用于降低深度神经网络加速器链路功耗的基于“1”位计数的排序单元

Ruichi Han、Yizhi Chen、Tong Lei、Jordi Altayo Gonzalez、Ahmed Hemani 电子与嵌入式系统系，瑞典皇家理工学院，瑞典斯德哥尔摩{ruichi, yizhic, tonglei, jordiag, hemani}@kth.se

arXiv:2601.14087v1 [cs.AR] 20 Jan 2026

摘要—**互连功耗仍是深度神经网络（DNN）加速器的一大瓶颈。根据“1”位计数对数据进行排序，能够通过降低开关活动来缓解这一问题，但实用的硬件排序实现仍未得到充分探索。本文提出了一种面向卷积神经网络（CNN）优化的无比较排序单元硬件实现方案。通过采用近似计算，将总体计数归入粗粒度桶，我们的设计在保留数据重排序带来的链路功耗优势的同时，缩减了硬件面积。与精确实现相比，我们的近似排序单元在实现 19.50% BT 降低（精确实现为 20.42%）的同时，最多可缩减 35.4% 的面积。**

索引词—**基于“1”位计数的排序、近似计算、位翻转减少、链路功耗**

I. 引言

深度神经网络（DNN）加速器是现代计算的关键组成部分 [1] [2]。尽管深度神经网络加速器执行大量算术运算，近期研究表明，数据移动主导着功耗 [3] [4] [5]。

随着模型规模和并行度不断扩大，片上互连和片上网络结构消耗的芯片功耗占比日益增加，尤其是在数据密集型工作负载中 [1] [5]。在基于片上网络的架构中，物理链路占能量预算的很大一部分 [4] [5]。由于动态功耗本质上由链路上的开关活动决定，且每次位翻转（BT）都会产生电容充电成本，因此减少 BT 是提升加速器能效的有效手段 [3] [4]。

为抑制接口开关活动，已有多种位翻转减少编码方案 [2] [6]。标准编码算法通常无法适应神经计算独特的流量模式 [1] [4] [5] [6]。近期研究利用累加顺序不敏感性，通过“1”位计数或总体计数（人口计数）排序来最大限度地减少开关活动 [7]。然而，这些大多停留在理论层面的方案缺乏实际硬件评估，无法证明排序机制所带来的面积与功耗开销是合理的 [1] [7]。

据我们所知，这是首个采用近似排序、面向深度神经网络的硬件人口计数排序单元实现。为保持术语一致，尽管这两个术语含义相同，“人口计数”指硬件单元，而““1”位计数”用于非硬件讨论。我们的贡献总结如下：

1) **硬件实现**：不同于以往基于模拟的研究，我们提出了一种硬件实现无比较器总体计数排序，并将其与其他总体计数排序实现进行比较。2) **近似计算优化**：我们不仅实现了精确排序单元，还将其扩展为近似排序，以降低面积成本。3) **全面功耗分析**：我们通过布局后功耗分析，研究排序对链路功耗和计算单元功耗的影响。4) **深度神经网络层的工作负载**：我们在卷积与池化工作负载下评估了设计，并采用与 LeNet 前两层相同的配置。

本文其余部分的组织结构如下：第二节回顾相关工作。第三节介绍我们的无比较排序单元架构和近似计算技术。第四节介绍实验设置与结果。第五节总结全文。

II. 相关工作

深度神经网络加速器中片上互连的功耗不断上升，因此需要采取策略来降低链接上的动态切换开销 [5]。Eyerissv2 所采用的编码技术等方法通过信号级转换，减少数据传输过程中的比特翻转 [8]。然而，这类编码方法会带来编码/解码开销，影响总体硬件效率 [9]。

深度神经网络累加运算具有顺序无关性，因此可以采用数据重排序策略来降低切换活动 [4]。Chen 等 [7] 证明，基于“1”位计数的排序可以减少链路位翻转，但会带来不可忽视的硬件开销。这促使我们在保留功耗优势的同时，着重降低排序单元面积 [3]。

传统排序网络（如 Batcher双调排序网络） [10] 比较器逻辑密集。常数时间方法（如竞争排序网络（CSN）） [11] [12] 虽然提升了速度，但需要多出 80% 的逻辑单元。无比较器架构通过消除显式比较器逻辑，提供了更高面积效率的基础。AdbelHafeez 等人 [13] 提出了一种线性时间排序器，可将输入映射到硬件友好结构中，从而实现低缓冲开销和极低复杂度。后续改进包括用于控制的有限状态机集成 [14] 以及用于降低延迟的双向数据通路 [15]。OnSort [16] 则利用稀疏感知元数据过滤器解决了可扩展性问题。

