记法如何演变：一项对支持用户自定义抽象具有启示意义的历史分析

Jingyue Zhang 蒙特利尔大学计算机科学与运筹学研究系（DIRO）加拿大魁北克蒙特利尔
jingyue.zhang@umontreal.ca

J.D. Zamfirescu-Pereira 加州大学洛杉矶分校计算机科学系 美国加利福尼亚洛杉矶
zamfi@cs.ucla.edu

Elena L. Glassman 哈佛大学工程与应用科学学院 美国马萨诸塞剑桥
glassman@seas.harvard.edu

Damien Masson 蒙特利尔大学计算机科学与运筹学研究系（DIRO）加拿大魁北克蒙特利尔
d
amien.masson@umontreal.ca

Ian Arawjo
计算机科学与运筹学系（DIRO）蒙特利尔大学 加拿大魁北克省蒙特利尔
ian.arawjo@umontreal.ca

arXiv:2602.01525v1 [cs.HC] 2 Feb 2026

摘要

传统的人机交互通过形式化
指定的系统（如结构化用户界面和编程语言）进行。近期的人工智能系统承诺通过自然语言和其他记法形式，带来一系列与计算机进行非正式交互的新方式。这些非正式交互随后可以形成形式化表征，但依赖于人类和人工智能都已知的既有形式体系。新颖的形式体系和记法又如何？新的抽象如何随着时间推移而产生、演变并逐步形式化？反过来，又如何明确设计新系统来支持这些过程？我们对记法发展进行了比较历史分析，以识别其中一些相关特征。这些特征包括记法发展的三个社会阶段：发明与孵化、扩散与分化，以及制度化与神圣化；以及三个功能阶段：描述、生成和评估。在这些阶段之内及其间，我们详细阐述了若干模式，例如关联与锚定隐喻的作用、有意义变异的维度，以及类比对齐。最后，我们提出了一些设计启示。

CCS 概念

• **以人为中心的计算 → 人机交互理论、概念与模型**。

关键词

记法演化、历史、抽象、隐喻

1 引言

> “抽象的形成‘可以分解为一系列微小、出乎意料且实用的技能’……[曾经]被视为自发的个人创造力成果的 [科学成就] ……如今则成为可分析、透明的表现……这些表现与集体共享的资源以及更广泛文化中的能指工具箱相联系。”
>
> —Ursula Klein [76,第 241 页-2]

人机交互历来通过形式定义的结构进行中介，例如命令行界面、图形用户界面和编程语言；这些结构能够明确地映射到底层形式模型。人工智能的最新进展带来了一种新范式：从含糊的输入中主观推断人类意图 [165]。这些系统能够借助现有抽象，通过自然语言实现丰富的非正式交互，并迅速将想法形式化。例如，在人工智能代码生成中，成功的交互依赖于人类和人工智能模型在运用这些抽象及在不同抽象之间进行转换方面的互补能力。

然而，人类协作时并不总是遵循预先存在的形式体系。相反，他们可以临时创造新的记法来奠定沟通基础 [22, 34, 143], ，将记法视为灵活可变的资源，而非僵化的系统。当前人工智能技术依赖、复现并放大既有的主导性抽象和记法——它们早已形式化——以实现自身功能。由此我们不禁思考：人类最终是如何发展出新的记法、新的形式体系和新的抽象，并用它们与机器及彼此沟通的？探讨这一问题或许能为未来系统的设计指明方向，使其能够支持记法创造、演化和渐进式形式化 [125]。

在本研究中，我们通过对科学、计算和艺术学科中的记法发展进行平行的比较历史分析 [129] ，为回答这一问题奠定了一些基础。我们的分析确定了记法随时间被创建、演变和形式化的 33 种模式。这些模式在不同历史中大体相通，并被粗略地划分为三个社会阶段（发明/孵化、传播/分化和制度化/神圣化）以及三个功能阶段（描述性、生成性和评价性）。基于这一分析，我们为设计能够创建新抽象的未来系统提出了一组初步启示（第 5 节），其中包括：记法主要通过连接隐喻产生，并且最常见于社会背景而非技术背景；围绕将哪些内容纳入“有意义”范畴（以及因此排除哪些内容）所做的记法设计决策，往往由发明者隐含地作出，但

可以通过实体化 [10]将其明确化，并使之成为可操作的对象。

我们发现，记法的演化始于非正式的构想，这些构想借鉴了发明者触手可及的周遭环境中的既有文化 [3]——尤其是对既有记法的重新混搭，以及对连接与奠基这两种概念隐喻的运用 [79]。随着记法接触到不同的情境和人群，它会不断修订和扩展，并且（有时）传播得更广；由此可能引发争议 [114]，而采用者也可能建立机构来处理争议。尽管记法的可用性在其采纳过程中发挥着中介作用，社会权力以及对环境情境（物质或社会）的适切性也同样发挥中介作用。此后，流行的记法会在物质层面和社会层面被制度化，神圣化的不仅是表征，还有其设计中蕴含的审美与价值观；它们还可在某一学科中充当修辞性边界工作 [55] 。与此同时，形式化符号系统会成为原材料，以新的方式重新部署，构成不断生长的抽象之塔的层层基石，共同推动人类知识的发展。

我们的研究响应了人机交互领域学者的呼吁，主张该领域应更加重视历史主义，以便更好地将技术置于“涌现、变化、延续、衰退、消失或复兴的动态时间进程中”加以理解，并将“过去……视为设计知识与经验的宝库” [131, 第4页-11]。我们希望，本研究能为未来的人机交互研究提供启示，探讨如何支持人类定义自己的抽象和形式体系，而不是受限于设计者的定义，从而创造出更具“可塑性”、能够适应用户需求的界面 [83, 102, 143, 161]。我们首先将研究置于人机交互领域的背景下，并阐释“记法”等术语，随后开展比较历史分析（第3节）。接着，我们进行总结与反思（第4节），提出记法发展的功能阶段，最后在结论部分讨论一些启示与要点（第5—6节）。

2 背景与相关工作

2.1 人机交互及更广领域中的记法

对于“记法”这一术语，我们采用Blackwell与Green的包容性定义：“记法由在某种媒介上作出的标记构成（这些标记通常可见，但也可能通过其他方式被感知）” [20, 第 328]页。纸面上的书面语言、屏幕上的编程语言、代数方程、维恩图、有限自动机图表、表格、条形图、化学式、JSON——就我们当前的目的而言，这些都被视为记法。尽管一种记法可以仅由若干彼此之间不存在交互规则的孤立标记组成，但在此我们主要关注的是以类似于Goodman [58]的方式构成的书面记法系统：即具有句法和语义的标记系统，它们（试图）划定记法中的“有效”成员，从而使每一个成员都有一种解释。

记法在日常生活中以微小的方式不断被使用，例如，人们每当围绕白板或纸张协作、朝着共同目标努力时，就会用到记法。人们会随手画下 X、方框和箭头，代表正在探讨的概念。当人们发现某些记法有助于思考或协调时，便开始将这些记法“形式化”——明确系统的规则，以消除意义上的歧义并预见可能的误解 [34]。由此创造的记法随后可能超越最初的讨论，成为嵌入计算机系统，甚至启发新的探究，让某些想法比其他想法更容易被想象 [3, 20, 67, 76]。我们几乎所有与计算机相关的活动都涉及记法。

记法贯穿并嵌入于人机交互与软件开发的整个过程。例如，开发者会使用UML图 [115], 流程图 [3], 编程语言，以及“双钻”设计模型 [40]。界面设计实践在广义上要求为用户界面开发记法 [20, 121, 162], ，尤其是在设计计算媒体创作软件时。EuterPen [29] 使作曲家能够在排版形式与手写形式之间转换音符；该界面通过让手写音符更易于删除，将形式性具体化。一些系统工作研究探讨了基于笔的硬件与深度学习模型的新发展可能如何重塑编程实践 [3, 163]。记法作为界面，也会促成某些动作并抵制另一些动作 [20]。像Blender那样的可视化数据流环境支持可读性以及操作的串联，但会抵制需要迭代循环的操作。

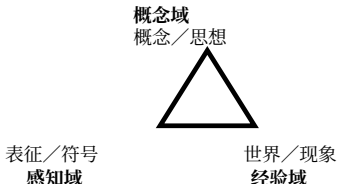
研究人员也会针对各种现有的记法实践开展研究，这些实践通常属于特定领域（例如，程序员如何绘制图表来交流想法 [34, 62, 63]）。在对软件团队的研究中，Cherubini 等人 [34] 发现，人们“倾向于采用非正式、临时的记法”，而且“对任何标准的遵循都有限”。他们展示了，随着对话从白板上的内部临时讨论扩展到与客户和高层管理者的沟通，记法也会以类似的方式逐步系统化：从临时、非正式、短暂的草图，发展为“反复完善”的草图；当草图“重要到值得投入时间和精力，将其转化为更持久的形式”时，就会成为“绘制图” [34, 563-4]。研究人员对个别记法开展了丰富的历史研究（例如，[3, 70, 74, 76, 115]），但对于新记法得以创建和形式化的更普遍机制与模式，我们仍知之甚少。

许多记法是通过文化习得并代代相传的。当我们在人机交互领域谈论“将”某个设计构想“形式化”时，往往会不自觉地将其理解为把非正式想法转化为现有的“形式的”表征：例如，从自然语言转化为编程语言 [124–126]。Shipman和McCall在关于渐进式形式化的论文中，主要采用的正是这种含义 [124, 125]。“记法”一词在日常用法中通常也被理解为一种已经形式化的书写记录系统：数学记法、分子式等。然而，如今看似显而易见的记法，其历史往往相对短暂：例如，图表中的箭头大约在18世纪才出现。对学者来说似乎“显而易见”的记法，对其他人而言也未必如此：例如，美国和德国人口中，约有三分之一的人阅读数据可视化的能力较低 [51]。即使人们无法读懂它们，形式化符号系统往往仍发挥着修辞目的——营造“理性印象” [90, p. 1]，并通过可信度传递可信赖性与权威性。

为了更全面地理解记法，我们关注它们如何在功能或技术属性之外被社会建构。符号学和认知科学将为我们提供工具，帮助我们剖析记法，以及我们所说的“形式”和“形式化”是什么意思。

2.2 关于记法与形式性的符号学、认知与社会视角

从符号学和科学技术研究（STS）的视角来看，记法是一种抽象，具有一个用于对某些经验性、可观察现象进行建模的表征系统。记法将现象带入“实验室”，在那里，现象可以被拆解、操纵、读取、呈现、比较和组合 [80]。优秀的记法系统在使用过程中会生成理论，这些理论能够以可预测的方式映射到世界状态。例如，在科学中，记法会提出可通过经验验证的实验；在音乐中，记法协调演出的可复现性（各个维度）。因此，采用经典符号学的三元框架[1] 颇有助益；该框架将符号、思想和指称对象区分开来，这三者构成了嵌入社会生活中的记法系统的基础：



从对世界的经验（现象 W）出发，我们形成概念与理论（C）。日常概念借助口头或书面语言得到表征（R）。复杂现象的系统可以催生书写系统，帮助我们从认知上应对该系统的复杂性、记住情境，并计算或预判物理过程的结果。这些“思维工具” [67] 或“纸面工具” [76] 可以激发新的探究领域，促使人们动员并组装物理世界中的实体，以“施行”或“检验”由页面上特定线条排列所创造的“理论”。因此，起初从经验领域到表征的动作（W→R）可以反作用于世界（R→W）；同样，表征也可以通过反思，经由抽象激发新的概念，例如将“外观相似”的记法归入同一个类（R→C）。尽管这是一种简化的抽象，但将 W、C 与 R 分开，有助于我们澄清某一层面的发展——例如对世界现象的概念理解——如何影响另一层面。

从认知科学的视角来看，记法是一种指向抽象现象的具身隐喻。Lakoff & Núñez 认为，数学符号系统源于对概念隐喻的反复运用：“一种神经机制，使我们能够利用一个概念域（例如几何学）的推理结构来对另一个概念域（例如算术）进行推理” [79, p. 6]。数学中的隐喻起初是奠基隐喻，因为它们将抽象奠基于日常（可把握的）人类经验之中，例如婴儿在大约一岁时习得的计数、数量以及内外关系等概念。抽象一旦形成，随后便可用来启发新的抽象——一种连接 隐喻——以类比方式逐层构建抽象，从而形成我们今天所知的数学系统。例如，布尔的集合论源于试图将集合与算术的结合律、交换律等定律进行类比对齐——简直就是要

[1]此处不便全面讨论符号学，但可参考 Ogden & Richard 的指称三角 [110] 和 Pierce 的符号／对象／解释项三元组 [166]。

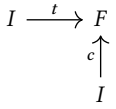


图 1：“形式化方格”揭示了形式化中隐藏的社会维度。非正式（*I*）通过“水平”动作（翻译 *t* 为既有形式）和“垂直”动作（创建 *c* 新形式）转变为形式化（*F*），这代表了两种形式化过程。在实践中，跨越形式化方格的动作往往会混合进行（采用既有记法，并与新记法混用）。

想象，“如果集合就是算术，会怎样？”——然后从认知上推演其意涵，并据此修正记号系统 [79,p. 142]。尽管 Lakoff & Núñez 仅考察了数学符号系统，但他们关于概念隐喻的观念将被证明有助于更广泛地理解记法。

最后，对我们的分析而言，重要的是批判性地审视“形式性”一词；在计算领域中，人们几乎总是不加解释地使用它。我们在附录 A 中进行了词源分析，揭示了形式性所具有的社会维度，例如它早期与法律和仪式相关的含义，以及后来与意义丧失相关的含义。归根结底，形式化主要通过两种“动作”实现，而这两种动作并非必然相互排斥：翻译为既有、成熟的形式，以及创建新形式，这些形式通过社会过程获得权力和公信力（图 1）。

2.3 关于可用性、定制与权力的相关研究

我们的研究承接了 Green、Petre、Blackwell 等人 [20] 关于将界面视为认知技术时其可用性的早期研究。他们提出的记谱法的认知维度 [20] 框架提供了一套可用性启发式，用于分析现有和拟议中的图形界面。这些界面的定义，既取决于其特定的预定义记法 [59]，也取决于其“环境”和“媒介”。我们后来对领域与（表征性）记法之间映射的观察，补充了认知维度中关于“映射接近性”的讨论。在这里，我们聚焦于记法发展的过程——记法的历史创制、演化与集体完善。

近期关于图形用户界面定制的研究，让用户无需编写代码便拥有了前所未有的灵活性，但在支持创建全新记法方面仍有不足。例如，DynaVis [144] 根据用户的自然语言命令，从预定义元素中动态生成定制的图形用户界面控件；Meridian [103] 则支持可塑总览－细节界面的设计与实现，用户可以定制属性在不同视图中的呈现方式。借助大语言模型等技术，可塑界面甚至能够根据用户查询生成新属性并将其纳入视图 [102], , 但用户定义这些属性并对其进行计算时所采用的交互方式，仍受限于现有的自然语言和编程语言。

近期，Li 等人对创意支持工具的分析 [84]提供了经验证据，表明系统的预定义抽象和记法正在固化系统设计者与用户之间持续存在的权力失衡——无论这些抽象和记法设计得多么周全。本文探讨人们如何设计自己的表达语言，这一过程提供了相关实例，有望为未来的系统设计者提供指引，帮助他们通过支持用户自定义抽象和用户自定义记法来缓解这种权力失衡。

3 记法发展的阶段与模式

为了更好地理解新记法系统的形式化过程，我们对不同记法的发展进行了比较历史分析；这些记法各自都已在既有文献中有所记录。具体而言，我们开展了平行比较史，这种方法“首先力求证明某一理论同样适用于不同案例..... [并且不同] 案例之间的差异主要是情境性的特殊性，而正是通过这些特殊性来凸显被 [理论化的] 过程的普遍性” [129, 第 178] 页。我们的方法力求在“ [案例] 选择中实现广泛覆盖” [129, 第 179] 页，并通过反复比较和对照这些案例，归纳出一般模式；这一方法遵循比较历史研究中的归纳迭代 [164], ，其“循环逻辑是在数据与理论之间来回移动，研究者据此构建最为可信的决定论解释，以说明所有案例中观察到的结果” [164, 第 618] 页。这是一种 [39] 历史、人文学科和定性研究中常见的解释主义方法论，并且与科学技术研究中“使用历史案例研究”来形成理论的实践相契合 [131, 第 6], 页，例如 Pinch 和 Bijker 对自行车史的比较分析 [114]。

由于特定记法的历史往往缺少围绕初始创造过程的详细、直接观察，我们通过偶尔引用实验符号学、传播理论与认知科学中基于实验的文献，来补充这一“宏”分析；这些文献探讨人们如何使用记法为沟通奠定基础，且大多来自实验室研究。我们在此聚焦于那些已具备明确特征与目的的记法。以下问题指导了我们的综述：

> （1）诱发因素：记法为何出现？是什么促使或要求创造一种新记法？（2）创造过程：记法最初是如何被创造的？是否存在共同的步骤或技巧？（3）演化：记法要经历哪些阶段，才会被视为已充分“形式化”？（4）形式化目的：形式化在何时、为何有帮助，又在何时没有帮助？

