arXiv:2304.06200v1 [quant-ph] 13 Apr 2023

最优控制大型量子系统：评估 GRAPE 的内存和运行性能

Yunwei Lu, [1], * Sandeep Joshi, [1] Vinh San Dinh, [1] 和JensKoch[II]美国
伊利诺伊州埃文斯顿市60208号西北大学物理与天文学系（日期：2023年4月14日）

梯度上升脉冲工程（GRAPE）是量子最优控制中一种广泛使用的技术，并且可以与自动微分（自动微分）相结合，以便即时计算代价函数的梯度。我们说明，自动微分的便利性会带来显著的内存代价，这是由于需要累积存储大量状态和传播算符。对于希尔伯特空间规模不断增大的量子系统，这会造成严重的瓶颈。我们重新审视了将梯度硬编码的策略，该策略完全避免了传播算符存储，并显著降低了内存需求。此外，我们还对数值状态传播进行了改进，以提升运行性能。我们对运行时间和内存使用量进行了基准测试，并将该方法与基于自动微分的实现进行了比较，重点是推动其应用于更大的希尔伯特空间规模。结果证实，这种无自动微分方法有助于将最优控制应用于大型量子系统，而这类系统若采用其他方法则难以处理。

I. 引言

快速发展的量子信息处理领域激发了人们对量子最优控制的浓厚兴趣，其应用任务包括状态制备 [1], 误差校正 [2, 3]以及逻辑门的实现 [4–7]。这一方法已应用于各种量子系统 [8], , 例如核磁共振 [9–14], 、囚禁离子 [15, 16],、金刚石中的氮空位中心 [17–23], 、玻色–爱因斯坦凝聚 [24–27]以及中性原子 [28, 29]。

实现量子最优控制的一项重要技术是梯度上升脉冲工程（GRAPE）。最优控制的核心目标是调整控制参数, 从而使所需态转移或门操作的保真度损失最小化。在GRAPE中，最小化基于梯度下降，因此通常需要计算梯度。自动微分（自动微分） [30] 是一种便捷的工具，也已对使用GRAPE [31–33]的量子最优控制产生了影响。在内部，自动微分构建计算图，并应用链式法则来自动计算给定函数的梯度。

然而，自动微分的便利性是以存储完整计算图所需的内存为代价的。使用半自动微分（半自动微分） [34]与全自动微分（全自动微分）相比，可部分降低内存开销，但对于规模迅速增长的量子系统的最优控制问题，这种降低仍可能不够 [35, 36]。因此，我们重新审视了最初的GRAPE 方案 [11]，该方案基于硬编码梯度（混合梯度），内存开销尽可能小。我们对内存使用量和运行时间进行了标度分析，并据此构建了一棵决策树，以指导在混合梯度、半自动微分和全自动微分之间选择最优策略。我们通过具体优化任务的基准测试，展示了其标度行为。

本文结构如下。第II 节简要回顾 GRAPE 并介绍必要的符号。在第III节中，我们展示最常用成本函数贡献项的梯度推导及其内存高效实现，并介绍 HG 的数值实现如何进一步提高状态传播的效率。在第IV节中，我们分析并比较 HG、AD 和半自动微分的运行时间与内存使用量随规模的变化，并给出具体基准测试研究的结果。在第V节中，我们制定一套决策树，以帮助选择适用于基于 GRAPE 的量子最优控制的最优数值策略。我们在第VI节总结研究发现并给出结论。

II. GRAPE 简述

我们简要概述 GRAPE 的基础知识 [11], , 主要是为了确立后续章节将使用的概念。

考虑一个由哈密顿量描述的系统 ian

$$H(t) = H_s + a(t)\, h_c. \tag{1}$$

这里， H_s 表示静态系统哈密顿量， h_c 是将经典控制场 $a(t)$耦合到系统[1]的算符。量子最优控制旨在调节 $a(t)$, 使系统在给定时间区间 $0 \le t \le T$内实现所需的幺正门操作或态转移。为便于优化，总控制时间 T 被划分为 N 个时长为$\Delta t = T/N$的区间，并通过离散时刻 $t_n = n\Delta t$ $(n = 1, 2, \ldots\ldots, N)$处的幅值来指定控制场。离散间隔 Δt 的选择应确保控制幅值在每个 Δt内近似保持恒定[2]。我们将时间离散值记为

$$a_n := a(t_n), \tag{2}$$

并将这些组合起来，形成向量 $\mathbf{a} \in \mathrm{R}^N$。在这个分段常值控制场的作用下，系统的

* yunweilu2020@u.northwestern.edu

[1]为简单起见，我们考虑一个具有实值 $a(t)$的控制通道。推广到多个控制通道以及复数 a 的情形也很直接。[2]这种处理方式实际上与任意波形发生器实际输出的建模方式十分接近，其中 Δt 可选择为与可用分辨率相匹配。

量子态 $|\psi_n\rangle := |\psi(t_n)\rangle$ 根据单个时间步演化 naapertorlugu

$$|\psi_n\rangle = U_n|\psi_{n-1}\rangle. \tag{3}$$

这里， U_n 是短时传播子

$$U_n := U(t_n, t_{n-1}) = e^{-iH_n\Delta t} \qquad (\hbar = 1)\ , \tag{4}$$

其中 $H_n = -i\,(H_s + a_n\, h_c)\Delta t$。控制场的优化通过最小化代价函数 C 来实现。通常， C 包含态转移或门保真度损失，以及用于调节脉冲特性（如最大功率或带宽）的其他代价项。为此，构造一个复合代价函数 $C\mathbf{(a)} = \sum_\nu \alpha_\nu C_\nu\mathbf{(a)}$，其中 α_ν 是经验选取的权重因子， C_ν 是各个代价项。GRAPE 使用最速下降法，通过根据代价函数梯度更新控制幅度，来最小化 $C\mathbf{(a)}$:

$$\mathbf{a}^{(j+1)} = \mathbf{a}^{(j)} - \eta_j\, \nabla_{\mathbf{a}} C(\mathbf{a}^{(j)}). \tag{5}$$

这里， j 表示最速下降迭代次数， $\eta_j > 0$ 是控制步长的学习率 g。

III. 梯度的计算

GRAPE 的一个关键环节是计算代价函数梯度。一种常用方法是利用自动微分来完成这一任务，例如 `Tensorflow` [37] 和 `Autograd` [38] 中实现的方法。这种策略的明显优势在于，复杂的代价函数梯度可以在运行时轻松地自动求得。然而，这种便利是以大量内存消耗为代价的，对于大型量子系统而言，这可能难以承受。当最优控制应用于大型量子系统时，这种内存资源的浪费尤为不可接受。因此，我们着手重新研究如何以计算高效的方案对关键代价项使用硬编码梯度。

A. 解析梯度

本小节将讨论两个关键代价项的解析梯度。其他代价项的梯度表达式见附录A。

在态转移问题中，量子最优控制的一个核心目标是最小化态转移保真度损失，其定义为。

$$C_1^{st} = 1 - |\langle\phi_T|\psi_N\rangle|^2\,, \tag{6}$$

其中， $|\phi_T\rangle$ 是目标态， $|\psi_N\rangle$ 是在最终时刻 t_N实现的状态。C_1^{st}的梯度形式为

$$\begin{aligned}\frac{\partial C_1^{st}}{\partial a_n} &= -\langle\phi_T|U_N\cdots\frac{\partial U_n}{\partial a_n}\cdots U_1|\psi_0\rangle\langle\psi_N|\phi_T\rangle - \text{c. c.}\\ &= -\langle\phi_n|\frac{\partial U_n}{\partial a_n}|\psi_{n-1}\rangle\langle\psi_N|\phi_T\rangle - \text{c. c.},\end{aligned} \tag{7}$$

其中，状态 $|\phi_n\rangle$ 满足递归关系

$$|\phi_n\rangle = U_{n+1}^\dagger|\phi_{n+1}\rangle. \tag{8}$$

这可以解释为反向传播。

除了态转移之外，幺正门是另一类值得关注的操作。通过最小化门保真度损失 $C_1^g = 1 - \left|\mathrm{tr}\,(U_T^\dagger U_R)/d\right|^2$可以实现量子门优化。其中，$U_T$是目标幺正门，而 $U_R = U_N\cdots U_1$ 是实际实现的门。对于给定的正交归一基 $\{|\psi_0^h\rangle\}_h$ （h枚举基态）， C_1^g 的梯度可以写成

$$\frac{\partial C_1^g}{\partial a_n} = -\frac{1}{d^2}\sum_h\langle\phi_n^h|\frac{\partial U_n}{\partial a_n}|\psi_{n-1}^h\rangle\sum_{h'}\langle\psi_N^{h'}|\phi_N^{h'}\rangle - \text{c. c.}, \tag{9}$$

其中 $|\psi_n^h\rangle = U_n\cdots U_1|\psi_0^h\rangle$。状态 $|\phi_n^h\rangle$ 的获得方式如下：将目标幺正算符 U_T作用于基态 $\{|\psi_0^h\rangle\}_h$，然后将结果反向传播至时间 t_n,

$$|\phi_n^h\rangle = U_{n+1}^\dagger\cdots U_N^\dagger U_T|\psi_0^h\rangle. \tag{10}$$

公式（7）和（9）给出了梯度下降所需的解析表达式。在接下来的两个小节中，我们将讨论如何以节省内存的方式对这些表达式进行数值求值。

B。硬编码梯度的数值实现 度

计算公式（7）和（9）中的梯度，需要对以下形式的表达式进行数值求值： $U_n|\Psi\rangle$ 和 $\frac{\partial U_n}{\partial a_n}|\Psi\rangle$。这里，短时传播子 U_n 由矩阵指数给出 e^A 和 $A = -iH_n\Delta t$。

对 $e^A|\Psi\rangle$ 进行数值求值并非易事。正如 Moler 和 Van Loan 所指出的 [39], , 存在多种方法，但没有一种真正最优。参考文献 [39] 中评价较高的三种方法分别基于常微分方程求解、矩阵对角化以及缩放与平方（结合级数展开）。在此情境下，求解常微分方程通常运行时间开销最大 [34]。我们主要关注缩放与平方，但也会在第V节讨论更适合采用矩阵对角化的情况。

