arXiv:2611.28495v1 [cs.CG] 21 Nov 2026

A现实假设下的近似地图匹配 S

Lindsey Deryckere* Joachim Gudmundsson† André van Renssen‡ Sampson Wong§

摘要

给定一个图 $G = (V,\ E)$ 和一条复杂度为 n 、位于 $\mathbb{R}^d$ 中的多边形曲线 P，我们研究地图匹配问题，即计算 G 中一条到 P 的弗雷歇距离最小的路径。Chen、Driemel、Guibas、Nguyen 和 Wenk （ ALENEX 2011 ）针对 G 为 λ- 低密度且 P 为 c- 打包的情形，提出了一种近线性时间的 $(1+\varepsilon)$-近似算法。我们证明这些假设可以放宽，并在较弱条件下给出高效的 $(1+\varepsilon)$-近似算法。首先，若 P 为 c-打包，且 G 为任意图，我们可以得到一个运行时间为 $\tilde{O}(cn + c^2|E|)$ 的算法。其次，若 G 和 P 均为 λ-低密度，我们可以得到一个期望时间为 $\tilde{O}(\lambda^2((n|V|)^{1-1/d} + n + |V|))$ 的算法；在平面上，该算法的运行时间可降为 $\tilde{O}(\lambda^2(n + |V|))$。

关键词： 弗雷歇距离，地图匹配

1 引言

支持 GPS 的设备广泛普及，使得车辆轨迹数据得以持续采集，并广泛应用于导航系统、交通管理和城市规划。然而，原始 GPS 数据常受到测量噪声和低采样率的影响，导致轨迹偏离实际路线。地图匹配是一项基础的预处理任务，通过将记录的位置序列映射到表示道路网络的几何图上的路径，来解决这一问题。二十多年来，多个领域的研究者对地图匹配问题进行了广泛研究；参见综述 [8, 22–24, 26, 28]。形式上，地图匹配问题的输入包括：（i）一条轨迹，表示车辆随时间变化的位置；以及（ii）一个网络，表示车辆可能行驶过的道路。地图匹配问题的输出是道路网络上的一条路径，该路径与输入轨迹相似。见图 1。直观而言，地图匹配路径是对车辆实际行驶路径的“最佳猜测”。

在理论计算机科学中，解决地图匹配问题最常用的方法是：(i) 将轨迹建模为欧几里得平面中的多边形曲线；(ii) 将网络建模为欧几里得平面中的图；(iii) 使用某种弗雷歇距离变体来衡量地图匹配路径与输入轨迹的相似性 [1, 3, 5, 7, 9, 10, 19, 20, 27]。采用标准弗雷歇距离变体的优点是能够保证匹配路径

在空间上与输入轨迹相似。

*悉尼大学，澳大利亚， lindsey.deryckere@sydney.edu.au
†悉尼大学，澳大利亚， joachim.gudmundsson@sydney.edu.au
‡悉尼大学，澳大利亚， andre.vanrenssen@sydney.edu.au§哥本哈根大学，丹麦， sawo@di.ku.dk



图 1：一条　条带噪声的轨迹（蓝色）、道路网络（黑色）和地图匹配路径（绿色）　　.

问题 1（地图匹配）。给定一个连通图 G 和一条轨迹 P，计算一条路径 $\pi \in G$，使 d_F（π、P）最小，其中 d_F 表示弗雷歇距离。

请注意，根据此定义，一条路径可以多次经过任意边或顶点，但必须完整遍历边。换言之，不能只遍历边的一部分，且只能在图的顶点处改变方向。我们将问题 1 的最优值 $\min_\pi d_F(\pi, P)$ 记为 r^*。

本文　主要考虑下文所定义的问题的判定版本　　.

问题2（地图匹配——判定版本）。给定参数 $\delta \geq 0$，以及图 G 和位于 $\mathbb{R}^d$中的多边形曲线 P，是否存在一条路径 $\pi \in G$，使得 π 与P 之间的弗雷歇距离至多为 δ?

2003 年，Alt、Efrat、Rote 和 Wenk [1] 提出了问题 1，并给出了一种 $O(mn\log\ m)$ 时间算法，其中，$m = |E| + |V|$ 是图 $G = (V, E)$ 的复杂度，n 是曲线 P的复杂度。他们的主要贡献之一，是通过引入自由空间曲面，将自由空间图 [2] 扩展到地图匹配。

在正交向量假设成立的条件下，对于任意固定的 $\delta > 0$ [4,6,19]，不存在用于问题 1 的 $O((mn)^{1-\delta})$ 时间算法。

尽管问题 1 的复杂度已经确定，但针对现实输入图和曲线类别的地图匹配研究仍在持续。直觉上，通过采用能够反映车辆轨迹和道路网络现实属性的假设，可以规避 [4,6,19] 中不切实际的下界。这类方法在测量曲线相似度方面已被证明行之有效。一种现实输入模型是 c-紧致性。若对每个半径 r 和每个半径为 r 的球 B，其中 $B \cap P$ 内边的总长度至多为 $c\cdot r$，则边集 P 称为 c-紧致。Driemel、Har-Peled 和 Wenk [15] 表明，若两条输入曲线满足 c-紧致性，则可以使用简化技术，在近线性时间内近似计算它们的弗雷歇距离，即使是在高维空间中也是如此。他们考虑的另一种现实模型是 λ-低密度。若对每个半径 r 和每个半径为 r 的球 B，与 B 相交且长度为 $\geq r$ 的 G 中边的数量至多为 λ，则边集 G 称为 λ-低密度。注意，c-紧致性是比 λ-低密度严格更强的假设；任何 c-紧致曲线都是 $2c$-低密度的。尽管如此，Driemel、Har-Peled 和 Wenk [15] 表明，即使在高维空间中，仍然可以在次二次时间内近似计算两条 λ-低密度曲线之间的弗雷歇距离。Diatzko、Lasseux 和 Storandt [13] 最近确立了互补关系 $c \in O(\lambda\sqrt{n})$，并证明了对 n 的依赖是紧确的。

近似比	Dim	图 G	曲线 P	运行时间	参考文献
1	2	any	any	$\tilde{O}(\|E\| \cdot n)$	[1]
$1+\varepsilon$	d	λ-低密度	c-紧密排列	$\tilde{O}(\lambda n + c\|E\|)$	[9]
$1+\varepsilon$	d	any	c-紧密排列	$\tilde{O}(cn + c^2\|E\|)$	定理 4
$1+\varepsilon$	d	λ-低密度	λ-低密度	$\tilde{O}(\lambda^2((n\|V\|)^{1-1/d} + n + \|V\|))$	定理 18
$1+\varepsilon$	2	λ-低密度	λ-低密度	$\tilde{O}(\lambda^2(n + \|V\|))$	定理 18

表 1：问题 1 的已有结果及本文结果。这里，$|E|$ 和 $|V|$ 分别是图 G 的边数和顶点数，而 P 的复杂度为 n。$O(\cdot)$ 隐藏了对 ε 和多对数因子的依赖，并假设 d 为常数。另请参见第 1.2 节中的并行工作。

2011 年，Chen、Driemel、Guibas、Nguyen 和 Wenk [9] 结合 [15] 与 [1]的技术，在现实输入图和曲线上近似求解问题 1。具体而言，他们假设图 G 是 λ-低密度的，且曲线 P 是 c-紧致的。不同于 [15] 中对两条输入曲线都进行简化的做法，[9] 的作者证明，如果其中一条曲线是 λ-低密度的，那么为了在次二次时间内近似计算弗雷歇距离，只需对c-紧致的曲线进行简化。他们为问题 1 提供了一种近线性时间近似算法。具体来说，他们的 $(1+\varepsilon)$-近似算法的运行时间为 $O((m+n)\log(m+n) + (\lambda n + cm)\log(mn)\log(m+n) + (\lambda n/\varepsilon^d + cm/\varepsilon)\log mn)$。

Gudmundsson、Mai 和 Wong [18] 首次考虑了在仅有一条曲线为 c-packed 时，对两条多边形曲线之间的弗雷歇距离进行近似。他们获得了一种近线性时间近似算法，其运行时间为 $O(c^3(n+m)\log^{2d+1}(n)\log m)$。Conradi、van der Hoog、van der Horst 和 Ophelders [12] 将运行时间改进为 $O(c^{\frac{n+m}{\varepsilon}}\log\frac{n+m}{\varepsilon})$，其中 n 和 m 分别表示两条曲线的复杂度。

本文考虑以下问题。

问题3。在 *[9]*，中，这两个假设是否都是必要的，即 G 是 λ-*low-dense* 且 P 是 c-*packed*? 能否使用更宽松的假设来获得次二次时间的 $(1+\varepsilon)$-近似算法?

如果可以使用更宽松的假设，那么所得的地图匹配算法便可应用于更广泛的轨迹和道路网络。

1.1 贡献

我们证明，为问题 1 获得次二次时间的（$1+\varepsilon$）-近似算法，并不需要同时假设 G 是 λ-low-dense 且 P 是 c-packed，这一结果对问题3给出了肯定回答。对于问题2，当满足以下条件时，我们提供（$1+\varepsilon$）-近似判定器：

(1) 图 G 是任意一般图，曲线 P 是 c-打包的。特别地，我们的判定器的运行时间为 $O(cn + \frac{1}{\varepsilon^{2d}}\log^2\frac{1}{\varepsilon}\cdot n\log^2 n + |V|\frac{1}{\varepsilon^d}(c+\log n) + |E|\frac{c}{\varepsilon^3}\log\frac{c}{\varepsilon}\cdot\log n\log\log n + |E|\frac{c^2}{\varepsilon^2})$。

(2) 图 G 是 λ-低密度的，曲线 P 是 λ-低密度的。特别地，经过期望为 $O(\lambda|V|\log|V|)$ 的预处理时间后，我们的判定器可在 $O\left(\lambda^2\left((n|V|)_{\varepsilon^2}^{1-1/d} + \frac{n+|V|}{\varepsilon^d}\right)\log(n|E|)\right)$ 时间内运行。当 $d=2$ 时，运行时间为 $O\left(\frac{\lambda^2}{\varepsilon^2}(n+|V|)\log(n|E|)\right)$。

如果 ε 和 d 是常数，并且忽略 n 和 E中的多对数因子，那么运行时间分别变为 $\tilde{O}(cn + c^2|E|)$ 和 $\tilde{O}(\lambda^2((n|V|)^{1-1/d} + n + |V|))$。参见表 1。使用这些

判定器，我们可以针对问题 1 获得 $(1+\varepsilon)$-近似算法，其运行时间相同（相差对数因子）（第 6 节）。

1.2 并行工作

在一篇近期的预印本中，Conradi、van der Hoog、Uldahl 和 Rotenberg [11] 独立研究了当 G 和 P 均为 λ-低密度时的问题 1。对于具有 n 个顶点的曲线和具有 n条边的图，他们给出了一种（$1+\varepsilon$）-近似算法，其运行时间为 $\tilde{O}(\lambda^{2/d}n^{2-2/d})$，并证明在 $d \geq 3$ 且满足正交向量假设的条件下，对于任意 $\delta > 0$，不存在运行时间为 $\tilde{O}((\lambda^{2/d}n^{2-2/d})^{1-\delta})$ 的算法。

1.3 其他相关工作

基于弗雷歇距离的地图匹配算法已得到广泛研究。然而，能否将图预处理为一种数据结构，从而支持亚线性时间地图匹配查询，是较新的研究方向。在正交向量假设下，不存在能够在多项式时间内构建的数据结构，使其可以针对几何平面图处理查询，并且相对于输入结构的复杂度 [19]在严格次二次时间内完成。因此，研究兴趣已转向在输入图上应用现实结构假设，以执行近似查询。

Gudmundsson、Seybold 和 Wong [19] 首次提出了用于弗雷歇距离下地图匹配的查询数据结构，并假设图是 c-打包的。在这一假设下，他们的数据结构占用 $\tilde{O}(cm)$空间，并能在 $O(c^4n\log^4 m)$时间内回答$(1+\varepsilon)$-近似判定查询，其中 n 是曲线的复杂度， $m = |V| + |E|$ 是 c-打包图的复杂度。

随后，Buchin、Buchin、Gudmundsson、Popov 和 Wong [5] 将地图匹配查询扩展到 λ-低密度 t-伸展图；这类图旨在更好地刻画现实道路网络的性质。他们的数据结构占用 $\tilde{O}(\lambda m^{3/2})$空间，并能在 $\tilde{O}((\lambda+t^2)^4 \cdot \lambda^2 \cdot n\sqrt{m})$时间内回答近似弗雷歇距离查询。该数据结构还会报告一条路径，该路径对弗雷歇距离的近似比为$(1+\varepsilon)$。

2 技术概述

我们简要概述所采用的技术，并着重说明本文与以往研究在技术上的差异。

2.1 基于 c-packedness 的地图匹配

我们首先概述判定过程，然后介绍如何通过 [15]，利用曲线简化和 c-packedness 加速近似弗雷歇距离计算，从而应用该框架。

近似判定过程概述。给定图 G 、多边形曲线P和 $\mathbb{R}^d$中的查询值 $\delta \geq 0$，我们构造一个搜索结构 $H(U, F)$，其节点集合U 对应于 P上的点（不一定是顶点），其有向弧集合 F 表示将 G 中的边 e 映射到 P 的子轨迹的单调映射，并满足以下两个条件：

1. 如果 H 中存在一条路径，其起点是对应于 p_1 的节点，终点是对应于 p_n的节点，那么就存在一条路径 $Q \subseteq G$，使得 $d_F(Q, P) \leq (1+\varepsilon)\delta$。

2. 如果 H 中不存在这样的路径，那么对于 G 中的所有路径 Q， $d_F(Q,\ P) > \delta$。

换言之，我们将近似判定问题归约为在 H 中寻找一条路径。

我们分两个阶段构造 H。在阶段 1 中，我们构造 H 的节点。

每个节点都对应简化曲线 P' 的一个样本。该样本是沿 P' 的一维网格上的一个点，网格间距为 $\varepsilon'\delta$；它位于 P' 的某条子轨迹上，且与 G 的某个顶点之间的距离小于 $r = (1+2\varepsilon')\delta$。由于匹配可以从 P 上的任意点开始和结束，因此节点不能仅限于 P 的顶点。我们在 O 时间内对 P' 进行预处理（$n(\log\ n + c)$），将其构造成一棵（下推式）压缩四叉树，从而能够针对 G 的每个顶点，高效查询 P' 中与该顶点距离小于 r 的子轨迹。

在阶段 2 中，我们构建 H 的有向弧。对于每条边 $e \in G$，我们计算 P 的哪些子轨迹可以在弗雷歇距离 $(1+\varepsilon)\,\delta$ 内单调映射到 e。对于每条子轨迹，如果它可以在弗雷歇距离 $(1+\varepsilon)\,\delta$ 内映射到该边，我们就在 H 中与该子轨迹的样本相对应的节点之间添加一条有向弧。在此阶段，我们利用 [14] 中介绍的技术，对曲线 P 进行预处理，将其构建为一种数据结构，从而支持高效查询曲线的子轨迹与 G 的边之间的弗雷歇距离。由于多边形曲线的性质以及弗雷歇距离映射的单调性，得到的结构 H 是一个有向图。

最后，我们在 H 中搜索一条路径，该路径连接 H 中与 p_1 对应的节点和与 p_n 对应的节点。这样的路径表示 G 中的一条路径与整条曲线 P 之间的单调映射，并且满足弗雷歇距离和近似值要求。

通过曲线简化加速近似查询。我们采用 [15] 的框架，通过曲线简化和 c-紧致性来加速弗雷歇距离的近似计算。具体而言，[15] 的作者证明，可以在线性时间内按照参数 μ 简化一条 c-紧致曲线 P，使所得曲线 P' 满足：

- $P\prime$ 除最后一条线段外，其余线段的长度均至少为 μ。
- P 与 $P\prime$ 之间的弗雷歇距离至多为 μ。
- $P\prime$ 是 $6c$-紧致的。 [[15]] 中的引理 4.3

上述对 c-紧致曲线进行简化所得到的性质，使我们能够对距图中某个顶点不超过 δ 的 P' 的子轨迹数量给出上界，从而限制搜索结构的复杂度。因此，在计算搜索结构之前，我们首先按照查询值 δ 和近似值 ε 简化 c-紧致曲线 P。随后，我们针对 P' 而非 P 构建搜索结构 H，以降低搜索结构的复杂度，进而提高运行效率。

与先前工作的关系。虽然我们的方法建立在 [18] 中提出的用于近似两条曲线之间弗雷歇距离的框架之上，但我们调整并优化了其辅助图构造，以适应地图匹配（即，将一条 c-紧致曲线与一般图进行匹配，而不是与一般曲线进行匹配）。[18] 的作者计算一条 c-紧致曲线 P（具有 n 个顶点）与一条一般曲线 Q（具有 m

个顶点）之间的近似弗雷歇距离。相比之下，我们的输入是一条 c-紧致曲线与一个一般图 $G = (V, E)$ 组成的匹配对。我们的方法存在以下差异。

阶段 *1*（顶点构造）：我们不再使用 [25]，的范围搜索数据结构为辅助图采样 $O(nc/\varepsilon^2)$ 个顶点，而是改造 [17] 的作者所采用的数据结构。这使 H 中的顶点总数减少至 $O(|V|c/\varepsilon)$，从而缩短了顶点构造时间。

阶段 *2*（边构造）：我们将对 [14] 的数据结构的查询次数减少，并将边的总数从 $O(mc^2/\varepsilon^4)$ 降至 $O(|E|c/\varepsilon)$。因此，我们缩短了 H 中边的构造时间。

图结构与搜索：由于输入是一般图，生成的辅助图不再具有 [18] 中的分层有向无环图结构，而成为普通有向图。因此，我们将他们的标准广度优先搜索替换为经过修改的深度优先搜索，使总体运行时间达到 $\tilde{O}(cn + c^2|E|)$。

我们得到以下定理，其证明见第 4 节。

定理 *4.* 设 P 是复杂度为 n 的 c-打包多边形曲线，$G = (V, E)$ 是空间 $\mathbb{R}^d$ 中复杂度为 $|V| + |E|$ 的任意图。给定查询值 $\delta > 0$ 和近似值 $0 < \varepsilon < 1$，存在一个判定算法，使得若其返回是，则存在一条路径 Q 属于 G，满足 $d_F(P, Q) \leq (1 + \varepsilon)\delta$；若其返回否，则对于 G 中的所有路径 Q，都有 $d_F(P, Q) > \delta$。该判定器的运行时间为 $O(cn + \frac{1}{\varepsilon^{2d}} \log^2 \frac{1}{\varepsilon} \cdot n \log^2 n + |V| \cdot \frac{1}{\varepsilon^d}(c + \log n) + |E| \frac{c}{\varepsilon^3} \log \frac{c}{\varepsilon} \cdot \log n \log \log n + |E| \frac{c^2}{\varepsilon^2})$，使用空间 $O(cn + |E| \frac{c}{\varepsilon} + (\frac{1}{\varepsilon^{2d}} \log^2 \frac{1}{\varepsilon})n)$。

2.2 使用 λ-低密度的地图匹配

我们的方法建立在 [15] 提出的框架之上，该框架用于界定两条 λ-低密度曲线之间的近似自由空间复杂度；同时结合了 [9] 提出的地图匹配近似判定框架（该框架以 [15] 和 [1] 为基础）。

[15] 的作者提出了一种曲线简化算法，可确保输入曲线 P 具备两个基本性质：

1. 除最后一条边外，其余边的长度均不小于某个最小边长。
2. P 与简化曲线 P' 之间的弗雷歇距离相对于该最小边长有界。

尽管这种简化能以不超过 6 倍的因子保留 c-紧密性，但它无法以较小的因子同样保留 λ-低密度性质。不过，[15] 的作者观察到，如果简化后的 λ-低密度曲线有许多边与某个给定区域重叠（从而形成高密度区域），那么原曲线附近必定有大量顶点，才能满足原曲线的低密度性质。他们利用这一观察，将对自由空间复杂度有贡献的简化曲线边数收费到图的原始顶点上，并确保每个顶点被收费的次数仅为次线性。因此，他们证明，简化曲线之间的自由空间复杂度相对于原始曲线的复杂度为次二次。[15] 的作者所采用的收费论证要求两条曲线都经过简化，因为该论证使用了双向收费方案。

