脉冲深度学习梯度替代算法研究综述

方 维[1] 朱耀宇[2] 黄梓涵[3] 姚 满[4] 余肇飞[5] 田永鸿[1),3),6)]

[1)](北京大学深圳研究生院信息工程学院 广东 深圳 518055)

[2)](中国科学院计算技术研究所 北京 100190)

[3)](北京大学计算机学院 北京 100871)

[4)](中国科学院自动化研究所 北京 100190)

[5)](北京大学人工智能研究院 北京 100871)

[6)](鹏城实验室 广东 深圳 518000)

摘 要 被誉为第三代神经网络模型的脉冲神经网络(Spiking Neural Network, SNN)具有二值通信、稀疏激活、事件驱动、超低功耗的特性,但也因其复杂的时域动态和离散不可导的脉冲发放过程而难以训练。近年来以梯度替代法和人工神经网络(Artificial Neural Network, ANN)转换SNN方法为代表的深度学习方法被提出,大幅度改善SNN性能,形成了脉冲深度学习这一全新领域。本文围绕梯度替代法的研究进展,对其中的基础学习算法、编码方式、神经元和突触改进、网络结构改进、正则化方法、ANN辅助训练算法、事件驱动学习算法、在线学习算法以及训练加速方法进行系统性地回顾和综述,并选择其中的代表性方法进行实验对比分析,讨论了目前的研究挑战和可能的解决方案,最后展望了未来可能取得突破的研究方向。

关键词 脉冲神经网络;梯度替代法;类脑计算;神经形态计算;脉冲深度学习

中图法分类号 TP18 **DOI号** 10.11897/SP.J.1016.2025.01885

Review of Surrogate Gradient Methods in Spiking Deep Learning

FANG Wei[1)] ZHU Yao-Yu[2)] HUANG Zi-Han[3)] YAO Man[4)]

YU Zhao-Fei[5)] TIAN Yong-Hong[1),3),6)]

[1)](School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, Shenzhen, Guangdong 518055)

[2)](Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

[3)](School of Computer Science, Peking University, Beijing 100871)

[4)](Institute of Automation, Chinese Academy of Sciences, Beijing 100190)

[5)](Institute for Artificial Intelligence, Peking University, Beijing 100871)

[6)](Peng Cheng Laboratory, Shenzhen, Guangdong 518000)

Abstract Neuromorphic computing is an emerging research area inspired by the structure and function of biological neural systems, designs brain-inspired software algorithms and hardware chips. This research paradigm has achieved remarkable progress including the vision sensors (the dynamic vision sensor, the Vidar spike camera, etc.), the computing chips (IBM True North, Intel Loihi, and Tsinghua Tianjic, etc.), and Spiking Neural Networks (SNNs). Inspired by

收稿日期:2024年8月29日;在线发布日期:2025年3月7日。本课题得到国家自然科学基金杰出青年科学基金(No. 62425101)、国家自然科学基金重点项目(No. 62332002)、国家自然科学基金重大科研仪器研制项目(No. 62027804)、国家自然科学基金科学中心项目(No. 62088102)、国家自然科学基金青年科学基金(No. 62406322)资助。 **方 维**,博士,助理研究员,主要研究领域为脉冲神经网络。E-mail:fwei@pku.edu.cn。 **朱耀宇**,博士,特别研究助理,主要研究领域为类脑计算。**黄梓涵**,博士研究生,主要研究领域为脉冲神经网络。**姚满**,博士,助理研究员,主要研究领域为神经形态计算。**余肇飞**,博士,助理教授,主要研究领域为神经形态计算和计算神经科学。**田永鸿**,博士,教授,国家杰出青年科学基金入选者,IEEE Fellow,中国计算机学会(CCF)高级会员,主要研究领域为视频大数据分析处理和类脑计算。本文的实验代码、训练日志可以从如下网址获取:https://github.com/fangwei123456/chinese_snn_surrogate_gradient_review。

biological neural systems, SNNs are regarded as the third generation of neural network models with binary communication, sparse activation, event-driven computations, and power-efficient characteristics. SNNs can achieve up to several orders of magnitude lower energy consumption in asynchronous neuromorphic computing chips, making them a promising alternative to Artificial Neural Networks (ANNs) for addressing the significant energy demands of current ANN-based Artificial Intelligence (AI) systems. However, the training of SNNs is challenging because of their complex temporal dynamics and non-differentiable firing mechanisms, resulting in the large performance gap between SNNs and ANNs, which restricts the practical value of SNNs. Recently, deep learning methods, including the surrogate gradient methods and the ANN to SNN conversion methods, have been proposed and have greatly promoted the performance of SNNs. Compared with the conversion methods, the surrogate learning methods have the advantages of low latency and temporal information processing ability, which attract increasing research interest from the neuromorphic community. This article focuses on the surrogate gradient methods and provides a systemic review. Firstly, the history of three generations of neural networks and deep learning is briefly retraced. Then, the basic components and benchmarks of deep SNNs are introduced, including the synapses, spiking neuron models, static datasets, and neuromorphic datasets. After the introduction of the background above, this article categorizes the existing learning methods into the following topics systemically: (1) the basic learning methods; (2) encoding methods; (3) neuron and synapse model modifications; (4) network structure designs; (5) normalization methods; (6) ANN-auxiliary training methods; (7) event-driven learning methods; (8) online learning methods; (9) training acceleration methods. Almost all methods in surrogate learning methods are covered by these topics, which provides a comprehensive and coherent view. Exhaustive experiments are conducted to compare methods from different categories fairly, including the static/sequential CIFAR classification tasks, the neuromorphic Spiking Heidelberg Digits voice recognition task, and the neuromorphic Gen1 and static COCO objection detection tasks. The involved metrics include accuracy, training/inference speed, memory consumption, and synaptic operations. These experimental results assess the existing representative methods from a holistic viewpoint and demonstrate their advantages and drawbacks. Then, the current challenging issues and potential solutions are discussed. Finally, the advantages and shortcomings of each learning method category are concluded, with the suggested research directions to solve the corresponding shortcomings. While the technical roadmap of current high-performance learning methods is primarily shaped by research from deep learning communities-such as Quantized Neural Networks, Recurrent Neural Networks, and Tiny Machine Learning-with the influence of neuroscience diminished, this article suggests that brain-inspired algorithms could represent a significant breakthrough and should be emphasized in future research, which may pioneer a research path distinct from traditional deep learning approaches.

Keywords spiking neural networks; surrogate gradient methods; brain-inspired computing; neuromorphic computing; spiking deep learning

1 引 言

人工智能在近十几年取得了快速发展[1],在图像分类[2-5]、目标检测和跟踪[6-7]、语音识别[8-9]、机器翻译[10-12]、游戏对战[13-14]、聊天机器人[15-17]、图像生成[18-20]等领域获得了巨大成功,引领了新一轮的经济发展和产业变革。在人工智能技术的演进过程中,神经科学提供的视野和灵感起到了重要作用[21-22],最典型的例子莫过于神经网络,其起源于神

经科学，并在人工智能领域作为主要的计算模型。

第一代神经网络又称为感知机（Perceptron）[23]，接收多个输入并输出布尔（Bool）值。感知机通过训练可以解决线性分类问题，引发了第一次神经网络热潮。但感知机不能处理非线性的异或（Exclusive OR，XOR）问题，且训练算法只能用于单层网络，这些缺点使得对神经网络的关注逐渐衰退。第二代神经网络是人工神经网络（Artificial Neural Network，ANN），不再输出布尔值，而是改用Sigmoid等非线性激活输出，结合反向传播算法[24]实现多层网络的构建和训练。ANN解决了异或分类问题，引发了第二次神经网络热潮。但受限于芯片行业的发展，90年代的算力无法支撑大规模神经网络的训练，而小规模神经网络在计算代价、任务性能、可解释性等方面相较于支持向量机[25]等当时人工智能领域的主流方法并不占优，因而对神经网络的研究又逐渐陷入第二次低谷。

脉冲神经网络（Spiking Neural Network，SNN）被誉为第三代神经网络模型[26]，与生物神经元的机制更为相似，拥有积分发放、阈值触发、稀疏激活、脉冲通信的特性。SNN凭借极高的生物可解释性，已经被计算神经科学领域广泛使用[27-29]，用于解释和探究生物神经系统的运行原理。但其复杂的时域动态、离散不可导的脉冲发放过程，使得训练SNN比训练ANN更为困难，导致SNN在任务性能为主要导向的人工智能领域关注度较少。

神经形态计算（Neuromorphic Computing）[30-31]的蓬勃发展为SNN提供了新的机遇。神经形态计算是一种全新的计算范式，旨在借鉴和模仿大脑的运行机理，实现超越传统冯诺依曼架构（Von Neumann Architecture）的全新软件算法和硬件设备，代表性成果包括动态视觉传感器（Dynamic Vision Sensor，DVS）[32]、视达（Vidar）[33]等神经形态视觉传感器和IBM True North[34]、Intel Loihi[35]、达尔文（Darwin）[36]、天机芯（Tianjic）[37]等神经形态计算芯片。SNN被视作神经形态计算领域的主要计算模型，其目标是结合神经形态视觉传感器和计算芯片，充分利用脉冲计算的二值量化、稀疏激活特性，实现感算一体、事件驱动的超低功耗边缘智能（Edge AI）系统[31]。然而，这一设想受限于SNN高性能学习算法的缓慢发展，一度难以实现。

2006年Hinton等[38]使用神经网络在MNIST数据集[39]上击败了基于径向基函数内核（Radial Basis Function Kernel）的支持向量机，以深度学习（Deep Learning）之名拉开了神经网络复兴的序幕[40]。2012年Alex等[41]构建了大规模深度卷积神经网络AlexNet并借助图形处理单元（Graphics Processing Unit，GPU）的强大并行计算能力训练，在ImageNet大规模图像识别挑战赛[42]上取得第一，相较于第二名有着10%正确率的断崖式性能领先，引发了第三次神经网络热潮。深度学习方法以革命般摧枯拉朽的力量将人工智能的各个领域重塑；在这一过程中，以梯度替代法（Surrogate Gradient Method）[43]和ANN转换SNN方法（ANN to SNN Conversion，ANN2SNN）[44]为代表的两大类深度学习方法被提出，并应用于SNN的训练，大幅提升SNN的任务性能至早期ANN的水平[45]，形成了脉冲深度学习（Spiking Deep Learning）这一研究领域。梯度替代法直接训练深度SNN，训练开销大，但获得的SNN时间步数少、延迟低，不局限于频率编码且能够用于神经形态数据分类等时域任务；ANN2SNN方法则是将训练好的ANN转换为SNN，避开直接训练SNN，转换速度快、任务精度高，但通常基于频率编码，时间步数多、延迟高且不能用于时域任务。本文聚焦于直接训练方法，对基于梯度替代法的深度SNN学习算法进行系统性介绍和总结。

图1总结了梯度替代法的发展历程。1990年Mead提出神经形态计算的概念[30]，其后Wolfgang于1997年提出并确立了SNN类脑计算模型[26]。2005年Delbruck团队研制出DVS相机[32]；它是目前最常用的神经形态视觉传感器之一，基于该传感器的神经形态数据集现已在脉冲深度学习中大量使用。2014年IBM研发出基于异步电路实现的事件驱动神经形态计算芯片True North[34]，使用非冯诺依曼架构，芯片能耗密度仅为20 mW/cm^2，相较于能耗密度典型值为50 W/cm^{2}的CPU展示出SNN巨大的能耗优势。2017浙江大学潘纲教授团队研发出国内首个神经形态计算芯片达尔文（Darwin）[36]，并用于手写数字识别和脑机信号识别任务。同年北京大学黄铁军教授团队模仿视网膜中央凹采样模型，研发出积分型脉冲相机视达（Vidar）[33]，其工作原理与脉冲神经元积分发放的特性一致，能够实现高速摄像并重构任意时刻的图像数据。2018年Intel研发出Loihi芯片[35]，并提供了完善的软硬件工具链，被大量研究者用于部署SNN。2019年清华大学施路平教授团队研发出全球首款神经形态异构芯片天机芯（Tianjic）[37]，支持ANN和SNN混合运行，可以充分结合两者的性能和能耗优势。至此，神经形态视

觉传感器和计算芯片已较为完善，脉冲深度学习的数据集和硬件载体已基本构建完成。2019年，施路平教授团队[46]、Zenke等[47]、Shrestha等[48]分别独立提出了通过重定义脉冲发放过程梯度的方式来训练深度SNN的学习算法；该类算法被统称为梯度替代法，与ANN转换SNN算法共同开启了脉冲深度学习时代，其后各类学习算法大量涌现。同年年末，北京大学田永鸿教授团队开源了国际上首批脉冲深度学习框架之一的SpikingJelly框架[49]，填补了深度SNN软件框架的空白。2021年田永鸿教授团队提出Spike-Element-Wise（SEW）ResNet[50]，首次训练出超过100层、最高152层的深度SNN，实现了SNN的残差学习。2023年北京大学朱跃生教授团队提出了首个符合神经形态硬件计算特性的脉冲Transformer架构Spikformer[51]，同年中国科学院自动化研究所李国齐研究员团队提出了Spike-Driven Transformer[52]，标志着Transformer架构开始逐渐在SNN领域推广。2024年，李国齐研究员团队将注意力SNN（Attention SNN）[53]部署到时识科技（SynSense）的异步神经形态感算一体芯片Speck[54]，利用注意力机制将脉冲的稀疏性再次提升，达到了比之前最先进的Intel Loihi芯片的还低的功耗水平。需要声明的是，在脉冲深度学习蓬勃发展的过程中，各类基于梯度替代法的学习算法层出不穷，因图片尺寸有限，难以一一列举，本文仅选取了其中部分影响力较大的代表性工作展示在图1中。

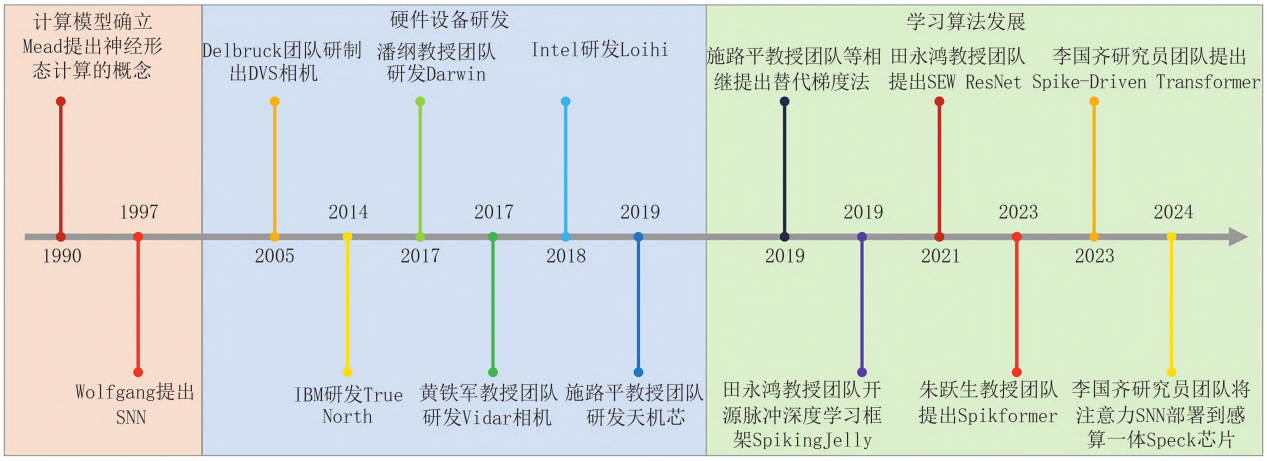


图1 梯度替代学习算法发展历程

本文将在第二章介绍SNN的基本组分和评测基准作为背景知识，随后在第三章对现有的梯度替代法相关学习算法进行系统分类和讲解。在第四章，本文将设置统一的实验环境，对各类学习算法中的代表性方法进行横向对比，公平比较和分析各类方法的性能。在第五章，本文展望了目前的研究挑战与未来研究方向。在第六章，本文对现有方法进行了总结，讨论了这些方法目前的缺陷和对应的改进方法，并展望梯度替代学习算法未来可能的突破点，即结合神经科学的视角见解与深度学习的强大优化能力，设计脑启发的学习算法，实现如大脑般通用的人工智能。

2 深度SNN的基本组分和评测基准

深度SNN通常由多个突触层和脉冲神经元层堆叠而成。SNN的突触层与ANN中的基本一致，主要包括卷积层、池化层、全连接层等。批量标准化（Batch Normalization，BN）[55]和层标准化（Layer Normalization，LN）[56]等正则化层也经常被使用。

SNN的脉冲神经元是其区别于ANN的显著标志，与生物神经系统中的神经元行为更为相似，具有较为复杂的神经动态。生物神经系统中，来自其他神经元的输入电信号通过树突（Dendrite）传递到神经元的胞体，累计为膜电位（Membrane Potential），当膜电位超过阈值（Threshold）电位时，神经元会将累计的电荷在极短的时间内（约为1—2毫秒）一次性释放，形成脉冲（Spike）并通过轴突（Axon）传递到其他神经元。神经元释放脉冲后，膜电位会瞬间降低，这一过程称之为放电后的重置（Reset）。

计算神经科学中构建的脉冲神经元模型对生物神经元进行了精细建模，通常使用一个或多个微分方程去描述其神经动态。例如，SNN中广泛使用的泄露积分发放（Leaky Integrate-and-Fire，LIF）神经

元的阈下神经动态为

$$\tau_m \frac{\mathrm{d}V(t)}{\mathrm{d}t} = -(V(t) - V_{\text{reset}}) + X(t) \tag{1}$$

其中，τ_m是膜时间常数，$V(t)$是膜电位，V_{reset}是静息电位，$X(t)$是输入电流。如果膜电位$V(t)$超过了阈值，则释放脉冲。使用Heaviside阶跃函数$\Theta(x)$描述这一放电过程：

$$S(t) = \Theta(V(t) - V_{th}) \tag{2}$$

其中，$x \geqslant 0$时$\Theta(x) = 1$，$x < 0$时$\Theta(x) = 0$。当神经元释放脉冲后，膜电位瞬间重置到V_{reset}：

$$\lim_{\Delta t \to 0^+} V(t + \Delta t) = V_{\text{reset}} \tag{3}$$

诸如Izhikevich模型[57]等更为精细的脉冲神经元模型通常需要更多数量的微分方程去描述，计算代价较高，因而在深度SNN中较少使用。

对脉冲神经元进行仿真时，一般做法是将连续时间微分方程转换为离散时间差分方程。Fang等[49]使用充电、放电、重置三个方程来构建通用离散时间脉冲神经元模型：

$$H[t] = f(V[t-1], X[t]) \tag{4}$$

$$S[t] = \Theta(H[t] - V_{th}) \tag{5}$$

$$V[t] = \begin{cases} H[t] \cdot (1 - S[t]) + V_{\text{reset}} \cdot S[t], \text{硬重置} \\ H[t] - V_{th} \cdot S[t], \text{软重置} \end{cases} \tag{6}$$

其中，$H[t]$表示充电后、重置前的膜电位，$X[t]$表示输入电流，V_{th}表示阈值，$S[t]$表示释放的脉冲，$V[t]$表示重置后的膜电位，V_{reset}表示重置电压。公式(4)表示神经元的充电方程，f因神经元而异，例如对于LIF神经元，参考其微分方程(1)式，可以得到充电的差分方程为

$$H[t] = V[t-1] + \frac{1}{\tau_m}(X[t] - (V[t-1] - V_{\text{reset}})) \tag{7}$$

公式(5)为放电方程，使用Heaviside阶跃函数来比较膜电位和阈值，并生成二值脉冲。公式(6)为重置方程，目前在脉冲深度学习领域主要存在两种重置方法，分别为硬重置(Hard Reset)和软重置(Soft Reset)。硬重置在释放脉冲后，将膜电位直接设置为V_{reset}，研究者们发现其用于梯度替代法训练的深度SNN性能较好[58]。软重置则是在神经元释放脉冲后，将膜电位减少V_{th}，使用这种重置方式的积分发放(Integrate-and-Fire，IF)神经元在理论上拟合ReLU(Rectified Linear Unit)函数的误差更小[59]，因而在ANN2SNN中普遍使用。

尽管多数深度SNN使用与ANN相同的无状态的突触，但神经元是有状态的，且状态是通过逐步迭代的方式生成，因此SNN相较于ANN引入了时间维度，其处理的输入数据是一个序列。通常用T表示输入序列长度，同时也表示运行SNN所需的时间步数(Time-step)，T有时也称为仿真步数。

脉冲深度学习蓬勃发展，大量实验结果不断涌现，其中静态图像数据集和神经形态数据集分类任务是最频繁使用的性能评测基准。静态图像数据集的“静态”是相较于动态的神经形态数据而言，因图像通常不包括时域信息，每个样本仅为单张图片。常用的静态图像数据集包括MNIST[39]、Fashion-MNIST[60]、CIFAR[61]和ImageNet[42]数据集，数据规模和分类难度依次递增。神经形态数据集是从神经形态视觉传感器直接收集，或软件仿真算法将静态图片转换而得到的事件集合，其中每个事件通常以异步的地址事件协议(AER (Address Event Representation) Protocol)来表示为(x_i, y_i, t_i, p_i)，其中i是事件索引，(x_i, y_i)是事件的横纵坐标，t_i是事件的时间戳，$p_i \in \{-1, 1\}$是事件的极性。神经形态数据集中的事件稀疏但数量众多，一个样本通常包含百万个事件，难以被神经网络直接处理，因而需要通过切片积分等下采样方式转换成帧数据后才能使用[46,49,62]。常用的神经形态数据集包括N-MNIST[63]、CIFAR10-DVS[64]、DVS Gesture[65]、ASL-DVS[66]、N-Caltech101[63]、ES-ImageNet[67]、Spiking Heidelberg Digits (SHD)[68]等。神经形态数据集常用于评估SNN的时域信息处理能力。但Laxmi等[69]等指出N-MNIST等拍摄静态图片得到的神经形态数据集的时域信息较少，因而对于网络的长期依赖学习能力评估，也有一些研究采用序列(Sequential)图像分类[70-71]。在序列图像分类任务中，图像会被从左到右逐列输入，网络在同一个时刻只能看到一列图像，最终的分类性能能够体现网络的记忆能力。

SNN的能源效率是其突出优势，因而网络的能耗也是评估各类学习算法的重要指标。由于进行实际硬件部署成本较高，目前多数研究都采用理论估算的方式汇报能耗结果。具体而言，该方法假设SNN在推理时，对于乘积累加运算(Multiply-Accumulate，MAC)，若参与计算的一方是脉冲，则脉冲为0的位置不需要计算，MAC操作转换为选择脉冲为1的位置进行累加(Accumulate，AC)实现；

对于每种操作的能耗，按照每个MAC操作消耗$E_{MAC}=4.6pJ$，每个AC操作则消耗$E_{AC}=0.9pJ$[72]来估算；不考虑在内存中读写数据带来的功耗；统计网络中所有乘加数量n_{MAC}和加法操作数量n_{AC}，即可得到网络整体能耗为$E_{total}=E_{MAC}n_{MAC}+E_{AC}n_{AC}$。

3 深度SNN的梯度替代训练算法

由于高性能学习算法的缺失，SNN一度只能解决MNIST分类这种简单任务，不具备处理复杂现实世界问题的能力。近年来随着脉冲深度学习方法的相继提出，SNN的性能大幅度提升至实用水平，研究者们甚至成功构建出基于脉冲计算的超低功耗边缘智能系统[37,54,73]。本章将对脉冲深度学习方法中的梯度替代法这一大类算法进行详细介绍，全面梳理现有研究成果和最新进展。

3.1 基础学习算法

反向传播和梯度下降是深度学习中参数优化的核心方法，其依赖于神经网络前向传播使用连续可导的操作。但SNN的脉冲发放过程，即(5)式使用的Heaviside阶跃函数$\Theta(x)$是离散、跃变的，其梯度为冲击函数$\delta(x)$：

$$\Theta'(x)=\delta(x)=\begin{cases}+\infty, x=0\\0, x\neq 0\end{cases} \tag{8}$$

