

Article

Robust Spike-Based Continual Meta-Learning Improved by Restricted Minimum Error Entropy Criterion

Shuangming Yang¹, Jiangtong Tan¹ and Badong Chen^{2,*} 

¹ School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China; yangshuangming@tju.edu.cn (S.Y.); shuangmingyang@tju.edu.cn (J.T.)

² Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China

* Correspondence: chenbd@mail.xjtu.edu.cn

Abstract: The spiking neural network (SNN) is regarded as a promising candidate to deal with the great challenges presented by current machine learning techniques, including the high energy consumption induced by deep neural networks. However, there is still a great gap between SNNs and the online meta-learning performance of artificial neural networks. Importantly, existing spike-based online meta-learning models do not target the robust learning based on spatio-temporal dynamics and superior machine learning theory. In this invited article, we propose a novel spike-based framework with minimum error entropy, called MeMEE, using the entropy theory to establish the gradient-based online meta-learning scheme in a recurrent SNN architecture. We examine the performance based on various types of tasks, including autonomous navigation and the working memory test. The experimental results show that the proposed MeMEE model can effectively improve the accuracy and the robustness of the spike-based meta-learning performance. More importantly, the proposed MeMEE model emphasizes the application of the modern information theoretic learning approach on the state-of-the-art spike-based learning algorithms. Therefore, in this invited paper, we provide new perspectives for further integration of advanced information theory in machine learning to improve the learning performance of SNNs, which could be of great merit to applied developments with spike-based neuromorphic systems.

Keywords: spiking neural network; meta-learning; information theoretic learning; minimum error entropy; artificial general intelligence



Citation: Yang, S.; Tan, J.; Chen, B. Robust Spike-Based Continual Meta-Learning Improved by Restricted Minimum Error Entropy Criterion. *Entropy* **2022**, *24*, 455. <https://doi.org/10.3390/e24040455>

Academic Editors: Lizhong Zheng and Chao Tian

Received: 25 February 2022

Accepted: 23 March 2022

Published: 25 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, deep learning has shown a superior performance that exceeds the human-level performance in various types of individual narrow tasks [1]. However, in comparison with human intelligence that can learn to learn continually in order to execute unlimited tasks, the current successful deep learning methods still have a lot of drawbacks and limitations. In fact, humans can learn to learn by accumulating knowledge across their life time, which is a great challenge for artificial neural networks (ANNs) [2]. From this point of view, continual meta-learning aims at realizing machine intelligence at a higher level by providing machines with the meta-learning capability of learning to learn continually [3].

The human brain can realize meta-learning continually and avoid the catastrophic forgetting problem based on a combination of neural mechanisms [4]. The catastrophic forgetting problem is the critical challenge for developing the capability of continual meta-learning [5]. The human brain has implemented an efficient and scalable mechanism for continual learning based on neuronal activity patterns that represent previous experiences [6]. Neurons communicate with each other and process the neural information by using neural spikes, which is one of the most critical fundamental mechanism in the brain. Based on this mechanism, the human brain can realize superior performance in

different aspects, such as low power consumption and high spatio-temporal processing capability [7]. Therefore, implementing a brain-inspired continual meta-learning algorithm based on spike patterns and the brain's mechanisms is a promising technique.

The spiking neural network (SNN) uses the biologically plausible neuron model based on spiking dynamics, while the conventional ANN only uses the neurons based on a static rate [8]. SNNs are applied to reproduce the brain's mechanisms and to deal with the cognitive tasks [9]. In addition, the neuromorphic hardware based on SNNs can realize high performance in artificial intelligence tasks, including low power consumption, high noise tolerance, and low computation latency [10]. Previous neuromorphic hardware researches have proven these advantages by using various types of tasks, such as Tianjic, Loihi, BiCoSS, CerebelluMorphic, and LaCSNN [11–15]. Researchers have proposed SNN models to realize the short-term memory capability in a spike-based framework [16]. However, the current SNN models still suffer from the continual meta-learning problem under the non-Gaussian noise, and no previous study has solved this problem. Therefore, this is the focus of this study.

Information theoretic learning (ITL) has attracted increasing attention in the field of machine learning in recent years to improve the learning robustness and enhance the explainable capability [17–19]. Previously, Chen et al. proposed researches focusing on maximum correntropy theory and minimum error entropy criteria to improve the robustness of machine learning theory [20–22]. In addition, a series of entropy-based learning algorithms have been presented to deal with the robustness improvement of machine learning models, including guided complement entropy and fuzzy entropy [23–25]. Nevertheless, there is no application of the ITL-based approach in the spike-based continual meta-learning to improve its learning robustness. Therefore, in this invited article, we aim to propose a novel approach to deal with this challenging problem. A novel model is presented, which is called meta-learning with minimum error entropy (MeMEE). We test the meta-learning capability of the proposed SNN model. Then, we investigate the robust working memory capability in non-Gaussian noise. Finally, the robust transfer learning performance is explored under a non-Gaussian noisy condition. Experimental results strongly suggest the robust meta-learning capability of the SNN model with a working memory feature in a non-Gaussian noisy environment.

2. Materials and Methods

2.1. SNN Model

Previous studies have shown that the firing timing and activity space of dendrites can significantly affect neural function. Excitability of dendrites can excite the membrane to fire, whereas inhibitory dendrites can have the opposite effect [26–29]. Inspired by this morphological structure and function of the neuron model, we propose a spiking neuron model, which has three compartments, including a somatic compartment and two dendritic compartments. The model utilizes distinct dendritic compartments to receive excitatory and inhibitory inputs, while using dendrites and somatic cells to receive and send spiking activities, respectively. The formulation for calculating the membrane potential of dendrites and soma are as follows

$$\begin{cases} \tau_m \frac{dU_m(t)}{dt} = -U_m(t) + R_m I_m(t) + g_i(U_i(t) - \theta_i) + g_e(U_e(t) - \theta_e) - \Gamma_j(t)z_j(t) \\ \tau_i \frac{dU_i(t)}{dt} = -U_i(t) + R_i I_i(t) \\ \tau_e \frac{dU_e(t)}{dt} = -U_e(t) + R_e I_e(t) \end{cases} \quad (1)$$

where τ_v represents the time constant of membrane. The variables $U(t)$, $U_i(t)$, and $U_e(t)$ represent the somatic membrane potentials, inhibitory dendritic membrane potentials, and excitatory dendritic membrane potentials, respectively. The parameters θ_e and θ_i represent the reversal membrane potential of excitatory dendrite and inhibitory dendrite, respectively. R_m , R_e , and R_i represent the membrane resistance of the soma, excitatory dendrite, and inhibitory dendrite, respectively. The parameters g_e and g_i represent the

synaptic conductance of excitatory dendrites and inhibitory dendrites, respectively. Neuron emits a spike at time t when it is currently not in a refractory period. The soma of neurons uses the spike adaptation mechanism. The threshold size can be changed by analyzing the firing pattern of neurons. Variable $z_j(t)$ represents the spike train of neuron j and assumes value in $\{0, 1/\Delta t\}$. The dynamics of $\Gamma_j(t)$ is changed with each spike, representing the firing rate of neuron j , which is defined as

$$\Gamma_j(t) = \tau_j^0 + \alpha \cdot \tau_j(t) \tag{2}$$

where α represents a constant that scales the deviation $\tau_j(t)$ from the baseline τ_j^0 . The variable $\tau_j(t)$ can be defined as

$$\tau_j(t + \Delta t) = \beta_j \tau_j(t) + (1 - \beta_j) z_j(t) \tag{3}$$

where $\beta_j = \exp(-\Delta t / \tau_{a,j})$. The constant $\tau_{a,j}$ represents the adaptation time constant. Variable $z_j(t)$ represents the spike train of neuron j and assumes value in $\{0, 1/\Delta t\}$. The parameter values of the spiking neuron model that we proposed are listed in Table 1. The input current $I_j(t)$ of a neuron is defined as the weighted sum of the pulses, which come from external neurons or other neurons. Its mathematical formula is as follows

