

Received May 26, 2019, accepted June 24, 2019, date of publication June 27, 2019, date of current version July 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2925530

# Learnable Markov Chain Monte Carlo Sampling Methods for Lattice Gaussian Distribution

ZHENG WANG<sup>1</sup>, (Member, IEEE), SHANXIANG LYU<sup>2</sup>, AND LING LIU<sup>1</sup>, (Member, IEEE)

<sup>1</sup>College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing 211106, China

<sup>2</sup>College of Information Science and Technology/College of Cyber Security, Jinan University, Guangzhou 510632, China

<sup>3</sup>Department of CT Lab, Huawei Technologies, Shenzhen 518000, China

Corresponding author: Zheng Wang (z.wang@ieee.org)

This work was supported in part by the open research fund of the Key Laboratory of Dynamic Cognitive System of Electromagnetic Spectrum Space (Nanjing University of Aeronautics and Astronautics), Ministry of Industry and Information Technology, Nanjing, China, under Grant KF20181913, in part by the open research fund of the National Mobile Communications Research Laboratory, Southeast University under Grant 2019D04, in part by the National Natural Science Foundation of China under Grant 61801216, and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20180420.

**ABSTRACT** As a key ingredient of machine learning and artificial intelligence, the sampling algorithms with respect to lattice Gaussian distribution has emerged as an important problem in coding and decoding of wireless communications. In this paper, based on the conventional Gibbs sampling, the learnable delayed metropolis-within-Gibbs (LDMWG) sampling algorithm is proposed to improve the convergence performance, which fully takes the advantages of the acceptance mechanism from the metropolis-hastings (MH) algorithm in the Markov chain Monte Carlo (MCMC) methods. The rejected candidate by the acceptance mechanism is utilized as a learnable experience for the generation of a new candidate at the same Markov move. In this way, the overall probability of remaining the same state at the Markov chain is greatly reduced, which leads to an improved convergence performance in the sense of Peskun ordering. Moreover, in order to reduce the complexity cost during the Markov mixing, a symmetric sampling structure which greatly simplified the sampling operation is further introduced and the symmetric learnable delayed metropolis-within-Gibbs (SLDMWG) sampling algorithm is given. Finally, the simulation results based on multi-input multi-output (MIMO) detections are presented to confirm the convergence gain and the complexity reduction brought by the proposed sampling schemes.

**INDEX TERMS** Lattice coding and decoding, lattice Gaussian sampling, Gibbs sampler decoding, Markov chain Monte Carlo.

## I. INTRODUCTION

Recently, lattice Gaussian distribution plays an important role in physical-layer of wireless communications. In coding, lattice Gaussian distribution was applied to obtain the full shaping gain for lattice coding [1]–[3], and to achieve the capacity of the Gaussian channel [4]. Meanwhile, it was also employed to achieve information-theoretic security in Gaussian wiretap channel [5]–[7]. Furthermore, lattice Gaussian distribution has been adapted to bidirectional relay network under the compute-and-forward strategy for the physical layer security [8]. In addition, it is also used to realize the probabilistic shaping for optical communication systems [9], [10]. In the field of decoding, lattice Gaussian distribution with a suitable variance naturally allows to solve the shortest vector

The associate editor coordinating the review of this manuscript and approving it for publication was Guan Gui.

problem (SVP) and the closest vector problem (CVP) [11], [12]. To this end, sampling over lattice Gaussian distribution has been widely applied in multi-input multi-output (MIMO) communications for signal detection [13]–[15]. Compared to the optimal sphere detection, it is not only much more efficient, but also can be realized flexibly to achieve the trade-off between decoding performance and complexity [16]. In addition, such a sampling decoding strategy can be easily extended to signal processing as an useful signal estimator or detector [17]–[21].

Due to the central role of lattice Gaussian distribution playing in physical-layer of wireless communications, its sampling algorithms become an important computational problem. However, different from the case of the continuous Gaussian density, sampling from the discrete lattice Gaussian distribution is by no means trivial even for a low-dimensional system. For this reason, the sampling methods from Markov

chain Monte Carlo (MCMC), which are already widely applied in machine learning (ML) and artificial intelligence (AI) [22]–[24], were introduced as an alternative way for lattice Gaussian sampling [25], [26]. MCMC sampling-based probabilistic inference has been a highly successful paradigm for machine learning and there are great opportunities for combining existing sub-optimal algorithms with MCMC in many artificial intelligence problems [27]–[29]. Some areas that are already benefiting from sampling methods include computer vision, speech and audio processing, regression and classification [30]–[32]. Meanwhile, MCMC can be used in deep reinforcement learning to sample from the array of possible actions available in any given state [33]–[35].

Typically, in MCMC, after a burn-in stage, which is normally measured by the *mixing time* in total variance distance, the Markov chain will step into a stationary distribution, where samples from the target distribution can be successfully obtained thereafter. As a basic MCMC method, the Gibbs algorithm, which employs univariate conditional sampling to build the Markov chain, has been introduced to lattice Gaussian sampling by showing its ergodicity [26]. In [36], the symmetric Metropolis-within-Gibbs (SMWG) algorithm was proposed for lattice Gaussian sampling to achieve the exponential convergence. Moreover, the Markov chain induced by random scan Gibbs sampling was shown to be geometric ergodicity [37], which means it converges exponentially fast. Besides Gibbs algorithm, other MCMC methods for lattice Gaussian sampling also exist, and the independent Metropolis-Hastings-Klein (IMHK) algorithm is not only uniformly ergodic but also enjoys an accessible convergence rate [25].

In this paper, in order to improve the convergence performance of Gibbs-based lattice Gaussian sampling, we try to update the Metropolis-within-Gibbs sampling algorithm by adding a learnable mechanism. In particular, the concept of multiple sampling stages within each Markov move is introduced, and the rejected candidate by the acceptance judgment at the first sampling stage is used as prior knowledge to serve for another candidate generation at the second stage. By doing this, the probability of staying at the same Markov state is effectively reduced, which results in a better convergence performance of the Markov mixing. Meanwhile, for the sake of complexity reduction, the symmetric sampling structure is also considered, where both the candidate generation and the acceptance ratio computation can be significantly simplified. Theoretically, the number of sampling stages can be expanded to higher order while we only consider two sampling stages through the context. Any extension based on it is possible by constructing the sampling stages in the same way.

The rest of this paper is organized as follows. Section II introduces the lattice Gaussian distribution and briefly reviews the basics of MCMC methods and Metropolis-within-Gibbs (MWG) sampling algorithm. In Section III, the proposed learnable delayed MWG (LDMWG) sampling algorithm is presented for lattice Gaussian sampling, where

the rejected candidate is further exploited for the following sampling. In Section IV, the symmetric learnable delayed MWG (SLDMWG) is proposed to reduce the complexity burden in terms of the univariate sampling. Simulations through MIMO systems are presented in Section V to illustrate the convergence gain and complexity reduction. At the end, Section VI concludes the paper.

*Notation:* Matrices and column vectors are denoted by upper and lowercase boldface letters, and the transpose, pseudoinverse of a matrix  $\mathbf{B}$  by  $\mathbf{B}^T$ ,  $\mathbf{B}^{-1}$ , and  $\mathbf{B}^\dagger$ , respectively. We use  $\mathbf{b}_i$  for the  $i$ th column of the matrix  $\mathbf{B}$ ,  $\widehat{\mathbf{b}}_i$  for the  $i$ th Gram-Schmidt vector of the matrix  $\mathbf{B}$ ,  $b_{i,j}$  for the entry in the  $i$ th row and  $j$ th column of the matrix  $\mathbf{B}$ .  $\lceil x \rceil$  denotes rounding to the integer closest to  $x$ . If  $x$  is a complex number,  $\lceil x \rceil$  rounds the real and imaginary parts separately. In addition, in this paper, the computational complexity is measured by the number of arithmetic operations (additions, multiplications, comparisons, etc.).

## II. PRELIMINARIES

In this section, the background and mathematical tools needed to describe and analyze the MCMC algorithm for lattice Gaussian sampling are introduced.

