arXiv:2603.06703v1 [cs.DB] 5 Mar 2026

预印本。已被第21届软件工程新方法评估国际会议（ENASE 2026）论文集接受发表。最终出版版本将刊登于SciTePress 数字图书馆。

第五图范式（5GNF）：用于属性图元数据规范化的基于特征的框架

Yahya Sa'd[1], Vojtˇech Merunka[1,2],RenzoAngles[3]1布拉格捷克生命科学大学经济与管理学院信息工程系，布拉格，捷克共和国2布拉格捷克理工大学核科学与工程学院软件工程系，布拉格，捷克共和国3塔尔卡大学工程学院计算机科学系，塔尔卡，智利 *sady@pef.czu.cz, merunka@pef.czu.cz, vojtech.merunka@fjfi.cvut.cz, rangles@utalca.cl*

关键词: 属性图、元数据规范化、特征节点、特征依赖、图数据库设计。

摘要: 图数据库正日益应用于依赖丰富元数据的软件系统中，但当前的建模实践往往会在多个节点间重复存储元数据，导致冗余和语义不一致。本文提出 5GNF，这是一种基于特征的规范化框架，通过将元数据组织为规范特征节点，并以 HAS TRAIT 关系将其连接起来。我们对特征依赖（tFDs）进行形式化定义，介绍 Trai-tExtrac̄tion5GNF 算法，并在 Neo4j 中实现该方法。为评估其实际应用价值，我们将 5GNF 应用于广泛使用的 Northwind 真实数据集，该数据集包含大量重复的位置和运输元数据。规范化将所有重复出现的元数据外置为可复用的特征，消除了数千个冗余属性实例，降低了模式复杂度，并简化了分析查询。得益于更具选择性的元数据访问路径，其性能仍具有竞争力。结果表明，5GNF 为元数据建模提供了一种可复现、语义精确且面向工程实践的方法，与 ENASE 对概念建模和软件设计方法的关注高度契合。

1 引言

图数据库日益成为现代信息系统的核心，这些系统依赖高度互联的数据，包括知识图谱、推荐系统和科研信息管理平台（Robinson et al., 2015; Bechberger and Perryman, 2021; Patil et al., 2024）。尽管属性图模型提供了强大的建模能力，但大规模部署中仍经常出现冗余、不一致和模式灵活性有限等问题——当元数据在异构节点和关系中被反复嵌入时，这些问题尤为明显（Ibrahim, 2023; Mazein et al., 2025）。

在本文中，元数据规范化统一适用于节点和关系，因为属性图支持将边上的属性作为一等元素。

在实践中，描述来源、时间有效性、监管背景或组织范围的元数据，常常会在多个图模式中重复，导致——维护成本和治理复杂性上升（Lu 等，2021；Koh 等，2019）。现有图建模实践对模块化和可复用元数据表示的支持有限，导致模式紧密耦合且难以演进。

规范化理论针对关系系统中的类似问题提出了解决方案，并最终形成了第五范式（5NF）（Codd, 1970）。图规范化的扩展包括 Frisendal 提出的图范式（GNF）（Frisendal, 2022），以及 Merunka 等人提出的前四种图范式（1GNF–4GNF）（Merunka et al., 2024）。后续关于图函数依赖（gFDs）和图唯一性约束（gUCs）的研究表明，基于依赖的规范化能够减少数据层面的冗余（Skavantzos and Link, 2025; Skavantzos and Link, 2023）。然而，这些方法仍以实体为中心，并未涉及可复用元数据结构的规范化。

本文介绍了第五图范式（*5GNF*），这是一种基于特质的规范化框架

它将规范化从数据层提升至元数据层。其核心构造是特征节点，这是一种原子化、可复用的模式组件，通过明确的 HAS TRAIT 关系与图元素相连。在设计科学研究方法论（DSRM）（Peffers 等，2007）以及基于组合的建模原则（Ambler，2003；Dori，2002；Dori，2016）的指导下，该框架形式化了元数据规范化的条件，并定义了系统化的特征提取与模式重写流程。

在真实世界数据集上进行的实验评估表明，5GNF 能提升元数据复用率并增强模式模块化，从而支持更易维护且更具互操作性的图设计。我们将 5GNF 定位为图规范化层级（1GNF–5GNF）的自然终点（Skavantzos 和 Link，2025；Skavantzos 和 Link，2023），将规范化理论从节点级和边级数据扩展到当代图应用中占主导地位的元数据结构。

尽管在图数据库中，将注释或元数据表示为独立节点是一种已知的建模惯例，但我们的贡献并非这一惯例本身。我们提出了一种图范式，以规范性方式阐明何时应将元数据外置；以特征函数依赖（tFDs）的形式提供基于依赖关系的形式化定义，并明确正确性标准；同时将该方法作为一种可重复的规范化过程进行评估，衡量其对元数据冗余、复用和查询成本的影响。

本研究延续了一项关于跨数据层与元数据层进行规范化的长期研究工作，基于面向对象概念建模的早期成果，并将其扩展至属性图的依赖驱动规范化。

本文其余部分结构如下。第2节回顾相关工作。第3节形式化5GNF框架。第4节介绍特征提取算法。第5节给出一个说明性示例。第6节报告实验评估，随后第7节讨论启示，第8节说明局限性，第9节给出结论。

2 背景与相关工作

2.1 关系数据库中的规范化

规范化自关系数据库理论问世以来一直是一个基础性概念，自引入-

关系模型以来。相继的范式——最终达到第五范式（5NF）——确立了原则性机制，用以减少冗余、保持依赖关系，并通过函数依赖关系、多值依赖关系和连接依赖确保无损分解。这些原则持续为现代模式设计实践提供指导。

2.2 从关系规范化到图规范化

随着图数据库得到采用，规范化概念被扩展到表格结构之外。Frisendal的图范式（GNF）引入了针对图的早期规范化原则，强调原子性与语义清晰性。Merunka 等人随后形式化了前四种图范式（1GNF–4GNF），处理属性图中的冗余、原子值和高阶结构，其关键洞见是节点与关系都需要规范化。

前四种图范式逐步处理数据层面的冗余。**1GNF** 要求节点和边属性采用原子值。**2GNF**将共享的属性值提取到独立节点中，以便复用。**3GNF** 分离语义上独立于其原始父实体的值。**4GNF**为共享的数据类型引入规范节点，从而在整个模式中提供类型级抽象。