$$\begin{cases} I_m^j(t) = \sum_{i=1}^n W_{ij} \chi_i(t - \kappa_{ij}) + \sum_{i=1}^n W_{ij}^{rec} \varepsilon_i(t - \kappa_{ij}^{rec}) \\ I_i^j(t) = \sum_{i=1}^n W_{ij}^i \chi_i(t - \kappa_{ij}^i) + \sum_{i=1}^n W_{ij}^{irec} \varepsilon_i(t - \kappa_{ij}^{irec}) \\ I_e^j(t) = \sum_{i=1}^n W_{ij}^e \chi_i(t - \kappa_{ij}^e) + \sum_{i=1}^n W_{ij}^{erec} \varepsilon_i(t - \kappa_{ij}^{erec}) \end{cases} \tag{4}$$

where W_{ij}^{rec} , W_{ij}^{erec} , and W_{ij}^{irec} represent the recurrent synaptic weights of soma, excitatory dendrites, and inhibitory dendrites, respectively. In addition, W_{ij} , W_{ij}^i , and W_{ij}^e represent the synaptic weights of soma, excitatory dendrite, and inhibitory dendrite, respectively. The constants κ_{ij} , κ_{ij}^e , and κ_{ij}^i represent the delays of input synapses for soma, excitatory dendrite, and inhibitory dendrite, respectively. The constants κ_{ij}^{rec} , κ_{ij}^{erec} , and κ_{ij}^{irec} represent the delays of recurrent synapses for soma, excitatory dendrite, and inhibitory dendrite, respectively. The spike trains $\chi_i(t)$ and $\varepsilon_i(t)$ are modeled as sums of Dirac pulses, representing the spike trains from input neurons and recurrent neurons with recurrent connections, respectively. The dynamics of the proposed spiking neuron model are shown in Figure 1 accordingly.

Table 1. Parameter settings of the spiking neuron model.

Parameter	Value	Parameter	Value
R_m	1 Ω	R_i, R_e	1 Ω
τ_m	20 ms	θ_i, θ_e	0 mV
$\kappa, \kappa^i, \kappa^e$	5 ms	$\kappa^{rec}, \kappa^{irec}, \kappa^{erec}$	5 ms
α	1.8	τ^0	0.01
τ_a	700 ms	g_i, g_e	1 nS

We integrate the spiking neuron model into an SNN framework and test the accuracy of this new model on different types of learning tasks. The structure of the SNN model is shown in Figure 2. The model is divided into three layers: input layer, hidden layer, and output layer. According to different tasks, we choose different encoding methods of the input layer and decoding methods of the output layer. In Figure 2, the solid blue lines represent feed-forward inhibitory synaptic connections, while the red dashed lines represent lateral inhibitory synaptic connections. The dendrites and soma of different neurons in the hidden layer are connected by lateral inhibitory synapses that are random and sparse at the same time. Information is transmitted from the input layer to the dendrites,

and the soma transmits impulse signals to the output layer. The initial network weights in the proposed SNN model are set via a Gaussian distribution $W_{ij} \sim \frac{w_0}{\sqrt{n_{in}}} N(0, 1)$, where n_{in} represents the number of input neurons in the spiking neural network in the weight matrix. $N(0, 1)$ represents the Gaussian distribution with zero mean and unit variance, while $w_0 = \Delta t / R_m$ represents a weight-scaling factor depending on the time step Δt and membrane resistance R_m . This scaling factor is significant as it is used to initialize the spiking neural network with a practical firing rate needed for efficient training.

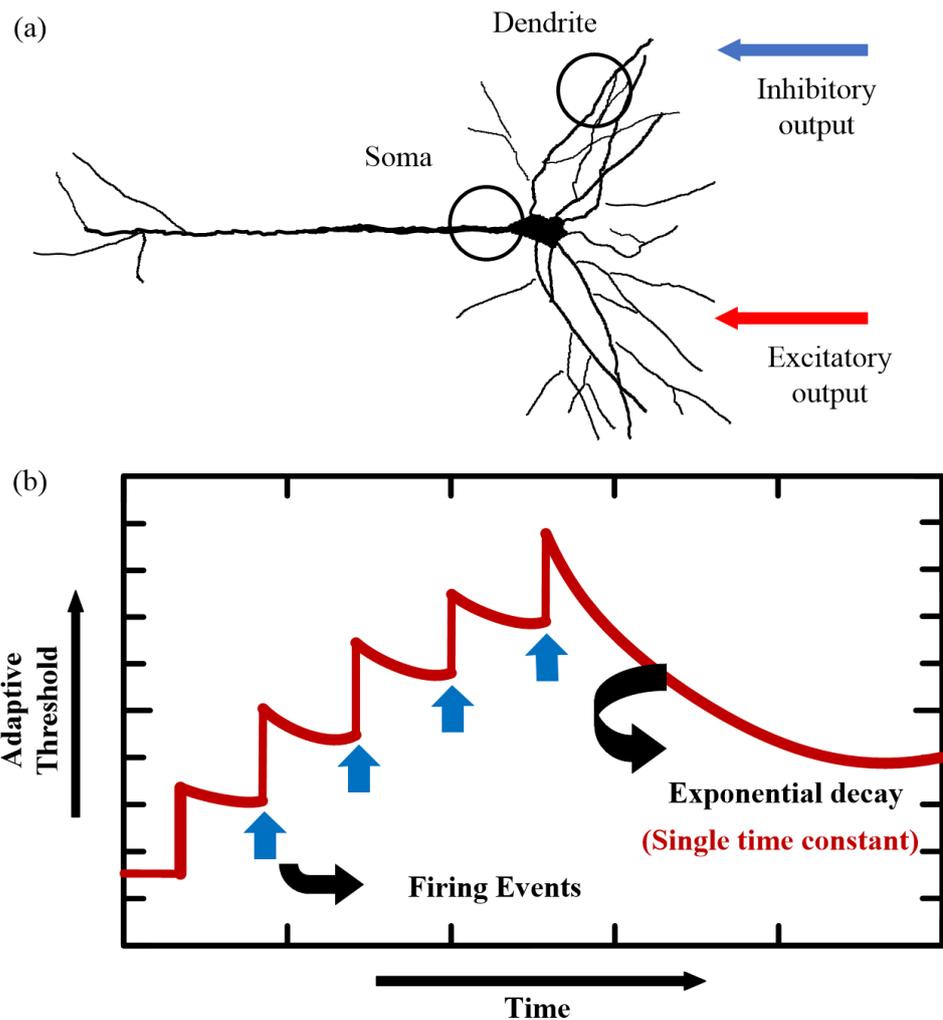


Figure 1. Dynamics of the proposed spiking neuron. (a) The biological structure that inspires the proposed neuron model. (b) The adaptive dynamics of the threshold along with the firing events.

We use a deep rewiring algorithm because it is able to maintain the sign of each synapse during the learning process [30]. Hence, this sign is inherited from the initial weights of the network. In consideration of this, the model needs efficient and reasonable initialization weights for both excitatory and inhibitory neurons. To achieve this, we sample neurons from a Bernoulli distribution, generating the symbol $sign k_i \in \{-1, 1\}$ randomly. At the same time, to avoid the problem of exploding gradients, we scale the weights so that the largest eigenvalue is less than 1. A large square matrix is generated with the number of rows selected, ultimately with uniform probability. This square matrix is then multiplied by a binary mask, resulting in a sparse matrix, as a part of the depth rewiring algorithm that we mentioned before. This algorithm achieves the goal of maintaining the level of sparse connectivity in the network by dynamically disconnecting some synapses while

reconnecting others. In this algorithm, we set the temperature parameter to 0 and the L1-norm regularization parameter to 0.01.

