准确的乘积，不准确的逆：分块三角求逆中的双精度浮点 BLAS 仿真

Erzheng Zhang 计算机科学系，多伦多大学 多伦多，加拿大 erwin@cs.toronto.edu

arXiv:2611.31030v1 [cs.MS] 23 Nov 2026

摘要—整数张量单元上的双精度浮点仿真如今已在 **cuBLAS** 和 **GEMMul8** 中提供，并且每个仿真矩阵乘积都可以根据精度等级进行检查。我们探究了由此类乘积构建的例程是否会继承它们的精度，并表明分块三角求逆（这是 **LAPACK** 对称正定矩阵求逆的核心）并不会。在定精度仿真下，每个乘积都通过 **Demmel** 等人的 **B** 级误差界，但逆矩阵相对于原生执行的 2.0×10^{-15}，误差达到 7.1×10^{-2}。这些矩阵在对角缩放后条件良好，但缩放跨度达到 **64** 个二进制阶区间。原因在于其中一个乘积 $T = X_{22}L_{21}$：其求和指标同时承载了对角缩放及其逆，而 **B** 级界无法吸收这一影响。原生计算这些乘积即可恢复原生精度；即使只保留其中一个乘积采用仿真，也无法恢复。该模式在四个 **GPU** 上均成立；对于实际协方差矩阵，定精度路径的精度损失最高约达四个数量级。均衡化、基于求解的结合方式以及 **cuBLAS** 的自动动态精度可以避免这种损失，而精确误差递推则可对已完成的运行进行审计。在 **RTX 4090** 上，速度最快且精确的仿真求逆耗时为 **cuSOLVER** 的 **0.40** 倍。每个乘积的缩放落在哪个范围，决定了逐乘积精度能否传递到复合例程中；这正是仿真库构建者需要检查的事项。

I. 引言

自 CUDA 13 起，cuBLAS 可通过仿真 FP64 算术，在整数张量单元上运行 DGEMM、TRMM 和 TRSM [1], [2]。自动动态精度（ADP）是一种保护机制，可估算每次调用中乘积所需的位数。GEMMul8 提供定精度仿真 [3], [4],，而仿真已被提议作为适用于广泛应用的默认算法 [5]。Demmel 等人根据每个仿真乘积达到的误差界对其进行评级 [6]。cuSOLVER、MAGMA 和 SLATE 等 LAPACK级库会调用这些乘积，因此它们的精度才是关键。

复合例程会组合许多乘积。Demmel 等人证明，B 级乘积可使分块 Cholesky 分解保持逐元素精确。对称正定矩阵的 LAPACK求逆还包括后半部分：三角矩阵求逆（TRTRI），然后将逆因子与其转置相乘（LAUUM）。其是否继承了乘积的精度，尚未有人探讨。

并非如此。在 $n = 4096$ 的 H100 上，分块求逆中的每个乘积都以充足余量通过行列缩放（B 型）误差测试；最大比值仅为容差的 0.12，且出现在原生乘积中。然而，经过定精度仿真后，逆矩阵从第二位有效数字起就不正确：相对误差为 7.1×10^{-2}，而同一良态输入在原生 FP64 下的相对误差为 2.0×10^{-15}。原因并非乘积有缺陷，而是对角缩放落在乘积的哪个索引上（图 1）。

变量具有不同单位时，用户就会遇到这类矩阵，协方差就是如此。当需要许多逆矩阵元素时，例如计算克里金方差 [7]，用户会使用显式逆矩阵。单位因子从 10^{-6} 到 10^{6} 的协方差会以同样的方式损失精度（14 个模数，RTX 4090，$n = 8192$：1.1×10^{-4} 对比 3.4×10^{-15}，附录 D）。

本文有四项贡献。

- 机制 （第三节）：一个三阶示例证明，B 级会导致这种损失；给出带有后验包围的精确误差递推关系；并建立一种可证明具有尺度等变性的对应关系。
- 诊断 （第五节 A、B 和 E 小节）：在四个 GPU 和两个库上，每个乘积都通过了测试，但逆矩阵却失败；因果测试需要考察每个内部索引乘积。
- 修复方案与成本（第 V-C–V-G 节）：三种修复方案都能让每个乘积保持仿真；一种可认证每次精确运行的包围区间；以及完整的设备驻留计时数据。
- 应用（第 V-H 节）：损失会影响人口普查数据上的协克里金预测值；在模型块金效应处，该损失处于认证界以内。

II. 背景与相关工作

A. FP64 仿真及其精度需求

Ozaki 方案 [8], [9] 在整数单元上计算双精度浮点乘积 $C = AB$。它用二的幂对 A 的每一行和 B 的每一列进行缩放，使每行或每列中的最大元素具有固定指数。随后，它将缩放后的元素拆分为 INT8 切片，并在张量核心上通过整数累加对切片对进行精确相乘。最后，以双精度浮点数对部分乘积求和。Ozaki-II 将切片替换为模 q 互素模数的剩余类，并通过中国剩余定理 [3]重构乘积。增加切片或模数可以提高精度，但代价也更高。

在这两种情况下，c_{ij} 可用的位数都相对于 $r_i s_j$设定，其中 r_i 和 s_j 分别是 $|A|$ 第 i 行和 $|B|$ [4]第j 列的最大项， [9]、 [10]。远小于 $r_i s_j$ 的项会损失约 $\log_2(r_i s_j/|c_{ij}|)$ 位，即其尺度超额，因此固定的位数

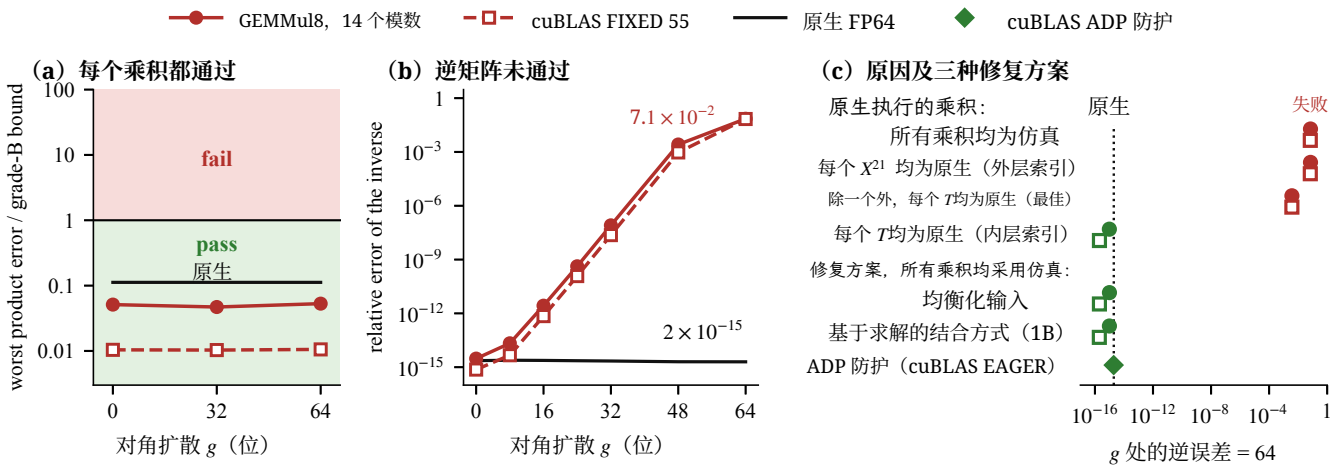


图 1。每个乘积都通过；逆矩阵 不满足 (H100, $n = 4096$)。（a）TRTRI 乘积误差超过其 B 级界的最大值；（b）逆矩阵误差；（c）一个 t $g = 64$，原生 T 可恢复这一精度，原生 X_{21} 不满足，而三种修复方案在所有乘积均采用仿真的情况下可恢复这一精度。

只有在每一项的超额误差都很小时才够用。Fasi、Mary 和 Mikaitis 给出了达到目标精度所需的定点宽度上界，并指出 ADP 的位估算值可能过于悲观 [11]。cuBLAS 运行 ADP 或 FIXED，后者采用固定的位数 [2]。ADP 估算一次调用所需的位数，然后用相应位数进行仿真，或分派至原生 DGEMM；其 DEFAULT 或 EAGER 策略决定仿真选择的宽松程度。GEMMul8 使用模数的数量 q。以上讨论均针对单个乘积；本文探讨的是当一个例程组合多个乘积时会发生什么。

B. 精度等级

Demmel 等人 [6]根据 GEMM 达到的上界对其进行分级：

- A 级，$|C - \widehat{C}| \le f(k)\, u\, |A||B|$ 按分量达到经典乘法的上界；
- B 级，$|C - \widehat{C}| \le f(k)\, u\, \mathrm{rowmax}(|A|)\, \mathrm{colmax}(|B|)$，这是行缩放和列缩放仿真的自然模型；
- C 级，即 LU 和 QR [12], [13]所适用的按范数上界。

对 A进行行缩放、对 B进行列缩放时，B 级是不变量；但对求和指标进行缩放时则不是。本文采用一种归一化方式，$f(k) = k$：其中 r_X 表示乘积误差与 u 乘以 B 级X 右端项之比的最大值；当 $r_X/k \le 1$时，该乘积通过A 型或B 型测试。由于 $(|A||B|)_{ij} \le k\, r_i s_j$，这两种测试并非彼此包含。测试 $r_B/k \le 1$ 比 Demmel 等人自己的基准阈值 $f_B(k) < k^2$更严格。当一个上界将所有舍入误差都纳入其中并据此证明某个量时，该量便是经认证的。

C. 分块三角求逆与 *LAUUM*

LAPACK 通过 Cholesky（potrf，$A = LL^T$）、三角求逆（trtri，$X = L^{-1}$）和 LAUUM（$A^{-1} = X^T X$）对称正定 A 求逆，合起来构成 potri。递归分块 TRTRI 将 $L = \left(\begin{smallmatrix} L_{11} & \\ L_{21} & L_{22} \end{smallmatrix}\right)$分割，求对角块的逆，$X_{ii} = L_{ii}^{-1}$，并形成

$$T = X_{22} L_{21}, \qquad X_{21} = -T X_{11}. \tag{1}$$

我们将 T 称为内部索引乘积，将 X_{21} 称为外部索引乘积。按照 Du Croz 和 Higham 的术语 [14], ，(1) 是方法 2B。方法 2C 形成 T，然后使用 L_{11}求解（参考值 dtrtri）；方法 1B 形成 $W = L_{21} X_{11}$ 并使用 L_{22} 求解，$X_{21} = -L_{22}^{-1} W$。在普通算术运算中，1B 和 2C 保持逐分量残差界，而 2B 则不保持。

D. 相关工作

混合精度求解器通常依据求解器层面的结果进行评判，而迭代细化往往是补救方法 [15]–[17]；采用 INT8 模拟的 LU 和 HPL 在分块数过少时无法通过残差检验 [18]。基于快速三级 BLAS 的分块算法在按范数意义下保持后向稳定，但可能损失逐分量精度，并且不具备基数幂缩放下的不变性 [12]。Carson 和 Ma 分析了采用 Ozaki 尾部更新的分块 LU，并给出了行列范数界 [19]；他们没有研究求逆或对角缩放。针对快速乘法，已有研究考察了对单个乘积操作数进行对角缩放 [20]，而均衡化则是经典方法 [21], [22]。因此，已有研究表明，操作数缩放会消耗固定精度乘积的位数 [9], [11], ，均衡化有所帮助，而且在常规算术运算中，分块求逆的结合方式很重要 [14]。本文针对 Demmel 等人的工作（其定理涉及 Cholesky 分解），补充了一个复合过程，其中 B 级乘积并不足够。针对 Du Croz 和 Higham 的工作，本文揭示了对角缩放会带来 $X_{22} L_{21}$，而原生 FP64 无法察觉这一点。针对 ADP 的单乘积评估 [1] 和 Fasi 等人的位宽界 [11]，本文进一步考察了完整已发布例程内部的行为，以及每项修复措施的代价。

III. 为什么逐乘积检查无法组合

a) 缩放发生的位置：令 $A = DMD$ with$D = \mathrm{diag}(2^{k_i})$, $k_i \in \{0, \dots, g\}$, and $M = L_M L_M^T$。则 $L = DL_M$and $L^{-1} = L_M^{-1} D^{-1}$。图 2 展示了每个乘积的行索引、求和指标和列索引分别携带哪种缩放。在$T = X_{22} L_{21}$ 中，求和指标 k 在列上携带 d_k^{-1}

$A = DMD$，因此 $L = DL_M$（行 $\times D$），且 $X = L^{-1} = L^{-1}{}_M D^{-1}$（列 $\times D^{-1}$）

乘积	行 i	求和 k	列。j	B 级界
$T = X^{22}L^{21}$	1	$D^{-1}\cdot D$	1	误差随 D的扩散
$X^{21} = -TX^{11}$	1	1	D^{-1}	被吸收
LAUUM X^TX	D^{-1}	1	D^{-1}	已吸收
1B：$W = L^{21}X^{11}$	D	1	D^{-1}	已吸收

图 2。只有 T 对缩放后的索引求和（红色）；B 级界吸收了行 i 和列 j 上的缩放（绿色）。

X_{22} 的，以及 L_{21}各行上的 d_k。$T = L_M^{-1}{}_{,\,22}L_{M,\,21}$ 本身不依赖于 D，但 X_{22} 的一行或L_{21} 的一列所跨越的二进制阶区间数，与收缩块中的指数数相同。只有当两个极值都落在同一个分块内时，跨度才是 g，因此它取决于 k_i的顺序。 X_{21} 和 LAUUM 的$X^TX = D^{-1}L_M^{-T}L_M^{-1}D^{-1}$ 仅在外部索引上携带 D；方法 1B 不会形成对缩放索引求和的乘积。原生双精度浮点不受影响：在此处的每项测试中，按 (1) 进行的原生求逆都具有逐位尺度等变性，也就是说，对 DMD 计算得到的逆，恰好等于 D^{-1} 乘以对M 计算得到的逆，再乘以 D^{-1}。要分析这种损失，需要一个能考虑内部缩放的乘积界。

b) B 级界无法保证逆矩阵：一次可接受的递归求值会针对每个乘积返回任意一个误差满足所述上界的值，并对其他每个运算进行精确求值。

命题 *1*：令 L_M为元素全为 1 的 3×3下三角矩阵，$D = \mathrm{diag}(1,\ 1,\ 2^g)$ 和 $L = DL_M$，在第一行和第一列之后进行分块。对于每个满足 $|\delta| \le f_B(2)\ u\,2^g$ 的 δ，都存在一种符合 B 级精度要求的分块逆矩阵计算方式，使得 $\|\widehat{X} - L^{-1}\|_F/\|L^{-1}\|_F = |\delta|/(4 + 4^{-g})^{1/2}$，而对于每个 g，$\kappa_2(M) \approx 16.39$。

