Brain-inspired global-local hybrid learning towards human-like intelligence

Yujie Wu[1], Rong Zhao[1], Jun Zhu[2], Feng Chen[3], Mingkun Xu[1], Guoqi Li[1], Sen Song[4], Lei Deng[5], Guanrui Wang[1], Hao Zheng[1], Jing Pei[1], Youhui Zhang[2], Mingguo Zhao[3], and Luping Shi[1*]

[1]Center for Brain-Inspired Computing Research (CBICR), Beijing Innovation Center for Future Chip, Optical Memory National Engineering Research Center, Department of Precision Instrument, Tsinghua University, Beijing, China

[2]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

[3]Department of Automation, Tsinghua University, Beijing 100084, China

[4]Laboratory of Brain and Intelligence, Department of Biomedical Engineering, IDG/ McGovern Institute for Brain Research, CBICR, Tsinghua University, Beijing, China

[5]Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106, USA

[*]e-mail: lpshi@mail.tsinghua.edu.cn.

Abstract

The combination of neuroscience-oriented and computer-science-oriented approaches is the most promising method to develop artificial general intelligence (AGI) that can learn general tasks similar to humans. Currently, two main routes of learning exist, including neuroscience-inspired methods, represented by local synaptic plasticity, and machine-learning methods, represented by backpropagation. Both have advantages and complement each other, but neither can solve all learning problems well. Integrating these two methods into one network may provide better learning abilities for general tasks. Here, we report a spike-based hybrid learning model that integrates the two approaches by introducing a meta-local module and a two-phase causality modelling method. The model can not only optimize local plasticity rules, but also receive top-down supervision information. In addition to flexibly supporting multiple spike-based coding schemes, we demonstrate that this model facilitates learning of many general tasks, including fault-tolerance learning, few-shot learning and multiple-task learning, and show its efficiency on the Tianjic neuromorphic platform. This work provides a new route for brain-inspired computing and facilitates AGI development.

Main text

Realizing human-level intelligence has been a long-term goal that has been sought through various means of artificial intelligence. Machine learning-oriented learning, represented by the backpropagation family, is a powerful method that can learn sophisticated mappings from abundant data and even surpass the human-level performance in some specific tasks, such as processing of images[1] and natural language[2] and playing games[3]. In contrast, neuroscience-based learning mainly focuses on achieving advanced cognitive functions in dealing with the complex environments, such as associative memory[4] and one-shot learning abilities[5], and exploring rich brain-inspired learning mechanisms with computational efficiency[6]. This approach is based on local plasticity rules, rich spike coding schemes and spatio-temporal dynamics which are represented as the Hebbian and spike timing-dependent learning[7-9].

Both approaches are built on neural networks, have specific advantages and complementarities, and share the same vision to advance intelligence with learning capability. However, neither method currently outperforms the other on all learning problems, and both methods are still inferior to human-level general intelligence. To further improve the learning capability and efficiency towards human-like intelligence, it is considerable to leverage their different advantages and integrate them into one single hybrid model[10-12]. Recently, Tianjic, a unified neuromorphic computing platform has been developed[12], which combines computer-science-oriented and neuroscience-oriented approaches and facilitates the the development of artificial general intelligence (AGI). It is also highly expected to integrate these two approaches and develop an effective complementary learning algorithm. However, because of fundamental differences in learning characteristics, combining these approaches is greatly challenging.

Backpropagation and most of its variants strictly yield two alternative spatial information circuits (top-down and bottom-up) and communicate with synchronized continuous signals. Its powerful learning capability largely benefits from hierarchical feature abstractions via layer-by-layer allocation of supervision errors in a topological space. In contrast, neuroscience-oriented learning prominently occurs between presynaptic and postsynaptic neurons and is triggered by asynchronous event-driven spike activity (**note that** such brain-inspired weight updates that are computed over adjacent neuronal activity are called 'local plasticity' (**LP**) in this context, and the former are called 'global plasticity' (**GP**) for distinguish). Hence, these two approaches exhibit prominent differences in spatial learning circuits and temporal learning scales.

Although these diverse learning features have drawn widespread attention, researchers have considered this problem with different interests and perspectives[13-20]. A related long-term study aims to determine the approximate equivalence between the backpropagation family and certain neuroscience-oriented learning rules[13,14]. Another vein is to experimentally merge biological learning features into artificial neural networks[15-17]. While it could improve performance of certain tasks, most work lacks theoretical interpretations and ignores some salient features of brain learning, such as rich neuronal dynamics and high-efficiency neural spike coding, which makes it difficult to support rich neuroscience-oriented learning models and learning rules. Recently, there exist several attempts to combine these two types of learning in a coarse manner[18-20], such as using local learning rules to abstract features in previous layers and feedback the propagation of a GP-based classifier. However, they did not reconcile the above-mentioned fundamental learning differences, thereby impeding the development of hybrid learning.

1 Two-phase hybrid learning by the hybrid plasticity

We present a spike-based hybrid plasticity (HP) model by introducing a meta-local module and a two-phase causality modelling strategy, that are derived from neuronal dynamics and can integrate the two approaches to form a synergic learning model. In developing this model, two challenges need to be overcome.

The first is to design a synergistic network prototype that can support different learning circuits. By drawing inspiration from brain plasticity mechanisms, we introduce a meta-local module to bridge the LP and GP circuits (Fig. 1a). Our design principles are largely based on three neuroscience evidences: (1) Experimental data[21,22] show that our brain exhibits abundant top-down modulated signals, such as neuromodulators and neurotransmitters, which encodes supervision information into the local circuits resulting in sub-optimal performance, commonly referred to the third factor; (2) These modulated signals are shared among many synapses and modulates synaptic plasticity behaviors[23,24], such as the weight update rate and the plasticity consolidation. Thus, it acts as a meta-learning role on synaptic plasticity in a weight-sharing manner; (3) These modulated signals can encode errors on multiple scales and have a different learning scale from synapses[25,26]. Therefore, updating of these meta-hyperparameters can be different from weight updates and be performed by optimizing the expectations of objective functions across several time steps, epochs or tasks, depending on the specific context. On this basis, we introduce a parametric local module that conveys supervised signals for the modulation of local plasticity. Specifically, we first derive an explicit spatio-temporal signal propagation model of neuronal dynamics from both neuronal membrane dynamics and synaptic dynamics. Then, we transform the hyper-parameters of local plasticity, such as the learning rate and sliding threshold, into a group of trainable meta-parameters $\boldsymbol{\theta}$. When providing exclusive optimization of $\boldsymbol{\theta}$, the local module is expected to progressively learn to local plasticity rules and further producing useful inductive biases of historical input patterns to facilitate neuron computations.

Another challenge is to reconcile different synaptic learning scales and efficiently incorporate these two types of learning into one optimization framework. Because bio-plausible plasticity rules have local event-driven weight modifications and independent learning scales, directly combining them is difficult to ensure the convergence. As illustrated in Fig. 1b, we observe that *synaptic plasticity is mainly affected by the adjacent neuron activity in the current state, and in turn affects the spread of spike signals in time and space*. According to the equivalent causality modelling, we first formulate the entire local plasticity into a buffer function related to temporal spike activity (Fig. 1c). Thus, in the training phase, we can model local behaviours of neurons as the temporal effect of adjacent neuron activity on the subsequent signal transmission. This ensures that the local plasticity can be transformed in a parametric function and the effect of synaptic modifications is further incorporated in the temporal credit assignment problems. We note that this transformation is coincident with a variant of the synaptic dynamics. Specifically, derived from the synaptic differential dynamics, the synaptic weights $w(t)$ have two terms

$$w(t) = w(t_n)e^{\frac{t_n-t}{\tau_w}} + P(t, pre, post, w; \boldsymbol{\theta}) \triangleq w_{GP} + w_{LP}, \tag{1}$$

where $w(t_n)$ denotes the weight phasic value at time t_n, $k(t) = e^{\frac{t_n-t}{\tau_w}}$ denotes the synaptic decay function, τ_w denotes the synaptic constant, and $P(t, pre, post, w; \boldsymbol{\theta})$ denotes generic local modifications controlled by the presynaptic activity, pre, the postsynaptic activity, $post$, the local meta-parameters $\boldsymbol{\theta}$, and specific biological rules. On this basis, if we further assume supervision signals to modify the phasic state, $w(t_n)$, in the form of phasic signals, such as the top-down pulse current, and assume $P(t, pre, post, w; \boldsymbol{\theta})$ to represent local plasticity, then the equation (1) can be used to accordingly decompose the synaptic weight into two parts, w_{GP} and w_{LP}.

In this manner, our model achieves a global-local hybrid learning using a two-phase learning mechanism, which can reconcile the diverse features of LP and GP learning by allocating them to act on different weight fractions and time scales. In the self-organised phase, the network receives input stimulus and propagates the signals in an event-driven manner. In parallel, the local plasticity is driven by the neuronal co-activity in the adjacent layer. In the supervised phase, the errors of supervision act as phasic signals to instantly adjust the synaptic phasic value and are also used to optimise the local meta-parameters. Our goal is that in providing this two-phase learning mechanism and specific optimizations for different weight fractions, the network can progressively learn to optimize its local learning rules, and eventually converge into a stable synergic working mode. Detailed comparisons among our hybrid learning method and single learning methods can be found in supplemental Table S1.