在反向传播中使用$\delta(x)$会破坏正常的梯度传播，使得网络无法训练。

量化神经网络(Quantized Neural Network)领域的研究者，对权重或激活值进行量化时也遇到了类似的问题。量化函数是典型的输入连续、输出离散的函数，其梯度也几乎处处为0。Bengio等[74]提出了直通估计器(Straight-Through Estimator)以解决这一问题，其核心思路是在前向传播时使用量化函数，而在反向传播时重新定义前向传播的梯度。例如四舍五入量化的Round函数及其通过直通估计器定义的梯度可以表示为如下形式：

$$y=\mathrm{Round}(x) \tag{9}$$

$$\frac{\mathrm{d}y}{\mathrm{d}x}=1 \tag{10}$$

在上述定义下，Round函数的梯度被视作恒等函数$y=x$的梯度。

在SNN研究领域，施路平教授团队[46]、Zenke等[47]、Shrestha等[48]在2018年分别独立地提出了梯度替代算法，其思路与直通估计器类似，成为目前直接训练深度SNN算法的基石。梯度替代法在前向传播时使用Heaviside阶跃函数$\Theta(x)$生成二值脉冲，而在反向传播时重定义$\Theta'(x)$为替代函数$\sigma(x)$的导数$\sigma'(x)$。具体而言，(5)式仍然用于前向传播，而其反向传播则按照重定义的梯度：

$$\frac{\partial S[t]}{\partial(H[t]-V_{th})}=\sigma'(H[t]-V_{th}) \tag{11}$$

替代函数$\sigma(x)$通常是连续、光滑的函数，拥有数值正常的导数。常用的替代函数包括Rectangular[46]、SuperSpike[47]、ArcTan[62]、Sigmoid等，这些函数大多单调递增且关于$(x,y)=(0,0.5)$中心对称，值域为(0，1)，可以视作$\Theta(x)$的光滑近似。

除了光滑近似，对于替代函数梯度的另一种解释基于概率性发放脉冲的期望的梯度[48,75]，先前这一思想也被量化神经网络用于解释直通估计器[76]。具体而言，将前向传播视作概率性的脉冲发放：

$$p(s=1)=\sigma(x) \tag{12}$$

$$p(s=0)=1-\sigma(x) \tag{13}$$

其中$\sigma(x)$是Sigmoid等值域为(0,1)的函数，将输入转换为概率。相应的反向传播定义为

$$\frac{\mathrm{d}s}{\mathrm{d}x}\approx\frac{\mathrm{d}(E(s))}{\mathrm{d}x}=\frac{\mathrm{d}(\sigma(x))}{\mathrm{d}x}=\sigma'(x) \tag{14}$$

从而得到了与将$\sigma(x)$视作$\Theta(x)$的光滑近似时完全相同的梯度表达式。

替代函数是深度SNN训练算法的基础组件，对网络性能有着重要影响。Zenke等[43]首先以数值模拟的方式形象展示了替代函数为何能够训练网络，他们在一个两层小网络上对比数值梯度和替代梯度，发现两者相似性较高，表明替代梯度所指示的梯度下降方向与真实梯度接近；数值梯度时而在0和较大值之间跳动，而替代梯度则连续且数值有限，展示出替代梯度平滑稳定的性质。Zenke等[77]进一步研究了替代函数的形状对训练的影响。以SuperSpike替代函数为例，其定义为

$$\sigma'(x)=(\beta|x|+1)^{-2} \tag{15}$$

其中β为形状参数，β越大则梯度越集中于0附近且数值越大，$\sigma'(x)$越接近$\delta(x)$。他们在含有一个隐藏层的SNN上进行学习率和替代函数形状参数的网格搜索实验，发现只要替代梯度不是常数，则训练能达到的最高正确率与形状参数无关，表明梯度替代法对替代函数的形状具有鲁棒性。但他们的这一实验也发现，不同的形状参数对应的能达到最高性能的学习率的范围也不同，如果使用不恰当的形状参数，则需要非常精细地调整学习率才能达到目标

性能。Lian等[78]的研究表明，β直接影响替代梯度函数的宽度，太大的β会导致替代梯度和真实梯度之间误差较大，造成梯度不匹配；而太小的β则使得替代函数的宽度狭窄，造成梯度消失；两者都会导致网络难以训练。Li等[79]则发现不同形状参数下，替代梯度和数值梯度的余弦相似度存在较大差异。另一方面，Zenke等[73]也测试了替代梯度的尺度对学习的影响，他们将式(15)乘上β得到$\beta\sigma'(x)$以改变梯度的尺度，发现不同的β对性能影响较大，这一问题可能是较大的梯度在SNN使用通过时间反向传播(Back Propagation Through Time，BPTT)累乘梯度时引发梯度爆炸所致。目前研究者们通常将替代函数的梯度缩放到最大值为1来缓解梯度爆炸问题，例如使用SpikingJelly框架[49]中的$\sigma(x)=\mathrm{Sigmoid}(4x)$使得$\max\sigma'(x)=\sigma'(0)=1$.

目前也有少量研究去尝试改进替代函数的选择。Li等[79]基于数值梯度和替代梯度的余弦相似度来设置替代梯度函数的超参数，但由于数值梯度计算代价高昂，该方法只被用于网络首层脉冲神经元。Lian等[78]根据膜电位的分布动态调整替代梯度的宽度，避免梯度不匹配或梯度消失问题。Che等[80]对不同替代函数参数在训练时使用Softmax混合，而推理时使用Argmax选择，从而实现可微分参数搜索。

3.2　编码方式

神经编码(Neural Coding)泛指神经元如何将信息表示为电生理活动，是神经科学中的一个重要研究问题。在脉冲深度学习中，该问题细化为如何使用脉冲序列来表示信息，研究话题涵盖如何将非脉冲的输入编码成脉冲，以及SNN内部如何使用脉冲传递信息。从类型来看，编码方式可分为两类：频率编码(Rate Coding)和时间编码(Temporal Coding)，前者使用脉冲的发放频率表示数值的大小，而后者则通过脉冲的发放时刻传递信息。

常见的图像、视频、语音等数据都以整数或浮点值形式存储，与SNN期望的脉冲形式不符，研究者们提出了多种输入编码方式解决这一问题。泊松编码(Poisson Coding)是频率编码的代表性方法。对于输入$x\in(0,1)$，标准的泊松编码生成数量符合强度为x的泊松分布的脉冲，而简化的实现则用每个时间步中脉冲发放概率均为x的二项分布近似。泊松编码在早期的深度SNN研究中[81-82]较多使用。

时间编码方式则通过脉冲的发放时刻来表示信息，首达脉冲编码(Time-To-First-Spike Coding)[83-84]是其中的典型代表。首达脉冲编码遵循“刺激越强，发放越早”的规则，将输入$x\in(0,1)$转换成对应的发放时刻，即

$$S[t]=\begin{cases}1, t=t_f\\0, t\neq t_f\end{cases} \tag{16}$$

$$t_f=\mathrm{Round}((T-1)\cdot(1-x)) \tag{17}$$

其中，Round为四舍五入的量化函数，T为时间步数。

泊松编码对于数值较小的输入很难触发脉冲，随机发放的特性需要较长时间步才能获取稳定结果，因而时间步数多、延迟高、性能低；而首达脉冲编码只能释放单个脉冲，且式(17)中的Round函数损失了一定信息，实际性能也欠佳。目前多数高性能深度SNN[50-52]都采用直接输入编码[85]方式。如果输入是静态的图片，而SNN需要运行T次，则该方法将输入简单重复T次得到输入序列。在该输入方式下，连续浮点输入转换为离散二值脉冲的编码实际由首个突触层和脉冲神经元层完成，它们可以视作可学习的编码器[62]。Rathi等[85]的实验结果表明，尽管直接输入编码在首层引入了浮点计算，但相较于泊松编码，该方法的时间步数大幅度降低，因而最终网络的能耗也低于使用泊松编码。

绝大多数ANN2SNN方法使用频率编码，SNN内部通过脉冲的发放频率表示转换前的ANN中ReLU的激活值；此外，首达脉冲编码[86-88]、相位编码(Phase Coding)[89]和进制编码(Radix Coding)[90]等效率更高的时间编码方式也被逐渐提出。而在基于直接训练的深度SNN中，梯度替代法提供了强大的端到端训练能力，因而无需手动设计网络内部的编码方式。从实际表现看，梯度替代法训练出的SNN所需的时间步数也远少于由转换法得到的SNN，效果较好。但理解直接训练的SNN内部的编码方式，依然是一个值得探索的方向，现有研究较少。Li等[91]对基于梯度替代法训练的SNN和同结构ANN进行了相似性分析，发现SNN的特征与ANN具有高度相似性，时间维度并未提供太多额外信息；Hu等[92]则进而发现不同时间步的梯度相似性也很高；以上研究表明现有的深度SNN内部的编码方式可能较为接近频率编码。需要指出的是，上述研究使用直接输入编码、纯前馈SNN，其结论未必适用时间编码输入或者带有反馈连接的SNN。

3.3　神经元和突触改进

深度脉冲神经网络的主要组分是神经元和突

触，两者均对网络性能有着重要影响，因而有大量研究对其进行改进，提出了多种新型神经元和突触模型以提升SNN性能。

最早的神经动态可学习的神经元模型之一是PLIF神经元（Parametric Leaky Integrate-and-Fire Neuron）模型[62]，其将LIF神经元的膜时间常数τ_m参数化并设置为可学习，阈下神经动态为

$$H[t]=V[t-1]+k(a)\cdot(-(V[t-1]-V_{reset})+X[t]) \tag{18}$$

其中，膜时间常数的倒数，即τ_m^{-1}被重参数化为$\tau_m^{-1}=k(a)$，a是可学习参数。$k(a)\in(0,1)$是限幅函数，确保$\tau_m>1$以防止神经元出现自充电的情况，在实践中通常取$k(a)=\mathrm{Sigmoid}(a)$。PLIF神经元通常设置每一层只有一个可学习参数a，即该层神经元的膜时间常数是共享的，既大幅度减少了参数量，又与生理实验证据中相邻脑区神经元性质类似这一特性符合；而不同神经元层的参数a在训练后不尽相同，保持了神经元的异质性。以往的研究为了减少调参成本，倾向于在整个网络中使用相同的膜时间常数τ_m，丧失了神经元的异质性，并且只训练网络权重，使得网络的表达能力有所下降；PLIF神经元的提出解决了这一问题，并实现了突触权重和神经动态的联合学习。但PLIF神经元在训练完成后与LIF神经元无异，因而其可以视作一种参数化和训练技巧，而非一种新型神经元。

为进一步扩展神经动态的学习范围，GLIF神经元（Gated Leaky Integrate-and-Fire Neuron）[93]被提出，其将神经元对上一时刻的状态衰减、对输入的累计、释放脉冲引发的重置均进行参数化，分别表示为可学习的门控$\mathbb{G}_\alpha$，$\mathbb{G}_\beta$，$\mathbb{G}_\gamma$，具体形式为

$$\mathbb{G}_\alpha=(1-\alpha(1-\tau_{exp}))\cdot H[t-1]-(1-\alpha)\tau_{lin} \tag{19}$$

$$\mathbb{G}_\beta=(1-\beta(1-g[t]))\cdot X[t] \tag{20}$$

$$\mathbb{G}_\gamma=-\gamma\cdot\mathbb{G}_\alpha-(1-\gamma)\cdot V_{\mathrm{reset}} \tag{21}$$

其中，α,β,γ分别是可学习的门控系数；τ_{exp}和τ_{lin}分别表示指数和线性衰减系数；$g[t]$表示随时间变化的突触权重。GLIF神经元也使用了参数共享的技巧，其可学习参数支持设置为逐层或逐通道，因此也几乎不增加网络的参数量。GLIF神经元通过可学习的门控，实现了指数衰减和线性衰减、无状态突触和有状态突触、硬重置和软重置的混叠，因此具有很强的表达能力，但也带来了较大的计算量，相较于传统神经元，其训练速度有着较大下降。

MLF方法（Multi-Level Firing Method）[94]使用多个脉冲神经元构成一个神经元组，组内的神经元使用不同的阈值，并将输出的脉冲累计，相较于传统方法使用的单个神经元，具有更好的拟合能力，但神经元层的输出不再是纯二值脉冲，可能会难以在一些仅支持二值计算的神经形态计算芯片上实现。

LIF神经元的漏电行为可能会导致长期梯度衰减，为解决这一问题，CLIF神经元（Complementary Leaky Integrate-and-Fire Neuron）[95]通过增加补充电位（Complementary Potential）实现跨多个时间步的稳定梯度传播：

$$M[t]=M[t-1]\cdot\sigma(\frac{1}{\tau_m}H[t])+S[t] \tag{22}$$

$$V[t]=H[t]-S[t]\cdot(V_{th}+\sigma(M[t])) \tag{23}$$

其中，$M[t]$表示补充电位，$\sigma(\cdots)$是Sigmoid激活函数。公式（22）表示$M[t]$的更新过程，其自身衰减与膜电位的衰减程度相反，并在神经元释放脉冲，即膜电位瞬间下降时自增，实现了与膜电位的互补。公式（23）基于软重置的（6）式进行修改，引入了$M[t]$使得膜电位能自适应调整，避免过高或过低的发放率。尽管PLIF神经元和GLIF神经元神经动态中都使用了Sigmoid函数，但该函数只用于包装可学习参数，其输出在训练完成后是常数，神经元推理时并不需要计算；而CLIF神经元的式（22）和式（23）中Sigmoid函数的输入是依赖于数据的，不能在推理时去除。Sigmoid函数复杂的指数计算，可能带来CLIF神经元较高的硬件实现代价。

传统神经元皆为串行计算，不能充分利用GPU的大规模并行计算能力加速，是深度SNN训练速度缓慢的一个重要原因。PSN（Parallel Spiking Neurons）[71]是首个并行脉冲神经元模型，其灵感来自传统串行脉冲神经元在不发放脉冲的一段时刻内，膜电位的逐时间步迭代求解可以写成非迭代形式的解析解。受此现象启发，Fang等[71]去除了传统脉冲神经元的重置过程，并发现对于大多数神经元而言，$H[t]$可以表达为输入$X[i]$的线性组合，以此提出了PSN模型，其神经动态为

$$H=WX,\quad W\in\mathbb{R}^{T\times T},X\in\mathbb{R}^{T\times N} \tag{24}$$

$$S=\Theta(H-B),\quad B\in\mathbb{R}^{T},S\in\{0,1\}^{T\times N} \tag{25}$$

其中，X是输入序列，W是可学习权重，H是膜电位，B是可学习阈值，S是输出脉冲，N是神经元数量，T是时间步数。PSN膜电位的生成需要用到所有时刻的信息，而在一些实际任务中，未来信息不可在当下获取，为解决这一问题，Fang等[71]提出Masked PSN，其对式（24）中使用的权重增加掩模，只使用包

括t时刻在内的最新k个输入来生成$H[t]$，具体形式为

$$H=(W\bullet M_k)X,\ W\in\mathbb{R}^{T\times T},M_k\in\mathbb{R}^{T\times T},X\in\mathbb{R}^{T\times N} \tag{26}$$

其中M_k定义为

$$M_k[i][j]=\begin{cases}1, & j\leqslant i\leqslant j+k-1\\ 0, & \text{其他情况}\end{cases} \tag{27}$$

PSN和Masked PSN的权重均是逐时刻的，难以处理变长序列。Fang等[71]进而将Masked PSN的权重设置成时域共享，从而得到Sliding PSN，其神经动态为

$$H[t]=\sum_{i=0}^{k-1}W_i\bullet X[t-k+1+i] \tag{28}$$

$$S[t]=\Theta(H[t]-V_{th}) \tag{29}$$

其中，$W=[W_0,W_1,\ldots,W_{k-1}]\in\mathbb{R}^k$是可学习权重，约定$j<0$时$X[j]=0$，$V_{th}$是可学习的阈值。PSN、Masked PSN、Sliding PSN统称为PSN家族，相较于传统串行神经元，PSN家族无需逐步迭代，可以使用并行度更高的矩阵乘法来计算膜电位，仿真速度大幅度提升；使用直接的权重连接替换传统神经元的基于马尔科夫链的依赖关系，长期依赖的学习能力也得到增强。PSN家族最大的缺陷在于皆为高阶神经元，需要存储多个历史输入，推理的内存消耗会剧增。

与PSN并行化的思路类似的研究还包括随机并行脉冲神经元[96]，其也通过忽略重置来避免膜电位的迭代求解，但脉冲的生成不是直接使用Heaviside阶跃函数，而是采用概率性发放的形式。具体而言，其发放概率由膜电位决定，而梯度则使用替代函数来重新定义。

AMOS（At Most One Spike）神经元只能释放不超过一个脉冲，相较于不做任何限制的普通神经元，更少的脉冲发放次数带来了更低的理论功耗。AMOS神经元通常与首达脉冲编码结合用于ANN2SNN方法[86]，以单个脉冲精确的发放时刻来表示信息。而在SNN的直接训练算法中，AMOS神经元的足迹最早可以追溯到早期的经典SNN有监督学习算法SpikeProp[86]。Mostafa等[97]首次将AMOS神经元用于深度SNN，在训练算法上沿用了之前Mostafa[98]的方法，层之间传递的是脉冲发放时刻，借助于输入和输出脉冲发放时刻的因果（先后）关系来传递梯度，但确定时刻的先后关系需要排序和遍历，复杂度较高。Kheradpisheh等[83]提出的S4NN（Single-Spike Supervised Spiking Neural Network）也使用AMOS神经元，但层之间传递的是脉冲的值，脉冲发放时刻则被隐式地用于通过链式法则定义梯度，不再需要手动进行排序，相较于Mostafa等[97]的方法复杂度大幅度降低，易于实现，且任务性能更好。总体而言，AMOS神经元脉冲数量少的优势非常直观，但研究还处于早期阶段，与传统神经元的性能有着较大差距。

图2对本小节介绍的部分神经元进行了梳理，清晰地展示了神经元改进工作之间的脉络关系。表1总结了部分脉冲神经元改进研究在多个数据集上的时间步数和分类正确率，以“步数|正确率”的形式展示。整体来看，随着神经动态复杂度的提升，神经元的表达能力得到提高，因而网络的任务性能也进一步提升，但这通常也会导致计算代价的增加和训练速度的降低，而神经元的并行化则可能是这一问题的解决途径。需要注意的是，AMOS神经元类方法目前任务性能还较低，并且主要使用MNIST之类的简单数据集评测性能，因而没有列入到表1中进行对比。

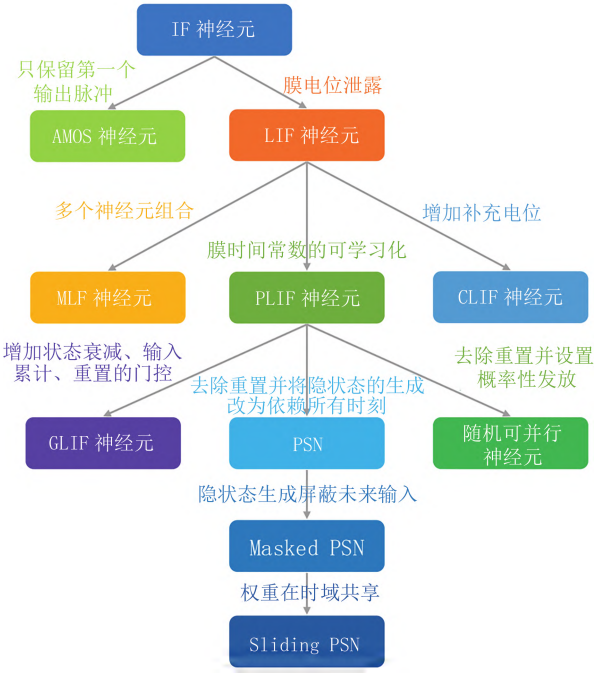


图2　部分神经元改进工作之间的联系

深度SNN中所使用的突触模型通常与深度ANN中相同，但也有一些研究者对突触进行了更精细的建模，引入额外的时域动态或突触延迟。Fang等[99]将常用的无状态的突触更改为由差分方程描述的有状态突触，使得突触也具有了一定的记忆，增强

表1 脉冲神经元分类任务时间步数和正确率

神经元\数据集	CIFAR10	CIFAR100	ImageNet	DVS Gesture	CIFAR10-DVS
PLIF	8|93.50			20|97.57	20|74.80
GLIF	2|94.44 4|94.85 6|95.03	2|75.48 4|77.05 6|77.35	4|67.52 6|69.09		16|78.10
MLF	4|94.25			40|97.29	10|70.36
CLIF	4|96.01 6|96.45 8|96.69	4|79.69 6|80.58 8|80.89			
PSN家族	4|95.32		4|70.54		4|82.30 8|85.30 10|85.90

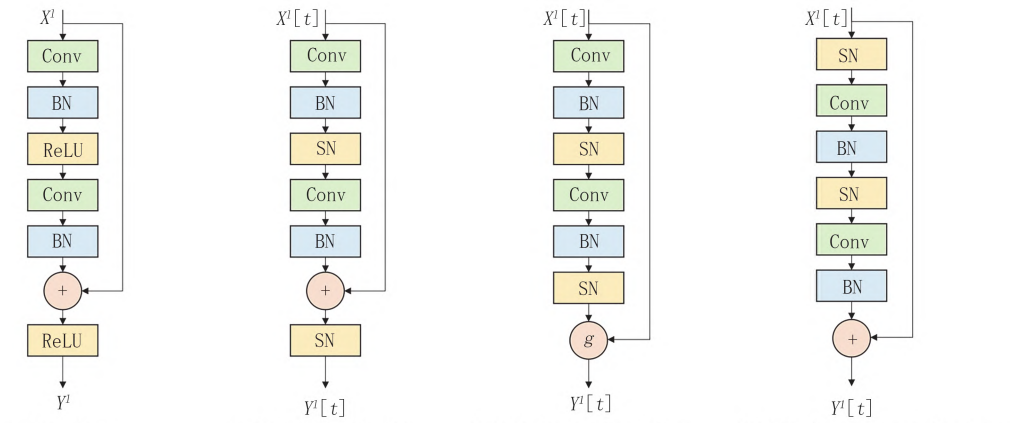
了整个网络在记忆任务上的学习能力。 Ilyass等[100]通过时间步维度上的扩张卷积来移动脉冲发放的位置，从而对突触延迟进行建模，同时使得突触延迟也参与到网络的训练，在时域任务上以更少的参数超越了传统方法的性能。但这些方法都使得突触的复杂度大幅度提升，网络的训练速度下降、内存消耗激增，因而尚未应用于大规模深度SNN。

3.4 网络结构改进

网络结构改进一直是深度学习领域的热门研究方向。ANN领域已有诸多成熟的网络结构，但它们在设计时并未考虑神经形态计算的特性，直接用于SNN会引发性能退化问题，因而脉冲深度学习领域的相关研究主要集中于对已有网络结构的脉冲化改进。

梯度替代法的出现使得SNN研究者能够训练中等规模的深度脉冲卷积网络。然而，研究者们发现若继续采用简单堆叠卷积层的方式来增加网络规模，则性能难以继续提升。研究者们开始考虑构建基于残差连接的深度SNN解决上述问题。残差连接起源于ResNet[4]，如图3(a)所示，是现代深度神经网络结构中不可缺少的一部分。而Spiking ResNet是ResNet的SNN版本，最早用于ANN转换SNN[101]并取得了较好的效果，其结构如图3(b)所示。但是，如果直接将ResNet的残差结构沿用至SNN中(即Spiking ResNet)，在训练十几层的网络时即出现性能退化[102]，表现为更深的模型相较于浅层模型，具有更高的训练集误差。 Fang等[50]从恒等变换和梯度传播角度进行分析，发现Spiking ResNet难以实现恒等变换、易于引发梯度消失或梯度爆炸，因此无法有效加深SNN以获取性能增益。为解决这一问题，Spike-Element-Wise (SEW) ResNet[50]被提出，残差块结构如图3(c)所示，其将脉冲神经元的位置调换到残差连接之前，然后使用一个逐元素操作函数g来实施残差连接，其中g可以是加法、乘法、取反后再乘法等。SEW ResNet在ImageNet数据集上进行了验证，实验结果证实了模型性能随深度稳定增加，首次实现了SNN中的残差学习，并将SNN规模扩大至数百层。Membrane-based Shortcut (MS) ResNet[103]是另一种能够实现恒等变换的脉冲残差连接方式，其将每个残差块中第一个脉冲神经元的输入和最后一个BN层的输出进行连接，结构如图3(d)所示，实现了神经元膜电位层次的残差学习，同样能够将SNN规模扩大至数百层。

SEW ResNet和MS ResNet都解决了深度SNN的退化问题，但同时也引发了新的问题。具体而言，SEW ResNet主要使用性能最好的加法来连接残差块的输入和最后一个SN的输出脉冲，导致



(a) ResNet (b) Spiking ResNet (c) Spike-Element-Wise ResNet (d) Membrane-based Shortcut ResNet

