arXiv:2611.32528v1 [cs.DS] 24 Nov 2026

Ollivier–Ricci 曲率的最小割表述

Giorgio Micaletto芝加哥大学统计学系
gmicaletto@uchicago.edu

摘要

通常，计算 Ollivier–Ricci 曲率需要局部距离矩阵，并且每条边都要解一个线性规划，因此人们转而采用基于熵正则化或组合代理的方法。我们证明，精确计算曲率两者皆不需要。对于任意单步测度，Kantorovich对偶都有一个整数三层最优解，从而将每条边的计算化为在一个无距离网络上求解一次最小 s–t 割；该网络的规模与读取邻接表所需时间成线性关系。这使得即便是需要查询二次数量级距离的实例，也能在线性时间内求解；而一个反向约化则表明，任何精确算法都无法超越二分图匹配。我们的求解器的中位运行速度比距离构建加上 `POT`的 C++simplex 快 $6\times$（最高可达 $506\times$）；比 `GraphRicciCurvature`快 $441\times$；比图最小费用流快 $23\times$。即便已知距离，在 `GraphRicciCurvature`的正则化下，Sinkhorn迭代的中位速度也慢 $12\times$；此外，组合代理会改变边排序，遗漏大部分曲率最负的边。

1 引言

Ollivier–Ricci（OR）曲率 [Ollivier，2009；Lin 等人， 2011] 将一条边两个端点处的随机游走测度之间的 Wasserstein-1 距离与该边的长度进行比较，并用于研究消息传递网络中的过压缩 [Topping 等人，2022；Nguyen 等，2023；Fesser 和 Weber，2024b]、结构编码 [Fesser 和 Weber，2024a]、池化 [Sanders 等人， 2025]，以及评估生成模型 [Southern 等人， 2023]，和检测社区 [Ni等人，2019；Sia 等人，2019；Tian 等人， 2025]。

障碍在于成本。对于 $(i,j) \in E$ 且度数为 d_i, d_j 的情况，OR 曲率需要求解一个运输问题，其中最多包含 $(d_i+1)(d_j+1)$ 个变量，其成本为图距离。`GraphRicciCurvature` [Ni 等人， 2019] 通过最短路径计算距离，并使用`POT` [Flamary 等人， 2021]的方法为每条边求解一个线性规划；对于大邻域，则切换到 Sinkhorn迭代 [Cuturi， 2013]，并将每个顶点的邻居数限制在最多 3,000 个。因此，对于大型图或需要重新计算 OR 曲率的算法而言，OR 曲率被认为成本过高；越来越多的研究使用代理方法 [Forman，2003；Sreejith 等人，2016；Samal 等人，2018；Topping 等人，2022；Fesser 和 Weber，2024b；Fesser 等人。

， 2024] 以及近似方法 [Park 和 Li，2025年，Tian 等人，2025年，Zhou 和 Jiao， 2026]，这些方法以速度换取精确性，或给出界 [Jost 和 Liu，2014，Kang 和 Park，2026，Gu 等人，2026，Micaletto 和 Nigrelli， 2026]，这些方法不会产生点估计。此外，DasGupta 等人 [2023] 证明了对于精确局部算法，在大小为 $n^2/6$ 的邻域上，即使是在等度邻域上，也需要 n次距离矩阵查询的最坏情况下界。

我们表明，精确的 OR 曲率既不需要距离矩阵，也不需要线性规划。对于支撑在闭邻域上的测度，每个相关距离都由邻接关系和公共邻居决定；Kantorovich对偶随后具有一个整数三层最优解，该最优解以一元编码表示后，可化为一个最小 s–t 割 [Picard，1976，Hochbaum， 2001]，而在图邻域上，该网络的规模与 i的邻接表、 j以及具有净质量的顶点的数量呈线性关系。

整数势函数 [Bhattacharya 和 Mukherjee，2020年，引理 2。2]，有界差分约束的闭包归约 [Hochbaum，2001，Ishikawa， 2003]，以及 $\{0,1,2,3\}$ 局部距离 [DasGupta 等人， 2023]均已为人所知。我们的贡献是一个无距离网络（其运行时间与读取邻接表的规模呈线性关系，这得益于图对距离-2 节点对的隐式双团覆盖 [Feder 和 Motwani， 1991]）以及一个二分图匹配下界（定理 9）：剩下的就是最大流的代价。

预印本。初步工作。

贡献。

（C1）**约化**（第3节）。对于 $\{1,\dots,D\}$， $\mathcal{W}_1 = DA - F$通过在 F上（$D-1$）（$|P|+|N|$）$+2$个节点的最大流（定理1）可得到精确证书（定理5）和极小极大公式（定理4）。一步测度（惰性、非惰性、任意闲置率、LLY [Lin 等人， 2011],加权游走）具有 $D=3$； r-步游走和 r-球具有 $D=2r+1$。

(C2) **线性规模网络** （第 4 节）。每个公共邻居对应一个中继节点，使网络规模界定为$3\ell_{ij}+2$ 个节点和 $5\ell_{ij}$ 条弧，并在 $O(\ell_{ij})$ 时间内构建完成（定理 7， $\ell_{ij} = d_i + d_j + \sum_{x\in P\cup N} d_x$）。对于惰性、非惰性、LLY 和固定惰性曲率，在 $\ell_{ij}^{1+o(1)}$ 时间（$O(n)$ 内即可完成 DasGupta 等人的 下界实例 [2023]计算；定理 8）。反之，一条边可以编码任意二分图最大匹配（定理9），因此除非能为其设计更快的算法，否则该界无法改进。

(C3) **求解器**（第5 节）。该求解器结合了位集、块/见证中继节点、孪生合并、带有 $O(\sqrt{A})$ 个阶段的热启动 Dinic 算法（定理 10），以及局部增量更新（定理11）。

(C4) **评估** （第 6节）。针对（C1）的 5 种测度，在 19 个图上评估精确性，并在最多包含 1.17×10^8 条边的 20 个真实网络上评估速度，与 LP、网络单纯形法、最小费用流和近似基线进行比较。

代码： `https://github.com/GiorgioMB/orcut` （标签 v1.4.4 会生成此处的所有结果）。单独的求解器是 PyPI 软件包 `faster-or-curvature`，其拥有独立的版本号：v1.2.2 包含 `orcut` 1.4.4 中的求解器，1.3.0 添加了第 7 节中的真实长度和权重，而 1.3.1 是最终发布版本。

2 背景与相关工作

Ollivier–Ricci曲率。令 $G=(V,E)$ 为一个有限简单连通无向图，其跳跃距离为 dist， $\mathcal{N}(u)$ 表示 u 的邻居， $\mathcal{N}[u]=\mathcal{N}(u)\cup\{u\}$，且 $d_u=|\mathcal{N}(u)|$。每个 $u\in V$ 都具有一个概率测度 m_u，而 (i,j) 的曲率 $\in E$ 为 $\kappa(i,j) = 1-\mathcal{W}_1(m_i,m_j)$。我们称$m_u$为一步的，当 supp $m_u\subseteq\mathcal{N}[u]$时；这涵盖了 p-惰性游走[1] 以及加权游走。LLY 曲率 $\kappa_{\mathrm{LLY}}=\lim_{p\to1}\kappa_p/(1-p)$ 具有 $\kappa_p=(1-p)\,\kappa_{\mathrm{LLY}}$，对于 $p\ge1/(\max\{d_i,d_j\}+1)$ 而言如此；Bourne 等人， 2018], 的默认值是 LLY 曲率的一半。

对于 $(i,j)\in E$，令 $S=\mathcal{N}[i]\cup\mathcal{N}[j]$, $\mathcal{C}=\mathcal{N}(i)\cap\mathcal{N}(j)$, $\mathcal{U}_i=\mathcal{N}(i)\setminus\mathcal{N}[j]$, $\mathcal{U}_j=\mathcal{N}(j)\setminus\mathcal{N}[i]$，使得 $S=\{i,j\}\sqcup\mathcal{C}\sqcup\mathcal{U}_i\sqcup\mathcal{U}_j$。对于 $\mu=m_i-m_j$，令 $P=\{\mu>0\}$, $N=\{\mu<0\}$, $\nu=-\mu$ 在 N 上，并且 $A=\mu(P)=\nu(N)$。两个测度共有的质量可以保持不动，因此 $\mathcal{W}_1(m_i,m_j)$ 是将 μ^+ 移动到 μ^- [Villani, 2003], 的最优成本，并由对偶性

$$\mathcal{W}_1(m_i,m_j)=\max\left\{\sum_{x\in P}\mu(x)u_x-\sum_{y\in N}\nu(y)v_y : u_x-v_y\le\mathrm{dist}(x,y)\ \ \forall(x,y)\in P\times N\right\}. \tag{1}$$

曲率计算。传输求解器需要 $|P|\times|N|$ 代价矩阵：网络单纯形法 [Bonneel 等人，2011，Flamary 等人， 2021], 最小费用流和指派算法 [Kuhn，1955，Ahuja 等，1993，Perron 和 Furnon， 2024], 或熵正则化 [Cuturi， 2013] （这是近似方法）。我们所知的唯一一种无需距离的表述是 Beckmann 的表述：在图上， $\mathcal{W}_1$ 对于跳数度量，它是边上单位代价且散度为 μ [的流的最小费用 1952], [Beckmann，3]Peyré 和 Cuturi，2019，第 6 节；我们将其用作基线。对于某些特殊类别，存在闭式解[Münch 和 Wojciechowski，2019，Bhattacharya 和 Mukherjee，2020，Kelly，2020，Hehl，2026，Jung 等人， 2026]。DasGupta 等人 [2023] 注意到局部距离属于 $\{0,1,2,3\}$；对于惰性随机游走和相等度数，他们将问题表示为一个最小权完美匹配问题，其代价属于 $\{1,2,3\}$，再将不等度数约化为此类匹配；当较小的d_i+1 不能整除较大的 d_j+1 时，会产生加性误差（其定理 3），并界定局部算法所需查询代价矩阵的次数。

[1]这涵盖了 $m_u(u)=p$和 $m_u(x)=(1-p)/d_u$对于 $x\in\mathcal{N}(u)$：在 $p=0$处的非惰性随机游走，以及在 $p=1/(d_u+1)$ 处的惰性随机游走，即 $\mathcal{N}[u]$上的均匀测度。

3 小整数度量上的传输是一个最小割

我们先针对有限度量空间（X，距离）陈述该约化，然后再对其进行特化。这是将带有差分约束和有界变量的整数规划约化为分层图上的闭包的经典约化的一个实例 [Hochbaum，2001，Ahuja 等人，2003，Ishikawa，2003]；这里的关键在于它在图邻域上的有效性。令 μ 为 X 上的一个符号测度，满足 $\mu(X) = 0$，并且对所有（x，y）$\in P \times N$，均有 dist(x，y) $\in \{1, \dots, D\}$。

定义 1（分层闭包网络）。分层闭包网络 $\mathsf{G}_D(\mu)$ 是一个带容量有向网络，其源点为 s，汇点为 t，节点为 $a_{x,\tau}$（$x \in P$，$\tau \in \{2, \dots, D\}$）以及 $b_{y,\sigma}$（$y \in N$，$\sigma \in \{1, \dots, D-1\}$）。其弧包括容量为 $\mu(x)$ 的 $\mathsf{s} \to a_{x,\tau}$，容量为 $\nu(y)$ 的 $b_{y,\sigma} \to \mathsf{t}$，以及具有无限容量的链 $a_{x,\tau+1} \to a_{x,\tau}$（$\tau < D$）和 $b_{y,\sigma+1} \to b_{y,\sigma}$（$\sigma < D-1$），连同蕴含关系 $a_{x,\tau} \to b_{y,\tau-\mathrm{dist}(x,y)}$：对于 ($x$，$y$) $\in P \times N$ 以及 $\tau \in \{$dist(x，y) $+1$，……，$D\}$。

定理 1。若 F 是 G_D 中最大 s–t 流的值，则 $\mathcal{W}_1(\mu^+, \mu^-) = DA - F$。

证明概要。式 (1) 的约束矩阵是全幺模的，因此 (1) 存在整数最优解（见 Bhattacharya 和 Mukherjee，2020，$u \in$引理。），该解 $\{1, \dots, D\}^P$ $v \in \{0, \dots, D-1\}^N$ $u_x - v_y \le$ x y 可归一化为 和 。在一元编码下，dist(，) 转化为定义 1 中的蕴含关系，因此可行势是网络的闭集，节点 $a_{x,\cdot}$ 上的权重为 $\mu(x)$，节点 $b_{y,\cdot}$ 上的权重为 $-\nu(y)$，而目标函数是 A 加上闭集的权重。最大权闭集的权重为 $(D-1)A - F$ [Picard，1976]。详见附录 A。 □

只有距离小于 D 的点对才会生成弧，且不存储任何代价：网络需要关系“dist(x, y) $\le \tau$”来表示 $\tau < D$，而图邻域使这一操作成本低廉。

引理 2。设 $(i, j) \in E$，$x \in \mathcal{N}[i]$，$y \in \mathcal{N}[j]$，$x \ne y$。则 $\mathrm{dist}(x, y) \le 2$，除非 $x \in \mathcal{U}_i$ 且 $y \in \mathcal{U}_j$；在这种情况下，当 $xy \in E$ 时，$\mathrm{dist}(x, y) = 1$，当 $xy \notin E$ 且 $\exists z$ 时为 2，否则为 3。

由于 $P \subseteq \mathcal{N}[i]$ 且 $N \subseteq \mathcal{N}[j]$ 对于一步测度，定理 2 意味着 $\mathrm{dist}(x, y) \in \{1, 2, 3\}$ 在$P \times N$上成立，因此定理 1 可取 $D = 3$。

推论 3（一步测度）。对于一步测度 m_i, m_j，$\mathcal{W}_1(m_i, m_j) = 3A - F$，其中$\sim F$ 是从 $G_3(\mu)$ 中删除弧 $b_{y,2} \to b_{y,1}$ 后得到的 G_3，的最大流值。若在 $P \times N$ 上 $dist(x, y) \le 2$，则对于二分网络 $G_2(\mu)$ 的最大流值 F_1，有 $\mathcal{W}_1(m_i, m_j) = 2A - F_1$；若没有任何一对相邻，则 $\mathcal{W}_1(m_i, m_j) = 2A$。

弧 $b_{y,2} \to b_{y,1}$ 是冗余的：所有指向 $b_{y,2}$ 的弧都来自某个 $a_{x,3}$，且该节点满足 $xy \in E$；这已经通过 $a_{x,2}$ 推出$b_{y,1}$，因而~（附录 A）。因此，G_3有 $2(|P| + |N|)$ 个节点和 $3|P| + 2|N| + 2e_1 + e_2$ 条弧，其中 e_1, e_2 分别统计 $P \times N$ 中距离为 1、2 的节点对（图 1）。对于 $\mathrm{supp}\, m_u \subseteq B_r(u)$（均匀球，Ollivier，2009；或 r 步游走），所有距离都至多为 $2r+1$，且适用 G_{2r+1}。

推论 4（极小极大公式）。对于单步测度和 $X \subseteq P$，令 $N_k(X) = \{y \in N$：距离（x, y）$\le k$ 对于某个 $x \in X\}$。则

$$\mathcal{W}_1(m_i, m_j) = A + \max_{X_3 \subseteq X_2 \subseteq P} \Big[\mu(X_2) - \nu\big(N_1(X_2) \cup N_2(X_3)\big) + \mu(X_3) - \nu\big(N_1(X_3)\big) \Big].$$

这里 X_2，X_3 是度至少为 2 和 3 的候选顶点。对于度相等的情况，已知匹配可确定曲率 [DasGupta 等人，2023年，第 2 节。2], [Hehl，2026，Jung 等人，2026]；定理 4是每个单步测度的对偶形式。

