INVITED PAPER

Braindrop: A Mixed-Signal Neuromorphic Architecture With a Dynamical Systems-Based Programming Model

This paper provides an overview of a current approach for the construction of a programmable computing machine inspired by the human brain.

By Alexander Neckar, Sam Fok, Ben V. Benjamin, Terrence C. Stewart, Nick N. Oza, Aaron R. Voelker, Chris Eliasmith, Rajit Manohar, *Senior Member IEEE*, and Kwabena Boahen, *Fellow IEEE*

ABSTRACT | Braindrop is the first neuromorphic system designed to be programmed at a high level of abstraction. Previous neuromorphic systems were programmed at the neurosynaptic level and required expert knowledge of the hardware to use. In stark contrast, Braindrop's computations are specified as coupled nonlinear dynamical systems and synthesized to the hardware by an automated procedure. This procedure not only leverages Braindrop's fabric of subthreshold analog circuits as dynamic computational primitives but also compensates for their mismatched and temperature-sensitive responses at the network level. Thus, a clean abstraction is presented to the user. Fabricated in a 28-nm FDSOI process, Braindrop integrates 4096 neurons in 0.65 mm^2. Two innovations—sparse encoding through analog spatial convolution and weighted spike-rate summation though digital accumulative thinning—cut digital traffic drastically, reducing the energy Braindrop consumes per equivalent synaptic operation to 381 fJ for typical network configurations.

KEYWORDS | Analog circuits; artificial neural networks; asynchronous circuits; neuromorphics.

I. INTRODUCTION

By emulating the brain's harnessing of analog signals to efficiently compute and communicate, we can build artificial neural networks (ANNs) that perform dynamic computations—tasks involving time—much more energy efficiently.

Harnessing analog signals in two important ways enables biological neural networks (BNNs) to save energy by using much more energetically expensive digital communication sparingly [1]. First, BNNs exploit the nerve membrane's local capacitance to continuously and dynamically update their analog somatic potentials, sparsifying their digital axonal signaling in time. Second, BNNs exploit local fan-out to reduce long-range communication by propagating their analog dendritic signals across $O(n)$ distance to $O(n^2)$ somas,[1] sparsifying their digital axonal signal-

[1]The cortical sheet's third dimension is much shorter than its first two (2–3 mm versus tens of centimeters).

Manuscript received April 24, 2018; revised October 12, 2018 and November 12, 2018; accepted November 12, 2018. Date of current version December 21, 2018. *(Corresponding author: Alexander Neckar.)*

This work was supported in part by the ONR under Grants N000141310419 and N000141512827. The work of A. Voelker was supported by OGS and NSERC CGS D. The work of C. Eliasmith was supported by the Canada Research Chairs Program under NSERC Discovery Grant 261453.

A. Neckar, **S. Fok**, and **B. V. Benjamin**, were with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: aneckar@gmail.com).

T. C. Stewart, **A. R. Voelker**, and **C. Eliasmith** are with the Centre for Theoretical Neuroscience, University of Waterloo, Waterloo, ON N2L 3G1, Canada.

N. N. Oza was with the Department of Bioengineering, Stanford University, Stanford, CA 94305 USA.

R. Manohar is with the Department of Electrical Engineering, Yale University, New Haven, CT 06520 USA.

K. Boahen is with the Department of Bioengineering, Stanford University, Stanford, CA 94305 USA, and also with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA.

Digital Object Identifier 10.1109/JPROC.2018.2881432

ing in space. This complementary relationship between analog dendritic signaling and digital axonal signaling is completely lost in traditional ANN implementations that replace discrete spike trains by continuous rates, dynamic neuronal behavior with static point nonlinearities, and spatially organized neuron arrays mimicking BNNs' local connectivity with globally connected neurons.

The price to be paid for mimicking the BNNs' energy-efficient mixed-signal approach with modern CMOS process technology is the uncertainty in manufacturing. This uncertainty results in exponentially mismatched responses and thermal variability in analog (i.e., physically realized) neuron circuits. Because these circuits are not time-multiplexed, they must be sized as small as possible to maximize neuron count, and because these circuits constantly conduct their bias current, they must be biased with as little current as possible to minimize (static) power consumption. However, these two requirements make designing analog circuits in modern CMOS processes even more challenging; smaller transistors have more mismatched threshold voltages, and minuscule currents are exponentially sensitive to this mismatch [2] as well as to the ambient temperature [3].

Thus, while using analog signaling promises energy efficiency because of its potential to sparsify the digital communication in space and time, analog circuits' inherent heterogeneity and variability impede programmability and reproducibility. This heterogeneity and variability are directly exposed to the user when the mixed-signal neuromorphic systems are programmed at the level of individual neuronal biases and synaptic weights [4]–[6]. Because each chip is different, for a given computation, each must be configured differently. In addition, the silicon neurons inherit their transistors' thermal variation, requiring further fine-tuning of programming parameters. This lack of abstraction and reproducibility limits adoption to experts who understand the hardware at the circuit level. To ease programmability and guarantee reproducibility, some recent large-scale neuromorphic systems adopt an all-digital approach [7], [8].

This paper presents Braindrop (see Fig. 1), the first mixed-signal neuromorphic system designed with a clean set of mismatch- and temperature-invariant abstractions in mind. Unlike previous approaches for analog computation [9]–[12], which use fewer, bigger analog circuits biased with large currents to minimize mismatch (and its associated thermal variation), Braindrop's hardware and software embrace mismatch, working in concert to harness the inherent variability in its analog electronics to perform computation, thereby presenting a clean abstraction to the user. Orchestrating hardware and software automatically is enabled by raising the level of abstraction at which the user interacts with the neuromorphic system.

The user describes their computation as a system of nonlinear differential equations, agnostic to the underlying hardware. Automated synthesis proceeds by characterizing the hardware and implementing each equation using a group of neurons that are physically colocated

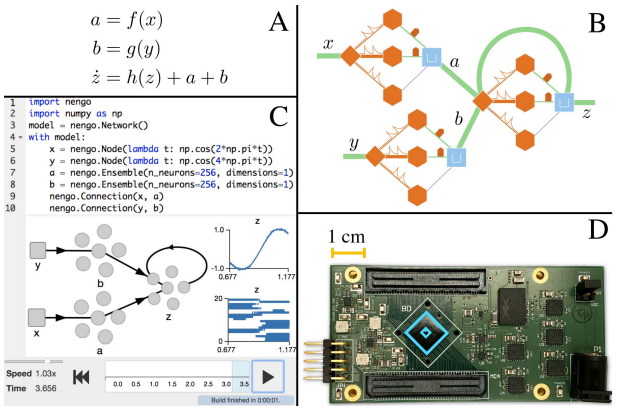


Fig. 1. ***Mapping a computation onto Braindrop. (a) Desired computation is described as a system of coupled dynamical equations. (b) NEF describes how to synthesize each subcomputation using a pool of dynamical neurons. (c) User uses Nengo, the NEF's Python-based programming environment, to translate the equations into a network of pools. (d) Computation is implemented on Braindrop (blue outer outline indicates the package; inner outline indicates the die's core circuitry). Nengo communicates with Braindrop through its driver software to provide a real-time interface.***

(called a pool). This computing paradigm, theoretically underpinned by the Neural Engineering Framework (NEF) [13], is not only tolerant of, but also reliant on, mismatch; neuron responses form a set of basis functions that must be *dissimilar* and *overcomplete*. Dissimilarity enables arbitrary functions of the input space to be approximated by a linear transform. Overcompleteness ensures that the solutions exist in the null-space of the set's thermal variation. Thus, these two properties enable us to abstract the analog soma and synapse circuits' idiosyncrasies away.

Section II briefly reviews the NEF, the level of abstraction at which a user interacts with the Braindrop system. Section III highlights accumulative thinning and sparse encoding, novel hardware implementations of the NEF's linear decoding and encoding that sparsify digital communication in time and space, respectively. Section IV describes Braindrop's architecture and discusses its hardware implementation and software support. Section V characterizes and validates the hardware's decoding and encoding operations. Section VI demonstrates the performance of several example applications currently running on the Braindrop. Section VII introduces an energy-efficient metric for spiking neural network (SNN) architectures with different connectivities—energy per equivalent synaptic operation—and determines it for Braindrop over varying operating configurations. Section VIII compares Braindrop's energy and area efficiencies with other SNN architectures. Section IX presents our conclusions.

II. NEURAL ENGINEERING FRAMEWORK

The NEF provides a way to translate a computation specified as a differential equation into a network of somas and

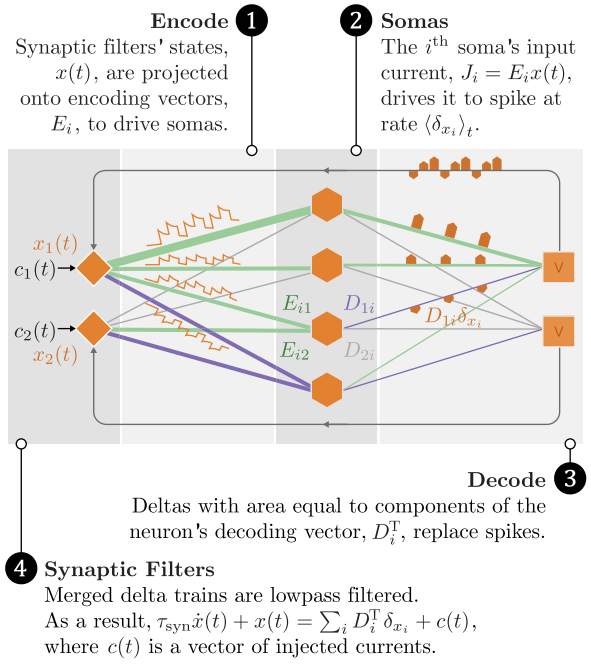


Fig. 2. ***Emulating the nonlinear dynamical system*** $\tau_{\text{dyn}}\dot{x}(t) = f(x) + u(t)$ ***with a SNN. This system's dynamics are matched by the SNN's state vector, x(t), when*** D_i^{T} ***and c(t) are assigned such that*** $\sum_i D_i^{\text{T}} \delta_{x_i} \approx \tau_{\text{syn}}/\tau_{\text{dyn}} f(x) + x$ ***(after synaptic filtering) and*** $c(t) = \tau_{\text{syn}}/\tau_{\text{dyn}} u(t)$***. Branches of a soma's dendrite and axon realize weighting. Line thickness depicts weight magnitude. Sign may be positive (green line), negative (purple line), or zero (gray line).***

synapses interconnected via dendrites and axons. A soma is viewed as implementing a static nonlinear function, whose argument is a continuous current and whose value is the soma's spike train. A synapse is viewed as the implementing leaky integration (i.e., low-pass temporal filtering), thereby converting these spike trains back into a continuous current. A differential equation's state variable (x), which may be multidimensional, is represented by a vector of d current signals. The equation specifies a transformation $[f(x)]$ of this vector of d input current signals into another vector of d output current signals. This transformation is realized—and temporally integrated—by a collection (or pool) of N somas and d synaptic filters in four steps (see Fig. 2).

First, differently weighted sums of the d input currents are fed into each of the N somas (one per soma), a linear mapping known as *encoding*. Based on its particular weighting, each soma in the pool will provide a stronger response for a particular set of input vectors. A soma is excited (receives positive current) when the vector points in its preferred direction, and it is inhibited (receives negative current) when it points away. The NEF chooses these directions—specified by *encoding vectors*—randomly to ensure that all directions are represented with equal probability.

Second, these N current inputs are transformed by the somas' static nonlinearities into N spike trains, a pointwise nonlinear mapping. Before passing the input current through its static nonlinearity, each soma scales it by a gain and adds a bias current. The NEF assigns somas gains and biases drawn from a wide distribution, resulting in a heterogenous set of nonlinearities. Compounded with their randomly drawn encoding vectors, the somas' nonlinear responses—called *tuning-curves*—form a dissimilar and overcomplete basis set for approximating the arbitrary multidimensional transformations of the input vector.

Third, these N spike trains are converted into d weighted sums, another linear mapping known as *decoding*. For the weights, a *decoding vector* is assigned to each soma by solving a least-squares optimization problem (see Fig. 3). When each spike is viewed as a delta function with unit area, replacing it with a delta function with area equal to a component of the decoding vector and merging the resulting delta trains together yields the desired weighted sum for one of the d dimensions.

Finally, the synaptic filters' leaky integration is cleverly exploited to integrate the transformed vector, a critical difference between the NEF, and other random-projection networks, such as Extreme Learning Machines [14]–[16], which cannot realize a dynamic transformation. This operation is accomplished by feeding the merged, scaled delta trains to the synaptic filters and adding their d output currents to the d input currents through (recurrent) feedback connections (see Fig. 2). Thus, nonlinear differential equations of arbitrary order may be implemented by a single pool [13].