### A. LATTICE GAUSSIAN DISTRIBUTION

Let  $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_n] \subset \mathbb{R}^n$  consist of  $n$  linearly independent vectors. The  $n$ -dimensional lattice  $\Lambda$  generated by  $\mathbf{B}$  is defined by

$$\Lambda = \mathcal{L}(\mathbf{B}) = \{\mathbf{B}\mathbf{x} : \mathbf{x} \in \mathbb{Z}^n\}, \quad (1)$$

where the full rank matrix  $\mathbf{B} \in \mathbb{R}^{n \times n}$  is called the lattice basis. The *Gaussian function* centered at  $\mathbf{c} \in \mathbb{R}^n$  with standard deviation  $\sigma > 0$  is defined as

$$\rho_{\sigma, \mathbf{c}}(\mathbf{z}) = e^{-\frac{\|\mathbf{z}-\mathbf{c}\|^2}{2\sigma^2}}, \quad (2)$$

for all  $\mathbf{z} \in \mathbb{R}^n$ . When  $\mathbf{c}$  or  $\sigma$  are not specified, they are assumed to be  $\mathbf{0}$  and 1 respectively. Then, the *discrete Gaussian distribution* over  $\Lambda$  is defined as

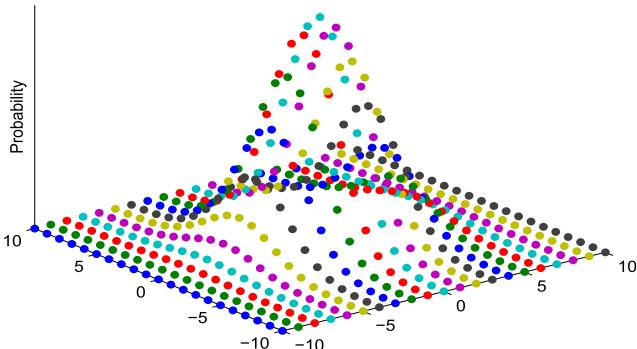
$$D_{\Lambda, \sigma, \mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma, \mathbf{c}}(\mathbf{B}\mathbf{x})}{\rho_{\sigma, \mathbf{c}}(\Lambda)} = \frac{e^{-\frac{1}{2\sigma^2} \|\mathbf{B}\mathbf{x}-\mathbf{c}\|^2}}{\sum_{\mathbf{x} \in \mathbb{Z}^n} e^{-\frac{1}{2\sigma^2} \|\mathbf{B}\mathbf{x}-\mathbf{c}\|^2}} \quad (3)$$

for all  $\mathbf{x} \in \mathbb{Z}^n$ , where  $\rho_{\sigma, \mathbf{c}}(\Lambda) \triangleq \sum_{\mathbf{B}\mathbf{x} \in \Lambda} \rho_{\sigma, \mathbf{c}}(\mathbf{B}\mathbf{x})$  is just a scaling to obtain a probability distribution and  $\sigma > 0$  represents the standard deviation.

At present, the default sampling algorithm for lattice Gaussian distribution is due to Klein, which is originally proposed for bounded-distance decoding (BDD) in lattices [38]. In [39], it has been demonstrated that Klein's algorithm is able to sample from  $D_{\Lambda, \sigma, \mathbf{c}}(\mathbf{x})$  within a negligible statistical distance if

$$\sigma \geq \omega(\sqrt{\log n}) \cdot \max_{1 \leq i \leq n} \|\widehat{\mathbf{b}}_i\|. \quad (4)$$

However, even with the help of lattice reduction, the requirement of standard deviation  $\omega(\sqrt{\log n}) \cdot \max_{1 \leq i \leq n} \|\widehat{\mathbf{b}}_i\|$  is too



**FIGURE 1.** Illustration of a two-dimensional lattice Gaussian distribution.

large to be useful in practice, rendering Klein's algorithm inapplicable to many cases of interest.

## B. MCMC METHODS

By establishing a Markov chain that randomly generates the next state, MCMC is capable of sampling from the target distribution of interest. In particular, as for lattice Gaussian sampling by Gibbs algorithm, each coordinate of  $\mathbf{x}$  is randomly chosen to sample from the following 1-dimensional conditional distribution

$$P_i(x_i|\mathbf{x}_{[-i]}) = D_{\Lambda,\sigma,\mathbf{c}}(x_i|\mathbf{x}_{[-i]}) = \frac{e^{-\frac{1}{2\sigma^2} \|\mathbf{B}\mathbf{x} - \mathbf{c}\|^2}}{\sum_{x_i \in \mathbb{Z}} e^{-\frac{1}{2\sigma^2} \|\mathbf{B}\mathbf{x} - \mathbf{c}\|^2}} \quad (5)$$

with  $\sigma > 0$ . Here  $1 \leq i \leq n$  denotes the coordinate index of  $\mathbf{x}$ ,  $\mathbf{x}_{[-i]} \triangleq [x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n]^T$ . During this univariate sampling, the other  $n - 1$  variables contained in  $\mathbf{x}_{[-i]}$  are leaving unchanged. By repeating such a procedure with a certain scan scheme, a Markov chain  $\{\mathbf{X}^0, \mathbf{X}^1, \dots\}$  is established, whose transition probability  $P(\mathbf{X}^t, \mathbf{X}^{t+1})$  can be expressed as

$$P(\mathbf{X}^t = \mathbf{x}, \mathbf{X}^{t+1} = \mathbf{y}) = P_i(x_i|\mathbf{x}_{[-i]}). \quad (6)$$

Clearly, every two adjacent states  $\mathbf{X}^t = \mathbf{x} = [x_1^t, \dots, x_i^t, \dots, x_n^t]^T$  and  $\mathbf{X}^{t+1} = \mathbf{y} = [x_1^t, \dots, x_i^{t+1}, \dots, x_n^t]^T$  differ from each other by at most only one coordinate  $x_i$ .

*Theorem 1 ([37]): Given the invariant lattice Gaussian distribution  $D_{\Lambda,\sigma,\mathbf{c}}$ , the Markov chain induced by Gibbs sampler is geometrically ergodic as*

$$\|P^t(\mathbf{x}, \cdot) - D_{\Lambda,\sigma,\mathbf{c}}\|_{TV} \leq M(\mathbf{x})\varrho, \quad (7)$$

where  $0 \leq \varrho < 1$ .

From Theorem 1, if time permits to exponentially converge to the stationary distribution, Gibbs algorithm will draw samples from  $D_{\Lambda,\sigma,\mathbf{c}}$  no matter what value  $\sigma > 0$  is, which means the obstacle encountered by Klein's algorithm is overcome.

## C. METROPOLIS-WITHIN-GIBBS ALGORITHM

On the other hand, Gibbs algorithm is actually a special case of Metropolis-Hastings (MH) sampling that tackles with multi-dimensional problems through conditional univariate sampling. Interestingly, the proposal distribution  $q(\mathbf{x}, \mathbf{y})$  in

MH-based algorithm can be set as any fixed distribution for conveniently drawing samples. Then, given the current state  $\mathbf{X}^t = \mathbf{x}$  for Markov chain, a state candidate  $\mathbf{y}$  for the next Markov move  $\mathbf{X}^{t+1}$  can be generated from  $q(\mathbf{x}, \mathbf{y})$ . After that, accepting  $\mathbf{y}$  as the state of Markov move  $\mathbf{X}^{t+1}$  or not is determined by the acceptance ratio  $\alpha$

$$\alpha(\mathbf{x}, \mathbf{y}) = \min \left\{ 1, \frac{\pi(\mathbf{y})q(\mathbf{y}, \mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x}, \mathbf{y})} \right\}, \quad (8)$$

where  $\pi(\cdot)$  denotes the target distribution. In this way, a valid Markov chain  $\{\mathbf{X}^0, \mathbf{X}^1, \dots\}$  is established with the transition probability  $P(\mathbf{X}^t, \mathbf{X}^{t+1})$  as follows:

$$P(\mathbf{X}^t = \mathbf{x}, \mathbf{X}^{t+1} = \mathbf{y}) = \begin{cases} q(\mathbf{x}, \mathbf{y})\alpha(\mathbf{x}, \mathbf{y}) & \text{if } \mathbf{y} \neq \mathbf{x}, \\ 1 - \sum_{\mathbf{z} \neq \mathbf{x}} q(\mathbf{x}, \mathbf{z})\alpha(\mathbf{x}, \mathbf{z}) & \text{if } \mathbf{y} = \mathbf{x}. \end{cases} \quad (9)$$

In [40], Metropolis-within-Gibbs (MWG) sampling algorithm for lattice Gaussian distribution was proposed, which takes advantages of both acceptance-mechanism from MH sampling and univariate sampling from Gibbs algorithm. Specifically, following the instruction of classic MH algorithm, given the Markov state  $\mathbf{X}^t = \mathbf{x} = [x_1^t, \dots, x_i^t, \dots, x_n^t]^T$ , a state candidate  $\mathbf{y} = [x_1^t, \dots, x_i^*, \dots, x_n^t]^T$  for  $\mathbf{X}^{t+1}$  is obtained through the proposal distribution

$$q(\mathbf{x}, \mathbf{y}) = q(x_i|\mathbf{x}_{-i}) = \frac{D_{\Lambda,\sigma,\mathbf{c}}(x_i|\mathbf{x}_{-i})}{1 - D_{\Lambda,\sigma,\mathbf{c}}(x_i^t|\mathbf{x}_{-i}^t)}, \quad x_i \in \mathbb{Z} \quad (10)$$

and  $x_i^t$  from the  $i$ th coordinate of  $\mathbf{x}$  is eliminated from the state space  $\mathbb{Z}$  in sampling  $x_i^*$ , which results in a reduced state space  $\bar{\mathbb{Z}}$  with

$$\bar{\mathbb{Z}} \cup x_i^t = \mathbb{Z} \quad \text{and} \quad \bar{\mathbb{Z}} \cap x_i^t = \emptyset. \quad (11)$$