命题 *5*（证书）。令 Z 为最大流在 $G_D(\mu)$ 的残量网络中从 s 可达的节点。则 $u_x = 1 + |\{\tau : a_{x,\tau} \in Z\}|$ 和 $v_y = |\{\sigma : b_{y,\sigma} \in Z\}|$ 可解出（*1*），并且在弧为 $s \to x$ *(*$\mu(x)$*)*、$y \to t$ *(*$\nu(y)$*)* 以及 $x \to y$ *(*∞*)* 的二分网络中，对于 $u_x - v_y =$ 距离（x, y）的最大流可得到一个最优传输计划。

检查证书需要计划的边缘分布、其支撑集上的距离，以及（1）的约束；对于单步测度，只需根据邻接关系和共同邻居查询检查满足 $u_x - v_y \ge 2$ 的配对。图重连需要这样的计划 [Nguyen 等人，2023]。

4 邻域中的网络线性化

~G_3在距离 2 的每一对顶点之间仍有一条弧，且 e_2可以达到 $|P||N|$阶；然而，这些顶点对是成块出现的：所有与某个公共顶点相邻的顶点对 w 之间的距离都不超过 2，并且一个零权节点就能承载它们的蕴含关系。

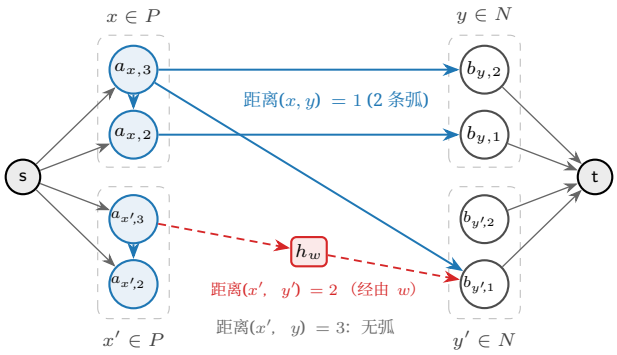


图 1：G_3（定理 3），针对 x、$x' \in P$ 和 y、$y' \in N$，按顶点分组（虚线框）。灰色弧带有有限容量（μ 从 s 出发，ν 进入 t；同一顶点两层上的容量相同）；蓝色实线弧具有无限容量，用于编码垂直链 $a_{x,3} \to a_{x,2}$ 以及距离蕴含关系（距离为 $=1$ 时有两条弧，距离为 $=2$ 时有一条，距离为 $=3$ 时没有弧）。红色虚线：中继节点 h_w（定理 6），用于替代所有 $P \times N$ 中共享公共邻居 w 的顶点对之间直接连接的距离为 2 的弧（定理 7）。

命题 6 (中继节点)。设 $P' \subseteq P$ 和 $N' \subseteq N$，对于所有 $(x, y) \in P' \times N'$ 均有 距离$(x, y) \le 2$。添加一个权重为 0 的节点 h，并添加无限容量弧 $a_{x,3} \to h\,(x \in P')$ 和 $h \to b_{y,1}\,(y \in N')$，同时删除弧 $a_{x,3} \to b_{y,1}$（对于 $(x, y) \in P' \times N'$），不会改变最大流值；即使逐个添加中继节点，结论也同样成立。

若 $w \in V$（位于 S 内部或外部）在 P 和 N 中都拥有邻居，则称其为 $(i,\ j)$ 的见证点。令 Ω 为见证点集合，$\ell_{ij} = d_i + d_j + \sum_{x \in P \cup N} d_x$ 为局部输入，即下文读取的邻接项数量。

定义 2（见证网络）。G^w 包含以下节点：G_3 中的节点，以及每个 $w \in \Omega$ 对应的一个节点；还包含 G_3 中的弧，但不包括 $a_{x,3} \to b_{y,1}$ 距离为 2 的节点对之间的弧。这些弧被 $a_{x,3} \to h_w$（$x \in P \cap \mathcal{N}$）对应的弧和 $h_w \to b_{y,1}$（$y \in N \cap \mathcal{N}$）对应的弧所替代，容量均为无限容量。

定理 7。对于一步测度，*(i)* $\mathcal{W}_1(m_i, m_j) = 3A - F^w$，其中 F^w 是G^w的最大流值；*(ii)* G^w 至多包含 $3\ell_{ij} + 2$ 个节点和 $5\ell_{ij}$ 条弧；*(iii)* 给定邻接表后，G^w 可在 $O(\ell_{ij})$ 时间内构建完成，并使用初始化一次、随后为每条边重复使用的 $O(|V|)$ 工作空间。

根据定理 2，$P \times N$ 中距离为 2 的每一对节点都共享一个见证点 $w \in \Omega$，因此将定理 6 应用于每个 $w \in \Omega$ 会将 G_3 转换为G^w（第 A节）。

对于整数质量（公分母 Λ，第 5 节）以及上限设为 $2A+1$ 的无限容量，在 G^w 上求最大流需要 $\ell_{ij}^{1+o(1)} \log \Lambda$ 时间，并且以高概率成立 [Chen 等人，2025] 或确定性地成立 [van den Brand 等人，2023]——因此，$\ell_{ij}^{1+o(1)}$ 只要 Λ 在 ℓ_{ij}（惰性、非惰性、LLY 和固定惰性参数测度）中为多项式时间，就无需列出距离为 2 或 3 的点对。

图最小费用流。Beckmann公式在 ij 及与 $P \cup N$ 相邻的边上（定理13）已经构成了一个至多包含 $\ell_{ij}+1$ 条边的近线性时间最小费用流问题 [Chen 等人，2025]。定理 7则消除了边费用，将 $\mathcal{W}_1$ 转化为最大流问题，在实践中速度更快（第 6 节）。

与 DasGupta 等人 [2023] 的关系。他们关于 $n^2/6$ 期望查询次数的下界，针对的是完全二分费用图 $H = (L, R, w)$ 中一个等度边（$|L| = |R| = n$、$w =$ 距离 $\in \{1, 2, 3\}$）的查询。他们的命题 2 通过一个具有 H 条边的图实现了 G_H，其中包含 $2n + 1 + e_1 + 2e_2$条边（e_1、e_2 计入权重为1 和权重为2 的点对），并且在其下界实例中具有 $e_2 = 0$ 和 $e_1 \le n$。

命题8。*(i)* 对于 G_H和惰性随机游走，G^w 有 $4n + e_2 + 2$ 个节点和 $5n + 2e_1 + 2e_2$ 条弧，并可从邻接表在 $O(n + e_1 + e_2)$ 时间内构建；在单位质量下，H 的完美匹配的最小权重等于 $3n$ 减去 G^w的最大流。*(ii)* 对于其下界证明中的实例，*Dinic* 算法至多经过一个阶段即可找到最大流，而精确曲率需要 $O(n)$ 时间。

该 $\Omega(n^2)$ 界反映了查询黑盒距离矩阵的成本：对于邻接表，相同实例需要 $O(n)$ 时间（表 1）。此外，尽管 (i)也适用于平衡匹配

通过对 $\{1,2\}$ [Kao 等人中的权重进行分解， 2001], 定理 1 和 7 适用于不等的度以及任意单步测度，而其匹配归约仅在 $\min\{d_i,d_j\}+1$ 整除 $\max\{d_i,d_j\}+1$时才对惰性随机游走精确。

定理 9（曲率至少与二分图匹配问题一样难）。设 $B=(L,R,E_B)$ 为一个含有 $|L|=|R|=n$ 的二分图，并令 G_B 由边 (i,j)、从 i 到 L 中每个顶点的边、从 j 到 R 中每个顶点的边，以及边 E_B 组成。对于惰性随机游走，$\mathcal{W}_1(m_i,m_j)=\big(3n-2\alpha'(B)\big)/(n+2)$，其中 $\alpha'(B)$ 是 B 的最大匹配大小，且 $\ell_{ij}=4n+2+2|E_B|$。

因此，运行时间为 $T(\ell_{ij})$ 的算法可在 $T(O(n+|E_B|))+O(n+|E_B|)$ 时间内求得 $\alpha'(B)$；而惰性随机游走在 $d_i=d_j$ 条件下的界 $\ell_{ij}^{1+o(1)}$（定理 7）和 $O(\ell_{ij}\sqrt{|\mathcal{U}_i|})$（定理 10），与二分图匹配已知的界相同 [Chen 等, 2025，Hopcroft 和 Karp， 1973]。除非二分图匹配可在线性时间内完成，否则不存在能在线性时间内计算、且对每条边都等于曲率的公式。G_B 是 DasGupta 等人的图 G_H [2023, 命题 2]，其权重取自 $\{1,3\}$。

5 Distance-FreeSolver

Theo剩余 7 为网络的界，但对于大多数实际邻域，可以使用更小的界。T
he 求解器
r（算法 1，第 B 节）结合了这两种构造。

精确质量。行是 $\mathcal{N}[i]$ 的顶点，列是 $\mathcal{N}[j]$ 的顶点，其中 $d_i\ge d_j$。质量通过公分母 Λ 缩放为整数，例如 (d_i+1) (d_j+1) 用于惰性随机游走，以及 $b\,d_i d_j$ 用于闲置率 a/b（第 B 节）。流为整数，并且 $\mathcal{W}_1=(3A-F)/\Lambda\in\mathbb{Q}$；具有 $\Lambda>2^{50}$ 的边会被跳过并标记。

位集和块中继。列按 $\lceil(d_j+1)/64\rceil$ 机器字进行索引。遍历各列的列表一次，即可为每个 $w\in V$ 列邻居，得到其列邻居的位集 Col(w)。行 x 与列 Col(x) 相邻，并且与 Cov(x) $=\bigcup_{w\in\mathcal{N}(x)}$ Col(w) 中的顶点距离为 2；这些信息按字逐一累积，并在 Cov(x) $\supseteq\mathcal{U}_j$ 时停止；根据定理 2，这些信息给出了所有距离。大多数距离为 2 的顶点对预先构成已知的块：每一行与 $\{i,j\}\cup\mathcal{C}$ 中的每一列距离均不超过 2，每个 $\{i,j\}\cup\mathcal{C}$ 中的行与每一列距离均不超过 2，完整行（Cov(x) $\supseteq\mathcal{U}_j$）连接 $\mathcal{U}_j$ 中的所有顶点，完整列连接 $\mathcal{U}_i$ 中的所有顶点。四个中继 H_1、……、H_4（定理 6）用于处理这些块；只有其余的顶点对需要单独的弧。

见证中继与有界位集运算。当剩余距离为2 的节点对很多时，只要能使网络至少缩小 $4\times$（通过快速上界进行检查），就会将它们经由见证中继（定义 2）进行路由。在大型稀疏邻域中，如果位集每个邻接项需要 $\lceil(d_j+1)/64\rceil$个字，但找到的距离为2 的节点对很少，我们会将每个行／列邻接项的位集运算限制为4个字；如果构建或累积位集的运算量超过此限制，且大多数行并非完整版本，求解器就会直接回退到G^{w}。

孪生节点、快速路径、热启动与执行。对 N的邻接关系、覆盖和块都相同的行具有相同的后继节点，因此会将它们合并，并累加其容量（定理 12）；枢纽的所有叶节点都会合并为一行。如果 $\mathcal{U}_i$ 的每一行都是完整版本，则不存在距离为3 的节点对，一次二分流即可满足要求（定理 3）。否则，在$\widetilde{\ }$的两层上进行路由的单层邻接网络最大流，是G$_3$ 的可行流，Dinic [1970] 算法将以此为起点。各边并行地由多个线程处理，每个线程使用可增长内存区，且不会为每条边单独分配内存。

Dinic 算法的阶段。除中继节点外，每个节点携带的质量最多是其顶点质量的两倍，并且任意两个中继节点都不相邻 [Even 和 Tarjan， 1975], [Feder 和 Motwani，1991，定理 4。2]；该界限定了阶段数。

$\widetilde{\ }$**命题 10**（阶段）。设 A 为最小整数标度中的总净质量。在 G$_3$, 上的 G^{w}以及由算法 *1*构建的每个网络上，*Dinic* 算法至多需要 $2\sqrt{24A}+1$ 个阶段，并且需要$O(\sqrt{A}\,m+A\log A)$ 时间，其中 m 为弧的数量。对于具有 $d_i=d_j$的 $A=|\mathcal{U}_i|$惰性随机游走，且计算需要 $O(\ell_{ij}\sqrt{|\mathcal{U}_i|})$ 时间，在 G^{w}上。

I增量更新。许多算法在每次迭代中只会改变少数几条边（例如重新布线）

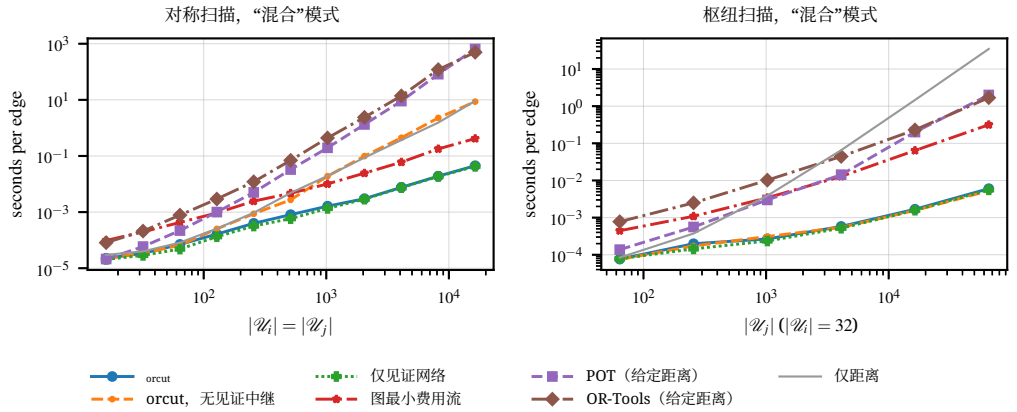


图2： 在半数 $\mathcal{U}_i \times \mathcal{U}_j$ 距离为 2 的邻域上每条边耗时（另见图 3）。传输求解器使用给定的距离（灰色部分：其构造过程）；其他耗时则包括所有步骤。

引理11 （局部性）。 设通过插入或删除顶点对（a_l、b_l）、$l = 1$、……、k，从 G 得到 G'，并且每个顶点的度量仅取决于其邻域（上述所有度量均满足此条件，对于加权游走，未改变的边具有固定权重）。G'中一条边（i、j）的曲率，只有在 i 或 j 位于$U = \bigcup_l (\mathcal{N}[a_l] \cup \mathcal{N}[b_l])$ 中（在 G'中取）时，才可能不同于其在 G 中的值。

6 实验

我们评估精确性（E1）、逐边标度（E2）、完整图速度（E3）、消融实验（E4）、近似误差（E5）以及其他度量（E6，第 D 节）。除非另有说明，我们在 Slurm 集群的一个节点上使用 32 个线程运行惰性随机游走（第 C 节）。

基线与数据。`POT` 的网络单纯形法 [Flamary 等人， 2021]（C++ 和 `ot.emd2`）； `OR-Tools` 的整数最小费用流 [Perron 和 Furnon， 2024]； `HiGHS` [Huangfu 和 Hall， 2018]；图本身上的最小费用流（定理13； `OR-Tools`）；默认 `GraphRicciCurvature` 0。6.1（GRC；Ni等人，2019年）；熵正则最优传输 [Cuturi， 2013]。传输求解器接收由我们的最快代码构建的局部距离，并单独计时。在完整图上，对 `POT` 的分层抽样边进行计时，外推后再除以线程数，得到其并行时间的一个下界。

