基于持久占用编码的图独立集约束保持量子电路编译†

Zhiyi Wang[a,c,*],XiaoruiXue[a,c],Xiaoxing Liu[a,b,c]

[a]东南大学网络空间科学与工程学院，中国南京市江宁区东南大学路2号，邮编211189[b]东南大学经济管理学院，中国南京市江宁区东南大学路2号，邮编211189[c]江苏省高校哲学社会科学实验室——东南大学金融科技与大数据实验室，中国南京市江宁区东南大学路2号，邮编211189

arXiv:2611.29313v1 [cs.CE] 22 Nov 2026

摘要

图独立集的约束保持混合器会反复评估重叠的邻域条件，导致有序量子优化电路中的受控非门成本增加。我们提出一种结构感知编译器，通过持久占用编码复用这些条件。利用真孪生类中的互斥性，我们的可逆编码在混合器操作和代价层期间，将类占用状态存储在原始数据寄存器中。局部混合器采用目标补偿、分块奇偶性查询和相干恢复进行综合；通过精确的系数变换保留非均匀代价相位。我们证明，对于任意指定的操作顺序和独立层参数，编译后的电路在可行子空间上复现原程序的幺正作用，并证明在固定综合方案下，源级缓存成本可精确分离。在134个评估程序和72个独立确认程序中，相较于针对每个程序从六种参考实现中选出的最佳实现，受控非门总数分别减少22.61%和19.87%。与匹配的无缓存实现相比，持久编码进一步分别减少4.10%和9.13%的受控非门数量。

关键词 ： 量子电路编译，约束保持混合器，持久化占用编码 g，可行子空间，相干查询综合

†代码和实验数据已公开，可在此获取：https://github.com/Moonooo24/persistent-occupancy-compiler。Xiaoxing Liu 感谢国家社会科学基金重大项目（项目编号：24&ZD117）的资助。我们感谢 CMU COLOR维护者和贡献者提供基准图，感谢 Qiskit开发者提供电路综合和路由工具，感谢 QuCLEAR作者提供基线实现，并感谢 Cirq开发者提供支持全局相位验证的工具。

*通讯作者。电子邮箱：wangzhiyi924@gmail.com

1. 引言

对于带有硬约束的量子组合优化问题，混合器必须保持可行子空间，使演化始终处于其中。量子交替算子框架使用约束保持混合器，在可行子空间内实现状态转移（Hadfield等人，2019）。例如，在图独立集问题中，顶点处的局部混合器操作受控于其所有邻居均未被占据这一条件，而不同目标顶点的控制条件可能涉及相互重叠的邻居集合。通过计算控制谓词、执行受控旋转并反计算这些谓词，直接实现局部混合会反复评估重叠的邻域条件，导致双量子比特门开销在连续的混合器扫描中不断累积。由于局部混合器操作通常不对易，其执行顺序决定了整体幺正演化。我们研究如何将重叠的邻域信息转化为一种可在多个操作间复用的电路表示，从而在保留指定操作顺序及演化语义的同时，降低完整电路的受控非门开销。

先前研究通过混合算子设计、可行态制备以及在受限子空间中的演化，提出了保持约束的量子优化方法（Farhi等人，2014；Bärtschi和Eidenbenz，2020；Fuchs等人，2022；Fuchs和Bassa，2024年，2025）。离散优化编码和条件旋转的结构化综合也为降低电路成本提供了途径（Sawaya等人，2023；Bergholm等人，2005；Vale等人，2024年）。在由多个局部混合器操作和其间的代价层组成的有序程序中，重叠的邻域条件为跨操作复用创造了机会。核心任务是构造一种占用表示，使其随合法局部跃迁同步更新，从而在有序演化过程中持续保留共享条件，同时确保能够访问原始变量、其合法翻转的语义、代价相位以及相干正确性。

在无向图中，闭邻域相同的顶点属于同一个真孪生等价类。类中的顶点两两相邻，且与外部顶点的邻接关系完全相同（Habib 和 Paul，2010年）。在独立集可行域中，每个类至多只能选取一个顶点，因此类内二元变量的奇偶性恰好表示其占用状态。我们利用这一结构，在数据坐标中编码类占用信息，并通过合法局部跃迁持续更新这些坐标，从而使重叠的邻域条件能够在有序混合器操作和中间代价层之间重复利用。随后，我们通过分块奇偶性查询、相干恢复以及代价系数变换，构造出在可行子空间上保持幺正作用的编译电路。图 1 展示了从真孪生类识别到完整电路选择的编译流程。

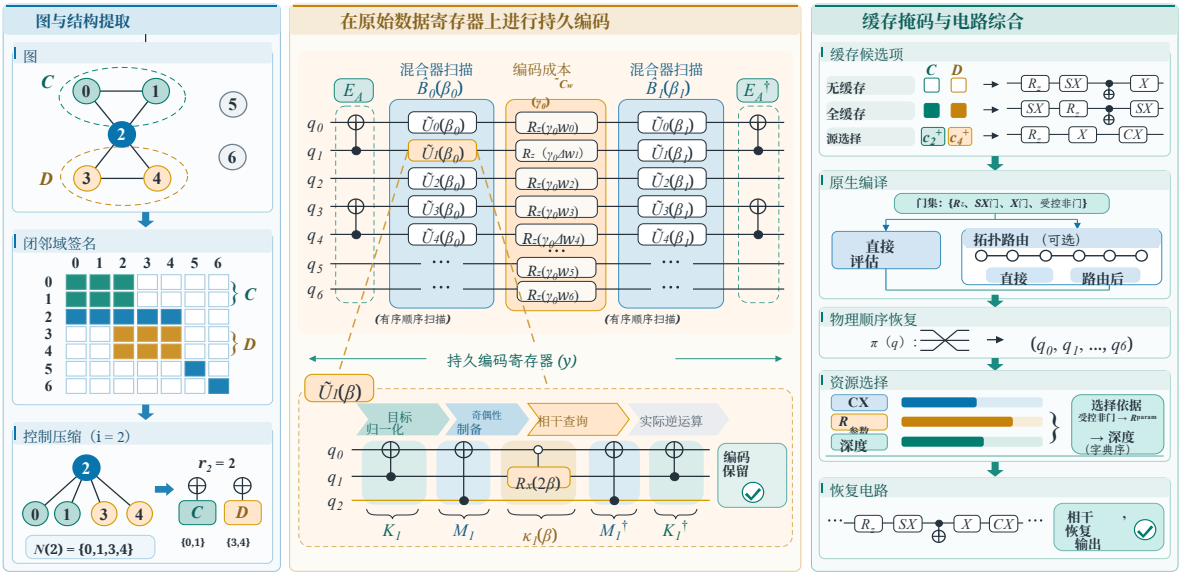

图 1：基于持久占用编码的结构感知编译流程闭邻域签名决定了真正的双射等价类和缓存候选项。在有序混合器操作和代价层期间，原始数据寄存器上的可逆占用编码会一直保留，并在程序末尾解码。局部混合器宏由目标归一化、奇偶条件查询、旋转核和实际反计算组成。编译器根据完整原生电路的资源，从空缓存、全缓存和源级最优缓存方案中选择最终电路，并可选择进行物理路由。

我们的实验采用六种固定参考实现来定义每程序 CX 基准，并使用采用相同查询合成机制的匹配无缓存对照，以量化持久维护占用信息的独立贡献。这些参考实现涵盖结构化受控旋转、开放和闭合 CPG–Walsh综合，以及匹配的全空间精确核实现（Fuchs 和 Bassa，2025年；Vale等人，2024年；Bergholm等人，2005年；Möttönen等人，2005年；Welch 等人，2014年）。与每程序最低参考 CX 数量相比，在包含 62 个不同带标记图和 134 个有序程序的主要评估集中，总 CX 数量减少了 22.61%。在包含 36 个非同构图和 72 个程序的独立确认集中，总 CX 数量减少了 19.87%。两个集合中的每个程序都实现了 CX 数量减少。匹配消融实验还表明，在采用相同查询合成机制的情况下，持久占用编码分别使总 CX 计数进一步减少 4.10% 和 9.13%。采用 SABRE 配置进行物理路由（Li 等人，2019年），并恢复原始线路顺序后，线性拓扑上的总 CX 计数减少了 9.14%，二维网格上的总 CX 计数减少了 12.50%。

本文的主要贡献如下。

1. **针对图独立集可行域的持久占用编码。** 我们将真孪生类中占用的互斥性转化为一种可持续更新的共享数据表示，该表示可以

持续更新。我们在原始数据寄存器上构造可逆编码，并使用目标补偿来保留访问原始变量及合法翻转原始变量的语义。编码表示在有序混合器操作和中间代价层之间持续存在，而精确的系数变换则实现了非均匀权重代价的相位作用。

2. **可相干组合的局部查询与旋转电路。** 我们利用真孪生结构来组织分块奇偶性查询，构造开放式格雷–沃尔什电路和条件相位查询核，并建立统一的相干恢复契约。我们证明，局部混合器宏及其有序组合能够在可行子空间上实现原程序的幺正作用。

3. **源代码级成本可分离性与端到端资源选择。** 我们证明，在固定电路方案下，源级缓存成本关于缓存类别选择严格可分离。我们推导出适用于多次混合器扫描的摊销判据，并将缓存决策归结为逐类边际成本评估。我们通过完整的原生综合、匹配消融实验、独立确认及物理路由，量化受控非门资源节省。

本文其余部分安排如下。第 2 节回顾相关工作；第 3 节阐明编译任务并确立相干正确性条件；第 4 节介绍编译方法；第 5 节介绍实验设置与性能评估；第 6 节作总结。

2. 相关工作

2.1. 约束保持混合器与受限演化

对于具有硬约束的组合优化问题，量子近似优化算法的混合器层必须连接可行解，同时防止振幅泄漏到不可行状态中（Farhi等人，2014）。在量子交替算子框架中，Hadfield等人（2019）构造了用于最大独立集的顶点旋转，其控制条件是所有相邻位均为零。相邻顶点上的局部旋转通常不对易，因此其执行顺序也是混合器层演化定义的一部分。Bärtschi 和 Eidenbenz （2020）采用 Grover 型混合器，将部分实现复杂性转移到可行状态均匀叠加态的制备上。这两种方法分别通过局部可行跃迁和全局可行态制备来实现约束保持混合。

混合器的电路实现也会直接影响受限演化的资源成本。Fuchs等人（2022年）研究了约束保持混合器的构造及其基门分解。Fuchs和Bassa（2024年）利用稳定子形式体系定义了可行子空间上的逻辑 X 混合器，并分析了其受控非门需求。后续工作利用

基变换和投影算符结构构造了受稀疏投影算符约束的演化（Fuchs和Bassa，2025年）。这些研究为降低受限演化的实现成本提供了方法。对于按顺序执行的局部混合器，在不同操作之间复用重叠的邻域条件，可进一步优化资源。

2.2. 图结构编码与条件旋转合成

编码的选择决定了优化变量、约束与量子电路之间的对应关系。 Sawaya 等人(2023年) 系统比较了离散变量编码的表示特性与实现成本。对于图独立集，模分解所捕获的均匀外部邻接关系可用于组织查询共享邻域条件的电路（Habib 和 Paul，2010年）。具有相同闭邻域的真孪生顶点构成一个团，并且与该类之外的顶点具有相同的邻接关系。在独立集可行域上，每个真孪生类中至多有一个被选中的顶点，因此其比特的奇偶性等于其占用量。由于不同的类之间要么完全相邻，要么完全不相邻，局部旋转所需的全零邻域条件可以用块占据信息和目标位来表示，从而将逐顶点的条件查询组织成具有共享结构的逐块查询。

这种结构能否带来电路节省，也取决于条件旋转的合成方式。通用酉分解和等距变换合成提供了通用构造（Shende 等人，2006；Iten 等人，2016）。均匀受控旋转、Gray 顺序和沃尔什展开则针对特定控制结构提供了专门的分解方法（Bergholm 等人，2005；Möttönen 等人，2005；Welch 等人，2014）。多重受控门的分解与资源权衡也已被单独研究（Vale 等人，2024年；Tomesh 等人，2024年）。相对相位托福利门和条件干净工作空间比特提供了进一步降低控制开销的方法（Maslov，2016；Khattar 和 Gidney，2025）。这些研究提供了多种降低条件旋转电路成本的合成方法，而在持久编码下联合优化局部查询与相干恢复，仍有待进一步调查。

2.3. 语义驱动量子编译与端到端资源评估

跨操作的电路优化可以利用量子程序中的状态关系和门序列结构。Häner 等人（2020）借助程序断言所表达的状态性质来指导电路优化，而Hietala 等人（2021）则对 VOQC 中电路变换的正确性进行形式化验证。Amy 和 Lunderville（2025）利用变量之间的线性和非线性关系来优化

相位计算，而Liu 等人（2025）提出的 QuCLEAR 方法通过提取并吸收 Clifford 操作来跨操作优化门。这些方法分别利用程序语义或电路的代数结构消除冗余计算。

通过操作间复用获得的净资源节省，取决于所消除的重复计算以及维护持久表示的成本。持久维护中间表示会为编码、状态更新、坐标变换和查询反计算引入额外开销。后端门抵消和重新综合也可能改变源级实现之间的资源关系。硬件耦合约束会进一步增加电路映射成本。例如，Li 等人（2019）提出了 SABRE，该方法采用启发式搜索来改进量子比特映射和交换操作的选择，从而降低与有限设备连通性相关的实现成本。因此，源级门数分析刻画了局部配方之间可计算的资源关系，而完整综合和物理映射则决定了端到端的实际电路节省。

上述研究为约束保持演化、结构化查询综合、程序语义优化和电路资源评估奠定了基础。我们进一步利用真孪生类中的互斥性，以一种会随着合法转移发生而持续更新的编码，表示局部邻域所需的占据信息。我们通过相干查询和成本坐标变换支持有序混合器程序。随后，源代码级成本可分离性为缓存类别选择提供计算基础，而完整电路综合和物理路由则用于评估通过操作间复用获得的端到端资源节省。

3. 问题表述与相干正确性

本节将有序混合器程序及其编译正确性条件形式化，通过占据编码示例阐明局部操作的编码与恢复，并为查询核建立相干组合定理及相位恢复条件。

3.1. 有序混合器程序与编译目标

设 $G=(V,\ E_G)$ 为一个带有 $|V|=n\geq 1$ 的简单无向图。满足独立集约束的可行比特串集合及其对应的子空间投影算子定义为

$$\mathcal{F}=\{x\in\{0,1\}^n : x_u x_v=0 \text{ for all } uv\in E_G\},\qquad \Pi_{\mathcal{F}}=\sum_{x\in\mathcal{F}}|x\rangle\langle x|. \tag{1}$$

令 $N(i)$ 表示开邻域，并令 $h_i(x) = \prod_{j \in N(i)} (1 - x_j)$。量子占用算符为 $\widehat{x}_j = (I - Z_j)/2$，与之关联的零值邻域控制投影算子定义为$h_i(Z) = \prod_{j \in N(i)} \frac{I+Z_j}{2}$。规范混合器操作和代价算符为

$$U_i(\beta) = \exp[-\mathrm{i}\beta X_i h_i(Z)], \qquad D_w = -\sum_{j=0}^{n-1} w_j \widehat{x}_j, \quad C_w(\gamma) = e^{-\mathrm{i}\gamma D_w}. \tag{2}$$

编译器将输入权重的十进制表示解析为精确有理数 w_j，其中正权重对应于标准加权独立集目标。在所有邻域比特均为零的计算基分支上，U_i 应用 $R_x(2\beta)$。它在所有其他分支上均充当恒等算子。

假设输入程序包含 $p \geq 1$ 个完整混合器扫描，并令 σ_ℓ 为第 ℓ次扫描所用的顶点置换。一次混合器扫描会按规定顺序执行所有顶点的局部混合器操作。其幺正算符为

$$B_\ell = U_{\sigma_\ell(n-1)}(\beta_\ell) \cdots U_{\sigma_\ell(0)}(\beta_\ell). \tag{3}$$

完整程序的幺正算符为

$$W = B_{p-1} C_w(\gamma_{p-2}) B_{p-2} \cdots C_w(\gamma_0) B_0. \tag{4}$$

层参数 β_ℓ 和 γ_ℓ 相互独立。对于 $p = 1$，其间没有代价层。目标$\widehat{\ }$编译电路 W 必须对所有实数参数值满足以下条件。

$$\widehat{W} \Pi_{\mathcal{F}} = W \Pi_{\mathcal{F}}. \tag{5}$$

此条件适用于任何可行输入态；当两侧均与任意外部参考系统上的恒等算子作张量积时，条件仍然成立。编译使用原有的 n 数据量子比特，并在物理路由后恢复逻辑数据原有的物理线路顺序。

3.2. 持久占用编码示例

考虑一个包含七个顶点的冲突图，其中有两个真孪生类 $C = \{0, 1\}$ 和 $D = \{3, 4\}$。每个类中的顶点彼此相邻，且两个类中的每个顶点都与顶点 2 相邻。两个类之间没有边，顶点 5 和 6 是孤立顶点。选择类 C 进行持久占用编码，以顶点 0 为其锚点，并设置 $E =$ 受控非门$(1 \to 0)$。由于可行状态满足 $x_0 x_1 = 0$，编码坐标满足 $y_0 = x_0 \oplus x_1 = x_0 + x_1$和 $y_1 = x_1$。这里，y_0 记录类 C的占用情况，而 y_1