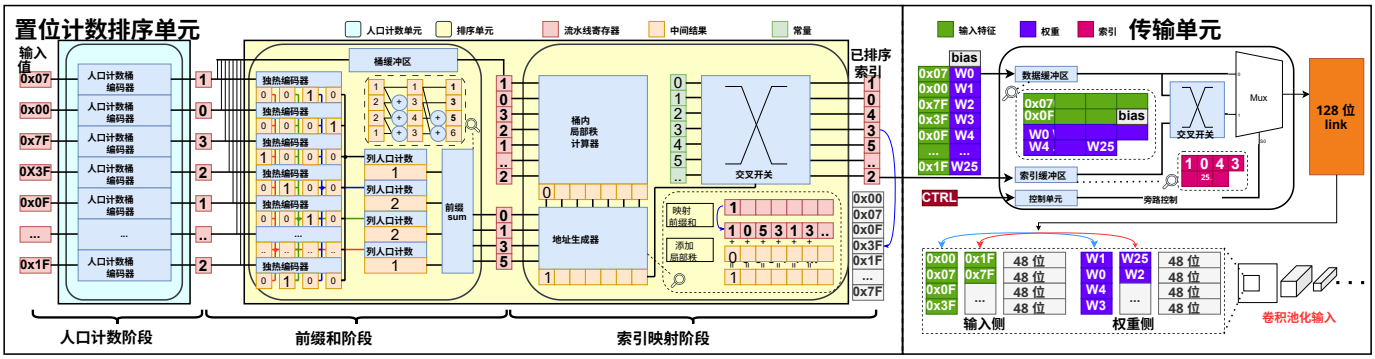

图 1：置位计数排序单元的架构及数据流示例。

III. 方法与设计

近似计算可提升多个领域的硬件效率。近似乘法器 [17]面向乘加运算，以计算精度换取功耗降低；而粗粒度分桶近似 [18] 面向排序运算，以放弃完美排序换取更高效率。后一种方法启发了我们对近似计算的应用。

本研究旨在通过最大限度减少互连链路上的比特翻转（BT），同时将硬件实现的面积开销控制在较低水平，从而降低功耗。

A. 基线无比较计数排序单元

我们工作的基线是采用无比较设计的精确计数排序单元（ACC-PSU），改编自 Yang 等人 [19]。该架构由人口计数单元和排序单元组成，分为三个阶段：人口计数阶段、前缀和阶段以及索引映射阶段，如图 1 所示。人口计数单元使用 4 位查找表（LUT）计算汉明重量，并通过加法器汇总查找表的输出，从而得出每个输入元素的“1”位计数。排序单元将“1”位计数编码为独热表示，生成频率直方图，通过前缀和计算各个“1”位计数值对应的起始地址，并将索引分散到排序输出中。随后，传输单元根据排序后的索引重新排列输入数据，并通过链路转发数据。

B. 置位计数排序单元内部的近似计算

为减少硬件开销，我们提出了一种近似计数排序单元（APP-PSU）。该设计将近似技术集成到人口计数模块中。其关键创新在于将精确的总体计数归入粗粒度桶，从而缩短数据通路宽度。

1) 近似排序的有效性：卷积累加对顺序不敏感，因此近似排序能够通过避免大幅切换，保留精确排序的大部分优势，并实现几乎相同的 BT 降低。第四节中的 Python 仿真验证了这一结论。

2) 近似设计：该近似策略通过确定性映射，将精确的“1”位计数归入 k 个粗粒度桶。对于 W位输入，每个精确的“1”位计数值 $P_{exact} \in [0, W]$ 都会被分配到 k 个桶中，从而生成紧凑的桶索引，仅需$log_2(k)$位（例如，$k = 4$ 个桶对应一个2位索引）。排序阶段不再保留完整的“1”位计数范围，而只传递这些桶索引，将值空间从 $W + 1$ 种可能的“1”位计数压缩至 k 个桶。

