribution, whose probability density function is expressed as

$$f(x;\alpha,\beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1} \tag{3.22}$$

$$B(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt \tag{3.23}$$

where $\alpha$ and *beta* are the non-negative shape parameters, $B(\alpha,\beta)$ is the normalization function which ensures the integral over the total probability equals 1. The uniform distribution corresponds to an $\alpha$ and $\beta$ of 1. By changing these parameters, the probability mass of the distribution can be modulated. Here, we follow the setup in *Petrovici et al.* (2016) (see Fig. 3.8) which reads:

$$W, b \sim 1.2 \cdot (f(x;0.5,0.5) - 0.5) \tag{3.24}$$

These settings ensure dissimilar distributions comprising several orders of magnitude and a linear projection on values in [-0.6, 0.6].

The simulation result can be seen in Fig. 3.9. Each noise rate is simulated with 10 different random seeds for initialization. The DKL time course with Gibbs sampling on the traditional RBM is included for comparison, which converges to the target distribution for an infinite time. Opposed to that LIF sampling converges to a higher value since it only approximates sampling. We observe that smaller rates lead to higher DKL

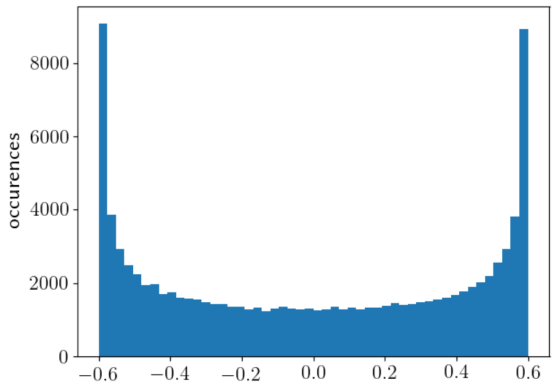


Figure 3.8.: Histogram of the beta distribution from which the network parameters are sampled from. The histogram is retrieved from $10^5$ samples. Figure is taken from *Korcsak-Gorzo* (2017).

values following our expectation, and rates from 400 Hz converge to similar small values. Based on this, we select 400 Hz as the lower boundary of the chosen rate range and 10 kHz as the upper.

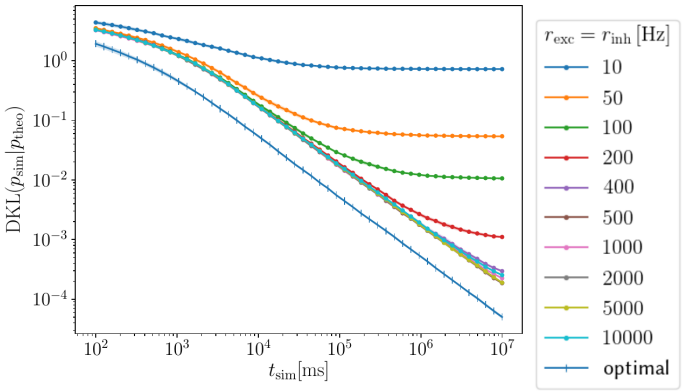


Figure 3.9.: DKL time course for a 10 unit LIF-based BM stimulated with 10 rates between 10 Hz and 10 kHz. Gibbs sampling is included as a reference. The lines correspond to the mean value from 10 different random seed initializations. For rates from 400 Hz to 10 kHz, the converged lines are sufficiently close, which establishes the range for the experiments. Figure is taken from *Korcsak-Gorzo* (2017).

Inspired from the wave patterns in neural oscillations, we adopt a sinusoidal rate variation scheme which ensure that it will oscillate around the reference rate (corresponding to $\beta = 1$) and visit both higher and lower rates. The sinusoidal scheme is

defined as

$$y = a \cdot \sin[b(x + c)] + d \tag{3.25}$$

where $d$ is the shift along the y-axis, $a$ is the amplitude of the sine function, $c$ is the phase and $b$ is the scale factor along the x-axis and encodes the period length. In practice, since the PyNN-NEST simulator requires a certain duration for a certain rate (minimum duration is 0.1 ms), we use a discretized sine function and approximate it with stepwise constants. An example is depicted in Fig. 3.10 left.

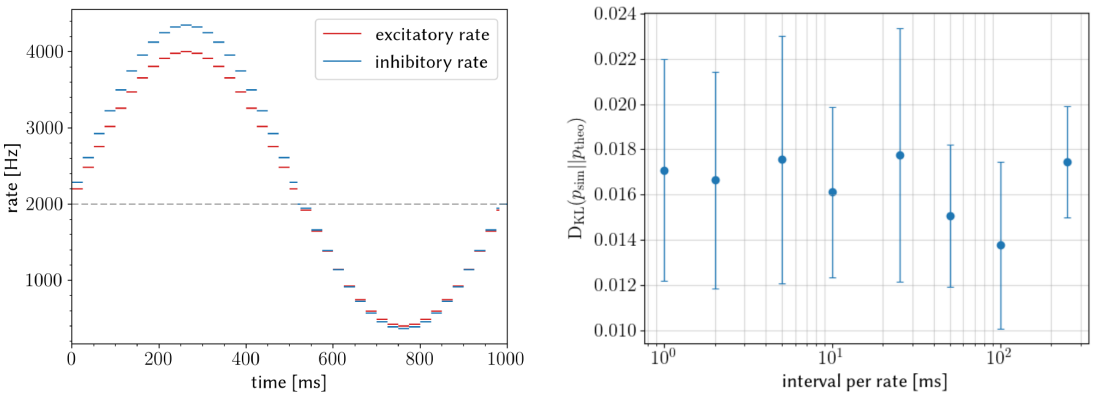


Figure 3.10.: **Left:** An example of the sinusoidal variation of excitatory and inhibitory Poisson noise rates between 0.4 and 4 kHz with a period length of 1s (corresponding to *beta* variation of approximately 2 to 0.5). The corresponding inhibitory rate is adjusted accordingly to compensate for the shift of activation function as described in section 3.2.3. At the reference rate (corresponding to $\beta = 1$) of 2 kHz inhibitory and excitatory rates overlap. The length of the red and blue dashes indicates how long the respective rate is present. The step size here is 25 ms. **Right:** Sampling accuracy of the LIF network under varying noise with different discretized step sizes. We stimulate a LIF-based RBM with noise of sinusoidal varying rates, which is discretized linearly in time in a 1000 s long simulation. The minimal rate is at 400 Hz and the maximal rate at 10 kHz. The DKL between the simulated and the theoretical joint distribution is plotted over step sizes 1, 2, 5, 10, 25, 50, 100 and 250 ms. The standard deviations (bars) are gained from 10 different random seed initializations. The DKL values are distributed in a close range leading to the conclusion that the step size is not critical for sampling accuracy. Figure is taken from *Korcsak-Gorzo* (2017).

To test the sampling accuracy of the LIF network under noise with varying rates, we measure the DKL of a small network of 5 neurons. We compare the theoretical with the simulated joint distribution, once over all rates and once specifically at the reference rate. For the theoretical distribution, we calculate the target distribution for different rates by mapping them to corresponding temperatures, and finally make an average for

each state configuration for the histogram. The results are depicted in Fig. 3.11. In both cases, simulation and theory are in good accordance.

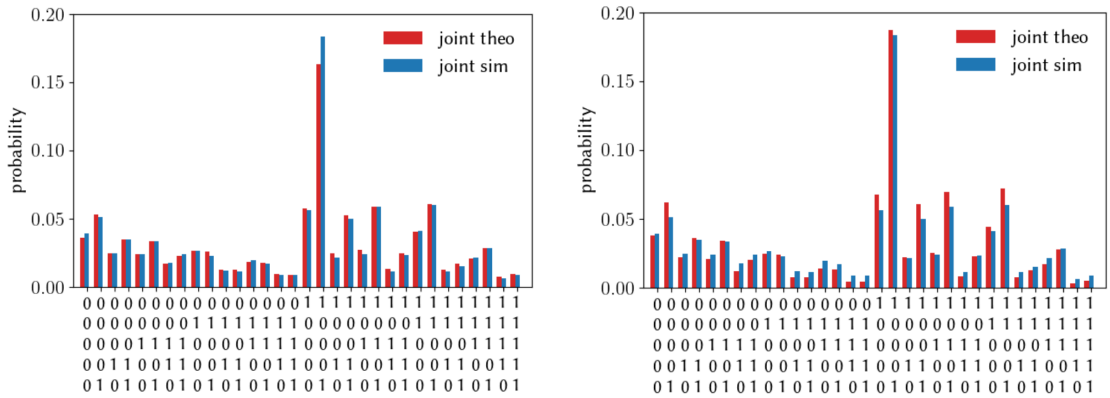


Figure 3.11.: Comparison between simulated and theoretical joint distribution. The joint distribution shows the probability of all possible state permutations of the binary states of the 5 samplers. The states are retrieved from a single simulation of a 5-unit LIF-based RBM, stimulated with sinusoidally varying Poisson noise. The minimal rate is 400 Hz, the maximal rate is 10 kHz and the reference rate is 2 kHz. 100 sine periods are simulated with equal step size of 10 ms. **Left:** Averaged over all rates in the simulated distribution and all $\beta$'s in the theoretical distribution. The simulated distribution approximates the theoretical well. **Right:** Specifically for the reference rate corresponding to $\beta = 1$ , i.e., 2 kHz. The simulated distribution approximates the theoretical well. Figure is taken from *Korcsak-Gorzo* (2017).

We further investigate the effects of different step sizes on the sampling accuracy of the network. A LIF-RBM is simulated with noise of sinusoidal varying rate discretized linearly in time in a 1000 second long simulation. The sine period is set to 1 second. Results (see Fig. 3.10 right) show that step sizes in a certain range (from 1 to 250 ms) leads to similar DKL values.

### 3.3.2. Image generation tasks

In this section, we apply the spike-based tempering approach in image generation tasks and compare its performance with other sampling methods. We construct an RBM with 784 visible and 400 hidden units and train it with the CAST algorithm (for details of the training see appendix) on 1000 samples (100 random samples from each class) draw from the MNIST handwritten digit dataset. The trained network parameters are then mapped to the LIF domain. Three other approaches are used for comparison, i.e. RBM with Gibbs sampling and AST algorithm, and LIF-based RBM with homogeneous Poisson noise.

The results of RBMs are plotted in Fig. 3.12. 100 consecutive samples are separately obtained from each sampling method. The pixels of generated images are calculated as the firing probabilities of visible units from the states of hidden units. The images generated by Gibbs sampling basically stay in class '1' and slowly transfer to class '2' in the end, showing slow mixing speed. In the case of AST, images are distributed in several classes, demonstrating the mixing-facilitation ability of the algorithm. Notice that the images are sometimes blurred during transitions between classes, which is expected since the network is traveling at the energy barriers corresponding to low probability region. We also plot a sequence of temperatures during the evolution process, from which one can see that the temperature changes adaptively with fluctuation. Though mixing faster, AST comes with much higher computational cost than Gibbs sampling: all samples other then those with temperature at 1 are discarded.

For the LIF-based RBM with homogeneous Poisson noise, we simulate the network with 2 kHz noise for 100 seconds with the first second as burn-in. In the LIF-sampling framework, one sampling step is defined as the duration of one refractory period. Therefore, after the simulation, 100 samples are gathered with a sample interval of 100. For spike-based tempering (LIF-based RBM with sinusoidal noise input), we set the minimum noise rate at 1.5 kHz, the maximum rate at 3 kHz, and the reference rate at 2 kHz where samples are taken. The sine period is 4000 ms and the network is simulated for 100 periods to gather 100 images. The results are plotted in Fig. 3.14. Images produced with homogeneous Poisson input noise are quite clear, but all reside in class '0', showing bad mixing. In contrast, images produced with spike-based tempering are much more diverse. Moreover, the image quality is even slightly better at mode transition compared with AST, presumably due to the smooth variation of noise rate.

### 3.3.3. Optimal tempering parameters and the measurement of mixing

To further investigate the influence of rate variation parameters on the generation performance, we run multiple simulations with a sweep over different parameter configurations, i.e. the sine period, the minimum and the maximum noise rate. To have a more quantitative comparison of mixing between different methods, we use the so-called indirect sampling likelihood (ISL) method (*Breuleux et al.*, 2010; *Desjardins et al.*, 2010b). ISL constructs a non-parametric density estimator to evaluate how close each test example is from any of the generated examples. The likelihood of a test sample $\mathbf{y}$ given a series of generated sample $\{\mathbf{x}_i\}$ is defined as:

$$p(\boldsymbol{y}) = \frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{d}\beta^{1_{\boldsymbol{y}_j=\boldsymbol{x}_{ij}}}(1-\beta)^{1_{\boldsymbol{y}_j\neq\boldsymbol{x}_{ij}}} \quad , \tag{3.26}$$

where $N$ is the number of generated samples, $d$ is the dimension of $\boldsymbol{y}$ and $\boldsymbol{x}_i$, and $\beta$ is a hyperparameter which controls the gain ($\beta$) and punishment ($1-\beta$) to the likelihood when comparing the test sample with the generated sample. In practice, all images are first binarized with a threshold of pixel value 0.5 for calculation.

We set the training set as $\{\mathbf{y}\}$ and calculated the average ISL value over all the training samples. Intuitively, the higher the averaged ISL value within the early sampling

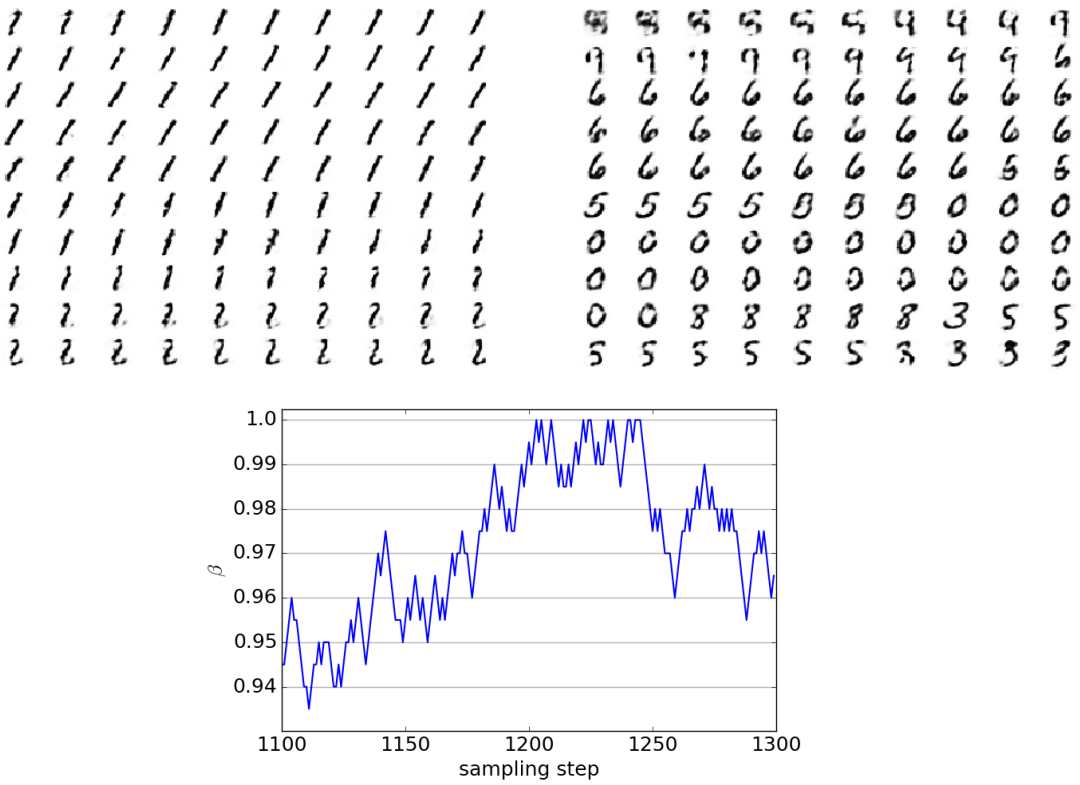


Figure 3.12.: **Top left:** 100 consecutive images (per row) produced by Gibbs sampling. The Gibbs sampling is run for 200 steps and the first 100 sampling steps are discarded as burn-in. The images mainly reside in class '1' and slowly transfer to class '2', showing bad mixing. **Top right:** 100 consecutive images (per row) produced by AST. In total 200 samples are obtained and the first 100 samples are discarded as burn-in. The network is able to switch between multiple image classes, showing fast mixing speed. **Bottom:** A sequence of temperature trace of AST during the evolution process. Samples are gathered when the temperature reaches 1, otherwise discarded. 20 temperatures are used here ranging from 0.9 to 1 with equal space.

process, the better the mixing. We therefore take the first 1000 generated samples as $\{\mathbf{x}_i\}$. We set $\beta = 0.95$ to optimize the likelihood; other values ($\beta \in (0.5, 1]$) would rescale the likelihood but without causing qualitative differences.

The result is shown in Fig. 3.14. For sine periods of 200, 300, 400, 500 and 600 ms, a scan of possible rate configurations with maximums from 3 kHz to 8 kHz (1 kHz as interval) and minimums from 1 kHz to 1.8 kHz (0.2 kHz as interval) are made. The reference rate is set at 2 kHz. The final ISL value is encoded with color and each square is an average over 10 simulations with different random seeds. The result shows that the

Figure 3.13.: **Left:** 100 images (per row) produced by LIF-based RBM with homogeneous Poisson noise. The images are obtained with a sample interval of 100. The network gets stuck in the "0" mode, showing poor mixing. **Right:** 100 images (per row) produced by spike-based tempering. The sine period length is 4000 ms, the minimum of the sine wave is at 1.5 kHz (corresponding to $\beta \approx 1.1$) and the maximum at 3 kHz (corresponding to $\beta \approx 0.8$). The reference rate is at 2 kHz. The handwritten digits are clear and distributed among multiple classes, showing good mixing. Figure is taken from *Korcsak-Gorzo* (2017).

optimal sine period is at 400 ms. Across all sine periods, the influence of the maximum noise rate is more significant than the minimum noise rate, with the optimum value at 8 kHz corresponding to $\beta \approx 0.5$. The result is in a way in accordance with our empirical finds with AST, that networks of relatively small size would need larger temperature variations for better mixing.

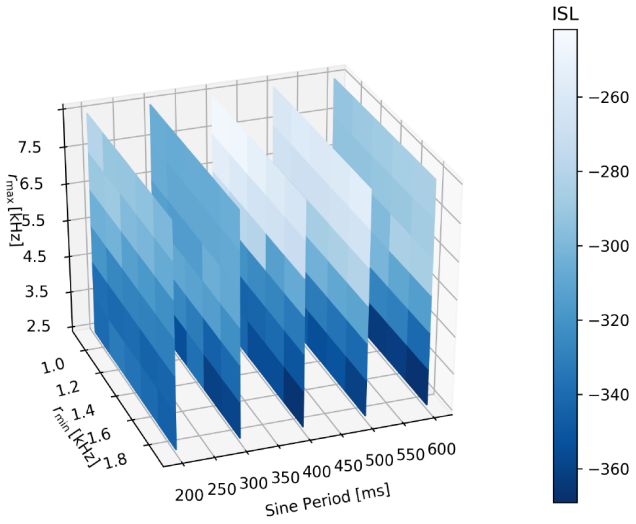


Figure 3.14.: ISL values of multiple simulations with different sine periods, the minimum and the maximum noise rate. The sine period ranges from 200 to 600 ms with 100 ms as an interval. The maximum noise rate ranges from 3 kHz to 8 kHz with 1 kHz as interval, corresponding to a $\beta$ range of approx. 0.8 to 0.5. The minimum noise rate ranges from 1 kHz to 1.8 kHz with 0.2 kHz as interval, corresponding to a $\beta$ range of approx. 1.4 to 1.1. The value of each square is an average of 10 simulations with different random seeds. The discretized step size of the sine wave is fixed to 10 ms. Figure is taken from *Korcsak-Gorzo* (2017).

## 3.4. Discussion

In this chapter, based on the membrane potential dynamics described in the LIF-sampling theory, we developed the spike-based tempering approach which improves the mixing capability of the network in high dimensional space. We studied the relation between temperature and the rate of the Poisson input noise, based on which we derived a mapping equation. Inspired from neural oscillations, we designed a sinusoidal rate variation scheme of the input noise during the sampling process and further applied the LIF network to generation tasks along with other methods as comparisons. The results showed significant improvement in the mixing of spike-based tempering, competitive to sophisticated machine learning algorithms. Finally, in order to search for optimal parameter configurations, we performed multiple simulations with a sweep on rate variation parameters including sine period and rate changing scales, during which we used the ISL method for quantitative measurements of mixing.

Although we have demonstrated the mixing ability of spike-based tempering in an image generation task (section 3.3.2), its pre-simulation defined rate variation scheme is very different from the mixing-facilitating principle in traditional tempering algorithms.

In AST, the Markov chain samples from the joint distribution of the state and temperature $p(\mathbf{x}, k)$, and guarantees its convergence to the target distribution $p(\mathbf{x})$ by only keeping samples obtained at $k = 1$. Meanwhile, its mixing efficiency is maintained by adaptively changing the adaptive weight $\{g\}$ of state space partitions. Further experiments are needed to investigate the convergence property of spike-based tempering, i.e., what stationary distribution it will converge to under different rate variation schemes, or, even whether it will converge eventually? To answer these questions, one can first perform multiple simulations on small size networks and measure the DKL value between the simulation and theoretical distribution.

Our sinusoidal rate variation scheme is inspired from neural oscillations observed in mammalian cortex (*Buzsáki and Draguhn*, 2004), which are separated into several bands covering frequencies from approximately 0.05 Hz to 500 Hz (see Fig. 3.15). This

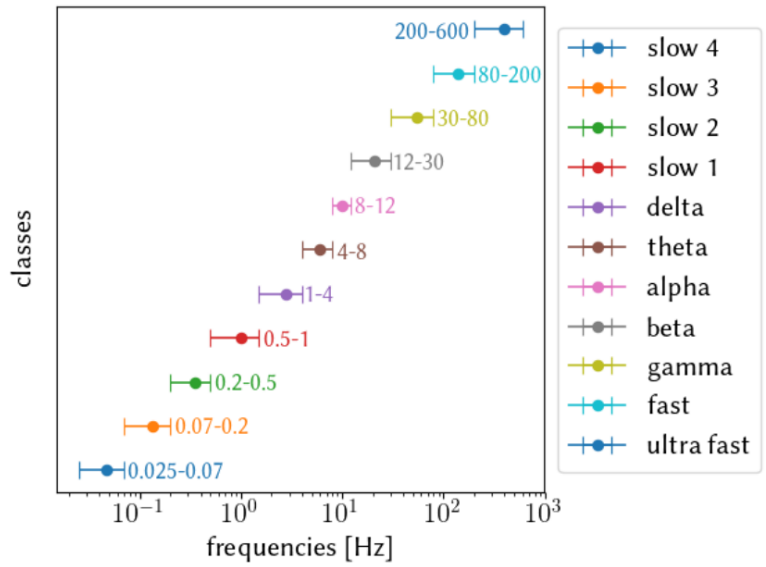


Figure 3.15.: Oscillatory classes in the rat cortex. For each band, the range of frequencies is shown, together with its commonly used term. Image is adapted from *Buzsáki and Draguhn* (2004).

oscillatory behavior on membrane potentials or local field potentials are suggested to be related to functional neural activities such as input selection, neuronal assemblies formation, synaptic plasticity and long-term consolidation of information (*Buzsáki and Draguhn*, 2004). The oscillation frequencies in our experimental setup are within the range suggested from biology. However, we directly implemented a background Poisson noise with oscillatory rates without specifying how it is generated and the underlying biophysical basis. Recent work (*Dold et al.*, 2018) proposed that the homogeneous Poisson noise required for LIF sampling can be generated from the functional output of

spiking networks. Future works can study the feasibility of introducing oscillation into the output of these networks.