图3 常见的残差块结构

残差块输出的实际上是脉冲之和，是非负整数而非二值脉冲，这可能丧失了SNN的二值特性以及对应的硬件实现时免乘法器的优势；MS ResNet则是使用残差连接在网络层之间传递稠密的浮点值，破坏了SNN事件驱动通信的特性，难以在异步芯片实现。

在ResNet中添加额外的注意力（Attention）模块能够提升神经网络的全局建模能力，从而有效提升任务性能[11,104-105]。这一做法在Spiking ResNet中同样有效。Yao等[106]提出了时域注意力（Temporal-wise Attention）机制，将输入在宽、高和通道维度上进行平均后，送入由2层多层感知机（Multilayer Perceptron，MLP）组成的小网络处理，并输出注意力分数，然后与不同时刻输入再进行点乘。这个额外插入的2层MLP网络就是注意力模块，起到辅助提取全局信息的作用。通过设计更高效的注意力模块[107-109]，或者将注意力机制应用于时间、空间、通道等多个维度[53,110]，SNN在各种任务中的性能得到显著提升。值得一提的是，与ANN相比，受益于事件驱动计算特性，在SNN中增加额外的注意力模块通常会使得整个网络的能耗进一步降低。Yao等的一系列工作[53-54,107-108]以Spiking ResNet为例，对这一特性进行了深入研究和应用。Spiking ResNet包含了循环和卷积两种基本操作，这可以提升参数在时间和空间上的利用效率，但也使得SNN具有“时空不变性[111]”，导致较差的全局建模能力[112]。与此同时，Spiking ResNet的时空不变性还会引入大量的噪声冗余特征[107]。注意力模块能够有效抑制SNN中的噪声脉冲，同时优化正常特征，因此能够在带来性能提升的同时显著降低能耗。注意力SNN的功能在边缘计算芯片上也得到了验证[54,113-114]。特别是，将注意力SNN部署到时识科技（SynSense）的异步神经形态感算一体芯片Speck[54]后，实测数据显示，在DVS128 Gesture数据集上，注意力机制能带来9%的性能提升，同时平均功耗由9.5 mW降低至3.8 mW。

Transformer[105]是继ResNet之后的影响力最大的网络结构，自提出以来便在多个领域刷新了性能指标，成为目前人工智能领域最常用的网络架构之一，其核心机制包括多头自注意力（Multi-Head Self Attention）和位置编码（Positional Encoding）等。传统的卷积神经网络结构中，除去神经元自身的运算，其余的矩阵运算发生在突触和激活值之间。在网络结构脉冲化后，参与矩阵运算的一方为脉冲，可以实现事件驱动、无乘法器的计算。而在Transformer中，除突触和激活值的运算外，自注意力机制中的矩阵乘法使用的是未经过激活函数作用的全连接层的原始输出，且其中还使用Softmax函数；前者涉及稠密浮点值的矩阵乘法，后者则需要指数运算，这些特性难以与神经形态计算芯片兼容。此外，位置编码通常需要浮点值，也与SNN的二值特性违背。因而如何解决上述问题，有效结合Transformer架构的高性能和SNN的低功耗，引起了脉冲深度学习领域内学者们的广泛兴趣。较早的Spiking Transformer[115-117]将Transformer中的部分人工神经元改为脉冲神经元，并保留诸如自注意力机制、归一化等关键操作来保证任务精度。这些Spiking Transformer架构事实上是ANN与SNN融合的异构设计，难以真正发挥出SNN低功耗的优势。脉冲深度学习领域的研究者们意识到发挥Spiking Transformer潜力的关键是如何设计脉冲自注意力算子，并围绕这一问题进行了大量改进。图4展示了目前Spiking Transformer中主流的自注意力机制。

Zhou等[51]指出，自注意力使用的浮点矩阵乘法以及Softmax激活涉及的指数运算难以在神经形态芯片上实现。为此，Zhou等[51]提出了Spikformer，使用脉冲自注意力（Spiking Self Attention，SSA）机制，如图4(a)所示。对于脉冲神经元的输出$Q[t],K[t],V[t]\in\{0,1\}^{N\times n_{head}\times n\times d}$，其中$N$表示批量大小、$n_{head}$表示注意力头数、$n$表示分块（Patch）的数量、$d$表示嵌入（Embedding）的维度，则SSA按照如下形式计算注意力分数$score$：

$$score[t]=\mathrm{SN}(Q[t]K[t]^{\mathrm{T}}V[t]\cdot s) \tag{30}$$

其中，s是缩放因子，SN表示脉冲神经元层。SSA的两个矩阵乘法参与方都至少包含一个脉冲矩阵，而缩放因子则可以被吸收进脉冲神经元层的阈值；相较于原始的注意力，在SSA中Softmax激活被去掉了，可以根据n和d来选择先计算$Q[t]K[t]^{\mathrm{T}}$或$K[t]^{\mathrm{T}}V[t]$以降低复杂度至$\min(\mathcal{O}(n^2d),\mathcal{O}(nd^2))$。

Yao等[52]进一步提出了脉冲驱动Transformer架构，其核心是脉冲驱动自注意力（Spike-Driven Self Attention，SDSA）机制，如图4(b)所示，其延续了SSA不使用Softmax激活的设计，但使用逐元素乘法替代矩阵乘法，同时去除了自注意力机制中的归一化操作：

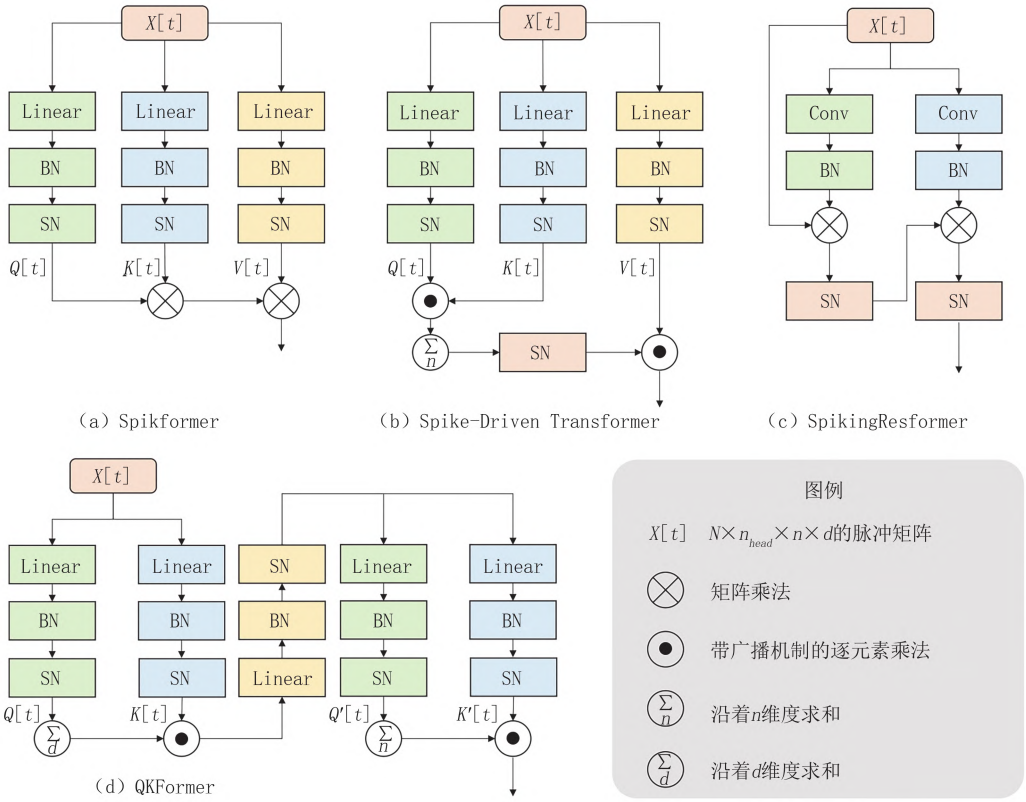


图4 深度SNN中的自注意力机制

$$score[t]=\mathrm{SN}(\sum_{n}(Q[t]K[t]))\cdot V[t] \tag{31}$$

其中$\sum_{n}(\cdots)$表示沿着分块(Patch)的维度求和。需要注意的是,由于脉冲形状满足$\mathrm{SN}(\cdots)\in\{0,1\}^{N\times n_{head}\times d}$,而膜电位$V[t]\in\{0,1\}^{N\times n_{head}\times n\times d}$,两者的逐元素乘法使用了广播(Broadcast)机制。SDSA算子的计算复杂度降低至$\mathcal{O}(nd)$,同时完全消除了乘法,从而使得整个脉冲驱动Transformer中仅有稀疏加法。

SpikingResformer[118]使用了双脉冲自注意力机制(Dual Spike Self Attention, DSSA),如图4(c)所示。这种注意力机制使用了双脉冲变换(Dual Spike Transformation, DST)算子来替换Transformer中的浮点矩阵乘法:

$$\mathrm{DST}(X,Y;f(\cdot))=Xf(Y)=XYW \tag{32}$$

$$\mathrm{DST}_{\mathrm{T}}(X,Y;f(\cdot))=Xf(Y)^{\mathrm{T}}=XW^{\mathrm{T}}Y^{\mathrm{T}} \tag{33}$$

其中$f(\cdots)$是Y上的广义线性变换,可以是无偏置的线性层、卷积层等。Shi等[118]证明了这种算子是脉冲驱动的,并且可以用这种算子替换Transformer中的浮点矩阵乘法。利用DST算子,DSSA按照如下形式计算注意力分数:

$$\mathrm{AttnMap}(X[t])=\mathrm{SN}(\mathrm{DST}_{\mathrm{T}}(X[t],X[t];f(\cdot))\cdot c_1) \tag{34}$$

$$score[t]=\mathrm{SN}(\mathrm{DST}(\mathrm{AttnMap}(X[t]),X[t];f(\cdot))\cdot c_2) \tag{35}$$

$$f(X[t])=\mathrm{BN}(\mathrm{Conv}_p(X[t])) \tag{36}$$

其中,c_1,c_2是缩放因子,BN是批归一化层,Conv_p是卷积核大小和步长为p的卷积。SpikingResformer最大的成功之处在于将自注意力机制巧妙融合进传统卷积架构,为SNN结构设计打开了新思路。

QKFormer[119]如其名字所暗示,只使用$Q[t]$,$K[t]$,并通过融合不同维度来提取信息,其结构展示在图4(d)。QKFormer首先使用token维度元素之和作为通道维度的掩码来提取特征:

$$score[t]=K[t]\cdot\mathrm{SN}(\sum_{d}Q[t]) \tag{37}$$

其中$K[t]\cdot\mathrm{SN}(\cdots)$用到了广播机制。QKFormer进而用通道维度元素之和作为token维度的掩码:

$$score'[t]=K'[t]\cdot\mathrm{SN}(\sum_{n}Q'[t]) \tag{38}$$

其中$Q'[t]$,$K'[t]$不同于原来的$Q[t]$,$K[t]$,经过了额外的全连接层处理,如图4(d)所示。QKFormer只涉及逐维度求和与逐元素乘法,不使用矩阵乘法,注意力机制的复杂度和脉冲驱动的自

注意力类似，也低至$\mathcal{O}(nd)$。

当脉冲化的自注意力机制被成功实现后，研究者们不再使用原有的ResNet等卷积架构作为网络骨架，而是使用Transformer类网络架构，但ResNet中分多个阶段(Stage)处理输入的设计仍然得到了保留。Spikformer[51]和Spike-Driven Transformer[52]使用Compact Convolutional Transformer[120]的网络架构。SpikingResformer[118]使用了3阶段的层级结构以提取不同尺度的特征，并在2层MLP之间插入分组卷积层以提取局部特征。Spike-driven Transformer V2[121]专门设计了Meta Transformer块，由带残差连接的Token维度的脉冲驱动的自注意力[52]和通道维度的MLP组成；在网络架构层次，前2个阶段使用带残差连接的大感受野的7×7可分离卷积和小感受野的3×3的普通卷积组成的卷积块，而在后2个阶段使用Meta Transformer块。QKFormer[119]则是使用类似Swin Transformer[122]的网络结构。

图5展示了本小节中介绍的部分网络结构改进研究之间的递进关系。整体来看，目前新的研究集中于Transformer架构改进，但其中也使用了大量来自ResNet相关研究基础。图6对比了常见深度SNN架构在ImageNet数据集的分类正确率、功耗和参数量。除MS-ResNet外，其他网络均使用时间步数$T=4$；默认使用224×224的图片分辨率进行推理，但也有部分研究者额外汇报了使用288×288

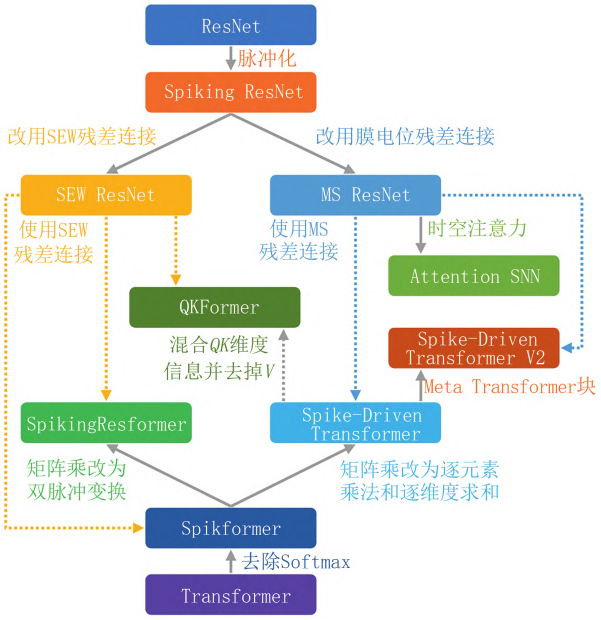


图5　部分网络结构改进研究的联系

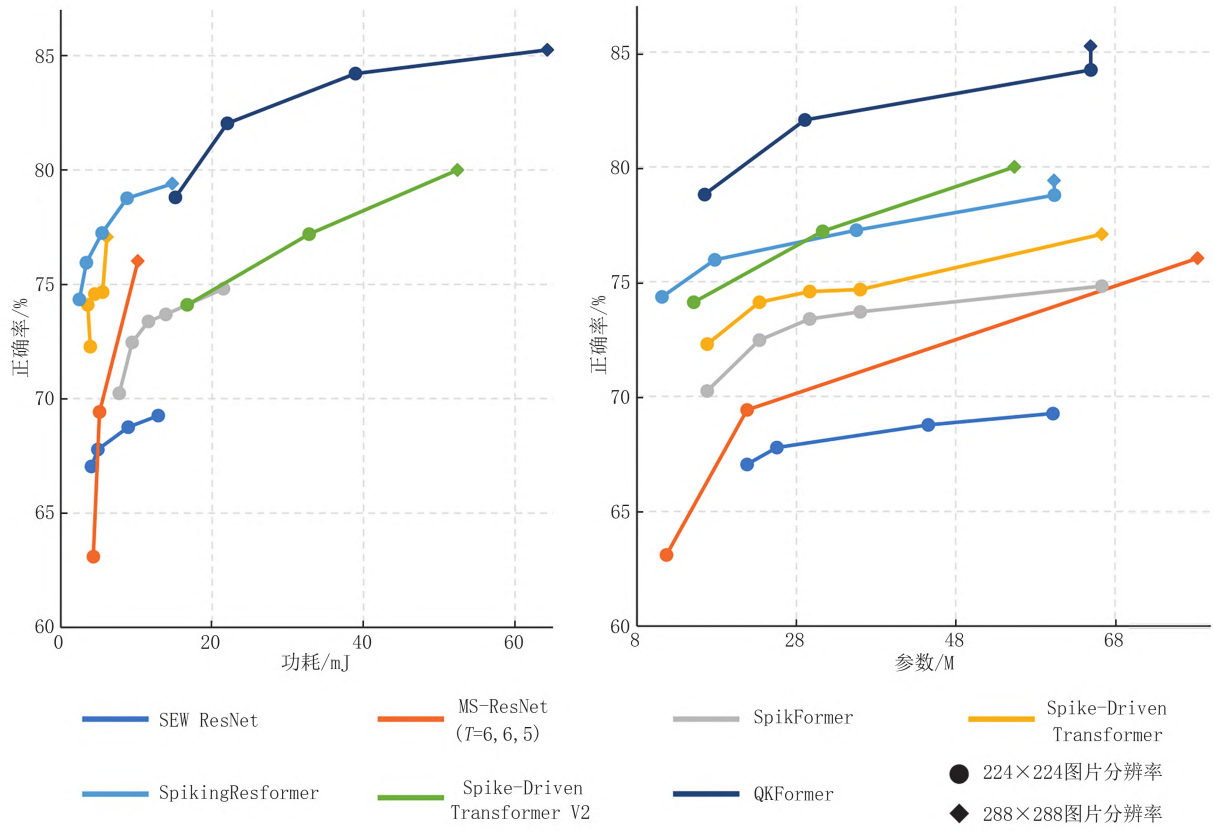


图6　常见深度SNN架构在ImageNet数据集的分类正确率、功耗和参数量

图片分辨率推理的结果，在图中以方形点进行了标注。图6的结果表明，随着残差结构、自注意力机制的引入，深度SNN的性能得到进一步提升，在ImageNet数据集上已经达到85%的正确率，同时能耗和参数量也不断优化，新的网络架构向着正确率更高且功耗和参数量更低的方向迅猛发展。

除最为核心的自注意力机制外，也有少量研究者探讨了Transformer中其他机制的改进。Lv等[123]受到人脑中枢模式发生器工作原理的启发，设计了基于正弦函数的位置编码方式，其产生二值的编码值，避免了传统位置编码带来的浮点计算，并提升了网络的时间序列处理性能。Zhou等[124]对脉冲Transformer的混合专家模型进行了研究，提出了适用于SNN的脉冲专家混合机制（Spiking Experts Mixture Mechanism，SEMM）。该研究指出，ANN中由于混合专家模型采用Softmax计算路由权重和Top-K，硬稀疏地选取专家，并不适用于SNN事件驱动的计算和动态稀疏激活的特性。为了解决该问题，SEMM将每个注意力头视为独立专家，并使用脉冲路由模块对其进行稀疏激活，在集成多头注意力的同时达到了动态稀疏激活的效果。

除手动设计网络结构外，也有研究者将神经网络结构搜索（Neural Architecture Search，NAS）技术引入SNN，实现自动化的模型设计。Na等[125]首次将NAS用于SNN。该方法使用超网训练和遗传算法来优化SNN框架，为了解决搜索方法带来的计算成本，还使用了遗传算法中常见的One Shot Weighting Sharing方法[126-127]，将权重在不同候选框架中共享；同时还提出了Spike-Aware优化方程，通过为发放更多脉冲的候选框架赋予更低的评分作为惩罚，从而限制脉冲数量。Kim等[125]把搜索空间延拓到前向和反馈连接，同时提出了Sparsity-Aware Hamming Distance（SAHD）作为指标去评估SNN框架，通过使用该指标避免了NAS方法中最耗时的超网训练过程，提升了搜索速度。Che等[80]把搜索空间延拓到层与细胞（Cell）维度，使其能够应用于深度估计等稠密预测领域，同时把替代函数也纳入搜索空间，优化替代函数的梯度。该方法首次把可微分网络结构搜索方式引入SNN，只训练代理参数来搜索网络结构，提升了训练速度和精度。Shen等[128]更关注于搜索的生物可解释性，提出了脑启发的神经电路演化策略。此策略搜索空间包括反馈连接、兴奋性和抑制性神经元[129]，以及基于脉冲时间依赖可塑性（Spike-Timing-Dependent Plasticity，STDP）的局部学习方法。同时该方法进一步将任务拓展到了强化学习，取得了与ANN相当的性能。为了更好地在准确性与计算成本之间权衡，Yan等[130]采用了单路径NAS方法，即将所有候选框架编码在一个无分支的脉冲超网中。该框架不再训练不同大小的独立卷积核，而是训练一个无分支的超网卷积核，显著降低了计算成本和搜索时间。整体来看，网络结构搜索类方法本身训练开销较大，与需要多个时间步运行的SNN结合后问题更为明显，已有的研究也多聚焦于解决计算成本问题。

受益于SNN网络结构的快速发展，目前梯度替代法也逐渐应用于难度较高的目标检测任务，这一任务曾经主要由ANN2SNN方法主导[131-132]。目标检测任务通常使用mean Average Precision（mAP）作为性能指标，其中常用的mAP@0.5:0.95表示按照Intersection over Union（IOU）阈值从0.50到0.95，以0.05为步长计算出的平均mAP；mAP@0.5表示按照IOU为0.5计算出的mAP。常用的数据集包括传统的静态图片目标检测数据集COCO[133]和神经形态的Gen1[134]数据集等。Lo¨ic等[135]最早成功将梯度替代法训练的Spiking DenseNet[136]用于Gen1数据集上的目标检测任务。他们的主要创新之处在于，首先在神经形态的NCAR[137]数据集上进行分类任务的预训练，然后再到Gen1数据集上进行目标检测任务的训练。Zhang等[138]在SEW残差连接[50]上增加了二元门控，使得残差块可以完全变成堆叠的卷积层或恒等变换，保持纯二值输出；他们还在SNN的输出和检测头之间增加了注意力模块以更好地提取时空特征。Su等[139]对SNN中的残差结构进行改进，使用最大池化替换原来下采样残差块中的步幅大于1的卷积，通过拼接（Concat）操作实现残差连接，同样保持了纯二值输出。Yao等[121]提出的Spike-Driven Transformer V2在前文已有介绍，该网络结构也在目标检测任务上进行了验证。他们首先在ImageNet数据集上预训练，然后再到COCO数据集进行目标检测训练，最终仅使用1个时间步就超越了前述其他研究方法。Fan等[140]基于Lo¨ic等[135]的方法，将网络中间层输出的不同尺度的特征，使用类似特征金字塔（Feature Pyramid）[141]的方式进行融合，大幅度提升了性能。表2中对这些方法以“时间步数|性能”的格式进行了统计汇总。

近年来还有一些研究将SNN处理的数据类型从图片和神经形态数据集扩展到图（Graph）、

表2　梯度替代法训练的SNN目标检测时间步数和性能

数据集	COCO		Gen1	
方法	mAP@0.5	mAP@0.5:0.95	mAP@0.5	mAP@0.5:0.95
Lo¨ic等[135]		5|0.19		5|0.19
Zhang等[138]	4|0.296			
Su等[139]	4|0.501		5|0.547	5|0.267
Yao等[121]	1|0.512			
Fan等[140]			5|0.593	5|0.321

点云(Point Clouds)等，图卷积神经网络(Graph Convolutional Network，GCN)、PointNet等架构的脉冲版本被相继提出。Gu等[128]针对触觉数据构建图结构，并使用图卷积预处理，特征由LIF神经元转换为脉冲，继而输出到使用LIF神经元的MLP，是较早的成功将SNN结合GCN的研究。Zhu等[129]提出的脉冲卷积图神经网络，则是使用频率编码来将图卷积得到的特征转换为脉冲，且MLP中使用三值激活的LIF神经元，在4个引文网络公共数据集上取得比部分经典GCN模型更好的性能。Li等[142]使用了自适应阈值的LIF神经元，并将其用于聚合邻域信息；LIF神经元动态的放电机制使得不同时刻的图结构也呈现动态变化，网络因此能够捕捉动态图的时序信息。Guo等[143]将PointNet中的激活函数换成LIF神经元以构建脉冲PointNet，并通过单步训练和多步推理以降低训练开销，同时通过训练时随机初始化膜电位的方式来降低其和多步推理时的性能差异。总体而言，将SNN推广至其他数据类型的网络结构，通常能带来理论上的激活值存储开销和推理能耗的降低，且脉冲神经元的特性使得网络具备一定时序信息提取能力，而主要挑战则包括梯度跨越多个时间步的衰减、BPTT带来的较大的训练开销，以及图卷积、点云采样等操作难以在现有神经形态计算芯片上实现等问题。

3.5　正则化方法

正则化方法已经在神经网络优化过程中大量使用，其中批量标准化(Batch Normalization，BN)[55]是SNN中最为广泛使用的方式。相较于层标准化(Layer Normalization，LN)[56]等其他的正则化方法，BN层常用于卷积层之后，并且可以在推理阶段与卷积层融合，无需额外的资源进行实现，因此在SNN中备受青睐。除ANN中已有的正则化方法外，一些专用于SNN的正则化方法也被提出，其中多数方法基于BN进行改进。少数则是针对脉冲神经元的特性设计，它们进一步提升了网络的训练效果。