我们的初步分析以五个学科中的多种记法为起点——音乐、舞蹈、化学、物理学和计算机编程；这些领域都有既有的历史文献可供参考。[2]

随着研究的推进以及与他人的讨论，我们逐步扩展到其他记法领域，例如手语书写系统、统一建模语言图表、Markdown 和杂技抛接记谱法，并通过迭代的方式，围绕主题达到饱和并发现任何遗漏的主题。[3] 附录B和表1列出了所有纳入考量的记法。我们首先参考了针对特定记法发展历程的既有详细历史分析（例如， [70, 76]），并根据这些历史资料查阅一手资料，以厘清其最初的表示形式，例如刘易斯结构式 [82]和布奇早期软件设计记法 [21]。

我们的综述在记法发展过程中发现了许多经验性模式。我们逐一陈述每种模式，简要描述并提供示例。少数模式需要更详细的阐释；对于这些案例，我们会联系相关研究。这些模式并非面面俱到——正如 CDs 框架经过数十年才得以完善 [20], 我们的工作只是初步探索，而非定论。为梳理这些模式并便于阐述，我们将它们归入记法发展的三个总体社会阶段。尽管记法的用途和使用方式差异很大，但不同历史轨迹中的各阶段却惊人地相似。记号系统的社会发展阶段如下：

- 阶段 1：发明与孕育
- 第二阶段：扩散与分化
- 阶段 3：制度化与神圣化

这些阶段构成一个连续谱，并非僵化的界限。为便于呈现，我们将各模式归入最相关的阶段；不过，这些模式也可能适用于多个阶段。图 2 展示了一种记法——萨顿手语书写系统——的演化，追溯其与乐谱记谱法的渊源，并将其与正文中的模式联系起来。

为了帮助读者浏览这些信息，我们还将这些模式归类，归纳出四个主要类别：

(1) 记法的灵感来源、设计与演化的认知因素。(2) 社会公信力与价值评估：围绕记法的采纳、传播与维护，记法如何获得价值或赋予价值的社会因素。(3) 形式化过程：记法如何得到改进、消除歧义并扩展，以及与之相关的过程。(4) 约束：制约记法演化方向的物质或文化因素。

我们添加了彩色徽标以提高可读性。与阶段一样，这些类别并非严格界定，而是作为视觉辅助，例如，认知因素当然也具有文化维度。

认知模式将要求我们从具身数学 [79], 变异理论 [25, 93, 94], 以及认知科学中的结构映射理论 [53]中剖析相关概念：连接与奠基的概念隐喻、类比对齐所发挥的突出作用，以及记法如何呈现并将概念/世界领域中有意义变异的维度映射到表征域中有意义的知觉 变异维度。

[2]计算系统编码并呈现许多记法，用户来自各行各业，而不仅是技术背景——因此，与未来系统共同发明的记法可能服务于多种目的。因此，我们不希望人为地将自己局限于仅与人机交互相邻的记法。

[3]当观察更多案例后，几乎不再出现新模式时，即可认为达到了饱和 [31]。例如，我们在研究后期对量子电路记法进行的考察，只是证实了已有模式：关联与锚定隐喻、影响设计的文化背景，以及常规化。

在本节中，我们出于学术评论的目的，在合理使用范围内呈现原始受版权保护材料的快照；在每种情况下，我们都会引用原始来源。

3.1 发明与孕育（阶段 1）

3.1.1 认**是为了管理复杂性，并协调行动的可复现性**。什么样的情境会促使人们创造一种新的记法？最主要的原因包括：使复杂问题易于处理、储存信息、便于比较、高效沟通、在心算时记住术语、减少书写努力，以及协调跨越时空的可复现行动。这些原因凸显了记法作为助记工具、存储工具、比较工具、计算工具、速记工具、协调工具和框架工具的用途。

例如，维恩发明了以自己名字命名的图表，将其用作记账工具，帮助“任何希望有效应对逻辑表达式中数学术语组合爆炸的人” [146, p. 7]。同样，费曼发明这些图表，“作为一种简化量子电动力学中冗长计算的记账工具” [70, p. 4]。音乐和舞蹈等其他记法旨在储存跨越时空的精确动作，使后世能够呈现结构相似的表演。Sutton 发明了 DanceWriting，因为他担心一种“古老而专门的 [丹麦] 芭蕾舞训练形式……可能会失传或被遗忘”，并强调效率、减少努力、易读性，以及非艺术使用者的易学性 [134, p. 1-2]。最后，普莱费尔发明了折线图绘制，使得“在五分钟内获得的信息，足以媲美用一张数字表格花费整整几天才能刻入记忆的信息……” [116, p. xii]。

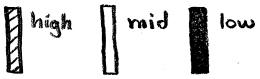
3.1.2 一**问题领域可能促使人们独立开发用于解决该问题的记法**。由于记法的出现是为了管理复杂性并协调行动，反复出现的复杂性管理需求（以及科学技术进步）可能促使人们独立地、有时甚至同时发明用于同一任务的记法。例如，牛顿和莱布尼茨的微积分符号；费曼-戴森图与大致同一时期在日本由 Koba 与 Takeda 为相同目的开发的“跃迁图” [70]；以及道尔顿的原子图和贝采利乌斯的化学式 [76, 第 40]页。在软件工程领域，“到 20 世纪 80 年代末，已有五十多种彼此独立的建模语言——每种语言都有自己的语法、结构和记法”，这导致“项目失败率高得令人担忧” [115, 第 348-9] 页，并由此促成了“统一”这些语言的努力（统一建模语言）。

3.1.3 调整**其周遭环境和背景中随手可用的常见表征**。发明者受到自身文化、学科和个人背景的影响，并会调整自己已经熟悉的表征和审美。例如，Arawjo 提到，戈德斯廷与冯·诺伊曼的流程图方法，灵感来自他们开展项目的摩尔电气工程学院常见的电气原理框图 [3]。在物理学中，彭罗斯图并非只是改编自闵可夫斯基时空图，也受到 M.C. 埃舍尔艺术的启发 [159]。这种相似之处不只是视觉上的——两者的概念也很容易对应起来。埃舍尔的画作呈现了“不可能物体”，并通过扭曲透视来表现无穷；而彭罗斯描绘广义相对论中时空共形变换的图，也以类似方式“扭曲了页面上距离之间的关系，使‘无穷’能够表示为有限值” [159, 第 137] 页。[4]

3.1.4 认**凸显问题领域中发明者认为最有意义的变异维度，使人们更容易读取和编码最能体现这些维度的细节，却更难甚至无法捕捉超出这些维度的细节**。记法中的显著特征对应着（发明者认为的）建模领域中最重要的特征；被省略或隐含的元素则不那么关键，或被视为无关紧要。当关键变异维度被省略时，就必须扩展记法或重新构想它。变异理论 [25] 指出，如果纳入无关的变异维度，这些维度可能会捕捉到所表征概念之间偶然的相似与差异，从而分散用户的注意力；也可能因其一致性而误导用户，使他们误以为该维度上的某个特定固定值至关重要。

随着时间推移，不变维度通常会被抽象化或省略。例如，在萨顿手语书写系统中，身体部位的位置是关键的变异点，并以简笔人像表示。然而，使用 SignWriting 的聋人成年人“[强烈]认为 [简]笔人像并非必需”，他们反对一种“全身”版的 SignWriting：无论面部表情是否具有意义，这种版本总会用一个圆圈表示头部 [136, 第 42]页。肩部和胸部线条被删去，只有当头部与符号的意义相关时才会显示（图 2）。[5] 这与克拉克与布伦南在交际基础建立中提出的“最省力原则”一致：“为了让受众理解你的意思，只需付出必要的努力，不要再多费力气 [就可以了]” [36, 第 226]页。通过免去书写不变（或通常不变）特征的需要，人们可以减少体力消耗；这可能是图像表征变得更抽象的一种机制 [45]。

变化维度可以用不同精细度的方式编码。以拉班舞谱为例：动作的“程度”由三种形状填充模式（空白、填充或条纹）表示，分别对应地面上、与地面平行和高处的位置 [77]：



这些离散方向一目了然，但无法指定诸如“15度”这样的精确角度。从该记法的成功可以推断，对于20世纪欧洲舞蹈编排的大多数用途而言，这种不精确性并未构成问题。然而，如果要创作一份需要

[4]根据结构映射理论，如果相似之处并不对应可对齐的概念，这种交叉映射就会妨碍而非帮助人们通过新记法进行认知 [54]。[5]有趣的是，有聋童的健听父母更倾向于选择这一变体用于学习，这或许是因为它更明确地呈现了与身体之间的映射；这种映射对有经验的手语使用者而言是不言自明的，因此没有必要特别说明。结构映射理论表明，用户的既有知识会影响他们比较和区分哪些不同的符号 [54]。

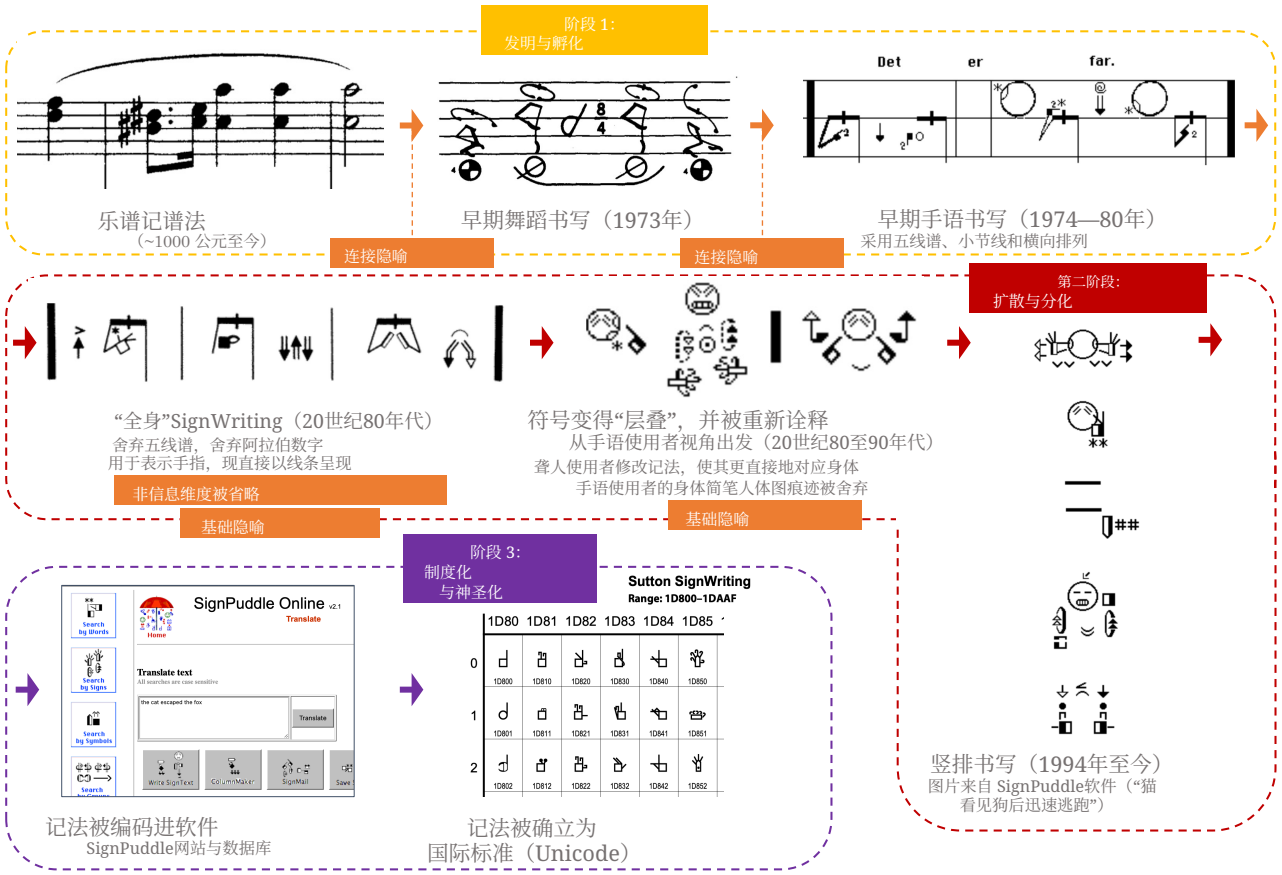


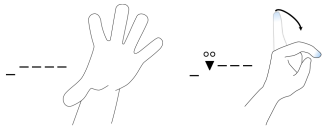
图 2：*SignWriting*的演化：这是一种书写手语的记法，最初由 Sutton 发明，并在数十年间与聋人社群及学术研究者共同设计而成。追溯其源自乐谱记谱法的脉络，我们可以看到连接隐喻和奠基隐喻的广泛运用 [79]，其中也包括打破先前记法基本要素的新隐喻：五线谱逐渐消失；释义方式发生了巨大变化，却没有影响图符；书写方式则从横排转为竖列。SignWriting 在软件和教育资源中实现了物质与社会层面的制度化，并被编码纳入 Unicode 标准。如今，任何变更都需要耗费大量成本重新印刷、重新编码，并动员委员会。图片为学术分析之目的，依据合理使用原则转载自 Sutton [134, 136]。

精确角度的舞谱——例如，使舞者的双腿与时钟刻度对齐——拉班舞谱将难以表现这一动作。

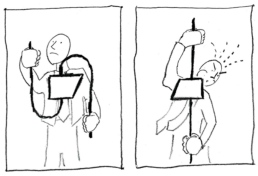
3.1.5 认[illegible]**用先天和文化形成的感知通道，并赋予它们更多意义。**记法将意义映射到感知通道。这些通道可以是先天的认知或习得的文化感知图式。先天通道是人类视空间认知的生物学可供性，包括实体层面的特征，如颜色、形状、大小、方向和纹理，以及关系层面（格式塔）特征，如“接触”、连接、接近性、排列、数量、顺序、叠加、包含（内外概念）和轨迹 [79]。习得的文化通道包括拉丁字母和阿拉伯数字，它们构成有界的顺序和符号集合。在记法中使用时，通道往往呈现数字式（而非模拟式）语义——某个符号要么在圆圈内，要么不在；实体要么相接触，要么不接触；要么 x 位于的左侧或右侧。这体现了Goodman的洞见：记法系统依赖由符号之间（隐含或明确的）边界所确立的数字式解读 [58]——例如，我们不会把“o”和“σ”看作连续谱上的点，而是看作彼此独立的实体。感知通道这一概念与数据可视化中对标记和视觉通道的分类有关 [105]。第二阶段的模式涵盖了记法发展过程中如何利用和扩展通道。

3.1.6 [illegible]**喻。**具身感知经验在记法设计以及评判其可用性方面都发挥着关键作用。奠基隐喻 [79] 可分为不同类型，从空间隐喻，例如微手势记法如何映射手指的位置并在视觉上与之相似 [30],，到具身隐喻，即能够引发操作的感知可供性 [52]。关于空间上的“映射接近性”（《记法的认知维度》 [20]），可以看看微手势记法是如何

设计成类似于拇指与其他手指分开的方式——最左侧的手指处于“较低”位置 [30]:



以具身隐喻为例，看看 Coecke 和 Kissinger 的量子计算记法 [37], 。他们扩展了先前的字符串图记法，该记法用线上的方框表示函数。发明者切去函数框的一个角，以便人们能感知形状的旋转；这样一来，“我们就能像拨弄项链上的珠子一样，在线上滑动方框” [37, 第 109] 页：



这种记法在视觉上酷似串在线上的珠子，自然让人想到滑动和拉扯；其创作者借此将形状旋转赋予了“对函数进行转置”的意义 [37, 第 109] 页。

在构成神经网络基础的 McColloch 和 Pitt 感知器研究中，作者在推导大脑活动的逻辑演算 [100] 时，绘制了看起来像显微镜下神经元的图表。后来，Kleene 采用了他们的“网”记法，证明形式自动机与正则表达式等价 [75]。

费曼图则是一个相反的例子：视觉上的相似性可能会误导用户——就界面设计而言，这是一种奠基隐喻失配 隐喻失配 [107], ，也称为跨域映射 [54]。费曼图源自与闵可夫斯基时空图相连接的隐喻，但这种做法起初并不受欢迎：“玻尔一再反对这样一种观点，即时空图能对研究……本质上不可视化的量子现象有所帮助” [70, 第 47] 页。[6] 如今，这种记法与粒子散射轨迹的相似性仍让“从事研究的物理学家和教师……忙于阻止同事和学生根据这种相似性作出错误解读” [160, 第 10] 页。

3.1.7 认 **喻。** 第二种主要的概念隐喻类型 [79] 是连接隐喻，即从一个隐喻到已有隐喻的映射中产生新的抽象。在发明新记法的过程中，连接隐喻极为常见。连接隐喻的例子包括源自代数记法的分子公式、源自音乐记谱法的舞蹈记谱法、源自闵可夫斯基时空图的费曼图，以及源自电气原理图中框图、用于 ENIAC 计算机的流程图绘制 [3]。一个系统的概念属性（不仅仅是它的表征）会被应用于新的任务领域，从而生成候选推论 [53]。