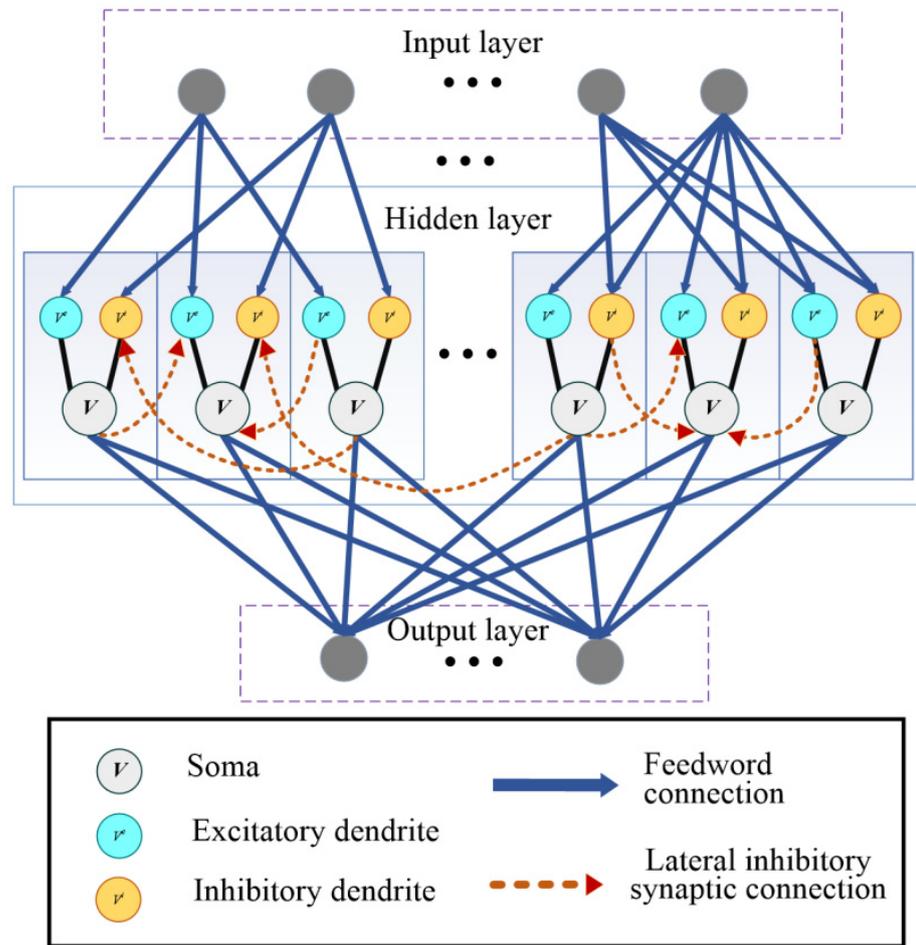


Figure 2. Network architecture for learning and memory integrated with the proposed SAM model. This network architecture is comparable to a 2-layer network of point neurons. The soma and dendrites of different neurons in the hidden layer are connected to lateral inhibitory synapses randomly. The gray circles in the input layer and output layer are not SAM neurons, representing the input spiking neuron and output spiking neuron, respectively. The input and output encodings are determined for different tasks, which will be described in the section of experimental results.

2.2. BPTT Training Algorithm

In common ANN models, the gradients of the loss function are obtained with respect to the weights in the network using back propagation. Nevertheless, the training method of back propagation cannot be directly applied to SNNs due to the non-differentiability of spikes from SNNs. Providing that time is discretized, the gradient needs to be propagated through continuous time or multiple time steps. To enable the SNN model to learn in the training process, we use a pseudo-derivative technique as shown below

$$\frac{dz_j(t)}{dv_j(t)} = k \max\{0, 1 - |v_j(t)|\} \tag{5}$$

where $k = 0.3$ (typically less than 1) is a constant value that can dampen the increase in back propagated errors through spikes by using a pseudo-derivative of amplitude to achieve the goal of stable performance. The variable $z_j(t)$ represents the spike train of neuron j that

assumes values in $\{0, 1\}$. The variable $v_j(t)$ represents the normalized membrane potential, which is defined as follows

$$v_j(t) = \frac{V_j(t) - \Gamma_j(t)}{\Gamma_j(t)} \tag{6}$$

where Γ_j represents the firing rate of neuron j . With the purpose of providing the self-learning capability required for reinforcement learning for the proposed SAM model, we utilize a proximal policy optimization algorithm [31]. This algorithm is easy to implement and allows the model to have self-learning capabilities. The clipped surrogate objective of this algorithm is defined as $O^{PPO}(\vartheta_{old}, \vartheta, t, k)$. Therefore, the loss function with respect to ϑ is formulated as

$$L_P(\vartheta) = -\frac{\sum_{k < K} \sum_{t < T} O^{PPO}(\vartheta_{old}, \vartheta, t, k)}{KT} + \mu_f \frac{1}{n} \sum_j \left\| \frac{\sum_{k,t} z_j(t, k) - f^0}{KT} \right\|^2 \tag{7}$$

where f^0 represents a target firing rate of 10 Hz and μ_f represents a regularization hyper-parameter. Variables t and k represent the simulation time step and the total number of epochs. The variable ϑ represents the current policy parameter, which is defined in the previous research [31]. In each iteration of training, $K = 10$ episodes of $T = 2000$ time steps are generated with a fixed parameter ϑ_{old} , which is the vector of policy parameters before the update as expressed in [31]. At the same time, the loss function $L(\vartheta)$ is minimized by the ADAM optimizer [32].

2.3. Minimum Error Entropy Criterion (MEEC)

The minimum error entropy (MEE) can minimize the entropy of the estimation error, so that decreases the uncertainty in the learning process. The α -order Renyi's entropy is used assuming a random variable e with probability density function $f^\alpha(e)$, which is defined as

$$H(e) \triangleq \frac{1}{1 - \alpha} \log \int f^\alpha(e) de \tag{8}$$

where α is set to 2 for 2-order Renyi's entropy in this study. The kernel density estimation (KDE) is used to estimate the PDF of the error samples, which has three advantages. First, it is a non-parameter approach, which does not require the prior knowledge of the error distribution. Second, it does not require the integration calculation. Third, it can be smooth and differentiable, which is vital for the gradient computation. Considering a set of i.i.d data $\{e_i\}_{i=1}^N$ drawn from the distribution, the KDE of the PDF can be formulated as

$$\hat{f}_E(e) = \frac{1}{N} \sum_{i=1}^N G_\Sigma(e - e_i) \tag{9}$$

where $G_\Sigma(e - e_i)$ represents the Gaussian function with the following expression as

$$G_\Sigma(e - e_i) = \frac{1}{\sqrt{2\pi(\det \Sigma)}} \cdot \exp\left(-\frac{1}{2}(e - e_i)^T \Sigma^{-1}(e - e_i)\right) \tag{10}$$

where N and Σ represent the number of the data points and the kernel parameter, respectively. In this research, Σ represents a diagonal matrix with the s -th diagonal element with

the variance δ_s^2 for e_s in e , where $s = 1, 2, \dots, S$. The kernel parameter represents a free parameter. Thus, the Renyi's quadratic entropy can be expressed as

$$\begin{aligned}
 H_2(e) &= -\log \int \left(\frac{1}{N} \sum_{i=1}^N G_{\Sigma}(e - e_i) \right)^2 de \\
 &= -\log \frac{1}{N^2} \int \left(\sum_{i=1}^N \sum_{j=1}^N G_{\Sigma}(e - e_i) G_{\Sigma}(e - e_j) \right) de \\
 &= -\log \frac{1}{N^2} \int \left(\sum_{i=1}^N \sum_{j=1}^N G_{\Sigma}(e - e_i) G_{\Sigma}(e - e_j) \right) de \tag{11} \\
 &= -\log \frac{1}{N^2} \left(\sum_{i=1}^N \sum_{j=1}^N G_{\sqrt{2}\Sigma}(e_i - e_j) \right) \\
 &= -\log \frac{1}{N^2} \left(\sum_{i=1}^N \sum_{j=1}^N G_{\Sigma_2}(e_i - e_j) \right)
 \end{aligned}$$

Based on the Formula (11), we define a function $V(e)$ to represent the information potential of variable e , which is formulated as

$$V(e) = \frac{1}{N^2} \left(\sum_{i=1}^N \sum_{j=1}^N G_{\Sigma_2}(e_i - e_j) \right) \tag{12}$$

