Neuromorphic Context-Dependent Learning Framework With Fault-Tolerant Spike Routing

Shuangming Yang, *Member, IEEE*, Jiang Wang, Bin Deng, *Senior Member, IEEE*, Mostafa Rahimi Azghadi, *Senior Member, IEEE*, and Bernabe Linares-Barranco, *Fellow, IEEE*

***Abstract*— Neuromorphic computing is a promising technology that realizes computation based on event-based spiking neural networks (SNNs). However, fault-tolerant on-chip learning remains a challenge in neuromorphic systems. This study presents the first scalable neuromorphic fault-tolerant context-dependent learning (FCL) hardware framework. We show how this system can learn associations between stimulation and response in two context-dependent learning tasks from experimental neuroscience, despite possible faults in the hardware nodes. Furthermore, we demonstrate how our novel fault-tolerant neuromorphic spike routing scheme can avoid multiple fault nodes successfully and can enhance the maximum throughput of the neuromorphic network by 0.9%–16.1% in comparison with previous studies. By utilizing the real-time computational capabilities and multiple-fault-tolerant property of the proposed system, the neuronal mechanisms underlying the spiking activities of neuromorphic networks can be readily explored. In addition, the proposed system can be applied in real-time learning and decision-making applications, brain–machine integration, and the investigation of brain cognition during learning.**

***Index Terms*— Brain inspired, context-dependent learning, fault tolerant, neuromorphic computing, spiking neural network (SNN).**

NOMENCLATURE

SNN	Spiking neural network.
FPGA	Field-programmable gate array.
LaCSNN	Large-scale conductance-based spiking neural network.
NoC	Network on chip.
STDP	Spike-timing-dependent plasticity.
LTD	Long-term depression.
SLM	Shift logic multiplier.
MFTN	Multiple-fault tolerant neuromorphic.
FCL	Fault-tolerant context-dependent learning.
mPFC	Medial prefrontal cortex.
AER	Address event representation.
LIF	Leaky integrate-and-fire.
LTP	Long-term potentiation.
WTA	Winner takes all.
BIST	Built-in self-test.
CDG	Component dependence graph.

Manuscript received 12 May 2020; revised 2 March 2021; accepted 19 May 2021. Date of publication 11 June 2021; date of current version 1 December 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62071324 and Grant 62006170, and in part by the China Postdoctoral Science Foundation under Grant 2020M680885. *(Corresponding author: Shuangming Yang.)*

Shuangming Yang, Jiang Wang, and Bin Deng are with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: yangshuangming@tju.edu.cn).

Mostafa Rahimi Azghadi is with the College of Science and Engineering, James Cook University, Townsville, QLD 4814, Australia (e-mail: mostafa.rahimiazghadi@jcu.edu.au).

Bernabe Linares-Barranco is with the Microelectronics Institute of Seville, 41092 Seville, Spain (e-mail: bernabe@imse-cnm.csic.es).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3084250.

Digital Object Identifier 10.1109/TNNLS.2021.3084250

I. INTRODUCTION

THE encoding and remembering of an event context rely on the episodic memory of the brain when observing an object or item [1]. The context can be an absolute time, relative time based on other events that happened before or after, or a specific place [2], [3]. Previous studies have shown that the interactions of neocortical and hippocampal circuits can enable contextual learning during an item-reward association task [4], [5].

The hippocampal firing activities are affected by the context in which a current task is performed [6]. In a sampling or an encoding phase of the task, the selectivity appears among the hippocampal CA1 neurons during a discrete delayed nonmatch-to-place task [7]. In addition, neural firing activities depend on the start or anticipated end location of a trajectory [8]. A previous experiment has revealed that neurons in the hippocampus develop selectivity toward specific items in an abstraction of spatial context [4]. The generation of behaviors based on the contextual representations depends on both the hippocampus and mPFC [9]. These studies reveal that context encoding may involve the interaction between the hippocampus and mPFC.

The experimental setup and preparation for the above-mentioned studies are usually time-prohibitive and involve interacting with live subjects. One approach to facilitate these experiments and improve our understanding of context-dependent learning is to build a computational model utilizing brain-inspired SNN models with neural spike representation. In this study, the concept of context-dependent learning is taken from the field of neuroscience. Here, context means the environmental sensory cues that are processed and learned in the hippocampus [52]–[57]. In previous work, Gulli *et al.* [52] studied monkeys to complete an associative memory task in a virtual environment for the investigation of the context-dependent representation of objects and space in the hippocampus. The context was defined by a texture applied to a maze wall. In addition, Zhao *et al.* [53] suggested that neurons in the hippocampus undergo context-dependent learning because they inherit different input patterns from presynaptic areas in different contexts, which occurs during decision-making and navigation tasks. Lee *et al.* [54] pointed out that navigation, context-dependent

learning, and episodic memory are produced in a recurrent collateral circuitry in the hippocampus. In addition, the mechanisms of context-dependent learning are explored in a series of neuroscience studies, using both behavioral paradigms and physiological observations [55]–[57]. The proposed model with the FCL framework can then be implemented on a neuromorphic architecture not only to better understand the brain but also to use it in various categories of applications, such as robotics and brain–computer interfaces.

Neuromorphic computing is a promising approach toward non-von Neumann systems for neuroscience and artificial intelligence applications, including the formulation of hypotheses regarding the function of neural systems, validation of self-consistency in the description of neural phenomena or function, neural computation instead of traditional computing structures, and biologically inspired engineering applications [10], [12], [13], [45]. Some of the large-scale neuromorphic systems used in these applications include BrainScaleS, TrueNorth, and SpiNNaker [10]–[13]. In addition to these large-scale systems, several other neuromorphic systems have been developed in the literature [14], [43], [47], [60]. We have previously developed LaCSNNs, which is a digital neuromorphic system designed for simulating SNNs using multicasting AER with 3-D NoC architecture [14]. Compared to state-of-the-art neuromorphic designs with similar capabilities, LaCSNN provides significant benefits in both biological accuracy and reconfigurability [14]. It is able to realize a large-scale SNN with one million biologically plausible neurons in real time.

When developing any large-scale neuromorphic system, such as those mentioned above, two main capabilities are required. These include online learning capability and fault-tolerant operation capability. It is also important to implement a system that integrates these two capabilities. This study focuses on implementing a neuromorphic system named FCL, for modeling large-scale SNNs with online FCL. It abstracts the mechanisms from both hippocampus and mPFC and realizes the learning capability in a context-dependent task responding to item reward. To the best of our knowledge, this article presents the first scalable FCL framework.

The remainder of this article is organized as follows. Section II introduces the fault-tolerance considerations in neuromorphic systems. The implemented network model for context-dependent learning is presented in Section III, while Section IV describes our digital neuromorphic architecture in detail. Section V proposes the fault-tolerant algorithm and methodology for the presented neuromorphic system. Experimental results of the digital neuromorphic system are presented in Section VI. Section VII discusses the advantages of the proposed neuromorphic model compared to state of the art. This article is concluded with a discussion on future works in Section VIII.

II. Fault-Tolerance Considerations in Neuromorphic Systems

State-of-the-art neuromorphic systems have used different architectures for the realization of SNNs, which are shown in Fig. 1. For the nonfault-tolerant neuromorphic systems, there are three conventional architectures, including shared bus, 2-D NoC, and 3-D NoC. The shared bus architecture can support both multicast and broadcast routing with low-cost SNN models, but it is constrained by its limited scalability [15], [16]. A number of studies have focused on the 2-D NoC architecture for neuromorphic systems, including H-NoC [17], Neurogrid [11], HiAER [13], and Truenorth [18]. The H-NoC architecture is based on the EMBRACE system using the LIF neuron model [17]. Analog implementation is used to realize the calculation of the ionic dynamics in Neurogrid and HiAER systems [11], [13], while a fully digital method is used in SpiNNaker, Truenorth, and Tianjic [10], [18], [19]. To enable the implementation of larger scale SNNs, 3-D NoC architecture is used in several works. In the first work, a multicast AER architecture is developed with biologically plausible conductance-based neuron models on the LaCSNN system, which has the intrinsic mechanisms underlying the neuronal spiking activities within large-scale multinucleus networks [14]. In other works, a multicast routing scheme is used in the 3-D mesh NoC architecture in KMCR [20], and multicompartment conductance-based neuron models are used in the IBFT-based CMN system, which can further enhance the system scalability in comparison with the previous works [21].

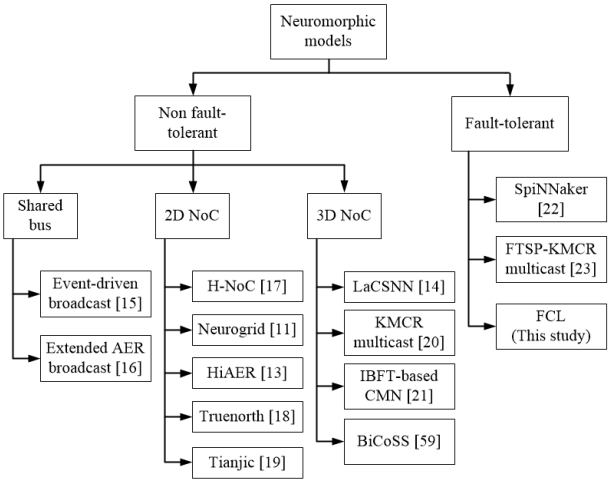


Fig. 1. Overview of the current neuromorphic models considering fault-tolerant properties.

As more neuromorphic designs are developed, fault-tolerance becomes essential and critical for reliable neuromorphic computing. Recently, a number of studies have focused on fault-tolerant neuromorphic designs. Notably, the SpiNNaker system presents a novel routing strategy to deal with the problems of congested or broken links in its digital neuromorphic architecture [22]. FTSP-KMCR proposes a multicast fault-tolerant architecture to implement a neural engineering framework [23]. However, there is a lack of fault-tolerant learning methodology for neuromorphic computing, especially based on brain-inspired learning mechanisms. To that end, this study proposes an FCL model and an AER multicast routing strategy on the IBFT architecture, which enables fault-tolerant context-dependent neuromorphic learning.

III. Network Model for Context-Dependent Learning

The network model for the implemented context-dependent learning mechanism is shown in Fig. 2(a). This network is composed of three layers: sensory, hippocampal, and

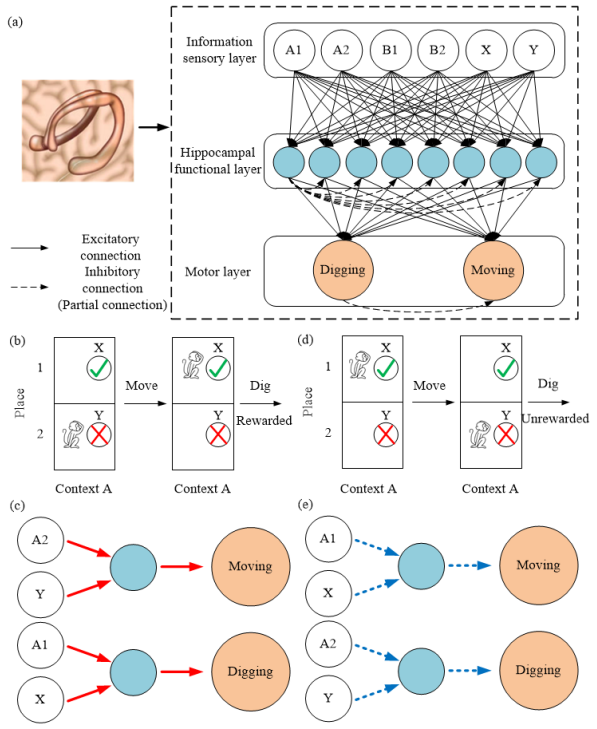


Fig. 2. Proposed learning scheme for a context-dependent task. (a) Schematic of the implemented SNN model for the context-dependent task. (b) Rewarded action sequence. (c) Rewarded action sequence leads to synaptic enhancement denoted by the red solid arrow. (d) Nonrewarded action sequence. (e) Nonrewarded action sequence leads to synaptic depression denoted by the blue dotted arrow.

motor layers. The neurons and learning synapses in these layers are explained in the following.

