Towards adaptive spiking label propagation[*]

Kathleen E. Hamilton
Oak Ridge National Library
Oak Ridge, Tennessee
hamiltonke@ornl.gov

Catherine D. Schuman
Oak Ridge National Library
Oak Ridge, Tennessee
schumancd@ornl.gov

ABSTRACT

Graph algorithms are a new class of applications for neuromorphic hardware. Rather than adapting deep learning and standard neural network approaches to a low-precision spiking environment, we use spiking neurons to analyze undirected graphs (e.g., the underlying modular structure). While fully connected spin glass implementations of spiking label propagation have shown promising results on graphs with dense communities, identifying sparse communities remains difficult. This work focuses on steps towards an adaptive spike-based implementations of label propagation, utilizing sparse embeddings and synaptic plasticity. Sparser embeddings reduce the number of inhibitory connections, and synaptic plasticity is used to simultaneously amplify spike responses between neurons in the same community, while impeding spike responses across different communities. We present results on identifying communities in sparse graphs with very small communities.

CCS CONCEPTS

• **Mathematics of computing → Graph algorithms**; Random graphs; • **Hardware → Emerging tools and methodologies**; **Neural systems**;

KEYWORDS

neuromorphic, community detection, graph algorithm, path finding, spiking neural networks

ACM Reference Format:
Kathleen E. Hamilton and Catherine D. Schuman. 2018. Towards adaptive spiking label propagation. In *Proceedings of International Conference on Neuromorphic Systems (ICONS 2018).* ACM, New York, NY, USA, Article X, 8 pages. https://doi.org/10.475/123_4

[*]This work was supported by the United States Department of Defense and used resources of the Computational Research and Development Programs at Oak Ridge National Laboratory. This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC0500OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for the United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan.

1 INTRODUCTION

Neuromorphic computing is commonly referred to as "brain-inspired" computing, utilizing hardware that executes processes similar to how the mammalian brain executes cognitive function. These systems have predominantly been used to accelerate deep learning algorithms [5, 14] but there has been growing interest in adapting non-deep learning algorithms and applications to these systems. For example, a spike-based implementation of simulated annealing has been used to solve various constraint satisfaction problems on the SpiNNaker system [3, 12].

Neuromorphic hardware is developed to mimic the biological processes of the mammalian brain, and one important element is synaptic plasticity. Recent hardware developments have focused on building neuromorphic systems that can update and change the weights associated with individual synapses (e.g., Intel's Loihi chip [2]). In this work we present an adaptation of spiking label propagation [7, 8], which was previously implemented with static synapses and fully connected spiking neural networks (SNNs). Our adapted algorithm is deployed on larger, sparser networks and utilizes plastic synapses and a sparse embedding of a network into a spiking neuron system.

The previous studies of spiking label propagation [7, 8, 15, 16] have demonstrated that communities can be identified on graphs from spiking output. In [15, 16] the use of global driving and global control lead to synchronization over large populations of neurons, which was used to infer graph communities. In [7, 8], local driving and localized spike responses were used to identify communities, based on similarities in the local spike responses.

The label propagation method introduced in [7] was inspired by spin glass physics, where local interactions can lead to localized patterns without influencing global behavior. Similar approaches for spin-glass inspired community detection, implemented with interacting spins [10, 17], have been shown to avoid the resolution limit which is common in other modality-based measures [1]. However, when implemented with spiking neurons and homogeneous, static synapses, it was seen in [7] that the method has difficulty identifying small communities in sparse graphs.

Spiking label propagation is dependent on localized spiking behavior. Driving a single neuron must only cause a bounded number of neurons in close proximity to fire spikes in response. Additionally, the spike responses of neurons that are in the same community should overlap significantly. In previous studies this locality constraint was enforced only by the addition of inhibitory synapses [7], but is also dependent on the underlying graph connectivity [9]. Very sparse graphs can have localized spike responses without the addition of inhibitory synapses. While this indicates that sparser embeddings are possible, there still remains the challenge of how to generate spike responses with sufficient overlap.

We modify spiking label propagation to address two challenges: generating a local spike response that can sufficiently cover a sparse community, and efficiently embedding a graph in a spiking neuron system with minimal synaptic connections. We use synaptic connections that are now able to implement plasticity-based learning rules, and the embedding of a graph into a SNN is made sparser by reducing the number of inhibitory synapses. The goal of these modifications is to allow for the localized spike response to grow larger as more neurons in a community are driven without causing spiking cascades that spread throughout the network.

These modifications are designed to overcome challenges faced when analyzing real world networks. Real world networks can contain millions, if not billions of vertices. Embedding them into a fully connected spin glass will lead to prohibitively large systems. In many real-world networks the communities may contain a large number of vertexes, yet have relatively low mean degree. The relative sparseness of these communities means that it cannot be assumed that the local response from one neuron can significantly overlap with a large fraction of the remainder of the community.

2 DEFINITIONS

We construct our spiking neuron systems out of leaky integrate and fire neurons and only apply our approach to identifying groups of vertices in simple graphs: no self-loops, multiple edges, or weighted edges. For spiking label propagation, driving a single neuron needs to generate a localized response and the network must be designed to inhibit the large-scale growth of a spike response.