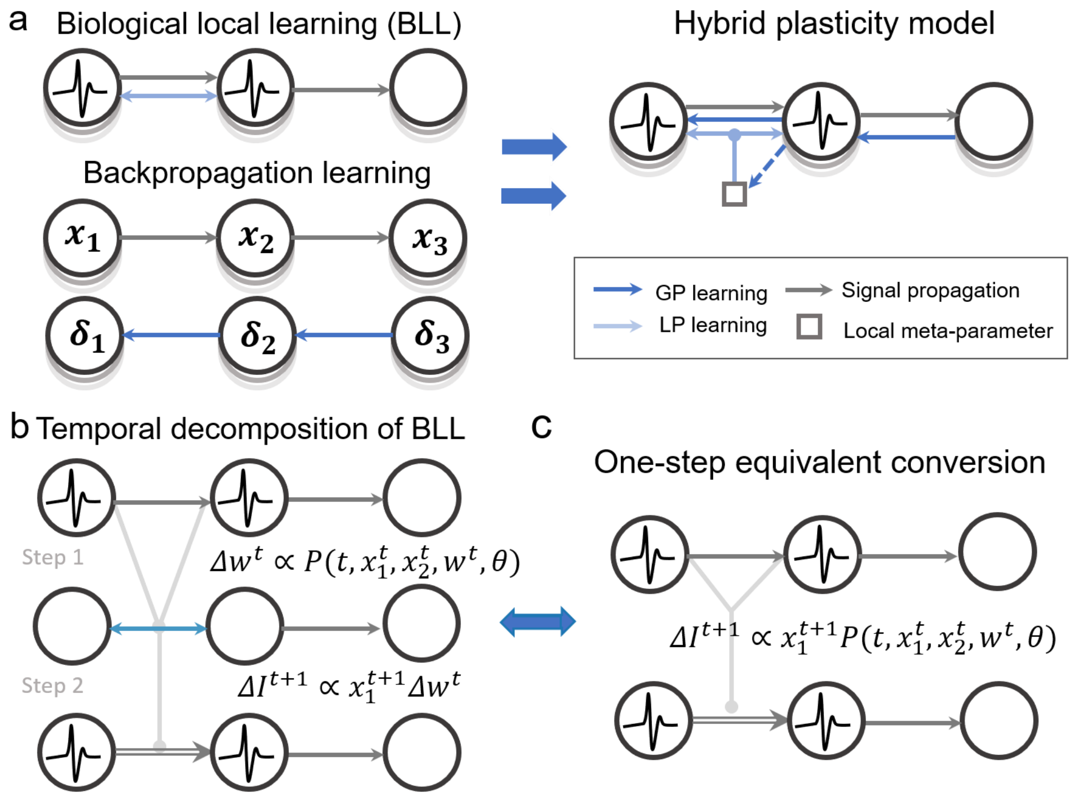
a
Biological local learning (BLL)
Backpropagation learning
x_1
x_2
x_3
δ_1
δ_2
δ_3
Hybrid plasticity model
GP learning
LP learning
Signal propagation
Local meta-parameter
b Temporal decomposition of BLL
Step 1
$\Delta w^t \propto P(t, x_1^t, x_2^t, w^t, \theta)$
Step 2
$\Delta I^{t+1} \propto x_1^{t+1} \Delta w^t$
c
One-step equivalent conversion
$\Delta I^{t+1} \propto x_1^{t+1} P(t, x_1^t, x_2^t, w^t, \theta)$

Figure 1. Illustration of two-phase hybrid plasticity (HP) methods. **a**, Backpropagation and most of its variants follow two information circuits: the top-down circuit and bottom-up circuit. Biological local methods usually learn via the concurrent activity between presynaptic and postsynaptic neurons. The HP method models the meta-parameters of local learning rules as a type of special hyperparameters to bridge these two learning circuits. **b**, A temporal decomposition of biological local learning (BLL). Biological short-term plasticity assumes that plasticity is mainly affected by the adjacent neuron activity at time t and in turn affects the signals spread in the future. **c**, To reconcile different learning scales, the HP model formulizes the BLL process as a parametric function of neuron activity over time, and uses a two-phase modulation strategy to reconcile different learning scales. In providing with specific optimizations for different weight parameters, it can efficiently integrate different learning into a unified temporal credit assignment problem and thereby achieves a two-phase hybrid learning.

2 Spike coding schemes

To accommodate general neuroscience-oriented models and achieve high efficiency, it is considerable to support spike-based neural networks (SNNs). Since the HP approach is derived from neuronal dynamics, it facilitates the establishment of this approach on

SNNs and instantiation on neuromorphic hardware. One of prominent facilitations is the flexibility to support multiple spike-based coding schemes to meet different task requirements. That is to say, it can not only use rate-based coding by counting the output spike number, but also supports the use of neuron firing time to encode information (Fig. 2a). Such flexibility primarily benefits from our modelling of neuronal dynamics. Unlike previous SNN learning algorithms[27,28] that neglect the modelling of synaptic dynamics for simplicity, HP SNNs are jointly derived from synaptic dynamics and membrane dynamics, and therefore naturally maintain the synaptic decay function, $k(t) = e^{\frac{t_n - t}{\tau_w}}$, during information transmission. Because spike signals emitted by presynaptic neurons must be filtered by $k(t)$ to the next layer, an early-firing neuron can convey a stronger stimulus than a late-firing neuron; thus, the order of firing time has effects on information transition.

On this basis, it is worth noting that the HP model can naturally encode information in a form of embedded rank order coding[29] (see *Method*), and we further develop an evidence-accumulation temporal decoding scheme suitable for the HP model. Formally, as long as the first spike is triggered by the winning neuron in the output layer, the HP network will stop signal inference and produce results based on the index of the winning neuron. Then, the scaled membrane potential of the output neurons is used as an output representation to calculate the loss. In this manner, we allow the unique threshold mechanism of spiking neurons to implement an event-driven inference ability. This enables HP SNNs to flexibly balance performance and efficiency (see *Results*), and support the exploration of multiple coding schemes.

3 Results

Backpropagation-based learning and neuroscience-based local learning have achieved many successes in their respective fields. For instance, Hebbian-based learning is conductive for achieving human-like cognitive functions, such as the associative memory and the few-shot learning capabilities. Backpropagation-based learning is powerful in abstracting representation mappings from abundant data. In the following section, we will demonstrate that the hybrid model can not only retain a powerful representation abstraction capability and some advanced neuro-inspired learning merits from these two types of learning but also achieve a synergic learning network with improved learning ability for handling more complex and general tasks. Firstly, we evaluate the performance of our model and analyse its convergence. Then, we elucidate the superiority of synergic learning through three demonstrations, which are usually challenging by using traditional single-learning models. Finally, we interpret the model effectiveness using an approximate implicit loss function and demonstrate its efficiency on the Tianjic neuromorphic platform[12].

Table 1: Comparison of the state-of-the-art results of spike-based networks on standard image datasets.

Model	Dataset	Method	Avg. coding latency.[b]	Accuracy (%)
Spiking MLP[30]	MNIST	LP based	350ms	95.00
Spiking CNN[27]	MNIST	GP Converted.[a]	200ms	99.12
Spiking CNN[31]	MNIST	GP based	100ms	99.59
Spiking CNN (This work)	**MNIST**	**LP + GP**	**7.5ms**	**99.51**
Spiking MLP[32]	F-MNIST	GP based	400ms	90.13
Spiking CNN (This work)	**F-MNIST**	**LP + GP**	**3.9ms**	**93.31**
Spiking CNN[28]	N-MNIST	GP based	300ms	99.36
Spiking CNN (This work)	**N-MNIST**	**LP + GP**	**39ms**	**99.51**
Spiking CNN[33]	CIFAR10	GP Converted.[a]	500ms	91.55
Spiking CNN[34]	CIFAR10	LP based	250ms	75.42
Spiking CNN (This work)	**CIFAR10**	**LP + GP**	**4.5ms**	**91.12**
Spiking CNN[35]	CIFAR100	GP Converted.[a]	200ms	55.13
Spiking CNN (This work)	**CIFAR100**	**LP + GP**	**6.9ms**	**62.01**

.[a] converted from a pre-trained GP-based ANN; .[b] Results from the training windows reported in the published work.

Comprehensive performance evaluation of the HP model

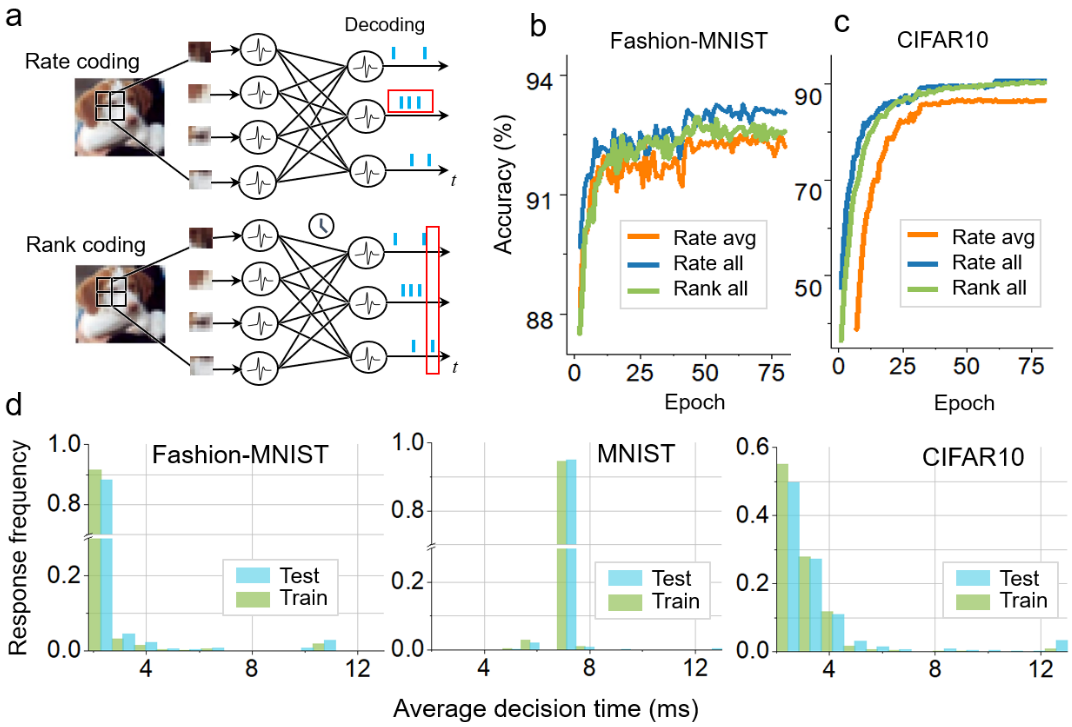

Figure 2. The hybrid plasticity spiking neural network can flexibly support different coding schemes and achieve a trade-off between performance and efficiency. a, Illustration of rate coding and temporal coding. In rate coding, the network counts the number of spikes fired in the last layer and outputs the results. In temporal coding, the timing of spike firing encodes additional information. **b-c,** Comparison of the training curves of the model with different coding schemes (called Rate-all or Rank-all) for the Fashion-MNIST (**b**) and CIFAR10 (**c**). The average firing time over ranking coding schemes is counted and rounded up, and the rate-based model is tested with this average time window (called Rate-avg). **d**, Average response time for each testing category in three datasets. We calculated the response timesteps required to produce results for each testing sample and generated a normalised frequency histogram.