将这一方法扩展到 λ-低密度图面临着若干挑战。标准图聚类算法可能会违反简化的某些性质，也可能显著提高局部图密度，却无法保证原图附近存在许多顶点。主要挑战在于生成一种简化，使其复杂度相对于

原图较低，同时保持图的原有连通性，并确保与原图之间的弗雷歇距离较小。

我们提出了一种两阶段图简化方案。首先，我们使用 r-网对图进行聚类，在保持有界弗雷歇误差的同时，确保顶点在原始图度量下具有最小图距离。然而，简化图可能会显著增加图的局部密度。由于图需要维持比曲线更多的连通性，其密度增幅可能远高于曲线。因此，我们采用第二阶段简化，其中包含一个剪枝步骤。该步骤会降低图的密度，使我们能够界定与任意区域重叠的简化边数量（区域大小与查询和近似因子相关），从而无需将简化图的局部密度与附近原图顶点的数量联系起来。在此步骤中，我们继续保持有界的弗雷歇距离误差。随后，我们结合上述技术以及与 [15], 中类似的论证，界定简化结构之间的自由空间复杂度，并将其与原始结构之间的近似自由空间联系起来。

最后，我们应用 [9] 的框架，利用结构之间的近似自由空间来设计高效的近似判定器。

2.3 从判定器到近似算法

遵循 Driemel、Har-Peled 和 Wenk [15, 第 3], 节的方法，我们将两个判定器都转化为问题 1 的 $(1+\varepsilon)$-近似算法，其运行时间仅相差对数因子；参见第 6 节。

3 预备知识

3.1 输入模型

我们定义两个与现实图和曲线相关的性质。

定义 5 （ λ-低密度边集）。 如果对于任意半径为 r、以任意点 $p \in \mathbb{R}^d$ 为中心的球 $B(p,r)$，与之相交的 G 中长度至少为 r 的边数至多为 λ，则称 G 中的边集 G 在 $\mathbb{R}^d$ 中是λ-低密度的。

定义 6 （ c-紧密曲线）。 如果对于任意半径为 $r > 0$、以任意点 $p \in \mathbb{R}^d$ 为中心的球$B(p,r)$，曲线与球重叠的长度，即 $|B \cap P|$，至多为 $c \cdot r$，则称 $\mathbb{R}^d$ 中的多边形曲线 P 是 c-紧密的。

3.2 Fréchet距离

多边形曲线 P由点的有序集合 $\{p_1, ..., p_n\} \in \mathbb{R}^d$定义，称为 P的顶点。P是一个分段线性函数 P: $[1,n] \to \mathbb{R}^d$，其中对于任意 $t \in [i, i+1]$， $P(t)$定义为 p_i和 p_{i+1}的线性插值。换句话说， $P(t) = p_i + (t-i)(p_{i+1}-p_i)$。 $[1,n]$ 的重新参数化是一个非递减满射 f: $[0,1] \to [1,n]$。分别对 $[1,n]$和 $[1,m]$ 进行重新参数化的两个重新参数化 f和 g，描述了曲线 P和 Q之间的一个匹配 (f,g)，其中 P和 Q分别具有 n和 m个顶点；对于任意 $t \in [0,1]$， 点 $P(f(t))$ 与 $Q(g(t))$ 相匹配。两条多边形曲线 P和 Q之间的连续静态弗雷歇距离 $d_F(P,Q)$ 定义为

$$d_F(P,Q) := \min_{(f,g)} \max_{t \in [0,1]} ||P(f(t)) - Q(g(t))||,$$

其中 $(f,\ g)$ 遍历 P 和 Q之间的所有有效匹配。在全文中，我们将 $d_F(P,\ Q)$ 记作 d_F。

设 $G=(V,\ E)$ 是嵌入在 $\mathbb{R}^d$ 中的无向图。设 $V=\{v_1,\ \cdots v_{|V|}\}\subseteq\mathbb{R}^d$ 为 G 的顶点。尽管 G 是无向图，我们仍将每条无向边 $(u,\ v)\in E$ 视为由两条有向边 $(u,\ v)$，$(v,\ u)\in E$ 表示。因此，E 中的边是有向的，但表示一个无向图。我们将图中的路径 π 定义为由图的边构成的多边形曲线。请注意，我们对路径的定义并未限制 π 重复使用同一条边或同一个顶点。

如果将 Q 视为 G 中使其与 P 之间的弗雷歇距离最小的路径，那么曲线 P 与图 G 之间的弗雷歇距离，便可类似于两条曲线 P 和 Q 的情形来定义。可视化两条多边形曲线 P 和Q 的可能重新参数化，一种常见方法是使用自由空间图。自由空间图（FD_δ）是一种常用的几何对象，用于判断两条多边形曲线 $P,\ Q$ 之间的弗雷歇距离是否至多为给定参数 $\delta\geq 0$。这一概念最早由 [2] 提出。

定义 7（自由空间）。（引自 [2]）设 $P\colon [0,1]\to\mathbb{R}^d$, $Q\colon [0,1]\to\mathbb{R}^d$ 是两条曲线。集合 $F_\delta(P,\ Q):=\{(s,\ t)\in[0,1]^2 \mid \|P(s)-Q(t)\|\leq\delta\}$ 表示 P 和 Q 的自由空间。我们将 $[0,1]^2$ 划分为属于 $F_\delta(P,\ Q)$ 的区域和不属于它的区域，这种划分称为 P 和 Q 的自由空间图（$\mathrm{FD}_\delta(P,\ Q)$）。

[2] 的作者证明，$d_F(P,\ Q)\leq\delta$ 当且仅当自由空间图中存在一条从 $(0,\ 0)$ 到 $(1,\ 1)$ 的路径，该路径在两个坐标上均单调，且完全包含于 $F_\delta(P,\ Q)$ 中。因此，只需在 FD_δ 中寻找一条单调路径，即可判断 P 和 Q 之间的弗雷歇距离是否至多为 δ。

[1] 的作者将这一概念扩展到了地图匹配场景。他们不再使用单个平面单位正方形 $[0,1]^2$ 作为自由空间图，而是为每条边 $e\in E$分别构造一个自由空间图 $[0,1]\times e$。沿着对应于 G 中共享顶点的边界，将这些独立的边图拼接起来，便得到一个连续且分支的拓扑空间，他们称之为自由空间曲面。

3.3 曲线简化

在 [15] 中，作者提出了一个框架，可以在现实输入模型下更高效地近似求解两条曲线之间的弗雷歇距离问题。具体做法是根据近似因子简化曲线，并对简化结构之间的自由空间进行界定。

事实 8 (算法 2.1，见 [15])。可以通过扫描曲线 P 来根据参数 μ 对其进行简化：从第一个顶点开始，每一步只选取与上一个已选顶点的距离至少为 μ 的下一个顶点。由此得到的曲线 P' 满足：

- $P\prime$ 除最后一条线段外，所有线段的长度至少为 μ。
- F 原始曲线与简化曲线之间的弗雷歇距离至多为 μ.

该算法的运行时间为线性时间。

4 具有 c-紧致性的地图匹配

本节针对多边形曲线 $P=\langle p_1,...,p_n\rangle$ 满足 c-紧致性的情况，为问题2给出一个 $(1+\varepsilon)$-近似判定器。我们不对输入图 $G=(V=v_1,\cdots v_{|V|},E)$作任何假设，并证明曲线的性质足以为该问题设计一个高效的近似判定器。本节将证明定理 4，下面重述该定理。

定理 *4.* 设 P 是一条复杂度为 n的 c-紧致多边形曲线，并设 $G=(V,E)$ 是一个空间为 $\mathbb{R}^d$、复杂度为 $|V|+|E|$ 的任意图。给定查询值 $\delta>0$和近似值 $0<\varepsilon<1$，存在一个判定算法，使得若其返回是，则存在一条路径 Q 位于 G 中，使得$d_F(P,Q)\leq(1+\varepsilon)\delta$；若其返回否，则对于 G中的所有路径 Q，均有 $d_F(P,Q)>\delta$。该判定器的运行时间为 $O(cn+\frac{1}{\varepsilon^{2d}}\log^2\frac{1}{\varepsilon}\cdot n\log^2 n+|V|\cdot\frac{1}{\varepsilon^d}(c+\log n)+|E|\frac{c}{\varepsilon^3}\log\frac{c}{\varepsilon}\cdot\log n\log\log n+|E|\frac{c^2}{\varepsilon^2})$，使用的空间为 $O(cn+|E|\frac{c}{\varepsilon}+(\frac{1}{\varepsilon^{2d}}\log^2\frac{1}{\varepsilon})n)$。

我们先介绍一些有用的记号。

设 $P=\langle p_1,\ldots,p_n\rangle$ 是 $\mathbb{R}^d$中的一条多边形曲线，其边总长度为 $L=\sum_{i=1}^{n-1}||p_{i+1}-p_i||$。在本节中，我们将 P视为一个连续映射 $P:[0,L]\to\mathbb{R}^d$，并以边长作为参数，其中 $P(0)=p_1$且 $P(L)=p_n$。对于满足 $s\leq t$的任意一对参数 $s,\ t\in[0,L]$，我们将沿 P、从点 $P(s)$开始并在点 $P(t)$结束的子轨迹记为 $P[s,t]$。注意，端点 $P(s)$和 $P(t)$不一定与 P的顶点重合。

近似判定过程概述 给定 $\mathbb{R}^d$中的图 G 和多边形曲线P，以及查询值 $\delta>0$ 和近似值 $0<\varepsilon<1$，我们构造一个有向图 $H(U,\ F)$，其节点集 U对应于 P中的点（不一定是顶点），其有向弧集 F表示将 G中的边 e映射到子轨迹 $P[s,t]\subseteq P$的单调映射，并满足以下两个条件：

1. 如果 H 中存在一条路径，其起点是对应于 p_1 的节点，终点是对应于 p_n的节点，那么存在一条路径 $Q\subseteq G$，使得 $d_F(Q,\ P)\leq(1+\varepsilon)\delta$。
2. 如果 H中不存在这样的路径，那么对于 G中的所有路径 Q，$d_F(Q,\ P)>\delta$。

我们分两个阶段构造 H。在阶段 1，我们构造 H的节点。每个节点对应于 P上的一个采样点，该采样点位于沿曲线间距为 $\varepsilon'\delta$ 的固定网格上，并且到 G的某个顶点的距离不超过$r:=(1+2\varepsilon')\delta$，其中 $\varepsilon'=\varepsilon/5$。在阶段 2，我们构造 H的有向弧。H 中的每条弧都表示将 P 的一个子轨迹单调映射到 G 的一条边，且弗雷歇距离至多为 $\tau=(1+\varepsilon')r$。

最后，我们在生成的图 H 中搜索一条有效路径，该路径连接p_1 对应的 H 中的一个节点与对应于 p_n的一个节点。这样的路径表示G 中的一条路径与整条曲线 P之间的单调映射，并满足弗雷歇距离和近似值的要求。

在第4.1节中，我们将介绍如何根据图 G、c-紧致曲线 P、查询值 δ和近似值 ε构造图 H。在第 4.2 节中，我们将说明如何使用 H 来回答判定问题。

4.1 构造有向图 H

在计算图之前，我们首先根据查询值 $\delta>0$ 和近似值 $0<\varepsilon<1$ 简化 c-紧密曲线 P。然后，我们根据简化曲线 P' 而非 P 构造有向图 H，以降低有向图的复杂度，从而提高运行效率。

我们使用事实 8 中描述的算法简化曲线 P，参数为 $\mu = \varepsilon'\delta$，其中 $\varepsilon' = \varepsilon/5$。所得的简化曲线记为 P'，并满足：

- P_I 除最后一条线段外，所有线段的长度都保证至少为 μ。
- P 与 P_I 之间的弗雷歇距离至多为 μ。
- P_I 是 $6c$-紧致的。 [[15]]中的引理 4.3

在本节余下部分，我们使用 P_I来构造图 $H = (U, F)$。为避免混淆，我们将 G 的顶点称为顶点，而将 H 的顶点称为节点。类似地，我们将 G 的边称为边，而将 H 的有向边称为有向弧。

阶段 1：构造 H 的节点。本节中，我们对 P' 进行预处理，以便能够针对 G 中的每个顶点，高效查询 P' 中靠近该顶点的子轨迹。这些子轨迹内的网格采样点将构成 H 的节点。此外，对于每个顶点 $v \in G$，我们定义其“δ-邻域”（或简称“邻域”），记作 $\mathcal{N}_v$，它是所有满足以下条件的极大连续子区间 $[s,t] \subseteq [0,L]$ 的集合： $P'[s,t]$ 完全包含在球 $B(v, r)$ 内，其中 $r = (1+2\varepsilon')\delta$。$\mathcal{N}_v$ 中的子轨迹按照其沿曲线 P' 从 p_1 到 p_n 的出现顺序存储。为了计算 G 中各顶点的邻域，我们首先计算 P' 中靠近每个顶点 $v \in G$ 的边。我们使用 [17] 提出的现有技术和结果，对 P' 进行预处理，从而能够高效计算此类边。利用这些边，可以轻松确定 G 中各顶点的邻域，随后我们用它们构造 H 的节点集合 U。

我们首先介绍如何将 P' 预处理为四叉树数据结构 T，以便高效查询：对于任意顶点 $v \in G$， P' 中哪些边位于距 v不超过 r 的范围内，其中 $r = (1+2\varepsilon')\delta$。

回顾事实 8， P'中的每条线段（可能的最后一条除外）长度至少为 $\varepsilon'\delta$。最后一条线段单独处理：在查询时，我们直接逐一检查它是否落在每个查询球内，每个顶点耗时 $O(1)$。因此，在后续构造中，我们假设 P_I中的每条线段长度都至少为 $\varepsilon'\delta$。令 S表示 P'中的线段集合，则 $|S| < n$。

预处理。 预处理分为三个步骤。

步骤 1（关联规范超立方体）。 如果一个超立方体的边长是 2 的幂，且它是该边长对应的层次网格（即一种多层网格，其单元格大小由较大的包围区域不断二分确定）中的一个单元，我们就称它为规范的 。设$\mathcal{B}$ 为包含 S的轴对齐包围盒，并设 $\mathcal{G}$ 为覆盖 $\mathcal{B}$的层次网格。对于每条线段 $s \in S$，令 $Q(s)$ 为以 s 的中点为中心且包含 s的最小轴平行超立方体；其边长记为 side($Q(s)$)，取值属于 $[|s|/\sqrt{d},\ |s|]$。利用 [17], 的引理 9，用 $O(1)$ 个 $\mathcal{G}$中的规范超立方体覆盖 $Q(s)$；这些超立方体的边长至少为 side($Q(s)$) $/2 \geq |s|/(2\sqrt{d})$，且至多为 2 side($Q(s)$) $\leq 2|s|$。我们称这些规范超立方体与 s相关联。令$\mathcal{C}$ 为所有线段对应的关联超立方体集合； $|\mathcal{C}| = O(n)$，且$\mathcal{C}$ 可在 $O(n)$ 时间内计算得到。

步骤 2（压缩四叉树）。给定 $\mathcal{B}$、$\mathcal{G}$ 和 $\mathcal{C}$，构建一个最小压缩四叉树 T，使得$\mathcal{C}$ 中的每个超立方体都是 T 中某个节点的区域；树 T 有 $O(n)$ 个节点，并可在 $O(n \log n)$ 时间内构建 [[21], 引理 2.11]。此外，我们使用来自 [21] 的定理 2.14，在 $O(n\log n)$ 时间内对 T进行预处理，以支持单元查询：给定一个规范超立方体 C，在 $O(\log n)$ 时间内返回区域包含 C的 T 中最低的节点；若不存在这样的节点，则报告不存在。

步骤 3（向下推送）。对于四叉树中的节点 u，我们用 R_u 表示 u 的超立方体（区域）。将 S 的线段按长度递增排序。 T 中的每个节点 u 都存储一个初始为空的线段有序列表 $L(u)$。我们按排序顺序处理这些线段。要处理 S 中的线段 s，

我们找到 T 中区域为与 s 关联的规范超立方体的节点（根据 T 的构造，这些节点必定存在），并从每个节点开始下降遍历：对于节点 u，如果 $s \cap R_u \neq \emptyset$，我们就将 s 前置到 $L(u)$，并递归访问 u 的所有子节点；否则停止。由于线段按长度递增的顺序处理，因此每个列表 $L(u)$ 中的线段长度都按递减顺序排列，如 [17] 中所示。

至此， T的构造完成。

回答查询。给定查询顶点 $v \in V$，令 $B = B(v,\ r)$。令 b 为满足 $b \le \varepsilon'\delta/(2\sqrt{d})$ 的最大 2 的幂，并考虑边长为 b 的规范网格。令 $\mathcal{K}_v$ 为该网格中与 B 相交的单元集合；则 $|\mathcal{K}_v| = O(((1+2\varepsilon')\delta/b)^d) = O(1/\varepsilon'^d)$，其中 d 为常数，且可在 $O(1/\varepsilon'^d)$ 时间内枚举 $\mathcal{K}_v$。

对于每个单元 $C \in \mathcal{K}_v$，我们对 T 执行单元查询，以在 $O(\log n)$ 时间内找到 T 中区域包含 C 的最低节点 u_C；如果不存在这样的节点，我们就将 C 丢弃（如下文所述， S 中没有线段与这样的单元相交）。然后，我们扫描列表 $L(u_C)$，并对其中的每条 $O(c)$ 线段，在 $O(1)$ 时间内检查其是否与 B 相交；报告所有相交的线段。

我们分析 T 的构造时间和大小。

引理 9. 给定参数 $\delta > 0$ 和 $0 < \varepsilon' < \frac{1}{3}$，设 P 是 $\mathbb{R}^d$ 中一条具有 n 个顶点的 $6c$-紧致多边形曲线，除最后一条线段外，其余线段的长度均至少为 $\varepsilon'\delta$。可以在 $O(n(\log n + c))$ 时间内按照上述方法构造 T。T 的大小为 $O(cn)$。

证明。 我们首先给出 T 的大小上界。注意，在步骤 1 中，所有线段的关联超立方体集合（$\mathcal{C}$）的大小是线性的。因此，根据 [21] 中的引理 2.11，压缩四叉树的大小至多为 $O(n)$。接下来需要对每个 $u \in T$，给出 $L(u)$ 的大小上界。从 s 的一个关联超立方体开始的下降遍历只会访问该节点及其后代。因此，当且仅当 $s \cap R_u \neq \emptyset$ 且与 s 关联的某个规范超立方体是 u 的区域或 u 在 T 中某个祖先的区域时，线段 s 才存储在节点 u 处。特别地，每个 $s \in L(u)$ 都满足

$$|s| \ \ge\ \mathrm{side}(R_u)/2, \tag{1}$$

因为经过 s 到达 u 的关联超立方体，其边长至少为 $\mathrm{side}(R_u)$，至多为 $2|s|$。现在可以对所有 $u \in T$，$|L(u)|$给出界。令 $\ell = \mathrm{side}(R_u)$，并令 z 为 R_u 的中心。每个 $s \in L(u)$ 都与 R_u 相交于某点 q，且根据 (1)，其长度至少为 $\ell/2$；因此，以 q 为端点的两条 s 子线段中至少有一条长度不小于 $\ell/4$，而它长度为 $\ell/4$ 的起始部分距 q 不超过 $\ell/4$，因此位于球 $B(z,\ \sqrt{d}\,\ell)$ 内。于是， $L(u)$ 中的每条线段都至少为 $B(z,\ \sqrt{d}\,\ell)$ 贡献 $\ell/4$ 的长度。由于 P' 是 $6c$-紧密的，此球内 P' 的总长度至多为 $6c\sqrt{d}\,\ell$，因此，

$$|L(u)| \ \le\ 24\sqrt{d}\,c \ =\ O(c). \tag{2}$$

接下来分析构造时间。预处理的步骤 1 中，$O(n)$ 时间内计算出$\mathcal{C}$。步骤 2 中，根据 [21] 中的引理 2.11，可在 $O(n\log\ n)$ 时间内构造 T。最后，步骤 3 中可在 $O(n\log\ n)$ 时间内对线段排序。利用步骤 2 的数据结构，定位每条线段的 $O(1)$ 个起始节点，每条线段需要 $O(\log\ n)$ 时间，即总计 $O(n\log\ n)$。每次插入操作需要 $O(1)$ 时间，根据 (2)，插入操作的总数为 $\sum_u |L(u)| = O(cn)$。此外，对于发生插入操作的每个节点，下降遍历都会检查其子节点（至多有 2^d 个）。因此，下降遍历所耗费的总时间为 $O(cn)$。由此可知，总预处理时间为 $O(n\log\ n + cn)$。

