时序超图的拓扑分析

E Audun Myers[1], Cliff Joslyn[1], Bill Kay[2],Emilie Purvine[1],Gregory Roek[1],和 Madelyn Shapiro[1]

[1] 太平洋西北国家实验室，数据科学数学[2] 太平洋西北国家实验室，
计算数学

arXiv:2302.02857v1 [cs.CG] 6 Feb 2023

摘要。 本文研究时序超图的拓扑性质。超图是图的一种更高维推广，能够捕捉多元连接。因此，超图已成为网络科学不可或缺的一部分。超图的一种常见用途是将事件建模为超边，其中事件可以涉及许多作为节点的元素。这提供了对事件更完整的描述，不受图的标准二元连接的限制。然而，事件通常还具有时间信息，即事件发生的时间区间。因此，时序超图应运而生，能够准确捕捉事件的时间信息及其多元连接。研究这类时序超图的常用工具，通常通过滑动窗口过程对采样得到的快照进行汇总统计，以捕捉底层动态的变化。然而，这些工具既不能刻画超图结构随时间的演变，也无法揭示对底层系统具有重要影响的持久分量。为满足这一需求，我们借助拓扑数据分析（TDA）领域的之字形持久性，研究随时间演变的超图的拓扑结构变化。我们将处理流程应用于网络安全和社交网络数据集，展示其时序超图的拓扑结构如何变化，以及如何利用这些变化理解底层动态。

1 引言

复杂网络是研究动力系统的一种自然工具，其中系统的元素以二元方式建模，并随时间演化。现实世界中有许多例子，例如社交网络 [28], 疾病传播动力学 [18], 制造商—供应商网络 [31],电网 [26], 以及交通网络 [9]。驱动这些网络的底层复杂动力系统会导致其结构随时间变化，随着动力系统的变化，连接和元素也会不断增删。我们可以将这类复杂网络概括为动力网络 [17] ，其生成的图是时序图，并且复杂网络的每个连接和/或元素都带有时间属性。

虽然时序网络有助于理解元素之间存在二元关系的系统，但复杂网络并不总是适合用于建模多个实体之间的关系 [11]。对于无法用二元连接描述的多元关系数据，超图能够捕捉更丰富的社区结构信息。例如，在第3节.1中，我们探讨了一个基于有关 COVID-19 的讨论串的 Reddit 数据（PAPERCRANE [5]）构建的超图。二元模型中，只有当两位用户在同一讨论串中发帖时，二者之间才会有一条超边；这种模型会丢失有关讨论串规模的所有信息。相比之下，在超图中，每个讨论串都是一条超边，且当且仅当用户在某个讨论串中发过帖时，该用户才属于该超边，因此能够保留数据的整体结构。因此，当关注高阶结构时，超图分析是一种强大的工具。超图的一些应用实例包括人类基因集 [19,12] ，其中基因以复杂组合方式相互作用；网络数据 [19] ，其中域名系统将多个域名与 IP地址相映射；以及涉及大型群体之间互动的社交网络 [11]。

在许多应用场景中，相较于分析复杂系统如何变化，系统的单个快照并没有那么重要。通常，还可以进一步将这些网络建模为时序超图（THG），方式与时序图相同，为多元连接和元素关联时间属性（例如时间区间或时间点）。这类应用可见于许多领域，例如汽车数据（CAN 总线）中的异常检测 [16] ，以及使用我们在第3节中考察的 Operationally Transparent Cybersecurity 数据进行网络安全研究。2 [15]。

许多用于研究 THG 特征的常用工具，都基于底层超图快照的汇总统计量。这些统计量有助于了解底层超图结构的动态变化。

信息发布编号：PNNL-SA-181478

例如，在 [8], Cencetti 等人的研究中，他们将时间社会网络视为超图，并利用突发性统计量测量多方社会互动的突发性。尽管此类统计量有助于进行变化检测并了解系统动力学，但在解释时序超图结构变化方面仍存在不足。研究时序超图的另一种方法是采用可视分析。在 [13], 中，有研究设计了一种用于时序超图分析的基于矩阵的可视分析工具，可帮助了解超图的动态变化。然而，可视化工具在自动解读方面天然存在局限，通常需要专业知识才能正确理解。

超图与图的区别在于，超边不仅可以具有任意大小，还可以连接成任意复杂的模式。因此，超图实际上可以具有复杂的数学拓扑[3] ，它们是多维对象的复杂“粘合”，而这些对象可以拥有复杂的形状和结构。研究超图拓扑正日益成为一个重要领域，其中常常利用超图作为抽象单纯复形（ASC）的表示形式。

拓扑数据分析（TDA） [10,33] 旨在衡量数据的形状。具体而言，数据被表示为一个 ASC，随后通过测量其同调来检测整体拓扑形状，包括连通分量、孔洞、空腔和高阶结构。然而，这通常需要选择参数以从数据生成 ASC，而这往往并非易事。另一种更自动化的数据形状测量方法是使用 TDA 中的持久同调。该方法通过从数据中提取一系列 ASC 来研究数据的形状，这一序列称为滤过。持久同调已成功应用于众多领域，包括制造业 [20,32], 生物学 [4], 动力系统 [22,29], 和医学 [27]。许多应用要么将数据表示为点云，要么表示为图。对于点云数据，通常通过识别欧几里得距离在不断增大的半径参数范围内的点，生成由一系列 Vietoris-Rips 复形 [10] 构成的滤过。对于图数据，类似的方法是使用基于最短路径距离的距离滤过 [22,3]。

超图也通过拓扑数据分析（TDA）工具进行了研究。具体而言， [14] 中的研究展示了如何使用各种抽象单纯复形（ASC）表示来研究超图的同调，例如关联抽象单纯复形 [25] 或相对／限制重心细分。

然而，应用持久同调的一个要求是，序列中相邻的抽象单纯复形（ASC）之间必须存在单调包含映射（即序列中的每个后续 ASC 都以前一个 ASC 为子集）。不过，与数据集相关的许多 ASC 序列并不具有单调性，但我们仍希望追踪其结构变化。时间数据中常见这种情况，例如，超图中的边会随时间出现又消失，从而破坏持久同调所要求的单调性。

为解决这一问题，可以应用之字形持久性 [7] 。之字形持久性并非通过距离滤过（例如 Vietoris-Rips过滤）来测量静态点云数据的形状，而是利用一个由 ASC 交替组成的序列（称为“之字形过滤”），测量拓扑生成元在 ASC 序列中持续存在的时间。
持久同调（PH）和之字形持久性都会通过持久图或条形码追踪同调的形成与消失。持久图或条形码是由持久配对（b，d）构成的二维摘要，其中 b 表示某一维度的“洞”的生成元的诞生或形成时间，而 d 表示其死亡或消失时间。例如，在 [30] 中，霍普夫分岔是通过对滑动窗口上的 Vietoris-Rips 复形进行之字形持久性分析，并利用一维同调检测到的。另一项近期应用 [23]研究了时序网络，其中图快照使用最短路径距离和 Vietoris-Rips 复形表示为 ASC。然而，这两种方法都需要选择一个距离参数，以便在每一步构造 ASC，而这通常并非易事。

之字形持久性得到的持久条形码也可以使用持久性图像 [1] 或持久景观 [6]等方法进行向量化。这样一来，就可以利用机器学习方法对得到的持久图进行自动分析，以完成分类或回归。