保留顶点 1 的原始变量值。

对于顶点 1 处的局部混合器操作 $U_1(\beta_a)$，局部混合器宏首先应用目标补偿$K_1 =$ 受控非门$(1 \to 0)$，以恢复临时表示中的原始坐标 x_0，x_1 并以顶点 1 为旋转目标查询邻域条件。随后，它应用实际准备电路的精确逆变换，以恢复编码表示。以空集基态 $|0\rangle \equiv |0^n\rangle$ 为输入时，由 $U_1(\beta_a)$ 产生的单顶点占据分支 $|e_1\rangle$ 对应于编码域中的 $E|e_1\rangle = |e_0 \oplus e_1\rangle$。其编码坐标为 $(y_0,\ y_1) = (1,\ 1)$，对应的原始变量为 $(x_0,\ x_1) = (0,\ 1)$。此操作使顶点 1 处的合法翻转与类 C的占用标志更新保持一致。后续局部操作会复用更新后的占据信息。

接下来，考虑依次在顶点 1 执行混合器 $U_1(\beta_a)$、中间的代价层 $C_w(\gamma)$，以及在顶点 2 执行混合器$U_2(\beta_b)$。对于 $w_0 = 3,\ w_1 = 5,\ |e_1\rangle$ 分支的编码成本为 $3y_0+(5-3)y_1 = 5$，因此相位因子为 $e^{5\mathrm{i}\gamma}$。由于顶点 1 和 2 相邻，后续的混合器 $U_2(\beta_b)$ 不会改变该分支，并使空集分支向 $|e_2\rangle$旋转。因此，应用编码、带有中间代价层的两个编译后的混合器操作，然后对空集基态进行解码，可得

$$\begin{aligned} &E^\dagger \widetilde{U}_2(\beta_b)\widetilde{C}_w(\gamma)\widetilde{U}_1(\beta_a)E|0\rangle \\ &\quad = \cos\beta_a \cos\beta_b|0\rangle - \mathrm{i}\cos\beta_a \sin\beta_b|e_2\rangle - \mathrm{i}e^{5\mathrm{i}\gamma}\sin\beta_a|e_1\rangle. \end{aligned} \tag{6}$$

该示例表明，局部混合所诱导的占据更新、代价相位以及后续的邻域查询，都可以在同一个持久编码表示中实现，而末端解码则会恢复原始数据表示。

供后续参考，表1 定义了占据编码、局部查询和源级资源分析中使用的主要符号。

表 1：主要符号

符号	含义
$\mathcal{F}, \Pi_{\mathcal{F}}$	独立集可行域及其投影算子
$\mathcal{C}, \mathcal{A}$	真孪生等价类集合与选定的缓存类集合
$E_{\mathcal{A}}, K_i, T_i$	持久占用编码、目标补偿和 $T_i = K_i E_{\mathcal{A}}$
S_{ib}, f_{ib}, r_i	原始邻域块、临时坐标中的奇偶查询行和控制秩
M_i, κ_i	奇偶准备和完整局部旋转核
$B(\mathcal{A})$ $F(\mathcal{A})$ $J(\mathcal{A})$	一次编码或解码的源级 CX 计数、一次完整混合器的源级 CX 计数
	扫描，以及固定方案下的源级成本目标

矩阵和行向量运算均在 $\mathbb{F}_2$ 上进行。成本中的加权和取自有理数或实数。

3.3. 局部恢复契约与相干组合

为形式化可更新表示，设 E 为一种可逆数据编码，且 $E\,\mathrm{im}\,\Pi_{\mathcal{F}}$ 为编码可行空间。对于目标 i，选择一种可逆准备变换 P_i 和一个完整旋转核 $\kappa_i(\beta)$，其中准备变换可以引入临时数据坐标。局部混合器宏定义为

$$\widetilde{U}_i(\beta) = P_i^{\dagger}\kappa_i(\beta)P_i. \tag{7}$$

定义 3.1（查询与恢复契约）。准备变换 P_iE 将原始可行输入映射到内核的可容许输入域 $\mathcal{Q}_i$。在此域上，κ_i 对目标量子比特实施规定的两能级旋转，同时保留可容许分支之间的复振幅及相位关系。随后应用实际准备电路的逆变换，即可将非目标数据和借用量子比特恢复至各自规定的状态。如果保留残余酉算子 R_i，则 R_i在相关输出子空间上充当恒等算子。启用翻转后进行解码，对应于对原始变量 i执行合法翻转。

~定理 *3.2*（相干组合）。若每个完整宏均满足 $U_i(\beta)E\Pi_{\mathcal{F}} = EU_i(\beta)\Pi_{\mathcal{F}}$，且~编码成本满足 $C_w(\gamma)E\Pi_{\mathcal{F}} = EC_w(\gamma)\Pi_{\mathcal{F}}$，则按照式（*4*）指定的顺序组合这些宏和代价层，并分别在程序开始和结束时应用 E 和 $E^{\dagger}$，即可得到满足式（*5*）中算子恒等式的完整编译电路。此结论适用于任意 p、各层分别指定的执行顺序以及相互独立的实参数。

证明。每个规范混合器和对角代价算符都保持可行子空间不变，因此相应编码宏操作的输出属于 $E\,\mathrm{im}\,\Pi_{\mathcal{F}}$。按照规定的操作顺序代入局部恒等式后，每一对相邻的 $E^{\dagger}E$ 编码与解码变换都会相互抵消为恒等算子，而末尾的 $E^{\dagger}$ 则恢复原始数据表示。这些局部恒等式作为线性算子恒等式在整个可行子空间上成立，因此在与任意参考系统取张量积后仍然成立。

□

3.4. 残余相位与完整旋转核

查询与恢复契约要求，完整混合器宏在编码后的可行子空间上保留指定的幺正作用。对于相邻顶点 u，v，我们有 $\mathsf{CZ}_{uv}\Pi_{\mathcal{F}} = \Pi_{\mathcal{F}}$。因此，如果综合产生的残余对角算子在相关输出子空间上作用为恒等算子，完整宏就可以保留该算子，同时在可行子空间上保持其幺正作用。

对于 r 压缩后的零控制位，将完整旋转核的可容许输入域定义为

$$\mathcal{Q}_i = \mathrm{span}\{|0\rangle_i|z\rangle : z \in \{0,1\}^r\} + \mathrm{span}\{|1\rangle_i|0^r\rangle\}. \tag{8}$$

在可容许输入域 $\mathcal{Q}_i$ 上，当所有控制位均为零时，旋转核施加 $R_x(2\beta)$；当控制串非零时，对于目标位为零的分支，旋转核的作用为恒等算子。具体的门级构造决定了其在其他子空间上的酉扩展。该作用与残余相位恢复条件共同定义了局部查询核的正确性。外围的制备及其实际逆运算将该查询核与持久编码表示连接起来。

4. 通过持久占用编码进行结构感知编译

本节构造真孪生驱动的持久占用编码和邻域条件查询电路，并提供实现非均匀加权代价层以及构造和选择缓存方案的方法。所有相应的局部电路均满足第 3 节的相干恢复契约。

4.1. 闭邻域模块的自动发现

当且仅当 $N[u] = N[v]$ 时，定义 $u \sim v$，其中 $N[u] = N(u) \cup \{u\}$。将每个顶点的闭邻域表示为按顶点标识符排序的列表（签名），并将签名相同的顶点分组。由此得到真孪生等价类划分 $\mathcal{C}$，该划分由闭邻域等价唯一确定。可通过对闭邻域签名进行哈希分组来构造该划分。给定已按顶点标识符排序的签名，并假设哈希操作的期望时间复杂度为常数，分组所需的期望时间与签名的总长度呈线性关系。

引理 4.1（闭邻域等价类的结构性质）。 每个真孪生等价类都构成一个团，而任意两个不同类之间的边要么构成完全二分图边集，要么为空集。

证明。 同一类中的不同顶点 u，v 满足 $u \in N[v]$，因此彼此相邻。若类 C，D 之间存在一条边，则这两个类中的闭邻域相等意味着，对于每个 $u \in C$ 和 $v \in D$，都有 $uv \in E_G$。

□

4.2. 持久占用编码与目标归一化

对于任意选定的非单例类集合 $\mathcal{A} \subseteq \mathcal{C}$，取 $a_C = \min C$ 作为锚点，并定义

$$E_{\mathcal{A}} = \prod_{C \in \mathcal{A}} \prod_{j \in C \setminus \{a_C\}} \mathrm{CX}(j \to a_C). \tag{9}$$

不同的缓存类作用于互不相交的数据量子比特集合，而每个类中的星形 CX 门两两对易，因此 $E_{\mathcal{A}}^{-1} = E_{\mathcal{A}}$。对于目标 $i \in C$，将局部补偿定义为

$$K_i = \begin{cases} I, & C \notin \mathcal{A}, \\ \mathrm{CX}(i \to a_C), & C \in \mathcal{A},\ i \neq a_C, \\ \prod_{j \in C \setminus \{i\}} \mathrm{CX}(j \to i), & C \in \mathcal{A},\ i = a_C. \end{cases} \tag{10}$$

将 $T_i = K_i E_{\mathcal{A}}$ 表示为 $\mathbb{F}_2$ 上的矩阵。

引理 4.2 (目标行列隔离**)。** T_i 的行 i 和列 i 均为单位向量 e_i。因此，临时表示中的目标位仍对应原始变量 x_i，翻转该目标位不会改变其他临时坐标。

对于缓存类中的非锚点目标，局部补偿会暂时移除目标位对类占用锚点的编码贡献，因此局部旋转保留了原始目标变量的翻转语义。撤销局部补偿后，合法翻转会同时更新目标位和类占用锚点。

4.3. 邻域划分、控制秩与奇偶性查询

按锚点递增顺序排列真孪生类。对于目标 $i \in C$，其邻域由其所属类的剩余顶点 $C \setminus \{i\}$ 以及所有相邻真孪生类组成。如果其所属类有剩余顶点，且存在外部相邻类，则将这些顶点与按指定类顺序排列的第一个外部相邻类合并。其他相邻类则构成彼此不相交的独立块 S_{i1}，……，S_{ir_i}。

按块的大小排序，并按顶点的字典序排列，在缓存选择前固定顺序。由于每个真孪生类内部以及任意合并块的不同分量之间的邻接关系均为完全邻接，因此在独立集可行域上，每个邻域块至多包含一个占用顶点。因此，其奇偶性等于其占用量。

令 b_{ib} 为原始邻域块的二进制行掩码。将临时数据坐标中的查询行定义为

$$f_{ib} = b_{ib} T_i^{-1}. \tag{11}$$

对于临时坐标 $y = T_i x$，我们有 $f_{ib} y = b_{ib} x$，因此变换后的查询行保留了原始邻域块的占用测试。在上述划分和目标归一化规则下，所有查询行均非零，其支撑集两两不相交，且都不包含目标位。因此，这些行线性无关，控制秩为 r_i。对于每个查询行，取其支撑集中索引最小的量子比特作为主元，以计算相应的奇偶性并构造变换M_i。完整的局部宏为

$$\widetilde{U}_i = K_i^{\dagger} M_i^{\dagger} \kappa_i(\beta) M_i K_i. \tag{12}$$

令 δ_C 表示商图中类 C 的度。划分规则给出

$$r_i = \begin{cases} \delta_C, & |C| = 1, \\ \max(1, \delta_C), & |C| > 1. \end{cases} \tag{13}$$

控制秩由目标类的商图度数以及该类是否为单例决定。当原始邻域包含较大的真孪生类时，分块查询的控制秩可能低于该邻域中的顶点数，从而减少局部旋转所需的控制条件数量。

4.4. 相干查询综合与自动恢复

本小节构造开放式格雷–沃尔什电路和条件相位查询内核，并证明这两种构造在可行子空间上均满足共同的恢复条件。

局部查询核按控制秩分段选择。对于 $r_i = 0$，使用不带控制位的旋转。对于 $1 \le r_i \le 3$，开放式格雷–沃尔什综合使用 2^{r_i} 参数化旋转，并保留终端相位框架。对于 $4 \le r_i \le 8$，使用固定的相干查询核。在线附录 B.4 节给出了其控制计算前缀和门级方案。对于 $r_i > 8$，可扩展条件相位核使用确定性控制计算及其实际逆运算。该构造选取两条控制线作为条件工作区，在其余控制位上计算一个中间谓词，并将这些操作组合成完整的查询核。结构化控制综合给出了高控制秩查询核的门数上界。在线附录 B.5 节给出了确定性门序列和资源

上界。

对于二进制查询行 f，定义特征 $\chi_f(y) = (-1)^{fy}$，并令 $\bar{h}_i(y) = \prod_b [1 - (f_{ib} y \bmod 2)]$表示编码坐标中的压缩零控制条件。于是

$$\bar{h}_i(y) = 2^{-r_i} \sum_{J \subseteq [r_i]} \chi_{\oplus_{b \in J} f_b}(y). \tag{14}$$

对于 $x \in \mathcal{F}$ 并且 $y = T_i x$，我们有 $h_i(T_i x) = h_i(x)$。按照固定的 Gray 顺序遍历奇偶项，在目标量子比特上合成旋转，并保留终端相位框架（Welch 等人，2014）。在目标量子比特上施加哈达玛门，对与闭合该框架相关的剩余控制变换进行共轭，可得到一个 CZ 星形算符。其控制位对应于邻域块并集中的原始变量。当目标位为 1 时，所有相关的邻域变量均为零，因此该残余相位算符在可行输出空间上等同于恒等算子。在线附录的 B.1 节比较了开放式与闭合式实现所需的资源，并给出了推导过程。

对于控制秩为 $r \geq 4$ 的条件相位查询核，将其幺正算符记为 $\kappa_r(\beta)$。将 r控制位划分为集合 L 和 R，其中 L 包含两个控制位，而 R 包含其余的 $r-2$。双控制子核满足 $B_2(\delta) = R_x(2\delta$或非$_L) D_B$，其中 D_B 是一个参数无关对角对合，在 $L = 0$ 时其作用为恒等算子。将控制寄存器上的计算前缀与其实际逆运算配对，并在二者之间放置一个精确 CCX，得到 $A_r|t,\ z\rangle = |t \oplus q(z)\ z\rangle$，且所有控制位均恢复为其原始值。完整的查询核定义为

$$\kappa_r(\beta) = B_2(\beta/2) H_i A_r H_i B_2(-\beta/2) Z_i, \tag{15}$$

其中 B_2 由一个三 CX 门序列实现，而秩为 4 到 8 的控制计算前缀采用固定方案。该构造满足

$$\kappa_r(\beta) = U_{\mathrm{NOR},r}(\beta) D_r, \qquad D_r = Z_i^{1-q(z)}, \quad D_r^2 = I, \quad [D_r, U_{\mathrm{NOR},r}(\beta)] = 0, \tag{16}$$

此处 q 与目标和参数无关，且 $q =$ 或非$_R$ 当 $L = 0$时。残余算子 D_r在式（8）所规定的可容许输入域上作用为恒等算子。

该核的解析上界为 $6r - 12$ 个 CX 门。在实验中，$4 \leq r \leq 8$ 使用满足恢复契约的预设门级方案，而 $r > 8$ 使用可扩展核。双控制子核 B_2 和控制计算算符 A_r 的门级构造

分别详见在线附录的 B.3 和 B.4 节。残余算子性质和半角配对恒等式的证明见 B.6 节。

这一门级构造还指定了可容许输入域之外的幺正作用，以下称为全空间补全核。依次应用两个参数均为 $\beta/2$ 的补全核，可在整个压缩目标与控制空间上，对任意实数参数精确实现规范旋转：

$$\kappa_r(\beta/2)^2 = U_{\mathrm{NOR},r}(\beta), \qquad r \geq 4. \tag{17}$$

相应电路的 CX 数量解析上界为 $12r - 24$。匹配的全空间精确核基准与我们的方法采用相同的邻域划分、非均匀加权代价函数和目标执行顺序。

定理 4.3 (自动恢复**)**。 对于任意缓存子集，以及上述针对每个控制秩指定的查询核，完整的局部混合器宏满足定理 *3.2*的条件。

证明概要。 $S_i = M_i K_i E_{\mathcal{A}}$ 保持目标行列隔离。经过制备变换S_i后，原始可行输入上的每个主元位都等于其邻域块的占用标志。当$x_i = 1$时，所有邻域块均为空，且制备后的状态属于由式（8）定义的可容许输入域。启用翻转后的实际解码满足 $K_i^{-1} M_i^{-1}(S_i x \oplus e_i) = E_{\mathcal{A}}(x \oplus e_i)$。当控制条件未启用时，相应分支保持不变。将这些等式从可行计算基态线性扩展到整个可行子空间，即得局部算子恒等式。在线附录的 A 节给出了完整证明，以及相关的结构、成本和摊销推导。

□

4.5. 非均匀代价与跨层表示维护

对于 $x \in \mathcal{F}$, 缓存类 C 上的编码位 y 满足 $y_{a_C} = \sum_{j \in C} x_j$ 和 $y_j = x_j$，对于 $j \neq a_C$，因此得到

$$\sum_{j \in C} w_j x_j = w_{a_C} y_{a_C} + \sum_{j \neq a_C} (w_j - w_{a_C}) y_j. \tag{18}$$