Therefore, the minimization of the Renyi's entropy $H_2(e)$ means the maximization of the information potential $V(e)$ because of the monotonic increasing feature of the log function. The Parzen window is used to decrease the computational complexity and the instantaneous information potential at time t , which can be formulated as

$$J_1(e) = \frac{1}{W} \sum_{i=k-W+1}^k G_{\Sigma_2}(e_k - e_i) \tag{13}$$

where W represents the length of the Parzen window. It should be noted that MEE is a kind of local optimization criterion but suffers from the shift-invariant problem. It can only determine the location of error PDF but cannot know the distribution location. The function $G_{\Sigma_2}(\cdot)$ can be defined as the Gaussian kernel function with bandwidth σ

$$G_{\Sigma_2}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{14}$$

In order to reduce the computational complexity, quantization technique is used to realize the quantized MEE (QMEE). Thus, the information potential is expressed as

$$V^Q(e) = \frac{1}{N^2} \left(\sum_{i=1}^N \sum_{j=1}^N G_{\Sigma_2}(e_i - Q|e_j|) \right) = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^M \varphi_j G_{\Sigma_2}(e_i - c_j) \tag{15}$$

where $Q[\cdot]$ represents a quantization operator mapping each $\{e_i\}_{i=1}^N$ to one of $\{c_j\}_{j=1}^M$, resulting in a codebook $C = (c_1, c_2, c_3, \dots, c_M)$. $\Phi = (\varphi_1, \varphi_2, \dots, \varphi_M)$ represents the number of the samples quantized to the corresponding set $\{c_j\}_{j=1}^M$. It should be noted that $\sum_{j=1}^M \varphi_j = N$. Theoretical proof of the robustness has been presented in [22].

2.4. Restricted MEEC

In this study, the fundamental inner product to measure the similarity is used, which is generalized from its vectors' application [33]. The inner product similarity between continuous pdfs $f_X(x)$ and $g_X(x)$ can be expressed as

$$\langle f_X(x), g_X(x) \rangle = \int_X f_X(x)g_X(x)dx \tag{16}$$

The desired distribution $\rho_E(e)$, which is expressed in [33] in detail, can be defined as follows

$$\rho_E(e) = \begin{cases} \zeta_0, & e = 0 \\ \zeta_{-1}, & e = -1 \\ \zeta_1, & e = 1 \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

where ζ_i ($i = 0, -1, 1$) denotes the corresponding density for each peak, which is simplified into a Dirac- δ function.

The maximization of the similarity measure between the error pdf $f_E(e)$ and the desired distribution $\rho_E(e)$ can be formulated as

$$\begin{aligned} & \max \langle f_E(e), \rho_E(e) \rangle \\ & \Leftrightarrow \max \int_X f_E(e)\rho_E(e)dx \\ & \Leftrightarrow \max \zeta_0 f_E(0) + \zeta_{-1} f_E(-1) + \zeta_1 f_E(1) \end{aligned} \tag{18}$$

Furthermore, the model parameter can be expressed as

$$\begin{aligned} w^* &= \operatorname{argmax} \zeta_0 \hat{f}_E(0) + \zeta_{-1} \hat{f}_E(-1) + \zeta_1 \hat{f}_E(1) \\ &= \operatorname{argmax} \begin{pmatrix} \zeta_0 \frac{1}{N} \sum_{i=1}^N G_{\Sigma 2}(0 - e_i) \\ + \zeta_{-1} \frac{1}{N} \sum_{i=1}^N G_{\Sigma 2}(-1 - e_i) \\ \zeta_1 \frac{1}{N} \sum_{i=1}^N G_{\Sigma 2}(1 - e_i) \end{pmatrix} \\ &= \operatorname{argmax} \frac{1}{N^2} \sum_{i=1}^N \begin{pmatrix} N\zeta_0 G_{\Sigma 2}(e_i) \\ + N\zeta_{-1} G_{\Sigma 2}(e_i + 1) \\ + N\zeta_1 G_{\Sigma 2}(e_i - 1) \end{pmatrix} \end{aligned} \tag{19}$$

In fact, QMEE converges the prediction errors $\{c_j\}_{j=1}^M$ to obtain a compact error distribution. Based on the method in [33], a predetermined codebook $C = (0, -1, 1)$ implements QMEE to restrict errors to three positions and avoid the undesirable double-peak learning consequence. Therefore, the restricted MEE (RMEE) algorithm can be formulated as

$$V^R(e) = \frac{1}{N^2} \sum_{i=1}^N \begin{pmatrix} \varphi_0 G_{\Sigma 2}(e_i) \\ + \varphi_{-1} G_{\Sigma 2}(e_i + 1) \\ + \varphi_1 G_{\Sigma 2}(e_i - 1) \end{pmatrix} \tag{20}$$

where $\Phi = (\varphi_0, \varphi_{-1}, \varphi_1) = (N\zeta_0, N\zeta_{-1}, N\zeta_1)$ that represents the corresponding number for each quantization word $C = (0, -1, 1)$. The proposed RMEE algorithm maximizes the inner product similarity between error pdf $f_E(e)$ and the optimal three-peak distribution $\rho_E(e)$. RMEE is a specific formation of QMEE where the codebook is predetermined as $C = (0, -1, 1)$ and converges learning errors on these three locations.

In order to optimize Equation (19), the half-quadratic technique is used to solve optimization issues. A convex function $g(x) = -x \log(-x) + x$ is defined, and the information potential can be expressed as

$$V^R(e) = \sum_{i=1}^N \left(\begin{array}{l} \varphi_0 \left\{ u_i \frac{e_i^2}{2\sigma^2} - g(u_i) \right\} \\ + \varphi_{-1} \left\{ v_i \frac{(e_i+1)^2}{2\sigma^2} - g(v_i) \right\} \\ + \varphi_1 \left\{ s_i \frac{(e_i-1)^2}{2\sigma^2} - g(s_i) \right\} \end{array} \right) \triangleq J_{R1}(w, u_i, v_i, s_i) \quad (21)$$

In half-quadratic technique, it has the following relationship

$$\begin{aligned} u_i^k &= -\exp\left(-\frac{e_i^2}{2\sigma^2}\right) < 0 \\ v_i^k &= -\exp\left(-\frac{(e_i+1)^2}{2\sigma^2}\right) < 0 \\ s_i^k &= -\exp\left(-\frac{(e_i-1)^2}{2\sigma^2}\right) < 0 \\ &(i = 1, 2, \dots, N). \end{aligned} \quad (22)$$

By attaining the optimal (u_i^k, v_i^k, s_i^k) in the k th iteration, the information potential can be formulated as

$$V^R(e) = \sum_{i=1}^N \left(\begin{array}{l} \varphi_0 u_i (t_i - y_i)^2 \\ + \varphi_{-1} v_i (t_i + 1 - y_i)^2 \\ + \varphi_1 s_i (t_i - 1 - y_i)^2 \end{array} \right) \triangleq J_{R2}(w) \quad (23)$$

The $J_{R2}(w)$ can be optimized based on gradient-based methods because the objective function is differentiable and continuous. For example, the gradient of $J_{R2}(w)$ can be expressed as

$$\frac{\partial}{\partial w} J_{R2}(w) = \sum_{i=1}^N \left(\begin{array}{l} \varphi_0 u_i \frac{\partial(t_i - y_i)^2}{\partial w} \\ + \varphi_{-1} v_i \frac{\partial(t_i + 1 - y_i)^2}{\partial w} \\ + \varphi_1 s_i \frac{\partial(t_i - 1 - y_i)^2}{\partial w} \end{array} \right) = -2 \sum_{i=1}^N \left(\begin{array}{l} \varphi_0 u_i e_i \\ + \varphi_{-1} v_i (e_i + 1) \\ + \varphi_1 s_i (e_i - 1) \end{array} \right) x_i y_i (1 - y_i) \quad (24)$$

The detailed algorithm of the HQ-based optimization and its convergence analysis for RMEE are presented in [33].