证明：$X_{11} = 1$、$X_{22} = \left(\begin{smallmatrix} 1 & 0 \\ -1 & 2^{-g} \end{smallmatrix}\right)$ 和 $L_{21} = (1,\ 2^g)^T$，因此 $T = (1,\ 0)^T$，其第二个元素是消去项 $-1 + 2^{-g}2^g$。对于它，$|X_{22}|$ 的行最大值为 1，列 $|L_{21}|$ 的最大值为 2^g 和 $k = 2$，因此 $T_2 = \delta$ 满足 B 级精度要求。在其他所有运算均精确的情况下，$\widehat{X}$ 与 L^{-1} 的差异仅在于 $(3,\ 1)$ 位置上的 $-\delta$，且 $\|L^{-1}\|_F^2 = 4 + 4^{-g}$。 ■

采用 $f(k) = k$ 时，上界允许的相对误差约为 2^{g-53}，在 $g = 64$ 处为 2,048；而在同一元素上，A 级仅允许 $O(u)$（附录B）。这一示例同样适用于方法 2C，因为使用 $L_{11} = 1$ 求解不会改变任何结果。分块 Cholesky分解能够避免这一问题，因为其 B级三角求解和 SYRK 的右端项由因子的行范数约束，最多为 $(a_{ii}a_{jj})^{1/2}$，这些行范数包含外部索引上的缩放 [6]。求逆乘积 T 则没有这种保护：其 B 级容差会随着收缩分块的扩散而增大。

c) 乘积误差如何传递到逆矩阵：L 是存储因子；$X = L^{-1}$ 和 $Y = X^TX = (LL^T)^{-1}$ 是精确的，因式分解的误差不在此论述范围内。在列块 ρ_1 和行块 ρ_2处进行拆分时，将计算得到的 $\widehat{\ }$ $\widehat{\ }$ $\widehat{\ }$ $\widehat{\ }\widehat{\ }$个乘积记为 $T = X_{22}L_{21} + E_s$ 和 $X_{21} = -(TX_{11} + F_s)$，$\widehat{\ }$ $\widehat{\ }$，并令 $\Phi_s = -(E_sX_{11} + F_s)$，$\Delta = X - X$ 和 Δ_ℓ 表示叶节点 ℓ 的逆矩阵误差。LAUUM 返回 $\widehat{Y} = \mathrm{fl}(\mathrm{sym}(\widehat{X}^T\widehat{X} + G))$，其输出误差为 G，且 $\mathrm{sym}(C) = (C + C^T)/2$，每个元素进行一次舍入加法。

定理 *1*：设取负运算和块复制均为精确运算，并且对称化中的每次加法和二等分均为有限值（就近舍入；允许下溢）。

（a）在每次分割处

$$\Delta_{21} = \Phi_s - \Delta_{22}L_{21}\widehat{X}_{11} - (\widehat{T} - E_s)\Delta_{11} + \Delta_{22}L_{21}\Delta_{11}, \quad (2)$$

并且在分割和叶节点 ν 上对 $\Delta = \sum_\nu \Lambda_\nu\Psi_\nu\Gamma_\nu$，其中 $\Lambda_\nu = X_{:,\ \rho_2}L_{22}$, $\Gamma_\nu = L_{11}X_{\rho_1,\ :}$，$\Psi_s = \Phi_s - \Delta_{22}L_{21}\Delta_{11}$ 以及 $\Psi_\ell = \Delta_\ell$（叶节点的分块同时为 ρ_1 和 ρ_2）。

（b）如果 $|E_s| \le \mathcal{E}_s$、$|F_s| \le \mathcal{F}_s$、$|\Delta_\ell| \le B_\ell$ 和 $|G| \le \mathcal{G}$，那么对于由叶节点自底向上通过$|\Delta| \le B$ 组装的 B

$$B_{21} = \mathcal{E}_s|\widehat{X}_{11}| + \mathcal{F}_s + B_{22}|L_{21}\widehat{X}_{11}| + (|\widehat{T}| + \mathcal{E}_s)B_{11} + B_{22}|L_{21}|B_{11}, \quad (3)$$

并且 $|\widehat{Y} - Y| \le |\widehat{X}|^TB + B^T|\widehat{X}| + B^TB + \mathrm{sym}(\mathcal{G}) + 2u|\widehat{Y}| + 2^{-1074}\mathbf{1}\mathbf{1}^T$。

(c) 设 $A = DMD$、$L = DL_M$，并写作 $B' = BD$、$\widehat{\ }$ $\widehat{\ }$ $X' = XD$（M T $\mathcal{E}_s =$ 的框架）。如果每个乘积都满足 A 级要求且具有 $c\,u|X_{22}||L_{21}|$，其余乘积满足 B 级要求，并且 c 和$B_\ell D_\ell$ 不依赖于 D，那么对于 B'，式 (3) 及其对 $D|Y - Y|D$ 所得的 $\widehat{\ }$ $\widehat{\ }$上界，对运行的 X'、T、$\widehat{\ }$ DYD、L_M 和 c 而言，都是同一个函数，且对每个二的幂 D均成立；但绝对值项会变为 $2^{-1074}dd^T$（d D的对角线）。如果只有$\sim T$ 满足 B 级要求，则 $\mathcal{E}_s|X_{11}|$ 携带的扩散至多为 D 在收缩块上的最大值$_{k\in\rho_2}\,d_k$/与最小值$_{k\in\rho_2}\,d_k$ 之差$\rho_s =$。

不假设任何误差较小，也不引入精确逆矩阵或高精度逆：包围区间根据 L、com-$\widehat{\ }$ $\widehat{\ }$puted X 和 T以及逐乘积误差预算计算得到。第 (c) 部分用一句话概括了该机制：T 上的 A 级条件使上界不含 D，B 级条件则允许其随 ρ_s增长，而命题 1 达到了这一增长。证明见附录B。

d) 将缩放保持在外部的关联：方法 1B 构造 $W = L_{21}X_{11} = D_2L_{M,\ 21}L_M^{-1}{}_{,\ 11}D_1^{-1}$，并递归向下求解 $X_{21} = -L_{22}^{-1}W$，直至原生 TRSM 叶节点，因此没有任何乘积会收缩缩放；每次规模为 $2h$的划分需耗费 $3h^3$ 次浮点运算，而乘积形式的开销为 $4h^3$。如果一个 GEMM 对于二的幂对角矩阵 D_1、D_2满足 $G(D_1A,\ BD_2) = D_1G(A,\ B)D_2$，则称其为 外等变。

命题 *2*：在附录B的假设 (i) 至 (v) 下，DMD 基于求解的逆等于 D^{-1} 的逆乘以M 的逆乘以 D^{-1}，逐位完全一致。

等变性会随着结果传递误差；但这并不意味着准确性，准确性需要进行测量。乘积形式没有这一性质，LAPACK 的方法 2C 在

下三角因子上也同样没有这一性质，而该因子构成相同的 $X_{22}L_{21}$：在透明 CPU 仿真器上（8 个切片， $n = 2048$），仅对该乘积进行仿真时，其误差会从 $g = 0$ 处的 6.0×10^{-14} 增长到 $g = 64$ 处的 8.9×10^{-2}，正如乘积形式的误差一样（附录 D）。

IV. 实验设置

表 I 列出了设计方案。两个实现采用相同的分块规则和调用结构，使用 $\beta = 0$ 计算每个乘积，并将每次更新作为独立操作执行。数值测试框架获取 LAPACK 的主机端分解因子，在主机上对叶节点求逆，并将每个 GEMM 发送到设备；所有精度结果和等级均采用该实现。等级涵盖每个乘积的每个输出元素，并与长双精度参考乘积进行比较；该参考乘积自身的误差按元素设有上界（附录 C）。设备驻留实现使用 cuSOLVER 进行分解，并通过 cublasDtrsm 求解叶节点；所有时间测量均采用该实现。每条计时路径均从上传矩阵开始，到得到逆矩阵结束，包括 potrf 和任何均衡化操作；每条路径运行 10 次，与其他路径交错执行，并以轮换顺序进行，在每个 GPU 上开展多次会话。其乘积不参与等级评定；审计会在 RTX 4090 上重新评定计时路径的乘积，从而将两个实现联系起来（附录 A）。CUBLAS_EMULATION_STRATEGY 未设置，因此 DEFAULT 表示该库自身的默认设置。

DMD的参考值是 $D^{-1}X_M D^{-1}$，即 X_MLA- PACK 对 M 求得的逆矩阵，并与高精度逆矩阵 $\widehat{\ }$（附录 A ）进行核对。逆矩阵的误差为 $\epsilon_X = \|X - X_{\mathrm{ref}}\|_F / \|X_{\mathrm{ref}}\|_F$；对于实矩阵，误差是在平衡框架中测量的（在两侧均乘以 diag(A) $^{1/2}$进行缩放），在此框架下，按范数计算的误差不会掩盖对角线较小的分块。斜率 表示每单位 g损失的 ϵ_X 位。路径按其 GEMM 后端命名（表 I）；“等式”表示二的幂均衡化输入，除非某行另有说明，否则分块求逆中的每次 GEMM 都在该后端上运行。对于一组矩阵，表中给出该组中最差的 ϵ_X 。

表 I 实验设计。

因子	取值
合成输入	DMD、 $M = W^T W/n + I$、 W标准正态分布（$\kappa(M) \approx 6$）； k_i在 $\{0, \ldots, g\}$上均匀分布， $g \in \{0, 8, 16, 24, 32, 48, 64\}$； $n \in \{2048, 4096, 8192\}$，宽度为 16384；随机种子 0–4
控制变量	指数已排序或按连续分块排列； $\kappa = 10^8$， 10^{14}
实矩阵	具有统一对角线；高斯过程核；单位因子 10^{-6} to 10^6 (对角线跨度 2^{80})
应用	16 个 SuiteSparse 对称正定矩阵和 8 个原始单位协方差（人口普查、空气质量）， n 1074 至 8190，对角线跨度 2^{20} 到 2^{44}
图形处理器和	协克里金、加利福尼亚住房、 $n \in \{4608, 9216\}$、块金效应 $\tau \in \{10^{-1}, 10^{-3}, 10^{-6}\}$
库	FP64 稀缺：RTX 4090、L40S；FP64 丰富：H100、A100。cuBLAS 13.8.1、cuSOLVER 12.3.5、GEMMul8 v3.5.4；十个版本，cuBLAS 13.0 到 cuBLAS 13.10 ，在 H100 上，L40S 以及 A100 上
GEMM 返回 ends	原生 FP64 双精度矩阵乘法；ADP 搭配 DEFAULT 或 EAGER；F55、F79、F111：FIXED，分别使用相应的位数； q = 14、16、20：GEMMul8 模数
路径	cuSOLVER potri；分块乘积形式；混合方案，搭配每个 T 或每个 X_{21} 原生；基于求解的（方法 1B）；也都经过均衡化；LAUUM 作为一次 GEMM 或分块；叶节点宽度为 256

表 IIϵ_X 在 $g = 64$， $n = 8192$（REAL：L40S，按组最差情况）。固定路径均失败；均衡化修复了除 BCSSTK23 之外的所有情况。

				均衡化的	
输入	原生	q=14	F55	q=14	F55
H100	2.9×10^{-15}	7.1×10^{-2}	6.8×10^{-2}	2.0×10^{-15}	2.9×10^{-16}
A100	2.9×10^{-15}	7.1×10^{-2}	6.8×10^{-2}	2.0×10^{-15}	2.7×10^{-16}
RTX 4090	2.8×10^{-15}	7.1×10^{-2}	6.8×10^{-2}	2.0×10^{-15}	2.7×10^{-16}
L40S	2.9×10^{-15}	7.1×10^{-2}	6.8×10^{-2}	2.0×10^{-15}	2.9×10^{-16}
人口普查	2.9×10^{-15}	1.5×10^{-11}	3.7×10^{-12}	8.4×10^{-15}	2.3×10^{-15}
SuiteSparse	3.6×10^{-14}	1.7×10^{-12}	3.2×10^{-13}	2.7×10^{-13}	3.4×10^{-14}

V. 结果

A. 每个乘积都通过了，逆矩阵却仍会失败吗？

在三种扩散下，对十条路径的每个 TRTRI 乘积中的每个输出元素进行分级，可以回答这一问题。每条路径的每个乘积都通过了 B 型测试（图 1(a)），但定精度路径的逆矩阵却失败了（图 1(b)）。因此，乘积级检查无法区分这些路径与准确路径。按范数审计和无参考残差的结果一致，因此失败并非分级或参考值造成的制品（附录 D）。

B. 内层索引是原因吗？

如果该机制是正确的，那么内层索引乘积会承受损失，而外层索引乘积不会。重新运行 14 个模数和 FIXED 55，并在 $g = 64$ 中使用由原生 DGEMM 计算的所选 TRTRI 乘积，可以验证这一点（图 1(c)）。将每个 T 都原生执行可恢复原生精度，但会牺牲速度（附录 F）；将每个 X_{21} 都原生执行不会带来任何变化，而仅保留一个 T 仿真仍至少保留 3.7×10^{-3}。

在测试过的所有替代方案（附录 D）中，让七个或更多 T保持仿真的 64 个方案，其误差均达到 1.6×10^{-2} 或更高；而将每个 T都原生执行的 10 个方案，其误差均为 2.0×10^{-15} 或更低。每个 T都通过其自身的收缩块产生影响，因此修复需要涵盖其中的每一个。

C. 如何修复？

任何将缩放移出求和指标的做法都能修复逆矩阵。二的幂均衡可以做到这一点：对 SAS 求逆，其中 S 是将每个对角元素带入 $[1, 4$ 的二的幂对角矩阵；再将结果按 S 进行缩放，即可再次消除 D ，且每个元素的误差精确地不超过二倍。每条经过均衡化的 FIXED 路径在每个 g 上都保持其 $g = 0$ 精度（表 II）。它只需要对角线，而每种对称正定矩阵求逆都有对角线，并且无需改动库。

基于求解的结合方式无需接触输入即可修复逆矩阵。在所有测试案例中，它都具有逐位尺度等变性，正如命题 2 所预测的那样；而且对于原始输入，其误差从未超过 5.5×10^{-15} （附录 D）。