例如，对于 $W = 8$ 位输入和 $k = 4$ 个桶，该映射将 $\{0, 1, 2\} \rightarrow Bucket0$、$\{3, 4\} \rightarrow Bucket1$、$\{5, 6\} \rightarrow Bucket2$ 和 $\{7, 8\} \rightarrow Bucket3$ 分配到相应桶中。因此，具有精确 “1” 位计数 $\{4, 1, 7, 5, 3, 5\}$ 的输入序列将映射为桶索引 $\{1, 0, 3, 2, 1, 2\}$。

3) 硬件实现与架构：如图1所示，APP-PSU 架构保留了与 ACC-PSU 相同的数据流，同时在人口计数分桶编码器中引入映射查找表以进行近似。综合过程中，编译器会消除不影响最终桶索引的逻辑路径，因此综合网表是一个经过简化的近似“1”位计数电路，可直接生成桶索引。

面积缩减主要来自桶数量的减少。桶数量决定了整个排序单元的数据通路宽度——所有排序逻辑的规模都与桶数量成正比，而非完整的“1”位计数范围。对于 $W = 8$ 位数据，ACC-PSU 需要 $W + 1 = 9$ 个桶来涵盖所有可能的“1”位计数值，而 APP-PSU 则将相邻的“1”位计数值合并到 $k = 4$ 个桶中。

IV. 实验

我们评估了四种配置：未优化基线、列优先排序、ACC 排序，以及采用 $k = 4$ 个桶的 APP 排序。所有实验均使用 8 位定点表示。我们还将本文的 PSU 与其他排序实现进行了比较。在硬件评估中，ACC-PSU 和 APP-PSU 部署在我们的平台上，而基线则通过旁路路径实现。

A. 软件仿真实验结果

表 I：不同排序策略下的位翻转

排序策略	每个 128 位 flit 的位转换次数			
	输入	权重	总体	减少量
未优化	31.035	32.036	63.072	-
列优先	26.004	28.007	54.011	14.366%
ACC 排序	22.333	28.013	50.346	20.177%
APP 排序	22.887	28.009	50.896	**19.305%**

我们使用 Python 对 128 位链路上的数据包传输进行模拟。共发送 100000 个数据包，每个数据包由 4 个流控单元组成，其中输入和权重均为随机值。如

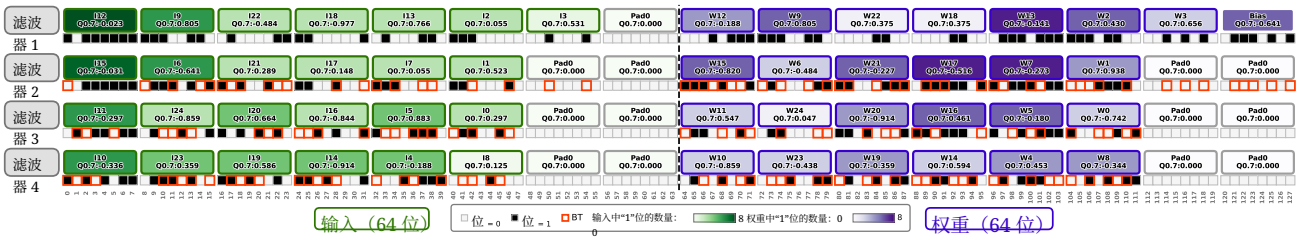

图 2：经过我们的 APP-PSU 后，在 128 位链路上排序的示例。

表 I 所示，与未优化基线相比，采用列优先排序可使总体比特翻转减少 14.366%。此外，应用基于输入“1”比特计数的排序策略后，输入侧比特翻转进一步减少，使总体比特翻转分别减少 20.177% 和 19.305%（对应 ACC 排序和 APP 排序）。

B. DNN 工作负载实验结果

图 3 展示了我们的评估平台，该平台由一个数据分配单元和 16 个处理单元（PE）组成，用于实现 LeNet-5 的第一个卷积层和池化层。分配单元向每个处理单元提供输入、权重和偏置。在分配单元内部，排序单元生成排序后的索引，传输单元根据这些索引对数据进行置换，然后将其转发给处理单元。

