AGIS：基于结构感知采样的快速近似图模式挖掘

Seoyong Lee 首尔
国立大学 韩国首尔
sylee2685@snu.ac.kr

Jinho Lee 首尔国
立大学 韩国首尔
leejinho@snu.ac.kr

arXiv:2601.01388v1 [cs.DS] 4 Jan 2026

摘要

近似图模式挖掘（AGPM）对于分析大规模图至关重要，因为精确计数的计算开销过高。尽管已有许多基于采样的 AGPM 系统，但它们都依赖均匀采样，忽略了底层概率分布。这一局限限制了它们扩展到更广泛模式的能力。

本文介绍 AGIS，这是一个速度极快的 AGPM 系统，能够从超大规模图中对任意模式进行计数。AGIS 采用一种新颖的采样技术——结构感知邻居采样。该技术不同于均匀采样，而是根据模式结构分配特定的采样概率。我们首先推导出 AGPM 的理想采样分布，然后提出一种实用方法来近似该分布。此外，我们还开发了一种兼顾收敛速度与计算开销的方法，以确定何时使用近似分布。

实验结果表明，AGIS 的表现显著优于最先进的近似图模式挖掘系统，几何平均加速比达到 28.5×，在特定情况下加速超过 100,000×。此外，AGIS 是唯一能够扩展到包含数百亿条边的图、并稳健处理各种模式的近似图模式挖掘系统，能够在数秒内成功提供准确估计。我们将开源 AGIS，以推动该领域的进一步研究。

PVLDB 引用格式：
Seoyong Lee 和 Jinho Lee。AGIS：利用结构信息采样进行快速近似图模式挖掘。PVLDB，19(2)：238 - 251，2025。doi:10.14778/3773749.3773761

PVLDB 工件可用性：
源代码、数据和/或其他工件已发布于 https://github.com/syleeKR/AGIS。

1 引言

图模式挖掘 (GPM) [1, 24, 32, 43, 52, 70, 75, 80]是图处理工作负载中最耗时的应用之一。给定一个大型数据图和一个相对较小的模式图，图模式挖掘会在数据图中搜索该模式的嵌入。明确的模式计数能够简洁、确定地概括高阶图结构，作为支撑严谨统计分析 [50]的基础描述性统计量。

本作品采用知识共享署名-非商业性使用-禁止演绎 4.0 国际许可协议授权。访问 https://creativecommons.org/licenses/by-nc-nd/4.0/ 查看此许可协议的副本。对于超出本许可协议范围的任何使用，请发送邮件至 info@vldb.org 获取许可。版权归所有者/作者所有。出版权授权给 VLDB 基金会。VLDB基金会会刊，第 19 卷，第 2 期，ISSN 2150-8097。doi:10.14778/3773749.3773761

此外，图无处不在，模式挖掘被应用于各种领域，包括生物信息学 [28, 39, 53], 、社交网络分析 [40, 85], 和化合物分类 [31]。更重要的是，在欺诈检测和网络安全等高风险场景中，模式计数对于可解释性和问责性至关重要，因为可审计性和法规合规 [4, 7, 27, 29]取决于精准、可复现的证据。尽管图模式挖掘在众多现实场景中意义重大，但其固有的高计算复杂度使其难以扩展至大图或复杂模式。在当前时代，这一挑战尤为突出，因为现代应用需要对超大图进行低延迟分析 [7, 27, 65]。

一种广受欢迎的缓解方法是近似图模式挖掘（AGPM）。利用这样一个事实：频繁子图挖掘 [84] 和模体计数 [54],等图模式挖掘任务常用于只需估算嵌入计数的应用场景，AGPM 可在指定的误差界和置信区间内，计算给定模式的近似嵌入计数。尽管已有若干开创性的 AGPM 系统 [8, 42, 90], 但随着数据图和模式规模增大，它们仍面临困难。通过对这些系统进行分析，我们发现某些模式的性能下降尤为严重。我们的分析表明，这可归因于真实世界图中常见的无标度分布 [11, 12, 30] 。这种分布会导致嵌入计数的方差很大，具体取决于数据图中被采样的部分。由于这使得估算计数难以收敛，AGPM 系统的执行时间因此很长。

幸运的是，我们发现，通过为每个采样候选项分配不同的概率，可以缓解这一问题。现有方法也试图降低估计方差，但通常侧重于缩小候选范围，例如只考虑已采样顶点的邻居 [42], ，或满足特定约束的邻居 [8]。然而，一旦构建好候选集，每个候选项被采样的概率都是均匀的。

我们不采用朴素方法的均匀采样概率分布，而是提出使用非均匀分布，其中采样每个候选顶点的概率与选择该顶点所能找到的潜在嵌入数量成正比。理论上，这只需单个采样器即可得到无方差的正确计数。然而，这需要事先知道真实的嵌入计数，从而违背了AGPM的目的。为了在不依赖真实嵌入计数的情况下构建合适的采样概率分布，我们提出一种基于数据与模式图的结构特征来近似该分布的方法。通过仔细考察附近结构中的

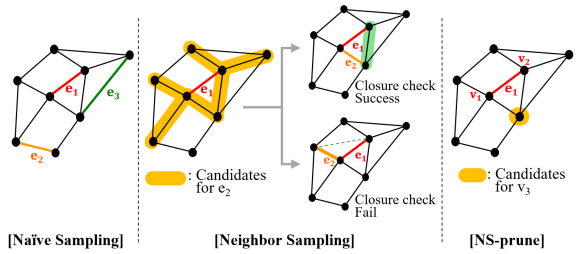


图 1：现有 AGPM 系统的采样策略。

待采样目标顶点，我们以较小的开销计算出接近理想情况的分布。

基于此，我们提出了 AGIS，这是一种快速且可扩展的 AGPM 系统。AGIS 的核心是结构感知邻居采样。结构感知邻居采样利用计算得到的分布，并策略性地确定何时应用该分布，从而在加快收敛速度与最大限度降低计算开销之间实现最佳平衡。此外，我们还提供了一种启发式方法来构建匹配顺序，以最大限度地发挥结构感知邻居采样的优势。我们在多种数据图和模式上对 AGIS 进行了评估，并将其与若干最先进的基线方法进行了比较。实验结果表明，AGIS 的表现优于所有基线方法，有时甚至高出数个数量级。AGIS 树立了新的最先进水平，进一步拓展了近似图模式挖掘方法的适用范围。

我们的贡献可总结如下：

- 我们表明，为每个采样候选项中的顶点分配不同的概率，可以显著加快 AGPM 的收敛速度。
- 我们提出了一种构造近似理想采样分布的方法，以及一种启发式决策流程，用于确定何时应用该方法。
- 我们构建了 AGIS，这是一个快速的 AGPM 系统，在一系列丰富多样的实验中显著优于现有方法。

2 背景

本节全面介绍近似图挖掘的背景及现有工作中的核心策略。我们分别用 $G = (V_G, E_G)$ 和 $P = (V_P, E_P)$ 表示数据图和模式图。我们假设这两个图都是无向图；因此，E_G 和 E_P 都是无序对的集合 $\{a, b\}$。对于顶点 v，$d\ v$ v ( ) 表示其度，N ( ) 表示其邻居顶点集合，而 v.id 表示其 ID。我们用 v 表示 G 中的顶点，用 u 表示 P 中的顶点。

2.1 近似图挖掘系统

图模式挖掘挖掘（GPM）旨在找出给定图 G 中模式 P 的所有嵌入，即匹配。形式上，我们将嵌入定义为从 P 到 G 的子图同构。具体而言，它是一个一一映射 M：

$V_P \to V_G$ $u_i\ u_j \in E_P$ u_i , $u_j \in$, E_G $C\ G\ P$

满足若 (,)，则 (M( ) M( ))。我们用 (,) 表示此类嵌入的总数。GPM 的示例任务包括子图计数 [32]、子图枚举（子图匹配）[32]、模体计数 [54],以及频繁子图挖掘 [84]。

近似图模式挖掘。近似图模式挖掘（AGPM）以近似方式处理子图计数问题。给定一个图 G、一个模式 P、一个误差界 ϵ和一个置信水平 $1-\delta$，AGPM 旨在以至少 $1-\delta$ 的概率返回嵌入的精确数量 $C(G, P)$ 的 $(1\pm\epsilon)$ 近似。AGPM 系统通过两个主要组件使用户能够挖掘任意模式：(1) 适用于任何模式的通用采样方法；(2) 收敛检测方法，用于根据 (ϵ, δ) 确定何时终止采样。

2.2 AGPM 的采样策略

为了获得

$C\ G\ P$ [8, (,) 的 42, 90]

近似值，现有方法采用采样方法。举个简单的例子，假设我们要估算图 G 中三角形的数量。图 1 展示了一种直接的朴素采样方法：采样器随机采样三条边 e_1, e_2, e_3，并检查它们是否构成一个三角形。如果构成三角形，采样器就将三角形数量估算为 $|E_G|^3$，因为三条边各自被随机采样到的概率均为 $\frac{1}{|E_G|}$。否则，采样器将三角形数量估算为 0。大量此类采样器的估算平均值将逐渐收敛至真实的三角形计数。

朴素采样方法的一个主要问题是采样器输出之间的方差很大。由于每个采样器的输出不是 0 就是数值大得多的 $|E_G|^3$，多个采样器的平均值收敛得很慢。邻居采样（NS）[59] 利用模式图的连通性解决了这一问题。具体而言，NS 采样器的操作如下：首先，它从 E_G 中均匀随机地采样一条边e_1，这与采用朴素方法的采样器类似。不过，它不会从 E_G 中均匀采样下一条边，而是从与 e_1 相邻的边集合中采样 e_2。然后，它执行一次 闭包检查，以确定 e_1 和 e_2 中未共享的顶点之间是否存在一条边 e_3。此过程可确认采样到的边能否构成一个三角形嵌入。其概率为 $\frac{1}{|E_G|}\cdot|N(e_1)|$，其中 $N(e_1)$ 表示 e_1 的邻接边集合。由于与 $|E_G|^3$ 相比，$|E_G|\cdot|N(e_1)|$ 更小且更接近真实的三角形计数，因此 NS 采样器输出的方差更低，收敛也更快。

上述方法可以轻松推广到任意模式 P。采样器依次采样顶点，直到已采样顶点的数量达到 $|V_P|$。然后，该采样序列的概率p 为

$$p = \prod_{i=1}^{|V_P|} Pr(v_i | v_1, \ldots, v_{i-1}),$$

其中，$Pr(v_i|v_1$、……、$v_{i-1})$ 表示在已采样 v_1、$\ldots v_{i-1}$ 的条件下采样 v_i 的概率。若这些顶点构成 P 的嵌入，则采样器返回 $\frac{1}{p}$，否则返回 0。该采样器是 $C(G$、$P)$ [42] 的无偏估计量。

NS 方法还可以通过对邻居选择施加限制来进一步改进 [19, 44]。例如，在采样 k 星模式（即 k 个顶点与一个中心顶点相连的模式）时，我们可以仅从中心顶点的邻居中采样下一条边，从而最大限度地提高闭包检查的成功率。基于这一思路，ScaleGPM [8] 提出了 NS 的一种改进版本，称为 NS-prune。该方法不仅利用模式的

连通性 P，还利用其特定的拓扑结构。它引入了精确 GPM 系统的两个概念：

匹配顺序。匹配顺序 [24, 37] π: $\{1, \ldots, |V_P|\} \to V_P$ 是模式顶点的一个排列，用于指定将 P 的顶点与图 G 进行匹配的顺序。我们将 $u_i = \pi(i) \in V_P$ 表示为匹配顺序中的第 i 个 P 顶点。相应地，我们将 $v_i \in V_G$ 写作第 G 个采样顶点中的第 i 个。

限制集。给定匹配顺序 π，限制集 [51] R^π 是由模式顶点有序对 (u_i, u_j) 构成的集合。R^π 与 对称性破缺 [51], 一起应用，以强制满足以下条件：对于所有 $(u_i, u_j) \in \mathrm{R}^\pi$，$\mathrm{id}\, v_i.\mathrm{id} < v_j.$。这对精确挖掘系统很有帮助，因为它能将搜索空间恰好缩小 P 的自同构数量。尽管这种方法不会枚举所有嵌入，但只需按相应倍数乘以结果，即可恢复总计数。