在这项工作中，我们利用之字形持久性研究时序超图。通过采用超图的 ASC 表示来测量时序超图结构的变化，我们能够检测超图中连通分量的形成、合并与分离，以及环或孔洞等更高维特征和空腔。检测这些更高维特征对于时序超图分析至关重要，因为它们可能会根据应用领域产生影响。此外，与创建抽象的

[3] 请注意，此处的“拓扑”采用其形式上的含义，以区别于图应用中非正式地用它来指代网络连通模式的用法。

从点云或图数据构造 ASC 不同，超图有其自然的 ASC 表示，因此无需选择距离参数 [14]。

本文第2节介绍时序超图以及超图的 ASC 表示，并概述之字形持久性，以及如何将这些内容融入我们运用之字形持久性研究时序超图的方法中。本节还通过一个简单示例展示处理流程中的每个步骤。第3节展示如何将我们的方法应用于从社交网络和网络数据中获取的两个数据集。最后，第4节给出结论并探讨未来工作。

2 方法与背景

本节在介绍必要背景知识的同时，阐述如何运用之字形持久性研究时序超图。我们的方法结合了之字形持久性与超图的 ASC 表示，用于对时序超图进行拓扑分析。具体而言，我们设计了一套处理流程，通过滑动窗口过程运用之字形持久性，研究时序超图形状的变化。该处理流程如图1所示。

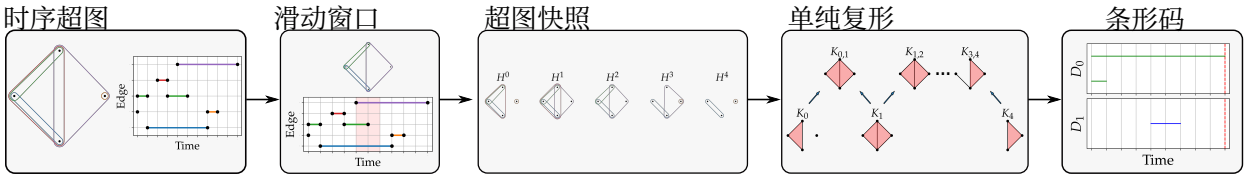


图1：运用之字形持久性分析时序超图的处理流程。

我们从图1–时序超图中的时间边属性超图开始，其中每条超边都有与之关联的活跃区间，如第2.1节所述。接下来，我们采用图1–滑动窗口 方法，选择窗口大小w 和步长s ，并以离散步长沿着这些区间所在的时间域滑动。利用每个滑动窗口，我们生成每个窗口对应的图1–超图快照 ，如第2.2节所述。随后，我们使用第2.3节中的关联抽象单纯复形，将每个快照表示为图1–*ASC* 。接下来，我们在第2.4节介绍单纯同调，以研究 ASC 的形状。由此引出一种用于关联一系列 ASC 中同调的方法，即第 2.5 节所述的之字形持久同调；该方法用于计算时序超图的持久同调，并将结果表示为持久图的条形码（图1–条形码）。

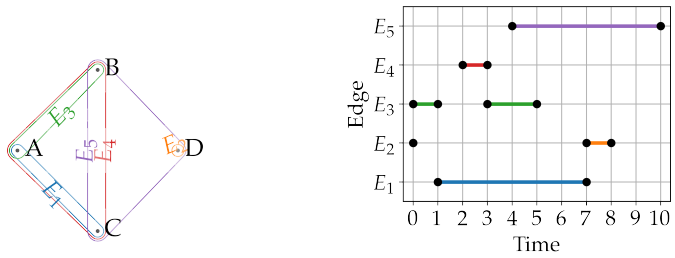
为说明我们的流程，我们在处理流程的每个步骤中都提供了一个简单示例。对于该示例及其余结果，我们使用 Python 软件包 HyperNetX[4] 生成超图，并使用 Dionysus2[5] 计算之字形持久性。

2.1 时序超图

图 G(V, E) 由一组顶点组成，这些顶点通过一组边相连，其中 $E \subseteq \binom{V}{2}$。超图 H(V, E) 由一组顶点 V 和一族边 E 组成，其中对每个 $E_i \in E$，$E_i \subseteq V$。这样，超图可以表示 k 个顶点之间的连接，并将其视为一条 k-超边。例如，考虑图 2a 中的玩具超图，它包含四个节点 $V = \{A、B、C、D\}$ 和五条超边 $E = \{E_1, E_2, E_3, E_4, E_5\}$。示例中的这些超边大小从作为 1-超边的边 $E_2 = (D)$，到作为 3-超边的边 $E_4 = (A, B, C)$ 不等。

时序超图 $H(V, E, T)$ 是其底层静态超图的一个副本，并增加了与顶点、边或关联关系相关联的时间属性 T。当与节点相关联的时间信息是相对于超边而言时，该属性便属于关联关系。在本研究中，我们仅使用归属于超边的时间信息。

[4] HyperNetX：https://pnnl.github.io/HyperNetX[5]
Dionysus2：https://mrzv.org/software/dionysus2/



(a) 静态超图　　(b) 以点的形式存储的时间信息
以及每条超边对应的时间区间。

图 2：玩具示例时序超图。

不过，我们的处理流程可以适配三种时间归属类型中的任意一种或全部。回到图 2a 中的玩具示例超图 H，我们将时间信息表示为一组时间区间，用于标明每条边处于活动状态的时间（例如，E_2 在点区间 $[0,0]$ 和区间 $[7,8]$ 内处于活动状态）。

2.2 超图快照的滑动窗口

滑动窗口过程是信号处理中的常见环节，其中时间序列或信号被划分为离散窗口，这些窗口沿着时间域滑动。具体而言，给定时间域 $[t_0,t_f]$、窗口大小 w 和步长 s，我们创建一组覆盖该时间域区间的窗口，如下所示

$$\mathcal{W} = \{[t_0,t_0+w],[t_0+s,t_0+s+w],[t_0+2s,t_0+2s+w],\ldots,[t_0+\ell s,t_0+\ell s+w]\}, \tag{1}$$

窗口大小和步长应满足 $s \leq w$。这样，所有窗口的并集便覆盖整个时间域，且相邻窗口之间的交集非空。

对于每个滑动窗口 $W_i \in \mathcal{W}$，我们根据滑动窗口时间区间 W_i 与时序超图中各超边对应的时间区间集合之间的相交条件，创建一个子超图快照。本文将这些时间区间视为闭区间。具体而言，若某条超边的时间区间集合与滑动窗口时间区间 W_i 之间存在非空交集，则将该超边纳入快照。我们将其形式化为

$$H_i = \{E_j \in E \mid I(E_j) \cap W_i \neq \emptyset\}, \tag{2}$$

其中，$E_j \in E$ 是静态超图中边集的一条边，$I(E_j)$ 是边 E_j 对应的时间区间集合。由此得到的子超图快照集合 $\mathcal{W}$ 为

$$\mathcal{H} = \{H_0, H_1, \ldots, H_t, \ldots, H_\ell\}.$$

我们可以将这一集合表示为离散动力过程 $H_t \mapsto H_{t+1}$，以深入了解底层系统的动力学。

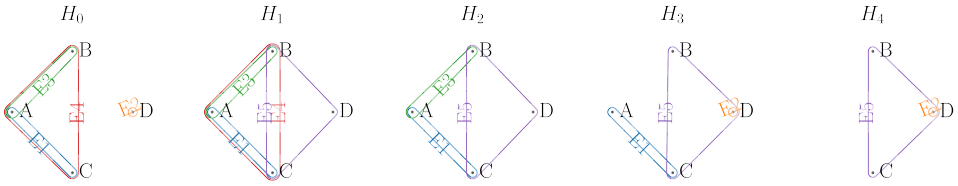


图 3 ： 滑动窗口过程生成的子超图序列 $\mathcal{H}$ 及其对应的 AS　　Cs.