NeuNorm[144]专用于脉冲卷积层，作用于脉冲神经元的输出，对于每层神经元，额外记录每个位置(i,j)在所有通道的脉冲发放次数之和，并随时间步进行移动平均来持续更新：

$$O_{norm}[t][i][j]=k_{decay}\cdot O_{norm}[t-1][i][j]+\frac{1-k_{decay}}{C^2}\cdot\sum_{c=0}^{C-1}O[t][c][i][j] \tag{39}$$

其中，k_{decay}是衰减因子，C是通道数，$O[t][c][i][j]$是c通道位置为(i,j)处的神经元在t时刻的输出，而$O_{norm}[t][i][j]$则是NeuNorm正则化项，该层传递给下一层的输出会减去该正则化项。该方法与视网膜细胞特定位置的响应会被邻近细胞进行正则化的行为类似[145-146]，有一定的生物可解释性。NeuNorm的效果可能来源于两方面，一是对脉冲进行了时间上的指数移动平均，使得输出更为平滑，避免了单个时刻过低或过高的发放率造成的扰动；二是使脉冲神经元层的输出变成浮点值形式的发放率，相较纯二值脉冲携带了更多信息。但NeuNorm无法融合进突触层，使用后会引入浮点操作，可能是这一原因导致其逐渐被BN类方法取代。

在具有时间维度的SNN中直接使用静态ANN中普通的BN层可能会引发问题，研究者们对此进行了探究并提出了多种改进的BN变体。表3对目前深度SNN中的BN类方法进行了总结和对比，其中μ、σ^2表示均值和方差统计量，γ、β表示仿射变换的权重和偏置，ρ表示动量系数。

普通的BN在SNN中使用时，其训练时会在每个时间步都计算当前t时刻输入的均值$\mu[t]$和方差$\sigma^2[t]$并进行标准化；而在推理时则是利用训练时的统计量来对推理输入标准化。需要注意的是，BN在训练时每次前向传播后都会按照动量的方式来更新均值和方差统计量。记在本次训练前均值和方差统计量分别为μ_k，$\sigma^2{}_k$，其中下标k表示统计量更新次数，则经过本次训练后，BN实际上进行了T次统计量的动量更新并得到μ_{k+T}，$\sigma^2{}_{k+T}$，展示在表3中。BN层通常还设置可学习的仿射变换，其权重和偏置项分别是β，γ，由梯度下降更新。

原始的BN这种随着时间步来动量更新统计量的方式可能并不准确。阈值依赖的BN(Threshold Dependent Batch Normalization，TDBN)[102]解决了这一问题，其将输入在时间维度上进行融合，直接计算整个序列的均值和方差，因而处理完一个序列后，

表3 深度SNN中的批量标准化方法及变体

方法	$t=0$	$t=1$	…	$t=T-1$	统计量更新
BN	$\mu[0],\sigma^2[0]$	$\mu[1],\sigma^2[1]$		$\mu[T-1],\sigma^2[T-1]$	$\mu_{k+T}=(1-\rho)^T\mu_k+\sum_{t=0}^{T-1}(1-\rho)^{T-1-t}\rho\mu[t]$ $\sigma^2{}_{k+T}=(1-\rho)^T\sigma^2{}_k+\sum_{t=0}^{T-1}(1-\rho)^{T-1-t}\rho\sigma^2[t]$
TDBN		γ,β μ,σ^2 γ,β			$\mu_{k+1}=(1-\rho)\mu_k+\rho\mu$ $\sigma^2{}_{k+1}=(1-\rho)\sigma^2{}_k+\rho\sigma^2$
BNTT	$\mu[0],\sigma^2[0]$ $\beta[0],\gamma[0]$	$\mu[1],\sigma^2[1]$ $\beta[1],\gamma[1]$		$\mu[T-1],\sigma^2[T-1]$ $\beta[T-1],\gamma[T-1]$	$\mu_{k+1}[t]=(1-\rho)\mu_k[t]+\rho\mu[t],t=0,1,\cdots,T-1$ $\sigma^2{}_{k+1}[t]=(1-\rho)\sigma^2{}_k[t]+\rho\sigma^2[t],t=0,1,\cdots,T-1$
TEBN	$\gamma p[0],\beta p[0]$	μ,σ^2 $\gamma p[1],\beta p[1]$		$\gamma p[T-1],\beta p[T-1]$	$\mu_{k+1}=(1-\rho)\mu_k+\rho\mu$ $\sigma^2{}_{k+1}=(1-\rho)\sigma^2{}_k+\rho\sigma^2$

BN的统计量只会动量更新一次，而不是按照原始BN的方式更新T次。TDBN还根据后续神经元的阈值对标准化后的输出做相应的线性缩放，以此抵消SNN中特有的阈值给权重的尺度带来的影响。考虑到SNN中不同时刻的数据分布可能并不相同，通过时间批量标准化（Batch Normalization Through Time，BNTT）[147]在每个时间步都使用一个独立的BN层，即均值、方差、统计量、仿射变换都是每个时间步一套单独的参数。时域有效批量标准化（Temporal Effective Batch Normalization，TEBN）[148]的思想则是介于TDBN和BNTT之间，其统计整个输入序列的均值和方差，但对每个时刻又设置单独的可学习仿射变换。为减少参数量，TEBN中不同时间步的仿射变换是使用类似于广播机制的方式生成的，其权重和偏置项γ、β只有一套，而每个时间步在使用时则是由可学习参数$p[t]$与γ、β相乘来生成t时刻的仿射变换参数。需要指出的是，BNTT和TEBN均含有逐时刻的参数，暗含输入序列长度固定不可变的要求，这与SNN中参数时域复用的特性不符，导致网络不能直接处理变长序列。

SNN中的正则化层通常被用于卷积层后、神经元前，对脉冲神经元的输入电流进行正则化，但也有例外，例如Guo等[149]对神经元每一步的膜电位也进行批量标准化并取得了性能提升。

正则化方法除使用正则化层外，还包括使用正则化损失和数据增强等。Guo等[150]将神经元释放脉冲的过程视作信息的量化，将神经元膜电位与输出脉冲的均方误差作为网络损失的一部分，以此减少量化误差；Deng等[151]使用每个时间步的输出与目标做交叉熵，然后在不同时间步上进行平均，以此替换传统的先平均每个时间步的输出再做交叉熵的损失，对神经形态数据分类等时域任务有较大的性能提升。数据增强方法通常在训练集样本上施加诸如亮度、尺寸等变换，以提升网络的泛化能力。ANN领域用于静态图片上的数据增强方法已经比较成熟，而Li等[152]则对神经形态数据集的增强进行了探索，表明剪切、旋转、平移等几何变换适用于神经形态数据；而亮度、色彩、锐度、均衡化等色彩变换会改变事件极性、破坏事件流的稀疏性，不适合使用；其研究结论可以作为神经形态数据增强的经验性准则。

3.6 ANN辅助训练算法

一些研究者另辟蹊径，通过ANN辅助的方式来训练高性能的SNN，主要分为两类方法：基于共享权重训练的方法和基于蒸馏的SNN训练方法。

基于共享权重的训练中，Wu等[153]和Kheradpisheh等[154]设计了共享相同权重的SNN网络和ANN网络，以SNN的输出在时间上的累计近似ANN的激活值，通过ANN的反向传播来获取参数梯度并更新共享权重。

具体而言，Wu等[153]提出一种串联学习框架，该框架包括一个SNN和一个通过权重共享耦合的ANN。图7(a)展示了该串联学习框架。在前向传播时SNN结构利用前一层输出的脉冲序列S^{l-1}计算当前层的输出脉冲序列S^l和脉冲数$c^l=\sum_{t=0}^{T-1}S^l[t]$，而ANN则利用前一层的脉冲数$c^{l-1}$计算当前层的激活值$a^l$来近似脉冲数。在反向传播时，使用ANN的激活值$a^l$的梯度代替脉冲数$c^l$的梯度，

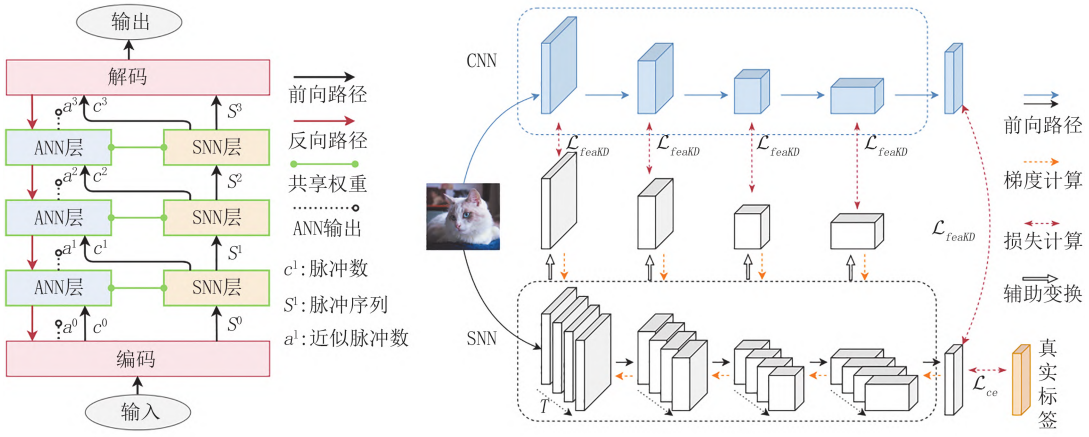


图7　两类ANN辅助训练方法 (a)共享权重法 (b)蒸馏法

通过ANN的反向传播计算前一层激活值和权重的梯度，具体计算方法为：

$$\frac{\partial \mathcal{L}}{\partial a^{l-1}} \approx \frac{\partial \mathcal{L}}{\partial c^{l-1}} = \frac{\partial \mathcal{L}}{\partial a^{l}} \cdot \frac{\partial a^{l}}{\partial c^{l-1}} \tag{40}$$

$$\frac{\partial \mathcal{L}}{\partial W^{l-1}} = \frac{\partial \mathcal{L}}{\partial a^{l-1}} \cdot \frac{\partial a^{l-1}}{\partial W^{l-1}} \tag{41}$$

其中，$\mathcal{L}$是模型的损失，W^{l-1}是第$l-1$层的权重。该方法在ANN中计算SNN输出的误差，使用ANN的梯度代替SNN的梯度更新权重，避开了脉冲释放过程不可导的问题和SNN复杂的梯度计算。

Kheradpisheh等[154]设计了一对由IF神经元组成的SNN网络和由ReLU激活函数组成的ANN网络，两个网络共享权重。该网络利用IF神经元输出的频率来近似ReLU神经元的输出，用SNN的输出近似ANN的输出。不同于Wu等[153]在前向传播时将SNN脉冲数作为ANN层的输入，Kheradpisheh等[154]在前向传播时分别运行SNN和ANN。在反向传播时，该方法不是直接计算ANN的真实梯度，而是将ANN输出替换为SNN输出，从而在ANN中计算SNN的近似梯度：

$$\mathcal{L} = -\sum_k Y_k \ln(O_k^A) \approx -\sum_k Y_k \ln(O_k^S) \tag{42}$$

$$\frac{\partial \mathcal{L}}{\partial W_{ji}^l} = \sum_k \frac{\partial \mathcal{L}}{\partial O_k^A} \sum_d \frac{\partial O_k^A}{\partial y_d^L} \frac{\partial y_d^L}{\partial W_{ij}^l} \approx \sum_k \frac{\partial \mathcal{L}}{\partial O_k^S} \sum_d \frac{\partial O_k^S}{\partial y_d^L} \frac{\partial y_d^L}{\partial W_{ij}^l} \tag{43}$$

其中，$\mathcal{L}$是网络的损失函数，Y_k是第k类的目标值，如果样本为第k类，则Y_k为1，否则为0；y_d^L是代理ANN网络最后一层第d个神经元的输出，O_k^A、O_k^S是代理ANN网络和SNN网络的在第k个类别的输出，W_{ij}^l是第l层的权重。该方法用SNN的输出O_k^S替换ANN的输出O_k^A，从而在ANN模型中反向传播计算SNN模型的误差。

基于蒸馏的SNN训练方法中，Xu等[155]和Qiu等[156]利用知识蒸馏方法，SNN模型作为学生从教师ANN模型中学习，该方法可以在很短的时间步长上有效地构建深层SNN网络。

Xu等[155]提出了基于响应的知识蒸馏和基于特征提取的知识蒸馏两种方法。基于响应的知识蒸馏只从教师ANN模型的最后一层的输出中提取知识，其损失函数包含SNN输出Q_s与真实标签y_{true}以及蒸馏标签Q_T的交叉熵损失：

$$\begin{aligned}\mathcal{L}_{KD} = {} & \alpha\tau^2 \cdot \mathrm{CrossEntropy}(Q_S^\tau, Q_T^\tau) + \\ & (1-\alpha) \cdot \mathrm{CrossEntropy}(Q_S, y_{\mathrm{true}})\end{aligned} \tag{44}$$

其中，τ是用于平滑概率分布的温度参数，Q_S^τ和Q_T^τ是利用模型最后一层第i个神经元的输出Z_i来计算得到的，其中第i个元素q_i的计算公式为$q_i = \mathrm{Softmax}(Z_i/T)$，$\alpha$用于权衡两种损失的重要程度。基于特征提取的知识蒸馏从教师ANN模型的中间层提取隐藏知识，其损失函数包含学生SNN的输出与真实标签的损失L_{task}以及中间层特征的$L2$距离损失$\mathcal{L}_{distill}$：

$$\mathcal{L}_{KD} = \mathcal{L}_{task} + \alpha \cdot \mathcal{L}_{distill} \tag{45}$$

$$\mathcal{L}_{distill} = \sum_i (T_i - S_i)^2 \tag{46}$$

其中，T_i是经过边缘ReLU处理后以抑制负信息影响的教师ANN模型的中间层特征，S_i是经过1×1卷积层匹配通道大小后的学生SNN的中间层特征。

Qiu等[156]通过神经网络结构搜索实验表明，与更大规模、更高性能的教师模型相比，具有相同架构的教师ANN模型在训练学生SNN模型时效果更好。基于这一发现，其提出了一个自架构知识蒸馏

框架。如图7(b)所示,该框架将教师ANN模型的知识转移到具有相同体系结构的学生SNN网络中。该网络的总损失函数$\mathcal{L}_{all}$包含以下三部分:传统的交叉熵损失$\mathcal{L}_{ce}$、让学生模型模仿教师模型特征图的特征蒸馏损失$\mathcal{L}_{feaKD}$以及约束学生模型的输出分布接近教师模型的输出分布的logits蒸馏损失$\mathcal{L}_{logKD}$:

$$\mathcal{L}_{all}=\alpha\cdot\mathcal{L}_{ce}+\beta\cdot\mathcal{L}_{feaKD}+\gamma\cdot\mathcal{L}_{logKD} \tag{47}$$

$$\hat{\mathcal{F}}_s=\mathcal{T}_s(\mathcal{F}_s)=\mathrm{BN}\left(\mathrm{Conv}\left(\frac{1}{T}\sum_T\mathcal{F}_s\right)\right) \tag{48}$$

$$\hat{\mathcal{F}}_t=\mathcal{T}_t(\mathcal{F}_t)=\mathcal{F}_t \tag{49}$$

$$\mathcal{L}_{feaKD}=\left\|\hat{\mathcal{F}}_s-\hat{\mathcal{F}}_t\right\|^2 \tag{50}$$

$$\mathcal{L}_{logKD}=\tau^2\sum p_\tau^t log\left(\frac{p_\tau^t}{p_\tau^s}\right) \tag{51}$$

$$p_\tau^s(i)=\frac{exp(p^s(i)/\tau)}{\sum exp(p^s/\tau)} \tag{52}$$

其中,α、β和γ是控制不同损失权重的超参数,$\mathcal{F}_s$和$\mathcal{F}_t$分别表示学生SNN模型和教师ANN模型的中间层特征,BN和Conv分别表示批量正则化层和卷积层,$\mathcal{T}_s$和$\mathcal{T}_t$表示SNN和ANN模型的特征变换,T表示时间步数,p_τ^t和p_τ^s分别表示ANN和SNN的预测分布,τ是平滑参数。卷积层将SNN的特征映射到连续空间,以解决特征维度不匹配的问题。

共享权重类方法直接避开了SNN计算代价高、训练耗时长、内存消耗大的反向传播流程,但ANN和SNN本身的差异,共享权重和不精确的梯度会导致训练出的SNN性能较其耦合的ANN有较大程度下降,因而这一类方法并未被广泛使用。基于蒸馏类的SNN训练方法通常需要额外引入ANN的输出以计算损失,训练代价比普通的梯度替代法更高,但由于ANN的指导作用,训练出的网络性能强于只使用数据集中目标值计算损失的普通SNN。两类方法均需要ANN的辅助,而ANN不具有时间维度,无法处理时域任务;这一缺陷使得ANN辅助类算法的应用范围非常受限。

3.7 事件驱动学习算法

事件驱动学习方法使用网络发放的脉冲传递梯度信息,其反向传播也是稀疏的,而普通方法则使用稠密的反向传播。事件驱动方法中梯度表示其传播至脉冲时脉冲发放时刻的改变量,而普通方法的梯度则表示脉冲的取值应该增加还是减少。

在事件驱动学习方法中,梯度在相邻层之间的传播一般从神经元的输出脉冲传递到释放脉冲时的膜电位,再从该膜电位分别向输入脉冲和对应的突触连接权重传递。Zhang等[157]在事件驱动学习的基础上,进一步考虑了脉冲响应模型(Spike Response Model, SRM)神经元中重置核导致的多个脉冲之间的相互作用,从而推导出更为细致的反向传播公式。Zhu等[158]基于SRM神经元,推导出了事件驱动学习方法在含有神经元的网络层反向传播中具有梯度之和不变性:

$$\sum_j\sum_{t_m(s_j^{(l-1)})}\frac{\partial\mathcal{L}}{\partial t_m(s_j^{(l-1)})}=\sum_i\sum_{t_k(s_i^{(l)})}\frac{\partial\mathcal{L}}{\partial t_k(s_i^{(l)})} \tag{53}$$

其中,等式左边是第$l-1$层所有脉冲携带的梯度之和,j和$t_m(s_j^{(l-1)})$分别对应第$l-1$层的单个神经元和单个脉冲,等式右边是第l层所有脉冲携带的梯度之和。该工作进一步分析了不含神经元的池化层,改进了平均池化层使其满足梯度之和不变性。在此基础上,Zhu等[159]进一步探究了损失函数对时序的事件驱动学习方法的影响。该研究发现,基于频率的损失函数同样适用于时序的事件驱动学习方法,并针对先前损失函数在目标类别输出神经元上梯度之和与脉冲发放数量差异不成正比的问题,提出了改善型计数损失。该工作还将权重归一化中使用的比例因子的训练转移至阈值并提升了性能。

目前事件驱动的学习算法研究还处于起步阶段,性能远低于传统算法,但其稀疏的反向传播在理论上能够部署于事件驱动的神经形态计算芯片,使得SNN的片上训练成为可能,前景广阔。

3.8 在线学习算法

在线学习方法为SNN这种需要多个时间步进行学习和推理的模型提供在单个时间步内训练网络的方法,避免了BPTT需要存储大量中间状态的需求。在线学习方法的内存消耗量通常为$\mathcal{O}(1)$,而BPTT则是$\mathcal{O}(T)$。因此,在线学习适用于资源受限或时间步数较多的场景。

DECOLLE(Deep Continuous Local Learning)[160]是最早的深度SNN在线学习方法之一,其针对双指数脉冲响应神经元,通过在每层的输出脉冲后引入一个读取层获取局部损失,实现了学习规则在时间和空间上的局部化。其后的在线学习算法则是对BPTT的完整梯度进行分析,对其在每个时刻的分量进行单步近似,相较于仅使用局部信息的DECOLLE性能更好。

由于梯度的计算通常不是逐元素的,涉及张量

(Tensor)之间的求导，在本小节中我们使用粗体字母表示张量，而字母含义则与前文一致。不失一般性，对于使用LIF神经元的多层SNN，基于BPTT的第l层的权重W^l的梯度为：

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mathbf{W}^l}=\sum_{t=0}^{T-1}\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mathbf{V}^{l+1}[t]}\frac{\partial\mathbf{V}^{l+1}[t]}{\partial\mathbf{S}^l[t]}\frac{\partial\mathbf{S}^l[t]}{\partial\mathbf{V}^l[t]}\frac{\mathrm{d}\mathbf{V}^l[t]}{\mathrm{d}\mathbf{W}^l}=\sum_{t=0}^{T-1}\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mathbf{V}^{l+1}[t]}\frac{\partial\mathbf{V}^{l+1}[t]}{\partial\mathbf{S}^l[t]}\frac{\partial\mathbf{S}^l[t]}{\partial\mathbf{V}^l[t]}\cdot\sum_{j=0}^{t}\left[\prod_{i\in[j+1,t],i\in\mathbb{N}}\left(\frac{\partial\mathbf{V}^l[i]}{\partial\mathbf{V}^l[i-1]}+\frac{\partial\mathbf{V}^l[i]}{\partial\mathbf{S}^l[i-1]}\frac{\partial\mathbf{S}^l[i-1]}{\partial\mathbf{V}^l[i-1]}\right)\right]\frac{\partial\mathbf{V}^l[j]}{\partial\mathbf{W}^l}\tag{54}$$

在线学习方法通常希望在网络运行到当前时间步t时，就能近似计算出式(54)的最左侧求和符号内的第t项。目前较为先进的在线学习方法，大多对式(54)中的部分梯度变量进行简化或近似，使得整个梯度表达式能够拆分成T个梯度的求和项；接下来，在每一步的前向传播完成后，通过反向传播即可计算出该求和式的第t项，并使用该梯度或多个时刻的梯度之和来更新参数，无需使用BPTT。例如，Xiao等[161]提出的Online Training Through Time (OTTT)的主要改动为：忽略重置过程的梯度，使$\frac{\partial\mathbf{V}^l[i]}{\partial\mathbf{S}^l[i-1]}=0$；使用软重置，使得$\frac{\partial\mathbf{V}^l[i]}{\partial\mathbf{V}^l[i-1]}=\lambda$为常数。基于上述简化设置，通过展开来自后续层的梯度，式(54)可以转写为如下形式：

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mathbf{W}^l}=\sum_{t=0}^{T-1}\left[\left(\frac{\partial\mathcal{L}}{\partial\mathbf{S}^{l_{max}}[t]}\frac{\partial\mathbf{S}^{l_{max}}[t]}{\partial\mathbf{V}^{l_{max}}[t]}\prod_{j=l+1}^{l_{max}-1}\frac{\partial\mathbf{V}^{j+1}[t]}{\partial\mathbf{S}^j[t]}\frac{\partial\mathbf{S}^j[t]}{\partial\mathbf{V}^j[t]}\right)\frac{\partial\mathbf{V}^{l+1}[t]}{\partial\mathbf{S}^l[t]}\frac{\partial\mathbf{S}^l[t]}{\partial\mathbf{V}^l[t]}\right]^{\mathrm{T}}\cdot\left(\sum_{\tau\leqslant t,\tau\in\mathbb{N}}\lambda^{t-\tau}\mathbf{S}^{l-1}[\tau]\right)^{\mathrm{T}}\tag{55}$$

定义资格迹(Eligibility Traces)变量：

$$\mathbf{E}^{l-1}[t]=\lambda\mathbf{E}^{l-1}[t-1]+\mathbf{S}^{l-1}[t]\tag{56}$$

并初始化为$\mathbf{E}^{l-1}[0]=\mathbf{S}^{l-1}[0]$。在该设置下式(55)中的$\sum_{\tau\leqslant t,\tau\in\mathbb{N}}\lambda^{t-\tau}\mathbf{S}^{l-1}[\tau]$即可转换为$\mathbf{E}^{l-1}[t]^{\mathrm{T}}$，此时该式的第$t$个求和项只需要$t$时刻的信息。该工作还从理论上论证了其梯度与基于脉冲表征的Differentiation on Spike Representation方法[162]之间的正相关性。

Spatial Learning Through Time (SLTT)[163]延续了OTTT忽略重置梯度和使用软重置的设置，并完全忽略膜电位在时刻之间的梯度，即令$\frac{\partial\mathbf{V}^l[i]}{\partial\mathbf{V}^l[i-1]}=0$。在上述设置下，式(54)简化为：

