# Final Scientific/Technical Report

**Project Title**:
Real‑Time Neuromorphic Processing of Spatiotemporal Data for Scientific Discovery
**Principle Investigator**: Peng Li (lip@ucsb.edu)
**Recipient Organization**: Regents of the University of California, Santa Barbara
**DOE Award Number**:  DE-SC0021319
**Sponsoring Program Office**: Office of Advanced Scientific Computing Research
**Period of Performance**: 09/01/2020-08/31/2023

## Abstract

Spiking Neural Networks (SNNs) are brain-inspired computing models incorporating unique temporal dynamics and event-driven processing. Rich dynamics in both space and time offer great challenges and opportunities for efficient processing of sparse spatiotemporal data compared with conventional artificial neural networks (ANNs).

Under this context, the goal of this project is to develop spiking neural network based neuromorphic computing to enable energy-efficient real-time learning and processing of spatiotemporal data. This report summarizes the key results on network architecture design, training methods, and SNN hardware acceleration achieved under this project, demonstrating the promise of spiking neural networks.

## 1. Project Objectives

The overall objective of this work to develop a spike-based analog neuromorphic computing framework to enable energy-efficient real-time learning and processing of spatiotemporal data to accelerate scientific discovery.

Broadly speaking, we attempt to address the following research challenges and needs:

1) Spiking neural network (SNN) architecture of computation:  neurally-inspired SNN models are often hand tuned for specific tasks; Lack of unifying SNN architectures hampers the application of neuromorphic computing to diverse domains including scientific discovery;

2) High-performance training: existing SNN training algorithms such as biologically-plausible spike-timing dependent plasticity, which lack a globally-defined learning objective, are unable to deliver competitive performance for challenging learning tasks.

3) Energy-efficient high-performance SNN hardware accelerators: in addition to overcoming the challenges in 1) and 2), it is desirable to develop highly efficient hardware accelerators to expedite spike-based workloads.

## 2. Main Findings of the Project

In the context of the above objectives, the research team has had developed a sequence of techniques, centering around the following key aspects.

### 2.1 Recurrent SNN compute fabrics and network architectures

It is instrumental to design recurrent fabrics that can be replicated for building large SNNs with good performance. This will avoid designing recurrent SNNs with randomly generated connectivity patterns, a current ad-hoc practice that does not guarantee good performance. Such fabrics shall possess high spatiotemporal computing power, be small in size, and inter-fabric connections shall be made in a structured manner to mitigate training difficulties.

Towards this end, first, we propose a new type of RSNNs called Skip-Connected Self-Recurrent SNNs (ScSr-SNNs) [1], as shown in Fig. 1. Recurrence in ScSr-SNNs is introduced in a stereotyped manner by adding self-recurrent connections to spiking neurons. In some sense, here the basic recurrent fabric consists of a simple spiking neuron with a self-recurrent connection to itself. In terms of inter-fabric connectivity, we use feedforward connections to wire up multiple layers each consisting of a set of the basic self-recurrent fabrics. The SNNs with self-recurrent connections can realize recurrent behaviors similar to those of more complex RSNNs while the error gradients can be more straightforwardly calculated due to the mostly feedforward nature of the network. The network dynamics is enriched by skip connections between nonadjacent layers.

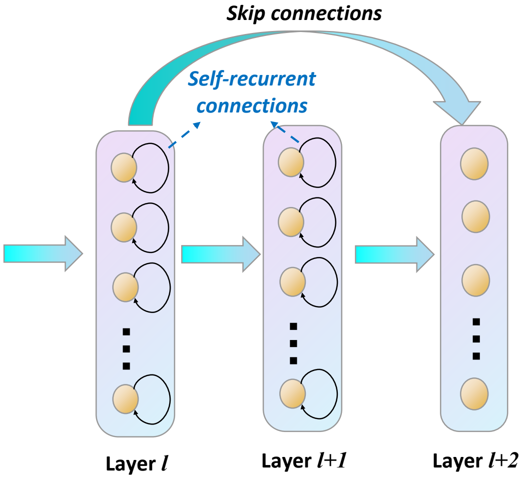


**Fig. 1. Skip-Connected Self-Recurrent SNNs (ScSr-SNN).**

Based on challenging speech, neuromorphic speech, and neuromorphic image datasets, the proposed ScSr-SNNs have been shown to outperform other types of recurrent SNNs reported before, for example on the widely used DVS-guesture recognition dataset as shown in Table 1.

