神经形态计算与工程

Crossmark

收稿日期dd Month yyyy修订日期dd Month yyyy

论文

基于行的事件预处理：迈向低能耗神经形态计算机视觉

Am´elie Gruel[1], Pierre Lewden[1], Adrien F. Vincent[1]和 Sylvain Sa¨ıghi[1]

[1]Lab材料与系统集成实验室，波尔多大学、波尔多国立综合理工学院、法国国家科学研究中心，法国塔朗斯 F-33400 ce
*Aut如需联系，请联系作者。

电子邮件： {firstname.surname}@波尔多大学

关键词： 神经形态视觉、能效、脉冲神经网络、事件数据预处理

arXiv:2601.10742v1 [cs.NE] 10 Jan 2026

摘要

近年来，神经形态视觉取得了显著进展，这得益于脉冲神经网络与事件数据在生物学启发、节能、动态视觉数据处理的延迟和内存使用量等方面天然契合。然而，优化其能耗需求仍是该领域的一项挑战，尤其是在嵌入式应用中。一种解决方案是对事件进行预处理，以优化数据量，从而降低神经形态硬件上的能耗；该能耗与突触运算次数成正比。为此，我们扩展了一种端到端神经形态直线检测机制，引入基于线的事件数据预处理。我们在三个基准基于事件的数据集上展示了，预处理能够在能耗与分类性能之间实现有利的权衡。根据基于线的预处理策略和分类任务的复杂程度，我们表明，在显著降低理论能耗的同时，可以保持或提高分类准确率。我们的方法能够系统性地显著提升神经形态分类的效率，从而为借助事件预处理实现更节能的神经形态计算机视觉奠定基础。

1 引言

计算机视觉是神经形态计算的重要应用领域，这在一定程度上得益于脉冲神经网络（SNN） [1]与基于事件的相机 [2]之间天然的契合。事实上，与传统人工神经网络（ANN）相比，SNN旨在实现神经架构并通过更充分地借鉴生物机制来进行学习。为此，每个神经元都以脉冲（即一系列电脉冲）的形式接收并处理信息，并利用其膜电位——这种行为模仿了生物神经元，从而在适配的硬件上实现时降低能耗。SNN尤其擅长处理基于事件的相机 [2], 所产生的数据。这类神经形态视觉传感器的设计灵感源自生物视网膜的工作机制：每个像素感知到显著的亮度变化时，便会异步地产生“事件”。

这两者结合后，能够高效地实现许多计算机视觉任务 [3, 4]。然而，在能耗和存储需求方面仍需显著改进：由于事件相机具有较高的时间分辨率（微秒级），例如，Prophesee 的百万像素相机 [5]记录动态场景时会产生大量数据，需要规模较大的架构和大量存储空间。此外，直接处理相机采集的如此大量输入数据，也会导致能耗相应增加。事实上，大多数神经形态架构都需要在每次突触事件发生时从存储器中读取神经元状态，然后再将其写回；这些操作都会消耗一定能量（通常约为 10 pJ） [6]。输入规模越大，所需架构也越大，即突触数量越多，发生突触事件的可能性也越高。因此，我们认为，在执行后续计算机视觉任务之前，应先对事件数据进行预处理，以降低空间分辨率和／或事件数量，从而减少能耗需求。

数据预处理是传统计算机视觉或信号处理中的关键第一步。许多方法可以组合使用，从而产生多种效果，例如降噪（使用高斯模糊、中值模糊、拉普拉斯滤波器等）、去除背景物体（使用边缘

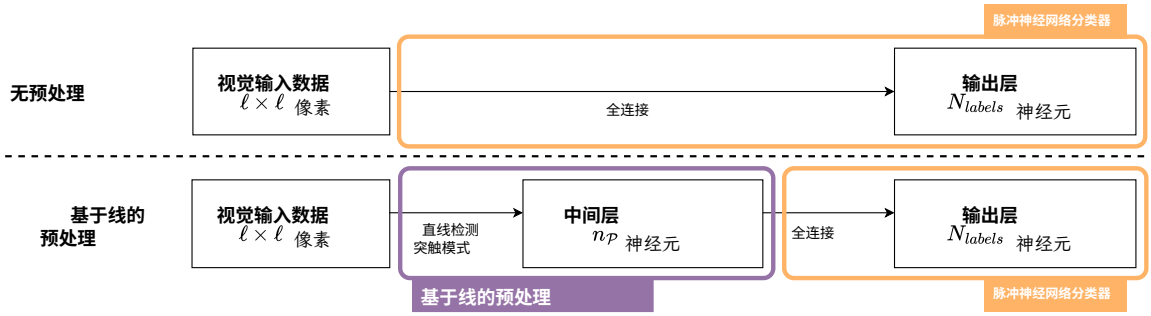


图 1：我们用于评估添加基于线的预处理对神经形态分类影响的方案。参数 ℓ 和 $n_{\mathcal{P}}$ 分别表示输入传感器的像素尺寸，以及预处理层中涉及的神经元数量（详见表 1）。

检测、阈值处理、分水岭等），或扩充数据集（使用裁剪、旋转、颜色调整等） [7]。近期研究已尝试将其中一些方法应用于事件数据，例如 Tonic 框架 [8] ——然而，这通常需要将事件累积成帧，并会失去事件视觉的许多优势。

应谨慎预处理事件数据，以保留甚至提升神经形态能效并减少内存使用量，同时保留数据中的相关信息。如前所述，在最先进的神经形态架构中，“减少 [突触运算]的数量是降低能耗最自然的方式” [6]。因此，应对这一挑战的一种方法是减少数据集中的事件数量，同时保留其所承载的信息，就像 [9, 10, 11] 中开发的神经形态缩放和中央凹处理所实现的那样。

第二种方法是减少突触总数，因为突触运算的数量与之成正比。可以通过预处理事件数据，将其转化为由较少神经元编码的可区分模式，从而减少突触数量。这类似于神经形态特征提取，其目标是“表征常见的事件时空模式 [，使其] 能够在特征空间中被唯一地区分” [12]。文献中已有一些神经形态特征提取的实例，既有基于角点检测 [13, 14], 边缘检测 [15, 16] 的方法，也有直接从数据中学习的方法 [17, 18, 19, 12]。

由于直线检测是生物体感知和理解视觉环境的基本组成部分 [20, 21], ，我们提议将仿生视觉机制中的直线检测应用于事件数据预处理，如图1所示。在这项工作中，我们扩展了近期发表的、用于事件数据直线检测的脉冲神经网络模型 [22], ；据我们所知，这是唯一采用端到端神经形态方法实现这一目标的现有模型。此前的工作无需学习阶段，采用精简的架构，仅依靠脉冲神经网络的内在行为便取得了令人信服的性能，并已在 SpiNNaker 上进行了实验部署 [23]。

我们将这一线检测模型应用于基于事件的基准数据集，以提取特征。我们介绍了不同的策略，并研究了这些策略在三个基准数据集分类任务中的应用；这三个数据集在难度、数据属性和规模方面各不相同：PokerDVS [24], N-MNIST [25] 和 DVS128 Gesture [26]。通过分析用于预处理和分类的架构中的神经元数量和突触数量，我们估算了硬件实现的复杂度以及内存使用量。我们评估了分类性能的变化，以及理论能效（即分类性能与突触事件数量之比），并比较了采用和未采用基于线的预处理的情况。

本研究聚焦于能效及其在传统神经形态计算机视觉任务中的提升。我们的目标是在实现这一点的同时保持令人满意的性能；不过，提升性能只是次要收益，并非必须达成的目标。总之，以下结果验证了我们的目标：我们证明，基于线的神经形态预处理能够在能效与性能之间实现颇具前景的权衡。

2 方法

2.1 直线检测的神经形态模型

我们在 [22] 中介绍的线检测模型，通过输出输入事件数据中横跨整个传感器的静止或运动直线与传感器边界的交点位置，来检测这些直线。对于输入事件数据中的每条直线，两个检测器会激活，分别对应输入直线穿过的边缘

的“上”、“下”、“右”和“左”半区（见图2a）。据此，可以推算出直线端点的坐标，从而确定整条直线的位置和方向。

该机制通过使用一个由 Leaky-Integrate-and-Fire (LIF) 神经元构成的输出层来实现 [27]该输出层包含四个并行的神经元群体，每个群体对应传感器的一条边，并据此命名。在本文其余部分中，这些神经元群体将被称为“检测器”，以区别于提供输入事件的 DVS 相机所指的“传感器”。神经元的漏电特性使其能够随着时间推移稳定地检测移动线条 [22]。通过对每个检测器应用胜者全得机制，确保了模型的选择性，通过 检测器内部的侧向抑制实现（见图 2b）。

每个检测器中的每个神经元，都会接收来自输入神经元的激活值；这些输入神经元位于横跨相应半传感器的对角线上。对角线连接在每个 k 可能的位置上实现，$k = 1$ 用于构成涵盖所有可能对角线的模式。图2b 示意了从输入传感器下半部到 $k = 4$ 底部探测器内神经元 idx 的突触模式。设定的 k 值越高，突触模式中的对角线数量就越少，因此需要实现的突触数量也越少，架构也就越轻量。

