HiFIVE：面向交互式地图探索的高保真矢量瓦片缩减

Tarlan Bahadori 加利福尼亚大学河滨分校 美国河滨市
tbaha001@ucr.edu

Ahmed Eldawy 加利福尼亚大学河滨分校 美国河滨市
eldawy@ucr.edu

arXiv:2603.10270v1 [cs.DB] 10 Mar 2026

摘要

交互式可视化是探索大型开放数据仓库的常用工具，用户借此快速浏览来自不同域的数据集。对于大规模空间数据，许多现有工具依赖服务器端渲染来生成可在客户端查看的小图像。然而，大多数用户更倾向于客户端渲染，因为它能够快速设置数据样式，从而带来更好的可视化体验。本文介绍 HiFIVE，这是一个支持可扩展、高保真客户端地理空间可视化的数据管理框架。我们将可视化感知瓦片缩减问题形式化，该问题刻画了瓦片大小与可视化失真之间的权衡，并证明其为 NP 困难问题。HiFIVE 提出了一种结合分流与稀疏化的两阶段解决方案，依据信息论和空间准则有选择地剪除记录、属性和值。实验表明，该方法在太字节级数据规模下大幅缩减瓦片大小，同时保持视觉保真度和交互性能。

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

1 引言

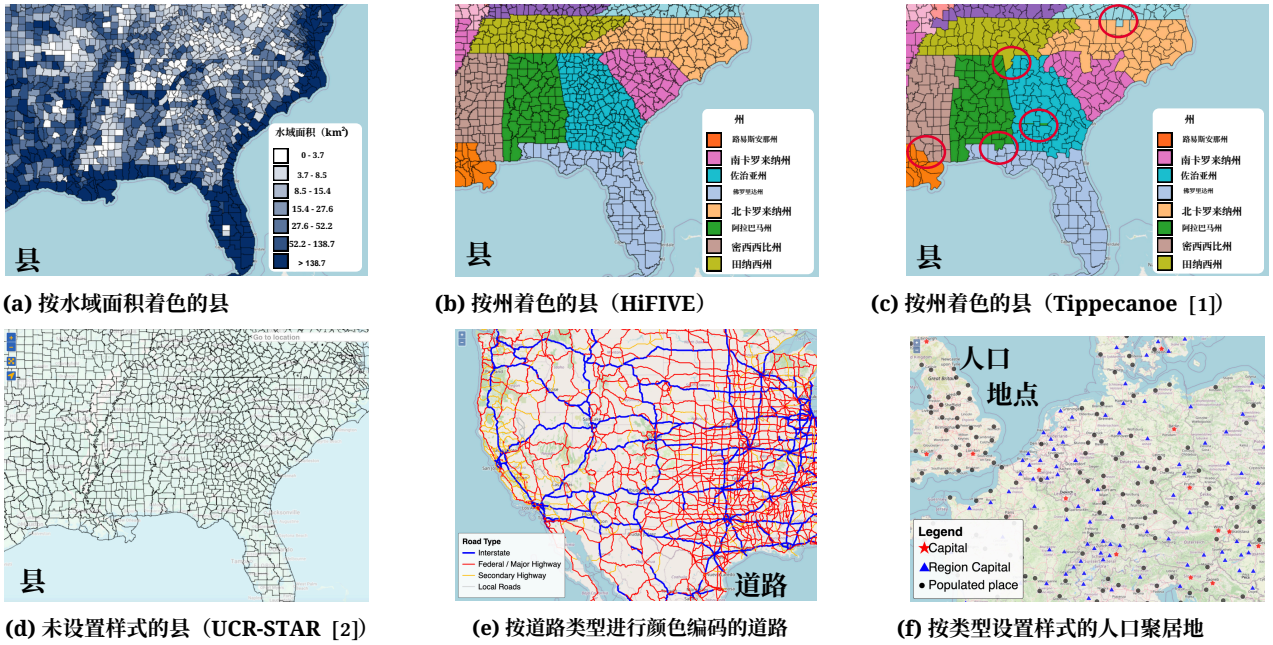
交互式可视化是确定数据集实际包含内容并评估其是否适合回答特定科学问题的最快手段之一，有时甚至是唯一手段。面对数十万个公开可用的数据集 [29, 31, 32], ，用户必须能够快速浏览、比较并排除候选数据集，以便为当前任务选择合适的数据。此外，多份报告估计，约有 60% 的公开可用数据包含地理空间要素，这凸显了地图作为探索各种域的主要界面的核心作用，包括城市规划 [7, 18], 、公共卫生 [5], 、环境监测 [21], 和交通运输 [34]。

将地理空间探索扩展到如今的数据规模仍然困难。尽管网络地图使用多分辨率瓦片金字塔实现交互式平移和缩放 [2, 12, 15], ，现有系统仍面临可扩展性与可视化保真度之间的核心权衡。现有方法大致可分为服务器端渲染 和客户端渲染。在服务器端渲染中，服务器生成并设置栅格瓦片的样式；由于计算在后端进行，且瓦片大小受像素分辨率限制，因此这种方式具有良好的可扩展性 [15, 20, 35]。然而，可视化效果实际上是固定的，客户端无法重新设置样式，也无法恢复未编码到像素中的细节。在客户端渲染中，服务器传送矢量瓦片 [3, 18, 26, 30], 即几何体和属性，由客户端进行样式设置，从而实现高保真、完全可定制的地图。然而，矢量瓦片的大小并无天然上限。其载荷会随着要素密度、几何复杂度和属性基数任意增长，常常超出实际的网络和浏览器渲染预算。已有一些工作尝试在 Mapbox 矢量瓦片 (MVT) 中使用字典式编码 [13] ，或在 Maplibre 瓦片 (MLT) 中使用列压缩 [30], ，但这些方法并未解决瓦片大小无上限的问题。

图 1 展示了三个数据集的客户端渲染。图 1(a-d) 展示了由美国人口普查局发布的美国县数据集 [4]。图 1a 展示了水文学家如何按照水域覆盖面积对县进行色彩编码。在图 1b 和图 1c 中，管理员可能会按照州对同一数据集进行色彩编码。然而，从图 1c 可以看出，一些现有工具（例如 Tippecanoe） [1],会生成低保真度可视化，其中州界发生了扭曲。原因在于，Tippecanoe 试图通过合并相邻多边形来减小数据大小，从而导致可视化不准确。另一方面，图 1d 展示了采用服务器端渲染的 AID* [15] 可视化，该方式不允许使用上述任何样式。图 1e 和图 1f 展示了另外两个示例，分别包含道路和有人居住的城市，说明用户可能会以各种方式设置数据样式，因此需要高效的客户端渲染。

本文介绍了 HiFIVE，这是一种用于地图探索的高保真可视化系统，旨在解决交互式矢量瓦片可视化中的两个根本瓶颈：可扩展性和可视化保真度。随着数据规模增大，生成的矢量瓦片可能变得非常庞大；与栅格瓦片不同，矢量瓦片没有固有的大小上限，这在处理数万个数据集时尤其构成迫切挑战。只有当生成的地图在视觉和语义上仍保持忠实时，大幅且简单地缩小瓦片才有价值。HiFIVE 将瓦片生成重新定义为一个数据管理优化问题。其核心思想是，并非所有记录和属性对于生成高保真可视化都同等重要。例如，在可视化中，空间范围较大的多边形比小多边形更加醒目。此外，与生成的通用唯一标识符（UUID）相比，某些属性（例如分类属性）通常占用空间较少，在样式设置中也更有用；而通用唯一标识符会占用空间，却几乎不能为可视化提供价值。HiFIVE 利用这些差异，决定在每个瓦片中保留哪些记录和属性，从而使瓦片大小保持在一定范围内，同时支持丰富的客户端样式设置。由于这一决策针对每个瓦片独立作出，随着用户放大地图、瓦片大小减小，瓦片中的细节也会随之增加。

本文介绍并形式化了可视化感知瓦片缩减问题。该问题建立在信息论概念之上，用于衡量缩减后瓦片中的信息损失，并结合空间数据度量来捕捉可视化损失。为此，我们提出



(a) 按水域面积着色的县　(b) 按州着色的县（HiFIVE）　(c) 按州着色的县（Tippecanoe [1]）

(d) 未设置样式的县（UCR-STAR [2]）　(e) 按道路类型进行颜色编码的道路　(f) 按类型设置样式的人口聚居地

图 1：应用于同一组矢量瓦片的不同客户端样式。

一种两阶段解决方案，即稀疏化和分流。稀疏化步骤通过将部分取值设置为空值，在最大限度减少信息损失和可视化损失的同时，缩小瓦片大小。分流步骤应用一组启发式方法，对瓦片进行快速且显著的缩减，使其准备好进入细粒度的稀疏化步骤。这些技术结合起来，可生成紧凑且符合标准的矢量瓦片，使完全在客户端进行交互式、用户自定义的样式设置成为可能，从而支持对太字节级数据集进行探索，而无需在服务器上针对每种样式重复计算。我们通过广泛的实验验证了该方法，结果表明，在保持视觉保真度并维持大规模交互性能的同时，瓦片大小得到了大幅缩减。

下面，我们总结本文的贡献：

- 我们提出瓦片失真度量，用于量化地图瓦片中的视觉信息损失。
- 我们对可视化感知瓦片缩减问题进行形式化定义，并证明该问题是 NP 困难的。
- 我们介绍 HiFIVE，并提出一种基于混合整数线性规划的两阶段解决方案。
- 我们进行了广泛的实验评估，以研究 HiFIVE 的可扩展性。

本文其余部分组织如下。第2节描述了空间数据可视化的相关工作。第3节总结了我们提出的框架，并概述其关键组成部分。第4节定义了可视化感知瓦片缩减问题，并证明其NP 困难性。第5节提出了使用混合整数线性规划的瓦片稀疏化算法。第6节描述了分流步骤。第7节报告了实验结果，证明了我们方法的有效性。最后，第8节总结全文并描述未来工作。

2 相关工作

地理空间可视化系统面临双重挑战：既要管理呈指数增长的数据集，又要保持交互性能。近年来，出现了两种主流方法来应对这些挑战：

首先，像 GeoSparkViz [35],HadoopViz [12], 和 AID* [15] 这样的**基于聚类的渲染**系统，利用 Spark 或 Hadoop MapReduce 等并行计算框架进行**服务器端渲染**。虽然这类系统能有效生成高分辨率静态地图，但由于瓦片在服务器端已部分或完全预生成，其架构限制了客户端交互性。此外，针对不同类型的可视化，还必须分别设计方案，例如核密度可视化 [5] 和热力图 [22]。这些方案能够扩展到大数据规模，但当用户请求不同的样式设置选项时，服务器很快就会过载。