ADP 无需更改输入或算法即可避免损失：每条保护路径与

表 III 只有固定路径未通过 A 型测试，在 T 上相差一个随 g (H100, $n = 4096$)增长的因子。

	r_A/k 在 T 上		
路径	$g = 0$	$g = 32$	$g = 64$
原生	1.5×10^{-2}	1.5×10^{-2}	1.5×10^{-2}
EAGER	7.2×10^{-3}	9.1×10^{-3}	1.3×10^{-2}
q=14	2.8×10^{1}	1.1×10^{9}	3.5×10^{13}
F55	4.3×10^{0}	5.6×10^{8}	3.5×10^{13}
q=14，等式	2.8×10^{1}	2.8×10^{1}	2.8×10^{1}
F55，等式	4.3×10^{0}	4.3×10^{0}	4.3×10^{0}

每个 GPU 的合成网格上的原生结果之比均保持在 1.05 以内。在 DEFAULT 下，选择取决于 GPU：在 cuSOLVER 的 potri 内部，FP64 丰富型 GPU 不模拟任何调用，而 FP64 稀缺型 GPU 则模拟部分调用（附录 E）。

D. 为什么这些修复方案有效？

定理 1(c)指出了其中的关联：T未达到 A 级的程度、其后续扩散的大小 ρ_s，以及给定的位数 T。四项测量对这些关联进行了检验。

a) 对 T 的未达标程度决定了误差：在 $g = 64$，每个 T 的第 99 百分位尺度超额 e_{99} 为 69.5 至 71.5 位，而 X_{21}至多为 8.7，因此固定预算会使 T 不足，除此之外没有其他影响。A 型测试证实了这一点（表 III）：只有固定路径在 T上未达标，且差距随 g增大；当差距较大时，逆矩阵误差约在一个数量级的范围内跟随 u 变化（附录 D）。关键在于未达标的程度，而非是否未达标：在 $g = 0$，固定路径未达到 A 级，但仍能准确求逆。

b) 关键的扩散发生在每个分块内部：重新排列相同的指数只会改变 ρ_s，不会改变其他任何内容。对指数进行排序后，使每个收缩块的跨度仅覆盖少数几个二进制阶区间，16 个模数在 $n = 8192$、$g = 64$ 处给出 5.5×10^{-11}，而随机排列时为 6.7×10^{-2}（连续分块：附录 D）。

c) 均衡化消除了扩散：均衡化后，对角线跨度在每个 g处都小于四倍，因此没有任何分块会承载随 g 增长的扩散。均衡化路径在 T、$g = 64$ 处的偏差仍保持其 $g = 0$ 规模（表 III），而 ADP 对均衡化输入的估算值最多为 75 位。

d) ADP 给出 T 它所消耗的位数：ADP 的估计值与 $53 + e_{99}$ 相吻合（最小二乘斜率为 1.00，见图 3）。通过调节内部索引跨度得到的单个乘积展示了后续情况（见图 4）：每种输出规模都会在大致相同的估计值处切换到双精度浮点路径，4096×4096 输出介于 103 和 107 位之间。该路径并非没有代价：它仍需计算估计值，而且在 H100 上，它使用仿真库执行双精度浮点 GEMM，因此耗时比原生 DGEMM 更长，在小规模输出上尤其如此（附录 E）。因此，每个 T 要么获得其所需的位数，要么以双精度浮点运行，而外层索引乘积则继续进行仿真。

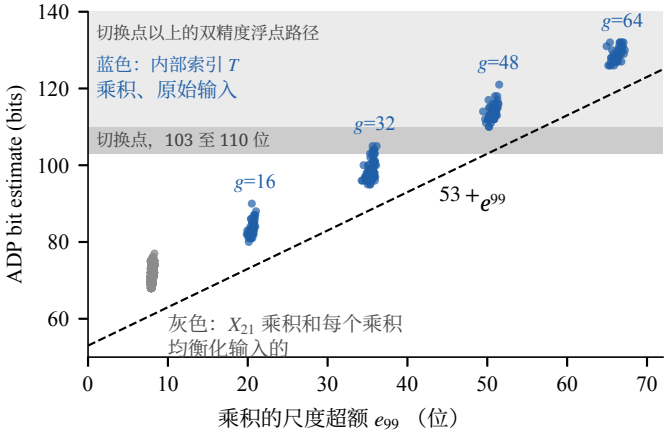


图 3。ADP 的估算值跟踪 $53 + e_{99}$，因此它要求的位数正好对应内部索引缩放所消耗的位数；超过切换点后，该调用以双精度浮点运行。

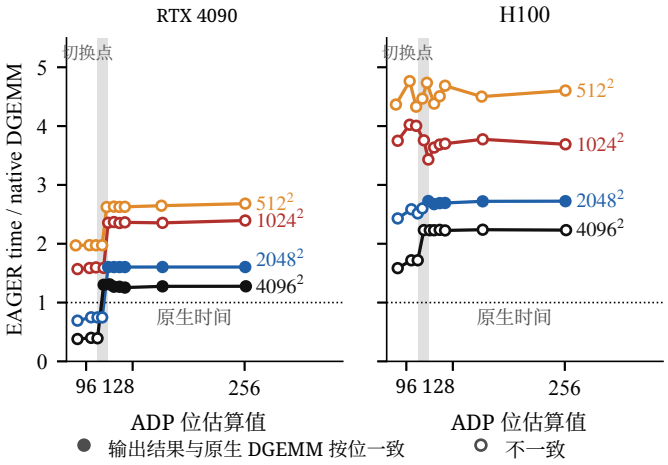


图 4。单个 EAGER 乘积的每种输出规模都在相同的报告估算值处切换路径（阴影部分）。

E. 它在其他图形处理器和实矩阵上也成立吗？

相同的网格在 $n = 8192$上运行于全部四个 GPU，并在 24 个实对称正定矩阵上测试了每条路径（表 II 展示了其中三组中的两组）。在每个 GPU 上，每条路径都在 $g = 64$达到相同的误差，精确到两位小数，并且在 g中以相同速率损失位数：这种损失是算术运算的特性，而非某一芯片的特性。它在 $n = 16384$ 以及不同随机种子下都持续存在。在 $g = 64$，14 和 16 个模数以及 FIXED 55 都达到约 6.7×10^{-2}，因此它们在该处的误差已经饱和；20 个模数和 FIXED 79 将误差降低了三到四个数量级，而 FIXED 111 的结果已接近原生实现，仅相差一个较小的倍数（附录 D）。

对于实矩阵，原始固定路径在 人口普查协方差矩阵 上的损失最大（表 II）。ADP 使每个矩阵与原生实现的差距都控制在 1.8 倍以内；对于均衡化的 14 个模数，除 bcsstk23 外的每个矩阵都控制在 10 倍以内。在 bcsstk23 上，行内的动态范围在对角缩放后仍然保留（均衡化后为 $\kappa = 3.9 \times 10^{7}$）；均衡化的 14 个模数仍是原生实现的 108 倍，因此应采用保护路径和基于求解的路径进行修复（附录 D）。

表 IV 包围区间认证了每一组准确结果（已认证的路由–扩散对，占 30 组；–：未运行）。

GPU	$n = 2048$	$n = 4096$	$n = 8192$
H100	27	26	–
A100	27	26	26
RTX 4090	27	26	26
L40S	27	26	26

在测试的 9 个支持双精度浮点仿真的 cuBLAS 版本中（cuBLAS 13.0 不支持该功能），保护路径均保持在原生水平，而固定路径均失败。cuSOLVER 自带的 potri 例程通过其模拟控制设置为 FIXED 55 后也会以相同方式失败，1.2×10^{-2} 在 $g = 64$；而在固定版本中，均衡化可以修复该问题，因此已发布例程在相同设置下呈现出相同的损失。

F. 包围区间认证了什么？

不使用参考逆矩阵，能否审计一次运行？定理 1 针对每次运行自身的操作数进行评估，并为每个乘积的误差给出分级比率界，同时将每次舍入纳入包围区间。它给出了相对于 $(LL^T)^{-1}$ 的已存储 L 的误差认证界；当该界小于 1 时，该次运行即通过认证。在每种 GPU 和规模的运行中，它都认证了至少 30 组路由–扩散对中的 26 组（表 IV）；在 H100 上的 $n = 4096$，这些包括全部 21 组准确结果和 5 组中度失败结果，只有误差最大的运行未获得误差界。该上界比实测误差高出 1 到 5 个数量级，且随着 n 增大，差距也更大，这正是最坏情况界的特点。因此，包围区间可用于审计已完成的运行，无需参考逆矩阵，也无需比该次运行更高的精度（附录 C）。

求值器本身通过精确算术、与独立的第二个评估器对照，以及在 CPU 上重建已存储的运行结果进行了验证（附录 C）。

G. 精度的代价是什么？

准确的仿真求逆比原生求逆更快吗？比较仅有一个因素不同的完整设备驻留路径即可回答这个问题（表 V，全文均采用分块 LAUUM；第 2 行和第 3 行相乘得到第 4 行）。均衡化几乎没有成本，因此是代价最低的修复方案。在双精度浮点性能稀缺的设备上，仿真很有优势：在 RTX 4090 上，16 个模数耗时仅为原生方案的 0.30，超过了弥补分块算法相较于 cuSOLVER 的性能差距所需的幅度，因此仿真路径的耗时仅为 cuSOLVER 的 0.40。在 H100 上，仿真路径相较于 cuSOLVER 的性能提升（耗时降至 0.64）来自分块算法，而非仿真：仅采用原生分块求逆，耗时就只有 cuSOLVER 的 0.44，且原生 FP64 速度最快。绝对耗时、传输、工作空间以及其他修复方案的成本见附录 F。

H. 损失是否会影响应用？

逆矩阵中的损失是否会影响用户实际读取的量？协克里金根据多个相关空间变量预测一个空间变量，并在每次预测中使用逆协方差

表 V 时间比（$n = 8192$，$g = 64$）：在 FP64 稀缺时仿真胜出，在 FP64 充足时分块算法胜出。

	FP64 丰富		FP64 稀缺	
时间比	H100	A100	RTX 4090	L40S
原生：平衡的与原始的	1.06	1.05	1.01	1.01
原生、等式与 cuSOLVER 对比	0.44	0.69	1.31	1.29
q=16 与原生对比，二者均已平衡	1.46	1.26	0.30	0.30
q=16，等式与 cuSOLVER 对比	0.64	0.88	0.40	0.39

表 VI 原始固定路线会移动协克里金预测值；均衡化和 ADP 不会（H100，$n = 4608$）。

路径	$\tau = 10^{-1}$	$\tau = 10^{-3}$	$\tau = 10^{-6}$
原生，cuSOLVER	1.0×10^{-13}	2.4×10^{-11}	6.5×10^{-9}
EAGER，cuSOLVER	1.7×10^{-13}	2.6×10^{-11}	6.4×10^{-9}
q=14	1.2×10^{-9}	4.2×10^{-8}	1.9×10^{-5}
F55	2.7×10^{-10}	3.3×10^{-8}	1.0×10^{-5}
q=14，等式	4.7×10^{-13}	2.2×10^{-11}	3.5×10^{-8}
F55，等式	1.3×10^{-13}	2.3×10^{-11}	1.9×10^{-8}
认证界，最差固定	7.9×10^{-7}	7.4×10^{-3}	—

及其方差 [7]。加利福尼亚住房数据 [23]，中的五个变量（单位与文件相同）给出了一个协方差$B_{\text{cov}} \otimes (C + \tau I)$，其对角线跨度覆盖多个二进制阶区间，并使用 C基于位置的平方指数核。块金效应 $\tau = 10^{-1}$分别构成模型、 10^{-3} 敏感性情形和 10^{-6} 压力测试情形。衡量指标是留出预测与经过认证的长双精度参考值之间的均方根距离，占预测器自身留出均方根误差的比例（附录 G）。

在两种尺寸和全部三种块金效应下，表现最差的固定路径使预测值偏离参考值的幅度达到 cuSOLVER 原生路径的 120 至 33,000 倍（表 VI 显示了 $n = 4608$）。ADP 的表现与原生路径持平，均衡化路径的误差也仅比它高出一个较小倍数。损失有所缓和，因为每个预测值只会通过投影到其协方差向量上而受到逆矩阵误差的影响；随着逆矩阵误差增大，实际误差占最坏情况的比例会下降（附录 G）。在该模型的块金效应下，认证界将损失限制在预测误差的 7.9×10^{-7} 以下。块金效应更小的应用，或直接读取逆矩阵元素的应用（例如计算克里金方差），受到的影响会更大。

VI. 经验与局限

a) 面向库构建者：以下三条规则值得遵循。对于复合例程中的每个乘积，不仅要检查误差有多大，还要检查操作数缩放落在哪个索引上。优先选择能将缩放保留在外部索引上的结合方式，例如方法 1B，或者在仿真乘积运算前进行均衡。LA- PACK 的参考 dtrtri 已针对上三角因子采用了这种做法：其上分支 $R = R_M D$ 是方法 1B 在 L上的转置；在第 III 节的仿真器模型中，它在每个 g 上都逐位尺度等变（5 项中 5 项通过），误差至多为 6.0×10^{-14}，而其下分支则是方法 2C。只有在逐调用保护机制（例如 ADP）的估算能够检测到

尺度超额时才依赖它，本例正是如此。如果保护机制只检查每个乘积的外部缩放，或者采用针对良好缩放输入调校的固定比特预算，那么它虽然能通过每项乘积测试，却仍会在此处失败。

b) 局限性：四个 GPU 均为 NVIDIA 的产品；我们无法获得 Blackwell GPU，而 cuBLAS 的 ADP 是我们所知唯一已发布的仿真防护机制。均衡化仅从 cuBLAS 13.7（H100 上为 13.8）起，才能将 FIXED 55 修复到原生水平。闭源库没有记录命题 2 的假设；它们只对输出进行检查。基于包围区间构建精度选择器属于未来工作。该应用使用一个数据集和一个协方差族。

VII. 结论

逐乘积精度等级无法组合用于分块三角求逆，因为其乘积形式会沿一个乘积的内索引压缩对角缩放。将缩放移出该索引，或为乘积提供其所消耗的位数，即可恢复精度；而在双精度浮点资源稀缺的情况下，精确的仿真求逆仍优于 cuSOLVER。

数据可用性

一个匿名化制品包含测试框架、保留的运行记录，以及支撑每个数值、表格和图形的脚本；补充材料_tables.pdf 包含所有完整表格和验证记录：https://figshare.com/s/bb220965206030883d9e（sha256 c68e11946d3fe33f51c8006806922fb0 5c084ea3b758ac3517e47bb4f53fc76f；单独的见证档案列于其 README 中）。

生成式人工智能的使用

Claude Code（Anthropic；Claude Opus 5.5）协助编写实验代码以及起草和修订文本；作者审阅并修订了全部内容，论文中的每个数字均由脚本根据结果文件生成。ChatGPT（OpenAI；GPT-5.6 Sol）用于批评草稿。