缩放与平方 [40] 其基于恒等式

$$e^A = \left(e^{A/s}\right)^s, \tag{11}$$

其中，右侧的矩阵指数现在涉及矩阵 $B = A/s$ ，其范数比 $||A||$ 小。这通常使得指数级数[3] ,的截断阶数 m 可以更低。

$$e^B \approx e_m^B := \sum_{q=0}^{m}\frac{B^q}{q!}. \tag{12}$$

[3] 虽然切比雪夫展开的收敛速度比泰勒展开更快，但两者的数值实现具有相同的运行时间标度。这里，我们因泰勒展开较为简单而重点讨论它。

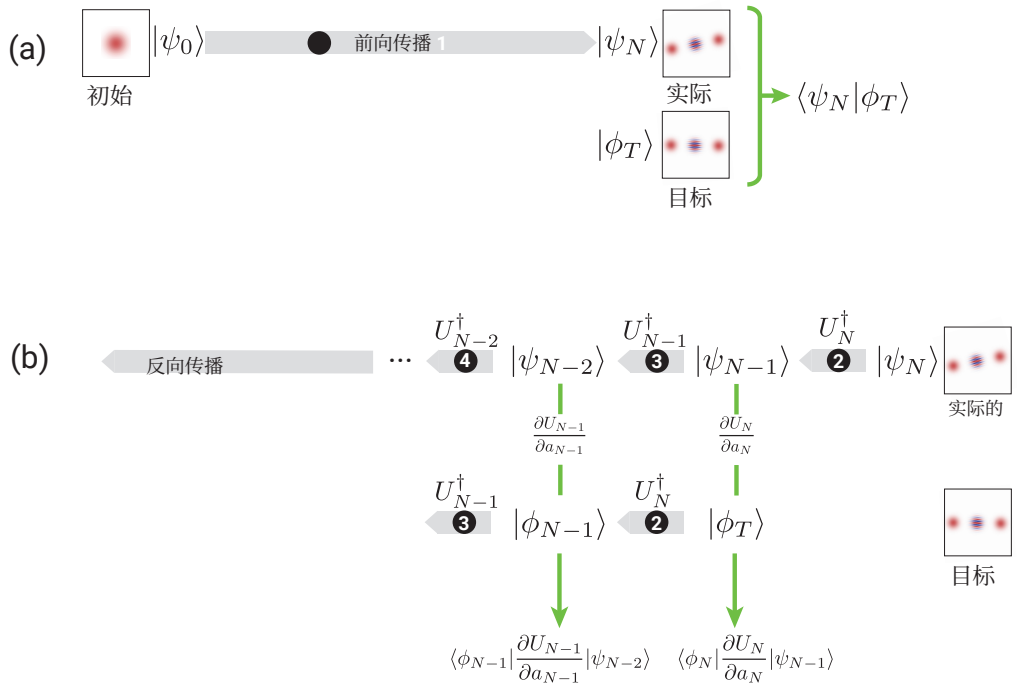

图 1. 用于计算式（7）中梯度分量的前向-反向传播方案。(a) 将初始态通过 $U_N \cdots U_1$ 前向传播，得到实际态 $|\psi_N\rangle$ 并计算重叠 $\langle\psi_N|\phi_T\rangle$，其中 $|\phi_T\rangle$ 是目标态。(b) 使用传播子 $U_N^\dagger$、$U_{N-1}^\dagger$、$\cdots$、$U_1^\dagger$ 迭代地反向传播 $|\psi_N\rangle$ 和 $|\phi_T\rangle$，并在每一步计算 $\langle\phi_n|\frac{\partial U_n}{\partial a_n}|\psi_{n-1}\rangle$。

通常， m 比原始 e^A 所需的值更小。我们的目标是近似态矢量 $e^A|\Psi\rangle \approx (e_m^B)^s|\Psi\rangle$，其中

$$e_m^B|\Psi\rangle = \left(1 + B + \frac{B^2}{2!} + \cdots + \frac{B^m}{m!}\right)|\Psi\rangle \tag{13}$$
$$= |\Psi\rangle + B|\Psi\rangle + \frac{B\left(B|\Psi\rangle\right)}{2} + \cdots + \frac{B\left(\cdots\left(B|\Psi\rangle\right)\right)}{m!}.$$

根据最后一个表达式， $e^A|\Psi\rangle$ 的近似因此可以通过对一个向量重复应用 $B = A/s$ 来实现，而无需进行矩阵乘法。这将运行性能从 $O(d^3)$提升至 $O(d^2)$。

为继续评估 $e^A|\Psi\rangle$，截断阶数 m 和缩放因子 s 的确定方式应确保误差低于误差容限 τ，

$$\varepsilon_{A,|\Psi\rangle}(m,s) := \frac{\left|\left|e^A|\Psi\rangle - [\![(e_m^B)^s|\Psi\rangle]\!]\right|\right|}{||e^A|\Psi\rangle||} < \tau. \tag{14}$$

此处， $[\![\bullet]\!]$ 表示 $\bullet$的浮点表示。由于难以且代价高昂地计算此误差 $\varepsilon_{A,\ |\Psi\rangle}$ 我们转而推导其上界 $E_A(m,\ s)$ [，参见附录 C]，并采用不等式

$$\varepsilon_{A,|\Psi\rangle}(m,s) < E_A(m,s) \leq \tau. \tag{15}$$

对于给定的 A，该不等式通常对于 m 和 s有多个解。如附录 C所示，评估需要 ms次矩阵-向量乘法，据观察这些乘法占据主导地位。运行时间。由于系统性地最小化 ms 本身需要大量运行时间，因此我们改为：(1) 选择满足方程 (15) 且 s 最小的解，并将其记为 $\mathbb{s}$；(2) 在这些解对中，选择 m 最小的一个（记为 $\mathbb{m}$）。因此，计算 $e^A|\Psi\rangle$ 需要

$$\mu := \mathbb{m}\mathbb{s} \tag{16}$$

矩阵-向量乘法。正如附录 D中的两个具体示例所示，我们用于评估 $e^A|\Psi\rangle$的方法在运行性能和精度方面均优于SciPy的expm 乘法 函数 [41]。

解决了数值评估 $e^A|\Psi\rangle$相关的挑战后，我们现在转而计算 $\frac{\partial U_n}{\partial a_n}|\Psi\rangle$，这对于计算方程（7）和（9）中的代价函数梯度至关重要。我们基于以下近似进行评估

$$\frac{\partial U_n}{\partial a_n}|\Psi\rangle \approx \frac{\partial (e_m^{B_n})^s}{\partial a_n}|\Psi\rangle, \tag{17}$$

其中 $B_n = -iH_n\Delta t/s$。我们通过使用基于恒等式 [42, 43] 的辅助矩阵方法来继续推导。

$$(e_m^{\mathcal{A}_n/s})^s \begin{pmatrix} 0 \\ |\Psi\rangle \end{pmatrix} = \begin{pmatrix} \frac{\partial (e_m^{B_n})^s}{\partial a_n}|\Psi\rangle \\ (e_m^{B_n})^s|\Psi\rangle \end{pmatrix}. \tag{18}$$

这将$(\partial (e_m^{B_n})^s/\partial a_n)|\Psi\rangle$ 与作用于向量的辅助矩阵的指数联系起来。这里， $\mathcal{A}_n$ 定义为

$$\mathcal{A}_n := \begin{pmatrix} -iH_n\Delta t & -ih_c\Delta t \\ 0 & -iH_n\Delta t \end{pmatrix}, \tag{19}$$

表 I. 关于希尔伯特空间维数 d、时间步数 N、存储的哈密顿量矩阵元素 $\kappa = \kappa(d)$，以及计算传播子-态乘积所需的矩阵-向量乘法次数 $\mu = \mu(d)$ 的内存使用量和运行时间标度。（μ 和 κ 的标度取决于具体的优化任务。）该表对比了硬编码梯度（HG）、全自动微分（全自动微分）和半自动微分（半自动微分）在态转移和门操作中的标度。

	态转移		门操作	
	运行时间	**内存使用量**	**运行时间**	**内存使用量**
硬编码梯度	$\Theta(N\mu\kappa)$	$\Theta(d+\kappa)$	$\Theta(N\mu\kappa d)$	$\Theta(d+\kappa)$
全自动微分	$\Theta(N\mu\kappa)$	$\Theta(N\mu d+\kappa)$	$\Theta(N\mu\kappa d)$	$\Theta(N\mu d^2)$
半自动微分	$\Theta(N\mu\kappa)$	$\Theta(Nd+\kappa)$	$\Theta(N\mu\kappa d)$	$\Theta(Nd^2)$

它是一个由算符组成的 2×2 矩阵，其中每个算符都作用于维数为 d 的希尔伯特空间。一旦选定一个基，$\mathcal{A}_n$ 就可以表示为一个由复数组成的 $2d\times 2d$ 矩阵。

C. 高效计算硬编码梯度

解析梯度表达式，例如 ∇C_1^{st}式（7），通常十分复杂，涉及大量相关量。根据运算的分组和顺序，这些量可能需要同时存储，也可能需要反复计算，从而对运行时间和内存使用量产生关键影响。Khaneja 等人 [11] 最初提出的方案在两者之间取得了良好平衡，避免了存储的中间状态向量数量激增。具体而言，最终量子态 $|\psi_N\rangle$ 通过向前传播 $|\psi_0\rangle$得到。随后，对 $|\psi_N\rangle$进行 $|\phi_T\rangle$ 反向传播，以计算矩阵元$\langle\phi_n|\frac{\partial U_n}{\partial a_n}|\psi_{n-1}\rangle$，用于求得梯度的第 n 个分量，从而按降序逐步构建梯度。对于两个传播方向，只存储当前的中间状态，因此内存开销为 $\Theta(1)$。[4]

IV. 扩展性分析与基准测试 NG

我们考虑在 GRAPE 中计算梯度的以下三种方法：计算硬编码梯度、全自动微分 [31], 半自动微分[34]。为了确定哪种方法最适合给定的问题，我们考察这三种方法的内存使用量和运行时间如何随问题规模变化。在此分析的基础上，我们展示了针对特定态转移和量子门操作问题的基准测试研究，以说明其规模变化特性。

[4] 这里，$f(x) = \Theta(g(x))$ 是通常的巴赫曼–朗道记号，表示 $f(x)$ 在渐近意义下有 $g(x)$ 作为上下界。

A. 运行时间和内存使用量的扩展性分析

对于希尔伯特空间维数较大（d）的量子系统，或需要大量时间步（N）的时间演化，执行优化任务时可能会面临显著挑战。本小节分析内存使用量和运行时间相对于 N 和 d的规模变化，重点考察态转移优化和幺正门优化中的代价项 C_1。我们的发现也可推广到C_1以外的其他代价项，详见附录A。

a. 使用前向-反向传播计算硬编码梯度（*HG*）。我们首先考虑态转移问题。内存使用量评估∇C_1^{st}所需的内存，主要取决于所需存储的量子态和哈密顿量。在前向-反向传播中，任一时刻存储的状态数量和哈密顿量矩阵实例数均为常数级。每个维度为 d 的态矢量会消耗 $\sim\Theta(d)$内存。哈密顿量的内存使用量取决于所采用的存储方案。在“最坏”情况下，采用稠密矩阵存储时，哈密顿量所需的内存为 $\Theta(d^2)$。对于稀疏矩阵存储，内存使用量通常取决于稀疏度，并由存储的哈密顿量矩阵元素数量决定（此处记为 κ）。例如，对于对角矩阵或三对角矩阵，内存使用量为 $\kappa\sim\Theta(d)$；对于具有最近邻σ_z 耦合的线性量子比特链，内存使用量为 $\Theta(d\log_2 d)$。综合上述扩展关系，可得内存使用量的总体渐近行为为 $\Theta(d+\kappa)$。

