# Mapping and Validating a Point Neuron Model on Intel's Neuromorphic Hardware Loihi

*Srijanie Dey\* and Alexander Dimitrov\**

*Department of Mathematics, Washington State University, Vancouver, WA, United States*

Neuromorphic hardware is based on emulating the natural biological structure of the brain. Since its computational model is similar to standard neural models, it could serve as a computational accelerator for research projects in the field of neuroscience and artificial intelligence, including biomedical applications. However, in order to exploit this new generation of computer chips, we ought to perform rigorous simulation and consequent validation of neuromorphic models against their conventional implementations. In this work, we lay out the numeric groundwork to enable a comparison between neuromorphic and conventional platforms. "Loihi"—Intel's fifth generation neuromorphic chip, which is based on the idea of Spiking Neural Networks (SNNs) emulating the activity of neurons in the brain, serves as our neuromorphic platform. The work here focuses on Leaky Integrate and Fire (LIF) models based on neurons in the mouse primary visual cortex and matched to a rich data set of anatomical, physiological and behavioral constraints. Simulations on classical hardware serve as the validation platform for the neuromorphic implementation. We find that Loihi replicates classical simulations very efficiently with high precision. As a by-product, we also investigate Loihi's potential in terms of scalability and performance and find that it scales notably well in terms of run-time performance as the simulated networks become larger.

Keywords: neuromorphic computing, LIF models, neural simulations, validation, performance analysis

## 1. INTRODUCTION

The human brain is a rich complex organ made up of numerous neurons and synapses. Replicating the brain structure and functionality in classical hardware is an ongoing challenge given the complexity of the brain and limitations of hardware. The advent of supercomputers now allows for complex neural models, but at a huge cost of both software complexity and energy consumption.

A recent intense focus on brain studies, with the BRAIN initiative at the US (Insel et al., 2013), the Human Brain Project (HBP) in Europe (Markram et al., 2011), and philanthropic endeavors like Janelia Research Campus (Winnubst et al., 2019), and the Allen Institute for Brain Science (AIBS) (Lein et al., 2007), has produced a wealth of new data and knowledge, from records of neuronal and network dynamics, to fine-grained data on network micro- and nano-structure, bringing in the era of big neural data. At the same time, advances in electronics and the search for post-von Neumann computational paradigms has led to the creation of neuromorphic systems like Intel's Loihi (Davies et al., 2018), IBM's TrueNorth (Akopyan et al., 2015; DeBole et al., 2019; Löhr et al., 2020) and HBP's SpiNNaker (Khan et al., 2008), and BrainScaleS (Grübl et al., 2020).

Neuromorphic chips, as the name suggests—"like the brain"—can mimic the brain's function in a truer sense as their design is analogous to the brain (Thakur et al., 2018; Roy et al., 2019). Inspired by its architecture, we work on developing a principled approach toward obtaining simulations of biologically relevant neural network models on a novel neuromorphic commercial hardware platform.

Computers today are limited in this respect because of the way they have been built historically and the way they process data leading toward more energy and resource consumption in order to maintain versatility (Nawrocki et al., 2016; Ou et al., 2020). Neuromorphic chips on the other hand claim to be faster and more efficient for a set of specialized tasks (Bhuiyan et al., 2010; Sharp and Furber, 2013). In this study, we lay out a numeric groundwork to validate this assertion based on neural models derived from the primary visual cortex (VISp) of the mouse brain, as seen in recent work done on SpiNNaker (Knight and Furber, 2016; Rhodes et al., 2019). Intel's neuromorphic chip "Loihi" serves as our neuromorphic platform. Results obtained in Loihi are validated against classical simulations (Rossant et al., 2010; Nandi et al., 2020; Wang et al., 2020) given by AIBS's software package the Brain Modeling Toolkit (BMTK) (Dai et al., 2020).

In this manuscript we focus on the Loihi architecture, as it is at present one of the most powerful platforms with specialized digital hardware and significant software support. While TrueNorth has a similar combination of hardware and programming support, its inter-neuron connectivity capability is relatively limited; Loihi approaches the human-scale connectivity density of interest to our research. SpiNNaker has similar capabilities, but is constructed of standard CPU hardware. Loihi's capabilities on the other hand, are built-in on a chip, thus forcing us to explore new programming paradigms. And recent and current state of the art hybrid analog-digital platforms, like Neurogrid (Benjamin et al., 2014), Braindrop (Neckar et al., 2019), DYNAP-SE2 (Moradi et al., 2017), and BrainScaleS(2) (Pehle et al., 2022) are beyond the scope of this manuscript. However, we believe that the simulation and programming paradigms developed on the Loihi platform can generalize to these analog platforms as well, and thus decrease the development time on these unfamiliar architectures.

We present one of our main motivations for this project in **Figure 1**, which highlights Loihi's advantage in performance when compared to standard simulations. Overall, our initial implementation indicates that Loihi is quite efficient in terms of compute-time in context of large brain network simulations and thus shapes our central motivation for this work (see **Figure 1** and **Table 3**). This manuscript mainly focuses on the trade-offs necessitated by these implementations, that is, how precise are the Loihi simulations when validated against BMTK simulations, given their very different hardware and programming architectures?

As a starting point, we focus on a class of neural network building blocks: point neuronal models as used in large AIBS simulations of biological neural networks. We do so because the Generalized Leaky Integrate and Fire Models (GLIFs, Teeter et al., 2018) have been found to be appropriate for reproducing cellular data under standardized physiological conditions. The
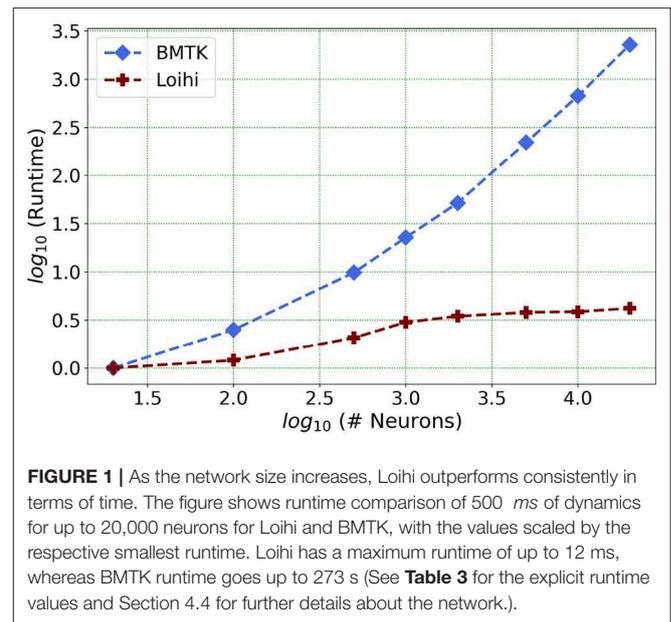


**FIGURE 1 |** As the network size increases, Loihi outperforms consistently in terms of time. The figure shows runtime comparison of 500 *ms* of dynamics for up to 20,000 neurons for Loihi and BMTK, with the values scaled by the respective smallest runtime. Loihi has a maximum runtime of up to 12 ms, whereas BMTK runtime goes up to 273 s (See **Table 3** for the explicit runtime values and Section 4.4 for further details about the network.).

data used for this study is made available by the AIBS (AIBS, 2020).

The paper is organized as follows. In Section 2, we describe in detail the features of Loihi and the differences between the neuromorphic and classical hardware that form the basis for this study. Section 3 explains the implementation of the continuous LIF equation on classical computational architecture using BMTK vs. the discrete Loihi setting. Also, we list the validation methods and the cost function that is used to draw comparisons between the implementations. In Section 4, we list out and explain the various results leading to a qualitative and quantitative assessment between the two platforms based in part on methods from Gutzen et al. (2018). Finally, Section 5 lays the ground for future work with expected improvements based on the second generation of the Loihi chip, Loihi 2 (Intel, 2022).

## 2. COMPARISON BETWEEN CLASSICAL AND NEUROMORPHIC PLATFORMS

At present, various simulators are available for implementing spiking neural networks (Brette et al., 2008). In this section, we lay out the details of the mathematical model and the platforms we use for our work. For the classical simulation, we use the Brain Modeling Toolkit (BMTK) (Dai et al., 2020) developed by the AIBS. Being open source, these resources enable us to experiment with a varied range of data and thus support our extensive validation of neuronal models in Loihi. Intel's fifth-generation chip Loihi provides us with the tools to implement and test out the various neuromorphic features. The output provided by Loihi simulations is then compared to the output of classical simulations implemented in BMTK.

### 2.1. The Brain Modeling Toolkit (BMTK)
The BMTK is a python-based software package for creating and simulating large-scale neural networks. It supports models of

different resolutions, namely, Biophysical Models, Point Models, Filter Models, and Population Models along with the use of the rich data sets of the Allen Cells Database (Lein et al., 2007; AIBS, 2020). It leverages the modeling file format SONATA which includes details on cell, connectivity and activity properties of a network along with being compatible with the neurophysiology data format Neurodata Without Borders (NWB), thus allowing easy access to a vast repertoire of experimental data.

In this study, we work with the Point Neuron Models with simulations supported by the BMTK module PointNet *via* NEST 2.16 (Kunkel et al., 2017; Linssen et al., 2018). For analysis and visualization, we use the HDF5 output format, underlying both SONATA and NWB's spike and time series storage.

The classical BMTK simulation are instantiated and run on a single node of Kamiak, the institutional high performance computing cluster. A typical Kamiak node contains 2 Intel Xeon E5-2660 v3 CPUs at 2.60 GHz, with 20 cores and 128–256 GB RAM (CIRC and WSU, 2021).

## 2.2. Loihi

Neuromorphic hardware inspired by the structure and functionality of the brain, envisioned to provide advantages such as low power consumption, high fault tolerance and massive parallelism for the next generation of computers, is called neuromorphic hardware. Toward the end of 2017, Intel Corporation unveiled its experimental neuromorphic chip called Loihi. We provide a summary of the platform here.

As of its 2020 rendition, the version on which these results were evaluated, Loihi is a 60-mm$^2$ chip that implements 131,072 leaky-integrate-and-fire neurons. According to Davies et al. (2018), it uses an asynchronous spiking neural network (SNN), comprising of 128 neuromorphic cores, each with 1,024 neural computational units; 3 × 86 cores; along with several off-chip communication interfaces that provide connectivity to other chips. As Loihi advances the modeling of SNNs in silicon, it comprises of a large number of features necessary for their implementation viz., hierarchical connectivity, dendritic compartments, synaptic delays and synaptic learning rules. Each neuron is represented as a compartment in the Loihi architecture, i.e., it is designed to resemble an actual biological neuron model comprising of all the functional units (**Figure 2**). The SYNAPSE unit processes all the incoming spikes from the previous compartment/neuron and captures the synaptic weight from the memory. The DENDRITE unit updates the different state variables. The AXON unit generates the spike message to be carried ahead by the fan out cores. The LEARNING unit updates the synaptic weights based on a learning rule and is not used in this project.