NS-prune 的应用方式如下。仍以三角形模式为例，它首先像以往的方法一样随机采样 $e_1 = \{v_1, v_2\}$。然后，它不再从邻居中任意采样 e_2，而是利用了以下事实：u_3 与 u_1 和 u_2相连。因此，下一个顶点 v_3 必须从 v_1 和 v_2的邻居集合的交集中选取，即 $v_3 \in \mathrm{N}(v_1) \cap \mathrm{N}(v_2)$。在每一步都施加这些连通性约束，可以提高找到嵌入的成功率，从而加快收敛。此外，三角形模式有 3! 个自同构，其限制集为 $\mathrm{R}^\pi = \{(u_2, u_1), (u_3, u_2), (u_3, u_1)\}$。通过应用对称性破缺，施加条件 $v_3.\mathrm{id} < \min(v_1.\mathrm{id}\, v_2.\mathrm{id})$。施加这些限制会缩小可能顶点的集合。由于采样概率等于集合大小的倒数，估计量方差得以降低。最后，对于一般模式，NS-prune 会避免从先前已采样的顶点中采样。

考虑上述技术，NS-prune 从候选集中均匀采样最后一个顶点 v_3

$$S = \{v \mid v \in \mathcal{N}(v_1) \cap \mathcal{N}(v_2),\ v.id < \min(v_1.id,\ v_2.id)\}.$$

然后返回 $|E_G| \cdot |S|$ 作为采样器的输出。在 ScaleGPM 中，由于使用了限制集，该值是 $C(G, P)$ 除以模式的自同构数量所得的无偏估计。由于采样器更频繁地产生非零输出，其方差得以降低，从而加快了收敛。

2.3 收敛方法

AGPM 需要解决的一个重要问题是确保 $C(G, P)$ 收敛到误差界 ϵ 以内。ASAP [42] 和 Arya [90]采用误差-延迟剖面（ELP）启发式方法，根据集中不等式预先确定收敛所需的采样器数量。例如，Arya 使用切比雪夫不等式 [9]推导出如下采样器数量：

$$\#\ \text{samplers needed} = \frac{K \times |E_G|^\rho}{C(G,P) \times \epsilon^2 \times \delta},$$

其中，ρ 表示特定模式值，K 是一个常数：然而，该方程需要 $C(G, P)$ 的值，而常数 K 也未知。为了解决这一问题，他们依靠从 G 的较小子图中进行采样，以估计 K和 $C(G, P)$。

遗憾的是，基于 ELP 的方法缺乏理论保证，因为它依赖于一个子图，而该子图的 $K$and $C(G,\ P)$ 与原始 K不同

算法1在线收敛检测

1: **过程** 收敛(ϵ, δ, **L**)
2: $N \leftarrow |\mathbf{L}|$ ▷ **L**是采样器输出的列表
3: $\mu \leftarrow -\ \frac{1}{N}\sum_{i=1}^{N} L_i$, $\sigma^2 \leftarrow \frac{1}{N}\sum_{i=1}^{N} L_i^2 - \mu^2$
4: $\hat{\epsilon} \leftarrow \Phi^{-1}(1 - \frac{\delta}{2}) \cdot \frac{\sigma}{\sqrt{N}} \cdot \frac{1}{\mu}$ ▷ 估计误差
5: **返回** $\hat{\epsilon} \le \epsilon$
6: **结束过程**

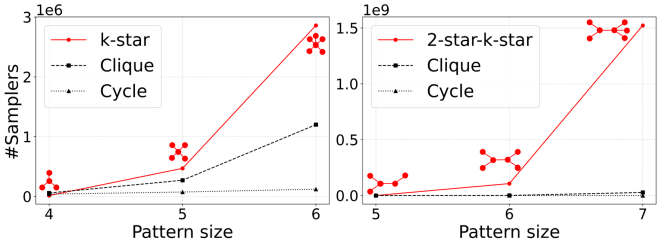


图2：收敛所需的采样器数量。LiveJournal 图，使用 $\epsilon = 0.1$, $\delta = 0.01$。

为了解决这一问题，ScaleGPM [8]提出了一种数学上严谨的方法来确定采样的终止时机，如算法 1 所示，其中 Φ 是标准正态分布的累积分布函数。该方法不预先设定采样器的数量，而是利用采样器结果的均值和方差来检测收敛。该方法已被证明适用于能够提供无偏估计的任意基于采样的技术。

3 动机

3.1 现有方法的局限性

尽管 AGPM 系统相比精确 GPM 系统的执行时间显著更短，但随着模式规模增大，其收敛速度较低的问题也广为人知。具体而言，我们发现，某些类型的模式会加剧这一问题。例如，含有许多桥的模式 [33],，其中桥指的是移除后会导致图分离的边。图2展示了使用 ScaleGPM 在 LiveJournal [10] 图上对两种模式——k-星和 2-星-k-星——实现收敛所需的采样器数量。与顶点数相同的团模式和环模式相比，我们考察的这两种模式需要更多采样器才能收敛，且增长速度快得多。考虑到在顶点数相同的情况下，团的边远多于所考察的模式，这些结果表明，除了纯粹的复杂度增长之外，还存在一些效率低下的问题。

造成这一现象的原因之一是许多真实世界图中观察到的无标度分布 [11, 12, 30]，其中顶点的度分布高度偏斜。假设我们沿着桥 $\{u_j, u_i\} \in E_p$（其中 $j < i$），对第 i 个顶点 v_i进行采样。因此，无法应用第 2.2 节中的连通性约束，因为根据桥的定义，除 u_j外，没有其他满足 $\pi^{-1}(u) < i$的顶点 u与 u_i共享一条边。因此，为简单起见省略对称性破缺后，$|S|$约等于 $d(v_j)$，即

v_j的度。由于无标度分布的图中度的差异很大，采样器的输出在估计中也会产生较大的方差，最终导致收敛速度变慢。此外，如果模式中的顶点 u_j 与 k 个桥相连，采样器的输出将大致与 $d(v_j)^{k-1}$ 成正比（见第 3.2 节），从而加剧这一问题。

3.2 重新思考采样概率

第 2.2 节介绍的三种采样方法（朴素方法、NS 和 NS-prune）通过缩小采样集的规模进行了改进。借助更有希望的候选集，找到嵌入的概率会提高，从而加快收敛。

这三种方法构建候选集S的方式各不相同，但它们都为 S中的元素赋予均匀采样概率。在朴素采样方法中，候选集是整个边集 E_G，并采用均匀采样概率。在 NS 中，候选集缩小为与已采样顶点相邻的边。对于 NS-prune，候选集进一步限定为满足连通性约束和对称性破缺限制的顶点。为这些集合元素赋予均匀概率是合理的选择，但并非必需。事实上，这些方法也可以解释为仅对第一个顶点赋予非均匀概率。三种方法在开始时都均匀采样一条随机边，这可以解释为：集合中包含所有顶点，且各顶点的概率与其度成正比。

基于这些观察，我们建议进一步增加一个额外步骤，用于计算定制的采样分布，从而为每个元素分配不同的概率，以实现更快的收敛。作为一个启发性示例，考虑 k-星模式。假设我们使用不进行对称性破缺的 NS-prune，并从中心顶点开始。由于均匀采样第一条边等价于按照与顶点度成正比的概率采样第一个顶点，因此中心顶点 v 被采样的概率为$d(v)\,2|E_G|$。随后，对于剩余的 k 个外围顶点，候选集定义为中心顶点的所有邻居中尚未被采样的顶点集合。因此，采样器输出为 $\frac{2|E_G|}{d\,v}() \times d(v) \times (d(v)-1) \times \cdots \times (d(v)-(k-1))$. 由于$d(v)$因顶点而异且变化很大，因此需要大量选代才能实现收敛。

另一方面，k-星的一个有趣之处在于，顶点 v 恰好可以构成 $\binom{d(v)}{k}$ 个这样的模式。如果我们放弃均匀采样，转而以与 $\binom{d(v)}{k}$成正比的概率对第一个顶点 v 进行采样，会怎么样？也就是说，$p = \binom{d(v)}{k}\Sigma_{x\in V_G}\binom{d(x)}{k}$？这样一来，采样器的输出将是 $\sum_{x\in V_G}\binom{d(x)}{k}\binom{d(v)}{k} \times d(v) \times (d(v)-1) \times \cdots \times (d(v)-(k-1)) = \sum_{x\in V_G}\binom{d(x)}{k} \times k!$，这是一个等于 $C(G,P)$ 的常数值。这意味着我们的采样器总是会返回相同的值，从而产生零方差，并且仅使用一个采样器就能立即收敛到 $C(G,P)$。

4 构建采样分布

尽管第3.2节的结果极具吸引力，但将这一原理应用于一般模式时，会面临两个关键挑战。

首先，对于 k-星这类简单模式，确定理想分布十分直接——我们可以精确地预先计算采样器输出；但如何为一般模式找到这样的分布尚不明确。其次，即使我们能够确定复杂模式的理想分布，直接计算也可能代价高昂，从而削弱更快收敛所带来的优势。本节将首先讨论一般模式的理想分布（第4.1节）、如何近似该分布（第4.2节）、其背后的理论直觉（第4.3节），以及如何据此对数据图进行预处理（第4.4节）。

4.1 一般模式的理想分布

本节首先给出广义模式的理想采样概率分布。

定义 1（采样轨迹）。给定一个图 G，采样轨迹 τ 是从 V_G 中采样得到的顶点有序序列，记为

$$\tau = (v_1, v_2, \ldots, v_k),$$

其中，$v_i \in V_G$ 是第 i个采样的顶点 .

如果我们在 τ之后采样一个顶点 $v \in V_G$，则将新的采样轨迹记为 $\tau' = \tau \circ v$，其中$\circ$ 表示连接。

定义 2（成功轨迹）。若采样轨迹 τ 表示一个嵌入，则该轨迹是成功的。也就是说，

$$|\tau| = |V_P| \quad and \quad \forall (u_i, u_j) \in E_P,\ (v_i, v_j) \in E_G.$$

定义 3（成功扩展的数量）。给定一个采样轨迹 $\tau = (v_1, v_2, \ldots, v_k)$，其中 $0 \le |\tau| \le |V_P|$，成功扩展的数量 n_τ 是通过从 V_G 中添加顶点，将 τ 扩展得到的不同成功轨迹的总数。形式化地，

$$n_\tau = \left| \left\{ \tau' \in V_G{}^{|V_P|} \;\middle|\; \begin{array}{l} \tau' = (v_1, v_2, \ldots, v_k, v_{k+1}, \ldots, v_{|V_P|}) \\ \tau' \textit{ is a successful trajectory} \end{array} \right\} \right|.$$

例如，若 $\tau = \emptyset$，则值 n_τ 为 $C(G,\ P)$。此外，对于满足 $|\tau| = |V_P|$的 τ，n_τ 要么为 1（τ 成功），要么为 0（不成功）。

定理 1（理想采样分布）。假设 P 在 G 中至少存在一个嵌入。为每个顶点 $v \in V_G$ 分配一个概率，使该概率与选择 v 时可生成的成功扩展数量成正比，其中 *i.e.*, $n_{\tau\circ v}$。形式化地，

$$\mathbf{f}_{ideal}(v \mid \tau) = \frac{n_{\tau\circ v}}{\sum_{x\in V_G} n_{\tau\circ x}}, \quad \mathbf{f}_{ideal}(\cdot \mid \tau) \in \mathbb{R}^{|V_G|}.$$

那么，使用这种概率分配的采样器是理想的，因为它返回的 $C(G, P)$ 具有零方差。

证明。假设我们按照上述分布 $\mathbf{f}_{\text{ideal}}$ 进行采样。由于 τ 在初始时是一个空序列 ()，我们从分布 $\mathbf{f}_1$, 中采样第一个顶点 v_1，该分布为

$$\mathbf{f}_1(v) = \mathbf{f}_{\text{ideal}}(v \mid \tau = ()) = \frac{n_{(v)}}{\sum_{x\in V_G} n_{(x)}}.$$