In other words, the sample  $x_i^*$  in  $\mathbf{y}$  is obtained according to the sampling from (10), namely,

$$x_i^* \sim q(x_i|\mathbf{x}_{-i}). \quad (12)$$

Once  $x_i^*$  in  $\mathbf{y}$  is obtained, then the acceptance ratio  $\alpha$  in (8) is calculated, and the decision about whether to accept it as  $\mathbf{X}^{t+1} = \mathbf{y}$  is performed thereafter. Note that in conventional Gibbs algorithm  $\mathbf{y}$  will be accepted by  $\mathbf{X}^{t+1}$  without uncertainty. This is the core difference between the proposed MWG and Gibbs algorithms since the uncertainty in the judgment of  $\mathbf{X}^{t+1}$  to choose  $\mathbf{y}$  or not is retained. It has been demonstrated in [40] that given the invariant lattice Gaussian distribution  $D_{\Lambda,\sigma,\mathbf{c}}$ , Metropolis-within-Gibbs algorithm achieves a better exponential convergence performance than Gibbs algorithm.

## III. LEARNABLE DELAYED MWG ALGORITHM

In this section, to enhance the convergence performance of MWG algorithm for lattice Gaussian sampling, learnable delayed Metropolis-within-Gibbs (LDMWG) algorithm is proposed. By reducing the rejection probability of the acceptance mechanism, the sampler turns out to be more efficient according to *Pusunk order*.

**Algorithm 1** Metropolis-Within-Gibbs Algorithm for Lattice Gaussian Sampling

---

**Input:**  $\mathbf{B}, \sigma, \mathbf{c}, \mathbf{X}^0, \beta_i$ 's,  $t_{\text{mix}}(\epsilon)$   
**Output:**  $\mathbf{x} \sim \pi$ ,  $\pi$  is within statistical distance of  $\epsilon$  to  $D_{\Lambda, \sigma, \mathbf{c}}$

- 1: **for**  $t = 1, 2, \dots$  **do**
- 2:   let  $\mathbf{x}$  denote the state of  $\mathbf{X}^{t-1}$
- 3:   randomly choose index  $i$  by distribution  $[\beta_1, \dots, \beta_n]$
- 4:   sample  $x_i^*$  by proposal distribution  $q(x_i | \mathbf{x}_{[-i]})$  in (12)
- 5:   calculate the acceptance quantity  $\alpha$  shown in (8)
- 6:   generate a sample  $u \sim U[0, 1]$
- 7:   **if**  $u \leq \alpha$  **then**
- 8:     get  $\mathbf{y}$  with the sampled  $x_i$  and let  $\mathbf{X}^t = \mathbf{y}$
- 9:   **else** let  $\mathbf{X}^t = \mathbf{x}$
- 10:   **end if**
- 11:   **if**  $t \geq t_{\text{mix}}(\epsilon)$  **then**
- 12:     output the state of  $\mathbf{X}^t$
- 13:   **end if**
- 14: **end for**

---

Since the classic acceptance-rejection mechanism from MH is retained by the MWG sampling algorithm, the case of candidate rejection during the Markov move is also brought back, which could be further optimized from the perspective of Peskun ordering. Specifically, Peskun ordering serves as an important way for the convergence comparison between Markov chains, and it was further generalized by Mira in [41].

*Lemma 1 ([41]): Given reversible Markov chains  $P$  and  $Q$  with stationary distribution  $\pi$ , if each of the off-diagonal elements of the transition matrix  $\mathbf{P}$  is greater than or equal to the corresponding off-diagonal elements of  $\mathbf{Q}$ , i.e.,  $P \succeq Q$ , then Markov chain  $P$  is more efficient than  $Q$  due to a better convergence rates with*

$$\varrho_P \leq \varrho_Q. \quad (13)$$

The insight behind Peskun ordering is that a Markov chain has smaller probability of remaining in the same position would explore the state space more efficiently, thus providing better Markov mixing. This is straightforward to understand since if the Markov chain retains the same state over subsequent Markov moves, the autocorrelation along the exploration path will increase accordingly, which naturally leads to an inefficient Markov mixing. Hence, based on Peskun ordering, the convergence efficiency of MWG sampling algorithm could be further improved by decreasing the rejection opportunities of proposed moves.

We now give the proposed learnable delayed MWG algorithm. Basically,  $q(\mathbf{x}, \mathbf{y})$  in (10) only serves as the proposal distribution at the first stage. If  $x_i^*$  is accepted by  $\mathbf{X}^{t+1}$  successively, then the Markov chain proceeds to the next move smoothly in the traditional way. However, once  $x_i^*$  is rejected, instead of retaining the same position as  $\mathbf{X}^{t+1} = \mathbf{x}$ , a second stage is proposed to sample another candidate  $x'_i$  according to

the following proposal distribution

$$\begin{aligned} q_2(\mathbf{x}, \mathbf{y}, \mathbf{z}) &= q(x_i | \mathbf{x}_{-i}, x_i^t, x_i^*) \\ &= \frac{D_{\Lambda, \sigma, \mathbf{c}}(x_i | \mathbf{x}_{-i})}{1 - D_{\Lambda, \sigma, \mathbf{c}}(x_i^t | \mathbf{x}_{-i}) - D_{\Lambda, \sigma, \mathbf{c}}(x_i^* | \mathbf{x}_{-i})}, \quad x_i \in \overline{\mathbb{Z}}' \end{aligned} \quad (14)$$

where  $\overline{\mathbb{Z}}' \cup x_i^t \cup x_i^* = \mathbb{Z}$ ,  $\overline{\mathbb{Z}}' \cap x_i^t = \emptyset$  and  $\overline{\mathbb{Z}}' \cap x_i^* = \emptyset$ . Note that  $\mathbf{z} = [x_1^t, \dots, x_i^t, \dots, x_n^t]^T$  is different from  $\mathbf{x}$  and  $\mathbf{y}$  by only one component. After that, another judgment based on acceptance ratio  $\alpha_2(\mathbf{x}, \mathbf{y}, \mathbf{z})$  will be carried out to decide whether accept the new candidate  $x'_i$  (i.e.,  $\mathbf{z}$ ) by  $\mathbf{X}^{t+1}$  or not.

By doing this, there is a second chance for the Markov move to accept the generated candidate from the proposal distribution. Therefore, the overall probability of remaining in the current state is reduced, which leads to improved convergence performance in the sense of Peskun ordering. Note that the proposal distribution at the second stage can be different from the first stage. Most importantly, it could depend on the previously rejected candidate, which implies we can learn from the previous experience to further enhance the convergence performance. For this reason, in our proposed learnable delayed MWG algorithm, besides  $x_i^t$  in  $\mathbf{x}$ ,  $x_i^*$  in  $\mathbf{y}$  is also considered in sampling of  $x'_i$  for  $\mathbf{z}$ .

However, the application of another proposal distribution  $q_2(\mathbf{x}, \mathbf{y}, \mathbf{z})$  may destroy the Markovian property of the underlying MCMC sampler, and the most important property is *detailed balance*, which is also known as reversibility [42]. More precisely, detailed balance is a sufficient condition for MCMC-based lattice Gaussian sampling since the considered state space  $\mathbf{x} \in \mathbb{Z}^n$  is countably infinite [43]. In this condition, the acceptance ratio  $\alpha_2(\mathbf{x}, \mathbf{y}, \mathbf{z})$  at second stage should be designed carefully to preserve the stationary distribution, and here we choose the following result as the solution

$$\begin{aligned} \alpha_2(\mathbf{x}, \mathbf{y}, \mathbf{z}) &= \min \left\{ 1, \frac{D(\mathbf{z})q_1(\mathbf{z}, \mathbf{y})q_2(\mathbf{z}, \mathbf{y}, \mathbf{x})[1 - \alpha_1(\mathbf{z}, \mathbf{y})]}{D(\mathbf{x})q_1(\mathbf{x}, \mathbf{y})q_2(\mathbf{x}, \mathbf{y}, \mathbf{z})[1 - \alpha_1(\mathbf{x}, \mathbf{y})]} \right\} \\ &= \min \left\{ 1, \beta \cdot \frac{[1 - \alpha_1(\mathbf{z}, \mathbf{y})]}{[1 - \alpha_1(\mathbf{x}, \mathbf{y})]} \right\}, \end{aligned} \quad (15)$$

where

$$\beta \triangleq \frac{[1 - D(x_i^t | \mathbf{x}_{-i})][1 - D(x_i^t | \mathbf{x}_{-i}) - D(x_i^* | \mathbf{x}_{-i})]}{[1 - D(x_i' | \mathbf{x}_{-i})][1 - D(x_i^* | \mathbf{x}_{-i}) - D(x_i' | \mathbf{x}_{-i})]}. \quad (16)$$