Graphs $\mathcal{G} = \mathcal{G}(V, E)$ are fully defined by a vertex set $V(\mathcal{G}) = \{v_i\}$ and a set of symmetric edges ($E(\mathcal{G}) = \{e_{ij}\}$, $e_{ij} = e_{ji} = (v_i, v_j)$). Graph sparsity is quantified by the ratio of edges in $\mathcal{G}$, to the maximum number of edges possible on $|V(\mathcal{G})|$ vertices: $\sigma = 2|E(\mathcal{G})|/(|V(\mathcal{G})|(|V(\mathcal{G})|-1))$. A SNN is defined similarly as an edge-weighted network $\mathcal{S} = \mathcal{S}(N, W, s_W)$ using a neuron set ($N = \{n\}$), a set of symmetric connections ($W = \{w_{ij}\}$, $w_{ij} = w_{ji} = (n_i, n_j)$) called *synapses*. Associated with each synapse is an edge weight (s_w) which may be positive or negative, and self-connections ($w_{ii} = (n_i, n_i) \notin W$) are not allowed. For all embeddings, each graph vertex is mapped to a leaky integrate and fire neuron $n_i \in N(\mathcal{S}) \iff v_i \in V(\mathcal{G})$, and each graph edge is mapped to an excitatory synapse $w_{ij} \in W(\mathcal{S}), s_w > 0 \iff e_{ij} \in E(\mathcal{G})$. The initial value of the inhibitory synapses can vary. For studies with static synapses, we choose the inhibitory weight to be equal and opposite the excitatory weight.

The dynamics of the leaky integrate and fire neurons are controlled by the dynamical system of equations:

$$\frac{dV_j}{dt} = \frac{\left(V_{ext}(t) - V_j(t)\right)}{\tau}, \tag{1}$$

$$V_{ext}(t) = I_{ext}(t)R + \sum_{\substack{i \to j,\\ i \neq j}} w_{ij}\delta(t - t^{(f)}). \tag{2}$$

In this work, the external current I_{ext} used to generate spiking dynamics is replaced by a discrete set of n input spikes, spaced at uniform intervals t_1, and sent to a single neuron. Each of these input spikes arrives with a synaptic weight that exceeds the firing threshold of the neuron being driven, ensuring that each input spike fed to a neuron causes that neuron to fire. We define the

Table 1: Benchmark graphs.

Graph	$\lvert V(\mathcal{G})\rvert$	$\lvert E(\mathcal{G})\rvert$	$\lvert\{Q_i\}\rvert$	D	σ
$\mathcal{G}_0$	128	256	16	4	0.032
$\mathcal{G}_1$	128	1024	4	16	0.126
$\mathcal{G}_2$	256	1014	16	7.9	0.031

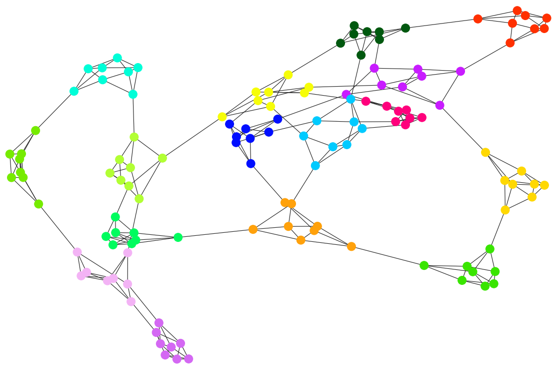

Figure 1: The benchmark graph on 128 **vertices with** 16 **unique communities. Many communities can be disconnected from the graph by removing** 2 **edges, but to disconnect a single vertex from a community requires the removal of** > 2 **edges.**

time during which input spikes are sent to an individual neuron as Δ_{EXT}, and the time between individual neuron driving is sufficient to ensure that any incurred charge on a neuron has dissipated. The spiking dynamics are simulated with the following neuron parameters: $v_{th} = 0.8$ V, $v_0 = v_R = 0.0$ V, $\tau = 25$ ms, $t_R = 20$ ms, and synaptic weight amplitude $|s_w|_0 = 0.75$V. Driving is done by sending 10 spikes to a neuron, spaced at 0.21 ms intervals.

We define three different benchmark graphs to test various embeddings and investigate the effects of synaptic plasticity (see Table 1). The graphs were generated using the software available at [4], and converted to spiking neural systems. The graph $\mathcal{G}_0$ (shown in Fig. 1) is the sparsest network analyzed in [7], the graph $\mathcal{G}_1$ is the standard Girvan-Newman benchmark graph, and $\mathcal{G}_2$ is a large graph with a sparsity similar to $\mathcal{G}_0$, but twice as many vertices, and with a mean degree $D(\mathcal{G}_2) \approx 2D(\mathcal{G}_0)$.

Our custom spiking neural network simulation software framework is written in C++ and utilizes a discrete-event simulation approach for rapid network evaluation. The simulator framework can be run in either serialized or parallel mode and can scale to utilize a distributed memory supercomputer or cluster in order to simulate the activity in larger networks (>1000 neurons with dense connectivity) more rapidly than on a single processor [18]. The software framework is modular and allows for different neuron models and synapse models to be easily implemented and evaluated using the simulation engine.

Once the full spike raster was generated, the label propagation algorithm of [7] was applied. Each spike train was decoded into a binary vector x_i using a time bin width of $\Delta t = 0.03$ sec, and the Hamming metric $H(x_i, x_j)$ defined in [11] was used to quantify the degree of similarity between pairs of decoded spike trains. A threshold value h_0 for the Hamming similarity was chosen and fixed, and each neuron of $\mathcal{S}$ was initialized with a unique label. Labels propagate through the system based on the degree of similarity between pairs of neuron spike trains; if $H(x_i, x_j) \geq h_0$ then the label of x_i is applied to x_j. For a driven neuron (n_0), the size of the spike response was calculated by counting the number of unique neurons that fired while (n_0) was being actively driven.