The aim of this study is to establish the groundwork required to execute an ambitious plan of simulating about ∼250,000 neurons with ∼500M synapses in the future, which encapsulates much of the experimentally observed dynamics in the mouse visual cortex available to the AIBS, thus providing a close functional replica of the mouse visual cortex. Loihi's specialized hardware features hold promise for a real-time, low-powered version of such an implementation.

## 2.3. Leaky Integrate and Fire Model (LIF)

A typical neuron consists of a soma, dendrites, and a single axon. Neurons send signals along an axon to a dendrite through junctions called synapses. The classical Leaky Integrate and Fire (LIF) equation (Gerstner and Kistler, 2002) is a point neuron model which reduces much of the neural geometry and dynamics in order to achieve computational efficiency. It is one of the simplest and rather efficient representations of the dynamics of the neuron, while still providing reasonable approximation of biological neural dynamics for *some* classes of neurons (Teeter et al., 2018). It is stated mathematically as:

$$V'(t) = \frac{1}{C}\left[I_e(t) - \frac{1}{R}(V(t) - E_L)\right] \tag{1}$$

$$V(t) \leftarrow V_r, \quad \text{if} \ \ V(t) > \Theta \tag{2}$$

where,

$$V(t) = \text{membrane potential (state)}$$
$$C = \text{membrane capacitance (parameter)}$$
$$R = \text{membrane resistance (parameter)}$$
$$E_L = \text{resting potential (parameter)}$$
$$I_e = \text{trans-membrane current (control and state)}$$
$$V_r = \text{reset membrane potential}$$
$$\Theta = \text{firing threshold}$$

Here, $' = d/dt$, $t$ is time in *ms*, the membrane potential $V(t)$ of the neuron is in mV. These specific physical units are followed based on what the AIBS datasets use to define the respective physiology measurements. A LIF neuron fires when $V(t) > \Theta$, i.e., the membrane potential exceeds the firing threshold $\Theta$ and subsequently the membrane potential is set to a reset value $V_r$.

The classical LIF model (point generalized LIF) has been shown to match well the dynamics of some mouse neurons under a variety of conditions (Teeter et al., 2018), as listed in the Allen Cell Types Database (Lein et al., 2007). In addition, this model matches the LIF abstraction in Loihi to some extent (as Loihi uses discrete time discrete state dynamics to emulate the continuous time continuous state dynamics of the model). Thus, we work with this model throughout this study to establish the basis for comparison for the two platforms, determine how closely such a discrete dynamical system can get to simulations of a continuous dynamical system, validate the neuromorphic implementation against the ground truth of a standard implementation, and provide evidence that our neuromorphic platform performs more efficiently.

## 2.4. Loihi LIF Model

In an SNN, spiking neurons form the primary processing elements. The individual neurons are connected through junctions called synapses and interact with each other through single-bit events called spikes. Each spike train can be represented as a list of event times, e.g., as a sum of Dirac delta functions $\sigma(t) = \sum_i \delta(t - t_i)$ where $t_i$ is the time of the $i$-th spike.

Since Loihi encapsulates the working of an SNN, one of the computational models it implements is a variation of the LIF
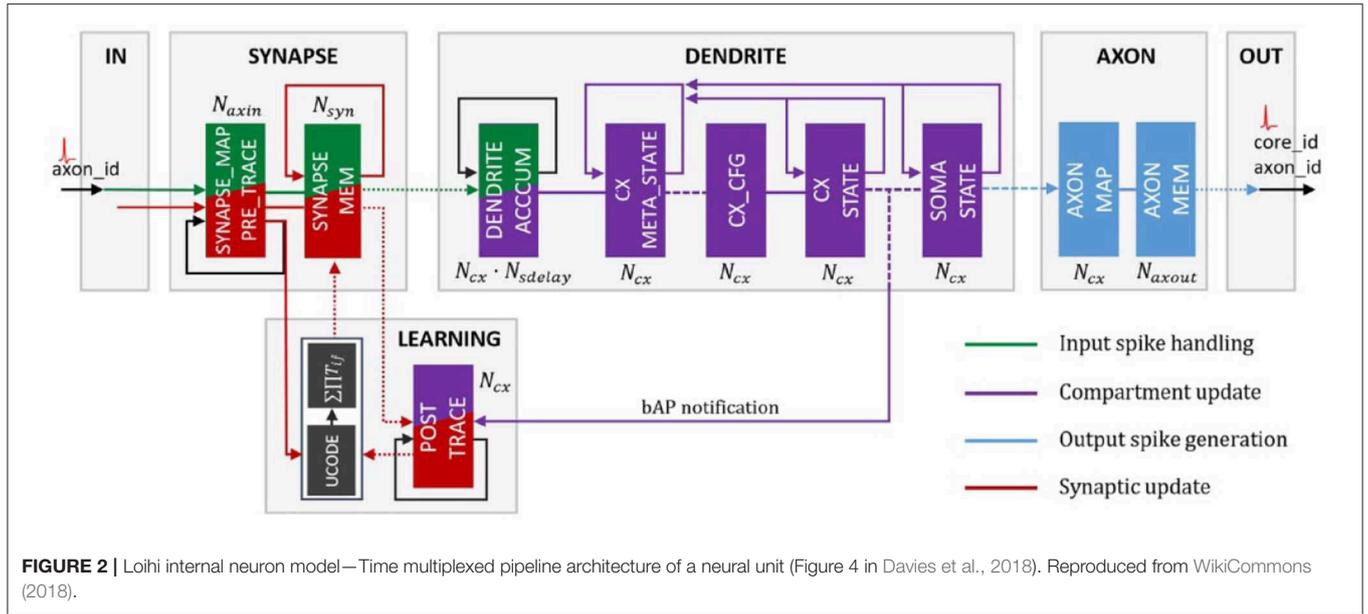
**FIGURE 2** | Loihi internal neuron model—Time multiplexed pipeline architecture of a neural unit (Figure 4 in Davies et al., 2018). Reproduced from WikiCommons (2018).

model based on two internal state variables : the synaptic current and the membrane potential (Davies et al., 2018).

$$u(t) = \sum_j w_j(\alpha_j * \sigma_j)(t) + b \qquad (3)$$

$$v'(t) = -\frac{1}{\tau_v}v(t) + u(t) \qquad (4)$$

$$v(t) \leftarrow 0, \text{ if } v(t) > \theta \qquad (5)$$

where,

$$v(t) = \text{membrane potential}$$
$$u(t) = \text{synaptic current}$$
$$w = \text{synaptic weight}$$
$$\alpha = \text{synaptic response function}$$
$$b = \text{constant bias current}$$
$$\tau_v = \text{time constant}$$
$$\theta = \text{firing threshold}$$

A neuron sends out a spike when its membrane potential exceeds its firing threshold $\theta$, i.e., $v(t) > \theta$. After a spike occurs, $v(t)$ is reset to 0. As in the classical LIF model, here $' = d/dt$. However, time and membrane potential values here are in arbitrary units.

Loihi follows a fixed-size discrete time-step model, similar to an explicit Euler integration scheme, where the time steps relate to the algorithmic time of the computation. This algorithmic time may differ from the hardware execution time. Moreover, to increase the efficiency of the chip, specific bit-size constraints are imposed on the state variables. We discuss the ones relevant for the LIF model implementation in the following section.

## 3. METHODS

### 3.1. Model Setup and Integration

The classical LIF model as represented in Equations (1) and (2) can be rewritten as :

$$V'(t) = -\frac{1}{\tau_v}V(t) + \frac{1}{C}\left[I_e(t) + \frac{1}{R}E_L\right] \qquad (6)$$

where $\tau_v = RC$ is membrane time constant of the neuron.

For a non-homogeneous linear differential equation,

$$\frac{df}{dt} = af + g \qquad (7)$$

the solution is given by the "variation of constants" method as :

$$f(t) = e^{at}\int_0^t g(s)e^{-as}ds$$

Comparing Equation (6) to Equation (7), we have,

$$a = \frac{1}{\tau_v}$$
$$f = V(t)$$
$$g = \frac{1}{C}(I_e) + \frac{1}{\tau_v}(E_L)$$

Here, the postsynaptic current $I_e$ is in the form of an exponent function. However, calculating the above integral at every step i.e., at all grid points $t_i \leq t$ proves to be quite expensive.

BMTK uses NEST as backend to implement the above membrane potential dynamics. To avoid the expensive computations, NEST chooses to use the linear exact integration method (Rotter and Diesmann, 1999), given below as follows :

Equation (6) is rewritten as a multidimensional homogeneous differential equation:

$$\frac{d}{dt}y = Ay \qquad (8)$$

where,

$$A = \begin{pmatrix} a_n & a_{n-1} & \cdots & \cdots & a_1 & 0 \\ 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 & 0 \\ 0 & 0 & \ddots & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{C} & \frac{1}{-\tau} \end{pmatrix}$$

The solution is given by :

$$y(t) = e^{At}y_0 \qquad (9)$$

$$y_{t+h} = y(t+h) = e^{A(t+h)}y(0) = e^{Ah} \cdot y_t \qquad (10)$$

for a fixed time-step $h$. It saves exorbitant computations since each evaluated step involves multiplication only, and intermediate steps between events do not have to be computed.

### 3.1.1. Mapping Between BMTK and Loihi Models
In this section, we illustrate the primary step of implementing the BMTK-NEST LIF integration with the Loihi dynamic computational model. Loihi follows a discrete-state, discrete-time computational model, similar to an explicit Euler integration scheme. This allows it more flexibility for integrating non-linear neural model, but, and unlike NEST's exact integration method, Loihi's engine accumulates errors at each time step. The time steps in the Loihi model relate to the algorithmic time of the computation which may differ from hardware execution time. Following the linear exact numerics in the NEST implementation, we implement our model in the Loihi discrete setting using the forward Euler method for guidance, as discussed below.

**Step 1**

First, we rewrite the standard LIF model in Equation (1) to resemble the Loihi form as given in Equation (4). Since Loihi parameters are unit-less, we introduce a re-scaling parameter $V_s$, which converts standard physical units used in BMTK to Loihi units.

As we compare Equations (2) and (5), it can be seen that for BMTK the membrane potential reset value is set to $V_r$ whereas it is set to zero for Loihi. To account for that, we shift the BMTK representation by $V_r$. Thus, the forward transformation from BMTK to Loihi looks as follows :

$$v = (V - V_r)/V_s \qquad (11)$$

which produces an inverse transformation, to arrive back at the BMTK values, given by :

$$V = v \cdot V_s + V_r \qquad (12)$$

**Step 2**

Substituting the expression in (12) in (1) and isolating $v$, we get :

$$V'(t) = \frac{1}{C}\left[I_e(t) - \frac{1}{R}(V(t) - E_L)\right] \qquad |V = vV_s + V_r \implies \qquad (13)$$

$$v'(t)V_s = -\frac{1}{RC}(vV_s + V_r - E_L) + \frac{1}{C}I_e(t) \qquad |/V_s \implies \qquad (14)$$

$$v'(t) = -\frac{1}{\tau_v}v(t) + \frac{1}{\tau_v}\frac{E_L - V_r}{V_s} + \frac{1}{C}\frac{I_e}{V_s} \qquad (15)$$

$$= -\frac{1}{\tau_v}v(t) + u(t) \qquad (16)$$

with

$$v(t) = \frac{V(t) - V_r}{V_s}, \qquad (17)$$

$$u(t) = \frac{1}{CV_s}I_e(t) + \frac{1}{\tau_v}\frac{E_L - V_r}{V_s}, \qquad (18)$$

$$\tau_v = RC, \qquad (19)$$

$$\theta = \frac{\Theta - V_r}{V_s} \qquad (20)$$

Here, we reintroduce the LIF threshold $\Theta$ and the corresponding Loihi threshold $\theta$ in Equation (20), which is derived from $\Theta$ by the same shift and re-scaling that converted $V$ to $v$.

