arXiv:2603.29975v1 [cs.DC] 31 Mar 2026

第一性原理电子结构计算 GPU 加速的精度仿真方法

HANG LIU、JUNJIE LI 和 YINZHI WANG，美国德克萨斯州奥斯汀市德克萨斯高级计算中心、德克萨斯大学奥斯汀分校

NIRAJ K. NEPAL 和 YANG WANG，美国宾夕法尼亚州匹兹堡市匹兹堡计算中心、卡内基梅隆大学

本研究探讨了如何利用基于 INT8 的模拟，在现代 GPU 架构上加速传统的基于 FP64 的高性能计算工作负载。借助适用于缓存一致性统一内存架构的 SCILIB-Accel 自动 BLAS 卸载工具，我们无需修改代码，便在 MuST 套件的 LSMS CPU 应用程序中模拟 FP64 矩阵乘法。我们发现，精度取决于算术精度和算子的特性，可通过可调精度模拟加以处理。与传统的混合精度方法不同，该方法在优化硬件利用率的同时保留原始算法。我们展示了同时提高精度和性能的潜力。本研究凸显了 AI 驱动的硬件变革高性能计算的潜力，并倡导在未来的科学计算中采用自适应精度策略。

CCS 概念：• **通用与参考 → 性能**；• **计算数学 → 数学软件性能**；
• **软件及其工程 → 软件性能**。

附加关键词　词语和短语：通用矩阵乘法、基础线性代数子程序、自动卸载、FP64 模拟、混合精度、Ozaki 方案、第一性原理，电子结构　计算

ACM 参考格式：
Hang Liu、Junjie Li、Yinzhi Wang、Niraj K. Nepal 和 Yang Wang。2026 年。《精度仿真方法：GPU 加速第一性原理电子结构计算》。载于请填写版权确认电子邮件中的正确会议名称（*PEARC '26*）。ACM，纽约，美国，6 页。
https://doi.org/XXXXXXX.XXXXXXX

1 引言

人工智能（AI）的快速发展推动了专用硬件的研发，例如配备张量核心的英伟达 GPU、配备矩阵核心的 AMD GPU 以及各种张量处理器，它们凭借高吞吐量的低精度算术单元[1]加速机器学习计算。这些处理器利用低精度/混合精度浮点格式（例如 FP16、BF16）和低位宽整数单元（例如 INT8、INT4），以提升性能、降低功耗并最大限度减少内存占用。尽管这些架构创新显著提升了人工智能训练和推理的性能，但它们在加速传统高性能计算工作负载方面的潜力仍然巨大，却尚未得到充分了解。这些工作负载历来依赖中央处理器和双精度浮点（FP64）计算，用于科学模拟、数值建模及其他计算密集型任务；要充分释放其潜力，仍需进一步研究与开发。

作者联系方式：Hang Liu，hliu@tacc.utexas.edu；Junjie Li，nicejunjie@gmail.com；Yinzhi Wang，yinzhi.wang.cug@gmail.com，德克萨斯高级计算中心，德克萨斯大学奥斯汀分校，美国德克萨斯州奥斯汀；Niraj K. Nepal，nnepal@psc.edu；Yang Wang，ywg@psc.edu，匹兹堡计算中心，卡内基梅隆大学，美国宾夕法尼亚州匹兹堡。

允许为个人或课堂使用而免费制作本作品的全部或部分数字或纸质副本，前提是副本不得用于营利或商业目的，也不得为此而制作或分发，并且副本须附有本声明及首页上的完整引文。必须尊重本作品中归其他作者所有的组成部分的版权。注明出处后，可以摘录。若要以其他方式复制、再版、发布到服务器或分发至列表，须事先获得具体许可和/或支付费用。请向 permissions@acm.org 申请许可。© 2026 年 版权归所有者/作者所有。出版权已授权给 ACM。稿件已提交至 ACM

FP64 一直是传统高性能计算的标准，可确保气候建模、量子化学模拟和计算流体力学等模拟所需的数值精度。然而，当前硬件市场深受人工智能热潮影响，市场强烈青睐低精度数据格式，导致近期一些图形处理器（如英伟达的 Blackwell 系列和 Rubin [14]）的 FP64 能力有所降低，甚至几乎消失。因此，服务于广泛科研群体的学术高性能计算中心必须在以人工智能为中心的硬件与传统高性能计算硬件之间权衡选择。这一现状为重新审视高性能计算策略、发挥低精度数据类型的巨大潜力提供了契机。通过将适合的工作负载转移到图形处理器上，并利用 INT8 等较低位宽的数据类型，可以在精度可接受的情况下模拟 FP64 运算，从而推动高性能计算与人工智能硬件的融合，并最大限度地提高硬件利用率。