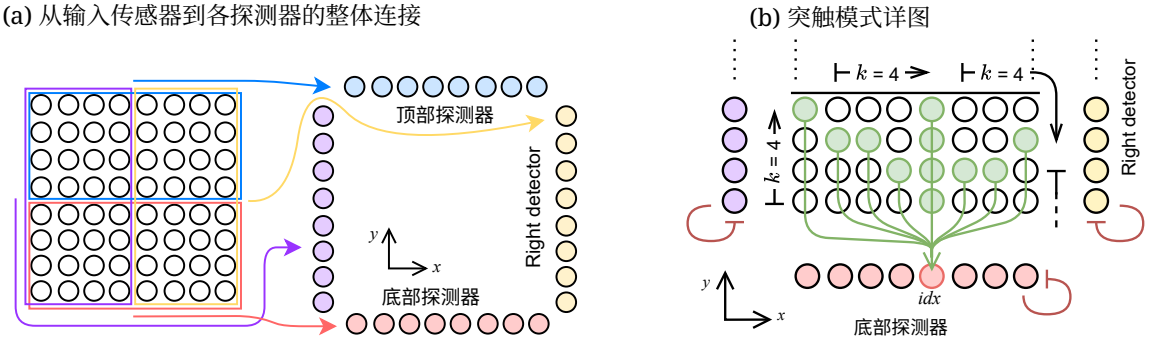


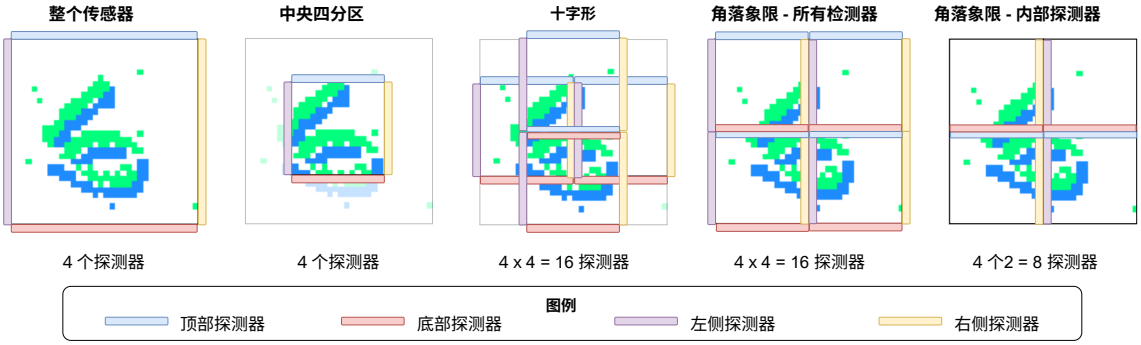
图 2：根据 [22] 改编的示意图，说明本研究所用的线检测模型。(a) 从输入传感器（绘制为由 8×8 个输入像素组成的正方形）到四个检测器的整体连接，即用于预处理的四个神经元群体。(b) 输入传感器与底部探测器之间的激活与抑制突触模式。在此示例中，来自传感器下半部的输入神经元与底部探测器中的神经元 idx 相连，对角线连接之间的步长为 $k = 4$，激活强度为 ω （详见第2.3节）。这一模式应用于四个探测器中的所有神经元。每个探测器均采用胜者全得机制，以确保模型选择性。

2.2 对事件数据预处理的适配

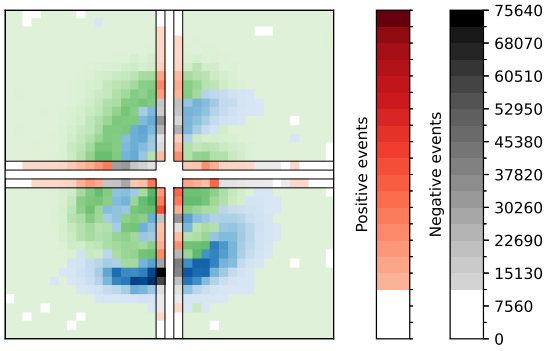
本文提出一种新颖的事件数据预处理方法，利用 [22] 前文所述的模型，从输入事件中提取线性特征。如图3a所示，并在表1中进一步详述，对于尺寸为 $\ell \times \ell$ 像素的输入传感器（其中 ℓ 为传感器以像素计的长度），我们考虑五种不同的策略：

- “整个传感器”：线检测机制原样应用于整个输入传感器。输出分布在 4 个检测器上，因此包含 4ℓ个神经元。
- “中央四分区”：对尺寸为 $\ell/2 \times \ell/2$ 像素的中心块进行直线检测，本文后文将其称为“四分区”。这种方法舍弃了中央四分区以外像素提供的信息，并采用 4 个尺寸减半的检测器，因此包含 2ℓ 个输出神经元。
- “十字形”：将输入传感器拆分为四个组成十字形的四分区。这些四分区在传感器中央区域重叠，而传感器角部产生的数据则被舍弃。线检测模型原样且独立地应用于每个四分区，最终得到 16 个检测器和 8ℓ 个输出神经元。
- “角落四分区——所有检测器”：该策略与前一种策略的区别在于四个象限的空间分布。每个象限都位于传感器的一个不同角落，因此不会忽略任何事件，且中心区域不会发生重叠。该策略还使用 16 个检测器和 8ℓ 个输出神经元。

(a) 基于线的不同预处理策略



(b) “角落四分区—内部检测器”策略的输入事件及其对应产生的脉冲。



(c) 中间“角落四分区—内部检测器”策略产生的聚合脉冲。

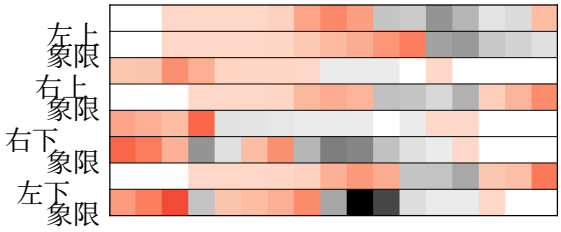


图3：(a) 以一个输入 N-MNIST 样本为例，示意基于线的不同预处理策略。(b) 对标记为 6 的 N-MNIST 样本，采用“角落象限—内部检测器”策略并拆分极性后得到的输入脉冲和输出脉冲。绿色和蓝色圆点分别表示正、负输入事件；红色和黑色分别表示正、负极性的模型输出。事件取自 7,000 个“训练”和“测试”样本并进行平均。(c) 相同情境下，中间预处理层产生并以二维格式聚合的输出脉冲。

- “角落象限——内部检测器”：该策略与前一种策略类似，不过每个象限不再保留 4 个检测器，而只使用 2 个内部检测器（即右下象限使用顶部和左侧检测器，左下象限使用顶部和右侧检测器，右上象限使用底部和左侧检测器，左上象限使用底部和右侧检测器）。事实上，Poker-DVS [24], N-MNIST [25], CIFAR10-DVS [28]或 DVS128 Gesture [26] 等神经形态数据集中的事件大多集中在传感器中心，因此在前一种策略中，外部检测器几乎没有活动。需要注意的是，减少检测器数量（从 16 个减少到 8 个）会使神经元数量减少（从 8ℓ 减少到 4ℓ个输出神经元），突触数量也会减少，从而降低内存使用量和能耗。

表1：采用不同策略时预处理层中的神经元数量。

策略	预处理神经元数量 $n_{\mathcal{P}}$
整个传感器	4ℓ
中央四分区	$4 \times \ell/2 = 2\ell$
十字形	$4 \times 4 \times \ell/2 = 8\ell$
角落四分区—所有检测器	$4 \times 4 \times \ell/2 = 8\ell$
角落四分区—内部检测器	$4 \times 2 \times \ell/2 = 4\ell$

每种策略都有两种变体：一种是以相同方式处理所有输入事件，而不考虑极性（“合并极性”）；另一种是区分正事件和负事件，并按极性分别进行直线检测（“拆分极性”）。在大多数现有的脉冲神经网络模拟器中，事件极性无法直接转换为正、负脉冲；要实现第二种变体，必须将事件拆分为两个神经元群体（每种极性对应一个群体）。因此，在我们的基于线的事件预处理机制中应用“拆分极性”

变体会使输入层中的神经元数量和突触数量都增加一倍。因此，我们分别考察每种策略的两种变体。

基于线的预处理步骤通过 NEST模拟器在中央处理器上实现 [29] ，并通过 PyNN 接口连接 [30]。此实现中使用的神经元和突触参数详见表2。需要注意的是，漏积分发放神经元是通过 PyNN 的对象`IF_cond_exp`实现的，该对象对应于具有指数衰减突触后电导的漏积分发放神经元。

表 2：基于线的预处理步骤中涉及的神经元和突触参数。

参数		数值
静息膜电位 v_{rest}		−60 mV
复位膜电位 v_{reset}		−60 mV
神经元阈值 θ		−30 mV
不应期 τ_r		0.1 毫秒
膜时间常数 τ	m	2.5 毫秒
胜者全得抑制权重 ω_{WTA}		1
兴奋性和抑制性延迟 δ		1
突触电导的衰减时间 τ_{syn}		5 ms

2.3 实验方案

在本文余下部分中，我们将从能耗和分类准确率的角度，讨论基于线的事件数据预处理相较于无预处理阶段的优缺点。本文余下部分将后者称为“无预处理”策略，即直接将输入事件提供给一个没有隐藏层的全连接脉冲神经网络分类器。对于五种基于线的预处理策略，在讨论结果时，我们会将预处理阶段视为已集成到分类流水线中：根据直线检测突触模式将输入事件提供给探测器，然后将探测器神经元的输出馈送至分类器，通过全连接突触模式（见图1）。