□

N接下来，我们证明该数据结构的正确性，并分析查询时间。

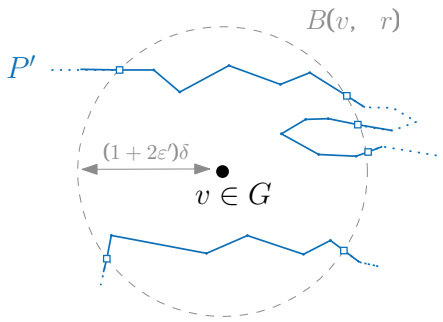

图2：与 $B(v,\ r)$ 相交的 P' 的边的可视化，其中 $r = (1+2\varepsilon')\delta$，以及这些边与球的边界之间的交集。

引理10. 给定参数 $\delta > 0$ 和 $0 < \varepsilon' < \frac{1}{3}$，设 P_I 是 $\mathbb{R}^d$ 中的一条 $6c$-紧密多边形曲线，具有 n 个顶点，除最后一条线段外，其余线段的长度均至少为 $\varepsilon'\delta$。可以在 $O(n\ (\log\ n + c)$ 时间内对 P' 进行预处理，构建上述大小为 $O(cn)$ 的数据结构。给定任意查询点集合 V，对于每个 $v \in V$，都可以报告 P' 中与球 $B(v, r)$ 相交的所有线段，以及每条线段与该球的每个交集的首末点，所需时间为 $O(|V|\ \frac{1}{\varepsilon'^d}(c + \log\ n))$。

证明。我们首先证明该数据结构的正确性。根据最终测试，所有报告的线段都与 B 相交，因此只需证明 S 中与 B 相交的每条线段都经过测试。设 $s \in S$ 满足 $s \cap B \neq \emptyset$，并设 $p \in s \cap B$。由于 $\mathcal{K}_v$ 的单元覆盖 B，因此对于某个 $C \in \mathcal{K}_v$，有 $p \in C$。设 Q 为与 s 关联且包含 p 的一个规范超立方体；这样的超立方体必定存在，因为 s 的关联超立方体覆盖 s。根据步骤 1 以及 P' 的最小线段长度，

$$\text{side}(Q)\ \geq\ \frac{|s|}{2\sqrt{d}}\ \geq\ \frac{\varepsilon'\delta}{2\sqrt{d}}\ \geq\ b\ =\ \text{side}(C).$$

规范超立方体 Q 和 C 都包含 p，因此它们彼此嵌套；根据上述不等式，$C \subseteq Q$。由于 Q 是 T 中包含 C 的某个节点的区域，因此对 C 进行的单元查询成功，并返回一个满足 $C \subseteq R_{u_C} \subseteq Q$ 的节点 u_C。从 Q 向下到 u_C 的路径上，每个节点 w 都满足 $p \in C \subseteq R_{u_C} \subseteq R_w$，因此 s 与 R_w 相交；于是，在步骤 3 中从 Q 开始对 s 进行下降遍历时，到达了 u_C，并将 s 插入 $L(u_C)$。因此，处理单元 C 时会测试 s。同样的论证表明，如果 T 中没有节点包含 C，那么 S 中就没有线段与 C 相交。

最后，我们分析查询时间。G 的每个顶点会产生 $O(1/\varepsilon'^d)$ 个单元。每个单元都需要在 T 中进行一次耗时为 $O(\log\ n)$ 的查询。到达 T 中包含目标区域的最低节点后，根据 (2)，扫描存储于该处的线段列表需要 $O(c)$ 时间。再加上对 P_I 可能较短的最后一条线段进行 $O(\frac{1}{\varepsilon'^d}(1)$ 时间的测试，每个顶点所花费的时间为 $O(\frac{1}{\varepsilon'^d}(c + \log\ n))$，所有顶点的总时间为 $O(|V| \cdot\ \frac{1}{\varepsilon'^d}(c + \log\ n))$。

T 的规模和构造时间由引理 9 得出。 □

我们使用上述数据结构，对每个顶点 $v \in G$ 查询所有与球 $B(v, r)$ 相交的边 $e \in P'$，以及边上与球相交的第一个和最后一个点（见图2）。令 $\mathcal{S}_v$ 表示与对应于 v 的球相交的边集。我们按照这些边在 P' 上的出现顺序对其排序。然后，对于 G 的每个顶点，我们按排序顺序遍历这些边，以找出完全位于各个球内的最大连续子轨迹的起点和端点。我们按照这些子轨迹沿 P' 的出现顺序，将每条子轨迹加入 v 的邻域 $\mathcal{N}_v$ 中（见图3）。

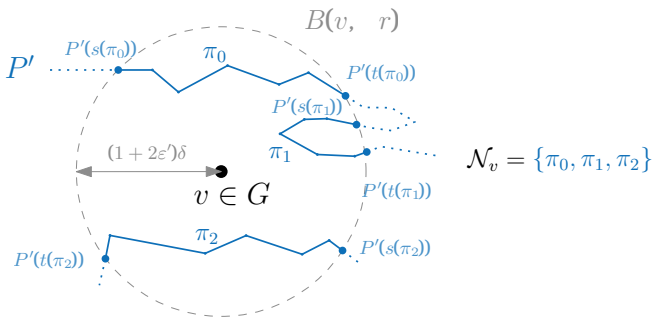

图3：添加到邻域 $\mathcal{N}_v$ 中的子轨迹及其在 P'上的端点。

我们证明，对于每个 $v \in G$，v的邻域可在 $O(\frac{1}{\varepsilon' d}(c+\log n)+\frac{c}{\varepsilon'}\log\frac{c}{\varepsilon'})$ 时间内构建，且每个邻域的大小为 $O(\frac{c}{\varepsilon'})$。

引理*11*。给定参数 $\delta > 0$和 $0 < \varepsilon' < \frac{1}{3}$，设 P'是 $\mathbb{R}^d$中的一条 $6c$-*packed* 多边形曲线，具有 n 个顶点，其所有线段（最后一条除外）长度至少为 $\varepsilon'\delta$。可以在 $O(n(\log n + c))$ 时间内对 P'进行预处理，将其构建为上述大小为 $O(cn)$的数据结构，使得在 $O\left(|V|\cdot(\frac{1}{\varepsilon' d}(c+\log n)+\frac{c}{\varepsilon'}\log\frac{c}{\varepsilon'})\right.$ 时间内，可以为所有 $v \in G$构造邻域 N_v。对于任意 $v \in G$，N_v 的大小为 $O(\frac{c}{\varepsilon'})$。

证明。由引理10可知，对于每个顶点 $v \in G$，计算 P'中与$B(v,r)$相交的线段集合，以及它们的起点和端点，需要 $O\left(|V|\cdot\left(\frac{1}{\varepsilon' d}(c+\log n)\right)\right)$ 时间。给定每个顶点对应的线段集合，随后还需要 $O(\frac{c}{\varepsilon'}\log\frac{c}{\varepsilon'})$时间对线段进行排序，并需要 $O(\frac{c}{\varepsilon'})$时间计算位于球内的极大连续子轨迹的起点和端点；这些子轨迹构成了 v的邻域 $\mathcal{N}_v$。

接下来需要证明所构造邻域的规模。由于 P'中除最后一条边外的所有边的长度都至少为 $\varepsilon'\delta$，任何与 $B(v,\ r)$相交的边，在 $B(v,\ r+\varepsilon'\delta)$中至少占据 $\varepsilon'\delta$。因为 P'是 $6c$-打包的，所以 $B(v,\ r+\varepsilon'\delta)$中与 $B(v,\ r)$重叠的 P'边至多有 $\frac{6c}{(r+\varepsilon'\delta)\,\varepsilon'\delta} = O(\frac{c}{\varepsilon'})$ 条。□

现在我们根据邻域构造 H的节点。令 $\Gamma = \{0,\ \varepsilon'\delta,\ 2\varepsilon'\delta,\ \ldots\} \cap [0, L] \cup \{L\}$为沿着 P'、间距为 $\varepsilon'\delta$的（1D）网格。对于每个顶点 $v \in G$以及 $\mathcal{N}_v$中的每条子轨迹 $P'[s,t]$，我们从 0 开始，针对每个网格点 $g \in \Gamma \cap [s,t]$，在 U 中添加一个节点。我们将这些称为$\mathcal{N}_v$的样本，并按照它们沿 P'的顺序记为 $g_1^v < g_2^v < \ldots < g_{m_v}^v$。对于每个节点，我们存储三个属性。第一个属性 $curvePoint$存储样本的参数 g，因此也存储节点对应的点 $P'(g)$[1]。第二个属性 $vertex$存储 G 中与该子轨迹邻域对应的顶点。最后一个属性 $index$存储样本 g_i^v 在$\mathcal{N}_v$ 的有序样本列表中的位置 i（见图4）。

$\mathcal{N}_v$ 的样本数量为 $m_v = O(c/\varepsilon')$：对于 $\mathcal{N}_v$ 中除 L 以外的每个样本 g，片段$P'[g, g+\varepsilon'\delta]$的长度为 $\varepsilon'\delta$，且位于 $B(v, r+\varepsilon'\delta)$ 中，这些片段彼此内部不相交；由于 P'是 $6c$-紧密的，因此此类片段至多有 $6c(r+\varepsilon'\delta)/(\varepsilon'\delta)+1 = O(c/\varepsilon')$ 个。给定按顺序排列的$\mathcal{N}_v$ 子轨迹后，可在 $O(|\mathcal{N}_v| + m_v) = O(c/\varepsilon')$ 时间内枚举这些样本。

由于 H 的节点由邻域中的样本构成，因此有 $|U| = O(\sum_{v\in G} m_v)$。因此，$H$至多有 $O(|V|\cdot\frac{c}{\varepsilon'})$ 个节点。我们得出，在

[1]注意，U中可能有多个不同的节点对应曲线 P' 上的同一个点，例如，该点同时位于 $B(v,r)$ 和 $B(w,r)$ 中。

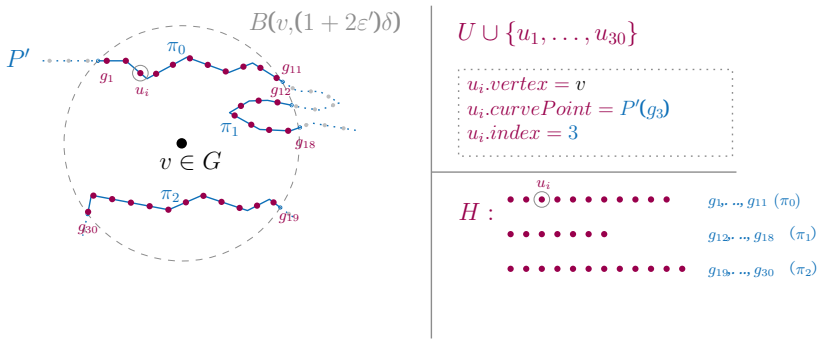

图4： H顶点构造的可视化。

经过$O(n(\log n + c))$预处理后，可以构造出 G中所有顶点的邻域，并在 $O(|V| \cdot \left(\frac{1}{\varepsilon'^d}(c + \log n) + \frac{c}{\varepsilon'} \log \frac{c}{\varepsilon'}\right))$ 时间内构造出 H的节点。

推论 12。 给定参数 $\delta > 0$ 和 $0 < \varepsilon' < \frac{1}{3}$，设 P' 是 $6c$-紧致的多边形曲线，位于 $\mathbb{R}^d$ 中并具有 n 个顶点，其中除最后一条线段外，所有线段的长度均至少为 $\varepsilon'\delta$。可以在 $O(n$ 中以 $+ |V| \left(\frac{1}{\varepsilon'^d}(c +$ 对数 $n) + \frac{c}{\varepsilon'}$ 对数 $\frac{c}{\varepsilon'}\right))$ 时间构造 H 的节点集合 U，并使用 $O(cn)$ 空间。此外，$|U| \leq O(|V| \cdot \frac{c}{\varepsilon'})$。

现在，我们利用 G 的顶点邻域来构造 H 的弧。

阶段 2：构造 H 的弧在本节中，我们利用上一阶段计算出的 G 中顶点的邻域来计算 H 的弧。回顾上文，$\varepsilon' = \varepsilon/5$、$r = (1 + 2\varepsilon')\delta$ 以及 $\tau = (1 + \varepsilon')r$。设 u_i、$u_j \in U$ 是 H 的节点，并设 $s = u_i.curvePoint \leq u_j.curvePoint = t$。设 $\pi = P'[s, t]$ 是由端点定义的 P' 上的子轨迹。仅当存在一条边 $e \in G$ 使得 $d_F(\pi, e) \leq \tau$，或存在一个顶点 $v \in G$ 使得 $d_F(\pi, v) \leq r$ 时，我们才向 F 添加一条弧 $f = (u_i, u_j)$；具体添加哪些弧将在下文说明。在这种情况下，我们称 π 可以映射到边 e （或退化边（v, v））。

注意到， G 中顶点的邻域已经包含了 P'中所有能够映射到该顶点的极大连续子轨迹。为了查询子轨迹 $\pi \subseteq P'$是否能够映射到边 $e \in G$，我们在 O中以 [14] 进行预处理，将 P' 在 O中的数据结构构建时间设为 $\frac{1}{\varepsilon'^{2d}} \log^2(\frac{1}{\varepsilon'}) n \log^2 n$（见事实13），使得对于 P' 的任意子轨迹 $P'[s, t]$ 和任意边 $e \in G$，都能在 $O(\frac{1}{\varepsilon'^2} \log n \log(\log n))$时间内计算出值 δ'，满足 $d_F(P'[s, t], e) \leq \delta' \leq (1 + \varepsilon') \cdot d_F(P'[s, t], e)$。对于 G中的每条边，我们使用与该边端点对应的邻域，确定需要向上述数据结构查询的 P' 中的子轨迹。为减少查询次数，我们利用以下事实：邻域按照其在P'上的出现顺序排列；结合多边形曲线的性质，可以采用二分搜索。这样，我们能够在 $O(|E| \frac{c}{\varepsilon'^3} \log \frac{c}{\varepsilon'} \cdot \log n \log \log n)$时间内构造出 H的弧集。下面将详细介绍我们的构造方法。首先，我们引入一些记号，以帮助建立P' 上的点（不一定是顶点）与 H 的节点之间的对应关系，反之亦然。

令 $\gamma : ([0, L], V \to U)$ 为一个函数，用于将采样点映射到其在 H 中对应的节点。例如， $\gamma(g_i^v, v)$ 返回 $u \in U$ 对应于 v 的邻域中的第 i 个样本的节点。换言之，当且仅当 $u.curvePoint = g$ 且 $u.vertex = v$ 时， $\gamma(g, v) = u$。如果满足这一条件，我们就称节点 $u \in H$ 对应于 P' 上的点 $P'(g)$。

我们将 H 的弧划分为三类：遍历弧、邻域间弧和邻域内弧。我们分别介绍每类弧的构造方法。

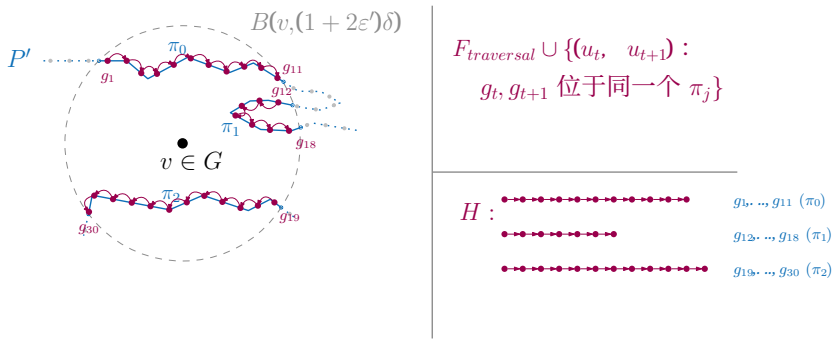

图5：相对于 v的邻域，遍历弧的示意图。同一子轨迹$\mathcal{N}_v$ 中的连续样本由遍历弧连接。

遍历弧遍历弧用于表示将 P' 的某条子轨迹单调映射到 G的某个顶点 v 的潜在单调映射。显然，对于每个顶点 $v \in G$，位于 v邻域中的所有子轨迹 π 均满足 $d_F(\pi, v) \leq r$。为在 H中体现这一点，对于每个顶点 $v \in G$，以及 g_i^v，g_{i+1}^v 这对属于$\mathcal{N}_v$中同一子轨迹的连续样本（它们位于$\mathcal{N}_v$ 中），我们添加一条从 $\gamma(g_i^v, v)$指向 $\gamma(g_{i+1}^v, v)$的有向弧，见图5。

邻域间弧邻域间弧 $(u_i, u_j) \in F$ 连接节点 $u_i, u_j \in U$，它们对应于 G中相邻顶点 (v, w) 的邻域内的样本。这类弧表示子轨迹 $P[u_i.curvePoint, u_j.curvePoint]$（属于 P'）与对应边 $(v, w) \in G$之间的潜在单调映射。

对于每条边 $e = (v, w) \in G$，令 $\mathbf{N}_v$ 和 $\mathbf{N}_w$ 分别为其端点对应的邻域。对于 $\mathbf{N}_v$中的每个样本 g，我们的目标是找到 $\mathbf{N}_w$ 中的第一个样本 g_k^w，使得 $g_k^w \geq g$ 且 $d_F(P'[g, g_k^w], (v, w)) \leq \tau$。若满足这一条件，则 g可以到达 g_k^w。找到后，我们在 H中添加一条从 $\gamma(g, v)$ 指向 $\gamma(g_k^w, w)$ 的有向弧。我们将该弧标记为 start，以表示这是 $\mathbf{N}_w$ 中 g 可以到达的第一个样本。随后，我们重复这一过程，找到 $\mathbf{N}_w$中的最后一个样本 g_ℓ^w，$k \leq \ell$，使得$d_F(P'[g, g_\ell^w], (v, w)) \leq \tau$。我们用结束标签标记这类弧（见图6）。注意，允许 $g_k^w = g$。随后，曲线在 $P'(g)$ 处等待，同时遍历边 (v, w)；当且仅当$P'(g)$与 v 和 w的距离都不超过 r 时，这是可行的。因此，H 可能包含节点对应相同参数的弧所构成的有向环。k 和 ℓ 的具体计算方式及其满足的保证将在下文推论 15 中给出。

虽然 G 是无向的，但我们假设每条无向边 (v, w) 都由两条有向边表示：(v, w) 和 (w, v)。因此，我们会对每次查询执行两次（一次针对 (v, w)，另一次针对 (w, v)）。

为了高效地确定 $d_F(P'[g, g_\ell^w], (v, w)) \leq \tau$，我们通过 [14] 将 P' 预处理到数据结构中。

事实*13*（[14] 中的定理*5.9*）。令 $0 < \varepsilon' < \frac{1}{3}$。给定一条在 $\mathbb{R}^d$ 中具有 n 个顶点的多边形曲线 P'，可以在 $O(\frac{1}{\varepsilon'^{2d}} \log^2 \frac{1}{\varepsilon'} \cdot n \log^2 n)$ 时间内构建一个数据结构，并使用 $O(\frac{1}{\varepsilon'^{2d}} \log^2 \frac{1}{\varepsilon'} \cdot n)$ 空间，使得对于查询线段 (v, w)，以及曲线上的任意两个点 $P'(a)$ $P'(b)$（及包含它们的曲线线段），都可以在 $O(\frac{1}{\varepsilon'^2} \log n \log \log n)$ 时间内对弗雷歇距离 $d_F(P'[s, t], (v, w))$ 进行 $(1+\varepsilon')$-近似。