我们观察到，该方案的运行时间主要取决于所需的 $\Theta(N)$ 次传播子-态乘积及其导数对应项的计算。数值近似计算单个传播子-态乘积需要 μ 次矩阵-向量乘法 [见式（16）], 其中，单次矩阵-向量乘法的运行时间随 $\Theta(\kappa)$变化。$\mu=\mu(d)$ 对 d 的隐式依赖关系因具体系统而异，且难以通过解析方法求得。数值计算发现，对于与腔耦合的驱动 transmon，有$\mu\sim\Theta(\sqrt{d})$；对于耦合量子比特的线性链，标度为 $\Theta(\log_2 d)$，更多详情见附录 E 。将上述各个标度关系结合起来后，可得总体

运行时间的渐近行为为 $\Theta(N\mu\kappa)$。例如，对于耦合量子比特的线性链，运行时间按$\Theta(Nd(\log_2 d)^2)$缩放。

接下来我们讨论门优化问题，其内存使用量和运行时间的标度行为如下。我们通过依次对每个 d 基态执行正向—反向传播来计算 ∇C_1^g。内存使用量为$\Theta(d+\kappa)$，与 ∇C_1^{st}相同。运行时间标度为$\Theta(N\mu\kappa d)$，比 ∇C_1^{st}的运行时间标度大 d 倍。 [5]

b. 全自动微分（全自动微分）反向模式自动微分通过沿计算树进行一次正向遍历和一次反向遍历来提取梯度。在正向遍历中计算代价函数 C，反向遍历则通过链式法则累积梯度。

计算∇C_1^{st}所需的内存使用量取决于一个要求：所有中间数值结果都必须在正向遍历过程中存储，并作为反向遍历的输入。具体而言，正向遍历经过 N 个时间步，将初始态矢量演化为最终态矢量。过程中需要存储的向量总数为$\Theta(N\mu)$，因为通过缩放与平方计算单个短时传播子与态矢量的乘积需要进行 μ 次迭代。此外，考虑哈密顿量所占的内存使用量后，整体标度为 $\Theta(N\mu d+\kappa)$。

评估运行时间 ∇C_1^{st}需要结合正向和反向遍历。其主要耗时来自计算 $\Theta(N)$ 个传播子-态乘积，每个乘积的运行时间标度为 $\Theta(\mu\kappa)$，因此总体运行时间标度为 $\Theta(N\mu\kappa)$。

对于门优化问题， C_1^g 需要演化的不仅是单个初始态，还包括一整组 d 基向量。因此，与态转移相比，内存使用量增加了 d 倍，总体内存标度为 $\Theta(N\mu d^2)$。假设各个基向量的演化不采用并行处理，运行时间也会增加 d 倍，因此 ∇C_1^g 的运行时间标度为 $\Theta(N\mu\kappa d)$。

c. 半自动微分（半自动微分）对于难以通过解析方法推导的代价项梯度，半自动微分 [34] 提供了一种替代方法：将部分导数硬编码，其余部分则通过自动微分计算。与全自动微分相比，半自动微分的优势在于，内存使用量标度不依赖于 μ，同时保持相同的运行时间标度。

d. 比较 该标度分析的结果总结于表 I 中。对于全自动微分，内存使用量相对于 μ 和时间步数 N呈线性增长。相比之下，混合梯度的内存使用量不会随 μ 和N增加，而是保持不变。这一优势并非以更差的运行时间标度为代价。因此，在大型量子系统的最优控制中，当内存使用量本会大到难以承受时，混合梯度优于全自动微分。

当内存瓶颈并不太严重时，半自动微分提供了一种有趣且可行的折中方案。

我们将在下一节通过针对具体优化问题的基准测试研究，展示上述扩展规律。

B. 基准测试研究

我们的第一个基准测试示例研究以下态转移问题。设想这样一种系统：一个受驱动的可调谐磁通 transmon 通过电容与超导腔耦合，并将腔从真空态转移到20光子福克态。与 transmon 驱动共同旋转的参考系中的哈密顿量为

$$H_1(t) = \Delta b^\dagger b + \frac{1}{2}\alpha b^\dagger b(b^\dagger b-1) + g(cb^\dagger + c^\dagger b) + a_z(t)b^\dagger b + a_x(t)(b+b^\dagger). \tag{20}$$

这里，驱动同时具有纵向和横向分量， c 和 b分别是腔光子和跨蒙特激发的湮灭算符。 α、 g 和 Δ分别表示非谐性、相互作用强度，以及 transmon 与腔频率之差。

优化通过最小化保真度损失 C_1^{st}来进行，并通过加入成本贡献来限制较高跨蒙特能级的占据 C_2^{st} [式（A1）]。后者还有一项额外好处，即可以将跨蒙特能级截断到少数几个。我们测量全自动微分[6] 和混合梯度在单次优化迭代中的运行时间与峰值内存使用量。运行时间由软件包`Temci` [44] 测量，峰值内存使用量则由内存分析器软件包 [45]监测。优化在一台配备 AMD Ryzen Threadripper 3970X 32 核中央处理器（频率为 3.7GHz）的计算机上进行。

态转移优化的基准测试结果如图 2 所示。我们考虑单个时间步（$N=1$），并测量运行时间随 d 的变化。图 2(a) 中的结果确认，全自动微分和混合梯度的每个时间步运行时间相对于 d 具有相同的标度（表 I）。观测到的运行时间标度 $\Theta(d^{\frac{5}{2}})$ 与 $\kappa\sim\Theta(d^2)$ 一致，这是由于 H_1 采用稠密矩阵存储，以及 $\mu\sim\Theta(d^{\frac{1}{2}})$（见附录 E）所致。图 2(b) 展示了所需内存资源的标度：对于全自动微分和混合梯度，内存使用量均随维度 d 按 $\Theta(d^2)$ 标度增长，这与 $\kappa\sim\Theta(d^2)$ 是主要贡献项相一致。正如预期，内存使用量相对于 N 的标度在全自动微分和混合梯度之间有所不同 [，如图 2(c) 所示]: 对于全自动微分，其标度随 N 线性增长；对于混合梯度，其标度与 N 无关。这证实了在时间步数量较多的优化问题中，混合梯度相比全自动微分具有更高的内存效率。

[5]另一种计算 ∇C_1^g 的方法是对所有基态进行并行化的正向和反向传播。其具体的扩展规律取决于并行化方案；相关讨论超出了本文的范围。

[6]本基准测试使用 Autograd 软件包 [38] 实现自动微分。该软件包不支持稀疏线性代数，因此为了公平比较，我们在全自动微分和混合梯度中均采用稠密矩阵存储来存储哈密顿量。

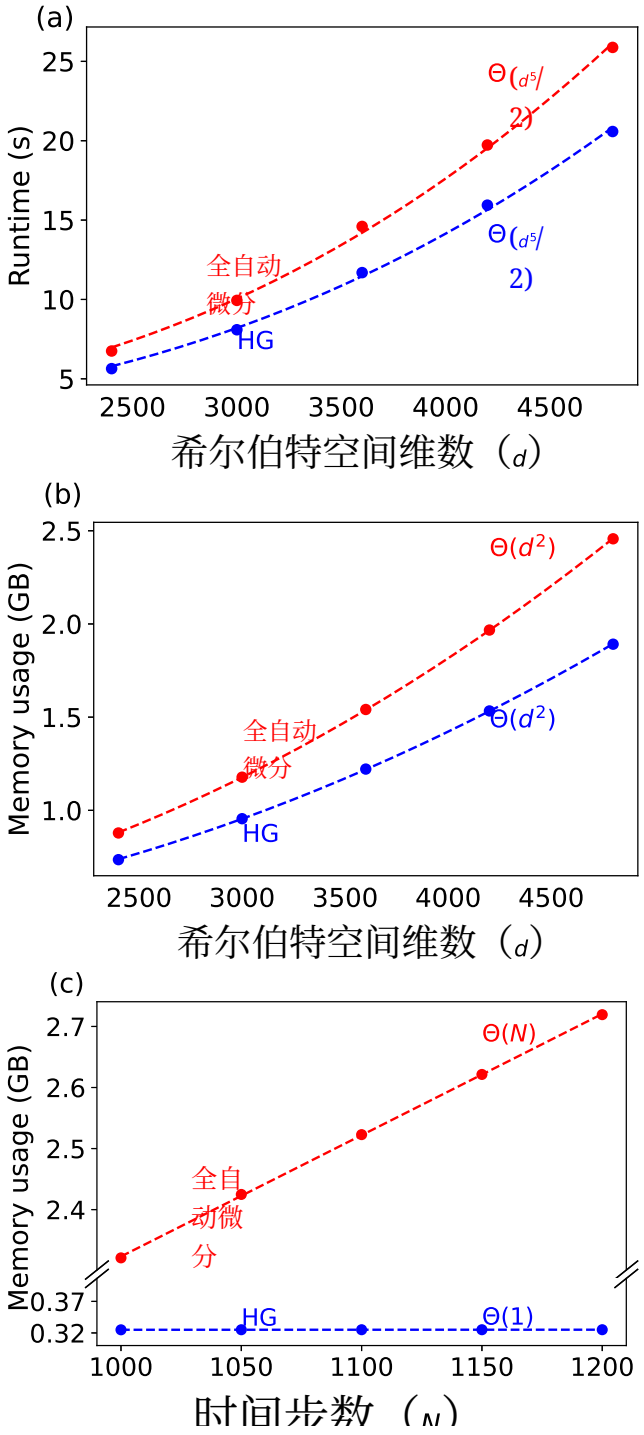

图2. 腔与 transmon 耦合构成的系统中福克态生成的基准测试结果。(a) 每个演化时间步的运行时间和 (b) 内存使用量随希尔伯特空间维数 d 的变化。通过增大腔维数并将 transmon 维数固定为 6 来改变维数。(c) 对于 $d = 600$，内存使用量随时间步数的变化。所有面板中，数据点为基准测试结果，虚线为幂律拟合。（结果基于以下参数获得：$\Delta/2\pi = 3\,\text{GHz}$、$g/2\pi = 0.1\text{GHz}$、$\alpha/2\pi = -0.225\,\text{GHz}$，以及恒定控制 $a_z/2\pi = a_x/2\pi = 0.1\,\text{GHz}$。）