超参数微调 为了获得五种基于线的预处理方法的最佳配置，我们针对每个变体对预处理层的两个超参数进行微调：

- 步长 k （线检测模型中对角线之间的步长，在第 2.1 节中介绍）针对 1、5、10、.....到 30 的取值进行评估。根据所研究的数据集，输入传感器尺寸从 34×34 到 128×128 （见表 3）。

- 强度 ω，即根据对角线突触模式，神经元在预处理层中从输入层接收到的激活值之和。其计算方法详见 [22]。强度 ω 在 {1范围内的取值中进行微调：2.5；5；7.5； 10}。

预计将 k 设置得越高，分类器中的突触事件数量就越少，因为突触之间的步长越大，突触数量就越少，从而突触事件也越少。然而，较高的 k 也会导致检测到的线条的精确率降低。同样，较高的 ω 会产生更强的激活，即发射的脉冲数量更多。对于每种预处理策略，可确定两组最优超参数：

- 使推理准确率达到最佳的参数对（k， ω），下文称为“最佳准确率参数化”；

- 第二组参数对（k， ω）则使推理效率达到最佳（即准确率与突触事件数量之比），下文称为“最佳效率参数化”。

分类器 本研究所使用的分类器有意设计得较为简单，因为我们希望在简单的背景下研究基于线的预处理对神经形态效率的影响。事实上，我们并不关注提升基准数据集上的整体分类准确率，而是旨在提高传统计算机视觉任务中的能效，尤其是面向专用嵌入式硬件平台的能效。如上所述，该分类器由单个全连接（即线性）层组成。它接收预处理后的事件作为输入，或直接接收原始事件；其输出层由 N_{labels} 个漏积分发放神经元组成，其中 N_{labels} 对应数据集中的标签数量。

分类器使用时间反向传播，依靠 Adam 优化器训练 10 个周期 [31]（初始学习率为 0.03），并采用均方误差脉冲计数损失。后者支持对脉冲活动进行基于频率的解码，即在给定时间内发放脉冲最多的输出神经元表示检测到的标签，同时通过鼓励错误神经元也产生输出活动，缓解死神经元问题。漏积分发放神经元使用 SNNtorch 框架实现如下 [32]：它们采用改编自 [33] 的反正切替代梯度函数；其膜电位的衰减率 β 设置为 0.9，并在每次发放脉冲时通过减法进行重置。漏积分发放神经元的阈值和批次大小根据数据集而变化（见表 3）。下面呈现的所有结果均为使用不同随机种子运行三次后的平均值。

表 3：所研究的数据集及相应的分类器参数。

数据集	**PokerDVS** [24]	**N-MNIST** [25]	**DVS128 Gesture** [26]
传感器尺寸（像素）	35×35	34×34	128×128
N_{labels}	4	10	11
样本分布（训练/测试）	48/12	60,000/10,000	1078/264
原始平均样本大小	17 毫秒	310 毫秒	6.5 秒
使用的缩短样本大小	10 毫秒	10 毫秒	1 s
批次大小	8	128	128
LIF 阈值	5	25	15

输入数据 参考类似分类器的现有示例，Tonic 框架提供的初步事件变换 [8] 会应用于输入事件，不受是否进行预处理的影响。每个样本均经过降噪处理，其事件会以 1 毫秒为间隔累积成帧。样本会缩短至表 3 所示的较短时间长度。此类预处理后的事件能够带来令人满意的分类性能，同时显著减少本研究中的计算时间。

评估指标 如上所述，本研究旨在评估基于线的预处理对神经形态计算机视觉的能效和性能所带来的益处。我们根据三个主要指标比较不同策略：分类准确率 A、突触事件数量 SE 以及推理时的理论效率 E。

准确率是指在完成 10 个周期的训练阶段后，分类器所进行的推理中正确预测的次数占预测总次数的比例。

基于现有研究 [34], ，可以通过推理过程中在脉冲神经网络内部传播的突触事件数量 SE 来近似其理论能耗。事实上，许多研究表明，在大多数最先进的神经形态架构 [6]中，脉冲神经网络模型消耗的能量与突触事件数量成正比。在应用预处理的模型架构中，突触事件数量 SE 的计算如下：

$$SE = SE_{\mathcal{P}}(k) + SE_{\mathcal{C}} \quad \text{with} \quad SE_{\mathcal{C}} = \mathcal{E}_{\mathcal{P}} \times n_{\mathcal{C}} \tag{1}$$

$\mathcal{P}$ 分别表示 $\mathcal{C}$ 预处理和分类器步骤，其中 $\mathcal{E}_{\mathcal{P}}$ 表示每个经过预处理的样本的平均事件数量。 $n_{\mathcal{C}}$ 表示分类器中的输出神经元数量，即需要分类的标签数量 N_{labels} （见表 3）。 $SE_{\mathcal{P}}$（k）表示输入数据集中每个像素的平均像素激活与每个像素的传出突触数量逐元素相乘后所得的总和。除通过实验获得外，很难以其他方式确定该值，因为它会根据输入数据大小、所采用的预处理策略以及所选参数 k （即直线检测中所涉及对角线之间的步长）而大幅变化。

其次，“无预处理”情形（即直接对事件数据进行分类）下的突触事件数量计算如下，其中 $\mathcal{E}$ 对应输入事件数据集中每个样本的平均事件数：

$$SE = SE_{\mathcal{C}} = \mathcal{E} \times n_{\mathcal{C}} \tag{2}$$

Fin最后，我们将脉冲神经网络的效率 E 定义为推理准确率与 e
对应 整个网络中相应的突触事件数量：

$$\mathbb{E} = \frac{\mathbb{A}}{SE} \tag{3}$$

需要注意的是，我们只考虑准确率达到令人满意的 A_E 的模型的效率：事实上，某个模型可能因脉冲活动微乎其微而表现极差、无法识别任何内容，却仍展现出令人印象深刻的效率。为此，我们任意设定了一个阈值 A_T，低于该阈值的效率将不予考虑：

$$\mathbb{A}_{\mathbb{E}} \geq \mathbb{A}_{\mathrm{T}} \quad \text{where} \quad \mathbb{A}_{\mathrm{T}} = \frac{2}{3} \times \mathbb{A}_{\text{no preprocessing}} \tag{4}$$

准确率和效率越高，整体性能就越好。本研究的目标是在神经形态基准数据集上实现显著更高的推理效率，同时保持令人满意的准确率。

3 模型架构分析

本节根据所采用的策略（见图 3a中六种可能的方法），从神经元和突触数量的角度，研究上述方案中涉及的脉冲神经网络模型的复杂度。我们评估了传感器尺寸不断增大时各项变量的变化，其取值接近当前最先进水平中常见的数值：N-MNIST [25] 或 PokerDVS [24] 等常见基准数据集的记录所用相机尺寸接近 30×30 像素；而顾名思义，用于创建 DVS128 Gesture [26]和 Prophesee 百万像素汽车检测数据集 [35] 的相机尺寸分别达到 128×128和1000×1000 像素。

大多数神经形态平台（如 SpiNNaker [23] 或 Loihi [36]）都将输入层实现为简单的“编码神经元”，其中每个神经元对应一个像素（针对视觉输入数据），并根据输入的地址事件表示数据直接发放脉冲。这些编码神经元不需要具备漏积分发放神经元的复杂动力学特性，因此不计入模型所涉及的神经元总数。图 4 （左）展示了整个分类流水线中涉及的神经元数量，包括或不包括基于线的预处理，也就是说，将预处理层中的神经元数量与输出层中的神经元数量相加。因此，“无预处理”方法中的神经元数量显然较少。然而，大多数神经形态平台支持神经元计算的时分复用 [37, 38, 39]：无需显著增加硬件资源或能耗，就能模拟大量神经元。

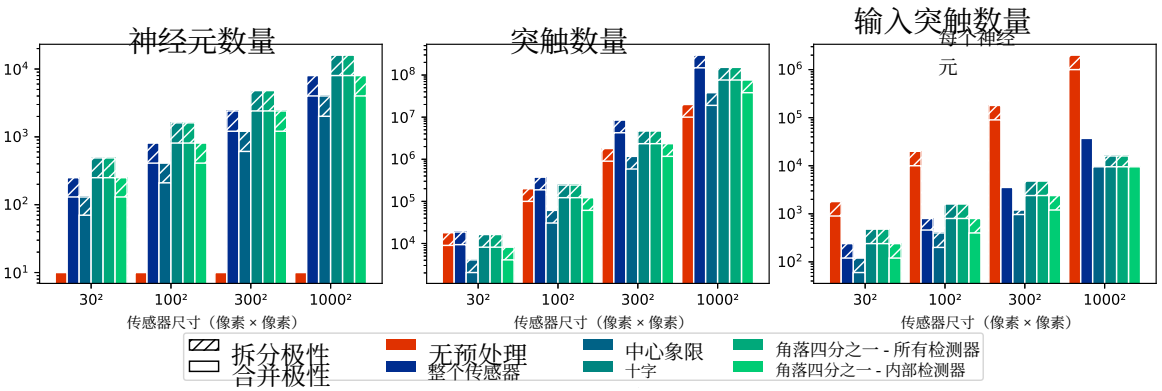


图 4：取决于传感器尺寸的模型架构。该模型的特征由所涉及的神经元数量（左图）和突触数量（中图），以及每个神经元的输入突触数量（右图）决定。突触数量按步长 $k = 30$ 计算。合并极性（无阴影）和拆分极性（对角线阴影）方法的柱状图相互重叠：拆分极性所得到的数值始终等于或高于合并极性。

