Mathematical Biology

Learning spiking neuronal networks with artificial neural networks: neural oscillations

Ruilin Zhang[1,2] · Zhongyi Wang[1,3] · Tianyi Wu[1,3] · Yuhang Cai[4] · Louis Tao[1,5] · Zhuo-Cheng Xiao[6] · Yao Li[7] ⬤

Abstract

First-principles-based modelings have been extremely successful in providing crucial insights and predictions for complex biological functions and phenomena. However, they can be hard to build and expensive to simulate for complex living systems. On the other hand, modern data-driven methods thrive at modeling many types of high-dimensional and noisy data. Still, the training and interpretation of these data-driven models remain challenging. Here, we combine the two types of methods to model stochastic neuronal network oscillations. Specifically, we develop a class of artificial neural networks to provide faithful surrogates to the high-dimensional, nonlinear oscillatory dynamics produced by a spiking neuronal network model. Furthermore, when the training data set is enlarged within a range of parameter choices, the artificial neural

Ruilin Zhang and Zhongyi Wang have equal contribution to this work.

✉ Louis Tao
taolt@mail.cbi.pku.edu.cn

✉ Zhuo-Cheng Xiao
xiao.zc@nyu.edu

✉ Yao Li
yaoli@math.umass.edu

1   Center for Bioinformatics, National Laboratory of Protein Engineering and Plant Genetic Engineering, School of Life Sciences, Peking University, Beijing 100871, China

2   Yuanpei College, Peking University, 100871 Beijing, China

3   School of Mathematical Sciences, Peking University, 100871 Beijing, China

4   Department of Mathematics, University of California, 94720 Berkeley, CA, USA

5   Center for Quantitative Biology, Peking University, 100871 Beijing, China

6   Courant Institute of Mathematical Sciences, New York University, 10003 New York, NY, USA

7   Department of Mathematics and Statistics, University of Massachusetts Amherst, 01003 Amherst, MA, USA

networks become generalizable to these parameters, covering cases in distinctly different dynamical regimes. In all, our work opens a new avenue for modeling complex neuronal network dynamics with artificial neural networks.

Keywords Artificial neural network · Gamma oscillations · Data-driven methods · Generalization

Mathematics Subject Classification 92C20 · 68T07 · 60J28

1 Introduction

The last few decades have seen rapid developments of first-principles-based mathematical models to study living systems. Based on a collection of *a priori* physiological and physical principles, the evolution of mathematical models can offer significant advantages in understanding, reproducing, and predicting complex biological phenomena. However, first-principle-based models can be prohibitively expensive to build due to the large number of parameters and variables characterizing the complexity of biological details, e.g., multiple time scales, complicated interactions between biological elements, among others. Alternatively, modern data-driven models focusing on phenomenological or empirical observations are gaining ground in mathematical biology, in that they are designed to deal with high dimensional and noisy data (Janes and Yaffe 2006; Hasenauer 2015; Solle 2017; Jack et al. 2018; AlQuraishi and Sorger 2021). However, one is still faced with the daunting task of making sense of the coordinates and parameters of the data-driven models to identify interpretable and biologically meaningful features.

In this study, we investigate how the combination of the two classes of methods can be used to study spiking neuronal networks (SNNs). SNNs are capable of producing highly nonlinear, high-dimensional, and multi-timescale dynamics, and have been widely used to investigate cognitive functions and their computation principles (see, e.g., Tao 2006; Ghosh-Dastidar and Adeli 2009; Ponulak and Kasinski 2011; Nobukawa et al. 2017; Börgers and Kopell 2003; Chariker et al. 2016). First-principles-based model reduction methods such as coarse-graining and mean-field theories have been developed to better understand SNN dynamics (Wilson and Cowan 1972; Brunel and Hakim 1999; Buice and Cowan 2007; Cai 2006; Cai et al. 2021; Li and Hui 2019). On the other hand, artificial neural networks (ANN, and its offspring, deep neural networks, or DNN) are a modern data-driven method inspired by the nervous system. DNN has been extremely successful in both engineering applications (image processing, natural language processing, etc.) and applied mathematics (parameter estimation, numerical ordinary/partial differential equations, inverse problems, etc.). See Aggarwal et al. (2018); Schmidhuber (2015); Chon and Cohen (1997); Li (2020); Raissi et al. (2019), for instance. In particular, it has been shown recently that DNN can well approximate both finite and infinite-dimensional operators (Barron 1994; Kovachki et al. 2021). The idea of using DNN surrogates in models to replace the firing rate of SNN was first explored by Zhang and Young (2020). This motivates us

to propose **a first-principle-informed deep-learning framework** that replaces the crucial and complex SNN dynamics by the representation of artificial neurons.

The neuroscience problem we address in this paper is the γ-band oscillations, a type of 30–90 Hz oscillatory physiological dynamics prominent in many brain regions (Andrew Henrie and Shapley 2005; Brosch et al. 2002; Bauer 2006; Buschman and Miller 2007; Pieter Medendorp 2007; van Wingerden 2010; Csicsvari 2003; Popescu et al. 2009; Van Der Meer and David Redish 2009). Remarkably, in previous studies, γ-oscillations can be produced in simple, idealized SNN models involving only two neural populations, excitatory (E) and inhibitory (I) (Chariker et al. 2018; Zhang 2014; Rangan and Young 2013; Li et al. 2019). More specifically, due to transient noise and/or external stimulus, highly correlated spiking patterns (previously termed multiple-firing events, or MFEs) are repeatedly produced from the competitions between E/I populations, involving the interplay between multiple timescales. MFEs are a type of stochastic, high-dimensional emergent phenomena, with rapid and transient dynamical features that are sensitive to the biophysical parameters of the network. The strong fluctuation from input and recurrent neuronal interactions hinders the ability of previous mean-field approaches to trace such transitions. Therefore, it is a very challenging task to build model reductions that can provide biological insights for γ-oscillations in a wide range of parameter regimes.

This paper explores learning the complex γ-oscillations with first-principle-informed DNNs. Our previous study revealed that the complex γ-oscillatory dynamics can be captured by a Poincare mapping F projecting the network state at one initiation of an MFE to the next initiation (Cai et al. 2021). Therefore, F is a high-dimensional mapping subjected to biophysical parameters of SNNs and is thus very hard to analyze. Some early theoretical work about sampling MFEs from integrate-and-fire neuronal networks with all-to-all connection are available in Zhang (2014), Zhang (2014), Zhang and Rangan (2015). However, to pursue analytical results, the scopes of studies were limited to very specialized situations. Instead of continue working on model-specific MFE sampling method, we approximate F by **A.** using coarse-graining (CG) and discrete cosine transform (DCT) to reduce the dimensionality of the state space, and **B.** benefiting from the representation power of DNNs. Specifically, DNNs provide a unified data-driven framework for varying SNN model parameters, revealing the potential of generalization to different dynamical regimes of the emergent network oscillations. Despite the significant underlying noise and the drastic dimensional reductions, our DNNs successfully capture the main feature of the γ-oscillations in SNNs. This effectively makes the DNN a surrogate of the SNNs. In principle, this work could be easily extended to partial synchrony and oscillatory dynamics in more biologically realistic networks using, e.g., Hodgkin-Huxley models or neuronal compartmental models (Hodgkin and Huxley 1952; Bressloff 1994).

The organization of this paper is as follows. Section 2 introduces the neuronal network model that serves as the ground truth. The descriptions and capturing algorithms of MFEs are depicted in Sect. 3. Section 4 discusses how to set up the training set for artificial networks. The main results are demonstrated in Sect. 5. Section 6 is the conclusion and discussion.

2 Neuronal network model description

Throughout this manuscript, we study SNN dynamics with a Markovian integrate-and-fire (MIF) neuronal network model. This model imitates a small, local circuit of the brain and shares many features of local circuits in living brains, including extensive recurrent interactions between neurons, leading to the emergence of γ-oscillations. We will evaluate the performance of DNNs based on their predictive power of dynamics produced by the MIF model.

2.1 An Markovian spiking neuronal network

We consider a simple spiking network consisting of N_E excitatory (E) neurons and N_I inhibitory (I) neurons, all homogeneously connected. The membrane potential (V) of each neuron is governed by Markovian dynamics, with the following assumptions:

1. V takes value in a finite, discrete state space;
2. For neuron i, V_i is driven by both external and recurrent E/I inputs from other neurons, through the effect of the arrival of spikes (or action potentials);
3. A spike is released from neuron i when V_i is driven to the firing threshold. Immediately after that, neuron i enters the refractory state before resetting to the rest state;
4. For a spike released by neuron i, a set of *post-synaptic* neurons is chosen randomly. Their membrane potentials are driven by this spike.

We now explain these assumptions in detail.

Single neuron dynamics. Let us index the N_E excitatory neurons from $1, \ldots, N_E$, and the N_I inhibitory neurons from $N_E + 1, \ldots, N_E + N_I$. For neuron i ($i = 1, \ldots, N_E + N_I$), the membrane potential V_i lies in a discrete state space Γ

$$V_i \in \Gamma := \{-M_r, -M_r + 1, \ldots, -1, 0, 1, \ldots, M\} \cup \{\mathcal{R}\},$$

where the states M_r, M, and $\mathcal{R}$ are the inhibitory reversal potential, the spiking threshold, and the refractory state, respectively. Once a neuron is driven to the threshold M, its membrane potential V_i enters the refractory state $\mathcal{R}$. After an exponentially distributed waiting time with mean $\tau_{\mathcal{R}}$, V_i is reset to the rest state 0. V_i is driven by the external (i.e., from outside the network itself) and recurrent inputs to neuron i. It is worth noting that, while a neuron is in the refractory state $\mathcal{R}$, V_i does not respond to any stimuli.

The external stimulus serves as an analog of feedforward sensory input, e.g., from the thalamus or from other brain regions. In this paper, the external inputs to individual neurons are modeled as series of impulsive kicks, whose arrival times are drawn from independent & identical Poisson processes. The rates of the Poisson processes, $\lambda^{E,I}$, are taken to be constants across the E/I populations. Each kick received by neuron i increases V_i by 1.

The recurrent inputs to a neuron are the spikes from other neurons. That is, an E/I neuron will *spike* when its membrane potential V_i reaches threshold M, sending an E/I

kick to its postsynaptic neurons (the choice of which will be discussed momentarily). Each recurrent E spike received by neuron i takes effect on V_i after an independent, exponentially distributed time delay $\tau \sim \mathrm{Exp}(\tau^E)$. The excitatory spikes received by neuron i that have not yet taken effect form *a pending E-spike pool*, with size H_i^E. Therefore, H_i^E increases by 1 when an E kick arrives at neuron i, and drops by 1 when a pending spike takes effect. This discussion applies to the I spikes as well: the size of the pending I-spike pool is H_i^I, and the waiting time of the pending spikes is subjected to $\tau \sim \mathrm{Exp}(\tau^I)$.

In summary, the state of neuron i is therefore described by a triplet

$$(V_i, H_i^E, H_i^I).$$

We note that the pool sizes H_i^E and H_i^I may be viewed as E and I synaptic currents of neuron i in the classical leaky integrate-and-fire neuron model (Gerstner 2014).

Impacts of spikes. The effects of the recurrent E/I spikes on membrane potentials are different. When a pending E-spike takes effect, V_i is increased by $[S^{Q,E}] + u^E$, where

$$u^E \sim \mathrm{Bernoulli}(p) \quad \text{and} \quad p = S^{Q,E} - [S^{Q,E}],$$

where $Q \in \{E, I\}$ and $[\ldots]$ denotes the floor integer function. Likewise, when a pending I-spike takes effect, V_i is decreased by $[S^{Q,I}] + u^I$, where

$$u^I \sim \mathrm{Bernoulli}(q) \quad \text{and} \quad q = S^{Q,I} - [S^{Q,I}],$$

V_i is strictly bounded to the state space Γ. Should V_i exceed M after an E-spike increment, it will reset to $\mathcal{R}$, and neuron i spikes immediately. On the other hand, should V_i go below $-M_r$ due to an I-spike, it will stay at $-M_r$ instead.

A homogeneous network architecture. Instead of having a predetermined network architecture with fixed synapses, the set of postsynaptic neurons of each spike are decided *on-the-fly*. That is to say, a new set of postsynaptic neurons is chosen independently for each spike. More specifically, when a type-Q' neuron spikes, the targeted postsynaptic neurons in the Q populations, excluding the spiking neuron itself, are chosen with probabilities $P^{QQ'}$ ($Q, Q' \in \{E, I\}$). We point out that the motivation of this simplification is for analytical and computational convenience by making neurons interchangeable within each subtype, and is standard in many previous theoretical studies (Cai 2006; Brunel and Hakim 1999; Wilson and Cowan 1972; Buice and Cowan 2007; Cai et al. 2021; Gerstner 2014).

To summarize, the state space of the network is denoted as $\boldsymbol{\Omega}$. A *network state* $\omega \in \boldsymbol{\Omega}$ consists of $3(N_E + N_I)$ components

$$\begin{aligned}
\omega = (&V_1, \ldots, V_{N_E}, V_{N_E+1}, \ldots, V_{N_E+N_I}, \\
&H_1^E, \ldots, H_{N_E}^E, H_{N_E+1}^E, \ldots, H_{N_E+N_I}^E, \\
&H_1^I, \ldots, H_{N_E}^I, H_{N_E+1}^I, \ldots, H_{N_E+N_I}^I).
\end{aligned} \tag{1}$$

Table 1 Parameters regarding the network architecture (first row) and individual neuronal physiology (second row)

Parameter group	Parameter	Meaning	Value
Network architecture	N^E	Number of E cells	300
	N^I	Number of I cells	100
	P^{EE}	E-to-E coupling probability	0.15
	P^{EI}	I-to-E coupling probability	0.50
	P^{IE}	E-to-I coupling probability	0.50
	P^{II}	I-to-I coupling probability	0.40
	S^{EE}	E-to-E synaptic weight	[3.5, 4.5]
	S^{EI}	I-to-E synaptic weight	[−2.5, −1.5]
	S^{IE}	E-to-I synaptic weight	[2.5, 3.5]
	S^{II}	I-to-I synaptic weight	[−2.5, −1.5]
Neuronal physiology	M	Threshold potential	100
	$-M_r$	Inhibitory reversal potential	−66
	$\tau^{\mathcal{R}}$	Expectation of refractory period	3 ms
	τ^E	Expectation of E-spike pending time	2 ms
	τ^I	Expectation of I-spike pending time	4 ms
	λ^E	Total external spikes/s to E	3 kHz
	λ^I	Total external spikes/s to I	3 kHz

Symbols, meanings, and values of relevant parameters are depicted

2.2 Parameters used for simulations

The choices of parameters are adopted from our previous studies (Li and Hui 2019; Cai et al. 2021; Wu et al. 2022). For all SNN parameters used in the simulations, we list their definitions and values in Table 1. Here, we remark that the projection probability between neurons $P^{Q'Q}$ ($Q, Q' \in \{E, I\}$) are chosen to match the anatomical data in the macaque visual cortex, see (Chariker et al. 2016) for reference. Also, $\tau^E < \tau^I$, since it is known that the Glu-AMPA receptors act faster than the GABA-GABA receptors, with both on a time scale of milliseconds (Koch 1999).