参考文献

[1] A. Schwarz、A. Anders、C. Brower、H. Bayraktar、J. Gunnels、K. Clark、R. G. Xu、S. Rodriguez、S. Cayrols、P. Tabaszewski 和 V. Podlozhnyuk，“通过 Ozaki 方案的扩展，在使用低精度张量核心时保证 DGEMM 精度”，载于《超级计算亚洲会议暨亚太地区高性能计算国际会议（*SCA/HPCAsia*）论文集》，2026年。[2] *cuBLAS* 库用户指南，第 *13* 版，NVIDIA 公司，2026年，浮点仿真相关章节；https://docs.nvidia.com/cuda/cublas/。[3] K. Ozaki、Y. Uchino 和 T. Imamura，“Ozaki 方案 II：使用整数模技术进行面向 GEMM 的浮点矩阵乘法仿真”，《国际高性能计算应用期刊》，2026年。[4] Y. Uchino 、 K. Ozaki 和 T. Imamura，“ 使用 INT8 矩阵引擎进行矩阵乘法的高性能低功耗仿真”，载于《*SC '25* 研讨会暨国际高性能计算、网络、存储与分析会议论文集》，2025年。[5] Y. Uchino、Q. Ma、T. Imamura、K. Ozaki 和 P. L. Gutsche，“基于中国剩余定理的复数矩阵乘法仿真”，载于《*ISC* 高性能计算 *2026* 研究论文集（第 *41* 届国际会议）》。IEEE，2026年，第 1–12 页。

[6] J. Demmel、G. Henry、I. Kozachenko、J. Langou、X. S. Li、J. Riedy 和 J. Vanover，“如何评定 BLAS 的精度”，《ACM数学软件汇刊》，2026年，在线发表于 2026年11月12日。[7] H. Wackernagel，《多元地统计学》，第 3 版。Springer，2003年。[8] K. Ozaki、T. Ogita、S. Oishi 和 S. M. Rump，“利用矩阵乘法快速例程及其应用实现无误差的矩阵乘法变换”，《数值算法》，第 59 卷，2012年。[9] H. Ootomo、K. Ozaki 和 R. Yokota，“整数矩阵乘法单元上的 DGEMM”，《国际高性能计算应用期刊》，2024年。[10] Y. Uchino、K. Ozaki 和 T. Imamura，“整数矩阵乘法单元上 Ozaki 方案的性能增强”，《国际高性能计算应用期刊》，2025年。[11] M. Fasi、T. Mary 和 M. Mikaitis，“仿真浮点矩阵乘法所需定点精度的更紧界”，《ACM数学软件汇刊》，2026年，在线发表于 2026年10月16日，文章编号 3858628；卷号和页码尚未分配。[12] J. W. Demmel 和 N. J. Higham，“采用快速三级 BLAS 的分块算法的稳定性”，《ACM数学软件汇刊》，第 18 卷，第 3 期，1992年。[13] J. Demmel、I. Dumitriu 和 O. Holtz，“快速线性代数是稳定的”，《数值数学》，第 108 卷，第 1 期，2007年。[14] J. J. Du Croz 和 N. J. Higham，“矩阵求逆方法的稳定性”，《IMA数值分析杂志》，第 12 卷，第 1 期，1992年。[15] E. Carson 和 N. J. Higham，“通过三种精度下的迭代细化加速线性系统求解”，《SIAM科学计算杂志》，第 40 卷，第 2 期，2018年。[16] A. Haidar、S. Tomov、J. Dongarra 和 N. J. Higham，“利用 GPU 张量核心进行快速 FP16算术，以加速混合精度迭代细化求解器”，载于 SC18：《国际高性能计算、网络、存储与分析会议》，2018年。[17] N. J. Higham 和 T. Mary，“数值线性代数中的混合精度算法”，《数值学报》，第 31 卷，2022年。[18] P. Luszczek、V. Gadepally、L. Anderson 等，“采用 INT8 中的双精度浮点仿真的分解型因子分解的性能与数值方面”，载于 IEEE高性能极限计算会议（HPEC），2025年。[19] E. Carson 和 Y. Ma，“Ozaki 矩阵乘法中的自适应精度驱动分裂及其在分块LU因子分解中的应用”，arXiv:2611.03158，2026年。[20] G. Ballard、A. R. Benson、A. Druinsky、B. Lipshitz 和 O. Schwartz，“改进快速矩阵乘法的数值稳定性”，《SIAM矩阵分析与应用杂志》，第 37 卷，第 4 期，2016年。[21] A. van der Sluis，“矩阵的条件数与均衡化”，《数值数学》，第 14 卷，1969年。[22] E. Anderson、Z. Bai、C. Bischof、S. Blackford、J. Demmel、J. Dongarra、J. Du Croz、A. Greenbaum、S. Hammarling、A. McKenney 和 D. Sorensen，《LAPACK用户指南》，第 3 版。SIAM，1999年。[23] R. K. Pace 和 R. Barry，“稀疏空间自回归”，《统计学与概率快报》，第 33 卷，第 3 期，1997年。[24] N. J. Higham，《数值算法的精度与稳定性》，第 2 版。SIAM，2002年。[25] IEEE，“IEEE浮点算术标准”，IEEE Std 754 2019，2019年。[26] S. M. Rump，“验证方法：使用浮点算术的严格结果”，《数值学报》，第 19 卷，2010年。

附录 A
设置与证据

a) 平台与库：四个 GPU 均运行相同版本的 Python wheel 包（nvidia-cublas 13.8.1.7、nvidia- cusolver 12.3.5.3、cudart 13.4.92）；H100 还运行 CUDA 13.2 模块中的库。驱动程序、时钟频率、功耗限制以及 NumPy 和 SciPy 版本列于补充材料中（制品中的 supplementary_tables.pdf）。GEM- Mul8 v3.5.4 使用其 INT8 后端，通过精确缩放处理 $q = 14$，16，20 个模数。主机 LAPACK 使用来自

PyPI wheel 包的 OpenBLAS；测试框架确认其在每项任务中都逐位尺度等变。

b) 路径：cuSOLVER potri 在 LAPACK 的因子上原生运行，也会在 ADP 下分别配合 DEFAULT 和 EAGER 运行，并在均衡因子上配合 EAGER 运行。分块 TRTRI 和 LAUUM 路径会将每次 GEMM 都交由同一个后端处理。递归不断减半，直至得到宽度为 256 的叶节点。LAUUM 要么是一次 X^TXGEMM，其两个操作数均为三角矩阵；要么采用相同的递归进行分块，从而不会产生两个操作数均不含三角结构的乘积；除标记为“分块 LAUUM”的路径外，所有路径都使用单次 GEMM。混合路径通过原生 DGEMM 计算每个 T或每个 X_{21}。基于求解的路径采用命题 2 中的结合方式。两种实现都使用下三角，cuSOLVER 路径也如此；每次更新和取负均作为单独操作执行（NumPy；cublasDgeam）。

c) 输入：缩放类为 DMD，其中 $M = W^TW/n + I$对应 $W \in \mathbb{R}^{n\times n}$，其元素相互独立且服从标准正态分布，而 k_i 在 $\{0,\dots,g\}$上均匀分布。随机种子 0 覆盖表 I 中 $n \in \{2048, 4096, 8192\}$ 处的网格；随机种子 1 至 4 在 $n = 4096$ 处重复$g \in \{0, 16, 32, 64\}$，并在 $n = 8192$处重复 $g = 64$；宽度运行还会加入 $n = 16384$。排序控制变量会对相同的指数重新排序。混合单位类通过单位因子对 M 的行和列进行缩放，因子取值介于 10^{-6} 和 10^6, 之间且不是 2 的幂；其对角线跨度约为 $g = 40$个数量级，即 2^{80}, 。

d) 参考值与指标：参考值 $D^{-1}X_MD^{-1}$与输入尺度一致。与使用 60 位十进制精度计算的 $n \le 512$ 和 $g \in \{0, 32, 64\}$的逆矩阵相比，它按范数的误差在 1.2×10^{-15} 以内，按分量的误差在 1.9×10^{-11} 以内；LAPACK 自身对 A求得的逆矩阵也如此。在其他情况下，$\widehat{\ }$参考值为 LAPACK 对 A求得的逆矩阵。 X 是对称计算逆矩阵。无需参考值的检验方法是计算缩放后的逆矩阵$\widehat{\ }$ $\widehat{\ }$残差：对于 $Y = DXD$，二的幂 D可精确表示，残差为 $\widehat{\ }$ $\widehat{\ }$ $\|I - YM\|_F/(\|M\|_F\|Y\|_F)$；每条完整路径均在设备上通过原生 DGEMM 计算该残差。

e) 计时：完整路径均在设备上运行，从上传矩阵到得到逆矩阵：包括 cuSOLVER 的 potrf（原生）、通过 cublasDdgmm 进行的任何二次幂均衡、求逆以及任何重新缩放。每条路径还会进行一次计时，将上传A 以及从可分页主机内存下载逆矩阵的时间计入计时。每条路径在全新进程中运行一次，并在运行期间采样设备内存。H100 进行了 3 次独立会话（独立作业，$n = 8192$、$g \in \{0, 64\}$），另有一次较早的会话提供了这些会话共有路径的更多数据；RTX 4090 仅在 $g = 64$ 下运行了 3 次会话。计时运行中不安装位计数指针。

f) 实现审计：表 VII 说明了各类证据所涵盖的内容。在 RTX 4090 ($n = 4096$,$g \in \{0, 64\}$) 上，11 条设备驻留路径中的每一条 TRTRI 路径在调用后都将乘积复制回主机，因此 L^{-1} 保持逐位一致；并使用测试框架的代码进行评估：设备上的最大 r_B^{up}/k 为 0.11（测试框架：0.12），而在 $g = 64$ 处，14 个模数下 T上的最大 r_A/k为 3.5×10^{13}（测试框架：3.5×10^{13}）。TRSM 叶节点原生执行：对于携带 DEFAULT、EAGER 或 FIXED 55 的句柄，叶节点 TRSM 会使用原生计算类型发出其内部乘积（4 种配置中为 4 种），并且

表 VII 证据覆盖范围。路径：计时路径，其最强链路为（H100 / RTX 4090）。

证据	执行与覆盖范围	路径
乘积精度等级	测试框架、 $n = 4096$、H100：每个输出元素的每个 TRTRI 乘积；每次运行抽取一个 LAUUM 分块	
全输出精度等级和包围区间	测试框架， $n = 4096$，两块图形处理器，30 个路由–扩散对和 4 种混合方案：每个 TRTRI 和 LAUUM 的乘积；来自它们的证书等级	7 / 4
其他逆矩阵误差	测试框架，所有图形处理器和尺寸： ϵ_X 仅	
设备审计	设备驻留，RTX 4090：11 个 TRTRI 乘积	0 / 5
已捕获的计时执行	路径： $n = 4096$、 $g \in \{0, 64\}$ 设备驻留， $n = 8192$, $g = 64$：每个乘积计时执行中的结果均复制回主机并进行评估；关键路径也逐次调用进行跟踪	3 / 4
计时	设备驻留， $n = 8192$，所有图形处理器：计时， ϵ_X 和残差；乘积不评分	9 / 7

发出的内核与原生实现完全一致（3 种中为 3 种）。命题 2 所假设的操作通过其可观测输入/输出等变性进行了逐位一致测试，测试位置为 $n = 4096$, $g = 64$。假设 (i) 对每个库中基于求解的路径的每个乘积均成立（224 项中为 224 项），而内部缩放改变了每个仿真乘积（160 项中没有一项保持不变）。

配对审计将每个 GPU（$n = 8192$, $g = 64$）上的失败计时路径与其修复方案联系起来：H100 上的 FIXED 55 和均衡化 FIXED 55（单 GEMM LAUUM），RTX 4090 上的 14 个模数和均衡化的 14 个模数；计时执行的每个乘积都被捕获，并根据其完整输出进行评分。LAUUM 的乘积在所有四条路径上均满足 $r_A/k \le 520$，而失败路径的 T 乘积距离 A 级最多差 3.6×10^{13}（H100）和 3.6×10^{13}（RTX 4090）；逆误差6.8×10^{-2} 和 7.1×10^{-2}，相较于 5.7×10^{-16} 和 2.1×10^{-15}已修复：损失出在计时执行的 T 乘积中，与测试框架中的情况一致。

在每个 GPU 上，四条关键路径均在独立作业中重新运行，并对每次 GEMM 调用的每个内核进行跟踪：RTX 4090 上 5610 次跟踪调用中的 5610 次，以及 H100 上 3740 次跟踪调用中的 3740 次，其内核序列均与捕获的执行一致；此外，每条路径返回的逆矩阵在未跟踪、计时、跟踪和捕获运行之间都完全一致。补充材料给出了完整的审计结果和逐路径证据矩阵。

g) 一个遗留设置：设置恢复为 ADP 后，cuBLAS 仍会在句柄上保留 FIXED 配置的最大尾数位数，而测试框架每个进程只使用一个句柄，因此没有将其恢复。于是，在同一进程中紧随 FIXED 路径运行的动态路径会在 55 或 79 位的上限下运行：在 H100 上，$n = 4096$, $g = 32$ 的 EAGER 在全新进程中的包围区间为 6.0×10^{-14}，而在 FIXED 55 之后则为 1.4×10^{-13}。现在，测试框架会在每次仿真调用时恢复并断言默认设置，同时记录每条路径的有效上限。所有产生动态路径打印数值的测试集均已重新运行；如果某个打印数值来自受限行，制品的来源检查就会失败。重新运行后，EAGER 在 $g = 32$ 和 64 处的等级恢复到其在$g = 0$ 处的水平，并且误差至多移动了最后一位打印数字；

被取代的记录和重新运行列表都在制品中。

附录B 理论全文

a) 命题 *1*：上界所允许的范围：对于本文的 $f(k)=k$，该上界允许达到 $|\delta|$，上限为 2^{g-52}；在 $g=64$ 时，$\delta=0.07$ 已经给出 0.035。A 级最多允许 $f_A(2)\ up$，其中同一项上的 $p=2$，而 $(|L^{-1}||L||L^{-1}|)_{31}$，对经典三角求逆给出的上界 [14], [24]，对每个g 都是 4；脚本使用有理数精确计算检查了 g 从 0 到 64 的全部情况。该命题讨论的是上界所允许的范围，而不是任何库，也不是 L^{-1} 的误差；下面的测量关注的是 $Y=X^TX$，扰动以相同阶数达到该量。

b) 定理 *1* 的证明：(a) 中一阶以后的余项为 $R=-\sum_s \Lambda_s \Delta_{22} L_{21} \Delta_{11} \Gamma_s$。(c) 对于给定的实现，并未说明框架量如何随 D 变化。