We comprehensively evaluated the performance of the HP SNNs on the accuracy, efficiency and convergence using five standard image classification datasets, including MNIST, Fashion-MNIST, N-MNIST CIFAR10 and CIFAR100. We first analysed its accuracy and coding efficiency with respect to the average response time of test data and reported it in Table 1. Because spike-based networks achieve a balance between performance and efficiency, a modest accuracy loss is usually observed compared with full precision neural networks. Thus, we mainly compared our model with spike-based or binary-based models. Table 1 shows that the HP SNNs achieve higher accuracies compared with other published work and has a significant lower average coding latency than other models.

To further analyse the coding efficiency, we compared the HP model with different coding schemes in Fig. 2b-d. Fig. 2b-c compare the results of different coding methods at different simulation times. It indicates that HP SNNs are well suited for rate-based and rank-based spiking networks. With a longer average time window, the rate-based model obtains higher accuracy. With more flexible and event-driven response characteristics, the rank-based model achieves faster inference with a slight accuracy loss. With the same average response time, the performance of the rank-based model is superior to that of the rate-based model, which indicates that the rank-based model can achieve a trade-off between accuracy and efficiency. Furthermore, we also plot the details of the average response time of each category for different datasets in Fig. 2d. Unlike most SNN models that either use a long time window to simulate neuronal dynamics[27,33] or represent a spiking neuron as a binary neuron[36], we found that HP SNNs use a more flexible strategy for decision-making. Using the CIFAR10 dataset as an example, for most categories, HP SNNs can make decisions within four time-windows, while for some complicated patterns, the model requires more time to make decisions. This flexible decision process significantly reduces the average inference latency and further leverages the high efficiency of neuromorphic hardware. We instantiated our model on the Tianjic chips and reported the energy evaluation in supplemental Table S2.

Next, we compared the convergence of the proposed approach with other related learning methods. Intuitively, since local learning and global learning have independent update methods, directly combining them cannot ensure convergence. Furthermore, regarding neuroscience-oriented learning models, systematically and generally configuring parameters over local learning rules has yet to be resolved. Although hand-designed or bio-simulated hyperparameters can alleviate the above problems to some extent, it is time-consuming and difficult to ensure the performance[37]. By parametrization of the local module, the HP approach can automatically optimise the local hyperparameters and further achieve a synergic learning mode. For a demonstration, we comprehensively compared the loss curves and accuracies of a single LP network, a single GP network, a fine-tuning $LP + GP$ network and the proposed HP network in Fig. 3. For fairness, we used the same initial weight configurations.

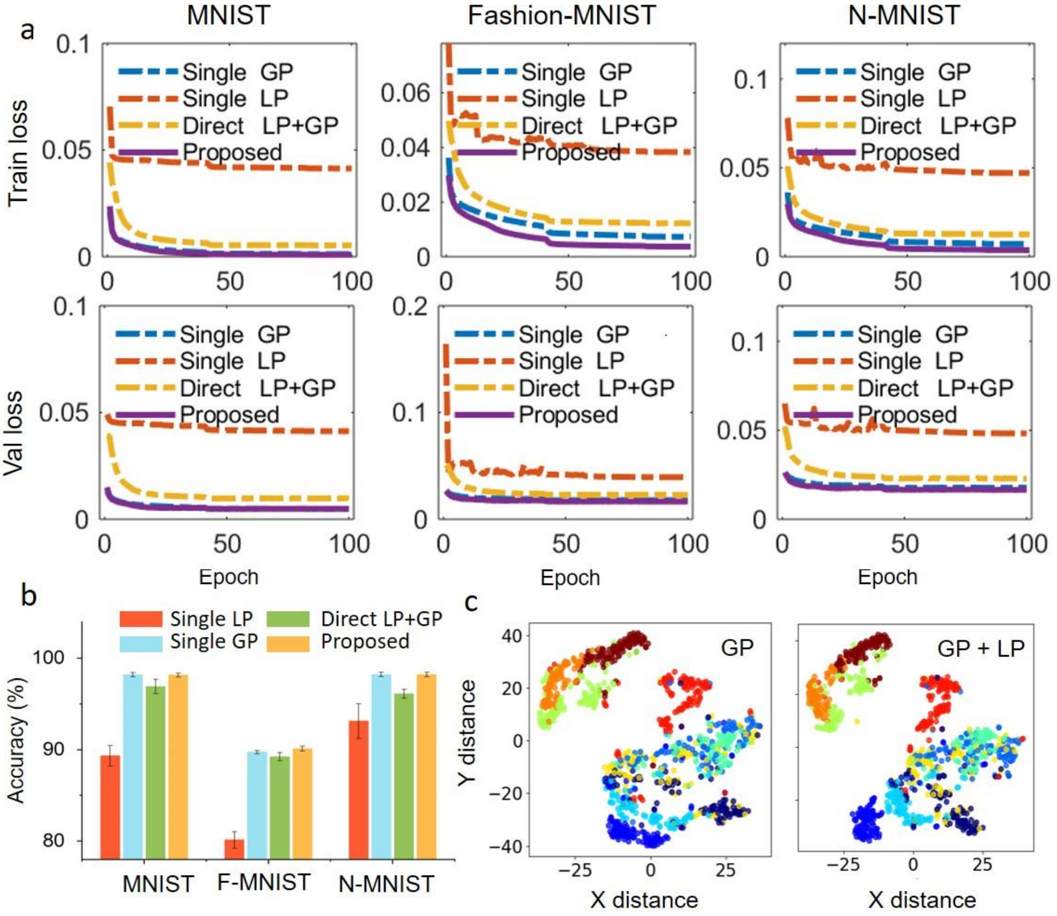

Figure 3. The hybrid plasticity approach can effectively ensure the convergence and accuracy of hybrid learning. **a,** Comparison of the convergence curves of the single GP-based, LP-based, fine-tuning-based (denoted by direct LP + GP) and proposed HP model. **b,** Accuracy histogram for the four models. **c**, T-distributed stochastic neighbour embedding visualisation of the membrane potential in the first hidden layer for 1000 random test samples.

Fig. 3a shows that the fine-tuning method has a poor convergence and performs worse than the other models. Through the proposed compatible design, the HP model significantly improves over the LP model in accuracy (Fig. 3b) and achieves the best convergence, indicating that the proposed method can efficiently integrate LP and GP methods. Furthermore, we visualised the activations in the first hidden layer by 2D embedding visualization of T-distributed stochastic neighbour embedding[38] with the Fashion-MNIST. Fig. 3c illustrates that the HP method can abstract points within each class more compactly and push different classes farther. Overall, the above results indicate that the HP can efficiently coordinate GP and LP methods and is helpful for abstracting data features.

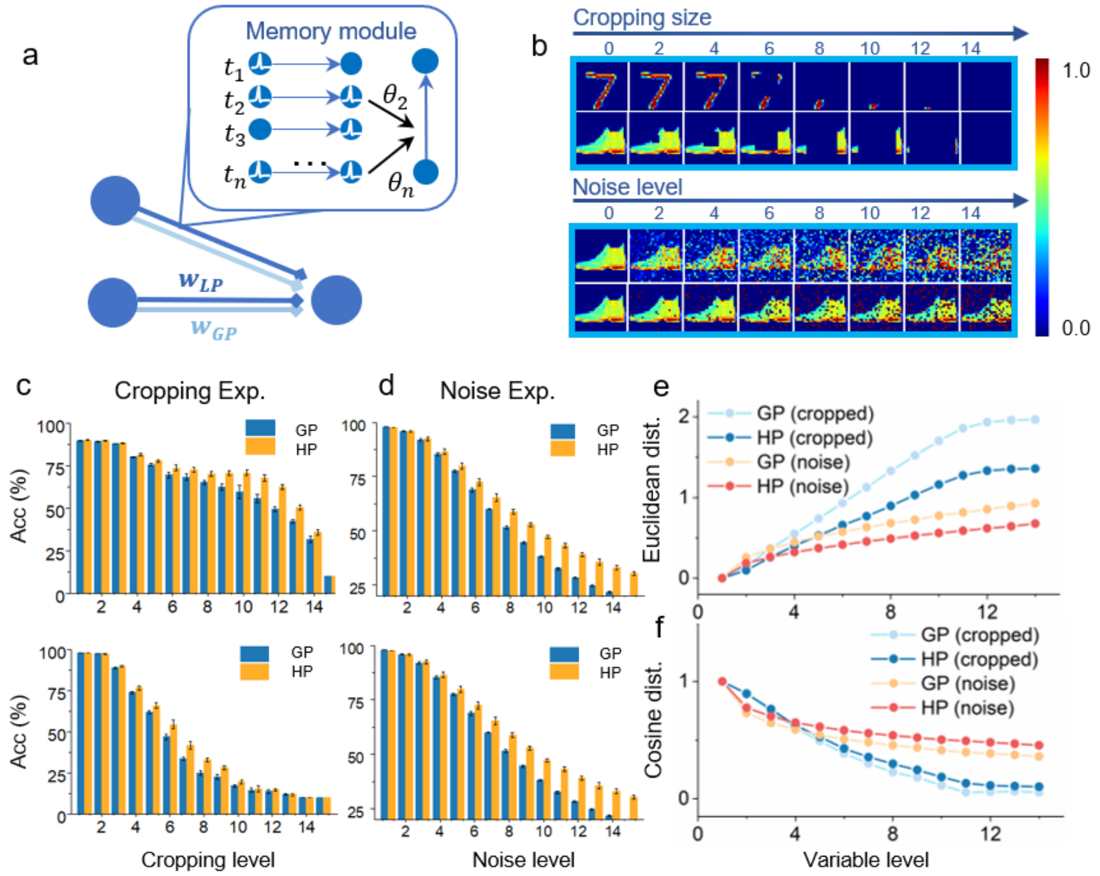

Figure 4. Hybrid plasticity improves the learning tolerance. a, An illustration of the memory based on Hebbian-like local module. Local plasticity helps neurons detect a repeated stimulus and in turn strengthens the synaptic weight with the learnable local hyper-parameters. In this manner, it can memorize the repeated pattern and facilitate the recognition of the similar stimulus. **b**, Generation of incomplete data by cropping (upper) and noise mixing (lower). **c,** Error bars for performance in the cropping experiments using the MNIST (upper) and Fashion-MNIST (lower) datasets. **d.** Error bars for performance using the MNIST dataset in experiments with Gaussian noise (upper) and salt-and-pepper noise (lower). **e,** Euclidean distance between the hidden activation (membrane potential in the last time step) of collapsed data and those of the original data. **f,** Cosine distance between the hidden activation of collapsed data and original data.

