nature computational science

Supplementary information https://doi.org/10.1038/s43588-024-00674-9

Network model with internal complexity bridges artificial intelligence and neuroscience

In the format provided by the
authors and unedited

Supplementary Information for Network Model with Internal Complexity Bridges Artificial Intelligence and Neuroscience

Contents

1 Proof of Theorem 1

The equations corresponding to LIF model and HH model are shown as follows. C is the capacitance, $I(t)$ is the current input, $u(t)$ is the neuronal potential, $g=\frac{1}{R}$ is the conductance of LIF model, u_{rest} is the resting potential of LIF model and $g_{Na}(t), g_K(t), g_L$ are the conductances of corresponding ion channels with potential reference values E_{Na}, E_K, E_L.

$$
\begin{aligned}
C\frac{\mathrm{d}u(t)}{\mathrm{d}t} &= -g(u(t)-u_{rest})+I(t), \\
C\frac{\mathrm{d}u(t)}{\mathrm{d}t} &= -g_{Na}(t)(u(t)-E_{Na})-g_K(t)(u(t)-E_K)-g_L(u(t)-E_L)+I(t).
\end{aligned} \tag{1}
$$

We first let 3 solutions $u_1(t), u_2(t), u_3(t)$ of LIF equations satisfy the following form. R, C are any given fixed value of resistance and capacitance and $I_1(t), I_2(t), I_3(t)$ are the current inputs for 3 LIF in this case.

$$
\begin{aligned}
C\frac{\mathrm{d}u_1(t)}{\mathrm{d}t} &= -\frac{1}{R}(u_1(t)-E_{Na})+I_1(t), \\
C\frac{\mathrm{d}u_2(t)}{\mathrm{d}t} &= -\frac{1}{R}(u_2(t)-E_K)+I_2(t), \\
C\frac{\mathrm{d}u_3(t)}{\mathrm{d}t} &= -\frac{1}{R}(u_3(t)-E_L)+I_3(t).
\end{aligned} \tag{2}
$$

Then we use a function $f: u \to f(u)$ to represent the nonlinear response to the original HH neuron contributed by the output of a particular LIF neuron. Because we can hardly guarantee the output is linear in common cases, we adopt a general function output f to represent. The HH neuron response is assumed to be $u(t)=f_1(u_1)+f_2(u_2)+f_3(u_3)+u_s(t)$, where $u_s(t)$ is a remainder temporarily unknown here, and satisfies the constraints of HH equation, so we have the relationship below.

$$
\begin{aligned}
C\frac{\mathrm{d}u(t)}{\mathrm{d}t} &= -\frac{1}{R}\frac{df_1(u_1)}{du_1}(u_1(t)-E_{Na})-\frac{1}{R}\frac{df_2(u_2)}{du_2}(u_2(t)-E_K)-\frac{1}{R}\frac{df_3(u_3)}{du_3}(u_3(t)-E_L)+C\frac{\mathrm{d}u_s(t)}{\mathrm{d}t} \\
&\quad+\left(\frac{df_1(u_1)}{du_1}I_1(t)+\frac{df_2(u_2)}{du_2}I_2(t)+\frac{df_3(u_3)}{du_3}I_3(t)\right), \\
&= -\frac{1}{R}\left(\frac{df_1(u_1)}{du_1}u_1(t)+\frac{df_2(u_2)}{du_2}u_2(t)+\frac{df_3(u_3)}{du_3}u_3(t)\right)+C\frac{\mathrm{d}u_s(t)}{\mathrm{d}t}+\frac{1}{R}\left(\frac{df_1(u_1)}{du_1}E_{Na}+\frac{df_2(u_2)}{du_2}E_K+\frac{df_3(u_3)}{du_3}E_L\right) \\
&\quad+\left(\frac{df_1(u_1)}{du_1}I_1(t)+\frac{df_2(u_2)}{du_2}I_2(t)+\frac{df_3(u_3)}{du_3}I_3(t)\right), \\
&= -g_{Na}(t)(u(t)-E_{Na})-g_K(t)(u(t)-E_K)-g_L(u(t)-E_L)+I(t), \\
&= -(g_{Na}(t)+g_K(t)+g_L)\,u(t)+(g_{Na}(t)E_{Na}+g_K(t)E_K+g_LE_L)+I(t).
\end{aligned} \tag{3}
$$

We observe the above results and find the following equations that keep the equivalence.

$$
\begin{aligned}
\frac{df_1(u_1)}{du_1} &= g_{Na}(t)R, \\
\frac{df_2(u_2)}{du_2} &= g_K(t)R, \\
\frac{df_3(u_3)}{du_3} &= g_LR, \\
\frac{1}{R}I(t) &= g_{Na}(t)I_1(t)+g_K(t)I_2(t)+g_LI_3(t), \\
C\frac{\mathrm{d}u_s(t)}{\mathrm{d}t} &= -(g_{Na}(t)+g_K(t)+g_L)u_s(t)+(g_{Na}(t)u_1(t)+g_K(t)u_2(t)+g_Lu_3(t)) \\
&\quad-(g_{Na}(t)+g_K(t)+g_L)(f_1(u_1(t))+f_2(u_2(t))+f_3(u_3(t))).
\end{aligned} \tag{4}
$$

Ion channel current $I_1(t), I_2(t), I_3(t)$ can be chosen among any sets of solutions satisfying the constraint above and we take $I_1(t)=I_2(t)=I_3(t)=I_0(t)=\frac{1}{(g_{Na}(t)+g_K(t)+g_L)R}I(t)$ for the sake of convenience. Then we have the following equations,

$$
\begin{aligned}
g_s(t) &= g_{Na}(t)+g_K(t)+g_L, \\
I_s(t) &= -g_s(t)(f_1(u_1(t))+f_2(u_2(t))+f_3(u_3(t)))+(g_{Na}(t)u_1(t)+g_K(t)u_2(t)+g_Lu_3(t)), \\
I_1(t) &= I_2(t)=I_3(t)=I_0(t)=\frac{1}{g_s(t)R}I(t), \\
C\frac{\mathrm{d}u_s(t)}{\mathrm{d}t} &= -g_s(t)u_s(t)+I_s(t).
\end{aligned} \tag{5}
$$

Therefore, we derive the theoretical equivalence in Theorem 1 and subsection "Theoretical equivalence of tv-LIF2HH model and HH model" of Methods.

1) The definitive formula of ion-channel LIF potential u_1, u_2, u_3 satisfies the LIF equation is as follows:

$$\begin{cases} g_s(t) = g_{Na}(t) + g_K(t) + g_L, \\ I_0(t) = \frac{1}{g_s(t)R} I(t), \\ C\frac{\mathrm{d}u_1(t)}{\mathrm{d}t} = -\frac{1}{R}(u_1(t) - E_{Na}) + I_0(t), \\ C\frac{\mathrm{d}u_2(t)}{\mathrm{d}t} = -\frac{1}{R}(u_2(t) - E_K) + I_0(t), \\ C\frac{\mathrm{d}u_3(t)}{\mathrm{d}t} = -\frac{1}{R}(u_3(t) - E_L) + I_0(t). \end{cases} \tag{6}$$

The theoretical solutions of u_1, u_2, u_3 with constants a_1, a_2, a_3 which rely on the initial value of the system are as follows:

$$\begin{cases} a(t) & = \int_0^t \frac{1}{C} e^{\frac{1}{CR}\tau} \frac{1}{g_s(\tau)R} I(\tau) d\tau, \\ u_1(t) & = (a_1 + a(t)) e^{-\frac{1}{CR}t} + E_{Na} \\ & = (a_1 + \int_0^t \frac{1}{C} e^{\frac{1}{CR}\tau} \frac{1}{g_s(\tau)R} I(\tau) d\tau) e^{-\frac{1}{CR}t} + E_{Na}, \\ u_2(t) & = (a_2 + a(t)) e^{-\frac{1}{CR}t} + E_K \\ & = (a_2 + \int_0^t \frac{1}{C} e^{\frac{1}{CR}\tau} \frac{1}{g_s(\tau)R} I(\tau) d\tau) e^{-\frac{1}{CR}t} + E_K, \\ u_3(t) & = (a_3 + a(t)) e^{-\frac{1}{CR}t} + E_L \\ & = (a_3 + \int_0^t \frac{1}{C} e^{\frac{1}{CR}\tau} \frac{1}{g_s(\tau)R} I(\tau) d\tau) e^{-\frac{1}{CR}t} + E_L. \end{cases} \tag{7}$$