第二种方法是**客户端渲染**，重点是将原始数据交付给客户端，由客户端根据用户需求设置样式并进行渲染，而不会给服务器带来任何额外开销。一些工具会先在服务器端聚合数据，再交付汇总后的版本，以便高效地进行可视化，但这仍要求服务器了解可视化类型和样式 [20]。另一种方法是对数据进行采样以供可视化 [25]，但这种方法仅适用于点数据，且不考虑自定义样式，例如颜色编码点。其他方法则会构建矢量瓦片，例如 MVT [13] 或 MLT [30],，其中包含所有记录及其全部属性。与栅格瓦片不同，矢量瓦片的大小没有上限，可能会变得任意大，从而增加传输和渲染这些瓦片的开销。Planetiler [23] 提出了一种生成地图瓦片的并行算法，但它使用的是针对 OpenStreetMap 数据的特定应用规则。同样， [26] 提出了稀疏化问题

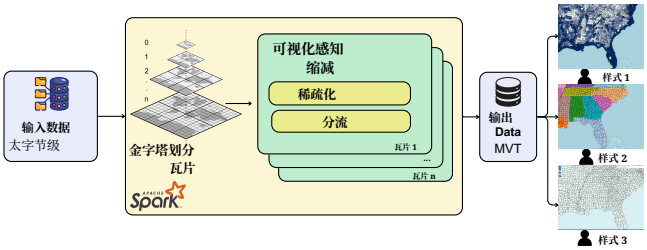


图 2：HiFIVE 流程概览。

在 Google 地图中限制每个瓦片的记录数，同时保持瓦片之间的地理连贯性，但它没有考虑各个属性的重要性。Tippecanoe [1] 提供了一种通用的 MVT 瓦片生成器，采用基于贪心策略和密度的要素剔除技术，确保地图在不同缩放级别下清晰易读。然而，其由启发式规则驱动的简化过程忽略了非空间属性的语义效用，导致对数据应用样式时视觉保真度下降。

本文采用客户端渲染方案，但做出了关键转变：将瓦片缩减视为首要优化问题，而非临时应对的次要问题。给定用户指定的大小预算，HiFIVE 会选择要在每个瓦片中具体化的记录、属性和值，以尽量减少信息损失，同时保留可视化语义。由此生成的紧凑瓦片可在客户端进行样式设置，并支持可调节的细节层次参数。

3 概述

如图 2 所示，本工作通过两阶段数据处理管线生成紧凑且可设置样式的矢量瓦片，从而实现可扩展的高保真地理空间可视化。我们的目标是支持太字节级数据集，同时确保每个瓦片的大小预算足以实现快速网络传输，并在现代网络制图应用中实现流畅的客户端渲染。

该管线始于数据准备阶段，此阶段（i）将输入几何体重新投影到网络地图所用的 Web 墨卡托投影；（ii）将数据划分为与标准网络可视化库 [17, 24, 30] 兼容的多级缩放瓦片金字塔结构。该阶段遵循成熟的实践，完整细节请参阅 [12, 15]。

与栅格瓦片 [12, 15, 35], 不同，矢量瓦片存储单个几何体及其属性。因此，瓦片大小差异很大，在密集区域可能超出浏览器和网络预算。因此，本文聚焦于可视化感知瓦片缩减：即给定一个瓦片和用户指定的大小预算 B，目标是在尽可能减少可视化保真度损失的同时缩减瓦片大小，使其序列化后的大小不超过 B 字节。问题定义的更多细节见第 4 节。

为了解决可视化感知瓦片缩减问题，我们在每个瓦片上分别运行两种算法：分流 和稀疏化。其中，分流算法会大幅削减数据，以缩小瓦片大小；稀疏化算法则是主要算法，负责将其精确缩减至所需大小。

该稀疏化算法将问题建模为线性规划（LP），并使用线性规划求解器同时减小瓦片

表1：包含 $N = 4$ 个湖泊和 $d = 3$ 个属性的玩具示例。盐度属性表示湖泊是淡水湖（f）还是咸水湖（s）。

(a) 输入瓦片 T_{in}

几何	name	盐度
G1	Azul	f
G2	Birch	s
G3	Cobalt	s
G4	Dune	f

(b) 输出瓦片 T_{out}

几何	name	盐度
G1	Azul	f
G2	Birch	s
G3	⊥	s
G4	⊥	⊥

大小，同时保持可视化保真度。其核心思想是将某些属性值设为空值，以减小数据大小并最大限度地降低对可视化的影响。具体做法是通过一组约束和目标函数，在以下因素之间取得平衡：(i) 以像素占用面积衡量的视觉显著性，(ii) 存储大小，以及 (iii) 以 KL 散度衡量的信息损失。该算法的更多细节见第 5 节。

为确保在极其密集的瓦片上仍具备可扩展性，我们还引入了一种轻量级的 可视化感知分流算法，该算法采用多种经验方法，在执行成本更高的稀疏化步骤之前缩小瓦片大小。分流步骤包括移除选定的行和列，并量化数值列。操作顺序由属性显著性决定，该显著性通过精简所导致的熵下降来衡量。这样可以确保分流针对的是最不可能在后续输出可视化样式设置中使用的属性。有关此步骤的更多细节，请参见第 6 节。

两种算法均通过分布式 Spark 作业并行运行，从而可随数据规模进行水平扩展。它们输出标准 MVT 瓦片 [13]，可使用任何现有的客户端可视化库（如 OpenLayers [24] 或 MapLibre [30]）进行渲染。由于我们的缩减算法不会改变瓦片模式或格式，现有的样式设置工具无需修改即可使用。

4 可视化感知瓦片缩减问题定义

HiFIVE 采用 Mapbox 矢量瓦片（MVT）文件格式来存储所有瓦片。这使得客户端（例如 Web 浏览器）能够根据用户输入对数据进行样式设置，而不会给服务器带来任何额外开销 [3]。为实现客户端渲染，每个瓦片都包含其所覆盖地理区域内的全部记录。这些数据存储在一个具有 N 行和 d 列的表中，其中每一行表示一个地图要素（即记录），每一列表示一个属性，例如空间几何、国家名称和人口。遗憾的是，这种结构无法扩展到非常大的数据集，因为瓦片会变得过大，不利于网络传输和客户端渲染。

我们要解决的问题是：如何在保持最终渲染地图的语义和视觉保真度的同时，缩减这些瓦片的大小。更具体地说，给定一个输入瓦片，如何生成一个输出瓦片，使其能够表示该区域的数据，同时符合期望的瓦片大小。

本节首先介绍一些初步定义，随后给出正式的问题定义。我们使用表1中的简单示例瓦片来帮助说明这些定义。

定义 *4.1*（模式（S））。模式 (S) 是属性类型 $S = (\tau_1, \ldots\ldots, \tau_d)$ 构成的有序集合，其中每个属性（列） $j \in$

{1, ..., d} 都关联着各自的域 （载体集）Dom$_j$，它是该特定列的容许值构成的有限 集合：

$$\mathrm{Dom}_j \subseteq \mathrm{Dom}(\tau_j), \qquad |\mathrm{Dom}_j| < \infty.$$

因此，即使 $\tau_j = \tau_k$ （例如，两个 Int属性），相应的域 Dom$_j$ 和 Dom$_k$ 也可能不同。域始终包含空值 $\perp$。不失一般性，我们假设第一个属性始终是几何属性，即 τ_1 =Geometry。

在表1中，模式为 (Geometry, String, String)。名称属性的域 Dom$_2$ = {Azul, Birch, Cobalt, Dune, $\perp$}，其取值仅限于此瓦片中的值。尽管名称属性和盐度属性的类型都是 String，但它们的域因取值不同而异。

定义 *4.2*（要素）。给定模式 S，一个要素（f）是一个类型化的 d-元组 $f \in \prod_{j=1}^{d} \mathrm{Dom}_j$，其中第 j 个属性值按位置引用为 $f[j] \in \mathrm{Dom}_j$。

定义 *4.3*（瓦片）。给定模式 S，瓦片 (T) 是一个有限的要素集合

$$T = \{f_i\}_{i=1}^{N}, \qquad f_i \in \prod_{j=1}^{d} \mathrm{Dom}j$$

我们用 $T \in \mathrm{Rel}(\boldsymbol{\tau})$ 表示 T是模式 S 上的一个关系。 T的基数是该瓦片中的要素数量，即 $N = |T|$。

在表1a中，输入瓦片有四个要素，即 $|T_{\text{in}}| = 4$。输出瓦片 T_{out} $|T_{\text{out}}|$ = 具有相同的模式和基数，即 $|T_{\text{in}}| = 4$，但其属性值不同。

定义 *4.4*（列大小）。给定瓦片 T，第 j 列 是其第 j 个属性值的序列：

$$C_j(T) = \langle t_1[j], \ldots, t_N[j] \rangle \in \mathrm{Dom}(\tau_j)^N$$

设 Enc$_j$ 是一个依赖于类型的列编码函数，用于联合编码列 j 中的所有取值：

$$\mathrm{Enc}_j : \mathrm{Dom}(\tau_j)^N \to \{0,1\}^*$$

列 j的大小 定义为其编码表示中的字节数：

$$\mathrm{Size}_j(T) = \left|\mathrm{Enc}_j(C_j(T))\right|_{\text{bytes}}$$

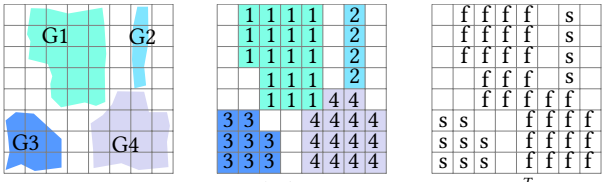
这描述了列式格式，例如 MLT [30], , 以及字典式格式，例如 MVT [13], , 其中一列中的值会一起编码，而不是按行分别编码。