为了演示如何通过滑动窗口过程获取超图快照，我们使用图 2 中的玩具示例时序超图，以及窗口参数 $w = 2$ 和 $s = 2$。使用这些参数，我们得到的滑动窗口如下：

$$\mathcal{W} = \{[0,2],[2,4],[4,6],[6,8],[8,10]\}.$$

每个窗口对应的超图是静态 H 的子集，其生成取决于窗口与每条边相关活动区间的重叠情况。例如，根据 W_2 与图 2b 所示各边的时间区间集合之间的重叠情况，窗口 $W_2 = [4,6]$ 对应的超图H_2 包含边 $\{E_1,E_3,E_5\}$。此外，现在每个超图都关联了索引和时间。索引如前所述（例如，H_2 的索引为 2），时间则是相应窗口的平均时间（例如，W_2 的平均时间为 $(4+6)/2 = 5$）。使用滑动窗口应用这一超图快照生成方法，可得到图 3 所示的五个超图。

2.3 超图的关联抽象单纯复形

ASC K 是由单形组成的集合，其中单形 $\sigma \subseteq P$ 是点集 P 中 n 个点的子集，单形维数为 $n-1$。因此，点（1 条边）是 0-单形，线（2 条边）是 1-单形，三角形（3 条边）是 2-单形，依此类推。若 $\sigma \subseteq \tau$，且 τ 是另一个单形，则将单形 σ 称为一个面。此外，维数为 $n-1$ 的单形 σ 必须在面关系下封闭，也就是说，它的所有子单形（面）都属于该单形的幂集。ASC 的维数是其最大单形的维数。ASC 常用于表示几何结构，因此也称为几何单纯复形。不过，出于纯组合学目的，我们也可以称其为抽象单纯复形。

我们可以利用与每条超边相关联的单形，并构建面关系下的闭包（即每条超边的幂集），来生成超图 [25] 的关联抽象单纯复形。

为运用之字形持久性研究超图快照拓扑的变化，我们首先需要将超图快照集合 $\mathcal{H}$ 表示为一系列 ASC $\mathcal{K}$，随后将用它们来构建之字形持久性模。尽管有多种方法可将超图表示为 ASC [14],，我们采用了对文献 [25] 中关联抽象单纯复形方法的改编。超图H 的关联抽象单纯复形定义为

$$K(H) = \{\sigma \in \mathcal{P}(E_i) \setminus \emptyset \mid E_i \in E\}, \tag{3}$$

其中，E 是超图H的边集，$E_i \in E$，而 $\mathcal{P}(E_i)$是E_i的幂集。公式 3 为计算之字形持久性提供了一个初步起点，但计算上十分繁重。具体而言，对于一条规模较大的k阶超边，计算需要

$$\sum_{j=0}^{k} \binom{k+1}{j+1} = 2^{k+1} - 1$$

子单形。然而，计算维数p 的同调只需将大小为$p+1$ 的单形纳入 ASC。因此，我们将修正后的关联抽象单纯复形定义为

$$K(H,p) = \{\sigma \in \mathcal{P}_{p+1}(E_i) \setminus \emptyset \mid E_i \in E\}, \tag{4}$$

其中，$\mathcal{P}_{p+1}$ 是经过修改的幂集，仅包含集合中大小不超过$p+1$ 或 $\binom{E_i}{p+1}$的元素。修改后的关联 ASC 通过仅要求

$$\sum_{j=0}^{p+1} \binom{k+1}{j+1}$$

一个k超边的子单形，从而降低了计算需求。

将式（4）应用于 $\mathcal{H}$ 中的每个超图，即可得到相应的 ASC 序列 $\mathcal{K}$。图 3 所示的超图快照 $\mathcal{H}$ 对应的修改后关联抽象单纯复形 $\mathcal{K}$ 如图 4 所示。

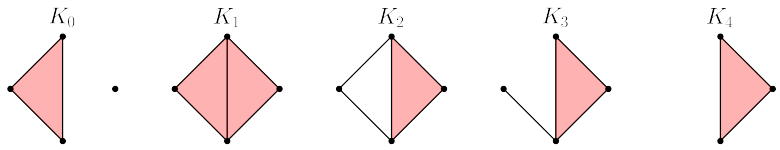


图 4：图 3 中超图快照对应的抽象单纯复形序列。

2.4 单纯同调

单纯同调是一种通过计数 ASC 中p维孔洞数量来研究其形状的代数方法，其中$p = 0$ 是连通分支，$p = 1$ 是图三角形，$p = 2$ 是三维空心四面体，依此类推。我们可以将 ASC K 中p维孔洞的集合表示为贝蒂向量 $\beta(K) = [b_0, b_1, b2, \ldots]$，其中$b_p$是$p$维孔洞的数量，称为贝蒂数。本文不概述贝蒂数的计算细节，但读者可参阅 [24,21] 了解正式介绍。

通过计算图 4 中 ASC 序列的贝蒂数，我们得到图 5 中的贝蒂向量。这些贝蒂

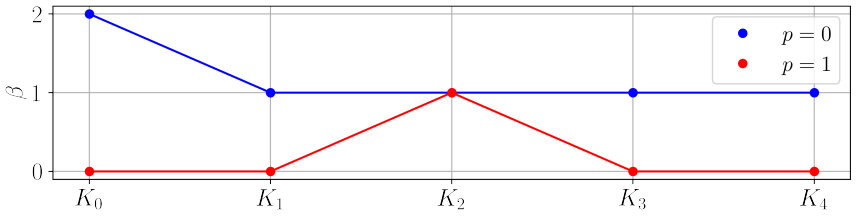


图 5：图 4 中 ASC 的贝蒂数。

数能够反映图 3 中超图快照拓扑的变化；然而，它们无法捕捉快照之间拓扑的关联信息。例如，通过观察超图快照，我们知道有一个主要连通分量贯穿整个 ASC 序列，但无法直接从贝蒂数中得知这一信息。贝蒂数无法完整呈现这个连通分量始终存在的情况。尽管它们能说明每个快照中至少有一个连通分量，但要得到相同的贝蒂向量，各快照中的这些连通分量不一定必须是同一个。因此，我们需要使用一种方法来追踪 ASC 序列中同调如何变化以及彼此之间的关联。为此，我们采用之字形持久同调。

2.5 之字形持久同调

本节将从概念层面介绍持久同调，以及它如何推广为之字形持久同调。我们建议阅读 [24,21]，以深入了解持久同调。

持久同调 [33], 是一种拓扑数据分析（TDA）领域中的滤过工具 [10,33],，用于了解数据集在多个维度和滤过值下的形状与大小。例如，它可以测量连通分支（零维）、孔洞（一维）、空腔（二维）及更高维的类似结构，同时也能大致了解它们的大小或几何形态。持久同调通过参数化滤过来检测同调群的生成（出现）与消亡（消失），从而测量这些形状。