在此基础上，Skavantzos 和 Link 提出了图函数依赖（gFDs）和图唯一性约束（gUCs），为经典依赖理论提供了图层面的对应概念。他们的工作表明，由依赖关系驱动的规范化可以减少属性图中节点级和边级数据的冗余，并提升其完整性。

2.3 元数据级规范化的动机

尽管图规范化取得了重大进展，现有图范式仍主要以实体为中心，侧重于规范化节点级和边级属性。虽然这有助于减少数据层面的冗余，但它们并未处理描述整个模式中的实体和关系的重复出现的元数据，例如时间有效性、来源信息、监管状态、类别成员关系或地理背景。

在实际的图模式中，此类元数据常常直接嵌入异构

节点和关系中。这种做法会导致元数据值系统性重复、命名和类型约定不一致、引入新元数据维度时扩展性有限，以及查询模式反复筛选不同实体类型中等价的元数据。

这些局限性表明，有必要在元数据层引入专门的规范化层。第五图范式（5GNF）通过定义何时应将重复出现的元数据外部化为规范模式组件，弥补了这一空白。在 5GNF 中，可复用元数据被表示为原子特征节点，并且仅通过显式的 HAS TRAIT 关系与图元素关联，从而能够基于组合进行元数据建模，而无需依赖重度依赖继承的模式结构。

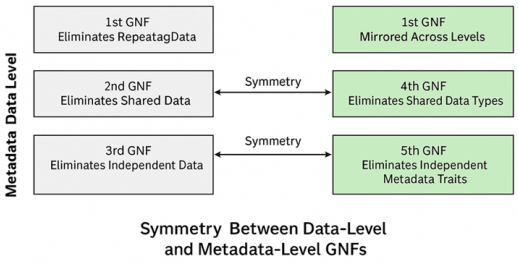


图 1：数据层图范式（1GNF–3GNF）与 4GNF 和 5GNF 中引入的元数据层细化之间的概念对称性。

2.4 与 RDF-star 和元属性图的比较

RDF-star 和元属性图模型支持元数据注释，但并不强制执行规范化原则（Hartig，2017年；Angles，2018年）。RDF-star 将元数据直接嵌入三元组中，而元属性图模型允许在属性上定义属性；这两种方法都允许元数据重复，并且无法提供保持依赖关系或无损分解的保证。

相比之下，5GNF 为元数据定义了一种规范化框架，明确将可复用元数据与承载数据的图元素区分开来。5GNF 不会将元数据直接嵌入节点或关系中，而是将反复出现的元数据外置到规范结构中，以支持无损且依赖保持的分解。该框架所依据的形式化概念将在第3节介绍。

表1：5GNF与RDF-star及元属性图模型的概念比较。

特征	5GNF	RDF-star	元属性图
消除元数据冗余	✓	×	×
依赖保持	✓	×	×
规范元数据复用	High	Low	中
规范化理论	✓	×	×
无损分解	✓	未定义	未定义
模式模块化	High	Low	中

2.5 总结

先前的研究展示了从关系规范化到依赖驱动的图规范化这一发展历程。5GNF 通过为元数据层定义规范化规则，推动了这一进程，并填补了现有图范式留下的空白。

3 第五图范式（5GNF）

第五图范式（5GNF）扩展了 Merunka 等人提出的对称图规范化层次结构（Merunka 等人，2024）。前四种图范式分别消除了重复值、共享值、独立值和类型层级抽象，但仍未解决与多个节点和关系相关联的语义独立元数据的重复问题。5GNF 通过为元数据特征引入一种规范且可复用的表示方式，弥补了这一缺口。

我们假定读者熟悉 Merunka 等人提出的第四图范式（4GNF）的正式定义（Merunka 等人，2024），该定义是理解 5GNF 的直接前提。

在此语境下，规范特征节点表示语义上独立的元数据值在模式级的唯一表示。对于每个这样的元数据值，模式中都恰好存在一个对应的特征节点，所有共享该值的图元素都必须引用同一个特征节点。

规则 5（5GNF）。如果图模式满足 4GNF（Merunka 等人，2024）的条件，并且每个不依赖于特定节点或关系类型而具有独立含义的元数据值都表示为规范特征节点，则该图模式属于第五图范式（5GNF）。此类元数据必须提取到专用特征节点中，并且只能通过显式的 HAS TRAIT 边与图元素关联。只有所有实例都符合此规则，模式才属于 5GNF。

在本研究中，语义独立性以基于规则的方式进行界定。如果某个元数据元素在多个节点或边之间具有稳定的解释，可以

在无需依赖上下文重新解释的情况下复用，并且能够独立于其所注释实体的模式而演化，则该元数据元素被视为具有语义独立性。这一定义足以指导 5GNF 中的规范化决策。

我们通过一份简明清单来操作化语义独立性，该清单涵盖以下三个条件：跨类型复用、解释不受上下文影响，以及独立演化。完整的、基于依赖关系的形式化刻画留待未来工作。

我们使用术语 *节点类型* 来表示一种标签（或一组标签），用于描述图模式中具有共同结构角色的一类节点。*元数据值* 指描述或限定节点或边的属性值，而非构成主要领域数据的值；此类值可能在多个节点或边中重复出现。

特征节点 是一种模式级节点，表示单个语义独立的元数据值。引入特征节点，是为了将可复用的元数据从承载数据的节点和边中抽离出来，确保每个此类元数据值只表示一次，并能在整个图中得到一致共享。

定义 5. 设节点类型A 的实例具有元数据值 $data(A) = \{t_1, t2, \ldots, t_k\}$。如果元数据值$t_i$ 在语义上独立——即出现在多个节点中，或代表领域级概念——则必须将其表示为唯一的特征节点 τ_i，使得$data(\tau_i) = \{t_i\}$。所有共享t_i 的节点都必须通过 $(A) \xrightarrow{\text{具有特征}} (\tau_i)$ 与 τ_i 相连。当每个此类元数据值都恰好出现一次，并表示为规范的特征节点时，该模式即处于 5GNF。

直观说明。 5GNF 通过将位置、类别、有效期或监管状态等共享且可复用的特征提升为显式的特征节点，消除了元数据重复。由此形成对称的元模型，数据层和元数据层的语义都通过图结构以统一的方式表示。通过将元数据外置，5GNF 提高了一致性、复用性和清晰度，同时避免了物理重复。