With each benchmark graph in Table 1, we also have a set of known labels that act as the ground truth. This known distribution of labels over the original vertex set (L_0) is compared to the label propagation output, which returns a predicted label distribution (L'). The difference between two graph partitionings is quantified using the variance of information $\mathrm{VI}(L_0, L')$ [13], which vanishes when two partitionings are identical: $\mathrm{VI}(L_0, L') = 0$. The variance of information is maximum when every vertex on the graph is assigned a unique label. We are most interesting in finding for which threshold similarity does the minimum variance of information (MVI) vanish. For reference we include the values of VI for the two trivial solutions: all vertices classified in the same community:

$$\begin{aligned} \mathrm{VI}(L_0, \{1\})_{(\mathcal{G}_0)} &= 4.0, \\ \mathrm{VI}(L_0, \{1\})_{(\mathcal{G}_1)} &= 2.0, \\ \mathrm{VI}(L_0, \{1\})_{(\mathcal{G}_2)} &= 6.0; \end{aligned} \tag{3}$$

and all vertices classified with unique labels:

$$\begin{aligned} \mathrm{VI}(L_0, \{1, 128\})_{(\mathcal{G}_0)} &= 3.0, \\ \mathrm{VI}(L_0, \{1, 128\})_{(\mathcal{G}_1)} &= 5.0, \\ \mathrm{VI}((L_0, \{1, 256\})_{(\mathcal{G}_2)} &= 4.0. \end{aligned} \tag{4}$$

3 SNN CONSTRUCTION

In the following sections we investigate how the size of spiking responses in an SNN and the ability to recover a known label distribution, is affected by: synapse plasticity and the sparsity of the SNN.

3.1 Sparse embedding of local spin glass networks

Fully connected spin glass models have been used to demonstrate proof-of-concept results that spiking neuron systems can be used to implement label propagation. However fully connected systems will be dominated by inhibitory synaptic connections for most real-world networks, which tend to be sparse. The embedding discussed in this section is used for spiking label propagation, where the generation of a localized response from a driving a single neuron is needed. For the sparse graphs we consider in this work, embedding into a fully connected spin glass will result in a graph dominated by inhibitory synapses. This leads to very sparse spike rasters and as seen in Fig. 2 loss of distinction between small communities. We now define a spiking neuron systems which has excitatory synapses for any edge on the original graph, but inhibitory synapses are only added between small localized regions on the graph, identified from the graph adjacency matrix.

The design of our spiking neuron systems for label propagation require that two spikes must arrive from a driven neuron in order to cause a neuron to fire in response. A neuron can fire when it is directly connected to a driven neuron, or if there are a number of short paths (length 2 paths) connecting to the driven neuron. Inhibitory connections are only added if there are a_0 length 2 paths between two neurons, and neurons that are far apart on the graph are left disconnected. On a sparse graph, there may be few neurons connected by multiple length-2 paths, as a result the choice of a_0 can lead to a SNN with very few inhibitory connections. For the full spin glass embedding, an inhibitory synapse is added to the SNN for every edge that is absent on the original graph, resulting in a SNN which is fully connected. To design sparser embeddings, we choose to only attach inhibitory synapses between neurons that have a certain number of length-2 paths between them:

$$w_{ij} < 0 \in W(\mathcal{S}) \iff (A^2)_{ij} = a_0. \tag{5}$$

In Table 2 we compare the number of inhibitory, and excitatory connections for various values of a_0. A set of sparse embeddings were generated for $\mathcal{G}_0$, and the spiking dynamics were generated using only static synapses. Reducing the number of inhibitory synapses in the SNN results in a larger spike response from driving a single neuron. This can be quantified by simply counting the number of neurons that are active when one neuron is driven (see Fig. 3). In Fig. 4 the sparse embeddings on $\mathcal{G}_0$ lead to an increased degree of dissimilarity between different communities when $a_0 = 1$ or $a_0 = 3$. Label propagation is run on the decoded spike output for each

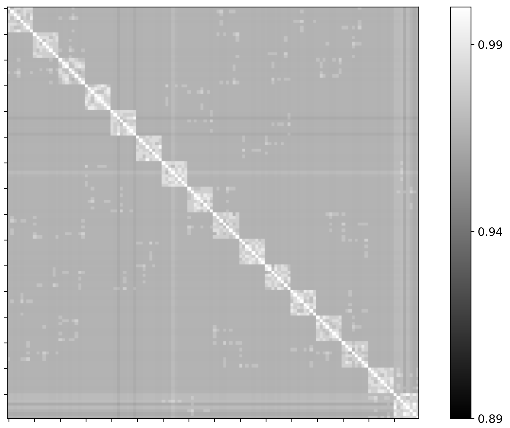

Figure 2: The comparison between binary decoded spikes trains generated under the fully connected spin glass model. Individual communities are not significantly different and similarity within individual communities is not significantly high.

Table 2: Embedding sizes for each of the benchmark graphs.

Graph	$\|W(\mathcal{S})_+\|$	$\|W(\mathcal{S})_-\|$ (full glass)	$\|W(\mathcal{S})_-\|$ $a_0 = 1$	$\|W(\mathcal{S})_-\|$ $a_0 = 2$	$\|W(\mathcal{S})_-\|$ $a_0 = 3$
$\mathcal{G}_0$	256	7872	456	238	66
$\mathcal{G}_1$	1024	7104	4022	840	116
$\mathcal{G}_2$	1014	31626	3124	536	542

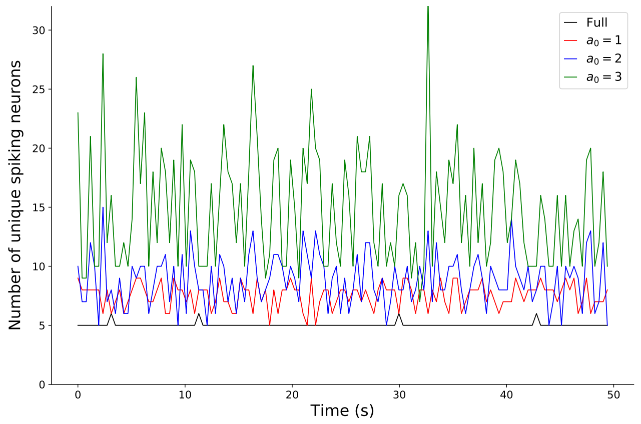

Figure 3: The number of neurons that fire when a single neuron is driven for a SNN generated from $\mathcal{G}_0$ under several embeddings. For the full spin-glass embedding, the spike response is limited to just nearest neighbors. Under sparser embeddings, the number of spiking neurons can grow to be very large.

sparse embedding shows that for $\mathcal{G}_0$ the variance of information vanishes with $a_0 = 1$ and $a_0 = 3$.