A. Neuron Model

The neuron model used in our implementation is based on the LIF model. In this model, the membrane potential V_i is governed by capacitance C and driven by the input current I_j, and the leaking current is affected by a leaky channel of conductance G_l. The membrane has the resting potential V_{rest} and is influenced by small fluctuations of a noise term η. The noise term here denotes a random variable $\eta \in N(\mu, \sigma)$ based on a Gaussian distribution with a mean value μ =0 and a standard deviation σ. The dynamic equation of the membrane voltage can be expressed as

$$C\frac{dV_i}{dt} = G_l(V_i - V_{\text{rest}}) + I_k + \eta \tag{1}$$

where $i = 1, \ldots, n_k$ and $k \in \{\text{sensory}, \text{hippo}, \text{motor}\}$ is one of the network layers. The input current for the sensory layer is $I_{\text{sensory}} = 1.00$ nA, while it is $I_{\text{hippo}} = 0.98$ nA for the hippocampal layer and $I_{\text{motor}} = 0.96$ nA for the motor layer. The model parameter values are listed in Table I, accordingly. The main purpose of our conducted experiments is to replicate, in brain-inspired hardware, the neuroscience context-dependent learning studies, which will give us confidence that we can use the hardware, in some cases, to simulate aspects of the brain. Therefore, our parameter values for the studied context-dependent learning task are adapted from the empirical behavioral data of a context-dependent learning task performed in [4].

TABLE I
PARAMETER VALUES OF THE NEURON MODEL

Parameter	Description	Value
C	Membrane capacitance	5.5×10^{-9} F
G_l	Leaky membrane conductance	10×10^{-9} S
V_{peak}	Peak membrane potential	0 mV
V_{th}	Threshold membrane potential	-50 mV
V_{reset}	Reset membrane potential	-70 mV
σ	Standard deviation of Gaussian noise	1 μV per step

TABLE II
PARAMETER VALUES OF STDP

Parameter	Description	Value
τ_+	Pre- before post-synaptic spike time constant	10 ms
τ_-	Pre- after post-synaptic spike time constant	10 ms
A_+	Pre- before post-synaptic spike amplitude	+1.2
A_-	Pre- after post-synaptic spike amplitude	-0.4
$w_{\min}$	Minimum activation for synaptic weight	0.0
$w_{\max}$	Maximum activation for synaptic weight	1.0
τ_w	Learning rate for weight adaptation	10 ms

B. Synaptic Connections and Learning Algorithm

In order to realize weight adaptation for learning, the STDP rule for synaptic modification is used [24]. Synaptic weights between the sensory layer, the hippocampal layer, and the motor layer are modified. The rule employs the time difference Δ between the presynaptic and postsynaptic spikes. If the presynaptic spike occurs before the postsynaptic spike, it induces a positive time difference $\Delta > 0$, resulting in a synaptic LTP. When the presynaptic spike happens after the postsynaptic spike, this results in a negative time difference $\Delta < 0$, inducing a synaptic LTD. This effect occurs within a small time window of ≈20 ms, with the weight dynamic range between $w_{\min} = 0$ and $w_{\max} = 1$. The STDP learning rule can be implemented as a differential equation as follows:

$$\tau_w \frac{dW_{ij}^{\text{exc}}}{dt} = \left(w_{\max} - W_{ij}^{\text{exc}}\right) \cdot A_+ \exp(-\Delta/\tau_+) - \left(w_{\min} - W_{ij}^{\text{exc}}\right) \cdot A_- \exp(+\Delta/\tau_-) \tag{2}$$

where $i = 1, \ldots, n_k$, $j = 1, \ldots, n_l$, $k \in \{\text{sensory, hippo, motor}\}$, $l \in \{\text{sensory, hippo, motor}\}$, and $k \neq l$. Indices "i" and "j" represent neurons from two connected layers, for instance, the hippocampal layer with the motor layer. The weight alterations, which may result in LTP and LTD, are controlled by the time constants τ_w, τ_+, and τ_-, respectively. The parameter values of the implemented STDP learning rule are listed in Table II.

Neural spikes are transmitted between layers via the excitatory weights W^{exc} and inhibitory weights W^{inh}, as shown in Fig. 2(a). The WTA rule is used at the receiving terminal to generate a current of $I_{\text{hippo}} = 0.98$ nA for neurons in the hippocampal layer or $I_{\text{motor}} = 0.96$ nA for neurons in the motor layer. The WTA rule is defined as follows:

$$I_{j*} = I_k \text{if} j* = \arg\max_j \left\{ \begin{array}{l} \sum_{i=1}^{n_k} (V_i - V_{\text{reset}}) W_{ij}^{\text{exc}} \\ - \sum_{\substack{i=1 \\ i \neq j}}^{n_k} (V_i - V_{\text{reset}}) W_{ij}^{\text{inh}} \end{array} \right\} \quad \text{and } j = 1, \ldots, n_l \tag{3}$$

TABLE III
PARAMETER VALUES OF THE SNN MODEL USED IN THE TARGETED CONTEXT-DEPENDENT LEARNING TASK

Parameter	Description	Value
T_{trial}	Maximum time interval for a trial	10 ms
T_{replay}	Maximum time interval for replay	400 ms
Δt	Time increment per simulation step	+1.2
$I_{sensory}$	Input current for sensory neuron	1.0 nA
I_{hippo}	Input current for hippocampus neuron	0.98 nA
I_{motor}	Input current for motor neuron	0.96 nA

where the input current $I_j = 0$. Similar to (2), here, $k \in$ {sensory, hippo, motor}, and $l \in$ {sensory, hippo, motor}.

C. Network Architecture

The presented SNN model is inspired by visual, odors, and tactile sensory inputs in the form of binary vectors [25]. The sensory signals are delivered through six input neurons, i.e., $n_{\text{sensory}} = 6$. Four of them provide context-place information, and the other two provide item information. As shown in Fig. 2(a), the first input neuron is activated with context place A1, and the second input neuron is activated with context place A2, while the third and fourth neurons are activated with context places B1 and B2, respectively. The fifth and sixth neurons are activated with item information *X* and *Y,* respectively. The input neurons are connected to hippocampal neurons using adaptive weights W^{exc} to represent excitatory connections in an all-to-all setting. The hippocampal neurons have inhibitory connections W^{inh} among them, but not inhibiting themselves. Fig. 2(a) only shows the inhibitory connections of the first hippocampal cell to the other nine cells, for simplicity. Similar inhibitory connections are included among all other hippocampal neurons. The hippocampal cells are connected to two motor output neurons with adaptive weights using all-to-all connectivity. All the weights are uniformly randomly initialized with values in the range of 0–1. The two cells in the motor layer represent the action "digging" and "moving," respectively.

Fig. 2(b) shows a cartoon of the context-dependent learning at hand. Here, a model monkey can only move between places 1 and 2 in either context A or B, without intermediary places. It can perform no action to change the context, but the context can change randomly between trials. Some trials can start the model in context A and others in context B. Table III lists the model parameters used in our experimentation.

D. Learning of the Context-Dependent Task

The sequence of monkey actions can be represented using Fig. 2(b) and (d). The two square areas in the figure represent place 1 and place 2 of context A. In each square area, there is a ball representing item, item X and item Y, respectively. If there is a green check in the ball, it means that the reward is hidden under this item. If there is a red cross in the ball, it means that there is no hidden reward. As shown in Fig. 2(b), a case of a rewarded action sequence is A2Y, move, A1X, dig, and receive a reward. In the examples shown in Fig. 2(b) and (c), the learning in the first stage is assumed to occur to establish the correct connection to activate the first hippocampal neuron, inducing the synaptic activation of the neurons encoding the action "move." This neuron spikes several times, inducing the action "move" executed by the monkey. After moving, the monkey can sense place A1 and item X, which activates a hippocampal neuron because of the established connection, resulting in the activation of the neuron coding the action "dig," and the monkey obtains the reward. Since, in the first stage, the monkey has dug a reward, this action can be enhanced in the next stage, which is represented by the red solid lines, as shown in Fig. 2(c). For the procedure without reward, the monkey is first located at A1X, as shown in Fig. 2(d), and then moved to the location of A2Y and dug. Since the reward is at A1X, the monkey cannot obtain the reward. Thus, this action will be depressed in the next stage, which is represented by the blue dashed lines in Fig. 2(e).

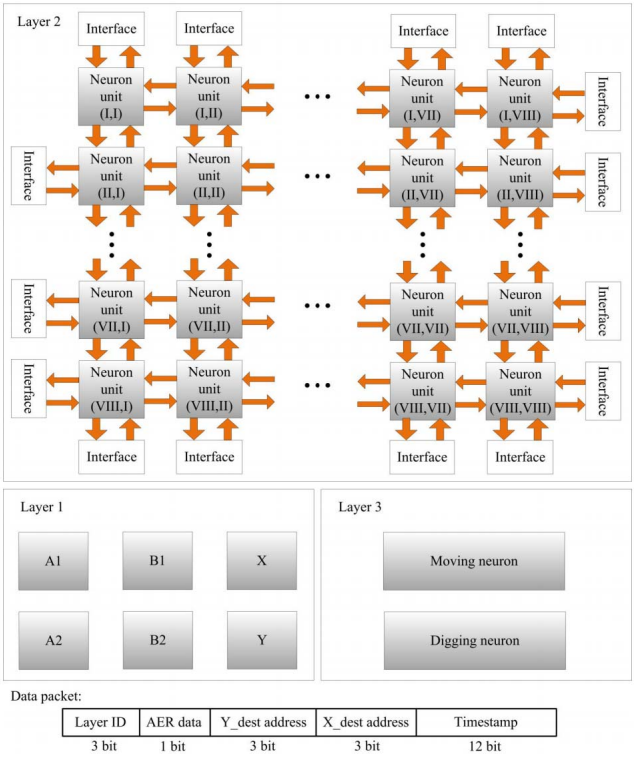


Fig. 3. 3-D NoC architecture of the proposed neuromorphic network.

IV. DIGITAL NEUROMORPHIC ARCHITECTURE

A. Network-on-Chip Architecture

The proposed FCL model is realized using FPGA and evaluated based on its average spike latency and throughput. Here, the FCL model for the targeted context-dependent learning task (shown in Fig. 2) is mapped onto the LaCSNN digital neuromorphic system in a layer-to-layer fashion, where our proposed routing algorithm is combined with the 3-D mesh NoC topology. The mapping strategy of SNNs onto NoC-based neuromorphic systems is critical for neuromorphic applications because it significantly influences both the overall performance and the power consumption. Fig. 3 shows the mapping of the context-dependent task to the 3-D NoC architecture, in which neurons in the same network layer are mapped onto the same architecture layer. In a previous study with fault-tolerant spike routing, neurons can only send spikes to the next layer [23]. In this study, this limitation is eliminated by using a novel mapping and NoC architecture.