3. Results

3.1. Proposed Network with RMEE Criterion

Since MEE has the shift-invariant feature, and estimation results based on MEEC will not always converge to the true value. A consideration is to combine the RMEE criterion with CEE for a global optimal solution. The cross-entropy loss function, also regarded as log loss, is the most commonly used loss function for back propagation. The cross-entropy loss function increases as the predicted probability deviates from the actual label, and can be described as follows

$$L_{ce}(\hat{y}_i, y_i) = -\sum_i y_i \log(\hat{y}_i) \quad (25)$$

In this paper, the label l^n of each image is used, which is only assumed to be 1 for images belonging to the same class of images during testing, and 0 otherwise. The cross-entropy formula can be expressed as

$$J_2 = \sum_{n=1}^5 -l^n \log \sigma(y^{20+20 \cdot n}) - (1 - l^n) \log(1 - \sigma^{20+20 \cdot n}) \quad (26)$$

where the output of the SNN model is only counted after all images are fully rendered. Therefore, for the novel criterion, the performance index can be formulated as

$$J_k(e) = \mu \left[\sum_{i=1}^N \left(\begin{array}{l} \varphi_0 u_i (t_i - y_i)^2 \\ + \varphi_{-1} v_i (t_i + 1 - y_i)^2 \\ + \varphi_1 s_i (t_i - 1 - y_i)^2 \end{array} \right) \right] + (1 - \mu) \left[\sum_{n=1}^5 \left(\begin{array}{l} -l^n \log \sigma(y^{20+20 \cdot n}) \\ -(1 - l^n) \log(1 - \sigma^{20+20 \cdot n}) \end{array} \right) \right] \quad (27)$$

where μ represents a weighting constant. In the supervised learning tasks, there only exist cross-entropy and RMEE, which is described in Equation (27).

3.2. Autonomous Navigation

We first apply the proposed SNN model in the agent navigation task, which requires the network to have reinforcement learning capabilities. The agent needs to learn to find objects in a 2D area and eventually be able to navigate to find objects at random locations in the area. This task is interrelated with the neuroscience paradigm of the well-known Morris water maze task, which is designed to study learning in the brain [34]. In this task, a virtual agent is simulated as a point in the 2D simulation arena and is controlled by the proposed SNN model. The position of the agent is configured randomly with a uniform probability in the overall arena at the beginning of an episode. The agent produces a small velocity vector of the Euclidean norm and selects an action at each time step. It receives a reward value '1' after reaching the destination.

In the navigation task, the information $s(t)$ of the current environment state and the reward score $r(t)$ are received as input data by neurons in the input layer at each time step. The coordinate information of the position is encoded by the input neurons through the Gaussian population rate encoding method. Furthermore, each neuron in the input layer is assigned a coordinate value with a firing rate, which is defined as: $r_{\max} = \exp(-100(\zeta_i - \zeta)^2)$, where ζ_i and ζ represent the actual coordinate value and the preferred coordinate value, respectively. r_{\max} is supposed to be set as 500 Hz. Moreover, the instantaneous reward $r(t)$ is encoded by two sets of input neurons. In the first group, the neurons generate spikes in sync when a positive reward is received, while in the second group, the neurons generate spikes as long as the proposed SNN model receives a negative reward. The output of the network is represented by five readout neurons in the output layer with membrane potential $\lambda_i(t)$. The action vector $\zeta(t) = (\zeta_x(t), \zeta_y(t))^T$ is used to determine the movement of the agent in the navigation task that we mentioned before. It is calculated from a Gaussian distribution with mean $\mu_x = \tanh(\lambda_1(t))$ and $\mu_y = \tanh(\lambda_2(t))$ as well as variances $\Phi_x = \sigma(\lambda_3(t))$ and $\Phi_y = \sigma(\lambda_4(t))$. In the end, the output of the last readout neuron λ_5 is calculated to predict the value function $\mu_\theta(t)$. This predicts the expected discounted sum of future rewards $\Omega(t) = \sum_{t' > t} \gamma^{t-t'} \omega(t')$, where $\omega(t')$ represents the reward at time t' and γ represents the discount factor, whose value is usually 0.99.

The agent based on the proposed SNN model learns to learn in the navigation task towards the correct destination location after the meta-learning process. The overall training process in the reward learning process is described by Algorithm 1. We add other loss functions to support the reinforcement learning framework, maintaining the loss function consistent with Equation (26). Figure 3 shows the successful destination reached number (DRN) per learning iteration. Each iteration contains a batch of ten episodes, and network weights are updated during the navigation task. For each episode, the model is expected to explore until reaching and storing the destination location, and uses the prior knowledge to find the shortest path to the destination. This reveals that the proposed SNN model has meta-learning capability in the autonomous navigation task.

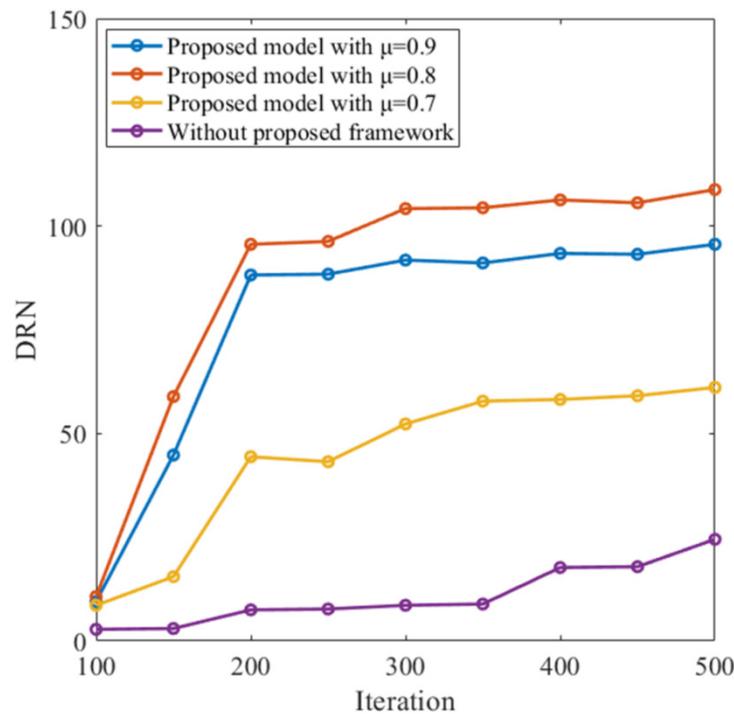


Figure 3. Navigation performance of the proposed model with different settings.

Algorithm 1 Training process in the reward learning process

Input: number of full episodes K , timesteps T , fixed parameters θ_{old} , target firing rate f^0 , regularization hyper-parameters $\mu_v, \mu_e, \mu_{firing}$, bandwidth σ , predicted value function $V_\theta(t, k)$ and sum of future rewards $R(t, k)$

Output: total loss L_θ .

1. Parameters setting: $f^0, \mu_v, \mu_e, \mu_{firing}$ and σ .
2. **for** n **in** batch size N :
3. Set $e_n = R(t, k) - V_\theta(t, k)$
4. **if** number of iteration is 0:
5. $(\varphi_0, \varphi_{-1}, \varphi_1) = (N, 0, 0)$
6. **else:**
7. $(\varphi_0, \varphi_{-1}, \varphi_1) = (\#\{e_n \in (-0.5, 0.5)\},$
 $\#\{e_n \in (-1, -0.5)\},$
 $\#\{e_n \in (0.5, 1)\})$
8. where $\#\{\cdot\}$ indicates counting the samples that satisfy the condition
9. $(u_n, v_n, s_n) = (-\exp(-\frac{e_n^2}{2\sigma^2}), -\exp(-\frac{(e_n+1)^2}{2\sigma^2}), -\exp(-\frac{(e_n-1)^2}{2\sigma^2}))$
10. $L_n^{RMEE} = \varphi_0 u_n e_n^2 + \varphi_{-1} v_n (e_n + 1)^2 + \varphi_1 s_n (e_n - 1)^2$
11. **end for**
12. **for** k **in** K :
13. **for** t **in** T :
14. $L_{(t,k)}^{PPO} = O^{PPO}(\theta_{old}, \theta, t, k)$
15. **end for**
16. **end for**
17. Calculate the total loss:
18. $L(e) = L_p(e) + J_k(e)$
19. **return** $L(e)$