对于未缓存类，代价层会暂时在原始坐标中计算团奇偶校验，并在相位操作后应用逆变换，以恢复数据表示。在相同的邻域划分、查询综合规则及其他编译设置下，缓存方案

实现相同的成本函数。它们在电路资源上的差异反映了持久占用表示及其维护操作所需的成本。

对于二元变量 $y \in \{0,1\}$，与相位因子 $e^{\mathrm{i}\gamma c y}$ 对应的单量子比特对角酉算子满足

$$\mathrm{diag}(1, e^{\mathrm{i}\gamma c}) = e^{\mathrm{i}\gamma c/2} R_z(\gamma c). \tag{19}$$

编译器根据编码坐标中的实际代价系数综合每个 R_z 旋转，并累积相应的全局相位。对于均匀权重，每个缓存类的成本相位均由一个参数化旋转实现。对于非均匀权重，则根据式（18）保留非锚点变量的权重差。

中间代价层使用独立参数 γ_ℓ。当某一层中的所有代价系数均为零时，可以省略相应的旋转门。如果综合结果仅包含全局相位，则将该相位记录在电路的全局相位中。每个代价层都在持久编码中实现相位操作，并为后续混合器操作保留该表示，因此完整程序只需在开始时编码一次，并在结束时解码一次。

4.6. 源 级成本分解：多次扫描中的缓存选择与摊销

令 $z_C \in \{0,1\}$ 表示是否缓存类 C。在以下分析中，锚点的选择、邻域划分、奇偶因子顺序、格雷遍历顺序、查询核方案、主元选择规则和初始布局均固定，并为每个源级 CX 门指定一个预设的非负权重。源级成本包括编码与解码边界、所有混合器操作以及中间代价层。

定理 4.4（固定方案下的源级成本可分离性）。目标函数 J 满足上述条件 es

$$J(\mathcal{A}) = J(\varnothing) + \sum_{C \in \mathcal{A}} \big[J(\{C\}) - J(\varnothing) \big]. \tag{20}$$

因此，所有边际成本严格为负的类构成该固定源级目标函数的一个最优缓存子集。边际成本为正的类不进行缓存，而边际成本为零的类则按规定的规则不进行缓存。当在固定布局下按 *CX* 门端点之间的距离计费时，该结果同样成立。

证明概要。缓存某个模块只会改变它对边界编码、目标补偿和奇偶因子的贡献，而不同模块的贡献互不重叠。最低索引的支撑

位置对于外部模块始终是锚点，而目标模块的最低索引支撑位置也保持不变。因此，缓存决策不会改变枢轴选择、查询映射以及其他模块的指令。因此，源级 CX 门的端点重数向量是缓存指示变量的仿射函数 $\{z_C\}_{C\in\mathcal{C}}$。将其与固定的逐门权重相乘并求和，可得式（20）。

□

精确的源代码级成本可分离性将缓存选择简化为针对每个真孪生类分别进行独立的边际成本检验。给定固定的电路方案和固定的逐门权重，评估无缓存方案及每种单类缓存方案，即可确定源级最优缓存子集。随后，完整编译会在通用后端配置下合成源级最优方案、空缓存和全缓存候选项，并根据完整电路资源选择输出，同时考虑跨宏门消除和拓扑路由的实际影响。

多次扫描的摊销与加权源级成本. 考虑在单位 CX 权重下进行多次扫描的摊销。令 $B(\mathcal{A}) = \sum_{C\in\mathcal{A}}(|C|-1)$ 表示一个编码或解码边界处的源级 CX 计数，令 $F(\mathcal{A})$ 和 $H(\mathcal{A})$ 分别表示一次完整混合器扫描和一个代价层的源级 CX 计数。如果所有扫描都采用相同的局部混合器宏配方，且所有代价层都采用相同的加权模式，那么

$$J_p(\mathcal{A}) = 2B(\mathcal{A}) + pF(\mathcal{A}) + (p-1)H(\mathcal{A}). \tag{21}$$

令 $\Delta_F = F(\varnothing) - F(\mathcal{A})$ 和 $\Delta_H = H(\varnothing) - H(\mathcal{A})$。严格收益等价于

$$p\Delta_F + (p-1)\Delta_H > 2B(\mathcal{A}). \tag{22}$$

当各次扫描使用不同的混合器宏配方，或各代价层使用不同的权重配置时，分别计算源级资源并求和。对于一般的固定端点权重 ρ，定义

$$B_\rho = \sum_{C\in\mathcal{A}} \sum_{j\in C\setminus\{a_C\}} \rho(j, a_C). \tag{23}$$

使用相同的端点权重评估 F 和 H，并将 B 替换为 B_ρ，可将摊销判据扩展到加权源级成本。当扫描和代价层带来的累计加权 CX 节省超过以下成本时，所选缓存子集在固定方案下可实现严格的源级收益：

两端的编码与解码。

当商图度数具有固定上界 Δ、团占用成本均匀且查询核库的秩有界时，缓存所有非平凡类的源级电路使用 $O(pn(\Delta+1)+n)$ 个受控非门。每个代价层中，每个类最多使用一个参数化旋转。

4.7. 完整电路选择与原始线路顺序的恢复

根据定理 4.4确定源级最优缓存子集后，编译器会综合三类候选方案：源级最优方案、空缓存和全缓存。重复的缓存方案会合并。每个候选方案均在相同的后端配置下进行完整电路综合。候选方案按照受控非门数量、参数化旋转数量和电路深度的字典序排序；资源指标相同时，则按固定的候选方案顺序打破平局，以选出最终输出电路。所有资源指标均根据该输出电路计算。

对于给定的线性或二维网格拓扑，编译器会在固定初始布局下，根据端点距离生成源级候选方案，并对候选电路进行物理路由。随后，沿设备耦合边插入交换门，以恢复逻辑量子比特的原始线路顺序。恢复过程中的受控非门数量和深度均计入最终资源计数，并保留完整电路的全局相位。

5. 实验设置与性能评估

本节采用固定参考实现、独立确认和匹配消融实验，评估完备电路的 CX 节省及其来源。分析了真孪生类结构、混合器扫描次数、代价权重和物理拓扑的影响。公开图实验和扩展性实验分别考察了不同图结构下的编译性能和经典编译开销。

5.1. 基准程序、基线与通用后端

任务与指标。主要评估集涵盖 $n=32$、64、128 个顶点，以及商图度数为 2、4、6、8 的情况。其中包括所有真孪生类均含两个顶点的模式，以及包含单顶点、双顶点和四顶点类的混合模式。根据带标签顶点和边集对图进行去重，并依据输入图、完整混合器扫描次数、权重配置和目标执行顺序移除重复程序。由此得到包含 62 个不同带标记图和 134 个有序程序的主要评估集。

独立确认集在编译规则确定后单独构建，涵盖 $n = 48$、96、192 个顶点，以及商图度数为 3、5、7 的情况。对保留模块大小属性的商图进行精确同构检查后，得到 36 个非同构图和 72 个有序程序。两个集合均使用两次完整混合器扫描、均匀权重和一个介于其间的代价层，并采用独立参数β_0、β_1、γ_0。每个程序中的两次扫描使用相同的目标执行顺序，每个图则分别按升序和固定伪随机顺序进行评估。在线附录 C.1 节给出了生成与去重规则以及随机种子设置。

除主组和独立确认集外，实验还包括公开基准图、用于正确性检查的固定小型实例，以及扩展实例。主组和确认集由连通正则商图构造而成，并通过将顶点替换为真孪生团来控制模块大小和商图度数。公开图用于评估其他图结构，固定实例用于检查正确性和机制，扩展实例则用于衡量经典编译开销。

主要指标是完整电路的 CX 数量。我们还报告电路深度、参数化R_z 数量、单量子比特门数量和经典编译时间。每个程序的参考 CX 包络是相关固定参考实现中 CX 数量的最低值，配对降幅为 $1-C_{\mathrm{ours}}/C_{\mathrm{ref}}$。多资源比较采用同一个实际电路的完整资源向量。分别针对各项指标得到的参考下界则单独列出。

固定基线. 六种固定参考实现分为三组。第一组包括两种结构化/Vale实现，将Fuchs 和 Bassa (2025年) 的相关结构化构造与 Vale受控旋转分解（Vale等人，2024年）结合，分别采用原始成本和团代价。第二组包括开放式和闭合 CPG–Walsh综合（Bergholm等人，2005；Möttönen等人，2005；Welch 等人，2014）。第三组由两种配对的全空间精确核实现组成，它们采用与我们的方法相同的自动分区前端，分别使用原始成本和团代价。前两组涵盖不同的条件旋转综合方法。采用相同的自动分区前端，使第三组能够在资源比较中控制邻域划分差异。

结构化/Vale实现采用最小主元并进行实际反计算。对于控制秩$r \le 8$，匹配的精确内核使用预先指定的全空间精确核库；对于 $r > 8$，则使用式（17）中的精确半角配对。所有实现使用相同数量的数据量子比特、目标执行顺序和层参数，并在可行子空间上保持相同的成本相位作用。

采用不同成本表示的实现将分别测量，以分析成本表示对电路资源的影响。

其他比较包括对规范 NOR 控制操作的直接分解，以及 Liu 等人 (2025) QuCLEAR 实现中的固定顺序配置和树形配置。每种实现都根据其完整门级方案保留所需的末端恢复和相位操作。六个主要参考实现涵盖主组和独立确认集中的所有程序，公开图实验的三个主要参考实现则涵盖全部 24 个程序。其他参考实现在符合相同实验规范且能够成功完成的程序共同集合上进行比较。在线附录 C.2 节给出了固定参考实现的具体方案。

原生综合与物理路由. 所有实现均使用相同的 Qiskit 2.1.2 后端和原生门集 $\{R_z, \mathrm{SX}\ X、\mathrm{CX}\}$。转换为原生门集后，对所有实现应用相同的、轮数受限的单量子比特门化简和相邻 CX/X 逆操作抵消，同时保留符号全局相位。

物理路由实验针对每个选定程序和拓扑使用三个固定种子。线性拓扑和矩形二维网格均采用恒等初始布局和固定的 SABRE 配置。路由后的资源计数包含终端线路顺序恢复。

5.2. 完整电路资源与独立确认

表2 总结了六种固定参考实现的原生电路资源、每程序参考 CX 包络以及我们的完整输出。在主要评估集的 134 个程序中，CX 总数从 611,250 降至 473,016，减少了 22.61%。在独立确认集的 72 个程序中，CX 总数从 442,568 降至 354,650，减少了 19.87%。在两个集合中的每个程序里，输出的 CX 数量都低于相应的参考包络；独立确认集中的电路深度总计为 344,505。这些结果表明，对于两个集合中的有序程序，结构感知编译均可减少完整电路的 CX 数量；这两个集合在图规模和商图度数方面有所不同。

表 2：主要评估集和独立确认集的完整原生电路资源

实现	主组 受控非门↓	确认集 受控非门↓	确认集 深度↓	确认集 单量子比特↓	确认。参数旋转↓
Structured/Vale / 原始成本	1,039,536	801,648	990,223	1,622,326	72,576
Structured/Vale / 团代价	1,034,528	797,490	990,128	1,618,294	**68,544**
开放CPG	2,927,610	1,487,440	1,270,996	1,003,968	907,200
闭合 CPG	2,998,668	1,545,744	1,327,273	1,003,968	907,200
匹配的精确核 / 原始成本	620,368	446,832	420,307	667,404	180,096
匹配的精确核/团代价	615,360	442,674	420,220	**663,372**	176,064
参考 CX 包络	611,250	442,568	—	—	—
我们的完整输出	**473,016**	**354,650**	**344,505**	690,424	133,056

参考 CX 包络取每个程序的六种固定参考实现中 CX 数量最低者。它不一定对应单个固定参考电路，因此其他资源列留空。单量子比特表示单量子比特门数量。我们方法的所有资源指标均来自同一个最终输出电路，加粗表示对应列中的最低测量值。

表3 在附加参考实现覆盖的共同程序集上，并在 Qiskit 优化级别 1 下，提供了完整的 CX 对比。在所有九种参考实现均成功完成的 12 个含 32 个量子比特的程序中，我们的方法将 CX 总数从 5,780 降至 4,478，减少了 22.53%。在 Qiskit 优化级别 1 下，134 个主要评估程序和 72 个独立确认程序的 CX 总数分别减少了 22.60% 和 19.83%，且每一对程序的 CX 数量均有所减少。在两种转译器优化配置下，CX 减少幅度相近，这表明我们方法的 CX 优势在配置发生变化时仍保持稳定。在线附录的 D.1 节介绍了参考实现的覆盖情况和具体配置。

表 3：完　在共同的额外参考集合上，并采用 Qiskit 优化级别 1 进行的 CX 完整比较

配对集合／配置	程序	参考 CX 门	我们的 CX	CX 门总减少量
九个参考程序的共同集合（32 个量子比特）	12	5,780	4,478	22.53%
1级　，主要评估集	134	611,134	473,016	22.60%
1级，独立确认集	72	442,392	354,650	19.83%

第一行对应于 12 个包含 32 个量子比特的程序，所有九个参考文献均已完成。其余两行对各自的程序集采用 Qiskit 优化级别 1 和每个程序的参考 CX 包络。

图 2 展示了每个程序的 CX 节省量，以及这些节省量在不同图实例、商图度数和目标执行顺序下的分布。主要评估集涵盖偶数商图度数，独立确认集涵盖奇数度数。确认集中的每个程序的 CX 减少量介于 8.96% 至 27.57% 之间，在不同图规模和奇数度设置下均实现了正向 CX 节省。

对于图级统计量，首先针对每个图对不同目标执行顺序下的 CX 减少量取均值，然后对结果按图等权汇总。主要评估集中的 62 个图，其图级减少量的均值、中位数和四分位区间分别为 20.08%、16.01% 和 [14.21%、29.02%], 。独立确认集中的 36 个图对应的统计量分别为 18.92%、21.28% 和 [12.92%、23.37%]。在线附录的 D.2节列出了独立确认集的逐程序资源记录。

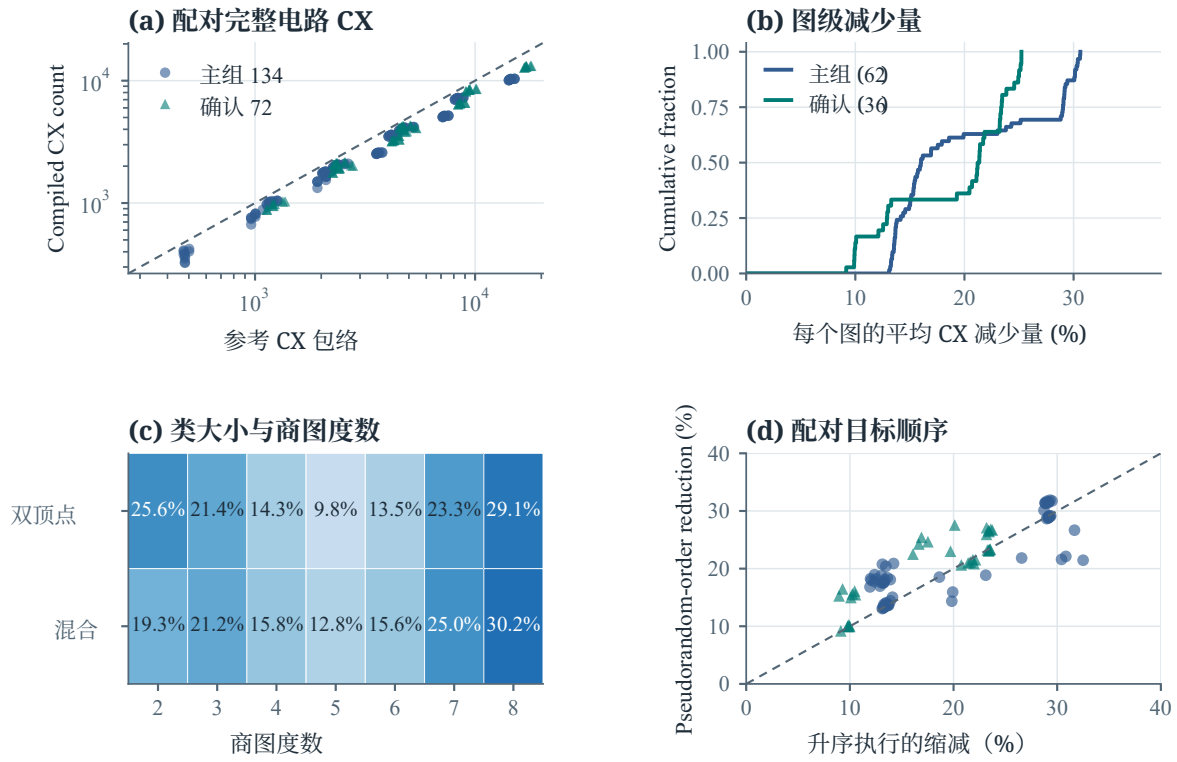

图2：主要评估集和独立确认集中的 CX 计数及配对节省量分布(a) 每程序参考 CX 包络与我们电路的 CX 数量对比。对角线表示两者相等。(b) 每张图中逐程序 CX 缩减均值的经验累积分布，分别涵盖 62 个带标记图和 36 个非同构图。(c) 按商图度数和模块模式汇总的每图缩减均值。两组数据分别涵盖偶数度数和奇数度数。(d) 同一张图上升序和固定伪随机目标顺序的配对结果。