To reiterate, Loihi implements the continuous LIF as a discrete finite state machine model (Jin et al., 2008; Mikaitis et al., 2018) implemented in silicon. The actual computation is similar to a forward Euler scheme with some peculiarities reflecting engineering design trade-offs. Specifically, the $v(t)$ state evolves on-chip according to the update rule,

$$v(t+1) = v(t)\left[1 - \frac{\delta_v}{2^{12}}\right] + b + u(t) \qquad (21)$$

where $\delta_v$ is the membrane potential decay constant and $b$ is the constant bias current listed in Equation (3).

**Step 3**

Using the forward Euler method :

$$y_{n+1} = y_n + f(t_n, y_n).dt$$

where $y_{n+1} = y(t_{n+1})$ and $t_{n+1} = t_n + dt$ for a fixed time-step $dt$, we transform the classical LIF model into a form followed in Equation (21). Thus, transforming the LIF model into the discrete form and grouping terms to match the Loihi integration (9) yields the following :

$$\frac{v(t+dt) - v(t)}{dt} = -\frac{1}{\tau_v}v(t) + u(t) \qquad (22)$$

$$\implies v(t+dt) = v(t)(1 - \frac{dt}{\tau_v}) + u(t)dt \qquad (23)$$

where $dt$ is the fixed time-step with which we can adjust the temporal precision of the Euler integration scheme.

In order to equate the Loihi computation (21) with the Euler scheme (23), we use $dt$ with units *ms/Loihi timestep* i.e., 1 BMTK millisecond per Loihi timestep. Thus, comparing Equations (21) and (23) defines the Loihi voltage decay parameter $\delta_v$ in terms of the timestep $dt$, i.e.,

$$(2^{12} - \delta_v)\, 2^{-12} = (1 - \frac{dt}{\tau_v}) \tag{24}$$

$$\implies \delta_v = \frac{dt}{\tau_v} 2^{12} = \frac{dt}{RC} 2^{12} \tag{25}$$

### 3.1.2. Bit Constraints

Given its discrete setting, there are specific bit-size constraints that Loihi imposes on the state variables and parameters. State variables—membrane potential and current—are allotted $\pm 23$ bits each. The membrane potential decay constant $\delta_v$ is allotted 12 bits and the membrane potential threshold is assigned 17 bits interpreted as the 17 high bits of a 23 bit word to match the state variables size. Details on other parameters can be found under Table 1 in Davies et al. (2018) and Table 6 in Michaelis et al. (2021).

## 3.2. Validation Methods

### 3.2.1. Data

The data used here is provided by the Allen Mouse Brain Atlas (Lein et al., 2007; AIBS, 2020), which is a survey of single cells from the mouse brain, obtained *via* intracellular electrophysiological recordings done through a highly standardized process. We focus on neurons of different types with available GLIF parameters. The data used can be accessed in the Allen Cell Types Database. Our LIF model is implemented and simulated on BMTK based on this data, and these simulations form the ground truth for validating the Loihi implementations.

The datasets used for the simulations in this work can be found in our Github repository (Dey, 2022).

### 3.2.2. Cost Functions

To quantify the error between the BMTK and Loihi membrane potential values, we use two related cost functions: the Root Mean Square Error (RMSE) and the Pearson correlation coefficient ($r$) with values as follows :

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(y_L^i - y_B^i\right)^2} \tag{26}$$

where,

$$i = \text{index of data point}$$
$$y_L = \text{transformed Loihi values}$$
$$y_B = \text{original BMTK values}$$
$$n = \text{number of data points}$$

and

$$r = \frac{\sum_{i=1}^{n} \left(y_L^i - \bar{y}_L\right)\left(y_B^i - \bar{y}_B\right)}{\sqrt{\sum_{i=1}^{n} \left(y_L^i - \bar{y}_L\right)^2 \sum_{i=1}^{n} \left(y_B^i - \bar{y}_B\right)^2}} \tag{27}$$

where,

$$\bar{y}_L = \text{mean of the transformed Loihi values}$$
$$\bar{y}_B = \text{mean of the original BMTK values}$$

### 3.2.3. Other Methods

Since BMTK and Loihi run on two different computing environments, visual comparisons in graphs are helpful for diagnostics of discrepancies that may be obscured in the single numbers reported by the cost function. They also contribute to assess the level of similarity between the two implementations.

We compare the simulation dynamics for both implementations based on the following:

- **Distribution Function:** We compare the distributions of attained state values in the two cases. We use density plot as a representation of those distributions, thus allowing us to compare the two implementations in terms of concentration and spread of the values and provide a basis for comparing the collective dynamics of the implementations.
- **Raster Plot:** We evaluate the membrane potential response at each time-step. The X-axis represents the membrane potential and the Y-axis represents the time-step. Raster plot helps to visually communicate similarities between the BMTK and Loihi states, and highlight potential state-localized difference in the dynamics at each step which may otherwise be lost in the average error measures.
- **Scatter Plot:** For examining association between the two implementations, we use color-coded scatter plots identifying the correlation relationships. We add a trend line to illustrate the strength of the relationship and pin down the outliers to improve the simulation results. Since we anticipate an almost perfect linear relationship, we quantify the match with its Pearson correlation coefficient.

## 4. RESULTS

In order to lay the groundwork for simulating a network of over 250,000 neurons with a connectivity of over 500M synapses in the neuromorphic hardware, we begin by ensuring a high quality replication of individual neural and smaller network models. The replication performance here is evaluated based on membrane potential and current responses, the two state variables. We conjecture that securing a good replica for smaller models will ensure that parameters can be calibrated correctly and thus can be carried forward for the bigger networks needed in biological context (Herz et al., 2006; Gutzen et al., 2018; Trensch et al., 2018).

We begin our work on a single-neuron network[1] sub-threshold dynamics driven by both bias current and external spikes to ensure Loihi is able to handle both stimuli efficiently. Our test suite consists of LIF models based on 20 different parameter sets. We perform rigorous analysis of our results based

---

[1]A network is the smallest executable structure in Loihi, hence the peculiar term *single-neuron network.*
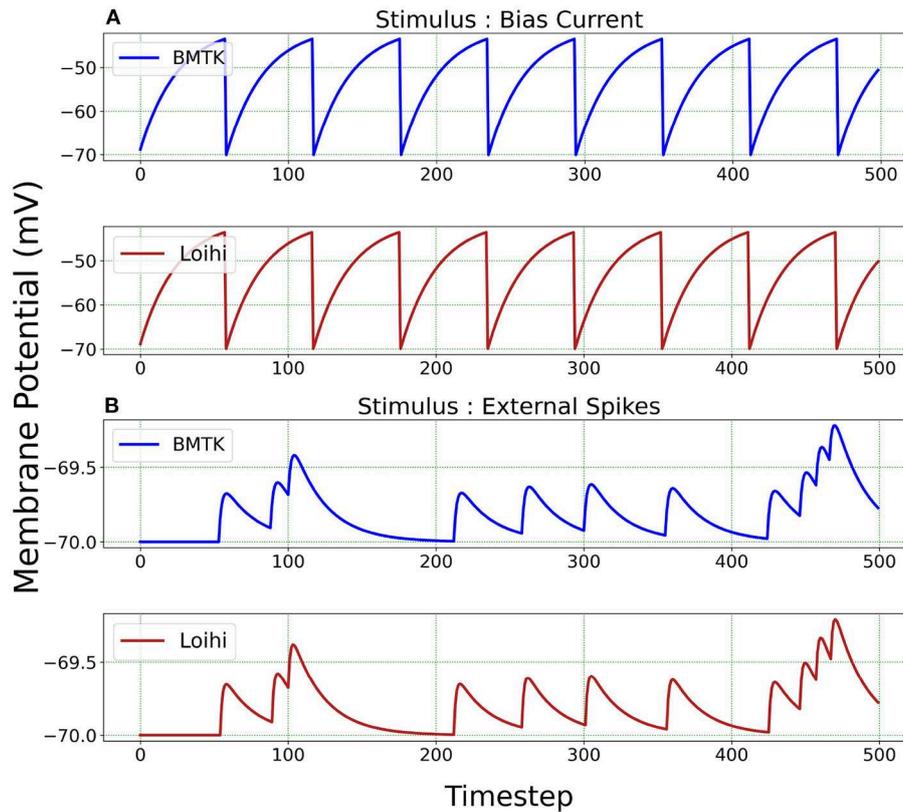
**FIGURE 3 |** Membrane potential response for single-neuron network based on two different neuron parameters. **(A)** Simulation is driven by bias current. **(B)** Simulation is driven by external spikes.

on various statistical measures and visualizations to demonstrate that we have replication of high quality. It is important to restate here that we test our results based on neurons with different morphologies and biophysics, which attribute to the different parameter sets.

## 4.1. Simulations of a Single Neuron

We begin by simulating a single-neuron network in BMTK. The simulation is run for 500 *ms*. The classical parameters are translated to Loihi values and the corresponding LIF model is implemented as an one-neuron SNN executed for 500/*dt* time steps in Loihi. The simulations are driven either by bias current or external spikes.

In the Loihi network, neurons are denoted by compartments. The compartment dynamics are hardware-constrained and determined by the parameters *bias current mantissa, membrane potential threshold, membrane potential decay*, and *current decay*. It is worth iterating here that the membrane potential values in Loihi are unit-less as opposed to the BMTK values which are assigned units of millivolts (mV) and milliseconds (ms) based on the AIBS datasets.

We test the precision of our replication, both qualitatively and quantitatively, for all 20 parameters sets and find that the results are consistent with the ones described below. In **Figure 3**, we illustrate the implementations achieved through bias current and

**TABLE 1 |** Parameter set for LIF models.

| Parameters | Dataset (1) | Dataset (2) | Units |
|---|---|---|---|
| Membrane time constant | 25.0 | 22.0 | *ms* |
| Membrane potential threshold | −43.0 | −43.0 | *mV* |
| Resting potential | −70.0 | −70.0 | *mV* |
| Voltage reset | −70.0 | −70.0 | *mV* |
| Current | 200.0 | 0.0 | *pA* |
| Membrane capacitance | 170.21 | 170.0 | *pF* |

external spikes on two different parameter sets (**Table 1**). The remaining 18 parameter sets can be found in our Github page (Dey, 2022). The parameters in the BMTK platform are mapped to Loihi using the transformations described in Section 3.1.1 with respect to the bit constraints described in Section 3.1.2.

