用于估计带符号边流中三角平衡的完美可分布式量子-经典算法

STEVEN KORDONOWY, Fujitsu Research of America Inc.，美国圣克拉拉和美国加利福尼亚大学圣克鲁兹分校BIBHAS ADHIKARI 和 HANNES LEIPOLD，Fujitsu Research of America Inc.，美国圣克拉拉

我们开发了一种完美可分布式量子-经典流算法，用于处理带符号边，并在单遍边流中高效估计具有多种带符号配置的三角形数量。我们的方法引入了一个量子草图寄存器来处理带符号边流，并在量子估计器中使用测量算子进行查询对调用；同时，互补的经典估计器负责统计量子过程未捕获的三角形。这种混合设计相较于纯经典方法具有多项式空间优势，将已知结果从无符号边流数据推广到了带符号情形。我们量化了随机带符号图实例中的不平衡程度，并展示了经典算法和混合算法在实践中如何估计平衡。

关键词：量子-经典算法、量子流处理、带符号网络、结构平衡、可完美分布式的量子算法

arXiv:2603.16029v1 [quant-ph] 17 Mar 2026

1 引言

在大数据时代，如何在内存受限的资源条件下计算统计量，以概括大规模数据（尤其是大型网络），变得日益重要。量子计算系统有望逐渐成熟，并在这一不断发展的领域发挥关键作用 [1]。例如，在单遍流处理设置中，量子设备已被证明能在许多场景中带来空间优势，包括单向布尔函数 [2], 、三角形计数 [3],和有向最大割 [4]。尽管经典流算法的分布式计算框架层出不穷，文献中却缺乏适用于量子流算法的量子分布式框架。本文尝试填补这一空白，开发一种在分布式计算环境中实现的量子-经典流算法，用于对带符号边流进行三角形计数。

在边流图模型中寻找并计数三角形是一项基础问题，因为它对网络分析和算法至关重要。在该模型中，图的边以数据流的形式依次到达，算法需要使用次线性空间（通常只需遍历一遍）检测或估计三角形数量。早期研究为仅插入流中的三角形计数确立了下界和随机化算法，凸显了固有的空间-精度权衡 [5]。后续方法则引入了基于采样的技术和草图技术，以近似估计大规模图中的三角形计数 [6, 7]。参考文献 [8], 中描述的算法（下文称为 Jayaram-Kallaugher（JK）算法）已被证明在空间复杂度上是最优的（相差至多多对数因子），它在该问题的标准参数化下匹配了已知下界，从而解决了边流设置中的经典三角形计数问题。参考文献 [3], 中，Kallaugher 提出了一种用于近似三角形计数的量子-经典算法，其空间复杂度相比经典最优 JK 算法具有多项式级优势。

本文考虑单遍仅插入带符号图流模型，其中每条带符号边只出现一次。带符号图有两种边：正边和负边（更多详情请参见第 2 节）。带符号图和平衡度量的应用十分广泛。在社会网络分析中，它们用于模拟信任/不信任关系，并帮助识别极化社群 [9, 10]。在国际关系中，带符号网络用于刻画国家间的联盟与冲突，其平衡动态与稳定性相关 [11, 12]。在金融领域，

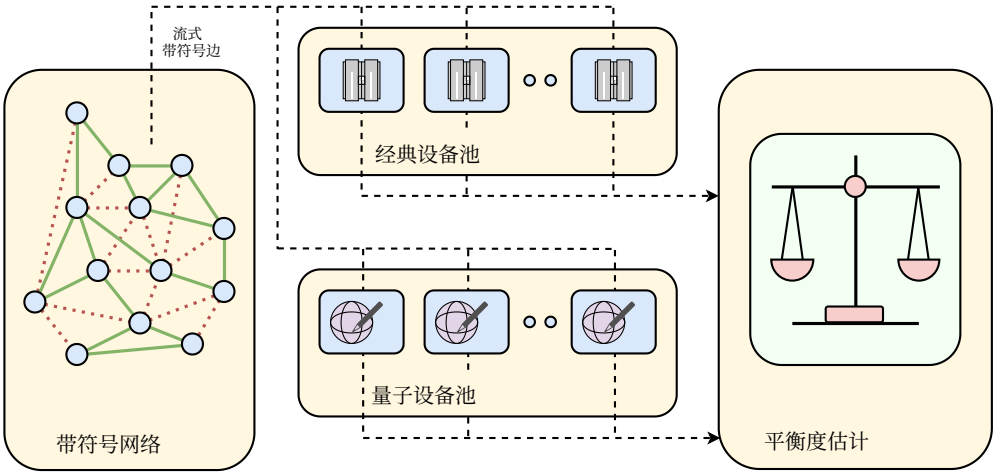

图1。**量子　　经典流模型中的平衡估计**。我们混合
量子经典　　用于估计平衡的算法。

有符号相关图描述资产收益率的共同变动与反向变动，平衡性分析则有助于深入了解系统性风险和市场稳定性 [13–18]。在生物学中，带符号网络用于模拟基因调控网络和蛋白质网络中的合作与竞争性相互作用 [19, 20]。在物理学和复杂系统领域，平衡理论是对抗性相互作用下共识与合作模型的基础 [21, 22]。

在经典设定下，我们将用于边流中三角形计数的经典最优流式 JK 算法扩展到有符号边流设定，并称之为带符号JK 算法。在该模型中，数据流中的每条边都关联一个符号，因此三角形可分为四种不同类型（见图2）。这带来了额外的挑战，因为算法必须跟踪边符号的乘积，以便在查询处理过程中正确识别每个三角形的类型。随后，利用不同三角形类型的估计值来近似底层带符号图的平衡程度，该程度由三角形平衡度量定义。

此外，我们通过开发一种针对带符号网络的三角形计数算法，重点研究平衡的统计特征，进一步拓展了量子草图在重要且实际的数据流场景中的空间优势。随后，我们提出了一种混合量子-经典算法，其相较于有符号 JK 算法具有量子多项式优势。所提出的算法依赖于两个互补的估计量：量子估计器和经典估计器。量子估计器采用一组量子处理器，每个处理器都执行类似的任务，包括采样和执行量子查询；经典估计器则由一组经典处理单元组成，执行类似的基于采样的过程。这两个估计器在混合框架中发挥互补作用。具体而言，由于量子测量具有概率性，量子估计器可能遗漏某一给定类型的三角形，而经典估计器能够以非零概率检测到这些三角形。整体混合方法的灵感来自参考文献 [3] 中提出的用于估计无符号边流中三角形计数的经典—量子算法。本研究所要解决的主要技术挑战在于构造一个量子草图寄存器，用以编码在带符号边流中观测到的带符号边，同时设计适当的

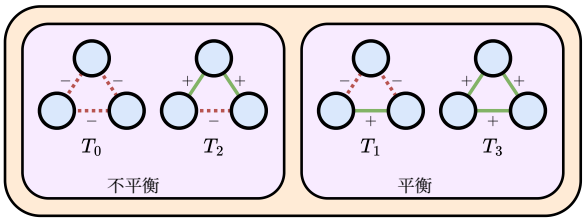

图 2. **带符号网络中的三角形类型及其平衡性**。三角形类型按其正边数量标记；正边数量为奇数的三角形归类为平衡三角形，而正边数量为偶数的三角形则归类为不平衡三角形。根据结构平衡理论，现实网络的动态变化可能会趋向于形成更平衡的局部子结构。因此，统计不同类型的三角形对于分析网络至关重要。

用于对该寄存器执行量子查询的 POVM 算子。图3给出了所提方法的电路级描述。随后，我们利用这种对三角形类型的近似来估计网络的平衡。

最后，我们介绍一种融合经典处理器与量子处理器的分布式计算框架，用于实现所提出的混合量子-经典算法，以估计不同类型的带符号三角形，并据此近似底层带符号图的结构平衡性。为评估性能，我们从随机带符号图生成带符号边流，并通过态矢量模拟，将所提出的混合方法与其纯经典对应算法进行比较。据我们所知，这是文献中首个专门用于在图数据流上实现流式算法的分布式量子-经典计算模型。本文提出的框架如图 1 所示。

我们强调，本文提出的分布式框架与标准分布式框架有着根本区别。经典方法依赖互连机器，必须应对诸如并行性、负载均衡和网络效率 [23]等挑战。标准的分布式量子计算通常侧重量子网络方面的挑战，包括如何在多个量子处理单元（QPU）之间实现量子门等量子操作。完全可分布式 量子算法是指每个处理器都能独立参与、无需处理器间通信的算法 [24]。我们证明，参考文献 [3]中的混合算法和本文提出的算法都属于这一类：一组量子处理单元和经典处理单元（中央处理器）独立运行，最终估计值通过对各自结果进行经典后处理并加以汇总得到（见第 5 节）。

2 图流、流中的三角形与带符号图的平衡

本节简要回顾现有的图流分布式计算框架，以及带符号图的结构平衡理论。我们还将讨论后文所用的一组参数。

2.1 图流

在经典流数据分析模型中，图流模型将经典数据流范式扩展到图结构数据，其中顶点和边以数据流的形式依次到达。在这种设定下，输入图过于庞大，无法显式存储在内存中，因此算法需要使用次线性空间处理更新，例如边插入、删除或权重变化

并且通常只需对数据流进行一次或少数几次扫描。根据更新规则，图流通常分为仅插入、转闸模型和动态模型。图流框架已成为分析社交媒体、通信系统和生物数据中大规模网络的核心工具，并为基本图问题确立了空间-精度权衡 [25–29]。

经典流算法，尤其是图流算法，自然而然地应用于分布式计算框架，以进行大规模图分析，并在空间、通信复杂度和近似精度方面提供理论保证 [23, 25, 26, 30]。目前已有多种开源分布式图处理框架，例如以顶点为中心的编程模型：谷歌的 Pregel [31],Pregel+ [32]、GraphLab [33], PowerGraph [34],、Giraph [35],和 GPS [36]。这些框架采用并行循环、消息接收与发送以及广播等抽象。此外，还提出了以边为中心和以子图为中心的分布式图处理框架，以提升性能；这类框架需要边列表，并绕过与顶点相关的随机内存访问，例如 X-Stream [37], WolfGraph [38], Giraph++ [39]、GoFFish [40],Blogel [41]。

2.2 有符号图和有符号三角形

带符号图 $G = (V, E, \sigma)$ 是一种图，其中每条边 $(i, j) \in E$ 都被赋予一个符号 $\sigma(i, j) \in \{+1, -1\}$，表示正向（友好、合作、正相关）或负向（敌对、对抗、负相关）关系 [42, 43]。可以通过统计网络中包含的小型子结构数量来分析带符号网络；最简单的例子就是三角形的数量。由于网络带有符号，三角形的类型由三角形中三条边的符号决定，如图2所示。带符号图中有四种类型的三角形，具体取决于三角形中正边的数量。将每种类型三角形的计数定义为

$$T_j, \textit{ the number of triangles with } j \textit{ positive edges in } G. \tag{1}$$

那么，三角形总数就是各类型三角形数量之和：

$$T = T_0 + T_1 + T_2 + T_3, \textit{ the number of triangles in } G. \tag{2}$$

2.3 结构平衡理论

Harary 提出的经典平衡概念指出，如果带符号图的顶点集可以划分为两个互不相交的子集，使所有正边都位于各子集内部，所有负边都位于两个子集之间，那么该带符号图就是平衡的。等价地，如果每个环的符号均为正，即沿环各边符号的乘积为 +1，那么该带符号图就是平衡的。

已有研究提出了多种平衡的定量指标。一种广泛使用的指标是平衡指数，其定义为

$$\mathcal{B}(G) = \frac{\#\{\text{balanced cycles in } G\}}{\#\{\text{total cycles in } G\}}, \tag{3}$$

它反映了网络 [44] 中平衡结构所占的比例。另一种指标是挫折指数，即使图达到平衡所需的最少边符号变化次数（或删除数量） [45]。

因此，在结构平衡理论中，类型为 T_1 和 T_3 的三角形是平衡的，而类型为 T_0 和 T_2 的三角形是不平衡的。因此，平衡的带符号图只包含类型为 T_0 和 T_2 的三角形，因此可以通过统计所有类型的三角形来估计带符号图的平衡缺失程度

图。特别地，带符号图的三角形平衡指数 G 定义为

$$\mathcal{B}_{\triangle}(G) = \frac{\#\{\text{balanced triangles in } G\}}{\#\{\text{total triangles in } G\}}. \tag{4}$$

由于三角形是由三个节点构成的环，$\mathrm{B}_{\triangle}$ 是 B 的一种特殊类型（见式 (3)）。

2.4 数据流中的参数近似

内存受限设备在解决问题时无法访问全部输入数据，因为数据可能过于庞大，无法一次性装入内存，例如规模极大的社交媒体网络；也可能是由于数据本身会在一段时间内陆续到达，例如金融系统中的交易。在单遍流式处理场景中，我们获得由边 $uv \in E(G)$ 及其符号 $\sigma_{uv} \in \{+1,\ -1\}$ 组成的数据流 $S = ((uv,\ \sigma_{uv})_i)_{i=1}^{m}$。我们希望对网络 G 中的 T_j（根据式（1））进行（近似）计数。我们采用仅插入模型，其中边逐条处理且从不删除。给定误差参数 $\epsilon,\ \delta \in (0, 1]$,，任务是以概率 $1 - \delta$ 使用尽可能少的空间资源，对 T 求一个 $(1 \pm \epsilon)$-乘法近似。

任何算法近似保证都应独立于 S的排序。事实上，为了给出最坏情况保证，可以构造 S 的对抗性排序。为了构造困难的图实例，我们为图 G定义以下标准图参数：

$n :=|V|$，图中的节点数Δ_V，与任一节点相邻的三角形最大数量

$m :=|E|$，图中的边数Δ_E，与任一边相邻的三角形最大数量

当前最先进的三角形计数流式算法的空间复杂度是根据这些参数给出的。请注意，$T,\ \Delta_E,\ \Delta_V$无法预先获知，否则算法就失去了意义。

参考文献 [1] 提出了一个框架，用于设计混合量子-经典算法，以近似求解能够借此获得优于纯经典解法的算法的问题。

3 一种用于带符号三角形计数的纯经典流式算法

受参考文献 [8], 中针对（无符号）边流提出的最优经典流式 JK 算法启发，我们开发了一种流式算法，用于估计带符号边流中类型为 T_1 的三角形数量。该算法可以轻易地调整为统计任意（或所有）T_j 类型的三角形。

我们先简要回顾流式 JK 算法。在他们的工作之前，针对仅插入流中的三角形计数以及线性草图算法的若干上界和下界已经确立；参见参考文献 [8] 及其中的参考文献。他们方法背后的关键洞见如下：给定数据流中依次到达的三条边 $uv,\ vw,\ wu \in E$，如果前两条边 uv 和 vw 被采样并存储，那么在观察到闭合边 wu 时，就能检测到底层图 G 中是否存在三角形 uvw。

然而，仅采用朴素的随机边采样通常会导致性能欠佳。为克服这一局限，Jayaram 和 Kallaugher 提出了一种经过精心参数化的采样策略，将边采样和顶点采样结合起来。具体而言，以概率 p_E 采样边，同时以概率 p_V 独立采样顶点，并在采样楔形闭合时统计三角形。采样概率选取为

$$p_E = \frac{\Delta_V}{T}, \qquad p_V \geq \max\left\{\frac{\Delta_V}{\Delta_E}, \frac{1}{\sqrt{\Delta_V}}\right\},$$

其中， T 表示三角形总数， Δ_V 和 Δ_E 表示上文定义的适当集中参数。采用这一组参数后，所得算法的空间复杂度为

$$O\left(\frac{m}{T}\left(\Delta_E+\sqrt{\Delta_V}\right)\right),$$

并得出一个方差为 O(T^2) 的估计量。

算法 1 T_1 估计量（纯经典）

过程 $\mathbf{T}_1$_估计量_子程序_完全经典(E; $\mathbf{h}_E$, $\mathbf{h}_V$; $\mathbf{p}_E$, $\mathbf{p}_V$) 输入： E **带符号边流；** h_E, h_V **概率哈希；** $p_E, p_V \in (0,1]$**. 输出：估计值** R_1**. 1:** $R_1 \leftarrow 0, S \leftarrow$ **{:} 2: 对** $(vw, \sigma_{vw}) \in E$**：执行 3: 对** $u \in V$**：执行 4: 如果** $h_V(u) = 1$ **且** $uv, uw \in S.keys()$**，则** ▷ **发现一个三角形，检查它是否是带符号边 5: 如果** $\sigma_{vw} = +1$ **且** $S[uv], S[uw] = -1$**，则 6:** $R_1 \leftarrow R_1 + 1/(p_V p_E^2)$**7: 否则如果** $\sigma_{vw} = -1$ **且** $S[uv] \neq S[uw]$**，则 8:** $R_1 \leftarrow R_1 + 1/(p_V p_E^2)$**9: 如果** $h_E(vw) = 1$ **且 (**$h_V(v) = 1$ **或** $h_V(w) = 1$**)，则 10:** $S[vw] \leftarrow \sigma_{vw}$**11: 返回** R_1

针对带符号边流采用类似的方法，我们对顶点和边使用相同的采样概率，并分别通过概率哈希函数 h_V 和h_E实现。然而，在带符号边流中，我们考虑 T_j、 $j = 0$、1、2、3这几种三角形类型。为处理数据流中入边的符号，我们将带有符号的边存储在草图中。对于一条入边（， σ_{vw}），当顶点 u被采样，且 uv， $uw \in S$时，我们根据 σ_{vw}中 uv和 uw.的符号，增加三角形类型 T_j的计数。草图会根据概率 $h_E(vw)$ $h_V(v)$和 $h_V(w)$，以随机方式更新，以纳入 vw。我们在算法 1 中描述了 $j = 1$ 的算法。请注意，我们的算法扩展了 JK 算法，将数据流中边的符号纳入其中，因此称为有符号 JK 算法。

那么我们得到以下推论，它给出了该算法空间的比特复杂度 .

