

Training Energy-Based Models with Non-MCMC Samplers and Efficient Temperature Estimation

Kentaro Kubo^{1,*} and Hayato Goto^{1,2}

¹*Corporate Laboratory, Toshiba Corporation, Kawasaki, Kanagawa 212-8582, Japan*

²*RIKEN Center for Quantum Computing (RQC), Wako, Saitama 351-0198, Japan*



(Received 6 January 2026; accepted 31 July 2026; published 17 September 2026)

Efficient sampling from Boltzmann distributions over discrete variables is a fundamental operation in a wide range of applications. While fast non-Markov chain Monte Carlo (MCMC) samplers have recently emerged as promising alternatives to conventional MCMC methods, their practical use for probabilistic learning remains hindered by the difficulty of estimating the effective temperature of the generated samples. In this work, we begin by introducing Langevin simulated bifurcation (LSB), a Boltzmann sampler that enables fast and parallel sampling with accuracy comparable to sequential MCMC methods. To address the challenge of unknown effective temperature, we propose conditional expectation matching (CEM), an efficient estimation method applicable to energy-based models (EBMs) with exploitable conditional independence structures. Building on these components, we further develop a learning framework, termed sampler adaptive learning (SAL), which adaptively adjusts the model temperature to match that of the distribution induced by fast non-MCMC sampling. We demonstrate the effectiveness of LSB, CEM, and SAL on semirestricted Boltzmann machines (SRBMs), a class of EBMs that are difficult to train using conventional approaches. LSB achieves orders-of-magnitude acceleration over Gibbs sampling while maintaining comparable or higher accuracy, and CEM enables accurate temperature estimation of the resulting distribution with negligible computational overhead. As a consequence, SAL enables efficient training of SRBMs and outperforms conventional Boltzmann machine learning methods on synthetic spin-glass datasets. In addition, the trained models achieve strong performance across multiple tasks, including image generation, reconstruction, and classification. These results establish LSB as a fast and accurate Boltzmann sampler and provide key insights that enable practical applications of fast non-MCMC sampling methods via efficient temperature estimation with CEM.

DOI: [10.1103/2vb5-2lt8](https://doi.org/10.1103/2vb5-2lt8)

I. INTRODUCTION

Sampling from Boltzmann distributions over discrete variables is a fundamental computational primitive that appears across a wide range of scientific and engineering disciplines. A particularly prominent instance arises in machine learning, where energy-based models (EBMs) rely on sampling to estimate gradients and generate data [1,2], with Boltzmann machines serving as a prominent example [3–5]. In quantum many-body physics, similar sampling tasks also arise in neural quantum states, including Boltzmann machine-based ansätze, where samples are used to evaluate observables and optimize variational wave functions [6]. More broadly, Boltzmann sampling plays a central role in a wide range of problems in statistical physics, Bayesian inference, and stochastic optimization [7–9], and its hardware implementations have also attracted attention as a potential pathway toward addressing the growing power consumption of modern AI systems [10]. Despite this broad applicability, efficient sampling from discrete Boltzmann distributions remains a major computa-

tional challenge. Conventional Markov chain Monte Carlo (MCMC) methods [11], such as Gibbs sampling [12] or the Metropolis-Hastings algorithm [13,14], are inherently sequential and often exhibit poor scalability to practical large-scale applications. Furthermore, these methods often suffer from slow mixing in high-dimensional and rugged energy landscapes, severely limiting their practical efficiency. Therefore, there is a strong demand for alternative sampling approaches that can overcome these limitations.

In this work, we first introduce a Boltzmann sampler inspired by a quantum-inspired combinatorial optimization known as simulated bifurcation [15–19]. We refer to this algorithm as Langevin simulated bifurcation (LSB). LSB enables parallel sampling while maintaining accuracy comparable to the conventional reliable method, namely, sequential MCMC. However, a key challenge remains: The inverse temperature of the output distribution, β_{eff} , is generally unknown and depends on both the problem instance and sampler parameters. Importantly, this limitation is not unique to LSB, but is a common challenge encountered in other non-MCMC samplers, such as D-Wave’s quantum annealers [20–27], coherent Ising machines [28–30], their simulated counterparts [31,32], quantum bifurcation machines [33], noise-injected optoelectronic Ising machines [34], networks of coupled optical parametric oscillators [35], and adaptive Ising machine [36]. Unless β_{eff} is accurately estimated or controlled, sampling does not correspond to the desired distribution, leading to biased estimation of expectation values and probabilistic quantities,

*Contact author: kentaro.kubo.b91@mail.toshiba

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article’s title, journal citation, and DOI.

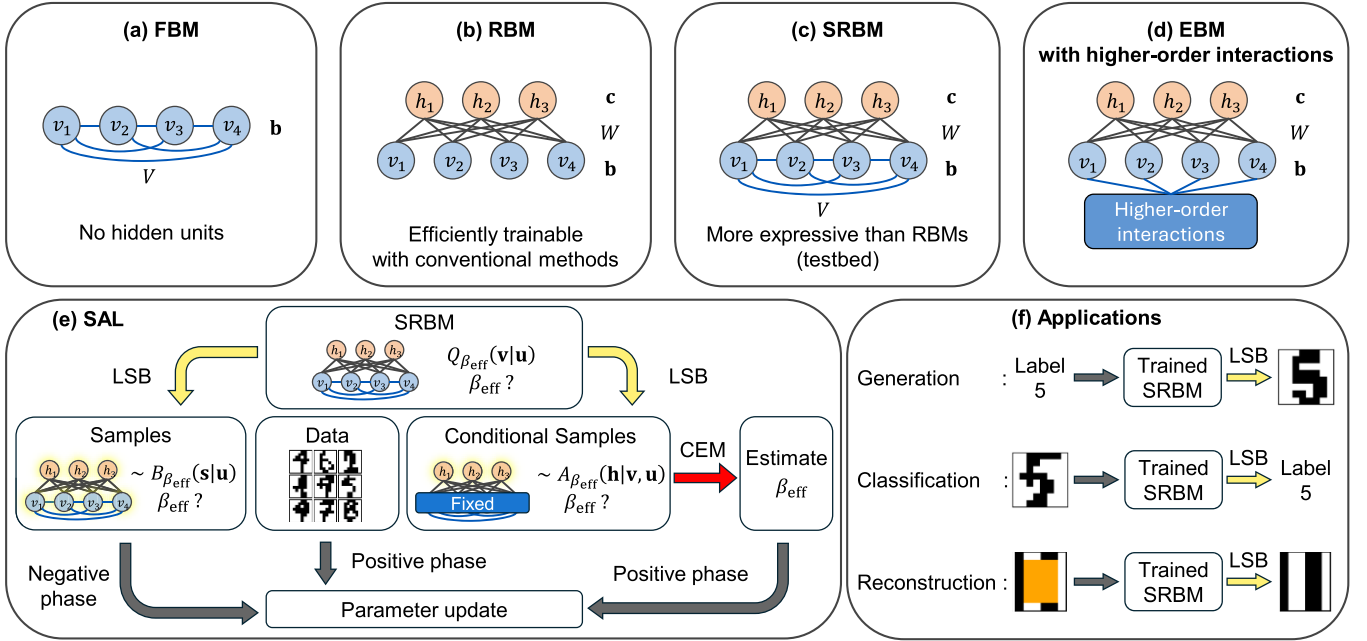


FIG. 1. Overview of model structures and proposed methods: (a)–(d) Energy-based model structures: All models consist of visible nodes v_i (blue circles, arranged in the lower row) and, if present, hidden nodes h_j (orange circles, arranged in the upper row). (a) FBM: only visible nodes are present, and all visible nodes are mutually connected. (b) RBM: Every visible node is connected to every hidden node, but there are no connections among visible nodes or among hidden nodes. (c) SRBM: obtained by adding pairwise connections among visible nodes to an RBM. (d) EBM with higher-order interactions: obtained by extending SRBMs with higher-order interactions among visible units. (e) Schematic of the SAL framework, which combines a non-MCMC sampler with conditional expectation matching (CEM, red arrow) for efficient estimation of the effective inverse temperature β_{eff} . In this work, we use LSB (yellow arrow) as the sampler and train SRBMs. (f) Examples of applications of trained SRBMs: image generation, classification, and reconstruction.

which in turn degrades downstream inference and learning tasks.

However, the standard approach for estimating β_{eff} , based on minimizing the Kullback-Leibler (KL) divergence, is computationally prohibitive within practical computational pipelines and becomes infeasible for large-scale problems [37]. Although alternative methods have been explored, they often suffer from accuracy and stability issues [32,37,38]. For the simplest class of EBMs, namely, fully visible Boltzmann machines (FBMs) [see Fig. 1(a)], learning methods have been proposed to avoid explicit estimation of β_{eff} [39]. As these models lack hidden variables, their expressive power is limited. Thus, the use of fast non-MCMC samplers, including LSB, demands a scalable and accurate β_{eff} estimation method. Here, we introduce an efficient method for estimating β_{eff} , which we term conditional expectation matching (CEM). The proposed method is applicable to EBMs in which conditional independence structures can be exploited. In particular, it is not limited to restricted Boltzmann machines (RBMs) [see Fig. 1(b)], which can be efficiently trained using conventional methods [4,5], but also extends to more expressive models such as semirestricted Boltzmann machines (SRBMs) [40–42] and their further extensions with higher-order interactions [see Figs. 1(c) and 1(d)]. CEM is naturally parallelizable with sampling processes of non-MCMC samplers and provides accurate estimation of β_{eff} . Furthermore, we propose a learning framework, termed sampler adaptive learning (SAL) [see Fig. 1(e)], which combines fast non-MCMC sampling with CEM and adaptively adjusts the temperature of the model

based on the distribution induced by the sampler. This integration enables scalable and efficient training of EBMs with conditional independence structures.

As a representative testbed, we apply LSB, CEM, and SAL to SRBMs [see Fig. 1(c)]. SRBMs extend RBMs by introducing interactions among visible units, thereby enhancing model expressiveness [40–42]. However, efficient training of such models using conventional methods is challenging, as the introduction of these additional interactions breaks the conditional independence structure of the visible units exploited by block Gibbs sampling and contrastive divergence [5]. We first demonstrate that LSB enables fast and accurate sampling from the Boltzmann distribution of SRBMs, achieving accuracy comparable to or better than Gibbs sampling while providing orders-of-magnitude speedup. We further show that the inverse temperature of the resulting distribution can be efficiently and accurately estimated using CEM. These capabilities translate into improved learning performance, and we demonstrate that SAL-trained SRBMs outperform conventional learning methods in minimizing the training cost function on synthetic spin-glass datasets [43,44]. Furthermore, we achieve good performance across multiple tasks, including high-fidelity image generation and reconstruction on the bars-and-stripes (BAS) dataset [45,46], both unconditional and conditional image generation, and high classification accuracy on OptDigits [47] [see Fig. 1(f)].