More elaborate computations are first decomposed into a coupled system of differential equations, and then, each one is implemented by one of an interconnected set of pools. These pools are interconnected by linking one pool's decoder to another pool's encoder to form large network graphs. Linear transforms may be placed between decoders and encoders (see Fig. 4). The resulting SNN's connectivity is defined by encoding vectors, decoding vectors, and

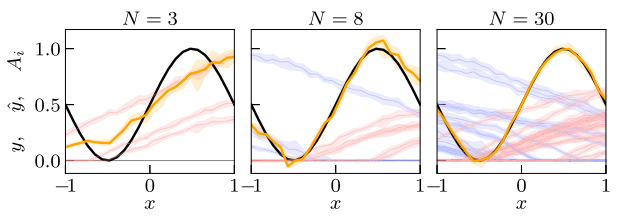


Fig. 3. ***Transformation*** $y = 1/2(\sin(\pi x) + 1)$ ***(black curve) is approximated by*** $\hat{y} = AD$ ***(yellow curve), where each of*** A***'s columns represent a single neuron's spike rates over*** x***'s range (blue curve for negative- and red curve for positive preffered directions) and*** D ***is a vector of decoding weights, or, more generally, a matrix of decoding vectors.*** D ***is obtained by solving for*** $\text{argmin}_D ||AD - y||_2^2 + \lambda ||D||_2^2$***. To produce each panel, a basis pursuit is first performed on 1024 neurons' tuning curves collected from Braindrop to select the best sets of 3, 8, and 30 neurons' responses to form*** A***, thereby demonstrating the effect of using more neurons on performance. Error bars represent 10th and 90th percentiles when sampling for 0.3 s/point.***

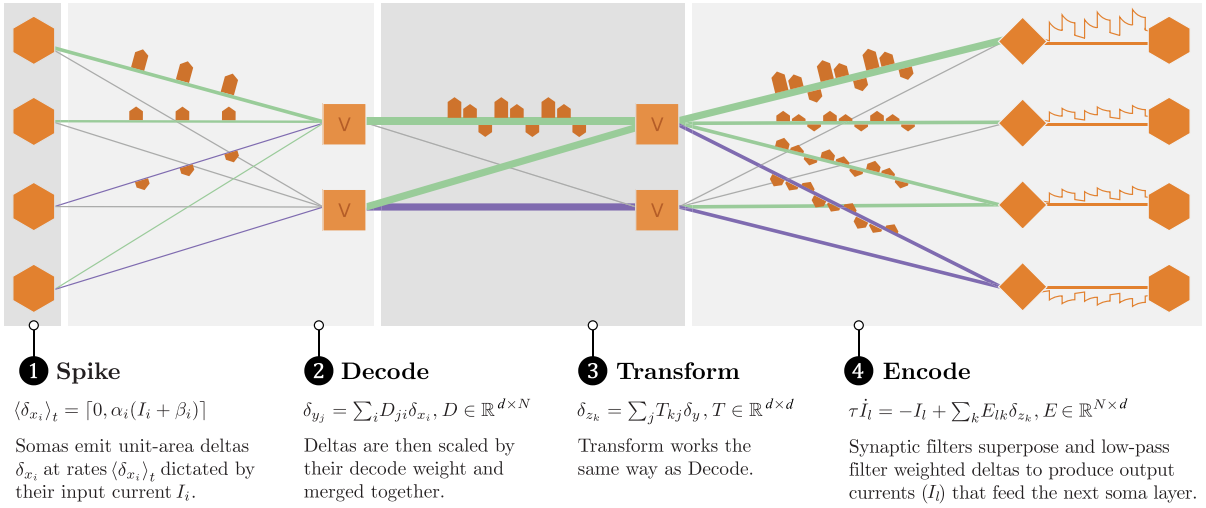


Fig. 4. ***Operations and signal representations for an NEF network with two pools of neurons connected by decode-transform-encode weights. Unlike Fig. 2, filtering takes place after encoding, leveraging these two linear operations' commutativity. Three neurons emit three deltas each. Their areas are equal to the weights:*** $1/2$, $1/3$, ***and*** $-1/4$, ***from top to bottom. Merges (*** $\vee$***) conserve deltas and fan-out replicates them, and thus, the original nine deltas turn into 36 deltas by the time they reach the four synaptic filters.***

transform matrices, which occupy much less memory than fully connected synaptic weight matrices, an appealing feature of the NEF.

The following five features make the NEF a particularly appealing synthesis framework for mixed-signal neuromorphic hardware.

1) Heterogeneous neuronal gains and biases—due to transistor threshold-voltage mismatch—naturally yield a diverse set of basis functions, which is desirable for function approximation.
2) This set's overcompleteness may be exploited to mitigate the tuning curves' thermal variation by finding solutions in the null-space [17], [18].
3) Analog synaptic filtering provides a native dynamical primitive, which is necessary for emulating arbitrary dynamical systems.
4) Theoretical extensions enable mismatch in synaptic filters' time constants as well as their higher order dynamics to be accurately accounted for [19], [20].
5) Communicating between neurons using spikes increases scalability because digital signaling is resilient to noise—unlike analog signaling.

III. MAPPING THE NEF TO NEUROMORPHIC HARDWARE

In a large-scale neuromorphic system, digital communication constitutes the largest component of the total system's power budget. Therefore, to improve energy efficiency, we have focused on reducing the digital communication in both time and space (i.e., making it more sparse, spatiotemporally). For temporal sparsity, we invented *accumulative thinning*, and for spatial sparsity, we invented *sparse encoding* (see Fig. 5).

Accumulative thinning is a digitally implemented, linear-weighted-sum operation that sparsifies digital communication in time. This operation is performed in three steps: 1) translate spikes into deltas with area equal to their weights; 2) merge these weight-area deltas into a single train; and 3) convert this train into a unit-area delta train (i.e., back to spikes),[2] whose rate equals the weighted sum of input-spike rates. For the usual case of weights smaller than 1, this method reduces total delta counts through layers, communicating more sparsely than prior approaches, which used probabilistic thinning, while achieving the same signal-to-noise ratio (SNR).

Sparse encoding represents the encoder not as a dense matrix but as a sparse set of digitally programmed locations in a 2-D array of analog neurons. Each location, called a *tap point*,[3] is assigned a particular preferred direction, called an *anchor-encoding vector*. The *diffusor*—a transistor-based implementation of a resistive mesh—convolves the output of these tap points with its kernel to realize well-distributed preferred directions. Thus, neurons can be assigned with encoding vectors that tile the multidimensional state space fairly uniformly using tap points as sparse as one per several dozen neurons.

Using digital accumulators for decoding and analog convolution for encoding supports the NEF's abstractions while sparsifying digital communication spatiotemporally, thereby improving Braindrop's energy efficiency.

A. Decoding by Accumulative Thinning

Matrix-multiply operations lead to the traffic explosion when the weighted spike trains—represented by delta trains—are combined by merging. One merge is associated with each output dimension. When a spike associated

[2]Only a spike's time of occurrence carries information, whereas a delta's area carries information as well as its timing.

[3]Meant to evoke the *taproot* of some plants, a thick central root from which smaller roots spread.

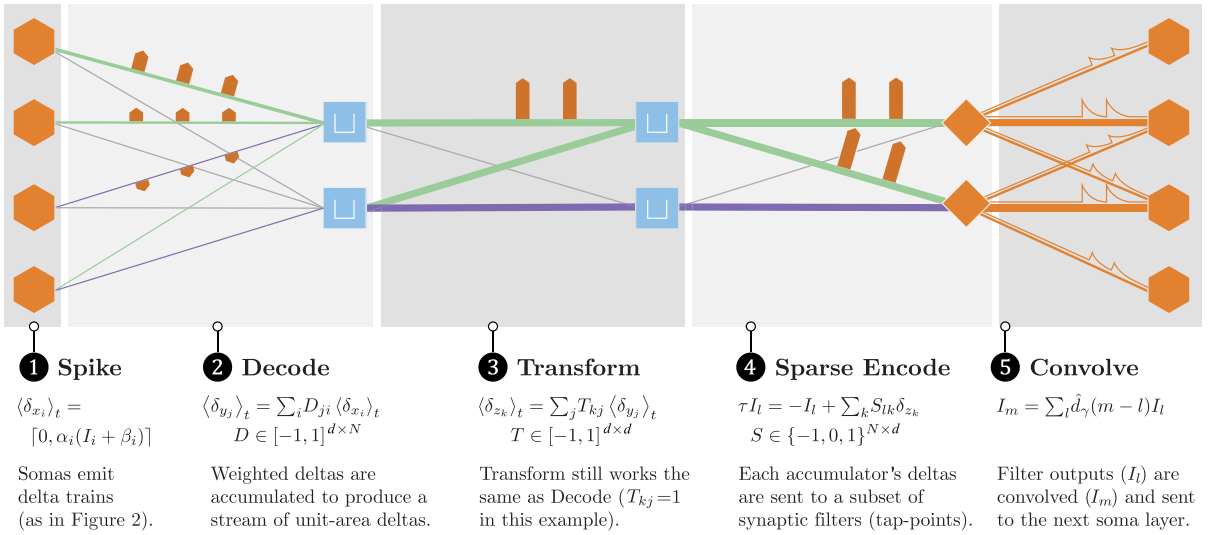


Fig. 5. ***Braindrop's computational model replaces merges in Fig. 4 with accumulators (⊔) and its encoding with sparse encoding followed by convolution. The nine original deltas are thinned to two before being sparsely fanned out, delivering four deltas to the synaptic filters instead of 36, while still reaching all the somas.***

Algorithm 1 Accumulator Update

Require: input $w \in [-1, 1]$
  $x := x + w$
  if $x \geq 1$ **then**
    emit $+1$ output
    $x := x - 1$
  else if $x \leq -1$ **then**
    emit -1 output
    $x := x + 1$
  end if

with a particular input dimension occurs, the matrix's corresponding column is retrieved and deltas with area equal to each of the column's components are merged onto the output for each corresponding dimension, multiplying the traffic. As a result, $O(d_{\text{in}})$ spikes entering a matrix $M \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$ result in $O(d_{\text{in}} d_{\text{out}})$ deltas being the output. This multiplication of traffic compounds with each additional weight-matrix layer. Thus, $O(N)$ spikes entering a $N \times d \times d \times N$ decode-transform-encode multiply to $O(N^2 d^2)$ deltas (see Fig. 4).

Accumulative thinning computes a linearly weighted sum of spike rates while reducing the number of deltas that must be fanned out. The deltas' areas—set equal to the weight—are summed into a two-sided thresholding accumulator to produce a train of signed unit-area deltas (see Algorithm 1). Because this operation is area-conserving, the synaptic filter's leaky integration produces virtually the same result as it does for the merged weight-area delta trains. For the usual case of weights smaller than one, however, the accumulator's state variable x (see Algorithm 1) spaces output deltas further apart (and more evenly).[4] Thus, it can cut a $N \times d \times d \times N$ decode-transform-encode network's delta traffic from $O(N^2 d^2)$ to $O(Nd)$ (compare Figs. 4 and 5), sparsifying its digital communication in time by a factor Nd.

In practice, the total traffic is dictated by point-process statistics of delta trains fed to synaptic filters and desired SNR of their outputs. Traditionally, spike-train thinning-as-weighting for neuromorphic chips has been performed probabilistically as Bernoulli trials [21], [22], which also produce unit-area-delta trains, but with Poisson statistics, whereas the accumulator operates as a deterministic thinning process (i.e., decimation), which produces trains with periodic statistics (when the weights are small and assumed to be equal). With the Poisson statistics, SNR scales as $\sqrt{\lambda}$, where λ is the rate of deltas, whereas with periodic statistics, it scales as λ. Thus, the accumulator can preserve most of its input's SNR while outputting much fewer deltas than prior probabilistic approaches (see Fig. 6), thereby sparsifying digital communication in time. (For a mathematical analysis, see Appendix A.)

Replacing the merging of inhomogeneous-area deltas with the accumulation of their areas not only avoids traffic explosion but also enables us to use a much simpler analog synaptic filter circuit. For 8-bit weights, delta's areas are represented by 8-bit integers. If these multibit values serve as input, a digital-to-analog converter (DAC) is needed to produce a proportional amount of current. This analog operation must be performed with the requisite precision, making the DAC extremely costly in

[4]For an output-delta rate of F_{out}, the accumulator's step response is jittered in time by up to $1/F_{\text{out}}$ due to variation in its initial state. This jitter is negligible if it is much shorter than the synaptic filter's time constant, τ, because the filter's delay is of the order of τ. Its step response rises to 63% $(1 - 1/\text{e})$ of its final value in τ s.

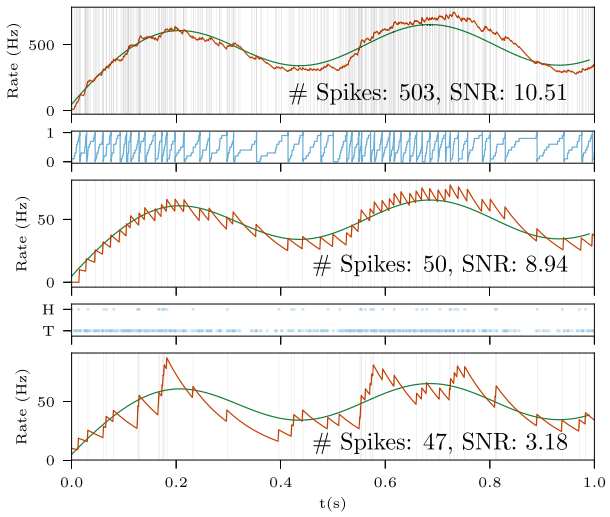


Fig. 6. ***Standard Decoding versus Accumulator versus Bernoulli Trials. Top: deltas (vertical lines) generated by an inhomogeneous Poisson process are synaptically filtered (orange curve); the ideal output is also shown (green curve). Middle: accumulator fed the same deltas yields a similar SNR*** $= E[Z]/\sqrt{\mathrm{var}(Z)}$***, calculated for the filtered train*** Z ***over a longer run. Its state (first inset) increments by*** $w = 0.1$ ***with each input delta and thresholds at 1, triggering an output delta. Bottom: biased coin (*** $p = 0.1$***) is flipped for each delta. When the coin returns heads (second inset), an output delta is generated. This method accomplishes the same weighting but yields poor SNR.***

area. To avoid using it, compact silicon-synapse circuit designs take in unit-area deltas with signs denoting excitatory and inhibitory inputs [23]. The accumulator produces the requisite unit-area deltas—modulating only their sign and rate to convey the decoded quantity.

In summary, the accumulator produces a unit-area-delta train with statistics approaching a periodic process, yielding higher SNR—for the same output delta rate—than Bernoulli weighting, which only produces Poisson statistics.

B. Sparse Encoding by Spatial Convolution

We use convolution, implemented by analog circuits, to efficiently fan-out and mix outputs from a sparse set of tap points, leveraging the inherent redundancy of NEF encoding vectors. In the NEF, encoding vectors that uniformly tile a d-sphere's surface are generally desirable. Because these encoding vectors form an overcomplete basis for the d-D input space, the greatest fan-out takes place during encoding (see Fig. 4). Performing this high fan-out efficiently as well as reducing the associated large number of digitally stored encoding weights motivated us to use convolution (see Fig. 5).

The diffusor—a hexagonal resistive mesh implemented with transistors—convolves synaptic filter outputs with a radially symmetric (2-D) kernel and projects the results to soma inputs (see Fig. 7). It interfaces the synaptic filter's output currents with the somas' input currents. The output currents decay exponentially as they spread among nearby somas [24]. The space constant of decay is tuned by adjusting the gate biases of the diffusor's transistors.

We leverage the commutativity of synaptic filtering and convolution operations in our circuit design. By performing synaptic filtering before convolution, and recognizing that a kernel need not be centered over every single neuron, we may reduce the number of (relatively large) synaptic filter circuits. Hence, the diffusor acts on temporally filtered subthreshold currents, and there is only one synaptic filter circuit for every four neurons. Through these two operations' commutativity, this solution is equivalent to the NEF's usual formulation (compare Figs. 4 and 5).