推论 1. 对于较小的误差参数 $\epsilon, \delta > 0$，存在一种适用于仅插入图流的中位数均值算法，能够以概率 $1 - \delta$ 将图 G 中的 T_1 三角形数量近似到 ϵT_1 精度，使用

$$O\left(\frac{m}{T_1}\left(\Delta_E+\sqrt{\Delta_V}\right)\log n\frac{1}{\epsilon^2}\log\frac{1}{\delta}\right) \tag{5}$$

比特空间。该算法使用算法 *1* 作为子程序。

由于这是 JK 算法的一个简单带符号扩展，证明过程几乎与参考文献 [8] 中定理 1.1 的证明完全相同。

4 用于带符号三角形计数的混合量子-经典算法

这里我们介绍用于估计 T_1 三角形的混合算法。类似的方法也可用于其他类型的三角形。与估计无符号三角形 [3] 一样，整个框架分为 2 个并行运行的子程序。用于估计 $T_1^{<k}$ 的量子算法在第 4.2 节中描述。用于估计 $T_1^{>k}$ 的经典算法在第 4.1 节中描述。将两者相加即可得到 T_1的估计值，具体如第 4.3 节所述。

4.1 针对 $T_1^{>k}$ 的经典估计器

我们将参考文献 [3] 中的 ClassicalEstimator 扩展为以下算法中的带符号版本。

算法 2 $T_1^{>k}$估计器（经典）

过程：T_1_估计量_经典_子程序(E; k; h_V, h_I)
输入：E带符号边流；$k \in [1, m]$；h_V，h_I概率哈希。
输出：估计值 $R_1^{>k}$。

$R_1^{>k} \leftarrow 0$，$S \leftarrow \emptyset$，$D_- \leftarrow$ {:} $D_+ \leftarrow$ {:} $\ell \leftarrow 0$
对于 $(vw, \sigma_{vw}) \in E$: **执行**
　对于 $u \in V$: **执行**
　　if $\sigma_{vw} = +$ **然后**
　　如果 $(u, v, -), (u, w, -) \in S$**然后**
　　$R_1^{>k} \leftarrow R_1^{>k} + 1 - \left(1 - 1/k\right)^{D_-[(u,v)]+D_+[(u,v)]+D_-[(u,w)]+D_+[(w,v)]}$
　　else
　　如果 $(u, v, +), (u, w, -) \in S$**然后**
　　$R_1^{>k} \leftarrow R_1^{>k} + 1 - (1 - 1/k)^{D_-[(u,v)]+D_-[(u,w)]+D_+[(u,w)]}$
　　如果 $(u, v, -), (u, w, +) \in S$**然后**
　　$R_1^{>k} \leftarrow R_1^{>k} + 1 - (1 - 1/k)^{D_-[(u,v)]+D_+[(u,w)]+D_-[(u,w)]}$
　对于 $(x, y, t) \in S$**执行**
　　如果 $y \in \{v, w\}$**则**
　　if $\sigma_{vw} = +1$**则**
　　$D_+[(x, y)] \leftarrow D_+[(x, y)] + 1$
　　else
　　$D_-[(x, y)] \leftarrow D_-[(x, y)] + 1$
　如果 $h_V(v) = 1$ 且 $h_I(\ell) = 1$ **则**
　　$S.add((v, w, \sigma_{vw}))$ 和集合 $D_-[(v, w)]$，$D_+[(v, w)] \leftarrow 0$。
　如果 $h_V(w) = 1$ 且 $h_I(\ell + 1) = 1$ **则**
　　$S.add((w, v, \sigma_{vw}))$ 和集合 $D_-[(w, v)]$，$D_+[(w, v)] \leftarrow 0$。
　$\ell \leftarrow \ell + 2$
返回 $R^{>k}{}_1$

概率哈希函数必须满足 $Pr(h_V(u)) = 1/\sqrt{km}$ 且 $Pr(h_I(t)) = \sqrt{k/m}$。h_V需要存储起来，因为它会被调用多次，但可以简化为仅两两独立，这需要 $O(\log n)$ 位。h_I只需查询一次，然后即可丢弃，因此我们不需要存储完整函数。

引理 2 (算法 2 的分析)。*(1)* 平均而言，该算法运行需要使用$O(\log n)$比特。

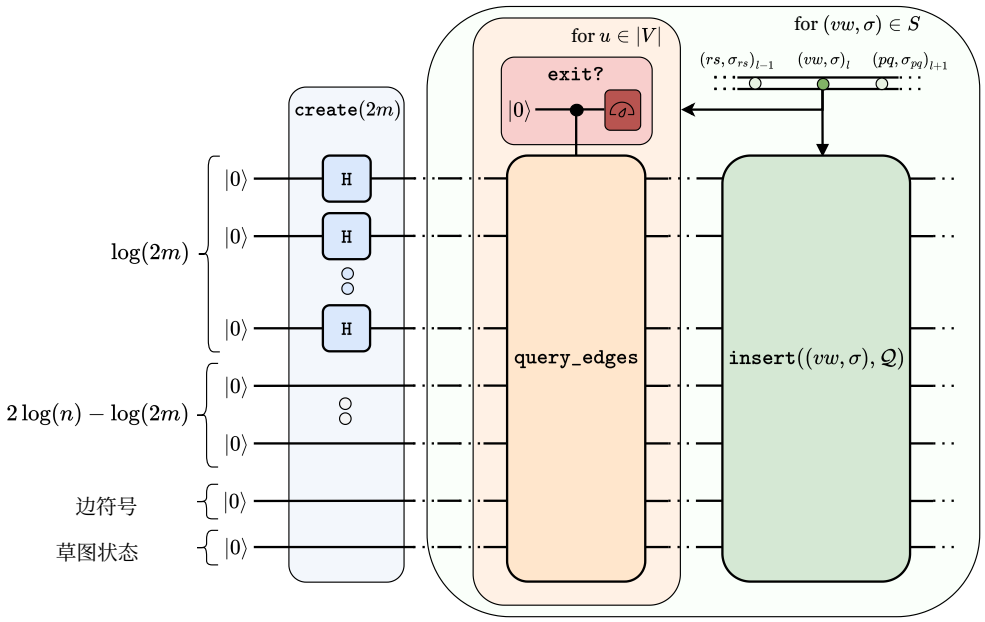

图 3. **量子草图算法的电路图**。量子草图板的电路通过 3 种操作来更新量子态，如第 4.2 节所述。首先，应用 create 来生成初始草图板；图中展示的是 log(2m) 为 2 的幂的情况，因此可以使用哈达玛门。对于数据流中的每条带符号边 $(vw,\ \sigma)$，对每个与 vw 和 σ 相关的 $u \in |V|$ 应用 query_edges。如果某个投影值测量的测量结果不是 $\perp$，算法就会停止并返回该值。最后，应用 insert 将 $(vw,\ \sigma)$ 和 $(wv,\ \sigma)$ 添加到草图板中，然后继续处理下一条边。

(2) $\mathbb{E}(R_1^{>k}) = \frac{T_1^{>k}\sqrt{k}}{m^{3/2}}$

(3) $Var(R_1^{>k}) \leq 4\frac{T_1^{>k}\sqrt{k}\Delta_E}{m^{3/2}}$

推论3。 存在一种以算法*2*为子程序的中位数均值算法，它能以概率 $1-\delta$ 输出 $T_1^{>k}$ 至 ϵT_1 精度的估计值。该算法平均需要使用 $O\left(\frac{m^{3/2}}{T\sqrt{k}}\Delta_E \log n \frac{1}{\epsilon^2}\log\frac{1}{\delta}\right)$个经典比特和量子比特。

这些陈述的证明与参考文献 [3] 中引理8–11的证明相似。特别是，唯一的区别在于第 5、8 和 10 行，我们会检查是否存在一个以楔形 $(vw,\ \sigma_{vw})$ 为中心的 T_1 三角形。参考文献 [3]，中的 ClassicalEstimator 在第 7 行检查的是以楔形 vw 为中心的任意三角形。在这两种情况下，一旦找到所需的三角形，就会将相应的值加到估计值中。

4.2 $T_1^{<k}$的量子估计器

我们现在使用一个量子子程序来近似 $T_1^{<k}$。该子程序依赖于参考文献 [1]中概述的 3 个量子原语。从较高层次来看，该算法将已见边的草图 S 记录到量子态 Q_S 中，并通过测量 Q_S来查询该草图。我们在$2\lceil log_2 n\rceil + 2$ 个量子比特上维护一个状态，其中 n 是底层图的顶点数。在读取一条

带符号边 $(vw,\ \sigma_{vw})$ 后，会查询底层草图，以查找楔形 $wu,\ wv$（以及符号信息）。此查询通过测量完成（更多详情请参见 C.2 节）。如果没有找到楔形，则将 $(vw,\ \sigma_{vw})$ 插入草图，然后继续处理下一条边。需要注意的是，该算法具有破坏性；处理一条边 $(vw,\ \sigma_{vw})$ 会破坏与 v 和 w相连的边的先前信息。与此对应的电路如图3所示。

算法3 $T_1^{<k}$估计量子程序

过程T$_1$_估计量_量子_子程序(E，k，g)
输入：E带符号边流；$k \in [1, m]$；g概率哈希。
输出：估计值 $R_1^{<k}$。

1: $\mathsf{Q}_{\mathcal{S}}$:= 创建($[2m] \times \{\xi\}$).
2: $\ell := 0$
3: **对于** $(vw, \sigma_{vw}) \in E$: **执行**　　▷ 假设 $v < w$
4:　　**如果** $g(\ell) = 1$ **则**
5:　　**对于** $u \in V$: **执行**
6:　　**if** $\sigma_{vw} = +$ **然后**
7:　　r_{--} := 查询_边($(uv,\ -)$，$(uw,\ -)$，Q)
8:　　**如果** $r_{--} \neq \perp$ **则返回** $r_{--} \cdot km$
9:　　**else**
10:　　r_{+-} := 查询_边($(uv,\ +)$，$(uw,\ -)$，Q)
11:　　**如果** $r_{+-} \neq \perp$ **则返回** $r_{+-} \cdot km$
12:　　r_{-+} := 查询_边($(uv,\ -)$，$(uw,\ +)$，Q)
13:　　如果 $r_{-+} \neq \perp$ **则返回** $r_{-+} \cdot km$
14:　　插入($((vw, s)\ \ell, \mathsf{Q}_{\mathcal{S}})$
15:　　$\ell := \ell + 1$
16: **返回**0

在参考文献 [3], 中的量子估计器扩展中，我们需要跟踪带符号边并对其进行查询。注意，第 9 行和第 11 行的查询针对互不相交的边对，因此彼此独立。

引理 4（算法 3 的性能）。令 $R_1^{<k}$为算法 *3* 的输出。
（*1*）该算法运行需要$2\log（n）+2 = O（\log n）$ 个量子比特。（*2*） $\mathbb{E}[R_1^{<k}] = T_1^{<k}$。（*3*） $Var（R_1^{<k}）\ \leq （km）^2$。

该算法中（2）和（3）的证明见 D 节。

推论 5。存在一种中位数均值算法，以算法 *3* 为子程序，以概率 $1-\delta$ 输出精度为 ϵT_1 的 $T_1^{>k}$ 估计值。该算法期望使用$O\left(\left(\frac{km}{T_1}\right)^2 \log（n）\ \frac{1}{\epsilon^2}\log\left(\frac{1}{\delta}\right)\right)$个经典比特和量子比特。

4.3 $T_1 = T_1^{>k} + T_1^{<k}$ 的混合算法

现在，我们可以给出用于估计 T_1的完整算法了。请注意，该算法假设已知 m，Δ_E，以及 T_1 本身的一个合理上界。这与参考文献 [3] 中的定理 1 在陈述和证明上几乎完全相同。

定理 6. 对于任意 $\epsilon, \delta \in (0, 1],$，都存在一种混合量子-经典流式算法，其使用

$$O\left(\frac{m^{8/5}}{T_1^{6/5}}\Delta_E^{4/5}\log n \cdot \frac{1}{\epsilon^2}\log\frac{1}{\delta}\right) \tag{6}$$

期望使用量子比特和经典比特，在仅插入图流中以概率 $1-\delta$ 返回 T_1 的$(1\pm\epsilon)$乘法近似。

证明。设 T_1_Estimator_Classical 和 T_1_Estimator_Quantum 分别为推论 3 和推论 5 所保证的中位数均值算法。以下算法可达到定理所述的性能。第 2 行和第 3 行按顺序书写，但实际上应并行运行。

算法 4 带符号 T_1 估计量（量子-经典）

过程 T_1_Estimator_Hybrid(E ϵ, δ) 输入： E
带符号边数据流； ϵ、δ**误差参数。输出：估计值** R_1。
1： $k \leftarrow \left\lceil \frac{(T_1)^{2/5}}{m^{1/5}}\left(\frac{\Delta_E}{\epsilon}\right)^{2/5} \right\rceil$ **并行运行以下子程序：2：**
$R_1^{>k} \leftarrow$ **T_1_Estimator_Classical(E, k ϵ/2, δ/2) 3：**
$R_1^{<k} \leftarrow$ **T_1_Estimator_Quantum(E, k ϵ/2, δ/2)**
4：返回 $R_1^{>k} + R_1^{<k}$

为证明准确性，回顾一下：$R_1^{>k}$是对 $T_1^{>k}$的估计值，精度为 $\epsilon T_1/2$，概率为$1-\delta/2$。同样，$R_1^{<k}$是对 $T_1^{<k}$的估计值，精度为 $\epsilon T_1/2$，概率为 $1-\delta/2$。因此，通过并集界，$R_1 = R_1^{>k} + R_1^{<k}$ 是对 $T_1 = T_1^{<k} + T_1^{>k}$的估计值，精度为 ϵT_1，概率为 $1-\delta$。

至于所需空间，注意到忽略共同的对数项后，估计 $R_1^{>k}$和 $R_1^{<k}$分别需要 $\tilde{O}\left(\frac{m^{3/2}}{T_1\sqrt{k}}\Delta_E\right)$ 和$\tilde{O}\left(\left(\frac{km}{T_1}\right)^2\right)$空间。当 $k = \left\lceil \frac{(T_1)^{2/5}}{m^{1/5}}\left(\frac{\Delta_E}{\epsilon}\right)^{2/5} \right\rceil$时，这两者都变为 $\tilde{O}\left(\frac{m^{8/5}}{T_1^{6/5}}\Delta_E^{4/5}\right)$。

□

最后，我们想比较混合算法 4 与其经典对应算法算法 1 所使用的空间。正如参考文献 [3], 中指出的，能够最大化差距的图参数是 $\Delta_E = O(1)$ 和 $\Delta_V = \Omega(T_1) = \Omega(m)$，由此得到一个使用 $\tilde{O}(m^{2/5})$ 空间的混合算法，以及一个使用 $\Omega(\sqrt{m})$ 空间的纯经典算法。在我们的设定中，情况也同样如此。

5 用于估计带符号图平衡的分布式量子-经典计算模型

本节中，我们使用上一节开发的流式算法来估计带符号图的三角形平衡指数。为了以概率 $1-\delta$ 得到精度为 $(1\pm\epsilon)$-乘法近似的平衡估计值，我们以概率 $1-\delta$. 对三角形类型 T_j 的计数进行估计，使其达到精度为 $(1\pm\epsilon/(2+\epsilon))$-乘法近似。通过简单的代数运算即可证明这一点。随后，我们讨论一种量子分布式模型，用于实现所提出的估计平衡的量子-经典算法。

5.1 估计三角形平衡的算法

现在，我们使用各个三角形估计量，构造一种估计三角形平衡度的算法（式（4））。

推论 7。　　对于任意 ϵ、 $\delta \in (0,1]$,，存在一种混合量子-经典流式算法，其使用

$$O\left(\frac{m^{8/5}}{T^{6/5}}\Delta_E^{4/5}\log n \cdot \frac{1}{\epsilon^2}\log\frac{1}{\delta}\right) \tag{7}$$

量子比特和经典比特的期望数量，返回 $(1\pm\epsilon)$乘法近似的 $\mathcal{B}_{\triangle}(G)$，并且在仅插入图流中的成功概率为 $1-\delta$。

令 T_估计量_混合为参考文献 [3] 中引理 11 所述的算法。

算法 5 平衡估计器（混合）

过程 平衡估计器_混合(E ϵ, δ)
输入： S带符号边流； $\epsilon, \delta \in$ （0, 1] 误差参数。
输出：平衡估计值 $\mathrm{B}^{-H}_{\triangle}(G)$。
并行运行以下子程序：

1: $R_1 \leftarrow$ T$_1$_估计量_混合(S ϵ/(2 + ϵ) δ/3)
2: $R_3 \leftarrow$ T_估计量_混合(S ϵ/(2 + ϵ) δ/3)　　▷ 丢弃 −1边
3: $R \leftarrow$ T_估计量_混合(S ϵ/(2 + ϵ) δ/3)　　▷ 忽略符号
4: **返回** $(R_1 + R_3)/R$

作为混合算法的对比，我们还使用纯经典算法来估计 $\mathrm{B}_{\triangle}(G)$。这是对算法 1 的一个小幅改动。

算法 6 平衡估计器（纯经典）

过程 平衡_估计量_子程序_完全经典(E；h_E、h_V；p_E、p_V)
输入：E带符号边流；h_E、h_V概率哈希；p_E、$p_V \in (0,1]$。
输出：估计值 $B_{\triangle}^{-C}(G)$。