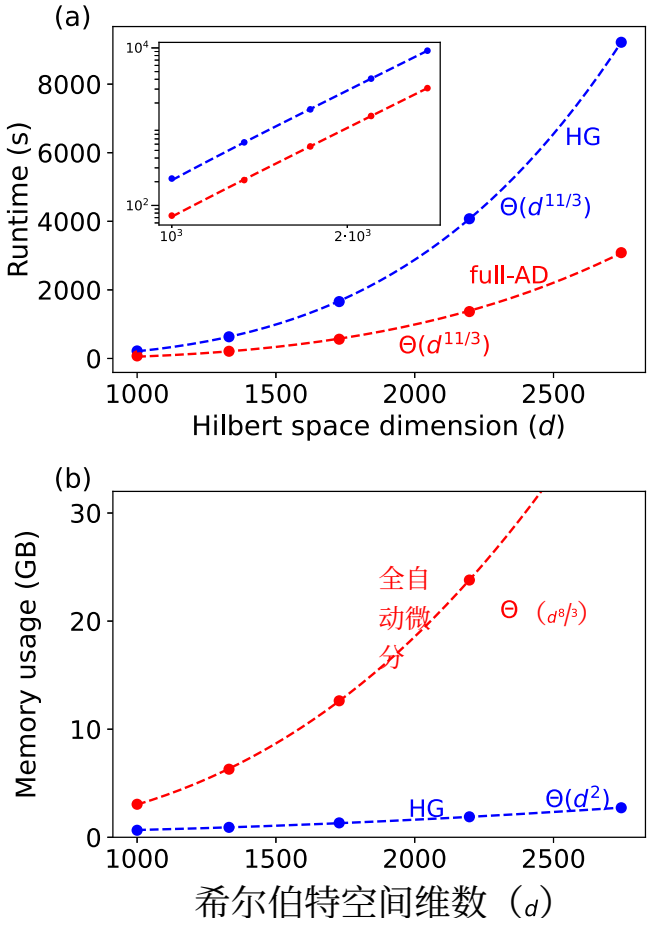

图3. 由三个耦合跨蒙特量子比特组成的系统中，幺正门优化的基准测试结果。(a) 运行时间和 (b) 每个演化时间步的内存使用量随希尔伯特空间维数 d 的变化。通过同时增大每个 transmon 的维数来增加系统维数。插图证实，混合梯度和全自动微分的运行时间标度具有相同的幂律。在所有面板中，数据点表示基准测试结果，虚线表示幂律拟合。（结果使用以下参数获得：$g/2\pi = 0.1$ GHz、$\alpha/2\pi = -0.225$GHz 和 $a^{(\nu)}/2\pi = 0.1$ GHz。）

接下来，我们来看一个优化幺正门的示例。此基准测试针对三个具有最近邻耦合的受驱动 transmon 进行。目标门由分别作用于这三个 transmon 上的同时 Hadamard 操作构成。为明确起见，我们考虑频率相同且受到共振驱动的 transmon。驱动旋转框架中的完整哈密顿量为

$$H_2(t) = \sum_{\nu=1}^{3}\left[\frac{1}{2}\alpha b_\nu^\dagger b_\nu(b_\nu^\dagger b_\nu - 1) + a^{(\nu)}(t)(b_\nu + b_\nu^\dagger)\right] + g(b_1 b_2^\dagger + b_2 b_3^\dagger + \text{h.c.}). \tag{21}$$

优化的目标是最小化保真度损失 C_2^g。此基准测试的设置与前一个示例相同。

图 3 记录了基于单个时间步的基准测试

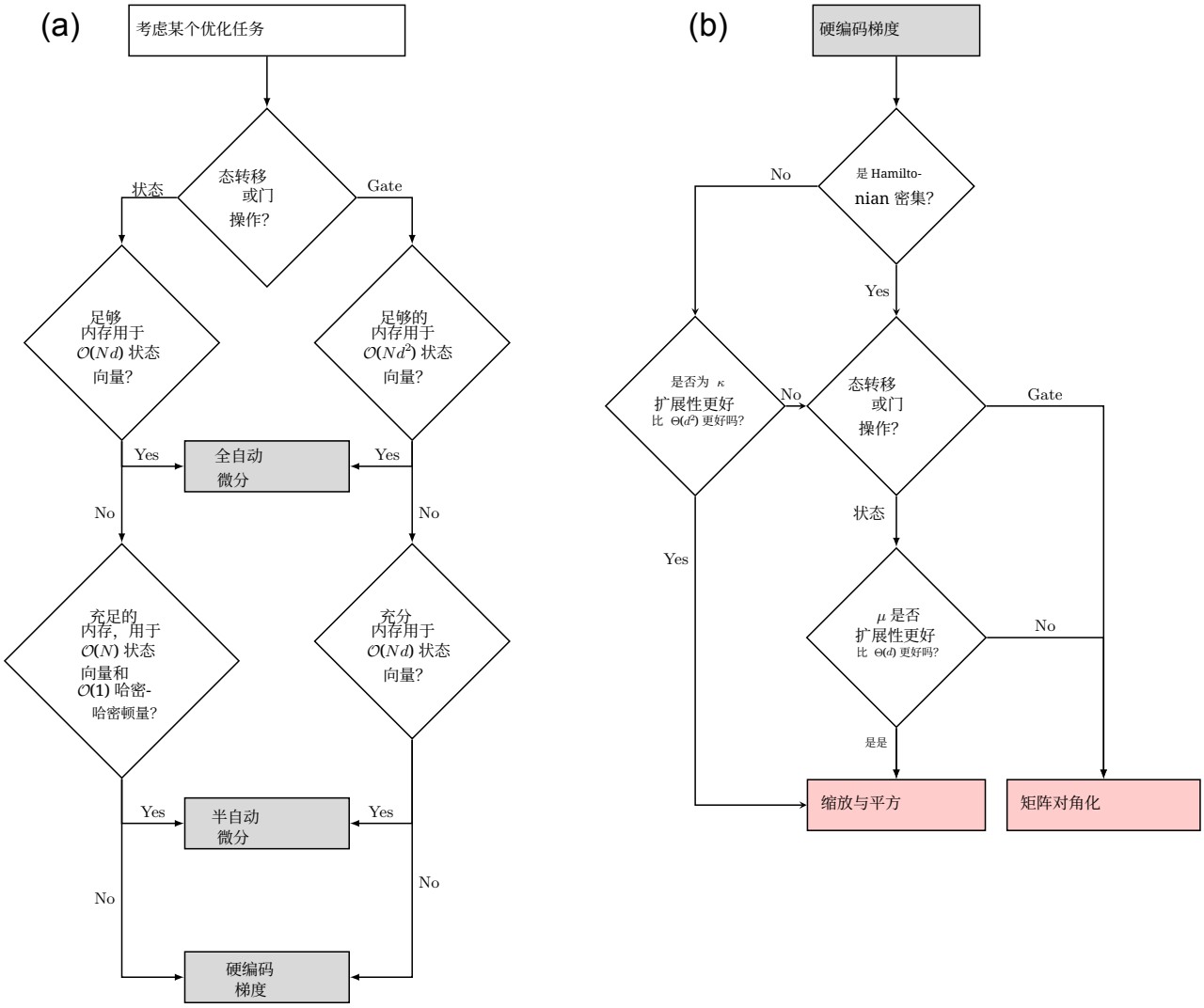

图 4. 用于选择最优数值策略的决策树。(a) 在三种计算梯度的数值方法中进行选择：全自动微分、半自动微分、硬编码解析梯度。(此处我们假设 $\mu \sim \Theta(d)$；不同的标度见表 I。) (b) 确定计算传播子-态乘积的适当策略。

($N = 1$)。对于全自动微分和混合梯度，观测到的运行时间随维度 d 的标度均为 $\Theta(d^{\frac{11}{3}})$，见图 3(a)。这一幂律源于 $\Theta(\mu\kappa d)$ 对于 $\kappa \sim \Theta(d^2)$ 和 $\mu \sim \Theta(d^{\frac{2}{3}})$ [，见附录 E]。内存使用量随 d 的标度如图 3(b) 所示，这证实了与混合梯度相比，全自动微分具有不利的额外 $\mu \sim \Theta(d^{\frac{2}{3}})$ 依赖性（表 I）。在我们的所有基准测试中，这导致全自动微分消耗的内存约为混合梯度的 20 倍。这表明，当大希尔伯特空间维度的优化问题受到内存限制时，混合梯度更为可取。我们强调，这一改进并未恶化运行时间标度[7]。

V. 在不同优化策略之间进行选择

有多个因素决定了对于给定的优化问题，全自动微分、混合梯度还是半自动微分是最优策略，参见决策树图 4。当希尔伯特空间维数 d 和时间步数N 足够小时，内存使用量可能不会成为瓶颈。在这种情况下，全自动微分可能更可取，因为它无需硬编码梯度。如果全自动微分的内存成本难以承受，但解析梯度的计算又十分繁琐，那么半自动微分可能是一种可行的折中方案，不过其内存效率仍低于混合梯度。在其他所有情况下，

[7] 我们注意到，通过缩放与平方实现的混合梯度（HG）的运行时间标度 $\Theta(N\mu\kappa d)$，原则上可能超过矩阵指数运算的标度 $O(Nd^3)$ 基于矩阵对角化（见附录 F）。在这种情况下，后者是更好的选择。

HG 是首选方法，因为与全自动微分和半自动微分相比，它能够处理具有更大 N and d 的优化。

选择通过缩放与平方还是矩阵对角化来实现混合梯度，应取决于哈密顿量是否稀疏，以及 κ 相比 $\Theta(d^2)$ 是否具有更好的扩展性。后一种情况下，缩放与平方更可取，其总体内存使用量为 $\sim \Theta(d+\kappa)$，优于矩阵对角化的内存扩展关系 $O(d^2)$。相反，对于具有 $\kappa \sim \Theta(d^2)$ 的哈密顿量，这两种方法在内存使用量的扩展关系上等效，因此可以根据具体优化任务选择运行时间更短的方法。使用矩阵对角化进行优化时，运行时间按 $O(d^3)$ 扩展，详见附录 F（ []）；而基于缩放与平方的优化运行时间则因态转移和量子门优化而异：（1）对于态转移，运行时间按 $\Theta(\mu d^2)$ 扩展；因此，只要 μ 比 $\Theta(d)$ 扩展得更快，缩放与平方就优于矩阵对角化。（2）对于量子门优化，运行时间按 $\Theta(\mu d^3)$ 扩展，因此矩阵对角化更可取。

VI. 结论

受大型系统量子最优控制中内存瓶颈问题的启发，我们重新审视并比较了 GRAPE 的多种形式在内存使用量和运行时间标度方面的表现：硬编码梯度（混合梯度）、全自动微分和半自动微分。这三种方法的运行时间标度相同，但内存使用量的标度各有特点。按效率从高到低排序如下：混合梯度、半自动微分、全自动微分。因此，面对内存瓶颈问题时，混合梯度是首选方案。

我们分别通过具体的态转移和门优化的基准测试研究，展示了混合梯度和全自动微分的扩展性。为此，我们介绍了缩放与平方算法实现方面的改进，该算法用于数值计算传播子-态乘积。在所研究的示例中，与 SciPy的expm 乘法 函数相比，我们观察到了加速效果。图 4 所示的决策树总结了针对给定优化问题选择适当数值策略的方法。

致谢

我们感谢 Thomas Propson 富有成效的讨论。本材料基于美国能源部科学办公室国家量子信息科学研究中心超导量子材料与系统中心（SQMS）在合同编号 DE-AC02-07CH11359 下支持的工作。我们感谢工作中使用的以下开源软件包：matplotlib [46],NumPy [47], SciPy [48], SymPy [49], scqubits [50], 和 qoc [32]。

附录 A：解析梯度

在正文中，我们专门关注于最小化态转移或门保真度损失。在本附录中，我们为其他感兴趣的代价项提供解析梯度。

成本贡献

$$C_2^{st} = \frac{1}{N}\sum_{n'=1}^{N}\langle\psi_{n'}|\Omega|\psi_{n'}\rangle \tag{A1}$$