*Proposition 1: With the acceptance ratio  $\alpha_2(\mathbf{x}, \mathbf{y}, \mathbf{z})$  at the second stage, the underlying Markov chain induced by the proposed learnable delaying MWG sampling is detailed balance by satisfying*

$$D(\mathbf{x})P(\mathbf{x}, \mathbf{z}) = D(\mathbf{z})P(\mathbf{z}, \mathbf{x}). \quad (17)$$

*Proof:* Assume candidate  $x_i^*$  in  $\mathbf{y}$  is rejected at the first stage, then the application of the second stage is invoked. Then, the transition probability  $P(\mathbf{X}^t = \mathbf{x}, \mathbf{X}^{t+1} = \mathbf{z})$  can be written as

$$P(\mathbf{x}, \mathbf{z}) = q_1(\mathbf{x}, \mathbf{y}) \cdot [1 - \alpha_1(\mathbf{x}, \mathbf{y})] \cdot q_2(\mathbf{x}, \mathbf{y}, \mathbf{z}) \cdot \alpha_2(\mathbf{x}, \mathbf{y}, \mathbf{z}), \quad (18)$$

where the detailed balance condition can be easily verified by

$$\begin{aligned} D(\mathbf{x})P(\mathbf{x}, \mathbf{z}) &= \min[D(\mathbf{x})q_1(\mathbf{x}, \mathbf{y})q_2(\mathbf{y}, \mathbf{z})[1 - \alpha_1(\mathbf{x}, \mathbf{y})], \\ &\quad D(\mathbf{z})q_1(\mathbf{z}, \mathbf{y})q_2(\mathbf{y}, \mathbf{x})[1 - \alpha_1(\mathbf{z}, \mathbf{y})]] \\ &= D(\mathbf{z})P(\mathbf{z}, \mathbf{x}), \end{aligned} \quad (19)$$

completing the proof. ■

Notice that in order to obtain the numerator of the acceptance ratio at the second stage, it is necessary to compute the acceptance probability  $\alpha_1(\mathbf{z}, \mathbf{y})$ . Based on it, reject the candidate  $x_i^*$  and accept a second stage move to  $x_i^t$ . Nevertheless, this is only a mental trial, which is not in fact implemented. Therefore, to summarize, the proposed learnable delayed MWG sampling algorithm can be described by the following steps.

#### A. STAGE 1

1) Sample from the proposal distribution  $q_1(\mathbf{x}, \mathbf{y})$  in (10) to obtain the candidate  $x_i^*$ .

2) Make a judgment based on  $\alpha_1(\mathbf{x}, \mathbf{y})$  in (8) about whether accept  $\mathbf{y} = [x_1^t, \dots, x_i^*, \dots, x_n^t]^T$  by  $\mathbf{X}^{t+1}$ . If  $\mathbf{y} = [x_1^t, \dots, x_i^*, \dots, x_n^t]^T$  is rejected, go to the operations at next stage.

#### B. STAGE 2

3) Sample from the proposal distribution  $q_2(\mathbf{x}, \mathbf{y}, \mathbf{z})$  in (14) to obtain the candidate  $x_i'$ .

4) Make a decision for  $\mathbf{X}^{t+1}$  based on  $\alpha_2(\mathbf{x}, \mathbf{y}, \mathbf{z})$  in (15) to accept  $\mathbf{z} = [x_1^t, \dots, x_i', \dots, x_n^t]^T$  or not (i.e.,  $\mathbf{X}^{t+1} = \mathbf{x}$ ).

To make it clear, the operations of the proposed learnable delayed Metropolis-within-Gibbs (LDMWG) sampling algorithm is shown in Algorithm 2. As a comparison, the information of  $x_i^t$  cannot be used in the later iteration in a regular MH algorithm since the Markovian property like detailed balance does not hold any more. On the contrary, the learnable delayed MWG allows to use the prior knowledge obtained at different stages within the same Markov move, which still retains the Markovian property for the whole sampler. Clearly, if the candidate at the second stage is also rejected, it is possible to move on to a third stage and so on to further exploit the convergence potential. Note that the proposal distribution at the second stage could be the same with that at stage 1. However, setting it in a learnable way is helpful to the convergence performance by Peskun order [44]. Also, how to further take advantages of the previous experience is able to provide another freedom for the convergence gain, which will be one of our work in future.

*Corollary 1:* Compared to the Metropolis-within-Gibbs (MWG) sampling, the overall probability of remaining the same state at the Markov chain is reduced in the proposed learnable delayed Metropolis-within-Gibbs (LDMWG) sampling algorithm, which leads to a better Markov mixing in the sense of Peskun order.

On the other hand, the price paid for this convergence gain actually requires additional computational cost within each Markov move. With respect to it, related tentative simulations

---

#### Algorithm 2 Learnable Delayed Metropolis-Within-Gibbs Algorithm for Lattice Gaussian Sampling

---

**Input:**  $\mathbf{B}, \sigma, \mathbf{c}, \mathbf{X}^0, \beta_i$ 's,  $t_{\text{mix}}(\epsilon)$

**Output:**  $\mathbf{x} \sim \pi$ ,  $\pi$  is within statistical distance of  $\epsilon$  to  $D_{\Lambda, \sigma, \mathbf{c}}$

```

1: for  $t = 1, 2, \dots$  do
2:   let  $\mathbf{x}$  denote the state of  $\mathbf{X}^{t-1}$ 
3:   randomly choose index  $i$  by distribution  $[\beta_1, \dots, \beta_n]$ 
4:   sample  $x_i^*$  by proposal distribution  $q_1(x_i | \mathbf{x}_{[-i]})$  in (12)
5:   calculate the acceptance quantity  $\alpha_1$  shown in (8)
6:   generate a sample  $u \sim U[0, 1]$ 
7:   if  $u \leq \alpha_1$  then
8:     get  $\mathbf{y}$  with the sampled  $x_i$  and let  $\mathbf{X}^t = \mathbf{y}$ 
9:   else sample  $x_i'$  by proposal distribution  $q_2(\cdot)$  in (14)
10:    calculate the acceptance quantity  $\alpha_2$  in (15)
11:    generate a sample  $v \sim U[0, 1]$ 
12:    if  $v \leq \alpha_2$  then
13:      get  $\mathbf{z}$  with the sampled  $x_i'$  and let  $\mathbf{X}^t = \mathbf{z}$ 
14:    else let  $\mathbf{X}^t = \mathbf{x}$ 
15:    end if
16:  end if
17:  if  $t \geq t_{\text{mix}}(\epsilon)$  then
18:    output the state of  $\mathbf{X}^t$ 
19:  end if
20: end for

```

---

have been made to show that the performance gain is more than the incurred price [45]. Another merit of the proposed learnable delayed sampling scheme comes from the application of parallel tempering, which fully exploits the usage of multiple Markov chains [46]. More specifically, when multiple Markov chains are applied, a better convergence performance of each Markov chain under limited Markov moves turns out to be rather important for the following sample exchange under different temperatures.

## IV. SYMMETRIC LEARNABLE DELAYED MWG ALGORITHM

In this section, in order to reduce the computational cost brought by the learnable delayed operation within each Markov move, the symmetric sampling structure is considered, which leads to the symmetric learnable delayed MWG sampling algorithm.

Specifically, at each sampling stage, the proposal distribution is designed as a 1-dimensional conditional symmetric Gaussian distribution:

$$\begin{aligned} q_1(\mathbf{x}, \mathbf{y}) &= q_{\text{symmetric}}(x_i^t \rightarrow x_i^* | \mathbf{x}_{[-i]}) \\ &= \frac{e^{-\frac{1}{2\sigma^2}|x_i^* - x_i^t|^2}}{\sum_{x_i^* \in \mathbb{Z}} e^{-\frac{1}{2\sigma^2}|x_i^* - x_i^t|^2}} \\ &= q_{\text{symmetric}}(x_i^* \rightarrow x_i^t | \mathbf{x}_{[-i]}) \\ &= q_1(\mathbf{y}, \mathbf{x}) \end{aligned} \quad (20)$$

for  $\mathbf{x} \neq \mathbf{y}$  with  $x_i^* \in \overline{\mathbb{Z}}$ .