我们以概率 $\frac{n_{(v_1)}}{\Sigma_{x\in V_G} n_{(x)}}$ 采样第一个顶点 v_1，然后从以下分布中采样下一个顶点

$$\mathbf{f}_2(v) = \mathbf{f}_{\text{ideal}}(v \mid \tau = (v_1)) = \frac{n_{(v_1)\circ v}}{\sum_{x\in V_G} n_{(v_1)\circ x}}.$$

重复此过程，在第 i 步，我们得到采样轨迹 $\tau = (v_1, v_2, \ldots, v_{i-1})$，并从分布$\mathbf{f}_i$中采样第 i 个顶点 v_i，即：

$$\mathbf{f}_i(v) = \mathbf{f}_{\text{ideal}}(v \mid \tau = (v_1, \ldots, v_{i-1})) = \frac{n_{\tau \circ v}}{\sum_{x \in V_G} n_{\tau \circ x}}.$$

由于 P in G中至少存在一个嵌入，并且我们从不采样满足 $n_{\tau \circ v} = 0$的顶点 v，因此采样过程会持续到 $|\tau| = |V_P|$，而采样轨迹 $\tau = (v_1, v_2, \ldots\ldots, v_{|V_P|})$始终是成功的。

此外，对于任意此类 τ，采样器输出 X_τ 变为

$$X_\tau = \frac{\sum_{x \in V_G} n_{(x)}}{n_{(v_1)}} \cdot \frac{\sum_{x \in V_G} n_{(v_1) \circ x}}{n_{(v_1, v_2)}} \cdot \cdots \cdot \frac{\sum_{x \in V_G} n_{(v_1, \ldots v_{|V_P|-1}) \circ x}}{n_{(v_1, \ldots v_{|V_P|})}} = C(G, P),$$

由于$n_\tau = n_{(v_1, \ldots, v_{|V_P|})} = 1$，$\sum_{x \in V_G} n_{(x)} = C(G, P)$，并且各项会抵消，因为$\sum_{x \in V_G} n_{(v_1, \ldots v_i) \circ x} = n_{(v_1, \ldots v_i)}$。采样器始终返回嵌入数量 $C(G, P)$，其方差为 0。□

4.2 近似理想分布

我们已经证明，从理想分布 $\mathbf{f}_{\text{ideal}}(\cdot \mid \tau)$ 中采样可以得到无偏且方差为零的估计。然而，构建理想概率分布需要预先了解 $n_{\tau \circ v}$ 的比率，因此必须事先知道 $C(G, P)$。由于这是不可行的，我们需要一种高效的方法来构建近似分布$\mathbf{f}^{\text{approx}}(\cdot \mid \tau)$，使其能够紧密近似$\mathbf{f}_{\text{ideal}}(\cdot \mid \tau)$并产生无偏结果。

4.2.1 采样分布的无偏性。 在构造 $\mathbf{f}^{\text{approx}}$ 之前，一个自然的问题是：能否采用任意采样分布，同时仍保证所得估计量的无偏性？以下定理在特定条件下对此给出了肯定的回答。

定理 2。 假设采样分布 $\mathbf{f}(v \mid \tau)$ 为每个满足 $n_{\tau \circ v} > 0$ 的顶点 $v \in V_G$ 分配正概率。也就是说，该分布确保每条成功轨迹都可达。那么，该采样器是 $C(G, P)$ 的无偏估计量。

证明。设 X 为采样器的输出。根据定义，X 的期望值为

$$\mathbb{E}[X] = \sum_\tau \mathbb{1}_{\{\tau \text{ is successful}\}} X_\tau \, p_\tau.$$

其中，X_τ 表示采样器针对轨迹 τ 的输出，p_τ 表示采样器沿轨迹 τ前进的概率。该表达式成立，是因为对于任何不成功的轨迹 τ，采样器都会输出 $X_\tau = 0$。

首先，注意每条成功轨迹都对应 P 的唯一嵌入。其次，每个唯一嵌入都可以通过采样过程到达。因此，对于 G 的所有 $C(G, P)$ 个嵌入，都存在对应的成功轨迹 τ_1，……，$\tau_{C(G,P)}$。因此，我们可以写成

$$\mathbb{E}[X] = \sum_\tau \mathbb{1}_{(\tau \text{ is successful})} X_\tau \, p_\tau = \sum_{i=1}^{C(G,P)} X_{\tau_i} p_{\tau_i}.$$

此外，对于每条成功 τ_i，采样器都会输出 $X_{\tau_i} = \frac{1}{p_{\tau_i}}$，这是定义良好的。因此，期望值变为

$$\mathbb{E}[X] = \sum_{i=1}^{C(G,P)} X_{\tau_i} \, p_{\tau_i} = \sum_{i=1}^{C(G,P)} \frac{1}{p_{\tau_i}} \, p_{\tau_i} = \sum_{i=1}^{C(G,P)} 1 = C(G, P).$$

因此，该采样器是 $C(G, P)$ 的无偏估计量。 □

4.2.2 广义近似框架。 我们首先按照 [8] 的方法应用连通性约束并排除已采样的顶点，从而过滤掉不太可能的顶点。为采样第 i 个顶点（即 $i = |\tau| + 1$），我们将有希望的顶点构成候选集 $\mathbf{S}_\tau$，定义如下：

$$\mathbf{S}_\tau = \left\{ v \;\middle|\; v \notin \tau,\ v \in \bigcap_{u_j \in e \in \mathcal{B}, j < i} \mathcal{N}(v_j) \right\},$$

其中，$\mathbf{B}$ 是连接 u_i 与已采样顶点的后向边集合，即

$$\mathcal{B} = \{\{u_j, u_i\} \in E_P \mid j < i\}.$$

我们为所有 $v \notin \mathbf{S}_\tau$ 赋值 $\mathbf{f}^{\text{approx}}(v \mid \tau) = 0$，因为可以保证 $n_{\tau \circ v} = 0$。

现在，我们的目标是近似顶点 $v \in \mathbf{S}_\tau$ 对应的 $n_{\tau \circ v}$ 比率。关键思路是结合数据图 G 考虑模式 P 的结构组件。首先，根据各顶点在子图$P_i = (V_i, E_i)$ 中与 u_i 的距离对其分组；该子图包含 P 中尚未采样的顶点：

$$V_i = \{u_j \in V_P \mid j \geq i\},\ E_i = \{\{u_a, u_b\} \in E_P \mid u_a, u_b \in V_i\}. \quad (1)$$

接着，我们将 k 跳顶点组定义为 P_i 中与 u_i 的距离在 P_i 中为 k 的顶点集合。利用这些信息，我们将 $n_{\tau \circ v}$ 分解为若干项的乘积，每一项分别对应不同类型的边，并按 k 跳信息进行划分：

$$n_{\tau \circ v} \approx \prod_{k=1}^{D} T[\mathcal{F}_k](v) T[\mathcal{I}_k](v), \quad (2)$$

其中

- $T[\mathbf{F}_k](v)$ 是用于表示前向边集合 $\mathbf{F}_k$ 的项。具体而言，$\mathbf{F}_k \subseteq E_i$ 由所有连接第 $k-1$ 跳顶点与第 k 跳顶点的边组成。
- $T[\mathbf{I}_k](v)$ 是用于表示内部边集合 $\mathbf{I}_k$ 的项。具体而言，$\mathbf{I}_k \subseteq E_i$ 由同一跳 k 上的所有顶点之间的边组成。
- D 是从 u_i 到 P_i中任何其他连通顶点的最大跳数。

请注意，方程 (2) 未考虑 P 的所有边。例如，P_i 中可能存在未与 u_i 相连的顶点，也可能存在连接 hop k 处顶点与先前已采样顶点的边。尽管理论上可以考虑所有这些边，但我们发现，只关注 $\mathbf{F}_k$ 和 $\mathbf{I}_k$ 类型的边已足够有效。此外，虽然计算所有 D-hop 项可以得到更准确的概率近似，但我们发现这会带来过高的计算开销，而改进却微乎其微。我们发现，使用方程的前三项即可实现有效近似：

$$n_{\tau \circ v} \approx T[\mathcal{F}_1](v) \cdot T[\mathcal{I}_1](v) \cdot T[\mathcal{F}_2](v).$$

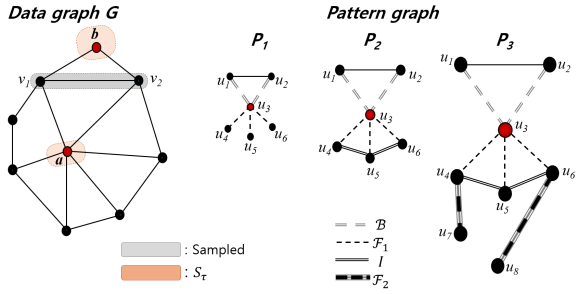


图3：用于近似的示例图和模式。

这会引入少量近似误差；不过，这些误差只会影响收敛速率，不会影响嵌入计数 $C(G,\ P)$ 的正确性。我们将在第 4.2.3 节中给出每一项对应的方程，确保除非某个顶点的概率确实为零，否则不会将其分配为零概率，从而保证无偏性（定理 2）。

4.2.3 近似细节。 在图3所示的示例中，我们从图 G 中对模式 P_1, P_2, P_3 的第三个顶点 $\pi(3) = u_3$ 进行采样。由于 u_3 与 u_1 和 u_2 相连，因此对应的顶点 v_3 必须同时与 v_1 和 v_2 在 G 中相连。因此，候选集 S_τ 由同时邻接于 v_1 和 v_2、且尚未包含在 τ 中的顶点组成；在此例中，$S_\tau = \{a, b\}$。

现在，考虑模式 P_1 对应的项 $T[\mathcal{F}_1](v)$。在 P_1 中，我们有 $\mathcal{B} = \{\{u_1, u_3\}, \{u_2, u_3\}\}$，以及 $\mathcal{F}_1 = \{\{u_3, u_4\}, \{u_3, u_5\}, \{u_3, u_6\}\}$。对于顶点 $v \in S_\tau$，它已经通过 $|\mathcal{B}|$ 条边与先前采样的顶点相连。因此，我们需要从 $d(v) - |\mathcal{B}|$ 个可能的邻居中选取 $|\mathcal{F}_1|$ 条边。因此，我们使用，

$$T[\mathcal{F}_1](v) = \binom{d(v) - |\mathcal{B}|}{|\mathcal{F}_1|}.$$

例如，对于顶点 $a \in S_\tau$, $T[\mathcal{F}_1](a) = \binom{6-2}{3}$。

接下来考虑项 $T[\mathcal{I}_1]$，其模式更一般，为 P_2。在这种情况下，$\mathcal{I}_1 = \{\{u_4, u_5\}, \{u_5, u_6\}\}$。为了考虑这些边，我们必须估计 p_v：在 v 的 $|\mathcal{F}_1|$ 个邻居之间，存在 $|\mathcal{I}_1|$ 条必要边的概率。一种朴素的估计方法是利用 v [83] 的局部聚类系数 C_v 来估计 p_v。由于 C_v 表示 v 的两个邻居之间存在边的概率，而所需的边有 $|\mathcal{I}_1|$ 条，因此我们可以将 p_v 估计为 $C_v^{|\mathcal{I}_1|}$。

然而，朴素地将 p_v 估计为 $C_v^{|\mathcal{I}_1|}$，往往会过分强调 C_v 的影响，导致近似结果不准确。这是因为真实世界图往往会形成簇 [34, 35, 56, 57, 82],，而且 $|\mathcal{I}_1|$ 条边彼此并非独立。例如，在已知 $a, b, c \in \mathcal{N}(v)$，且边 $\{a, c\}, \{b, c\}$ 存在的条件下，边 $\{a, b\}$ 存在的概率，通常会因聚类效应而高于 C_v。

