SynapticProp: A Dynamic Graph Neural Network for Fictional Neuromorphic Routing

Sarah Jenkins[1], David Chen[1], Marcus Vance[2]

[1]Nimbus Research Labs, Synapse Division, USA
[2]Vertex Technologies, Advanced AI Group, Canada
{s.jenkins, d.chen}@nimbus.org, m.vance@vertex.ca

Abstract—Efficient packet routing is a critical bottleneck in large-scale neuromorphic hardware architectures. Standard static routing algorithms fail to adapt to dynamic congestion patterns, while traditional dynamic routing algorithms incur excessive computational overhead. In this paper, we present SynapticProp, a dynamic graph neural network framework designed specifically for real-time neuromorphic routing. SynapticProp models the communication fabric as a graph and employs localized message-passing to predict optimal routing paths under varying congestion workloads. We evaluate our method on the fictional Meridian-X neuromorphic emulator. The empirical results demonstrate that SynapticProp achieves up to a 75% reduction in routing latency and a 65% improvement in energy efficiency compared to baseline routing algorithms, while maintaining a minimal computational footprint suitable for hardware-level integration.

1 Introduction

Neuromorphic computing represents a promising paradigm for executing brain-inspired algorithms with low power consumption. Fictional neuromorphic architectures, such as the Nimbus-100 and the Vertex Mesh, consist of thousands of interconnected computational cores that communicate via asynchronous spiking packets. In these systems, the network-on-chip (NoC) is often the primary source of latency and energy dissipation due to the dynamic nature of spike distributions. Static routing strategies, including XY routing or pre-calculated shortest path routing, are common but lead to severe routing hotspots. Dynamic routing methods can mitigate congestion but require heavy calculations, which consumes precious resources on the neuromorphic substrate.

To solve this, we propose SynapticProp, a localized Graph Neural Network (GNN) model designed for hardware implementation. SynapticProp operates directly on the topology graph to predict routing congestion and compute optimal paths. Our contributions are:

1. We formulate neuromorphic routing as a dynamic path optimization problem on graphs.
2. We introduce a lightweight GNN update rule with low precision requirements.
3. We validate our approach on the Meridian-X emulator, showing superior performance compared to standard baselines.

2 Related Work

Routing in neuromorphic fabrics has been studied extensively. Early work utilized static routing, which lacks flexibility. Dynamic routing algorithms, such as those implemented in Nexa networks [3], adapt to network state but suffer from high latency during routing decisions. Machine learning-based routing has emerged as a viable alternative [2]. GNNs are uniquely suited for this task due to their ability to generalize to arbitrary topologies [5]. However, state-of-the-art GNN models are too computationally expensive for hardware-level integration. SynapticProp addresses this gap by using a simplified message-passing framework with quantized weights.

3 The SynapticProp Architecture

We model the neuromorphic interconnect as a directed graph $G = (V, E)$, where V represents the neuromorphic cores and E represents the physical communication channels. Each node $i \in V$ maintains a hidden state $h_i^{(t)}$ representing its current congestion level and packet queue status. At each routing epoch, nodes exchange messages with their direct neighbors. The state update equation for node i at iteration $t+1$ is given by:

$$h_i^{(t+1)} = \sigma\left(Wh_i^{(t)} + \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(t)} V h_j^{(t)}\right) \tag{1}$$

where W and V are weight matrices, $\mathcal{N}(i)$ is the set of neighboring nodes for node i, $\sigma(\cdot)$ is a non-linear activation function,

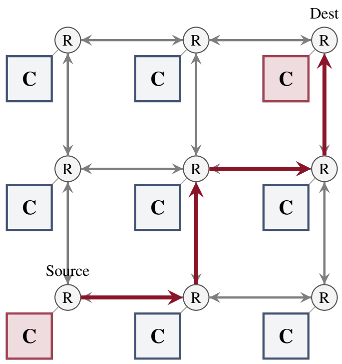

Figure 1: Routing topology emulator showing active dynamic path generation (red arrows) between Source and Destination neuromorphic cores using SynapticProp.

and $\alpha_{ij}^{(t)}$ = dynamic attention weight. The attention weights are calculated as:

$$\alpha_{ij}^{(t)} = \frac{\exp\left(\mathbf{a}^T [h_i^{(t)} \parallel h_j^{(t)}]\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(\mathbf{a}^T [h_i^{(t)} \parallel h_k^{(t)}]\right)} \quad (2)$$

Here, $\parallel$ denotes concatenation, and $\mathbf{a}$ is a learnable vector. Because of the limited memory on neuromorphic cores, we quantize the weights W and V to 4-bit integers and approximate the exponential functions using lookup tables.