对于 $\mathcal{N}_w$ 中的每个样本 g_i^w，都可以使用边（v，w）和子轨迹 $P'[g, g_i^w]$ 查询数据结构，以确定（$1+\varepsilon'$）近似弗雷歇距离。如果该近似

[2]此处的“第一”是根据样本的排序 d 相对于样本顺序 o 定义的

即 $\mathcal{N}_w$。

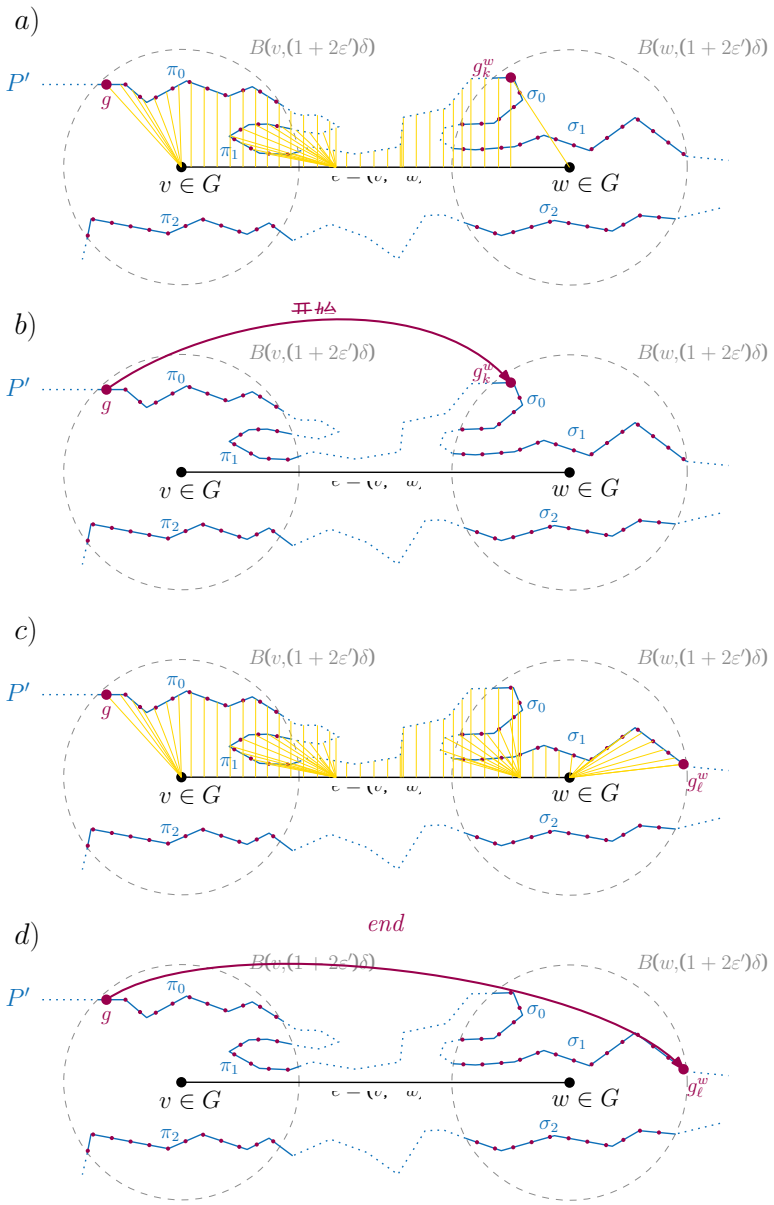

图6：a）部分展示了$\mathcal{N}_v$ 的一个样本 g，以及$\mathcal{N}_w$ 的第一个样本 g_k^w，其中 $g_k^w \geq g$ 满足条件$P\prime[g, g_k^w]$可在弗雷歇距离 τ内映射到 e。b）部分展示了因此我们从 g 到 g_k^w绘制一条标记为起点的邻域间弧。类似地，c）部分展示了 g_ℓ^w是$\mathcal{N}_w$的最后一个样本，满足 $P\prime[g, g_\ell^w]$可在弗雷歇距离 τ内映射到 e；d）部分展示了由此得到的从 g 到g_ℓ^w的邻域间弧，标记为终点。

弗雷歇距离， δ'，数据结构输出的距离至多为 τ，我们称 g 可以到达 g_i^w。我们进一步利用邻域按排序顺序存储样本这一观察来减少查询次数。对于$\mathcal{N}_v$中的每个样本 g，不再查询$\mathcal{N}_w$ 中的所有样本以找到它可以到达的第一个和最后一个样本，而是在$\mathcal{N}_w$ 的样本上（按照它们沿P'的顺序）进行二分搜索，从而针对$\mathcal{N}_v$中的每个样本，找到$\mathcal{N}_w$ 中它可以到达的第一个和最后一个样本。二分搜索的依据是以下单调性性质。

观察 *14*。令 $e=(a,b)$、$\rho>0$ 和 $s\le x\le y$ 满足 $d_F(P'[s,y],e)\le\rho$。（*i*）如果 $\|P'(x)-b\|\le\rho$，那么 $d_F(P'[s,x],e)\le\rho$。（*ii*）如果 $\|P'(x)-a\|\le\rho$，那么 $d_F(P'[x,y],e)\le\rho$。

证明。在每个匹配对的距离至多为 ρ 的匹配中，$P'(x)$ 与某个点 $e(\mu_x)$ 匹配。由于 $\|P'(x)-e(\mu)\|$ 在 μ 上是凸的，因此在 $e(\mu_x)$ 与 b (对于 (i)) 之间的整个子线段上，它至多为 ρ；对于 (ii)，则在 $e(\mu_x)$ 与 a (对于 (ii)) 之间的整个子线段上至多为 ρ。保留 ($P'(x)$ $e(\mu_x)$) 一侧的匹配，并将 $P'(x)$ 与 e 的其余部分匹配。□

固定 N_v 的一个样本 g 和有向边 $e=(v,w)$，并将 N_w,$1\le j\le m_w$ 的样本记为 $g_j=g_j^w$。对于 $g_j\ge g$，记 $D(j)=d_F(P'[g,g_j],e)$，并令 δ'_j 为事实13中的数据结构针对 $P'[g,g_j]$ 和 e 返回的值，因此 $D(j)\le\delta'_j\le(1+\varepsilon')D(j)$。令 k 为满足 $g_k\ge g$ 的最小索引；通过对 N_w 的样本进行二分搜索即可找到它。接着，我们在 $\{k-1,\dots,m_w+1\}$ 中通过二分搜索找到 ℓ，同时维护索引 $\mathsf{lo}<\mathsf{hi}$，并保持以下不变量：$\mathsf{lo}=k-1$ 或$\delta'_{lo}\le\tau$，且 $\mathsf{hi}=m_w+1$ 或 $\delta'_{hi}>\tau$。初始时 $\mathsf{lo}=k-1$，$\mathsf{hi}=m_w+1$；每一步查询中间索引，并相应更新 lo 或 hi，直到 $\mathsf{hi}=\mathsf{lo}+1$。令 $\ell=\mathsf{lo}$。若 $\ell\ge k$，则添加起始弧 ($\gamma(g,v)$ $\gamma(g_k,w)$) 和终止弧 ($\gamma(g,v)$ $\gamma(g_\ell,w)$)；否则不添加弧。

推论 15. 采用上述记号，搜索需要 $O(\log\frac{c}{\varepsilon'})$ 次查询，并且 –

(i) $D(j)\le\tau$ 对所有 $k\le j\le\ell$均成立，并且

(ii) $D(j)>r$ 对所有 $j>\ell$ 均成立（注意，$j>\ell\ge k-1$ 蕴含 $j\ge k$，因此 $D(j)$ 有定义）。

证明. 查询次数由 $m_w=O(c/\varepsilon')$ 得出。

对于 (i)，令 $k\le j\le\ell$。根据不变量，$D(\ell)\le\delta'_\ell\le\tau$。此外，$g\le g_j\le g_\ell$ 和$\|P_\prime(g_j)-w\|\le r\le\tau$，因为 g_j 是 $\mathcal{N}_w$ 的一个样本。由 $\rho=\tau$，观察 14(i) 得出 $D(j)\le\tau$。

对于 (ii)，令 $h=\ell+1\le m_w$。根据不变量，$\delta'_h>\tau$，因此 $D(h)\ge\delta'_h/(1+\varepsilon')>\tau/(1+\varepsilon')=r$。假设对于某个 $j>h$，$D(j)\le r$。由于 $g\le g_h\le g_j$ 且 $\|P'(g_h)-w\|\le r$，由 $\rho=r$，观察 14(i) 得出 $D(h)\le r$，矛盾。□

邻域内弧 邻域内弧连接同一邻域内的样本。这些弧对于将上面构建的邻域间弧限制为每个样本两条弧（一个起始 弧和一个结束 弧）是必要的。邻域内弧将某个邻域的子轨迹中最后一个样本，连接到该邻域后续包含样本的子轨迹中的第一个样本。换言之，对于所有 $v\in G$，对于$\mathcal{N}_v$ 中位于$\mathcal{N}_v$的不同子轨迹中的每一对连续样本 g_i^v，g_{i+1}^v，我们构建一条从 $\gamma(g_i^v,\ v)$ 到 $\gamma(g_{i+1}^v,\ v)$ 的有向邻域内弧（见图 7）。

顶点 $v\in G$所在邻域中的每条邻域内弧（$\gamma(g_i^v,\ v)$ $\gamma(g_{i+1}^v,\ v)$）都会为每个相邻顶点 $w\in G$存储$\mathcal{N}_w$ 中样本索引的一个区间 $I_w=[a,b]$，使得$d_F(P'[g_x^w,g_{i+1}^v],\ (w,\ v))\le\tau$对每个 $x\in I_w$均成立；I_w的选取方式将在下文说明。

为了计算每条邻域内弧的范围，我们按顺序遍历每个邻域中的样本，并利用入射邻域间弧的信息。对于每个

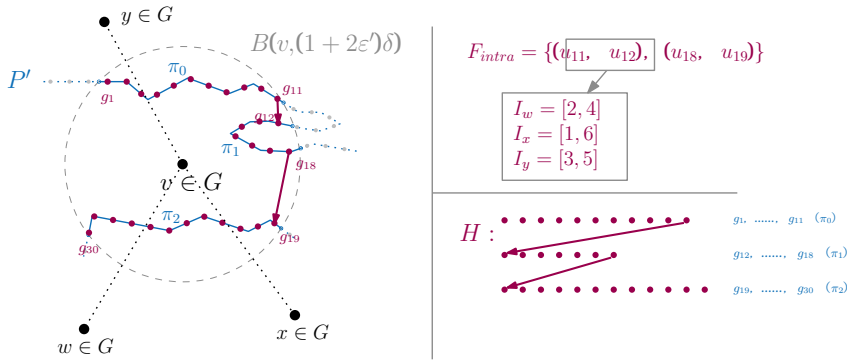

图 7：邻域内弧的可视化，这些弧将子轨迹的最后一个样本与下一条子轨迹的第一个样本连接起来，其中子轨迹属于$\mathcal{N}v$，以及相对于 v 的每个邻域的弧范围示例。

含有样本 $g_1^v, \ldots, g_{m_v}^v$ 的邻域 $\mathbf{N}v$，我们按照样本沿 P' 出现的顺序[3]进行处理。在遍历过程中，对于每个相邻顶点 $w \in G$，我们都会跟踪 $\mathbf{N}w$ 的活动采样点集合。若在遍历 $\mathbf{N}v$ 时遇到一个样本g_i^v，且它对应某条邻域间弧 $(\gamma(g_x^w, w)\ \gamma(g_i^v, v)) \in H$ 的端点，该弧带有起始标签并从 g_x^w 出发，而我们尚未遇到具有对应结束标签的邻域间弧，则称 $\mathbf{N}w$ 的样本 g_x^w 处于活动状态。我们按如下方式处理 $\mathbf{N}v$ 的每个样本 g_i^v。对于每个相邻顶点 $w \in G$，我们首先检查进入 $\gamma(g_i^v, v)$ 的邻域间弧集合。对于任何带有起始标签的弧，我们都会更新与该边起点所属邻域相关的活动采样点集合。随后，对于每个相邻顶点 $w \in G$，若有一条邻域内弧进入 $\gamma(g_i^v, v)$，我们就更新其范围 I_w，将当前 $\mathbf{N}w$ 活动采样点集合的索引纳入其中。最后，如果进入 $\gamma(g_i^v, v)$ 的邻域间弧中有任何一条带有结束标签且起点为 g_x^w，我们就通过从相关活动采样点范围中移除该样本的索引，使其变为非活动状态。我们将 $\mathbf{N}w$ 活动采样点索引所构成的最小包含区间存储为 I_w（使用带延迟删除的优先队列维护）。虽然近似查询时活动集合本身不一定连续，但 I_w 中的每个索引都是有效的：设 $x \in I_w$，则存在活动采样点g_y^w, g_z^w，满足 $y \le x \le z$。由于 g_y^w 处于活动状态，根据推论 15（应用于 g_y^w 和边 (w, v)），可得$d_F(P'[g_y^w, g_i^v], (w, v)) \le \tau$, $g_x^w \le g_z^w \le g_i^v$，且 $||P'(g_x^w) - w|| \le r \le \tau$。因此，由观察 14(ii) 可得 $d_F(P'[g_x^w, g_i^v], (w, v)) \le \tau$。

至此， H的构造完成。接下来，我们论证该数据结构的正确性，并分析其规模和构造时间。为方便记号起见，如果 H 中的一条路径以满足 $u_i.curvePoint = 0$ 的节点 $u_i \in H$ 为起点，并以满足 $u_j.curvePoint = L$的节点 u_j 为终点，且路径上的每条邻域内弧都满足第 4.2 节中相对于路径上它之前最近的邻域间弧所规定的条件，则称该路径为有效的。

引理 *16.* 给定参数 $\delta > 0$和 $0 < \varepsilon' < \frac{1}{3}$，令 P为一条复杂度为 n的 c-紧致曲线，且令 $G(V, E)$ 为 $\mathbb{R}^d$中的一般图。可以在 $O(\frac{1}{\varepsilon'^{2d}} log^2 \frac{1}{\varepsilon'} \cdot n log^2 n + cn + |V| \frac{1}{\varepsilon'^{d}} (c + log\ n) + |E| \frac{c}{\varepsilon'^{3}} log \frac{c}{\varepsilon'} \cdot log\ n log\ log\ n)$ 时间内，使用 $O(cn + |E| \frac{c}{\varepsilon'} + (\frac{1}{\varepsilon'^{2d}} log^2 \frac{1}{\varepsilon'}) n)$ 空间，构造一个规模为 $O(|E| \frac{c}{\varepsilon'})$ 的有向图 H，使得：如果 H中存在有效路径，则存在一条路径 $Q \in G$，满足 $d_F(Q, P) \le (1 + \varepsilon)\delta$；如果不存在有效路径，则对于所有路径 $Q \in G$，均有 $d_F(Q, P) > \delta$。

[3]请注意，这些内容已按其沿 P'出现的顺序排好序并存储。

证明。对于一个节点 u，我们称 $u.curvePoint$ 为其参数。遍历和邻域内弧将某个$\mathcal{N}_v$ 的一个样本连接到$\mathcal{N}_v$的下一个样本，而从 $\gamma(g, v)$ 出发的邻域间弧终止于一个参数至少为 g的节点。因此，沿任意路径 H 的参数均为非递减的。回顾 $\tau = (1+\varepsilon')r$，其中 $r = (1+2\varepsilon')\delta$ 且 $\varepsilon' = \varepsilon/5$，并且 $d_F(P, P') \le \varepsilon'\delta$。

可靠性。设 W 是一条有效路径，其邻域间弧依次为 $(\gamma(a_i, v_{i-1})\ \gamma(b_i, v_i))$、$1 \le i \le m$。那么 $Q = (v_0, \ldots, v_m)$ 是 G 中的一条路径。令 $a_{m+1} = L$。在两个连续的邻域间弧之间，W 始终位于 $\mathbf{N}_{v_i}$ 中，沿遍历弧和邻域内弧从 b_i 移动到 a_{i+1}。令 c_i 为这段路径上最后一条邻域内弧的终点；若不存在这样的弧，则令 $c_i = b_i$。那么，$P'[c_i, a_{i+1}]$ 由遍历弧覆盖，因此它位于 $\mathbf{N}_{v_i}$ 的某一条子轨迹中，进而位于 $B(v_i, r)$ 中。同样，由于 W 有效，$P'[0, a_1]$ 位于 $B(v_0, r)$ 中。此外，$d_F(P'[a_i, c_i], (v_{i-1}, v_i)) \le \tau$。若 $c_i = b_i$，则根据推论 15(i)，此结论成立，因为 b_i 是为 a_i 找到的第 k 或第 ℓ 个样本。否则，进入 $\gamma(c_i, v_i)$ 的邻域内弧所包含的范围包括 a_i 的索引，因此上述范围的有效性保证了该界限成立。将每个片段 $P'[c_i, a_{i+1}]$ 匹配到 v_i，并将每个片段 $P'[a_i, c_i]$ 匹配到边 (v_{i-1}, v_i)，相邻的匹配在其公共端点处一致。由此得到 $d_F(Q, P') \le \tau$，因此 $d_F(Q, P) \le \tau + \varepsilon'\delta = (1 + 4\varepsilon' + 2\varepsilon'^2)\delta \le (1+\varepsilon)\delta$。

完备性。设 $Q = (v_0, \ldots, v_m)$ 是 G 中的一条路径，其中 $d_F(Q, P) \le \delta$，因此 $d_F(Q, P') \le (1+\varepsilon')\delta$。固定这样的匹配，并选择参数 $0 = y_0 \le y_1 \le \cdots \le y_m = L$，使得 $P'(y_i)$ 与 v_i 匹配。对于 $1 \le i \le m$，我们于是有 $d_F(P'[y_{i-1}, y_i], (v_{i-1}, v_i)) \le (1+\varepsilon')\delta$。令 g_i 为满足 $g_i \le y_i$ 的 Γ 的最大网格点；于是 $g_0 = 0$、$g_m = L$ 以及 $g_0 \le \cdots \le g_m$。由于 $P'[g_i, y_i]$ 的长度至多为 $\varepsilon'\delta$，它位于 $B(v_i, r)$ 中。因此，g_i 是 $\mathbf{N}_{v_i}$ 的一个样本，在前面加上与 v_{i-1} 匹配的 $P'[g_{i-1}, y_{i-1}]$，得到 $d_F(P'[g_{i-1}, y_i], (v_{i-1}, v_i)) \le r$。根据 $\rho = r$ 和 $x = g_i$，观察 14(i) 随即得到 $d_F(P'[g_{i-1}, g_i], (v_{i-1}, v_i)) \le r$。

令 k 和 ℓ 分别为针对样本 g_{i-1} 和边 (v_{i-1}, v_i) 计算得到的索引，并令 j_i 为 g_i 在 $\mathbf{N}_{v_i}$ 中的索引。则 $j_i \ge k$，因为 $g_i \ge g_{i-1}$；根据推论 15(ii)，$j_i \le \ell$。因此，H 包含从 $\gamma(g_{i-1}, v_{i-1})$ 到 $\gamma(g_k^{v_i}, v_i)$ 的起始弧。$\mathbf{N}_{v_i}$ 中的连续样本由遍历弧或邻域内弧连接。g_{i-1} 的索引在 $g_k^{v_i}$ 到 $g_\ell^{v_i}$ 期间处于活动状态，因此它属于每条进入满足 $k < j' \le j_i$ 的样本 $g_{j'}^{v_i}$ 的邻域内弧的范围。因此，沿着一条有效路径，可以从 $\gamma(g_{i-1}, v_{i-1})$ 到达 $\gamma(g_i, v_i)$。将这些路径依次连接起来（$i = 1, \ldots, m$），即可得到一条从 $\gamma(0, v_0)$ 到 $\gamma(L, v_m)$ 的有效路径。如果 $m = 0$，那么 P' 位于 $\mathbf{N}_{v_0}$ 的一个子轨迹中，而遍历弧则从 $\gamma(0, v_0)$ 通向 $\gamma(L, v_0)$。