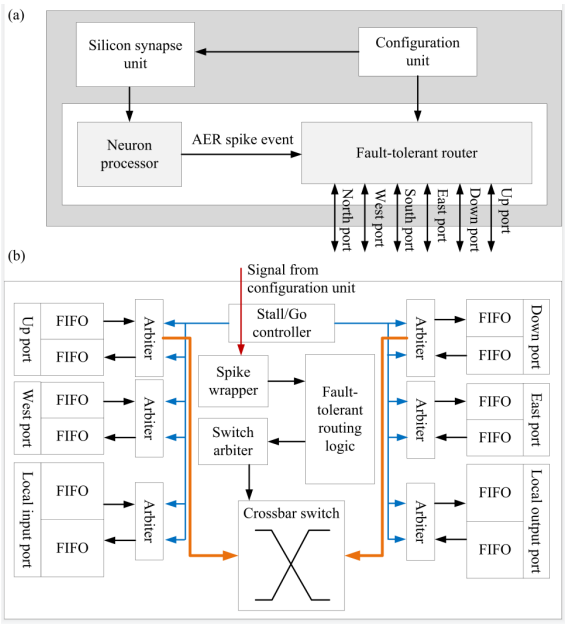


Fig. 4. Detailed digital neuromorphic architecture of the neuron unit and routers in the proposed decision-making spiking network. (a) Digital neuromorphic architecture of the neuron unit. (b) Digital neuromorphic architecture of the fault-tolerant router.

In the first layer, six neuron units representing the six neurons in the information sensory layer are distributed in parallel and are fully connected with the neuron units in the second layer. In the second layer, the neuron units are implemented on an 8 × 8 NoC architecture, which uses a mesh-based multicasting AER strategy. The moving and digging neurons are implemented digitally in the third layer. All three network layers are fully connected. The spike event packet transferred in this network contains 22 bits, which include 3-bit layer ID, 1-bit AER data, 3-bit Y_dest address, 3-bit X_dest address, and 12-bit Timestamp.

The detailed digital architecture of the neuron unit is shown in Fig. 4(a). It contains a neuron processor, a fault-tolerant router, a synapse unit, and a configuration unit. Each router has six ports, including up, down, north, west, east, and south to route the AER packets to another neuron unit. Compared to a previous study [26], the proposed architecture has three features: 1) computation using events with synaptic weighting; 2) implementation of physical synapses; and 3) fault-tolerant neuromorphic routing capability. Therefore, the proposed architecture is more suitable for the neuromorphic SNN computation aiming at complicated cognitive behaviors, such as context-dependent learning.

B. Fault-Tolerant Router Architecture of the Proposed Neuromorphic Network

The router is critical in the proposed neuromorphic architecture because it plays a vital role in achieving the fault-tolerance targeted. The fault-tolerant multicast 3-D router architecture implemented in the neuron units of our system is shown in Fig. 4(b). At the first stage, the spike events are received from the four neighboring nucleus processors, and their packets are stored in the input buffer before being processed. The spike wrapper unit is used to convert a single spike event into a valid AER spike packet using the information received from

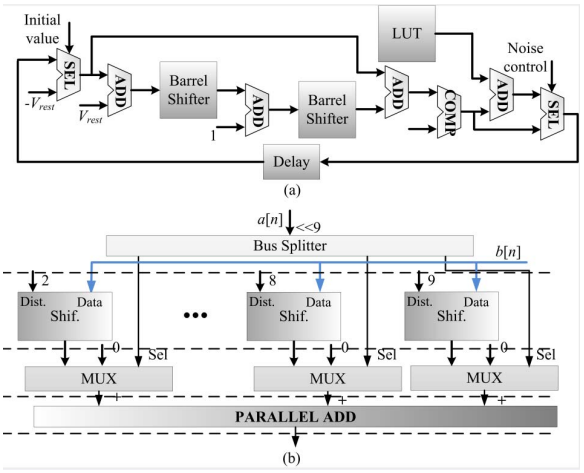


Fig. 5. Detailed digital architecture of the neuron processor and the SLM block. (a) Detailed digital implementation of the neuron unit. (b) Detailed digital implementation of the SLM block.

the configuration processor. This processor can be started at any time based on neuronal connectivity. The configuration processor contains four types of registers: the chip address register, the layer address register, the node address register, and the timestamp register. Incoming spike events and the corresponding deliver-at time are stored in the on-chip memory after the deliver-at time stamps are reached. Then, the source address of the packet is extracted and calculated to determine the output port. The fault-tolerant routing calculation, which will be introduced in Section V, is then used to route the packet. The switch arbiter is used using the least recently served priority to provide fast computation, inexpensive implementation, and strong fairness, as presented in previous studies [27]. Finally, the packet is sent to the desired output port through the crossbar. The crossbar switch is controlled by the switch arbiter and implemented by multiplexers.

C. Digital Architecture of the Conductance-Based LIF Neuron Model

In this study, the Euler method is used for the discretization of the neuron model since it can save hardware resources compared to the Runge–Kutta method. Based on the Euler method, the original equations can be transformed into the following equations:

$$\begin{cases} V(n+1) = \Delta t \cdot \left(\dfrac{1}{C}(G_l(V_i - V_{\text{rest}}) + I_k + \eta)\right) + V(n) \\ W_{ij}^{\text{exc}}(n+1) = \Delta t \cdot \left(\begin{array}{l} \left(w_{\max} - W_{ij}^{\text{exc}}\right) \cdot A_+ \exp(-\Delta/\tau_+) \\ -\left(w_{\min} - W_{ij}^{\text{exc}}\right) \cdot A_- \exp(+\Delta/\tau_-) \end{array}\right) \\ \qquad + W_{ij}^{\text{exc}}(n). \end{cases} \tag{4}$$

The digital architecture of the neuron model in the information sensory, hippocampal, and motor layers is shown in Fig. 5(a). Multipliers are extravagant hardware resources in digital design and are usually avoided as much as possible to gain energy and hardware cost benefits. Thus, in the proposed digital architecture, SLMs are used to replace multipliers to realize multiplication operations. Fig. 5(b) shows the detailed

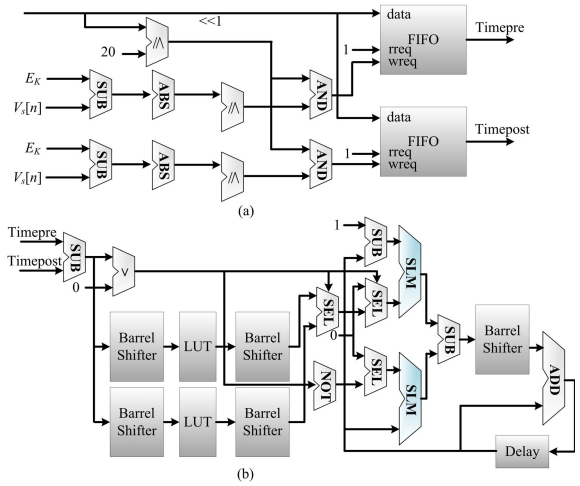


Fig. 6. Digital architecture of the synapse. (a) Detailed digital implementation for the computation of "Timepre" and "Timepost." (b) Digital implementation of the STDP weight updating module.

digital implementation of the SLM block, which is used to replace the multipliers in this study.

D. Digital Architecture of the Synapse Module Implementing STDP

The learning capability of the proposed FCL model is based on the STDP learning algorithm. The detailed digital implementation of this learning algorithm is shown in Fig. 6. As demonstrated in Fig. 6(a), the presynaptic and postsynaptic spike timings "Timepre" and "Timepost" are first calculated, which are then used in the digital implementation of the STDP learning rule shown in Fig. 6(b). Here, LUTs are used to calculate the exponential part in the STDP algorithm, and barrel shifters are used to replace the required multipliers. These components help to significantly cut down the hardware resource cost and power consumption.

V. Fault-Tolerant Spike Routing

Several previous studies have investigated fault-tolerant routing schemes for NoC topologies [22], [23]. These schemes are based on different approaches, including virtual channels, path-finding, and bypass methods to perform efficient routing in the presence of faulty nodes. The virtual channel-, path-finding-, and bypass-based routing schemes are different in terms of the hardware resource cost and circuit complexity. The routing schemes based on virtual channels divide a single physical link into several virtual channels, but they require complex control circuitry and cost large hardware resources and power consumption. The routing schemes based on path-finding methods need a large number of routing tables, which induces a large hardware resource cost. Contrarily, bypass methods divide the fault nodes into nonoverlapping fault areas and make use of the normal nodes and links around the fault areas to form a new routing path. When the data packet reaches a fault area, it will be bypassed along a new routing path based on certain rules, thus avoiding the fault without deadlock. Here, we utilize the bypass method without virtual channels to perform fault-tolerant routing in the proposed neuromorphic architecture for the targeted context-dependent learning task. BIST technique is used to get the location information of the fault node. Furthermore, the realization of the load balancing in the bypass loop and the reduction of the communication latency can be performed by optimizing the routing distance of the spike events.

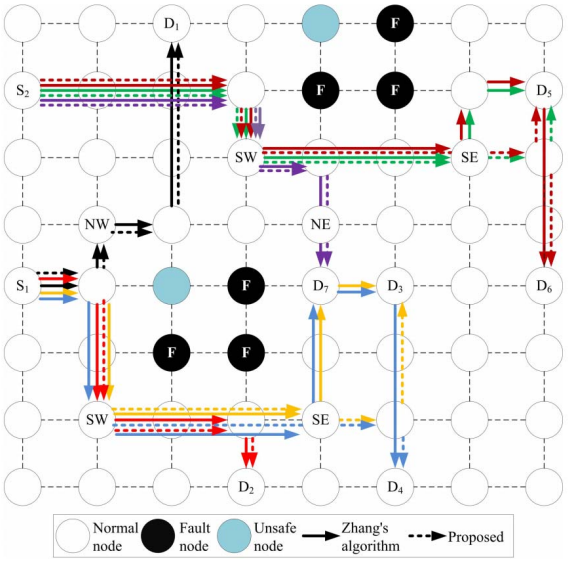


Fig. 7. Bypass route of the spike event along the X-direction.

A. Multiple-Fault-Tolerant Neuromorphic Algorithm

In the proposed algorithm, the XY routing strategy is used, which means that the neural information is routed first along the X-direction, then along the Y-direction. Thus, the proposed algorithm is divided into two parts, which are along the X- and Y-directions, respectively. In a previous study, Chen and Chiu [48] have presented an essential fault-tolerant solution for NoC design. However, Chen's algorithm requires significant computations and a large area, with a low node utilization. In addition, its communication load is heavy, and its latency is high. On the contrary, our presented MFTN algorithm modifies and expands the single-fault bypass method in Zhang's algorithm [46] to improve the multiple-fault situation. The MFTN algorithm along both X- and Y-directions will be introduced and compared to Zhang's algorithm to further illustrate its improvement based on Zhang's algorithm in this section. The experimental results will be shown in Section VI and compared to Chen's algorithm to demonstrate its performance improvement.