为了计算持久同调，我们对数据集应用参数化函数，生成一系列嵌套的 ASC。

$$K_0 \subseteq K_1 \subseteq K_2 \subseteq \ldots \subseteq K_n. \tag{5}$$

随后，我们可以计算每个复形的p 维同调$H_p(K_i)$，它是一个向量空间，表示空间的p维结构，例如连通分量、洞、空腔和更高维特征。然而，这些信息尚未揭示各个 ASC 的同调与下一个 ASC 的同调之间有何关联。为获得这一信息，持久同调用 ASCs 上的包含映射在向量空间之间诱导线性映射，从而构造出一种称为持久模的结构 $\mathcal{V}$:

$$H_p(K_{\alpha_0}) \hookrightarrow H_p(K_{\alpha_1}) \hookrightarrow H_p(K_{\alpha_2}) \hookrightarrow \ldots \hookrightarrow H_p(K_{\alpha_n}), \tag{6}$$

其中，$\hookrightarrow$ 是由 ASC 之间的包含映射诱导的映射。需要注意的是，在 ASC 序列中，每个顶点都必须唯一且始终以一致的方式标识。

该对象中不同维度的同调类的出现与消失可以被追踪，并汇总为持久性条形码（也称为持久图） $\mathcal{D} = \{D_0, D1, \ldots, D_p\}$。对于每个在$K_b$ 出现、并在K_d消失的同调生成元，我们在条形码中绘制一个时间区间 $[b,d]$ 。这些时间区间共同构成持久性条形码，即持久区间的集合（在持久图中也称为持久性对）。

这一持久同调框架可直接用于研究超图：如 [25],所述，通过生成超图的一系列子集 ASC 表示，可以从超图生成持久模 $\mathcal{V}$。然而，持久同调的一个局限是，要构成持久模（如式 (5) 所示），它要求后续的每个 ASC 都是前一个 ASC 的子集；这意味着在每一步中，我们都不能在下一个 ASC 中移除单形。现实应用中有许多情况会涉及一系列由参数确定的 ASC，其中单形可以在整个序列中进入或离开复形。为解决这一问题，可以应用之字形持久性 [7] ，它允许 ASC 序列中的包含方向任意变化：

$$K_0 \leftrightarrow K_1 \leftrightarrow K_2 \leftrightarrow \ldots \leftrightarrow K_n, \tag{7}$$

其中， $\leftrightarrow$ 表示两个包含映射之一：$\hookleftarrow$ 或 $\hookrightarrow$。该定义的一个常见特殊情况是左右包含映射交替出现，形成之字形。对于大多数使用之字形持久同调的数据分析应用，我们会通过将原始 ASC 与相邻 ASC 的并集或交集交织在一起，人为构造出这种形式的 ASC 序列。例如，在图 6a 中，我们使用图 3 所示原始超图快照的关联抽象单纯复形之间的并集。这个交织的 ASC 序列满足之字形包含映射方向的标准，即

$$K_0 \hookrightarrow K_{0,1} \hookleftarrow K_1 \hookrightarrow K_{1,2} \hookleftarrow K_2 \hookrightarrow \ldots \hookleftarrow K_{\ell-1} \hookrightarrow K_{\ell-1,\ell} \hookleftarrow K_\ell. \tag{8}$$

对于并集，或

$$K_0 \hookleftarrow K_{0,1} \hookrightarrow K_1 \hookleftarrow K_{1,2} \hookrightarrow K_2 \hookleftarrow \ldots \hookrightarrow K_{\ell-1} \hookleftarrow K_{\ell-1,\ell} \hookrightarrow K_\ell \tag{9}$$

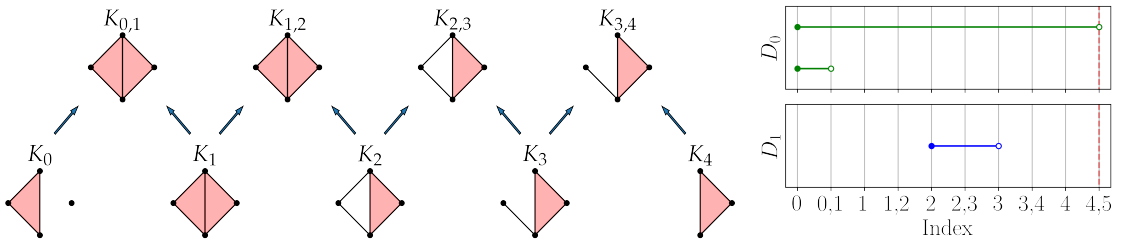
对于交集，其中 $K_{i,\ i+1} = K_i \cup K_{i+1}$。

包含映射被扩展为同调群之间的线性映射，从而得到之字形持久模，用于追踪公式 (8) 或 (9) 中同调的变化，正如标准持久同调中的情形一样。聚焦于并集的情况，之字形持久同调模为

$$H_p(K_0) \hookrightarrow H_p(K_{0,1}) \hookleftarrow H_p(K_1) \hookrightarrow H_p(K_{1,2}) \hookleftarrow H_p(K_2) \hookrightarrow \ldots \hookleftarrow H_p(K_{n-1}) \hookrightarrow H_p(K_{n-1,n}) \hookleftarrow H_p(K_n). \tag{10}$$

在持久同调中，利用同调群之间的线性映射来跟踪标准滤过中的持久配对的同一代数，也使我们能够根据之字形持久模计算同调特征在何处（何时）出生和消亡，不过其中一些直观性有所损失。具体而言，我们仍然可以使用持久图 $D = \{D_0, D1, \ldots, D_p\}$ 来跟踪持久同调，该持久图由半开区间（持久配对） $[b,d)$ 组成；不过，现在我们使用 ASC 的索引作为出生与死亡时间，而不是滤过参数。例如，如果一维同调（即一个环）在K_2 处出现，并持续存在直到在K_3, 处消失，我们就将其表示为持久配对（2,3）。如果某个类在并集（或交集）复形$K_{i,i+1}$处出现或消失，我们使用半索引配对$i, i+1$。如果某个拓扑特征在之字形持久模的最后一个 ASC 中持续存在，我们将其死亡时间设为超出最后一个索引，并记为配对 $\ell, \ell+1$，其中 ℓ是 ASC 的数量（不包括交织的并集或交集）。

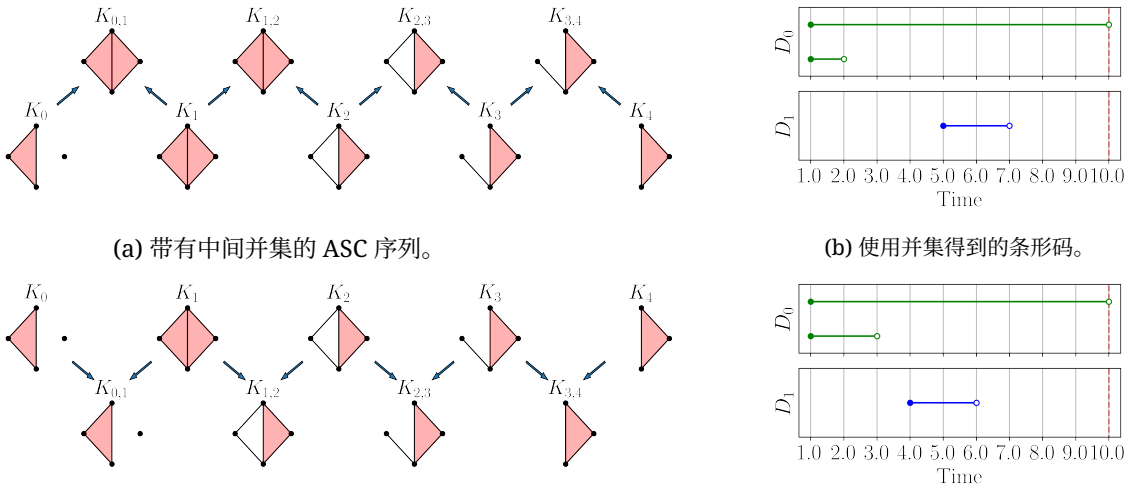
为展示之字形持久性如何追踪一系列 ASC 中变化的拓扑，我们使用图 4 中简单的 ASC 序列。这些 ASC 是通过第2.2节所述的滑动窗口过程，从图 2 的玩具示例中得出的



