# EnforceSNN：面向嵌入式系统、考虑近似DRAM的具备韧性且能源高效的脉冲神经网络推理

**Rachmad Vidya Wicaksana Putra** [1,*]、**Muhammad AbdullahHanif** [2] **和 Muhammad Shafique** [2]

[1] 嵌入式计算系统（*ECS*），计算机工程研究所，维也纳工业大学（*TU Wien*），维也纳，奥地利 ¨*eBrain*实验室，工程系，纽约大学阿布扎比分校（*NYUAD*），阿布扎比，阿拉伯联合酋长国[2]

通讯作者*：Rachmad Vidya Wicaksana Putra rachmad.putra@tuwien.ac.at

arXiv:2304.04039v1 [cs.NE] 8 Apr 2023

**摘要**

脉冲神经网络（SNN）凭借其生物合理计算，在无监督设置下展现了实现高准确率以及低运行功耗/能耗的能力。先前的研究发现，基于DRAM的片外存储器访问是SNN处理能耗的主要来源。然而，最先进的研究并未优化DRAM 单次访问能耗，从而阻碍了基于SNN的系统进一步提升能效。大幅降低DRAM 单次访问能耗的一种有效方法是降低DRAM 电源电压，但这可能导致DRAM单元发生错误（即所谓的近似 *DRAM*）。为此，我们提出了*EnforceSNN*，这是一种全新的设计框架，可为嵌入式系统提供利用降压DRAM实现高可靠、能源高效的脉冲神经网络推理的解决方案。EnforceSNN的核心机制包括：（1）采用量化权重以降低DRAM 访问能耗；（2）设计高效的DRAM映射策略，以尽量降低DRAM 单次访问能耗；（3）分析SNN 错误容忍度，了解其在不同位错误率（BER）值下的准确率特征；（4）利用相关信息开发高效的故障感知训练（FAT），考虑不同的BER 值及DRAM中的位错误位置，以提升SNN 错误容忍度；（5）开发一种算法，选择在准确率、内存和能耗之间实现良好权衡的SNN模型。实验结果表明，与使用精确DRAM的基线脉冲神经网络相比，EnforceSNN在保持准确率的同时（即 $BER \leq 10^{-3}$ 无准确率损失），在不同网络规模下实现了最高84.9%的DRAM能耗节省，并将DRAM数据吞吐量最高提升至4.1倍。

**关键词：脉冲神经网络、高性能、能效、近似计算、近似 DRAM、DRAM 错误、误差容忍度、弹性。**

## 1 引言

脉冲神经网络（SNN）凭借其符合生物学特征的基于脉冲的计算，在无监督设置下以较低的运行能耗获得高准确率方面展现出了潜力（Putraand Shafique，2020年）。通常情况下，较大的脉冲神经网络模型更具优势，因为与较小的模型相比，其准确率更高，如我们在图 1(a) 中的实验结果所示。在这里，1MB大小的网络仅达到 75% 的准确率，而 200MB 大小的网络在 MNIST 数据集上的准确率达到 92%。MNIST 数据集是一组用于手写数字 0-9 的训练图像和测试图像（Lecun等人，1998 年）。另一方面，大多数 SNN硬件平台的片上存储器相对较小，例如小于100MB（Roy等人，2017年；Sen 等人，2017年；Frenkel 等人，2019年；Frenkel等人，2019年）。因此，在SNN硬件平台的片上存储器容量大于其片上存储器大小的情况下运行脉冲神经网络模型，将需要频繁访问片外存储器。先前的研究表明，对片外存储器（即动态随机存取存储器）的一次访问所产生的能耗，显著高于对片上存储器（即静态随机存取存储器）的一次访问（Sze等人，2017年；Putra 等人，2021年b）。此外，先前的工作还发现，内存访问在SNN处理的能耗中占主导地位，在不同的 SNN硬件平台上占系统总能耗的 50%-75%，如图 1(b)所示。其原因在于，DRAM 访问能耗显著高于其他 SNN操作（例如神经元操作）（Krithivasan等人，2019年）。对于具有严格约束的人工智能应用（例如片上存储器大小较小的低成本嵌入式设备）而言，这一问题更加严重（Shafique 等人，2021年），因为这会导致更加频繁的 DRAM 访问。因此，该问题阻碍了基于 SNN 的嵌入式系统进一步提升效率。

**目标研究问题：** 如何在保持准确率的同时，大幅降低脉冲神经网络推理的*DRAM*访问能耗。解决这一问题将为能量受限嵌入式设备及其在边缘人工智能和智能信息物理系统中的应用实现高效的脉冲神经网络推理。边缘人工智能 是指在网络边缘的资源和能量受限计算设备上运行人工智能（AI）算法的系统，即靠近数据源的设备（Chen 和 Ran，2019年；Shi 等，2016年；Satyanarayanan，2017年；Yu 等，2018年；Liu 等，2019年；Cao 等，2020年）。与此同时，智能信息物理系统（信息物理系统）是包含计算组件（例如计算与存储设备）、物理组件（例如传感器和执行器）以及人类用户之间交互网络的系统（Griffor 等，2017年；Chattopadhyay 等，2017年；Shafique 等，2018年；Kriebel 等，2018年）。

### 1.1 先进技术及其局限性

为降低脉冲神经网络推理的能耗，当前先进研究提出了多种技术，大致可分为以下几类。

- **通过近似神经元运算减少脉冲神经网络运算**（Sen 等人，2017年）、权重剪枝（Rathi 等人，2019年）以及神经元移除（Putra and Shafique，2020年）。这些技术减少了相应模型参数的 DRAM 访问次数。

- **量化**通过缩小脉冲神经网络参数（例如权重）的可表示值范围来实现（Rathi 等人，2019年；Putra and Shafique，2020年；Putra and Shafique，2021a）。这些技术减少了需要存储在动态随机存取存储器中并从中读取的脉冲神经网络参数（例如权重）的数量。

**局限性：** 这些最先进的工作主要旨在减少DRAM 访问次数，但没有优化DRAM 单次访问能耗，也没有在DRAM 中采用近似技术，而近似技术可提供额外的调节手段以实现高能效。因此，这些工作带来的优化收益并非最优，阻碍了SNN 推理系统充分发挥 DRAM 能耗节省的潜力。因此，有效的优化应联合最小化 DRAM 单次访问能耗以及

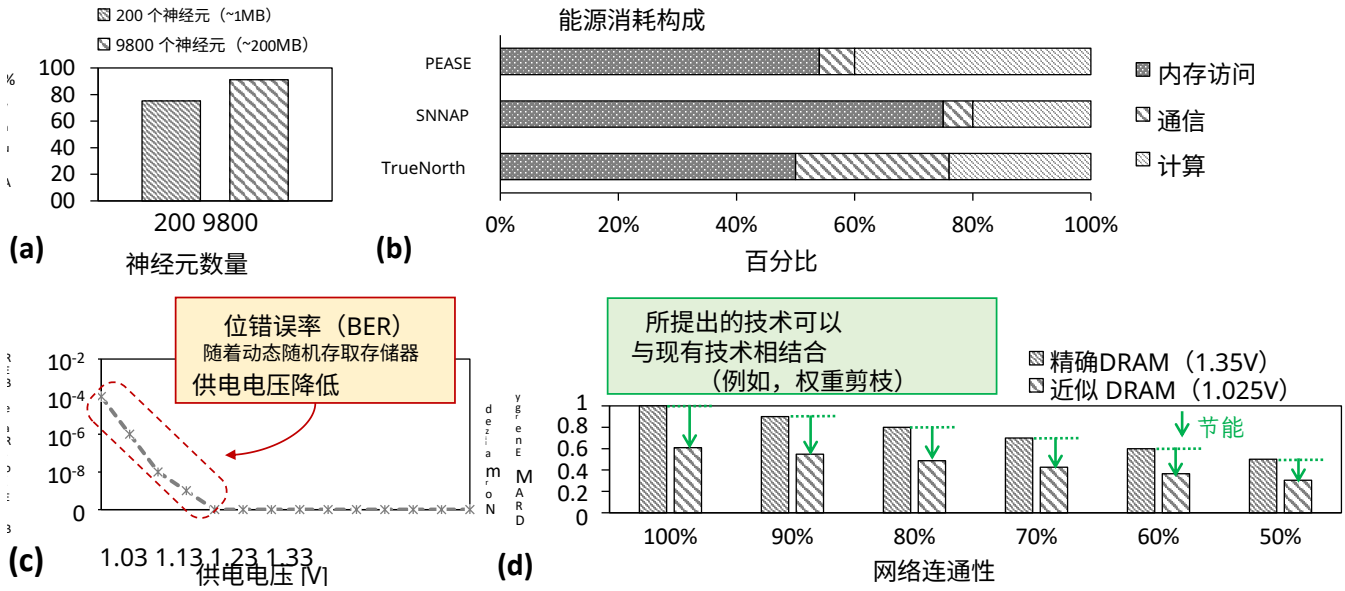



**图1**。 (a) 通过我们的实验获得的小型和大型脉冲神经网络模型在 MNIST 数据集上的准确率曲线。(b) 不同 SNN 硬件平台上的 SNN 处理能耗分解，即 PEASE （ Roy 等人，2017 年）、 SNNAP （ Sen 等人，2017年）和TrueNorth（Akopyan 等人，2015年）；改编自（Krithivasan 等人，2019年）的研究。(c)近似DRAM 的位错误率（BER）及其相应的供电电压 $V_{supply}$；改编自（Chang等人，2017年）的研究。(d)对于包含 4900 个兴奋性神经元的网络，在不同网络连通性（即突触连接）比例下，采用我们的方法与权重剪枝相结合所实现的DRAM 能耗节省估计值。结果通过使用 LPDDR3-1600 4Gb DRAM配置和 DRAMPower模拟器（Chandrasekar，2014年）进行实验获得。

DRAM 访问的数量，通过利用DRAM 中的近似来释放全部节能潜力，同时克服近似引起的误差（即 DRAM单元中的位翻转）所带来的负面影响。图 1(c) 展示了 DRAM 中近似引起的错误率。

为解决这些局限，我们采用近似 *DRAM*（即降低供电电压的 *DRAM*），并结合高效的 *DRAM*数据映射策略和故障感知训练，在保持准确率的同时，大幅降低 *SNN* 推理系统中的 *DRAM* 访问能耗。此外，我们提出的技术还可与最先进的技术相结合，进一步提升 SNN 推理的能效。例如，图 1(d) 展示了将我们的技术与权重剪枝相结合时所实现的预估 DRAM 能耗节省。为凸显降低电压的近似 DRAM 的潜力，我们将在下一节开展一项实验案例研究。

### 1.2 动机案例研究与关键挑战

在该案例研究中，我们旨在研究：(1) 精确和近似 DRAM 设置下 DRAM 位线电压（$V_{bitline}$）的动态变化；以及 (2) 不同访问条件（包括行缓冲区命中、未命中或冲突）下的 DRAM 访问能量。请注意， $V_{bitline}$ 指的是施加 DRAM 供电电压（$V_{supply}$）时，在每条 DRAM 位线上测得的电压，如图 2(a) 和 4(c) 所示。关于 $V_{bitline}$ 动态变化的更多细节见第 2.2.2 节。对于 DRAM 访问条件，行缓冲区命中意味着所请求的数据已加载到 DRAM 行缓冲区中，因此无需执行额外的 DRAM 操作即可访问数据。而行缓冲区未命中或冲突则需要先打开所请求的 DRAM 行，

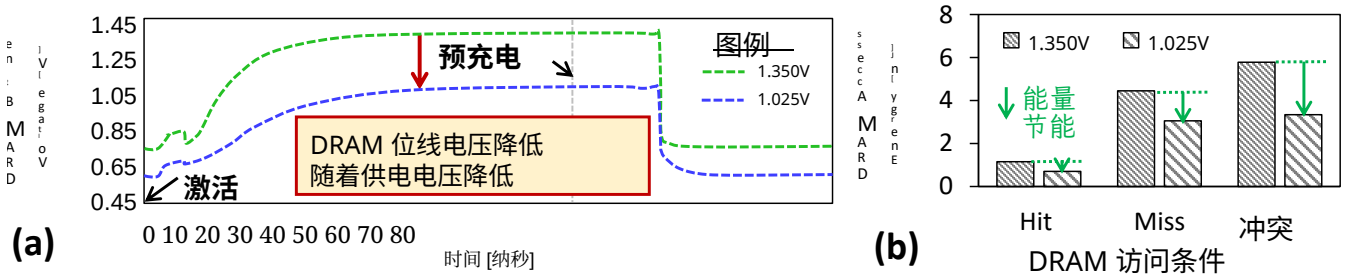



**图2**。 (a)不同 $V_{supply}$ 值下 $V_{bitline}$ 的动态特性。(b)在不同 $V_{supply}$ 值下，行缓冲区命中、行缓冲区未命中和行缓冲区冲突的 DRAM 访问能耗。

然后才能将数据加载到行缓冲区并进行访问。关于 DRAM 访问条件的更多信息见第 2.2.1 节。

对于实验设置，我们采用Chang等人（2017）的动态随机存取存储器电路模型和SPICE仿真器来研究 $V_{bitline}$的动态特性。精确动态随机存取存储器的供电电压（$V_{supply}$）为1.35V，而近似动态随机存取存储器的供电电压为1.025V。实验设置的更多细节将在第5节讨论。此外，我们考虑LPDDR3-1600 4Gb DRAM配置，因为它代表了适用于嵌入式系统的低功耗动态随机存取存储器类型。我们采用DRAMPower模拟器来估算动态随机存取存储器访问能耗，因为该模拟器已通过真实测量验证（Chandrasekar，2014），并已在计算机体系结构领域得到广泛使用。图2展示了实验结果，据此我们得出以下关键观察。