连接隐喻可以帮助熟悉源记法的用户学习新的记法。例如，看看普莱费尔如何引入所谓的“图表”来进行数据可视化——这个词本身原本指用于导航的地图 [111]。普莱费尔的图表抽象了地理地图中的距离度量标准：如今，南北和东西方向代表的不再是距离，而是年份和财富。普莱费尔明确提到了灵感来源：“对于学过地理学或数学任何分支的人来说，这些图表都将十分易懂” [116, ix]。普莱费尔指出，一位法国贵族将他的书呈献给路易十六，“国王熟悉地理学，因此很快就理解了这本书，并表示极大的满意”——普莱费尔认为，这种联系随后帮助他获得了在法国传播其作品的法律特权。

连接隐喻可以串联起来，也就是说，记法可以相互建立在彼此的基础之上（图 2）。萨顿手语书写系统“源自 DanceWriting” [136], ，这是萨顿创制的一种舞蹈记谱法，灵感来自佐恩早期创制的一种舞蹈记谱法 [135] ；后者借用了乐谱记谱法的连接隐喻，将简笔人像叠加在五线谱记法之上。

3.1.8 认 **新的隐喻，记法可能需要打破与过去隐喻的对应关系。** 关联与锚定隐喻都依赖于类比对齐这一认知过程 [53]：“在两个概念之间建立结构化对齐的过程”，其中“既涉及对齐，也涉及投射”。在投射过程中，基础概念会向目标领域引入某些信息。在对齐过程中，尝试对齐两个概念域，需要找出它们共有的结构，并可能“揭示出此前在任一”领域中都不明显的共同点 [53, 第 673 页-4]。

如图 2 所示，关联与锚定隐喻可以推动记法的演化。然而，引入新的关联与锚定隐喻，可能需要记法放宽或打破与既有隐喻之间的类比对齐。失去既有对齐所带来的“连贯性和推理能力” [53] ，使人们能够借助新的关联与锚定隐喻，探索实现连贯性和推理能力的新可能。

以化学式为例。贝采利乌斯试图将代数记法应用于化学 [76, p. 10], ，发明了 $CuO + SO_3$ 这样的化学式（最初写作 $CuO + S^3O$ ）——由此形成了与代数的连接隐喻（即贝采利乌斯的想法：化学就是数学）。在类比对齐过程中，代数记号系统的某些属性很好地映射到了新情境中，例如交换律 + 以及用括号将运算分组的好处；而另一些属性，如 × 和结合律，则在语义上失效了，或是遭到违背，或是失去了具体意义。用大写字母表示化学元素的简写方式，是另一种连接隐喻和设计选择：道尔顿的替代记法并未使用字母，而是将记法奠基于一种视觉隐喻，即把原子视为不可分割的球体，并用大小相同的圆形来表示 [76, p. 35-40]。

引入新的奠基隐喻，可能会打破连接隐喻的某些方面（ 3.1.7）。拉班舞谱最初源自与乐谱记谱法的连接隐喻，但当一位共同发明者建议将水平谱表旋转为

[6] 物理学家担心，图表在视觉上与离散粒子相似，可能会误导人们作出费曼所否定的解读 [160, p. 6]。然而，耐人寻味的是，戴森在将这些图表引入物理学时，采用了基础隐喻：“在费曼理论中，与某个特定矩阵元相对应的图，不仅被视为一种计算辅助工具，还被视为引发该矩阵元的物理过程的图像” [41, p. 496；着重号为后加]。

垂直方向时，它迎来了一个“突破”：“与其横向书写一系列十字……不如将竖直排列的列用于标示身体部位，这样动作符号就可以连续书写” [155, 页 278]。因此，这种记法能够更贴近身体从左到右的方向排列（例如，将左臂手势放在最左侧的列中）。但五条谱线对应的五个频道仍被保留为五列——这是乐谱记谱法遗留下来的痕迹。SignWriting 也经历了类似的演变（图 2）。Sutton 展示了五线谱和小节线如何逐渐淡出：首先，谱表从五条线变成三条线，随后彻底舍弃（“谁还需要那些多余的线条？” [136, 页 34]）。后来，随着聋人社群越来越多地使用这种记法，“他们自然而然地将符号[竖直] 排列起来，使其看起来像人类的身体”，因为这种排法“模仿了现实生活中手语的样子” [136, 页 37]——由此运用了新的奠基 隐喻，关注映射的空间接近性，从而摆脱了人为的线性排列。最后改用竖列书写，彻底消除了乐谱记谱法遗留下来的唯一句法残留。

类比投射过程可能会导致目标领域中独有的维度——这些维度在基域中没有对应项——被舍弃或暂时遗忘，即作为不可对齐的差异，其认知显著性较低 [54]。例如，在舞蹈中，一支舞蹈作品中多位舞者的位置安排可能十分重要，但拉班舞谱继承了乐谱记谱法的特点，关注的是脱离空间的单个舞者的表演。

3.1.9 认 **过快速迭代，将记法应用于多种情境，以增强其稳健性**。 一些早期发展历程，例如费曼、Gruber（Markdown）、拉班和 Sutton 的经历，都类似于一种孕育过程，与迭代设计相仿。例如，拉班和他的同事们辛勤工作了约十年，之后拉班才“向热情的支持者展示[了] 他的系统”，以便对其进行推广、筹资和传播 [155, 第 279]页。现代记法（如编程 API）由发明者或小型社群通过快速应用和迭代，在多种使用情境中不断孕育发展。然而，并非所有记法都有漫长的孕育期：例如，数学符号是随着时间推移，通过不断提出并筛选渐进式建议而形成的 [26]。

3.1.10 约 **束的能动性**。 记法借助各种媒介得以表达：钢笔、铅笔和纸张；键盘、鼠标和屏幕；白板、手写笔和板擦。正如先前工作中所讨论的 [3, 20]，这些材料约束在记法的初期发展中起着决定性和约束性作用，限定了哪些表征最容易表达，哪些则受到抵制。例如，FORTRAN 语言是围绕当时 IBM 商用键盘的局限而构建的 [3]；若采用配备下标和上标换档键的 Flexowriter，数组索引的访问方式或许就不会是 `a[i]`，而是 a_i。

延迟和交互也很重要。共同创造共享语言的人们，如果能够同时互动，而不是轮流进行，往往会趋向于采用更抽象的形式：“即使只有最低限度的图形交互（例如用一个勾号表示理解），合作伙伴的图画也会逐渐趋同，并从图像式符号发展为任意符号……如果没有这种交互，图画则仍停留在图像式的状态” [50, 第 484 页-5]。有证据表明，比较会促使关系抽象，这也印证了这一点；即使依次看到两样事物，若未被提示进行比较，成年人也会忽略它们共有的抽象特征 [53]。

3.1.11 **既定记法发生非预期的符号冲突**。 另一些时候，人们做出决策是为了预先应对与其他既有记法系统的句法和符号之间的文化“冲突”；你可以称之为“文化交叉映射”。如果一种新记法看起来与另一种广为人知的记法过于相似（至少在目标用户中广为人知），就可能造成解读上的混淆：习惯于另一种记法的读者可能会系统性地误读这一新记法。例如，Church 曾暗示，他使用 λ 符号，源于预先考虑到会与其他意义饱和的希腊字母发生冲突；当被问及为何选择 λ 而非其他据推测较少使用的希腊字母时，他答道：“eeny, meeny, miny, moe” [123]。

3.2 扩散与分化（阶段2）

在发明与孕育期之后，一种记法可能会扩散到其他社群和用户，由他们重新诠释、修正并据为己有。从历史上看，这通常紧随以书籍、学术期刊或大众媒介发布为形式的首次发表之后。当用户在各自独特的情境中将记法应用于不同情形时，随着歧义通过本地辩论或公开争议得到解决，记法便会与原初形式发生分化。若针对同一情形发明了多种记法，竞争者争夺关注与用户时，便可能出现“战争”。

3.2.1 社 **采用者推广并传播该记法的触发事件**。 与一般创新一样 [120], 早期采用者对于推广、传播并使记法形式化至关重要。以Kaiser所述费曼图的传播为例 [70]。对费曼而言，他最初的记法是非正式的：记法规则，例如所有顶点都恰好需要三条边，并未明确写下来，而且他只打算自己将该记法用作计算工具。实际上是弗里曼·戴森完善了该理论并率先发表 [70, 第175页-195]。

3.2.2 **着新的社群开始使用并掌控记法，根据自身独特的情境对其进行扩充、解读和应用，记法便会发生分化**。 一旦记法离开其最初的发明者或共同体，在解读上的歧义以及将其应用于新情境或未曾遇到的情境，都可能使记法偏离原始形式。这些偏离可能体现在表征上，也可能体现在解读上。记法实践社群彼此隔绝的情况（例如战争期间）可能导致分化，而且这种分化或许永远无法弥合 [155]。

支撑分化的，可能是信仰体系之间的争议。例如，Kaiser 叙述了费曼图许多受地域限制的分化，并将另类信念追溯为原因之一 [70]。

分化似乎是一种自然过程 [50] ，并不一定是负面的。有些记法，例如彭罗斯图 [159]，主要具有教育性质，可以保持情境性和启发性。然而，有时分化也会引发争议。Gruber 关于 Markdown 的原始帖子“没有明确规定句法

”，这导致在实现解析器时出现不同的解读：“目前解决 Markdown 歧义的最佳方法……是 Babelmark 3，它会将 20+ 个 Markdown 实现的输出相互比较，以判断能否形成共识” [38]。另一个例子是拉班舞谱——拉班使用的是术语动觉记谱法，而拉班舞谱则是由 Guest 创造的；他在英国和美国对这种记法进行了改编。根据一个人所遵循的舞蹈记谱法流派，相同的视觉表征可能会有不同的读法 [155, 第 284]页。

3.2.3 **所需的形式化程度取决于需求和情境。** 记法能够达到多大程度的明确性，取决于记法的目的，以及形式化对实现该目的有多大帮助。对于被认为不太重要或无关紧要的变异维度，记法可以有意保留模糊性。例如，在音乐、舞蹈和戏剧中，音符仅仅充当脚本，但有些通道，如音量和音色，可能不会被明确编码，而是留给表演者自行诠释 [58]。对于 Markdown，Gruber 认为其成功“得益于其缺乏标准化，而非尽管缺乏标准化仍取得成功” [12]。相较之下，计算机代码不能容许任何出错的余地，因为逐步构建系统的需求 要求严格的确定性。因此，编程语言社群会制定规范，以协调“正确”的执行（实现）。

3.2.4 **展：在现有感知通道内对 记法的扩展。** 每种记法都会利用人类视觉和具身认知的感知通道（3.1.5）。通道是一种具有具体取值的变化维度——通常是映射到意义上的类别。记法可以通过扩展现有通道中的类别而演变。例如，闵可夫斯基时空图使用直线表示轨迹。彭罗斯提出的一种扩展，在同一通道中引入了“波浪”线，用来表示运动中的“观察者和粒子……打破了光速物体沿着 45 [度]运动的惯例” [158,页。 114]。突然之间，线条的波浪形 ~ 有了意义。费曼图也通过对轨迹进行类似的类别扩展而得到扩展，其中胶子以弹簧状的曲线表示。

这里的关键在于这样一种观点：记法创建了类别通道，之后还可以对其进行扩展。例如，莱布尼茨为求和和积分分别引入了 $\int$ [26,第 II 卷第 36 页]，，利用了一个未使用的大小通道（该符号比周围的方程符号更高） [44]。后来欧拉将该符号改为 $\sum$，柯西又将其扩展为 [26, 第 II 卷第 61 页] 垂直堆叠通道（ n 和 i 的句法分别置于其上方和下方）：

$$\sum_{i=1}^{n} x_i$$

通过进行类别扩展，高斯利用如今已经确立的 $\sum$ 通道引入了乘积符号 $\prod$，并在这一过程中将循环运算的概念推广到了乘法。

3.2.5 **展：将记法扩展到未使用的感知通道。** 人们开始使用记法后，会扩展或调整记法，以适应自身独特的使用情境。然而，记法在社会上越受欢迎、越趋于“定型”，人们就越会预期，在修改现有句法以及“形状”和“大小”等感知通道时遇到更大的阻力。修改现有感知通道可能会带来难以预见、逐渐显现的不一致或冲突（这与软件开发者不愿进行“破坏性变更” [23]的现象类似）。为降低修改现有通道的风险，记法的使用者会进行通道扩展——寻找此前未使用或利用不足的感知通道，并赋予它们丰富的意义。

例如，在白板上，人们可以用单一颜色勾勒用户界面，但随后可能意识到还需要表示布局约束，于是可以使用第二种颜色来减少混淆；这样，第二种颜色就传达了“布局约束”，而第一种颜色传达了“界面元素”。引入第二种颜色创造了一个新的变化维度 [94]，并且隐含地赋予颜色以意义。

另请考虑贝采利乌斯的分子记法。在后期的著作中，贝采利乌斯开始在简化式上划线，这些简化式“表示该化合物的两个‘原子’” [76, 第 176 页-7]——也就是说，拓展到一个未使用的感知通道，形成连续与重叠的格式塔。他还用元素上方的点来表示氧 [16, 第 623] 页：

$$\dot{\mathrm{E}}\dot{\mathrm{N}} + \dot{\mathrm{H}} = \text{Salpeteräther}$$

后来，化学家将贝采利乌斯的公式改造为图记法，以展现结构，即现代的“结构式”。分子记法只记录元素的比例，而结构式则规定原子如何组织和连接，其中包括用线条的数量表示共价键的“强度”（即共享电子）等特征。刘易斯 [82] 引入了点式记法，将点置于贝采利乌斯化学符号的旁边，以表示化学键和价电子 [82, 第 777] 页：

> We may go further and give a complete formula for each compound by using the symbol of the kernel instead of the ordinary atomic symbol and by adjoining to each symbol a number of dots corresponding to the number of electrons in the atomic shell. Thus we may write $\mathbf{H:H}$, $\mathbf{H:\ddot{O}:H}$, $\mathbf{H:\ddot{I}:}$, $\mathbf{:\ddot{I}:\ddot{I}:}$, but we shall see that in many cases such a formula represents only one of the numerous extreme tautomeric forms.

（请注意，刘易斯在此如何改造了先前形成的句法——字母上方的点——剥离其原有语义并以新的语义取而代之。）感知通道可以成为可供性 [52]，，借助隐喻与具身或文化经验建立联系。在 Coecke 和 Kissinger 对字符串图记法的改编中（3.1.6），削去方框的一端，使观察者能够追踪 180度旋转，而此前他们做不到——为承载意义的形状方向这一未使用的感知通道注入了信息。这样，通道扩展便试图避免扰动记法既有的句法和语义。

3.2.6 **形式化操作：增补使此前已形式化（已确定）的句法变得非形式化（不稳定）。** 对记法进行临时扩展，能够使记法重新回到非形式性的领域——这是一种非形式化操作，并可能因此获得理论能力。一个例子是，贝采利乌斯在自己的化学记法已在科学界流行之后，开始对其进行扩展。在加入贯穿斜线等新元素（3.2.5）时，贝采利乌斯特意将此前已成为规范的（未经增补的）记法区分为“经验”公式——“它们直接源自正确的分析，并且

不可改变”——而他称自己的临时扩展为“理性”公式，认为这种公式“旨在提供一种概念”，并且可以根据化学家的“视角而变化”，但究竟哪个才是“真正的”公式却并不明确 [76, 第 177-8]页。因此，尽管所谓的“经验”公式仅具有描述性 功能，“理性”公式却力图成为生成性的公式——用以生成关于反应和化学组成的理论，而这些理论能否经受后续实验的检验尚未可知。贝采利乌斯并不确定，他的扩展在多大程度上有助于构建与经验证据相一致的理论。

3.2.7 认**义经过重新加工或舍弃，同时保留相似的表征**。有时，原有的 语义 可能会在保留相似表征的同时被彻底改造。例如，费曼图的语义已与其原有语义大相径庭，但仍然有用：“微扰理论失效后，人们或许会认为费曼图将不再有用……然而，许多理论学家并没有抛弃费曼图……而是开始以新颖、前所未有的方式利用这些图表……同时舍弃了使用它们的原有数学规则” [69, 第 56]页。SignWriting 是另一个例子：聋人使用者主张，记法应从自我中心视角（即手语者的视角）书写，而非采用外部视角（即观察者的视角）。句法和词形书写保持不变，但由于每个符号现在都代表其镜像，语义发生了分化 [136]。

3.2.8 **到无法表征的情形时，必须加以扩展、修订，或发展出一种次级记法**。当记法所依据的假设受到局限，遇到无法建模的情形（或者其显著特征会在该记法中被掩盖，或使读写变得繁琐）时，作者可能会选择绕开这种记法（例如在页边空白处书写，或发展一种次级记法 [20]）、修订或打破它（至少违背其一项核心原则），或者转而寻找一种全新的记法，以更好地捕捉那些被遮蔽或无法表征的信息。例如，情感无法编码进逻辑记法，因此作者会尝试“为这一新概念域发展一种用于形式化的表征语言” [89,第 815]页。