3.1 预备知识与符号说明

我们采用属性图模型 $G = (V, E, \lambda, P)$，其中 V 是节点集合，$E \subseteq V \times V$ 是有向边集合，$\lambda : (V \cup E) \to L$ 是一个标记函数，用于从标签域L 为节点和边分配标签，而 P 是一个部分属性函数。

遵循（Skavantzos和Link，2025；Skavantzos和Link，2023），我们将数据级属性与元数据级特征区分开来。数据级属性编码主要领域信息，且该信息特定于各个节点或边，例如姓名、标识符、数量或事务值。相比之下，元数据级特征描述可复用的、情境性的或描述性的特征——例如来源、时间有效性、监管状态或分类——这些特征可统一适用于多个节点或边，且其语义独立于特定实体实例。

表 2 总结了 5GNF 形式化定义中使用的符号，并将在后续定义和理论陈述中引用。

表 2：5GNF 形式化框架中所用符号汇总。

符号	含义
$G = (V, E, \lambda, P)$	属性图
$T \subseteq V$	特征节点集
具有特征	特征关联边
$tFD: X \to Y$	特征函数依赖
Σ	特征依赖集
X^+	特征集X的闭包

在本工作中，我们引入特征这一概念，将其作为属性图模式中可复用元数据的概念抽象。特征表示一种语义上独立的描述性特性，可统一关联到多个节点或关系，而独立于其领域特定身份。与领域实体不同，特征并不建模现实世界中的对象，也不参与分类或继承层次结构；相反，它们刻画的是图元素如何被描述，而非该元素是什么。

特征针对属性图中一个常见的建模问题：时间有效性、来源或地理背景等元数据尽管语义稳定，却被反复嵌入异构的模式元素之中。通过将这些特性外化为显式且可复用的模式组件，特征使模块化的元数据建模成为可能，并为元数据层面的规范化提供了原则性基础。

使用“特征”这一术语，是为了强调一种组合式抽象：图元素可以关联多个特征，同一特征也可在多个元素之间共享，而这并不意味着类型隶属、继承或特化。

示例：考虑一个包含概念节点类 Project、Publication 和 Grant 的图模式，每个节点类都包含开始日期、结束日期和资助机构等元数据属性。在传统的属性图模式中，这些元数据会在各节点类之间重复。在基于特征的模式中，

重复出现的元数据会被抽离出来，分别纳入 TemporalTrait 和资助特征，并通过显式的 HAS_TRAIT 关系与 Project、Publication 和 Grant 关联。特征不表示领域实体，也没有独立的领域级实例；它们仅作为可复用元数据的规范表示。这一转换消除了元数据重复，并支持在异构节点类之间进行统一的元数据查询。

3.2 特征节点

定义 1（特征节点）。特征节点 $t \in V$ 是一个模式级节点，其标签包含专用的模式标签“特征”，表示一种原子且语义独立的元数据特征。标签“特征”专门用于标识特征节点，并与用于对领域节点和边进行分类的一般标签域 L 分开处理。

特征节点可重复使用且可组合，并且仅通过显式的 HAS_TRAIT 边与节点或边关联。由特征节点表示的元数据的典型示例包括时间有效性、地理位置、监管或合规状态、来源以及类别描述符。

这种抽象符合 ISO/IEC 39075 图查询语言（GQL）标准（ISO/IEC JTC 1，2023），该标准倡导在属性图中使用规范描述符并明确遵循模式。特征节点将这些原则付诸实践，使元数据设计具备模块化特性、无需继承，并能在异构图元素之间一致地复用元数据。

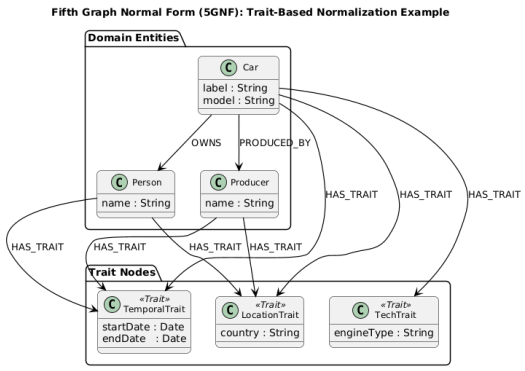


图 2：5GNF 中基于特质的规范化。领域实体表示主要应用对象（例如汽车、人、生产者），其属性编码特定于领域的数据。可重复使用且语义独立的元数据特征（例如时间有效性、位置或发动机类型）被提取为规范特征节点，并且仅通过 HAS_TRAIT 边与领域实体关联。

在此示例中，领域实体对应于对应用领域核心对象进行建模并承载主要数据属性的节点。特征节点用于捕获语义独立于特定实体类型、且可能在多个节点中反复出现的元数据。通过将此类元数据外置到规范特征节点中，5GNF 消除了重复，使元数据得以一致地复用，并清晰地区分了领域级数据与元数据级特征。

3.3 特征依赖

令 $T \subseteq V$ 表示图模式中所有特征节点的集合。在此语境下，特征 是指由特征节点表示的元数据值。每个特征节点恰好对应一个原子且语义独立的元数据特征。

定义 2（特征依赖）。给定特征节点集合 $X, Y \subseteq T$，若对于图中与 X 中的所有特征相关联的每个节点或边，都存在与 Y 中的特征唯一关联的情形，则特征依赖 $X \to Y$ 成立。

特征依赖将经典函数依赖（Codd，1970）和图函数依赖（Skavantzos 和 Link，2025；Skavantzos 和 Link，2023）推广到元数据层，从而能够对元数据特征的冗余消除、一致性和依赖保持进行形式化推理。

特征依赖满足与 Armstrong 公理类似的推理规则：

- **自反性：** 如果 $Y \subseteq X$，那么 $X \to Y$。
- **增广：** 如果 $X \to Y$，那么 $XZ \to YZ$ 对任意 $Z \subseteq T$ 均成立。
- **传递性：** 如果 $X \to Y$ 且 $Y \to Z$，那么 $X \to Z$。