惩罚厄米算符 Ω 的积分期望值。该成本贡献的梯度可写为

$$\begin{aligned}\frac{\partial C_2^{st}}{\partial a_n} &= \frac{1}{N}\sum_{n'=1}^{N}\langle\psi_{n'}|\Omega\frac{\partial}{\partial a_n}\prod_{n''=1}^{n'}U_{n''}|\psi_0\rangle + c.c.\\ &= \frac{1}{N}\left(\sum_{n'=n}^{N}\prod_{n''=n+1}^{n'}\langle\psi_{n'}|\Omega U_{n''}\frac{\partial}{\partial a_n}U_n|\psi_{n-1}\rangle\right) + c.c.\\ &= \frac{2}{N}\,\mathrm{Re}(\langle\Phi_n|\frac{\partial}{\partial a_n}U_n|\psi_{n-1}\rangle),\end{aligned} \tag{A2}$$

with

$$|\Phi_n\rangle := \sum_{n'=n}^{N} U_{n+1}^{\dagger}\cdots U_{n'}^{\dagger}\Omega|\psi_{n'}\rangle \tag{A3}$$

向量 $|\Phi_n\rangle$ 遵循递推关系

$$|\Phi_n\rangle = U_{n+1}^{\dagger}|\Phi_{n+1}\rangle + \Omega|\psi_n\rangle. \tag{A4}$$

由于这一关系，式（A2）中的表达式可以采用前向-后向方案进行计算，与第 IIIC节中讨论的方法类似。

成本贡献 $C_3^{st} = 1 - \frac{1}{N}\sum_{n'=1}^{N}|\langle\phi_T \mid \psi_{n'}\rangle|^2$累加了所有时间步的保真度损失（而非仅记录最终状态的保真度损失，如 C_1^{st}中所示）。这会推动系统更快地趋向其目标态，从而有助于缩短态转移时长。与公式（A2）类似， C_3^{st}的梯度为：

$$\frac{\partial C_3^{st}}{\partial a_n} = -\frac{2}{N}\,\mathrm{Re}\left(\langle\Phi_{T,n}|\frac{\partial}{\partial a_n}U_n|\psi_{n-1}\rangle\right), \tag{A5}$$

with

$$|\Phi_{T,n}\rangle := \sum_{n'=n}^{N} U_{n+1}^{\dagger}\cdots U_{n'}^{\dagger}|\phi_T\rangle\langle\phi_T|\psi_{n'}\rangle. \tag{A6}$$

该向量遵循递归关系

$$|\Phi_{T,n}\rangle = U_{n+1}^{\dagger}|\Phi_{T,n+1}\rangle + |\phi_T\rangle\langle\phi_T|\psi_n\rangle. \tag{A7}$$

这使我们能够使用前向-后向方案来评估公式（A5）中的梯度表达式。

对于门操作，类似于 C_3^{st}的成本贡献表示为 $C_3^{g} = 1 - \sum_{n'=1}^{N}\left|\mathrm{tr}\left(U_T^{\dagger}U_{n',1}\right)\right|^2/Nd^2$，其中

$U_{n',1} = U_{n'} \cdots U_1$ 描述了从时间步 1 到 n'的演化。我们得到梯度

$$\frac{\partial C_3^g}{\partial a_n} = -\frac{2}{Nd^2} \sum_{h=1}^{d} \mathrm{Re}\left(\langle \Phi_{T,n}^h | \frac{\partial}{\partial a_n} U_n | \psi_{n-1}^h \rangle\right), \tag{A8}$$

with

$$|\Phi_{T,n}^h\rangle := \sum_{n'=n}^{N} U_{n+1}^\dagger \cdots U_{n'}^\dagger U_T |\psi_0^h\rangle \,\mathrm{tr}(U_{n',1} U_T^\dagger). \tag{A9}$$

这里，$\{|\psi_0^h\rangle\}_{h=1,2,\ldots,d}$ 构成任意正交归一基，而 $h = 1, 2, \ldots, d$ 枚举这些基态。向量 $|\Phi_{T,n}^h\rangle$ 满足递推关系

$$|\Phi_{T,n}^h\rangle = U_{n+1}^\dagger |\Phi_{T,n+1}^h\rangle + U_T |\psi_0^h\rangle \,\mathrm{tr}(U_{n,1} U_T^\dagger), \tag{A10}$$

从而再次能够通过前向-后向传播方案计算梯度表达式。

附录 B：标度分析

对于附录 A中讨论的代价项，我们现在分析全自动微分、半自动微分和混合梯度的内存使用量及运行时间标度。

a. 标度分析：硬编码梯度。得益于表达式的相似性以及递推关系的存在，也可以将类似于图1 中的前向-后向方案应用于 ∇C_2^{st},∇C_3^{st}和∇C_3^g。因此，我们发现第IVA 节中的标度分析得出的内存使用量和运行时间需求与 ∇C_1^{st}和 ∇C_2^g 相同。

*b.*标度分析：全自动微分。 尽管代价项的表达式之间存在具体差异 C_2^{st}， C_3^{st}以及 C_3^g ，我们发现，运行时间和内存使用量的标度再次由存储中间向量以及反复执行矩阵-向量乘法的需求所决定。因此，其标度与第IVA 中讨论的代价项相同。

总之，态转移和门操作的代价项的缩放与表格中所示一致I。关于半自动微分的缩放，请参见参考文献 34。

附录 C：确定上界 $E_A(m, s)$

对 $e^A|\Psi\rangle$ 在第IIIB 中的近似需要确定一个上界 $E_A(m,\ s)$ 对于误差$\varepsilon_{A,\ |\Psi\rangle}(m,\ s)$ [式（14）]。在本附录中，我们将展示如何获得这样的上界。

考虑误差 [式（14）]，，我们注意到分母恰好为 1。误差 $\varepsilon_{A,\ |\Psi\rangle}$ 满足以下界限：

$$\begin{aligned}\varepsilon_{A,|\Psi\rangle} &= \left|\left|e^A|\Psi\rangle - (e_m^B)^s|\Psi\rangle + (e_m^B)^s|\Psi\rangle - [\![(e_m^B)^s|\Psi\rangle]\!]\right|\right|_2 \\ &\le \underbrace{\left|\left|e^A|\Psi\rangle - (e_m^B)^s|\Psi\rangle\right|\right|_2}_{\varepsilon_t} + \underbrace{\left|\left|(e_m^B)^s|\Psi\rangle - [\![(e_m^B)^s|\Psi\rangle]\!]\right|\right|_2}_{\varepsilon_r}.\end{aligned} \tag{C1}$$

误差 ε_t 源于泰勒级数的截断，而舍入误差 ε_r 则由有限精度算术引起。下面我们分别推导这两种误差的上界。

1. 截断误差 ε_t的上界

我们将截断误差改写为

$$\begin{aligned}\varepsilon_t &= \left|\left|e^A|\Psi\rangle - (e^B - R_m(B))^s|\Psi\rangle\right|\right|_2 \\ &= \left|\left|\sum_{i=1}^{s} \frac{s!}{i!(s-i)!}(-R_m(B))^i (e^B)^{s-i}|\Psi\rangle\right|\right|_2,\end{aligned} \tag{C2}$$

其中， $R_m(B) = \sum_{q=m+1}^{\infty} B^q/q!$ 表示截断指数级数的矩阵值误差。利用三角不等式和次乘性 [51]，我们得到该界限

$$\begin{aligned}\varepsilon_t &\le \sum_{i=1}^{s} \frac{s!}{i!(s-i)!}\left(R_m(||B||_2)\right)^i \left|\left|e^B\right|\right|_2^{s-i} |||\Psi\rangle||_2 \\ &= \sum_{i=1}^{s} \frac{s!}{i!(s-i)!}\left(R_m(||B||_2)\right)^i.\end{aligned} \tag{C3}$$

此处第二行利用了状态已归一化且 e^B 是幺正的这一事实。

必须通过数值计算此上界，以确定 m 和 s。然而，计算矩阵 2-范数并不方便，因为这需要使用特征值求解器。为了解决这一问题，我们利用[8]，改用矩阵 1-范数。

$$||B||_2 \le ||B||_1 . \tag{C4}$$

R_m 的单调性使我们

$$\begin{aligned}\varepsilon_t &\le \sum_{i=1}^{s} \frac{s!}{i!(s-i)!}\left(R_m(||B||_1)\right)^i \\ &< sR_m(||B||_1)\frac{1-\left(sR_m(||B||_1)\right)^s}{1-sR_m(||B||_1)}.\end{aligned} \tag{C5}$$

2. 舍入误差的上界 ε_r

舍入误差 ε_r 源于实数的浮点表示所具有的有限精度，以及基本算术运算的有限准确度。实数 ξ 的浮点表示 $[\![\xi]\!]$ 可写为 $[\![\xi]\!] = \xi(1+\delta)$，其中 $\delta = \delta(\xi)$ 是相对偏差 [51]。其大小始终小于机器精度，即 $|\delta| < u$。类似地，复数 z 满足

$$[\![z]\!] = z(1+\delta) \tag{C6}$$

[8] 通常， $||B||_2 \le \sqrt{||B||_\infty ||B||_1}$ [52] 对任意 B均成立。当 B 为反厄米时， $||B||_\infty = ||B||_1$ 可得不等式（C4）。

其中 $|\delta| \leq u$。对两个复数 z_1，z_2 进行加法和乘法等算术运算时，会产生相对偏差 $\delta = \delta(z_1, z_2)$，因此

$$\begin{aligned} &[\![z_1 + z_2]\!] = ([\![z_1]\!] + [\![z_2]\!])(1+\delta),\ |\delta| \leq u, \\ &[\![z_1 z_2]\!] = [\![z_1]\!][\![z_2]\!](1+\delta),\ |\delta| \leq u', \end{aligned} \tag{C7}$$

其中 $u' := 2\sqrt{2}u/(1-2u)$ [51]。公式（C6）、（C7）中所示的相对偏差通常取决于所涉及的两个数以及所进行的运算。下文中，我们将区分不同的 δ，其下标为 ν。

利用公式（C6）、（C7，我们推导点积计算中舍入误差的一个上界。点积是根据公式（$e_m^B|\Psi\rangle$ 13）计算的基本算术运算。

a. 计算点积时舍入误差的上界

计算点积时产生的舍入误差为 $|\Sigma_d - [\![\Sigma_d]\!]|$，其中 $\Sigma_d = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{z}$（$\boldsymbol{w}$，$\boldsymbol{z} \in \mathbb{C}^d$）。这里，$\boldsymbol{w}^{\mathrm{T}}$ 是 $B = i\delta tH/s$ 的一行。我们说明如何界定 $d = 1$，2的舍入误差，并随后给出一般结果。根据公式（C6）和（C7），我们有

$$\begin{aligned} |[\![\Sigma_1]\!] - \Sigma_1| &= |[\![w_1 z_1]\!] - w_1 z_1| \\ &< |w_1||z_1|\Big|\prod_{\nu=1}^{3}(1+\delta_\nu) - 1\Big| < \gamma_3 |w_1||z_1|. \end{aligned} \tag{C8}$$