R_{bal}，$R_{unbal} \leftarrow 0$，$S \leftarrow \{:\}$
对于 $(vw, \sigma_{vw}) \in E$: **执行**
  对于 $u \in V$: **执行**
    若 $h_V(u) = 1$ 且 $uv, uw \in S.keys()$ **则** ▷ 发现一个三角形，检查类型是否正确
      if $\sigma_{vw} = +$ **则**
        如果 $(S[uv], S[uw] = -1)$ 或 $(S[uv], S[uw] = +1)$ **则**
          $R_{bal} \leftarrow R_{bal} + 1/(p_V p_E^2)$ ▷ 三角形类型 T_1 或 T_3
        else
          $R_{unbal} \leftarrow R_{unbal} + 1/(p_V p_E^2)$ ▷ 三角形类型 T_0 或 T_2
      else
        如果 $(S[uv], S[uw] = -1)$ 或 $(S[uv], S[uw] = +1)$ **则**
          $R_{unbal} \leftarrow R_{unbal} + 1/(p_V p_E^2)$ ▷ 三角形类型 T_0 或 T_2
        else
          $R_{bal} \leftarrow R_{bal} + 1/(p_V p_E^2)$ ▷ 三角形类型 T_1 或 T_3
  如果 $h_E(vw) = 1$ 且 $(h_V(v) = 1$ 或 $h_V(w) = 1)$ **则**
    $S[vw] \leftarrow \sigma_{vw}$
返回 $R_{bal}/(R_{bal} + R_{unbal})$

猜想 8. 与最先进的纯经典估计器相比，混合量子-经典三角平衡算法具有多项式空间优势。

该猜想的动机在于，混合算法由 3 个子程序组成，并且每个子程序相较于其纯经典对应算法都具有空间优势。然而，目前尚不清楚是否存在一种更优的纯经典算法来估计平衡，且该算法不依赖于显式计算平衡三角形和非平衡三角形。

5.2 分布式计算设置

现在，我们提出一种分布式计算框架，用于实现混合算法。回顾一下，分布式量子计算（DQC）旨在通过在多个量子设备之间利用量子和/或经典通信链路 [46]来扩展量子计算。标准 DQC 架构面临的一项核心挑战，是实现分布式量子操作，例如跨不同设备执行的量子门。相比之下，我们的框架不需要任何分布式量子操作。相反，每个小型量子设备独立运行，并将所得的经典输出传送至经典处理器进行后处理，以获得最终估计值。这种结构使所提出的算法成为一种完美可分布式 算法，其含义见参考文献 [24]。我们的方法属于易于并行化的 MapReduce 范式，如图 4 所示。下面，我们将介绍所需的量子和经典资源，并详细说明如何在这一分布式设置中实现混合算法。

5.2.1 计算资源与流程。 对于一个固定的符号图 G，其 n 个顶点上的真实三角平衡记为 $B_{\triangle}$。给定 $0 < \epsilon$，$\delta < 1$ 混合算法以精度 $\epsilon B_{\triangle}$ 估计 $B_{\triangle}$，其概率为 $1 - \delta$。固定常数 $N_Q = \Theta\left((km/T\epsilon)^2 \log \frac{1}{\delta}\right)$ 和 $N_C =$

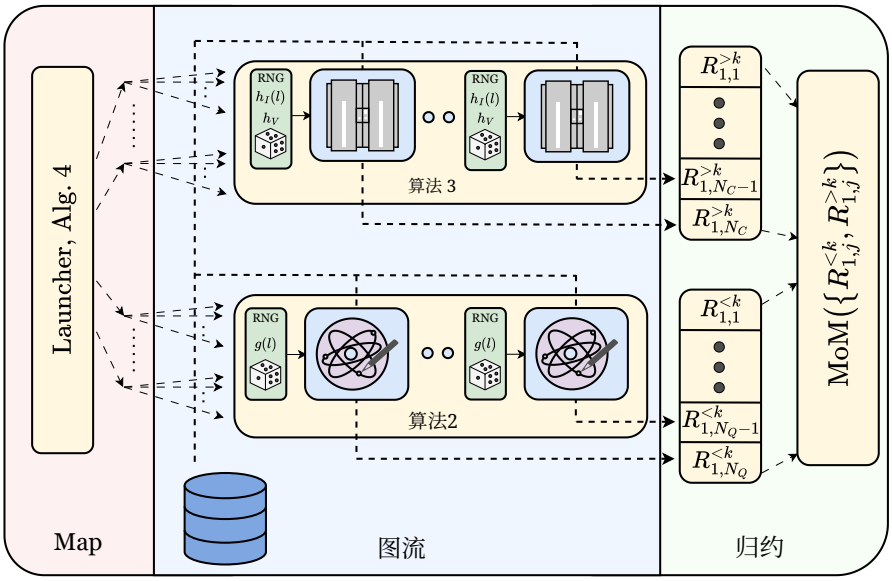

图4. **使用分布式资源进行带符号三角形估计的示意图**。在映射过程中，我们根据算法 4 为量子处理单元和中央处理器分配各自的初始（空）草图板。在图流过程中，每条边都会传递给这些分布式中央处理器和量子处理单元。在数据流的第 ℓ 步，根据哈希函数 $g(\ell)$、 $h_I(\ell)$ 和 h_V，每个中央处理器或量子处理单元都会执行一次查询（见图3）。每个中央处理器和量子处理单元都会异步终止并提供其估计值。在归约过程中，我们根据参数 δ 和 ϵ 计算中位数均值法（图中简称 MoM）。

$\Theta\left(\frac{m^{3/2}\Delta_E}{T\sqrt{k}\epsilon^2}\log\frac{1}{\delta}\right)$。该分布式计算配置需要 $3N_Q$ 个小型量子处理单元和 $3N_C$ 个中央处理器。此处，$\bar{m}$ 表示数据流中边数的近似值， T 表示 G 中三角形数量的近似值，以及 $k \approx T^{2/5}\Delta_E^{2/5}/m^{1/5}$.

(1) 量子资源：每个量子处理单元都配有一个经典寄存器，该寄存器针对每条入边 e 独立实现哈希函数 h_E，使得 $h_E(e) = 1$ 的概率为 $1/k$，否则为 0。这里有一个 $(2\lceil\log_2 n\rceil + 2)$ 量子比特的量子寄存器，用于实现草图板。具体而言，工作量子比特寄存器由 $2\lceil\log_2 n\rceil$ 个量子比特构成，另有两个单量子比特寄存器分别用于辅助量子比特，最后一个寄存器则存储数据流中入边的符号。每个量子处理单元还配有一个测量设备，用于按照算法3所述对量子草图板执行查询。(2) 经典资源：每个中央处理器都配有一个哈希函数 h_V，使得对于 G. 的每个顶点 u， $\mathbb{E}[h_V(u)] = 1/\sqrt{km}$。此外，每个中央处理器还配有关联的草图板存储寄存器，用于存储一条入边带符号边，并按照算法2所述进行概率更新。

现在我们讨论分布式计算环境的工作流程，如图 4 中的示意图所示。整个流程并行近似计算三个值： R_1、 R_3 和 R。输出 $(R_1 + R_3)/R$，即可得到 $\mathrm{B}_\triangle$ 的一个近似值。

中央经典协调器负责通过准备 $3N_Q$ 个量子处理单元和 $3N_C$ 个中央处理器来完成任务。每个处理器都需要独立的随机性，并负责计算上述三个值之一的估计值。粗略地说， N_Q 个量子处理单元和 N_C 个中央处理器共同参与近似值的计算

R_1，N_Q 个量子处理单元和 N_C 个中央处理器共同参与 R_3的近似计算，而 N_Q 个量子处理单元和 N_C 个中央处理器共同参与 R的近似计算。随着边以流式方式输入，它们会同时广播给每个处理器（量子和经典处理器）。所有设备（量子和经典设备）完成后，便会将各自的值报告给中央处理器，由中央处理器进行经典的中位数均值法后处理，以获得三个单独的估计值。

为了使这一过程更加精确，我们关注 R_1 近似（见图 4）。对于每个$j \in [N_C]$，一个中央处理器会独立生成 $R_1^{>k}{}_{,\,j}$ 并将其返回给中央协调器。令$N_{C,\,\epsilon} := \frac{m^{3/2}\Delta_E}{T_1\sqrt{k}\epsilon^2}$ 且 $N_{C,\,\delta} :=$ ，取 $\log\frac{1}{\delta}$ 使得$\overline{N_C} = N_{C,\,\epsilon}\cdot N_{C,\,\delta}$。我们将前 $N_{C,\,\epsilon}$ 个值分桶，以计算

$$\widehat{R_{1,1}^{>k}} := \mathsf{mean}\left(R_{1,1}^{>k},\ldots,R_{1,N_{C,\epsilon}}^{>k}\right). \tag{8}$$

接下来，通过分桶并对接下来的 $\left\{R_1^{>k}{}_{,\,N_{C,\,\epsilon}+1},\ \ldots,\ R_1^{>k}{}_{,\,2N_{C,\,\epsilon}}\right\}$ 个值取平均值，计算出 $\widehat{R_{1\,2}^{>k}}$。继续进行，直到得到平均值 $R_{1\,1}^{>k}{}_{,}$ ，…，$R_1^{>k}{}_{,\,N_{C,\,\delta}}$。由此，我们得到最终的经典估计值$T_1^{>k}$，即

$$R_1^{>k} := \mathsf{median}\left(\widehat{R_{1,1}^{>k}},\ldots,\widehat{R_{1,N_{C,\delta}}^{>k}}\right). \tag{9}$$

$R_1^{>k}$因此是对 $T_1^{>k}$ 的一个良好估计量。

对量子输出进行类似的后处理。首先，将负责估计 $R_1^{<k}$ 的 N_Q 个量子处理单元分成 $N_{Q,\,\epsilon} := (km/T_1\epsilon)^2$ 个桶，每个桶针对 $j \in [N_{Q,\delta}]$ 得出一个平均值 $R_1^{<k}{}_{,\,j}$，其中 $N_{Q,\,\delta} := \log\frac{1}{\delta}$。取这些值的中位数，得到 $R_1^{<k}$，它是对 $T_1^{<k}$ 的估计值。将 $R_1^{>k}$ 和 $R_1^{<k}$ 相加，可得到对 T_1 的良好估计值。并行重复这一过程，以获得 R_3 和 R 的估计值。最后，输出 $(R_1 + R_3)/R$，从而得到 $\mathrm{B}^H{}_{\triangle}$ 的估计值，其中 $\mathrm{B}_{\triangle}$。

5.3 数值模拟

5.3.1 模拟量子草图板。为了在真实图上测试混合算法 Alg. 5，我们需要实现量子操作。任何非平凡示例所需的空间都过于庞大，无法在当前设备上运行，因此我们采用模拟的方式。具体而言，我们使用一个经典草图，其操作遵循第 4.2 节中描述的概率。特别地，模拟草图包含三个方法：create、insert、query_边，用于模拟量子草图的概率行为。代码实现见附录 A。

5.3.2 结果。我们在一组额外随机分配了边符号的埃尔德什–雷尼图上运行算法5和算法6。这些图由两个概率参数生成：边概率 p_e 以及每条边的符号为正的概率 p_+。对于每组参数 $(n,\ p_e,\ p_+) \in \{10,\ 20,\ 30,\ 40,\ 50\} \times \{0.5,\ 0.75\} \times \{0.25,\ 0.5,\ 0.75\}$，我们创建了5个随机图。表1至表3展示了在这些实验中生成并使用的 $n \in \{30,\ 40,\ 50\}$个实例。为简洁起见，省略了$n \in \{10,\ 20\}$个实例，但算法在这些较小图上的表现与较大图相似。

每个算法均以误差参数 $\epsilon,\ \delta = 0.1$ 运行。对于给定的图，设 $\mathrm{B}^H{}_{\triangle}$ 和 $\mathrm{B}^C{}_{\triangle}$分别为算法 5 和算法 6 的输出，$\mathrm{B}_{\triangle}$ 为真实值。使用这些参数时，$\mathrm{B}^H{}_{\triangle}$ 和 $\mathrm{B}^C{}_{\triangle}$ 均以至少 0.9 的概率，将 $\mathrm{B}_{\triangle}$ 估计到误差不超过 $0.1\mathrm{B}_{\triangle}$。如图 5 所示，这两个估计量的表现均优于 $\epsilon = 0.1$ 误差阈值。此外，混合算法的表现略优于纯经典的有符号 JK 算法。

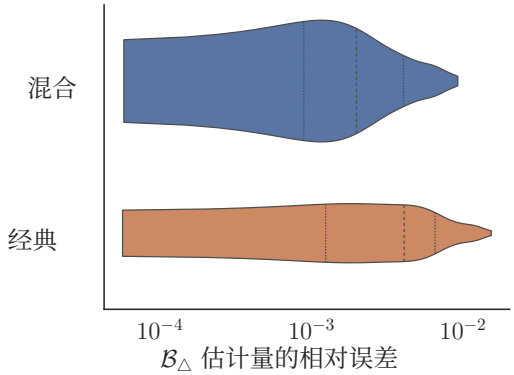

图 5. **比较经典与量子-经典三角平衡估计量性能的小提琴图**。比较算法 5 与算法 6 分别输出的 $\mathrm{B}^{H}{}_{\triangle}$ 和 $\mathrm{B}^{C}{}_{\triangle}$。这些算法在大小为 $n \in \{30, 40, 50\}$ 的随机埃尔德什–雷尼实例上进行测试。相对误差计算为 $|\mathrm{B}^{A}{}_{\triangle} - \mathrm{B}_{\triangle}|/\mathrm{B}_{\triangle}$，其中 $A \in \{H, C\}$。两种算法均采用精度参数 $\epsilon = 0.1$，且性能优于预期。混合算法的表现略优于纯经典算法，但方差确实更高。这是意料之中的，因为混合算法存在两个误差来源，一个来自量子估计器，另一个来自经典估计器。

6 结论

本文阐述了将用于计数无符号三角形的空间最优经典算法自然地扩展到有符号情形的方法。在这种情形下，我们需要对带符号网络流中可能出现的四种不同类型的带符号三角形进行计数。随后，我们阐述了一种混合量子-经典算法，该算法相较于经典算法具有多项式空间优势。

通过结构平衡理论的视角，近似统计带符号网络中不同类型的三角形，与许多重要的现实网络动态密切相关。具体而言，我们考虑估计带符号网络平衡程度的任务，这有助于理解社交网络、金融市场、生物系统等领域中的冲突或不和。

我们的研究表明，量子计算机可用于监测和解读大规模网络，并且在估计平衡等核心参数时，相较于最优的经典算法能够实现量子优势。

参考文献

[1] J. Kallaugher、O. Parekh 和 N. Voronova，《如何在不了解量子计算的情况下设计量子流处理算法》，2024年10月，arXiv:2410.18922。[2] D. Gavinsky、J. Kempe 和 R. d. Wolf，《布尔函数的量子与经典单向通信复杂度的指数级分离》，2006年7月，arXiv:quant-ph/0607174。[3] J. Kallaugher，《一种自然流处理问题中的量子优势》，载于*2021 IEEE* 第*62*届计算机科学基础年会（*FOCS*）。IEEE，2022年，第897–908页。[4] J. Kallaugher、O. Parekh 和 N. Voronova，《流模型中近似求解最大有向割的指数级量子空间优势》，2023年11月，arXiv:2311.14123。[5] Z. Bar-Yossef、R. Kumar 和 D. Sivakumar，《数据流中的三角形计数》，载于第*21*届 *ACM SIGMOD-SIGACT-SIGAI* 数据库系统原理研讨会（*PODS*）论文集。ACM，2002年，第134–141页。[6] L. S. Buriol、G. Frahling、S. Leonardi、A. Marchetti-Spaccamela 和 C. Sohler，《数据流中的三角形计数》，载于第*25*届 *ACM SIGMOD-SIGACT-SIGAI* 数据库系统原理研讨会（*PODS*）论文集。ACM，2006年，第253–262页。

表 1. **带符号的 Erdős–Rényi 图的数值结果 $n = 30$ 个节点**。加粗的数值表示该估计值更接近真实值 $\mathrm{B}_{\triangle} = (T_1 + T_3)/\sum_i T_i$。在此表中，$\mathrm{B}^{A}{}_{\triangle}$ 数值为简洁起见截断至三位小数，因此某些行中的数值可能看起来相同。

p_E	p_+	E_+	E_-	Δ_E	Δ_V	T_0	T_1	T_2	T_3	$\mathcal{B}_{\triangle}$	$\overline{\mathcal{B}}^{C}_{\triangle}$	$\overline{\mathcal{B}}^{H}_{\triangle}$
0.75	0.5	148	171	21	233	215	676	557	147	0.516	0.513	**0.517**
0.75	0.25	75	260	23	255	852	768	202	19	0.427	0.428	**0.428**
0.5	0.25	42	156	13	79	185	165	26	5	0.446	0.445	**0.445**
0.75	0.5	150	195	23	255	355	861	633	158	0.508	0.506	**0.508**
0.5	0.5	108	111	13	101	72	199	196	61	0.492	**0.492**	0.492
0.5	0.5	107	116	12	86	70	224	176	53	0.530	0.523	**0.530**
0.5	0.25	62	161	14	104	221	261	77	14	0.480	**0.479**	0.479
0.75	0.5	171	161	24	271	189	661	730	239	0.495	0.497	**0.494**
0.5	0.25	51	164	12	80	221	202	63	5	0.422	0.420	**0.421**
0.75	0.25	79	239	20	218	697	648	221	26	0.423	0.427	**0.427**
0.5	0.5	101	110	12	86	71	174	171	53	0.484	0.481	**0.482**
0.75	0.5	152	165	21	215	201	612	561	195	0.514	**0.514**	0.512
0.5	0.5	94	114	12	72	63	173	138	42	0.517	0.522	**0.520**
0.75	0.75	247	71	22	237	22	187	631	765	0.593	**0.593**	0.593
0.5	0.25	50	168	15	103	230	241	57	10	0.467	0.472	**0.464**
0.5	0.75	163	64	14	106	15	90	262	211	0.521	0.518	**0.523**
0.5	0.75	150	57	11	70	10	86	167	164	0.585	0.585	**0.585**
0.5	0.25	45	151	11	68	172	132	45	3	0.384	0.387	**0.386**
0.75	0.25	100	233	24	267	632	795	345	51	0.464	0.468	**0.465**
0.5	0.75	166	51	12	84	5	61	202	217	0.573	0.566	**0.570**
0.75	0.75	254	78	23	284	28	257	689	813	0.599	**0.599**	0.598
0.75	0.75	224	92	21	228	37	290	672	546	0.541	0.534	**0.543**
0.75	0.5	170	137	20	240	122	470	569	248	0.510	0.513	**0.508**
0.5	0.75	147	54	12	75	4	66	182	142	0.528	0.525	**0.529**
0.75	0.75	255	76	22	247	22	226	720	816	0.584	0.587	**0.583**
0.75	0.25	95	223	20	214	520	702	301	36	0.473	**0.474**	0.470
0.75	0.75	255	76	23	263	23	217	739	819	0.576	0.572	**0.574**
0.5	0.5	97	125	14	94	94	221	192	46	0.483	**0.483**	0.483
0.75	0.25	87	244	23	245	719	799	264	23	0.455	0.456	**0.456**
0.5	0.75	161	63	14	96	10	91	238	211	0.549	0.546	**0.549**