Development of the HP model for general tasks

Although current AI techniques achieve high (or even super-human) performance for sub-tasks in specialised domains where the data is often abundant, it still remains difficult to solve complex problems with few or incomplete information associated with many systems. It is desirable to develop an AGI system to improve the above defects. To demonstrate the applicability of HP model, we conducted three experiments closely related to the above-mentioned problems, involving the fault tolerance learning, learning with few-shot data and learning multiple tasks.

3.1.1 Improving fault-tolerance learning

Development of general systems for real-word applications requires the model to be strongly robust to incomplete data, such as accurately recognizing the variation of input patterns when only parts of the patterns are presented. As the only existing reference, our brain demonstrates remarkable fault tolerance learning for confounded data via associative memory, that is, an ability to automatically associate and retrieve prior knowledge for identification. Researchers have proved the necessity of Hebbian-based plasticity and neuromodulation for brain associative memory[4], and further shown their vital role in fault tolerance in generative models[39-41], such as bidirectional associative memory and the Hopfield network. Thus, by virtue of Hebbian-based learning and the hybrid approach (Fig. 4a), our model may improve the robustness of GP-based networks. We examined the ability of this model to handle incomplete data using MNIST and Fashion-MNIST datasets. We used incomplete data to refer to cropping data (e.g., parts of image information are masked, Fig. 4b) and noise-mixed data (data mixed with different types of noise, Fig. 4c). The models were trained on the standard datasets and tested using the incomplete test samples (see *Method*).

Fig. 4 shows as the cropping area increases, the HP model exhibits stronger resistance to the cropping area on the MNIST (upper) and Fashion-MNIST datasets (lower). Meanwhile, the noise experiments also show that the HP model achieves good robustness and mitigates the interference of different types of noise. Similarly, the superiority of the HP model becomes apparent as the noise level increases (Fig. 4d). To obtain a more insightful analysis, we calculated the Euclidean distance (Fig. 4e) and the cosine similarity (Fig. 4f) between the first hidden layer activation of the incomplete data and those of the original data using the same model on the MNIST dataset. The results illustrate that the local module can help the network reconstruct the original features from incomplete data by diminishing the pattern distance, which therefore benefits network fault tolerance.

3.1.2 Learning with few samples

One of hallmarks of high-level intelligence is the flexibility to learn with a severely limited number of samples. A typical machine learning scenario is few-shot learning. In this case, the classifier must adapt to new classes not seen in the training phase when only given a limited number of samples in each class. To efficiently establish a mapping relationship from the limited data, it is vital to leverage prior knowledge or acquire inductive biases. The GP-based networks succeed in abstracting useful features; however, they do not have a suitable mechanism to exploit prior knowledge hidden in the limited data. In contrast, the human brain is highly efficient in learning from limited data. Neuroscience findings have revealed that[5] the response of cortical neurons to sensory stimulus can be reliably increased after just a few repetitions by virtue of local synaptic plasticity.

Therefore, by introducing the local module, we expect that the hybrid model can solve this problem through a two-fold mechanism: (1) abstract sufficiently discriminant representation of input data mainly by the GP module; (2) find a useful inductive bias from the limited number of example pairs mainly using the LP module. Here, we used the Omniglot dataset to examine the performance of proposed model. We adopted a widely-accepted network configuration[42-44] and structure for comparison. Additionally, we allowed the training label to feed into the last layer only during the presentation time to help the network establish an input-to-output mapping. Fig. 5a-b depict the comparison results. The best accuracy of our model for five-way one-shot and twenty-way one-shot tasks is 98.85% and 95.07%, respectively. The significant improved accuracy compared with the single GP-based model indicates that the local module plays a critical role in performance. With the synergic learning design, the hybrid model can achieve competitive results that are comparable to the state-of-the-arts as shown in Table 2.

Table 2: Comparison of the published works with the similar network configurations on Omniglot datasets.

Model	5-way 1-shot Acc (%)	20-way 1-shot Acc (%)
MAML[42]	98.7	95.8
Siamese nets[43]	97.3	88.2
Plastic Network[17]	99.1	-
Matching Networks[44]	98.1	93.8
Memory Netwok[45]	98.4	95.0
Human-level[46]	-	95.5
GP SNNs (this work)	23.13	7.5
HP SNNs (this work)	98.85	95.07

3.1.3 Learning multiple tasks

A pivotal point of developing general models is the capacity for continual learning, that is, an ability to learn new tasks without forgetting the previous tasks[10]. Recent studies[47,48] have shown that the motor cortex disinhibits a sparse subset of dendritic branches for new tasks and thereby reduces the disruption of network memory for previous tasks. This indicates that the brain may multiplex some neuro-circuits while highly activating some synaptic connections to represent task-related information for solving new tasks. On this basis, we explored a distributed hybrid learning paradigm, that is, activating a sparse overlapping subset of weight connections by GP learning and controlling other synaptic connections by a task-sharing meta-LP module. Unlike the previous work [49] that uses a sub-network to solve a sub-task, our method allows the hybrid network to use a small number of GP-based connections to represent task-specific information, and LP learning to learn common features among tasks. By doing so, we expect to alleviate the disruption of network memory in different tasks and expand the learning capacity of hybrid networks to handle multiple tasks. In addition, because this paradigm allows the hybrid model to flexibly allocate different learning methods on different connections, this configuration flexibility can leverage the many-core neuromorphic architecture to optimize the deployment of on-chip resources, which is friendly for implementations on many-core neuromorphic hardware (see *Discussion*).

We first examined the model performance on the standard shuffled MNIST dataset and compared it with the state-of-the-art results [15,49,50]. We ran all models for five times and reported the testing results after fifty-task learning (Fig. 5c). We randomly activated 3% sparse and overlapping connections with GP-based learning for each task and used the LP learning to learn other connections. The meta-parameters of LP learning were trained using the first thirty-five tasks and fixed in the last fifteen tasks (*see Method*). Fig. 5c indicates that the proposed model obviously outperforms the single GP-based model, and the improvement becomes more pronounced as the task number increased. During the fifty-task learning, the HP SNNs consistently achieves the best results compared with other works, indicating the effectiveness of the proposed learning paradigm.

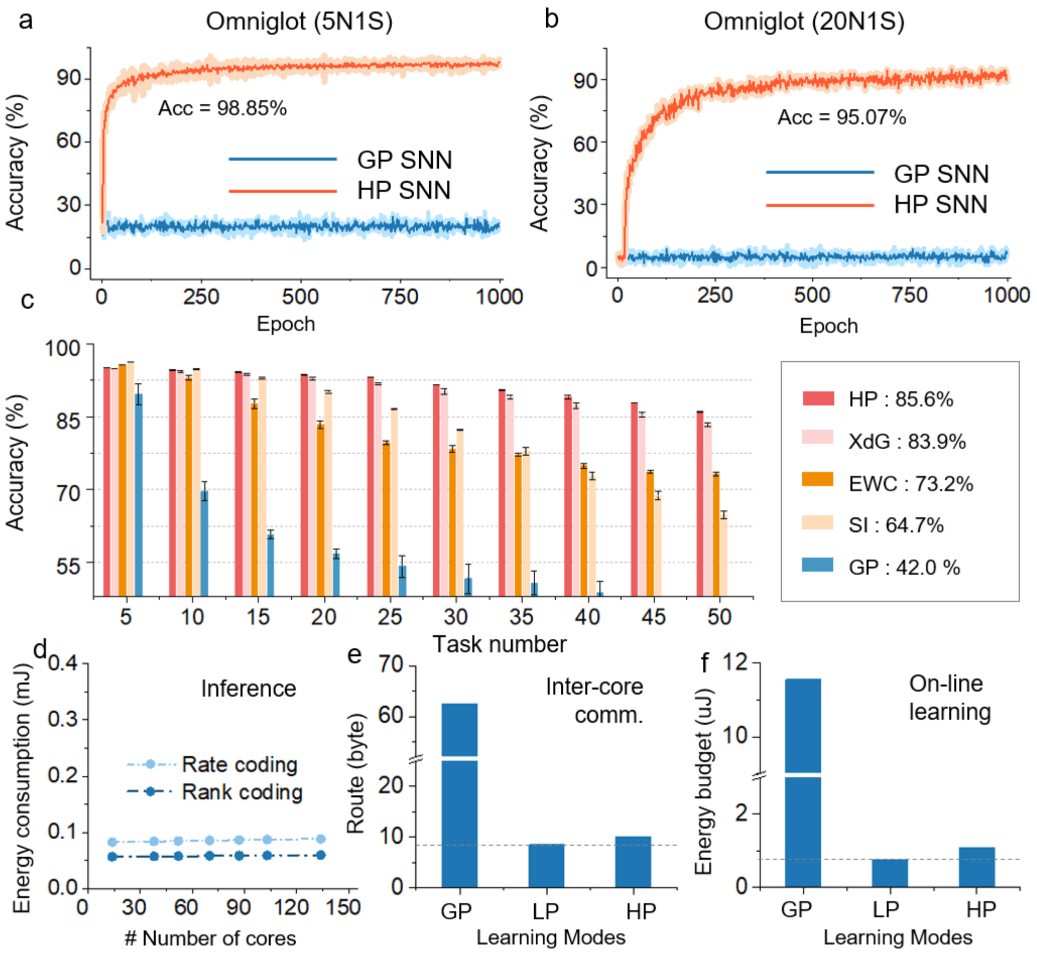

Figure 5. Performance evaluations of hybrid plasticity spiking neural networks. **a-b**, Performance comparison on the Omniglot dataset with five-way, one-shot (5N1S, a) and twenty-way, one-shot (20N1S, b) experiments. **c**, Histogram of the accuracy of different techniques using the shuffle-MNIST dataset. The right side shows the average performance over fifty tasks. Note that all models are based on spiking neurons and slightly differ from that in other published work. **d**, The energy consumptions of HP model on the MNIST datasets. Our model can flexibly deploy multiple spike coding schemes on the Tianjic chip and achieve low-cost inference. Critically, the energy consumption increases slowly as the network size expands owning to the spike-based paradigm and local-memory structure. **e**. The comparison of inter-core communication resources in three different learning modes on MNIST datasets. **f**. The energy evaluation for on-chip multitask learning in three different learning modes. The model accuracies on Tianjic chips are shown in the Table S2.