5.3. 编译机制消融实验与图结构的影响

完整编译带来的 CX 节省源于局部条件查询综合以及跨操作的占用复用。表 4 比较了固定的参考 CX 包络、采用相同查询综合机制但不使用持久缓存的电路，以及我们的完整输出，以评估总体 CX 节省量和持久编码的增量贡献。缓存带来的 CX 减少在主要评估集、独立确认集、扫描与权重

评估集以及公共图评估集中分别为 4.10%、9.13%、3.64% 和 0.47%。

表4：完整编译的 CX 节省量及持久缓存的额外贡献

评估集	参考 CX 包络	无缓存 CX	完整输出受控非门	缓存削减	节省占比
主要评估集（134 个程序）	611,250	493,228	473,016	4.10%	14.62%
独立确认（72 程序）	442,568	390,280	354,650	9.13%	40.53%
扫描与权重（120 个程序）	972,132	856,548	825,400	3.64%	21.23%
公开图（24 个程序）	205,334	167,404	166,620	0.47%	2.03%

净缓存减少量是未使用持久缓存的电路与完整输出之间的 CX 数量差，再除以未使用持久缓存的电路的 CX 数量。缓存节省份额是持久缓存带来的额外 CX 节省，再除以参考包络与完整输出之间的 CX 总节省。

图 3 进一步展示了编译机制的消融实验，以及真孪生类结构对 CX 节省的影响。在主要评估集中，闭合和开放的 CPG–Walsh电路的 CX 总数分别为 2,998,668 和 2,927,610。保持终端帧开放可使 CX 数量减少 2.37%。在采用相同的自动分区前端和团代价时，匹配的全空间精确内核基准与未使用持久缓存的相位查询电路的 CX 总数分别为 615,360 和 493,228，后者减少了 19.85%。加入持久编码并执行候选选择后，CX 总数进一步降至 473,016，相较于无缓存方案减少了 4.10%。

为考察真孪生类覆盖率和类大小分布的影响，我们针对图结构设置了两项受控实验，图含 32 个顶点，商图度数为 4，且权重均匀。覆盖率定义为覆盖率 $=\sum_{C\in\mathcal{C},\ |C|>1}|C|/n$。第一项实验调整单例类和双顶点类的比例，使覆盖率分别达到 0%、25%、50%、75% 和 100%。第二项实验将商图固定为 16 个顶点，并调整真孪生类之间的大小分配。12 个图分别与两种目标执行顺序和三种扫描次数 $p=1$，2，4 相结合，共生成 72 个程序。

图 3(d,e) 给出了关于图结构的两项受控实验结果。持久缓存带来的额外 CX 节省测量为 $\big(C_{\text{nocache}}-C_{\text{out}}\big)/C_{\text{matched}}$。在覆盖率为零时，相位查询综合相较于匹配的精确核将 CX 数量减少了 14.29%。在结合单例类与双顶点类的系列中，缓存带来的额外节省达到 1.19%。当额外顶点集中在一个包含 17 个顶点的真孪生类中时，在对应的六个程序中，缓存带来的额外节省介于 39.16% 到 45.01% 之间。在顶点数量和商图度数固定时，受控结果表明，真孪生类大小的分布会显著影响持久缓存的收益。将顶点集中在一个更大的真孪生类中，可能带来更显著的节省

这是通过跨操作复用占据信息实现的。

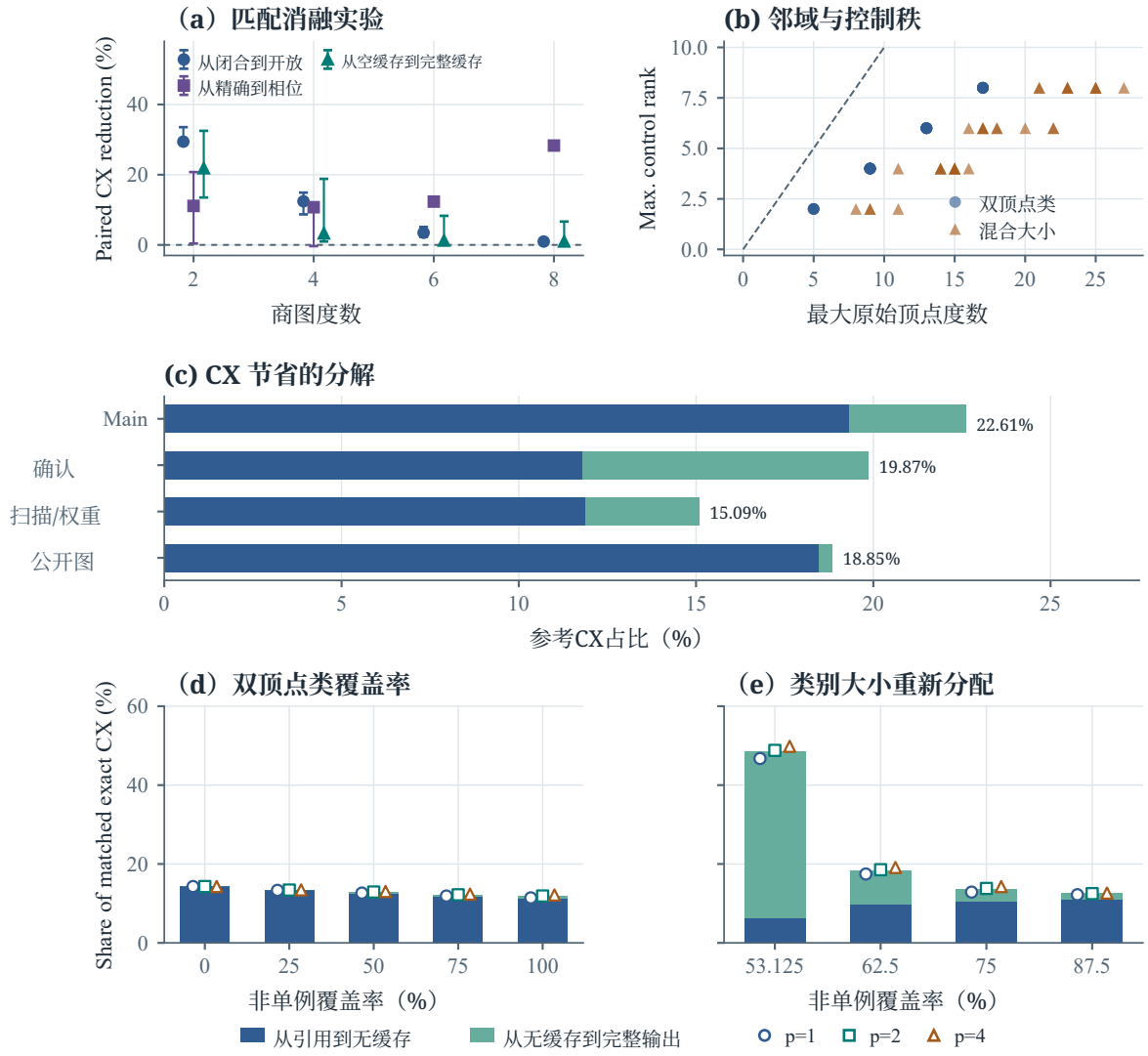

图3：局部查询、持久缓存和真孪生类结构带来的受控非门节省

(a) 针对 134 个主要评估程序的三组匹配消融实验。每项削减均以对应比较中第一种方案的 CX 数量为分母。标记表示中位数，条形表示观测范围。缓存比较采用空缓存和全缓存。(b) 62 个带标记图的最大顶点度和最大控制秩。虚线表示相等。(c) 使用各组固定参考 CX 包络作为分母，对总节省量进行分解。(d,e) 分别改变双顶点类的覆盖率，以及固定商图上类大小的分配。条形高度表示每种结构条件下的平均削减量，分母为各程序匹配的精确核 CX 数量。标记形状用于区分扫描次数；标记表示总计 72 个受控程序中，完整输出相对于匹配的精确核的平均 CX 削减量。

5.4. 多次混合器扫描和非均匀权重下的缓存摊销

为研究持久占用编码在多次混合器扫描中的摊销优势，以及非均匀权重对电路资源的影响，我们使用 12 个固定生成的输入，设置四种扫描次数$p = 1$：2、4、8，以及三种成本模式：均匀、周期性非均匀和两级权重。对于 $p = 1$，中间没有代价层，因此不同权重模式会产生相同的电路。去重后得到 120 个有序程序，各层的成本参数相互独立。实验考察了公式（22）中的源级摊销条件，以及完整后端电路的实际资源。

图 4 展示了不同扫描次数和权重模式下的电路节省与源级摊销结果。相较于三个固定参考实现的逐程序 CX 基准，120 个程序的 CX 总数量从 972,132 降至 825,400，减少了 15.09%，且每个程序的 CX 数量均有所减少。完整电路深度总计减少了 22.12%。对于扫描次数 $p = 1$、2、4、8，CX 减少幅度分别为 12.96%、14.35%、15.05% 和 15.39%。在本实验中，CX 减少幅度随扫描次数增加而提高，这与编码边界成本在多次混合器扫描中的摊销一致。

对于这 36 种图结构—权重配置组合，在固定方案下，源级缓存成本的差异对每种测试的扫描次数都符合式（21）中的仿射关系。对于一个含 128 个顶点且商图度数为 4 的固定混合模块输入，在均匀权重下，源级 CX 节省满足 $J_p(\varnothing) - J_p(\mathcal{C}) = 1088p - 256$。非均匀权重程序中代价层的系数变换已纳入源级摊销模型，而完整电路资源则根据后端综合输出进行测量。图 4(c) 进一步展示了一个六顶点示例，其商图是一个三顶点路径 P_3，且所有真孪生类都包含两个顶点。相较于无缓存，完全缓存的源级 CX 节省为 $4p - 12$。在 $p = 3$ 时，两种方案都使用 90 个源级 CX 门，而后处理后的无缓存电路和全缓存电路分别包含 90 个和 68 个 CX 门。因此，当源级成本持平时，完全缓存实际上可节省 22 个 CX 门，这表明后端门优化可能会改变源级成本模型所预测的资源差异。

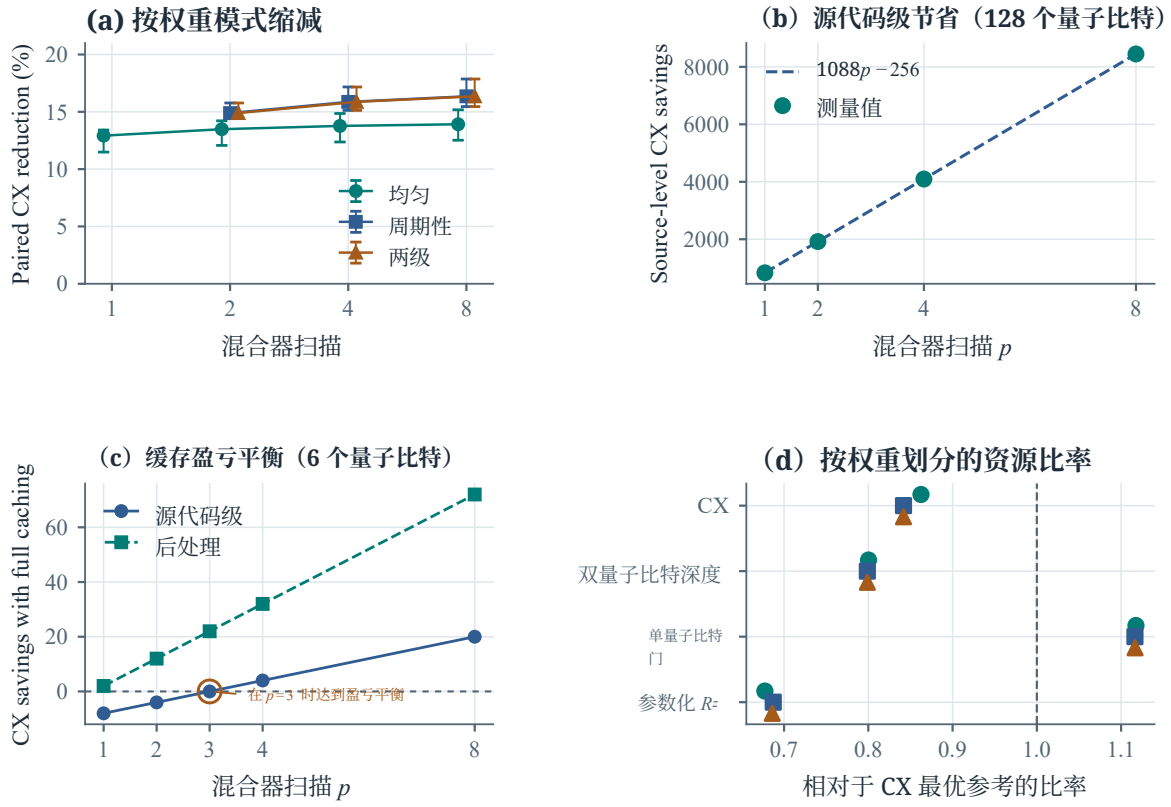

图 4：不同混合器扫描次数和权重模式下的缓存摊销（a）12 个固定输入的平均节省量和观测范围。为区分相同扫描次数下的不同权重模式，各点在水平方向上略作偏移。仅针对 $p \geq 2$ 比较非均匀权重。（b）128 顶点混合模块实例的源代码级 CX 节省及其仿射关系。（c）固定六顶点示例中的源代码级和后处理 CX 节省，其中源代码级成本在 $p = 3$ 时持平。（d）仅针对具有 $p \geq 2$ 的共有条件计算统计数据。每个程序的参考资源向量均取自其对应的 CX 最优参考电路。

5.5. 公共图实例与物理拓扑路由

公共图评估集包含来自 CMU COLOR 图着色基准的 12 个图（Trick，n.d.），其中包括 Mycielski 构造生成的图（Mycielski，1955），以及来自斯坦福图数据库的 Queen、character association 和体育赛程图（Knuth，1993）。我们保留每个原始图的顶点和无向边，将每条边 $\{u, v\}$ 解释为一个独立集约束 $x_u x_v = 0$，并按照第3 节（Hadfield等人，2019）构造约束保持型混合器程序。每个图均采用均匀权重和两次完整混合器扫描，分别考察递增和固定伪随机执行顺序，共得到 24 个有序程序。

对于每个公开图实例，表5 列出了非平凡真孪生类的数量、两种顺序下无缓存电路和完备电路的 CX计数，以及缓存带来的净减少量。四个图包含可通过闭邻域等价识别的非平凡真孪生类。持久缓存带来了

额外的 CX 减少，David、Huck 和 Jean 分别为 1.64%、6.06% 和 1.63%。对于其余九个公开图，完整输出与无缓存方案的 CX 数量相同。此控制实验采用相同的查询合成和成本实现，以评估持久缓存带来的增量贡献。在线附录的 D.3 节提供了各程序的完整资源数据。

表 5：公共图实例上持久缓存带来的额外受控非门节省

图	非平凡真孪生类	无缓存受控非门（A/S）	完整受控非门（A/S）	净缓存减少
david.col	9	6,904/6,874	6,780/6,772	1.64%
games120.col	1	12,326/12,330	12,326/12,330	0.00%
huck.col	11	3,660/3,634	3,432/3,420	6.06%
jean.col	7	3,560/3,568	3,500/3,512	1.63%
myciel3.col	0	226/226	226/226	0.00%
myciel4.col	0	1,152/1,152	1,152/1,152	0.00%
myciel5.col	0	4,536/4,536	4,536/4,536	0.00%
myciel6.col	0	15,840/15,840	15,840/15,840	0.00%
queen5_5.col	0	3,240/3,240	3,240/3,240	0.00%
queen6_6.col	0	6,096/6,096	6,096/6,096	0.00%
queen7_7.col	0	10,248/10,248	10,248/10,248	0.00%
queen8_8.col	0	15,936/15,936	15,936/15,936	0.00%

非平凡类通过闭邻域等价进行识别。A 表示递增顺序，S 表示固定伪随机顺序。无缓存和完整输出使用相同的查询与成本实现。净减少量根据两种顺序下 CX 数量的总和计算。

图 5(a,b) 展示了公开图上的 CX 节省。相较于开放CPG 和两个匹配的精确内核所构成的每程序参考CX包络，24 个程序的 CX 总数量从 205,334 降至 166,620，减少了 18.85%。其中，16 个程序的 CX 数量低于参考值，其余 8 个与参考值持平。纳入其他已完成的参考结果后，每程序参考CX包络保持不变。