在此示例中，将属性值置空会减少不同非空条目的数量，并提高空值的出现频率（$\perp$），从而缩小受影响列的编码表示。

定义 *4.5*（瓦片大小）。瓦片大小 T 是其所有列的大小之和：

$$\mathrm{Size}(T) = \sum_{j=1}^{d} \mathrm{Size}_j(T)$$

像素加权语义差异。以下部分定义了两个瓦片 T_1 和 T_2 如何根据其值和感知到的渲染效果产生差异。我们的目标是尽量减少两个瓦片之间的失真，其中考虑了：**(1) 视觉显著性：**



(a) T_{in} (b) 栅格化 (c) $I_3^{T_{\text{in}}}$

图 3：瓦片 T_{in}的栅格化示例。(a) 输入的几何体。(b) 几何体的栅格化结果。(c) 盐度属性（$j = 3$）的栅格化结果。空单元格表示 $\perp$。

占据更多屏幕空间、即具有更大像素覆盖范围的要素，其属性值在视觉上更突出；以及 **(2) 语义影响：** 我们应尽量减少可能用于样式设置的属性中的信息损失，这种损失由值分布表示。我们发现，熵较低的属性（例如分类属性）更可能用于样式设置，因为它们具有可观察到的模式。

定义 *4.6*（渲染函数（R））。给定分辨率为 R像素的瓦片以及一个几何体 g，渲染（或栅格化）函数会返回该几何体 g 覆盖的像素集合。

$$\mathcal{R} : g \to \{1, \ldots, R\} \tag{1}$$

图 3 展示了将四个几何体栅格化到 $R = 64$ 像素上的示例。我们用行号（即 1 到 4）表示栅格化函数的结果。

定义 *4.7*（属性图像）。给定一个瓦片 T、一个渲染函数 R, 以及一个非几何属性 $j \in \{2, \ldots, d\}$，我们定义一个属性图像 I_j^T：

$$I_j^T : \{1, \ldots, R\} \to \mathrm{Dom}_j$$

将每个像素赋值为覆盖该像素的要素的属性值：

$$I_j^T(\mathbf{x}) = \begin{cases} f_i[j] & \text{if } \mathbf{x} \in \mathcal{R}(f_i[1]) \text{ for some feature } i, \\ \perp & \text{if no geometry covers } \mathbf{x}. \end{cases} \tag{2}$$

因此， I_j^T可被视为通过用其属性值绘制每个几何而得到的合成图像。

图3(c)展示了 T_{in} 的盐度属性图像，其中每个像素标记为“f”或“s”，或留空以表示空值。

定义*4.8*（像素加权经验分布）。值 $v \in \mathrm{Dom}_j$ 的像素加权计数是 I_j^T中值等于 x的像素数量：

$$c_j^T(v) = \left|\left\{\mathbf{x} \in \{1, \ldots, R\} \;:\; I_j^T(\mathbf{x}) = v\right\}\right| \tag{3}$$

表2展示了示例中所用两块瓦片的像素计数。请注意，尽管取值‘f’和‘s’在表中各出现两次，它们对应的像素加权计数分别为 $c_3^{T_{\text{in}}}($ s = 12 = 32)和 $c_3^{T_{\text{in}}}(f)$。

表2：玩具示例（$R = 64, \varepsilon = 1$）的像素加权计数 $c_j^T(v)$ 及对应的拉普拉斯平滑分布 $\tilde{p}_j^T(v)$。未覆盖像素分配给 $\perp$。粗体值表示 T_{in} 和 T_{out}之间的变化。

属性 j	值 v	$c_j^{T_{\text{in}}}(v)$	$c_j^{T_{\text{out}}}(v)$	$\tilde{p}_j^{T_{\text{in}}}(v)$	$\tilde{p}_j^{T_{\text{out}}}(v)$
name	Azul	18	18	0.275362	0.275362
	Birch	14	14	0.217391	0.217391
	Cobalt	8	**0**	0.130435	**0.014493**
	Dune	4	**0**	0.072464	**0.014493**
	$\perp$	20	**32**	0.304348	**0.478261**
盐度	淡水	32	32	0.492537	0.492537
	salt	12	**8**	0.194030	**0.134328**
	$\perp$	20	**24**	0.313433	**0.373134**

对于名称，$|\text{Dom}_j| = 5$，归一化常数为 $R + \varepsilon|\text{Dom}_j| = 64 + 5 = 69$。对于盐度，$|\text{Dom}_j| = 3$，归一化常数为 $64 + 3 = 67$。

基于这些计数，我们计算取值的经验分布，同时应用拉普拉斯平滑，将参数 $\varepsilon > 0$加到每个计数 c_j^T上，以处理零计数：

$$\tilde{p}_j^T(v) \;=\; \frac{c_j^T(v) + \varepsilon}{R + \varepsilon|\text{Dom}_j|}, \qquad v \in \text{Dom}_j \tag{4}$$

表2展示了调整后的经验权重。请注意，由于平滑处理，计数为零的项会变成较小的调整后权重。

定义 *4.9*（可视化感知属性差异）。给定两个具有相同模式和行数的瓦片 T_1 和 T_2，我们将属性 $j \in \{2, \ldots, d\}$ 的可视化感知差异定义为它们像素加权经验分布之间的詹森-香农散度。首先，我们定义混合分布：

$$m_j = \tfrac{1}{2}\left(\tilde{p}_j^{T_1} + \tilde{p}_j^{T_2}\right) \tag{5}$$

据此，可将可视化感知差异定义为：

$$\text{VAD}_j(T_1, T_2) = \tfrac{1}{2}\text{KLD}\left(\tilde{p}_j^{T_1} \,\|\, m_j\right) + \tfrac{1}{2}\text{KLD}\left(\tilde{p}_j^{T_2} \,\|\, m_j\right) \tag{6}$$

其中：

$$\text{KLD}(p\|q) = \sum_{x \in \text{Dom}_j} p(x) \log_2 \frac{p(x)}{q(x)} \tag{7}$$

请注意，使用 $\log_2$ 可确保对于所有列 j，$\text{VAD} \in [0, 1]$。

表 3 展示了两个样例瓦片之间名称列与盐度列的散度。

解释。由于这些分布由绘制图像 I_j^T诱导产生，对像素覆盖范围较大的几何体的属性值所做的任何更改都会移动更多概率质量，从而更显著地增大可视化感知属性差异$_j$。特别是，将视觉上较大的几何体的某个值置空，会把质量从其原始取值箱转移到 $\perp$ 箱，从而产生相应更大的散度。

接下来，我们通过聚合所有列的散度来计算两个瓦片之间的散度。然而，我们观察到，用户倾向于使用熵较低的属性进行样式设置，例如取值较少的分类属性。在此示例中，‘盐度’属性的熵低于‘名称’属性（每条记录的取值都不同），因此不太可能用于对

表 3：计算得到的熵 $H_j(T_{\text{in}})$、逐属性散度可视化感知属性差异$_j(T_{\text{in}}, T_{\text{out}})$**（JSD，$\log_2$）、逆熵权重 $w_j(T_{\text{in}})$（其中 $\delta = 10^{-9}$、$\gamma = 1$），以及最终** TLD。

属性 j	$H_j(T_{\text{in}})$（比特）	可视化感知属性差异$_j$（T_{in}, T_{out}）	$w_j(T_{\text{in}})$
name	2.170965	0.067751	0.406485
盐度	1.486845	0.005812	0.593515

$$\text{TLD}(T_{\text{in}},\ T_{\text{out}}) = \sum_{j \in \{\text{name},\ \text{盐度}\}} w_j(T_{\text{in}}) \cdot \text{可视化感知属性差异}_j(T_{\text{in}},\ T_{\text{out}}) = 0.030989$$

可视化进行样式设置。因此，我们使用如下所示的逆熵加权聚合来计算瓦片散度。

定义 *4.10*（瓦片失真）。给定两个具有相同模式的瓦片 T_1 和 T_2，我们将瓦片失真定义为按熵倒数加权聚合的逐属性散度。我们将属性 j 在 T_1 中的熵定义为：

$$H_j(T_1) \;=\; -\sum_{x \in \text{Dom}_j} \tilde{p}_j^{T_1}(x)\, \log_2 \tilde{p}_j^{T_1}(x). \tag{8}$$

为突出熵较低的属性，我们为每个属性分配一个与其熵成反比的权重：

$$w_j(T_1) \;=\; \frac{(H_j(T_1) + \delta)^{-\gamma}}{\sum_{k=2}^{d} (H_k(T_1) + \delta)^{-\gamma}},$$

其中，$\delta > 0$ 用于防止除以零，$\gamma \geq 0$ 用于控制对低熵属性的强调程度（当 $\gamma = 0$ 时，权重均匀）。因此，瓦片级失真为：

$$\text{TLD}(T_1, T_2) \;=\; \sum_{j=2}^{d} w_j(T_1) \cdot \text{VAD}_j(T_1, T_2). \tag{9}$$

表 3 展示了每列的熵、调整后的权重以及最终的瓦片失真，从而完整呈现了这一示例。

可视化感知瓦片缩减问题定义。给定一个输入瓦片 T_{in}，其模式为 S，字节预算为 B，我们的目标是构造一个输出瓦片 T_{out}（具有相同的模式），使其大小不超过预算，同时相对于 T_{in}最小化视觉/语义失真。形式化地说，我们求解：

$$\begin{aligned} \min_{T_{\text{out}} \in \text{Rel}(S)} \quad & \text{TLD}(T_{\text{in}}, T_{\text{out}}) \\ \text{s.t.} \quad & \text{Size}(T_{\text{out}}) \leq B, \\ & |T_{\text{out}}| = |T_{\text{in}}| \end{aligned} \tag{10}$$

定理4.11。 公式*10*中定义的可视化感知瓦片缩减问题是*NP*困难的。

证明概要。我们通过展示从 0-1背包问题出发的多项式时间归约来证明该定理。考虑一个 0-1背包问题实例，其中包含 n 个物品，大小为 s_i，值为 v_i，容量为 B。我们构造一个瓦片缩减问题实例，其中输入瓦片 T_{in} 包含 n 个要素和两个属性。第一个属性是几何，第二个属性为每个要素赋予一个唯一值，例如，$_i\ [2] = i$ 中的 f。此外，我们调整渲染函数，使每个要素恰好栅格化为 v_i 个像素，彼此不重叠且没有空像素。如果每个几何都是长度为 v_i 的线，且像素总数为 $\sum v_i$，则可以轻松实现这一点。然后，我们将其中的取值限制为