硬件实现的主要限制通常在于突触数量。如图4 （中）所示，不同方法的突触总数差异小于神经元数量的差异。由于时分复用也适用于神经元之间的突触连接，因此，与模型中涉及的突触总数相比，每个神经元的输入突触数量更值得关注。事实上，连接到脉冲神经元的突触的硬件实现往往受可用资源限制；例如，SpiNNaker 板卡上每个神经元的输入突触数量上限为 1，000 个 [23]。根据图 4 （右），将输入传感器与输出层完全连接的基线方法需要大量输入突触，且其数量随传感器尺寸呈二次增长。添加中间预处理层（例如基于线的预处理方法中的做法），可将输入突触数量大幅减少多个数量级。

正如预期，“十字形”和“角落四分区—所有检测器”策略的神经元数量和突触数量相同，因为两者使用的检测器数量相同，且尺寸也相同。架构更精简的是“中心四分之一”策略，它只需要四个（半长度）检测器。由于该策略所需的突触数量较少，可以预期它的效率最高；然而，只在传感器中央四分区提取特征，可能会导致分类准确率较低。

4 玩具数据集上的概念验证

我们首先将不同的基于线的预处理方法应用于 PokerDVS [24]，，这是一个原生神经形态数据集，用于扑克牌符号的视觉分类。它包含 58 个样本，每个样本对应四种可能花色之一：“红桃”“黑桃”“方块”和“梅花”。该数据集于 2013 年使用一台128×128 DVS 相机记录，主要用于验证 DVS 的高帧率记录能力。如表3所示，每个花色样本均从原始记录中提取，并将相应样本的空间尺寸调整为 35×35 像素。

4.1 超参数微调

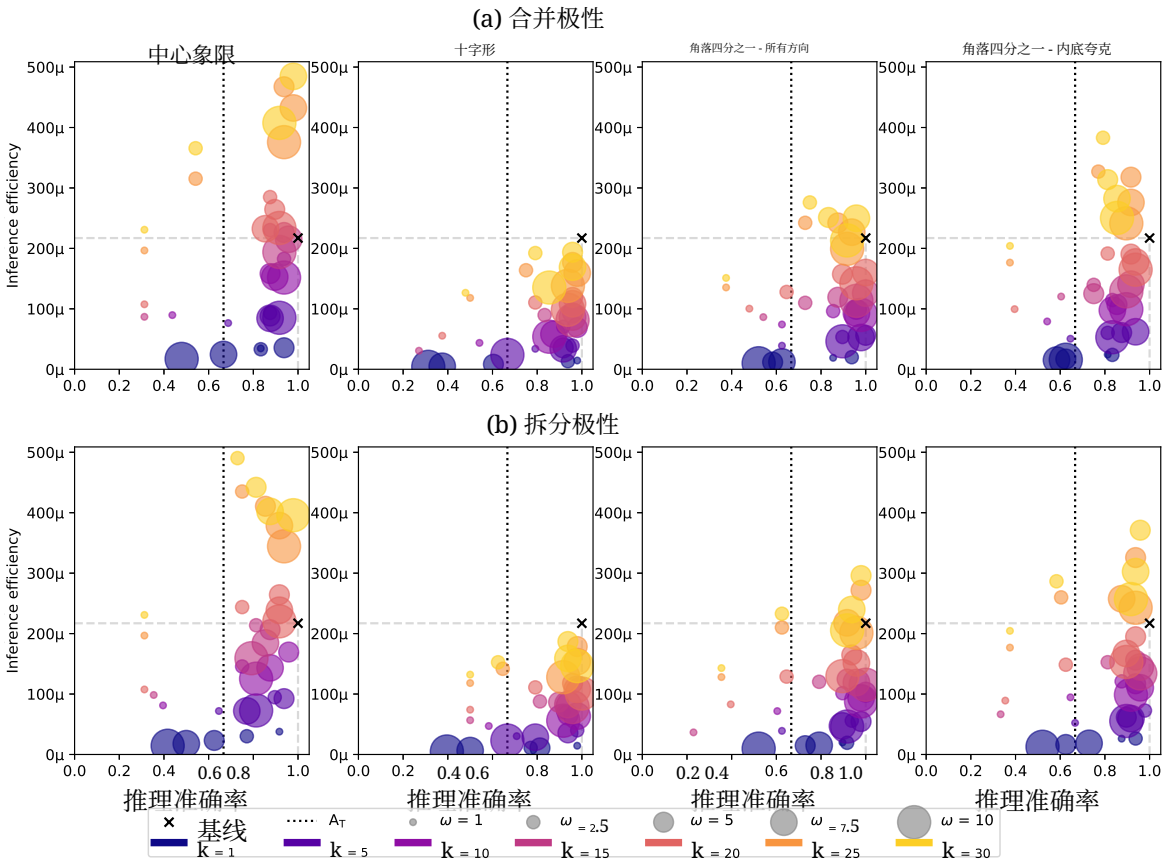


图 5：根据对 PokerDVS 应用不同基于线的预处理策略时，不同超参数集合（k、ω）下推理效率 E（纵轴）与推理准确率 A（横轴）之间的权衡。（a）和（b）分别对应合并极性和拆分极性，共用一个图例。“无预处理”的权衡由黑色十字形表示。超参数 k 的变化通过颜色变化表示，而每个点的大小则随强度 ω 增大而增大。黑色虚线表示 A_T 阈值。

我们选择了这个简单的数据集；通过 采用时间反向传播进行训练，经过 10 个周期后，神经形态分类很容易达到 100% 的准确率。我们将此作为验证基于行的预处理概念的初步依据。第 2.3 节所述超参数微调阶段的结果见图 5：图中展示了各种策略在推理准确率与推理效率之间的权衡，分别将极性合并（图5a）或分开处理（图 5b）。我们的目标是在保持准确率的同时提高推理效率。换言之，我们希望取得高于水平灰色虚线的权衡结果（该线表示基线

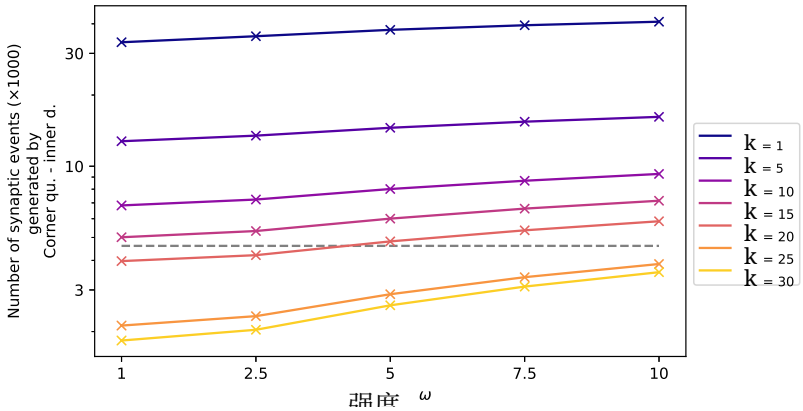


图 6：对 PokerDVS 应用拆分极性的“角落四分区—内部检测器”策略时，整个流水线中产生的突触事件数量。颜色表示步长 k 的变化，横轴表示强度 ω 逐渐增大。黑色虚线表示采用“无预处理”方法时产生的基线突触事件数量。请注意，纵轴采用对数刻度。

“无预处理”方法所达到的效率），并尽可能靠近图表右侧，也就是基线所达到的完美分类准确率。

根据图 5，对于大多数策略，许多超参数组都能达到目标的右上角：我们可以特别指出，“中心四分之一”策略如预期般实现了较高的效率，同时，“角落四分区—内部检测器”策略中的多个超参数组也实现了稳定的准确率，并兼具显著较高的效率。我们还可以观察到，较高的 k 值能够在保持良好准确率的同时提升效率。图 6所示的进一步研究证实了这一点，该研究展示了对 PokerDVS 应用拆分极性的“角落四分区—内部检测器”策略时，整个流水线中的突触事件数量。如前所料，连接输入与基于线条的预处理层的突触数量直接取决于步长 k，并显著影响整个流水线（预处理步骤与分类器合计）中的突触事件数量：值 k越大，突触数量越少，突触事件数量也越少。当 k 大于 20 时，突触事件数量及相应的理论能耗均低于基线方法，尽管增加了一个额外层。

值得注意的是，较低的 k 距离会同时损害效率和准确率。我们最初假设情况恰恰相反，因为较小的 k 会使模型检测到的线条之间的步长更小，从先验角度看，这能为分类保留更精细的细节（见图3）。应当在比 PokerDVS 分类更复杂的情境中对这一反直觉的观察进行比较；总的来说，PokerDVS 是一项相当容易解决的任务。

4.2 与类似方法的比较

如前所述，本文提出的基于线条的预处理方法与全文采用的“无预处理”方法相对。不过，神经形态视觉中常用的其他网络架构和方法，也可视为能够进行特征提取的中间预处理。图7 中的结果有助于我们评估，将本文方法与两种此类技术进行比较的科学价值：卷积，以及使用带剪枝和不带剪枝的中间隐藏层。

卷积和剪枝方法均采用第 2.3 节所述的分类器参数化方式实现，并分别在合并和拆分事件极性的情况下计算结果。卷积方法首先包含一个二维卷积层，将输入事件传送至由 n_i 个漏积分发放神经元组成的中间隐藏层，该层与输出层全连接。卷积的步幅为 1，不进行填充，并通过计算卷积核大小，使神经元数量 n_i 与基于线的预处理方法中预处理层的神经元数量相等。权重在整个网络中学习，而偏置仅在第二个全连接层中学习。