In the case of multiple-fault-tolerant schemes, there are two scenarios. In the first scenario, the source and destination nodes are located on different sides of the fault region, and the source node is located in one of the rows of the fault region. For example, see source 1 (S1) and destination 3 (D3), or S2 and D5 in Fig. 7. In the second case, the source node is located within the rows of the fault region, and the destination node is located in the columns of the fault region. For an example of this case, see the positions of S1 and D2 in Fig. 7. As shown in this figure, we first delineate the fault regions that need to be bypassed, which includes the fault nodes and the unsafe nodes surrounded by the fault nodes. The coordinate information of the four SW, NW, SE, and NE nodes is transmitted to all the normal nodes in the corresponding column by all the nodes in the bypass loop and

Pseudo code for single-fault neuromorphic tolerant algorithm along X direction

```
if (C is destination D)
      {Processing the spike event ; }// Reaching the destination node
else if (fault information memory of C is empty) {
            Continue route according to XY algorithm;
            }// If the fault information is NULL, routing based on XY algorithm
else if (fault information memory of C will not affect routing from S to D ) {
            Continue route according to XY algorithm;
            }
          // If the fault region is not located on the pathway of XY routing,
continue routing based on XY algorithm
else if (X_SW<=X_D<=X_SE && Y_SW<=Y_S<=Y_NW){
            Route according to Zhang algorithm;
            }
else if (Y_SW<=Y_S<=Y_NW && (X_D<X_SW<X_SE<=X_S ||
X_S<=X_SW<X_SE<X_D) ){
                  // When satisfying the first case, begin optimization mode
            if (!(X_D<X_SW<X_SE<=X_C || X_C<=X_SW<X_SE<X_D)){
                  Route to X_D according to Zhang algorithm;
                        // Routing to the column of D based on Zhang's
algorithm
                  Continue route according to XY algorithm;// Routing spike
event based on XY algorithm
                  }// C and D do not locate on the both sides of the fault region
            else {
                  do {
                        Route to next node according to Zhang algorithm;
                              // Routing to the next node based on Zhang's
algorithm
                  }while (!(X_C==X_SW<X_S || X_C==X_SE>X_S));
                  Continue route according to XY algorithm;// Continue routing
based on XY algorithm
                  }// C and D locate on the both sides of the fault region
            }
```

Fig. 8. Pseudocode for MFNT algorithm along the X-direction. C and D represent the current and destination nodes, respectively. In the XY algorithm, the neural information will be routed along the X-direction at first and then routed along the Y-direction.

stored in the on-chip memory to determine whether the routing process passes through the fault region. In Fig. 7, the solid arrow is based on Zhang's algorithm [46], and the dotted line arrow is based on the proposed MFTN routing algorithm. In comparison with Zhang's algorithm, the proposed algorithm optimizes the bypass strategy, which decreases the routing distance accordingly. For example, in the case from S1 to D4, Zhang's algorithm will turn up at SE, which induces a longer distance. In contrast, the proposed MFTN algorithm will provide a direct route to the D4 node. The pseudocode of the proposed MFTN algorithm is shown in Fig. 8. In the proposed pseudocode, C and D represent the current and destination nodes, respectively. In the XY algorithm, the neural information will be routed along the X-direction at first and then routed along the Y-direction. It is majorly for the case where the source and destination nodes are located on the east and west sides of the fault region, separately. The bypass loop is separated from the original pathway when certain conditions are satisfied; therefore, the load of the bypass loop and the routing distance are decreased.

The situation for the bypass along the Y-direction in the case of multiple fault nodes is that the destination and source nodes are located on the south and north sides of the fault region, respectively, and the destination node is located within the columns of the fault region. According to Zhang's algorithm, more change of the routing direction is required along the X-direction, which induces the increment of the bypass distance, and the load is majorly on the left-hand side of the bypass loop. When the current node (C) is on the north side of the bypass, the spike event is first routed to the nearest

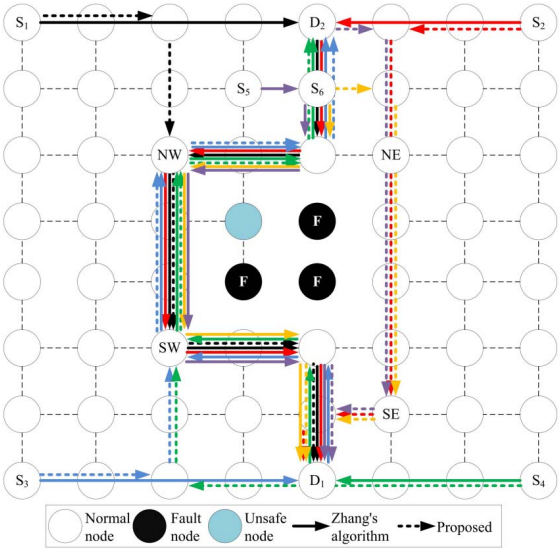


Fig. 9. Bypass route of the spike event along the Y-direction.

corner based on the XY algorithm and then bypassed based on Zhang's algorithm. As shown in Fig. 9, if the current node is on the south side of the bypass loop, the spike event is first routed to the SW node due to the prohibition of turning on the NE corner and then bypassed based on Zhang's algorithm. The pseudocode for the proposed MFNT algorithm along the Y-direction is shown in Fig. 10.

B. Deadlock-Free Fault-Tolerant Routing

According to a previous study [44], the necessary and sufficient condition for any routing algorithms to be deadlock-free is that there is no loop in its corresponding CDG.

In the case of networks without any fault nodes, since the turn from the Y- to X-directions is prohibited, there is no cycle in the CDG of the mesh-based network. When the fault region is located inside the network, the proposed MFTN algorithm adds turns from the Y- to X-directions on the northwest, southwest, and southeast corners of the bypass loop and removes the turn from the X- to Y-directions on the northeast corner. Since this turn is removed, no cycle will exist, and the proposed algorithm is deadlock-free.

When the fault region is located on the edge of the mesh network, the proposed algorithm adds the turns from the Y- to X-directions on the vertex of the bypass loop in order to make spike events avoid the fault region. Since the bypass loop is not a cyclic link, it will not result in a deadlock. Therefore, the proposed MFTN algorithm is completely deadlock-free. It is also independent of the area and location of the fault region and the NoC scale.

VI. Experimental Results

In this section, we present experimental results implemented on the digital neuromorphic system LaCSNN [14], which is modified to include the proposed MFTN algorithm and implement the required context-dependent learning task. First, the experimental oscilloscope outputs are displayed in Fig. 11. This figure demonstrates the real-time operation of our hardware on an oscilloscope. It shows that the silicon neurons in our digital architecture are spiking in a similar way that biological neurons fire in a context-dependent learning setting.

Pseudo code for multiple-fault neuromorphic tolerant algorithm along Y direction

```
if (C is destination D)
        {Processing the spike event ; }// Reaching the destination node
else if (fault information memory of C is empty) {
                Continue route according to XY algorithm;
            }// If the fault information is NULL, routing based on XY
algorithm
else if (fault information memory of C will not affect routing from S to D ) {
                Continue route according to XY algorithm;
            }
          // If the fault region is not located on the pathway of XY routing,
routing spike event based on XY algorithm
else if (X_SW<=X_D<=X_SE && ((Y_S>Y_NW &&
Y_D<=Y_SW)||(Y_S<Y_SW && Y_D>=Y_NW))){              if
(Y_C>Y_D){
                if (abs(X_C-X_NW)>abs(X_C-X_NE)){
                    Route to NE according to XY algorithm;              }
                else {
                    Route to NW according to XY algorithm;
        }
            }
            else {
                Route to SW according to XY algorithm;        }
            Route to X_D according to Zhang algorithm;
            Continue route according to XY algorithm;
            }
```

Fig. 10. Pseudocode for the MFNT algorithm along the Y-direction.

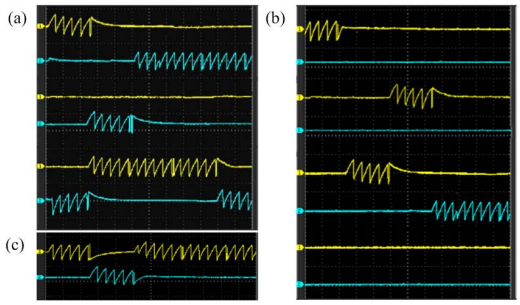


Fig. 11. Experimental oscilloscope output results. The time division is set to ms, while the amplitude division is mV. (a) Firing activities in the information sensory layer. (b) Firing activities of eight randomly chosen neurons in the hippocampal functional layer. (c) Firing patterns of the two neurons in the motor layer.

It shows the firing activities of the neurons in the three different layers of the implemented neuromorphic network realizing the targeted context-dependent learning task. Fig. 11(a) shows the spiking activities of all the six neurons in the information sensory layer. The first four neurons spike alternatively due to the specific input combination, and the last two neurons fire alternatively due to the WTA mechanism. Fig. 11(b) shows the spiking activities of eight neurons randomly chosen from the hippocampal functional layer. This figure shows that only one neuron spikes at any given time due to the WTA mechanism. Fig. 11(c) shows the firing activities of each neuron in the motor layer, in which the two neurons spike alternatively. These real-time millisecond-scale spiking activities reveal that biological behaviors can be reproduced accurately in real time.

In addition to the behavioral analysis of the neurons firing patterns, their selectivity is also evaluated rigorously in 200 runs, each with different weight initialization and random noise. In order to characterize the selectivity of the neurons in the proposed neuromorphic network, a criterion of selectivity index (SI) is defined as follows:

$$\mathrm{SI} = \left(n - \sum_{i=1}^{n} \lambda_i / \lambda_{\mathrm{pref}}\right) \Big/ (n-1) \tag{5}$$

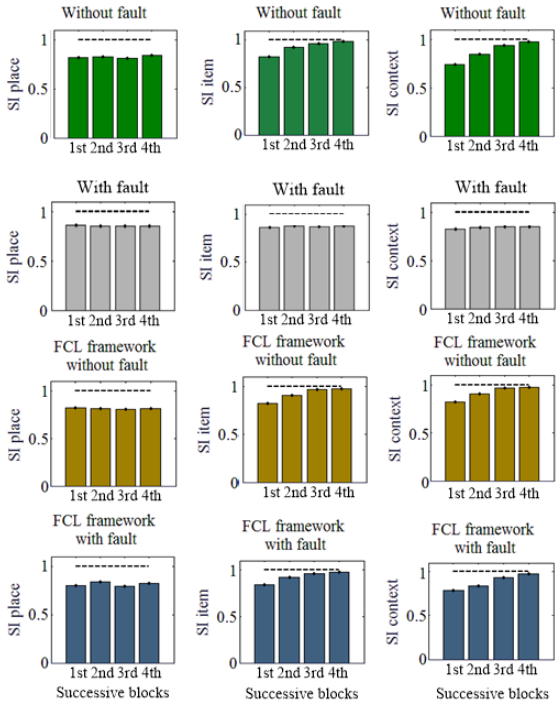


Fig. 12. Selectivity of the neuromorphic network shown in Fig. 2(a) without the proposed MFTN algorithm in two cases of without and with faults (first two rows) and the proposed FCL framework with and without faults for different parts of the context-dependent learning task, i.e., place, item, and context selections. Here, the x-axis shows four successive blocks each with 50 trials.

where n represents the stimulus events, λ_i represents the firing rate in response to the ith stimulus event for a single neuron, and λ_{pref} represents the preferred stimulus event for the same neuron. The variable λ_{pref} is calculated based on the maximum firing rates of all the experienced stimulus events for each neuron. For the case of place selectivity, $n = 4$ because the context-dependent task contains four physically different places, including A1, A2, B1, and B2. In order to implement the place selectivity, the mean firing rate for when the toy monkey encounters items X and Y is combined when the toy monkey is in each of these four places. In the case of item selectivity, $n = 2$ because two different items exist in the proposed context-dependent learning task.

For the SI calculations, only neurons in the hippocampal functional layer are explored because this layer determines the selectivity capability. For all values of SI, the mean and standard deviation are calculated over four 50 trials, as shown in Fig. 12. In this figure, four cases are investigated, including: 1) a neuromorphic network implementing the network shown in Fig. 2(a) but without the proposed fault-tolerant routing mechanisms and without faulty nodes (the first row of Fig. 12); 2) the neuromorphic network without the proposed fault-tolerant routing mechanisms but with faulty nodes (the second row); 3) the proposed FCL framework without faulty nodes (the third row); and 4) the FCL framework with faulty nodes (the fourth row). As shown in Fig. 12, the mean value of SI for place selectivity is around 0.8 and remains constant in the four different cases. The mean value of SI for item selectivity begins from around 0.8 and increases to around 1.0, while the mean value of context selectivity begins around 0.7 increasing to around 1.0. These values suggest that the

place selectivity is at a continuous rate, but item and context selectivities improve during the context-dependent learning task. Overall, Fig. 12 shows that, compared to a network without fault-tolerant routing, the proposed FCL network can solve the fault problems, successfully. In the following, we present more experimental results, in all of which, the FCL framework with faulty nodes is used.