输出瓦片 T_{out} 中的值只能是与 T_{in} 对应的相同值，或空值（$\perp$），即 $f\text{out}_i\ [2] \in \{f\text{in}_i\ [2], \perp\}$。最后，我们定义瓦片编码器函数，使每个属性的大小等于背包问题中对应的大小（如果该值非空），即 $\text{Size}(T_{\text{out}}) = \sum s_i \colon\ f\text{out}_i\ [2] \neq \perp$。

此设置可确保非空取值的大小不超过背包容量 B，这一点显而易见。同时，它会将被置为空值的取值总值最小化，从而最大化非空取值的值，如下所示。

首先，任何非空（即保留的）取值 $i \text{ in } T_{\text{out}}$ 对 KLD 的贡献都为零，因为 $p_i^{T_{\text{in}}}(x) = p_i^{T_{\text{out}}}(x) = m(x)$，如方程 5 和 7 所示。另一方面，如果取值 i 被置为空值，其所有质量都会分配给空值（$\perp$），即 $p_i^{T_{\text{out}}} = 0$ 和 $m_i = \frac{1}{2} p_i^{T_{\text{in}}}$。因此，它在方程 7 中的项为：

$$p_i^{T_{\text{in}}} \log_2 \left(\frac{p_i^{T_{\text{in}}}}{\frac{1}{2} p_i^{T_{\text{in}}}} \right) = p_i^{T_{\text{in}}} \log_2(2) = p_i^{T_{\text{in}}}$$

由于每个要素 i 的像素数量等于其背包值 v_i，且这些值各不相同，因此有 $p_i^{T_{\text{in}}} = v_i$。此外，由于输入瓦片没有空像素（即 $p_\perp^{T_{\text{in}}} = 0$），空值不会对 KLD 产生贡献。因此，瓦片缩减会使被置为空值的取值总大小最小化，这等价于使保留（非空）取值的总大小最大化。因此，它解决了背包问题，证明完毕。□

请注意，上述证明要求使用非平滑分布，即 $\varepsilon = 0$ 在方程 4中使用的分布，但这不会改变问题的性质。

5 基于混合整数线性规划的稀疏化

给定一个包含 N 条记录（行）和 d 个属性（列）的矢量瓦片，我们的目标是在保留视觉显著性和属性效用的同时，将序列化瓦片大小降至用户指定的阈值 B 以下。此问题适用于金字塔中所有大小超过所需阈值的瓦片。鉴于问题的复杂性，我们将其简化并重新表述为一个混合整数线性规划（MILP），以共同决定（i）保留哪些记录，以及（ii）为保留的记录保留哪些属性值（单元格）。

决策变量。 不失一般性，我们假设第一列包含几何属性。我们引入以下二元变量：

$$y_i \in \{0,1\} \quad \text{if geometry of record } i \in [1,N] \text{ is kept} \tag{11}$$
$$u_j \in \{0,1\} \quad \text{if non-geometric attribute } j \in [2,d] \text{ is kept} \tag{12}$$
$$x_{i,j} \in \{0,1\} \quad \text{if attribute cell } (i \in [1,N], j \in [2,d]) \text{ is kept} \tag{13}$$

变量 $x_{i,\ j}$ 仅针对非空单元格实例化；若列非空，则实例化 u_j。

结构约束。 一个单元格仅当其几何及其列都被保留时才能被保留：

$$x_{i,j} \le y_i \qquad \forall i \in [1,N], j \in [2,d], \tag{14}$$
$$x_{i,j} \le u_j \qquad \forall i \in [1,N], j \in [2,d]. \tag{15}$$

表4：输入 T_{in}（表1a）的混合整数线性规划变量示例。

	名称 (u_2)	盐度 (u_3)
y_1	$x_{1,2}$	$x_{1,3}$
y_2	$x_{2,2}$	$x_{2,3}$
y_3	$x_{3,2}$	$x_{3,3}$
y_4	$x_{4,2}$	$x_{4,3}$

表 5：混合整数线性规划求解示例；结果见 T_{out}（表 1b）。

	1	1
1	1	1
1	1	1
1	0	1
1	0	0

线性大小模型。 为了针对我们的优化约束近似瓦片大小，我们将总存储分解为三个主要组成部分：几何数据、固定列开销，以及每单元格的摊销成本。令 $\widehat{b}_i^{geom}$ 表示记录 i的几何所贡献的估计字节数，该估计由其顶点数得出。在矢量瓦片编码中，属性值并非独立存储；相反，每个图层维护一个由不同取值组成的共享字典。因此，属性列 j 的存储成本由字典成本 $\widehat{b}_j^{dict}$（表示唯一非空值所需的字节数）和用于引用该字典的每单元格指针成本 b^{ptr} 组成。

为了保持模型的线性，我们将字典成本分摊到该列的n_j 个非空单元格上。因此，保留一个单元格（$i,\ \ j$）时，其估算成本为 $b^{ptr} + \widehat{b}_j^{dict}/n_j$。虽然只有在完全移除某个唯一值时，删除单个单元格才会减小字典大小，但这种近似仍然非常有效。基数较高的列自然会产生较高的摊销成本，因此在固定大小预算下是优先考虑削减的对象。相反，具有少量不同取值的列对总大小的贡献很小，因此这种近似对整体约束的影响可以忽略不计。未来的工作可以进一步完善这种线性近似，以更好地捕捉 NP 困难的可视化感知瓦片缩减问题中字典大小固有的离散跃变。由此得到的线性尺寸约束为：

$$\sum_{i=1}^{N} \widehat{b}_i^{geom}\, y_i + \sum_{(i,j)} \left(b^{ptr} + \frac{\widehat{b}_j^{dict}}{n_j} \right) x_{i,j} \ \le\ B, \tag{16}$$

其中 B 是目标瓦片大小。

目标函数。 该混合整数线性规划最大化加权效用，在记录的视觉显著性与保留属性的保真度之间取得平衡：

$$\max \ \ \alpha \sum_{i=1}^{N} U_i^{rec}\, y_i + (1-\alpha) \sum_{(i,j)} U_{i,j}^{cell}\, x_{i,j}. \tag{17}$$

其中 U_i^{rec}表示记录 i 基于其视觉显著性的效用，$U_{i}^{\text{cell}}{}_{,j}$ 表示用于衡量信息保留情况的单元格效用，而 $\alpha \in [0,1]$ 是用于平衡这两部分重要性的参数。下文将进一步介绍这两个效用函数。

记录效用。 视觉显著性通过记录在渲染后的瓦片中的准确像素占用面积 pc_i of record i 进行量化。我们将其除以瓦片中的最大像素占用面积进行归一化，

$$\widetilde{pc}_i = \frac{pc_i}{\max_j pc_j}, \tag{18}$$

并将记录效用定义为

$$U_i^{rec} = \lambda_{rec}\, \widetilde{pc}_i^{\,p}, \tag{19}$$

其中，$\lambda_{rec} > 0$ 控制显著性偏好的强度，$p \geq 1$ 控制其非线性程度。指数 p 会放大归一化像素占用面积的差异，增强视觉上较大与较小记录之间的对比。这会促使线性规划求解器保留较大的几何体；否则，在满足大小约束时，由于这些几何体的字节开销较高，它们会显得格外适合被删除。

单元格效用。 对于每个属性单元格 (i, j)，我们使用归一化散度分数 $\widetilde{D}_{i,j} \in [0,1]$ 衡量保留其值的重要性，例如，计算属性 j 在 T_{in} 中的值相对于参考分布的归一化 KL 散度。

$$D_{i,j} \;=\; \mathrm{KLD}\Big(\tilde{p}_j^{T} \,\Big\|\, \tilde{p}_j^{T^{(i,j\leftarrow\perp)}}\Big). \tag{20}$$

$$\widetilde{D}_{i,j} \;=\; \frac{D_{i,j}}{D_j^{\max}}, \qquad D_j^{\max} \;=\; \max_{r\in\{1,\ldots,N\}} D_{r,j}. \tag{21}$$

这里，$\tilde{p}_j^{T^{(i,j\leftarrow\perp)}}$ 表示表 T 中列 j 的经验分布：将单元格 (i, j) 中的值替换为 $\perp$（空值），同时保持所有其他单元格不变。

我们定义一个保留奖励

$$\mathrm{KB}_{i,j} = 1 - \widetilde{D}_{i,j}, \tag{22}$$

并赋予单元格效用

$$U_{i,j}^{cell} = \mathrm{KB}_{i,j}. \tag{23}$$

这种表述清晰地划分了职责：记录效用优先考虑视觉上显著的几何体，而单元格效用则在剩余大小预算内保留最具信息量的属性值。因此，混合整数线性规划会优先移除较小或视觉上不显著的记录，并将剩余预算用于选择能够最大限度减少分布失真的属性值。

问题表述完成后，我们将其交给线性规划求解器，为决策变量赋值。为了创建输出瓦片 Tout，我们复制输入瓦片 Tin，并在对应变量 $x_{ij} = 0$ 时，将 $f_i[j]$ 处的每个单元格设为空值。此外，如果 $y_i = 0$，我们会移除整行；如果 $u_j = 0$，则移除整列。

算法 1概述了用于稀疏化矢量瓦片的混合整数线性规划表述。

算法 1 CellSparsifyMILP(T_{in}, B)

需要： 输入瓦片 $T_{\text{in}} = \{f_i\}_{i=1}^{N}$，包含 d 个属性（第 1 列为几何）
需要： 大小阈值 B
确保： 缩减后的瓦片 T_{out}

计算记录显著性
1: **对于** $i \leftarrow 1$ 到 N **执行**
2: 计算记录 i的像素占用面积 pc_i
3: **对于** $i \leftarrow 1$ 到 N **执行**
4: 根据 pc_i计算记录效用 U_i^{rec}

计算用于大小估算的列统计量
5: **对于** $j \leftarrow 2$ 到 d **执行**
6: 计算字典字节数 $\widehat{b}_j^{dict}$和非空计数n_j
7: **对于** $i \leftarrow 1$ 至 N **执行**
8: 估算几何字节数 $\widehat{b}_i^{geom}$

计算单元效用
9: **对于**每个非空单元格 $(i,\ j)$ **执行**
10: 计算归一化散度 $\widetilde{D}_{i,j}$
11: $U_{i,j}^{cell} \leftarrow 1 - \widetilde{D}_{i,j}$

