清華大學
Tsinghua University School of Medicine
Tsinghua Laboratory of Brain and Intelligence

Evolving Connectivity for Recurrent Spiking Neural Networks

■ Presenter: Huimiao Chen

2023-10-25

CONTENTS

Evolving Connectivity for Recurrent Spiking Neural Networks

KEY WORDS:

Brain-like Computing;
Spiking Neural Networks;
Computational Neuroscience;
Neuromorphic Computing;
Artificial General Intelligence.

01 Background

01 BACKGROUND – Energy-consuming AI

An Elephant in the room!

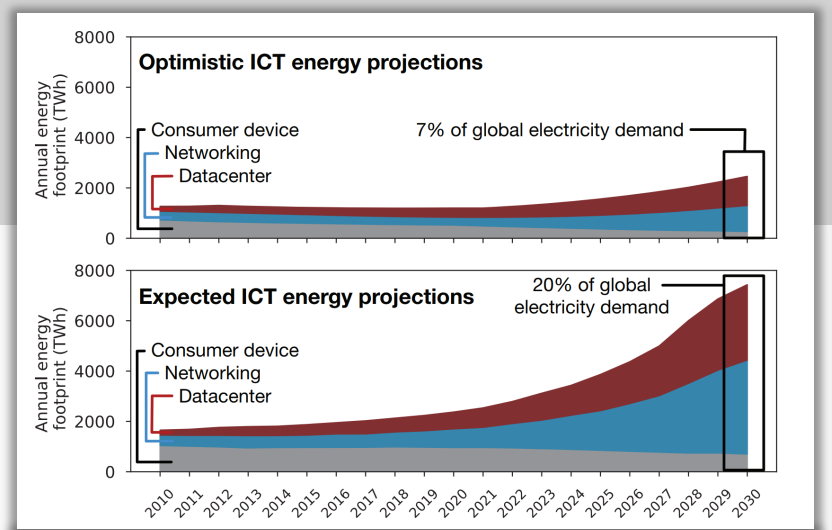
1. ICT industry by 2030 will account for between 7% to 20% of the total global demand.

2. Data centers are one of the top 10 water-consuming commercial industries US.

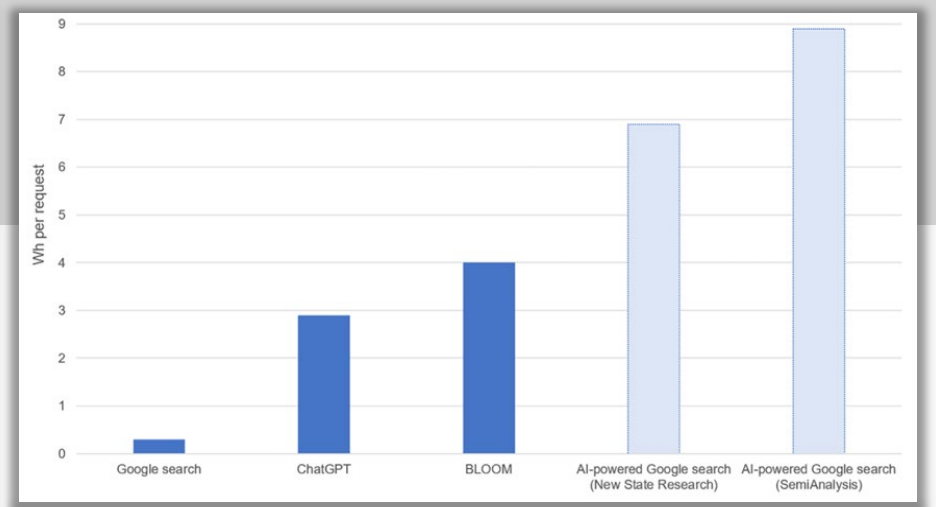
3. 300 requests of ChatGPT use 1 kWh energy; other chatbot can be even high.

The China Association for Science and Technology recently released the **Major Scientific Questions of 2023**, with the first cutting-edge scientific questions being:

How to achieve low-energy artificial intelligence?



