尖峰变压器：基于变压器的尖峰神经网络的尖峰驱动残差学习

Chenlin Zhou[1] , Liutao Yu[1] , Zhaokun Zhou[1,2] , Zhengyu Ma ,[1]* Han Zhang[1,3] , Huihui Zhou ,[1]* Yonghong Tian[1,2]

[1]鹏程实验室，中国深圳 518055
[2]北京大学计算机科学与技术系
[3]哈尔滨工业大学计算机科学与技术系

摘要

尖峰神经网络（SNN）由于其事件驱动的尖峰计算，为人工神经网络提供了一种前景广阔的节能替代方案。然而，最先进的深度 SNN（包括 Spikformer 和 SEW ResNet）因其残差连接结构而存在非尖峰计算（整数-浮点乘法）问题。这些非尖峰计算增加了 SNN 的功耗，使其不适合部署在只支持尖峰操作的主流神经元硬件上。在本文中，我们为 SNNs 提出了一种硬件友好的尖峰驱动残差学习架构，以避免非尖峰计算。基于这种残差设计，我们开发了基于纯变压器的尖峰神经网络 Spikingformer。我们在 ImageNet、CIFAR10、CIFAR100、CIFAR10-DVS 和 DVS128 手势数据集上对 Spikingformer 进行了评估，结果表明 Spikingformer 作为一种新型高级骨干网络，在直接训练纯 SNN 方面优于最先进的技术（在 Ima- geNet 上的 top-1 准确率为 75.85%，与 Spikformer 相比提高了 1.04%）。此外，我们的实验还验证了 Spikingformer 能有效避免非尖峰计算，与 Spikformer 相比，在 ImageNet 上的能耗显著降低了 57.34%。据我们所知，这是首次开发出基于纯事件驱动变压器的 SNN。代码将在 Spikingformer 上提供。

1 导言

受大脑启发的尖峰神经网络（SNN）被认为是第三代神经网络[1]，由于其高度的生物学拟真性、事件驱动特性以及在神经形态硬件上的低功耗[2]，它是人工神经网络（ANN）的潜在竞争对手。In particular, the utilization of binary spike signals allows SNNs to adopt low-power accumulation (AC) instead of the traditional high-power multiply-accumulation (MAC), leading to significant energy efficiency gains and making SNNs increasingly popular [3].

随着 SNN 越来越深入，其性能也得到了显著提高 [4、5、6、7、8、9、10]。为了扩展 SNN 的深度，人们对具有跳转连接的 ResNet 进行了广泛研究 [5，8]。最近，SEW ResNet [5] 作

为基于卷积的 SNN 的代表，轻松实现了身份映射，并克服了 Spiking ResNet [11] 的梯度消失/爆炸问题。SEW ResNet 是第一个直接训练出超过 100 层的深度 SNN。Spikformer [8]是一种直接训练的具有残差连接的基于变压器的代表性 SNN，它是利用 SNN 的自注意能力和生物特性而提出的。这是首次将蓬勃发展的变压器架构应用于 SNN 设计的成功探索，并显示出强大的性能。

---

*通讯作者 Preprint.正在

审稿。

然而，Spikformer 和 SEW ResNet 都面临着 ADD 残余连接引起的非尖峰计算（整数-浮点乘法）的挑战。这不仅限制了它们充分发挥事件驱动处理在能效方面的优势，也使得它们难以在神经形态硬件上部署和优化性能[12, 3]。

开发一种纯粹的 SNN 来解决 Spikformer 和 SEW ResNet 中的非尖峰计算挑战，同时保持高性能极为重要。本文受二元神经网络（BNN）[13, 14, 15, 16]架构设计的启发，为 SNN 提出了尖峰驱动残差学习（Spike-driven Residual Learning）架构，以避免非尖峰计算。基于这种残差设计，我们开发了一种基于纯变压器的尖峰神经网络，命名为 Spikingformer。我们在静态数据集 ImageNet[17]、CIFAR[18]（包括 CI- FAR10 和 CIFAR100）和神经形态数据集（包括 CIFAR10-DVS 和 DVS128 手势）上评估了 Spikingformer 的性能。实验结果表明，Spikingformer 能有效避免 Spikformer 中的整数-浮点乘法。此外，作为一种新型的高级 SNN 骨干，Spikingformer 在上述所有数据集上的表现都远远优于 Spikformer（例如，在 ImageNet 上 + 1.04%，在 CIFAR100 上 + 1.00%）。

2 相关工作

2.1 基于卷积的尖峰神经网络

获得基于深度卷积的 SNN 模型有两种主流方法：ANN 到 SNN 的转换和通过代梯度直接训练。

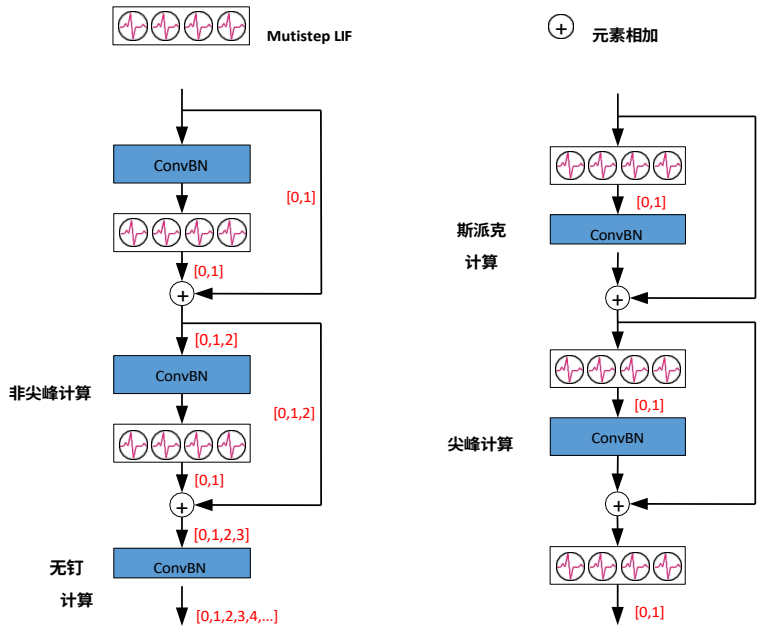
ANN 到 SNN 的转换。在 ANN 到 SNN 的转换中 [19, 20, 21, 22, 23, 24]，通过将 ReLU 激活层替换为尖峰神经元，并添加权值归一化和阈值平衡等缩放操作，将预先训练好的 ANN 转换为 SNN。这种转换过程需要较长的转换时间步骤，并受到原始 ANN 设计的限制。

通过代梯度直接训练。在直接训练领域，SNN 在模拟时间步长内展开，并通过时间反向传播进行训练 [25，26]。由于尖峰神经元的不可分性，反向传播采用了代梯度法 [27, 28]。SEW ResNet[5]是通过直接训练建立的基于卷积的 SNN 模型的代表，也是第一个将 SNN 的层数增加到大于 100 层的模型。然而，SEW ResNet 剩余连接中的 ADD 门会在深度卷积层中产生非尖峰的整数-浮点乘法计算。文献[3]指出了 SEW ResNet 和 Spikformer 中的非尖峰计算问题，并试图通过在训练过程中添加辅助累加通路并在推理过程中移除该通路来解决这一问题。这种策略需要进行繁琐的额外操作，与原始模型相比，性能明显下降。

2.2 基于变压器的尖峰神经网络

现有的 SNN 大多借鉴卷积神经网络（CNN）的架构，因此其性能受到 CNN 性能的限制。变换器架构最初是为自然语言处理而设计的[29]，它在许多计算机视觉任务中取得了巨大成功，包括图像分类[30, 31]、物体检测[32, 33, 34]和语义分割[35, 36]。变换器的结构为一种新