混合整数线性规划表述
12: 创建混合整数线性规划模型 M
13: 添加决策变量 y_i、u_j 和 $x_{i,\ j}$
14: 添加目标函数（式 17）
15: 添加结构约束，将 $x_{i,\ j}$ 与 y_i 和 u_j 相连
16: 添加线性尺寸约束（式 16）
17: 求解 M
18: 根据 $(y,\ u,\ x)$ 构造 T_{out}
19: **返回** T_{out}

第 0–4 行计算每条记录的视觉显著性。第 2 行测量每个几何体在渲染函数 R（式 1）下的像素占用面积 pc_i。随后，根据式 19 对这些值进行归一化，并转换为记录效用 U_i^{rec}，以优先保留占据较大可见面积的几何体。

第 4–8 行计算线性大小模型所需的统计数据。对于每个属性列 j，第 6 行估算字典大小 $\widehat{b}_j^{dict}$和非空单元格数量 n_j。字典大小对应于在共享矢量瓦片字典中对列 j 的不同非空取值进行编码所需的字节数。第 8 行根据每条记录的顶点数估算其几何对象开销 $\widehat{b}_i^{geom}$。第 8–11 行为各个属性单元格分配效用。对于每个非空单元格 $(i,\ j)$，~第 10 行使用公式 20–21 中定义的按列 KL 散度计算归一化散度 $D_{i,\ j}$。第 11 行将此分数转换为保留奖励 $\mathrm{KB}_{i,\ j} = 1 - \widetilde{D}_{i,\ j}$。因此，那些单元格的移除后造成最小分布变化的记录会获得更高效用，因此更有可能被保留。第 11–17 行对混合整数线性规划进行表述并求解。目标函数（公式 17）使用参数 α平衡记录效用与单元效用，而大小约束（公式 16）确保序列化后的瓦片不超过目标大小 B。最后，第 18 行通过移除具有$y_i = 0$的记录、删除具有 $u_j = 0$的列，并在 x_i 时将相应单元格设为空值，来构造输出瓦片$_j = 0$。

6 可视化感知分流

稀疏化问题及其混合整数线性规划表述在缩小瓦片大小的同时，能够很好地保留可视化保真度。然而，由于表述复杂，它可能无法扩展到超大型瓦片。这一点对于高缩放级别的瓦片尤为重要，因为这些瓦片覆盖的地理区域非常广阔，例如整个国家或大陆。在极端情况下，零级瓦片包含整个数据集。

为解决这一问题，本节介绍可视化感知分流问题：该方法采用一组启发式方法，以极低的开销显著减小瓦片大小，使其为稀疏化步骤做好准备。这些启发式方法分为两种主要方法：记录级分流通过仅保留高优先级记录进行水平切分；基于列的分流则删除部分列，并缩减其他部分列的大小。

6.1 记录级分流

限制瓦片大小的一种自然策略是减少记录数。一种直接的解决方案是选取随机样本

采用固定大小，例如蓄水池采样。可视化感知采样器 [25] 侧重于未设置样式的点散点图，但不适用于非点几何体或基于其他属性的样式设置。

本部分通过添加优先级扩展了蓄水池采样的理念，该优先级旨在优先保留预计对可视化贡献更大的记录，从而提高可视化保真度。这涉及我们需要解决的两个问题：首先，如何定义蓄水池容量；其次，在选择记录时应使用哪种优先级。

为解决第一个问题，我们必须决定如何设置瓦片的容量，该容量决定了记录被纳入蓄水池时所受的约束。请注意，蓄水池容量不同于稀疏化算法使用的缩减阈值 B，且通常要大得多。我们考虑了四种容量指标，每种指标分别反映瓦片复杂度的不同方面：

记录数： 这是采样算法中一种简单直接且广泛采用的选项，例如传统的蓄水池采样或可视化感知采样（VAS） [25]。

总字节大小： 直接控制预期输出大小，并且与客户端的加载时间最为密切相关。

顶点数： 能够反映几何复杂度，并可直接影响可视化质量和渲染时间。

单元格数： 反映非空间属性对整体瓦片大小的贡献。

第二个需要解决的问题是如何对添加到瓦片中的记录进行优先级排序。采样优先级决定在竞争蓄水池容量时哪些记录会被优先纳入，其中包括：

记录字节大小： 用于在大量小型记录与较少但包含更多信息的大型记录之间进行选择。

顶点数： 探讨在保留较少复杂几何体与保留较多简单几何体之间的权衡。**元组大小（属性）：** 检验保留大量稀疏记录还是较少信息丰富的记录能否获得更高质量。

像素大小： 按投影后的屏幕投影范围对要素进行优先级排序。

随机： 作为无偏基线。

请注意，蓄水池容量与优先级的决策是相互正交的。例如，若容量为总字节大小 且优先级为最大像素大小，我们将按像素大小降序对所有记录进行逻辑排序，并逐一选取，直到总字节大小超过容量。因此，我们可以用这种方法探索不同选项，并研究其可视化保真度。

由于我们的目标是为稀疏化步骤准备瓦片，我们决定采用单元格数 ，它直接影响线性规划求解器的处理成本。我们让稀疏化步骤执行如前所述的更全面的约简。

为了更好地理解候选优先级指标之间的关系，我们计算了三个代表性数据集（省份、道路和国家）的斯皮尔曼相关矩阵。如表6所示，我们发现顶点数、像素数量和字节大小之间存在很强的相关性。这是合理的，因为顶点数较多的几何通常面积更大，也会占用更多字节。这意味着，按其中任一指标排序，结果都大致相同，因此我们决定采用最大的

表6：优先级指标之间的斯皮尔曼相关性。

指标	顶点	像素	属性
省份			
像素	0.743	–	–
属性	0.195	0.089	–
字节大小	0.979	0.731	0.226
道路			
像素	0.901	–	–
属性	0.061	0.070	–
字节大小	0.879	0.811	0.460

作为默认优先级指标，因为顶点数 最容易计算。

总而言之，记录级分流提供了一种计算开销低且高度并行的方式，可在进行任何细粒度分析之前缩减大型瓦片。然而，由于这种方法会丢弃整条记录，因此可能移除在视觉或语义上重要的内容，即使只有这些记录中的特定属性或单元格导致了大小超限问题。因此，纯粹的记录级分流虽然效率高，却未必总能保留密集区域的保真度。

6.2 列级筛选

第二组启发式方法作用于列级，旨在减少列的数量以及其中存储的信息量。如果瓦片在记录级分流后仍超出大小约束，则会在稀疏化之前执行此步骤。

6.2.1 属性子集选择。 首先，这一步通过移除可视化中未使用的列来减小数据大小。所有行中的值都相同（即零熵）的列都可以安全移除，因为它们不会带来任何视觉信息。此外，用户可以手动移除与当前应用无关的属性，也可以借助大语言模型 [3]，移除。例如，某些数据集包含多个对用户没有意义的内部标识符，或包含多种语言的标签，其中一些可以移除。这一步是可选的，用户也可以选择继续使用整个数据集。

6.2.2 几何简化。 由于各级别的瓦片大小不同，我们可以借此机会简化某些几何，以降低其细节层次和大小，同时保持其在目标缩放级别下的视觉保真度。对于这一步，系统提供三种方法：Douglas-Peucker算法、拓扑保持简化或网格捕捉简化。

Douglas-Peucker算法接收一个几何和容差距离 ε。随后，它会迭代地连接两个端点，并移除所有距离小于 ε 的中间点。接着，几何会被拆分为两部分，并重复这一过程。通过将 ε 设置为接近相应缩放级别的像素大小，简化几何体在视觉上就会与原始几何相似。

拓扑保持简化器的工作方式与 Douglas-Peucker 类似，但增加了额外约束，以确保简化几何体不会导致任何自相交或拓扑变化问题。

网格吸附简化会应用固定大小的均匀网格，并将每个顶点吸附到最近的网格单元。随后，它会删除多个连续且吸附到同一网格单元的点。

除简化之外，系统还会进行几何裁剪，确保仅保留与瓦片边界相交的几何体部分。移除超出瓦片范围的几何体，并将跨越边界的几何体裁剪至瓦片范围内。系统会针对不同几何类型（如点、多点、线串和多边形）处理特殊情况，并对每种类型采用适当的简化策略。

最终渲染时，所有坐标都会根据 Mapbox 矢量瓦片（MVT）规范转换为整数坐标。

6.2.3 数值量化。对于数值列，我们会进行量化，将取值分组成少数几个离散值，例如默认分为 10 个值。鉴于这些取值通常用于为记录创建配色方案，可视化中的颜色差异是可以接受的，尤其是在使用色标时。量化后，基于列和基于字典的存储方式（如 MVT）占用的磁盘空间更少。此外，量化会降低列的熵，使稀疏化步骤能够赋予这些属性更高的优先级。为缩短计算时间，仅对唯一值超过 20 个的数值列进行量化。

6.2.4 字符串截断。 为了减小文本属性的大小，我们将其截断为固定长度。字符串属性通常用作标签，截断后可显著减小瓦片大小，同时仍能让用户了解标签的内容。例如，若原始标签为“Massachusets”，将其缩短为“Mass…”，仍然具有参考价值。为了衡量信息损失，我们在对应映射取值的同时，计算原始取值与截断后取值之间的 KL 散度。如果截断导致标签合并，例如“Santa Cruz”和“Santa Clara”都变成“Santa C…”，则会产生更高的散度分数。

6.2.5 优先列约简。 对于数值量化和字符串截断，我们通过计算每个属性约简前后概率分布之间的 KL 散度，对属性约简进行优先级排序。这种信息论度量量化了转换所导致的预期信息损失。KL 散度值越低，表示约简越能保留属性原有的信息内容。约简过程以迭代方式进行，首先处理 KL 散度变化最小的属性。每次约简属性后，系统都会重新计算瓦片大小，并继续这一过程，直到大小低于目标阈值或所有属性均已处理。

7 实验

在本节中，我们对 HiFIVE 进行全面评估，以提供其可扩展性的实验证据。我们还使用若干客观指标评估所生成的风格化矢量瓦片，以评估可视化质量。

表7：实验中使用的主要数据集。

数据集	Size	#记录	几何
eBird [28]	935.3 GB	801M	点
OSM_道路 [11]	31 GB	70M	线串
OSM_建筑物 [11]	187 GB	455M	多边形
邮政_编码 [11]	7 GB	236k	多边形
道路_NA [9]	81 MB	49k	线串
县 [4]	193 MB	3k	多边形