因此，我们不再对 $\mathcal{I}$ 中的每条边都乘以 C_{v1},，而是构造一个由边组成的有效子集 $\mathcal{I}_{\text{effective}} \subseteq \mathcal{I}_1$，其中各边可视为相互独立。为此，我们考虑由 $\mathcal{I}_1$ 诱导的子图： $P_{\text{ind}} = (V_{\text{ind}},\ E_{\text{ind}})$，其中 $V_{\text{ind}} = \{u \mid u \in e \in \mathcal{I}_1\}$，且 $E_{\text{ind}} = \mathcal{I}_1$。然后，我们选取 P_{ind} 的一个生成森林，其边集为 $\mathcal{I}_{\text{effective}}$。接着，我们估计

$$T[\mathcal{I}_1](v) = C_v^{|\mathcal{I}_{\text{effective}}|}.$$

对于模式 P_2，诱导子图 P_{ind} 在三个顶点 u_4，u_5，u_6 上构成一条简单路径。由于路径本身已经是一棵树，整个边集 $\mathcal{I}_1$ 就是它自身的生成树。因此，对于这一特定情况，我们设 $\mathcal{I}_{\text{effective}} = \mathcal{I}_1 = \{\{u_4,\ u_5\},\ \{u_5,\ u_6\}\}$。于是我们得到 $T[\mathcal{I}_1](a) = C_a^2 = (5/15)^2$。

最后， $T[\mathcal{F}_2](v)$ 考虑了模式的两跳拓扑。对于模式 P_3，$\mathcal{F}_2 = \{\{u_4,\ u_7\},\ \{u_6,\ u_8\}\}$。对于 $\mathcal{F}_2$，中的每条边，成功扩展的数量估计会与对应顶点的度成正比。例如， $T[\mathcal{F}_2](v) = d(v_4) \cdot d(v_6)$ 恰当地考虑了与 u_7 和 u_8 的潜在连接。然而，由于我们不知道未来将采样到哪些确切的一跳邻居（v_4，v_6），因此实际计算时使用 v 的邻居的平均度。因此，

$$T[\mathcal{F}_2](v) = \left(\frac{\sum_{x \in \mathcal{N}(v)} d(x)}{d(v)}\right)^{|\mathcal{F}_2|}.$$

例如， $T[\mathcal{F}_2](a) = \left(\frac{4+4+3+3+3+3}{6}\right)^2$，按照顺时针方向从 v_1 开始考虑 a 的邻居。

我们给出了正式定义，并提供了一个用于计算 $\mathbf{f}_{\text{approx}}(\cdot \mid \tau)$ 的方程。

定义 4（辅助数组）。给定一个模式 P及其匹配顺序 π：$\{1, \ldots, |V_P|\} \to V_P$，使得 $u_i = \pi(i) \in V_P$是第i个模式顶点，我们将辅助数组 A^π 定义为由集合数组构成的集合：

$$A^\pi = \left\{\mathcal{B}^\pi,\ \mathcal{F}_1^\pi,\ \mathcal{F}_2^\pi,\ \mathcal{I}^\pi_{\text{effective}}\right\},$$

其中，每个 $\mathcal{B}^\pi$、$\mathcal{F}_1^\pi$、$\mathcal{F}_2^\pi$、$\mathcal{I}^\pi$有效 $i = 1, \ldots, |V_P|$索引的数组，而每个元素 $\mathcal{B}^\pi[i], \ldots, \mathcal{I}^\pi$有效$[i]$ 是与顶点 $u_i = \pi(i)$相关联的集合。

基于 A^π，用于近似 $n_{\tau \circ v}$ 的方程如下，其中 v 是第 i 个顶点 v_i 的候选项：

$$n_{\tau \circ v} \approx \underbrace{\binom{d(v) - |\mathcal{B}^\pi[i]|}{|\mathcal{F}_1^\pi[i]|}}_{T[\mathcal{F}_1]} \times \underbrace{C_v^{|\mathcal{I}^\pi_{\text{effective}}[i]|}}_{T[\mathcal{I}_1]} \times \underbrace{\left(\frac{\sum_{x \in \mathcal{N}(v)} d(x)}{d(v)}\right)^{|\mathcal{F}_2^\pi[i]|}}_{T[\mathcal{F}_2]}. \quad (3)$$

我们为所有候选顶点 $v \in S_\tau$计算该值，并使用归一化值作为分布 $\mathbf{f}_{\text{approx}}(\cdot \mid \tau)$。

4.3 样本复杂度分析

为了深入理解我们在第 4.2.3 节提出的采样器，我们分析其样本复杂度。对于每条成功轨迹 τ_i，采样器的输出可以写为

$$X_{\tau_i} = C(G, P) \times (1 + \eta_i),$$

其中， η_i 表示由于使用$\mathbf{f}_{\text{approx}}$ 代替理想分布$\mathbf{f}_{\text{ideal}}$而产生的与轨迹$\tau_i$ 相关的乘法误差。由于不成功的轨迹会产生 $X_\tau = 0$，因此采样器输出的二阶矩为

$$\mathbb{E}[X^2] = \sum_\tau \mathbb{1}_{\{\tau \text{ successful}\}} X_\tau^2 p_\tau = \sum_{i=1}^{C(G,P)} X_{\tau_i}$$
$$= C(G, P)^2\ (1 + \overline{\eta}),$$

其中，$\bar{\eta}$ 为平均乘法误差

$$\bar{\eta} = \frac{1}{C(G,P)} \sum_{i=1}^{C(G,P)} \eta_i.$$

结合定理 2 与切比雪夫不等式，当

$$N \geq \frac{\bar{\eta}}{\epsilon^2 \delta}. \tag{4}$$

这给出了 N（所需采样器的数量）、准确度 $\mathbf{f}_{\text{approx}}$ 以及参数（ϵ、δ）之间的明确关系。$\mathbf{f}_{\text{approx}}$ 越接近$\mathbf{f}_{\text{ideal}}$，$\eta$ 就越接近零，所需的采样器也越少。

作为示例，考虑最简单的非退化模式——三角形。对于一条成功轨迹 $\tau_i = (v_1, v_2, v_3)$，所提出的采样器输出的乘法误差 η_i 为

$$\eta_i = \left[\frac{1}{d(v_1)\big(d(v_1)-1\big)\, C_{v_1}} \sum_{v \in \mathcal{N}(v_1)} \big(d(v)-1\big) \cdot \frac{\big|\mathcal{N}(v_1) \cap \mathcal{N}(v_2)\big|}{d(v_2)-1} \right] - 1. \tag{5}$$

比率 $\frac{|\mathcal{N}(v_1) \cap \mathcal{N}(v_2)|}{d(v_2)-1}$ 近似于从 v_1 的某个邻居出发且仍留在 $\mathrm{N}(v_1)$ 内的边的比例。由于 $\sum_{v \in \mathcal{N}(v_1)} (d(v) - 1)$ 统计了与 $\mathrm{N}(v_1)$ 相连的所有边，因此乘积

$$\sum_{v \in \mathcal{N}(v_1)} \big(d(v)-1\big) \, \frac{|\mathcal{N}(v_1) \cap \mathcal{N}(v_2)|}{d(v_2)-1}$$

可作为 $\mathrm{N}(v_1)$ 内部边数的代理，其精确值为 $d(v_1)\,\big(d(v_1)-1\big) C_{v_1}$。因此，无论数据图 G 如何，式 (5) 因而也包括 η，通常都接近于零。

另一方面，对于基于 NS-prune 且未进行对称性破缺的采样器，误差为

$$\eta_i' = \left[\frac{2|E_G| \cdot |\mathcal{N}(v_1) \cap \mathcal{N}(v_2)|}{C(G,P)} \right] - 1. \tag{6}$$

如果 G 是均匀的，每条边都会产生相同数量的三角形，且式 (6) 恰好为 0。然而，随着图偏离均匀性，η_i' 也会进一步偏离 0，因此需要更多采样器。因此，式 (4) 至式 (6) 表明，第 4.2.3 节中的近似策略可能比现有方法有效得多。

4.4 数据图预处理

对于给定的图 G，我们会预处理三项信息，用于根据式 (3) 计算$\mathbf{f}_{\text{approx}}$。

(1) $\binom{n}{k}$ for $n \geq k$ 的二项式系数表。(2) 平均邻居度信息 $\sum_{x \in \mathcal{N}(v)} d(x)\, d(v)$。(3) 聚类系数 C_v。

前两项在图加载阶段的处理成本相对较低，而计算 C_v 的精确值则十分耗时，因为这相当于统计图中的所有三角形。我们改为通过采样来估计 C_v。对于每个顶点v，我们执行 $d(v)$次采样，并根据收集到的数据估计 C_v。我们将在第 6 节讨论预处理时间。我们还将估计聚类系数的值限制在一个较小的非零常数，以确保$\mathbf{f}_{\text{approx}}$ 不包含任何意外的零值，从而保证定理 2 的假设成立。

5 AGIS 系统设计

基于$\mathbf{f}_{\text{approx}}$，我们构建 AGIS 系统。我们将介绍如何策略性地利用$\mathbf{f}_{\text{approx}}$ 来加速系统（第 5.1 节）、如何构建匹配顺序（第 5.2 节），随后介绍完整流程（第 5.3 节），以及系统概述和实现细节（第 5.4 节）。

5.1 结构感知邻居采样

所提出的 $\mathbf{f}_{\text{approx}}$ 可以大幅减少达到收敛所需的采样器数量。然而，直接使用 $\mathbf{f}_{\text{approx}}$ 有时会比使用 $\mathbf{f}_{\text{uniform}}$ 导致更长的执行时间。虽然从 $\mathbf{f}_{\text{uniform}}$ 进行采样只需 $O(1)$ 时间，但从 $\mathbf{f}_{\text{approx}}$ 进行采样时，构建分布需要 $O(|V_G|)$ 时间。尽管采样器的数量仍显著少于使用$\mathbf{f}_{\text{uniform}}$时所需的数量，但总体延迟可能更长。

为了解决这一问题，所提出的结构感知邻居采样包含一种决策启发式方法，该方法根据 P 中剩余未采样子图的结构，确定每个顶点所采用的采样分布类型。在该决策启发式方法中，我们考虑剩余未采样子模式的密度。我们发现，模式中未采样部分越稠密，$n_{\tau \circ v}$ 就越难以预测，此时使用$\mathbf{f}_{\text{uniform}}$ 更有益。

使用边数与顶点数之差来表示密度，我们定义如下决策函数：

定义 5（决策函数）。给定一个模式 P、一个匹配顺序 π：$\{1, \ldots, |V_P|\} \to V_P$，使得 $u_i = \pi(i) \in V_P$，以及一个阈值 $\beta \leq 1$，决策函数 $\mathcal{D}^\pi$：$\{1, \ldots, |V_P|\} \to \{0, 1\}$ 定义如下：

(1) 使用$\mathbf{f}_{approx}$：的确定性建模

- *Define* h_i 个操作点值于 P_i（V_i, E_i）$h \subseteq P$ 处（式 (1)）：

$$V_i^{\geq 2hop} = \{u \in V_i \mid 2 \leq dist_{P_i}(u_i, u) < \infty\}.$$

- 定义与 $V^{\geq 2}$跳$_i$ 相连的边集合：

$$E_i^{\geq 2hop} = \left\{ \{u_a, u_b\} \in E_P \,\middle|\, u_a \in V_i^{\geq 2hop} \text{ or } u_b \in V_i^{\geq 2hop} \right\}.$$

- 定义使用$\mathbf{f}approx$ 的确定性：

$$Certainty(i) = 1 - \frac{|E_i^{\geq 2hop}| - |V_i^{\geq 2hop}|}{|V_i|}.$$

该值的上限为 *1*，即确定性$(i) \leq 1$。

(2) 决策函数：

$$\mathcal{D}^\pi(i) = \mathbb{1}_{(Certainty(i) \geq \beta)}$$

这里，我们使用 $\beta = 0.8$ 作为默认值。$\mathrm{D}^\pi(i) = 1$ 表示我们使用 $\mathbf{f}_{\text{approx}}$ 进行采样 v_i，而 $\mathrm{D}^\pi(i) = 0$ 表示我们使用 $\mathbf{f}_{\text{uniform}}$。这样，我们就能根据剩余子模式的拓扑，结合这两种分布各自的优势。请注意，仅凭 P 即可预先计算此值，且成本可忽略不计。

