流程挖掘中的轨迹编码：综述与基准测试

Sylvio BarbonJr.的里雅斯特大学 的里雅斯特，意大利
sylvio.barbonjunior@units.it

Paolo Ceravolo, Rafael S. Oyamada,Gabriel M. Tavares米兰大学 米兰，意大利 {paolo.ceravolo, rafael.oyamada, gabriel.tavares}@unimi.it

arXiv:2301.02167v1 [cs.LG] 5 Jan 2023

摘要

编码方法广泛应用于多种流程挖掘任务，包括预测性流程监控、异常案例检测、轨迹聚类等。这些方法通常作为预处理步骤执行，负责将复杂信息转换为数值特征空间。大多数论文会任意选择现有的编码方法，或采用基于特定专家知识领域的策略。此外，现有方法通常使用默认超参数，而不评估其他选项。这种做法可能导致诸多缺陷，例如性能欠佳，以及与最先进水平进行不公平的比较。因此，本文旨在通过比较来自不同性质的 27 种方法，从表达能力、可扩展性、相关性和领域无关性等方面，对事件日志编码开展全面综述。据我们所知，这是迄今为止聚焦于流程挖掘中轨迹编码的最全面研究。本研究有助于深化对轨迹编码在流程挖掘流水线中所发挥作用的认识，并阐明编码方法在弥合机器学习模型与流程挖掘之间差距时所涉及的问题、关注点及未来研究方向。

关键词 编码方法 ·流程挖掘 ·异常检测

1 引言

编码方法负责将复杂信息转换为具有代表性的特征空间。在流程挖掘（PM）中，若干任务（例如预测性流程监控、轨迹聚类、异常检测等）必须先对数据进行编码，再将其输入特定算法。正确考虑用户的目标至关重要。例如，如果问题要求解决方案具备可解释性，那么应使用倾向于实现这些目标的方法对数据进行编码。另一方面，如果最关键的要求是空间或时间复杂度，用户就应接受牺牲部分上述优势，以满足这些要求。

在过程挖掘文献中，大多数研究都致力于设计新的算法和分析方法，但很少关注编码方法对现有任务的影响。例如，在预测性流程监控中， [1] 采用词嵌入 方法，将事件日志中的案例映射为实值向量；而 [2] 采用独热编码。 [3], 采用自定义函数，而 [4]则使用*count2vec* （活动发生频率）。因此，若研究人员想要比较这些研究的结果，就会面临一个无法控制的因素：编码的影响尚未得到记录，而且所用方法各不相同。此外，在这项工作中，我们强调，社区采用的替代编码方法寥寥无几，并证明对数据进行任意编码可能导致结果欠佳，且与用户目标不一致。我们

认为，根据数据集的特征，更深入地理解编码方法的影响，对于开发更具可解释性、稳健性和准确性的流程挖掘解决方案至关重要。

我们以异常检测为案例研究，在考虑多个方面的基础上扩展了先前论文 [5] 的研究结果。首先，我们增加了编码方法的数量，并提出了一种新的分类体系，依据不同维度对这些方法进行分类。其次，我们纳入了更多数据集，并考虑更多类型的异常，以扩大特征空间，更好地理解各编码方法在不同数据属性下的表现。第三，我们采用了对流程挖掘从业者有价值的评估标准，可帮助他们根据自身目标选择合适的编码方法。最后，我们对热门流程挖掘任务中使用的编码方法进行了系统综述：预测性监控、轨迹聚类、异常检测、在线过程挖掘，以及流程挖掘中的安全与隐私。

更具体地说，我们首先指出，选择合适的编码方法及其参数都并非易事。随后，我们针对 420 个合成事件日志，使用不同参数对 27 种编码方法进行了广泛的实验评估。我们还讨论了当前过程挖掘文献如何因未考虑编码方法对任何问题领域的影响，而限制了实验的范围。因此，我们讨论研究结果，并将本研究的贡献聚焦于回答以下研究问题：

1. 编码方法在区分问题类别方面的表达能力如何？2. 要获得合适的编码方法，需要投入多少时间和内存？3. 编码方法与算法在流程挖掘任务中的性能之间是否存在相关性？4. 编码方法的通用性如何？也就是说，一种编码方法能否应用于任意流程挖掘任务？

我们根据不同标准提出了具体的评估指标，以回答这些问题。通过深入分析，我们考察了以下标准：表达能力，旨在捕捉所用数据集不同特征中的模式；可扩展性，用于衡量编码方法所耗费的时间和内存使用量；相关性能力，用于将数据特征与算法性能关联起来；以及领域无关性，用于考察编码方法是否依赖于问题领域。我们通过广泛的实验评估说明，选择合适的编码方法可能相当困难，因为在每项评估指标上表现最佳的方法各不相同。因此，本研究的主要贡献包括：

- 对流程挖掘中编码方法的系统综述，以及根据该综述构建的新分类体系。
- 提出新的评估指标，用于衡量流程挖掘任务中编码方法的质量。
- 对流程挖掘中此前从未采用过的若干编码方法进行了深入的实验评估。
- 讨论了流程挖掘编码未来研究的相关见解。

我们将本文的介绍安排如下。首先，在第 2 节中，我们定义了如何选择合适的编码方法及其参数这一问题。在第 3 节中，我们提供了理解本文所需的背景知识。在第 4 节中，我们首先对不同流程挖掘任务中的编码方法进行了系统综述。随后，我们介绍了一个用于编码事件数据的新分类体系，按照算法系列对所发现的编码方法进行组织，描述每种方法，并讨论相关工作。第 5 节介绍了用于实施实验评估的方法论，第 6 节则展示了所开展的实验及其结果。在第 7 节中，我们讨论了从本文中获得的主要见解，并提出未来的研究方向。最后，我们在第 8 节中总结全文。

2 问题定义

本节讨论在流程挖掘任务中任意使用编码方法如何导致性能不佳，并造成评估不公平。如今可用的编码方法种类繁多，针对特定问题选择合适的方法颇具挑战。当前文献中，不同领域已提出多种自动化解决方案，以减少算法和数据科学流水线设计中的人工干预 [6, 7, 8]。我们认为，在流程挖掘中，这一问题更具挑战性，因为事件日志具有其独特性质：事件既可由数值型属性描述，也可由类别属性描述；事件按案例聚合，并受流程控制流的约束。例如，只有具备一定数量的资源，才可能观察到某个事件（如某项活动的执行），而该事件还可能依存于其他先前发生或并发发生的事件。要将所有这些信息浓缩到单一编码方法中并非易事；在实际应用中，每种方法都只能捕捉特定方面的信息。

流程挖掘的编码方法通常借鉴自其他领域。例如，常会采用简单技术，如独热编码方案 [2] 或基于频率的编码方法 [9]。为捕捉事件日志的序列特性，研究人员采用了最初由自然语言处理（自然语言处理）领域提出的方法 [10, 11]。然而，尽管轨迹的序列特性与自然语言句子具有相似之处，我们也必须讨论二者之间的差异。例如，自然语言处理任务通常处理规模庞大的词汇表，即由不重复的单词或令牌构成的集合；而流程通常由规模小得多的词汇表表示（如业务流程活动）。为尝试捕捉更复杂的特征，近期文献开始研究图神经网络 [12]。卷积神经网络也已用于特征提取 [13]。 [14, 15, 12] 引入了类图像数据工程方法。近期，还有多种流水线采用了领域特定编码方法；我们将在第 4 节进一步介绍这些方法。它们利用派生特征，例如 [16] 使用的资源池发现算法，该算法用于编码事件资源。

在我们的工作中，我们强调，采用正确的编码方法并选择最优超参数，可能会直接影响给定任务的最终性能。此外，通过将新算法（例如轨迹聚类算法）与其他解决方案进行比较来评估该算法，但采用了不同的数据输入（即在聚类之前执行了不同的编码步骤），会导致不公平的评估。 [17]强调了这一问题，指出如果两个模型的实现所考虑的特征空间不同，就无法将一个模型与另一个模型进行比较。 [16], 提供了一个简短的示例来说明这一问题，其中作者正在处理预测性监控问题。在评估中，作者采用基线方法，将其提案与现有的长短期记忆网络架构进行比较，而每种架构都基于不同的预处理流程。至于其他预测性监控工作， [16, 18, 19] 采用了词嵌入，而 [20, 21, 22], 使用了传统的独热编码，从而阻碍了对这些研究进行比较。由于流程挖掘社区缺乏可用于算法评估和比较的共享基准测试，这一问题进一步加剧。

为简要说明任意编码事件日志所产生的影响，我们在图 1 中展示了以下场景。我们比较了两种编码算法，改变向量维度参数，将它们应用于两个特征不同的数据集，并衡量随机森林分类器在异常检测问题上的准确率[1]。这两个数据集的基数、异常类型和异常注入率各不相同。第一个数据集包含 $10k$ 条轨迹、 $444k$ 个事件，插入异常率为 20%（在轨迹中插入一个随机活动）；第二个数据集包含 $5k$ 条轨迹、 $221k$ 个事件，返工异常率为 15%（在轨迹中将某个活动重复一次）。如图 1 所示，对于第一个事件日志（事件日志 1），在低维度下，使用*Walklets* 方法对数据进行编码的效果优于使用 *NMF-ADMM* ，而在中、高维度下则效果较差。此外，后者在不同维度下的准确率波动较大。对于第二个事件日志（事件日志 2）， *Walklets* 编码的准确率更加稳定，而*NMF-ADMM*在该数据集上的准确率高于另一个数据集，但始终不如*Walklets*。这个简要示例表明，并不存在适用于所有数据集的最佳编码方法或可直接套用的默认参数设置。

鉴于编码方法所呈现的诸多细微差异，我们通过评估流程挖掘任务的编码方法来应对这一现有限制。在本研究中，我们试图详细展示不同方法在具有不同特征和属性的多个数据集上的表现。首先，我们通过广泛的实验评估表明，每种算法在不同事件日志和流水线上的性能各不相同。我们采用多个指标来概括每种方法的整体行为，并将总体评估重点放在表达能力、可扩展性、相关性和领域无关性上，这些内容将在第 5 节中进一步详述。

3 背景概念

流程挖掘可以定义为一组从事件日志中提取知识的技术 [23]。其目标是提供利用事件数据进行分析，以提取流程相关洞见，即创建专门针对业务流程及其利益相关者的解决方案。

因此，让我们首先考虑 Σ 事件全集，即所有可能事件标识符的集合。 Σ^* 表示在 Σ上所有序列的集合。

定义 3.1（事件、属性）事件可能具有各种属性，例如时间戳、活动、资源等。令 $\mathcal{AN}$ 为属性名称集合。对于任意事件 $e \in \Sigma$ 和属性 $n \in \mathcal{AN}$，$\#_n(e)$ 表示事件 e 的属性 n 的值。通常，值被限制在某个域内。例如，$\#_{activity} \in A$，其中 A 是业务流程中合法活动的全集，例如 $\{a, b, c, d, e\}$。

[1]详细描述 了材料和方法的内容见第 第 6 节。

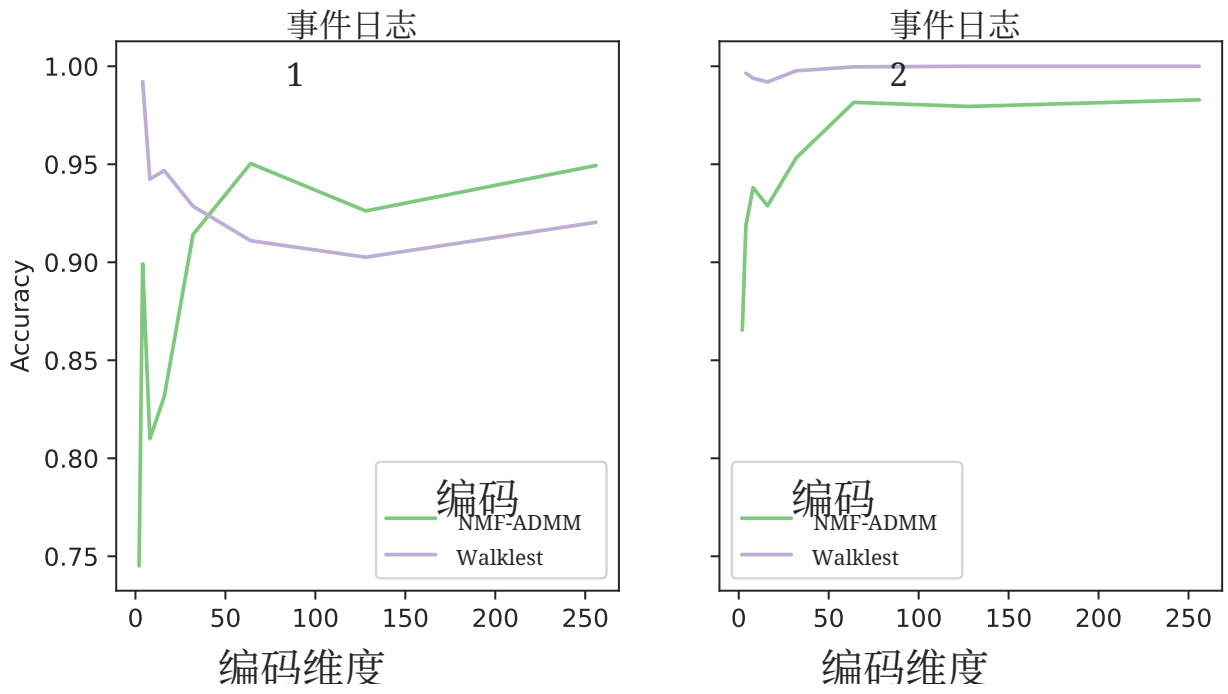


图 1：针对异常检测问题，两个不同数据集所取得性能的简要示例。对于两个数据集，我们固定采用了两种在维度参数化方面不同的基于图的编码方法。

为简化记法，我们将事件活动的名称 $\#_{activity}(e)$ 称为事件本身。因此，$\langle a, b, d\rangle$表示由三个连续事件组成的轨迹。事件也可以用其在序列中的位置表示，例如 e_i，其中 e_n是该序列中的最后一个事件。事件序列构成轨迹 $t \in \Sigma^*$，其定义如下。

定义 3.2（轨迹、子轨迹）在一条轨迹中，每个事件只出现一次，并且时间非递减，即对于 $1 \leq i < j \leq |t|$：$t(i) \neq t(j)$。轨迹也可以表示为一个函数，该函数为其序列中的每个位置生成相应的事件：$t(i \to n) \mapsto \langle e_i, ..., e_n\rangle$。子轨迹是一个序列 $t(i \to j)$，其中 $0 < i \leq j < n$。

现在令 C 为案例全集，即业务案例执行的所有可能标识符的集合。C 是属性 $case \in \mathcal{AN}$的定义域。

定义 3.3（案例、事件日志）我们将案例 $c_i \in C$ 表示为 $\langle a, b, d\rangle_{c_i}$，这意味着所有事件共享同一个案例。例如，对于 c_i，我们有$\#_{case}(e_1) = \#_{case}(e_2) = \#_{case}(e_3)$。事件日志 L 是案例 $L \subseteq \Sigma^*$ 的集合，其中每个事件在日志中只出现一次，即对于任意两个不同的案例，它们的事件交集为空。

在流程挖掘中，编码是一个关键步骤，旨在将事件日志中包含的信息投影到另一个特征空间，以便与聚类等后续算法结合。在本文中，我们通过基准测试编码方法来研究异常检测问题。因此，设$\mathbf{M}$ 为表示事件日志的流程模型，f 为一个测试函数，用于指示日志$\mathbf{L}$ 中的某条轨迹是否为模型$\mathbf{M}$的实例。由此，我们可以将异常检测问题定义如下：

