TRAINING DEEP NORMALIZATION-FREE SPIKING NEURAL NETWORKS WITH LATERAL INHIBITION

Peiyu Liu[1,2], **Jianhao Ding**[1] **& Zhaofei Yu**[1,2,*]
[1] School of Computer Science, Peking University
[2] Institute for Artificial Intelligence, Peking University

ABSTRACT

Spiking neural networks (SNNs) have garnered significant attention as a central paradigm in neuromorphic computing, owing to their energy efficiency and biological plausibility. However, training deep SNNs has critically depended on explicit normalization schemes, leading to a trade-off between performance and biological realism. To resolve this conflict, we propose a normalization-free learning framework that incorporates lateral inhibition inspired by cortical circuits. Our framework replaces the traditional feedforward SNN layer with distinct excitatory (E) and inhibitory (I) neuronal populations that capture the key features of the cortical E-I interaction. The E-I circuit dynamically regulates neuronal activity through subtractive and divisive inhibition, which respectively control the excitability and gain of neurons. To stabilize end-to-end training of the biologically constrained SNNs, we propose two key techniques: E-I Init and E-I Prop. E-I Init is a dynamic parameter initialization scheme that balances excitatory and inhibitory inputs while performing gain control. E-I Prop decouples the backpropagation of the circuit from the forward pass, regulating gradient flow. Experiments across multiple datasets and network architectures demonstrate that our framework enables stable training of deep normalization-free SNNs with biological realism, achieving competitive performance. Therefore, our work not only provides a solution to training deep SNNs but also serves as a computational platform for further exploring the functions of E-I interaction in large-scale cortical computation. Code is available at `https://github.com/vwOvOwv/DeepEISNN`.

1 INTRODUCTION

Inspired by computational principles of biological neurons, SNNs stand at the intersection of artificial intelligence and neuroscience (Maass, 1997; Ghosh-Dastidar & Adeli, 2009). They not only enable highly energy-efficient computation on neuromorphic hardware (Roy et al., 2019; Xiao et al., 2025) but also provide models for understanding cortical computation across multiple scales (Kumarasinghe et al., 2021; Korcsak-Gorzo et al., 2022; N'dri et al., 2024). This duality places SNNs at the heart of the emerging field of NeuroAI (Sadeh & Clopath, 2025), fostering a synergy between artificial intelligence and neuroscience. While a deeper understanding of the brain's computational principles inspires novel SNN architectures (Fang et al., 2021b; Pan et al., 2025), advances in deep learning have concurrently enabled the training of large-scale, high-performance SNNs (Wu et al., 2018; Fang et al., 2021a; Bu et al., 2022). Some of these well-trained models, in turn, can serve as *in silico* platforms for investigating multi-scale cortical computation that is difficult to access through wet-lab experiments (Bellec et al., 2018).

Despite the promising synergy, realizing the potential of SNNs as an ideal platform for exploring both machine and biological intelligence is hindered by a trade-off between computational performance and biological plausibility. Many learning algorithms achieve high performance by adopting backpropagation-based techniques, treating spiking neurons as recurrent units unrolled over time (Neftci et al., 2019; Fang et al., 2021a). While this strategy yields models whose performance is comparable to their ANN counterparts, it reduces SNNs to mere deep learning artifacts, sacrificing

[*] Corresponding author: yuzf12@pku.edu.cn

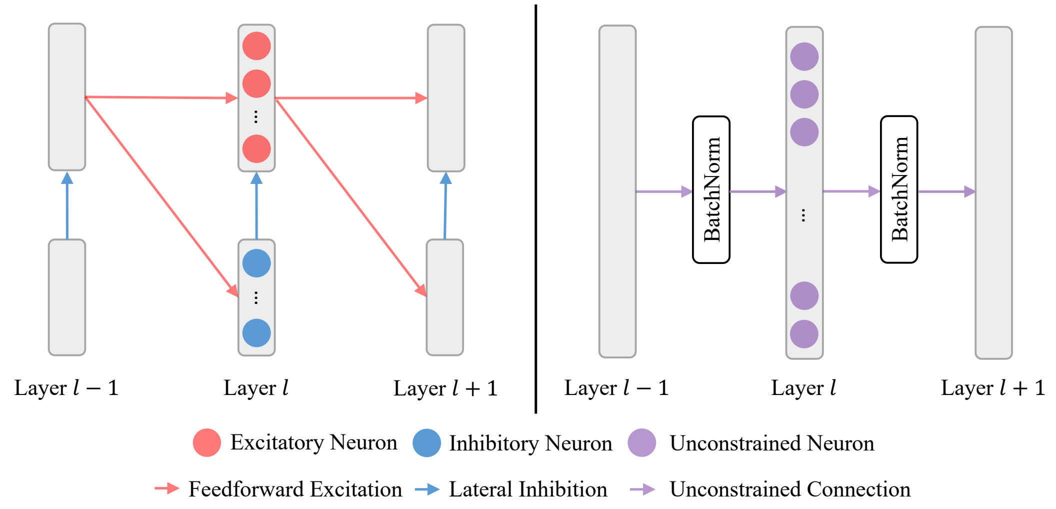


Figure 1: The proposed feedforward E-I circuit (**left**), compared with normalization-equipped architecture (**right**, BN as an example). Neurons in layer $l-1$ and $l+1$ are not shown.

basic biological properties. As a result, these models often ignore fundamental principles in neuroscience such as E-I dynamics, which are crucial in gain control (Goldwyn et al., 2018; Del Rosario et al., 2025), neural oscillation (Buzsáki & Draguhn, 2004), selective attention (Zhang et al., 2014), etc. Consequently, deriving meaningful insights for neuroscience from these biologically unfaithful models becomes a challenge.

However, approaches that prioritize biological realism also face the challenge of unstable training. Biologically plausible learning rules such as spike-timing-dependent plasticity (STDP) (Gerstner et al., 1996; Bi & Poo, 1998), often struggle with the instability during training and thus can only be applied to shallow SNNs (Habenschuss et al., 2012; Beyeler et al., 2013). In the field of deep learning, this training instability issue is partly overcome by explicit normalization schemes, most notably batch normalization (BN) (Ioffe & Szegedy, 2015). While such normalization schemes are powerful tools to accelerate and stabilize training, they explicitly collect statistics from inputs, which has no known biological analogue, making it implausible for brain-inspired models. This widens the gap between high performance and biological plausibility, highlighting the need for a biologically grounded alternative.

Here, we address the challenge of training deep biologically plausible SNNs by introducing lateral inhibition, a canonical interaction mechanism between excitatory and inhibitory neurons in cortex. We propose a normalization-free learning framework based on an E-I circuit, as shown in Figure 1. Our framework presents a brain-inspired alternative to standard normalization schemes, bridging the gap between high-performance deep learning and biologically plausible neural computation. The main contributions of our work are summarized as follows:

1. We incorporate a canonical E-I circuit, composed of distinct excitatory and inhibitory neuron populations, into deep SNNs to enable normalization-free training.
2. We introduce a dynamic initialization scheme to ensure effective learning from the very beginning of training.
3. We integrate adaptive stabilization of divisive inhibition and straight-through estimator (STE) into the framework. These techniques prove to be essential for stable learning of deep SNNs with the E-I circuit.
4. Experiments demonstrate that the framework achieves competitive performance across different datasets and architectures, indicating the viability of our brain-inspired learning algorithm.

2 Related Work

2.1 Normalization in SNNs

Recent methods for training deep SNNs can be broadly categorized into two approaches, ANN-to-SNN conversion (ANN2SNN) (Rueckauer et al., 2017; Sengupta et al., 2019; Han et al., 2020; Han & Roy, 2020; Ding et al., 2021; Stöckl & Maass, 2021; Bu et al., 2022; Zhao et al., 2025) and direct end-to-end training (Lee et al., 2016; Neftci et al., 2019; Li et al., 2021; Fang et al., 2021a;b; Guo et al., 2022b; Xiao et al., 2022; Zhu et al., 2024). Normalization is critical in both methods. Many ANN2SNN methods merge normalization parameters of ANNs into synaptic weights of spiking neurons (Sengupta et al., 2019; Han et al., 2020; Bu et al., 2022). In contrast, training SNNs from scratch usually directly adopts normalization schemes developed for ANNs, especially BN (Ioffe & Szegedy, 2015). There are also BN-derived normalization schemes designed for SNNs, like NeuNorm (Wu et al., 2019), BNTT (Kim & Panda, 2021), tdBN (Zheng et al., 2021), TEBN (Duan et al., 2022), and TAB (Jiang et al., 2024). However, these strategies still inherit the biological implausibility of normalization schemes due to their dependence on statistics collected from batches of inputs throughout the training. Therefore, the need for fully brain-inspired normalization alternatives remains.

2.2 Neural networks with separate excitatory and inhibitory units

The interaction between excitatory and inhibitory neurons has been a key topic in neuroscience (Haider et al., 2006; Ahmadian & Miller, 2021; Cohen Kadosh, 2025). Historically, computational models of these circuits have been confined to shallow networks, often focusing on the dynamics of a few interacting populations/neurons to explain basic principles (Somers et al., 1995; Wilson & Cowan, 1972; Carandini & Heeger, 2012). One of the reasons for this limitation is that training deep networks with the E-I circuit proves to be a significant challenge. This has left the whole picture of E-I dynamics largely unexplored. A remarkable step towards deep E-I networks was taken by Cornford et al. (2021). By developing techniques for parameter initialization, they demonstrated that ANNs with separate excitatory and inhibitory neuron populations could be effectively trained. However, the model is built fully with the rectified linear units (ReLU) and thus does not capture the temporal properties of the circuit. While SNNs can address temporal processing, such initialization techniques cannot be directly applied to SNNs with the E-I circuit.

3 Preliminaries

3.1 Excitation and inhibition in cortex

A fundamental principle in the cortex is the functional segregation of neurons into distinct excitatory and inhibitory populations (Barranca et al., 2022). This circuit-level architecture implies that a given neuron typically exerts a uniform influence (either depolarization or hyperpolarization) on all its postsynaptic targets. When translating this principle to artificial neural networks, all outgoing synaptic weights from a given neuron should share the same sign. This makes standard initialization techniques like Xavier (Glorot & Bengio, 2010) and Kaiming (He et al., 2015) initialization inapplicable, as they sample weights from zero-centered distributions that assign both positive and negative weights to each neuron.

3.2 Neuron models

Excitatory neurons. We model excitatory neurons with the widely adopted leaky-integrate-and-fire (LIF) model (Gerstner et al., 2014). For a given layer l with $n_{\mathrm{E}}^{[l]}$ excitatory neurons (we use superscript $[l]$ to denote the l-th layer), the sub-threshold dynamics of the membrane potential $\mathbf{u}_{\mathrm{E}}^{[l]}(t) \in \mathbb{R}^{n_{\mathrm{E}}^{[l]}}$ are described by the following equation:

$$\tau_{\mathrm{E}} \frac{\mathrm{d}\mathbf{u}_{\mathrm{E}}^{[l]}(t)}{\mathrm{d}t} = -\left(\mathbf{u}_{\mathrm{E}}^{[l]}(t) - u_{\mathrm{E,rest}}\right) + \mathbf{I}_{\mathrm{E}}^{[l]}(t), \tag{1}$$

where τ_{E} and $u_{\mathrm{E,rest}}$ are the membrane time constant and resting potential of all excitatory neurons, respectively. $\mathbf{I}_{\mathrm{E}}^{[l]}(t) \in \mathbb{R}^{n_{\mathrm{E}}^{[l]}}$ represents the input currents to the excitatory neurons at time t, assuming a unit membrane resistance. For discrete-time simulation, we approximate Equation 1 using the first-order Euler method with a time step $\Delta t = 1$. By setting $u_{\mathrm{E,rest}}$ to 0 and omitting decay of input currents, we obtain the following iterative update rule:

$$\mathbf{u}_{\mathrm{E}}^{[l]}[t+1] = \left(1 - \frac{1}{\tau_{\mathrm{E}}}\right) \mathbf{u}_{\mathrm{E}}^{[l]}[t] + \mathbf{I}_{\mathrm{E}}^{[l]}[t]. \tag{2}$$

An excitatory neuron emits a spike when its membrane potential exceeds a firing threshold θ_{E} (which is set to 1 for all excitatory neurons). To model the subsequent reset, we employ a soft reset mechanism where the potential of a firing neuron is reduced by θ_{E}. This leads to full dynamics for excitatory neurons in layer l,

$$\mathbf{u}_{\mathrm{E}}^{[l]}[t+1] = \left(1 - \frac{1}{\tau_{\mathrm{E}}}\right) \left(\mathbf{u}_{\mathrm{E}}^{[l]}[t] - \theta_{\mathrm{E}} \cdot \mathbf{s}_{\mathrm{E}}^{[l]}[t]\right) + \mathbf{I}_{\mathrm{E}}^{[l]}[t], \tag{3}$$

where the spikes are generated by the Heaviside step function H, i.e., $\mathbf{s}_{\mathrm{E}}^{[l]}[t] = H\left(\mathbf{u}_{\mathrm{E}}^{[l]}[t] - \theta_{\mathrm{E}}\right)$.