In an alternate study on $\mathcal{G}_0$, a sparse embedding was simulated using Brian2 [6] with added inhibitory synapses between neurons that satisfy $(A^2)_{ij} \leq a_0$. This led to denser embeddings ($|W(\mathcal{S})_-| = 456, a_0 \leq 1, |W(\mathcal{S})_-| = 694, a_0 \leq 2$, $|W(\mathcal{S})_-| = 760, a_0 \leq 3$) and while the variance of information was reduced, it only vanished for $a_0 = 1$ [9].

3.2 Synapse plasticity

We implement spike timing dependent plasticity using Hebbian learning. Only one neuron fires spikes due to driving by an external source, while any other neuron that fires during the active driving of a neuron will fire due to "internal firing," firing a spike because of the arrival of multiple spikes from a neighboring neurons. We define the learning rule according to the total output of the neurons n_i, n_j during the time that neuron n_i is driven by an external source $\Delta(t_i)$.

We implement plasticity and synaptic learning through a simple update rule: after the external driving of a neuron (n_i) ends, a subset of synaptic weights are updated based upon their spiking behavior during the external driving. Only synapses which terminate at n_i (have n_i as a post-synaptic neuron) will update: $\{w_{ji}\} \in W(\mathcal{S})$. We define $\eta(n_j)$ over the entire period of external driving Δ_{EXT}: $\eta(n_j) = 1$ if n_j fires a spike during Δ_{EXT} and $\eta(n_j) = 0$ if n_j does not fire at all during Δ_{EXT}.

$$s'_w = s_w - \Delta_s(-1)^{\eta(n_j)}. \tag{6}$$

If n_j does not fire, then $s'_w < s_w$. If n_j does fire, then $s'_w > s_w$. We used two sets of learning rates: in Sec. 3.2.1 the learning rate was homogeneous and chosen to be $\Delta_s = |s_w|$ for all synapses, in Sec. 3.2.2 the learning rate was dependent on the local connectivity of the underlying graph.

3.2.1 Homogeneous plastic synapses.

If n_j fires in response to n_i being driven, then the excitatory synapse s_{ji} is increased to $s_{ji} = 2|s_w|$, which exceeds the spiking threshold $2|s_w| > v_{th}$. This ensures that when n_j is driven, the strengthened synapse is able to immediately cause response spiking in n_i. While excitatory synapses are increased, inhibitory synapses are further depressed: if n_j does not fire when n_i then the inhibitory synapse is depressed to $s'_{ji} = -2|s_w|$. This is necessary to prevent runaway spike cascades from spreading through the entire system. Additionally, there are inhibitory synapses that are reduced in magnitude, but do not become excitatory: for neurons that are densely connected (e.g a large number of length-2 paths exist between n_i and n_j) but not directly connected, the fact that n_j will still spike when n_i is driven will reduce the strength of the inhibitory synapse s_{ji} and depending on the initial value of s_{ji} this synapse may be effectively removed ($s'_{ji} = 0$).

Incorporating synapse plasticity leads to a system with dynamic spiking behavior: the local spike response of a single neuron is not just determined by the local connectivity but is also dependent on whether or not a neighboring neuron has been driven recently. The benchmark graph shown in Fig. 1, has 128 vertices, 256 edges and is degree regular with all vertices having degree $k = 4$. When it is embedded into a spiking neuron system as a fully connected spin glass, this translates to a system with 256 symmetric pairs of excitatory synapses, and 7872 symmetric pairs of inhibitory synapses. This leads to extremely localized spike responses when driving individual neurons. Introducing plasticity to the synapses will lead to more varied spike responses, but it can lead to runaway spike cascades (see Fig. 5).

3.2.2 Heterogeneous plastic synapses.

The final network design we considered combined multiple sparse embedding with plastic synapses, where now the initial weight of the inhibitory synapses was dependent on the local connectivity of the original graph, and the synapses are plastic. Rather than choose a single value of a_0, we use $a_0 = 1, 2, 3$ to add inhibitory synapses of different initial weights.

$$\begin{aligned} w_{ij} \in W(\mathcal{S}),\ s_w &= -0.75\text{ V} \iff (A^2)_{ij} = 1 \\ w_{ij} \in W(\mathcal{S}),\ s_w &= -0.5\text{ V} \iff (A^2)_{ij} = 2 \\ w_{ij} \in W(\mathcal{S}),\ s_w &= -0.25\text{ V} \iff (A^2)_{ij} = 3 \end{aligned} \tag{7}$$

Thus the strongest inhibitory weights are added between neurons with the sparsest connections. For this embedding, the learning rate was reduced to $\Delta_s = 0.25$ V. This learning rate is large enough to ensure that excitatory synapses increase to a weight that surpasses the spike threshold.

The heterogeneous SNN has dynamic spiking behavior, but does not lead to cascades. In Fig. 6 we show the size of spike responses generated in graphs $\mathcal{G}_0$ and $\mathcal{G}_2$ for static synapses, homogeneous plastic synapses, and heterogeneous synapses.