(a) 带有并集的 ASC 序列。 (b) 之字形持久性条形码。

图 6：之字形持久性模 块及图 2 中引入的玩具示例在维度 0 和 1 下得到的条形码 .

。作为将之字形持久性应用于研究时序超图的首个示例，我们回到玩具示例。我们利用 ASC 之间的并集，得到图 6a 中所示的 ASC：$[K_0, K^{0,1}, K1, \ldots\ldots, K^{3,4}, K_4]$，以及图 6b 中相应的之字形持久性条形码。在这个例子中，我们只研究维度 0 和 1 的拓扑变化，因为不存在更高维的特征。ASC 的同调中有两个主要变化，它们体现在持久性条形码中。对于维度 0，我们追踪连通分支及其相互关系。在K_0 处，我们有两个连通分支（作为三角形的 2-单形和作为点的 0-单形）。因此，我们将这两个分支同时出现的索引设为其诞生时刻：0。接下来，在 $K^{0,1}$ 处，这两个分支合并为两个相连的 2-单形。分支的合并会导致其中一个分支消亡，而另一个继续存在；较小的分支（即 0-单形）在索引 0，1 处消亡，其持久区间为 (0，(0，1))，如图 6b 的D_0条形码所示。合并后的分支此后不再分裂，也不再与其他分支合并，因此它在剩余的持久模中持续存在，直到K_4 之后（即索引 4，5 处）最终消亡（以红色虚线表示），其在D_0中的持久区间为 (0，(4，5))。转到维度 1，我们现在关注持久性条形码如何呈现持久模中环的形成与消失。一个环首先在K_2 处形成，并持续到K_3。因此，这一特征在图 6b 的D_1 中表示为持久区间 (2，3)。这个例子凸显了之字形持久性如何捕捉一系列 ASC 的拓扑变化。



(a) 带有中间并集的 ASC 序列。 (b) 使用并集得到的条形码。

(c) 包含中间交集的 ASC 序列。 (d) 使用交集得到的条形码。

图 7：由滑动窗口超图快照得到的 ASC 序列，分别对应并集和交集。具有与时间相关的 ASC 的时序超图示例的之字形持久性条形码。

在这项工作中，我们关注时序超图的分析，因此希望条形码追踪同调出现和消失的时间，而不是索引。为此，我们将索引替换为与每个 ASC 相关联的时间窗口的平均时间，如图 7 所示。对于中间的 ASC（并集或交集），我们使用两个窗口的平均时间。图 6b 和图 7b 中的 ASC 序列唯一的区别是，在计算之字形持久性时，图 7b 使用了与 ASC 相关联的窗口时间。因此，持久性条形码的横轴表示时间，其中D_0 中的两个时间区间和D_1 中的一个时间区间具有相同的来源（生成元），如图 6b 所述。

图 7 中的结果条形码表明，交织 ASC 的交集方法和并集方法得到的条形码相似。在对本工作研究的数据集应用之字形持久性时，我们也发现了相同的结果。在本工作的其余部分，我们将使用并集方法，通过之字形持久性研究时序超图。

3 应用

3.1 社交网络分析

为展示如何通过之字形持久性分析时序超图数据，我们使用与 COVID 相关的 Reddit 子版块数据。这些数据称为 PAPERCRANE 数据集 [5]。
我们使用的数据集子集涵盖 2020 年 1 月 20 日至 2020 年 3 月 31 日。这一部分记录了 COVID-19 暴发初期各子版块的形成情况。数据集中这一时期活跃的 COVID-19 相关子版块列于表 1，并附有帖子串数量和作者数量的汇总统计量。

表 1：PAPERCRANE 数据集中的 COVID 相关子版块，以及各子版块的帖子串数量和作者数量

子版块	活跃日期	讨论串	作者
CCP_病毒	3月27日—3月31日	169	79
COVID-19	2月15日—3月31日	8668	22020
COVID-19阳性	3月13日—3月31日	1462	6682
中国 _Flu	1月20日—3月31日	55466	62944
冠状病毒	1月20日—3月31日	153025	396427
加利福尼亚州冠状病毒	3月1日—3月31日	2930	5370
冠状病毒衰退	3月19日—3月31日	1574	6548
英国冠状病毒	2月15日—3月31日	8654	10230
CoronavirusUS	2月16日—3月31日	18867	29809
Covid2019	2月16日—3月31日	2437	1531
cvnews	1月25日—3月31日	4233	2181
nCoV	1月20日—3月31日	3949	1902

在此分析中，我们仅使用新冠病毒子版块，因为其规模适中且易于解读。超边的时间区间根据作者交互信息构建。我们根据作者在各讨论串中发帖的首次和末次时间构建超边时间区间。这些时间区间显示在图 8 的上方子图中。

我们将窗口大小设为 1 小时，步长设为 15 分钟。这样的窗口大小能够提供必要的粒度，以观察子版块动态的变化。应用此滑动窗口后，共得到 6899 个窗口。图 8 展示了每个超图快照中的节点数和边数。这项初步数据探索表明，子版块规模起初不断扩大，并在创建约两周后（即第 14 天）达到人气峰值。此后，规模稳步下降。图 8 上方子图中的边时间区间显示，大多数区间都很短，只有少数区间持续时间较长，最长可达 38 天。这项初步探索并未揭示底层超图快照的形状如何演变。