To distinguish this process from the dynamics of inhibitory neurons in layer l, we encapsulate it into an operator $\mathcal{F}_{\mathrm{E}}^{[l]}$. This operator takes the input currents at the current time step as its arguments and produces the corresponding output spikes.

$$\mathbf{s}_{\mathrm{E}}^{[l]}[t] = \mathcal{F}_{\mathrm{E}}^{[l]}\left(\mathbf{I}_{\mathrm{E}}^{[l]}[t]; \mathbf{u}_{\mathrm{E}}^{[l]}[t], \tau_{\mathrm{E}}, \theta_{\mathrm{E}}\right). \tag{4}$$

Inhibitory neurons. Many inhibitory neurons, such as parvalbumin (PV$^+$) neurons, are known to be fast-spiking (FS), characterized by a much smaller membrane time constant τ_{I} compared to that of excitatory pyramidal neurons (Hu et al., 2014; Prince et al., 2021). In our discrete-time simulation, the time step $\Delta t = 1$ is chosen to be on a similar scale as the time constant of excitatory neurons (e.g., $\tau_{\mathrm{E}} = 2$), which implies $\tau_{\mathrm{I}} \ll \Delta t$. Under this condition, the dynamics of inhibitory neurons can reach a steady state almost instantaneously within a single time step. This allows us to apply an approximation to LIF model by treating τ_{I} as negligible, which leads to

$$0 = -\left(\mathbf{u}_{\mathrm{I}}^{[l]}[t] - u_{\mathrm{I,rest}}\right) + \mathbf{I}_{\mathrm{I}}^{[l]}[t]. \tag{5}$$

Here we use notations similar to those in the excitatory neuron model, where the subscript I denotes inhibitory neurons. By setting $u_{\mathrm{I,rest}}$ to 0, we find the membrane potential of an inhibitory neuron is determined purely by its input currents at time t:

$$\mathbf{u}_{\mathrm{I}}^{[l]}[t] = \mathbf{I}_{\mathrm{I}}^{[l]}[t]. \tag{6}$$

Since this potential remains constant throughout the duration of Δt, the neurons can fire $\left\lfloor \max\left(0, \mathbf{I}_{\mathrm{I}}^{[l]}[t]\right) / \theta_{\mathrm{I}} \right\rfloor$ times if we apply soft reset and a fixed firing threshold θ_{I}. Finally, by setting $\theta_{\mathrm{I}} = 1$, we can directly model the total spike outputs of inhibitory neurons at time t as

$$\mathbf{s}_{\mathrm{I}}^{[l]}[t] = \left\lfloor \max\left(0, \mathbf{I}_{\mathrm{I}}^{[l]}[t]\right) \right\rfloor \approx \max\left(0, \mathbf{I}_{\mathrm{I}}^{[l]}[t]\right). \tag{7}$$

Similar to the excitatory neuron model, we encapsulate this process into an operator $\mathcal{F}_{\mathrm{I}}^{[l]}$:

$$\mathbf{s}_{\mathrm{I}}^{[l]}[t] = \mathcal{F}_{\mathrm{I}}^{[l]}\left(\mathbf{I}_{\mathrm{I}}^{[l]}[t]\right) \approx \max\left(0, \mathbf{I}_{\mathrm{I}}^{[l]}[t]\right). \tag{8}$$

See Appendix D.1 for a detailed derivation.

4 METHODS

4.1 E-I CIRCUIT IN SNNS

The model is constructed according to the canonical E-I circuit shown in Figure 1. Each layer l comprises $n_{\mathrm{E}}^{[l]}$ excitatory neurons and $n_{\mathrm{I}}^{[l]}$ inhibitory neurons, where $n_{\mathrm{E}}^{[l]}/n_{\mathrm{I}}^{[l]} = 4$ according to biological evidence (Markram et al., 2004). Computation conducted by this circuit at each time step t is demonstrated in Figure 2.

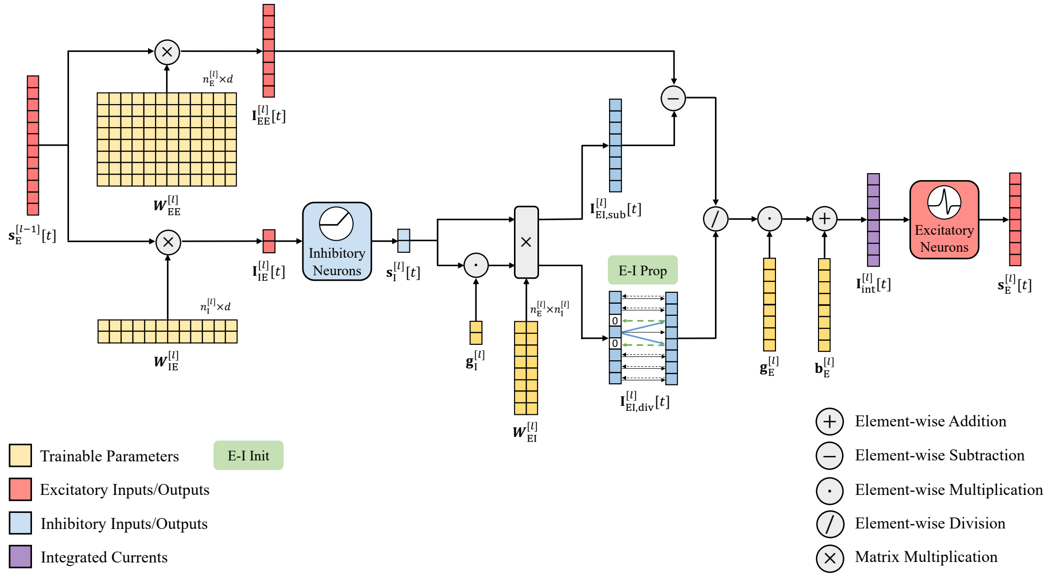


Figure 2: An overview of the proposed framework. E-I Init enables effective learning from the very beginning through a dynamic parameter initialization scheme. E-I Prop then ensures stable end-to-end training by regulating the forward and backward passes. For the sake of brevity, our discussion in the main text focuses on a fully connected architecture. Extension of our method to convolutional neural networks (CNNs) is detailed in Appendix E.1.

Excitatory projections. First, excitatory population in layer $l-1$ undergoes dynamics described by Equation 1 to Equation 4 and emits spikes $\mathbf{s}_{\mathrm{E}}^{[l-1]}[t] \in \{0,1\}^d$, where d is the dimension of input spikes. These input spikes project forward, inducing two excitatory currents into both the excitatory and inhibitory populations of layer l:

$$\mathbf{I}_{\mathrm{EE}}^{[l]}[t] = \boldsymbol{W}_{\mathrm{EE}}^{[l]} \mathbf{s}_{\mathrm{E}}^{[l-1]}[t], \tag{9}$$

$$\mathbf{I}_{\mathrm{IE}}^{[l]}[t] = \boldsymbol{W}_{\mathrm{IE}}^{[l]} \mathbf{s}_{\mathrm{E}}^{[l-1]}[t]. \tag{10}$$

Here, subscript AB denotes projections from population B to population A. $\boldsymbol{W}_{\mathrm{EE}}^{[l]} \in \mathbb{R}^{n_{\mathrm{E}}^{[l]} \times d}$ and $\boldsymbol{W}_{\mathrm{IE}}^{[l]} \in \mathbb{R}^{n_{\mathrm{I}}^{[l]} \times d}$ are corresponding synaptic weight matrices, which are constrained to be non-negative during training due to E-I segregation.

Lateral inhibition. Following the fast-spiking approximation in Section 3.2, the activity of inhibitory neurons is modeled by $\mathcal{F}_{\mathrm{I}}^{[l]}$,

$$\mathbf{s}_{\mathrm{I}}^{[l]}[t] = \mathcal{F}_{\mathrm{I}}^{[l]}\left(\mathbf{I}_{\mathrm{IE}}^{[l]}[t]\right). \tag{11}$$

This inhibitory signal then laterally regulates the excitatory population. Motivated by the biophysical distinction between dendritic and somatic inhibition, we decompose this regulation into subtractive inhibition for E-I balance and divisive inhibition for gain control.

$$\mathbf{I}_{\mathrm{EI,sub}}^{[l]}[t] = \boldsymbol{W}_{\mathrm{EI}}^{[l]} \mathbf{s}_{\mathrm{I}}^{[l]}[t], \tag{12}$$

$$\mathbf{I}_{\mathrm{EI,div}}^{[l]}[t] = \boldsymbol{W}_{\mathrm{EI}}^{[l]} \left(\mathbf{g}_{\mathrm{I}}^{[l]} \odot \mathbf{s}_{\mathrm{I}}^{[l]}[t]\right), \tag{13}$$

where $\boldsymbol{W}_{\mathrm{EI}}^{[l]} \in \mathbb{R}^{n_{\mathrm{E}}^{[l]} \times n_{\mathrm{I}}^{[l]}}$ is the weight matrix for inhibitory-to-excitatory projections, $\mathbf{g}_{\mathrm{I}}^{[l]} \in \mathbb{R}^{n_{\mathrm{I}}^{[l]}}$ is a trainable parameter modulating the strength of divisive inhibition, and $\odot$ denotes the Hadamard product.

Input integration and spiking. Finally, the excitatory population integrates the excitatory currents with both forms of inhibition to compute the total input currents.

$$\mathbf{I}_{\mathrm{int}}^{[l]}[t] = \mathbf{g}_{\mathrm{E}}^{[l]} \odot \frac{\mathbf{I}_{\mathrm{EE}}^{[l]}[t] - \mathbf{I}_{\mathrm{EI,sub}}^{[l]}[t]}{\mathbf{I}_{\mathrm{EI,div}}^{[l]}[t]} + \mathbf{b}_{\mathrm{E}}^{[l]}, \tag{14}$$

where $\mathbf{g}_{\mathrm{E}}^{[l]}, \mathbf{b}_{\mathrm{E}}^{[l]} \in \mathbb{R}^{n_{\mathrm{E}}^{[l]}}$ are trainable parameters, and the division is performed element-wise. Taking the integrated currents as input currents, excitatory neurons emit spikes to the next layer,

$$\mathbf{s}_{\mathrm{E}}^{[l]}[t] = \mathcal{F}_{\mathrm{E}}^{[l]}\left(\mathbf{I}_{\mathrm{int}}^{[l]}[t]; \mathbf{u}_{\mathrm{E}}^{[l]}[t], \tau_{\mathrm{E}}, \theta_{\mathrm{E}}\right). \tag{15}$$

4.2 E-I Init: dynamic parameter initialization

As discussed in Section 3.1, standard zero-centered initialization schemes like Xavier (Glorot & Bengio, 2010) and Kaiming (He et al., 2015) are incompatible with the strict sign constraints of E-I segregation. Some naive initializations under this constraint lead to pathological network activity, making the training of deep architectures infeasible (see Section 5.2). Therefore, we propose E-I Init, an initialization scheme designed for deep SNNs with the E-I circuit. Its design is guided by two primary objectives: (1) establishing an initial E-I balance to prevent neurons from silencing or saturating, and (2) setting an appropriate initial gain to ensure stable signal propagation.