3. **通过翻译为已有的形式化代理记法来获得可信度并加速形式化**。 将一种记法映射到已获认可、广为确立的“形式”记法，往往能加速其获得形式化社会地位，并以某种方式“证明”新记法在能力上与之相似或等价。无论是通过计算方式（例如转译），还是通过修辞方式（例如在学术研究中证明不同记法之间“等价”），都可以通过翻译为既有记法，借助现有记法“引导”新记法实现形式化。图灵完备性的概念就是一个很好的例子：为了更轻松地证明一种新编程语言具有图灵完备性，人们会构造从该语言到图灵机记法或 Church 的映射 λ演算。在编程语言领域，治理委员会为语言管理制定的官方“规范”文档，例如 TS39 的 ECMAScript 规范，会通过翻译为其他记法（如形式文法）来实现形式化。

借鉴旧记法的新记法，也可能因此更快给人留下其形式特征足够鲜明的印象：Rust 是一种“形式的”编程语言，它借鉴了 C，因此更容易获得 C 和 C++ 程序员的信任并被他们采用。

3.2.10 社**性替代方案之间的记法之争，由可用性和社会权力所左右**。Klein 将为分子化学确定一种记法的过程描述为一场“战争” [76, 第 27] 页，获胜的符号得以留存，落败的符号则消亡，仿佛经历了一场自然选择。在数学领域，Cajori 展示了许多符号竞争的情形，例如莱布尼茨和牛顿的微积分符号之争 [26]。在这场“战争”中落败的记法会被 relegated to history，或虽继续沿用，但地位不如从前。通过有意推动统一和标准化，也可以平息这些纷争（阶段 3 开始）。

一种记法能否在这场竞争中“胜出”，不仅取决于易用性，也取决于权力。例如，数学中的等号 = 由 Recorde 于 1557年 引入，曾与笛卡尔提出的另一种符号相抗衡，后者看起来像是将 $\propto$ 水平翻转。数学家们因地理上与笛卡尔的接近程度不同，而各自采用了其中一种符号。Cajori 认为，最终选择 = 作为等号，是权力使然；具体而言，是莱布尼茨的声望使他有权作出裁定 [26,第 301–306 页，第一卷]。

同样，费曼和冯·诺依曼都是备受赞誉的科学家，他们的声望更容易影响其他人的选择。普林斯顿等实力雄厚的教育机构，是确保这种记法得以传播、延续并保持活力的关键力量，最终使其胜过 Koba 和 Takeda 的“跃迁图” [70]。

3.2.11 认**繁排列进行视觉简化和缩写**。随着记法的传播和常规使用，它们可能会得到简化：既可以在视觉上简化（形成更抽象、书写所需努力更少的形式），也可以通过简写来简化——即创造能够压缩信息和书写努力的新记法（后者取决于媒介）。例如，爱因斯坦求和记号，以及 + 符号如何从早期代数中反复书写“and”所需努力的压缩过程中产生 [26]。图 3 展示了托福利门记号在量子电路中如何通过逐步采用而得到简化。实验符号学表明，当人们共同在场时，抽象且简化的形式——就符号与其指称对象在视觉上的差异而言——最容易、最快地形成；随着使用者人数增加，这些形式会趋于一致，例如在紧密社群中第 486]。这表明，随着某一学科围绕一种记法组织起来，它可能会不断将该记法抽象化，以减少书写努力并消除不变性，但这同时也可能让新手更难掌握。

3.2.12 认**理解不断深化，推动了符号演化**。在原始记法的创建过程中，源域新的语义维度可能被发现（例如在科学领域），也可能最初被忽略（无论是由于文化特异性还是简化性观念），从而催生扩展或新的记法。例如，电子的发现促使 Lewis 将贝采利乌斯的分子记法扩展到价电子和共价键 [82]。SignWriting 通过承认手语中那些曾被边缘化的、有意义的差异维度，偏离了斯托基和汉堡符号体系

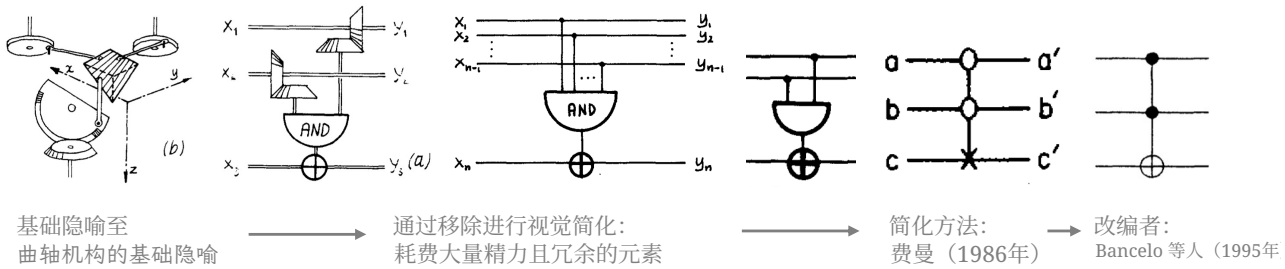


图 3：量子电路中托福利门记号的起源，揭示了托福利 [140], 所借鉴的曲轴机构基础隐喻；随后费曼对其进行了改编和简化 [48]，而后 Bancelo 等人又对其进行了改编 [9]。图片转载自原始文本 [9, 48, 140]，用于学术分析，属于合理使用。

：手语中存在一些曾被边缘化的差异：“这些系统基于一组符号，代表四个‘核心’参数……（或者至少是斯托基及其同事认为恰当的那些）…… [身体姿势和面部表情，以及视线的运用，尽管至关重要…… [却] 很少被考虑在内” [17, 第118页；着重号为后加]。

3.3 制度化与神圣化（阶段 3）

3.3.1 社 **系统开发者和用户会成立委员会和机构，以统一相互竞争的替代方案和／或规范权威解释与实践，从而遏制记法的分化**。随着记法的传播，其用法和解释自然会发生分化除非 建立社会和物质基础设施，以澄清并 执行 权威表征、解释和用法。因此，我们称一种记法变得“形式化”——得到扩展、完善、调整，并在审美上趋于稳定。采用者会形成社群，着手界定某种记法的恰当且有效的运用。形式化的目的在于确保 行动的可复现性 跨越时空：产生 明确无歧义的解释，从而引导确定性行动，而这些行动随后可以被 判定 为有效或无效的 运用 该记法的行为。[7] 由于解释是行动的中介，最终目标是实现行动协调，无论是人体运动还是机械动作。[8]

对于 Markdown，Gruber 原始规范的解读存在分歧，这促使高级用户齐聚一堂，创建了 *CommonMark*——“一种针对 Markdown 的标准、明确无歧义的句法规范，以及一套全面的测试，用于验证实现” [38]——也就是说，将各种记法统一为标准，再通过社会和物质手段加以贯彻。另一个例子是 Python 增强提案（PEPs），它们是“为 Python 提出重大新功能的主要机制” [154]。语言联盟类似于政府和立法过程，

[7]这里，我们受到 Goodman [58]对记法语境中表演一词用法的启发。[8]例如，数学方程式置于文化规则的背景下理解，便界定了一组有效的数学表演。对记法的使用，即对符号的“操作”，是在“规则”约束下进行的表演。有效的证明是可重复的表演。正确地进行表演是一种实践和社会层面的成就——因为新手可能会误用某种记法。

称自身为一种“正式的治理过程” [138]。即便数据可视化记法也有委员会的历史 [49]。

通过制度化来推动记法统一和标准化的努力可能会失败。ALGOL 是计算历史上的一个著名例子。最后几次会议之所以破裂，部分原因在于国际合作者分散在各地小组开展工作，且每次会议的与会者各不相同，时断时续地碰面，最终无法就一种记法达成一致 [87, 106]。

3.3.2 并 **式化的推进过程，是由社区成员提出修订和扩展建议、对其进行批评并提出有说服力的论据，再由用户以不同方式评判这些建议**。一旦某种记法得到广泛使用，包括发明者在内的社区成员就不能随意修改它；如果希望修改得到社会群体的“正式”认可，就必须提出支持修订的论据。有说服力的论据可以调动各种可用性维度，例如所需精力和书写便利性、避免文化冲突、可读性、映射接近性 [20]，甚至诉诸社区价值观（例如 Python 之禅 [1]）。在计算领域，修改往往采取征求意见稿（RFC）的形式；这类文档围绕潜在功能展开，供共同体进行讨论。

3.3.3 约 **过软件和硬件在物质层面被制度化并物化**。一旦某种记法变得流行，它就会在机器和排版机等物质中被物化：乐谱记谱法在软件中被物化，其形式的规则可被强制执行并施加给用户 [29]；统一建模语言记法在 Visual Paradigm 软件中变得可交互，并可从图表生成代码 [150]；自古代以来使用的表格 [27], 在电子表格软件中被物化。

3.3.4 社 **法在社会层面被制度化，成为一种入门仪式以及知识传递的预期媒介（例如考试、已发表论文）**。对“恰当”或“正典”诠释的协调，也通过教育材料及其他副文本材料在社会层面实现；这些材料教导人们如何正确诠释记法，并促使人们在社会层面达成共识。在权力结构的强化下，规范用法也随之得到巩固。例如，物理学家在求职报告中正确地“演绎”费曼图，与他们能否获得这份工作息息相关。

3.3.5 认[illegible]**无法捕捉的变异维度加以神圣化，并继而贬低它们**。广受欢迎且形式化的记法可能会变得神圣化——原本旨在描述世界的记法，转而成为使用者认识世界的图式，筛选出何者“有意义”、何者无意义。围绕某种记法形成文化后，它对使用者而言可能具有近乎宗教般的意义。Lilliestam在谈到音乐理论如何围绕乐谱记谱法展开时指出：“记法设定了一个框架，为作曲家呈现各种可能性与局限..... [声]音色、微音程、“蓝调音”以及节奏上的细微差别，都无法通过 [乐谱] 记谱法捕捉。” [86, 第 198]页。在西方古典音乐中被视为无关紧要的世界变异维度，往往难以或无法捕捉。他将此称为记谱中心主义——“我们将‘音乐’等同于记谱音乐..... [以及]认为记法包含了关于音乐的最终真相”——这进而导致音乐学家贬低那些无法表征的维度 [86, 第 196]页。

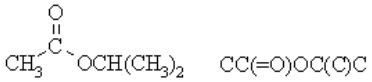
与此相关的是，需要不断澄清解读或使用所需的“正确视角”，无论这种视角是空间性的、具身性的，还是某种理论或哲学信念；在争论某项记法扩展是否合适时，这些视角都可能派上用场。例如，Python 之禅是一份官方 PEP，其中有这样的条目：“应该有一种——最好只有一种——显而易见的做法。”尽管这些主观原则不属于记法的“形式”定义，却构成了一种共同的底层文化，后来人们据此评判 Python 的使用方式、语言扩展，甚至用户本身 [1, p. 9]。

3.[illegible]**式价高昂（在经济、物质和社会层面），且传播缓慢**。由于记法已在社会和物质层面实现制度化，传播变更在这两个方面都面临相当大的阻力。例如，Sign Writing 最初是从观察者而非手语者的视角书写的。然而在1984年，“两位聋人五线谱成员”强调说：“我们是从自己的视角表达自己的语言。书写时，我们看到的是自己的手，而不是别人的手。”Sutton 写道：“他们是对的。但这需要一番调整，因为在此之前的十年里，所有出版的 SignWriting 教科书和文档都是以接受者的视角书写的.....这项变更花了四年多才完成” [136, 第 37]。计算领域一个著名的例子是从 Python 2 迁移到 Python 3；破坏性变更引发了强烈的“抗拒变更”，以至于“Python开发者开始对选定的功能进行‘向后移植’”，以抵制升级 [92, 第 765 页8]。

3.[illegible]**接修订的成本很高，用户会非正式地分叉出该记法。分支最终可能启发“官方”记法的发展，或再次与其合并**。为避免官方修订记法所引发的阻力，用户可能会选择绕过漫长的流程，转而创建未经治理机制认可的记法版本。2010 年代前后 JavaScript ES6 中转译技术的广泛应用就是一个很好的例子：在新修订内容被纳入浏览器等物质基础设施的更新之前，Web 开发者就会使用语言的最新特性编写代码。这些“非官方”的改编随后可能启发“官方”记法的发展；例如，CoffeeScript（一种句法类似 Python 的可编译为 JavaScript 的语言）启发了 JavaScript 的发明者 Brendan Eich [42]。负责以形式化方式扩展该语言的 TC39 联盟随后采纳了箭头函数、可选链（?.）、类和 super 等扩展 [43]。

3.3.8 社[illegible]**作为基础记法周围的次级记法或惯例出现的内容，可能会被形式化：要么自身成为具有严格实践规则的记法，要么被纳入基础记法**。 Python 将空白字符正式纳入语言本身（这一理念借鉴自 ABC），就是一个将利用未定义感知通道的非正式语法实践引入语言形式语义的例子（为空白字符赋予意义）。同样，最初作为次级记法或纯粹“惯例”出现的内容，例如文档字符串 [57]——用户可以选择不遵循它而不会受到惩罚——，后来也可能在下游解析器要求精确性时被形式化；例如，软件集成开发环境中的工具提示就依赖预期的语法，以便以印刷方式呈现文档字符串。

3.3.9 约[illegible]**码：适用于一种媒介的记法会被重新编码，以适应另一种媒介的可供性与约束**。 一种记法一旦证明有用并实现制度化，其发展并不会就此结束。它往往必须适应新的物质约束，尤其是计算机系统及其标准化编码方案和输入设备所带来的限制。手语书写符号必须加入 Unicode 标准；费曼图必须在 FeynArts 中进行描述；电子电路以 SPICE 网表编码；化学记法则在线性化为简化分子线性输入规范（SMILES）记法，以便进行电子搜索和存储（右侧是 SMILES [109]）：



CC(=O)OC(C)C

尽管这些线性化编码受到围绕 ASCII 和拉丁字母键盘的文化标准的约束，但技术发展能够放宽以往的材料约束；例如，笔输入设备的进步如何促成了将打字输入的编程与手绘图表相结合 [4, 163]。

3.3.10 社[illegible]**用者失去控制：随着记法被更广泛的社群采纳并形式化，最初的意图遭到扭曲和破坏，而这些社群逐渐形成不同的价值观和承诺**。 纵观各种历史，一个令人意外的共同模式是：原发明者似乎会“失去对记法的控制”。一旦某个共同体开始拥有这套记法，其价值观和意图便会偏离原发明者，进而引发争议，最终可能导致发明者离开共同体、否认新的演变形式，或重新确立控制权。例如，Guess 对拉班动觉记谱法的采用方式与拉班的做法有所不同，导致拉班撰写了一篇长篇论述，以传播“真正的”版本 [155]；Gruber 因人们聚在一起制定 Markdown 标准而“非常不满”，于是试图管控该术语的用法 [5]；以及 Guido van Rossum 在围绕 PEP 572：赋值表达式发生激烈冲突后，辞去了 Python “终身仁慈独裁者”的职务 [2]：“我再也不想为了一个 PEP 如此奋力争斗，却发现那么多人鄙视我的决定。” [145]。

4 记法发展理论的兴起

我们的历史分析表明，从认知和社会层面来看，一种记法的发展过程如下：

(1) 列举目标领域中有意义变异的维度；随着人们（无论是发明者还是使用者）遇到或考虑更多情境，这些维度会不断增加；(2) 将有意义变异的维度映射到表征的感知通道；(3) 通过视觉类比来设计记法，借助倾倒杯子、旋转形状等具身变换，发挥感知可供性，并确保这些“自然”的操作在目标领域中具有意义

因此，记法的定义过程是：将概念／经验领域中有意义变异的一个维度映射到表征领域中可通过感知区分的一个变化维度。有意义变异的内涵由实验、文化知识或单纯的自由选择来确定。先前未知或被认为是不变的变化维度，可能会在进一步的经验证据表明其有意义时，转而成为有意义的维度（例如，电子键的发现促使 Lewis 扩展了分子记法 [82]）。记法通过剥离不变维度，摆脱图像表征；书写不变维度需要耗费精力，却不会提供额外信息。

除上述记法发展的社会阶段外，我们通过反思分析还发现了三个功能阶段——描述、生成和评估阶段（术语借鉴自《交互生成理论》 [11]）：

(1) 描述阶段：人们试图描述世界或概念现象，以管理复杂性和/或协调行动。此阶段的记法发展遵循映射接近性 [20, p. 330], ，即关注指称对象能否通过记法加以表征。记法通过应用于各种场景而不断发展；在这些场景中，人们会检验并修订记法，以判断它能否表征目标领域中有意义变异的维度，以及用户能否相应地一致应用和解读它。