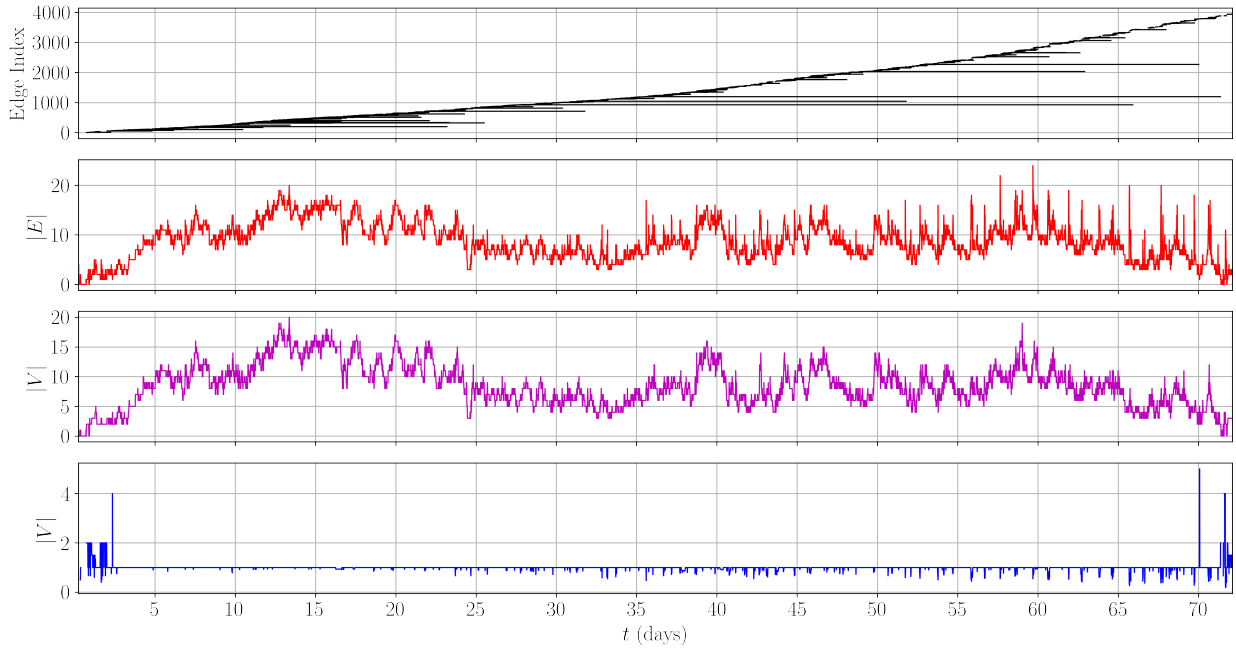


图 8：时序超图快照规模的汇总统计量。顶部显示与每条边相关联的时间区间（按开始时间排序），中间图显示超图快照中的边数，底部图显示超图快照中的顶点数。

关于底层网络，有许多问题无法通过这些简单的汇总统计量直接回答。例如，每个帖子串是由独特的作者主导，还是许多帖子串共享用户？这个社交网络是稠密、中心化，还是碎片化的？这些特征会随时间变化吗？

了解超图快照的拓扑摘要，对于理解正在发生的通信类型至关重要。例如，许多一维同调特征代表通信结构中彼此断开的对话或孔洞。不过，只需考察每个快照的贝蒂序列，也能捕捉到这一点。之字形持久性还能根据孔洞的持续时间，反映其显著程度。设想一个通信中的孔洞持续了数天，这可能意味着整个社区长期缺乏信息交流。此外，这些汇总统计量并未提供有关子版块中的帖子串如何相互关联及其持续时间的信息。利用之字形持久性，我们可以通过零维同调捕捉通信持续时间的信息。零维之字形持久性条形码中的长时间区间，代表一段持续了很长时间的对话。图 9 展示了通过对超图快照的关联 ASC 取并集得到的之字形持久性条形码。

首先，我们可以通过零维条形码中一个贯穿子版块几乎整个持续时长的主要连通分量，来呈现社交网络的碎片化程度。此外，零维中的短时间区间是其他旁支对话的特征；这些对话可能从主要对话中分离出来、并入其中，或完全独立。图 9 中第 10 天的超图快照展示了其中一段对话：主要连通分量由除一个讨论串外的所有帖子串构成，而那个讨论串仅涉及两位作者。主要连通分量的存在表明，子版块中的许多帖子串之间至少共享一位作者。

我们还可以证明，该网络的中心化程度会随时间发生变化。具体而言，在存在许多D_1持久区间的时段，我们知道该网络包含多个环，这是非中心化社交网络的典型特征。社交超图快照从中心化转为非中心化，很可能是由于活跃作者数量的变化以及社交网络动态的分岔所致。例如，在图 9 中第 10 天的快照里，所捕获的超图快照的主连通分量中有一个主环，且该主连通分量没有明显的中心化结构。然而，大约一周后的第 18 天，出现了

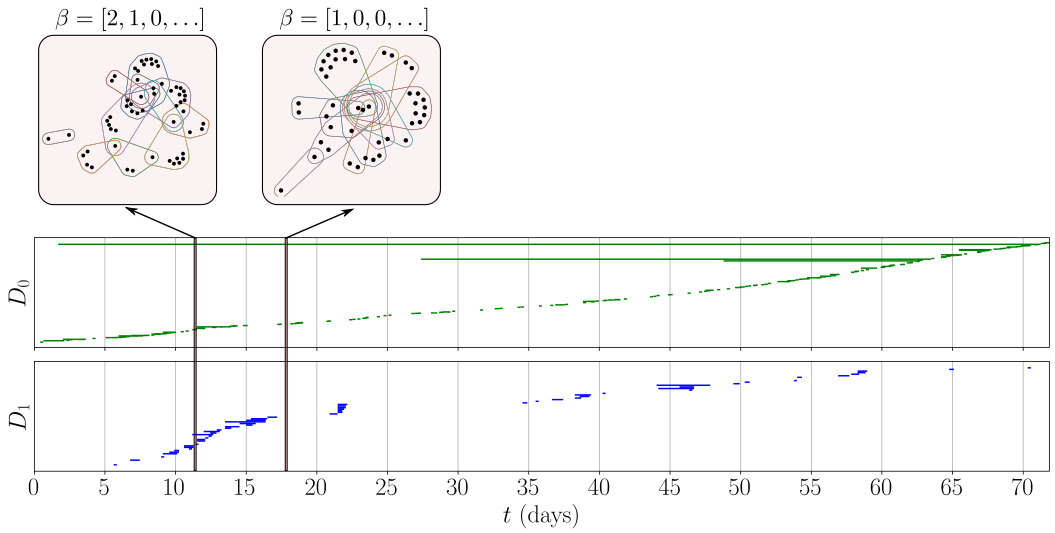


图 9：CCP_病毒子版块的时序超图表示的之字形持久性，包含超图快照及其关联抽象单纯复形示例。

超图呈现出明显的中心化结构，且没有一维特征。当一维特征数量很少（或为零），且只有一个连通分量时，之字形持久性便能揭示超图及其底层社交网络的中心化程度。

3.2 网络数据分析

为了分析网络数据，我们使用美国国防高级研究计划局（DARPA）创建的 Operationally Transparent Cyber 数据集（OpTC） [2] 。该数据集包含为期一周的数百台 Windows 主机的网络和主机日志记录。数据集包含两类用户活动：良性活动和恶意活动。恶意活动发生在为期三天的时间内，其间执行了多次攻击。