The present results provide a proof of principle for the effectiveness of combining non-MCMC sampling and accurate temperature estimation via CEM, rather than establishing

immediate competitiveness with state-of-the-art generative AI. These ideas are expected to bridge fast non-MCMC sampling methods and their practical applications across a wide range of domains, including neural quantum states, more expressive EBM, and energy-efficient hardware implementations of sampling-based learning.

II. ENERGY-BASED MODEL AND BOLTZMANN DISTRIBUTION

EBMs are expressive probabilistic models that capture complex dependencies and latent structures [1,2]. We consider EBM defined over visible units $\mathbf{v}$, which represent the observed data, and hidden units $\mathbf{h}$, which enhance the model's expressive capacity, forming an undirected model. Both $\mathbf{v}$ and $\mathbf{h}$ are binary vectors of dimensions N_v and N_h , respectively, taking values in $\{-1, +1\}$, like Ising spins. The full state of the model is denoted by an N -dimensional Ising vector $\mathbf{s} = (\mathbf{v}^T, \mathbf{h}^T)^T$, where $N = N_v + N_h$. In generative modeling, the EBM often models a target distribution $P_D(\mathbf{v})$ using a marginal Boltzmann distribution

$$Q_\beta(\mathbf{v}|\mathbf{u}) = \sum_{\mathbf{h}} B_\beta(\mathbf{s}|\mathbf{u}), \quad (1)$$

where

$$B_\beta(\mathbf{s}|\mathbf{u}) = \frac{e^{-\beta E(\mathbf{s}|\mathbf{u})}}{Z_\beta} \quad (2)$$

is a Boltzmann distribution with inverse temperature β , $Z_\beta = \sum_{\mathbf{s}} e^{-\beta E(\mathbf{s}|\mathbf{u})}$ is the partition function, $E(\mathbf{s}|\mathbf{u})$ is the global energy function, and $\mathbf{u}$ denotes the model parameter. Throughout this paper, for an Ising vector $\mathbf{S}$, the notation $\sum_{\mathbf{S}}$ denotes the summation over all configurations of $\mathbf{S}$. Learning aims to minimize the KL divergence

$$D_{\text{KL}}(P_D||Q_\beta) = \sum_{\mathbf{v}} P_D(\mathbf{v}) \ln \frac{P_D(\mathbf{v})}{Q_\beta(\mathbf{v}|\mathbf{u})} \quad (3)$$

for $\mathbf{u}$, typically via gradient-based methods. Since the exact calculation of the gradients scales exponentially with the system size, it is standard to approximate the gradients using Boltzmann sampling, most commonly via MCMC techniques, such as Gibbs sampling [12] or the Metropolis-Hastings algorithm [13,14]. However, these MCMC methods are often computationally inefficient due to their inherent sequentiality and slow mixing property, motivating the need for fast and accurate alternative sampling approaches.

Among the above EBM, those whose energy function takes the quadratic form

$$E(\mathbf{s}|\mathbf{u}) = -\frac{1}{2} \mathbf{s}^T \mathbf{J} \mathbf{s} - \mathbf{f} \cdot \mathbf{s} \quad (4)$$

are referred to, in particular, as Boltzmann machines. Here, $\mathbf{J}$ is an $(N \times N)$ weight matrix satisfying $J_{ii} = 0$ and $J_{ij} = J_{ji} \in \mathbb{R}$ for $i \neq j$, and $\mathbf{f}$ is an N -dimensional bias vector with $f_i \in \mathbb{R}$. The set of model parameters is denoted by $\mathbf{u} = (\mathbf{J}, \mathbf{f})$. For specific architectures, the parameter set $\mathbf{u}$ for Boltzmann machines can be expressed according to their structure. In particular, for fully visible, restricted, and semirestricted Boltzmann machines (FBMs, RBMs, SRBMs), the parameters can be written as $\mathbf{u} = (V, \mathbf{b})$, $(W, \mathbf{b}, \mathbf{c})$, and $(V, W, \mathbf{b}, \mathbf{c})$, respectively, where V and W denote the weight matrices between visible-visible and visible-hidden units, respectively,

and $\mathbf{b}$ and $\mathbf{c}$ are bias vectors for the visible and hidden units, respectively.

In the following section, we focus on SRBMs [40–42], as a representative testbed for our approaches (LSB, CEM, and SAL), whose energy function is given by

$$E(\mathbf{s}|\mathbf{u}) = -\frac{1}{2} \mathbf{v}^T V \mathbf{v} - \mathbf{v}^T W \mathbf{h} - \mathbf{b} \cdot \mathbf{v} - \mathbf{c} \cdot \mathbf{h}. \quad (5)$$

Because of the additional interactions among visible units, SRBMs are more expressive than the celebrated RBMs [4,5], but their efficient training remains challenging using conventional methods due to the lack of available parallel sampling. As will be shown in Sec. III, the proposed LSB overcomes this limitation by enabling parallel sampling while maintaining high accuracy. In addition, the proposed CEM allows accurate estimation of the effective inverse temperature associated with the distribution induced by the sampler, as will be demonstrated in Sec. IV. By combining these components, the SAL framework enables effective training of SRBMs, achieving performance beyond conventional Boltzmann machine learning approaches (see Sec. V). While SRBMs serve as a convenient testbed in this study, we note that the proposed framework can be applied to EBM with exploitable conditional independence structures.

III. LANGEVIN SIMULATED BIFURCATION

Several optimization machines and algorithms, such as D-Wave's quantum annealers, coherent Ising machines, their simulated counterparts, quantum bifurcation machines, noise-injected optoelectronic Ising machines, networks of coupled optical parametric oscillators, and adaptive Ising machine have been shown to function effectively as Boltzmann samplers [20–36]. Inspired by these findings, we investigated the output distributions of another class of optimization algorithms, simulated bifurcation algorithms [15–19]. Our results show that special cases of conventional simulated bifurcation algorithms approximate the Boltzmann distribution to a nontrivial degree, but their sampling accuracy is substantially lower than that achieved by MCMC methods (see Appendix A). In this section, we propose a modified sampling algorithm that achieves sampling accuracy comparable to MCMC. We refer to this algorithm as LSB, since its continuous limit resembles the celebrated continuous-variable sampler, Langevin Monte Carlo, whose stationary distribution is guaranteed to be the Boltzmann distribution [48,49].

Similarly to discrete simulated bifurcation (dsB) [16], LSB is based on the following equations of motion for positions x_i and momenta y_i corresponding to s_i in the general global energy function $E(\mathbf{s}|\mathbf{u})$:

$$\dot{y}_i = - \left. \frac{\partial E(\mathbf{x}|\mathbf{u})}{\partial x_i} \right|_{\mathbf{x}=\mathbf{s}}, \quad (6)$$

$$\dot{x}_i = y_i, \quad (7)$$

where $\mathbf{s} = \text{sgn}(\mathbf{x})$ and the dots denote time derivatives. Equations (6) and (7) are solved with the symplectic Euler method [50]:

$$y_i(k+1) = y_i(k) - \left. \frac{\partial E(\mathbf{x}(k)|\mathbf{u})}{\partial x_i} \right|_{\mathbf{x}=\mathbf{s}} \Delta, \quad (8)$$

$$x_i(k+1) = x_i(k) + y_i(k+1)\Delta, \quad (9)$$

where Δ is the time step and $k \in \{0, \dots, M\}$ denotes iteration. To improve sampling accuracy, at the end of each iteration, LSB applies discretization and stochastic initialization to x_i and y_i , respectively:

$$x_i \rightarrow \text{sgn}(x_i), \quad (10)$$

$$y_i \rightarrow \xi_i \sim \mathcal{N}(0, \sigma), \quad (11)$$

where ξ_i is a stochastic variable drawn from a Gaussian distribution $\mathcal{N}(\mu, \sigma)$ with mean $\mu = 0$ and standard deviation σ , and the notation $\sim$ indicates that the variable on the left-hand side follows the distribution on the right-hand side. Here, Δ and σ are the only hyperparameters of LSB. Note that the update rule inherits the properties of the conventional simulated bifurcation algorithm and can be implemented in parallel for each component. In this work, x_i and y_i are initialized by sampling uniformly from $\{-1, 1\}$ and from $N(0, \sigma)$ using the same σ as in the update rule, respectively, though other choices are possible in general. After M iterations of the above updates, the resulting $\mathbf{x}$ provides a sample of $\mathbf{s} = \text{sgn}(\mathbf{x})$; executing this procedure L times, we obtain a sampled Ising vector set $\{\mathbf{s}^{(l)} | l = 1, \dots, L\}$.

To assess the sampling performance of LSB, we evaluated its accuracy against conventional Gibbs sampling (MCMC) at $\beta = 1$ on random SRBMs with $N_v = 10$ and $N_h = 5$, using $M = 100$ iterations for LSB and the same M Monte Carlo steps for Gibbs sampling. Edge weight V_{ij} and W_{ij} were sampled from $\mathcal{N}(0, 2/\sqrt{N})$ with $N = N_v + N_h = 15$, and bias b_i and c_i were set to zero. Sampling accuracy was defined as the KL divergence $D_{\text{KL}}(P_S \| B_{\beta_{\text{eff}}})$, where $P_S(\mathbf{s}) = \frac{1}{L} \sum_{l=1}^L \delta(\mathbf{s} - \mathbf{s}^{(l)})$ represents the empirical distribution of the sampled Ising vectors, and $\delta(\mathbf{x}) = 1$ if $\mathbf{x} = \mathbf{0}$ and 0 otherwise. The effective inverse temperature β_{eff} was determined self-consistently by minimizing the KL divergence [37]: $\beta_{\text{eff}} = \text{argmin}_{\beta} D_{\text{KL}}(P_S \| B_{\beta})$. We initialized β at 1 and performed the minimization over $[0, \infty)$ using the L-BFGS-B method implemented in `scipy.optimize.minimize` [51].