(2) 生成阶段：一旦某种记法足以描述某个领域中的现象，人类便会将其用作生成理论：以生成新的想法、概念和假设，并对其进行测试或实施。例如，音乐家可以不接触乐器便用记法创作音乐 [29]；化学家可以依据演绎式符号操作推测反应 [76]；人机交互研究者则会通过“在记法中构建所有可能字形组合的树”[ing]，探索“新微手势的生成”，并通过与现有文献涵盖的微手势记法数据集进行比较，找出空缺 [30,p. 16]。

（3）评价 阶段：如果一种记法在某个领域广受欢迎，它就可能成为评判经验现象或概念现象的度量标准——这种记法会变成一副有色眼镜，人们透过它来解读世界。起初，它是将现象的变异维度映射到感知变异维度的映射，如今却发生了倒转，即记法中现象的“可表示性”筛选出了哪些经验变异维度才是重要的。[9] 并非所有记法都会以这种方式被神圣化。例如，一些聋人担心，SignWriting 可能会成为一种评价性度量标准，决定人们如何评判和重视美国手语的新形式 [127]。

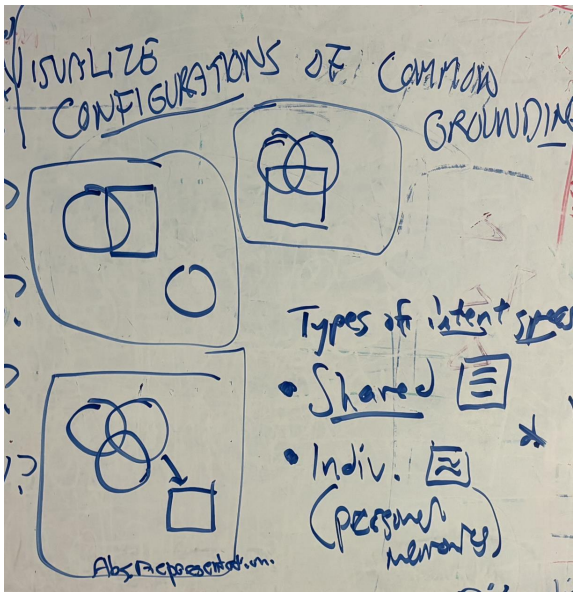
一个演算示例。 为了将早期认知模式具体化，并反思我们作为人机交互与软件设计者自身的记法实践，我们考察了一种在另一研究项目的合作过程中、于人机对齐语境下涌现的记法。

图4将共同基础记法展示为一项进行中的工作，它由指向维恩图的连接隐喻创建而成。该隐喻最初源于一种结构相似性：“人类”与“人工智能”各自拥有对方未知的独立“理解”，同时也共享一种共同理解（“共同基础” [35]）。很快我们意识到需要标示实体的类型；因此，我们通过扩展先前不变的形状感知通道来扩展维恩记法——如今，圆形表示“柔软”的人类，方形表示“锐利”的人工智能（一种奠基 隐喻）。如此一来，我们开启了一个具有两类表征的形状感知通道，若概念情境需要区分新的实体，后来的合作者或许可以加以扩展（例如，用三角形表示组织）。除两个重叠形状之外，维恩图还具有一个良好特性：可推广到更广泛的人类与人工智能排布，其中存在多个人类/人工智能，并以不同连接方式关联（图4，左）。

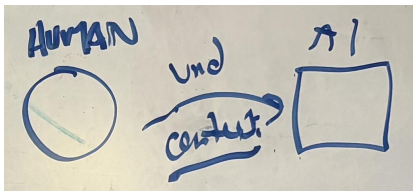
然而，在使用过程中，我们发现这种类比可能会造成误导：重叠部分营造出一种假象，仿佛人类和人工智能拥有共同的信念，但实际上两者之间始终存在差距。要递归地表示“A 对 B 的模型”以及反过来的情况也很困难（“A 对 B 如何看待人工智能的模型”）。人工智能智能体还具有不同于人类的特性，例如可能永远不会忘记说过的话。此外，在这个过程中，我们也忽略了合作者之间可以共享记忆文档这一概念。重叠区域是记忆文档吗？那么，人工智能收集的有关人类 A 的合作者 B 的信息又该如何表示？这些信息对 B 是隐藏的。我们考虑过在某个实体的底部角落添加一个 ■，使其看起来像一份“文档”，以表示该实体拥有外部记忆文档，但这同样会引发歧义和疑问。是否所有且仅有人工智能智能体才需要这类记忆文档（也就是说，“这难道不是一种不变量吗？”）？ 如果是这样，为什么还要明确画出这些文档呢?

总而言之，维恩图 [146] 与人机对齐之间的类比对齐这一认知过程，揭示了那些‘并不完全契合’的语义断裂。有意义的变异维度——此处即来自克拉克的共同基础理论的概念 [35], ——并不容易被表示，这促使我们要么扩展该记法，要么彻底放弃它。但这一过程同样富有成效。采用维恩图记法形成了一种框架装置，即一种图式或“思维工具”，我们随后借此审视并澄清那些至关重要的意义维度。

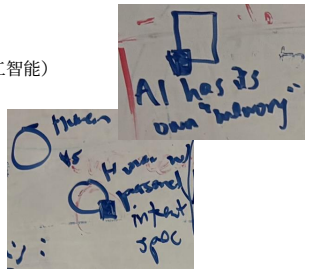
[9]对这一阶段的一种合理认知解释是，记法会形成新的图式 [141], ，这些图式进而“塑造我们所看到和听到的内容、我们储存这些信息的方式，以及我们日后获取这些信息的方式” [151, 第 921]页。冯·希佩尔等人通过一系列实验表明，习得的图式“会通过抑制相关信息和无关信息的感知编码，减少被记住的信息量”——从字面意义上说，这会使用户倾向于突出图式中具有意义的维度，而忽略不具意义的维度。



试图用维恩图描绘不同人类（○）与人工智能（□）排列方式之间的“共同基础”



有意义的概念被映射到一种形状感知通道（圆形代表人类，方形代表人工智能）



临时扩展，用于描绘某个实体拥有一份“记忆文档”。注意到不一致之处在 ■ 的位置安排上

随着记法在更多情境中接受检验，其他实体（如“组织”）也加入进来。于是问题出现了：“我们是否应该扩展它？”

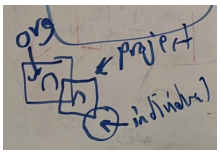


图 4：孕育中的一种记法：摘自一位人机交互研究者的白板。“共同基础”是人类与人工智能之间的概念，通过连接维恩图的隐喻来呈现。在类比对齐过程中，人们发现目标领域中的变化维度无法表示或被遮蔽，因此促使符号系统扩展、改变或弃用。

5 设计的初步启示

下面我们为系统设计者提出一些建议，并提供受我们总结的模式启发的具体示例。这些只是我们想到的一些有趣点子，并非详尽无遗的清单。虽然这些想法不依赖特定实现，但许多都可以借助生成式人工智能和大语言模型的世界知识，预判歧义、揭示与其他记法之间的文化冲突，或帮助用户完善支持符号系统修订的有说服力的论据。

帮助识别、梳理并确定问题领域中变异维度的优先级。 系统可以帮助用户从概念域/经验域中收集示例，揭示变异维度，估算其出现频率，并标记被忽略的维度。这与以用户为中心的设计中的需求探索相呼应；在这一过程中，从业者会列举用户差异和需求的各个维度。

> **为什么？** 记法将概念域/经验域中有意义变异的维度映射到感知通道（3.1.4）。记法一旦确立，之后再引入此前忽略的维度就很困难（3.2.12）。反过来，突出不变或罕见的维度既会浪费精力，也可能误导用户（3.1.4）。

> **示例：** 例如，对于 μ-字形 [30], ，系统可以帮助设计师查阅微手势文献，并协助对微手势的变异及其维度进行分类。

明确哪些感知通道仍可使用，哪些感知通道已承载了过多意义。在创建记法时，应引导用户使用感知通道（例如颜色、大小、接近性、闭合）。通过展示这些通道如何被该记法使用（“饱和”），用户可以做出更明智的决策：过度使用的感知通道或许可以简化，而未使用的通道则可能揭示新的机会。

> **为什么？** 记法通过利用尚未使用的感知通道（3.2.5）以及为现有通道赋予意义（3.2.4）而不断演变。有些通道可能会被忽略（3.1.5）。

> **示例：** 在图 3 中，“形状”和“重叠”等感知通道已饱和，而颜色和大小等通道目前没有意义。系统可以列出这些通道及其语义映射，并建议随着新场景的出现和新的变异维度的产生，引入尚未使用的通道。

预见歧义、不一致和潜在的误解。 用户应能获得一些信号，了解自己对记法所做更改的质量。系统应标记不一致或歧义，并帮助用户解决这些问题。系统还可以识别可能被他人误解的元素（例如，使用不适用于红绿色盲者的颜色，或某个符号与另一种广为人知的记法重叠）。

> **为什么？** 外观相似的记法会造成混淆（3.1.11），未解决的歧义可能导致不同的解读（3.2.2），而偶然的视觉相似性可能误导用户（3.1.6）。

> **示例：** 设想一款用户手绘的网格游戏，玩家可以放置并“移动”自己的“单位”（X 和 O），让它们与相邻单位“战斗”。系统可以提醒设计者，这种设计在视觉上与井字棋相似，可能会造成混淆，并建议其他设计方案。系统还可以提示用户明确符号大小是否重要。

鼓励简化设计，以减少读写记法所需的努力。 用户应能够发现简化方法，并从书写努力、记忆努力和空间效率等方面衡量其记法的复杂程度。系统应建议进行简化，为频繁排列提供缩写，并帮助用户根据定量指标（例如所需笔画数）比较不同方案。

> **为何？** 记法的目标是管理复杂性、减少书写努力并便于比较（3.1.1）。随着记法演变，频繁排列往往会被缩写（3.2.11）。

> **示例：** 系统可以存储并比较某种记法的示例用法，以识别关键变异维度和频繁排列。由此可以找出可删除的无意义细节，以及用于重复模式的简写。系统还可以预测书写和记忆所需的努力。

帮助用户将现有记法与隐喻相联系，并支持由此产生的类比对齐和错配识别过程。系统应帮助用户搜索、发现并建立与现有记法的连接隐喻。这一类比过程应考虑源记法和目标记法的应用领域、所利用的感知通道以及可供操作。尤为重要的是，应突出显示结构错位，并针对可能的解决方案提供建议。

> **为何？** 在记法发明过程中，将隐喻与以往的记法相联系至关重要（3.1.7），但从源记法到目标记号的类比对齐过程可能导致用户忽略重要维度（3.1.8），或引发误解（3.1.6）。

> **示例：** 在我们的研究中（图 3），维恩图暗示重叠区域代表共享内容，但这会造成误导，因为目标领域涉及的视角永远无法完全一致。系统或许可以提醒我们，这种结构上的不匹配可能会引发误导性结论。

鼓励使用植根于具身经验的隐喻。 除了支持将隐喻与现有记法相连接之外，还应鼓励将记法奠基于具身的人类经验。系统应建议如何通过能够引发特定动作的物理隐喻来传达记法的使用方法。在此过程中，系统应确保所建议的隐喻不会造成交叉映射及其他不匹配，从而引发被禁止的动作或持续的误解。

> **为什么？** 将记法奠基于具身感知经验，或许能借助用户直觉提高可用性（3.1.6）。但这需要谨慎处理，因为当可供动作与目标领域中的有意义操作不对应时，这些隐喻也可能误导用户（3.1.6）。

> **示例：** 对于表示身体运动的记法，建议在视觉或结构上与身体部位相似，但应尽可能简化（抽象化），以减少书写努力。

针对材料约束提供定制化建议。 系统应了解将用于呈现记法的媒介类型（例如使用钢笔与铅笔在实体纸上书写），并针对该媒介提供约束和建议。此外，系统还可以为希望将给定记法编码到新媒介中的用户提供支持，建议与原记法尽可能保持一致的调整方案。

> **为何如此？** 材料约束在记法发展中发挥着决定性和限制性作用（3.1.10）。同一种记法在不同媒介之间转换时，可能需要重新编码（3.3.9）。

> **示例：** 对于通过键盘输入的记法，系统应建议使用兼容 ASCII 的字符，并根据特定键盘布局上的字符可访问性估算书写努力。

支持在使用记法与迭代构建记法之间灵活切换。 程序员可以编写代码，然后运行代码进行测试；同样，用户也应能够编写记法并定期测试。记法的整个发展过程中都应进行测试，并且多个用户应能够同时参与。系统应针对测试中发现的问题提供修正建议。

> **为何如此？** 从历史上看，记法发展与迭代设计相似（3.1.9）。这些测试阶段往往会促成对记法的修订（3.2.8）。测试也有助于协作，共同推动记法发展（3.1.10）。

> **示例：** 系统可以提供一个测试沙盒，确保记法的使用符合规范。当出现歧义时，系统也可以提出澄清问题，或提示用户考虑边缘情况：“你的记法能否表示 [不寻常的情景]？”并追踪那些会导致失效或需要扩展的情景。

支持对成熟既定记法进行个人变体，这些变体可能为原始符号体系带来启发。系统应支持创建个人变体，即使这些变体可能扰乱或打破先前的形式化（类似于让软件变得“可塑” [88]）。同样，规范记法的作者应能够追踪这些变体，并从中汲取灵感。

> **为什么？** 由于直接修订的成本很高（3.3.7），用户会创建记法的非官方变体。反过来，这些变体可能启发原始符号体系的发展。此外，随着记法被他人共享和改编，它们往往会逐渐分化（3.2.2）。临时扩展可以暂时使既有记法的某些方面变得非正式，从而有望带来理论上的收益（3.2.6）。

> **示例：** 设想一位 Lua 用户怀念 CoffeeScript 的高阶特性。[10] 系统可以允许用户以内联方式引入新的句法和语义——不是通过宏，而是以临时方式。这样，系统便将 Lua 从僵化的形式性领域中“提升”出来——这是一种“非形式化操作”（3.2.6），使其在用户的局部范围内更具可塑性，并模拟其执行过程。

促进新记法的共享、发现与批评。 共享记法应该很方便，让其他人能够使用或修改。共享的记法应当可以搜索，以便其他人发现、采用、批评并对其进行“分叉”。

[10]这种情况类似于 MoonScript：这是一种编译为 Lua 的脚本语言，其开发源于作者希望编写更简洁的代码（避免“关键字噪声”），并采用类似 CoffeeScript 的抽象：https://moonscript.org/。

> **为什么？** 推广一种记法并非易事。早期采用者对于推广、传播和形式化记法至关重要（3.2.1）。通过多样化场景进行测试对于强化记法设计至关重要，但设计者往往受个人与文化偏见的限制（3.1.4, 3.1.9）。

> **示例：** 记法可以采用标准文件格式，以便在支持高级搜索的平台上轻松共享，并可按领域、材料约束、感知通道（例如“使用接近性”）或基础隐喻进行搜索。

通过讨论、共识构建和调查机制，促进记法治理的协调。 可以为记法相关共同体提供支持，使其能够讨论变更。同样，也应建立处理冲突与分歧的流程，并将领导职位传承给未来成员。

> **为什么？** 记法制度化涉及成立委员会并建立治理结构，以确保记法在时空中的可复现性（3.3.1）。多样的使用情境可能推动记法演化（3.2.3）。共同体可能会从记法的原发明者手中接管控制权，从而引发冲突（3.3.10）。

> **示例：** 平台可以通过定义角色（维护者、贡献者、用户），设置调查机制以就提案达成社区共识，或监测使用情况以识别常见模式或利用不足的特性，来支持社群开展讨论。

帮助用户为请求修订由共同体管理的既定记法提出有说服力的论据。应通过考虑可用性、付出的努力、一致性以及对共同体价值观和原则的遵循等因素，支持用户阐明为何需要进行修改。此外，平台还应估算修改的成本，并建议如何降低这些成本。

> **为什么？** 一旦某种记法得到广泛使用，共同体成员就不能简单地对其进行修改，而必须提出论据，使修订得到正式接受（3.3.2）。如果没有缓解计划，修改会遭遇阻力，并且代价高昂（3.3.6）。

> **示例：** 例如，系统可以支持用户准备针对 Python 语言的 Python增强提案（PEP） [154]。系统可以检索相似的讨论和以往的 Python 增强提案，以避免重复提出相同想法。系统还可以帮助作者调整设计，并诉诸共同体价值观，例如某种做法是否符合“Pythonic” [1]。

6 迈向动态、用户定义、增量形式化抽象

在本文中，我们为记法发展理论奠定了一些基础。然而，梳理记法演化是一项艰巨的任务，无法在一篇论文中全面完成。我们希望研究结果能为构建更完善的记法与抽象发展理论提供有益的起点。具体而言，我们的研究主要立足于宏观尺度；我们勾勒了高层次的模式，但没有详细描绘形式性如何在实践中于每时每刻得以实现。未来的研究或许可以通过追踪每时每刻的（非）形式化举措来丰富我们的形式性社会建构主义观点，或许还可以联系实验符号学领域的研究 [50]。