2) The definitive formula of ion-channel LIF response pattern f_1, f_2, f_3 is as follows:

$$\begin{cases} f_1(u_1(t)) & = \int_0^t g_{Na}(\tau) R \frac{du_1(\tau)}{d\tau} d\tau \\ & = \int_0^t (-\frac{1}{C}a_1 - \frac{1}{C}a(\tau) + R\frac{da(\tau)}{d\tau}) e^{-\frac{1}{CR}\tau} g_{Na}(\tau) d\tau \\ & = \int_0^t (-\frac{1}{C}a_1 - \frac{1}{C}\int_0^\tau \frac{1}{C} e^{\frac{1}{CR}x} \frac{1}{g_s(x)R} I(x) dx + \frac{R}{C} e^{\frac{1}{CR}\tau} \frac{1}{g_s(\tau)R} I(\tau)) e^{-\frac{1}{CR}\tau} g_{Na}(\tau) d\tau, \\ f_2(u_2(t)) & = \int_0^t g_K(\tau) R \frac{du_2(\tau)}{d\tau} d\tau \\ & = \int_0^t (-\frac{1}{C}a_2 - \frac{1}{C}a(\tau) + R\frac{da(\tau)}{d\tau}) e^{-\frac{1}{CR}\tau} g_K(\tau) d\tau \\ & = \int_0^t (-\frac{1}{C}a_2 - \frac{1}{C}\int_0^\tau \frac{1}{C} e^{\frac{1}{CR}x} \frac{1}{g_s(x)R} I(x) dx + \frac{R}{C} e^{\frac{1}{CR}\tau} \frac{1}{g_s(\tau)R} I(\tau)) e^{-\frac{1}{CR}\tau} g_K(\tau) d\tau, \\ f_3(u_3(t)) & = \int_0^t g_L R \frac{du_3(\tau)}{d\tau} d\tau \\ & = \int_0^t (-\frac{1}{C}a_3 - \frac{1}{C}a(\tau) + R\frac{da(\tau)}{d\tau}) e^{-\frac{1}{CR}\tau} g_L d\tau \\ & = \int_0^t (-\frac{1}{C}a_3 - \frac{1}{C}\int_0^\tau \frac{1}{C} e^{\frac{1}{CR}x} \frac{1}{g_s(x)R} I(x) dx + \frac{R}{C} e^{\frac{1}{CR}\tau} \frac{1}{g_s(\tau)R} I(\tau)) e^{-\frac{1}{CR}\tau} g_L d\tau. \end{cases} \tag{8}$$

The response pattern of each ion-channel f_i satisfies LIF equation with time-variant conductance as follows:

$$C\frac{df_i(u_i(t))}{dt} = g_i(t) RC \frac{du_i}{dt} = -g_i(t)(u_i(t) - E_i) + g_i(t) R I_0(t). \tag{9}$$

3) The definitive formula of residual-response LIF potential u_s satisfies the LIF equation with time-variant conductance as follows:

$$\begin{cases} C\frac{\mathrm{d}u_s(t)}{\mathrm{d}t} & = -g_s(t) u_s(t) + I_s(t), \\ g_s(t) & = g_{Na}(t) + g_K(t) + g_L, \\ I_s(t) & = -g_s(t)(f_1(u_1(t)) + f_2(u_2(t)) + f_3(u_3(t))) \\ & \quad + (g_{Na}(t) u_1(t) + g_K(t) u_2(t) + g_L u_3(t)). \end{cases} \tag{10}$$

The theoretical solution of residual-response LIF potential u_s with constant a_s which relies on the initial value of the system is as follows:

$$\begin{cases} P(t) & = \int_0^t \frac{1}{C} g_s(\tau) d\tau, \\ u_s(t) & = a_s e^{-P(t)} + e^{-P(t)} \int_0^t I_s(\tau) e^{P(\tau)} d\tau \\ & = a_s e^{-\int_0^t \frac{1}{C} g_s(\tau) d\tau} + e^{-\int_0^t \frac{1}{C} g_s(\tau) d\tau} \int_0^t I_s(\tau) e^{\int_0^\tau \frac{1}{C} g_s(x) dx} d\tau. \end{cases} \tag{11}$$

2 Supporting experiments of network equivalence

To support our equivalent findings, we conduct numerical simulation experiments between HH model and tv-LIF2HH model in networks. We pick a particular solution to carry out the experimental verification of the equivalence of HH and tv-LIF2HH sub-networks, in which the weights of tv-LIF2HH from all the neurons in the former group to the first three neurons in the posterior group are set to the same value of the weight between two HH neurons, and the weights inside tv-LIF2HH from the first three neurons to the fourth neuron in a group are set to 1. The networks in all the simulations follow these setups.

We construct a simple neural network to solve the XOR problem, which includes two input neurons and one output neuron, and the release threshold and the potential-to-spike encoding mechanisms (see Equation 12 in Methods) are introduced to HH neurons and tv-LIF2HH sub-networks for further illustration of network functionality. The HH neurons used in the network are standardized HH models from the referred paper[1]. The potential updates within the tv-LIF2HH sub-network follow the equations in the subsection "Theoretical equivalence of tv-LIF2HH model and HH model" in Methods. The current injection of the tv-LIF2HH sub-network follows the regulations demonstrated in the subsection "Equivalence between a single neuron with internal complexity and external complexity" in Results. This means current $\frac{1}{g_s(t)R} syn(u)$ lasting for Δt period is provided to 3 tv-LIF within the post-synaptic sub-network when the preceding sub-network spikes as a whole, where $syn(u)$ also lasting for Δt period is the corresponding current injection to the post-synaptic HH neuron in the HH network. Afterward, the network with fixed weights is given four types of XOR inputs ([0,0], [0,1], [1,0], [1,1], e.g. if the input is [0,1], then the first hidden neuron's input is 0, while the second neuron's input is $10\mu A$), and we measure and record the outputs of the network. Meanwhile, we build a tv-LIF2HH network with the equivalent structure as the HH network according to the previous deduction, and we simulate it with the same method as the HH network and compare their outputs. The total time of the simulation is 40 seconds, and the update time step is set to 0.001 seconds.

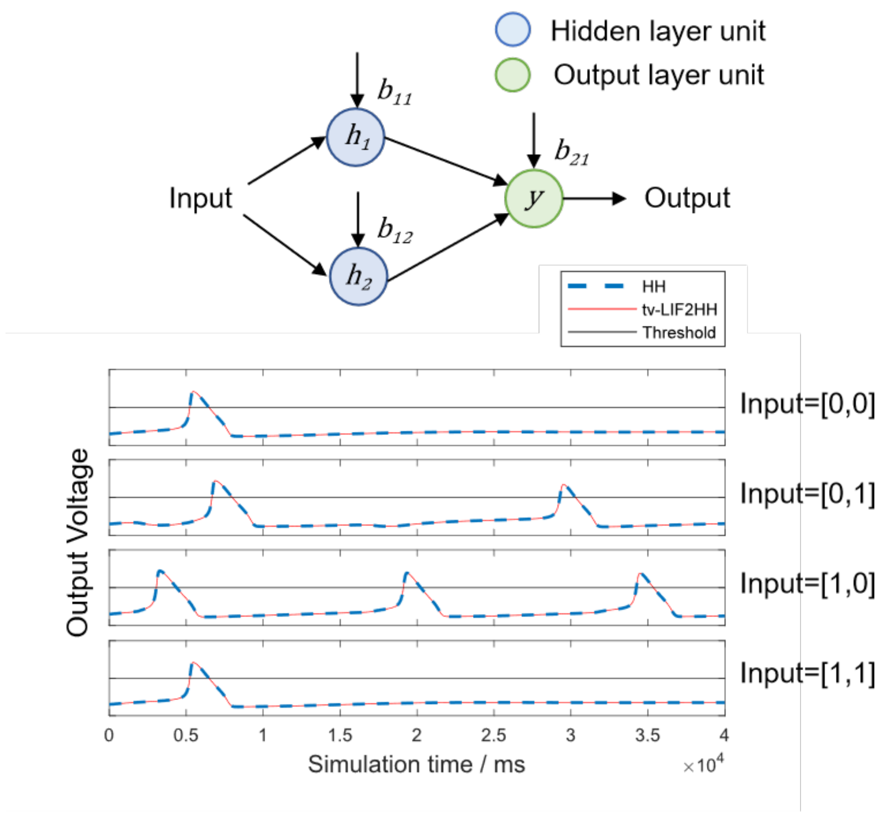


Fig.S2.1 | To verify the equivalence between HH and tv-LIF2HH model, we use a simple 2-layer perceptron in the context of the XOR problem as a supporting experiment. "h"s refer to the hidden layer units and "y" indicates the output layer unit. "b"s are the additional input currents to the units and we set them to 0 in this experiment. Inputs are separately given to the hidden neurons (e.g. if the input is [0,1], then the first hidden neuron's input is 0, while the second neuron's input is $10\mu A$). The whole XOR gate (all 3 neurons: h_1, h_2, y) is made up of the same kind of model (either HH or tv-LIF2HH). The outputs of the XOR gate (i.e., the membrane potentials of the output units) of HH neurons and tv-LIF2HH sub-networks are shown with different input states.