# 4. LIF networks with short-term synaptic plasticity

The discriminative capacity of the neocortex is well-established, as evidenced by the difficulty of artificial systems to achieve superhuman classification performance *Schmidhuber* (2015). Simultaneously, however, the brain also appears to learn a generative model of its sensory environment (*Fiser et al.*, 2010; *Jezek et al.*, 2011; *Hindy et al.*, 2016). How these capabilities are achieved remains an open question. In the previous chapter, we introduced the spike-based tempering approach inspired by traditional tempering algorithms which significantly improves the mixing of the network by modulating the global energy landscape. However, analogous to its traditional counterparts, samples at the target distribution can only be obtained when the system evolves to the base temperature (or base noise rate), otherwise they are wasted, therefore, the system will need longer time to obtain the same number of valid samples, compared to plain MCMC sampling algorithms.

One mechanism that is capable of modulating synaptic efficacy and thereby shaping the probability landscape of a neural network is short-term plasticity (STP). The activity-dependent nature of STP enables it to adaptively changing the connection strength of subpopulations, which is potentially more computationally efficient than the global change of tempering methods. Throughout this chapter, we investigate the ability of this biologically ubiquitous mechanism to improve the mixing capabilities of generative neural networks. Furthermore, we show how hierarchical LIF networks endowed with STP can simultaneously become good discriminative and generative models, a feature that is difficult to achieve due to the conflicting nature of these two tasks.

We first give a brief introduction about the information transmission process on the synapse and the corresponding STP model (section 4.1). Under the context of LIF sampling, we develop synapses of specific functionalities with different configurations of STP parameters.

In section 4.2, we construct LIF-based RBMs with STP and study their performances on various tasks. We start by discussing how STP can improve the sampling accuracy of small networks configured to sample from a fully specified target distribution where mixing is easy (section 4.2.1). This is no longer the case when networks are trained on multimodal datasets. In the case of a bar experiment (section 4.2.2) we compare the mixing performances between LIF sampling with STP and traditional Gibbs sampling. We then train the network on the MNIST benchmark datasets in which we study the influence of STP on both its generative and discriminative properties (section 4.2.3). With

quantitative measurements on mixing, we demonstrate the advantage of LIF networks with STP in generative tasks. In the end, as preliminary functional applications, we show how STP can aid balance sampling and pattern completion when networks are trained on highly imbalanced datasets (section 4.2.4).

In section 4.3, based on previous experiments, we further study the influence of STP on the probability distribution of network states and demonstrate its local modulation effect on active attractors and inhomogeneous modification on the energy landscape.

A large part of this work has been done in collaboration with Mihai A. Petrovici, some of the contents have already been included in publications (*Leng et al.*, 2016, 2018). Others involved in this work include Roman Martel who performed early studies of the generative properties of spiking networks (*Martel*, 2015), Oliver Breitwieser who developed the spike-based-sampling (SBS) module (*Breitwieser*, 2015), a software module based on NEST which enabled faster, larger-scale simulations and Ilja Bytschok who was involved in the early analysis of results and discussions that shaped the study.

The neuron parameters for all simulations in this chapter are described in appendix A.2.1 unless specifically mentioned. All details regarding the training of networks can be found in appendix A.2.2.

## 4.1. STP and its functional application in sampling

As mentioned in section 2.2.3, short-term depression (STD) is implemented in the LIF-sampling framework to guarantee the approximation accuracy of sampling dynamics. In this section, based on the signal transmission process of the synapse, we introduce the short-term plasticity (STP) mechanism and corresponding models. We then demonstrate how STP modulates the synaptic efficacy based on the spiking frequency of the presynaptic neurons and develop specific PSP envelopes for functional use in generative models.

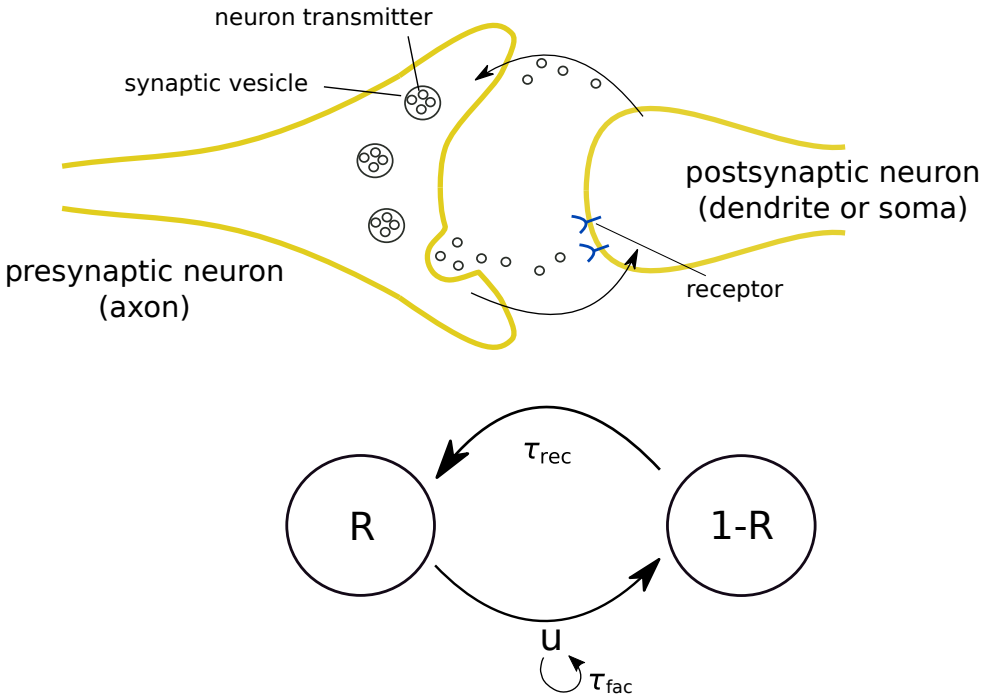


Figure 4.1.: **Top:** A rough plot of information transmission in the chemical synapse. The presynaptic neuron transmits electrical signals via chemical messengers called neurotransmitters which are stored and released from synaptic vesicles by fusion with the presynaptic membrane. They then diffuse into and across the synaptic cleft which is approximately 20-40 nm wide and are recycled back for reprocessing after binding with receptors located on the membrane of the postsynaptic neuron. **Bottom:** The Tsodyks-Markram mechanism models this process in a simplified way, with $R$ representing the available synaptic resources, $\tau_{\text{rec}}$ representing the exponential recovery time constant of neurotransmitters, $u$ representing the utilized fraction of neurotransmitters upon each spike and $\tau_{\text{fac}}$ representing its exponential decay time constant during facilitation.

In the nervous system, neurons transmit electrical signals through synapses[1], which are structures formed usually between the axon of the presynaptic neuron and the dendrite or soma of the postsynaptic neuron (see Fig. 4.1). When an electrical signal or a spike is generated at the axon, it produces an influx of calcium ions triggering the release of neurotransmitters from the plasma membrane of the presynaptic neuron, which are chemical molecules like glutamate or choline stored in synaptic vesicles. They diffuse across the synaptic cleft and bind to the receptors located on the membrane of the postsynaptic neuron which triggers flux of certain ions to generate postsynaptic potentials (PSPs). After binding the neurotransmitters are recycled back to the presynaptic site for reprocessing.

### 4.1.1. Tsodys-Markram model

Simplified from the complex biophysical process of synaptic transmission, *Tsodyks and Markram* (1997); *Markram et al.* (1998); *Fuhrmann et al.* (2002) described a phenomenological model of synaptic efficacy depending on the history of presynaptic activity (see Fig. 4.1). This model of STP comprises STD whose underlying phenomenon is the depletion of neurotransmitters consumed during the synaptic signaling process of presynaptic neuron, and short-term facilitation (STF) caused by the influx of calcium after spike generation which increases the release probability of neurotransmitters. The momentary synaptic efficacy is reflected in the size of the elicited PSP

$$PSP \propto w \cdot U \cdot R \tag{4.1}$$

where $U$ and $R$ are described by

$$\frac{\mathrm{d}R}{\mathrm{d}t} = \frac{1-R}{\tau_{\mathrm{rec}}} - U \cdot R \cdot \delta(t - t_s) \tag{4.2}$$

$$\frac{\mathrm{d}U}{\mathrm{d}t} = -\frac{U}{\tau_{\mathrm{fac}}} + U_0 \cdot (1-U) \cdot \delta(t - t_s) \quad . \tag{4.3}$$

Here, $w$ represents the (static) synaptic weight and $U \in [0,1]$ the utilized fraction of available synaptic resources $R \in [0,1]$. Upon the arrival of a presynaptic spike at time $t_s$, the synapse is depressed by subtracting $U$ from $R$, which recovers exponentially with the time constant $\tau_{\mathrm{rec}}$. Facilitation is modeled by a pulsed increase in $U$ by the amount of $U_0(1-U)$, followed by an exponential decay with a time constant $\tau_{\mathrm{fac}}$. Notice that the model in *Tsodyks and Markram* (1997) only captures STD, the following work in *Markram et al.* (1998) further extends the model to include STF and describes the following discrete equation on which our simulations are based[2]:

$$R_{n+1} = R_n(1 - U_{n+1})\exp\left(\frac{-\Delta t}{\tau_{\mathrm{rec}}}\right) + 1 - \exp\left(\frac{-\Delta t}{\tau_{\mathrm{rec}}}\right) \tag{4.4}$$

$$U_{n+1} = U_n\exp\left(\frac{-\Delta t}{\tau_{\mathrm{fac}}}\right) + U_0\left(1 - U_n\exp\left(\frac{-\Delta t}{\tau_{\mathrm{fac}}}\right)\right) \tag{4.5}$$

[1] Here, we only discuss chemical synapse.

[2] In practice, we take $R_1$ as 1 different from the $1 - U_0$ in the original paper, for consistency with equation 4 in the paper.

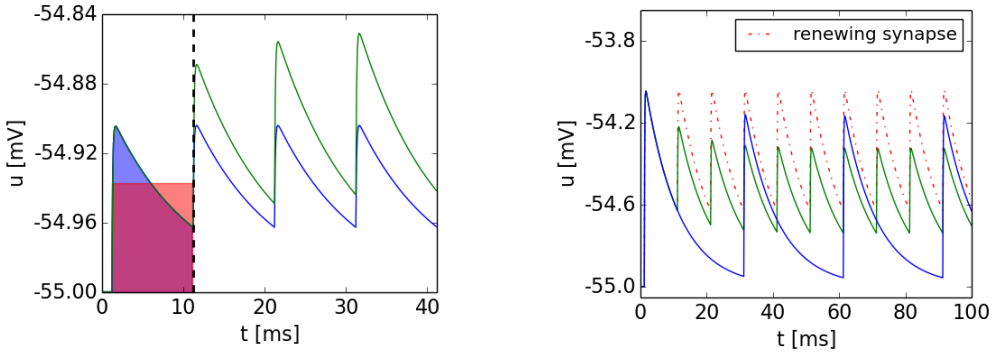


Figure 4.2.: **Left:** The mapping between conventional RBM weight and neural synaptic strength is achieved by making the area below the rectangular PSP (pink) and exponential PSP (blue) to be equal. Short-term depression (STD) maintains the LIF PSP height (blue), renewing the synapse and keeping the synaptic efficacy constant. Without STD, consecutive PSPs will exceed the height of the first one due to the accumulation on the exponential tail of the previous kernel. **Right:** STD modulates synapse according to the spiking frequency of the presynaptic neuron. With decreasing spiking frequency (blue), the modulation effect start to vanish and the amplitude of consecutive PSPs recovers towards the level of the renewing synapse.

### 4.1.2. Renewing synapse and modulated synapse

By modulating synaptic interactions, STP shapes the sampled distribution. This can be helpful when a spiking network needs to approximate a distribution that is otherwise incompatible with biological neuro-synaptic dynamics, as we discuss in the following.

In the case of LIF-based BMs, when a neuron needs to continuously represent a state $z_k(t) = 1$ for an extended period, it fires a sequence of $n$ spikes at maximum frequency $1/\tau_{\text{ref}}$. Following equation 3.7, the resulting PSPs should increase a postsynaptic neuron's membrane by a constant $\Delta u_i = W_{ji}$, which implies a rectangular PSP shape. However, this is not a realistic shape for a more biologically plausible scenario, where PSPs have an exponentially shaped decay. This causes them to accumulate (Fig. 4.2 left), such that the average increment $\langle \Delta u_i \rangle_n$ becomes a function of the burst length $n$, thereby distorting the sampled distribution.

STD can mitigate this effect (Fig. 4.2 left) by causing a gradual decrease in the amplitude of consecutive PSPs. By setting $\tau_{\text{rec}}$ to values close to $\tau_{\text{ref}}$, we can create a renewing synapse which maintains the average synaptic efficacy as a constant (see section 4.2.1 for more details), thus fulfill the requirement for sampling approximation. Since both tempering and STP effectively modify the energy landscape by changing network parameters during sampling, they clearly bear some conceptual resemblance. However, while tempering simultaneously affects all synaptic weights, STP only affects the efferent connections of those neurons with high firing rate, and the modulation effect decreases

when the firing rate gets lower (Fig. 4.2 right). Therefore, in contrast to the global modifications of the energy landscape incurred by tempering, STP has a more local effect focusing on active attractors, as sketched in Fig. 4.3. For applications on high-dimensional datasets, we use STP to create a modulated synapse (Fig. 4.3 top) with potentiation-depression envelope which enables the network to produce clear patterns during potentiation and escape from local energy minimum during the depression (see section 4.2.3 for more details).

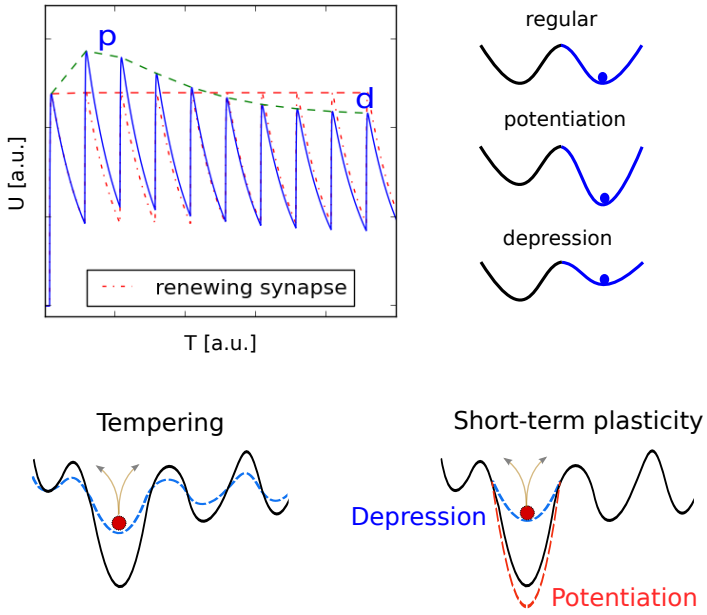


Figure 4.3.: **Top:** PSPs with potentiation-depression pattern first strengthen, then weaken the effective interaction. The potentiation-depression changes the energy landscape, first deepening the energy trough and sharpening the produced image, followed by a local flattening of the energy trough which pushes the network state into a different mode. **Bottom:** To facilitate mixing, tempering methods globally rescale the energy landscape with an inverse temperature (top). In contrast, STP can be viewed as only modulating the energy landscape locally, thereby only affecting the currently active attractor (bottom).

## 4.2. LIF-based RBMs with STP as generative and discriminative models

Throughout this section, we study the effects of STP on the performance of LIF-based RBMs trained for different tasks. We start by discussing how STP can improve the sampling accuracy of small networks configured to sample from a fully specified target distribution (section 4.2.1), even when the energy landscape is shallow enough to not cause mixing problems. We gradually increase the size of the network to model a bar experiment (section 4.2.2) where mixing becomes hard for traditional sampling methods, and demonstrate the mixing advantage of STP-endowed spiking networks. We further extend the network size for training on the MNIST benchmark dataset of handwritten digits (section 4.2.3), in which we study the influence of STP on both generative and discriminative properties. Finally, we show how STP can aid balance sampling and pattern completion when the training datasets are highly imbalanced (section 4.2.4). These experiments are the result of discussions together with Mihai A. Petrovici, Ilja Bytschok, and others, and have already been reported at the *Leng et al.* (2016), as well as published in *Leng et al.* (2018).

### 4.2.1. Sampling from a fully specified target distribution

As discussed in section 4.1.2, by modulating the amplitude of consecutive PSPs, STP can be helpful in maintaining a constant average synaptic efficacy and thus accurately approximating the sampling process. We verify this theory by constructing a 10-neuron (5 hidden, 5 visible) LIF-based RBM and study its sampling performance under different STP parameters. We use a target Boltzmann distribution $p_{\mathrm{B}}(z|W,b)$, with parameters drawn from a Beta distribution

$$W, b \sim 1.2 \cdot (f(x; 0.5, 0.5) - 0.5) \tag{4.6}$$

which is the same as Eq. 3.24, that produce a diverse energy landscape but not so rough as to create problems with mixing, similar to the approach in 3.3.1.

We simulate the LIF-based RBM with a sweep over the $(U_0, \tau_{\mathrm{rec}}, \tau_{\mathrm{fac}})$ parameter space. For parameter sets with $U_0 < 1$, the weights of the network are rescaled by a weight dividing factor $f_w = U_0$ to maintain the amplitude of the first PSP:

$$W = \frac{W}{f_w} \tag{4.7}$$

Each simulation is initialized with 5 different random seeds and run for 4.8 $\times 10^6$ms until convergence. After simulation, we calculate the averaged DKL between the sampled and target distribution over all random seeds. The results are plotted in Fig. 4.4. We find that an optimal reproduction of the target distribution is achieved for $\tau_{\mathrm{rec}} \approx 15\,\mathrm{ms}$ (Fig. 4.4 left bottom), which is close to the synaptic time constant of $\tau_{\mathrm{syn}} = 10\,\mathrm{ms}$. This affords an intuitive explanation: In the HCS, the effective membrane time constant becomes small and $\tau_{\mathrm{syn}}$ dominates the PSP decay. If the recovery of synaptic resources $R$ (equation 4.2)

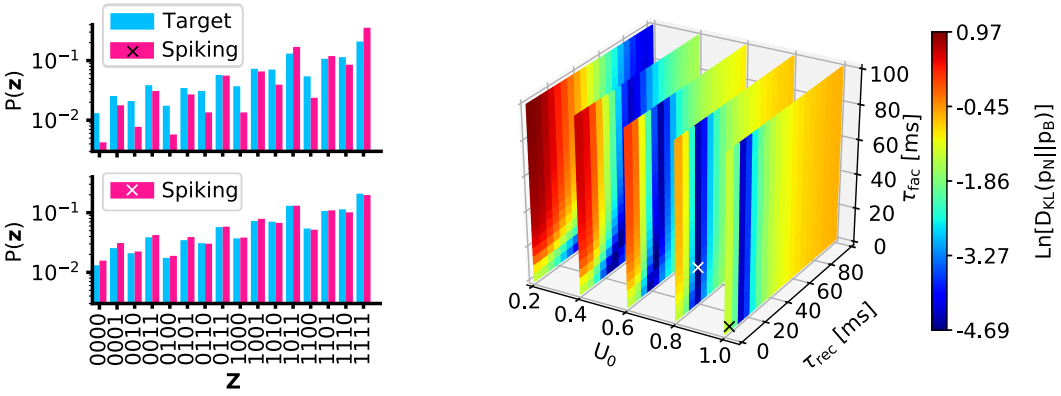


Figure 4.4.: **Left:** Distribution sampled by the LIF-based RBM for two different configurations of synaptic parameters. Depressing synapses (bottom) allow the network to come much closer to the target distribution (blue) than non-plastic ones (top). The colored crosses in labels indicate their corresponding positions in the right plot. Note that we plot here the joint distribution of four neurons randomly selected from the network. **Right:** Kullback-Leibler divergence between sampled ($p_N$) and target ($p_B$) distribution of the LIF-based RBM with 10 neurons (5 hidden, 5 visible) for different STP parameters ($U_0, \tau_{rec}, \tau_{fac}$). Note that many different parameter combinations lead to close to optimal (white cross) sampling, but static synapses (black cross) are not among them. Figure is taken from *Leng et al.* (2018).

happens at the same speed as the PSP decay, the STP mechanism essentially emulates a renewing synapse with an approximately constant running average (Fig. 4.2 left). The slightly larger optimal recovery time constant further compensates for the long tails of exponential PSPs, which potentiate interaction strengths compared to the ideal case of rectangular PSPs. Note that the manifold for which the target distribution is close-to-optimally reproduced contains many different STP configurations, including the range of biologically observed parameters (*Wang et al.*, 2006), but not the $(u, \tau_{rec}, \tau_{fac}) = (1, 0, 0)$ triplet for static synapses (Fig. 4.4 left top).

In the above experiment, training is not needed, as synaptic weights of the LIF network can be computed directly from the parameters $W$ and $b$ of the Boltzmann distribution (Eq. 2.29, 2.30). This changes when the network parameters are learned from data, as we discuss in the following.

### 4.2.2. Mixing in a simple learning scenario

As discussed in section 2.1.3, when learning from high-dimensional multimodal datasets, traditional MCMC sampling algorithms such as Gibbs sampling are prone to get trapped in local minima due to high energy barriers, which is known as the mixing problem. This can be improved with STP which adaptively modulates the local active attractor during

the sampling process, as we proposed in section 4.1.2. We demonstrate it with the following experiments.


Figure 4.5.: Training data for the easy (top) and hard (bottom) learning scenario in grayscale. The 3 images from the training set, each containing a single oriented bar, are superimposed to highlight the overlap of the oriented bars (or lack thereof). Note the actual pixel value (for training) of black pixels is 0 and for white pixels is 0.5. Figure is adapted from *Leng et al.* (2018).