[7] M. Jha、C. Seshadhri 和 A. Pinar，“一种利用生日悖论估计传递性和三角形计数的空间高效流式算法”，载于 *ACM* 数据知识发现汇刊，第 9 卷，第 4 期，2015 年，第 1–21 页。[8] R. Jayaram 和 J. Kallaugher，“图流中三角形计数的最优算法”，载于 第 *53* 届 *ACM SIGACT* 计算理论年度研讨会论文集（*STOC*）。ACM，2021年，第 1283–1296 页。[9] J. Leskovec、D. Huttenlocher 和 J. Kleinberg，“社交媒体中的带符号网络”，载于 *SIGCHI* 计算系统中的人为因素会议论文集，2010 年，第 1361–1370 页。[10] G. Facchetti、G. Iacono 和 C. Altafini，“在大规模带符号社交网络中计算全局结构平衡”，美国国家科学院院刊，第 108 卷，第 52 期，第 20 953–20958 页，2011 年。[11] T. Antal、P. Krapivsky 和 S. Redner，“网络中的社会平衡动力学”，《物理评论 *E*》，第 72 卷，第 3 期，文章编号 036121，2005 年。

表 2. **带符号的 Erdős–Rényi 图的数值结果（含 $n = 40$ 个节点）**。粗体值表示该估计值更接近真实值 $\mathrm{B}_{\triangle} = (T_1 + T_3)/\sum_i T_i$。在此表中，为节省篇幅，$\mathrm{B}^{A}{}_{\triangle}$ 值仅保留三位小数，因此某些行中的数值看起来可能相同。

p_E	p_+	E_+	E_-	Δ_E	Δ_V	T_0	T_1	T_2	T_3	$\mathcal{B}_{\triangle}$	$\overline{\mathcal{B}}^{C}_{\triangle}$	$\overline{\mathcal{B}}^{H}_{\triangle}$
0.5	0.25	114	297	18	189	549	636	240	23	0.455	**0.453**	0.459
0.75	0.75	443	156	31	480	74	669	1901	1810	0.557	**0.557**	0.560
0.75	0.5	284	306	30	450	602	1641	1513	505	0.504	0.505	**0.503**
0.75	0.5	301	298	30	455	530	1681	1694	562	0.502	0.503	**0.503**
0.5	0.75	315	73	17	151	6	99	436	667	0.634	**0.635**	0.637
0.5	0.75	274	104	16	156	19	204	475	424	0.560	**0.560**	0.563
0.5	0.25	98	311	17	171	631	584	176	19	0.428	0.429	**0.429**
0.5	0.25	99	287	16	145	520	479	176	23	0.419	0.421	**0.419**
0.75	0.75	443	147	29	443	60	597	1796	1778	0.561	0.559	**0.561**
0.75	0.5	293	286	30	439	503	1522	1460	543	0.513	**0.512**	0.513
0.75	0.25	123	450	29	431	1898	1562	424	44	0.409	0.407	**0.408**
0.5	0.5	217	199	18	177	168	508	583	193	0.483	0.481	**0.482**
0.75	0.5	300	285	30	418	483	1502	1617	565	0.496	0.499	**0.496**
0.75	0.75	440	138	29	395	39	528	1677	1716	0.567	0.568	**0.566**
0.75	0.25	134	425	29	445	1587	1553	442	56	0.442	0.438	**0.444**
0.75	0.25	155	428	28	424	1625	1789	656	82	0.451	0.454	**0.448**
0.75	0.75	466	130	30	436	36	487	1775	2134	0.591	**0.591**	0.591
0.5	0.5	186	198	18	166	167	416	458	129	0.466	**0.466**	0.465
0.5	0.25	87	290	17	172	519	470	142	12	0.422	0.424	**0.424**
0.5	0.75	282	102	17	160	17	180	521	448	0.539	0.544	**0.542**
0.5	0.75	297	109	18	171	27	226	586	590	0.571	0.573	**0.570**
0.5	0.5	189	208	18	160	172	506	476	128	0.495	**0.493**	0.496
0.75	0.25	148	423	29	439	1584	1636	585	64	0.439	0.436	**0.438**
0.5	0.25	92	317	19	200	644	576	174	19	0.421	0.419	**0.421**
0.75	0.75	441	144	29	421	59	581	1743	1760	0.565	0.567	**0.564**
0.5	0.5	197	195	16	167	159	465	485	150	0.488	**0.489**	0.493
0.75	0.5	306	282	30	426	445	1545	1637	611	0.509	**0.509**	0.507
0.75	0.25	164	421	29	440	1528	1796	719	96	0.457	0.460	**0.457**
0.5	0.5	204	186	20	189	141	412	501	175	0.478	**0.478**	0.477
0.5	0.75	291	94	15	130	14	141	504	514	0.558	0.561	**0.556**

[12] S. Marvel、S. Strogatz 和 J. Kleinberg，“社会平衡能景观”，《物理评论快报》，第 103 卷，第 19 期，第 198701 页，2009 年。[13] V. Boginski、S. Butenko 和 P. Pardalos，“金融网络的统计分析”，《计算统计与数据分析》，第 48 卷，第 2 期，第 431–443 页，2005 年。[14] D. Kenett、M. Tumminello、A. Madi、G. Gur-Gershgoren、R. Mantegna 和 E. Ben-Jacob，“通过对股票市场进行偏相关分析揭示金融部门的主导性关联”，*PLOS ONE*，第 5 卷，第 12 期，第 e15032 页，2010 年。[15] L. Bargigli、G. di Iasio、L. Infante、F. Lillo 和 F. Pierobon，“银行间网络的多重结构”，《定量金融》，第 15 卷，第 4 期，第 673–691 页，2015 年。[16] P. Bartesaghi、F. Diaz-Diaz、R. Grassi 和 P. Uberti，“金融相关网络中的整体平衡与系统性风险”，《物理学*A*：统计力学及其应用》，第 130698 页，2025 年。[17] B. Adhikari，“用于投资组合优化的带符号网络模型”，*arXiv* 预印本 *arXiv:2510.05377*，2025 年。

表 3. **带符号的 Erdős–Rényi 图的数值结果，其中包含 $n = 50$ 个节点**。加粗的数值表示该估计值更接近真实值 $\mathrm{B}_{\triangle} = (T_1 + T_3)/\sum_i T_i$。在此表中，为简洁起见，$\mathrm{B}^{A}{}_{\triangle}$ 值仅保留三位小数，因此某些行中的数值看起来可能相同。

p_E	p_+	E_+	E_-	Δ_E	Δ_V	T_0	T_1	T_2	T_3	$\mathcal{B}_{\triangle}$	$\overline{\mathcal{B}}^{C}_{\triangle}$	$\overline{\mathcal{B}}^{H}_{\triangle}$
0.75	0.5	453	479	37	683	1251	3250	3184	957	0.487	**0.487**	0.487
0.5	0.5	301	297	21	229	271	819	894	304	0.491	0.494	**0.494**
0.5	0.25	137	438	21	245	887	838	236	31	0.436	**0.438**	0.438
0.5	0.75	444	160	20	217	39	348	1018	905	0.542	0.539	**0.543**
0.75	0.5	453	460	37	669	1063	3020	3008	975	0.495	**0.493**	0.499
0.5	0.25	152	456	22	232	988	1012	335	45	0.444	0.451	**0.444**
0.75	0.25	222	697	35	610	3580	3447	1080	116	0.433	0.428	**0.435**
0.75	0.5	441	473	33	610	1127	3191	2854	927	0.508	0.511	**0.510**
0.5	0.5	313	310	21	221	281	986	961	315	0.512	0.515	**0.509**
0.75	0.5	452	462	37	666	1034	3093	3039	925	0.497	0.502	**0.496**
0.5	0.75	455	154	22	261	19	354	1041	981	0.557	0.558	**0.557**
0.5	0.5	275	353	21	260	473	1104	839	220	0.502	0.505	**0.504**
0.75	0.25	234	689	34	648	3462	3519	1160	152	0.443	0.442	**0.443**
0.75	0.75	661	226	33	598	147	1038	3126	3120	0.560	**0.559**	0.559
0.75	0.25	229	703	39	726	3684	3615	1195	108	0.433	0.429	**0.434**
0.5	0.5	304	320	21	213	343	952	980	295	0.485	0.485	**0.485**
0.5	0.25	148	455	21	234	1021	995	311	37	0.437	0.443	**0.435**
0.75	0.5	449	472	37	706	1105	3179	3017	1001	0.503	**0.506**	0.508
0.5	0.75	435	164	23	286	47	413	1023	893	0.550	**0.550**	0.547
0.5	0.5	312	320	24	296	378	982	962	339	0.496	0.496	**0.496**
0.5	0.75	450	153	20	225	37	328	1012	973	0.554	0.556	**0.555**
0.5	0.25	134	466	21	220	1067	928	246	23	0.420	0.416	**0.419**
0.75	0.25	238	665	36	647	3191	3331	1202	142	0.442	0.444	**0.441**
0.5	0.25	155	457	20	252	1019	1017	336	39	0.438	0.438	**0.438**
0.75	0.25	230	684	39	731	3429	3433	1192	126	0.435	0.436	**0.435**
0.75	0.75	664	228	36	654	130	1091	3299	3075	0.549	0.549	**0.549**

[18] ——，《用于投资组合优化降维的带符号网络模型》，*arXiv* 预印本 *arXiv:2602 21362*，2026。.[19] E. Estrada，《带符号生物网络中的平衡》，*Journal of Theoretical Biology*，第 263 卷，第 4 期，第 556–561 页，2010 年。[20] A. Saade、F. Krzakala 和 L. Zdeborová，《带符号网络的谱聚类》，载于神经信息处理系统进展（*NeurIPS*），2014 年，第 1–9 页。[21] C. Altafini，《具有对抗性相互作用的网络中的共识问题》，*IEEE Transactions on Automatic Control*，第 58 卷，第 4 期，第 935–946 页，2013 年。[22] V. Traag、P. Van Dooren 和 P. DeLeenheer，《解释社会平衡与合作演化的动力学模型》，*PLOS ONE*，第 8 卷，第 4 期，文章编号 e60063，2013 年。[23] L. Meng、Y. Shao、L. Yuan、L. Lai、P. Cheng、X. Li、W. Yu、W. Zhang、X. Lin 和 J. Zhou，《大规模图上的分布式图算法综述》，*ACM Computing Surveys*，第 57 卷，第 2 期，第 1–39 页，2024 年。[24] M. Caleffi、M. Amoretti、D. Ferrari、J. Illiano、A. Manzalini 和 A. S. Cacciapuoti，《分布式量子计算：综述》，*Computer Networks*，第 254 卷，文章编号 110672，2024 年。[25] K. J. Ahn、S. Guha 和 A. McGregor，《图摘要：稀疏化、生成器与子图》，载于第 *31* 届 *ACM SIGMOD-SIGACT-SIGAI* 数据库系统原理研讨会（*PODS*）论文集。ACM，2012 年，第 5–14 页。[26] A. McGregor，《图流算法：综述》，*SIGMOD Record*，第 43 卷，第 1 期，第 9–20 页，2014 年。

[27] M. Kapralov，“一般图中的匹配流式算法”，载于第24届ACM-SIAM离散算法年会（SODA）论文集。SIAM，2013年，第167–183页。[28] G. Cormode和S. Muthukrishnan，“三角形计数与最后一个归约器的难题”，载于第20届国际万维网大会（WWW）论文集，2010年，第607–614页。[29] D. P. Woodruff，“草图作为数值线性代数的工具”，《理论计算机科学基础与趋势》，第10卷，第1–2期，第1–157页，2014年。[30] S. Muthukrishnan，《数据流：算法与应用》。Now出版社，2005年。[31] G. Malewicz、M. H. Austern、A. J. Bik、J. C. Dehnert、I. Horn、N. Leiser和G. Czajkowski，“Pregel：一种大规模图处理系统”，载于2010年ACM SIGMOD国际数据管理会议论文集，丛书：SIGMOD ’10。美国纽约州纽约市：计算机协会，2010年，第135–146页。 [在线]。网址：https://doi.org/ 10 1145 1807167 1807184 。/。D. Yan、[32] J. Cheng、Y. Lu和W. Ng，“分布式图计算中的消息缩减与负载均衡有效技术”，载于第24届国际万维网会议论文集，2015年，第1307–1317页。[33] Y. Low、J. Gonzalez、A. Kyrola、D. Bickson、C. Guestrin和J. M. Hellerstein，“分布式GraphLab：云端机器学习框架”，arXiv预印本 arXiv:1204.6078，2012年。[34] J. E. Gonzalez、Y. Low、H. Gu、D. Bickson和C. Guestrin，“PowerGraph：自然图上的分布式图并行计算”，载于第10届USENIX操作系统设计与实现研讨会（OSDI），2012年，第17–30页。[35] M. Han和K. Daudjee，“Giraphunchained：类Pregel图处理系统中的无屏障异步并行执行”，《VLDB基金会会刊》，第8卷，第9期，第950–961页，2015年。[36] S. Salihoglu和J. Widom，“GPS：一种图处理系统”，载于第25届国际科学与统计数据库管理会议论文集，2013年，第1–12页。[37] A. Roy、I. Mihailovic和W. Zwaenepoel，“X-Stream：使用流式分区的以边为中心的图处理”，载于第24届ACM操作系统原理研讨会论文集，2013年，第472–488页。[38] H. Zhu、L. He、M. Leeke和R. Mao，“WolfGraph：图形处理器上的以边为中心的图处理”，《未来一代计算机系统》，第111卷，第552–569页，2020年。[39] Y. Tian、A. Balmin、S. A. Corsten、S. Tatikonda和J. McPherson，“从‘像顶点一样思考’到‘像图一样思考’”，《VLDB基金会会刊》，第7卷，第3期，第193–204页，2013年。[40] Y. Simmhan、A. Kumbhare、C. Wickramaarachchi、S. Nagarkar、S. Ravi、C. Raghavendra和V. Prasanna，“GoFFish：一种面向大规模图分析的子图中心框架”，载于《欧洲并行处理会议》。施普林格，2014年，第451–462页。[41] D. Yan、J. Cheng、Y. Lu和W. Ng，“Blogel：一种面向真实世界图分布式计算的以块为中心的框架”，《VLDB基金会会刊》，第7卷，第14期，第1981–1992页，2014年。[42] F. Harary，“论带符号图的平衡概念”，《密歇根数学杂志》，第2卷，第2期，第143–146页，1953年。[43] T. Zaslavsky，“带符号图、增益图及相关领域的数学文献目录”，《组合学电子期刊》，第DS8–Dec页，2012年。[44] D. Cartwright和F. Harary，“结构平衡：海德理论的推广”，《心理学评论》，第63卷，第5期，第277–293页，1956年。[45] M. Brusco和D. Steinley，“通过挫折指数的松弛实现关系网络与带符号网络聚类”，《数学心理学杂志》，第90卷，第16–29页，2019年。[46] J. C. Boschero、N. M. Neumann、W. van der Schoot、T. Sijpesteijn和R. Wezeman，“分布式量子计算：应用与挑战”，载于《智能计算——计算会议论文集》。施普林格，2025年，第100–116页。[47] A. Shukla和P. Vedula，“一种高效制备均匀量子叠加态的量子算法”，《量子信息处理》，第23卷，第2期，2024年1月。[48] G. Lüders，“关于测量过程引起的状态变化”，《物理学年鉴》，第443卷，第5–8期，第322–328页，1950年。

量子草图的经典模拟

这里提供了实现量子草图功能的 Python 代码。

```
1来自 随机 导入 随机2  类 QuantumSketch：3 定义
目
（
```

```
        if (x 在 自身.项目) 和        (y 在 自身.项目):
            if 随机函数() < 2.0/长度(自身.项目):
                自身.项目 = []
                返回 +1
            否则:
                自身.项目.移除(x)
                自身.项目.移除(y)
                返回 0

        elif (x 在 自身.项目)      或(y 在自身.项目):
            if 随机函数() < 1.0/长度(自身.项目):
                自身.项目 = []
                if 随机函数() < 0.5:
                    返回 -1
                否则:
                    返回 +1
            否则:
                self.项目.丢弃(x)
                self.项目.discard(y)
                返回 0

        否则:
            返回 0

    def 更新(self, x, l):
        self.项目.remove(l)
        self.items.add(x)
```