5.2 匹配顺序构建

由于决策函数 D^π 取决于匹配顺序 π，因此明智地构造 π 对于最大限度地利用 $\mathbf{f}_{\text{approx}}$至关重要。为了

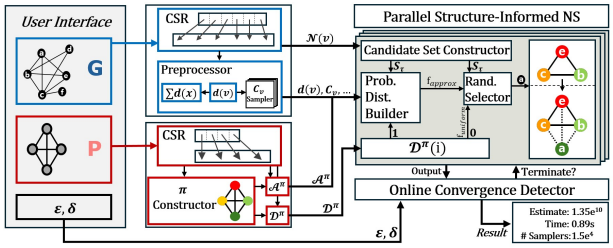


图4：AGIS 系统概览。

为此，我们提出一种贪心算法，逐步构建匹配顺序，具体如算法 2 所示。

算法 2 匹配顺序构建

```
1: 初始化空列表 π ← [ ]
2: 对于 i 在 [1, |V_P|] 执行
3:     如果 i = 1，则
4:         N ← V_P
5:     else
6:         N ← {u ∈ V_P \ π | ∃u' ∈ π 使得 {u, u'} ∈ E_P}
7:     结束 if
8:     对于 u ∈ N 执行
9:         π_u ← π + u, s(u) ← |F_1^{π_u}[i]| + |I_1^{π_u}[i]|
10:    结束 for
11:    π ← π + argmax_{u∈N} s(u)
12: </style}结束循环
13: 返回 π
```

如第 3–7 行所示，我们首先构造一个由相邻顶点组成的候选集 N，以便始终以连通的方式探索模式。在第 8–10 行中，假设顶点 $u \in V_P$ 被选为匹配顺序中的下一个顶点，我们按照第 4.2.3 节中的定义，计算所有 $u \in$N 的分数 $s(u)$，其中 $|F_1^{\pi_u}[i]|$ 和 $|I_1^{\pi_u}[i]|$。在第 11 行中，通过选择使该总和最大的顶点，我们优先选择具有最多尚未采样的 1 跳结构的顶点。这种方法使 f_{approx} 被选中的频率更高，因为定义 5 中的确定性 值会更高。

5.3 完整采样过程

AGIS 中使用的采样算法流程如算法 3 所示。在第 4–9 行，我们利用连通性约束构造候选集 S_τ。如果不存在可能的顶点（第 10 行），则返回 0 并终止采样过程。否则，我们根据决策函数 D^π（第 11–17 行）构建概率分布函数。随后，我们按照该分布从 S_τ中采样一个顶点，并相应地更新概率 p（第 18–19 行）。重复此顶点采样过程 $|V_P|$ 次后，返回 $1/p$ 作为采样器输出。

我们注意到，仅基于$f_{uniform}$ 进行采样，等同于不应用对称性破缺的 ScaleGPM NS-prune 程序。$f_{uniform}$ 的定义因步骤 i而异。对于$i = 1$ （采样第一个顶点），$f_{uniform}$ 为边均匀，这意味着顶点被采样的概率与其

算法 3结构感知邻居采样

```
1: 过程 Sample(G, P, π, D^π, A^π)
2:     初始化 τ ← [], p ← 1.0
3:     for i属于 [1, |V_P|] do
4:         ▷ 构建候选顶点集合 S_τ
5:         if i = 1 then
6:             S_τ ← V_G
7:         else ← {v | v ∉ τ, v ∈ ⋂{u_j, u_i}∈E_P, j<i N(v_j)}
8:             S_τ
9:         结束 if
10:        如果 |S_τ| = 0 则返回 0
11:        ▷ 构建概率分布
12:        如果 D^π（i） = 1 则
13:            使用公式（3）计算 f_approx，其中（S_τ, A^π, i）
14:            f ← f_近似
15:        else
16:            f ← f_uniform
17:        结束 if
18:        v ← rand_select(S_τ, f)
19:        τ ← τ + v, p = p · f(v)
20:    结束循环
21:    返回 1/p
22: 结束过程
```

度成正比。对于 $i \geq 2$，$f_{uniform}$ 为顶点均匀，因此S_τ 中的每个顶点被采样的概率都相同。

5.4 系统概览

基于结构感知邻居采样，我们提出了 AGIS，这是一个用于快速近似图模式挖掘的系统。图4展示了 AGIS 的高层概览。系统首先接收用户提供的四项输入：(1) 一个图文件 G，(2) 一个模式文件 P，(3) ϵ，以及 (4) δ。系统以 CSR 格式读取并加载图。如果图尚未经过预处理，预处理器会计算额外信息（第4.4节）。对于模式，系统首先构建匹配顺序 π，随后利用该顺序计算 D^π和 A^π。

准备好这些信息后，采样引擎将并行运行，每个线程都会分配一个独立的采样器。每经过固定数量的迭代，在线收敛检测器（算法1）便会检查采样器输出是否收敛。该检测器由 ScaleGPM 提出，并利用 ϵ和 δ。最后，系统会将估算的模式计数等结果输出给用户。

当查询模式 P 为团时，一种常用方法是应用定向优化 [24, 41, 72, 77]。它通过移除从高度顶点指向低度顶点的边，将无向图转换为有向无环图（DAG）。与采用这种方法的 ScaleGPM [8], 类似，我们对团模式应用相同的优化。

6 评估

AGIS 使用 C++ 实现，并通过 OpenMP 支持多线程。我们在多种数据图和模式上评估了 AGIS。我们使用了涵盖多种规模的真实世界图，如表 1 所示。对于待挖掘的模式，我们选取了多种模式，其中一部分取自先前的研究 [8, 90], ,

图	$\|V_G\|$	$\|E_G\|$	最大度
LiveJournal（Lj） [10, 49]	4.8×10^6	4.3×10^7	2.0×10^4
Uk-2002（Uk） [15–17]	1.8×10^7	2.6×10^8	1.9×10^5
Twitter（Tw） [48, 49]	4.2×10^7	1.2×10^9	3.0×10^6
Friendster（Fs） [49, 86]	6.6×10^7	1.8×10^9	5.2×10^3
Gsh-2015 (Gsh) [16, 17]	9.9×10^8	2.6×10^{10}	5.9×10^7

表 1：用于评估的图数据集。

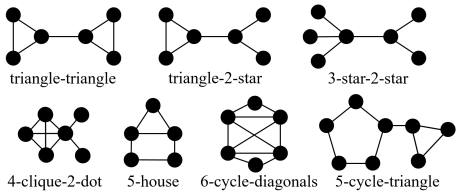


图 5：用于评估的模式。

实例	4-团				4-团-2-点				6-团			
	Lj	Uk	Tw	Fs	Lj	Uk	Tw	Fs	Lj	Uk	Tw	Fs
AGIS	**0.002**	**0.006**	**0.016**	0.028	**0.003**	**0.069**	**5.330**	**0.045**	**0.001**	**0.005**	**0.017**	**0.143**
ScaleGPM	0.007	0.014	0.018	**0.005**	0.052	191.9	1930	0.089	0.095	0.843	0.377	2.460
Arya	504.4	688.6	TO	TO	2126	1528	TO	TO	TO	TO	TO	TO
ASAP	958.2	462.4	TO	TO	TO	TO	TO	TO	TO	TO	TO	TO
Peregrine	6.779	128.0	TO	199.5	TO	TO	TO	TO	TO	TO	TO	1114
实例	3-星-2-星				三角形-2-星				三角形-三角形			
	Lj	Uk	Tw	Fs	Lj	Uk	Tw	Fs	Lj	Uk	Tw	Fs
AGIS	**0.002**	**0.024**	**0.840**	**0.028**	**0.002**	**0.081**	**4.701**	**0.030**	**0.001**	**0.013**	**4.738**	**0.044**
ScaleGPM	404.1	TO	TO	0.054	0.197	40.53	5009	0.043	0.025	0.186	33.75	0.096
Arya	216.4	1345	154.1	135.4	140.2	177.2	632.1	1640	686.3	1827	9148	TO
ASAP	2372	TO	TO	3.067	TO	TO	TO	6273	4628	TO	TO	TO
Peregrine	TO	TO	TO	TO	TO	TO	TO	TO	TO	TO	TO	TO

表 2：$\mathcal{D}^{\pi} = 1$对应模式的执行时间（秒）。TO：超时。bol 中最快的时间 d.

如图 5 所示。我们将 AGIS 与最先进的 AGPM 系统 ASAP [42], 、Arya [90] 和 ScaleGPM [8]，以及一个精确挖掘系统 Peregrine [43] 进行了比较。我们使用了 Peregrine 和 Arya 的官方实现。由于 ASAP 的代码并未公开，我们从头忠实地复现了 ASAP。由于 ScaleGPM 的官方实现需要针对每种模式单独定制，我们实现了一个通用版本的 ScaleGPM；在适用的情况下，报告我们实现与官方代码中观察到的最短执行时间。

对于所有系统，我们均未计入加载输入图所需的时间。我们还排除了针对模式的预处理步骤所花费的时间，例如 Arya 中求解模式分解的线性规划、ScaleGPM 中构造自同构和限制集，以及 AGIS 中计算 A^{π}，D^{π}。对于 ScaleGPM，我们仅使用其严格模式（默认模式）。这是因为 ScaleGPM 的宽松模式采用图稀疏化和精确计数，不使用在线收敛，因此无法提供高置信度的结果。

为了验证估计值，我们将结果与 Peregrine 进行了比较，比较范围涵盖 Peregrine 能在适当时间内挖掘的所有（图，模式）对。对于 Peregrine 无法给出精确计数的配对，真实值未知。不过，对于所有情况，我们都确认 ScaleGPM 和 AGIS 得出的估计值相近，从而确保 $|(X_s - X_a)/((X_s + X_a)/2)| < 2\epsilon$，其中 X_s 和 X_a 分别是 ScaleGPM 和 AGIS 得出的结果。

所有基线方法和 AGIS 的实验均在配备 48 个物理核心和 512GB 动态随机存取存储器的 AMD Ryzen Threadripper PRO 7985WX 上进行。所有运行的超时时间均设为 1.0×10^4 秒。

6.1 整体性能

表 2 和表 3 展示了不同模式的主要结果。除非另有说明，我们遵循最常见的做法 $\epsilon = 0.1$，$\delta = 0.01$。总体而言，AGIS 在大多数模式上速度最快，并成功挖掘了所有模式，未遇到超时。具体来说，AGIS 相较于 ScaleGPM 的几何平均加速比为 28.5×（超时时间设为 10^4 秒），相较于 Peregrine 的几何平均加速比为 63, 108×。

正如预期，精确 GPM 系统 Peregrine 在大多数情况下都无法在给定的时间限制内得出结果。对于大于 Lj 的图，Peregrine 只能挖掘可进行重度剪枝的团模式。这清楚地表明，相较于精确挖掘系统，我们需要 AGPM 系统。

在基线方法中，ScaleGPM 的整体性能最佳。我们还观察到，得益于有向无环图定向优化，ScaleGPM 在团模式上的表现最佳。然而，对于某些（图，模式）组合，ScaleGPM 需要很长时间才能收敛。例如，对于 Lj 上的 3 星-2 星模式，ScaleGPM 的收敛时间比 AGIS 长 100 多000×。这一观察结果与第 3.1 节中的发现一致：在偏斜度分布下，包含桥的模式会带来显著挑战。

通过比较 Fs 图上的结果，我们可以进一步观察到这一现象。Fs 的独特之处在于其度分布近似均匀。如表 1 所示，Fs 的边比 Tw 多，但其最大度仅约为 Tw 的 0.2%。由于

实例	5-屋		6-环对角线		5-环三角形	
	Lj	Tw	Lj	Tw	Lj	Tw
AGIS	**0.003**	**2.775**	**0.003**	**0.580**	**0.036**	**188.9**
ScaleGPM	0.009	31.72	0.022	27.25	0.041	2002
Arya	65.83	5490	TO	TO	TO	TO
ASAP	117.8	TO	TO	TO	TO	TO
Peregrine	669.1	TO	1483	TO	TO	TO

表 3：$\mathcal{D}^{\pi} \neq 1$的模式执行时间（秒）。TO：超时。

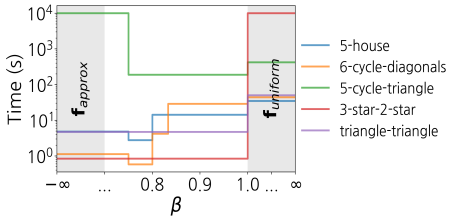


图 6：β 对 Tw图执行时间的影响。

模式	4-团	5-屋	三角形-三角形
AGIS	0.443	**89.11**	**1120**
ScaleGPM	**0.130**	731.1	TO
Arya	OoM	OoM	OoM
ASAP	TO	TO	TO
Peregrine	TO	TO	TO

(a) 在 Gsh图上的执行时间。

	Lj，4-团			Lj，5-屋		
误差界 ϵ	10%	1%	0.1%	10%	1%	0.1%
AGIS	**0.002**	**0.008**	**0.528**	**0.003**	**0.026**	**2.314**
ScaleGPM	0.008	0.689	68.86	0.012	1.240	129.0
Arya	504.4	8545	TO	65.83	1037	3910
ASAP	958.2	1901	TO	117.8	573.6	TO

(b) 不同误差界下的执行时间。

表 4：执行时间 （秒），分别对应 (a) 大型图和 (b) 不同误差界。TO：超时。最快时间以粗体显示 .

