深度脉冲神经网络梯度替代学习算法研究综述

方维[1)] 朱耀宇[2)] 黄梓涵[3)] 姚满[4)] 余肇飞[6),3)] 田永鸿[1),3),5)]

[1)](北京大学深圳研究生院信息工程学院，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)](鹏城实验室，深圳，518000)
[6)](北京大学人工智能研究院，北京，100871)

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，但也因复杂的时域动态、离散不可导的脉冲发放过程难以训练. 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、ANN 辅助训练算法、神经元和突触改进、网络结构改进、正则化方法、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，讨论了各类方法的优缺点，并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 TP DOI 号: * 投稿时不提供 DOI 号

Review of surrogate learning methods in deep spiking neural networksReview of surrogate learning methods in deep spiking neural networks

Wei Fang[1)] Yaoyu Zhu[2)] Zihan Huang[3)] Man Yao[4)] Zhaofei Yu[6),3)] Yonghong Tian[1),3),5)]

[1)] School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, Shenzhen, 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)](Peng Cheng Laboratory, Shenzhen, 518000)
[6)](Institute for Artificial Intelligence, Peking University, Beijing, 100871)

Abstract Spiking Neural Networks (SNNs) are regarded as the third generation of neural network models with binary communication, sparse activation, event-driven, and extremely power-efficient characteristics. However, the training of SNNs is difficult because of their complex temporal dynamics and non-differentiable firing mechanisms.

收稿日期： - - ；最终修改稿收到日期： - - . 本课题得到国家自然科学基金 (No.62425101, 62332002, 62027804, 62088102) 资助. **方维**，男，博士，助理研究员，主要研究领域为脉冲深度学习.E-mail: fwei@pku.edu.cn. **朱耀宇**，男，博士，特任研究助理，主要研究领域为类脑计算.E-mail: zhuyaoyu@ict.ac.cn. **黄梓涵**，男，博士研究生，主要研究领域为神经形态计算.E-mail: hzh@stu.pku.edu.cn. **姚满**，男，博士，特任研究助理，主要研究领域为神经形态计算.E-mail: man.yao@ia.ac.cn. **余肇飞**，男，博士，助理教授，主要研究领域为计算机视觉、神经形态计算和计算神经科学. E-mail: yuzf12@pku.edu.cn. **田永鸿**，博士，教授，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、机器学习、类脑计算. E-mail: yhtian@pku.edu.cn. 第 1 作者手机号码: 13041160166, E-mail: fangwei123456g@gmail.com.

Recently, deep learning methods, including the surrogate gradient methods and the Artificial Neural Networks (ANN) to SNN conversion methods, have proposed and promoted the performance of SNNs greatly, which develops the Spiking Deep Learning area. This article focuses on the advances of surrogate gradient methods and reviews the basic learning methods, ANN-auxiliary training methods, neuron and synapse model modifications, network structure designs, normalization methods, event-driven learning methods, and training acceleration methods systemically. Finally, the potential breakthrough research topics are prospected.

Keywords Spiking Neural Networks, Surrogate Gradient Methods, Brain-inspired Computing, Neuromorphic Computing, Spiking Deep Learning

1 引言

人工智能在最近十几年取得了跳跃式的发展[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[18-19]等领域取得了突破性进展，引领了新一轮的经济发展和产业变革. 在人工智能的发展过程中，神经科学提供的视野和灵感起到了重要作用[20-21]，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最主要的计算模型.

第一代神经网络又称为感知机 (Perceptron)[22]，接收多个输入并输出布尔值，可以通过训练解决线性分类问题，引发了第一次神经网络热潮. 感知机不能处理非线性的异或 (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[23]实现多层网络的构建和训练，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[24]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[25]，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[26-28]，用于解释和探究生物神经系统的运行原理. 由于复杂的时域动态、离散不可导的脉冲发放过程，训练 SNN 比 ANN 更为困难，因而 SNN 在任务性能为主要导向的人工智能领域的地位一度较为边缘.

神经形态计算 (Neuromorphic Computing)[29-30]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)[31]、视达 (Vidar)[32] 等神经形态视觉传感器和 IBM True North[33]、Intel Loihi[34]、浙江大学达尔文 (Darwin)[35]、清华大学天机芯 (Tianjic)[36] 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，有望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[30]，但这一设想受限于 SNN 高性能学习算法的缓慢发展，一度难以实现.

2006 年 Hinton 等[37] 使用神经网络在 MNIST 数据集[38] 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[39]. 2012 年 Alex 等[40] 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[41] 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 Learning Method)[42] 和 ANN 转换 SNN 方法 (ANN to SNN Conversion, ANN2SNN)[43] 为代表的两大类深度学习方法被应用于 SNN 的训练，大幅提升 SNN 的任务性能至早期 ANN 的水平[44]，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 仿真步数少、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 方法则是将已有的 ANN 转换为 SNN，避免训练开销，转换速度快、任务精度高，但通常基于频率编码，仿真步数多、延迟高且不能用于时域任务. 本文聚焦直接训练方法，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.

2 深度脉冲神经网络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)[45] 和层标准化 (Layer Normalization, LN)[46] 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. 生物神经元接收其他神经元通过树突 (Dendrite) 传递来的输入电信号，累计为自身的膜电位 (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，使用最为广泛的泄露积分发放 (Leaky Integrate-and-Fire, LIF) 神经元的阈下神经动态为：

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

其中 τ_m 是膜时间常数，$V(t)$ 是膜电位，V_{rest} 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 $\Theta(x)$ 描述这一过程：

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

其中 $x \geq 0$ 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}，这一重置过程可以描述为：

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

诸如 Izhikevich 模型[47] 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，通用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[48-49] 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：

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

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

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

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 神经元而言，参考其微分方程 (1) 式，可以得到充电的差分方程为：

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

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[50]. 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 函数的误差更小[51]，因而在 ANN2SNN 中普遍使用.

脉冲深度学习蓬勃发展，大量研究结果不断涌现，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. 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. 常用的静态图像数据集包括 MNIST[38]、Fashion-MNIST[52]、CIFAR[53] 和 ImageNet[41] 数据集，数据集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 (Address Event Representation, AER) 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[48-49,54]. 常用的神经形态数据集包括 N-MNIST[55]、CIFAR10-DVS[56]、DVS Gesture[57]、ASL-DVS[58] N-Caltech101[55]、ES-ImageNet[59]、Spiking Heidelberg Digits (SHD)[60] 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[61] 等指出多数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，序列 (Sequential) 图像分类更为常用[62-63]. 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，因而最终的分类结果能够体现网络的记忆能力.

3 深度脉冲神经网络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 分类这种玩具级别的任务，不具备处理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设备[36,64-65]. 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

SNN 不能直接使用梯度下降和反向传播训练算法的原因在于，脉冲发放过程，即 (5) 式使用的 Heaviside 阶跃函数 $\Theta(x)$，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.Wu 等[54]、Zenke 等[66]、Shrestha 等[67] 在 2018 年分别独立地提出了梯度替代方法，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，而其反向传播的按照 $\frac{\partial S[t]}{\partial (H[t]-V_{th})} = \sigma'(H[t] - V_{th})$. 替代函数 $\sigma(x)$ 通常是连续、光滑的函数，拥有数值正常的梯度，可以视作 $\Theta(x)$ 的近似. 常用的替代函数包括 Rectangular[54]、SuperSpike[66]、ArcTan[48]、Sigmoid 等. 尚无理论明确哪种替代函数是最优的，Zenke 等[68] 通过网格搜索的实验性结论表明，不同的替代函数能达到的最优性能相同，但对超参数的敏感度存在很大差异，因而替代函数的选择对网络训练较为关键.Li 等[69] 使用数值梯度来辅助替代函数形状参数的选取，取得了比常规替代函数更好的训练结果.

3.2 ANN 辅助训练

为了充分利用 ANN 的性能优势与 SNN 的能耗优势，一些研究者通过 ANN 辅助训练来获得高性能的 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.