- $V_{bitline}$ 会随着 $V_{supply}$ 的降低而降低，因此迫使 DRAM 单元在可靠性更低的情况下运行，因为弱单元可能无法保持正确的比特。弱单元是指在降低 DRAM 参数（例如电压、时序）时会发生故障的 DRAM 单元（Chang等人，2017年；Kim等人，2018年）。
- 降压DRAM 可在不同访问条件下降低 DRAM 单次访问能耗，即每次访问的能耗最多可降低 42%。

- 行缓冲区命中的能耗低于行缓冲区未命中或冲突。此外，行缓冲区命中的延迟也低于行缓冲区未命中或冲突（Putra等，2020年；Putra等，2021年b）。因此，应利用行缓冲区命中来优化动态随机存取存储器的延迟和能耗。

尽管采用近似 DRAM 可以大幅降低 DRAM 单次访问能耗，但也会降低 DRAM 的可靠性，因为如图 1(c) 所示，随着 $V_{supply}$ 降低，位错误会增加。这些误差可能会降低脉冲神经网络推理的准确率，因为它们会改变 DRAM 中的权重值，进而损害神经元行为。

**相关研究挑战：**如何利用近似 *DRAM* 降低脉冲神经网络推理的 *DRAM* 访问能耗，同时尽量减轻其对准确率的负面影响。

### 1.3 我们的新贡献

为克服上述研究挑战，我们提出了**EnforceSNN框架**，该框架针对嵌入式系统中的近似DRAM（即降压DRAM），实现了具备韧性且能源高效的脉冲神经网络。根据我们所知，这是首个采用近似 DRAM提高脉冲神经网络推理能效，同时增强其针对DRAM中位误差的误差容忍能力的研究。我们的 EnforceSNN 框架采用以下关键步骤。

1. **采用权重量化**以减少脉冲神经网络权重的内存占用和脉冲神经网络推理的 DRAM 访问次数，从而优化DRAM 访问能耗。2. **设计高效的DRAM 数据映射**以最大化行缓冲区命中，从而在考虑 DRAM 中误码率的同时优化 DRAM 单次访问能耗。 3. **分析SNN 错误容忍度** 以了解不同DRAM 电源电压和不同 BER 值下的 SNN 准确率曲线。 4. **提高SNN错误容忍度** ，通过开发和采用高效的故障感知训练（FAT），考虑 SNN 准确率曲线和 DRAM 中的位错误位置。 5. **设计一种算法来选择 SNN模型** ，利用所提出的奖励函数，从给定的模型候选项中选择在准确率、内存和能耗之间实现良好权衡的模型。

**关键结果：**我们通过在多 GPU 机器上使用基于 PyTorch 的仿真（Hazan et al., 2018），针对 MNIST 和 Fashion MNIST 数据集评估 EnforceSNN 框架的（1）分类准确率[1], ，并使用 DRAMPower（Chandrasekar, 2014）评估（2）DRAM 访问能耗。对于脉冲神经网络模型、工作负载和训练误码率的每种组合，我们执行一个 epoch 的无监督学习（60K 次实验），以完成每个再训练过程；随后针对脉冲神经网络模型、工作负载和测试误码率的每种组合执行推理（10K 次实验）。实验结果表明，我们的 EnforceSNN 在 $BER \leq 10^{-3}$不同网络规模下将 DRAM 访问能耗最多降低 84.9%，并将加速比最多提高至 4.1 倍，同时保持准确率（准确率无损失）。

## 2 背景

### 2.1 脉冲神经网络（SNN）

脉冲神经网络是采用生物合理计算、并通过脉冲序列（即脉冲序列）传递信息的神经网络模型。这些脉冲采用特定的脉冲编码方式进行编码。文献中已提出多种脉冲编码方案（Gautrais and Thorpe, 1998; Thorpe and Gautrais, 1998; Kayser et al., 2009; Park et al., 2019; Park et al., 2020）。本文采用频率编码，因为它应用广泛，且对多种学习规则都具有鲁棒性（Diehl and Cook, 2015; Putra et al., 2021a）。在学习规则方面，本文采用脉冲时序依赖可塑性（STDP），因为已有研究广泛采用该方法（Diehl and Cook, 2015; Saunders et al., 2018; Hazan et al., 2018; Putra and Shafique, 2021b）。在脉冲神经网络模型中，神经元和突触按照特定架构相互连接（Pfeiffer and Pfeil, 2018; Tavanaei et al., 2019; Mozafari et al., 2019）。本文采用全连接架构，因为它支持无监督学习场景；见图3(a)。在该架构中，每个输入像素都与所有（兴奋性）神经元相连，而每个神经元的输出则与其他神经元相连，以实现抑制作用。对于神经元模型，本文采用漏积分发放（LIF）模型，因为它能以较低的计算复杂度实现符合生物学特性的神经元动力学（Izhikevich, 2004; Putra et al., 2022）；见图3(b)。

### 2.2 近似 DRAM

#### 2.2.1 动态随机存取存储器基础知识

动态随机存取存储器由通道、模块、秩、芯片、存储体、子阵列、行和列组成（Putra等人，2020年；Olgun等人，2021年）；见图4(a)。单个动态随机存取存储器请求可以并行访问同一秩内多个动态随机存取存储器芯片中的数据。在每个动态随机存取存储器芯片中，请求会被路由至特定的存储体、

[1] 基于无监督学习的脉冲神经网络研究仍处于早期阶段，通常使用 MNIST 和 Fashion MNIST 等小型数据集。我们也采用了脉冲神经网络研究领域广泛使用的相同测试条件。

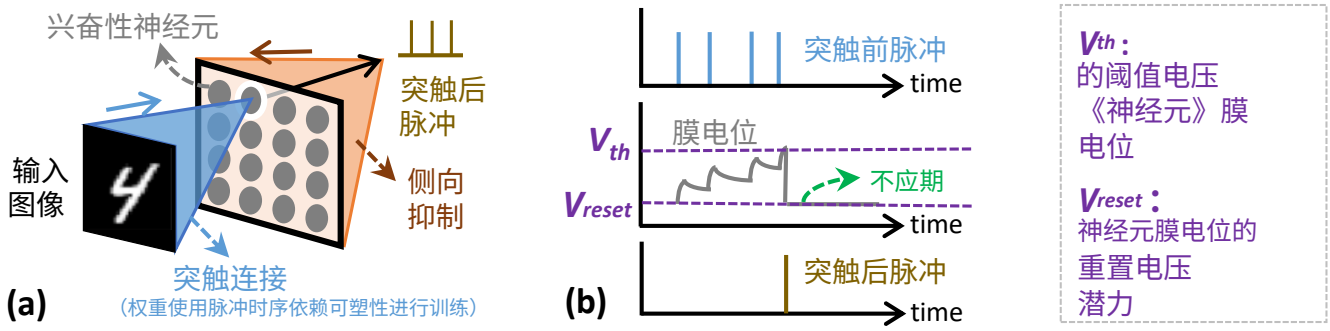



**图3**。 (a) 本文所考虑的脉冲神经网络架构，改编自（Putraand Shafique，2020年）。(b) 漏积分发放（LIF）神经元模型的神经元动力学。

行和列地址。当发出激活（ACT）命令时，请求的行会被打开，其数据会被复制到行缓冲器中。如果发出读取（RD）命令，则可以读取行缓冲器中的数据。如果发出写入（WR）命令，则可以用新数据替换行缓冲器中的数据。对于每个动态随机存取存储器请求，都存在不同的可能访问条件，即行缓冲区命中、行缓冲区未命中和行缓冲区冲突（Ghose等人，2019年）。行缓冲区命中是指请求的行已被激活，且其数据已经位于行缓冲器中的情况。因此，无需执行额外操作即可直接访问数据。否则，请求的行仍处于关闭状态，此时可能是行缓冲区未命中或行缓冲区冲突。行缓冲区未命中是指请求发生时没有已激活的行，因此在访问数据之前应先激活请求的行。同时，行缓冲区冲突是指请求的行仍处于关闭状态，但行缓冲器已被另一已激活的行占用。因此，在使用激活（ACT）命令激活请求的行之前，应先使用预充电（PRE）命令关闭已激活的行。图4(b)展示了动态随机存取存储器命令（即 ACT、RD 或 WR、PRE）及其时序参数（即 $t_{RCD}$：行地址到列地址延迟、 $t_{RAS}$：行激活时间，以及 $t_{RP}$：行预充电时间）。

### 2.2.2 降压DRAM

我们使用 SPICE仿真器和 Chang等人（2017年）提出的动态随机存取存储器电路模型开展了大量实验，同时考虑不同的供电电压（$V_{supply}$）值，以表征降压DRAM的参数，包括位线电压（$V_{bitline}$）以及相应的时序参数（即$t_{RP}$、 $t_{RCD}$和 $t_{RAS}$）。实验结果如图 4(c)所示，所得参数将用于进一步的 DRAM能耗估算。当$V_{bitline}$ 达到 $V_{supply}$的 75% 时，将该条件定义为就绪访问电压；如 所示，该电压代表可靠DRAM操作所需的最低 $t_{RCD}$。当 $V_{bitline}$ 达到$V_{supply}$的 98% 时，将该条件定义为就绪预充电电压；如 所示，该电压代表可靠DRAM操作所需的最低 $t_{RAS}$。同时，当 $V_{bitline}$ 处于 $V_{supply}/2$的 2% 范围内时，将该条件定义为就绪激活电压；如 所示，该电压代表可靠DRAM操作所需的最低 $t_{RP}$。

## 3 近似 DRAM 错误建模

Koppula等（2019年）的研究提出了四种误差模型，可高度贴合真实的基于降电压的近似DRAM，具体如下。**误差模型-0**：位错误在一个DRAM存储体中呈均匀随机分布；**误差模型-1**：位错误沿一个DRAM存储体的位线呈垂直分布；**误差模型-2**：位错误沿一个DRAM存储体的字线呈水平分布；以及**误差模型-3**：位错误呈均匀随机分布

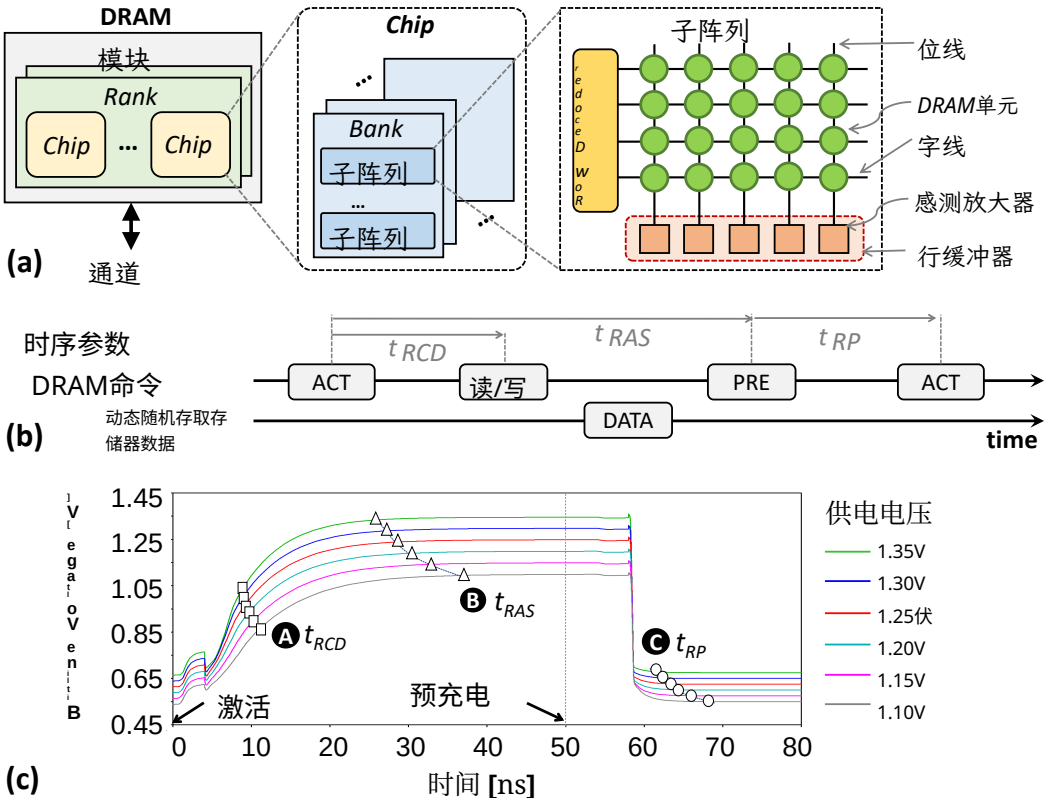



**图 4.** (a) 商用动态随机存取存储器的内部结构。(b) DRAM命令（即 ACT、RW 或 WR，以及 PRE）和 DRAM 时序参数（即 $t_{RCD}$、$t_{RAS}$ 和 $t_{RP}$）示意图。(c) DRAM 位线电压 $V_{bitline}$ 及其相应时序参数的动态变化。

该分布取决于DRAM单元的内容。在本研究中，我们采用**DRAM误差模型-0**，原因如下：（1）通过使用弱单元的百分比以及任一弱单元中的误差概率，该模型产生的误差与真实的基于降电压的近似DRAM中的误差高度相似；（2）它能够合理近似其他误差模型，包括近似（a）类似于误差模型-1的位线方向误差，（b）类似于误差模型-2的字线方向误差，以及（c）类似于误差模型-3的均匀随机分布；（3）它能够通过软件快速注入错误。先前的研究（Koppula等，2019年）也主要出于类似原因采用了DRAM误差模型-0。