Information and Computing Tech (ICT) Electricity Demand



Energy Consumption per Request of Chatbots and Standard Google Search

01 BACKGROUND – Neuron Models

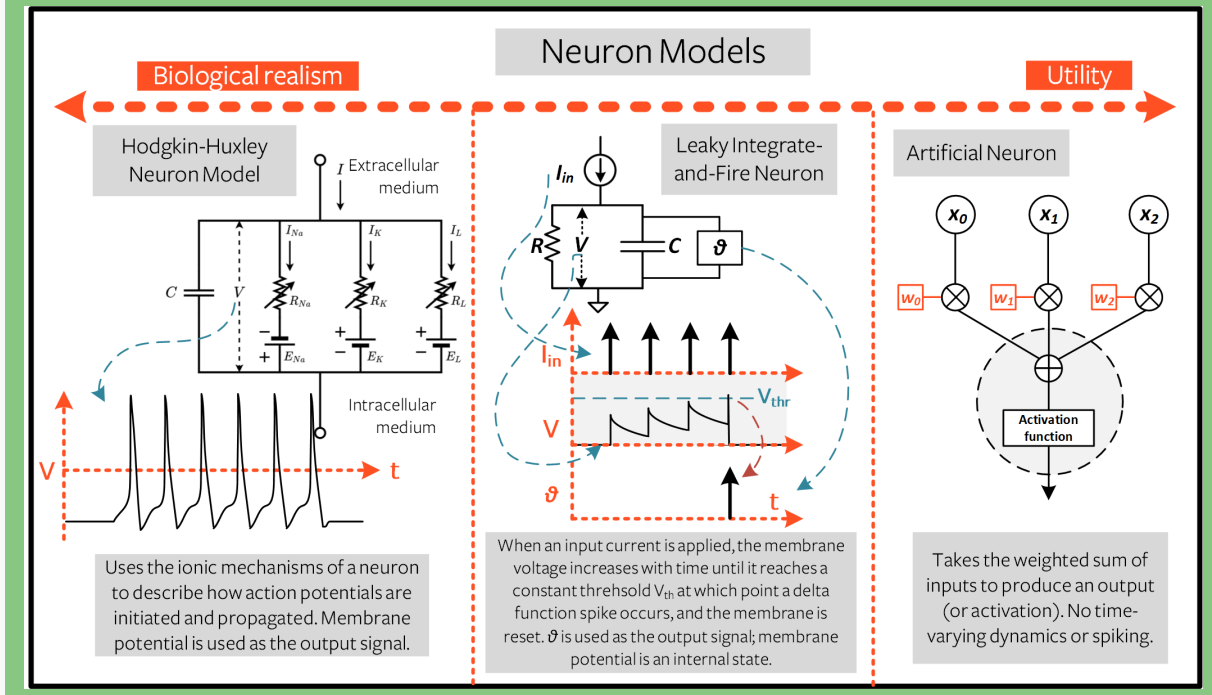
HH model is the most influential realistic model; LIF is simplified; AI model is different.

1. Continuous Activity: AI models are active continuously, while biological neurons spike only when necessary, conserving energy.

2. Complex Activation Functions: AI models employ intricate activation functions, demanding more computational resources than biological neurons' simpler ion channel dynamics.

3. Backpropagation: AI models rely on computationally intensive backpropagation for training, unlike the distributed, energy-efficient learning mechanisms in biological systems.

4. Hardware: Energy efficiency in biological brains arises from both neural models and specialized hardware, like neuromorphic processors, emulating the brain's event-driven nature.



HH model vs LIF model vs AI neuron model

01 BACKGROUND – Spiking Neural Networks

Networks of spiking neurons: the third generation of neural network models, 1997.

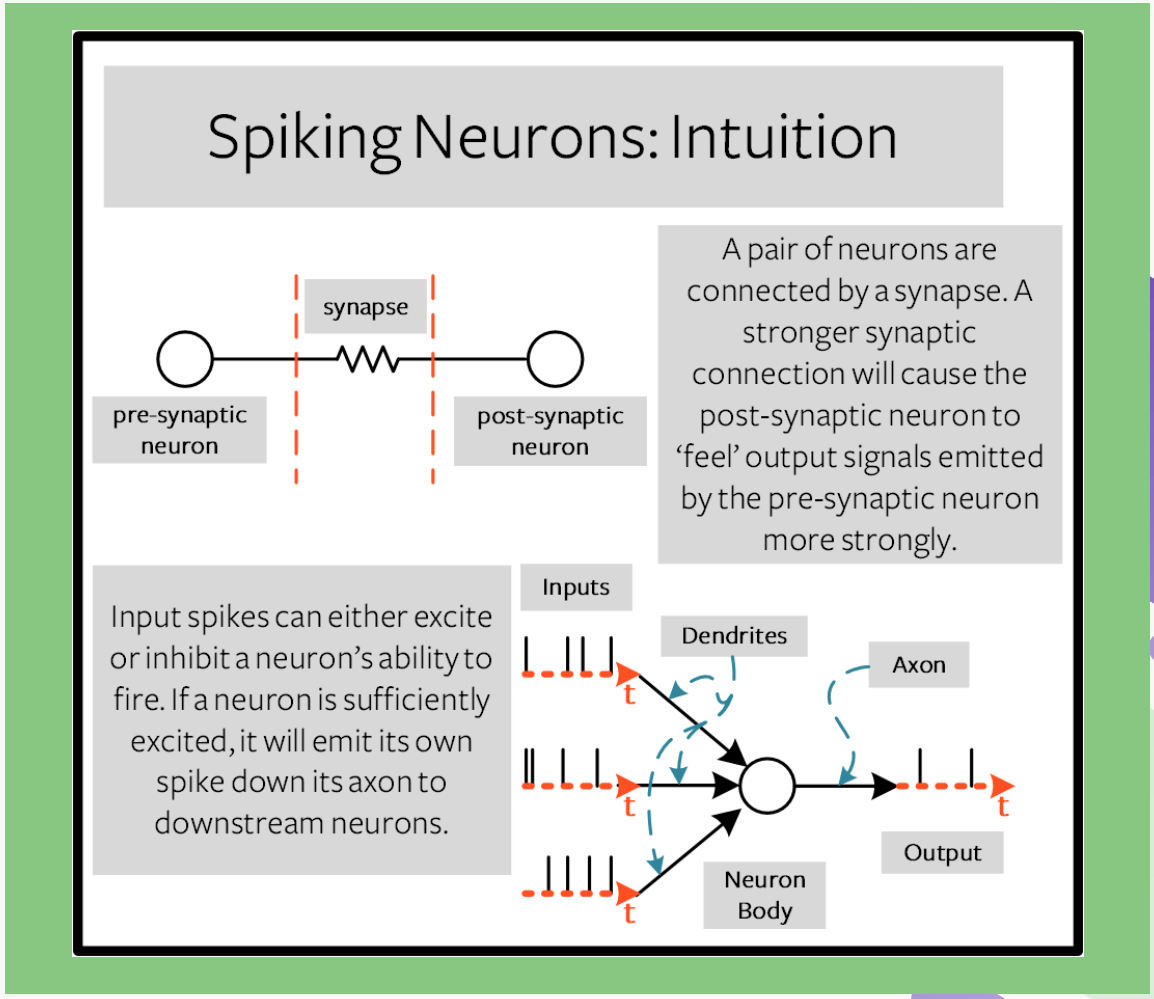
Energy-Efficiency: SNNs are energy-efficient, ideal for edge devices and IoT.