The results are shown in Fig.S2.1. Fig.S2.1 indicates that the tv-LIF2HH network and the HH network present the same feedback despite the types of inputs to the networks if we consider the iterative errors of simulation within a reasonable range. We also observe that the two networks both have an output peak with the input [0,0] and [1,1] and they have multiple output peaks with the input [0,1] and [1,0], which indicates that they can both solve the XOR problem.

Meanwhile, We build a fully-connected network with 128 HH neurons in the first layer and 10 HH neurons in the second layer to solve the image classification problem based on the MNIST dataset. We build a tv-LIF2HH network with the equivalent structure of the HH network and compare the membrane potentials of the neurons in the second layer. The current settings are the same as in the XOR gate experiment above. For each input digit image, all the pixels are first flattened to a vector, and then each hidden layer unit uses one kind of weighted sum of the pixels as its input. The simulation process lasts 40 seconds, while the update time step is still 0.001 seconds. The results are shown in Fig.S2.2.

Fig.S2.2 shows the results in a slightly larger scale network, where we randomly initialize the connection weights and use the samples from MNIST to trigger spikes. The neurons in the second layer of the tv-LIF2HH and HH network still show the same firing time of spikes with the same input, which indicates that the functional equivalence between tv-LIF2HH sub-networks and HH neurons exists regardless of the network scales. Moreover, each neuron in the second layer responds in different spike patterns to the specific network inputs as shown in Fig.S2.2, suggesting that two networks can extract various information from the inputs and they both own the capacity of image classification.

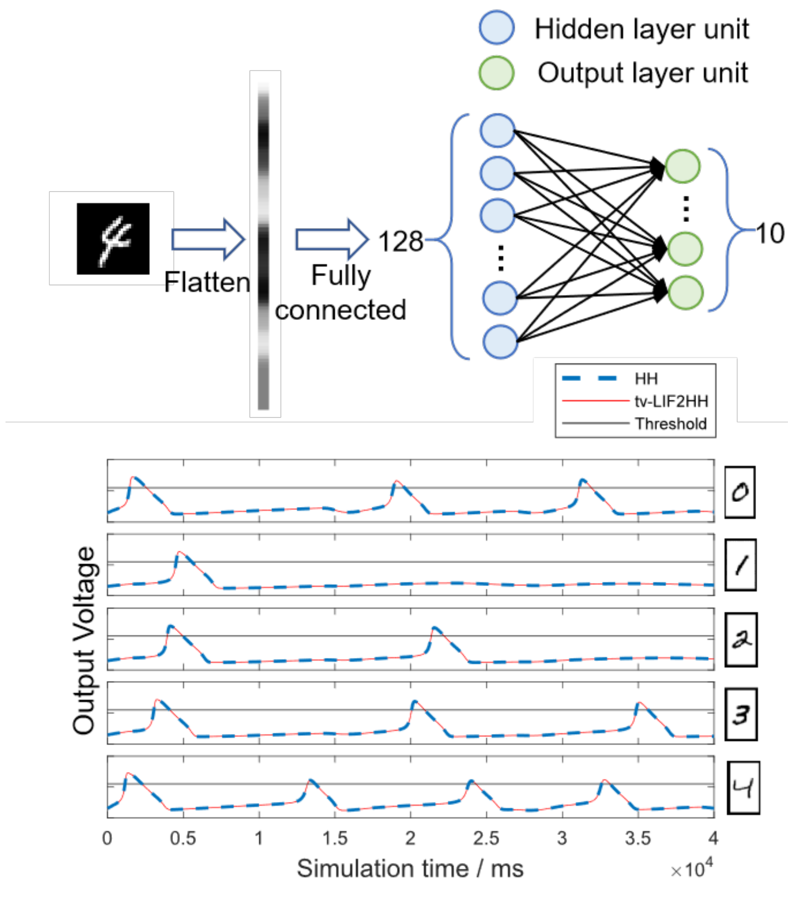


Fig.S2.2 | With randomly initialized MLP, we examine the output of tv-LIF2HH and HH networks with the same input sample. For each input digit image, all the pixels are first flattened to a vector, and then each hidden layer unit uses one kind of weighted sum of the pixels as its input. For illustration, we use an image of the number 4 as input, and we observe the outputs of the first five neurons in the output layer, which correspond to the classification 0 to 4. The output peaks are still totally matched. As in Fig.S2.1, the whole network is made up of the same kind of model (either HH or tv-LIF2HH).

3 Supplementary Tables

3.1 Quantitative performance and computational consumption of various models

The quantitative performance and computational consumption of different models further demonstrated the advantages of the "small model with internal complexity" (HH models) compared to the "big model with external complexity" (s-LIF2HH and 4×LIF). Take HH and s-LIF2HH for example, when outperforming s-LIF2HH in multi-task learning, reinforcement learning and robustness experiments, HH models use only 32.96% parameters of s-LIF2HH in fully-connected network, and 25.63% in convolutional network. For inference and backpropagation speed, HH models achieve 63.96% time of s-LIF2HH in fully-connected network and 46.75% in convolutional network. Meanwhile, as for the computational efficiency, HH models only need 33.74% of the s-LIF2HH models in fully-connected network and 56.94% in convolutional network (this estimate is based on that, the cost of addition is about a quarter of the cost of multiplication.). These representation results are consistent with our research on equivalence, indicating that the "small model with internal complexity" HH approaches do not consume substantially more resources compared to the "big model with external complexity", which is a breakthrough of the long-standing point of view.

Supplementary Table 1 | Quantitative comparative results of the multi-task learning experiment with various internal complexity.

Model Type	0% noise		10% noise		20% noise		30% noise	
	Acc1	Acc2	Acc1	Acc2	Acc1	Acc2	Acc1	Acc2
HH-fc	81.96±0.24	80.91±0.18	80.74±0.37	80.09±0.19	78.75±0.50	79.07±0.27	76.51±0.51	78.05±0.30
s-LIF2HH-fc	75.36±1.12	77.64±0.49	73.58±1.21	76.80±0.51	71.31±1.29	75.64±0.62	69.14±1.39	74.36±0.61
LIF-fc	64.19±0.44	72.27±0.23	61.86±0.73	71.69±0.27	58.17±0.79	70.18±0.35	54.32±1.04	68.88±0.36
4×LIF-fc	68.10±0.33	74.26±0.27	64.81±0.34	73.08±0.28	61.39±0.64	71.46±0.22	58.09±0.89	70.00±0.38
b-ANN	64.81±0.45	72.54±0.26	62.25±0.40	71.89±0.29	58.85±0.79	70.41±0.41	54.92±0.88	69.19±0.38

* The results are obtained from 20 repeated trials (mean ± std).

Supplementary Table 2 | Quantitative results of FLOPs within different networks under multi-task learning setups.

Model type	Add operations/M	Multiply operations/M
HH-fc	10.094	10.156
HH-conv	11.510	11.753
s-LIF2HH-fc	30.047	30.077
s-LIF2HH-conv	19.894	20.720
LIF-fc	9.994	10.002
LIF-conv	6.650	6.893
4×LIF-fc	39.977	40.008
4×LIF-conv	79.088	80.060

Supplementary Table 3 | Quantitative results of trainable parameters within different networks under multi-task learning setups.

Model type	Fully-connected networks	Convolutional networks
HH	690734	42580
s-LIF2HH	2095666	166104
LIF	690734	42580

Supplementary Table 4 | Quantitative results of time consumption of different networks in the inference and backpropagation process.

Model type	Inference/s	Inference+Backpropagation/s
HH-fc	7.07	8.36
HH-conv	16.23	18.18
s-LIF2HH-fc	10.31	13.07
s-LIF2HH-conv	29.55	38.07
LIF-fc	2.37	3.87
LIF-conv	5.91	8.31

3.2 Normalized mutual information values

Supplementary Table 5 | Quantitative comparative results of mutual information values of different models.

Model type	I(X,Z)	I(Z,Y)
s-LIF2HH	0.7926±0.0181	0.5968±0.0083
HH	0.3544±0.0054	0.6540±0.0036
LIF	0.4962±0.0099	0.4648±0.0055

* The results are obtained from 10 repeated trials (mean ± std).