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mathbf{W}^l}=\sum_{t=0}^{T-1}\left[\left(\frac{\partial\mathcal{L}}{\partial\mathbf{S}^{l_{max}}[t]}\frac{\partial\mathbf{S}^{l_{max}}[t]}{\partial\mathbf{V}^{l_{max}}[t]}\prod_{j=l+1}^{l_{max}-1}\frac{\partial\mathbf{V}^{j+1}[t]}{\partial\mathbf{S}^j[t]}\frac{\partial\mathbf{S}^j[t]}{\partial\mathbf{V}^j[t]}\right)\frac{\partial\mathbf{V}^{l+1}[t]}{\partial\mathbf{S}^l[t]}\frac{\partial\mathbf{S}^l[t]}{\partial\mathbf{V}^l[t]}\right]^{\mathrm{T}}\mathbf{S}^{l-1}[t]^{\mathrm{T}}\tag{57}$$

因而SLTT只需要保存当前时间步的脉冲$\mathbf{S}^{l-1}[t]$即可实现在线学习。

在OTTT的基础上，Neuronal Dynamics-based Online Training (NDOT)[164]对层内的时间依赖性进行了更细致的建模，不像式(56)中简单地使用膜电位的衰减λ，而是将重置过程也纳入：

$$\mathbf{E}^{l-1}[t]=\mathbf{E}^{l-1}[t-1]\cdot\frac{U^l[t]-V_{th}\mathbf{S}^l[t]}{U^l[t-1]-V_{th}\mathbf{S}^l[t-1]}+\mathbf{S}^{l-1}[t]\tag{58}$$

Zhu等[165]则考虑在SNN在线学习中加入归一化机制。由于在线学习过程中无法使用未来信息，而直接在每一步进行BN存在协方差漂移问题，该工作提出了包含BN和线性仿射变换的Online Spiking Renormalization (OSR)模块以保证训练和推理时归一化变换参数的一致性，还引入了在线阈值稳定器以稳定时间步之间的神经元发放率。OSR模块训练时的计算流程如下：

$$\hat{I}[t]=\frac{I[t]-\mu[t]}{\sqrt{\sigma^2[t]+\epsilon}}\tag{59}$$

$$\tilde{I}[t]=\gamma\cdot\left(\hat{I}[t]\cdot\mathrm{NoGrad}\left(\frac{\sqrt{\sigma^2[t]+\epsilon}}{\sqrt{\widehat{\sigma^2}+\epsilon}}\right)+\mathrm{NoGrad}\left(\frac{\mu[t]-\hat{\mu}}{\sqrt{\widehat{\sigma^2}+\epsilon}}\right)\right)+\beta\tag{60}$$

在第t个时间步中，$I[t]$是未经变换的输入电流，$\mu[t]$和$\sigma^2[t]$分别是$I[t]$的均值和方差，$\hat{\mu}$、$\widehat{\sigma^2}$分别是BN层内记录的均值和方差统计量，$\hat{I}[t]$是BN变换后的值，$\tilde{I}[t]$是二次线性变换之后的值，ϵ是一

个非常小的正数以防止分母为0，NoGrad(…)内的运算不参与反向传播。在推理时，OSR的行为则和BN完全一致。

Hu等[92]从工程实践角度提出降低SNN训练内存开销的另一个思路，其通过实验发现常规BPTT训练中只有最后一层的时序信息对训练所得权重影响大，于是在前向传播中断开了除最后一层外的时序传播计算图。这一操作简单易行，且实验效果良好。对于保留了前向时序传播的最后一层，该工作使用可逆模块实现了使用$\mathcal{O}(1)$存储空间记录T步信息的效果，其核心方法是用后一时间步的信息来表示前一时间步膜电位。此外该工作在网络中使用了ConvNeXt模块[166]，并将前一时间步的高层信息融合到了当前时间步的低层信息中以提升网络的任务性能。

未来信息不可在当下获取，故已有的在线学习方法都假设$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}S^l[t]} \approx \frac{\partial \mathcal{L}[t]}{\partial S^l[t]}$，这一近似相当于认为当前时刻的输出脉冲只参与当前时刻的损失计算，而对未来时刻的损失不会产生影响。这一近似其实并不符合实际情况。例如典型的分类任务中，每个时间步发放的脉冲都会参与发放频率的计算，并影响分类结果。这一特性也使得在线学习方法难以处理时序任务。

3.9 训练加速方法

相较于ANN，SNN额外增加了时间维度，在不使用在线学习方法、默认使用BPTT方法训练的情况下，网络的训练耗时和内存消耗通常和总时间步T近似成正比，带来了显著高于ANN的训练开销。如何对SNN训练加速成为研究者们日益关心的话题。GPU拥有强大的并行计算能力，是训练SNN的首选设备，目前已有的SNN训练加速方法都基于GPU和SNN的特性进行设计。

稀疏脉冲梯度下降[167]在反向传播时，将满足$|H[t]-V_{th}| \geqslant B_{th}$的神经元视作不活跃的神经元，其中$B_{th}$是梯度阈值超参数，并将其脉冲释放过程的梯度$\frac{\partial S[t]}{\partial H[t]}$视作0，从而使得本应稠密的反向传播的计算图变得稀疏，然后使用PyTorch中自带的稀疏计算库进行加速。稀疏脉冲梯度下降方法相较于普通的梯度下降方法，在GPU上最高可达150倍的训练反向传播加速和85%的内存消耗减少，但其只在简单的全连接SNN上进行了实现和验证。SpikingJelly框架[49]提供了更为通用的深度SNN加速方法。SpikingJelly框架首先定义了SNN传播模式的概念，并提出逐步传播和逐层传播这两种计算图的构建方式。在逐层传播模式下，网络中的每层可以同时接收到尺寸为($T \times N \times \cdots$)的整个序列作为输入，其中T是序列长度，N是批量大小。对于无状态的卷积、全连接等突触层，SpikingJelly框架提供了包装器，将输入的时间和批量维度融合，即将输入尺寸变换到($TN \times \cdots$)，然后再送入无状态层计算，计算得到的结果再重新拆成序列，恢复到尺寸为($T \times N \times \cdots$)的序列。由于时间维度被当作了批量维度，不同时间步的计算也是并行的，速度远快于传统的通过循环实现的逐步计算。对于有状态的神经元等层，SpikingJelly框架使用自定义的CuPy[168]后端，将神经元遍历所有时间步的迭代计算封装到单个CUDA内核，相较于PyTorch实现的神经元在计算时调用多个小CUDA内核，单个大CUDA内核的调度开销更小、计算速度更快，在T较大时能有数十倍加速效果。综合使用无状态层和有状态层的加速方法，SpikingJelly框架相较于其他SNN框架实现的SNN，最高可达11倍的训练加速效果。

Luke等[169]提出了一种加速脉冲神经元的时间分组仿真方式，在仿真脉冲神经元时，将时间步分组，每组时间步内忽略神经元的重置过程，从而使得膜电位的计算从迭代计算改为直接求解，并将膜电位与阈值比较，输出脉冲，这一思路与PSN[70]类似；前述过程忽略了重置，会导致输出脉冲数量多于有重置的正常神经元仿真方式，PSN没有采取任何处理措施，而该方法则对输出脉冲进行修正，仅保留每组时间步内的第一个脉冲，一定程度上缓解了PSN去掉重置可能导致的发放率升高问题。该方法相较于正常仿真过程，性能有所降低，但仿真速度大幅度提升。

4 综合对比实验

此前尚未有工作将不同类别的方法进行统一的比较，因而本章选取了各类学习算法中的代表性方法，在相同的设置下进行实验，参与比较的方法包括SpikingJelly框架[49]中基于CuPy后端加速的IF神经元和LIF神经元作为基准(Baseline)、神经元和突触改进算法中的CLIF神经元[95]和PSN家族[71]、正则化方法中的TEBN[148]、在线学习算法中的OSR[165]、训练加速算法中的时间分组仿真方式加速的BlockALIF神经元[169]、ANN辅助训练算法中的

Tandem学习方法[153]和响应与特征蒸馏[155]。这些方法都提供了开源代码，可以直接使用。实验任务包括静态CIFAR10和序列CIFAR10的分类、神经形态的SHD语音数据集[68]分类、神经形态的Gen1[134]数据集和静态COCO数据集的目标检测。需要指出的是，Tandem学习方法[153]和响应与特征蒸馏[155]依赖于ANN，因而不适用于序列CIFAR10分类和神经形态数据集相关任务，仅在静态CIFAR10图片分类任务上进行实验。需要注意的是，本次实验中并没有纳入网络结构改进类方法，因为这些方法已经在复杂的ImageNet数据集上进行了公平的性能比较，结果如前文图6所示。本章还选取了代表性的训练加速方法，包括SpikingJelly框架中融合内核实现的LIF神经元[49]、并行脉冲神经元PSN[71]和时间分组仿真方式加速的BlockALIF神经元[169]并对比了他们的加速效果。关于本章中实验的代码下载链接和详细超参数则参见附录。

4.1　CIFAR图片分类任务性能

本文使用Fang等[71]的网络结构，测试各类方法分类静态CIFAR10和序列CIFAR10的任务性能，以此检验各类方法的静态数据集分类性能和长期依赖学习能力。除CLIF神经元和PSN家族的网络外，ANN辅助训练类算法的网络中使用IF神经元，其他网络均使用LIF神经元。BlockALIF神经元均使用每组2个时间步，因为如果每组1个时间步与普通神经元无异，则没有任何加速效果；如果每组更多时间步，则实验发现其分类性能剧烈下降。对于PSN家族的网络，CIFAR10分类任务使用PSN，而序列CIFAR10分类使用$k=4$的Sliding PSN。图8全面对比了各类方法的性能。

图8(a)展示了各类方法的分类正确率。对于静态的CIFAR10分类，神经动态中不带衰减的IF神经元表现强于LIF神经元，而序列CIFAR10分类则是神经动态更为复杂的LIF神经元性能更好。在静态CIFAR10分类任务上，PSN性能略高于CLIF神经元，均强于IF神经元；而在序列CIFAR10分类任务上，CLIF神经元相较于LIF神经元提升明显，而Sliding PSN性能又大幅度超越CLIF神经元，表明PSN家族通过直接权重连接替换马尔科夫链，极大增强了长期依赖学习能力，而CLIF神经元增加补充电位的神经动态也有利于缓解梯度随时间的衰减。TEBN在两种任务上都相较于普通网络提升显著，表明使用全部时刻的统计量和逐时刻的仿射变换，相较于普通的BN有效捕捉到了输入的分布并提升了拟合能力。OSR方法在CIFAR10分类任务上性能非常接近其他BPTT类方法，但在序列CIFAR10分类任务上性能较差，表明静态任务中未来的梯度或可忽略，而时域任务中未来的梯度则至关重要。BlockALIF神经元性能较差，而且在$T=32$的序列CIFAR10分类任务上性能下降更严重，表明时间上的分组限制了脉冲发放次数，对性能有着很大的负面影响。Tandem学习方法由于使用不精确的梯度，性能弱于基于替代函数训练的IF神经元。蒸馏方法相较于原始的使用IF神经元的网络，性能均有一定提升，其中特征蒸馏提升稍高，且均高于Tandem学习方法，表明来自ANN的知识帮助较大。

图8(b)展示了各类方法的训练速度，可以发现基于CuPy后端加速的IF和LIF神经元速度最快，体现了SpikingJelly框架中融合CUDA内核带来的巨大优势。令人意外的是，PSN家族的速度慢于CuPy后端加速的IF和LIF神经元，这可能是由于该实验设置下，批量数为128、网络通道数为256，内存读写需求较大，PSN家族的核心运算矩阵乘法遭遇了内存瓶颈(Memory-Bound)。通过额外实验发现，设置批量大小32、网络通道数32时，内存读写需求大大降低，此时PSN训练速度达到2345 samples/s，高于CuPy后端加速的IF神经元1649 samples/s的训练速度，佐证了前述猜想。TEBN方法中使用的仍然是CuPy后端加速的神经元，但由于逐时刻仿射变换的额外计算而拖慢了速度。OSR速度大幅度落后于其他方法，因其是在线学习方法，只能使用SpikingJelly框架中的逐步(Step-By-Step)传播，而其他方法则可以使用逐层(Layer-By-Layer)传播并通过融合时间批量维度来加速无状态层。BlockALIF神经元的加速效果较差，主要原因在于神经动态极为复杂，且使用时间上的卷积实现并行计算，而Fang等[71]在设计Sliding PSN的经验表明卷积并行度较低，速度远慢于矩阵乘法。关于BlockALIF神经元的加速效果，将在后文予以详细讨论。Tandem方法速度较慢，原因在于其需要ANN部分的计算，且神经元使用纯PyTorch实现，速度远不如CuPy实现。蒸馏方法中仍然使用CuPy后端加速的IF神经元，但由于ANN部分的计算和蒸馏损失拖累了速度，其中特征蒸馏需要更多的损失项，因而速度比响应蒸馏更慢。CLIF神经元的作者也是使用PyTorch实现，且神经动态较为复杂，故速度较慢。图8(c)展示了各类方法的推理速度，整体和训练速度一致，不再赘述。

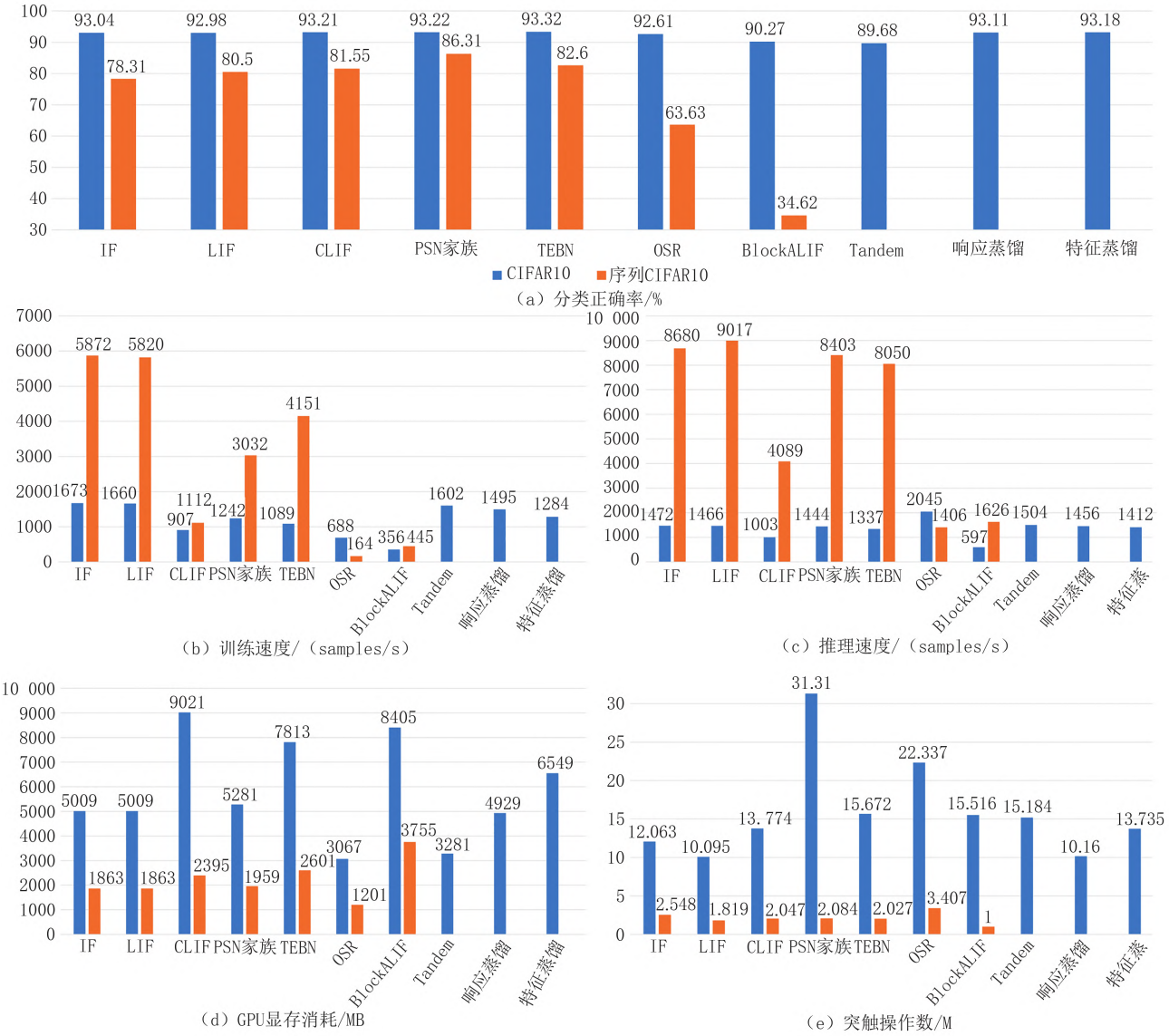


图8 CIFAR图片分类性能对比

图8(d)展示了各类方法在训练时的GPU显存消耗。神经网络在训练时，需要保存权重和各种计算的中间变量以用于反向传播。权重的数量在网络结构确定后就固定了，因而中间变量的数量对内存消耗起决定性作用。使用BPTT训练的SNN，需要保存所有时刻的中间变量，而在线学习方法只需要保存单个时间步的中间变量，因而OSR方法是各类方法中消耗内存最少的。Tandem方法的反向传播基于ANN，也消除了时间维度，故内存消耗量与在线学习方法接近。其他方法基于BPTT训练，其中CLIF神经元引入了额外的补充电位以及Sigmoid激活函数、BlockALIF神经元具有复杂的神经动态、TEBN逐时刻的仿射变换使BN层的计算变为原来的T倍，三者都引入了大量中间变量，因而带来了较大的内存消耗。蒸馏类方法一方面需要ANN计算，另一方面引入了额外损失，故内存消耗量比直接训练的方法更大；特征蒸馏的损失项比响应蒸馏更多，故内存消耗量进一步提升。相较于IF神经元和LIF神经元，PSN家族并未引入额外计算或中间变量，因而内存消耗量与它们接近。需要注意的是，尽管LIF神经元相较于IF神经元神经动态更复杂、中间变量更多，但在CuPy后端实现时这些中间变量由CUDA内核内部处理，而不是通过PyTorch的自动微分机制，因而这些变量在CUDA内核执行完毕后就自动释放了，所以两者的内存消耗完全一致。

图8(e)展示了各类方法的突触操作数(Synaptic Operations)，在网络结构确定后，该指标主要由神经元的发放率决定。 PSN家族的突触操作数显著高于作

为基准的IF神经元和LIF神经元，表明其去除重置确实导致了发放率的显著升高。OSR和TEBN的突触操作数也较高，需要注意的是前者尽管是在线学习方法，却在静态CIFAR10上取得了接近BPTT类方法的正确率，而后者则取得了静态CIFAR10的最高正确率，可能是它们在训练过程中通过提升发放率来获得了性能增益。在该任务中，作为基准的IF神经元和LIF神经元每层脉冲神经元的发放率都低于50%，而Guo等[170]则猜想发放率越接近50%则信息熵越大，网络性能越好，OSR和TEBN的表现与其猜想一致。其他方法的突触操作数则较为接近，但也存在一定差异。例如，LIF神经元低于IF神经元，表明膜电位的泄露行为降低了整体的发放率。Tandem方法使用脉冲频率传递信息，而特征蒸馏则是直接拟合ANN的ReLU的输出，两者都暗含使用频率编码表示信息，因而发放率和突触操作数稍高；而响应蒸馏中，蒸馏温度以及ANN输出的“暗知识(Dark Knowledge)”，导致拟合目标不是理想的独热编码形式，损失计算相较于仅使用真实标签计算交叉熵时有所松弛，可能是这一特性导致其突触操作数反而低于作为基准的IF神经元。BlockALIF神经元在序列CIFAR10分类的突触操作数极低，考虑到其正确率只有34.62%，表明确实是时间维度分组限制发放导致的较低性能水平。

4.2　神经形态语音数据SHD分类性能

本文使用神经形态的SHD语音数据集[68]进行实验，并使用Ilyass等[100]的开源代码和网络结构。需要注意的是，该实验使用固定时间间隔积分来处理输入事件[49]，因而输入的帧数为88～126。

实验结果展示在表4中。在正确率方面，该数据集上各种方法的表现与CIFAR分类存在较大差异。作为基准的IF神经元和LIF神经元，前者性能高于后者，而在常见的神经形态数据集分类任务中则通常是LIF神经元性能更好，这可能暗示了语音数据和视觉数据存在较大差异。TEBN与性能更好的IF神经元配合使用，但正确率低于作为基准的IF神经元，可能是其难以处理变长序列的输入所致。BlockALIF神经元性能最好。通过额外的实验发现，当去除BlockALIF神经元的自适应的阈值后，其性能会下降到和IF神经元持平，表明自适应阈值使其较好地捕捉了音频数据中的时域信息。其他方法都表现较差，低于作为基准的IF神经元，尤其是OSR方法，可能是该任务的时间步较大，因而在线学习方法相较于BPTT难以补偿时间梯度的缺失。而在突触操作数方面，由于网络本身规模很小且层数浅，故各类方法的差别不大。

表4　对比各类代表性方法SHD分类和Gen1、COCO目标检测任务性能

评估指标\方法	IF	LIF	CLIF	Sliding PSN	TEBN	OSR	BlockALIF
SHD分类正确率(%)	78.33	66.98	66.1	69.23	74.54	40.93	82.36
SHD分类突触操作数(M)	0.0251	0.0262	0.0263	0.0264	0.0262	0.0228	0.0239
NCAR预训练分类正确率(%)	81.00	91.69	85.34	91.41	90.96	56.23	68.14
Gen1目标检测mAP@0.5	0.0013	0.4399	0.2388	0.4781	0.5213	0.0007	0.1347
Gen1目标检测mAP@0.5:0.95	0.0002	0.2106	0.090 204	0.2391	0.2656	0.0003	0.0457
Gen1目标检测突触操作数(M)	385.86	296.33	353.03	293.53	214.72	305.93	232.69
COCO目标检测mAP@0.5	0.409	0.397	0.410	0.389	0.408	0.247	-
COCO目标检测mAP@0.5:0.95	0.225	0.216	0.228	0.213	0.223	0.120	-
COCO目标检测突触操作数(G)	363.79	444.67	537.37	1004.68	360.62	561.75	-

4.3　Gen1和COCO目标检测性能

本文使用神经形态的Gen1[134]数据集和静态的COCO数据集进行目标检测任务训练。Gen1数据集由事件相机采集的驾驶场景组成，COCO数据集则是最常用的静态目标检测数据集，两者已经被大量SNN目标检测相关研究使用。对于Gen1数据集，本文使用Fan等[140]的开源代码和网络结构，使用Spiking DenseNet121-16[136]作为检测的骨干网络，首先在神经形态的NCAR[137]数据集上预训练分类任务，然后在Gen1数据集上进行目标检测任务训练并记录最终的mAP。对于COCO数据集，本文使用Su等[139]提出的全脉冲结构的Energy-efficient Membrane-Shortcut (EMS) ResNet-10并直接训练。

实验结果展示在表4中。对于Gen1数据集的目标检测任务，NCAR预训练结果显示，LIF神经元优于IF神经元，也强于CLIF神经元、Sliding PSN、TEBN。OSR和BlockALIF神经元性能都较差。需要指出的是，根据Fan等[140]的设计，预训练设置了早停(Ealy Stopping)，连续5轮训练的最小验证集损失不减少则退出训练。该预训练仅为后

续目标检测任务提供较优的初始参数，因而本身的分类正确率无需过多关注。从目标检测性能看，性能排序为TEBN>Sliding PSN>LIF>CLIF>BlockALIF>IF>OSR，且IF神经元和OSR几乎无精度，表明训练完全不收敛。对于能够正常收敛的方法，突触操作数排序为TEBN<BlockALIF<Sliding PSN<LIF。整体来看，TEBN在该检测任务中表现极好，精度最高、突触操作数最低，可能是其凭借逐时刻的仿射变换增强了网络对时序数据的拟合能力；Sliding PSN和LIF神经元表现稳定；IF神经元不收敛，可能是其仅积分不作衰减的特性难以捕捉时序动态所致；CLIF神经元和BlockALIF神经元性能都较差，它们都使用了自适应的阈值更新机制，可能表明该机制未必总对性能有益；OSR方法不收敛，可能是该任务的时序信息较多，在线学习方法在缺失完整梯度的情况下难以训练参数。对于COCO数据集，以mAP@0.5:0.95作为主要指标，性能排序是CLIF>IF>TEBN>LIF>Sliding PSN>OSR；BlockALIF神经元训练速度太慢，耗时约为其他方法的8倍，不具备完成训练的可行性，因而在表格中的性能是空缺值。整体来看，除BlockALIF神经元训练太慢、OSR方法性能较差，其余方法的性能较为接近。突触操作数排序是TEBN<IF<LIF<CLIF<OSR<Sliding PSN。值得注意之处在于，CLIF神经元达到了最优性能，远强于在Gen1数据集的表现，但突触操作数明显高于LIF神经元；TEBN方法则相较于使用普通BN、作为基准的IF神经元性能略微下降，而其在Gen1数据集则表现最好，但该方法在两种数据集上的突触操作数都最低；Sliding PSN的突触操作数约为其他方法的2倍，表明去除重置带来脉冲发放率增加的负面效果较为严重。需要指出的是，在Gen1数据集的目标检测任务中，本文使用的Fan等[140]的网络结构和训练流程，将SNN作为检测骨干(Backbone)网络，而检测头则是由ANN实现，这也是绝大多数已有研究使用的方式。而对于COCO数据集的目标检测任务，本文沿用的Su等[139]的方法，则是使用全脉冲的流程，检测骨干网络和检测头均由SNN实现；且输入图片的分辨率较高，为640×640；由于上述原因，在COCO数据集上的突触操作数要远高于Gen1数据集，因而在表4中两者的单位分别是G和M。