同样，全连接方法（在本文后文中称为“全连接层”）由一个包含 n_i 个漏积分发放神经元的隐藏层组成，该层通过两个具有学习得到的权重和偏置的全连接层接收并发出脉冲，途经。随后使用这些预训练参数来

获得原样的推理性能（图 7中的“未剪枝的全连接层”），以及经过剪枝阶段后的推理性能（图 7中的“剪枝后的全连接层”）。后者的实现方式是将绝对值低于特定阈值的预训练突触权重设为零。剪枝阈值经实验设定为 ± 0.01。

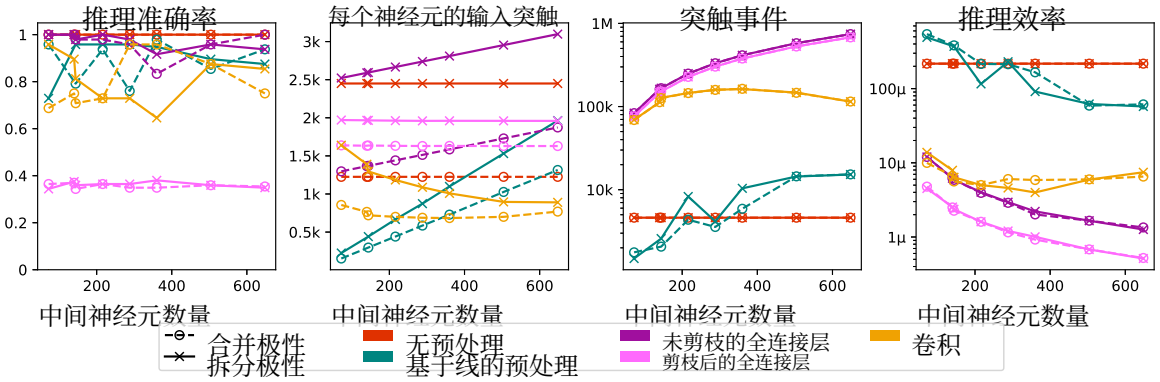


图 7：比较“无预处理”（红线）与各种基于线的预处理策略（青绿色线）的性能，并与卷积方法（黄线）和全连接方法（“全连接层”）进行比较；第一层剪枝前（紫线）或剪枝后（粉线）。最后三种方法使用一个中间隐藏层，其神经元数量（横轴）根据本文介绍的五种基于线的预处理策略中预处理层涉及的神经元数量而变化。

图 7 展示了推理准确率和效率的变化，以及每个神经元的输入突触数量（关于时分复用的进一步讨论，请参见第 3 节）和各网络发出的突触事件数量，依据 n_i。为了实现降低理论能耗的目标，最佳情况下应兼具较高的准确率和效率，以及较少的每个神经元输入突触数量和突触事件数量。卷积和“未剪枝的全连接层”都达到了与基于行的预处理方法相近或更高的准确率，接近“无预处理”所达到的理想性能。然而，在全连接层中加入剪枝会显著降低推理准确率，使其接近随机猜测水平；这似乎表明，即使剪枝前后的突触事件数量相近（见图 7，第三幅图），全连接层的分类性能仍容易受到低影响连接所检测到的细节影响。我们本可以采用更深层网络评估全连接层方法，但根据我们的结果很容易预测，增加网络深度会提高突触活动，从而按比例降低能效。

聚焦于高性能方法：随着 n_i 中间神经元数量增加，“未剪枝的全连接层”的表现与基于线的预处理相似，每个神经元的输入突触数量不断增加，但始终更多；而卷积则呈现相反的趋势，其输入突触数量与之成反比 n_i。输入突触数量减少有利于提升硬件实现效率，但突触活动是衡量理论能耗更重要的指标。如图7的第三幅图所示，基于线的预处理所产生的突触事件数量明显更少，接近“无预处理”的水平，且比全连接层和卷积得到的数值最多低两个数量级。综合这些不同的观察结果，我们得出结论——图7的第四幅图验证了这一结论：基于线的预处理是唯一能够达到与“无预处理”这一基线相近或更高推理效率的方法。卷积和全连接层方法的推理效率均比基线低一到两个数量级；不过，卷积的推理效率有趣地保持在相近水平，因此在 n_i变化时，能够恰当地平衡准确率与突触活动。

由于卷积和深度全连接网络在推理效率方面都显著较低，而本研究采用的指标旨在评估在分类 PokerDVS 数据集这一简单任务上的理论能耗，因此，尽管这两种方法已展现出在神经形态视觉领域的潜力，我们仍决定在本文余下部分中舍弃它们。

5 基准数据集上的实验性能

上一节介绍了我们在 PokerDVS 上取得的初步结果。我们在一个玩具数据集上验证了基于线条的预处理方法，该方法可降低理论能效，同时支持对原生神经形态数据进行简单分类。接下来，本节将评估我们的方法在另外两个更复杂的基准数据集上的效果；这两个数据集同样是原生神经形态数据，并广泛用于视觉分类。

5.1 大规模数据集

我们首先将研究扩展到神经形态 MNIST（N-MNIST）数据集 [25]。该数据集通过向一台尺寸为 34×34 像素的 DVS 相机展示 0 到 9 的手写数字进行记录，相机按预设的扫视轨迹移动。数据集包含超过 70,000 个样本，均匀分布于 10 个类别。创建该数据集旨在展示其较小分辨率带来的优势，即“在对新想法进行原型设计时，能够快速测试和迭代算法” [25]。图 3c 展示了将一种基于线条的预处理策略（即“角落四分区——内部检测器”）应用于一个 N-MNIST 类别后得到的输出。

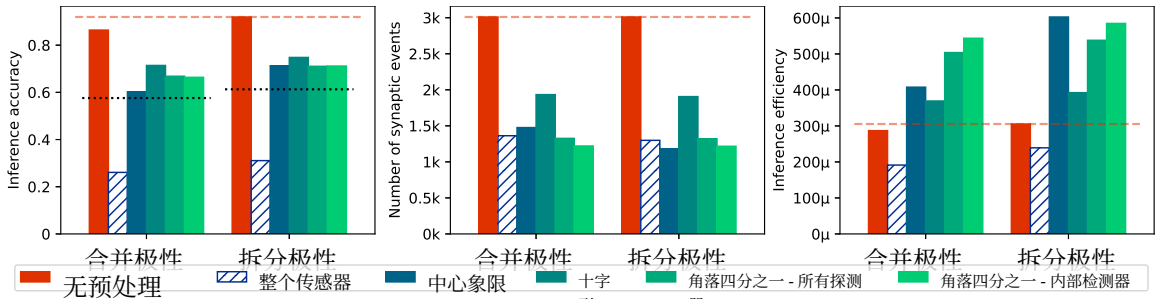


图 8：比较 N-MNIST 上五种线预处理方法与“无预处理”方法的表现，包括推理准确率（左）、推理时的突触事件活动（中）和推理活动（右），并按极性处理方式（合并或拆分）区分。基于线的预处理策略的结果对应“最佳效率参数化”。红色虚线表示“无预处理”达到的最高值；黑色点线（第一幅图中）表示$\mathbb{A}_T$ 阈值。“整个传感器”策略以阴影标出，因为它不满足式 4中详述的效率条件，即其准确率低于“无预处理”准确率的三分之二。

图 8 展示了本研究中引入的各种基于线的预处理策略在应用与不应用情况下，对 N-MNIST 分类的推理准确率和效率，以及整个网络产生的突触事件数量。结果采用“最佳效率参数化”获得。预处理对 N-MNIST 分类性能的影响大于对 PokerDVS 的影响：“十字形”策略结合拆分极性，在本研究考察的五种策略（采用“最佳效率参数化”）中取得了最高的分类准确率，但其数值仅为 0.748，即“无预处理”准确率的 81.4%。尽管如此，它仅用 1905 个突触事件就达到了这一结果，即所需事件数仅为“无预处理”的三分之二，尽管用于预处理的网络多了一个中间层。除“整个传感器”外，所有预处理策略都呈现出类似的表现：其推理效率均超过“无预处理”，其中“中央四分区”策略结合拆分极性，使这一基线值翻倍。

尽管“整个传感器”的数值不符合式中详述的效率条件，但此处仍会显示该数值（尽管已被遮挡）。4，也就是说，其准确率低于“无预处理”策略准确率的三分之二。其准确率较低，同时突触事件数量也有所减少；然而，两者并未相互抵消，无法实现足够好的权衡。将该策略应用于 N-MNIST 时，只有它在推理效率方面未能超越“无预处理”策略。值得注意的是，其准确率较低并不能单纯归因于突触事件数量少，更可能是因为与突触模式相关的空间密度不足，因为“中央四分区”和两种“角落象限”策略的突触活动相近或更低，准确率却高出一倍。这凸显了在事件数据预处理中采用相关突触模式以降低能耗的必要性，因为这种表现也会对效率产生积极影响。