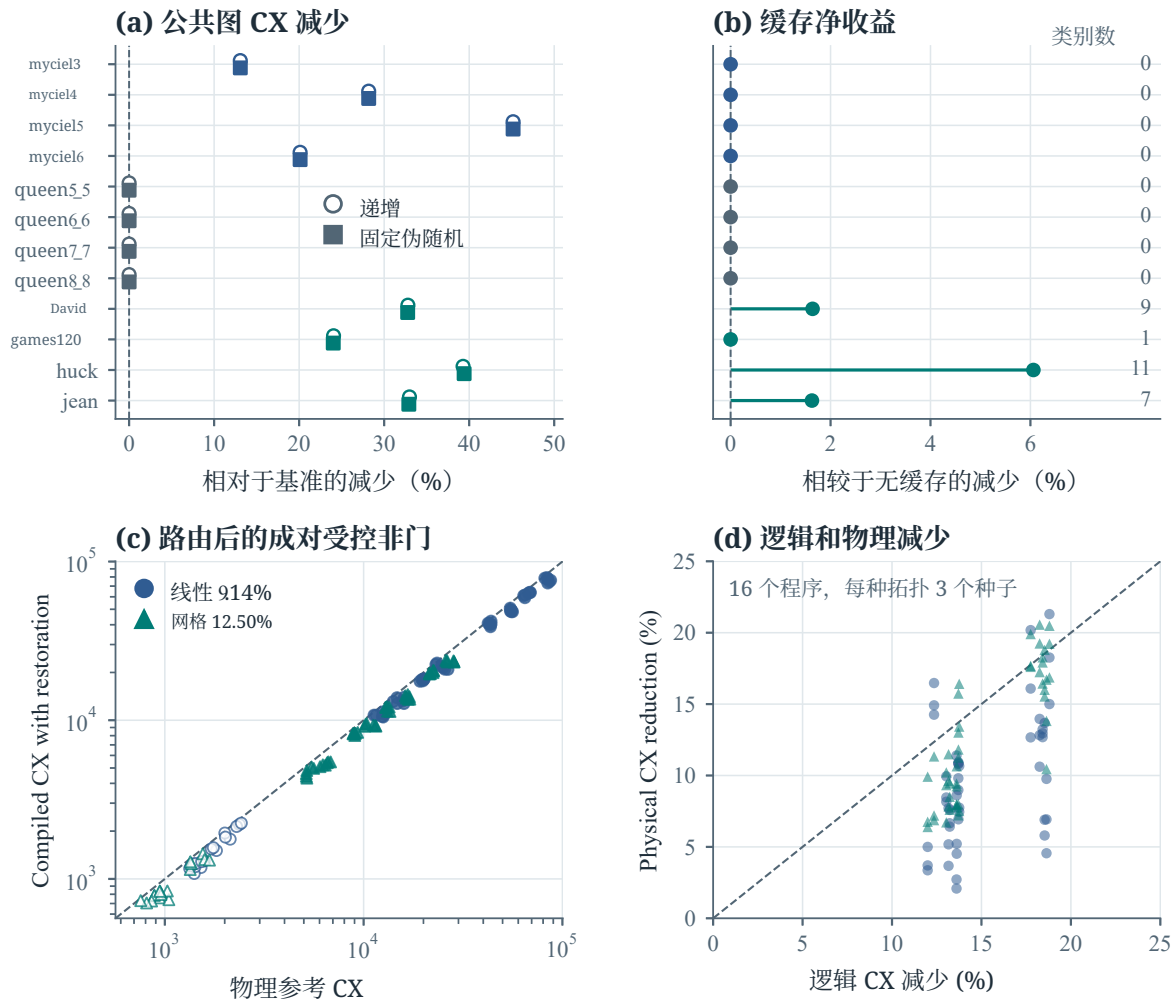

图 5：公开图实验和物理路由实验中的 CX 节省(a,b) 来自 12 个公开图的 24 个程序。(a) 空心圆表示升序，实心方块表示固定伪随机顺序。减少量是相对于每个程序在三种固定参考实现中的最低 CX 数量计算的。(b) 汇总两种顺序的净缓存减少量，以无缓存 CX 计数为分母。右侧显示非平凡真孪生类的数量。(c,d) 物理路由集包含线性和网格拓扑各 66 个程序–种子对。(c) 实心标记表示主要评估子集，空心标记表示用于机制检查的固定图。物理参考值是六种参考实现中的最低 CX 数量。我们的计数包含末端恢复。(d) 16 个主要评估程序的逻辑减少量和每次运行的物理减少量。

Mycielski 图、特征关联图和体育赛程图的 CX 门总减少量分别为 27.40% 和 29.98%。四个 Queen 实例采用与参考实现相同的高秩查询实现，其 CX 数量与参考值持平。对于公开图集合，局部查询综合是整体 CX 减少 18.85% 的主要来源。在采用相同查询综合机制的情况下，持久占用编码进一步将 CX 总数量减少了 0.47%。结合 David、Huck 和 Jean 的结果可见，额外的缓存节省集中出现在一些具有可复用真孪生结构的图中。

物理路由实验包含主要评估集中的 16 个程序，以及由用于机制检查的三个固定图构成的六个程序（每个图采用两种顺序），共计 22 个程序。每个程序在两种拓扑上分别使用三个固定种子运行，因此每种拓扑有 66 个程序–种子对。六个主要参考实现均在所有条件下完成运行。参考 CX 包络取每个匹配的程序、拓扑和种子对应结果中的最小值。在线附录中的表 I-6 分别记录路由输入、最终物理输出和终端线路顺序恢复的 CX 成本。在线附录 C.3 节介绍了路由与恢复协议。

图 5(c,d) 展示了物理路由后的 CX 节省。计入终端线路顺序恢复后，线性拓扑上的 CX 总数从参考值 1,960,198 降至 1,781,104，降幅为 9.14%。二维网格上的 CX 总数则从 684,282 降至 598,736，降幅为 12.50%。两种拓扑下的全部 66 个配对均实现了 CX 节省。对于每个程序，先对三个路由种子对应的配对降幅取平均，再基于 22 个程序计算节省量分布。线性拓扑和二维网格拓扑上的中位降幅分别为 9.54% 和 12.76%，四分位区间分别为 [7.09%，12.83%]和 [8.86%，16.46%], 。

5.6. 多资源权衡与经典编译开销

基于 CX评估，本小节考察 CX优先编译下的多资源权衡与经典开销。图6(a,b)分别展示了每个程序中 CX 与单量子比特门数量之间的权衡，以及每个集合的资源总量变化。比较时，选取每个程序中 CX 数量最少的实际参考电路，并使用其完整资源向量评估 CX优先优化所带来的深度、单量子比特门数量和参数化旋转数量变化。在这一比较下，独立确认集的 CX 总数量减少了 19.87%，总深度减少了 18.04%，参数化旋转数量减少了 24.43%，而单量子比特门数量增加了 4.10%。

对于每个评估集，我们还会针对该集合中的各个固定参考实现，逐个程序分别取每项资源指标的最小值，从而得到该指标的独立参考下包络。按照这一比较方式，独立确认集中的总深度和双量子比特深度分别降低了 18.01% 和 12.82%，而单量子比特门数量和参数化旋转数量分别增加了 7.12% 和 94.12%。在扫描与权重评估集中，后两项指标分别增加了 17.93% 和 86.51%；在公共图评估集中，则分别降低了 6.05% 和 34.28%。CX优先配对参考与独立逐资源下包络采用了不同的比较标准。我们完整输出电路的资源向量在两种比较中相同。

我们按字典序选择候选项，以 CX 数量为主要指标，同时报告所选电路的深度、单量子比特门数量和参数化旋转数量。在独立确认集中，CX优先输出同时减少了 CX 总数量和电路深度，但使用了 690,424 个单量子比特门，比六个固定参考实现中单量子比特门总数量最低的那个多 27,052 个。

对于经典编译开销，表6 按问题规模报告了实际合成的候选项数量、实际运行时间、CPU 时间和峰值驻留内存。图6(c,d) 展示了编译时间和内存开销如何随规模变化。在包含 128 个真孪生类的 256 顶点实例中，源级选择和完全缓存得到相同的子集。去重后，仅合成空缓存和全缓存候选项。完整编译的实际运行时间中位数为 20.04 秒，三个独立 Linux 进程中的最大峰值 RSS 为 195.5 MiB。

表 6：完整原生编译的时间和内存开销

n	真孪生类	合成候选项	墙钟时间中位数（秒）	CPU 时间中位数（秒）	RSS 最大值（MiB）
32	16	2	0.47	0.47	115.0
64	32	2	1.26	1.26	122.9
128	64	2	4.35	4.35	147.6
256	128	2	20.04	20.03	195.5

实验使用商图度数为 4、每个真孪生类包含两个顶点的生成实例。每种规模均在三个单线程 Linux 进程中运行（Intel Xeon Gold 6448Y，Python 3.12.3），最多有 12 个任务同时运行。墙钟时间从导入运行时后开始进行结构发现时起算，到选定最终原生电路时结束。每个进程的峰值常驻集大小（RSS）取自 ru_maxrss，表中报告的是三次测量中的最大值，单位为 MiB。

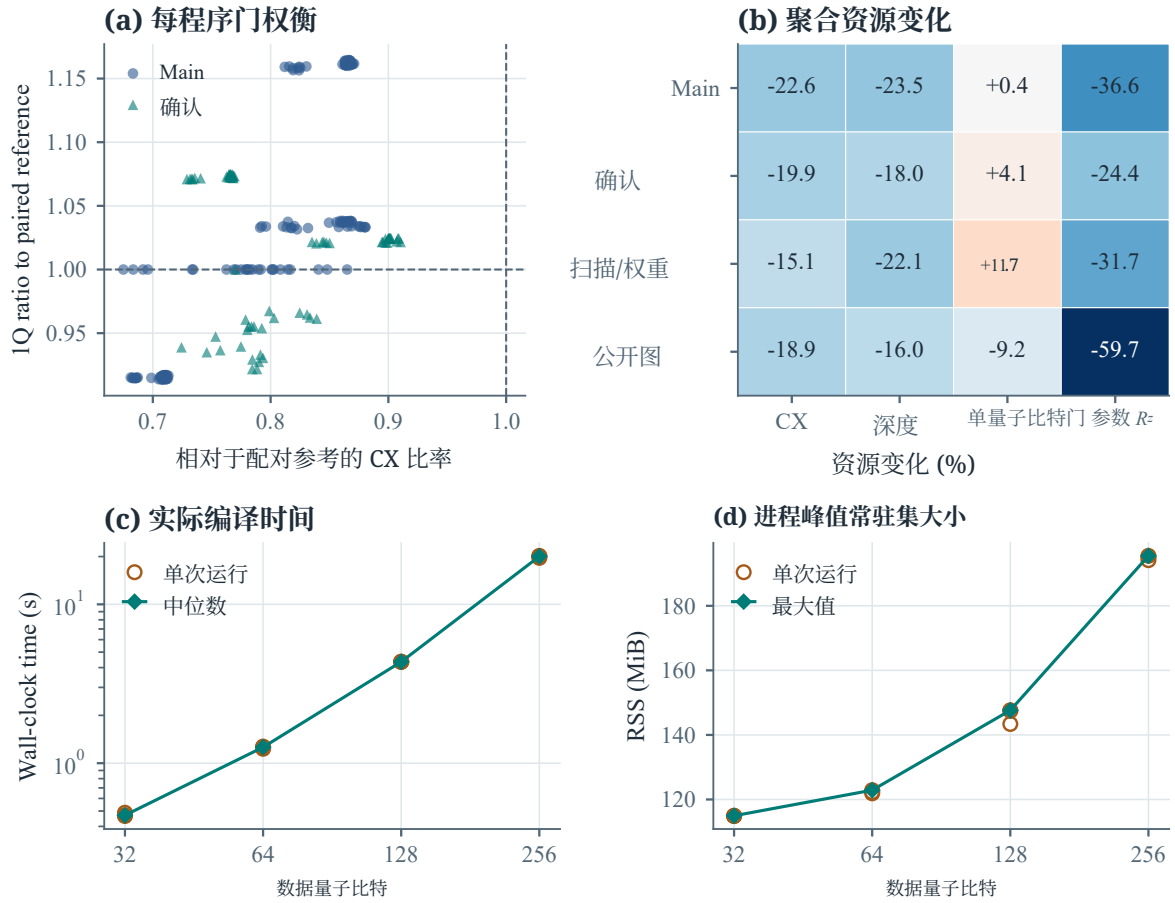

图 6：完备电路的门资源权衡与经典编译开销(a) 主要评估集和独立确认集中的每程序 CX 门数量及单量子比特门数量，相对于配对的 CX 最优参考的比值。虚线位于 1 处。(b) 按集合汇总本方法与相同参考电路的各项资源后计算出的资源变化。负值表示资源减少。(c,d) 扩展性实验中的完整原生编译开销。每种规模均展示了三个独立进程的测量结果。(c) 实线表示实际运行时间的中位数。(d) 实线表示最大峰值 RSS。

为区分真孪生类的数量和每个类的大小对编译开销的影响，我们还将商图固定为$K_{2,\,6}$，其中包含八个模块，将数据量子比特的数量从 16 增加到 256，并分别测量全缓存方案和源级最优缓存方案的单候选编译开销。在线附录中的表 I-7 给出了结果。两条路径的阶段耗时中位数之和分别从 0.26 秒和 0.28 秒增加到 4.82 秒和 6.99 秒。对于 256 顶点实例，采样峰值 RSS 中位数分别为 162.9 MiB 和 164.7 MiB。

5.7. 数值正确性检查

为检查特定门序列及后端变换的实现是否正确，我们在固定的小型程序上比较复振幅，包括全局相位。包含多次扫描的完整程序的最大误差为 2.06×10^{-13}，而经过物理路由后，在选定可行输入上的最大误差为 2.52×10^{-14}。这些检查还涵盖局部查询核、向原生门集的转换以及线路顺序恢复。在线附录 C.6 节给出了测试条件和误差定义。

6. 结论

针对图独立集约束保持混合器程序中重复计算重叠邻域条件的问题，我们提出了一种基于持久占用编码的结构感知编译方法。在独立集可行域上利用真孪生类内占用的互斥性，我们在原始数据寄存器上构造了可逆编码。这使类占用信息能够随合法局部翻转同步更新，并在有序混合器操作及其间的代价层之间持续保持。通过目标补偿、分块奇偶性查询、相干恢复以及对非均匀代价系数的变换，我们证明，完整编译电路在可行子空间上的幺正作用与规范程序相同。在固定电路方案下，源级缓存成本可针对真孪生类的选择精确分解。针对多次混合器扫描的摊销判据刻画了复用节省与编码、解码边界成本之间的关系，为缓存选择提供了可计算的源级依据。

相较于六种固定参考实现中每个程序的最低 CX 数量，我们的方法使主要评估集和独立确认集中的 CX 总数分别减少了 22.61% 和 19.87%，且两个集合中的每个程序都节省了 CX。与采用相同查询综合机制但不使用持久缓存的电路相比，持久编码进一步使 CX 总数分别减少了 4.10% 和 9.13%。针对图结构的受控实验表明，真孪生类大小的分配会影响跨操作的占用复用收益。在测试条件下，将更多顶点集中到较大的真孪生类中，可带来更显著的增量缓存节省。在另一项物理路由实验中，计入原始线路顺序的末端恢复后，线性拓扑上的 CX 总数减少了 9.14%，二维网格上的 CX 总数减少了 12.50%。总体而言，将真孪生类互斥的占用状态转化为一种可在操作间持续

更新的电路表示，可在保留有序程序相干语义的同时，降低完备电路的 CX 成本。

参考文献

M. Amy 和 J. Lunderville。量子程序优化的线性与非线性关系分析。*ACM* 编程语言会议录，9(POPL):1072–1103，2025年。doi:10.1145/3704873。

A. Bärtschi 和 S. Eidenbenz。用于量子近似优化算法的 Grover 混合算子：将复杂性从混合算子设计转移到态制备。载于 *2020*年 *IEEE* 量子计算与工程国际会议（*QCE*），第72–82页，2020年。doi:10.1109/QCE49297.2020.00020。

V. Bergholm、J. J. Vartiainen、M. Möttönen 和 M. M. Salomaa。采用均匀受控单量子比特门的量子电路。*Physical Review A*，71:052330，2005年。doi:10.1103/PhysRevA.71.052330。

E. Farhi、J. Goldstone 和 S. Gutmann。量子近似优化算法。arXiv:1411.4028，2014年。预印本。

F. G. Fuchs、K. O. Lye、H. M. Nilsen、A. J. Stasik 和 G. Sartor。《量子近似优化算法的约束保持混合算子》。《算法》，15(6):202，2022年。doi:10.3390/a15060202。

F. G. Fuchs 和 R.P. Bassa。《量子近似优化算法的 LX 混合算子：受限于子空间和稳定子形式体系的最优混合算子》。*Quantum*，8:1535，2024年。doi:10.22331/q-2024-11-25-1535。

F. G. Fuchs 和 R. P. Bassa。《稀疏算子受约束量子演化的紧凑电路》。arXiv:2504.09133v3，2025年。预印本。

M. Habib 和 C. Paul。《模分解的算法方面综述》。《计算机科学评论》，4(1):41–59，2010年。doi:10.1016/j.cosrev.2010.01.001。

S. Hadfield、Z. Wang、B. O'Gorman、E. G. Rieffel、D. Venturelli 和 R. Biswas。从量子近似优化算法到量子交替算子 Ansatz。《算法》，12(2):34，2019。doi:10.3390/a12020034。

T. Häner、T. Hoefler 和 M.Troyer。量子程序的基于断言的优化。*ACM* 编程语言会议录，4(OOPSLA):133，2020。doi:10.1145/3428201。

K. Hietala、R. Rand、S.-H. Hung、X. Wu 和 M. Hicks。量子电路的经验证优化器。*ACM* 编程语言会议录，5(POPL)，第 37 号文章，2021。doi:10.1145/3434318。

R. Iten、R. Colbeck、I. Kukuljan、J. Home 和 M. Christandl。用于等距变换的量子电路。*PhysicalReview A*，93:032318，2016。doi:10.1103/PhysRevA.93.032318。