| Method | Network | Performance |
|---|---|---|
| TrueNorth[6] | CNN-based 16 layers | 91.77% |
| Slayer[7] | CNN-based 8 layers | 93.64% |
| RNN[8] | CNN-based | 92.01% |
| LSTM[8] | CNN-based | 93.75% |
| SNN[3] | CNN-based 8 layers | 93.40% |
| ScSr-SNN) | CNN-based 8 layers | **95.49%** |

**Table 1: SNN model acurracies of several different models trained over the neuromorphic video dataset DVS-Gesture.**

Second, we further propose a novel recurrent structure called the Laterally-Inhibited Self-Recurrent Unit (LISR), which consists of one excitatory neuron with a self-recurrent connection wired together with an inhibitory neuron through excitatory and inhibitory synapses [2]. The self-recurrent connection of the excitatory neuron mitigates the information loss caused by the firing-and-resetting mechanism and maintains the long-term neuronal memory. The lateral inhibition from the inhibitory neuron to the corresponding excitatory neuron, on the one hand, adjusts the firing activity of the latter. On the other hand, it plays as a forget gate to clear the memory of the excitatory neuron. The LISR units can be leveraged to realize recurrent SNNs as illustrated in Fig. 2.

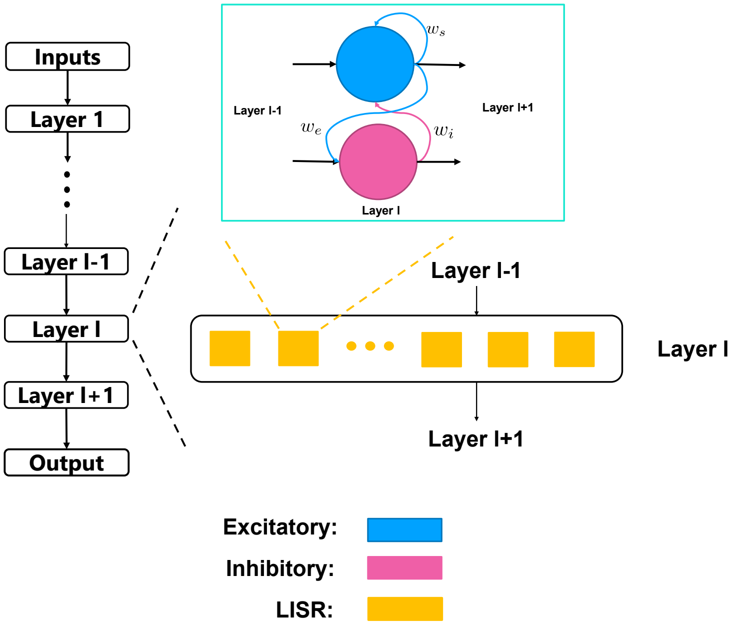


**Fig. 2. Recurrent SNNs based upon the proposed LISR units.**

The excellent performance of the proposed LISR has been reported in [2].

## 2.2 High-Accuracy SNN Training Algorithms

For training the proposed Skip-Connected Self-Recurrent SNNs (ScSr-SNNs), we propose a new backpropagation (BP) method called backpropagated intrinsic plasticity (BIP) to further boost the performance of ScSr-SNNs by training intrinsic model parameters [1]. Unlike standard intrinsic plasticity rules that adjust the neuron's intrinsic parameters according to neuronal activity, the proposed BIP method optimizes intrinsic parameters based on the backpropagated error gradient of a well-defined global loss function in addition to synaptic weight training. By comprehensive benchmarking, the proposed ScSr-SNNs can boost performance by up to 2.85% compared with other types of RSNNs trained by state-of-the-art BP methods.

Furthermore, we adapted a BP method developed recently by our team to specifically train recurrent SNNs based on the proposed LISR unit, which improves learning performance significantly by up to 9.26% over feedforward SNNs with similar computational costs on a set of speech and image datasets [2].

## 2.3 SNN Training Methods with Reduced Complexity for Deployment on Neuromorphic Hardware

While promising backpropagation (BP) methods have been developed for SNNs, they tend to be either not biologically plausible or to be computationally complex. We study two biologically plausible alternatives to backpropagation while retaining high temporal precision for SNN training. These two methods, namely TSSL-DFA and TSSL-KP, are extensions to direct feedback alignment (DFA) and a method by Kollen-Pollack (KP), respectively.