By choosing the adjacent tap points' anchor-encoding vectors to be orthogonal, it is possible to assign varied encoding vectors to all neurons without encoding each one digitally. The diffusor's action implies that nearby neurons receive similar input currents and, therefore, have similar encoding vectors (i.e., relatively small angles apart). Hence, neighboring anchor-encoding vectors that are aligned with each other yield similarly aligned encoding vectors for the neurons in between. This redundancy does not contribute to tiling the d-sphere's surface. Conversely, vectors between tap points with orthogonal anchors span a 90° arc, boosting coverage.

For 2-D and 3-D input spaces, just four and nine tap points, respectively, provide near-uniform tiling (see Fig. 8). Coverage may be quantified by plotting the distribution of angles to the nearest encoding vector for points randomly chosen on the unit d-sphere's surface

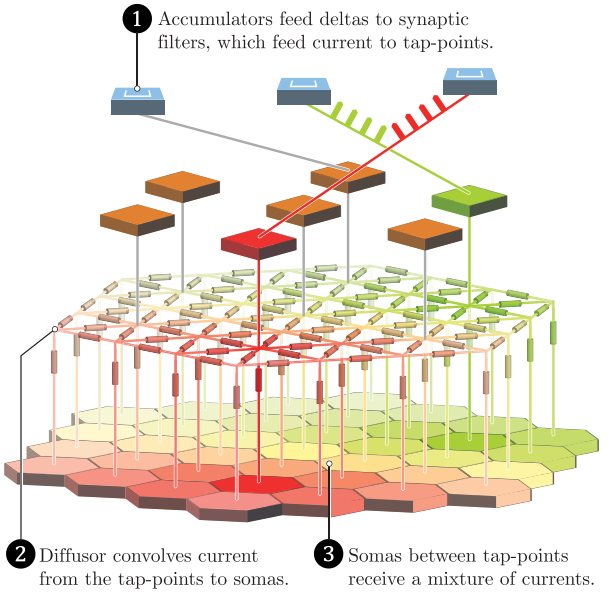


Fig. 7. ***Diffusor operation. Somas are colored according to the proportion of input received from the red or green delta trains and shaded according to the total input magnitude received.***

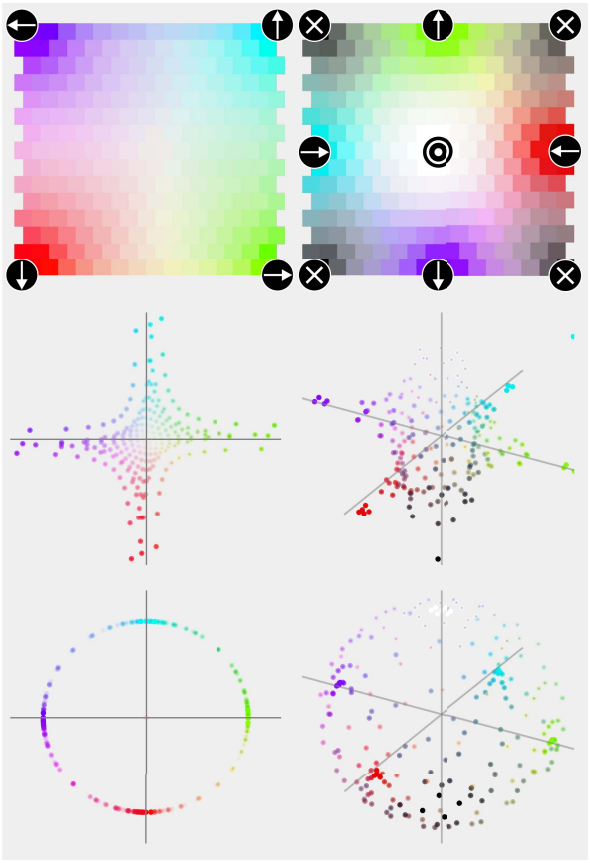


Fig. 8. *Encoding 2-D (left) and 3-D (right) spaces in a 16×16 array of neurons. Top: for 2-D and 3-D, encoding vectors are generated using four and nine tap points (black dots, labeled with their anchor-encoding vector), respectively. For 2-D, vector direction maps to hue. For 3-D, the first dimension maps to luminance (white to black) and the other two to hue. Shorter vectors are more transparent. Middle: resulting vectors are plotted. Bottom: vectors are normalized, showing that they achieve reasonable radial coverage.*

(see Fig. 9). By doing digital fan-out to four or nine tap points and leveraging analog convolution, we achieve performance nearing that of the reference approach that digitally fans out to all 256 neurons.[5] It is not surprising that the diffusor's 2-D metric space efficiently tiles 2-D and 3-D encoders that sit on 1-D and 2-D surfaces and, thus, can be embedded with little distortion of relative distances.[6]

Beyond 3-D, there are no longer clever arrangements of a constant number of tap points that yields good coverage. Achieving good coverage becomes challenging because the encoders can no longer be embedded in the diffusor's 2-D metric space without distorting the relative distances. Furthermore, the number of neurons a single-pool network needs to approximate arbitrary functions of d dimensions is exponential in d [25]. Therefore, we simply make adjacent anchor encoders as orthogonal as possible and increase their number exponentially. (For empirical results, see Section V-B.)

In summary, using the diffusor to implement encoding is a desirable tradeoff. Encoding vectors are typically chosen randomly, so the precise control of the original $\mathbb{R}^{N \times d}$ matrix is not missed. In exchange, we sparsify digital communication in space—and reduce memory footprint—by a factor equal to the number of neurons per tap point.

IV. BUILDING BRAINDROP

Braindrop implements a single core of the planned Brainstorm chip, architected to support building a million-neuron multicore system. As a target application to guide our design, we chose the Spaun brain model [26], which has an average decode dimensionality of eight. Thus, we provisioned the core with 16 8-bit weights per neuron, leaving ample space for transforms. Braindrop's digital logic was designed in a quasidelay-insensitive asynchronous style [27], whereas its analog circuits were designed using subthreshold current-mode techniques [28]. Automated synthesis software supports translating an abstract description of a dynamic nonlinear transformation into its robust implementation on Braindrop's mismatched, temperature-variable analog circuits.

A. Architecture

To maximize utilization of the core's resources, 4096 neurons, 64 KB of weight memory (WM), 1024 accumulator buckets, and 1024 synaptic filters, two small

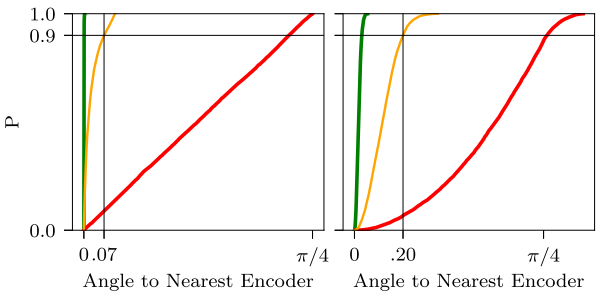


Fig. 9. *Angle-to-nearest-encoder CDFs, evaluated over the surface of the d-sphere (orange curve) for the 2-D (left) and 3-D (right) tap-point-and-diffusor encoders in Fig. 8. The 90th percentiles are 0.07 and 0.20 rad, respectively, implying that 90% of the space has gaps in coverage no bigger than these angles. For reference, CDFs for an equal number (256) of encoding vectors distributed uniformly randomly on the d-sphere's surface (green curve) and for four (2-D) or six (3-D) vectors aligned with the positive- or negative-dimensional axes (red curve) are also shown.*

[5] All distributions are approximated by the Monte Carlo method: generate a large number of random unit vectors ($\max(1000, 100 \cdot 2^d)$) and find each vector's largest innerproduct among the normalized encoding vectors.

[6] In general, a d-D encoder sits on a $(d-1)$-D surface. Thus, the easily embeddable spaces could be extended to 4-D by building a 3-D diffusor, for instance, by stacking the thinned die and interconnecting them using dense thru-silicon vias.

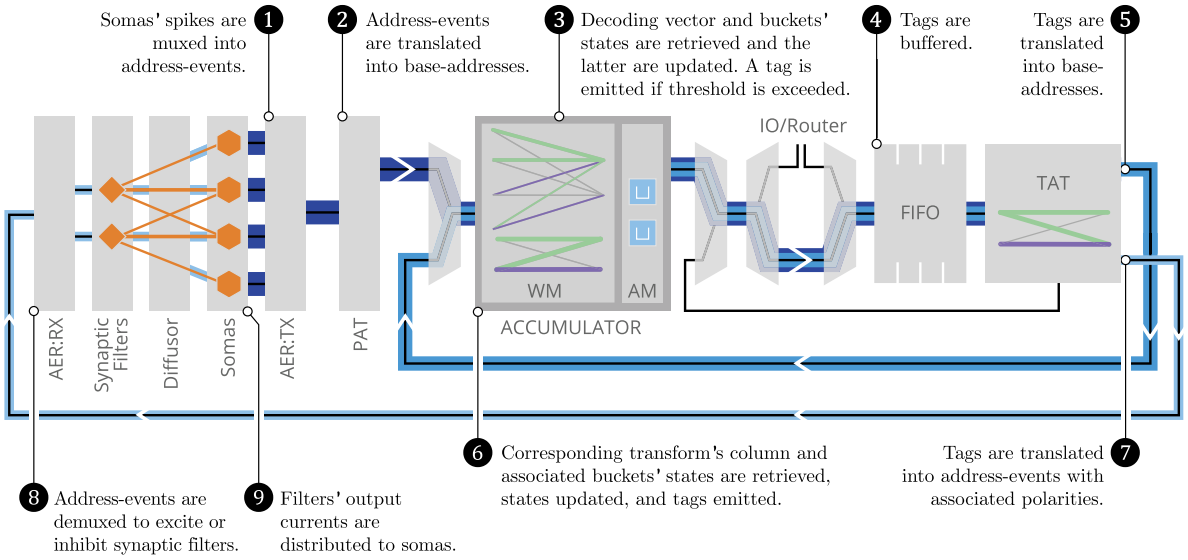


Fig. 10. ***Traffic flow in a decode-transform-encode network (see Fig. 5) mapped onto Braindrop.***

memories, named Pool Action Table (PAT) and Tag Action Table (TAT), provide a level of indirection (see Fig. 10). By programming PAT, indexed by neurons' addresses, the neuron array is flexibly divided up into pools, and each pool's decoders and accumulator buckets are allocated space in WM and Accumulator Memory (AM), respectively. By programming TAT, indexed by a tag, which is emitted by each accumulator when it crosses threshold,[7] arbitrary mappings from accumulators to accumulators (transform) or from accumulators to synaptic filters (sparse encode) are configured.

PAT has 64 entries that enable the 4096-neuron array to be divided into sixty-four 64-neuron subarrays. This division implies a pool-size granularity of 64, slightly more than the NEF rule-of-thumb of 50 neurons per dimension for linear functions. Spikes emitted by somas are multiplexed into an address-event stream by a novel bit-serial Address-Event Representation Transmitter (AER:TX) [29]. These address events identify a neuron using two fields: subarray index and neuron index (within that subarray). The former field indexes PAT, retrieving a base address in WM, corresponding to the first column of D, the subarray's decoding matrix, and another base address in AM, corresponding to the state of the first accumulator bucket. The latter field indexes D's columns, retrieving a decoding vector to add to the buckets.

TAT has 2048 total entries that redirect incoming tags to a (listed) subset of accumulator buckets, synaptic filters, or other cores (by retrieving a stored global route). Each input tag triggers a series of actions on one of these three output types. Retrieved base addresses specify transforms (stored in WM) and their associated bucket states (stored in AM), similar to the PAT entries. Retrieved synaptic filter addresses specify tap points, realizing compact storage of the sparse encode (its zeros do not take up space). This address-event stream is demultiplexed into synaptic-filter-targeting spikes by a novel bit-serial Address-Event Representation Receiver (AER:RX) [29]. Unlike PAT, however, where neurons in the same subarray share a single entry, TAT does not share its entries among multiple tags. This redundancy is acceptable because there are far fewer multidimensional quantities than neurons.

[7]These tags are individually programmable, thereby providing an additional level of indirection.

B. Physical Implementation

Asynchronous digital logic's active power–work rate proportionality is particularly desirable, because Braindrop's digital computation and communication are sparse in time. Such proportionality is difficult to achieve in synchronous design. The implementation challenge is that the asynchronous intellectual property blocks are not available. Subthreshold analog circuits' ability to sparsify digital communication in time (by serving as dynamic computational primitives) and to sparsify digital communication in space (by providing efficient local fan-out) is particularly desirable for maximizing energy efficiency. The implementation challenge is that in the subthreshold region, current signals are exponentially sensitivity to threshold-voltage mismatch between transistors (and to changes in ambient temperature).

To minimize static (i.e., idle) power dissipation and enable deep subthreshold operation, Braindrop is implemented in a 28-nm FDSOI process that supports reverse body bias (RBB). Reverse biasing a transistor's body terminal reduces its leakage current, letting us trade peak throughput for exponentially lower static power dissipation. Unfortunately, the foundry SRAM bit cell, which was

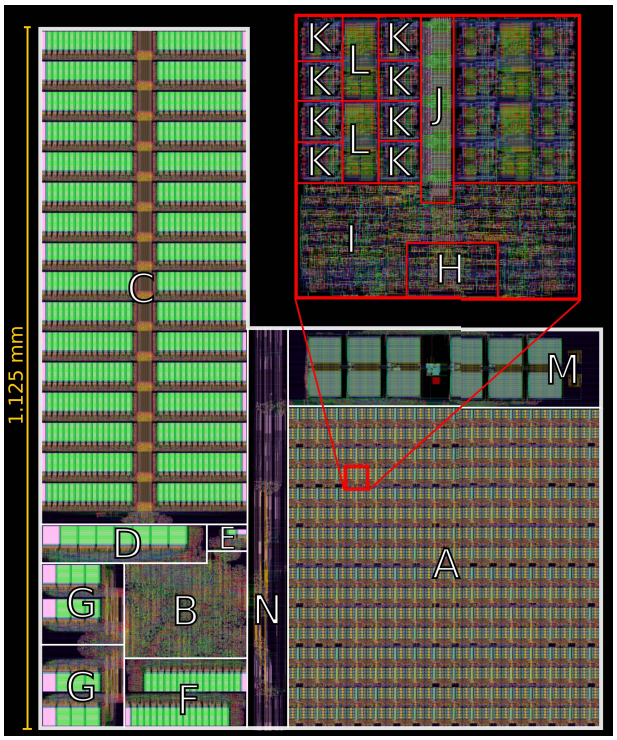


Fig. 11. *Layout of the core. Inset shows the detail of 16-neuron tile (red outline). A: 4096-neuron array, B: digital datapath, C: WM, D: AM, E: PAT, F: FIFO, G: TAT, H: AER tree logic, I: AER leaf logic, J: CM, K: neuron, L: synaptic filter, M: 12 DACs and two ADCs, and N: routing between neuron array, datapath, and IO pads.*

used to implement all the memories [except Configuration Memory (CM); explained in the following] and dominates digital transistor count, does not take advantage of this. RBB also enables the subthreshold analog circuits to operate with mere femtoamps of current. These minuscule currents not only consume negligible power but also reduce the sizes of capacitors, whose area dominates the soma's and synaptic filter's layout.