Borrowing from observations in the early visual system, we generate 2 sets of oriented bar images, with 3 bars in each set (Fig. 4.5). Each bar is a $20 \times 20$ pixels grayscale image where the value of the background pixel is 0 and otherwise 1. The bars are positioned in a way that gave rise to an "easy" (overlapping) and a "hard" (non-overlapping) dataset. We then train a LIF-based RBM (400 visible, 30 hidden units) on each of these datasets using CAST (*Salakhutdinov*, 2010) algorithm.

Intuitively, the difficulty of learning a generative model of this data increases when the bars have little or no overlap: in this case, training gives rise to three nearly disjoint populations that have, on average, excitatory connections within and inhibitory connections between them. The emergence of such a population-based winner-take-all structure can be characterized by the mean interaction strength $\bar{w}_{ij} = \langle z_i^{\mathrm{T}} \rangle W \langle z_j \rangle$ between two population activity vectors $\langle z_i \rangle$ and $\langle z_j \rangle$, which represent the average network activity during the presentation of the $i$th and $j$th input pattern, respectively. In practice, we calculate $\langle z_i \rangle$ by running the network (with classical Gibbs sampling or LIF-sampling with STP) for 5000 sampling steps[3] with the visible layer clamped to corresponding training image, and average the neural activities over the number of collected samples.

In the case of Gibbs sampling, for the easy dataset, learning give rise to a mean within-population interaction strength of $\langle \bar{w}_{ii} \rangle_i = 92.75$ and a mean between-population interaction strength of $\langle \bar{w}_{ij} \rangle_{i \neq j} = -145.48$. These values change to $\langle \bar{w}_{ii} \rangle_i = 102.82$ and $\langle \bar{w}_{ij} \rangle_{i \neq j} = -164.66$ for the hard dataset, reflecting the increased competition and disjointedness between the three emerging populations. For the LIF network, we use an STP parameter set of $(U_0 = 1, \tau_{\mathrm{rec}} = 19\,\mathrm{ms}, \tau_{\mathrm{fac}} = 0\,\mathrm{ms})$ (see Fig. 4.6) to create

[3]For the LIF-sampling framework, one sampling step is defined as the duration of one refractory period.

STD in the sampling process. For the easy dataset, we obtain $\langle \bar{w}_{ii} \rangle_i = 54.32$ and $\langle \bar{w}_{ij} \rangle_{i \neq j} = -93.40$. And $\langle \bar{w}_{ii} \rangle_i = 67.74$ and $\langle \bar{w}_{ij} \rangle_{i \neq j} = -113.93$ for the hard dataset. The result shows that STD causes a reduction in the mean within-population interaction and an increase in the mean between-population interaction, which facilitates the network to switch between different modes.

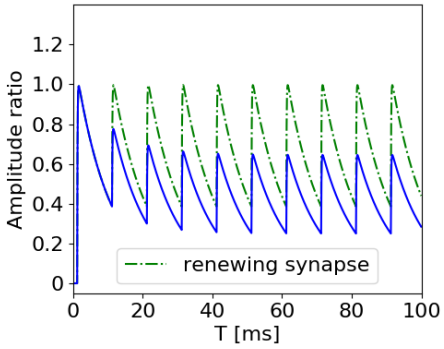


Figure 4.6.: PSPs with STP parameter set of ($U_0 = 1, \tau_{\text{rec}} = 19\,\text{ms}, \tau_{\text{fac}} = 0\,\text{ms}$) (blue) and renewing synapse (dashed green) as comparison. The convergence height of the depressing synapse is about 60% of the renewing one. Note that this ratio is close to the ratio of change in the mean within- and between-population interaction strength observed from the LIF network.

To have a more intuitive comparison, we further plot a sequence of consecutive freely generated samples using Gibbs sampling and the LIF network with STD, as shown in Fig. 4.7. For the easy dataset, both the Gibbs sampler and the LIF network are able


Figure 4.7.: Sequences of images generated by a Gibbs sampler and an STP-endowed LIF network after learning the easy (top) and hard (bottom) cases. For each method, 20 samples are taken from 5000 consecutively generated images with an equal interval. Note that to strengthen pattern visibility, for the easy case, we set all pixel values above 0.07 to be 1, otherwise 0. For the hard case, the threshold value is 0.1.

to mix, although the former spent on average 100 times longer in the same mode before

switching, thereby requiring more time to converge to the target distribution. For the hard dataset, the spiking network retains its ability to mix, whereas Gibbs sampling is unable to leave the (randomly initialized) local mode.

While this simple experimental setup is specifically designed to illustrate the potential problems of sampling-based generative models and the ability of STP-endowed spiking networks to circumvent them, we show in the following that these properties are preserved in more complex scenarios.

### 4.2.3. Generation and classification of handwritten digits

In the previous section, we have demonstrated the ability of STP to facilitate mixing in multimodal distributions created by bar images. The problem of mixing becomes even more pronounced when dealing with larger, more complex datasets. Here, we increase the number of neurons and train a hierarchical 3-layer network with 784 visible, 600 hidden and 10 label units (Fig. 4.8) on handwritten digits from the MNIST dataset *LeCun* (1998) (60,000 training and 10,000 testing 28 × 28 pixels images), which is one of the most widely used benchmark datasets in machine learning. By treating the label units as part of the visible layer during learning, the RBM can be trained in a supervised way. In this way, we simultaneously train a generative as well as a discriminative model of the data. This objective is particularly challenging because mechanisms that improve mixing tend to disrupt classification and vice-versa *Bengio et al.* (2013).

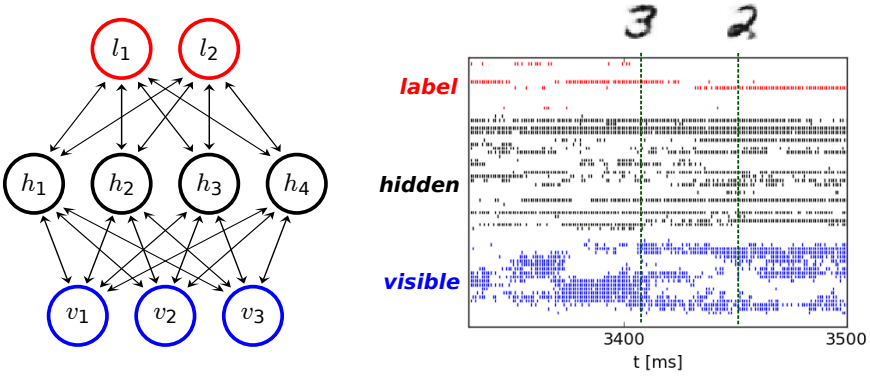


Figure 4.8.: **Left:** A hierarchical LIF-based RBM for classification and generation tasks. The network consisting of 784 visible, 600 hidden and 10 label units is trained on the full MNIST dataset with CAST. **Right:** Selected spike trains from the LIF network: Note the increasing sparseness of the activity in consecutive layers. The network produces different images when the activity in the label layer switches on from one neuron to another.

After training (see section A.2.2 for training details), to evaluate the quality of generated samples, we compute a log-likelihood estimation of 2000 test images (not used during training, with 200 samples randomly selected from each class) using the ISL method mentioned in section 3.3.3. Due to the size of the network, a full scan of the parameter space

for finding optimal STP parameters is no longer feasible. Therefore, starting from a good parameter set found by trial and error, we perform two 2D-scans of the $(U_0, \tau_{\text{rec}}, \tau_{\text{fac}})$ parameter space (Fig. 4.9). As in the previous examples, we find short-term depression to be essential for achieving high ISL values. Furthermore, a small value of $U_0$ combined with short-term facilitation is also beneficial, allowing an initial strengthening followed by a weakening of the active attractor, as sketched in Fig. 4.3. Similar observations have been made in cortex, where STP can promote the enhancement of transients *Abbott and Regehr* (2004).

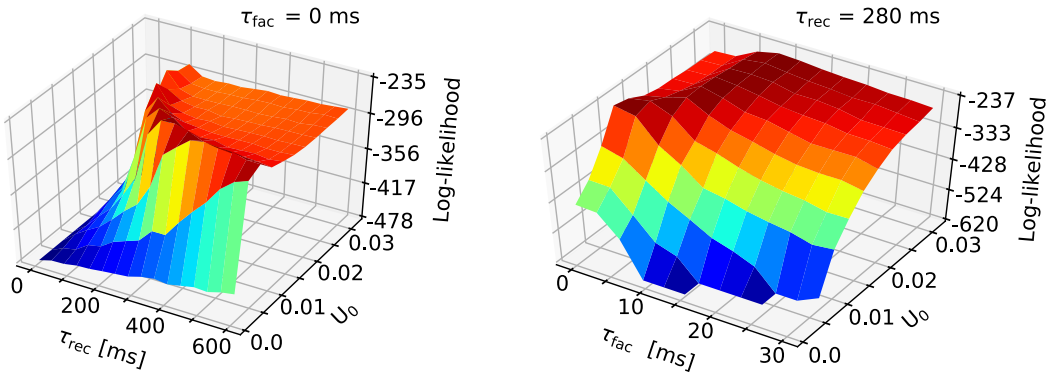


Figure 4.9.: 2D parameter scans of the STP parameters $(U_0, \tau_{\text{rec}}, \tau_{\text{fac}})$ with multiple configurations leading to good generative performance. For each parameter set, the simulation is initialized with 5 different random seeds and run for $10^4$ ms to collect $10^3$ samples, based on which we calculate the mean ISL value. **Left:** Parameter scans of $(U_0, \tau_{\text{rec}})$ with $\tau_{\text{fac}}$ fixed to 0 ms. **Right:** Parameter scans of $(U_0, \tau_{\text{fac}})$ with $\tau_{\text{rec}}$ fixed to 280 ms. Figure is taken from *Leng et al.* (2018).

We use one of the optimal STP parameter sets $(U_0 = 0.01, \tau_{\text{rec}} = 280\,\text{ms}, \tau_{\text{fac}} = 0\,\text{ms})$ (see Fig. 4.10) to compare the generative performance of LIF networks to classical Gibbs sampling. During experiments, we find that the network generates clearer digits when the synaptic efficacy can be maintained for a while before depression. This effect was not significant is the previous section. The intuition behind this phenomenon could be that with the increase of the average number of neurons representing a local minimum, the network needs a longer time to synchronize those neurons into a local attractor. Therefore, as discussed in section 4.1.2, we develop a potentiation-depression synapse by using small $U_0$ values to create a PSP envelope with a plateau in the initial period. To compensate the lose of synaptic efficacy due to small $U_0$, we also need to rescale the weight matrix by an appropriate factor. Further discussions of the influence of the shape of the PSP envelope on mixing performance are in section 5.1.

To observe the mixing performance of the network over time, we plot the mean log-likelihood of 2000 samples from the test set against the number of generated samples

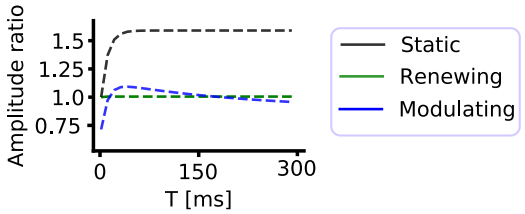


Figure 4.10.: Envelope of consecutive PSPs (with inter spike interval of 10 ms) for three parameter sets $(U_0, \tau_{\text{rec}}, \tau_{\text{fac}})$ from the manuscript: $(1, 0, 0)$ (black), $(1, \tau_{\text{syn}}, 0)$ (green) and $(0.01, 280\,\text{ms}, 0\,\text{ms})$ (blue). Note how the latter only weakly modulates the PSP height. To compensate the lose of synaptic efficacy due to small $U_0$, we rescale the weight matrix by dividing $f_w = 0.014$ .

(Fig. 4.11). For each method, we initialize the simulation with 10 different random seeds and collect $10^5$ samples from each simulation. To provide a frame of reference, we also plot two additional ISL curves. The POM (product of marginals) sampler generates images by sampling each pixel individually from its intensity distribution over the entire training set. This sampler preserves the marginal probability distributions for each pixel but discards any further structure of the image (encoded in correlations between pixel intensities). The OPT (optimal) sampler starts out with a base set of $10^5$ images generated with AST, from which it randomly picks images sequentially. This guarantees optimal mixing for the underlying model, because the base set covers all main modes of the state space, but consecutive samples have no correlation.

Due to its improved mixing capability, the LIF network is able to quickly cover a large portion of the relevant state space, as reflected by the on average faster ISL compared with Gibbs sampling (Fig. 4.11). This is a systematic effect and only weakly dependent on initial conditions, as can be seen in Fig. 4.11 right, which shows a histogram over $10^3$ random seeds. For this comparison, we chose a sampling duration of 10 s as a conservative estimate for the maximum duration for a biological agent to experience stable stimulus conditions and therefore sample from a stable target distribution. The faster mixing is the result of the spiking network's ability to jump out of local attractors, which is reflected in a much shorter time spend on average within the same mode (Fig. 4.12). Here, we define a mode as the dominant class of the currently represented image; a mode is therefore defined by the identity of the neuron in the label layer with the highest firing rate.

However, it is important to note that, due to the STP-modulated interaction, the spiking network does not sample from the exact same distribution as the Gibbs sampler, despite using an equivalent $(\boldsymbol{W}, \boldsymbol{b})$ parameter set. For a very large number of samples ($> 10^5$), the two methods converge towards the same ISL (Fig. 4.11), indicating that the discrepancy in performance for shorter sampling durations is not due to a fundamental difference in their respective ground truths.

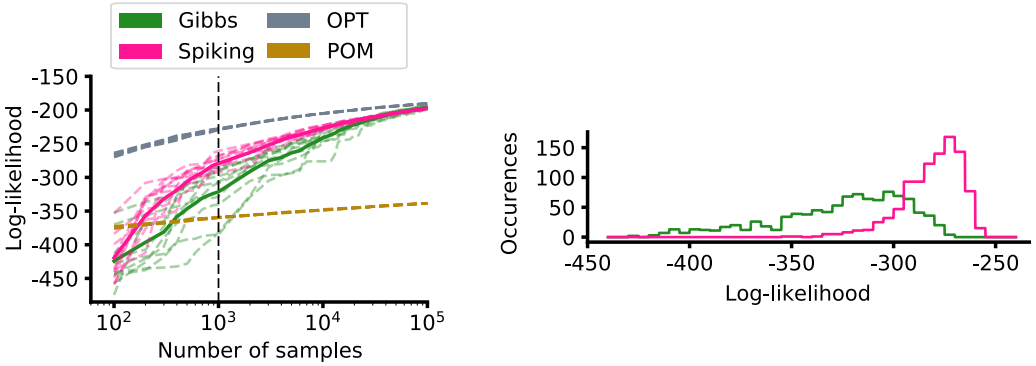


Figure 4.11.: **Left:** Log-likelihood from ISL of the test set calculated from an increasing number of samples. Each sampling method is simulated with 10 different random seeds (dashed lines) and their mean value is calculated (solid lines). The ISLs of an optimal sampler with the same parameters (OPT, gray) and the product of marginals (POM, brown) are shown for comparison. **Right:** Direct comparison between the two sampling methods for $10^3$ samples, equivalent to a sampling duration of 10 s in the biological domain. ISL histogram generated from $10^3$ random seeds. Figure is taken from *Leng et al.* (2018).

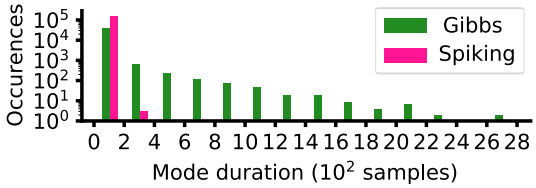


Figure 4.12.: Histogram of times spend within the same mode. For each method, the statistics are made on 10 simulations initialized with different random seeds with each collecting $10^5$ samples. The LIF network spend much shorter time on average within the same mode compared to Gibbs sampling.

While the ISL, as an abstract quantity, provides a useful numerical gauge of the quality of a generative model, a direct depiction of the produced images is particularly instructive. Here, we use the t-SNE method *Maaten and Hinton* (2008) (see section A.2.3) to project the generated images on a 2D plane. The similarity between samples is largely reflected by their distances on the plane and a large jump can be interpreted as a switch between attractors. As seen in Fig. 4.13, within the same sampling steps, the LIF network produces a significantly more diverse set of samples compared to the Gibbs sampler.

When the visible layer is clamped to a particular input, the same network can be

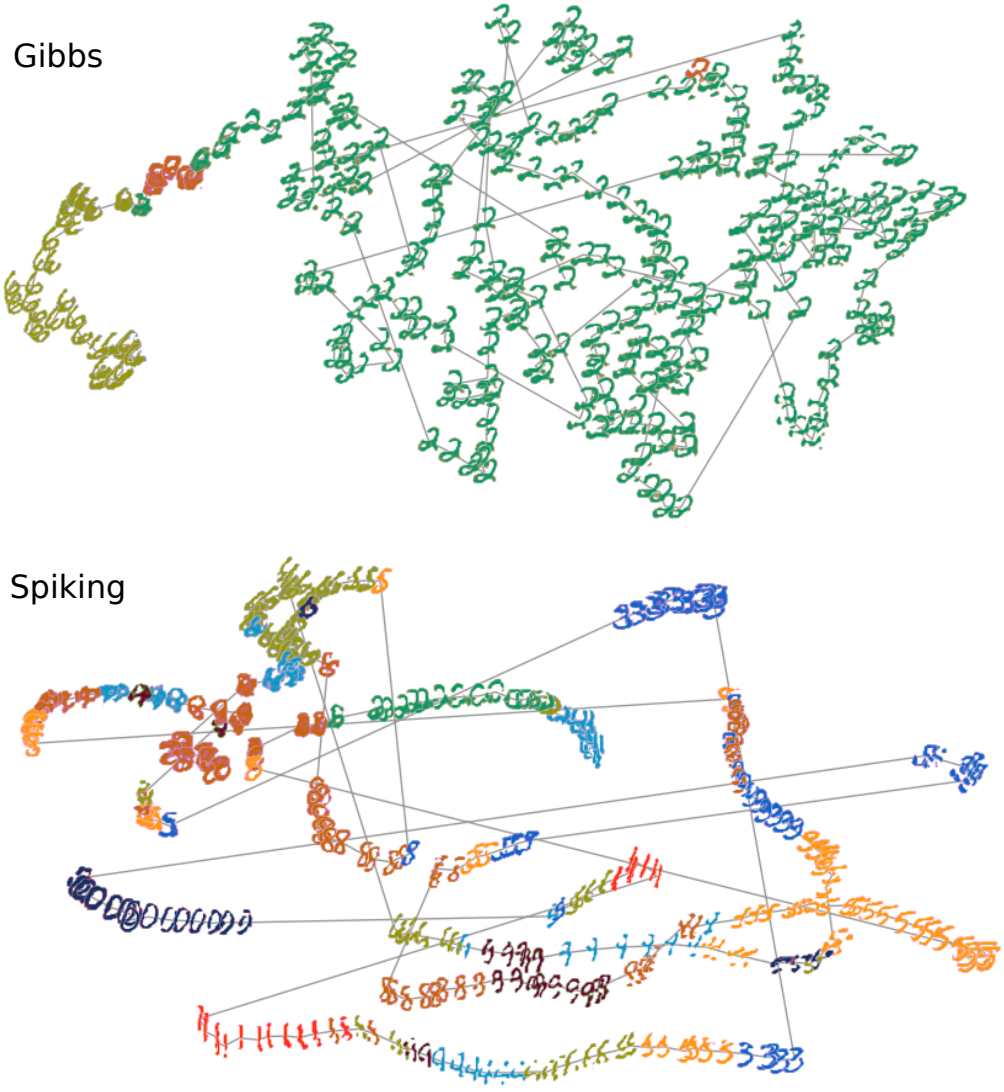


Figure 4.13.: Superior generative performance of an STP-endowed spiking network compared to an equivalent Gibbs sampler. t-SNE plots of images produced by the two methods over 1800 consecutive samples. For every 6th of these samples, an output image is shown. Consecutive images are connected by gray lines. Different colors represent different image classes, defined by the label unit that shows the highest activity at the time the sample is generated. Note that t-SNE inherently normalizes the area of the 2D projection; the volume of phase space covered by the Gibbs chain is, in fact, much smaller than the one covered by the spiking network. Figure is taken from *Leng et al.* (2018).

used as a discriminative model of the learned data. Using the same parameters as for the generative task, the benchmark Gibbs sampler obtained a classification accuracy of 93.4% on the MNIST test data. The LIF network with STP performed only slightly worse, at 93.2%. The additional generative capabilities gained by the spiking networks through STP were therefore not strongly detrimental to their classification accuracy. Better classification performances can be achieved by increasing the number of hidden units and direct training of the LIF network.

### 4.2.4. Modeling on imbalanced dataset

In the previous section, we studied the mixing performance of LIF-based RBMs with STP in benchmark dataset where the distribution of samples among different classes are roughly equal. In many real-world scenarios, the available data is imbalanced, with much of the data belonging to one class and significantly less samples being distributed over others. It is well-known that imbalanced data can cause severe problems for data mining and classification *Chawla* (2005); *García and Herrera* (2009). One solution is to create a more balanced dataset from the imbalanced one, which can be achieved by methods such as under- or over-sampling *García and Herrera* (2009); *Chawla et al.* (2002). However, such an a-priori modification of the input data does not seem biologically plausible. Still, cognitive biological agents appear to easily overcome this problem: humans will have little difficulty imagining a platypus from seeing only its bill, despite having likely seen many more ducks throughout their lifetime. In this section, we demonstrate in the following experiments that LIF networks with STP provide a simple solution to the problem of imbalanced training data, without any need for preprocessing.

We create an imbalanced dataset of 1000 images by randomly selecting 820 digits of class '1' and 45 from the '0', '2', '3' and '8' classes from the MNIST dataset. After the training of the model (see section A.2.2 for training details), we compare the generative output of a Gibbs sampler, an AST sampler and the LIF-based RBM with STP. Note that the effective sampling speed of AST is roughly $N_{\text{temp}}$ times slower compared to Gibbs sampling where $N_{\text{temp}}$ is the number of temperatures[4]. For the LIF network, we use three STP parameter sets of $(U_0 = 0.07, \tau_{\text{rec}} = 60\,\text{ms}, \tau_{\text{fac}} = 0\,\text{ms}, f_w = 0.085, )$, $(U_0 = 0.01, \tau_{\text{rec}} = 280\,\text{ms}, \tau_{\text{fac}} = 0\,\text{ms}, f_w = 0.014)$ and the renewing synapse of $(U_0 = 1.0, \tau_{\text{rec}} = 10\,\text{ms}, \tau_{\text{fac}} = 0\,\text{ms})$. Their envelopes of consecutive PSPs are shown in Fig. 4.14. These parameter sets are largely empirical, and we believe that there exist a wide range of parameter sets which will produce similar result.