3.3 Nomenclatures of different models

Supplementary Table 6 | Model nomenclatures and corresponding descriptions.

Model nomenclature	Description
LIF	Leaky-Integrate-And-Fire model
tv-LIF	Time-varied LIF model
4×LIF	LIF model with four times as many neurons
HH	Hodgkin-Huxley model
tv-LIF2HH	"HH model" constructed by four time-varied LIF, or time-varied LIF-to-HH model
s-LIF2HH	Simplified LIF-to-HH model
b-ANN	Binary ANN model, i.e. normal ANN model with binary output

3.4 Summary of parameters in different models

Supplementary Table 7 | Hyperparameter descriptions and selection methods.

Hyperparameters	Descriptions	Corresponding models	Selection methods
R	Neuronal resistance	LIF,4×LIF, s-LIF2HH	We set $R \in [1, 10]$ since it fits for all the experiments.
C	Neuronal capacitance	LIF, tv-LIF, 4×LIF, HH, tv-LFI2HH, s-LIF2HH	For simplicity, we set $C = 1$ in all the cases.
u_{th}	Threshold potential	LIF, 4×LIF, HH, s-LIF2HH, b-ANN	Corresponding to real neurons in simulation, while considering the range of network data flow in deep learning experiments.
u_{rest}	Resting potential	LIF, tv-LIF, 4×LIF, tv-LIF2HH, s-LIF2HH	We set $u_{rest} = 0$ in most cases, except that for tv-LIF2HH model, it is set according to HH model.
g_K, g_{Na}, g_L	Conductance	HH	We set them according to reference[1].
α, β	Channel parameters	HH	In simulation, we calculate them directly from membrane potential u, while in deep learning tasks, we set them randomly around the values in simulation.

* Hyperparameters are not trainable.

Supplementary Table 8 | Hyperparameters and trainable parameters for each model.

Model nomenclature	Hyperparameters: Simulation	Hyperparameters: Deep learning tasks	Trainable parameters
LIF	–	$R \in [1, 10], C = 1, u_{th} \in [0, 3], u_{rest} = 0$	W, b
tv-LIF	$C = 1, u_{rest} = 0$	–	–
4×LIF	–	$R \in [1, 10], C = 1, u_{th} \in [0, 3], u_{rest} = 0$	W, b
HH	$g_K = 36, g_{Na} = 120, g_L = 0.3, C = 1, u_{th} = 70$	$C = 1, u_{th} \in [0, 3], \alpha, \beta \in [0, 1]$	W, b
tv-LIF2HH	$C = 1, u_{1,rest} = 50, u_{2,rest} = -77, u_{3,rest} = -54.387, u_{s,rest} = 0$	–	–
s-LIF2HH	$R = 10, C = 1, u_{th} = 70, u_{rest} = 0$	$R \in [1, 10], C = 1, u_{th} \in [0, 3], u_{rest} = 0$	W_{inside}, W, b
b-ANN	–	$u_{th} \in [0, 3]$	W, b

* Trainable parameters are only trained in deep learning experiments. W, b are the weights and bias. W_{inside} is the weights of internal connection in s-LIF2HH sub-networks.

3.5 Ablation results of the simplification of LIF2HH model

Membrane potential-encoding and spike-encoding refers to how the sub-networks encode information and transfer between layers. All the components of membrane potential-encoded and spike-encoded tv-LIF2HH models are consistent except for the output form. The results suggest that the membrane potential-encoded tv-LIF2HH model can not be trained because of the gradient explosion. The spike-encoded tv-LIF2HH model performs the best (close to the HH model we actually use), followed by the s-LIF2HH model, and the s-LIF2HH without reset mechanism performs the worst.

Supplementary Table 9 | Ablation results of LIF2HH models under multi-task learning setup.

Model type	Spike	Reset	Acc1	Acc2
Membrane potential-encoded tv-LIF2HH	No	No	10.14±0.07	11.21±0.09
Spike-encoded tv-LIF2HH	Yes	No	82.48±0.23	80.63±0.26
s-LIF2HH without reset mechanism	Yes	No	68.89±1.77	74.58±0.69
s-LIF2HH	Yes	Yes	75.36±1.12	77.64±0.49

[*] All networks are fully connected and have 512 sub-networks. The hyperparameters of s-LIF2HH without reset mechanism is consistent with s-LIF2HH. The results are obtained from 6 repeated trials (mean ± std).

3.6 Statistical tests for results in multi-task learning experiment

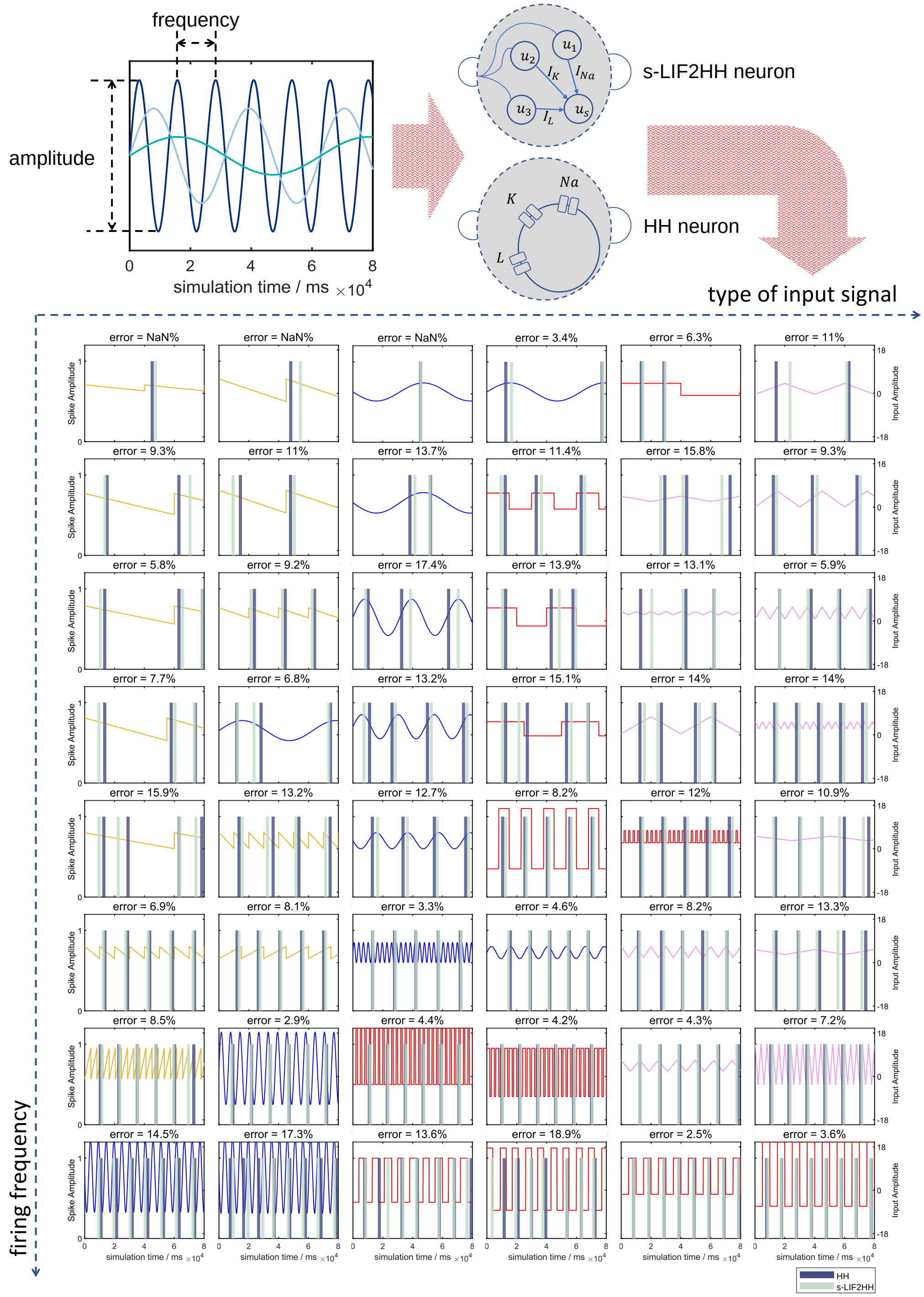
We have conducted two-sided t-test for all the models. Results are shown in Supplementary Tables 10.

Supplementary Table 10 | P values for multi-task learning experiment.