TSSL-KP and TSSL-DFA are for SNN training and incorporate recent BP-based gradient computation techniques and additional simplifications [3], as illustrated in Fig. 3.

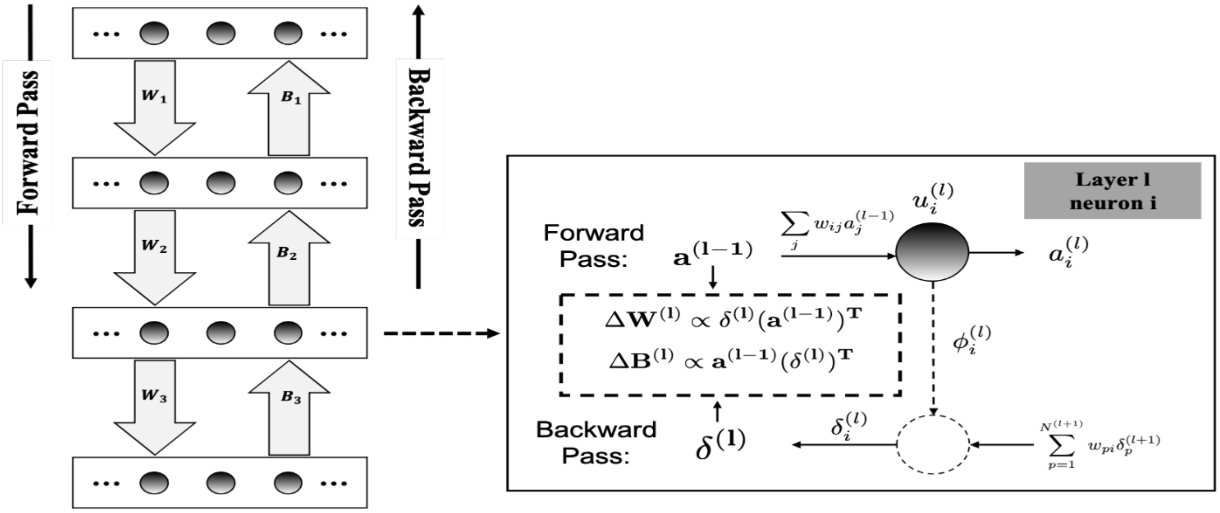

Forward Pass
Backward Pass
$W_1$
$B_1$
$W_2$
$B_2$
$W_3$
$B_3$
Layer l
neuron i
Forward Pass:
$\mathbf{a}^{(\mathbf{l-1})}$
$\sum_j w_{ij} a_j^{(l-1)}$
$u_i^{(l)}$
$a_i^{(l)}$
$\Delta \mathbf{W}^{(\mathbf{l})} \propto \delta^{(\mathbf{l})} (\mathbf{a}^{(\mathbf{l-1})})^{\mathbf{T}}$
$\Delta \mathbf{B}^{(\mathbf{l})} \propto \mathbf{a}^{(\mathbf{l-1})} (\delta^{(\mathbf{l})})^{\mathbf{T}}$
$\phi_i^{(l)}$
Backward Pass:
$\delta^{(\mathbf{l})}$
$\delta_i^{(l)}$
$\sum_{p=1}^{N^{(l+1)}} w_{pi} \delta_p^{(l+1)}$

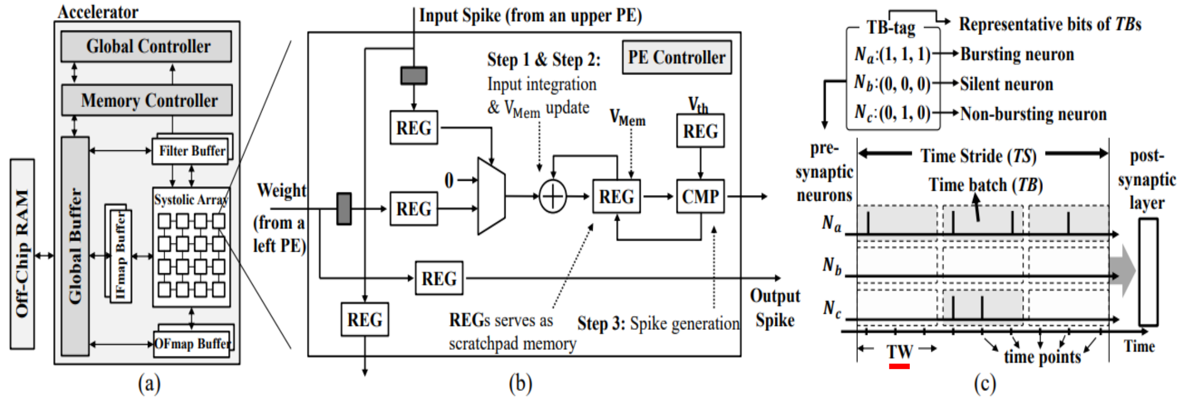
**Fig. 3 Proposed low-complexity SNN training methods TSSL-KP and TSSL-DFA.**