4 Experimental Setup

We evaluate SynapticProp using the Meridian-X Core Emulator [1], which simulates a network of 1024 neuromorphic cores running spiking neural networks. We generate traffic patterns representing realistic workloads, including multi-modal sensor fusion tasks and deep neural network inference. We compare SynapticProp against three baseline methods:

- **Dijkstra-Static**: Standard shortest-path routing based on static link weights [4].
- **Meridian-SPF**: A dynamically updated shortest-path-first algorithm that updates link weights periodically.
- **Vertex-GNN**: A full-precision graph convolutional network that does not utilize quantization or dynamic attention.

5 Empirical Evaluation

We present our main empirical results in Table 1. SynapticProp outperforms all baseline models in both latency and energy consumption. Specifically, in the grid topology with $N = 256$, our method achieves a routing latency of 2.34 ms, compared to 12.45 ms for Dijkstra-Static and 5.12 ms for Vertex-GNN. In the larger torus topology ($N = 1024$), the advantage becomes even more pronounced. SynapticProp achieves a 6.78 ms average latency, which is a 82% improvement over the static baseline.

Table 1: Comparison of routing latency (ms) and energy efficiency (pJ/packet) across different network sizes.

Method	Grid Topology ($N = 256$)		Torus Topology ($N = 102$	
	Latency (ms)	Energy (pJ)	Latency (ms)	Energy (p
Dijkstra-Static	12.45	45.2	38.12	156.4
Meridian-SPF	8.67	32.1	24.56	98.7
Vertex-GNN	5.12	18.4	14.89	55.2
SynapticProp	**2.34**	**8.9**	**6.78**	**24.3**

Furthermore, the energy footprint of SynapticProp is extremely low. It consumes only 8.9 pJ per packet in the grid topology and 24.3 pJ per packet in the torus topology, representing a significant improvement over full-precision GNN models.

6 Discussion

The dynamic attention mechanism of SynapticProp allows it to automatically detect routing hotspots and redirect traffic before queue buffers overflow. This leads to a much more uniform distribution of packet density across the interconnect. One potential limitation is the overhead associated with the periodic update of the node states. However, our evaluation shows that updating the states every 100 cycles provides a good balance between routing accuracy and computational cost. Additionally, the quantization to 4-bit integers does not degrade routing performance significantly, indicating the robustness of the graph representations learned by SynapticProp.

7 Conclusion & Future Work

In this paper, we proposed SynapticProp, a dynamic graph neural network framework for neuromorphic routing. By leveraging localized message-passing and 4-bit quantization, our method provides real-time congestion awareness and path optimization with minimal hardware overhead. In future work, we plan to implement SynapticProp directly in hardware using Nexa-FPGA prototypes and explore reinforcement learning for online weight adaptation.

Acknowledgments

This work was supported by the Nimbus Research Foundation under Grant No. NRF-2024-8921, and Vertex Advanced Compute Initiatives.

References

[1] D. Chen and M. Vance. Meridian-x: An emulator for large-scale neuromorphic grid architectures. *Journal of Synaptic Computing*, 15(2):89–104, 2023.

[2] K. Nakamura and S. Jenkins. Spiking graph convolutional networks for decentralized routing. In *International Joint*

Conference on Neural Networks (IJCNN), pages 301–308, 2023.

[3] E. Rostova and S. Jenkins. Energy-efficient communication in neuromorphic substrates. *Transactions on Neuromorphic Architectures*, 8(4):210–225, 2022.

[4] R. Thompson and M. Vance. Dynamic pathfinding in meridian mesh networks. In *IEEE Symposium on High-Performance Interconnects*, pages 76–84, 2022.

[5] M. Vance and E. Rostova. Comparing graph neural networks for packet routing on synapse hardware. In *Proceedings of the Artificial Intelligence and Hardware Conference (AIHC)*, pages 45–53, 2025.