定义 3.4（异常检测）设 $f: \mathbf{L} \to \{R, A\}$ 为一个测试函数，用于判断一条轨迹是正常的（R）还是异常的（A）。如果一条轨迹无法由$\mathbf{M}$完全解析，则认为它是异常的。因此，

$$f(t) = \begin{cases} Regular, & if\ it\ can\ be\ replayed\ by\ \mathbf{M} \\ Anomalous, & otherwise \end{cases} \tag{1}$$

考虑到流程挖掘数据的特殊粒度，即由事件组成的轨迹中包含数值型、类别型和时间类值，本文提出了一种处理事件数据的新分类体系。

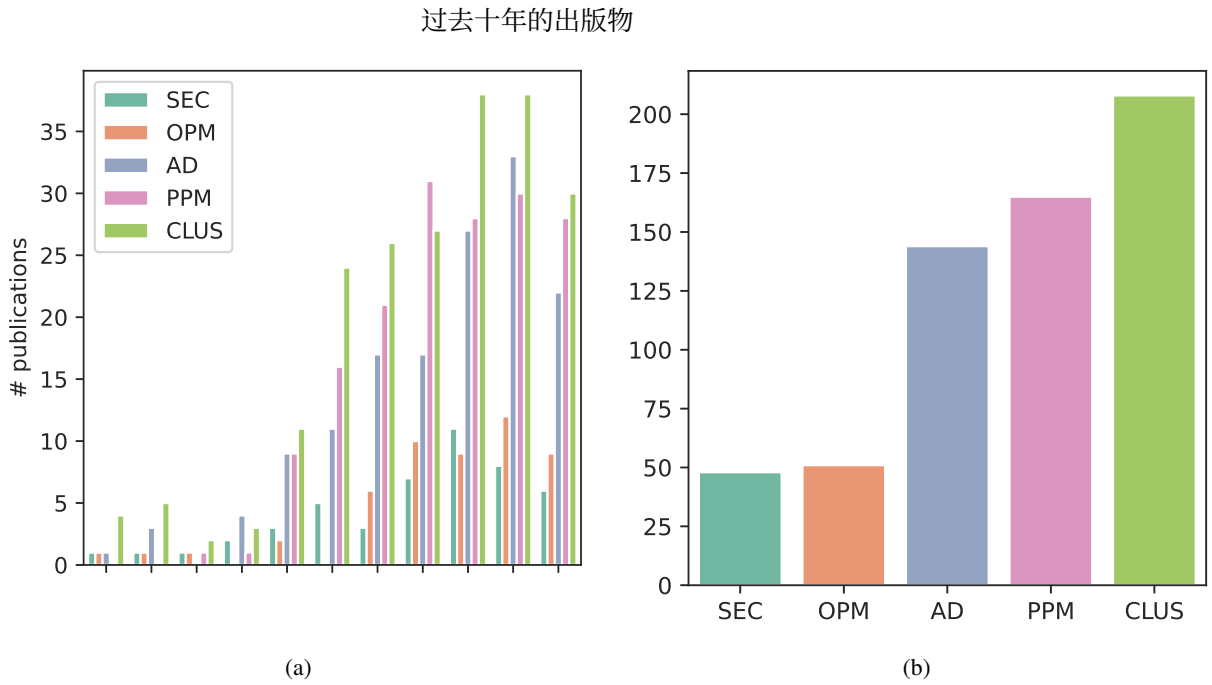
4 编码方法

文献综述为我们提出编码方法分类体系提供了指导，相关内容将在后续章节中讨论。据我们所知，这是流程挖掘文献中首篇针对流程挖掘任务的编码方法系统综述。已有针对特定算法类别的综述和基准测试，例如图嵌入 [24] 或文本嵌入 [25], ，但这些工作不属于过程挖掘应用的范畴。在流程挖掘中，不同任务需要采用不同的编码方法；本文将综述重点放在轨迹聚类、预测性监控和异常检测任务上。

4.1 系统综述

我们通过分析文献中采用的方法进行了系统综述。所使用的在线数据库有 ACM数字图书馆[2], IEEE Xplore [3], 和 Scopus[4]。为了缩小搜索范围，我们未将 Google Scholar 纳入其中，因为它通常收录与其他数据库相同的论文，此外还会收录来自未知数据库的论文。此外，我们仅搜索了本次综述开展日期之前最近10年的研究，即2012年至2022年。我们定义了一个基础查询，并根据每项流程挖掘任务进行了部分修改：*“流程挖掘” AND (“编码” OR “编码”) AND* $< task >$，其中关键词任务 可以是*“聚类”*、*(“预测性监控” OR “过程监控”)*、*(“异常检测” OR “一致性检查”)*、*(“在线过程挖掘” OR “流过程挖掘”)*或*(“安全” AND “隐私”)*。请注意，我们将“一致性检查”和“异常检测”这两个术语互换使用，因为异常检测可视为一致性检查的一项子任务。

筛选时仅保留会议论文和期刊论文，并删除重复项后，我们检查了每篇检索到的文档摘要，以剔除不相关的论文。最终共获得616篇论文，其中208篇归为聚类（CLUS），165篇归为预测性流程监控（PPM），144篇归为异常检测（AD），51篇归为在线过程挖掘（OPM），48篇归为安全（SEC）。图2a展示了各任务每年的发表数量，图2b展示了各任务的发表总数。所有检索到的研究均可公开获取[5]



(a) (b)

图 2：（a）每年的出版物数量。（b）过去十年的出版物总数 ars.

[2]https://dl.acm.org/ [3] https://ieeexplore.ieee.org/Xplore/home.jsp [4] https://www.scopus.com/search [5] shorturl.at/uwJNW

4.2 分类体系

为引导讨论，我们将综述的方法整理成一个分类体系，如图3所示。首先，图3展示了现有的编码问题类别及其交集。我们可以在控制流层级或数据流层级进行编码 [26], : 前者仅考虑事件数据，而后者则分析案例[26]中的其余数据。近期提出了案例间和案例内这两个术语[27]，以通过分别对每种流类型进行编码，部分涵盖这一差异。具体而言，案例间层级旨在捕捉案例之间的关系，例如对案例类型进行分类，以及衡量案例之间的相似性。这一概念的动机之一，是需要区分例如两个活动序列（前缀）完全相同、但标签（下一活动）不同的情况。对于案例内编码，重点是通过编码单个活动或已完成的轨迹来表示过去的执行情况。尽管前者与现实场景更相关，但在现有基于机器学习的应用中较少采用，因为这一概念是由 [27]近期提出的。另一方面，案例内编码广泛应用于流程挖掘的不同任务中，因此我们将贡献重点放在这一编码层级上。

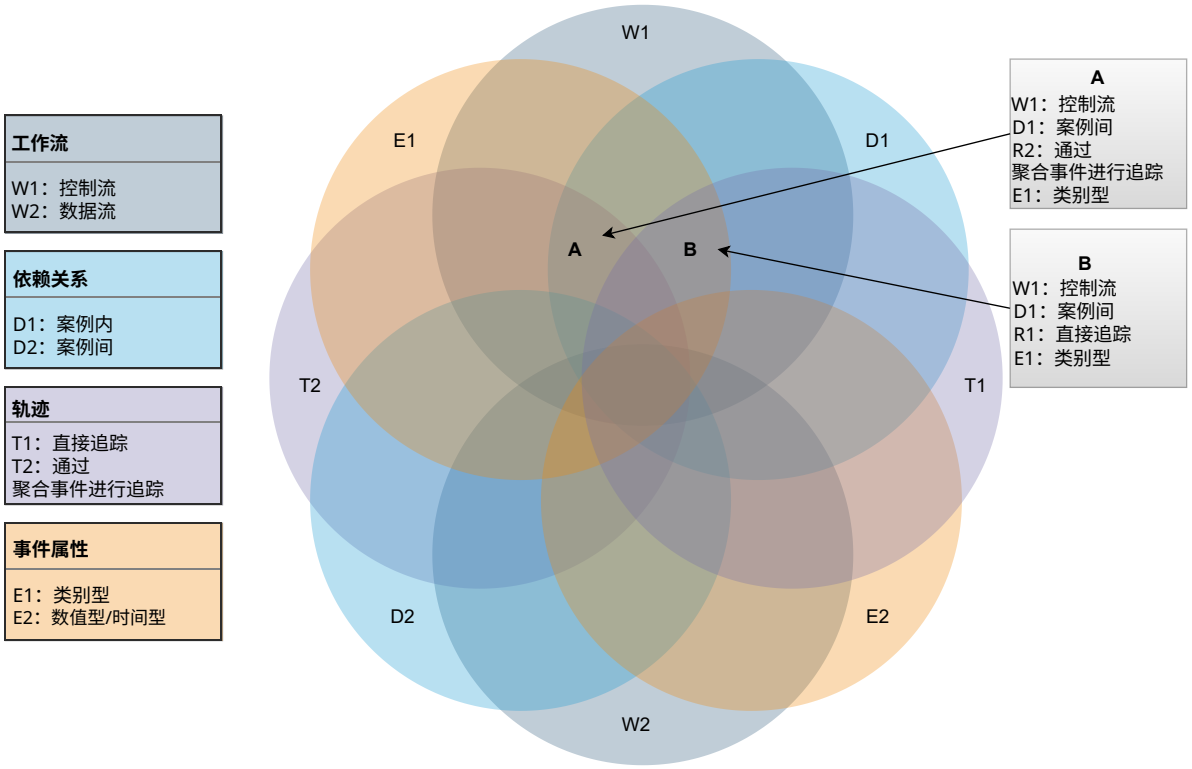


图3：事件日志编码的通用分类体系。编码可以在事件属性层级进行，即对每个属性分别编码；也可以在轨迹层级进行，即聚合编码后的事件属性，以概括整个轨迹。

此外，在这一层级进行数据编码时，可能存在特定的目标。例如，事件属性可能需要单独编码。这是预测性流程监控中的常见设置，其中对每个活动进行编码，以预测下一个活动。另一方面，某些特定应用（例如聚类任务）可能需要对完整轨迹进行编码。在这种场景下，可以由所采用的算法直接对轨迹进行编码，也可以通过聚合各个事件属性来编码。

在对流程挖掘中的编码方法进行系统综述、考察文献中大量可选编码方法并提出编码示意图后，我们得以总结本研究的发现，并将这些编码方法整理为不同的类别。图3所提出的分类体系展示了各种可能场景之间的交集。在图中，我们仅指出两个交集（*A* 和*B*），因为它们在文献中最为常见；不过，我们预计未来研究或许会补充其他交集。某些交集即便在逻辑上并非不可能，也难以实现。例如，同时采用控制流与数据流，或同时涉及案例内与案例间的某种方法，设计起来就很复杂。无论如何，我们综述的全部编码方法仅涵盖两个交集，这一事实表明流程挖掘领域仍有潜力探索新方法。例如，我们综述的编码方法中，没有一种利用案例的时间维度。借助系统综述，我们可以进一步扩展上述

交集，并将发现的编码方法归为图4所示的不同类别。因此，这些类别可以基于或借鉴以下研究领域的技术：流程挖掘、文本挖掘和图嵌入。此外，图4还展示了两个交集之间的差异，即应如何进行轨迹编码：是由给定算法直接编码，还是通过聚合先前编码的事件属性来完成。

考虑到文献中发现的编码方法类别，在以下各小节中，我们将按照各自所属的类别介绍每种编码方法。据我们所知，本研究采用的大多数方法此前从未用于流程挖掘任务。此外，为了鼓励研究人员和从业者更多地考虑这些替代方法，本研究仅纳入具有开源实现的方法。

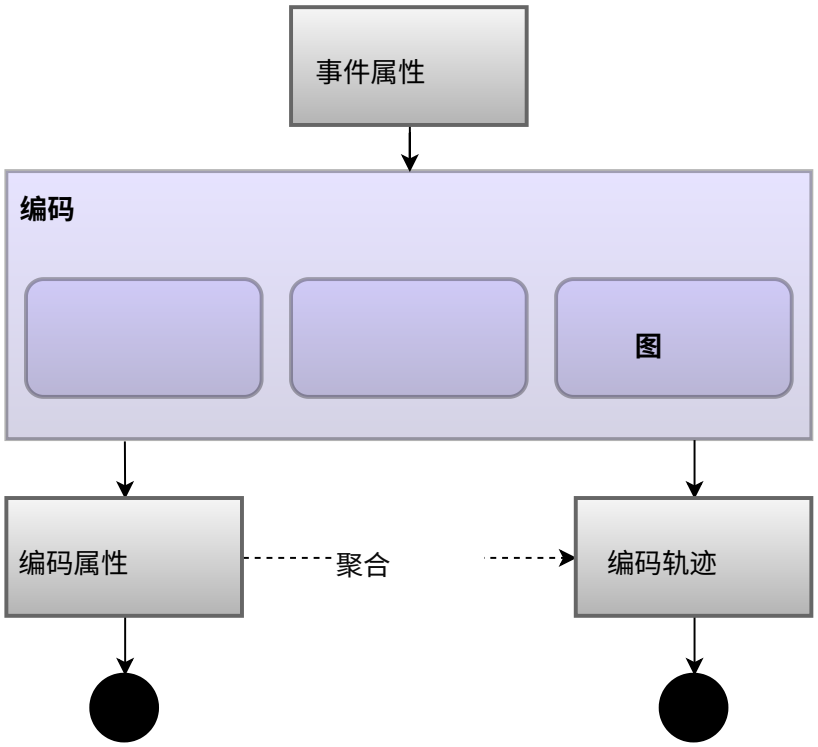


图4：数据编码的不同层次。首先，可以对事件属性分别进行编码。其次，可以通过某种算法直接对轨迹进行编码，也可以通过聚合已编码的事件属性来编码轨迹。

4.3 基于过程模型的编码

给定一个事件日志，我们获取其对应的流程模型，并采用一致性检查技术来衡量该日志与模型的一致程度。事件日志中的每条轨迹都会被评估，所得结果用作该轨迹的编码表征。以下将介绍本综述考察的方法。

轨迹重放：给定一个流程模型，将轨迹在其中重放，以获得用于衡量其符合度的数值 [28]。具体而言，每一步累积的数值包括：正确消耗的词元数量（c）、正确产生的词元数量（p）、执行下一步事件所缺少的词元数量（m），以及最后一个事件执行后未消耗的词元数量（r）。因此，轨迹重放度量定义的最终度量为 $fitness = \frac{1}{2}(1 - \frac{m}{c}) + \frac{1}{2}(1 - \frac{r}{p})$。所有生成的数值 $\langle c, p, m, r, fitness \rangle$ 均用作给定轨迹的特征向量。

轨迹对齐：对流程模型与某条轨迹进行比较，并将该轨迹与有效的执行序列（即模型允许的序列）关联起来 [29]。对齐是由移动构成的序列，这些移动可以是同步的、依赖模型的或依赖日志的。还需注意，日志与模型之间可能存在多个对齐，而相关技术旨在找到最优对齐。最终的特征向量由对齐成本、访问过的状态数、排队中的状态数、遍历过的弧数以及生成的适应度值组成。

日志骨架：该技术旨在通过捕捉适用于整个日志中各项活动的一组约束，来概括活动轨迹 [30]。例如， R_L^{eq} 捕捉两个活动之间的等价关系；如果这两个活动在每条轨迹中的出现频率都相同，则该等价关系成立。另一方面， C_L^{df} 统计每对活动的直接跟随出现次数。捕捉关系的其他度量示例包括始终在后和从不同时出现；计数指标的示例包括某项活动在整个日志中的出现总次数，以及任意轨迹中某项活动出现次数的最小值和最大值。本文所用的实现采用了六种不同的约束。

位置配置文件：该技术通过矩阵表示事件日志，其中每个位置对应于所有轨迹 [31] 的 $activity \times position$。它可以形式化定义为三元组 $apf = (a, p, f) \in E$，其中 a 表示活动， p表示活动的位置， f 表示给定活动的出现频率，而 E 表示事件全集。