在最后一步中，我们使用了 $\left|\prod_{\nu=1}^{n}(1+\delta_\nu) - 1\right| < \frac{nu'}{1-nu'}$ [51]，并定义了 $\gamma_n := \frac{nu'}{1-nu'}$。对于 $d = 2$，误差的上界为

$$\begin{aligned} |[\![\Sigma_2]\!] - \Sigma_2| &= |([\![\Sigma_1]\!] + [\![w_2 z_2]\!])(1+\delta_1) - \Sigma_2| \\ &< \gamma_4 \sum_{i=1}^{2} |w_i||z_i|. \end{aligned} \tag{C9}$$

任意 d 的一般上界为：

$$|[\![\Sigma_d]\!] - \Sigma_d| < \gamma_{d+2} \sum_{j=1}^{d} |w_j||z_j|. \tag{C10}$$

由于 $\boldsymbol{w}^{\mathrm{T}}$ 与哈密顿量矩阵的一行成正比，H 的稀疏性意味着 $\boldsymbol{w}^{\mathrm{T}}$的稀疏性。$\boldsymbol{w}^{\mathrm{T}}$中的零元素具有特殊性，因为它们不会产生舍入误差。因此，可以得到更强的上界

$$|[\![\Sigma_d]\!] - \Sigma_d| < \gamma_{\sigma+2} \sum_{j=1}^{d} |w_j||z_j| \tag{C11}$$

其中，σ 是 $\boldsymbol{w}^{\mathrm{T}}$中的非零元素数量。

b. $e_m^B|\Psi\rangle$的舍入误差上界

向量 $e_m^B|\Psi\rangle$ 通过以下方式递归求值：

$$b_j = \frac{B}{j} b_{j-1}, \qquad e_j^B b_0 = b_{j-1} + b_j \tag{C12}$$

其中 $j = 1, 2, \cdots, m$，而 b_0是在一组正交归一基下对 $|\Psi\rangle$ 的向量表示。在计算误差 $\left|\left|[\![e_m^B b_0]\!] - e_m^B b_0\right|\right|_2$的上界之前，我们首先推导向量分量$\left|[\![e_m^B b_0]\!]^{(i)} - (e_m^B b_0)^{(i)}\right|$误差的上界。我们说明如何对 $m = 0$，1的这一误差进行界定，然后给出一般结果。

对于 $m = 0$ 情形，我们有如下界限

$$\left|[\![e_0^B b_0]\!]^{(i)} - (e_0^B b_0)^{(i)}\right| = \left|[\![b_0]\!]^{(i)} - b_0^{(i)}\right| < \gamma_2 |b_0^{(i)}|. \tag{C13}$$

然后，我们得到 $m = 1$ 示例的误差上界：

$$\begin{aligned} &|[\![e_1^B b_0]\!]^{(i)} - (e_1^B b_0)^{(i)}| \\ &= \left|\left([\![b_0]\!]^{(i)} + [\![b_1]\!]^{(i)}\right)(1+\delta_1) - (e_1^B b_0)^{(i)}\right| \\ &= \left|[\![b_0]\!]^{(i)}(1+\delta_1) + \frac{[\![Ab_0]\!]^{(i)}}{s}\prod_{\nu=1}^{3}(1+\delta_\nu)^3 - (e_1^B b_0)^{(i)}\right| \\ &\overset{(C11)}{<} \gamma_3 |b_0^{(i)}| + \gamma_{\sigma_i+5} \sum_l |B^{(il)}||b_0^{(l)}|, \end{aligned} \tag{C14}$$

其中，σ_i 是矩阵 B 的第 i 行中的非零元素数量。此处，第三行源自复数与实数相除时产生的除法误差，即 $[\![z/\xi]\!] = z/\xi(1+\delta)$，$|\delta| < u$ [51]。任意 m 的一般上界为：

$$\left|[\![e_m^B b_0]\!]^{(i)} - (e_m^B b_0)^{(i)}\right| < \sum_{k=0}^{m} \gamma_{k(\sigma_i+2)+m+2} \left(\frac{|B|^k}{k!}|b_0|\right)^i, \tag{C15}$$

其中 $|B|$, $|b_0|$ 分别表示元素为$|B^{il}|$、$|b_0^i|$的矩阵和向量。定义

$$\sigma' := \max_i \sigma_i \tag{C16}$$

并使用公式（C15），我们得到

$$\begin{aligned} \left|\left|[\![e_m^B b_0]\!] - (e_m^B b_0)\right|\right|_2 &= \left|\left|\left|[\![e_m^B b_0]\!] - (e_m^B b_0)\right|\right|\right|_2 \\ &< \sum_{k=0}^{m} \gamma_{k(\sigma'+2)+m+2} \left|\left|\frac{|B|^k}{k!}|b_0|\right|\right|_2 \\ &< \sum_{k=0}^{m} \gamma_{k(\sigma'+2)+m+2} \frac{\left|\left|\left||B|\right|\right|_2^k}{k!} \left|\left|\left||b_0|\right|\right|\right|_2 \\ &\overset{(C4)}{<} \sum_{k=0}^{m} \gamma_{k(\sigma'+2)+m+2} \frac{\left|\left|\left||B|\right|\right|_1^k}{k!} \left|\left|\left||b_0|\right|\right|\right|_2 \\ &= \beta ||b_0||_2 \end{aligned} \tag{C17}$$

使用 $\beta := \sum_{k=0}^{m} \gamma_{k(\sigma'+2)+m+2} \frac{||B||_1^k}{k!}$。这里，$\beta||b_0||_2$ 所需的舍入误差上界为$e_m^B|\Psi\rangle$。

c. 误差的上界 ε_r

向量 $(e_m^B)^s b_0$ 通过递归关系求得。

$$c_l = e_m^B c_{l-1}, \qquad l = 1, 2, \cdots, s, \tag{C18}$$

其中 $c_0 = b_0$。我们说明如何为 $\varepsilon_r = \left|\left|[\![(e_m^B)^s b_0]\!] - (e_m^B)^s b_0\right|\right|_2$（$s = 1,2$）中的误差给出上界，并随后给出一般结果。

对于 $s = 1$，我们有

$$\left|\left|[\![e_m^B b_0]\!] - e_m^B b_0\right|\right|_2 \overset{(C17)}{<} \beta ||b_0||_2 = \beta. \tag{C19}$$

对于 $s = 2$，我们得到

$$\begin{aligned}
&\left|\left|[\![(e_m^B)^2 b_0]\!] - (e_m^B)^2 b_0\right|\right|_2 \\
&\leq \left|\left|[\![(e_m^B)^2 b_0]\!] - e_m^B [\![e_m^B b_0]\!]\right|\right|_2 + \left|\left|e_m^B [\![e_m^B b_0]\!] - (e_m^B)^2 b_0\right|\right|_2 \\
&< \left|\left|[\![e_m^B [\![e_m^B b_0]\!]]\!] - e_m^B [\![e_m^B b_0]\!]\right|\right|_2 + \beta \left|\left|e_m^B\right|\right|_2,
\end{aligned} \tag{C20}$$

其中最后一步由方程（C19）得出。最后，我们将第二项中的矩阵范数界定为

$$\begin{aligned}
\left|\left|e_m^B\right|\right|_2 = \left|\left|e^B - R_m(B)\right|\right|_2 &\leq 1 + R_m(||B||_2) \\
&\overset{(C4)}{\leq} 1 + R_m(||B||_1) =: \alpha.
\end{aligned} \tag{C21}$$

由于方程（C17）对任意复向量都成立，b_0，方程（C20）中的误差还可进一步界定为：

$$\begin{aligned}
&\left|\left|[\![(e_m^B)^2 b_0]\!] - (e_m^B)^2 b_0\right|\right|_2 \overset{(C17)}{<} \beta \left|\left|[\![e_m^B b_0]\!]\right|\right|_2 + \beta\alpha \\
&= \beta \left|\left|[\![e_m^B b_0]\!] - e_m^B b_0 + e_m^B b_0\right|\right|_2 + \beta\alpha \\
&\overset{(C19)}{<} \beta(\beta + \alpha) + \beta\alpha.
\end{aligned} \tag{C22}$$

进一步推广这一策略，我们得到 ε_r 上界的表达式：

$$\begin{aligned}
\varepsilon_r &= \left|\left|[\![(e_m^B)^s b_0]\!] - (e_m^B)^s b_0\right|\right|_2 \\
&\leq \sum_{l=0}^{s-1} \left|\left|(e_m^B)^l [\![(e_m^B)^{s-l} b_0]\!] - (e_m^B)^{l+1} [\![(e_m^B)^{s-l-1} b_0]\!]\right|\right|_2 \\
&< \beta \sum_{l=0}^{s-1} (\alpha + \beta)^{s-l-1} \alpha^l = (\alpha + \beta)^s - \alpha^s.
\end{aligned} \tag{C23}$$

结合 ε_r 和 ε_t 的上界，我们回到公式（C1）并得到

$$\begin{aligned}
\varepsilon_{A,|\Psi\rangle} < E_A(m, s) = s R_m(||B||_1) &\frac{1 - \left(s R_m(||B||_1)\right)^s}{1 - s R_m(||B||_1)} \\
&+ (\alpha + \beta)^s - \alpha^s,
\end{aligned} \tag{C24}$$

其中 $||B||_1 = ||A||_1 / s$，$\alpha = \alpha(||A||_1, m, s)$ 和 $\beta = \beta(\sigma', ||A||_1, m, s)$。

附录 D：缩放与平方实现之间的数值比较

使用缩放与平方来计算传播子与状态的乘积，$e^{-iH\Delta t}|\Psi\rangle$。我们比较自定义缩放与平方实现和SciPy实现的相对误差和expm 乘法 [41]。自定义实现和SciPy实现的相对误差遵循方程（14）中的一般表达式，但 m 和 s的具体选择有所不同。作为精确传播子与状态乘积的近似值，我们使用SymPy 以 400 位精度对矩阵指数进行泰勒展开，所得乘积在小数点后 30 位达到收敛。借助`Temci` 软件包测量运行时间[44]。

作为这项比较的基础，我们使用哈密顿量（20）和（21）的示例，并对其采用稀疏存储。相对误差分别作为矩阵范数 $||-iH\Delta t||_2$ 的函数显示在图 τ5（a）和（c）中。对于这两个示例，我们的实现所产生的相对误差均受误差容限 τ限制，而 SciPy 的实现所产生的相对误差在范数较大时往往会超过该容限。这种趋势有两个原因。首先，在推导误差 [41]的上界时没有考虑舍入误差；其次，e_m^B [公式（12）]的截断仅基于启发式标准。

尽管SciPy实现的相对误差更大，但其矩阵-向量乘法次数更少，记为 μ。图5(b) 和 (d) 展示了 μ 在我们的实现与SciPy实现之间的比值始终大于 1。虽然SciPy实现进行矩阵-向量乘法所花费的时间更少，但其总运行时间却长于我们的实现，见图5(b) 和 (d)。这是因为SciPy实现确定(m, s)时的运行时间开销更大。

总之，在我们的数值实验中，与 SciPy 的实现相比，我们的实现在数值稳定性和运行时间方面表现更好。

附录E：μ 的缩放

正文中讨论的运行时间渐近行为由 μ (16) 随希尔伯特空间维数 d的缩放规律决定。其依赖关系 $\mu(d)$ 取决于待求指数的矩阵 A （该矩阵与哈密顿量成正比）。这一点以及最小化约束(15) 的复杂性，使我们无法给出一般的缩放关系。因此，下面转而讨论几个具体示例。