T. Khattar 和 C. Gidney。条件清洁辅助比特的兴起：高效量子电路构造。*Quantum*，9:1752，2025。doi:10.22331/q-2025-05-21-1752。

D. E. Knuth。斯坦福图数据库：一个用于组合计算的平台。ACM出版社和 Addison-Wesley，1993。ISBN 0-201-54275-7。

G. Li、Y. Ding 和 Y. Xie。解决 NISQ时代量子设备的量子比特映射问题。载于第*24*届编程语言与操作系统架构支持国际会议（*ASPLOS*）论文集，第1001–1014页，2019。doi:10.1145/3297858.3304023。

J. Liu、A. Gonzales、B. Huang、Z. H. Saleem 和 P. Hovland。QuCLEAR：面向量子电路优化的 Clifford 提取与吸收。载于 *IEEE*高性能计算机体系结构国际研讨会（*HPCA*）论文集，2025。doi:10.1109/HPCA61900.2025.00023。

D. Maslov. 使用相对相位托福利门的优势及其在多控Toffoli优化中的应用。*Physical Review A*，93:022311，2016。doi:10.1103/PhysRevA.93.022311。

M. Möttönen、J. J. Vartiainen、V. Bergholm 和 M. M.Salomaa。使用均匀受控旋转变换量子态。*QuantumInformation&Computation*，5(6):467–473，2005。doi:10.26421/QIC5.6-5。

J. Mycielski。论图的着色。*Colloquium Mathematicum*，3(2):161–162，1955。doi:10.4064/cm-3-2-161-162。

N. P. D. Sawaya、A. T. Schmitz 和 S. Hadfield。离散优化量子算法中的编码权衡与设计工具包：着色、路由、调度及其他问题。*Quantum*，7:1111，2023。doi:10.22331/q-2023-09-14-1111。

V. V. Shende、S. S. Bullock 和 I. L. Markov。量子逻辑电路的综合。*IEEE* 集成电路与系统计算机辅助设计汇刊，25(6):1000–1010，2006。doi:10.1109/TCAD.2005.855930。

T. Tomesh、N.Allen、D.Dilley 和 Z. H. Saleem。《变分算法中的量子-经典权衡与多控制量子门分解》。*Quantum*，8:1493，2024 年。 doi:10.22331/q-2024-10-04-1493。

M. A. Trick。《图着色实例》。卡内基梅隆大学。
https://mat.tepper.cmu.edu/COLOR/instances.html。

R. Vale、T. M. D. Azevedo、I. C. S. Araújo、I. F. Araujo 和 A. J. da Silva。《多重控制特殊单量子比特门的电路分解》。 *IEEE*《集成电路与系统计算机辅助设计汇刊》，43(3):802–811，2024年。
doi:10.1109/TCAD.2023.3327102。

J. Welch、D. Greenbaum、S. Mostame 和 A. Aspuru-Guzik。《无需辅助量子比特的对角幺正算子高效量子电路》。《新物理学杂志》，16:033040，2014年。 doi:10.1088/1367-2630/16/3/033040。

数据可用性

这 12 个公共图实例取自 CMU COLOR基准集合（Trick，不详），可通过以下网址获取
https://mat.tepper.cmu.edu/COLOR/instances.html。

编译器源代码、基准数据、实验配置、电路资源测量数据以及用于复现结果的脚本均已公开发布于
https://github.com/Moonooo24/persistent-occupancy-compiler。

致谢

Xiaoxing Liu 感谢国家社会科学基金重大项目（项目编号：24&ZD117）的资助。我们感谢 CMU COLOR维护者和实例贡献者提供基准图（https://mat.tepper.cmu.edu/COLOR/instances.html），感谢 Qiskit贡献者提供电路综合和路由工具（https://github.com/Qiskit/qiskit），感谢 QuCLEAR作者提供其基线实现（https://github.com/revilooliver/QuCLEAR），并感谢 Cirq开发者提供用于全局相位检查的稳定子代码（https://github.com/quantumlib/Cirq）。

作者贡献

Zhiyi Wang：概念化、方法论、软件、形式分析、调查及初稿撰写。Xiaorui Xue：方法论、验证、可视化及审阅与编辑。Xiaoxing Liu：概念化、监督、项目管理及审阅与编辑。所有作者均审阅并批准了最终稿件。

利益冲突

作者声明不存在利益冲突。

声明　　在稿件制备过程中使用生成式人工智能和人工智能辅助技术

在准备本手稿期间，作者使用 ChatGPT（OpenAI）润色语言，并协助识别潜在的算法错误。作者审阅并核实了生成的内容，并对本作品的内容承担全部责任。

在线附录

附录 A. 结构编码与相干组合的证明 附录 B. 相干查询核的门级构造与证明 附录 C. 实验、实现协议与验证 附录 D. 补充实验结果与逐程序资源 表 I-1. 固定秩查询核的控制计算门序列 表 I-2. 高秩相位查询核与全空间精确配对核的资源 表 I-3. 参考实现、消融实验与我们的方法的查询核方案 表 I-4. 独立确认集的逐程序资源 表 I-5. 两种目标顺序下公开图的资源 表 I-6. 物理路由与线路顺序恢复的 CX 资源 表 I-7. 两条单候选路径的时间和内存开销

附录 A。证明　　结构编码与相干恢　　复

本附录给出了持久占用编码、可逆目标补偿和相干恢复所依据的结构性质的完整证明。此外，还推导了正文第 3 节和第 4 节所使用的成本层系数变换、源级缓存可分离性以及多轮扫描摊销准则。

A.1。闭邻域等价类与分块控制秩

闭邻域相等在顶点集上定义了一个等价关系。对于同一类中的不同顶点u，$v \in C$，$u \in N[u] = N[v]$，因此 $uv \in E_G$，且每个类都是一个团。假设一条边 uv连接两个不同的类 C，D，其中 $u \in C$且 $v \in D$。对于每个 $u' \in C$，都有 $v \in N[u'] = N[u]$。再对每个 $v' \in D$ 使用$N[v'] = N[v]$ ，即可得出对于所有 $u' \in C$ 和 $v' \in D$，都有 $u'v' \in E_G$ 。因此，不同类之间的边要么构成一个完全二分图边集，要么为空集。

对于目标 $i \in C$，原邻域是不相交并集，由 $O = C \setminus \{i\}$ 和所有外部相邻类组成。当 O 非空时，O 是一个团，并且与每个外部相邻类完全相邻，因此 O可以与一个固定的外部相邻类合并为一个团块。如果 $|C| = 1$，那么 $r_i = \delta_C$。如果$|C| > 1$ 且 $\delta_C > 0$，合并后仍有 δ_C 个块。如果 $\delta_C = 0$ 且 $|C| > 1$，目标类的剩余顶点构成一个块。这三种情况证明了正文中关于目标 i的控制秩的式（13）。

闭邻域等价确保了这些划分和占用性质。作为对比，三顶点路径的两个端点具有相同的开邻域，但可以同时被选中，因此它们的奇偶性无法准确表明该顶点集是否被占用。一般的极大团覆盖也可能包含彼此重叠的团，而且这些团之间的邻接关系未必是完全的。此处采用的闭邻域等价划分使每个类内部互斥，并确保不同类之间要么完全邻接，要么完全不邻接。

A.2. 可逆占用编码与目标行列隔离

Let x 是 $\mathbb{F}_2$上的原始数据列向量。缓存类 C 的编码行是

$$(E_{\mathcal{A}}x)_{a_C} = \bigoplus_{j\in C} x_j, \qquad (E_{\mathcal{A}}x)_j = x_j \quad (j \neq a_C). \tag{A.1}$$

如果目标 i 是缓存类中的非锚点顶点， K_i 会从锚点行中移除 x_i 。目标本身具有单位行，且没有其他行包含 x_i，因此列 i 也是单位列。如果目标是锚点，K_i 会完全撤销该类的星形 CX 编码。如果目标类未缓存，则其编码和补偿均为恒等变换。其他类的支撑集与目标类互不相交，因此不会改变上述结论。

临时数据坐标满足 $y = T_i x$，查询条件满足 $fy = bx$。因此，查询行由可逆变换的逆变换确定，即 $f = bT_i^{-1}$ 。

每个原始邻域块在临时坐标中的支撑集都可以分解为以下类分量。整个未缓存外部类的支撑集 D 保持为 D，而整个缓存外部类的支撑集为 $\{a_D\}$。对于未缓存的目标类，剩余顶点的集合 O 保持不变。对于目标为非锚点的缓存类，其支撑集为 $\{a_C\}$；对于目标为锚点的缓存类，其支撑集保持为 O。合并操作仅取两个先前互不相交的类分量的并集。因此，所有奇偶查询行的非零支撑集均非空、两两互不相交，且不包含目标位。这些支撑集互不相交且非空，因此查询矩阵的秩等于邻域块的数量。

A.3. 合法翻转后的相干恢复与程序复合

令 $S_i = M_i K_i E_{\mathcal{A}}$。其目标行和列仍为单位向量 e_i。对于原始可行状态，每个真孪生类的占用量至多为 1，因此每个主元位都等于对应邻域块的占用标志。当 $x_i = 1$时，所有邻域块均为空，且输入属于正文式（8）定义的可容许输入域。所有主元均为零，当且仅当规范邻域 NOR 条件成立。

当邻域零控制条件不成立时，查询核在对应的可接受分支上执行恒等算子，且完整宏输出 $E_{\mathcal{A}}x$。当该条件成立时，查询核在临时坐标中的作用为

$$\cos\beta|S_i x\rangle - \mathrm{i}\sin\beta|S_i x \oplus e_i\rangle. \tag{A.2}$$

由 $S_i^{-1}e_i = e_i$ 和 $S_i^{-1} = E_{\mathcal{A}}^{-1}K_i^{-1}M_i^{-1}$ ，可得

$$K_i^{-1}M_i^{-1}e_i = E_{\mathcal{A}}e_i. \tag{A.3}$$

通过实际制备变换的逆变换恢复数据坐标后，所得状态为

$$\cos\beta|E_{\mathcal{A}}x\rangle - \mathrm{i}\sin\beta|E_{\mathcal{A}}(x \oplus e_i)\rangle. \tag{A.4}$$

两个原始态都满足独立集约束，且系数与规范的$R_x(2\beta)$旋转一致。从可行计算基态线性延拓，可得到整个可行子空间上的算子恒等式；与任意参考系统进行张量积后，该恒等式仍然成立。对于缓存类中的非锚点目标， $E_{\mathcal{A}}e_i$ 包括同时更新目标位和锚点位，而局部逆变换使目标翻转与占用表示保持一致。

A.4. 非均匀代价与全局相位一致性

对于缓存类 C，同一真孪生类中至多有一个原始变量为 1，因此在二元域上的奇偶和等于相应的整数和。因此 $y_a = \sum_{j\in C} x_j$ 且 $y_j = x_j$ 对于 $j \neq a$。由此可得 $x_a = y_a - \sum_{j\in C\setminus\{a\}} y_j$。将其代入原始成本表达式 $\sum_{j\in C} w_j x_j$，得到正文中的式（18）

。对于未缓存类，代价层会暂时计算团占用奇偶性，并在应用相位操作后通过实际逆运算恢复原始数据坐标。

若实际表示中的各项为 $c_s\ell_s(y)$，其中 ℓ_s 是单比特变量或奇偶函数，则每一项的全局相位因子为 $e^{\mathrm{i}\gamma c_s/2}$，总因子为 $\exp(\mathrm{i}\gamma\sum_s c_s/2)$。不同表示的系数和可能与原始权重之和不同。例如，对于权重为 (2，-3，1) 的三顶点团，原始权重之和为 0。若选择权重为 2 的顶点作为锚点，则表示系数为 (2，-5，-1)，其和为 -4。因此，完整电路中必须包含相应的全局相位因子 $e^{-2\mathrm{i}\gamma}$。

编译器以稀疏仿射形式记录全局相位的符号系数，提取每个独立参数对应的常数系数，并根据原始表达式重建门角度和全局相位。从精确有理数转换为浮点数会引入数值表示误差。因此，通过独立的复振幅比较，可以检查完整电路在指定参数下的数值一致性。

A.5. 缓存子集可分离性与多次混合器扫描摊销

在固定的源级电路方案下，整个外部类组件中索引最小的物理支撑位置始终是其锚点。对目标模块进行补偿也会使该块中索引最小的支撑位置保持不变。合并块的主元是两个固定最小值中较小的那个，因此控制线排序和查询核的物理映射均不受其他模块缓存状态的影响。

每个可变CX门的端点重数都可归属于唯一的缓存类。边界编码和目标补偿仅取决于各自所属的类，而非锚点控制线上的奇偶校验准备指令则按类独立变化。开放格雷综合中的控制位也遵循相同规则。不同类所属的数据线彼此不相交，因此源级CX端点重数向量是缓存类指示变量的仿射函数。将任意预先指定的非负门权重应用于该向量，即得到正文中的式(20)。

程序边界处的编码和解码各需要 $B(\mathcal{A})$ 个源级 CX 门。p 混合器扫描和 $p-1$ 中间代价层的源级成本分别为 $pF(\mathcal{A})$ 和 $(p-1)H(\mathcal{A})$，由此得到正文中的式（21）。若 $\Delta_F+\Delta_H>0$，满足严格节省判据的最小正整数扫描次数为

$$p_{\min}=\max\left\{1,\left\lfloor\frac{2B(\mathcal{A})+\Delta_H}{\Delta_F+\Delta_H}\right\rfloor+1\right\}. \tag{A.5}$$

当 $\Delta_F+\Delta_H\le 0$ 时，直接使用正文中的式（22）计算每种扫描次数对应的净节省量。若不存在中间代价层，该判据可简化为 $p\Delta_F>2B$。此推导适用于固定的源级电路方案。由后端门抵消和物理路由导致的成本变化，则在完整的候选电路上进行评估。

附录 B. 相干查询核的门级构造与证明

本附录给出正文第 4.4 节所述条件查询核的门级构造及正确性证明。表I-1 列出了固定秩控制计算序列（秩 4–8），表 I-2 报告了更高控制秩下的资源界和编译后资源。本附录给出开放式格雷–沃尔什电路和条件查询核的门级定义、残余相位证明及解析资源界。主目标的索引为 0，按序排列的控制位索引为 1，…，r。门序列按时间顺序从左向右书写和执行。定义 $R_x(\theta)=e^{-\mathrm{i}\theta X/2}$，$R_z(\theta)=e^{-\mathrm{i}\theta Z/2}$，以及 $T=\mathrm{diag}(1,e^{\mathrm{i}\pi/4})$。

B.1. 开放式格雷–沃尔什电路与终端框架相位

对于互不相交的查询行 f_1、……、f_r，所有异或组合均不相同，因此正文中的式子 (14) 包含2^r 项。令 $g(k)=k\oplus\lfloor k/2\rfloor$, $k=0$、……、2^r-1，并按固定的因子顺序计算 $m_k=\bigoplus_{b:\ g(k)_b=1} f_b$。从 $m_0=0$ 开始，对每个 $k\ge 1$，按照量子比特顺序递增，在

$m_{k-1} \oplus m_k$ 的非零位置应用 CX 门，并将相应的控制位用作控制位、目标位用作目标位。这样会更新累积奇偶性。每个旋转项都通过目标位 Hadamard 变换和$R_z(2\beta/2^r)$ 实现。闭合版本会在末尾撤销当前二进制掩码 f_* 对应的控制变换，而开放版本则保留此终端帧。目标位经过哈达玛共轭后，两者之间的相位差算子为

$$R_* = \prod_{j \in \mathrm{supp}\, f_*} \mathrm{CZ}_{ij}. \tag{B.1}$$

$f_* T_i$ 对应若干原始邻域块的并集，其非零位置位于 $N(i)$ 中。当目标为零时，上述所有 CZ 门均作用为恒等算子。当目标为一时，独立集可行性会强制所有相关邻域位均为零。因此，在完整混合器宏的可行输出空间上，残余算子为恒等算子。此性质在宏输出边界处得到验证，而内部临时数据坐标则由相应的可逆变换确定。

B.2. 三 *CX* 相对相位 *Toffoli* 原语

令 $\tau(a,\ b;\ c)$ 表示以下门序列：

$$[H_c, T_c, \mathrm{CX}(b,c), T_c^\dagger, \mathrm{CX}(a,c), T_c, \mathrm{CX}(b,c), T_c^\dagger, H_c]. \tag{B.2}$$

对于固定的 $(a,\ b)$，当 $a = 0$ 时，该门在目标线上施加 I；当 $(a,\ b) = (1,\ 0)$ 时施加 Z；当 $(a,\ b) = (1,\ 1)$ 时施加 Y。因此，其计算基置换与 CCX 一致，尽管基态相位可能不同，而且它使用 3 个 CX 门。以下控制计算前缀仅作用于控制寄存器，随后执行其实际逆变换。中间对主目标位的翻转使用一个精确 CCX，其固定分解包含 6 个 CX 门。对于控制位的每个计算基态，相位都会通过计算前缀及其实际逆变换精确抵消。