清单 1. 量子草图功能的经典模拟的 Python 实现。

B 随机函数有符号图参数计算

这里我们更详细地描述为数值实验生成的图。我们生成随机埃尔德什–雷尼图，其中 p_e 表示边存在的概率， p_+ 表示给定边具有符号 +1的概率。随后，在实现和运行 T_1估计量时，可以使用这些理论值，因为运行算法本身所需的资源数量会用到这些值。对于真实世界图，可以根据历史数据估计这些值。

B.1 $|T_1|$ 期望与方差计算

引理 9.

$$\mathbb{E}[|T_1|] = 3\binom{n}{3}p_e^3 p_+(1-p_+)^2, \tag{10}$$

$$\mathrm{Var}(|T_1|) = 3\binom{n}{3}p_e^3 p_+(1-p_+)^2(1-3p_e^3 p_+(1-p_+)^2) \tag{11}$$

$$+ 2\binom{n}{2}\binom{n-2}{2}\left(p_e^5 p_+(1-p_+)^3(1+3p_+) - 9p_e^6 p_+(1-p_+)^4\right) \tag{12}$$

证明。考虑三个不同的顶点 u， v， $w \in V$。它们构成三角形的概率为 p_e^3。令I_{uvw}为指示变量，当 uvw是一个 T_1时，其值为 1。边恰好为一条 -1和两条 +1的概率为 $p_+(1-p_+)^2$。这种情况有三种可能。因此， $\mathbb{P}[I_{uvw}=1] = 3p_+(1-p_+)^2$。注意到这样的三元组共有 $\binom{n}{3}$ 个，我们得到

$$\mathbb{E}[|T_1|] = 3\binom{n}{3}p_e^3 p_+(1-p_+)^2 \tag{13}$$

接下来讨论方差。令 $p_\tau := \mathbb{P}[I_\tau]$ 表示一个三元组 τ。方差可计算为

$$\mathrm{Var}(|T_1|) = \sum_{\tau}\mathrm{Var}(I_\tau) + 2\sum_{\tau<\tau'}\mathrm{Cov}(I_\tau, I_{\tau'}) \tag{14}$$

由于 I_τ 是伯努利分布，我们有 $\mathrm{Var}(I_\tau) = p_\tau(1-p_\tau^2)$。对于协方差，设 uvw 和 xyz 是两个三角形。我们需要了解它们共享顶点时会发生什么。

- 当 $\{u,\ v,\ w\} \cap \{x,\ y,\ z\} = \emptyset$ 时，这些三角形相互独立，协方差为 0。
- 当 $|\{u,\ v,\ w\} \cap \{x,\ y,\ z\}| = 1$ 时，这些三角形相互独立，协方差为 0。
- 假设 $|\{u,\ v,\ w\} \cap \{x,\ y,\ z\}| = 1$。不失一般性，设它们共享边（$u,\ v$）和$w \neq z$。如果 $\sigma_{uv} = -1$，那么其余 4 条边都必须具有 $+1$ 符号，这种情况发生的概率为 p_+^4。另一方面，如果 $\sigma_{uv} = -1$，那么 $\sigma_{uw},\ \sigma_{vw}$ 中一个必须为 $+1$，另一个必须为 -1，$\sigma_{uz},\ \sigma_{vz}$ 也同样如此。每种情况发生的概率为 $p_+^2(1-p_+)^2$。综合起来，我们得到

$$p_+(1-p_+)^4 + 4p_+^2(1-p_+)^3 = p_+(1-p_+)^3(1+3p_+). \tag{15}$$

- W当 $|\{u、\ v、\ w\} \cap \{x、\ y、\ z\}| = 3$ 时，这些三角形相同，且协方差为 0。

唯一非零的情况是它们共享一条边，此时我们可以计算

$$\mathbb{P}[I_\tau, I_{\tau'} = 1] = p_e^5 p_+(1-p_+)^3(1+3p_+) \tag{16}$$

$$\Rightarrow \mathrm{Cov}(I_\tau, I_{\tau'}) = p_e^5 p_+(1-p_+)^3(1+3p_+) - p_\tau^2 \tag{17}$$

$$= p_e^5 p_+(1-p_+)^3(1+3p_+) - 9p_e^6 p_+^2(1-p_+)^4. \tag{18}$$

注意，三角形对共享一条边的方式有 $\binom{n}{2}\binom{n-2}{2}$ 种。现在我们可以计算完整的方差:

$$\mathrm{Var}(|T_1|) = \binom{n}{3}p_{uvw}(1-p_{uvw}) + 2\binom{n}{2}\binom{n-2}{2}[p_e^5 p_+(1-p_+)^3(1+3p_+) - p_{uvw}^2] \tag{19}$$

$$= \begin{aligned} &3\binom{n}{3}p_e^3 p_+(1-p_+)^2(1-3p_e^3 p_+(1-p_+)^2) \\ &+ 2\binom{n}{2}\binom{n-2}{2}\left(p_e^5 p_+(1-p_+)^3(1+3p_+) - 9p_e^6 p_+(1-p_+)^4\right) \end{aligned} \tag{20}$$

□

B.2　Δ_{E_1} 期望计算

接下来，我们想了解 Δ_{E_1}，即共享一条边的三角形的最大数量。设 Y_e 为与边 e 共享 T_1 的数量。那么 $\Delta_{E_1} = \max_e Y_e$。因此，我们通过 Y_e 研究 Δ_{E_1}。

固定边 $e = (u,\ v)$。我们还可以定义指示变量 Y_{uvw}：当 uvw构成一个 T_1时取值为 1。 uw和 vw 同时为边的概率是 p_e^2。如果 $\sigma_{uv} = +1$，则另外两个符号都必须是 -1，其发生概率为 $(1-p_+)^2$。否则，其中一条边为 $+1$，另一条为 -1。这两种情况各自发生的概率为 $p_+(1-p_+)$，且有 2 种可能。因此:

$$\mathbb{P}[Y_{uvw}=1] = p_e^2\left[\overbrace{p_+}^{\sigma_{uv}=+1}(1-p_+)^2 + \overbrace{(1-p_+)}^{\sigma_{uv}=-1}\cdot 2p_+(1-p_+)\right] \tag{21}$$

$$= 3p_e^2 p_+(1-p_+)^2 \tag{22}$$

令 $q_{uvw} := \mathbb{P}[Y_{uvw}=1]$。由于边及其符号相互独立，$Y_{uv} \sim Bin(n-2,\ q_{uvw})$，由此得到

$$\mathbb{E}[Y_{uv}] = (n-2)q_{uvw}, \quad \mathrm{Var}(Y_{uv}) = (n-2)q_{uvw}(1-q_{uvw}). \tag{23}$$

接下来，令 M 为图中边数的随机变量。这是另一个大小为 $\binom{n}{2}$、成功概率为 p_e的二项随机变量。特别地，$\mathbb{E}[M] \approx \frac{n^2}{2}p_e$。

二项随机变量是次高斯的，满足

$$\mathbb{P}\Big[Y_e - \mathbb{E}[Y_e] > t\Big] \le \exp\Big(-t^2/2\,\mathrm{Var}(Y_e)\Big) \tag{24}$$

因此，根据简单的并集界：

$$\mathbb{P}\Big[\max_{i=1,\dots,M} Y_i \ge \mathbb{E}[Y_i] + t\Big] = \mathbb{P}\Big[\bigcup_i \{Y_i - \mathbb{E}[Y_i] \ge t\}\Big] \le \sum_i \mathbb{P}\Big[Y_e - \mathbb{E}[Y_e] > t\Big] \tag{25}$$

$$\le M\exp\Big(-t^2/2\,\mathrm{Var}(Y_e)\Big). \tag{26}$$

令 $t = \sqrt{2\,\mathrm{Var}(Y_e)\log M}$，即可得到如下结论

$$\mathbb{E}\big[\Delta_{E_1}\big] \lesssim (n-2)q_{uvw} + \sqrt{2(n-2)q_{uvw}(1-q_{uvw})\log(n^2p_e)}. \tag{27}$$

C 量子流式原语

本节概述算法 3 中使用的量子流式范式。为方便起见，我们重新命名了参考文献 [1]中使用的一些方法；有关这一范式的更一般性论述，请参阅该文稿。

C.1 量子寄存器

设 n和 m分别为图中的顶点数和边数。草图的符号 全集为 $U = \{(u,v) \times \{+,-\} :\ u, v \in [n]\} \cup \{(\ell,\xi) :\ \ell \in [2m]\}$，其中 ξ表示一个暂存符号。为具体说明并阐释其底层逻辑，我们考虑 n 为 2 的幂的情形。此时，波函数态可分解为：

$$|Q\rangle = \sum_{j=0}^{n-1}\sum_{k=0}^{n-1}\sum_{s=0}^{1}\sum_{t=0}^{1}\alpha_{jkst}\underbrace{|j\rangle_{\log(n)}\otimes|k\rangle_{\log(n)}}_{\text{index register}}\otimes\underbrace{|s\rangle_1}_{\text{sign register}}\otimes\underbrace{|t\rangle_1}_{\text{scratch register}}, \tag{28}$$

其中，状态 $|j\rangle_{\log(n)}$ 是在计算基下由 $\log(n)$ 个量子比特构成的逐位分解 $j \in [n]$，其状态归一化为 $\sum_{jkst}\alpha_{jkst}\alpha_{jkst} = 1$。

我们展示了符号宇宙中的每个元素如何映射到量子波函数的一个具体基元素。我们将辅助状态定义为

$$|\ell, \xi\rangle \equiv |q\rangle_{\log(n)} |r\rangle_{\log(n)} |0\rangle_1 |0\rangle_1 , \tag{29}$$

其中 $\ell = q\,n + r$，使得商为 $q \in [n]$，余数为 $r \in [n]$。

我们将边状态定义为:

$$|u, v, \sigma_{uv}\rangle \equiv |u\rangle_{\log(n)} |v\rangle_{\log(n)} |\sigma_{uv}\rangle_1 |1\rangle_1 \tag{30}$$

在如图3所述的所有量子操作均无噪声地执行的情况下，假设算法尚未退出，则在任意时间戳 ℓ结束时，系统状态可由以下引理描述。

引理10 （草图板状态：处理 ℓ条边后）。 对于所有 $\ell = 0, \ldots, m-1$，算法处理完ℓ 条边后，若尚未终止，则底层集合 Qs_ℓ 为

$$\mathcal{S}_\ell := \{(j, \xi) : j = 2\ell, \ldots, 2m-1\} \cup \mathcal{K}_\ell \tag{31}$$

其中

$$\mathcal{K}_\ell := \left\{ (u, v, \sigma_{uv}) : \begin{array}{c} \Big(\exists i \in [\ell] \ s.t. \ (u, v, \sigma_{uv}) = (e_i, \sigma_i)\Big) \bigwedge \\ \Big(\underset{j=i+1,\ldots,\ell}{\forall} \Big(g(j) = 0\Big) \vee \Big(v \notin e_j\Big) \vee \Big(\big(\sigma_j = +\big) \wedge \big(\sigma_{uv} = +\big) \Big) \Big) \end{array} \right\} \tag{32}$$

其中 (e_i, σ_i) 表示数据流中的第 i^{th} 条带符号边。$\mathcal{S}_\ell$ 的大小为 $|\mathcal{S}_\ell| = 2m - 2\ell + |\mathcal{K}_\ell|$。

具体而言，在处理数据流中的 ℓ 条边后，如果算法尚未返回，则草图的量子态为

$$\left|Q_{S_\ell}\right\rangle = \frac{1}{\sqrt{2m - 2\ell + |\mathcal{K}_\ell|}} \left(\sum_{j=2\ell}^{2m-1} |j, \xi\rangle + \sum_{(u,v,\sigma_{uv}) \in \mathcal{K}_\ell} |u, v, \sigma_{uv}\rangle \right) \tag{33}$$

集合 K_ℓ 包含我们已见到且尚未删除的所有边。假设 (uv, σ_{uv}) 是一条在 ℓ^{th} 次更新之前某个时刻处理过的边。考虑时间步 $j = i+1, \ldots, \ell$。当且仅当以下条件成立时，(uv, σ_{uv}) 仍保留在草图中:

(1) $g(j) = 0$，因此完全不会进行测量。(2) v 不参与 j^{th} 条边，因此没有被删除的可能。(3) v 位于 j^{th} 条边中，但两者均为 $\sigma_{uv}, \sigma_j = +1$。这是因为测量集合得到了 $\perp$，但由于 $\sigma_j = +1$，我们没有删除 $+1$ 邻居。

有关该引理的形式化证明，请参见 D.1节。

C.2 测量算子

这里我们介绍用于实现算法 3 所用的查询_边调用的投影值测量（PVM）。回想一下，量子草图是至多 $2m$ 个计算基态的叠加。非零振幅的态可以用两种不同的方式描述。第一种是态 $|t, 0, 0\rangle$，其中 $t \in [2m]$，而第三个寄存器表示这是一个“暂存”态。第二种是类似 $|v, w, \pm, 1\rangle$ 的态，其中前两个寄存器 $v, w \in V$ 是顶点，第三个寄存器是vw的符号 σ_{vw}，第四个寄存器表示“激活”状态。这样，我们就可以查询处于激活状态的态，而无需担心 $t = (v, w)$（作为比特串）之间的冲突。

考虑我们正在处理带有符号 $+1$ 的边 vw，从而进入**步骤 8**的情形。我们希望查询草图以获取楔形 uv, uw 使得两条边都具有 -1 符号，从而与 uv组成一个 T_1 三角形。这通过调用查询_edges($(u, v, -)$, $(u, w, -)$)来实现

在算法 3 的**步骤 8** 中。以下三个算子构成了该操作 $\mathrm{M}_{uv-,\ uw-}$ 的投影值测量=$\{O^{+1}_{uv-,\ uw-},\ O^{-1}_{uv-,\ uw-},\ O^{\perp}_{uv-,\ uw-}\}$:

$$O^{+1}_{uv-,uw-} := \frac{1}{2}\Big(|u,v,-\rangle + |u,w,-\rangle\Big)\Big(\langle u,v,-| + \langle u,w,-|\Big), \tag{34}$$

$$O^{-1}_{uv-,uw-} := \frac{1}{2}\Big(|u,v,-\rangle - |u,w,-\rangle\Big)\Big(\langle u,v,-| - \langle u,w,-|\Big), \tag{35}$$

$$O^{\perp}_{uv-,uw-} := I - O^{+1}_{uv-,uw-} - O^{-1}_{uv-,uw-}. \tag{36}$$

可观测量 $O^{+1}_{uv-,\ uw-}$，$O^{-1}_{uv-,\ uw-}$ 是投影到两个类贝尔态上的秩-1投影算子 $\frac{1}{\sqrt{2}}(|u,\ v,\ -\rangle \pm |u,\ \overline{w},\ -\rangle)$。

考虑这样一种情形：我们正在处理带符号 -1的边 vw，这对应于**步骤 11-13**。我们希望查询草图以获取一个楔形 $uv,\ uw$，使其中一条边具有 $+1$符号，另一条边则具有 -1，从而构成一个三角形 T_1。这通过算法 3 的**步骤 11** 中的查询_edges($(u,\ v,\ +)$，$(u,\ w,\ -)$)调用来实现。以下三个算子构成此操作的投影值测量 $\mathrm{M}_{uv+,\ uw-}$:=$\{O^{+1}_{uv+,\ uw-},\ O^{-1}_{uv+,\ uw-},\ O^{\perp}_{uv+,\ uw-}\}$:

$$O^{+1}_{uv+,uw-} := \frac{1}{2}\Big(|u,v,+\rangle + |u,w,-\rangle\Big)\Big(\langle u,v,+| + \langle u,w,-|\Big), \tag{37}$$

$$O^{-1}_{uv+,uw-} := \frac{1}{2}\Big(|u,v,+\rangle - |u,w,-\rangle\Big)\Big(\langle u,v,+| - \langle u,w,-|\Big), \tag{38}$$

$$O^{\perp}_{uv+,uw-} := I - O^{+1}_{uv+,uw-} - O^{-1}_{uv+,uw-}. \tag{39}$$

上标$O^{\pm1}_{uv+,uw-}$ 表示这是对叠加态 $\frac{1}{\sqrt{2}}(|u,v,+\rangle \pm |u,w,-\rangle)$的投影。将第四个寄存器设为 1，是为子只查询“活跃”数据态；特别是，不 查询辅助态。算法 3 的**第 13 步** 中的投影值测量类似，$\mathrm{M}_{uv-,uw+} := \{O^{+1}_{uv-,uw+}, O^{-1}_{uv-,uw+}, O^{\perp}_{uv-,uw+}\}$:

$$O^{+1}_{uv-,uw+} := \frac{1}{2}\Big(|u,v,-\rangle + |u,w,+\rangle\Big)\Big(\langle u,v,-| + \langle u,w,+|\Big), \tag{40}$$

$$O^{-1}_{uv-,uw+} := \frac{1}{2}\Big(|u,v,-\rangle - |u,w,+\rangle\Big)\Big(\langle u,v,-| - \langle u,w,+|\Big), \tag{41}$$

$$O^{\perp}_{uv-,uw+} := I - O^{+1}_{uv-,uw+} - O^{-1}_{uv-,uw+}. \tag{42}$$

与第 8 步的投影值测量类似，$O^{\pm1}_{uv-,uw+}$ 分别是对叠加态 $\frac{1}{\sqrt{2}}(|u,v,-\rangle \pm |u,w,+\rangle)$的投影。

C.3 电路基本操作

现在我们解释算法 3 所需的三个量子子电路：创建,插入，以及查询_边。

这 3 个量子操作详述如下。

*C.3.1*创建。 该方法通过创建 $2m$ 个计算基态的叠加来实例化该草图。这里，$\mathrm{S}_0 := \{(j,\ \xi) :\ j = 0,\ \ldots,\ 2m-1\}$，并且我们有

$$\left|Q_{S_0}\right\rangle = \frac{1}{\sqrt{2m}} \sum_{j=0}^{2m-1} |j,\xi\rangle \tag{43}$$