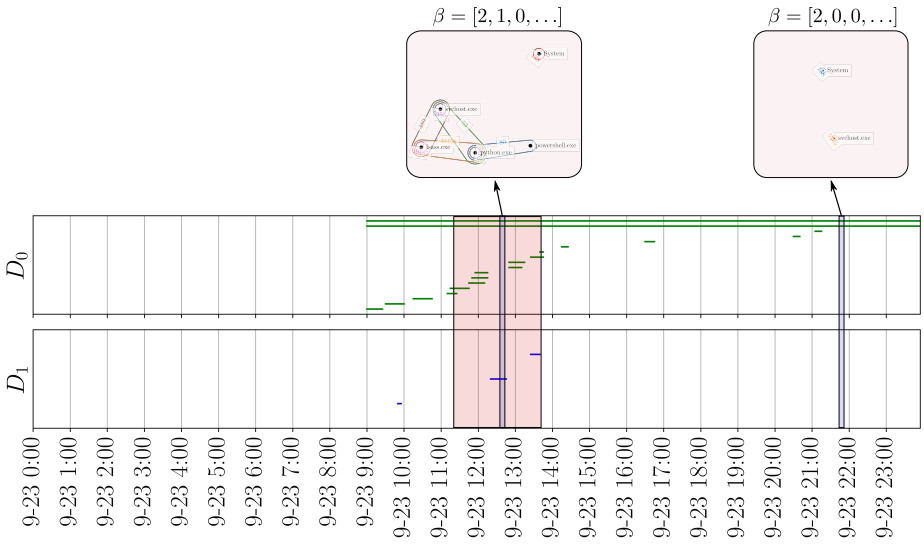
我们的目标是分析并展示这些攻击如何体现在根据数据日志构建的超图的之字形持久条形码中。数据日志由 64 列组成，用于描述网络中的每项操作。本节仅使用时间戳、src_ip、image_path 和 dest_port，因为这些信息是利用之字形持久性构建我们所研究数据的时序超图表示所必需的。

我们再次使用滑动窗口过程来构建超图快照，但现在与每条超边关联的时间区间仅包含时间点，因为网络数据只有操作发生时的时间戳。我们使用宽度为$w = 30$ 分钟、步长为$s = 5$ 分钟的滑动窗口。我们根据恶意活动持续约 2 小时这一情况选择了该窗口大小；30 分钟的窗口足够精细，可以捕捉从良性状态转变为恶意状态的过程。

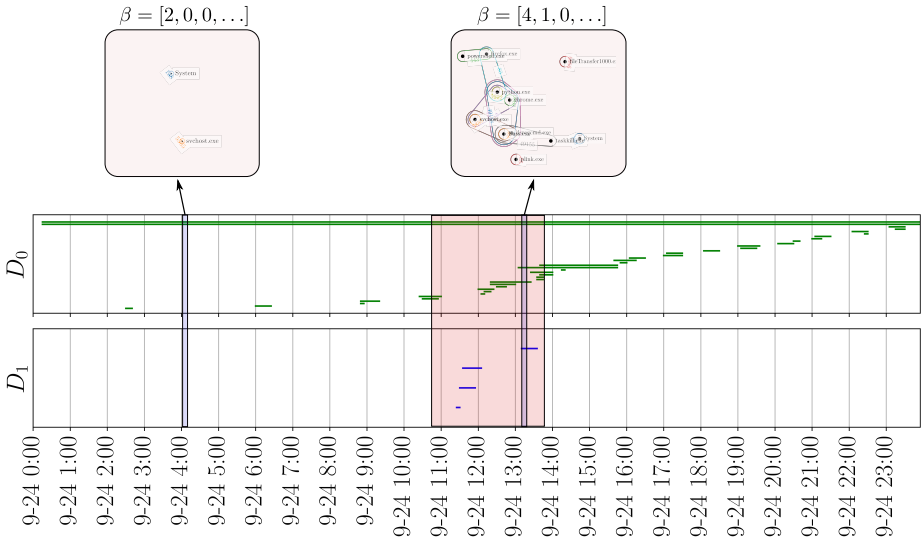
为了展示之字形持久性如何检测网络攻击，我们将考察两个不同主机上发生的两起恶意活动。具体而言，我们调查了两起网络攻击案例：第一起发生于2019年9月23日，攻击者是红队代理 LU- AVR71T，源 IP 为 142.20.56.202，目标主机为 201；第二起发生于2019年9月24日，攻击者是代理 4BW2MKUF，源 IP 为142.20.57.246，目标主机为 501。第一段攻击序列大约从2019年9月23日11:23持续到13:24，第二段序列则大约从10:46持续到13:11。

这些超图以目标端口作为超边、图像路径作为节点构建而成。这种构造方式捕捉了网络数据的结构：作为超边的目标端口反映了所用操作（图像路径）之间的关系。此外，对于单个源 IP，我们只使用了完整数据的一个子集。通过仅考察这个子超图，我们可以捕捉与该源 IP 关联的特定代理的信息。

第一组攻击中，与目的端口/图像路径超图快照相关的之字形持久条形码如图10a所示。9:00之前没有网络活动，因此也没有条形码。



(a) 红队代理 LUAVR71T 在主机 201 上于 2019年9月23日活动时的之字形持久性条形码，源 IP 为 142.20.56.202。



(b) 红队代理 4BW2MKUF 在主机 501 上于 2019年9月24日活动时的之字形持久性条形码，源 IP 为 142.20.57.246。

图 10：攻击发生于 23 日和 24 日期间的 OpTC 数据之字形持久性条形码，以及两个时间窗口中的超图示例。红色高亮区域表示红队代理进行攻击活动的时间。

这一时段。从11:23到13:24以红色突出显示的区域是网络攻击的活跃时段。在此区域中，我们重点展示了一个大约从12:35开始的时间窗口所对应的特定超图，它是恶意活动的典型例子。此外，大约在21:50，我们还展示了另一个体现常见良性活动的典型时间窗口。在此活动期间，通常只有两个单点会持续存在。2019年9月24日第二段恶意活动中，也展示了类似的一对恶意活动与良性活动超图。然而，快照未能体现的是：恶意活动期间快照的动态变化及拓扑的快速变化，以及良性活动期间相对稳定的动态变化和简单的拓扑结构。

之字形持久性能够捕捉恶意网络活动所特有的动态变化和拓扑特征。从图10所示的两组恶意活动序列中，D_0 和 D_1 的条形码均体现了这一点。

具体而言，在恶意活动期间，D_0 中往往会出现更多短寿命的持久配对，而D_1中则会出现一维同调。相比之下，在良性活动期间，几乎不会出现一维同调，且通过D_0捕捉到的连通分量数量几乎没有变化。

4 结论

在本研究中，我们开发了用于研究时序超图的之字形持久性实现。为了展示该方法的功能，我们将其应用于研究表示为时序超图的社交网络和网络安全数据。对于社交网络分析，我们展示了所得的之字形持久性条形码如何捕捉时序超图拓扑的动态变化，而这种拓扑能够通过D_1捕捉超图中心性的变化信息。此外，我们展示了该对话由一个贯穿所研究社交网络整个时间段的主要连通分量组成。在研究网络数据时，我们发现利用之字形持久性能够检测出恶意活动与良性活动的区别。我们展示了，在恶意活动期间，D_1 中往往存在持久性对，并且与良性活动期间相比，D_0 中的持久性对也更多。

这一方法的后续工作包括研究如何利用之字形持久性图的向量化技术，实现网络安全分析的自动化。我们还计划研究如何利用时序超图表示和之字形持久性来检测社交网络数据中的机器人活动。