The spiking output generated under the different sparse embeddings were incorporated into our label propagation analysis. In previous studies [7] we have quantified the performance of spiking label propagation in terms of: vanishing variance of information, and the existence of a "plateau" over the choice of threshold value for h_0. We summarize the results for static synapses in Table 3 and

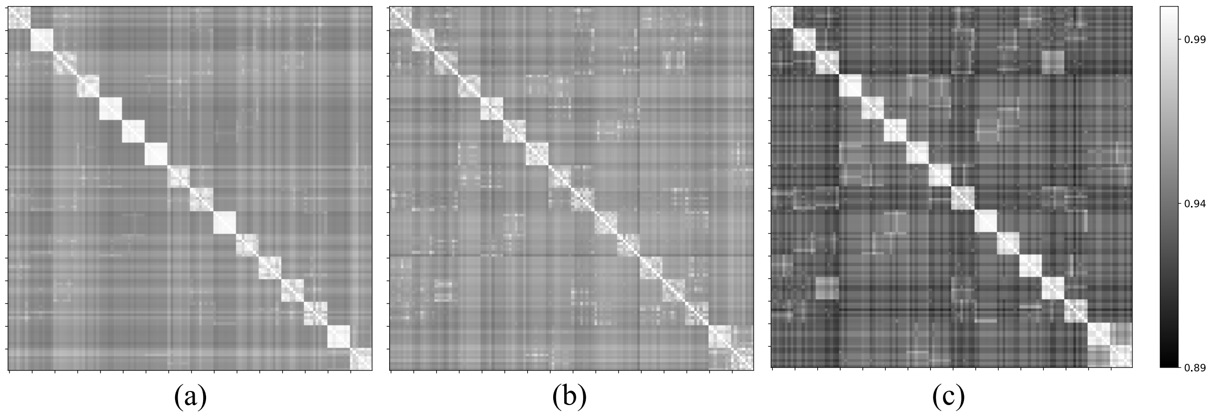

Figure 4: The degree of similarity between binary decoded spike trains for a benchmark graph with 16 communities each with 8 vertices using the adjacency defined embedding and static synapses. (a) shows the similarity between spike trains with inhibitory connections added only to pairs of disconnected neurons which have 1 length-2 paths between them ($a_0 = 1$), (b) shows the similarity between spike trains generated under the sparser embedding, with inhibitory connections added only to pairs of disconnected neurons which have 2 length-2 paths between them ($a_0 = 2$), (c) shows the similarity between spike trains generated under the sparser embedding, with inhibitory connections added only to pairs of disconnected neurons which have 3 length-2 paths between them ($a_0 = 3$). connections weighted according to Eq. 6.

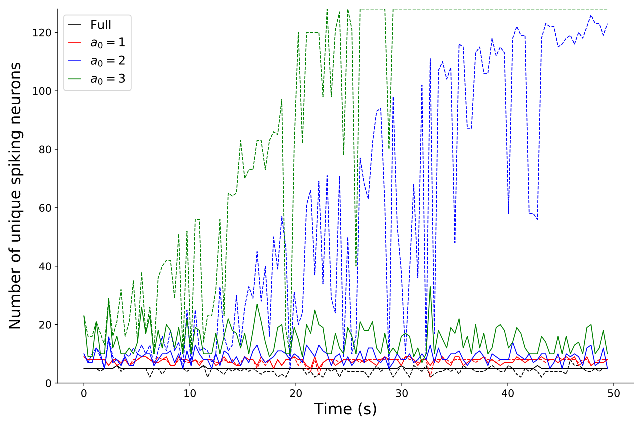

Figure 5: The number of neurons that fire when a single neuron is driven for a SNN generated from $\mathcal{G}_0$ under several embeddings and the addition of plasticity. Under sparser embeddings, the number of spiking neurons can grow to be very large and with the addition of synaptic plasticity, can lead to spike responses that cover the entire network (e.g. $a_0 = 2, a_0 = 3$ embeddings). The solid lines are generated using static synapses, the dashed lines are generated using plastic synapses.

plastic synapses in Table 4. To compare the performance of each embedding, we use the MVI and the width of the plateau if the MVI vanishes.

Table 3: Minimum information of variance for SNN with static synapses

Graph	Embed	MVI	$\{h_0\}$	Δh_0
$\mathcal{G}_0$	full	0.583	$\emptyset$	$\emptyset$
$\mathcal{G}_0$	$a_0 = 1$	0.000	$\{0.9796\}$	0
$\mathcal{G}_0$	$a_0 = 2$	1.11	$\emptyset$	$\emptyset$
$\mathcal{G}_0$	$a_0 = 3$	0.000	$\{0.9755\}$	0
$\mathcal{G}_1$	full	0.000	$\{0.9143, 0.9388\}$	0.0245
$\mathcal{G}_1$	$a_0 = 1$	0.000	$\{0.8204, 0.9837\}$	0.1633
$\mathcal{G}_1$	$a_0 = 2$	0.000	$\{0.8122, 0.9714\}$	0.1592
$\mathcal{G}_1$	$a_0 = 3$	2.000	$\emptyset$	$\emptyset$
$\mathcal{G}_2$	full	0.062	$\emptyset$	$\emptyset$
$\mathcal{G}_2$	$a_0 = 1$	0.000	$\{0.9755, 0.9837\}$	0.0082
$\mathcal{G}_2$	$a_0 = 2$	0.381	$\emptyset$	$\emptyset$
$\mathcal{G}_2$	$a_0 = 3$	0.245	$\emptyset$	$\emptyset$

4 DISCUSSION

Previous implementations of spiking label propagation used fully connected spin glass systems with static and homogeneous synaptic weights [7, 8]. While these systems could generate spiking responses that could accurately identify large, densely-connected communities, locating and isolating small, sparse communities is difficult. The results in Secs. 3.1, 3.2.1 and 3.2.2 indicate that performance on sparse graphs with small communities can be improved by implementing synapse plasticity or sparse embeddings. We compare these results to a graph on which the original iteration of label propagation performed very well, the standard Girvan-Newman benchmark ($\mathcal{G}_1$).