We minimized the silicon neuron's overall area by optimizing the transistor sizes in concert with a few locally stored programmable bits that adjust its offset and gain. Heterogeneity in the neurons' responses provides a diverse set of basis functions for smooth nonlinear function approximation. For low-order polynomials, neuron spiking thresholds ought to be distributed uniformly across the function's domain [13]. This uniformity is difficult to achieve by relying on the threshold-voltage mismatch alone. With really small transistors, mismatch is excessive, causing many neuron's thresholds to fall outside the domain (i.e., they either always spike or remain silent). However, sizing up the transistors is not the way to go; it reduces area density drastically while unevenly concentrating thresholds in the middle of the domain. Using medium-sized transistors augmented with digital correction to rescue outliers proved optimal.

Equipping each soma with six bits of digital correction effectively reduced silicon area by 38%, compared to relying on mismatch alone [23]. These bits are stored in a 128-bit write-only CM added to the 16-neuron tile (8 bits/neuron). An additional inverter in each SRAM bit-cell drives a switch in the soma circuit, allowing us to choose among seven offsets ($[-3, 3]$ units of a bias current) and four attenuations ($1, 1/2, 1/3, 1/4$), or to kill that soma.[8] We determined the optimal combination of bit count and transistor size using a detailed model of tuning-curve variability, based on an in-house, compact device model validated with SPICE simulations [3]. These correction bits also enabled us to compensate for deviations in the threshold voltage's standard deviation from its nominal value, which proved to be a useful postfabrication.

Table 1 Areas of Braindrop's Major Components

Component	Count	Unit Area (μm^2)	Total Size ($10^3\mu$m^2)	Percent of Total Area		
Neuron	4096	27.5	112.7	13.3	26.0	44.5
Syn. Filter	1024	43.8	44.9	5.3		
DAC	12	5300	63.6	7.5		
AER	256	538	137.8	16.2	18.5	
CM	256	75.1	19.5	2.3		
WM	1	0.637	334.0	39.3	39.3	55.5
AM	1	0.540	21.0	2.5	11.5	
PAT	1	2.344	3.0	0.4		
TAT	2	0.814	45.0	5.3		
FIFO	1	0.661	27.8	3.3		
Datapath	1	N/A	39.9	4.7	4.7	

The core's overall area is dominated by the neuron array, its associated DACs, and WM (see Fig. 11 and Table 1).[9] Other memories were designed to be large enough that their associated resources were unlikely to cause mapping constraints but not so large that they had a large impact on the total system area. Because no async-compatible memory designs are available from the foundry, we designed the entirety of Braindrop's memories (except for the bit cell). Due to time constraints, all of our memories use some standard-cell layout in their peripheries and, thus, have high overhead (peripheral logic takes up about 50% of the area).

C. Software Stack

Automated synthesis software supports translating an abstract description of a computation into its implementation on Braindrop's mismatched, temperature-variable analog circuits. Nengo is a software environment that realizes the NEF [30]. It consists of a frontend, which provides a set of objects (pools, nodes, connections, and so on) that the user's computation is mapped onto, and a back end, which provides a means to execute those objects. Nengo also provides a GUI to make interaction with the frontend more user friendly. The back end interfaces with Braindrop's driver software, which provides

[8]The remaining switches are used to kill synaptic filters, cut the diffusor at pool boundaries, or feed a soma's or synaptic filter's output to the on-chip ADC.

[9]Areas listed include unused space inside bounding boxes. Datapath memory unit areas are reported as area per bit.

its own set of objects, nearly isomorphic to the hardware itself, and also provides methods to communicate with and control an attached Braindrop chip through a field-programmable gate array (FPGA). This software gives access to Braindrop's hardware using Python—the same implementation-agnostic Nengo abstraction used by other back ends [7], [31].

V. VALIDATING BRAINDROP

We now characterize the accumulator's performance, which depends on how nonlinear the function being decoded is, and extend our analysis of encoder coverage beyond 3-D spaces, where the anchor-encoding vectors must increase as 2^d to preserve near-uniform tiling.

A. Accumulative Thinning

With the NEF, computational errors arise from two sources: poor function approximation and spurious spike coincidences. Function approximation is generally improved by increasing the number of basis functions [i.e., allotting more neurons to the pool (see Fig. 3)] and made more difficult as the function becomes more nonlinear (i.e., has more peaks and throughs).[10] Spurious spike coincidences produce fluctuations in the decoded state variable. These fluctuations' relative amplitude decreases as the total spike rate increases and as the synaptic filter's time constant τ_{syn} gets longer.[11] Hence, having more neurons mitigates both effects, improving performance. To decrease an NEF network's error by a factor of 4, neuron count (N) has to increase fourfold to sixteenfold, as function-approximation and spike-coincidence errors drop as $1/N$ and $1/\sqrt{N}$, respectively [13].

When decoding is performed with the accumulator, however, additional considerations come into play. Because of the restriction that $|w_i| \leq 1$, $F_{\max}$, which sets the accumulator output's range of delta rates, must be chosen wisely. For larger $F_{\max}$ (or fewer neurons), more weights will saturate, limiting the usefulness of the associated neurons and also increasing approximation error.[12] Furthermore, although the merged spike trains fed to the accumulator are well-approximated by a Poisson process for low spike rates and equal weights, both assumptions are broken in practice with opposite effects on SNR. On the one hand, individual spike rates are high, which makes the accumulator's input pseudoperiodic, helping SNR. On the other hand, weights have both positive and negative signs, which increases the variance but not the mean (fixed by decode), hurting SNR. (For an in-depth analysis, see Appendix B.)

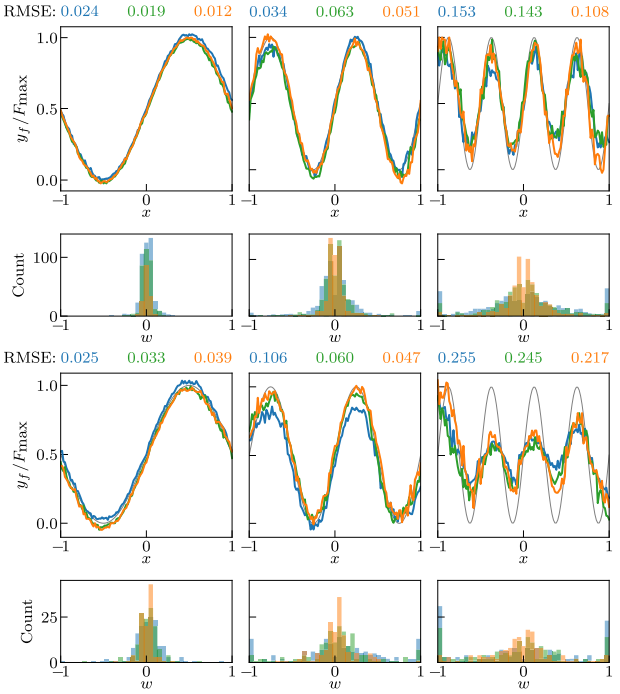


Fig. 12. ***Braindrop function approximations and decode-weight distributions for increasingly higher frequency sinusoidal functions of a single dimension [left to right:*** $y_f(x) = F_{\max}(0.5 + \sin(f\pi x))$***, for*** $f \in \{1, 2, 4\}$***]. Top: 1024 neurons. Bottom: 256 neurons. Decodes are performed for three different*** $F_{max} \in$ ***{500 Hz (orange curve), 1000 Hz (green curve), 1500 Hz (blue curve)}. RMSEs (normalized by*** F_{max}***) are reported (see plot titles). Histograms show weight distributions for the decodes directly above. For the 256 and 1024 pool sizes, respectively, 46% and 42% of the neurons did not fire in the*** $[-1, 1]$ ***range. Thus, these neurons did not contribute much to decode.***

In practice, we found that a fourfold increase in function nonlinearity (measured by number of extrema in $-1 < x < 1$) was only mitigated to some extent by a fourfold increase in neuron count (see Fig. 12). For sinusoids with two extrema, as $F_{\max}$ varied from 500 to 1500 Hz, Braindrop's root-mean-square error (RMSE) varied from 3.9% to 2.5% for 256 neurons and from 1.2% to 2.4% for 1024 neurons (normalized by $F_{\max}$). Whereas for sinusoids with eight extrema, as $F_{\max}$ varied from 500 to 1500 Hz, Braindrop's RMSE varied from 21.7% to 25.5% for 256 neurons and from 10.1% to 15.3% for 1024 neurons. The latter error (10.1%–15.3%) is four times higher than that for one-quarter the number of neurons (256) decoding a function four times simpler (2.5%–3.9%).

B. Sparse Encoding

Anchor-encoding vectors are picked to be orthogonal to their neighbors using a greedy method. The algorithm traverses the tap-point grid from left to right and, then, from top to bottom. It picks an anchor vector in the null-space of a set of one to $\min(4, d-1)$ anchor vectors already assigned to the current location's nearest neighbors. The

[10]Thermal sensitivity also increases for more nonlinear functions because slight perturbations in tuning curves have larger impact; advanced techniques compensate for this [17].

[11]One must also consider the time constant, τ_{dyn}, of the dynamical system being implemented (see Fig. 2): 1) $\tau_{\text{dyn}} > \tau_{\text{syn}}$ requires gain $\tau_{\text{syn}}/\tau_{\text{dyn}} < 1$, which will attenuate the fluctuations; and 2) $\tau_{\text{dyn}} < \tau_{\text{syn}}$ requires gain $\tau_{\text{syn}}/\tau_{\text{dyn}} > 1$, which will amplify them. Hence, if τ_{dyn} is short, pool size must be increased to further reduce (relative) fluctuation amplitude.

[12]This effect could be offset if we could tune the mean spike rate, but no such control was implemented.

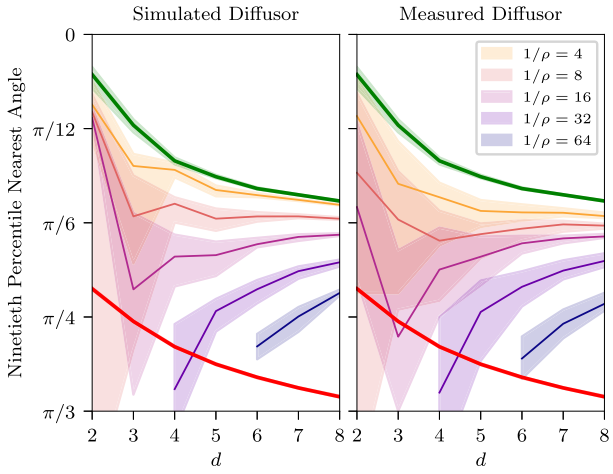


Fig. 13. *Performance of a greedy tap-point algorithm for varying dimensionality (d) and density (ρ). Left: with ideal diffusor kernels, simulated by solving the circuit equations without mismatch. Right: with actual diffusor kernels measured from Braindrop. For each value of ρ, minimum coverage over 90% of the space (as defined in Section III-B) is plotted as a line (median) or shading ([5%,95%] intervals). Performance varies trial-to-trial due to the greedy algorithm's heuristic, random nature. Two limiting cases are also shown: N uniformly randomly distributed (green curve) and $2 \times d$-D-axis aligned or antialigned (red curve) encoding vectors. Savings in digital communication and memory relative to fully specified encoding vectors is $1/\rho$.*

anchors' directions are specified by a $\rho N \times d$ transform matrix, where ρ is the tap-point density. As d, the input space's dimensionality, increases, anchor-encoding vectors must increase as 2^d to preserve near-uniform tiling.

For the results reported here, we increased neuron count as $N = 16 \times 2^d$ (i.e., 16 per quadrant) and swept tap-point density ρ from $1/4$ to $1/64$. Tap points were placed on a regular grid, evenly dividing the neuron array; the diffusor's space constant varied inversely ($\propto \sqrt{1/\rho}$). As the diffusor's kernel decays exponentially, encoding vectors of neurons further away have much smaller norms. These weakly sensitive neurons are much less useful in decoding, as our decode weights have only 8-bit dynamic range. For this reason, we discarded any neuron whose spike rate changed by less than 50 Hz over the input range.

For the moderate values of d studied (2–8), when neuron and tap point's counts increase exponentially, our sparse-encoding scheme trades off modest drops in coverage for linear reductions in communication costs (see Fig. 13). Thus, it is very advantageous to decompose the high-dimensional functions into stages of lower dimensional functions as much as possible to minimize total resources. With this strategy, performance on functions of smaller d is most important.

VI. APPLICATION PERFORMANCE

We previously demonstrated Braindrop's most basic functionality: approximating nonlinear static functions of a single input dimension (see Fig. 12). Its approximation error decreases with larger numbers of neurons or for smoother functions. Here, we explore its capability to approximate static functions of multiple dimensions (2-D-vector rotation) as well as dynamic transformations of order one and three (an integrator and a delay line).

To demonstrate nonlinear static function approximations involving more than one input dimension, we implemented 2-D vector rotation: $x' = x\cos\theta - y\sin\theta$ and $y' = x\sin\theta + y\cos\theta$. Both x' and y' are 3-D functions (of x, y, and θ), but each decomposes along its sum into two 2-D functions (of x and θ or y and θ). Hence, we implemented four 2-D pools with 256 neurons each, merging their decodes to compute the summations; $F_{\max}$ was set to 500 Hz. For angles spanning half a cycle ($\theta \in [-\pi/2, \pi/2]$), these 2-D functions were approximated with error (see Fig. 14) similar to that obtained across a full cycle of a 1-D sinusoid (see Fig. 12, lower left, orange).

To validate Braindrop's ability to model dynamical systems, we implemented an integrator, a basic building block for higher dimensional, nonlinear, dynamical systems, widely used in applications, such as adaptive robot-arm control [32]. The integrator is described by $\tau_{\text{unit}}\dot{x}(t) = u(t)$, where τ_{unit} is the unit of time (e.g., 1 s). Thus, the network's state, $x(t)$, must equal the integral of its input, $u(t)$, scaled by $1/\tau_{\text{unit}}$.