数据：20 个真实网络和带种子的随机图（第 C 节）。代理方法包括 BF，即 Topping 等人 [2022] 提出的平衡 Forman 曲率$_3 = 4 - d_i - d_j + 3|\mathcal{C}|$；Fesser 和 Weber [2024提出的]三角形增强 Forman 曲率；以及Jost 和 Liu [2014，（下文简称 JL）] 均值（JL 非惰性曲率界的中点），即 Tian 等人 [2025]提出的代理方法。

6.1 精确性（E1）

在 19 个图和 5 种测度（惰性、非惰性、LLY、闲置率 $1/2$、加权惰性游走）上，算法1在所有已检查的边上均与 `OR-Tools` 390,548 `POT` $3.6\times$ 的整数 10^{-14} 最小费用流一致（最多可达，且对于惰性、非惰性和 LLY 曲率，还与第 B 节的预言机以及定理 1 的分层闭包在每条边上均一致。所有 56,880 个证书均已验证（表 8）。见证网络和 E4 的每种配置均返回相同的值。

6.2 单条边成本（E2）

图2分离出具有指定 $|\mathcal{U}_i|$、$|\mathcal{U}_j|$、公共邻居和距离结构（五种状态，C节）的单条边。在 $|\mathcal{U}_i| = |\mathcal{U}_j| = 16$ 384 处，给定距离后（包括构造它们的时间，最多为 15,131$\times$ ），求解器比 `POT` 快 6,603$\times$–14,492$\times$ 倍，比 `OR-Tools` 快最多 $2.33 \times 10^5\times$ 倍，并且比图最小费用流快 5.8$\times$–16$\times$ 倍；对于“枢纽”边，给定距离后，求解器最多快 1,609$\times$ 倍于 `POT`。一个标志位可以关闭见证中继、位集运算上限以及回退到 G^{w}；在该规模下，不关闭它们时，求解器会慢 28$\times$–192$\times$ 倍（表 9）：大多数属于“混合”机制，其中许多点对的距离为 2，且只有少数行是完整版本；此外还属于“树”机制，其中没有点对

表 1： DasGupta 等人的下界实例 DasGupta 等人 [2023] 及相关族（第 C 节）：最大 n 时的局部输入、弧数和时间，以及我们的求解器和 POT 在最大 n 时的时间；其 n^2 距离符合内存限制；k：我们求解器时间的拟合指数 $\propto n^k$。弧数：求解器构建的网络中的弧数，经过孪生合并及第 B 节的其他约简之后；定理 8 (i) 统计这些处理之前的 G^w。

				最大 n		$n = 65,536$		
族	n	ℓ_{ij}	arcs	orcut	图流	orcut	POT + 距离	k
一对，位于距离 1	1,048,576	4,194,308	2,097,160	232 毫秒	2.40 秒	12.2 毫秒	500 秒	1.0
完美匹配距离为 1	1,048,576	6,291,458	7,340,032	1.58 秒	4.43 秒	59.1 毫秒	482 秒	1.2
有界加权度	1,048,576	12,582,914	11,534,336	158 秒	83.9 分钟	770 毫秒	9.71 小时	1.6
$\sqrt{n}$块位于距离 2	1,048,576	6,291,458	4,485,100	1.18 秒	32.3 秒	35.9 毫秒	5.14 小时	1.1

的距离为 2，代价是未封顶位集计算。在稠密区域中，这些机制一个都不使用，因此差异只是噪声。在所有受控边上，见证网络每单位局部输入至多包含 1.7 条弧（定理7：5），并且在 $\ell_{ij}^S = \sum_{x \in S} d_x$ 中以指数 1 增长，而显式点对最多为 1.9（表 9）。在实图上，按度数层在边中的占比加权后，18 个图的增益中位数为 $2.4\times$，而 POT 在给定距离后为（$7\times$ 包括构造它们的时间），并且 $19\times$ 相较于图最小费用流。

下界实例及其他。表 1 将四个族扩展到 $n = 1{,}048{,}576$：一对处于距离 1 [的顶点和一个完美匹配、DasGupta 等，2023，定理 4], 图 G_H，其加权度有界（这是其近似算法所属的类别，即定理 5），以及在此基础上，$G_{H,\sqrt{n}}$ 个共享中间顶点使 $n^{3/2}$ 对顶点处于距离 2，并通过 $O(n)$ 条边相连。我们的时间增长为 n^k，其中 k 介于 1 和 1.6 之间；最慢的族在最大 n 时耗时 158 秒；POT 在 $n = 65{,}536$ 时就已经慢 $156\times$–$5.16 \times 10^5\times$ 倍，之后距离超过内存限制，而图最小费用流则慢 $2.8\times$–$32\times$ 倍。

6.3 完整图（E3）与消融（E4）

表 2 报告了 20 个真实网络中的每条边；其中最大的 Orkut 含有 1.17×10^8 条边，耗时 49.6 小时。相对于 POT 流程的下界，加速比为 $3.2\times$–$506\times$（中位数为 $6\times$）；在具有枢纽的图上加速最大，此时孪生合并可将最多 33,375 行压缩为一行，而在稠密的聚类邻域中加速最小。中位数为 10.4% 的边采用单层快速路径，最多 3.7% 使用见证中继。在表 2 中除 Orkut 之外的 19 个网络上，非惰性和 LLY 曲率的计算成本至多是惰性曲率的 $1.1\times$；在 E6 的图上至多是 $1.3\times$（表 11）。求解器在 32 个线程上可扩展 $12\times$–$22\times$（DBLP 上为 $22\times$，GitHub 上为 $18\times$，YouTube 上为 $16\times$，LiveJournal 上为 $12\times$）。在随机图上（图 5 和表 10），在平均度固定时，时间随 m 大致线性增长；而在枢纽随 n 增长的图上，增长更快（Barabási 和 Albert 图以及 Chung–Lu 图），因为每条边的局部输入（定理 7）会随度增长。

表 3 展示了第 5 节各步骤的启用和停用情况。仅在位集上运行双层割的中位耗时比完整求解器$1.7\times$慢（最高达 $3.1\times$）；在稠密邻域中，中继节点很重要；在具有枢纽的图中，孪生合并很重要（最高达 $2.1\times$）；热启动的效果最高达 $1.9\times$。在这些图上，见证中继带来的变化很小（$0.96\times$–$1\times$）：实际邻域较小或呈聚类状，位集运算的界未达到，因此见证中继适用于 E2 中的大型稀疏邻域。线性规模并不能说明全部情况：仅使用见证网络的中位耗时比完整求解器 $3\times$ 慢；图最小费用流$10\times$–$39\times$（中位数 $23\times$），而距离加上 POT $3.8\times$–$127\times$，则适用于 8 个能装入内存的图。

6.4 近似管线（E5）

采用默认设置的 GRC 在大多数边上都能达到浮点数精度，但它对每个顶点只保留 3，000 个邻居；当两个端点的度都超过 2，000 时，则切换到 Sinkhorn。在这类枢纽较多的图上，其误差可达 0.40，速度中位数 $441\times$ 比采用相同线程数的求解器慢。

表 2： 所有边的惰性曲率（32 个线程）。 `POT`：距离的下界估计加网络单纯形法（第 C 节）。内存：预热后计时运行期间的常驻集增长（MB，每 50 毫秒采样一次），不包括复用的内存（0：除一页外没有增长）。孪生节点：在端点度为 ≥ 100 的边上，每个合并行对应的行数。

图	n	m	$d_{\max}$	`orcut`	`POT`（估计）	加速比	内存	孪生节点
Deezer	28,281	92,752	172	57.4 毫秒	274 毫秒	4.8×	5.8	25
Amazon-Photo	7,487	1.19×10^5	1,434	1.62 秒	9.79 秒	6.1×	429	2.5
Twitch-DE	9,498	1.53×10^5	4,259	7.67 秒	59.9 秒	7.8×	1,042	9.4
Crocodile	11,631	1.71×10^5	3,546	2.30 秒	303 秒	131×	220	63
Facebook	22,470	1.71×10^5	709	1.29 秒	4.83 秒	3.7×	0	4.3
Enron	33,696	1.81×10^5	1,383	3.68 秒	17.0 秒	4.6×	33	75
AstroPh	17,903	1.97×10^5	504	1.24 秒	4.70 秒	3.8×	0	4.5
Squirrel	5,201	1.98×10^5	1,903	22.2 秒	21.6 分钟	59×	855	4.8
Brightkite	56,739	2.13×10^5	1,134	1.28 秒	5.80 秒	4.5×	1.3	23
Amazon-Computers	13,381	2.46×10^5	2,992	8.10秒	71.7秒	8.9×	925	2.6
Coauthor-Physics	34,493	2.48×10^5	382	514 毫秒	1.66 秒	3.2×	0	3.1
GitHub	37,700	2.89×10^5	9,458	14.1 秒	391 秒	28×	1,396	93
Epinions	75,877	4.06×10^5	3,044	28.3 秒	96.2 秒	3.4×	1,105	29
亚马逊	3.35×10^5	9.26×10^5	549	332 毫秒	1.99 秒	6.0×	57	52
DBLP	3.17×10^5	1.05×10^6	343	653 毫秒	4.82 秒	7.4×	64	23
roadNet-CA	1.96×10^6	2.76×10^6	12	355 毫秒	4.16 秒	12×	168	–
YouTube	1.13×10^6	2.99×10^6	28,754	182 秒	3.10 小时	61×	6,658	388
Skitter	1.69×10^6	1.11×10^7	35,455	43.4 分钟	15.3 天	506×	14,939	66
LiveJournal	4.00×10^6	3.47×10^7	14,815	42.9 分钟	3.58 小时	5.0×	14,827	9.1
Orkut	3.07×10^6	1.17×10^8	33,313	49.6 小时	7.91 天	3.8×	52,547	2.8

对于局部问题本身，在 GRC 的闲置率 $1/2$（图 4）下，熵正则最优传输在 $\varepsilon = 0.1$时，GRC 的选择[2], 平均绝对误差的中位数为 6.4×10^{-3}，耗时为我们的 $12\times$；而在 $\varepsilon = 0.01$ 时，误差可忽略不计，为5，$565\times$。

[2]GRC 自身的 Sinkhorn 方法在每条边上运行，并将邻居截断至 3,000 个，其平均绝对误差高于表 4中的数值，闲置率也为 $1/2$。

表3： 相对于完整求解器的单线程时间，根据边的分层抽样进行外推（第 C 节中的图；第 B 节中的预言机）。–：未估算（预言机在 GitHub 上运行超时； `POT` 的局部问题在 CL2.1 和 GitHub 上超过了内存限制）。

配置										
双层割，位集	1.3	1.7	1.0	2.6	1.5	2.3	2.2	3.1	1.5	1.7
+ 中继节点 H_1, ..., H_4	1.3	1.6	1.1	1.8	1.3	1.6	2.1	2.9	1.4	1.5
+ 孪生合并	0.99	0.99	0.97	1.3	1.2	1.2	1.8	1.3	1.4	1.0
+快速路径	0.99	0.99	0.97	1.3	1.2	1.2	1.8	1.2	1.4	0.98
+ 热启动	1.0	1.0	1.0	1.0	0.99	0.98	0.99	0.99	0.96	0.99
+ 见证中继（orcut）	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
完整版本 – 所有中继	1.0	1.1	1.0	1.5	1.2	1.4	1.2	1.1	1.2	1.0
完整版本 – 孪生节点	1.3	1.7	1.1	1.5	1.1	1.4	1.1	2.1	1.0	1.4
完整版本 – 快速路径	0.99	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.98	1.1
完整版本 – 热启动	1.0	0.98	0.99	1.2	1.2	1.2	1.9	1.3	1.4	0.98
仅见证网络	1.1	1.8	2.4	4.2	2.8	3.7	3.3	3.4	2.9	3.2
图最小费用流	30	39	10	30	19	24	23	22	20	23
预言机	3.3	7.2	5.2	373	11	–	59	150	21	8.6
距离矩阵 + POT	7.6	8.5	4.5	–	3.8	–	16	127	6.4	8.9

表4： 在15个图上将近似方法与精确曲率进行比较：GRCmethods（第 C节）；按排名与 LLY 曲率进行比较的代理方法，以及 JL 均值与其所估计的非惰性曲率之间按误差进行的比较。平均绝对误差：各图上的中位数；最大值：最差边；符号：符号不同的边所占比例；ρ：斯皮尔曼；rec.：恢复出的 1% 负曲率边中最负的边所占比例。

方法	MAE	max	sign	ρ	rec.
GRC 默认方法	1.0×10^{-14}	0.40	0.0%	1.00	100%
GRC OTD	4.1×10^{-15}	0.34	0.0%	1.00	100%
GRC OTD ，所有邻居	1.3×10^{-14}	0.037	0.0%	1.00	100%
GRC Sinkhorn	9.6×10^{-3}	0.40	1.3%	0.998	99%
GRC ATD	2.8×10^{-1}	0.98	17.6%	0.755	45%
JL 均值	2.9×10^{-1}	0.99	5.9%	0.734	12%
BF	–	–	14.3%	0.696	15%
AF_3	–	–	14.2%	0.388	4%

代理方法成本低，但其边排序与它们所代表的曲率不同（见表4）。在 1%曲率最低的边中，BF、AF_3, 和 JL 均值分别只能找回 15%、4% 和 12%。我们不衡量图重连等下游方法的准确性，因为这些方法带来的收益仍有争议[Tori 等人， 2025]；近似方法改变的是其作用于哪些边。

7 局限性与结论

该约化需要在较小范围内使用整数距离。定理 1 适用于整数边长，其中 D 表示最大局部距离；而实值边长（如里奇流中的边长） [Ni等人提出的 2019], 则不在涵盖范围内。PyPI 软件包的 1.3.1 版本 `faster-or-curvature` 通过在局部问题上使用网络单纯形法实现了实数边长，并通过使用实数容量的同一个割，以浮点数实现了实数游走权重。本文不讨论这一版本，因为它增加的是工程实现，而非新的公式；在该软件包的基准测试中，它在加权网络上比 2.6 至 23 快 `GraphRicciCurvature` 0.6.1 倍，同时结果值达到 10^{-13}。当 $|P||N|$ 的邻居 i 和 j 具有较大度数时，局部输入可能超过 $|P||N|$；而我们的最大流采用 Dinic 算法，其运行时间随总质量的平方根增长（定理10），而不是接近线性增长。对于 $r \geq 2$，我们的实现仍会构建距离。在这些限制范围内，OR 曲率不一定是基于曲率的图学习中最昂贵的部分：每条边只需进行一次最小割即可得到其数值、一个证书以及边变化后的更新规则；而在我们的设置中，用于避免最小割的近似方法要么更慢，要么会以不同方式排列边，遗漏许多基于曲率的算法所处理的负曲率边。

致谢

我们感谢阿尔托科学信息技术项目提供的计算资源。

参考文献

Ravindra K. Ahuja、Thomas L. Magnanti 和 James B. Orlin。网络流：理论、算法与应用。Prentice-Hall, Inc., 美国，1993年。国际标准书号 013617549X。2，13，16Ravindra K. Ahuja、Dorit S. Hochbaum 和 James B. Orlin。求解凸成本整数对偶网络流问题。《管理科学》，49(7):950–964，2003年。doi: 10.1287/mnsc.49.7.950.16384。3Albert-László Barabási 和 Réka Albert。随机网络中标度的出现。《科学》，286(5439): 509–512，1999年。doi: 10.1126/science.286.5439.509。7，19Martin Beckmann。运输的连续模型。《计量经济学》，