On the other hand, four parameters concerning recurrent synaptic coupling strength will be tested and varied in this study. This is because they are sensitive to SNN dynamics, and yet hard to directly measure by current experiments methods (Chariker et al. 2016; Xiao et al. 2021). The range of tested parameter set $\Theta = \{S^{EE}, S^{EI}, S^{IE}, S^{II}\} \subset \mathbb{R}^4$ is also given by Table 1.

3 Multiple-firing events

In the studies of γ-oscillations. In many brain regions, electrophysiological recordings of the local field potentials reveal temporal oscillations with power peaking in the

γ-frequency band (30–90 Hz) (Andrew Henrie and Shapley 2005; Brosch et al. 2002; Bauer 2006; Buschman and Miller 2007; Pieter Medendorp 2007; van Wingerden 2010; Csicsvari 2003; Popescu et al. 2009; Van Der Meer and David Redish 2009). Because of the belief that these coherent rhythms play a crucial role in cognitive computations, there has been much work on understanding their mechanisms in different brain areas, in disparate dynamical regimes, and within various brain states (Azouz and Gray 2000, 2003; Frien 2000; Womelsdorf 2012; Liu and Newsome 2006; Fries 2001, 2008; Bauer et al. 2007; Pesaran 2002; Womelsdorf 2007; Başar 2013; Krystal 2017; Mably and Colgin 2018).

To explain the neural basis of γ-oscillations, a series of theoretical models have found transient, nearly synchronous collective spiking patterns that emerge from the tight competitions between E and I populations (Whittington 2000; Rangan and Young 2013; Traub 2005; Chariker and Young 2015; Chariker et al. 2018). More specifically, rapid & coherent firing of neurons occurs within a short interval, and such spiking patterns recur with γ-band frequencies in a stochastic fashion. This phenomenon is termed multiple-firing events (MFEs). MFEs are triggered by a chain reaction initiated by recurrent excitation, and terminated by the accumulation of inhibitory synaptic currents. In this scenario, the γ-oscillations appear in electrophysiological recordings as the local change of the electric field generated by MFEs. We refer readers to Rangan and Young 2013; Chariker and Young 2015; Chariker et al. 2018; Li and Hui 2019; Cai et al. 2021; Wu et al. 2022 for further discussions of MFEs.

The alternation of fast and slow phases is one of the most significant dynamical features of MFEs (Fig. 1). Namely, at the beginning of an MFE, a chain reaction leads to a transient state where many neurons fire collectively in a short time interval (the fast phase). On the other hand, during the interval between two successive MFEs (an inter-MFE interval, or IMI), the neuronal network exhibits low firing rates while the system recovers from the extensive firings via a relatively quiescent period (the slow phase). These two phases can be discriminated by the temporal coherence of spikes in the raster plot where the time and location of spikes are indicated (Fig. 1A, blue dots). Here, MFEs and IMIs are separated by vertical lines - we will discuss the method of capturing MFEs momentarily in Sect. 3.2.

The complexity of MFEs is partially reflected by their sensitivity to spike-timing fluctuations (Xiao and Lin 2022). Specifically, during an MFE, the missing or misplacement of even a small number of spikes can significantly alter the path of network dynamics. Furthermore, the high dimensionality of state space $\mathbf{\Omega}$ and the high degree of nonlinearity make MFE hard to analyze. On the other hand, the network dynamics during IMIs are more robust to perturbations and exhibit low-dimensional features (Cai et al. 2021; Wu et al. 2022).

3.1 Spike-timing sensitiveness of transient dynamics

We illustrate this slow-fast dichotomy through a comparison between two different numerical simulations, one using the stochastic simulation algorithm (SSA) and another one using tau-leaping (Fig. 1).

A   Raster plot

B   HEE trajectory
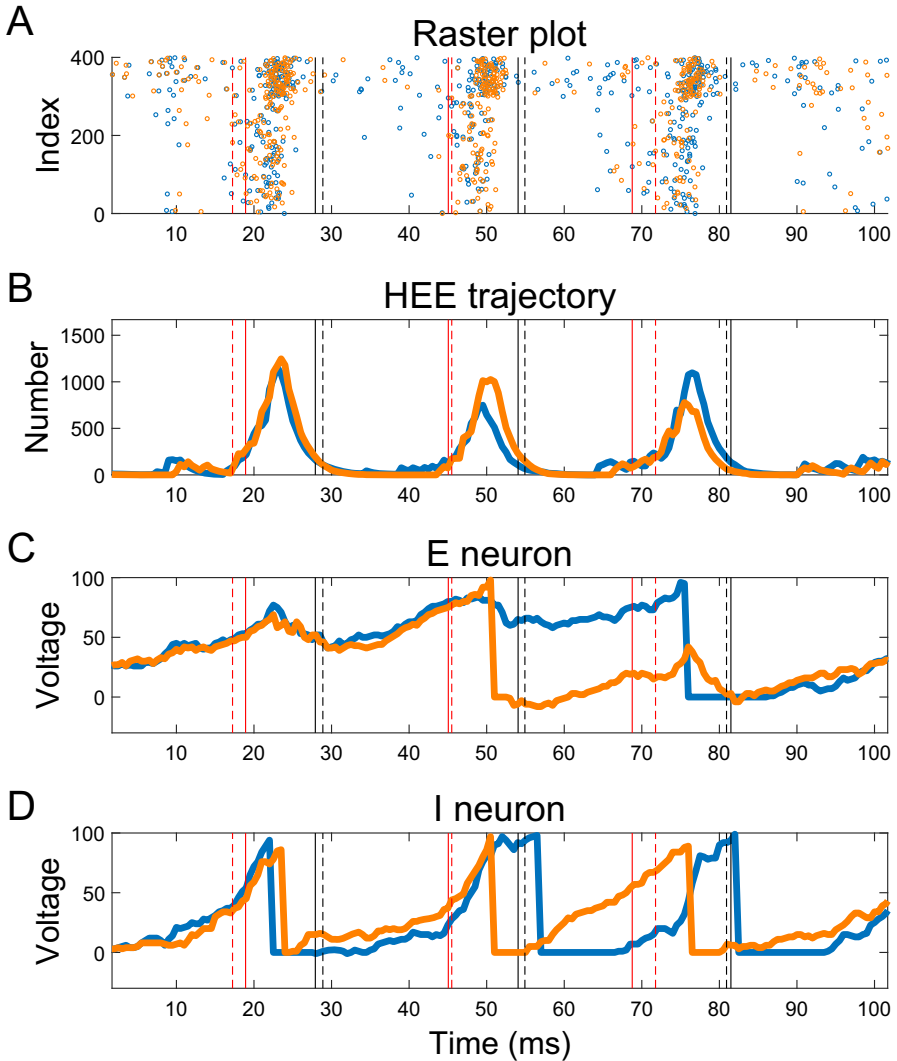
C   E neuron

D   I neuron

Fig. 1 Divergence between SSA and tau-leaping methods. **A** Raster plot from SSA (blue) and tau-leaping method (orange) using same seed. **B** HEE trajectories from SSA and tau-leaping method. **C-D** Voltage trajectories of an E/I neuron from SSA and tau-leaping method. Vertical red/black lines indicate the beginnings/endings of MFEs. (solid: SSA, dashed: tau-leaping) (color figure online)

Stochastic simulation algorithms vs. tau-leaping. In SSA, the timings of state transitions in phase space are exact, since the next state transition is generated by the minimum of finitely many independent, exponentially distributed random variables (see Appendix). On the other hand, algorithms simulating random processes with fixed time steps, such as tau-leaping, introduce errors to event times at every step. In a tau-leaping simulation with a fixed step size of Δt, the effect of spikes (interactions among neurons) is processed after each time step. Hence, within each time step, all events are

uncorrelated, and changes are held off until the next update. Therefore, the precision of SSA is determined by the computational precision of the C++ code, which is much higher than tau-leaping.

Comparison. Intuitively, tau-leaping methods can well capture the slow network dynamics during IMIs, but not the fast chain reactions during MFEs. The detailed comparison is depicted in Fig. 1, where we choose $\Delta t = 0.5$ ms for the tau-leaping method. To constrain differences induced by stochastic fluctuations, we couple the two simulations with the same external noise, leaving the intrinsic noise (the random waiting times of pending spikes and refractory states) generated within the algorithms themselves. The raster plots depict spiking events produced by both methods and diverge rapidly during MFEs (Fig. 1A. blue: SSA; orange: tau-leaping), since the crucial coherence between firing events is not accurately captured by tau-leaping. This point is also supported by the comparison between voltage traces of single-neurons (Fig. 1C, D): Although well aligned at the beginning, the spike timings of the neurons are strongly affected by the transient fluctuation during MFEs.

In addition to the firing events, we also illustrate the comparison between pending spikes. Here we define

$$H^{Q'Q} = \sum_{i \in Q} H_i^{Q'}, \quad Q, \, Q' \in \{E, I\},$$

i.e., the total number of pending Q'-spikes for type-Q neurons. Figure 1B demonstrates the divergence of H^{EE} collected from different simulations. Because of the identical external random noise, trajectories from the two methods are similar in the first hundred milliseconds. However, very rapidly, the accumulation of errors in the first fast phase causes a large divergence. Other pending spikes statistics yield similar disagreement between the two methods (data not shown).

Therefore, in the rest of this paper, we use SSA to simulate the SNN dynamics, which is treated as the "ground truth".

3.2 Capturing MFEs from network dynamics

The first step of investigating MFE is the accurate detection of MFEs patterns from the temporal dynamics of SNN. Due to the lack of a rigorous definition in previous studies, we develop an algorithm to capture MFEs based on the indication of recurrent excitation (Algorithm 1), splitting the spiking patterns into consecutive phases of MFEs and IMIs.

The algorithm capturing MFEs. Our basic assumption is that the existence of the cascade of recurrent excitation has a causal relation with MFEs. Therefore, given a temporal spiking pattern generated by the SNN, Algorithm 1 detects the initiation of a candidate MFE with the following criteria:

- Two E-to-E spikes take place consecutively, where the second one is triggered due to the influence of the first one;
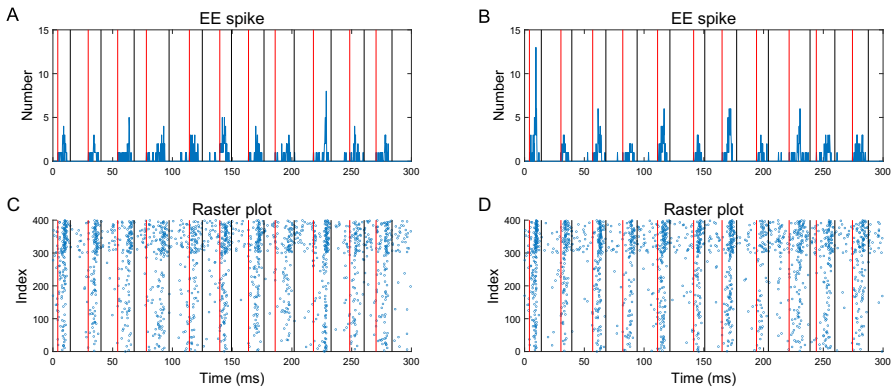
Fig. 2 The amount of recurrent excitation serves as the indicator of MFEs. **A-B** Number of E spikes caused by recurrent excitation within each 0.25 ms interval. $(S^{EE}, S^{IE}, S^{EI}, S^{II}) = (4, 3, -2.2, -2)$ for **A**; $(4.44, 2.88, -2.22, -1.70)$ for **B C-D** Raster plot of the two simulations. Vertical red and black lines indicate the beginnings and endings of MFEs respectively

- The second E-to-E spike should occur within a 4 ms interval following the first one (twice the excitatory synaptic timescale $\tau_E = 2$ms).

After the cascade of spiking events, the candidate MFE is deemed terminated if no additional E spike takes place within a 4 ms time-window. Furthermore, to exclude isolated firing events clustering by chance, we apply merging and filtering processes to each of the MFE candidates. More specifically, consecutive MFE candidates are merged into one if they occur within 2 ms. Candidates are eliminated if their duration is less than 5 ms *or* the number of spikes involved is less than 5. We comment that the filtering threshold is chosen based on the size of 400-neuron networks. A different filtering standard is employed for larger networks, presented in Sect. 4.3.

Algorithm 3.2, based on the timing of recurrent E-spikes, is employed throughout this manuscript to detect MFEs. Two examples of different choices of parameters are illustrated in Fig. 2, where the initiation and termination of MFEs are indicated by red and black vertical lines. In the rest of the paper, we denote the time sections of initiations and terminations of the m-th MFE as t_m and s_m. Therefore, the m-th MFE takes place during the interval $[t_m, s_m]$, whereas the IMI after that is $[s_m, t_{m+1}]$.