这些推理规则支持闭包计算，并为元数据层的依赖保持分解提供了形式化基础。

3.4 5GNF 的定义

我们使用术语图模式来表示属性图的抽象规范，其中包括节点类型和边类型的集合、它们的标签、允许的属性以及结构约束，并且独立于具体的图实例。

在此语境下，元数据规则指的是一种反复出现的元数据模式，其语义独立于特定的节点或边类型，并且会出现在图模式的多个元素中，因此适合进行抽象和复用。

如果某个元数据值表示一个单一且不可分割的语义单元，则称其为原子的；如果其含义不依赖于它所注释的节点或边的身份或类型，则称其为语义独立的。

定义 3（第五图范式）。当且仅当满足以下条件时，图模式处于第五图范式（5GNF）：

- 模式中的每个元数据规则恰好由一个特征节点表示；
- 每个特征节点表示单一的原子且语义独立的元数据值；并且
- 域元素与元数据之间的所有关联都仅通过具有特征的边来表达。

该定义通过引入一个用于可复用元数据的原则性规范化层，扩展了 Merunka 等人（Merunka 等人，2024）提出的图规范化层级（1GNF–4GNF）。

3.5 基于特性的语义与组合

在本节中，语义属性指由特征节点承载的一种抽象的、具有意义的特征，例如分类、有效性约束、上下文解释或监管含义。它会影响对相关节点或边的解读，但本身不构成主要领域数据。

特性语义以组合方式传播：如果节点或边与某个特征节点相关联，就会继承该特征的语义属性。与经典继承不同，这种传播是在查询时进行评估，而不是在模式中实体化，从而避免元数据重复。

通过以组合取代继承，5GNF 可防止模式膨胀，并支持模块化、可复用的元数据设计，这符合软件工程最佳实践。

3.6 5GNF 的理论性质

对于元数据规范化，5GNF 满足以下理论性质：

无损性。设G是符合模式S 的图实例，且G'是对G应用 5GNF 规范化后的结果。那么，存在一个重构函数R ，使得$R(G') = G$ 在元数据值方面成立。也就是说，规范化过程中不会丢失任何元数据信息，因为所有元数据值都可以通过其关联的特征节点恢复。

最小性。在 5GNF 规范化模式中，每个特征节点恰好表示一个原子且语义独立的元数据值。合并任意两个不同的特征节点，都会破坏原子性或违反现有的特征依赖。因此，5GNF 模式中的特征节点集合相对于所表示的元数据规律而言是最小的。

同构意义下的唯一性。对于给定的图模式S，由S 推导出的所有有效 5GNF 规范化模式，在其特征节点和具有特征边方面均彼此同构。任意两个此类模式仅在特征节点的命名上有所不同，并对应于相同的语义独立元数据等价类。

4 5GNF 规范化

4.1 概述

所提出的方法通过将属性图模式 $S = (V_T, E_T, \lambda_T, K)$转换为规范化模式$S'$，实现第五图范式（5GNF）的原则。其中，$V_T$ 和E_T 分别表示节点类型集和边类型集，λ_T 是作用于这些类型的类型标注函数，K 则是模式级定义的属性键集。

一个属性键规定了可分配给符合该模式的节点或边实例的属性名称及其预期含义。属性键定义了数据级属性和模式级特征，并在该模式的所有图实例之间共享。

该方法以模式S 及特征依赖集 Σ作为输入，并生成一个规范化模式，其中所有可复用元数据均以特征节点的形式显式表示，并且仅通过具有特征边与节点类型或边类型关联。

规范化过程分为三个主要阶段：（i）检测由K中的属性键定义的重复出现的元数据属性；（ii）将这些属性提取到规范特征节点中；（iii）重写模式，使节点和边类型显式引用特征节点，同时保留特质依赖语义。

阈值 τ 是我们原型中用于优先处理基数较低的重复出现的元数据的一种实现层面启发式方法。它不会影响 5GNF 的形式化定义；该定义不依赖任何特定的阈值选择。

4.2 算法规范

算法 1 描述了 TraitExtraction5GNF 过程，该过程将属性图模式 转换为等价的第五图范式（5GNF）模式。该算法识别由属性键定义的重复出现的元数据，将此类元数据提取为规范特征节点，强制执行图函数依赖（gFDs），并应用保持依赖关系的无损分解步骤，以确保满足 5GNF 的所有形式化要求。

图函数依赖（gFD）约束属性图中节点和边之间属性值的共现方式，将经典函数依赖推广到图结构数据（Skavantzos 和 Link，2025；Skavantzos 和 Link，2023）。该算法使用 gFD 来检测冗余，并在保留语义等价性的同时指导规范化步骤。

输入：属性图模式 S $=(V_T,E_T,\lambda_T,K)$；属性键 K；图函数依赖集合 Σ_{gFD}输出：第五图范式（5GNF）下的规范化模式 S′ S TraitSet $\leftarrow \emptyset$ TraitMap $\leftarrow \{\}$// 阶段 1：特征检测与创建 对每个属性键 k $\in$K 执行 计算 $D_k = \{val \mid val$ 在键 k 下出现 $\}$if $1 \leq |D_k| \leq \tau$ then 对每个 val $\in D_k$ 执行 创建特征节点 t =Trait(k,val) TraitSet $\leftarrow$ TraitSet $\cup\{t\}$TraitMap[(k,val)] $\leftarrow$t 结束 结束 结束 // 阶段 2：元数据抽取 对每个节点或边类型 x $\in V_T \cup E_T$执行 对 x 上定义的每个属性键 k 执行 如果 (k,val) $\in$ TraitMap 则 将 x 与 TraitMap[(k,val)]通过 具有特征 关联 从 x 中移除属性键 k 结束 结束 结束 // 阶段 3：依赖实施 对每个 X $\rightarrow$Y $\in \Sigma_{gFD}$ 执行 对每种 X 特征组合，确保 Y 特征唯一 结束 // 阶段 4：无损分解 仅当元数据重构无损时才应用分解步骤 返回 S′算法 1：TraitExtraction5GNF(S,K Σ_{gFD})

4.3 运行示例

为了说明该算法的行为，我们采用图 2 所示的运行示例。该示例在第3节中引入，基于一个简化的未规范化（0GNF）模式，其中包含人、汽车和生产者等实体。在初始模式中，日期、位置和技术特征等重复的元数据属性直接嵌入节点中，从而导致冗余和潜在的不一致。

在逐步进行规范化的各个阶段中，该算法通过检测重复出现的元数据值，并将其提取为可复用的特征节点（例如 TemporalTrait、LocationTrait、TechTrait），逐步转换此运行示例。随后，每个实体都会通过 HAS TRAIT 关系关联到相应的特征，从而消除重复