Models	Fc-accuracy1	Fc-accuracy2	Conv-accuracy1	Conv-accuracy2
HH + s-LIF2HH	9.82E-26	4.73E-27	2.35E-5	2.11E-8
HH + LIF	3.96E-55	4.71E-52	1.05E-4	1.07E-4
HH + 4×LIF	1.92E-54	4.74E-46	1.53E-11	5.76E-17
HH + b-ANN	1.82E-54	1.63E-50	7.07E-15	4.24E-11
s-LIF2HH + LIF	2.60E-33	2.13E-34	3.62E-13	4.96E-15
s-LIF2HH + 4×LIF	6.36E-27	1.97E-26	1.14E-3	1.20E-5
s-LIF2HH + b-ANN	2.28E-32	3.14E-33	3.37E-19	1.05E-17
LIF + 4×LIF	6.35E-29	3.83E-25	1.49E-24	2.99E-26
LIF + b-ANN	6.91E-5	1.15E-3	1.02E-13	6.46E-9
4×LIF + b-ANN	4.17E-26	3.14E-22	9.88E-24	2.69E-24

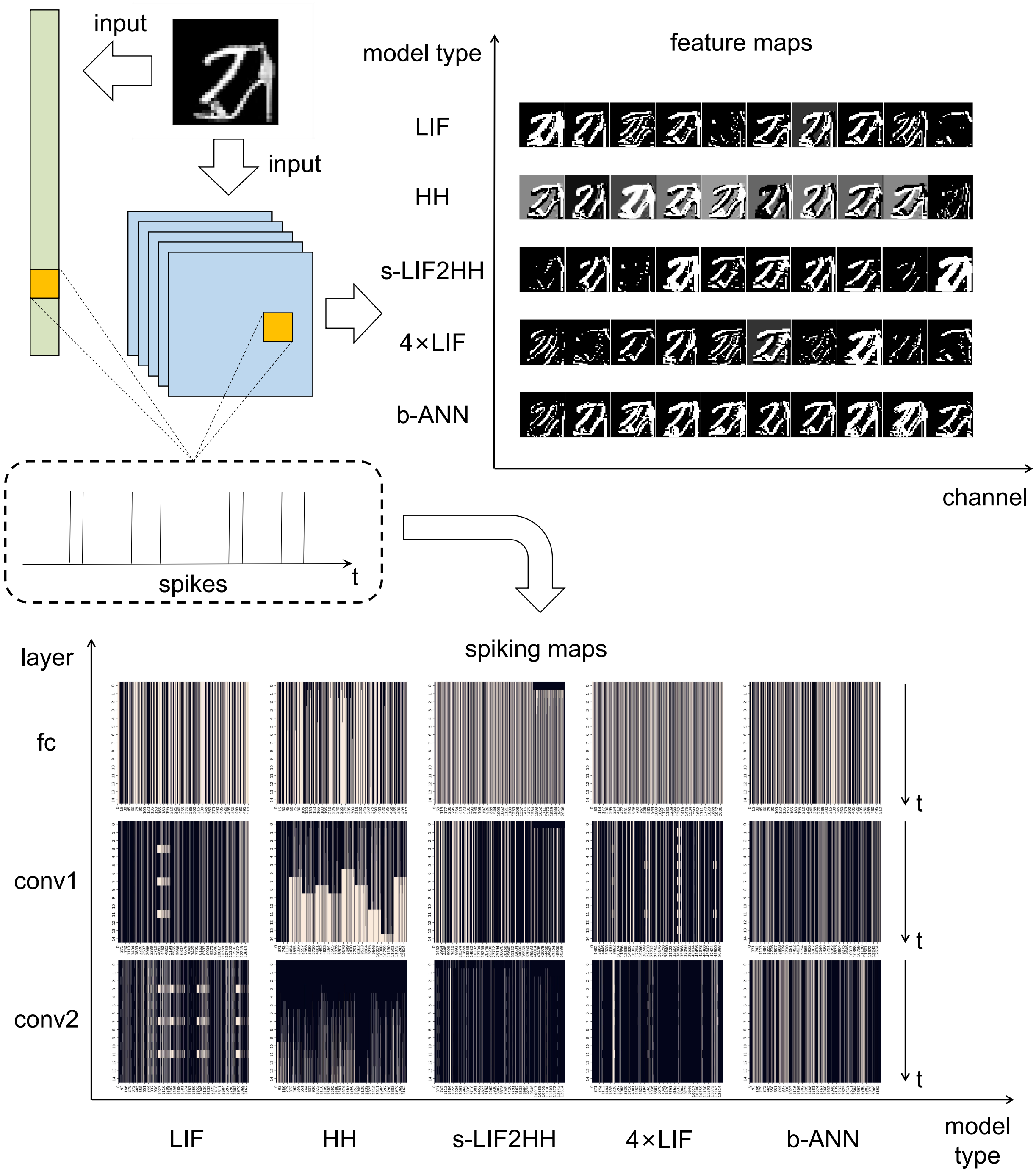
4 Supplementary Figures

4.1 Results of the simulation experiment for simplified models



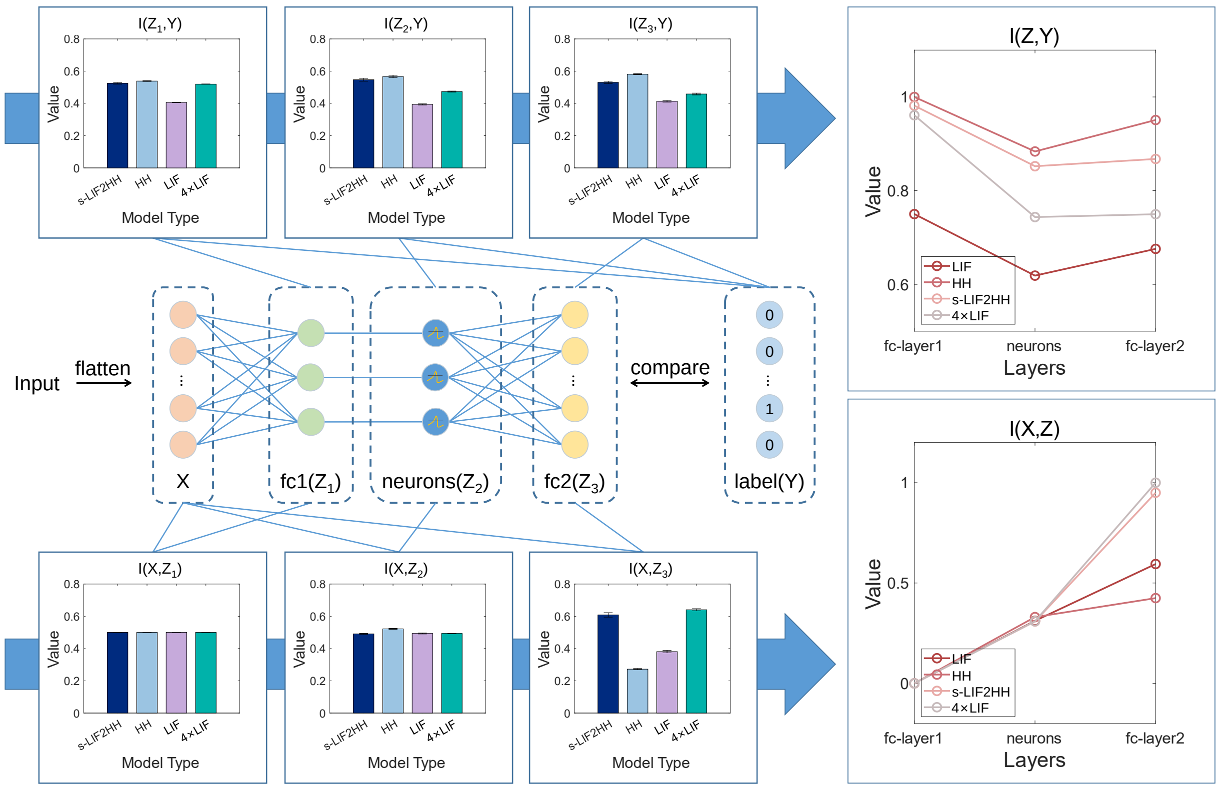
Supplementary Figure 1 | Verification of approximate dynamic properties between HH neuron and s-LIF2HH sub-network. The firing time errors yielded in most situations are quite small, which indicates that with the inputs of different types, amplitudes and frequencies, the s-LIF2HH and HH models not only produce the same number of spikes but also keep their firing times close. The results demonstrate that their overall dynamic properties are very similar and they can be considered approximate.