Biological Plausibility: They mimic real neuron behavior, enhancing our understanding.

Temporal Processing: SNNs excel at handling temporal patterns for tasks like speech recognition.

Sparse Data Representation: They naturally create sparse spiking patterns, aiding information encoding and memory efficiency.

Neuromorphic Hardware: It has the potential to revolutionize brain-inspired computing.



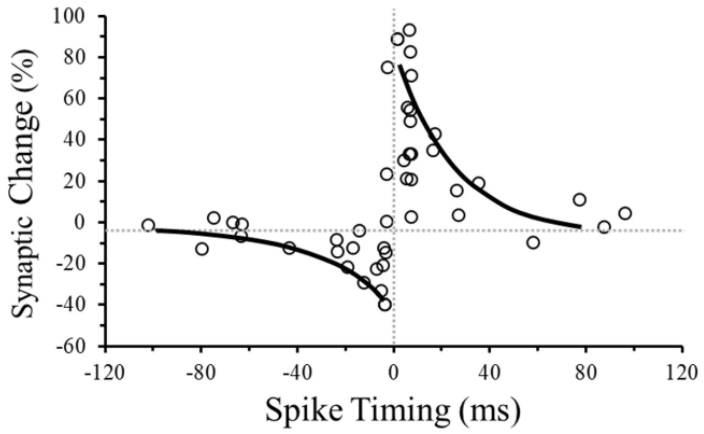
02 Related Work

02 RELATED WORK – SNN Learning

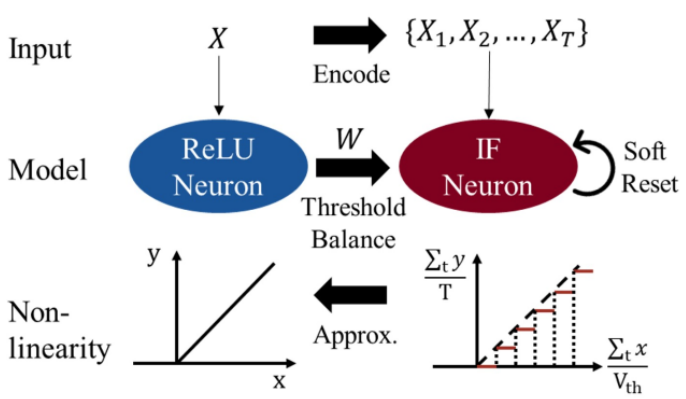
Bio-inspired Learning: SNNs can be trained using unsupervised methods inspired by biology, like Hebbian learning and STDP. These techniques adapt weights using local information, making them suitable for distributed computing and online learning.

Conversion from ANNs: ANNs can be transformed into SNNs for training. This method leverages neuromorphic computing and established deep learning techniques by approximating ANNs' activation values with SNNs' firing rates. It has achieved accuracy comparable to ANNs but faces challenges of latency.

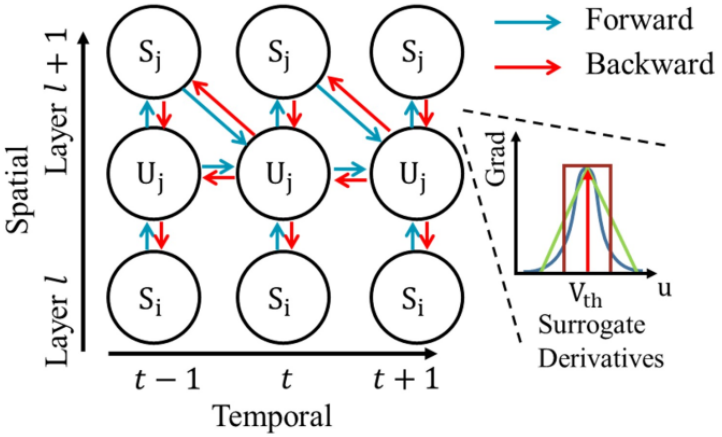
Surrogate Gradients: Researchers have applied error backpropagation with surrogate gradients to tackle the non-differentiability challenge, similar to deep learning. However, this approach introduces inaccuracies and compatibility challenges with neuromorphic devices, limiting its implementation.