表8：实验设置。

设置	取值（默认值）
瓦片大小约束（B）	32 KB – 4 MB（256 KB）
分诊优先级	顶点数最多
瓦片容量	（100K 个单元格）
缩放级别	1 – 15 (8)
α	0.0 – 1.0（0.5）
输入样本量	0.001% – 100%（100%）

7.1 实验设置

我们在一个十二节点的 Spark集群上运行了实验。主节点配备两颗 8 核 Intel Xeon E5-2609v4 CPU（1.70 GHz）和 128 GB 内存，而每个工作节点配备两颗 6 核 Intel Xeon E5-2603 v4 CPU（1.70 GHz）和 64 GB 内存。所有机器均运行 CentOS 7.5.1804，使用本地固态硬盘作为操作系统存储，并使用机械硬盘进行数据存储。

表7列出了实验中使用的数据集。这些数据集包括点、线和多边形，规模最高达935GB。对于eBird、OSM道路和OSM建筑等大规模数据集，我们还使用均匀随机采样以不同比例（例如10%、1%、0.1%及更小）额外生成多个子集。这些子集支持可扩展性实验，并使我们能够在广泛的数据量范围内研究瓦片大小约束对性能和视觉保真度的影响，同时保留原始数据集的空间与属性特征。

基线：我们考虑两种基线。AID* [15] 是一种基于Spark的可视化索引，它将预生成瓦片与数据索引相结合，以支持栅格瓦片的可扩展可视化。Tippecanoe [1] 是Mapbox [13] 用于创建缩减大小矢量瓦片的工具。

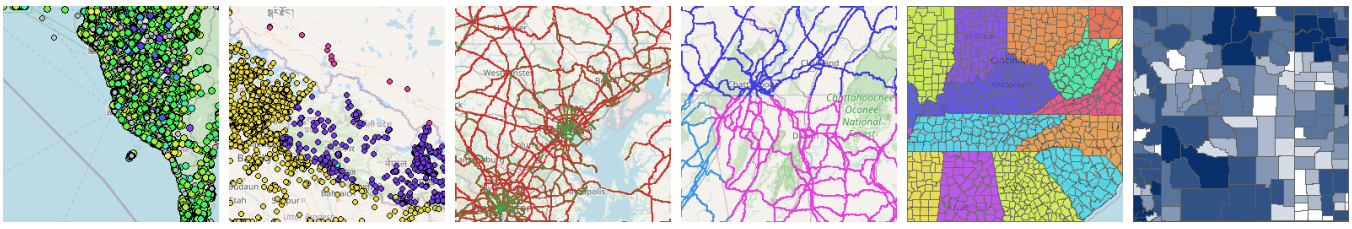
参数：表8总结了我们调整的实验参数及其对应的默认值。

7.2 质量指标

为评估视觉保真度，我们使用相同的客户端样式设置，将每个瓦片（基准版本和精简版本）渲染为大小为 $H \times W$（在我们的实验中为 $H = W = 256$）的 RGB图像。图 4 总结了我们在评估中使用的样式。设 $A, B \in \{0, \ldots, 255\}^{H \times W \times 3}$ 表示生成的基准图像和精简图像，并设 $A_{u,v,c}$ 表示像素 (u, v) 处通道 $c \in \{1, 2, 3\}$ 的强度。

均方根误差。 我们计算所有像素和通道上的逐像素均方误差（MSE），

$$\mathrm{MSE}(A, B) = \frac{1}{3HW} \sum_{u=1}^{H} \sum_{v=1}^{W} \sum_{c=1}^{3} \left(A_{u,v,c} - B_{u,v,c}\right)^2, \qquad (24)$$



(a) eBird（GRAD） (b) eBird（CAT） (c) 道路（GRAD） (d) 道路（CAT） (e) 县（CAT） (f) 县（渐变）

图 4：实验中使用的可视化样式示例。我们展示了两种 eBird 样式（基于一天中观测时间的渐变样式和基于国家代码的分类样式）、两种道路样式（基于长度的渐变样式和基于州的分类样式），以及两种县样式（基于水域面积的渐变样式和基于州的分类样式）。

并报告均方根误差（RMSE），

$$\mathrm{RMSE}(A,B) = \sqrt{\mathrm{MSE}(A,B)}. \tag{25}$$

峰值信噪比。峰值信噪比（PSNR）使用 8 位峰值 $I_{\max} = 255$，以分贝（dB）为单位报告。PSNR 越高，表示视觉一致性越好。

$$\mathrm{PSNR}(A,B) = \begin{cases} +\infty & \text{if } \mathrm{MSE}(A,B) = 0, \\ 20\log_{10}(I_{\max}) - 10\log_{10}(\mathrm{MSE}(A,B)) & \text{otherwise.} \end{cases} \tag{26}$$

结构相似性指数（SSIM）。SSIM 指标 [33]是一种感知指标，旨在通过比较亮度、对比度和结构的局部模式，而非全局逐像素误差，与人类视觉判断保持一致。该指标通过在参考图像 A 和对比图像 B上滑动固定大小的窗口，在每个位置提取局部图像块 x 和 y。这些图像块之间的相似性定义如下：

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{27}$$

其中，μ和 σ分别表示局部均值和方差，σ_{xy}表示协方差，而 C_1、C_2 则是稳定常数。

最终得分是对所有窗口位置和颜色通道上的 SSIM 求平均值。在我们的评估中，我们使用 7×7 个窗口，强度范围为 $[0, 255]$。SSIM 的取值范围为 $[-1, 1]$，数值越高表示结构保真度越好。未来的工作可以探索调整这一优化目标，直接最小化问题表述中定义的瓦片级失真，从而有可能免去约简过程中完整的图像渲染。

7.3 端到端评估

本节评估 HiFIVE 相较于 Tippecanoe 和 AID* 的端到端瓦片生成运行时间、存储占用和可视化保真度。图 5 展示了三个最大数据集（eBird、OSM_道路和 OSM_建筑物）的运行时间和存储占用，输入大小从 0.001% 扩展至 100%。对于每种设置，我们都会生成一个包含八个缩放级别的金字塔（总计约 20,000 个瓦片），并报告输出总大小。由于 HiFIVE 和 AID* 使用 Spark，两者都能扩展至最大的那个数据集。值得注意的是，HiFIVE 在不到一小时内即可处理近 1 TB 的数据集。生成的瓦片可用于实时生成任意数量的样式，且不会增加服务器负担。相比之下，由于 Tippecanoe 针对

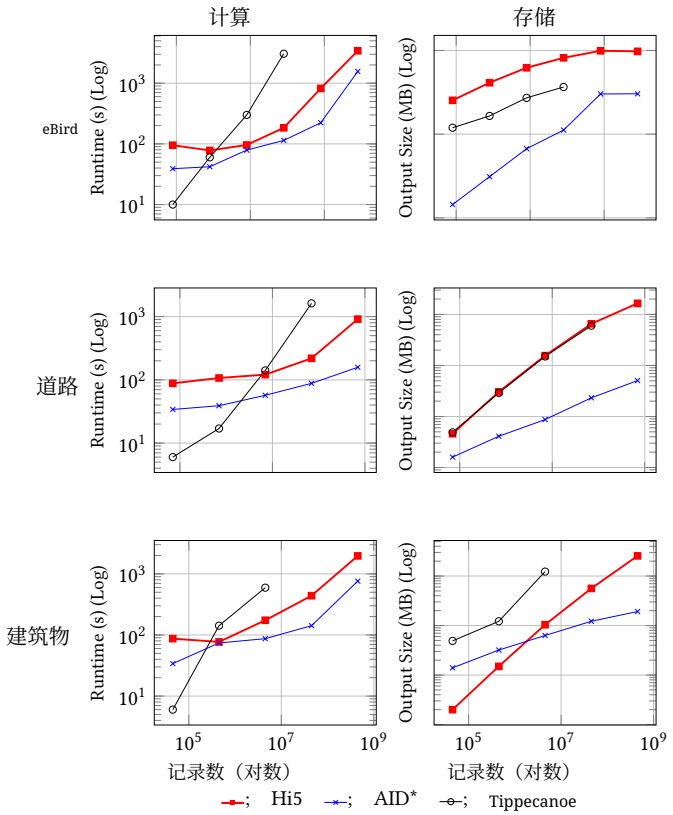


图 5：eBird 子集（8k 至 801M 条记录）、道路子集（7k 至 70M 条记录）和建筑物子集（45k 至 455k 条记录）的扩展性对比。两个坐标轴均采用对数刻度。

在单机环境下，它仅在小型数据集上表现良好，在较大型数据集上则会耗尽内存。随着数据集规模扩大，所有方法的输出大小都会增加，因为更多瓦片变得有内容；栅格瓦片（AID*）仍然小得多，因为它们只编码像素，而不编码几何体和属性。最后，即使名义上的瓦片预算相同，HiFIVE 和 Tippecanoe 得到的实际大小也有所不同，这是因为它们采用了根本不同的简化策略和大小模型。

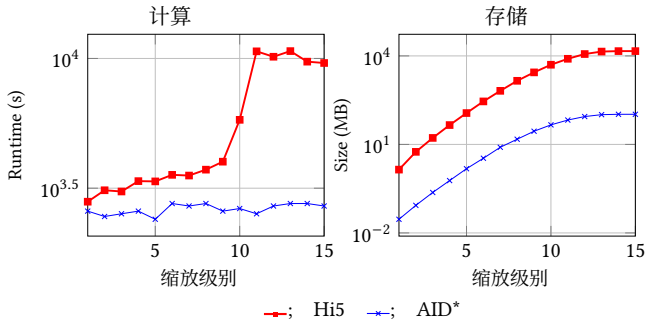


图6：eBird 数据集在缩放级别 1–15 下的性能对比。图中以对数刻度展示基于线性规划的方法（Hi5）和 AID* 的运行时间与输出大小。

表9：跨数据集的 TLD 与可视化误差指标之间的斯皮尔曼等级相关。TLD 越高表示偏差越大，因此它与 RMSE 呈正相关，与 PSNR/SSIM 呈负相关。

指标	县	NE_道路	eBird 0.01
斯皮尔曼(TLD，均方根误差)	0.999823	0.813129	0.882601
斯皮尔曼(TLD，峰值信噪比)	-0.951515	-0.813129	-0.882601
斯皮尔曼(TLD，结构相似性)	-0.999867	-0.797708	-0.856090