构造时间与空间。根据推论 12，邻域和 $O(|V|\frac{c}{\varepsilon'})$ 个 H 节点可在 $O\big(n(\log n + c) + |V|(\frac{1}{\varepsilon'^{2d}}(c + \log n) + \frac{c}{\varepsilon'}\log\frac{c}{\varepsilon'})\big)$ 时间和 $O(cn)$ 空间内计算。由于$\overline{m_v} = O(\frac{c}{\varepsilon'})$，共有 $O(|V|\frac{c}{\varepsilon'})$ 条遍历弧和邻域内弧，且它们可在相同时间内构建。事实13中的数据结构需要 $O(\frac{1}{\varepsilon'^{2d}}\log^2\frac{1}{\varepsilon'}\cdot n\log^2 n)$ 时间和 $O(\frac{1}{\varepsilon'^{2d}}\log^2\frac{1}{\varepsilon'}\cdot n)$ 空间。对于样本与关联有向边构成的 $\sum_v \overline{m_v}\deg(v) = O(|E|\frac{c}{\varepsilon'})$ 个配对，推论 15 使用 $O(\log\frac{c}{\varepsilon'})$ 次查询，并产生至多两条邻域间弧。总计需要 $O(|E|\frac{c}{\varepsilon'^3}\log\frac{c}{\varepsilon'}\cdot\log n\log\log n)$ 时间，并产生 $O(|E|\frac{c}{\varepsilon'})$ 条弧。最后，每条进入 $\mathbf{N}_v$ 的 $O(\frac{c}{\varepsilon'}\deg(v))$ 条邻域间弧都会导致 $O(1)$ 次优先队列操作，而 $\mathbf{N}_v$ 的每条 $O(\frac{c}{\varepsilon'})$ 邻域内弧都会存储 $\deg(v)$ 个范围。因此，范围需要 $O(|E|\frac{c}{\varepsilon'}\log\frac{c}{\varepsilon'})$ 时间和 $O(|E|\frac{c}{\varepsilon'})$ 空间。将这些界相加即可得到所述的时间与空间复杂度。 □

4.2 在 H 中寻找路径

接下来需要说明如何确定 H 中是否存在有效路径。是否可以使用邻域内弧，取决于此前经过的最后一条邻域间弧，因此我们搜索由节点及其上下文组成的对。请记住，H 可能包含有向环，而这些环全部由参数相同的节点之间的邻域间弧构成。

对于节点 u 和节点 a（或 $a=\perp$），我们将这一对（u，a）称为一个状态。它表示当前位于 u，且最后经过的邻域间弧起始于 a；其中 $a=\perp$ 表示尚未经过任何邻域间弧。从状态（u，a）出发，允许以下转移：

- 沿遍历弧（u，w）到达状态（$w\,a$）；
- 沿邻域内弧（u，w）到达状态（wa），前提是 $a\neq\perp$ 且 $a.index$ 位于弧上存储的范围 $I_{a.vertex}$ 内；
- 沿邻域间弧（u，w）到达状态（wu）。

根据定义，H 中的路径有效，当且仅当它对应于一系列允许的转移：该序列从满足 $u.curvePoint=0$ 的状态（u，$\perp$）开始，并在满足 $u'.curvePoint=L$的状态（u'，a）结束。

为在 O时间内检验范围条件，每条对应边（w，v）的邻域间弧都会存储该边在 v邻接表中的位置，而 $\mathcal{N}v$ 的每条邻域内弧都会将其范围存储在一个以这些位置为索引的数组中。因此，状态也会携带这个位置；为简洁起见，我们省略了这一记号。

搜索。 我们维护一个状态栈 $\mathcal{S}$，并为每个节点 u 设置一个标志 $expanded(u)$，初始值为 false。

1. 初始化。 对于每个满足 $u.curvePoint=0$ 的节点 u，将 $(u,\perp)$ 压入 $\mathcal{S}$。
2. 当 $\mathcal{S}$ 不为空时，弹出一个状态 (u,a)。

   (a) 如果 $u.curvePoint=L$，返回 是。 (b) 如果 u 有一条出向遍历弧 (u,u')，则将 (u',a) 压入栈中。 (c) 如果 u 有一条出向邻域内弧 (u,u')，则 $a\neq\perp$，并且 $a.index\in I_{a.vertex}$，将 (u',a) 压入栈中。 (d) 如果 $expanded(u)$ 为 false，则将其设为 true，并对每条出向邻域间弧 (u,u') 将 (u',u) 压入栈中。

3. 返回 否。

节点的出向邻域间弧只会被扩展一次。这样做足够了，因为通过邻域间弧 (u,u') 到达的状态始终是 (u',u)，无论我们从u 的哪个状态到达。为每个压入的状态存储将其压入的状态，可以让我们在搜索返回 是 时报告一条有效路径，从而得到 G 中的路径 Q 。

引理17。 当且仅当 H 包含一条有效路径时，搜索才会返回是，且其运行时间为 $O(|V|\frac{\epsilon}{\varepsilon'}+|E|\frac{\epsilon^2}{\varepsilon'^2})$ 。

证明。 每个被压入的状态都是通过允许的转移从初始状态到达的，因此只有存在有效路径时，搜索才会返回是。

相反，如果某个状态可以通过允许的转移从初始状态到达，则称该状态为可达状态。我们证明每个可达状态都会被压入。初始状态在初始化步骤中被压入。假设可达状态（u，a）已被压入；那么它也会被弹出，除非搜索已经返回是。当（u，a）被弹出时，沿遍历和邻域内弧到达的后继状态会在步骤 (b) 和 (c) 中被压入。沿邻域间弧到达的后继状态是弧（u，u'）对应的状态（u'，u），且这些状态不依赖于 a。当展开 u 时，这些状态会被压入，而这最迟发生在（，）被弹出时，这

一发生。因此，如果存在有效路径，则状态（u'， a'）会被压入，且搜索会在该状态被弹出时返回是。

关于运行时间，注意到$\mathcal{N}_v$ 的每个节点至多有一条出向遍历或邻域内弧，且该弧指向$\mathcal{N}_v$中的下一个样本。因此，在步骤 (b) 和 (c) 中压入的状态构成若干条链：每条链从一个初始状态或步骤 (d) 中压入的状态开始，并按递增顺序访问同一邻域中的样本。因此，一条链的长度至多为 $m_v = O(\frac{c}{\varepsilon'})$。初始状态至多有 $|V|$ 个，每个包含 p_1 的球对应一个。步骤 (d) 中每条邻域间弧至多跟随一次，因为每个节点至多展开一次，因此步骤 (d) 中压入的状态至多有 $O(|E|\frac{c}{\varepsilon'})$ 个。除步骤 (d) 中的展开外，每次弹出耗时 $O(1)$，而所有展开的总耗时为 $O(|E|\frac{c}{\varepsilon'})$。综上，搜索耗时为 $O((|V|+|E|\frac{c}{\varepsilon'})\cdot\frac{c}{\varepsilon'}) = O(|V|\frac{c}{\varepsilon'} + |E|\frac{c^2}{\varepsilon'^2})$。特别是，即使 H 可能包含环，搜索也会终止。

□

结合引理16和引理17，我们可以得出定理 4。

5 低密度地图匹配

本节针对输入图和多边形曲线均满足 λ-低密度的情况，给出问题2的$(1+\varepsilon)$-近似判定器。我们的判定器在 $O\left(\lambda^2\left(\frac{(n|V|)^{1-1/d}}{\varepsilon^2} + \frac{n+|V|}{\varepsilon^d}\right)\log(n|E|)\right)$ 期望时间内运行。请注意，尽管上文用 π 表示多边形轨迹 P 的子轨迹，但本节用 π 表示图中的一条路径。

定理 *18.* 给定参数 $\delta > 0$ 和 $0 < \varepsilon < 1$，以及复杂度为 $|V|+|E|$的 λ-低密度图 $G(V, E)$和复杂度为 n 的 λ-低密度曲线 P（位于 $\mathbb{R}^d$中）。可以在期望 $O\left(\lambda^2\left(\frac{(n|V|)^{1-1/d}}{\varepsilon^2} + \frac{n+|V|}{\varepsilon^d}\right)\log(n|E|)\right)$ 时间内判定：是否存在 G 中的路径 π，使得$d_F(P,\pi) \le (1+\varepsilon)\delta$；或者，对于 G中的所有路径 π，是否有 $d_F(P,\pi) > \delta$。当 $d = 2$ 时，该界可简化为 $O\left(\frac{\lambda^2}{\varepsilon^2}(n+|V|)\log(n|E|)\right)$。

我们通过分别针对参数 δ 和 ε简化两个数据结构，并证明简化后的数据结构之间的自由空间复杂度为次二次，来获得这些结果。在这里，沿用 Alt、Efrat、Rote 和 Wenk [1], 的定义，自由空间是由满足距离至多为 δ的点对组成的集合，其中一个点位于曲线上，另一个点位于图上。自由空间曲面为曲线的每条边与图的每条边的每个配对分别对应一个单元，而自由空间复杂度则是包含自由空间的单元数量。随后，我们结合 [9] 和 [15] 中的现有结果，证明这意味着存在一个用于地图匹配问题的近似判定器，并且该判定器可在次二次时间内运行。具体而言，给定值 δ 和 ε，我们证明可以在预期 $O(\lambda|V|\log|V|)$时间内，将 G 和 P 预处理为简化数据结构 G' 和 P'，使得两个数据结构之间的自由空间复杂度为次二次。然后，我们利用这一自由空间为原始数据结构设计一个近似判定器，用于判定是否存在路径 $\pi \in G$，使得 π与 P之间的弗雷歇距离至多为 $(1+\varepsilon)\delta$，或者对于 G中的所有路径 π，均有 $d_F(P,\pi) > \delta$。如果存在这样的路径，判定器就输出该路径。

[15] 的作者证明了一种线性时间算法，可以根据参数 $\mu = \varepsilon'\delta$ 简化多边形曲线 P（事实 8）。然而，要简化低密度图，同时保留图所需的性质，则需要更多工作。我们将在第 5.2 节说明如何简化该图，以获得有助于界定其自由空间复杂度的性质。首先，我们回顾一些对分析有用的已有结果。

在本节中，我们用 C 表示一个超立方体，用 xC 表示以其中心为基准按 x 倍缩放C 。

5.1 有用的事实

我们使用 Driemel、Har-Peled 和 Wenk [15]的曲线简化（事实 8）。

我们将使用以下事实来界定边长为 ℓ 的 λ-低密度图中，与边长为 ℓ的超立方体相交的边数。为简化论证，本节中我们始终使用超立方体，而非球。

事实 19 （ [15]中的命题4.7）。设 G 是一个 λ-低密度图， C 是 $\mathbb{R}^d$ 中的一个边长为 ℓ的超立方体。 G 中与 C相交且边长至少为 ℓ 的边数，上界为 $\lambda\left\lceil \left(\frac{\sqrt{d}}{2}\right)^d \right\rceil$。

正如 [15], 中的引理4.9所示，要使一个 λ-低密度数据结构在某个区域内具有较大的边长总和，其附近区域内必定存在许多顶点。我们将利用这一结果，把简化图和简化曲线的自由空间复杂度与原始结构中的顶点数联系起来。然而， [15] 中引理4.9的原始证明假设 λ为常数，并未说明对参数的依赖关系。本文不将 λ 视为常数。因此，我们在下文给出完整证明，并明确其对 λ 的依赖关系。我们利用这样一个事实： λ-低密度图中的每个顶点的度至多为 λ。

引理20 （参见 [15] ，其依赖于 λ）。 设 G 是 $\mathbb{R}^d$ 中的一个 λ-低密度图或路径，且设 C 是边长为 ℓ 的超立方体。设 α 为 G 与 C 交集的长度，并设 M 为 G 中包含于 $3C$ 内的顶点数。则 $M=\Omega\left(\left(\frac{\alpha}{\lambda\ell}\right)^{1+\frac{1}{d-1}}\right)$，这等价于 $\alpha=O\left(\lambda\ell M^{1-1/d}\right)$ 。

证明。 我们首先注意到， G 的每个顶点 u 的度为 $O(\lambda)$。固定一个 ρ、 $0<\rho\leq\ell$，并根据长度将与 C 相交的 G 的边划分为两组。与 C 相交且长度小于 ρ 的边，其两个端点都位于距 C $\rho\leq\ell$ 的距离内，因此它们必定位于 $3C$ 中。由于每个顶点的度为 $O(\lambda)$，这样的边有 $O(\lambda M)$ 条，并且它们共同对 α 贡献 $O(\lambda M\rho)$。

接下来，用边长为 ρ的 $O((\ell/\rho)^d)$个立方体覆盖 C 。根据事实 19，这些立方体中的每一个至多与 $\lambda\lceil(\frac{\sqrt{d}}{2})^d\rceil=O(\lambda)$条长度至少为 ρ的边相交，而每条这样的边对小立方体的贡献至多为 $\sqrt{d}\cdot\rho$。因此，长度至少为 ρ 的边贡献了 $O(\lambda(\ell/\rho)^d\rho)=O(\lambda\ell^d/\rho^{d-1})$给 α，所以总计为 $\alpha=O(\lambda(M\rho+\frac{\ell^d}{\rho^{d-1}}))$。

取 $\rho=\ell/M^{1/d}\leq\ell$可得 $\alpha=O\left(\lambda\ell M^{1-1/d}\right)$。重新整理各项即可得到第一个结论。

□

5.2 低密度图的预处理

简化 λ-低密度图 G 的目的，是减少简化图中与 P的边的距离在 δ 以内的边数，从而降低这两个数据结构之间的自由空间复杂度。目标是在实现这一性质的同时，保持对于原图中的任意路径，简化图中都存在一条相对于我们的弗雷歇距离查询值而言“接近”的路径。 [15]的作者表明，可以在线性时间内对曲线进行简化，使简化曲线接近原曲线（见事实 8）。此外，他们还表明，特别对于 λ-低密度曲线，在任何区域内，事实 8 所得到的简化结构的复杂度，相较于原曲线在邻近区域内的复杂度都必须较低。这是因为简化只能在曲线原本具有大量边长的区域中违反低密度，而为了保持低密度，这些区域必然具有大量顶点。因此，可以将该区域内简化结构的复杂度分摊给邻近区域中原曲线的顶点，使得每个顶点被计费的次数不超过某个次线性数量。

这是因为简化只能在曲线原本具有大量边长的区域中违反低密度，而为了保持低密度，这些区域必然具有大量顶点。因此，可以将该区域内简化结构的复杂度分摊给邻近区域中原曲线的顶点，使得每个顶点被计费的次数不超过某个次线性数量。

另一方面，简化低密度图并没有那么直接。一种降低图复杂度的方法是对其进行聚类。然而，标准聚类方法无法实现新图的复杂度与原图附近顶点数量之间的相同关系。这是因为聚类图必须保持与原图相同的连通性，才能确保对于原图中的任意路径，简化图中都存在一条“接近”它的路径。因此，聚类图在特定区域内的边数可能大幅增加，却无法保证原图在该区域内有许多顶点可供分摊。

本节将展示如何针对判定问题的查询值 δ以及参数 ℓ，构造低密度图 G 的简化 $G' = (V, E')$，其中该参数与近似因子 ε 和 δ有关，使得简化图中距离 P 的一条边不超过 δ 的边数至多为 G复杂度的次线性量。具体而言，我们将 G处理为 G'，使其满足以下性质：

性质 1： 对于所有路径 $\pi \in G$，都存在一条路径 $\sigma \in G'$，使得π 与 σ 之间的弗雷歇距离至多为 $(1 + 4\sqrt{d})\ell$。反之，对于所有路径 $\sigma \in G'$，都存在一条路径 $\pi \in G$，使得 π 与 σ 之间的弗雷歇距离至多为 ℓ。

性质 2： 在 G 中，由 E' 中一条边连接的 G' 的顶点之间的图距离至少为ℓ。换句话说，$\forall v' \neq u' \in V, d_G(u', v') \geq \ell$。注意，这并不意味着 G' 具有最小边长。

性质 3： 设 C 是一个边长为 δ 的立方体。G' 与 $3C$ 重叠的边不能超过 $O\left(\lambda\left(\frac{|V \cap 27C|^{1-1/d}}{\varepsilon} + \frac{|V \cap 27C|^{1/d}}{\varepsilon^{d-1}}\right)\right)$ 条。

我们分两个阶段，针对两个参数 δ 和 ℓ 构造 G'。在这两个阶段中，我们始终保持性质 1。通过将图聚类为中间图 G^*，可以实现性质 2。然而，对图进行聚类可能会显著增加与某个区域重叠的简化边数量，从而违反性质 3。因此，为实现第三个性质，我们会对聚类图进行剪枝。

阶段 1：对 λ-低密度图进行聚类 该阶段旨在减少图中短边的数量，并在顶点之间建立最小图距离（其中，图距离是相对于原图 G 测量的）。但请注意，这可能不会使聚类图 G^* 中的边长达到最小值。

为了对图进行聚类，我们使用 r-网。G 的 r-网是满足以下两个性质的子集 $\mathcal{C} \subseteq V$：

分离性 $\mathcal{C}$ 中的任意两个点之间的图距离都不小于 r。

覆盖性 V 中的每个点与 $\mathcal{C}$ 中的某个点之间的图距离都不超过 r。

通过观察 $|E| \leq \lambda |V|$，可以将引理 2.4 中的结果推广到 λ-低密度图 [16]，从而得到：

事实 21 (引理 2.4 中 [16])。 给定一个 λ-低密度图 $G=(V,E)$，可以在 $O(\lambda|V|\log|V|)$ 期望时间内计算出一个 r-网 G。

我们使用 [16] 的随机化构造来构造一个 r-网，其中 r 等于指定参数 ℓ。设得到的 r-网为 $\mathcal{C}=\{c_1,...,c_{|\mathcal{C}|}\}$。我们将这些称为簇中心。我们通过将 G 中的每个顶点分配给距离其最近的簇中心，来构造对应的簇 $\mathbb{C}=\{C_1,...,C_{|\mathcal{C}|}\}$；这里"最近"是指在 G 中按图距离计算（从 O 中的所有簇中心执行一次 Dijkstra 算法的多源运行，耗时 $|E|+|V|\log|V|$）。根据覆盖性质，每个顶点到其最近簇中心的图距离至多为 ℓ。

我们按如下方式根据 $\mathcal{C}$ 和 G 构造聚类图 $G^*(V^*,E^*)$。令 $V^*=\mathcal{C}$。如果 G 中的任意一条边连接 C_i 中的某个顶点与 C_j 中的某个顶点，我们就在 c_i 和 c_j 之间添加一条边，并将其加入 E^*。我们将 G 中这样一条边存储为 (c_i,c_j) 的见证边。注意，对于低密度图中的任意路径，都存在聚类图中的一条路径，使其相对于 ℓ"接近"。

观察 22。 对于任意路径 $\pi\in G$，存在一条路径 $\sigma\in G^*$，使得 $d_F(\pi,\sigma)\le\ell$。

该观察可由 r-网的覆盖性质直接得到，其中 $r=\ell$。

观察23。 对于任意路径 $\sigma\in G^*$，存在一条路径 $\pi\in G$，使得 $d_F(\pi,\sigma)\le\ell$。

证明。 回顾一下，G 中的每个顶点都根据 G中的图距离被分配给距离其最近的簇中心；此外，根据 ℓ-网的覆盖性质，G 中的每个顶点与其簇中心之间的图距离至多为 ℓ。另外，对于 G^* 中的每条边 (c_i,c_j)，我们存储一条见证边 $(a,b)\in E$，其中 $a\in C_i$ 且 $b\in C_j$。

设 (c_i,c_j) 是 σ 中的一条边，其见证边为 (a,b)。设 π_a 是 G 中从 c_i 到 a的一条最短路径，设 π_b 是 G 中从 b 到 c_j的一条最短路径。两条路径的长度均至多为 ℓ，因此 π_a中的每个点与 c_i之间的欧氏距离均为 ℓ 以内，而 π_b 中的每个点与 c_j之间的欧氏距离均为 ℓ 以内。我们将 G 中的游走 $\pi_a\circ ab\circ\pi_b$ 匹配到线段 c_ic_j，如下所示:

- π_a 的所有点都与 c_i匹配;

ab 边与c_ic_j线性匹配，因为 $\|a-c_i\|\le\ell$ 和 $\|b-c_j\|\le\ell$，欧氏范数的凸性意味着每个匹配对的距离至多为 ℓ;