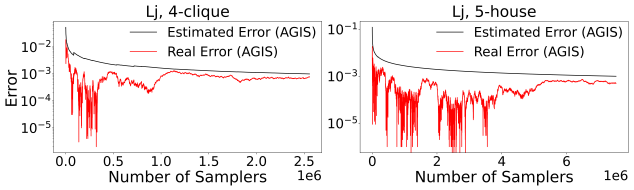


图 7：AGIS 上在线收敛算法的实际误差与估计误差对比。

Fs 的度分布并未严重偏斜，因此 ScaleGPM 能够有效处理 3-星-2-星和三角形-2 星模式。此外，从更根本的角度来看，$\mathbf{f}_{\text{uniform}}$ 在 Fs 上获得了意料之外的优势，因为$\mathbf{f}_{\text{uniform}}$ 会更接近$\mathbf{f}_{\text{ideal}}$。

此外，通过类似的推理过程，我们可以理解为什么在高度偏斜的图中，Arya 在此类模式上的表现优于 ScaleGPM。这是因为，与 ScaleGPM 不同，Arya 试图在子采样图上实现收敛，从而有效缩小其度分布的范围。然而，如第 2.3 节所述，ELP 不提供收敛保证，并且观察到 ELP 启发式方法的结果极不稳定。

这些模式分为两张表。对于表 2 中考察的模式，确定性值（定义 5）始终最大（等于 1）。因此，无论选择哪个 $\beta \leq 1$ 值，AGIS 都会始终使用$\mathbf{f}_{\text{approx}}$。相比之下，表 3 中的模式，其确定性值并非始终等于 1。具体而言，在五边形房屋模式中，会从五个顶点中均匀采样一个；在 6-环对角线中，会从六个顶点中均匀采样一个；而在 5-环三角形中，则会从八个顶点中均匀采样三个。通过对包括图 6 所示模式在内的大量模式进行测试，我们发现，选择阈值 $\frac{4}{5} \leq \beta \leq \frac{5}{6}$ 能在多种模式上取得稳健且出色的性能。图 6 展示了 β 对性能的影响。这里，$\beta = -\infty$ 表示始终使用$\mathbf{f}_{\text{approx}}$，而 $\beta = 1$ 表示以最保守的方式使用$\mathbf{f}_{\text{approx}}$。为进行实验，我们将 β 的取值范围扩展到包含满足 $\beta > 1$的取值，这等同于始终选择$\mathbf{f}_{\text{uniform}}$。可以看到，与使用$\mathbf{f}_{\text{uniform}}$ ($\beta = +\infty$)相比，采用$\mathbf{f}_{\text{approx}}$ ($\beta = -\infty$)通常能更快收敛。不过，每种模式都能在范围 $-\infty < \beta \leq 1$内的某个 β取值下达到最佳性能，这表明在完全使用$\mathbf{f}_{\text{approx}}$ 和完全使用$\mathbf{f}_{\text{uniform}}$这两个极端之间存在一个最优点。通过采用所提出的 D^{π}，AGIS 能有效平衡$\mathbf{f}_{\text{approx}}$ 与$\mathbf{f}_{\text{uniform}}$的使用。

表 4a 展示了在超大图 Gsh 上的性能结果，该图包含超过 250 亿条边。如图所示，只有 AGIS 能够扩展到如此大的规模。对于 4 团模式，ScaleGPM 的速度更快，因为构建 $\mathbf{f}_{\text{approx}}$ 的开销超过了该模式本身的复杂度。此外，表 4b 报告了不同误差界下的性能。如表中所示，只有 AGIS 能够在几秒内处理 $\epsilon = 10^{-3}$。我们还验证了返回的计数符合给定的 ϵ。在图 7 中，我们将在线收敛检测器给出的估计误差与使用精确挖掘结果测得的实际误差进行比较。对于不超过 10^{-3} 的误差水平，实际误差始终不超过估计误差，这证实了在线收敛方法在 AGIS 中能够有效运行。

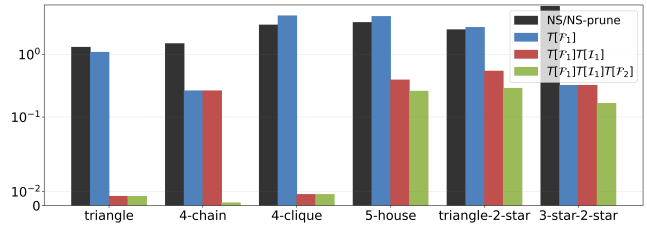
我们注意到，在少数边界情况下，AGIS 的速度不如 ScaleGPM（Fs 和 Gsh 上的 4 团）。当目标模式较小且理想分布接近均匀时，预先计算全图顶点采样分布的成本可能超过其收益，使得复杂的前期设置变得没有必要。

6.2 近似分布分析

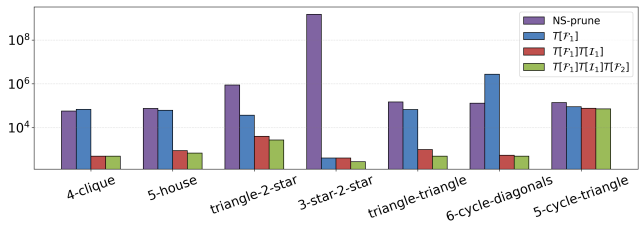
现在，我们将更详细地考察用于采样分布的估计技术。表5比较了 AGIS 与 ScaleGPM 达到收敛所需的采样器数量，二者采用相同的收敛检测算法。如表所示，AGIS 所需的采样器数量少了数个数量级。例如，ScaleGPM 需要多出三百万倍的采样器才能实现

实例	4-团		4-团-2-点		6-团		3-星-2-星		三角形-2-星		三角形-三角形	
	Lj	Tw	Lj	Tw	Lj	Tw	Lj	Tw	Lj	Tw	Lj	Tw
AGIS（结构信息增强的NS）	**5.0e2**	**2.3e3**	**5.0e2**	**4.0e3**	**7.5e2**	**1.0e3**	**5.0e2**	**5.0e2**	**2.8e3**	**1.4e4**	**5.0e2**	**1.1e4**
ScaleGPM (NS-prune)	5.7e4	1.3e5	2.9e5	8.2e7	1.2e6	2.8e6	1.5e9	χ	8.8e5	5.6e7	1.5e5	2.3e6

表5：收敛所需的采样器数量。X表示收敛失败。最小数量以粗体显示。



（a）各分布（$\mathbf{f}_{approx}$ 和 $\mathbf{f}_{uniform}$）与理想分布 $\mathbf{f}_{ideal}$之间的 KL散度。



（b）收敛所需的采样器数量。

图8：所提出近似技术的效率。在 Lj图上进行测试。

Lj 中 3 星-2 星模式的收敛。这表明，AGIS 的根本加速来自利用经过准确计算的 $\mathbf{f}_{approx}$，而不是依赖朴素的$\mathbf{f}_{uniform}$。

图8展示了随着纳入更多项，$\mathbf{f}_{approx}$ 如何得到改进。在图8a中，我们首先计算每个顶点 $v \in V_G$的$n(v)$ 实际值。利用这些信息，我们计算用于对模式的第一个顶点进行采样的理想分布，即$\mathbf{f}_{ideal}(v|\emptyset)$。随后，我们测量$\mathbf{f}_{ideal}$ 与其他几种分布之间的KL散度：NS/NS-prune的$\mathbf{f}_{uniform}$，以及AGIS的$\mathbf{f}_{approx}$。我们分析$\mathbf{f}_{approx}$的三个增量版本：(1) 仅使用$T[F_1],$；(2) 加入 $T[I_1],$；(3) 进一步纳入 $T[F_2],$，即AGIS所采用的完整版本。通过这种分解，我们可以观察每个新增项对近似准确度的影响。

对于所有模式，AGIS的$\mathbf{f}_{approx}$ 所达到的KL散度都比$\mathbf{f}_{uniform}$小几个数量级，这表明其对$\mathbf{f}_{ideal}$的近似程度高得多。我们还发现，三个项都是必需的。例如，在4-团中，仅使用第一项得到的KL散度略大于$\mathbf{f}_{uniform}$。加入项 $T[I_1],$后，该值降至接近0。同样，对于4-链，纳入最后一项 $T[F_2]$ 可进一步显著降低KL散度。

图8b展示了随着逐步添加各项，收敛所需的采样器数量。这里，我们纳入了更大的模式，对于这些模式，计算 $\mathbf{f}_{ideal}$ 不可行。由图可见，包含全部三项带来了明显的好处。然而，正如第4.2节所述，添加第三项带来的边际改善最小。这一观察符合这样一种观点：纳入高阶项会产生收益递减，因为模式中位置更靠后的边会引入更大的不确定性。

6.3 与基于MCMC的方法比较

基于马尔可夫链蒙特卡洛（MCMC）的计数技术 [14, 26, 38, 81] 为基于NS的采样方法提供了一种替代方案，可用于估计浓度，即给定图中某种特定模式在所有同尺寸模式中所占的比例。这些技术会构建一个高阶辅助图，其

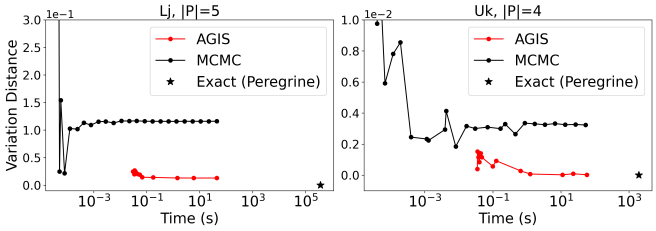
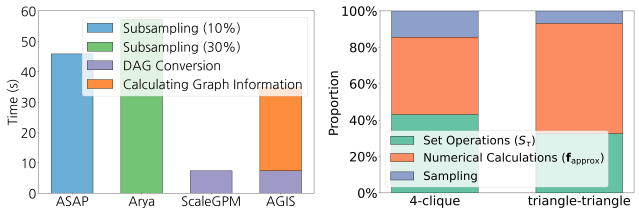


图9：与真实浓度之间的变差距离。

顶点对应于原图中的嵌入。如果两个辅助图顶点对应的嵌入仅通过小幅局部修改（例如替换、删除或添加一个顶点）而不同，则这两个顶点相邻。随后，MCMC算法会在该辅助图上执行随机游走，直至达到混合状态，并利用访问频率近似计算浓度。

图9比较了 AGIS 与一种基于 MCMC 的计数算法在 Lj图上的五阶模式和 Uk图上的四阶模式上的表现。我们实现了 SRW2，即 [26], 中描述的基于 MCMC 的算法，它可以返回指定模式阶数的估计浓度。对于 AGIS，共有 21 种连通的五阶非同构模式和 6 种连通的四阶非同构模式。我们逐一枚举每种模式以获得其近似计数，然后将这些计数除以相应阶数的模式总数，得到其浓度。真实浓度由精确系统 Peregrine 得出。准确度通过变差距离衡量，该距离定义为浓度向量之间 L_1 距离的一半。图中显示，尽管 MCMC 方法具有很强的理论保证，AGIS 在相同时间预算内仍能产生准确得多的估计。具体而言，在 1 秒时，MCMC 方法的变差距离分别比 AGIS 大 8.3× 倍和 41.7× 倍。