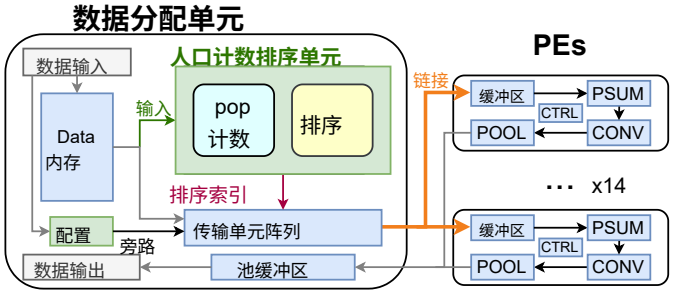

图 3：用于卷积层和池化层、带有排序单元和处理单元的平台。*1) APP-PSU* 的波形验证：我们展示了四种具有代表性的输入模式的波形，以说明 APP-PSU 的行为。如图 4 所示，来自“1”位计数较高的桶的输出索引排在来自计数较低的桶的输出索引之后。对于（1）全 1 和（2）全 0 模式，输出索引按升序排列。对于（3）“1”位计数从 8 递减至 0 的重复模式，图中显示的最后八个索引清楚地展示了 APP 排序算法的行为。对于（4）随机输入模式，也观察到了相同的排序效果。

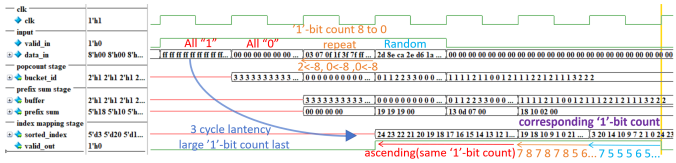

图 4：APP-PSU 的 QuestaSim 波形。

2) 有序流控单元示例：图 2 展示了 APP-PSU 处理单个数据包时 128 位链路传输的快照。在输入侧，传输值的“1”位计数总体呈下降趋势；例如，“1”位计数为 7 的数据排在计数为 5 或 6 的数据之前，而权重侧仍然是随机的。这种排序可减少人口计数的剧烈变化，并有助于使链路上的 BT 梯度保持较小。

3) 排序单元面积缩减：我们使用面向 22 纳米工艺节点、目标频率为 500 MHz 的商用 EDA 工具，对四种人口计数排序单元设计进行了综合——其中一种基于 Bitonic 网络，[10]，一种基于 CSN，[11] [12]，ACC-PSU，以及 APP-PSU——所有设计均采用相同的流水线深度实现。评估涵盖两种卷积核尺寸，5×5 和 7×7。如图 5 所示，对于两种内核尺寸，APP-PSU 的面积均为 $2193\mu m^2$ 和 $6928\mu m^2$，低于所有其他架构。内核尺寸为 25 时，该近似方法使人口计数单元和排序单元的面积分别缩减 24.9% 和 36.7%，总体减少量达到 35.4%。

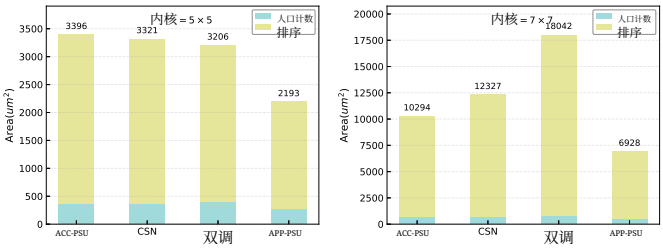

图 5：不同排序单元设计的面积细分。

4) 功耗结果：我们使用商用 EDA 工具，依据回标注开关活动评估布局后功耗。将一组包含 100 个卷积核的测试向量应用于三种配置：未优化基线、ACC 排序和 APP 排序。所有报告的百分比均为平均值。在处理单元层级，ACC-PSU 可将功耗降低 4.98%，而 APP-PSU 的减少量为 4.58%。图 6 展示了 APP 排序与基线的处理单元功耗分解。