A. Neuromorphic Context-Dependent Learning Capability

In order to evaluate the learning performance of the proposed network, a criterion representing the weight change significance is defined as follows:

$$B_{ij} = 4\left(W_{ij} - 0.5\right)^2 \tag{6}$$

where $i \in 1, 2, \ldots, n_{\text{sensory}}$ and $j \in 1, 2, \ldots, n_{\text{hippo}}$. This criterion can show the binariness of the synaptic weights between the sensory layer and the hippocampal layer by quantifying the functional connectivity. It can map the extreme points 0 and 1 to 1 and the midpoint 0.5–0. That is to say, the binariness will be large if weights are either close to 0 or 1, while it will be small if the weights have values close to 0.5. This criterion will be used in Fig. 13(b) to show the weight change significance. If more weights are around 1, it means a higher change in them, and if more are around 0, it means lower changes. Note that the weight variation is larger when higher spiking activities occur within the network.

In the performed experiments, the proposed neuromorphic network is trained in 100 epochs, each containing 130 trials. There are two phases in a trial. During the first phase, the model monkey explores the environment. After the first phase of a trial, the second phase replays the action sequence that is generated in the first phase. As shown in Fig. 13(a), the accuracy of the network is improved with an increase in training epochs. In order to further enhance the learning capability, three improvement schemes can be investigated in the base network. The first scheme is to increase the number of layers in the proposed network. Full connections with excitatory synapses are then used between these layers, and lateral connections with inhibitory synapses are used on each layer. By doing this, the deep learning capability of the proposed network can be investigated. The results of this change shown in Fig. 13(b) demonstrate that the learning accuracy drops and stays around 60%. This means that the learning capability of the proposed neuromorphic network cannot be enhanced by increasing the number of layers of the hippocampal functional network. The second scheme is to increase the number of neurons within the hippocampal functional layer to explore whether the increasing neuron number will improve the learning capability.

Fig. 13(c) and (d) shows the learning accuracy of the networks whose hippocampal functional layers contain 16 and 64 neurons, respectively. As shown in Fig. 13(d), the network with the hippocampal functional layer containing 64 neurons has the highest learning capability, which induces higher learning accuracy and higher learning speed compared to the other three schemes. As shown in Fig. 13(d) and (h), when the learning capability of the network is higher, the value of B_{ij} is larger. In these figures, the first six neurons are in the information sensory layer, and the last two are in the motor layer. We randomly select eight synapses connected to

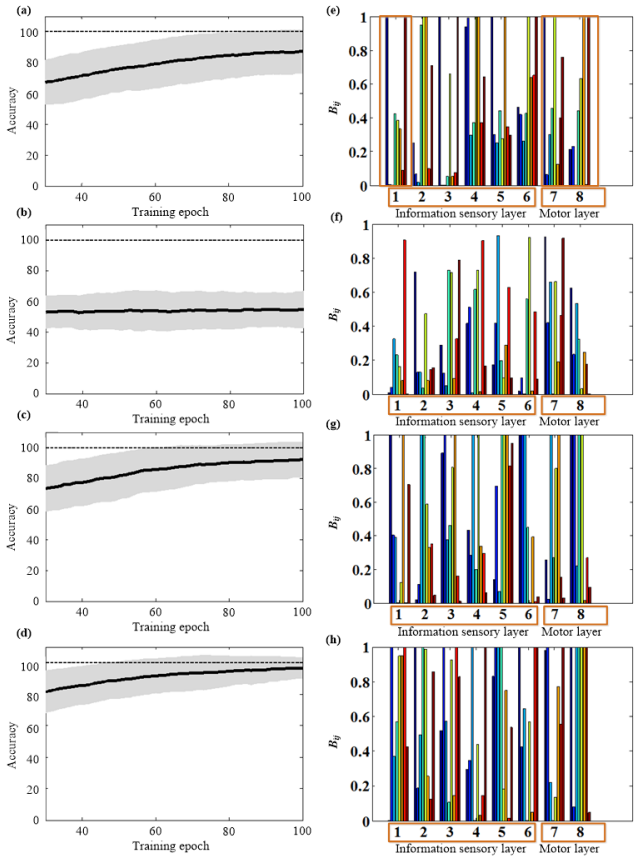


Fig. 13. Learning performance of the FCL framework with different network structures. (a) Learning accuracy of the original FCL framework trained in 100 epochs. (b) Learning accuracy by adding another layer of spiking neurons. (c) Learning accuracy with 16 neurons in the hippocampal functional layer. (d) Learning accuracy with 64 neurons in the hippocampal functional layer. (e) Corresponding weight changes with the original FCL framework. (f) Corresponding weight changes by adding another layer of spiking neurons. (g) Corresponding weight changes with 16 neurons in the hippocampal functional layer. (h) Corresponding weight changes with 64 neurons in the hippocampal functional layer.

the chosen neurons and evaluate the value of B_{ij} on those synapses. As shown in Fig. 13(h), when training does not actively happen, the proposed network does not experience significant weight changes. Thus, the learning capability is not available in this network architecture.

B. Context-Dependent Learning Analysis

In the implemented context-dependent learning task, there are eight input combinations according to the place, item, and context, which requires eight hippocampal neurons to process the information. These eight combinations are shown on the x-axis of Fig. 14, which is used to show the spiking situations with different inputs in the context-dependent learning task. In this figure, each of the eight neurons (six in the input sensory layer and two in the output motor layer) shown on the y-axis, can only account for one input combination. The problem with this naive connectivity setting is that two or more neurons may learn the same combination, while others do not learn any combination. This results in the loss of useful information processed by the proposed network. In addition, with the combination number increasing, the network will be enlarged inducing larger hardware resource costs. Therefore,

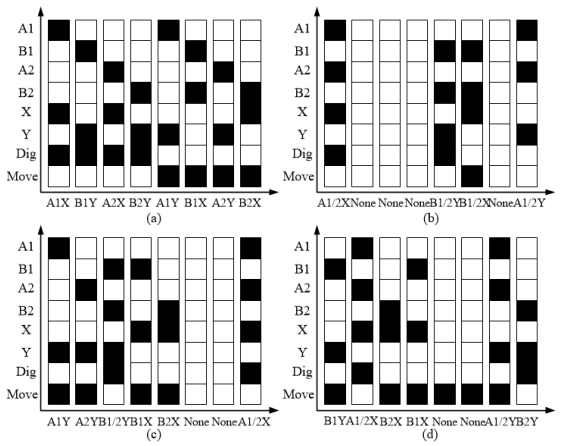


Fig. 14. Spiking situations with different inputs in the context-dependent learning task. (a)–(d) Four kinds of inputs for the FCL framework and its corresponding outputs.

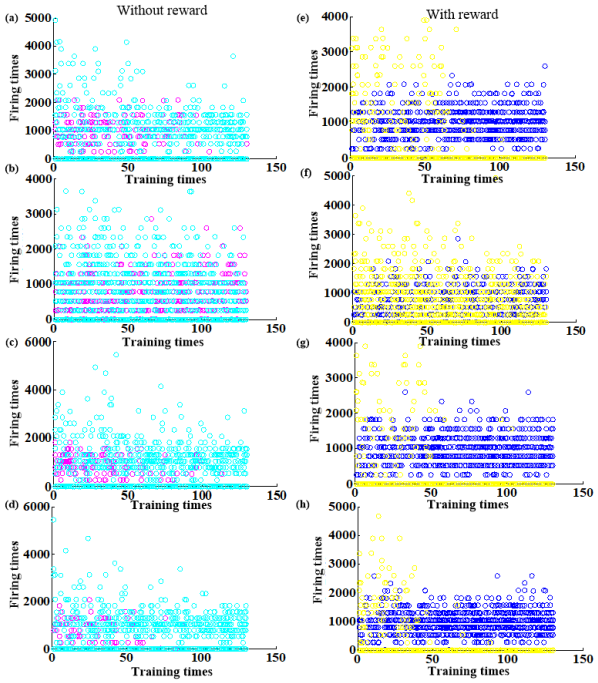


Fig. 15. Spiking activities of the proposed neuromorphic network with different layer numbers and neuron numbers in the hippocampal functional layer. (a) Raster plot with the original FCL framework without reward. (b) Raster plot with two layers in the hippocampal functional layer without reward. (c) Raster plot with 16 neurons in the hippocampal functional layer without reward. (d) Raster plot with 64 neurons in the hippocampal functional layer without reward. (e) Raster plot with the original FCL framework with reward. (f) Raster plot with two layers in the hippocampal functional layer with reward. (g) Raster plot with 16 neurons in the hippocampal functional layer with reward. (h) Raster plot with 64 neurons in the hippocampal functional layer with reward.

a different scheme with the STDP learning rule is used. As shown in Fig. 14(b), each neuron can recognize two situations, and all the information can be learned. The neuron number of the hippocampal functional layer is increased to 16 and 40, respectively, as shown in Fig. 14(c) and (d). The network can recognize more combinations of situations; thus, more information combinations can be learned. In addition, as shown in Fig. 14(b)–(d), the activated neuron numbers are increased along with the increasing neuron number in the hippocampal layer, which results in the increasing learning accuracy, as shown in Fig. 13(c) and (d).

In order to analyze the difference in the learning accuracy of the various networks shown in Fig. 14, the firing activities of these networks are assessed. In order to analyze the difference in the learning accuracy of the various networks shown in Fig. 14, the firing activities of these networks are assessed. Fig. 15 demonstrates these activities in different network structures and settings by using raster plots. In Fig. 15(a)–(d), the purple circles represent the number of spikes the digging neuron fires during training time without reward, and the light blue circles indicate the number of spiking activities with "moving" action without reward. The spike numbers of the "moving" and "digging" actions with reward are represented by yellow and dark light circles, respectively, in Fig. 15(e)–(h). As shown in Fig. 15(a), the number of false (unrewarded) "digging" actions is considerably large at the beginning of the learning (more purple circles). As the training continues, the unrewarded (false) actions begin to decrease. The same tendency occurs in Fig. 15(e), in which the number of the "moving" actions (yellow circles) decreases with the training time increasing. Fig. 15(b) and (f) shows the same results but for a larger network with two layers in the hippocampal functional layer, respectively. These figures show that a larger network cannot improve the learning capability. The right and wrong actions occur alternatively without regularity in this case. Fig. 15(c) and (d) and (g) and (h) shows the network with X neurons in its hippocampal functional layer, where $X = 16$ for Fig. 15(c) and (g) and $X = 64$ for Fig. 15(d) and (h). These figures suggest that the false actions can be reduced with the neuron number increasing. It enhances the context-dependent learning capability, which is consistent with the experimental results shown in Fig. 13. In addition, by comparing Fig. 15(a) and (b), it shows that the tendency toward a firing of certain kinds of neurons in the network with reward will be stronger compared to the network without reward.