证明：（a）$X_{21}=-X_{22}L_{21}X_{11}$ 以及 $\widehat{X}_{21}=-\widehat{X}_{22}L_{21}X_{11}+\Phi_s$；代入 $X_{ii}=\widehat{X}_{ii}-\Delta_{ii}$ 以及 $\widehat{X}_{22}L_{21}=T-E_s$ 即可得到（2）。显然，$\Delta=X(L\Delta L)X$；根据（2），并在每个节点应用 $L_\nu X_\nu=I$，$L\Delta L$的分块（ρ_2，ρ_1）为 $L_{22}\Psi_s L_{11}$（下方的单位矩阵）。（b）对（2）$\widehat{Y}-Y=\widehat{X}^T\Delta+\Delta^T\widehat{X}-\Delta^T\Delta+G+\varrho$ 取绝对值，$|G|\le \mathcal{G}$ 从叶节点开始归纳；对于 sym()，其中 sym() sym()（除非它是对称的，否则不为 $\mathcal{G}$），且对称化的舍入 ϱ 逐元素满足 $|\varrho|\le 2u|\widehat{Y}|+2^{-1074}$：加法的误差至多为相对 u，如果结果为次正规数则运算是精确的；二等分的误差至多为 2^{-1075}。（c）代入 $L=DL_M$ 和 $\widehat{X}=\widehat{X}'D^{-1}$：每个 D 都会抵消，例如 $B_{22}|L_{21}X_{11}|D_1=B'_{22}|L_{M\,21}X'_{11}|$ 以及 $|X_{22}||L_{21}|=c\,u\,r_i s_j$。在容差上采用 B 级时，其值至多为 $|\widehat{X}'_{22}||L_{M\,21}|$ $D=I$处值的 ρ_s 倍，其中包含 $r_i=\max_k |x'_{ik}|/d_k$ 和 $s_j=\max_k d_k|l_{M,\,kj}|$ over$k\in\rho_2$。 ∎

(2) 是 Du Croz 和 Higham 对方法 2B 进行残差分析的前向误差对应形式，保持精确且允许任意乘积误差。附录 C 评估了 (b) 的一种更紧的形式，其中保留了 Λ_ν 和 Γ_ν。(c) 有三个重要的局限。$D|Y-Y|D$作为框架的计算量的函数，其包围区间不含 D，这是因为这些计算量位于 B' 和 $\mathcal{E}_s$ 之内，且与 D-free L_M^{-1} 和$L_M^{-1}{}_{,\,22}L_{M,\,21}$相差不超过相应容差，因此该包围区间在容差的一阶范围内独立于 D。未缩放的按范数 ϵ_X 会按 D^{-1} 对各元素加权，因此不具备这一性质。此外，该条件是充分而非必要的；命题 1 达到了 ρ_s 增长，因此仅凭 B 级条件无法去除 D。

单位矩阵。固定递归中索引集为连续索引集 $\nu=\rho_1\cup\rho_2$的一个节点。由于 L 和 Δ 是下三角矩阵，因此 $L\Delta L$ 的分块（ρ_2，ρ_1）只涉及 ν 中的索引，并等于 $L_{21}\Delta_{11}L_{11}+L_{22}\Delta_{21}L_{11}+L_{22}\Delta_{22}L_{21}=L_{22}(\Delta_{21}+X_{22}L_{21}\Delta_{11}+\Delta_{22}L_{21}X_{11})L_{11}$。将 $X_{22}L_{21}X_{11}$ 用 $\widehat{X}_{ii}=X_{ii}-\Delta_{ii}$ 展开可得 $\Delta_{21}=\Phi_s-X_{22}L_{21}\Delta_{11}-\widehat{\Delta}_{22}L_{21}X_{11}+\Delta_{22}L_{21}\Delta_{11}$，这就是式（2）；再次使用 $X=X+\Delta$ 后得到 $\Delta_{21}+X_{22}L_{21}\Delta_{11}+\Delta_{22}L_{21}X_{11}=\Phi_s-\Delta_{22}L_{21}\Delta_{11}=\Psi_s$。每个严格下三角元素都位于恰好一次分割的分块（$\rho_2$，$\rho_1$）中，而下三角矩阵的其他每个元素都位于一个叶节点中，其中 $L\Delta L$ 的分块为 $L_\ell\Delta_\ell L_\ell$。因此 $\Delta=X(L\Delta L)X=\sum_\nu X_{:,\,\rho_2}L_{22}\Psi_\nu L_{11}X_{\rho_1,\,:}$。这两个恒等式都是多项式恒等式；脚本对式（2）进行符号检查，并在 66 个随机案例上将二者作为精确有理数等式进行验证。

所评估的上界。$\widehat{X}^T\Delta+\Delta^T\widehat{X}+\Delta^T\Delta+G$ $X^T\Delta=\sum_\nu X^T\Lambda_\nu\,\Psi_\nu\Gamma_\nu$ $X=$对称( )和 ( )。取自 (3) 的 $X-\Delta$ 和 $|\Delta|\le B$，

$$
\begin{aligned}
|X^T\Lambda_\nu| &\le |\widehat{X}^T\widehat{X}_{:,\rho_2}L_{22}|+|\widehat{X}|^T B_{:,\rho_2}|L_{22}| \\
&\quad + B^T\big(|\widehat{X}_{:,\rho_2}L_{22}|+B_{:,\rho_2}|L_{22}|\big) =: \mathcal{Y}_\nu,\\
|\Gamma_\nu| &\le |L_{11}\widehat{X}_{\rho_1,:}|+|L_{11}|B_{\rho_1,:} =: \bar{\Gamma}_\nu,\\
|\Psi_s| &\le \mathcal{E}_s|\widehat{X}_{11}|+\mathcal{F}_s+B_{22}|L_{21}|B_{11} =: \bar{\Psi}_s,
\end{aligned}
$$

和 $\Psi_\ell=B_\ell$，因此

$$
|\widehat{Y}-Y| \le \sum_\nu\big(\mathcal{Y}_\nu\bar{\Psi}_\nu\bar{\Gamma}_\nu+(\cdot)^T\big)+B^TB +\operatorname{sym}(\mathcal{G})+2u|\widehat{Y}|+2^{-1074}\mathbf{1}\mathbf{1}^T,
$$

最后两项涵盖了对称化时的舍入，与定理 1(b) 中相同。没有绝对值项时，上界不成立：对于 $X=X=I_2$、$B=0$ 以及一个非对角元素为 2α 和 α的输出误差 G，二等分将 $3\alpha/2$ 舍入为 2α，该值大于 sym($\mathcal{G}$)$_{12}+2u|Y_{12}|$；自测试会检查这个例子。求值器会将绝对值项提高到其下限 $\tau=2^{-500}$，这是一个更大、因而有效的容差。容差的对称化很重要：对于 $\widehat{X}=I_2$ 和 $G=e_1e_2^T$，sym(G) 在位置 (2, 1) 上的元素为 $1/2$，而包络 $|G|$ 为零。此处使用的容差 $\min(r_A^{\mathrm{up}}u|\widehat{X}|^T|\widehat{X}|,r_B^{\mathrm{up}}u\,cc^T)$，其中 c是 $|X|$ 的列最大值，并且整个乘积共用一对比率，在精确算术下是对称的；不过，代码仍会将收到的任何容差替换为 $\max(\mathcal{G},\mathcal{G}^T)\ge$sym($\mathcal{G}$)，并断言结果为对称矩阵（此步骤之前计算所得容差的最大相对非对称性：0）。容差中的一阶项会在取绝对值之前形成 $X^T\Lambda_\nu$ 和 Γ_ν，并以本次运行的 X 代替精确逆；B 仅在二阶项中出现。

叶节点。对于 $N\ge|X_\ell L_\ell-I|$和 $\|N\|_\infty=\nu_\ell<1$，$\Delta_\ell=\widehat{X}_\ell L_\ell-I$ $\widehat{X}_\ell-\Delta_\ell$ $|\Delta_\ell|\le I-N^{-1}N|\widehat{X}_\ell|\le$( )( ) 给出 $N|\widehat{X}_\ell|+\frac{\nu_\ell}{1-\nu_\ell}Ne\;e^T|\widehat{X}_\ell|=B_\ell$ ( ) ( )( )：。残差以长双精度计算，N包括其舍入，$\gamma^{ld}_{n_b+1}(|\widehat{X}_\ell||L_\ell|+I)$；在下面的运行中，$\nu_\ell\le 4.6\times10^{-16}$。

c) 命题 *2* 全文：在方法 1B 中，每次更新都通过 GEMM 使用 $\beta=0$ 形成$P=L_bZ_1$，然后作为独立运算进行减法，W_2-P；叶节点是宽度为n_b的原生 TRSM。完整表述如下：在 L_M以及 DL_M 上运行基于求解的递归，D 为二次幂对角矩阵；两者采用相同的分块方式和相同的运算序列，使用同一种二进制浮点格式和舍入模式 [25]（在所测试的库中为 binary64、就近舍入），且不存在改变控制流的陷阱或标志位，D 和 DL_M 的元素均为精确值。假设在两次运行中：(i) 每个乘积均通过具有外等变性的 G计算；(ii) 每次更新和每次取负均对每个元素执行一次 IEEE 运算，既不与其后所跟的乘积融合或合并，也不重新结合；(iii) 叶节点具有等变性：对于 D_aL_a，求逆返回 $\widehat{X}(L_a)D_a^{-1}$；对于 (D_aL_a，D_aWC)，求解返回 $Z(L_a,\,W)C$，D_a、C

二次幂对角矩阵；(iv) 每个例程都是确定性的，并且两次运行采用相同的计算路径，每次归约内部的运算顺序也相同；此外，没有任何运算（包括中间运算和对称化）发生溢出、下溢、返回次正规数或引发其他异常。然后 $\widehat{X}(DL_M) = \widehat{X}(L_M)D^{-1}$ 逐位一致，并且对 G 执行带对称化的 LAUUM 返回 $D^{-1}\widehat{Y}(L_M)D^{-1}$。如果还满足 (v)，即 DMD 的计算所得 Cholesky 因子是 D 乘以M的 Cholesky 因子，则 DMD的计算所得逆矩阵逐位一致地等于 D^{-1}乘以 M再乘以 D^{-1}（± 0 相等）。

证明：对递归进行归纳；叶节点满足 (iii)。若 $\widehat{X}_{ii} = \widehat{X}_{M,ii}D_i^{-1}$，则依据 (i)，$W = G(D_2L_M$,21, $\widehat{X}_M$, $_{11}D_1^{-1}) = D_2W_MD_1^{-1}$。在求解过程中，叶节点将 ($D_aL_{M,a}$, $D_aW_{M,a}D_1^{-1}$) 映射为 $\widehat{Z}_{M,a}D_1^{-1}$，依据 (iii)；一次更新形成 G ($D_bL_{M,b}$, $\widehat{Z}_{M,a}D_1^{-1}) = D_bP_MD_1^{-1}$，而对带有相同二的幂因子的操作数进行正确舍入的减法或取负，在 (ii) 和 (iv) 下会返回因子乘以未缩放的结果，因此 $\widehat{X}_{21} = \widehat{X}_{M,21}D_1^{-1}$。LAUUM 的 $G(\widehat{X}^T, \widehat{X})$ 仅在外部索引上进行 D^{-1}，且 $(Y + Y^T)/2$ 与缩放可交换。(v) 完成整个流程。 ∎

(i)、(iii)、(iv) 和 (v) 是库例程的性质，文档并未保证这些性质；附录 A 对输出进行了逐位检查，这为所测试的版本、配置和输入提供了证据，但不构成对其他情况的保证。在等变性 $\widehat{Y}(DMD)$–$(DMD)^{-1} = D^{-1}(\widehat{Y}(M)$–$M^{-1})D^{-1}$ 下，每个元素的相对误差，以及在 M 的坐标系中测得的误差，都等于 M 求逆时的误差，无论D如何；按范数计算的 ϵ_X 不一定恒定：对于 $M = I$，在 M 的逆矩阵中的误差为 $\eta e_1e_1^T$，且 $D =$diag$(1, 2^g, 2^g)$，它 从$|\eta|/3$ 上升至接近 $|\eta|$。乘积形式不具备这一性质，因为外部等变性并未说明沿缩并指标共享的缩放。在第三节的模拟器运行中，使用 L_{11} 的求解是原生的；采用原生乘积时，方法 2C 和乘积形式都逐位尺度等变且准确（10 次运行中 10 次如此）。

d) 三角形： 此处的 LAPACK 指的是参考 dtrtri，其特征在于源代码级别的关联；本文未检查供应商实现，因此不对其作任何声称。对于 uplo 的两个取值，参考 dtrtri 会先将已经求逆的分块乘入三角形的非对角分块（DTRMM），然后使用对角块求解（DTRSM），缩并三角形的行索引；它从不调用 DGEMM，因此只有在这两个例程本身经过仿真时才会受到影响。下三角因子 $L = DL_M$ 在其行上携带D，上三角因子 $R = R_MD$ 在其列上携带。第六节在第三节的仿真器模型中陈述了上分支的结果，其中三角乘积经过仿真，而求解和叶节点采用原生实现；这是一项关于该模型中参考关联的陈述。它并不意味着将 uplo= ‘U’作为修复方案：缩并指标还取决于库如何构造其三角乘积和执行求解，以及如何对其对角块求逆；例如，针对单位矩阵进行左三角求解时，会缩并三角形的列索引，而对于 R而言，这正是经过缩放的索引。编译版 LAPACK 和 采用上三角的 cuSOLVER 未在本文中进行测量。

e) 一阶反事实：在包围区间之前，使用 LAPACK 的 binary64 逆矩阵替代精确逆矩阵，并采用 H100 的分级比率（LAUUM 的分级比率取自其抽样块），计算了一阶项。该数值不是上界。在 30 对中的 30 对里，它都超过测得的 ϵ_X，超出幅度为 23 至 9,880 倍。将原生 DGEMM 的分级比率赋予替代方案原生计算的乘积，而无需重新运行；在 38 个方案中，对于每个以原生级别测量的方案，该值至多为 1.2×10^{-13}，对于其他每个方案则至少为 8.7×10^{0}；对于 14 个模数，在 $g = 64$ 时，它会从 8.0×10^{1} 降至 1.2×10^{-13}，其中每个 T 均为原生，并且在每个 X_{21} 均为原生时保持在 8.0×10^{1}。这可以在不运行替代方案的情况下对其排序；但不能证明任何事情。