# 4 ENFORCESNN 框架

## 4.1 概述

我们的 EnforceSNN 框架采用了图 5 所示的几个关键步骤。首先，我们对脉冲神经网络权重进行量化，以减少内存占用和 DRAM 访问能耗（第 4.2 节）。其次，我们设计感知错误的 *DRAM* 数据映射策略，以优化 DRAM 单次访问能耗（第 4.3 节）。这些优化使推理速度提升 4.1 倍，并节省 84.9% 的 DRAM 访问能耗（第 6 节）。随后，我们分析 *SNN* 错误容忍度，以了解不同 BER 值下脉冲神经网络推理的准确率曲线（第 4.4 节）。我们利用这些信息开发一种高效 *FAT* 技术，以提高 *SNN* 错误容忍度（第 4.5 节）。这些容错技术使 $BER \leq 10^{-3}$ 的重新训练速度提升 2.71 倍，且准确率没有损失（第 6 节）。我们还开发了一种脉冲神经网络模型选择算法，以找出在准确率、内存和能耗之间实现良好权衡的模型（第 4.6 节）。以下小节将详细介绍这些机制。

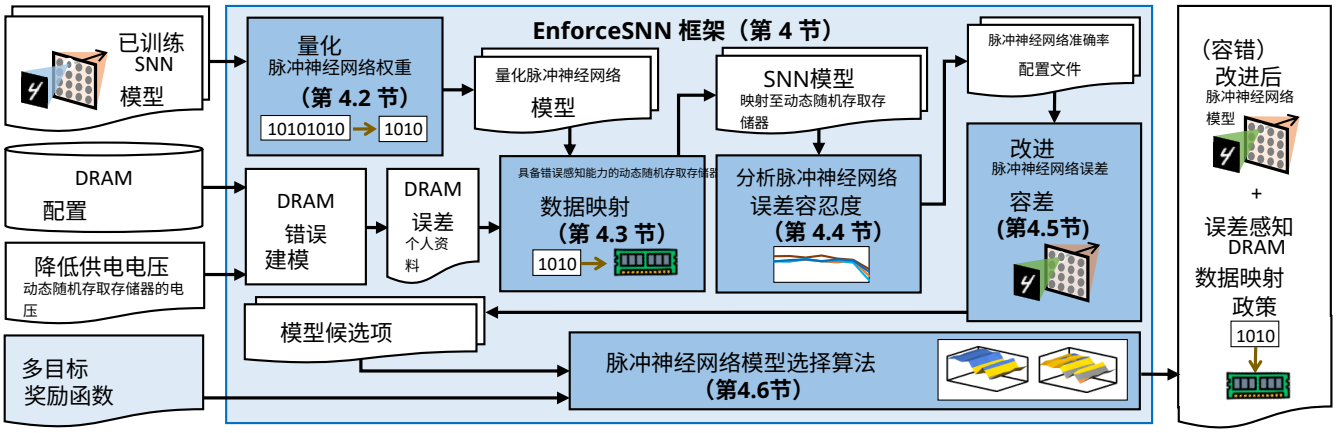



**图 5。** EnforceSNN 框架及其新颖机制（以蓝色框突出显示）。

### 4.2 量化脉冲神经网络权重

我们进行权重量化，以大幅减少内存占用和 *DRAM* 访问次数，从而节省 *DRAM* 能耗。这是因为量化是一种显著降低神经网络内存占用且不会明显降低准确率的技术（Gupta et al., 2015; Micikevicius et al., 2018）。此外，这是首次研究并利用考虑 DRAM 近似误差的 SNN 权重量化，因此相较于以往关于 SNN 权重量化的研究，带来了新的见解。我们的权重量化采用定点格式，可表示为 $Qi.f$。该格式由 1 个符号位、$i$ 个整数位和 $f$ 个小数位组成，并遵循二进制补码格式（即有符号 $Qi.f$）。此处，可表示值的范围为 $[-2^i, 2^i - 2^{-f}]$，精度为 $\epsilon = 2^{-f}$。我们选择“有符号 $Qi.f$”格式，以展示 EnforceSNN 框架能够提供通用且适用性强的解决方案，支持不同类型的生物学上合理的学习规则（例如 STDP 变体）；这些规则可能产生正值或负值权重（Rahimi Azghadi et al., 2014; Diehl and Cook, 2015）。为此，我们使用特定的舍入方案（例如截断、就近舍入或随机舍入），对训练后的 SNN 权重进行定点量化。作为一个研究案例，我们选择截断，因为它在计算复杂度较低的同时，能够提供有竞争力的准确率（Putra and Shafique, 2021a, 2022a,b）。为说明这一点，我们通过一项实验案例研究评估不同舍入方案对准确率的影响，结果如图 6 所示。截断（TR）保留 $f$ 位，并移除其余小数位。因此，对于给定的实数 $x$，采用 $Qi.f$ 格式时，其定点输出定义为 $TR(x,\ Qi.f) = \lfloor x \rfloor$。在我们的 SNN 模型中，我们采用 Diehl and Cook（2015）提出的成对权重依赖型 STDP 学习规则，将每个权重值（$w$）限制在规定范围内，即 $w = [0, 1]$。因此，对权重进行截断会将数值向下取整。在本研究中，我们采用 8 位定点格式，使用“有符号 $Q1.6$”和二进制补码格式（即 1 个符号位、1 个整数位和 6 个小数位），因为这种格式在无监督学习场景下能够为 SNN 提供较高的准确率（Putra and Shafique, 2021a）。请注意，如有需要，我们也可以在 EnforceSNN 中采用不带符号位的“无符号 $Qi.f$”格式，表示 8 位非负权重（即 1 个整数位和 7 个小数位）。对于“有符号 $Qi.f$”和“无符号 $Qi.f$”两种格式，整数部分均需 1 位来表示可能的最大权重值（即 $w = 1$）。

**量化步骤：**我们仅通过模拟量化方法对权重进行量化。该方法以定点格式表示权重值，并以浮点格式执行计算（Krishnamoorthi，2018年；Jacob等，2018年；Gholami等，2021年；van Baalen等，2022年）。为执行

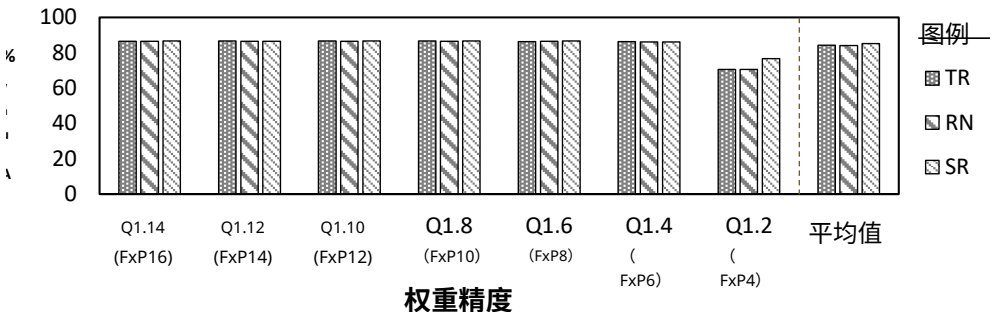



**图 6.** 在不同精度级别和舍入方案下，900 个神经元网络在 MNIST 上的准确率，即截断（TR）、就近舍入（RN）和随机舍入（SR）。这里，我们采用 Hopkins 等人（2020）研究中对 TR、RN 和 SR 的定义。请注意，定点格式 $Qi.f$ 也可以表示为 FxP(1+$i$+$f$)。这些结果表明，在 FxP8 精度基础上采用 TR，与其他舍入方案（即 RN 和 SR）相比，准确率具有竞争力。

量化，我们将权重值从 32 位浮点格式（FP32）转换为 8 位定点格式（带符号的 $Q1.6$）：在 32 位整数格式（INT32）下构造其 8 位二进制表示，从而在考虑符号位和舍入方案（即截断）的同时，便于执行按位修改和舍入操作。随后，我们通过类型转换将量化后的权重值（INT32）转换为 FP32 格式，再通过 $2^f$ 对这些值进行归一化。因此，量化后权重值的 8 位二进制表示以 FP32 格式保存，可用于基于 FP32 的算术计算。

**动态随机存取存储器错误注入：** 如果不存在动态随机存取存储器错误，则执行量化步骤，并可使用量化后的权重值（FP32）进行脉冲神经网络处理中的计算。如果存在动态随机存取存储器错误，则在考虑动态随机存取存储器错误注入的情况下执行量化步骤。这些误差会根据特定的动态随机存取存储器数据映射策略，注入量化权重（INT32）的 8 位二进制表示中。随后，我们将量化权重（INT32）的二进制表示转换为 FP32 格式，以便使用量化后的权重值进行脉冲神经网络处理中的计算。

### 4.3 面向错误的动态随机存取存储器数据映射策略

将脉冲神经网络模型正确映射到动态随机存取存储器中至关重要，这可以确保：(1) 权重受动态随机存取存储器中误差的影响尽可能小，从而维持准确率；(2) 优化动态随机存取存储器单次访问能耗。为此，我们设计并采用误差感知*DRAM*映射策略，将脉冲神经网络权重放置在动态随机存取存储器中，同时优化动态随机存取存储器单次访问能耗。所提出的动态随机存取存储器映射策略如图 7(a) 所示，其核心思想将在下文阐述。

1.权重被映射到适当的动态随机存取存储器部分（例如芯片、存储体或子阵列）中，其错误率满足误码率要求，即 $\leq$ 最大可容忍误码率（$BER_{th}$）。这里，我们考虑数据映射的子阵列级粒度，因为它使我们能够利用以下特征。

- 多银行突发特性，可在商品级动态随机存取存储器中使用，以提高吞吐量。其时序图如图 7(b) 所示。
- 子阵列级并行也可用于提高吞吐量，这种并行方式在新型动态随机存取存储器架构中得以实现（Kim 等，2012年）。

我们通过实验确定 $BER_{th}$，实验研究了网络在不同BER 值下的准确率特征。如果准确率得分显著低于不存在位误差时的基线准确率（即 $>$3% 的准确率下降），我们将相应的BER 值称为不可容忍的 *BER*，如图 8 中的1 和2 所示。

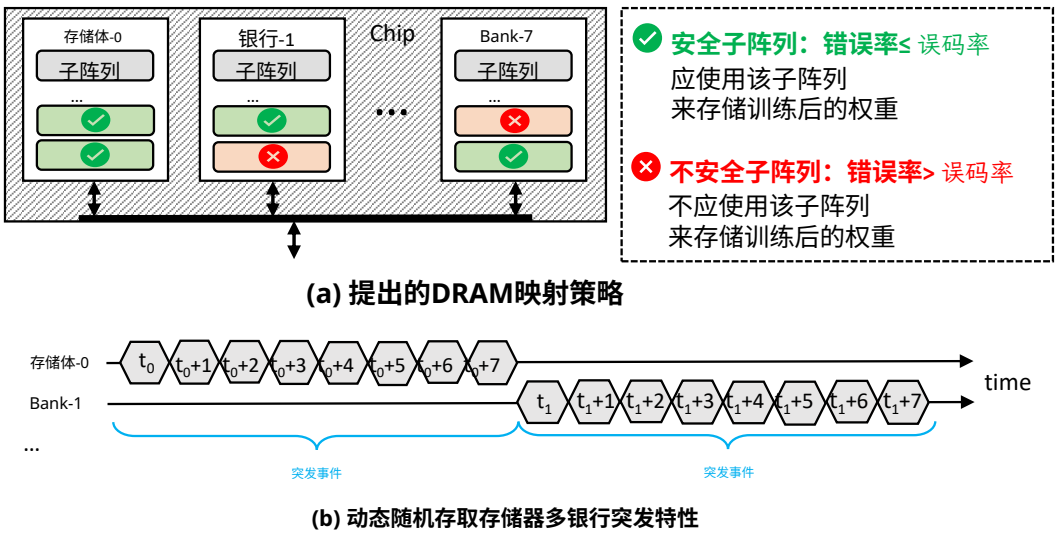



**图 7.** (a) 我们提出的 DRAM映射策略，利用了子阵列级粒度。(b) 动态随机存取存储器多银行突发特性的时序图。

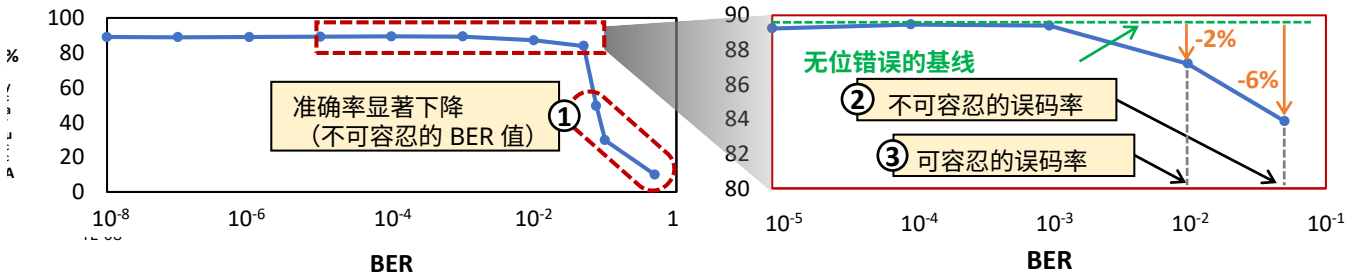



**图 8.** 一个包含 900 个神经元的网络在不同 BER 值下的测试准确率曲线，展示了可容忍和不可容忍的 BER。该网络采用类似图 3(a) 所示的全连接架构，其中每个输入像素都连接到所有神经元，并且每个神经元的输出都连接到其他神经元以执行抑制，从而在神经元之间形成竞争，以正确识别输入类别。