π_b 的所有点都与c_j匹配。

这是一种单调匹配，其中每个匹配对的距离至多为 ℓ，因此 $d_F(\pi_a\circ ab\circ\pi_b, c_ic_j)\le\ell$。

将这些游走沿着 σ 的边连接起来，可得到 G中的一条游走 π。相邻的游走在一个簇中心处相交，该簇中心在一条片段的末尾与自身匹配，并在下一条片段的开头再次与自身匹配，因此这些匹配可以连接起来，并且 $d_F(\pi,\sigma)\le\ell$。 □

聚类后，图在某些连通性较高的区域中的密度可能会显著增加。为了限制与某个区域重叠的边数，我们对所得图进行剪枝。

阶段 2：剪枝聚类图此阶段的目的是减少图中位于较小区域内的高度连通分量（相对于参数 ℓ而言较小）。因此，我们可以限定剪枝后的图中任意区域内边的数量，该区域的大小与 ℓ相关；同时确保对于原图中的任意路径，剪枝后的图中都存在一条与之接近的路径。

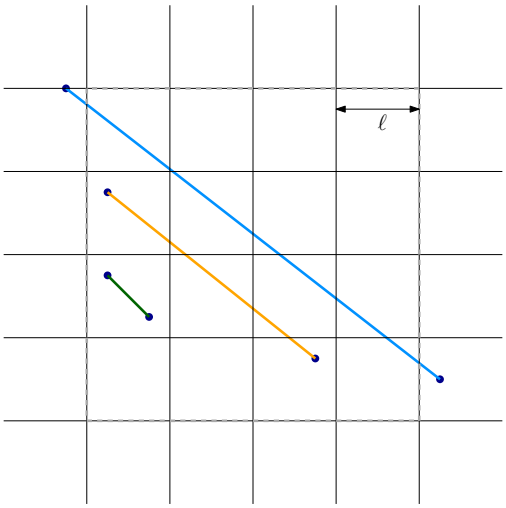

图 8： ℓ-网格示例。绿色边是局部边。橙色边是短跨越边。蓝色边是长跨越边。

我们先介绍一些术语。考虑一个覆盖在 λ-低密度图 G上的网格$\mathcal{G}$，其单元格大小为 ℓ。在第5节余下部分中，我们用 $\square_\tau$ 表示$\overbrace{\tau \times \tau \times \ldots \times \tau}$的一个块

d个单元可位于任意预先指定单元格大小的 d维网格中的任意位置。对于 $X \subseteq V^*$，令$G^*[X]$为由 X诱导的子图。令 $cell(v)$表示包含顶点 v的单元。我们将边集 E^* 划分为三种类型（各类型的示意图见图 8）：

局部边：若边$(u,\ v)$满足 $cell(u) = cell(v)$，则称其为局部边。

短跨越边：若边满足 $cell(u) \neq cell(v)$，且这些单元属于同一个 $\square_4$ 块，则称其为短跨越边。

长跨越边：若边满足 $cell(u) \neq cell(v)$，且这些单元不属于同一个 $\square_4$ 块，则称其为长跨越边。

最后，设 s 是网格的一个单元。若 G^* 的一个子图是 s 的极大连通分量，我们就称它为 s 的**单元分量**。对于顶点 $v \in V^*$（见图 9），用 $\tau(v)$ 表示其所在的单元分量。若 G^* 的一个子图是单元对 $s,s' \in \mathcal{G}$ 的极大连通分量 $G^*[V^* \cap (s \cup s')]$，并且 s 和 s' 位于同一个 $\square_4$ 块中（见图 10），我们就称它为这对单元的**对分量**。注意，s 或 s' 的每个单元分量都恰好包含在$\{s, s'\}$的一个对分量中。

现在可以介绍如何通过剪枝过程得到剪枝后的图 G'。该过程分为两个步骤。

步骤 1（单元）。对于每个包含顶点的单元 s，以及 s的每个单元分量 K，计算 K 的一棵生成树 T_K。删除 K 中不属于 T_K 的所有边。

步骤 2（共享一个 $\square_4$ 块的单元对）。对于每一对单元 $\{s,\ s'\}$，若它们位于同一个 $\square_4$ 块中，则令 $E^*_{s,\ s'}$ 为一条端点位于 s、另一端点位于 s' 的短跨越边集合。若 $E^*_{s,\ s'} = \emptyset$，则处理完成。否则，按如下方式构造多重图 $H_{s,\ s'}$。$H_{s,\ s'}$ 的节点是 s 和 s' 的单元分量，其中包含 $E^*_{s,\ s'}$中某条边的一个端点。对于每一条

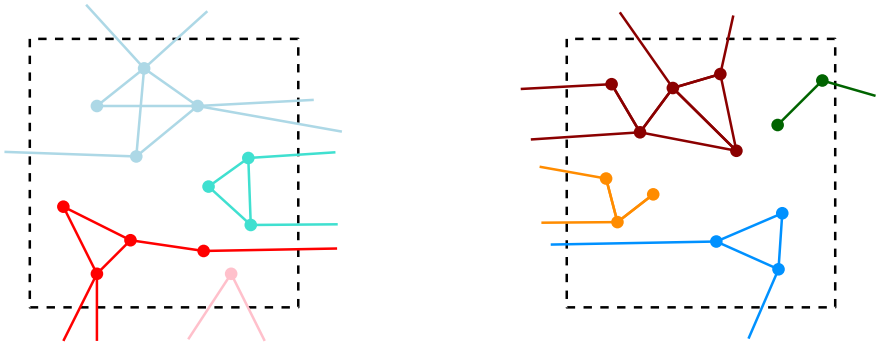

图 9： ℓ网格中的两个单元，其中 G 的每个单元分量在各自单元内都被赋予了不同的颜色。

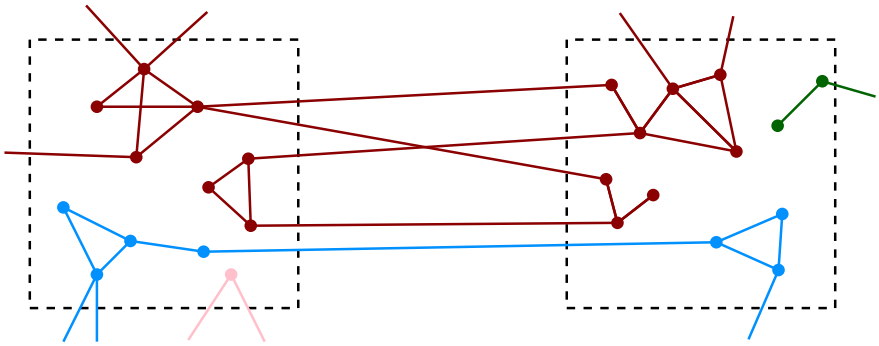

图 10： ℓ网格中位于同一个 $\square_4$内的一对单元。 G 的每个对分量都被赋予了不同的颜色。请注意，图 9 中的多个单元分量已合并为一个对分量。

边 $(u,\ v) \in E^*_{s,\ s'}$，在 $H_{s,\ s'}$ 中的 $\tau(u)$ 与 $\tau(v)$ 之间添加一条边。计算 $H_{s,\ s'}$ 的生成森林 $F_{s,\ s'}$。仅保留 $E^*_{s,\ s'}$ 中副本位于 $F_{s,\ s'}$ 内的边，并删除 $E^*_{s,\ s'}$ 的其他边。参见图 11。

我们不会删除 E^* 中的任何长跨越边。得到的剪枝后的图 G' 由顶点集 V^* 构成，其边集包括在步骤 1 和步骤 2 中保留的 E^* 的边，以及 E^* 中的所有长跨越边。

至此，简化图 G' 的构造完成。

我们作出以下观察，这些观察可由上述构造推出。由步骤 1 可知

观察24. 对于每个单元 s，图 $G'[V^* \cap s]$ 是一片森林，其各个分量的顶点集与 s 的单元分量具有相同的顶点集。

由步骤 2 可知

观察25。 对于位于同一 $\square_4$ 块中的任意单元对 $\{s, s'\}$，图 $G'[V^* \cap (s \cup s')]$ 是一片森林，其各连通分量的顶点集与 $\{s, s'\}$ 的成对分量具有相同的顶点集。

观察 26。 对于每条边 $(u, v) \in E^*$，存在一条从 u 到 v 的路径，该路径位于 G'中。若(u, v) 是一条局部边或短跨越边，则该路径仅使用 $cell(u) \cup cell(v)$中的顶点。

我们论证对于 G 中的任意路径，都存在 G' 中的一条路径，使其在以下方面“接近” ℓ.

引理 27。 对于任意路径 $\pi \in G$，都存在一条路径 $\sigma \in G'$，使得 $d_F(\pi, \sigma) \leq (1 + 4\sqrt{d})\ell$。

证明。我们首先证明，对于聚类图 G^*中的任意路径 π，都存在剪枝后的图 G' 中的一条路径 σ，使得 $d_F(\pi,\ \sigma) \leq 4\sqrt{d}\ell$。然后结合观察 22 得出引理。

设 π 为聚类图 G^*中的任意一条路径。根据网格单元边界，将 π 分解为一系列连续子路径 $\pi = \{\pi_1, \ldots\ldots, \pi_k\}$。每条子路径 π_i的起点和终点分别是 $V^* = V'$ 中的顶点 s_i 和 t_i，并且属于以下三种情况之一。

1. π 中完全位于同一个网格单元 s 内的极大连续子路径（注意，此类子路径完全由局部边组成）。
2. π中的一条短跨越边。
3. π中的一条长跨越边。

现在证明，给定 G^*中的任意路径 $\pi = \{\pi_1, \ldots\ldots, \pi_k\}$，我们可以在剪枝后的图 G' 中构造一条具有相同端点的路径 $\sigma = \{\sigma_1, \ldots\ldots, \sigma_k\}$。注意，由于 G^* 和 G' 由相同的顶点集合构成，且保持连通性（根据观察 26），这条路径始终存在。下面分别说明每种情况的构造方法。

情况 *1*：由于 π_i 是 $G^* \cap s$中的一条路径，因此 π_i 存在于 s的某个单元分量内。根据观察 24，该单元分量在 G'中被约简为一棵生成树，而该生成树也完全包含于 s中。因此，在 G' 中，s_i 与 t_i 之间存在唯一的、完全包含于 s中的简单路径 σ_i。我们将该路径记为 σ_i。由于两条路径都完全包含在边长为 ℓ的凸单元中（且 $V\prime = V^*$），因此可得 $d_F(\pi_i,\ \sigma_i) \leq \sqrt{d}\ell$。

情况 *2*：由于 s_i 和 t_i 属于位于同一个 $\square_4$ 块中的单元，因此短跨越边必然包含在一个对分量中。根据观察 25 s_i，t_i 通过 G' 中的一棵生成树内的一条路径连通，该路径仅使用 $cell(s_i) \cup cell(t_i)$中的顶点。将此路径记为 σ_i。由于 σ_i

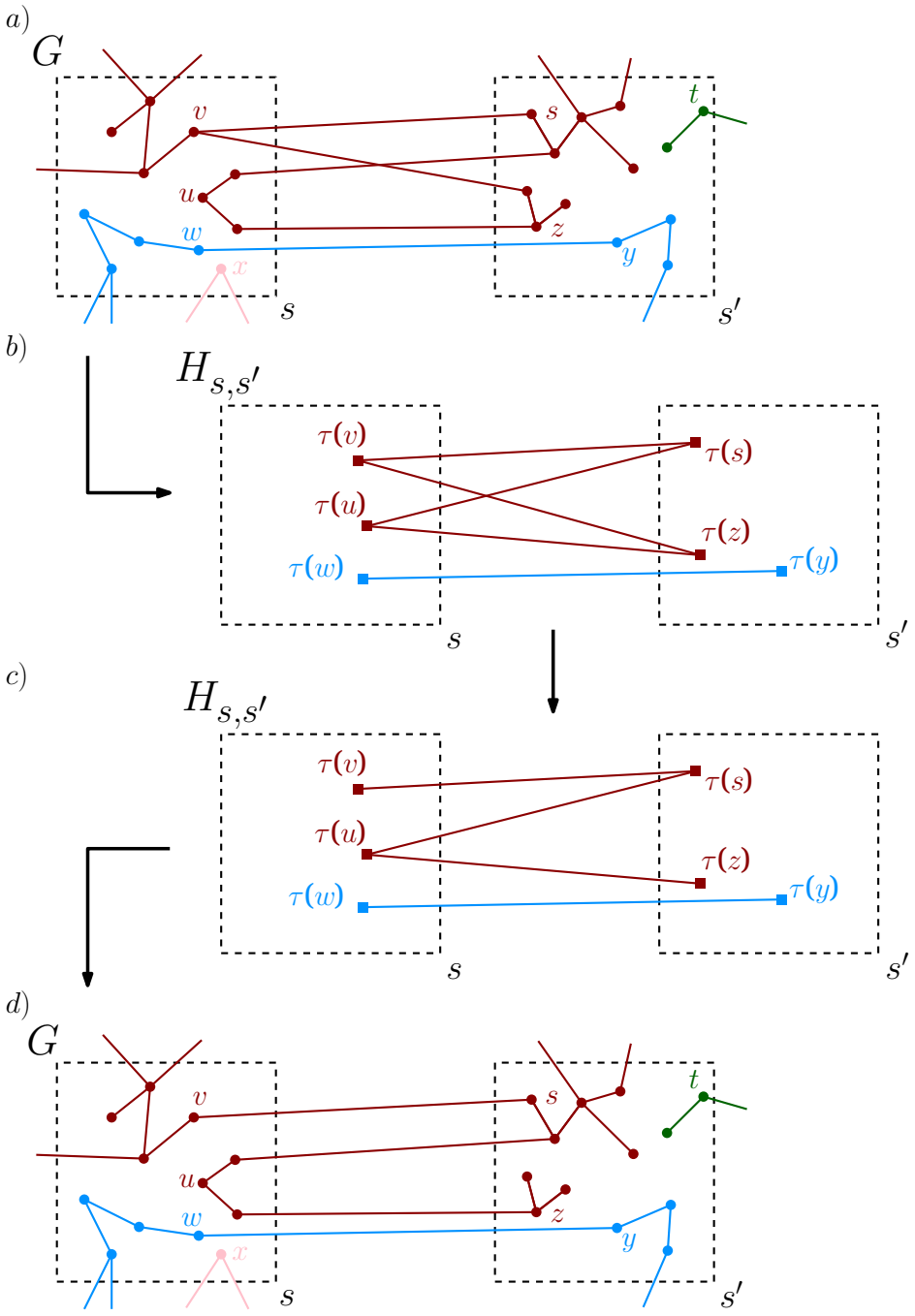

图 11：剪枝过程步骤 2 的可视化。

必须完全包含在一个 $\square_4$ 块中，便可以将 σ_i 的所有部分映射到 π_i 边上，且弗雷歇距离为 $4\sqrt{d}\ell$。

情况 *3*：在这种情况下，该边不可能在剪枝过程中被移除。因此，该边必定存在于 G'中。我们设 $\pi_i = \sigma_i$，这意味着在这种情况下，$d_F(\pi_i, \sigma_i) = 0$。

因此，在按照上述方式构造 σ 之后，$d_F(\pi, \sigma) \leq 4\sqrt{d}\ell$。

结合观察 22，即可完成该引理的证明。 □

在分析构造时间之前，我们首先证明 G' 满足性质 3。为此，我们证明，剪枝之后，与边长为 3δ的立方体相交的 G' 的边不会太多。

引理 28。设 $\ell = \frac{\varepsilon\delta}{1+4\sqrt{d}}$ 为G'的聚类参数。对于 $\mathbb{R}^d$ 中的任意边长为 δ的超立方体 C，与 $3C$ 相交的 G' 的边数至多为 $O\Big(\lambda\Big(\frac{|V\cap 27C|^{1-1/d}}{\varepsilon} + \frac{|V\cap 27C|^{1/d}}{\varepsilon^{d-1}}\Big)\Big)$。

证明。从 $7C$ 中的每个簇中心出发，沿着 G 中通往另一个中心的一条最短路径，取其前 $\ell/3$ 个单位。这样的路径总是存在，因为 G 是连通的。所取的路径前缀作为 G 的子集两两不相交：若它们有公共点 x，则两个中心 c, c' 之间会有 $d_G(c, c') \leq d_G(c, x) + d_G(x, c') \leq 2\ell/3$，这与它们的 ℓ- 分离性矛盾。由于 $\ell/3 < \delta$，这些路径都位于 $9C$ 中。令 k 为 $7C$ 中簇中心的数量。那么，G 中包含于 $9C$ 的边长总和至少为 $k \cdot \ell/3$。如果$k = 0$，则下文关于 k 的界显然成立。否则，$1 \leq k \leq |V \cap 27C|$，我们对 G 和立方体 $9C$ 应用引理20，对其三倍扩张$27C$ 中包含的 G 顶点数量给出下界，从而得到：

$$|V \cap 27C| = \Omega\left(\left(\frac{k\ell/3}{\lambda\delta}\right)^{1+\frac{1}{d-1}}\right).$$

因此，

$$k = O\Big(\lambda\frac{|V \cap 27C|^{1-1/d}}{\varepsilon}\Big).$$

为了界定与 $3C$ 重叠的 G' 的边数，我们将 E'_C 的边分为三类，并分别进行界定。

1. 在剪枝过程中被视为局部边 的 E'_C 的边。
2. 在剪枝过程中被视为短跨越边 的 E'_C 的边。
3. e E'_C 的边，在剪枝过程中被视为长跨越边 .