(a)



(b)



(c)

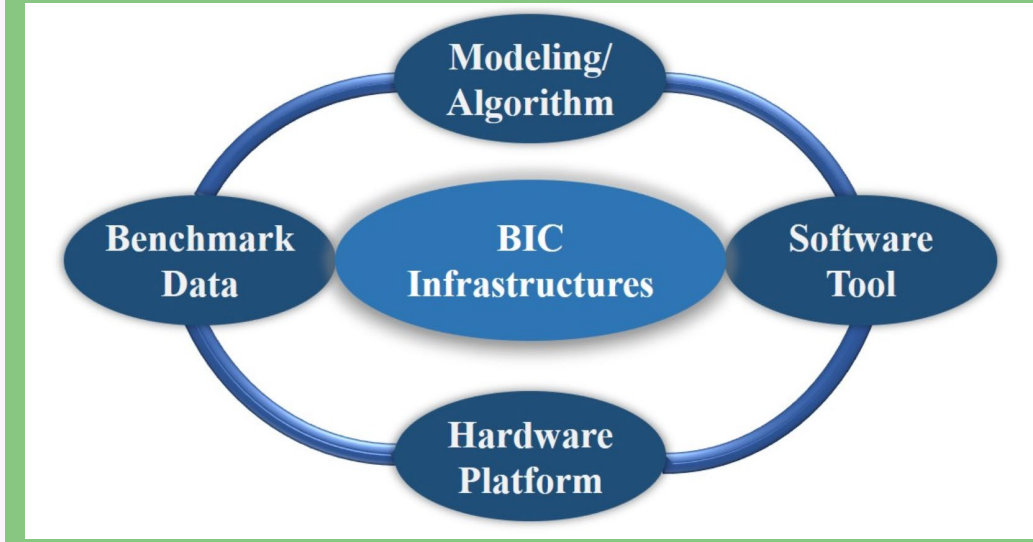
02 RELATED WORK – Neuron Evolutions

Natural Evolution Strategies (NES) and Evolution Strategies (ES): These are leading evolutionary algorithms used for training deep neural networks. ES optimizes network weights through Gaussian perturbations, and it's been effective for tasks involving sequential decision-making. Several other algorithms have also emerged, such as genetic algorithms for weight mutations.

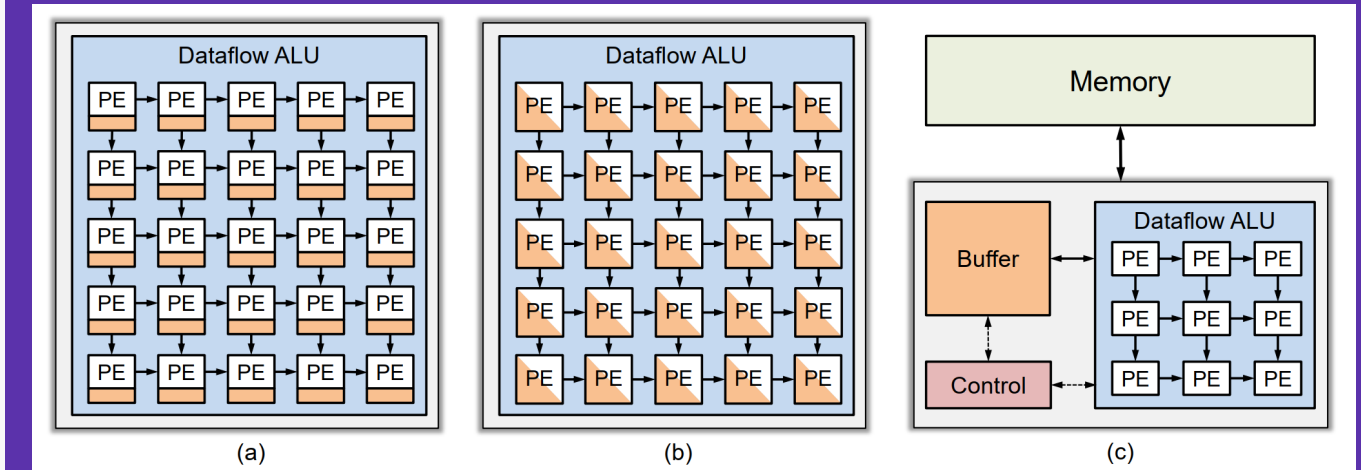
Exploration and Novelty-Seeking: Some algorithms, like ES, incorporate exploration and novelty-seeking to overcome local minima during training.

Hebbian Learning and Neuron Evolution: Hebbian learning, a biological learning rule, can be integrated into the framework of neuron evolution. In neuroevolution, this combination fine-tunes synaptic weights after evolving network architecture. This approach aligns more closely with biological principles and can lead to more efficient networks.