(a) Tw 预处理时间。 **(b) Tw 挖掘时间细分。**

图 10：执行时间分析。

图	Lj	Uk	Tw	Fs
总时间（秒）	0.477	3.069	34.93	55.37

表 6：不同图数据集的预处理时间。

6.4 执行时间分解

现在我们来考察预处理所花费的时间。对于 ASAP 和 Arya，每个给定的图都必须执行一次子采样步骤。该步骤包括按指定比率随机采样边，将其转换为无向图表示，然后将其进一步转换为 CSR格式。对于 ScaleGPM，则必须构建有向无环图（DAG），以进行定向优化。对于 AGIS，除了 DAG构建之外，我们还必须计算所需的图信息（第4.4节）。

图10a展示了四个 AGPM 系统的预处理时间。通过采用第4.4节所述的采样技术，AGIS 的预处理阶段并不会慢得难以接受，其性能仍与基于ELP的方法相当。为完整起见，表6报告了每个图的总预处理时间。这项一次性开销可在所有模式查询中摊销，因此在实践中可以忽略不计，尤其是在挖掘复杂模式或在严格的误差界下进行挖掘时。

最后，我们在图10b中详细分解了 AGIS 的运行时间。我们将执行划分为三个组成部分：(1) 用于构建候选集 S_r 的集合运算，(2) 用于计算 f_{approx} 的数值计算，以及 (3) 通过构建累积分布进行采样。分析表明，计算 f_{approx} 所需的时间略超过总运行时间的一半。然而，如第6.1节和第6.2节所示，这一额外开销是合理的，因为经过精心构建的f_{approx} 能够显著改善整体运行时间。

7 相关工作

精确图模式挖掘系统。人们已多次尝试开发高效的精确图模式挖掘系统。 [22, 25, 43, 51, 80], 等系统采用多种优化技术来提升性能。例如，Dwarves-Graph [22] 采用分解算法，并计算每个分解后模式的计数。GraphZero [51]采用重新标记技术，将定向算法推广到任意模式。Peregrine [43] 采用模式感知算法进行早期剪枝，从而绕过开销高昂的同构性检查和规范性检查。[23, 60]开发了一种组合框架，以实现高效计数。值得注意的是， [52, 70] 采用了与 AGIS 近似逻辑相似的性能模型。它们使用一个全局常数（例如三角形总数）来近似集合交集的大小，而 AGIS 则利用局部的细粒度信息。**其他近似计数方案。** 除基于 NS 的方法和基于 MCMC 的方法外，人们还开发了几种用于近似子图计数的替代方案。其中一个例子是颜色编码（CC）方法 [5, 6, 20, 73, 89]。其核心思想是为每个顶点分配一种随机颜色，并关注彩色匹配，即模式中的所有顶点颜色各不相同。通过动态规划可以高效地完成这一步计数；多次重复颜色分配和计数过程，则可以逐步提高近似精度。

另一条研究路线专注于在应用精确计数过程之前对图进行稀疏化。多项研究提出了不同的稀疏化技术，以近似计算特定模式，包括三角形 [58, 76],团 [68], 5-环 [69], 和蝴蝶模式 [67]。类似地，ScaleGPM [8] 引入了“宽松模式”，当其主要策略预计会失败或耗时过长时，便利用稀疏化技术。

一些流算法 [2, 79] 通过为每条边分配与其对整体模式计数的边际贡献成比例的纳入概率，来维护边储存池。这与 AGIS 类似，在 AGIS 中，最优采样概率与每个顶点所关联的模式实例数量成正比。

不同图类型上的计数。 尽管本文研究的是同质图，但异构图和超图上的计数是另一条重要的研究路线。人们已经开发出类型感知计数技术，以高效处理异构图上的约束。这些技术包括利用代数关系来避免穷举的组合方法 [3, 36, 61, 62, 88], 、发现频繁类型化模式的判别式挖掘方法 [21, 91], 以及基于采样的估计器 [63, 71]。

对于超图和单纯复形，近期系统 [74, 87] 利用超边特征和并行性。越来越多的研究提出基于草图的近似方法，以大规模计数小型子超图 [18, 45, 55, 78]。近期研究则对基于随机游走的采样进行改进 [13, 46, 47, 64, 66]。

8 结论

本文介绍了 AGIS，这是一种快速近似图模式挖掘系统，采用了一种新颖的结构感知邻居采样技术。通过整合数据图和模式图的结构属性，AGIS 构建近似的理想采样分布，并自适应地决定何时应用该分布，以实现最佳效率。在多个不同数据集上的实验表明，AGIS 的性能比最先进的基线方法高出一个数量级以上（几何平均值为 28.5× ），能够快速收敛，并可扩展至拥有数百亿条边的图。

致谢

本工作得到 Institute of Information & Communications Technology Planning & Evaluation (IITP) 的资助（RS-2024-00395134、RS-2024-00347394、RS-2023-00256081、RS-2021II211343、RS-2025-00564840）。Jinho Lee 为通讯作者。

参考文献

[1] Charu C Aggarwal、Haixun Wang 等。2010。《图数据的管理与挖掘》。第 40 卷。Springer。[2] Nesreen K Ahmed、Nick Duffield、Theodore L Willke 和 Ryan A Rossi。2017。《从大规模图流中采样》。VLDB基金会会刊 10，11（2017）。[3] Nesreen K. Ahmed、Jennifer Neville、Ryan A. Rossi、Nick G. Duffield 和 Theodore L. Willke。2017。《图元分解：框架、算法与应用》。Knowledge and Information Systems 50，3（2017），689–722。https://doi.org/10 1007 10115 016 0965 5 。/s - - - Leman Akoglu、Hanghang Tong 和 Danai Koutra。2015。《基于图的异常检测与描述：综述》。数据挖掘与知识发现 29（2015），626–688。[5] Noga Alon、Phuong Dao、Iman Hajirasouliha、Fereydoun Hormozdiari 和 S Cenk Sahinalp。2008。《利用颜色编码进行生物分子网络模体计数与发现》。Bioinformatics 24，13（2008），i241–i249。[6] Noga Alon、Raphael Yuster 和 Uri Zwick。1995。《颜色编码》。J. ACM 42，4（1995），844–856。[7] Naomi A. Arnold、Peijie Zhong、Cheick Tidiane Ba、Ben Steer、Raul Mondragon、Felix Cuadrado、Renaud Lambiotte 和 Richard G. Clegg。2024。《从加密货币交易网络中的局部与全局时间模体挖掘中获得的见解与注意事项》。Scientific Reports 14，26569（2024）。https://doi.org/10.1038/s41598-024-75348-7[8] Anna Arpaci-Dusseau、Zixiang Zhou 和 Xuhao Chen。2025。《大规模场景下准确且快速的近似图模式挖掘》。VLDB基金会会刊 18，2（2025），93–107。[9] Sepehr Assadi、Michael Kapralov 和 Sanjeev Khanna。2018。《一种通过边采样计数任意子图的简单次线性时间算法》。arXiv 预印本 arXiv:1811.07780（2018）。[10] Lars Backstrom、Dan Huttenlocher、Jon Kleinberg 和 Xiangyang Lan。2006。《大型社交网络中的群体形成：成员关系、增长与演化》。载于 KDD。[11] Albert-László Barabási 和 Réka Albert。1999。《随机网络中标度的涌现》。《科学》286，5439（1999），509–512。[12] Albert-László Barabási、Réka Albert 和 Hawoong Jeong。1999。《无标度随机网络的平均场理论》。Physica A：Statistical Mechanics and its Applications 272，1-2（1999），173–187。[13] Hamid Beigy、Mohammad Mahini、Salman Qadami 和 Morteza Saghafian。2024。《单纯复形单纯形频率分布的近似》。arXiv:2402.16777 [cs.CG] https://arxiv.org/abs/2402.16777[14] Mansurul A Bhuiyan、Mahmudur Rahman、Mahmuda Rahman 和 Mohammad Al Hasan。2012。《Guise：面向大图分析的图元均匀采样》。载于 ICDM。[15] Paolo Boldi、Bruno Codenotti、Massimo Santini 和 Sebastiano Vigna。2004。《Ubicrawler：可扩展的完全分布式网络爬虫》。Software: Practice and Experience 34，8（2004），711–726。[16] Paolo Boldi、Marco Rosa、Massimo Santini 和 Sebastiano Vigna。2011。《分层标签传播：一种用于压缩社交网络的多分辨率无坐标排序》。载于 WWW。[17] Paolo Boldi 和 Sebastiano Vigna。2004。《WebGraph框架 I：压缩技术》。载于 WWW。[18] Marco Bressan、Julian Brinkmann、Holger Dell、Marc Roth 和 Philip Wellnitz。2025。《计数小型子超图的复杂性》。arXiv:2506.14081 [cs.CC] https://arxiv.org/abs/2506.14081[19] Marco Bressan、Flavio Chierichetti、Ravi Kumar、Stefano Leucci 和 Alessandro Panconesi。2018。《五个节点以上的模体计数》。Transactions on Knowledge Discovery from Data 12，4（2018），1–25。[20] Venkatesan T Chakaravarthy、Michael Kapralov、Prakash Murali、Fabrizio Petrini、Xinyu Que、Yogish Sabharwal 和 Baruch Schieber。2016。《子图计数：超越树的颜色编码》。载于 IPDPS。[21] Yuan Fang；Wenqing Lin；Vincent W. Zheng；Min Wu；Jiaqi Shi；Kevin Chen-Chuan Chang 和 Xiao-Li Li。2021。《异构图上的基于元图的学习》。IEEE Transactions on Knowledge and Data Engineering 33，1（2021），154–168。[22] Jingji Chen 和 Xuehai Qian。2020。《Dwarvesgraph：一种采用模式分解的高性能图挖掘系统》。arXiv 预印本 arXiv:2008.09682（2020）。[23] Jingji Chen 和 Xuehai Qian。2022。《Decomine：一种采用模式分解的基于编译的图模式挖掘系统》。载于第 28 届 ACM 编程语言与操作系统架构支持国际会议论文集，第 1 卷。47–61。[24] Xuhao Chen 等。2022。《GPU上的高效且可扩展的图模式挖掘》。载于 OSDI。[25] Xuhao Chen、Roshan Dathathri、Gurbinder Gill、Loc Hoang 和 Keshav Pingali。2021。《Sandslash：用于高效图模式挖掘的两级框架》。载于 ICS。[26] Xiaowei Chen、Yongkun Li、Pinghui Wang 和 John Lui。2016。《一种通过随机游走估计图元统计量的通用框架》。arXiv 预印本 arXiv:1603.07504（2016）。[27] Yuhang Chen、Jiaxin Jiang、Shixuan Sun、Bingsheng He 和 Min Chen。2024。RUSH：动态图中的实时突发子图检测。VLDB基金会会刊 17，11（2024），3657–3665。[28] Young-Rae Cho 和 Aidong Zhang。2009。通过在蛋白质相互作用网络中挖掘频繁功能关联模式来预测蛋白质功能。《生物医学信息技术汇刊》14，1（2009），30–36。[29] Sutanay Choudhury、Lawrence Holder、George Chin、Khushbu Agarwal 和 John Feo。2015。一种基于选择率的方法，用于流图中的连续模式检测。arXiv 预印本 arXiv:1503.00849（2015）。[30] Aaron Clauset、Cosma Rohilla Shalizi 和 Mark EJ Newman。2009。经验数据中的幂律分布。SIAM Review 51，4（2009），661–703。[31] Mukund Deshpande、Michihiro Kuramochi、Nikil Wale 和 George Karypis。2005。用于化合物分类的基于频繁子结构的方法。《知识与数据工程汇刊》17，8（2005），1036–1050。[32] Vinicius Dias、Carlos HC Teixeira、Dorgival Guedes、Wagner Meira 和 Srinivasan Parthasarathy。2019。Fractal：通用图模式挖掘系统。载于 SIGMOD。1357–1374。[33] Reinhard Diestel。2024。图论。Springer（印刷版）；Reinhard Diestel（电子书）。[34] David Easley、Jon Kleinberg 等。2010。《网络、群体与市场：对高度互联世界的推理》。第1卷。剑桥大学出版社，剑桥。[35] Mark S Granovetter。1973。《弱关系的力量》。《美国社会学杂志》78，6（1973），1360–1380。[36] Shawn Gu、John Johnson、Fazle E. Faisal 和 Tijana Milenković。2018。从同质网络对齐到异构网络对齐：基于彩色图元的方法。Scientific Reports 8，12524（2018）。[37] Wentian Guo、Yuchen Li、Mo Sha、Bingsheng He、Xiaokui Xiao 和 Kian-Lee Tan。2020。分区图上的GPU加速子图枚举。载于 SIGMOD。[38] Guyue Han 和 Harish Sethu。2016。摇摆随机游走：快速且精确地挖掘大图中的模体统计信息。载于 ICDM。[39] Tomaž Hočevar 和 Janez Demšar。2014。一种图元计数的组合方法。Bioinformatics 30，4（2014），559–565。[40] Paul W Holland 和 Samuel Leinhardt。1976。社交网络中的局部结构。《社会学方法论》7（1976），1–45。[41] Yang Hu、Hang Liu 和 H Howie Huang。2018。TriCore：GPU上的并行三角形计数。载于 SC。[42] Anand Padmanabha Iyer 、 Zaoxing Liu 、 Xin Jin 、 Shivaram Venkataraman、Vladimir Braverman 和 Ion Stoica。2018。ASAP：大规模快速近似图模式挖掘。载于 OSDI 18。[43] Kasra Jamshidi、Rakesh Mahadasa 和 Keval Vora。2020。Peregrine：一种支持模式感知的图挖掘系统。载于 EuroSys。[44] Madhav Jha、C Seshadhri 和 Ali Pinar。2015。路径采样：一种快速且可证明有效的四顶点子图计数估计方法。载于 WWW。[45] John Kallaugher、Michael Kapralov 和 Eric Price。2018。图与超图计数的草图复杂度。载于 2018 IEEE第59届年度计算机科学基础研讨会（FOCS）。556–567。https://doi.org/10.1109/FOCS.2018.00059[46] Hyunju Kim、Jihoon Ko、Fanchen Bu 和 Kijung Shin。2023。通过计数四个节点以上的单纯元来刻画单纯复形。载于 ACM网络会议 *2023* 317 327 10 1145 3543507 3583332 。–。2025 https://doi.org/。/。Hyunju Kim、Heechan Moon、Fanchen Bu、Jihoon Ko 和 Kijung Shin。单纯元计数的采样估计。《VLDB期刊》34，2（2025），15。[48] Haewoon Kwak、Changhyun Lee、Hosung Park 和 Sue Moon。2010。Twitter是什么：社交网络还是新闻媒体？载于 WWW。[49] Jure Leskovec 和 Andrej Krevl。2014。SNAP数据集：斯坦福大型网络数据集合集。[50] Dean Lusher、Johan Koskinen 和 Garry Robins。2013。社交网络的指数随机图模型：理论、方法与应用。剑桥大学出版社。[51] Daniel Mawhirter、Sam Reinehr、Connor Holmes、Tongping Liu 和 Bo Wu。2021。GraphZero：高性能子图匹配系统。《ACM SIGOPS操作系统评论》55，1（2021），21–37。[52] Daniel Mawhirter 和 Bo Wu。2019。Automine：兼顾高层抽象与高性能的图挖掘。载于 SOSP。[53] Tijana Milenković、Weng Leong Ng、Wayne Hayes 和 Nataša Pržulj。2010。利用图元度向量实现最优网络对齐。《癌症信息学》9（2010），CIN–S4744。[54] Ron Milo、Shai Shen-Orr、Shalev Itzkovitz、Nadav Kashtan、Dmitri Chklovskii 和 Uri Alon。2002。网络模体：复杂网络的简单构件。《科学》298，5594（2002），824–827。[55] Richard Montgomery 和 Matías Pavez-Signé。2024。稠密超图中的生成子图计数。《组合学、概率与计算》33，6（2024 729 741 10 1017 0963548324000178 ），–。[56] https://doi.org/。/S Mark EJ Newman。2001。聚类与增长网络中的优先连接。《物理评论E辑》64，2（2001），025102。