4.2 Spiking maps and feature maps in the multi-task learning experiment



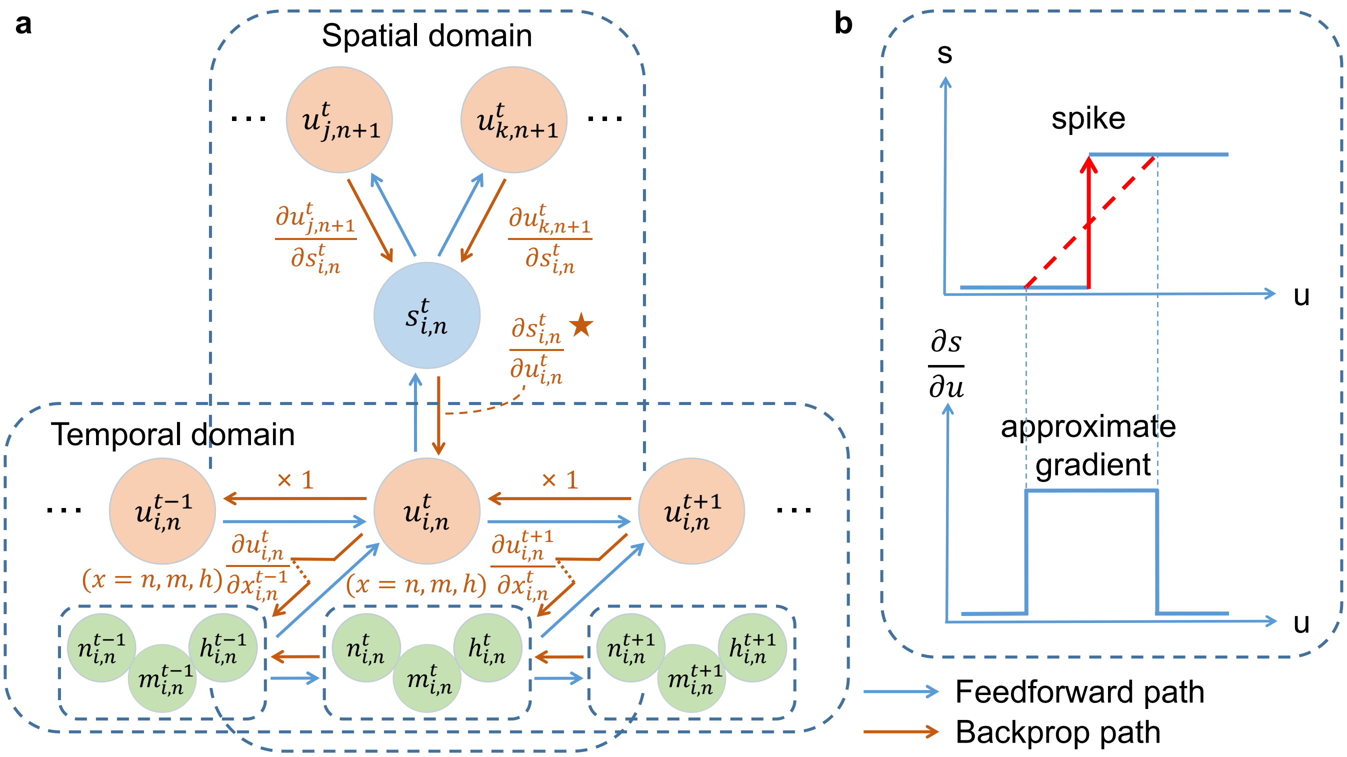
Supplementary Figure 2 | Feature maps and spiking maps of different networks in the multi-task experiment. The inputs are transmitted to the fully-connected (left) and convolutional (lower) SNN with various types of neurons. The convolutional networks (conv) extract features from the inputs in each channel, and each model has a unique tendency of feature to extract. Meanwhile, the spiking maps of each model also have distinctions, especially between HH models and others. (the X-axis and Y-axis of a single spiking map refer to the neuron number and time step, respectively, and the white points represent the spikes while the black points represent the opposite.)

4.3 Mutual information analysis



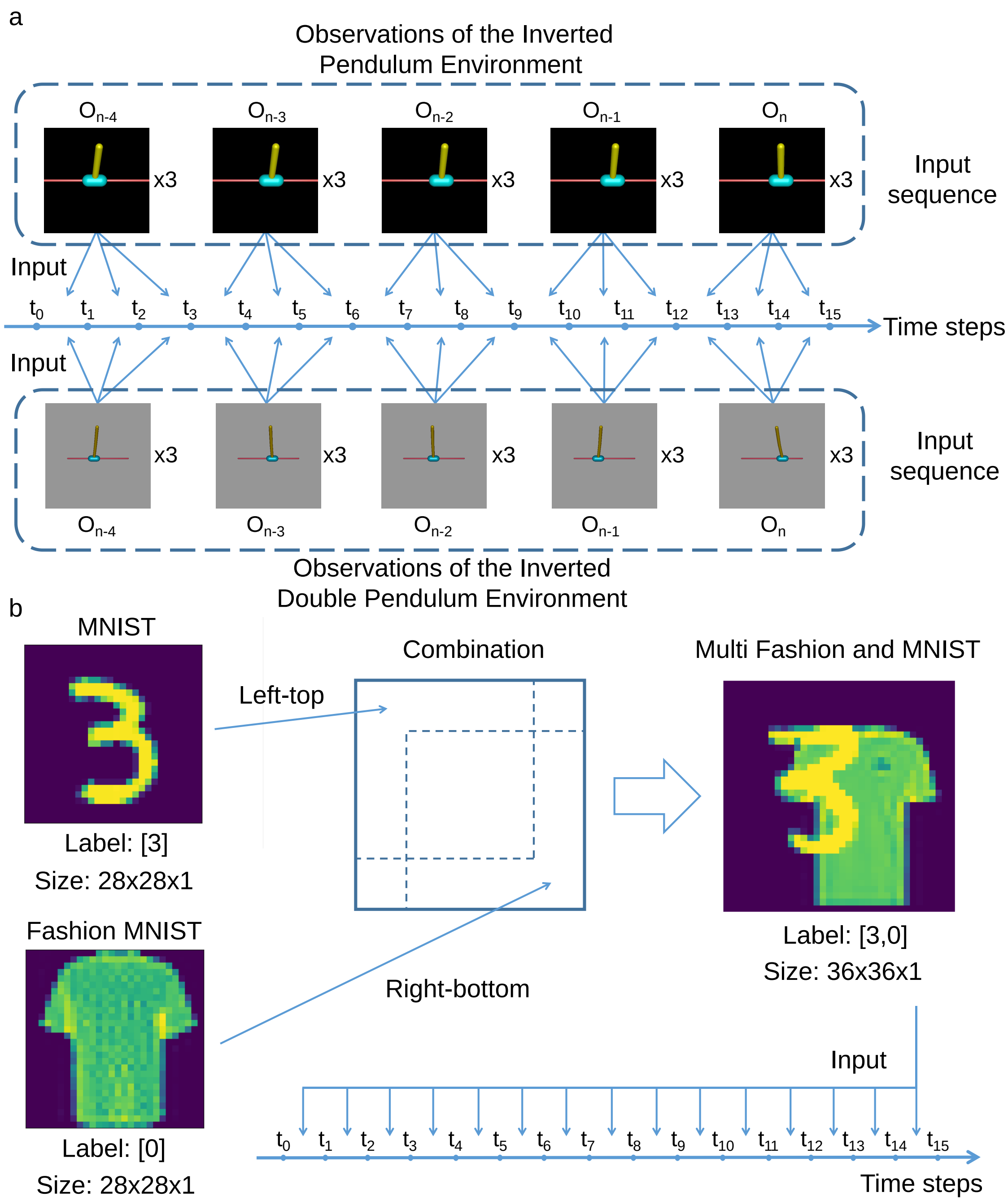
Supplementary Figure 3 | Mutual information of each layer in the SNNs. The $I(X,Z)$ values are similar after the neurons layer but show great difference after the fully-connected layer2 (fc-layer2), which demonstrates that the network complexity advantage of HH model is primarily from the smaller fc layer (or less external complexity). Meanwhile, the variation of the $I(Z,Y)$ values presents that the representation capabilities of the SNNs can be derived from the neuron model (or internal complexity) as well as the topological structure (or external complexity). The results are obtained from 10 repeated trials.