4.4 加速性能测试

已有的SNN加速的研究集中在神经元层次，故本文选取PyTorch实现的LIF神经元、SpikingJelly框架中融合内核实现的LIF神经元[49]、并行脉冲神经元PSN[71]和时间分组仿真方式加速的BlockALIF神经元[169]进行实验，对比加速性能。实验环境为Intel Core i9-10900X CPU，64 G内存，Nvidia RTX 2080 Ti GPU；神经元数量为4096；分别测试不同神经元在时间步数$T=2,4,8,16,32,64$时进行训练(前向传播、反向传播和梯度下降)的耗时；输入是从取值范围(0,1)的均匀分布采样的随机张量。以PyTorch实现的LIF神经元作为速度基准，其他神经元与LIF神经元的速度之比展示在了表5中。实验结果显示，随着时间步数的增大，SpikingJelly优势明显，最高可达接近15倍训练加速效果，原因在于T较大时PyTorch实现的神经元会调用大量琐碎的CUDA内核，而SpikingJelly融合内核后可以大幅度降低琐碎内核的调度开销；PSN加速效果比SpikingJelly更胜一筹，最高可达近44倍加速，展现了并行加速相较于串行计算的巨大优势；BlockALIF神经元则加速效果较差，多数情况下速度反而慢于LIF神经元，一方面原因可能是在实验中时间步数最大为$T=64$，如此之大的时间步数在深度SNN中很少使用，但还是不足以大到能够弥补神经元内部使用卷积本身的调度开销。如果时间步数达到Luke等[169]测试的数千，则BlockALIF神经元有可能快于LIF神经元。另一方面原因在于BlockALIF神经元在处理没有同层反馈连接的神经元时会增加计算复杂度，其更适用于对有同层反馈连接的神经元进行加速，尤其是在分组较大时候，BlockALIF神经元可以将分组内部多个仿真步的反馈连接并行计算从而提高计算效率，从而达到Luke等[169]得到的较高加速比。总体而言，SpikingJelly对串行神经元加速效果好，但仍弱于并行的PSN，后者可能代表了未来的神

表5 对比加速方法性能

T	相较于LIF神经元的加速比						LIF耗时/ms
	SpikingJelly	PSN	BlockALIF分组大小				
			2	4	8	16	
2	1.03	2.20	0.20				1.44
4	1.48	4.07	0.17	0.38			3.02
8	2.72	6.81	0.15	0.29	0.60		4.79
16	6.19	12.60	0.22	0.29	0.56	1.29	9.48
32	16.61	17.76	0.25	0.40	0.59	1.01	17.14
64	14.83	43.75	0.24	0.45	0.72	0.98	30.60

经元加速方向。

4.5　性能对比测试小结

本章对不同方法进行了对比测试，并通过多个指标进行评价。综合多个数据集的结果来看，尚无某种方法能够在所有任务中都取得最好性能，但这也符合机器学习领域著名的没有免费午餐定理（No Free Lunch Theorem）[171]。研究者需要根据自身需求进行取舍来决定使用哪些方法。根据本章的实验可以得到一些初步结论。对于静态图片相关任务，推荐的方法包括结构简单、代码易于实现的IF神经元和PSN；如果对性能有较高追求，还可以考虑牺牲一定训练速度，使用CLIF神经元、TEBN或蒸馏类方法；如果显存受限，则使用OSR方法。对于时序数据处理，在线学习方法性能较差，不适合使用。对于神经形态数据集相关任务，使用LIF神经元、Sliding PSN或TEBN通常能达到较好效果，而在它们性能不好的情况下也值得尝试IF神经元和BlockALIF神经元。如果研究者需要加速训练过程，则推荐使用SpikingJelly中基于CuPy实现的神经元，其与PyTorch实现的计算等价，且加速效果较为明显。而突触操作数则随着方法和具体任务变化，尚无统一结论，例如Sliding PSN在CIFAR分类任务上突触操作数明显高于LIF神经元，但在Gen1目标检测任务则与之接近；TEBN在目标检测任务上突触操作数都低于其他方法，但在其他任务上则与其他方法差距不大。

5　研究挑战与未来研究方向

梯度替代算法近年来取得了飞速发展，成绩斐然，但仍有部分困扰整个研究领域的难题尚待解决。一定程度上，目前深度SNN性能的提升主要来自深度学习方法的贡献，这一方面带来了性能的飞跃，另一方面也则使得研究集中于ANN的脉冲化，而对SNN独有的编码方式、神经动态、学习算法等关注不够。针对这一现状，本文总结了以下研究挑战和对应的研究方向，值得领域内研究者关注。

（1）生物启发的高效神经编码算法设计

生物神经系统中最早被发现的编码方式是频率编码[172]。其后更多证据表明，生物神经系统中还存在更高效的编码方式，例如人类触觉感知系统可以通过单个脉冲的精确发放时刻来编码触感[173]，苍蝇可以凭借30毫秒内到达的一个或两个脉冲对环境做出快速反应[174]。目前爆发编码（Burst Coding）[175-176]、相位编码[89]等已经用于ANN2SNN方法，而在梯度替代学习算法中的应用较少。研究者们可以考虑设计生物启发的高效神经编码算法，并应用于SNN内部的信息表征，一方面能够充分利用时域信息以降低时间步数和能耗，另一方面也有望与脑机接口相关研究领域形成合力，加速理解大脑的工作原理。

（2）神经元动态过于简化

现有的神经元改进方案[62,71,94-95]中，所使用的神经元也都较为简化，并不具备计算神经科学中常用的Izhikevich[57]神经元模型相当的复杂神经动态。需要注意的是，SNN与ANN最大的区别即在于神经元；Wolfgang证明SNN能够实现与ANN相同的拟合能力，且使用更少的神经元[26]，其关键在于脉冲神经元相较于ANN中激活函数所不具备的神经动态；He等[177]通过简单网络结构与复杂的神经元动态结合，实现与复杂网络结构相同的性能，且内存消耗更少、运行速度更快。以上理论和实验结果表明，在SNN中使用复杂神经元具有诸多优势。但受限于其高昂的计算成本、复杂的参数调试，IF神经元、LIF神经元等高度简化的脉冲神经元模型仍然是深度SNN的首选。未来的研究可以考虑通过并行加速算法降低计算代价，以梯度下降法自动优化神经元参数，从而构建具有复杂神经动态和脉冲模式的神经元模型并应用于深度SNN。

（3）网络结构层次的时域动态被忽略

现有的SNN结构与ANN类似，包含堆栈式卷积层和池化层、残差连接。这一结构擅长提取空间特征而非时域特征，后者则通常被认为是由脉冲神经元负责。一个典型的例子是，即便是在SpikFormer[51]和Spike-Driven Transformer[52]这样最先进的脉冲Transformer架构中，其自注意力计算也是局限于单个时间步内，而不跨越时间步。这一设计理念导向了目前深度SNN的纯前馈网络结构，忽略了网络结构层次的时域动态，但大脑结构却并非如此。大脑中存在大量的同脑区和跨脑区稠密连接，共同构成了一个巨大的循环神经网络；传统观点认为视觉信息处理是一系列前馈过程，以此也衍生出卷积神经网络架构。但最近越来越多证据表明这一过程中存在反馈途径，高阶的认知和视网膜的信息存在相互作用[178]。Yin等[70]和Rao等[179]在SNN中增加了反馈连接，大幅度改善了网络的长期依赖学习能力，但遗憾的是其反馈只局限于脉冲神经元层内，并只在小网络、简单数据集上进行了验证。网

络结构层次的时域动态尚未在深度SNN中得到重视，这一问题值得研究者们关注。

（4）突触可塑性学习算法研究进展缓慢

反向传播算法根据网络输出计算最终损失，并将误差逐层回传，同时计算网络参数的更新量，是一种全局的学习方式。在反向传播算法通过替代梯度法引入SNN后，诸如赫布学习规则[180]、STDP[181]等突触可塑性学习算法则因性能低下而较少使用。然而，这些方法亦有独特优点：在理论研究方面，它们对应着生物实验中发现的现象和数据，对其研究有助于理解大脑学习的奥秘，因而备受计算神经科学的青睐；在实际应用方面，它们是局部的学习规则，在硬件上实现时只需要记录神经元和前后突触的活动信息，资源消耗远少于需要记录整个网络中间层信息的反向传播算法，适合片上学习。Nabil等[182]在Intel Loihi芯片[35]上实现了基于STDP的片上实时学习并用于气味识别，并且能够减缓灾难性遗忘；Wu等[183]将突触可塑性学习算法与梯度替代法共同使用，发现这种混合学习机制在小样本学习、持续学习和容错学习方面均优于纯梯度替代法，同时将该算法部署到Tianjic芯片[37]，受益于突触可塑性的局部性，各个计算核心之间的通信开销也大幅度降低。突触可塑性学习算法尽管已经展现出诸多潜力，但将其结合梯度替代法并用于改善大规模深度SNN的学习，则尚未有成功的先例报道，这一无人区值得研究者们探索。

（5）软件仿真和硬件部署的沟壑

SNN的目标运行设备是神经形态计算芯片，但现有学习算法更多关注于软件仿真，而对硬件部署面临的挑战关注不够。首先总结硬件相关的研究问题如下：

1）模型量化（Model Quantization）：GPU通常配备GB级别的显存，使用float32精度的突触，但神经形态计算芯片上资源有限，例如Loihi[35]至多支持9比特表示的突触权重。已经有部分研究者在中等规模SNN上进行了量化并达到较好效果[184-187]，其中Wei等[187]将权重量化到1比特、膜电位量化到2比特，在CIFAR100、Tiny ImageNet[188]等多个中等规模的数据集上达到了较好性能，展现出低精度SNN部署的巨大潜力。

2）网络剪枝（Network Pruning）：神经形态计算芯片上的内存容量有限，无法容纳太多突触和神经元，例如Loihi至多支持存储空间不超过16 MB的突触权重和12 800个神经元。而典型的VGG11网络则含有132.86 M的突触数，如果是以float32精度实现则需要531.44 MB存储空间，网络规模远超Loihi的容纳能力。对突触和神经元进行剪枝可以大幅度降低模型大小。目前SNN中的剪枝技术可以分为两类，一类是通用剪枝技术[189-194]，可以同时用于ANN和SNN；另一类则是SNN独有的，例如基于突触可塑性的剪枝方法[195-198]、基于神经元发放率的剪枝方法[199-200]等。

3）硬件非理想性（Hardware Non-idealities）：基于模拟电路/数模混合电路或忆阻器实现的神经形态芯片，例如Neurogrid[201]等，受限于制作工艺和自身特性，其运行SNN时存在一定噪声，输出结果与在CPU/GPU上的仿真存在一定差异。目前有少量研究者对这一问题进行了探索。Bhattacharjee等[202]发现SNN多步运行的特性在忆阻器上会累计误差，并通过利用BN的统计量来记录噪声，减少软件仿真和硬件部署的差异。Moro等[203]则是通过训练时注入噪声来模拟硬件运行的环境，使硬件推理和软件模拟的结果更一致。Christensen等[204]指出，设计“硬件感知”的软件学习算法，即在软件中模拟硬件特性，是解决硬件非理想性的关键途径。

4）同步仿真和异步部署：深度SNN中通常使用元素取值仅为0或1的张量表示脉冲，多个时刻的脉冲序列类似于视频帧的表示格式；而DVS相机和Loihi、Speck[54]等基于异步电路实现的神经形态芯片则是使用AER协议表示脉冲事件，两者的差异导致CPU/GPU上的训练和神经形态芯片上推理的精度不一致。尽管增大时间步数可以降低两者的差异，但训练代价会显著增加。关于这一问题的研究还较少，Yao等[54]通过在单步释放多个脉冲来降低误差并在Speck芯片上进行了验证。设计异步仿真算法可能是解决这一问题的关键，目前已经有初步工作[205]进行了尝试。

现有的梯度替代算法，通常使用任务性能和理论功耗作为评估指标，而对硬件部署的可行性未作过多考量。但需要注意的是，SNN的目标运行设备并非CPU或GPU，而是低功耗的神经形态硬件。因此，未来的研究者们可以更多地考虑硬件约束，例如在有限内存和精度的条件下设计高性能SNN神经元和网络架构、软件模拟硬件特性等，将模型量化、网络剪枝等技术引入，同时充分考虑硬件非理想性、同步仿真和异步部署的差异，进行软件-硬件协

同设计(Software Hardware Codesign)，提升SNN的片上推理性能，这一设计理念也是诸多SNN研究者所倡导的[31,204]。

6　总结与展望

本文介绍了基于梯度替代法直接训练的深度脉冲神经网络学习算法研究进展，将已有算法进行分类，并详细介绍和比较。本文还选取了各类梯度替代算法中的代表性方法，并在多个数据集和任务上对比了它们的多个性能指标。根据前文的系统性梳理和对比实验结果，现对各类方法现状和可能的改进方向总结如下：

(1) 基础学习算法是梯度替代法训练SNN的基石，但目前关于不同替代函数优劣、网络收敛条件等理论分析较少，多数已有的研究都基于实验性结论，需要研究者们重视。

(2) 编码方式在ANN2SNN中研究较多，而梯度替代法能够直接进行端到端训练，因而研究者很少手工设计网络内部编码方式。但目前的SNN在时间步数极小时性能会剧烈下降，人为设置网络内部的时间编码规则或许能够解决这一问题，从而实现超低延迟的SNN推理。降低时间步数的另一优势是，BPTT训练的开销也会大幅下降，因而对编码方式的研究也有助于降低SNN的训练开销。

(3) 神经元和突触改进方法通常会不可避免地增加模型的复杂度，甚至引入一些难以在现有硬件上实现的操作，例如CLIF神经元[95]中的Sigmoid激活函数涉及硬件上昂贵的指数运算；Sliding PSN[71]作为k阶神经元需要k个历史输入的存储消耗；有状态的突触[99]也需要额外的资源存储和更新突触上电流的状态，且在训练时会显著增加内存消耗。因而，未来的研究中应更多地考虑神经元在GPU上的并行加速算法和神经形态硬件兼容性，以增强模型的训练速度和实用性。

(4) 网络结构改进方法已经取得了较大成功，但也有一些遗留问题尚未解决，例如前文已经讨论过的SEW ResNet[50]和MS ResNet[103]的硬件部署问题。此外，目前SNN网络结构设计整体思路仍然延续自ANN，而生物神经系统中的反馈连接、侧向抑制等特性尚未得到探索，这些特殊的结构可能是实现人脑级别通用人工智能的关键，有待进一步探索。

(5) 正则化方法中较新的BNTT[147]、TEBN[148]等方法使用逐时刻的参数，因此要求输入序列的长度不可变，在处理实际任务时可能不够灵活，这一问题有待改进。此外，目前BN类方法较多，而其他方法较少。考虑到脉冲化的Transformer架构目前性能更高，而ANN中的结论已经表明Transformer架构使用LN性能更好，故未来的研究可更多聚焦于LN的变体在SNN中的应用。原始的LN无法与卷积层合并，这一问题仍有待解决。此外，也可考虑针对脉冲神经元的特性，设计诸如NeuNorm[144]类型的SNN专用正则化方法。

(6) ANN辅助训练算法中基于ANN耦合的算法梯度误差较大，但其相较于普通梯度替代法能够避免BPTT的巨大内存消耗量，值得进一步研究。其本质可以认为是使用脉冲在时间上的累计来计算梯度，因而未来的研究方向可以聚焦于设计低误差的脉冲累计表示方法。基于ANN蒸馏的算法则主要存在计算代价高、超参数数量多且调试困难的缺陷需要改进。两类方法均不能用于时域任务，根源在于ANN不具有时间维度，而通过循环神经网络或Transformer辅助训练或许能够解决这一问题。

(7)事件驱动学习算法适合硬件实现，但目前研究还处在初级阶段，实际性能较为落后，且对超参数敏感、稳定性差，存在很大改进空间。值得注意的是，事件驱动算法使用脉冲传递梯度的这一特性，更适合基于稀疏计算的实现方式，即在前向传播和反向传播时使用脉冲发放时刻表示脉冲。而目前的事件驱动仿真方式仍然使用基于二值张量的方式表示脉冲，无脉冲的位置表示成0，存在很大的表示冗余。如何设计一套稀疏加速仿真方式，也是值得整个领域内研究者们重视的话题。

(8) 在线学习算法有望解决SNN使用BPTT训练内存消耗量过大的问题，且适合在神经形态硬件上进行片上学习。该类方法目前在静态数据集上表现优秀，但对时域任务还不能很好地处理，未来的相关研究可对此问题重点关注。

(9) 训练加速方法中SpikingJelly[49]框架加速效果较好且通用性最强，但其加速思路更类似于加速RNN，没有充分利用脉冲的二值量化、稀疏激活特性；稀疏脉冲梯度下降[167]则一定程度上利用了SNN的稀疏特性，但其受限于工程难度，只在MLP上进行了实验，没有在更常用的卷积架构上实现。研究者们如果能够充分利用SNN的特性，通过稀疏计算降低计算量和内存消耗，通过二值脉冲和浮点权重的混合精度运算提升计算速度，则SNN相较于ANN的低功耗优势或许能从仅推理阶段延伸到更

具实用价值的训练阶段，从而彻底解决现有人工智能训练成本高昂的难题，这将使得SNN的科学和应用价值进一步提升。

从宏观视角来看，作为神经科学和计算科学融合产物的脉冲深度学习领域，梯度替代类学习方法目前的灵感和方法论多来自深度学习已有的研究范式，技术路线与量化神经网络、循环神经网络、微型机器学习等领域也存在一定重合。这种研究范式是一把双刃剑，既带来了SNN性能的快速提升，也不可避免地引入了深度学习的固有缺陷，例如依赖大量有标注数据并进行多轮训练才能达到较好效果，而人脑则可以仅使用少量样本高效学习；在新任务上学习会引发旧任务的性能骤降，即灾难性遗忘(Catastrophic Forgetting)，而人脑则擅长利用已有经验迅速学习新任务，具有很强的迁移学习(Transfer Learning)能力；对随机扰动敏感，容易被对抗攻击(Adversarial Attack)诱导，而人类对攻击样本则展现出惊人的正确率和鲁棒性[206]。

考虑到神经科学在人工智能发展中的历史地位，以及人脑仍是已知最智能的系统这一现实，模仿大脑的结构功能和运行原理来设计SNN学习算法，或许能够解决传统深度学习方法面临的挑战，并推动人工智能领域取得新一次重大进展。令人欣喜的是，已经有部分研究者在这一方向进行了探索，例如通过突触可塑性[183,207-208]或脉冲神经元的阈值调节[209]，缓解灾难性遗忘，提升持续学习和小样本学习能力；利用循环连接[70]、超极化电流(After-hyperpolarizing Current)[179]、精细神经元模型[177]、非局部的突触可塑性[210]、兴奋型/抑制型神经元[70]、突触延迟[100]等生理结构和机制改善网络记忆能力、任务性能或参数效率；利用离散的脉冲发放过程和神经元的积分泄露机制[211]来增强对抗攻击的鲁棒性。这类研究在神经科学的指引下，充分利用了SNN独有的神经动态和突触可塑性机制，设计脑启发的结构和算法，并在特定任务上超越了传统深度学习方法的性能，开辟了有别于传统深度学习方法之外的研究道路，值得后来的学者参考。

参 考 文 献

[1] Yann LeCun, Yoshua Bengio, Geoffrey Hinton. Deep learning. Nature, 2015, 521(7553): 436-444

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Delving deep into rectifiers: surpassing human-level performance on imagenet classification//Proceedings of the IEEE/CVF International Conference on Computer Vision. Santiago, Chile, 2015: 1026-1034

[3] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, et al. Going deeper with convolutions//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Santiago, Chile, 2015: 1-9

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep residual learning for image recognition//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Las Vegas, USA, 2016: 770-778

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, et al. An image is worth 16x16 words: transformers for image recognition at scale//Proceedings of the International Conference on Learning Representations. Vienna, Austria, 2021

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik. rich feature hierarchies for accurate object detection and semantic segmentation//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Columbus, USA, 2014: 580-587

[7] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. You only look once: unified, real-time object detection//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Las Vegas, USA, 2016: 779-788

[8] Alex Graves, Abdel-Rahman Mohamed, Geoffrey Hinton. Speech recognition with deep recurrent neural networks// IEEE International Conference on Acoustics, Speech and Signal Processing. Vancouver, Canada, 2013: 6645-6649

[9] Alex Graves, Navdeep Jaitly, Abdel-Rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm//Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding. Olomouc, Czech Republic, 2013: 273-278

[10] Ilya Sutskever, Oriol Vinyals, Quoc V. Le. Sequence to sequence learning with neural networks//Advances in Neural Information Processing Systems. Montreal, Canada, 2014, 27

[11] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural machine translation by jointly learning to align and translate// Proceedings of the International Conference on Learning Representations. San Diego, USA, 2015

[12] Rico Sennrich, Barry Haddow, Alexandra Birch. Neural machine translation of rare words with subword units// Annual Meeting of the Association for Computational Linguistics. Berlin, Germany, 2016: 1715-1725

[13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, et al. human-level control through deep reinforcement learning. Nature, 2015, 518(7540): 529-533

[14] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, et al. Mastering the game of go without human knowledge. Nature, 2017, 550 (7676): 354-359

[15] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, et al. Language models are few-shot learners//Proceedings of the Advances in Neural Information Processing Systems. Virtual, 2020, 33: 1877-1901

[16] Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, et al. PanGu-α: large-scale autoregressive pretrained chinese language models with auto-parallel computation, arXiv, 2021: 2104.12369

[17] Openai, Josh Achiam, Steven Adler, Agarwal, Sandhini Lama Ahmad, Ilge Akkaya, et al. GPT-4 technical report, arXiv, 2024: 2303.08774

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, et al. Generative adversarial nets//Proceedings of the Advances in Neural Information Processing Systems. Montreal, Canada, 2014, 27

[19] Alec Radford, Luke Metz, Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks//Proceedings of the International Conference on Learning Representations. San Juan, Puerto Rico, 2016

[20] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, Bjorn Ommer. High-resolution image synthesis with latent diffusion models// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New Orleans, USA, 2022: 10684-10695

[21] Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, Matthew Botvinick. Neuroscience-inspired artificial intelligence. Neuron, 2017, 95(2): 245-258

[22] Anthony Zador, Sean Escola, Blake Richards, Bence Lveczky, Yoshua Bengio, Kwabena Boahen, et al. Catalyzing next-generation artificial intelligence through neuroai. Nature Communications, 2023, 14(1): 1597

[23] Frank Rosenblatt. The Perceptron: a Probabilistic model for information storage and organization in the brain. Psychological Review, 1958, 65(6): 386

[24] David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams. Learning representations by back-propagating errors. Nature, 1986, 323(6088): 533-536

[25] Corinna Cortes, Vladimir Vapnik. Support-vector networks. Machine Learning, 1995, 20(3): 273-297

[26] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. Neural Networks, 1997, 10(9): 1659-1671

[27] Marc-Oliver Gewaltig, Markus Diesmann. Nest (neural simulation tool). Scholarpedia, 2007, 2(4): 1430

[28] Chris Eliasmith, Terrence C. Stewart , Xuan Choo, Trevor Bekolay, Travis Dewolf, Yichuan Tang, Daniel Rasmussen. A large-scale model of the functioning brain. science, 2012, 338 (6111): 1202-1205

[29] Marcel Stimberg, Romain Brette, Dan F. M. Goodman. Brian 2, an intuitive and efficient neural simulator. eLife, 2019, 8: e47314

[30] Carver Mead. Neuromorphic electronic systems. Proceedings of the IEEE, 1990, 78(10): 1629-1636