4 Learning MFEs with artificial neural networks

Viewed within the framework of random dynamical systems, SNN dynamics exhibiting γ-oscillation can be effectively represented by a Poincare map theory (Wu et al. 2022). Consider a solution map of the SNN dynamics depicted in Sect. 2,

$$\omega(t) = \Phi_t^\theta(\omega_0, \xi_0^t),$$

where $\omega_0 \in \Omega$ is the initial condition of the SNN, $\theta \in \Theta$ indicates SNN parameters and ξ_0^t is the realization of all external/internal noises during the simulation from time

---

Algorithm 1 Capturing MFE

---

   procedure RECORDING, MERGING AND FILTERING MFE
     repeat
       A new event in simulation happens
       if State == Recording MFE **then**
5:        **if** # of EE spikes within previous 4 ms < 2 **then**
           State ← Not in MFE
           Record post-MFE network state ω
        end if
       else
10:       **if** Two EE spikes occur within 4 ms **then**
           Record at first EE spikes as pre-MFE network state ω
        end if
       end if
     until current time > terminate time
15:    Merge any consecutive MFE candidates with space < 2 ms
     for $i = 1$ to number of merged candidates **do**
       if (Length of candidate i)>5 ms & (Number of spike)>threshold & (peak H^{EE})>(100 + start H^{EE}) **then**
          Register candidate i as MFE, output all recorded network state
       end if
20:   **end for**
   end procedure

---

0 to t. (The solution map satisfies $\Phi_{t+\tau} = \Phi_t \circ \Phi_\tau$.) To study the recurrence of MFEs, we focus on the time sections $\{t_m\}$ on which $\omega(t_m)$ returns to the initiation of the mth MFE. Therefore, we define a sudo-Poincare map:

$$\omega(t_{m+1}) = F^\theta(\omega(t_m)) = \Phi^\theta_{t_{m+1}-t_m}(\omega(t_m), \xi^{t_{m+1}}_{t_m}). \tag{2}$$

We note that F^θ is not a Poincare map in the rigorous sense, since the initiation of MFEs depends on the temporal dynamics in a short interval (Algorithm 3.2).

    This paper aims at investigating F^θ. It is generally impractical to study F^θ in an analytical manner due to the high dimensionality of $\boldsymbol{\Omega}$. To overcome the difficulty, we propose a *first-principle-informed* deep learning framework.

4.1 Dissecting F^θ into slow/fast phases

We first dissect F^θ guided by SNN dynamics. In the regime of γ-oscillations, the SNN dynamics is composed of the regular alternation of slow/fast phases, though the duration and detailed dynamics of each phase may vary. Therefore, the pseudo-Poincare map F^θ is equivalent to the composition of two maps:

$$F^\theta = F^\theta_1 \circ F^\theta_2, \quad F^\theta_{1,2} : \boldsymbol{\Omega} \to \boldsymbol{\Omega}. \tag{3}$$

More specifically, F^θ_1 maps the network state at the initiation of an MFE to its termination, while F^θ_2 maps the network state from the beginning of an IMI to the initiation

of the next MFE, i.e.,

$$F_1^\theta(\omega(t_m)) = \omega(s_m), \quad F_2^\theta(\omega(s_m)) = \omega(t_{m+1}).$$

We denote $F_{1,2}^\theta$ as MFE and IMI mappings, respectively. To summarize, the dynamical flow of Φ^t is equivalent to:

$$\dots \omega(t_m) \xrightarrow[\text{MFE}]{F_1^\theta} \omega(s_m) \xrightarrow[\text{IMI}]{F_2^\theta} \omega(t_{m+1}) \xrightarrow[\text{MFE}]{F_1^\theta} \omega_{s_{m+1}} \dots$$

Our previous studies demonstrated that the slow and relatively quiescent dynamics during IMIs can be well captured by classical coarse-graining methods, i.e., the IMI mapping F_2^θ may be represented by the evolution of certain Fokker-Planck equations (Cai 2006). However, this is not the case for the MFE mapping F_1^θ due to the highly transient and nonlinear dynamics. Instead, we turn to deep neural networks (DNN) due to their success in tackling high-dimensional and nonlinear problems. In the rest of this section, our goal is to train a DNN with relevant data and generate surrogates of F_1^θ.

4.2 First-principle-based reductions of the problem

Instead of requiring the DNN to learn the full map F_1^θ and directly link the network state from $\omega(t_m)$ to $\omega(s_m)$, we prepare a training set $\mathcal{T}_{\text{train}}^\theta$ with first-principle-based dimensional reduction, noise elimination, and enlargement for robustness to facilitate the training process.

The DNN used in this paper has a feedforward structure with 4 layers, consisting of 512, 512, 512, and 128 neurons, respectively. We leave the detailed information of the DNN architecture to **Appendix**.

4.2.1 Coarse-graining SNN states

Approximating the features of MFE mapping F_1^θ with a DNN (or any statistical machine learning method) immediately faces the curse of dimensionality. Namely, F_1^θ maps a $3N$-dimensional space $\boldsymbol{\Omega}$ to itself. To cope with N, we propose a coarse-graining model reduction with *a priori* physiological information.

Instead of enumerating the actual state of every neuron, we assume the E and I neuronal populations form two *ensembles*. That is, the state of a type-Q neuron can be viewed as randomly drawn from a distribution $\boldsymbol{\rho}^Q(v, H^E, H^I)$ of the Q-ensemble, where $Q \in \{E, I\}$. Furthermore, note that there is no fixed architecture of synaptic connection in SNN (see Sect. 2)—any neurons in the same population has the same probability to be projected when a spike occurs. Therefore, it is reasonable to decorrelate v and the (H^E, H^I), i.e.,

$$\boldsymbol{\rho}^Q(v, H^E, H^I) \sim \boldsymbol{p}^Q(v) \cdot \boldsymbol{q}^Q(H^E, H^I), \tag{4}$$

where $\boldsymbol{p}^Q$ and $\boldsymbol{q}^Q$ are marginal distributions of $\boldsymbol{\rho}^Q$.

More specifically, $\boldsymbol{p}^E(v)$, $\boldsymbol{p}^I(v)$ yield the distributions of neuronal voltages for both populations on a partition of the voltage space Γ:

$$\Gamma = \Gamma_1 \cup \Gamma_2 \cup ... \cup \Gamma_{22} \cup \Gamma_{\mathcal{R}}$$
$$= [-M_r, 0) \cup [0, 5) \cup ... \cup [M - 5, M = 100) \cup \{\mathcal{R}\}. \qquad (5)$$

On the other hand, the distribution of pending spike $\boldsymbol{q}^Q(H^E, H^I)$ is effectively represented by the total number of pending spikes summed over each population, H^{QE} and H^{QI}. To summarize, we define a coarse-grained mapping $\mathcal{C} : \boldsymbol{\Omega} \mapsto \tilde{\boldsymbol{\Omega}}$ by projecting the network state ω onto a 50-dimensional network state $\tilde{\omega}$:

$$\mathcal{C}(\omega) = \tilde{\omega}$$
$$= \left(\boldsymbol{p}^E(v), \boldsymbol{p}^I(v), H^{EE}, H^{EI}, H^{IE}, H^{II} \right)$$
$$= \left(n_1^E, n_2^E, \cdots, n_{22}^E, n_R^E, n_1^I, n_2^I, \cdots, n_{22}^I, n_R^I, \right.$$
$$\left. H^{EE}, H^{EI}, H^{IE}, H^{II} \right). \qquad (6)$$

Here, n_k^Q denotes the number of type-Q neurons whose potentials lie in Γ_k, and $\tilde{\boldsymbol{\Omega}} \subset \mathbb{R}^{50}$ represents the coarse-grained state space. The precise definition of n_k^Q is given in the Appendix.

4.2.2 Training DNN

We now train a DNN to learn the coarse-grained mapping $\widetilde{F}_1^\theta$ with the coarse-grained network states, $\tilde{\omega}_{s_m} = \widetilde{F}_1^\theta(\tilde{\omega}_{t_m})$, where $\tilde{\omega}_t = \mathcal{C}(\omega(t))$. That is, for a set of fixed SNN parameters (θ), the DNN forms a mapping

$$\widehat{F}_1^{\theta,\vartheta} : \tilde{\boldsymbol{\Omega}} \mapsto \tilde{\boldsymbol{\Omega}}$$

where ϑ is the hyperparameter for the trained DNN. From a simulation of SNN dynamics producing K MFEs, each piece of data of the training set is composed by

– Input: $x_m = \tilde{\omega}_{t_m}$, the coarse-grained network states at the initiation of the m-th MFE, and
– Output: $y_m = (\tilde{\omega}_{s_m}, \mathrm{Sp}_E, \mathrm{Sp}_I)$, the coarse-grained state at the m-th MFE termination and the numbers of E/I-spikes during MFEs ($m \in \{1, 2, ..., K\}$).

The training process aims for the optimal ϑ to minimize the following L^2 loss function

$$\mathcal{L}^\theta(\vartheta) = \frac{1}{K} \sum_{m=1}^K \|y_m - \widehat{F}_1^{\theta,\vartheta}(x_m)\|_{L^2}^2, \qquad (7)$$

and we hope to obtain the optimal coarse-grained MFE mapping $\widehat{F}_1^\theta$ as an effective surrogate of $\widetilde{F}_1^\theta$.
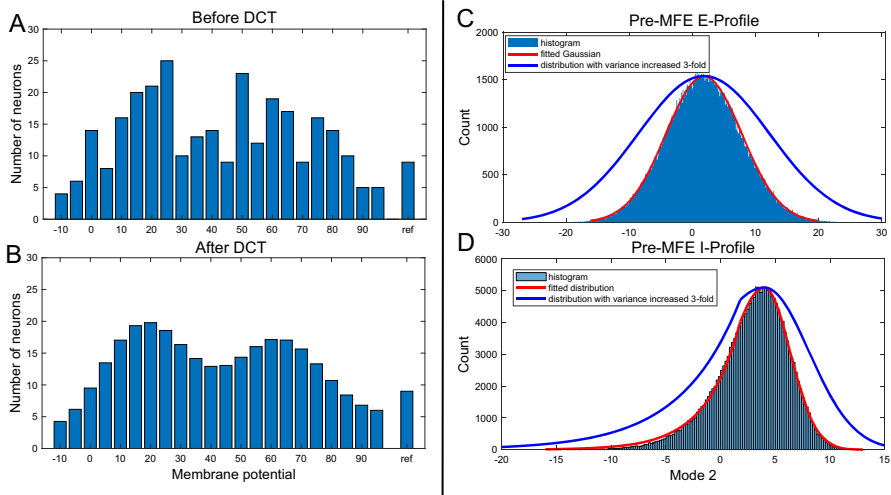
Fig. 3 Pre-processing and enlarging the training data-set. **A** An example of pre-MFE E-neuron voltage distribution $p^E(v)$ before DCT + iDCT. For clarity, the number of refractory neurons is depicted by the rightmost bin; **B** The pre-processed $\widehat{p^E}(v)$ after DCT + iDCT; **C** The distribution of the 2nd DCT mode of $\widehat{p^E}(v)$; **D** The distribution of the second DCT mode of $\widehat{p^I}(v)$

4.2.3 Pre-processing: Eliminating high-frequency noises

Using a discrete cosine transform (DCT), we pre-process the training data (x, y) by eliminating the noisy dimensions in potential voltage distributions p^Q, i.e., the high-frequency modes.

The raw distributions of membrane potentials $p^Q(v)$ contain significant high-frequency components (Fig. 3A). This is partially due to the stochasticity and the small number of neurons in the MIF model. Unfortunately, the high-frequency components could lead to "conservative" predictions, i.e., the training of DNN converges to the *averages* of post-MFE voltage distributions to minimize $\mathcal{L}^\theta(\vartheta)$. Therefore, different inputs would yield similar outputs.

To resolve this issue, we apply a DCT method to remove the high-frequency components from training data (x, y), only preserving the first eight modes. This is sufficient to discriminate a pair of neurons whose membrane potentials $|v_1 - v_2| > 5$, where $v_1, v_2 \in \Gamma$. For $|v_1 - v_2| < 5$, the two neurons are either placed in the same or consecutive two bins after coarse-graining. Therefore, the E spikes needed to involve them in the upcoming MFE differ by at most 1.

More specifically, in the training set, 1.86×10^5 pairs of network state (x, y) are collected from a 5000-second simulation of the SNN network (see Eq. 6). For the voltage distribution components of each state, $p^E(v)$ and $p^I(v)$, we remove the high-frequency components by

$$\widehat{p^Q} = \mathcal{F}_c^{-1} \circ T \circ \mathcal{F}_c\{p^Q\}, \quad Q \in \{E, I\}. \tag{8}$$

Here, $\mathcal{F}_c$ indicates the DCT operator and T is a truncation matrix preserving the first eight components. Finally, in the training data (x, y), the coarse-grained network state $\tilde{\omega}$ is replaced as

$$\tilde{\omega} = \left(\widehat{\boldsymbol{p}^E}(v), \widehat{\boldsymbol{p}^I}(v), H^{EE}, H^{EI}, H^{IE}, H^{II} \right).$$

Fig. 3A, B depicts an example of pre-processing a voltage distribution. We leave the rest of the details regarding the DCT methods to the Appendix.

4.2.4 Robustness of training

The robustness issue of DNN has been addressed in many previous studies (Goodfellow et al. 2014; Yuan 2019). One approach to improving robustness is via data augmentation, in which the training set is enlarged by adding modified copies of existing data (Shorten and Khoshgoftaar 2019). Motivated by this idea, we propose a training set $\mathcal{T}_{\text{train}}^{\theta}$ to account for more irregular inputs than SNN-produced network states, helping the trained DNN generate realistic and robust predictions for MFE dynamics. Based on the pre-processed network states $\tilde{\omega}$, we increase the variability of the first eight frequency modes of $\widehat{\boldsymbol{p}^Q}(v)$ as they are the most salient information revealed by DCT.