3.3. Working Memory Performance on Store–Recall Task with Non-Gaussian Noise

To further demonstrate the robust working memory capability of the proposed SNN model, we apply the model in a store–recall task with non-Gaussian noise. The detailed

settings of the store–recall task have been previously presented in [35]. The SNN model receives a sequence of frames that are represented by ten spike trains in a period of time. The inputs #1 and #2 are represented by the spiking activities of input neurons from #1 to #10 and from #11 to #20, respectively. As shown in Figure 4, the neurons from #21 to #30 and from #31 to #40 receive the random store and recall commands, respectively. The store command means direct attention is paid to the specific frame of input data flow. Then, this frame will be reproduced when receiving the recall command. Figure 4 shows one test example with the spiking activities after working memory training. The dynamic threshold changes along with the learning procedure, which is shown in Figure 4. This reveals that the proposed SNN model can exhibit the working memory performance and realize the store–recall task successfully. Since working memory is a vital feature and the foundation for meta-learning, this also suggests that the MeMEE model can exhibit the meta-learning tasks based on its working memory mechanisms with a robust performance.

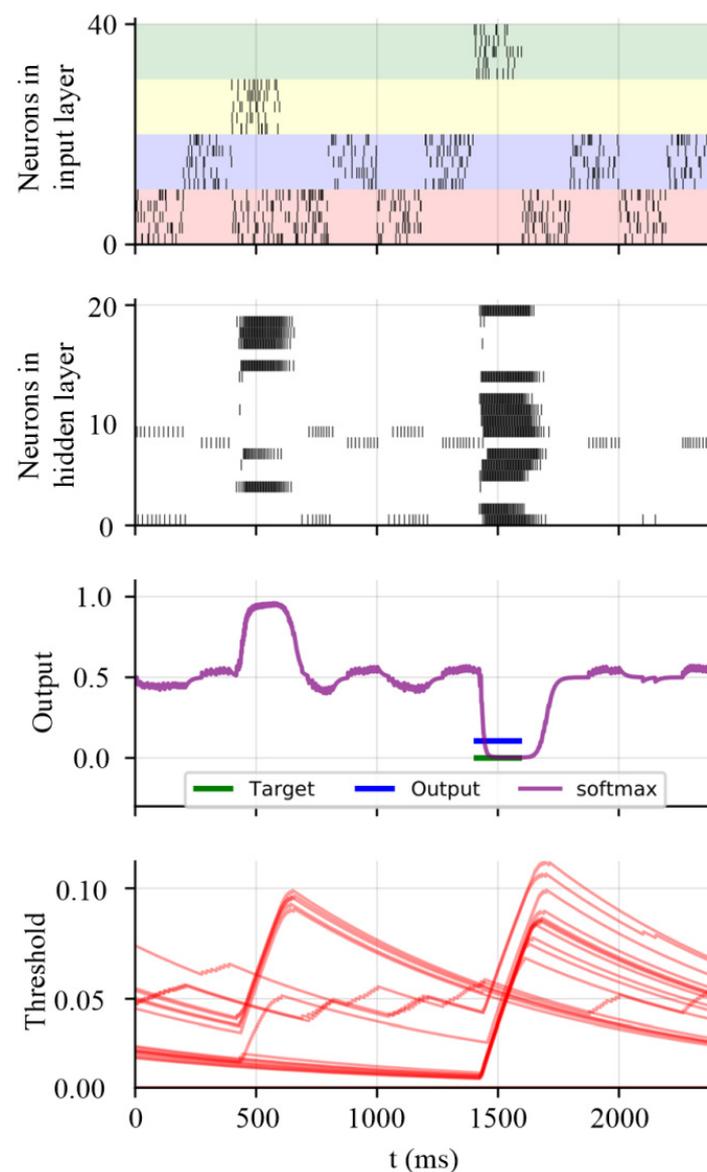


Figure 4. Working memory capability of the proposed SNN model after training.

3.4. Meta-Learning Performance on Sequential MNIST Data Set with Non-Gaussian Noise

We further demonstrate the meta-learning capability of the proposed SNN model in a transfer learning task based on the sequential MNIST (sMNIST) data set. We divide the

sMNIST data set into two parts. The first part includes 30,000 images for digits '0', '1', '2', '3', and '4', and the second part includes 30,000 patterns for digits '5', '6', '7', '8', and '9'. In the first phase, the first part is employed to train the SNN model, and the second part is then used for training. In the second phase, 10% salt and pepper noise is added to the testing data set as the non-Gaussian noise for the performance evaluation. Figure 5 shows the performance of the MeMEE model and compares it with other counterpart models, including recurrent SNN (RSNN) and the conventional LIF-based SNN model without the RMEE criterion. This shows that the proposed model outperforms the other solutions, and the reasoning behind this includes three points. Firstly, the proposed model has the meta-learning capability, so it can illustrate the transfer learning capability, and its transfer learning performance is superior to the RSNN model accordingly, considering accuracy and convergence speed. Secondly, due to the RMEE criterion being the loss function, its robustness to the non-Gaussian noise is superior to the model without the RMEE criterion in terms of the learning accuracy. The result suggests that the MeMEE model with RMEE criterion has a more powerful robust meta-learning capability in learning sequential spatio-temporal patterns.

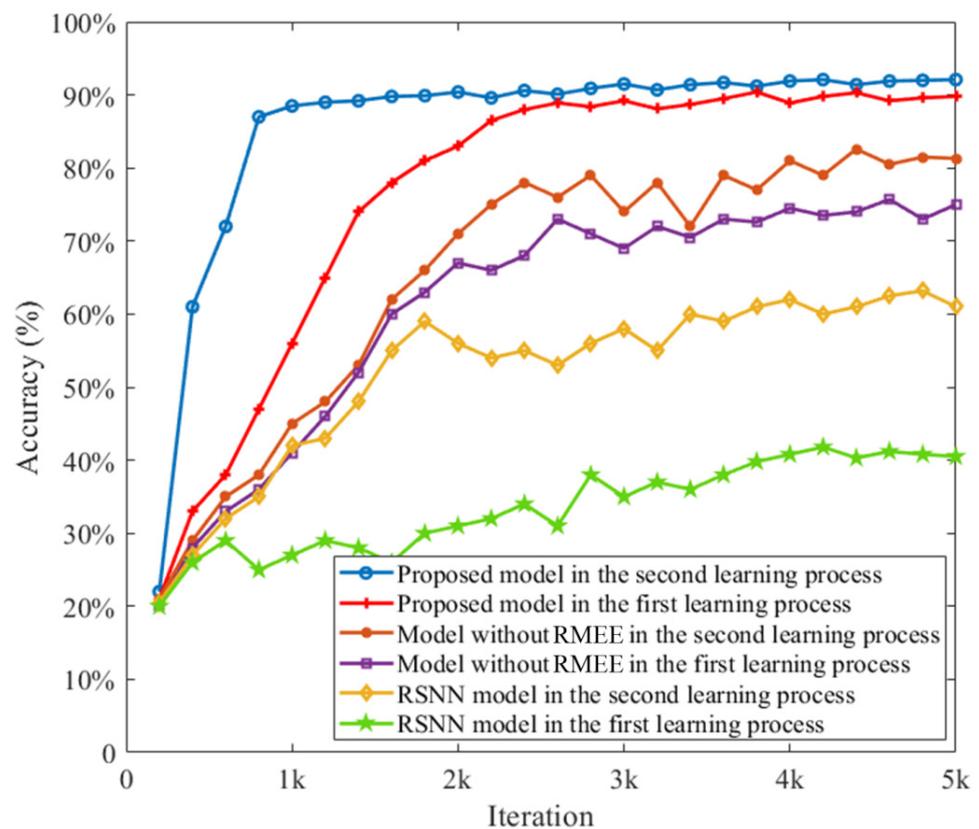


Figure 5. Meta-learning capability of the proposed MeMEE model on sequential MNIST data set.