附录 C 分级与认证

a) 分级：表 VIII 对 $n = 4096$ 下分块求逆的乘积进行分级：所有路径和 g 上的全部 900 个 TRTRI 乘积，均检查每个输出元素；LAUUM 的 $n \times n$ 乘积则在每次运行中随机抽取一个 512×512 输出分块进行检查（共 30 个分块）。参考值是顺序 x87 长双精度乘积，其误差的上界将在下一段给出，为 $\gamma_k^{ld}p_{ij}$。因此，分级会逐个元素给出真实比率的上界，即 $r_B^{\mathrm{up}} = \max_{ij}(\lceil c_{ij} - c_{ij}^{ld}| + \gamma_k^{ld}p_{ij})/(ur_is_j)$，并给出其上界和下界；独立的向外舍入脚本重新计算这些界，结果精度在 8.7×10^{-16} 以内。任何分类都不受参考值自身误差的影响（A 级和 B 级判定均无变化），将容差加倍至 γ_{2k}^{ld} 也不会影响分类（分别为 0 和 0）。对全部十条路径、所有三种 g 及每个分级元素取最大值后，r_B^{up}/k 在 450 个 T乘积上至多为 0.052，在 450 个 X_{21} 乘积上至多为 0.12，在抽样的 30 个 LAUUM 分块上至多为 0.019。第二次运行对每个乘积的每个输出元素进行分级，包括 LAUUM 的 $n \times n$ 乘积，并将容差加倍：在 H100 上，r_B^{up}/k 至多为 0.053、0.12 和 0.033；在 RTX 4090 上则至多为 0.05、0.11 和 0.032（失败的乘积为 0 和 0）。每一步的比率都在长双精度下向外舍入，包括转换为 binary64 的步骤，因此存储比率构成真实比率的上界。比率是针对所有参与分级的输出元素计算的，零也包括在内；右端为零的元素，若其误差为零，则计为 0，否则视为失败。

b) 参考舍入模型：参考值是 NumPy 的长双精度矩阵乘积，其背后没有 BLAS：这是一个编译后的循环，对每个条目，将 k 个乘积逐一加到从零开始的总和中。在 x86-64 上，该类型采用 80 位 x87 格式，其中乘法和加法都会各自舍入一次，得到 64 位有效数；而 x87 不支持融合乘加，因此不可能进行合并。于是，一个项的相对扰动至多是 k 个因子的乘积：$1 + \delta$、$|\delta| \le u_{ld}$（其中一个因子来自该项的乘积，加法之后的每次加法至多贡献 $k - 1$ 个因子），误差至多为 $\gamma_k^{ld}p_{ij}$ [24, 第 3 节1], 在 r_A/k 单位下可忽略（至多为 4.9×10^{-4}），但对于一般操作数，在较大的 r_B/k 单位下则并非如此：舍入后的乘积再进行舍入后的加法，其标准界已经是 k，无论求和顺序如何；而融合累加具有

相同的界。一个脚本将库乘积与长双精度标量中按上述明确序列计算的结果逐位比较，并与使用有理数构造的正确舍入融合累加进行比较：540 个条目中的 540 个与前者相等，而两种模型在其中 219 个条目上存在差异；最大误差为 $\gamma_k^{ld} p_{ij}$ 的 0.37 倍。该模型取决于平台和构建版本：类型必须是 x87 80位格式，循环不得进行向量化、重新结合，也不得以更宽或更窄的精度求值，并且 x87 精度控制必须保持其 64 位默认值。分级代码会在类型不具有 64 位有效数时停止；每个完整输出任务还会先在其自身的主机和构建版本上执行逐位比较，如果有任何一个条目不同，就会停止且不进行任何分级（8 台主机上的 18 个任务，共 9,720 个条目中的 9,720 个相等）。因此，在长双精度为 binary64 或二进制128位的平台上，分级会安全失败。

c) 分量误差： 分量误差是在缩放值$d_i d_j x_{ij}$| 超过最大值的 10^{-8} 的各个条目中取 max$\lceil x_{ij} - x_{ij}|/|x_{ij}|$，因此，自身尺度下低于舍入误差的条目以及零值都不会被计入。在透明CPU Ozaki-I模拟器（7位切片、精确整数切片乘积、 $n = 2048$）上，最大的 TRTRI e_{99} 从 $g = 0$处的 8 上升到 $g = 64$处的 67，增幅由缩放项承担。

d) 证书： 每个乘积的误差均可按条目界定为 min($r_A^{\mathrm{up}} u|A||B|, r_B^{\mathrm{up}} u\, rs^T$)，并使用其自身的分级比率，因此可以根据一次运行自身的操作数来计算附录B的上界：该因子 L 由路径及其计算出的叶节点、乘积和逆矩阵给出，并利用叶节点的残差封闭其误差。求值器使用 binary64 并采用就近舍入。只有当每个操作的求值都符合所使用误差界的假设时，它才会返回证书；否则将拒绝并不返回任何数值。

- 不变量。 它形成的每个非负矩阵在进一步使用前都会经过检查和规范化：如果某个元素为 NaN、无穷大、负数或大于 2_{500}，则拒绝该元素；对于非零元素，如果其小于 $\tau = 2^{-500}$，则将其提升至 τ。这是可靠的，因为每个界都关于每个非负操作数单调。带符号输入必须位于 $[2^{-480}, 2^{480}]$ 中，因此矩阵乘积内部的乘法不会发生下溢，也不会溢出。

- 扩展运算。 每个非负操作数的乘积和求和都会乘以一个因子，该因子涵盖任意求和顺序下的舍入误差，无论是否使用融合运算 [24]；绝对乘积 $|AB|$都包含在 |fl(AB)| 加上 $\gamma_k|A||B|$ 的范围内；容差 min($r_{\mathrm{up}\,A}\, u|A||B|$， $r_{\mathrm{up}\,B}\, u\, rs^T$) 带有 $1 + 8u$ 的余量。
- 步骤。 (1) 根据长双精度残差计算叶节点界，除非满足 $\nu_\ell < 1/2$，否则拒绝；(2) 按照 (3) 进行拆分，并优先处理子节点；(3) 对拆分和叶节点求和，采用上述更精确的形式；(4) 对称化 LAUUM 的容差，并计入对称化过程中的舍入误差；(5) 两个弗罗贝尼乌斯范数 $\frown\beta \geq \|W\mathcal{B}W\|_F$和 $\eta \leq \|WYW\|_F$，在路径的输出缩放 W中分别通过定向平方根计算，每个范数都由精确平方的精确求和构成；(6) 在有理数精确计算中计算 $\beta/(\eta-\beta)$。

当该商小于 1 时，状态为已认证；当 $\eta \leq \beta$ 时，状态为无相对界；任何检查失败时，状态为拒绝。根据不变量可知，每次扩展都作用于满足标准模型的运算，因此根据归纳，每个归一化矩阵都大于或等于其所代表的精确量；步骤 5 和 6 不引入舍入。该论断基于 BLAS 和 NumPy 中的 IEEE binary64 算术运算及常规乘积、评分参考实现的 x87 模型，以及所用路径根据测试框架传入的操作数构造乘积；步骤 1 的残差论证是验证方法 [26] 中的标准论证。补充材料逐项说明了求值器、一个参考过程和测试向量。

e) 如何检查求值器：在有理数精确计算中，对 66 个随机三角系统进行测试；对角线跨度最高为 2^{40}，乘积误差最高为 B 级右端项的 0.2 倍，$\widehat{\ }$在全部情况下（ 66/66 ）评估界逐项支配 $|\Delta|$ 和 $|Y - Y|$。在 CPU Ozaki-I 模拟器上进行的 7 个案例中，界在每个案例中都占优，且比实际误差大 40 到 402 倍。对抗测试分别向 150 个中间值中的每一个注入 NaN、无穷大或负值（共 444 次注入，全部被拒绝）；两个求值器对 94 个测试向量全部通过，且获得精确的预期结果（0 次遗漏）。独立的第二个评估器与其不共享任何代码，在 H100 上的 34 对样本以及 RTX 4090 上的 34 对样本中，状态和界均一致（最大相对差异为 3.9×10^{-9}）。在$n = 64$ H100 上 42 次运行和 RTX 4090 上 42 次运行的操作数均已存储，并使用精确整数运算重新计算了每个界：精确界逐项支配真实误差，且求值器的状态与精确状态一致（42 次和 42 次运行均无差异）。通过预先选定规则抽取的 6 次运行和另外 5 次 $n = 4096$ 运行，均使用已存储的全部 GPU 输出在 CPU 上仅根据这些输出重建并重新运行：抽取的 6 次运行和全部 5 个见证样本均逐位一致地复现了其状态和界；在独立的 x86-64 主机上，5 个见证样本的状态也全部复现。对评估主机进行的预检（6 项检查）为每份证书提供门控。档案、其 SHA-256 值及选择规则均载于该制品的 README 和补充材料中。

f) 结果： 表 VIII 列出了每个 H100 对的包围区间。没有任何一对被拒绝，也没有一对的有限相对界达到 1 或更高：没有证书的 4 对全部未通过分母检验，其 β/η 在 H100 上从 1.3 到 3.6×10^{17}。在 M的框架中，原生路径的包围区间在 $g = 0$、32 和 64 处分别为 6.5×10^{-13}、6.5×10^{-13} 和 6.5×10^{-13}。仅一阶部分（不含叶节点、 B 和舍入项）在包围区间低于1.0×10^{-6}的所有情况下都与包围区间相差不超过 1.8%；叶节点的贡献最多为 2.7×10^{-15}。超过该阈值后，矩阵 B在递归的每一层都取绝对值，使二阶项占据主导：FIXED 79 在 $g = 64$ 处经认证比其一阶部分高出 272%，而 $g = 64$处的 20 个模数的一阶部分低于 1，因此没有相对界。

附录 D 完整结果

a) 等级与包围区间：表 VIII 给出了三个扩散下的全部十条路径；对 LAUUM 的每个条目也进行分级后，B 型比率至多为 0.12；原生乘积和 ADP 乘积在每个 下均满足

$r_A/k \le 8.7\times10^{-2}$。对于最差 A 型比率（针对 TRTRI 乘积）超过 100 的 9 个路由–扩散对，逆矩阵误差为该比率的 0.11 到 18 倍 u倍。每个 T 均为原生时，包围区间至多为 1.5×10^{-13}；每个 X_{21} 均为原生时，不存在相对界。

b) 四个 *GPU*：表 IX 列出了四个 GPU 在 $n = 16384$下的所有路径。每单位 g 损失的位数从 14 个模数的 0.76 到 FIXED 79 的 0.96 不等，其起始点与单乘积分析 [11]一致；增加位数会降低误差，FIXED 111 的精度与原生精度仅相差一个较小的倍数$g = 64$。H100 上的斜率与 RTX 4090 相比，每单位 g最多相差 9.4×10^{-5} 位；在 $n = 4096$ 下，随机种子 0 到 4 的斜率最多相差 0.01（H100）和 0.01（RTX 4090）。H100 上，随机种子 1 到 4 在 $n = 8192, g = 64$ 时，14 个模数的结果为 7.1×10^{-2} 至7.6×10^{-2}，均衡化 FIXED 55 的结果为 2.7×10^{-16}至 2.9×10^{-16}；其他 GPU 的表现相近（补充材料）。将指数分成连续分块时，16 个模数得到 9.9×10^{-4}；EAGER 无论排序后还是分块处理，均保持原生精度（4.9×10^{-15}, 3.5×10^{-15}）。

另外两项检查也得出一致结论。对 C 级类型进行的范数审计，在 $n = 4096$处的每个 TRTRI 乘积上均通过：$g = 64$（$\|C - C_h\|_F \le \beta u\|A\|_F\|B\|_F$ 与主机 DGEMM 的结果比较，其上界约为 $\beta = k$）。缩放残差无需参考值，也能像 ϵ_X 一样准确区分不同路径：在 H100 上，所有准确的完整路径均不超过 5.9×10^{-17}，而所有失败路径均不低于 1.9×10^{-3}（RTX 4090 上则分别为 5.9×10^{-17} 和 1.9×10^{-3}）。逐位一致的尺度等变性是一种无需参考值的征兆：在每个 GPU 上，原生路径均通过了全部 30 项检查；在 RTX 4090 上，所有带有$g > 0$ 的原始固定路径均失败，而在 H100 上，除 1 个（路径、输入）组合外，其余均失败。EAGER 路径也失败了（带有$g > 0$的 15 个案例全部失败），尽管结果准确；因此，失败表明路径发生了变化，而非出现了误差。

当困难源于条件数问题而非缩放时，仿真结果与原生执行相当。在 $\kappa = 10^8$下，potrf 后接 potri 与 LAPACK 的差异，在 RTX 4090 上原生执行为 5.3×10^{-9}、在 EAGER 下为 5.3×10^{-9}（在 H100 上分别为5.4×10^{-9} 和 5.4×10^{-9}），差异在数量级上仅为 κu；高斯过程核在两种方式下均给出 3.1×10^{-7}。在 H100 的第二个版本中，即 cuBLAS 13.3.0 和 cuSOLVER 12.1.0，所有 cuSOLVER 误差均在固定版本误差的 1% 以内。

c) 替换与重新结合：表 X 将图 1(c)扩展为 $n = 8192$、LAUUM，以及超额误差最大和最小的 16 个乘积。即使只将任意一个 T 设为仿真，其余全部保持原生，在$n = 8192$下仍会得到 1.9×10^{-3} 或更差的结果。在透明 CPU 仿真器（$n = 2048$，8 个切片）上，DMD 的基于求解的逆，在每个测试过的 g 上（5 次中 5 次）都与其对 M 的逆乘以 D^{-1} 逐位相同，误差范围为 4.2×10^{-14} 至 6.0×10^{-14}，而乘积形式的误差可达 8.9×10^{-2}。通过测试框架，在$n \in \{4096, 8192\}$ 下将 g 从 16 增至 64 时，在 H100 上，14 个模数的 8 个案例全部逐位尺度等变，FIXED 55 的 8 个案例也全部如此；RTX 4090 上同样如此（分别为 8 个和 8 个案例中的 8 个），cuSOLVER 的 potrf 所构成的每个完整设备流水线也都如此（5 次中 5 次；乘积形式：2 次中 0 次）。在 H100 上，对原始输入的误差从未超过 5.5×10^{-15} 和 9.8×10^{-16}，而乘积形式的误差可达 7.1×10^{-2}；在混合单位下，误差至多为 2.7×10^{-15}。

d) 实矩阵和库版本：这 16 个 SuiteS- 解析矩阵是 NIST Matrix Market 镜像中同时满足 $1024 \le n \le 8192$ 且对角线跨度至少为 2^{20} 的矩阵；其选取规则在查看任何矩阵之前就已固定。协方差使用 1990 年加利福尼亚州人口普查（七个变量）和 UCI 空气质量数据（十二个变量），采用原始单位。表 XI 给出了各组最大值。均衡化 FIXED 55 在每个矩阵上都保持在原生执行的 2.2 倍以内。均衡化的 16 个模数在每个矩阵上都保持在原生执行的 1.0 倍以内；均衡化的 14 个模数在 A100 上的 bcsstk23 中达到原生执行的 138 倍，而在其他每个矩阵上最多为 10 倍（在 A100 上为 9 倍）。在 9 个版本中，保护路径相对于原生执行的最大倍数，在 L40S 上为 1.01，在 A100 上为 1.00，在 H100 上为 1.00。均衡化 FIXED 55 从 cuBLAS 13.7 起达到原生水平（在 L40S 上为 2.9×10^{-16}），并在此之前的 1.3×10^{-14}处停止；在 H100 上，这一变化在 13.8 中出现（13.7 下为1.3×10^{-14}，13.8 下为 2.9×10^{-16}）。cuSOLVER 在其自身的 FIXED 控制下对 potri 给出 1.2×10^{-2}、1.7×10^{-7}和 9.8×10^{-16}（在 L40S 上分别使用 55、79 和 111 位），在 RTX 4090 上使用 55 位时为1.2×10^{-2}；均衡化后为 6.7×10^{-16}。补充材料列出了每个矩阵和每个版本。