我们的工作回应了人机交互领域由来已久的愿景：实现动态抽象，让用户能够在表达时，借助当下最得心应手的记法（界面）动态地交流和表达自我，而不受开发者预先设定的记法所限 [96, 143, 144, 148, 149]。这种“动态落地” [143] 有望扭转创意支持工具中传统的单向权力关系 [83]。Bret Victor强调，动态落地仍是人机交互未来发展的关键瓶颈。他指出，尽管他在 DynamicLand 取得的进展颇具前景，但仍然 [148]:

> “……大多局限于已经创作好的工具集中。我们还没有通用的从零开始计算草图工具。基于文本和键盘的编程速度太慢、要求太高，而且过于内向，无法支持一场讨论自身以外事物的对话……如果我们能在实时对话中，用即兴手绘的动态模型表达新想法，那将是一个重要的里程碑。”

最后，我们联系用户界面的历史，并对支持语言演化的系统提出更广泛的启示。

6.1 用户界面作为记法

鉴于用户界面是一种记法形式，其演化密切遵循我们所识别的模式也就不足为奇。人机交互共同体已经认识到奠基隐喻的重要性，并发展了理论和原则，以帮助界面设计者发现并发展此类隐喻 [28, 65, 108]。例如，基于现实的交互 [68]主张将界面奠基于具身体验，例如我们对物理学的理解以及身体意识。早期界面基于奠基隐喻进行设计，例如模拟办公室的桌面隐喻，文件存储在文件夹中，信息显示在窗口中 [104]。同样，Textoshop等近期工作运用连接隐喻，使一种界面（如用于图像的Photoshop）被适配到不同的语境或领域，如写作 [95]。

然而，Blackwell [19] 指出，隐喻在促进人机交互设计的可用性或效用方面并非总是“稳操胜券”。早期界面通过使用、约束（例如出现在桌面上的垃圾桶 [104]）以及利用新的感知通道（例如图标取代文本 [130]）而演化，以至于原始隐喻几乎已难以辨认：很少有人知道文字处理器中的插入符是打字机机头的隐喻，也很少有人知道我们选择文本的方式是用笔划过文本这一做法的隐喻 [104]。事实上，对许多新用户而言，一些隐喻已不再有多大意义。例如，许多人点击软盘图标来保存文档，却并不知道它代表什么。

我们对记法演化的研究揭示了类似的模式：记法发展时，往往会从启发其最初创作的隐喻中彻底脱离。随着时间推移，用户和用法不断变化，它通常不再是指代其他对象的隐喻。摆脱隐喻包袱后，它便可以更自由地修改（例如 MacOS 中的“空间”，就与

物理桌面的特性毫无关系 [19])。换句话说，隐喻，或者说至少与既有知识之间可辨识的对应关系，其价值或许恰恰在于最初的对应关系最终会被遗忘。即使最初的家族相似性已难以辨认，这种新的抽象也可以在后续发展链条中重新用作新的连接隐喻 [79]。因此，我们认为，“可遗忘性”是记法演化不可避免的结果，甚至是其成功的标志。

6.2 未来的系统如何支持记法演化，而不仅仅是翻译?

近期的人工智能系统在翻译为既有形式体系方面极其有效，但在创建新的形式体系方面却并非如此。它们使我们沿着形式化方格（图1）“水平”移动，却很少促进新记法的创建与传播，甚至可能造成阻碍。有研究表明，大语言模型会通过复制并放大主导语言来遏制文化演化 [46, 142]。如今，软件开发者选择编程记法时，考虑的已不再是自己有多喜欢该记法，而是大语言模型对其理解得有多好。[11] 如果没有一种拥抱语言演化的新范式，“我们将面临永远困在2020年–2023年的软件生态系统中”的风险 [119]——其他语言生态系统也可能如此。

在这样的背景下，未来的信息系统如何继续支持语言的自然演化？一些新兴的上下文学习研究正在探讨大语言模型如何生成灵活的领域特定语言，例如语法提示 [153]。然而，这类研究通常将需要学习的新抽象视为既定事实，再次将其起源隐藏起来。当前的人工智能系统支持从非正式想法到既定记法的“水平”翻译，那么，我们应如何确保创建这一“垂直”过程——新记法、新抽象——也得到支持？我们如何与机器共同创建一种新记法，随后通过该记法进行交流，甚至将这种记法分享给更广泛的社群?

我们的模式和阶段对这些问题有一些启示。首先，新记法似乎主要（但并非仅仅）通过将隐喻与既有记法相连接而产生。那么，记法演化或许可以类比自然语言演化：洋泾浜方言会发展成克里奥尔语 [18]。近距离观察时，记法似乎彼此独立；但从远处看，记法发展更像是一个渐进突变、偶有突破的演化过程，这与技术的社会建构主义视角相吻合。该视角认为，“技术人工制品.....的发展过程，是变异与选择交替进行的过程” [114, 第 411] 页。

其次，我们的研究提醒我们，形式化是一个社会过程。形式性既不是一个光谱，也不是一个可以抵达的终点，而是需要经过审慎协商，并且始终可能受到质疑。对于旨在支持渐进式形式化的未来系统而言，将形式化视为充满辩论与价值设定的过程——而不是纯粹的翻译、技术性或机械性任务——至关重要。尽管我们希望工具能够与我们共同发展，并为我们服务，但我们不希望未经社区同意，就强行推行大幅度的记法变更。

第三，尽管许多有意义的概念／经验变异维度对发明者而言仍未明言，但系统有机会让这些假设变得更加明确、具体。Beaudouin-Lafon 的实体化概念对此很有帮助：“将概念转化为对象的过程”，借此“[i]n用户界面中，生成的对象可以在屏幕上明确呈现，并可对其进行操作” [10, 第 449] 页。未来用于支持记法发展的界面，如何将变异维度实体化，使之具体可感，甚至可以操控？“替换”已饱和的感知通道——例如用形状替代颜色——意味着什么?

第四，尽管标准化委员会及 RFC 等实践至少与计算本身一样古老，但我们不禁注意到，几乎没有人机交互研究关注如何支持诸如提交提案、扩展编程语言等记法的过程。直接致力于设计记法统一的研究也明显缺失。软件工程领域关于语言治理的实证研究颇有助益 [8, 71], ，能为有针对性的设计提供更多见解。

最后，记法也关乎可解释性以及人类认知的持续繁荣 [137]。例如，如果人工智能系统开始能够完成超越人类认知的任务，而人类必须努力理解并验证这些任务，那么过去的记法或许并非与人类沟通的最佳形式。无论是人类还是人工智能，都可能受益于新记法，即便是在现有领域中；例如，一些研究表明，概率论和电学领域的新记法能改善初学者的学习效果 [32, 33]。我们希望这项工作能启发读者，并为解决这些棘手问题指明前进方向。

[11]例如：“我现在主要使用 Python 工作，最主要的原因是大语言模型更擅长使用它，而且默认使用它，而不是 R 或 Julia” [119]。

致谢

我们感谢加拿大自然科学与工程研究委员会（NSERC）提供的支持，项目编号为 RGPIN-2024-04049。

参考文献

[1] Carol V. Alexandru、José J. Merchante、Sebastiano Panichella、Sebastian Proksch、Harald C. Gall 和 Gregorio Robles。2018年。Python惯用表达的使用情况。载于《*2018*年美国计算机协会 *SIGPLAN* 编程与软件新思想、新范式与反思国际研讨会论文集》（美国马萨诸塞州波士顿）（*Onward! 2018*）。美国计算机协会，美国纽约州纽约市， 1 11 10 1145 3276954 3276960 –。https://doi.org/。/。Chris [2] Angelico、Tim Peters 和 Guido van Rossum。2018年。PEP 572——赋值表达式。PEP。https://peps.python.org/pep-0572/ Python增强提案的最终草案。[3] Ian Arawjo。2020年。《编写代码：编程记法与实践的文化建构》。载于《*2020*年计算机系统中的人为因素国际会议论文集》（美国夏威夷州檀香山）（*CHI '20*）。美国计算机协会，美国纽约州纽约市，1–15。https://doi.org/10.1145/3313831.3376731[4] Ian Arawjo、Anthony DeArmas、Michael Roberts、Shrutarshi Basu 和 Tapan Parikh。2022年。笔记本环境中的记法编程：量子电路案例研究。载于《第*35*届美国计算机协会用户界面软件与技术年度研讨会论文集》（美国俄勒冈州本德）（*UIST '22*）。美国计算机协会，美国纽约州纽约市，文章编号62，共20页。https://doi.org/ 10 1145 3526113 3545619 。/。 2014 [5] Jeff Atwood。。标准Markdown如今已成为通用Markdown。 https://blog.codinghorror.com/standard-markdown-is-now-common-markdown/。Coding Horror 博客，访问日期：2025年9月2日。[6] John Backus。1978 13, 8 1978 165 180。Fortran I、II 和 III 的历史。*ACM SIGPLAN*通讯（），–。[7] J. W. Backus、R. J. Beeber、S. Best、R. Goldberg、L. M. Haibt、H. L. Herrick、R. A. Nelson、D. Sayre、P. B. Sheridan、H. Stern、I. Ziller、R. A. Hughes 和 R. Nutt。1957年。FORTRAN自动编码系统。载于发表于

1957年2月26日至28日西部联合计算机会议：可靠性技术（洛杉矶，加利福尼亚）论文集（IRE-AIEE-ACM '57（西部））。美国计算机协会，美国纽约州纽约市，188–198。https://doi.org/10.1145/1455567.1455599[8] Flore Barcellini、Françoise Détienne 和 Jean-Marie Burkhardt。2009年。在线交互空间中的参与：开源软件共同体中的设计-使用中介。《国际工业人因学杂志》39, 3（2009），533–540。[9] Adriano Barenco、Charles H Bennett、Richard Cleve、David P DiVincenzo、Norman Margolus、Peter Shor、Tycho Sleator、John A Smolin 和 Harald Weinfurter。1995年。量子计算的基本门。《物理评论 A》52, 5（1995），3457。[10] Michel Beaudouin-Lafon。2000年。工具性交互：用于设计后 WIMP 用户界面的交互模型。载于 SIGCHI 人机交互系统中的人为因素会议论文集（荷兰海牙）（CHI '00）。美国计算机协会，美国纽约州纽约市，446–453。https://doi.org/10 1145 332040 332473 。/。Michel Beaudouin-Lafon、Susanne [11] Bødker 和 Wendy E Mackay。2021年。交互的生成理论。《ACM 人机交互汇刊》（TOCHI）28, 6（2021），1–54。[12] Dawid Bednarski。2022年。Markdown 的历史：无代码运动的序曲。https://www.taskade.com/blog/markdown-history/ 访问日期：2025-09-07。[13] Peter J Beek 和 Arthur Lewbel。1995年。杂耍科学。《科学美国人》273, 5（1995），92–97。[14] Robert W Bemer。1971年。COBOL 历史概述。《霍尼韦尔计算机杂志》5, 3（1971），130–135。[15] Jöns Jakob Berzelius。1814年。试图通过应用电化学理论和化学配比，建立纯粹的矿物学科学体系。R. Baldwin 和 W. Blackwood，伦敦和爱丁堡。[16] Jöns Jacob Berzelius。1833年。《关于有机原子组成的思考》。《物理学年鉴》104, 8（1833），617–630。[17] Claudia S. Bianchini 和 Fabrizio Borgia。2012年。《书写手语：分析 SignWriting 系统从1995年至2010年的演化，并提出未来发展建议》。载于《瓦尔纳技术大学国际周年纪念大会论文集》，第6卷。瓦尔纳技术大学，保加利亚瓦尔纳，118–123。[18] Derek Bickerton。2016年。《语言的根源》。语言学经典第3卷。Language Science Press，柏林。https://doi.org/10 17169 91 109 。/langsci.b。[19] Alan F Blackwell。2006。隐喻作为设计工具的实体化。《ACM 人机交互汇刊》（TOCHI）13, 4（2006），490–530。[20] Alan F. Blackwell、Carol Britton、Anna Louise Cox、Thomas R. G. Green、Corin A. Gurr、Gada F. Kadoda、Maria Kutar、Martin Loomes、Chrystopher L. Nehaniv、Marian Petre、Chris Roast、Chris Roe、Allan Wong 和 Richard M. Young。2001年。符号系统的认知维度：认知技术的设计工具。载于第四届认知技术国际会议论文集：心智工具（CT '01）。Springer-Verlag，柏林，海德堡，325–341。[21] Grady Booch。1981年。用 Ada 描述软件设计。ACM SIGPLAN通讯16, 9（1981），42–47。[22] Susan E Brennan。1991年。与计算机对话及通过计算机进行对话。《用户建模与用户适应性交互》1, 1（1991），67–86。[23] Aline Brito、Marco Tulio Valente、Laerte Xavier 和 Andre Hora。2020年。你破坏了我的代码：理解应用程序接口破坏性变更的动机。《实证软件工程》25, 2（2020），1458–1492。[24] James Robert Brown。2018年。费曼图如何运作？《科学视角》26, 4（2018），423–442。[25] Thomas J Bussey、MaryKay Orgill 和 Kent J Crippen。2013年。变异理论：一种学习理论，也是化学教育研究的实用理论框架。《化学教育研究与实践》14, 1（2013），9–22。[26] Florian Cajori。1993年。《数学记号史》（两卷合订为一册）。第1–2卷。多佛出版社，美国纽约州纽约市。[27] Martin Campbell-Kelly、Mary Croarken、Ray mond Flood 和 Eleanor Robson。2003年。《数学表格的历史：从苏美尔到电子表格》。牛津大学出版社，英国牛津。https://doi.org/10.1093/acprof:oso/9780198508410.001.0001[28] John M. Carroll 和 John C. Thomas。1982 年。隐喻与计算系统的认知表征。《 IEEE 系统、人类与控制论汇刊》12, 2 1982 107 116 10 1109 1982 4308795 （），–。[29] https://doi.org/。/TSMC.。Vincent Cavez、Catherine Letondal、Caroline Appert 和 Emmanuel Pietriga。2025年。EuterPen：释放乐谱书写中的创造性表达。载于《2025年计算机系统中的人为因素国际会议论文集》（CHI '25）。美国计算机协会，美国纽约州纽约市，文章编号760, 16 10 1145 3706598 3713488 页。https://doi.org/。/。[30] Adrien Chaffangeon Caillet、Alix Goguey 和 Laurence Nigay。2023年。µGlyph：一种微手势记法。载于《2023年计算机系统中的人为因素国际会议论文集》（德国汉堡）（CHI '23）。美国计算机协会，美国纽约州纽约市，文章编号269，共28页。https://doi.org/10.1145/3544548.3580693 [31] Kathy Charmaz. 2017. 建构主义扎根理论。《积极心理学杂志》12, 3 (2017), 299–300. [32] Peter C-H Cheng. 2002. 用于学习的电气化图表：复杂表征系统的原则。《认知科学》26, 6 (2002), 685–736. [33] Peter C-H Cheng. 2011. 或许适用于学习的良好图表：概率论的表征认识重编码。《认知科学专题》3, 3 (2011), 475–498. [34] Mauro Cherubini, Gina Venolia, Rob DeLine, and A my J. Ko. 2007. 我们去白板上画吧：软件开发者如何以及为何使用绘图。载于《计算机系统中的人为因素国际会议论文集》（美国加利福尼亚州圣何塞）（CHI '07）。美国计算机协会，纽约，美国，557 566 10 1145 1240624 1240714 –。[35] https://doi.org/。/。Herbert H Clark.。使用语言。剑桥大学出版社，英国剑桥。1996。[36] Herbert H. Clark and Susan E. Brennan. 1991. 沟通中的共同基础。美国心理学会，美国，127–149。https://doi.org/10.1037/ 10096-006[37] Bob Coecke and Aleks Kissinger. 2017. 《量子过程图示》。剑桥大学出版社，英国剑桥。[38] CommonMark. 2025. CommonMark：一份定义严谨、高度兼容的 Markdown 规范。https://commonmark.org/。访问日期：2025-08-05。[39] Andy Crabtree. 2025. H代表人类，以及如何（不）评估人机交互中的定性研究。《人机交互》0, 0 (2025), 1–24。https://doi.org/10.1080/07370024.2025.2475743 arXiv:https://doi.org/10 1080 07370024 2025 2475743 。/。//。设计委员会。无日[40] 期。双钻模型的历史。https: // www.designcouncil.org.uk/our-resources/the-double-diamond/history-of-the-double-diamond/。访问日期：2025-09-07。[41] Freeman J Dyson. 1949. 朝永振一郎、施温格和费曼的辐射理论。《物理评论》75, 3 (1949), 486。[42] Brendan Eich. 2011. 我的梦想之和谐。https://brendaneich.com/2011/01/harmony-of-my-dreams/ 访问日期：2025-09-07。[43] Brendan Eich. 2011. 我的 JSConf.US 演讲。https://brendaneich.com/2011/05/my-jsconf-us-presentation/ 访问日期：2025-09-07。[44] 莱昂哈德·欧拉。1755. 微分学原理。彼得罗波利，彼得罗波利。[45] Judith E Fan, Wilma A Bainbridge, Rebecca Chamberlain, and Jeffrey D Wammes. 2023. 绘图：一种多用途的认知工具。《自然评论：心理学》2, 9 (2023), 556–568。[46] Mirko Farina and Andrea Lavazza. 2025. 大语言模型中的英语：人工智能在避免文化同质化方面的作用。载于《牛津交叉领域：社会中的人工智能》。牛津大学出版社，英国牛津。[47] RP Feynman. 1949. 量子电动力学的时空方法。《物理评论》76, 6 (1949), 769。[48] Richard P Feynman. 1986. 量子力学计算机。《基础物理学》16, 6 (1986), 507–532。[49] Susan N Friel, Frances R Curcio, and George W Bright. 2001. 理解图表：影响理解的关键因素及其教学启示。《数学教育研究杂志》32, 2 (2001), 124–158。[50] Bruno Galantucci, Simon Garrod, and Gareth Roberts. 2012. 实验符号学。《语言与语言学指南》6, 8 (2012), 477–493。[51] Mirta Galesic and Rocio Garcia-Retamero. 2011. 图表素养：跨文化比较。《医学决策》31, 3 (2011), 444–457。[52] William W. Gaver. 1991. 技术可供性。载于《计算机系统中的人为因素国际会议论文集》（美国路易斯安那州新奥尔良）（CHI '91）。美国计算机协会，纽约，美国，79 84 10 1145 108844 108856 –。https://doi.org/。/。Dedre [53] Gentner and Christian Hoyos.。2017。类比与抽象。《认知科学专题》9, 3 (2017), 672–693。[54] Dedre Gentner and Arthur B Markman. 1997. 类比与相似性中的结构映射。《美国心理学家》52, 1 (1997), 45。[55] Thomas F. Gieryn. 1983. 边界工作与科学和非科学的划界：科学家专业意识形态中的张力与利益。《美国社会学评论》48, 6 (1983), 781–795。http://www.jstor.org/stable/ 2095325[56] Herman H. Goldstine and 约翰·冯·诺伊曼. 1947. 电子计算仪器问题的规划与编码。技术报告。普林斯顿高等研究院电子计算项目，普林斯顿，美国新泽西州。电子计算仪器的数学与逻辑方面报告，第二部分。[57] David Goodger and Guido van Rossum. 2001. PEP 257——文档字符串约定。https://peps.python.org/pep-0257/ 访问日期：2025-09-07。[58] 纳尔逊·古德曼。1976. 《艺术的语言：符号理论入门》。Hackett，美国印第安纳州印第安纳波利斯。[59] T. R. G. Green and A. F. Blackwell. 1998. 运用认知维度进行可用性设计。英国计算机学会人机交互会议（HCI '98）的教程环节。[60] John Gruber. 2004. Markdown。https://daringfireball.net/projects/markdown/。