$\mathcal{T}_{\text{train}}^{\theta}$ consists of pairs of pre/post-MFE network states (x_m, y_m). A candidate initial state close to a pre-MFE state x_m in $\mathcal{T}_{\text{train}}^{\theta}$ is generated as follows:

1. The voltage distributions $\widehat{\boldsymbol{p}^Q}(v)$. We first collect the empirical distributions of all frequency modes (by concerning each entry of $\mathcal{F}_c\{\boldsymbol{p}^Q\}$) from 1.86×10^5 pre-processed pre-MFE network states. The empirical distributions are fitted by direct combinations of Gaussian and exponential distributions (see, e.g., Fig. 3C, D). We then increase the variances of fitted distributions by a factor of three, from which the 2nd-8th frequency modes are sampled. On the other hand, the first mode comes from the distributions with the original variances, since it indicates the number of neurons outside the refractory state. Finally, the higher order of DCT frequency modes are treated as noises and truncated.
2. The pending spikes $H^{Q'Q}$. Likewise, the "fit-expand-sample" operations similar to how we treated the 2nd-8th frequency modes in $\widehat{\boldsymbol{p}^Q}(v)$ are applied to sample the number of pending spikes.

It is also important to make sure all training data are authentic abstractions of SNN states in MFE dynamics, i.e., an MFE can be triggered in a short-time simulation (5ms) from a pre-MFE state. Therefore, we perform a simulation that begins with each initial state generated by the enlargement above, and collect the corresponding pre-MFE states (x) and post-MFE states (y) if an MFE emerges. The enlargement of distributions of MFE magnitudes is shown in Fig. 18

In summary, the enlarged $\mathcal{T}_{\text{train}}^{\theta}$ consists of 3×10^5 pairs of (x_m, y_m):

$$\mathcal{T}_{\text{train}}^{\theta} = \left\{ (x_m, y_m) : m = 1, 2, ..., 3 \times 10^5 \right\},$$

where 50% of the data comes directly from network simulation (after pre-processing of DCT), and the rest comes from the data augmentation process.

4.3 Generalization for different SNN parameters

Varying the synaptic coupling strengths. Trained with data from a dynamical system parameterized by θ, DNN produces a surrogate mapping $\widehat{F}_1^\theta$. Recall that θ is a parameter point in a 4D cube of recurrent synaptic coupling strength

$$\Theta = \Big\{ (S^{EE}, S^{IE}, S^{EI}, S^{II}) \in$$
$$[3.5, 4.5] \times [2.5, 3.5] \times [-2.5, -1.5] \times [-2.5, -1.5] \Big\}.$$

We treat $\widehat{F}_1^\theta$ as a milestone, and further propose a parameter-generic MFE mapping

$$\widehat{F}_1 : \tilde{\Omega} \times \Theta \mapsto \tilde{\Omega}.$$

That is, given any point $\theta \in \Theta$ and a pre-MFE state, the DNN predicts the post-MFE state. The optimization problem and loss function are analogous to Eq. 7.

The training set $\mathcal{T}_{\text{train}}$ for parameter-generic MFE mapping consists of SNN states from 20000 different of parameters points, which are randomly drawn from Θ. To ensure reasonable SNN dynamics, each parameter point θ is tested by the following criteria

$$(f_E, f_I) = L(\theta)$$
$$f_E \leq 50 \text{ Hz and } f_I \leq 100 \text{ Hz},$$

where $L(\theta)$ is a simple linear formula estimating firing rates of E/I populations based on recurrent synaptic weights (for details see Appendix and Li et al. 2019). For each accepted point in parameter space, we perform a 500 ms simulation of SNN and collect 20 pairs of pre-processed, coarse-grained pre- and post-MFEs states (see Sect. 4.2). The chosen parameter space is wide to generate various network dynamics and various MFEs. The firing rates of networks with parameters that can produce accepted MFEs are shown in Fig. 19, and the various sizes of produced MFEs are shown in Fig. 17.

Varying the network size. Our method is generic to γ-oscillations produced by SNNs of different sizes. We demonstrate this on a 4000-neuron SNN (3000 E and 1000 I neurons) sharing all parameters with the previous 400-neuron SNN, except that the synaptic weights ($S^{QQ'}$) are 1/10 of the values listed in Table 1. This change of synaptic weights aims to control the total recurrent E/I synaptic drives received by each neuron and allows the different network models to have the same mean-field limit. While deferring DNN predictions of SNN dynamics to Sect. 5, we here note the minor modifications to adopt our methods to the 4000-neuron SNN.

First, the filtering threshold of MFE is increased from 5 to 50 spikes when collecting MFE-related data (see Sect. 3.2), since the MFEs are larger. Second, because of the central limit theorem, less intrinsic stochasticity are observed in the SNN dynamics.

This leads to an immediate side effect: The span of pre/post-MFE network states collected from SNN simulations is relatively narrower within $\boldsymbol{\Omega}$, i.e., the training data is less "general". To compensate, we expand the spans of $\mathcal{T}^{\theta}_{\text{train}}$ more courageously. That is, during the enlarging step of the training set, the components of pre-MFE states are sampled from distributions with more significant variance expansions.

4.4 Training result

The trained DNNs provide faithful surrogate MFE mappings for both $\widehat{F}^{\theta}_1$ and $\widehat{F}_1$.
Parameter-specific MFE mappings. We first illustrate $\widehat{F}^{\theta}_1$ at a particular parameter point

$$\theta = (S^{EE}, S^{EI}, S^{IE}, S^{II}) = (4, 3, -2.2, -2).$$

We test the predictive power of $\widehat{F}^{\theta}_1$ on a testing set

$$\mathcal{T}^{\theta}_{\text{test}} = \left\{ (x'_m, y'_m) : m = 1, 2, ..., M = 6 \times 10^4 \right\},$$

where (x'_m, y'_m) are coarse-grained pre/post-MFE states without pre-processing collected from SNN simulations. In Fig. 4A, one example of DNN prediction to post-MFE $p^E(v)$ is compared to the SNN simulation results starting from the same x'_m. Also, the comparison between predicted vs. simulated E/I-spike numbers during MFEs is depicted in Fig. 4B. To demonstrate the accuracy of $\widehat{F}^{\theta}_1$, Fig. 4C depicts the L^2 losses of different components of post-MFE states. The L^2 loss of predicted $p^Q(v)$ is ~4, while the averaged L^2 difference between the post-MFE voltage distributions in the testing set is

$$\underset{1 \le m, \ell \le M}{\text{mean}} \| p^E_m(v) - p^E_\ell(v) \|^2 + \| p^I_m(v) - p^I_\ell(v) \|^2 \approx 20,$$

Notably, DCT pre-processing effectively improves DNN predictions of voltage distributions by reducing the L^2 loss from ~10 to ~4. A similar comparison is observed for the prediction of pending spike numbers in the post-MFE states (Fig. 4C inset).

Parameter-generic MFE mappings. Likewise, $\widehat{F}_1$ also provides faithful predictions following SNN dynamics in different parameter regimes by labeling training data with synaptic strength parameters. Figure 4D shows a similar comparison between L^2 losses of different components of post-MFE states.

5 Producing a surrogate for SNN

Here, we depict how our ANN predictions provide a surrogate of the spiking network dynamics. We focus on the algorithm to replace F^{θ} for a fixed θ. The algorithm for parameter-generic F is analogous.
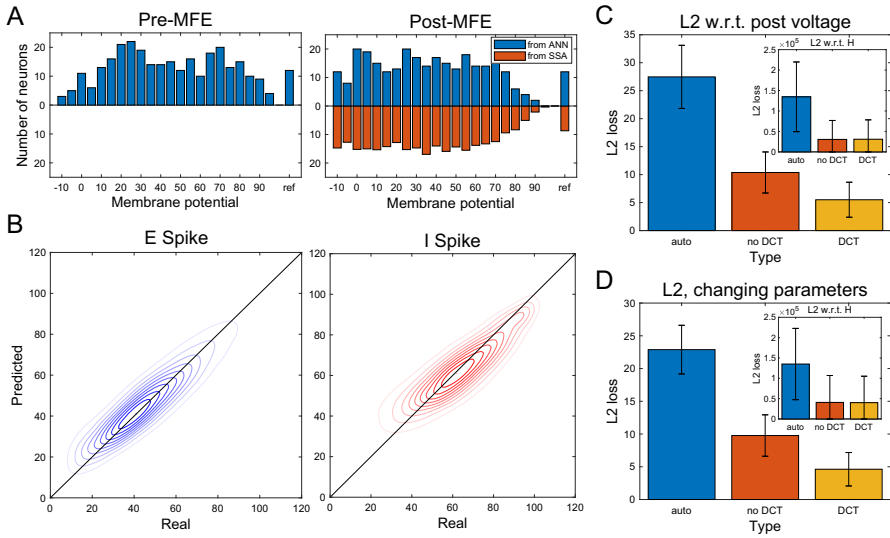
Fig. 4 DNNs predictions of post-MFE states. **A** Mapping $\widehat{F}_1^\theta$ for $\theta = (S^{EE}, S^{EI}, S^{IE}, S^{II}) = (4, 3, -2.2, -2)$. **A** Left: a pre-MFE $\boldsymbol{p}^E(v)$; Right: post-MFE $\boldsymbol{p}^E(v)$ produced by ANN (blue) vs. spiking network simulations (orange). **B** Comparison of E and I spike number during MFEs, ANN predictions vs. SSA simulations. The distributions are depicted by 10th contours of max in ks-density estimation; **C** L^2 losses of predicted post-MFE $\boldsymbol{p}^Q(v)$ and pending spike number Hs. Blue: auto-L^2 differences between training data; orange: averaged L^2-loss without pre-processing; yellow: averaged L^2-loss with pre-processing. **D** L^2 loss for the parameter-generic mapping $\widehat{F}_1$. The meaning of all bins are similar to **B** (color figure online)

Recall that SNN dynamics is divided into a fast phase (MFEs) and a slow phase (IMIs). Our first-principle–informed DNN framework produces a surrogate mapping $\widehat{F}_1^\theta$ and replaces the pseudo-Poincare mapping F^θ if complemented by F_2^θ. We approximate the IMI mapping F_2^θ by evolving SNN with a *tau-leaping* method, thereby producing the surrogate to the full SNN dynamics by alternating between the two phases.

To resemble the IMI dynamics, we first initialize the network state ω from the coarse-grained post-MFE state $\tilde{\omega}$ predicted by F_1^θ. Since neurons in the MIF model are exchangeable, ω can be randomly sampled from $\mathcal{C}^{-1}(\tilde{\omega})$. Specifically, we evenly assign voltage to each type-Q neuron based on $\widehat{p^Q}$. On the other hand, same-category pending spikes stay "pooled" and are assigned to each neuron interchangeably. After that, the network state ω is evolved by a tau-leaping method with 1-ms timesteps. This process is terminated if more than three E-to-E spikes or six E spikes occur within 1 ms, after which the next MFE is deemed to start and the network state ω is fed to F_1^θ for another round of prediction.[1] The loop F^θ alternating between F_1^θ and F_2^θ is thus closed. The mathematical descriptions of the SNN surrogate algorithm are summarized as follows:

(i) $(\tilde{\omega}_{s_m}, \mathrm{Sp}_E, \mathrm{Sp}_I) = \widehat{F}_1^\theta(x_m)$, where $\tilde{\omega}_{s_m}$ is the coarse-grained post-MFE state.

---

[1] The different criteria of MFE initiation from Algorithm 1 aims to ensure the robustness of capturing MFE, due to the lack of network state information within each timestep in the tau-leaping simulations.
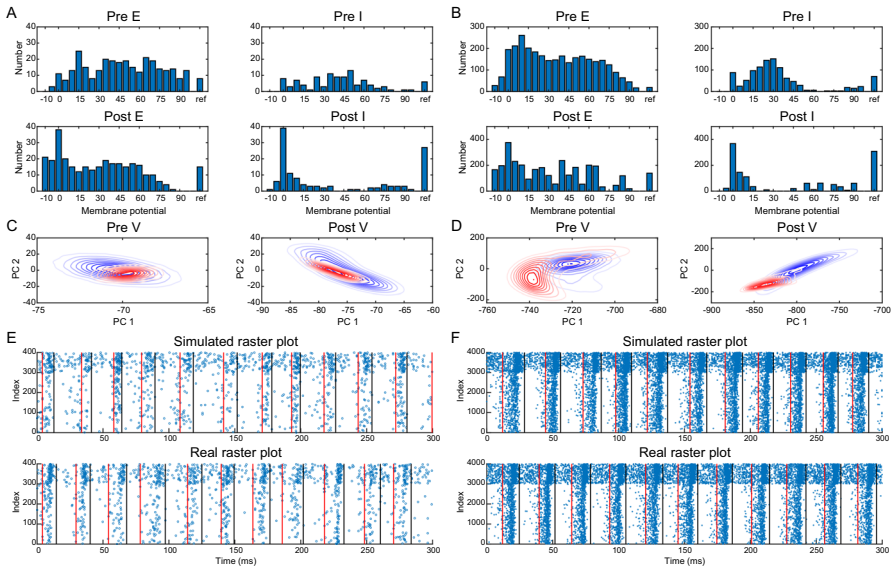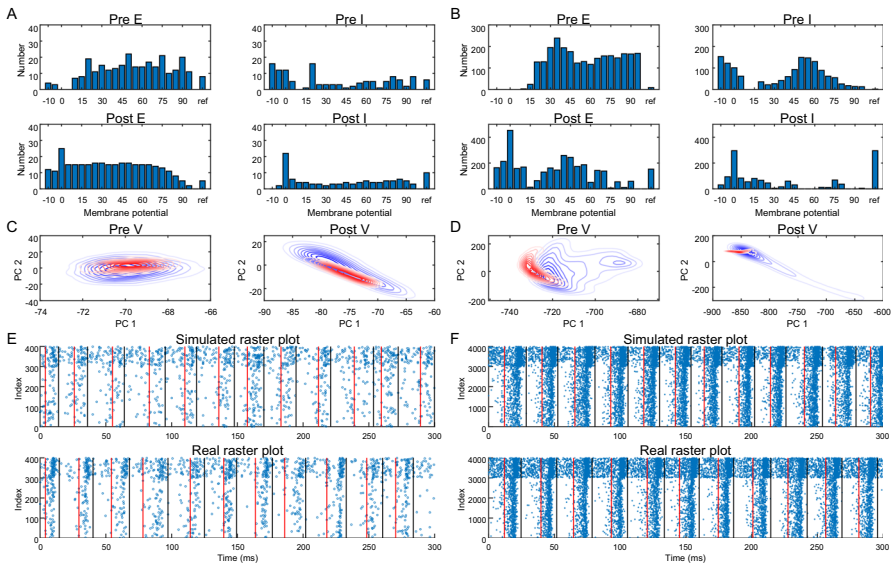
Fig. 5 Surrogates of spiking network dynamics produced by the parameter-specific MFE mapping $\widehat{F}_1^\theta$. **A, C, E** resembling a 400-neuron spiking network; **B, D, F** resembling a 4000-neuron network. **A, B** Example of pre and post-MFE voltage distributions p^E and p^I in the surrogate dynamics. **C, D** 10th level curves of the first two principal components of p^E and p^I. (Blue: 20k examples from the enlarged training set, red: 2k examples from the surrogate dynamics.) **E, F** Raster plots of simulated surrogate dynamics and the real dynamics starting from the same initial profiles (color figure online)

(ii) Sample $\omega(s_m)$ from $\mathcal{C}^{-1}(\tilde{\omega}_{s_m})$.