E-I balance via subtractive inhibition. A key goal of our initialization scheme is to ensure that neurons operate in a responsive regime. We achieve this by setting the expected subtractive inhibitory currents to approximately balance the expected excitatory currents, which is defined as

$$\mathbb{E}\left[\mathbf{I}_{\mathrm{EE},i}^{[l]}\right] \approx \mathbb{E}\left[\mathbf{I}_{\mathrm{EI,sub},i}^{[l]}\right], \tag{16}$$

for each neuron i in layer l and results in a near-zero expected net input, preventing neurons from being saturated or silent at initialization. To implement this under the constraint of E-I segregation, we draw inspiration from Cornford et al. (2021) and leverage the exponential distribution for weight initialization. Specifically, we draw the excitatory weights $\mathrm{W}_{\mathrm{EE}}^{[l]}$ and $\mathrm{W}_{\mathrm{IE}}^{[l]}$ from an exponential distribution with rate parameter $\lambda^{[l]}$. The inhibitory weights $\mathrm{W}_{\mathrm{EI}}^{[l]}$ are deterministically set to $1/n_{\mathrm{I}}^{[l]}$ to uniformly distribute the inhibitory signals. Here $\mathrm{W}_{\mathrm{AB}}^{[l]} \in \mathbb{R}$ denotes elements of $\boldsymbol{W}_{\mathrm{AB}}^{[l]}$. Assuming that presynaptic neurons fire independently with an average probability of p (i.e., the spike from each neuron at any time step is an independent and identically distributed (i.i.d.) Bernoulli trial with probability p), the expected excitatory input to neuron i in layer l is

$$\mathbb{E}\left[\mathbf{I}_{\mathrm{EE},i}^{[l]}\right] = dp\mathbb{E}\left[\mathrm{W}_{\mathrm{EE}}^{[l]}\right], \tag{17}$$

where d is the dimension of input spikes. Similarly, the expected subtractive inhibitory currents are

$$\mathbb{E}\left[\mathbf{I}_{\mathrm{EI,sub},i}^{[l]}\right] = n_{\mathrm{I}}^{[l]} dp\mathbb{E}\left[\mathrm{W}_{\mathrm{IE}}^{[l]}\right]\mathbb{E}\left[\mathrm{W}_{\mathrm{EI}}^{[l]}\right]. \tag{18}$$

Therefore, by setting $\mathbb{E}\left[\mathrm{W}_{\mathrm{EE}}^{[l]}\right] = \mathbb{E}\left[\mathrm{W}_{\mathrm{IE}}^{[l]}\right] = 1/\lambda^{[l]}$ and $\mathrm{W}_{\mathrm{EI}}^{[l]} = 1/n_{\mathrm{I}}^{[l]}$, we arrive at the desired balance defined by Equation 16 (see Appendix D.2 for a detailed derivation).

Gain control via divisive inhibition. Our second objective is establishing stable signal propagation by setting an appropriate initial gain for excitatory neurons. In our model, gain is primarily modulated by the divisive inhibitory currents. Inspired by normalization techniques, our strategy is modulating divisive inhibition through $\mathbf{g}_{\mathrm{I}}^{[l]}$ such that the expected value of this divisive inhibitory currents approximates the standard deviation of the excitatory inputs for each neuron i. This can be formulated as

$$\mathbb{E}\left[\mathbf{I}_{\mathrm{EI,div},i}^{[l]}\right] = \mathrm{std}\left(\mathbf{I}_{\mathrm{EE},i}^{[l]}\right) \tag{19}$$

for each neuron i in layer l. In this way, the divisive operation effectively scales the final input, leading the excitatory neurons to a responsive regime at initialization. Similar to Equation 18, for inhibitory neuron i in layer l,

$$\mathbb{E}\left[\mathbf{I}_{\mathrm{EI,div},i}^{[l]}\right] = n_{\mathrm{I}}^{[l]} dp\mathbb{E}\left[\mathbf{g}_{\mathrm{I}}^{[l]}\right]\mathbb{E}\left[\mathrm{W}_{\mathrm{IE}}^{[l]}\right]\mathbb{E}\left[\mathrm{W}_{\mathrm{EI}}^{[l]}\right] = \frac{dp\mathbb{E}\left[\mathbf{g}_{\mathrm{I}}^{[l]}\right]}{\lambda^{[l]}}. \tag{20}$$

By assuming that the spike from each neuron at any time step is i.i.d. Bernoulli trial with probability p, the standard deviation of excitatory input currents of neuron i can be formulated as

$$\operatorname{std}\left(\mathbf{I}_{\mathrm{EE},i}^{[l]}\right) = \frac{\sqrt{dp(2-p)}}{\lambda^{[l]}}, \tag{21}$$

see Appendix D.3 for details. Therefore, by setting each element of $\mathbf{g}_{\mathrm{I}}^{[l]}$ to $\sqrt{\frac{2-p}{dp}}$, we achieve a normalization effect at initialization, making the effective training of deep SNNs possible.

Initialization of other parameters. Finally, we initialize $\lambda^{[l]} = \sqrt{\frac{d(2-p)}{1-p}}$ under the condition $\operatorname{std}\left(\mathbf{I}_{\mathrm{EE},i}^{[l]}\right) = \sqrt{p\,(1-p)}$ (see Appendix D.4). $\mathbf{g}_{\mathrm{E}}^{[l]}$ and $\mathbf{b}_{\mathrm{E}}^{[l]}$ are initialized as vector $\mathbf{1}$ and $\mathbf{0}$, respectively.

Dynamic firing probability estimation. Since the initialization depends on the averaged firing probability p, we use the first batch in training set to compute point estimations of p and other statistics, leading to a dynamic initialization regime (see Appendix E.2 for implementation details).

4.3 E-I Prop: stabilizing end-to-end training

While E-I Init provides a stable initial state, the interplay between divisive inhibition and discrete input spikes induces training instabilities. These instabilities arise mainly from two sources: first, near-zero divisive currents in the forward pass trigger numerical explosions; and second, disproportionately large gradients in the lateral inhibitory pathway destabilize training. To overcome these issues, we propose E-I Prop, a toolkit that decouples the backpropagation of the E-I circuit from the forward pass, regulating gradient flow. Forward stability is ensured by adaptive divisive inhibition, while backward stability is achieved through a straight-through estimator (STE) combined with gradient scaling.

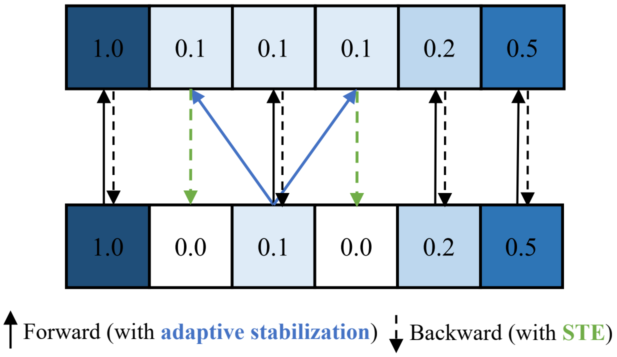


Figure 3: Mechanism of adaptive stabilization and STE. **Forward (bottom-up):** The adaptive stabilization handles numerical instability by dynamically replacing zero elements in the denominator with the smallest positive value in the sample, preserving maximal dynamic range. **Backward (top-down):** The STE allows gradients to bypass the replacement operation, treating it as an identity function.

Adaptive stabilization of divisive inhibition. To prevent division-by-zero error in the forward pass, a common technique is adding a small constant ϵ to the denominator. However, a fixed ϵ is ill-suited for our network because the divisive inhibitory currents are designed to provide a suitable dynamic range and perform gain control. A pre-defined ϵ that is too small may fail to prevent numerical instability if the denominator collapses towards zero, while one that is too large will artificially suppress the dynamic range by dominating the denominator. As shown in Figure 3, here we propose an adaptive stabilization method. Instead of using a static constant, our approach sets a dynamic, input-dependent lower bound for the denominator. Specifically, for each sample within a batch, we identify any zero values in $\mathbf{I}_{\mathrm{EI,div}}$. Then, these zero values are replaced by the smallest positive value found within the same sample, which proves to be necessary for effective training (see Section 5.2 for details).

Straight-through estimator (STE). Since the replacement operation in our adaptive stabilization is non-differentiable, it misdirects the gradient flow and destabilizes learning. To address this, we employ STE, a common technique for handling non-differentiable operations in neural networks (Bengio et al., 2013). In the forward pass, we perform the adaptive stabilization as described above. In the backward pass, we approximate the derivative of this non-differentiable operation with an identity function (see Appendix E.3 for implementation). This approach decouples the forward-pass requirement for numerical stability from the backward-pass requirement for a clean gradient path, ensuring that the network can learn robustly.

Gradient scaling. To ensure stable training, it is crucial to balance the influence of the feedforward excitatory and lateral inhibitory pathways on parameter updates. Both theoretical and empirical analyses reveal that gradients for the lateral inhibitory weight, $\boldsymbol{W}_{\mathrm{EI}}^{[l]}$, are disproportionately larger than those for other synaptic weights (see Appendix D.5 and F.1). To counteract this gradient amplification, we scale the gradients of $\boldsymbol{W}_{\mathrm{EI}}^{[l]}$ by a factor of $1/d$, where d is the dimension of the input spikes. This effectively balances the update magnitudes between the two pathways.

As illustrated in Figure 2, our method provides a completely normalization-free learning framework composed of E-I Init and E-I Prop, enabling stable and effective end-to-end training from scratch.

5 Experiments

5.1 Performance on classification tasks

We evaluate our framework on both static and event-based datasets, with results summarized in Table 1. Our ResNet-18 model (He et al., 2016; Fang et al., 2021a) achieves 92.05±0.11% top-1 accuracy on CIFAR-10 (Krizhevsky, 2009), surpassing all normalization-free baselines. Notably, on the more challenging TinyImageNet (Le & Yang, 2015), the model attains 50.29% accuracy, demonstrating scalability to large-scale datasets. Furthermore, to validate temporal processing capabilities, we test on neuromorphic datasets where our method achieves 94.86±0.86% on DVS-Gesture (Amir et al., 2017) and 77.66±0.48% on CIFAR10-DVS (Li et al., 2017), outperforming several BN-based methods. Collectively, these results across diverse datasets confirm that our E-I circuit, empowered by E-I Init and E-I Prop, serves as a robust and scalable alternative to explicit normalization in deep SNNs.

5.2 Ablation study

Results of ablation study are summarized in Table 2. It confirms that each component in our method is indispensable for stable and high-performance training. Replacing E-I constraint (on sign of weights) or E-I Init by applying a standard/clamped Kaiming initialization (He et al., 2015) leads to either a complete training failure or a significant accuracy drop, proving its necessity for establishing a proper E-I balance and gain control through E-I segregation and E-I init. Furthermore, the common ϵ-stabilization fails across all tested values, confirming that our adaptive stabilization mechanism is crucial for maintaining both numerical stability and gain control. Finally, removing gradient scaling on $\boldsymbol{W}_{\mathrm{EI}}$ causes training collapse, which validates its role in stabilizing learning dynamics. These findings demonstrate that E-I constraint, E-I Init, adaptive stabilization of divisive inhibition, and gradient scaling are necessary and coupled mechanisms for effective training of deep E-I SNNs.

5.3 Functional implications of the E-I circuit

Visualization of the integrated currents distribution reveals that our framework leverages a normalization-like effect at initialization but ultimately learns a more sophisticated representation. As shown in Figure 4, E-I Init successfully produces stable, zero-centered Gaussian-like distributions at the beginning of training. However, after training, some of the distributions evolve into a distinct bimodal shape, in contrast to the Gaussian-like outputs of SNNs with vanilla BN (see Appendix F.3).

This emergent bimodality can be considered as a mixture of two Gaussian-like distributions, with one distribution centered at the negative regime, as shown in Figure 4. This indicates that the E-I circuit is not equivalent to simple normalization, but rather a dynamic separation of neurons into

Table 1: Comparison with other E-I constrained, normalization-free, and BN-equipped methods.