Interpretation of hybrid effectiveness

We further analyse the model effectiveness from a theorical interpretation. Because the learning of the hybrid model is affected by both the external supervision error and internal synaptic behaviours, according to different learning circuits, we assume that the overall loss of the hybrid model can be decomposed of an explicit classification loss and an implicit loss driven by the network dynamics. Then we remodel the local weight update from the perspective of the optimization, and explain the model effectiveness from the approximate regularization and metric learning (see *Method*).

If we consider the local weight increment as a derivative of the implicit loss function, it can approximately act as what we term structured regularization over the network topological structure and temporal dynamics. Specifically for the fault tolerance test, we integrated the local weight increment and derived the corresponding implicit loss in the form of an energy function[40,41] in which Hebbian-based operations can encode the previous patterns into a local minimum. We note that in the inference phase, the local module can gradually decrease this energy cost over time, and further force the network to move into lower energy states, thereby associating with the previous appeared data features.

We deliberate from the perspective of metric learning to discuss the model performance on few-shot learning. By clamping the label signals into the local module, a constraint is placed with respect to the distribution of classes in the metric space. We prove that Hebbian-based computation, as the kernel of the local module, can project an input pattern into a cosine-based embedding space and further produce a simple inductive bias by measuring the similarity between the query sample and the centres of each previously appearing samples. In this manner, the network is forced to learn from the embedding space representations to make the distance between samples within a class sufficiently small while the distance between samples from different classes sufficiently large.

Through the above analysis, we demonstrated that the LP and GP modules complement each other to form the hybrid model. An interesting finding is that we apply the hybrid model to different tasks only with minor modifications to the local module. It may provide other ways for the design of loss functions. Considering that our brain prominently uses local learning to perform tasks, transferring a part of the design of loss functions to local modules is instructive and can bring benefits from at least two-folds: (1) it can reduce the number of manually-designed hyperparameters in the overall loss functions, such as converting the original regularized weighting coefficients to model the learning rates of the local module; (2) the local-based operation endows a network with on-line and low-power properties, which especially facilitates the implementation on many-core neuromorphic hardware by virtue of the fine-grained parallelism architecture (see *Discussion*).

4 Discussion and conclusion

Computational efficiency on the Tianjic neuromorphic platform

The HP SNN, inspired by the membrane dynamic and synaptic dynamics, is essentially a combinational model that aims at leveraging the benefits of the neuro-inspired and machine learning approaches to advance the learning intelligence and computational efficiency. In addition to the potentials that we have shown in dealing with many general tasks, another important benefit is the computational efficiency on the neuromorphic computing, which fully utilizes the spike-based mechanism of neural coding and computations to achieve massive parallelism and remarkable energy efficiency[51].

By instantiating our model on the Tianjic neuromorphic chips (see *Methods*), we showed that the proposed model can flexibly deploy rate-based and temporal-based coding schemes (Fig. 5d) to meet different requirements for accuracy and inference latency (Table S2). As the network size increases, the energy consumption scales very slowly in the neuromorphic hardware owning to the spike-based paradigm and local-memory structure (Fig. 5d). Most importantly, by virtue of the neuromorphic hardware architecture, our model can significantly alleviate the overall power consumption and achieves orders of magnitude faster speed than that of the general-purpose computer (Table S2).

Furthermore, we evaluated the computational resources of on-chip hybrid learning based on the Tianjic's structure and RRAM simulations. Taking the multitask learning as an example, we demonstrated the promise of implementing the hybrid model on such many-core hybrid architectures. Note that in addition to the complementarity in performance, the GP and LP learning also complement each other in the computational resources due to the different updating methods (Table S1). The proposed hybrid model supports the flexibly configurations of these two learning on different connections, which can facilitate the deployment of computational resources. Since only a small number of weight connections are used to receive task-specific supervision signals, the workloads of inter-core communications on many-core architectures can be significantly alleviated (Fig. 5e), and the LP learning can be further deployed in core-in resources by utilizing the decentralized many-core architecture[12,51]. With the highly parallel and near-memory computing architecture of Tianjic-like architecture, the hybrid model can efficiently realize LP learning and further facilitate on-chip learning with a low-energy budget (Fig. 5f).

Summary

In an attempt to design efficient learning algorithms, modelling efforts have usually focused either on single brain-inspired local plasticity or global-based learning. Few studies have explored how to coordinate these two such learning paradigms to synergistically solve problems and improve efficiency. In this study, we presented a two-phase global-local hybrid approach and demonstrated its effectiveness by combining Hebbian-based learning and backpropagation-based learning in one model. The competitive accuracy and efficiency when performing several common image classification tasks show that this model can effectively combine these two types of learning, resulting in stable convergence and compensation. More importantly, we demonstrated that the proposed model can handle complex and general tasks by three examples, including dealing with incomplete data, learning with few-shot examples and learning multiple tasks. Finally, we explained the role of the local module from the perspective of the implicit overall loss function and demonstrated the high efficiency on the Tianjic neuromorphic systems. This work opens a new avenue of learning for general task, which could be a stepping stone towards building neural networks with superior learning capabilities and high efficiency for the development of AGI.

5 Methods

Model establishment

Hybrid plastic approach is based on two sets of differential equations. The first set describes the membrane potential dynamics as follows,

$$\tau_u \frac{du_i^l}{dt} = -u_i^l(t) + \sum_{j=1}^{l_n} w_{ij}^l(t) s_j^{l-1}(t), \tag{2}$$

$$s_j^{l-1}(t) = \sum_{t_j^{l-1,f}<t} \delta(t - t_j^{l-1,f}), \tag{3}$$

where w_{ij}^l denotes the weight of the synapse connecting pre-neuron j and post-neuron i, u_i^l denotes the membrane potential of neuron i, τ_u denotes the membrane time constant, $s_j^{l-1}(\mathrm{t})$ denotes the afferent spike trains, $t_j^{l-1,f}$ denotes the firing time, and l_n denotes the number of neurons in the l_{th} layer.

The second set of equations describes the synaptic weight dynamics w_{ij}^l, modelled by

$$\tau_w \frac{dw_{ij}^l}{dt} = w_{ij}^{l,e} - w_{ij}^l(t) + P\big(t,,pre_j^l(t), post_i^l(t), w_{ij}^l; \boldsymbol{\theta}^l\ \big), \tag{4}$$

where τ_w denotes the synaptic constant. The first term $w_{ij}^{l,e} - w_{ij}^l(t)$ in the right of equation (4) denotes the recovery of $w_{ij}^l(t)$ into a ground state $w_{ij}^{l,e}$, which we set zero in the experiments. The second term $P(*)$ represents a general local plasticity controlled by presynaptic spike activity, $pre_j^l(t) \triangleq \{u_j^l(t), s_j^l(t)\}$, postsynaptic spike activity, $post_i^l \triangleq \{u_i^l(t), s_i^l(t)\}$, and a group of controllable factors $\boldsymbol{\theta^l}$. Note that we use the bold to represent vectors in this context.

As $P(*)$ is generic for modelling local learning rule, we take a specific expression, a variant of Hebbian rule in the experiments, which is formulized by

$$P\big(t, pre_j^l(t), post_i^l(t); \boldsymbol{\theta}^l\ \big) \triangleq k^{l,corr} s_j^{l-1}(t)\big(\rho\ (u_i^l(t))\ + \beta_i^l\big), \tag{5}$$

where $k^{l,corr}$ is a weight parameter, $\rho(x)$ is a potentially nonlinear function, and $\beta_i^l \leq 0$ is an optional sliding threshold to control weight change directions and prevent weight explosions. It therefore can update the weight according to concurrent presynaptic firing and postsynaptic membrane activity. In the case of $\beta_i^l = 0$, equation (5) reduces to the Hebbian rule. Integrating the equation (4), we get

$$w_{ij}^l(t) = w_{ij}^l(t_{n0}) e^{-\frac{t-t_{n0}}{\tau_w}} + \int_{t_{n0}}^{t} P\big(x, pre_j^l(x), post_i^l(x), \boldsymbol{\theta}^l\big)\, e^{-\frac{x-t}{\tau_w}} dx, \tag{6}$$

where $\mathrm{w}_{\mathrm{ij}}^{\mathrm{l}}(\mathrm{t}_{\mathrm{n0}})$ denotes an initial condition of synaptic weight at the phase time t_{n0}. Because the