[31] Kaushik Roy, Akhilesh Jaiswal, Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. Nature, 2019, 575(7784): 607-617

[32] Patrick Lichtsteiner, Christoph Posch, Tobi Delbruck. A 128x128 120 db 15us latency asynchronous temporal contrast vision sensor. Proceedings of the IEEE Journal of Solid-State Circuits, 2008, 43(2): 566-576

[33] Siwei Dong, Tiejun Huang, Yonghong Tian. Spike camera and its coding methods//Proceedings of the Data Compression Conference. Snowbird, USA, 2017: 437-437

[34] Paul A. Merolla, John V. Arthur, Rodrigo Alvarez-Icaza, Andrew S. Cassidy , Jun Sawada, Filipp Akopyan, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. science, 2014, 345 (6197): 668-673

[35] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, et al. Loihi: a neuromorphic manycore processor with on-chip learning. IEEE Micro, 2018, 38(1): 82-99

[36] De Ma, Juncheng Shen, Zonghua Gu, Ming Zhang, Xiaolei Zhu, Xiaoqiang Xu, et al. Darwin: a neuromorphic hardware co-processor based on spiking neural networks. Journal of Systems Architecture, 2017, 77: 43-51

[37] Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, et al. Towards artificial general intelligence with hybrid tianjic chip architecture. Nature, 2019, 572(7767): 106-111

[38] Geoffrey E. Hinton, Simon Osindero, Yee-Whye Teh. A fast learning algorithm for deep belief nets. Neural Computation, 2006, 18(7): 1527-1554

[39] Yann LeCun, Lon Bottou, Yoshua Bengio, Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11): 2278-2324

[40] Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep learning[m]. MIT Press, 2016

[41] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks//Proceedings of the Advances in Neural Information Processing Systems. Lake Tahoe, USA, 2012, 25

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, et al. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 2015, 115(3): 211-252

[43] Emre O. Neftci, Hesham Mostafa, Friedemann Zenke. Surrogate gradient learning in spiking neural networks: bringing the power of gradient-based optimization to spiking neural networks. IEEE Signal Processing Magazine, 2019, 36(6): 51-63

[44] Yongqiang Cao, Yang Chen, Deepak Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. International Journal of Computer Vision, 2015, 113(1): 54-66

[45] Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothe Masquelier, Anthon Maiday. deep learning in spiking

neural networks. Neural Networks, 2019, 111: 47-63

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

[47] Friedemann Zenke, Surya Ganguli. Superspike: supervised learning in multilayer spiking neural networks. Neural Computation, 2018, 30(6): 1514-1541

[48] Sumit Bam Shrestha, Garrick Orchard. Slayer: spike layer error reassignment in time//Proceedings of the Advances in Neural Information Processing Systems. Montreal, Canada, 2018: 1419-1428

[49] Wei Fang, Yanqi Chen, Jianhao Ding, Zhaofei Yu, Timothée Masquelier, Ding Chen, et al. Spikingjelly: an open-source machine learning infrastructure platform for spike-based intelligence. Science Advances, 2023, 9(40): eadi1480

[50] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothe Masquelier, Yonghong Tian. Deep residual learning in spiking neural networks//Proceedings of the Advances in Neural Information Processing Systems. 2021, 34

[51] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang Y. A. N. Shuicheng, Yonghong Tian, Li Yuan. Spikformer: when spiking neural network meets transformer//Proceedings of the International Conference on Learning Representations. Kigali, Rwanda, 2023

[52] Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, Guoqi Li. Spike-driven transformer//Proceedings of the Advances in Neural Information Processing Systems. New Orleans, USA, 2023, 36: 64043-64058

[53] Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, et al. Attention spiking neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(8): 9393-9410

[54] Man Yao, Ole Richter, Guangshe Zhao, Ning Qiao, Yannan Xing, Dingheng Wang, et al. Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip. Nature Communications, 2024, 15(1): 4464

[55] Sergey Ioffe, Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift//Proceedings of the International Conference on Machine Learning. Lille, France, 2015: 448-456

[56] Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton . Layer normalization, arxiv, 2016: 1607.06450

[57] Eugene M. Izhikevich. Simplemodel of spiking neurons. IEEE Transactions on Neural Networks, 2003, 14(6): 1569-1572

[58] Eimantas Ledinauskas, Julius Ruseckas, Alfonsas Jurnas, Giedrius Buraas. Training deep spiking neural networks, arXiv, 2020: 2006.04436

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

[60] Han Xiao, Kashif Rasul, Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, arXiv, 2017: 1708.07747

[61] Alex Krizhevsky, Geoffrey Hinton. Learning multiple layers of features from tiny images, 2009

[62] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothe Masquelier, Tiejun Huang, Yonghong. Tian incorporating learnable membrane time constant to enhance learning of spiking neural networks//Proceedings of the IEEE/CVF International Conference on Computer Vision. Virtual, 2021: 2661-2671

[63] Garrick Orchard, Ajinkya Jayawant, Gregory K. Cohen, Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. Frontiers in Neuroscience, 2015, 9

[64] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, Luping Shi. Cifar10-dvs: an event-stream dataset for object classification. Frontiers in Neuroscience, 2017, 11

[65] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey Mckinstry, Carmelo Di Nolfo, et al. A low power, fully event-based gesture recognition system//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Honolulu, USA, 2017: 7243-7252

[66] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, Yiannis Andreopoulos. Graph-based object classification for neuromorphic vision sensing//Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul, Republic of Korea, 2019: 491-501

[67] Yihan Lin, Wei Ding, Shaohua Qiang, Lei Deng, Guoqi Li. Es-imagenet: a million event-stream classification dataset for spiking neural networks. Frontiers in Neuroscience, 2021, 15

[68] Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, Friedemann Zenke. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(7): 2744-2757

[69] Laxmi R. Iyer, Yansong Chua, Haizhou Li. Is neuromorphic mnist neuromorphic? analyzing the discriminative power of neuromorphic datasets in the time domain. Frontiers in Neuroscience, 2021, 15

[70] Bojian Yin, Federico Corradi, Sander M. Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. Nature Machine Intelligence, 2021, 3 (10): 905-913

[71] Wei Fang, Zhaofei Yu, Zhaokun Zhou, Ding Chen, Yanqi Chen, Zhengyu Ma, et al. Parallel spiking neurons with high efficiency and ability to learn long-term dependencies//Proceedings of the Advances in Neural Information Processing Systems. New Orleans, USA, 2023: 53674-53687

[72] Mark Horowitz. 1.1 Computing's energy problem (and what we can do about it)//Proceedings of the IEEE International Solid-State Circuits Conference Digest of Technical Papers. 2014: 10-14

[73] Zheyu Yang, Taoyi Wang, Yihan Lin, Yuguo Chen, Hui Zeng, Jing Pei, et al. A vision chip with complementary pathways for open-world sensing. Nature, 2024, 629(8014): 1027-1033

[74] Yoshua Bengio, Nicholas Lonard, Aaron Courville. estimating or propagating gradients through stochastic neurons for

conditional computation, arXiv, 2013: 1308.3432

[75] Julia Gygax, Friedemann Zenke. Elucidating the theoretical underpinnings of surrogate gradient learning in spiking neural networks, arXiv, 2024: 2404.14964

[76] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, Yoshua Bengio. Quantized neural networks: training neural networks with low precision weights and activations. Journal of Machine Learning Research, 2018, 18(187): 1-30

[77] Friedemann Zenke, Tim P. Vogels . The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. Neural Computation, 2021, 33(4): 899-925

[78] Shuang Lian, Jiangrong Shen, Qianhui Liu, Ziming Wang, Rui Yan, Huajin Tang. Learnable surrogate gradient for direct training spiking neural networks//Proceedings of the International Joint Conference on Artificial Intelligence. Macao, China, 2023: 3002-3010

[79] Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, Shi Gu. Differentiable spike: rethinking gradient-descent for training spiking neural networks//Proceedings of the Advances in Neural Information Processing Systems. Virtual, 2021

[80] Kaiwei Che, Luziwei Leng, Kaixuan Zhang, Jianguo Zhang, Qinghu Meng, Jie Cheng, et al. Differentiable hierarchical and surrogate gradient search for spiking neural networks// Proceedings of the Advances in Neural Information Processing Systems. New Orleans, United States, 2022, 35: 24975-24990

[81] Chankyu Lee, Syed Shakib Sarwar, Priyadarshini Panda, Gopalakrishnan Srinivasan, Kaushik Roy. Enabling spike-based backpropagation for training deep neural network architectures. Frontiers in Neuroscience, 2020, 14

[82] Xiang Cheng, Yunzhe Hao, Jiaming Xu, Bo Xu. Lisnn: improving spiking neural networks with lateral interactions for robust object recognition//Proceedings of the International Joint Conference on Artificial Intelligence. Virtually, 2020: 1519-1525

[83] Saeed Reza Kheradpisheh, Timothe Masquelier. Temporal backpropagation for spiking neural networks with one spike per neuron. International Journal of Neural Systems, 2020, 30(06): 2050027

[84] Ana Stanojevic, Stanisław Woźniak, Guillaume Bellec, Giovanni Cherubini, Angeliki Pantazi, Wulfram Gerstner. High-performance deep spiking neural networks with 0.3 spikes per neuron. Nature Communications, 2024, 15(1): 6793

[85] Nitin Rathi, Kaushik Roy. Diet-snn: a low-latency spiking neural network with direct input encoding andleakage and threshold optimization. IEEE Transactions on Neural Networks and Learning Systems, 2021, 34(6): 3174-3182

[86] Bodo Rueckauer, Shih-Chii Liu. Conversion of analog to spiking neural networks using sparse temporal coding// Proceedings of the IEEE International Symposium on Circuits and Systems. Florence, Italy, 2018: 1-5

[87] Lei Zhang, Shengyuan Zhou, Tian Zhi, Zidong Du, Yunji Chen. Tdsnn: from deep neural networks to deep spike neural networks with temporal-coding//Proceedings of the AAAI Conference on Artificial Intelligence. Honolulu, USA, 2019, 33 (01): 1319-1326

[88] Ana Stanojevic, Stanisław Woźniak, Guillaume Bellec, Giovanni Cherubini, Angeliki Pantazi, Wulfram Gerstner. An exact mapping from Relu networks to spiking neural networks. Neural Networks, 2023, 168: 74-88

[89] Jaehyun Kim, Heesu Kim, Subin Huh, Jinho Lee, Kiyoung Choi. Deep neural networks with weighted spikes. Neurocomputing, 2018, 311: 373-386

[90] Z. Wang, X. Gu, R. S. M.Goh, J. T.Zhou, T.Luo. Efficient spiking neural networks with radix encoding. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(3): 3689-3701

[91] Yuhang Li, Youngeun Kim, Hyoungseob Park, Priyadarshini. Panda uncovering the representation of spiking neural networks trained with surrogate gradient. transactions on machine learning research, 2023: 2835-8856

[92] Jiakui Hu, Man Yao, Xuerui Qiu, Yuhong Chou, Yuxuan Cai, Ning Qiao, et al. High-performance temporal reversible spiking neural networks with o(l) training memory and o(1) inference cost//Proceedings of the International Conference on Machine Learning. Vienna, Austria, 2024, 235: 19516-19530

[93] Xingting Yao, Fanrong Li, Zitao Mo, Jian Cheng. Glif: a unified gated leaky integrate-and-fire neuron for spiking neural networks//Proceedings of the Advances in Neural Information Processing Systems. New Orleans, USA, 2022: 32160-32171

[94] Lang Feng, Qianhui Liu, Huajin Tang, De Ma, Gang Pan. Multi-level firing with spiking ds-resnet: enabling better and deeper directly-trained spiking neural networks//Proceedings of the International Joint Conference on Artificial Intelligence. Vienna, Austria, 2022: 2471-2477

[95] Yulong Huang, Xiaopeng Lin, Hongwei Ren, Haotian Fu, Yue Zhou, Zunchang Liu, et al. Clif: complementary leaky integrate-and-fire neuron for spiking neural networks//Proceedings of the International Conference on Machine Learning. Vienna, Austria, 2024: 24

[96] Sidi Yaya Arnaud Yarga, Sean U. N. Wood. accelerating snn training with stochastic parallelizable spiking neurons// Proceedings of the International Joint Conference on Neural Networks. Gold Coast, Australia, 2023: 1-8

[97] Hesham Mostafa, Bruno U. Pedroni , Sadique Sheik, Gert Cauwenberghs. Fast classification using sparsely active spiking networks//Proceedings of the IEEE International Symposium on Circuits and Systems. Baltimore, USA, 2017: 1-4

[98] Hesham Mostafa. Supervised learning based on temporal coding in spiking neural networks. IEEE Transactions on Neural Networks and Learning Systems, 2017, 29(7): 3227-3235

[99] Haowen Fang, Amar Shrestha, Ziyi Zhao, Qinru Qiu. Exploiting neuron and synapse filter dynamics in spatial temporal learning of deep spiking neural network//Proceedings of the International Joint Conference on Artificial Intelligence. Virtual, 2020: 2799-2806

[100] Ilyass Hammouamri, Ismail Khalfaoui-Hassani, Timothe

Masquelier. Learning delays in spiking neural networks using dilated convolutions with learnable spacings//Proceedings of the International Conference on Learning Representations. Vienna, Austria, 2024

[101] Yangfan Hu, Huajin Tang, Gang Pan. Spiking deep residual networks. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(8): 5200-5205

[102] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, Guoqi Li. Going deeper with directly-trained larger spiking neural networks// Proceedings of the AAAI Conference on Artificial Intelligence. Virtual, 2021, 35: 11062-11070

[103] Yifan Hu, Lei Deng, Yujie Wu, Man Yao, Guoqi Li. Advancing spiking neural networks toward deep residual learning. IEEE Transactions on Neural Networks and Learning Systems, 2024, 36(2): 2353-2367

[104] Alex Graves. Generating sequences with recurrent neural networks, 2014

[105] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, et al. AttentioniIs all you need//Proceedings of the Advances in Neural Information Processing Systems. Long Beach, USA, 2017, 30

[106] Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, Guoqi Li. Temporal-wise attention spiking neural networks for event streams classification// Proceedings of the IEEE/CVF International Conference on Computer Vision. Virtual, 2021: 10221-10230

[107] Man Yao, Jiakui Hu, Guangshe Zhao, Yaoyuan Wang, Ziyang Zhang, Bo Xu, Guoqi Li. Inherent redundancy in spiking neural networks//Proceedings of the IEEE/CVF International Conference on Computer Vision. Paris, France, 2023: 16924-16934

[108] Man Yao, Hengyu Zhang, Guangshe Zhao, Xiyu Zhang, Dingheng Wang, Gang Cao, Guoqi Li. Sparser spiking activity can be better: feature refine-and-mask spiking neural network for event-based visual recognition. Neural Networks, 2023, 166: 410-423

[109] Qi Xu, Yuyuan Gao, Jiangrong Shen, Yaxin Li, Xuming Ran, Huajin Tang, Gang Pan. Enhancing adaptive history reserving by spiking convolutional block attention module in recurrent neural networks//Proceedings of the Advances in Neural Information Processing Systems. New Orleans, USA, 2023, 36: 58890-58901

[110] Rui-Jie Zhu, Malu Zhang, Qihang Zhao, Haoyu Deng, Yule Duan, Liang-Jian Deng. Tcja-snn: temporal-channel joint attention for spiking neural networks. IEEE Transactions on Neural Networks and Learning Systems, 2024, 36(3): 5112-5125

[111] Ziyuan Huang, Shiwei Zhang, Liang Pan, Zhiwu Qing, Mingqian Tang, Ziwei Liu, Marcelo H. Ang Jr. Tada! temporally-adaptive convolutions for video understanding// Proceedings of the International Conference on Learning Representations. Virtual 2022

[112] Xiaolong Wang, Ross Girshick, Abhinav Gupta, Kaiming He. Non-local neural networks//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, USA, 2018: 7794-7803

[113] Sangyeob Kim, Soyeon Kim, Seongyon Hong, Sangjin Kim, Donghyeon Han, YooHoi-Jun. C-dnn: a 24.5-85.8 tops/w complementary-deep-neural-network processor with heterogeneous cnn/snn core architecture and forward-gradient-based sparsity generation//Proceedings of the IEEE International Solid-State Circuits Conference. San Francisco, USA, 2023: 334-336

[114] Muya Chang, Ashwin Sanjay Lele, Samuel D. Spetalnick, Brian Crafton, Shota Konno, Zishen Wan, et al. A heterogeneous rram in-memory and sram near-memory soc for fused frame and event-based target identification and tracking// Proceedings of the IEEE International Solid-State Circuits Conference. San Francisco, USA, 2023: 426-428

[115] Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, Xin Yang. Spiking transformers for event-based single object tracking//Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. New Orleans, USA, 2022: 8801-8810

[116] Jiyuan Zhang, Lulu Tang, Zhaofei Yu, Jiwen Lu, Tiejun Huang. Spike transformer: monocular depth estimation for spiking camera// European Conference on Computer Vision. Tel Aviv, Israel, 2022: 34-52

[117] Minglun Han, Qingyu Wang, Tielin Zhang, Yi Wang, Duzhen Zhang, Bo Xu. Complex dynamic neurons improved spiking transformer network for efficient automatic speech recognition// Proceedings of the AAAI Conference on Artificial Intelligence. Washington,USA, 2023, 37(1): 102-109

[118] Xinyu Shi, Zecheng Hao, Zhaofei Yu. Spikingresformer: bridging resnet and vision transformer in spiking neural networks//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, USA, 2024: 5610-5619

[119] Chenlin Zhou, Han Zhang, Zhaokun Zhou, Liutao Yu, Liwei Huang, Xiaopeng Fan, et al. Qkformer: hierarchical spiking transformer using q-k attention//Proceedings of the Advances in Neural Information Processing Systems. Vancouver, Canada, 2024, 37: 13074-13098

[120] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, Humphrey Shi. Escaping the big data paradigm with compact transformers, arXiv, 2022: 2104.05704

[121] Man Yao, Jiakui Hu, Tianxiang Hu, Yifan Xu, Zhaokun Zhou, Yonghong Tian, et al. Spike-driven transformer v2: meta spiking neural network architectureinspiring the design of next-generation neuromorphic chips//Proceedings of the International Conference on Learning Representations. Vienna, Austria, 2024

[122] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, et al. Swin transformer: hierarchical vision transformer using shifted windows//Proceedings of the IEEE/CVF International Conference on Computer Vision. Virtual, 2021: 10012-10022

[123] Changze Lv, Dongqi Han, Yansen Wang, Xiaoqing Zheng, Xuanjing Huang, Dongsheng Li. Advancing spiking neural

networks for sequential modeling with central pattern generators//Proceedings of the Advances in Neural Information Processing Systems. Vancouver, Canada, 2024, 37: 26915-26940

[124] Zhaokun Zhou, Yijie Lu, Yanhao Jia, Kaiwei Che, Jun Niu, Liwei Huang, et al. Spiking transformer with experts mixture// Proceedings of the Advances in Neural Information Processing Systems, Vancouver, Canada, 2024

[125] Byunggook Na, Jisoo Mok, Seongsik Park, Dongjin Lee, Hyeokjun Choe, Sungroh Yoon. Autosnn: towards energy-efficient spiking neural networks//Proceedings of the International Conference on Machine Learning. Baltimore, USA, 2022, 162: 16253-16269

[126] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, Jeff Dean. Efficient neural architecture search via parameter sharing// Proceedings of the International Conference on Machine Learning. Stockholm, Sweden, 2018: 4095-4104

[127] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, Song Han. Once-for-all: train one network and specialize it for efficient deployment, arXiv, 2019: 1908.09791

[128] Guobin Shen, Dongcheng Zhao, Yiting Dong, Yi Zeng. Brain-inspired neural circuit evolution for spiking neural networks. Proceedings of the National Academy of Sciences, 2023, 120(39): e2218173120

[129] Norimitsu Suzuki, John M. Bekkers . Microcircuits mediating feedforward and feedback synaptic inhibition in the piriform cortex. Journal of Neuroscience, 2012, 32(3): 919-931

[130] Jiaqi Yan, Qianhui Liu, Malu Zhang, Lang Feng, De Ma, Haizhou Li, Gang Pan. Efficient spiking neural network design via neural architecture search. Neural Networks, 2024, 173: 106172

[131] Seijoon Kim, Seongsik Park, Byunggook Na, Sungroh Yoon. Spiking-yolo: spiking neural network for energy-efficient object detection// Proceedings of the AAAI Conference on Artificial Intelligence. New York, USA, 2020, 34: 11270-11277

[132] Y. Hu, Q. Zheng, X. Jiang, G. Pan. Fast-snn: fast spiking neural network by converting quantized ann. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(12): 14546-14562

[133] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, et al. Microsoft coco: common objects in context, Cham, 2014: 740-755

[134] Pierre De Tournemire, Davide Nitti, Etienne Perot, Davide Migliore, Amos Sironi. A large scale event-based detection dataset for automotive, arXiv, 2020: 2001.08499

[135] Loïc Cordone, Benoît Miramond, Philippe Thierion. Object detection with spiking neural networks on automotive event data//Proceedings of the International Joint Conference on Neural Networks. Padua, Italy, 2022: 1-8

[136] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, Kilian Q. Weinberger. densely connected convolutional networks// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Honolulu, USA, 2017: 4700-4708

[137] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, ryad benosman. hats: histograms of averaged time surfaces for robust event-based object classification// Proceedings of the IEEE/CVF International Conference on Computer Vision. Salt Lake City, USA, 2018: 1731-1740

[138] Hong Zhang, Yang Li, Bin He, Xiongfei Fan, Yue Wang, Yu Zhang. Direct training high-performance spiking neural networks for object recognition and detection. Frontiers in Neuroscience, 2023, 17

[139] Qiaoyi Su, Yuhong Chou, Yifan Hu, Jianing Li, Shijie Mei, Ziyang Zhang, Guoqi Li. Deep directly-trained spiking neural networks for object detection//Proceedings of the IEEE/CVF International Conference on Computer Vision. Paris, France, 2023: 6532-6542

[140] Yimeng Fan, Wei Zhang, Changsong Liu, Mingyang Li, Wenrui Lu. Sfod: spiking fusion object detector//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Washington, USA, 2024: 17191-17200

[141] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie. Feature pyramid networks for object detection//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Honolulu, USA, 2017: 936-944

[142] Jintang Li, Zhouxin Yu, Zulun Zhu, Liang Chen, Qi Yu, Zibin Zheng, et al. Scaling up dynamic graph representation learning via spiking neural networks//Proceedings of the AAAI Conference on Artificial Intelligence. Washington, USA, 2023, 37(7): 8588-8596

[143] Dayong Ren, Zhe Ma, Yuanpei Chen, Weihang Peng, Xiaode Liu, Yuhan Zhang, Yufei Guo. Spiking pointnet: spiking neural networks for point clouds//Proceedings of the Advances in Neural Information Processing Systems. Vancouver, Canada, 2024, 36: 41797-41808

[144] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, Luping Shi. Direct training for spiking neural networks: faster, larger, better//Proceedings of the AAAI Conference on Artificial Intelligence. Honolulu, USA, 2019, 33: 1311-1318

[145] Matteo Carandini, David Heeger J.. Normalization as a canonical neural computation. Nature Reviews Neuroscience, 2012, 13(1): 51-62

[146] Valerio Mante, Vincent Bonin, Matteo Carandini. Functional mechanisms shaping lateral geniculate responses to artificial and natural stimuli. Neuron, 2008, 58(4): 625-638

[147] Youngeun Kim, Priyadarshini Panda. Revisiting batch normalization for training low-latency deep spiking neural networks from scratch. Frontiers in Neuroscience, 2021, 15

[148] Chaoteng Duan, Jianhao Ding, Shiyan Chen, Zhaofei Yu, Tiejun Huang. Temporal effective batch normalization in spiking neural networks//Proceedings of the Advances in Neural Information Processing Systems. New Orleans, USA, 2022, 35: 34377-34390

[149] Yufei Guo, Yuhan Zhang, Yuanpei Chen, Weihang Peng, Xiaode Liu, Liwen Zhang, et al. Membrane potential batch normalization for spiking neural networks//Proceedings of the IEEE/CVF International Conference on Computer Vision.