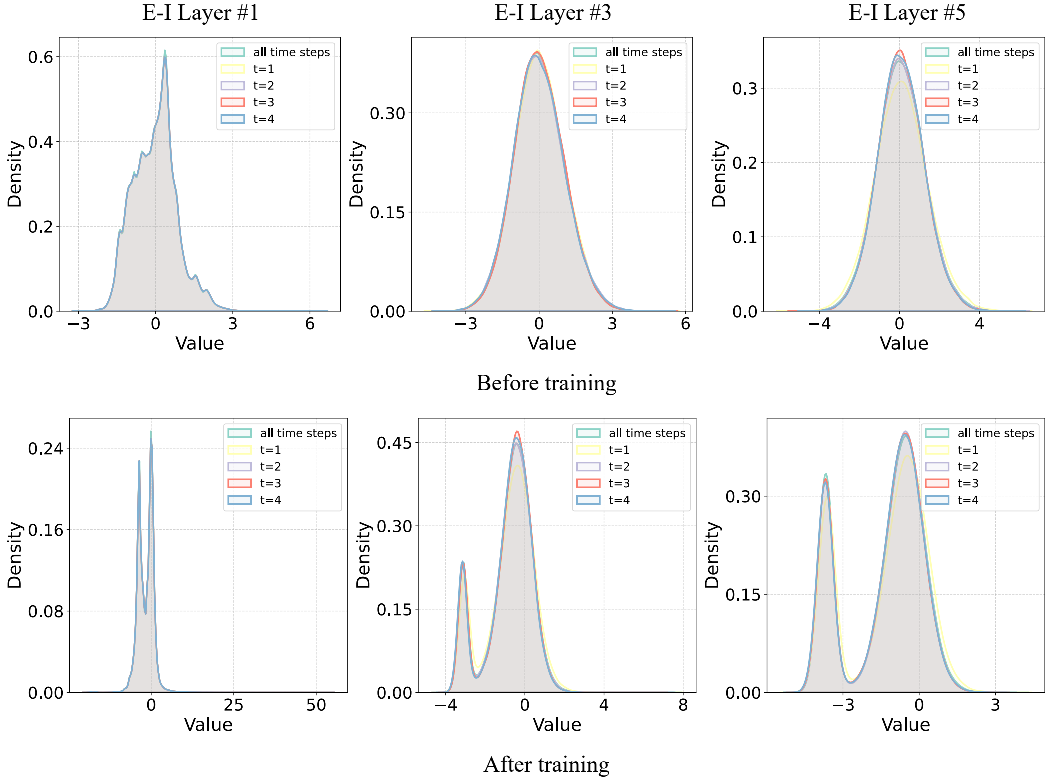
Dataset	E-I	Normalization	Method	Architecture	Time steps	Top-1 Accuracy (%)
CIFAR-10	×	×	DAP (Micheli et al., 2025)	7-layer CNN	3	57.52
			SRI (Ding et al., 2025)	VGG-9	20	87.62
			B-SNN (Karimah et al., 2025)	VGG-8	64	87.73
			EICIL (Shao et al., 2023)	ResNet-18	–	90.34
			IM-Loss (Guo et al., 2022a)	CIFARNet	4	90.90
	×	✓	*Vanilla BN	VGG-16	4	94.29 ± 0.07
				VGG-19	4	94.11 ± 0.15
				ResNet-18	4	95.37 ± 0.13
			BNTT (Kim & Panda, 2021)	VGG-9	20	90.30
			NeuNorm (Wu et al., 2019)	CIFARNet	12	90.53
			TEBN (Duan et al., 2022)	VGG-9	4	92.81
				ResNet-19	4	94.70
			tdBN (Zheng et al., 2021)	ResNet-19	4	92.92
			TAB (Jiang et al., 2024)	VGG-9	4	93.41
				ResNet-19	4	94.76
	✓	×	FDI (Rossbroich et al., 2022)	6-layer CNN	50	65.60
			*DANN (Cornford et al., 2021)	VGG-16	–	88.54 ± 0.38
				VGG-19	–	88.28 ± 0.27
			EICIL (Shao et al., 2023)	E-I Net	–	89.43
			BackEISNN (Zhao et al., 2022)	5-layer CNN	20	90.93
			DeepEISNN (Ours)	**VGG-16**	**4**	$\mathbf{90.80 \pm 0.16}$
				VGG-19	**4**	$\mathbf{90.93 \pm 0.33}$
				ResNet-18	**4**	$\mathbf{92.05 \pm 0.11}$
CIFAR-100	×	×	SRI (Ding et al., 2025)	VGG-11	20	54.94
			EICIL (Shao et al., 2023)	ResNet-18	–	63.47
	×	✓	BNTT (Kim & Panda, 2021)	VGG-11	50	66.60
			TEBN (Duan et al., 2022)	VGG-11	4	74.37
			TAB (Jiang et al., 2024)	VGG-11	4	75.89
	✓	×	EICIL (Shao et al., 2023)	E-I Net	–	53.86
			DeepEISNN (Ours)	**VGG-16**	**4**	$\mathbf{64.95 \pm 1.14}$
				VGG-19	**4**	$\mathbf{64.31 \pm 0.41}$
CIFAR10-DVS	×	×	IM-Loss (Guo et al., 2022a)	ResNet-19	10	72.60
	×	✓	NeuNorm (Wu et al., 2019)	7-layer CNN	40	60.50
			BNTT (Kim & Panda, 2021)	7-layer CNN	20	63.20
			tdBN (Zheng et al., 2021)	ResNet-19	10	67.80
			TEBN (Duan et al., 2022)	7-layer CNN	10	75.10
			TAB (Jiang et al., 2024)	7-layer CNN	4	76.70
	✓	×	**DeepEISNN (Ours)**	**VGG-8**	**10**	$\mathbf{77.30 \pm 0.64}$
				VGG-11	**10**	$\mathbf{77.66 \pm 0.48}$
DVS-Gesture	✓	×	FDI (Rossbroich et al., 2022)	6-layer CNN	500	86.70
			DeepEISNN (Ours)	**VGG-8**	**16**	$\mathbf{94.86 \pm 0.86}$
TinyImageNet	✓	×	**DeepEISNN (Ours)**	**ResNet-18**	**4**	**50.29**

* Results marked with an asterisk are reproduced (averaged over 5 independent runs). Other results are cited from existing literature.

Table 2: Ablation study on individual components of the method on CIFAR-10 with VGG-8.

Ablation Setting					
E-I Cons.	E-I Init	Adap. Stab.	Grad. Scale	Ablation Details	Top-1 Accuracy (%)
✓	✓	✓	✓	**Proposed Method (Ours)**	**87.03**
✓	×	✓	✓	Kaiming Init on $\boldsymbol{W}_{\mathrm{EE}}$ and $\boldsymbol{W}_{\mathrm{IE}}$ (clamped to $[0, +\infty)$)	85.61
✓	✓	×	✓	Fixed $\epsilon = 10^{-8}$	Failed to converge
				Fixed $\epsilon = 10^{-7}$	Failed to converge
				Fixed $\epsilon = 10^{-6}$	Collapsed (Epoch 22)
				Fixed $\epsilon = 10^{-5}$	Collapsed (Epoch 5)
✓	✓	✓	×	No scaling on $\boldsymbol{W}_{\mathrm{EI}}$	Collapsed (Epoch 10)
×	×	✓	✓	Kaiming Init on $\boldsymbol{W}_{\mathrm{EE}}$ and $\boldsymbol{W}_{\mathrm{IE}}$ (no sign constraint)	Failed to converge

activated and suppressed populations, confirming its role as a strategy distinct from standard normalization.



Before training

After training

Figure 4: Distributions of the integrated input currents in the first, third and fifth layers of our model before and after training.

6 Conclusion and Discussion

In this work, we address the critical challenge in training deep normalization-free SNNs by introducing biologically inspired E-I segregation and lateral inhibition. Through a fine-grained initialization scheme E-I Init, and a toolkit of stabilization techniques E-I Prop, we enable the stable end-to-end training of deep normalization-free SNNs that capture features of canonical E-I circuit in the cortex. Our experiments demonstrate that the framework not only achieves competitive performance on multiple datasets but also learns a sophisticated mechanism of activity regulation that is functionally distinct from standard normalization. We show that the E-I circuit starts at an initial normalization-like state, and ultimately learns an activity regulation mechanism distinct from explicit normalizations like BN. Therefore, our work provides both a practical solution for building powerful, normalization-free SNNs and a compelling computational model for exploring how canonical cortical circuits perform complex and large-scale computation, further bridging the gap between deep learning and neuroscience.

While our framework successfully eliminates the need for normalization during training and inference, the deployment of the E-I circuit on digital neuromorphic hardware would require approximation techniques, such as bit-shifts or look-up tables. While there are obstacles for digital chips, emerging analog or mixed-signal platforms like DYNAP-SE2 (Richter et al., 2024) have already supported such operations. Therefore, our framework offers direct algorithmic compatibility with next-generation analog or mixed-signal neuromorphic hardware.

Acknowledgments

This work is supported by STI 2030-Major Projects 2025ZD0217101, the National Natural Science Foundation of China (62422601, U24B20140), Beijing Municipal Science and Technology Program (Z241100004224004), Beijing Nova Program (20230484362, 20240484703), and National Key Laboratory for Multimedia Information Processing.

References

Yashar Ahmadian and Kenneth D. Miller. What is the dynamical regime of cerebral cortex? *Neuron*, 109(21):3373–3391, 2021.

Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, Jeff Kusnitz, Michael Debole, Steve Esser, Tobi Delbruck, Myron Flickner, and Dharmendra Modha. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7388–7397, 2017.

Victor J. Barranca, Asha Bhuiyan, Max Sundgren, and Fangzhou Xing. Functional implications of Dale's law in balanced neuronal network dynamics and decision making. *Frontiers in Neuroscience*, 16:801847, 2022.

Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. In *Advances in Neural Information Processing Systems*, pp. 795–805, 2018.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Michael Beyeler, Nikil D. Dutt, and Jeffrey L. Krichmar. Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule. *Neural Network*, 48:109–124, 2013.

Guo-qiang Bi and Mu-ming Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, 18 (24):10464–10472, 1998.

Tong Bu, Wei Fang, Jianhao Ding, PengLin Dai, Zhaofei Yu, and Tiejun Huang. Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks. In *International Conference on Learning Representations*, 2022.

György Buzsáki and Andreas Draguhn. Neuronal oscillations in cortical networks. *Science*, 304: 1926–1929, 2004.

Matteo Carandini and David J. Heeger. Normalization as a canonical neural computation. *Nature Reviews Neuroscience*, 13:51–62, 2012.

Roi Cohen Kadosh. Rethinking excitation/inhibition balance in the human brain. *Nature Reviews Neuroscience*, 26(8):451–452, 2025.

Jonathan Cornford, Damjan Kalajdzievski, Marco Leite, Amélie Lamarquette, Dimitri Michael Kullmann, and Blake Aaron Richards. Learning to live with Dale's principle: ANNs with separate excitatory and inhibitory units. In *International Conference on Learning Representations*, 2021.

Joseph Del Rosario, Stefano Coletta, Soon Ho Kim, Zach Mobille, Kayla Peelman, Brice Williams, Alan J. Otsuki, Alejandra Del Castillo Valerio, Kendell Worden, Lou T. Blanpain, Lyndah Lovell, Hannah Choi, and Bilal Haider. Lateral inhibition in v1 controls neural and perceptual contrast sensitivity. *Nature Neuroscience*, 28:836–847, 2025.

Jianhao Ding, Zhaofei Yu, Yonghong Tian, and Tiejun Huang. Optimal ANN-SNN conversion for fast and accurate inference in deep spiking neural networks. In *International Joint Conference on Artificial Intelligence*, pp. 2328–2336, 2021.

Jianhao Ding, Jiyuan Zhang, Tiejun Huang, Jian K. Liu, and Zhaofei Yu. Assisting training of deep spiking neural networks with parameter initialization. *IEEE Transactions on Neural Networks and Learning Systems*, 36(8):15015–15028, 2025.

Chaoteng Duan, Jianhao Ding, Shiyan Chen, Zhaofei Yu, and Tiejun Huang. Temporal effective batch normalization in spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 34377–34390, 2022.

Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21056–21069, 2021a.

Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2641–2651, 2021b.

Wulfram Gerstner, Richard Kempter, J. Leo van Hemmen, and Hermann Wagner. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383:76–78, 1996.

Wulfram Gerstner, Werner M. Kistler, Richard Naud, and Liam Paninski. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge University Press, 2014.

S. Ghosh-Dastidar and H. Adeli. Spiking neural networks. *International Journal of Neural Systems*, 19(04):295–308, 2009.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, volume 9, pp. 249–256, 2010.

Joshua H. Goldwyn, Bradley R. Slabe, Joseph B. Travers, and David Terman. Gain control with A-type potassium current: IA as a switch between divisive and subtractive inhibition. *PLOS Computational Biology*, 14:1–23, 2018.

Yufei Guo, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Yinglei Wang, Xuhui Huang, and Zhe Ma. IM-Loss: Information maximization loss for spiking neural networks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 156–166, 2022a.

Yufei Guo, Xinyi Tong, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Zhe Ma, and Xuhui Huang. RecDis-SNN: Rectifying membrane potential distribution for directly training spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 326–335, 2022b.

Stefan Habenschuss, Johannes Bill, and Bernhard Nessler. Homeostatic plasticity in Bayesian spiking networks as expectation maximization with posterior constraints. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

Bilal Haider, Alvaro Duque, Andrea R. Hasenstaub, and David A. McCormick. Neocortical network activity In Vivo is generated through a dynamic balance of excitation and inhibition. *Journal of Neuroscience*, 26(17):4535–4545, 2006.

Bing Han and Kaushik Roy. Deep spiking neural network: Energy efficiency through time based coding. In *European Conference on Computer Vision*, pp. 388–404. Springer, 2020.

Bing Han, Gopalakrishnan Srinivasan, and Kaushik Roy. RMP-SNN: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13555–13564, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1026–1034, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 770–778, 2016.