我们首先对 E'_C 中类型 1 和类型 2 的边数给出上界。

局部边的长度至多为 $\sqrt{d}\ell$，短跨越边的长度至多为 $4\sqrt{d}\ell < \delta$。因此，所有与 $3C$ 重叠的类型 1 或类型 2 的边，其两个端点都在 $5C \subseteq 7C$ 中。根据观察24，两个端点都在 $7C$ 中的局部边构成一片森林，至多包含 k 个顶点。根据观察25，对于同一 $\square_4$ 块中的每一对单元，它们之间的短跨越边构成一片森林；并且，每个单元与另外 O 个单元位于同一 $\square_4$ 块中。因此，E'_C 中此类边至多有 $O(k)$ 条。

接下来，我们对第 3 类边的数量给出上界。对于这一类型，我们给出两个界。首先注意，在聚类图中，所有第 3 类边的长度都必须至少为 3ℓ。要实现这一点，对于每条第 3 类边，原始 λ-低密度图 G 中必须存在一条长度至少为 ℓ 的唯一边，

因为对于任一端点，聚类最多只能将边长增加 ℓ 。我们将 G 中的这些原始边称为见证边。由于每个顶点到其簇中心的图距离都不超过 ℓ，边 (c_i,c_j) 的见证边 (a,b) 满足 $||a-c_i|| \le \ell$ 和 $||b-c_j|| \le \ell$；特别地， ab 与线段 c_ic_j 的距离不超过 ℓ ，因此与 $4C$ 相交。G' 中的不同边具有不同的见证边。

为得到第一个界，我们用边长为 ℓ 的 $= O(1/\varepsilon^d)$ 个立方体覆盖 $4C$ 的常数倍扩张 $O((\delta/\ell)^d)$。我们利用原图是 λ-低密度图这一事实，并结合事实 19，将此类边的数量界定为 $O(\lambda/\varepsilon^d)$。

第二个界将 G 中的见证边分为两组。第一组由所有端点位于 $27C$中的见证边组成。第二组由所有端点均不位于 $27C$中的见证边组成。对于第一组，由于原图是低密度图，我们知道原图的最大度将此类边的数量界定为 $O(\lambda \cdot |V \cap 27C|)$。第二组中的每条剩余边都必须足够长，使其在 G' 中的对应边与 $3C$ 重叠，同时两个原始顶点均位于 $27C$之外。因此，此类边的数量界定为 $O(\lambda)$。所以，我们的第二个界为$O(\lambda|V \cap 27C|)$。

取这两个界中较优的一个，并利用 $\min\{x,y\} \le x^\theta \cdot y^{1-\theta}$ with$\theta = (d-1)/d$ 这一事实，可将长跨越边的数量界定为

$$\min\{O(\frac{\lambda}{\varepsilon^d}), O(\lambda \cdot |V \cap 27C|)\} \le O\big(\lambda \cdot \frac{|V \cap 27C|^{1/d}}{\varepsilon^{d-1}}\big).$$

将局部边和跨越边的界与长跨越边的界结合起来，我们可以得出， G' 中与 $3C$ 重叠的边数至多为

$$O\Big(\lambda\Big(\frac{|V \cap 27C|^{1-1/d}}{\varepsilon} + \frac{|V \cap 27C|^{1/d}}{\varepsilon^{d-1}}\Big)\Big).$$

□

剩下的工作是分析构造时间，并论证 G'满足所有上述性质。

引理 29. 给定 $\mathbb{R}^d$中的一个 λ-低密度图 G ，可以构造一个满足上述性质 *1–3* 的简化图 G' ，其期望$(\lambda|V|\log |V|)$ 时间为 O。

证明。我们有 $|V^*| \le |V|$ 和 $|E^*| \le |E| \le \lambda|V|$。对图进行聚类需要$O(\lambda|V|\log |V|)$ 的期望时间（根据事实 21），此外，将顶点分配给中心还需要 $O(|E|+|V|\log |V|)$ 的时间。我们证明，剪枝需要 $O(|V^*|+|E^*|) = O(\lambda|V|)$ 的期望时间。

预处理。使用下取整函数获取顶点的单元索引；对于固定的 d，该操作耗时为常数。我们将非空单元的索引存储在一个最多包含 $|V^*|$ 个条目的哈希表中，并将每条边分配到其端点所在单元对对应的桶中。该操作的期望时间为 $O(|V^*|+|E^*|)$。

步骤 *1*。通过在每个非空单元上使用该单元对应桶中的局部边进行广度优先搜索，计算单元分量及其生成树。由于各单元互不相交，且每条边恰好位于一个桶中，因此该步骤耗时 $O(|V^*|+|E^*|)$。

步骤 *2*。对于单元对 $\{s,s'\}$ 及其桶 $X_{s,s'}$，多重图 $H_{s,s'}$ 至多包含 $2|X_{s,s'}|$ 个节点，并且恰好包含 $|X_{s,s'}|$ 条边。通过广度优先搜索可以在期望时间 $O(|X_{s,s'}|)$ 内构建该图并计算其生成森林（其中的期望来自使用哈希表识别节点）。各桶互不相交，因此步骤 2 的总期望时间为 $O(|E^*|)$。

现在我们证明 G' 的性质。根据引理 27，性质 1 得以满足；由于 G' 是 G^* 的子图，根据观察23，其逆命题部分也得以满足。根据 r-网的定义，以及剪枝不会缩短顶点之间路径长度这一事实，性质 2 得以满足。最后，根据引理 28，性质 3 得以满足。

□

我们现　在准备界定简化结构之间的自由空间复杂度。

5.3 界定近似自由空间复杂度

为了界定 λ-低密度图 G 与 λ-低密度曲线 P 之间的 $(1+\varepsilon)$-近似自由空间复杂度，我们首先构造简化结构 G' 和 P'。我们利用事实 8，将 P 简化为 $\varepsilon\delta$-简化结构 P'。然后，我们按照第 5.2 节中的描述，针对参数 $\ell = \frac{\varepsilon\delta}{1+4\sqrt{d}}$ 构造 G'。我们将 G' 称为 G 的 $(\varepsilon,\ \delta)$-简化结构。为了界定原始结构之间的 $(1+\varepsilon)$-近似自由空间，我们先界定简化结构 G' 和 P' 之间的自由空间。具体而言，我们证明 G' 与 P' 之间的自由空间复杂度至多为 $O\Big(\lambda^2\Big(\frac{(n|V|)^{1-1/d}}{\varepsilon^2} + \frac{n+|V|}{\varepsilon^d}\Big)\Big)$。然后，我们应用 [9] 中的现有技术，证明由此可以设计出一个 $(1+\varepsilon)$-近似判定器，该判定器能在 $O\Big(\lambda^2\Big(\frac{(n|V|)^{1-1/d}}{\varepsilon^2} + \frac{n+|V|}{\varepsilon^d}\Big)\log(n|E|)\Big)$ 时间内，解决 P 与 G 之间的地图匹配问题。

引理 30。设 $G=(V,E)$ 是一个连通的 λ-低密度图，且 P 是 $\mathbb{R}^d$中的一条复杂度为 n 的 λ-低密度曲线。设 $\delta>0$，且 $0<\varepsilon<1$。设 G' 是 G的一个 (ε,δ)-简化结构，且设P'是 P的一个 $(\varepsilon\delta)$-简化结构。相对于 δ，G'与 $P\prime$之间的自由空间复杂度（记为 $FS_\delta(P\prime G')$）至多为 $O\Big(\lambda^2\Big(\frac{(n|V|)^{1-1/d}}{\varepsilon^2} + \frac{n+|V|}{\varepsilon^d}\Big)\Big)$。

证明。我们如下界定自由空间图中具有自由空间的 (P', G') 的顶点-边对数量。[9] 的判定器的运行时间取决于自由空间曲面中单元的数量；每个单元分别对应于 P' 的一条边和 G'的一条边，并且这些边与相关自由空间相交。除包含起点的单元外，一个单元只能通过非空的自由边界到达；自由边界对应于一个结构的顶点与另一个结构的边，且二者之间的距离至多为 δ。P' 中一个顶点的边界至多由两个单元共享，而 G' 中一个顶点 v 的边界则由 $\deg_{G'}(v)$ 个单元共享。因此，我们将每个由 P'的一个顶点和 G'的一条边组成的对计数两次，并将每个由 G'的一个顶点 v和 $P\prime$ 的一条边组成的对计数 $\deg_{G'}(v)$ 次；所得的加权计数界定了 $FS_\delta(P', G')$。令

$$B = \lambda^2\Big(\frac{(n|V|)^{1-1/d}}{\varepsilon^2} + \frac{n+|V|}{\varepsilon^d}\Big).$$

我们根据每对中的顶点属于 P' 还是 G'，将这些对分为两类。我们将每一对归到包含该顶点的单元中收费。然后分别对每一类中的对数进行界定，并证明每一类至多包含 $O(B)$ 对。

考虑覆盖两个简化结构的 δ-网格。对于每个网格单元 C，令 $V_P(C)$ 和 $V_G(C)$ 分别表示与 C 重叠的 P 和 G 的顶点集合。类似地，令 $E_P(C)$ 和 $E_G(C)$ 分别表示与 C 重叠的 P 和 G 的边集合。回顾一下，xC 表示将 C 以其中心为基准按 x 倍缩放后得到的立方体。

我们首先考虑 $3C$ 中 G' 的边向 C 中 P' 的顶点收取的费用数量。注意，G' 中距离 $V_{P'}(C)$ 的某个顶点不超过 δ 的所有边都必须与 $3C$ 重叠。由引理 28 可得：

$$|E_{G'}(3C)| = O\Big(\lambda\Big(\frac{|V_G(27C)|^{1-1/d}}{\varepsilon} + \frac{|V_G(27C)|^{1/d}}{\varepsilon^{d-1}}\Big)\Big). \tag{3}$$

接下来，我们界定 P' 中位于 C内的顶点数。回想一下，P'的每条边（可能最后一条除外）长度至少为 $\varepsilon\delta$。对于 $V_{P'}$ 中除最后一个顶点外的每个顶点，我们将其后续子曲线 P 的前 $\varepsilon\delta/2$ 个单位长度计入该顶点。这些子曲线

参数不相交，且完全位于 $5C$内。我们可以假设 $V_{P\prime}(C) \neq \emptyset$；由于 $P\prime$的顶点也是 P的顶点，因此 $|V_P($ $15C)| \geq 1$。由于 P是 λ-低密度的，对 $5C$应用引理20可得

$$|V_P(15C)| = \Omega\left(\left(\frac{|V_{P'}(C)|\varepsilon\delta}{\lambda\delta}\right)^{1+\frac{1}{d-1}}\right).$$

因此，

$$|V_{P'}(C)| = O\Big(\lambda\frac{|V_P(15C)|^{1-1/d}}{\varepsilon}\Big). \tag{4}$$

利用方程 (3) 和 (4)，我们可以将 C 中由包含 P' 的一个顶点和 G' 的一条边的点对所产生的收费界定如下：

$$|V_{P'}(C)| \cdot |E_{G'}(3C)| =$$

$$O\left(\lambda^2\left(\frac{(|V_P(15C)| \cdot |V_G(27C)|)^{1-1/d}}{\varepsilon^2} + \frac{|V_P(15C)|^{1-1/d} \cdot |V_G(27C)|^{1/d}}{\varepsilon^d}\right)\right). \tag{5}$$

注意，扩展单元的重叠次数有界。因此， $\sum_C |V_P(15C)| = O(n)$，且$\sum_C |V_G(27C)| = O(|V|)$。利用 $2(1-1/d) \geq 1$ 这一事实，以及柯西 - 施瓦茨不等式和赫尔德不等式，可得

$$\sum_C (|V_P(15C)| \cdot |V_G(27C)|)^{1-1/d} = O((n|V|)^{1-1/d}),$$

and

$$\sum_C |V_P(15C)|^{1-1/d} \cdot |V_G(27C)|^{1/d} = O(n^{1-1/d}|V|^{1/d}) = O(n + |V|).$$

将这些界代入方程 (5)，我们得出结论：第一类点对的总数为 $O(B)$。

接下来，我们界定第二类点对的数量，其中顶点属于 $V_{G'}(C)$，边属于 $E_{P\prime}(3C)$。由方程 (3)，可得

$$\sum_{v \in V_{G'}(C)} \deg_{G'}(v) \leq 2 \cdot |E_{G'}(3C)| = O\Big(\lambda\Big(\frac{|V_G(27C)|^{1-1/d}}{\varepsilon} + \frac{|V_G(27C)|^{1/d}}{\varepsilon^{d-1}}\Big)\Big). \tag{6}$$

剩下要界定的是 P'中与 $3C$重叠的边数。回想一下， $P\prime$的每条边（可能最后一条除外）的长度至少为 $\varepsilon\delta$。不妨设 $\varepsilon \leq 1/4$。设 $e' = p_ip_j$ 是 $E_{P\prime}(3C)$中除最后一条外的一条边，令 $x \in e' \cap 3C$，并设 $\sigma \subseteq e'$ 是一个包含 x、长度为 $\min\{|e'|, \delta\}$ 的子线段；则 $\sigma \subseteq 5C$。由于 $d_F(P[p_i, p_j], e')$ $\leq \varepsilon\delta \leq \delta/4$，线段 σ与 $P[p_i, p_j]$的一条位于 $6C$中的子曲线相匹配。若 $\sigma = e'$，该子曲线就是 $P[p_i, p_j]$本身，其长度至少为 $|e'| \geq \varepsilon\delta$；否则其长度至少为 $\delta - 2\varepsilon\delta \geq \delta/2 \geq \varepsilon\delta/2$。因此， P中与 e' 对应的子曲线至少为 $6C$贡献 $\varepsilon\delta/2$ 。

我们将此贡献的前 $\varepsilon\delta/4$ 个单位计入每条对应的子曲线。由于 $P\prime$的不同边表示 P中参数不相交的子曲线，因此这些计入的部分彼此参数不相交。由此可知， P在 $6C$ 内的总长度至少为$(|E'_P($ $3C)| - 1) \cdot \varepsilon\delta/4$。对 $6C$ 应用引理20可得，若 $V_P(18C) \neq \emptyset$,

$$|V_P(18C)| = \Omega\left(\left(\frac{|E_{P'}(3C)|\varepsilon\delta}{\lambda\delta}\right)^{1+\frac{1}{d-1}}\right).$$

因此，

$$|E_{P'}(3C)| = O\Big(\lambda \frac{|V_P(18C)|^{1-1/d}}{\varepsilon}\Big). \tag{7}$$

如果 $V_P(18C) = \emptyset$，那么引理20 的第一条陈述给出的是 $|E_{P'}(3C)| = O(\lambda/\varepsilon)$。只有满足 $V_{G'}(C) \neq \emptyset$ 的立方体才会产生第二类收费，且这样的立方体至多有 $|V|$ 个，因此根据公式 (6)，这些立方体总共会产生 $O(\frac{\lambda^2}{\varepsilon} \sum_C ((\frac{|V_G(27C)|}{\varepsilon})^{1-1/d} + \frac{|V_G(27C)|^{1/d}}{\varepsilon^{d-1}}))$ = ~~$O(B)$。对于其余立方体，~~公式 (7) 成立。

结合公式 (6) 和公式 (7)，我们得到收费总数的以下上界。

$$|E_{P'}(3C)| \cdot \sum_{v \in V_{G'}(C)} \deg_{G'}(v) = O\Big(\lambda^2 \frac{|V_P(18C)|^{1-1/d}}{\varepsilon} \cdot \Big(\frac{|V_G(27C)|^{1-1/d}}{\varepsilon} + \frac{|V_G(27C)|^{1/d}}{\varepsilon^{d-1}}\Big)\Big).$$

展开该等式，使用柯西-施瓦茨不等式和赫尔德不等式，并对所有 $x \geq 1$ 使用 $x^{1-1/d} \leq x$，可得

$$\sum_C |E_{P'}(3C)| \cdot \sum_{v \in V_{G'}(C)} \deg_{G'}(v) = O(B).$$

因此，我们得出第二类配对总共贡献了 $O(B)$ 个配对。

注意，每个原始结构中的顶点至多只能被收费常数次，因为它只属于常数个网格单元的固定扩展。因此，两个简化结构之间的自由空间复杂度至多为

$$O(B) = O\Big(\lambda^2 \Big(\frac{(n|V|)^{1-1/d}}{\varepsilon^2} + \frac{n + |V|}{\varepsilon^d}\Big)\Big),$$

如前所述。 □

对于 $d = 2$ 维，利用不等式 $\sqrt{n|V|} \leq \frac{n+|V|}{2}$，~~近似自由空间复杂度界可简化为~~ $O(\frac{\lambda^2}{\varepsilon^2}(n + |V|))$。

—

5.4 一个$(1+\varepsilon)$-近似判定器

令 $FS_\delta(P, G)$ 表示自由空间曲面中相对于参数 δ、位于 P 和 G 之间且具有相关自由空间的单元。[9] 的作者证明，对于复杂度为 n 的多边形曲线 P，以及在 $\mathbb{R}^d$ 中具有 $|V|$ 个顶点和 $|E|$ 条边的图G，存在一个算法 Decider(P, G, δ)，能够在 $O(|V| + FS_\delta(P, G) \log(n|E|))$ 时间和$O(|V| + FS_\delta(P, G))$ 空间内，判定是否存在一条路径 $\pi \in G$，使得 $d_F(\pi, P) \leq \delta$。若存在这样的 π，该算法也会将其返回。我们将此判定器应用于这些简化数据结构，从而得到以下引理。

引理 *31.* *设 P 是复杂度为 n 的多边形曲线，$G = (V, E)$ 是复杂度为 $|V| + |E|$ 的任意图。设 $\delta > 0$ 和 $0 < \varepsilon < 1$。设 G' 是 G 的一个 (ε, δ)- 简化，且 P' 是 P 的一个 $(\varepsilon\delta)$- 简化。存在一个名为 **ApproxDecider** 的算法，可在 $O(|E| + FS_{(1+2\varepsilon)\delta}(P', G') \log(|E|n))$ 时间内判定是否存在一条路径 $\pi \in G$，使得 $d_F(\pi, P) \leq (1 + 4\varepsilon)\delta$；或者对于所有路径 $\pi \in G$，$d_F(\pi, P) > \delta$。*

证明。令 $\delta_1 = (1+2\varepsilon)\delta$。近似判定器运行 Decider$(P' G', \delta_1)$ 并返回其答案，耗时 $O(|E| + FS_{\delta_1}(P' G') \log(|E|n))$。若存在一条路径 $\pi \in G$ with $d_F(\pi, P) \leq \delta$，则根据性质 1，存在一条路径 $\sigma \in G'$ with $d_F(\pi, \sigma) \leq (1 + 4\sqrt{d})\ell = \varepsilon\delta$，且根据事实 8，$d_F(P, P') \leq \varepsilon\delta$，因此 $d_F(\sigma, P') \leq \delta_1$，判定器回答是。反过来，若判定器返回一条路径 $\sigma \in G'$with $d_F(\sigma, P') \leq \delta_1$，则根据观察23（由于 G' 是 G^* 的子图），存在一条路径 $\pi \in G$ with$d_F(\pi, \sigma) \leq \ell \leq \varepsilon\delta$，且 $d_F(\pi, P) \leq d_F(\pi, \sigma) + d_F(\sigma, P') + d_F(P' P) \leq (1 + 4\varepsilon)\delta$。

□

我们结合引理 29、30 和 31 的结果，得出定理 18。我们将引理 31 应用于 $\varepsilon/4$，并在下文中用 ε 表示 $\varepsilon/4$。接下来需要界定 $FS_{\delta_1}(P', G')$，其中$\delta_1 = (1+2\varepsilon)\delta$。令 $\varepsilon_1 = \varepsilon\delta/\delta_1 \in [\varepsilon/3, \varepsilon]$。这些简化仅取决于乘积$\ell = \frac{\varepsilon\delta}{1+4\sqrt{d}} = \frac{\varepsilon_1\delta_1}{1+4\sqrt{d}}$和 $\varepsilon\delta = \varepsilon_1\delta_1$，因此 G'也是 G 的 $(\varepsilon_1, \delta_1)$-简化，而 P'是 P 的 $(\varepsilon_1\delta_1)$-简化。将引理 30 应用于 δ_1 和 ε_1，可得 $FS_{\delta_1}(P'G') = O(B)$。构造$P'$需要 $O(n)$ 时间，构造 G'需要期望 $O(\lambda|V|\log|V|)$ 时间；由于 $\lambda \geq 1$、$|E| \leq \lambda|V|$和 $\log|V| = O(\log(n|E|))$，这两项以及引理 31 中的 $O(|E|)$ 项都被$O(B\log(n|E|))$ 所支配。

6 从判定器到近似算法

在本节中，我们将定理 4 和 18 中的判定器转化为问题 1 的近似算法。回顾一下，P 是一条具有 n 个顶点的多边形曲线，$G = (V, E)$是一个连通图，而 $r^* = \min_\pi d_F(P, \pi)$是问题 1 的最优值。由于这些判定器只能将 r^* 与给定半径进行近似比较，因此我们需要一个候选半径列表来进行搜索。

定理 32。给定一条曲线 P 和一个图 $G = (V, E)$，以及固定常数 ε 和 d，可以在运行时间为 $\tilde{O}(cn + c^2|E|)$ 的情况下，计算出 $r^* = \min_\pi d_F(P, \pi)$ 的 $(1+\varepsilon)$-近似值，其中曲线 P 是 *c-packed*；如果曲线 P 是*λ-low-dense* 且图 G 是 *λ-low-dense*，则期望运行时间为 $\tilde{O}(\lambda^2((n|V|)^{1-1/d} + n + |V|))$。

证明。我们遵循 Driemel、Har-Peled 和 Wenk [15, 节的方法。具体而言，我们先将模糊判定器转化为完整判定器，再用它搜索顶点距离，以确定 P 和 G的简化，最后在这些简化上搜索临界值。

步骤 1：完整判定器。定理 4 和 18 中的判定器是模糊的：对于半径 s和精度 $\varepsilon/8$，若结果为是，则可确认 $r^* \leq (1+\varepsilon/8)s$；若结果为否，则可确认 $r^* > s$。我们希望对候选半径进行二分搜索，以找到 r^*；但当 r^* 接近 s时，这两种结果都无法告诉我们最优值位于 s 的哪一侧。因此，我们构造一个完整判定器，给定 $t > 0$后，它会返回以下结果之一：(i) $r^* \leq t$、(ii) $r^* > t$，或 (iii) r^*的 $(1+\varepsilon)$-近似。