型 SNN 带来了希望，具有突破 SNN 性能瓶颈的巨大潜力。迄今为止，相关的研究主要有两个：Spikformer[8] 和 Spikeformer[37] 提出了基于变压器结构的尖峰神经网络。虽然 Spikeformer 将前馈层使用的激活函数替换为尖峰激活函数，但仍存在大量非尖峰操作，包括浮点乘法、除法、指数操作等。Spikformer 提出了一种新颖的尖峰自注意（SSA）模块，使用尖峰形式的查询、键和值，不使用 softmax，并在许多数据集上实现了最先进的性能。然而，Spikformer 的残差连接结构仍包含非尖峰计算。在我们的研究中，我们采用了 Spikformer 中的 SSA 模块，并将残差结构修改为纯事件驱动，在提高性能的同时对硬件友好且节能。



(a) Spikformer，SEW ResNet　　(b) Spikingformer（我们的）

图 1：Spikformer、SEW ResNet 和 Spikingformer（我们的）中的残差学习。(a) 显示了 Spikformer 和 SEW ResNet 的残差学习，它们在 ConvBN 层包含非尖峰计算（整数-浮点数乘法）。(b) 显示了我们在 Spiking- former 中提出的尖峰驱动残差学习，它遵循尖峰驱动原则，可有效避免浮点乘法和整数-浮点乘法。请注意，Mutistep LIF 是时间步长 $T > 1$ 的尖峰神经元泄漏积分与火焰（LIF）模型[5, 8]。在其他层中，它与批次大小合并。在本研究中，我们使用 ConvBN 表示卷积层及其后续的 BN 层。

3　方法

3.1　Spikformer 和 SEW ResNet 的缺点

目前，Spikformer [8] 是将深度 SNN 与变换器架构相结合的代表作，而 SEW ResNet [5] 则是基于卷积的深度 SNN 的代表作。残差学习在 Spikformer 和 SEW ResNet 中都扮演着极其重要的角色，但 Spikformer 和 SEW ResNet 中的 ADD 残差连接会导致非尖峰计算（整数-浮点数乘法），而这不是事件驱动计算。如图 1（a）所示，Spikformer 和 SEW ResNet 的残差学习可表述如下：

$$O_l = \mathrm{SN}_l\left(\mathrm{ConvBN}_l\left(O_{l-1}\right)\right) + O_{l-1} = S_l + O_{l-1} \tag{1}$$

$$O_{l+1} = \mathrm{SN}_{l+1}\left(\mathrm{ConvBN}_{l+1}\left(O_l\right)\right) + O_l = S_{l+1} + O_l \tag{2}$$

其中 S_l 表示学习到的残差映射，即 $S_l = \mathrm{SN}(\mathrm{ConvBN}(O_{l-1}))$。这种残差设计不可避免地会带来非尖峰数据，从而在下一层/块中进行 MAC 运算。其中，S_l 和 O_{l-1} 为尖峰信号，其输出 O_l 为非尖峰信号，范围为 $\{0, 1, 2\}$。在计算 O_{l+1} 的 S_{l+1} 时，非尖峰数据会破坏下一个卷积

层的事件驱动计算。随着网络深度的增加，传输到网络深层的非尖峰数据值范围也会扩大。在我们的 Spikformer 实现中，在 ImageNet 2012 上测试 Spikormer-8-512 时，非尖峰数据的范围可能会增加到 {0, 1, 2, ..., 16}。显然，非尖峰数据的范围与 Spikformer 和 SEW ResNet 中残差块的数量大致成正比。

事实上，整数-浮点乘法在硬件中的实现方式通常与浮点乘法相同。在这种情况下，网络将产生高能耗，接近具有相同结构的 ANN 的能耗，这对于 SNN 来说是不可接受的。

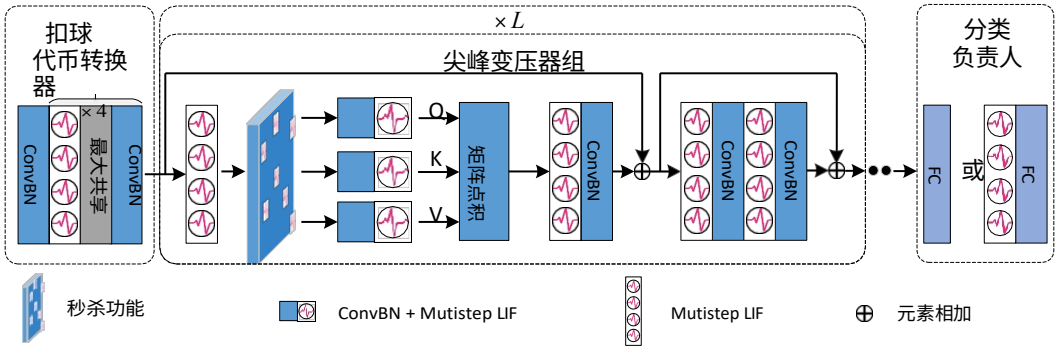


图 2：尖峰变压器概述，，它由一个尖峰标记器、几个尖峰变压器模块和一个分类头组成。

3.2 Spikingformer 中的尖峰驱动残差学习

图 1(b) 展示了我们在 Spikingformer 中提出的尖峰驱动残差学习。根据尖峰驱动原则，它可以有效避免浮点乘法和整数-浮点乘法。尖峰驱动的残差学习可以简单地表述如下：

$$O_l = \mathrm{ConvBN}_l\left(\mathrm{SN}_l\left(O_{l-1}\right)\right) + O_{l-1} = S_l + O_{l-1} \tag{3}$$

$$O_{l+1} = \mathrm{ConvBN}_{l+1}\left(\mathrm{SN}_{l+1}\left(O_l\right)\right) + O_l = S_{l+1} + O_l \tag{4}$$

我们提出了用于残差学习的 SN - ConvBN，以替代 Spikformer 和 SEW ResNet 中的 ConvBN - SN。在我们的结构中，$S_l + O_{l-1}$ 属于浮点加法运算，与 SN 层的加法运算相同。浮点加法运算是 SNN 最基本的运算。显然，输出 O_l 也是浮点运算，在参与下一次 ConvBN 计算之前会经过 SN 层。因此，经过 SN 层处理后将生成纯尖峰形式特征，ConvBN 层的计算将变成纯浮点加法运算，遵循尖峰驱动原则，大大降低能耗。

3.3 Spikingformer 建筑

我们提出了 Spikingformer，它是一种通过整合尖峰驱动残差块而形成的基于纯变压器的新型尖峰神经网络。本节将讨论 Spikingformer 的细节。Spikingformer 的流水线如图 2 所示。

我们提出的 Spikingformer 包含一个尖峰标记器 (ST)、多个尖峰变换器块和一个分类头。给定一个二维图像序列 $I \in \mathrm{R}^{T\times C\times H\times W}$（注意，在 ImageNet 2012 等静态数据集中，C=3；在 DVS-Gesture 等神经形态数据集中，C=2），我们使用 Spiking Tokenizer 块进行下采样和补丁嵌入，输入可投射为尖峰形式的补丁 $X \in \mathrm{R}^{T\times N\times D}$。显然，当以静态图像作为输入时，第一层 Spiking Tokenizer 也起着尖峰编码器的作用。在尖峰标记器之后，尖峰形式的补丁 X_0 将进入尖峰变换器块 L。与标准 ViT 编码器块类似，尖峰转换器块包含一个尖峰自注意（SSA）[8] 和一个尖峰 MLP 块。最后，一个全连接层（FC）用于分类头。请注意，我们在全连接层之前使用了全局平均池化（GAP），以减少全连接层的参数，提高尖峰转换器的分类能力。

$$X = \mathrm{ST}(I)、 \quad i \in R^{T \times C \times H \times W}, x \in R^{T \times N \times D} \tag{5}$$

$$X'_l = \mathrm{SSA}(X_{l-1}) + X_{l-1}, X'_l \in R^{T \times N \times D}, l = 1..L \tag{6}$$

$$X_l = \mathrm{SMLP}(X'_l) + X'_l, X_l \in R^{T \times N \times D}, l = 1..L \tag{7}$$

$$y = \mathrm{fc}(\mathrm{gap}(x))_L \qquad 或 \quad y' = \mathrm{fc}(\mathrm{gap}(\mathrm{sn}(x)))_L \tag{8}$$