20 4 643 660, 1952 00129682, ( ): – 。

14680262 http://www.jstor.org/stable/1907646 2, 15

ISSN 。URL 。Bhaswar B. Bhattacharya 和 Sumit Mukherjee。关于图粗里奇曲率的精确与渐近结果，2020年。URL https://arxiv.org/abs/1306.6741。1，2，3Nicolas Bonneel、Michiel van de Panne、Sylvain Paris 和 Wolfgang Heidrich。使用拉格朗日质量输运进行位移插值。《*ACM*图形学汇刊》，30(6):1–12，2011年12月。ISSN 0730-0301。doi: 10.1145/2070781.2024192。2

D. P. Bourne、D. Cushing、S. Liu、F. Münch 和 N. Peyerimhoff。图的Ollivier–Ricci闲置函数。SIAM离散数学杂志，32(2):1408–1424，2018年。doi: 10.1137/17M1134469。2 Li Chen、Rasmus Kyng、Yang P. Liu、Richard Peng、Maximilian Probst Gutenberg 和 Sushant Sachdeva。近线性时间内的最大流和最小费用流。《ACM计算机学会志》，72(3)，2025年5月。ISSN 0004-5411。doi: 10.1145/3728631。4，5 Eunjoon Cho、Seth A. Myers 和 Jure Leskovec。友谊与移动性：基于位置的社交网络中的用户移动。在第17届ACM SIGKDD知识发现与数据挖掘国际会议论文集，KDD '11，第1082–1090页，美国纽约州纽约市，2011年。Association for Computing Machinery。国际标准书号 9781450308137。doi: 10.1145/2020408.2020579。21 Fan Chung 和 Linyuan Lu。具有给定期望度数序列的随机图中的连通分量。《组合学年刊》，6(2):125–145，2002年11月。ISSN 0219-3094。doi: 10.1007/PL00012580。7，19 Marco Cuturi。Sinkhorn距离：以极快速度计算最优传输。在第27届神经信息处理系统国际会议论文集——第2卷，NIPS'13，第2292–2300页，美国纽约州雷德胡克，2013年。Curran Associates Inc.。1，2，6 Bhaskar DasGupta、Elena Grigorescu 和 Tamalika Mukherjee。图的离散化Ricci曲率计算：局部算法与（局部化）细粒度归约。《理论计算机科学》，975(C)，2023年10月。ISSN 0304-3975。doi: 10.1016/j.tcs.2023.114127。1，2，3，4，5，7，16，20 E. A. Dinic。求解网络最大流问题的算法及其复杂度估计。《苏联数学文摘》，11:1277–1280，1970年。5 Paul Erd˝os 和 Alfréd Rényi。关于随机图 I。《德布勒森数学出版物》，6:290–297，1959年。doi: 10.5486/pmd.1959.6.3-4.12。19 Shimon Even 和 R. Endre Tarjan。网络流与图连通性测试。《SIAM计算杂志》，4(4): 507–518，1975年12月。ISSN 0097-5397。doi: 10.1137/0204043。5，16 Tomás Feder 和 Rajeev Motwani。团划分、图压缩与算法加速。在第二十三届ACM年度计算理论研讨会论文集，STOC '91，第123–133页，美国纽约州纽约市，1991年。Association for Computing Machinery。国际标准书号 0897913973。doi: 10.1145/103418.103424。1，5 Lukas Fesser 和 Melanie Weber。通过局部曲率剖面实现有效的结构编码。在 B. Kim、Y. Yue、S. Chaudhuri、K. Fragkiadaki、M. Khan 和 Y. Sun（编），国际学习表征会议，第

2024， 22571 22589, 2024 https://proceedings.iclr.cc/paper_ 卷页码 – files/paper/2024/file/6188c02ccc16a7587716de2efd754033-Paper-Conference.pdf l a。URL 。Lukas Fesser 和 Melanie Weber。利用 Forman-Ricci 曲率增强缓解过平滑和过压缩。在 Soledad Villar 和 Benjamin Chamberlain（编），第二届图学习会议论文集，《机器学习研究论文集》第231卷，第19:1–19:28页。PMLR，2024年11月b。URL https://proceedings.mlr.press/v231/fesser24a.html。1，6 Lukas Fesser、Sergio Serrano de Haro Iváñez、Karel Devriendt、Melanie Weber 和 Renaud Lambiotte。Forman-Ricci曲率的增强及其在社区检测中的应用。《物理学：复杂性》，5(3)，2024年8月。ISSN 2632-072X。doi: 10.1088/2632-072x/ad64a3。1 Rémi Flamary、Nicolas Courty、Alexandre Gramfort、Mokhtar Z. Alaya、Aurélie Boisbunon、Stanislas Chambon、Laetitia Chapel、Adrien Corenflos、Kilian Fatras、Nemo Fournier、Léo Gautheron、Nathalie T.H. Gayraud、Hicham Janati、Alain Rakotomamonjy、Ievgen Redko、Antoine Rolet、Antony Schutz、Vivien Seguy、Danica J. Sutherland、Romain Tavenard、Alexander Tong 和 Titouan Vayer。POT：Python最优传输。《机器学习研究杂志》，22(78):1–8，2021年。URL http://jmlr.org/papers/v22/20-451.html。1，2，6 Robin Forman。胞腔复形与组合Ricci曲率的Bochner方法。《离散与计算几何》，29(3):323–374，2003年2月。ISSN 1432-0444。doi: 10.1007/s00454-002-0743-x。1 Johannes Gehrke、Paul Ginsparg 和 Jon Kleinberg。2003年KDD杯概述。《SIGKDD探索通讯》，5(2):149–151，2003年12月。ISSN 1931-0145。doi: 10.1145/980972.980992。19 Pablo M. Gleiser 和 Leon Danon。Jazz中的社区结构。《复杂系统进展》，6 573，2003 4 565 10 1142 0219525903001067 19 ( ): – 。doi: . /S 。Xiang Gu、Huichun Zhang 和 Jian Sun。Ollivier–Ricci曲率的基于残差壳的下界。《应用数学快报》，183:110095，2026年。ISSN 0893-9659。doi: 10.1016/j.aml.2026.110095。1 Aric A. Hagberg、Daniel A. Schult 和 Pieter J. Swart。使用NetworkX探索网络结构、动态与功能。《科学中的Python会议》，2008年。doi: 10.25080/TCWV9851。18 Moritz Hehl。具有正Ollivier–Ricci曲率的正则图。《数学年刊》，394(1):24，2026年2月。ISSN 1432-1807。doi: 10.1007/s00208-026-03327-6。2，3

Dorit S. Hochbaum. 一种用于图像分割、马尔可夫随机场及相关问题的高效算法。《J. ACM》，48(4):686–701，2001年7月。国际标准书号 0004-5411。doi: 10.1145/502090.502093. 1, 3 John E. Hopcroft and Richard M. Karp. 一种用于二分图最大匹配的 $n^{5/2}$ 算法。《SIAM计算杂志》，2(4):225–231，1973年。doi: 10.1137/0202019. 5 Q. Huangfu and J. A. J. Hall. 并行化对偶修正单纯形法。《数学规划计算》，10(1):119–142，2018年3月。国际标准书号 1867-2957。doi: 10.1007/s12532-017-0130-5. 6 Hiroshi Ishikawa. 带凸先验的马尔可夫随机场精确优化。《IEEE模式分析与机器智能汇刊》，25(10):1333–1336，2003年。doi: 10.1109/TPAMI.2003.1233908. 1, 3 Jürgen Jost and Shiping Liu. 图上的 Ollivier Ricci曲率、局部聚类与曲率-维数不等式。《离散计算几何》，51(2):300–322，2014年3月。国际标准书号 0179-5376。doi: 10.1007/ s00454-013-9558-1. 1, 6 Jaewoo Jung, Young Soo Kwon, Seungjae Lee, and Jihye Park. 一种研究图上 α-Ricci曲率和 Lin-Lu-Yau Ricci曲率的组合方法，2026年。URL https://arxiv.org/abs/2606.24261. 2, 3 Wonwoo Kang and Heehyun Park. 高阶网络中 Ollivier–Ricci曲率的无传输下界：范围、紧致性与计算应用，2026 https://arxiv.org/abs/2405. 。13302 1

URL . Ming-Yang Kao, Tak-Wah Lam, Wing-Kin Sung, and Hing-Fung Ting. 最大权重二分图匹配的分解定理。《SIAM计算杂志》，31(1):18–26，2001年。doi: 10.1137/ S0097539799361208. 5 Christy Kelly. 图上 Ollivier 曲率的精确表达式与约化线性规划，2020年。URL https://arxiv.org/abs/1909.12156. 2 H. W. Kuhn. 用于指派问题的匈牙利算法。《海军研究后勤季刊》，2(1-2): 83 97, 1955 10 1002 3800020109 https://onlinelibrary.wiley.com/doi/abs/10. – 1002/nav.3800020109 2 。doi: 。/nav. 。URL 。 Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba：基于 LLVM 的 Python 即时编译器。载于《第二届 LLVM 编译器基础设施在 HPC 中的应用研讨会论文集》，LLVM '15，美国纽约州纽约市，2015年。Association for Computing Machinery。国际标准书号 9781450340052。doi: 10.1145/2833157.2833162. 20 Jure Leskovec and Andrej Krevl. SNAP数据集：斯坦福大型网络数据集集合。http://snap. stanford.edu/data 2014 21 ，。

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 随时间演化的图：致密化规律、直径收缩及可能的解释。载于《第十一届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集》，KDD '05，第177–187页，美国纽约州纽约市，2005年。Association for Computing Machinery。国际标准书号 159593135X。doi: 10.1145/1081870.1081893. 21 Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 图演化：致密化与直径收缩。《ACM知识发现与数据汇刊》，1(1):2–es，2007年3月。国际标准书号 1556-4681。doi: 10.1145/1217299.1217301. 21 Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. 大型网络中的社区结构：自然簇大小与缺乏大型明确定义的簇。《互联网数学》，6(1): 29–123，2009年。doi: 10.1080/15427951.2009.10129177. 21 Yong Lin, Linyuan Lu, and Shing-Tung Yau. 图的 Ricci曲率。《东北数学杂志》，63(4): 605–627，2011年。doi: 10.2748/tmj/1325886283. 1, 2 Giorgio Micaletto and Tebe Nigrelli. 可扩展图曲率：通过局部组合结构界定 Ollivier–Ricci曲率，应用于几何深度学习。载于 NeurIPS 2026 Workshop: Bridging Optimal Transport, Learning and Structured Data: Toward Geometric Distributional Learning，2026 https://neurips.cc/virtual/ 。URL . 2026/loc/paris/159361 2603 13535 5 1 非正式出版。扩展版本：arXiv: 。v 。R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. 网络模体：复杂网络的简单构建模块。《科学》，298(5594):824–827，2002年。doi: 10.1126/science.298.5594.824. 19 Florentin Münch and Radosław K. Wojciechowski. 一般图拉普拉斯算子的 Ollivier Ricci曲率：热方程、拉普拉斯比较、非爆炸性与直径界。《数学进展》，356 106759, 2019 0001 8708 10 1016 2019 106759 2 : 。国际标准书号 - 。doi: 。/j.aim. 。 。Khang Nguyen, Nong Minh Hieu, Vinh Duc Nguyen, Nhat Ho, Stanley Osher, and Tan Minh Nguyen. 使用 Ollivier-Ricci曲率重新审视过平滑与过压缩。载于 Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett 编，《第40届国际机器学习会议论文集》，《机器学习研究论文集》第202卷，第25956–25979页。PMLR，2023年7月。1, 3

Chien-Chun Ni, Yu-Yao Lin, Feng Luo, and Jie Gao. 基于 Ricci流的网络社区检测。《Scientific Reports》，9(1):9984，2019年7月。国际标准书号 2045-2322。doi: 10.1038/s41598-019-46380-9. 1, 6, 9 Yann Ollivier. 度量空间上马尔可夫链的 Ricci曲率。《Journal of Functional Analysis》，256(3): 810 864, 2009 0022 1236 10 1016 2008 11 001 https://www.sciencedirect. – 。国 com/science/article/pii/S002212360800493X 1, 3
际标准书号 - 。doi: 。/j.jfa. 。 。URL 。Yun Jin Park and Didong Li. 用于高效社区检测的下Ricci曲率。《Transactions on Machine Learning Research》，2025年。国际标准书号 2835-8856。URL https://openreview.net/forum?id=EoiuRII7MQ. 1 Mathew Penrose. 《随机几何图》。牛津大学出版社，2003年5月。国际标准书号 9780198506263。doi: 10.1093/acprof:oso/9780198506263.001.0001. 19 Laurent Perron and Vincent Furnon. OR-Tools。Google，https://developers.google.com/ optimization/ 2024 9 2, 6 ，。版
本 。Gabriel Peyré and Marco Cuturi. 用于数据科学的计算最优传输及其应用。《机器学习基础与趋势》，11:355–607，2019年2月。doi: 10.1561/2200000073. 2, 15 Jean-Claude Picard. 图的极大闭包及其在组合问题中的应用。《管理科学》，
22 11 1268 1272, 1976 00251909, 15265501 http://www.jstor.org/stable/ ( ): – 。 2630227 1, 3, 13, 14
国际标准书号 。URL 。Matthew Richardson, Rakesh Agrawal, and Pedro Domingos. 语义网的信任管理。载于《语义网 - ISWC 2003》，第351–368页，柏林，海德堡，2003年。Springer Berlin Heidelberg。国际标准书号 978-3-540-39718-2. 21 Benedek Rozemberczki and Rik Sarkar. 图上的特征函数：物以类聚，从统计描述符到参数模型。载于《第29届 ACM 国际信息与知识管理会议论文集》，CIKM '20，第1325–1334页，美国纽约州纽约市，2020年。Association for Computing Machinery。国际标准书号 9781450368599。doi: 10.1145/3340531.3411866. 21 Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 多尺度属性节点嵌入。《Journal of Complex Networks》，9(2):1–22，2021年4月。国际标准书号 2051-1329。doi: 10.1093/comnet/cnab014. 21 Areejit Samal, R. P. Sreejith, Jiao Gu, Shiping Liu, Emil Saucan, and Jürgen Jost. 复杂网络中两种 Ricci曲率离散化方法的比较分析。《Scientific Reports》，8(1):8650，2018年6月。国际标准书号 2045-2322。doi: 10.1038/s41598-018-27001-3. 1 Cedric Sanders, Andreas Roth, and Thomas Liebig. 图神经网络中的基于曲率的池化。载于《Machine Learning and Principles and Practice of Knowledge Discovery in Databases》，第471–485页，Cham，2025年。Springer Nature Switzerland。国际标准书号 978-3-031-74633-8. 1 Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Step han Günnemann. 图神经网络评估中的陷阱，2019年。URL https://arxiv.org/abs/1811.05868. 21 Jayson Sia, Edmond Jonckheere, and Paul Bogdan. 基于 Ollivier-Ricci曲率的复杂网络社区检测方法。《Scientific Reports》，9(1):9800，2019年7月。国际标准书号 2045-2322。doi: 10.1038/s41598-019-46079-x. 1 Joshua Southern, Jeremy Wayland, Michael Bronstein, and Bastian Rieck. 用于图生成模型评估的曲率过滤。载于《第37届神经信息处理系统国际会议论文集》，NIPS '23，美国纽约州雷德胡克，2023年。Curran Associates Inc. 1 R.P. Sreejith, Karthikeyan Mohanraj, Jürgen Jost, E mil Saucan, and Areejit Samal. 复杂网络的 Forman曲率。《Journal of Statistical Mechanics: Theory and Experiment》，2016(6):063206，2016年6月。doi: 10.1088/1742-5468/2016/06/063206. 1 Christian L. Staudt, Aleksejs Sazonovs, and Henning Meyerhenke. NetworKit：大规模复杂网络分析工具套件。《Network Science》，4(4):508–530，2016年。doi: 10.1017/nws.2016.20. 18 Yu Tian, Zachary Lubberts, and Melanie Weber. 图上的基于曲率的聚类。《机器学习研究杂志》，26(52):1–67，2025年1月。国际标准书号 1532-4435。1, 6 Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. 通过曲率理解图上的过压缩与瓶颈。载于《国际学习表征会议》，2022年。URL https://openreview.net/forum?id=7UmjRGzp-A. 1, 6 Floriano Tori, Vincent Holst, and Vincent Ginis. 重新审视基于曲率的重连效果及超参数在图神经网络中的作用。载于《第十三届国际学习表征会议》，2025年。URL https://openreview.net/forum?id=EcrdmRT99M. 9 Jan van den Brand, Li Chen, Rasmus Kyng, Yang P. Liu, Richard Peng, Maximilian Probst Gutenberg, Sushant Sachdeva, and Aaron Sidford. 一种用于最小费用流的确定性近线性时间算法。载于《2023年 IEEE 第64届计算机科学基础年度研讨会（FOCS）》，第503–514页，美国加利福尼亚州洛斯阿拉米托斯，2023年11月。IEEE计算机学会。doi: 10.1109/FOCS57990.2023.00037. 4