We show that both methods can near the accuracy of a state-of the-art BP method while maintaining biological plausibility, and in the case of TSSL-DFA greatly reducing the complexity of the required feedback algorithms. We assess the complexity of these algorithms to show their usefulness under neuromorphic hardware [3].

## 2.4 Efficient Systolic-Array SNN Hardware Accelerator Architecture

Spiking Neural Networks (SNNs) are brain-inspired computing models incorporating unique temporal dynamics and event-driven processing. Rich dynamics in both space and time offer great challenges and opportunities for efficient processing of sparse spatiotemporal data compared with conventional artificial neural networks (ANNs). Specifically, the additional overheads for handling the added temporal dimension limit the computational capabilities of neuromorphic accelerators. Iterative processing at every time-point with sparse inputs in a temporally sequential manner not only degrades the utilization of the systolic array but also intensifies data movement.

We propose a novel technique and architecture, called parallel time batching (PTB), that significantly improve utilization and data movement while efficiently handling temporal sparsity of SNNs on systolic arrays. As illustrated in Fig. 4, unlike time-sequential processing in conventional SNN accelerators, we pack multiple time points into a single time window (TW) and process the computations induced by active synaptic inputs falling under several TWs in parallel, leading to the proposed parallel time batching. It allows weight reuse across multiple time points and enhances the utilization of the systolic array with reduced idling of processing elements, overcoming the irregularity of sparse firing activities. We optimize the granularity of time-domain processing, i.e., the TW size, which significantly impacts the data reuse and utilization. We further boost the utilization efficiency by simultaneously scheduling non-overlapping sparse spiking activities onto the array.



(a): Overall architecture. (b): Simplified schematic representation of the processing element (PE) in systolic array. (c): Schematic representation of *time point*, *time batch* (*TB*), *TB*-tag and *time stride* (*TS*).

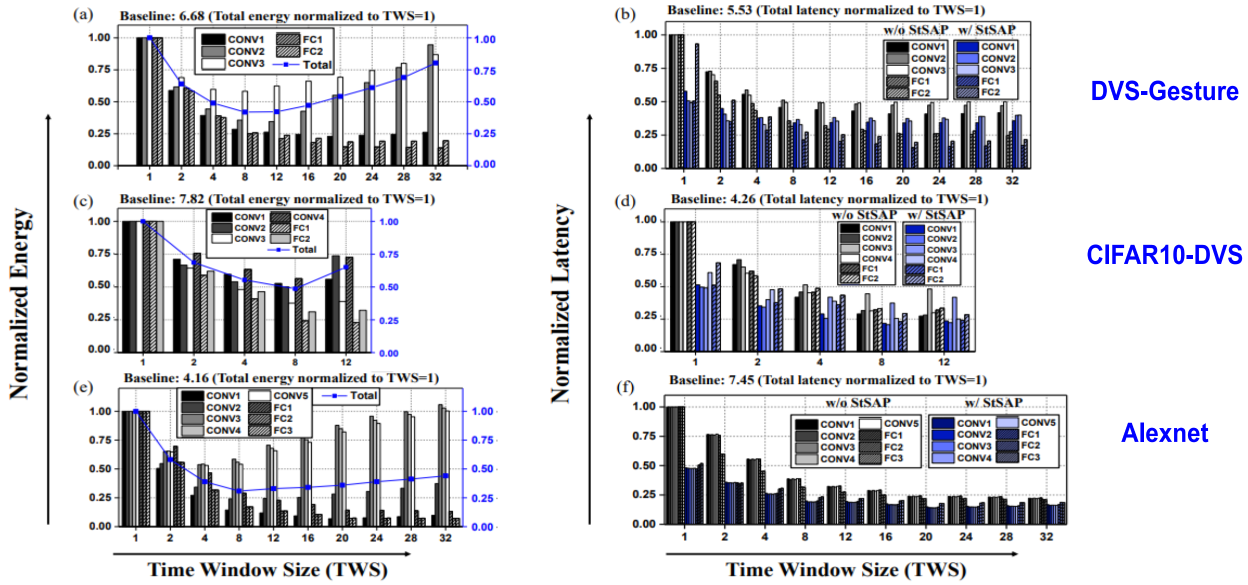
**Time stride(TS):** Full range of time points
**Time window(TW):** One TW packs the firing activity of one (active) synaptic neuron over multiple time points.
**Time batch(TB):** basic unit of workload assignable to a PE.