4.4 文本启发式编码

流程挖掘中使用的许多轨迹编码方案都改编自自然语言处理中采用的方法。句子中的词语按序排列，并受依存关系约束；利用这一特点，应用于文本的编码方法能够捕捉这些信息。由于轨迹由活动序列构成，同样的信息似乎也与刻画轨迹有关。具体而言，在本综述中，我们考察以下方法。

n 元语法 [32]：该方法通过长度为 n 个元素的子序列来表示给定的元素序列。因此，对于序列 **s** $= \{s_1, ..., s_i\}$，这些序列的 ***n* 元语法** 表征为 ***n* 元语法** $= \{(s_1, ..., s_n), (s_2, ..., s_{n+1}) ... (s_{i-n}, ..., s_i)\}$。

独热编码 [33]：给定一个包含 n 个不同值的变量，会将该变量转换为一个数组，其中每个唯一值都表示为一个二进制向量：将第 i 个位置设为 1，其余位置设为 0。显然，向量的维度取决于向量空间中唯一值的数量 n，很容易达到很高的维度。

计数向量化器（count2vec） [33]：给定一组类别型文档，此方法会生成一个词元出现次数矩阵，矩阵中的每一行代表一个文档，每一列代表一个词元。向量空间的大小取决于其中的 n唯一值。

哈希向量化器（hash2vec） [33]：其功能与*count2vec*相同。不过，它不会存储词元，而是直接将每个词元映射到出现次数矩阵中的一个列位置。此方法主要适用于大型数据集；与独热编码和*count2vec*不同，后两者的维度与词汇表长度相同，而此方法可灵活地将词元哈希到任意维度。

词频-逆文档频率 [34]：词频（TF）表示特定词元相对于给定文档的出现频率，而逆文档频率（IDF）则衡量该词元在语料库中的常见程度。TF 可以简单地表示为词元出现的次数，而 IDF 的计算如下：$\overline{idf(t,\ D)} = log(\frac{N}{count\ (d \in D:\ t \in d)})$，其中 t 是词元， N 是语料库中 d 的文档数量 D。因此，词频-逆文档频率 是将 TF-IDF$(t,\ d,\ D) = tf(t,\ d) \times idf(t,\ D)$ 相乘得到的。

word2vec [35, 36]：*word2vec* 的主要贡献在于学习词语的分布式表示，并相较于当时的最先进方法降低了计算成本。尽管最初有两种用于学习词向量的模型架构，即连续词袋（连续词袋模型）和连续跳字模型（跳字模型），但*word2vec* 的核心特征是移除了简单神经网络语言模型中的隐藏层。连续词袋模型 根据其周围的 t 词语预测当前词，即根据（w_{t-i}，..... $w_t - 1$，w_{t+1}，$...w_{t+i}$）预测 w_t。另一方面，给定 w_t后，跳字模型 预测其周围的词语（w_{t-i}，..... $w_t - 1$，w_{t+1}，$...w_{t+i}$）。在这两种情况下，参数 i 都表示围绕当前词 w_t的范围。

Doc2vec [37]：该算法是 *word2vec* 的扩展，用于学习文档（句子、段落、文章等）的嵌入。与 *word2vec* 的区别在于，其学习过程通过分布式记忆和分布式词袋模型完成，并在输入中添加了另一个向量（文档标识符）。

GloVe [38]：这是一种无监督学习算法，用于获取词语的向量表示。该模型背后的核心思想是捕捉词语共现概率的比率，以同时捕捉局部和全局依赖关系。这可以表示为 $\overline{F(w_i,\ w_j,\ w_k)} = \frac{P_{ik}}{P_{jk}}$，其中 P_{ik} 和 P_{jk} 分别表示词语 k 出现在词语 i和 j 的上下文中的概率。

4.5 基于图的编码

图嵌入方法的核心思想是将图中的节点表示为低维向量，这些向量应具有足够的代表性，以保持图的原始关系（边）不变。我们可以将这一总体思想形式化地定义如下。图可以描述为 $G=(V,\ E)$，其中 $V=\{v_1,...,v_n\}$ 是顶点（节点）的集合，E 是连接一对顶点 $u,\ v\in V$ 的边集合 $e=(u,\ v)$。给定一个图 G，图嵌入是一个映射函数 $f:\ v_i\rightarrow y_i\in R^d$，使得 $d<<|v|$ 和 f 能够保留其局部邻域的原始结构，并最大限度地减少信息损失。本节介绍用于事件日志编码的图嵌入方法。

DeepWalk [39]：它可以被视为一个两阶段算法。首先，通过随机游走来发现局部结构。这里有两个参数：随机游走的次数 α，以及每次随机游走要访问的顶点数量 t。其次，与 *word2vec* 类似，通过执行跳字模型来学习嵌入。该算法背后的直觉是：如果两个嵌入经常出现在相似的结构上下文中，它们就会彼此接近。

Node2vec [40]：该算法与*DeepWalk*类似，不同之处在于它采用有偏随机游走，以权衡广度优先搜索和深度优先搜索。在实践中，这种平衡能够提供比*DeepWalk*更具信息量的嵌入。

Walklets [41]：*DeepWalk* 和*node2vec* 通过生成多次经过 A_{ij}^k 的随机游走，隐式捕获一定程度的依存关系；而该算法则通过结合分解方法与随机游走来显式捕获依存关系。它通过对顶点上的短随机游走进行子采样，并跳过每次随机游走中的若干步，来保留依存关系。由此生成固定长度的路径，构成由顶点对组成的集合。因此，这些集合被用于学习潜在表示。

role2vec [42]：这是一个利用随机游走近似点互信息矩阵的框架，该矩阵通过将结构特征矩阵与汇总邻接矩阵幂相乘得到。

LaplacianEigenmaps [43]：该算法直观地认为，当权重 W_{ij} 较高时，两个节点的嵌入应彼此接近。给定一个图 G，该算法会计算特征值和特征向量 $Ly=\lambda Dy$，其中 D 是对角权重矩阵 $D_{ii}=\sum_j W_{ji}$，而 W 是权重矩阵。因此，$L=D-W$ 是拉普拉斯矩阵，可用于最小化函数 $\rho(Y)=\frac{1}{2}\sum|Y_i-Y_j|^2W_{ij}=tr(Y^TLY)$。

GraRep [44]：该算法学习加权图中顶点的潜在表示 $W\in\mathbf{R}^{|V|\times d}$。它利用全局结构信息来捕捉长距离连接。总体思路是，首先为每个 k 计算转移概率矩阵 $A=D^{-1}S$，其中 $1<=k<=K$ 是最大转移步数。随后，通过对对数概率矩阵进行奇异值分解，得到每个 k 步的表示。最后，将图中每个顶点的 k 步表示拼接起来，并用作潜在表示。

Hope [45]：该嵌入算法与*GraRep*类似，但它不使用转移概率矩阵，而是采用相似度矩阵 S。因此，通过使用不同的相似度度量，可以得到 S，从而保留高阶依赖关系。

BoostNE [46]：该算法通过非负矩阵分解计算先前嵌入模型产生的残差。它采用了集成学习中梯度提升方法的相同思路，即将多个弱学习器集成起来，从而得到更优的学习器。给定通过图的邻接矩阵得到的连通矩阵后，该算法按以下公式计算 k 个残差矩阵，并将每个残差矩阵作为下一个矩阵的输入：

$$R_i=\begin{cases}X, & \text{if } i=1\\ max(R_{i-1}-U_{i-1}V_{i-1},0), & \text{if } i\geq 2\end{cases} \tag{2}$$

其中，$U_i\in R_+^{n\times d_s}$ 和 $V_i\in R_+^{n\times d_s}$ 分别直观地充当 $i-th$ 层中中心节点和上下文节点的嵌入表示。假设已定义残差矩阵，通过最小化损失函数 $L=\min_{U_i,\ V_i,\ \geq 0}||R_i-U_iV_i||_F^2$，即可得到 $i-th$ 层的嵌入表示，其中 $1<=i<=k$。

Diff2vec [47]：该算法的整体思路是对图中每个节点的扩散图进行子采样，并通过欧拉巡回生成顶点序列。给定一个图 G，在类扩散随机过程中对一个包含 l 个顶点的图 G' 进行子采样。然后，从 G' 出发，通过执行欧拉游走生成顶点序列。在此过程中，首先将 G' 转换为多重图，即将每条边加倍。因此，该算法采用欧拉游走而非随机游走，因为欧拉游走能够更完整地捕捉具有这一特征的图。随后使用生成的顶点序列创建图嵌入。

GLEE [48]: 与大多数期望相似节点的嵌入彼此接近的图嵌入算法不同，该算法使用给定图的拉普拉斯矩阵来寻找具有几何属性的嵌入。这些属性包括点积（角度）、线段（或多边形）的长度（面积或体积）、一组向量的凸包等。因此，给定一个图 G 及其拉普拉斯矩阵 L，该过程会提取 L中对应于最大特征值的特征向量。这些向量将用作节点嵌入。

NetMF [49]: 该方法建立在一项理论分析之上，该分析表明，基于*DeepWalk*的不同图嵌入算法具有等价性。在原论文中，作者指出，使用负采样的方法（如*DeepWalk* 和*node2vec*）会隐式地执行矩阵分解。因此，提出了*NetMF* 框架，以统一现有方法并执行显式分解。

NMF-ADMM [50]: 给定一个邻接矩阵，*NMF-ADMM* 算法通过使用交替方向乘子法求解非负矩阵分解问题来学习嵌入。

GraphWave [51]: 给定一个无向图 $G = (V, E)$、一个邻接矩阵 A （二值或加权）以及一个度矩阵 $D_{ii} = \sum_j A_{ij}$，该方法可以学习每个顶点 $v \in V$的结构嵌入。该学习过程基于谱图小波，以无监督的方式进行。*GraphWave* 由$GraphWave = U\ Diag(g_s(\lambda_1) ..., g_s(\lambda_n))U^T \alpha_v$给出，其中 α_v是顶点 v的独热向量， U 是来自 A的特征向量分解， g_s 则是用于调制特征谱的核函数。

NodeSketch [52]: 该方法以递归方式生成 k 阶节点嵌入。这些嵌入分为低阶（$k = 2$）和高阶（$k > 2$）。在每一步 k，都会生成一个自环增强（SLA）邻接矩阵，以获取嵌入。低阶 SLA 只需将单位矩阵加到原始邻接矩阵 $M' = M + I$ 上即可得到。另一方面，高阶嵌入会先对当前节点的近似 k 阶 SLA 邻接矩阵进行采样，再以加权方式将其与（$k - 1$）阶 SLA 邻接矩阵合并。

4.6 相关工作

流程挖掘中的若干任务，如预测性监控、轨迹聚类和异常检测，都需要对数据进行编码，以便为后续流程中应用的算法提供输入。尽管将事件数据转换到合理的特征空间是一项有意义的工作（即这一转换可能会显著影响算法的性能），但编码方法却很少受到关注。就其他问题领域的相关文献而言，已有综述和基准测试尝试根据不同问题的特征，对编码方法进行标准化，并更深入地研究其表现。例如， [24] 对不同数据集上的若干基于图的嵌入方法进行了综述，并讨论了该领域未来研究面临的主要挑战。无论所处理的任务是什么（例如链接预测、节点分类等），作者都指出，选择合适的算法以及合适的一组参数（主要是维度）并非易事。始终需要权衡多种因素，例如，增加内存使用量以提高精度，或降低维度以缩短计算时间。另一方面， [53] 涵盖了范围广泛的文本信息编码方法。该研究聚焦于基于编码的方法，旨在为神经网络架构提供不同任务所需的输入，同时还为每类任务提供了历史背景介绍。

上述研究通常聚焦于表征学习，即利用神经网络学习高质量的数据表征（编码）。在自然语言领域，*word2vec* [35, 36] 可视为实现这一目标最重要的方法之一，它有两种架构变体：一种采用连续词袋模型算法，另一种采用跳字模型 。从这一角度出发，衍生出了若干方法，例如 [37, 10]。表示原始数据的特征向量也称为嵌入。

近年来，表征学习也已应用于流程挖掘（PM）领域。 [10]提出了*act2vec*、*trace2vec*、*log2vec*和*model2vec*。每种方法都源自文献中已有的编码方法，并利用前一层级的信息来丰富学习过程。具体而言，第一层级是*act2vec*，它扩展了*word2vec* 架构，用于学习活动的表征。随后，*trace2vec* 采用*doc2vec* 的概念，联合学习活动和轨迹的表征。*log2vec* 架构源自与*trace2vec* 相同的思路，将日志表征纳入架构以进行联合学习。最后，对于*model2vec*，作者扩展了图表征学习技术，用于表征从事件日志中发现的流程模型。最终的架构还纳入了此前所有表征，以进行联合学习。

在相关文献中，我们还发现了“手工设计”的方法，这些方法通常依据某些领域知识开发。 [12] 提出使用图卷积网络进行预测性监控。在其方法中，作者首先执行特征工程步骤来处理时间特征，然后将事件日志中的每个活动转换为矩阵 $num_unique_activities \times num_time_features$。 [16] 采用流程挖掘算法对资源进行编码，

用于事件日志。简而言之，所应用的算法能够自动发现资源池，从而通过对类别值进行分组来降低其维度。在 [14], 中，作者提出使用卷积神经网络进行预测性监控。因此，他们将数据转换为类似图像的结构，以便训练神经网络。 [54] 提出了一种特征提取方法，可视为一种编码方法：给定一个 Petri 网，从每个活动中提取七种不同的特征。这些特征旨在捕获该活动相对于其当前案例的局部信息。

尽管近期关于预测性流程监控的研究探索了更多替代方案，但值得注意的是，流程挖掘领域的研究通常只采用种类有限的编码方法。大多数论文使用独热编码等简单技术。此外，其他编码方法通常会与特征工程处理的结果结合使用，以处理数值信息和时间相关信息。近期研究中，不同任务最常用的编码方法包括独热编码、 [2, 22, 55, 21, 19], 统计类别数据的频率 [9], 某种嵌入网络[18, 56, 16, 19], 或手工构造的表示 [16, 12, 14]。因此，我们开展这项工作，旨在通过探索更广泛的编码方法来弥补这一不足。

5 方法论

本节将介绍为评估编码方法而开展的实验分析。我们将详细说明所用的软件和材料，以及用于评估所调查编码方法质量的指标。

5.1 实现概述

我们的实现可分为三个步骤：（i）数据集准备，（ii）编码生成，以及（iii）从多个角度评估编码方法。源代码可在此代码仓库[6]中在线获取。

首先，我们使用 PLG2 工具 [57]生成了合成日志。随后，如表 1 所述，我们使用 Python 中的开源库对生成的日志进行编码，其中包括 Sklearn[7], Karate Club[8], PM4PY[9],NLTK[10], Gensim[11], GloVe[12], 以及 GitHub 上的位置配置文件 实现[13]。我们根据提出的分类体系对各方法进行组织，并提供原始论文和在线实现的相应参考资料。此外，我们将*count2vec*、独热编码、n 元语法和位置配置文件设为基线方法，它们实现了最简单的转换。对于事件级编码，首先在活动级进行处理，然后将结果聚合，以获得轨迹表示（轨迹级编码）。这种聚合会提取每个活动编码后的信息，并将其取平均值，形成一个唯一的特征向量。对于基于图的方法，这种聚合通过两种不同方式实现：基于边或基于节点。

5.2 评估指标

假设使用编码方法将原始问题空间映射到不同的向量空间，我们根据若干标准评估了新空间的质量。此外，每种编码方法在性能、描述能力、计算成本以及超参数空间的复杂度方面都有其特点。因此，要有效发挥作用，编码方法应满足以下标准：

- 表达能力：编码方法相对于原始问题空间改变映射空间复杂度的能力。编码方法应能够映射复杂度各异的事件日志，其中，简单的流程对应直观的表征，复杂的流程则对应复杂的表征。