Cédric Villani。 最优传输专题，第58卷。美国数学学会，2003年1月。国际标准书号 9780821833124。doi：10.1090/gsm/058。 2Duncan J. Watts 和 Steven H. Strogatz。‘小世界’网络的集体动力学。 《自然》，393(6684)：440–442，1998年6月。ISSN 1476-4687。doi：10.1038/30918。 19Jaewon Yang 和 Jure Leskovec。基于真实情况定义和评估网络社区。载于*ACM SIGKDD*数据语义挖掘研讨会论文集，MDS ’12，美国纽约州纽约市，2012年。Association for Computing Machinery。国际标准书号 9781450315463。doi：10.1145/2350190.2350193。 21Wayne W. Zachary。小群体中冲突与分裂的信息流模型。 《人类学研究杂志》，33(4)：452–473，1977年。ISSN 00917710。URL http://www.jstor.org/stable/3629752。19Zexian Zhou 和 Bo Jiao。环重叠模式下的 Ollivier-Ricci 曲率， 2026 https://arxiv.org/ 。URL 。 abs/2606.03317 1

A 证明

全文中， μ 是有限度量空间 $(X,\ \text{dist})$ 上的符号测度，其中 $\mu(X)=0, P=\{\mu>0\}, N=\{\mu<0\}$, $\nu=-\mu|_N$, $A=\mu(P)=\nu(N)>0$(若 $A=0$，则每个陈述都显然成立)，且 $\text{dist}(x,\ y)\in\{1,\dots,D\}$ 于 $P\times N$。

A.1 定理1

证明。对偶性。 $\mathcal{W}_1(\mu^+,\mu^-)$ 是传输问题 $\min\{\sum_{x,y}\text{dist}(x,y)\pi_{xy}:\pi\geq 0,\ \sum_y\pi_{xy}=\mu(x)$ $\sum_x\pi_{xy}=\nu(y)\}$ 的值，该问题可行且有界；其对偶问题为 (1)，且两者的值相同。(1) 的约束矩阵是完全二部图在 $P\sqcup N$ 上的关联矩阵的转置，因此是全幺模的；又由于右端 $\text{dist}(x,y)$ 均为整数，故 (1) 存在整数最优解 (u,v) [Ahuja 等人， 1993]。

范围。设 $v_0=\min_{y\in N}v_y$ 且 $\bar{u}_x=\min_{y\in N}\{v_y+\text{dist}(x,y)\}$。则根据 $\bar{u}_x\geq u_x$ 可行性，$(\bar{u},v)$ 按构造是可行的，并且其取值不会更小，因为 $\mu(x)>0$；此外 $v_0+1\leq\bar{u}_x\leq v_0+D$，因为 $\text{dist}(x,y)\geq 1$ 且 $v_y\geq v_0$，并且在 v的一个最优解处，$\text{dist}(x,y)\leq D$。设 $\bar{v}_y=\min\{v_y,\ v_0+D-1\}$。如果 $\bar{v}_y=v_y$，则 y 的约束不变；否则$\bar{u}_x-\bar{v}_y\leq(v_0+D)-(v_0+D-1)=1\leq\text{dist}(x,y)$。由于 $\bar{v}_y\leq v_y$ 且 $\nu(y)>0$，取值不会降低。最后，从所有势函数中减去 v_0 会保持可行性和取值不变，因为 $\mu(P)-\nu(N)=0$。因此，(1) 存在一个满足 $u\in\{1,\dots,D\}^P$且 $v\in\{0,\dots,D-1\}^N$的最优解，我们可以将 (1) 限制在此类向量上。

一元编码。对于这样的（u，v），令 $\alpha_{x,\ \tau}=[u_x\geq\tau]$ 表示 $\tau\in\{2,\dots,D\}$，令 $\beta_{y,\ \sigma}=[v_y\geq\sigma]$ 表示 $\sigma\in\{1,\dots,D-1\}$。这是到满足在 τ 和 σ 中均为非递增、且 $u_x=1+\sum_\tau\alpha_{x,\ \tau}$，$\tau$ 以及 $v_y=\sum_\sigma\beta_{y,\ \sigma}$，$\sigma$ 的 0/1 向量的一个双射。对于（x，y），$\in P\times N$ 满足 $d=$距离（x，y），我们声称：$u_x-v_y\leq d$ 当且仅当对于所有 $\tau\in\{d+1,\dots,D\}$，$\alpha_{x,\ \tau}\leq\beta_{y,\ \tau-d}$。如果 $u_x-v_y\leq d$ 且 $\alpha_{x,\ \tau}=1$，则 $v_y\geq u_x-d\geq\tau-d$，因此 $\beta_{y,\ \tau-d}=1$。反之，如果 $u_x\leq d$，则该约束成立，因为 $v_y\geq 0$；否则 $\tau=u_x\in\{d+1,\dots,D\}$ 具有 $\alpha_{x,\ \tau}=1$，所以 $\beta_{y,\ u_x-d}=1$，即 $v_y\geq u_x-d$。

闭包。为节点$a_{x,\tau}$ 赋予权重 $\mu(x)>0$，为节点 $b_{y,\sigma}$ 赋予权重 $-\nu(y)<0$。根据上一步，受限范围内的可行对 (u,v) 对应于节点集合 Z（即指示变量为 1 的节点）；这些节点在定义 1 中除与 s 和 t 相连的弧之外的弧下是闭合的：链编码了单调性，蕴含弧则编码了约束。式 (1) 的目标函数等于 $\sum_x\mu(x)(1+\sum_\tau\alpha_{x,\tau})-\sum_y\nu(y)\sum_\sigma\beta_{y,\sigma}=A+w(Z)$。根据 Picard，闭集的最大权重等于正权重之和减去以下网络中最小割的容量：弧 s $\to z$ 的容量为 $w(z)$（适用于 $w(z)>0$），弧至 t 的容量为 $-w(z)$（适用于 $w(z)<0$），闭包弧的容量为无限容量。该网络即 $\mathsf{G}_D(\mu)$。正权重之和为 $(D-1)A$，因此根据最大流最小割定理， $\mathcal{W}_1=A+(D-1)A-F=DA-F$。 □

定理 1 适用于成本属于 $\{1,\dots,D\}$ 的任何问题，我们在定理 H 中使用这一点8。

A.2 定理 2、定理 3和定理4

定理的证明 2。 如果 $y\in\mathcal{N}[i]$，那么 x，$y\in\mathcal{N}[i]$ 并且 $\text{dist}(x,y)\leq 2$ 经过 i。如果 $x\in\mathcal{N}[j]$，那么$\text{dist}(x,y)\leq 2$ 经过 j。否则， $x\in\mathcal{N}[i]\setminus\mathcal{N}[j]=\mathcal{U}_i$ 并且 $y\in\mathcal{N}[j]\setminus\mathcal{N}[i]=\mathcal{U}_j$。路径

x， i， j， y给出 dist$(x,y) \le 3$；此外，dist$(x,y) = 1$ 当且仅当 $xy \in E$，而 dist$(x,y) \le 2$ 当且仅当 $xy \in E$ 或 x 和 y 具有一个公共邻居。 □

定理 3 的证明。若 $\mu(x) > 0$，则 $m_i(x) > 0$，因此 $P \subseteq \mathcal{N}[i]$；同理， $N \subseteq \mathcal{N}[j]$，且 $P \cap N = \emptyset$。根据定理 2，dist$(x,y) \in \{1,2,3\}$ 在 $P \times N$ 上成立，因此定理 1 可取 $D = 3$ 应用；其蕴含弧为：对于 dist$(x,y) = 1$，有 $a_{x,2} \to b_{y,}$1 和 $a_{x,3} \to b_{y,}$2；对于 dist$(x,y) = 2$，有 $a_{x,3} \to b_{y,}$1。接下来只需证明可以去掉链弧 $b_{y,2} \to b_{y,}$1。在 $G_3(\mu)$ 中闭合的每个集合，在去掉这些弧后仍然闭合。反过来，设 Z 是去掉这些弧后的网络中的一个最大权闭集，并假设 $b_{y,2} \in Z$。若没有满足 $xy \in E$ 的 $a_{x,}$3 属于 Z，则没有弧从 Z 进入 $b_{y,}$2，而 $Z \setminus \{b_{y,2}\}$ 将是一个权重更大的闭集（$b_{y,}$2 的权重为负），这就产生矛盾。因此，某个与 y 相邻的 x 满足 $a_{x,3} \in Z$，进而 $a_{x,2} \in Z$ 且 $b_{y,1} \in Z$。由此，两个网络的最大权闭集具有相同的权重，其最小割也相同。若不存在距离为 3 的点对，则 dist$(x,y) \in \{1,2\}$ 在 $P \times N$ 上成立，定理 1 取 $D = 2$ 可得 $\mathcal{W}_1 = 2A - F_1$；其中 G_2 的每个顶点对应一个节点，并包含满足 dist$(x,y) = 1$ 的弧 $a_{x,2} \to b_{y,}$1；若不存在相邻对，则 $F_1 = 0$。

□

定理 4 的证明。根据定理 1 和定理 3 的证明， $\mathcal{W}_1 = A + w(Z)$ 对 G_3 中的最大权重集合 Z 闭合。令 $X_3 = \{x : a_{x,3} \in Z\}$ 和 $X_2 = \{x : a_{x,2} \in Z\}$；这些链给出$X_3 \subseteq X_2$。对于 $y \in N_1(X_2)$（弧 $a_{x,2} \to b_{y,}$1）以及 $y \in N_2(X_3)$（距离 2 处的弧 $a_{x,3} \to b_{y,}$1，以及距离 1 处的 $a_{x,3} \to a_{x,2} \to b_{y,}$1），闭性要求 $b_{y,1} \in Z$；对于 $y \in N_1(X_3)$，则要求$b_{y,2} \in Z$；这些节点集合是闭的，并且由于 b 节点的权重为负，最大权重的 Z 不包含其他节点。其权重就是括号中的值。 □

球与多步游走。若 supp $m_i \subseteq B_r(i)$ 和 supp $m_j \subseteq B_r(j)$，则 $P \subseteq B_r(i)$、 $N \subseteq B_r(j)$，且 dist$(x,y) \le r + 1 + r$ for $(x,y) \in P \times N$，因此定理 1 适用于 $D = 2r + 1$。

A.3 定理5

证明。设 f 是一个最大流。残量网络中从 s 可达的节点集合 Z 是最小割的源点侧；它不包含任何无限容量弧的尾节点（若其头节点位于 Z 之外，该割的容量将为无限），因此 $Z \setminus \{s\}$ 是闭合的，并且根据定理 1 的证明和 Picard [1976], ，它具有最大权重。由于这些链是 G_D 的弧，集合 $\{\tau : a_{x,\tau} \in Z\}$ 和 $\{\sigma : b_{y,\sigma} \in Z\}$ 是初始段，而解码 $u_x = 1 + |\{\tau : a_{x,\tau} \in Z\}|$, $v_y = |\{\sigma : b_{y,\sigma} \in Z\}|$ 是一元编码的逆过程；因此 (u,v) 对于 (1) 是可行且最优的。根据互补松弛，每个最优传输计划都支撑在紧集 $T = \{(x,y) : u_x - v_y = \text{dist}(x,y)\}$ 上；并且最优方案存在，所以限制在 T 上的传输问题是可行的，且该二分网络中的最大流会饱和从 s 出发的弧，即其流值为 A；其弧流构成一个方案 π，具有正确的边缘分布且支撑在 T 内，其成本为 $\sum \pi_{xy}(u_x - v_y) = \sum \mu(x)u_x - \sum \nu(y)v_y$，即最优对偶值。反过来，任何方案和任何取值相等的可行势函数都能证明这个数是 $\mathcal{W}_1$ （弱对偶性）。

□

A.4 定理6 与双节点合并

定理 6 的证明。若一个网络由 G 的节点以及源点和汇点~弧组成（$_3$, 还可以包含权重为 0 的额外节点和无限容量弧），并且其闭集限制到 G 的节点上~ ~后恰好是 G 的闭集，则称该网络为可容许网络。可容许网络具有~G 的最大闭包权重$_3$, ，因而具有相同的最大流值（闭集对应于割，~如定理 1 的证明所示；额外节点的权重为 0）。G_3 是可容许的；我们证明，在可容许网络 G 中添加一个中继节点后得到的网络 G′仍然是可容许的。设 Z' 在 G′中是闭的。那么 $Z' \setminus \{h\}$ 在 G 中是闭的：G 中的每条弧都是 G′ 中的弧，除了被删除的弧 $a_{x,3} \to b_{y,}$1、$(x,y) \in P' \times N'$；对于这些弧，$a_{x,3} \in Z'$蕴含 $h \in Z'$，进而蕴含$b_{y,1} \in Z'$。反过来，设 Z_0~在 G_3 中是闭的，并且 Z 是 G 的一个闭集，其限制为 Z_0。如果 Z 不包含任何满足 $x \in P'$ 的 $a_{x,}$3，则它在 G′中是闭的。否则 $Z \cup \{h\}$ 在 G′中是闭的，因为对于每个 $y \in N'$，$b_{y,1} \in Z_0$：若 dist$(x,y) = 2$，则 G_3 具有弧 $a_{x,3} \to b_{y,}$1；若 dist$(x,y) = 1$，则具有路径 $a_{x,3} \to a_{x,2} \to b_{y,}$1。两个方向~都保持到 G_3, 的节点的限制，因此 G′ 是可容许的。 □

第 5 节中的四个中继满足定理 2的假设： H_1 将所有行（位于 $\mathcal{N}[i]$）连接到 $\{i,j\} \cup \mathcal{C} \subseteq \mathcal{N}[i]$中的列；$H_2$ 将 $\{i,j\} \cup \mathcal{C} \subseteq \mathcal{N}[j]$ 中的行连接到所有列（位于 $\mathcal{N}[j]$）； H_3

将满行连接到 $\mathcal{U}_j$；而 H_4 则将 $\mathcal{U}_i$ 中剩余的行连接到满列。再加上剩余距离为 2 的点对所对应的直接弧或见证中继，它们实现了距离为 2 的点对的每个蕴含关系$a_{x,\ 3} \Rightarrow b_{y,\ 1}$，且不实现其他蕴含关系。

命题12 (孪生节点)。 设 $x, x' \in P$ 满足：对于每个 τ，节点 $a_{x,\tau}$ 和 $a_{x',\tau}$ 除了各自链上的后继节点 $a_{x,\tau-1}$ 和 $a_{x',\ \tau-1}$ 之外，具有相同的后继节点。对于每个 τ，将这两个节点替换为一个源容量为 $\mu(x)+\mu(x')$ 的节点，不会改变最大流值。