4.4 Error propagation for HH model



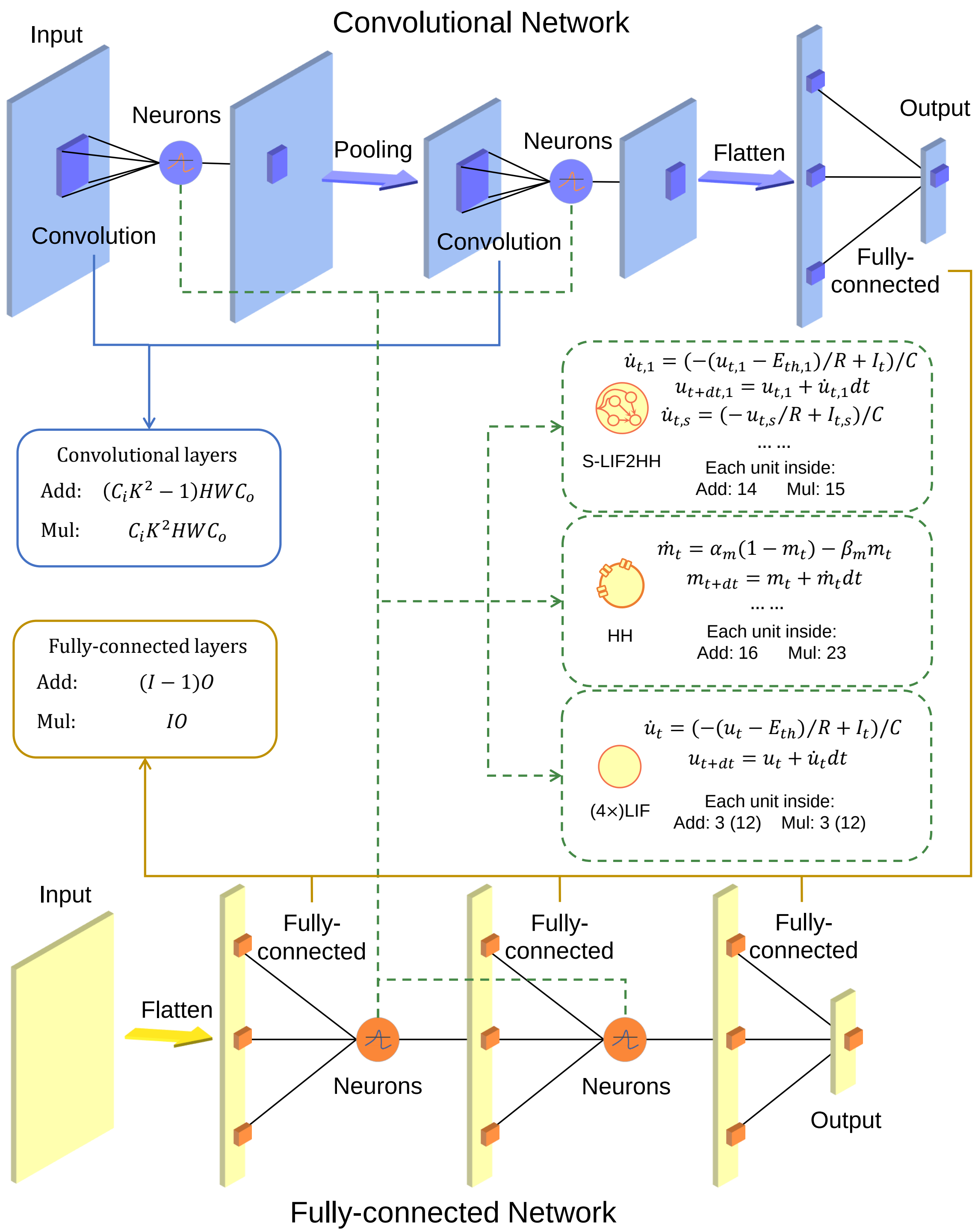
Supplementary Figure 4 | Error propagation for HH networks. (a) In the spatial domain, similar propagation occurs for HH networks and other common SNNs, while in the temporal domain, HH networks have a gradient switch between membrane potential u and three special parameters (m, n, h). We are capable to open this propagation path, but in our work, it is not opened. **(b)** The approximate calculation method of the gradient between spike s and membrane potential u (marked by red star in left part). The red dashed line is the approximate s-u curve.

4.5 Datasets and environments



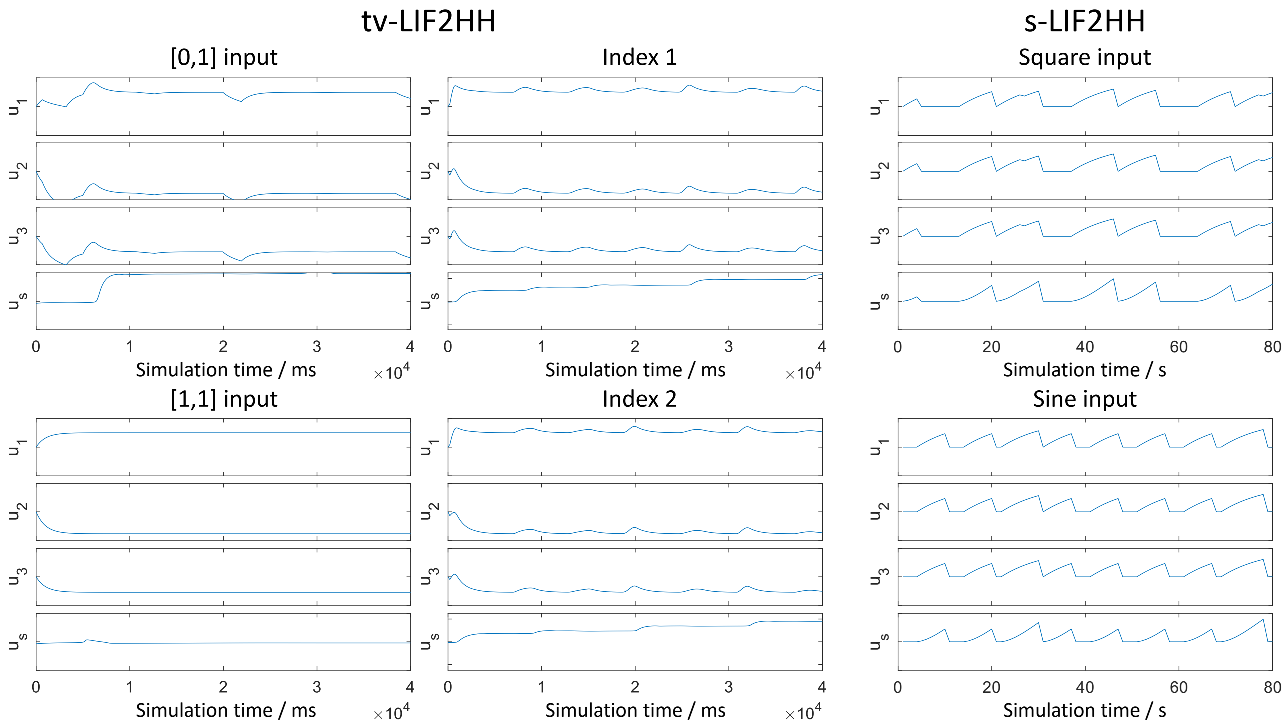
Supplementary Figure 5 | Datasets and environments used in the multi-task learning and deep reinforcement learning experiments. a, In the deep reinforcement learning experiment, 5 contiguous observed states serve as inputs of the 15 time steps of the Policy networks. **b,** A sample from the MNIST dataset and a sample from the Fashion MNIST dataset are put separately on the left-top and right-bottom of a single image, converting to a sample of the Multi Fashion and MNIST dataset. The vacant parts in the corner are padded with zero, and then the entire image with size [36, 36] feeds into the networks.

4.6 Calculation method of FLOPs



Supplementary Figure 6 | Calculation method of the FLOPs of each model. The FLOPs of the fc and conv layers of all the models are counted in the same way presented in the left part, while the FLOPs of neuron layers vary with the type of model, as shown in the right part. The add operations of each time step inside an HH neuron is at the same level as an s-LIF2HH sub-network or four LIF neurons, but an HH neuron conducts substantially more multiply operations than an s-LIF2HH sub-network and four LIF neurons, as the dynamical equations of HH neuron are more complicated. Meanwhile, note that 4×LIF model connects all the neurons between layers, and the three LIF neurons in s-LIF2HH sub-network receive inputs from all the LIF neurons in the previous sub-network. The wider fc layers and the more conv layers in s-LIF2HH and 4×LIF models result in more FLOPs outside the neuron layers, in comparison to HH and LIF models.