We collect 16000 consecutive samples from each sampling method and plot their mode distribution and evolution traces (Fig. 4.15 top). The result shows that samples from Gibbs sampling and AST mainly stay in mode '1' which is the dominant class. In contrast, the LIF network presents diverse sampling dynamics with different STP parameters. With $\tau_{\text{rec}} = 280\,\text{ms}$ it maintains a close approximation of the data distribution, but with faster mode switching compared to Gibbs sampling as shown in the trace evolution. With $\tau_{\text{rec}} = 60\,\text{ms}$ the PSP envelope achieves a faster and higher potentiation followed by a more rapid decay, which further improves the mixing of the network and enables it to generate much more uniformly distributed samples, without focusing on the majority mode. The STP-induced weakening of active attractors balances out their activity and facilitates the network to switch between different modes. With renewing synapse of $\tau_{\text{rec}} = 10\,\text{ms}$ the network seems to be trapped in some minority modes, this could be largely due to the latency effect induced from an exponential shape of PSP which will be better illustrated and discussed in the next chapter. With different

[4]Here we use 20 temperatures, more details see section A.2.2.

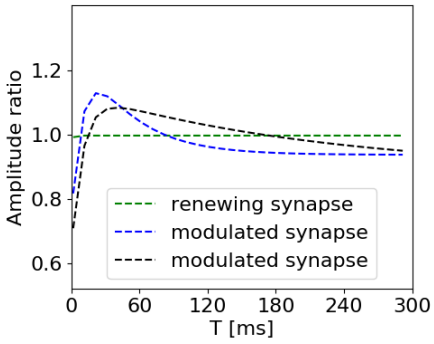


Figure 4.14.: Envelopes of consecutive PSPs of renewing synapse (green) and STP parameter sets of ($U_0 = 0.07, \tau_{\text{rec}} = 60\,\text{ms}, \tau_{\text{fac}} = 0\,\text{ms}$) with $f_w$ of 0.085 (blue), and ($U_0 = 0.01, \tau_{\text{rec}} = 280\,\text{ms}, \tau_{\text{fac}} = 0\,\text{ms}$) with $f_w$ of 0.014 (black). Notice that the duration of the potentiation plateau of $\tau_{\text{rec}} = 60\,\text{ms}$ is much shorter than $\tau_{\text{rec}} = 280\,\text{ms}$ which is also used in the previous section.

configuration of STP parameters, the LIF network exhibits diverse sampling statistics, demonstrating potentials in multi-functional tasks.

A t-SNE plot further shows consecutive images generated by the LIF network with $\tau_{\text{rec}} = 60\,\text{ms}$ during the sampling process (Fig. 4.15 bottom).

In another scenario, the ability of STP-endowed LIF network to escape active attractors become particularly useful for inference based on incomplete information, which we demonstrate with a pattern completion example. Here, we create a training set of 5000 images with 6 majority classes ('0', '1', '2', '3', '4', '6', 800 samples each) and one minority class (200 samples of '5') from the MNIST dataset. We use an RBM with the same size as the first experiment. After training, we generated an ambiguous image by clamping the lower half of the visible layer to a configuration compatible with both a '3' and a '5' (Fig. 4.16). We clamp the ambiguous image to the network by multiplying the biases of corresponding neurons with a factor of 5, and clamp off the rest background neurons in the lower half by setting their biases to -50. The top half neurons of the visible layer are left for free to complete the pattern.

We generate a sequence of consecutive images with different samplers (Fig. 4.17 top left) and plot their mode distributions (Fig. 4.17 top right). The result shows Gibbs and AST strongly undersample the minority class '5', with a '3' to '5' ratio of 95.6 and 90.0. In contrast, the LIF network produces a much more balanced set of images, with swift transitions between modes (Fig. 4.17 middle, bottom). The ratio between '3' and '5' is 3.7, which closely reflects their ratio in the dataset. With an appropriate choice of parameters, STP enables the LIF network to become a better sampler with fast mixing. The estimate of the possible realities underlying the incomplete observation is therefore improved both on long and on short time scales. This can be particularly

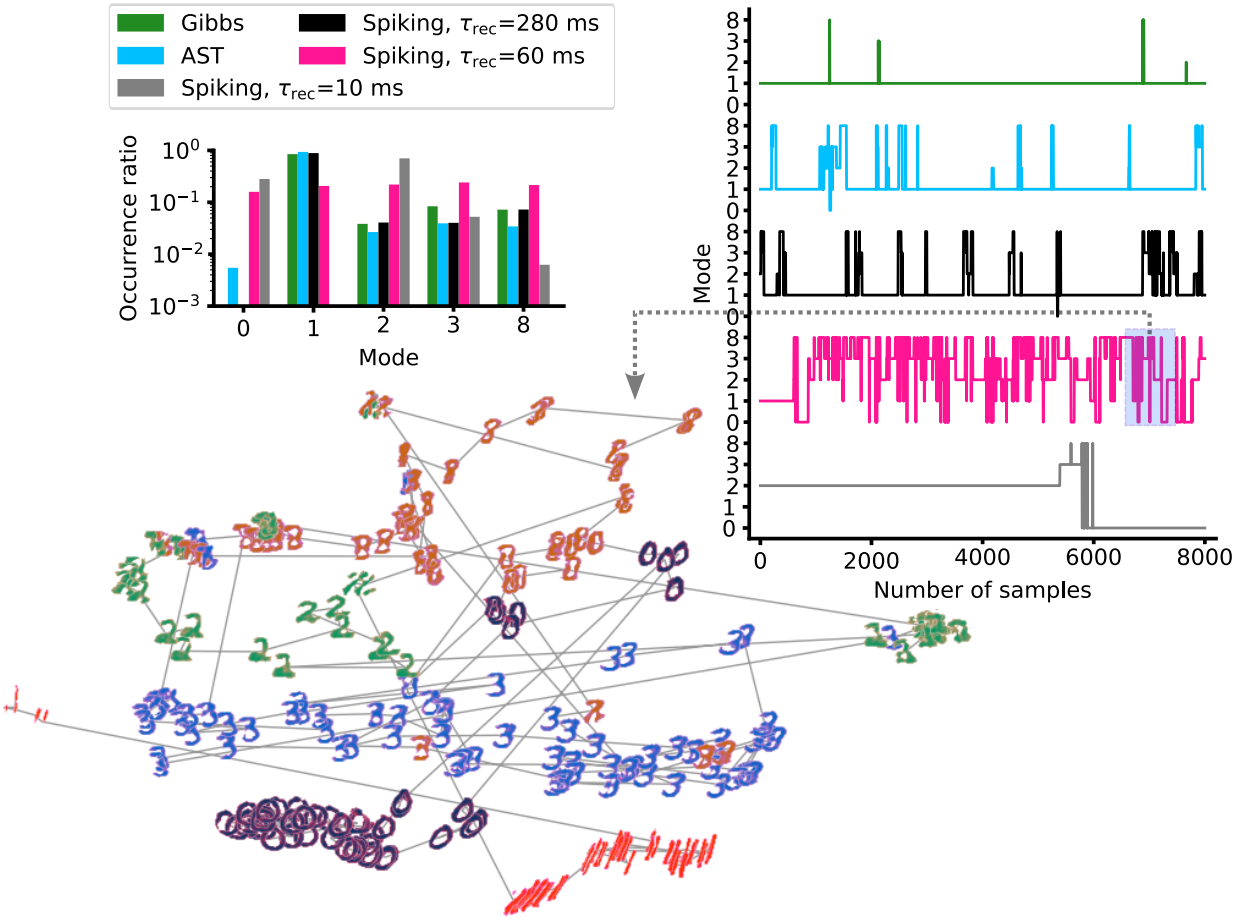


Figure 4.15.: Comparison of Gibbs and AST samplers with STP-endowed LIF networks for imbalanced training data. **Top left:** Histogram of relative time spent in different modes calculated from 16,000 samples. The Gibbs and AST sampler mainly stay in the dominant class '1'. In contrast, the LIF network shows diverse sampling statistics with different STP parameters. With $\tau_{\text{rec}} = 280\,\text{ms}$ it mixes faster then Gibbs sampling meanwhile closely approximates the data distribution. With $\tau_{\text{rec}} = 60\,\text{ms}$ the PSP envelope varies more rapidly with a higher potentiation level, which further improves the mixing of the network and enables it to generate much more uniformly distributed samples. With renewing synapse of $\tau_{\text{rec}} = 10\,\text{ms}$ the network seems to be trapped in some minority modes, this could be largely due to the latency effect induced from an exponential shape of PSP. **Top right:** Mode evolution over 8,000 consecutive samples. Gibbs and AST sampler basically trapped in the majority mode. Modulated synapse improves the mixing of the LIF network: with $\tau_{\text{rec}} = 280\,\text{ms}$ the network achieves faster mixing compared to the Gibbs sampler. With $\tau_{\text{rec}} = 60\,\text{ms}$ a more rapid variation of the PSP envelope further improves its mixing enabling it to produce much more balanced samples. **Bottom:** t-SNE plot of 250 consecutive images generated by the LIF network over a duration of 10 s, with 40 ms between consecutive images.


Figure 4.16.: Ambiguous input to the visible layer. The upper half is not clamped and free to complete the pattern.

useful for an agent in need of a quick reaction, as, for example, often required in nature in a fight-or-flight scenario.

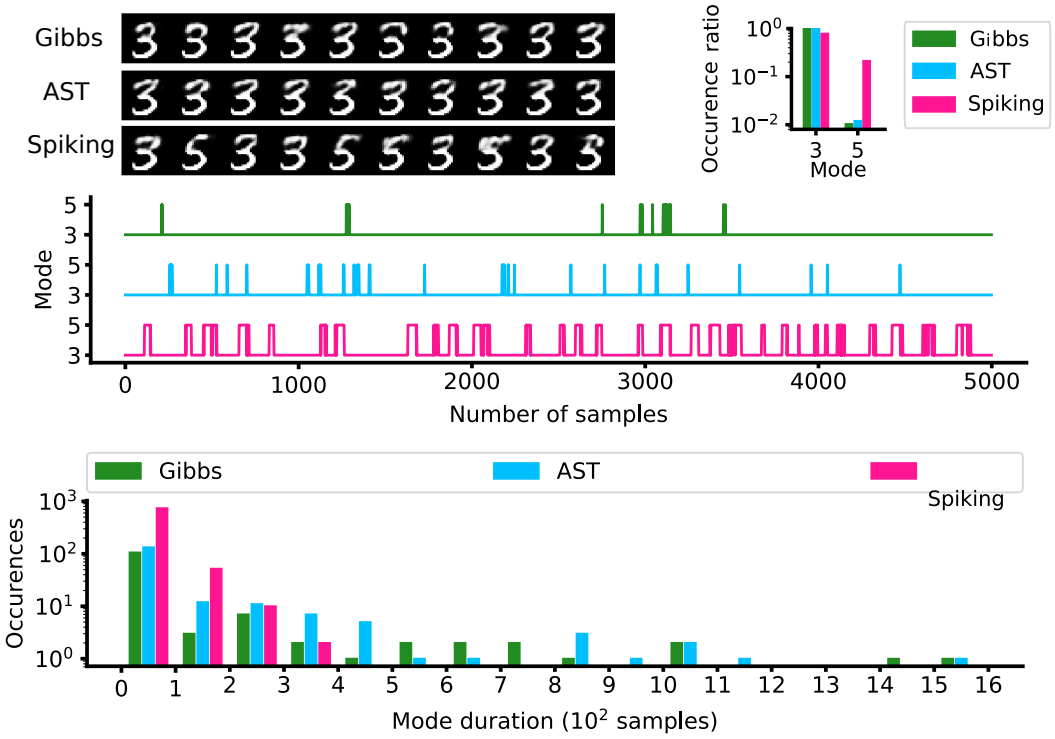


Figure 4.17.: Comparison of different samplers for pattern completion after learning imbalanced dataset. Here, we use an STP parameter set of ($U_0 = 0.01, \tau_{\text{rec}} = 280\,\text{ms}, \tau_{\text{fac}} = 0\,\text{ms}$) with $f_w$ of 0.014, the same as the previous section. **Top left:** Comparison of sequences of images generated by different methods over 5000 samples (only every 500th is shown). **Top right:** Histogram of relative time spent in different modes during the pattern completion task, measured over 20,000 consecutive samples. **Middle:** Mode evolution over 5,000 consecutive samples. The Gibbs and AST sampler spend most of the time in class '3' and seldom go to class '5'. In contrast, the STP-endowed LIF network is able to switch to class '5' much more frequently. **Bottom:** Histogram of times spend within the same mode over 20,000 samples. The LIF network spend much shorter time on average within the same mode compared to the AST and Gibbs sampler.

## 4.3. Modulation of STP on probability distributions

In the previous section, we have demonstrated the ability of STP for improving mixing in generative tasks ranging from low to high dimension. The network performances are measured by their generated images and criterion established upon them. However, the generated images reflect the variation of visible states which is essentially a phenomenon caused by STP-endowed sampling, it does not explain why or how mode variations happen. Throughout this section, we study the modulation of STP on probabilities of network states, which provides more details and an intuitive understanding of the STP-endowed sampling dynamics.

In the previous section we designed a bar experiment to compare the mixing performances between LIF-sampling with STP and the traditional Gibbs sampling. We continuously use this example and further illustrate the variation of marginal probabilities of network states during the sampling process. Subsequently, we reduce the network dimension and reveal the local variation of attractors through strict calculation of the conditional probability distribution of network states.

### 4.3.1. Modulation on marginal probability distributions

As discussed in section 2.1.1, the probability for a visible state to occur in a BM is given by the marginal distribution

$$p(\mathbf{v}) = \frac{p^*(\mathbf{v})}{Z} = \frac{1}{Z} \sum_{\mathbf{h}} \exp\left[-E(\mathbf{v}, \mathbf{h}; \mathbf{W}, \mathbf{b})\right] \ . \tag{4.8}$$

For an RBM in our case, the computation of $p(\mathbf{v})$ can be further simplified by making use of the connection restriction between the visible and hidden layer, which can be derived

as following

$$
\begin{aligned}
p(\mathbf{v}) &= \frac{1}{Z}\sum_{\mathbf{h}} \exp\left[-E(\mathbf{v},\mathbf{h})\right] \\
&= \frac{1}{Z}\sum_{\mathbf{h}} \exp\left(\sum_{i=1}^{N} a_i v_i + \sum_{j=1}^{M} b_j h_j + \sum_{i,j} W_{ij} v_i h_j\right) \\
&= \frac{1}{Z}\exp\left(\sum_{i=1}^{N} a_i v_i\right)\sum_{\mathbf{h}} \exp\left[\sum_{j=1}^{M} h_j\Big(b_j + \sum_{i=1}^{N} W_{ij} v_i\Big)\right] \\
&= \frac{1}{Z}\exp\left(\sum_{i=1}^{N} a_i v_i\right)\sum_{h_1}\cdots\sum_{h_M}\prod_{j=1}^{M} \exp\left[h_j\Big(b_j + \sum_{i=1}^{N} W_{ij} v_i\Big)\right] \\
&= \frac{1}{Z}\exp\left(\sum_{i=1}^{N} a_i v_i\right)\sum_{h_1} \exp\left[h_1\Big(b_1 + \sum_{i=1}^{N} W_{i1} v_i\Big)\right]\cdots\sum_{h_M} \exp\left[h_M\Big(b_M + \sum_{i=1}^{N} W_{iM} v_i\Big)\right] \\
&= \frac{1}{Z}\exp\left(\sum_{i=1}^{N} a_i v_i\right)\prod_{j=1}^{M}\left\{\sum_{h_j\in\{0,1\}} \exp\left[h_j\Big(b_j + \sum_{i=1}^{N} W_{ij} v_i\Big)\right]\right\} \\
&= \frac{1}{Z}\exp\left(\sum_{i=1}^{N} a_i v_i\right)\prod_{j=1}^{M}\left[1 + \exp\Big(b_j + \sum_{i=1}^{N} W_{ij} v_i\Big)\right].
\end{aligned}
\tag{4.9}
$$

This avoids the ergodic calculation of all $\mathbf{h}$ states as required by Eq. 4.8. We used a similar approach in *Martel* (2015). Considering the bipartite architecture of an RBM, the marginal probability of a hidden state can also be expressed as

$$
p(\mathbf{h}) = \frac{1}{Z}\exp\left(\sum_{i=1}^{M} b_j h_j\right)\prod_{i=1}^{N}\left[1 + \exp\Big(a_i + \sum_{j=1}^{M} W_{ij} h_j\Big)\right] \tag{4.10}
$$

For traditional Gibbs sampling or LIF-sampling with renewing synapse, the marginal probability of a certain visible or hidden state is constant since the synaptic connections are (or on average) constant during the sampling process. For modulated synapse, $p(\mathbf{v})$ or $p(\mathbf{h})$ becomes a variable due to the temporary change of synaptic efficacy caused by STP. Since STP modifies the weights $\mathbf{W}$ of the network according to the activity of the presynaptic neuron. After simulation, we can compute the STP variables $R$ and $U$ at each spike based on the spiking history of each neuron (see Eq. 4.4 and 4.5). The marginal state probability during STP-endowed sampling process can then be calculated by multiplying these variables to the weight matrix.

Specifically, for the calculation of $p(\mathbf{v})$ and $p(\mathbf{h})$ at sampling step $n$, we multiply weight $W_{ij}$ by $(U_i^n R_i^n + U_j^n R_j^n)/2$. For neurons which are silent (not in refractory state) at the measured sampling step, their corresponding $U$ and $R$ are set to 1. Since the computation cost for the partition function increases exponentially with the number of neurons which makes it impractical for large size networks, we temporally denote the new partition

function at sampling step $n$ as $Z^n$. The unnormalized probabilities $p^*(\mathbf{v})$ and $p^*(\mathbf{h})$ at sampling step $n$, therefore, are expressed as:

$$p_n^*(\mathbf{v}) = \exp\left(\sum_{i=1}^{N} a_i v_i\right) \prod_{j=1}^{M} \left[1 + \exp\left(b_j + \sum_{i=1}^{N} \frac{U_i^n R_i^n + U_j^n R_j^n}{2} W_{ij} v_i\right)\right], \tag{4.11}$$

$$p_n^*(\mathbf{h}) = \exp\left(\sum_{i=1}^{M} b_j h_j\right) \prod_{i=1}^{N} \left[1 + \exp\left(a_i + \sum_{j=1}^{M} \frac{U_i^n R_i^n + U_j^n R_j^n}{2} W_{ij} h_j\right)\right]. \tag{4.12}$$

Now, we apply this approach to the bar experiment presented in section 4.2.2 to inspect probability variations of certain network states. Due to the size of the network, it is impractical to calculate for all possible visible or hidden states. Instead, we collect all emerged visible and hidden states in simulation, which are in total 4944 visible and 3009 hidden states for a simulation of 5000 sampling steps (50,000 ms). To obtain a more panoramic visualization, we project these high dimensional vectors into a 2 dimensional plane (Fig. 4.18, 4.19 bottom) using the t-SNE method *Maaten and Hinton* (2008) (see section A.2.3) as before . The similarity between vectors is reflected by their 2D distances and the vectors roughly form three clusters, each corresponding to a mode of the bar image.

We calculate the ratio change of $p^*(\mathbf{v})$ and $p^*(\mathbf{h})$ relative to their original value (all $U$ and $R$ equal to 1) during three mode switches observed from the generated images (Fig. 4.18, 4.19 top), i.e. 200-400 ms, 1000-1200 ms and 1700-1900 ms. Specifically, these ratios represent the modulation level of the probability of each state multiplied by a factor of $\frac{Z}{Z^n}$. Since we don't know the value of $Z^n$, the ratio reflects more of the modulation on the state's probability relative to other states at the same sampling step. An alternative option is to calculate the Boltzmann factors between states.

The results show that for both visible and hidden states, when the network generate a certain bar image, a large number of marginal probabilities in the corresponding cluster are weakened accordingly (since similar states tend to encode similar patterns), while states in other clusters either maintain or increase their values, intuitively demonstrating the local modulation effect of STP. The relative activity changes of the cluster are also in pace with the variations of image in the visible layer. For the visible case, one can observe that the intersected region of clusters always has a relatively high ratio, indicating a mixing principle which facilitates the transition states. For the hidden case, states in the same cluster change their ratio less uniformly compare to the visible case, indicating a more sparse encoding in the hidden layer.

In addition, we randomly choose 6 networks states (separated into visible and hidden states, indicated by the black dots in Fig. 4.18 and 4.19) when the network presents certain bars (two states for each bar) and plot the temporal evolution of the ratio of their unnormalized marginal probabilities. The results are shown in Fig. 4.20 and 4.21. It can be observed that when the network produces a certain bar image, the marginal

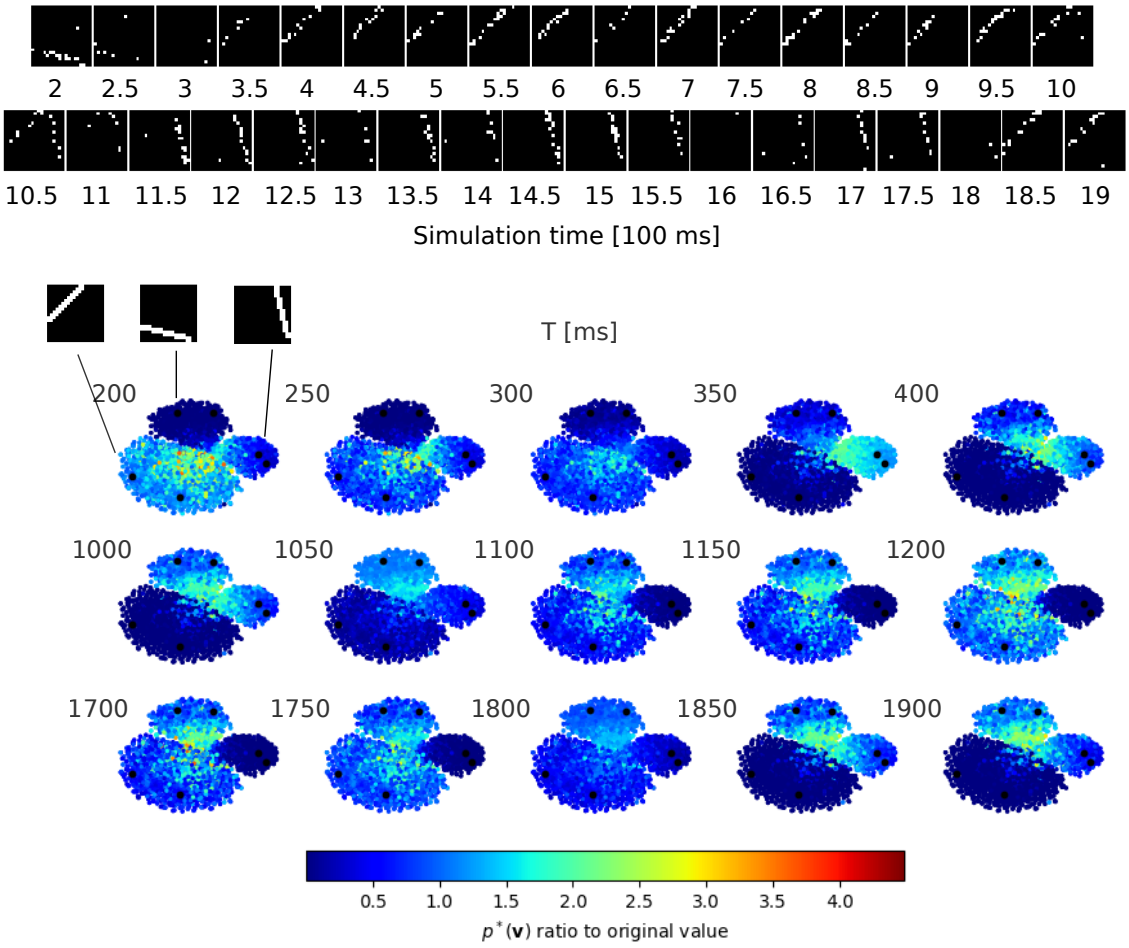


Figure 4.18.: **Top:** Image sequence generated from simulation, along with simulation times the image samples are obtained. Note that here we directly use the binary activity of the visible layer, different from Fig. 4.7. **Bottom:** Ratio change of $p^*(\mathbf{v})$ during three mode switches observed from the generated images, i.e. 200-400 ms (fist row), 1000-1200 ms (second row) and 1700-1900 ms (third row). The times at which we measure the probabilities are indicated on the upper left of each plot. The black dots refers to the selected visible states plotted in Fig. 4.20. One can observe that the intersected region of clusters high ratio, indicating the mixing facilitation principle. The inhomogeneous ratio change of $p^*(\mathbf{v})$ intuitively demonstrates the local modulation effect of STP. More details during the mode switch are revealed. For example, STP not only decreases probabilities of local states to encourage mixing, but also simultaneously increases probabilities of potential states in other clusters, as observed in 350 - 400 ms, 1100 -1150 ms and 1700 - 1750 ms. Notice that to increase image contrast we set all ratios above 4.5 (approx. 0.1%) to be 4.5.