尖峰标记器如图 2 所示，Spiking Tokenizer 主要包含两个功能：1) 卷积尖峰补丁嵌入，以及 2) 下采样，将特征图投射到一个"..... "中。

较小的固定尺寸。尖峰补丁嵌入类似于 Vision Transformer [38, 39] 中的卷积流，在每个卷积层中，尖峰形式特征通道的维度逐渐增加，最终与补丁的嵌入维度相匹配。此外，当使用静态图像作为输入时，Spiking Tokenizer 的第一层被用作尖峰编码器。如公式 9 和公式 10 所示，ConvBN 的卷积部分代表二维卷积层（步长-1，核大小为 3×3）。MP 和 SN 分别代表 maxpooling（步长-2）和 mutistep 尖峰神经元。公式 9 用于不带下采样的尖峰片段嵌入（SPE），公式 10 用于带下采样的尖峰片段嵌入（SPED）。我们可以针对具有不同降采样要求的特定分类任务使用多个 SPE 或 SPED。例如，在输入大小为 224*224 的 ImageNet 2012 数据集分类中，我们使用了 4 个 SPED（使用 16 倍下采样）；在输入大小为 32*32 的 CIFAR 数据集分类中，我们使用了 2 个 SPE 和 2 个 SPED（使用 4 倍下采样）。经过 Spiking Tokenizer 模块处理后，输入 I 被分割成一个图像补丁序列 $X \in \mathbb{R}^{T \times N \times D}$。

$$I_i = \mathrm{ConvBN}(\mathrm{SN}(I)) \tag{9}$$

$$I_i = \mathrm{ConvBN}(\mathrm{MP}(\mathrm{SN}(I))) \tag{10}$$

尖峰变压器区块。我们的尖峰自注意部分与 Spik- former [8] 中的尖峰自注意类似，后者是一种纯尖峰形式的自注意。不过，我们做了一些修改：1) 根据我们提出的尖峰驱动残差机制，我们改变了尖峰神经元层的位置，避免了整数和浮点权重的相乘。2) 我们在 Spikformer 中选择 ConvBN 代替 *LinearBN*（线性层和批量归一化）。因此，Spikingformer 中的 SSA 可以表述如下：

$$X' = \mathrm{SN}(X)、 \tag{11}$$

$$Q = \mathrm{SN}_Q(\mathrm{ConvBN}_Q(X')),\ K = \mathrm{SN}(\mathrm{ConvBN}_{KK}(X')),\ V = \mathrm{SN}_V(\mathrm{ConvBN}_V(X))' \tag{12}$$

$$\mathrm{SSA}(Q, K, V) = \mathrm{ConvBN}(\mathrm{SN}(QK^{\mathrm{T}} \vee * s)) \tag{13}$$

其中 Q、K、$V \in \mathbb{R}^{T \times N \times D}$ 为纯尖峰数据（仅包含 0 和 1）。s 为缩放因子，如文献[8]所述，用于控制矩阵乘法结果的大值。尖峰 MLP 模块由两个 SPE 组成，其公式如公式 9 所示。尖峰变压器块如图 2 所示，是尖峰变压器的主要组成部分。

分类头。我们使用全连接层作为最后一个尖峰变压器模块后面的分类器。具体来说，分类器可以以四种形式实现：AvgPooling - FC、SN - AvgPooling - FC、FC - AvgPooling、SN - FC - AvgPooling，其形式如下：

$$Y = \mathrm{FC}(\mathrm{AvgPooling}(X))_L \tag{14}$$

$$Y = \mathrm{FC}(\mathrm{AvgPooling}(\mathrm{SN}(X)))_L \tag{15}$$

$$Y = \mathrm{AvgPooling}(\mathrm{FC}(X))_L \tag{16}$$

$$Y = \mathrm{AvgPooling}(\mathrm{FC}(\mathrm{SN}(X)))_L \tag{17}$$

FC 后的 AvgPooling（如 SN - FC - AvgPooling、FC - AvgPooling）可以看作是对神经元发射的平均值进行计算，是对网络的后处理，但这种方法通常需要大量参数。与前几种方法相比，FC 之前的 AvgPooling（如 AvgPooling - FC、SN - AvgPooling - FC）可以有效减少参数。只有 SN - FC - AvgPooling 可以避免浮点乘法运算，但与 AvgPooling - FC 或 SN - AvgPooling - FC 相比，它需要更多的 FC 参数。此外，它还降低了网络的分类能力。本文主要采用 AvgPooling 先于 FC 的方式，默认选择 AvgPooling - FC 作为 Spikingformer 的分类器。关于分类头的一些实验分析将在第 5.1 节中讨论。

3.4 理论突触操作和能量消耗计算

卷积的同质性允许在部署时将下面的 BN 和线性缩放变换等效地融合到卷积层中，并增加偏置[40, 41, 11, 3]。因此，在计算理论能耗时，可以忽略 BN 层的能耗。在计算 Spikingformer 的理论能耗之前，我们先计算尖峰的突触操作次数。

$$SOP^l = fr \times T \times FLOPs^l \tag{18}$$

表 1：ImageNet-1k 分类结果。功耗按 ImageNet 上图像推理的平均理论能耗计算，其细节见公式 19。与 Spikformer 相同，我们的 Spikingformer-L-D 表示具有 L 个尖峰变压器块和 D 个特征嵌入维度的 Spikingformer 模型。OPs 指 SNN 中的 SOPs 和 ANN 中的 FLOPs。请注意，推理的默认输入分辨率为 224×224。

方法	建筑	帕拉姆（男）	业务方案 (G)	时间步骤	能耗（兆焦耳）	Top-1 Acc
TET[51]	Spiking-ResNet-34	21.79	-	6	-	64.79
	SEW ResNet-34	21.79	-	4	-	68.00
尖峰 ResNet[4]	ResNet-34	21.79	65.28	350	59.30	71.61
	ResNet-50	25.56	78.29	350	70.93	72.75
STBP-tdBN[6]	Spiking-ResNet-34	21.79	6.50	6	6.39	63.72
SEW ResNet[5]	SEW ResNet-34	21.79	3.88	4	4.04	67.04
	SEW ResNet-50	25.56	4.83	4	4.89	67.78
	SEW ResNet-101	44.55	9.30	4	8.91	68.76
	SEW ResNet-152	60.19	13.72	4	12.89	69.26
MS-ResNet[7]	ResNet-104	44.55+	-	5	-	74.21
ANN[8]	变压器-8-512	29.68	8.33	1	38.34	80.80
Spikformer[8]	Spikformer-8-384	16.81	6.82	4	12.43	70.24
	Spikformer-8-512	29.68	11.09	4	18.82	73.38
	Spikformer-8-768	66.34	22.09	4	32.07	74.81
斯派金格玛	Spikingformer-8-384	16.81	3.88	4	**4.69(-62.**27%)	**72.45(+2.21)**
	Spikingformer-8-512	29.68	6.52	4	**7.46(-60.**36%)	**74.79(+1.41)**
	Spikingformer-8-768	66.34	12.54	4	**13.68(-57.**34%)	**75.85(+1.04)**