Figure 2 shows the sampling accuracy $D_{\text{KL}}(P_S \| B_{\beta_{\text{eff}}})$ and the corresponding β_{eff} values for ten random instances. We fixed Δ at 1 and optimized σ from the candidate set $\sigma^{-2} \in \{0.5, 0.6, \dots, 1.9, 2.0\}$ to maximize LSB sampling accuracy for each instance. LSB outperformed Gibbs sampling in six out of ten instances based on the sampling accuracy metric $D_{\text{KL}}(P_S \| B_{\beta_{\text{eff}}})$. Across all ten instances, the mean $D_{\text{KL}}(P_S \| B_{\beta_{\text{eff}}})$ ($\pm$ standard error) was 0.09 ± 0.02 for Gibbs and 0.07 ± 0.01 for LSB. Taking the errors into account, LSB demonstrates performance comparable to or even better than Gibbs sampling. We use LSB for SAL in Sec. V.

We also confirmed that the same qualitative trends persist for more complex Boltzmann machines such as the Sherrington-Kirkpatrick model (see Appendix A), as well as a system size 20% larger than the original setting (see Appendix B). In addition, as discussed in Appendix B, elapsed-time measurements on CPU under the same computational resources show that LSB achieves these improvements while providing orders-of-magnitude speedup over Gibbs sampling, with memory usage of the same order of magnitude. This speed advantage reflects the intrinsically sequential nature of Gibbs updates, in contrast to the matrix-vector op-

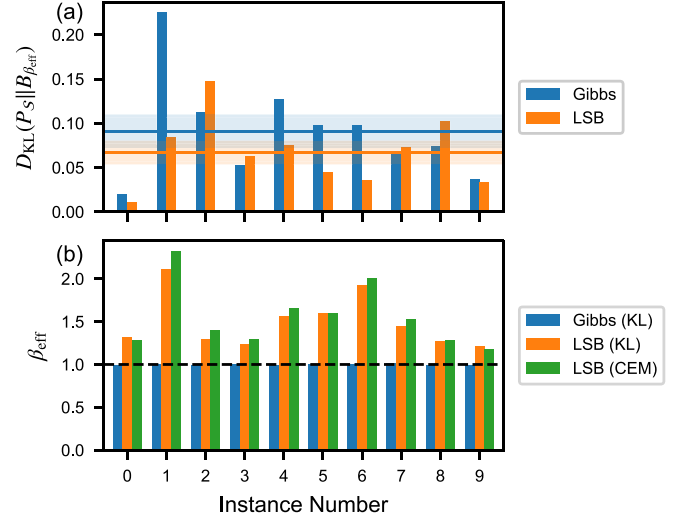


FIG. 2. Sampling accuracy and effective inverse temperature for LSB and Gibbs sampling: (a) Sampling accuracy $D_{\text{KL}}(P_S \| B_{\beta_{\text{eff}}})$ for LSB and Gibbs sampling across ten random instances of SRBM with $N_v = 10$ and $N_h = 5$. For each instance, Δ was fixed at 1 and σ was optimized from the candidate set $\sigma^{-2} \in \{0.5, 0.6, \dots, 2.0\}$ to maximize LSB sampling accuracy. Horizontal lines and shaded regions indicate the means KL divergence and their standard errors: 0.09 ± 0.02 for Gibbs and 0.07 ± 0.01 for LSB. (b) The effective inverse temperature β_{eff} for the output distribution produced by each sampler. For Gibbs sampling, β_{eff} was estimated by minimizing the KL divergence. For LSB, β_{eff} was estimated by both minimizing the KL divergence and CEM.

erations underlying LSB, which are naturally amenable to parallel execution and are therefore expected to benefit further from hardware acceleration such as field-programmable gate array (FPGA) implementations.

Furthermore, to assess the practical role of the discretization and stochastic-initialization steps introduced in LSB, we performed additional comparisons with closely related variants, as detailed in Appendix A. Although a complete theoretical explanation of the observed behavior remains open, these comparisons provide clear empirical evidence that both discretization and stochastic initialization are important for achieving improved sampling accuracy in LSB.

As noted above, LSB resembles Langevin Monte Carlo in the continuous-time limit, whose stationary distribution is known to follow a Boltzmann distribution [48,49]. Therefore, LSB is also expected to generate samples that approximately follow a Boltzmann distribution. However, for LSB, there is currently no theoretical guarantee that the generated distribution is exactly described by a Boltzmann distribution. Moreover, even when the output distribution approximates a Boltzmann distribution, its effective inverse temperature generally depends on the problem instance and sampler parameters due to discretization effects. As a result, the effective inverse temperature cannot be determined in advance. Therefore, estimating the effective temperature is essential for practical applications of LSB. In the next section, we introduce an efficient method for estimating this effective temperature.

IV. CONDITIONAL EXPECTATION MATCHING

Figure 2(b) shows that the effective inverse temperature β_{eff} estimated by minimizing the KL divergence varies across instances for LSB sampling (orange), whereas Gibbs sampling consistently yields $\beta_{\text{eff}} \simeq 1$ (blue). Estimating the unknown β_{eff} is essential for successful probabilistic learning. However, the conventional approach based on minimizing the KL divergence [37] scales exponentially with the system size N , making it intractable for practical problems. Thus, the instance-dependent variability of β_{eff} in LSB output distributions poses a significant challenge to Boltzmann machine learning with LSB. To overcome this, we propose an efficient β_{eff} estimation method: CEM, as detailed below.

In a Boltzmann distribution $B_{\beta_{\text{eff}}}(\mathbf{s}|\mathbf{u})$ of an SRBM, hidden variables are conditionally independent when the visible variables $\mathbf{v}$ are fixed to a condition vector $\mathbf{r}$. This property allows analytical expressions for their conditional expectations [5]:

$$\begin{aligned} \langle h_j \rangle_{\mathbf{r}, \beta_{\text{eff}}} &:= \sum_{\mathbf{h}} h_j A_{\beta_{\text{eff}}}(\mathbf{h}|\mathbf{v}, \mathbf{u}) \delta(\mathbf{v} - \mathbf{r}), \\ &= \tanh \left[\beta_{\text{eff}} \left(c_j + \sum_{i=1}^{N_v} r_i W_{ij} \right) \right], \end{aligned} \quad (12)$$

where

$$A_{\beta_{\text{eff}}}(\mathbf{h}|\mathbf{v}, \mathbf{u}) := \frac{B_{\beta_{\text{eff}}}(\mathbf{s}|\mathbf{u})}{\sum_{\mathbf{h}} B_{\beta_{\text{eff}}}(\mathbf{s}|\mathbf{u})} \quad (13)$$

is a conditional Boltzmann distribution. On the other hand, $\langle h_j \rangle_{\mathbf{r}, \beta_{\text{eff}}}$ can also be directly obtained by conditional LSB sampling (see Appendix C). Hereafter, this sampling-based estimation of $\langle h_j \rangle_{\mathbf{r}, \beta_{\text{eff}}}$ is denoted by $\langle h_j \rangle_{\mathbf{r}, C}$. We estimate β_{eff} by minimizing the squared difference between $\langle h_j \rangle_{\mathbf{r}, C}$ and the right-hand side of Eq. (12) across all hidden units:

$$\beta_{\text{eff}} = \text{argmin}_{\beta > 0} F_{\mathbf{r}}(\beta), \quad (14)$$

$$F_{\mathbf{r}}(\beta) := \sum_{j=1}^{N_h} \left\{ \langle h_j \rangle_{\mathbf{r}, C} - \tanh \left[\beta \left(c_j + \sum_{i=1}^{N_v} r_i W_{ij} \right) \right] \right\}^2. \quad (15)$$

The minimization is performed similarly to the above minimization of the KL divergence. This procedure, termed CEM, leverages consistency between conditionally sampled expectations $\langle h_j \rangle_{\mathbf{r}, C}$ and analytical conditional expectations expressed as a function of β_{eff} [the right-hand side of Eq. (12)]. CEM can be executed in parallel with full-system sampling to approximate $B_{\beta_{\text{eff}}}(\mathbf{s}|\mathbf{u})$, in sharp contrast to conventional β_{eff} estimators [30, 32, 37, 38]. Furthermore, the conditional sampling cost to compute $\langle h_j \rangle_{\mathbf{r}, C}$ is lower than full-system sampling. Therefore, CEM does not become a bottleneck during learning (see also Appendix B). The condition vector $\mathbf{r}$ can be randomly generated or selected from the dataset during learning, but is not restricted to these choices.

In Fig. 2(b), we compare β_{eff} estimated by CEM (green) with that obtained by minimizing the KL divergence (orange) for LSB output distributions. For CEM, each component of the condition vectors $\mathbf{r}$ was uniformly sampled from $\{-1, 1\}$. The mean signed relative error between them across ten instances is approximately 3.6%. Comparisons with other

inverse-temperature estimation methods, including modified versions of CEM (CEM- n), are provided in the Supplemental Material [52].

Thus, CEM enables accurate inverse temperature estimation. Moreover, because it is parallelizable with LSB sampling, it does not become a bottleneck in training, unlike estimation based on minimizing the KL divergence [37]. By leveraging inverse temperature estimation through CEM in learning, we overcome the computational difficulty of the positive phase (see Appendix D and Ref. [39]).

While the present demonstration is performed using LSB as the sampler, CEM is not tied to this particular choice. In principle, similar performance is expected to be possible with other non-MCMC samplers. Moreover, since CEM relies only on the conditional independence of hidden variables, it is not restricted to the SRBM setting studied here. As long as this structural property [Eq. (12)] is satisfied, CEM is expected to be applicable to EBMs beyond SRBMs. For example, one could consider extensions to models with higher-order interactions in the visible layer [see Fig. 1(d)], beyond the standard pairwise couplings in Boltzmann machines.

V. SAMPLER ADAPTIVE LEARNING

In conventional energy-based generative learning (see Sec. II), the inverse temperature β of the model distribution $Q_{\beta}(\mathbf{v}|\mathbf{u})$ is fixed, typically to 1. In contrast, we introduce a learning framework, termed SAL, in which β is set to β_{eff} , the inverse temperature of the output distribution produced by a sampler, which can depend on both the model parameters $\mathbf{u}$ and the sampling settings (see Sec. IV). As $\mathbf{u}$ is updated during learning, β_{eff} also changes. Thus, SAL involves not only updating the learning parameters $\mathbf{u}$ but also adapting the model function $Q_{\beta_{\text{eff}}}$ to the inverse temperature β_{eff} . This stands in sharp contrast to the conventional learning procedure, which updates only $\mathbf{u}$ while keeping β (and consequently the model function Q_{β}) fixed. Learning is performed by gradient descent on $D_{\text{KL}}(P_D || Q_{\beta_{\text{eff}}})$. The gradient decomposes into a negative phase, defined as an expectation under the Boltzmann distribution $B_{\beta_{\text{eff}}}(\mathbf{s}|\mathbf{u})$, and a positive phase, defined using the data distribution P_D together with β_{eff} (see Appendix D for the details). In this work, we adopt LSB as a sampler and CEM as an efficient β_{eff} estimator.