Clearly, by doing this, the generation of the state candidate  $\mathbf{y}$  is completely independent of the other  $n-1$  unchanged components, but heavily depends on the  $i$ -th component of  $\mathbf{x}$ , i.e.,  $x_i$ , in the previous state  $\mathbf{x}$ . Meanwhile, since the chain is symmetric, the calculation of the acceptance ratio  $\alpha_1(\mathbf{x}, \mathbf{y})$  at the first stage is also greatly simplified by such an inherent elegance:

$$\begin{aligned}\alpha_1(\mathbf{x}, \mathbf{y}) &= \min \left\{ 1, \frac{D(\mathbf{y})q_1(\mathbf{y}, \mathbf{x})}{D(\mathbf{x})q_1(\mathbf{x}, \mathbf{y})} \right\} \\ &= \min \left\{ 1, \frac{D(\mathbf{y})}{D(\mathbf{x})} \right\} \\ &= \min \left\{ 1, \frac{D(x_i^* | \mathbf{x}_{[-i]}) \cdot D(\mathbf{x}_{[-i]})}{D(x_i^t | \mathbf{x}_{[-i]}) \cdot D(\mathbf{x}_{[-i]})} \right\} \\ &= \min \left\{ 1, \frac{e^{-\frac{1}{2\sigma_i^2} |x_i^* - c|^2}}{e^{-\frac{1}{2\sigma_i^2} |x_i^t - c|^2}} \right\}, \quad (21)\end{aligned}$$

where  $c$  stands for the summation of the rest of  $n-1$  components of  $\mathbf{x}$  multiplied by their own scaling coefficients. It should be noticed that at each step, the proposed sampling algorithm focuses on the sampling over a 1-dimensional symmetric distribution rather than a full dimension distribution. Therefore, compared to the MWG algorithm, it is more suitable for multi-dimensional target distributions. Furthermore, it has been demonstrated in [40] that the symmetric Gibbs algorithm for lattice Gaussian sampling is geometric ergodicity by satisfying the *drift condition*, which means the Markov chain converges exponentially fast to the stationary distribution. Here, by removing the effect of  $x_i^t$  from  $\mathbf{X}^{t+1}$ , a better convergence performance can be obtained to explore the state space more efficiently.

Intuitively,  $x_i^*$  from  $\mathbf{y}$  is accepted by the Markov state  $\mathbf{X}^{t+1}$  if  $D(x_i^* | \mathbf{x}_{[-i]}) > D(x_i^t | \mathbf{x}_{[-i]})$ , otherwise it would be accepted with probability  $D(x_i^* | \mathbf{x}_{[-i]}) / D(x_i^t | \mathbf{x}_{[-i]})$ . On the other hand, if  $x_i^*$  is rejected, then the sampling moves to stage 2 and another candidate  $x_i'$  will be generated from the proposal distribution

$$\begin{aligned}q_2(\mathbf{x}, \mathbf{y}, \mathbf{z}) &= q(x_i^* \rightarrow x_i' | \mathbf{x}_{[-i]}, x_i^t) \\ &= \frac{e^{-\frac{1}{2\sigma^2} |x_i' - x_i^*|^2}}{\sum_{x_i' \in \mathbb{Z}} e^{-\frac{1}{2\sigma^2} |x_i' - x_i^*|^2}} \\ &= q(x_i' \rightarrow x_i^* | \mathbf{x}_{[-i]}, x_i^t) \\ &= q_2(\mathbf{z}, \mathbf{y}, \mathbf{x}) \quad (22)\end{aligned}$$

for  $\mathbf{x} \neq \mathbf{y} \neq \mathbf{z}$  with  $x_i' \in \overline{\mathbb{Z}}$ . Clearly, at the second stage,  $x_i'$  from  $\mathbf{x}$  and  $x_i^*$  from  $\mathbf{y}$  are learnt as prior knowledge to enhance the convergence performance. By doing this, the sampling result of  $\mathbf{z}$  cannot be  $\mathbf{x}$  or  $\mathbf{y}$ , which implies the probability of  $\mathbf{x}$  being retained as the Markov state of  $\mathbf{X}^{t+1}$  is reduced.

Next, in order to maintain the Markovian properties for the stationary distribution, the acceptance ratio  $\alpha_2(\mathbf{x}, \mathbf{y}, \mathbf{z})$  should be defined carefully. Specifically, considering the symmetric learnable delayed sampling scheme, the transition probability

---

**Algorithm 3** Symmetric Learnable Delayed Metropolis-Within-Gibbs Algorithm for Lattice Gaussian Sampling

---

**Input:**  $\mathbf{B}, \sigma, \mathbf{c}, \mathbf{X}^0, \beta_i$ 's,  $t_{\text{mix}}(\epsilon)$   
**Output:**  $\mathbf{x} \sim \pi$ ,  $\pi$  is within statistical distance of  $\epsilon$  to  $D_{\Lambda, \sigma, \mathbf{c}}$

```

1: for  $t = 1, 2, \dots$  do
2:   let  $\mathbf{x}$  denote the state of  $\mathbf{X}^{t-1}$ 
3:   randomly choose index  $i$  by distribution  $[\beta_1, \dots, \beta_n]$ 
4:   sample  $x_i^*$  by proposal distribution  $q_1(x_i | \mathbf{x}_{[-i]})$  in (20)
5:   calculate the acceptance quantity  $\alpha_1$  shown in (21)
6:   generate a sample  $u \sim U[0, 1]$ 
7:   if  $u \leq \alpha_1$  then
8:     get  $\mathbf{y}$  with the sampled  $x_i$  and let  $\mathbf{X}^t = \mathbf{y}$ 
9:   else sample  $x_i'$  by proposal distribution  $q_2(\cdot)$  in (22)
10:    calculate the acceptance quantity  $\alpha_2$  in (25)
11:    generate a sample  $v \sim U[0, 1]$ 
12:    if  $v \leq \alpha_2$  then
13:      get  $\mathbf{z}$  with the sampled  $x_i'$  and let  $\mathbf{X}^t = \mathbf{z}$ 
14:    else let  $\mathbf{X}^t = \mathbf{x}$ 
15:    end if
16:   end if
17:   if  $t \geq t_{\text{mix}}(\epsilon)$  then
18:     output the state of  $\mathbf{X}^t$ 
19:   end if
20: end for

```

---

at the second stage can be expressed as

$$P(\mathbf{x}, \mathbf{z}) = q_1(\mathbf{x}, \mathbf{y}) \cdot [1 - \alpha_1(\mathbf{x}, \mathbf{y})] \cdot q_2(\mathbf{x}, \mathbf{y}, \mathbf{z}) \cdot \alpha_2(\mathbf{x}, \mathbf{y}, \mathbf{z}). \quad (23)$$

Then, in order to make the condition of detailed balance shown in (17) satisfied, we choose the following result as the solution

$$\alpha_2(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \min \left\{ 1, \frac{D(x_i' | \mathbf{x}_{[-i]}) [1 - \min \left[ 1, \frac{D(x_i^* | \mathbf{x}_{[-i]})}{D(x_i' | \mathbf{x}_{[-i]})} \right]]}{D(x_i^t | \mathbf{x}_{[-i]}) - D(x_i^* | \mathbf{x}_{[-i]})} \right\}, \quad (24)$$

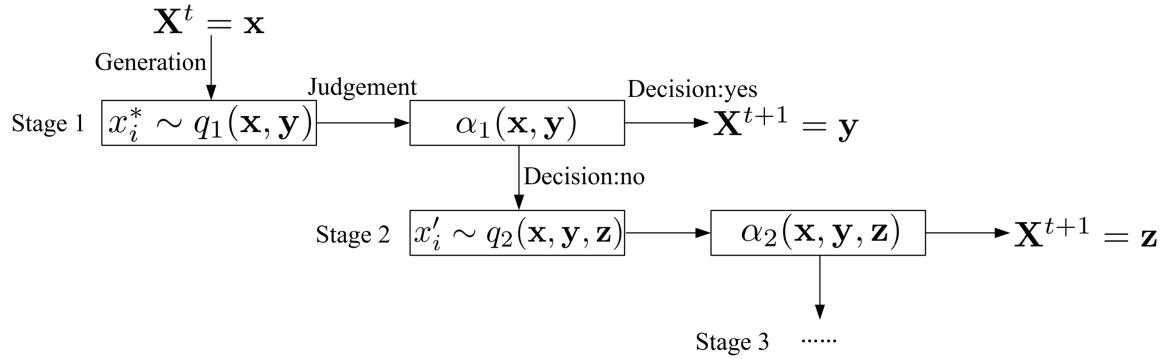
where the related proof is omitted due to the simplicity.