证明。 设 Z 是一个最大权闭集。如果 $a_{x,\tau} \in Z$，那么对所有 $\sigma \le \tau$ 添加 $a_{x',\sigma}$ 都能保持闭合（它们的后继节点与 $a_{x,\sigma}$ 的后继节点相同，而后者根据链的关系均属于 Z），且不会降低权重，因为这些节点的权重为正；由对称性可知，存在一个最大权闭集，对于每个 τ，要么同时包含 $a_{x,\tau}$ 和 $a_{x',\tau}$，要么两者都不包含。这样的集合恰好是合并网络的闭集，且权重相同。 □

在求解器中，一行的后继节点由它与 N 的邻接关系、其覆盖 Cov（这决定它是否为完整版本以及它在距离 2 处的弧），以及它是否位于 $\{i, j\} \cup \mathcal{C}$ 中（这决定其中继节点）确定，因此键相同的行满足该假设；在见证网络中，则由其相邻的列及其见证确定。

A.5 定理 7 与图最小费用流

定理 7 的证明。(i) 令 Ω $= \{w_1, \ldots, w_k\}$ 和 $P_l = P \cap \mathcal{N}(w_l)$，$N_l = N \cap \mathcal{N}(w_l)$。对于 (x, y) $\in P_l \times N_l$，我们有 $x \neq y$（因为 $P \cap N = \emptyset$），且路径为 x, w_l, y，因此 dist$(x, y) \le 2$。从定理 3 的 G_3 出发，按照定理 6 的方法逐个添加中继节点 $h_{w_1}, \ldots, h_{w_k}$，并删除 $P_l \times N_l$ 中的弧 $a_{x,3} \to b_{y,}$ 1；根据该命题，最大流值不变。一对距离 2 为 $\in P \times N$ 的顶点 (x, y) 不相邻，且有一个公共邻居 w，它与 $x \in P$ 和 $y \in N$ 相邻，因此是一个见证点；对于 $w = w_l$，有 $(x, y) \in P_l \times N_l$；因此，$\tilde{\mathsf{G}}_3$ 中的每条弧 $a_{x,3} \to b_{y,}$1 都会被删除，结果为 G^{w}。定理 3 给出 $\mathcal{W}_1 = 3A - F^{\mathrm{w}}$。

(ii) 令 $\Sigma = \sum_{x \in P \cup N} d_x \le \ell_{ij}$。 $P \cup N \subseteq S$的每个顶点的度至少为 1，因此 $|P| + |N| \le \Sigma$。中继弧与对 $(x,$ $w)$、$x \in P \cup N$、$w \in$Ω $\cap \mathcal{N}(x)$ 一一对应，即与 $P \cup N$ 的某些邻接项对应：至多有 Σ 个。每个见证点至少有一条中继弧进入且至少有一条中继弧离开，因此 $|\Omega| \le \Sigma/2$。相邻对 $(x, y) \in P \times N$ 使用 x 列表中的项 y 以及 y $2e_1 \le \Sigma$ ^{w} $2\ |P| + |N|\ + |\ \ | + 2 \le$ 列 $\frac{5}{2}\Sigma + 2 \le 3\ell_{ij} + 2$ $2|P| + 2|N| + |P| + 2e_1 +$ $\le 3\Sigma + \Sigma + \Sigma \le 5\ell_{ij}$ 表中的项 x，且不同的相邻对使用不同的项，因此 。所以 G 有 ( ) Ω 个节点和 (中继弧) 条弧。

(iii) 在 V 上保留整数数组，所有元素仅初始化一次为 -1，并为每条边递增计数器 c；通过将 c 写入数组来标记顶点，因此无需清除，之前边的标记会自动失效。标记 $\mathcal{N}(i)$ 并扫描 $\mathcal{N}(j)$，将两个列表划分为 $\mathcal{U}_i$、$\mathcal{U}_j$、$\mathcal{C}$（$O(d_i + d_j)$，无需排序），并计算质量、P 和 N。在第一个数组中标记 P 的顶点，在第二个数组中标记 N 的顶点，并在两个伴随数组中记录其局部下标。扫描 P 顶点的列表：对于条目 w，如果 w 在第二个数组中已被标记，则记录该相邻对，并在第三个数组中标记 w（“从 P 看到”）。扫描 N 顶点的列表：在第三个数组中已标记的条目 w 是见证点；如果它尚未在第四个数组中标记，则在那里标记它，在伴随数组中为其赋予下一个下标，并为其分配节点 h_w；追加弧 $h_w \to b_{y,\ 1}$。再次扫描 P 的列表，并为第四个数组中已标记的每个条目 w 追加 $a_{x,\ 3} \to h_w$。每个列表至多读取两次，并通过一次计数遍将数量如 (ii) 中所界定的弧转换为压缩邻接形式。 □

引理13 （图最小费用流）。 对于单步测度，$\mathcal{W}_1(m_i, m_j) = \min\{\sum_{(x,\ y)} f_{xy} :\ \ f \ge 0,$ $\sum_y f_{xy} - \sum_y f_{yx} = \mu(x)\ \ \forall x\}$，其中 f 遍历该边 ij的两种定向以及端点位于 $P \cup N$的各条边上的流，这类边最多有 $\ell_{ij} + 1$条。

证明。在整个图上，右端（包含所有边）对于跳数度量为 $\mathcal{W}_1$：流可分解为路径和环，而从 P到 N 的路径，其代价至少为距离；沿最短路径传输的最优方案则给出代价为 $\mathcal{W}_1$ [的流（Beckmann, 1952年，Peyré 和 Cuturi, 2019]）。限制使用的边只会使最小值增大。最优方案将顶点 $x \in P$与顶点 $y \in N$配对，每一对都可以通过定理 2 中某种形式的最短路径连接：边 xy；x, w, y（对应一个公共邻居 w，可能为 i 或 j）；或 x, i, j, y。除最后一种形式中的 ij 外，这些路径上的每条边都有一个端点属于 $P \cup N$，因此沿这些路径传输该方案，可得到一条

在受限边上的可行流，其代价为 $\mathcal{W}_1$。其数量之所以有界，是因为除 ij 外，每条这样的边都是 $P \cup N$的邻接项。 □

A.6 定理8

证明。DasGupta 等人 [2023, 命题 2] 中的图 G_H 具有边 (i, j)，i 与$L = \{x_1, \ldots, x_n\}$相邻，j与 $R = \{y_1, \ldots, y_n\}$相邻，每一对权重为 1 的顶点之间都有一条边 $x_a y_b$，并且有一个新顶点 z_{ab}，它与 x_a 和 y_b 中的每一对权重为 2 的顶点相邻。于是 $d_i = d_j = n + 1, \mathcal{C} = \emptyset$；对于惰性随机游走 μ 在 $\{i, j\}$ 上消失，因此根据定理 2，$P = L, N = R$，每个质量为 $1/(n + 2)$ 的顶点，以及 dist(x_a, y_b) $= w(x_a, y_b)$。见证点恰好是顶点 z_{ab}（i 和 j只与一侧相邻，并且 L中的任意两个顶点，或 R中的任意两个顶点，都不相邻）。（i）按照˜定理 7中的方法计数：G_3 中的 $4n+2$ 个节点加上 e_2 个中继；$5n$ 条源弧、汇弧和链弧，$2e_1$ 条邻接弧，以及每个中继对应的两条弧。局部输入为 $\ell_{ij} = 2(n + 1)$ $+ 2n + 2e_1 + 2e_2$。由于 $\mathcal{W}_1 = M(H)/(n + 2)$[DasGupta 等，2023，第 2 节.2]，$M(H)$ 个带单位质量的 w $= 3n - F$。（ii）在其定理 4 的证明中，$e_2 = 0$，且权重为 1 的顶点对构成一个匹配（可能为空）。于是，除了链 $a_{x,3} \to a_{x,2}$ 之外，从 $a_{x,2}$ 和 $a_{x,3}$ 发出的弧只通向与 x匹配的列 y，网络由匹配顶点对的路径 s $\to a_{x,\tau} \to b_{y,\tau-1} \to$ t ($\tau = 2, 3$)、链，以及彼此之间没有弧的行和列组成。如果不存在匹配顶点对，则 t 不可达，Dinic 算法在第一次搜索后停止。否则，在其第一个阶段中，链连接同一层级的节点，因而不会被使用；阻塞流沿长度为 3 的路径使匹配行的每条源弧都饱和，之后唯一未饱和的源弧通向未匹配行，而从这些行无法到达任何列。该阶段耗时与网络规模成线性关系，因此结合（i），精确曲率的计算需要 $O(n)$ 时间。 □

A.7 定理9

证明。在 G_B、$d_i = d_j = n + 1$ 和 $\mathcal{C} = \emptyset$ 中，因此对于惰性随机游走，μ 在 $\{i, j\}$、$P = L$、$N = R$ 上消失，并且 $L \cup R$ 的每个顶点都具有质量 $1/(n + 2)$。L 中的一个顶点只与 i 以及 R 中的顶点相邻，而 R 中的一个顶点只与 j 以及 L 中的顶点相邻，因此 $L \times R$ 的任意一对都没有公共邻居；根据定理 2，dist(x, y) 在 $xy \in E_B$ 时为 1，否则为 3。在质量相等的情况下，运输问题是一个指派问题，并且存在一个最优解，它是 $L \times R$ [的完美匹配的 $1/(n + 2)$ 倍（Ahuja 等人，1993]）。E_B 中具有 k 对的完美匹配的代价为 $k + 3(n - k) = 3n - 2k$，而 B 中的每个匹配都可以扩展为 $L \times R$ 的完美匹配，因此最小值为 $3n - 2\alpha'(B)$。局部输入为 $d_i + d_j + \sum_{x \in L \cup R} d_x = 2(n + 1) + 2n + 2|E_B|$，并且 G_B 可在 $O(n + |E_B|)$ 时间内构建。 □

A.8 定理10

证明。将所有容量乘以相同的因子不会改变 Dinic 算法的各个阶段（层次图和选择都相同，且每个流都按相同比例缩放），因此我们可以采用最小的整数缩放：流均为整数，且每个阶段都会使流至少增加 1。将节点 h_w 称为中继节点，H_1、...、H_4 称为中继节点。中继节点只与 a节点和 b节点相连，而 s 和 t 只与 a 节点和 b节点相连，因此任意两个中继节点都不相邻，在残量网络中也如此。对于任意可行流，进入节点 v 的流量至多为 $T(v)$，其中 $T(a_x,3) = \mu(x)$（它唯一的入弧来自 s；对于合并后的行类别，为该类别的 μ），$T(a_x,2) = 2\mu(x)$（它的入弧来自 s 和 a_x,3），$T(b_y,1) = \nu(y)$，以及 $T(b_y,2) = 2\nu(y)$（它们的出弧通向 t；若保留该链，则 b_y,2 的出弧还通向 b_y,1）；这些流量之和至多为 $6A$。

˜设 f 为某阶段开始时的流，f^* 为一个最大流，M 为二者流值之差。˜则 $g = f^* - f$ 是残量网络 f 中的一个流，其流值为 M，且进入每个节点 v 的流量至多为 $2T(v)$：进入 v 的弧最多承载 f^* 单位的正向流，而离开 v的弧最多承载 f单位的反向流。˜将 g 分解为单位流量的 s–t 路径和环。每条路径的长度至少为当前从 s 到 t 的距离 l，因此其中至少有 $(l - 1)/2$ 个中间节点不是中继节点；而每个不是中继节点的节点 v 至多位于 $2T(v)$条路径上。因此 $M(l - 1)/2 \le 12A$，也就是 $l \le 24A/M + 1$。每个阶段的距离都会增加 [Even 和 Tarjan，1975]；而 $M \ge 1$、$l \le \sqrt{24A} + 1$，因此此类阶段至多有 $\sqrt{24A} + 1$ 个；此后至多还有 $\sqrt{24A}$个阶段，因为每个阶段都会使流值至少增加 1。

一个阶段在广度优先搜索、当前弧指针和死路方面需要 $O(m)$ 时间，此外还要加上每条增广路径的长度。每次增广至少发送一个单位，因此当还剩下 r 个单位时

它沿着一条长度至多为 $24A/r+1$ 的路径进行；流至多为 $2A$，即从 s 流出的容量，因此增广路径的总长度为 $O(A\log A)$，所需时间为 $O(\sqrt{A}\,m + A\log A)$。对于具有 $d_i = d_j$ 的惰性随机游走，μ 在 $\{i,j\}\cup\mathcal{C}$ 上消失，并且 $\mathcal{U}_i$ 的每个顶点和 $\mathcal{U}_j$ 具有相同的质量，因此 $A = |\mathcal{U}_i|$；结合 $m \le 5\ell_{ij}$（定理 7）和 $|\mathcal{U}_i|\log|\mathcal{U}_i| \le \ell_{ij}\sqrt{|\mathcal{U}_i|}$，所需时间为 $O(\ell_{ij}\sqrt{|\mathcal{U}_i|})$。 □

A.9 定理11

证明。设 (i,j) 是 G' 中一条边，且有 $i,j\notin U$。没有变更对包含 i 或 j（每个 $a_l,b_l\in U$），因此 $(i,j)\in E$，$\mathcal{N}(i)$ 和 $\mathcal{N}(j)$ 在 G 和 G' 中相同，S, m_i 和 m_j 也相同。此外，S 中没有 a_l 或 b_l：如果 $a_l\in\mathcal{N}[i]$ 在 G' 中，那么 $i\in\mathcal{N}[a_l]\subseteq U$；如果 $a_l\in\mathcal{N}[i]$ 在 G 中但不在 G' 中，那么 (i,a_l) 是一个变更对，其中包含 i；对于 j 和 b_l 也是如此。根据定理 2 和定理 1，$\kappa(i,j)$ 由 m_i, m_j 决定；对于各对 $x,y\in S$，还取决于 $xy\in E$ 是否成立，以及 x 和 y 是否有公共邻居。变更对 (a_l,b_l) 仅在 $\{x,y\}=\{a_l,b_l\}$ 时改变第一个关系，并且仅对包含 a_l 或 b_l 的各对改变第二个关系（路径 x,w,y 使用了各对 $\{x,w\}$ 和 $\{w,y\}$），而 a_l 和 b_l 均不在 S 中。因此，$\kappa(i,j)$ 在 G 和 G' 中相同。 □

B 实现

算法 1 单条边的精确曲率（i，j），$d_i \ge d_j$（一步测度）

将排序列表 $\mathcal{N}(i)$ 和 $\mathcal{N}(j)$ 合并为 $\mathcal{U}_i, \mathcal{U}_j, \mathcal{C}$；行 $\mathcal{N}[i]$，列 $\mathcal{N}[j]$
整数质量 $\Lambda m_i, \Lambda m_j$（每块，或加权游走时每个顶点）；行上的净质量 μ，ν 列上
如果 $A = 0$**则**
  返回 $\kappa = 1$
结束 if
$\theta \leftarrow 4\big(\sum_{x\in\mathcal{N}[i]} d_x + \sum_{y\in\mathcal{N}[j]} d_y\big)$ 位集运算的界（以字数计）
对于每个与列相邻的顶点 w：位集 Col(w)，其中包含其列邻居；如果这需要超过 θ 个字，**转到**最后一行
对于每一行 x **执行**
  调整(x) $\leftarrow$ 列(x)；覆盖(x) $\leftarrow \bigcup_{w\in\mathcal{N}(x)}$ 列(w)，当覆盖(x) $\supseteq \mathcal{U}_j$时停止；如果超过 θ 单词已合并，且目前大多数行都不是完整版本，**转到**最后一行
结束循环
如果 $\mathcal{U}_i$ 的每一行都是完整版本**则**
  合并调整相同的行 $\cap N$；$F_1 \leftarrow$ 单层网络的最大流；
  返回 $\kappa = 1 - (2A - F_1)/\Lambda$
结束 if
合并键相同的行（Adj $\cap N$，Cov，块）；构建 $\widetilde{G}_3$，其中包含中继节点 H_1、……、H_4，并且对于其余距离为 2 的点对，若其弧的上界小于显式弧数量的四分之一，则使用见证中继，否则使用显式弧
如果网络至少有 300 条弧**则**
  $F_1 \leftarrow$ 单层网络的最大流，在两层上加载 {warm start}
else
  $F_1 \leftarrow 0$
结束 if
$F \leftarrow 2F_1$ +在残量网络上执行 Dinic 增广；
返回 $\kappa = 1 - (3A - F)/\Lambda$
(弃用位集) 构建 G^w 定理7 并进行孪生合并；热启动和 Dinic 同上；
返回 $\kappa = 1 - (3A - F^w)/\Lambda$