我们首先考虑 transmon-腔系统的例子，见式（20），其中我们通过增大腔的维数 d 来增大 d_c，同时保持 transmon 截断维数不变。相应的哈密顿量 H_1用裸乘积基表示时，每一行的非零元素最大数量恒定。（即，参数 σ' (C16) 与 d 无关。）数值结果表明，在这种情况下，μ 随..... 线性增长

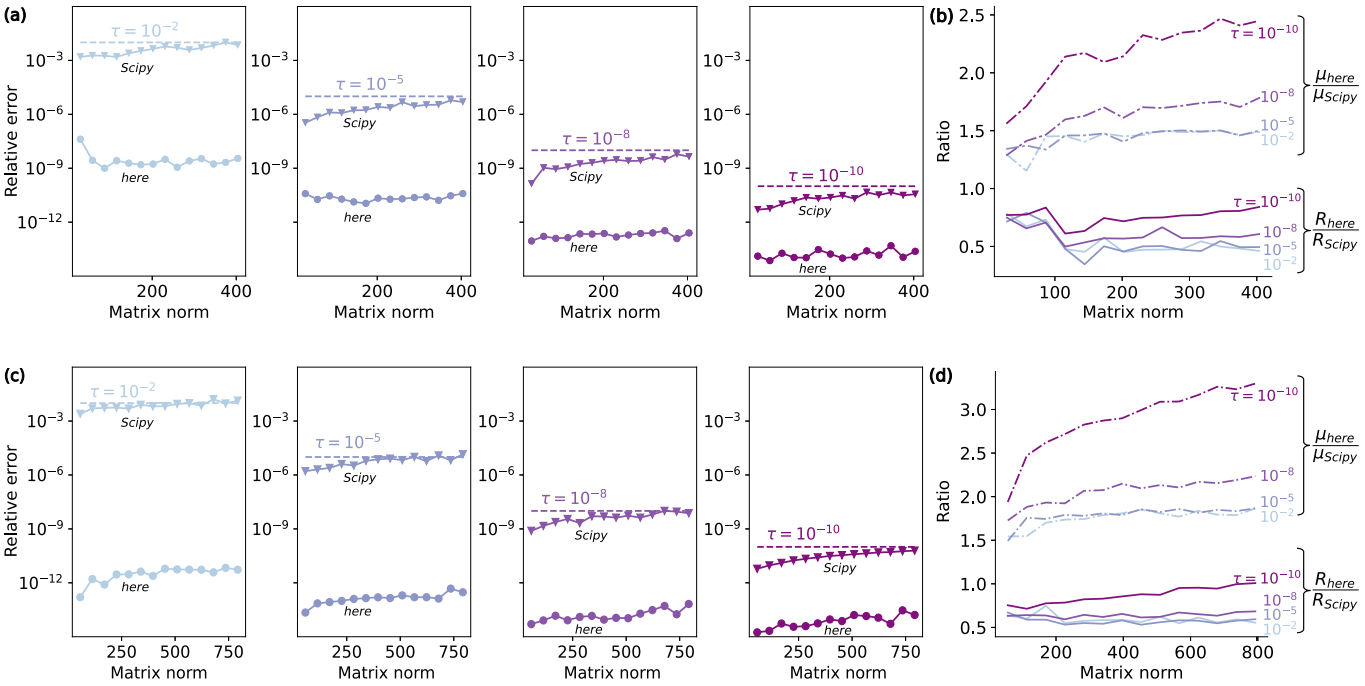

图 5。比较本文实现的缩放与平方方法与 SciPy 的 expm 乘法，考察相对误差和运行时间。(a)、(c) 相对误差随矩阵范数的变化。(b)、(d) 我们的方法与 μ（即矩阵-向量乘法次数）之比随矩阵范数的变化。矩阵范数 $||-iH\Delta t||_2$ 通过增大 $\Delta t \in [1, 10]$来改变。所有面板中，颜色表示不同的误差容限 τ。第一行和第二行的结果分别基于腔-跨蒙特量子比特哈密顿量和三个耦合 Transmon 哈密顿量得出。这两个哈密顿量的系统参数分别与图注中的参数相同：图 2、图 3。

$||A||_1$ ($\mu \sim \Theta(||A||_1)$)，且这种线性关系在很宽的 τ 范围内成立，见图 6(a)。在此例中，有两个因素导致了 $||A||_1 \sim \Theta(d^{\frac{1}{2}})$ 的缩放：（1） d 是 d_c 的常数倍；（2） H_1 中的腔升降算符具有 1-范数缩放 $\Theta(d_c^2)$。因此，我们得到总体缩放关系 $\mu \sim \Theta(d^{\frac{1}{2}})$，见图 6(e)。在第二个例子中，我们考虑由三个耦合跨蒙特量子比特组成的系统，见式（21）。哈密顿量 H_2 用谐振子基表示，且具有恒定的 σ'，由此得到 $\mu \sim \Theta(||A||_1)$ [图 6(b)]。算符 $(b^\dagger b)^2$ 在 H_2 中的 1- 范数随每个 transmon 的维数呈二次缩放。由于我们通过同时增大三个 transmon 的维数来增大 d，因此得到 $||A||_1 \sim \Theta(d^{\frac{2}{3}})$。这种缩放使得 $\mu \sim \Theta(d^{\frac{2}{3}})$，见图 6(f)。

对于其他满足 σ' 取决于 d 的哈密顿量矩阵，缩放关系 $\mu \sim \Theta(||A||_1)$ 仍可能成立。接下来，我们展示两个具体例子：一个例子中线性关系成立，另一个例子中则不成立。我们首先考虑一个由 N_q 个量子比特组成、具有最近邻 σ_z 耦合的受驱动系统。在旋转框架中，该系统可描述为

$$H_3(t) = \sum_{\nu=1}^{N_q} [a_x^{(\nu)}(t)\sigma_x^{(\nu)} + a_y^{(\nu)}(t)\sigma_y^{(\nu)}] + \sum_{\nu=1}^{N_q-1} g\sigma_z^{(\nu)}\sigma_z^{(\nu+1)}.$$

每个量子比特都通过驱动场进行控制 $a_x^{(n)}$和 $a_y^{(n)}$。对于这个示例，数值结果表明，缩放 $\mu \sim \Theta(||A||_1)$ 仍然成立，见图6(c)。由于我们将 d 增加了增大 N_q，我们得到 $||A||_1 \sim \Theta(N_q) \sim \Theta(\log_2 d)$。这种缩放使 $\mu \sim \Theta(\log_2 d)$，见图6(g)。对于第二个示例，我们考虑两个电容耦合 Fluxonium 量子比特，其哈密顿量为

$$H_4(t) = \sum_{i=1}^{2} \left(4E_{Ci}n_i^2 - E_{Ji}\cos(\varphi_i) + \frac{E_{Li}}{2}[\varphi_i - \phi_i(t)]^2\right) + gn_1n_2. \tag{E1}$$

此处，E_{Ci}、E_{Ji}和 E_{Li} 分别表示两个Fluxonium量子比特的充电能、电感能和约瑟夫森能，而 g 表示相互作用强度。φ 和 n 分别是相位算符和电荷数算符，按通常方式定义。我们在谐振子基中表示哈密顿量。图6(d)显示，μ 随 $||A||_1$ 的标度呈超线性，适用于 $\tau = 10^{-8}$。因此，我们通过同时增加两个Fluxonium量子比特的维数截断值，数值确定 μ 与 d之间的 $\tau = 10^{-8}$ 标度关系。观察到的标度关系为 $\mu \sim \Theta(d^{0.65})$，见图6(h)。

附录F：混合梯度方案的另一种数值实现及相应的标度分析

硬编码梯度方案的数值实现需要计算短时传播子及其导数。在某些条件下，以数值方式计算短时传播子会更有优势。

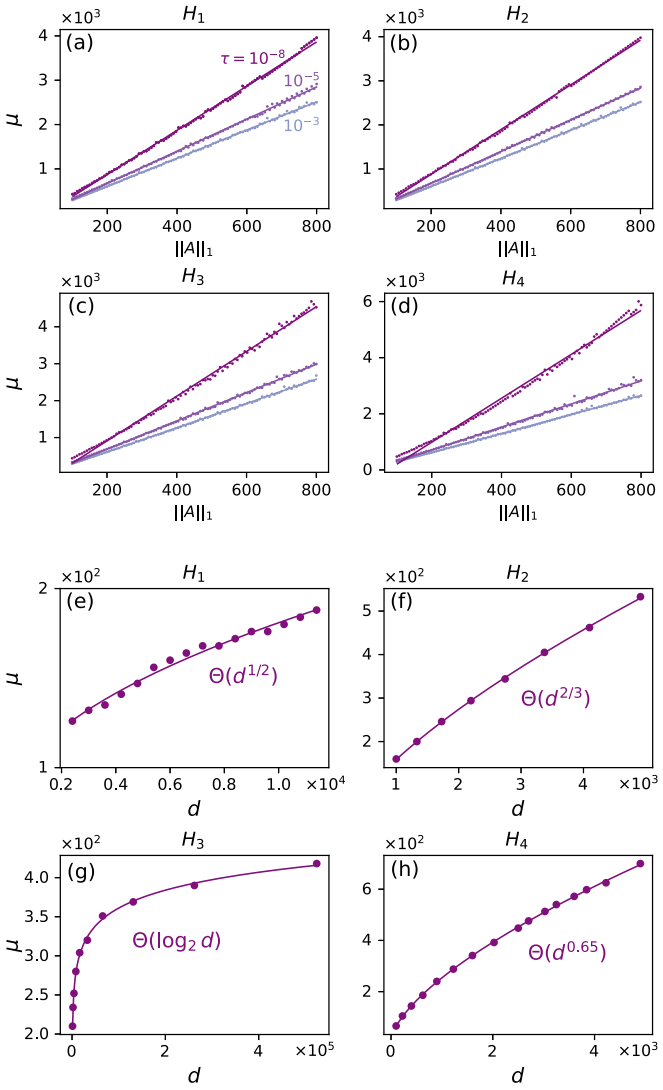

图6. μ的缩放，即矩阵-向量乘法的次数。(a)-(d) 当考虑多个误差容限时， μ 随 $||A||_1$ 的缩放情况，对应哈密顿量 H_j、$j = 1\ 2\ 3\ 4$ τ 、、、。(e)-(h) 分别考虑每个 H_j 的 $\tau = 10^{-8}$ 时， μ 随维度 d 的缩放情况。所有图中，点表示采样数据，实线表示拟合曲线。 H_1、 H_2 的参数设置与图2、图3的图注中给出的相同。 H_3 和 H_4 的参数如下：对于 H_3， $E_{Ci}/2\pi = 2.5$ GHz、 $E_{Ji}/2\pi = 8.9$ GHz、 $E_{Li}/2\pi = 0.5$ GHz、 $g/2\pi = 0.1$ GHz和 $\phi = 0.33$；对于 H_4，$a_x^{(\nu)}/2\pi = a_y^{(\nu)}/2\pi = 0.5$ GHz和 $g/2\pi = 0.1$ GHz。