Furthermore, the acceptance ratio  $\alpha_2(\mathbf{x}, \mathbf{y}, \mathbf{z})$  can be rewritten as

$$\begin{aligned}\alpha_2(\mathbf{x}, \mathbf{y}, \mathbf{z}) &= \min \left\{ 1, \frac{\max \{ 0, D(x_i' | \mathbf{x}_{[-i]}) - D(x_i^* | \mathbf{x}_{[-i]}) \}}{D(x_i^t | \mathbf{x}_{[-i]}) - D(x_i^* | \mathbf{x}_{[-i]})} \right\} \\ &= F \left( \frac{D(x_i' | \mathbf{x}_{[-i]}) - D(x_i^* | \mathbf{x}_{[-i]})}{D(x_i^t | \mathbf{x}_{[-i]}) - D(x_i^* | \mathbf{x}_{[-i]})} \right), \quad (25)\end{aligned}$$

where  $F$  is the cumulative distribution function of an uniform random variable over the interval  $(0, 1)$ . Subsequently, to summarize, there are three possible cases at this point:

- 1) if  $D(x_i' | \mathbf{x}_{[-i]}) > D(x_i^* | \mathbf{x}_{[-i]})$ , then  $\alpha_2(\mathbf{x}, \mathbf{y}, \mathbf{z}) = 1$ , accept  $x_i'$  and set  $\mathbf{X}^{t+1} = \mathbf{z}$ .
- 2) if  $D(x_i' | \mathbf{x}_{[-i]}) > D(x_i^* | \mathbf{x}_{[-i]}) > D(x_i^t | \mathbf{x}_{[-i]})$ , accept  $x_i'$  with probability  $\frac{D(x_i' | \mathbf{x}_{[-i]}) - D(x_i^* | \mathbf{x}_{[-i]})}{D(x_i^t | \mathbf{x}_{[-i]}) - D(x_i^* | \mathbf{x}_{[-i]})}$ .
- 3) if  $D(x_i' | \mathbf{x}_{[-i]}) < D(x_i^* | \mathbf{x}_{[-i]})$ , then  $\alpha_2(\mathbf{x}, \mathbf{y}, \mathbf{z}) = 0$ , reject  $x_i'$  and set  $\mathbf{X}^{t+1} = \mathbf{x}$ .



**FIGURE 2.** Illustration of the learnable delayed Metropolis-within-Gibbs sampling scheme from Markov move  $\mathbf{x}^t$  to  $\mathbf{x}^{t+1}$  with  $\mathbf{x} = [x_1^t, \dots, x_i^t, \dots, x_n^t]^T$ ,  $\mathbf{y} = [x_1^t, \dots, x_i^*, \dots, x_n^t]^T$  and  $\mathbf{z} = [x_1^t, \dots, x_i', \dots, x_n^t]^T$ .  $q_1(\cdot)$  and  $q_2(\cdot)$  denote the proposal distributions at stages 1 and 2 respectively.  $\alpha_1(\cdot)$  and  $\alpha_2(\cdot)$  denote the acceptance ratios at stages 1 and 2 respectively.

Clearly, by fully taking advantages of symmetric structure of the 1-dimensional sampling, the acceptance ratio  $\alpha_2(\mathbf{x}, \mathbf{y}, \mathbf{z})$  of the second stage in (25) is considerably simpler than that in (15). Meanwhile, the generation of candidates of  $x_i^*$  and  $x_i'$  is also significantly simplified for the complexity reduction. On the other hand, if  $x_i'$  is also rejected, it is possible to move on to the next stage and obtain another candidate from the proposal distribution  $q_3(\cdot)$  based on the previous samples  $\mathbf{x}$ ,  $\mathbf{y}$  and  $\mathbf{z}$ . Meanwhile, the acceptance probability  $\alpha_3(\cdot)$  for this new candidate can be well designed in the same way. Overall, by doing this, the probability of remaining the current state along the Markov move is certainly reduced, which leads to improved samplers in the sense of Peskun ordering. To summarize, the operations of the proposed symmetric learnable delayed Metropolis-within-Gibbs (SLDMWG) algorithm for lattice Gaussian sampling is presented in Algorithm 3.

Another point should be emphasized is that in specific applications, the state space of the Markov chain would belong to a finite state space. For example, in MIMO systems,  $\mathbf{x}$  normally comes from the designed constellation of the QAM modulation. In this condition, the space of each component of  $\mathbf{x}$  is limited by the system configuration. From the perspective of the learnable acceptance mechanism, this is beneficial for the candidate generation as the rejected ones in the previous stage can be easily removed from the candidate list. More specifically, in MIMO detection with 4-QAM, there is only two integer candidates, which can be solved within one sampling stage. In other words, the learnable delayed sampling mechanism is more suitable for the detection case under high-order modulation schemes due to the application of multiple sampling stages.

## V. SIMULATION RESULTS

In this section, the performance of the proposed MCMC sampling schemes for lattice Gaussian distribution are exemplified in the context of MIMO detection.

Specifically, simulation results for an  $n \times n$  MIMO system with a square channel matrix containing i.i.d. Gaussian entries are presented. The  $i$ -th entry of the transmitted signal

$\mathbf{x}$ , denoted as  $x_i$ , is a modulation symbol taken independently from a  $Q^2$ -QAM constellation  $\mathcal{X} \in \mathbb{Z}$  with Gray mapping. Meanwhile, it is assumed a flat fading environment, where the channel matrix  $\mathbf{H}$  contains uncorrelated complex Gaussian fading gains with unit variance and remains constant over each frame duration. Let  $E_b$  represents the average power per bit at the receiver, then  $E_b/N_0 = n/(\log_2(M)\sigma_w^2)$  holds where  $M$  is the modulation level and  $\sigma_w^2$  is the noise power. Then, we can construct the system model of MIMO as

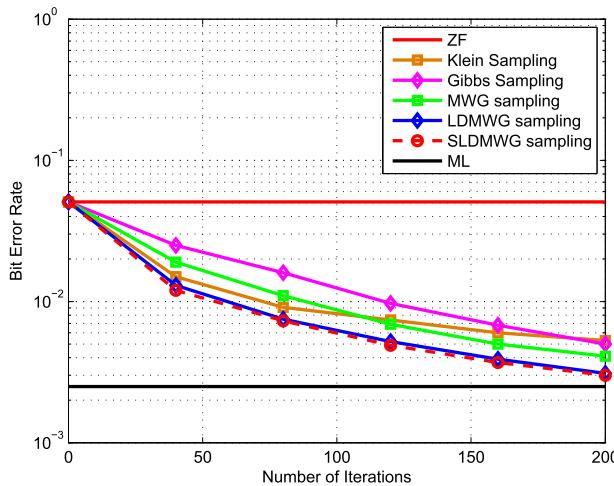
$$\mathbf{c} = \mathbf{H}\mathbf{x} + \mathbf{w}, \quad (26)$$

and this decoding problem of  $\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathcal{X}^n} \|\mathbf{c} - \mathbf{H}\mathbf{x}\|^2$  can be solved by sampling over the discrete Gaussian distribution

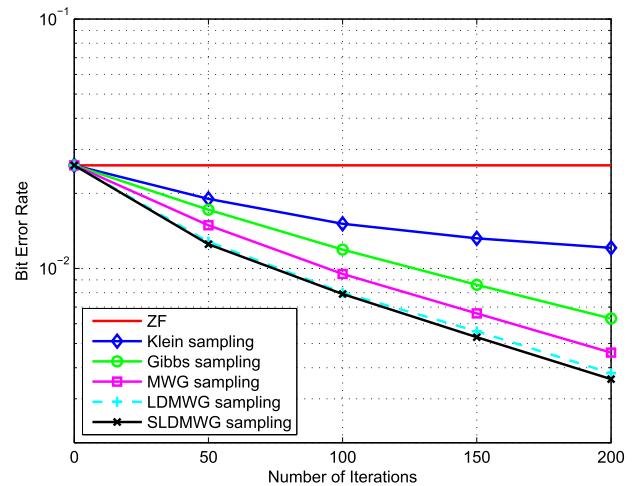
$$P_{\mathcal{L}(\mathbf{H}), \sigma, \mathbf{c}}(\mathbf{x}) = \frac{e^{-\frac{1}{2\sigma^2} \|\mathbf{H}\mathbf{x} - \mathbf{c}\|^2}}{\sum_{\mathbf{x} \in \mathcal{X}^n} e^{-\frac{1}{2\sigma^2} \|\mathbf{H}\mathbf{x} - \mathbf{c}\|^2}} \quad (27)$$

because the optimal solution has the largest probability making it most likely be encountered by sampling (this complex decoding system is straightforward to be extended to the real-valued system [15], [47]). For this reason, we examine the decoding error probabilities to approximately compare the convergence performance of Markov chains. Here, Babai rounding algorithm (also known as zero-forcing decoding) is applied to output the suboptimal result for the initial Markov state [48]. Meanwhile, as for the proposed LDMWG and SLDMWG sampling schemes, only two sampling stages are considered for the simplicity, where further extension of the number of sampling stages can be easily made.