质量。对于 $\varrho = \max\{d_i, d_j\}$，块 $\{i\},\{j\},\mathcal{C},\mathcal{U}_i,\mathcal{U}_j$ 上的整数质量，分别对 m_i 和 m_j 如下：惰性，$\Lambda = (d_i+1)(d_j+1)$，$m_i \equiv d_j+1$ 在 $\mathcal{N}[i]$ 上，$m_j \equiv d_i+1$ 在 $\mathcal{N}[j]$ 上；非惰性，$\Lambda = d_i d_j$，$m_i \equiv d_j$ 在 $\mathcal{N}(i)$ 上，$m_j \equiv d_i$ 在 $\mathcal{N}(j)$ 上；闲置率 a/b，$\Lambda = b\,d_i d_j$，$m_i \equiv$ (i) $= a\,d_i d_j$ 以及 $(b-a)d_j$ 在 $\mathcal{N}(i)$ 上（对 j 对称）；LLY，闲置率 $1/(\varrho+1)$ 和 $\kappa_{\mathrm{LLY}} = (\varrho+1)\kappa/\varrho$。对于边权重为整数 w 且闲置率为有理数 a_u/b_u 的加权游走，$m_u(u) = a_u/b_u$ 和 $m_u(x) = (1-a_u/b_u)\,w_{ux}/\sum_z w_{uz}$；逐顶点分母只约分一次，且 Λ 是 i 和 j 对应分母的最小公倍数。

中继节点详解。 行依次为 $\mathcal{U}_i$ 、 i、 j和 $\mathcal{C}$；列也同样如此。 H_1 从每个行类的$a_{x,3}$ 接收弧，并向 $b_{y,1}$ 发送弧，这些列属于 N 在 $\{i,j\}\cup\mathcal{C}$中的部分； H_2 从 P在 $\{i,j\}\cup\mathcal{C}$ 中的行类接收弧，并向 N的所有列发送弧； H_3 从满行类接收弧，并向 $N\cap\mathcal{U}_j$的列发送弧； H_4 从 $\mathcal{U}_i$ 的非满行类接收弧，并向满列发送弧。只有当中继节点两侧都非空时，才会创建该中继节点。位集网络中的见证中继仅用于剩余的配对：中间顶点 w 会从与其相邻的 $\mathcal{U}_i$ 非满行类获得一个中继，连接到与其相邻的 $N\cap\mathcal{U}_j$ 非满列；只有当两侧的元素都多于一个时才会创建（否则，其配对会获得直接弧）。

见证网络。 G^{w} 中具有相同邻接列和相同见证点的行将被合并；它们的键是排序后的编码 $2v+[v$ 是一个见证点$]$，对应于其相关邻居，这些邻居按顺序从排序后的邻接表中产生。若某个见证点只连接到一个行类或一列，则将其替换为直接弧。质量和方向采用位集内核中的设置，因此精确分数完全相同（已针对六种测度和每个标志进行测试）。

执行。 边以循环方式分成 $8T$ 个分块，由 T 个线程处理（Numba `prange`）；每个线程拥有可增长的整数存储区，因此不会为每条边分配内存；除 CSR图之外的工作空间为每个线程中最大的本地网络所需的空间，另加见证网络的数组，这些数组在线程首次接收一条边时分配。对于至少包含 300 条弧的网络，将使用热启动。

接口与测试。 该软件包提供 `curvature(graph, measure)`，可按需返回精确分数和逐边诊断信息，并允许关闭第 5 节中的每个步骤以进行消融； `certify(graph, edge, measure)`（定理 5），可返回一个最优方案和一个最优势函数，对二者进行精确验证，并可通过广度优先距离从头重新检查；`ball_curvature` 以及 `walk_curvature` （定理 1 ，使用 $D=2r+1$）；还有 `affected_edges`（定理 11）。测试会在随机插入和删除后，将 `affected_edges` 与完整重新计算的结果进行比较，并将所有核彼此比较，同时与线性规划以及表 *3*和图 3中的预言机进行比较。该独立实现会构建完整的局部距离矩阵、合并孪生节点，并使用自有的距离和最大流代码求解三级闭包。

C 实验设置

硬件与软件。 Slurm 集群中的节点配备以下处理器之一：两颗 Intel Xeon Gold 6148（每颗 20 核，2.4 GHz）、两颗 Xeon Gold 6248（20 核，2.5 GHz）、两颗 Xeon Gold 6230（20 核，2.1 GHz）、两颗 AMD EPYC 7713（64 核，2.0 GHz），或四颗 Xeon Gold 6148；每个作业均使用 32 个线程。受时间限制而停止的步骤会在这些节点中任一空闲节点上继续执行，因此表格中的各行以及最大行的分块可能来自不同的处理器（Orkut 的数据来自两台节点）。求解器和每项实验均使用 Python 3.12.16、NumPy 2.2.5、Numba 0.62.1、 `POT`0.9.7.post1、 `OR-Tools` 9.15.9999 和 NetworkX 3。3 [Hagberg 等人， 2008]; GraphRicciCurvature 0.6.1 在独立环境中使用 SciPy 1.13 和 NetworKit [Staudt 等人， 2016]运行。

不同节点类型之间的校准。 为了衡量处理器差异对时间和加速比的影响，我们在 3 种类型的各一个节点上重新测量了 Epinions 和 DBLP（表 5）。在不同节点类型之间，我们的时间最多相差 $1.3\times$， `POT`的估计值最多相差 $1.1\times$，而它们的比值最多相差 $1.3\times$。这种差异主要来自 AMD EPYC 节点，在该节点上我们的求解器运行更快，并产生更高的加速比；两种 Intel Xeon 处理器之间差异很小，并且每个图的仅求解比值在不同节点类型之间变化很小（表 5）。E3 与 `POT` 的比较不受影响：除 Orkut 外，每一行数据都在配备两颗 Xeon Gold 6248 的节点上通过一次运行测得；Orkut 的分块则在该节点以及配备四颗 Xeon Gold 6148 的节点上运行，后者也给出了其 `POT` 估计值。这两种 Intel 处理器见表 5，并产生更为保守的加速比。

每一行对应的节点。日志记录每一行写入的时间，调度器记录每个作业在哪个节点上运行，因此每一行都可以追溯到其处理器（表6）。在追溯的1,150行中，有12行（全部来自E3的随机图）是在工作节点同时运行于两种节点类型上时写入的。除上述E3分块行外，每一行自身的比较均来自同一次运行。同一图的各行之间进行比较时，仅E5混用了处理器：其中有4个图用于与GRC比较，另有2个图用于与Sinkhorn比较；在E4和E6中，每个图的各行均来自同一个处理器。

表 5： 在每种类型的一个节点上对相同图进行计时，取三次运行的中位数，使用 32 个线程；每个度层中对 `POT` 个样本的 20 条边进行估计。比值： `POT` 的估计时间与我们的时间之比，包括其距离矩阵计算时间，以及仅进行求解的时间之比。

处理器	图	ours	`POT`（估计值）	比值	仅求解
EPYC 7713	Epinions	22.9 秒	89.5 秒	3.9×	1.4×
Xeon Gold 6148	Epinions	28.8 秒	90.4 秒	3.1×	1.3×
Xeon Gold 6248	Epinions	29.1 秒	85.6 秒	2.9×	1.2×
EPYC 7713	DBLP	549 毫秒	4.49 秒	8.2×	2.1×
Xeon Gold 6148	DBLP	685 毫秒	4.89 秒	7.1×	2.2×
Xeon Gold 6248	DBLP	688 毫秒	4.50 秒	6.5×	2.0×

表6： 根据日志和调度器记录追踪到的每个实验所使用的处理器：每个处理器写入的行数。未确定（ND）：在该步骤的工作节点运行于两种类型的节点上时写入。

实验	处理器（行）
E1	Xeon Gold 6248（95）
E2，受控边	四个 Xeon Gold 6148（168）、Xeon Gold 6248（75）
E2，实图	Xeon Gold 6248（175）
E2，下界	四个 Xeon Gold 6148（19）、Xeon Gold 6248（9）
E3，真实网络	Xeon Gold 6248（57）、四个 Xeon Gold 6148（1）
E3，线程	EPYC 7713（24）
E3，随机图	EPYC 7713 (56)，四个 Xeon Gold 6148 (20)、Xeon Gold 6230 (1)、ND (12)
E4	EPYC 7713 (150)
E5，完整图	EPYC 7713 (63)、Xeon Gold 6148 (18)
E5，Sinkhorn	Xeon Gold 6148 (90)、EPYC 7713 (15)
E6，测度	EPYC 7713 (70)
E6，球与游走	Xeon Gold 6148 (32)

图。每个真实网络都限制为其最大连通分量，并转换为简单无向图：表7中的20个网络，以及随代码附带的5个小型网络：Karate [Zachary、1977], Jazz [Gleiser 和 Danon、 2003], Yeast [Milo 等、 2002], Power Grid [Watts 和 Strogatz、1998], 以及 arXiv HepPh [Gehrke 等， 2003]。随机族均使用固定种子，包括 Erd˝os 和 Rényi[1959, ER] $G(n, m)$、Barabási 和 Albert [1999, BA], 、单位正方形中的随机几何图（RGG，Penrose，2003年）、具有0.2重连概率的 Watts 和 Strogatz [1998, WS] ，以及具有幂律指数2.1和2.5的 Chung 和 Lu [2002]图（CL2.1、CL2.5）；随机图写作 family(n，平均度)，例如ER(2k, 10)表示 $n = 2$，000个节点且平均度为10。

E1 使用五个小型网络 ER(2k, 10)、BA(2k, 10)、WS(2k, 10)、RGG(2k, 12)、CL2.1(3k, 8)、CL2.5(3k, 8)，以及八个典型图： K_{20}、 Q_6、 20×20 网格、深度为 8 的二叉树、C_{100}、 K_1、200、具有 50 个轮缘顶点的轮图，以及 K_{10}、30。E2 使用这些小型网络（Karate 除外）和表7中边数少于 10^6 的 14 个网络。E3 使用表7中的全部 20 个网络和每个随机族：对于 $n = 10^3$、$3 \cdot 10^3$、.....、107（CL2.1 达到 $3 \cdot 10^6$），平均度为 16；对于 n 至 10^6（ER 和 BA 达到 $3 \cdot 10^5$，CL2.1 和 CL2.5 达到 10^5），平均度为 $2.1\, n^{0.4}$，每种情况使用一个种子。E4 使用 ER(20k, 20)、BA(20k, 10)、RGG(20k, 20)、CL2.1(50k, 10)、arXiv HepPh、GitHub、Twitch-DE、Squirrel、Epinions 和 DBLP；E5 使用图4中的 15 个图；E6 则针对其他一步测度使用 arXiv HepPh、Deezer、Twitch-DE、GitHub、Amazon-Computers、Epinions、DBLP、YouTube、RGG(100k, 20) 和 CL2.1(30k, 12)，针对球与多步游走使用 Power Grid、Yeast、Jazz、ER(5k, 8)、BA(5k, 6)、WS(5k, 8)、RGG(5k, 10) 和 CL2.5(5k, 6)。

受控边。 E2 的一条受控边（i， j）有 $|\mathcal{U}_i| = a$、 $|\mathcal{U}_j| = b$ 和 t 个公共邻居。对称扫描对于 $s = 2^4$， 2_5、.....、2_{14}的最大值为 $a = b = s$和 $t =$，而枢纽扫描的最大值为 $a = 32$、 $t = 4$和 $b = s \in \{2^6, 2^8, \ldots, 2^{16}\}$；每种规模都有三个种子。距离为 1 的顶点对是交叉边，即 $\mathcal{U}_i$ 与 $\mathcal{U}_j$之间的边；距离为 2 的顶点对来自连接器，即与 $\mathcal{U}_i$ 和 $\mathcal{U}_j$中的随机顶点相邻的额外顶点。八个连接器各自与 $\mathcal{U}_i$ 和 $\mathcal{U}_j$中的一部分顶点相邻，这一比例为 f，从而约有 $1 - (1 - f^2)^8$ 的顶点对拥有公共邻居，并选择 f 使得

该比例为 q。交叉边是从 $\mathcal{U}_i \times \mathcal{U}_j$ 中均匀且有放回地抽取的 Bin(ab, p) 个顶点对，因此一对顶点成为交叉边的概率约为 $1 - e^{-p}$。五种状态分别为：树，没有交叉边和连接器；稀疏，具有 $p = 2/b$ 和 $q = 0.05$；混合，具有 $p = 2/b$和 $q = 1/2$ （图2）；稠密，具有 $p = 0.2$（约占顶点对的 18%）以及一个与 $\mathcal{U}_i \cup \mathcal{U}_j$中的所有顶点相邻的连接器，因此不存在距离为 3 的顶点对（仅适用于 $s \le 2^{12}$，再往上交叉边过多，无法生成该距离）；以及 hubleaf，其中 $\mathcal{U}_j$ 中除 2% 以外的所有顶点都是 j的叶节点，因而是孪生节点，而剩余的 2% 承载着混合状态的结构。

下界实例。 表 1 中的各个族都包含一条边 (i, j)，其中 i 与 L相邻， j 与 R相邻，并且 $|L| = |R| = n$分别为：单个， $L \times R$ 的一个随机相邻对（DasGupta 等人的定理 4(a)的实例，2023年）；匹配，一个随机完美匹配（其定理 4(b)）；有界， L的每个顶点拥有两个距离 1 的随机伙伴和两个距离 2 的随机伙伴，每一对距离 2 的顶点都有各自的中间顶点（具有有界加权度的图 G_H）；以及块，round($\sqrt{n}$)个共享中间顶点，其中每个顶点都与 L 和 R中的 round($\sqrt{n}$)个随机顶点相邻； $n = 2^8, 2_{10}, \ldots, 2_{20}$，使用两个种子。

计时。每条边的时间是在编译后对一个批次进行多次重复运行所得的最快时间；整图时间是三次运行的中位数（超过 10^7 条边）。单线程比较为每种方法固定使用一个线程。 `POT`的网络单纯形法通过其 C++ 入口调用，迭代次数上限设置得足够高，因而始终不会触发； `ot.emd2` 按照 GRC 调用它的方式进行调用。整图 `POT` 时间通过每个分层最多取 50 条边进行估算， $\lfloor \log_2(d_i + d_j) \rfloor$，优先处理最大分层；每个图的预算为 30 分钟，其中包括构造距离矩阵的时间，并除以线程数；如果某条边的稠密局部问题会超过 $5 \cdot 10^7$ 个条目，则将其计为免费，这使估计值保持为下界。图最小费用流（定理13）使用 `OR-Tools' SimpleMinCostFlow`，在该边 ij 的两个定向以及端点属于 $P \cup N$的边上使用整数质量；这些边由 Numba [根据 Lam 等人的 2015]代码构建，其构建时间计入总时间。