本研究探讨了将基于 INT8 的 FP64 通用矩阵乘法仿真与自动基础线性代数子程序卸载 [9] 集成到传统的、基于密度泛函理论（DFT）的第一性原理电子结构计算软件包 MuST [11]中的方法。通过改造 Ozaki 框架，我们利用现代图形处理器上的高吞吐量 INT8 算术单元，在不同数值保真度水平下实现可调精度。研究结果表明，MuST 的底层算法天然兼容通用矩阵乘法仿真，既能系统地控制物理精度，又可显著提升性能。本研究展示了在现代 GPU 架构上通过自动卸载和可调精度模拟，在不牺牲吞吐量的前提下保持较高数值精度的潜力。

2 相关工作

2.1 自动基础线性代数子程序卸载的相关工作

传统高性能计算代码实现 GPU 加速的主要局限，在于难以改造旧有 CPU 应用程序。手动移植劳动量大，对大多数科学研究群体而言并不切实际。本研究考察的 MuST 就是一个此类案例：它是原生 CPU 代码，要实现高性能 GPU 移植极具挑战性。

多项研究致力于将基础线性代数子程序调用从 CPU 自动卸载到图形处理器。Cray LIBSCI_ACC [3, 16]开创了这一方法，IBM 的 ESSL [6] 和英伟达的 NVBLAS [12] 也采用了类似的方式。这些工具将 CPU 基础线性代数子程序调用替换为 GPU 基础线性代数子程序调用，并自动处理数据移动和卸载决策。然而，它们面向传统 GPU 架构，每次卸载基础线性代数子程序调用时，数据都要在 CPU 和 GPU 之间传输，导致开销高得难以接受，并使实际高性能计算应用的性能显著下降 [7, 9]。

Li 等人 [7–9], 带来了一项重大进展，他们开发了高性能自动基础线性代数子程序卸载工具 SCILIB-Accel。该工具提出了一种新颖的数据移动策略，利用近期 CPU–GPU 系统中的缓存一致性，例如 NVIDIA Grace-Hopper，以及采用缓存一致性 Infinity Fabric 的 AMD GPU。该数据移动策略旨在最大限度地提高常见基于矩阵运算的科学工作负载中的数据复用率。在真实的高性能计算应用中进行评估后，该工具无需修改代码即可显著提升性能，克服了以往方法的局限性，并为高性能计算中的基础线性代数子程序卸载树立了新标准。

2.2 整数矩阵乘法单元上的 GEMM 仿真

浮点仿真的研究并非新鲜事。最广为人知的仿真方式，是使用双精度仿真四倍精度 [4]，以满足那些精度要求超出 FP64 的科学应用需求。近年来，陆续发表了一系列关于 Ozaki 方案的论文并推出了相应实现，探索如何利用较低精度的硬件（例如 INT8/FP8 张量核心）仿真高精度矩阵乘法（例如 FP64/双精度），同时保持较高精度的计算准确性。这是一种通用的矩阵乘法方案：将矩阵分解为多个低精度分量。稿件已提交至 ACM

高精度矩阵分解为多个低精度分量，执行多次矩阵乘法，再重构结果。原始的 Ozaki-I 根据高精度输入矩阵的有效位和指数对齐情况，将其切分为低精度分量，然后对这些分量执行低精度矩阵乘法，并以更高精度累加结果[15, 18]。其开源实现 ozIMMU 及改进版本 ozIMMU-H 可在 [5]和 [22]获取。NVHPC/26.1 SDK 中的 CUDA13 包含基于 Ozaki-I 的英伟达实现 [13]。后来开发的 Ozaki-II 利用中国剩余定理（CRT）仿真矩阵乘法 [17]。它将浮点矩阵转换为整数，使用较小的两两互质模数执行多次矩阵乘法，并利用中国剩余定理重构最终结果。研究普遍证明，在配备更快 INT8/FP8 张量核心的现代硬件（例如 NVIDIA GB200）上，Ozaki-II 的性能通常优于 Ozaki-I，能效更高，精度控制也更好。其开源实现 GEMMul8 可在 [23]获取。