其中，l 是 Spikingformer 中的区块/层，fr 是区块/层的发射率，T 是尖峰神经元的模拟时间步长。$FLOPs^l$ 指的是块/层 l 的浮点运算，即乘法累加（MAC）运算的次数。SOP^l 是基于尖峰的累加（AC）操作数。我们根据文献[42, 7, 12, 43, 44, 45, 46]估算 Spikingformer 的理论能耗。我们假设 MAC 和 AC 操作是在 45 纳米硬件 [12] 上实现的，其中 $E_{MAC} = 4.6pJ$ 和 $E_{AC} = 0.9pJ$。Spikingformer 的理论能耗可计算如下：

$$E^{\text{灰度}}_{\text{Spikingformer}} = E_{AC} \times \left(\sum_{i=2}^{N} \text{SOPConv}_i + \sum_{j=1}^{M} \text{SOPSSA}_j \right) + E_{MAC} \times \text{FLOPConv}_1 \tag{19}$$

$$E^{\text{神经}}_{\text{Spikingformer}} = E_{AC} \times \left(\sum_{i=1}^{N} \text{SOPConv}_i + \sum_{j=1}^{M} \text{SOPSSA}_j \right) \tag{20}$$

公式 19 显示了 Spikingformer 在输入 RGB 图像的静态数据集上的能耗。
$FLOP_{Conv}^1$ 是将静态 RGB 图像编码成尖峰形式的第一层。然后，N SNN 的 SOP Conv 层和 M SSA 层相加后乘以 E_{AC}。公式 20 显示了 Spikingformer 处理神经形态数据集的能耗。

4 实验

在本节中，我们将在静态数据集 ImageNet [17]、静态数据集 CIFAR [18]（包括 CIFAR10 和 CIFAR100）和神经形态数据集（包括 CIFAR10-DVS 和 DVS128 手势 [47]）上进行实验，以评估 Spikingformer 的性能。用于实验的模型是基于 Pytorch [48]、SpikingJelly [49] 和 Timm [50] 实现的。

4.1 图像网络分类

ImageNet 包含约 130 万张用于训练的 1000 级图像和 5 *万张*用于验证的图像。我们的 ImageNet 模型的输入大小默认设置为 224 × 224。优化器

表 2：CIFAR10/100 分类结果。与 Spikformer 相比，Spikingformer 在所有任务中都提高了网络性能。请注意，Spikingformer-4-384-400E 表示 Spikingformer 包含四个尖峰变换器块和 384 个特征嵌入维度，训练时间为 400 epochs。Spikingformer 的其他模型默认使用 310 个历时进行训练，这与 Spikformer 一致。

方法		架构参数		CIFAR10	CIFAR100
时间		(M)	步骤	Top-1 Acc	Top-1 Acc
混合训练[53]	VGG-11	9.27	125	92.22	67.87
饮食-SNN[54]	ResNet-20	0.27	10/5	92.54	64.07
STBP[55]	CIFARNet	17.54	12	89.83	-
STBP NeuNorm[56]	CIFARNet	17.54	12	90.53	-
TSSL-BP[57]	CIFARNet	17.54	5	91.41	-
STBP-tdBN[6]	ResNet-19	12.63	4	92.92	70.86
TET[51]	ResNet-19	12.63	4	94.44	74.47
MS-ResNet[7]	ResNet-104	-	-	91.72	66.83
	ResNet-482	-	-	91.90	-
ANNs[8]	ResNet-19*	12.63	1	94.97	75.35
	变压器-4-384	9.32	1	96.73	81.02
Spikformer[8]	Spikformer-4-256	4.15	4	93.94	75.96
	Spikformer-2-384	5.76	4	94.80	76.95
	Spikformer-4-384	9.32	4	95.19	77.86
	Spikformer-4-384-400E	9.32	4	95.51	78.21
斯派金格玛	Spikingformer-4-256	4.15	4	**94.77(+0.83)**	**77.43(+1.47)**
	Spikingformer-2-384	5.76	4	**95.22(+0.42)**	**78.34(+1.39)**
	Spikingformer-4-384	9.32	4	**95.61(+0.42)**	**79.09(+1.23)**
	Spikingformer-4-384-400E	9.32	4	**95.81(+0.30)**	**79.21(+1.00)**

为 AdamW，在 310 个训练历元期间，批量大小设置为 192 或 288，余弦衰减学习率的初始值为 0.0005。在 ImageNet 和 CIFAR 上训练时，缩放因子为 0.125。Spiking Tokenizer 中的四个 SPED 将图像分割成 196 个 16 × 16 补丁。

与 Spikformer 一样，我们针对 ImageNet 尝试了不同嵌入维度和变换块数量的各种模型，如表 1 所示。1. 我们还显示了突触操作（SOP）[52] 和理论能耗的比较。一方面，Spkingformer 遵循尖峰驱动计算规则，有效避免了浮点乘法和整数-浮点乘法。The histogram of the input data for each transformer block of Spikingformer and Spikformer is shown in Appendix E of Supplementary Material, our Spikingformer effectively avoids producing non-spike data of non-spike computations in Spikformer.另一方面，Spikingformer-8-512 使用 4 个时间步骤在 ImageNet 上实现了 74.79% 的 top-1 分类准确率，显著优于 Spikformer-8-512 1.41%，优于 MS-ResNet 模型 0.58%，优于 SEW ResNet-152 模型 5.53%。Spikingformer-8-512 is with 7.463 mJ theoretical energy consumption, which reduces energy consumption by 60.36%, compared with 18.819 mJ of Spikformer-8-512.Spikingformer-8-768 使用 4 个时间步数在 ImageNet 上实现了 75.85% 的 top-1 分类准确率，显著优于 Spikformer-8-768 1.04%，优于 MS-ResNet 模型 1.64%，优于 SEW ResNet-152 模型 6.59%。Spikingformer- 8-768 is with 13.678 mJ theoretical energy consumption, which reduces energy consumption by 57.34%, compared with 32.074 mJ of Spikformer-8-768.In addition, we recalculate the energy consumption of Spikformer in Appendix G because the non-spike computation of Spikformer can not be directly calculated by Sec.3.4.The main reason why Spikingformer can significantly reduce energy consumption compared with Spikformer is that Spikingformer could effectively avoid integer-float

multiplications, and the secondary reason is that our models have lower firing rate on ImageNet, which is shown in Appendix E.

4.2 CIFAR 分类

CIFAR10/CIFAR100 提供 50,000 张训练图像和 10,000 张测试图像，分辨率为 32×32。不同之处在于，CIFAR10 包含 10 个分类类别，而 CIFAR100 包含

表 3：神经形态数据集、CIFAR10-DVS 和 DVS128 手势的结果。请注意，Spikformer 的结果是我们根据其开源代码实现的。

方法	CIFAR10-DVS		DVS128 手势	
	时间步骤	Acc	时间步骤	Acc
LIAF-Net [58]TNNLS-2021	10	70.4	60	97.6
TA-SNN [59]ICCV-2021	10	72.0	60	98.6
推出 [60]Front.神经科学-2020	48	66.8	240	97.2
DECOLLE [61]Front.Neurosci-2020	-	-	500	95.5
tdBN [6]AAAI-2021	10	67.8	40	96.9
PLIF [62]ICCV-2021	20	74.8	20	97.6
SEW ResNet [5]NeurIPS-2021	16	74.4	16	97.9
Dspike [63]NeurIPS-2021	10	75.4	-	-
SALT [64]神经网络w-2021	20	67.1	-	-
DSR [23]CVPR-2022	10	77.3	-	-
MS-ResNet [7]	-	75.6	-	-
	10	78.6	10	95.8
	16	80.6	16	97.9
Spikformer[8] （我们的实施方案）	10	**79.9(+1.3)**	10	**96.2(+0.4)**
	16	**81.3(+0.7)**	16	**98.3(+0.4)**