In the remainder of this section, we apply SAL to train SRBMs on three distinct datasets, to demonstrate its scalability and effectiveness. The target distributions $P_D(\mathbf{v})$ are set as the empirical distribution of each dataset $\{\mathbf{v}^{(d)} | d = 1, \dots, D\}$, where each data point $\mathbf{v}^{(d)}$ is an N_v -dimensional Ising vector: $P_D(\mathbf{v}) = \frac{1}{D} \sum_{d=1}^D \delta(\mathbf{v} - \mathbf{v}^{(d)})$. Details of each dataset and the training settings are provided in the Supplemental Material [52].

A. Learning performance on the three-spin model

We first use a dataset sampled from the exact Boltzmann distribution of a three-spin model [43, 44]. The three-spin model is an Ising system with random three-body interactions and has a highly rugged energy landscape. Consequently, the

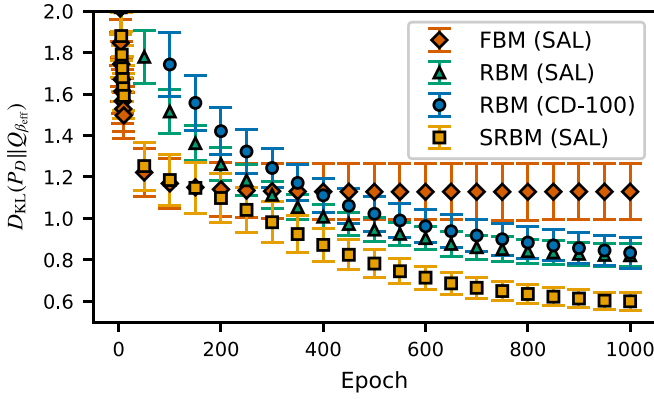


FIG. 3. Learning performance on the three-spin model: The vertical axis shows the cost function $D_{\text{KL}}(P_D||Q_\beta)$, and the horizontal axis indicates training epochs. Four models are compared: FBMs, RBMs, and SRBMs trained using SAL; RBMs with $\beta = 1$ trained using CD-100. The number of hidden variables N_h for both RBMs and SRBMs was set to 5. Each point represents the mean over ten independent instances, and error bars denote the standard error. Lower $D_{\text{KL}}(P_D||Q_\beta)$ values indicate closer agreement between Q_β and P_D .

training data, drawn from its Boltzmann distribution, exhibit a nontrivial overlap distribution characterizing spin-glass systems (see the Supplemental Material [52]). To allow exact computation of the cost function $D_{\text{KL}}(P_D||Q_\beta)$ and enable quantitative comparison with conventional methods, we restricted N_v to 10. We prepared ten independent instances to ensure statistically reliable performance evaluation.

Figure 3 compares the learning performance of four approaches: FBMs, RBMs, and SRBMs trained with SAL (see Appendix D), and RBMs with $\beta = 1$ trained via contrastive divergence-100 (CD-100) [5]. Here, CD-100 means that the initial state is set to one of the data vectors and that 100 Monte Carlo steps of blocked Gibbs sampling (MCMC) are performed. The choice of 100 was made to match LSB sampling steps, $M = 100$ (see the Supplemental Material [52] for details of parameter setting). FBMs are fully connected Boltzmann machines without hidden variables ($N_h = 0$) [see Fig. 1(a)]. The number of hidden variables N_h for both RBMs and SRBMs [see Figs. 1(b) and 1(c)] was set to 5. All the four approaches allow parallel sampling.

Among the four models, FBMs exhibited the poorest performance [highest $D_{\text{KL}}(P_D||Q_\beta)$]. This is natural because the three-spin model includes nonpairwise (three-body) interactions that cannot be captured by FBMs, which rely solely on

pairwise couplings. Introducing hidden variables, RBMs enabled the representation of higher-order correlations, leading to superior performance compared to FBMs. The observation that RBMs trained using SAL exhibit performance comparable to, or even surpassing, RBMs trained with CD-100 within the standard error suggests that SAL constitutes an effective learning strategy. Furthermore, SRBMs consistently achieved lower values of the cost function than RBMs across epochs. The difference between SRBMs and RBMs remained statistically significant even when considering the standard error, confirming the robustness of SRBM's advantage over RBMs. We also confirmed that SRBMs trained by replacing CEM in SAL with other computationally light but inaccurate inverse temperature estimation methods (maximum log-pseudo-likelihood estimation [37,38]) or by using contrastive divergence with damped mean-field iterations [40] resulted in significantly lower performance compared to SRBMs trained with SAL (see the Supplemental Material [52]). This suggests that learning SRBMs, enabled by SAL, provides a distinct advantage, particularly for modeling highly rugged energy landscapes as exemplified by spin-glass systems.

B. Generative and reconstruction performance on the BAS dataset

We next consider the synthetic BAS dataset, consisting of binary images with horizontal and vertical stripe patterns (see Refs. [45,46]). We constructed a 7×6 BAS dataset with 192 distinct patterns (all possible patterns), split into $D = 96$ training and $D_{\text{Test}} = 96$ test samples [52]. While exact computation of $D_{\text{KL}}(P_D||Q_\beta)$ was possible for the three-spin model with $N_v = 10$, it becomes challenging for the 7×6 BAS dataset ($N_v = 42$) due to the exponential growth of the state space. Therefore, here we focus on demonstrating the generative and reconstruction capabilities of SRBMs ($N_h = 21$) trained with SAL on the 7×6 BAS dataset, without computing $D_{\text{KL}}(P_D||Q_\beta)$. Training settings are detailed in the Supplemental Material [52].

The generative capabilities are shown in Fig. 4. The SRBM was trained via SAL on $D = 96$ BAS patterns [panel (a) shows randomly selected 36 ones], and at various training epochs, 36 samples were generated by LSB from the trained SRBM [panels (b)–(f)]. As training progressed, the SRBM rapidly captured the essential features of both horizontal and vertical stripes, and at 6000 epochs [panel (f)] generated only valid BAS patterns without misgeneration.

Reconstruction experiment results shown in Fig. 5 further demonstrate the robustness of SAL. Here, we masked a

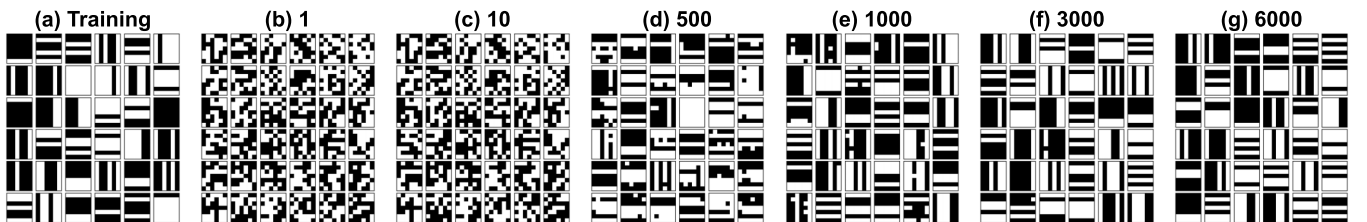


FIG. 4. Generative performance on the BAS dataset: (a) 36 randomly selected training samples. (b)–(g) 36 samples generated by the trained SRBM with LSB after 1, 10, 500, 1000, 3000, and 6000 training epochs, respectively.

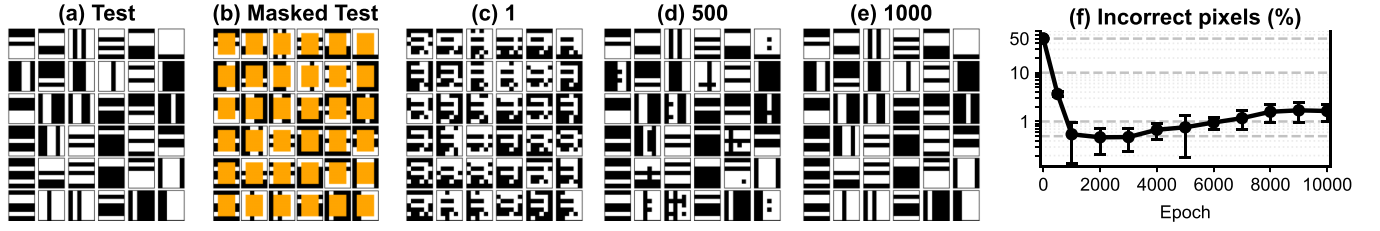


FIG. 5. Reconstruction performance on the BAS dataset: (a) 36 randomly selected test samples. (b) The same samples with a central 5×4 block masked (47.6% of pixels missing, shown in orange). (c)–(e) Reconstructions of panel (b) using the trained SRBM at 1, 500, and 1000 epochs, respectively, obtained by conditional LSB sampling. (f) The ratio of incorrect pixels per reconstruction as a function of training epoch. Each data point represents the mean over ten independent runs with different random initializations of $\mathbf{u}$; error bars indicate the standard deviation.

central 5×4 block of each test image [Fig. 5(a)], resulting in a collapsed image with 47.6% of pixels missing (shown in orange) [Fig. 5(b)]. By fixing the visible variables of the SRBM corresponding to the intact pixels in panel (b) to their observed values, we inferred the masked pixel values via conditional sampling [see Fig. 1(f) and Appendix C]. After 1000 epochs of learning, almost all patterns were successfully reconstructed by leveraging the learned distribution [Fig. 5(e)]. Figure 5(f) shows the fraction of incorrect pixels per reconstruction, evaluated across all $D_{\text{Test}} = 96$ test samples. This fraction rapidly dropped from 50.0% (corresponding to random restoration) at 1 epoch to 0.5% at 1000–3000 epochs, indicating near-perfect reconstruction across all test samples. As training progressed beyond 3000 epochs, the fraction gradually increased to around 2%, suggesting potential overfitting with excessive training. Importantly, these results demonstrate that the SRBM trained with SAL learns the underlying distribution rather than memorizing individual samples, enabling robust and generalizable reconstruction even under substantial pixel loss.