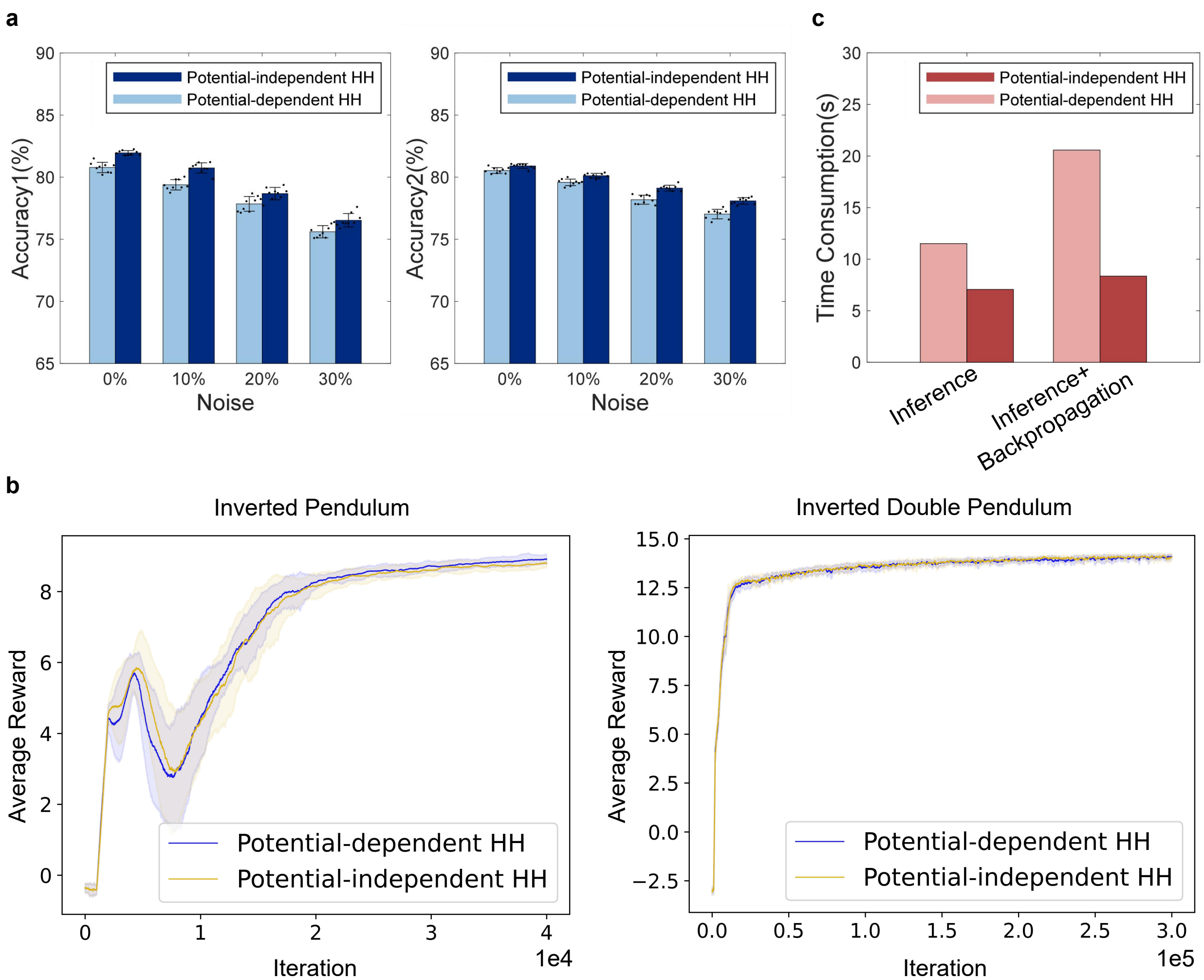
4.7 Membrane potentials of LIF neurons inside LIF2HH sub-network



Supplementary Figure 7 | Membrane potentials of the neurons within LIF2HH sub-network. The tv-LIF2HH part is a supplement to Fig. S2.1 and S2.2, showing the membrane potential of each tv-LIF neuron within tv-LIF2HH sub-network. Similarly, the s-LIF2HH part is a supplement to Fig. 3d in Results. For tv-LIF2HH model, the first three neurons emulate the behavior of Na^+, K^+, L three ion channels, while the last neuron emulates the residual response of the HH neuron. For s-LIF2HH model, the first three neurons behave the same since we set their input weights to be equal, while the behavior of the last neuron is slightly delayed compared to the first three neurons since it is triggered by their membrane potentials.

4.8 Comparative results for potential-dependent and independent HH models

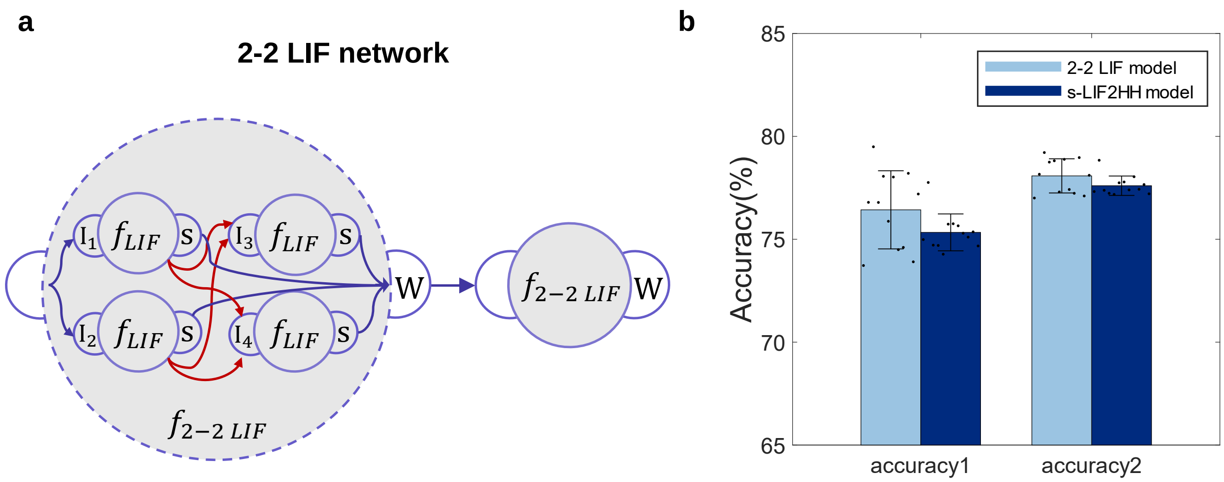
Potential-independent means that the channel parameters (α and β) of HH model are not functions of membrane potential u anymore. For the deep learning tasks, we initially experimented with both potential-dependent and independent setups. Our results indicate that potential dependency produces a nuanced difference in task performance but requires longer computational time. As a matter of fact, under the multi-task experimental setting with 0%, 10%, 20% and 30% noise, the difference in the performance of potential-dependent and independent HH models is only about 1%, while potential-dependent HH model consumes at least 40% more time than potential-independent HH model. Hence, to improve the efficiency, we chose to fix the channel parameters in our simulations.



Supplementary Figure 8 | Comparative results for potential-dependent and independent HH models in learning tasks. a, Results of the two metrics in the multi-task learning experiment. 20 biological replicates are performed, and data are presented as mean values +/- SD. The performance of potential-dependent and independent HH models only has a slight difference. **b,** Reward curves in the training process (Left: Inverted Pendulum environment. Right: Inverted Double Pendulum environment). 11 biological replicates are performed, and data are presented as mean values +/- SD. Potential-dependent HH model and potential-independent HH model show similar performance. **c,** Time consumption in the inference and backpropagation process. Potential-independent HH network is faster than potential-dependent HH network.

4.9 Comparative results for 2-2 LIF and s-LIF2HH models

We conduct this comparative experiment to demonstrate that the high performance raised from internal complexity in basic units is not limited to HH and s-LIF2HH models alone. We constructe a 2-2 LIF model with 4 neurons inside the sub-network, which has similar complexity to s-LIF2HH model, as evidenced by the need for four equations to describe their dynamics. It also demonstrates comparable performance to s-LIF2HH model, as shown in Supplementary Figure 9.



Supplementary Figure 9 | Comparative results for 2-2 LIF and s-LIF2HH models in learning task. a, Inner structure of 2-2 LIF model. The arrows refer to the flow of information between different layers, s is the binary function for spike generation, and f denotes the dynamic function of each neuron. I is the input, and I_1, I_2, I_3, I_4 refer to different inputs. **b,** Results in the multi-task learning experiment. 12 biological replicates are performed, and data are presented as mean values +/- SD. The 2-2 LIF model and s-LIF2HH model with the same number of sub-networks have similar performance.

References

1. Hodgkin, A. L. & Huxley, A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The J. physiology* **117**, 500 (1952).