斯派克成型机（我们的）

100 个类别，为分类算法提供了更好的区分能力。Spikingformer 的批量大小设置为 64。我们在 Spiking Tokenizer 模块中选择两个 SPE 和两个 SPED，将输入图像分割成 64 个 4 × 4 补丁。

实验结果如表 2 所示。 从结果中，我们发现 Spikingformer 模型的性能超过了所有具有相同参数数的 Spikformer 模型。在 CIFAR10 中，我们的 Spikingformer-4-384-400E 达到了 95.81% 的分类准确率，明显优于 Spikformer-4-384-400E 0.30%，优于 MS-ResNet-482 3.91% 。在 CIFAR100 中， Spikingformer-4-384-400E 的分类准确率为 79.21%，明显优于 Spikformer-4-384-400E 1.00%， 优于 MS-ResNet-110 12.38%。据我们所知，我们的 Spikingformer 在 CIFAR10 和 CIFAR100 上都达到了直接训练的纯尖峰驱动 SNN 模型的最高水平。在 CIFAR10 和 CIFAR100 中，ANN-Transformer 模型分别比 Spikformer-4-384 高出 1.12% 和 1.93%。在本实验中，我们还发现 Spikingformer 的性能与 CIFAR 数据集中一定范围内的块数、维数和训练历时呈正相关。

4.3 DVS 分级

CIFAR10-DVS 分类。CIFAR10-DVS 是一个神经形态数据集，由静态图像数据集转换而来，通过移动 DVS 摄像机捕捉的图像样本，提供 9,000 个训练样本和 1,000 个测试样本。我们在 DVS-Gesture 上比较了我们的方法和 SOTA 方法。具体来说，由于 CIFAR10-DVS 的图像大小为 128*128，我们在尖峰标记块中采用了 4 个 SPED，并采用了 2 个 256 补丁嵌入维度的尖峰变换器块。尖峰神经元的时间步数为 10 或 16。训练历元数为 106，与 Spikformer 相同。学习率初始化为 0.1，并按余弦计划衰减。

CIFAR10-DVS 的结果如表 3 所示。Spikingformer 在 16 个时间步数下的 top-1 准确率为 81.3%，在 10 个时间步数下的 top-1 准确率为 79.9%，分别比 Spikformer 高出 1.3% 和 0.7% 。据我们所知，在 CIFAR10-DVS 上直接训练的纯尖峰驱动 SNN 模型中，我们的 Spikingformer 达到了最先进水平。

DVS128 手势分类。DVS128 手势是一个手势识别数据集，其中包含 29 个个体在 3 种光照条件下做出的 11 种手势类别。DVS128 Gesture 的图像大小为 128*128。DVS128 手势分类的主要超参数设置与 CIFAR10-DVS 分类相同。唯一不同的是，DVS 手势分类的训练历元数设置为 200，这与 Spikformer 相同。

表 3 将我们的方法与 CIFAR10-DVS 上的 SOTA 方法进行了比较。Spikingformer 在 16 个时间步长下获得了 98.3% 的 top-1 准确率，在 10 个时间步长下获得了 96.2% 的准确率，分别比 Spikformer 高出 0.4% 和 0.4%。

5 讨论

5.1 最后一层的进一步分析

我们在 CIFAR10 和 CIFAR100 数据集中对 Spikingformer 的最后一层进行了分析，以研究其影响，结果如表 4 所示。最后一层对模型性能有重大影响，尽管它只在 Spikingformer 中，尖峰信号只是一个很小的组成部分。我们的实验表明，使用尖峰信号的 *Spikingformer-L-D** 整体表现不如 Spikingformer-L-D。这可以归因于 AvgPooling 的 AvgPooling 层--FC pro--D-- 和 Spikingformer-L-D 的 AvgPooling 层--FC pro--。Spikingformer-L-D 使用浮点数，其分类能力强于 SN - AvgPooling - FC 中的尖峰信号。不过，*Spikingformer*-L-D* 仍然优于使用 SN - AvgPooling - FC 层作为分类器的 Spikformer-*L*-D。一方面，Spikingformer 有效地避免了 Spikformer 在残差学习中的整数-浮点乘法。另一方面，与 Spikformer 相比，Spikingformer 具有更好的尖峰特征提取和分类能力。这些结果进一步验证了 Spikingformer 作为骨干的有效性。

表 4：关于 Spikingformer 最后一层的讨论结果。*Spikingformer-L-D** 表示最后一层为 SN - AvgPooling - FC 的 Spikingformer。Spikingformer-L-D *指*默认情况下最后一层为 AvgPooling - FC 的 Spiking-前者。

数据集	模型	时间步	前 1 加速
CIFAR10	Spikingformer-4-384-400E	4	95.81
	Spikingformer-4-384-400E 4*		95.58
	Spikformer-4-384-400E	4	95.51
CIFAR100	Spikingformer-4-384-400E	4	79.21
	Spikingformer-4-384-400E 4*		78.39
	Spikformer-4-384-400E	4	78.21

5.2 关于 "*激活-转换-批量规范*"范式的更多讨论

*激活-反转-批处理规范*是二元神经网络（BNN）[13, 14, 15, 16]的基本构件。 MS-ResNet[7] 继承了基于卷积的二元神经网络中的*激活-反转-批处理规范，*在 CIFAR10 上成功地将深度扩展到了 482 层，而且没有出现退化问题。MS-ResNet 主要验证了 *Activation-Conv-BatchNorm* 克服基于卷积的 SNN 模型梯度爆炸/消失和性能下降问题的能力。 相比之下，据我们所知，Spikingformer 是第一个基于变换器的 SNN 模型，它使用*激活*-反转-*批规范*范式实现了纯尖峰驱动计算。这项工作进一步验证了*激活-反转-批量规范*作为 SNN 设计基本

模块的有效性和基础性。具体来说，Spikingformer 在五个数据集上取得了最先进的性能，并以显著的优势超过了 MS-ResNet：在 ImageNet、CIFAR10、CIFAR100、CIFAR10-DVS、DVS-Guesture 数据集上，MS-ResNet （我们的 Spikingformer）取得了 74.21% (75.85%)、91.90
(95.81%)、66.83% (79.21%)、75.6% (81.3%)、- (98.3%)。

6 结论

在本文中，我们提出了尖峰驱动残差学习 SNN，以避免 Spikformer 和 SEW ResNet 中的非尖峰计算。基于这种残差设计，我们开发了一种纯尖峰驱动的基于变压器的尖峰神经网络，命名为 Spikingformer。我们在 ImageNet、CIFAR10、CIFAR100、CIFAR10-DVS 和 DVS128 手势数据集上对 Spikingformer 进行了评估。实验结果验证了 Spikingformer 能有效避免 Spikformer 中的整数-浮点乘法。此外，Spikingformer 作为一种新的先进 SNN 骨干，在上述所有数据集中的表现都远远优于 Spikformer（例如，在 ImageNet 中 + 1.04%，在 CIFAR100 中 + 1.00%）。据我们所知，Spikingformer 是第一个纯尖峰驱动的基于变换器的 SNN 模型，在上述数据集上取得了最先进的直接训练纯 SNN 模型的性能。

7 鸣谢

本研究得到国家自然科学基金 62236009 和 62206141 的资助。

参考资料

[1] 沃尔夫冈-马斯尖峰神经元网络：第三代神经网络模型
神经网络，10（9）：1659-1671，1997。

[2] Kaushik Roy、Akhilesh Jaiswal 和 Priyadarshini Panda。利用神经形态计算实现基于尖峰的机器智能。*自然》*，575（7784）：607-617，2019。

[3] Guangyao Chen, Peixi Peng, Guoqi Li, and Yonghong Tian.通过辅助积累途径训练全尖峰神经网络》，*arXiv preprint arXiv:2301.11929*, 2023.