C. Generative and classification performance on the OptDigits dataset

The OptDigits dataset consists of 8×8 images of handwritten digits and their class labels [47]. We flattened and binarized the images, and represented the data class using one-hot encoding, resulting in $D = 3823$ training samples and $D_{\text{Test}} = 1797$ test samples with $N_v = 74$ visible variables [52]. Compared to the three-spin model and BAS dataset, OptDigits has approximately 7.4-fold and 1.8-fold larger N_v , respectively, providing a more challenging benchmark. In the following, we demonstrate that the SRBM trained with SAL

on OptDigits captures the data distribution for generative modeling and enables accurate image classification using the learned representation. Training settings are detailed in the Supplemental Material [52].

The generative performance is shown in Fig. 6. The SRBM was trained via SAL on $D = 3823$ digit images [panel (a) shows 64 randomly selected samples], and at various training epochs, 64 samples were generated by LSB sampling from the trained SRBM [panels (b)–(e)]. While samples at 1 epoch [panel (b)] exhibited little recognizable structure, subsequent epochs revealed a progressive emergence of digit-specific patterns consistent with the training data [panels (c)–(e)]. By 10 epochs [panel (c)], the generated images began to resemble the target digits, and by 500 and 3000 epochs [panels (d) and (e)], most samples captured the essential characteristics of handwritten digits and were visually recognizable.

We further evaluated the ability to perform conditional generation by fixing the class label $\mathbf{C}$ to the desired value and sampling the image variables $\mathbf{I}$ using LSB [see Fig. 1(f), Appendix C, and the Supplemental Material [52]]. Figure 7 illustrates the results. For each $n \in \{0, 1, \dots, 9\}$, panel (a- n) shows 16 randomly selected test images of digit n , and panel (b- n) shows 16 samples generated from the SRBM after 3000 training epochs by conditional LSB sampling with the digit class n . Most of the generated images exhibited clear digit structures consistent with the specified condition, demonstrating that the SRBM successfully reproduced the overall data distribution and enabled class-conditional sample generation.

Finally, we evaluated the capability of the learned SRBM to perform image classification. For each test sample $d \in \{1, \dots, D_{\text{Test}}\}$, the visible variables corresponding to the image $\mathbf{I}$ were fixed to their observed values $\mathbf{I}^{(d)}$, and conditional

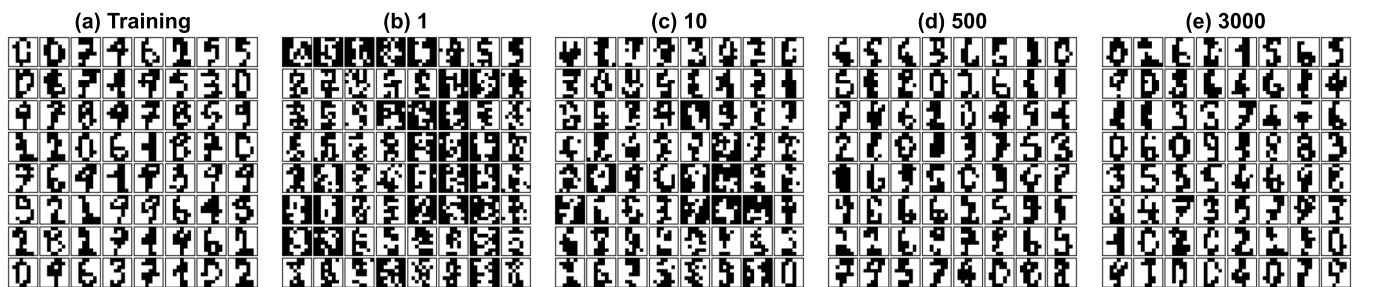


FIG. 6. Generative performance on the OptDigits dataset: (a) 64 randomly selected training samples. (b)–(e) 64 samples generated by the trained SRBM with LSB at 1, 10, 500, and 3000 training epochs, respectively.

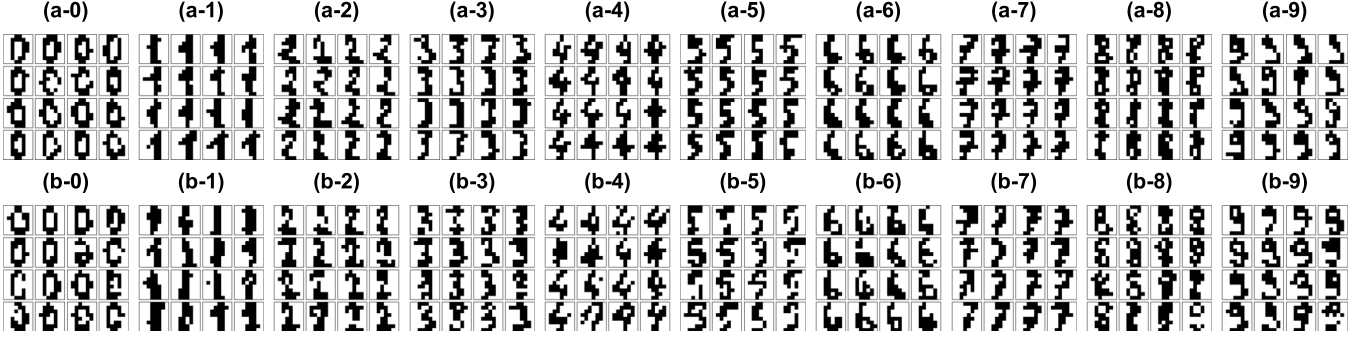


FIG. 7. Class-specified generation on the OptDigits dataset: (a- n) 16 randomly selected test samples for digit $n \in \{0, 1, \dots, 8, 9\}$. (b- n) 16 samples generated by conditional LSB sampling from the trained SRBM at 3000 epochs, conditioned on class label $n \in \{0, 1, \dots, 8, 9\}$.

LSB sampling of $\mathbf{C}$ was performed to infer the class label $\mathbf{C}^{(d)}$ (see Appendix C and the Supplemental Material [52]). The average label vector computed from conditionally sampled labels for each image d was transformed into a one-hot representation $\tilde{\mathbf{C}}$ by setting +1 for the component with the highest value and -1 for all others. Classification was deemed correct when this one-hot representation $\tilde{\mathbf{C}}$ matched the true class label $\mathbf{C}^{(d)}$. Figure 8 shows that the classification accuracy improved substantially with training, rapidly reaching nearly 90% at 500 epochs from nearly 10% (corresponding to random guessing in a ten-class problem) before training, underscoring the effectiveness of the proposed learning approach.

VI. SUMMARY AND OUTLOOK

In summary, we have proposed a fast and accurate non-MCMC Boltzmann sampler, LSB, with highly parallelizable dynamics. We have also proposed CEM, a method that resolves the inverse-temperature estimation problem inherent to fast non-MCMC samplers, not limited to LSB, when the underlying EBM exhibits suitable conditional independence

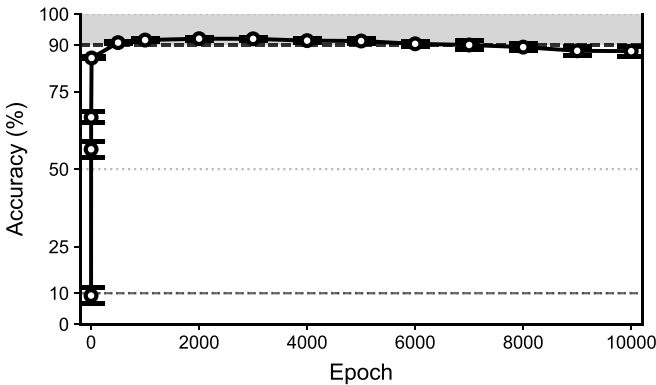


FIG. 8. Classification accuracy on the OptDigits dataset: The vertical axis shows the classification accuracy (%), and the horizontal axis indicates training epochs. Each data point represents the mean over ten independent runs with different random initializations of $\mathbf{u}$; error bars indicate the standard deviation. The shaded region marks accuracy above 90%, and the horizontal dashed lines indicate the 10% baseline (random guessing) and the 90% reference line.

structure. By combining these components, we have also developed SAL, a framework that enables generative learning with non-MCMC samplers by adaptively matching the model temperature to that of the output distribution of a sampler. We demonstrated the effectiveness of the proposed methods (LSB, CEM, and SAL) on SRBMs, a class of more expressive models that have been difficult to train using conventional methods due to the challenges in both sampling and learning.

Although SRBMs serve as a representative testbed, the proposed methods are not restricted to this specific model. We have shown in Appendix A that LSB exhibits favorable sampling performance also on the Sherrington-Kirkpatrick model, suggesting that efficient sampling is possible for more complex EBMs. Furthermore, CEM and SAL rely only on structural properties such as conditional independence, and are therefore applicable to EBMs with exploitable conditional independence structures. Exploring these applications and demonstrating practical implementations constitute important directions for future work. It is also of interest to investigate applications of non-MCMC sampling combined with CEM to neural quantum states [6], where EBMs play a central role in representing complex quantum many-body systems.

Finally, we discuss several important remarks and open questions regarding LSB. As mentioned, the continuous limit of LSB resembles Langevin Monte Carlo [48,49], whose output distribution is guaranteed to be the Boltzmann distribution. However, the theoretical basis for why LSB can well approximate the discrete Boltzmann distribution remains unclear, representing an important avenue for future research. Furthermore, developing systematic or automated strategies for hyperparameter selection in LSB is a topic that warrants further investigation. While this work has mainly addressed the parallelizability of samplers, elucidating their relaxation properties is also an important subject for future research.

ACKNOWLEDGMENTS

K.K. conceived and developed LSB, CEM, and SAL; designed and performed all numerical experiments; analyzed the data; and drafted the manuscript. H.G. suggested the exploration of SB-based sampling; discussed the results with K.K.; improved the manuscript; and supervised the project.

K.K. and H.G. are inventors in Japanese patent applications related to this work, filed by Toshiba Corporation on 2 October 2025 under application numbers P2025-166272 and P2025-166278.

The authors declare no other competing interests.

DATA AVAILABILITY

Some of the data supporting the findings of this study are openly available [47]. The datasets and code generated and/or analyzed during this study are not publicly available because of company policy and intellectual property restrictions. They may, however, be made available by the corresponding author upon reasonable request and with the appropriate company approval.