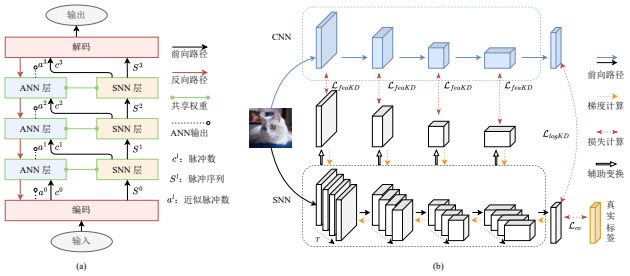


图 X　两类 ANN 辅助训练方法. (a) 共享权重方法. (b) 蒸馏方法

基于共享权重的训练中，Wu 等[70] 和 Kheradpisheh 等[71] 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，通过在 ANN 的反向传播来更新共享权重. 具体而言，Wu 等[70] 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??(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, 通过 ANN 的反向传播计算前一层激活值和权重的梯度：

$$\frac{\partial \mathcal{L}}{\partial a^{l-1}} \approx \frac{\partial \mathcal{L}}{\partial c^{l-1}} = \frac{\partial \mathcal{L}}{\partial a^{l}} \cdot \frac{\partial a^{l}}{\partial c^{l-1}}, \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial W^{l-1}} = \frac{\partial \mathcal{L}}{\partial a^{l-1}} \cdot \frac{\partial a^{l-1}}{\partial W^{l-1}}, \tag{10}$$

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，且不需要在每个时间步都进行复杂的梯度计算.

Kheradpisheh 等[71] 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[70] 在前向传播时将 SNN 脉冲数作为 ANN 层的输入，Kheradpisheh 等[71] 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：

$$\mathcal{L} = -\sum_k Y_k \ln(O_k^A) \approx -\sum_k Y_k \ln(O_k^S), \tag{11}$$

$$\frac{\partial \mathcal{L}}{\partial W_{ji}^l} = \sum_k \frac{\partial \mathcal{L}}{\partial O_k^A} \sum_d \frac{\partial O_k^A}{\partial y_d^L} \frac{\partial y_d^L}{\partial W_{ij}^l} \approx \sum_k \frac{\partial \mathcal{L}}{\partial O_k^S} \sum_d \frac{\partial O_k^S}{\partial y_d^L} \frac{\partial y_d^L}{\partial W_{ij}^l}, \tag{12}$$

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. 共享权重类方法直接避开了 SNN 计算代价高、训练耗时长的反向传播流程，但由于 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.

基于蒸馏的 SNN 训练中，Xu 等[72] 和 Qiu 等[73] 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[72] 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：

$$\mathcal{L}_{KD} = \alpha\tau^2 * \text{CrossEntropy}(Q_S^\tau, Q_T^\tau) + (1-\alpha) * \text{CrossEntropy}(Q$$

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，第 i 个元素 q_i 的计算公式为 $q_i =$

$\frac{exp(Z_i/T)}{\sum_j exp(Z_j/T)}$，$\alpha$ 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 $\mathcal{L}_{distill}$：

$$\mathcal{L}_{KD} = \mathcal{L}_{task} + \alpha \cdot \mathcal{L}_{distill}, \tag{14}$$

$$\mathcal{L}_{distill} = \sum_i (T_i - S_i)^2, \tag{15}$$

其中 T_i 是经过边缘 ReLU 处理以抑制负信息影响后的教师 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 模型的中间层特征.

Qiu 等[73] 通过神经结构搜索 (Neural architecture search, NAS) 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 SAKN，如图**??**(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 $\mathcal{L}_{logKD}$：

$$\mathcal{L}_{all} = \alpha * \mathcal{L}_{ce} + \beta * \mathcal{L}_{feaKD} + \gamma * \mathcal{L}_{logKD}, \tag{16}$$

$$\hat{\mathcal{F}}_s = \mathcal{T}_s(\mathcal{F}_s) = \text{BN}(\text{Conv}(\frac{1}{T}\sum_T \mathcal{F}_s)), \hat{\mathcal{F}}_t = \mathcal{T}_t(\mathcal{F}_t) = \mathcal{F}_t, \tag{17}$$

$$\mathcal{L}_{feaKD} = \left\| \hat{\mathcal{F}}_s - \hat{\mathcal{F}}_t \right\|^2, \tag{18}$$

$$\mathcal{L}_{logKD} = \tau^2 \sum p_\tau^t log(\frac{p_\tau^t}{p_\tau^s}), p_\tau^s(i) = \frac{exp(p^s(i)/\tau)}{\sum exp(p^s/\tau)}, \tag{19}$$

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 表示仿真步数，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，从而允许学生 SNN 模型模仿教师 ANN 模型的特征图. 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，训练代价比普通的替代梯度法更高，但由于 ANN 的指导作用，训练出的网络性能也强于只使用数据集中目标值计算损失的普通 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，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.

PLIF 神经元 (Parametric Leaky Integrate-and-Fire Neuron) 模型[48] 是最早的神经动态可学习的神经元模型之一，其基于经典的 LIF 神经元模型，将膜时间常数 τ_m 参数化并设置为可学习，其神经元的阈下神经动态为：

$$H[t] = V[t-1] + k(a) \cdot \Big(-(V[t-1] - V_{reset}) + X[t] \Big), \tag{20}$$

其中膜时间常数的倒数，即 $\frac{1}{\tau_m}$ 被重参数化为 $\frac{1}{\tau_m} = k(a)$，而 a 是真正的可学习参数.$k(a) \in (0,1)$ 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 $k(a) = \frac{1}{1+\exp(-a)}$.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. GLIF 神经元 (Gated Leaky Integrate-and-Fire Neuron)[74] 进一步扩展了神经动态的学习范围，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：

$$\mathbb{G}_\alpha = (1-\alpha(1-\tau_{exp}))\cdot H[t-1] - (1-\alpha)\tau_{lin}, \tag{21}$$

$$\mathbb{G}_\beta = (1-\beta(1-g[t]))\cdot X[t], \tag{22}$$

$$\mathbb{G}_\gamma = -\gamma\cdot\mathbb{G}_\alpha - (1-\gamma)\cdot V_{reset}, \tag{23}$$

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. MLF 方法 (Multi-Level Firing Method)[75] 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，具有更好的拟合能力. CLIF 神经元 (Complementary Leaky Integrate-and-Fire, Neuron)[76] 旨在解决 LIF 神经元中漏电行为导致的长期梯度衰减问题，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：

$$M[t] = M[t-1]\cdot\sigma\Big(\frac{1}{\tau_m}H[t]\Big) + S[t], \tag{24}$$

$$V[t] = H[t] - S[t]\cdot(V_{th} + \sigma(M[t])), \tag{25}$$

其中 $M[t]$ 表示补充电位，$\sigma(...)$ 是 Sigmoid 激活函数. 公式 (24) 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 (25) 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.

PSN(Parallel Spiking Neurons)[63] 是首个并行脉冲神经元模型，其灵感来自于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[63] 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：

$$H = WX, \qquad W\in\mathbb{R}^{T\times T}, X\in\mathbb{R}^{T\times N} \tag{26}$$

$$S = \Theta(H-B), \quad B\in\mathbb{R}^{T}, S\in\{0,1\}^{T\times N} \tag{27}$$

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 是仿真步数.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[63] 提出 Masked PSN，其对 (26) 式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：

$$H = (W\cdot M_k)X, \qquad W\in\mathbb{R}^{T\times T}, M_k\in\mathbb{R}^{T\times T}, X\in\mathbb{R}^{T\times N} \tag{28}$$

其中 M_k 定义为：

$$M_k[i][j] = \begin{cases} 1, & j\le i\le j+k-1 \\ 0, & \text{其他情况} \end{cases}. \tag{29}$$

PSN 和 Masked PSN 的权重均是逐时刻的，难以处理变长序列.Fang 等[63] 进而将 Masked PSN 的权重设置成时域共享，得到 Sliding PSN，其神经动态为：

$$H[t] = \sum_{i=0}^{k-1} W_i\cdot X[t-k+1+i], \tag{30}$$

$$S[t] = \Theta(H[t]-V_{th}), \tag{31}$$

其中 $W = [W_0, W_1, ..., W_{k-1}] \in \mathbb{R}^k$ 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. 随机并行脉冲神经元[77] 与 PSN 的思路类似，也通过忽略重置来避免膜电位的迭代求解，但其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，其梯度也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更低的理论功耗.AMOS 神经元通常与首达时刻编码 (Time to First Spike Encoding) 结合用于 ANN2SNN 方法[78]，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[79]，Mostafa 等[80] 则是首次将 AMOS 神经元用于深度 SNN，其在训练算法上沿用了之前 Mostafa[81] 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[82] 提出的 S4NN(Single-spike Supervised Spiking Neural Network) 也基于 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，相较于 Mostafa 等[80] 的方法复杂度大幅度降低且任务性能更好.

表**??**总结了部分脉冲神经元改进研究在多个数据集上的仿真步数和分类正确率，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提升和训练速度的降低，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**??**中进行对比.