Our primary motivation in adding synapse plasticity and sparser embedding is to increase the size of spike responses. For all the three graphs studied, sparser embeddings lead to larger spike responses.

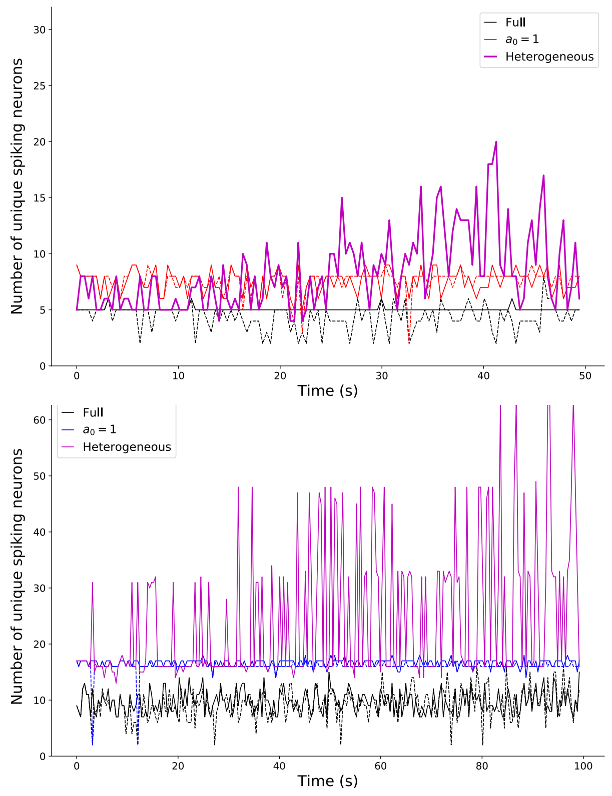

Figure 6: (Top) The number of neurons that spike during external driving for $\mathcal{G}_0$ under the full spin glass embedding with static synapses or plastic synapses (black solid, black dashed), the $a_0 = 1$ sparse embedding with static synapses or homogeneous plastic synapses (red solid, red dashed), and the heterogeneous system (magenta solid).(Bottom) The number of neurons that spike during external driving for $\mathcal{G}_2$ under the full spin glass embedding with static synapses or plastic synapses (black solid, black dashed), the $a_0 = 1$ sparse embedding with static synapses or homogeneous plastic synapses (blue solid, blue dashed), and the heterogeneous system (magenta solid).

However, simply generating larger (yet bounded) spike responses was not sufficient to find the minimum variance of information (MVI). The addition of synapse plasticity was not guaranteed to increase spike responses. If the spike response was less than the mean degree of the graph, meaning that not all neighbors of a driven neuron fired, we identify that as a significantly suppressed spike response (SSR). For each of the three graphs studied we will discuss how the various embeddings and synaptic types affected the spike response sizes, the MVI, and the width of the h_0-plateau (if the MVI vanishes).

First, we discuss the spike responses on graph instance $\mathcal{G}_0$ (see Fig. 6); the sparsest graph, with the smallest communities and the lowest mean degree. With static synapses, all three sparse embeddings lead to larger spike responses yet only $a_0 = 1$ and $a_0 = 3$

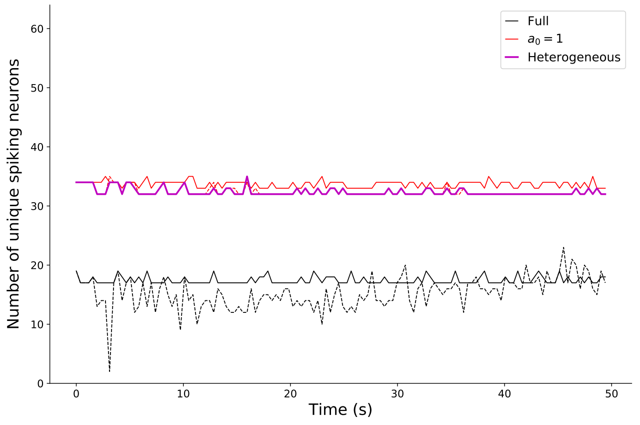

Figure 7: The number of neurons that fire when a single neuron is driven in $\mathcal{G}_1$. Under the full spin-glass or $a_0 = 1$ embedding with static synapses (solid black, red) the spike response is bounded. This localized response is maintained with plastic synapses (dashed black, dashed red) and with the heterogeneous embedding (magenta, solid), though there is significant variance in the spike response size under the spin glass embedding with plastic synapses.