To implement the integrator's dynamics precisely, we applied a theoretical extension to the NEF, which accounts for mismatch between synaptic time

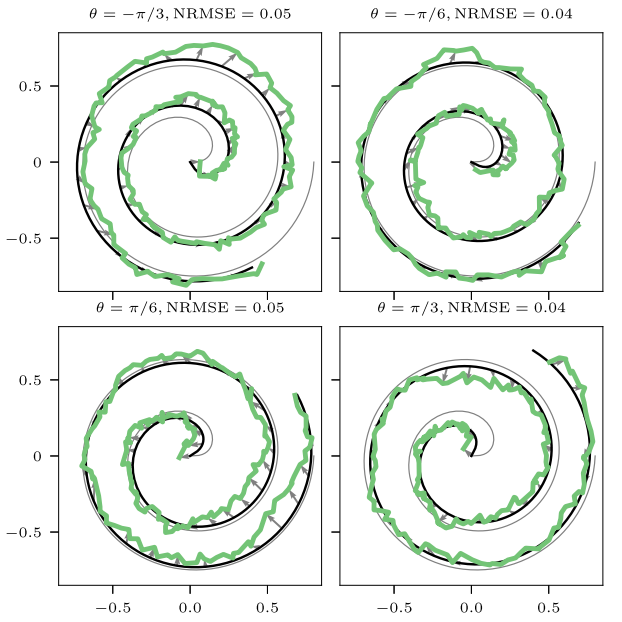


Fig. 14. *Braindrop implements 2-D-vector rotation. For each rotation angle θ (one per panel), x and y are varied in a spiral pattern (thin gray line). Braindrop's output (green line) is ideally a rotated version of the input spiral (black line). Arrows show the error of individual points. Errors [reported as NRMSE=RMSE$((x', y')/F_{\text{out}})$] tend to be made in certain directions (e.g., in $[1, -1, \pi/6]$), suggesting poor neuron sensitivity in that direction.*

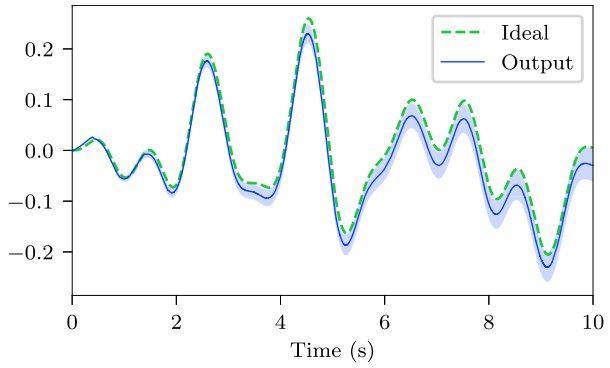


Fig. 15. ***Braindrop implements an integrator. The state is decoded from a 1024-neuron network's spiking activity using a synaptic filter with $\tau = 0.2$ s. We used bootstrapping to visualize the mean and 95% confidence interval from 200 trials. Training and testing signals are independently sampled from the derivative of a 1-Hz band-limited white-noise process.***

constants [19]. First, we characterized each tap point's synaptic filter by applying a step function to its input and recording its closest neurons' responses. Fitting an exponential to these step responses yielded its time constant, τ_{syn_i} ($\mu \pm \sigma$ =179±54 ms). Next, we determined the gain, $\tau_{\text{syn}_i}/\tau_{\text{unit}}$, that effectively drives each tap point with $(\tau_{\text{syn}_i}/\tau_{\text{unit}})u(t) + \hat{x}(t)$ [i.e., setting $f(x) = 0$ in Fig. 2], where $\hat{x}(t) \approx x(t)$ is decoded from the pool's spike trains. In total, we provided recurrent feedback to 1024 neurons through 72 calibrated tap-point filters ($1/\rho = 14.2$). Performance is extremely close to ideal: the ideal target falls within a 95% confidence interval of the network's mean performance, averaged across 200 trials (see Fig. 15).

To provide an example of a more sophisticated dynamical system, we implemented a delay network that buffers a continuous-time rolling window of its input [20]. This network approximates a θ-s delay-line's transfer function, $F(s) = e^{-\theta s}$. We analytically derive state-space matrices, A and B, corresponding to this system's Padé approximant of order $[(q-1)/q]$, for some chosen dimensionality q, and, then, numerically compute its balanced realization. Then, we determine a linear transformation, $C_{\theta'/\theta}$, from the state vector, $\mathbf{x}(t) \in \mathbb{R}^q$, to a delayed version of its input, $c(t-\theta')$.[13] In summary, we have

$$\theta \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + Bc(t)$$
$$c(t-\theta') \approx C_{\theta'/\theta}\mathbf{x}(t), \quad 0 \le \theta' \le \theta$$

where the approximation error is $O((\theta s)^{2q})$.

We have mapped our approximation to a delay line onto Braindrop with $\theta = 0.1$ s and $q = 3$ using three 1-D pools,[14] with 128 neurons each, for a total 384 neurons (see Fig. 16). In this case, we set tap-point density to 1/4 (the maximum) and found performance to be robust. Hence, we did not need to compensate for time constant mismatch. Across the 0.1-s time window, the normalized RMSE is 14.6%.

VII. BRAINDROP's ENERGY EFFICIENCY

Without a clear set of benchmarks, the efficiency of neuromorphic architectures is typically quantified by energy per synaptic operation [7], [33], but what exactly constitutes a synaptic operation differs from one architecture to another. Importantly, the work involved varies depending on how weight matrices are represented (e.g., sparse, low-rank versus dense, and full-rank), whether the network size necessitates intercore communication, and what signal representations are used—physical analog versus integer-valued digital versus unary spikes—each choice yields a different energy–precision scaling [1]. Here, we introduce a metric useful for comparing Braindrop's $N \times d \times N$ decode–encode architecture with the more common $N \times N$ neurosynaptic architecture (i.e., direct all-to-all connectivity).

[13] Alternatively, different linear transformations can be determined to approximate the input's continuous-time convolution with any linear kernel—or integral transform—up to the window's length.

[14] In the NEF, multidimensional pools are only necessary to decode a nonlinear function of the state vector (e.g., as in the Lorenz attractor).

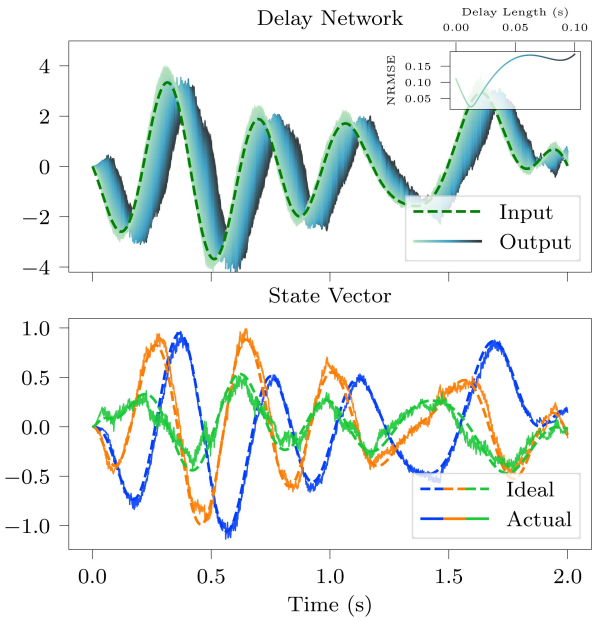


Fig. 16. ***Braindrop implements a delay line. Top: linear transformation of the (3-D) state vector yields the input signal delayed by 0–0.1 s. Across this time window, the normalized RMSE (insert) ranged between 0% and 20%. The test signal is white noise, band limited to 3 Hz. Bottom: state vector, decoded from the 384-neuron network's spiking activity (using a synaptic filter with $\tau = 18.3$ ms) closely matches the numerical solution (obtained with zero-order hold and 1-ms time step).***

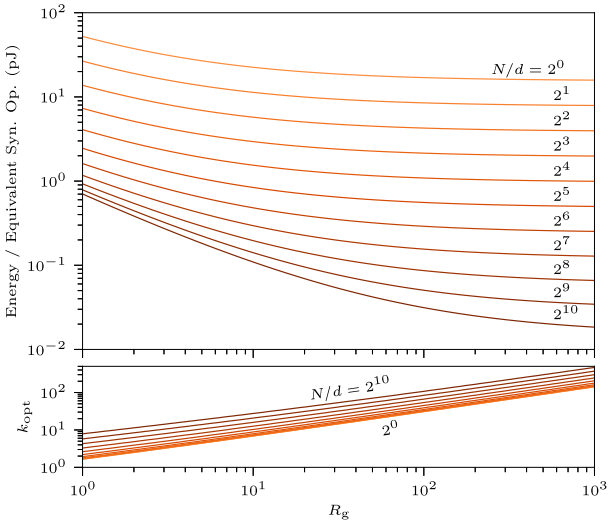


Fig. 17. ***Braindrop's minimum energy per equivalent synaptic operation. Dynamic power consumed by an $N \times d \times N$ decode–encode network is divided by the throughput an $N \times N$ neurosynaptic network requires to achieve the same SNR (R_g). This energy (top) and k's optimal value (bottom) are plotted versus R_g for various numbers of neurons per dimension ($N/d = 2^{\tilde{d}}$). As $N \propto 2^d$, the exponent ($\tilde{d}$) approximately equals the pool's dimensionality (d). Braindrop is most efficient when $d \ll N$ and $R_g \gg 1$.***

To compare SNNs' with different weight-matrix representations, we define the *energy per equivalent synaptic operation* ($E_{\widetilde{\text{op}}}$) metric. An $N \times d \times N$ decode–encode network's $E_{\widetilde{\text{op}}}$ is its dynamic power consumption divided by the throughput (T_{FC}), an $N \times N$ neurosynaptic network requires to achieve the same SNR. This definition relies on the fact that, in general, an $N \times d \times d \times N$ decode-transform-encode network is mathematically identical to an $N \times N$ neurosynaptic network with weights $W = E^{\text{T}} T D$. Here, E's and D's columns specify encoding and decoding vectors, while T specifies a transform. For simplicity, we drop T and calculate $E_{\widetilde{\text{op}}}$ for an $N \times d \times N$ network on Braindrop.

We determine Braindrop's minimum energy per equivalent synaptic operation ($\check{E}_{\widetilde{\text{op}}}$) in four steps. First, we determine the throughputs a given SNR (R_g) observed at the input to each soma in the second N-neuron pool implies in the decoding (T_{d}), FIFO (T_{f}), and encoding (T_{e}) stages. Second, we multiply these throughputs by the decode's, FIFO's, and encode's energies per operation, E_{d}, E_{f}, and E_{e}, respectively, to compute the total power consumed. Third, we optimize k to obtain Braindrop's minimum power consumption, $\check{P}_{\text{BD}}$, for any desired output SNR, R_g. Finally, we divide our expression for $\check{P}_{\text{BD}}$ by T_{FC}, the neurosynaptic network's throughput, to obtain

$$\check{E}_{\widetilde{\text{op}}} \approx \frac{1}{2}\frac{d}{N}\left(1+\sqrt{1+2\left(\frac{K}{R_g}\right)^{2/3}}\right)\left(1+\left(\frac{K}{R_g}\right)^{2/3}\right)E_{\text{d}} \quad (1)$$

where $K = \sqrt{2/3}E_{1/k}/E_{\text{d}}$, with $E_{1/k} = E_{\text{f}} + PE_{\text{e}}$, and P is the number of tap points per dimension. For $R_g \ll K$, $\check{E}_{\widetilde{\text{op}}}$ scales as $R_g^{-2/3}$ because $\check{P}_{\text{BD}}$ scales as $R_g^{4/3}$, while T_{FC} scales as R_g^2. For $R_g \gg K$, $\check{E}_{\widetilde{\text{op}}}$ asymptotes to $1/2E_{\text{d}}/(N/d)$ because $\check{P}_{\text{BD}}$ and T_{FC} both scale as R_g^2. (For a complete derivation, see Appendix C.)

Equation (1) predicts that as the number of neurons per dimension (N/d) increases, Braindrop's minimum energy per equivalent synaptic operation, $\check{E}_{\widetilde{\text{op}}}(R_g)$, reaches a lower and lower asymptote and takes longer and longer to get there. It reaches a lower asymptote because each first-pool soma's spike evokes d weighting operations in the $N \times d \times N$ network versus N in the $N \times N$ one. It takes longer to get there because K is (roughly) proportional to N/d. Recall that $K = \sqrt{2/3}(E_{1/k}/E_{\text{d}})$ and $E_{1/k} = E_{\text{f}} + PE_{\text{e}}$. Hence, for $E_{\text{d}} \approx E_{\text{e}} \approx E_{\text{f}}$, $K \approx \sqrt{2/3}(1+P)$. For constant tap-point density ρ, however, $P = \rho N/d$. Thus, Braindrop's energy per equivalent synaptic operation is highest when $R_g \gg \rho N/d$ and $d \ll N$.

Braindrop's minimum energy per equivalent synaptic operation, $\check{E}_{\widetilde{\text{op}}}$, varied as we expected with R_g and N/d, for tap-point density $\rho = 1/8$ and experimentally determined values of E_{d}, E_{f}, and E_{e} (see Fig. 17). We obtained these values from slopes of measured power consumption versus operating frequency plots for Braindrop's major digital components; these experiments also yielded each component's maximum throughput (see Table 2).[15] As N/d increases, $\check{E}_{\widetilde{\text{op}}}$ indeed reaches lower and lower asymptotes at higher and higher values of R_g. That is, it is lowest when the equivalent synaptic weight matrix's rank is much lower than N and the SNR is much higher than $\rho N/d$. For $\rho = 1/8$, $N/d = 64$, and $R_g = 20$—a typical operating point for NEF networks—the minimum energy per equivalent synaptic operation is $\check{E}_{\widetilde{\text{op}}} = 381$ fJ.

For comparison with non-SNNs, which use physical analog or integer-valued digital signal representations, Braindrop's power per unit bandwidth, $\check{E}_{\text{BD}} = 2\pi\tau\check{P}_{\text{BD}}$, is a more useful metric. Substituting $\check{P}_{\text{BD}}$'s expression (see Appendix C), we have

$$\check{E}_{\text{BD}} \approx \left(1+\sqrt{1+5.7R_g^{-2/3}}\right)\left(0.35+R_g^{-2/3}\right)67dR_g^2 \text{ pJ} \quad (2)$$

[15]Measurements marked as × in Table 2 were infeasible. Static power dissipation was 6 mW—50 times higher than its nominal value. We suspect that this anomaly is due to a problem in the foundry-provided pad frame. Active power and static power for the analog components are negligible, compared to either component of digital power.