表 X 脉冲神经元分类任务仿真步数和正确率 (%)

神经元	CIFAR10	CIFAR100	ImageNet	DVS Gestur
PLIF [48]	8|93.50			20|97.57
GLIF [74]	2|94.44 4|94.85 6|95.03	2|75.48 4|77.05 6|77.35	4|67.52 6|69.09	
MLF [75]	4|94.25			40|97.29
CLIF [76]	4|96.01 6|96.45 8|96.69	4|79.69 6|80.58 8|80.89		
PSN 家族 [63]	4|95.32		4|70.54	

深度 SNN 中所使用的突触模型通常与深度 ANN 中一致，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等. Fang 等[83] 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，增强了整个网络在记忆任务上的学习能力. Ilyass 等[84] 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.

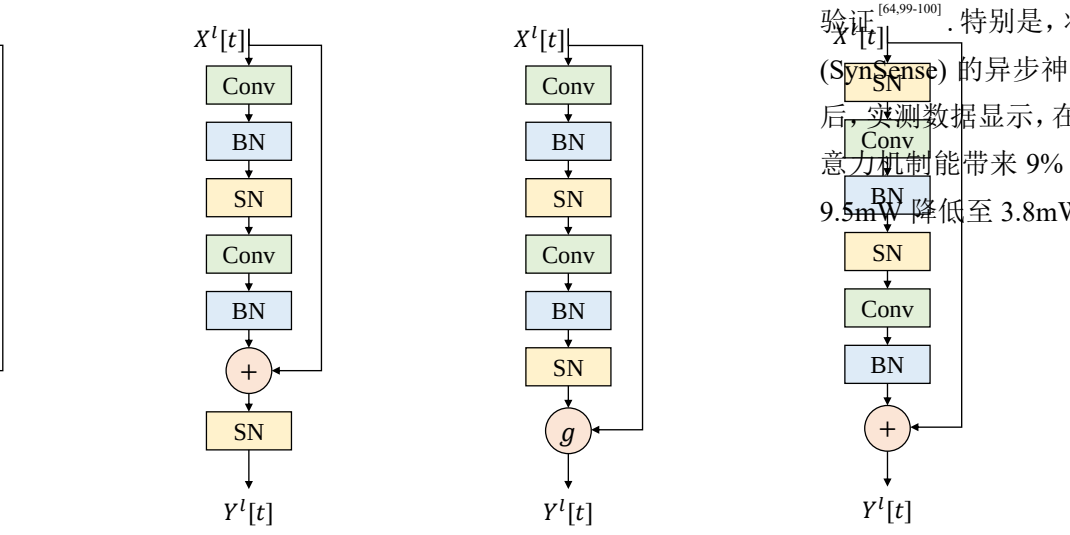
3.4 网络结构改进

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.

替代梯度基础学习算法的出现使得 SNN 领域能够训练 3 至 5 层的浅层卷积网络. 继续采用堆叠卷积层的简单方式来增加网络规模，SNN 的性能会降低. 残差连接[4] 起源于 ResNet[4]，如图**??**(a) 所示，是现代深度神经网络结构中不可缺少的一部分，对

神经网络的规模化起到了至关重要的作用. Spiking ResNet 是 ResNet 的 SNN 版本，最早用于 ANN 转换 SNN[85] 并取得了较好的效果，其结构如图**??**(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[86]，也就是，更深的模型相较于浅层模型，具有更高的训练集误差. Fang 等[87] 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. 为解决这一问题，Spike-Element-Wise (SEW) ResNet[87] 被提出，残差块结构如图**??**(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 ResNet[88] 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**??**(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.

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[11,89-90]. 这一做法在 Spiking ResNet 中同样有效.Yao 等[91] 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[92-94]，或者将注意力机制应用于时间、空间、通道等多个维度[95-96]，SNN 在各种任务中的性能能得到显著提升. 值得一提的是，与注意力 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[64,92-93,95] 对这一现象进行了深入分析.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[97]”，从而导致较差的全局建模能力[98]. 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[92]. 注意力模块能够有效抑制 SNN 中的噪声脉冲，且优化正常特征，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[64,99-100]. 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 Speck 芯片[64] 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 9.5mW 降低至 3.8mW.



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

图 X　常见的残差块结构

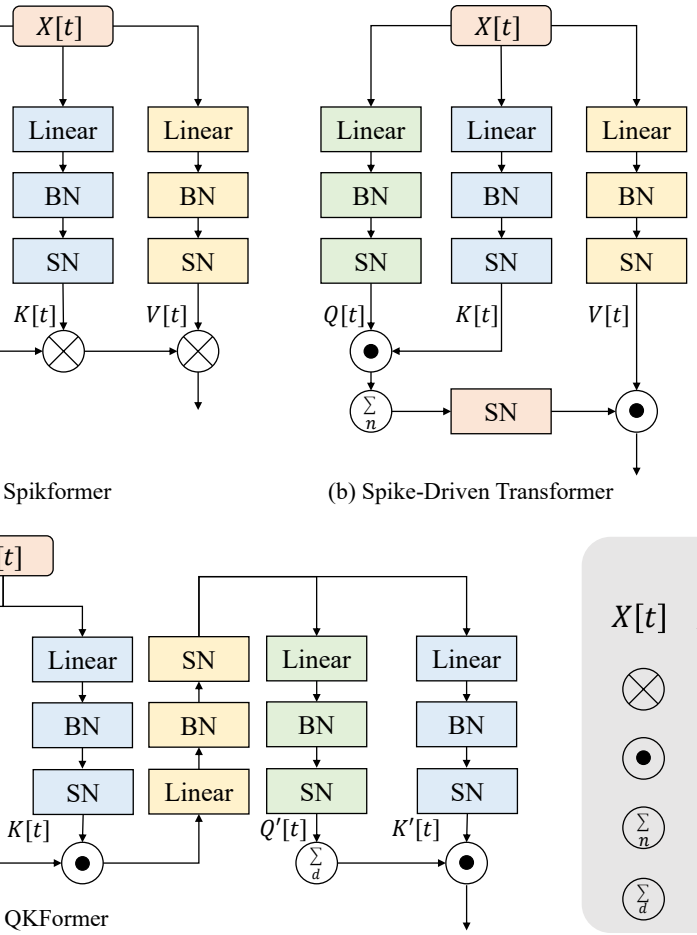


图 X 深度 SNN 中的自注意力机制

以自注意力 (Self Attention) 机制为基础的 Transformer[90] 是另一类典型的深度学习架构，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. 如何有效结合 Transformer 架构的高性能和 SNN 的低功耗引起了领域内学者的广泛兴趣. 较早的 Spiking Transformer[101-103] 的主要设计思路是，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**??**展示了目前 Spiking Transformer 中主流的自注意力机制.

Zhou 等[104] 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[104] 提出了 Spikformer，使用脉冲自注意力 (Spiking Self Attention, SSA) 机制，如图**??**(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 $score$：

$$score[t] = \mathrm{SN}(Q[t]K[t]^{\mathrm{T}}V[t]\cdot s), \tag{32}$$

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[105] 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**??**(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：

$$score[t] = \mathrm{SN}\Big(\sum_{n}(Q[t]K[t])\Big)\cdot V[t], \tag{33}$$

其中 $\sum_n$ 表示沿着分块 (Patch) 的维度求和. 需要注意的是，$\mathrm{SN}(...) \in \{0,1\}^{N\times n_{head}\times d}$ 而 $V[t] \in \{0,1\}^{N\times n_{head}\times n\times d}$，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[106] 使用了双脉冲自注意力机制 (Dual Spike Self Attention, DSSA)，如图**??**(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：

$$\mathrm{DST}(X,Y;f(\cdot))=Xf(Y)=XYW, \tag{34}$$

$$\mathrm{DST}_{\mathrm{T}}(X,Y;f(\cdot))=Xf(Y)^{\mathrm{T}}=XW^{\mathrm{T}}Y^{\mathrm{T}}, \tag{35}$$

其中 $f(\cdot)$ 是 Y 上的广义线性变换，可以是无偏置的线性层、卷积层和 BN 层等.Shi 等[106] 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：

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

$$score[t]=\mathrm{SN}(\mathrm{DST}(\mathrm{AttnMap}(X[t]$$

$$f(X[t])=\mathrm{BN}(\mathrm{Conv}_{\mathrm{p}}(X[t])),$$

其中 c_1,c_2 是缩放因子，BN 是批
是卷积核大小和步长为 p 的卷积.

QKFormer[107] 如其名字所
$Q[t],K[t]$，并通过融合不同维度
结构展示在图**??**(d).QKFormer 首
用脉冲神经元输出的脉冲作为掩

$$score[t]=K[t]\cdot\mathrm{SN}\Big(\sum_{n}($$

其中 $K[t]\cdot\mathrm{SN}(...)$ 用到了广播机制
使用相同的操作在通道维度提取特

$$score'[t]=K'[t]\cdot\mathrm{SN}\Big(\sum_{d}$$

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) 的设计得到了保留.