- 可扩展性：指编码计算成本随事件日志大小变化而增加或减少的特性。编码方法应能够快速映射事件日志，同时不影响流程挖掘流水线的运行时间。

[6]https://github.com/gbrltv/business_流程_编码 [7] https://github.com/scikit-learn/scikit-learn [8] https://github.com/benedekrozemberczki/karateclub [9] https://github.com/pm4py/pm4py-core [10]https://github.com/nltk/nltk [11] https://github.com/RaRe-Technologies/gensim [12] https://github.com/maciejkula/glove-python [13]https://github.com/gbrltv/meta_轨迹_clustering/blob/main/clustering.py#L64

算法	Year	类别	实现
count2vec [33]	-	基线	Scikit-learn
n 元语法 [32]	-	基线	NLTK
位置配置文件 [31]	2017	基线	GitHub
独热编码 [33]	-	基线	Scikit-learn
GraphWave [51]	2018	图	空手道俱乐部
拉普拉斯特征映射 [43]	2001	图	空手道俱乐部
NMF-ADMM [50]	2014	图	空手道俱乐部
DeepWalk [39]	2014	图	空手道俱乐部
GraRep [44]	2015	图	空手道俱乐部
node2vec [40]	2016	图	空手道俱乐部
Walklets [41]	2017	图	空手道俱乐部
role2vec [42]	2018	图	空手道俱乐部
NetMF [49]	2018	图	空手道俱乐部
NodeSketch [52]	2019	图	空手道俱乐部
BoostNE [46]	2019	图	空手道俱乐部
GLEE [48]	2020	图	空手道俱乐部
Hope [45]	2016	图	空手道俱乐部
diff2vec [47]	2018	图	空手道俱乐部
日志骨架 [30]	2018	PM	PM4PY
令牌重放 [23]	2016	PM	PM4PY
对齐 [23]	2016	PM	PM4PY
word2vec（跳字模型） [35]	2013	Text	Gensim
hash2vec [33]	-	Text	Scikit-learn
GloVe [38]	2014	Text	GloVe
doc2vec [37]	2014	Text	Gensim
word2vec（连续词袋模型） [36]	2013	Text	Gensim
词频-逆文档频率 [34]	1958	Text	Scikit-learn

表 1：编码方法及相关详情。

- 相关性能力：指编码方法改善原始问题空间的能力。新的特征向量需要与流程挖掘任务的目标高度相关，即编码后的特征向量应提升流程挖掘任务的性能。

- 领域无关性：指给定编码方法将不同领域的数据映射的能力。非特定领域的编码方法只能用于特定应用。

根据所提出的标准，可以采用不同的策略和指标来评估编码方法。本文采用了以下方法：利用主成分分析（PCA） [58]验证向量空间能否有效压缩；利用分类复杂度指标 [59] 衡量样本（即编码后的轨迹）在各类别中的分布情况；利用 F1 分数 [60] 考察编码方法对准确率的影响；利用时间和空间复杂度评估计算性能。表 2 总结了本文采用的各项指标及其作用。

5.3 实验设计

我们的实验设计依赖带标签的数据，对所比较的编码方法进行真实标注评估，这是对 [5] 的扩展。我们按照流程挖掘研究中的标准实践生成合成事件日志，并在生成的轨迹中注入异常，以构建异常检测流程挖掘任务。随后，我们将轨迹标记为异常或正常，使数据集适用于监督学习。我们还在事件日志中加入异质行为，使数据集更贴近现实。

PLG2 [57] 通过随机生成流程，创建了五种不同的流程模型，这些流程能够捕捉多种行为，例如顺序、并行和迭代控制流。PLG2 的原理基于传统控制流模式的组合 [61]，例如序列、并行分支和同步。为了模拟现实场景，这些模式按照预定规则逐步组合。生成的五种流程模型各自定义了五种不同的基础场景，具体取决于场景中包含的活动和网关。

标准	分析	缩略语	描述
表达能力	主成分分析	PCA	使用 PCA 空间的二维投影可以观察到不同场景中的模式这些场景具有不同的复杂度，从极低、低、一般，到高和极高的表达能力。
	主成分分析维度与原始维度的比率	T4	该度量与比例相关编码特征向量所包含的相关维度数。较大的 T4 值表示需要更多编码特征来描述数据变异性。
可扩展性	编码时间	Time	编码任务期间累计耗时（秒）编码任务
	编码内存	Mem	累计内存用量（兆字节）编码任务期间
相关性能力	类内/类间近邻距离的比值 类内/类间近邻距离	N2	该度量对数据在各类别中的分布方式以及数据中的标注噪声较为敏感。较低的数值表明问题较为简单。
	F1 分数	F1	从.....获得的 F1 分数平均值流程挖掘分类任务，代表某种方法所实现的预测性能编码方法。
领域无关性	算法的通用用法	DA	这是一种二元评估（非特定领域或特定领域），用于评估是否具有领域无关性（针对流程挖掘领域）。

表 2：评估流程挖掘中编码方法的标准、策略和指标列表 g.

创建日志需要对流程模型进行模拟。为此，我们应用了 ProM 插件[14] 来模拟随机佩特里网。我们运行了 1 万个模拟案例，案例到达率约为 30 分钟，并将其他超参数保留为默认值。我们按照 [62], 所述注入异常，方法是扰动常规轨迹，如 [63], 所提议，详见表 3。

异常	描述
skip	跳过了 3 个或更少的必要事件
插入	插入 3 个或更少的随机活动到案例中
返工	由 3 个或更少必要事件组成的序列被执行两次
过早	包含 2 个或更少事件的序列执行得过早，随后在该案例中被跳过
late	包含 2 个或更少事件的序列执行得过晚
all	事件日志所在的场景受上述所有异常影响上述

表 3：用于模拟实际事件日志的异常。

对于每种场景，我们通过替换正常轨迹注入不同比例的异常（5%、10%、15% 和 20%）。采用标签和描述作为额外属性，基于五种流程模型、六种异常场景和四种异常比例，共生成了 420 个事件日志。标签表示正常执行或异常执行。描述属性说明异常及其对案例的影响。总体概览见表4。需要注意的是，这些场景是按照复杂度、轨迹长度和日志规模（1k、5k 和 10k 个案例）逐步递增的方式构建的。

[14]http://www.promtools.org/doku.php

log	#gw	轨迹大小	#活动	#案例 (10^3)	#事件 (10^3)	#变量 (10^3)
场景 1	8	9-13	22	1	75 ± 1	1 ± 0
				5	378 ± 5	2 ± 0
				10	755 ± 9	2 ± 1
场景 2	12	26-30	41	1	186 ± 1	1 ± 0
				5	929 ± 5	5 ± 0
				10	1857 ± 10	10 ± 0
场景 3	22	42-50	64	1	308 ± 1	1 ± 0
				5	1538 ± 5	5 ± 0
				10	3077 ± 10	10 ± 0
场景 4	30	3-30	83	1	89 ± 3	0 ± 0
				5	439 ± 6	2 ± 0
				10	879 ± 12	4 ± 0
场景 5	34	4-37	103	1	133 ± 3	1 ± 0
				5	659 ± 7	3 ± 0
				10	1318 ± 13	6 ± 0

表4：五个流程模型概览。对于每个场景，我们通过组合三种不同的基数、注入七种不同的异常，并采用四种不同的注入率，生成了事件日志。因此，每个场景生成了84个事件日志，总计420个事件日志。网关、活动、事件和变量 分别表示网关、活动、事件和变体的数量。

6 流程挖掘编码的基准测试

本节将报告我们在每一类编码方法（基线、图、文本和流程挖掘）实验中取得的结果。

6.1 表达能力

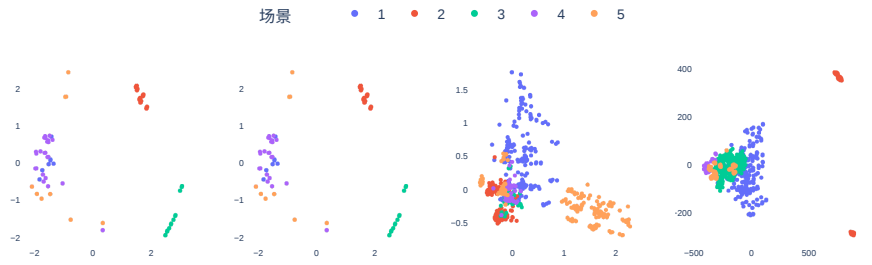
在本研究中，编码方法的表达能力基于主成分分析和T4分析。我们使用每种编码方法的所有事件日志的编码向量计算主成分分析模型，并采用第一主成分和第二主成分的二维子空间投影，以识别映射空间的复杂程度。由于原始问题（即事件日志）相同，编码事件的分布和密度差异可以帮助解释每种编码方法所提供的映射质量。在主成分分析中，每个点代表一个特征向量，每种颜色代表一个场景。该图示突出了生成特征向量的编码能力，即保留轨迹之间和轨迹内部相似性的能力。当相似场景的样本（即编码后的轨迹）彼此聚集时，即在每个二维投影中，相同颜色的点靠近其他相同颜色的点，这一特性便得到了体现。较高的表达能力依赖于同一场景样本形成的、彼此不重叠的簇，并且按场景复杂度排序的簇占据整个投影子空间。较低的表达能力与被遮蔽的样本相关，这些样本呈现混合的稀疏分布或密集的重叠分布。当投影分布部分符合高表达能力和低表达能力的特征时，则可认为具有平均水平的表达能力。当某种编码方法完全符合上述特征（无论是正向还是负向）时，则可获得非常高或非常低的表达能力。

图 5 展示了基线方法（*count2vec*、n 元语法、位置配置文件和独热编码）的 PCA 投影。图 5.a 和图 5.b 分别展示*count2vec* 和独热编码的结果，二者看起来相似，为相同场景分配了大致相同的区域，但仍有很大一部分空间未被覆盖。这是因为这两种方法密切相关，区别在于*count2vec* 会考虑频率。由于大多数轨迹中重复活动的数量不多，使用 PCA 降低维度后，低维空间中的距离几乎相同。另一方面，n 元语法 （图 5.c）和位置配置文件 （图 5.d）的表达能力一般，因为不同场景会重叠在相同区域。

图 6 展示了基于流程挖掘的编码方法的投影。它们分别具有极高和较高的表达能力：对齐 （图 6.a）的表达能力最高，其次是令牌重放 （图 6.b）和日志骨架（图 6.c）。值得注意的是，对齐 投影中的分布与场景的复杂度相对应。

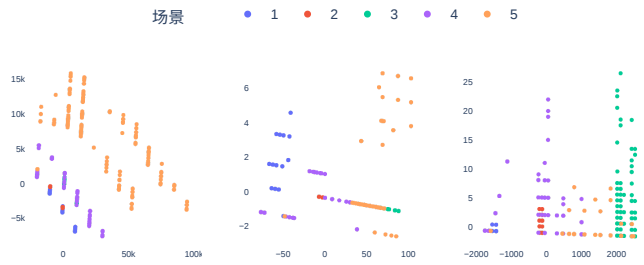
如图7所示，文本编码系列呈现出中等、较低和极低的表达能力。*GloVe*和*hash2vec* 的表达能力为中等，如图7.a和图7.b所示。词频-逆文档频率 （图7.c）、连续词袋模型 （图7.d）和跳字模型 （图7.e）的表达能力较低。*doc2vec*的表达能力最差，为极低水平（图7.f）。

使用图编码系列时，观察到的表达能力水平包括极高、高和中等（图8）。*BoostNE* 和*role2vec* 的表达能力为中等（图8.a和图8.m）。由*DeepWalk*、*diff2vec*、*GLEE*、*GraRep*、*Hope*、拉普拉斯特征映射、*NetMF*、*NMF-ADMM*、*node2vec*、NodeSketch和*Walklets*的编码向量计算得到的主成分分析投影表现出较高的表达能力，分别对应图8.b、图8.c、图8.d、图8.f、图8.g、图8.h、图8.i、图8.j、图8.k、图8.l和图8.n。使用以下方法观察到了极高的表达能力



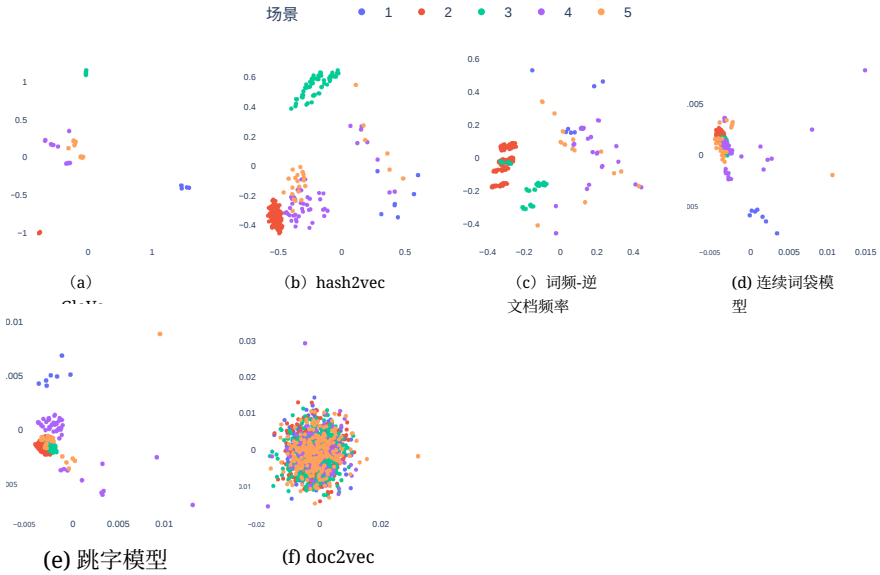
(a) count2vec (b) 独热编码 (c) n 元语法 (d) 位置配置文件

图 5：基于五种不同场景（1、2、3、4 和 5）中事件日志的基线方法族编码方法的特征向量经主成分分析变换得到的二维投影。每个投影对应一种编码方法，每个点代表一条编码轨迹，每种颜色代表一个场景。



(a) 对齐 （b）令牌重放 （c）日志骨架

图6：基于五个不同场景（1、2、3、4 和 5）中事件日志的流程挖掘编码家族方法的特征向量进行 PCA变换得到的二维投影。每个投影对应一种编码方法，每个点对应一个编码事件日志，每种颜色代表一个场景。