In Fig. 2 depicts the bit error rates (BER) of the different sampling schemes in a  $4 \times 4$  uncoded MIMO system with 16-QAM. This corresponds to lattice dimension  $n = 8$  and the SNR is fixed as  $E_b/N_0 = 10$  dB. The performance of zero-forcing (ZF) and maximum-likelihood (ML) decoding are shown as benchmarks. For a fair comparison, we follow Klein's choice of  $\sigma = \min_{1 \leq i \leq n} \|\hat{\mathbf{b}}_i\|/\sqrt{\log n}$  and run the univariate sampling in both MWG and Gibbs algorithm for  $n$  times as a full iteration. As can be seen clearly, the decoding performance improves with the number of Markov chain



**FIGURE 3.** Bit error rate versus the number of iterations for the uncoded  $4 \times 4$  MIMO system using 16-QAM.

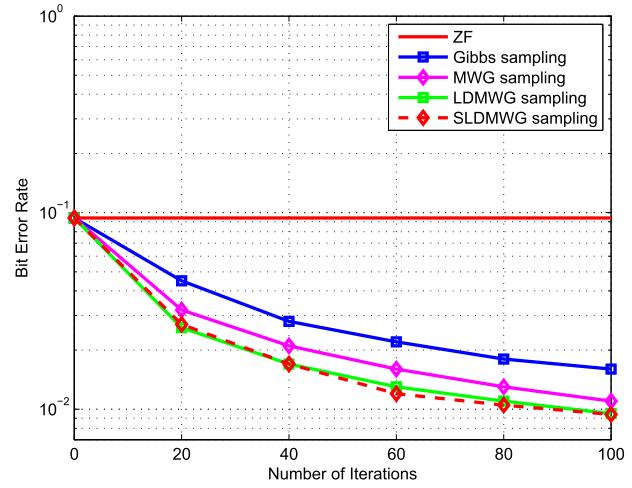


**FIGURE 4.** Bit error rate versus the number of iterations for the uncoded  $6 \times 6$  MIMO system using 16-QAM.

iterations. In particular, Klein's sampling is not as good as MCMC sampling schemes since it does not really produce Gaussian samples [39]. On the other hand, as demonstrated, the proposed LDMWG algorithm outperforms MWG algorithm under the same number of iterations, implying a better convergence performance. Meanwhile, the decoding performance of the SLDMWG algorithm is also presented, which is nearly the same with that of LDMWG. This is because the underlying Markov chain also converges exponentially even under the symmetric sampling structure. Nevertheless, the complexity cost by SLDMWG turns out to be much lower due to the usage of symmetric sampling, which will be illustrated in the following.

Fig. 3 depicts the bit error rates (BER) of the different sampling schemes in a  $6 \times 6$  uncoded MIMO system with 16-QAM. The SNR is fixed as  $E_b/N_0 = 10$  dB. This corresponds to lattice dimension  $n = 12$ . The performance of zero-forcing (ZF) is added as a benchmark. For a fair comparison, we follow Klein's choice of  $\sigma = \min_{1 \leq i \leq n} \|\hat{\mathbf{b}}_i\| / \sqrt{\log n}$  and run the univariate sampling in both MWG and Gibbs algorithm for  $n$  times as a full iteration. As shown in Fig. 3, the decoding performance improves with the number of Markov chain iterations. As expected, the proposed LDMWG algorithm outperforms MWG and Gibbs algorithms under the same number of iterations, which implies a better convergence performance. On the other hand, SLDMWG is comparable to LDMWG in terms of decoding performance, which means the application of symmetric sampling does not affect the convergence performance.

In Fig. 4, the BERs of MCMC sampling detectors are evaluated against the number of Markov moves (i.e., iterations) in a  $8 \times 8$  uncoded MIMO system with 16-QAM. The SNR is fixed as  $E_b/N_0 = 15$  dB and the standard deviation  $\sigma$  is set as 2 at this time. Clearly, the performances of all the MCMC detectors improve with the number of Markov moves. By taking advantages of acceptance mechanism and univariate sampling, a certain of decoding performance can be obtained



**FIGURE 5.** Bit error rate versus the number of iterations for the uncoded  $8 \times 8$  MIMO system using 16-QAM.

by MWG algorithm. Based on it, the proposed LDMWG further exploits the potential of the sampled candidate within each Markov move, which results in a better convergence performance. Moreover, the proposed SLDMWG is able to reduce the complexity cost with negligible performance loss, which is shown in the following Table 1.

Specifically, Table 1 shows the average acceptance proportions for the three sampling stages of the proposed LDMWG and SLDMWG in a  $4 \times 4$  uncoded MIMO system with 64-QAM. The SNR is fixed as  $E_b/N_0 = 10$  dB while the standard deviation is chosen as  $\sigma = \min_{1 \leq i \leq n} \|\hat{\mathbf{b}}_i\| / \sqrt{\log n}$ . Intuitively, in the conventional Gibbs sampling, all the generated candidates will be accepted by the next Markov move without uncertainty. However, the candidate could be the same with the last Markov state, making the Markov move unchanged. On the other hand, in MWG algorithm, the acceptance mechanism is invoked and the generated candidate is accepted with probability 0.72, which means the Markov chain has probability 0.28 to stay at the same state.

**TABLE 1.** Average acceptance proportions for the three sampling stages for uncoded  $4 \times 4$  MIMO system with 64-QAM.

	Stage 1	Stage 2	Stage 3	Rejected
Gibbs	1	0	0	0
MWG	0.72	0	0	0.28
LDMWG	0.72	0.16	0.04	0.08
SLDMWG	0.74	0.15	0.05	0.06

**TABLE 2.** Average complexity in flops of sampling schemes for uncoded MIMO system with 16-QAM.

	Gibbs	MWG	LDMWG	SLDMWG
$n = 8$	288	314	465	376
$n = 12$	903	986	1451	1187
$n = 16$	2095	2257	3351	2650

Furthermore, in the proposed algorithms, by fully exploiting those rejected candidates at the first sampling stage, the overall probability of the Markov chain remaining the same state is reduced to 0.08 and 0.06 respectively, which leads to a certain of convergence gain by exploring the state space more efficiently.

Table 2 illustrates the average complexity comparison in flops with different system dimensions, where the flops evaluation scenario that we use comes from [49]. More specifically, the  $n/2 \times n/2$  uncoded MIMO systems are applied with 16-QAM and the SNR is fixed as  $E_b/N_0 = 15$  dB. Note that here we consider the average induced flops in one full iteration. As can be seen clearly, Gibbs sampler has the lowest complexity cost due to simplicity, and MWG is slightly higher than Gibbs since the acceptance mechanism is introduced. As for LDMWG, because extra sampling stage is introduced for the candidate generation, more complexity cost is required to achieve the convergence improvement. Due to this, the symmetric sampling structure is applied in SLDMWG to reduce the complexity burden, whose computational cost is much less than LDMWG.

## VI. CONCLUSION

In this paper, the learning mechanism was introduced to Gibbs-based MCMC algorithms for lattice Gaussian sampling. According to Peskun ordering, the probability of remaining the current Markov state is significantly reduced, which leads to an improved convergence performance of the Markov mixing. More specifically, the learnable delayed Metropolis-within-Gibbs algorithm is proposed to exploit the potential of the prior knowledge, which is provided by the acceptance mechanism from Metropolis-Hastings algorithm. On the other hand, to further reduce the complexity cost,

the symmetric sampling structure is also considered and symmetric learnable delayed Metropolis-within-Gibbs algorithm is given, which not only reduces the complexity burden in candidate generation but also simplifies the computation of the acceptance ratio.