Hua Hu, Jian Gan, and Peter Jonas. Fast-spiking, parvalbumin$^+$ GABAergic interneurons: From cellular design to microcircuit function. *Science*, 345(6196):1255263, 2014.

Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*, pp. 448–456, 2015.

Haiyan Jiang, Vincent Zoonekynd, Giulia De Masi, Bin Gu, and Huan Xiong. TAB: Temporal accumulated batch normalization in spiking neural networks. In *International Conference on Learning Representations*, 2024.

Hasna Nur Karimah, Chankyu Lee, and Yeongkyo Seo. Batchnorm-free binarized deep spiking neural network for a lightweight machine learning model. *Electronics*, 14(8):1602, 2025.

Youngeun Kim and Priyadarshini Panda. Revisiting batch normalization for training low-latency deep spiking neural networks from scratch. *Frontiers in Neuroscience*, 15:773954–773954, 2021.

Agnes Korcsak-Gorzo, Michael G. Müller, Andreas Baumbach, Luziwei Leng, Oliver J. Breitwieser, Sacha J. van Albada, Walter Senn, Karlheinz Meier, Robert Legenstein, and Mihai A. Petrovici. Cortical oscillations support sampling-based computations in spiking neural networks. *PLOS Computational Biology*, 18:1–41, 2022.

Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009.

Kaushalya Kumarasinghe, Nikola Kasabov, and Denise Taylor. Brain-inspired spiking neural networks for decoding and understanding muscle activity and kinematics from electroencephalography signals during hand movements. *Scientific Reports*, 11(1):2486, 2021.

Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge, 2015.

Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience*, 10:508, 2016.

Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: An event-stream dataset for object classification. *Frontiers in Neuroscience*, 2017.

Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu. Differentiable Spike: Rethinking gradient-descent for training spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 23426–23439, 2021.

Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

Henry Markram, Maria Toledo-Rodriguez, Yun Wang, Anirudh Gupta, Gilad Silberberg, and Caizhi Wu. Interneurons of the neocortical inhibitory system. *Nature Reviews Neuroscience*, 5(10): 793–807, 2004.

Aurora Micheli, Olaf Booij, Jan van Gemert, and Nergis Tömen. Deep activity propagation via weight initialization in spiking neural networks. In *Neuro Inspired Computational Elements (NICE)*, pp. 1–9, 2025.

Antony W. N'dri, William Gebhardt, Céline Teulière, Fleur Zeldenrust, Rajesh P. N. Rao, Jochen Triesch, and Alexander Ororbia. Predictive coding with spiking neural networks: a survey, 2024.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36:51–63, 2019.

Yuqi Pan, Yupeng Feng, Jinghao Zhuang, Siyu Ding, Zehao Liu, Bohan Sun, Yuhong Chou, Han Xu, Xuerui Qiu, Anlin Deng, Anjie Hu, Peng Zhou, Man Yao, Jibin Wu, Jian Yang, Guoliang Sun, Bo Xu, and Guoqi Li. SpikingBrain technical report: spiking brain-inspired large models. *arXiv preprint arXiv:2509.05276*, 2025.

Luke Y. Prince, Matthew M. Tran, Dorian Grey, Lydia Saad, Helen Chasiotis, Jeehyun Kwag, Michael M. Kohl, and Blake A. Richards. Neocortical inhibitory interneuron subtypes are differentially attuned to synchrony- and rate-coded information. *Communications Biology*, 4(1):935, 2021.

Ole Richter, Chenxi Wu, Adrian M Whatley, German Köstinger, Carsten Nielsen, Ning Qiao, and Giacomo Indiveri. DYNAP-SE2: a scalable multi-core dynamic neuromorphic asynchronous spiking neural network processor. *Neuromorphic Computing and Engineering*, 4(1):014003, 2024.

Julian Rossbroich, Julia Gygax, and Friedemann Zenke. Fluctuation-driven initialization for spiking neural network training. *Neuromorphic Computing and Engineering*, 2(4):044016, 2022.

Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575:607–617, 2019.

Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience*, 11:682, 2017.

Sadra Sadeh and Claudia Clopath. The emergence of NeuroAI: bridging neuroscience and artificial intelligence. *Nature Reviews Neuroscience*, 26:583–584, 2025.

Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in Neuroscience*, 13:95, 2019.

Zihang Shao, Xuanye Fang, Yaxin Li, Chaoran Feng, Jiangrong Shen, and Qi Xu. EICIL: Joint excitatory inhibitory cycle iteration learning for deep spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 36, pp. 32117–32128, 2023.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

DC Somers, SB Nelson, and M Sur. An emergent model of orientation selectivity in cat visual cortical simple cells. *Journal of Neuroscience*, 15(8):5448–5465, 1995.

Christoph Stöckl and Wolfgang Maass. Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes. *Nature Machine Intelligence*, 3(3):230–238, 2021.

Hugh R. Wilson and Jack D. Cowan. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical Journal*, 12(1):1–24, 1972.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, 12:331, 2018.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: faster, larger, better. In *AAAI Conference on Artificial Intelligence*, 2019.

Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Di He, and Zhouchen Lin. Online training through time for spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 20717–20730, 2022.

Yu Xiao, Yize Liu, Bihua Zhang, Peng Chen, Huaze Zhu, Enhui He, Jiayi Zhao, Wenju Huo, Xiaofei Jin, Xumeng Zhang, et al. Bio-plausible reconfigurable spiking neuron for neuromorphic computing. *Science Advances*, 11(6):eadr6733, 2025.

Siyu Zhang, Min Xu, Tsukasa Kamigaki, Johnny Phong Hoang Do, Wei-Cheng Chang, Sean Jenvay, Kazunari Miyamichi, Liqun Luo, and Yang Dan. Long-range and local circuits for top-down modulation of visual cortex processing. *Science*, pp. 660–665, 2014.

Dongcheng Zhao, Yi Zeng, and Yang Li. BackEISNN: A deep spiking neural network with adaptive self-feedback and balanced excitatory–inhibitory neurons. *Neural Networks*, 154:68–77, 2022.

Lusen Zhao, Zihan Huang, Jianhao Ding, and Zhaofei Yu. TTFSFormer: A TTFS-based lossless conversion of spiking transformer. In *Proceedings of the International Conference on Machine Learning*, 2025.

Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11062–11070, 2021.

Yaoyu Zhu, Jianhao Ding, Tiejun Huang, Xiaodong Xie, and Zhaofei Yu. Online stabilization of spiking neural networks. In *International Conference on Learning Representations*, 2024.

A LLM Usage

We utilize LLMs to assist in proofreading and enhancing the clarity and readability of the manuscript. The core scientific contributions, experimental design, and data analysis are conducted entirely by the authors.

B Reproducibility Statement

All essential details regarding datasets, model architectures, and training hyperparameters are provided in Section 5 and Appendix F.6 to ensure reproducibility. The source code for our framework has been made publicly available at `https://github.com/vwOvOwv/DeepEISNN`.

C Ethics Statement

This work is foundational research focused on learning algorithms for brain-inspired neural networks and exclusively uses standard public datasets. We do not foresee any direct negative societal impacts or ethical concerns arising from this research.

D Detailed Mathematical Derivations

This section provides detailed derivations for mathematical formulations introduced in the main text, using same notations.

D.1 Dynamics of the fast-spiking inhibitory neurons

In this section, we provide a formal derivation justifying the modelling of inhibitory neurons as stateless ReLU-like units in our discrete-time simulation. We demonstrate that this approximation is mathematically rigorous under the condition $\tau_{\mathrm{I}} \ll \Delta t$. The sub-threshold dynamics of a LIF inhibitory neuron are governed by the following differential equation:

$$\tau_{\mathrm{I}} \frac{\mathrm{d}\mathbf{u}_{\mathrm{I}}^{[l]}(t)}{\mathrm{d}t} = -\left(\mathbf{u}_{\mathrm{I}}^{[l]}(t) - u_{\mathrm{I,rest}}\right) + \mathbf{I}_{\mathrm{I}}^{[l]}(t), \tag{22}$$

where τ_{I} is the membrane time constant, $u_{\mathrm{I,rest}}$ is the resting potential, and $\mathbf{I}_{\mathrm{I}}^{[l]}(t)$ is the input currents. Assuming that $u_{\mathrm{I,rest}} = 0$ and the input currents remain constant at $\mathbf{I}_0$ over the duration of a discrete time step Δt (starting from time t), the analytical solution for the membrane potential at time $t + \Delta t$ is

$$\mathbf{u}_{\mathrm{I}}^{[l]}(t + \Delta t) = \mathbf{u}_{\mathrm{I}}^{[l]}(t) e^{-\frac{\Delta t}{\tau_{\mathrm{I}}}} + \mathbf{I}_0 \left(1 - e^{-\frac{\Delta t}{\tau_{\mathrm{I}}}}\right). \tag{23}$$

In our simulation setup, excitatory neurons typically have a time constant $\tau_{\mathrm{E}} = 2$, which is on the same order of magnitude as the simulation time step $\Delta t = 1$. In contrast, inhibitory neurons in our model represent biological fast-spiking interneurons (e.g., PV+ neurons), which are characterized by significantly smaller time constants compared to excitatory pyramidal neurons (Hu et al., 2014; Prince et al., 2021). This biological property implies the condition $\tau_{\mathrm{I}} \ll \tau_{\mathrm{E}}$, and consequently, $\tau_{\mathrm{I}} \ll \Delta t$. Under this condition, the ratio $\frac{\Delta t}{\tau_{\mathrm{I}}}$ becomes very large, causing the exponential decay factor to approach zero:

$$\lim_{\tau_{\mathrm{I}} \to 0} e^{-\frac{\Delta t}{\tau_{\mathrm{I}}}} = 0. \tag{24}$$

Substituting this limit into the update rule, the term $\mathbf{u}_{\mathrm{I}}^{[l]}(t)$ vanishes, and the equation simplifies to:

$$\mathbf{u}_{\mathrm{I}}^{[l]}(t + 1) \approx \mathbf{I}_0. \tag{25}$$

This result indicates that the membrane potential reaches a steady state determined entirely by the input currents almost instantaneously within a single time step. Consequently, the inhibitory neurons effectively converge to stateless units that do not carry temporal information across time steps.

Furthermore, the simulation time step Δt is much longer than the intrinsic dynamics of the inhibitory neurons, which allows them to fire multiple times within a single step when the input is strong. By applying a firing threshold of 1 and soft-reset mechanism, the total spike count at time $t+1$ is given by

$$\mathbf{s}_{\mathrm{I}}^{[l]}(t+1) = \left\lfloor \max\left(0, \mathbf{u}_{\mathrm{I}}^{[l]}(t+1)\right) \right\rfloor \approx \lfloor \max\left(0, \mathbf{I}_0\right) \rfloor \approx \max\left(0, \mathbf{I}_0\right). \tag{26}$$

This derivation justifies the ReLU-like approximation for inhibitory neurons in the main text. Importantly, since the transient response is completed within a single step Δt, there is no accumulation of approximation error over time. The validity of this model is strictly determined by the timescale separation $\tau_{\mathrm{I}} \ll \Delta t$, making it a reasonable approximation.