图6展示了对于 eBird 数据集，金字塔深度从 1 变化到 15 对运行时间和存储大小的影响（Tippecanoe 因内存耗尽而未纳入）。正如预期，AID* 未设置样式的栅格瓦片大小最小、运行速度最快，但代价是不支持任何客户端样式设置。HiFIVE 的运行时间和大小起初因瓦片数量庞大而增加，但随着每个瓦片的大小受到上限约束，最终趋于稳定。

图7评估了道路样式化渲染的可视化质量。由于 Tippecanoe 存在可扩展性问题，我们在 1%样本上进行这项实验。HiFIVE 和 Tippecanoe 的目标瓦片预算均设置为 $B = 256$kb。HiFIVE 中的分流步骤配置为容量 100k 个单元格，并优先处理顶点较多的几何体。应用于道路数据集的着色样式包括基于“highway”属性的分类 样式，以及基于“max speed”属性的渐变样式。我们在其他数据集上也观察到一致的趋势，但由于篇幅所限，未展示相关结果。在这两种样式下，HiFIVE 的保真度最高，即结构相似性和峰值信噪比最高，均方根误差最低。结构相似性的差异最为显著，因为它与感知到的视觉结构相关性更高，这表明 HiFIVE 的可视化感知降采样更善于保留可视化特征。我们注意到，峰值信噪比和均方根误差表现出相同的趋势，也可以在一定程度上用于评估可视化质量。最后，AID* 在这些指标上的表现不具竞争力，因为它生成的栅格瓦片不提供任何用户样式设置选项。

为评估公式9中的瓦片级失真函数（TLD）的效用，表9展示了上述三种质量指标与所提出的TLD指标之间的斯皮尔曼相关系数。我们注意到，TLD与

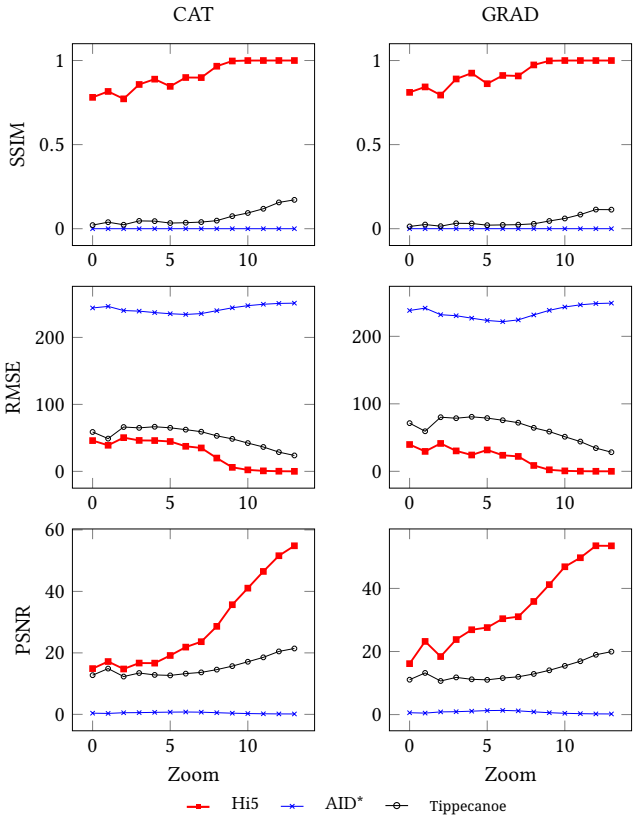


图7：CAT 与 GRAD 渲染样式（道路）的各缩放级别平均指标。

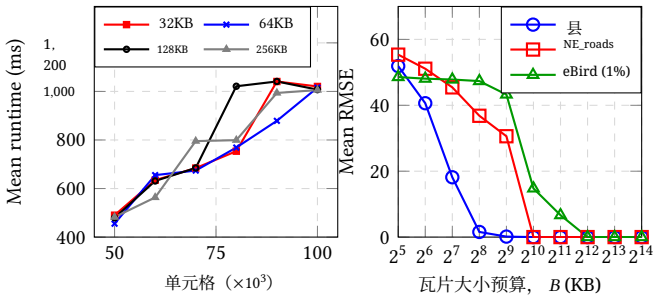


图 8：左：对于 MILP求解器，在不同输出大小约束下，运行时间与输入单元格数量的关系。右：使用不同大小约束进行缩减后，初始字节大小最高的前 20 个瓦片的平均均方根误差。

将这三种指标应用于风格化图像时，结果均如此。这是一个非常重要的发现，因为 TLD 是一种风格无关指标，但可以用来替代对风格化图像的评估。这意味着，未来研究这一问题的研究人员可以专注于解决所提出的可视化感知的瓦片缩减问题，而不必针对特定风格调整方法。

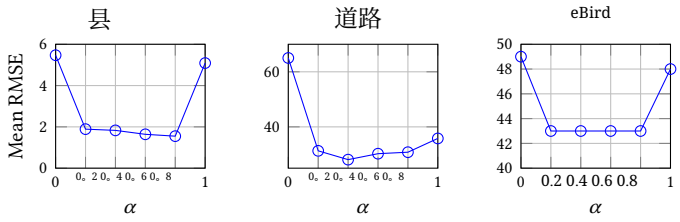


图 9：三个数据集的平均均方根误差随 α 的变化。

7.4 稀疏化实验

所有混合整数线性规划实例均使用 Gurobi 优化器（11.0.2）[19] 求解，相对最优性间隙为 1%。本节研究稀疏化步骤的性能，以及它如何影响生成瓦片的质量。为衡量其性能，我们运行八层的完整金字塔生成过程，并记录每次调用稀疏化步骤（算法 1）的情况。图 8 左侧报告了单元格数（$N \times d$）从 50K 变化到 100K 时的平均耗时。此范围涵盖了大于瓦片预算（B）的瓦片（以触发稀疏化算法），以及小于分流步骤蓄水池容量（100k 个单元格）的瓦片。如图所示，运行时间随单元格数增加而近似线性增长，因为单元格数决定了问题中的变量数量。此外，可以观察到，瓦片预算 B 对总运行时间几乎没有影响，因为它不会影响问题规模。

图 8 右侧报告了包含点、线和多边形的三个数据集中最大的 20 个瓦片的平均误差。我们将瓦片大小预算（B）从 32 kb 调整到 4 mb，并衡量其对可视化误差（均方根误差）的影响。正如预期，误差随预算 B 的增大而迅速下降。该实验为系统设计者如何选择瓦片预算提供了指导。影响这一选择的另一个维度是客户端能力和加载时间，但这超出了本研究的范围。

最后，图 9 展示了公式 17 中参数 α 对输出瓦片质量的影响。回顾可知，α 在由像素占用面积衡量的视觉显著性与由 KL 散度量化的语义信息之间取得平衡。结果表明，两种极端情况（$\alpha = 0$ 和 $\alpha = 1$）表现最差，因为它们只关注其中一个方面。结果还表明，α 可设置的取值范围很宽，这使得系统设计者设置该值相对简单。

如图 8 所示，输入稀疏化步骤的单元格数是 MILP求解器运行时间的主要驱动因素，因此我们在分流步骤为瓦片设定固定的单元格容量，并力求以不同的可能优先级填满这 10 万个单元格的预算，如第 6 节所述。

7.5 定性评估

图 10 展示了采用四种方法渲染的两个代表性瓦片。第一行描绘了按 AWATER 属性进行渐变着色的县数据集。在相同的瓦片大小约束下，Tippecanoe 和 HiFIVE 都与原始数据高度匹配，不过红色圆圈标出了 Tippecanoe 丢弃或合并记录的区域。第二行展示了采用分类

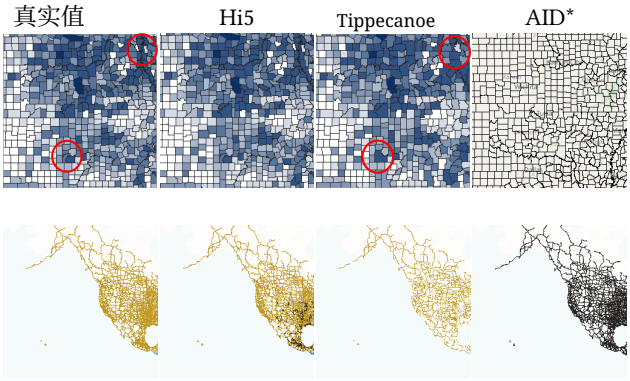


图 10：两个瓦片的定性比较。每一行显示同一风格化瓦片，依次为真实值、Hi5、Tippecanoe 和 AID*

着色的道路数据集，其依据是“continent”属性。在这里，HiFIVE 移除了部分较小记录的该属性取值，使其呈现为黑色；而 Tippecanoe 则直接移除了更多记录（即任何保留的记录都会保留其全部属性）。因此，Tippecanoe 的输出更加稀疏。

8 结论

本文提出了 HiFIVE，这是一种感知可视化框架，旨在缩减矢量瓦片大小，同时在交互式地图探索中保留视觉显著性和语义保真度。HiFIVE 将瓦片缩减建模为一个优化问题，明确刻画存储成本与可视化质量之间的权衡，并综合考量几何显著性和属性信息损失。基于混合整数线性规划的稀疏化方法采用以几何复杂度和基于字典的编码为依据的线性大小模型，有选择地移除记录和属性值；同时，根据信息论散度得出的单元效用可确保保留的取值对属性分布的影响尽可能小。结合轻量级分流阶段以提升可扩展性，HiFIVE 可生成紧凑且符合标准的矢量瓦片，支持客户端样式设置。鉴于所提出的无样式瓦片级失真（TLD）指标与感知图像质量高度相关，未来工作应探索能够最小化这种失真的替代优化算法。