[57] Mark EJ Newman。2003。复杂网络的结构与功能。*SIAM Review* 45，2（2003），167–256。[58] Rasmus Pagh 和 Charalampos E Tsourakakis。2012。彩色三角形计数及其MapReduce实现。信息处理快报 112，7（2012），277–281。[59] Aduri Pavan、Kanat Tangwongsan、Srikanta Tirthapura 和 Kun-Lung Wu。2013。图流中的三角形计数与采样。*VLDB*基金会会刊 6，14（2013），1870–1881。[60] Ali Pinar、Comandur Seshadhri 和 Vaidyanathan Vishal。2017。Escape：高效统计所有5顶点子图。载于第*26*届国际万维网会议论文集。1431–1440。[61] Ryan A. Rossi、Nesreen K. Ahmed、Aldo Carranza、David Arbour、Anup Rao、Sungchul Kim 和 Eunyee Koh。2019。异构网络模体。arXiv: 1901 10026 [ ] 1901 10026 。cs.SI [62] https://arxiv.org/abs/。Ryan A. Rossi、Nesreen K. Ahmed、Aldo Carranza、David Arbour、Anup Rao、Sungchul Kim 和 Eunyee Koh。2020。异构图小型结构。*ACM*知识发现数据汇刊（*TKDD*）15，1，文章9（2020），43页。https://doi.org/ 10 1145 3418773 。/ Ryan A. Rossi、Anup Rao、Tung Mai 和 [63] Nesreen K. Ahmed。2020。快速且精确地估计带类型图元。载于万维网会议伴随论文集 *2020* 32 34 10 1145 3366424 338268 。–。 [64] 2020 https://doi.org/。/。Marc Roth 和 Johannes Schmitt。诱导子图计数：一种针对# W[1]-难问题的拓扑方法。*Algorithmica* 82，8（2020），2267–2291。https://doi.org/10.1007/s00453-020-00676-9[65] Siddhartha Sahu、Amine Mhedhbi、Semih Salihoglu、Jimmy Lin 和 M Tamer Özsu。2017。大图的普遍性及图处理面临的意外挑战。*VLDB*基金会会刊 11，4（2017），420–431。[66] Gennady Samorodnitsky 和 Takashi Owada。2023。多参数单纯复形中子复形计数与贝蒂数的大偏差。*Random Structures & Algorithms* 63, 2 2023 533 556 10 1002 21146 （），–。 [67] 2018 https://doi.org/。/rsa。Seyed-Vahid Sanei-Mehri、Ahmet Erdem Sariyuce 和 Srikanta Tirthapura。二部网络中的蝴蝶计数。载于*KDD*。2150–2159。[68] Jessica Shi、Laxman Dhulipala 和 Julian Shun。2021。并行团计数与剥离算法。载于*ACDA*。[69] Jessica Shi、Louisa Ruixue Huang 和 Julian Shun。2022。并行五环计数算法。*Journal of Experimental Algorithmics* 27（2022），1–23。[70] Tianhui Shi、Mingshu Zhai、Yi Xu 和 Jidong Zhai。2020。Graphpi：通过有效消除冗余实现高性能图模式匹配。载于*SC*。[71] Wonseok Shin、Siwoo Song、Kunsoo Park 和 Wook-Shin Han。2024。子图匹配的基数估计：一种过滤-抽样方法。*VLDB*基金会会刊 17, 7 2024 1697 1709 2309 15433 （），–。 [72] 2015 https://arxiv.org/abs/。Julian Shun 和 Kanat Tangwongsan。多核三角形计算，无需调优。载于*ICDE*。[73] George M Slota 和 Kamesh Madduri。2013。快速近似子图计数与枚举。载于*ICPP*。[74] Yuhang Su、Yu Gu、Zhigang Wang、Ying Zhang、Jianbin Qin 和 Ge Yu。2023。基于超边特征的高效子超图匹配。*IEEE*知识与数据工程汇刊 35，6（2023），5808–5822。https: //doi.org/10.1109/TKDE.2022.3160393 [75] Carlos HC Teixeira, Alexandre J Fonseca, Marco Serafini, Georgos Siganos, Mohammed J Zaki 和 Ashraf Aboulnaga。2015。Arabesque：一个用于分布式图挖掘的系统。发表于 *SOSP*。[76] Charalampos E Tsourakakis、U Kang、Gary L Miller 和 Christos Faloutsos。2009。Doulion：使用硬币对海量图进行三角形计数。发表于 *KDD*。[77] Chad Voegele、Yi-Shan Lu、Sreepathi Pai 和 Keshav Pingali。2017。使用以图为中心的方法进行并行三角形计数和 k-悬挂识别。发表于 *HPEC*。[78] Jozef Skokan Vojt˘ech Rödl。2005。准随机四均匀超图中的子图计数。*Random Structures & Algorithms* 26, 1-2 (2005), 160–203。https: //doi.org/ 10 1017 0963548324000178 。/S Kaixin Wang、Cheng [79] Long、Da Yan、Jie Zhang 和 HV Jagadish。2023。强化学习增强的加权采样：在完全动态图流中实现精确子图计数。发表于*2023 IEEE39th InternationalConference on Data Engineering (ICDE)*。IEEE，1084–1097。[80] Kai Wang、Zhiqiang Zuo、John Thorpe、Tien Quang Nguyen 和 Guoqing Harry Xu。2018。RStream：将关系代数与流处理结合，在单机上实现高效图挖掘。发表于 *OSDI*。[81] Pinghui Wang、John CS Lui、Bruno Ribeiro、Don Towsley、Junzhou Zhao 和 Xiaohong Guan。2014。高效估计大型网络的模体统计数据。*Transactions on Knowledge Discovery from Data* 9, 2 (2014), 1–27。[82] Duncan J Watts。1999。小世界：介于有序与随机之间的网络动力学。[83] Duncan J Watts 和 Steven H Strogatz。1998。“小世界”网络的集体动力学。《自然》393, 6684 (1998), 440–442。[84] Xifeng Yan 和 Jiawei Han。2002。gspan：基于图的子结构模式挖掘。发表于 *ICDM*。[85] Carl Yang、Mengxiong Liu、Vincent W Zheng 和 Jiawei Han。2018。节点、模体与子图：通过结构卷积利用网络功能模块。发表于 *ASONAM*。[86] Jaewon Yang 和 Jure Leskovec。2012。基于真实标注定义和评估网络社区。发表于 ***SIGKDD***数据语义挖掘研讨会。[87] Zhengyi Yang、Wenjie Zhang、Xuemin Lin、Ying Zhang 和 Shunyang Li。2023。HGMatch：一种基于超边的超图子图匹配方法。发表于 *2023 IEEE 39th International Conference on Data Engineering (ICDE)*。 2063 2076 10 1109 55515 2023 00160 – 。 [88] https://doi.org/ 。/ICDE 。 。Shuo Yu、Feng Xia、Honglong Chen、Ivan Lee、Lianhua Chi 和 Hanghang Tong。2024。异构网络模体的编码、计数和特征分析。*ACM*数据知识发现汇刊18, 9 (2024), 文章 231。https: //doi.org/ 10 1145 3687465 。/ Zhao Zhao、Maleq Khan、VS Anil Kumar 和 [89] Madhav V Marathe。2010。使用并行颜色编码和流处理，在大型社会接触网络中枚举子图。发表于 *ICPP*。[90] Zeying Zhu、Kan Wu 和 Zaoxing Liu。2023。Arya：通过基于分解的采样进行任意图模式挖掘。发表于 *NSDI*。[91] Bo Zong、Xusheng Xiao、Zhichun Li、Zhenyu Wu、Zhiyun Qian、Xifeng Yan、Ambuj K. Singh 和 Guofei Jiang。2015。系统生成的时序图中的行为查询发现。*VLDB*基金会会刊9, 14 (2015), 240–251。https://doi.org/10.14778/2856318.2856320