参考文献

[1] Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., Ziegelmeier, L.：持久性图像：持久同调的稳定向量表示。机器学习研究杂志 **18**(8), 1–35 (2017), http://jmlr.org/papers/v18/16-337.html[2] Agency, D.A.R.P.：操作透明网络空间（optc）数据发布（2020）[3] Aktas, M.E., Akbas, E., Fatmaoui, A.E.：网络的持久同调：方法与应用。应用网络科学**4**(1) (2019年8月)。https://doi.org/10.1007/s41109-019-0179-3[4] Amézquita, E.J., Quigley, M.Y., Ophelders, T., Munch, E., Chitwood, D.H.：未来事物的形态：从分子到生物体的拓扑数据分析与生物学。发育动力学**249**(7), 816–833 (2020年4月)。https://doi.org/10.1002/dvdy.175[5] Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., Blackburn, J.：Pushshift Reddit 数据集。PUSHSHIFT (2020年1月)。https://doi.org/https://doi.org/10.5281/zenodo.3608135，reddit-hazelnut，由社会网络问题研讨会准备（2022年1月24日至2月4日）。加拿大渥太华。该数据是通过 pushshift.io API 获取的 Reddit 数据的衍生数据，涵盖2019年1月1日至2019年2月28日。[6] Bubenik, P.：使用持久景观进行统计拓扑数据分析。机器学习研究杂志**16**(3), 77–102 (2015), http://jmlr.org/papers/v16/bubenik15a.html[7] Carlsson, G., de Silva, V.：之字形持久性。计算数学基础**10**(4), 367–405 (2010年4月)。https://doi.org/10.1007/s10208-010-9066-0[8] Cencetti, G., Battiston, F., Lepri, B., Karsai, M.：社交网络中高阶交互的时序特性。《科学报告》**11**(1) (2021年3月)。https://doi.org/10.1038/s41598-021-86469-8[9] David Boyce, B.R.：动态交通网络建模。施普林格·柏林·海德堡 (2012)[10] Edelsbrunner, Letscher, Zomorodian：拓扑持久性与简化。离散与计算几何**28**(4), 511–533 (2002年11月)。https://doi.org/10.1007/s00454-002-2885-2[11] Estrada, E., Rodríguez-Velázquez, J.A.：复杂超网络中的子图中心性与聚类。物理学 A：统计力学及其应用**364**, 581–594 (2006年5月)。https://doi.org/10.1016/j. physa.2005.12.002[12] Feng, S., Heath, E., Jefferson, B., Joslyn, C., Kvinge, H., Mitchell, H.D., Praggastis, B., Eisfeld, A.J., Sims, A.C., Thackray, L.B., Fan, S., Walters, K.B., Halfmann, P.J., Westhoff-Smith, D., Tan, Q., Menachery, V.D., Sheahan, T.P., Cockrell, A.S., Kocher, J.F., Stratton, K.G., Heller, N.C., Bramer, L.M., Diamond, M.S., Baric, R.S., Waters, K.M., Kawaoka, Y., McDermott, J.E., Purvine, E.：用于识别病原病毒反应关键基因的生物网络超图模型。BMC 生物信息学**22**(1) (2021年5月)。https://doi.org/10.1186/ s12859-021-04197-2[13] Fischer, M.T., Arya, D., Streeb, D., Seebacher, D., Keim, D.A., Worring, M.：用于时序超图模型探索的可视分析。IEEE 可视化与计算机图形学汇刊**27**(2), 550–560 (2021年2月)。https://doi.org/10.1109/tvcg.2020.3030408[14] Gasparovic, E., Gommel, M., Purvine, E., Sazdanovic, R., Wang, B., Wang, Y., Ziegelmeier, L.：图与超图的同调 (2021年5月), https://www.youtube.com/watch?v=XeNBysFcwOw[15] Golczynski, A., Emanuello, J.A.：通过基于自然语言处理的日志嵌入端到端检测恶意网络行为。arXiv 预印本 arXiv:2108.12276 (2021)[16] Hanselmann, M., Strauss, T., Dormann, K., Ulmer, H.：Canet：一种用于高维 CAN 总线数据的无监督入侵检测系统。IEEE Access**8**, 58194–58205 (2020)[17] Harary, F., Gupta, G.：动态图模型。数学与计算机建模**25**(7), 79–87 (1997年4月)。https://doi.org/10.1016/s0895-7177(97)00050-2[18] Husein, I., Mawengkang, H., Suwilo, S., Mardiningsih：动态网络中的传染病传播建模。物理学杂志：会议系列**1255**(1), 012052 (2019年8月)。https://doi.org/10. 1088/1742-6596/1255/1/012052[19] Joslyn, C.A., Aksoy, S.G., Callahan, T.J., Hunter, L.E., Jefferson, B., Praggastis, B., Purvine, E., Tripodi, I.J.：超网络科学：从多维网络到计算拓扑。载于：复杂系统统一主题 X，第377–392页。施普林格国际出版 (2021)。https://doi.org/10.1007/ 978-3-030-67318-5

[20] Khasawneh, F., Munch, E.：利用持久同调检测车削中的颤振。《机械系统与信号处理》**70-71**, 527 – 541 (2016年). https://doi.org/10.1016/j.ymssp.2015.09.046[21] Munch, E.：拓扑数据分析用户指南。《学习分析杂志》**4**(2) (2017年7月). https: //doi.org/10.18608/jla.2017.42.6[22] Myers, A., Munch, E., Khasawneh, F.A.：复杂网络持久同调与动态状态检测。《物理评论E》**100**(2) (2019年8月). https://doi.org/10.1103/physreve.100.022314[23] Myers, A., Muñoz, D., Khasawneh, F., Munch, E.：使用之字形持久性进行时序网络分析 (2022年)[24] Otter, N., Porter, M.A., Tillmann, U., Grindrod, P., Harrington, H.A.：持久同调计算路线图。《EPJ数据科学》**6**(1) (2017年8月). https://doi.org/10.1140/epjds/s13688-017-0109-5[25] Ren, S.：超图持久同调与计算工具——用户综述。《结理论及其应用》**29**(13), 2043007 (2020年11月). https://doi.org/10.1142/s0218216520430075[26] Schäfer, B., Witthaut, D., Timme, M., Latora, V.：电网中动态诱发的级联故障。《自然通讯》**9**(1) (2018年5月). https://doi.org/10.1038/s41467-018-04287-5[27] Skaf, Y., Laubenbacher, R.：生物医学中的拓扑数据分析：综述。《生物医学信息学杂志》**130**, 104082 (2022年6月). https://doi.org/10.1016/j.jbi.2022.104082[28] Skyrms, B., Pemantle, R.：社会网络形成的动态模型。《美国国家科学院院刊》**97**(16), 9340–9346 (2000年8月). https://doi.org/10.1073/pnas.97.16.9340[29] Tempelman, J.R., Khasawneh, F.A.：通过拓扑数据分析探究混沌检测。《物理学D：非线性现象》**406**, 132446 (2020年5月). https://doi.org/10.1016/j.physd.2020.132446[30] Tymochko, S., Munch, E., Khasawneh, F.：使用之字形持久同调检测动力系统中的Hopf分岔。《算法》**13**(11) (2020年10月). https://doi.org/10.3390/a13110278[31] Xu, M., Radhakrishnan, S., Kamarthi, S., Jin, X.：互利型供应商—制造商网络的韧性。《科学报告》**9**(1) (2019年9月). https://doi.org/10.1038/s41598-019-49932-1[32] Yesilli, M.C., Chumley, M.M., Chen, J., Khasawneh, F.A., Guo, Y.：利用拓扑测度探索压电振动冲击处理 (PVST) 中的表面纹理量化。载于：第2卷：制造工艺；制造系统。美国机械工程师学会 (2022年6月). https://doi.org/10.1115/ msec2022-86659[33] Zomorodian, A., Carlsson, G.：计算持久同调。《离散与计算几何》**33**(2), 249–274 (2004年11月). https://doi.org/10.1007/s00454-004-1146-y