3 实验

3.1 实现

为了在 GPU 的 INT8 张量核心上运行 CPU 代码，我们的实验使用了两种开源工具：SCILIB-Accel [7–9] 和 GEMMul8 [23]。GEMMul8 挂钩模式通过动态链接器将 cuBLAS GEMM 函数调用重定向到其 GEMMul8 实现，从而实现基于 INT8 的 GEMM 仿真。SCILIB-Accel 利用 PEAK 性能分析框架 [20], , 采用 DBI 透明地拦截基础线性代数子程序调用，并通过调用 cuBLAS 自动将其卸载到 GPU。透明的 DBI 实现经设计可使这两种工具协同工作，只需设置一个 LD_PRELOAD 即可实现自动卸载。这种方法无需修改代码或重新编译，即可在任何 GEMM 密集型 CPU 程序上实现可调精度仿真。GEMM 仿真的精度主要由环境变量 [23]设置的模数控制。我们的实验还使用了 cuda13 中的 Ozaki-I 实现。它提供了环境变量，可在不修改代码的情况下控制仿真策略和精度。使用该实现只需通过 cuda13 [13]构建 SCILIB-Accel。

3.2 MuST 实验

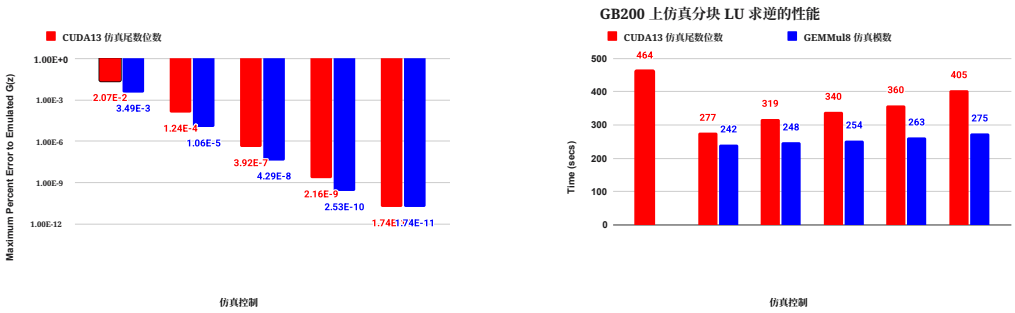
MuST（多重散射理论） [11] 是一款用于电子结构计算的第一性原理软件包，采用格林函数方法求解 Kohn–Sham 方程。MuST 中实现的局域自洽多重散射（LSMS）方法 [21]专为大规模系统设计，其计算复杂度随原子数线性增长（$O(N)$）。LSMS 代码曾两次获得戈登·贝尔奖，分别于 1998 年和 2009 年获奖，以表彰其在高性能计算领域的突破性性能 [2, 19]。LSMS 方法高度依赖基础线性代数子程序运算，主要是 ZGEMM 和 ZTRSM，这两项运算通常占 CPU 运行时间的 80% 以上。

由于 MuST 中 LSMS 方法的主要计算瓶颈是在复能量轮廓上求多重散射矩阵的基于 LU 分解的求逆，而其 ZGEMM 运算强度很高，因此它是评估模拟精度的理想测试平台。在多重散射形式中，Kohn-Sham 方程的格林函数 $G(\mathbf{r},\ \mathbf{r};z)$，$\left[-\nabla^2 + V_{\text{eff}}(\mathbf{r})\right]\psi(\mathbf{r};z) = z\,\psi(\mathbf{r};\ z)$，通过该矩阵求逆确定。随后，沿着上半平面内一条始于价带下方、终于费米能的复平面轮廓 C 对格林函数积分，得到价电子密度：$\rho(\mathbf{r}) = -\frac{1}{\pi}\mathrm{Im}\int_C G(\mathbf{r},\ \mathbf{r};\ z)\ dz$。实际中，通常使用 ~30 点高斯求积，沿半圆路径进行轮廓积分。由于电子密度及所有其他可观测量的精度从根本上取决于格林函数的精度，我们研究了能量相关的、按原子体积（Ω）积分的格林函数的百分比误差