In order to further explore the context-dependent learning capability of the proposed FCL framework, another experiment is conducted. Our experiment simulates macaque monkeys navigating a virtual maze [see Fig. 16(a)] during a context-dependent learning task, similar to the one performed in [52]. At the end of the roads in the three-way maze considered in our experiment, a reward is buried in soil [see Fig. 16(b)]. Different soils have different touch sensations, which provide tactile information for the macaque monkey. A reward has a special smell, which can attract monkeys to the reward site and provide olfactory information. The rewards buried at the end of the maze roads correspond to different context-object situations. The monkey needs to identify the reward place based on the context and dig out the reward. For further information about this experiment, please refer to [52].

To replicate the context-object associative memory task on our FCL hardware, we developed a model, whose inputs contain information of context, place, and reward. In our experiments, visual, tactile, and olfactory information is encoded and integrated into binary inputs, which are included in the context information. Three kinds of context information include A, B, and C. The rewards are X, Y, and Z. The locations of rewards are represented by 1, 2, and 3. The three contexts are combined with three locations, as the model inputs A1, B1, C1, A2, B2, C2, A3, B3, and C3. The expectations

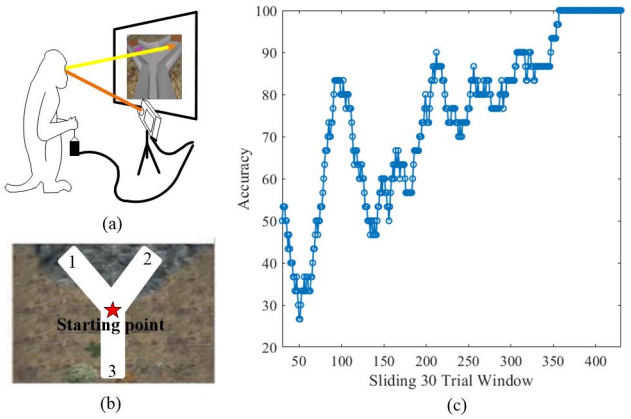


Fig. 16. Behavioral task for context-dependent learning in maze navigation and the learning accuracy. (a) Experimental setup of the task with the monkey navigating the maze via a joystick to reach the reward place taken from [52]. (b) Overhead view of the maze in the case of our developed experiment where the reward can be available. (c) Our proposed FCL system learning accuracy for 430 trials with a sliding 30-trial window.

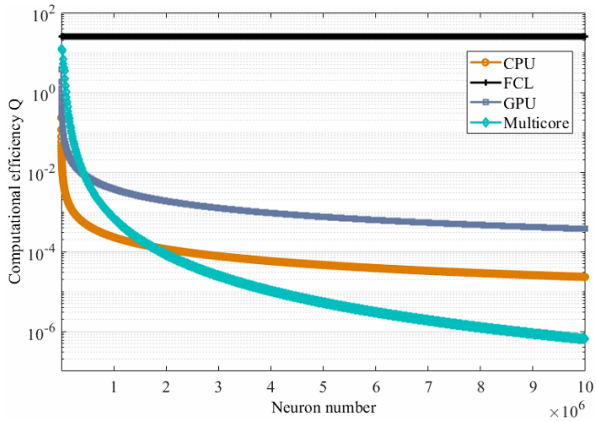


Fig. 17. Computational efficiency and scalability in comparison with other approaches, including CPU, GPU, and multicore systems.

of three kinds of rewards, X, Y, and Z, are used as the other three inputs. There are three output neurons in our FCL model for the three places. We simulated the proposed FCL network with randomly initialized weights. After several trials, the monkey model learned the task of digging in a context-dependent environment. As shown in Fig. 16(c), the accuracy of our monkey model in decision-making reaches 100% after 350 trials. The accuracy calculation method is given as follows. When the reward value that the monkey received is 1, it means that the decision-making is correct. Situations A, B, and C mean that the real reward values are in positions 1, 2, and 3, respectively. When the expected reward value is in the same position as the real reward and the digging action is realized, the monkey will receive the reward value 1. The accuracy in Fig. 16(c) is obtained in 30 consecutive trials, and the average accuracy value during 30 trials is used as the calculated accuracy.

C. Performance Analysis of the Proposed FCL Framework and Fault-Tolerant Algorithm

In order to demonstrate the computational efficiency and scalability of the proposed FCL framework on the LaCSNN system, we have performed experiments comparing our system with three different hardware platforms, including CPU-, GPU-, and multicore-based implementations. Similar to a previous study [61], a function for representing the computational efficiency is defined as

$$Q = t_{\text{exp}}/t_{\text{bio}} \tag{7}$$

where t_{exp} and t_{bio} are the experimental computational time on the simulation system and the biological activity time, respectively. The CPU-based simulation system uses the quad-core Intel Xeon CPU. The multicore bus implementation uses the Altera EP3E340 FPGA chip, and the GPU-based implementation uses an NVIDIA GTX 280 GPU. As shown in Fig. 17, the computational efficiency of our FCL platform is consistently the highest with the network size increasing, while the GPU, CPU, and multicore systems' efficiencies drop with the increase in the network size.

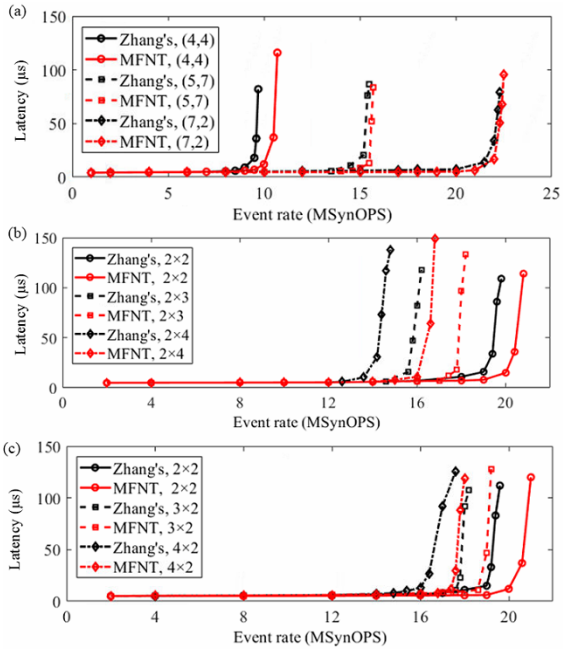


Fig. 18. Performance analysis of the fault-tolerant algorithms for multiple fault nodes. (a) Communication latency with different locations of faulty regions in the (8, 8) mesh network of neurons. (b) Communication latency with different areas of fault regions changing along the horizontal direction. (c) Communication latency with different areas of fault regions changing along the vertical direction.

The performance analysis of our proposed fault-tolerant algorithm for multiple fault nodes is presented in Fig. 18. Here, the northeast node on the bypass loop is defined as the reference node. As shown in Fig. 18(a), the performance of the proposed MFTN algorithm is better than Chen's algorithm [48] with different locations of faulty regions in the (8, 8) mesh network of neurons. As shown in Fig. 18, these algorithms follow a similar trend. The latency is the highest when the faulty region is in the network center, but it is the lowest when the fault region is located in the network vertex. This is because more nodes are affected when the fault region is in the network center, inducing the most influence on the data transmission latency. The figure shows that, no matter where the fault region is located, the latency performance with the proposed MFTN algorithm is better than Chen's. When the communication latency is 80 μs, the improvements of the

maximum event rate are 8.3%, 1.9%, and 0.9%, respectively, when the reference node is located in the network center (4, 4), the network edge (5, 7), and the network vertex (7, 2), respectively. Therefore, the best performance achieved using the proposed MFTN algorithm is when the fault region is located in the network center. This is because when the fault area is located in the network center, the number of source and destination nodes satisfying the optimization condition is the largest; therefore, the improved performance is the most obvious. When the fault nodes are located at the network vertex, the number of source and destination nodes satisfying the optimization condition is the lowest.

With the enlargement of fault area, compared with Chen's algorithm, the proposed algorithm provides a more significant latency advantage. In Fig. 18(b), the latency of the proposed algorithm is compared with Chen's algorithm when the fault area is 2×2 [the location of the reference node is (4,4)], 2×3 [the location of the reference node is (4,4)], and 2×4 [the location of the reference node is (6,5)]. Fig. 18(c) shows the same fault area size and reference node location as in part (b) but with fault regions changing along the vertical direction. The results show that, no matter the fault area enlarges along with the horizontal or vertical directions, the network latency of the proposed algorithm is better than Chen's algorithm.

When the network delay is 100 μs, compared with Chen's algorithm, the saturation injection rate increases by 5.2%, 11.7%, and 16.1%, respectively, when the fault area enlarges horizontally [see Fig. 18(b)]; however, when the fault area enlarges vertically, the saturation injection rate increases by 5.2%, 5.5%, and 6.1%, respectively [see Fig. 18(c)]. This is because when the data encounter the fault region along the Y-direction, the proposed algorithm cannot reduce the distance from the source node to the destination node and the transmission length on the bypass. It also allocates part of the original data that need to bypass along the left half of the fault region to the right half to improve the load balance on the bypass loop. Furthermore, when the fault area enlarges along the horizontal direction, the number of nodes needed to bypass the fault area along with the X-direction decreases, and the number of nodes needed to bypass the fault area along with the Y-direction increases. Therefore, the network performance of the proposed algorithm is more prominent when the fault area increases along the horizontal direction.

VII. Discussion

In this study, a neuromorphic context-dependent learning framework is proposed with a novel multiple-fault-tolerant spike routing scheme. The main contribution of our paper is implementing a fault-tolerant on-chip learning platform, which is applied to context-dependent learning. This has not been addressed before and is completely new. The novelty of our work is mainly around the introduction of a fault-tolerant learning architecture, where we have demonstrated its use using the simple task of context-dependent learning from neuroscience. We have shown that our network achieves good context-dependent learning performance despite many possible faults. In addition, to further verify the functionality and capabilities of our fault-tolerant on-chip learning platform, we have performed a more complex context-dependent learning of object and space in a virtual maze navigation task [52]. In order to implement the targeted learning framework in hardware, the digital neuromorphic system LaCSNN is used, which uses FPGAs to implement neuromorphic models. With the advantages of low energy consumption, high reconfigurability, parallel processing capability, and fast time to market [28]–[35], FPGA implementations show promising potential for high-performance neuromorphic systems [49], [50]. In the proposed FCL framework, the mean firing rates and rate-based coding are used to calculate the SI, which is used to characterize the selectivity of the neurons and for the implementation of the place selectivity. The firing rate is used to describe the criteria of SI. As shown in (2), the spike timing is employed to determine the time difference between the presynaptic and postsynaptic spikes. A time window of 20 ms is used for the calculation of the time difference. The positive and negative time differences induce LTP and LTD of the STDP learning rule. In its current form, our FCL framework can be used for time-based STDP learning only. In summary, there are three critical points to be discussed to highlight the contributions of the proposed framework and present potential directions for future studies.

First, fault-tolerant neuromorphic architecture with a novel multicast routing scheme is presented, which is scalable and applicable to a variety of neuromorphic applications. The bypass method without virtual channels is used in the proposed fault-tolerant routing scheme implemented in the neuromorphic network in this study. In order to comprehensively compare our work with state of the art, a multifault routing scenario is considered. As shown in Fig. 16, in comparison with a previous fault-tolerant routing scheme using the bypass method [48], the improvements achieved in the event rate, when the communication latency is 80 μs, are 5.3%, 1.9%, and 0.9% when the fault region is located in the network center, network edge, and network vertex, respectively. Furthermore, the event rate of the proposed fault-tolerant algorithm can be improved by 5.2%, 11.7%, and 16.1% when the fault area is enlarged along the horizontal direction and improved by 5.2%, 5.5%, and 6.1% with the fault area increasing in size along the vertical direction.