否则，我们将其定义为可容忍误码率。例如，图8中的3显示了与可容忍误码率相关联的准确率。基于这一讨论，我们定义 $BER_{th} = 10^{-2}$。2. 权重的映射方式旨在最大化行缓冲区命中次数，从而优化DRAM每次访问的能耗，同时利用多银行突发特性来最大化数据吞吐量。原因在于，与其他访问条件（即行缓冲区未命中或冲突）相比，行缓冲区命中产生的DRAM 访问能耗最低，正如我们在图 2(b) 中的实验结果所示。

为高效实现上述思想，我们设计了算法 1，其关键步骤如下。首先，我们识别出错误率为 $\leq BER_{th}$ 的子阵列，并将其称为安全子阵列，这些子阵列应当用于存储权重。相反，我们将错误率为 $> BER_{th}$ 的子阵列称为不安全子阵列，这些子阵列不应用于存储权重。该步骤在算法 1 的第 7 行中表示。其次，为最大化行缓冲区命中并利用多银行突发特性，每个动态随机存取存储器芯片中的数据映射应遵循以下策略（如算法 1 第 3–8 行所示）。

- **步骤 1：** 假设我们考虑将数据映射到目标存储体中的目标子阵列，该目标子阵列具有以下初始索引，即：子阵列索引 $= i$，存储体索引 $= j$。
- **步骤 2：** 如果目标子阵列是安全子阵列，我们会优先将数据映射到同一行的不同列，以最大化行缓冲区命中。否则，不使用该子阵列，并

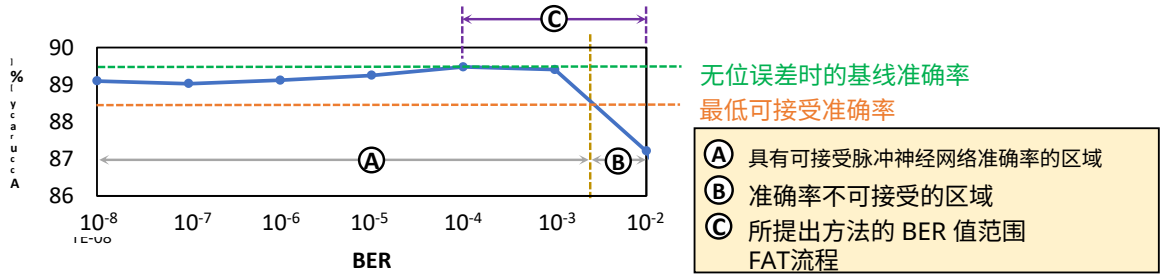



**图 9**。一个包含 900个《神经元》的网络的测试准确率曲线显示了准确率可接受区域、准确率不可接受区域，以及所提出的故障感知训练的 BER 值范围。注意，该网络与图 8 中的网络相同。此处，观察重点是再训练过程中可考虑的 BER 值，因此其范围比图 8 中的范围更小。

转而选择不同存储体中的另一个目标子阵列（子阵列索引 = *，存储体索引 += 1*）。然后，我们再次执行**步骤 2** 。如果所有存储体中同一行的列都已填满或不可用，则转而选择初始存储体中的另一个子阵列（子阵列索引 += *1*，存储体索引 =*j*），以利用子阵列级并行（如果适用）。

- **步骤-3：** 在目标子阵列中，剩余数据以与**步骤-2**相同的方式进行映射。当所有存储体中所有安全子阵列的同一行中的所有列都填满后，剩余数据将被放置到初始目标子阵列和存储体的另一行（子阵列索引 = 存储体索引 = *j*）。随后，我们再次执行**步骤-2** 至**步骤-3** ，直到所有数据都映射到动态随机存取存储器芯片中。如果仍有数据剩余，但动态随机存取存储器芯片中没有可用空间，则转到**步骤-4**。

- **步骤-4：** 剩余数据将分别使用**步骤-1** 至**步骤-3** 映射到不同的 DRAM 芯片、秩和通道中（如果适用）。

### 4.4 分析脉冲神经网络错误容忍度

前面的讨论强调，脉冲神经网络权重中的位错误会降低准确率，因为它们会改变权重值，使神经元行为偏离正确分类。因此，应提高 SNN错误容忍度，使脉冲神经网络模型即使存在较高错误率也能实现高准确率。为了有效增强SNN 错误容忍度，了解 DRAM 错误下的 SNN 准确率曲线非常重要。为此，我们的 *EnforceSNN*分析了考虑 *DRAM* 中数据映射模式和不同 *BER* 值的脉冲神经网络模型准确率曲线。我们观察到，当BER 较低时，准确率曲线通常具有可接受的准确率（即，与无误差时的基线相比，准确率下降在 1% 以内）；而当 BER 较高时，准确率会显著下降。因此，我们将准确率曲线划分为两个区域：A 具有可接受准确率的区域，以及B 具有不可接受准确率的区域，如图 9 所示。我们将在第4.5节中利用这些见解，开发一种高效的增强技术，以提高 SNN 错误容忍度。

### 4.5 Improving the SNNErrorTolerance

我们的 *EnforceSNN*通过故障感知训练（*FAT*）技术增强 *SNN* 错误容忍度，该技术融入了近似*DRAM*的错误特征。我们通过为再训练过程执行少量但有效的迭代，同时避免准确率崩溃，考虑在再训练过程中高效执行 FAT，以最小化训练时间、能耗和碳排放（Strubell等，2019年，2020年）。准确率崩溃定义为：由于在再训练过程开始时立即引入高误码率而导致训练发散，进而造成显著准确率下降（Koppula等，2019年）。所提出的 FAT技术包含以下关键步骤，这些步骤也在算法 2中进行了介绍。

**算法 1**所提出的 DRAM映射策略

**输入：(1)** 动态随机存取存储器（$DRAM$）：通道数（$n_{ch}$）、每通道的 rank 数（$n_{ra}$）、每个 rank 的芯片数（$n_{cp}$）、每个芯片的存储体数（$n_{ba}$）、每个存储体的子阵列数（$n_{su}$）、每个子阵列的行数（$n_{ro}$）、每行的列数（$n_{co}$）；
**(2)** 位错误率（BER）：子阵列的误码率（$BER\ subarray$）、最大可容忍误码率（$BER_{th}$）；
**(3)** 数据（$data$）；
**输出：** 动态随机存取存储器 ($DRAM$);

**BEGIN**
**进程**:
**对于** $ch = 0$ 到 $(n_{ch} - 1)$ **执行**
  **对于** $ra = 0$ 到 $(n_{ra} - 1)$ **执行**
    **对于** $cp = 0$至 $(n_c{}_p - 1)$ **执行**
      **对于** $ro = 0$ 至 $(n_{ro} - 1)$ **执行**
        **对于** $su = 0$ 至 $(n_{su} - 1)$ **执行**
          **对于** $ba = 0$ 至 $(n_{ba} - 1)$ **执行**
            **如果** $BER\ subarray[ch, ra, cp, ba, su] \leq BER_{th}$ **则**
              **对于** $co = 0$ 至 $(n_{co} - 1)$ **执行**
                $DRAM[ch, ra, cp, ba, su, ro, co] \leftarrow data$;
              **结束循环**
            **结束 if**
          **结束 for**
        **end for**
      **end for**
    **结束循环**
  **结束循环**
**结束循环**
**返回** $DRAM$;
**END**

- **步骤 1：**我们定义将在训练过程中纳入的 BER 值范围，以使脉冲神经网络模型能够适应动态随机存取存储器错误，如图 9 中的区域C 所示。我们在训练过程中纳入：（1）区域A 中接近区域B 的 BER 值；（2）区域B 中的所有 BER 值。具体而言，我们在训练过程中考虑区域A 中最高的两个 BER 值，使模型能够安全地适应较高的故障率，同时避免准确率大幅下降（即准确率崩溃），并缩短训练时间。

- **步骤 2：** 根据遵循动态随机存取存储器银行内均匀随机分布的 DRAM 误差模型-0，生成动态随机存取存储器中的位误差，适用于不同的 BER 值（对应不同的$V_{supply}$ 值）。

- **步骤 3：** 随后将生成的位误差注入 DRAM 单元位置，并翻转这些位置上的权重位。在此步骤中，我们采用第 4.3 节所述的拟议 DRAM 数据映射，以最大限度地提高行缓冲区命中次数并利用多银行突发特性。

- **步骤 4：** 之后，我们按照**步骤 1**中定义的 BER 值范围，将 BER 从最低值逐步提高到最高值，并在重新训练过程中纳入生成的位错误。每完成一轮重新训练，就按设定的比例提高 BER 值（例如，设为上一轮错误率的 10 倍）。通过这种方式，脉冲神经网络模型逐步经过训练，能够容忍从定义的最低错误率到最高错误率的动态随机存取存储器错误，从而稳步提升脉冲神经网络错误容忍度。

### 4.6 脉冲神经网络模型选择算法

根据前面的步骤，我们可以得到不同规模的容错脉冲神经网络模型，作为给定嵌入式应用的潜在解决方案。因此，我们需要考虑设计权衡，以便在给定的准确率、内存和能量约束下选择最合适的模型。为此，我们提出

**算法 2**所提出的 FAT 技术

输入：(1) 基线预训练脉冲神经网络：模型（$model_0$），准确率（$model_0.acc$）；(2) 动态随机存取存储器错误模型（$DRAMerr$）；(3) 重新训练的误码率：错误率（$BER$），错误率数量（$N_{BER}$）；(4) 训练数据集：样本（$S_{train}$），样本数量（$N_{train}$）；(5) 测试数据集：样本（$S_{test}$），样本数量（$N_{test}$）；输出：(1) 改进后的脉冲神经网络：模型（$model_1$），准确率（$model_1.acc$）；开始 初始化：1：$model_{temp} = model_0$；2：$model_1 = model_0$；3：$model_1.acc = 0$；处理：4：对于 $i = 0$ 到（$N_{BER} - 1$）执行 5：$error\ map = DRAMerr（BER[i]）$；// 错误生成 6：将 $error\ map$ 注入 $model_{temp}$；// 错误注入 7：对于 $r = 0$ 到（$N_{train} - 1$）执行 8：使用 $S_{train}[r]$ 训练 $model_{temp}$；// 训练 9：结束循环 10：对于 $s = 0$ 到（$N_{test} - 1$）执行 11：使用 $S_{test}[s]$ 测试 $model_{temp}$；// 测试 12：结束循环 13：如果 $model_{temp}.acc > model_1.acc$ 则 14：$model_1 = model_{temp}$；15：$model_1.acc = model_{temp}.acc$；16：结束条件判断 17：结束循环 18：返回 $model_1$；结束

一种利用我们提出的奖励函数量化 *SNN* 模型候选项权衡收益的算法，然后选择收益最高的模型。我们的多目标奖励函数（$R$）旨在优先选择准确率高、内存占用小且能耗低的模型。奖励 $R$ 定义为准确率与内存占用和能耗之间的结果，如公式 1 所示。在该公式中，$acc_x$ 表示所研究 SNN 模型（$x$）的准确率。$m_{norm}$ 表示归一化内存，其定义为所研究模型（$mem_x$）与浮点模型（$mem_{fp}$）的内存占用之比；见公式 2。模型的内存占用通过利用权重数量（$N_{wgh}$）及其对应的位宽（$BW_{wgh}$）进行估算；见公式 3。同时，$E_{norm}$ 表示归一化能耗，其定义为所研究模型中近似 DRAM（$E_{DRAM\ approx}$）与精确 DRAM（$E_{DRAM\ accurate}$）的 DRAM 访问能耗之比；见公式 4。为定义在计算 $R$时内存和能耗相对于准确率的重要性，我们分别采用 $\mu$ 和 $\varepsilon$ 作为内存和能耗的可调权衡变量。其中，$\mu$ 和 $\varepsilon$ 为非负实数。

$$R(acc_x, m_{norm}, E_{norm}) = acc_x - (\mu \cdot m_{norm} + \varepsilon \cdot E_{norm}) \tag{1}$$

$$m_{norm} = \frac{mem_x}{mem_{fp}} \tag{2}$$

$$mem = N_{wgh} \cdot BW_{wgh} \tag{3}$$

$$E_{norm} = \frac{E_{DRAM_approx}}{E_{DRAM_accurate}} \tag{4}$$

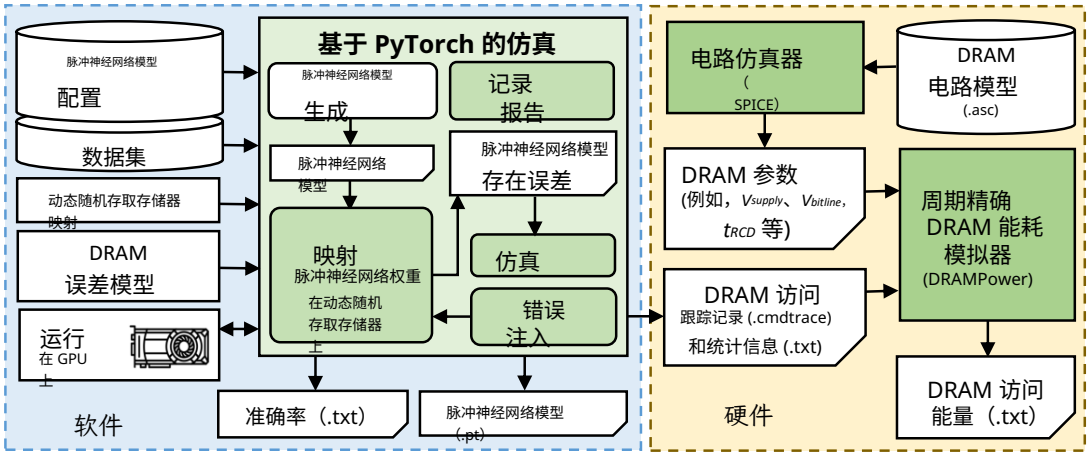



**图 10.** 我们的实验设置和工具流程。

## 5 评估方法

图 10 展示了用于评估我们 EnforceSNN 框架的实验设置和工具流程，具体说明如下。

**准确率评估：** 我们采用基于 PyTorch 的仿真（Hazan等人，2018年），使用 32位浮点（FP32）和 8 位定点精度（即采用“signed $Q1.6$”和“unsigned $Q1.7$”格式的 FxP8），在多GPU机器（即 NvidiaGeForceRTX2080Ti）上运行。对于网络架构，我们考虑具有不同数量兴奋性神经元的全连接网络，为简洁起见，将其称为 N-$i$，其中$i$ 表示兴奋性神经元的数量。我们使用频率编码将每个输入像素转换为脉冲序列，并使用 MNIST和 FashionMNIST数据集。作为对比对象，我们使用未考虑 DRAM错误进行预训练的脉冲神经网络模型作为基线。对于每个重新训练过程，针对脉冲神经网络模型、数据集和训练误码率的每种组合，我们通过60K 次实验执行一个epoch 的基于脉冲时序依赖可塑性的无监督学习。随后，针对脉冲神经网络模型、数据集和测试误码率的每种组合，我们通过 10K 次实验执行推理。