It is to be noted here that stimulus bias current acts as one of the parameters of the LIF model and hence is mapped into Loihi according to Equation (18). When stimulating with external spikes as stimulus, we make use of the fixed-time step *dt* that we introduce in Equation (23). Here, the external spike-times are in "*ms*" and we assign unit "ms/Loihi time-step" to *dt*. Thus, the external spike-times are scaled as spike-time/*dt* and then injected into a Loihi neural unit for each time-point, with *dt* guiding the

temporal precision scale. **Table 2** shows the external spike-times used in the simulations, which are generated by five spike sources using a random Poisson spike generator with a max firing rate of 5Hz and then frozen to stimulate the different models in both BMTK and Loihi.

For a qualitative comparison, it can be seen from **Figure 3** that Loihi implementations simulate BMTK results very closely. We have close correspondence in terms of spike frequency, spike amplitude, and response values. Since Loihi membrane potential values are unit-less, we map them back to BMTK values (mV, ms) before performing the comparison. The inverse mapping from Loihi to BMTK is performed based on Equation (12), i.e.,

$$V = v \cdot V_s + V_r$$

We perform a quantitative assessment of the replication using RMSE and correlation coefficient between the values obtained from the two platforms. As seen from **Table 3**, the values are highly correlated with a relatively small RMSE.

**Figure 4** illustrates the comparison of Loihi implementations against the BMTK implementations for the two different stimuli using various graphing data—(a) Distribution function approximating the membrane potential dynamics, (b) Raster plot of the spiking network activity, (c) Scatter Plot highlighting the positive coefficient between the two implementations. These visualizations help us track discrepancies which might remain unobserved based on single quantitative averages given by the cost function or the correlation coefficient.

**TABLE 2** | External spike-time values.

| Source | Spike-times (*ms*) |
|---|---|
| 0 | 446 |
| 1 | 355 |
| 2 | 53, 258, 300, 424, 457 |
| 3 | 88, 466 |
| 4 | 100, 212 |

**TABLE 3** | Correlation and RMSE between BMTK and Loihi membrane potential values.

| Stimulus | Correlation | RMSE |
|---|---|---|
| Bias current | 0.999992 | $1.1374 \times 10^{-4}$ mV/*ms* |
| External spikes | 0.999942 | $4.208 \times 10^{-5}$ mV/*ms* |

## 4.2. Simulation Using Varied Precision

As already stated, Loihi follows a fixed-size discrete time-step model along with bit-size constraints for the different parameters. Thus, we examine how the numerics of Loihi affect its ability to faithfully implement neuron models. More precisely, we investigate how changing the precision of the time scale and the neuron state values affects the accuracy of the simulations. We explore this property for the two state variables—membrane potential and current.

**Figure 5** illustrates the membrane potential and current responses of a single neuron model in BMTK which form the basis of our comparison for the results below.
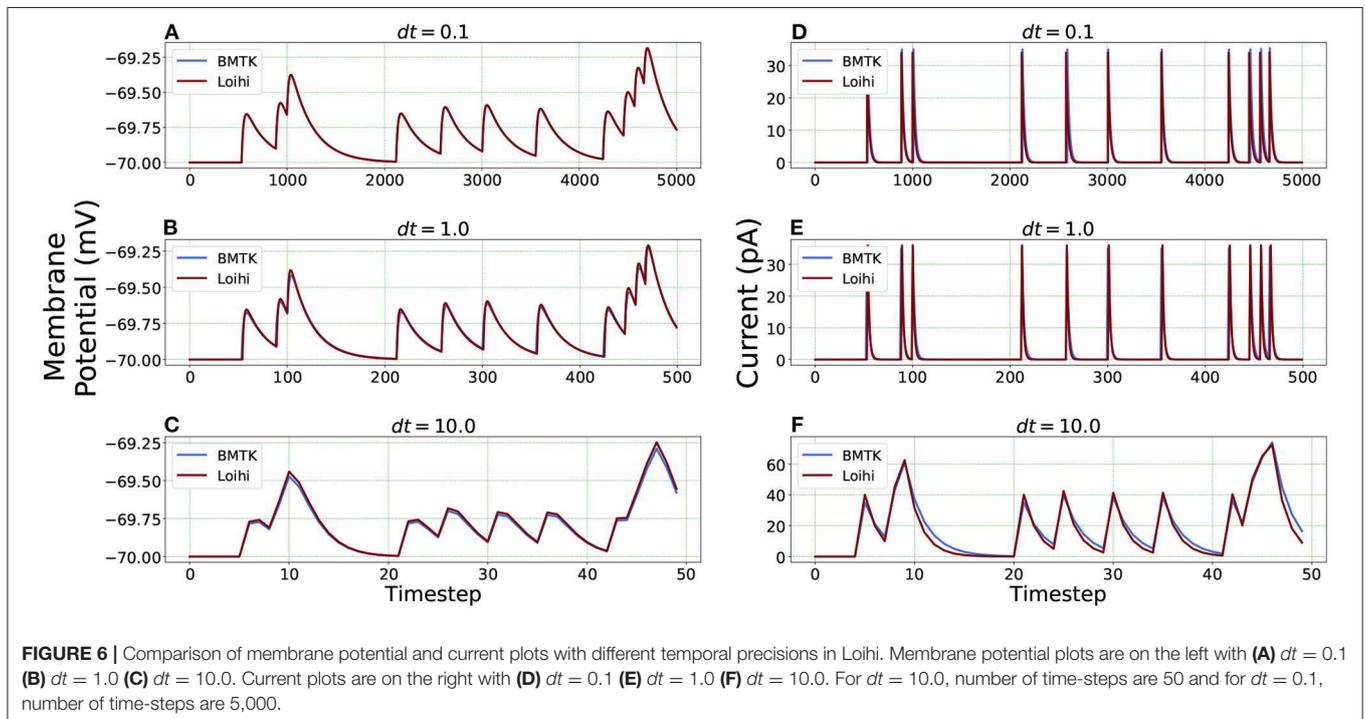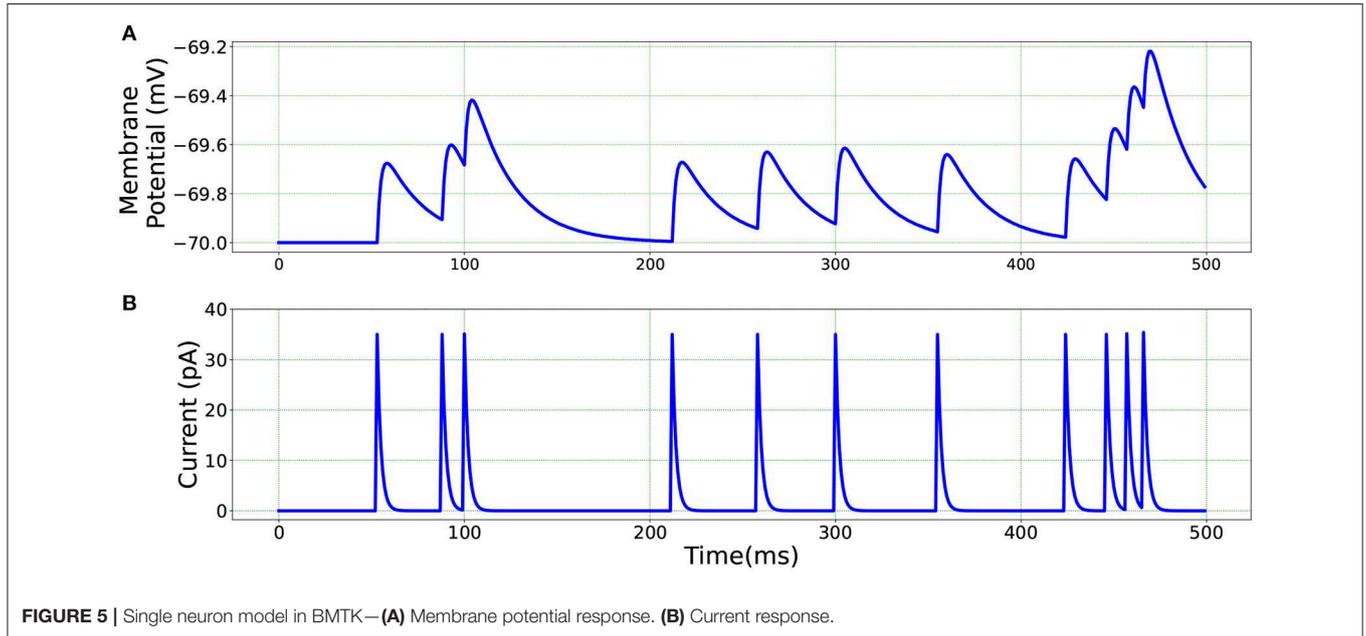


**FIGURE 4** | Validation plots for simulations based on two different stimuli—**(A)** Validation plots for bias current stimulus **(B)** Validation plots for external spike stimulus, based on the Distribution Function, Raster Plot, and Scatter Plot, respectively.
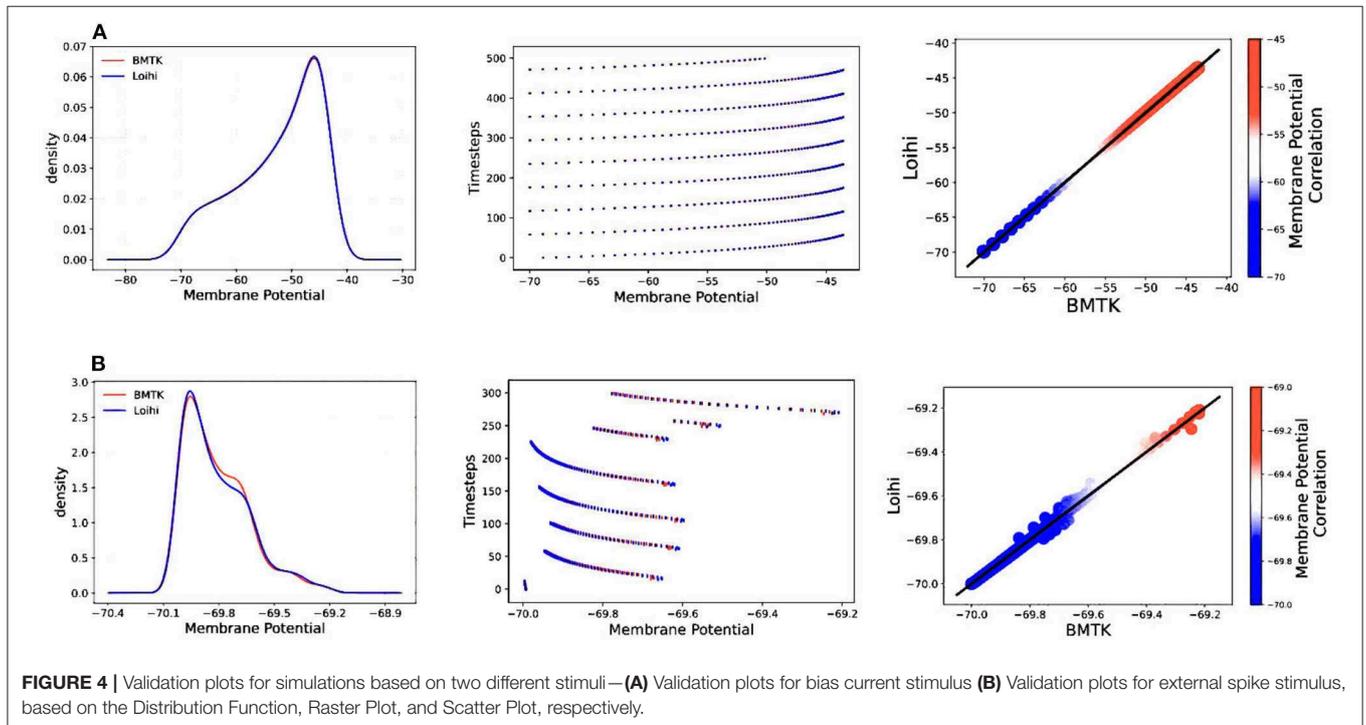
#### 4.2.1. Simulation Using Varied Temporal Precision

For Loihi's fixed simulation time-step, we assign different time units to each step and test the corresponding simulation precision. This is achieved through the "$dt$" parameters available in our equations while transforming the classical LIF model to Loihi neural model. It enables us to experiment with several time units (Hopkins and Furber, 2015). Following Equation (24), the

change of a time-step while working with the Loihi neural model necessitates a corresponding variation of the time constant "$\tau_v$" to yield the desired results.