D.2 Initialization of $\boldsymbol{W}_{\mathrm{EE}}^{[l]}, \boldsymbol{W}_{\mathrm{IE}}^{[l]}, \boldsymbol{W}_{\mathrm{EI}}^{[l]}$

Our primary goal is to achieve a zero-mean expected net input for each excitatory neuron i in layer l (Equation 16) at initialization. From Equation 9 and Equation 10 we have

$$\mathbf{I}_{\mathrm{EE},i}^{[l]}[t] = \sum_{j=1}^{d} \mathrm{W}_{\mathrm{EE},ij}^{[l]} \mathbf{s}_{\mathrm{E},j}^{[l-1]}[t], i = 1, 2, \ldots, n_{\mathrm{E}}^{[l]}, \tag{27}$$

$$\mathbf{I}_{\mathrm{IE},i}^{[l]}[t] = \sum_{j=1}^{d} \mathrm{W}_{\mathrm{IE},ij}^{[l]} \mathbf{s}_{\mathrm{E},j}^{[l-1]}[t], i = 1, 2, \ldots, n_{\mathrm{I}}^{[l]}. \tag{28}$$

Here $\mathrm{W}_{\mathrm{EE},ij}^{[l]}$ and $\mathrm{W}_{\mathrm{IE},ij}^{[l]}$ denote the (i, j) element of $\boldsymbol{W}_{\mathrm{EE}}^{[l]}$ and $\boldsymbol{W}_{\mathrm{IE}}^{[l]}$, respectively. Other notations are consistent with those in the main text. Then similarly, from Equation 11 and Equation 12 we obtain

$$\mathbf{I}_{\mathrm{EI,sub},i}^{[l]}[t] = \sum_{j=1}^{n_{\mathrm{I}}^{[l]}} \mathrm{W}_{\mathrm{EI},ij}^{[l]} \mathbf{s}_{\mathrm{I},j}^{[l]}[t] \approx \sum_{j=1}^{n_{\mathrm{I}}^{[l]}} \mathrm{W}_{\mathrm{EI},ij}^{[l]} \mathbf{I}_{\mathrm{IE},j}^{[l]}[t] \tag{29}$$

$$= \sum_{j=1}^{n_{\mathrm{I}}^{[l]}} \mathrm{W}_{\mathrm{EI},ij}^{[l]} \sum_{k=1}^{d} \mathrm{W}_{\mathrm{IE},jk}^{[l]} \mathbf{s}_{\mathrm{E},k}^{[l-1]}[t], i = 1, 2, \ldots, n_{\mathrm{E}}^{[l]}. \tag{30}$$

Note that $\mathbf{s}_{\mathrm{I},j}^{[l]}[t] = \mathcal{F}_{\mathrm{I}}^{[l]}\left(\mathbf{I}_{\mathrm{IE},j}^{[l]}[t]\right) \approx \max\left(0, \mathbf{I}_{\mathrm{IE},j}^{[l]}[t]\right) = \mathbf{I}_{\mathrm{IE},j}^{[l]}[t]$ since the elements of $\boldsymbol{W}_{\mathrm{IE}}^{[l]}$ and $\mathbf{s}_{\mathrm{E}}^{[l-1]}$ are non-negative. Therefore, by assuming $\mathbf{s}_{\mathrm{E},k}^{[l-1]}[t] \overset{\text{i.i.d.}}{\sim} \mathrm{Bern}(p)$ and elements of weights are i.i.d. at initialization, we obtain Equation 17 and Equation 18.

D.3 Initialization of $\mathbf{g}_{\mathrm{I}}^{[l]}$

$\mathbf{g}_{\mathrm{I}}^{[l]}$ is a gain factor modulating the strength of divisive inhibition. As mentioned in the main text, we initialize it by setting the condition $\mathbb{E}\left[\mathbf{I}_{\mathrm{EI,div},i}^{[l]}\right] = \mathrm{std}\left(\mathbf{I}_{\mathrm{EE},i}^{[l]}\right)$.

Similar to the derivation of $\mathbf{I}_{\mathrm{EI,sub,i}}^{[t]}[t]$ in Appendix D.2,

$$\mathbb{E}\left[\mathbf{I}_{\mathrm{EI,div},i}^{[l]}\right] = n_{\mathrm{I}}^{[l]} dp \mathbb{E}\left[\mathbf{g}_{\mathrm{I}}^{[l]}\right] \mathbb{E}\left[\mathrm{W}_{\mathrm{IE}}^{[l]}\right] \mathbb{E}\left[\mathrm{W}_{\mathrm{EI}}^{[l]}\right] = \frac{dp \mathbb{E}\left[\mathbf{g}_{\mathrm{I}}^{[l]}\right]}{\lambda^{[l]}}. \tag{31}$$

Below we derive the standard deviation of $\mathbf{I}_{\mathrm{EE},i}^{[l]}$. The variance of the excitatory input currents to neuron i is

$$\mathrm{Var}\left(\mathbf{I}_{\mathrm{EE},i}^{[l]}\right) = \mathrm{Var}\left(\sum_{j=1}^{d} \mathrm{W}_{\mathrm{EE},ij}^{[l]} s_{\mathrm{E},j}^{[l-1]}\right), \tag{32}$$

where $\mathrm{W}_{\mathrm{EE},ij}^{[l]}$ denotes the (i, j) element of $\boldsymbol{W}_{\mathrm{EE}}^{[l]}$. Assuming the terms $\mathrm{W}_{\mathrm{EE},ij}^{[l]} s_{\mathrm{E},j}^{[l-1]}$ are independent for each j, the variance of the sum is the sum of the variances,

$$\mathrm{Var}\left(\mathbf{I}_{\mathrm{EE},i}^{[l]}\right) = \sum_{j=1}^{d} \mathrm{Var}\left(\mathrm{W}_{\mathrm{EE},ij}^{[l]} s_{\mathrm{E},j}^{[l-1]}\right). \tag{33}$$

By further assuming that the weights $\mathrm{W}_{\mathrm{EE},ij}^{[l]}$ and input signals $s_{\mathrm{E},j}^{[l-1]}$ are independently distributed, we can simplify this to

$$\mathrm{Var}\left(\mathbf{I}_{\mathrm{EE},i}^{[l]}\right) = d \cdot \mathrm{Var}\left(\mathrm{W}_{\mathrm{EE}}^{[l]} s_{\mathrm{E}}^{[l-1]}\right) \tag{34}$$

$$= d\left(\mathbb{E}\left[(s_{\mathrm{E}}^{[l-1]})^2\right] \mathrm{Var}\left(\mathrm{W}_{\mathrm{EE}}^{[l]}\right) + \mathrm{Var}\left(s_{\mathrm{E}}^{[l-1]}\right) \mathbb{E}^2\left[\mathrm{W}_{\mathrm{EE}}^{[l]}\right]\right). \tag{35}$$

As established in the main text, we model the input spikes as an i.i.d. Bernoulli distribution with parameter p. Thus, $\mathbb{E}\left[s_{\mathrm{E}}^{[l-1]}\right] = p$, $\mathbb{E}\left[(s_{\mathrm{E}}^{[l-1]})^2\right] = p$, and $\mathrm{Var}\left(s_{\mathrm{E}}^{[l-1]}\right) = p(1-p)$. The weights $\mathrm{W}_{\mathrm{EE}}^{[l]}$ are drawn from an exponential distribution with rate $\lambda^{[l]}$, for which $\mathbb{E}\left[\mathrm{W}_{\mathrm{EE}}^{[l]}\right] = 1/\lambda^{[l]}$ and $\mathrm{Var}\left(\mathrm{W}_{\mathrm{EE}}^{[l]}\right) = 1/(\lambda^{[l]})^2$. Substituting these into Equation 35 yields:

$$\mathrm{Var}\left(\mathbf{I}_{\mathrm{EE},i}^{[l]}\right) = d\left(p \cdot \frac{1}{(\lambda^{[l]})^2} + p(1-p) \cdot \frac{1}{(\lambda^{[l]})^2}\right) \tag{36}$$

$$= \frac{d}{(\lambda^{[l]})^2}\left(p + p(1-p)\right) \tag{37}$$

$$= \frac{dp(2-p)}{(\lambda^{[l]})^2}. \tag{38}$$

Therefore, the standard deviation of $\mathbf{I}_{\mathrm{EE},i}^{[l]}$ is:

$$\mathrm{std}\left(\mathbf{I}_{\mathrm{EE},i}^{[l]}\right) = \sqrt{\mathrm{Var}\left(\mathbf{I}_{\mathrm{EE},i}^{[l]}\right)} = \frac{\sqrt{dp(2-p)}}{\lambda^{[l]}}. \tag{39}$$

D.4 Selection of rate parameter $\lambda^{[l]}$

To ensure stable signal propagation at initialization, we set the standard deviation of the input currents to match that of the input spikes, i.e., $\mathrm{std}\left(\mathbf{I}_{\mathrm{EE},i}^{[l]}\right) = \mathrm{std}\left(\mathbf{s}_{\mathrm{E}}^{[l-1]}\right) = \sqrt{p(1-p)}$. Using the result from Equation 39 we have

$$\frac{\sqrt{dp(2-p)}}{\lambda^{[l]}} = \sqrt{p(1-p)}, \tag{40}$$

$$\lambda^{[l]} = \frac{\sqrt{dp(2-p)}}{\sqrt{p(1-p)}} = \sqrt{\frac{d(2-p)}{1-p}}. \tag{41}$$

D.5 Backpropagation of the E-I circuit

In this section, we provide a detailed backpropagation derivation for the E-I circuit to theoretically justify the necessity of the gradient scaling on $\mathbf{W}_{\mathrm{EI}}$.

We first re-write the forward pass formulation at time step t for layer l:

$$\mathbf{I}_{\mathrm{int}}^{[l]}[t] = \mathbf{g}_{\mathrm{E}}^{[l]} \odot \frac{\mathbf{I}_{\mathrm{EE}}^{[l]}[t] - \mathbf{I}_{\mathrm{EI,sub}}^{[l]}[t]}{\mathbf{I}_{\mathrm{EI,div}}^{[l]}[t]} + \mathbf{b}_{\mathrm{E}}^{[l]}, \tag{42}$$

where the currents are defined as:

$$\mathbf{I}_{\mathrm{EI,sub}}^{[l]}[t] = \boldsymbol{W}_{\mathrm{EI}}^{[l]} \mathbf{s}_{\mathrm{I}}^{[l]}[t], \tag{43}$$

$$\mathbf{I}_{\mathrm{EI,div}}^{[l]}[t] = \boldsymbol{W}_{\mathrm{EI}}^{[l]}(\mathbf{g}_{\mathrm{I}}^{[l]} \odot \mathbf{s}_{\mathrm{I}}^{[l]}[t]), \tag{44}$$

$$\mathbf{s}_{\mathrm{I}}^{[l]}[t] \approx \boldsymbol{W}_{\mathrm{IE}}^{[l]} \mathbf{s}_{\mathrm{E}}^{[l-1]}[t]. \tag{45}$$

Note that here we use the linear approximation for $\mathbf{s}_{\mathrm{I}}$ derived in Appendix D.1.

Let $\boldsymbol{\delta}^{[l]}[t] = \frac{\partial \mathcal{L}}{\partial \mathbf{I}_{\mathrm{int}}^{[l]}[t]}$ be the error signal backpropagated from subsequent layers at time t. The gradients are accumulated over T time steps. Below we provide the derivative of loss w.r.t. each trainable parameter in layer l of the E-I circuit.

1. $\boldsymbol{W}_{\mathrm{EE}}^{[l]}$.

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{\mathrm{EE}}^{[l]}} = \sum_{t=1}^{T} \left[\left(\boldsymbol{\delta}^{[l]}[t] \odot \frac{\mathbf{g}_{\mathrm{E}}^{[l]}}{\mathbf{I}_{\mathrm{EI,div}}^{[l]}[t]} \right) \left(\mathbf{s}_{\mathrm{E}}^{[l-1]}[t] \right)^{\top} \right]. \tag{46}$$

2. $\boldsymbol{W}_{\mathrm{EI}}^{[l]}$.

$\boldsymbol{W}_{\mathrm{EI}}^{[l]}$ contributes to both the subtractive and divisive pathways. Therefore, its gradient can be decomposed into subtractive and divisive components that correspond to the two types of inhibitory currents.

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{\mathrm{EI}}^{[l]}} = \sum_{t=1}^{T} \Bigg[\underbrace{\left(\boldsymbol{\delta}^{[l]}[t] \odot \frac{-\mathbf{g}_{\mathrm{E}}^{[l]}}{\mathbf{I}_{\mathrm{EI,div}}^{[l]}[t]} \right) \left(\mathbf{s}_{\mathrm{I}}^{[l]}[t] \right)^{\top}}_{\text{Subtractive Component}} + \underbrace{\left(\boldsymbol{\delta}^{[l]}[t] \odot \frac{-\mathbf{g}_{\mathrm{E}}^{[l]} \odot \mathbf{I}_{\mathrm{balanced}}^{[l]}[t]}{\mathbf{I}_{\mathrm{EI,div}}^{[l]}[t] \odot \mathbf{I}_{\mathrm{EI,div}}^{[l]}[t]} \right) \left(\mathbf{g}_{\mathrm{I}}^{[l]} \odot \mathbf{s}_{\mathrm{I}}^{[l]}[t] \right)^{\top}}_{\text{Divisive Component}} \Bigg], \tag{47}$$

where $\mathbf{I}_{\mathrm{balanced}}^{[l]}[t] = \mathbf{I}_{\mathrm{EE}}^{[l]}[t] - \mathbf{I}_{\mathrm{EI,sub}}^{[l]}[t]$.

3. $\boldsymbol{W}_{\mathrm{IE}}^{[l]}$.

The error propagates back through the inhibitory neurons.