(iii) Evolve the network dynamics with the tau-leaping method and initial condition $\omega(s_m)$. Stop the simulation with terminal value $\omega_{s_{m+1}}$.

(iv) $x_{m+1} = \mathcal{C}(\omega_{s_{m+1}})$.

(v) Repeat (i)-(iv) with $m = m + 1$.

We demonstrate the resembling power by producing the raster plot of SNN dynamics labeling all spiking events with neuron index *vs.* time in Figs. 5 and 6. While the spiking events during IMIs are given by tau-leaping simulation, the spiking patterns during MFEs consist of events uniformly randomly assigned to neurons and times within the MFE interval (with the total number of spikes $(\mathrm{Sp}_E, \mathrm{Sp}_I)$ predicted by our DNN). The durations of the MFEs are randomly sampled from the empirical distribution collected from the SNN simulations. In the rest of the section, we give the details of our surrogate SNN dynamics constructed upon $\widehat{F}_1^\theta$ and $\widehat{F}_1$.

Parameter-specific predictions. We use the enlarged training set as described in Section 4 to generate $\widehat{F}_1^\theta(x_m)$. Here, $\theta = (S^{EE}, S^{IE}, S^{EI}, S^{II}) = (4, 3, -2.2, -2)$ for the 400-neuron SNN, and the synaptic weights are normalized by 10 times in the 4000-neuron SNN.

We first focus on the surrogate dynamics of the 400-neuron SNN. Figure 5A gives examples of pre-MFE voltage distributions p^Q generated by the tau-leaping simulation and post-MFE p^Q generated by DNN predictions. We further compare the distributions

Fig. 6 Surrogates of spiking network dynamics produced by the parameter-generic MFE mapping $\widehat{F}_1$. **A-F** are in parallel to Fig. 5

of the first two *principal components* of $\boldsymbol{p}^Q$ occurring in the surrogate dynamics and the training set (Fig. 5B. red: surrogate dynamics; blue: the training set). On the plane of the first two principal components, voltage distributions produced by surrogate dynamics distribute consistently with training sets, suggesting that the surrogate dynamics align very well with the ground-truth network dynamics. The red/blue contours indicate each tenth of the level curves of the ks-density of the distributions. We also compare the raster plots of the two dynamics, where the initial SNN conditions are the same (Fig. 5E).

The right half of Fig. 5 depicts the surrogate dynamics to the 4000-neuron SNN model. The biased distribution of the principal components of $\boldsymbol{p}^Q$ is probably due to **A.** the relatively more narrow training sets (see discussions in Sect. 4.3), and **B.** the large 1 ms timestep in tau-leaping.

Parameter-generic predictions. The surrogate dynamics generated by the parameter-generic surrogate MFE mapping $\widehat{F}_1$ and tau-leaping are depicted in Fig. 6, whose panels are analogous to Fig. 5. (The "real" raster plots of SNN dynamics in Fig. 6E is fixed the same as Fig. 6E.) The comparable results demonstrate that the DNNs produce faithful surrogates to the SNN dynamics. Interestingly, by comparing panels C and D of Figs. 5 and 6, we find that the principal components of $\boldsymbol{p}^Q$ in the surrogate dynamics generated by $\widehat{F}_1$ are much less biased than $\widehat{F}_1^\theta$. While a detailed explanation is beyond the current scope of this study, our current conjecture is that the more general training set of $\widehat{F}_1$ helps it deal with the less regular pre-MFE network states in the surrogate dynamics.

6 Discussions and conclusions

In this paper, we build an artificial neural network (ANN) surrogate of γ-dynamics arising from a biological neuronal circuit. The neuronal circuit model is a stochastic integrate-and-fire network that has been well-studied. Similar to many other models (Li et al. 2019; Zhang 2014; Rangan and Young 2013), it can exhibit semi-synchronous spiking activities called the multiple-firing events (MFEs), which are transient and highly nonlinear emergent phenomena of spiking neuronal networks.

In our study, the sensitive & transient MFE dynamics are represented by the MFE mappings that project the pre-MFE network states to post-MFE network states. The MFE mappings are faithfully approximated by ANNs, despite the significant intrinsic noise in the model. On the other hand, the slower and quieter dynamics between consecutive MFEs are evolved by standard tau-leaping simulations. Remarkably, a surrogate of spiking network dynamics is produced by combining ANN approximations and tau-leaping simulations, generating firing patterns consistent with the spiking network dynamics. Furthermore, the ANN surrogate can be generalized to a wide range of synaptic coupling strengths.

This paper explores the methodology of learning biological neural circuits with ANNs. In this study, the biggest challenges of developing a successful ANN surrogate are **A.** processing the high-dimensional, noisy data and **B.** building a representative training set. Both challenges are addressed by the first-principle-based model reduction techniques, i.e., coarse-graining and discrete cosine transform. The model reductions remove the excessive intrinsic noise. The training set collects network states from simulations of SNNs and is enlarged to represent a broader class of voltage distributions. Therefore, the training set covers the "rare" voltage distributions occurring in spiking network dynamics with low probabilities.

Future work. The idea of ANN surrogates elaborated in this paper can be extended and applied to other network models. First, many models of brain dynamics share the difficulties of dimensionality, robustness, and generalizability. Therefore, we propose to extend our ideas to model more sophisticated dynamical phenomena of the brain, such as other types of neural oscillations, and to neural circuits with more complicated network architecture. Furthermore, the power of ANNs to handle large data sets may allow us to extend our framework to deal with experimental data directly. In general, we are motivated by the recent success demonstrating the capability of deep neural networking in representing infinite-dimensional maps (Li et al. 2020; Lu 2021; Wang et al. 2021). Therefore, in future work, we suggest exploring more complex network structures (e.g., the DeepONet Lu 2021) to build mappings between the states of neural circuits.

Another interesting but challenging issue is the *interpretability* of ANN surrogates, e.g., relating statistics, dynamical features, and architectures of the spiking network models to the ANNs. A potentially viable approach is to map neurons in the spiking networks to artificial neurons, then examine the connection weight after training. However, it is likely that this idea may need ANNs more complicated than the simple feed-forward DNN we considered here. To achieve this goal, one may consider different ANNs with different architectures, such as ResNet or LSTM (He et al. 2016;

Hochreiter and Schmidhuber 1997). These studies may shed some light on how the dynamics and information flow in neural systems are represented in ANNs.

7 Appendix

7.1 Tau-leaping and SSA algorithms

The simulations of the SNN dynamics are carried out by two algorithms: Tau-leaping and Stochastic Simulation Algorithm (SSA). The key difference is that, The tau-leaping method processes events that happen during a time step τ in bulk, while SSA simulates the evolution event by event. Of the two, tau-leaping can be faster (with properly chosen τ), while SSA is usually more precise with the precision that scales with C++ execution. Here we illustrate a Markov jump process as an example.

Algorithms. Consider $X(t) = \{x_1(t), x_2(t), ..., x_N(t)\}$, where $X(t)$ can take values in a discrete state space

$$S = \{s_1, s_2, \ldots, s_M \subset \mathbb{R}^N\}.$$

The transition from state X to state s_i at time t is denoted as $T_{s_i}^t(X)$, taking an exponential distributed waiting time with rate $\lambda_{s_i \leftarrow X}$. Here, $s_i \in S(X)$ which are states adjacent to state X with a non-zero transition probability. For simplicity, we assume $\lambda_{s_i \leftarrow X}$ does not explicitly depend on t except via $X(t)$.

Tau-leaping only considers $X(t)$ on a time grid $t = jh$, for $j = 0, 1, ..., T/h$, assuming state transfer occurs for at most one time within each step:

$$P(X^{(j+1)h} = s_i)$$
$$= \begin{cases} h\lambda_{s_i \leftarrow X^{jh}} & \forall s_i \in S(X^{jh}), \\ 1 - h \sum_{s_i \in S(X^{jh})} \lambda_{s_i \leftarrow X^{jh}} & s_i = X^{jh}, \\ 0 & \text{otherwise}, \end{cases}$$

On the other hand, SSA accounts for this simulation problem as:

$$X(T) = T_{X_k}^{t_k} \circ T_{X_{k-1}}^{t_{k-1}} \circ \ldots \circ T_{X_1}^{t_1}(X(0)),$$

i.e., starting from X^0, X transitions to $X_1, X_2, \ldots, X_k = X(T)$ at time $0 < t_1 < t_2 < \ldots < t_k < T$.

For $t_\ell < t < t_{\ell+1}$, we sample the transition time from $\mathrm{Exp}(\sum_{s_i \in S(X(t))} \lambda_{s_i \leftarrow X(t)})$. That is, for independent, exponentially distributed random variables

$$\tau_i \sim \mathrm{Exp}(\lambda_{s_i \leftarrow X(t)}),$$

we have

$$t_{\ell+1} - t_\ell = \min_{s_i \in S(X(t))} \tau_i \sim \mathrm{Exp}\left(\sum_{s_i \in S(X(t))} \lambda_{s_i \leftarrow X(t)}\right).$$

Therefore, in each step of an SSA simulation, the system state evolves forward by an exponentially distributed random time, whose rate is the sum of rates of all exponential "clocks". Then we randomly choose the exact state s_i to which transition takes place with probability weighted by the sizes of the pending events.

Implementation on spiking networks. We note that $X(t)$ will changes when

1. neuron i receives external input (v_i goes up for 1, including entering $\mathcal{R}$);
2. neuron i receives a spike (H_i^E or H_i^I goes up for 1);
3. a pending spike takes effect to neuron i (v_i goes up/down according to synaptic strengths);
4. neuron i walks out from refractory (v_i goes from $\mathcal{R}$ to 0).

The corresponding transition rates are directly given (λ^E and λ^I) or the inverses of the physiological time scales (τ^E, τ^I, and $\tau^{\mathcal{R}}$). In an SSA simulation, when the state transition elicits a spike in a neuron, the synaptic outputs generated by this spike are immediately added to the pool of corresponding types of effects, and the neuron goes into the refractory state. However, in a tau-leaping simulation, the spikes are recorded but the synaptic outputs are processed in bulk at the end of each time step. Therefore, all events within the same time step are uncorrelated.

7.2 The coarse-graining mapping

Here we give the definition of the coarse-grained mapping $\mathcal{C}$ in Eq. 6. For $\forall \omega \in \boldsymbol{\Omega}$ that

$$\begin{aligned} \omega = (&V_1, \ldots, V_{N_E}, V_{N_E+1}, \ldots, V_{N_E+N_I}, \\ &H_1^E, \ldots, H_{N_E}^E, H_{N_E+1}^E, \ldots, H_{N_E+N_I}^E, \\ &H_1^I, \ldots, H_{N_E}^I, H_{N_E+1}^I, \ldots, H_{N_E+N_I}^I), \end{aligned}$$

we define

$$\begin{aligned} \mathcal{C}(\omega) = \tilde{\omega} = (&n_1^E, n_2^E, \ldots, n_{22}^E, n_R^E, n_1^I, n_2^I, \ldots, n_{22}^I, n_R^I, \\ &H^{EE}, H^{EI}, H^{IE}, H^{II}), \end{aligned}$$

where,

$$n_i^E = \sum_{j=1}^{N_E} \mathbf{1}_{\Gamma_i}(V_j), \quad \text{for } i = 1, \ldots, 22;$$

$$n_R^E = \sum_{j=1}^{N_E} \mathbf{1}_{\{\mathcal{R}\}}(V_j);$$

$$n_i^I = \sum_{j=N_E+1}^{N_E+N_I} \mathbf{1}_{\Gamma_i}(V_j), \quad \text{for } i = 1, \ldots, 22;$$

$$n_R^I = \sum_{j=N_E+1}^{N_E+N_I} \mathbf{1}_{\{\mathcal{R}\}}(V_j);$$

and

$$H^{EE} = \sum_{j=1}^{N_E} H_j^E; \qquad H^{IE} = \sum_{j=N_E+1}^{N_E+N_I} H_j^E;$$

$$H^{EI} = \sum_{j=1}^{N_E} H_j^I; \qquad H^{II} = \sum_{j=N_E+1}^{N_E+N_I} H_j^I.$$