当 m 是 2 的幂时，只需使用哈达玛门即可完成；若不是，也仍可高效完成 [47]。

C.3.2 插入$((vw,\sigma_{vw})\ \ell,\mathrm{Q}_{\mathcal{S}})$。将 (vw,σ_{vw}) 存入草图 S。该操作通过将两个辅助态旋转为 $|v,w,\sigma_{vw}$ 和 $|w,v,\sigma_{vw}$ 来“使用”它们。

假设我们处于第 ℓ 步。我们执行置换$\pi_{\ell,\ vw,\ \sigma_{vw}}$，该置换交换：

- 将 $(2\ell,\ \xi)$ 与 $(v,\ w,\ \sigma_{vw})$ 互换
- 将 $(2\ell+1,\ \xi)$ 与 $(w,\ v,\ \sigma_{vw})$ 互换

由于我们只处理一次边 $\{v,\ w\}$，因此这次交换的效果是将状态$|2\ell,\ \xi$映射为状态 $|v,\ w,\ \sigma_{vw}$，并将状态 $|2\ell+1,\ \xi$映射为状态 $|w,\ v,\ \sigma_{vw}$，这等价于以下映射

$$|v,w,\sigma_{vw}\rangle\langle 2\ell,\xi| \equiv |v\rangle\langle q_{2\ell}|\otimes|w\rangle\langle r_{2\ell}|\otimes|\sigma_{vw}\rangle\langle 0|\otimes|1\rangle\langle 0|, \tag{44}$$

$$|w,v,\sigma_{vw}\rangle\langle 2\ell+1,\xi| \equiv |w\rangle\langle q_{2\ell+1}|\otimes|v\rangle\langle r_{2\ell+1}|\otimes|\sigma_{vw}\rangle\langle 0|\otimes|1\rangle\langle 0|, \tag{45}$$

其中 $2\ell=q_{2\ell}\,n+r_{2\ell}$和 $2\ell+1=q_{2\ell+1}\,n+r_{2\ell+1}$。

此时的整个酉算子（以免扰动状态的其余部分）在所有其他状态上都只是恒等算子：

$$\begin{aligned} U_{\pi_{\ell,vw,\sigma_{vw}}} = {} & I-|2\ell,\xi\rangle\langle 2\ell,\xi|-|v,w,\sigma_{vw}\rangle\langle v,w,\sigma_{vw}| \\ & +|v,w,\sigma_{vw}\rangle\langle 2\ell,\xi|+|2\ell,\xi\rangle\langle v,w,\sigma_{vw}| \\ & -|2\ell+1,\xi\rangle\langle 2\ell+1,\xi|-|w,v,\sigma_{uv}\rangle\langle w,v,\sigma_{wv}| \\ & +|v,w,\sigma_{vw}\rangle\langle 2\ell+1,\xi|+|2\ell+1,\xi\rangle\langle v,w,\sigma_{vw}| \end{aligned} \tag{46}$$

C.3.3 查询_边。操作 query_edges$((wu,\tau_{wu}),(wv,\tau_{wv}))$ 向 S 查询一个楔形 (wu,wv)，使其与 uv（以及边信息）共同构成一个三角形。这通过对 $\mathrm{Q}_{\mathcal{S}}$ 进行测量来完成。可能有三种情形：(I) 两条带符号边都在当前草图中，(II) 其中一条在草图中，(III) 两条都不在草图中。根据当前适用的情形，以下说明可能的结果及其对应概率。

(I) (a) 以 $2/|\mathrm{S}|$ 的概率销毁 $\mathrm{Q}_{\mathcal{S}}$ 并返回 $+1$。(b) 以 $1-2/|\mathrm{S}|$ 的概率，将草图更新为 $\mathrm{S}'=\mathrm{S}\setminus\{(wu,\tau_{wu}),(wv,\tau_{wv})\}$，并将 $\mathrm{Q}_{\mathcal{S}}$ 更新为 $\mathrm{Q}_{\mathcal{S}'}$。返回 $\perp$。(II) (a) 以 $1/|\mathrm{S}|$ 的概率销毁 $\mathrm{Q}_{\mathcal{S}}$，并以相同概率返回 $\{+1,-1\}$。(b) 以 $1-1/|\mathrm{S}|$ 的概率，将草图更新为 $\mathrm{S}'=\mathrm{S}\setminus\{(wu,\tau_{wu}),(wv,\tau_{wv})\}$，并将 $\mathrm{Q}_{\mathcal{S}}$ 更新为 $\mathrm{Q}_{\mathcal{S}'}$。返回 $\perp$。

（III）保持 $\mathrm{Q}_{\mathcal{S}}$ 不变并返回 $\perp$

如 C.2 节所述，这种查询方法通过对量子态 $|\mathrm{Q}_{\mathcal{S}}$ 进行测量来实现。下面我们以测量集合 $\mathrm{M}_{wu-,wv+}=\{O^{+1}_{wu-,wv+},O^{-1}_{wu-,wv+},O^{\perp}_{wu-,wv+}\}$ 为例说明步骤，不过类似的逻辑也可用于 $\mathrm{M}_{wu-,wv-},\mathrm{M}_{wu+,wv-}$。

在进行测量时，系统的波函数为

$$|Q_{\mathcal{S}}\rangle=\frac{1}{\sqrt{|\mathcal{S}|}}\left(\sum_{j=2\ell}^{2m-1}|j,\xi\rangle+\sum_{(u,v,\sigma_{uv})\in\mathcal{S}}|u,v,\sigma_{uv}\rangle\right) \tag{47}$$

（I）（a）以 $2/|\mathrm{S}|$ 的概率销毁 $\mathrm{Q}_{\mathcal{S}}$ 并返回 $+1$。

(b) 以 $1-2/|\mathrm{S}|$ 的概率，将草图更新为 $\mathrm{S}'=\mathrm{S}\setminus\{(wu,\ \tau_{wu}),\ (wv,\ \tau_{wv})\}$，并将 $\mathrm{Q}_{\mathcal{S}}$ 更新为 $\mathrm{Q}_{\mathcal{S}'}$。返回 $\perp$。

在情形 I 中，$(w,\ u,\ -)$，$(w,\ v,\ +) \in \mathsf{S}$。然后

$$\langle Q_{\mathcal{S}}|\, O^{+1}_{wu-,wv+}\, |Q_{\mathcal{S}}\rangle = \langle Q_{\mathcal{S}}|\, \frac{1}{2}\Big(\, |w,u,-\rangle + |w,v,+\rangle\,\Big)\Big(\,\langle w,u,-| + \langle w,v,+|\,\Big)\, |Q_{\mathcal{S}}\rangle \tag{48}$$

$$= \frac{1}{2}\left(\frac{1}{\sqrt{|\mathcal{S}|}} + \frac{1}{\sqrt{|\mathcal{S}|}}\right)\left(\frac{1}{\sqrt{|\mathcal{S}|}} + \frac{1}{\sqrt{|\mathcal{S}|}}\right) \tag{49}$$

$$= \frac{2}{|\mathcal{S}|}, \tag{50}$$

and

$$\langle Q_{\mathcal{S}}|\, O^{-1}_{wu-,wv+}\, |Q_{\mathcal{S}}\rangle = \langle Q_{\mathcal{S}}|\, \frac{1}{2}\Big(\, |w,u,-\rangle - |w,v,+\rangle\,\Big)\Big(\,\langle w,u,-| - \langle w,v,+|\,\Big)\, |Q_{\mathcal{S}}\rangle \tag{51}$$

$$= \frac{1}{2}\left(\frac{1}{\sqrt{|\mathcal{S}|}} - \frac{1}{\sqrt{|\mathcal{S}|}}\right)\left(\frac{1}{\sqrt{|\mathcal{S}|}} - \frac{1}{\sqrt{|\mathcal{S}|}}\right) \tag{52}$$

$$= 0, \tag{53}$$

and

$$\langle Q_{\mathcal{S}}|\, O^{\perp}_{wu-,wv+}\, |Q_{\mathcal{S}}\rangle = \langle Q_{\mathcal{S}}|\, I - O^{+1}_{wu-,wv+} - O^{-1}_{wu-,wv+}\, |Q_{\mathcal{S}}\rangle \tag{54}$$

$$= 1 - \frac{2}{|\mathcal{S}|}. \tag{55}$$

在$O^{+}_{wu-,wv+}$ 被观测到的情况下，波函数明确地坍缩到一维空间。在$O^{\perp}_{wu-,wv+}$ 被观测到的情况下，波函数由吕德尔测量后更新规则 [48]给出：

$$\frac{O^{\perp}_{wu-,wv+}\, |Q_{\mathcal{S}}\rangle}{\| O^{\perp}_{wu-,wv+}\, |Q_{\mathcal{S}}\rangle\, \|} = \frac{1}{\| O^{\perp}_{wu-,wv+}\, |Q_{\mathcal{S}}\rangle\, \|}\, \frac{1}{\sqrt{|\mathcal{S}|}}\left(\sum_{j=2\ell}^{2m-1} |j,\xi\rangle + \sum_{(u,v,\sigma_{uv})\in\mathcal{S}\setminus\{(w,u,-),(w,v,+)\}} |u,v,\sigma_{uv}\rangle\right) \tag{56}$$

$$= \frac{\sqrt{|\mathcal{S}|}}{\sqrt{|\mathcal{S}|-2}}\, \frac{1}{\sqrt{|\mathcal{S}|}}\left(\sum_{j=2\ell}^{2m-1} |j,\xi\rangle + \sum_{(u,v,\sigma_{uv})\in\mathcal{S}\setminus\{(w,u,-),(w,v,+)\}} |u,v,\sigma_{uv}\rangle\right) \tag{57}$$

$$= \frac{1}{\sqrt{|\mathcal{S}|-2}}\left(\sum_{j=2\ell}^{2m-1} |j,\xi\rangle + \sum_{(u,v,\sigma_{uv})\in\mathcal{S}\setminus\{(w,u,-),(w,v,+)\}} |u,v,\sigma_{uv}\rangle\right) \tag{58}$$

(II) (a) 以 $1/|\mathsf{S}|$ 的概率销毁 $\mathsf{Q}_{\mathcal{S}}$，并以相等的概率返回 $\{+1,-1\}$。

(b) 以 $1-1/|\mathsf{S}|$ 的概率，将草图更新为 $\mathsf{S}' = \mathsf{S}\setminus\{(wu,\tau_{wu}),(wv,\tau_{wv})\}$，并将 $\mathsf{Q}_{\mathcal{S}}$ 更新为 $\mathsf{Q}_{\mathcal{S}'}$。返回 $\perp$。

在情形 II 中，$(w,\ u,\ -)$，$(w,\ v,\ +)$ 中只有一个属于 S。于是

$$\langle Q_{\mathcal{S}}|\, O^{+1}_{wu-,wv+}\, |Q_{\mathcal{S}}\rangle = \langle Q_{\mathcal{S}}|\, \frac{1}{2}\Big(\, |w,u,-\rangle + |w,v,+\rangle\,\Big)\Big(\,\langle w,u,-| + \langle w,v,+|\,\Big)\, |Q_{\mathcal{S}}\rangle \tag{59}$$

$$= \frac{1}{2}\left(\frac{1}{\sqrt{|\mathcal{S}|}}\right)\left(\frac{1}{\sqrt{|\mathcal{S}|}}\right) \tag{60}$$

$$= \frac{1}{2|\mathcal{S}|}, \tag{61}$$

and

$$\langle Q_{\mathcal{S}}|\, O^{-1}_{wu-,wv+}\,|Q_{\mathcal{S}}\rangle = \langle Q_{\mathcal{S}}|\, \frac{1}{2}\Big(\,|w,u,-\rangle - |w,v,+\rangle\,\Big)\Big(\,\langle w,u,-| - \langle w,v,+|\,\Big)\,|Q_{\mathcal{S}}\rangle \tag{62}$$

$$= \frac{1}{2}\left(\pm\frac{1}{\sqrt{|\mathcal{S}|}}\right)\left(\pm\frac{1}{\sqrt{|\mathcal{S}|}}\right) \tag{63}$$

$$= \frac{1}{2|\mathcal{S}|}, \tag{64}$$

and

$$\langle Q_{\mathcal{S}}|\, O^{\perp}_{wu-,wv+}\,|Q_{\mathcal{S}}\rangle = \langle Q_{\mathcal{S}}|\, I - O^{+1}_{wu-,wv+} - O^{-1}_{wu-,wv+}\,|Q_{\mathcal{S}}\rangle \tag{65}$$

$$= 1 - \frac{1}{|\mathcal{S}|}. \tag{66}$$

与情形 I 类似，如果我们观测到 $O^{+1}_{wu-,\ wv+}$，$O^{-1}_{wu-,\ wv+}$，状态就明确坍缩到了一个一维子空间。在观测到情形 $O^{\perp}_{wu-,\ wv+}$ 时，按照情形 I 的相同逻辑，我们可以得出结论：S 中存在的元素 $(w,\ u,\ -)$ 或 $(w,\ v,\ +)$ 已被删除。

(III) 保持 $\mathrm{Q}_{\mathcal{S}}$ 不变并返回 $\perp$。在这种情况下，显然

$$\langle Q_{\mathcal{S}}|\, O^{+1}_{wu-,wv+}\,|Q_{\mathcal{S}}\rangle = 0 \tag{67}$$

$$\langle Q_{\mathcal{S}}|\, O^{-1}_{wu-,wv+}\,|Q_{\mathcal{S}}\rangle = 0 \tag{68}$$

因此

$$\langle Q_{\mathcal{S}}|\, O^{\perp}_{wu-,wv+}\,|Q_{\mathcal{S}}\rangle = \langle Q_{\mathcal{S}}|\, I - O^{+1}_{wu-,wv+} - O^{-1}_{wu-,wv+}\,|Q_{\mathcal{S}}\rangle \tag{69}$$

$$= 1. \tag{70}$$

由于两者均不在集合中，波函数也保持不变。

D 量子估计器分析

本节的主要目标是证明推论 5。为清晰起见，我们在此重述如下：

推论11。 对于任意 $\varepsilon,\ \delta \in (0,1]$, 存在一种量子流算法，使用

$$O\left(\frac{(km)^2 - (T_1^{<k})^2}{(T_1^{<k})^2}\log n\frac{1}{\epsilon^2}\log\frac{1}{\delta}\right) \tag{71}$$

量子比特和经典比特，以精度 εT_1 估计值 $T_1^{<k}$，其概率为 $1-\delta$。

首先，我们需要证明分母与该算法期望值的（平方根）相匹配：

引理 12。

$$\mathbb{E}\left[R_1^{<k}\right] = T_1^{<k} \tag{72}$$

接下来，我们需要证明分子是方差的上界：

引理 13。

$$\mathrm{Var}\left(R_1^{<k}\right) \le (km)^2 - \left(T_1^{<k}\right)^2 \le (km)^2 \tag{73}$$

利用这两个事实，我们能够证明中位数均值算法可以达到所需的性能，且其空间复杂度与推论 5 相同。这些结论将在后文证明。

D.1 量子草图更新

为了证明上述结论，我们需要了解随着算法的推进，量子草图如何演变。在本节余下部分，令 $g\colon [m] \to \{0, 1\}$ 为一个完全独立哈希函数[1]，满足

$$g(\ell) = \begin{cases} 1 & \text{with probability } 1/k \\ 0 & \text{otherwise.} \end{cases} \tag{74}$$

证明（引理10）。我们对 ℓ 进行归纳。对于 $\ell = 0$,

$$\mathcal{S}_0 = \{(j, \xi) \mid j \in [2m]\} \tag{75}$$

由于我们准备了均匀态

$$\left|Q_{\mathcal{S}_0}\right\rangle = \left(\frac{1}{\sqrt{2m}} \sum_{j=0}^{2m-1} |j\rangle_{2\log(n)}\right) |0\rangle_1 |0\rangle_1 \tag{76}$$

$$= \frac{1}{\sqrt{2m}} \sum_{j=0}^{2m-1} \left|p_j\right\rangle_{\log(n)} \left|r_j\right\rangle_{\log(n)} |0\rangle_1 |0\rangle_1 \tag{77}$$

$$\equiv \frac{1}{\sqrt{2m}} \sum_{j=0}^{2m-1} |j, \xi\rangle \qquad \text{(symbolic form, Eq. (29))} \tag{78}$$

因此结果成立。现在我们用归纳法证明。对于任意 $\ell \in [m-1]$，假设经过 ℓ 条边后结果成立。令 (x, y, σ_{xy})（其中 $x < y$）为第 $(\ell+1)^{\text{th}}$ 条边。

首先考虑查询_边操作的影响。如果 $g(\ell+1) = 0$，这些操作不会发生，因此 S 和 $\mathrm{Q}_{\mathcal{S}}$ 均保持不变。如果 $g(\ell+1) = 1$，我们可以假设它们全部返回 $\perp$，否则算法就会终止。