Spikformer[104] 和 Spike-Driven Transformer[105] 使用 Compact Convolutional Transformer[108] 的网络架构. SpikingResformer[106] 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[109] 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[105] 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[107] 则是使用类似 Swin Transformer[110] 的网络结构.

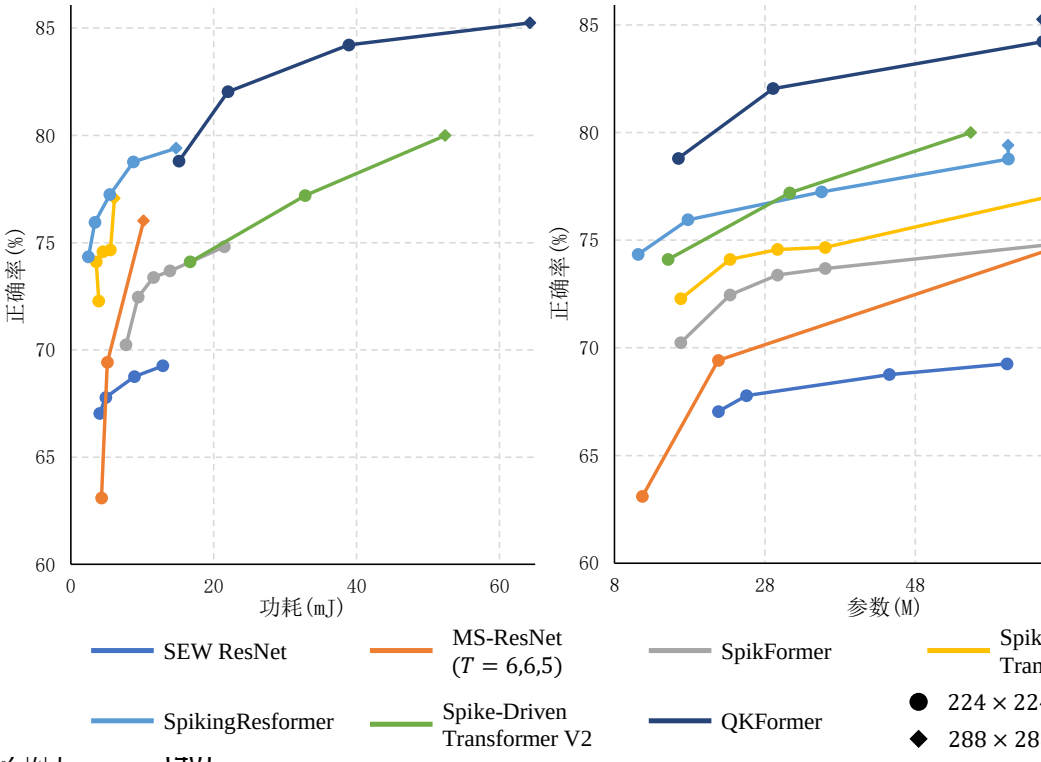


图 X　常见深度 SNN 架构在 ImageNet 数据集的分类正确率、功耗和参数量

图**??**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，其他网络均使用仿真步数 $T=4$；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 图片分辨率推理的结果，在图中以方形点进行了标注. 需要说明的是，图**??**中的功耗皆为理论估算值，其假设 SNN 在推理时若参与计算的一方是脉冲，则脉冲为 0 的位置不

需要计算，脉冲为 1 的位置对应的乘法可以使用加法实现；按照每个乘加操作消耗 $E_{MAC} = 4.6pJ$，每个加法操作则消耗 $E_{AC} = 0.9pJ$[111]；不考虑在内存中读写数据带来的功耗. 图**??**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.

除手动设计网络结构外，也有研究者将网络结构搜索技术引入 SNN，实现自动化的模型设计.Na 等[112] 首次将 NAS 引入 SNN，提出了 spike-aware 优化方程以限制脉冲数量，通过训练超网和使用遗传算法优化 SNN 结构；Kim 等[113] 提出了新的 SNN 框架初始化评估指标，通过这一指标避开训练来搜索合适的 SNN 结构，大幅提升了搜索的速度；Che 等[114] 首次把可微分网络结构搜索方式引入 SNN，直接通过训练代理参数来搜索网络结构，提升了训练速度和性能，同时首次将 SNN 拓展到深度估计等稠密预测领域.

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)[45] 是 SNN 中最为广泛使用的方式. 相较于层标准化 (Layer Normalization)[46] 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，进一步提升了网络的训练效果.NeuNorm[115] 专用于脉冲卷积层，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：

$$O_{norm}[t][i][j] = k_{decay} \cdot O_{norm}[t-1][i][j] + \frac{1-k_{decay}}{C^2} \cdot \sum_{c=0}^{C-1} O[t][c][i][j], \quad (41)$$

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.NeuNorm 对神经元层的输出进行了平滑，可以避免过高或过低的发放率.

在 SNN 中直接使用普通的 BN 层可能会造成一些问题，因而研究者们提出了多种 BN 的变体进行改进，表**??**对目前深度 SNN 中的 BN 类方法进行了总结. 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 $\sigma[t]$ 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, σ_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}, σ_{k+T}，展示在表**??**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)[86] 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，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，因而通过时间批量标准化 (Batch Normalization Through Time, BNTT)[116] 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)[117] 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. SNN 中的正则化层通常被用于卷积层后、神经元前，用于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[118] 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.

表 X 深度 SNN 中的批量标准化类方法

正则化方法	$t=0$	$t=1$	...	$t=T-1$	统计量更新
BN [45]	$\mu[0],\sigma[0]$	$\mu[1],\sigma[1]$		$\mu[T-1],\sigma[T-1]$	$\mu_{k+T}=(1-\rho)^T\mu_k+\sum_{t=0}^{T-1}(1-\rho)^{T-1-t}\rho\mu[t]$ $\sigma_{k+T}=(1-\rho)^T\sigma_k+\sum_{t=0}^{T-1}(1-\rho)^{T-1-t}\rho\sigma[t]$
			γ,β		
TDBN [86]			μ,σ		$\mu_{k+1}=(1-\rho)\mu_k+\rho\mu$ $\sigma_{k+1}=(1-\rho)\sigma_k+\rho\sigma$
			γ,β		
BNTT [116]	$\mu[0],\sigma[0]$	$\mu[1],\sigma[1]$		$\mu[T-1],\sigma[T-1]$	$\mu_{k+1}[t]=(1-\rho)\mu_k[t]+\rho\mu[t],t=0,1,...,T-1$ $\sigma_{k+1}[t]=(1-\rho)\sigma_k[t]+\rho\sigma[t],t=0,1,...,T-1$
	$\beta[0],\gamma[0]$	$\beta[1],\gamma[1]$		$\beta[T-1],\gamma[T-1]$	
TEBN [117]			μ,σ		$\frac{\partial\mathcal{L}}{\partial t_m(s_j^{(l-1)})}=\sum_i\frac{\partial\mathcal{L}}{\partial t_k(s_i^{(l)})}$, $\mu_{k+1}=(1-\rho)\mu_k+\rho\mu$ $\sigma_{k+1}=(1-\rho)\sigma_k+\rho\sigma$ (42)
	$\gamma p[0],\beta p[0]$	$\gamma p[1],\beta p[1]$		$\gamma p[T-1],\beta p[T-1]$	

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[119] 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[120] 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[121] 则对神经形态数据增强进行了探索，通过对常用的变换进行随机选取和组合并施加于神经形态数据集，提升了 SNN 的泛化性能.

3.6 事件驱动学习算法

在 SNN 的学习方法中，事件驱动方法使用网络发放的脉冲传递梯度信息，因而可以节省反向传播次数进而节省反向传播的能耗. 此外，时序梯度只能被脉冲携带，因此事件驱动方法常使用时序梯度. 由于只能依赖脉冲传递梯度信息，事件驱动方法较难训练，网络性能表现普遍不佳.

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[122] 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[123] 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：

$$\sum_j\sum_{t_m(s_j^{(l-1)})}\frac{\partial\mathcal{L}}{\partial t_m(s_j^{(l-1)})}=\sum_i\sum_{t_k(s_i^{(l)})}\frac{\partial\mathcal{L}}{\partial t_k(s_i^{(l)})},\tag{42}$$

其中等式左边是第 $l-1$ 层所有脉冲携带的梯度之和，其中 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[124] 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案，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. 此外，该工作还将权重归一化中所使用的比例因子的训练转移至阈值，提升了网络的性能.