3.5. Effects of Loss Parameters on Learning Performance

In this study, we further investigate how each loss function affects the learning performance of the proposed MeMEE model. We use the sMNIST data set to evaluate and quantify the learning accuracy along with the changing loss parameter. In order to demonstrate the learning robustness based on the proposed MeMEE model, salt and pepper noise is added to the sMNIST data set. Different levels are considered, which are selected from 3.19% to 19.13%. Different values of parameter μ are investigated, which are set from 0.3 to 1.0. As shown in Figure 6, the value of μ with 0.7, 0.8, and 0.9 can induce the higher learning accuracy on sequential visual recognition. This reveals that the RMEE criterion can further enhance the robustness of the proposed MeMEE model without the RMEE

criterion, i.e., $\mu = 1$. Since the model without RMEE criterion with 3.19% non-Gaussian noise only reaches 83.6% accuracy, the RMEE criterion can improve the learning accuracy of the proposed MeMEE model with non-Gaussian salt and pepper noise.

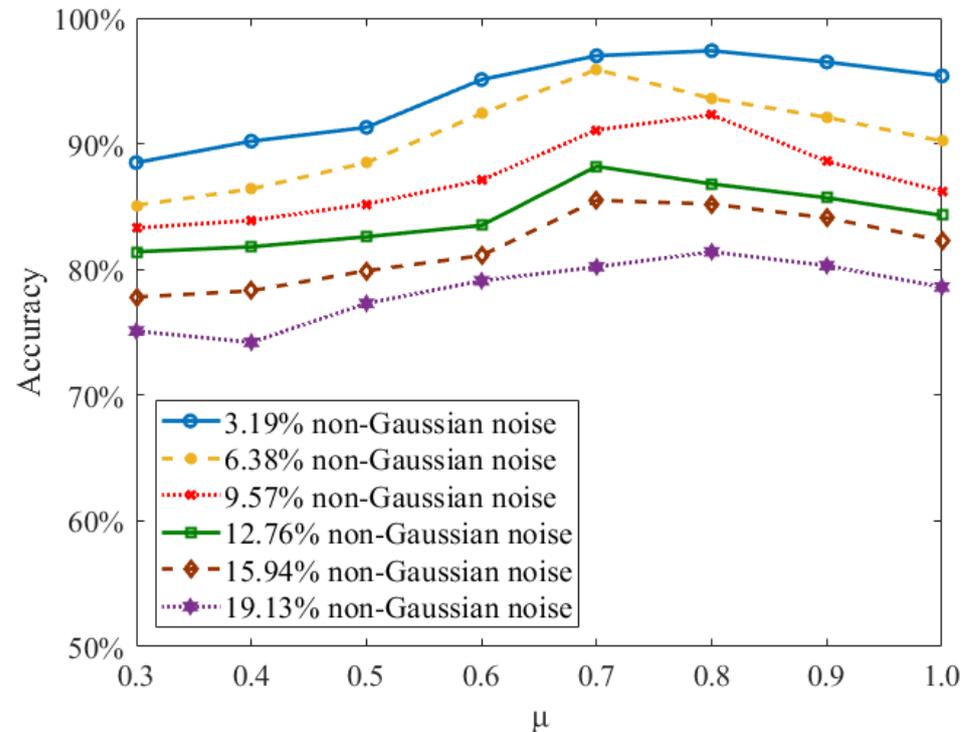


Figure 6. Effects of loss parameters on the learning performance of sequential classification.

4. Discussion

This paper presents an information theoretic learning framework for robust spike-driven continual meta-learning. Different from the previous SNN learning research, we first introduce the RMEE criterion to develop and improve the spike-based learning framework, which is significantly general and can also provide a series of theoretic insights. Moreover, the information theoretic framework allows us to obtain a direct understanding and better interpretation of the robust learning solutions of SNN models, compared with some previous studies focusing on improving the learning robustness of SNNs [36].

As a first step in establishing a rigorous framework for SNN continual meta-learning with RMEE, the presented research can be extended in both theoretical and practical aspects. From the theoretical point of view, one extension is to use the information potential to train the presented SNN model. For example, as shown in [37], Chen et al. presented a survival information potential algorithm for adaptive system training. This does not require computing of the kernel function and has good robustness performance accordingly. The other extension is to apply the proposed framework in other spike-based learning paradigms, including few-shot learning, multitask learning, and unsupervised learning [38].

From a practical point of view, the model is expected to be implemented on neuromorphic platforms to realize low-power and real-time systems for various types of applications. The state-of-the-art digital neuromorphic systems include Loihi [12], Tianjic [11], BiCoSS [13], Cerebellumorphic [14], LaCSNN [15], TrueNorth [39], and SpiNNaker [40]. By implementing embedded neuromorphic systems, it can be applied in different fields such as edge computing devices, brain-machine integration systems, and intelligent systems [41–43].

5. Conclusions

In this invited paper, we first presented an ITL-based scheme for robust spike-based continual meta-learning, which is improved by the RMEE criterion. A gradient descent learning principle is presented in a recurrent SNN architecture. Several tasks are realized to demonstrate the learning performance of the proposed MeMEE model, including autonomous navigation, robust working memory in the store–recall task and robust meta-learning capability for the sMNIST data set. In the first autonomous navigation task, the SNN model learns to find the correct destination by continual meta-learning from the task reward and punishment. This demonstrates that the MeMEE model based on the proposed RMEE criterion realizes the meta-learning capability for navigation and outperforms the conventional RSNN model. In the second task, the proposed MeMEE model improves the working memory performance by recalling the stored noisy patterns. In the third task, the proposed MeMEE model with RMEE criterion can enhance the robustness in the meta-learning task for noisy sMNIST images. This invited paper provides a novel insight into the improvement of the spike-based machine learning performance based on information theoretic learning strategy, which is critical for the further research of artificial general intelligence. In addition, it can be implemented by the low-power neuromorphic system, which can be applied in edge computing of internet of things (IoT) and unmanned systems.

Author Contributions: S.Y. and B.C. contributed to the conceptualization, methodology, and writing of this paper. J.T. helped to conduct the experiment. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded partly by the National Natural Science Foundation of China with grant numbers (Grant No. 62006170, No. 62088102, No. U21A20485) and partly by China Postdoctoral Science Foundation (Grant Nos. 2020M680885, 2021T140510).

Acknowledgments: We would like to thank the editor and reviewer for their comments on the manuscript.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 89–94. [[CrossRef](#)]
2. Parisi, G.I.; Kemker, R.; Part, J.L.; Kanan, C.; Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Netw.* **2019**, *113*, 54–71. [[CrossRef](#)] [[PubMed](#)]
3. Yao, H.; Zhou, Y.; Mahdavi, M.; Li, Z.; Socher, R.; Xiong, C. Online structured meta-learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6779–6790.
4. Javed, K.; White, M. Meta-learning representations for continual learning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 172.
5. Serrà, J.; Surís, D.; Miron, M.; Karatzoglou, A. Overcoming catastrophic forgetting with hard attention to the task. In Proceedings of the International Conference on Machine Learning (PMLR 80), Stockholm, Sweden, 10–15 July 2018; pp. 4548–4557.
6. Zeng, G.; Chen, Y.; Cui, B.; Yu, S. Continual learning of context-dependent processing in neural networks. *Nat. Mach. Intell.* **2019**, *1*, 364–372. [[CrossRef](#)]
7. van de Ven, G.M.; Siegelmann, H.T.; Tolias, A.S. Brain-inspired replay for continual learning with artificial neural networks. *Nat. Commun.* **2020**, *11*, 4069. [[CrossRef](#)]
8. Tavanaei, A.; Ghodrati, M.; Kheradpisheh, S.R.; Masquelier, T.; Maida, A. Deep learning in spiking neural networks. *Neural Netw.* **2019**, *111*, 47–63. [[CrossRef](#)]
9. Lee, C.; Panda, P.; Srinivasan, G.; Roy, K. Training deep spiking convolutional neural networks with stdp-based unsupervised pre-training followed by supervised fine-tuning. *Front. Neurosci.* **2018**, *12*, 435. [[CrossRef](#)]
10. Xia, Q.; Yang, J.J. Memristive crossbar arrays for brain-inspired computing. *Nat. Mat.* **2019**, *18*, 309–323. [[CrossRef](#)]
11. Pei, J.; Deng, L.; Song, S.; Zhao, M.; Zhang, Y.; Wu, S.; Wang, G.; Zou, Z.; Wu, Z.; He, W.; et al. Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature* **2019**, *572*, 106–111. [[CrossRef](#)]