Table 2 Component Energy per Operation and Throughput ($V_{\text{DD}} = 1$ V)

Energy per op (pJ)		Throughput (MHz)	Comment
E_{rx}	1.09	18.3	AER:RX
E_{tx}	3.81	27.0	AER:TX
E_{d}	15.1	65.6	TX + PAT + Acc
E_{f}	28.3	×	FIFO
E_{e}	7.55	×	TAT + RX
$\check{E}_{\widetilde{\text{op}}}$	0.381	N/A	$N/d = 64$, $R_g = 20$

Table 3 Comparison With Other Spiking Architectures

	Node (F/nm)	Neurons	Synapses (Max)	F^2 per synapse	pJ per synop
TrueNorth4×4[1]	28	16M	4B	2130	26
Braindrop[2]	28	4K	16M	49.4	0.38
Loihi[3]	14	128K	126M	2440	24

[1]Synapse count is for 1-bit weights that specify whether or not a connection exists. If it does, one of the four 9-bit values—programmed for each neuron—is added to the membrane potential. The incoming axon's type determines which value is chosen.

[2]Synapse count is for equivalent neurosynaptic weight matrix; its rank cannot exceed 16.

[3]Synapse count is for 1-bit weights; synop-energy is from simulations [7].

for $N/d = 64$ and $\rho = 1/8$ (i.e., $P = 8$). For instance, when $R_g = 20$, $\check{E}_{\mathrm{BD}} = 31$ nW/Hz per dimension.

VIII. COMPARISON WITH OTHER SPIKING ARCHITECTURES

Braindrop's encode-transfrom-decode architecture was inspired by Neurogrid, a 1M-neuron, 8B-synapse, mixed-signal, multichip system that fans out digitally communicated spike trains using the analog diffusor [4] and computes weighted sums of their rates using probabilistic thinning [22]. A 6.5 × 7.5-in^2 printed circuit board (PCB) interconnects 16 Neurocore chips, linking their on-chip routers into a tree network that supports multicast routing (one to many) [34]. Each 1.6-cm^2 die, fabricated in a 180-nm CMOS process, has a single core with 65 536 neurons tiled in a 256×256 array. A daughterboard, located at the tree's root, houses 32 Mb of memory (for 8-bit weights) together with an FPGA that performs the probabilistic thinning.[16]

While Braindrop is the only application-specific integrated circuit (ASIC) explicitly designed with the NEF in mind, the NEF has been used to synthesize networks running on Neurogrid [22], [35] as well as on other neuromorphic processors [36]–[38]. We compare Braindrop's synaptic areal density and energy efficiency to that of two contemporary ASICs that realize the neurosynaptic architecture, IBM's TrueNorth and Intel's Loihi, and analyze our empirical observations. We also compare Braindrop's application performance to these ASIC's as well as to SpiNNaker, an older general-purpose computing platform that targets neuromorphic applications.

A. Areal Density and Energy Efficiency

Braindrop's F^2-per-synapse and energy-per-synop (synaptic operation) are 43–49× and 68–63× higher than TrueNorth's–Loihi's, respectively (see Table 3), demonstrating the effectiveness of its convolutional sparse encoder and accumulative-thinning decoder.

IBM's 16M-neuron, 48-synapse, all-digital, TrueNorth4×4 is the neurosynaptic architecture's most aggressively scaled instantiation [8], [33]. A 8.5 × 11.2 in^2 PCB interconnects 16 TrueNorth chips, tiled in a 4 × 4 array. Each 4.2-cm^2 die, fabricated in a 28-nm CMOS process, has 4096 neurosynaptic cores, tiled in a 64 × 64 array. Neighboring cores are linked to form a mesh network that supports point-to-point routing (unicast) and extends seamlessly from chip to chip. Each core has 256 neurons, interconnected by 1-bit synaptic weights hardwired in a 256×410-bit SRAM. That is, its columns and rows correspond to presynaptic axons and postsynaptic dendrites, respectively (crossbar arrangement). In addition to its membrane voltage at the last time step, the extra 154 bits-per-row store multiple programmable parameters for each neuron, including leak, spiking threshold, destination axon, axonal delay, and four 9-bit weight values (sign included) that an in-coming axon can select.

Intel's 128K-neuron, 126M-synapse, all-digital, Loihi implements the neurosynaptic architecture flexibly, including support for variable weight resolution (1–9 bits) and convolutional kernels [7], [39]. Its 60-mm^2 die, fabricated in a 14-nm CMOS process, integrates 128 cores with 1024 neurons each. These cores are interconnected by a two-level 2-D mesh-network—intrachip and interchip—that supports core-to-core unicast spike communication (as well as management and synchronization operations). A spike can be sequentially unicast to a list of destination cores and distributed to a list of neurons within each one (similar to the TAT's functionality). This digital local-fan-out also includes reusable connectivity templates to support regular connections between pools of neurons, such as those in convolutional neural networks (similar to the diffusor's functionality).[17] As Loihi's metrics are extremely similar to TrueNorth's (see Table 3), we focused our analysis on the later, whose architecture has also been described in more detail.

We may better understand TrueNorth's and Braindrop's empirical synaptic areal density and energy efficiency by deriving their surrogates: bits-stored-per-synapse and bits-accessed-per-synop, analytically. Expressions for these surrogates (derived in Appendix D) are summarized in Table 4. We consider accessing data from memory rather than performing computations on it because the former is more energetically expensive. This energy disparity is particularly large for spike-based architectures, where addition replaces multiplication, as unit-area deltas represent spikes.

Our bits-stored-per-synapse and bits-accessed-per-synop surrogates also account for the structural differences between TrueNorth's neurosynaptic network and Braindrop's decode–encode network (see Section VII). Defining

[16]Neurocore itself supports synaptic weights that take on one of four values, determined by which of a neuron's four synaptic filter circuits the spike is delivered to. However, these four values are not programmable at the neuronal level; they are common to all of a Neurocore's neurons.

[17]The architecture provides traces of the synapses' prespike and postspike activity to three ×86-based microprocessors (in the same package), which can adapt synaptic weights according to a programmable learning rule.

Table 4 Neurosynaptic Versus Decode–Encode Architecture

	TrueNorth 256×256	Braindrop $256\times4\times32$
SRAM (bits/synapse)	B 1.60	$(d/N)(B_{\mathrm{D}}+\rho B_{\mathrm{T}}+B_{\mathrm{A+F}}/N)$ 0.158
Reads (bits/synop)	$Bf_{\mathrm{cyc}}/f_{\mathrm{spk}}$ 80	$\frac{1}{2}\frac{d}{N}\left(1+\sqrt{1+2\left(\frac{K}{R_g}\right)^{2/3}}\right)\cdot\left(1+\left(\frac{K}{R_g}\right)^{2/3}\right)B_{\mathrm{D+A}}$, $K=\sqrt{\frac{2}{3}\frac{3B_{\mathrm{F}}+PB_{\mathrm{T}}}{B_{\mathrm{D+A}}}}$ 1.04

a synapse as a weighted connection between two neurons only makes sense for neurosynaptic architectures, which connect somas directly together. Decode–encode architectures connect decoder outputs to encoder inputs, an indirect way to connect somas. Reducing this intermediate stage's width (equal to d) cuts bits stored and accessed but limits the equivalent synaptic weight matrix's rank (normally equal to N). Thus, while TrueNorth's core has rank 256 but uses 1-bit weights, Braindrop's core has rank 16 but uses 8-bit weights. Our surrogates account for this difference by normalizing the decode–encode network's bits-stored and bits-accessed by the neurosynaptic network's synapse or synop count, respectively, yielding expressions that depend on the equivalent neurosynaptic weight matrix's relative rank (d/N).

For relative rank $d/N = 1/64$ and $1/256$, and tap-point density $\rho = 1/8$ and $1/4$, respectively, we predict that Braindrop's bits-stored-per-synapse is 11 and 35 times (see Table 4 and Appendix D) less than TrueNorth's (despite Braindrop's much more precise weights). The latter result matches their empirically measured F^2-per-synapse's 43:1 ratio [$2130F^2/49.4F^2$ (see Table 3)], confirming our hypothesis that memory-use drives synaptic density.

For relative rank $d/N = 1/64$ and tap-point density $\rho = 1/8$, we predict that Braindrop's bits-accessed-per-synop is 76 times less than TrueNorth's (see Table 4 and Appendix D). This prediction, which takes into account TrueNorth's clocked operation and Braindrop's event-driven operation, matches their empirically measured pJ-per-synop's 68:1 ratio (26/0.38 pJ) (see Table 3), confirming our hypothesis that memory-accesses drive energy costs.

B. Performance on Applications

NEF networks with as many as 0.5M neurons have ran in real time on three SpiNNaker boards (personal communication), each of whose 48 interconnected chips has 18 Advanced RISC Machines processor cores, integrated on a 1-cm^2 die fabricated in a 130-nm CMOS process. To realize an efficient mapping, its neurosynaptic network was converted into a decode–encode network by adapting its spike-communication packets' optional payload (32 bits) to carry a decoded variable (i.e., d spike packets per d-D state vector) [40]. The resulting weight-matrix compression enabled a processor core's 64 KB of on-chip memory to fit networks with two to three times more neurons as its 8 MB of off-chip memory fit (sans compression). In comparison, the neuronal density of Braindrop—whose 4096 neurons in 0.65-mm^2 scale to 0.63M neurons in 1 cm^2—is 60.5 times higher, or 2.8 times higher if SpiNNaker was refabricated in the same 28-nm process.

More recently, the NEF has been used to synthesize networks running on TrueNorth and Loihi. In initial attempts, TrueNorth's 1M neurons could only implement a 629-neuron NEF network (representing five dimensions) [41]. Recent work from the same group increased the 5-D NEF-network's size to 11 947—88 TrueNorth neurons per NEF neuron (personal communication). That many TrueNorth neurons aggregate enough of its 1-bit synaptic weights to achieve 9-bit resolution (sign included) for 5-D decoding and encoding weights, which require 90 fully programmable bits.[18] This mapping only applies to the static transformations—unlike the dynamic tasks we demonstrated here—because TrueNorth's digital solution does not approximate exponentially decaying synaptic filters well.

As to applying the NEF on Loihi, it does support a discretized approximation of the exponentially decaying synaptic filters that the NEF leverages to perform dynamic transformations, its key distinguishing feature. Since the project to synthesize networks on Loihi using the NEF is ongoing,[19] it remains to be seen how well Loihi's particular digitized approximation can be leveraged to perform the dynamic transformations we have demonstrated here with Braindrop.

IX. CONCLUSION

Braindrop unites analog efficiency and digital programmability, thereby supporting an NEF-based synthesis procedure for mapping high-level abstractions to subthreshold-analog and asynchronous-digital neuromorphic chips. Achieving this goal required codesigning all layers of the software–hardware stack, keeping the theoretical framework in mind even at the lowest levels of the hardware design. This painstaking process has resulted in a new computational platform that marries hardware embodying the brain's microarchitectural techniques with an accessible programming framework.

Realizing analog circuits' efficiency was only possible through optimizing the NEF's encoding and decoding operations to minimize digital communication without violating the abstractions they present to the user. Braindrop realizes two such optimizations: sparse encoding and accumulative thinning. Together, they allow for a massive reduction in digital traffic when the equivalent synaptic weight matrix has low rank and the desired SNR is high. Thus, these optimizations greatly enhance Braindrop's

[18]Decoding weights must have $1/2\log_2(N/d)$ more bits than synaptic weights to accommodate the $\sqrt{(N/d)}$ improvement in SNR that summing N/d neurons' weighted spike rates provides.

[19]https://www.nengo.ai/nengo-loihi/

energy efficiency while remaining nominally invisible to the user.

Braindrop's hardware was architected with user transparency in mind, minimizing the possibility that inflexibility in allocating its limited resources constrains application performance. In exchange for some small up-front area costs, we were able to ensure high utilization of the area-dominant WM, analog somas, and synaptic filters. The application results demonstrate synthesized networks running on the hardware, realizing the project's goals.

APPENDIX A: Accumulator With Equal Weights and Low Spike Rates

To formalize our intuition that the accumulator performs the desired weighting operation while yielding improved point-process statistics, we consider the case of a single accumulator with a single Poisson input. This is equivalent to the case where the accumulator performs a decode from many neurons, each with the same weight. Given a constant input, individual neurons fire periodically, rather than with Poisson statistics. However, as long as the synaptic filter's time constant (τ) is less than one-sixth of each neuron's period, the interdelta intervals (IDIs) of their merged delta trains will be independently distributed within the scope of the filter window [42]. Poisson statistics apply under these conditions.

We may quantify the randomness of the accumulator's output delta train by calculating its IDIs' coefficient of variation [$\mathrm{CV}(Y)$], a measure that varies from 0 for a periodic process to one for a Poisson process. Given uniform weights $w = 1/k$ for each incoming delta and IDIs X_i, the accumulator's IDIs are $Y_j = \sum_{i=kj}^{k(j+1)-1} X_i$. As $X_i \sim \text{Exponential}(\lambda)$, we have $Y_j \sim \text{Gamma}(\lambda/k, k)$. Y's coefficient of variation is therefore

$$\mathrm{CV}(Y) = \frac{\sigma_Y}{E[Y]} = \frac{\sqrt{k\mathrm{var}(X)}}{kE[X]} = \frac{\sqrt{k/\lambda^2}}{k/\lambda} = \frac{1}{\sqrt{k}} = \sqrt{w}.$$

$\mathrm{CV}(Y)$ tends to 0 as the weight (w) becomes smaller and the delta count (k) required to cross threshold becomes larger, suggesting that the accumulator output's IDIs transition smoothly from Poisson to periodic.

As a result of transitioning from a Poisson process' quadratic scaling to a periodic process' linear scaling, an accumulator produces much fewer deltas than merging or Bernoulli trials for the same SNR (see Fig. 18). It can be shown that the filtered accumulator output's SNR, R_g, is related to its filtered Poisson input's SNR, R_p, by

$$R_g^2 = R_p^2 \,/\, (1 + k^2/3R_p^2) \tag{3}$$

assuming $k^2 \gg 1$ and $\lambda\tau \gg 1$ [42]. Hence, k may be increased until it is close to R_p, which drops R_g close to $R_p/(1+1/3)^{1/2}$. Thus, the accumulator can thin the Poisson process' rate, F_{in}, by its SNR, $R_p = (2\tau F_{in})^{1/2}$, while degrading the SNR by only a factor of $\sqrt{4/3}$. As a result, if $d < R_p$, traffic through a $N \times d$ decode gets multiplied by $d/R_p < 1$. Increasing k beyond R_p, however, will degrade R_g. This degradation sets in when the accumulator's output-delta rate, F_{out}, approaches the rate of a periodic process with SNR equal to R_p.