假设 $\sigma_{xy} = +$，然后在**第 8 步**，我们对与 $(u, x, -)$、$(u, y, -)$ 相关联的查询_边应用于每个 $u \in V$。随后，根据 C.3.3 节，相关状态为：

$$\left|Q_{\mathcal{S}_{\ell+1}}\right\rangle = \frac{O^{\perp}_{ux-,uy-} \left|Q_{\mathcal{S}_\ell}\right\rangle}{\| O^{\perp}_{ux-,uy-} \left|Q_{\mathcal{S}_\ell}\right\rangle \|} \tag{79}$$

$$= \frac{1}{\| O^{\perp}_{ux-,uy-} \left|Q_{\mathcal{S}_\ell}\right\rangle \|} \left(I - O^{+1}_{ux,-,uy,-} - O^{-1}_{ux-,uy-}\right) \left|Q_{\mathcal{S}_\ell}\right\rangle \tag{80}$$

$$= \frac{1}{\| O^{\perp}_{ux,-,uy,-} \left|Q_{\mathcal{S}_\ell}\right\rangle \|} \left(\left|Q_{\mathcal{S}_\ell}\right\rangle - \left(\left\langle u, x, -\middle|Q_{\mathcal{S}_\ell}\right\rangle\right) |u, x, -\rangle - \left(\left\langle u, y, -\middle|Q_{\mathcal{S}_\ell}\right\rangle\right) |u, y, -\rangle\right) \tag{81}$$

特别地，括号内积给出

$$\left\langle u, x, -\middle|Q_{\mathcal{S}_\ell}\right\rangle = \begin{cases} \frac{1}{\sqrt{\mathcal{S}_\ell}} & (u, x, -) \in \mathcal{K}_\ell \\ 0 & \text{otherwise} \end{cases}, \tag{82}$$

$$\left\langle u, y, -\middle|Q_{\mathcal{S}_\ell}\right\rangle = \begin{cases} \frac{1}{\sqrt{\mathcal{S}_\ell}} & (u, y, -) \in \mathcal{K}_\ell \\ 0 & \text{otherwise} \end{cases}. \tag{83}$$

[1]对于每个 $\ell \in [m]$ 在步骤 **5** 上，我们从随机数生成器中获取 $\log(n)$ 个随机比特。

因此，如果 $(u,x,-)$ 或 $(u,y,-)$ 属于 S_ℓ，那么鉴于测量结果为 $\perp$，它们必定已被删除。然后，在**步骤 6** 的循环结束后，我们删除了

$$\mathcal{D}^{+}_{\ell+1} = \{(u,x,-) \in \mathcal{K}_\ell\} \cup \{(u,y,-) \in \mathcal{K}_\ell\} \tag{84}$$

$$= \{(u,z,-) \in \mathcal{K}_\ell : z \in \{x,y\}\} \tag{85}$$

因此

$$\mathcal{K}_{\ell+1} = \mathcal{K}_\ell \setminus \mathcal{D}^{+}_{\ell+1} \cup \{(x,y,+)\} \cup \{(y,x,+)\}. \tag{86}$$

处理完 $\ell+1$ 条边后，底层状态为

$$\left|Q_{S_{\ell+1}}\right\rangle = \frac{1}{\sqrt{2m-2\ell+|\mathcal{K}_\ell|-|\mathcal{D}^{+}_{\ell+1}|}} \left(\sum_{j=2\ell+2}^{2m-1} |j,\xi\rangle + |x,y,-\rangle + |y,x,-\rangle + \sum_{u,v,\sigma_{uv} \in \mathcal{K}_\ell \setminus \mathcal{D}^{+}_{\ell+1}} |u,v,\sigma_{uv}\rangle \right) \tag{87}$$

$$= \frac{1}{\sqrt{2m-2\ell-2+|\mathcal{K}_{\ell+1}|}} \left(\sum_{j=2\ell+2}^{2m-1} |j,\xi\rangle + \sum_{u,v,\sigma_{uv} \in \mathcal{K}_{\ell+1}} |u,v,\sigma_{uv}\rangle \right) \tag{88}$$

假设 $\sigma_{xy} = -$，那么在**第 11 步**，我们对与 $(u,x,+),(u,y,-)$ 相关联的 $\_$ 条边进行查询，对每个 $u \in V$ 都如此。于是相关联的状态为：

$$\left|Q_{S_{\ell+1}}\right\rangle = \frac{O^{\perp}_{ux+,uy-}\left|Q_{S_\ell}\right\rangle}{\|O_{ux+,uy-}\left|Q_{S_\ell}\right\rangle\|} \tag{89}$$

$$= \frac{1}{\|O^{\perp}_{ux+,uy-}\left|Q_{S_\ell}\right\rangle\|} \left(I - O^{+1}_{ux+,uy-} - O^{-1}_{ux+,uy-}\right)\left|Q_{S_\ell}\right\rangle \tag{90}$$

$$= \frac{1}{\|O^{\perp}_{ux-,uy-}\left|Q_{S_\ell}\right\rangle\|} \left(\left|Q_{S_\ell}\right\rangle - \left(\left\langle u,x,+\middle|Q_{S_\ell}\right\rangle \right) |u,y,+\rangle - \left(\left\langle u,y-1\middle|Q_{S_\ell}\right\rangle \right) |u,y,-\rangle \right) \tag{91}$$

特别地，括号内积得出

$$\left\langle u,x,+\middle|Q_{S_\ell}\right\rangle = \begin{cases} \frac{1}{\sqrt{S_\ell}} & (u,x,+) \in \mathcal{K}_\ell \\ 0 & \text{otherwise} \end{cases}, \tag{92}$$

$$\left\langle u,y,-\middle|Q_{S_\ell}\right\rangle = \begin{cases} \frac{1}{\sqrt{S_\ell}} & (u,y,-) \in \mathcal{K}_\ell \\ 0 & \text{otherwise} \end{cases}. \tag{93}$$

因此，如果 $(u,\ x,\ +)$ 或 $(u,\ y,\ -)$ 在 S_ℓ 中，那么鉴于测量结果为 $\perp$，它们必定已被删除。然后，在**第 6 步**的循环结束后，我们已删除

$$\mathcal{D}^{+-}_{\ell+1} = \{(u,x,+) \in \mathcal{K}_\ell\} \cup \{(u,y,-) \in \mathcal{K}_\ell\}, \tag{94}$$

然后在**第 13 步**，我们同样对与 $(u,\ v,\ -)$、$(u,\ y,\ +)$ 相关联的边应用查询 $\_$，针对每个 $u \in V$。然后，通过对称论证，删除集合为

$$\mathcal{D}^{-+}_{\ell+1} = \{(u,x,-) \in \mathcal{K}_\ell\} \cup \{(u,y,+) \in \mathcal{K}_\ell\}, \tag{95}$$

因此，经过**步骤 11–13**后的删除集合为

$$\mathcal{D}^{-}_{\ell+1} = \mathcal{D}^{-+}_{\ell+1} \cup \mathcal{D}^{+-}_{\ell+1} \tag{96}$$

$$= \{(u,x,\sigma_{ux}) \in \mathcal{K}_\ell\} \cup \{(u,y,\sigma_{uy}) \in \mathcal{K}_\ell\} \tag{97}$$

$$= \{(u,z,\sigma_{uz}) \in \mathcal{K}_\ell : z \in \{x,y\}\} \tag{98}$$

因此，加入 $(x,\ y,\ -)$ 和 $(y,\ x,\ -)$ 后的集合为

$$\mathcal{K}_{\ell+1} = \mathcal{K}_\ell \setminus \mathcal{D}^-_{\ell+1} \cup \{(x,y,-)\} \cup \{(y,x,-)\} \tag{99}$$

因此，对于两种情形，如果 $g(\ell+1) = 1$要删除，则其对 Q 底层集合的影响可概括为

$$\mathcal{D}_{\ell+1} = \begin{cases} \{(u,z,-) \in \mathcal{K}_\ell : z \in \{x,y\}\} & \text{if } \sigma_{xy} = + \wedge g(\ell+1) = 1 \\ \{(u,z,\sigma_{uz}) \in \mathcal{K}_\ell : z \in \{x,y\}\} & \text{if } \sigma_{xy} = - \wedge g(\ell+1) = 1 \\ \emptyset & \text{otherwise.} \end{cases} \tag{100}$$

$$= \{(u,z,\sigma_{uz}) \in \mathcal{K}_\ell : z \in \{x,y\} \wedge (\sigma_{xy} = -1 \vee \sigma_{uz} = -1) \wedge (g(\ell+1) = 1)\} \tag{101}$$

从中删除。换言之，如果 $\sigma_{xy} = -$，则与 xy 相连的负边或正边会被删除；而如果 $\sigma_{xy} = +$，则只会删除与 xy 相连的负边。

调用 insert 前后的更新集合为

$$\mathcal{K}_\ell \setminus \mathcal{D}_{\ell+1} = \left\{(u,v,\sigma_{uv}) \in \mathcal{K}_\ell : (g(\ell+1) = 0) \vee (v \notin xy) \vee (\sigma_{xy} = + \vee \sigma_{uv} = +)\right\}, \tag{102}$$

$$\mathcal{K}_{\ell+1} = \mathcal{K}_\ell \setminus \mathcal{D}_{\ell+1} \cup \{x,y,\sigma_{xy}\} \cup \{y,x,\sigma_{xy}\}, \tag{103}$$

这与我们预期的引理相符。

具体而言，$\exists i \in [\ell+1]$ s.t. $(u,\ v,\ \sigma_{uv}) = (e_i,\ \sigma_i)$ 现在包含来自 $i = \ell+1$的 $(x,\ y,\ \sigma_{xy})$，$(y,\ x,\ \sigma_{xy})$，这是由于 insert$((xy,\ \sigma_{xy}),\ \mathrm{Q})$ 所致。条件：

$$\bigvee_{j=i+1,\ldots,\ell+1} \Big(g(j) = 0\Big) \vee \Big(v \notin e_j\Big) \vee \Big(\ (\sigma_j = +) \wedge (\sigma_{uv} = +)\ \Big) \tag{104}$$

这表明，如果 $g(\ell+1) =$，则 $\mathrm{D}_{\ell+1}$ 是空集合；否则，$\mathrm{D}_{\ell+1}$ 将包含与$(e_{\ell+1}, \sigma_{\ell+1})$关联的所有负边或正边（如果 $\sigma_{\ell+1} = -$），或者包含所有关联的负边（如果 $\sigma_{\ell+1} = +$）。

□

为了分析该算法，我们需要讨论草图在不同时间点的状态。令 E^ℓ_T 表示草图在第 ℓ步时被销毁的概率。令 E^ℓ_{NT} 表示在处理完第 ℓ^{th} 条边时算法尚未终止这一事件。具体而言，量子草图仍然存在且尚未坍缩，并且我们已将第 ℓ^{th} 条边添加到草图中。令 $\mathrm{E}^\ell_{\mathtt{exit}}$ 表示我们在处理第 ℓ^{th} 条边时退出算法这一事件。

引理 14。对于任意 $\ell \in [m]$,

$$\mathbb{P}\big[\mathcal{E}^\ell_{NT}\big] = \frac{|\mathcal{S}_\ell|}{2m}, \tag{105}$$

这可以看作参考文献 [1]中引理 1的一个应用。

引理 15。 设(vw, σ_{vw}) 为数据流中的第 ℓ^{th} 条边。在以 $\mathcal{E}^{\ell-1}_{NT}$为条件的情况下，

(a) 若$(uv, -) \prec (vw, \sigma_{vw})$，则$(uv, -)$仍在草图中的概率为$(1-1/k)^{d_{uvw}}$。*(b)* 若$(uv, +) \prec (vw, \sigma_{vw})$，则$(uv, +)$仍在草图中的概率为$(1-1/k)^{d^-_{uvw}}$。

*I*特别地，正号边只能被负号邻边删除。

证明。（a）根据 $\mathrm{K}_{\ell-1}$ 的定义，只有当 $(uv, -)$ 在某个时刻 $j < \ell$ 进入草图，并且对于某条边 $j < k < \ell$，不存在任何在 $(uv, -)$ 之后到达、与 v 关联且满足 $g(k) = 1$的带符号边时，$(uv, -)$ 才在草图中。假设 vq 是满足 $j < k < \ell$ 且满足 $g(k) = 1$的 k^{th} 边。如果 $\sigma_{vq} = -1$，则第11行中的查询_边$((uq, +), (uv, -), \mathrm{Q})$ 调用（以其返回 $\perp$ 为条件）将从草图中删除 $(uv, -)$。同样地，如果 $\sigma_{vq} = +1$，则第8行中的查询_边$((uq, -), (uv, -), \mathrm{Q})$ 调用会删除 $(uv, -)$（以其返回 $\perp$ 为条件）。无论哪种情况，在到达 ℓ 的条件下，$(uv, -)$ 仍留在草图中的唯一可能是，所有在 $(uv, -)$ 之后到达、与 v 关联的边均未

触发查询操作。这 j^{th} 集查询以 $1/k$ 的概率发生，因此，对于每个这样的 j，所有查询均不发生的概率为 $(1-1/k)^{d^+_{uvw}+d^-_{uvw}} = (1-1/k)^{d_{uvw}}$，因为 g 是完全独立的。

(b) 根据 $\mathrm{K}_{\ell-1}$ 的定义，$(uv,+)$ 位于草图中，当且仅当它在某个时刻 $j<\ell$ 进入草图，并且对于某条边 $j<k<\ell$，不存在任何在 $(uv,+)$ 之后到达且与 v 相连的 -1 带符号边，使得 $g(k)=1$。假设 vq 是满足 $j<k<\ell$ 且 $g(k)=1$ 的 k^{th} 边。如果 $\sigma_{vq}=-1$，那么第 13 行中的查询边 $((uq,-),(uv,+),\mathrm{Q})$ 调用（以其返回 $\perp$ 为条件）将从草图中删除 $(uv,-)$。另一方面，如果 $\sigma_{vq}=+1$，那么 $(uv,+)$ 不可能从草图中被删除。即使 $g(k)=1$，第8行中的查询边$((uq,-),(uv,-),\mathrm{Q})$ 调用也与 $(uv,+)$ 无关。因此，我们只需追踪与 v 相连的 -1 边，由此可得 $(uv,+)$ 仍在草图中的概率为 $(1-1/k)^{d^-_{uvw}}$。□

D.2 数据流中的有符号三角形

在数据流上固定一种排序。对于边 e，f，我们用 $e<f$ 表示 e 在数据流中先于 f 到达。固定顶点 u，v，$w\in V$，使得 $(uv,\ \sigma_{uv})<(vw,\ \sigma_{vw})$。将正度 d^+_{uvw} 定义为：与 v 相连、且在 $(uv,\ \sigma_{uv})$ 和 $(vw,\ \sigma_{vw})$ 之间到达的 $+$-边的数量（不包括这两条边本身）。类似地定义 d^-_{uvw}。那么总度为 $d_{uvw}:=d^+_{uvw}+d^-_{uvw}$。

$$t^{<k}_{uvw,+--}=\begin{cases}(1-1/k)^{d^-_{uvw}+d_{uwv}} & \text{if } (uv,+)<(uw,-)<(vw,-) \text{ is a } T_1 \text{ in } G\\ 0 & \text{otherwise.}\end{cases} \tag{106}$$

$$t^{<k}_{uvw,-+-}=\begin{cases}(1-1/k)^{d_{uvw}+d^-_{uwv}} & \text{if } (uv,-)<(uw,+)<(vw,-) \text{ is a } T_1 \text{ in } G\\ 0 & \text{otherwise.}\end{cases} \tag{107}$$

$$t^{<k}_{uvw,--+}=\begin{cases}(1-1/k)^{d_{uvw}+d_{uwv}} & \text{if } (uv,-)<(uw,-)<(vw,+) \text{ is a } T_1 \text{ in } G\\ 0 & \text{otherwise.}\end{cases} \tag{108}$$

我们记作 $T^{<k}_{+--}:=\sum_{(u,\ v,\ w)\in V^3} t^{<k}_{uvw,\ +--}$。类似地定义 $T^{<k}_{-+-}$，$T^{<k}_{--+}$ 并令 $T^{<k}_1=T^{<k}_{+--}+T^{<k}_{-+-}+T^{<k}_{--+}$。同样，定义

$$t^{>k}_{uvw,+--}=\begin{cases}1-(1-1/k)^{d^-_{uvw}+d_{uwv}} & \text{if } (uv,+)<(uw,-)<(vw,-) \text{ is a } T_1 \text{ in } G\\ 0 & \text{otherwise.}\end{cases} \tag{109}$$

$$t^{>k}_{uvw,-+-}=\begin{cases}1-(1-1/k)^{d_{uvw}+d^-_{uwv}} & \text{if } (uv,-)<(uw,+)<(vw,-) \text{ is a } T_1 \text{ in } G\\ 0 & \text{otherwise.}\end{cases} \tag{110}$$

$$t^{>k}_{uvw,--+}=\begin{cases}1-(1-1/k)^{d_{uvw}+d_{uwv}} & \text{if } (uv,-)<(uw,-)<(vw,+) \text{ is a } T_1 \text{ in } G\\ 0 & \text{otherwise.}\end{cases} \tag{111}$$

以及相应的 $T^{>k}_{+--}$，$T^{>k}_{-+-}$，$T^{>k}_{--+}$，$T^{>k}_1$ 值。于是 $T_1=T^{<k}_1+T^{>k}_1$。

回想一下，$R^{<k}_1$是算法 3 的输出。我们根据算法返回的不同方式，进一步分解这个随机变量。考虑 ℓ^{th} 边（$vw,\ \sigma_{vw}$）。对于任意 $u\in V$和 $\sigma_{uv},\ \sigma_{uw}\in\{\pm1\}$，我们将随机变量$R_{u,\ vw;\sigma_{uv}\sigma_{uw}\sigma_{vw}}$定义如下。在以下任一情况下，$R_{u,\ vw;\sigma_{uv}\sigma_{uw}\sigma_{vw}}=0$:

（1）如果 $g(\ell)=0$，这意味着我们不进行任何测量（因此无法在该边上返回）；或者（2）如果任何先前的查询_边操作破坏了草图（因此我们甚至无法到达边 ℓ）；或者