Table 4: Minimum information of variance for SNN with plastic synapses

Graph	Embed	MVI	$[h_0]$	Δh_0
$\mathcal{G}_0$	full	1.601	∅	∅
$\mathcal{G}_0$	$a_0 = 1$	0.069	∅	∅
$\mathcal{G}_0$	$a_0 = 2$	1.935	∅	∅
$\mathcal{G}_0$	$a_0 = 3$	0.608	∅	∅
$\mathcal{G}_0$	heterogeneous	0.497	∅	∅
$\mathcal{G}_1$	full	0.000	[0.9388]	0
$\mathcal{G}_1$	$a_0 = 1$	0.000	[0.8, 0.9387]	0.1387
$\mathcal{G}_1$	$a_0 = 2$	0.250	∅	∅
$\mathcal{G}_1$	$a_0 = 3$	2.000	∅	∅
$\mathcal{G}_1$	heterogeneous	0.000	[0.8, 0.9878]	0.1878
$\mathcal{G}_2$	full	1.685	∅	∅
$\mathcal{G}_2$	$a_0 = 1$	0.000	[0.9551, 0.9918]	0.0367
$\mathcal{G}_2$	$a_0 = 2$	0.888	∅	∅
$\mathcal{G}_2$	$a_0 = 3$	0.926	∅	∅
$\mathcal{G}_2$	heterogeneous	0.000	[0.9673, 0.9755]	0.0082

showed a complete vanishing of the variance of information (returned the ground truth partitioning). With plastic synapses, the spike responses became unstable for the sparsest embeddings; both $a_0 = 2$ and $a_0 = 3$ embeddings exhibited runaway spike cascades, which lead to an increase in MVI. Of the three embeddings with plastic synapses that had bounded spike responses (the full spin glass, $a_0 = 1$, and the heterogeneous embeddings), all showed SSR. This SSR is most prominent for the full spin glass embedding, which also had the largest increase in MVI. For the $a_0 = 1$ embedding and for the heterogeneous embedding, 3 neurons exhibited SSR and an increase in MVI.

Next, we discuss the spike responses on graph instance $\mathcal{G}_2$ (see Fig. 6); this was the largest graph, but with a degree of sparsity close to $\mathcal{G}_0$. With static synapses, only the full glass embedding and the $a_0 = 1$ embedding result in bounded spike responses, and the larger response in the $a_0 = 1$ embedding was sufficient to cause the MVI to vanish. While not full spike cascades, $a_0 = 2$ and $a_0 = 3$ both lead to very large spike responses that nearly covered the entire graph and their respective MVI did not vanish. With the addition of plastic synapses, the full glass embedding exhibited SSR and an increase in MVI, the $a_0 = 1$ embedding only exhibited SSR for 2 neurons and its MVI still vanished. For $\mathcal{G}_2$, the plastic synapses and the sparse $a_0 = 1$ embedding also showed an increase in plateau width over which the ground truth partitioning was returned. Finally, the heterogeneous embedding with plastic synapses showed a bounded, larger spike response without any SSR. For this embedding the MVI vanished over a range of h_0 values of plateau width $\Delta h_0 = 0.0082$.

Of all three graphs studied, only $\mathcal{G}_1$, the densest graph, exhibited significant SSR yet also had a MVI that vanished (see Fig. 7). However the plateau of h_0 over which the MVI vanished was reduced to just a single value. With the addition of plasticity, the sparse embedding $a_0 = 1$ did not exhibit any SSR, but the spike response sizes were smaller than that of the static synapse embedding. While both $a_0 = 1$ embeddings exhibited a plateau over which the MVI vanished, the static synapse embedding had a wider plateau. With the addition of plasticity, the sparse embedding $a_0 = 2$ showed spike cascades for 2 neurons and a non-vanishing MVI. The sparse embedding $a_0 = 3$, with static or plastic synapses, showed spike cascades for all neurons. Finally, the heterogeneous embedding showed the largest plateau over which the MVI vanished: $\Delta h_0 = 0.1878$.

Finally, for all of the graphs studied there was at least one sparse embedding, either using static or plastic synapses, that led to neurons driving a spike response from the entire system. For the sparser graphs $\mathcal{G}_0, \mathcal{G}_2$, this occurred for sparse embeddings with plastic synapses while on the denser graph $\mathcal{G}_1$ this occurred for sparse embeddings with either synapse type. Spike cascades on sparser graphs only occurred after a significant number of neurons were driven (e.g. in Fig. 3 the spike response on $\mathcal{G}_0$ gradually grows in size), while for $\mathcal{G}_2$ cascades could be triggered by driving only one neuron. Of all three graphs studied, only $\mathcal{G}_2$ returned a trivial solution with all vertices being assigned the same community.

5 CONCLUSIONS

We have presented preliminary results on improving the performance of spiking label propagation on sparse graphs. Spiking label propagation depends on localized spike responses and the spike responses between neurons in the same community must have significant overlap. Depending on the sparsity of the graph under analysis, a fully connected spin glass may lead to spike responses which are localized to just nearest neuron neighbors, whereas a sparse embedding applied to a dense graph may lead to runaway spike cascades. Thurs, we have made modifications to the spin-glass SNNs used for spiking label propagation in order to increase the size of spike responses on sparse graphs: using sparser embeddings via the reduction of inhibitory synapses, and the introduction of plasticity in the synaptic connections.

Designing sparse embeddings to suppress specific degrees of connectivity requires a significant amount of information about the graph connectivity, and is not guaranteed to improve performance. Relying on only the graph connectivity to define a sparse embedding is difficult to do without making assumptions about the connectivity within communities. For all graphs, the $a_0 = 1$ embedding with static synapses led to a consistent vanishing MVI, which is likely due to our assumption that graph communities are at least 2-connected, meaning that at least 2 edges must be removed in order to disconnect an entire community from the rest of the graph. However, this does not guarantee that any two neurons within the same community have at multiple length-2 paths connecting them. On $\mathcal{G}_0$ there are multiple neurons which only have a single length-2 path between, but on denser graphs it is likely that two neurons with only a single length-2 path between them contain an edge that connects two different communities. Adding inhibitory synapses according to the condition $(A^2)_{ij} = 1$ will prevent neurons in different communities from firing at the same time. Predicting a threshold for sparser embedding becomes difficult for denser graphs; on $\mathcal{G}_2$ there are more connections between communities and a larger number of inhibitory synapses are needed to prevent runaway spike cascades. Thus the $a_0 = 3$ embedding performs poorly for $\mathcal{G}_1$ but returns the correct partitioning for $\mathcal{G}_0$.