In summary, for equal weights and low spike rates, the accumulator produces a unit-area-delta train with statistics that transition from Poisson to periodic as the weight, w, decreases. When $w = 1$—with the accumulator acting as a pass-through—its SNR matches the input Poisson process. When $w < 1$, its SNR still matches the input's, but it uses fewer deltas. When $w \ll 1$, its SNR approaches a periodic process. In contrast, the Bernoulli weighting only produces the Poisson statistics, always yielding lower SNR for the same output rate.

However, the assumptions that neurons' spikes are equally weighted and their merge is Poisson are often broken in practice. First, and most importantly, decoding weights not only vary in magnitude but also in sign, which degrades SNR due to a greater summation of variance without a corresponding summation of magnitude. Second, the Poisson distribution applies only if individual neurons spike less than once every 6τ s, where τ is the synaptic time constant [42]. If they do not, the accumulator input has better-than-Poisson statistics, leading to improved SNR. Thus, breaking these two assumptions has opposite effects on SNR.

APPENDIX B: Accumulator With Realistic Weights and Spike Rates

Motivated by practical considerations, we derive limiting expressions for the accumulator's output SNR and compare them with its SNR for actual weight and spike rate distributions. These distributions are from the 1024-neuron pool trained to decode 1-D sinusoidal functions of varying frequency (see Fig. 12). The weight distribution—centered around 0—broadens as the approximated function's extrema count increases. Its positively and negatively signed weights cause SNR to be limited by the rules governing subtracting random variables. We find that our limiting expressions bound the SNR's degradation so long as the accumulator's output SNR is not limited by the degree of thinning (i.e., its output spike train is not nearly periodic).

To see why the weight-distribution's width limits SNR, consider a sum of N random variables X_i with weights w_i. This sum is akin to performing a deterministic weighting on the neurons' spike trains, a limiting case on the accumulator's performance. For $Y = \sum_i w_i X_i$, the SNR is $\mu_Y/\sigma_Y = (\sum_i w_i\mu_i)/(\sum(w_i\sigma_i)^2)^{1/2}$, where μ_i and σ_i are the ith variable's mean and standard deviation. We fix μ_Y when we solve for our decoders, but σ_Y can become arbitrarily large as w_i's distribution broadens.

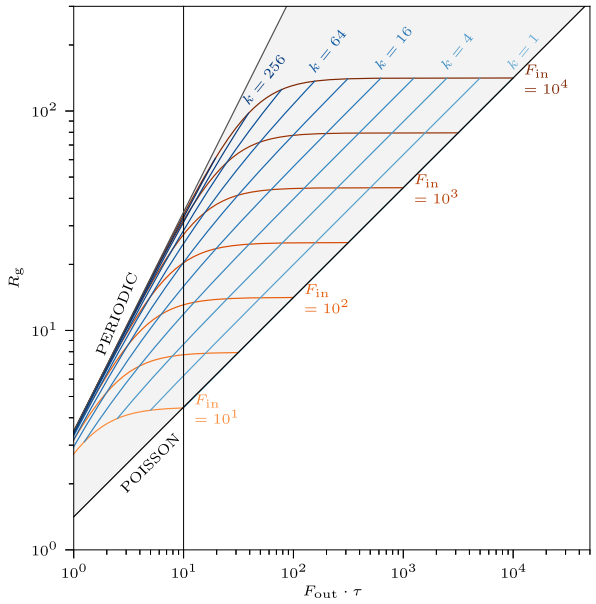


Fig. 18. *Relationship between accumulator's SNR (R_g) and its output-delta rate (F_{out}). For a given Poisson input rate, F_{in} (orange to red curves), as k increases (cyan to blue curves), F_{out} drops while R_g remains constant, until a particular value of k is reached. Feasible combinations of F_{out} and R_g (gray region) correspond to point-process statistics transitioning from Poisson (square-root scaling) to periodic (linear scaling). When the accumulator is producing less than 10 spikes every τ s (left region), the jitter its initial state adds is nonnegligible.*

We can use the above-mentioned formula to compute the SNR expressions for periodic (upper bound) and Poisson (lower bound) point processes. SNR will fall below

$$R_{\mathrm{UB}} = \left(\sum_i w_i \lambda_i\right) \Big/ \sqrt{\sum_i w_i^2 \lambda_i^2 \left(\frac{1}{2\lambda_i \tau} \coth\left(\frac{1}{2\lambda_i \tau}\right) - 1\right)} \tag{4}$$

computed for filtered periodic spike trains with rates λ_i and above

$$R_{\mathrm{LB}} = \left(\sum_i w_i \lambda_i\right) \Big/ \sqrt{\sum_i w_i^2 \lambda_i / 2\tau} \tag{5}$$

computed for Poisson spike trains with the same rates [42]. For $\lambda_i \gg 1/\tau$, the SNR is probably closer to R_{UB}; for $\lambda_i < 1/6\tau$, it is closer to R_{LB}. In effect, these results tell us that for mixed-sign weight distributions, we need to shift the orange iso-F_{in} $R_g(F_{\mathrm{out}}, F_{\mathrm{in}})$ curves' flat region down (see Fig. 18).

We compared the accumulator's measured output SNR for the 1024-neuron-pools' decodes [$y_f(x)$ for $x = 0$] to our bounds (R_{UB} and R_{LB}) and to values the simple analysis predicted [see (3)]. We observe the accumulator output over time at $y_f(0)$ for the three different fs and the three different F_{out}s. In all nine cases, its SNR falls within our bounds. It only comes close to the simple analysis' prediction for $f = 1$ (see Fig. 19). This f has the narrowest distribution of (oppositely signed) weights, which produces the smallest drop in SNR. This small drop is counterbalanced by the increase in SNR produced by the pseudoperiodic point processes.[20] For higher fs, the broader weight distributions increase summed variance further, decreasing SNR further.

APPENDIX C: Energy per Equivalent Synaptic Operation

To determine Braindrop's minimum energy per equivalent synaptic operation ($\breve{E}_{\widetilde{\mathrm{op}}}$), we first determine the throughputs implied by the given SNR (R_g) in its decoding (T_d), FIFO (T_f), and encoding (T_e) stages. For standard orthonormal-basis anchor-encoding vectors, each tap point receives the output of a single accumulator (with a positive or negative sign). To ensure this output has SNR R_g, we must feed each accumulator a point process—assumed to be Poisson—with appropriate SNR $R_p = (2\tau F_{\mathrm{ac}})^{1/2}$, determined by this process' rate, F_{ac}, and the synaptic filter's time constant, τ. Therefore, for d accumulators, the total decoding throughput is $T_d = dF_{\mathrm{ac}} = dR_p^2/(2\tau)$. The d accumulators thin T_d by k, offering throughput $T_f = dR_p^2/(2\tau k)$ to the FIFO. Each stream fans out to

[20]The pool's neurons spiked at hundreds of hertz, whereas $\tau = 100$ ms.

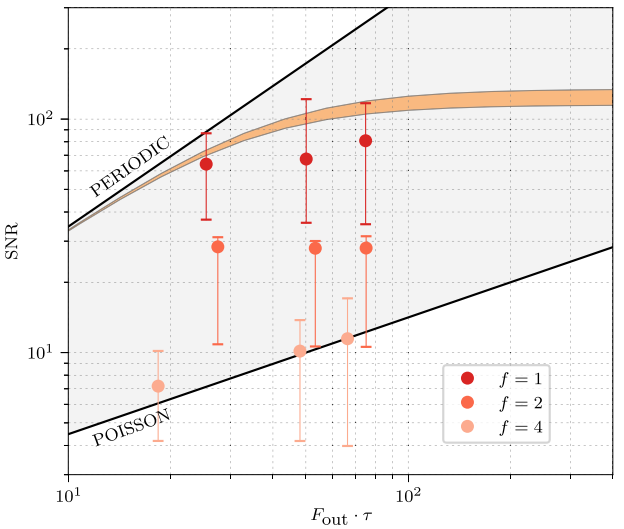


Fig. 19. *Given a broad weight distribution, the accumulator's SNR is lower for the same F_{out}. SNR and F_{out} measurements [taken at $y_f(0)$] are plotted as dots for the 1024-neuron networks in Fig. 12. The idealized SNR-versus-F_{out} relationship (in Fig. 18) is plotted as an orange band. F_{in} is computed as the sum of spike rates of neurons with nonzero weights; it varies slightly depending on results of the decoding-weight optimization. The range shown around each dot spans between R_{UB} and R_{LB} (whiskers), computed using the neurons' measured spike rates at $x = 0$.*

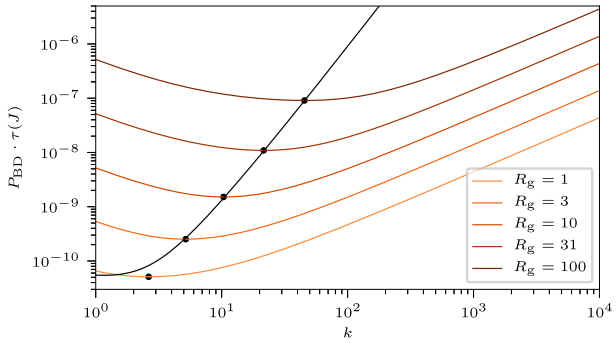


Fig. 20. ***Minimizing dynamic power per unit bandwidth ($1/\tau$) for a desired output SNR (R_g) by optimizing number (k) of deltas required to trip the accumulator [see (6)]. As k increases, energy decreases, reaches its minimum (dots), and, then, rises steeply. These different slopes arise, respectively, from flat and sloped segments of the iso-F_{in} $R_g(F_{out}, F_{in})$ curves (see Fig. 18). The minimum's predicted locus (black line) is also plotted [see (7)].***

P tap points, giving total encoding throughput $T_e = dPR_p^2/(2\tau k)$.

Next, we multiply these throughputs by the decode's, FIFO's, and encode's energies-per-operation, E_d, E_f, and E_e, respectively, to compute the total power consumed. Thus, we obtain

$$\begin{aligned} P_{BD} &= E_d T_d + E_f T_f + E_e T_e \\ &= \frac{dR_p^2}{2\tau}\left(E_d + \frac{1}{k}E_f + \frac{P}{k}E_e\right) \\ &= \frac{dR_g^2}{4\tau}\left(1 + \sqrt{1 + \frac{4k^2}{3R_g^2}}\right)\left(E_d + \frac{1}{k}E_{1/k}\right) \end{aligned} \quad (6)$$

after inverting (3) to express R_p in terms of R_g and k and substituting $E_{1/k} = E_f + PE_e$.

Then, we optimize k to minimize Braindrop's power consumption for any desired output SNR, R_g (see Fig. 20). To find its optimal value, $\check{k}$, we differentiate (6) with respect to k, set the result to 0, and solve for

$$\check{k} \approx (3/2)^{1/3}(E_{1/k}/E_d)^{1/3}R_g^{2/3} = \sqrt{3/2}K^{1/3}R_g^{2/3} \quad (7)$$

defining $K = \sqrt{2/3}E_{1/k}/E_d$ and assuming $R_g \gg K$.[21] Substituting this expression back into (6) yields

$$\check{P}_{BD} \approx \frac{d}{4\tau}\left(1 + \sqrt{1 + 2\left(\frac{K}{R_g}\right)^{2/3}}\right)\left(R_g^2 + K^{2/3}R_g^{4/3}\right)E_d. \quad (8)$$

As the R_g^2 term exceeds the $R_g^{4/3}$ term when $R_g > K$, Braindrop's power consumption scales quadratically with SNR for $R_g \gg K$.

Finally, we divide our expression for $\check{P}_{BD}$ [see (8)] by T_{FC}, the neurosynaptic network's throughput, to obtain $\check{E}_{\widetilde{op}}$. Each soma in the neurosynaptic network's second N-neuron pool receives a point process—also assumed to be Poisson—whose rate F_{so} determines its SNR: $R_g = \sqrt{2\tau F_{so}}$, after low-pass filtering with time constant τ. Thus, $T_{FC} = NF_{so} = NR_g^2/(2\tau)$, dividing by this yields (1).

APPENDIX D: Synaptic Density and Energy-Efficiency Comparison

To calculate bits-stored-per-synapse for TrueNorth's 256×256 neurosynaptic network, we take into account the 256×1-bit synaptic weights as well as the 154-bit parameter-field appended to each of the weight matrix's rows. Dividing the total (410) by the number of synapses per neuron (256) yields its table entry (1.6 bits/synapse).

To calculate bits-stored-per-synapse for Braindrop, we study a 256×4×32 network configuration ($N/d = 64$ and $\rho = 1/8$) to match TrueNorth's core size as well as a 4096×16×1024 network configuration (the largest possible) to predict Braindrop's relative F^2-per-synapse (see Table 3). We assign the following memory-word sizes: WM's is $B_D = 8$ bits (decoding weight), AM's is $B_A = 38$ bits (15-bit state, 3-bit threshold, 19-bit global tag, and stop bit), FIFO's is $B_F = 20$ bits (11-bit local tag, 8-bit count, and dirty bit), and TAT's is $B_T = 15$ bits (tap-point address). The total memory used is, thus, $NdB_D + dB_{A+F} + \rho NdB_T$, where $B_{A+F} = 58$ bits. Dividing by $N^2 = 2^{16}$, the equivalent number of synapses, yields its table entry (0.16 bits/synapse), which scales as d/N, the synaptic weight matrix's relative rank. For the 4096×16×1024 network configuration, the table's formula yields 0.046 bits/synapse—with $\rho = 1/4$ instead of 1/8—a 3.4× drop that reflects d/N's fourfold decrease.

To calculate bits-accessed-per-synop for TrueNorth's 256×256 core, we take its clock-driven operation into account. A controller cycles through the SRAM's 256 rows every millisecond (410-Kb/s access rate), reading their contents. Dividing by the rate at which synapses are activated—the reported typical mean spike rate ($f_{spk} = 20$ Hz) times the number of neurons targeted by each spike (256)—yields the table entry (80 bits/synop).