总而言之，此前在简单数据集上评估的预处理方法，能够有效降低大规模神经形态分类的理论能耗。

5.2 现实场景

此前评估的两个神经形态数据集具有一个共同的显著特征：事件由在均匀背景上移动的静态图像生成（PokerDVS 中代表扑克牌点数的图像 [24] 以及 N-MNIST 中的手写数字 [25]）。尽管这些数据集原生属于神经形态数据集，但它们并未充分展现使用事件相机检测细微动作的附加价值。下一节将研究基于线的预处理在一个避免这一缺陷、并专门评估时间信息的神经形态数据集上的应用：DVS128 手势数据集 [26]。该数据集针对手势识别，由 130 多个时长为 6 秒的样本组成，这些样本代表了在不同光照条件下约 30 名人类受试者的动作。受试者在 DVS128 相机前移动双手，以完成 11 种不同的动作。由于 N-MNIST 和 PokerDVS 主要包含空间信息，因此分类仅需每个样本中的 10 ms（见表 3）；然而，分类器至少需要每个样本 1 秒的时间来积累足够的时间信息，从而根据 DVS128 Gesture 识别手势。

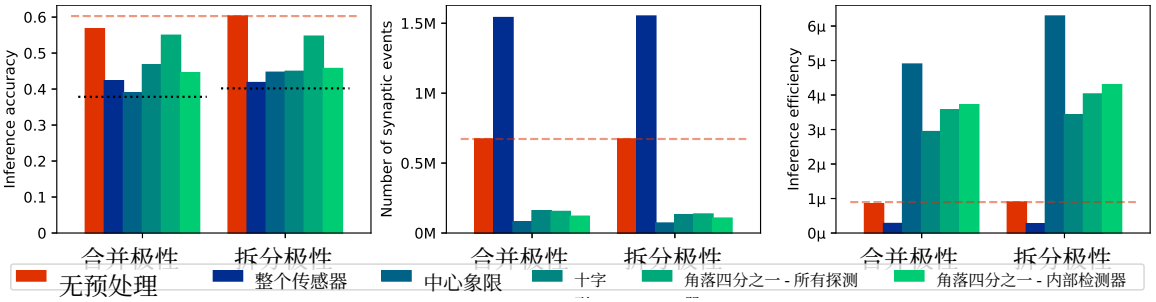


图 9：在 DVS128 Gesture 上，比较五种基于线的预处理方法与“无预处理”在推理准确率（左）、推理时的突触活动（中）和推理效率（右）方面的性能，并根据极性处理方式进行区分。基于线的预处理策略的结果对应于“最佳效率参数化”。红色虚线表示“无预处理”达到的较高值；黑色点线（在第一幅图中）表示 A_T 阈值。

图 9 展示了采用或不采用五种基于线的预处理策略（使用“最佳效率参数化”）时，DVS128 Gesture 分类的推理准确率和效率，以及相应的突触事件数量。与 N-MNIST 类似，“整个传感器”方法难以实现适合低能耗的最佳权衡：其准确率仅达到“无预处理”准确率的 70%，而相应的突触活动几乎增加至三倍，导致效率较低。然而，其他四种策略在理论能耗方面均显著优于“无预处理”：其推理效率至少高出 3 倍（“合并极性”的“十字形”），最高则超过 6 倍（“拆分极性”的“中央四分区”）。这可以通过其生成的突触事件数量极低来解释：所需突触活动仅为“无预处理”的 16.51%（拆分极性）至 19.05%（合并极性），同时分类准确率最高可达到基线分类准确率的 96%（采用合并极性的“角落四分区——所有检测器”）。

由于 DVS128 Gesture 携带的信息更为复杂，我们基于线的预处理策略可能不够复杂，无法有效捕获分类所需的所有相关特征，因此性能不尽理想。这里我们介绍基于线的预处理的最后一种变体：五种累积策略，每种由两到三种策略组合而成（即将“中心四分区”与“十字形”及两个“角落象限”分别组合，以及将这三者全部累积），以改进特征提取。在本节余下部分，我们将使用简写名称：Ce 表示“中心”，Cr 表示“十字形”，CQ-AD 和 CQ-ID 分别表示“中心四分区——所有检测器”和“内部检测器”，累积则用一个符号表示 + 。初始策略和生成事件中的突触与神经元群体会被累加，然后输入分类器。图10 展示了采用“无预处理”（红色）、五种初始的基于线的预处理策略和五种累积策略时，训练和推理阶段在“最佳准确率参数化”下准确率（横轴）与效率（纵轴）之间的权衡结果。“无预处理”构成了我们力求超越的基线：无论是超越其准确率（红色竖直虚线）还是其效率（红色水平虚线）均可，而图表右上角代表同时超越两者的理想情形。

根据图 10，得益于累积策略，这一理想情形在推理时确实得以实现：Cr+Ce、CQ-AD+Cr+Ce 和 CQ-ID+Cr+Ce 这三种方法在合并极性后，均达到了

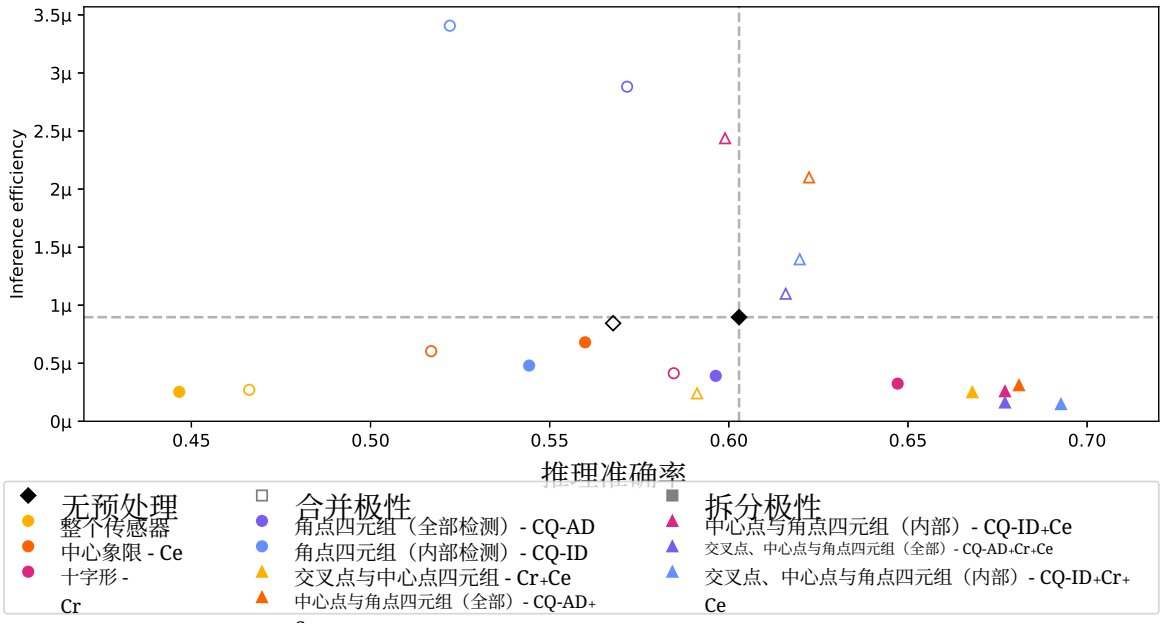


图 10：在推理阶段采用“最佳准确率参数化”时，各种方法在 DVS128 Gesture 上效率（纵轴）与准确率（横轴）之间权衡的性能比较（与图 9 中展示的“最佳效率参数化”相对）。图中展示了“无预处理”以及本研究提出的五种基于线的预处理方法；还展示了五种累积方法，即对最初的基于线的预处理策略进行两种或三种组合。合并极性的结果以空心标记表示，拆分极性的结果以实心标记表示。黑色虚线分别表示“无预处理”所达到的较高准确率值和效率值。所有策略均满足 $\mathbb{A}_{\mathrm{T}}$ 条件，即准确率高于原始准确率的三分之二。

比“无预处理”更高的准确率，同时效率也有所提高——Cr+Ce 的效率超过基线的两倍。此外，在纯分类准确率方面，多种方法的表现优于“无预处理”设定的基线：大多数累积方法都是如此，采用拆分极性的“十字形”方法也值得一提，其准确率比基线高出 5 个百分点。最后，值得注意的是，与“无预处理”相比，采用合并极性或拆分极性的累积策略往往分别能带来更高的效率或更高的准确率。正如前文所述，这可以解释为：分别考虑不同极性能够保留更多时间信息和更精细的细节，从而实现更精确的分类和更高的准确率，尤其是在针对后者进行优化的“最佳准确率参数化”下。相反，合并极性会造成信息丢失并降低分类准确率——在 DVS128 Gesture 中尤为明显，因为手势方向可以部分地根据事件极性的空间分布推断出来——但同时会减少突触事件数量，从而提高效率。

总而言之，最后这些实验结果表明，我们的基于线的预处理能够更高效地对真实场景中的复杂视觉数据进行神经形态分类，并且在特定情况下还能取得更好的性能。

6 讨论

正如本文引言所述，我们旨在提高传统神经形态计算机视觉任务的能效；在实现这一目标的同时，还要保持令人满意的性能。这意味着，提升性能只是次要的益处，并非必须实现的目标。为此，我们在三个数据集上评估了五种主要的基于线的预处理策略：PokerDVS [24]，N-MNIST [25] 和 DVS128 Gesture [26]。此外，我们还对后者组合应用了多种策略，以捕捉这个更复杂数据集中的更精细细节。

表 4 总结了本研究的主要结果，重点展示了针对三个基于事件的数据集进行分类时，在优化分类准确率或理论能效的情况下所采用的最佳策略。根据表4，“最佳准确率参数化”使 PokerDVS 和 N-MNIST 的准确率达到令人满意的水平，接近“无预处理”所设定的基线。对于