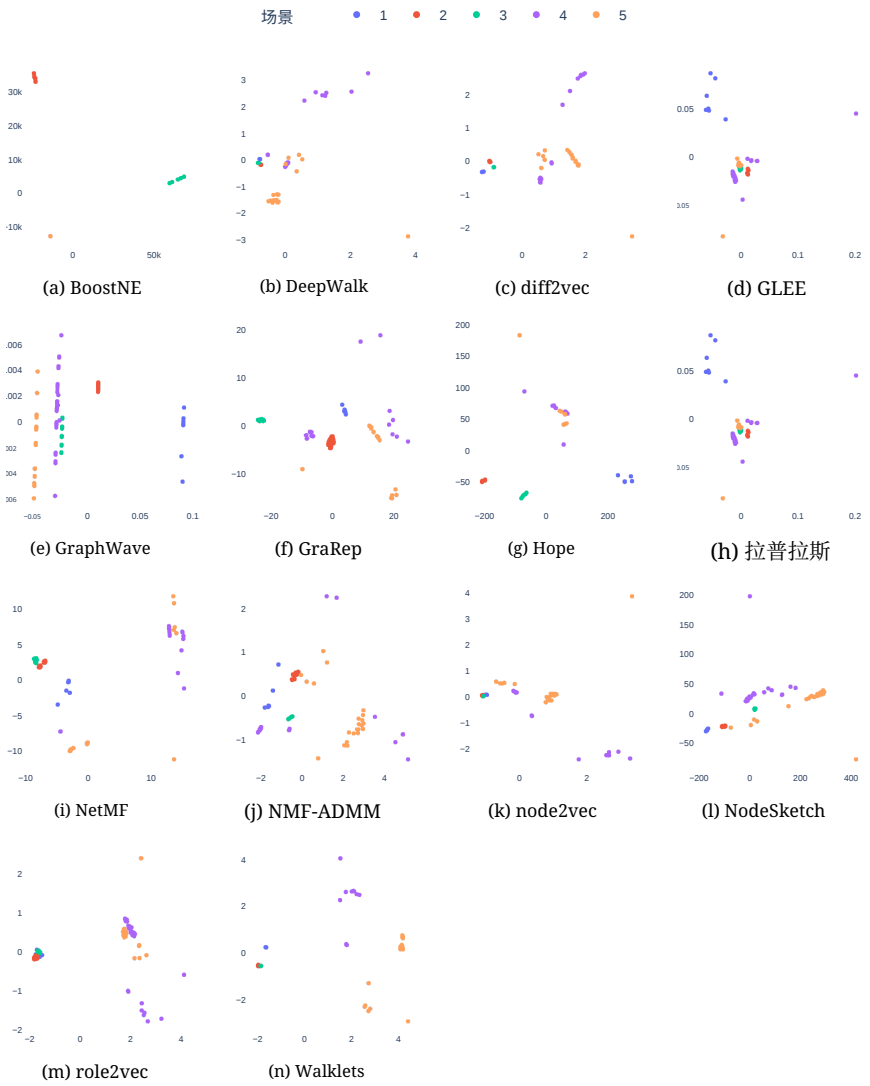


（a） （b）hash2vec （c）词频-逆文档频率 (d) 连续词袋模型

(e) 跳字模型 (f) doc2vec

图7：该图展示了来自五种不同场景（1、2、3、4 和 5）的事件日志中，文本编码系列方法的特征向量经 PCA 变换后得到的二维投影。每个投影对应一种编码方法，每个点代表一个编码后的事件日志，每种颜色代表一个场景。

GraphWave （图 8.e）中，可以观察到按场景复杂度排列的有序梯度，所有不同的事件日志都分散在二维投影中。T4 度量的测量结果证实了这一结果。



(a) BoostNE (b) DeepWalk (c) diff2vec (d) GLEE

(e) GraphWave (f) GraRep (g) Hope (h) 拉普拉斯

(i) NetMF (j) NMF-ADMM (k) node2vec (l) NodeSketch

(m) role2vec (n) Walklets

图8：基于五种不同场景（1、2、3、4 和 5）下事件日志的图编码系列方法特征向量的 PCA 变换所得的二维投影。每个投影对应一种编码方法，每个点代表一个编码事件日志，每种颜色代表一个场景。

T4 粗略衡量编码向量用于映射事件日志 [5] 的相关维度比例，取值范围为 0 到 1。相关性根据 PCA准则确定，该准则旨在通过特征 [59]的不相关线性函数，尽可能描述数据中的大部分变异性。T4 值越高，表示输入变量之间的关系越复杂，也意味着需要更多原始特征来描述数据变异性。如图9所示，基于图的方法的表达能力最佳，其次是基线方法族。就 T4 值而言，*doc2vec* 的表达能力最低，为 0.92。

6.2 可扩展性

我们通过考察整个编码过程中累积消耗的时间（秒）和内存（KB）来讨论可扩展性。对于数据量巨大的实际事件日志，成本高昂的方法并不适用，因此时间和内存成本会直接影响编码方法的选择。在我们的实验中，我们只考虑了

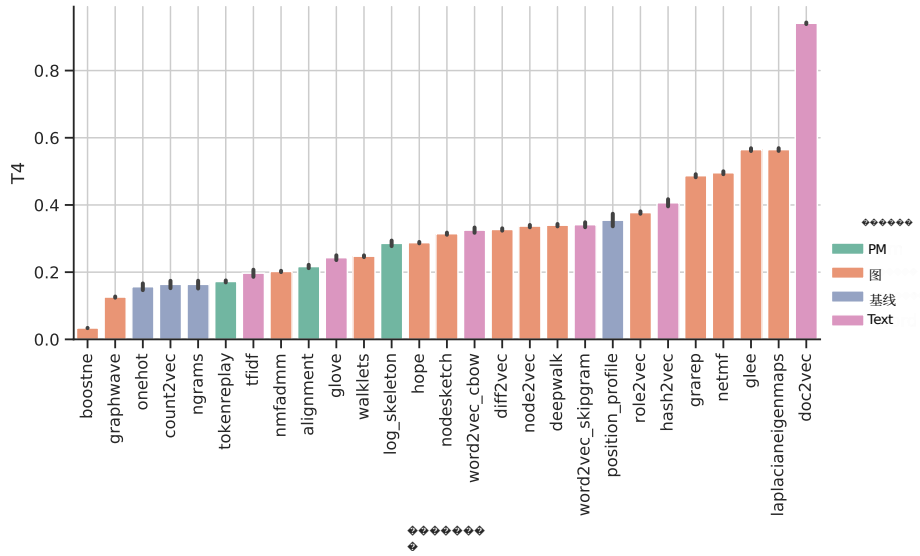


图9：用于表达能力评估的编码方法所获得的 T4 值。较低的 T4 值与良好的表达能力相关，而较高的 T4 值表示表达能力较差。

编码任务期间消耗的时间和内存。通过比较三种不同大小的事件日志（1k、5k 和 10k），我们可以观察使用不同方法对同一组问题进行编码时，成本会受到怎样的影响。图 10、11、12 和 13 用于展示时间和内存方面的可扩展性；对于每种编码方法，纵轴范围均根据所有实验中观测到的较高值设定（左侧为时间，右侧为内存）。

基线方法族的分析如图 10 所示。在内存成本方面，基线编码家族表明，n 元语法是成本最高的方法，空间复杂度很高。另一方面，位置配置文件在时间复杂度方面表现最差。独热编码表明，随着问题规模扩大，内存成本的增长幅度超过了时间成本。在可扩展性方面，*count2vec* 在基线方法族中展现出最佳的可扩展性。

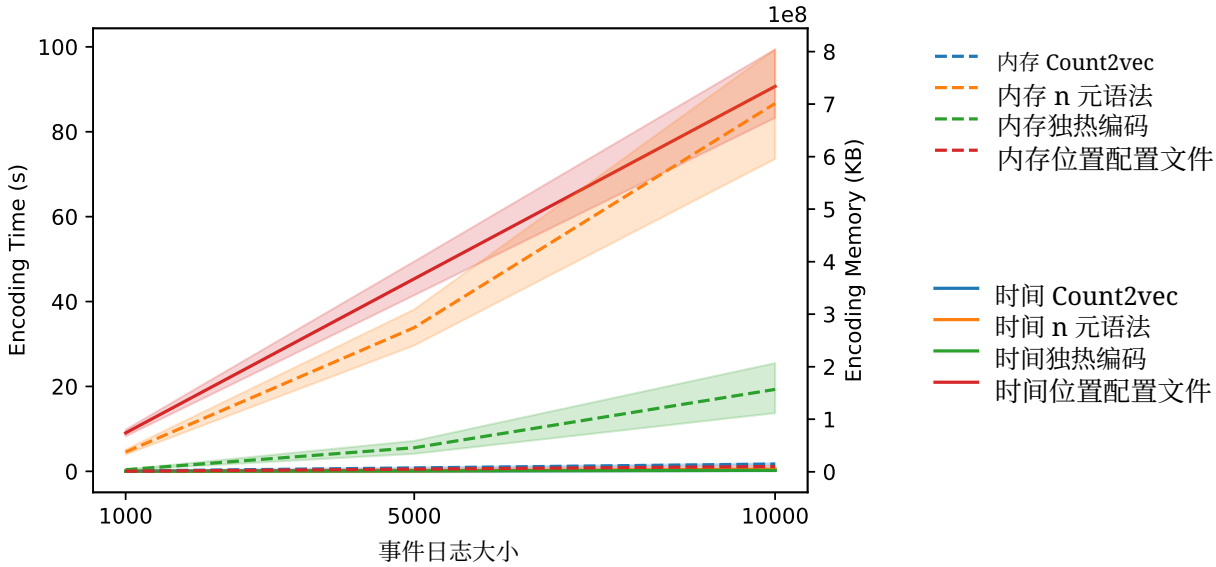


图 10：使用基线编码家族时，不同事件日志规模（1k、5k 和 10k）下的时间和内存开销。

流程挖掘编码系列的内存成本较高、时间成本中等，但两项指标的可扩展性都较好；如图 11 所示，随着数据集增大，性能略有提升。对齐方法的内存成本最低，令牌重放在时间方面表现出良好的平衡，而日志骨架在内存和时间两方面的成本都最高。

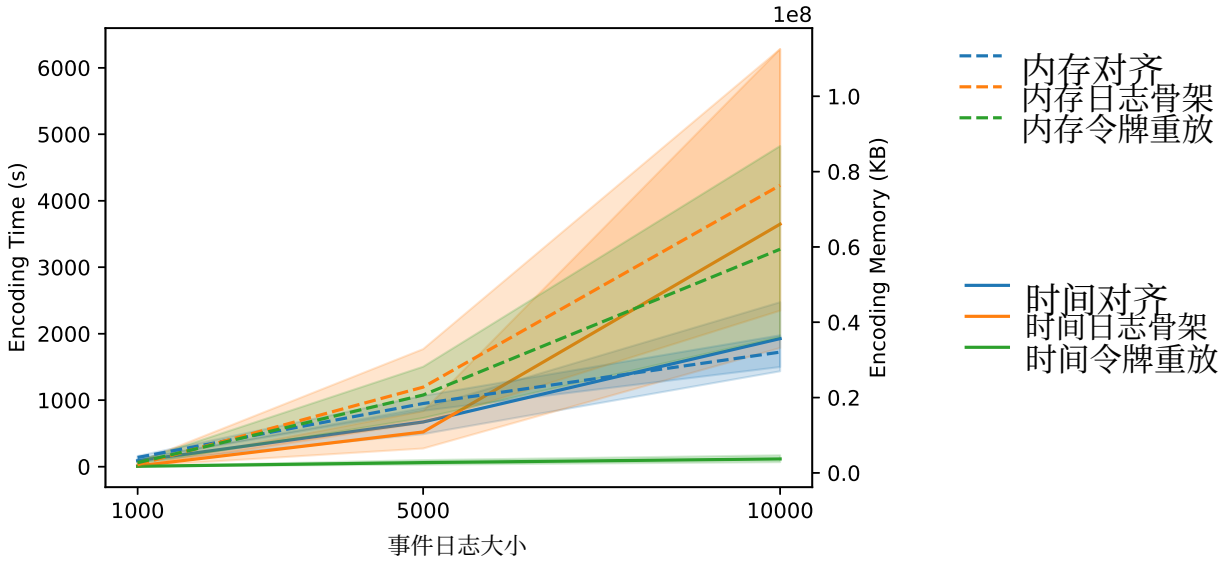


图 11：时间和 使用流程挖掘时，不同事件日志规模（1k、5k 和 10k）下的内存开销编码家族。

文本编码系列中的方法速度最快，且内存消耗较低。然而，如图12所示，大多数方法（*GloVe*、*hash2vec*、连续词袋模型、跳字模型 和 *doc2vec*）都存在时间可扩展性问题。*doc2vec*的可扩展性最差。一个显著的例外是词频-逆文档频率，即使问题规模增大，其内存使用量仍然较低。需要注意的是，可扩展性是根据能否避免受到数据增长的影响来评估的；文本编码系列的平均时间和内存使用量是最低的。

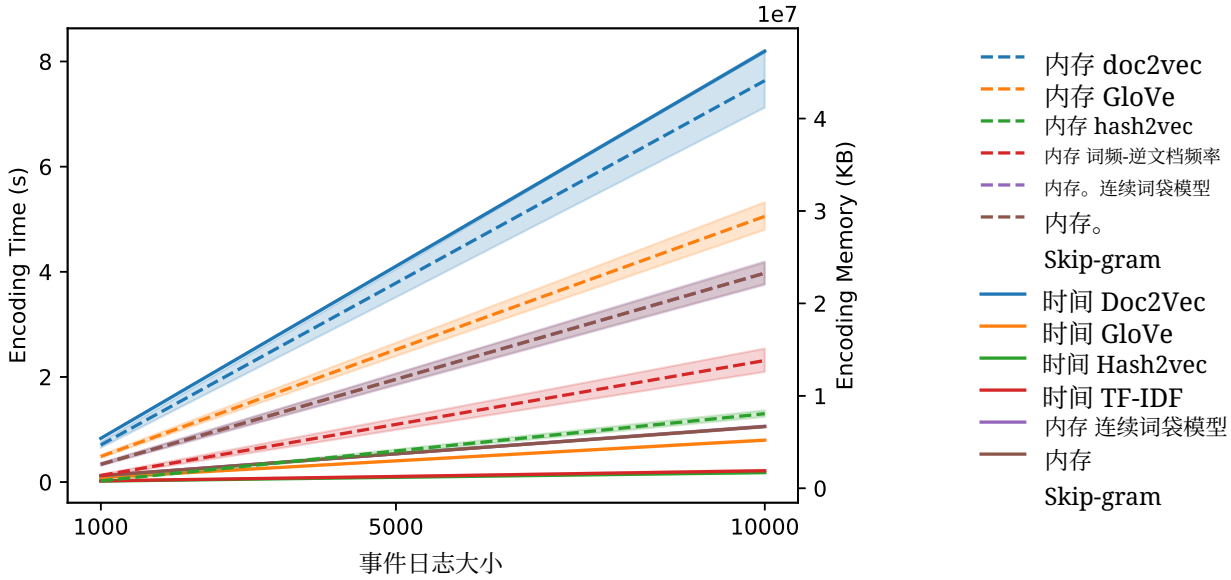


图12：时间和 使用文本编码系列时，不同事件日志规模（1k、5k 和 10k）下的时间和内存开销 .