GRC.. 版本 0 6 1, `OllivierRicci` `compute_..` `ricci_curvature_edges` 32 2×10^4 按照用户通常调用的方式，通过 及其方法 ，使用进程，计算至 顶点的全点对最短路径以及超出该范围的成对最短路径（GRC 在内存峰值时会从 n^2 浮点数、 $\approx 48n^2$ 字节构建矩阵），且不设时间限制； `ATD` 始终构建此矩阵，并且不会在矩阵无法放入作业内存时运行。方法：默认方法（`OTDSinkhornMix`, `nbr_topk` $= 3000$）、 `OTD`、 `OTD` （不截断）、`Sinkhorn` （正则化 0.1）和 `ATD`，全部为 $\alpha = 1/2$.[3]

D 附加结果

其他测度（E6）。非惰性、LLY、闲置率 $1/4$、 $1/2$和 $3/4$，加权惰性游走的代价在 $5.6\times$ 之内，与惰性随机游走相当（中位数 $0.99\times$；表11）。对于半径为2和3 的球上的均匀测度，以及两步和三步游走，我们对G$_{2r+1}$的参考实现（仍会构建 P和 N之间的距离）在中位数上快 $11\times$ 于 `POT` ，适用于半径为2 的球，并能验证其返回的每个值（表12）； $r \ge 2$ 的无距离构造是未来工作。

[3] `OTD` 是 `ot.emd2`，精确到 `POT`的默认上限 100,000 次网络单纯形迭代：表 4 中的 0.037 是由于 GitHub 的规模（6.7×10^7 个条目）所致。ATD 是 GRC 的平均运输距离，而 OTDSinkhornMix 使用 OTD，除非两个端点的度都高于 2,000，否则使用 Sinkhorn。

表7： E3 的真实网络（最大连通分量）。除 Amazon-Photo、Amazon- Computers 和 Coauthor-Physics 外，均来自 Leskovec 和 Krevl [2014]；最后一列给出了每个网络的原始来源。

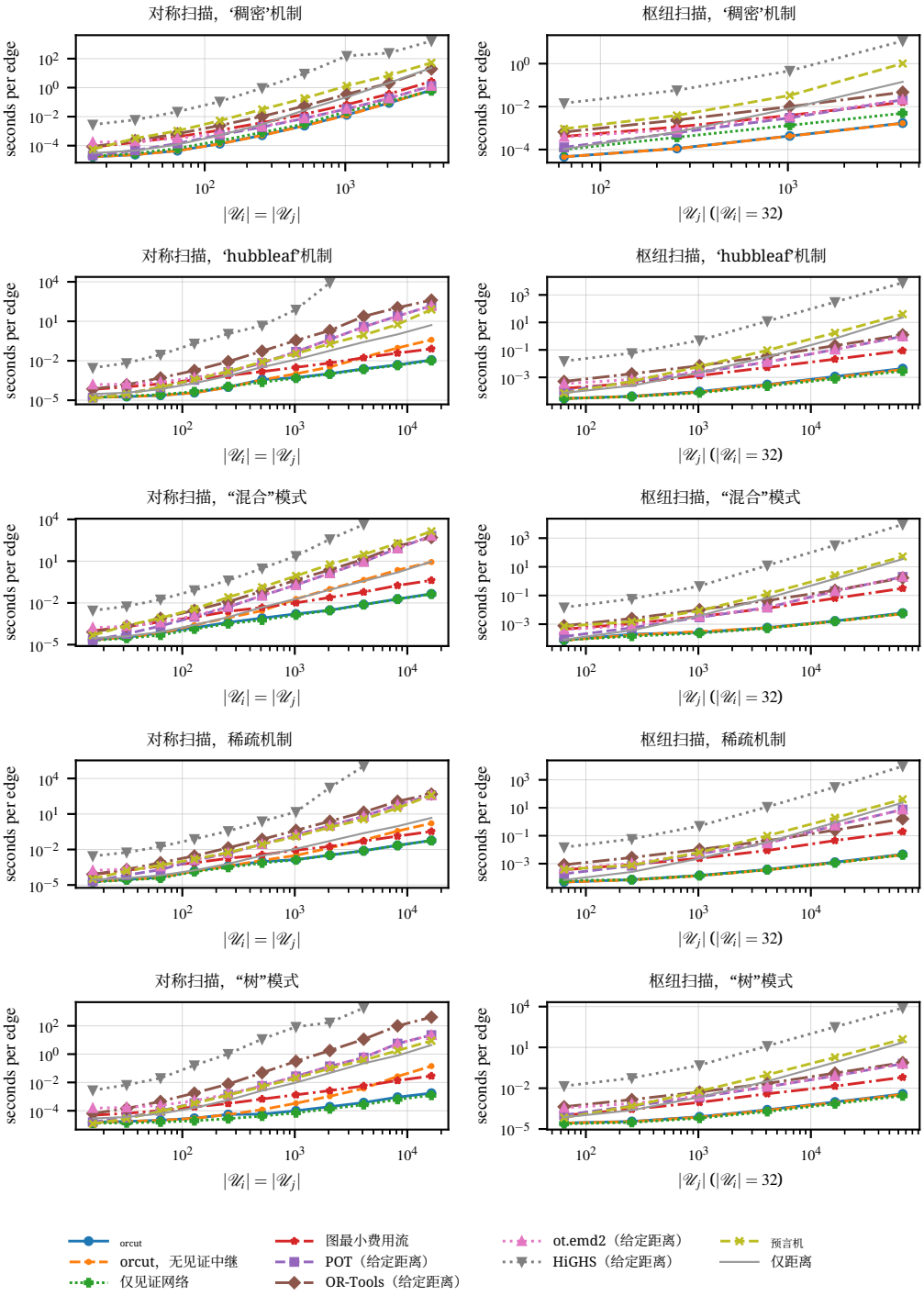
图	n	m	$d_{\max}$	平均度	源点
Deezer	28,281	92,752	172	6.56	Rozemberczki 和 Sarkar [2020]
Amazon-Photo	7,487	119,043	1,434	31.8	Shchur 等人 [2019]
Twitch-DE	9,498	153,138	4,259	32.2	Rozemberczki 等人 [2021]
Crocodile	11,631	170,773	3,546	29.4	Rozemberczki 等人 [2021]
Facebook	22,470	170,823	709	15.2	Rozemberczki 等人 [2021]
Enron	33,696	180,811	1,383	10.7	Leskovec 等 [2009]
AstroPh	17,903	196,972	504	22.0	Leskovec 等 [2007]
Squirrel	5,201	198,353	1,903	76.3	Rozemberczki 等人 [2021]
Brightkite	56,739	212,945	1,134	7.51	Cho 等人 [2011]
Amazon-Computers	13,381	245,778	2,992	36.7	Shchur 等人 [2019]
Coauthor-Physics	34,493	247,962	382	14.4	Shchur 等人 [2019]
GitHub	37,700	289,003	9,458	15.3	Rozemberczki 等人 [2021]
Epinions	75,877	405,739	3,044	10.7	Richardson 等人 [2003]
亚马逊	334,863	925,872	549	5.53	Yang 和 Leskovec [2012]
DBLP	317,080	1,049,866	343	6.62	Yang 和 Leskovec [2012]
roadNet-CA	1,957,027	2,760,388	12	2.82	Leskovec 等 [2009]
YouTube	1,134,890	2,987,624	28,754	5.27	Yang 和 Leskovec [2012]
Skitter	1,694,616	11,094,209	35,455	13.1	Leskovec 等 [2005]
LiveJournal	3,997,962	34,681,189	14,815	17.3	Yang 和 Leskovec [2012]
Orkut	3,072,441	117,185,083	33,313	76.3	Yang 和 Leskovec [2012]

表 8： `OR-Tools`：边比较结果（分数/不匹配）。核：与预言机及分层闭包（惰性、非惰性和 LLY 曲率）之间的不匹配。 `POT`：最大绝对差值。证书：已检查/失败。

度量	图	边	`OR-Tools`	核	`POT`	证书
lazy	19	496,631	95,847 / 0	0	2.8×10^{-14}	12,430 / 0
非惰性	19	496,631	95,847 / 0	0	3.6×10^{-14}	12,430 / 0
Lin–Lu–Yau	19	496,631	95,847 / 0	0	2.9×10^{-14}	12,430 / 0
闲置率 $1/2$	19	496,631	95,847 / 0	–	2.2×10^{-14}	12,430 / 0
加权惰性游走	19	496,631	7,160 / 0	–	–	7,160 / 0

表 9： 最大对称规模下的网络规模和运行时间。输入： ℓ_{ij}^{S}；弧：显式距离二对、求解器和 G^w 的计数；指数：弧相对于 ℓ_{ij}^{S}的拟合标度指数；时间 / 求解器：相对于不使用见证中继的求解器的运行时间（同时禁用位集上限和 G^w 回退机制），分别针对单独使用 G^w 以及图最小费用流。

			arcs			指数		时间 / 求解器		
机制	s	ℓ_{ij}^{S}	显式	求解器	见证点	显式	见证点	no 见证点缩写	见证点	图 flow
稠密	4,096	6,108,502	3,049,130	3,049,130	6,110,548	1.0	1.0	0.99	0.89	4.0
hubbleaf	16,384	177,610	2,455,787	173,413	173,413	1.6	1.0	34	0.89	7.0
混合	16,384	214,822,119	6,038,847	212,929	212,929	1.9	0.99	192	0.95	9.3
稀疏	16,384	160,262	11,927,732	160,189	160,189	1.7	1.0	28	0.97	5.8
tree	16,384	73,730	32,771	32,771	32,771	0.99	0.99	81	0.70	16



Figu图 3： 受控邻域上每条边耗时，涵盖全部五种状态和所有方法（如图 2所示）。预言机是第 B 节中基于显式距离矩阵的独立实现。

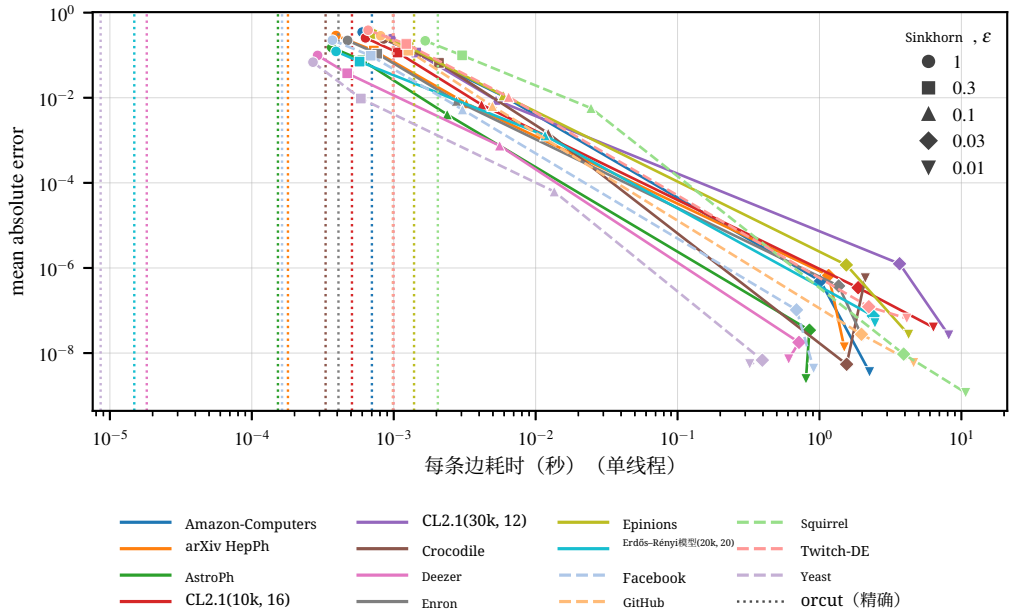


图4： 闲置率 $1/2$ 下的曲率平均绝对误差与每条边的平均时间之间的关系；对于每个图（颜色和线型）以及每个正则化 ε（标记），均使用单线程；Sinkhorn 使用给定的距离。点状竖线标出了精确求解器在每个图上的每条边平均时间，颜色对应于该图；其误差为零，因此在对数坐标轴上没有对应的点。

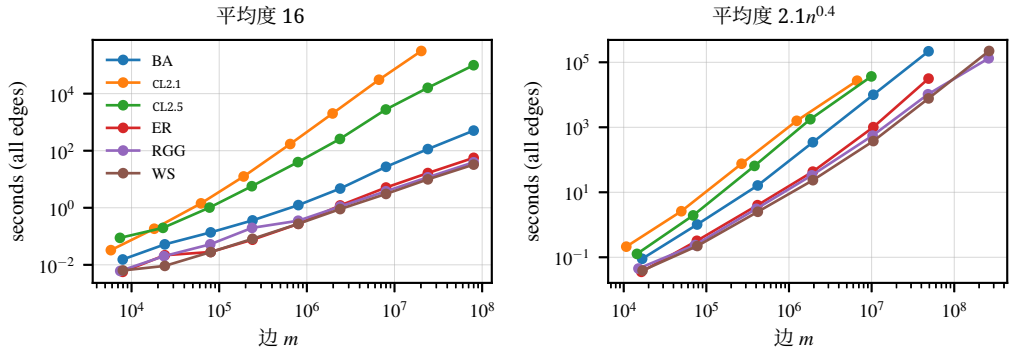


图 5： 随机图所有边的时间与边数之间的关系，平均度为 16（左）和 $2.1\,n^{0.4}$（右），使用 32 个线程；规模见第 C节。

表 10： 指数 k 时间 $\propto m^k$ 在随机族上的结果，通过最小二乘法对第 C 节规模范围内的对数进行拟合。

族	度	最大 m	k
BA	平均值 16	8.00×10^7	1.13
CL2.1	平均值 16	2.02×10^7	2.01
CL2.5	平均值 16	7.99×10^7	1.57
ER	平均值 16	8.00×10^7	1.01
RGG	平均值 16	7.99×10^7	0.93
WS	平均值 16	8.00×10^7	0.97
BA	$2.1n^{0.4}$	4.89×10^7	1.85
CL2.1	$2.1n^{0.4}$	6.68×10^6	1.86
CL2.5	$2.1n^{0.4}$	9.93×10^6	1.96
ER	$2.1n^{0.4}$	4.89×10^7	1.69
RGG	$2.1n^{0.4}$	2.61×10^8	1.57
WS	$2.1n^{0.4}$	2.64×10^8	1.61

表 11： 相对于惰性随机游走的全图时间（在各图上的中位数和最大值）。

度量	中位数	max
lazy	1.0	1.0
非惰性	1.0	1.3
Lin–Lu–Yau	1.0	1.0
闲置率 $1/4$	0.98	1.0
闲置率 $1/2$	0.98	1.0
闲置率 $3/4$	0.98	1.0
加权惰性游走	1.0	5.6

表 12： 更大的支撑集，分层网络$D = 2r + 1$（所有值均已在每个图上验证）。POT/ours：估计的单线程时间比（中位数、图上的范围）；最大差异：与POT的最大差异。

度量	POT/ours	范围	最大差异
B_2上的均匀分布	11	3.6–21	4.5×10^{-14}
在 B_3 上均匀分布	6.2	0.81–12	5.1×10^{-14}
2步懒惰随机游走	13	7.9–17	3.8×10^{-10}
3步懒惰随机游走	5.5	1.6–10	6.0×10^{-10}

E 生成式人工智能的使用

曾使用生成式人工智能助手（Claude，Anthropic）协助实现实验流程并对文本进行编辑。作者设计了方法和实验，核实了每一项陈述和证明，并对内容承担全部责任；所有数值均由已发布代码根据原始结果生成。