同时保留语义。

后续阶段将验证所得分解是否无损，以及是否不再存在非平凡依赖。所得的 5GNF 模式已在前文图 2 中展示。图 2 构成了本节所述 Cypher 语句的概念基础。每个查询都直接对应于 5GNF 规范化过程中引入的一项转换或约束，包括创建特征节点、建立具有特征关联，以及移除重复的元数据属性。

这些转换共同展示了同一非结构化模式如何从 0GNF 逐步规范化为 5GNF，最终通过特征节点实现明确的元数据规范化。

4.4 Cypher 脚本的可复现性

从 0GNF 到 5GNF 的所有规范化阶段均已使用 Cypher 在 Neo4j 中实现，并已公开以确保可复现性。

表 3：规范化阶段（0GNF–5GNF）中使用的 Cypher 脚本。

范式阶段	Cypher 脚本	描述
0GNF	0GNF.cypher	带有冗余属性的原始未规范化图
1GNF	1GNF.cypher	一级实体抽取
2GNF	2GNF.cypher	从汽车中抽取出的生产者
3GNF	3GNF.cypher	位置被分离为独立节点
4GNF	4GNF.cypher	技术元数据分解
5GNF	5GNF.cypher	特征提取与最终规范化结构

所有脚本及配套材料均可在随附的代码仓库[1]中公开获取。

4.5 正确性保证

所提出的算法满足（5GNF）所要求的核心理论性质。首先，它保留了所有特征依赖：对元数据特征定义的依赖关系在规范化后仍然有效，因为元数据在不损失语义的情况下被外置。其次，该转换在元数据语义方面是无损的，因为每个被移除的属性值都可以通过关联的特征节点和具有特征关系重构。最后，该算法通过仅为重复出现的元数据值创建特征节点来确保最小性，从而避免引入虚假或冗余的特征。

4.6 复杂度分析

令 $G = (V, E, \lambda, P)$ 表示符合给定图模式的属性图实例。令 $n = |V|$ 为节点实例的数量，$m = |E|$ 为边实例的数量，$k = |K|$ 为模式级定义的不同属性键的数量。令 Σ 表示规范化过程中考虑的图函数依赖（或特征依赖）集合。

针对这些实例级参数，分析 5GNF 规范化过程的复杂度。

特征检测需要扫描所有节点和边属性，以识别重复出现的元数据值，最坏情况下耗时为 $O(nk + mk)$。特征提取和属性重写对相同集合进行线性遍历，因此仍保持 $O(nk + mk)$ 的复杂度上界。

依赖实施取决于依赖集 Σ 的大小以及提取出的特征节点数量。设 $|\Sigma|$表示依赖关系的数量，$|T|$ 表示规范化过程中生成的特征节点数量。因此，实施依赖需要 $O(|\Sigma| \cdot |T|)$ 时间。

总体而言，规范化过程的运行时间为$O(nk + mk + |\Sigma| \cdot |T|)$。

4.7 Neo4j 中的原型实现

该算法使用 Cypher 和 APOC 程序在 Neo4j 5.x 中实现，从而能够检测和提取特征，并在不损失语义的情况下执行规范化约束。

4.8 用例：科研项目元数据

为展示该算法的实际适用性，我们将其应用于科研信息管理场景。项目、资助和出版物等实体包含重复出现的元数据属性，例如资助机构、开始日期、国家和项目状态。这些属性被识别为语义上独立的元数据特征。

该算法将这些值提取到可复用的特征节点中，并用显式的具有特征关系替代嵌入式属性。因此，元数据复用得以提升，查询模式得以简化，而元数据更新和验证等维护操作也实现了集中管理。例如，时间查询可以统一表示为：

```
MATCH (e)-[:HAS_TRAIT]->(t:时间特征) WHERE t.开始日期 <= date("2025-01-01") RETURN e
```

由于元数据特征采用集中管理的方式，对单个特征节点的更新会自动传播到所有关联实体，从而提升一致性并改-

在整个模式中保持一致。

[1] https://github.com/yahyazuh/5GNF-normalization-example

5 说明性案例研究

5.1 场景概述

为展示第五图范式（5GNF）的实际效果，我们采用一个简化的研究信息场景，涉及项目、出版物、研究人员和资助实体。在传统的属性图模式中，诸如开始日期、资助机构、国家或许可证等元数据会在异构节点和边中反复出现。这种设计会导致冗余、元数据用法不一致，并增加维护工-

5.2 5GNF前模式

在基线（未规范化）模式中，元数据属性直接嵌入实体节点。例如，Project 和 Publication 节点都存储开始日期、结束日期和资助机构等元数据。这种重复会导致模式僵化，并且在查询不同实体类型共享的元数据时，需要重复编写筛选逻辑。

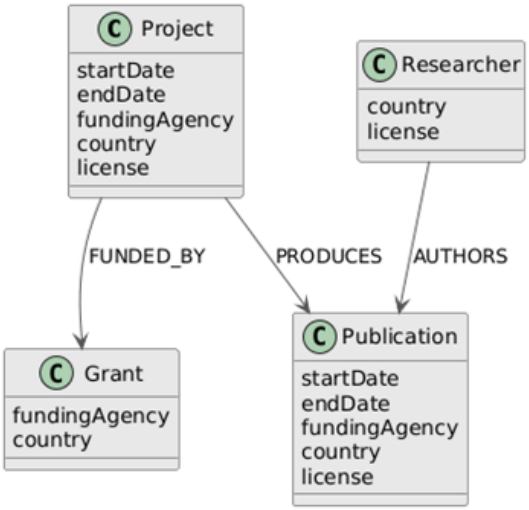


图 3：5GNF前模式，其中冗余元数据嵌入在多个实体类型中。

5.3 转换为 5GNF

应用 TraitExtraction5GNF 程序，通过将重复出现的元数据外置到规范特征节点来重构模式。时间元数据被提取到 TemporalTrait，资助相关元数据被提取到资助特征，地理信息则被提取到地理特质。嵌入式