**动态随机存取存储器误差生成与注入：** 首先，我们根据动态随机存取存储器误差模型 0 生成位错误，并在考虑动态随机存取存储器中的数据映射策略的同时，将其注入动态随机存取存储器单元位置。随后，存储在发生错误的动态随机存取存储器单元位置中的权重位将发生翻转。对于基线数据映射，我们将权重位放置在动态随机存取存储器存储体中的连续地址，以充分利用动态随机存取存储器的突发特性；如果一个存储体已满，则将权重位映射到同一动态随机存取存储器芯片中的另一个存储体。同时，我们在 EnforceSNN 中采用算法 1 所提出的动态随机存取存储器映射方法。

**DRAM 能耗评估：** 我们采用 Chang等人（2017）提出的 DRAM 电路模型和 SPICE仿真器来提取 DRAM 的工作参数（例如，$V_{supply}$、$V_{bitline}$、$t_{RCD}$、$t_{RAS}$、$t_{RP}$），同时考虑了 LPDDR3-1600 4Gb DRAM 的配置，该配置具有代表性，可用于嵌入式系统的主存。精确 DRAM 的 $V_{supply}$为 1.35V，而近似 DRAM 的 $V_{supply}$为 1.025V–1.325V。随后，我们使用最先进的周期精确 DRAMPower（Chandrasekar，2014），结合 DRAM 访问跟踪记录和统计信息以及提取的 DRAM 参数，估算 DRAM 访问能耗。

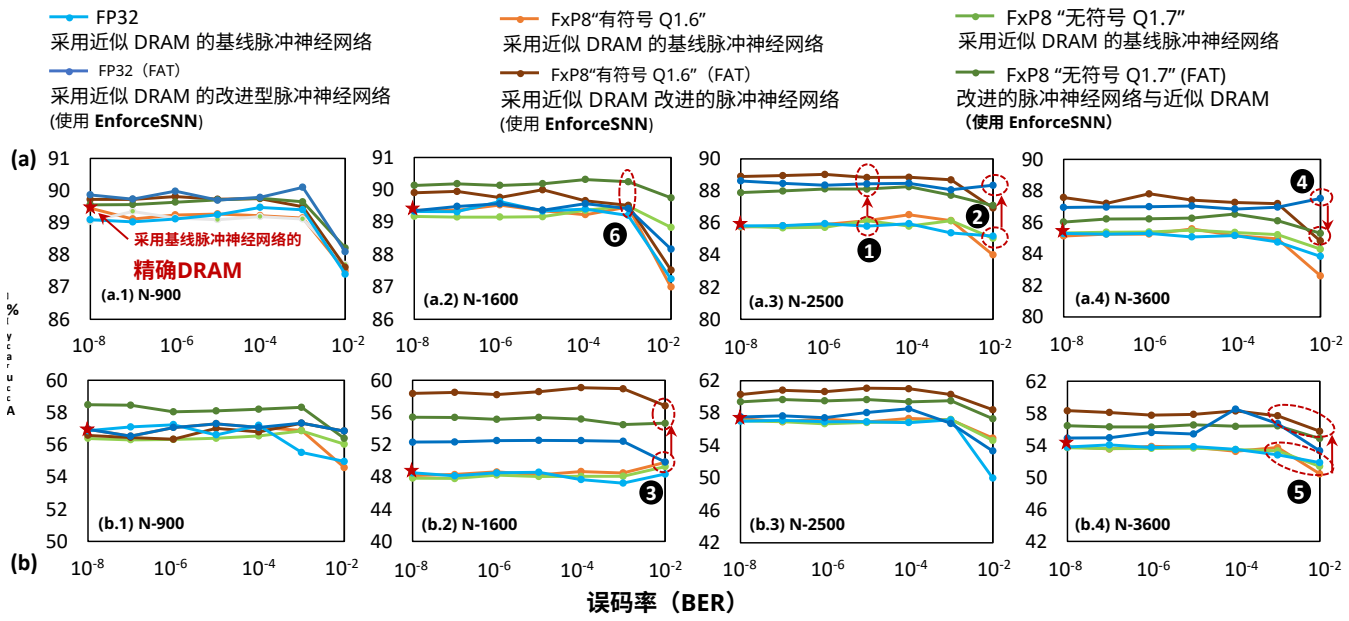



**图 11**。在不同精度级别、不同 BER 值和不同网络规模下，（a）MNIST和（b）Fashion MNIST 数据集的准确 DRAM 与近似 DRAM中基线模型的准确率，以及采用近似 DRAM 的 EnforceSNN 改进模型的准确率。

## 6 结果与讨论

### 6.1 SNN 错误容忍度的改进

图 11 展示了基线模型和 EnforceSNN 改进模型在精确和近似 DRAM 条件下，针对不同 BER 值、精度级别（即 FP32 和 FxP8（“有符号 $Q1.6$”和“无符号 $Q1.7$”））、网络规模及工作负载（即 MNIST 和 Fashion MNIST 数据集）的准确率。总体而言，我们发现，采用近似 DRAM 的基线模型准确率低于采用精确 DRAM 的基线模型，且准确率会随着 BER 增大而下降。在不同权重精度级别、网络规模和数据集中，均可观察到这些趋势。原因在于，权重若存储在存在故障的 DRAM 单元中就会发生变化（即翻转），而模型并未经过训练来适应此类位翻转。因此，相应的神经元行为会偏离预期，进而导致准确率下降。另一方面，如 所示，在不同 BER 值、网络规模和数据集中，采用近似 DRAM 的 EnforceSNN 改进模型，其准确率均高于采用精确和近似 DRAM 的基线模型。我们还发现，即使在高错误率情况下（即 $BER = 10^{-2}$），采用近似 DRAM 的 EnforceSNN 改进模型的准确率仍高于采用精确和近似 DRAM 的基线模型；对于 FP32 权重精度级别，如 所示；对于 FxP8 权重精度级别，如 所示。原因在于，我们的 EnforceSNN 在训练过程中纳入了不同 BER 值下近似 DRAM 的误差特征，使脉冲神经网络模型能够适应 DRAM 错误的存在，从而提高 SNN 错误容忍度。对于 MNIST 数据集，高错误率（即 $BER = 10^{-2}$）通常会使 SNN-FxP8 的准确率比 SNN-FP32 下降得更多，如 所示。原因在于，MNIST 数据集中每个类别用于表征数字特征的权重分布较窄，因此，由于 FxP8 的位宽较短，位错误可能比在 FP32 中更显著地改变权重值。因此，相应的神经元行为会偏离理想行为，进而降低准确率。对于 Fashion MNIST 数据集，在某些情况下，SNN-FxP8 的准确率可能高于 SNN-FP32，如 所示。可能的原因如下：Fashion MNIST 数据集的特征比 MNIST 更复杂

数据集，因此每个类别的权重分布更广，以表示其各种特征，而这些特征可能与其他类别的特征重叠（即非独有特征）。随后，量化通过消除训练后的权重中不太重要的位（类似于去噪效应）来去除这些非独有特征，而再训练则使量化权重能够适应位翻转，因此其准确率高于未量化权重。此外，我们还观察到，SNN-FxP8 的准确率在错误率较高时（即 $BER = 10^{-2}$）开始出现明显下降。对于量化模型，总体而言，“无符号 $Q1.7$”和“有符号 $Q1.6$”格式呈现出相似的趋势，且准确率相近，因为它们表示的权重值相似，差异仅在于最低有效小数位，因此神经元行为和准确率也相近。如6所示，在某些情况下（例如再训练过程之后），这些格式的准确率可能存在显著差异。可能的原因是，这些格式中符号位、整数位和小数位的位置各不相同，因此 DRAM 错误会影响不同的权重位，并导致各自在故障感知训练期间具有不同的学习质量。

总而言之，与使用精确DRAM的基线相比，我们的 EnforceSNN 在 $BER \leq 10^{-3}$ 跨不同数据集时保持了准确率（即准确率没有损失）。与此同时，对于更高的 BER 值（即 $10^{-3} < BER \leq 10^{-2}$），我们的 EnforceSNN在不同数据集上的准确率仍然高于使用精确DRAM的基线。因此，这些结果表明，我们的 *EnforceSNN* 框架以最小的重新训练开销，有效提升了脉冲神经网络对 *DRAM* 错误的 *SNN* 错误容忍度。

### 6.2 DRAM 访问能耗节省与吞吐量提升

图 12(a) 展示了基线模型和 EnforceSNN 改进模型在精确和近似 DRAM 下，为执行一次推理（即推断一个输入样本）所需 DRAM 访问的归一化能耗，涵盖不同 $V_{supply}$ 取值、精度级别、网络规模和工作负载。我们观察到，不同网络规模的归一化 DRAM 访问能耗相似，因此仅展示一张图来表示所有网络规模的实验结果。对于不同网络规模下的精确 DRAM 情况，基线模型采用量化技术后可实现 75% 的 DRAM 能耗节省；如 所示，我们的 EnforceSNN 改进模型通过量化和所提出的 DRAM 映射策略，实现了 75.1% 的 DRAM 能耗节省。与此同时，这些 DRAM 能耗节省的差异源于 DRAM 映射策略。也就是说，我们的 EnforceSNN 通过最大限度地提高行缓冲器命中率并利用多银行突发特性，优化 DRAM 单次访问能耗，因此相比仅利用单银行突发特性的基线模型，行缓冲区冲突更少。对于不同网络规模下的 FP32 精度，与使用精确 DRAM 相比，基线模型使用近似 DRAM 会使 DRAM 能耗节省最多降低 39.2%。与此同时，如 所示，与使用精确 DRAM 相比，EnforceSNN 改进模型使用近似 DRAM 会使 DRAM 能耗节省最多降低 39.5%。这些能耗节省源于运行 $V_{supply}$减少后 DRAM 单次访问能耗的降低。此外，基线模型与我们的 EnforceSNN 在能耗节省方面的差异也源于 DRAM 映射策略。对于不同网络规模下的 FxP8 精度（即“带符号 $Q1.6$”和“无符号$Q1.7$”），与使用精确 DRAM 相比，基线模型使用近似 DRAM 会使 DRAM 能耗节省最多降低 84.8%。与此同时，如 所示，与使用精确 DRAM 相比，EnforceSNN 改进模型使用近似 DRAM 会使 DRAM 能耗节省最多降低 84.9%。这些能耗节省源于权重精度降低，以及 $V_{supply}$ 减少后 DRAM 单次访问能耗的降低。此外，基线模型与我们的 EnforceSNN 在能耗节省方面的差异也源于 DRAM 映射策略。此外，我们还观察到，在不同 $V_{supply}$ 取值、工作负载和网络规模下，我们的 EnforceSNN 改进模型相较于基线模型实现了 4.1 倍的吞吐量加速；见 。