HP method uses a potential trajectory rather than an equilibrium state for computation, the dependence on the initial parameter $\mathrm{w}_{\mathrm{ij}}^{\mathrm{l}}(\mathrm{t}_{\mathrm{n0}})$ is non-trivial. Based on it, we assume that HP model can perform supervised learning through modifying the phasic values with supervision signals at certain time intervals, the role of which is similar to an instantaneous current. Consequently, we substitute the synaptic equation (6) into the membrane potential equation (2) by

$$\tau_u \frac{du_i^l}{dt} = -u_i^l + \sum_{j=1}^{l_n} s_j^{l-1}(t)\, w_{ij}^l(t_{n0}) e^{-\frac{t-t_{n0}}{\tau_w}} + \sum_{j=1}^{l_n} \left(\int_{t_{n0}}^{t_m} P\left(x; pre_j^l(x), post_i^l(x), \boldsymbol{\theta}^l\right) e^{-\frac{x-t}{\tau_w}} \right) s_j^{l-1}(t) dx. \quad (7)$$

To enable the continuous dynamics compatible with backpropagation-based methods, we use a modified Euler method to get an explicit iterative version of equation (7)

$$\begin{aligned} \tau_\mathrm{u} \frac{u_i^l(t_{m+1}) - u_i^l(t_m)}{h} &= -u_i^l(t_m) + \sum_{j=1}^{l_n} s_j^{l-1}(t_m)\, w_{ij}^l(t_{n0}) e^{-\frac{t_m-t_{n0}}{\tau_w}} \\ &+ \sum_{j=1}^{l_n} s_j^{l-1}(t_m) \int_{t_{n0}}^{t_m} P\left(x, pre_j^l(x), post_i^l(x), \boldsymbol{\theta}^l\right) e^{-\frac{x-t}{\tau_w}} dx. \end{aligned} \quad (8)$$

where we use t_m to refer to the simulation timestep. Sorting the formula and substituting the specific expression of $P(x)$, we can get a final signal propagation equation as follows

$$\begin{cases} u_i^l(t_{m+1}) = k_u' u_i^l(t_m) + k_u \sum_{j=1}^{l_n} s_j^{l-1}(t_m)\, w_{ij}^l(t_{n0}) e^{-\frac{t_m-t_{n0}}{\tau_w}} + \\ \qquad k_u \sum_{j=1}^{l_n} s_j^{l-1}(t_m) \sum_{t_i^f, t_j^f > t_{n0}} k_{ij}^{l,corr} \delta_j^{l-1}(t_m - t_j^f)(\rho(u_i(t_m)) - \beta_j) e^{-\frac{t_m - t_j^f}{\tau_w}}, \\ s_i^l(t_m) = \psi(u_i^l(t_m) - v_{th}), \end{cases} \quad (9)$$

where $k_u \triangleq \frac{h}{\tau_u}, k_u' \triangleq 1 - k_u$, and $\psi(x)$ is the firing function determined by the Heaviside function. Specifically, if $u_i^l(t_m)$ exceeds the threshold v_{th}, $\psi(x) = 1$, otherwise $\psi(x) = 0$. Regarding the non-differential points of $\psi(x)$, we take a regular function to approximate its derivative by following the work[52,53].

Finally, to make the expression clearly, we replace the summation of local activity into an iterative variable $P_{ij}^l(t_m)$, relax $k_{ij}^{l,corr}$ by two elastic regular factors, $k_{ij}^{l,corr} = \alpha_i^l \eta_j^l$, and transform equation (8) into an iterative version,

$$\begin{cases} u_i^l(t_m) = k_u' u_i^l(t_{m-1}) + k_u \sum_{j=1}^{l_n} \left(w_{ij}^l(t_0) e^{\frac{t_n - t_m}{\tau_w}} + \alpha_i^l P_{ij}^l(t_m) \right) s_j^{l-1}(t_m), \\ P_{ij}^l(t_m) = \gamma^l P_{ij}^l(t_{m-1}) + \eta_j^l s_j^{l-1}(t_m)\left(\rho(u_i^l) + \beta_i^l\right), \\ s_i^l(t_m) = \psi\left(u_i^l(t_m) - v_{th}\right), \end{cases} \quad (10)$$

where, $\gamma^l \triangleq e^{-\frac{dt}{\tau_w}}$, $\mathrm{d}t$ denotes the length of timestep, α^l controls the impact of local modules and

η_j^l controls the local learning rate. For classification output, we take a one-hot encoding to use N neurons of the output layer for each class. Then we incorporate different spike coding schemes into a general framework and describe the classification loss function C by

$$C(\boldsymbol{w},\boldsymbol{\theta}) \triangleq L\big(\boldsymbol{y}, \textstyle\sum_{m=1}^{T} z_{t_m} q(\boldsymbol{u}^{n_l}(t_m))\big), \tag{11}$$

where $\boldsymbol{y}$ is the ground truth, n_l is the number of layers, T denotes the simulation windows, L is a common classification loss (such as the mean square error), $q(x)$ is an increasing bounded function depending on specific coding schemes, $z_{t_m} \in R_{\geq 0}$ is the weight associated with time-step t_m. This formulization can adapt to the rate-based coding when $z_{t_m} = 1/T$, $q(x) = \psi(x)$, and adapt to the rank-based coding when $z_{t_m} = \mathbf{1}[t_m = T]$.

Given the signal propagation equations (10) and a specific decoding scheme (11), we can optimize the network parameters, $\boldsymbol{\theta}$ and $\boldsymbol{w}$, through minimizing the loss C using the Backpropagation Through Time (BPTT) algorithm[54]. Since the meta-parameter $\boldsymbol{\theta}$ is a special type of hyper-parameters to control the learning behaviours of $\boldsymbol{w}$, we exclusively establish the optimization of $\boldsymbol{\theta}$ and formulize a general expression as follows

$$\min_{\boldsymbol{\theta}} \textstyle\sum_{\pi_i \in \Gamma} C_{\pi_i}(\boldsymbol{w}^*, \boldsymbol{\theta}), \mathrm{s.t.}, \boldsymbol{w}^*(\theta) = \arg\min_{\boldsymbol{w}} C_{\pi_i}(\boldsymbol{w}^*, \boldsymbol{\theta}), \tag{12}$$

where the task $\pi_i \triangleq \{L(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{w}|\boldsymbol{\theta}), \{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}\}$, consists of a certain loss function L and a set of training data $\{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}_k$. Γ refers to a task distribution set, and it reduces to a single task set for ordinary classification problems. The precise evaluation of the optimal weight $\boldsymbol{w}^*$ satisfying the equation (12) is usually prohibitive due to the expensive computation. Following the related works[42,55,56] on the gradient-based hyperparameter optimization, we approximate the $\boldsymbol{w}^*$ by adapting $\boldsymbol{w}$ using its multiple-step iteration values. Specifically, we iterate the parameters of $\boldsymbol{w}$ several steps, use the iterative values in place of $\boldsymbol{w}^*$, and alternately optimize the $\boldsymbol{\theta}$ by solving equation (12).

Linking the HP SNNs with rank order coding

In the following, we give an overview of rank order coding[29] and analysis its relationship with the HP model thereon.

Rank order coding assumes that the biological neurons encode information by the firing orders across a neuron population. Suppose target neuron i receives inputs from a presynaptic neuron population A_l, and each neuron only fires a spike once. Let activations of afferent neurons be a_j^l.

Then the rank order coding can record the relative firing orders of afferent neurons, and updates the activation of a_i^{l+1} by,

$$a_i^{l+1} = \sum_{j\in A_l} r^{order(a_j^l)} w_{ij}^{l+1}, \tag{13}$$

where $r \in (0,1)$ is a given punishment constant, $order(a_j^l)$ is the firing order of neuron j in the presynaptic population. Equation. (13) shows that the ranking factor $r^{order(a_j^l)}$ is a key for the rank order coding, which can encourage the early firing of neuron meanwhile punish the later firing of neuron. As the encoded information is prominent in earlier spikes, this coding schemes is more ready for fast-decision and network sparsity[29]. Next, we show that the ranking factor can be equivalently converted into the decay function of our model, and therefore the information propagation of HP model encodes information by equation (13).

Theorem 1 *Assume each neuron fires at most one spike in a given short time window. The HP model encodes information in the form of rank order coding.*

Proof. According to the assumption, we first formulize the input current I of equation. (13) as

$$I = \sum_{j=1}^{l_n} r^{order(s_j^l)} w_{ij}^{l+1}\, s_j^l, \tag{14}$$

where we add a spike signal $s_j^{l-1} \in \{0,1\}$ to incorporate all presynaptic neurons and make the updating compatible with our neuron update equations. By formula deformations, it holds

$$I = \sum_{j=1}^{l_n} w_{ij}^{l+1} e^{log(r) order(s_j^l)} s_j^l = \sum_{j=1}^{l_n} w_{ij}^{l+1} e^{-\frac{order(s_j^l)}{-\frac{1}{log(r)}}}\, s_j^l = \sum_{j=1}^{l_n} (w_{ij}^{l+1} e^{-\frac{t_m - t_0}{\tau_w}}) s_j^l, \tag{15}$$

where $\tau_w = \frac{1}{-log\,(r)}$, $order(s_j) = t_m - t_0$. In this manner, the ranking factor $r^{order(s_j^l)}$ can be equivalently converted into the decay function of the HP method.

Model interpretation

We further consider the impact of local module as a form of implicit loss and interpret the hybrid model from the perspective of optimization. Because the learning of the hybrid model is affected by the supervision signals and internal dynamics, accordingly, its overall loss function F is more likely not only incorporating the classification loss C, but also building on an energy function E_{in} generated by the inherent dynamics of network. According to the different learning circuits, we first make the decomposability assumption on the general overall loss as follows

$$F \triangleq C\big(t, \boldsymbol{x}, \boldsymbol{y}; \tilde{\alpha} \boldsymbol{w_{GP,t}}, \tilde{\beta} \boldsymbol{w_{LP,t}}, \boldsymbol{\theta}\big) + \tilde{\gamma} \sum_{l=1}^{n_l} E_{in}(t, pre^l, post^l, \boldsymbol{w}^l, \boldsymbol{\theta}), \tag{16}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ is external input data, n_l denotes the total number of layers, and $\tilde{\alpha}^l, \tilde{\beta}^l, \tilde{\gamma}^l \in R_{\geq 0}$ denote the influential factor of each part. We follow the notations of equations (1) and in a slight abuse of notation, we explicitly express the composition $\tilde{\alpha}^l \boldsymbol{w}^l_{\boldsymbol{GP,t}}, \tilde{\beta}^l \boldsymbol{w}^l_{\boldsymbol{LP,t}}$ on C to highlight the difference between the hybrid model and single learning-based model. Interestingly, we note that the similar decomposition hypothesis can be supported by the investments on brain learning assumptions[57,58], and here we based on it for the development of the hybrid model. Indeed, the expression of F can be regarded as an extension of single learning-based network. In the case of $\tilde{\beta}^l = \tilde{\gamma} = 0$, F degenerates to the conventional classification loss for the GP-based network, and in the case of $\tilde{\alpha}^l = 0$, the network reduces to the LP-based network.