元数据属性会从实体节点中移除，并由明确的具有特征关系取代。

这一转换保留了依赖语义，使元数据能够在异构实体间复用，并厘清了模式的概念结构。

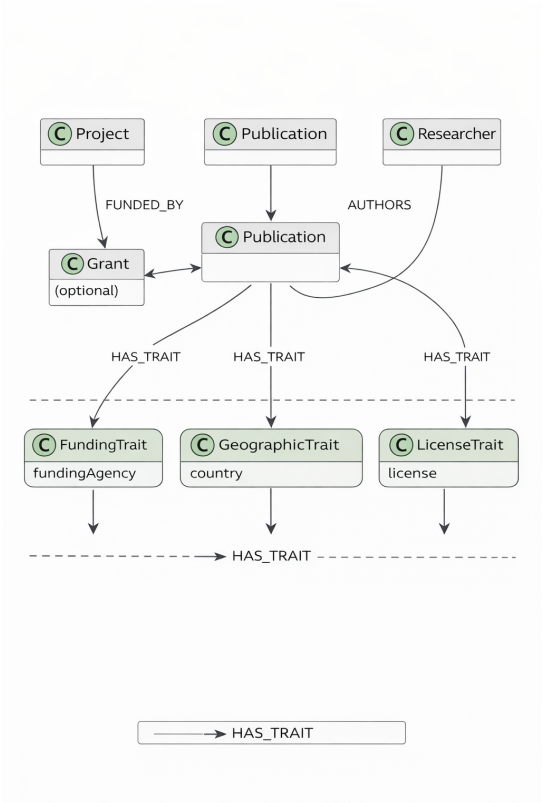


图 4：通过将重复出现的元数据外置为可复用的特征节点，并通过 HAS TRAIT 关系与之关联，得到的 5GNF 规范化模式。

5.4 查询示例

基于特质的规范化简化了涉及共享元数据的查询。无需使用针对实体的特定条件扫描多个实体类型，只需一个模式即可检索由特定机构资助的所有项目：

```
MATCH (p:项目)-[:HASTRAIT]->(t:资助特征) WHERE t.资助机构 ="欧盟委员会" RETURN p.name
```

时态查询同样可以在不同实体类型之间保持统一：

```
MATCH (n)-[:HASTRAIT]->(t:时间特征) WHERE t.开始日期 <= date("2025-01-01") AND date("2025-01-01") <= t.结束日期 RETURN n
```

使用特征节点消除了重复筛选条件和实体特定查询逻辑的需求。

5.5 观察到的改进

转换为 5GNF 带来了多项实际改进。首先，由于多个实体类型共享特征节点，元数据可复用性得以提高。其次，随着元数据定义独立于数据节点演变，模式模块化程度也有所提升。第三，查询构造变得更加简单、统一，以单一的基于特征的模式取代复杂的多实体筛选。最后，可维护性得到增强：更新元数据定义时，只需修改单个特征节点，而不必修改每个嵌入该元数据的实体。

6 实验评估

本节使用公开可用的 Northwind 数据集评估 TraitExtraction5GNF 算法。该数据集通过官方 CSV 分发包导入 Neo4j 5.x。为确保完全可复现，所有导入脚本、规范化流程和查询工作负载均收录于随附的 GitHub 代码仓库中。本评估使用具有代表性的分析查询，考察应用 5GNF 前后的元数据冗余、模式复杂度和查询性能表现。尽管 Northwind 源自关系模式，本评估仍有意选用该数据集，以表明即使在结构良好的数据集中，将其映射到属性图后，元数据冗余和规范化问题依然存在。在实践中，关系型到图的转换经常会将上下文信息（例如来源、时间有效性或地理元数据）作为重复属性嵌入多个节点和边中。因此，Northwind 适合作为基线，可用于独立评估元数据级规范化，而不受领域特定的图结构异常影响。

6.1 数据集与方法论

评估使用了广泛采用的 Northwind 数据集，该数据集最初作为关系型基准数据集发布，随后被改编为多种数据模型。我们使用《从使用图数据库的实践视角看概念数据规范化》的作者公开提供的 CSV 文件，将该数据集导入 Neo4j。导入后，图中约有 1,470 个节点和 4,000 多个关系，数千个元数据属性直接嵌入在节点中。

Neo4j 使用动态的基于成本的查询规划器，其运算符选择取决于运行时统计信息、页面缓存状态和基数估计。因此，同一查询的重复执行可能会生成不同的物理执行计划。在应用 5GNF 后，这种现象会更加明显，因为元数据规范化会通过引入规范特征节点和额外关系来修改图结构，同时移除嵌入其中的属性。因此，评估侧重于相对性能趋势，而非不同运行之间执行计划的完全等价性，这与 Neo4j 的查询规划与执行模型相一致。

Northwind 中的元数据冗余尤为突出，主要体现在客户和供应商位置、订单运输元数据以及地区描述符上。这些特征使该数据集非常适合用于评估 5GNF 下的元数据提取与复用。

6.2 基线分析（5GNF 之前）

应用 5GNF 之前，元数据直接嵌入实体节点中，导致客户、供应商和订单实体之间存在大量冗余。为了评估原始模式的表现，我们执行了五个以元数据为重点的分析查询，这些查询能够代表 OLAP 风格图工作负载：

- 按城市列出客户，
- 按国家列出供应商，
- 检索运送至指定国家的订单，
- 按城市和国家筛选客户，
- 匹配位于同一城市的供应商和客户。

每个查询均使用 Neo4j 的 PROFILE 命令执行，以捕获数据库访问（数据库命中次数）、执行时间、内存分配和查询计划运算符。

表 4：应用 5GNF 之前的基线性能结果。

查询目的		数据库访问次数	时间（毫秒）	备注
测试 1	客户元数据扫描	274	175	冗余城市值
测试 2	供应商元数据扫描	88	97	简单元数据分组
测试 3	按国家筛选订单	2,491	94	复杂属性筛选
测试 4	按城市和国家划分客户	298	105	多元数据筛选
测试 5	供应商–客户城市匹配	304	250	笛卡尔积

结果揭示了将元数据直接嵌入节点中所固有的低效性。尤其是，测试 3 会产生开销高昂的属性扫描，而测试 5 则由于缺乏共享元数据结构而触发笛卡尔积。这些模式凸显了对规范元数据表示的需求，而 5GNF 正是为提供这种表示而设计的。

6.3 元数据冗余测量（5GNF 之前）

在规范化之前，Northwind图将所有元数据直接嵌入实体节点中。城市、国家、地区、邮政编码、发货城市和发货国家等属性，在客户、供应商和订单中被反复存储。我们通过统计所有位置相关属性中的嵌入式元数据值总数，以及不同元数据值的数量，对这种冗余进行了量化。