[61] Ann Hutchinson Guest. 1970. 拉班舞谱，或拉班动觉记谱法：动作分析与记录系统。Theatre Arts Books，美国纽约。限量版。[62] Devamardeep Hayatpur, Brian Hempel, Kathy Chen, William Duan, Philip Guo, and Haijun Xia. 2024. 认真看待 ASCII 图画：程序员如何绘制代码图示。载于《2024年计算机系统中的人为因素国际会议论文集》（美国夏威夷州檀香山）（CHI '24）。美国计算机协会，纽约，美国，文章编号931，共16页。https://doi.org/10.1145/ 3613904.3642683[63] Devamardeep Hayatpur, Brian Hempel, Richard Lin, and Haijun Xia. 2025. 数据结构图中的抽象形态。载于《2025年计算机系统中的人为因素国际会议论文集》（ CHI '25）。美国计算机协会，纽约，美国，文章编号883，共12页。https: //doi.org/ 10 1145 3706598 3713723 。/。//。George E. Hein.。凯库勒与分子结构。[64] 1966[65] Edwin Hutchins. 1987. 界面设计隐喻。在北约主办的《包含语音的多模态对话》研讨会上发表的论文（法国科西嘉岛 Venaco，1986年9月）。Facebook Twitter教育部—教育统计研究所。[66] Tim Ingold. 2016. 线条：简史。Routledge，英国伦敦。[67] Kenneth E. Iverson. 1980. 记法：一种思想工具。《美国计算机协会通讯》23, 8（8月 1980 444 465 10 1145 358896 358899 ），–。[68] https://doi.org/。/。Robert J.K. Jacob, Audrey Girouard, Leanne M. Hirshfield, Michael S. Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. 2008. 基于现实的交互：后 WIMP 界面的框架。载于《计算机系统中的人为因素国际会议论文集》。美国佛罗伦萨，201– 210 10 1145 1357054 1357089 。[69] 2000 https://doi.org/。/。David Kaiser.。简笔画写实主义：费曼图的惯例、实体化与延续，1948–1964。《表征》70 (2000), 49–86。[70] David Kaiser. 2019. 将理论绘制开来：费曼图在战后物理学中的扩散。芝加哥大学出版社，美国伊利诺伊州芝加哥。[71] Smitha Keertipati, Sherlock A. Licorish, and Bastin Tony Roy Savarimuthu. 2016. 探索 Python 中的决策过程。载于《第20届软件工程评估与评价国际会议论文集》（爱尔兰利默里克）（EASE '16）。美国计算机协会，纽约，美国，文章编号 43, 10 10 1145 2915970 2915993 页数。[72] 1858 https://doi.org/。/。A. 凯库勒.。论化学化合物的结构与转变以及碳的化学性质。《化学年鉴》106卷第二册129 (1858), 106。[73] 奥古斯特·凯库勒。1865. 论芳香族物质的结构。《巴黎化学会月报》3 (1865), 98。[74] Steve Klabnik. 2016. Rust 的历史。载于 Applicative 2016（美国纽约）（Applicative 2016）。美国计算机协会，纽约，美国， 1 10 1145 2959689 2960081 页数。https://doi.org/。/。[75] 1956 S. C. Kleene.。《神经网络与有限自动机中的事件表示》。载于《自动机研究》，C. E. Shannon and J. McCarthy（编）。普林斯顿大学出版社，普林斯顿， 3 42 10 1515 9781400882618 002 –。[76] 2003 https://doi.org/。/ - Ursula Klein.。实验、模型、纸面工具：19世纪有机化学的文化。斯坦福大学出版社，美国加利福尼亚州斯坦福。[77] Albrecht Knust. 1959. 拉班动觉记谱法（拉班舞谱）导论。《国际民间音乐理事会杂志》11 (1959), 73–76。[78] DONALD E. KNUTH and LUIS TRABB PARDO. 1980. 编程语言的早期发展**本文的撰写部分得到了美国国家科学基金会资助（编号 MCS 72-03752 A03）、美国海军研究办公室合同（编号 N00014-76-C-0330）以及 IBM 公司的支持。作者感谢文中提及的各语言创始人对本文早期草稿提出的宝贵意见。忭转载自 J. Belzer、A. G. Holzman 和 A. Kent （编），《计算机科学与技术百科全书》第 6 卷，第 419–493 页。Dekker，纽约，1977年。经 Marcel Dekker 公司许可。载于《二十世纪计算史》，N. METROPOLIS、J. HOWLETT 和 GIAN-CARLO ROTA（编）。Academic Press，美国加利福尼亚州圣迭戈， 197 273 10 1016 978 0 12 491650 0 50019 8 –。https://doi.org/。/[79] 2000 B - - - -。- George Lakoff and Rafael Núñez.。《数学从何而来》。第6卷。纽约：Basic Books，美国纽约。[80] Bruno Latour. 2011. 将事物绘制在一起。载于《地图读本》。John Wiley & Sons, Ltd，美国新泽西州霍博肯， 65 72 10 1002 9780470979587 9 –。[81] 1994 https://doi.org/。/。ch Arthur Lewbel.。《杂技抛接记谱法的发明》（《学术杂耍者》）。《杂耍者世界》45 (1994), 34–35。[82] Gilbert N Lewis. 1916. 原子与分子。《美国化学会志》38, 4 (1916), 762–785。[83] Jingyi Li, Eric Rawn, Jacob Ritchie, Jasper Tran O'Leary, and Sean Follmer. 2023. 超越制品：以权力为视角的创意支持工具。载于《第36届 ACM 用户界面软件与技术研讨会论文集》（美国加利福尼亚州旧金山）（UIST '23）。美国计算机协会，纽约，美国，文章编号 47, 15 10 1145 3586183 3606831 页数。[84] 2023 https://doi.org/。/。Jingyi Li, Eric Rawn, Jacob Ritchie, Jasper Tran O'Leary, and Sean Follmer.。超越制品：以权力为视角的创意支持工具。载于《第36届 ACM 用户界面软件与技术研讨会论文集》（美国加利福尼亚州旧金山）（UIST '23）。美国计算机协会，美国纽约州纽约市，文章 47, 15 10 1145 3586183 3606831 页。[85] 2019 https://doi.org/ 。/ 。Cassi L. Liardét、Sharyn Black 和 Vani Sharren Bardetta。 。界定形式性：适应学术话语的抽象要求。《学术用途英语期刊》38（2019），146–158。https://doi.org/10.1016/j. jeap. 2019 02 007 。 。Lars Lilliestam。[86] 1996。凭耳演奏。《流行音乐》15，2（1996），195–216。[87] C. H. Lindsey。1996。ALGOL 68 的历史。美国计算机协会，美国纽约州纽约市， 27 96 10 1145 234286 1057810 – 。[88] 2023 https://doi.org/ 。/ 。Geoffrey Litt。 。大语言模型时代的可塑软件。[89] Emiliano Lorini 和 François Schwarzentruber。2011。一种关于反事实情绪推理的逻辑。《人工智能》175，3-4（2011），814–847。[90] Michael Lynch。1991。《空无一物的图像？社会理论中的视觉建构》。《社会学理论》 9, 1 1991 1 21 201870 （），–。[91] 1987 http://www.jstor.org/stable/ Vera Maletic。 。身体、空间、表达：鲁道夫·拉班动作与舞蹈概念的发展。第 75 卷。Walter de Gruyter，柏林和纽约。[92] Brian A Malloy 和 James F Power。2019。Python 2 向 Python 3 过渡的实证分析。《实证软件工程》24，2（2019），751–778。[93] Ference Marton。1981。现象图析学——描述我们对周遭世界的观念。《教学科学》10，2（1981），177–200。[94] Ference Marton。2014。《学习的必要条件》。Routledge，英国伦敦。[95] Damien Masson、Young-Ho Kim 和 Fanny Chevalier。2025。Textoshop：借鉴绘图软件交互以促进文本编辑。载于《2025 年计算机系统中的人为因素国际会议论文集》（CHI ' 25）。美国计算机协会，美国纽约州纽约市，文章 1087，14 页。https://doi.org/ 10 1145 3706598 3713862 。/ 。Damien Masson、Sylvain [96] Malacria、Géry Casiez 和 Daniel Vogel。2024。Direct-GPT：用于与大语言模型交互的直接操控界面。载于《计算机系统中的人为因素国际会议论文集》（美国夏威夷州檀香山）（CHI '24）。美国计算机协会，美国纽约州纽约市，文章 975, 16 10 1145 3613904 3642482 页。[97] 2004 https://doi.org/ 。/ 。Amy L McCarty。 。用于阅读和书写手语的记谱系统。《言语行为分析》20，1（2004），129–134。[98] Dick McCaw。2012。《拉班资料集》。Routledge，英国伦敦。[99] Dick McCaw 和 Pat Lehner。2023。《动作的艺术：鲁道夫·拉班未发表的著作》。Routledge，英国伦敦。[100] Warren S McCulloch 和 Walter Pitts。1943。神经活动中内在观念的逻辑演算。《数学生物物理学公报》5，4（1943），115–133。[101] Letitia Meynell。2008。费曼图为何能够表征。《科学哲学国际研究》22，1（2008），39–59。[102] Bryan Min、Allen Chen、Yining Cao 和 Haijun Xia。2025。《可塑总览 - 细节界面》。载于《 2025 年计算机系统中的人为因素国际会议论文集》（CHI '25）。美国计算机协会，美国纽约州纽约市，文章 688, 25 10 1145 3706598 3714164 页。[103] 2025 https://doi.org/ 。/ 。Bryan Min 和 Haijun Xia。 。Meridian：可塑总览-细节界面的设计框架。载于《第 38 届 ACM 用户界面软件与技术年度研讨会论文集》（UIST '25）。美国计算机协会，美国纽约州纽约市，文章 200，14 页。https://doi.org/10.1145/3746059.3747654[104] Bill Moggridge。2007。《设计交互》。麻省理工学院出版社，剑桥（马萨诸塞州）。[105] Tamara Munzner。2014。《可视化分析与设计》。Routledge，英国伦敦。[106] Peter Naur。1968。ALGOL 项目的成败。《ALGOL 公报》28（1968），58–62。[107] Dennis C. Neale 和 John M. Carroll。1997。第 20 章——隐喻在用户界面设计中的作用。载于《人机交互手册（第二版）》（第二版），Marting G. Helander、Thomas K. Landauer 和 Prasad V. Prabhu（编）。北荷兰出版社，阿姆斯特丹，441–462。https://doi.org/10.1016/B978-044481862-1.50086-8[108] Dennis C. Neale 和 John M. Carroll。1997。第 20 章——隐喻在用户界面设计中的作用。载于《人机交互手册（第二版）》，Marting G. Helander、Thomas K. Landauer 和 Prasad V. Prabhu（编）。北荷兰出版社，阿姆斯特丹，441–462。https://doi.org/10.1016/B978-044481862-1.50086-8[109] 卡尔加里大学。2025。SMILES 绘图工具。http://chem.ucalgary.ca/courses/350/smiles.html。卡尔加里大学，课程 350，访问日期：2025 年 9 月 2 日。[110] Charles Kay Ogden 和 Ivor Armstrong Richards。1927。《意义的意义：语言对思想的影响及符号学研究》。第 29 卷。K. Paul, Trench, Trubner & Company, Limited，伦敦和纽约。[111] 《牛津英语词典》。无日期。《牛津英语词典》。在线版。https://www.oed.com/dictionary/ 访问日期：2025 年 9 月 7 日。[112] Roger Penrose。1996。时空与宇宙学。《坦纳人类价值讲座》17（1996），347–391。[113] Roger Penrose 等。1971。负维张量的应用。《组合数学及其应用》1，221-244（1971），3。