附录 E 防护机制：估算值、切换点与 DEFAULT

a) 一种解释性代理指标：对于 $C = AB$，其中 $c_{ij} \neq 0$ 和 r_i、s_j、p_{ij} 如第 II 节所述，超额 $e_{ij} = \log_2(r_i s_j/|c_{ij}|)$ 是 $\log_2(r_i s_j/p_{ij})$、A 级与 B 级右端之比，以及 $\log_2(p_{ij}/|c_{ij}|)$（普通消去）之和。测试框架记录其第 99 百分位数 e_{99} 和每个乘积的最大值。它需要该乘积，因此只能事后解释库的选择，而不是在线估算。在两块 GPU 上，缩放网格中的全部 636 个 TRTRI 乘积都符合图 3 的拟合；在 $n = 8192$ 时，它们在 $g = 0$ 报告为 68 至 75 位。该图绘制了 RTX 4090 上 $n \in \{2048, 4096, 8192\}$、$g \in \{0, 16, 32, 48, 64\}$ 条件下 EAGER 的每个 TRTRI 乘积，包括原始数据和平衡的数据；H100 的估算值相同。按分量来看，在整个网格上，任何 ADP 路径的误差与其原生对应项的误差之比均不超过 1.95（RTX 4090）或 1.56（H100）。

b) 乘积在何处切换路径：cuBLAS 会将其位估算值写入调用方提供的输出指针。无论乘积是否采用仿真，该指针都会保存估算值（cuBLASLt 文档说明了 -1 不使用仿真的情况），因此必须根据输出、耗时和内核来判断分派情况。图 4 中的探测结果将每个输出与原生 DGEMM 逐位比较。在两款 GPU 上，跨度为零时估算值为 62 位，跨度为 200 位时为 257 位。在配备 cuBLAS 13.8.1 的 RTX 4090 上，1024×1024 及更小的输出会在 107 至 109 位之间切换（例如，从原生性能的 5.59 倍变为 6.51 倍）；2048×2048 的输出则在 107 至 110 位之间切换（从至多原生性能的 0.98 倍变为至少 1.60 倍）。超过切换点后，2048×2048及更大的输出与 RTX 4090 上的原生 DGEMM 逐位一致；较小的输出则不一致，因此切换点以上的路径并不总是默认 DGEMM 内核。在

表 VIII TRTRI 乘积的等级、逆矩阵误差以及 $\|\widehat{Y} - (LL^T)^{-1}\|_F / \|(LL^T)^{-1}\|_F$(H100, $n = 4096$) 的认证包围区间。全程采用 $X_{21}, r_A/k \le 1.2 \times 10^1$。无相对界：无相对界。

路径	r_B/k，全部 g	$g=0$ r_A/k 关于 T	$g=0$ ϵ_X	$g=0$ 包围区间	$g=32$ r_A/k 关于 T	$g=32$ ϵ_X	$g=32$ 包围区间	$g=64$ r_A/k 开启 T	$g=64$ ϵ_X	$g=64$ 包围区间
原生	1.1×10^{-1}	1.5×10^{-2}	2.3×10^{-15}	6.5×10^{-13}	1.5×10^{-2}	2.2×10^{-15}	1.5×10^{-13}	1.5×10^{-2}	2.0×10^{-15}	1.1×10^{-13}
DEFAULT	1.1×10^{-1}	1.5×10^{-2}	2.3×10^{-15}	6.5×10^{-13}	1.5×10^{-2}	2.2×10^{-15}	1.5×10^{-13}	1.5×10^{-2}	2.0×10^{-15}	1.1×10^{-13}
EAGER	7.1×10^{-3}	7.2×10^{-3}	2.1×10^{-15}	2.3×10^{-13}	9.1×10^{-3}	2.0×10^{-15}	6.0×10^{-14}	1.3×10^{-2}	2.0×10^{-15}	6.6×10^{-14}
q=14	5.3×10^{-2}	2.8×10^{1}	3.0×10^{-15}	2.9×10^{-12}	1.1×10^{9}	8.1×10^{-8}	3.1×10^{-5}	3.5×10^{13}	7.1×10^{-2}	无相对界
q=16	3.9×10^{-3}	1.0×10^{-1}	3.1×10^{-16}	3.0×10^{-13}	3.0×10^{6}	3.8×10^{-10}	2.0×10^{-7}	3.5×10^{13}	6.7×10^{-2}	无相对界
q=20	3.9×10^{-3}	3.9×10^{-3}	3.1×10^{-16}	1.5×10^{-13}	1.5×10^{2}	1.2×10^{-14}	4.8×10^{-11}	1.7×10^{12}	2.3×10^{-5}	无相对界
F55	1.1×10^{-2}	4.3×10^{0}	7.3×10^{-16}	5.0×10^{-13}	5.6×10^{8}	2.3×10^{-8}	8.5×10^{-6}	3.5×10^{13}	6.9×10^{-2}	无相对界
F79	9.1×10^{-3}	8.7×10^{-3}	3.3×10^{-16}	2.2×10^{-13}	3.0×10^{1}	1.4×10^{-15}	1.6×10^{-11}	2.5×10^{11}	3.1×10^{-6}	2.6×10^{-2}
q=14，等式	5.1×10^{-2}	2.8×10^{1}	3.0×10^{-15}	2.9×10^{-12}	2.8×10^{1}	1.1×10^{-15}	6.5×10^{-13}	2.8×10^{1}	1.0×10^{-15}	5.0×10^{-13}
F55，等式	1.0×10^{-2}	4.3×10^{0}	7.3×10^{-16}	5.0×10^{-13}	4.3×10^{0}	2.5×10^{-16}	1.1×10^{-13}	4.3×10^{0}	2.0×10^{-16}	8.6×10^{-14}
q=14, T 原生									9.9×10^{-16}	1.5×10^{-13}
q=14, X_{21} 原生									7.1×10^{-2}	无相对界
F55, T 原生									2.0×10^{-16}	7.5×10^{-14}
F55, X_{21} 原生									6.9×10^{-2}	无相对界

表 IXϵ_X 在 $g = 64$、$n = 8192$、四个 GPU 上以及在 $n = 16384$ (H100) 上。斜率：H100；单位：混合单位（RTX 4090）；ADP：各图形处理器网格上最差有界值与原生值之比。

路径	斜率	H100	A100	RTX 4090	L40S	n=16384	单位
原生	0.00	2.9×10^{-15}	2.9×10^{-15}	2.8×10^{-15}	2.9×10^{-15}	4.8×10^{-15}	3.4×10^{-15}
q=14	0.76	7.1×10^{-2}	7.1×10^{-2}	7.1×10^{-2}	7.1×10^{-2}	7.4×10^{-2}	1.1×10^{-4}
q=16	0.87	6.7×10^{-2}	6.7×10^{-2}	6.7×10^{-2}	6.7×10^{-2}	7.0×10^{-2}	6.1×10^{-7}
q=20	0.96	4.7×10^{-5}	4.7×10^{-5}	4.7×10^{-5}	4.7×10^{-5}	8.3×10^{-5}	1.9×10^{-11}
F55	0.81	6.8×10^{-2}	6.8×10^{-2}	6.8×10^{-2}	6.8×10^{-2}	7.1×10^{-2}	2.3×10^{-5}
F79	0.96	4.1×10^{-6}	4.1×10^{-6}	4.1×10^{-6}	4.1×10^{-6}	5.9×10^{-6}	1.4×10^{-12}
F111	0.00	3.3×10^{-14}	3.3×10^{-14}	3.3×10^{-14}	3.3×10^{-14}	–	2.6×10^{-16}
q=14，等式	0.00	2.0×10^{-15}	2.0×10^{-15}	2.0×10^{-15}	2.0×10^{-15}	3.9×10^{-15}	2.8×10^{-15}
q=16，等式	0.00	2.3×10^{-16}	2.2×10^{-16}	2.1×10^{-16}	2.3×10^{-16}	–	2.5×10^{-16}
F55，等式	0.00	2.9×10^{-16}	2.7×10^{-16}	2.7×10^{-16}	2.9×10^{-16}	3.5×10^{-16}	4.5×10^{-16}
F79，等式	0.00	2.4×10^{-16}	2.3×10^{-16}	2.3×10^{-16}	2.4×10^{-16}	–	2.7×10^{-16}
ADP		1.05×	1.03×	1.02×	1.02×		

表 Xϵ_X 采用原生执行的选定产品（H100，$g = 64$）。前/后 16 项：最大的/最小的尺度超额。

原生执行计算	GEMMul8 $q = 14$ n=4096	GEMMul8 $q = 14$ n=8192	FIXED 55 n=4096	FIXED 55 n=8192
none	7.1×10^{-2}	7.1×10^{-2}	6.9×10^{-2}	6.8×10^{-2}
LAUUM	7.1×10^{-2}	7.1×10^{-2}	6.9×10^{-2}	6.8×10^{-2}
每个 X_{21}	7.1×10^{-2}	7.1×10^{-2}	6.9×10^{-2}	6.8×10^{-2}
底部 16 个	7.0×10^{-2}	7.1×10^{-2}	6.8×10^{-2}	6.8×10^{-2}
顶部 16 个	9.9×10^{-16}	1.6×10^{-2}	2.0×10^{-16}	1.6×10^{-2}
除一个之外的每个 T	$\ge 3.7 \times 10^{-3}$	$\ge 1.9 \times 10^{-3}$	$\ge 3.7 \times 10^{-3}$	$\ge 1.9 \times 10^{-3}$
每个 T	9.9×10^{-16}	2.0×10^{-15}	2.0×10^{-16}	2.9×10^{-16}

H100 上，NsightSystems 跟踪显示，切换点两侧启动的是相同内核，只是执行时长有所变化：对于 4096^3 乘积，INT8 矩阵乘法在 103 位时每次调用耗时 2,350 μs，在 107 位时耗时 47.2 μs。超过切换点后，乘积由仿真库中的 CUTLASS 双精度浮点张量核心矩阵乘法计算，而非 cuBLAS 的默认双精度浮点内核；这解释了为何超出位数上限的输出虽然准确，却与原生 DGEMM 并非逐位一致。4096^3 乘积在 103 位时耗时为原生性能的 1.69 倍，在 107 位时为 2.17 倍，这将 H100 的切换点置于 RTX 4090 上的情况。两个操作数均为精确零值时（如单精度 GEMM LAUUM），路径也会发生变化：8192^3乘积在 RTX 4090 上以 1.16 倍原生时间返回原生的输出。

c) 在 *cuSOLVER* 内部并采用 *DEFAULT* 时：cuSOLVER 12.3.5 从 cuBLAS 导入 11 个双精度浮点三级入口点，而一个 LD_PRELOAD 拦截器会包装所有这些入口点，并记录每次内部调用的操作、形状、策略和估算值。包装全部 11 个入口点后，potri 在 $n = 4096$ 和 $g = 64$ 下的每种模式中都会进行 108 次内部双精度浮点三级调用；不使用拦截器时，cuBLAS 对同一调用的 API 日志在原生、DEFAULT 和 EAGER 模式下分别记录 108、108 和 108 次，与拦截器逐项计数的结果一致。在 $n = 8192$,$g = 64$下，两块 GPU 上的内部 TRSM 调用报告的位数均为 66 至 129 位，TRMM 则最高为 125 位；63 次 SYRK 调用均未采用仿真，这与计算能力 10.0 以下的文档说明一致。DEFAULT 是调用方不主动要求 EAGER 时采用的策略，它会自行做出决定。在 RTX 4090 上，分块路径发出的 1090 次 GEMM 中，有 500 次写入了估算值：其中 421 次不超过 103 位，在所有测试的输出尺寸下均采用仿真；59 次达到或超过 110 位，在所有尺寸下均采用原生计算；另有 20 次介于两者之间，未分类；具体路径是根据估算值相对于切换点推断得出的。

表 XI 实对称正定矩阵：平衡框架下每组最大误差（L40S；最后一列为 A100）。跨度：对角线跨度的 $\log_2$。ADP：cuSOLVER EAGER。

矩阵	span	原生	ADP	q=14	F55	平衡的 q=14	q=16	F55	q=14, A100
人口普查 (4)	32	2.9×10^{-15}	1.7×10^{-15}	1.5×10^{-11}	3.7×10^{-12}	8.4×10^{-15}	1.2×10^{-15}	2.3×10^{-15}	8.3×10^{-15}
空气 (4)	20	1.4×10^{-15}	9.9×10^{-16}	1.6×10^{-13}	6.3×10^{-14}	3.0×10^{-15}	5.9×10^{-16}	7.9×10^{-16}	3.0×10^{-15}
SuiteSparse (16)	23–44	3.6×10^{-14}	1.2×10^{-14}	1.7×10^{-12}	3.2×10^{-13}	2.7×10^{-13}	2.9×10^{-14}	3.4×10^{-14}	2.6×10^{-13}
bcsstk23	35	2.5×10^{-15}	3.1×10^{-15}	1.7×10^{-12}	5.1×10^{-14}	2.7×10^{-13}	2.5×10^{-15}	5.4×10^{-15}	2.6×10^{-13}