在客户、供应商和订单实体中，我们观察到：

- 嵌入式元数据值：3,200+，
- 不同元数据值：120。

我们将元数据复用率（MRR）定义为元数据出现总次数与不同元数据值数量之比。对于基准模式，由此可得：

$$\mathrm{MRR}_{\mathrm{before}} = \frac{3200}{120} \approx 26.67.$$

该结果表明，平均而言，每个元数据值在图中大约重复 27 次。这种冗余导致查询执行效率低下，因为查询规划器必须反复扫描嵌入式属性，而不是遍历共享元数据结构。

6.4 应用 TraitExtraction5GNF

为了规范化数据集，我们应用 TraitExtraction5GNF 算法，将重复的元数据属性替换为规范特征节点。在 Northwind 数据集中，我们识别并提取了两个主要特征族：

- LocationTrait，包含城市、国家、地区、地址和邮政编码，应用于客户和供应商实体；
- ShippingTrait，包含发货城市、发货国家、发货地区、发货地址和发货邮政编码，应用于订单实体。

我们使用 Cypher 唯一性约束创建特征节点，以确保规范表示。随后，通过显式的具有特征式关系将实体链接到相应的特征节点，然后安全地从原始节点中移除冗余的嵌入式元数据属性。

规范化完全在 Neo4j 中实现，并按以下顺序执行：

- 为特征节点创建唯一性约束，
- 生成所有不同的特征实例，
- 通过精确或部分元数据匹配，将实体关联到特征，
- 验证实体覆盖是否完整，
- 移除冗余的嵌入式属性。

规范化后，图中包含 120 个 LocationTrait 节点和 89 个 ShippingTrait 节点，它们通过 950 多个特征关系相连。所有与位置和运输相关的元数据都仅通过特征节点表示，从而形成了完全规范化的 5GNF 模式。

6.5 转换后测量

应用 TraitExtraction5GNF 后，Northwind 数据集中的所有位置和运输元数据都已完全规范化为规范特征节点。此转换消除了冗余的嵌入式属性，并重构了图，使元数据仅通过显式关系表示。

规范化后的统计数据如下：

- LocationTrait 节点：120，
- ShippingTrait 节点：89，
- HAS_TRAIT 关系： 950+，
- 剩余嵌入式元数据：0%。

按照与之前相同的定义，规范化后的元数据复用率为：

$$\mathrm{MRR}_{\mathrm{after}} \approx 1.74.$$

与基线相比，约移除了 2,991 个冗余元数据值。这几乎彻底消除了属性级重复，并显著提升了元数据复用。

模式复杂度降低。我们将模式复杂度定义为

$$\mathrm{SCM} = |Nodes| + |Edges| + |Attributes|.$$

应用 5GNF 后，属性数量大幅减少，而节点和关系数量则因特征的实体化而适度增加。尽管结构发生了变化，整体模式复杂度仍有所降低，因为 5GNF 前模式中大量嵌入的属性已被消除。这证实了，即使引入了额外节点，5GNF 仍能降低结构复杂度。

6.6 5GNF 后的查询性能

我们在规范化后的图上重新运行了相同的五个分析查询。由于元数据现在通过特征节点表示，因此我们重写了每个查询，使其遍历“具有特征”式关系，而不是扫描嵌入的属性。

需要注意的是，在规范化前后，Neo4j 的基于成本的规划器可能会为同一个逻辑查询选择不同的执行策略。这些差异源于 5GNF 改变了结构基数、减少了重复的属性扫描，并支持基于遍历的筛选。因此，评估侧重于整体性能趋势，而非预期不同运行中会产生完全相同的执行计划。

表 5：应用 5GNF 后的性能结果。

查询	目的	数据库访问次数	时间（毫秒）	备注
测试 1	客户元数据扫描	1,138	102	特征查找取代属性扫描
测试 2	供应商元数据扫描	264	82	共享 LocationTrait 遍历
测试 3	按国家划分的订单	685	106	选择性特征遍历
测试 4	按城市和国家划分客户	613	113	基于特征的筛选
测试 5	供应商–客户城市匹配	544	111	消除笛卡尔积

最显著的改进出现在元数据密集型查询中。对于测试 3（按国家查询订单），数据库访问次数从 2,491 次降至 685 次，减少约 $3.6\times$ 。对于测试 5（供应商与客户匹配），执行时间从 250ms 缩短至 111ms，且基准执行计划中出现的笛卡尔积不复存在。

尽管引入了额外的特征节点，所有查询的运行时间均有所改善或保持稳定。这一表现体现了遍历深度增加与属性过滤减少之间的权衡，证实 5GNF 在大幅降低元数据冗余的同时，仍能保持有竞争力或更优的查询性能。

6.7 不同范式下的消融研究

为阐明 5GNF 的影响，我们比较了应用于 Northwind 数据集的多个规范化阶段中，元数据冗余、复用和模式复杂度的变化。这种消融式比较展示了范式逐步演进如何持续减少冗余并改善模式结构，而 5GNF 则为可复用元数据提供了最强的抽象。

表 6：不同规范化级别（0GNF–5GNF）下的冗余、元数据复用和模式复杂度。

范式	冗余 y	元数据复用 模式复杂度	
0GNF（原始）	高 低		高（数千个嵌入式属性）
3GNF	中–高 低–中		中–高
4GNF	中	中	中
5GNF（特征提取）	Low	High	Low

结果表明，5GNF 通过将元数据整合到规范特征节点中，消除冗余的特征实例，降低整体模式复杂度，并确保异构实体之间的语义一致性，从而提供最强的元数据抽象。较早的范式能够减少数据层面的冗余，但只有 5GNF 能够全面解决元数据重复问题。

6.8 可复现性

本次评估中使用的所有数据集、规范化脚本、查询工作负载和实验结果，均可在配套代码仓库 github.com/yahyazuh/5GNF-normalization-example 中公开获取。

该代码仓库包含：

- /northwind/csv/ – 用于导入 Northwind 数据集的官方 CSV 文件，
- /northwind/import.cypher – 用于将数据集加载到 Neo4j 的 Cypher 脚本，
- /normalization/trait extraction 5gnf.cypher – 5GNF 规范化的完整实现，包括 LocationTrait 和 ShippingTrait，
- /queries/pre 5gnf/ 和 /queries/post 5gnf/ —— 规范化前后执行的所有分析查询，
- /results/ —— 本节报告的性能分析输出和汇总表，
- 使用 Neo4j Desktop 或基于 Docker 的 Neo4j 环境复现实验的详细说明。