Here, $\mathbf{1}_A(a)$ is an indicator function of set A, i.e., $\mathbf{1}_A(a) = 1 \; \forall a \in A$, otherwise $\mathbf{1}_A(a) = 0$. Γ_i is a subset of the state space for membrane potential, and

$$\Gamma_i = [-15 + 5i, -10 + 5i) \cap \Gamma.$$

7.3 Pre-processing surrogate data: discrete cosine transform

Here we explain how discrete cosine transform (DCT) works in the pre-processing. For an input probability mass vector

$$\boldsymbol{p} = (n_1, n_2, \cdots, n_{22}),$$

its DCT output $\mathcal{F}_c(\boldsymbol{p}) = (c_1, c_2, ..., c_{22})$ is given by

$$c_k = \sqrt{\frac{2}{22}} \sum_{l=1}^{22} \frac{n_l}{\sqrt{1 + \delta_{kl}}} \cos\left(\frac{\pi}{2N}(2l - 1)(k - 1)\right), \tag{9}$$

where δ_{kl} is the Kronecker delta function. The iDCT mapping $\mathcal{F}_c^{-1}$ is defined as the inverse function of $\mathcal{F}_c$.
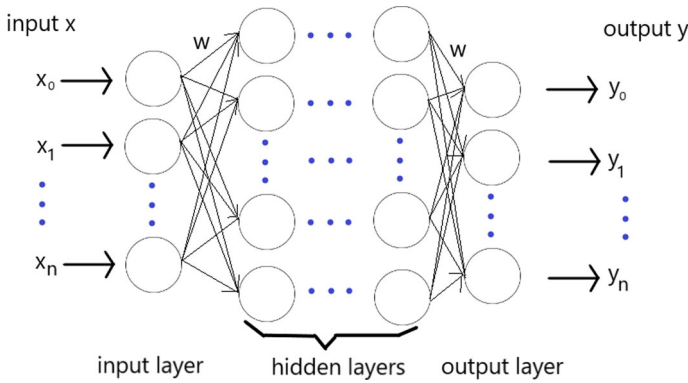
Fig. 7 Diagram of a feedforward ANN

7.4 The linear formula for firing rates

When preparing the parameter-generic training set, we use simple, linear formulas to estimate the firing rate of E neurons and I neurons (f_E and f_I, see Li and Hui 2019; Li et al. 2019). We take $\theta \in \Theta$ for the synaptic coupling strength, while other constants are the same as in Table 1.

$$f_E = \frac{\lambda^E(M + C^{II}) - \lambda^I C^{EI}}{(M - C^{EE})(M + C^{II}) + (C^{EI}C^{IE})}$$
$$f_I = \frac{\lambda^I(M - C^{EE}) + \lambda^E C^{IE}}{(M - C^{EE})(M + C^{II}) + (C^{EI}C^{IE})}$$

where

$$C^{EE} = N^E P^{EE} S^{EE}, \ C^{IE} = N^E P^{IE} S^{IE}$$
$$C^{EI} = N^I P^{EI} S^{EI}, \ C^{II} = N^I P^{II} S^{II}.$$

7.5 The deep network architecture

In general, artificial neural networks (ANNs) are interconnected computation units. Many different architectures are possible for ANNs; in this paper, we adopt the *feedforward* deep network architecture, which is one of the simplest (Fig. 7).

A feedforward ANN has a layered structure, where units in the $i-$th layer drive the $(i + 1)-$th layer with a weight matrix W_i and a bias vector b_i. Computation is processed from one layer to the next. The first, "input layer" takes an input vector x, sending its output $W_1 x + b_1$ to the first "hidden layer"; the first hidden layer then sends output $W_2 f(W_1 x + b_1) + b_2$ to the next layer, and so on, until the last, "output layer" produces an output vector y. In this paper, we implemented a feedforward ANN with four layers containing 512, 512, 512, and 128 neurons, respectively. We chose the
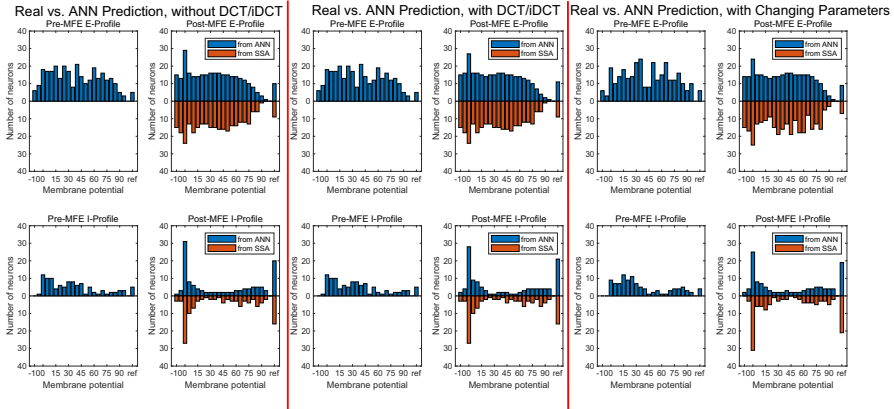
Fig. 8 **Left:** pre-, post- and predicted MFE profiles without DCT + iDCT; **Middle:** pre-, post- and predicted MFE profiles with DCT + iDCT; **Right:** pre-, post- and predicted MFE profiles with network parameters as additional inputs of ANN. (Left and Middle: S^{EE}, S^{IE}, S^{EI}, S^{II} = 4, 3, −2.2, -2; Right: 3.82, 3.24, −2.05, −1.87.)

Leaky ReLU function with a default negative slope of 0.01 as our activation function $f(\cdot)$.

The training of feedforward ANNs is achieved by the back-propagation (BP) algorithm. Let $\mathcal{NN}(x)$ denote the prediction of the ANN with input x, and $L(\cdot)$ the loss function. With each entry (x, y) in the training data, we minimize the loss $L(y - \mathcal{NN}(x))$ following the gradients on each dimension of W_i and b_i. The computation of gradients takes place from the last layer W_n's and b_n, then "propagated back" to adjust previous W_i and b_i on each layer. We chose the mean-square error as our loss function, i.e. $L(\cdot) = || \cdot ||_{L^2}^2$.

7.6 Pre-processing in ANN predictions

Here we provide more examples of the ANN predictions of the voltage profiles. We compare how ANNs predict post-MFE voltage distributions $\boldsymbol{p}^E$ and $\boldsymbol{p}^I$ in three different settings in Fig. 8. In each panel divided by red lines, the left column gives an example of pre-MFE voltage distributions, while the right column compares the corresponding post-MFE voltage distributions collected from ANN predictions (red) vs. SSA simulation. Results from ANNs without pre-processing, with pre-processing, and the parameter-generic ANN are depicted in the left, middle, and right panels.

7.7 Principal components of voltage distributions

The voltage distribution vectors in the form below are used to plot the distribution in the phase space as shown in middle panels of Figs. 5, 6, 9D, 10D, 11D, 12D, 13, 14, and 15

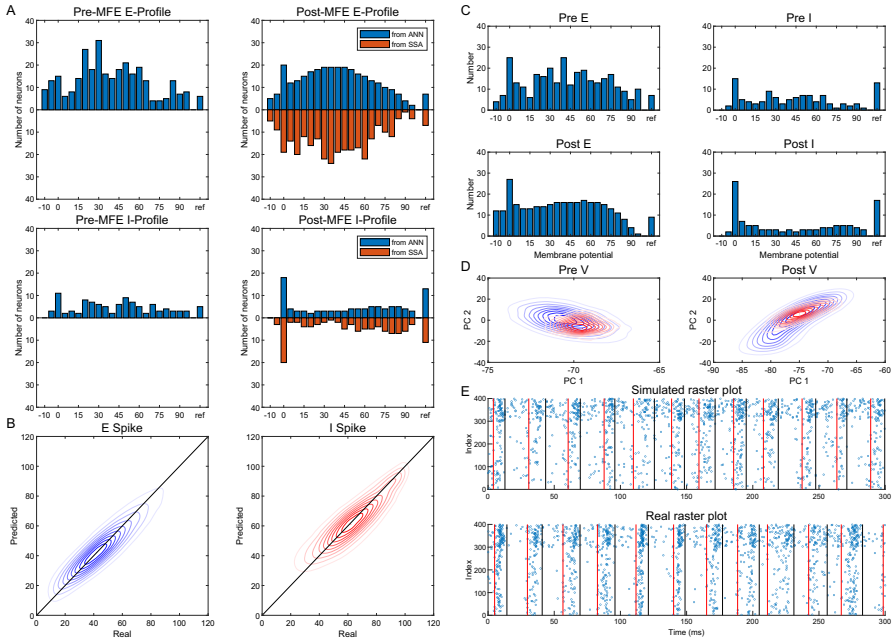$$(n_1^E, n_2^E, \ldots, n_{22}^E, n_R^E, n_1^I, n_2^I, \ldots, n_{22}^I, n_R^I)$$
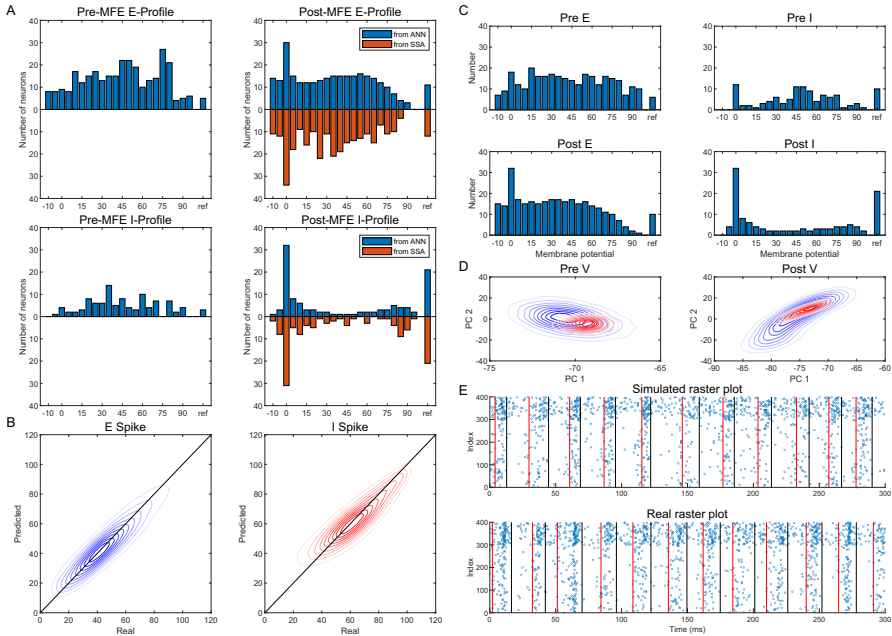
Fig. 9 DNN predictions and surrogate dynamics in ER random network with 400 neurons. **A.** Mapping $\widehat{F}_1^\theta$ in ER network for $\theta = (S^{EE}, S^{EI}, S^{IE}, S^{II}) = (4, 3, -2.2, -2)$. **A.** Left: a pre-MFE $\boldsymbol{p}^E(v)$ and a $\boldsymbol{p}^I(v)$; Right: post-MFE $\boldsymbol{p}^E(v)$ and $\boldsymbol{p}^I(v)$ produced by ANN (blue) vs. spiking network simulations (orange). **B.** Comparison of E and I spike number during MFEs, ANN predictions vs. SSA simulations. The distributions are depicted by 10th contours of max in ks-density estimation; **C.** Example of pre and post-MFE voltage distributions $\boldsymbol{p}^E$ and $\boldsymbol{p}^I$ in the surrogate dynamics. **D.** Distribution depicted by 10th-contours of the first two principal components of $\boldsymbol{p}^E$ and $\boldsymbol{p}^I$. **E.** Raster plots of simulated surrogate dynamics and the real dynamics starting from the same initial profiles (color figure online)

The vectors from the training set (colored in blue in figures) are selected to generate the basis of the phase space through **svd** function in *numpy.linalg* in Python. The first two rows of the $V^\top$ are the first two PCs of the space. The scores of vectors from the training set and approximated results are dot products of these vectors and the normalized PCs.

The plain ks-density function in MATLAB is used to estimate the kernel smoothing density of the profile distribution based on the data points generated above. The contours show the level of each tenth of the maximal height (with 0.1% bias for demonstrating the top) in the distributions.

7.8 Consistent results from fixed random networks

Here we test the capability of our method with fixed network architectures. We select two types of random graphs: 1. Erd?s-Rényi random graph (ER), and 2. random graphs with log-normal degree distribution (LN). In both types of graphs, the average edge density is consistent with Ps in Table 1. Their adjacency matrices are shown in Fig. 16. The sampling of four types of edges in LN random graphs leverages a