12. Davies, M.; Srinivasa, N.; Lin, T.-H.; China, G.; Cao, Y.; Choday, S.H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* **2018**, *38*, 82–99. [[CrossRef](#)]
13. Yang, S.; Wang, J.; Hao, X.; Li, H.; Wei, X.; Deng, B.; Loparo, K.A. BiCoSS: Toward large-scale cognition brain with multigranular neuromorphic architecture. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *11*, 1–15. [[CrossRef](#)] [[PubMed](#)]
14. Yang, S.; Wang, J.; Zhang, N.; Deng, B.; Pang, Y.; Azghadi, M.R. Cerebellumorphic: Large-scale neuromorphic model and architecture for supervised motor learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *23*, 1–15. [[CrossRef](#)] [[PubMed](#)]
15. Yang, S.; Wang, J.; Deng, B.; Liu, C.; Li, H.; Fietkiewicz, C.; Loparo, K.A. Real-time neuromorphic system for large-scale conductance-based spiking neural networks. *IEEE Trans. Cybern.* **2019**, *49*, 2490–2503. [[CrossRef](#)]
16. Bellec, G.; Salaj, D.; Subramoney, A.; Legenstein, R.; Maass, W. Long short-term memory and learning-to-learn in networks of spiking neurons. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 247.
17. Li, Y.; Zhou, J.; Tian, J.; Zheng, X.; Tang, Y.Y. Weighted error entropy-based information theoretic learning for robust subspace representation. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *19*, 1–15. [[CrossRef](#)]
18. Chen, J.; Song, L.; Wainwright, M.; Jordan, M. Learning to explain: An information-theoretic perspective on model interpretation. In Proceedings of the 35th International Conference on Machine Learning (PMLR 80), Stockholm, Sweden, 10–15 July 2018; pp. 883–892.
19. Xu, Y.; Cao, P.; Kong, Y.; Wang, Y. DMI: A novel information-theoretic loss function for training deep nets robust to label noise. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 76.
20. Chen, B.; Xing, L.; Zhao, H.; Du, S.; Principe, J.C. Effects of outliers on the maximum correntropy estimation: A robustness analysis. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 4007–4012. [[CrossRef](#)]
21. Chen, B.; Li, Y.; Dong, J.; Lu, N.; Qin, J. Common spatial patterns based on the quantized minimum error entropy criterion. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 4557–4568. [[CrossRef](#)]
22. Chen, B.; Xing, L.; Xu, B.; Zhao, H.; Principe, J.C. Insights into the robustness of minimum error entropy estimation. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 731–737. [[CrossRef](#)]
23. Chen, H.-Y.; Liang, J.-H.; Chang, S.-C.; Pan, J.-Y.; Chen, Y.-T.; Wei, W.; Juan, D.-C. Improving adversarial robustness via guided complement entropy. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 4880–4888.
24. Rachdi, M.; Waku, J.; Hazgui, H.; Demongeot, J. Entropy as a robustness marker in genetic regulatory networks. *Entropy* **2020**, *22*, 260. [[CrossRef](#)] [[PubMed](#)]
25. Borin, J.A.M.S.; Humeau-Heurtier, A.; Virgilio Silva, L.E.; Murta, L.O. Multiscale entropy analysis of short signals: The robustness of fuzzy entropy-based variants compared to full-length long signals. *Entropy* **2021**, *23*, 1620. [[CrossRef](#)] [[PubMed](#)]
26. Grienberger, C.; Milstein, A.D.; Bittner, K.C.; Romani, S.; Magee, J.C. Inhibitory suppression of heterogeneously tuned excitation enhances spatial coding in CA1 place cells. *Nat. Neurosci.* **2017**, *20*, 417–426. [[CrossRef](#)]
27. Muñoz, W.; Tremblay, R.; Levenstein, D.; Rudy, B. Layer-specific modulation of neocortical dendritic inhibition during active wakefulness. *Science* **2017**, *355*, 954–959. [[CrossRef](#)]
28. Poleg-Polsky, A.; Ding, H.; Diamond, J.S. Functional compartmentalization within starburst amacrine cell dendrites in the retina. *Cell Rep.* **2018**, *22*, 2898–2908. [[CrossRef](#)]
29. Ranganathan, G.N.; Apostolides, P.F.; Harnett, M.T.; Xu, N.L.; Druckmann, S.; Magee, J.C. Active dendritic integration and mixed neocortical network representations during an adaptive sensing behavior. *Nat. Neurosci.* **2018**, *21*, 1583–1590. [[CrossRef](#)]
30. Bellec, G.; Kappel, D.; Maass, W.; Legenstein, R. Deep rewiring: Training very sparse deep networks. *arXiv* **2017**, arXiv:1711.05136.
31. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
32. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
33. Li, Y.; Chen, B.; Yoshimura, N.; Koike, Y. Restricted minimum error entropy criterion for robust classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *2*, 1–14. [[CrossRef](#)] [[PubMed](#)]
34. Vasilaki, E.; Frémaux, N.; Urbanczik, R.; Senn, W.; Gerstner, W. Spike-based reinforcement learning in continuous state and action space: When policy gradient methods fail. *PLoS Comput. Biol.* **2009**, *5*, e1000586. [[CrossRef](#)]
35. Wolff, M.J.; Jochim, J.; Akyürek, E.G.; Stokes, M.G. Dynamic hidden states underlying working-memory-guided behavior. *Nat. Neurosci.* **2017**, *20*, 864–871. [[CrossRef](#)] [[PubMed](#)]
36. Yang, S.; Gao, T.; Wang, J.; Deng, B.; Lansdell, B.; Linares-Barranco, B. Efficient spike-driven learning with dendritic event-based processing. *Front. Neurosci.* **2021**, *15*, 601109. [[CrossRef](#)] [[PubMed](#)]
37. Chen, B.; Zhu, P.; Principe, J.C. Survival information potential: A new criterion for adaptive system training. *IEEE Trans. Signal Process.* **2012**, *60*, 1184–1194. [[CrossRef](#)]
38. Jiang, R.; Zhang, J.; Yan, R.; Tang, H. Few-shot learning in spiking neural networks by multi-timescale optimization. *Neural Comput.* **2021**, *33*, 2439–2472. [[CrossRef](#)]
39. DeBole, M.V.; Appuswamy, R.; Carlson, P.J.; Cassidy, A.S.; Datta, P.; Esser, S.K.; Garreau, G.J.; Holland, K.L.; Lekuch, S.; Mastro, M.; et al. Truenorth: Accelerating from zero to 64 million neurons in 10 years. *Computer* **2019**, *52*, 20–29. [[CrossRef](#)]
40. Furber, S.B.; Galluppi, F.; Temple, S.; Plana, L.A. The SpiNNaker project. *Proc. IEEE* **2014**, *102*, 652–665. [[CrossRef](#)]
41. Krestinskaya, O.; James, A.P.; Chua, L.O. Neuromemristive circuits for edge computing: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 4–23. [[CrossRef](#)]

-
42. Yoo, J.; Shoaran, M. Neural interface systems with on-device computing: Machine learning and neuromorphic architectures. *Curr. Opin. Biotechnol.* **2021**, *72*, 95–101. [[CrossRef](#)]
 43. Cho, S.W.; Kwon, S.M.; Kim, Y.; Park, S.K. Recent progress in transistor-based optoelectronic synapses: From neuromorphic computing to artificial sensory system. *Adv. Intell. Syst.* **2021**, *3*, 2000162. [[CrossRef](#)]