参考文献

[1] Tippecanoe GitHub 仓库。https://github.com/mapbox/tippecanoe/blob/master/README.md。访问日期：2025-02-15。[2] [无日期。UCR STAR。https://star.cs.ucr.edu。访问日期：2025-02-15。[3] Tarlan Bahadori、Sai Sreekar Sarvepalli 和 Ahmed Eldawy。2025年。LASEK：面向地理空间数据的 LLM 辅助式样探索工具包。《*VLDB* 会议论文集》18，12（ 2025 5435 5438 10 14778 3750601 3750690 年8月），–。https://doi.org/。/。[4] 2019 10 6086 12 7 美国人口普查局。无日期。县。https://doi.org/。/N F KGT，取自 UCR-STAR: https://star.cs.ucr.edu/?TIGER2018/COUNTY&d。[5] Tsz Nam Chan、Pak Lon Ip、Kaiyan Zhao、Leong Hou U、Byron Choi 和 Jianliang Xu。2022年。LIBKDV：用于地理空间分析的多功能核密度可视化库。《*VLDB* 会议论文集》15，12（2022年），3606–3609。https://doi.org/ 10 14778 3554821 3554855 。/。Liming Dong、Qiushi Bai、Taewoo Kim、Taiji Chen、Weidong Liu 和 Chen Li。2020年。Marviq：通过物化实现范围选择查询的质量感知地理空间可视化。载于*2020年ACM SIGMOD*国际数据管理会议论文集。67–82。[7] Harish Doraiswamy、Eleni Tzirita Zacharatou、Fabio Miranda、Marcos Lage、Anastasia Ailamaki、Cláudio T. Silva 和 Juliana Freire。2018年。交互式可视化 [6]

HiFIVE：面向交互式地图探索的高保真矢量瓦片缩减

使用 Urbane 交互式探索时空城市数据集。载于*2018*年国际数据管理会议论文集，*2018*年*SIGMOD*会议，美国得克萨斯州休斯敦，*2018*年*6*月*10*日至*15*日，Gautam Das、Christopher M. Jermaine 和 Philip A. Bernstein（编）。ACM，1693–1696。https://doi.org/10.1145/3183713. 3193559[8] Carsten F. Dormann、Jana M. McPherson、Miguel B. Araújo、Roger Bivand、Janine Bolliger、Gudrun Carl、Richard G. Davies、Alexandre Hirzel、Walter Jetz、W. Daniel Kissling 等。2007年。在物种分布数据分析中考虑空间自相关的方法：综述。《*Ecography*》30，5（2007年），609–628。https://doi.org/10.1111/j.2007.0906-7590.05171.x [9] Natural Earth。2019年。北美洲主要道路。https://doi.org/10.6086/N1JM27N3，取自 UCR-STAR：https://star.cs.ucr.edu/?NE/roads_NA&d。[10] Ahmed Eldawy、Vagelis Hristidis、Saheli Ghosh、Majid Saeedan、Akil Sevim、A.B. Siddique、Samriddhi Singla、Ganesh Sivaram、Tin Vu 和 Yaming Zhang。2021年。Beast：时空数据的可扩展探索性分析。载于第*30*届*ACM*国际信息与知识管理会议论文集（*CIKM '21*）。美国计算机协会，纽约，美国，3796 3807 10 1145 3459637 3481897 –。https://doi.org/。/。[11] Ahmed Eldawy 和 Mohamed F. Mokbel。无日期。SpatialHadoop：用于空间数据的 MapReduce 框架。载于第*31*届*IEEE*国际数据工程会议（*ICDE 2015*），韩国首尔，*2015*年*4*月*13*日至*17*日，Johannes Gehrke、Wolfgang Lehner、Kyuseok Shim、Sang Kyun Cha 和 Guy M. Lohman（编）。IEEE计算机学会，1352 1363 10 1109 2015 7113382 –。https://doi.org/。/ICDE。。Ahmed Eldawy、Mohamed F. Mokbel 和 Christopher Jonathan。无日期。HadoopViz：用于大规模空间数据可扩展可视化的 MapReduce 框架。（2016 601 612 10 1109 2016 7498284），–。[13] https://doi.org/。/ICDE。。GDAL文档团队。无日期。*MVT*：*Mapbox* 矢量瓦片 — *GDAL*文档。GDAL项目。https://gdal.org/drivers/vector/mvt.html。[14] Kareem El Gebaly、Guoyao Feng、Lukasz Golab、Flip Korn 和 Divesh Srivastava。2018年。解释表。*IEEE*数据工程技术委员会通报41，4（2018年），43–55。[15] Saheli Ghosh 和 Ahmed Eldawy。2022年。AID*：用于地理空间数据可视化探索的空间索引。《*IEEE*知识与数据工程汇刊》34, 8 2022 3569 3582 10 1109 2020 3026657（），–。[16] https://doi.org/。/TKDE。。Saheli Ghosh、Tin Vu、Mehrad Amin Eskandari 和 Ahmed Eldawy。2019年。UCR-STAR：UCR时空活跃数据仓库。《*SIGSPATIAL Special*》11，2（2019 34 40 10 1145 3377000 3377005 年12月），–。[17] https://doi.org/。/。谷歌。无日期。Google 地图。https://maps.google.com。[18] Tao Guo、Kaiyu Feng、Gao Cong 和 Zhifeng Bao。2018年。高效选择地图上的地理空间数据以进行交互式可视化探索。载于*2018*年国际数据管理会议论文集，*2018*年*SIGMOD*会议，美国得克萨斯州休斯敦，*2018*年*6*月*10*日至*15*日，Gautam Das、Christopher M. Jermaine 和 Philip A. Bernstein （编）。ACM，567–582。https://doi.org/10 1145 3183713 3183738 。/。Gurobi Optimization, LLC。无日期。Gurobi优化器参考手册。https://www.gurobi.com [19] [20] Jianfeng Jia、Chen Li、Xi Zhang、Chen Li、Michael J. Carey 和 Simon Su。2016年。面向十亿条推文的交互式分析与可视化。载于第*24*届*ACM SIGSPATIAL*地理信息系统进展国际会议论文集，*GIS 2016*，美国加利福尼亚州伯灵格姆，*2016*年*10*月*31*日至*11*月*3*日，Siva Ravada、Mohammed Eunus Ali、Shawn D. Newsam、Matthias Renz 和 Goce Trajcevski（编）。ACM，85:1–85:4。https://doi.org/10.1145/2996913.2996923 [21] Yunfan Kang、Ziang Zhao、Amr Magdy、Win Cowger 和 Andrew B. Gray。2019年。面向人为垃圾数据的可扩展多分辨率空间可视化。载于第*27*届 *ACM SIGSPATIAL* 地理信息系统进展国际会议论文集，*SIGSPATIAL 2019*，美国伊利诺伊州芝加哥，*2019*年*11*月*5*日至*8*日，Farnoush Banaei Kashani、Goce Trajcevski、Ralf Hartmut Güting、Lars Kulik 和 Shawn D. Newsam（编）。ACM，560–563。https://doi.org/10.1145/ 3347146.3359074[22] Lauro Didier Lins、James T. Klosowski 和 Carlos Eduardo Scheidegger。2013年。用于实时探索时空数据集的 Nanocubes。*IEEE* 可视化与计算机图形学汇刊19, 12 2013 2456 2465 10 1109 2013 179（），–。[23] https://doi.org/ . /TVCG. 。OnthegoMap。。Planetiler：用于构建行星尺度矢量瓦片的快速工具。https://github.com/onthegomap/planetiler。 [24] OpenLayers 贡献者。2025年。OpenLayers——网页制图库。https: //openlayers.org/ [25] Yongjoo Park、Michael Cafarella 和 Barzan Mozafari。2016年。面向超大型数据库的可视化感知采样。（2016），755–766。https://doi.org/10.1109/ ICDE.2016.7498289[26] Anish Das Sarma、Hongrae Lee、Hector Gonzalez、Jayant Madhavan 和 Alon Y. Halevy。2013年。用于地图可视化的大型地理数据一致性稀疏化。*ACM* 数据库系统汇刊38，4（2013），22。https://doi.org/10.1145/2539032. 2539034[27] Bettina Speckmann 和 Kevin Verbeek。2010年。项链地图。*IEEE* 可视化与计算机图形学汇刊16，6（2010），881–889。https://doi.org/10. 1109/TVCG.2010.197[28] Brian L Sullivan、Christopher L Wood、Marshall J Iliff、Rick E Bonney、Daniel Fink 和 Steve Kelling。2009年。eBird：生物科学领域的公民参与式鸟类观测网络。生物保护142，10（2009），2282–2292。[29] 世界银行。2026年。世界银行开放数据。https://data.worldbank.org/。访问日期：2026年2月28日。[30] Markus Tremmel 和 Roland Zink。2025年。MapLibre Tile：新一代矢量瓦片格式。载于第*33*届 *ACM* 地理信息系统进展国际会议论文集（*SIGSPATIAL '25*）。ACM，美国明尼苏达州明尼阿波利斯。https://doi.org/10 1145 3748636 3763208 . / 。联合国统计司。。UNdata：联合国数据服务。https://data.un.org/。访问日期：2026年2月28日。[32] 美国总务管理局。2026年。Data.gov：美国政府开放数据门户。https://data.gov/。访问日期：2026年2月28日。[33] Zhou Wang、Alan C. Bovik、Hamid R. Sheikh 和 Eero P. Simoncelli。2004年。图像质量评估：从误差可见性到结构相似性。*IEEE* 图像处理汇刊13, 4 2004 600 612 10 1109 2003 819861（），–。[34] https://doi.org/ . /TIP. 。Shi Yin、Moyin Li、Nebiyou Tilahun、Angus G. Forbes 和 Andrew E. John-son。2015 年。通过交互式可视化了解芝加哥大都会区的交通可达性。载于第*1*届 *ACM SIGSPATIAL* 智慧城市与城市分析国际研讨会论文集，*UrbanGIS@SIGSPATIAL 2015*，美国华盛顿州贝尔维尤，*2015*年*11*月*3*日至*6*日。ACM，77–84。https://doi.org/10 1145 2835022 2835036 . / 。Jia Yu、Zongsi Zhang 和 Mohamed Sarwat。2018年。GeoSparkViz：Apache Spark 生态系统中的可扩展地理空间数据可视化框架。（2018），12:1–12:12。https://doi.org/10.1145/3221269.3223040[36] Zhiguang Zhou、Fengling Zheng、Jin Wen、Yuanyuan Chen、Xinyu Li、Yuhua Liu、Yigang Wang 和 Wei Chen。2023年。一种通过卷积神经网络实现大规模地理点数据可视化的用户驱动采样模型。*IEEE* 人机系统汇刊53，5（2023），885–896。https://doi.org/10.1109/THMS.2023.3296692