3.7 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了一种实时更新权重的策

略. 这种学习方式避免了通过时间反向传播 (Back Propagation Through Time, BPTT) 需要存储大量中间状态的需求，因此得以节省内存消耗，适用于资源受限或时间步数较多的场景.

Deep Continuous Local Learning (Decolle)[125] 是最早的深度 SNN 在线学习方法之一，其针对双指数 SRM 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. Online Training Through Time (OTTT)[126] 对在线学习方法中层内反向传播进行展开并避免了反向传播中的时间步反向依赖，推导出第 l 层的权重 W^l 上的梯度为：

$$\frac{\partial \mathcal{L}}{\partial W^l}=\sum_{t=0}^{T-1}\frac{\partial \mathcal{L}}{\partial S^l[t]}\frac{\partial S^l[t]}{\partial U^l[t]}\left(\frac{\partial U^l[t]}{\partial W^l}+\sum_{k<t}\prod_{i=k}^{t-1}\epsilon^l[i]\frac{\partial U^l[k]}{\partial W^l}\right) \tag{43}$$

其中 $\epsilon^l[t]=\frac{\partial U^l[t+1]}{\partial U^l[t]}+\frac{\partial U^l[t+1]}{\partial S^l[t]}\frac{\partial S^l[t]}{\partial u^l[t]}$. 随后 OTTT 对 (43) 式进行简化，只保留了 $\epsilon^l[t]$ 中膜电位衰减的部分，忽略了来自未来时刻的梯度，从而使得梯度计算能够避免 BPTT：

$$\frac{\partial \mathcal{L}}{\partial W^l}=\sum_{t=0}^{T-1}\frac{\partial \mathcal{L}}{\partial S^l[t]}\frac{\partial S^l[t]}{\partial U^l[t]}\left(\sum_{k\le t}\lambda^{t-k}\frac{\partial U^l[k]}{\partial W^l}\right). \tag{44}$$

此外，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[127] 之间的正相关性. Spatial Learning Through Time (SLTT)[128] 在 OTTT 的基础上进行了进一步的资源优化.SLTT 随机选取了少量时间步进行反向传播，在其余时间步中省去了反向传播过程，提升了存储效率和计算速度. 而 Neuronal Dynamics-based Online Training (NDOT)[129] 在 OTTT 的基础上对层内的时间依赖性进行了更细致的建模，没有像 OTTT 一样简化 (43) 式，而是将其中的 $\epsilon^l[t]$ 替换为了描述连续时间内膜电位变化的 $e^l[t]=\frac{U^l[t]-V_{th}S^l[t]}{U^l[t-1]-V_{th}S^l[t-1]}$. Zhu 等[130] 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 模块训练时的过程可以用以下公式描述：

$$\hat{I}[t]=\frac{I[t]-\mu[t]}{\sqrt{\sigma^2[t]+\epsilon}}, \tag{45}$$

$$\tilde{I}[t]=\gamma\cdot\left(\hat{I}[t]\cdot\text{NoGrad}\left(\frac{\sqrt{\sigma^2[t]+\epsilon}}{\sqrt{\hat{\sigma^2}+\epsilon}}\right)+\text{NoGrad}\left(\frac{\mu[t]-\hat{\mu}}{\sqrt{\hat{\sigma^2}+\epsilon}}\right)\right) \tag{46}$$

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，$\hat{\mu}$、$\hat{\sigma^2}$ 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，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[131] 则使用了另外一种思路，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，于是在前向传播中关闭了除最后一层外的时序传播过程. 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解决了使用 $\mathcal{O}(1)$ 存储空间记录 $\mathcal{O}(T)$ 步信息的问题，其关键是推导出前一时间步膜电位用后一时间步信息来表示的方法. 此外该工作在网络中使用了 ConvNeXt 块[132]，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表现.

3.8 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 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[133] 在反向传播时，将满足 $|H[t]-V_{th}|\ge B_{th}$ 的神经元视作不活跃的神经元，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 $T \times N \times ...$ 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 $TN \times ...$，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 $T \times N \times ...$ 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 CUDA 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 仿真方式，最高可达 11 倍的训练加速效果. Luke 等[134] 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[63] 一致；由于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，该方法进而对输出脉冲进行修正，仅保留每组时间步内的第一个脉冲. 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，因而本文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，实验类型包括分类任务性能对比和训练加速性能对比.

4.1 分类任务性能

本文使用 Fang 等[63] 的网络结构，测试各类方法分类静态 CIFAR10 和序列 (Sequential) 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.CIFAR10 分类设置 $T = 4$，而序列 CIFAR10 分类的 $T = 32$ 与图片宽度一致；统一使用 128 通道数的卷积层，训练 256 轮；默认使用 SGD 优化器，学习率 0.1，如果网络不收敛则再额外调整优化器和学习率. 参与比较的方法包括 ANN 辅助训练算法中的 Tandem 学习方法[70] 和响应与特征蒸馏[72]、神经元和突触改进算法中的 CLIF 神经元[76] 和 PSN 家族[63]、正则化方法中的 TEBN[117]、在线学习算法中的 OSR[130] 和训练加速算法中的时间分组仿真方式加速的 BlockALIF 神经元[134]. 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，结果如图**??**所示. 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，其它网络均使用 LIF 神经元.BlockALIF 均使用每组 2 个时间步，因每组 1 个时间步则与普通神经元无异，没有任何加速效果；每组更多时间步则性能剧烈下降. 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.

表**??**展示了各类学习算法中的代表性方法的任务性能. 对于序列 CIFAR10 分类，由于 ANN 不能直接处理时域任务，故 ANN 辅助类方法无法使用，在表**??**中留白. 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. 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. 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有效提升了拟合能力. OSR 作为在线学习方法，在 CIFAR10 分类任务上性能反而强于普通网络，但在序列 CIFAR10 分类任务上性能大幅度下降，表明静态任务的梯度较易近似，而动态任务的梯度则很难逼近. BlockALIF 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.

表 X 对比各类代表性方法任务正确率 (%)

	IF	LIF	Tandem [70]	响应蒸馏 [72]	特征蒸馏 [72]	CLIF [76]	PSN 家族 [63]	TEBN [117]	OSR
CIFAR10	93.04	92.98	89.36	93.11	93.18	93.21	93.22	93.32	93.2
序列 CIFAR10	78.31	80.5				81.55	86.31	82.60	64.0

表 X 对比加速方法性能

T	相较于 LIF 神经元的加速比					
	SpikingJelly [49]	PSN [63]	BlockALIF [134] 分组大小 2	4	8	16
2	1.03	2.20	0.20			
4	1.48	4.07	0.17	0.38	0.38	
8	2.72	6.81	0.15	0.29	0.29	
16	6.19	12.60	0.22	0.29	0.29	1.29
32	16.61	17.76	0.25	0.40	0.40	1.01
64	14.83	43.75	0.24	0.45	0.45	0.98

4.2 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、测试 SpikingJelly 框架中融合内核实现的 LIF 神经元[49]、并行脉冲神经元 PSN[63] 和时间分组仿真方式加速的 BlockALIF 神经元[134] 进行实验，对比加速性能. 实验环境为 Intel Core i9-10900X CPU，64G 内存，Nvidia RTX 2080 Ti GPU；神经元数量为 4096；分别测试不同神经元在仿真步数 $T=2,4,8,16,32,64$ 时进行训练（前向传播、反向传播和梯度下降）的耗时. 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??中. 实验结果显示，随着仿真步数的增大，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 则加速效果较差，只在 T 较大且分组大小较大时能略微快于 LIF 神经元，其他情况则速度更慢，原因可能在于其使用 1 维卷积实现并行加速，但 Fang 等[63] 在实现 Sliding PSN 时也尝试过 1 维卷积，指出卷积并行度低、速度慢. 值得注意的是，Luke 等[134] 在带有循环连接的 SNN 上使用 BlockALIF 神经元，加速效果

优秀，可能的原因是循环的连接的计算量较大，而 BlockALIF 神经元在每个时间组内只调用一次循环连接，并在这次调用中并行处理多个时间步的循环信息，因此相比于传统 SNN 的在多个时间步中多次循环计算的方法有着较大速度优势.

5 总结与展望

本文介绍了脉冲深度学习中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. 整体来看，现有算法在很大程度上解决了 SNN 的学习问题，推动 SNN 向着更高性能、更低功耗的方向不断前进，使得以 SNN 为计算模型、神经形态硬件为计算设备并构建超低功耗脉冲智能系统成为现实.