如图13所示，图编码系列的内存使用情况各不相同，平均时间成本居中。*GLEE*、*Hope* 和*NetMF* 是该系列中速度最快的方法。这些方法的可扩展性一般。*NodeSketch*表现出最佳的可扩展性。图编码系列中内存成本最高的是 *GraRep* ，其成本与令牌重放 （过程挖掘系列）相当，但时间可扩展性一般。最慢的方法是*node2vec*；使用最小的事件日志时，其耗时是该系列其他方法平均耗时的4倍。处理较大的事件日志时，耗时差距达到其他方法的7倍。

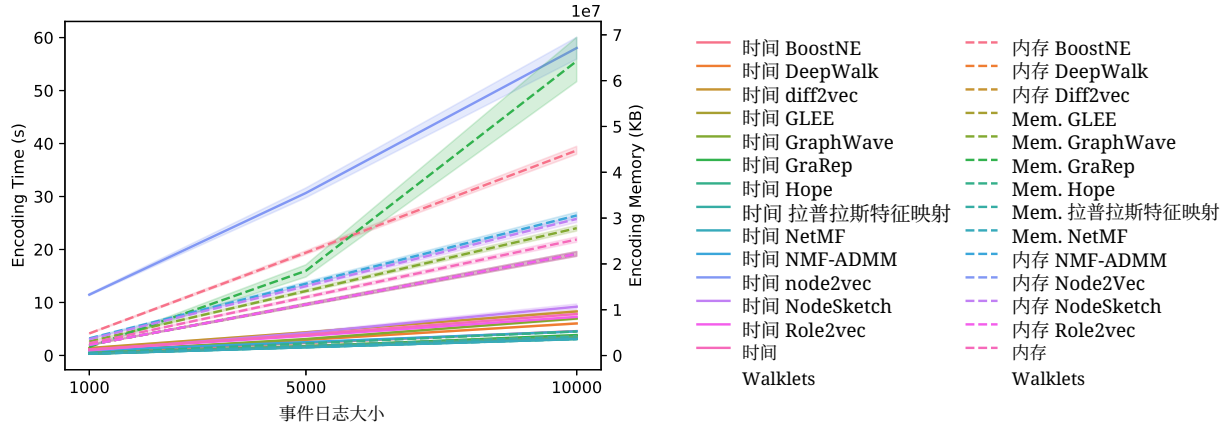


图13：使用图编码系列时，不同事件日志规模（1k、5k 和 10k）下的时间和内存开销。

我们将时间和内存分析的可扩展性及消耗结果整理为热图（图14）。从图中可以看出，尽管 *count2vec* 通常内存消耗较低、耗时较少，但这并不意味着它具有良好的可扩展性；也就是说，随着事件日志规模增加，某些方法的时间和内存开销会呈二次复杂度增长。相比之下，日志骨架、令牌重放和 *NodeSketch* 的可扩展性很强，但内存和时间开销较高。

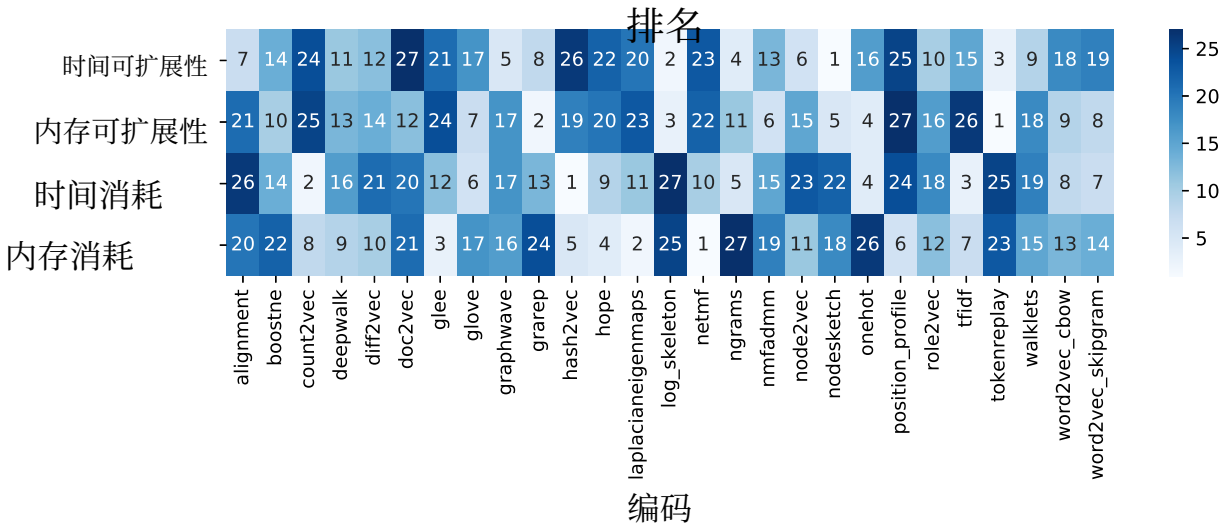


图14：时间可扩展性、内存可扩展性、时间消耗和内存消耗的排名。表现较好的方法位于排名首位，以白色标示。开销最大且可扩展性最差的方法位于排名末位，以深色标示。

6.3 相关性能力

相关性是一项重要的分析视角，因为它反映了编码方法在性能方面与所执行任务的相关程度。换言之，它体现了该编码对最终性能的积极贡献。在

我们的基准测试中，我们基于异常轨迹检测任务评估了相关性。具体而言，我们评估了检测异常行为所获得的 F1 分数，以及映射空间中的 N2。N2 是一个比率，用于计算样例与其在特定类别内最近邻（类内）的距离，以及样例与不同类别中最近邻（类间）的距离。当不同类别样例之间的距离大于同一类别样例之间的距离时，N2 值较低，其取值范围为 0 到 1。因此，N2 值较低的映射空间，意味着其表征更有利于区分类别并支持监督学习。本文按照 [59] 所述的方法计算 N2。

图 15 展示了所有编码方法在编码空间中得到的 N2 值。图中以不同颜色表示各个类别，并按 N2 值从优到劣排序。位置配置文件取得了最佳 N2 值，平均值为 0.48，也是唯一在该指标上低于 0.50 的方法。排名前 5 的 N2 值来自图和文本类别。流程挖掘编码系列，尤其是对齐 和令牌重放，取得了较高的 N2 值，超过 0.85。相比之下，日志骨架 的 N2 值低于 0.70，在编码方法中排名倒数 10，但在异常检测任务中取得了最佳 F1 分数。

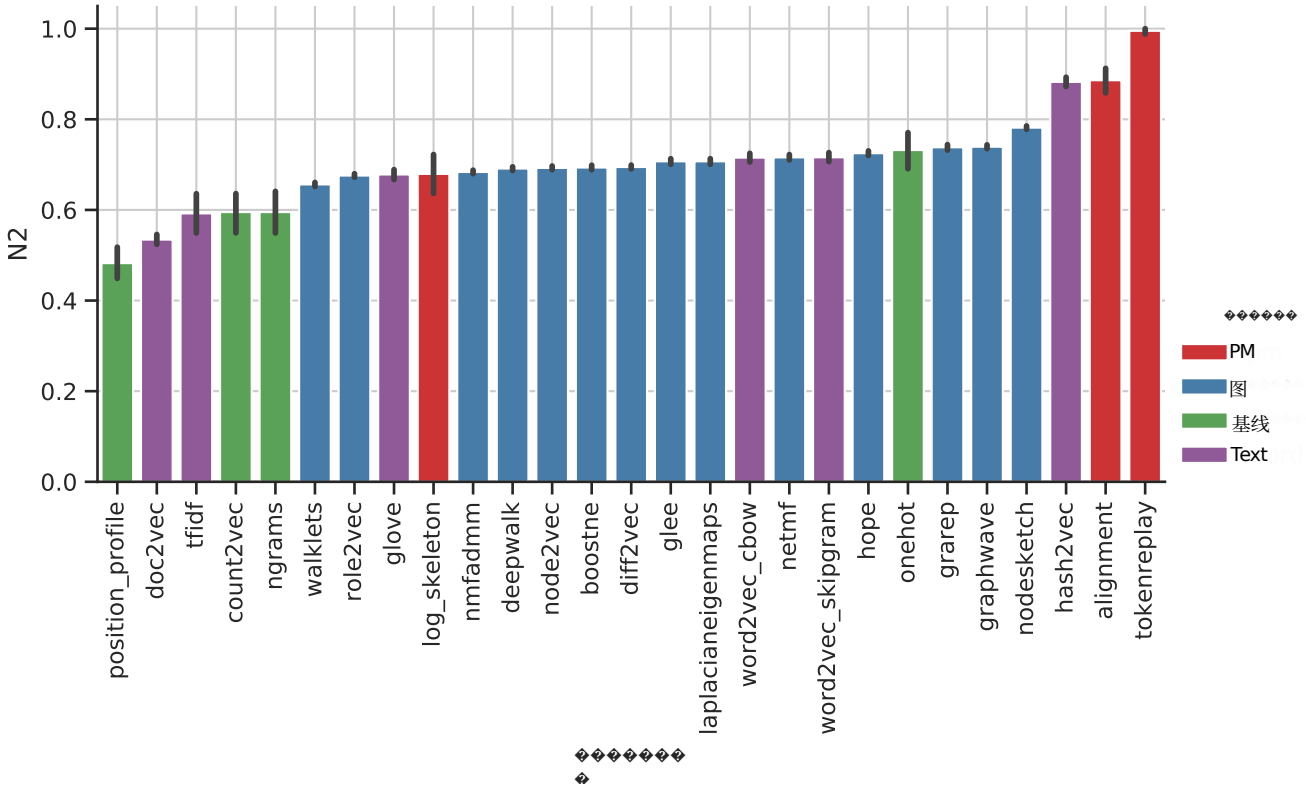


图 15：同一组任务所提供的类内/类间最近邻距离比（N2），该组任务考虑了 27 种不同的编码方法。柱形按得分从最佳（左侧）到最差（右侧）排序。每个柱形根据编码家族着色。

在评估 F1 分数时，日志骨架的性能最佳（均值为 0.942），其次是位置配置文件（均值为 0.935），以及图编码系列中的所有编码（均高于 0.915）。文本编码系列的结果均高于 0.903，只有*doc2vec* 例外，其平均 F1 分数约为 0.845。令人意外的是，流程挖掘中最传统的编码独热编码表现最差，F1 分数低于 0.840。图 16 按性能从高到低对所得 F1 分数进行了排序。

为了对预测性能进行公平且有统计依据的比较，我们采用了 Friedman 统计检验和 Nemenyi 事后检验。两种检验均用于验证各编码方法在 F1 分数性能上的差异是否具有统计显著性。统计比较结果可通过临界差异图呈现，如图 17 所示。临界差异检验可用于检查各分段器之间是否存在统计差异；每个图中各方法的平均排名标注在横轴上，排名最佳者位于右侧。实线连接的是性能差异不显著的编码方法。因此，图 17 表明，日志骨架、位置配置文件、

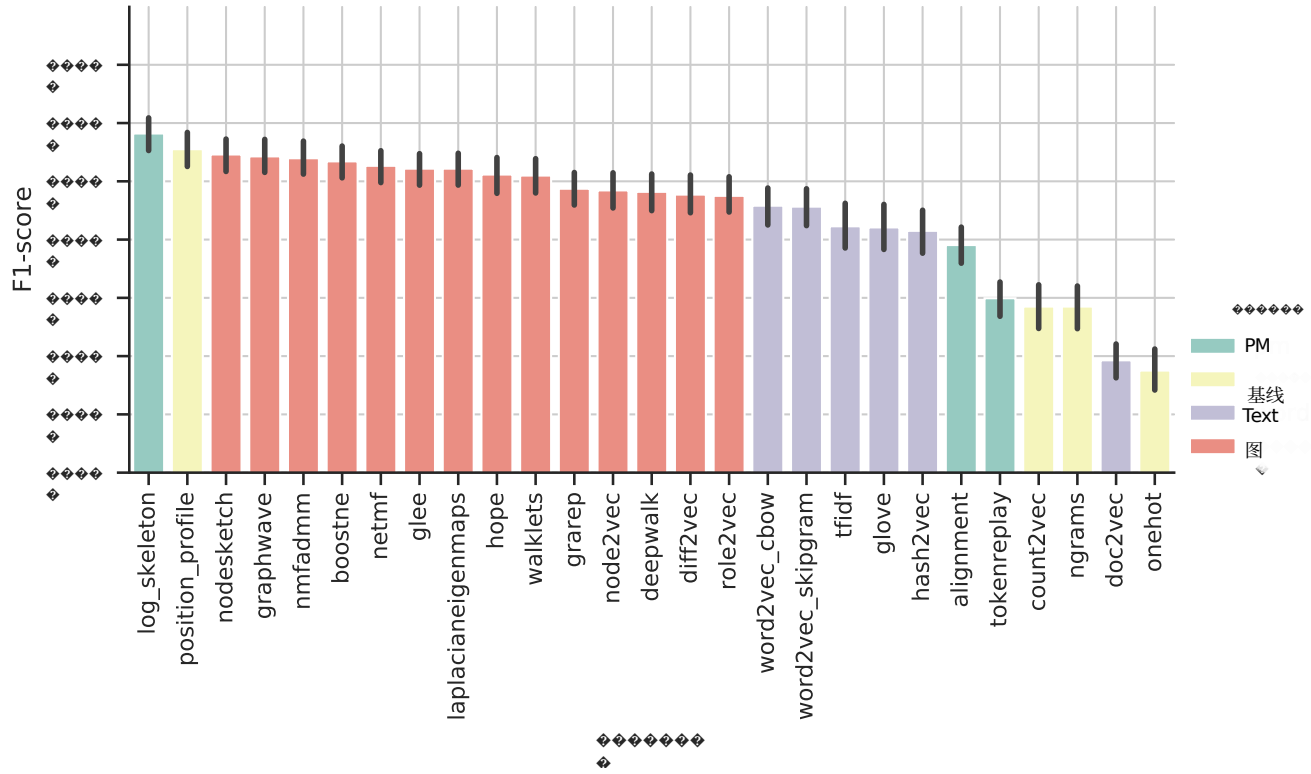


图 16：在 27 种不同编码任务中，使用随机森林算法执行多项分类任务所得的 F1 分数。各柱按分数从高（左）到低（右）排列。每个柱的颜色对应一种编码家族。

NodeSketch、*NMF-ADMM*、*GraphWave*、*BoostNE*、*NetMF*、*Walklets*、*Hope*、*GLEE*、拉普拉斯特征映射、*node2vec*之间不存在统计差异，它们的预测性能均较高。另一方面，对于提供较低预测性能模型的编码方法，即独热编码、*doc2vec*、*n* 元语法、*count2vec*、令牌重放、对齐、*hash2vec*、*GloVe*、跳字模型、连续词袋模型 和词频-逆文档频率，它们之间也不存在统计差异。

6.4 领域无关性

最后一项比较标准是领域无关性。我们认为，这项标准与表达能力、可扩展性和相关性同等重要，有助于理解所得结果。因为一种适用于多种应用的方法，对于构建自适应数据科学流水线至关重要。

基线方法族中的方法传统上用于不同的流程挖掘任务。它们通过将轨迹映射为特征向量，实现直接的转换。这些方法的用途并不局限于流程挖掘流水线，实际上，它们最初是在其他数据挖掘领域中创建的。基于这些原因，我们认为基线方法族不受领域限制。同样，文本和图类别也被广泛用于各种用途。后两个类别是领域无关性的良好范例，既可用于简单的编码场景，也可用于复杂且高度结构化的表示。

根据我们的标准，基于流程挖掘的解决方案并非领域无关。这类编码方法旨在表征事件日志，且仅用于这一目的。这不应被视为一种局限，而应看作专用方法的一项特征：这类方法展现出较强的相关性能力，即使需要付出较高的计算成本，例如，日志骨架提供了编码空间，用于推导出 F1 分数较高的模型。

图18展示了领域无关性概览，以及编码方法性能和资源消耗的一些影响。最值得注意的观察结果是，非领域无关方法（令牌重放、对齐和日志骨架）具有较高的 N2 值，以及较高的空间和时间复杂度成本。在领域无关方法中，最佳的 N2 和 F1 性能是在平均水平的空间和时间复杂度下获得的，例如使用 *GraphWave* 或*BoostNE*时。