APPENDIX A: SAMPLING PERFORMANCE ON SK MODEL

In this section, we study the sampling performance on the Sherrington-Kirkpatrick (SK) model:

$$E_{\text{SK}}(\mathbf{s}) = - \sum_{i=1}^N \sum_{j>i}^{N_v} J_{ij} s_i s_j, \quad (\text{A1})$$

$$J_{ij} \sim \mathcal{N}\left(0, \frac{\zeta}{\sqrt{N}}\right) \quad \text{for } 1 \leq i < j \leq N, \quad (\text{A2})$$

where $E_{\text{SK}}(\mathbf{s})$ denotes the global energy function of the SK model (the two-spin model) [43,44], and J_{ij} represents the pairwise interaction coefficients drawn from a Gaussian distribution $\mathcal{N}(0, \zeta/\sqrt{N})$ with mean 0 and standard deviation $\zeta/\sqrt{N}$. The SK model is a fully connected spin-glass system, making the sampling task substantially more complex than the SRBM case discussed in Sec. III. We executed sampling tasks for ten random instances, each characterized by a distinct set of J_{ij} values. The parameter ζ was set to a sufficiently

large value of 2 to yield a nontrivial spin-glass phase at $\beta = 1$ [43,44]. We sampled $D = 9600$ Ising vectors $\mathbf{s}$ with $N = 15$ using various sampling methods.

Figure 9(a) indicates that the sampling accuracy of LSB and that of Gibbs sampling with $\beta = 1$ are comparable within the standard error, similarly to the results in Sec. III. Here, cLSB refers to a variant of LSB in which the discretization procedure [Eq. (10)] is replaced by a clipping rule of bSB and dSB [16]:

$$x_i \rightarrow \begin{cases} 1 & \text{for } x_i \geq 1, \\ x_i & \text{for } -1 < x_i < 1, \\ -1 & \text{for } x_i \leq -1. \end{cases} \quad (\text{A3})$$

Despite the relatively high sampling accuracy of cLSB, LSB achieves better sampling performance for all instances. This observation indicates that applying discretization at each iteration plays a key role in enhancing the sampling accuracy.

Figure 9(b) shows the inverse temperature, obtained by minimizing the KL divergence, corresponding to sampling results of panel (a). It was confirmed that the inverse temperature of the output distributions from LSB and cLSB also depends on the instance, similarly to what was observed in Sec. III.

Figure 9(c) shows the sampling accuracy of conventional simulated bifurcation algorithms: aSB, bSB, and dSB [15,16], which, unlike LSB, do not incorporate stochastic initialization or discretization applied at each iteration. Although we tuned the parameters to achieve the best possible accuracy (see the caption of Fig. 9 for details of parameters), the sampling accuracy of these conventional simulated bifurcations are clearly inferior to that of Gibbs sampling, LSB, and cLSB. Among them, bSB exhibits the poorest performance [highest $D_{\text{KL}}(P_S||B_{\beta_{\text{eff}}})$]. Although not shown in the figure, we have found that using variants, such as HbSB and HdSB [17], which are modified versions of bSB and dSB, can improve the

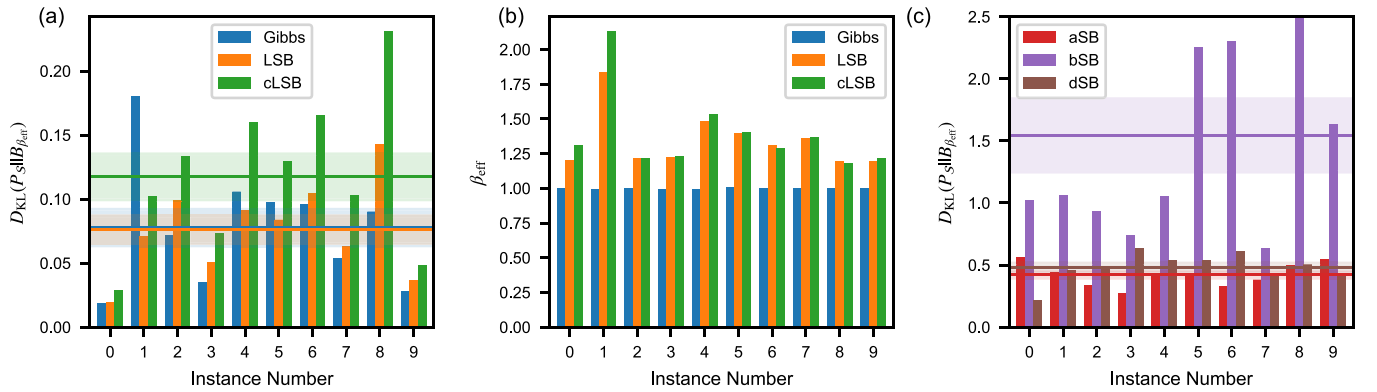


FIG. 9. Sampling accuracy $D_{\text{KL}}(P_S||B_{\beta_{\text{eff}}})$ and effective temperature β_{eff} across ten SK model instances for various sampling methods: The horizontal axis of all panels indicates the instance number. Panel (a) shows $D_{\text{KL}}(P_S||B_{\beta_{\text{eff}}})$ for LSB, Gibbs, and cLSB. Panel (b) shows β_{eff} estimated by minimizing the KL divergence corresponding to panel (a). Panel (c) shows $D_{\text{KL}}(P_S||B_{\beta_{\text{eff}}})$ for aSB, bSB, and dSB. In panels (a) and (c), horizontal lines and shaded regions indicate the means $D_{\text{KL}}(P_S||B_{\beta_{\text{eff}}})$ and their standard errors. Parameter settings in these sampling experiments were as follows: The number of iteration steps for simulated bifurcation related methods and the number of Monte Carlo steps for Gibbs sampling were both set to $M = 100$. For LSB and cLSB, the discretization step size Δ was set to 1, and σ was optimized from the candidate set $\sigma^{-2} \in \{0.5, 0.6, \dots, 1.9, 2.0\}$ to maximize sampling accuracy. For aSB, bSB, and dSB, parameters were set according to the notation in Ref. [16]: $dt = 1.0$, $a_0 = 1.0$, $a(t) = 0$ for all methods, and c_0 was set to 1.0 for aSB, $\sqrt{\sum_{j=1}^N x_j^2 / [\sum_{i=1}^N (\sum_{j=1}^N J_{ij} x_j)^2]}$ for bSB, and $\sqrt{N / \{\sum_{i=1}^N [\sum_{j=1}^N J_{ij} \text{sgn}(x_j)]^2\}}$ for dSB.

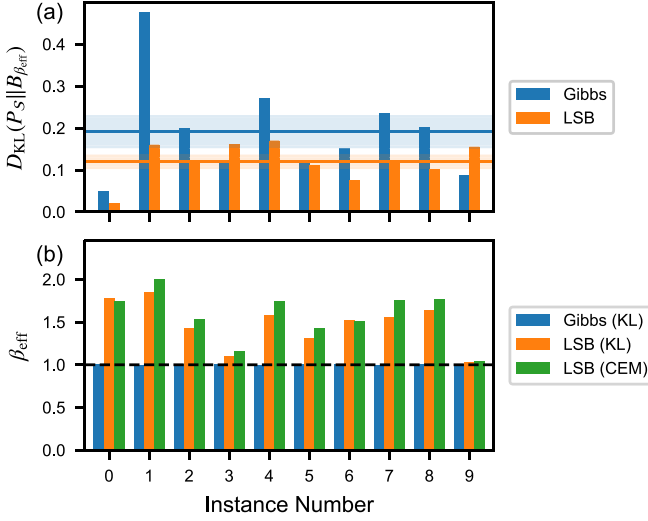


FIG. 10. Sampling accuracy and effective inverse temperature for LSB and Gibbs sampling: (a) Sampling accuracy measured by the KL divergence $D_{\text{KL}}(P_S \| B_{\beta_{\text{eff}}})$ for LSB and Gibbs sampling across ten random instances of an SRBM with $N_v = 12$ and $N_h = 6$. Across the ten instances, LSB achieved lower $D_{\text{KL}}(P_S \| B_{\beta_{\text{eff}}})$ values than Gibbs sampling in 8 out of 10 cases. The mean $D_{\text{KL}}(P_S \| B_{\beta_{\text{eff}}})$ ($\pm$ standard error) was 0.19 ± 0.04 for Gibbs sampling and 0.12 ± 0.01 for LSB. (b) Corresponding effective inverse temperature β_{eff} of the output distributions produced by each sampler. The mean signed relative error between the β_{eff} values estimated by CEM and those obtained by minimizing the KL divergence for the output distributions produced by LSB is approximately 5.7%.

sampling accuracy. However, even with these improvements, the performance has not surpassed that of LSB and cLSB. These results indicate that the proposed enhancement strategy, namely, the application of discretization and stochastic initialization at each iteration, plays a meaningful role in improving the sampling accuracy.

APPENDIX B: ADDITIONAL COMPUTATIONAL CHARACTERISTICS OF LSB AND CEM

In this section, we present additional analyses to further characterize the computational properties of LSB and CEM using the same benchmark problems as in Secs. III and IV. These analyses complement the results presented in the main text and provide further insight into the practical characteristics of the proposed sampling approach (LSB) and inverse-temperature estimation method (CEM).

1. Results for larger systems

To assess whether the qualitative behavior of LSB and CEM observed in Secs. III and IV persists when the system size is moderately increased, we performed additional simulations for larger SRBMs with $N_v = 12$ and $N_h = 6$, corresponding to a 20% increase over the original setting. All other aspects of the experimental setup were kept the same as in Secs. III and IV.

The results, shown in Fig. 10, indicate that the same qualitative trends observed in the smaller systems are also present in this moderately larger setting. In particular, the

relative sampling performance of LSB and the behavior of the effective inverse temperature remain consistent with those reported in main text. While this analysis does not constitute a systematic study of system-size scaling, it provides empirical evidence that the observed behavior is not specific to the smallest problem instances considered.

2. Computational cost

We evaluate the computational cost of LSB by comparing its run-time with that of conventional Gibbs sampling under a fair CPU-based setting. Timing measurements were performed on the original problem instances considered in Secs. III and IV using a system equipped with AMD EPYC 7742 processors. For the run-time comparison, we used 16 CPU cores to generate 16 samples, and results were averaged over ten independent instances. LSB sampling was implemented using NumPy-based matrix-vector operations. In contrast, Gibbs sampling was parallelized using Python multiprocessing, with 16 independent Markov chains assigned to individual CPU cores.