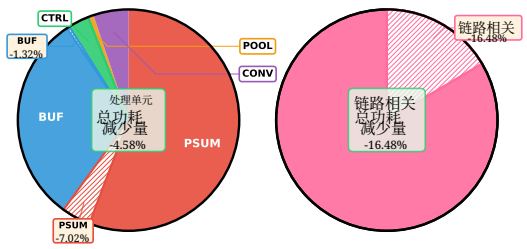

图 6：已实现的非链路功耗和链路功耗减少量细分。

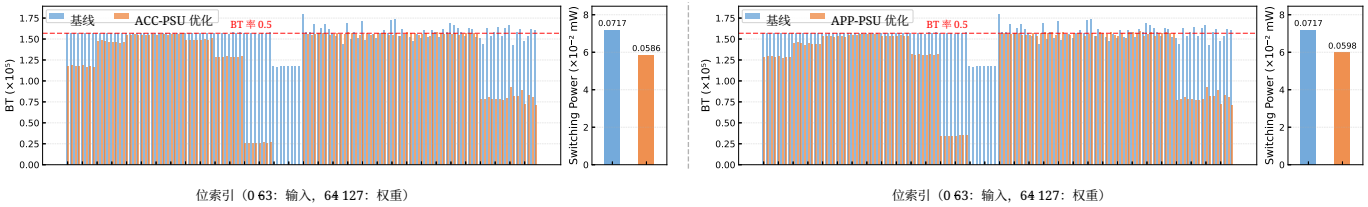

图 7：比特翻转降低量和链路相关功耗降低量（左：ACC-PSU；右：APP-PSU）。

提取传输寄存器的开关功耗，作为链路功耗的替代指标，称为链路相关功耗。如图 7 所示，ACC-PSU 可将链路相关功耗降低 18.27%，并将链路位翻转降低 20.42%；而 APP-PSU 的相应降幅分别为 16.48% 和 19.50%。

尽管 APP-PSU 带来的功耗节省略低于 ACC-PSU，但其功耗开销仅为 1.43 mW，而 ACC-PSU 为 2.28 mW，功耗开销降低了 37.3%。

C. 讨论与局限性

1. 比特翻转降低效果与未优化基线相比，我们提出的 APP-PSU 可将链接中的比特翻转降低 19.50%，达到 ACC-PSU 所实现的 20.42% 比特翻转降低量的 95.5%。

2. APP-PSU 的硬件效率APP-PSU 可将链路相关功耗降低 16.48%，相当于 ACC-PSU 所实现的链路功耗节省量的 90.2%。与此同时，它将面积开销降低 35.4%，并将功耗开销降低 37.3%，从而显著降低了硬件成本。

3. 端到端功耗与跳数 我们简化的平台中，每个数据包仅进行单跳传输，在增加 1.43 mW 功耗开销的同时，实现了 16.48% 的链路功耗降低。对于具有更长多跳互连的全规模加速器，随着每次路由器间传输都累积 BT 降低带来的收益，我们的方法应能随着跳数增加而实现更大的节能效果。由于本研究聚焦于排序单元，因此该平台仅作为示例。

4. 局限性作为未来工作，我们计划将分析从单个卷积层扩展到更复杂的神经网络，例如残差网络和 Transformer，并在具有详细 NoC 实现的加速器平台中评估多跳路径上的 BT 降低效果和功耗节省。

V. 结论

我们展示了用于部分 LeNet 的人口计数排序单元的硬件实现。通过引入近似计算来简化逻辑，面积和功耗均有所降低。22 纳米工艺下的布局后分析表明，与精确基线相比，我们的 APP-PSU 实现了 35.4% 的面积缩减和 37.3% 的功耗降低，同时保持了 95.5% 的 BT 降低效率（19.5% 对比 20.4%）。采用回标注开关活动进行的布局后功耗分析证实，链路相关功耗降低了 16.48%，处理单元功耗降低了 4.58%。未来工作将扩展至更大型号（如残差网络、Transformer），并将我们的方法集成到完整的基于片上网络的平台中，以量化端到端收益