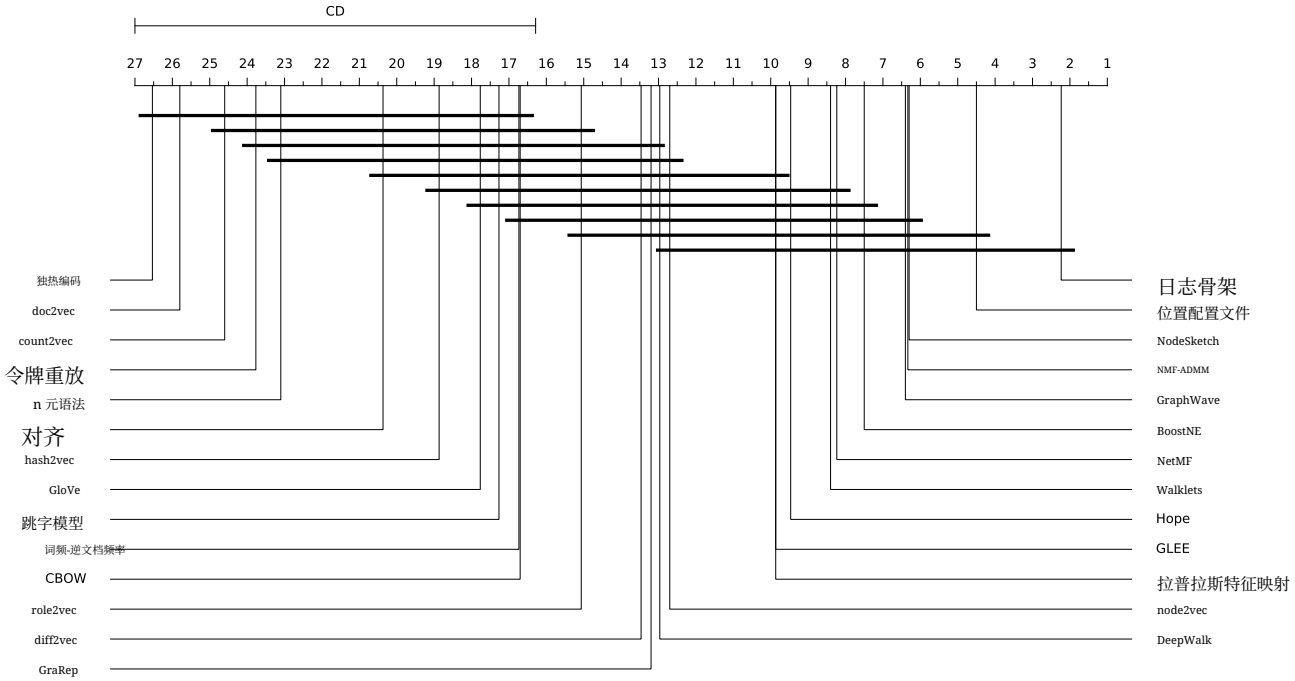


图 17：使用所有编码方法在 10 个场景（从五个不同场景中各取 1k、5k 和 10k 条轨迹）上执行分类任务时得到的**F1 分数**（预测性能）准确率的 Nemenyi 事后检验（ $\alpha = 0.05$ 的显著性和 10.71 的临界距离）。统计上相似的方法由实线连接。

7 问题、关注事项与未来方向

针对流程挖掘的编码方法研究比较仍处于初步阶段。在对第 4 节中的论文进行分类后，我们观察到，所提出的解决方案并未探讨每种方法的优缺点。因此，我们提出了一项研究，涵盖来自不同系列的大量方法。

通过考察不同编码类别和数百个具有各种复杂度的事件日志中，编码方法在表达能力、可扩展性、相关性能力以及领域无关性 方面的优缺点，我们得以深入了解相关问题，并提出一些假设，为未来方向和可能的新型流程挖掘编码方法提供参考。我们认为，本研究采用的评估标准是需要编码事件日志的任何流程挖掘任务要取得重大成果时最应考虑的方面。
此外，所提出的指标也可作为用户需求，用于决定特定问题应采用哪种编码方法。编码方法的表达能力 可以衡量根据事件日志的复杂度，其表征是否直观易懂。可扩展性 也是一项重要考量，因为实际事件日志包含大量数据，可能导致较高的计算成本，包括耗时和内存使用量。了解相关性能力 ，即编码方法的特性与所执行任务的性能之间的关系，也是一项相关分析，可帮助从业者估算分析特定事件日志的复杂度。最后，领域无关性 是本研究提出的一项新颖且重要的议题，旨在探讨编码方法能否适用于不同的问题领域。

针对流程挖掘中编码方法的其他问题，在线流程挖掘可能会给现有编码方法带来新的挑战。正如 [64], 所强调的，准确率和内存消耗等指标应推动相关方法的开发，使其契合在线流程挖掘的目标，例如此类解决方案中采用的编码方法。 [65], 也强调了在线流程挖掘任务所带来的局限性，并指出需要在应对概念漂移时进行适应，同时关注活动间隔时间的影响。STARDUST [66] 是在线流程挖掘的一个实例，尤其适用于轨迹流。作者讨论了如何处理未记录最终活动的轨迹。 因此，亟需研究能够支持在线流程挖掘任务的编码方法，以应对内存消耗降低、映射部分轨迹等新挑战。在我们的实验中，

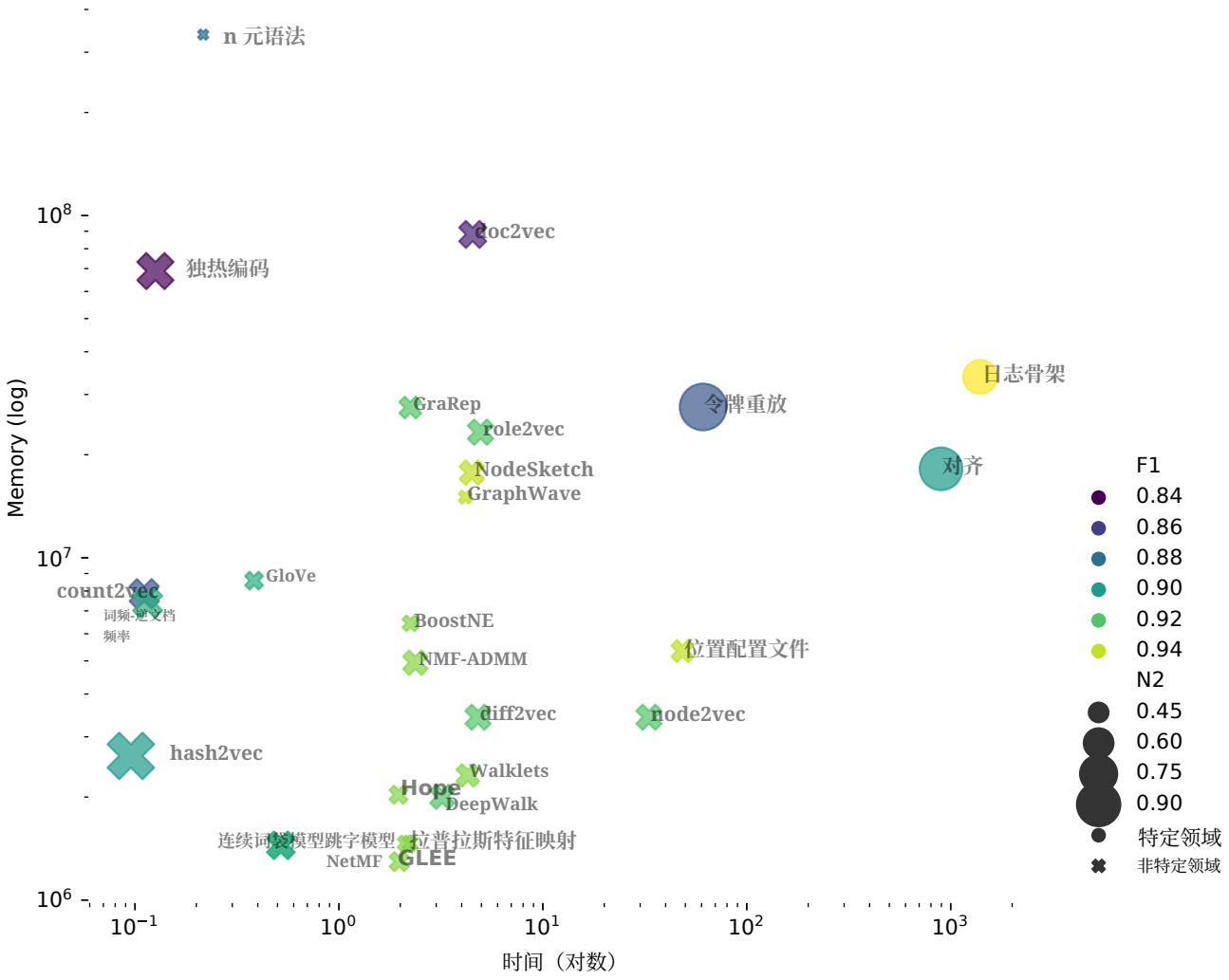


图18：时间、内存和预测性能（F1）的平均值，以对数刻度按时间和内存投影。标记大小表示 N2，其颜色渐变表示 F1 方面的性能。标记符号表示领域无关性评估。

我们使用可扩展性 指标来支持对这一主题的见解和讨论。编码类别是衡量可扩展性的一个关键指标。

目前，预测性流程监控问题也面临着流程挖掘缺乏专用编码方法这一问题。表征学习或特征学习是一种近期由 [10] 引入该领域的学习范式。我们认为，这是改进流程挖掘任务中编码流程的一条有前景的途径。在上述研究中，作者基于*word2vec*提出了他们的新方案。不过，我们认为，流程挖掘需要一种专用方法，或者一种足够通用、能适用于不同问题领域的方法，因为事件数据可能比序列文本更复杂。我们认为，这是因为事件数据的性质通常包含序列规则、关系信息、资源并发、并行活动等。

编码是指为不同目标将数据映射为另一种表示。新的空间可能无法像原始数据那样支持解释和可解释性。此外，正如 [67] 所提到的，从业者需要信任生成的映射空间。特别是在许多预测性监控任务中，所采用的预测模型无法解释其为何给出错误的预测，因此无法理解预测模型出错的原因。 [68] 就这一主题进行了探讨，介绍了事后解释器和不同的编码方法，用于识别重要特征。总体而言，我们关于表达能力 和领域无关性 的实验确认，不同的编码方法能够提供种类繁多的表示，即使它们属于

同一类别。关于可解释人工智能（XAI），一种趋势是，应将能够提供高可解释性水平的编码方法作为关键组件集成到流程挖掘流水线中，而不是作为面向特定应用的后续独立步骤。

编码方法数量众多，而专家资源相对有限，这为编码方法的选择与恰当设置带来了额外挑战。选择方法时通常会采用侧重准确率或时间性能的策略，但测试不同配置的成本以及高昂的调优策略，可能会影响流程挖掘流水线的设计。人们已通过基于元学习的有前景策略来应对这一问题 [69, 70, 71], ，但目前的解决方案需要创建一个包含各种可能解决方案历史记录的元数据库。此外，用于推荐特定算法的标准仍局限于简单的性能函数。因此，自动化机器学习（AutoML） [6, 8] 成为一个重要的研究领域，可应对上述与流程挖掘任务中所用编码方法有关的问题。另一种可针对这类数据探索的学习范式是自监督学习（SSL）。自监督学习背后的总体思路是：给定一个输入，学习一组可能的结果，而不是像传统方法那样预测一个唯一值。例如，*data2vec* 最近由[72], 提出，作者为任意类型的数据提出了一个通用编码框架，不过目前仅考虑了图像、语音和语言领域。直观而言，这种方法或许有助于捕捉事件数据中的相互依存关系。

8 结论

本研究的主要贡献包括：对使用编码方法的流程挖掘任务进行系统综述；提出新的分类体系，将各类方法划分为不同的类别；并开展广泛的实验评估和基准测试，评估相关指标以衡量编码方法的有效性。我们相信，本研究能够帮助研究人员和从业者在流程挖掘的不同应用领域取得重大成果。此外，我们强调了文献中当前面临的挑战和问题，尤其是选择合适算法及其参数的困难。我们还讨论了任意选择算法如何导致不公平的评估和次优解。这是首项聚焦于事件日志预处理详细分析的研究，而非仅关注任务算法本身（即聚类或学习算法）。

我们还强调，需要更好地了解不同事件日志场景和不同流程挖掘任务下各方法的表现。因此，为弥补这一不足，我们借助 PLG2 工具生成具有不同属性的合成流程，模拟了这些场景，并将结果作为基准测试呈现。总计有 27 种编码方法在 420 个包含不同异常的事件日志上接受了评估。我们采用四项不同的评估标准来衡量编码方法对流程挖掘任务的有效性。我们的评估仅限于一项任务——异常检测，但本研究所呈现的分析和见解也可用于其他应用，例如预测性监控和聚类。

最后，我们强调，根据用户的偏好选择合适的算法及其参数并非易事，因为不同的流水线设置会因事件日志的特征而呈现不同的效果。这或许是开发新型自动化解决方案的一个方向，无论是针对完整流水线，还是仅针对预处理步骤。我们还认为，专门处理流程特性的编码方法对于推动最先进水平的发展至关重要。之所以这样说，是因为大多数方法都是从其他领域改造或借鉴而来的。因此，这一研究方向前景广阔，还有许多值得探索的工作。

参考文献

[1] Joerg Evermann、Jana-Rebecca Rehse 和 Peter Fettke。采用深度学习方法预测运行时的流程行为。载于 Marlon Dumas 和 Marcelo Fantinato 编，业务流程管理研讨会（*BPM*），第 281 卷，商业信息处理讲义，第 327–338 页，2016年。[2] Niek Tax、Ilya Verenich、Marcello La Rosa 和 Marlon Dumas。利用 LSTM 神经网络进行预测性业务流程监控。载于 Eric Dubois 和 Klaus Pohl 编，高级信息系统工程会议（*CAiSE*），第 10253 卷，计算机科学讲义，第 477–492 页。施普林格，2017年。[3] B.F.A. Hompes、J.C.A.M. Buijs、W.M.P. van der Aalst、P.M. Dixit 和 J. Buurman。利用轨迹聚类发现偏离案例和流程变体。载于比荷卢人工智能会议（*BNAIC*），2015年11月。[4] Annalisa Appice 和 Donato Malerba。用于流程挖掘中多视图聚类的协同训练策略。*IEEE* 服务计算汇刊，9(6)：832–845，2016年。

[5] Sylvio Barbon Junior、Paolo Ceravolo、Ernesto Damiani 和 Gabriel Marques Tavares。评估流程挖掘中的轨迹编码方法。载于国际研讨会：从数据到模型再返回，第 174–189 页。施普林格，2020年。[6] Randal S. Olson 和 Jason H. Moore。TPOT：用于自动化机器学习的基于树的流水线优化工具。载于 Frank Hutter、Lars Kotthoff 和 Joaquin Vanschoren 编，自动机器学习研讨会（*AutoML*），与 *ICML* 联合举办，第 64 卷，*JMLR* 研讨会与会议论文集，第 66–74 页。JMLR.org，2016年。[7] Hyunjik Kim 和 Yee Whye Teh。扩展自动统计学家：利用高斯过程实现可扩展的结构发现。载于 Amos J. Storkey 和 Fernando Pérez-Cruz 编，人工智能与统计国际会议（*AISTATS*），第 84 卷，机器学习研究论文集，第 575–584 页。PMLR，2018年。[8] Matthias Feurer、Aaron Klein、Katharina Eggensperger、Jost Tobias Springenberg、Manuel Blum 和 Frank Hutter。Auto-sklearn：高效而稳健的自动化机器学习。载于 Frank Hutter、Lars Kotthoff 和 Joaquin Vanschoren 编，自动化机器学习——方法、系统与挑战，施普林格机器学习挑战系列，第 113–134 页。施普林格，2019年。[9] Chiara Di Francescomarino、Marlon Dumas、Fabrizio Maria Maggi 和 Irene Teinemaa。基于聚类的预测性流程监控。*IEEE* 服务计算汇刊，12(6)：896–909，2019年。[10] Pieter De Koninck、Seppe vanden Broucke 和 Jochen De Weerdt。act2vec、trace2vec、log2vec 和 model2vec：业务流程的表征学习。载于 Mathias Weske、Marco Montali、Ingo Weber 和 Jan vom Brocke 编，业务流程管理（*BPM*），第 11080 卷，计算机科学讲义，第 305–321 页。施普林格，2018年。[11] Gabriel Marques Tavares 和 Sylvio Barbon。分析受语言启发的轨迹表示以进行异常检测。载于 Ladjel Bellatreche、Mária Bieliková、Omar Boussaïd、Barbara Catania、Jérôme Darmont、Elena Demidova、Fabien Duchateau、Mark Hall、Tanja Merˇcun、Boris Novikov、Christos Papatheodorou、Thomas Risse、Oscar Romero、Lucile Sautot、Guilaine Talens、Robert Wrembel 和 Maja Žumer 编，*ADBIS*、*TPDL* 与 *EDA 2020* 联合研讨会及博士生论坛，第 296–308 页，尚，2020年。施普林格国际出版。[12] Ishwar Venugopal、Jessica Töllich、Michael Fairbank 和 Ansgar Scherp。用于分析和预测业务流程的深度学习方法比较。载于国际神经网络联合会议（*IJCNN*），第 1–8 页。IEEE，2021年。[13] Nijat Mehdiyev、Lea Mayer、Johannes Lahann 和 Peter Fettke。基于深度学习的流程聚类及其可视化探索：面向中小型企业的工业4.0用例。专家系统，n/a(n/a)：e13139，2022年。[14] Vincenzo Pasquadibisceglie、Annalisa Appice、Giovanna Castellano 和 Donato Malerba。利用卷积神经网络进行预测性流程分析。载于国际流程挖掘会议（*ICPM*），第 129–136 页。IEEE，2019年。[15] Arik Senderovich、Chiara Di Francescomarino 和 Fabrizio Maria Maggi。从知识驱动到数据驱动的预测性流程监控跨案例特征编码。信息系统，84：255–264，2019年。[16] Manuel Camargo、Marlon Dumas 和 Oscar González Rojas。学习准确的业务流程 LSTM 模型。载于 Thomas T. Hildebrandt、Boudewijn F. van Dongen、Maximilian Röglinger 和 Jan Mendling 编，业务流程管理（*BPM*），第 11675 卷，计算机科学讲义，第 286–302 页。施普林格，2019年。[17] Efren Rama-Maneiro、Juan Vidal 和 Manuel Lama。用于预测性业务流程监控的深度学习：综述与基准测试。*IEEE* 服务计算汇刊，第 1–1 页，2021年。[18] Abdulrhman Al-Jebrni、Hongming Cai 和 Lihong Jiang。利用卷积神经网络预测下一个流程事件。载于信息学与计算进展国际会议（*PIC*），第 332–338 页，2018年。[19] Farbod Taymouri、Marcello La Rosa 和 Sarah M. Erfani。用于事件序列后缀和剩余时间预测的深度对抗模型。载于 Carlotta Demeniconi 和 Ian Davidson 编，数据挖掘国际会议（*SDM*），第 522–530 页。SIAM，2021年。[20] Mirko Polato、Alessandro Sperduti、Andrea Burattin 和 Massimiliano de Leoni。业务流程实例的时间与活动序列预测。计算，100(9)：1005–1031，2018年。