基础学习算法是目前梯度替代法训练 SNN 的基石，但对其研究主要为实验对比，而理论分析较少，需要研究者们重视. **ANN 辅助训练算法**中基于 ANN 耦合的算法梯度误差较大，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；而基于 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，而通过循环神经网络 (Recurrent Neural Network, RNN) 或 Transformer 辅助训练或许能够解决这一问题. **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[76] 中的 Sigmoid 激活函数涉及硬件上昂贵的指数运算，Sliding PSN[63] 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[83] 也需要额外的资源存储和更新状态. 因而，未来的研究中应更多的考虑神经形态硬件兼容性和并行加速算法，以增强模型的实用性. 目前的**网络结构改进方法**已经取得了较大成功，但整体思路仍然延续了 ANN 的惯性，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. **正则化方法**中 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，而其中更倾向于使用 LN，故未来的研究可更多聚焦于 LN 的变体，尤其是原始的 LN 无法与卷积层合并这一问题扔有待 SNN 的研究者解决. **事件驱动学习算法**适合硬件实现，但目前还难以训练、性能较低，存在很大改进空间；值得注意的是，事件驱动算法理论上更适合基于稀疏计算的实现方式，使用脉冲发放时刻直接表示脉冲，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，存在很大的计算冗余，如何为事件驱动算法设计一套稀疏加速仿真方式，也是值得研究者们重视的话题. **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，且适合在神经系统硬件上进行实时学习. 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的处理，需要引起研究者们关注. **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[133] 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申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.

不可否认的是，作为神经科学和计算科学融合的产物的脉冲深度学习领域，目前灵感多来自于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 (TinyML) 等领域也存在一定重合. 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，从大脑的结构功能和运行原理出发，设计脑启发的深度 SNN 学习算法，或许是推动脉冲深度学习取得下一次重大进展的突破方向.

参 考 文 献

[1] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.

[2] HE K, ZHANG X, REN S, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2015: 1026-1034.

[3] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C/OL]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2015: 1-9. DOI: 10.1109/CVPR.2015.7298594.

[4] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2016: 770-778.

[5] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[C]//International Conference on Learning Representations. 2021.

[6] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2014: 580-587.

[7] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2016: 779-788.

[8] GRAVES A, MOHAMED A R, HINTON G. Speech recognition with deep recurrent neural networks[C]//IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013: 6645-6649.

[9] GRAVES A, JAITLY N, MOHAMED A R. Hybrid speech recognition with deep bidirectional lstm[C]// IEEE workshop on Automatic Speech Recognition and Understanding. IEEE, 2013: 273-278.

[10] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[C]//Advances in Neural Information Processing Systems: Vol. 27. 2014.

[11] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate[C]// International Conference on Learning Representations. 2015.

[12] SENNRICH R, HADDOW B, BIRCH A. Neural machine translation of rare words with subword units[C/OL]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, Germany, 2016: 1715-1725. DOI: 10.18653/v1/P16-1162.

[13] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J/OL]. Nature, 2015, 518(7540): 529-533. DOI: 10.1038/nature 14236.

[14] SILVER D, SCHRITTWIESER J, SIMONYAN K, et al. Mastering the game of go without human knowledge[J]. Nature, 2017, 550(7676): 354-359.

[15] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[C]//LAROCHELLE H, RANZATO M, HADSELL R, et al. Advances in Neural Information Processing Systems: Vol. 33. Curran Associates, Inc., 2020: 1877-1901.

[16] ZENG W, REN X, SU T, et al. Pangu-α: Large-scale autoregressive pretrained chinese language models with auto-parallel computation[A]. 2021. arXiv: 2104.12369.

[17] OPENAI, ACHIAM J, ADLER S, et al. Gpt-4 technical report[A]. 2024. arXiv: 2303.08774.

[18] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial nets[C]//GHAHRAMANI Z, WELLING M, CORTES C, et al. Advances in Neural Information Processing Systems: Vol. 27. Curran Associates, Inc., 2014.

[19] RADFORD A, METZ L, CHINTALA S. Unsupervised representation learning with deep convolutional generative adversarial networks[C]//BENGIO Y, LECUN Y. International Conference on Learning Representations. 2016.

[20] HASSABIS D, KUMARAN D, SUMMERFIELD C, et al. Neuroscience-inspired artificial intelligence[J]. Neuron, 2017, 95(2): 245-258.

[21] ZADOR A, ESCOLA S, RICHARDS B, et al. Catalyzing next-generation artificial intelligence through neuroai [J/OL]. Nature Communications, 2023, 14(1): 1597. DOI: 10.1038/s41467-023-37180-x.

[22] ROSENBLATT F. The perceptron: a probabilistic model for information storage and organization in the brain[J]. Psychological Review, 1958, 65(6): 386.

[23] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning representations by back-propagating errors[J]. Nature, 1986, 323(6088): 533-536.

[24] CORTES C, VAPNIK V. Support-vector networks[J]. Machine Learning, 1995, 20(3): 273-297.

[25] MAASS W. Networks of spiking neurons: the third generation of neural network models[J]. Neural Networks, 1997, 10(9): 1659-1671.

[26] GEWALTIG M O, DIESMANN M. Nest (neural simulation tool)[J]. Scholarpedia, 2007, 2(4): 1430.

[27] ELIASMITH C, STEWART T C, CHOO X, et al. A large-scale model of the functioning brain[J/OL]. Science, 2012, 338(6111): 1202-1205. DOI: 10.1126/science.1225266.

[28] STIMBERG M, BRETTE R, GOODMAN D F. Brian 2, an intuitive and efficient neural simulator[J/OL]. eLife, 2019, 8: e47314. DOI: 10.7554/eLife.47314.

[29] MEAD C. Neuromorphic electronic systems[J]. Proceedings of the IEEE, 1990, 78(10): 1629-1636.

[30] ROY K, JAISWAL A, PANDA P. Towards spike-based machine intelligence with neuromorphic computing[J]. Nature, 2019, 575(7784): 607-617.

[31] LICHTSTEINER P, POSCH C, DELBRUCK T. A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor[J]. IEEE Journal of Solid-State Circuits, 2008, 43(2): 566-576.

[32] DONG S, HUANG T, TIAN Y. Spike camera and its coding methods[C]//Data Compression Conference. IEEE Computer Society, 2017: 437-437.

[33] MEROLLA P A, ARTHUR J V, ALVAREZ-ICAZA R, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface[J]. Science, 2014, 345(6197): 668-673.

[34] DAVIES M, SRINIVASA N, LIN T H, et al. Loihi: a neuromorphic manycore processor with on-chip learning [J/OL]. IEEE Micro, 2018, 38(1): 82-99. DOI: 10.1109/MM.2018.112130359.

[35] MA D, SHEN J, GU Z, et al. Darwin: a neuromorphic hardware co-processor based on spiking neural networks [J]. Journal of Systems Architecture, 2017, 77: 43-51.

[36] PEI J, DENG L, SONG S, et al. Towards artificial general intelligence with hybrid tianjic chip architecture [J]. Nature, 2019, 572(7767): 106-111.

[37] HINTON G E, OSINDERO S, TEH Y W. A fast learning algorithm for deep belief nets[J/OL]. Neural Computation, 2006, 18(7): 1527-1554. DOI: 10.1162/neco.2006.18.7.1527.

[38] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[39] GOODFELLOW I, BENGIO Y, COURVILLE A. Deep learning[M]. MIT Press, 2016.

[40] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[C]//PEREIRA F, BURGES C, BOTTOU L, et al. Advances in Neural Information Processing Systems: Vol. 25. Curran Associates, Inc., 2012.

[41] RUSSAKOVSKY O, DENG J, SU H, et al. Imagenet large scale visual recognition challenge[J]. International Journal of Computer Vision, 2015, 115(3): 211-252.

[42] NEFTCI E O, MOSTAFA H, ZENKE F. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks[J]. IEEE Signal Processing Magazine, 2019, 36 (6): 51-63.

[43] CAO Y, CHEN Y, KHOSLA D. Spiking deep convolutional neural networks for energy-efficient object recognition[J]. International Journal of Computer Vision, 2015, 113(1): 54-66.

[44] TAVANAEI A, GHODRATI M, KHERADPISHEH S R, et al. Deep learning in spiking neural networks[J]. Neural Networks, 2019, 111: 47-63.

[45] IOFFE S, SZEGEDY C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]//International Conference on Machine Learning. PMLR, 2015: 448-456.

[46] BA J L, KIROS J R, HINTON G E. Layer normalization [A]. 2016.

[47] IZHIKEVICH E M. Simple model of spiking neurons [J]. IEEE Transactions on Neural Networks, 2003, 14(6): 1569-1572.