Under these conditions, the measured elapsed time was 1.79 ± 0.10 ms for LSB and 75.74 ± 5.58 ms for Gibbs sampling, indicating a substantially longer run-time for Gibbs sampling. This difference is consistent with the intrinsic update structures of the two methods: Gibbs sampling requires sequential conditional updates, whereas LSB updates all N variables simultaneously via matrix-vector operations that are naturally amenable to parallel execution.

In learning scenarios, LSB may additionally require inverse-temperature estimation via CEM when the effective output temperature of the sampler needs to be estimated. In practice, the dominant computational cost of CEM arises from the conditional sampling steps, which require running LSB on a reduced problem of size $N_h < N$. Accordingly, the run-time associated with these conditional sampling steps is bounded by that of LSB sampling itself. In the present experiment with 16 samples, the elapsed time required for the conditional sampling within CEM was 0.89 ± 0.02 ms. Consequently, even when LSB sampling and CEM are executed sequentially, the total run-time remains substantially smaller than that of Gibbs sampling. Furthermore, the inverse-temperature estimation procedure in CEM is naturally parallelizable with LSB sampling, since both rely on independent sampling operations. Therefore, when sufficient computational resources are available, the total elapsed time of LSB including CEM is upper bounded by the slower of the two components, which in typical settings is LSB itself.

Finally, we briefly comment on the memory cost. In the present implementation, LSB and standard Gibbs sampling require comparable memory usage for sampling itself, as both methods store the same set of model parameters and state variables. When inverse-temperature estimation via CEM is incorporated, additional memory is required to perform LSB sampling within CEM. However, this additional cost remains limited, since the sampling involved in CEM is applied only to the hidden variables. As a result, even when LSB sampling and CEM are combined in parallel, the total memory cost remains within a small constant factor (of less than approximately twice) of that of LSB or Gibbs sampling alone.

APPENDIX C: CONDITIONAL SAMPLING

In CEM and applications, such as image reconstruction, conditional generation, and classification [see Fig. 1(f)], conditional sampling, which fixes a subset of variables while sampling the others, is often required. This is relatively straightforward with sequential MCMC methods; however, its implementation in non-MCMC samplers that perform parallel updates requires careful consideration. To achieve conditional sampling, Ref. [39] employed approaches that impose strong biases on the fixed variables in a D -Wave's quantum annealer. However, because both the inverse temperature and sampling accuracy depend strongly on $\mathbf{u}$ in LSB sampling, such methods raise concerns about potential changes in output characteristics when applied to LSB-based sampling. Therefore, we adopt an alternative strategy as described below.

For simplicity, we consider the SRBM setting where a subset of visible variables $\mathbf{v}_f$ is fixed. This formulation covers all cases of conditional sampling in this study, and fixing more general variables, including hidden ones, can be formulated similarly. In this scenario, the energy of SRBMs, up to an additive constant, takes the following form:

$$E(\mathbf{v}_v, \mathbf{h}|\mathbf{v}_f, \mathbf{u}) = -\frac{1}{2}\mathbf{v}_v^T V_{vv} \mathbf{v}_v - \mathbf{v}_v^T W_v \mathbf{h} - (\mathbf{b}_v^T + \mathbf{v}_f^T V_{fv}) \mathbf{v}_v - (\mathbf{c}^T + \mathbf{v}_f^T W_f) \mathbf{h}. \quad (\text{C1})$$

Here, $\mathbf{v}_v$ denotes the unfixed visible variables, $V_{\mu\rho}$ ($W_{\mu\rho}$) is a weight matrix between $\mathbf{v}_\mu$ and $\mathbf{v}_\rho$ (between $\mathbf{v}_\mu$ and $\mathbf{h}$) for $\mu, \rho \in \{v, f\}$, and $\mathbf{b}_v$ is a bias vector of $\mathbf{v}_v$. The energy $E(\mathbf{v}_v, \mathbf{h}|\mathbf{v}_f, \mathbf{u})$ can be regarded as that of a conditional Boltzmann machine, $E(\mathbf{s}'|\mathbf{u}')$, defined over $\mathbf{s}' = (\mathbf{v}_v^T, \mathbf{h}^T)^T$ with parameter set $\mathbf{u}' = (J', \mathbf{f}')$. Here, $\mathbf{f}' = (\mathbf{b}^T, \mathbf{c}'^T)^T$, $\mathbf{b}' = \mathbf{b}_v + (\mathbf{v}_f^T V_{fv})^T$, $\mathbf{c}' = \mathbf{c} + (\mathbf{v}_f^T W_f)^T$, and

$$J' = \begin{pmatrix} V_{vv} & W_v \\ W_v^T & O_h \end{pmatrix}, \quad (\text{C2})$$

where O_h is an $(N_h \times N_h)$ -dimensional zero matrix.

By executing LSB sampling for $E(\mathbf{s}'|\mathbf{u}')$ using the same hyperparameters as those used for sampling $B_{\beta_{\text{eff}}}(\mathbf{s}|\mathbf{u})$, we obtain a conditionally sampled Ising vector set $\{\mathbf{s}'^{(l)}|l=1, \dots, L'\}$. For the sake of simplicity in this study, we assumed $L' = L$. We expect the empirical distribution $P_{S'}(\mathbf{s}') = \frac{1}{L'} \sum_{l'=1}^{L'} \delta(\mathbf{s}' - \mathbf{s}'^{(l')})$ to approximate the Boltzmann distribution of the conditional Boltzmann machine $B_{\beta_{\text{eff}}}(\mathbf{s}'|\mathbf{u}')$, which shares the same inverse temperature β_{eff} as $B_{\beta_{\text{eff}}}(\mathbf{s}|\mathbf{u})$. This conjecture is supported by numerical results obtained under the same conditions as Fig. 2(b), in which we confirmed that $D_{\text{KL}}[P_{S'}(\mathbf{s}')||B_{\beta_{\text{eff}}}(\mathbf{s}'|\mathbf{u}')] \ll D_{\text{KL}}[P_S(\mathbf{s})||B_{\beta_{\text{eff}}}(\mathbf{s}|\mathbf{u})]$.

Note that $B_{\beta_{\text{eff}}}(\mathbf{s}'|\mathbf{u}')$ corresponds to the conditional Boltzmann distribution of the original Boltzmann machine:

$$A_{\beta_{\text{eff}}}(\mathbf{s}'|\mathbf{v}_f, \mathbf{u}) = \frac{B_{\beta_{\text{eff}}}(\mathbf{s}|\mathbf{u})}{\sum_{\mathbf{v}_v} \sum_{\mathbf{h}} B_{\beta_{\text{eff}}}(\mathbf{s}|\mathbf{u})}. \quad (\text{C3})$$

In CEM, we set $\mathbf{v}_f$ to $\mathbf{v}$ and $\mathbf{s}'$ to $\mathbf{h}$. We then compute sampling-based estimation of $\langle h_j \rangle_{\mathbf{r}, \beta_{\text{eff}}}$,

$$\langle h_j \rangle_{\mathbf{r}, \mathbf{C}} := \sum_{\mathbf{h}} h_j P_{S'}(\mathbf{h}), \quad (\text{C4})$$

by conditional LSB sampling. This is compared to the analytical expression of $\langle h_j \rangle_{\mathbf{r}, \beta_{\text{eff}}}$ [Eq. (12)] to estimate β_{eff} (see Sec. IV).

In image restoration, conditional generation, and classification, $\mathbf{v}_f$ are set to the intact image pixels, the target class label $\mathbf{C}$, and the image $\mathbf{I}$ of the test data, respectively.

APPENDIX D: LEARNING RULE OF SAL

In energy-based generative modeling, the goal is to minimize the KL divergence between the target distribution P_D and Q_β defined in Eq. (3), typically using gradient-based methods. In SAL, where the inverse temperature β is set to β_{eff} , the inverse temperature of the output distribution produced by the sampler, the parameters of SRBMs are updated as follows:

$$V_{ij}(k+1) = V_{ij}(k) - \eta [\langle v_i v_j \rangle_D - \langle v_i v_j \rangle_{\beta_{\text{eff}}}], \quad (\text{D1})$$

$$W_{ij}(k+1) = W_{ij}(k) - \eta \left\{ \left\langle v_i \tanh \left[\beta_{\text{eff}} \left(c_j + \sum_{i=1}^{N_v} v_i W_{ij} \right) \right] \right\rangle_D - \langle v_i h_j \rangle_{\beta_{\text{eff}}} \right\}, \quad (\text{D2})$$

$$b_i(k+1) = b_i(k) - \eta [\langle v_i \rangle_D - \langle v_i \rangle_{\beta_{\text{eff}}}], \quad (\text{D3})$$

$$c_j(k+1) = c_j(k) - \eta \left\{ \left\langle \tanh \left[\beta_{\text{eff}} \left(c_j + \sum_{i=1}^{N_v} v_i W_{ij} \right) \right] \right\rangle_D - \langle h_j \rangle_{\beta_{\text{eff}}} \right\}, \quad (\text{D4})$$

where k denotes the learning step, $\eta > 0$ is the β_{eff} -renormalized learning rate [39], $\langle \bullet \rangle_D$ indicates the expectation value under the data distribution $P_D(\mathbf{v})$, and $\langle \bullet \rangle_{\beta_{\text{eff}}}$ indicates the expectation value under the Boltzmann distribution $B_{\beta_{\text{eff}}}(\mathbf{s})$ with inverse temperature β_{eff} . The terms defined by $\langle \bullet \rangle_D$ are referred to as the positive phase, whereas those defined by $\langle \bullet \rangle_{\beta_{\text{eff}}}$ are referred to as the negative phase. These update rules assume that variations in β_{eff} are small compared to changes

in $\mathbf{u}$. The negative phase is computed using LSB as follows:

$$\langle \bullet \rangle_{\beta_{\text{eff}}} \simeq \sum_{l=1}^L P_S(\mathbf{s}) \bullet, \quad (\text{D5})$$

where $P_S(\mathbf{s}) = \frac{1}{L} \sum_{l=1}^L \delta(\mathbf{s} - \mathbf{s}^{(l)})$ represents the empirical distribution of Ising vectors $\{\mathbf{s}^{(l)}|l=1, \dots, L\}$ sampled by LSB. Thus, the negative phase does not require explicit infor-

mation about β_{eff} . On the other hand, computing the positive phase associated with the hidden units [Eqs. (D2) and (D4)] requires the effective inverse temperature β_{eff} , which is estimated using CEM.