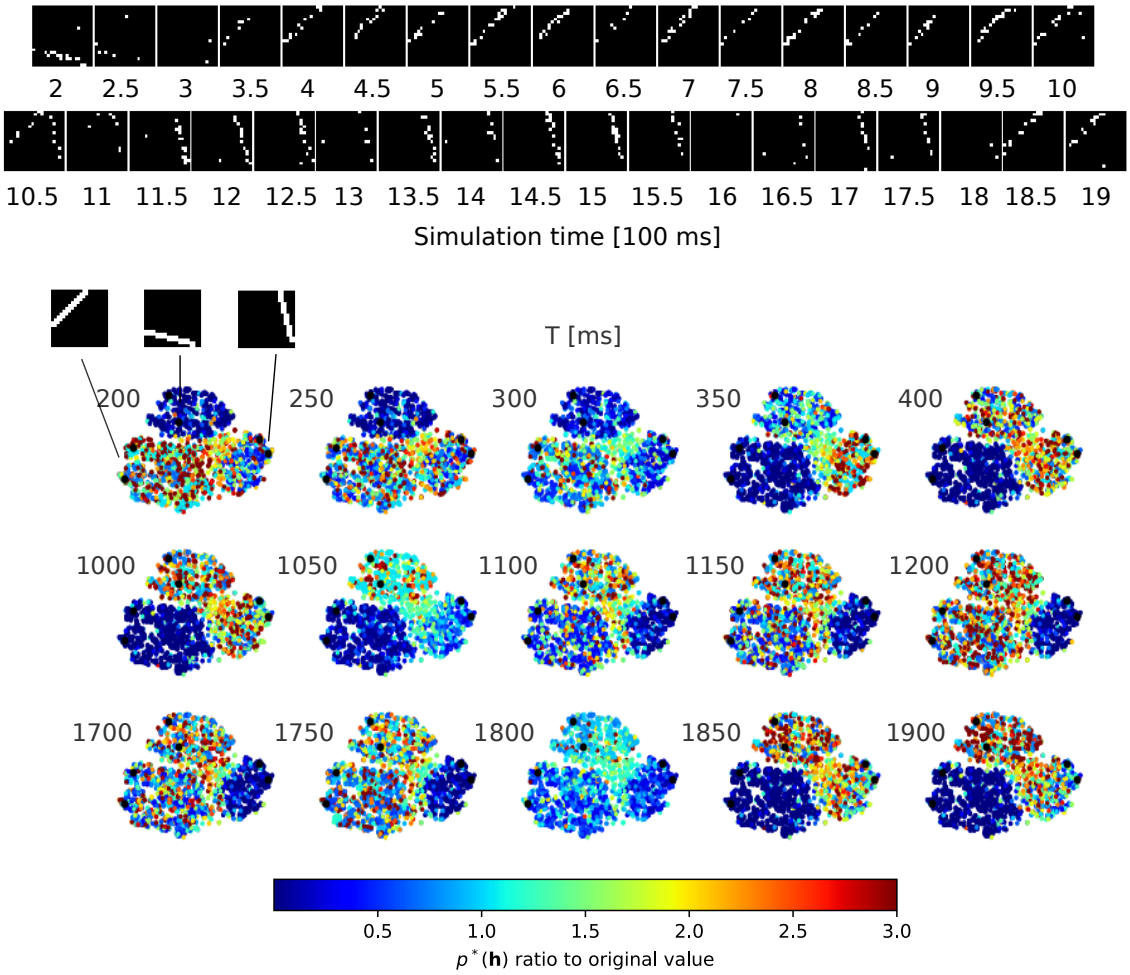


Figure 4.19.: **Top:** Image sequence generated from simulation, along with simulation times the image samples are obtained. Note that here we directly use the binary activity of the visible layer, different from Fig. 4.7. **Bottom:** Ratio change of an ensemble of $p^*(\mathbf{h})$ during three mode switches observed from the generated images, i.e. 200-400 ms (fist row), 1000-1200 ms (second row) and 1700-1900 ms (third row). The black dots corresponds to the hidden states plotted in Fig. 4.21. Similar local modulation effect of STP occurs as in the case of marginal visible probabilities. Notice that for states in the same cluster their ratio change less uniformly compare to the former visible case, the reason could be that hidden states are more sparse than visible states in terms of encoding. Notice that to increase image contrast we set all ratios above 3 (approx. 6%) to be 3.

probabilities encoding the corresponding bar image are significantly weakened due to STD while those encoding other bars are fluctuating around values of the same order of the original value, therefore only the strength of the local active attractor is intensely depressed. This mechanism facilitates the network to jump out of local minima and the weakened marginal probabilities recover when the network switch to other modes. Finally, we also calculate the average ratio variation of $p^*(\mathbf{v})$ and $p^*(\mathbf{h})$ in terms of different modes. The mode allocation of a network state $(\mathbf{v}, \mathbf{h})$ is determined by the minimum Euclidean distance between the firing probability of the visible state $p(\mathbf{v})$ and pixel values of three bar images. The results are plotted in Fig. 4.22 and 4.23 which show similar variation pattern.

### 4.3.2. Modulation on conditional probability distributions

To avoid the influence of potential variations in the partition function induced by STP, an alternative option is to calculate the conditional probability $p(\mathbf{v}|\mathbf{h})$ and $p(\mathbf{h}|\mathbf{v})$, which represent the transition probability from one state to another during the sampling process.

The conditional probability during the STP-endowed sampling at sampling step $n$ is calculated as:

$$p_n(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{N} p_n(v_i|\mathbf{h})\,, \tag{4.13}$$

$$p_n(v_i|\mathbf{h}) = \frac{1}{1+\exp(-\sum_{j=1}^{J} R_j^n U_j^n W_{ji} h_j - b_i)}\,. \tag{4.14}$$

$$p_n(\mathbf{h}|\mathbf{v}) = \prod_{j=1}^{M} p_n(h_i|\mathbf{v})\,, \tag{4.15}$$

$$p_n(h_j|\mathbf{v}) = \frac{1}{1+\exp(-\sum_{i=1}^{I} R_i^n U_i^n W_{ji} v_i - b_j)}\,. \tag{4.16}$$

Notice that here we assume a synchronized synaptic transmission from one layer to the other, which is an approximation since spike transmission in the LIF network is asynchronous, This problem no longer exist when we implement STP mechanism in traditional RBMs in the next chapter.

To perform an ergodic calculation of all possible states, we reduce the network size to 12 visible and 12 hidden units and created a training set of 3 3 $\times$4 horizontal bar images (Fig. 4.24). After training (see section A.2.2 for training details), we run the LIF network with STP ($U_0 = 1, \tau_{\text{rec}} = 13\,\text{ms}, \tau_{\text{fac}} = 0\,\text{ms}$) and generate a sequence of images (Fig. 4.25 top). At each sampling step $n$, we calculate for all visible (or hidden) states their conditional probabilities given the current state, and sum up for each mode

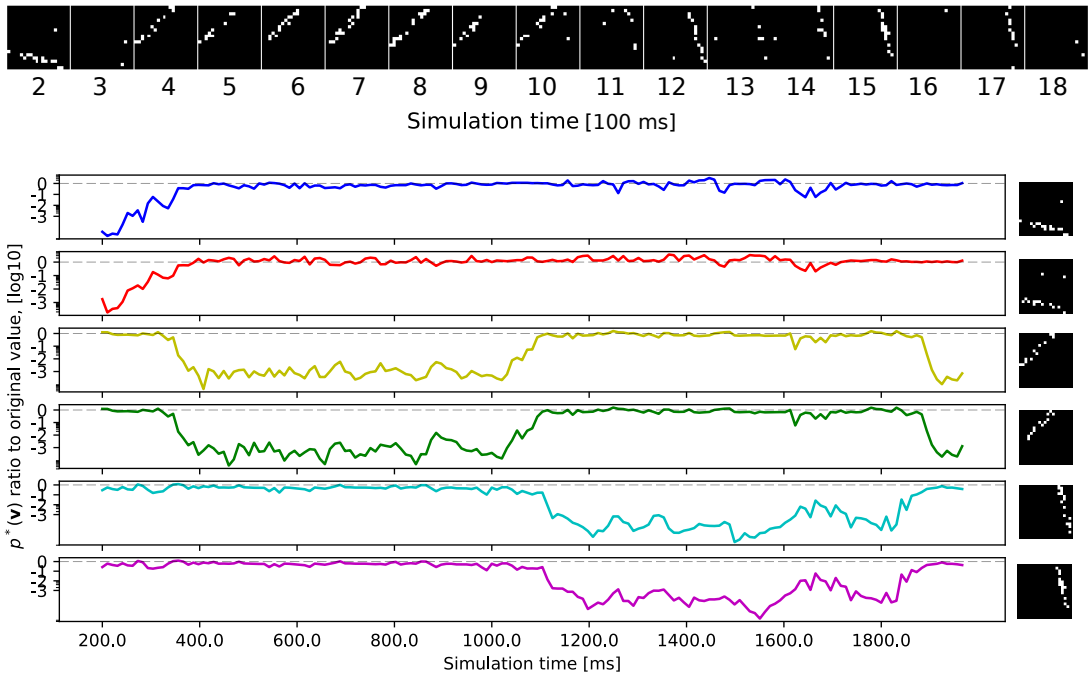


Figure 4.20.: Ratio change of $p^*(\mathbf{v})$ for selected states during simulation. **Top:** Images (binary states) produced by the network on the visible layer. Images are sampled with an interval of 100 ms, starting from 200 ms to 1800 ms. The time span corresponds to the bottom plot. **Bottom:** $p^*(\mathbf{v})$ evolutions of 6 randomly selected visible states, with two states for each bar. Their corresponding visible images are plotted in the end. The network first generates bottom bars, resulting a strong local weakening of $p^*(\mathbf{v})$ for $\mathbf{v}$ encoding the corresponding bar image (blue and red lines). This encourages the network to jump out of the local mode and the once weakened $p^*(\mathbf{v})$ are recovered after the network switches to another bar, which again leads to a local weakening of $p^*(\mathbf{v})$ for $\mathbf{v}$ encoding that bar (yellow and green lines), and this pattern repeats. Notice that roughly between 1600 and 1700 ms, there is a decrease in the weakening of the local attractor, indicating that attractors are competing with each other, resulting that no clear bar images are produced.

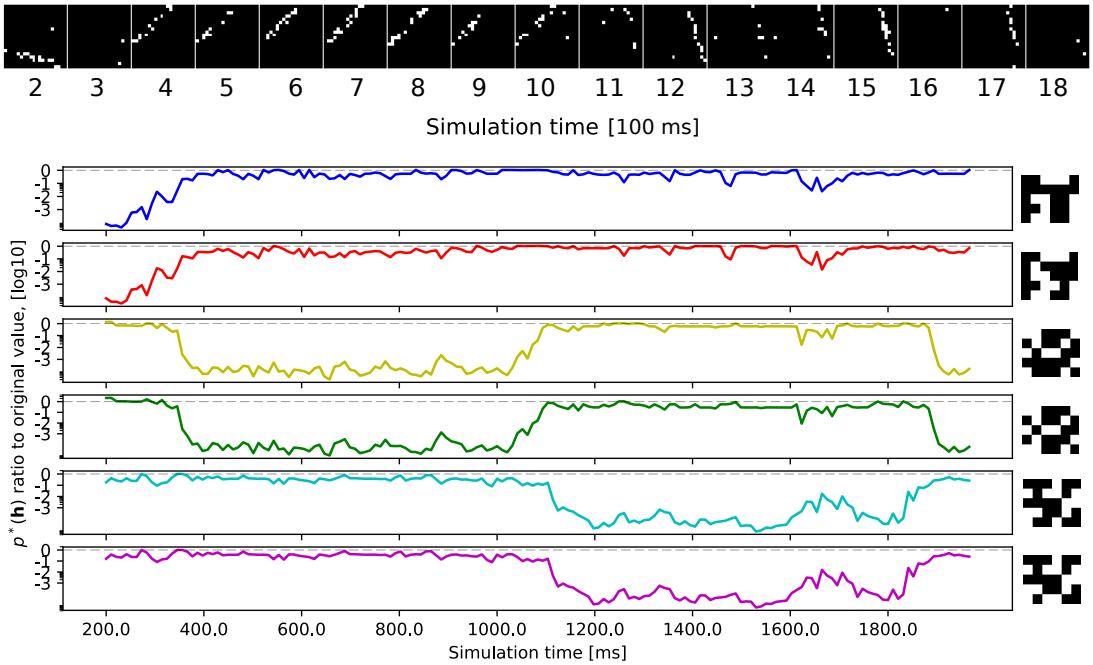


Figure 4.21.: Ratio change of $p^*(\mathbf{h})$ for selected states during simulation. **Top:** Images (binary states) produced by the network on the visible layer. Images are sampled with an interval of 100 ms, starting from 200 ms to 1800 ms. The time span corresponds to the bottom plot. **Bottom:** $p^*(\mathbf{h})$ evolutions of 6 randomly selected hidden states, with two states for each mode. The corresponding hidden states are plotted in the end. Similar phenomena can be observed as in the visible state case.

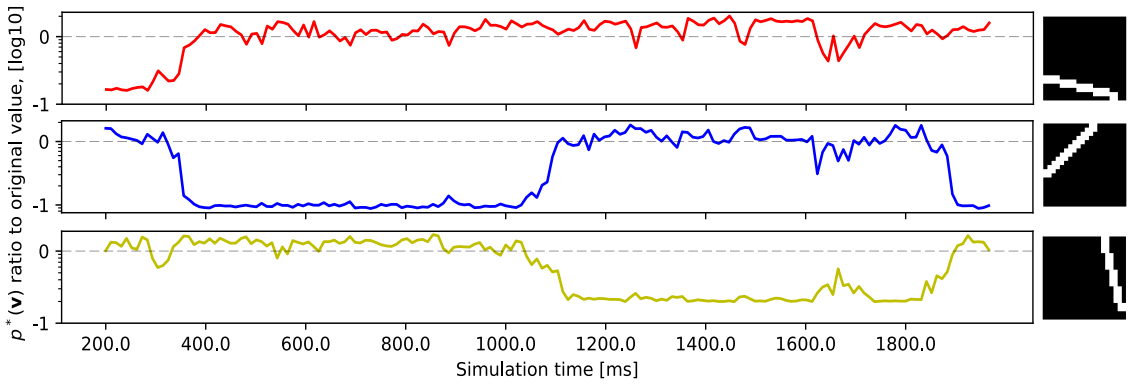


Figure 4.22.: Variations of local active attractors reflected on the ratio change of mean $p^*(\mathbf{v})$ for each mode during simulation. Their corresponding modes are plotted in the end.

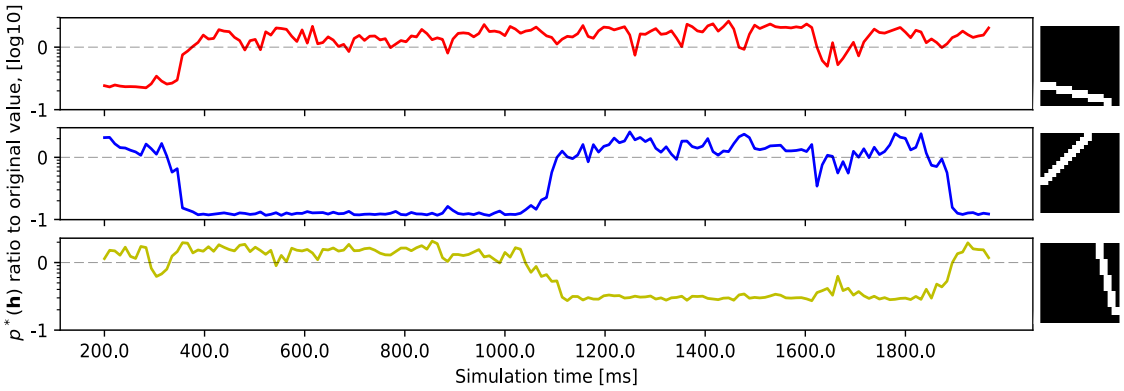


Figure 4.23.: Variations of local active attractors reflected on the ratio change of mean $p^*(\mathbf{h})$ for each mode during simulation. Their corresponding modes are plotted in the end.

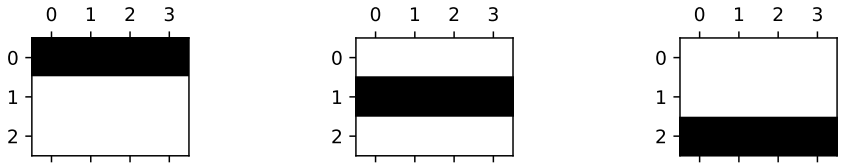


Figure 4.24.: Training set of 3 bar images. Each image has the size of 3 × 4 pixels. The pixel value (for training) of black pixels is 1 and for white pixels is 0.

according to their allocation:

$$p_n^{\mathbf{v}_k\mathbf{h}} = \sum_{\mathbf{v}\in\text{mode k}} p_n(\mathbf{v}|\mathbf{h}_n) \tag{4.17}$$

$$p_n^{\mathbf{h}_k\mathbf{v}} = \sum_{\mathbf{h}\in\text{mode k}} p_n(\mathbf{h}|\mathbf{v}_n) \tag{4.18}$$

The result (Fig. 4.25 middle) shows that both $p^{\mathbf{v}_k\mathbf{h}}$ and $p^{\mathbf{h}_k\mathbf{v}}$ are dominant if the network in in mode $k$, which is as expected. This phenomenon should also occur for plain Gibbs sampling.

In addition, for all visible (or hidden) states, we calculate its averaged conditional probability over all potential hidden (or visible) states,and sum them up for each mode, obtaining:

$$p_n^{\mathbf{v}_k\bar{\mathbf{h}}} = \sum_{\mathbf{v}\in\text{mode k}} E_{\mathbf{h}} \langle\, p_n(\mathbf{v}|\mathbf{h}) \,\rangle \tag{4.19}$$

$$p_n^{\mathbf{h}_k\bar{\mathbf{v}}} = \sum_{\mathbf{h}\in\text{mode k}} E_{\mathbf{v}} \langle\, p_n(\mathbf{h}|\mathbf{v}) \,\rangle \tag{4.20}$$

The result (Fig. 4.25 bottom) shows that during the sampling process, both $p^{\mathbf{v}_k\bar{\mathbf{h}}}$ and $p^{\mathbf{h}_k\bar{\mathbf{v}}}$ are depressed if the network is in mode $k$, demonstrating the local modulation effect of STP. Note that for Gibbs sampling these two quantities will be constant since the model parameters are unchanged.

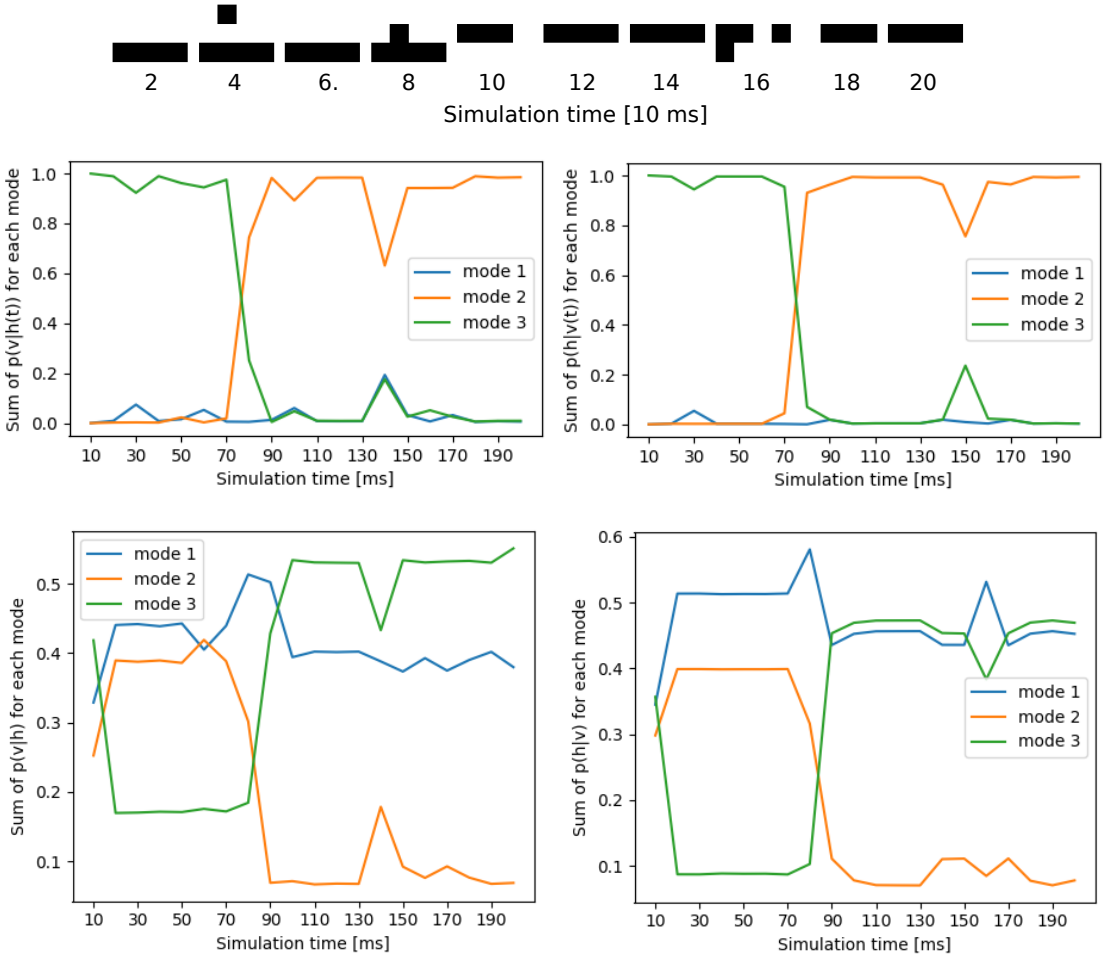


Figure 4.25.: **Top:** Images (binary states) produced by the network on the visible layer. Images are sampled with an interval of 20 ms, starting from 20 ms to 200 ms. The time span corresponds to the middle and bottom plots. **Middle:** Temporal evolution of $p^{\mathbf{v}_k\mathbf{h}}$ (left) and $p^{\mathbf{h}_k\mathbf{v}}$ (right). **Bottom:** Temporal evolution of $p^{\mathbf{v}_k\bar{\mathbf{h}}}$ (left) and $p^{\mathbf{h}_k\bar{\mathbf{v}}}$ (right).