## REFERENCES

- [1] G. D. Forney, "Multidimensional constellations—Part II: Voronoi constellations," *IEEE J. Sel. Areas Commun.*, vol. 7, no. 6, pp. 941–958, Aug. 1989.
- [2] F. R. Kschischang and S. Pasupathy, "Optimal nonuniform signaling for Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 913–929, May 1993.
- [3] L. Liu and C. Ling, "Polar codes and polar lattices for independent fading channels," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4923–4935, Dec. 2016.
- [4] C. Ling and J.-C. Belfiore, "Achieving the AWGN channel capacity with lattice Gaussian coding," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5918–5929, Oct. 2014.
- [5] L. Liu, Y. Yan, and C. Ling, "Achieving secrecy capacity of the Gaussian wiretap channel with polar lattices," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1647–1665, Mar. 2018.
- [6] C. Ling, L. Luzzi, J.-C. Belfiore, and D. Stehlé, "Semantically secure lattice codes for the Gaussian wiretap channel," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 6399–6416, Oct. 2014.
- [7] H. Mirghasemi and J.-C. Belfiore, "Lattice code design criterion for MIMO wiretap channels," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Oct. 2015, pp. 277–281.
- [8] S. Vatedka, N. Kashyap, and A. Thangaraj, "Secure compute-and-forward in a bidirectional relay," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2531–2556, May 2015.
- [9] T. Fehenberger, D. Lavery, R. Maher, A. Alvarado, P. Bayvel, and N. Hanik, "Sensitivity gains by mismatched probabilistic shaping for optical communication systems," *IEEE Photon. Technol. Lett.*, vol. 28, no. 7, pp. 786–789, Apr. 1, 2016.
- [10] T. Fehenberger, A. Alvarado, G. Böcherer, and N. Hanik, "On probabilistic shaping of quadrature amplitude modulation for the nonlinear fiber channel," *J. Lightw. Technol.*, vol. 34, no. 21, pp. 5063–5073, Nov. 1, 2016.
- [11] D. Aggarwal, D. Dadush, O. Regev, and N. Stephens-Davidowitz, "Solving the shortest vector problem in  $2^n$  time using discrete Gaussian sampling: Extended abstract," in *Proc. STOC*, 2015, pp. 733–742.
- [12] D. Aggarwal, D. Dadush, and N. Stephens-Davidowitz, "Solving the closest vector problem in  $2^n$  time—The discrete Gaussian strike again!" in *Proc. FOCS*, Oct. 2015, pp. 563–582.
- [13] Z. Wang and C. Ling, "Lattice Gaussian sampling by Markov chain Monte Carlo: Bounded distance decoding and trapdoor sampling," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3630–3645, Jun. 2019.
- [14] Z. Wang, S. Liu, and C. Ling, "Decoding by sampling—Part II: Derandomization and soft-output decoding," *IEEE Trans. Commun.*, vol. 61, no. 11, pp. 4630–4639, Nov. 2013.
- [15] S. Liu, C. Ling, and D. Stehlé, "Decoding by sampling: A randomized lattice algorithm for bounded distance decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 5933–5945, Sep. 2011.
- [16] S. Yang and L. Hanzo, "Fifty years of MIMO detection: The road to large-scale MIMOS," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 1941–1988, Sep. 2015.
- [17] K. Luo and A. Manikas, "Joint transmitter-receiver optimization in multitarget MIMO radar," *IEEE Trans. Signal Process.*, vol. 65, no. 23, pp. 6292–6302, Dec. 2017.
- [18] H. Cheng, Y. Xia, Y. Huang, L. Yang, and D. P. Mandic, "A normalized complex LMS based blind I/Q imbalance compensator for GFDM receivers and its full second-order performance analysis," *IEEE Trans. Signal Process.*, vol. 66, no. 17, pp. 4701–4712, Sep. 2018.
- [19] Q. Wu, G. Ding, J. Wang, and Y. D. Yao, "Spatial-temporal opportunity detection for spectrum-heterogeneous cognitive radio networks: Two-dimensional sensing," *IEEE Trans. Wireless Commun.*, vol. 12, no. 2, pp. 516–526, Feb. 2013.
- [20] J. Zhuang, H. Xiong, W. Wang, and Z. Chen, "Application of manifold separation to parametric localization for incoherently distributed sources," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2849–2860, Jun. 2018.

- [21] M. Xiang, B. S. Dees, and D. P. Mandic, "Multiple-model adaptive estimation for 3-D and 4-D signals: A widely linear quaternion approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, pp. 72–84, Jan. 2019.
- [22] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Mach. Learn.*, vol. 50, no. 1, pp. 5–43, Jan. 2003.
- [23] T. Wang, Y. Wu, D. Moore, and S. J. Russell, "Meta-learning MCMC proposals," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NIPS)*, Montreal, QC, Canada, 2018, pp. 1–11.
- [24] I. Murray. (2012). *Introduction to MCMC for Deep Learning*. [Online]. Available: <https://www.youtube.com/watch?v=Em6mQQy4wYA>
- [25] Z. Wang and C. Ling, "On the geometric ergodicity of metropolis-Hastings algorithms for lattice Gaussian sampling," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 738–751, Feb. 2018.
- [26] Z. Wang, C. Ling, and G. Hanrot, "Markov chain Monte Carlo algorithms for lattice Gaussian sampling," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, Jun. 2014, pp. 1489–1493.
- [27] R. Salakhutdinov, "Learning deep Boltzmann machines using adaptive MCMC," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, 2010, pp. 1–8.
- [28] M. Liu, T. Song, J. Hu, J. Yang, and G. Gui, "Deep learning-inspired message passing algorithm for efficient resource allocation in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 641–653, Jan. 2019.
- [29] M. Liu, T. Song, and G. Gui, "Deep cognitive perspective: Resource allocation for NOMA-based heterogeneous IoT with imperfect SIC," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2885–2894, Apr. 2019.
- [30] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
- [31] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, Apr. 2019.
- [32] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo, and Z. Wang, "Predicting head movement in panoramic video: A deep reinforcement learning approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
- [33] I. Goodfellow, Y. Bengio, A. Courville, and F. Bach, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [34] H. Huang, Y. Song, J. Yang, G. Gui, and F. Adachi, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, Mar. 2019.
- [35] G. Gui, H. Huang, Y. Song, and H. Sari, "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8440–8450, Sep. 2018.
- [36] Z. Wang and C. Ling, "Symmetric Metropolis-within-Gibbs algorithm for lattice Gaussian sampling," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Sep. 2016, pp. 394–398.
- [37] Z. Wang and C. Ling, "On the geometric ergodicity of Gibbs algorithm for lattice Gaussian sampling," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2017, pp. 269–273.
- [38] P. Klein, "Finding the closest lattice vector when it's unusually close," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, 2000, pp. 937–941.
- [39] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. 40th Annu. ACM Symp. Theory Comput.*, Victoria, BC, Canada, 2008, pp. 197–206.
- [40] Z. Wang and C. Ling, "Symmetric metropolis-within-Gibbs algorithm for lattice Gaussian sampling," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Cambridge, U.K., Sep. 2016, pp. 394–398.
- [41] A. Mira, "Ordering and improving the performance of Monte Carlo Markov chains," *Stat. Sci.*, vol. 16, no. 4, pp. 340–350, 2001.
- [42] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
- [43] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times*. Providence, RI, USA: American Mathematical Society, 2008.
- [44] P. J. Green and A. Mira, "Delayed rejection in reversible jump metropolis-Hastings," *Biometrika*, vol. 88, pp. 1035–1053, Dec. 2001.
- [45] L. Tierney and A. Mira, "Some adaptive Monte Carlo methods for Bayesian inference," *Statist. Med.*, vol. 18, pp. 2507–2515, Sep. 1999.
- [46] D. J. Earl and M. W. Deem, "Parallel tempering: Theory, applications, and new perspectives," *Phys. Chem. Chem. Phys.*, vol. 7, no. 23, pp. 3910–3916, 2005.
- [47] Y. Xia and D. P. Mandic, "Augmented performance bounds on strictly linear and widely linear estimators with complex data," *IEEE Trans. Signal Process.*, vol. 66, no. 2, pp. 507–514, Jan. 2018.
- [48] L. Babai, "On Lovász' lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.
- [49] H. Qian, "Counting the floating point operations (FLOPS)," MATLAB Central File Exchange, Tech. Rep. 50608, Ver. 1.0, Jun. 2015.

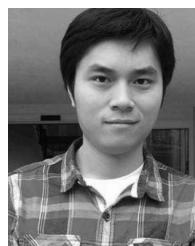


**ZHENG WANG** received the B.S. degree in electronic and information engineering from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2009, the M.S. degree in communications from the Department of Electrical and Electronic Engineering, The University of Manchester, Manchester, U.K., in 2010, and the Ph.D. degree in communication engineering from Imperial College London, U.K., in 2015.

From 2015 to 2016, he served as a Research Associate with Imperial College London, U.K. From 2016 to 2017, he was a Senior Engineer with the Radio Access Network R&D division, Huawei Technologies Company. He is currently an Assistant Professor with the College of Electronic and Information Engineering, NUAA. His current research interests include lattice methods for wireless communications, cognitive radio, and physical layer security.



**SHANXIANG LYU** received the B.Eng. and M.Eng. degrees in electronic and information engineering from the South China University of Technology, Guangzhou, China, in 2011 and 2014, respectively, and the Ph.D. degree from the Electrical and Electronic Engineering Department, Imperial College London, in 2018. He is currently a Lecturer with the College of Cyber Security, Jinan University. His research interests include lattice theory and algebraic number theory, and their applications.



**LING LIU** received the B.S. degree from Nanjing University, Nanjing, China, in 2008, the M.S. degree from Peking University, Beijing, China, in 2012, and the Ph.D. degree from the Electrical and Electronic Engineering Department, Imperial College London, in 2016. He is currently with the CT Lab, Huawei Technologies, Shenzhen, China. His research interests include classical and quantum information theory, coding theory, physical layer security, and lattices.