To calculate the bits-accessed-per-synop for Braindrop, we rederive (1), replacing empirical energy measurements with the number of bits that each component accesses per operation. $B_{D+A} = 46$ bits is substituted for E_d and $3B_F + PB_T = 60 + 15P$ for $E_{1/k}$,[22] where $P = \rho N/d$. As stated before, we set $R_g = 20$, $N/d = 64$, and $\rho = 1/8$, which yields the table entry (1.04 bits/synop). ■

[21] Notice that if $\check{k} > d \Leftrightarrow R_g > (2/3)^{3/4}\, d^{3/2}/\sqrt{K}$, traffic drops through the decode's matrix-multiply instead of increasing.

[22] The factor of three accounts for FIFO's three read/write operations per entry.

REFERENCES

[1] K. Boahen, "A neuromorph's prospectus," *Comput. Sci. Eng.*, vol. 19, no. 2, pp. 14–28, 2017.

[2] A. Pavasović, A. G. Andreou, and C. R. Westgate, "Characterization of subthreshold MOS mismatch in transistors for VLSI systems," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 8, no. 1, pp. 75–85, 1994, doi: 10.1007/BF02407112.

[3] B. V. Benjamin and K. Boahen, "Mismatch and temperature aware compact MOS model for subthreshold circuit design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, to be published.

[4] 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.

[5] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 106–122, Feb. 2018.

[6] J. Park, S. Ha, T. Yu, E. Neftci, and G. Cauwenberghs, "A 65 k-neuron 73-mevents/s 22-pj/event asynchronous micro-pipelined integrate-and-fire array transceiver," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2014, pp. 675–678.

[7] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018.

[8] A. S. Cassidy *et al.*, "Real-time scalable cortical computing at 46 Giga-synaptic OPS/Watt with $\sim$100$\times$ speedup in time-to-solution and $\sim$100,000$\times$ reduction in energy-to-solution," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2014, pp. 27–38.

[9] W. J. Karplus, *Analog Simulation: Solution of Field Problems*. New York, NY, USA: McGraw-Hill, 1958.

[10] Y. Huang, N. Guo, M. Seok, Y. Tsividis, and S. Sethumadhavan, "Analog computing in a modern context: A linear algebra accelerator case study," *IEEE Micro*, vol. 37, no. 3, pp. 30–38, Jun. 2017.

[11] S. S. Woo, J. Kim, and R. Sarpeshkar, "A digitally programmable cytomorphic chip for simulation of arbitrary biochemical reaction networks," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 2, pp. 360–378, Apr. 2018.

[12] J. Schemmel, D. Briiderle, A. Griibl, M. Hock, K. Meier, and S. Millner, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *Proc. IEEE Int. Symp. Circuits Syst.*, May/Jun. 2010, pp. 1947–1950.

[13] C. Eliasmith and C. H. Anderson, *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems*. Cambridge, MA, USA: MIT Press, 2003.

[14] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 2, Jul. 2004, pp. 985–990.

[15] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[16] E. Yao and A. Basu, "VLSI extreme learning machine: A design space exploration," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 1, pp. 60–74, Jan. 2017.

[17] E. Kauderer-Abrams, A. Gilbert, A. Voelker, B. Benjamin, T. C. Stewart, and K. Boahen, "A population-level approach to temperature robustness in neuromorphic systems," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.

[18] S. Reid, A. Montoya, and K. Boahen, "PINT: Polynomial in temperature decode weights in a neuromorphic architecture," in *Proc. Design, Autom. Test Eur. Conf. (DATE)*, Mar. 2019.

[19] A. R. Voelker, B. V. Benjamin, T. C. Stewart, K. Boahen, and C. Eliasmith, "Extending the neural engineering framework for nonideal silicon synapses," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.

[20] A. R. Voelker and C. Eliasmith, "Improving spiking dynamical networks: Accurate delays, higher-order synapses, and time cells," *Neural Comput.*, vol. 30, no. 3, pp. 569–609, 2017.

[21] D. H. Goldberg, G. Cauwenberghs, and A. G. Andreou, "Probabilistic synaptic weighting in a reconfigurable network of VLSI integrate-and-fire neurons," *Neural Netw.*, vol. 14, nos. 6–7, pp. 781–793, Jul. 2001.

[22] S. Choudhary *et al.*, "Silicon neurons that compute," in *Artificial Neural Networks and Machine Learning—ICANN*, A. E. P. Villa, W. Duch, P. Érdi, F. Masulli, and G. Palm, Eds. Berlin, Germany: Springer, 2012, pp. 121–128.

[23] A. Neckar, T. Stewart, B. Benjamin, and K. Boahen, "Optimizing an analog neuron circuit design for nonlinear function approximation," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5.

[24] D. I. Feinstein, "The hexagonal resistive network and the circular approximation," California Inst. Technol., Tech. Rep. CS-TR-88-07, 1988. [Online]. Available: http://resolver.caltech.edu/CaltechCSTR:1988.cs-tr-88-07

[25] A. R. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 930–945, May 1993.

[26] C. Eliasmith *et al.*, "A large-scale model of the functioning brain," *Science*, vol. 338, no. 6111, pp. 1202–1205, 2012.

[27] A. J. Martin, "Synthesis of asynchronous VLSI circuits," California Inst. Technol., Tech. Rep. CS-TR-93-28, 1993. [Online]. Available: http://resolver.caltech.edu/CaltechCSTR:1988.cs-tr-93-28

[28] A. G. Andreou and K. A. Boahen, "Translinear circuits in subthreshold MOS," *Analog Integr. Circuits Signal Process.*, vol. 9, no. 2, pp. 141–166, 1996.

[29] S. Fok and K. Boahen, "A serial H-tree router for two-dimensional arrays," in *Proc. 24th IEEE Int. Symp. Asynchronous Circuits Syst.*, 2018, pp. 1–8.

[30] T. Bekolay *et al.*, "Nengo: A Python tool for building large-scale functional brain models," *Frontiers Neuroinf.*, vol. 7, no. 48, pp. 1–13, 2014.

[31] 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.

[32] T. DeWolf, T. C. Stewart, J.-J. Slotine, and C. Eliasmith, "A spiking neural model of adaptive arm control," *Proc. Roy. Soc. B, Biol. Sci.*, vol. 283, no. 1843, pp. 1–9, 2016. [Online]. Available: http://rspb.royalsocietypublishing.org/content/283/1843/20162134.abstract

[33] 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.

[34] P. Merolla, J. Arthur, R. Alvarez, J.-M. Bussat, and K. Boahen, "A multicast tree router for multichip neuromorphic systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 3, pp. 820–833, Mar. 2014.

[35] S. Menon, S. Fok, A. Neckar, O. Khatib, and K. Boahen, "Controlling articulated robots in task-space with spiking silicon neurons," in *Proc. 5th IEEE RAS/EMBS Int. Conf. Biomed. Robot. Biomechatronics*, Aug. 2014, pp. 181–186.

[36] F. Galluppi, S. Davies, S. Furber, T. Stewart, and C. Eliasmith, "Real time on-chip implementation of dynamical systems with spiking neurons," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2012, pp. 1–8.

[37] F. Corradi, C. Eliasmith, and G. Indiveri, "Mapping arbitrary mathematical functions and dynamical systems to neuromorphic VLSI circuits for spike-based neural computation," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 2014, pp. 269–272.

[38] R. Wang, T. J. Hamilton, J. Tapson, and A. van Schaik, "A compact neural core for digital implementation of the Neural Engineering Framework," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2014, pp. 548–551.

[39] C. Lin *et al.*, "Programming spiking neural networks on Intel's Loihi," *Computer*, vol. 51, no. 3, pp. 52–61, 2018.

[40] A. Mundy, J. Knight, T. C. Stewart, and S. Furber, "An efficient SpiNNaker implementation of the neural engineering framework," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.

[41] K. D. Fischl, T. C. Stewart, K. L. Fair, and A. G. Andreou, "Implementation of the neural engineering framework on the TrueNorth neurosynaptic system," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, Oct. 2018, pp. 587–590.

[42] S. Fok, A. Neckar, and K. Boahen, "Weighting and summing spike trains by accumulative thinning," in *Proc. Design Autom. Conf.*, submitted for publication.

ABOUT THE AUTHORS

Alexander Neckar received the B.S. degrees in electrical engineering and in computer engineering from Northwestern University, Evanston, IL, USA, in 2010, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 2012 and 2018.

He was a member of the Brains in Silicon Laboratory, Stanford University, where he was involved in developing Braindrop's system architecture and designing and implementing its digital datapath. He is currently with a startup, where he is involved in researching and developing neuromorphic circuits for ultralow-power sensor processing.

Sam Fok received the B.S. degrees in electrical engineering and in biomedical engineering from Washington University in St. Louis, St. Louis, MO, USA, in 2011, and the M.S. and the Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 2013 and 2018, respectively.

He was a member of the Brains in Silicon Laboratory and the Braindrop Chip Design Team, Stanford University, where he developed a statistical analysis of spiking communication and computation and the chip's asynchronous-digital router. He is currently with a startup, where he is involved in researching and developing neuromorphic circuits for ultralow-power sensor processing.

Ben V. Benjamin received the B.Tech. degree in electronics and communication engineering from Mahatma Gandhi University, Kottayam, India, in 2005, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 2010 and 2018, respectively.

He was a Design Engineer with the VSBU Group, Wipro Technologies, Bangalore, India, for three years. He led the testing and characterization of Neurogrid, the design and implementation of the software driver used to program and communicate with Neurogrid, and the design of Braindrop's soma and synapse circuits. He is currently with a startup, where he is involved in researching and developing dexterous robots. He holds two U.S. patents for his work on standard cell characterization.

Dr. Benjamin received the Prodigy Award for the Best Incoming Employee of the Year from Wipro Technologies.

Terrence C. Stewart received the B.A.Sc. degree in systems design engineering from the University of Waterloo, Waterloo, ON, Canada, in 1999, the M.Phil. degree in computer science and artificial intelligence from Sussex University, Brighton, U.K., in 2000, and the Ph.D. degree in cognitive science from Carleton University, Ottawa, ON, Canada, in 2007.

He is currently a Post-doctoral Research Associate with the Department of Systems Design Engineering, Center for Theoretical Neuroscience, University of Waterloo, where he is involved in creating large-scale computational neuroscience models of human cognition and implementing these models on neuromorphic hardware.

Nick N. Oza received the B.S degree (honors) in computer engineering from University of California at Santa Barbara, Santa Barbara, CA, USA, in 2004, and the M.S. degrees in bioengineering and in management science and engineering from Stanford University, Stanford, CA, USA, in 2006 and 2007, respectively.

He was a Staff Hardware Engineer with the Brains in Silicon Laboratory, Stanford University, where he contributed to the many peripheral tasks involved with developing, integrating, and testing Braindrop.

Aaron R. Voelker received the B.Math. degrees (honors) in computer science and in combinatorics and optimization from the University of Waterloo, Waterloo, ON, Canada, in 2013, where he is currently pursuing the Ph.D. degree in computer science with the Center for Theoretical Neuroscience, under the supervision of Dr. C. Eliasmith.

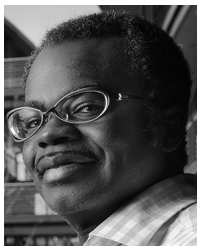
Chris Eliasmith received the B.A.Sc. degree in systems design engineering and the M.A. degree in philosophy from the University of Waterloo, Waterloo, ON, Canada, and the Ph.D. degree in philosophy–neuroscience–psychology from the Washington University in St. Louis, St. Louis, MO, USA.

He is currently the Director of the Centre for Theoretical Neuroscience, University of Waterloo, where he holds the Tier I Canada Research Chair in Theoretical Neuroscience. He has authored or co-authored two books and over 100 publications in philosophy, psychology, neuroscience, computer science, and engineering. His book *How to Build a Brain* (Oxford, U.K.: Oxford Univ. Press, 2013), describes the Semantic Pointer Architecture for constructing large-scale brain models. It builds on his earlier work describing the Neural Engineering Framework with C. H. Anderson. His team built what is currently the world's largest functional brain model, Spaun, which received the NSERC Polanyi Prize in 2015 for the most outstanding scientific or engineering advance in Canada.

Rajit Manohar (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the California Institute of Technology, Pasadena, CA, USA, in 1994, 1995, and 1998.

From 1998 to 2016, he was a Stephen H. Weiss Presidential Fellow on the Faculty of Cornell University, Ithaca, NY, USA. Since 2017, he has been with the Yale Faculty, Yale University, New Haven, CT, USA, where his group conducts research on the design, analysis, and implementation of self-timed systems. He is currently the John C. Malone Professor of electrical engineering and a Professor of computer science at Yale University. He founded the Computer Systems Lab at both Cornell University and Yale University. His work includes the design and implementation of a number of self-timed VLSI chips, including the first high-performance asynchronous microprocessor, the first microprocessor for sensor networks, the first asynchronous dataflow field-programmable gate array (FPGA), the first radiation hardened SRAM-based FPGA, and the first deterministic large-scale neuromorphic architecture.

Dr. Manohar was a recipient of the NSF CAREER award, nine best paper awards, and nine teaching awards, and was named MIT Technology Review's Top 35 Young Innovators Under 35 for his contributions to the low-power microprocessor design.



Kwabena Boahen (Fellow, IEEE) received the B.S. and M.S.E. degrees in electrical and computer engineering from Johns Hopkins University, Baltimore, MD, USA, in 1989, and the Ph.D. degree in computation and neural systems from the California Institute of Technology, Pasadena, CA, USA, in 1997.

From 1997 to 2005, he was with the Bioengineering Faculty, University of Pennsylvania, Philadelphia, PA, USA, where he held the first Skirkanich Term Junior Chair. He is currently a Professor of bioengineering with Stanford University, Stanford, CA, USA, with a joint appointment in electrical engineering, where he is also the Founding Director of the Brains in Silicon Laboratory, which develops silicon integrated circuits that emulate the way neurons compute, linking the seemingly disparate fields of electronics and computer science with neurobiology and medicine. His scholarship is widely recognized, with over 90 publications to his name, including a cover story in the May 2005 Issue of *Scientific American*, featuring his lab's work on a silicon retina that could be used to give the blind sight.

Dr. Boahen was an Elected Fellow of the American Institute for Medical and Biological Engineering in 2016, in recognition of his lab's work on Neurogrid, a specialized hardware platform that enables the cortex's inner workings to be simulated in real time—something outside the reach of even the fastest supercomputers. He received several distinguished honors, including the National Institutes of Health Director's Pioneer Award in 2006. His 2007 Technology Entertainment and Design (TED) talk, A Computer That Works Like the Brain, has been viewed half-a-million times.