We check the results for $dt = 0.1, 1.0$ and $10.0 (ms/timestep)$. As mentioned earlier, we run the simulation for 500 $ms$, thus the corresponding number of time steps in Loihi for $dt = 0.1$ and $dt = 10.0$ becomes 5000 and 50 respectively, and for $dt = 1.0$ it
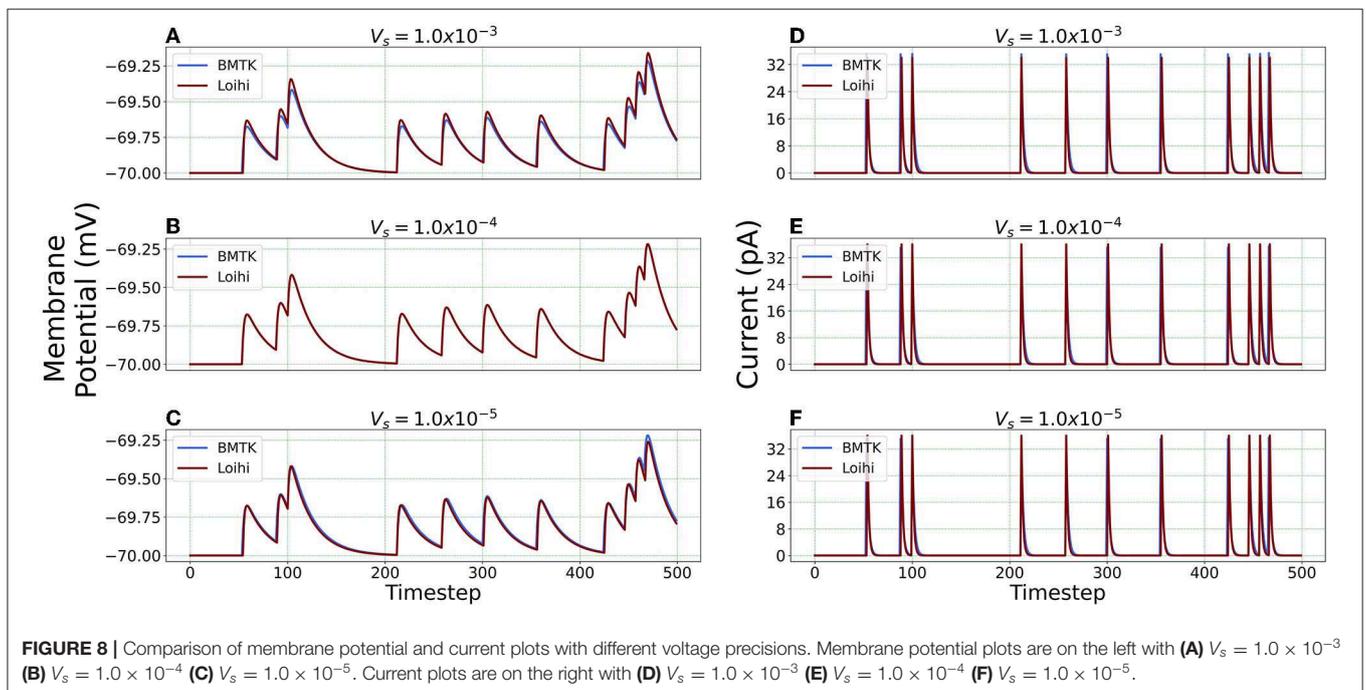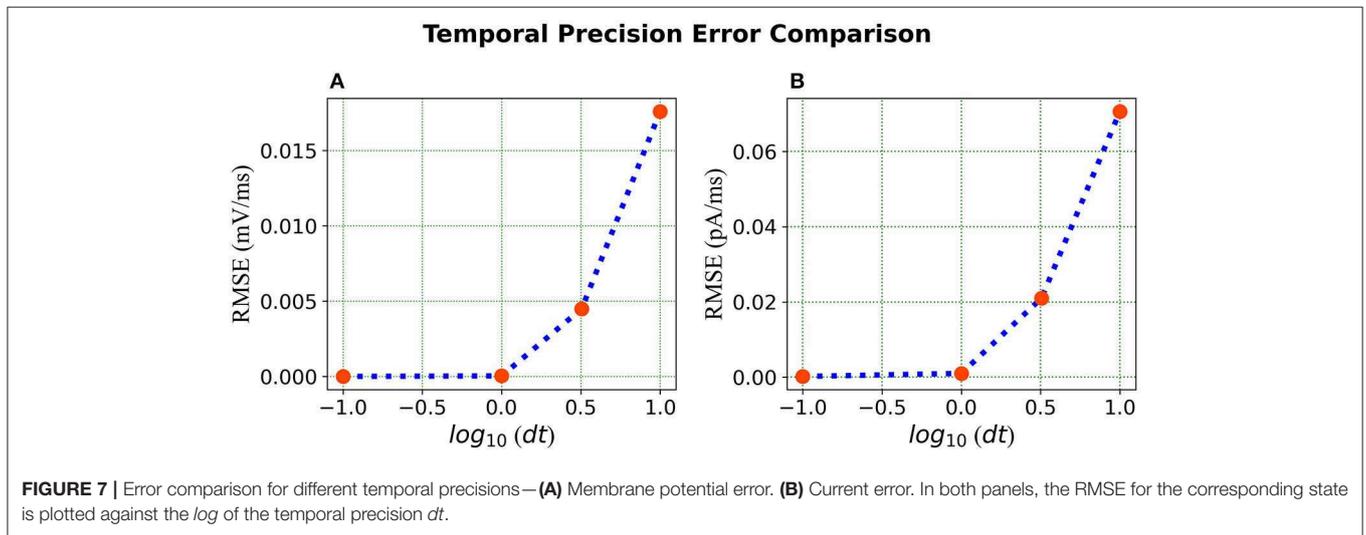


**FIGURE 5 |** Single neuron model in BMTK—**(A)** Membrane potential response. **(B)** Current response.



**FIGURE 6 |** Comparison of membrane potential and current plots with different temporal precisions in Loihi. Membrane potential plots are on the left with **(A)** $dt = 0.1$ **(B)** $dt = 1.0$ **(C)** $dt = 10.0$. Current plots are on the right with **(D)** $dt = 0.1$ **(E)** $dt = 1.0$ **(F)** $dt = 10.0$. For $dt = 10.0$, number of time-steps are 50 and for $dt = 0.1$, number of time-steps are 5,000.

remains at 500, (i.e., 500/$dt$ for each $dt$). **Figure 6** illustrates the related Loihi simulation for membrane potential and current.

**_Error comparison for temporal precision_** Unlike continuous analysis in which the error decreases monotonically with $dt$, Loihi's discrete-time, discrete-state simulation dynamics suggests that there may be an optimal $dt$ which minimizes the LIF dynamics error.

We compare the simulations in Loihi with different temporal precisions against the simulations in BMTK. We calculate the RMSE to be able to deduce the result. As can be seen from **Figure 7**, the error is lowest when 1 $ms$ of simulation time in

BMTK equates to 1 time-step in Loihi for membrane potential and current. Thus, for the LIF model simulations, representing 1$ms$ with a Loihi hardware time step provides the best match between the two simulations. As to using larger $dt$ for efficiency, panels (C) and (F) clearly show that large time steps (larger than the synaptic time constant in this case) significantly degrade the quality of simulations.

It should be noted that the selected simulation timestep $dt$ affects the range of physical time constants $\tau_v$ that can be represented in Loihi. Since $\delta_v = \frac{dt}{\tau_v} 2^{12}$ (Equation 24), then $\tau_v = \frac{dt}{\delta_v} 2^{12}$. In Loihi, $\delta_v \in [1, 2^{12}]$ (stored as a 12-bit word,



**FIGURE 7 |** Error comparison for different temporal precisions—**(A)** Membrane potential error. **(B)** Current error. In both panels, the RMSE for the corresponding state is plotted against the *log* of the temporal precision $dt$.