Setting the initial values of V_{ij} to zero and updating only W_{ij} , b_i , and c_j reduces the update rule to that of an RBM. Similarly, setting $N_h = 0$ and updating only V_{ij} and b_i reduces the update rule to that of an FBM. In the FBM case, only Eqs. (D1) and (D3) remain, and β_{eff} estimation is no longer required. The FBM version of SAL has already been reported as the gray-box algorithm [39]. That study reported that when hidden variables are present,

computing the positive phase becomes difficult, and consequently they restricted their approach to FBMs, which can be trained without explicit knowledge of β_{eff} . SAL overcomes this challenge by leveraging SRBM's (and RBM's) structure together with CEM for efficient β_{eff} estimation, enabling scalable learning without sacrificing expressive power.

We note that the same learning principle is applicable to EBMs with higher-order interactions in the visible units, yielding similarly simple update rules. In such cases, only the update rules for the visible-unit coupling coefficients need to be modified.

-
- [1] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, A tutorial on energy-based learning, 2006, <http://yann.lecun.com/exdb/publis/orig/lecun-06.pdf>.
 - [2] Y. Song and D. P. Kingma, How to train your energy-based models, [arXiv:2101.03288](https://arxiv.org/abs/2101.03288).
 - [3] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, A learning algorithm for Boltzmann machines, *Cogn. Sci.* **9**, 147 (1985).
 - [4] G. E. Hinton, Training products of experts by minimizing contrastive divergence, *Neural Comput.* **14**, 1771 (2002).
 - [5] G. E. Hinton, A practical guide to training restricted Boltzmann machines, in *Neural Networks: Tricks of the Trade*, Lecture Notes in Computer Science (Springer, Berlin, 2012), Vol. 7700, pp. 599–619.
 - [6] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, *Science* **355**, 602 (2017).
 - [7] D. P. Landau and K. Binder, *A Guide to Monte Carlo Methods in Statistical Physics* (Cambridge University Press, Cambridge, England, 2000).
 - [8] R. M. Neal, *Bayesian Learning for Neural Networks* (Springer Science and Business Media, New York, 2012), Vol. 118.
 - [9] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing, *Science* **220**, 671 (1983).
 - [10] S. Chowdhury, A. Grimaldi, N. A. Aadit, S. Niazi, M. Mohseni, and S. Kanai, A full-stack view of probabilistic computing with p-bits: Devices, architectures, and algorithms, *IEEE J. Explor. Solid-State Comput. Devices Circuits* **9**, 1 (2023).
 - [11] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, An introduction to MCMC for machine learning, *Mach. Learn.* **50**, 5 (2003).
 - [12] S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6**, 721 (1984).
 - [13] W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* **57**, 97 (1970).
 - [14] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, *J. Chem. Phys.* **21**, 1087 (1953).
 - [15] H. Goto, K. Tatsumura, and A. R. Dixon, Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems, *Sci. Adv.* **5**, eaav2372 (2019).
 - [16] H. Goto, K. Endo, M. Suzuki, Y. Sakai, T. Kanao, Y. Hamakawa, R. Hidaka, M. Yamasaki, and K. Tatsumura, High-performance combinatorial optimization based on classical mechanics, *Sci. Adv.* **7**, eabe7953 (2021).
 - [17] T. Kanao and H. Goto, Simulated bifurcation assisted by thermal fluctuation, *Commun. Phys.* **5**, 153 (2022).
 - [18] T. Kanao and H. Goto, Simulated bifurcation for higher-order cost functions, *Appl. Phys. Express* **16**, 014501 (2023).
 - [19] H. Goto, R. Hidaka, and K. Tatsumura, Edge-of-chaos enhanced quantum-inspired algorithm for combinatorial optimization, *Phys. Rev. Appl.* **25**, 044011 (2026).
 - [20] M. W. Johnson *et al.*, Quantum annealing with manufactured spins, *Nature (London)* **473**, 194 (2011).
 - [21] R. Harris *et al.*, Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor, *Phys. Rev. B* **82**, 024511 (2010).
 - [22] P. I. Bunyk, E. M. Hoskinson, M. W. Johnson, E. Tolkacheva, F. Altomare, A. J. Berkley, R. Harris, J. P. Hilton, T. Lanting, A. J. Przybysz, and J. Whittaker, Architectural considerations in the design of a superconducting quantum annealing processor, *IEEE Trans. Appl. Supercond.* **24**, 1 (2014).
 - [23] K. Boothby, P. Bunyk, J. Raymond, and A. Roy, Next-generation topology of D-wave quantum processors, [arXiv:2003.00133](https://arxiv.org/abs/2003.00133).
 - [24] M. H. Amin, Searching for quantum speedup in quasistatic quantum annealers, *Phys. Rev. A* **92**, 052323 (2015).
 - [25] S. H. Adachi and M. P. Henderson, Application of quantum annealing to training of deep neural networks, [arXiv:1510.06356](https://arxiv.org/abs/1510.06356).
 - [26] V. Dixit, R. Selvarajan, T. Aldwairi, Y. Koshka, M. A. Novotny, T. S. Humble, M. A. Alam, and S. Kais, Training a quantum annealing based restricted Boltzmann machine on cybersecurity data, *IEEE Trans. Emerg. Top. Comput. Intell.* **6**, 417 (2021).
 - [27] T. Sato, M. Ohzeki, and K. Tanaka, Assessment of image generation by quantum annealer, *Sci. Rep.* **11**, 13523 (2021).
 - [28] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeki, K. Enbutsu, O. Tadanaga, H. Takenouchi, K. Aihara, K. Kawarabayashi, K. Inoue, S. Utsunomiya, and H. Takesue, A coherent Ising machine for 2000-node optimization problems, *Science* **354**, 603 (2016).
 - [29] P. L. McMahon, A. Marandi, Y. Haribara, R. Hamerly, C. Langrock, S. Tamate, T. Inagaki, H. Takesue, S. Utsunomiya, K. Aihara, R. L. Byer, M. M. Fejer, H. Mabuchi, and Y. Yamamoto, A fully programmable 100-spin coherent Ising machine with all-to-all connections, *Science* **354**, 614 (2016).
 - [30] T. Hiroki, Y. Yamada, K. Inaba, T. Ikuta, Y. Yonezu, T. Inagaki, T. Honjo, T. Kazama, K. Enbutsu, T. Umeki, and R. Kasahara,

- Observing a phase transition in a coherent Ising machine, *Phys. Rev. Appl.* **19**, L031001 (2023).
- [31] E. S. Tiunov, A. E. Ulanov, and A. I. Lvovsky, Annealing by simulating the coherent Ising machine, *Opt. Express* **27**, 10288 (2019).
- [32] A. E. Ulanov, E. S. Tiunov, and A. I. Lvovsky, Quantum-inspired annealers as Boltzmann generators for machine learning and statistical physics, *arXiv:1912.08480*.
- [33] H. Goto, Z. Lin, and Y. Nakamura, Boltzmann sampling from the Ising model using quantum heating of coupled nonlinear oscillators, *Sci. Rep.* **8**, 7154 (2018).
- [34] F. Böhm, D. Alonso-Urquijo, G. Verschaffelt, and G. Vander Sande, Noise-injected analog Ising machines enable ultrafast statistical sampling and machine learning, *Nat. Commun.* **13**, 5847 (2022).
- [35] J. Tosca, M. C. Strinati, C. Conti, and C. Ciuti, Emergent equilibrium in all-optical single quantum-trajectory Ising machines, *Phys. Rev. Lett.* **134**, 230404 (2025).
- [36] E. Raimondo, A. Grimaldi, V. Tyberkevych, R. Tomasello, A. Giordano, M. Carpentieri, A. Slavin, M. Chiappini, and G. Finocchio, Adaptive Ising Machine based on phase-locking of an auto-oscillator to a biharmonic external driving with noise, *Phys. Rev. Lett.* **136**, 177201 (2026).
- [37] J. Raymond, S. Yarkoni, and E. Andriyash, Global warming: Temperature estimation in annealers, *Front. ICT* **3**, 23 (2016).
- [38] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, Estimation of effective temperatures in quantum annealers for sampling applications: A case study with possible applications in deep learning, *Phys. Rev. A* **94**, 022308 (2016).
- [39] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, Quantum-assisted learning of hardware-embedded probabilistic graphical models, *Phys. Rev. X* **7**, 041052 (2017).
- [40] S. Osindero and G. E. Hinton, Modeling image patches with a directed hierarchy of Markov random fields, *Adv. Neural Inf. Process. Syst.* **20**, 1121 (2008).
- [41] R. Salakhutdinov, Learning and evaluating Boltzmann machines, Technical Report UTML TR 2008-002 (Department of Computer Science, University of Toronto, 2008).
- [42] H. Schulz, A. Müller, and S. Behnke, Exploiting local structure in stacked Boltzmann machines, in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)* (d-side, Evere, Belgium, 2010).
- [43] M. Mézard, G. Parisi, and M. A. Virasoro, *Spin Glass Theory and Beyond* (World Scientific, Singapore, 1987).
- [44] M. Mézard and A. Montanari, *Information, Physics, and Computation* (Oxford University Press, Oxford, 2009).
- [45] G. E. Hinton and T. J. Sejnowski, Learning and Relearning in Boltzmann Machines, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (MIT Press, Cambridge, MA, 1986), Vol. 1.
- [46] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms* (Cambridge University Press, Cambridge, 2003).
- [47] M. Lichman, M. Lichman, UCI Machine Learning Repository, 2013, <http://archive.ics.uci.edu/ml/>.
- [48] P. J. Rossky, J. D. Doll, and H. L. Friedman, Brownian dynamics as smart Monte Carlo simulation, *J. Chem. Phys.* **69**, 4628 (1978).
- [49] G. O. Roberts and O. Stramer, Langevin diffusions and Metropolis–Hastings algorithms, *Methodol. Comput. Appl. Probab.* **4**, 337 (2002).
- [50] B. Leimkuhler and S. Reich, *Simulating Hamiltonian Dynamics* (Cambridge University Press, Cambridge, 2004).
- [51] P. Virtanen *et al.*, SciPy 1.0: Fundamental algorithms for scientific computing in Python, *Nat. Methods* **17**, 261 (2020).
- [52] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/2vb5-2lt8> for details of the data preparation, hyperparameter settings, additional training procedures, and additional numerical results.