B.3. 双控制相位查询子核 B_2

给定控制位 $u,\ v$ 和目标 t，定义

$$\begin{aligned} B_2(\delta) = \mathrm{seq}[&H_t, R_{z,t}(\delta/2), \mathrm{CX}(u,t), R_{z,t}(\delta/2), \\ &\mathrm{CX}(v,t), R_{z,t}(\delta/2), \mathrm{CX}(u,t), R_{z,t}(\delta/2), H_t]. \end{aligned} \tag{B.3}$$

完整局部空间上的恒等算子为

$$B_2(\delta) = U_{\mathrm{NOR},2}(\delta)\mathrm{CZ}_{tv} = R_{x,t}(2\delta\, \mathrm{NOR}_{\{u,v\}}) Z_t^v. \tag{B.4}$$

在末尾添加控制变换 CX$(v,\ t)$ 即可闭合相位框架。随后，四个权重相等的奇偶旋转组合成零控制投影算子的沃尔什展开，对应于完整的 NOR 控制旋转。在目标哈达玛共轭之后，终端帧为 CZ_{tv}。对于 $u = v = 0$，该算子是恒等算子，因此与 NOR 控制旋转对易。该门序列包含 3 个 CX 门和 4 个参数化旋转。

这里使用的三 CX 子核也可以通过等价的两步因式分解推导得到。首先，设单控制核为

$$B_1(\delta) = \mathrm{seq}[H_t, R_z(\delta), \mathrm{CX}(u,t), R_z(\delta), H_t]. \tag{B.5}$$

按 $u = 0,\ 1$ 划分得到 $B_1 = U_{\mathrm{NOR},\ 1}(\delta) Z_t^u$。使用 X_t^{1-v}作为精确翻转，构造以下乘积 $B_1(\delta/2) Z_t^{1-v} B_1(-\delta/2) Z_t$。两个剩余对合 Z_t^u相互抵消，所得的完整幺正作用与上述双控制子核一致。

B.4. 秩 *4* 至秩 *8* 查询核的门级构造

在整个过程中使用 $L = \{1,2\}$ 和 $R = \{3,\ldots,r\}$。令 $F_R = \prod_{j \in R} X_j$ 为开放控制翻转。表 I-1给出了计算前缀 V_r 以及精确 CCX 的两个控制位 $a_r,\ b_r$ 定义如下：

$$A_r = F_R V_r^\dagger \mathrm{CCX}(a_r, b_r; 0) V_r F_R, \tag{B.6}$$

矩阵从右向左作用时，完整的查询核为

$$\kappa_r(\beta) = B_2(\beta/2)H_0A_rH_0B_2(-\beta/2)Z_0. \tag{B.7}$$

双控制子核可写为 $U_{\mathrm{NOR},2}(\delta)D_L$，其中 $D_L = Z_0^{f(L)}$，$f(L) \in \{0,1\}$，$f(0) = 0$，且 D_L 是参数无关的。该对合与条件查询兼容，并根据上述代数关系在复合核中抵消。评估中的资源计数对应于实验前确定的门级实现。控制计算前缀仅作用于控制寄存器，而 B_2 始终作用于主目标以及原始控制位 1 和 2。

令 $q_r(z)$ 表示完整翻转 A_r（无残余相位）的布尔控制条件。对于 $r = 4,5,6$，相应的计算前缀分别构成与计算链，并分别以 0、1 和 2 条可条件使用的控制线作为工作空间。它们满足 $q_r = \mathrm{NOR}_R$，当 $L = 0$ 时。当 $L \neq 0$ 时，剩余控制扇区中 q_r 的取值由特定补全幺正算子的计算前缀决定。该前缀的实际逆运算可恢复所有控制位。

对于 $r = 7$，将 R 经过开放控制转换后的各位记为 a、b、c、d、e。在 $L = 0$ 上，计算前缀之后两个中间控制位的布尔值分别为 ab 和 $1 \oplus a \oplus e$（$1 \oplus b \oplus cd$）。如果 $ab = 0$，则 $q_7 = 0$。如果 $ab = 1$，后者为 ecd，因此 $q_7 = abcde$。对于 $r = 8$，将开放控制转换后的各位记为 a、b、c、d、e、f。第二个中间控制位为 $1 \oplus a \oplus (1 \oplus d \oplus ef)$（$1 \oplus b \oplus cd$）。当 $ab = 1$ 时，它化简为 $cd(1 \oplus d \oplus ef) = cdef$，因此 $q_8 = abcdef$。这证明所有列出的前缀都满足 $L = 0 \Rightarrow q_r = \mathrm{NOR}_R$。

代入方程（B.4）和（B.7）可得 $\kappa_r(\beta) = U_{\mathrm{NOR},\,r}(\beta)Z_0^{1-q_r}$。剩余对合在目标位为零的分支以及所有控制位均为零的分支上作用为恒等算子。它满足正文中定义的查询与恢复契约，并保证将两个半角补全核配对后在全空间上精确。表中列出的 CX 数量是相对相位计算前缀及其实际逆变换、一个精确 CCX 和两个三 CX 子核的数量之和。解析计数为 $6r - 12$。

对于控制秩 $r > 8$，保留含有 $|L| = 2$ 的因式分解，并使用 KG 条件清洁综合（Khattar 和 Gidney，2025）生成控制计算前缀。该构造满足相干恢复条件，其 CX 资源界随控制秩线性增长。

表 I-1：固定秩查询核的控制计算门序列

r	V_r的门序列	$(a_r,\ b_r)$	CX 数量
4	$\varnothing$	(3, 4)	12
5	$\tau_{34;1}$	(1, 5)	18
6	$\tau_{34;1};\ \tau_{15;2}$	(2, 6)	24
7	$\tau_{34;1};\ \tau_{56;4};\ X_4;\ \tau_{74;3};\ X_3$	(1, 3)	30
8	$\tau_{34;1};\ \tau_{56;4};\ X_4;\ \tau_{78;6};\ X_6;$ $\tau_{64;3};\ X_3$	(1, 3)	36

$\tau_{ab;c}$ 表示式 r 在式（B.2）中定义的相对相位托福利门。分号表示按执行顺序串接各门，而 $\varnothing$ 表示空序列。构 造出的查询核仅作用于原始目标线和 r 控制线。

B.5. 高秩条件查询核的确定性实现

对于 $r \geq 9$，计算前缀使用条件洁净多重控制综合原语`synth_mcx_1_clean_kg24`（位于 Qiskit 2.1.2 中）。此处 $m = r - 2$，该原语的输入线顺序由以下部分构成：m正控制线、一条目标线，以及一条在指定扇区内满足洁净性前提条件的工作空间线。对于按顺序排列的控制位 1、……、r，以下确定性线映射和门序列提取可得到式（B.6）中的 V_r，并据此构造相位核。

1. 取 L =(1、2) 和 R =(3、……、r)。按顺序将 SDK 电路的线路索引映射为 (R、0、1)，其中 0 是主目标，原控制位 1 是工作空间。不添加单独的辅助数据量子比特。2. 找到门序列中唯一的精确 ccx。将其执行顺序之前的所有门定义为 V_r。验证每个门的类型为 x、rccx 或 rccx_dg，且映射后没有任何门作用于主目标。读取精确 ccx 映射后的两个控制位，并验证其目标为 0。

3. 执行 F_R、V_r、CCX $V_r^\dagger$、F_R。通过对前缀电路取实际逆运算，并按相反顺序执行逆门，得到 $V_r^\dagger$。根据公式 (B.7)，将此翻转与两次调用 B_2 组合。对于全空间精确核，按照正文中的公式 (17)，依次应用两个补全核，每个核使用一半的目标角度。

在 $L = 0$ 子空间上，映射到原始控制位 1 的工作空间满足 KG原语的条件清洁性前提，其对应的翻转条件为NOR_R。在其他控制扇区中，计算前缀是控制计算基上的带相位置换。将精确 CCX 与该前缀的实际逆运算结合，可得到不含依赖于控制基的残余相位的条件置换，并恢复所有控制位。两个双控制子内核和目标补偿决定了补全幺正算子的残余相位。其可容许输入域由正文中的公式（8）规定。该前缀至多包含 $m-2$ 个三 CX 相对相位托福利门。计入其实际逆运算、六 CX 精确 CCX 以及两个三 CX 子核，可得构造上界$6(m-2)+6+6 = 6r-12$。

B.6. 残余相位对合、精确半角配对与资源界

对于 $r \geq 4$的条件相位查询核，本节推导其全空间残余相位及解析的 CX 资源界。将控制位 N 划分为固定的一组 L 双控制线和 $R = N \setminus L$，并令 $s(z)$ =或非$_L(z)$。对于 $r = 4$，$|R| = 2$，将 R 上的控制位转换为开放控制后，直接应用精确 CCX。对于 $5 \leq r \leq 8$，使用表 I-1所指定的固定计算前缀。对于$r \geq 9$，使用 KG 单清洁辅助位综合方案（one-clean）构造计算前缀，并将其条件清洁工作空间线映射到 L中的原始控制线 1。条件清洁原语的洁净性前提条件仅施加于 $L = 0$ 子空间。

令 $\mathcal{C}$ 包含 R 上的开放控制 X 翻转以及该计算前缀。$\mathcal{C}$ 仅作用于控制数据，并且是计算基上的带相位置换。中间应用一个精确 CCX，随后执行 $\mathcal{C}$的实际逆运算。因此，对于任意初始控制值，$\mathcal{C}^\dagger$CCX$\mathcal{C}$ 中的计算相位都会精确抵消，从而得到 $A|t,\ z\rangle = |t \oplus q(z)\ z\rangle$。此处，$q$是一个与主目标位及旋转参数无关的布尔函数。实际逆运算相干地恢复所有控制数据，而上述构造保证了所需的条件

$$s(z) = 1 \implies q(z) = \mathrm{NOR}_R(z), \qquad q(0^r) = 1. \tag{B.8}$$

当 $L \neq 0$, q 由实际完备幺正算符的控制位计算确定时 ry.

固定的双控制子核必须在整个局部目标–控制位空间上满足完备幺正条件

$$B(\delta) = R_{x,i}(2\delta s) D_B, \qquad D_B = Z_i^{f(L)}, \quad f(L) \in \{0,1\}, \quad f(0^2) = 0. \tag{B.9}$$

同一个子核对于参数的正值和负值具有相同且与参数无关的 D_B。它与 $Z_i^{q(z)}$ 以及 $R_{x,\ i}(\alpha s)$对易，因为后者在 $s = 0$时是恒等算子，而 $D_B = I$ 在$s = 1$时则是恒等算子。

矩阵按从右向左的顺序作用时，原始构造为

$$\begin{aligned} \kappa_r(\beta) &= B(\beta/2) Z_i^{q(z)} B(-\beta/2) Z_i && \text{(B.10)} \\ &= R_{x,i}(\beta s) Z_i^{q(z)} R_{x,i}(-\beta s) Z_i && \text{(B.11)} \\ &= R_{x,i}(2\beta s q) Z_i^{1-q(z)}. && \text{(B.12)} \end{aligned}$$

由于 $sq = \mathrm{NOR}_L\ \mathrm{NOR}_R = \mathrm{NOR}_N$，令 $D_r = Z_i^{1-q(z)}$ 即可得到正文中的公式（16）。D_r 是一个与参数无关的对角对合，在 NOR 启用扇区上作用为恒等算子，因此它与$U_{\mathrm{NOR},\ r}(\beta)$对易。它在零目标分支上也作用为恒等算子。因此，残余算子的对合性质和对易性质可推出以下恒等式，该恒等式对目标和控制寄存器上的每个实数 β 均成立：

$$\kappa_r(\beta/2)^2 = U_{\mathrm{NOR},r}(\beta/2)^2 D_r^2 = U_{\mathrm{NOR},r}(\beta) \tag{B.13}$$

该构造仅使用目标量子比特和原始控制量子比特，原始控制线提供所需的条件工作区。令 $k = |R| = r - 2 \geq 2$。计算前缀及其实际逆运算总共至多使用 $2(k-2)$ 个三 CX 相对相位托福利门。精确 CCX 使用 6 个 CX 门，两个固定的双控制子核各使用 3 个 CX 门。当 $k = 2$ 时，计算前缀为空。因此，CX 解析上界为

$$3 \cdot 2(k-2) + 6 + 2 \cdot 3 = 6r - 12, \qquad \mathrm{CX}_{\text{exact pair}} \leq 12r - 24. \tag{B.14}$$

该解析界适用于 $r \geq 4$ 的条件相位构造。实验使用这一可扩展实现来处理 $r > 8$。低秩基元使用各自预先规定的方案，并满足相同的相干恢复契约。评估分别报告解析界、实测编译计数和拓扑路由成本。

表I-2比较了相位查询内核规范与全空间精确内核规范下同一构造的资源成本。秩 9 和秩 10 还分别进行了独立的复振幅验证。秩 12 和秩 16 报告了相应门级构造的实测编译资源。

表 I-2：高秩相位查询核和全空间精确配对核的资源

r	CX 解析界	原生 CX	后处理 CX	参数旋转
9	42/84	42/84	42/84	8/16
10	48/96	48/96	48/96	8/16
12	60/120	60/120	60/120	8/16
16	84/168	84/168	84/168	8/16

每对 a/b 分别表示一个单相位核，以及由两个半角相位核依次组成的精确核。表中报告了 CX 解析上界、原生 CX 计数、后处理 CX 计数和参数化旋转计数。

附录 C. 实验、实现协议与验证

本附录记录了正文第5节中使用的图和程序生成规则、随机种子、固定参考实现、拓扑路由、软件环境以及数值验证。表 I-3 汇总了参考电路、消融实验和所提方法的查询内核方案。本附录介绍图和程序的生成协议、固定参考电路方案、实现环境以及正确性验证流程。

C.1. 数据组、实例生成与随机种子

主要评估集的实例生成涵盖不同的数据规模、规则的商图度数以及真孪生类大小模式。在指定的生成配置下，这些因素共同决定了查询控制秩的范围。按带标签顶点和无向边集对图去重后，得到 62 个图。再按图、混合器扫描次数、权重配置和实际目标置换去重后，从 144 条原始程序记录中得到 134 个程序。独立确认集采用单独的规模配置和随机生成流。通过基于哈希的预筛选，以及对保留模块大小属性的商图进行精确同构检查，得到 36 个非同构图和 72 个程序。全部 36 个图均在首次生成尝试时满足连通性和同构筛选条件。独立确认集的样本接纳仅依据连通性检查，以及保留模块大小属性的商图同构检查。

公开图均记录了其原始数据、来源 URL、文件头和规范化无向边集。反向重复的边视为同一无向约束。固定小图用于对特定电路进行数值一致性检查，扩展性测试图则用于测量各阶段耗时和内存占用。

主要评估集和确认集中的输入图，均通过将规则商图的顶点替换为真孪生团构造而成。每个类内的顶点按连续标识符分配。每个类都是一个团，相邻类之间则完全连通。主要评估集的初始种子为 $202611090000 + 10000n + 100d + 30I_{\text{mixed}} + j$，其中 d 为商图度数，$j = 0$、1、2表示

生成重复次数。生成器会依次尝试递增的种子，直到生成满足连通性要求的图。确认集的候选种子为 $2026111000000+1000003i+15485863a$，其中 i 表示固定生成顺序的索引，a 为重试索引。接受与否取决于连通性检查以及保留模块大小的精确同构筛选。对于这两个集合，固定伪随机目标顺序的种子均为相应图生成种子加701。规范输入记录存储实际图、顺序和重试索引。

受控结构实验中的图和有序程序在资源计数之前便已固定。第一组实验使用的五个覆盖率级别分别对应以下（单元素类数量，双顶点类数量）：（32，0）、（24，4）、（16，8）、（8，12）和（0，16）。每个中间级别有两种固定布局，每个端点级别有一种，共计 8 个图。第二组实验固定商图顶点数为 16，并采用四种类大小分布：15 个单元素类和一个 17 顶点类、12 个单元素类和四个五顶点类、8 个单元素类和八个三顶点类，以及 4 个单元素类、8 个双顶点类和 4 个三顶点类。它们的覆盖率分别为 53.125%、62.5%、75% 和 87.5%。每个商图中的扩展类按照固定伪随机排列进行布局，布局种子为 $202611110 + j$，其中 j 是布局标识符。固定伪随机目标顺序的种子为 202611117。所有输入顶点的权重均为 1，同一图上的所有扫描均使用相同顺序，且 $p - 1$ 成本层参数彼此独立。两组实验共包含 12 个图和 72 个程序。为进行资源测量，对相同的缓存子集进行去重。

C.2. 参考实现与固定配方

表 I-3 列出了先前构造的实现、固定的全空间精确参考实现，以及比较方法共用的高秩查询实现。在主要评估集和独立确认集中，控制秩均不超过 8。两项完整比较均使用在实验前确定的有限秩配方。

具体配方按控制秩选定。对于 $r = 1,2$，封闭式沃尔什综合分别使用 2 个和 4 个 CX 门。对于 $r = 3$，相位无关成对查询构造和双控制部分子核使用 8 个 CX 门。对于 $r = 4$，将相对相位中间托福利门与精确双控制子核配对，使用 14 个 CX 门。对于 $r = 5,6$，成对 RC3X 构造在 $|L| = 2,3$ 时分别使用 20 个和 28 个 CX 门。对于 $r = 7,8$，固定 Iten 精确翻转分别与 $|L| = 4,5$ 和 $|R| = 3$ 对应的部分子核配对，分别使用 40 个和 52 个 CX 门。这些配方由预先指定的分块配对和门级原语定义。每个精确宏都在整个局部空间上实现指定的 NOR 控制旋转。原始成本和团代价分别作为独立的参考实现进行测量，以明确比较配置。