[4] Yangfan Hu、Huajin Tang 和 Gang Pan.尖峰深度残差网络。*IEEE 神经网络与学习系统论文集》*，第 1-6 页，2021 年。

[5] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timoth Masquelier, and Yonghong Tian. 尖峰神经网络中的深度残差学习。*国际神经信息处理系统会议（NeurIPS）论文集》*，第34卷，第21056- 21069页，2022年。

[6] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li.利用直接训练的大型尖峰神经网络深入研究。美国*人工智能学会（AAAI）会议论文集》*，第 11062-11070 页，2021 年。

[7] 胡一帆、吴玉杰、邓磊、李国琦。推进残差学习以实现强大的深度尖峰神经网络》，*arXiv preprint arXiv:2112.08954*, 2021.

[8] 周兆坤、朱跃生、何超、王耀伟、严水成、田永红和袁莉。Spikformer：当尖峰神经网络遇到变压器。*第十一届学习表征国际会议*，2023 年。

[9] Ali Lotfi Rezaabad 和 Sriram Vishwanath.长短期记忆尖峰网络及其应用。*2020年神经形态系统国际会议（ICONS）论文集》*，第1-9页，2020年。

[10] Zulun Zhu, Jiaying Peng, Jintang Li, Liang Chen, Qi Yu, and Siqiang Luo.尖峰图卷积网络。*第三十一届国际人工智能联合会议（IJCAI）论文集》*，第 2434-2440 页，2022 年。

[11] Yangfan Hu、Huajin Tang 和 Gang Pan.尖峰深度残差网络。*电气和电子工程师学会神经网络与学习系统论文集》*，2021 年。

[12] 马克-霍洛维茨1.1 计算的能源问题（以及我们能做些什么）。In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 10-14.IEEE, 2014.

[13] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng.双实数网：利用改进的表示能力和高级训练算法提高 1 位 cnns 的性能。*欧洲计算机视觉会议（ECCV）论文集》*，第 722-737 页，2018 年。

[14] Nianhui Guo, Joseph Bethge, Haojin Yang, Kai Zhong, Xuefei Ning, Christoph Meinel, and Yu Wang.Boolnet: minimizing the energy consumption of binary neural networks. *arXiv preprint arXiv:2106.06991*, 2021.

[15] Yichi Zhang、Zhiru Zhang 和 Lukasz Lew。Pokebnn：对轻量级准确性的二元追求。*IEEE/CVF 计算机视觉与模式识别大会论文集*》，第 12475-12485 页，2022 年。

[16] Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng.Reactnet：实现具有广义激活函数的精确二元神经网络。In *Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XIV 16*, pages 143-159.Springer, 2020.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei.Imagenet：大规模分层图像数据库。*IEEE/CVF 计算机视觉与模式识别大会论文集》*，第 248-255 页，2009 年。

[18] 亚历克斯-克里热夫斯基从微小图像中学习多层特征2009.

[19] 曹永强、陈扬和 Deepak Khosla.用于高能效物体识别的尖峰深度卷积神经网络。*国际计算机视觉杂志》*，113（1）：54-66，2015。

[20] Eric Hunsberger 和 Chris Eliasmith.具有生命神经元的尖峰深度网络》，*arXiv preprint arXiv:1510.08829*, 2015.

[21] Bodo Rueckauer、Iulia-Alexandra Lungu、Yuhuang Hu、Michael Pfeiffer 和 Shih-Chii Liu。将连续值深度网络转换为高效事件驱动网络用于图像分类。*神经科学前沿》*，11:682，2017.

[22] 卜彤、方伟、丁建浩、戴鹏林、于兆飞、黄铁军。高精确度和超低延迟尖峰神经网络的最优ann-snn转换。*国际学习表征会议（ICLR）*，2021 年。

[23] 孟庆艳、肖明清、沈艳、王义森、林周臣、罗志权。通过尖峰表征差异化训练高性能低延迟尖峰神经网络*ArXiv preprint arXiv:2205.00459*, 2022.

[24] 王雨辰、张璐璐、陈怡和曲虹。带记忆的符号神经元：实现简单、准确和高效的 ann-snn 转换。*国际人工智能联合会议*，2022 年。

[25] Jun Haeng Lee、Tobi Delbruck 和 Michael Pfeiffer.使用反向传播训练深度尖峰神经网络。*神经科学前沿》*，10:508，2016.

[26] 苏米特-B-什雷斯塔（ Sumit B Shrestha）和加里克-奥查德（Garrick Orchard）。杀手尖峰层错误及时重新分配。*国际神经信息处理系统会议论文集》（NeurIPS）*，第 31 卷，2018 年。

[27] Chankyu Lee、Syed Shakib Sarwar、Priyadarshini Panda、Gopalakrishnan Srinivasan 和 Kaushik Roy。基于尖峰的反向传播训练深度神经网络架构。*神经科学前沿》*，14:119，2020 年。

[28] Emre O Neftci、Hesham Mostafa 和 Friedemann Zenke。尖峰神经网络中的替代梯度学习：将基于梯度的优化功能引入尖峰神经网络。*IEEE 信号处理杂志*，36（6）：51-63，2019.

[29] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。注意力就是你所需要的一切。*神经信息处理系统国际会议论文集》（NeurIPS）*，第 30 卷，2017 年。

[30] Alexey Dosovitskiy、Lucas Beyer、Alexander Kolesnikov、Dirk Weissenborn、Xiaohua Zhai、Thomas Unterthiner、Mostafa Dehghani、Matthias Minderer、Georg Heigold、Sylvain Gelly 等。图像胜过 16x16 个单词：大规模图像识别变换器。*国际学习表征会*

议（ICLR），2020 年。

[31] Li Yuan、Yunpeng Chen、Tao Wang、Weihao Yu、YuJun Shi、Zi-Hang Jiang、Francis EH Tay、Jiashi Feng 和 Shuicheng Yan。令牌到令牌的维度：在图像网络上从头开始训练视觉转换器。*IEEE/CVF 计算机视觉国际会议（ICCV）论文集》*，第 558-567 页，2021 年。

[32] Nicolas Carion、Francisco Massa、Gabriel Synnaeve、Nicolas Usunier、Alexander Kirillov 和 Sergey Zagoruyko。用变换器进行端到端物体检测。*欧洲计算机视觉会议（ECCV）论文集》*，第 213-229 页。Springer, 2020.

[33] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai.可变形检测器：用于端到端对象检测的可变形变换器。*arXiv preprint arXiv:2010.04159*, 2020.

[34] Ze Liu、Yutong Lin、Yue Cao、Han Hu、Yixuan Wei、Zheng Zhang、Stephen Lin 和 Baining Guo。Swin 变换器：使用移位窗口的分层视觉变换器。In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012-10022, 2021.

[35] 王　文海、谢恩泽、李翔、范登平、宋开涛、梁鼎、卢彤、罗平、邵玲。金字塔视觉转换器：无需卷积即可实现密集预测的多功能骨干网。*IEEE/CVF 计算机视觉国际会议（ICCV）论文集》*，第 568-578 页，2021 年。

[36] Li Yuan、Qibin Hou、Zihang Jiang、Jiashi Feng 和 Shuicheng Yan。Volo：用于视觉识别的视觉展望器。*ArXiv 预印本 arXiv:2106.13112*, 2021.

[37] 李玉东、雷云林、杨旭。Spikeformer：用于训练高性能低延迟尖峰神经网络的新型架构。*arXiv preprint arXiv:2211.10686*, 2022.