Second, a digital neuromorphic context-dependent learning model inspired by the hippocampus-mPFC pathway is proposed and implemented in this study. The neural mechanisms underlying the spiking activities of the hippocampus-mPFC network and context-dependent learning are fully investigated using the proposed neuromorphic FPGA framework. Fig. 12 shows the neuronal selectivity of the network, which reveals that place selectivity rate remains constant during context-dependent learning tasks, while item and context selectivities improve with learning. Fig. 13 shows that the learning performance can be enhanced by increasing the number of neurons in the hippocampal functional layer, while Fig. 15 displays the spiking activities of the neurons underlying the context-dependent learning and confirms that 64 neurons in the hippocampal layer can lead to better context-dependent learning. However, the learning accuracy decreases when the neuron number in the hippocampal layer exceeds 128, and the learning accuracy with 256 neurons will be reduced to 93.33%.

Third, our designed neuromorphic system integrates the proposed fault-tolerant routing capability with the hippocampus-mPFC pathway-inspired context-dependent learning, in a unified framework. In comparison with the study by Liu *et al.* [62], the proposed FCL framework has two advantages. First, the multiplication operation is replaced by an

TABLE IV
HARDWARE RESOURCE COST OF THE PROPOSED FCL FRAMEWORK

Resource	Available	Used
Computational ALUTs	270400	62%
Memory ALUTs	135200	58%
Dedicated logic registers	270400	10%
Block memory bits	16662528	0
DSP block 18-bit elements	576	0

SLM block, which is meaningful for the scalability of the large-scale SNN implementation and the reduction of the power consumption for digital neuromorphic systems. Second, the average throughput of the proposed FCL framework is 2.86∼14.38 times more than the algorithm presented by Liu *et al.* [62]. This high throughput is essential for the transmission of spikes between NoC nodes on digital neuromorphic systems. In comparison with the routing algorithm of SpiNNaker, it uses ARM cores for fault-tolerant routing, which is based on the von Neumann architecture for SNN computation [10]. The fault-tolerant algorithm used in SpiNNaker cannot be applied in NoC architectures. This limits the scalability and parallel computational capability on a single chip for digital neuromorphic systems. In contrast, our proposed FCL framework is implemented on FPGA, which is a parallel distributed architecture with higher computational efficiency. As shown in Table IV, although there are several digital neuromorphic systems with different types of neuron and synapse models and learning algorithms, none of them, except Spinnaker [10], considers the important issue of faults and designing a fault-tolerant system [10], [14], [18], [21], [28], [29], [35]–[43], [51]. Compared to Loihi, Truenorth, and BiCoSS, the proposed study solves the problem of the integration of learning and fault-tolerant capabilities, which is a fundamental and vital topic in the field of neuromorphic engineering [18], [58], [59]. Besides, to the best of our knowledge, there has been no previous implementation of a fault-tolerant framework for brain-inspired context-dependent learning. As shown in Figs. 12, 13, and 15, based on the proposed fault-tolerant framework, context-dependent learning can perform flawlessly while being affected by faulty nodes and regions of different sizes and locations.

The proposed high-performance fault-tolerant digital neuromorphic system helps conveniently prove any neuromorphic study concept. While our proposed system provides many advantages, which were shown above, it also has some limitations. One of the main limitations of our designed network model is the use of a simple LIF neuron and synapse model, which proved the concept of FCL. However, more complex neuronal and synaptic models can be implemented and studied using our proposed FCL system to investigate more complex neuroscience and artificial intelligence experiments. Besides, compared to analog neuromorphic systems, such as Neurogrid [11], the proposed system consumes higher power and occupies a larger area. While these are problems in the way of system scalability, the shorted FPGA design time compared to ASIC design time and the reconfigurability that FPGA-based design offers make our system very useful. In addition, other implementation technologies, such as analog neuromorphic chips and memristive designs, can be also investigated to implement the proposed FCL framework. Besides, due to its fault-tolerant capability, the proposed neuromorphic framework presents a versatile platform for the study of the neuronal mechanisms of many biologically inspired SNNs for cognitive behaviors, such as motor learning and visual recognition. It can also be explored to be applied in robotic decision-making tasks and other applications, including unmanned aerial vehicles and brain–machine interfaces.

TABLE V
COMPARISON OF THIS STUDY WITH STATE-OF-THE-ART DIGITAL NEUROMORPHIC SYSTEMS

Studies	Contribution	Learning	Fault-tolerant
Kim et al., 2012 [42]	Memristor synapse	Yes	No
Ambroise et al., 2013 [36]	Izhikevich network	No	No
Moore et al., 2012 [38]	Bluehive	Yes	No
Furber et al., 2014 [10]	SpiNNaker	Yes	Yes
Merolla et al., 2014 [18]	Truenorth	Yes	No
Yang et al., 2015 [28]	Basal ganglia	No	No
Qiao et al., 2015 [43]	ROLLS	Yes	No
Cheung et al., 2016 [39]	NeuroFlow	Yes	No
Luo et al., 2015 [29]	Cerebellar network	Yes	No
Kim et al., 2016 [37]	Neurocube	Yes	No
Pani et al., 2017 [40]	Izhikevich network	No	No
Yang et al., 2018 [14]	LaCSNN	Yes	No
Lammie et al., 2018 [64]	Triplet STDP	Yes	No
Yang et al., 2018 [35]	Dopamine network	No	No
Wang et al., 2018 [41]	Cortex simulator	No	No
Davies et al., 2018 [59]	Loihi	Yes	No
Yang et al., 2019 [21]	IBFT-based CMN	Yes	No
Heidarpur et al., 2019 [63]	CORDIC-SNN	Yes	No
Yang et al.,2021 [58]	BiCoSS	Yes	No
Yang et al., 2021 [61]	CerebelluMorphic	Yes	No
This study	FCL framework	Yes	Yes

VIII. CONCLUSION

This study presented a brain-inspired framework for context-dependent learning tasks implemented on the digital neuromorphic system, LaCSNN. The proposed framework uses SNN models for information processing and the STDP rule for learning. These are directly inspired by the mechanisms of biological hippocampus-mPFC networks. Experimental results show that context-dependent learning can be conducted in real time. In addition, a fault-tolerant spike routing algorithm was proposed to make the proposed neuromorphic system for context-dependent learning prone to faulty hardware. We have demonstrated that various fault scenarios cannot impact the learning capability of the proposed system, and the targeted context-dependent learning can be performed flawlessly. In addition, using the proposed novel routing method, the average latency and the maximum throughput of the system were shown to be significantly improved compared to previous routing strategies. Furthermore, unlike many previous digital neuromorphic systems shown in Table IV, the proposed system in this work is one of the few neuromorphic systems that have fault-tolerant capabilities. The proposed fault-tolerant spike routing scheme will be applied to other brain-inspired computing tasks in neuromorphic systems.

REFERENCES

[1] D. Shohamy and A. D. Wagner, "Integrating memories in the human brain: Hippocampal-midbrain encoding of overlapping events," *Neuron*, vol. 60, no. 2, pp. 378–389, Oct. 2008.

[2] C. J. MacDonald *et al.*, "Hippocampal 'time cells' bridge gap memory for discontiguous events," *Neuron*, vol. 71, no. 4, pp. 737–749, Aug. 2011.

[3] B. E. Pfeiffer and D. J. Foster, "Hippocampal place-cell sequences depict future paths to remembered goals," *Nature*, vol. 497, no. 7447, pp. 74–79, May 2013.

[4] R. W. Komorowski, J. R. Manns, and H. Eichenbaum, "Robust conjunctive item-place coding by hippocampal neurons parallels learning what happens where," *J. Neurosci.*, vol. 29, no. 31, pp. 9918–9929, Aug. 2009.

[5] O. Bichler *et al.*, "Pavlov's dog associative learning demonstrated on synaptic-like organic transistors," *Neural Comput.*, vol. 25, no. 2, pp. 549–566, Feb. 2013.

[6] E. R. Wood, P. A. Dudchenko, R. J. Robitsek, and H. Eichenbaum, "Hippocampal neurons encode information about different types of memory episodes occurring in the same location," *Neuron*, vol. 27, no. 3, pp. 623–633, Sep. 2000.

[7] J. C. Stanley *et al.*, "UK food standards agency workshop report: The effects of the dietary *n-6:n-3* fatty acid ratio on cardiovascular health," *Brit. J. Nutrition*, vol. 98, no. 6, pp. 1305–1310, Dec. 2007.

[8] J. Ferbinteanu and M. L. Shapiro, "Prospective and retrospective memory coding in the hippocampus," *Neuron*, vol. 40, no. 6, pp. 1227–1239, Dec. 2003.

[9] I. Lee and F. Solivan, "The roles of the medial prefrontal cortex and hippocampus in a spatial paired-association task," *Learn. Memory*, vol. 15, no. 5, pp. 357–367, May 2008.

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

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

[12] M. A. Petrovici *et al.*, "Characterization and compensation of network-level anomalies in mixed-signal neuromorphic modeling platforms," *PLoS ONE*, vol. 9, no. 10, Oct. 2014, Art. no. e108590.

[13] J. Park, T. Yu, S. Joshi, C. Maier, and G. Cauwenberghs, "Hierarchical address event routing for reconfigurable large-scale neuromorphic systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2408–2422, Oct. 2017.

[14] S. Yang *et al.*, "Real-time neuromorphic system for large-scale conductance-based spiking neural networks," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2490–2503, Jul. 2019.

[15] A. Mortara, E. A. Vittoz, and P. Venier, "A communication scheme for analog VLSI perceptive systems," *IEEE J. Solid-State Circuits*, vol. 30, no. 6, pp. 660–669, Jun. 1995.

[16] J. Lazzaro and J. Wawrzynek, "A multi-sender asynchronous extension to the AER protocol," in *Proc. 16th Conf. Adv. Res. VLSI*, Mar. 1995, pp. 158–169.

[17] S. Carrillo *et al.*, "Scalable hierarchical network-on-chip architecture for spiking neural network hardware implementations," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 12, pp. 2451–2461, Dec. 2013.

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

[19] J. Pei *et al.*, "Towards artificial general intelligence with hybrid Tianjic chip architecture," *Nature*, vol. 572, no. 7767, pp. 106–111, Jul. 2019.

[20] T. H. Vu and A. B. Abdallah, "Low-latency K-means based multicast routing algorithm and architecture for three dimensional spiking neuromorphic chips," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2019, pp. 1–8.

[21] S. Yang *et al.*, "Scalable digital neuromorphic architecture for large-scale biophysically meaningful neural network with multi-compartment neurons," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 148–162, Jan. 2020.

[22] J. Wu and S. Furber, "A multicast routing scheme for a universal spiking neural network architecture," *Comput. J.*, vol. 53, no. 3, pp. 280–288, Apr. 2009.

[23] T. H. Vu, O. M. Ikechukwu, and A. B. Abdallah, "Fault-tolerant spike routing algorithm and architecture for three dimensional NoC-based neuromorphic systems," *IEEE Access*, vol. 7, pp. 90436–90452, Jun. 2019.

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

[25] M. C. Hagen, O. Franzén, F. McGlone, G. Essick, C. Dancer, and J. V. Pardo, "Tactile motion activates the human middle temporal/V5 (MT/V5) complex," *Eur. J. Neurosci.*, vol. 16, no. 5, pp. 957–964, Sep. 2002.

[26] C. Zamarreno-Ramos, A. Linares-Barranco, T. Serrano-Gotarredona, and B. Linares-Barranco, "Multicasting mesh AER: A scalable assembly approach for reconfigurable neuromorphic structured AER Systems. Application to ConvNets," *IEEE Trans. Biomed. Circuits Syst.*, vol. 7, no. 1, pp. 82–102, Feb. 2013.