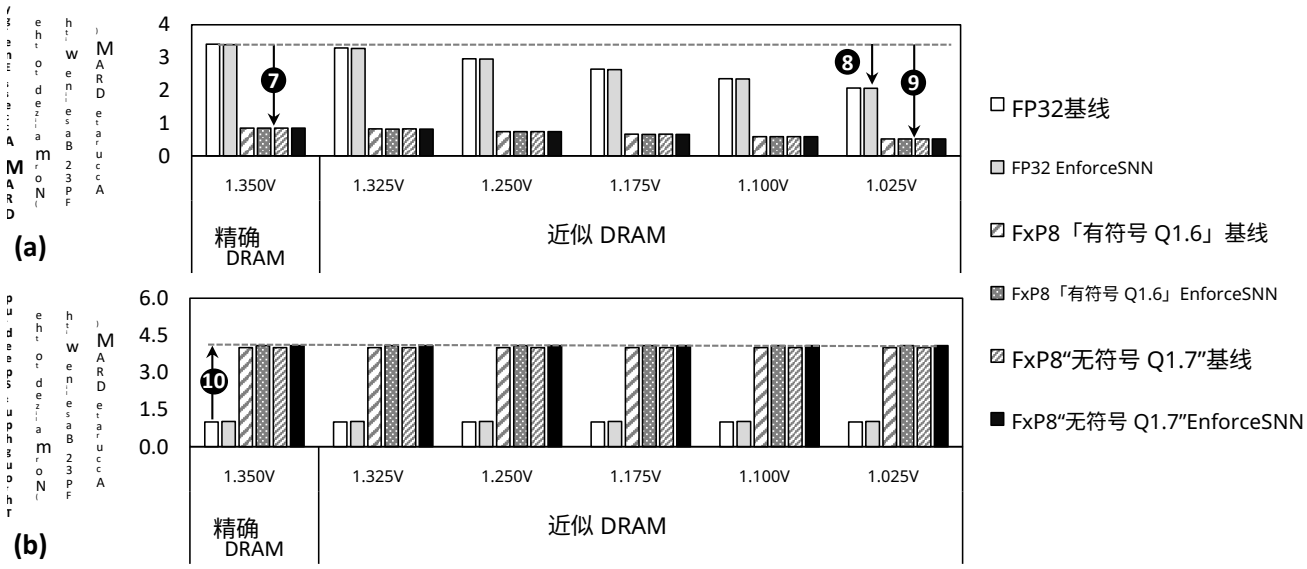



**图 12.** (a) 基线模型和 EnforceSNN 改进模型在采用精确 DRAM 和近似 DRAM 时，每次推理产生的归一化 DRAM 访问能耗；(b) 在不同 $V_{supply}$ 值、不同工作负载（数据集）以及不同网络规模（N-900、N-1600、N-2500 和 N-3600）下，我们的 EnforceSNN 改进模型相对于基线模型实现的每次推理 DRAM 数据吞吐量归一化加速比。这些结果适用于所有网络规模，也适用于 MNIST 和 Fashion MNIST 数据集，因为这两种工作负载每次推理的权重数量和 DRAM 访问次数相同，因此 DRAM 访问能耗相近。

这一结果通过以下方式实现：(1) 量化技术减少了 DRAM 访问次数；(2) 我们提出的 DRAM 映射策略通过最大限度地提高行缓冲器命中率并利用多银行突发特性，优化了 DRAM 单次访问延迟。结果还表明，"无符号 $Q1.7$"和"带符号 $Q1.6$"实现了相近的 DRAM 访问能耗节省和吞吐量提升，因为它们采用相同位宽的权重，因此具有相似的 DRAM 访问行为。

总之，图 12 的结果表明，我们的 *EnforceSNN* 框架通过采用降电压近似 *DRAM* 和高效的 *DRAM* 映射策略，大幅降低了 *DRAM* 访问能耗，同时主要得益于量化，有效提升了 *DRAM* 数据吞吐量。

### 6.3 设计权衡下的模型选择

图 13 和图 14分别展示了MNIST 和 FashionMNIST数据集的准确率-内存-能量权衡结果。在此评估中，量化模型采用带有"signed $Q1.6$"格式的FxP8精度。对于给定的 SNN 模型候选项，我们观察到，内存大小较小的模型通常采用 FxP8精度，如图 13(a) 中的 MNIST和图 14(a) 中的 Fashion MNIST 所示。考虑到基于FxP8的模型的准确率与基于 FP32 的模型相当，我们将候选项缩减为仅包含基于 FxP8 的模型。

为分析设计权衡，我们探究不同 $\mu$ 和 $\varepsilon$ 取值对奖励的影响。例如，如果我们认为准确率应优先于内存和能耗，则将 $\mu$ 和 $\varepsilon$ 设为较低值（例如， $\mu = 0$ 和 $\varepsilon = 0$）。另一方面，如果我们认为内存应优先于能耗，则将 $\mu$ 设为高于 $\varepsilon$ （例如， $\mu = 10$ 和 $\varepsilon = 0$）。在这两种情况下，在 $10^{-5}$ 错误率下，MNIST 的最高奖励由 EnforceSNN 改进型 N-1600 FxP8 获得，Fashion MNIST 的最高奖励则由 EnforceSNN 改进型 N-2500 FxP8 获得；参见 。

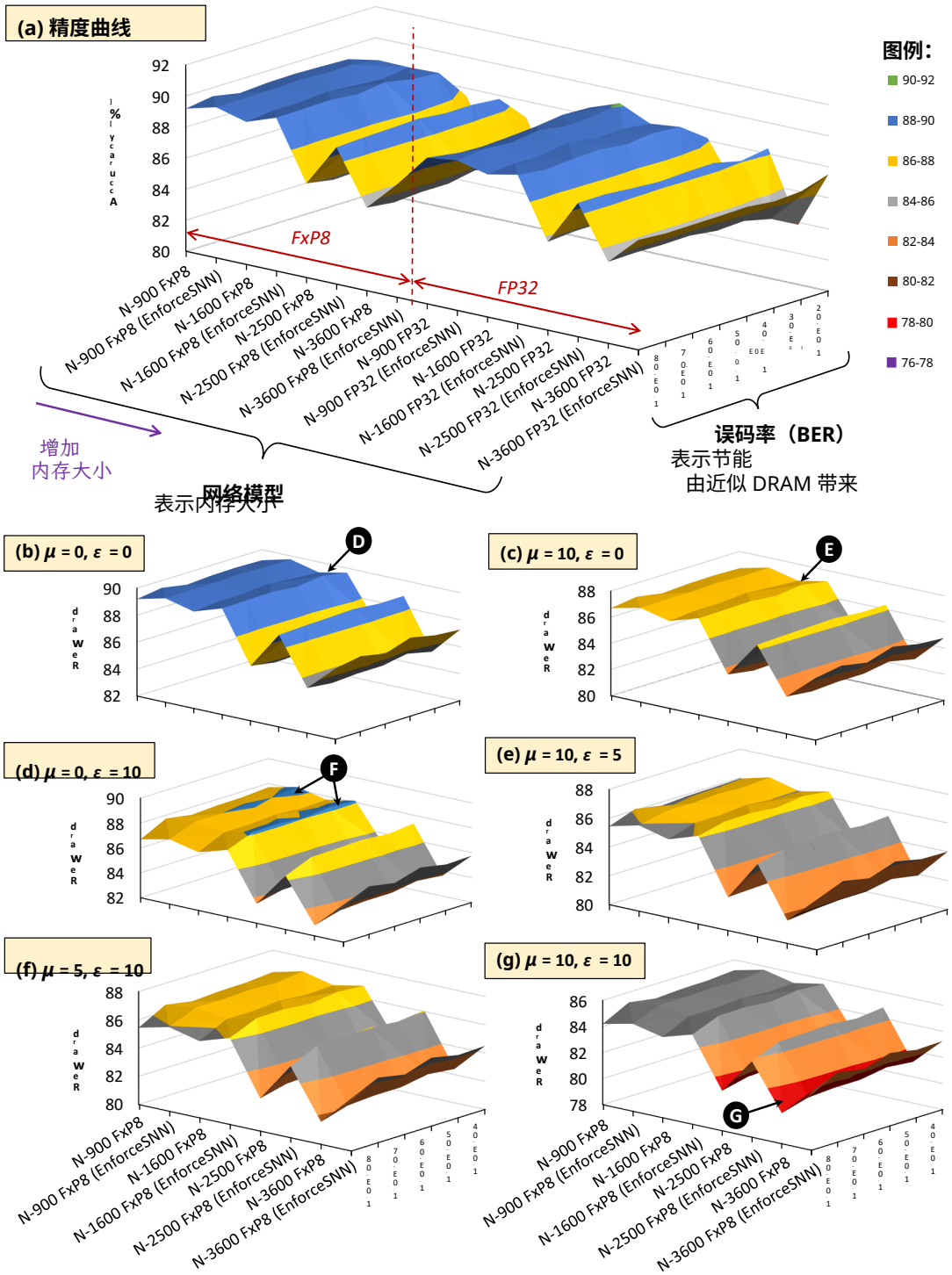



**图13。** MNIST 的准确率、内存占用和能耗之间的权衡。(a) SNN模型的准确率曲线。(b-g) SNN模型的奖励曲线。网络规模表示内存大小，BER 值表示近似 DRAM 带来的能耗节省。

$\mu = 0$ 和 $\varepsilon = 0$ 的情况，并参[redacted]，了解 $\mu = 10$ 和 $\varepsilon = 0$ 的情况。原因在于，这些模型采用了我们的高效 FAT 技术来提高误差容忍度，因此即使在高错误率下也能保持较高的准确率。我们还观察到，当 $\mu$ 高于 $\varepsilon$ 时，高奖励会转向规模较小的模型，如图 13(c) 中的[redacted]所示。原因是，较高的 $\mu$ 会使较小的 $m_{norm}$ 对奖励降低的影响小于较大的 $m_{norm}$，从而维持较高的奖励值。如果能耗应优先于内存占用，则将 $\mu$ 设为低于 $\varepsilon$（例如，

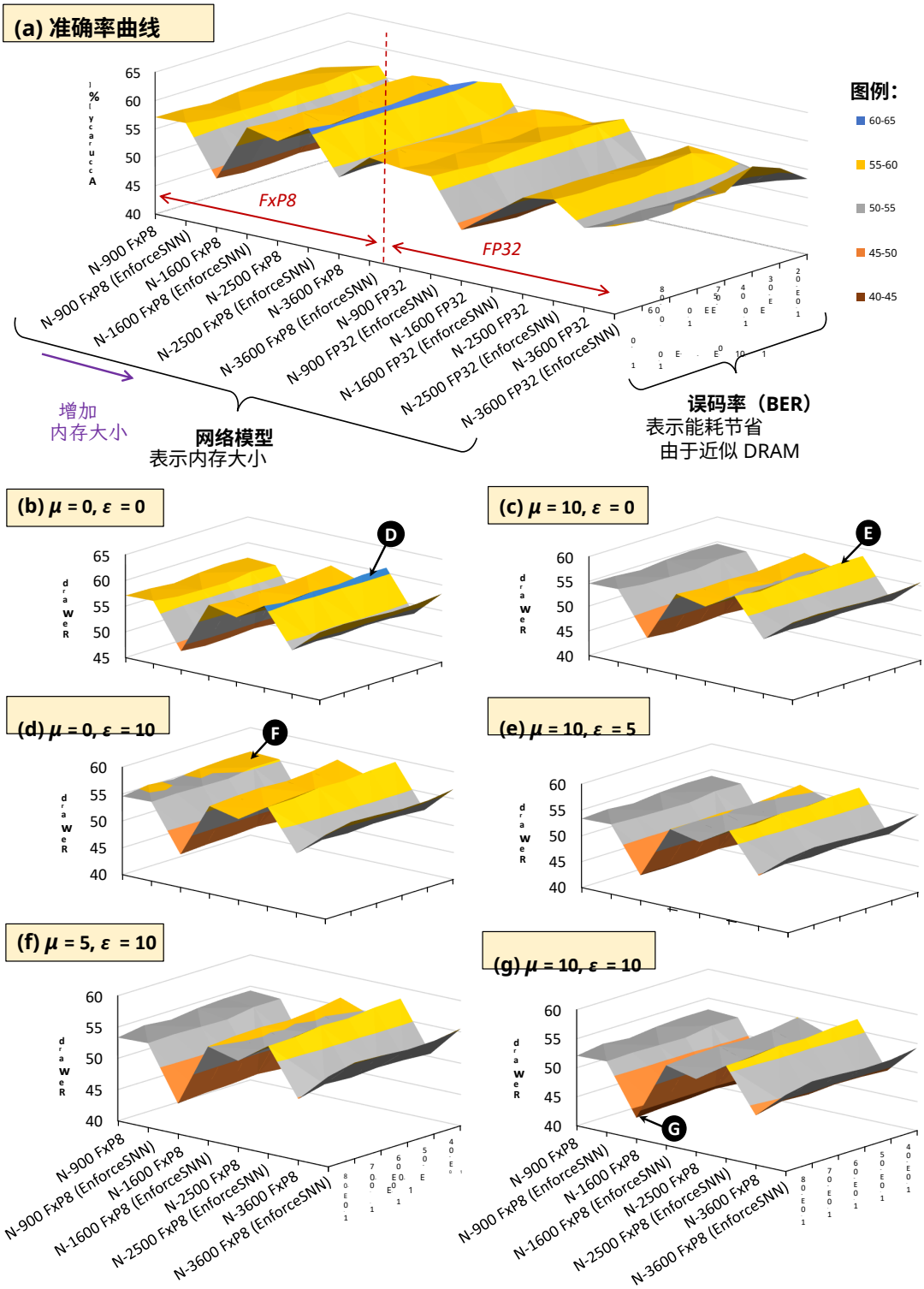



**图 14.** Fashion MNIST 中准确率、内存和能耗之间的权衡。(a) 脉冲神经网络模型的准确率曲线。(b-g) 脉冲神经网络模型的奖励曲线。网络规模表示内存大小，BER 值表示近似 DRAM 带来的能耗节省。

$\mu = 0$ 以及 $\varepsilon = 10$）。EnforceSNN 改进型 N-1600 FxP8 在$10^{-5}$ MNIST的错误率以及 EnforceSNN 改进型 N-2500 FxP8 在 $10^{-4}$ FashionMNIST的错误率方面获得了最高奖励。在这种情况下，我们观察到，高奖励正转向能耗更低的模型（以更高的误码率表示）；参见 ，图 13(d) 和图 14(d)。原因是，更高的 $\varepsilon$ 会使较小的 $E_{norm}$ 对奖励降低的影响小于较大的 $E_{norm}$，