Likewise, our initial implementation of homogeneous synaptic learning did not significantly improve performance to systems with uniform synaptic weights, and by introducing the possibility of SSR actually reduced the accuracy. For example, on $\mathcal{G}_0$ the three sparse embeddings $a_0 = 1$, $a_0 = 2$ or $a_3 = 0$ all showed increased spike responses, but only $a_0 = 1$ and $a_0 = 3$ showed vanishing MVI. This vanishing MVI was lost after the addition of synapse plasticity with homogeneous learning. While a sparse embedding with plastic synapses was not found for $\mathcal{G}_0$ that lead to vanishing MVI, a more general embedding that utilizes heterogeneous synaptic values and heterogeneous learning rates improved performance on $\mathcal{G}_1$ and $\mathcal{G}_2$. This embedding, which added differently weighted inhibitory synapses with different learning rates between neurons that satisfied $(A^2)_{ij} = 1$, $(A^2)_{ij} = 2$ or $(A^2)_{ij} = 3$ was able to generate larger yet still localized spike responses and improved the stability of non-trivial partitioning solutions. It resulted in the largest stable plateau for $\mathcal{G}_1$.

Overall, the preliminary results we have obtained so far indicate that simply increasing the size of spike responses of relying on knowledge of graph connectivity to design a sparse embedding is insufficient to ensure that the modular structure of a graph can be fully described, and is of limited applicability to graphs with unknown ground truths. Also, the full control of spike cascades remains an open question, but our results indicate that if spike cascades can be limited to a few neurons this can mitigate the catastrophic effects on label propagation and avoid trivial partition results. This leads us to conclude that a fully adaptable implementation of spiking label propagation will be a system with heterogeneous synapses and learning rates. In such a system we believe that the current implementation of spiking label propagation will need to be fully integrated into the spike generation. Instead of executing label propagation after a full spike raster is generated, future implementations will focus on using label assignments to drive the synaptic learning.

ACKNOWLEDGMENTS

Research sponsored in part by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U. S. Department of Energy.

REFERENCES

[1] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Phys. Rev. E* 70 (Dec 2004), 066111. Issue 6. https://doi.org/10.1103/PhysRevE.70.066111

[2] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. 2018. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro* 38, 1 (2018), 82–99.

[3] Gabriel Andrés Fonseca Guerra and Steve B Furber. 2017. Using Stochastic Spiking Neural Networks on SpiNNaker to Solve Constraint Satisfaction Problems. *Frontiers in neuroscience* 11 (2017), 714.

[4] Santo Fortunato. 2017. Santo Fortunato's Website: Software. https://sites.google.com/site/santofortunato/inthepress2. (2017). Accessed: 2017-05-10.

[5] L. Gomes. 2017. Special report : Can we copy the brain? - The neuromorphic chip's make-or-break moment. *IEEE Spectrum* 54, 6 (June 2017), 52–57. https://doi.org/10.1109/MSPEC.2017.7934233

[6] Dan Goodman and Romain Brette. 2008. Brian: A Simulator for Spiking Neural Networks in Python. *Frontiers in Neuroinformatics* 2 (2008).

[7] Kathleen E. Hamilton and Travis S. Humble. 2018. Spiking spin-glass models for label propagation and community detection. *arXiv preprint arXiv: 1801.2128201* (2018).

[8] Kathleen E. Hamilton, Neena Imam, and Travis S. Humble. 2017. Community detection with spiking neural networks for neuromorphic hadware. In *Proceedings of ORNL Neuromorphic workshop*. ACM. to be published.

[9] Kathleen E. Hamilton, Catherine D. Schuman, and Travis S. Humble. 2018. Algebraic analysis of spiking neural networks for graph partitioning. Presentation at SIAM Conference on Discrete Mathematics.

[10] Dandan Hu, Peter Ronhovde, and Zohar Nussinov. 2012. Phase transitions in random Potts systems and the community detection problem: spin-glass type and dynamic perspectives. *Philos. Mag.* 92, 4 (2012), 406–445.

[11] Mark D Humphries. 2011. Spike-train communities: finding groups of similar spike trains. *Journal of Neuroscience* 31, 6 (2011), 2321–2336.

[12] Zeno Jonke, Stefan Habenschuss, and Wolfgang Maass. 2016. Solving Constraint Satisfaction Problems with Networks of Spiking Neurons. *Frontiers in Neuroscience* 10 (2016), 118. https://doi.org/10.3389/fnins.2016.00118

[13] Marina Meilă. 2007. Comparing clusterings?an information based distance. *Journal of multivariate analysis* 98, 5 (2007), 873–895.

[14] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. 2014. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 6197 (2014), 668–673.

[15] Marcos G Quiles, Liang Zhao, Fabricio A Breve, and Anderson Rocha. 2010. Label propagation through neuronal synchrony. In *Neural Networks (IJCNN), The 2010 International Joint Conference on.* IEEE, 1–8.

[16] Marcos G Quiles, Ezequiel R Zorzal, and Elbert EN Macau. 2013. A dynamical model for community detection in complex networks. In *Neural Networks (IJCNN), The 2013 International Joint Conference on.* IEEE, 1–8.

[17] Peter Ronhovde and Zohar Nussinov. 2010. Local resolution-limit-free Potts model for community detection. *Physical Review E* 81, 4 (2010), 046114.

[18] Catherine D Schuman, Raphael Pooser, Tiffany Mintz, Md Musabbir Adnan, Garrett S Rose, Bon Woong Ku, and Sung Kyu Lim. 2017. Simulating and Estimating the Behavior of a Neuromorphic Co-Processor. In *Proceedings of the Second International Workshop on Post Moores Era Supercomputing*. ACM, 8–14.