匹配的精确配对也可以由以下算子恒等式构造。令 $E_L(\theta)$ 表示在 L 上具有零控制位的全空间精确旋转，并令 $\mathcal{Z}_L(\theta) = H_i E_L(\theta) H_i$。令 A_R 为在 R 上具有正控制位的单项式翻转。它可以带有对角相位，但必须恢复所有其他数据。其计算基置换给出 $A_R^\dagger Z_i A_R = (-1)^{\prod_{j\in R} x_j} Z_i$。这些对角相位在此共轭下相互抵消。使用 F_R 将 R 中的控制位转换为零控制位。配对构造为

$$E_{L,R}(\beta) = F_R H_i \mathcal{Z}_L(\beta/2) A_R^\dagger \mathcal{Z}_L(-\beta/2) A_R H_i F_R. \tag{C.1}$$

将两个指数算子合并，得到 $e^{-\mathrm{i}\beta Z_i \,\mathrm{NOR}_L \,\mathrm{NOR}^R}$。随后对目标进行哈达玛共轭，即可得到完整的 NOR 控制旋转。当 A_R 是不带额外相位的精确翻转时，也可以使用一个部分子核，其中包含 $C_L = U_L D_L$、$D_L = Z_i^{f(L)}$ 和 $f(0) = 0$。共轭后的剩余项为 $X_i^{f(L)}$，它与 A_R 对易，并在两个子核之间相互抵消。对于 $r = 3,7,8$，表格采用不带额外相位的精确翻转，并搭配指定的部分子核。其余高秩精确核则结合相对相位翻转与精确子核。

表 I-3：参考实现、消融配置和我们的方法的查询核方案

实现或配置 tion	$r \leq 8$	$r > 8$
Structured/Vale	最小主元 CPG、固定 Vale 旋转， 以及实际逆运算	相同的 CPG　采用固定的无辅助量子比特精确 双查询实现（使用 Qiskit）
闭合 CPG	按目标 Gray 顺序进行完整 Walsh 合成 采用闭合框架	奇偶准备，固定的无辅助量子比特精确双- 可查询，且实际逆运算
开放CPG（包括完整的 Walsh 消融实验）	按目标 Gray 顺序进行 Open Walsh 合成	共享的高秩条件相位查询核- nel
匹配的精确核	固定的全空间精确查询核库	两个半角相位核的精确配对 采用相同构造
我们的方法	Open Walsh for $r \leq 3$ and Table I-1 核 用于 $r \geq 4$	高秩条件相位查询核

所有实现使用相同数量的数据量子比特，并保留相干正确性所需的末端操作。

C.3. 拓扑路由与原始线路顺序恢复

线性拓扑由相邻顶点之间的边 $(i, i+1)$ 构成。对于二维网格，令 a 为不大于 $\sqrt{n}$ 的图规模最大正因数，并构造一个 $a \times (n/a)$ 矩形耦合网格。设备耦合边支持双向 CX 门。初始布局为恒等映射，路由完成后，编译器后端会提供从逻辑量子比特到物理量子比特的置换。为恢复逻辑数据原本的物理线路顺序，我们在耦合图上选取确定性最短路径。沿路径交换相邻位置，并在必要时执行反向交换门，从而交换两端的位置，同时使内部逻辑量子比特回到端点交换前的位置。每次交换后都会更新逻辑到物理映射，所有交换完成后则恢复恒等置换。

每个 SWAP 门都分解为三个遵循设备连通性的 CX 门。最终电路会进行相同数量上限的局部化简轮次，并检查双量子比特门是否满足物理边约束，以及是否恢复原始逻辑线路顺序。三个路由种子分别为 20261109、20261110 和 20261111。物理评估包含 22 个有序程序。其中 16 个来自主要评估集中的双顶点类和混合类实例，使用 $n \in \{32, 64\}$ 和 $d \in \{4, 6\}$，并采用两种目标顺序。其余 6 个来自三个固定机制图各自对应的两种顺序：一个 14 顶点的 H7 图；一个 12 顶点的图，该图通过将二部商图 $K_{2,\,4}$ 中的每个顶点替换为一个双顶点团而得到；以及一个 14 顶点的图，该图通过对 $K_{2,\,5}$采用相同的替换过程而得到。复现数据记录了完整输入和图标识符。路由后还会检查数值一致性。

C.4. 实现环境

主要运行环境使用 Python 3.12.3、Qiskit 2.1.2、NumPy 1.26.4、SymPy 1.14.0 和 mpmath 1.3.0。在独立进程中进行计时，并排除运行时导入开销。正文表 6 中的 Linux 内存指标是进程的峰值常驻集大小（RSS），取自`ru_maxrss`，并以 MiB 为单位报告。

C.5. 输入条件与结构记录

输入程序包括一个顶点标记为 0、……、$n-1$ 的简单无向图（没有自环或重边）、$p \geq 1$ 次混合器扫描、每次扫描对应的一个完整顶点置换，以及解析为精确有理数的十进制权重。编译记录存储闭邻域划分、商图、临时目标变换及其逆变换、邻域块和查询行，以便重建特定电路。相同的构造规则适用于单元素类、孤立顶点和非连通图。比较结果会区分无效输入、不受支持的实现或基本操作，以及不完整的资源计算。资源比较包括在程序规范一致的情况下实际成功完成的电路。

C.6. 数值正确性与实现一致性检查

我们逐列比较局部核的完整矩阵，包括全局相位；比较范围涵盖相对相位托福利门基元、双控制子核，以及秩为 4 到 8 的固定核。我们还独立检查秩为 9 和秩为 10 的相位查询核及其精确配对核。对于小型完整程序，我们使用独立的层参数值，将规范有序演化与编译电路进行比较。测试角度包括不同正负号和大小的零值与非零值。数值检查涵盖原始构造、原生门集转换和有界后处理。

对于 h实际 输出态 ψ^{actual} 和规范参考态 $\psi^{\mathrm{reference}}$，我们使用复数 - 振幅误差 包括全局相位

$$\epsilon_\infty = \max_j \left|\psi_j^{\mathrm{actual}} - \psi_j^{\mathrm{reference}}\right|. \tag{C.2}$$

秩-9 和秩-10 局部核的最大误差为 4.10×10^{-14}，而对于包含多次扫描的完整程序，最大误差为 2.06×10^{-13}。对路由后的固定可行输入进行相同比较，最大误差为 2.52×10^{-14}。相对于路由输入电路进行全空间回归时，最大误差为 9.67×10^{-17}。

附录 D。补充实验结果与逐程序资源

本附录为正文第5节提供详细的资源记录和补充实验比较。表 I-4 列出了全部72个独立确认程序；表 I-5 报告了12个公开图上的两个目标顺序；表 I-6分别列出了物理路由和终端恢复成本；表 I-7报告了经典计算时间和内存开销。本附录报告了其他参考文献的完成覆盖率、独立确认集和公开图的逐程序资源、物理路由详情，以及固定模块数量下的单候选编译开销。

D.1. 额外参考覆盖率与后端配置

在 48 个使用 32 个量子比特的程序中，QuCLEAR 的固定顺序配置和树形配置分别完成了 48 个和 36 个程序，直接规范 NOR 分解完成了 12 个程序。六个主要参考实现和三个额外参考实现均在 12 个程序上完成。在 22 个物理路由程序中，QuCLEAR 的固定顺序配置和树形配置分别完成了 17 个和 16 个程序。六个主要参考实现与这两个额外参考实现的交集包含 48 个程序–种子对。额外参考实现的 CX 数量仅在匹配规范下均完成的共享程序集上进行比较。

后端敏感性检查使用 Qiskit 2.1.2 和optimization_level=1、全连接逻辑后端以及固定随机种子 20261101，在相同配置下编译源级最优方案、空缓存候选项、全缓存候选项和六个固定参考实现。比较涵盖去重后的 134 个主要评估程序和 72 个独立确认程序。

D.2. 独立确认集的各程序结果

Ta表I-4列出了独立确认集内各程序的完整电路资源 .

表 I-4：独立确认集的 36 个非同构图上 72 个有序程序的各程序资源

n	d	模块	ID/顺序	参考 CX	我们的 CX	我们的深度	我们的 1Q
48	3	混合	0/A	1,228	986	714	1,368
48	3	混合	0/S	1,338	1,030	704	1,368
48	3	混合	1/A	1,198	988	752	1,368
48	3	混合	1/S	1,370	1,032	665	1,368
48	3	双顶点	0/A	1,130	880	480	1,368
48	3	双顶点	0/S	1,216	954	564	1,368
48	3	双顶点	1/A	1,134	886	502	1,368
48	3	双顶点	1/S	1,206	946	553	1,368
48	5	混合	0/A	2,324	2,108	3,348	3,972
48	5	混合	0/S	2,572	2,148	2,763	4,046

n	d	模块	ID/顺序	参考 CX	我们的 CX	我们的深度	我们的 1Q
48	5	混合	1/A	2,352	2,104	3,486	3,970
48	5	混合	1/S	2,546	2,152	2,688	4,032
48	5	双顶点	0/A	2,278	2,052	3,269	4,000
48	5	双顶点	0/S	2,352	2,118	2,623	4,056
48	5	双顶点	1/A	2,270	2,048	3,208	3,996
48	5	双顶点	1/S	2,360	2,124	2,556	4,084
48	7	混合	0/A	4,244	3,260	6,633	6,884
48	7	混合	0/S	4,538	3,308	6,205	6,994
48	7	混合	1/A	4,272	3,258	7,087	6,870
48	7	混合	1/S	4,516	3,310	6,193	6,984
48	7	双顶点	0/A	4,178	3,206	6,860	6,896
48	7	双顶点	0/S	4,256	3,268	5,710	6,964
48	7	双顶点	1/A	4,184	3,204	6,743	6,908
48	7	双顶点	1/S	4,292	3,292	6,381	7,016
96	3	混合	0/A	2,426	1,938	1,018	2,736
96	3	混合	0/S	2,786	2,018	639	2,736
96	3	混合	1/A	2,378	1,976	1,402	2,736
96	3	混合	1/S	2,730	2,036	721	2,736
96	3	双顶点	0/A	2,266	1,768	665	2,736
96	3	双顶点	0/S	2,406	1,904	790	2,736
96	3	双顶点	1/A	2,246	1,780	560	2,736
96	3	双顶点	1/S	2,422	1,922	614	2,736
96	5	混合	0/A	4,704	4,212	5,292	7,958
96	5	混合	0/S	5,102	4,280	3,458	8,132
96	5	混合	1/A	4,622	4,208	4,035	7,938
96	5	混合	1/S	5,068	4,294	3,098	8,092
96	5	双顶点	0/A	4,548	4,094	4,317	7,992
96	5	双顶点	0/S	4,736	4,262	3,629	8,148
96	5	双顶点	1/A	4,514	4,102	3,959	7,988
96	5	双顶点	1/S	4,674	4,244	3,490	8,124
96	7	混合	0/A	8,512	6,518	11,456	13,750
96	7	混合	0/S	8,972	6,600	9,148	13,904
96	7	混合	1/A	8,536	6,522	11,285	13,750
96	7	混合	1/S	9,024	6,602	8,987	13,976
96	7	双顶点	0/A	8,366	6,410	9,446	13,824
96	7	双顶点	0/S	8,566	6,590	8,593	14,052
96	7	双顶点	1/A	8,378	6,406	11,026	13,812
96	7	双顶点	1/S	8,592	6,584	9,235	14,048
192	3	混合	0/A	4,762	3,996	1,537	5,472
192	3	混合	0/S	5,338	4,136	778	5,472
192	3	混合	1/A	4,744	3,954	1,075	5,472
192	3	混合	1/S	5,414	4,100	737	5,472
192	3	双顶点	0/A	4,512	3,546	742	5,472
192	3	双顶点	0/S	4,830	3,816	708	5,472
192	3	双顶点	1/A	4,514	3,538	704	5,472
192	3	双顶点	1/S	4,898	3,862	730	5,472
192	5	混合	0/A	9,380	8,420	5,301	15,874
192	5	混合	0/S	10,158	8,572	4,423	16,112
192	5	混合	1/A	9,368	8,422	5,608	15,884
192	5	混合	1/S	10,074	8,564	4,206	16,158
192	5	双顶点	0/A	9,084	8,192	4,769	15,996
192	5	双顶点	0/S	9,472	8,524	3,807	16,284
192	5	双顶点	1/A	9,090	8,190	5,148	15,984
192	5	双顶点	1/S	9,464	8,494	4,322	16,268
192	7	混合	0/A	17,012	13,034	15,435	27,504
192	7	混合	0/S	18,014	13,218	11,588	27,942
192	7	混合	1/A	16,972	13,038	15,219	27,518
192	7	混合	1/S	17,824	13,200	10,335	27,856
192	7	双顶点	0/A	16,752	12,812	13,809	27,632
192	7	双顶点	0/S	17,140	13,144	11,720	28,044
192	7	双顶点	1/A	16,736	12,816	13,915	27,616
192	7	双顶点	1/S	17,088	13,130	10,339	27,984

ID 表示生成重复索引。A 表示递增顺序，S 表示固定伪随机目标顺序。

D.3. 公开图上两种目标顺序的配对结果

表I-5 按图汇总了 24 个有序程序的资源。每个斜线前后的数值分别对应升序（A）和固定伪随机顺序（S）。每种顺序均作为单独的程序进行测量。

表 I-5：两种目标顺序下 12 个公开图中 24 个程序的资源

图	n	参考 CX A/S	我们的 CX A/S	我们的深度 A/S	参数旋转 A/S
david.col	87	10,088/10,072	6,780/6,772	13,615/13,453	1,274/1,274
games120.col	120	16,228/16,226	12,326/12,330	12,561/12,951	2,039/2,039
huck.col	74	5,652/5,644	3,432/3,420	5,677/5,910	1,057/1,057
jean.col	80	5,222/5,234	3,500/3,512	6,138/6,456	908/908
myciel3.col	11	260/260	226/226	397/417	187/187
myciel4.col	23	1,604/1,604	1,152/1,152	2,107/2,428	391/391
myciel5.col	47	8,272/8,272	4,536/4,536	9,000/9,879	799/799
myciel6.col	95	19,828/19,828	15,840/15,840	32,164/34,439	1,615/1,615
queen5_5.col	25	3,240/3,240	3,240/3,240	7,162/7,612	425/425
queen6_6.col	36	6,096/6,096	6,096/6,096	13,388/14,291	612/612
queen7_7.col	49	10,248/10,248	10,248/10,248	21,830/23,374	833/833
queen8_8.col	64	15,936/15,936	15,936/15,936	34,076/36,318	1,088/1,088

参照值为开放CPG和两个匹配的精确内核的逐程序 CX 基准。A/S 值按相同顺序显示。

D.4. 物理路由和终端线路顺序恢复的 *CX*开销

表 I-6 总结了线性拓扑和二维网格的路由输入 CX 计数、最终物理 CX 计数以及终端线路顺序恢复成本。新增恢复 CX 计数衡量的是：在最后一次局部化简遍之前，为恢复原始线路顺序而分解所需 SWAP 门时新增的 CX 门数量。物理 CX 的净增量是最终物理电路与路由输入电路之间的差值，反映了物理映射、终端线路顺序恢复和局部化简的综合影响。路由输入已经过通用原生综合。

表 I-6：物理路由和终端线路顺序恢复的 CX 资源

拓扑	路由输入 CX 门	最终物理 CX 门	新增恢复 CX 门	物理 CX 门净增量
线性	104,876	1,781,104	166,149	1,676,228
二维网格	104,738	598,736	49,896	493,998

每种拓扑汇总计 22 个程序和 3 个固定种子的 66 个程序–种子对。输出电路与路由输入进行比较，比较对象为相同的程序和种子。

D.5. 固定模块数量下单候选编译时间和内存开销

表 I-7 比较了全缓存路径和源级最优缓存路径在八个真孪生类下的时间和内存成本，每条路径编译一个候选项。为区分模块大小和模块数量的影响，实验将商图固定为 $K_{2,\,6}$，并使用 $n = 16$，32，64，128，256 个数据量子比特。每种条件均在三个独立的单线程 Windows 进程中运行。时间指标是五个独立阶段（结构识别、源级选择、电路构造、原生门集转换和原生电路后处理）各自三次运行的中位时间之和。该指标不同于一次完整编译的实际运行时间。RSS 指标是三次运行中采样到的进程峰值的中位数。

表 I-7：固定包含八个真孪生类的两种单候选路径的时间和内存开销

n	全缓存 time	源选择 time	全缓存 RSS	源选择 RSS
16	0.26	0.28	105.9	106.6
32	0.35	0.39	109.1	109.0
64	0.55	0.73	115.3	115.4
128	1.25	1.80	128.1	129.7
256	4.82	6.99	162.9	164.7

时间是以下各项之和：五个阶段的执行时间中位数（秒）。RSS 是三个独立进程的采样峰值的中位数（MiB）。