[21] Nicola Di Mauro、Annalisa Appice 和 Teresa M. A. Basile。使用 Inception CNN 模型预测业务流程实例中的活动。载于 Mario Alviano、Gianluigi Greco 和 Francesco Scarcello（编），《人工智能进展（AI*IA）》，《计算机科学讲义》第 11946 卷，第 348–361 页。施普林格，2019年。[22] Wolfgang Kratsch、Jonas Manderscheid、Maximilian Röglinger 和 Johannes Seyfried。业务流程监控中的机器学习：用于结果预测的深度学习方法与经典方法比较。《商业与信息系统工程》，63(3):261–276，2021年。[23] Wil M. P. van der Aalst。《流程挖掘——行动中的数据科学》，第二版。施普林格，2016年。[24] Palash Goyal 和 Emilio Ferrara。图嵌入技术、应用与性能：综述。《知识型系统》，151:78–94，2018年。[25] Sebastian Ruder、Ivan Vulic 和 Anders Søgaard。跨语言词嵌入模型综述。《人工智能研究杂志》，65:569–631，2019年。[26] Massimiliano de Leoni 和 Wil M. P. van der Aalst。数据感知流程挖掘：使用对齐发现流程中的决策。载于 Sung Y. Shin 和 José Carlos Maldonado（编），应用计算研讨会（SAC），第 1454–1461 页。ACM，2013年。[27] Arik Senderovich、Chiara Di Francescomarino、Chiara Ghidini、Kerwin Jorbina 和 Fabrizio Maria Maggi。预测性流程监控中的案例内与案例间特征：两个维度的故事。载于 Josep Carmona、Gregor Engels 和 Akhil Kumar（编），业务流程管理（BPM），《计算机科学讲义》第 10445 卷，第 306–323 页。施普林格，2017年。[28] Alessandro Berti 和 Wil MP van der Aalst。重振基于词元的重放：在提升诊断能力的同时加快速度。载于 ATAED@ Petri Nets/ACSD，第 87–103 页，2019年。[29] Josep Carmona、Boudewijn F. van Dongen、Andreas Solti 和 Matthias Weidlich。《符合性检查——关联流程与模型》。施普林格，2018年。[30] H. M. W. Verbeek 和 Renata Medeiros de Carvalho。日志骨架：一种流程发现分类方法。CoRR，abs/1806.08247，2018年。[31] Paolo Ceravolo、Ernesto Damiani、Mohammadsadegh Torabi 和 Sylvio Barbon Junior。迈向新一代流程挖掘日志预处理方法。载于 Josep Carmona、Gregor Engels 和 Akhil Kumar（编），业务流程管理论坛（BPM），《商业信息处理讲义》第 297 卷，第 55–70 页。施普林格，2017年。[32] Andrea Gasparetto、Matteo Marcuzzo、Alessandro Zangari 和 Andrea Albarelli。文本分类算法综述：从文本到预测。《信息》，13(2):83，2022年。[33] Sholom M. Weiss、Nitin Indurkhya 和 Tong Zhang。《预测性文本挖掘基础》，第二版。《计算机科学文本》。施普林格，2015年。[34] H. P. Luhn。文献摘要的自动生成。《IBM研究与开发杂志》，2(2):159–165，1958年。[35] Tomás Mikolov、Kai Chen、Greg Corrado 和 Jeffrey Dean。在向量空间中高效估计词表示。载于 Yoshua Bengio 和 Yann LeCun（编），国际表征学习会议（ICLR），2013年。[36] Tomás Mikolov、Ilya Sutskever、Kai Chen、Gregory S. Corrado 和 Jeffrey Dean。词语和短语的分布式表示及其组合性。载于 Christopher J. C. Burges、Léon Bottou、Zoubin Ghahramani 和 Kilian Q. Weinberger（编），神经信息处理系统（NIPS），第 3111–3119 页，2013年。[37] Quoc Le 和 Tomas Mikolov。句子和文档的分布式表示。载于《第31届国际机器学习会议论文集》第 32 卷，ICML’14，第 II–1188–II–1196 页。JMLR.org，2014年。[38] Jeffrey Pennington、Richard Socher 和 Christopher D. Manning。GloVe：用于词语表示的全局向量。载于自然语言处理经验方法会议（EMNLP），第 1532–1543 页，2014年。[39] Bryan Perozzi、Rami Al-Rfou 和 Steven Skiena。DeepWalk：社交表示的在线学习。载于 Sofus A. Macskassy、Claudia Perlich、Jure Leskovec、Wei Wang 和 Rayid Ghani（编），国际知识发现与数据挖掘会议（SIGKDD），第 701–710 页。ACM，2014年。[40] Aditya Grover 和 Jure Leskovec。Node2vec：面向网络的可扩展特征学习。载于国际知识发现与数据挖掘会议（SIGKDD），KDD ’16，第 855–864 页，美国纽约州纽约市，2016年。计算机协会。

[41] Bryan Perozzi、Vivek Kulkarni、Haochen Chen 和 Steven Skiena。别走，跳过！多尺度网络嵌入的在线学习。载于国际社会网络分析与挖掘会议（*ASONAM*），ASONAM '17，第 258–265 页，美国纽约州纽约市，2017年。计算机协会。[42] Nesreen K. Ahmed、Ryan A. Rossi、John Boaz Lee、Theodore L. Willke、Rong Zhou、Xiangnan Kong 和 Hoda Eldardiry。基于角色的图嵌入。《*IEEE*知识与数据工程汇刊》，34(5):2401–2415，2022年。[43] Mikhail Belkin 和 Partha Niyogi。用于嵌入和聚类的拉普拉斯特征映射及谱技术。载于神经信息处理系统国际会议：自然与合成（*NIPS'01*），NIPS'01，第 585–591 页，美国马萨诸塞州剑桥，2001年。麻省理工学院出版社。[44] Shaosheng Cao、Wei Lu 和 Qiongkai Xu。GraRep：利用全局结构信息学习图表示。载于国际信息与知识管理会议（*CIKM*），CIKM '15，第 891–900 页，美国纽约州纽约市，2015年。计算机协会。[45] Mingdong Ou、Peng Cui、Jian Pei、Ziwei Zhang 和 Wenwu Zhu。保持非对称传递性的图嵌入。载于 Balaji Krishnapuram、Mohak Shah、Alexander J. Smola、Charu C. Aggarwal、Dou Shen 和 Rajeev Rastogi（编），国际知识发现与数据挖掘会议（*SIGKDD*），第 1105–1114 页。ACM，2016年。[46] Jundong Li、Liang Wu、Ruocheng Guo、Chenghao Liu 和 Huan Liu。借助增强型低秩矩阵近似实现多层网络嵌入。载于 Francesca Spezzano、Wei Chen 和 Xiaokui Xiao（编），国际社会网络分析与挖掘会议（*ASONAM*），第 49–56 页。ACM，2019年。[47] Benedek Rozemberczki 和 Rik Sarkar。利用扩散图进行快速的基于序列的嵌入。*CoRR*，abs/2001.07463，2020年。[48] Leo Torres、Kevin S. Chan 和 Tina Eliassi-Rad。GLEE：几何拉普拉斯特征映射嵌入。《复杂网络杂志》，8(2)，2020年。[49] Jiezhong Qiu、Yuxiao Dong、Hao Ma、Jian Li、Kuansan Wang 和 Jie Tang。将网络嵌入视为矩阵分解：统一 DeepWalk、LINE、PTE 和 node2vec。载于国际网络搜索与数据挖掘会议（*WSDM*），WSDM '18，第 459–467 页，美国纽约州纽约市，2018年。计算机协会。[50] Dennis L. Sun 和 Cédric Févotte。用于采用 β 散度的非负矩阵分解的交替方向乘子法。载于国际声学、语音与信号处理会议（*ICASSP*），第 6201–6205 页，2014年。[51] Claire Donnat、Marinka Zitnik、David Hallac 和 Jure Leskovec。通过扩散小波学习结构节点嵌入。载于国际知识发现与数据挖掘会议（*SIGKDD*），KDD '18，第 1320–1329 页，美国纽约州纽约市，2018年。计算机协会。[52] Dingqi Yang、Paolo Rosso、Bin Li 和 Philippe Cudre-Mauroux。NodeSketch：通过递归草图实现高效图嵌入。载于国际知识发现与数据挖掘会议（*SIGKDD*），KDD '19，第 1162–1172 页，美国纽约州纽约市，2019年。计算机协会。[53] Yoav Goldberg。自然语言处理神经网络模型入门。《人工智能研究杂志》，57:345–420，2016年。[54] Andrea Chiorrini、Claudia Diamantini、Laura Genga、Martina Pioli 和 Domenico Potena。将流程结构嵌入活动以实现流程映射与比较。载于 Silvia Chiusano、Tania Cerquitelli、Robert Wrembel、Kjetil Nørvåg、Barbara Catania、Genoveva Vargas-Solar 和 Ester Zumpano（编），数据库与信息系统新趋势（*ADBIS*），第 1652 卷，第 119–129 页。施普林格，2022年。[55] Chiara Di Francescomarino、Chiara Ghidini、Fabrizio Maria Maggi、Giulio Petrucci 和 Anton Yeshchenko。窥见未来：在预测性业务流程监控中利用先验知识。载于 Josep Carmona、Gregor Engels 和 Akhil Kumar（编），业务流程管理（*BPM*），《计算机科学讲义》第 10445 卷，第 252–268 页。施普林格，2017年。[56] Li Lin、Lijie Wen 和 Jianmin Wang。MM-Pred：一种用于多属性事件序列的深度预测模型。载于 Tanya Y. Berger-Wolf 和 Nitesh V. Chawla（编），数据挖掘国际会议（*SDM*），*SDM 2019，2019*年*5*月*2*日至*4*日，加拿大艾伯塔省卡尔加里，第 118–126 页。SIAM，2019年。[57] Andrea Burattin。PLG2：用于在线和离线环境的多视角流程随机化与模拟，2015年。[58] Andreas Daffertshofer、Claudine JC Lamoth、Onno G Meijer 和 Peter J Beek。主成分分析在研究协调与变异性中的应用：教程。《临床生物力学》，19(4):415–428，2004年。

[59] Ana C Lorena、Luís PF Garcia、Jens Lehmann、Marcilio CP Souto 和 Tin Kam Ho。你的分类问题有多复杂？关于衡量分类复杂度的综述。*ACM Computing Surveys (CSUR)*，52(5)：1–34，2019年。[60] Yutaka Sasaki 等人。F 值的真相。*Teach tutor mater*，1(5)：1–5，2007。[61] Nick Russell、ter Ahm Arthur Hofstede、Wil M.P. van der Aalst 和 Na Nataliya Mulyar。工作流控制流模式：修订后的观点。2006。[62] Fábio Bezerra 和 Jacques Wainer。流程感知信息系统日志中轨迹的异常检测算法。信息系统，38(1)：33–44，2013。[63] Timo Nolle、Stefan Luettgen、Alexander Seeliger 和 Max Mühlhäuser。Binet：多视角业务流程异常分类。信息系统，第 101458 页，2019年。[64] Paolo Ceravolo、Gabriel Marques Tavares、Sylvio Barbon Junior 和 Ernesto Damiani。在线流程挖掘的评估目标：概念漂移视角。*IEEE* 服务计算汇刊，2020年。[65] Gabriel Marques Tavares、Paolo Ceravolo、Victor G Turrisi Da Costa、Ernesto Damiani 和 Sylvio Barbon Junior。在线流程挖掘中的重叠分析阶段。载于服务计算国际会议（*SCC*），第 167–175 页。IEEE，2019年。[66] Vincenzo Pasquadibisceglie、Annalisa Appice、Giovanna Castellano、Nicola Fiorentino 和 Donato Malerba。STARDUST：一种从轨迹流中发现演化模型的新型流程挖掘方法。*IEEE* 服务计算汇刊，第 1–14 页，2022年。[67] Ghada Elkhawaga、Mervat Abu-Elkheir 和 Manfred Reichert。预测性流程监控结果的可解释性：你能看到我的数据问题吗？应用科学，12(16)：8192，2022年。[68] Williams Rizzi、Chiara Di Francescomarino 和 Fabrizio Maria Maggi。预测性流程监控中的可解释性：理解何时有助于改进。载于 Dirk Fahland、Chiara Ghidini、Jörg Becker 和 Marlon Du- mas 编，业务流程管理论坛 - *BPM*，第 392 卷，商业信息处理讲义，第 141–158 页。施普林格，2020年。[69] Gabriel Marques Tavares 和 Sylvio Barbon Junior。通过元学习实现流程挖掘编码，以增强异常检测。载于欧洲数据库与信息系统进展会议，第 157–168 页。施普林格，2021年。[70] Gabriel Marques Tavares、Sylvio Barbon Junior、Ernesto Damiani 和 Paolo Ceravolo。利用元学习选择最优轨迹聚类流水线。载于巴西智能系统会议（*BRACIS*），第 150–164 页。施普林格，2022年。[71] Gabriel Marques Tavares、Sylvio Barbon Junior 和 Ernesto Damiani。通过元学习实现流程发现自动化。载于 Mohamed Sellami、Paolo Ceravolo、Hajo A. Reijers、Walid Gaaloul 和 Hervé Panetto 编，协作信息系统（*CoopIS*），第 205–222 页，尚，2022年。施普林格国际出版。[72] Alexei Baevski、Wei-Ning Hsu、Qiantong Xu、Arun Babu、Jiatao Gu 和 Michael Auli。data2vec：用于语音、视觉和语言自监督学习的通用框架。载于 Kamalika Chaudhuri、Stefanie Jegelka、Le Song、Csaba Szepesvári、Gang Niu 和 Sivan Sabato 编，国际机器学习会议（*ICML*），第 162 卷，机器学习研究论文集，第 1298–1312 页。PMLR，2022年。