通过采用矩阵对角化。在本附录中，我们首先讨论计算传播子导数的方法，然后分析混合梯度的缩放。

1. 传播子导数的评估

算符 $-iH\Delta t$ 是反厄米的，因此可以对角化并按如下方式求指数。

$$A = SDS^\dagger, \tag{F1}$$
$$e^A = Se^D S^\dagger. \tag{F2}$$

S 是幺正的， D 是对角的。因此，传播子导数可以写成

$$\frac{de^A}{da} = Se^D\frac{dD}{da}S^\dagger + \frac{dS}{da}e^D S^\dagger + Se^D\frac{dS^\dagger}{da}. \tag{F3}$$

我们对 $\frac{de^A}{da}$ (F3) 应用逆幺正变换，得到

$$S^\dagger\frac{de^A}{da}S = e^D\frac{dD}{da} + S^\dagger\frac{dS}{da}e^D + e^D\frac{dS^\dagger}{da}S. \tag{F4}$$

基于 $\frac{dS^\dagger}{da}S + S^\dagger\frac{dS}{da} = 0$，方程（F4）可改写为

$$S^\dagger\frac{de^A}{da}S = e^D\frac{dD}{da} + E\circ(S^\dagger\frac{dS}{da}) \tag{F5}$$

其中 $E_{ij} := e^{D_{jj}} - e^{D_{ii}}$。这里， $\circ$ 表示 Hadamard乘积（逐元素乘法）。为了得到 dD/da 和 dS/da，我们对式（F1）求导，并重复式（F4）和（F5）中的相同步骤，得到

$$S^\dagger\frac{dA}{da}S = \frac{dD}{da} + E'\circ(S^\dagger\frac{dS}{da}), \tag{F6}$$

其中 $E'_{ij} := D_{jj} - D_{ii}$。由于 E'的对角元素为零，因此可得

$$\frac{dD}{da} = I\circ(S^\dagger\frac{dA}{da}S). \tag{F7}$$

对于式（F6）中的算符的非对角元素，我们得到

$$F\circ(S^\dagger\frac{dA}{da}S) + G = S^\dagger\frac{dS}{da}, \tag{F8}$$

其中我们定义

$$F_{ij} := \begin{cases}\frac{1}{D_{jj}-D_{ii}}, & \text{if } D_{jj}-D_{ii}\neq 0.\\ 0, & \text{otherwise.}\end{cases} \tag{F9}$$
$$G_{ij} := \begin{cases}0, & \text{if } D_{jj}-D_{ii}\neq 0.\\ (S^\dagger\frac{dS}{da})_{ij}, & \text{otherwise.}\end{cases}$$

将式（F7）和（F8）代入式（F5）可得

$$\frac{de^A}{da} = S\big((e^D + E\circ F)\circ(S^\dagger\frac{dA}{da}S)\big)S^\dagger, \tag{F10}$$

其中我们使用 $E\circ G = 0$。

2. 标度分析

对于代价函数各项的梯度（针对态转移情况），采用正向—反向传播方案进行计算。在该方案中，状态所需的存储空间不会随中间时间步的数量增加而增长。必须存储的矩阵包括系统和控制哈密顿量，以及稠密矩阵 dense matrices S, E 和 E'。因此，总内存使用量按 $O(d^2)$ 增长。该方案的运行时间主要由 $O(N)$ 决定传播子及其导数的计算。采用矩阵对角化计算传播子所需的运行时间标度为 $\sim O(d^3)$ [53]。计算传播子导数涉及矩阵乘法和Hadamard乘积，其运行时间标度分别为 $O(d^3)$ 和 $O(d^2)$。因此，HG方案的总体运行时间标度为 $O(Nd^3)$。

门操作代价函数各项贡献的梯度以类似的方式进行评估，从而得到运行时间和内存的扩展规律。

[1] K. Rojan、D. M. Reich、I. Dotsenko、J.-M. Raimond、C. P. Koch 和 G. Morigi，《物理评论 A》90，023824（2014）。[2] G. Waldherr、Y. Wang、S. Zaiser 等，《自然》506，204（2014）。[3] J. M. Gertler、B. Baker、J. Li、S. Shirol、J. Koch 和 C. Wang，《自然》590，243（2021）。[4] M. Abdelhafez、B. Baker、A. Gyenis、P. Mundada、A. A. Houck、D. Schuster 和 J. Koch，《物理评论 A》101，022321（2020）。[5] J. L. Allen、R. Kosut、J. Joo、P. Leek 和 E. Ginossar，《物理评论 A》95，042325（2017）。[6] S.-Y. Huang 和 H.-S. Goan，《物理评论 A》90，012318（2014）。[7] M. Werninghaus、D. J. Egger、F. Roy、S. Machnes、F. K. Wilhelm 和 S. Filipp，《npj 量子信息》7，14（2021）。[8] S. J. Glaser、U. Boscain 等，《欧洲物理学杂志 D》69，279（2015）。[9] C. Bretschneider、A. Karabanov、N. C. Nielsen 和 W. K¨ock- enberger，《化学物理学杂志》136，094201（2012）。[10] C. T. Kehlet、A. C. Sivertsen、M. Bjerring、T. O. Reiss、N. Khaneja、S. J. Glaser 和 N. C. Nielsen，《美国化学会志》126，10202（2004）。[11] N. Khaneja、T. Reiss、C. Kehlet、T. Schulte-Herbr¨uggen 和 S. J. Glaser，《磁共振杂志》172，296（2005）。[12] Z. To˘sner、T. Vosegaard、C. Kehlet、N. Khaneja、S. J. Glaser 和 N. C. Nielsen，《磁共振杂志》197，120（2009）。[13] H. Hogben、M. Krzystyniak、G. Charnock、P. Hore 和 I. Kuprov，《磁共振杂志》208，179（2011）。[14] D. Xu、K. F. King、Y. Zhu、G. C. McKinnon 和 Z.-P. Liang，《医学磁共振》59，547（2008）。[15] M. M. M¨uller、U. G. Poschinger、T. Calarco、S. Montangero 和 F. Schmidt-Kaler，《物理评论 A》92，053423（2015）。[16] V. Nebendahl、H. H¨affner 和 C. F. Roos，《物理评论 A》79，012312（2009）。[17] A. Angerer，《金刚石中氮空位量子比特的鲁棒相干最优控制》，博士论文，维也纳工业大学（2013）。[18] Y. Chou、S.-Y. Huang 和 H.-S. Goan，《物理评论 A》91，052315（2015）。[19] F. Dolde、V. Bergholm、Y. Wang、I. Jakobi、B. Naydenov、S. Pezzagna、J. Meijer、F. Jelezko、P. Neumann、T. Schulte-Herbr¨uggen、J. Biamonte 和 J. Wrachtrup，《自然通讯》5，3371（2014）。[20] F. Poggiali、P. Cappellaro 和 N. Fabbri，《物理评论 X》8，021059（2018）。[21] A. F. L. Poulsen、J. D. Clement、J. L. Webb、R. H. Jensen、L. Troise、K. Berg-Sørensen、A. Huck 和 U. L. Andersen，《物理评论 B》106，014202（2022）。

[22] P. Rembold, N. Oshnik, M. M¨uller, S. Montangero, T. Calarco, and E. Neu,《AVS量子科学》2, 024701 (2020). [23] R.-Y. Yan and Z.-B. Feng,《JETP快报》114, 314 (2021). [24] S. Amri, R. Corgier, D. Sugny, E. M. Rasel, N. Gaaloul, and E. Charron,《科学报告》9, 5346 (2019). [25] S. Amri, R. Corgier, E. Rasel Maria, E. Charron, N. Gaaloul, and Q. Team，载于《美国物理学会原子、分子和光学物理分会会议摘要》，第65卷（2020）。[26] J.-F. Mennemann, D. Matthes, R.-M. Weish¨aupl, and T. Lan- gen,《新物理学杂志》17, 113027 (2015). [27] J. Sørensen, J. Jensen, T. Heinzel, and J. Sherson,《计算物理通讯》243, 135 (2019). [28] J. Guo, X. Feng, P. Yang, Z. Yu, L. Chen, C.-H. Yuan, and W. Zhang,《自然·通讯》10, 148 (2019). [29] S. Rosi, A. Bernard, N. Fabbri, L. Fallani, C. Fort, M. Inguscio, T. Calarco, and S. Montangero,《物理评论A》88, 021601 (2013). [30] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind,《机器学习研究杂志》18, 1 (2018). [31] N. Leung, M. Abdelhafez, J. Koch, and D. Schuster,《物理评论A》95, 042318 (2017). [32] T. Propson, "qoc", https://github.com/ SchusterLab/qoc.git（2019）。[33] M. Abdelhafez, D. I. Schuster, and J. Koch,《物理评论A》99, 052327 (2019). [34] M. H. Goerz, S. C. Carrasco, and V. S. Malinovsky,《Quantum》6, 871 (2022). [35] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, 等，《自然》574, 505 (2019). [36] M. Gong, S. Wang, C. Zha, M.-C. Chen, H.-L. Huang, Y. Wu, Q. Zhu, Y. Zhao, S. Li, S. Guo, 等，《科学》372, 948 (2021). [37] M. Abadi 等，"TensorFlow：异构系统上的大规模机器学习"（2015），软件可从 tensorflow.org 获取。[38] D. Maclaurin, D. Duvenaud, and R. P. Adams，载于 ICML 2015 AutoML 研讨会，第238卷（2015）。[39] C. Moler and C. Van Loan，《工业与应用数学学会评论》45, 3 (2003). [40] B. Codenotti and C. Fassino，《Calcolo》29, 1 (1992). [41] A. H. Al-Mohy and N. J. Higham，《工业与应用数学学会科学计算杂志》33, 488 (2011). [42] D. Goodwin and I. Kuprov，《化学物理学杂志》143, 084113 (2015). [43] I. Najfeld and T. Havel，《应用数学进展》16, 321 1995( ). [44] J. Bechberger，"Temci"，https://github.com/ parttimenerd/temci.git（2015）。

[45] F. Pedregosa，“内存分析器”，https://github.com/pythonprofilers/memory_分析器.git (2012)。[46] J. D. Hunter，《计算机与工程》**9**，90 (2007)。[47] C. R. Harris, K. J. Millman, S. J. Van Der Walt，等，《自然》**585**，357 (2020)。[48] P. Virtanen, R. Gommers, T. E. Oliphant，等，《自然·方法》**17**，261 (2020)。[49] A. Meurer, C. P. Smith, M. Paprocki，等，《PeerJ计算机》科学**3**，e103（2017年）。[50] P. Koch，《Quantum》**5**，583（2021）。[51] N. J.Higham，数值算法的精度与稳定性，第4版（工业与应用数学学会，2002）第2–3章。[52] J. H. Wilkinson，代数运算中的舍入误差，（Courier Corporation，1994）第82页。[53] V. Y. Pan 和 Z. Q. Chen，载于 第三十一届 *ACM* 年度计算理论研讨会论文集，STOC ’99（计算机协会，1999）第507–516页。