## 4.4. Discussion

In this chapter, we have shown how a combination of spike-based communication and short-term plasticity can enhance the ability of neural networks to perform probabilistic inference in high-dimensional data spaces. Specifically, we discussed the functionality of different types of PSP envelopes and developed a modulated synapse based on the Tsodyks-Markram model of STP for generation tasks where mixing are hard. Here, we demonstrated through both simulations and theoretical calculations that the spike-triggered plasticity rule modifies active local attractors, in contrast to simulated tempering methods used for classical neural networks which require complex computations on the global network state and long waiting times between valid samples. The spiking networks outperformed their classical counterparts as generative models of benchmark dataset, with little disturbance to their classification capability, which we expect to be largely remediable by additional fine-tuning of the network parameters. Furthermore, they were also able to cope with imbalanced training data, as demonstrated by their superior performance in the balance sampling task and the pattern completion task on ambiguous input. Intriguingly, the synaptic parameters used to achieve this performance are compatible to experimental data (*Wang et al.*, 2006). We thereby offer a potential explanation for the generative capabilities of cortical networks, while at the same time proposing a simple but efficient mechanism to bolster the usefulness of spiking networks for machine learning applications.

Depending on the nature of the task and the associated optimal parameters, STP can play multiple roles. For low-dimensional spaces in which networks only rarely have mixing problems, STP can narrow the gap between the sampled and the target distribution, as demonstrated in section 4.2.1. With an appropriate deviation from the benchmark renewing synapse, STP improves mixing in multimodal distributions while closely approximates the target data distribution (section 4.2.3, 4.2.4). When the data distribution is highly imbalanced, by further increasing its potentiation level, STP is able to generate much balanced samples presenting a functionally advantageous distortion of the network's underlying distribution. In search of an optimal range of parameters, a theoretical study of how variations of PSP envelope leads to different sampling statistics on certain probability distributions is needed.

The STP parameters themselves require only little tuning, as evidenced by the comparatively large volume in parameter space that enhances performance, especially for high-dimensional problems. However, the optimal parameter set may vary, depending on the goal of the task. In a machine learning context, various algorithms for meta-parameter optimization have been proposed and could be applied to STP as well (*Reif et al.*, 2012; *Thornton et al.*, 2013; *Shahriari et al.*, 2016). With respect to biology, as the function and location of individual brain areas remain largely conserved both within and among species, we speculate evolution to have played a key role in parameter optimization.

In fact, it was suggested that during a working memory task studied in vivo with

rats, short-term synaptic depression in the medial prefrontal cortex sets the "life time" of high-dimensional neuronal assemblies that code for the integrated representation of position and sensory inputs (*Fujisawa et al.*, 2008). While the rat navigates in a maze, the representation moves from one assembly to another on a time scale that roughly corresponds to the one of synaptic depression. Short-term synaptic plasticity, originally found in rat sensory cortices (*Zucker and Regehr*, 2002), has also been found in the prefrontal cortex using paired recordings in vitro *Hempel et al.* (2000).

As a potential downside of the functional gains discussed in the manuscript, the inclusion of more complex membrane and synapse dynamics are likely to increase the computational cost of applying our paradigm to classical neural networks such as Boltzmann machines. However, we expect a simple, local synaptic update rule to be overall more efficient than global updates required by, e.g, tempering schedules. Moreover, in physical neuromorphic emulation, added complexity in neural dynamics incurs no runtime penalty compared to conventional simulation platforms. Based on specifically designed hardware (*Pfeil et al.*, 2013; *Schemmel et al.*, 2010), our approach can potentially create fast and energy-efficient physical models of neuro-synaptic dynamics.

In a physical system such as a biological brain, the studied plasticity mechanism essentially comes for free, as it only requires a limited pool of synaptic resources. Together with other activity-modulating mechanisms such as neuronal adaptation, it could be a key contributor to the ability of the brain to navigate efficiently in a very-high-dimensional stimulus space. Importantly, these mechanisms provide immediate computational advantages for spike-based neuromorphic devices, facilitating the development of efficient artificial agents that replicate the inferential capabilities of their biological archetypes.

# 5. RBMs with STP

First invented by Smolensky in 1986 (*Smolensky*, 1986), RBMs and related deep architectures are among the earliest efficient discriminative and generative models in deep learning (*Hinton et al.*, 2006; *Salakhutdinov and Hinton*, 2009) and are used in recent applications such as concept learning (*Mordatch*, 2018) and many-body quantum mechanics (*Carleo and Troyer*, 2017). However, as introduced in chapter 2, its mixing problem inherited from general MCMC methods has proposed a challenge for sampling and learning tasks, particularly in high dimensional space.

In the previous chapter, we have demonstrated the mixing-facilitation mechanism of STP by modulating active local attractors and inhomogeneously modifying the energy landscape. The high efficiency of this self-adaptive principle compared to traditional solutions naturally motivates the question of whether this biologically inspired mechanism can be adapted to traditional machine learning algorithms and artificial neural networks. As a start, throughout this chapter, we study the implementation of STP in traditional RBMs and test its influences on sampling and mixing performances of the network in several experiments.

Specifically, we first give a brief introduction of the model and then measure the sampling accuracy of STP-RBMs with different STP parameter configurations using the Kullback-Leibler divergence. Subsequently, we increase the network dimension and study its mixing behavior in a previous designed experiment where we compare and identify the differences between the STP-RBM and the LIF network with STP. In the end, we apply STP-RBMs to high dimensional generative tasks of handwritten digits, in which we compare the generation performances of different STP parameters and discuss their corresponding influences on sampling dynamics.

Preliminary studies of STP-RBM were done together with Ruyi Zuo, however, works presented in this chapter has its own focus.

## 5.1. RBM with STP in sampling

In this section, we implement STP into the sampling process of traditional RBMs and study its influence on generative properties of the network.

We select RBMs for the implementation rather than general BMs because it is convenient to make comparisons of mixing in high-dimensional generation tasks from the previous chapter. In the RBM, the binary state $z = 1$ is interpreted as a spike with a duration of one sampling step, corresponding to the refractory period of the LIF neuron. With STP embedded in the sampling process, the only variable we modify is the potential of the neuron. Since STP modulates the connection weights depending on the firing history of the afferent neuron, the potential of a neuron can thus be expressed as:

$$u_j = \sum_{i=1}^{I} R_i^n U_i^n W_{ji} z_i + b_j \ , \tag{5.1}$$

$$\tag{5.2}$$

where $R_i^n$ and $U_i^n$ are described similarly as in Eq. 4.4, 4.5:

$$R_i^{n+1} = R_i^n(1 - U_i^{n+1})\exp\left(\frac{-\Delta t}{\tau_{\text{rec}}}\right) + 1 - \exp\left(\frac{-\Delta t}{\tau_{\text{rec}}}\right) \tag{5.3}$$

$$U_i^{n+1} = U_i^n \exp\left(\frac{-\Delta t}{\tau_{\text{fac}}}\right) + U_0\left[1 - U_i^n \exp\left(\frac{-\Delta t}{\tau_{\text{fac}}}\right)\right] \tag{5.4}$$

where $\Delta t$ is the number of sampling steps from the last spike.

The original sampling process in the RBM is Gibbs sampling, which is an MCMC method and guarantees convergence to the equilibrium distribution given enough simulation time. By clamping $R$ and $U$ to be constant at 1 the STP sampling retrieves back to Gibbs sampling. However, in practice, these variables will change according to the neuron's self-firing activity and break symmetric interactions between neuron pairs as defined in BMs, which is required for sampling from the target distribution. This can be further elaborated in the derivation of the conditional firing probability using the Bayesian rule.

The probability of neuron $k$ to fire given the states of other neurons, $p(z_k = 1|z_{\setminus k})$ is expressed as according to the Bayesian rule:

$$\begin{aligned} p(z_k = 1|z_{\setminus k}) &= \frac{p(z_k = 1, z_{\setminus k})}{p(z_{\setminus k})} \\ &= \frac{p(z_k = 1, z_{\setminus k})}{p(z_k = 1, z_{\setminus k}) + p(z_k = 0, z_{\setminus k})} \\ &= \frac{1}{1 + \dfrac{p(z_k = 0, z_{\setminus k})}{p(z_k = 1, z_{\setminus k})}} \ . \end{aligned} \tag{5.5}$$

When we plug into the Boltzmann distribution (Eq. 2.4), the second term of the denominator in Eq. 5.5 becomes

$$\begin{aligned}\frac{p(z_k=0,z_{\setminus k})}{p(z_k=1,z_{\setminus k})} &= \frac{\frac{1}{Z}\exp\big[-E(z_k=0,z_{\setminus k})\big]}{\frac{1}{Z}\exp\big[-E(z_k=1,z_{\setminus k})\big]} \\ &= \exp\big[-E(z_k=0,z_{\setminus k})+E(z_k=1,z_{\setminus k})\big] \\ &= \exp\Big(\sum_{i,j\neq k}\frac{1}{2}W_{ij}z_iz_j+\sum_{i\neq k}b_iz_i-\sum_j W_{kj}z_j-\sum_{i,j\neq k}\frac{1}{2}W_{ij}z_iz_j-b_k-\sum_{i\neq k}b_iz_i\Big) \\ &= \exp\Big(-\sum_j W_{kj}z_j-b_k\Big)\,. \end{aligned} \tag{5.6}$$

So the conditional firing probability is expressed as

$$p(z_k=1|z_{\setminus k}) = \frac{1}{1+\exp\Big[-\big(\sum_j W_{kj}z_j+b_k\big)\Big]}\,, \tag{5.7}$$

where $\sum_j W_{kj}z_j + b_k$ is the original formulation of the potential $u_k$ (Eq. 3.7). Notice that the derivation from the third last to the second last step in Eq. 5.6 only holds when $W_{kj} = W_{jk}$. Therefore, the induced STP variables in the neural potential (Eq. 5.1) no longer guarantees convergence to the predefined Boltzmann distribution. Further researches are ongoing regarding the convergence of STP-RBMs, including by forcing symmetric connections within each sampling step. Our aim here is a direct mechanistic approximation of the LIF network with STP.

To test the sampling accuracy and mixing performances of STP-RBMs, we set up several experiments similar to the previous chapter.

### 5.1.1. Sampling from a fully specified target distribution

We start from the case of a fully specified target distribution as presented in section 4.2.1 with the same network parameters. We scan through different combinations of STP parameters and measure the sampling accuracy of the network with DKL. For parameters with $U_0 < 1$, the weights of the network are rescaled by dividing $f_w = U_0$, the same as in section 4.2.1. Each parameter set is initialized with 5 different random seeds and run for 2 $\times 10^5$ steps until convergence. The averaged DKLs of all random seeds are plotted in Fig. 5.1.

We find that an optimal reproduction of the target distribution is achieved at ($U_0 = 1, \tau_{\text{rec}} = 0$) and for all parameter sets with ($\tau_{\text{rec}} = 0, \tau_{\text{fac}} = 0$), which are essentially Gibbs sampling. This is as expected since Gibbs sampling guarantees convergence to the target distribution and any perturbations induced by other STP parameters will deviate from it.

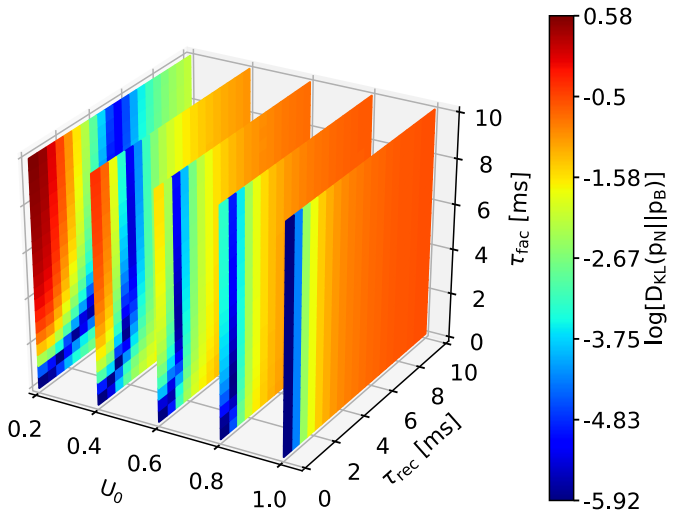


Figure 5.1.: Kullback-Leibler divergence between sampled ($p_{\mathrm{N}}$) and target ($p_{\mathrm{B}}$) distribution of the STP-RBM with 10 neurons (5 hidden, 5 visible) for different STP parameters ($U_0, \tau_{\mathrm{rec}}, \tau_{\mathrm{fac}}$). The minimum value is achieved with parameter sets with $\tau_{\mathrm{rec}} = 0, \tau_{\mathrm{fac}} = 0$, which are essentially Gibbs sampling. Note that for $U_0 = 1$, the facilitation mechanism is shut down by definition (Eq. 5.4) and $U$ becomes a constant, i.e. the effective $\tau_{\mathrm{fac}}$ equals 0. change u to U0

### 5.1.2. Mixing in a simple learning scenario

To further investigate the mixing performance of STP-RBMs in high dimensional tasks, we apply the STP-RBM to previous bar experiments in section 4.2.2.

To facilitate mixing, we construct STD envelopes similar to those used in the LIF-based RBM. The guiding rule is to equalize the average synaptic impact within each sampling step between two networks. However, due to the tail-overlapping phenomenon in LIF PSPs, it is impossible to match the envelope of rectangular PSPs of the STP-RBM to be entirely the same as the exponential ones, even when using the same STP parameter sets. This tail effect of exponential PSPs becomes more evident in the case of lager inter-spike intervals (ISIs).

Figure 5.2 gives an intuitive illustration of this problem. The left plot shows PSPs of a single neuron triggered by spikes trains of another neuron with ISI = 10ms (maximum firing frequency, since the refractory period of the LIF neuron is set to 10 ms), and right plot with ISI = 20ms. The equivalent integrated rectangular area of the exponential PSP within each sampling step[1] is indicated by light red shadows. When using parameter set ($U_0 = 1.0, \tau_{\mathrm{rec}} = 1.9, \tau_{\mathrm{fac}} = 0.0$) for the STP-RBM which is effectively the same parameter set as the parameter set ($U_0 = 1.0, \tau_{\mathrm{rec}} = 19.0\mathrm{ms}, \tau_{\mathrm{fac}} = 0.0\mathrm{ms}$) for the LIF neuron , the integrated area under PSPs of the STP-RBM is significantly lower than the

[1] For the LIF-sampling framework, one sampling step is defined as the duration of one refractory period.

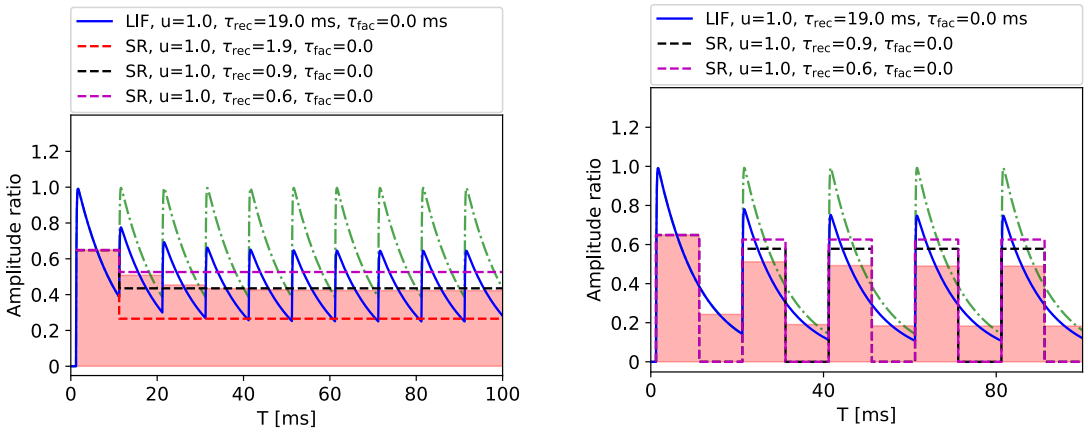


Figure 5.2.: PSPs triggered by consecutive inputs with constant ISI of 10 ms (left) adnd 20 ms (right). The red shadow represents rectangular PSPs with the equivalent integral area as the exponential PSPs (blue) over each ISI duration. The light green dashed line denotes PSPs of a renewing synapse. In practice, for STP-RBM (SR) we use STP parameters with slightly larger envelope to compensate for the exponential tails for the case of ISI $>$ 10 ms.

LIF counterparts, due to consecutive PSPs in the LIF neuron will overlap upon the tails from former PSPs. In the case for ISI = 10ms, compensation can be made by decreasing $\tau_{\text{rec}}$ for the STP-RBM to 0.9 (black dash line). However, when the ISI is doubled[2] the tails will continuously influence the membrane potential which is not the case for rectangular PSPs in the STP-RBM. In practice, this gives a latency for the sampling dynamics of LIF networks and leads to imprecisions of sampling. When sampling from multimodal distributions with strong attractors, in the majority of the period, neurons either fire with minimum ISI or remain silent. Continuous firing with low frequencies are considered to be minor cases and their influences on overall sampling statistics are limited, but might scale with the size of the network.

We apply two sets of STP parameters for the STP-RBM in the bar generation experiments, one with $(U_0 = 1.0, \tau_{\text{rec}} = 0.9, \tau_{\text{fac}} = 0.0)$ and the other $(U_0 = 1.0, \tau_{\text{rec}} = 0.6, \tau_{\text{fac}} = 0.0)$ as shown in Fig. 5.2. The latter one is designed to have a higher envelope endeavored to compensate the effect of tails for large ISI in a certain degree, though it also induces more deviations for the minimum ISI case.

Similar approaches as in section 4.3.1 are taken to inspect the modulation effect of STP on marginal probability distributions of network states. We collected all emerged marginal states in a simulation of 5000 sampling steps (4998 visible and 2867 hidden states) and the average ratio variation of $p^*(\mathbf{h})$ in terms of different modes are plotted in

[2]We show doubled ISI here for the convenience of comparison, in practice, the ISI for a LIF neuron can change continuously.

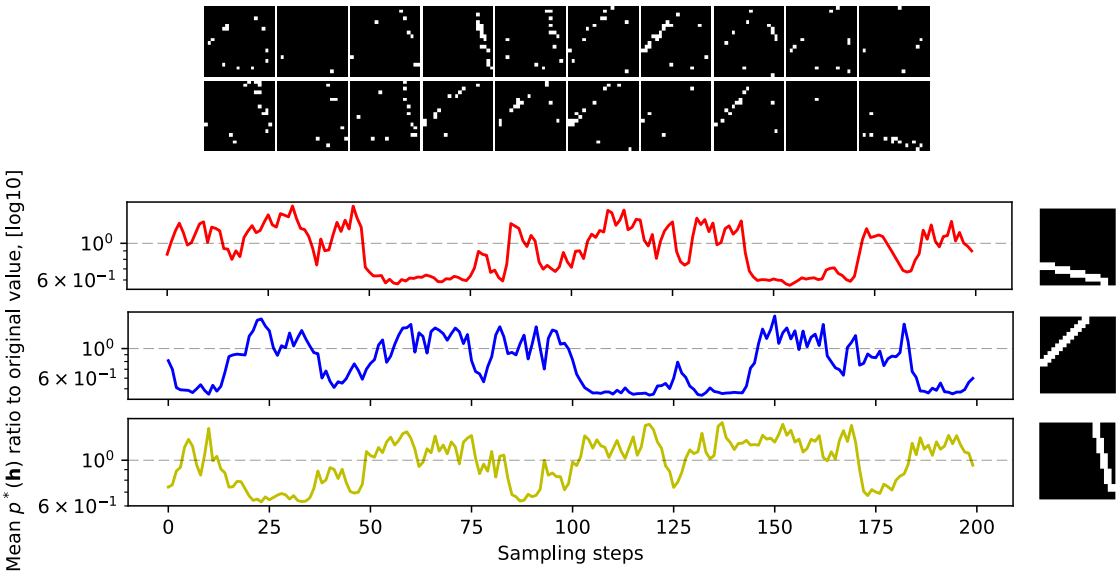


Figure 5.3.: **Top:** Images generated from an STP-RBM with ($U_0 = 1.0, \tau_{\text{rec}} = 0.9, \tau_{\text{fac}} = 0.0$). Samples cover 200 sampling steps (corresponding to the bottom plot, count by rows, from left to right) with a sample interval of 10. The pixels of most images are scatter into multiple modes, less concentrated than images produced with $\tau_{\text{rec}} = 0.9$ in Fig. 4.23. **Bottom:** Ratio variations of averaged $p^*(\mathbf{h})$ for each mode during simulation. Their corresponding modes are plotted in the end.

Fig. 5.3 and 5.4. By comparing the generated image quality, one can see that network with $\tau_{\text{rec}} = 0.6$ produces better separated bars than $\tau_{\text{rec}} = 0.9$ and its variation of $p^*(\mathbf{h})$ also more resembles to what we observed in the LIF case (Fig. 4.23). This indicates that for the STP-RBM, a relative higher envelope in the minimum ISI case could better reproduce the dynamics of the LIF counterparts. However, more simulations need to be performed to find an optimal range of envelopes.