$G(z) = \int_\Omega G(\mathbf{r}, \mathbf{r}; z)\, d\mathbf{r}$ 在 CUDA13 和 GEMMul8 的各种仿真精度模式下进行测试，涵盖尾数位数分别为 31、39、47、55、63 的 cuda13 eager 策略，以及模数分别为 10、12、14、16、18 的 GEMMul8，并以原生 FP64 GEMM 作为基准真值。这是评估不同仿真模式保真度时最灵敏、最关键的指标。随后，我们监测了三项主要物理可观测量：每原子全电子总能量 E_{tot}、局域磁矩 μ 和局部净电子电荷 δQ。这些量都是自旋向上和向下态电子密度的泛函。我们选用了 MuST 发行版中的 $FeNi_3$ 基准案例。该案例使用 LSMS 方法对 $FeNi_3$ 合金（具有 $L1_2$ 晶体结构）进行非共线磁性研究。MuST 的 CPU 版本、SCILIB-Accel 和 GEMMul8 均使用加载 nvhpc-hpcx-cuda13/26.1 模块的 nvhpc/26.1 构建。基准测试在 NVL4 节点上的 NVIDIA GB200 分区进行。

4 讨论

为建立基准，我们分别在 CPU 上运行 MuST LSMS，并使用 SCILIB-Accel 进行 CPU-GPU 卸载，均采用原生 FP64 模式。图 1a 展示了在不同模拟模式下，对所有高斯求积能量点计算得到的 $G(z)$ 相对于原生 FP64 模式的最大百分比误差。数据标签 i bits/j mods 表示不同的模拟模式：i 表示 cuda13 eager 策略中的尾数位数，j 表示 GEMMul8 中的模数数量。精度较低的 31bits/10mods 模式最大百分比误差高达 10^{-2}。随后，39bits/12mods 和 47bits/14mods 模式展现出指数级的改进。55bits/16mods 模式的精度达到 10^{-10}，其误差水平已足以涵盖不同编译器或高性能计算系统上 FP64 代码常见的差异。尽管 63bits/18mods 模式可能提供更高精度，但其影响受 MuST 软件包中其余 FP64 代码的限制。图 1b 列出了各模拟模式下，散射矩阵求逆在 2 次 SCF 迭代中的累计执行时间。结果展示了在 INT8 上加速 FP64 模拟求逆的效果。尤其是，GEMMul8 高精度模式平均实现 1.7 倍加速，凸显了数值精度与 INT8 张量核心高吞吐能力之间的有效权衡。



（a）仿真 $G(z)$ 的最大百分比误差。　　（b）格林函数计算的累计耗时。

图 1. 使用 CUDA13 和 GEMMul8 时，混合精度仿真对 LSMS 方法的影响。$G(z)$ 统计了所有能量点和原子位点，需要对一个 $33,750 \times 33,750$ 阶双精度复矩阵求逆，这对标准 FP64 求解器构成了重大的计算挑战。

表 1 总结了仿真控制对物理可观测量精度的影响。值得注意的是，使用相同的初始势进行 SCF 计算时，所有更高精度的仿真模式均在 10^{-6} 内实现了有效势和电子密度的自洽，其数值精度与 FP64 基准相当。稿件已提交至 ACM

而 31 位模式未能收敛。值得注意的是，31 位模式下的总能量也与 FP64 高度吻合，尽管该模式下的 $G(z)$ 百分比误差高达 10^{-2}。这种稳定性符合 LSMS 方法的特点。首先，LSMS 方法的数值稳定性得益于 $G(z)$ 的谱特征。

表 1. 仿真控制对物理可观测量精度的影响。每原子总能量 E_{tot} 的单位为里德伯，局域磁矩 μ 的单位为玻尔磁子。计算中采用 PBE 交换-相关泛函。

mode	E_{tot}	μ(Fe)	$\mu(\mathrm{Ni}_1)$	$\mu(\mathrm{Ni}_2)$	$\mu(\mathrm{Ni}_3)$	δQ(Fe)	$\delta Q(\mathrm{Ni}_1)$	$\delta Q(\mathrm{Ni}_2)$	$\delta Q(\mathrm{Ni}_3)$
GPU FP64*	-2894.19443	3.00418	0.58427	0.58427	0.58427	-0.29313	0.09771	0.09771	0.09771
cuda13 31 位	-2894.19443	3.00417	0.58426	0.58429	0.58425	-0.29313	0.09771	0.09769	0.09772
cuda13 55 位	-2894.19443	3.00418	0.58427	0.58427	0.58427	-0.29313	0.09771	0.09771	0.09771
GEMMul8 10 个模数	-2894.19443	3.00417	0.58427	0.58427	0.58427	-0.29313	0.09771	0.09771	0.09771
GEMMul8 16 个模数	-2894.19443	3.00418	0.58427	0.58427	0.58427	-0.29313	0.09771	0.09771	0.09771