5.1.1 Approximate regularization

If we treat the local weight increment as an implicit derivative for a part of the overall loss function, it inspires us to integral its weight increment to obtain the implicit loss function E_{in}. However, not all local weight rules can easily derive an integral function. Besides that, most local rules are approximate and abstract for biological learning. Thus, rather than to obtain the precise integral solution of local increments, we are more interested in simplifying the derivation and finding the task-related parts in overall loss functions. In particular, we mainly consider the impact of local weight increment on the current layer, and integral it to obtain a corresponding implicit loss. Then we note crucial components in the overall loss function to current tasks and illustrate that the local learning module can progressively decreases this component with iterations. For example, in the fault tolerance learning, we first assume that the local weight increment $\Delta w^l_{ij,t}$ is proportional to the derivative of E^l_{in} by

$$\frac{\partial E_{in}}{\partial w^l_{ij,t}} \propto -\Delta w^l_{ij,t} \propto \ -s^l_{j,t}(\rho(u^{l+1}_{i,t}) \ + \beta^{l+1}_i).$$

Integrating above equation, we can get a loss expression E_{in} as follows,

$$E_{in} \cong \ - \ \sum_{t=1}^{T} \sum_{l=2}^{n_l} \left({\boldsymbol{s}^l_t}^T \boldsymbol{w}^{\boldsymbol{l}}_{\boldsymbol{t}} \boldsymbol{\rho}(\boldsymbol{u}^l_t) + {\boldsymbol{\beta}^l}^T \boldsymbol{u}^l_t \right), \tag{17}$$

In fact, a loss function in the above form of E_{in} can be viewed as a specific energy function[40,41], in which the Hebbian-based operation can encode the previous pattern into a local minimum and the dynamical update can decrease the energy surface at every update. Thus, by implicitly optimizing this surface, local modules relax the hierarchical representation of networks to local minimum states that are more likely to encode previous associative patterns, thereby exploiting the correlation embedded in the training examples.

5.1.2 Adding inductive biases of prior knowledge

We mainly analyse the impact of local module in the last two layers. Assume that we have a set of training samples $S = \{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}_{k=1}^{N_S}$ where $\boldsymbol{x_k} \in R^{m\times 1}$ is an m-dimensional feature vector, $\boldsymbol{y}_k \in R^{n\times 1}$ is the one-hot label, N_S denotes the sample number of *S*, and S_k is the set of examples within the class *k*. We refer $\boldsymbol{x}_k$ to the general features coming from raw data or the last $n_l - 1$ layer. The kernel operation of the local module is to construct the Hebbian-like matrix by the outer product

$$H = \sum_{k\in S} \eta_k \boldsymbol{y}_k \boldsymbol{x}_k^T = \sum_{k=1}^{K} (\boldsymbol{y}_k \sum_{i=1}^{N_{S_k}} \eta_i \boldsymbol{x}_i^T) = \sum_{k=1}^{K} \boldsymbol{y}_k \boldsymbol{c}_k^T, \tag{18}$$

where *K* denotes the class number of *S*. Here we set the learning rate $\eta_i = \frac{1}{N_{S_k}\|x_i\|_2}$ and refer $\boldsymbol{c}_k \triangleq \frac{1}{N_{S_k}} \sum_{i=1}^{N_{S_k}} \frac{x_i}{\|x_i\|_2}$ to the sample mean of the class S_k. To keep the clarity of proof, we simplify the modelling of meta-parameters, such as the decay factor and the gating parameter. Based on the equation. (18), when entering a query sample $\boldsymbol{x}_t$, the local module produces an inductive bias I_{LP} by

$$I_{LP}(\boldsymbol{x}_t) = H f_{\phi_{l_n-1}}(\boldsymbol{x}_t) = \sum_{k=1}^{K} \boldsymbol{y}_k (\boldsymbol{c}_k^T f_{\phi_{l_n-1}}(\boldsymbol{x}_t)) = \sum_{k=1}^{K} \boldsymbol{y}_k d\left(\boldsymbol{c}_k, f_{\phi_{l_n-1}}(\boldsymbol{x}_t)\right),$$

where $d(\boldsymbol{c}_k, f_{\phi_{l_n-1}}(\boldsymbol{x}_t)) \triangleq \cos\left(\boldsymbol{c}_k, f_{\phi_{l_n-1}}(\boldsymbol{x}_t)\right) \|c_k\|_2 \left\|f_{\phi_{l_n-1}}(\boldsymbol{x}_t)\right\|_2$ measures the similarity between the sample centres $\boldsymbol{c}_k$ and the activations $f_{\phi_{l_n-1}}(\boldsymbol{x}_t)$ in the $l_n - 1$ layer. Then the membrane potential of output neuron is governed by

$$u_i^{l_n}(t_m) = \tilde{\alpha} I_{GP,i}(t_m) + (1-\tilde{\alpha}) I_{LP,i}(t_m) = \tilde{\alpha} \sum_{j=1}^{l_n-1} w_{ij} f_{\phi_{l_n-1}}\left(x_j(t_m)\right) + (1-\tilde{\alpha}) \sum_{k=1}^{K} y_{k,i}\, d\left(\boldsymbol{c}_k, f_{\phi_{l_n-1}}(\boldsymbol{x}_t)\right), \tag{19}$$

where $y_{k,i}$ is the i element of the one-hot label $\boldsymbol{y}_k$, which satisfies that $y_{k,i} = \mathbf{1}_{\{k\}}(i)$. We follow the notations of equation. (16) and set $\tilde{\alpha}$ with a small positive value to strengthen the impact of I_{LP} and simplify analysis. In this manner, we can give an intuitive interpretation for the inductive bias I_{LP} from the Euclidean distance L_ϕ between the label $\boldsymbol{y}$ and $\boldsymbol{u}^{l_n}$. Assuming that the input pattern belongs to i class, L_ϕ can be calculated by

$$L_\phi(\boldsymbol{u}^{l_n}, \boldsymbol{y}) = \left(1 - \left((\tilde{\alpha} I_{GP,i} + (1-\tilde{\alpha}) d(\boldsymbol{c}_i, f_{\phi_{l_n-1}}(\boldsymbol{x}))\right)\right)^2 + \sum_{q\neq i} \left(\tilde{\alpha} I_{GP,q} + (1-\tilde{\alpha}) d\left(\boldsymbol{c}_q, f_{\phi_{l_n-1}}(\boldsymbol{x})\right)\right)^2$$

$$= \left(\left(1 - d\left(\boldsymbol{c}_i, f_{\phi_{l_n-1}}(\boldsymbol{x})\right)\right) - \tilde{\alpha}((I_{GP,i} - d(\boldsymbol{c}_i, f_{\phi_{l_n-1}}(\boldsymbol{x})))\right)^2 + \sum_{q\neq i} \left(\tilde{\alpha} I_{GP,q} + (1-\tilde{\alpha}) d\left(\boldsymbol{c}_q, f_{\phi_{l_n-1}}(\boldsymbol{x})\right)\right)^2, \tag{20}$$

where we set $T = 1$ and omit the index t for neatness. Note that the $\tilde{\alpha}$ is a pre-defined small amount, thus, the L_ϕ has the two main components $\left(1 - d(\boldsymbol{c}_i, f_{\phi_{l_{n-1}}}(\boldsymbol{x}))\right)^2$ and $\left((1 - \tilde{\alpha})d(\boldsymbol{c}_q, f_{\phi_{l_{n-1}}}(\boldsymbol{x}))\right)^2$. To minimize the distance L_ϕ, the network is forced to learn the specific feature mapping that projects the distance between samples within a class sufficiently small (by the punishment of $\left(1 - d(\boldsymbol{c}_i, f_{\phi_{l_{n-1}}}(\boldsymbol{x}))\right)^2$) while the distance between samples from different classess sufficiently large (by the punishment of $d^2(\boldsymbol{c}_q, f_{\phi_{l_{n-1}}}(\boldsymbol{x}))$). In this manner, the inductive bias I_{LP} enables the hybrid model to learn from the feature similarity between the query data and the previous feature centres in the training data in the metric space.

Task-related details

5.1.3 Basic performance evaluation

In the MNIST and Fashion-MNIST experiments, we used Bernoulli sampling to encode the pixel into spike data. In the CIFAR10 and CIFAR100 dataset, following the work[53], we used the first spiking layer as an encoding layer to produce spike signals. We optimized the cross entropy in CIFAR100, and optimized the mean square error in MNIST, Fashion-MNIST and CIFAR10. A six-layer network with [input-128C3-AP2-256C3-AP2-256C3-AP2-512FC-10] was trained in MNIST and Fashion-MNIST datasets, and a nine-layer spiking CNN with CIFARNet structure[53] was trained in CIFAR10 and CIFAR100. To reduce computation, we equipped the local synaptic module in the hidden fully-connected layers in image classification experiments.

Details of Hardware implementation

Neuromorphic computing systems aim to emulate biological neural coding and computation in hardware by distributing memory and computation in a large number of self-contained computational units. The sparse spike-triggered paradigm and the colocation of memory and compute in neuromorphic systems lead to a highly efficient computing paradigm, and the decentralized many-core architecture can significantly increase the parallelism, thereby providing orders of magnitude greater efficiency than general-purpose computers. Tianjic, a unified network computing platform, is an advanced many-core neuromorphic system with innovations in hybrid computing paradigm and architecture. In the following, we give a brief overview of Tianjic neuromorphic systems and our model implementations thereon.