从而维持较高的奖励值。此外，如果内存和能耗应优先于准确率，我们将 $\mu$ 和 $\varepsilon$ 设置为高（例如， $\mu = 10$ 和 $\varepsilon = 10$）。在 $10^{-5}$ 误码率下，EnforceSNN改进型 N-1600 FxP8 在 MNIST 上获得最高奖励；在 $10^{-4}$ 误码率下，EnforceSNN 改进型 N-2500 FxP8 在 FashionMNIST上获得最高奖励。在这种情况下，高奖励会转向内存和能耗较小的模型（由较高的 BER 表示），但随着 $\mu$ 和 $\varepsilon$ 的值增加，其总体奖励会降低；参见 ，图 13(e) 和图 14(e)。原因是，较高的 $\mu$ 和 $\varepsilon$ 会共同使 $m_{norm}$ 和 $E_{norm}$ 降低奖励。这意味着，如果我们希望显著减少内存占用和能耗，就必须接受更大的准确率下降。

总之，图 13 和图 14 中的结果表明，我们的 *EnforceSNN* 框架采用了有效的算法，可在给定脉冲神经网络模型的准确率、内存占用和能耗之间进行权衡，从而为具有各自约束条件的多种嵌入式应用提供良好的适用性。

### 6.4 再训练成本优化

用于神经网络的传统FAT通常会在再训练过程中以递增速率将误差从最小值注入到最大值，以避免准确率崩溃（Koppula等，2019年）。因此，在本文中，传统FAT考虑 $BER = \{10^{-8}, 10^{-7}, 10^{-6}, ..., 10^{-2}\}$，而我们在EnforceSNN中提出的高效FAT仅在再训练过程中考虑 $BER = \{10^{-4}, 10^{-3}, 10^{-2}\}$ 。

**再训练加速：** 采用 FxP8 的传统 FAT（cFAT8）相较于采用 FP32 的传统 FAT（cFAT32），在 MNIST 和 Fashion MNIST 上的加速比分别最高可达1.16 倍和1.14倍，这是因为 cFAT8 采用了量化权重，从而实现了更快的错误注入和学习过程。同时，采用 FP32的高效 FAT（eFAT32）相较于 cFAT32 实现了2.33倍的加速，这是因为我们的 eFAT32 的再训练过程迭代次数更少。此外，我们还观察到，采用 FxP8 的高效 FAT（eFAT8）可实现更高的加速，如 在图15(a)中所示，在MNIST 和 Fashion MNIST 上的加速比分别最高可达 2.71 倍和 2.65 倍，这是因为我们的 eFAT8 除了采用量化权重之外，再训练过程的迭代次数也更少。

**重新训练能耗节省：** 对于 MNIST，cFAT8 相较于 cFAT32 的节能幅度最高可达 13.9%；对于 Fashion MNIST，最高可达 12%，这是因为 cFAT8 采用了量化权重，在错误注入和学习过程中产生的能耗更低。同时，我们的 eFAT32 相较于 cFAT32 可节能 57.1%，因为与 cFAT32 相比，eFAT32 执行的再训练过程迭代次数更少。正如 图15(b)所示，我们的 eFAT8 进一步实现了节能：对于 MNIST，最高可达 63.1%；对于 Fashion MNIST，最高可达 62.3%。这是因为它除了采用量化权重外，还减少了再训练过程的迭代次数，从而实现了更高的节能。

**碳排放减少：** 再训练过程也带来了额外挑战，与环境问题有关，即碳排放。近期研究指出，应尽量减少神经网络训练产生的碳排放，以防自然灾害发生率不断上升（Strubell 等，2019年，2020年）。为估算神经网络训练的碳排放，Strubell 等（2019年）的研究提出了公式 5 和公式 6。在这些公式中， $CO_2e$ 表示训练期间预计产生的碳（$CO_2$）排放，它是训练期间总功耗（$p_t$）的函数。同时， $t$ 是训练时长， $p_c$ 是所有 CPU 的平均功耗， $p_r$ 是所有主存（DRAM）的平均功耗， $p_g$ 是 GPU 的平均功耗，而 $g$ 是 GPU 的数量。

$$CO_2e = 0.954 \cdot p_t \tag{5}$$

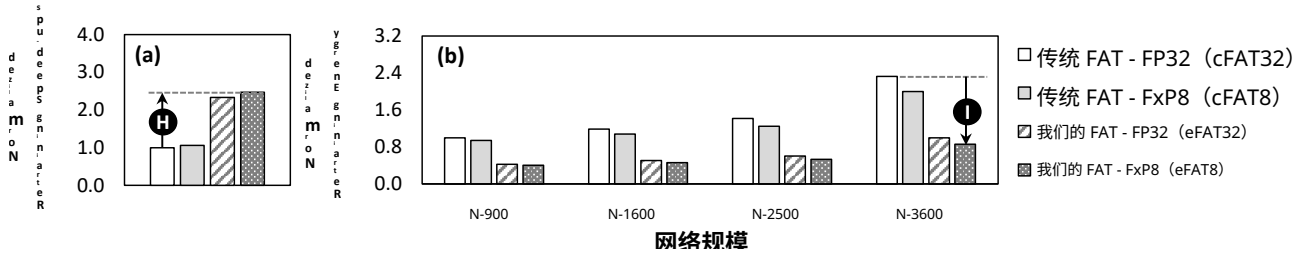



**图15**。 (a) 不同网络规模（即 N-900、N-1600、N-2500 和 N-3600）下的重新训练加速比，以及(b) MNIST 的重新训练能量；这些结果均以 900个神经元网络中采用 FP32 的传统FAT为基准进行归一化。由于这些工作负载在一个训练阶段中具有相同数量的权重和 DRAM 访问，因此 Fashion MNIST 的结果与 MNIST呈现出相似的趋势，因为二者具有相似的DRAM访问延迟和能量。此处，FxP8 同时表示“有符号 $Q1.6$”和“无符号 $Q1.7$”格式。

$$p_t = \frac{1.58 \cdot t(p_c + p_r + g \cdot p_g)}{1000} \tag{6}$$

这些方程表明，如果假设不同 FAT 技术的 $p_c$、 $p_r$、 $p_g$ 和 $g$ 相同，那么差异将来自训练时长 $t$。因此，EnforceSNN 中的高效FAT采用比传统 FAT 更少的再训练过程迭代次数，从而减少碳排放。此外，EnforceSNN 还通过降低电压近似方法降低主存（动态随机存取存储器）的运行功耗，进而进一步减少排放。

总而言之，我们的*EnforceSNN* 框架与传统的 *FAT*技术相比，能够有效缩短重新训练时间、降低重新训练能量并减少碳排放，从而更加环保。

### 6.5 进一步讨论

此前利用降压 DRAM 概念的研究主要旨在提升移动系统（Haj-Yahya 等，2020 年）、个人计算系统（Nabavi Larimi 等，2021 年；Fabrıcio Filho 等，2022 年）和服务器系统（Nabavi Larimi 等，2021 年；David 等，2011 年；Deng 等，2011 年、2012a、b）的能效。该概念也被用于降低深度神经网络（DNN）的能耗（Koppula 等，2019 年）。与 DNN 相比，脉冲神经网络（SNN）的数据表示、计算模型和学习规则各不相同，因此我们的 EnforceSNN 提供了一个不同的框架，并采用专门设计用于提升 SNN 弹性和能效的不同技术。此外，降压 DRAM 还被用于生成噪声（即来自 DRAM 错误的噪声），以保护知识产权（IP）免受 IP 窃取等安全威胁（Xu 等，2020 年）。

我们的 EnforceSNN 框架属于近似计算领域，尤其是通过电压缩放近似主存这一方向（Venkataramani et al., 2015; Xu et al., 2016; Mittal, 2016）。因此，我们的 EnforceSNN 中的一些技术也适用于脉冲神经网络以外的不同领域：(1) 降低 DRAM 电压以优化 DRAM 访问能耗，(2) 通过量化减少内存占用，以及 (3) 误差感知 DRAM 数据映射策略，以尽量降低 DRAM 错误对数据造成的负面影响。这些技术适用于容错应用，例如图像/视频处理（如数据压缩）和数据分析应用（如数据聚类）。

## 7 结论

我们提出了一种新颖的 EnforceSNN 框架，通过权重量化、误差感知 DRAM 映射、脉冲神经网络容错分析、高效的误差感知脉冲神经网络训练以及有效的脉冲神经网络模型选择，在采用基于降电压的近似 DRAM 时实现稳健且能源高效的脉冲神经网络推理。与采用精确 DRAM 的基线脉冲神经网络相比，当误码率为 $\leq 10^{-3}$ 时，我们的 EnforceSNN 在几乎无需重新训练的情况下实现了准确率无损，同时节省高达 84.9% 的 DRAM 能耗，并将 DRAM 数据吞吐量提升至最高 4.1 倍。由此，我们的工作有望为边缘人工智能等能量受限嵌入式设备实现高效的脉冲神经网络推理。

## 数据可用性声明

本研究使用了公开可用的数据集。这些数据集可在以下网站找到：http://yann.lecun.com/exdb/mnist/（MNIST），http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/（Fashion MNIST）。

## 资助

本研究部分得到了 ASPIRE AARE基金（S1561）项目“通过跨层近似计算实现极致能效”的支持。

## 参考文献

Akopyan, F., Sawada, J., Cassidy, A., Alvarez-Icaza, R., Arthur, J., Merolla, P., 等（2015）。TrueNorth：一款65毫瓦、拥有100万个神经元的可编程神经突触芯片的设计与工具流程。*IEEE*集成电路与系统计算机辅助设计汇刊34。doi:10.1109/TCAD.2015.2474396 Cao, K., Liu, Y., Meng, G., 和 Sun, Q.（2020）。边缘计算研究综述。*IEEE*开放获取 8, 85714–85728。doi:10.1109/ACCESS.2020.2991734 Chandrasekar, K.（2014）。*DRAM*的高层次功耗估算与优化。博士论文，代尔夫特理工大学 Chang, K. K., Yaˇglıkcı, A. G., Ghose, S., Agrawal, A., Chatterjee, N., Kashyap, A., 等（2017）。了解现代DRAM器件的降电压运行：实验表征、分析与机制。《*ACM*计算系统测量与分析论文集》1。doi:10.1145/3084447 Chattopadhyay, A., Prakash, A., 和 Shafique, M.（2017）。安全信息物理系统：当前趋势、工具与开放研究问题。载于*2017*年欧洲设计、自动化与测试会议暨展览。1104–1109。doi:10.23919/DATE.2017.7927154 Chen, J. 和 Ran, X.（2019）。边缘计算与深度学习：综述。*IEEE*会刊107, 1655–1674。doi:10.1109/JPROC.2019.2921977 David, H., Fallin, C., Gorbatov, E., Hanebutte, U. R., 和 Mutlu, O.（2011）。通过动态电压/频率调节进行内存功耗管理。载于第*8*届*ACM*自主计算国际会议。31–40。doi:10.1145/1998582.1998590 Deng, Q., Meisner, D., Bhattacharjee, A., Wenisch, T. F., 和 Bianchini, R.（2012a）。CoScale：协调服务器系统中的CPU与内存系统动态电压频率调节。载于*2012*年*IEEE/ACM*第*45*届年度国际微体系结构研讨会。143–154。doi:10.1109/MICRO.2012.22 Deng, Q., Meisner, D., Bhattacharjee, A., Wenisch, T. F., 和 Bianchini, R.（2012b）。MultiScale：采用多个内存控制器的内存系统动态电压频率调节。载于*2012*年*ACM/IEEE*国际低功耗电子与设计研讨会。297–302。doi:10.1145/2333660.2333727

Deng, Q., Meisner, D., Ramos, L., Wenisch, T. F., 和 Bianchini, R.（2011）。MemScale：主存的主动低功耗模式。载于第*16*届编程语言与操作系统体系结构支持国际会议。225–238。doi:10.1145/1950365.1950392 Diehl, P. 和 Cook, M.（2015）。使用脉冲时序依赖可塑性进行数字识别的无监督学习。《计算神经科学前沿》9, 99。doi:10.3389/fncom.2015.00099 Fabr´ıcio Filho, J., Felzmann, I., 和 Wanner, L.（2022）。SmartApprox：基于学习的近似存储器配置，实现能效执行。《可持续计算：信息学与系统》34, 100701。doi:10.1016/j.suscom.2022.100701 Frenkel, C., Lefebvre, M., Legat, J., 和 Bol, D.（2019）。一款采用28纳米互补金属氧化物半导体工艺、面积为0.086平方毫米、能耗为7皮焦/突触操作、具有64k个突触和256个神经元的在线学习数字脉冲神经形态处理器。《*IEEE*生物医学电路与系统汇刊》13, 145–158。doi:10.1109/TBCAS.2018.2880425 Frenkel, C., Legat, J.-D., 和 Bol, D.（2019）。MorphIC：一款采用65纳米工艺、每平方毫米具有738k个突触的四核二进制权重数字神经形态处理器，支持随机脉冲驱动在线学习。《*IEEE*生物医学电路与系统汇刊》13, 999–1010。doi:10.1109/TBCAS.2019.2928793 Gautrais, J. 和 Thorpe, S.（1998）。频率编码与时间顺序编码：一种理论方法。《生物系统》48, 57–65。doi:10.1016/S0303-2647(98)00050-1 Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., 和 Keutzer, K.（2021）。高效神经网络推理量化方法综述。*arXiv*预印本 *arXiv:2103.13630*Ghose, S., Li, T., Hajinazar, N., Cali, D. S., 和 Mutlu, O.（2019）。揭开复杂工作负载与DRAM交互的神秘面纱：一项实验研究。《*ACM*计算系统测量与分析论文集》3。doi:10.1145/3366708 Griffor, E., Greer, C., Wollman, D., 和 Burns, M.（2017）。信息物理系统框架：第*1*卷，概述 （特别出版物（NIST SP），美国国家标准与技术研究院，马里兰州盖瑟斯堡）。doi:10.6028/NIST.SP.1500-201 Gupta, S., Agrawal, A., Gopalakrishnan, K., 和 Narayanan, P.（2015）。有限数值精度下的深度学习。载于第*32*届国际机器学习会议。1737–1746 Haj-Yahya, J., Alser, M., Kim, J., Yaˇglıkc¸ı, A. G., Vijaykumar, N., Rotem, E., 等（2020）。SysScale：利用多域动态电压与频率调节提升移动处理器能效。载于*2020*年*ACM/IEEE*第*47*届年度国际计算机体系结构研讨会。227–240。doi:10. 1109/ISCA45697.2020.00029 Hazan, H., Saunders, D., Sanghavi, D. T., Siegelmann, H., 和 Kozma, R.（2018）。自组织脉冲神经网络的无监督学习。载于*2018*年国际神经网络联合会议。1–6。doi:10.1109/IJCNN.2018.8489673 Hazan, H., Saunders, D. J., Khan, H., Patel, D., Sanghavi, D. T., Siegelmann, H. T., 等（2018）。Bindsnet：面向机器学习的Python脉冲神经网络库。《神经信息学前沿》doi:10.3389/fninf.2018.00089 Hopkins, M., Mikaitis, M., Lester, D. R., 和 Furber, S.（2020）。用于求解神经常微分方程的随机舍入与低精度定点运算。《皇家学会哲学汇刊*A*辑》378, 20190052。doi:10.1098/rsta.2019.0052 Izhikevich, E. M.（2004）。皮层脉冲神经元应采用哪种模型？《*IEEE*神经网络汇刊》15, 1063–1070 Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., 等（2018）。面向高效纯整数运算推理的神经网络量化与训练。载于*IEEE*计算机视觉与模式识别会议。2704–2713