为此，我们在 $t/(1+\varepsilon/8)$ 和 t [19, 第 6]节中运行模糊判定器。如果第一次调用返回是，我们返回 (i)。如果第二次调用返回否，我们返回 (ii)。否则 $t/(1+\varepsilon/8) < r^* \leq (1+\varepsilon/8)t$，因此$(1+\varepsilon/8)t \leq (1+\varepsilon/8)^2 r^* \leq (1+\varepsilon)r^*$，并将 $(1+\varepsilon/8)t$ 作为 (iii) 返回。每当完整判定器返回 (iii) 时，我们就停止。

步骤 2：求简化结果。我们使用完整判定器，对由顶点距离导出的候选半径进行二分搜索。目标是获得 (iii)，或者找到两个连续的候选半径 $L \leq r^* \leq U$，使得没有顶点距离位于 L 的一个小分数与 U 的一个大倍数之间。令 $N = n + |V|$。根据欧几里得 WSPD，我们得到 $O(N)$个值 z，使得输入顶点之间的每个正距离都属于 $[z/2, 2z]$ 中的某个区间，具体见z [9, 引理 3。8]。对于每个 z，我们按几何方式对 $[z/16, 32z/\varepsilon]$ 进行采样，包括两个端点，且连续比值至多为 $1+\varepsilon/8$。我们将所有这些值放在一起进行二分搜索，并以零和包围盒对角线作为初始边界。

在调用完整判定器 $O(\log N)$次后，我们要么通过返回 (iii) 停止，要么得到连续的值 $L \leq r^* \leq U$。如果 $U \leq (1+\varepsilon/8)L$，我们返回 U，因为 $U \leq (1+\varepsilon/8)r^*$。否则$U > (1+\varepsilon/8)L$，因此不存在与 (L, U) 重叠的区间 $[z/16, 32z/\varepsilon]$：其样本之间的距离至多相差一个因子$1+\varepsilon/8$，且没有任何样本严格位于连续值 L 和 U 之间。因此，对于每个 z，都有$32z/\varepsilon \leq L$ 或 $z/16 \geq U$。由于每个顶点距离都属于某个 $[z/2, 2z]$，所以它要么至多为 $2z \leq \varepsilon L/16$，要么至少为 $z/2 \geq 8U$。我们利用这一性质来得到所需的简化。

现在，我们通过对 P 和 G 的顶点进行聚类来简化它们。我们对 P 和 G 的输入顶点进行聚类，使得两个顶点属于同一个簇，当且仅当它们之间的距离至多为 $\varepsilon L/16$。这种聚类确实存在：如果 u, v 和 v, w 之间的距离至多为 $\varepsilon L/16$，那么 u, w 之间的距离至多为 $\varepsilon L/8 < 8U$，因此根据上述性质，至多为 $\varepsilon L/16$。利用网格，我们可以在 $O(N)$ 时间内计算出~个簇。我们为每个簇固定一个簇中心，并将每个顶点移动到其所在簇的中心。简化曲线 P 是由此得到的曲线，其中删除了连续重复的顶点。¯简化图 G 是由此得到的图，其中收缩了簇内的每条边，因此 G 的每个¯顶点都替代了 G 在一个簇中的一个连通顶点集。

该简化使弗雷歇距离至多改变 $\varepsilon L/16$，因为每个顶点移动的距离至多为¯ $\varepsilon L/16$，而 G 中的每条路径都可通过聚类提升为 G 中的一条路径。在本证明余下部分中，我们使用 P和 G。令 $\bar{r} = \min_{\bar{\pi}} d_F(P, \bar{\pi})$表示 P与 G中距其最近的路径 $\bar{\pi}$ 之间的弗雷歇距离。注意 $|\bar{r} - r^*| \le \varepsilon L/8$。

步骤 3：搜索临界值。 我们还需要近似 $\bar{r}$。为此，我们找出一个包含 $\bar{r}$的临界值列表。根据 Alt 和 Godau 的临界值论证 [2]（地图匹配见 [1, 第 2.6] 节，事件说明见 [15, 第 2.2.1] 节），$\bar{r}$ 是以下情况之一：(a) 两个端点之间的距离；(b) 一个顶点与一条线段之间的距离；或 (c) 两个顶点到一条线段上某个公共点的距离。我们证明情况 (a) 和 (c) 都可归约为情况 (b)。由于简化顶点之间的距离至少为 $8U$ $\bar{r} \le r^* + \varepsilon L/8 < 2U$，任意两个距离至多为 $2\bar{r}$ 的简化顶点都重合。在情况 (a) 中，这两个顶点之间的距离为 $\bar{r}$，因此它们重合，且 $\bar{r} = 0$。在情况 (c) 中，这两个顶点之间的距离至多为 $2\bar{r}$，因此它们重合，而 $\bar{r}$ 是该顶点与线段之间的距离。因此，我们只需考虑情况 (b)，即 $\bar{r}$ 是一个简化顶点与一条线段之间的距离。

我们如何搜索 $\bar{r}$ 取决于输入模型。对于低密度曲线和图，我们在可达自由空间中对顶点–线段距离进行二分搜索。对于 c-打包曲线，我们按照 [18, 第2节2], 中的方法，判定一个有向图是否包含有向路径，并对半径进行二分搜索。

低密度曲线和图。我们枚举半径为 $2U$时的可达自由空间单元，并收集情形 (b) 中出现在这些单元边上的顶点–线段距离。我们只保留不超过 $2U$的顶点–线段距离；这些距离包括上文所示的 $\bar{r}$。我们使用 Chen 等人的精确判定器 [9], 对这些距离进行二分搜索；该判定器适用于任意曲线和图（见第5.4节），从而得到 $w = \bar{r}$。由于 $|\bar{r} - r^*| \le \varepsilon L/8$ 和 $L \le r^*$，我们有 $r^* \le w + \varepsilon L/8 \le r^* + \varepsilon L/4 \le (1 + \varepsilon)r^*$，因此 $w + \varepsilon L/8$ 是 r^*的（$1 + \varepsilon$）-近似，符合要求。

最后，我们分析运行时间。每次调用步骤 1 的完整判定器都会调用定理 18 的判定器两次。在步骤 2 中，我们进行 $O(\log N)$ 次此类调用，这些调用的总期望时间为 $\widetilde{O}(\lambda^2((n|V|^{1-1/d} + n + |V|))$，计算候选半径和~聚类需要 $O(N)$ 时间。在步骤 3 中，枚举以及每次调用精确判定器所需的时间，都与 $n + |V|$ 加上可达单元的数量近似线性相关。为界定单元的数量，我们将每个单元归因于其两条边中较短的一条，其长度为 ℓ。另一条边来自 P 或 G，其长度至少为 $\ell/2$，并且与以较短边的中点为中心、半径为 2ℓ 的球相交。低密度条件至多允许 $O_d(\lambda)$条此类边，因此每条边被归因 $O_d(\lambda)$次；由于 $|E| \le \lambda|V|$，单元总数为$O_d(\lambda(n + |E|)) = O_d(\lambda^2(n + |V|))$。因此，期望运行时间为 $\widetilde{O}(\lambda^2((n|V|^{1-1/d} + n + |V|))$。

~~P 与 P 的 $\varepsilon L/16$ 简化曲线。对于每个顶点 v 的~~ 条，因为我们使用引理 10 来找到 P 仍然是 $O(c)$- 紧密的 [15, 引理4.3]。接下来，我们构建一个有向图：对于引理10报告的、顶点 v 在 G 上的每个投影 ¯ ¯，都创建一个节点 (v, x)；其中 x 是它在 P 上的参数。对于¯ 的每条有向边 uv 和满足 $x \le y$ 的节点 (u, x)、(v, y)，我们使用事实 13 [14] 计算一个

¯(1 + ε/8)-近似 D'，用于 $D = d_F(P[x, y], uv)$。给定半径 t，如果 $D' \le t$，我们就插入一条从 (u, x) 指向 (v, y) 的有向边，并使用广度优先搜索判断是否存在一条有向路径¯，从一个节点 (v, x) 出发，该节点的顶点 v 与 P 的第一个顶点重合，且其参数 x 是 P 的起点；路径到达的节点，其顶点与 P 的最后一个顶点重合，且其参数¯是 P 的终点。我们对值 D' 进行二分搜索，找出存在这样一条 ¯ ¯路径时最小的 $t = w$。沿这条路径拼接各匹配，即可得到 P 与 G 中一条路径之间的匹配，其中每一对匹配点的距离至多为 w，因此 $w \ge \bar{r}$。此外， $w \le (1 + \varepsilon/8)\bar{r}$，因为在¯一个最优匹配中，我们可以将 P 上与 $\bar{\pi}$ 的每个顶点配对的点移至该顶点的投影处，因为与 $\bar{r}$ 相比，简化顶点彼此相距很远。所得匹配将 $\bar{\pi}$ 的每个顶点 v 与一个点 x 配对，使得 (v, x) 是一个节点；在任意两个连续的此类节点之间，它都有$D \le \bar{r}$，因此这些节点构成一条有向路径，对应于 $t = (1 + \varepsilon/8)\bar{r}$。

因此， w 是 $\bar{r}$的（$1 + \varepsilon/8$）近似。由于 $|\bar{r} - r^*| \le \varepsilon L/8$ 和 $L \le r^*$，我们有$r^* \le w + \varepsilon L/8 \le$ （$1 + \varepsilon/8$）（$r^* + \varepsilon L/8$）$+ \varepsilon L/8 \le$（$1 + \varepsilon$）r^*，所以 $w + \varepsilon L/8$ 是 r^*的（$1 + \varepsilon$）近似，正如所要求的。

最后，我们分析运行时间。步骤 1 中对完整判定器的每次调用都会对定理 4 中的判定器进行两次调用。在步骤 2 中，我们进行 O（log N）次此类调用，总计耗时 $\widetilde{O}$（$cn + c^2|E|$）¯，而计算候选半径和聚类需要 O（N）时间。在步骤 3 中，有 O（$c^2|E|$）条有向边，计算每个值 D' 需要多对数时间，而二分搜索会调用对数次数的广度优先搜索。因此，运行时间为 $\widetilde{O}$（$cn + c^2|E|$）。

□

7 结论

本文提出了两种用于地图匹配问题的算法，放宽了 [9] 对输入图和曲线所作的结构假设。首先，我们证明，仅假设曲线具有 c-紧致性，就足以获得次二次时间近似算法，而无需对图作任何假设。这将 [18] 和 [12] 的结果扩展到了地图匹配问题。其次，对于图和曲线都满足 λ-低密度的情形，我们获得了一种次二次时间近似算法。这将 [15] 针对两条 λ-低密度曲线的结果扩展到了地图匹配问题。这两项结果都改进了 [9]：第一项移除了对图 G 的假设，第二项则推广了对曲线 P 的假设。

最后，我们提出两个开放问题。首先，对于将一条 c-紧致曲线匹配到一般图的问题，我们能否将对 c 的依赖改进为线性？当前的 c^2 因子源于在有向图 H 中搜索一条 有效的路径。因此，一个问题是，这一部分能否得到改进，或者是否可以通过新的方法加以改进。一种看似自然的方法，是扩展 [12]，的作者所取得的结果；他们设计了次二次时间近似算法，用于在仅有一条曲线具有 c-紧致性时计算两条曲线之间的弗雷歇距离。这些作者采用了不同于 [18]，的方法，并证明如果弗雷歇距离接近某个给定的猜测值，则自由空间复杂度必然很小。为论证这一点，他们依赖于这样一个事实：两个数据结构必须彼此进行完全且单调的映射。他们利用这一事实论证，一般数据结构中的长边必须满足某种近似 c-紧致性，才能与 c-紧致曲线足够接近。然而，在地图匹配中，并不需要将整个图映射到曲线上；相反，只要求图中的一条路径映射到曲线上。因此，除非该路径已知，否则这种方法无法直接扩展到地图匹配问题。由此，在一般图与 c-紧致曲线之间界定自由空间复杂度可能并不可行。

其次， λ-低密度对于单侧地图匹配而言是否是一个足够强的假设？在

相关文献中， c-紧致图经常被用作获得 λ-低密度图良好界限的第一步。因此，一个自然的问题是：类似于 c-紧致性的情形，我们是否可以去除图上的 λ-低密度假设，或者说， λ-低密度对于地图匹配中的单侧结果而言是否不是一个足够强的假设。关于c-紧致性的单侧地图匹配研究，很大程度上受到了计算曲线之间弗雷歇距离时关于单侧 c-紧致性的研究结果的启发。因此，一个自然的第一步是考察：当仅有一条曲线满足 λ-低密度时，是否能够设计出用于近似两条曲线之间弗雷歇距离的次二次时间算法。

8 AI 声明

本文中的所有观点均由作者独立提出，未借助大型语言模型。仅在最后阶段使用 Claude 5.5 辅助撰写、润色和查找错误。

参考文献

[1] Helmut Alt、Alon Efrat、Günter Rote 和 Carola Wenk。平面地图匹配。《算法杂志》，49(2):262–283，2003。 doi:10.1016/S0196-6774(03)00085-3。[2] Helmut Alt 和 Michael Godau。计算两条多边形曲线之间的弗雷歇距离。《国际计算几何与应用杂志》，5:75–91，1995。 doi:10.1142/S0218195995000064。[3] Sotiris Brakatsoulas、Dieter Pfoser、Randall Salas 和 Carola Wenk。关于车辆跟踪数据的地图匹配。载于第*31*届超大型数据库国际会议（*VLDB*）论文集，第853–864页。ACM，2005。[4] Karl Bringmann。为什么遛狗需要时间：除非 SETH 不成立，否则弗雷歇距离不存在强次二次算法。载于第*55*届 *IEEE* 计算机科学基础年会（*FOCS*）论文集，第661–670页。IEEE计算机学会，2014。doi:10.1109/FOCS.2014.76。[5] Kevin Buchin、Maike Buchin、Joachim Gudmundsson、Aleksandr Popov 和 Sampson Wong。低密度伸展子图上基于弗雷歇距离的地图匹配查询。载于第*40*届计算几何国际研讨会（*SoCG*）论文集，*LIPIcs*第293卷，第27:1–27:15页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2024。 doi:10.4230/LIPICS.SOCG.2024.27。[6] Kevin Buchin、Tim Ophelders 和 Bettina Speckmann。SETH 表明：弱弗雷歇距离更快，但仅当它是连续的且处于一维情况下才如此。载于 Timothy M. Chan 编，第第三十届 *ACM-SIAM* 离散算法研讨会论文集，*SODA 2019*，美国加利福尼亚州圣迭戈，*2019*年*1*月*6*日至*9*日，第2887–2901页。SIAM，2019。 doi:10.1137/1.9781611975482.179。[7] Erin W. Chambers、Brittany Terese Fasy、Yusu Wang 和 Carola Wenk。使用最短路径进行地图匹配。《*ACM*空间算法与系统汇刊》，6(1):6:1–6:17，2020。 doi:10.1145/3368617。[8] Pingfu Chao、Yehong Xu、Wen Hua 和 Xiaofang Zhou。地图匹配算法综述。载于第*31*届澳大拉西亚数据库会议论文集，*ADC 2020*，《计算机科学讲义》第12008卷

计算机科学讲义，第121–133页。施普林格，2020。 doi:10.1007/978-3-030-39469-1_10。[9] Daniel Chen、Anne Driemel、Leonidas J. Guibas、Andy Nguyen 和 Carola Wenk。关于弗雷歇距离的近似地图匹配。载于第13届算法工程与实验研讨会（ALENEX）论文集，第75–83页。SIAM，2011。 doi:10.1137/1.9781611972917.8。[10] Daniel Chen、Christian Sommer 和 Daniel Wolleb。基于顶点单调弗雷歇距离的快速地图匹配。载于 Matthias Müller-Hannemann 和 Federico Perea 编，第21届交通建模、优化与系统算法方法研讨会，ATMOS 2021，葡萄牙里斯本（线上会议），2021年9月9日至10日，OASIcs第96卷，第10:1–10:20页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2021。URL： https://doi.org/10.4230/OASIcs.ATMOS.2021.10， doi:10.4230/OASICS.ATMOS.2021.10。[11] Jacobus Conradi、Ivor van der Hoog、Frederikke Uldahl 和 Eva Rotenberg。低密度曲线的紧弗雷歇界，2026。URL： https://arxiv.org/abs/2604.24135， arXiv:2604.24135。[12] Jacobus Conradi、Ivor van der Hoog、Thijs van der Horst 和 Tim Ophelders。当仅有一条曲线为紧致时计算弗雷歇距离：一种简单且近乎紧致的算法。载于 Sepehr Assadi 和 Eva Rotenberg 编，算法简洁性研讨会（SOSA）论文集，第43–55页。SIAM，2026。 doi:10.1137/1.9781611978964.4。[13] Gregor Diatzko、Félix Lasseux 和 Sabine Storandt。几何图中的紧致性与低密度：理论与实践，2026。URL： https://arxiv.org/abs/2609.27231， arXiv:2609.27231。[14] Anne Driemel 和 Sariel Har-Peled。遛狗抄近路：使用捷径计算弗雷歇距离。SIAM计算杂志，42(5):1830–1866，2013。[15] Anne Driemel、Sariel Har-Peled 和 Carola Wenk。在线性近似时间内近似现实曲线的弗雷歇距离。离散计算几何，48(1):94–127，2012。 doi:10.1007/S00454-012-9402-Z。[16] David Eppstein、Sariel Har-Peled 和 Anastasios Sidiropoulos。图度量的近似贪心聚类与距离选择。《计算几何杂志》，11(1):629–652，2020。 doi:10.20382/JOCG.V11I1A25。[17] Joachim Gudmundsson、Zijin Huang 和 Sampson Wong。近似低密度值。载于 Weili Wu 和 Guangmo Tong 编，第29届计算与组合国际会议（COCOON）论文集，《计算机科学讲义》第14422卷，第71–82页。施普林格，2023。 doi:10.1007/978-3-031-49190-0_5。[18] Joachim Gudmundsson、Tiancheng Mai 和 Sampson Wong。当仅有一条曲线为紧致时近似弗雷歇距离。载于第35届算法与计算国际研讨会（ISAAC）论文集，《LIPIcs》第322卷，第37:1–37:14页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2024。 doi:10.4230/LIPICS.ISAAC.2024.37。[19] Joachim Gudmundsson、Martin P. Seybold 和 Sampson Wong。在弗雷歇距离下现实输入图上的地图匹配查询。《ACM算法汇刊》，20(2):14，2024。doi:10.1145/3643683。

[20] Joachim Gudmundsson 和 Michiel H. M. Smid。几何树中近似弗雷歇匹配查询的快速算法。计算几何，48(6):479–494，2015。 doi:10.1016/J.COMGEO.2015.02.003。Sariel Har-Peled。几何近似算法。美国数学会，2011。[22] Mahdi Hashemi 和 Hassan A. Karimi。实时地图匹配算法的批判性综述：当前问题与未来方向。计算机、环境与城市系统，48:153–165，2014。 doi:10.1016/j.compenvurbsys.2014.07.009。Matej Kubicka、Arben Çela、Hugues Mounier 和 Silviu-Iulian Niculescu。车辆地图匹配方法的比较研究与面向应用的分类。*IEEE*智能交通系统杂志，10(2):150–166，2018。 doi:10.1109/MITS.2018.2806630。Mohammed A. Quddus、Washington Y. Ochieng 和 Robert B. Noland。交通应用的当前地图匹配算法：现状与未来研究方向。《交通研究*C*辑：新兴技术》，15(5):312–328，2007。[25] Otfried Schwarzkopf 和 Jules Vleugels。低密度环境中的范围搜索。信息处理快报，60(3):121–127，1996。 doi:10.1016/S0020-0190(96)00154-8。[26] Hong Wei、Yin Wang、George Forman 和 Yanmin Zhu。地图匹配：稀疏且含噪数据下各种方法的比较。载于第*21*届地理信息系统进展国际会议（*SIGSPATIAL*）论文集，第434–437页。ACM，2013。doi:10.1145/2525314.2525456。[27] Carola Wenk、Randall Salas 和 Dieter Pfoser。满足地图匹配速度需求：全局曲线匹配算法的局部化。载于第*18*届科学与统计数据库管理国际会议（*SSDBM*）论文集，第379–388页。IEEE计算机学会，2006。 doi:10.1109/SSDBM.2006.11。[28] Yu Zheng 和 Xiaofang Zhou，编。空间轨迹计算。施普林格，2011。doi:10.1007/978-1-4614-1629-6。