6 Reference

1 LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436-444, doi:10.1038/nature14539 (2015).

2 Tom, Y., Devamanyu, H., Soujanya, P. & Erik, C. Recent Trends in Deep Learning Based Natural Language Processing *IEEE Computational Intelligence Magazine* **13**, 55-75 (2018).

3 Silver, D. *et al.* Mastering the game of Go without human knowledge. *Nature* **550**, 354-359 (2017).

4 Johansen, J. P. *et al.* in *Proceedings of the National Academy of Sciences* Vol. 111 E5584-E5592 (2014).

5 Yger, P., Stimberg, M. & Brette, R. Fast learning with weak synaptic plasticity. *Journal of Neuroscience* **35**, 13351-13362 (2015).

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

7 Henry, M. The blue brain project. *Nature Reviews Neuroscience* **7**, 153-160 (2006).

8 Song, S., Miller, K. D. & Abbott, L. F. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience* **3**, 919-926 (2000).

9 Eliasmith, C. *et al.* A large-scale model of the functioning brain. *Science* **338**, 1202-1205, doi:10.1126/science.1225266 (2012).

10 Hassabis, D., Kumaran, D., Summerfield, C. & Botvinick, M. Neuroscience-Inspired Artificial Intelligence. *Neuron* **95**, 245-258, doi:10.1016/j.neuron.2017.06.011 (2017).

11 Marblestone, A. H., Wayne, G. & Kording, K. P. Toward an integration of deep learning and neuroscience. *Frontiers in computational neuroscience* **10**, 94 (2016).

12 Pei, J. *et al.* Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature* **572**, 106-111, doi:10.1038/s41586-019-1424-8 (2019).

13 Bengio, Y., Mesnard, T., Fischer, A., Zhang, S. & Wu, Y. STDP-Compatible Approximation of Backpropagation in an Energy-Based Model. *Neural Comput* **29**, 555-577, doi:10.1162/NECO_a_00934 (2017).

14 Hinton, G. in *Invited talk at the NIPS'2007 deep learning workshop.*

15 Zenke, F., Poole, B. & Ganguli, S. in *Proceedings of the 34th International Conference on*

Machine Learning Vol. 70 3987-3995 (2017).

16 Bengio, S., Bengio, Y., Cloutier, J. & Gecsei, J. in *Preprints Conf. Optimality in Artificial and Biological Neural Networks.* 6-8 (Univ. of Texas).

17 Miconi, T., Clune, J. & Stanley, K. O. in *International Conference on Machine Learning* (2018).

18 Lee, C., Panda, P., Srinivasan, G. & Roy, K. Training Deep Spiking Convolutional Neural Networks With STDP-Based Unsupervised Pre-training Followed by Supervised Fine-Tuning. *Front Neurosci* **12**, 435, doi:10.3389/fnins.2018.00435 (2018).

19 Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J. & Masquelier, T. STDP-based spiking deep convolutional neural networks for object recognition. *Neural Netw* **99**, 56-67, doi:10.1016/j.neunet.2017.12.005 (2018).

20 Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T. & Maida, A. Deep learning in spiking neural networks. *Neural Networks* **111**, 47-63 (2019).

21 Nicolas, F. & Wulfram, G. Neuromodulated Spike-Timing-Dependent Plasticity, and Theory of Three-Factor Learning Rules. *Frontiers in Neural Circuits* **9** (2016).

22 Gerstner, W., Lehmann, M., Liakoni, V., Corneil, D. & Brea, J. Eligibility Traces and Plasticity on Behavioral Time Scales: Experimental Support of NeoHebbian Three-Factor Learning Rules. *Front Neural Circuits* **12**, 53, doi:10.3389/fncir.2018.00053 (2018).

23 Lisman, J., Grace, A. A. & Duzel, E. A neoHebbian framework for episodic memory; role of dopamine-dependent late LTP. *Trends Neurosci* **34**, 536-547, doi:10.1016/j.tins.2011.07.006 (2011).

24 Johansen, J. P., Cain, C. K., Ostroff, L. E. & LeDoux, J. E. Molecular mechanisms of fear learning and memory. *Cell* **147**, 509-524, doi:10.1016/j.cell.2011.10.009 (2011).

25 Kuśmierz, Ł., Isomura, T. & Toyoizumi, T. Learning with three factors: modulating Hebbian plasticity with errors. *Current Opinion in Neurobiology* **46**, 170-177 (2017).

26 Wilmes, K. A. & Clopath, C. Inhibitory microcircuits for top-down plasticity of sensory representations. *Nature communications* **10**, 1-10 (2019).

27 Diehl, P. U., Neil, D., Binas, J., Cook, M. & Liu, S. C. in *Neural Networks (IJCNN), 2015 International Joint Conference on.*

28 Jin, Y., Zhang, W. & Li, P. in *Advances in Neural Information Processing Systems* 7005-7015

(2018).

29 Thorpe, S. & Gautrais, J. in *Computational neuroscience* 113-118 (Springer, 1998).

30 Diehl, P. U. & Matthew, C. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience* **9** (2015).

31 Lee, C., Sarwar, S. S. & Roy, K. Enabling spike-based backpropagation in state-of-the-art deep neural network architectures. *arXiv preprint arXiv:1903.06379* (2019).

32 Zhang, W. & Li, P. in *Advances in Neural Information Processing Systems.* 7800-7811 (2019).

33 Sengupta, A., Ye, Y., Wang, R., Liu, C. & Roy, K. Going Deeper in Spiking Neural Networks: VGG and Residual Architectures. *Front Neurosci* **13**, 95, doi:10.3389/fnins.2019.00095 (2019).

34 Panda, P. & Roy, K. in *2016 International Joint Conference on Neural Networks (IJCNN).*

35 Hunsberger, E. & Eliasmith, C. Training Spiking Deep Networks for Neuromorphic Hardware. *arXiv preprint arXiv:1611.05141.* (2017).

36 Severa, W., Vineyard, C. M., Dellana, R., Verzi, S. J. & Aimone, J. B. Training deep neural networks for binary communication with the Whetstone method. *Nature Machine Intelligence* **1**, 86-94, doi:10.1038/s42256-018-0015-y (2019).

37 Kietzmann, T. C., McClure, P. & Kriegeskorte, N. Deep neural networks in computational neuroscience. *BioRxiv*, 133504 (2018).

38 Tang, J., Liu, J., Zhang, M. & Mei, Q. Visualizing Large-scale and High-dimensional Data.

39 Acevedo-Mosqueda, M. E., Yáñez-Márquez, C. & Acevedo-Mosqueda, M. A. Bidirectional associative memories: Different approaches. *ACM Computing Surveys (CSUR)* **45**, 18 (2013).

40 Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci U S A* **79**, 2554-2558, doi:10.1073/pnas.79.8.2554 (1982).

41 Watson, R. A., Buckley, C. & Mills, R. The effect of Hebbian learning on optimisation in Hopfield networks. (2009).

42 Finn, C., Abbeel, P. & Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *Proceedings of the 34th International Conference on Machine Learning* **70**, 1126-1135 (2017).

43 Koch, G., Zemel, R. & Salakhutdinov, R. in *ICML deep learning workshop* Vol. 2 (2015).

44 Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K. & Wierstra, D. Matching Networks

for One Shot Learning. *Advances in neural information processing systems*, 3630-3638 (2016).

45 Kaiser, Ł., Nachum, O., Roy, A. & Bengio, S. Learning to remember rare events. *arXiv preprint arXiv:1703.03129* (2017).

46 Lake, B. M., Salakhutdinov, R. & Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science* **350**, 1332-1338 (2015).

47 Cichon, J. & Gan, W.-B. Branch-specific dendritic Ca2+ spikes cause persistent synaptic plasticity. *Nature* **520**, 180-185.

48 Yang, G., Pan, F. & Gan, W.-B. Stably maintained dendritic spines are associated with lifelong memories. *Nature* **462**, 920-924.

49 Masse, N. Y., Grant, G. D. & Freedman, D. J. Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization. *Proceedings of the National Academy of Sciences* **115** (2018).

50 Kirkpatrick, J. *et al.* Overcoming catastrophic forgetting in neural networks. *Proc Natl Acad Sci U S A* **114**, 3521-3526.

51 Imam, N. & Cleland, T. A. Rapid online learning and robust recall in a neuromorphic olfactory circuit. *Nature Machine Intelligence* **2**, 181-191 (2020).

52 Wu, Y., Deng, L., Li, G., Zhu, J. & Shi, L. Spatio-Temporal Backpropagation for Training High-Performance Spiking Neural Networks. *Front Neurosci* **12**, 331, doi:10.3389/fnins.2018.00331 (2018).

53 Wu, Y. *et al.* in *Proceedings of the AAAI Conference on Artificial Intelligence.* 1311-1318.

54 Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* **78**, 1550-1560 (1990).

55 Liu, H., Simonyan, K. & Yang, Y. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055* (2018).

56 Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R. & Pontil, M. Bilevel programming for hyperparameter optimization and meta-learning. *arXiv preprint arXiv:1806.04910* (2018).

57 Scellier, B. & Bengio, Y. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience* **11**, 24 (2017).

58 Marblestone, A. H., Wayne, G. & Kording, K. P. Toward an Integration of Deep Learning

and Neuroscience. *Front Comput Neurosci* **10**, 94, doi:10.3389/fncom.2016.00094 (2016).

59 Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

60 Deng, L. *et al.* Tianjic: A Unified and Scalable Chip Bridging Spike-Based and Continuous Neural Computation. *IEEE Journal of Solid-state Circuits*, 1-19 (2020).

7 Supplementary materials

Table S1: Comparisons between the hybrid method and single learning methods.

Method	Learning source	Update	Memory	Bio-fidelity	Accuracy	Few-shot learning	Learning capacity
LP	Neuronal activity driven	Asynchronous, Adjacent layers	Associative memory	High	Low	N/A	N/A
GP	Data driven	Synchronous, Across networks	Representation mapping for data	Low	High	Poor	Poor
HP	Both	Both	Both	High	High	Good	Good