**FIGURE 8 |** Comparison of membrane potential and current plots with different voltage precisions. Membrane potential plots are on the left with **(A)** $V_s = 1.0 \times 10^{-3}$ **(B)** $V_s = 1.0 \times 10^{-4}$ **(C)** $V_s = 1.0 \times 10^{-5}$. Current plots are on the right with **(D)** $V_s = 1.0 \times 10^{-3}$ **(E)** $V_s = 1.0 \times 10^{-4}$ **(F)** $V_s = 1.0 \times 10^{-5}$.
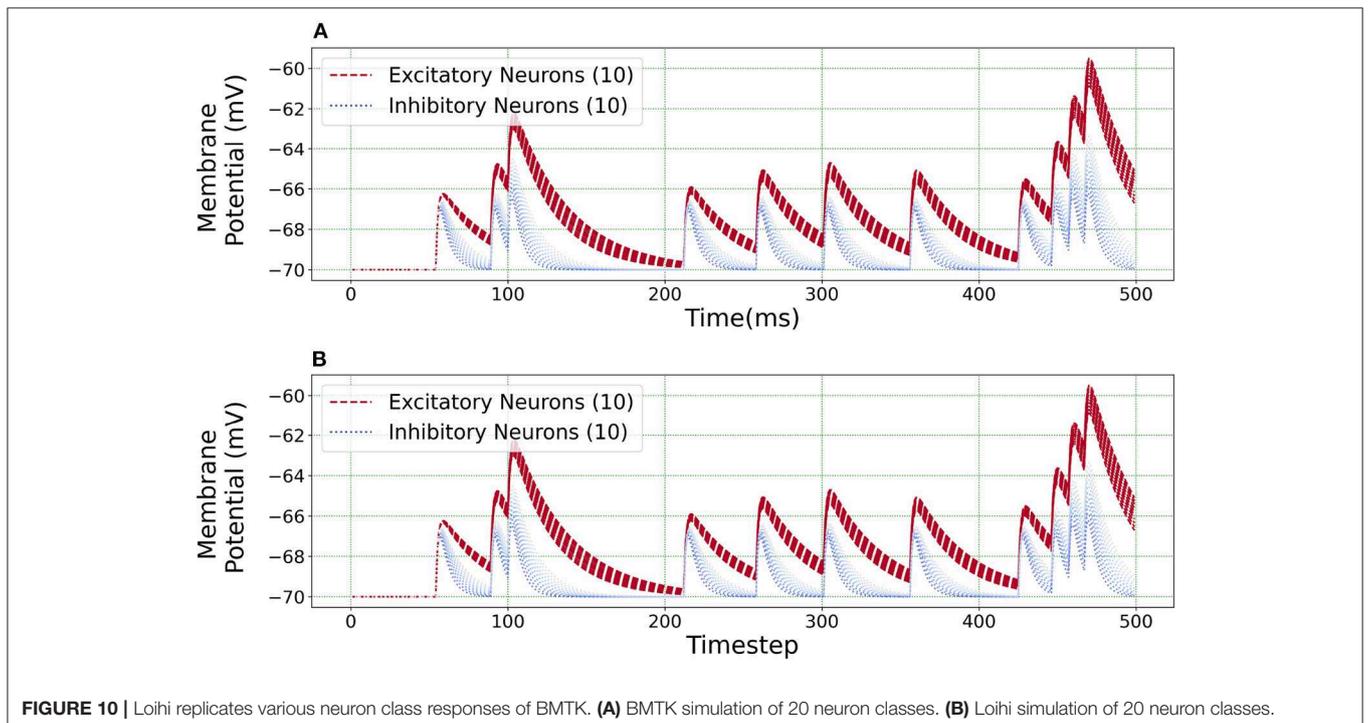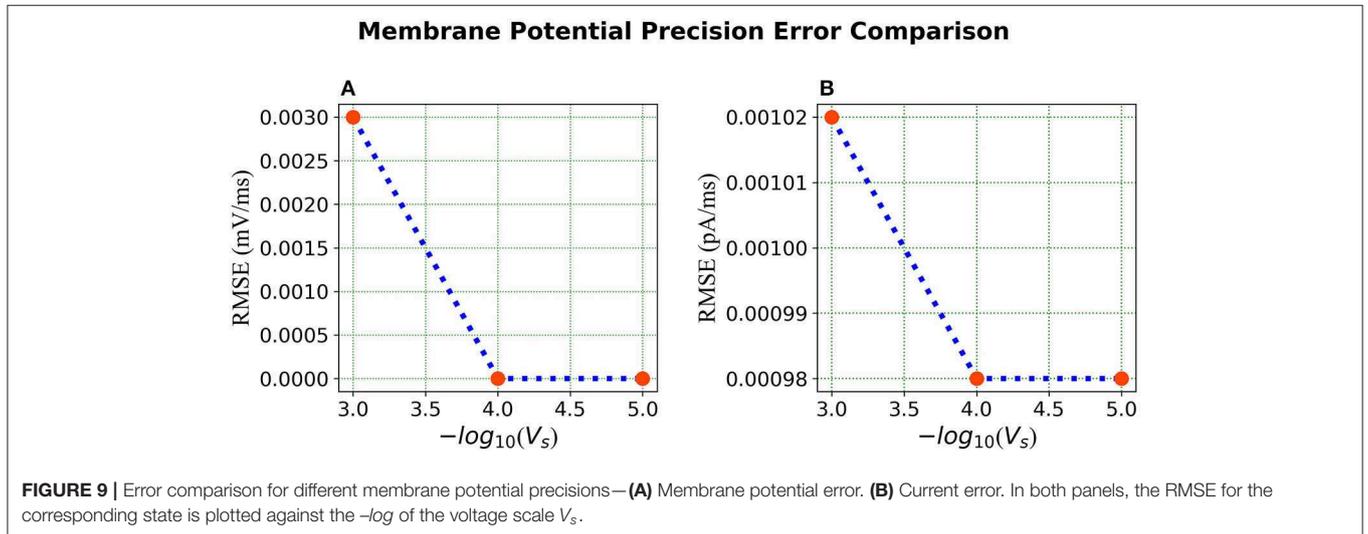
with 0 representing $\delta_v = 2^{12}$). Hence, $\tau_v \in [1, 2^{12}]dt$, that is, the highest physical time constant that can be represented is $\tau_v = 2^{12}dt \approx 4,100dt$. This is not a big constraint for $dt = 1ms$, but e.g., a higher simulation precision of $dt = 0.01ms$ can be performed only for neurons with time constants $\tau_v < 41ms$, which already excludes some models found in the Allen Institute's Cell Types Database. This shortcoming of this Loihi 1 platform is being addressed by Intel in subsequent hardware like Loihi 2 (Intel, 2022), and the new Lava SDK. Similarly affected are potential spike propagation delays (not

used here). Loihi supports ranges from 1 to 62 time steps, which translate to $dt$ to $62dtms$ of physical time. This is a minor constraint for $dt = 1ms$, but quickly becomes a significant constraint for short $dt$.

### 4.2.2. Simulation Using Varied Voltage Precision

We repeat the precision study by changing the voltage precision values using the re-scaling parameter $V_s$. To check different precision results, we try 1K/mV, 10K/mV and 100K/mV (state level/mV) by using $V_s = 1.0 \times 10^{-3}, 1.0 \times 10^{-4}$ and $1.0 \times 10^{-5}$



**FIGURE 9 |** Error comparison for different membrane potential precisions—**(A)** Membrane potential error. **(B)** Current error. In both panels, the RMSE for the corresponding state is plotted against the –*log* of the voltage scale $V_s$.



**FIGURE 10 |** Loihi replicates various neuron class responses of BMTK. **(A)** BMTK simulation of 20 neuron classes. **(B)** Loihi simulation of 20 neuron classes.

respectively. **Figure 8** illustrates the neuron state simulations based on different voltage precisions.

***Error Comparison for Voltage Precision*** As can be seen from **Figure 9**, for membrane potential—error decreases significantly as the precision increases from $V_s = 1.0 \times 10^{-3}$ to $V_s = 1.0 \times 10^{-4}$. However, the error is extremely small for the current simulation and remains the same for $V_s = 1.0 \times 10^{-4}$ and $V_s = 1.0 \times 10^{-5}$.

### 4.2.3. Effects of State Precision on Simulations

In conclusion, the effect of precision on both scales depends on the model parameters and the information needing to be preserved. However, there are important performance differences. Increased voltage precision is essentially free, as it does not tax the hardware resources any further, and the sole risk is from computation overflow in cases of the Loihi voltage state nearing the capacity of the voltage register. Increased time precision on the other hand has two important drawbacks: it increases simulation time (proportionately to increased precision), and it decreases the range of voltage decay timescales that can be represented (again, proportionately to increased precision). Thus, the choice of simulation time step and corresponding precision should be weighed against these tradeoffs.

## 4.3. Simulation of Different Neuron Classes

After establishing and verifying the calibrated Loihi parameters for a single neuron, we extend our simulation to an ensemble

**TABLE 4** | Correlation and RMSE for different neuron classes.

| Neuron class | Correlation | RMSE |
|---|---|---|
| Excitatory | 0.999989 | $0.532 \times 10^{-4}$ mV/*ms* |
| Inhibitory | 0.999982 | $0.612 \times 10^{-4}$ mV/*ms* |

of neurons comprising of different neuron classes with varying parameters.

**Figure 10** illustrates an equivalent simulation for 20 different neuron classes between BMTK and Loihi indicating that Loihi is capable of emulating BMTK results in spite of varying parameters. Here, we found an average correlation of 0.99985 with an RMSE of $0.57 \times 10^{-4}$ mV/*ms* (**Table 4**). This also validates the fact that the calibration of parameters for a single neuron done earlier is valid.

The scatter plots in **Figure 11** capture the range of the parameters—**Figure 11(A)** $C_m$ vs. $\tau_v$ and **Figure 11(B)** $I_{bias}$ vs. $\tau_v$, in the (E)xcitatory and (I)nhibitory classes used for the simulations. The size of the markers represents RMSE errors for those models, with ranges as indicated on the legend. This lays the foundation for building more complicated networks encompassing different neuron classes.
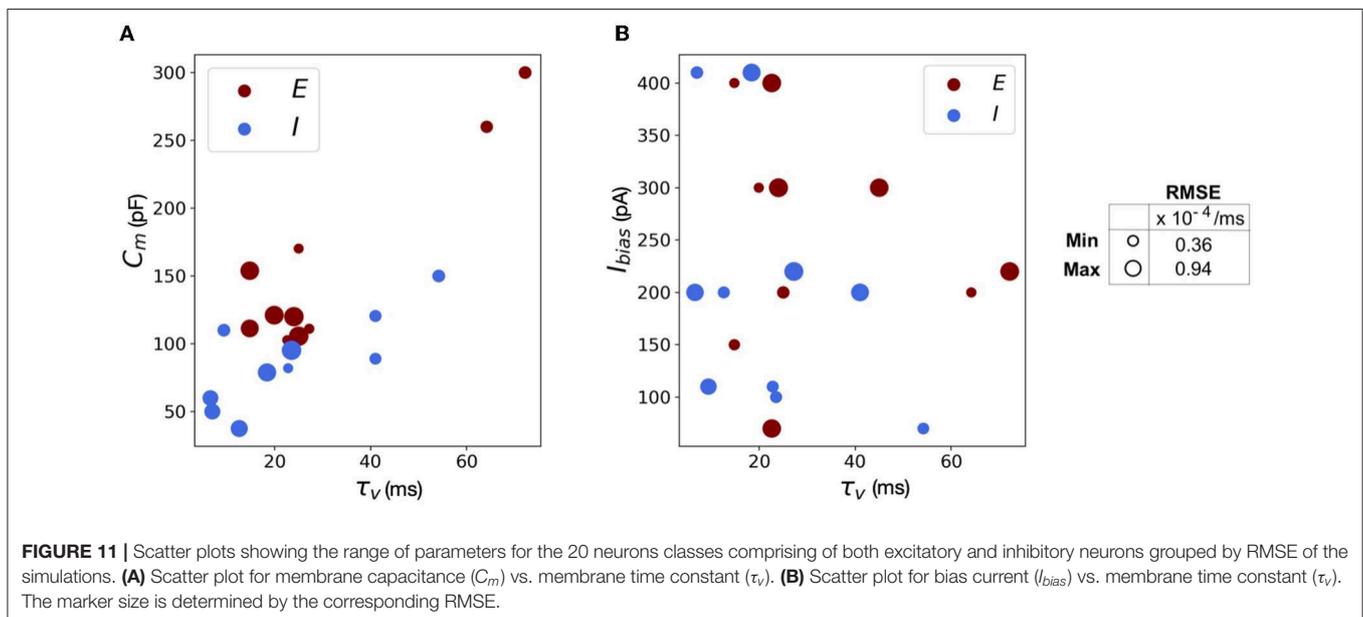
We reiterate here that Loihi imposes certain bit constraints on the parameters. For instance, membrane potential threshold ranges from 0 to $\pm 2^{23}$, membrane time constant allows 0 to $2^{12}$ bits. The membrane capacitance is integrated with bias current (Equation 18) with bias mantissa allowed a range between $[-2^{12}, 2^{12}]$ and bias exponent a range between $[0, 7]$. Thus, a good range of parameters can be mapped well into Loihi and a limit to the "exactness" can be attributed to the low-fixed-precision nature of Loihi as most state and configuration variables are in the range of 8–24 bits.