The t-SNE plots for STP-RBM with $\tau_{\text{rec}} = 0.6$ during 3 mode switches are shown in Fig. 5.5. While similar local modulation effects of STP are found compared to the LIF case (Fig. 4.19), we see a clearer separation of states between different modes and less randomness in terms of a more uniform variation of ratio within the same cluster. This could be largely attributed to synchronized synaptic transmission of the STP-RBM, which is fundamentally different from the LIF network.

### 5.1.3. Generation of handwritten digits

We further applied STP-RBMs to generation tasks of MNIST handwritten digits in which the same model parameters (weights and biases) are used as in section 4.2.3.

Based on the experience from the previous section, we again apply two sets of STP

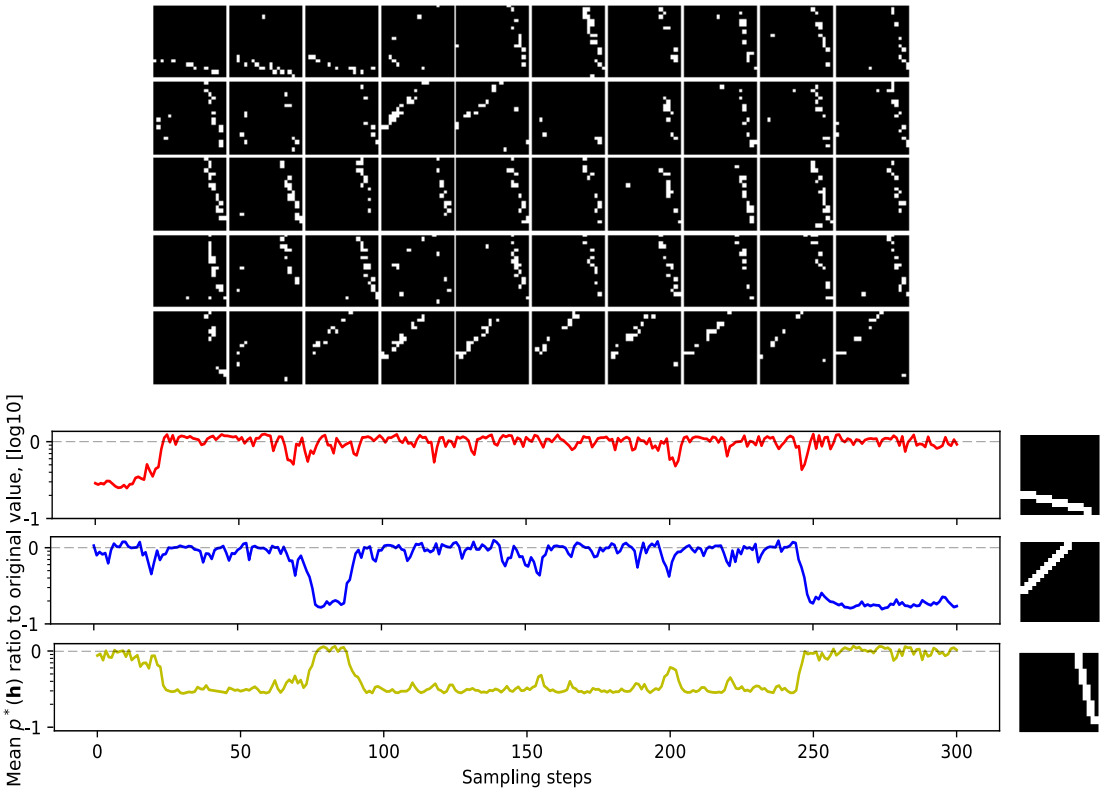


Figure 5.4.: **Top:** Images generated from an STP-RBM with ($U_0 = 1.0, \tau_{\text{rec}} = 0.6, \tau_{\text{fac}} = 0.0$). Samples cover 300 sampling steps (corresponding to the bottom plot, count by rows, from left to right) with a sample interval of 6. **Bottom:** Ratio variations of averaged $p^*(\mathbf{h})$ for each mode during simulation. Their corresponding modes are plotted in the end. The variation resembles to what we observed in the LIF case (Fig. 4.23).

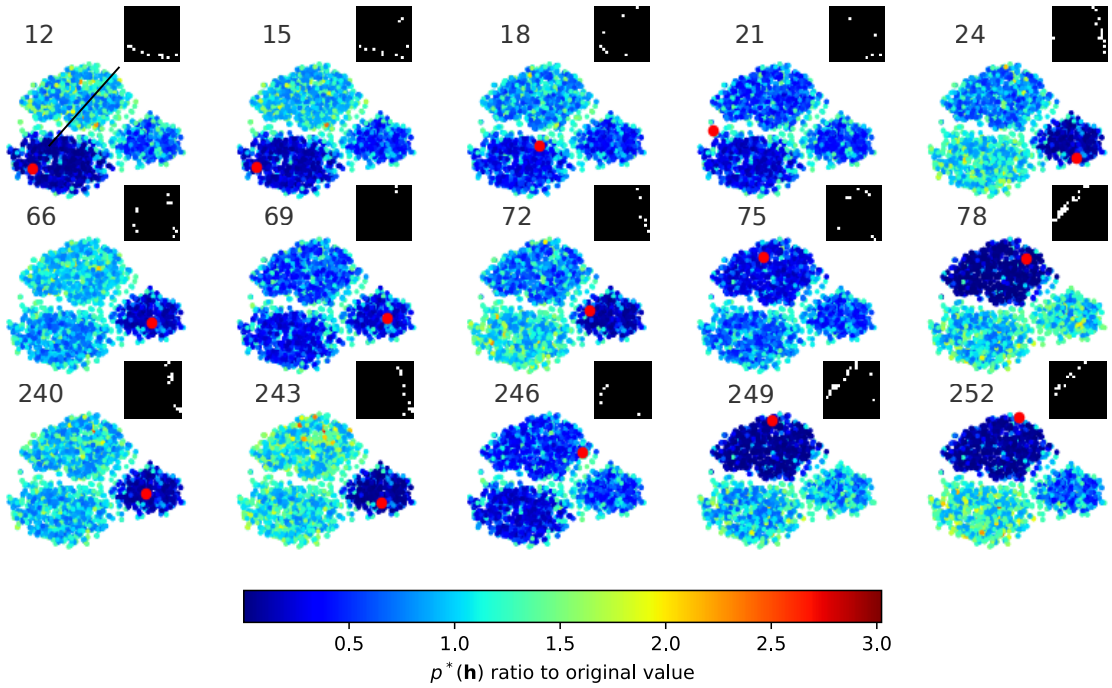


Figure 5.5.: Ratio change of an ensemble of $p^*(\mathbf{h})$ during three mode switches observed from simulation, i.e. sampling steps from 12-24 (fist row), 66-78 (second row) and 240-252 (third row). The whole image sequences are plotted in Fig. 5.4 top. The red dot indicates the position of the current network state, and its corresponding visible state is shown on the top right. Similar local modulation effect of STP occurs as in the case of LIF-based RBM with STP (Fig. 4.19). However, it can be observed that ratio changes in the same cluster are more uniform compare to the LIF case.

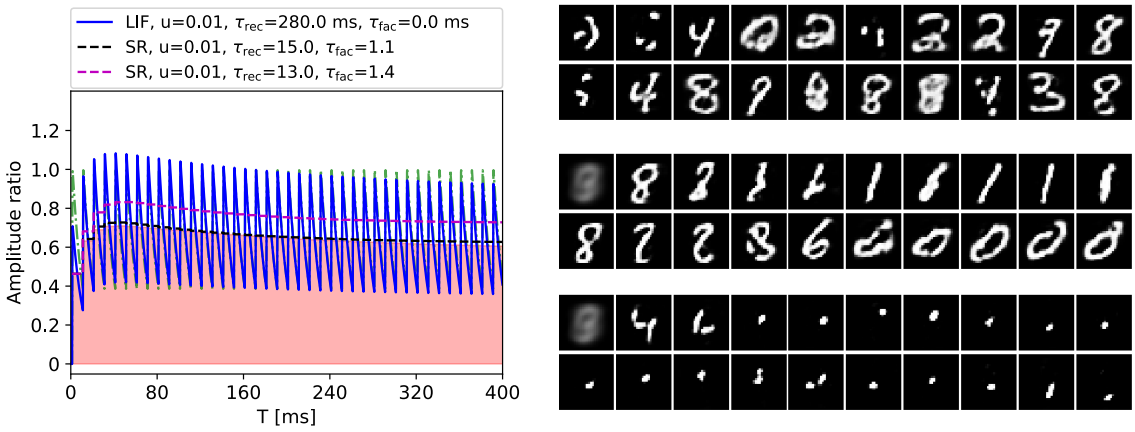


Figure 5.6.: **Left:** PSPs triggered by consecutive inputs with constant ISI of 10 ms. The STP parameter set (blue solid line) of the LIF network is the same as the one used in section 4.2.3 for generation of handwritten digits. The red shadow represents rectangular PSPs with the equivalent integral area as the exponential PSPs (blue) over each ISI duration. The light green dashed line denotes PSPs of a renewing synapse. Two sets of STP parameters are plotted for the STP-RBM, both with $f_w$ of 0.014, the same as the LIF PSPs. **Right:** Image samples generated from the LIF network (top) and STP-RBMs (middle: $\tau_{\text{rec}} = 15.0$, bottom: $\tau_{\text{rec}} = 13.0$). A black pixel represents 0 firing probability of the corresponding visible neuron and white corresponds to 1. The pixel value is generated from the conditional firing probability of the visible neuron given the state of hidden layer.

parameters: $(U_0 = 0.01, \tau_{\text{rec}} = 13.0, \tau_{\text{fac}} = 1.4)$ and $(U_0 = 0.01, \tau_{\text{rec}} = 15.0, \tau_{\text{fac}} = 1.1)$. The former is a close match of the exponential PSPs used for the LIF network in the minimum ISI case and the latter creates a slightly higher envelope, as shown in Fig. 5.6. The generated images show that the STP-RBM with higher envelope generates samples of similar qualities to the LIF network, which is in accordance with the hypothesis in the previous section. Notice that different from the previous bar experiment, a close PSP envelope match at the minimum ISI now generates almost blank pixels, this could be due to the increase of the size of the network and more complicated probability distribution, which magnifies the integral difference caused by the tails of exponential PSPs.

In addition, we further investigate the influences of difference STP envelopes on sampling dynamics in terms of the firing rate of the hidden layer, as shown in Fig. 5.7. Together we plot the corresponding STP envelopes and image samples produced by these networks as references.

The result shows that a pure STD for the LIF network with $(U_0 = 1.0, \tau_{\text{rec}} = 15.0\text{ms}, \tau_{\text{fac}} = 0.0\text{ms})$ decreases the strength of each individual synaptic transmis-

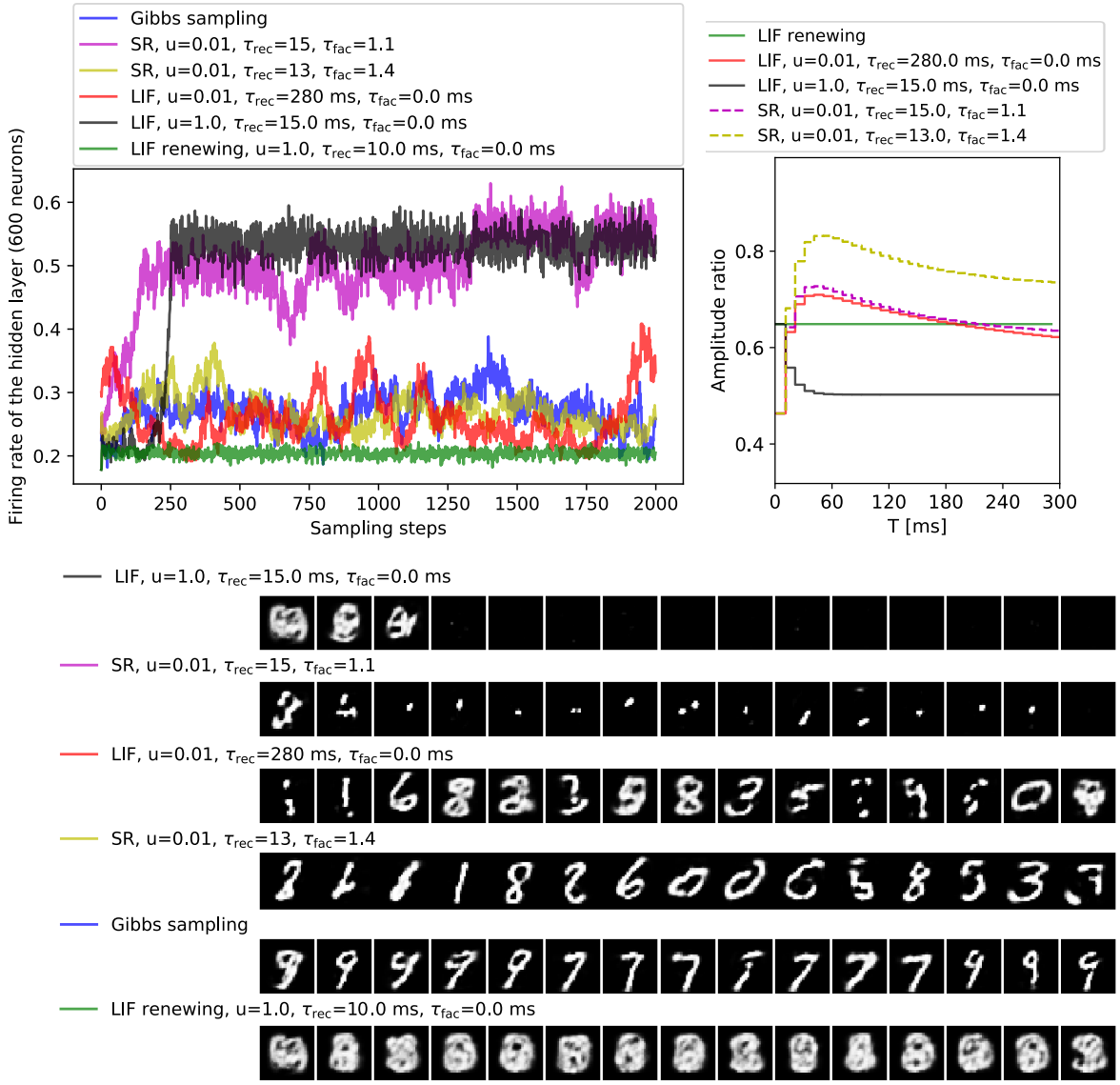


Figure 5.7.: **Top left:** Firing rate of the hidden layer (600 neurons in total) for networks with different STP parameter sets. **Top right:** Envelopes of PSPs caused by consecutive spikes with minimum ISI. Envelopes of equivalent rectangular PSPs of LIF neurons are denoted with solid lines. **Bottom:** Images generated by networks with corresponding STP parameters. Each image sequence cover 1500 sampling steps with a sample interval of 100.

sion leading to a much higher firing ratio in the hidden layer and insufficient firing in the visible layer. This also occurs for the STP-RBM with ($U_0 = 0.01, \tau_{\text{rec}} = 13.0, \tau_{\text{fac}} = 1.4$), indicating a similar scenario.

For the two STP parameter sets, i.e. ($U_0 = 0.01, \tau_{\text{rec}} = 13.0, \tau_{\text{fac}} = 1.4$) and ($U_0 = 0.01, \tau_{\text{rec}} = 280.0\text{ms}, \tau_{\text{fac}} = 0.0\text{ms}$) which produce recognizable images, their firing rates are in similar range as Gibbs sampling. This indicates a good approximation of sampling dynamics to the benchmark network. Note that their images switch to different modes much faster than Gibbs sampling.

Finally, for the LIF network using renewing synapse with ($U_0 = 1.0, \tau_{\text{rec}} = 10.0\text{ms}, \tau_{\text{fac}} = 0.0\text{ms}$), the firing rate in the hidden layer is much lower compared to other networks and the visible layer much higher, resulting in quite blurred images. This could be largely due to the tail effect of the exponential PSP which causes deviation to the ideal sampling process.

Notice that for the LIF network a potentiation-depression envelope close to the renewing envelope approximates Gibbs sampling dynamics much better than the renewing synapse itself. This is in accordance to what we observed previously in Fig. 4.15. Theoretically, similar latency effects should also occur for potentiation-depression envelopes, however, it seems that this special pattern with a certain deviation from the benchmark can counteract the effect in a degree. This specific shape of PSP envelope could be necessary and particularly effective for large size networks where latency effects can be alleviated by asynchronous activities from other neurons encoding the same mode. Interestingly, similar shapes are also found in biology (*Tsodyks and Markram*, 1997).

## 5.2. Discussion

Throughout this chapter, we discussed the implementation of STP in the sampling of RBMs and presented initial results on several applications ranging from low to high dimensions. When sampling from a distribution defined in a limited state space where mixing is easy, the STP mechanism seems to be redundant for fast convergence to the target distribution.

For multimodal distributions in higher dimensions, STP-RBMs achieved similar performances as LIF networks, however with a different range of optimal STP parameters due to the difference in PSP shapes. Based on experiments, we derived a hypothesis (or rule of thumb) that the envelope of STP-RBM under minimum ISI needs to be slightly higher than the LIF one in order to compensate for the potential latency effects and achieve similar sampling statistics. More simulations are needed to quantify an optimal range of parameters and collecting the firing spectrum of all neurons can be helpful for the investigation.

In this study, a quantitative comparison between the mixing performances of STP-RBMs and LIF networks is still missing, which can be performed in the future by a broad sweep of potential STP parameter configurations and a subsequent ISL calculation, as in section 4.2.3. We conjecture that the overlapping effect of exponential

PSPs for individual synapse combining with STP could make the LIF network a sharper filter than the STP-RBM, enable it to jump in and out of a local mode faster. Relate works concerning sampling with different PSP shapes can be found in *Gürtler* (2018). Furthermore, from a theoretical perspective, a convergence study of STP-RBMs remains to be done, related works can be found in *Apolloni et al.* (1991). Moreover, the influence of synchronized computation in mixing needs to be further investigated, a comparison can be made by implementing STP in the abstract spiking neuron model introduced in section 2.2.1, which uses rectangular PSP but with asynchronous computing.

Another potential future direction is to study the implementation of STP in learning. However, one needs to be cautious if STP-endowed sampling is used for the approximation of the model distribution due to its mixing advantage, which could be helpful if the distribution is multimodal but could also be harmful when the distribution is imbalanced as demonstrated in section 4.2.4. For the latter case, a more careful choice of STP parameters is needed. Another option is to run the STP-endowed sampling as a parallel chain to facilitate mixing.

# 6. Conclusion & Outlook

## Conclusion

In this research, we started from a brief introduction of generative models and the corresponding mixing problem, then focused on Boltzmann machines which were used as benchmark models throughout the work because of their efficiency and resemblance to biological neural networks (section 2.1). Subsequently, we discussed the implementation of stochastic sampling on LIF neurons (section 2.2) and scaled up the network for high dimensional generative and discriminative tasks, where they rival or surpass traditional machine learning counterparts by leveraging certain biological mechanisms (chapter 3 and 4).

By studying the membrane potential distribution of the LIF neuron and its activation function (section 3.1), we established a mapping relation between the rate of background Poisson noise and the temperature of energy based models (section 3.2). On the network level, we further developed a spike-based tempering framework implemented with sinusoidal background noise inspired by neural oscillation, which generates with sufficient mixing comparable to traditional tempering methods (section 3.3).

In contrast to noise which changes the dynamics of the global system, synaptic short-term plasticity (section 4.1) modulates the local active population by facilitating or depressing instant synaptic efficacy. With a certain range of STP parameters, we created modulated synapses with specific shapes of PSP envelopes (section 4.1.2) and implemented them in LIF networks for generative tasks on different scales (section 4.2). The activity-dependent modulation mechanism of STP naturally facilitates the system to escape from local attractors, outperforming plain MCMC methods in mixing meanwhile maintaining its discriminative ability (section 4.2.3). When applied to imbalanced datasets, while traditional sampling methods stuck in the majority mode, with different synaptic envelopes created by STP the LIF network realizes diverse sampling statistics, demonstrating its versatility (section 4.2.4). In addition, we revealed the effect of STP on the probability distribution of network states, providing a theoretical explanation for its functionality (section 4.3).

Motivated by the mixing advantage of STP-endowed sampling, we applied a similar mechanism to traditional restricted Boltzmann machines and studied their performances with preliminary experiments (chapter 5). The variation of PSP shape and synchrony of computation make the system different from LIF networks. We discussed potential effects caused by these factors and demonstrated that similar generative performances can be achieved with a different range of parameters.

## 6.1. Outlook

Plain MCMC methods are well known for their pool mixing in high dimensional multimodal distributions (*Salakhutdinov*, 2010; *Bengio et al.*, 2013). Traditional solutions like tempering algorithms usually have high computational cost and are inefficient in obtaining valid samples. The STP-endowed sampling mechanism takes a different strategy: the active attractor triggers the modulation of STP which in turn causes the deactivation of itself. More versatile sampling algorithms can be inspired by this adaptive mechanism, with the potential for different sampling tasks by modifying only a few parameters. To this end, a theoretical study of how variations of PSP envelope leads to different sampling statistics on certain probability distributions is needed. With a spiking history dependency, STP-endowed sampling loosely resembles a high order Markov chain. Their connections could be a direction for future studies.

Throughout the work, we train the network using traditional learning algorithms and then map the model parameters to the LIF domain. As demonstrated in our experiments, the latency effect causes deviation of the mapping which eventually contributes to the imprecision of sampling. A direct on-line learning algorithm of the LIF network could potentially solve this problem. It has been recently proved that contrastive divergence can be approximated by spike-timing-dependent plasticity (*Neftci et al.*, 2014). However, its robustness to noise, which is inevitable for analog neuromorphic hardware, is still questionable. Counter solutions have been proposed and researches are ongoing. This approach could be integrated with STP-endowed sampling to create more efficient learning algorithms for generative SNNs.

Currently, the scale of our LIF networks is limited by the computation power of conventional simulation platforms. The efficiency of our framework will be magnified through a physical neuromorphic emulation where added complexity in neural dynamics incurs no runtime penalty. A circuit emulating STP (*Billaudelle*, 2017) has recently been implemented in the latest HICANN-DLS chip (*Aamir et al.*, 2018) which covers a broad range of optimal parameters that can facilitate the mixing of LIF networks. Based on specifically designed hardware (*Schemmel et al.*, 2010; *Aamir et al.*, 2018), our approach can potentially create fast and energy-efficient physical models of neuro-synaptic dynamics.