（3）测量查询_边（u，v，σ_{uv}，（u，w，σ_{uw}））返回的值 $r_{\sigma_{uv}\sigma_{uw}}$ 等于 $\perp$。

否则，$R_{u,\ vw;\sigma_{uv}\sigma_{uw}\sigma_{vw}} = r_{\sigma_{uv}\sigma_{uw}} km$。定义

$$R_{--+} = \sum_{(v<w)\in E}\sum_{u\in V} R_{u,vw;--+}, \quad \text{(from step 7)} \tag{112}$$

$$R_{+--} = \sum_{(v<w)\in E}\sum_{u\in V} R_{u,vw;+--}, \quad \text{(from step 10)} \tag{113}$$

$$R_{-+-} = \sum_{(v<w)\in E}\sum_{u\in V} R_{u,vw;-+-}. \quad \text{(from step 12)} \tag{114}$$

于是 $R_1^{<k} = R_{--+} + R_{-+-} + R_{+--}$。求和中对一条边上的节点显式排序，是为了避免重复计数问题。例如，考虑三角形 $(12,\ +) < (13,\ -) < (23,\ -)$。该算法还确保，即使边被读作 $(32,\ -)$，由于 $2 < 3$，第 10 行中的查询_edges((1，2，+)，(1，3，–)) 调用也会被触发。因此，接下来我们假设 $v < w$。

在证明引理 12 之前，我们需要了解各个 $\mathbb{E}\left[R_{u,vw;\sigma_{uv}\sigma_{uw}\sigma_{vw}}\right]$ 值。

引理16。

$$\mathbb{E}\left[R_{u,vw;--+}\right] = t^{<k}_{uvw,--+} + t^{<k}_{uwv,--+} \tag{115}$$

这两个值都对应于三角形 $\{(uv,\ -)、(uw,\ -)、(vw,\ +)\}$，其中 $(vw,\ +)$ 完成该三角形。此外，由于这些值的定义方式，二者中恰有一个可以非零；而一旦非零，其值便等于 $(1-1/k)^{d_{uvw}+d_{uwv}}$。这将对下面的证明有所帮助。

证明。首先考虑 $t^{<k}_{uvw,--+} + t^{<k}_{uwv--+} = 0$ 的情况。根据 $t^{<k}_{uvw,--+}$ 的定义，这意味着 $\{(uv,-)、(uw,-)、(vw,+)\}$ 中要么有一个不是 G 中的三角形，要么虽然是三角形，但 $(uv,-)$ 或 $(uw,-)$ 中至少有一个在数据流中晚于 $(vw,+)$ 到达。在这两种情况下，根据引理10，在 $(vw,+)$ 查询发生时（如果确实发生），$(uv,-)$ 和 $(uw,-)$ 中至少有一个不会位于草图中。如果两条边都不在草图中，则该查询必然返回 $\perp$，因此 $\mathbb{E}\left[R_{u,vw;--+}\right] = 0$。如果其中恰有一条边在草图中，则查询要么返回 $\perp$，要么以相同概率返回 1 或 -1，因此仍然有 $\mathbb{E}\left[R_{u,vw;--+}\right] = 0$。无论哪种情况，$\mathbb{E}\left[R_{u,vw;--+}\right] = 0 = t^{<k}_{uvw,--+} + t^{<k}_{uwv--+}$。

现在假设 $t^{<k}_{u,\ vw--+} + t^{<k}_{u,\ wv--+} > 0$。注意，这些项中至多只有一项非零。根据定义，无论哪一项非零，$t^{<k}_{u,\ vw--+} + t^{<k}_{u,\ wv--+} = (1-1/k)^{d_{uvw}+d_{uwv}}$。不失一般性，设 $t^{<k}_{u,\ vw--+} > 0$，则 $(uv,\ -) < uw- < vw+$。我们证明 $\mathbb{E}\left[R_{u,vw;--+}\right] = (1-1/k)^{d_{uvw}+d_{uwv}}$，从而证明 $\mathbb{E}\left[R_{u,vw;--+}\right] = t^{<k}_{u,\ vw--+} + t^{<k}_{u,\ wv--+}$。

设 $(vw,\ +)$ 是第 ℓ^{th} 条到达的边。令 G_ℓ 表示事件：$g(\ell) = 1$，并回忆 $\mathrm{E}_{NT}^{\ell-1}$表示在开始处理 $(vw,\ +)$ 时我们尚未终止这一事件。此外，根据引理 10，此时底层草图为 $\mathrm{S}_{\ell-1}$。令 F_{--} 表示在给定 $\mathrm{E}_{NT}^{\ell-1}$ 的条件下，$(uv,\ -)$ 和 $(uw,\ -)$ 仍在 $\mathrm{S}_{\ell-1}$ 中这一事件。由于 g 完全独立，我们有

$$\mathbb{P}\left[\mathcal{F}_{--}|\mathcal{E}_{NT}^{\ell-1}\right] = \mathbb{P}[(u,v,-)\in\mathcal{S}_{\ell-1}]\cdot\mathbb{P}[(u,w,-)\in\mathcal{S}_{\ell-1}] \tag{116}$$

$$= (1-1/k)^{d_{uvw}}\cdot(1-1/k)^{d_{uwv}} \tag{117}$$

$$= (1-1/k)^{d_{uvw}+d_{uwv}} \tag{118}$$

根据引理 15 得出这些概率。

在给定 F_{--} 和 $\mathrm{E}_{NT}^{\ell-1}$的条件下，查询_边时，以概率 $2/|T_{l-1}|$返回 $+1$，否则返回 $\perp$。当返回 $\perp$ 时，$R_{u,\ vw;--+}=0$ 根据定义成立。然后

$$\mathbb{E}\left[R_{u,vw;--+}|\mathcal{F}_{--},\mathcal{E}_{NT}^{\ell-1},\mathcal{G}^{\ell}\right]=km\cdot\frac{2}{|\mathcal{S}_{\ell-1}|}+0\cdot\left(1-\frac{2}{|\mathcal{S}_{\ell-1}|}\right)=\frac{2km}{|\mathcal{S}_{\ell-1}|} \tag{119}$$

根据引理 14 以及 g的随机选择，

$$\mathbb{P}\left[\mathcal{G}^{\ell},\mathcal{E}_{NT}^{\ell-1}\right]=\mathbb{P}\left[\mathcal{G}^{\ell}\right]\cdot\mathbb{P}\left[\mathcal{E}_{NT}^{\ell-1}\right]=\frac{1}{k}\cdot\frac{|\mathcal{S}_{\ell-1}|}{|T_0|}=\frac{|\mathcal{S}_{\ell-1}|}{2km}. \tag{120}$$

另请注意，如果我们不进行任何查询，那么 $R_{u,\ vw;--+}=0$，因此 $\mathbb{E}\left[R_{u,vw;--+}|\mathrm{G}^{\ell}\right]=0$。利用这一事实以及全期望定律：

$$\mathbb{E}\left[R_{u,vw;--+}|\mathcal{F}_{--}\right]=\mathbb{E}\left[R_{u,vw;--+}|\mathcal{F}_{--},\mathcal{E}_{NT}^{\ell-1},\mathcal{G}^{\ell}\right]\mathbb{P}\left[\mathcal{E}_{NT}^{\ell-1},\mathcal{G}^{\ell}|\mathcal{F}_{--}\right] \tag{121}$$

$$=\mathbb{E}\left[R_{u,vw;--+}|\mathcal{F}_{--},\mathcal{E}_{NT}^{\ell-1},\mathcal{G}^{\ell}\right]\mathbb{P}\left[\mathcal{E}_{NT}^{\ell-1},\mathcal{G}^{\ell}\right]\qquad\text{(Independent from }\mathcal{F}_{--}\text{)} \tag{122}$$

$$=\frac{2km}{|\mathcal{S}_{\ell-1}|}\cdot\frac{|\mathcal{S}_{\ell-1}|}{2km} \tag{123}$$

$$=1. \tag{124}$$

另一方面，以 F_{--} 为条件，有两种可能：草图中没有边，或者草图中有一条边。情况 1 中，查询以概率 1 返回 $\perp$。情况 2 中，查询以相同概率返回 $+1$ 和 -1，二者在期望中相互抵消。无论哪种情况，$\mathbb{E}\left[R_{u,vw;--+}|\mathrm{F}_{--}\right]=0$。

因此我们得出结论：

$$\mathbb{E}\left[R_{u,vw;--+}\right]=\mathbb{E}\left[R_{u,vw;--+}|\mathcal{F}_{--}\right]\mathbb{P}[\mathcal{F}_{--}] \tag{125}$$

$$=1\cdot(1-1/k)^{d^{\rightarrow}_{uvw}+d^{\rightarrow}_{uwv}} \tag{126}$$

$$=t^{<k}_{uvw,\ --+}+t^{<k}_{uwv,\ --+} \tag{127}$$

因此，无论 $t^{<k}_{uvw,\ --+}+t^{<k}_{uwv,\ --+}$ 是否为 0，我们都有 $\mathbb{E}\left[R_{u,vw;--+}\right]=t^{<k}_{uvw,\ --+}+t^{<k}_{uwv,\ --+}$。□

引理 17。

$$\mathbb{E}\left[R_{u,vw;+--}\right]=t^{<k}_{uvw,+--}+t^{<k}_{uwv,-+-} \tag{128}$$

该陈述中的下标有些微妙，但这两个值都对应于三角形 $\{(uv,\ +),\ (uw,\ -),\ (vw,\ -)\}$，其中 $(vw,\ -)$ 补全了这个三角形。此外，由于这些值的定义方式，它们中恰有一个可以非零；而当其非零时，它等于 $\left(1-\frac{1}{k}\right)^{d^{\rightarrow}_{uvw}+d^{\rightarrow}_{uwv}}$。这将有助于下面的证明。

证明。这个证明与引理17的证明几乎完全相同。主要区别在于，现在我们希望事件 F_{+-} 表示在给定 $\mathrm{E}_{NT}^{\ell-1}$的条件下，$(uv,\ +)$ 和 $(uw,\ -)$ 仍在 $\mathrm{S}_{\ell-1}$ 中。利用

引理 15 中的概率，我们得到

$$\mathbb{P}\left[\mathcal{F}_{+-}|\mathcal{E}_{NT}^{\ell-1}\right] = \mathbb{P}[(u,v,+) \in \mathcal{S}_{\ell-1}] \cdot \mathbb{P}[(u,w,-) \in \mathcal{S}_{\ell-1}] \tag{129}$$

$$= (1-1/k)^{d^{-}_{uvw}} \cdot (1-1/k)^{d_{uwv}} \tag{130}$$

$$= (1-1/k)^{d^{-}_{uvw}+d_{uwv}} \tag{131}$$

由此可得 $\mathbb{E}\left[R_{u,vw;+--}\right] = (1-1/k)^{d^{-}_{uvw}+d_{uwv}}$（当其非零时），因此我们有 $\mathbb{E}\left[R_{u,vw;+--}\right] = t^{<k}_{uvw,\ +--} + t^{<k}_{uwv,\ -+-}$。

□

引理 18。

$$\mathbb{E}\left[R_{u,vw;-+-}\right] = t^{<k}_{uvw,-+-} + t^{<k}_{uwv,+--}. \tag{132}$$

由对称性可知，该证明与引理 17 的证明完全相同。

现在我们可以证明 $\mathbb{E}\left[R_1^{<k}\right] = T_1^{<k}$。

（引理 12 的证明）。回想一下，$T_1^{<k} = T^{<k}_{+--} + T^{<k}_{-+-} + T^{<k}_{--+}$。目标是证明，算法会对 $t^{<k}$ 值恰好计数一次。首先注意到

$$\mathbb{E}[R_{--+}] = \sum_{(v<w)\in E}\sum_{u\in V}\mathbb{E}\left[R_{u,vw;--+}\right] \tag{133}$$

$$= \sum_{(v<w)\in E}\sum_{u\in V} t^{<k}_{uvw,--+} + t^{<k}_{uwv,--+} \qquad \text{(Lemma 16)} \tag{134}$$

$$= \sum_{(u,v,w)\in V^3} t^{<k}_{uvw,--+} \tag{135}$$

$$= T^{<k}_{--+} \tag{136}$$

其中，倒数第二行由以下事实得出：对于一个三元组 $(u,\ v,\ w) \in V^3$，只有一种排序可能导致非零的 $t^{<k}_{uv,\ w;--+}$ 值。

对于剩余的情况，我们必须更加谨慎。与上述类似，我们可以利用期望的线性性，观察到以下结果：

$$\mathbb{E}[R_{+--}] = \sum_{(v<w)\in E}\sum_{u\in V}\mathbb{E}\left[R_{u,vw;+--}\right] = \sum_{(v<w)\in E}\sum_{u\in V} t^{<k}_{uvw,+--} + t^{<k}_{uwv,-+-} \qquad \text{(Lemma 17)} \tag{137}$$

$$\mathbb{E}[R_{-+-}] = \sum_{(v<w)\in E}\sum_{u\in V}\mathbb{E}\left[R_{u,vw;-+-}\right] = \sum_{(v<w)\in E}\sum_{u\in V} t^{<k}_{uvw,-+-} + t^{<k}_{uwv,+--} \qquad \text{(Lemma 18)} \tag{138}$$

这里我们不能忽略每一行中的排序，因为我们统计的是略有不同的三角形 $-+-$和 $+--$，这取决于数据流中的边排序。不过，将这些值相加并重新排列后，我们得到

$$\mathbb{E}[R_{+--}] + \mathbb{E}[R_{-+-}] = \sum_{(v<w)\in E}\sum_{u\in V} t^{<k}_{uvw,+--} + t^{<k}_{uwv,+--} + \sum_{(v<w)\in E}\sum_{u\in V} t^{<k}_{uvw,-+-} + t^{<k}_{uwv,-+-} \tag{139}$$

$$= \sum_{(u,v,w)\in V^3} t^{<k}_{uv,w;+--} + \sum_{(u,v,w)\in V^3} t^{<k}_{uv,w;-+-} \tag{140}$$

$$= T^{<k}_{+--} + T^{<k}_{-+-}. \tag{141}$$

In conclusion, $\mathbb{E}\left[R_1^{<k}\right] = T_{+--}^{<k} + T_{-+-}^{<k} + T_{--+}^{<k} = T_1^{<k}$. □

接下来，$R_1^{<k}$的方差上界显而易见。

（引理 13 的证明）。这可直接由 $\left|R_1^{<k}\right| \le km$和 $\mathbb{E}\left[R_1^{<k}\right] = T_1^{<k}$ 得出。 □

最后，我们证明中位数均值算法所需的性能和空间界。

（推论11的证明）。回顾一下，$\mathbb{E}\left[R_1^{<k}\right] = T_1^{<k}$和 $\mathrm{Var}\left(R_1^{<k}\right) \le (km)^2 - \left(T_1^{<k}\right)^2$。

首先，对量子估计器的 N_ϵ 个独立副本取平均，得到一个值 X_1，其方差为 $Var(X_1) = \frac{Var\left(R_1^{<k}\right)}{N_\epsilon} \le \frac{(km)^2 - \left(T_1^{<k}\right)^2}{N_\epsilon}$。利用切比雪夫不等式，我们得到

$$\mathbb{P}\left[\left|\overline{X}_1 - T_1^{<k}\right| \ge \epsilon T_1^{<k}\right] \le \frac{(km)^2 - (T_1^{<k})^2}{N_\epsilon (T_1^{<k})^2} \frac{1}{\epsilon^2} \tag{142}$$

选择 $N_\epsilon \ge 4\frac{(km)^2 - (T_1^{<k})^2}{(T_1^{<k})^2} \frac{1}{\epsilon^2}$得到 $\mathbb{P}\left[\left|\overline{X}_1 - T_1^{<k}\right| \le \epsilon T_1^{<k}\right] \ge 3/4$。

重复此过程 N_δ 次，得到估计值 $\overline{X}_{1,1}, \ldots, \overline{X}_{1,N_\delta}$，并定义变量

$$\hat{X}_1 := \mathrm{median}\left(\overline{X}_{1,1}, \ldots, \overline{X}_{1,N_\delta}\right) \tag{143}$$

我们关注的是，何时 $\hat{X}_1$ 是 $R_1^{<k}$的良好估计量。定义伯努利指示变量

$$Z_j := 1\left\{\left|\overline{X}_{1,j} - T_1^{<k}\right| \ge \epsilon T_1^{<k}\right\} \tag{144}$$

用于判断估计值 $\overline{X}_{1,j}$是否不准确。超过半数的估计量不准确时，中位数就会失效。也就是说，当 $Z \ge N_\delta/2$ 对于 $Z := \sum_j Z_j$ 时。根据上述内容，我们知道 $\mathbb{E}[Z] \le N_\delta/4$。应用霍夫丁不等式可得

$$\mathbb{P}[Z \ge N_\delta/2] = \mathbb{P}\left[Z - \mathbb{E}[Z] \ge N_\delta/4\right] \le \exp\left(-2N_\delta\left(\frac{1}{4}\right)^2\right) = \exp(-N_\delta/8) \tag{145}$$

使用 $N_\delta \ge 8\ln\frac{1}{\delta}$可确保$\hat{X}_1$ 以概率 $1-\delta$表现良好。

总共，我们需要运行 $N_\epsilon \cdot N_\delta = \frac{4(km)^2 - (T_1^{<k})^2}{(T_1^{<k})^2} \frac{1}{\epsilon^2} \cdot 8\ln\frac{1}{\delta} = 32\frac{(km)^2 - (T_1^{<k})^2}{(T_1^{<k})^2} \frac{1}{\epsilon^2}\ln\frac{1}{\delta}$次算法迭代。每次迭代需要 $2\log(n)$ 个量子比特和 $m = O(\log n)$ 个经典比特来运行。因此，这个整体并行化中位数均值算法所需的总空间为

$$O\left(\frac{(km)^2 - (T_1^{<k})^2}{(T_1^{<k})^2} \log n \frac{1}{\epsilon^2} \log\frac{1}{\delta}\right). \tag{146}$$

□