[38] Tete Xiao、Mannat Singh、Eric Mintun、Trevor Darrell、Piotr Dollár 和 Ross Girshick。早期卷积帮助变压器看得更清楚。*国际神经信息处理系统会议（NeurIPS）论文集》*，第 34 卷，第 30392-30400 页，2021 年。

[39] Ali Hassani、Steven Walton、Nikhil Shah、Abulikemu Abuduweili、Jiachen Li 和 Humphrey Shi。用紧凑型转换器摆脱大数据范式。*arXiv 预印本 arXiv:2104.05704*, 2021.

[40] Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungong Han.Acnet：通过非对称卷积块强化内核骨架，实现强大的 cnn。In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1911-1920, 2019.

[41] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun.Repvgg：让 Vgg 风格的 Convnets 再次伟大。*IEEE/CVF 计算机视觉与模式识别会议论文集》*，第 13733-13742 页，2021 年。

[42] Souvik Kundu、Massoud Pedram 和 Peter A Beerel。Hire-snn：通过精心设计的输入噪声训练利用高能效深度尖峰神经网络的固有鲁棒性。In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5209-5218, 2021.

[43] Souvik Kundu、Gourav Datta、Massoud Pedram 和 Peter A Beerel。Spike-thrift：通过注意力引导的压缩限制尖峰活动，实现高能效深度尖峰神经网络。*IEEE/CVF 计算机视觉应用冬季会议（WACV）论文集》*，第 3953-3962 页，2021 年。

[44] Bojian Yin, Federico Corradi, and Sander M Bohté.自适应尖峰递归神经网络的精确高效时域分类。*自然机器智能》*，3（10）：905-913，2021 年。

[45] Priyadarshini Panda、Sai Aparna Aketi 和 Kaushik Roy.利用后向残差连接、随机软最大值和杂交实现可扩展、高效和精确的深度尖峰神经网络。*神经科学前沿》*，14:653，2020 年。

[46] Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li.注意力尖峰神经网络。*电气和电子工程师学会模式分析与机器智能论文集》*，2023 年。

[47] 阿农-阿米尔、布赖恩-塔巴、戴维-伯格、蒂莫西-梅拉诺、杰弗里-麦金斯特里、卡梅洛-迪-诺尔福、塔潘-纳亚克、亚历山大-安德烈奥普洛斯、纪尧姆-加罗、马塞拉-门多萨、杰夫-库斯尼茨、迈克尔-德伯勒、史蒂夫-埃塞尔、托比-德尔布吕克、迈伦-弗里

克纳和达曼德拉-莫达。低功耗、完全基于事件的手势识别系统。In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7243-7252, 2017.

[48] Adam Paszke、Sam Gross、Francisco Massa、Adam Lerer、James Bradbury、Gregory Chanan、Trevor Killeen、Zeming Lin、Natalia Gimelshein、Luca Antiga 等 Pytorch： 命令式高性能深度学习库。*国际神经信息处理系统会议（NeurIPS）论文集》*，第 32 卷，2019 年。

[49] 方伟、陈彦琪、丁建豪、陈定、余兆飞、周慧慧、Timothée Masquelier、田永红及其他撰稿人。Spikingjelly. https://github.com/ fangwei123456/spikingjelly, 2020.已访问：年月日。

[50] 罗斯-怀特曼。Pytorch 图像模型。https://github.com/rwightman/ pytorch-image-models, 2019.

[51] 邓世光、李宇航、张尚航、顾石通过梯度再加权实现尖峰神经网络的时空高效训练*国际学习表征会议（ICLR）*，2021年。

[52] Paul A Merolla、John V Arthur、Rodrigo Alvarez-Icaza、Andrew S Cassidy、Jun Sawada、Filipp Akopyan、Bryan L Jackson、Nabil Imam、Chen Guo、Yutaka Nakamura 等：具有可扩展通信网络和接口的百万穗状神经元集成电路。*科学*》，345（6197）：668-673，2014。

[53] Nitin Rathi、Gopalakrishnan Srinivasan、Priyadarshini Panda 和 Kaushik Roy.利用混合转换和尖峰时序相关反向传播实现深度尖峰神经网络。*arXiv preprint arXiv:2005.01807*, 2020.

[54] Nitin Rathi 和 Kaushik Roy.Diet-snn：*arXiv preprint arXiv:2008.03658*, 2020.

[55] 吴玉洁、邓磊、李国琦、朱军、史鲁平。用于训练高性能尖峰神经网络的时空反向传播。*神经科学前沿*》，12:331，2018.

[56] 吴玉洁、邓磊、李国琦、朱军、谢媛和史鲁平。尖峰神经网络的直接训练：更快、更大、更好。2019年*美国人工智能学会（AAAI）会议论文集*，第1311-1318页。

[57] 张文瑞和李鹏通过反向传播进行深度尖峰神经网络的时间尖峰序列学习。*国际神经信息处理系统会议（NeurIPS）论文集*》，第 33 卷，第 12022-12033 页，2020 年。

[58] Zhenzhi Wu, Hehui Zhang, Yihan Lin, Guoqi Li, Meng Wang, and Ye Tang.LIAF-Net：用于轻量级高效时空信息处理的漏积分和模拟火网络。*IEEE 神经网络与学习系统论文集*》，第 1-14 页，2021 年。

[59] Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li.用于事件流分类的时空注意力尖峰神经网络。*IEEE/CVF 计算机视觉国际会议（ICCV）论文集*》，第 10221-10230 页，2021 年。

[60] Alexander Kugele、Thomas Pfeil、Michael Pfeiffer 和 Elisabetta Chicca。利用尖峰神经网络高效处理时空数据流》。*神经科学前沿*》，14:439，2020 年。

[61] Jacques Kaiser、Hesham Mostafa 和 Emre Neftci。深度连续局部学习（DECOLLE）的突触可塑性动力学。*神经科学前沿*》，14:424，2020 年。

[62] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian.利用可学习膜时间常数增强尖峰神经网络的学习能力*IEEE/CVF 计算机视觉国际会议（ICCV）论文集*》，第 2661-2671 页，2021 年。

[63] Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu.Differ- entiable Spike：梯度下降训练尖峰神经网络的反思。*国际神经信息处理系统会议论文集》（NeurIPS）第*34 卷，第 23426-23439 页，2021 年。

[64] Youngeun Kim 和 Priyadarshini Panda。为动态视觉传感优化深度尖峰神经网络》。*神经网络*》，144:686-698，2021 年。

附录

A 尖峰神经元模型

尖峰神经元是 SNN 的基本单元，我们在工作中选择泄漏积分与发射（LIF）模型作为尖峰神经元。LIF 神经元的动态过程可表述如下：

$$H[t] = V[t-1] + \frac{1}{\tau}(X[t] - (V[t-1] - V_{\text{reset}})), \tag{21}$$

$$S[t] = \Theta(H[t] - V_{th}) \text{、} \tag{22}$$

$$V[t] = H[t](1 - S[t]) + V_{\text{reset}} S[t] \tag{23}$$

当膜电位 $H[t]$ 超过点火阈值 V_{th} 时，尖峰神经元将触发尖峰 $S[t]$。

$\Theta(v)$ 是海维塞德阶跃函数，当 $v \geq 0$ 时等于 1，否则等于 0。$V[t]$ 代表触发事件后的膜电位，如果没有产生尖峰，则等于 $H[t]$，否则等于复位电位 V_{reset}。

B 卷积与批量归一化融合的理论分析

很明显，二进制尖峰值在通过卷积层后就变成了浮点数值，这就导致了随后在批量归一化（BN）层中的 MAC 运算。然而，卷积的同质性使得后续的 BN 和线性缩放变换可以等效地融合到卷积层中，并在部署时增加偏置。具体来说，每个 BN 层及其前一个卷积层都会融合成一个带有偏置向量的卷积 ConvBN 层。ConvBN 的内核和偏置 $\{W, B\}$ 可由 $\{W, \mu, \sigma, \gamma, \beta\}$ 计算得出。输入元素 x_i 的卷积和批归一化过程可表述如下：

$$y_{Conv} = w_{Conv} * x_i + b_{Conv} \tag{24}$$

$$y_i = BN_{\gamma,\beta}(x_i) = \gamma \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta = \frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}} x_i + \beta - \frac{\gamma \mu}{\sqrt{\sigma^2 + \varepsilon}} \tag{25}$$