02 RELATED WORK – Neuromorphic Architecture



Four components of brain inspired computing (BIC) infrastructures

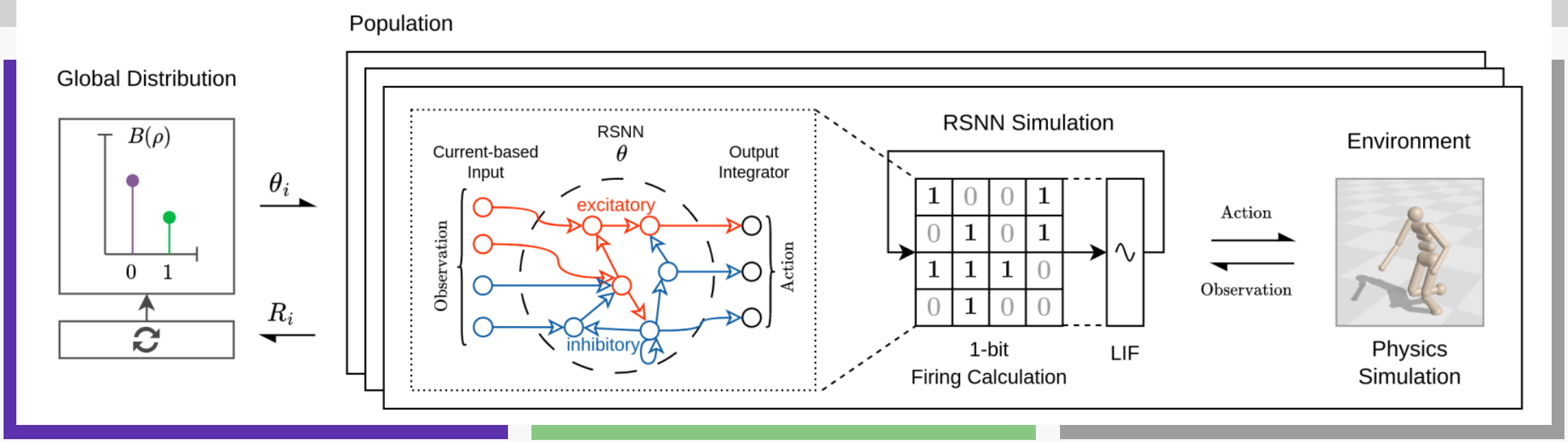


Brain inspired computing hardware architecture comparison: (a) distributed manycore design with computation close to memory (near-memory); (b) distributed manycore design with computation within memory (in-memory); (c) version specialized for ANN acceleration.

03 Research Proposal

03 RESEARCH PROPOSAL – SNN Evolving Connectivity

EC Framework for RSNNs Training: RSNNs, inspired by the human nervous system, show promise for complex tasks. Traditional training methods using surrogate gradients face accuracy issues and aren't compatible with neuromorphic hardware. Our lab introduces the Evolving Connectivity (EC) framework, which offers an inference-only approach. Instead of adjusting weights directly, it explores parameterized connection probability distributions using Natural Evolution Strategies (NES). The EC framework eliminates the need for gradients, providing hardware-friendly characteristics like sparse Boolean connections and scalability.



03 RESEARCH PROPOSAL – SNN Evolving Connectivity

LIF neuron

Potential

$$\tau_m \frac{\mathrm{d}\mathbf{u}^{(g)}}{\mathrm{d}t} = -\mathbf{u}^{(g)} + R\mathbf{c}^{(g)}$$

$$g = \{Exc, Inh\}$$

Current

$$\frac{\mathrm{d}\mathbf{c}^{(g)}}{\mathrm{d}t} = -\frac{\mathbf{c}^{(g)}}{\tau_{syn}} + \sum_{g_j} I_{g_j} \sum_j \mathbf{W}_{ij}^{(g_i g_j)} \delta(t - t_j^{s(g_j)}) + \mathbf{I}_{ext}$$

The external current is generated linearly by the observation.

$$\mathbf{c}^{(t,g)} = d_c \mathbf{c}^{(t-1,g)} + \sum_{g_j} I_{g_j} \mathbf{W}^{(g_i g_j)} \mathbf{s}^{(t-1,g_j)} + \mathbf{I}_{ext}^{(t,g)}$$

$$\mathbf{v}^{(t,g)} = d_v \mathbf{u}^{(t-1,g)} + R\mathbf{c}^{(t,g)}$$

$$\mathbf{s}^{(t,g)} = \mathbf{v}^{(t,g)} > 1 \qquad \textbf{Fire}$$

$$\mathbf{u}^{(t,g)} = \mathbf{v}^{(t,g)}(1 - \mathbf{s}^{(t,g)}) \qquad \textbf{Reset}$$

Output

$$\mathbf{o}^{(t)} = \sum_{\tau} k(\tau) \sum_{g} \mathbf{W}_{out}^{(g)} \mathbf{s}^{(t-\tau,g)}$$

03 RESEARCH PROPOSAL – SNN Evolving Connectivity

Population: generated by sampling from a Bernoulli distribution.

$$\mathbf{W}_{ij} = \mathbf{w}_{ij} \cdot \boldsymbol{\theta}_{ij}, \text{where } \boldsymbol{\theta}_{ij} \sim B(\rho)$$

If weight is unit size:

$$\mathbf{W}_{ij} = \boldsymbol{\theta}_{ij}, \text{where } \boldsymbol{\theta}_{ij} \sim B(\boldsymbol{\rho}_{ij}).$$

Optimization: maximizing the expected performance metric function $R(\cdot)$ across individual network samples. $R(\cdot)$ is from the environment.

$$\boldsymbol{\rho}^* = \text{argmax}_{\boldsymbol{\rho}} J(\boldsymbol{\rho}) = \text{argmax}_{\boldsymbol{\rho}} \mathrm{E}_{\boldsymbol{\theta} \sim B(\boldsymbol{\rho})}[R(\boldsymbol{\theta})]$$

Estimate the gradient of $J(\boldsymbol{\rho})$ to update the connectivity:

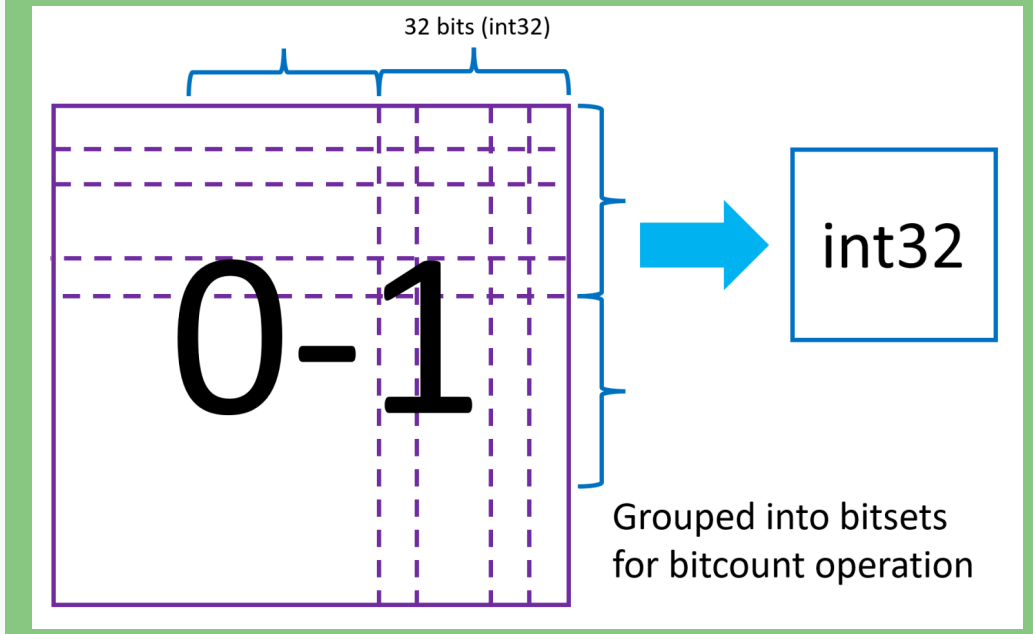
$$\begin{aligned} \nabla_{\boldsymbol{\rho}} J(\boldsymbol{\rho}) &= \mathrm{E}_{\boldsymbol{\theta} \sim B(\boldsymbol{\rho})}[\nabla_{\boldsymbol{\rho}} \log P(\boldsymbol{\theta}|\boldsymbol{\rho}) R(\boldsymbol{\theta})] \\ &= \mathrm{E}_{\boldsymbol{\theta} \sim B(\boldsymbol{\rho})}[\frac{\boldsymbol{\theta} - \boldsymbol{\rho}}{\boldsymbol{\rho}(1-\boldsymbol{\rho})} R(\boldsymbol{\theta})] \\ &\approx \frac{1}{N} \sum_{k=1}^{N} \frac{\boldsymbol{\theta}_k - \boldsymbol{\rho}}{\boldsymbol{\rho}(1-\boldsymbol{\rho})} R_k \end{aligned}$$

$$\alpha = \eta \cdot \mathrm{Var}[B(\boldsymbol{\rho})]$$

The step of learning can be tried with different expressions in the simulation to find the best one.

03 RESEARCH PROPOSAL – Learning Acceleration

Binary Operation Acceleration



Schematic diagram of transforming a binary matrix into a int32 matrix

Excitatory and Inhibitory Neuron Weights Acceleration

$$(\mathbf{W}^{+}-\mathbf{W}^{-})x = (2\mathbf{W}^{+}-\mathbf{1})x = 2\mathbf{W}^{+}x-\mathbf{1}x$$

W+ and W- are logically opposite. Then we can avoid -1 in a matrix and continue to use bitcount.

The multiplication between a binary matrix and a binary vector is an “and” operation plus a “sum” operation, which is equivalent to a “bitcount” operation for an int32 number. A “bitcount” operation is very fast, which counts the number of 1 for an int32 number in binary memory.

03 RESEARCH PROPOSAL – Performance Optimization

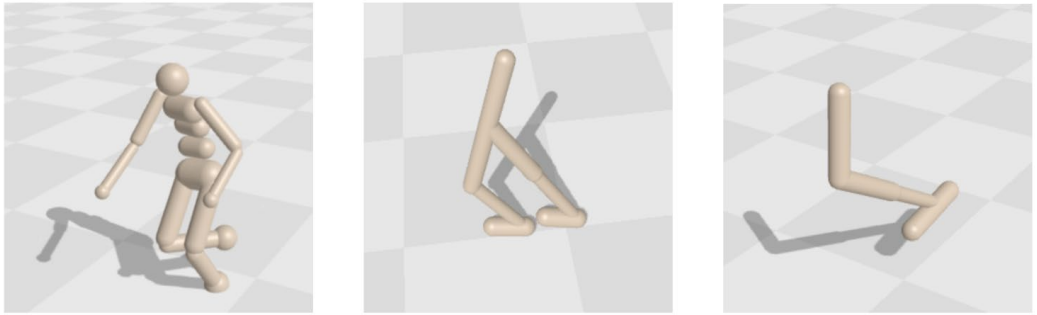
Key Advantages of the EC Framework:

1.Hardware Compatibility: The EC framework addresses the challenge of hardware-friendly learning algorithms, enabling training on neuromorphic chips without direct error backpropagation.