[48] FANG W, YU Z, CHEN Y, et al. Incorporating learnable membrane time constant to enhance learning of spiking neural networks[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 2661-2671.

[49] FANG W, CHEN Y, DING J, et al. Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence[J/OL]. Science Advances, 2023, 9(40): eadi1480. DOI: 10.1126/sciadv.adi1480.

[50] LEDINAUSKAS E, RUSECKAS J, JURŠĖNAS A, et al. Training deep spiking neural networks[A]. 2020.

[51] RUECKAUER B, LUNGU I A, HU Y, et al. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification[J]. Frontiers in Neuroscience, 2017, 11: 682.

[52] XIAO H, RASUL K, VOLLGRAF R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms[A]. 2017.

[53] KRIZHEVSKY A, HINTON G. Learning multiple layers of features from tiny images[Z]. 2009.

[54] WU Y, DENG L, LI G, et al. Spatio-temporal backpropagation for training high-performance spiking neural networks[J]. Frontiers in Neuroscience, 2018, 12: 331.

[55] ORCHARD G, JAYAWANT A, COHEN G K, et al. Converting static image datasets to spiking neuromorphic datasets using saccades[J/OL]. Frontiers in Neuroscience, 2015, 9. DOI: 10.3389/fnins.2015.00437.

[56] LI H, LIU H, JI X, et al. Cifar10-dvs: an event-stream dataset for object classification[J/OL]. Frontiers in Neuroscience, 2017, 11. DOI: 10.3389/fnins.2017.00309.

[57] AMIR A, TABA B, BERG D, et al. A low power, fully event-based gesture recognition system[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2017: 7243-7252.

[58] BI Y, CHADHA A, ABBAS A, et al. Graph-based object classification for neuromorphic vision sensing[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

[59] LIN Y, DING W, QIANG S, et al. Es-imagenet: a million event-stream classification dataset for spiking neural networks[J/OL]. Frontiers in Neuroscience, 2021, 15. DOI: 10.3389/fnins.2021.726582.

[60] CRAMER B, STRADMANN Y, SCHEMMEL J, et al. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks[J/OL]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(7): 2744-2757. DOI: 10.1109/TNNLS.2020.3044364.

[61] IYER L R, CHUA Y, LI H. Is neuromorphic mnist neuromorphic? analyzing the discriminative power of neuromorphic datasets in the time domain[J/OL]. Frontiers in Neuroscience, 2021, 15. DOI: 10.3389/fnins.2021.608567.

[62] YIN B, CORRADI F, BOHTÉ S M. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks[J]. Nature Machine Intelligence, 2021, 3(10): 905-913.

[63] FANG W, YU Z, ZHOU Z, et al. Parallel spiking neurons with high efficiency and ability to learn long-term dependencies[C]//Advances in Neural Information Processing Systems. 2023.

[64] YAO M, RICHTER O, ZHAO G, et al. Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip[J/OL]. Nature Communications, 2024, 15(1): 4464. DOI: 10.1038/s41467-024-47811-6.

[65] YANG Z, WANG T, LIN Y, et al. A vision chip with complementary pathways for open-world sensing[J/OL]. Nature, 2024, 629(8014): 1027-1033. DOI: 10.1038/s41586-024-07358-4.

[66] ZENKE F, GANGULI S. Superspike: Supervised learning in multilayer spiking neural networks[J/OL]. Neural Computation, 2018, 30(6): 1514-1541. DOI: 10.1162/neco_a_01086.

[67] SHRESTHA S B, ORCHARD G. Slayer: Spike layer error reassignment in time[C]//Advances in Neural Information Processing Systems. 2018: 1419-1428.

[68] ZENKE F, VOGELS T P. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks[J]. BioRxiv, 2020.

[69] LI Y, GUO Y, ZHANG S, et al. Differentiable spike: Rethinking gradient-descent for training spiking neural networks[C]//Advances in Neural Information Processing Systems. 2021.

[70] WU J, CHUA Y, ZHANG M, et al. A tandem learning rule for effective training and rapid inference of deep spiking neural networks[J/OL]. IEEE transactions on neural networks and learning systems, 2023, 34(1): 446—460. DOI: 10.1109/tnnls.2021.3095724.

[71] KHERADPISHEH S R, MIRSADEGHI M, MASQUELIER T. Spiking neural networks trained via proxy[J]. IEEE Access, 2022.

[72] XU Q, LI Y, SHEN J, et al. Constructing deep spiking neural networks from artificial neural networks with knowledge distillation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 7886-7895.

[73] QIU H, NING M, SONG Z, et al. Self-architectural knowledge distillation for spiking neural networks[J/OL]. Neural Networks, 2024, 178: 106475. DOI: https://doi.org/10.1016/j.neunet.2024.106475.

[74] YAO X, LI F, MO Z, et al. GLIF: a unified gated leaky integrate-and-fire neuron for spiking neural networks[C]// Advances in Neural Information Processing Systems. 2022.

[75] FENG L, LIU Q, TANG H, et al. Multi-level firing with spiking ds-resnet: Enabling better and deeper directly-trained spiking neural networks[C/OL]//International Joint Conference on Artificial Intelligence. 2022: 2471-2477. DOI: 10.24963/ijcai.2022/343.

[76] HUANG Y, LIN X, REN H, et al. CLIF: Complementary leaky integrate-and-fire neuron for spiking neural networks[C]//Forty-first International Conference on Machine Learning. 2024.

[77] YARGA S Y A, WOOD S U N. Accelerating snn training with stochastic parallelizable spiking neurons[C/OL]// International Joint Conference on Neural Networks. 2023: 1-8. DOI: 10.1109/IJCNN54540.2023.10191884.

[78] RUECKAUER B, LIU S C. Conversion of analog to spiking neural networks using sparse temporal coding [C/OL]//IEEE International Symposium on Circuits and Systems. 2018: 1-5. DOI: 10.1109/ISCAS.2018.8351295.

[79] BOHTE S M, KOK J N, LA POUTRE H. Error-backpropagation in temporally encoded networks of spiking neurons[J]. Neurocomputing, 2002, 48(1-4): 17-37.

[80] MOSTAFA H, PEDRONI B U, SHEIK S, et al. Fast classification using sparsely active spiking networks [C/OL]//IEEE International Symposium on Circuits and Systems. 2017: 1-4. DOI: 10.1109/ISCAS.2017.8050527.

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

[82] KHERADPISHEH S R, MASQUELIER T. Temporal backpropagation for spiking neural networks with one spike per neuron[J]. International Journal of Neural Systems, 2020, 30(06): 2050027.

[83] FANG H, SHRESTHA A, ZHAO Z, et al. Exploiting neuron and synapse filter dynamics in spatial temporal learning of deep spiking neural network[C/OL]// International Joint Conference on Artificial Intelligence. 2020: 2799-2806. DOI: 10.24963/ijcai.2020/388.

[84] HAMMOUAMRI I, KHALFAOUI-HASSANI I, MASQUELIER T. Learning delays in spiking neural

networks using dilated convolutions with learnable spacings[C]//The Twelfth International Conference on Learning Representations. 2024.

[85] HU Y, TANG H, PAN G. Spiking deep residual networks [J/OL]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(8): 5200-5205. DOI: 10.1109/TNNLS.2021.3119238.

[86] ZHENG H, WU Y, DENG L, et al. Going deeper with directly-trained larger spiking neural networks[C]// Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 35. 2021: 11062-11070.

[87] FANG W, YU Z, CHEN Y, et al. Deep residual learning in spiking neural networks[C]//Advances in Neural Information Processing Systems: Vol. 34. 2021.

[88] HU Y, DENG L, WU Y, et al. Advancing spiking neural networks toward deep residual learning[J/OL]. IEEE Transactions on Neural Networks and Learning Systems, 2024: 1-15. DOI: 10.1109/TNNLS.2024.3355393.

[89] GRAVES A. Generating sequences with recurrent neural networks[A]. 2014. arXiv: 1308.0850.

[90] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems: Vol. 30. 2017.

[91] YAO M, GAO H, ZHAO G, et al. Temporal-wise attention spiking neural networks for event streams classification [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 10221-10230.

[92] YAO M, HU J, ZHAO G, et al. Inherent redundancy in spiking neural networks[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 16924-16934.

[93] YAO M, ZHANG H, ZHAO G, et al. Sparser spiking activity can be better: Feature refine-and-mask spiking neural network for event-based visual recognition[J]. Neural Networks, 2023, 166: 410-423.