因此，在部署过程中，批量归一化可以表述为：$y_{BN} = w_{BN} * x_i + b_{BN}$。因此，上述步骤可以融合：

$$y_i = w_{BN}(w_{Conv} * x_i + b_{Conv}) + b_{BN} = w_{BN} * w_{Conv} * x_i + w_{BN} b_{Conv} + b_{BN} \tag{26}$$

等效卷积层 ConBN：$W = w_{BN} * w_{Conv}$；$B = w_{BN} * b_{Conv} + b_{BN}$。

C Spikingformer 中的多头尖峰自我关注

多头尖峰自注意（MSSA）可以简单地表述如下：

$$X' = SN(X) \tag{27}$$

$$Q、K、V = SN_q(ConvBN_q(X')), SN_k(ConvBN_k(X')), SN_v(ConvBN_v(X'))' \tag{28}$$

$$Q', K', V' = (q_1, q_2, \ldots, q_H), (k_1, k_2, \ldots, k_H), (v_1, v_2, \ldots, v_H) \tag{29}$$

$$MSSA(Q', K', V') = ConvBN\ SN\ q\ k_1^T v_1 * s, \ldots, q\ k_H^T v_H * s \tag{30}$$

其中 $Q, K, V \in R^{T \times N \times D}$，并重塑为 H 头形式 $Q', K', V' \in R^{T \times H \times N \times d}$，$D =$。
注意，在 SSA 或 MSSA 中，缩放因子 s 是一个常数，可以很容易地融合到以下的尖峰神经元 LIF 层。

D 补充实验细节

在实验中，我们使用 8 个 GPU 训练 ImageNet 上的模型，而使用 1 个 GPU 训练其他数据集（CIFAR10、CIFAR100、DVS128 Gesture、CIFAR10-DVS）。此外，在 DVS 数据集上训练模型时，我们调整了尖峰神经元的膜时间常数 τ 值。在使用代理函数直接训练 SNN 模型时，我们使用了以下方法

$$Sigmoid(x) = \frac{1}{1 + \exp(-\alpha x)} \tag{31}$$

在所有实验中，我们选择$\alpha = 4$ 的 Sigmoid 函数作为替代函数。

E Spikformer 和 Spikingformer 的尖峰数据统计

E.1 Spikformer 在 ImageNet 上的统计数据

图 3 显示了 Spikformer-8-512 在 ImageNet 上每个区块输入数据的直方图。具体来说，我们在整个 ImageNet 测试集上运行经过训练的 Spikformer-8-512 得到了结果。Spikformer-8-512 有 8 个变压器块，每个变压器块有两个重合连接。因此，Spikformer-8-512 中的非尖峰数字可以累积到 16 个。可视化结果进一步验证了我们的分析：Spikformer 的残差学习导致 ConvBN 层的非尖峰计算（整数-浮点乘法）。

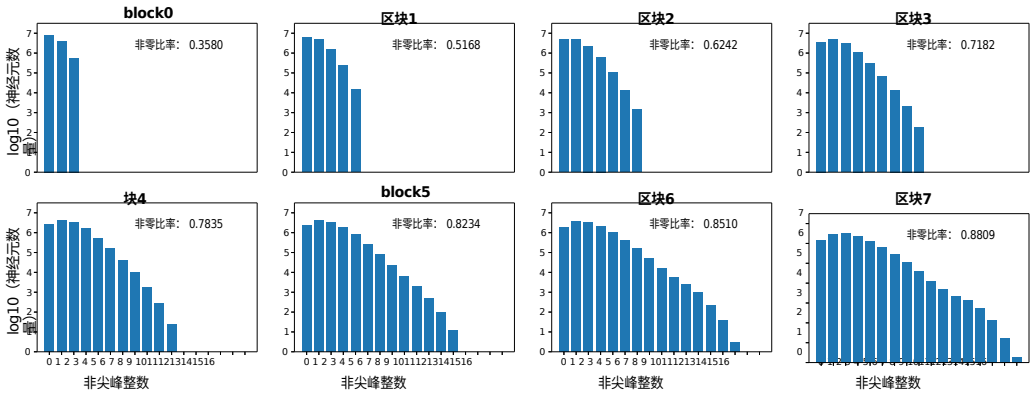


图 3：Spikformer-8-512 中各 块在 ImageNet 上的输入数据直方图。腹轴表示 Spikformer 变换器块中 ConvBN 层之前的非尖峰数据范围，即 {0、1、2、……、16}。纵座标指每种情况下神经元数量的*对*数值（以 10 为底的对数）。非零比率表示每个区块非零输入单元的比率。

E.2 Spikingformer 在 ImageNet 上的统计数据

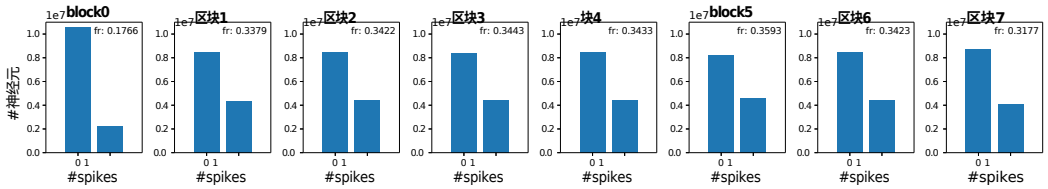


图 4：Spikingformer-8-512 中各 块在 ImageNet 上的输入数据直方图。abscissa指的是 Spikingformer 变压器区块中 ConvBN 层之前 {0, 1} 的二进制尖峰数据。柱状图表示{0, 1} 的神经元编号。

如 E.1 节所述，我们在图 4 中绘制了 Spikingformer-8-512 在 ImageNet 上每个区块输入数据的直方图。结果表明，我们在 Spikingformer 中提出的尖峰驱动残差学习可以有效避免 Spikformer 中常见的整数-浮点乘法。此外，图 4 中的 fr 表示 Spikingformer 中每个尖峰变换器块的输入数据的发射率。我们观察到，与 Spikformer（图 3）相比，Spikingformer 在 ImageNet 上的触发率更低，这进一步减少了突触操作，从而降低了能耗。

F 其他结果

F.1 CIFAR10 的其他分类结果

我们发现 Spikingformer 在上表中的某些数据集上并不完全收敛。例如，Spikingformer 训练到 600 个 epoch 后，准确率提高到 96.04%。为了在相同条件下与其他方法进行比较，我们选择了 300 或 400 epochs。实际上，我们发现在实践中，SNN 模型的收敛速度要比类似结

构的 ANN 模型慢得多。

表 5：在 CIFAR10 上训练 Spikingformer 达到 600 个历元。

骨干网	时间进度	CIFAR10
Spikingformer-4-384-300E	4	95.61
Spikingformer-4-384-400E	4	95.81
Spikingformer-4-384-600E	4	**96.04**

G ImageNet 上 Spikformer 的能耗重新计算。

Spikformer 的能耗将 ConvBN 层的非尖峰计算（整数-浮点乘法）视为基于二进制尖峰的累加操作。这显然是不合理的。因此，我们提供了两种方法来重新计算 Spikformer 在 ImageNet 上的能耗：

1) 将整数 N（$N > 1$）与浮点数的乘法运算视为 N 次基于二进制尖峰的累加运算；2）将整数 N 与浮点数的乘法运算视为浮点乘法运算，这会导致更高的能耗。本工作中的重新计算是按照第一种方法进行的。