这些材料确保实验完全可重复，并使其他研究人员能够独立验证所报告的结果。

7 讨论

7.1 概念贡献

第五图范式（5GNF）是图规范化层级的概念终点。它将保持依赖的分解从数据级属性推广至元数据特征，完成了 Merunka 等人所暗示的对称性（Merunka 等人, 2024），并扩展了 Skavantzos 和 Link 针对属性图中 BCNF 所形式化的基于依赖的规范化原则（Skavantzos 和 Link, 2025; Skavantzos 和 Link, 2023）。通过特征节点与特征依赖（tFDs），5GNF 提供了首个规范化

框架，明确专门面向元数据语义。

7.2 实际收益

5GNF 为基于图的信息系统的设计与维护带来若干实际优势。首先，通过规范特征节点将重复出现的元数据仅表示一次，从而降低冗余。其次，模式模块化得到改善，因为元数据定义可以独立于数据实体演进。第三，通过显式外化元数据结构，使元数据表示与 ISO/IEC 39075（GQL）对齐，从而增强互操作性。这种对齐遵循 ISO/IEC 39075 标准（ISO/IEC JTC 1，2023）所倡导的原则。最后，基于特征的元数据通过支持特征提取、可解释性以及面向治理的数据管理，提升人工智能就绪度。

这些收益在前文给出的贯穿示例和实验评估中均得到一致体现。

7.3 回应常见异议

一个常见的担忧是，引入特征节点可能会增大模式规模。然而，实验评估表明，尽管引入了额外的节点和关系，总体模式复杂度仍净减少，并且元数据重复被完全消除。

8 局限性与未来工作

当前的 5GNF 实现基于 Neo4j 5.x 和 APOC 过程，因此难以直接移植到其他属性图系统。此外，实验评估主要关注模式级指标和具有代表性的分析型工作负载；对大型生产部署规模下的性能进行评估仍有待未来研究。

另一项局限源于 Neo4j 的基于成本的查询规划器具有动态特性。由于运行时统计信息和缓存效应可能导致物理执行计划在不同运行中有所变化，因此性能结果应解读为趋势，而非固定的绝对值。这种行为在动态图库引擎中是意料之中的，并不影响结论的有效性。

未来的研究方向包括利用机器学习技术自动发现特征，并进一步对特征依赖进行理论形式化，将其与 RDF-star 和元属性图模型进行比较基准测试，以及为 ISO/IEC 39075（GQL）标准中考虑特征的模式规范化贡献力量。

9 结论

理论贡献

本文的理论贡献如下：

- 我们提出了第五图范式（5GNF），其定义于属性图模式的元数据层。
- 我们将特质函数依赖（tFDs）形式化为一种依赖框架，用于推理特质冗余和规范化条件，包括必要的推理与闭包性质。
- 我们提出了一种规范化过程，其正确性标准确保无损，并在减少元数据冗余的同时保留依赖语义。

本文介绍了第五图范式（5GNF），这是一种基于特质的规范化框架，将依赖保持分解从数据值扩展到属性图中的元数据特征。通过将重复出现的元数据外置为规范特征节点，5GNF 消除了冗余，提升了模式模块化程度，并使元数据建模与新兴的治理和互操作性标准（包括 ISO/IEC 39075（GQL））保持一致。

通过形式化定义、可执行的规范化算法，以及在 Northwind 数据集上的实证评估，我们证明了 5GNF 能够保留依赖关系、支持无损重构，并在保持有竞争力的查询性能的同时，显著提升元数据复用率、降低模式复杂度。该框架补全了 Merunka 等人所预期的规范化层级（Merunka 等人，2024），并与 Skavantzos 和 Link 的基于依赖的图规范化工作相辅相成（Skavantzos 和 Link，2025；Skavantzos 和 Link，2023）。

总体而言，5GNF为基于图的信息系统中的以元数据为中心的建模提供了有原则的基础，并对互操作性、人工智能就绪度和可解释的数据治理具有重要意义。

参考文献

Ambler, S. W. (2003). 敏捷数据库技术。Wiley。Angles, R. (2018). 属性图：模型、语义和查询语言。载于第*21*届扩展数据库技术国际会议（*EDBT*）论文集，第1–12页。OpenProceedings.org。Bechberger, D. and Perryman, J. J. (2021). 图数据库实战。Manning。Codd, E. F. (1970). 面向大型共享数据库的数据关系模型。《*ACM*通讯》，13(6):377–387。Dori, D. (2002). 对象过程方法论。施普林格。Dori, D. (2016). 基于对象过程方法论和系统建模语言的基于模型的系统工程。施普林格。Frisendal, T. (2022). 向图范式（GNF）问好。Hartig, O. (2017). RDF再化替代方法的基础。*arXiv*预印本 *arXiv:1705.04454*。Ibrahim, M. (2023). 面向多领域模型的图数据库设计。施普林格。ISO/IEC JTC 1 (2023). ISO/IEC 39075:2023 — 信息技术 — 图查询语言（GQL）。国际标准。Koh, J. et al. (2019). 交互式建筑元数据规范化。载于*ACM BuildSys*论文集，第241–250页。Lu, Y. et al. (2021). 数字图书馆中的元数据规范化。Mazein, I. et al. (2025). 系统生物学中的图数据库。《生物信息学简报》，26(1)。Merunka, V., Wijekoon, H., and Ber´anek, P. (2024). 从使用图数据库的实践视角看概念数据规范化。载于企业、业务流程与信息系统建模（*EOMAS*），《商业信息处理讲义》。施普林格。Patil, A. et al. (2024). 通过整合图数据库与机器学习提升银行业欺诈检测能力。*MethodsX*，12:102683。Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). 面向信息系统研究的设计科学研究方法论。《管理信息系统杂志》，24(3):45–77。Robinson, I., Webber, J., and Eifrem, E. (2015).图数据库。O’Reilly Media，第2版。Skavantzos, P. 和 Link, S.（2023）。属性图规范化。*VLDB* 捐赠基金会会议录，16(11):3031–3043。Skavantzos, P. 和 Link, S.（2025）。属性图的第三范式和博伊斯–科德范式。《*VLDB*杂志》。