* GPU FP64 模式是 CPU 二进制与 SCILIB-Accel 自动 BLAS GPU 卸载的组合。

如我们先前的先导研究所示， [10], 由模拟引起的显著百分比误差主要局限于费米能附近的狭窄区域，在此处格林函数呈现更尖锐的谱特征。然而，由于复平面轮廓 C 上的每个高斯求积点所对应的积分权重相对较小，这些局部模拟误差在轮廓积分过程中会被有效抑制并平均掉。其次，与基于密度泛函理论的第一性原理方法中的变分原理一致，总能量相对于电子密度处于驻定状态。因此，低精度模拟模式导致的密度一阶误差只会引起总能量的二阶误差。这种对模拟量化噪声表现出的稳健性表明，LSMS 框架适用于 GPU 加速，尤其是在利用基于模拟的张量核心运算时。

5 结论

本研究初步探讨了如何使用 INT8 数据类型模拟浮点矩阵乘法运算。我们将这一方法与此前开发的自动 BLAS GPU 卸载工具相结合，并应用于 MuST 套件中著名的、基于 FP64 的量子物理 CPU 代码 LSMS。借助 Ozaki 方案的实现，我们探索了多种模拟模式，以确定效率与科学模拟所需精度之间的最佳平衡。结果表明，这些模拟方案对于基于 FP64 的科学应用既有效又准确，能够在不牺牲精度的情况下加速模拟。

我们的模拟方法为科学模拟提供了更广泛、更通用的解决方案，有别于典型的混合精度研究；后者会修改特定的求解器算法，以便在中间计算中采用较低精度。与这类方法不同，我们的方法在优化硬件资源利用率的同时，保留原始算法和代码的完整性，不作任何修改。我们主张重新评估科学应用的精度要求，并确定真正必需的精度。我们鼓励采用低位宽类型来提升硬件利用率，尤其是在人工智能对新型硬件的影响日益增强的背景下。此外，我们呼吁硬件开发者与计算科学家开展合作，设计出能够更好满足未来科学计算与人工智能需求的最优数据类型。

致谢

本研究由美国国家科学基金会通过项目编号 OAC-2402542 和 OAC-2139536 资助，同时也得到领导级计算设施（LCCF）特色科学应用项目的支持。作者感谢提交给 ACM 的稿件

英伟达公司提供 NVL4 节点的访问权限；该节点在架构上与 LCCF Horizon 系统的节点相似。此外，作者感谢人工智能工具（Grok、ChatGPT、Claude 和 Gemini）在本稿件起草和语言润色过程中提供的技术协助。