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{\mathrm{IE}}^{[l]}} = \sum_{t=1}^{T} \left(\boldsymbol{\delta}_{\mathrm{I}}^{[l]}[t] \right) \left(\mathbf{s}_{\mathrm{E}}^{[l-1]}[t] \right)^{\top}, \tag{48}$$

where the error term $\boldsymbol{\delta}_{\mathrm{I}}$ combines gradients from both $\boldsymbol{W}_{\mathrm{EI}}^{[l]}$ pathways. Therefore,

$$\boldsymbol{\delta}_{\mathrm{I}}^{[l]}[t] = \boldsymbol{W}_{\mathrm{EI}}^{[l]\top} \left(\boldsymbol{\delta}^{[l]}[t] \odot \frac{-\mathbf{g}_{\mathrm{E}}^{[l]}}{\mathbf{I}_{\mathrm{EI,div}}^{[l]}[t]} \right) + \mathbf{g}_{\mathrm{I}}^{[l]} \odot \left[\boldsymbol{W}_{\mathrm{EI}}^{[l]\top} \left(\boldsymbol{\delta}^{[l]}[t] \odot \frac{-\mathbf{g}_{\mathrm{E}}^{[l]} \odot \mathbf{I}_{\mathrm{balanced}}^{[l]}[t]}{\mathbf{I}_{\mathrm{EI,div}}^{[l]}[t] \odot \mathbf{I}_{\mathrm{EI,div}}^{[l]}[t]} \right) \right]. \tag{49}$$

4. $\mathbf{g}_{\mathrm{I}}^{[l]}$ This parameter modulates the inhibitory signals before they are projected by $\boldsymbol{W}_{\mathrm{EI}}^{[l]}$ for the divisive pathway. Therefore, the error signal propagates back through the divisive branch of $\boldsymbol{W}_{\mathrm{EI}}^{[l]}$.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{g}_{\mathrm{I}}^{[l]}} = \sum_{t=1}^{T} \left[\boldsymbol{W}_{\mathrm{EI}}^{[l]\top} \left(\boldsymbol{\delta}^{[l]}[t] \odot \frac{-\mathbf{g}_{\mathrm{E}}^{[l]} \odot \mathbf{I}_{\mathrm{balanced}}^{[l]}[t]}{\mathbf{I}_{\mathrm{EI,div}}^{[l]}[t] \odot \mathbf{I}_{\mathrm{EI,div}}^{[l]}[t]} \right) \right] \odot \mathbf{s}_{\mathrm{I}}^{[l]}[t]. \tag{50}$$

5. $\mathbf{g}_{\mathrm{E}}^{[l]}$.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{g}_{\mathrm{E}}^{[l]}} = \sum_{t=1}^{T} \left(\boldsymbol{\delta}^{[l]}[t] \odot \frac{\mathbf{I}_{\mathrm{balanced}}^{[l]}[t]}{\mathbf{I}_{\mathrm{EI,div}}^{[l]}[t]} \right). \tag{51}$$

6. $\mathbf{b}_{\mathrm{E}}^{[l]}$.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}_{\mathrm{E}}^{[l]}} = \sum_{t=1}^{T} \boldsymbol{\delta}^{[l]}[t]. \tag{52}$$

E Implementation Details

E.1 CNNs with the E-I circuit

In experiments, we mainly use CNNs like VGG and ResNet, where all convolutional layers and the fully connected classifier are implemented with the proposed E-I circuit (for output layer, we do not apply divisive inhibition to better stabilize the output logits). Here we clarify the implementation of the convolution layer used in our experiments. Specifically, $\boldsymbol{W}_{\mathrm{EE}}$ and $\boldsymbol{W}_{\mathrm{IE}}$ are standard $K \times K$ convolution kernels, while the lateral connection $\boldsymbol{W}_{\mathrm{EI}}$ is a 1×1 point-wise convolution kernel. In this context, the input dimension d is defined as $d = C_{\mathrm{in}} \times K \times K$, where C_{in} is the number of channels of excitatory input, and the population sizes n_{E} and n_{I} are the number of excitatory and inhibitory output channels, respectively. This configuration allows inhibitory neurons to regulate excitatory neurons densely across the channel dimension while preserving the spatial structure.

E.2 Dynamic initialization

While our theoretical analysis under Bernoulli assumption provides basic ideas of E-I Init, we find it more effective and stable during training if we estimate statistics from training data at initialization, rather than formulating them with Bernoulli distribution parameter p and manually set p. Similar to the theoretical analysis under the Bernoulli assumption above, we have

$$\lambda^{[l]} = \sqrt{\frac{d\left(\mathbb{E}\left[\left(\mathbf{s}_{\mathrm{E}}^{[l-1]}\right)^2\right] + \mathrm{Var}\left(\mathbf{s}_{\mathrm{E}}^{[l-1]}\right)\right)}{\mathrm{Var}\left(\mathbf{s}_{\mathrm{E}}^{[l-1]}\right)}}, \tag{53}$$

$$\mathbf{g}_{\mathrm{I}}^{[l]} = \sqrt{\frac{\left(\mathbb{E}\left[\left(\mathbf{s}_{\mathrm{E}}^{[l-1]}\right)^2\right] + \mathrm{Var}\left(\mathbf{s}_{\mathrm{E}}^{[l-1]}\right)\right)}{d\mathbb{E}^2\left[\mathbf{s}_{\mathrm{E}}^{[l-1]}\right]}}. \tag{54}$$

Therefore, by computing mean, second raw moment, and sample-wise variance only once at initialization, the model performs self-regulated E-I balance and gain control, stabilizing training without explicit normalization. Algorithm 1 summarizes the implementation of the proposed E-I init.

Algorithm 1 E-I Init

Require: Input $\mathbf{X}$ of shape $(B, \dots)$ (first batch), input dimension d, inhibitory neuron count n_{I}.
Ensure: Initialized parameters $\boldsymbol{W}_{\mathrm{EE}}, \boldsymbol{W}_{\mathrm{IE}}, \boldsymbol{W}_{\mathrm{EI}}, \mathbf{g}_{\mathrm{I}}, \mathbf{g}_{\mathrm{E}}, \mathbf{b}_{\mathrm{E}}$.

```
procedure EI-Init($\mathbf{X}, d, n_{\mathrm{I}}$)
                                   ▷ Step 1: Estimate input statistics from the first batch $\mathbf{X}$
    mean ← $\mathbf{X}$.mean()
    var ← $\mathbf{X}$.var(dim = 0).mean()
    moment ← ($\mathbf{X}^2$).mean()
                                   ▷ Step 2: Calculate the rate parameter based on statistics
    exp_scale ← $\sqrt{\frac{\mathrm{var}}{d\cdot(\mathrm{moment}+\mathrm{var})}}$          ▷ exp_scale = $1/\lambda$
    gain_I ← $\frac{1}{\sqrt{d}} \cdot \frac{\sqrt{\mathrm{moment}+\mathrm{var}}}{\mathrm{mean}}$          ▷ Initial value for $\mathrm{g}_{\mathrm{I}}$
                                   ▷ Step 3: Initialize trainable parameters
    $\mathrm{W}_{\mathrm{EE}}$ ∼ Exponential(scale = exp_scale)
    $\mathrm{W}_{\mathrm{IE}}$ ∼ Exponential(scale = exp_scale)
    $\mathrm{W}_{\mathrm{EI}}$ ← $\frac{1}{n_{\mathrm{I}}}$
    $\mathrm{g}_{\mathrm{I}}$ ← gain_I
    $\mathbf{g}_{\mathrm{E}}$ ← $\mathbf{1}$
    $\mathbf{b}_{\mathrm{E}}$ ← $\mathbf{0}$
end procedure
```

E.3 ADAPTIVE STABILIZATION WITH STE

Algorithm 2 demonstrates the full procedure of our proposed adaptive stabilization of divisive inhibition mechanism, including the backward pass with STE.

Algorithm 2 E-I Prop

```
Require: Input $\mathbf{X}$ of shape $(B, \dots)$, where $B$ is the batch size.
Ensure: Output $\mathbf{X}_{\text{out}}$ with zeros adaptively replaced.
procedure AdaptiveStabilization($\mathbf{X}$)
    if $\mathbf{X}$ contains no zero values then
        return $\mathbf{X}$
    end if
                                        ▷ Step 1: Replace zeros with second minimum
    $\mathbf{M} \leftarrow (\mathbf{X} == 0)$           ▷ Create a boolean mask for all zero locations
    $\mathbf{X}_{\text{tmp}} \leftarrow \mathbf{X}$
    $\mathbf{X}_{\text{tmp}}[\mathbf{M}] \leftarrow \infty$          ▷ Temporarily replace zeros with infinity
    for each sample $i$ from 1 to $B$ do
        $s_i \leftarrow \min(\mathbf{X}_{\text{tmp}}[i])$     ▷ Find the smallest positive value of the original sample
        $\mathbf{S}[i] \leftarrow s_i$
    end for
    $\mathbf{X}_{\text{fwd}} \leftarrow \text{where}(\mathbf{M}, \mathbf{S}, \mathbf{X})$    ▷ Replace zeros with the smallest positive value of the sample
                                        ▷ Step 2: Construct the final output with STE
    $\mathbf{X}_{\text{out}} \leftarrow \text{detach}(\mathbf{X}_{\text{fwd}}) + (\mathbf{X} - \text{detach}(\mathbf{X}))$   ▷ STE via the detach trick
    return $\mathbf{X}_{\text{out}}$
end procedure
```

F SUPPLEMENTARY RESULTS AND EXPERIMENT DETAILS

F.1 EMPIRICAL ANALYSIS OF GRADIENT FLOW AND SCALING ROBUSTNESS

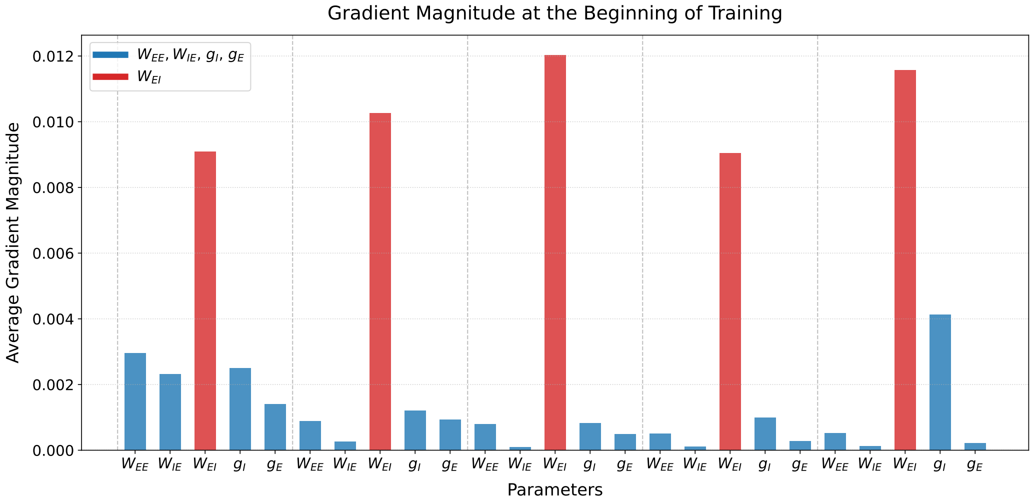


Figure 5: Empirical analysis of gradient norms at initialization for convolutional layers in VGG-8, without gradient scaling. The gradients for $\boldsymbol{W}_{\text{EI}}$ (**red**) are orders of magnitude larger than those for $\boldsymbol{W}_{\text{EE}}$, $\boldsymbol{W}_{\text{IE}}$, and gain parameters (**blue**), consistent with the theoretical analysis.

Figure 5 visualizes the magnitudes of gradient norms for all trainable parameters across the convolutional layers of VGG-8 at the first training iteration (without gradient scaling). Consistent with our theoretical derivation in Appendix D.5, the gradient norms for $\boldsymbol{W}_{\text{EI}}$ are disproportionately larger than those of other parameters. This imbalance stems primarily from the divisive operation, which introduces a term proportional to $1/\left(\mathbf{I}^{[l]}_{\text{EI,div}}[t] \odot \mathbf{I}^{[l]}_{\text{EI,div}}[t]\right)$ in the gradient, leading to quadratic amplification when the denominator is small. While the gradients for $\boldsymbol{W}^{[l]}_{\text{IE}}$ and $\mathbf{g}^{[l]}_{\text{I}}$ also contain this term, they are implicitly dampened since the gradients backpropagate through $\boldsymbol{W}^{[l]}_{\text{EI}}$, which acts as an averaging filter due to its deterministic initialization of $1/n^{[l]}_{\text{I}}$. In contrast, the gradient for $\boldsymbol{W}^{[l]}_{\text{EI}}$ lacks such averaging mechanism and is instead directly proportional to the inhibitory activity $\mathbf{s}^{[l]}_{\text{I}}$. Consequently, the gradient magnitude of $\boldsymbol{W}^{[l]}_{\text{EI}}$ is driven by the input spikes $\mathbf{s}^{[l]}_{\text{I}}[t] \approx \boldsymbol{W}^{[l]}_{\text{IE}}\mathbf{s}^{[l-1]}_{\text{E}}[t]$, scaling linearly with the input dimension d.