## 5. CONCLUSION AND FUTURE WORK

Inspired by the brain, neuromorphic computing holds great potential in tackling tasks with extremely low power and high efficiency. Many large-scale efforts including the TrueNorth, SpiNNaker and BrainScaleS have been demonstrated as a tool for neural simulations, each replete with its own strengths



**FIGURE 11** | Scatter plots showing the range of parameters for the 20 neurons classes comprising of both excitatory and inhibitory neurons grouped by RMSE of the simulations. **(A)** Scatter plot for membrane capacitance ($C_m$) vs. membrane time constant ($\tau_v$). **(B)** Scatter plot for bias current ($I_{bias}$) vs. membrane time constant ($\tau_v$). The marker size is determined by the corresponding RMSE.

and constraints. Fabricated with Intel's 14 nm technology, Loihi is a forward-looking and continuously evolving state-of-the-art architecture for modeling spiking neural networks in silicon. As opposed to its predecessors, Loihi encompasses a wide range of novel features such as hierarchical connectivity, dendritic compartments, synaptic delays and programming synaptic learning rule. These features, together with solid SDK support by Intel, and a growing research community, make Loihi an effective platform to explore a wealth of neuromorphic features in more detail than before.

In this work, we have demonstrated that Loihi is capable of replicating the continuous dynamics of point neuronal models with high degree of precision and does so with much greater efficiency in terms of time and energy. The work comes with its challenges as simulations built on the conventional chips cannot be trivially mapped to the neuromorphic platform as its architecture differs remarkably from the conventional hardware. Classical simulations from the Brain Modeling Toolkit (BMTK) developed by the Allen Institute of Brain Science (AIBS) serves as the foundation of our neuromorphic validation.

For comparison between the conventional and the neuromorphic platforms, we use both qualitative and quantitative measures. It can be seen that Loihi replicates BMTK very closely in terms of both membrane potential and current, the two state variables on which the Loihi LIF model

evolves. We use different validation methods and quantitative measures to assess the equivalence and identify sets of parameters which maximize precision while retaining high performance levels. Furthermore, simulation results indicate Loihi is highly efficient in terms of speed and scalability as compared to BMTK.

This work demonstrates that classical simulations based on Generalized Leaky Integrate-and-Fire (GLIF) point neuronal models can be successfully replicated on Loihi with a reasonable degree of precision.

Our future work is motivated by runtime performance comparisons for larger networks between the two platforms. As Loihi and BMTK are based on very different hardware systems that follow distinct dynamics and network-setup regimes, we use the runtime of the simulations to compare the performance of these implementations. As has been mentioned in the introduction, performance of Loihi far exceeds that of BMTK. **Figure 12** compares the runtime of Loihi and BMTK, for running a network of randomly connected neurons with the same parameters. The network consists of excitatory and inhibitory neurons in a 1:1 ratio driven by bias current, with connection probability set at 0.1.

As can be seen from **Figure 12** and **Table 5**, Loihi easily scales up to larger network sizes with a minuscule rise in runtime
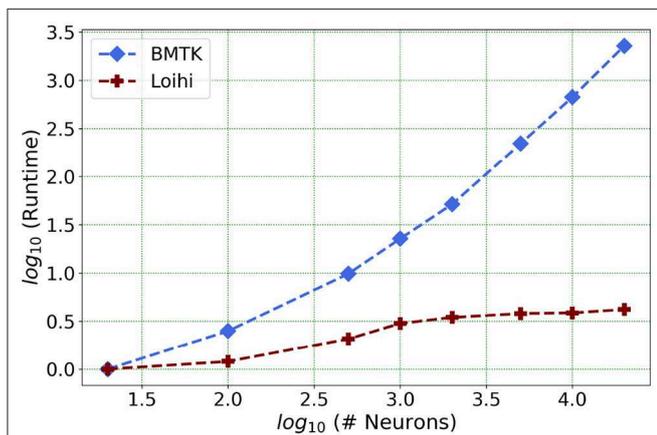


**FIGURE 12 |** Performance comparison between BMTK and Loihi for network sizes ranging from 1 to 20,000 for the simulation of 500 *ms* of dynamics. The values for each curve are scaled by the respective smallest runtime. The Loihi runtime units are in "*milliseconds*" and BMTK runtime is in "*seconds*".
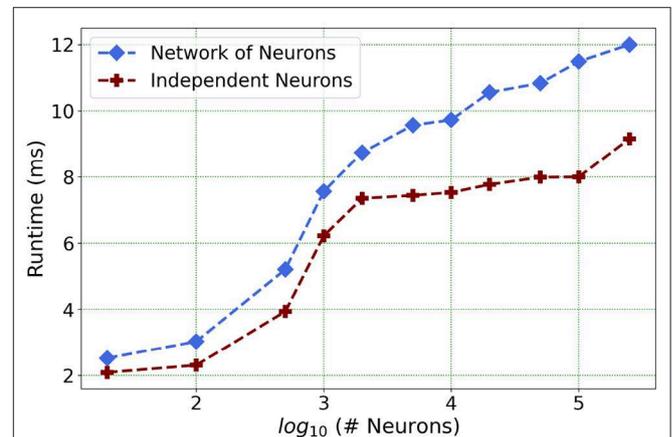


**FIGURE 13 |** Loihi runtime for a network of upto 250K neurons for the simulation of 500 *ms* of dynamics.

**TABLE 5 |** Simulation runtime in Loihi and BMTK.

| Network size | Loihi time (ms) | BMTK time (s) |
| --- | --- | --- |
| 20 | 2.52 | 0.12 |
| 100 | 3.03 | 0.3 |
| 500 | 5.21 | 1.13 |
| 1,000 | 7.56 | 2.72 |
| 5,000 | 9.57 | 26.47 |
| 10,000 | 9.73 | 80.45 |

**TABLE 6 |** Simulation runtime in Loihi for independent neurons vs. connected network.

| Network size | Connected network (ms) | Independent neurons (ms) |
| --- | --- | --- |
| 20 | 2.52 | 2.09 |
| 100 | 3.03 | 2.31 |
| 500 | 5.21 | 3.94 |
| 1,000 | 7.56 | 6.22 |
| 5,000 | 9.57 | 7.35 |
| 10,000 | 9.73 | 7.53 |
| 50,000 | 10.84 | 7.98 |
| 100,000 | 11.49 | 8.00 |
| 250,000 | 11.98 | 9.16 |

whereas for BMTK the increase is quite rapid. While both seem to exhibit a power-law scaling (string line on this graph), Loihi's scaling power is much smaller. It is also worth noting here that for Loihi the unit for the runtime are in "*milliseconds*" whereas for BMTK they are in "*seconds*". Here we stop at 20,000 neurons as it can be inferred from the graph that increasing the network size would increase the time cost for BMTK substantially.

Furthermore, following the above outcome, we extend our network size *in Loihi only* to 250K neurons in order to investigate what potential Loihi holds to execute the final goal of simulating about ∼250,000 neurons with ∼500M synapses in the future, a simulation scale comprising much of the experimentally observed dynamics in the mouse visual cortex available to the AIBS. We record our observations for a randomly connected network of neurons as well as an independent set of unconnected neurons. From **Figure 13** and **Table 6**, we can infer that the runtime remains consistent with the above result, with the independent set of neurons completing the simulation marginally faster.

This shows that Loihi performs well for connected networks, setting the stage for our main aim for neural simulations. Additionally, it also works well for independent set of neurons which contribute to solutions of problems that require on-chip parameter and meta-parameter searches, e.g., for Evolutionary Programming (Schuman et al., 2020).

We do not asses the state-based cost for these networks as their large sizes require multi-chip simulations which we expect to be better supported on Loihi 2 (Intel, 2022). Furthermore, other research groups have firmly established that we cannot expect exact replication of subthreshold network states between simulators except for few very simple small networks (van Albada et al., 2018; Crook et al., 2020). Thus, on the network level we need to develop cost functions that capture appropriate network activity details on different scales (e.g., average spike rates and correlations on the coarsest levels, as in van Albada et al., 2018).

In closing, we want to highlight that with the advent of Loihi 2 (Intel, 2022), we aim to address the limitations of the larger networks and carry out the next steps of our work in this new hardware. We are planning to investigate the full GLIF dynamics as we would have better support for more complex network topology and spiking dynamics. In addition, we hope to implement a connected network of 250K neurons with specific synaptic variables as available in the AIBS dataset. We also plan to investigate the control and performance of temporal precision choices. Till date, our limited conclusion for these cases is that ∼ 1*ms* timestep is sufficient. This need not generalize to networks in which other precision may be needed, with corresponding tradeoffs to changes in the parameters. We intend to explore this question further. Lastly, we are working on performing a sensitivity analysis on the GLIF parameters to assess the robustness of the model.

## DATA AVAILABILITY STATEMENT

The original contribution in the study are included in the article. The datasets used are available in https://github.com/srijanie03/bmtk_loihi_utils. Further inquiries can be directed to the corresponding author/s.

## AUTHOR CONTRIBUTIONS

SD and AD contributed to conception and design of the study. SD performed the simulations and analysis and wrote the first draft of the manuscript. AD wrote parts of the manuscript, edited, and reviewed. All authors contributed to manuscript revision, read, and approved the submitted version.

## FUNDING

## ACKNOWLEDGMENTS

## REFERENCES

AIBS (2020). *Allen Brain Atlas*. Available online at: https://celltypes.brain-map.org/ (accessed February 22, 2022).

Akopyan, F., Sawada, J., Cassidy, A., Alvarez-Icaza, R., Arthur, J., Merolla, P., et al. (2015). Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE Trans. Comput. Aided Design Integr. Circ. Syst.* 34, 1537–1557. doi: 10.1109/TCAD.2015.2474396

Benjamin, B. V., Gao, P., McQuinn, E., Choudhary, S., Chandrasekaran, A. R., Bussat, J.-M., et al. (2014). Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations. *Proc. IEEE* 102, 699–716. doi: 10.1109/JPROC.2014.2313565

Bhuiyan, M. A., Nallamuthu, A., Smith, M. C., and Pallipuram, V. K. (2010). "Optimization and performance study of large-scale biological networks for reconfigurable computing," in *2010 Fourth International Workshop on High-Performance Reconfigurable Computing Technology and Applications (HPRCTA)* (New Orleans, LA), 1–9. doi: 10.1109/HPRCTA.2010.5670796

Brette, R., Lilith, M., Carnevale, T., Hines, M., Beeman, D., Bower, J., et al. (2008). Simulation of networks of spiking neurons: a review of tools and strategies. *J. Comput. Neurosci.* 23, 349–398. doi: 10.1007/s10827-007-0038-6

CIRC and WSU (2021). *What Is Kamiak?* Available online at: https://hpc.wsu.edu/kamiak-hpc/what-is-kamiak/ (accessed February 22, 2022).