参考文献

[1] Ahmad Abdelfattah、Hartwig Anzt、Erik G Boman、Erin Carson、Terry Cojean 等。2021。混合精度算术数值线性代数方法综述。《国际高性能计算应用期刊》35卷4期（2021年7月），344–369。doi:10.1177/10943420211003313 出版商：SAGE Publications Ltd STM。[2] M. Eisenbach、C.-G. Zhou、D. M. Nicholson、G. Brown、J. Larkin 等。2009。纳米尺度系统自由能从头计算的可扩展方法。载于《高性能计算、网络、存储与分析会议论文集》（俄勒冈州波特兰）（SC '09）。计算机协会，纽约，纽约州，美国，论文编号64，8页。doi:10.1145/1654059.1654125[3] 慧与公司。2024。HPE Cray 编程环境文档。HPE Cray。https://h41374.www4.hpe.com/docs/csml/cray_libsci acc.html_[4] Y. Hida、X.S. Li 和 D.H. Bailey。2001。四双精度浮点算术算法。载于第15届 IEEE 计算机算术研讨会论文集。ARITH-15 2001。IEEE，科罗拉多州韦尔，美国，155–162。doi:10.1109/ARITH.2001.930115 ISSN：1063-6889。[5] Hiroyuki Ootomo 和 Piotr Luszczek。2024。ozIMMU。检索于2025年3月16日，网址：https://github.com/enp1s0/ozIMMU[6] IBM。2018。IBM POWER Linux 工程与科学子程序库。IBM。https://www.ibm.com/docs/en/SSFHY8_6.1/reference/essl_reference_pdf.pdf[7] Junjie Li。2025。采用 OpenMP 首次触及式数据移动的统一内存架构高性能自动 BLAS 卸载。arXiv:2501.00279 [cs.DC] https://arxiv.org/abs/2501.00279[8] Junjie Li 和 Yinzhi Wang。2024。SCILIB-accel：自动 BLAS 卸载工具。GitHub。https://github.com/nicejunjie/scilib-accel[9] Junjie Li、Yinzhi Wang、Xiao Liang 和 Hang Liu。2024。统一内存架构上的自动 BLAS 卸载：NVIDIA Grace-Hopper 研究。载于《2024年高级研究计算实践与经验会议：人力驱动计算》（PEARC '24）。计算机协会，纽约，纽约州，美国，1–5。doi:10.1145/3626203.3670561[10] Hang Liu、Junjie Li 和 Yinzhi Wang。2025。通过自动 BLAS 卸载实现可调精度仿真的初步研究。arXiv:2503.22875 [cs.DC]https://arxiv.org/abs/2503.22875 [11] MuST 开发者。2026。用于第一性原理计算的多重散射理论代码。检索于2026年3月16日，网址：https://github.com/mstsuite/MuST[12] 英伟达。2024。NVBLAS 文档。英伟达。https://docs.nvidia.com/cuda/nvblas[13] 英伟达。2025。NVIDIA cuBLAS。https://docs.nvidia.com/cuda/cublas/#floating-point-emulation。访问日期：2026年3月27日。[14] 英伟达公司。2026。NVIDIA HGX 平台。https://www.nvidia.com/en-us/data-center/hgx/ 访问日期：2026年3月29日。[15] Hiroyuki Ootomo、Katsuhisa Ozaki 和 Rio Yokota。2024。整数矩阵乘法单元上的双精度通用矩阵乘法。《国际高性能计算应用期刊》38卷4期（2024年7月），297–313。doi:10.1177/10943420241239588 出版商：SAGE Publications Ltd STM。[16] Heidi Poxon。2013。Cray 加速科学库简介。（2013）。https://www.olcf.ornl.gov/wp-content/uploads/2013/01/ Scientific_Libs.pdf[17] Yuki Uchino、Qianxiang Ma、Toshiyuki Imamura、Katsuhisa Ozaki 和 Patrick Lars Gutsche。2025。基于中国剩余定理的复矩阵乘法模拟。arXiv:2512.08321 [cs.DC] https://arxiv.org/abs/2512.08321[18] Yuki Uchino、Katsuhisa Ozaki 和 Toshiyuki Imamura。2025。整数矩阵乘法单元上 Ozaki 方案的性能提升。《国际高性能计算应用期刊》39卷3期（2025年），462–476。doi:10.1177/10943420241313064[19] B. Ujfalussy、Xindong Wang、Xiaoguang Zhang、D. M. C. Nicholson、W. A. Shelton 等。1998。用于复杂磁性第一性原理方法的高性能实现。载于《1998年 ACM/IEEE 超级计算会议论文集》（加利福尼亚州圣何塞）（SC '98）。IEEE 计算机学会，美国，1–15。[20] Yinzhi Wang 和 Junjie Li。2023。PEAK：高性能计算系统轻量级性能分析器。载于《国际高性能计算、网络、存储与分析会议 SC '23 研讨会论文集》（SC-W '23）。计算机协会，纽约，纽约州，美国，677–680。doi:10.1145/3624062.3624143[21] Yang Wang、G. M. Stocks、W. A. Shelton、D. M. C. Nicholson、Z. Szotek 等。1995。电子结构计算的线性标度多重散射方法。《物理评论快报》75（1995年10月），2867–2870。第15期。doi:10.1103/PhysRevLett.75.2867[22] Yuki Uchino。2024。ozIMMU-H。检索于2025年3月16日，网址：https://github.com/RIKEN-RCCS/accelerator_for_ozIMMU[23] Yuki Uchino、Emil Briggs。2026。GEMMul8。检索于2026年3月16日，网址：https://github.com/RIKEN-RCCS/GEMMul8

收稿日期：2026年3月30日