2.Scalability: Its inference-only design supports parallel evaluations across independent workers, improving scalability. **Efficient communication methods reduce overhead.**

3.1-Bit Sparse Connections: The EC framework employs 1-bit sparse connections for faster, cost-effective computations and potential advancements in 1-bit connection neuromorphic hardware.

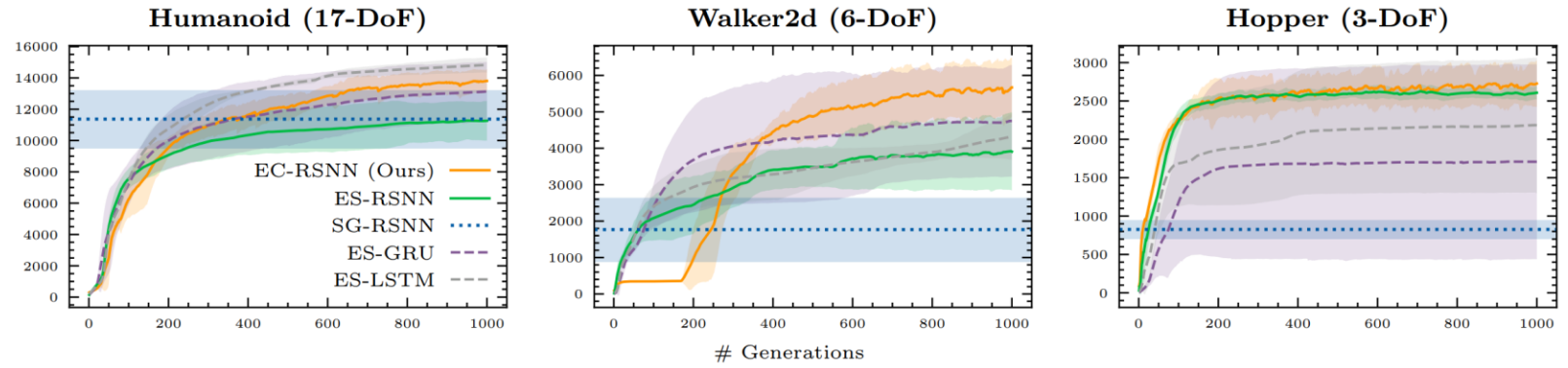
Task



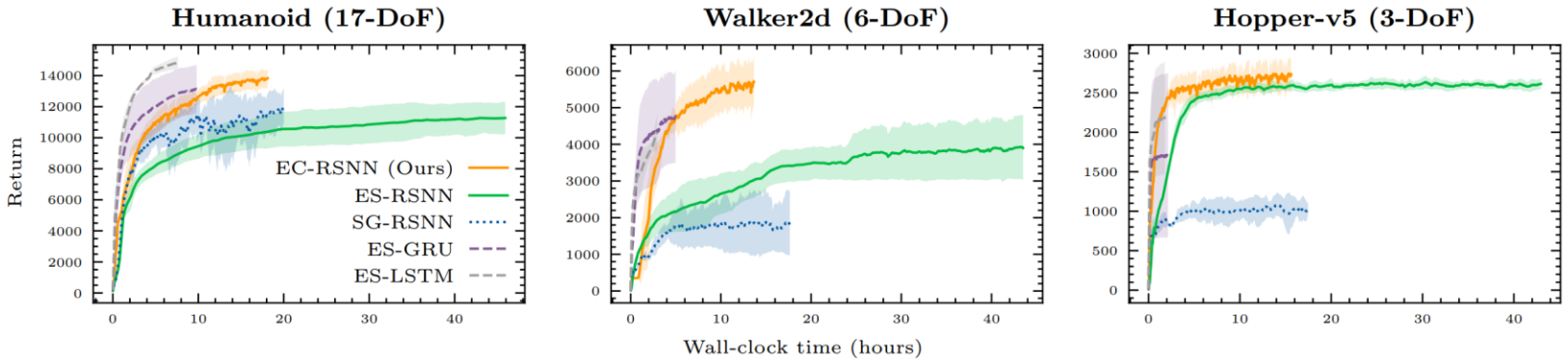
Locomotion tasks–Humanoid (17-DoF), Walker2d (6-DoF), and Hopper (3-DoF)

Our baselines are deep RNNs, RSNN trained with Surrogate Gradients (SG) and ES, and ES trained GRU and LSTM.

03 RESEARCH PROPOSAL – Performance Optimization



Comparison of performance of different models



Comparison of speed of different models

Basically, for performance, we can see that EC-RSNN outperforms ES-RSNN and is comparable with the ANN models; for speed, EC-RSNN outperforms ES-RSNN but is slower than LSTM and GRU because RSNN is highly complex and GPUS have been optimized for LSTM and GRU. Also, in these results, int32 acceleration is not used yet.

03 RESEARCH PROPOSAL – Discussion and Application to More Tasks and Networks

Key Impacts of the EC Framework (from Harware and NeuroSci):

Efficient On-Chip Learning: The EC framework overcomes the challenge of on-chip learning, offering an effective gradient-free solution for tasks like locomotion.

Versatility: It supports large-scale cloud learning and energy-efficient edge applications, making it versatile for diverse neuromorphic tasks.