To counteract this amplification, we choose the scaling factor as $1/d$. To verify the robustness of scaling factor choice, a sensitivity analysis is conducted on VGG-8 (CIFAR-10) by varying the scaling factor from $1/\sqrt{d}$ to $1/d^2$ (see Table 3). The result demonstrates that our method is stable across a broad range of scaling factors (e.g., $1/\sqrt{d}$ and $1/d$ yield comparable performance), whereas removing the scaling immediately leads to collapse.

Table 3: Sensitivity analysis of gradient scaling factor with VGG-8 on CIFAR-10.

Scaling Factor	**Top-1 Accuracy (%)**
No Scaling	Collapsed
$1/\sqrt{d}$	86.87
$1/d$ (Default)	**86.88**
$1/d^2$	86.33

F.2 Comparison with E-I ANNs

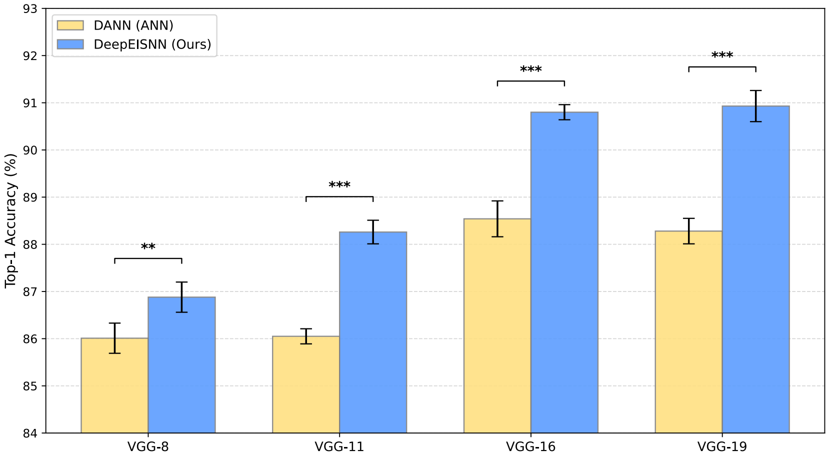


Figure 6: Comparison between our method and DANN on CIFAR-10. Error bars denote the standard deviation over multiple independent runs. Statistical significance between the two methods is indicated by asterisks (** $p < 0.01$, *** $p < 0.001$).

Comparison with DANN (Cornford et al., 2021) highlights the advantage of our method. Figure 6 shows that our method consistently and significantly outperforms DANN across all tested VGG architectures. Statistical analysis confirms that these improvements are statistically significant ($p < 0.01$ for VGG-8 and $p < 0.001$ for deeper models). Notably, the advantage of our method increases as the network depth increases. The performance gap widens from 0.87% on VGG-8 to 2.65% on VGG-19. This trend strongly suggests that our proposed mechanisms, E-I Init and E-I Prop, are more effective at preserving stable signal propagation and facilitating effective learning in very deep architectures.

F.3 Distributions of BN outputs

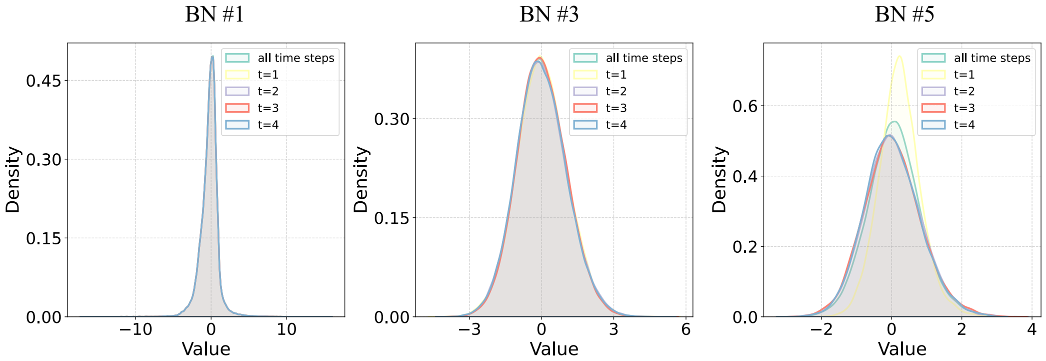


Figure 7: Distributions of the outputs in the first, third and fifth BN layers after training.

Figure 7 demonstrates output distributions of BN layers after training, which are all Gaussian-like and zero-centered.

F.4 Visualization of E-I interaction

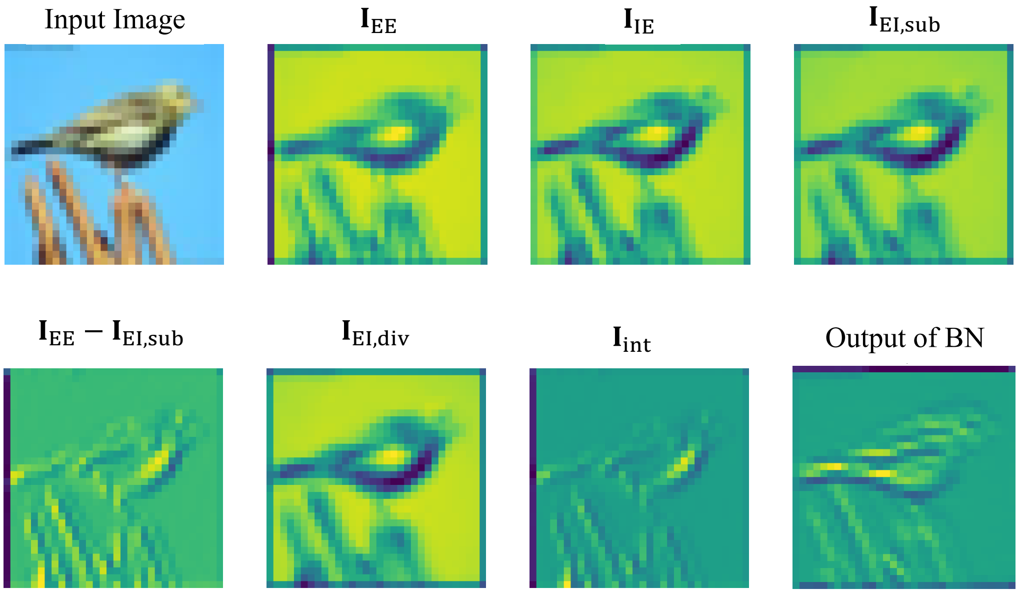

Input Image
$\mathbf{I}_{\mathrm{EE}}$
$\mathbf{I}_{\mathrm{IE}}$
$\mathbf{I}_{\mathrm{EI,sub}}$
$\mathbf{I}_{\mathrm{EE}} - \mathbf{I}_{\mathrm{EI,sub}}$
$\mathbf{I}_{\mathrm{EI,div}}$
$\mathbf{I}_{\mathrm{int}}$
Output of BN

Figure 8: Comparison between feature maps of the first E-I circuit layer in our model and the feature map of the first BN layer in SNN with vanilla BN after training.

A visual comparison of the feature maps suggests that our E-I circuit and vanilla BN forces network to focus on different feature representations. As shown in Figure 8, our E-I circuit produces a feature map where activations are concentrated along the object's contours, indicating a learned focus on feature edges and boundaries. In contrast, a standard BN layer may preserve a dense spatial output of its preceding convolution, normalizing the representation of the feature's overall shape and texture rather than isolating its boundaries.

F.5 Computational overhead analysis

To quantify the computational cost associated with ensuring biological fidelity, we measure the training time (per epoch) and peak GPU memory usage on a single NVIDIA GeForce RTX 4090. We compare our DeepEISNN with SNNs with vanilla BN across various architectures. The results are summarized in Table 4.

Table 4: Computational overhead ($T = 4$, batch size=128, single GPU).

Arch.	**Metric**	**SNN (Baseline)**	**DeepEISNN (Ours)**	**Factor**
VGG-8	Time/Epoch Memory	7.3s 2394 MB	18.1s 2684 MB	2.48× 1.12×
VGG-11	Time/Epoch Memory	9.5s 2392 MB	23.1s 2886 MB	2.43× 1.21×
VGG-16	Time/Epoch Memory	16.9s 3544 MB	42.2s 5072 MB	2.50× 1.43×
VGG-19	Time/Epoch Memory	19.4s 3734 MB	47.6s 5402 MB	2.45× 1.45×
ResNet-18	Time/Epoch Memory	38.8s 5918 MB	90.0s 9302 MB	2.32× 1.57×

As shown in Table 4, our method introduces a computational overhead of approximately $2.3\times \sim 2.5\times$ in training time and $1.1\times \sim 1.6\times$ in GPU memory usage. This increase is an expected and necessary trade-off for the E-I circuit with a 4:1 excitatory-to-inhibitory ratio. Unlike standard SNNs that utilize a single synaptic weight matrix per layer, our framework explicitly models three distinct synaptic projections ($\boldsymbol{W}_{\mathrm{EE}}, \boldsymbol{W}_{\mathrm{IE}}, \boldsymbol{W}_{\mathrm{EI}}$) and maintains an additional inhibitory population. Importantly, this overhead scales linearly with network size, ensuring tractability for deep architectures. Despite the increased per-epoch cost, our method demonstrates robust convergence comparable to BN-equipped baselines (as evidenced by the competitive accuracy in Table 1), thereby enabling normalization-free learning using biologically grounded mechanisms.

F.6 Experiment details

Code is implemented using the PyTorch framework and run on NVIDIA GeForce RTX 4090 GPUs.

Network architectures. We employ standard backbones, including VGG-8/11/16/19 (Simonyan & Zisserman, 2015) and ResNet-18 (He et al., 2016; Fang et al., 2021a). In these architectures, the standard convolutional blocks (Conv-BN-LIF) and linear classifiers are replaced by our proposed E-I circuit. To construct a lightweight classifier, we apply global average pooling (GAP) before the linear readout layer. VGG-8 is utilized primarily for ablation studies on CIFAR-10, while deeper models (VGG-16/19, ResNet-18) are employed for SOTA comparisons and large-scale benchmarks.

Data preprocessing. Our method is validated on multiple datasets, including CIFAR-10/100 (Krizhevsky, 2009), CIFAR10-DVS (Li et al., 2017), DVS-Gesture (Amir et al., 2017), and TinyImageNet (Le & Yang, 2015), using their standard train and validation splits. We apply distinct data augmentation strategies on different datasets. Specifically, for CIFAR-10 and CIFAR-100, we employ random cropping with a size of 32×32 (padding of 4 pixels), random horizontal flipping, and cutout. For TinyImageNet, images are downsampled to 32×32, and augmentations include random resized cropping, random horizontal flipping, color jittering, and cutout. Regarding neuromorphic datasets, both CIFAR10-DVS and DVS-Gesture are resized to a spatial resolution of 48×48. The training pipeline for these event-based datasets includes random resized cropping and random horizontal flipping. Additionally, we apply random temporal deletion specifically for the DVS-Gesture dataset to enhance temporal robustness.

Global configuration. All models are trained for 300 epochs. The optimization is performed using SGD with a momentum of 0.9 and a weight decay of 0.0005, together with a cosine annealing learning rate scheduler combined with an initial linear warm-up. The ratio of excitatory to inhibitory neurons is fixed at 4:1 across all layers. No dropout is applied. We use the standard cross-entropy loss. The final prediction is obtained by averaging the output logits of the classifier across all simulation time steps before computing the loss. For performance evaluation, we report the best top-1 accuracy achieved on the validation set throughout the training process.

Task-specific configuration. To accommodate the varying complexities and temporal dynamics of different datasets and network architectures, we finetune some hyperparameters for each specific task. The detailed task-specific configurations are summarized in Table 5.

Table 5: Task-specific hyperparameter configurations.

Dataset	**Architecture**	**Batch Size**	**Time Steps**	**Peak LR**	**Warm-up Epochs**
CIFAR-10	VGG-8/11	128	4	0.002	10
	VGG-16/19, ResNet-18	128	4	0.001	30
CIFAR-100	VGG-16/19	128	4	0.001	30
TinyImageNet	ResNet-18	128	4	0.003	10
CIFAR10-DVS	VGG-8/11	32	10	0.001	10
DVS-Gesture	VGG-8	32	16	0.001	10