2023: 19420-19430

[150] Yufei Guo, Xiaode Liu, Yuanpei Chen, Liwen Zhang, Weihang Peng, Yuhan Zhang, et al. Rmp-loss: regularizing membrane potential distribution for spiking neural networks//Proceedings of the IEEE/CVF International Conference on Computer Vision. Paris, France, 2023: 17391-17401

[151] Shikuang Deng, Yuhang Li, Shanghang Zhang, Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting//Proceedings of the International Conference on Learning Representations. Virtual, 2022

[152] Yuhang Li, Youngeun Kim, Hyoungseob Park, Tamar Geller, Priyadarshini Panda. Neuromorphic data augmentation for training spiking neural networks//Proceedings of the European Conference on Computer Vision. Tel Aviv, Israel, 2022: 631-649

[153] Jibin Wu, Yansong Chua, Malu Zhang, Guoqi Li, Haizhou Li, Kay Chen Tan. A tandem learning rule for effective training and rapid inference of deep spiking neural networks. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(1): 446-460

[154] Saeed Reza Kheradpisheh, Maryam Mirsadeghi, Timothe Masquelier. Spiking neural networks trained via proxy. IEEE Access, 2022, 10: 70769-70778

[155] Qi Xu, Yaxin Li, Jiangrong Shen, Jian K. Liu, Huajin Tang, Gang Pan. Constructing deep spiking neural networks from artificial neural networks with knowledge distillation//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Vancouver, Canada, 2023: 7886-7895

[156] Haonan Qiu, Munan Ning, Zeyin Song, Wei Fang, Yanqi Chen, Tao Sun, et al. Self-architectural knowledge distillation for spiking neural networks. Neural Networks, 2024, 178: 106475

[157] Wenrui Zhang, Peng Li. Temporal spike sequence learning via backpropagation for deep spiking neural networks//Proceedings of the Advances in Neural Information Processing Systems. Virtual, 2020: 12022-12033

[158] Yaoyu Zhu, Zhaofei Yu, Wei Fang, Xiaodong Xie, Tiejun Huang, Timothe Masquelier. Training spiking neural networks with event-driven backpropagation//Proceedings of the Advances in Neural Information Processing Systems. New Orleans, USA, 2022, 35: 30528-30541

[159] Yaoyu Zhu, Wei Fang, Xiaodong Xie, Tiejun Huang, Zhaofei Yu. Exploring loss functions for time-based training strategy in spiking neural networks//Proceedings of the Advances in Neural Information Processing Systems. Vancouver, Canada, 2024, 36

[160] Jacques Kaiser, Hesham Mostafa, Emre Neftci. Synaptic plasticity dynamics for deep continuous local learning (decolle). Frontiers in Neuroscience, 2020, 14: 424

[161] Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Di He, Zhouchen Lin. Online training through time for spiking neural networks//Proceedings of the Advances in Neural Information Processing Systems. New Orleans, USA, 2022, 35: 20717-20730

[162] Qingyan Meng, Mingqing Xiao, Shen Yan, Yisen Wang, Zhouchen Lin, Zhi-Quan Luo. Training high-performance low-latency spiking neural networks by nifferentiation on spike representation//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New Orleans, USA, 2022: 12444-12453

[163] Qingyan Meng, Mingqing Xiao, Shen Yan, Yisen Wang, Zhouchen Lin, Zhi-Quan Luo. Towards memory-and time-efficient backpropagation for training spiking neural networks//Proceedings of the IEEE/CVF International Conference on Computer Vision. Paris, France, 2023: 6166-6176

[164] Haiya Jiangn, Giulia De Masi, Huan Xiong, Bin Gu. Ndot: neuronal dynamics-based online training for spiking neural networks//Proceedings of the International Conference on Machine Learning. Vienna, Austria, 2024, 235: 21806-21823

[165] Yaoyu Zhu, Jianhao Ding, Tiejun Huang, Xiaodong Xie, Zhaofei Yu. Online stabilization of spiking neural networks//Proceedings of the International Conference on Learning Representations. Vienna, Austria, 2024

[166] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, Saining Xie. A convnet for the 2020s//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New Orleans, USA, 2022: 11976-11986

[167] Nicolas Perez-Nieves, Dan F. M. Goodman. Sparse spiking gradient descent//Proceedings of the Advances in Neural Information Processing Systems. Virtual, 2021, 34: 11795-11808

[168] Ryosuke Okuta, Yuya Unno, Daisuke Nishino, Shohei Hido, Crissman Loomis. Cupy: a numpy-compatible library for nvidia gpu calculations//Proceedings of the Advances in Neural Information Processing Systems. Long Beach, USA, 2017

[169] Luke Taylor, Andrew King, Nicol S. Harper. Addressing the speed-accuracy simulation trade-off for adaptive spiking neurons//Proceedings of the Advances in Neural Information Processing Systems. New Orleans, USA, 2023, 36: 59360-59374

[170] Yufei Guo, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Yinglei Wang, Xuhui Huang, Zhe Ma. Im-loss: information maximization loss for spiking neural networks//Proceedings of the Advances in Neural Information Processing Systems. New Orleans, USA, 2022, 35: 156-166

[171] David H. Wolpert. The lack of a priori distinctions between learning algorithms. Neural Computation, 1996, 8(7): 1341-1390

[172] Edgar D. Adrian, Yngve Zotterman. The impulses produced by sensory nerve endings: part 3. impulses set up by touch and pressure. The Journal of Physiology, 1926, 61(4): 465

[173] Roland S. Johansson, Ingvars Birznieks. First spikes in ensembles of human tactile afferents code complex spatial fingertip events. Nature Neuroscience, 2004, 7(2): 170-177

[174] Fred Rieke, David Warland, Rob De Ruyter Van Steveninck, William Bialek. Spikes: exploring the neural code. USA: MIT Press, 1999

[175] Seongsik Park, Seijoon Kim, Hyeokjun Choe, Sungroh Yoon. Fast and efficient information transmission with burst spikes in deep spiking neural networks//Proceedings of the 56 th Annual Design Automation Conference 2019. Las Vegas, USA, 2019: Article 53

[176] Yang Li, Yi Zeng. Efficient and accurate conversion of spiking neural network with burst spikes//Proceedings of the International Joint Conference on Artificial Intelligence. Vienna, Austria, 2022: 2485-2491

[177] Linxuan He, Yunhui Xu, Weihua He, Yihan Lin, Yang Tian, Yujie Wu, et al. Network model with internal complexity bridges artificial intelligence and neuroscience. Nature Computational Science, 2024, 36(6): 1-16

[178] Charles D. Gilbert, Wu Li. Top-down influences on visual processing. Nature Reviews Neuroscience, 2013, 14(5): 350-363

[179] Arjun Rao, Philipp Plank, Andreas Wild, Wolfgang Maass. A long short-term memory for ai applications in spike-based neuromorphic hardware. Nature Machine Intelligence, 2022, 4(5): 467-479

[180] Donald Olding Hebb. The organization of behavior: a neuropsychological theory. psychology press, 1949

[181] Guo-Qiang Bi, Mu-Ming Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. Journal of Neuroscience, 1998, 18(24): 10464-10472

[182] Nabil Imam, Thomas A. Cleland. Rapid online learning and robust recall in a neuromorphic olfactory circuit. Nature Machine Intelligence, 2020, 2(3): 181-191

[183] Yujie Wu, Rong Zhao, Jun Zhu, Feng Chen, Mingkun Xu, Guoqi Li, et al. Brain-inspired global-local learning incorporated with neuromorphic computing. Nature Communications, 2022, 13(1): 1-14

[184] Rachmad Vidya Wicaksana Putra, Muhammad Shafique. Q-spinn: a framework for quantizing spiking neural networks// Proceedings of the International Joint Conference on Neural Networks. Virtual, 2021: 1-8

[185] Saeed Reza Kheradpisheh, Maryam Mirsadeghi, Timothe Masquelier. Bs4nn: binarized spiking neural networks with temporal coding and learning. Neural Processing Letters, 2022, 54(2): 1255-1273

[186] Ayan Shymyrbay, Mohammed E. Fouda , Ahmed Eltawil. low precision quantization-aware training in spiking neural networks with differentiable quantization tunction//Proceedings of the International Joint Conference on Neural Networks, Gold Coast, Australia, 2023: 1-8

[187] Wenjie Wei, Yu Liang, Ammar Belatreche, Yichen Xiao, Honglin Cao, Zhenbang Ren, et al. Q-snns: quantized spiking neural networks//Proceedings of the ACM International Conference on Multimedia. 2024: 8441-8450

[188] LeYa, Xuan Yang. Tiny imagenet visual recognition challenge. CS 231N, 2015, 7(7): 3

[189] Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons// Proceedings of the Advances in Neural Information Processing Systems. Montréal, Canada, 2018, 31

[190] Lei Deng, Yujie Wu, Yifan Hu, Ling Liang, Guoqi Li, Xing Hu, et al. Comprehensive snn compression using admm optimization and activity regularization. IEEE Transactions on Neural Networks and Learning Systems, 2021, 34(6): 2791-2805

[191] Yanqi Chen, Zhaofei Yu, Wei Fang, Tiejun Huang, Yonghong Tian. Pruning of deep spiking neural networks through gradient rewiring//Proceedings of the International Joint Conference on Artificial Intelligence. Virtual, 2021: 1713-1721

[192] Yanqi Chen, Zhaofei Yu, Wei Fang, Zhengyu Ma, Tiejun Huang, Yonghong Tian. State transition of dendritic spines improves learning of sparse spiking neural networks//Proceedings of the International Conference on Machine Learning. Baltimore, USA, 2022: 3701-3715

[193] Youngeun Kim, Yuhang Li, Hyoungseob Park, Yeshwanth Venkatesha, Ruokai Yin, Priyadarshini Panda. Exploring lottery ticket hypothesis in spiking neural networks// Proceedings of the European Conference on Computer Vision. Tel Aviv, Israel, 2022: 102-120

[194] Yanqi Chen, Zhengyu Ma, Wei Fang, Xiawu Zheng, Zhaofei Yu, Yonghong Tian. A unified framework for soft threshold pruning//Proceedings of the International Conference on Learning Representations. Kigali, Rwanda, 2023

[195] Emre O. Neftci, Bruno U. Pedroni, Siddharth Joshi, Maruan Al-Shedivat, Gert Cauwenberghs. Stochastic synapses enable efficient brain-inspired learning machines. Frontiers in Neuroscience, 2016, 10: 241

[196] Nitin Rathi, Priyadarshini Panda, Kaushik Roy. Stdp-based pruning of connections and weight quantization in spiking neural networks for energy-efficient recognition. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 38(4): 668-677

[197] Yuhan Shi, Leon Nguyen, Sangheon Oh, Xin Liu, Duygu Kuzum. A soft-pruning method applied during training of spiking neural networks for in-memory computing applications. Frontiers in Neuroscience, 2019, 13: 405

[198] Yu Qi, Jiangrong Shen, Yueming Wang, Huajin Tang, Hang Yu, Zhaohui Wu, Gang Pan. Jointly learning network connections and link weights in spiking neural networks// Proceedings of the International Joint Conference on Artificial Intelligence. Stockholm, Sweden, 2018: 1597-1603

[199] Yaxin Li, Jiangrong Shen, Hongming Xu, Long Chen, Gang Pan, Qiang Zhang, Qi Xu. Towards efficient deep spiking neural networks construction with spiking activity based pruning, arXiv, 2024: 2406.01072

[200] Xinyu Shi, Jianhao Ding, Zecheng Hao, Zhaofei Yu. Towards energy efficient spiking neural networks: an unstructured pruning framework//Proceedings of the International Conference on Learning Representations. Vienna, Austria, 2024

[201] Ben Varkey Benjamin, Peiran Gao, Emmett Mcquinn, Swadesh Choudhary, Anand R. Chandrasekaran, Jean-Marie

Bussat, et al. Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations. Proceedings of the IEEE, 2014, 102(5): 699-716

[202] Abhiroop Bhattacharjee, Youngeun Kim, Abhishek Moitra, Priyadarshini Panda. Examining the robustness of spiking neural networks on non-ideal memristive crossbars// Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design. Boston, USA, 2022: 1-6

[203] Filippo Moro, E. Esmanhotto, T. Hirtzlin, N. Castellani, A. Trabelsi, T. Dalgaty, et al. Hardware calibrated learning to compensate heterogeneity in analog rram-based spiking neural networks//Proceedings of the IEEE International Symposium on Circuits and Systems. Austin, USA, 2022: 380-383

[204] Dennis V. Christensen, Regina Dittmann, Bernabe Linares-Barranco, Abu Sebastian, Manuel Le Gallo, Andrea Redaelli, et al. 2022 Roadmap on neuromorphic computing and engineering. Neuromorphic Computing and Engineering, 2022, 2(2): 022501

[205] Roel Koopman, Amirreza Yousefzadeh, Mahyar Shahsavari, Guangzhi Tang, Manolis Sifalakis. Overcoming the limitations of layer synchronization in spiking neural networks, arXiv, 2024: 2408.05098

[206] Zhenglong Zhou, Chaz Firestone. Humans can decipher adversarial images. Nature Communications, 2019, 10(1): 1334

[207] Tielin Zhang, Xiang Cheng, Shuncheng Jia, Chengyu T. Li, Mu-Ming Poo, Bo Xu. A brain-inspired algorithm that mitigates catastrophic forgetting of artificial and spiking neural networks with low computational Cost. Science Advances, 2023, 9(34): eadi2947

[208] Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Di He, Zhouchen Lin. Hebbian learning based orthogonal projection for continual learning of spiking neural networks//Proceedings of the International Conference on Learning Representations. Vienna, Austria, 2024: 2835-8856

[209] Ilyass Hammouamri, Timothe Masquelier, Dennis George Wilson. Mitigating catastrophic forgetting in spiking neural networks through threshold modulation. Transactions on Machine Learning Research, 2022,21(8):181-196

[210] Tielin Zhang, Xiang Cheng, Shuncheng Jia, Mu-Ming Poo, Yi Zeng, Bo Xu. Self-backpropagation of synaptic modifications elevates the efficiency of spiking and artificial neural networks. Science Advances, 2021, 7(43): eabh0146

[211] Saima Sharmin, Nitin Rathi, Priyadarshini Panda, Kaushik Roy. Inherent adversarial robustness of deep spiking neural networks: effects of discrete input encoding and non-linear activations//Proceedings of the European Conference on Computer Vision. Glasgow, UK, 2020: 399-414

[212] Ilya Loshchilov, Frank Hutter. Sgdr: stochastic gradient descent with warm restarts//Proceedings of the International Conference on Learning Representations. Toulon, France, 2017

[213] Leslie N Smith, Nicholay Topin. Super-convergence: nery fast training of neural networks using large learning rates. Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, 2019, 11006: 369-386

附录

1 可复现性

本文的实验代码、训练日志可以从如下网址获取，方便读者自行复现或进一步研究：

https://github. com/fangwei123456/chinese_snn_surrogate_gradient_review

2 实验细节

2.1 突触操作数统计

突触操作数的统计是在整个测试集上完成的，汇报的结果为平均到每个样本的突触操作数。需要注意的是，当突触处理非脉冲的整数类型输入时，例如神经形态数据集中的事件积分得到的帧，或由SEW残差连接导致的脉冲之和，突触操作数按照输入整数的数值计算。例如，当输入为脉冲[0,1,0,1]则突触操作数为2；而输入为非负整数[0,2,0,3]则突触操作数按5计算。

2.2 CIFAR分类实验细节

本文使用Fang等[71]的网络结构，记卷积层为Conv，批量标准化层为BN，脉冲神经元层为SN，平均池化层为AP，全连接层为FC，则网络结构为：(Conv-BN-SN)-(Conv-BN-SN)-(Conv-BN-SN)-AP-(Conv-BN-SN)-(Conv-BN-SN)-(Conv-BN-SN)-AP-FC-SN-FC,其中卷积层的通道数固定为128。对于静态CIFAR10分类任务，使用2维卷积，第一个FC层输出特征数为2048；对于序列CIFAR10分类任务，使用1维卷积，第一个FC层输出特征数为256。两个任务的网络结构中第二个FC层输出特征均为10，与总类别数一致。

数据增强方法包括Mixup($p=1,\alpha=0.2$)、Cutmix($p=1,\alpha=1$)、随机擦除($p=0.1$)、自动数据增强(Trivial Augment)、标签平滑($p=0.1$)、数据标准化。

对于训练超参数，在不做出特殊说明的情况下，默认批量大小为128、训练轮数256、使用动量大小为0.9的SGD优化器、学习率0.1、混合精度训练、周期为训练轮数的余弦退火学习率调节器[212]。对于静态CIFAR10分类任务，时间步数为4；对于序列CIFAR10分类任务，时间步数与图像宽度相同，为32。

对于不同方法，额外调节的超参数通常是通过网格搜索确定的，具体如下：

响应蒸馏：蒸馏损失的强度为0.001，温度为4。作为教师的ANN是将SNN中脉冲神经元换成ReLU激活函数，并使用默认超参数训练出来的。

特征蒸馏：蒸馏损失的强度为0.1，温度为1。作为教师的ANN与响应蒸馏中使用的是同一个ANN。

Tandem：使用学习率为0.001。

PSN家族：静态CIFAR10分类任务使用PSN，序列CIFAR10分类任务使用$k=4$的Sliding PSN。

Block ALIF：分块大小为1。使用更大的分块大小，则性能会

剧烈下降。

2.3　SHD分类实验细节

本文使用Ilyass等[100]的开源代码和网络结构。对于神经形态的SHD语音数据集，使用SpikingJelly框架[49]中的固定时间间隔积分方式，每20 ms积分为一帧，产生帧数为88~126、特征数为140的帧。使用3层全连接网络，具体结构为：(FC256-BN-SN-DP) - (FC256-BN-SN-DP) -FC20-LIF, 其中FC256表示输出特征数量为256的全连接层，FC20表示输出特征数量为20的全连接层，DP表示Dropout层，且丢弃率均为0.4。SN表示脉冲神经元层，根据不同方法使用不同的脉冲神经元，而最后输出层固定为LIF神经元层，且阈值设置为无穷大，网络的输出是其所有时间步的膜电位。

对于训练超参数，默认训练150轮，对于LIF神经元和其他有膜时间常数的神经元均初始化膜时间常数为1.005，使用SpikingJelly框架[49]中$\alpha=5$的ArcTan替代函数，使用Adam优化器，学习率0.001，权重衰减0.000 01，使用最大学习率为0.005、周期为训练轮数的OneCycle学习率调节器[213]。

对于不同方法，额外调节的超参数如下：

Sliding PSN：使用$k=3$。

TEBN：由于实验观测到IF神经元表现反而比LIF神经元好，故使用IF神经元配合TEBN。

BlockALIF：分块大小为1。

需要注意的是，TEBN和BlockALIF神经元要求固定时间步数，不能处理变长输入，而本实验中使用的固定时间间隔积分方式会产生长度不固定的输入帧。因而，在实验中将它们的时间步数都设置为帧的最大长度126，并在计算时将长度不足126的输入进行0填充，输出再重新去除填充部分。

2.4　Gen1目标检测实验细节

本文使用Fan等[140]的开源代码和网络结构，使用Spiking DenseNet121-16作为检测的骨干网络，首先在神经形态的NCAR[137]数据集上预训练分类任务。分类任务的默认超参数为：数据集积分到5帧（故时间步数为5），批量大小为64，使用AdamW优化器，学习率为0.005，余弦退火学习率调节器（周期和训练轮数一致，最小学习率0.000 01），训练轮数30并设置了早停机制（Early Stop，当连续5轮训练的最小损失没有改善则退出训练），启用混合精度训练。网络结构为标准的Spiking DenseNet121-16。

分类任务预训练完成后，卷积层部分权重作为目标检测任务的骨架网络的初始权重，并在Gen1数据集上进行训练，默认超参数为：数据集积分到5帧（故时间步数为5），训练轮数为50，使用AdamW优化器、学习率0.001、权重衰减0.0001、余弦退火学习率调节器（周期和训练轮数一致，最小学习率0.000 01），启用混合精度训练。需要注意的是，BlockALIF神经元的参数是逐神经元的，而该任务中，预训练分类任务和其后的目标检测任务，输入尺寸不一致。因而加载预训练权重时，卷积层可以正常加载，而BlockALIF神经元则由于参数尺寸不匹配无法加载，参数按照默认方式初始化。

对于不同方法，需要额外调节的超参数如下：

CLIF神经元：学习率0.0001。

Sliding PSN：使用$k=3$。

TEBN：实验发现LIF神经元好于IF神经元，故使用LIF神经元配合TEBN。

OSR：学习率0.0001。

BlockALIF：学习率0.0001。

2.5　COCO目标检测实验细节

本文使用Su等[139]提出的EMS ResNet进行直接训练。用于比较的各类方法，除方法本身的超参数外和训练轮数外，其余超参数均与Su等[139]保持一致，具体为：使用ResNet-10结构，批量大小为16，图片尺寸为640×640，使用动量为0.937的SGD优化器、OneCycle学习率调节器[213]、初始学习率为0.01、最终学习率为0.1、权重衰减0.0005，预热训练轮数为3，预热训练时动量为0.8、学习率为0.1，启用混合精度训练。由于COCO数据集图片分辨率较大，训练速度较慢，且本文需要对比多种方法，为减少实验成本，本文将训练轮数从Su等[139]设置的300轮减少为100轮。

对于不同方法，需要额外调节的超参数如下：

Sliding PSN：使用$k=3$。

TEBN：实验发现IF神经元好于LIF神经元，故使用LIF神经元配合TEBN。

2.6　加速性能对比的原始数据

正文表5对比了不同方法的加速比。各个方法的原始耗时数据，在表6中进行展示。

表6　不同加速方法的耗时（ms）

T	SpikingJelly	PSN	BlockALIF分组大小				LIF
			2	4	8	16	
2	1.40	0.65	7.08				1.44
4	2.03	0.74	17.38	7.87			3.02
8	1.76	0.70	32.18	16.34	8.04		4.79
16	1.53	0.75	42.90	32.48	17.06	7.37	9.48
32	1.03	0.97	67.67	42.99	28.90	16.90	17.14
64	2.06	0.70	125.98	68.75	42.40	31.16	30.60

FANG Wei, Ph. D., research assistant professor. His research interest is spiking neural networks.

ZHU Yao-Yu, Ph. D., assistant professor. His research interest is brain-inspired computation.

HUANG Zi-Han, Ph. D. candidate. His research interests are spiking neural networks.

YAO Man, Ph. D., assistant professor. His research interest is neuromorphic computing.

YU Zhao-Fei, Ph. D., assistant professor. His research interests include neuromorphic computing and computational neuroscience.

TIAN Yong-Hong, Ph. D., professor. His research interests include video big data analytics and brain-inspired computation.

Background

Artificial Neural Networks (ANNs) monopolize the current Artificial Intelligence (AI) systems for their higher performance than other computational models. However, the floating activation and intensive computation of ANNs cause high energy consumption. Spiking Neural Networks(SNNs), the third generation of neural network models, are the potential alternatives of ANNs for up to hundreds of times of power efficiency. Modules in SNNs communicate by asynchronous spikes as the human brain, which introduces sparse activations, event-driven computations, and low power consumption.

However, there is still a huge performance gap between SNNs and ANNs, which restricts the practical values of SNNs. Complex temporal dynamics and non-differentiable firing mechanisms make it challenging to design learning methods for SNNs. Traditional bio-inspired learning methods such as the Hebbian rule and the Spike Timing Dependent Plasticity rule are unsupervised algorithms and can only solve simple learning tasks such as classifying the MNIST dataset. Primitive supervised learning methods including SpikeProp, Tempotron, and ReSuMe are limited to train SNNs with a single layer or single spike. Recently, deep learning methods have been introduced into SNNs and overwhelmed previous algorithms, growing into the booming spiking deep learning research community.

The ANN to SNN conversion and surrogate learning methods are two mainstream methods in spiking deep learning. The former is based on rate coding and approximates the activations in ANNs by firing rates in SNNs. However, it requires the SNNs to run many time steps and causes high energy consumption and long latency. It cannot solve temporal tasks because the time dimension is already occupied to represent rates. On the contrary, the surrogate learning methods are more flexible. It re-defines the gradient of the discrete Heaviside function used in spike generation by that of a smooth surrogate function and then is capable of training SNNs directly. It is not based on rate coding and can fully utilize neural dynamics to process temporal tasks such as classifying the neuromorphic data. It is not restricted to rate coding and requires much fewer time steps than the conversion methods.

This survey reviews the latest research advancements of the surrogate learning methods in spiking deep learning. The basic concepts, components, and benchmarks of SNNs are first introduced. Then learning methods are systemically divided into different categories and illustrated. A comprehensive experiment is conducted to compare these methods fairly. The advantages and shortcomings of each category are then presented. Lastly, the future research directions are discussed.

This work is partially supported by the National Natural Science Foundation of China under contracts No. 62425101, No. 62332002, No. 62027804, No. 62088102 and No. 62406322.