Kayser, C., Montemurro, M. A., Logothetis, N. K. 和 Panzeri, S.（2009年）。脉冲相位编码增强并稳定由空间和时间脉冲模式承载的信息。《神经元》61，597–608。doi:10.1016/j.neuron.2009.01.008 Kim, J., Patel, M., Hassan, H. 和 Mutlu, O.（2018年）。Solar-DRAM：利用局部位线的变化降低 DRAM 访问延迟。载于 *2018*年 *IEEE* 第*36*届国际计算机设计会议。282–291。doi:10.1109/ICCD.2018.00051 Kim, Y., Seshadri, V., Lee, D., Liu, J. 和 Mutlu, O.（2012年）。论利用 DRAM 中的子阵列级并行（SALP）。载于*2012*年 第*39*届年度国际计算机体系结构研讨会。368–379。doi:10.1109/ISCA.2012.6237032 Koppula, S., Orosa, L., Yaˇglıkcı, A. G., Azizi, R., Shahroodi, T., Kanellopoulos, K. 等（2019年）。EDEN：利用近似 DRAM 实现能源高效、高性能的深度神经网络推理。载于 第*52*届年度 *IEEE/ACM* 国际微体系结构研讨会。166–181。doi:10.1145/ 3352460.3358280 Kriebel, F., Rehman, S., Hanif, M. A., Khalid, F. 和 Shafique, M.（2018年）。智能信息物理系统和物联网的鲁棒性：从自适应鲁棒性方法到机器学习的可靠性与安全性。载于 *2018*年 *IEEE*计算机学会年度超大规模集成电路研讨会。581–586。doi:10.1109/ISVLSI.2018.00111 Krishnamoorthi, R.（2018年）。用于高效推理的深度卷积网络量化：白皮书。*arXiv* 预印本 *arXiv:1806.08342*Krithivasan, S., Sen, S., Venkataramani, S. 和 Raghunathan, A.（2019年）。面向能源高效脉冲神经网络的动态脉冲捆绑。载于 *2019*年 *IEEE/ACM* 国际低功耗电子与设计研讨会。1–6。doi:10.1109/ISLPED.2019.8824897 Lecun, Y., Bottou, L., Bengio, Y. 和 Haffner, P.（1998年）。应用于文档识别的基于梯度的学习。*IEEE*会刊 86，2278–2324。doi:10.1109/5.726791 Liu, F., Tang, G., Li, Y., Cai, Z., Zhang, X. 和 Zhou, T.（2019年）。边缘计算系统与工具综述。*IEEE*会刊 107，1537–1562。doi:10.1109/JPROC.2019.2920341 Micikevicius, P., Narang, S., Alben, J., Diamos, G. F., Elsen, E., Garc´ıa, D. 等（2018年）。混合精度训练。载于 第*6*届国际学习表征会议Mittal, S.（2016年）。近似计算技术综述。《*ACM*计算调查》48。doi:10.1145/2893356 Mozafari, M., Ganjtabesh, M., Nowzari-Dalini, A. 和 Masquelier, T.（2019年）。Spyketorch：高效仿真卷积脉冲神经网络，每个神经元最多产生一个脉冲。*Frontiers inNeuroscience*13，625。doi:10.3389/fnins.2019.00625 Nabavi Larimi, S. S., Salami, B., Unsal, O. S., Kestelman, A. C., Sarbazi-Azad, H. 和 Mutlu, O.（2021年）。理解通过电压缩放降低高带宽存储器的功耗与可靠性。载于 *2021*年欧洲设计、自动化与测试会议暨展览。517–522。doi:10.23919/DATE51398.2021.9474024 Olgun, A., Luna, J. G., Kanellopoulos, K., Salami, B., Hassan, H., Ergin, O. 等（2021年）。PiDRAM：一种基于现场可编程门阵列的存内处理整体端到端框架。*arXiv* 预印本 *arXiv:2111.00082*Park, S., Kim, S., Choe, H. 和 Yoon, S.（2019年）。利用脉冲突发在深度脉冲神经网络中实现快速、高效的信息传输。载于 *2019*年 第*56*届年度设计自动化会议。doi:10.1145/3316781.3317822 Park, S., Kim, S., Na, B. 和 Yoon, S.（2020年）。T2FSNN：采用首脉冲时间编码的深度脉冲神经网络。载于 第*57*届 *ACM/IEEE*设计自动化会议。1–6。doi:10.1109/DAC18072.2020.9218689

Pfeiffer, M. 和 Pfeil, T.（2018年）。脉冲神经元深度学习：机遇与挑战。*Frontiers in Neuroscience*12。doi:10.3389/fnins.2018.00774 Putra, R. V. W., Hanif, M. A. 和 Shafique, M.（2020年）。DRMap：用于卷积神经网络能效处理的通用 DRAM数据映射策略。载于 *2020*年 第*57*届 *ACM/IEEE*设计自动化会议。1–6。doi:10.1109/DAC18072.2020.9218672 Putra, R. V. W., Hanif, M. A. 和 Shafique, M.（2021年a）。ReSpawn：面向不可靠存储器的脉冲神经网络能源高效容错。载于*2021*年 *IEEE/ACM*国际计算机辅助设计会议。1–9。doi:10.1109/ICCAD51958.2021.9643524 Putra, R. V. W., Hanif, M. A. 和 Shafique, M.（2021年b）。ROMANet：面向深度神经网络加速器的细粒度复用驱动片外存储器访问管理与数据组织。*IEEE*集成电路与系统计算机辅助设计汇刊29，702–715。doi:10.1109/TVLSI.2021. 3060509 Putra, R. V. W., Hanif, M. A. 和 Shafique, M.（2022年）。SoftSNN：软错误下脉冲神经网络加速器的低成本容错。*arXiv* 预印本 *arXiv:2203.05523*Putra, R. V. W. 和 Shafique, M.（2020年）。FSpiNN：面向存储器高效和能源高效脉冲神经网络的优化框架。*IEEE*集成电路与系统计算机辅助设计汇刊39，3601–3613。doi:10.1109/TCAD.2020.3013049 Putra, R. V. W. 和 Shafique, M.（2021年a）。Q-SpiNN：脉冲神经网络量化框架。载于 *2021*年国际神经网络联合会议。1–8。doi:10.1109/IJCNN52387.2021. 9534087 Putra, R. V. W. 和 Shafique, M.（2021年b）。SpikeDyn：具备动态环境中持续学习和无监督学习能力的能源高效脉冲神经网络框架。载于 *2021*年 第*58*届 *ACM/IEEE*设计自动化会议。1057–1062。doi:10.1109/DAC18074.2021.9586281 Putra, R. V. W. 和 Shafique, M.（2022年a）。lpSpikeCon：为自主智能体实现低精度脉冲神经网络处理，从而支持高效的无监督持续学习。*arXiv* 预印本 *arXiv:2205.12295*Putra, R. V. W. 和 Shafique, M.（2022年b）。tinySNN：迈向存储器高效和能源高效的脉冲神经网络。*arXiv* 预印本 *arXiv:2206.08656*Rahimi Azghadi, M., Iannella, N., Al-Sarawi, S. F., Indiveri, G. 和 Abbott, D.（2014年）。硅上的基于脉冲的突触可塑性：设计、实现、应用与挑战。*IEEE*会刊 102，717–737。doi:10.1109/JPROC.2014.2314454 Rathi, N., Panda, P. 和 Roy, K.（2019年）。用于能源高效识别的脉冲神经网络中基于脉冲时序依赖可塑性的连接剪枝与权重量化。*IEEE*集成电路与系统计算机辅助设计汇刊38，668–677。doi:10.1109/TCAD.2018.2819366 Roy, A., Venkataramani, S., Gala, N., Sen, S., Veezhinathan, K. 和 Raghunathan, A.（2017年）。用于评估脉冲神经网络的可编程事件驱动架构。载于*2017*年 *IEEE/ACM*国际低功耗电子与设计研讨会。1–6。doi:10.1109/ISLPED.2017. 8009176 Satyanarayanan, M.（2017年）。边缘计算的兴起。*Computer* 50，30–39。doi:10.1109/MC. 2017.9 Saunders, D. J., Siegelmann, H. T., Kozma, R. 和 Ruszink´o, M.（2018年）。利用卷积脉冲神经网络进行图像块的基于脉冲时序依赖可塑性的学习。载于 *2018*年国际神经网络联合会议。1–7。doi:10.1109/IJCNN.2018.8489684

Sen, S., Venkataramani, S. 和 Raghunathan, A.（2017）。面向脉冲神经网络的近似计算。载于欧洲设计、自动化与测试会议暨展览。193–198。doi:10.23919/DATE.2017. 7926981 Shafique, M., Khalid, F. 和 Rehman, S.（2018）。智能信息物理系统的智能安全措施。载于*2018*年 第*21*届*Euromicro*数字系统设计会议。280–287。doi:10.1109/DSD. 2018.00058 Shafique, M., Marchisio, A., Putra, R. V. W. 和 Hanif, M. A.（2021）。迈向能源高效且安全的边缘人工智能：跨层框架 ICCAD 专题会议论文。载于*2021*年电气电子工程师学会/计算机协会国际计算机辅助设计会议。1–9。doi:10.1109/ICCAD51958.2021.9643539 Shi, W., Cao, J., Zhang, Q., Li, Y. 和 Xu, L.（2016）。边缘计算：愿景与挑战。*IEEE*物联网期刊 3, 637–646。doi:10.1109/JIOT.2016.2579198 Strubell, E., Ganesh, A. 和 McCallum, A.（2019）。自然语言处理深度学习的能量与政策考量。载于计算语言学协会第*57*届年会论文集。3645–3650。doi:10.18653/v1/P19-1355 Strubell, E., Ganesh, A. 和 McCallum, A.（2020）。现代深度学习研究的能量与政策考量。*AAAI*人工智能大会论文集34, 13693–13696。doi:10.1609/aaai. v34i09.7123 Sze, V., Chen, Y., Yang, T. 和 Emer, J. S.（2017）。深度神经网络的高效处理：教程与综述。*IEEE*会刊 105, 2295–2329。doi:10.1109/JPROC.2017.2761740 Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T. 和 Maida, A.（2019）。脉冲神经网络中的深度学习。《神经网络》111, 47–63。doi:10.1016/j.neunet.2018.12.002 Thorpe, S. 和 Gautrais, J.（1998）。秩序编码。载于《计算神经科学》。113–118。doi:10.1007/978-1-4615-4831-7 19 van Baalen, M., Kahne, B., Mahurin, E., Kuzmin, A., Skliar, A., Nagel, M. 等（2022）。模拟量化，切实节省功耗。载于电气电子工程师学会/计算机视觉基金会计算机视觉与模式识别会议论文集。2757–2761 Venkataramani, S., Chakradhar, S. T., Roy, K. 和 Raghunathan, A.（2015）。近似计算与计算效率追求。载于*2015*年 第*52*届*ACM/EDAC/IEEE*设计自动化会议。1–6。doi:10.1145/2744769.2744904 Xu, Q., Mytkowicz, T. 和 Kim, N. S.（2016）。近似计算：综述。*IEEE*《设计与测试》33, 8–22。doi:10.1109/MDAT.2015.2505723 Xu, Q., Tanvir Arafin, M. 和 Qu, G.（2020）。MIDAS：使用近似存储系统的模型反演防御。载于*2020*年亚洲硬件安全与信任研讨会。1–4。doi:10.1109/ AsianHOST51057.2020.9358254 Yu, W., Liang, F., He, X., Hatcher, W. G., Lu, C., Lin, J. 等（2018）。物联网边缘计算综述。*IEEE*开放获取6, 6900–6919。doi:10.1109/ACCESS.2017.2778504