Fig. 10 DNN predictions and surrogate dynamics in a 400-neuron random network with log-normal degree distribution. **A-E** are in parallel to Fig. 9
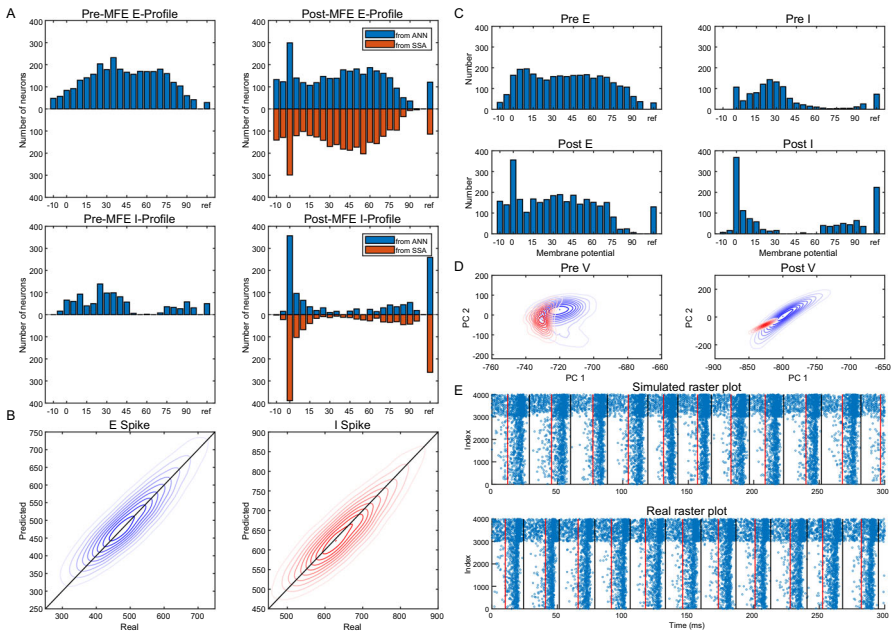
Fig. 11 DNN predictions and surrogate dynamics in ER random network with 4000 neurons. **A-E** are in parallel to Fig. 9, except that $\theta = (S^{EE}, S^{EI}, S^{IE}, S^{II}) = (0.4, 0.3, -0.22, -0.2)$
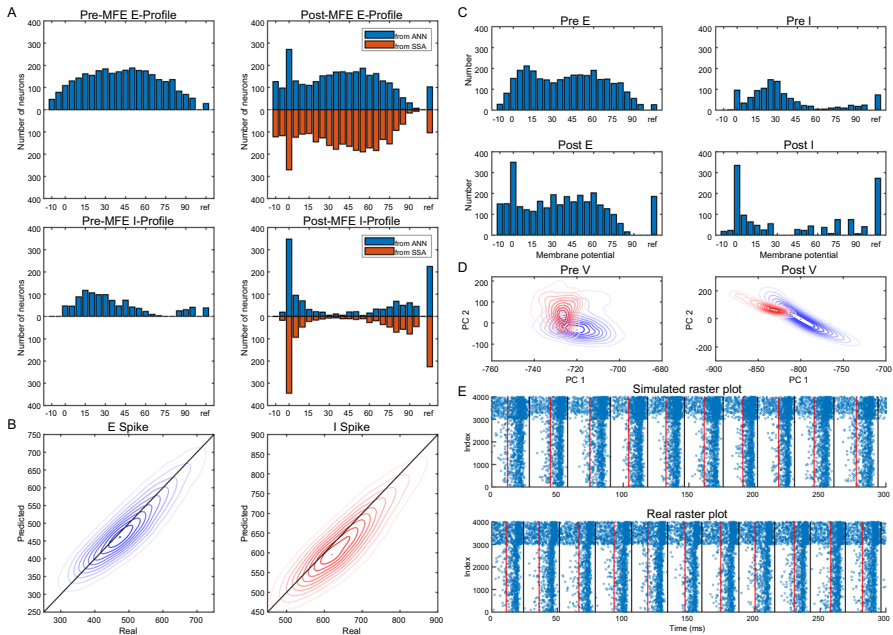
Fig. 12 DNN predictions and surrogate dynamics in a 4000-neuron random network with log-normal degree distribution. **A-E** are in parallel to Fig. 11

standard deviation of 0.2 in logarithm and constraints from the mean degrees. Results are obtained from both sizes, 400 and 4000, of the random graphs. For each network, MFEs are captured from original network simulations and simulations with enlarged initial profiles (Fig. 18). The trained parameter-specific MFE mappings $\widehat{F}_1^\theta$ are used to produce predictions of post-MFE states and surrogate network dynamics (Figs. 9, 10, 11, 12). MFEs are further captured in simulations with various parameter sets sampled from the 4D cube Θ. The trained parameter-generic $\widehat{F}_1$ is used to produce predictions and surrogate dynamics, which are shown in Figs. 13 and 14. As seen in Figs. 9, 10, 11, 12, 13 and 14, the performance of the ANN surrogate is consistent with the case when postsynaptic connections are decided on-the-fly.

7.9 Varying the synaptic coupling strengths generates a broad range of firing rates and magnitudes of MFEs

The training of parameter-generic MFE mapping $\widehat{F}_1$ needs MFEs from simulations with a variety of sets of parameter θ. As introduced in Sect. 4.3, the sets sampled from the 4D cube Θ are first filtered by the estimated firing rate computed by the linear formula. A large fraction (about 80%) of the sets from the previous step generate MFEs that can be captured and accepted by our algorithm. These sets generate a wide range of firing rates (Fig. 19). The simulated firing rates match the linear formula in general. The major rejected region appears with high S^{EE}, S^{IE} and low S^{II}, S^{EI},
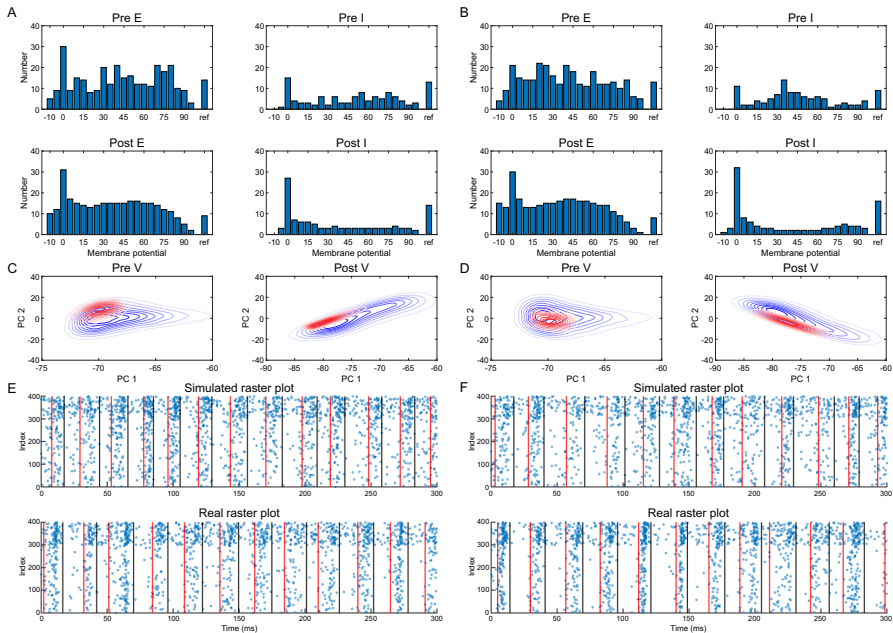
Fig. 13 Surrogate dynamics produced by parameter-generic MFE mapping $\widehat{F}_1$ in two fixed networks with 400 neurons. **Left**: ER network; **Right**: Network with log-normal degree distribution. **A-B**: Example of pre and post-MFE voltage distributions p^E and p^I in the surrogate dynamics. **C-D**: Distribution depicted by 10th-contours of the first two principal components of p^E and p^I. **E-F**: Raster plots of simulated surrogate dynamics and the real dynamics starting from the same initial profiles

which is in the neighborhood with the accepted region with high firing rate and near the singular region of the linear formula.

The magnitudes of MFEs (number of spikes) from various parameters show a much wider distribution than the original set of parameters (Fig. 17).

7.10 Extrapolating network dynamics out of Θ with parameter-generic MFE mapping $\widehat{F}_1$

To test the extrapolation ability of our method, we generate surrogate dynamics in networks with θ's outside of Θ with the parameter-generic MFE mapping $\widehat{F}_1$. The two θ's are $(S^{EE}, S^{IE}, S^{EI}, S^{II}) = (5, 3, -2.2, -2)$ and $(S^{EE}, S^{IE}, S^{EI}, S^{II}) = (4, 4, -2.2, -2)$.

Parameter-generic MFE mapping $\widehat{F}_1$ trained with MFEs in Θ can still capture the neuronal oscillations and predict post-MFE states in the two networks. (Fig. 15) Behaviors of networks under these two sets of parameters are also reproduced. Strong recurrent excitation in the former θ makes MFEs readily to be concatenated, while the less-synchronized character of the latter θ makes MFEs hard to trigger or identify. This result shows the robustness and capability of extrapolation of our method, while our future work can focus on improving the precision of prediction in detail (Figs. 18, 19).
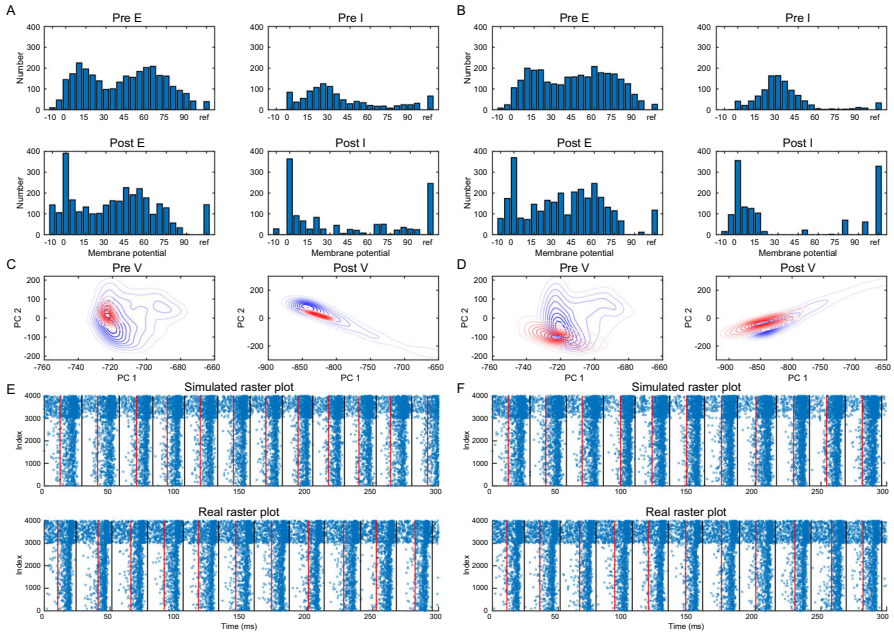
Fig. 14 Surrogate dynamics produced by parameter-generic MFE mapping $\widehat{F}_1$ in two fixed networks with 4000 neurons. **Left**: ER network; **Right**: Network with log-normal degree distribution. **A-F** are in parallel to Fig. 13
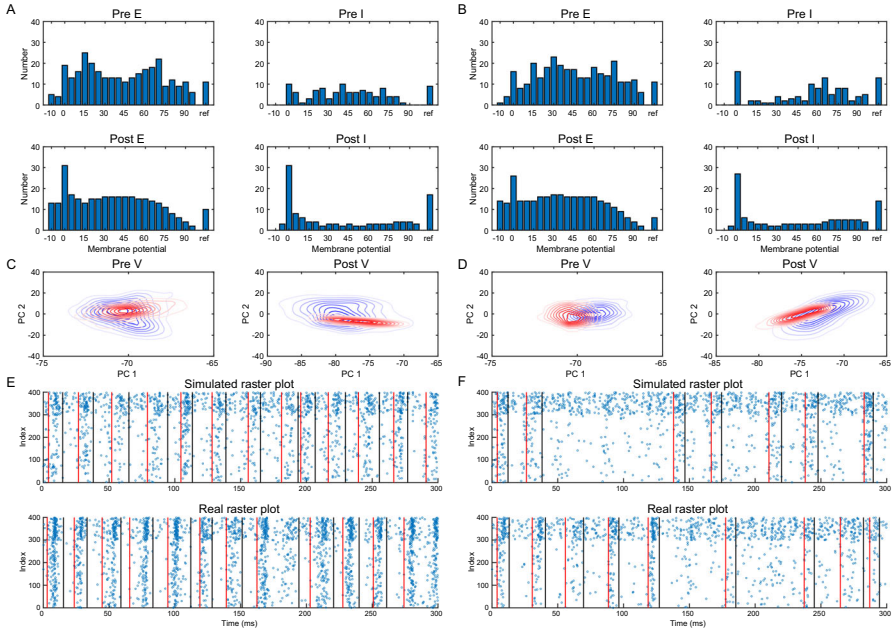
Fig. 15 Surrogate dynamics with a parameter set that lies out of the sampling 4D cube produced by parameter-generic MFE mapping $\widehat{F}_1$. **Left**: $(S^{EE}, S^{IE}, S^{EI}, S^{II}) = (5, 3, -2.2, -2)$; **Right**: $(S^{EE}, S^{IE}, S^{EI}, S^{II}) = (4, 4, -2.2, -2)$. **A-F** are in parallel to Fig. 13
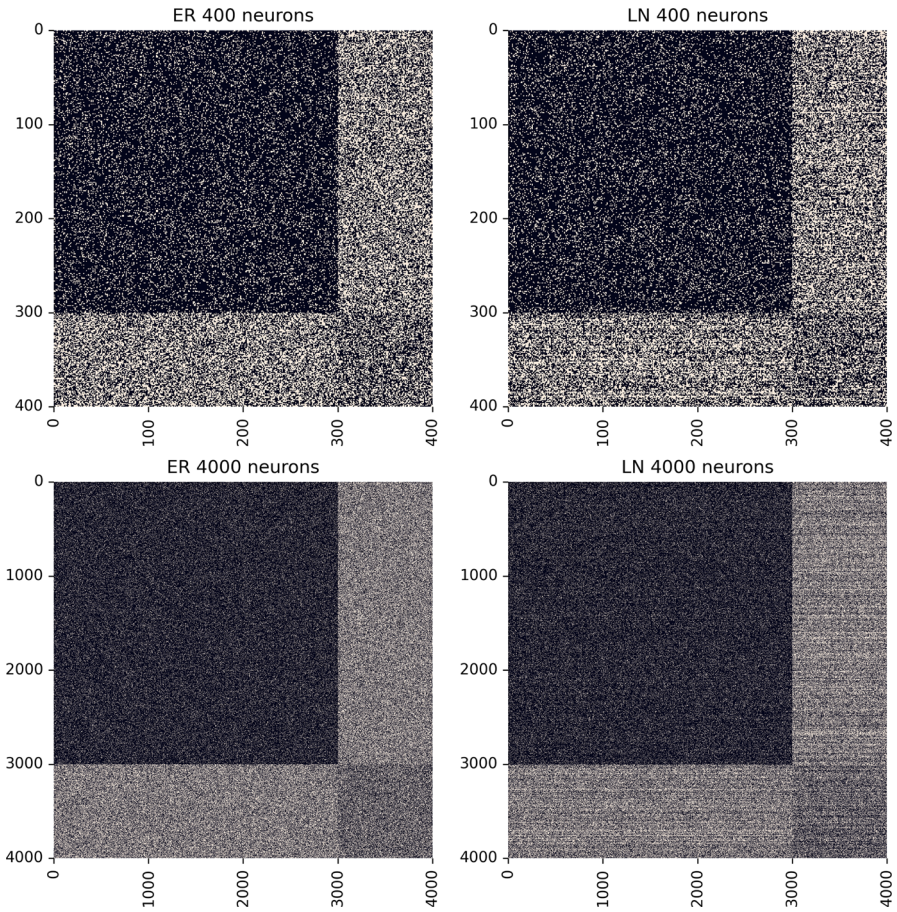
Fig. 16 Fixed random graphs used for SNN simulation. ER: Erd?s-Rényi random graph. LN: random graphs with log-normal degree distribution. A light-colored block at (i, j) represents a directed synaptic connection from i to j. The last quarter of the neurons are inhibitory, while the others are excitatory. This figure is compressed to reduce the file size
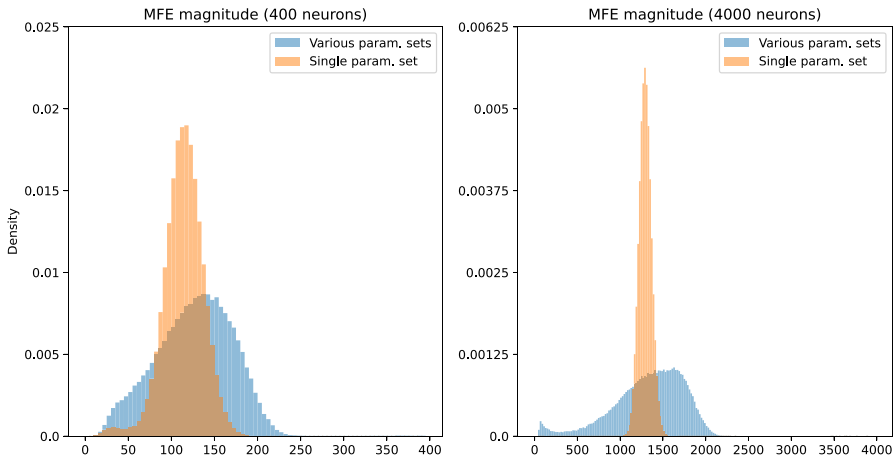
Fig. 17 Distributions of the magnitude of MFEs (number of spikes) from sampled parameters sets and the specific parameter set $(S^{EE}, S^{IE}, S^{EI}, S^{II}) = (4, 3, -2.2, -2)$
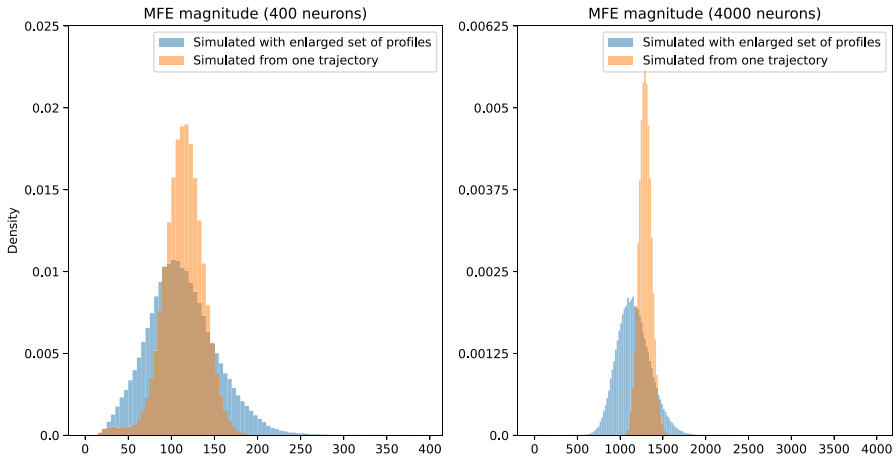
Fig. 18 Distributions of the magnitude of MFEs (number of spikes) from the enlarged set of initial profiles and a single trajectory
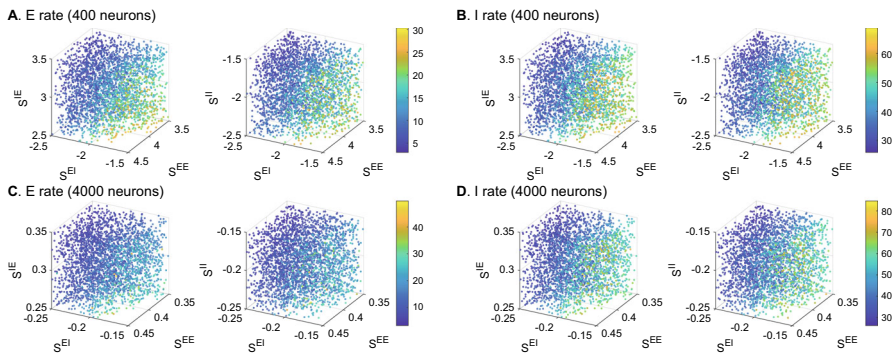
Fig. 19 Firing rates of networks with parameter sets that can produce accepted MFEs for training. (n=3000)