其余尺寸下均采用原生计算；另有 20 次介于两者之间，未分类；具体路径是根据估算值相对于切换点推断得出的。对一次完整的 RTX 4090 DEFAULT 执行进行逐内核 CUPTI 跟踪后发现，63 次调用中有 63 次采用了推断出的路径。在 H100 上，DEFAULT 对任何分块 GEMM（1090 次中 0 次）或任何 cuSOLVER 内部调用都不写入估算值；对 potri 的内核跟踪显示，830 次调用中有 0 次使用仿真内核。在 $n = 8192$,$g = 64$下，H100 和 A100 上 220 次内部 potri 调用中，DEFAULT 均未采用仿真；而 L40S 和 RTX 4090 上有 53 次采用仿真，EAGER 模式下四块 GPU 均有 155 次采用仿真。因此，调用方通过 cuSOLVER 的策略设置启用仿真后，在不同 GPU 上会得到不同的数值路径。

d) cuBLAS 13.3 上的位计数指针：在 H100 的第二个版本中，将位计数输出定向到页锁定主机内存（如测试框架所做的那样），会使单精度 EAGER 乘积的耗时达到原生时间的 16 至 148 倍。受控实验表明，增加的耗时来自仿真库的收尾内核；该内核以相同配置启动，使用指针时运行时间延长了 620 倍；cuSOLVER 在其内部调用中并未启用该指针。因此，对不透明库进行插桩可能会扰动测量结果（补充材料）。

附录 F 成本

下面的时间描述的是记录到的会话；它们并不构成对其他设备、驱动程序、时钟状态或会话所得结果的上界，结论依据的是各路径的排序以及远离 1 的比值。表 XII 列出了所有路径。在 H100 上的全部 7 组匹配比较（2 个节点上的 3 次会话）以及 RTX 4090 上的全部 7 组比较，均可由在计算前预先固定规则得到的 97.9% 次序统计区间判定；最接近 1 的是 H100 上均衡化的成本，会话中位数为 1.036 至 1.043。分块 LAUUM 在每个 GPU 上都能节省时间（分块 / 单 GEMM，原生：H100 上为 0.70，RTX 4090 上为 0.57），而在采用分块 LAUUM 时，均衡化 FIXED 55 所需时间是 H100 上均衡化原生路径的 1.13 倍；采用单 GEMM LAUUM 时为 0.92 倍（RTX 4090 上分别为 0.33 和 0.27）。

在 RTX 4090 上，最快的准确路径是使用 16 个模数的均衡化 GEMMul8 和分块 LAUUM，该路径在多次会话中对一个 8192×8192 矩阵求逆耗时 237 至 244 毫秒，而 cuSOLVER 需要 595 至 615 毫秒。在 H100 上，最快的准确路径是原生路径，即采用分块 LAUUM 的分块求逆；最快的准确仿真路径是采用分块 LAUUM 的均衡化 FIXED 55，其耗时为原生路径的 1.20 倍。在计时范围内包含从可分页内存进行上传和下载后，差距缩小：RTX 4090 上最快的准确路径耗时为 cuSOLVER 的 0.52 倍，而在 H100 上，数据复制占据主导地位（原生 cuSOLVER 为 236 至 273 毫秒）。仿真还需要工作空间：在 H100 上，采样得到的设备内存峰值在使用 14 个模数时达到 3,652 MiB，使用 FIXED 55 时达到 5,474 MiB，而原生路径仅为 2 MiB。这些 T 乘积的 GEMM 工作量是 TRTRI 的一半，因此 T-原生混合路径在 FP64 较慢的情况下收回了部分仿真收益：在 RTX 4090 上，其耗时为均衡化路径的 1.54 倍，而基于求解的路径为 1.07 倍。在已有因子的 cuSOLVER potri 中，扩散使 RTX 4090 上的 DEFAULT 从原生时间的 0.737（ $g = 0$ 时）升至 0.918（ $g = 64$ 时），而均衡化可恢复这一性能（均衡化输入上的 EAGER 为 1.050）。

附录 G 应用

a) 设计：在 B_{cov} 五个变量（年龄中位数、房间总数、人口、收入中位数和房屋价值中位数）的协方差以及 C 位置上的平方指数核条件下，模型的协方差为 $B_{\text{cov}} \otimes (C + \tau I)$。如果每个训练地点都观测到房屋价值，那么模型的次级权重恰好为零（制品中包含此类运行，作为阴性对照）。因此，该设计是异位的：四个次级变量在所有 m 训练地点都有观测值，而房屋价值只在预先确定的一半地点有观测值，所以协方差在原始单位下的阶数为 $n = 4.5m$, $n \in \{4608, 9216\}$。B_{cov} 和均值取自除 256 个留出街区组以外的所有街区组，因此不同设计采用固定的协方差模型。长度尺度为 0.1 和 0.3 度（约 11 和 33 千米）；148 个训练地点与另一个地点坐标相同，而块金效应使 K 保持正定。在模型的块金效应下，次级观测对预测的贡献为 14,629 至 33,421 美元均方根，协克里金使留出均方根误差变化 -2.3% 至 5.1%。二次幂均衡将$\kappa_2(K)$从 2.2×10^{12}–1.7×10^{13} 降至最多 1.4×10^{5}，而 TRTRI 乘积的第 99 百分位尺度超额达到 127 位（$n = 4608$）。在 $n = 4608$ （表 VI）中，参考预测器在三个块金效应下的留出均方根误差分别为 85,729、101,684 和 525,730 美元。在 $\tau = 10^{-6}$ 下，binary64 特征值求解器对 8 个矩阵中的 5 个返回了负的最小特征值估计；这些矩阵在精确算术下均为正定。该块金效应仅属于数值压力测试情形，相关结果均未获认证。

b) 结果：所有 ADP 路径相对于参考值都保持在原生水平（EAGER 的留出均方根误差最多为 1.9×10^{-7}，原生路径为 7.2×10^{-8}）。在两种规模下，表现最差的固定路径达到留出均方根误差的 1.2×10^{-8}

表 XII 完整路径、设备驻留、 $n = 8192$、 $g = 64$：中位时间（毫秒） [四分位距], 相对于原生 cuSOLVER 的比值以及 ϵ_X。所有列、行和会话：补充材料。

	RTX 4090			H100		
路径	毫秒 [四分位距]	比值	ϵ_X	毫秒 [四分位距]	比率	ϵ_X
cuSOLVER，原生	595 [0.74]	1.00	8.6×10^{-16}	58.4 [0.27]	1.00	8.7×10^{-16}
cuSOLVER，DEFAULT	548 [0.8]	0.92	8.7×10^{-16}	58.4 [0.34]	1.00	8.7×10^{-16}
cuSOLVER，EAGER	912 [1.1]	1.53	8.7×10^{-16}	186 [0.7]	3.18	8.7×10^{-16}
分块，原生，分块 LAUUM	775 [0.2]	1.30	8.8×10^{-16}	24.3 [0.14]	0.42	8.9×10^{-16}
q=14	241 [0.37]	0.40	7.1×10^{-2}	37.5 [0.075]	0.64	7.1×10^{-2}
q=14, T 原生	377 [0.51]	0.63	2.1×10^{-15}	34.9 [0.14]	0.60	2.1×10^{-15}
q=14，等式	245 [0.38]	0.41	2.1×10^{-15}	38.9 [0.099]	0.67	2.1×10^{-15}
q=14，基于求解的	263 [0.17]	0.44	2.1×10^{-15}	46.6 [0.17]	0.80	2.1×10^{-15}
q=16，等式	252 [0.43]	0.42	5.4×10^{-16}	41.5 [0.099]	0.71	5.5×10^{-16}
q=16，等式，分块 LAUUM	237 [0.11]	0.40	8.4×10^{-16}	37.5 [0.11]	0.64	8.5×10^{-16}
F55	360 [1.2]	0.61	6.8×10^{-2}	31.9 [0.18]	0.55	6.8×10^{-2}
F55, T 原生	497 [0.62]	0.84	5.6×10^{-16}	30.7 [0.22]	0.53	5.7×10^{-16}
F55，等式，分块 LAUUM	261 [0.15]	0.44	8.4×10^{-16}	29.2 [0.23]	0.50	8.5×10^{-16}
F55，基于求解的	381 [1.1]	0.64	5.6×10^{-16}	39 [0.32]	0.67	5.7×10^{-16}

模型块金效应的水平，在 $\tau = 10^{-3}$ 时为 6.0×10^{-7}，在压力测试情形下为 7.4×10^{-5}；均衡化使每条固定路径的误差都降至 9.0×10^{-8} 或以下。仅使用观测到的房屋价值（8 个案例）重新校准 B_{cov} 和均值后，在 $\tau = 10^{-1}$ 处的留出均方根误差从 85,729 美元变为 85,725 美元，这些发现依然成立：表现最差的原始固定路径从 1.5×10^{-9} 变为 1.3×10^{-9}，而均衡化路径保持在 3.0×10^{-11} 或以下。

c) 参考证书：参考预测值通过对 K 进行 binary64 Cholesky分解并迭代细化来计算，残差和迭代结果采用 x86 long double；对于全部 12 个系统（$n = 4608$），按分量后向误差在 4 个步骤内降至不超过 4.8×10^{-19} 。精确以下表示对所存储的 binary64 数组 K、 b、 k_j进行精确算术运算：$x = K^{-1}b$和 $s_j = k_j^T x$。令 $\hat{s}_j$为计算得到的预测值，$r = b - K\hat{x}$为所计算解的精确残差， S为二的幂均衡，$0 < \ell \le \lambda_{\min}(SKS)$。由于 $\hat{x} - x = -S(SKS)^{-1}Sr$，

$$|\hat{s}_j - s_j| \le \|Sk_j\|_2 \|Sr\|_2 / \ell + \eta_j, \tag{4}$$

其中， $\eta_j = \gamma_n |k_j|^T |\hat{x}| + u|\hat{s}_j|$ 包含长双精度点积及其转换的舍入误差。由于 $K_* = P[(C_* + \tau I) \otimes B_{\mathrm{cov}}]P^T$精确模型矩阵， $S_v B_{\mathrm{cov}} S_v \succeq \mu I$ 由精确有理数 LDL^T证明，且 $\Delta \ge \|S(K - K_*)S\|_F$ 界定了存储值与模型值之间的舍入误差，由Weyl不等式可得

$$\lambda_{\min}(SKS) \ge \tau\mu - \Delta =: \ell. \tag{5}$$

令 B_{cert} 为右端 (4) 在留出点上的均方根上界， R 为参考值留出均方根误差的下界， d 为某一路径的预测值与参考值之间均方根差异的上界，

$$\frac{\mathrm{RMS}(\hat{s}_{\mathrm{route}} - s)}{\mathrm{RMSE}_{\mathrm{exact}}} \le \frac{d + B_{\mathrm{cert}}}{R - B_{\mathrm{cert}}}, \quad R > B_{\mathrm{cert}}. \tag{6}$$

此处为 $\mu \ge 0.142$、 $\Delta \le 2.7 \times 10^{-13}$ 和 $\ell/\tau \ge 0.142$，涵盖全部 24 种情况。 $B_{\mathrm{cert}}/(R - B_{\mathrm{cert}})$ 在 $\tau = 10^{-1}$ 时至多为 7.8×10^{-7} 并且在 $\tau = 10^{-3}$ 时为 7.4×10^{-3} ；在 $\tau = 10^{-6}$ 时为 $B_{\mathrm{cert}} \ge R$ ，涵盖全部 8 种情况。认证上界由 B_{cert} 设定：测得的偏差 d 比它低两个数量级，因此偏差之间的比值仅比较计算得到的解。

d) 范围：该证书假设使用 IEEE binary64 和 x87 长双精度算术，且在其自身求值过程中不会发生溢出、下溢或无效操作；脚本会逐个检查每个案例。它不信任任何库指数函数：其自身的长双精度例程会在每次运行时检查所有可表示输入的自变量约化（见补充材料），误差小于 $167u_{ld}$。路径测量结果和证书使用同一个参考向量，通过其 SHA-256 摘要确认（16 个案例中的 16 个）。一个与证书代码完全独立的验证器，假设平台指数函数的误差不超过一个最后一位单位，通过全部 24 个案例，并给出$B_{\mathrm{cert}}/(R - B_{\mathrm{cert}})$，其最大值为 3.2×10^{-7}，出现在 $\tau = 10^{-1}$。包含的内容：ℓ、逐点界、 B_{cert}、 R ，以及在 $\tau = 10^{-1}$ 和 10^{-3} 处的比值 (6)。不包含的内容：预测方差、路径之间的比值，以及协方差模型在统计上的适当性。

e) 逆误差的投影：为什么逆误差本身很大，却对预测值影响很小？令 $E = \widehat{X} - K^{-1}$ 为相对于精确逆矩阵的误差。每个输出只通过投影感受到它：均值通过$k_j^T Eb$，方差 $v_j = c_0 - k_j^T K^{-1} k_j$ 通过 $-k_j^T E k_j$。相对于方差，该误差的上界为 $\epsilon_X \, \alpha_j \, \kappa_j$，其中 $\epsilon_X = \|E\|_2 / \|K^{-1}\|_2$、 $\alpha_j = \|K^{-1}\|_2 \|k_j\|_2^2 / (k_j^T K^{-1} k_j)$是上界所允许的对齐度，而 $\kappa_j = k_j^T K^{-1} k_j / v_j$ 是 v_j中的消去。测量值无法直接利用 E 本身。它们使用 $E_{\mathrm{ref}} = X - X_{\mathrm{LAPACK}}$，即与 LAPACK 计算得到的逆矩阵之差；范数采用 binary64 特征值估计，而方差差异则是相对于精化求解所得的方差计算的，这不在该证书的覆盖范围内。这些结果描述了计算结果之间的差异；对于精确逆矩阵相关的量，它们只能在参考值自身误差的范围内给出上界，而这些误差在此处未获认证。在 H100 上测得（$n = 4608$，位置优先，0.3 度），GEMMul8

采用 14 个模数后，将 ϵ_X （ E_{ref}）从 4.0×10^{-15} （原生、分块）提升至 $\tau=10^{-1}$时的 5.7×10^{-11}，但最坏情况的实际比例 $|k_j^T E_{\mathrm{ref}} k_j|/(\|E_{\mathrm{ref}}\|_2\|k_j\|_2^2)$ 则从1.3×10^{-11} 降至 4.0×10^{-14} （中位数）。相对于测试交叉协方差向量上的谱范数界，测得的二次投影较小；由于 E_{ref} 可能是不定的，该实验无法区分弱对齐与这些投影内部的消去。最大的相对方差差异仅从 4.1×10^{-12}变为 2.1×10^{-10}，而在 $\tau=10^{-3}$ 时，它不大于原生路径的差异（1.6×10^{-8} 对比 8.6×10^{-8}）。在三个块金效应参数下，抵消因子 κ_j分别达到 9.8、 9.7×10^{2}和 9.5×10^{5}，且每条路径的方差差异都随之增大；在 $\tau=10^{-6}$ 时，包括原生路径在内的每条路径都显示出较大的方差差异（cuSOLVER 为6.0×10^{-2}）。