Crook, S. M., Davison, A. P., McDougal, R. A., and Plesser, H. E. (2020). Editorial: reproducibility and rigour in computational neuroscience. *Front. Neuroinform.* 14, 23. doi: 10.3389/fninf.2020.00023

Dai, K., Gratiy, S. L., Billeh, Y. N., Xu, R., Cai, B., Cain, N., et al. (2020). Brain modeling toolkit: an open source software suite for multiscale modeling of brain circuits. *PLoS Comput. Biol.* 16, e1008386. doi: 10.1371/journal.pcbi.1008386

Davies, M., Srinivasa, N., Lin, T. H., Chinya, G., Cao, Y., Choday, S. H., et al. (2018). Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99. doi: 10.1109/MM.2018.112130359

DeBole, M., Taba, B., Amir, A., Akopyan, F., Andreopoulos, A., Risk, W., et al. (2019). Truenorth: Accelerating from zero to 64 million neurons in 10 years. *Computer* 52, 20–29. doi: 10.1109/MC.2019.2903009

Dey, S. (2022). *BMTK-Loihi Data*. Available online at: https://github.com/srijanie03/bmtk_loihi_utils (accessed February 22, 2022).

Gerstner, W., and Kistler, W. M. (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press. doi: 10.1017/CBO9780511815706

Grübl, A., Billaudelle, S., Cramer, B., Karasenko, V., and Schemmel, J. (2020). Verification and design methods for the brainscales neuromorphic hardware system. *J. Signal Process. Syst.* 92, 1277–1292. doi: 10.1007/s11265-020-01558-7

Gutzen, R., von Papen, M., Trensch, G., Quaglio, P., Grün, S., and Denker, M. (2018). Reproducible neural network simulations: statistical methods for model validation on the level of network activity data. *Front. Neuroinform.* 12, 90. doi: 10.3389/fninf.2018.00090

Herz, A. V. M., Gollisch, T., Machens, C. K., and Jaeger, D. (2006). Modeling single-neuron dynamics and computations: a balance of detail and abstraction. *Science* 314, 80–85. doi: 10.1126/science.1127240

Hopkins, M., and Furber, S. (2015). Accuracy and efficiency in fixed-point neural ODE solvers. *Neural Comput.* 27, 2148–2182. doi: 10.1162/NECO_a_00772

Insel, T. R., Landis, S. C., and Collins, F. S. (2013). The NIH BRAIN initiative. *Science* 340, 687–688. doi: 10.1126/science.1239276

Intel (2022). *Intel Lab's Loihi 2 Chip*. Technical report, Intel Corporation.

Jin, X., Furber, S. B., and Woods, J. V. (2008). "Efficient modelling of spiking neural networks on a scalable chip multiprocessor," in *2008 IEEE International Joint Conference on Neural Networks* (Hong Kong: IEEE World Congress on Computational Intelligence), 2812–2819. doi: 10.1109/IJCNN.2008.4634194

Khan, M. M., Lester, D. R., Plana, L. A., Rast, A., Jin, X., Painkras, E., et al. (2008). "Spinnaker: mapping neural networks onto a massively-parallel chip multiprocessor," in *2008 IEEE International Joint Conference on Neural Networks* (Hong Kong: IEEE World Congress on Computational Intelligence), 2849–2856. doi: 10.1109/IJCNN.2008.4634199

Knight, J. C., and Furber, S. (2016). Synapse-centric mapping of cortical models to the spinnaker neuromorphic architecture. *Frontiers in Neuroscience* 10. doi: 10.3389/fnins.2016.00420

Kunkel, S., Morrison, A., Weidel, P., Eppler, J. M., Sinha, A., Schenck, W., et al. (2017). *Nest 2.12.10*. Zenodo.

Lein, E., Hawrylycz, M., Ao, N., Ayres, M., Bensinger, A., Bernard, A., et al. (2007). Genome-wide atlas of gene expression in the adult mouse brain. *Nature* 445, 168–176. doi: 10.1038/nature05453

Linssen, C., Lepperød, M. E., Mitchell, J., Pronold, J., Eppler, J. M., Keup, C., et al. (2018). NEST 2.16.0. *Zenodo*. doi: 10.5281/zenodo.1400175

Löhr, M. P. R., Jarvers, C., and Neumann, H. (2020). "Complex neuron dynamics on the IBM truenorth neurosynaptic system," in *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)* (Genao), 113–117. doi: 10.1109/AICAS48895.2020.9073903

Markram, H., Meier, K., Lippert, T., Grillner, S., Frackowiak, R., Dehaene, S., et al. (2011). Introducing the human brain project. *Proc. Comput. Sci.* 7, 39–42. doi: 10.1016/j.procs.2011.12.015

Michaelis, C., Lehr, A. B., Oed, W., and Tetzlaff, C. (2021). Brian2Loihi: an emulator for the neuromorphic chip loihi using the spiking neural network simulator brian. *arXiv preprint arXiv:2109.12308*. doi: 10.48550/ARXIV.2109.12308

Mikaitis, M., Lester, D., Shang, D., Furber, S., Liu, G., Garside, J., et al. (2018). "Approximate fixed-point elementary function accelerator for the spinnaker-2 neuromorphic chip," in *2018 IEEE 25th Symposium on Computer Arithmetic (ARITH)* (Amherst, MA). doi: 10.1109/ARITH.2018.8464785

Moradi, S., Qiao, N., Stefanini, F., and Indiveri, G. (2017). A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs). *IEEE Trans. Biomed. Circ. Syst.* 12, 106–122. doi: 10.1109/TBCAS.2017.2759700

Nandi, A., Chartrand, T., Geit, W. V., Buchin, A., Yao, Z., Lee, S. Y., et al. (2020). Single-neuron models linking electrophysiology, morphology and transcriptomics across cortical cell types. *bioRxiv [Preprints]*. doi: 10.1101/2020.04.09.030239

Nawrocki, R. A., Voyles, R. M., and Shaheen, S. E. (2016). A mini review of neuromorphic architectures and implementations. *IEEE Trans. Electron Dev.* 63, 3819–3829. doi: 10.1109/TED.2016.2598413

Neckar, A., Fok, S., Benjamin, B. V., Stewart, T. C., Oza, N. N., Voelker, A. R., et al. (2019). Braindrop: a mixed-signal neuromorphic architecture with a dynamical systems-based programming model. *Proc. IEEE* 107, 144–164. doi: 10.1109/JPROC.2018.2881432

Ou, Q.-F., Xiong, B.-S., Yu, L., Wen, J., Wang, L., and Tong, Y. (2020). In-memory logic operations and neuromorphic computing in non-volatile random access memory. *Materials* 13, 3532. doi: 10.3390/ma13163532

Pehle, C., Billaudelle, S., Cramer, B., Kaiser, J., Schreiber, K., Stradmann, Y., et al. (2022). The brainscaleS-2 accelerated neuromorphic system with hybrid plasticity. *Front. Neurosci.* 16, 795876. doi: 10.3389/fnins.2022.795876

Rhodes, O., Peres, L., Rowley, A., Gait, A., Plana, L., Brenninkmeijer, C. Y. A., et al. (2019). Real-time cortical simulation on neuromorphic hardware. *Philos. Trans. Ser. A Math. Phys. Eng. Sci.* 378:20190160. doi: 10.1098/rsta.2019.0160

Rossant, C., Goodman, D., Platkiewicz, J., and Brette, R. (2010). Automatic fitting of spiking neuron models to electrophysiological recordings. *Front. Neuroinform.* 4:2. doi: 10.3389/neuro.11.002.2010

Rotter, S., and Diesmann, M. (1999). Exact digital simulation of time-invariant linear systems with applications to neuronal modeling. *Biol. Cybernet.* 81, 381–402. doi: 10.1007/s004220050570

Roy, K., Jaiswal, A., and Panda, P. (2019). Towards spike-based machine intelligence with neuromorphic computing. *Nature* 575, 607–617. doi: 10.1038/s41586-019-1677-2

Schuman, C. D., Mitchell, J. P., Patton, R. M., Potok, T. E., and Plank, J. S. (2020). "Evolutionary optimization for neuromorphic systems," in *Proceedings of the Neuro-inspired Computational Elements Workshop, NICE '20* (New York, NY: Association for Computing Machinery), 1–9. doi: 10.1145/3381755.3381758

Sharp, T., and Furber, S. (2013). "Correctness and performance of the spinnaker architecture," in *The 2013 International Joint Conference on Neural Networks (IJCNN)* (Dallas, TX), 1–8. doi: 10.1109/IJCNN.2013.6706988

Teeter, C., Iyer, R., Menon, V., Gouwens, N., Feng, D., Berg, J., et al. (2018). Generalized leaky integrate-and-fire models classify multiple neuron types. *Nat. Commun.* 9, 709. doi: 10.1038/s41467-017-02717-4

Thakur, C. S., Molin, J. L., Cauwenberghs, G., Indiveri, G., Kumar, K., Qiao, N., et al. (2018). Large-scale neuromorphic spiking array processors: a quest to mimic the brain. *Front. Neurosci.* 12, 891. doi: 10.3389/fnins.2018.00891

Trensch, G., Gutzen, R., Blundell, I., Denker, M., and Morrison, A. (2018). Rigorous neural network simulations: a model substantiation methodology for increasing the correctness of simulation results in the absence of experimental validation data. *Front. Neuroinform.* 12, 81. doi: 10.3389/fninf.2018.00081

van Albada, S. J., Rowley, A. G., Senk, J., Hopkins, M., Schmidt, M., Stokes, A. B., et al. (2018). Performance comparison of the digital neuromorphic hardware SpiNNaker and the neural network simulation software NEST for a full-scale cortical microcircuit model. *Front. Neurosci.* 12, 291. doi: 10.3389/fnins.2018.00291

Wang, Q., Ding, S.-L., Li, Y., Royall, J., Feng, D., Lesnar, P., et al. (2020). The allen mouse brain common coordinate framework: a 3D reference atlas. *Cell* 181, 936.e20–953.e20. doi: 10.1016/j.cell.2020.04.007

WikiCommons (2018). *Core Top-Level Microarchitecture*. Available online at: https://commons.wikimedia.org/wiki/File:Core_Top-Level_Microarchitecture.png (accessed March 17, 2022).

Winnubst, J., Bas, E., Ferreira, T. A., Wu, Z., Economo, M. N., Edson, P., et al. (2019). Reconstruction of 1,000 projection neurons reveals new cell types

and organization of long-range connectivity in the mouse brain. *Cell* 179, 268.e13–281.e13. doi: 10.1016/j.cell.2019.07.042