DVS128 Gesture，通过累积采用多种策略，准确率比无预处理的基线提高了 6 到 9 个百分点。此外，“最佳效率参数化”始终能提升效率，同时保持令人满意的准确率（至少达到基线准确率的四分之三，高于 $\mathrm{A_T}$ 在公式 4中设定的三分之二下限）。PokerDVS 的效率提高到原来的 2.5 倍，N-MNIST 的效率提高到原来的 2 倍，而最突出的是，DVS128 Gesture 的效率提高到原来的 7 倍。最后，将累积策略应用于最后一种情形，也实现了显著提升准确率和效率的折中方案。

表 4：在三个基于事件的数据集 PokerDVS、N-MNIST 和 DVS128 Gesture 上，无预处理基线和基于线的预处理策略（分别实现最佳准确率和最佳效率）的分类性能，包括准确率和效率。粗体数值表示各数据集取得的最佳准确率和最佳效率。

数据集	参数化	策略	极性	k	ω	准确率 A	效率 E
PokerDVS	无预处理		任一	/	/	**1.0**	2.171×10^{-4}
	最佳准确率	CQ-AD	合并	20	7.5	**1.0**	1.601×10^{-4}
	最佳效率	Ce	合并	30	5	0.958	$\mathbf{5.43 \times 10^{-4}}$
N-MNIST	无预处理		拆分	/	/	**0.919**	3.054×10^{-4}
	最佳准确率	Cr	拆分	30	10	0.901	1.935×10^{-4}
	最佳效率	Ce	拆分	30	7.5	0.713	$\mathbf{6.027 \times 10^{-4}}$
DVS128 手势	无预处理		拆分	/	/	0.603	8.966×10^{-7}
	最佳准确率	CQ-ID+Cr+Ce	拆分	1	7.5	**0.693**	1.479×10^{-7}
	最佳效率	Ce	拆分	25	10	0.447	$\mathbf{62.89 \times 10^{-7}}$
	综合表现最佳	CQ-AD+Ce	合并	15	7.5	**0.622**	$\mathbf{21.00 \times 10^{-7}}$

Note: Ce stands for the “central quarter” strategy, Cr for “cross”, CQ-AD for “corner quarters - all detectors” and CQ-ID for “corner quarters - inner detectors”. A symbol + denotes the cumulation of multiple strategies together.

感谢表 4，由此我们证实了最初的假设：在神经形态分类中应用相关的事件预处理，可提升理论能效。我们在三个难度、数据特性和规模各不相同的数据集上进行了验证。此外，我们还证明，复杂场景可通过基于线的预处理同时提升准确率和效率，这一点从 DVS128 Gesture 上的“综合表现最佳”结果（表 4最后一行）中可见。

值得注意的是，“整个传感器”方法的表现始终较差。这可以从多个参数来解释：线检测机制只有在线条穿过受监测传感器区域的边界时，才会检测到这些线条并输出相应的事件。大多数用于分类的基于事件的数据集，尤其是本研究选用的数据集，都包含集中分布的信息，即重要事件分布在传感器的中心。顾名思义，“整个传感器”方法涵盖整个传感器，因此，集中分布的事件数据中只有少数线条会穿过其受监测区域的边界。因此，该方法需要较高的激活（即较高的强度 ω）才能产生事件，而且这些事件还必须数量充足，才能携带相关信息。因此，“整个传感器”方法要么：

- 不产生事件，例如 PokerDVS；
- 准确率不够高，无法纳入本研究，例如 N- MNIST（见图 8）——提醒一下，我们只研究准确率至少达到基线准确率三分之二的策略的效率；
- 通过输出数量极其庞大的突触事件来达到令人满意的性能，从而导致效率显著降低，DVS128 Gesture 就是这种情况（见图 9）。

此外，“整个传感器”方法依赖于 4 个检测器，即至少比其他方法少一半（“中央四分区”方法除外）；检测器数量过少，无法捕捉足够精细的细节，因而也无法实现足够好的分类。“中央四分区”通过监测传感器中较小的区域克服了这一问题，在该区域中，线检测模型更容易感知相关信息。

从图8 （右）和图9 （右）中均可观察到，对于 N-MNIST 和 DVS128Gesture，拆分极性的策略整体表现优于相应的合并极性策略；这可以解释为，极性所提供的相关信息有所损失，从而抵消了

网络中突触连接数量减少所带来的优势（见图4）。在 PokerDVS 上，这种现象不太明显，因为该数据集较为简单，无论极性合并还是拆分，表现都相近。图5 展示了每种预处理策略的相似趋势：合并极性（上排）和拆分极性（下排）。

最后，本研究的目的并非提高分类准确率，而是验证事件预处理所带来的能耗改善。尽管如此，我们确实观察到，得益于事件预处理，DVS128 Gesture 上的分类准确率显著提高（见表4）。这些结果仍远不及近期文献中的成果：该数据集上最先进的神经形态分类方法准确率已超过 95% [40, 41, 42]。然而，后者采用的架构比本研究中刻意选用的简单架构复杂得多；因此，这些观察结果表明，预处理有望带来更出色的分类性能。

更值得注意的是，在 DVS128 Gesture 上观察到的“整体最佳”表现（见表 4，最后一行），体现了分类性能与能耗之间的成功权衡，同时提升了准确率和效率。如今，我们已在刻意采用简单脉冲神经网络分类器的背景下，证明了基于行的事件预处理对能效的积极影响；接下来便可将其应用于更复杂的模型，以重现这种“整体最佳”表现，并同步优化能耗与性能。

7 结论

据作者所知，本研究首次提出了一种端到端的神经形态线式预处理机制，并将其应用于事件数据。我们评估了该方法应用于多个基于事件的数据集的神经形态分类时，对理论能效产生的积极影响；这些数据集在难度、数据属性和规模方面各不相同。

我们展示了在执行神经形态计算机视觉任务之前对事件数据进行预处理的多项优势。我们重点介绍以下方面的改进：

- 能效：对于经过调优的一组超参数（k，ω），预处理可显著且稳定地减少突触事件数量，同时保持分类准确率；在复杂分类任务中，能效最高可达到基线的 7 倍；

- 内存使用量：基于线的预处理稳定地降低了每个神经元所需的输入突触数量。对于神经形态架构的硬件实现而言，这是主要挑战之一；相比之下，得益于时分复用，神经元和突触的数量本身并非主要挑战；

- 性能：在 DVS128 Gesture 所展示的复杂真实场景中应用基于线的预处理，可使基线准确率提高 9 个百分点。

最后，我们通过累积多种策略，使更复杂的场景同时优化了性能和理论能耗通过。我们的结果为事件预处理奠定了基础，并凸显了它对于实现高效、低能耗神经形态计算机视觉的必要性。与采用深度人工神经网络和标准相机的传统计算机视觉类似，在进行任何后续处理之前，确实必须先对输入数据进行预处理。此外，预处理必须适合所考虑的数据和所执行的任务，才能对降低能耗产生积极影响。

未来的工作包括进一步探究本文主体部分所呈现的结果，并将研究拓展至更多计算机视觉任务和更大规模的数据，例如在 Prophesee 1 Megapixel 数据集 [43]上进行分割。我们希望在更复杂和/或更贴近实际的任务中，进一步探索在 DVS128 Gesture 上观察到的准确率与效率同步优化现象。此外，我们还希望更深入地研究其他神经元和突触参数对预处理的影响。还可以采用更具生物启发性且更适合硬件实现的学习策略来评估分类性能，并更加重视时间维度，例如 [44]。本研究以突触活动近似理论能耗，未来可借助更复杂的神经形态能耗模型（例如 [34, 45]）深入研究理论能耗。最后，鉴于上述结果是使用 CPU 模拟器 [29]，获得的，接下来应将这些预处理机制部署在 SpiNNaker [23]等神经形态硬件或 FPGA 上，以评估其在实时场景中的应用及实际能耗，从而验证这一颇具前景的理论能耗收益。

致谢
本研究获得了一项公共资助，该资助由法国国家研究署（ANR）监管，属于“PEPR IA France 2030”计划的一部分（Emergences项目 ANR-23-PEIA-0002）。

作者贡献
作者 Am´elie Gruel、Pierre Lewden、Adrien F. Vincent 和 Sylvain Sa¨ıghi 参与了本研究的概念构思、方法设计和正式分析。Am´elie Gruel 负责项目协调、数据整理、调查研究、软件实施、验证和可视化。Sylvain Sa¨ıghi 和 Adrien F. Vincent 负责经费筹措。Am´elie Gruel 撰写了论文初稿；Pierre Lewden、Adrien F. Vincent 和 Sylvain Sa¨ıghi 通过审阅和编辑参与完成第二稿。所有作者均阅读并批准了最终稿。

参考文献
[1] W. Maass，“脉冲神经元网络：神经网络模型的第三代”，《*Neural Networks*》，第10卷，第9期，第1659–1671页，1997年。

[2] P. Lichtsteiner、C. Posch 和 T. Delbruck，“一款 128×128、120 dB、15 µs 延迟的异步时域对比视觉传感器”，《*IEEE*固态电路期刊》，2008年。

[3] G. Gallego、T. Delbruck、G. Orchard 等，“事件视觉：综述”，《*IEEE*模式分析与机器智能汇刊》，2020年。

[4] D. V. Christensen、R. Dittmann、B. Linares-Barranco、等，“2022年神经形态计算与工程路线图”，神经形态计算与工程，第2卷，第2期，2022年。