参考文献

[1] C. Silvano、D. Cattaneo、S. Cherubin 和 G. Agosta，《面向异构高性能计算平台的深度学习硬件加速器综述》，*ACM ComputingSurveys*，第 57 卷，第 11 期，第 1–52 页，2025 年 6 月。[2] A. Samanta、I. Hatai 和 A. K. Mal，《面向边缘设备的深度学习硬件加速器设计综述》，*Wireless Personal Communications*，第 137 卷，第 3 期，第 1715–1760 页，2024 年 8 月。[3] G. Armeniakos、G. Sylaidos、G. Zervakis、D. Soudris 和 J. Henkel，《深度神经网络加速器的硬件近似技术：综述》，*ACM Computing Surveys*，第 55 卷，第 4 期，第 1–36 页，2022 年。[4] Y.-H. Chen、T. Krishna、J. S. Emer 和 V. Sze，《Eyeriss：一种面向深度卷积神经网络的节能可重构加速器》，*IEEE Journal of Solid-State Circuits*，第 52 卷，第 1 期，第 127–138 页，2017 年。[5] G. Krishnan、S. K. Mandal、C. Chakrabarti、J.-s. Seo、U. Y. Ogras 和 Y. Cao，《片上互连对深度神经网络存内加速的影响》，*ACM Journal on Emerging Technologies in ComputingSystems*，第 17 卷，第 2 期，第 1–27 页，2021 年。[6] S. Jang、S. Jeon、K. Shin、D. Lee、H. Chi、W. Shin、C. Pyo、J. Kim 和 D. Jeon，《Witch：用于降低高带宽存储器串扰的加权编码方案》，载于第 *30* 届亚洲及南太平洋设计自动化会议（*ASP-DAC*）论文集，2025 年，第 945–951 页。[7] Y. Chen、J. Li、W. Zhu 和 Z. Lu，《通过数据传输排序降低基于片上网络的深度神经网络加速器中的位翻转》，2025 年。[8] Y.-H. Chen、T.-J. Yang、J. Emer 和 V. Sze，《Eyeriss v2：面向移动设备中新兴深度神经网络的灵活加速器》，*IEEE Journal on Emergingand Selected Topics in Circuits and Systems*，第 9 卷，第 2 期，第 292–308 页，2019 年。[9] F. Liu、N. Yang、Z. Wang、X. Zhu、H. Yao、X. Xiong、L. Jiang 和 H. Guan，《Inspire：通过硬件友好的索引对编码加速深度神经网络》，载于第 *61* 届 *ACM/IEEE* 设计自动化会议（*DAC*）论文集，2024 年，第 1–6 页。[10] K. E. Batcher，《排序网络及其应用》，载于*1968* 年 *4* 月 *30* 日至 *5* 月 *2* 日春季联合计算机会议论文集，AFIPS '68（Spring）系列。美国纽约州纽约市：计算机协会，1968 年，第 307–314 页。 [在线]。获取地址：https://doi.org/ 10 1145 1468075 1468121 。/。[11] A. A. Fairouz、J. M. Aljuraidan 和 A. Mohammed，《竞赛式排序网络（CSN）：基于硬件的排序运算框架》，*IEEE Transactions on Computers*，第 1–14 页，2025 年。[12] P. T. Jelodari、M. P. Kordasiabi、S. Sheikhaei 和 B. Forouzandeh，《一种用于少量输入、具有硬件实现的 O(1) 时间复杂度排序网络》，*Microprocessors and Microsystems*，第 77 卷，文章编号 103203，2020 年 9 月。[13] S. Abdel-Hafeez 和 A. Gordon-Ross，《一种高效的 o(n) 无比较排序算法》，*IEEE Transactions on Very Large Scale Integration (VLSI) Systems*，第 25 卷，第 6 期，第 1930–1942 页，2017 年。[14] T. A. S. Bhargav 和 E. Prabhu，《用于写入评估阶段和读取排序阶段的节能、面积高效有限状态机与无比较排序算法》，载于信号处理与智能识别系统进展，S. M. Thampi、O. Marques、S. Krishnan、K.-C. Li、D. Ciuonzo 和 M. H. Kolekar 编。新加坡：Springer Singapore，2019 年，第 433–442 页。[15] W.-T. Chen、R.-D. Chen、P.-Y. Chen 和 Y.-C. Hsiao，《准无比较排序算法的高性能双向架构》，*IEEE Transactions on Circuits and Systems I: Regular Papers*，第 68 卷，第 4 期，第 1493–1506 页，2021 年。[16] M. Gao、J. Jiang、S. Lei、H. Wu、J. Chen 和 Y. Liu，《OnSort：一种面向大规模数据集、带并行预取和稀疏感知机制的 O(n) 无比较排序器》，*IEEE Transactions on Circuits and Systems II: Express Briefs*，第 72 卷，第 7 期，第 933–937 页，2025 年。

[17] L. Du、L. Ni、X. Liu、G. Peng、K. Li、W. Mao 和 H. Yu，《一种采用均值误差最小化近似有符号乘法器的低功耗深度神经网络加速器》，*IEEE* 开放电路与系统期刊，第 5 卷，第 57–68 页，2024 年。[18] J. Xiao、H. Chen 和 J. Sun，《一种利用图形处理器的高性能近似排序算法》，载于*2015*年国际计算机科学与智能通信会议论文集。Atlantis Press，2015 年 7 月，第 121–124 页。 [在线]。获取地址：https: // doi.org/ 10 2991 15 2015 28 。/csic- 。。J. Yang，《探索一种基于 O(N) 1 位的无比较近似排序》，2025 年。[19]