Cost Reduction: The transition from floating-point to 1-bit connections in the EC framework may reduce manufacturing and energy costs.

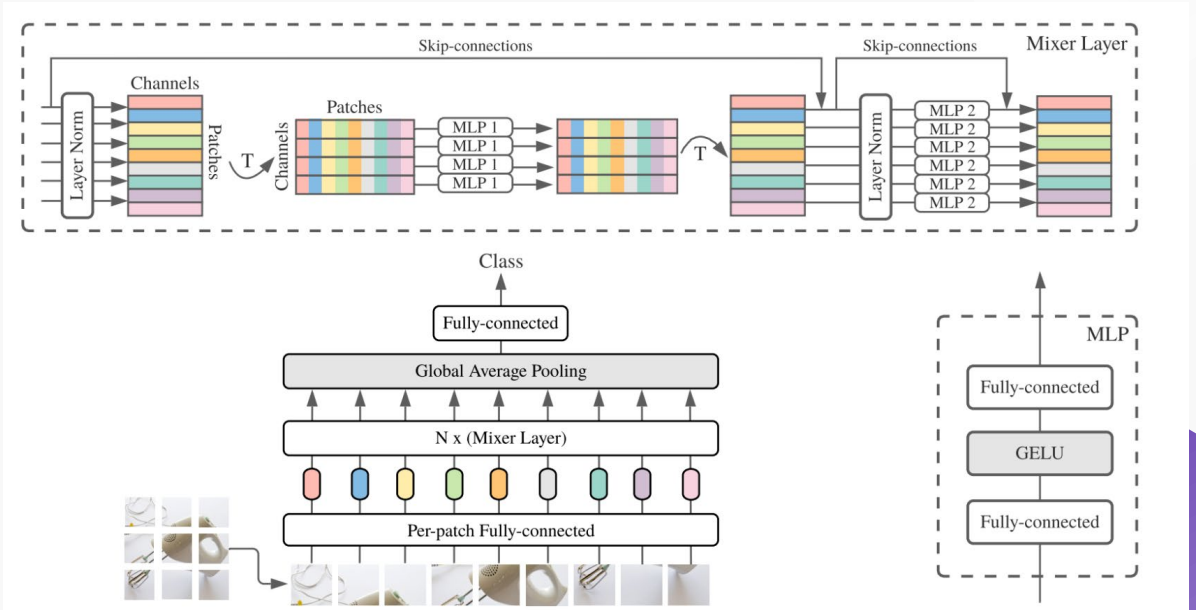
Advancing Neuroscience: It introduces real-world-like tasks for studying decision-making and motion control, providing valuable neuron-to-neuron connection data for exploring brainwide connectomes. It can incorporate neuroanatomical and neurophysiological data for data-driven neurosimulation models.

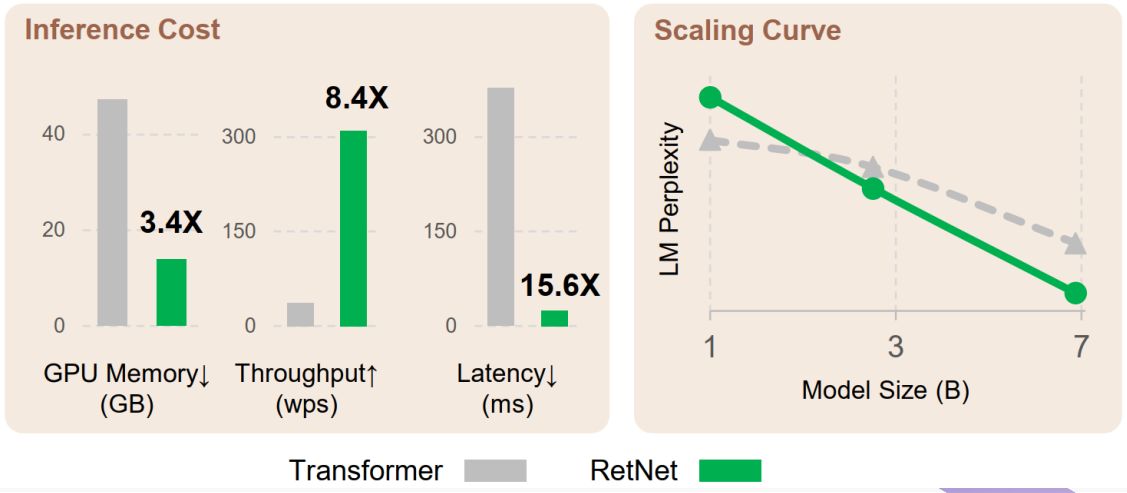
03 RESEARCH PROPOSAL – Discussion and Application to More Tasks and Networks

The next plan is to optimize the EC framework and extend it to **more tasks and network configurations**.

A new task -- image recognition: use the MLP-mixer–which is a vision transformer achieved by MLPs so that it is easier for implementation in RSNNs.

A new network configuration -- RetNet– which is a successor to transformer for LLMs with lower complexity compared to traditional transformers





04 Work Plan

04 WORK PLAN

Optimize and accelerate the framework.

Extend the framework to more new tasks.

Extend the framework to more new network configurations.

Achieve a unified EC methodology for SNN learning.

Thank You!

Q & A

- Presenter: Huimiao Chen