❑ **PTB**

- **Assigns firing activities in a TW to PEs**
- **Process multiple time batches with a temporal granularity defined by TW size**

**Fig. 4. Proposed Parallel Time Batching (PTB) Architecture for spiking neural networks.**

As shown in Fig. 5, the proposed architectures offer a unifying solution for general spiking neural networks with commonly exhibited temporal sparsity, a key challenge in hardware acceleration, delivering 248X energy-delay product (EDP) improvement on average compared to an SNN baseline for accelerating various networks. Compared to ANN based accelerators, our approach improves EDP by 47X on the CIFAR10 dataset.



248X energy-delay product (EDP) improvement
over  a baseline systolic array architecture

**Fig. 5 Improvements on energy dissipation and latency of the Batching (PTB) architecture on three SNN models.**

## 3. Summary and Ongoing and Future Work

Biologically-inspired neuromorphic computing provides exciting opportunities for advancing the field of machine learning and computing.

While demonstrating the promise of neuromorphic computing via development of several novel network architectures, training methods, and dedicated hardware in this project, our ongoing and future work will explore several research fronts. We will explore automated spiking neural architecture search, specifically for optimizing the network architecture of complex recurrent SNNs [9]. We will also tap into the promise of emerging large spiking transformer models, which can be widely adopted to process language, video, and other types of spatiotemporal data, by developing efficient quantization methods, aiming at dramatically reducing the computation and storage overheads of such large SNN models [10].

## References

[1] W. Zhang and P. Li, "Skip-Connected Self-Recurrent Spiking Neural Networks with Joint Intrinsic Parameter and Synaptic Weight Training," Neural Computation, (2021).

[2] W. Zhang and P. Li, "Spiking Neural Networks with Laterally-Inhibited Self-Recurrent Units," International Joint Conference on Neural Networks (IJCNN) (2021).

[3] R. Boone, W. Zhang, and P. Li, "Efficient Biologically-Plausible Training of Spiking Neural Networks with Precise Timing," International Conference on Neuromorphic Systems (ICONS) (2021).

[4] J-J. Lee and P. Li, "Parallel Time Batching: Systolic-Array Acceleration of Sparse Spiking Neural Computation," The 28th IEEE International Symposium on High-Performance Computer Architecture (HPCA) (2022).

[5] Zhang, W., & Li, P. (2020). Temporal Spike Sequence Learning via Backpropagation for Deep Spiking Neural Networks. Advances in Neural Information Processing Systems 33 pre-proceedings (NeurIPS 2020)

[6] Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., ... & Modha, D. (2017). A low power, fully event-based gesture recognition system. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7243-7252).

[7] Sumit Bam Shrestha and Garrick Orchard. Slayer: Spike layer error reassignment in time. In Advances in Neural Information Processing Systems, pages 1412–1421, 2018.

[8] He, W., Wu, Y., Deng, L., Li, G., Wang, H., Tian, Y., ... & Xie, Y. (2020). Comparing SNNs and RNNs on neuromorphic vision datasets: Similarities and differences. Neural Networks, 132, 108-120.

[9] W. Zhang, H. Geng, and P. Li, "Composing Recurrent Spiking Neural Networks using Locally-Recurrent Motifs and Risk-Mitigating Architectural Optimization," https://arxiv.org/abs/2108.01793.

[10] B. Xu, Y. Song, and P. Li, "Trimming Down Large Spiking Transformers via Heterogeneous Quantization," 2024 (under review).

## Acknowledgement

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award Number DE-SC0021319.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof.

The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.