[5] T. Finateu, A.Niwa, D.Matolin, 等,“5.10 一种 1280× 720 背照式堆叠时域对比事件视觉传感器，配备 4.86 μm 像素、十亿事件每秒读出、可编程事件速率控制器和压缩数据格式化流水线”，载于*2020 IEEE* 国际 *l*

固态电路会议*-(ISSCC)*，第112–114页，IEEE，2020。[6] M. Sorbaro、L.-Y. D. Liu、M. Bortone 和 S. Sheik，“优化……的能耗 f

用于神经形态应用的脉冲神经网络，” *Front. in Neuroscience*，第14卷，2019年。[7] M. Patel，“Python图像预处理技术完整指南”，2023年10月。[8]G. Lenz、K. Chaney、S. B. Shrestha、O. Oubari、S. Picaud和G. Zarrella，“Tonic：基于事件的数据集与变换”，2021年7月。文档可在 https://tonic.readthedocs.io 查阅。[9] A. Gruel、J. Martinet、T. Serrano-Gotarredona和B. Linares-Barranco，“用于嵌入式计算机视觉的事件数据下采样”，载于*VISAPP*，2022年。[10] A. Gruel、J. Martinet、B. Linares-Barranco和T. Serrano-Gotarredona，“使用脉冲神经网络比较DVS数据空间下采样方法的性能”，载于*2023 IEEE/CVF*冬季计算机视觉应用会议（*WACV*），第6483–6491页，2023年。[11] A. Gruel、D. Hareb、A. Grimaldi、J. Martinet、L. Perrinet、B. Linares-Barranco和T. Serrano-Gotarredona，“神经形态中央凹化的意义：嵌入式事件相机的美好前景”，*Biological Cybernetics*，第117卷，第1–18页，2023年9月。[12] S. Afshar、N. Ralph、Y. Xu、J. Tapson、A. v. Schaik和G. Cohen，“使用自适应选择阈值进行基于事件的特征提取”，*Sensors*，第20卷，第6期，2020年。[13] V. Vasco、A. Glover和C. Bartolozzi，“利用事件驱动相机优势的快速基于事件的Harris角点检测”，载于*2016*年*IEEE/RSJ*智能机器人与系统国际会议（*IROS*），第4144–4149页，2016年。[14] I. Alzugaray和M. Chli，“事件相机的实时异步角点检测与跟踪”，*IEEE*机器人与自动化快报，第3卷，第3177–3184页，2018年10月。[15] S.-H. Ieng、C. Posch和R. Benosman，“异步神经形态事件驱动图像滤波”，*Proceedings of the IEEE*，第102卷，2014年9月。

[16] C. Br¨andli、J. Strubel、S. Keller、D. Scaramuzza和T. Delbruck，“ELiSeD——一种基于事件的线段检测器”，载于2016年第二届基于事件的控制、通信与信号处理国际会议（EBCCSP），第1–7页，2016年。[17] G. Orchard、C. Meyer、R. Etienne-Cummings、C. Posch、N. Thakor和R. Benosman，“Hfirst：一种用于目标识别的时间方法”，IEEE模式分析与机器智能汇刊，第37卷，第10期，第2028–2040页，2015年。[18] T. N. Chandrapala和B. E. Shi，“从基于事件的刺激中提取不变特征”，载于第六届IEEE生物医学机器人与生物机电一体化国际会议，第1–6页，2016年。[19] X. Lagorce、G. Orchard、F. Galluppi、B. E. Shi和R. B. Benosman，“Hots：用于模式识别的基于事件的时间表面层次结构”，IEEE模式分析与机器智能汇刊，第39卷，第7期，第1346–1359页，2017年。[20] C. Blakemore和G. Cooper，“大脑发育取决于视觉环境”，《自然》，第228卷，第477–478页，1970年10月。[21] A. G´omez Egu´ıluz、J. Rodr´ıguez-G´omez、J. Mart´ınez-de Dios和A. Ollero，“用于无人航空系统接触时间机动的异步基于事件的线跟踪”，载于2020年IEEE/RSJ智能机器人与系统国际会议（IROS），第5978–5985页，2020年。[22] A. Gruel、A. F. Vincent、J. Martinet和S. Sa¨ıghi，“基于神经形态事件的SpiNNaker直线检测”，载于IEEE人工智能电路与系统国际会议（AICAS），2024年。[23] S. B. Furber等，“SpiNNaker系统架构概述”，IEEE计算机汇刊，2013年。[24] T. Serrano-Gotarredona和B. Linares-Barranco，“Poker-DVS和MNIST-DVS：其历史、制作过程及其他细节”，《神经科学前沿》，第9卷，2015年12月。[25] G. Orchard、A. Jayawant、G. K. Cohen和N. Thakor，“使用扫视将静态图像数据集转换为脉冲神经形态数据集”，《神经科学前沿》，第9卷，第437页，2015年。[26] A. Amir、B. Taba、D. Berg、T. Melano、J. McKinstry、C. Di Nolfo、T. Nayak、A. Andreopoulos、G. Garreau、M. Mendoza等，“一种低功耗、完全基于事件的手势识别系统”，载于CVPR，2017年。[27] H. Paugam-Moisy和S. M. Bohte，“使用脉冲神经元网络进行计算”，载于《自然计算手册》，施普林格出版社，2012年。[28] H. Li、H. Liu、X. Ji、G. Li和L. Shi，“CIFAR10-DVS：用于目标分类的事件流数据集”，《神经科学前沿》，第11卷，2017年。[29] M.-O. Gewaltig和M. Diesmann，“NEST（神经模拟工具）”，2007年。[30] A. P. Davison、D. Br¨uderle、J. Eppler、J. Kremkow、E. Muller、D. Pecevski、L. Perrinet和P. Yger，“PyNN：神经元网络模拟器的通用接口”，《神经信息学前沿》，第2卷，2009年。[31] D. P. Kingma和J. Ba，“Adam：一种随机优化方法”，2017年。[32] J. K. Eshraghian、M. Ward、E. Neftci、X. Wang、G. Lenz、G. Dwivedi、M. Bennamoun、D. S. Jeong和W. D. Lu，“借鉴深度学习经验训练脉冲神经网络”，Proceedings of the IEEE，第111卷，第9期，第1016–1054页，2023年。[33] W. Fang、Z. Yu、Y. Chen、T. Masquelier、T. Huang和Y. Tian，“引入可学习膜时间常数以增强脉冲神经网络的学习能力”，2021年IEEE/CVF计算机视觉国际会议（ICCV），第2641–2651页，2020年。[34] M. Dampfhoffer、T. Mesquida、A. Valentian和L. Anghel，“SNN真的比人工神经网络更节能吗？一项深入的硬件感知研究”，IEEE计算智能新兴主题汇刊，第7卷，第3期，第731–741页，2023年。[35] E. Perot、P. De Tournemire、D. Nitti、J. Masci和A. Sironi，“学习使用1百万像素事件相机检测目标”，《神经信息处理系统进展》，第33卷，第16639–16652页，2020年。

[36] M. Davies、N. Srinivasa、T.-H. Lin 等人，“Loihi：一种具备片上学习功能的神经形态众核处理器”，*IEEE Micro*，第 38 卷，第 1 期，第 82–99 页，2018 年。[37] K. Dugan、J. Reeve 和 A. Brown，“Spinlink：用于 SpiNNaker 仿生多计算机的互连系统”，2012 年 5 月。[38] V. Kornijcuk 和 D. S. Jeong，“实时可适应数字神经形态硬件的最新进展”，*Advanced Intelligent Systems*，第 1 卷，文章编号 1900030，2019 年 8 月。[39] D. Ma、X. Jin、S. Sun、Y. Li、X. Wu、Y. Hu、F. Yang、H. Tang、X. Zhu、P. Lin 和 G. Pan，“Darwin3：一种采用新颖指令集架构并具备片上学习功能的大规模神经形态芯片”，2023 年。[40] L. Shi、A. Li 和 B. Ward-Cherrier，“脉冲脑压缩：探索针对脉冲神经网络的一次性后训练剪枝与量化”，2025 年。[41] Y. Abdennadher、G. Perin、R. Mazzieri、J. Pegoraro 和 M. Rossi，“LightSNN：面向稀疏且准确的脉冲神经网络的轻量级架构搜索”，载于*2025 年国际高级机器学习与数据科学会议（AMLDS）*，第 84–89 页，2025 年。[42] P. G. Malettira、S. Negi、W. Ponghiran 和 K. Roy，“TSKips：通过脉冲神经网络中的显式时间延迟连接提升效率”，2025 年。[43] E. Perot、P. de Tournemire、D. Nitti、J. Masci 和 A. Sironi，“利用 100 万像素事件相机学习检测物体”，*CoRR*，第 abs/2009.13436 卷，2020 年。[44] P. Lewden、A. Vincent、J. Tomas 和 S. Saïghi，“利用混合模式脉冲神经网络和无源忆阻器阵列进行在线学习：神经形态相机的应用”，*IEEE Access*，第 PP 卷，第 1–1 页，2023 年 1 月。[45] E. Lemaire、L. Cordone、A. Castagnetti、P.-E. Novac、J. Courtois 和 B. Miramond，“脉冲神经网络能效的解析估算”，载于神经信息处理：第 *29* 届国际会议（*ICONIP 2022*），线上会议，*2022* 年 *11* 月 *22–26* 日，论文集，第 *I* 部分，（柏林、海德堡），第 574–587 页，施普林格出版社，2022 年。