References

Aggarwal CC et al (2018) Neural networks and deep learning, vol 10. Springer, Cham, p 3

AlQuraishi M, Sorger PK (2021) Differentiable biology: using deep learning for biophysics-based and data-driven modeling of molecular mechanisms. Nat Methods 18(10):1169–1180

Andrew Henrie J, Shapley R (2005) LFP power spectra in V1 cortex: the graded effect of stimulus contrast. J Neurophysiol 94(1):479–490

Azouz R, Gray CM (2000) Dynamic spike threshold reveals a mechanism for synaptic coincidence detection in cortical neurons in vivo. Proc Natl Acad Sci 97(14):8110–8115

Azouz R, Gray CM (2003) Adaptive coincidence detection and dynamic gain control in visual cortical neurons in vivo. Neuron 37:513–523

Barron AR (1994) Approximation and estimation bounds for artificial neural networks. Mach Learn 14(1):115–133

Bauer M et al (2006) Tactile spatial attention enhances gamma-band activity in somatosensory cortex and reduces low-frequency activity in parieto-occipital areas. J Neurosci 26(2):490–501

Bauer EP, Paz R, Paré D (2007) Gamma oscillations coordinate Amygdalo-Rhinal interactions during learning. J Neurosci 27(35):9369–9379

Börgers C, Kopell N (2003) Synchronization in networks of excitatory and inhibitory neurons with sparse, random connectivity. Neural Comput 15(3):509–538

Bressloff PC (1994) Dynamics of compartmental model recurrent neural networks. Phys Rev E 50(3):2308

Brosch M, Budinger E, Scheich H (2002) Stimulus-related gamma oscillations in primate auditory cortex. J Neurophysiol 87(6):2715–2725

Brunel N, Hakim V (1999) Fast global oscillations in networks of integrate-and-fire neurons with low firing rates. Neural Comput 11(7):1621–1671

Buice MA, Cowan JD (2007) Field-theoretic approach to fluctuation effects in neural networks. Phys Rev E 75(5):051919

Buschman TJ, Miller EK (2007) Top-down versus bottom-up control of attention in the prefrontal and posterior parietal cortices. Science 315:1860–1862

Cai D et al (2006) Kinetic theory for neuronal network dynamics. Commun Math Sci 4(1):97–127

Cai Y et al (2021) Model reduction captures stochastic Gamma oscillations on low-dimensional manifolds. Front Comput Neurosci 15:74

Chariker L, Young L-S (2015) Emergent spike patterns in neuronal populations. J Comput Neurosci 38(1):203–220

Chariker L, Shapley R, Young L-S (2016) Orientation selectivity from very sparse LGN inputs in a comprehensive model of macaque V1 cortex. J Neurosci 36(49):12368–12384

Chariker L, Shapley R, Young L-S (2018) Rhythm and synchrony in a cortical network model. J Neurosci 38(40):8621–8634

Chon KH, Cohen RJ (1997) Linear and nonlinear ARMA model parameter estimation using an artificial neural network. IEEE Trans Biomed Eng 44(3):168–174

Christof K (1999) Biophysics of computations. Oxford University Press, Oxford

Csicsvari J et al (2003) Mechanisms of gamma oscillations in the hippocampus of the behaving rat. Neuron 37:311–322

Erol B (2013) A review of gamma oscillations in healthy subjects and in cognitive impairment. Int J Psychophysiol 90(2):99–117. https://doi.org/10.1016/j.ijpsycho.2013.07.005

Frien A et al (2000) Fast oscillations display sharper orientation tuning than slower components of the same recordings in striate cortex of the awake monkey. Eur J Neurosci 12(4):1453–1465

Fries P et al (2001) Modulation of oscillatory neuronal synchronization by selective visual attention. Science 291:1560–1563

Fries P et al (2008) The effects of visual stimulation and selective visual attention on rhythmic neuronal synchronization in macaque area V4. J Neurosci 28(18):4823–4835

Gerstner W et al (2014) Neuronal dynamics: from single neurons to networks and models of cognition. Cambridge University Press, Cambridg

Ghosh-Dastidar S, Adeli H (2009) Spiking neural networks. Int J Neural Syst 19(04):295–308

Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. In: arXiv preprint arXiv:1412.6572

Hasenauer J et al (2015) Data-driven modelling of biological multi-scale processes. J Coupled Syst Multi-scale Dyn 3(2):101–121

He K et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Hodgkin AL, Huxley AF (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. J Physiol 117(4):500

Jack RE, Crivelli C, Wheatley T (2018) Data-driven methods to diversify knowledge of human psychology. Trends cognit Sci 22(1):1–5

Janes KA, Yaffe MB (2006) Data-driven modelling of signal-transduction networks. Nat Rev Mol Cell Biol 7(11):820–828

Krystal JH et al (2017) Impaired tuning of neural ensembles and the pathophysiology of schizophrenia: a translational and computational neuroscience perspective. Biol Psychiatr 81(10):874–885

Li Z et al (2020) Fourier neural operator for parametric partial differential equations. arXiv preprint arXiv:2010.08895

Li H et al (2020) NETT: solving inverse problems with deep neural networks. Inverse Probl 36(6):065005

Li Y, Hui X (2019) Stochastic neural field model: multiple firing events and correlations. J Math Biol 79(4):1169–1204

Li Y, Chariker L, Young L-S (2019) How well do reduced models capture the dynamics in models of interacting neurons? J Math Biol 78(1):83–115

Liu J, Newsome WT (2006) Local field potential in cortical area MT: stimulus tuning and behavioral correlations. J Neurosci 26(30):7779–7790

Lu L et al (2021) Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nat Mach Intell 3(3):218–229

Mably AJ, Colgin LL (2018) Gamma oscillations in cognitive disorders. Current Opin Neurobiol 52:182–187

Nikola K, Samuel L, Siddhartha M (2021) On universal approximation and error bounds for fourier neural operators. J Mach Learn Res 22:1–76

Nobukawa S, Nishimura H, Yamanishi T (2017) Chaotic resonance in typical routes to chaos in the Izhikevich neuron model. Sci Rep 7(1):1–9

Pesaran B et al (2002) Temporal structure in neuronal activity during working memory in macaque parietal cortex. Nat Neurosci 5(8):805–811

Pieter Medendorp W et al (2007) Oscillatory activity in human parietal and occipital cortex shows hemispheric lateralization and memory effects in a delayed double-step saccade task. Cereb Cortex 17(10):2364–2374

Ponulak F, Kasinski A (2011) Introduction to spiking neural networks: information processing, learning and applications. Acta Neurobiol Exp 71(4):409–433

Popescu AT, Popa D, Paré D (2009) Coherent gamma oscillations couple the amygdala and striatum during learning. Nature Neurosci 12(6):801–807

Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 378:686–707

Rangan AV, Young L-S (2013) Emergent dynamics in a model of visual cortex. J Comput Neurosci 35(2):155–167

Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117

Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6(1):1–48

Solle D et al (2017) Between the poles of data-driven and mechanistic modeling for process operation. Chem Ing Tech 89(5):542–561

Tao L et al (2006) Orientation selectivity in visual cortex by fluctuation-controlled criticality. Proc Natl Acad Sci 103(34):12911–12916

Traub RD et al (2005) Single-column thalamocortical network model exhibiting gamma oscillations, sleep spindles, and epileptogenic bursts. J Neurophysiol 93(4):2194–2232

Van Der Meer MAA, David Redish A (2009) Low and high gamma oscillations in rat ventral striatum have distinct relationships to behavior, reward, and spiking activity on a learned spatial decision task. Front Integr Neurosci 3:9

van Wingerden M et al (2010) Learning-associated gamma-band phase-locking of action-outcome selective neurons in orbitofrontal cortex. J Neurosci 30(30):10025–10038

Wang S, Wang H, Perdikaris P (2021) Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. Sci Adv 7(40):eabi8605

Whittington MA et al (2000) Inhibition-based rhythms: experimental and mathematical observations on network dynamics. Int J Psychophysiol 38(3):315–336

Wilson HR, Cowan JD (1972) Excitatory and inhibitory interactions in localized populations of model neurons. Biophys J 12(1):1–24

Womelsdorf T et al (2007) Modulation of neuronal interactions through neuronal synchronization. Science 316:1609–1612

Womelsdorf T et al (2012) Orientation selectivity and noise correlation in awake monkey area V1 are modulated by the gamma cycle. Proc Natl Acad Sci 109(11):4302–4307

Wu T et al (2022) Multi-band oscillations emerge from a simple spiking network. Chaos 33:043121

Xiao Z-C, Lin KK (2022) Multilevel monte Carlo for cortical circuit models. J Comput Neurosci 50(1):9–15

Xiao Z-C, Lin KK, Young L-S (2021) A data-informed mean-field approach to mapping of cortical parameter landscapes. PLoS Comput Biol 17(12):e1009718

Yuan X et al (2019) Adversarial examples: attacks and defenses for deep learning. IEEE Trans Neural Netw Learn Syst 30(9):2805–2824

Zhang J et al (2014) A coarse-grained framework for spiking neuronal networks: between homogeneity and synchrony. J Comput Neurosci 37(1):81–104

Zhang J et al (2014) Distribution of correlated spiking events in a population-based approach for integrate-and-fire networks. J Comput Neurosci 36:279–295

Zhang JW, Rangan AV (2015) A reduction for spiking integrate-and-fire network dynamics ranging from homogeneity to synchrony. J Comput Neurosci 38:355–404

Zhang Y, Young L-S (2020) DNN-assisted statistical analysis of a model of local cortical circuits. Sci Rep 10(1):1–16