[94] XU Q, GAO Y, SHEN J, et al. Enhancing adaptive history reserving by spiking convolutional block attention module in recurrent neural networks[C]//OH A, NAUMANN T, GLOBERSON A, et al. Advances in Neural Information Processing Systems: Vol. 36. Curran Associates, Inc., 2023: 58890-58901.

[95] YAO M, ZHAO G, ZHANG H, et al. Attention spiking neural networks[J/OL]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(8): 9393-9410. DOI: 10.1109/TPAMI.2023.3241201.

[96] ZHU R J, ZHANG M, ZHAO Q, et al. Tcja-snn: Temporal-channel joint attention for spiking neural networks[J/OL]. IEEE Transactions on Neural Networks and Learning Systems, 2024: 1-14. DOI: 10.1109/TNNLS.2024.3377717.

[97] HUANG Z, ZHANG S, PAN L, et al. Tada! temporally-adaptive convolutions for video understanding[C]// International Conference on Learning Representations. 2022.

[98] WANG X, GIRSHICK R, GUPTA A, et al. Non-local neural networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 7794-7803.

[99] KIM S, KIM S, HONG S, et al. 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[C]//IEEE International Solid-State Circuits Conference. IEEE, 2023: 334-336.

[100] CHANG M, LELE A S, SPETALNICK S D, et al. A heterogeneous rram in-memory and sram near-memory soc for fused frame and event-based target identification and tracking[C]//IEEE International Solid-State Circuits Conference. IEEE, 2023: 426-428.

[101] ZHANG J, DONG B, ZHANG H, et al. Spiking transformers for event-based single object tracking[C]// Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. 2022: 8801-8810.

[102] ZHANG J, TANG L, YU Z, et al. Spike transformer: Monocular depth estimation for spiking camera[C]// European Conference on Computer Vision. Springer, 2022: 34-52.

[103] HAN M, WANG Q, ZHANG T, et al. Complex dynamic neurons improved spiking transformer network

for efficient automatic speech recognition[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2023.

[104] ZHOU Z, ZHU Y, HE C, et al. Spikformer: When spiking neural network meets transformer[C]// International Conference on Learning Representations. 2023.

[105] YAO M, HU J, ZHOU Z, et al. Spike-driven transformer [C]//Advances in Neural Information Processing Systems: Vol. 36. 2023: 64043-64058.

[106] SHI X, HAO Z, YU Z. Spikingresformer: Bridging resnet and vision transformer in spiking neural networks[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 5610-5619.

[107] ZHOU C, ZHANG H, ZHOU Z, et al. Qkformer: Hierarchical spiking transformer using q-k attention[A]. 2024. arXiv: 2403.16552.

[108] HASSANI A, WALTON S, SHAH N, et al. Escaping the big data paradigm with compact transformers[A]. 2022. arXiv: 2104.05704.

[109] YAO M, HU J, HU T, et al. Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips[C]// The Twelfth International Conference on Learning Representations. 2024.

[110] LIU Z, LIN Y, CAO Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 10012-10022.

[111] HOROWITZ M. 1.1 computing's energy problem (and what we can do about it)[C/OL]//IEEE International Solid-State Circuits Conference Digest of Technical Papers. 2014: 10-14. DOI: 10.1109/ISSCC.2014.6757323.

[112] NA B, MOK J, PARK S, et al. AutoSNN: Towards energy-efficient spiking neural networks[C]//Proceedings of Machine Learning Research: Vol. 162 Proceedings of the 39th International Conference on Machine Learning. 2022: 16253-16269.

[113] KIM Y, LI Y, PARK H, et al. Neural architecture search for spiking neural networks[C]//European Conference on Computer Vision. Cham, 2022: 36-56.

[114] CHE K, LENG L, ZHANG K, et al. Differentiable hierarchical and surrogate gradient search for spiking neural networks[J]. Advances in Neural Information Processing Systems, 2022, 35: 24975-24990.

[115] WU Y, DENG L, LI G, et al. Direct training for spiking neural networks: Faster, larger, better[C/OL]// Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 33. 2019: 1311-1318. DOI: 10.1609/aaai.v33i01.33011311.

[116] KIM Y, PANDA P. Revisiting batch normalization for training low-latency deep spiking neural networks from scratch[J/OL]. Frontiers in Neuroscience, 2021, 15. DOI: 10.3389/fnins.2021.773954.

[117] DUAN C, DING J, CHEN S, et al. Temporal effective batch normalization in spiking neural networks [C]//Advances in Neural Information Processing Systems. 2022.

[118] GUO Y, ZHANG Y, CHEN Y, et al. Membrane potential batch normalization for spiking neural networks[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 19420-19430.

[119] GUO Y, LIU X, CHEN Y, et al. Rmp-loss: Regularizing membrane potential distribution for spiking neural networks[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 17391-17401.

[120] DENG S, LI Y, ZHANG S, et al. Temporal efficient training of spiking neural network via gradient re-weighting[C]//International Conference on Learning Representations. 2022.

[121] LI Y, KIM Y, PARK H, et al. Neuromorphic data augmentation for training spiking neural networks[C]// European Conference on Computer Vision. Cham, 2022: 631-649.

[122] ZHANG W, LI P. Temporal spike sequence learning via backpropagation for deep spiking neural networks [C]//Advances in Neural Information Processing Systems. 2020: 12022-12033.

[123] ZHU Y, YU Z, FANG W, et al. Training spiking neural networks with event-driven backpropagation[C]//

Advances in Neural Information Processing Systems. 2022.

[124] ZHU Y, FANG W, XIE X, et al. Exploring loss functions for time-based training strategy in spiking neural networks [J]. Advances in Neural Information Processing Systems, 2024, 36.

[125] KAISER J, MOSTAFA H, NEFTCI E. Synaptic plasticity dynamics for deep continuous local learning (decolle)[J]. Frontiers in Neuroscience, 2020, 14: 424.

[126] XIAO M, MENG Q, ZHANG Z, et al. Online training through time for spiking neural networks[C]//Advances in Neural Information Processing Systems. 2022.

[127] MENG Q, XIAO M, YAN S, et al. Training high-performance low-latency spiking neural networks by differentiation on spike representation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 12444-12453.

[128] MENG Q, XIAO M, YAN S, et al. Towards memory- and time-efficient backpropagation for training spiking neural networks[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 6166-6176.

[129] JIANG H, DE MASI G, XIONG H, et al. Ndot: Neuronal dynamics-based online training for spiking neural networks[C]//International Conference on Machine Learning. 2024.

[130] ZHU Y, DING J, HUANG T, et al. Online stabilization of spiking neural networks[C]//International Conference on Learning Representations. 2024.

[131] HU J, YAO M, QIU X, et al. High-performance temporal reversible spiking neural networks with $\mathcal{O}(l)$ training memory and $\mathcal{O}(1)$ inference cost [C]//International Conference on Machine Learning. 2024.

[132] LIU Z, MAO H, WU C Y, et al. A convnet for the 2020s[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 11976-11986.

[133] PEREZ-NIEVES N, GOODMAN D F M. Sparse spiking gradient descent[C]//Advances in Neural Information Processing Systems. 2021.

[134] TAYLOR L, KING A, HARPER N S. Addressing the speed-accuracy simulation trade-off for adaptive spiking neurons[C]//OH A, NAUMANN T, GLOBERSON A, et al. Advances in Neural Information Processing Systems: Vol. 36. Curran Associates, Inc., 2023: 59360-59374.

Wei Fang born in 1996, received his B.S. degree from Department of Automation, Tsinghua University, China in 2019 and Ph.D. degree from School of Computer Science, Peking University in 2024. He is currently the Research Assistant Professor in School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University. His reearch interests include the learning and network structure of Spiking Neural Networks. He has published many articles in journals such as Science Advances/Nature Communications, Neural Networks and conferences such as NeurIPS/ICML/ICLR/ICCV/IJCAI.

Yonghong Tian born in 1975, Ph.D., is currently dean of school of electronic and computer engineering, a Boya Distinguished Professor with the Department of Computer Science and Technology, Peking University, China, and is also the deputy director of Artificial Intelligence Research Center, PengCheng Laboratory, Shenzhen, China. His research interests include neuromorphic vision, brain-inspired computation and multimedia big data. He has co-authored over 200 technical articles in refereed journals such as Science Advances/Nature Communications/Scientific Data, IEEE TPAMI/TNNLS/TIP/TMM/TCSVT/TKDE/TPDS/TCYB, ACM CSUR/TOIS/TOMM and conferences such as NeurIPS/ICML/ICLR/CVPR/ICCV/ECCV/AAAI/IJCAI.

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 consequently 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, and No.62088102.