[27] A. B. Abdallah, *Advanced Multicore Systems-on-Chip: Architecture, on-Chip Network, Design*. Singapore: Springer, 2018.

[28] S. Yang *et al.*, "Cost-efficient FPGA implementation of basal ganglia and their parkinsonian analysis," *Neural Netw.*, vol. 71, pp. 62–75, Nov. 2015.

[29] J. Luo, G. Coapes, T. Mak, T. Yamazaki, C. Tin, and P. Degenaar, "Real-time simulation of passage-of-time encoding in cerebellum using a scalable FPGA-based system," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 3, pp. 742–753, Jun. 2016.

[30] S. Yang, X. Wei, B. Deng, C. Liu, H. Li, and J. Wang, "Efficient digital implementation of a conductance-based globus pallidus neuron and the dynamics analysis," *Phys. A, Stat. Mech. Appl.*, vol. 494, pp. 484–502, Mar. 2018.

[31] K. Akbarzadeh-Sherbaf, B. Abdoli, S. Safari, and A.-H. Vahabie, "A scalable FPGA architecture for randomly connected networks of Hodgkin–Huxley neurons," *Frontiers Neurosci.*, vol. 12, p. 698, Oct. 2018.

[32] S. Yang *et al.*, "Digital implementations of thalamocortical neuron models and its application in thalamocortical control using FPGA for Parkinson's disease," *Neurocomputing*, vol. 177, pp. 274–289, Feb. 2016.

[33] B. Sen-Bhattacharya *et al.*, "A spiking neural network model of the lateral geniculate nucleus on the SpiNNaker machine," *Frontiers Neurosci.*, vol. 11, p. 454, Aug. 2017.

[34] S. Yang *et al.*, "Efficient implementation of a real-time estimation system for thalamocortical hidden parkinsonian properties," *Sci. Rep.*, vol. 7, no. 1, pp. 1–16, Mar. 2017.

[35] S. Yang *et al.*, "Cost-efficient FPGA implementation of a biologically plausible dopamine neural network and its application," *Neurocomputing*, vol. 314, pp. 394–408, Nov. 2018.

[36] M. Ambroise, T. Levi, Y. Bornat, and S. Saighi, "Biorealistic spiking neural network on FPGA," in *Proc. 47th Annu. Conf. Inf. Sci. Syst. (CISS)*, Oct. 2013, pp. 1–6.

[37] D. Kim *et al.*, "Neurocube: A programmable digital neuromorphic architecture with high-density 3D memory," in *Proc. 43rd Annu. Int. Sym. Comput. Archit.*, 2016, pp. 380–392.

[38] S. W. Moore *et al.*, "Bluehive—A field-programable custom computing machine for extreme-scale real-time neural network simulation," in *Proc. IEEE 20th Int. Symp. Field-Program. Custom Comput. Mach.*, May 2012, pp. 133–140.

[39] K. Cheung, S. R. Schultz, and W. Luk, "NeuroFlow: A general purpose spiking neural network simulation platform using customizable processors," *Frontiers Neurosci.*, vol. 9, p. 516, Jan. 2016.

[40] D. Pani, P. Meloni, G. Tuveri, F. Palumbo, P. Massobrio, and L. Raffo, "An FPGA platform for real-time simulation of spiking neuronal networks," *Frontiers Neurosci.*, vol. 11, p. 90, Feb. 2017.

[41] R. M. Wang, C. S. Thakur, and A. van Schaik, "An FPGA-based massively parallel neuromorphic cortex simulator," *Frontiers Neurosci.*, vol. 12, p. 213, Apr. 2018.

[42] Y. Kim, Y. Zhang, and P. Li, "A digital neuromorphic VLSI architecture with memristor crossbar synaptic array for machine learning," in *Proc. IEEE Int. SOC Conf.*, Sep. 2012, pp. 328–333.

[43] N. Qiao *et al.*, "A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128 k synapses," *Frontiers Neurosci.*, vol. 9, p. 141, Apr. 2015.

[44] W. J. Dally and C. L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Trans. Comput.*, vol. COM-36, no. 5, pp. 547–553, May 1987.

[45] S. Yang, T. Gao, J. Wang, B. Deng, B. Lansdell, and B. Linares-Barranco, "Efficient spike-driven learning with dendritic event-based processing," *Frontiers Neurosci.*, vol. 15, p. 97, Feb. 2021.

[46] Z. Zhang, A. Greiner, and S. Taktak, "A reconfigurable routing algorithm for a fault-tolerant 2D-mesh network-on-chip," in *Proc. 45th Annu. Conf. Design Autom. (DAC)*, 2008, pp. 441–446.

[47] A. Neckar *et al.*, "Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model," *Proc. IEEE*, vol. 107, no. 1, pp. 144–164, Jan. 2019.

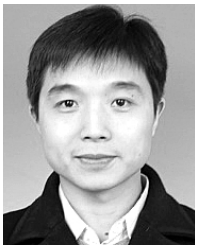
[48] K. Chen and G. Chiu, "Fault-tolerant routing algorithm for meshes without using virtual channels," *J. Inform. Sci. Eng.*, vol. 14, no. 4, pp. 765–783, 1998.
[49] S. Yang *et al.*, "Design of hidden-property-based variable universe fuzzy control for movement disorders and its efficient reconfigurable implementation," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 2, pp. 304–318, Feb. 2019.
[50] S. Yang *et al.*, "Efficient hardware implementation of the subthalamic nucleus–external globus pallidus oscillation system and its dynamics investigation," *Neural Netw.*, vol. 94, pp. 220–238, Oct. 2017.
[51] H. Asgari, B. M.-N. Maybodi, M. Payvand, and M. R. Azghadi, "Low-energy and fast spiking neural network for context-dependent learning on FPGA," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 11, pp. 2697–2701, Nov. 2020.
[52] R. A. Gulli *et al.*, "Context-dependent representations of objects and space in the primate hippocampus during virtual navigation," *Nature Neurosci.*, vol. 23, no. 1, pp. 103–112, Jan. 2020.
[53] X. Zhao, Y. Wang, N. Spruston, and J. C. Magee, "Membrane potential dynamics underlying context-dependent sensory responses in the hippocampus," *Nature Neurosci.*, vol. 23, no. 7, pp. 881–891, Jul. 2020.
[54] I. Lee, D. Yoganarasimha, G. Rao, and J. J. Knierim, "Comparison of population coherence of place cells in hippocampal subfields CA1 and CA3," *Nature*, vol. 430, no. 6998, pp. 456–459, Jul. 2004.
[55] J. Waider *et al.*, "Serotonin deficiency increases context-dependent fear learning through modulation of hippocampal activity," *Frontiers Neurosci.*, vol. 13, p. 245, Apr. 2019.
[56] J. F. Guzowski, J. J. Knierim, and E. I. Moser, "Ensemble dynamics of hippocampal regions CA3 and CA1," *Neuron*, vol. 44, no. 4, pp. 581–584, Nov. 2004.
[57] R. S. Ross, K. R. Sherrill, and C. E. Stern, "The hippocampus is functionally connected to the striatum and orbitofrontal cortex during context dependent decision making," *Brain Res.*, vol. 1423, pp. 53–66, Nov. 2011.
[58] S. Yang *et al.*, "BiCoSS: Toward large-scale cognition brain with multigranular neuromorphic architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 11, 2021, doi: 10.1109/TNNLS.2020.3045492.
[59] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018.
[60] S. Yang *et al.*, "FPGA implementation of hippocampal spiking network and its real-time simulation on dynamical neuromodulation of oscillations," *Neurocomputing*, vol. 282, pp. 262–276, Mar. 2018.
[61] S. Yang, J. Wang, N. Zhang, B. Deng, Y. Pang, and M. R. Azghadi, "CerebelluMorphic: Large-scale neuromorphic model and architecture for supervised motor learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 23, 2021, doi: 10.1109/TNNLS.2021.3057070.
[62] J. Liu, J. Harkin, Y. Li, and L. P. Maguire, "Fault-tolerant networks-on-chip routing with coarse and fine-grained look-ahead," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 2, pp. 260–273, Feb. 2016.
[63] M. Heidarpur, A. Ahmadi, M. Ahmadi, and M. R. Azghadi, "CORDIC-SNN: On-FPGA STDP learning with Izhikevich neurons," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 7, pp. 2651–2661, Jul. 2019.
[64] C. Lammie, T. J. Hamilton, A. van Schaik, and M. R. Azghadi, "Efficient FPGA implementations of pair and triplet-based STDP for neuromorphic architectures," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 4, pp. 1558–1570, Apr. 2019.

Shuangming Yang (Member, IEEE) received the M.S. and Ph.D. degrees from Tianjin University, Tianjin, China, in 2016 and 2020, respectively.

He is currently an Assistant Professor with the School of Electrical and Information Engineering, Tianjin University. His research interests include neuromorphic engineering, computational neuroscience, neuroscience-inspired intelligence, brain-inspired computing, and deep learning.

Jiang Wang was born in China, in 1964. He received the master's degree in power and automation engineering and the Ph.D. degree from Tianjin University, Tianjin, China, in 1989 and 1996, respectively.

He is currently a Professor with the School of Electrical and Information Engineering, Tianjin University. His research interests are nonlinear dynamical systems, neuroscience, and information processing and detecting.



Bin Deng (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Tianjin University, Tianjin, China, in 2007.

He is currently a Professor with the School of Electrical and Information Engineering, Tianjin University. His research interests include the dynamic analysis of neuron models and the nonlinear analysis of neuron electrical information.

Mostafa Rahimi Azghadi (Senior Member, IEEE) received the Ph.D. degree in electrical and electronic engineering from The University of Adelaide, Adelaide, SA, Australia, in 2014.

From 2012 to 2014, he was a Visiting Ph.D. Student with the Neuromorphic Cognitive System Group, Institute of Neuroinformatics, University and Swiss Federal Institute of Technology (ETH), Zürich, Switzerland. He is currently a Senior Lecturer with the College of Science and Engineering, James Cook University, Townsville, QLD, Australia, where he is researching neuromorphic engineering and brain-inspired architectures and developing custom hardware and software solutions for a variety of engineering applications ranging from medical imaging to precision agriculture.

Dr. Azghadi was a recipient of several national and international awards and scholarships, such as the 2020 JCU Award for Excellence in Innovation and Change, the Queensland Young Tall Poppy Science Award in 2017, and the South Australia Science Excellence Awards in 2015. He was a recipient of the Doctoral Research Medal and the Adelaide University Alumni Medal in 2014.

Bernabe Linares-Barranco (Fellow, IEEE) received the B.S. degree in electronic physics, the M.S. degree in microelectronics, and the Ph.D. degree from the University of Seville, Seville, Spain, in 1986, 1987, and 1990, respectively, and the Ph.D. degree from Texas A\&M University, College Station, TX, USA, in 1991.

Since 1991, he has been with the Sevilla Microelectronics Institute (IMSE-CNM), Seville, from the Spanish Research Council (CSIC) of Spain, where he is currently a Full Professor of Research and has been serving as the Director since February 2018. He has been a Visiting Professor/Fellow with Johns Hopkins University, Baltimore, MD, USA, Texas A&M University, The University of Manchester, Manchester, U.K., and the University of Lincoln, Lincoln, U.K. His recent interests are in address-event-representation (AER) VLSI, real-time AER vision sensing and processing chips, memristor circuits, and extending AER to the nanoscale.

Dr. Linares-Barranco received two IEEE Transactions Best Paper Awards. From 2011 to 2013, he was the Chair of the IEEE Circuits and Systems Society Spain Chapter. He has been an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II and *Frontiers in Neuromorphic Engineering*.