# A. Appendix

## A.1. Acronyms and Abbreviations

**AIS** - Annealed Importance Sampling

**ANN** - Artificial Neural Network

**AST** - Adaptive Simulated Tempering

**BM** - Boltzmann Machine

**CAST** - Coupled Adaptive Simulated Tempering

**GAN** - Generative Adversarial Network

**ISI** - Inter-Spike Interval

**ISL** - Indirect Sampling Likelihood

**MCMC** - Markov Chain Monte Carlo

**PCD** - Persistent Contrastive Divergence

**PSP** - PostSynaptic Potential

**RBM** - Restricted Boltzmann Machine

**SNN** - Spiking Neural Network

**STD** - Short-Term Depression

**STF** - Short-Term Facilitation

**STP** - Short-Term Plasticity

**VAE** - Variational Autoencoder

# A.2. Supplementary Information

## A.2.1. Neuron Parameters

Table A.1 lists the LIF CUBA and COBA neuron parameters used in simulations in chapter 3 and chapter 4.

Table A.1.: Neuron parameters

| | COBA | CUBA | |
|---|---|---|---|
| $C_\mathrm{m}$ | 0.1 nF | 0.2 nF | membrane capacitance |
| $\tau_\mathrm{m}$ | 20 ms | 0.1 ms | membrane time constant |
| $\tau_\mathrm{ref}$ | 10 ms | 10 ms | refractory time constant |
| $\tau_\mathrm{syn}$ | 10 ms | 10 ms | synaptic time constant |
| $\vartheta$ | −50 mV | −50 mV | threshold voltage |
| $\rho$ | −53 mV | −50.01 mV | reset potential |
| $E^\mathrm{rev}_\mathrm{exc}$ | 0 mV | - | excitatory reversal potential |
| $E^\mathrm{rev}_\mathrm{inh}$ | −100 mV | - | inhibitory reversal potential |

To speed up simulations, we used an effective current-based (CUBA) model to replace the COBA one (Table A.1). Fig. A.1 shows a comparison between the two models. Under appropriate parametrization, we could reduce the background input rates from $\nu = 5\,\mathrm{kHz}$ to $\nu = 0.4\,\mathrm{kHz}$.

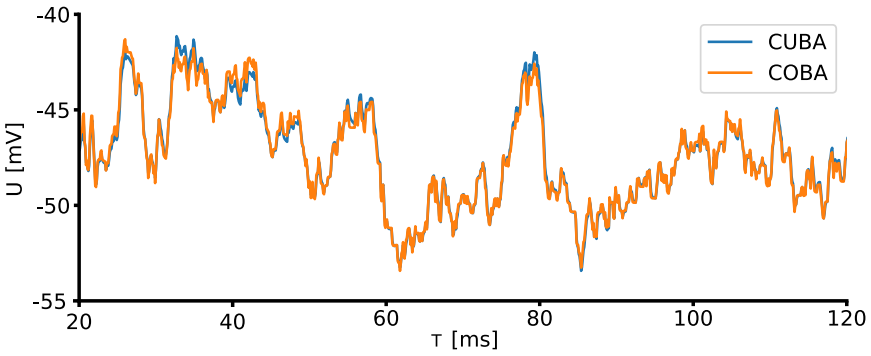


Figure A.1.: Free membrane potential ($\vartheta = 0$) of a biologically plausible COBA LIF neuron in the HCS compared to an equivalent CUBA LIF neuron (parameters given in Table A.1).

## A.2.2. Training hyperparameters

The used hyperparameters (number of epochs $T$, batch size $N$, learning rate $\eta$) were based on suggestions from previous work *Hinton* (2010) and empirical experience. For all datasets trained with CAST, we used 20 equidistant inverse temperatures $\beta_k \in [0.9, 1]$.

The adaptive weights $\{\mathbf{g}_k\}_{k=1}^K$ were initialized to 1 for all temperatures and as $\gamma_t \to 0$ the adaptive weights will converge. In all experiments, we set $\gamma_t$ as $90/(150+t)$.

For the bar example (section 4.2.2), we used $T = 100,000$, $N = 3$ and $\eta = 10/(2000+t)$.

For the full MNIST example (section 4.2.3), we used $T = 200,000$, $N = 100$ and $\eta = 40/(t+2000)$.

For the first example of an imbalanced dataset (Fig. 4.15), we used a network with 784 visible, 10 label and 400 hidden units with $T = 100,000$, $N = 100$ and $\eta = 20/(t+2000)$.

For the example of pattern completion from an imbalanced dataset (Fig. 4.17), we used a network with 784 visible, 10 label and 400 hidden units with $T = 200,000$, $N = 100$ and $\eta = 40/(t+2000)$.

For the reduced bar experiment in section 4.3.2, we trained the network using PCD with $N = 3$ and $\eta = 15/(t+2000)$.

### A.2.3. t-distributed stochastic neighbor embedding

The t-SNE method (*Maaten and Hinton*, 2008) finds a low-dimensional map for a high-dimensional data set, in which the similarity between samples is reflected by their distances in the low-dimensional map. Here, we projected the generated digits to a plane to provide an intuitive understanding of the network dynamics and the mixing between different modes (digit classes). The Euclidean distances between high-dimensional samples $\{\boldsymbol{x}_i\}$ are converted into symmetric pairwise similarities

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \quad , \tag{A.1}$$

where $n$ is the number of samples and $p_{j|i}$ is a conditional probability:

$$p_{j|i} = \frac{\exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_k\|^2/2\sigma_i^2)} \quad , \tag{A.2}$$

with variance $\sigma_i$, which is determined by first defining a so-called perplexity value as the effective number of neighbors of a data point, and then running a binary search. For the low-dimensional points $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ mapped from the high-dimensional data points $\mathbf{x}_i$ and $\mathbf{x}_j$, the similarity is defined using a t-distribution with one degree of freedom:

$$q_{ij} = \frac{(1 + \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)^{-1}}{\sum_{k\neq l}(1 + \|\boldsymbol{y}_k - \boldsymbol{y}_l\|^2)^{-1}} \quad . \tag{A.3}$$

If the mapped points correctly model the similarity between the high-dimensional data points, the similarities $p_{ij}$ and $q_{ij}$ will be equal.

With this motivation, tSNE minimizes the sum of Kullback-Leibler divergences over all data points using a gradient descent method. The cost function $C$ is given by

$$C = D_{KL}(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad . \tag{A.4}$$

Its gradient with respect to the map point $i$ can then be derived to provide an update of the mapping:

$$\Delta \boldsymbol{y}_i \propto \frac{\partial C}{\partial \boldsymbol{y}_i} = 4 \sum_j (p_{ij} - q_{ij})(\boldsymbol{y}_i - \boldsymbol{y}_j)(1 + \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)^{-1} \quad . \tag{A.5}$$

# Bibliography


Aamir, S. A., Y. Stradmann, P. Müller, C. Pehle, A. Hartel, A. Grübl, J. Schemmel, and K. Meier, An accelerated lif neuronal network array for a large-scale mixed-signal neuromorphic architecture, *IEEE Transactions on Circuits and Systems I: Regular Papers*, (99), 1–14, 2018.

Abbott, L., and W. G. Regehr, Synaptic computation, *Nature*, *431*(7010), 796, 2004.

Apolloni, B., A. Bertoni, P. Campadelli, and D. de Falco, Asymmetric boltzmann machines, *Biological cybernetics*, *66*(1), 61–70, 1991.

Bengio, Y., and Y. LeCun, Scaling learning algorithms towards ai, *Large-Scale Kernel Machines*, *34*, 1–41, 2007.

Bengio, Y., G. Mesnil, Y. Dauphin, and S. Rifai, Better mixing via deep representations., in *ICML (1)*, pp. 552–560, 2013.

Billaudelle, S., Design and implementation of a short term plasticity circuit for a 65 nm neuromorphic hardware system, Masterarbeit, Universität Heidelberg, 2017.

Breitwieser, O., Towards a neuromorphic implementation of spike-based expectation maximization, Masterarbeit, Universität Heidelberg, 2015.

Breuleux, O., Y. Bengio, and P. Vincent, Unlearning for better mixing, *Universite de Montreal/DIRO*, 2010.

Buesing, L., et al., Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons, *PLoS Comput Biol*, *7*(11), e1002,211, 2011.

Buzsáki, G., and A. Draguhn, Neuronal oscillations in cortical networks, *science*, *304*(5679), 1926–1929, 2004.

Carleo, G., and M. Troyer, Solving the quantum many-body problem with artificial neural networks, *Science*, *355*(6325), 602–606, 2017.

Chawla, N. V., Data mining for imbalanced datasets: An overview, in *Data mining and knowledge discovery handbook*, pp. 853–867, Springer, 2005.

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, *Journal of artificial intelligence research*, *16*, 321–357, 2002.

Dahl, G., A.-r. Mohamed, G. E. Hinton, et al., Phone recognition with the mean-covariance restricted boltzmann machine, in *Advances in neural information processing systems*, pp. 469–477, 2010.

Dale, H., Pharmacology and nerve-endings, 1935.

Davison, A. P., D. Brüderle, J. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, and P. Yger, Pynn: a common interface for neuronal network simulators, *Frontiers in neuroinformatics*, *2*, 2008.

Desjardins, G., A. Courville, Y. Bengio, P. Vincent, and O. Delalleau, Parallel tempering for training of restricted boltzmann machines, in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 145–152, MIT Press Cambridge, MA, 2010a.

Desjardins, G., A. Courville, Y. Bengio, P. Vincent, and O. Delalleau, Tempered markov chain monte carlo for training of restricted boltzmann machines, in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, vol. 9, pp. 145–152, 2010b.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805*, 2018.

Dold, D., I. Bytschok, A. F. Kungl, A. Baumbach, O. Breitwieser, W. Senn, J. Schemmel, K. Meier, and M. A. Petrovici, Stochasticity from function-why the bayesian brain may need no noise, *arXiv preprint arXiv:1809.08045*, 2018.

Fiser, J., P. Berkes, G. Orbán, and M. Lengyel, Statistically optimal perception and learning: from behavior to neural representations, *Trends in cognitive sciences*, *14*(3), 119–130, 2010.

Fuhrmann, G., I. Segev, H. Markram, and M. Tsodyks, Coding of temporal information by activity-dependent synapses, *Journal of neurophysiology*, *87*(1), 140–148, 2002.

Fujisawa, S., A. Amarasingham, M. T. Harrison, and G. Buzsáki, Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex, *Nature neuroscience*, *11*(7), 823, 2008.

García, S., and F. Herrera, Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy, *Evolutionary computation*, *17*(3), 275–306, 2009.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial nets, in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Gürtler, N., A markovian model of lif networks, Masterarbeit, Universität Heidelberg, 2018.

Hempel, C. M., K. H. Hartman, X.-J. Wang, G. G. Turrigiano, and S. B. Nelson, Multiple forms of short-term plasticity at excitatory synapses in rat medial prefrontal cortex, *Journal of neurophysiology*, *83*(5), 3031–3041, 2000.

Hindy, N. C., F. Y. Ng, and N. B. Turk-Browne, Linking pattern completion in the hippocampus to predictive coding in visual cortex, *Nature neuroscience*, *19*(5), 665, 2016.

Hinton, G., A practical guide to training restricted boltzmann machines, *Momentum*, *9*(1), 926, 2010.

Hinton, G. E., Training products of experts by minimizing contrastive divergence, *Neural computation*, *14*(8), 1771–1800, 2002.

Hinton, G. E., S. Osindero, and Y.-W. Teh, A fast learning algorithm for deep belief nets, *Neural computation*, *18*(7), 1527–1554, 2006.

Jezek, K., E. J. Henriksen, A. Treves, E. I. Moser, and M.-B. Moser, Theta-paced flickering between place-cell maps in the hippocampus, *Nature*, *478*(7368), 246, 2011.

Kingma, D. P., and M. Welling, Auto-encoding variational bayes, *arXiv preprint arXiv:1312.6114*, 2013.

Korcsak-Gorzo, A., Simulated tempering in spiking neural networks, Masterarbeit, Universität Heidelberg, 2017.

Krizhevsky, A., I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks, in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

LeCun, Y., The mnist database of handwritten digits, *http://yann. lecun. com/exdb/mnist/*, 1998.

LeCun, Y., Y. Bengio, and G. Hinton, Deep learning, *Nature*, *521*(7553), 436–444, 2015.

Lee, J. H., T. Delbruck, and M. Pfeiffer, Training deep spiking neural networks using backpropagation, *Frontiers in Neuroscience*, *10*, 2016.

Leng, L., M. A. Petrovici, R. Martel, I. Bytschok, O. Breitwieser, J. Bill, J. Schemmel, and K. Meier, Spiking neural networks as superior generative and discriminative models, *Cosyne Abstracts, Salt Lake City USA*, 2016.

Leng, L., R. Martel, O. Breitwieser, I. Bytschok, W. Senn, J. Schemmel, K. Meier, and M. A. Petrovici, Spiking neurons with short-term synaptic plasticity form superior generative networks, *Scientific reports*, *8*(1), 10,651, 2018.

Lillicrap, T. P., D. Cownden, D. B. Tweed, and C. J. Akerman, Random synaptic feedback weights support error backpropagation for deep learning, *Nature communications*, *7*, 13,276, 2016.

Maass, W., Networks of spiking neurons: the third generation of neural network models, *Neural networks*, *10*(9), 1659–1671, 1997.

Maaten, L. v. d., and G. Hinton, Visualizing data using t-sne, *Journal of Machine Learning Research*, *9*(Nov), 2579–2605, 2008.

Marinari, E., and G. Parisi, Simulated tempering: a new monte carlo scheme, *EPL (Europhysics Letters)*, *19*(6), 451, 1992.

Markram, H., Y. Wang, and M. Tsodyks, Differential signaling via the same axon of neocortical pyramidal neurons, *Proceedings of the National Academy of Sciences*, *95*(9), 5323–5328, 1998.

Martel, R., Generative properties of lif-based boltzmann machines, Master thesis, Ruprecht-Karls-Universität Heidelberg, hD-KIP 15-86, 2015.

Mead, C., Neuromorphic electronic systems, *Proceedings of the IEEE*, *78*(10), 1629–1636, 1990.

Mnih, V., et al., Human-level control through deep reinforcement learning, *Nature*, *518*(7540), 529, 2015.

Mordatch, I., Concept learning with energy-based models, 2018.

Neftci, E., S. Das, B. Pedroni, K. Kreutz-Delgado, and G. Cauwenberghs, Event-driven contrastive divergence for spiking neuromorphic systems, *Frontiers in neuroscience*, *7*, 272, 2014.

Neftci, E. O., C. Augustine, S. Paul, and G. Detorakis, Event-driven random back-propagation: Enabling neuromorphic deep learning machines, *Frontiers in Neuroscience*, *11*, 2017.

Oriol Vinyals, J. C. M. M. M. J. W. C. A. D. A. H. P. G. R. P. T. E. D. H. M. K. I. D. J. A. J. O. V. D. D. C. L. S. Y. S. S. V. J. M. T. C. D. B. T. P. C. G. Z. W. T. P. T. P. Y. W. D. Y. J. C. K. M. O. S. T. S. T. L. C. A. K. K. D. H. D. S., Igor Babuschkin, Alphastar-mastering-real-time-strategy-game-starcraft-ii, 2019.

Petrovici, M. A., *Form Versus Function: Theory and Models for Neuronal Substrates*, Springer, 2016.

Petrovici, M. A., J. Bill, I. Bytschok, J. Schemmel, and K. Meier, Stochastic inference with deterministic spiking neurons, *arXiv preprint arXiv:1311.3211*, 2013.

Petrovici, M. A., J. Bill, I. Bytschok, J. Schemmel, and K. Meier, Stochastic inference with spiking neurons in the high-conductance state, *Physical Review E*, *94*(4), 042,312, 2016.

Pfeil, T., et al., Six networks on a universal neuromorphic computing substrate, *Frontiers in neuroscience*, *7*, 2013.

Reif, M., F. Shafait, and A. Dengel, Meta-learning for evolutionary parameter optimization of classifiers, *Machine learning*, *87*(3), 357–380, 2012.

Rezende, D. J., S. Mohamed, and D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, *arXiv preprint arXiv:1401.4082*, 2014.

Ricciardi, L. M., and L. Sacerdote, The ornstein-uhlenbeck process as a model for neuronal activity, *Biological Cybernetics*, *35*, 1–9, 1979.

Sacramento, J., R. P. Costa, Y. Bengio, and W. Senn, Dendritic error backpropagation in deep cortical microcircuits, *arXiv preprint arXiv:1801.00062*, 2017.

Salakhutdinov, R., Learning deep boltzmann machines using adaptive mcmc, in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 943–950, 2010.

Salakhutdinov, R., and G. E. Hinton, Deep boltzmann machines., in *AISTATS*, vol. 1, p. 3, 2009.

Scellier, B., and Y. Bengio, Equilibrium propagation: Bridging the gap between energy-based models and backpropagation, *Frontiers in computational neuroscience*, *11*, 24, 2017.

Schemmel, J., D. Briiderle, A. Griibl, M. Hock, K. Meier, and S. Millner, A wafer-scale neuromorphic hardware system for large-scale neural modeling, in *Circuits and systems (ISCAS), proceedings of 2010 IEEE international symposium on*, pp. 1947–1950, IEEE, 2010.

Schmidhuber, J., Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook, Ph.D. thesis, Technische Universität München, 1987.

Schmidhuber, J., Deep learning in neural networks: An overview, *Neural networks*, *61*, 85–117, 2015.

Shahriari, B., K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, Taking the human out of the loop: A review of bayesian optimization, *Proceedings of the IEEE*, *104*(1), 148–175, 2016.

Silver, D., et al., Mastering the game of go without human knowledge, *Nature*, *550*(7676), 354, 2017.

Smolensky, P., Information processing in dynamical systems: Foundations of harmony theory, *Tech. rep.*, DTIC Document, 1986.

Srivastava, N., and R. R. Salakhutdinov, Multimodal learning with deep boltzmann machines, in *Advances in neural information processing systems*, pp. 2222–2230, 2012.

Sutskever, I., and T. Tieleman, On the convergence properties of contrastive divergence, in *International Conference on Artificial Intelligence and Statistics*, pp. 789–795, 2010.

Sutton, R. S., and A. G. Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

Thornton, C., F. Hutter, H. H. Hoos, and K. Leyton-Brown, Auto-weka: Combined selection and hyperparameter optimization of classification algorithms, in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 847–855, ACM, 2013.

Thrun, S., and L. Pratt, Learning to learn: Introduction and overview, in *Learning to learn*, pp. 3–17, Springer, 1998.

Tieleman, T., Training restricted boltzmann machines using approximations to the likelihood gradient, in *Proceedings of the 25th international conference on Machine learning*, pp. 1064–1071, ACM, 2008.

Tsodyks, M., and H. Markram, The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability, *Proceedings of the national academy of science USA*, *94*, 719–723, 1997.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Wang, F., and D. Landau, Efficient, multiple-range random walk algorithm to calculate the density of states, *Physical review letters*, *86*(10), 2050, 2001.

Wang, Y., H. Markram, P. H. Goodman, T. K. Berger, J. Ma, and P. S. Goldman-Rakic, Heterogeneity in the pyramidal network of the medial prefrontal cortex., *Nature neuroscience*, *9*(4), 2006.

Whittington, J. C., and R. Bogacz, An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity, *Neural computation*, *29*(5), 1229–1262, 2017.

Xu, K., J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in *International conference on machine learning*, pp. 2048–2057, 2015.

Zenke, F., and S. Ganguli, Superspike: supervised learning in multilayer spiking neural networks, *Neural computation*, *30*(6), 1514–1541, 2018.

Zucker, R. S., and W. G. Regehr, Short-term synaptic plasticity, *Annual review of physiology*, *64*(1), 355–405, 2002.

# B. Acknowledgments

I would like to express my gratitude to all who supported this work.

**Karlheinz Meier** for creating this pioneering group combining physics and neuroscience, which attracted me from more than 5000 miles (by the shortest spherical curve) away and stayed for almost 8 years. For being an enthusiastic and encouraging mentor who was always willing to help, despite an overwhelmingly busy schedule. For being a big fan of our work and promoting it with every possible opportunity. Seeds have been spread, now you may sleep in peace, the rest will carry on what is left to be done.

**Johannes Schemmel** for creating the amazing hardware platform and leading the group, for being so supportive and for all the nice advice during the final phase of my Ph.D.

**Daniel Durstewitz** for agreeing to review this thesis, for giving me the opportunity to present my work in ZI and asking sharp but inspiring questions.

**Ulrich Schwarz, Thomas Gasenzer** for agreeing to review this thesis and showing interest for the work done in our group.

**Mihai A. Petrovici** for all the amazing science you have created in this group, for your direct supervising and all the help you gave me from the start of my master all the way to my Ph.D., for your fantastic writing, insightful discussions and research spirit that have always motivated me to do better. I still remember your answer when I asked you how many hours you work per day, maybe you bluffed a bit but whatever, somehow it has become a benchmark when I judge my work. For your nice non-academical advice and 80% of jokes and 70% of music in your car. Sometimes it was hard to catch up with your pace and the process may be a pain, but what I learned always paid off. Thank you again for what you have taught me.

**Ilja Bytschok** for all the helpful discussions and nice suggestions from the start of my master thesis until Ph.D, for all the amazing table tennis and badminton we have played, for all the delightful conversations during lunch and your sense of humor (especially the feeling of calm when you joke about stuff, maybe that's the Russian sign... though I really thought you were French the first time we met.)

**Oliver Breitwieser** for all the amazing code you wrote. It is possible that without your 10-times faster (specifically to the size of the network I am using for MNIST)

SBS module I might be still struggling on those simulations and graduate 10-times slower. For your professional and countless helps on software and Linux whenever I messed up. For keep being the eldest TMA around the table during Mihai's absence, which really released my pressure.

**Dominik Dold** for all our academical discussions and amazing trips, for organizing journal clubs, for our delightful conversations not limited on anime, the roller coaster and night club dance in Tokyo.

**Andreas Baumbach** for your nice academical suggestions and frank opinions which improved the work, for your help on the writing of Agnes's paper.

**Akos Ferenc Kungl** for all our academical discussions in Bern and your helpful suggestions on Ruyi's work.

**Eric Müller** for your general support and allocation of computational resources. I really enjoyed our conversations on Chinese social and political issues, we can have more of this, but don't trust media, 百闻不如一见.

**Sebastian Schmitt** for your interest in implementing LIF-based Boltzmann machines with STP on the hardware, for your help on the translation of the abstract of this thesis.

**Agnes Korcsák-Gorzó** for your diligent work on spike-based tempering. For the amazing implementation of the noise generator with C++, for nice conversations we had during lunch.

**Ruyi Zuo** for your work on STP-based RBMs.

for all the others who offered help during my work and for all my lovely friends in Heidelberg.

and last, for my dearest yuanyuan and my parents, 爱你们

## Statement of Originality (Erklärung):

I certify that this thesis, and the research to which it refers, are the product of my own work. Any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline.

Ich versichere, daß ich diese Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, May 28, 2019

.....
(signature)