[114] Trevor J Pinch 和 Wiebe E Bijker。1984。事实与人工制品的社会建构：或者，科学社会学与技术社会学如何能够互惠互利。《科学社会研究》14，3（1984），399–441。[115] Rebecca Platt 和 Nik Thompson。2015。统一建模语言的演化。载于《信息科学与技术百科全书》第三版。IGI Global Scientific Publishing，美国宾夕法尼亚州赫希，1931–1936。[116] William Playfair。1801。《商业与政治图集》（第 3 版）。T. Burton，英国伦敦。[117] Python 软件基金会。2025。Python 最初为何诞生？——Python 常见问题解答。https://docs.python.org/3/faq/general.html#why-was-python-created-in-the-first-place。[118] Panu Raattkainen。2005。论哥德尔不完备定理的哲学意义。《国际哲学评论》234，4（2005），513–534。[119] Maxime Rivest。2025。推文：“当下编程领域最令人难过的事，同时也是最令人兴奋的事。我希望我们能通过创新化解这种取舍。大语言模型在成熟的编程库和语言中提供编程辅助，令人无比振奋，也确实大幅提升了生产力。然而，如今语言和库之间的生产力差异，已不再主要取决于它们本身的优劣，而是取决于你的编程助手对它们了解多少。这给学习新语言和库带来了新的、实实在在的阻力。我现在主要使用 Python，原因是大语言模型更擅长 Python，并且默认使用它，而不是 R 或 Julia。大语言模型有望让传统上用于选择语言的库、文档和教程方面的论据几乎变得无关紧要。它们承诺能够生成文档、编写教程，甚至将库移植到其他语言。我坚信，它们已经足以让移植和编写文档变得至少 10× 更快——但我们需要着手去做。否则，我们可能永远困在 2020–2023 年的软件生态系统里，而我不希望任何人遭遇这种情况。”。https://x.com/MaximeRivest/status/1965531319636570362 访问日期：2025-09-10。[120] Everett M Rogers。1962。《创新的扩散》。Macmillan Publishing Co.，加拿大。[121] Nazmus Saquib、Rubaiat Habib Kazi、Li-yi Wei、Gloria Mark 和 Deb Roy。2021。通过草图构建具身代数。载于《2021 年计算机系统中的人为因素国际会议论文集》（日本横滨）（CHI ’21 428，16 10 1145 3411764）。美国计算机协会，美国纽约州纽约市，3445460 文章页数。https://doi.org/ 。/ 。[122] 2019 Floris Schuiling。。《记谱文化：迈向记谱的民族音乐学》。《皇家音乐协会期刊》144，2（2019），429–458。https://doi.org/10 1080 02690403 2019 1651508 。/ 。[123] Dana Scott。。回顾过去；展望未来。在纪念 Dana Scott 85 岁生日暨域理论五十周年研讨会、FLoC 2018 上的特邀演讲。2018。[124] Frank M Shipman 和 Catherine C Marshall。1999。形式化有害论：交互系统中使用形式化表征的经验、新兴主题与发展方向。《计算机支持的协同工作》8（1999），333–352。[125] Frank M Shipman III 和 Raymond J McCall。1999。基于超对象基底的渐进式形式化。《ACM 信息系统汇刊》17，2（1999），199–227。[126] Ben Shneiderman 和 Pattie Maes。1997。直接操控与界面代理。《Interactions》4，6（1997 年 11 月），42–61。https://doi.org/10.1145/267505.267514[127] SignWriting Channel。1995。聋人对 SignWriting 的观点。https://www.youtube.com/watch?v=bCevkVcMkg0。YouTube 视频。原载于 DVD《聋人对 SignWriting 的观点视频系列》。[128] Charlie Simpson。1986。《纸上杂耍》。《杂耍者世界》37，4（1986），31。[129] Theda Skocpol 和 Margaret Somers。1980。宏观社会研究中的比较历史用途。《社会与历史比较研究》22，2（1980），174–197。[130] DAVID CANFIELD SMITH。1975。Pygmalion：一种创造性编程环境。博士学位论文。斯坦福大学。[131] Robert Soden、David Ribes、Seyram Avle 和 Will Sutherland。2021。是时候在计算机支持的协同工作领域引入历史主义了：一份倡议。《ACM 人机交互论文集》5，CSCW 2，459 2021 18 10 1145 3479603 文章[132] （10 月），页数。https://doi.org/ 。/ Ian Spence 和 Howard Wainer。。William Playfair 与统计图表的发明。载于《信息设计》。Routledge，英国伦敦，59–76。[133] William C Stokoe Jr. 2005。手语结构：美国聋人的视觉传播系统概述。《聋人研究与聋人教育期刊》10，1（2005），3–37。[134] Valerie Sutton。1973。Sutton 运动速记：第 1 册：古典芭蕾符号键。Movement Shorthand Society，美国加利福尼亚州拉霍亚。[135] Valerie Sutton。2014。SignWriting 研讨会 2014 演讲 40：书写手语四十年。YouTube。https://youtu.be/sYQn6crcBno 访问日期：2025-09-07。[136] Valerie Sutton。2015。SignWriting 的历史。[137] Lev Tankelevitch、Elena L. Glassman、Jessica He、Majeed Kazemitabaar、Aniket Kittur、Mina Lee、Srishti Palani、Advait Sarkar、Gonzalo Ramos、Yvonne Rogers，以及 Hari Subramonyam。2025年。《思维工具：运用生成式人工智能理解、保护和增强人类认知的研究与设计》。载于《计算机系统中的人为因素国际会议（CHI EA ’25）扩展摘要集》。美国计算机协会，美国纽约州纽约市，文章 804, 8 10 1145 3706599 3706745 页。[138] https://doi.org/ 。/ 。Python核心团队与社区。。PEP – Python语言治理。2018 13 https://peps.python.org/pep-0013/ 访问日期：2025-09-07。[139] Bruce Tiemann 和 Bengt Magnusson。1991年。杂耍技巧的记法：大量杂耍技巧。《杂耍者世界》43卷2期（1991年），31–33。[140] Tommaso Toffoli。1981年。可逆组合函数的双连续扩张。《数学系统理论》14卷1期（1981年），13–23。[141] Barbara Tversky。2000年。地图与图表传达信息的一些方式。载于《空间认知 II：整合抽象理论、实证研究、形式方法与实际应用》。Springer-Verlag，柏林、海德堡，72–79。[142] Lukas Twist、Jie M. Zhang、Mark Harman、Don Syme、Joost Noppen 和 Detlef Nauck。2025年。《大语言模型偏爱 Python：关于大语言模型对编程语言和库偏好的研究》。arXiv:2503.17181 arXiv预印本。[143] Priyan Vaithilingam、Ian Arawjo 和 Elena L. Glassman。2024年。想象人工智能设计的未来：动态落地、建设性协商与可持续动机。载于《2024年ACM交互系统设计会议论文集》（丹麦哥本哈根）（DIS ’24）。美国计算机协会，美国纽约州纽约市，289–300。https://doi.org/10 1145 3643834 3661525 。/ 。[144] Priyan Vaithilingam、Elena L. Glassman、Jeevana Priya Inala 和 Chenglong Wang。2024年。DynaVis：用于可视化编辑的动态合成用户界面组件。载于《计算机系统中的人为因素国际会议论文集》（美国夏威夷州檀香山）（CHI ’24）。美国计算机协会，美国纽约州纽约市，文章 985，17 页。https://doi.org/10.1145/3613904. 3642639[145] Guido van Rossum。2018年。权力移交。致 python-committers邮件列表的电子邮件。https://www.mail-archive.com/python-committers@python.org/msg05628.html[146] John Venn。1880年。命题与推理的图解和机械表征。《伦敦、爱丁堡和都柏林哲学杂志与科学杂志》10卷59期（1880年），1–18。[147] Bill Venners。2003年。《Python的诞生：与 Guido van Rossum 的对话，第一部分》。https://www.artima.com/intv/pythonP.html。访谈。[148] Bret Victor。2022年。《2022年进展报告》（DynamicLand）。https://dynamicland.org/2022/Progress_report/ 2024年发布。[149] Bret Victor、Luke Iannini 和 Shawn Douglas。2024年。《共同体科学实验室》（DynamicLand）。https://dynamicland.org/2024/The_communal_science_ lab.pdf 2024年发布。[150] Visual Paradigm。无日期。《什么是统一建模语言（UML）？》
h
t
t
p
s://
www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/ 访问日期：2025-09-07。[151] William Von Hippel、John Jonides、James L Hilton 和 Sowmya Narayan。1993年。图式加工对感知编码的抑制作用。《人格与社会心理学杂志》64卷6期（1993年），921。[152] Jeff Walker 和 S Francisco。1982年。数字杂耍者的变化。《杂耍者世界》34卷1期（1982年），11。[153] Bailin Wang、Zi Wang、Xuezhi Wang、Yuan Cao、Rif A Saurous 和 Yoon Kim。2023年。借助大语言模型，通过语法提示生成领域特定语言。《神经信息处理系统进展》36（2023年），65030–65055。[154] Barry Warsaw、Jeremy Hylton、David Goodger 和 Alyssa Coghlan。2000年。PEP 1 – PEP的宗旨与指南。https://peps.python.org/pep-0001/ 访问日期：2025-09-07。[155] Victoria Watts。2015年。《贝内什动作记谱法与拉班舞谱：从创立到确立（1919–1977）》。《舞蹈编年史》38卷3期（2015年），275–304。[156] Alan Weir。2011年。《数学哲学中的形式主义》。https://plato.stanford.edu/entries/formalism-mathematics/。[157] William J Wiswesser。1985年。化学符号体系的历史发展。《化学信息与计算机科学杂志》25卷3期（1985年），258–263。[158] Aaron Sidney Wright。2012年。将无穷置于有限之处的优势：作为直觉对象的彭罗斯图。《自然科学史研究》44卷2期（2012年），99–139。[159] Aaron Sidney Wright。2013年。彭罗斯图在物理学、艺术和知觉心理学中的起源，1958–62年。《奋进》37卷3期（2013年），133–139。[160] Adrian Wüthrich。2010年。《费曼图的起源》。第26卷。施普林格科学与商业媒体，柏林和海德堡。[161] Haijun Xia、Bruno Araujo、Tovi Grossman 和 Daniel Wigdor。2016年。面向对象的绘图。载于《2016年计算机系统中的人为因素国际会议论文集》（美国加利福尼亚州圣何塞）（CHI ’16）。美国计算机协会，美国纽约州纽约市，4610–4621。https://doi.org/10. 1145/2858036.2858075[162] Jun Xing、Rubaiat Habib Kazi、Tovi Grossman、Li-Yi Wei、Jos Stam 和 George Fitzmaurice。2016年。能量画笔：用于描绘风格化元素动力学的交互工具。载于第29届用户年度研讨会论文集

界面软件与技术（日本东京）（*UIST '16*）。美国计算机协会，美国纽约州纽约市，755–766。https://doi.org/10.1145/ 2984511.2984585[163] Ryan Yen、Jian Zhao 和 Daniel Vogel。2025年。代码塑形：运用人工智能解读的草图绘制进行自由形式的迭代式代码编辑。载于《*2025*年计算机系统中的人为因素国际会议论文集（*CHI '25*）》。美国计算机协会，美国纽约州纽约市，文章 872，17 页。https://doi.org/ 10 1145 3706598 3713822 。/ 。[164] 2015 Sean Yom。。从方法论到实践：比较研究中的归纳迭代。《比较政治学研究》48卷5期（2015年），616–644。[165] J.D. Zamfirescu-Pereira、Eunice J un、Michael Terry、Qian Yang 和 Bjoern Hartmann。2025年。超越代码生成：大语言模型支持的程序设计空间探索。载于《*2025*年计算机系统中的人为因素国际会议论文集（*CHI '25*）》。美国计算机协会，美国纽约州纽约市，文章 153，17 页。https://doi.org/10.1145/3706598. 3714154[166] J. Jay Zeman。1977年。皮尔士符号学理论。载于《符号的灌注》。印第安纳大学出版社，美国印第安纳州布卢明顿，22–39。[167] Konrad Zuse。1993年。《计算机——我的一生》。施普林格科学与商业媒体，柏林和海德堡。

拆解“形式性”

尽管理解正文并不需要这些内容，但我们发现，批判性地反思形式性的社会建构以及“形式的”等术语很有帮助。这有助于我们思考，既定记法如何成为社会成就，而不仅仅是技术成就。

如今，“形式的”一词会附加在“语言”“模型”“语法”“证明”“系统”和“语义”等术语之后，广泛用于语言学、计算机科学、逻辑和数学等学科。在这种用法中，形式系统通常以借自数理逻辑的记法来定义，并分别设置句法（表征）和语义（意义）部分。这种“形式的”概念似乎源自弗雷格，后来又见于希尔伯特的形式主义计划 [85]。编程语言中的语义可以通过操作方式（即关于如何转换记法模式的规则）、公理方式或指称方式来定义。在设计领域，Shipman 和 Marshall 将“形式体系”定义为“描述并约束其工作或其内容某个方面的抽象表示”，这要求对用户想法进行“形式化”——用户必须学习一种形式记法，并将自己的想法编码其中，才能获得机器的权力。界面可能会“对用户强加……过高程度的形式性”，这可能是“有害的” [124, 第 334] 页。

然而，在计算领域使用之前、且与之无关，“formal”一词在其他含义上已有悠久历史。为了解这一点，我们整理了《牛津英语词典》、韦氏词典和 Etymonline 中“formality”“formal”“informal”和“formalization”等词的释义及词源用法，[12] 编列出以下清单：

F1 形式的内容是需要遵循的规则与程序（例如法律、仪式、典礼）。F2 形式的内容精准、明确且毫无歧义。F3 形式的内容由共同体共同确立（文化、社会、学科等）。F4 形式的内容传递权力，并由掌权者界定。F5 形式的内容僵化而严格，甚至“极其”如此。F6 形式的内容属于逻辑与理性的领域。

F7 形式作为去语义化：关注表征与句法，以至于忽视其底层意义，即“从一切意义中抽象出来” [85, 第 318] 页。

“Formal”一词约于公元1390年出现，当时用于描述“既定的”典礼、仪式和法律，例如“依据公认的形式，或依照艺术或法律的规则。”[13] 形式性 一词出现得较晚，约在17世纪，其定义为“使之形式化”和“符合既定规则”（《牛津英语词典》），并带有法律方面的联想。这些早期用法凸显出，形式的东西首先并非技术性的，而是社会性的：何为形式，是社会依据“既定规则”共同决定的。“Informal”一词约于16世纪中叶出现，意为“未依照公认或规定的形式完成或制作；不遵循既定程序或规则；非官方的；不规则的”，“不拘仪式”，“随意、轻松” [111]。“既定规则”和“仪式”等用法表明，新系统的形式性不能由普通个体决定，而只能由共同体决定。

另一个值得思考的方面是，形式的东西如何与内容形成张力（F7）。形式固然带有“精确性”以及已确立并遵循的规则系统等内涵，但也涉及这种精确性如何变得“过度僵化”或沦为“纯粹的仪式”，并以形式取代意义——形式有使意义消亡的风险。相比之下，非正式的东西“没有仪式”（注意这里使用了同一个词——仪式），并且重内容而轻形式：非正式的言语具有功能性价值，而非表演性价值。形式主义则“相应地轻视内容”，而偏重形式——关注表征而非概念——并被描述为“僵化或仅仅是对形式的惯例性遵守”、“严格或过度遵循规定形式”、“过度规整”或“刻板”，甚至有时“做这件事只是为了走形式”。英语习语“这不过是走个形式”体现了这种双重含义。在数学哲学中，形式主义是一种观点，认为证明中对符号记法进行类似游戏的操作就是数学的全部：“数学话语没有意义……数学其实是一种演算，其中‘空’符号串按照固定规则进行转换”[156]。然而，这种形式性观念——例如巴科斯-诺尔范式这样的形式文法——在解释方式上仍存在非正式之处。比如，像巴科斯-诺尔范式这样的“形式”文法，在表达式的解析顺序上存在歧义；对于句法中的关键符号（如 ::=）如何用于表达式，也存在歧义。任何形式系统都不可能既完备又一致 [118]。

形式化方格。我们的词源分析使我们注意到形式性的社会建构。我们可以得出结论：形式化可以通过以下两种方式之一进行：

第一种也是更常见的方式，是通过将非正式的、含糊的抽象翻译为共同体中已经得到认可的预先存在的记法，从而赋予其具体的形状和意义（例如，提示大语言模型生成 Python 代码）。

第二种更为神秘的形式化实现方式，是一种新记法经历认知、社会和物质层面的

[12]为使呈现更清晰，本节引用了这些来源中的各种释义，但未在文中逐一标注出处。词语的定义也可能相互参照；例如，formalization 被定义为“使之形式化”的过程：“赋予……形式或明确的形状；使……具有形式” [111]。

[13]https://www.oed.com/dictionary/formal_adj?tab=意义_and_use#3806070

过程，在此过程中，它不断打磨、完善，最终带着“形式的”认可印记脱颖而出：一种新的记号系统由此诞生。

我们可以借助一个“形式性的正方形”（图 1）来记住这一差异：它揭示了形式化是如何通过将一个想法翻译成既定的目标记号（水平动作），以及积极地进行新形式的社会建构（垂直动作），形成三角关系的。与水平动作不同，垂直动作无法借助既有系统的便利，而必须“赢得”被视为形式化的资格。

这两种“形式化运动”并非截然不同。为了提升理论能力，我们将它们区分开来；但如今，身处各种记法的包围之中，我们在创造一种新记法时，通常会同时进行水平和垂直运动。可以说，具体情境决定了所需“垂直”运动的程度：许多新记法似乎受到科学领域革命性进展的推动，尤其是出于驾驭新概念的需要 [80]。尽管这种抽象有所简化，但它或许能成为一种“思考工具”。

B 比较分析中考察的记法

在历史回顾中，我们考察了：乐谱记谱法；拉班舞谱和DanceWriting等舞蹈记谱法；费曼-戴森图；彭罗斯图；范畴量子力学；量子电路；早期编程记法；Python和JavaScript语言及其社群；Markdown；微手势记法 [30], ；从贝采利乌斯的分子式到Lewis、凯库勒等人后来提出的结构式等化学符号体系；SignWriting和Stokoe符号体系等书面手语记法；统一建模语言图；早期计算中的“神经网络”图 [75, 100],；Cajori经典综述中的数学记法 [26], ；杂耍记法；以及普莱费尔的数据可视化。表1将这些信息与我们使用的关键数据来源并列呈现，其中包括一手文献和已发表的历史记述。请注意，并非所有流行记法都有完整的历史记录，尤其缺少发明者最初构建这些记法过程的细节。

记谱法名称	发明者	二手资料来源	一手资料来源
乐谱记谱法	自古代以来的各种	Goodman [58], Ingold [66], 民族音乐研究-学 [86, 122]	
费曼–戴森图	理查德·费曼、弗里曼·戴森，此后数十年间的其他人	Kaiser [70]; [24, 101]	[41, 47]
彭罗斯图	Roger Penrose	Wright [158, 159]	[112]
量子领域中的字符串图力学	Bob Coecke（范畴量子力学），Roger Penrose（张量记法）		Coecke [37], 彭罗斯 [113]
量子电路	各种（例如费曼、Barenco 等人 al.)		托福利 [140], 费曼 [48], Barenco 等人 [9]
拉班舞谱	鲁道夫·拉班，后来在此基础上发展而成，并由 Ann Hutchinson Guest	[77, 91, 98, 99]	一些拉班原作再版于 [99]; [61]
SignWriting 与 DanceWriting	Valerie Sutton	[17]	Sutton 的传记史；官方视频，发布于 SignWriting Channel [127, 134–136]
早期编程记法	Konrad Zuse、约翰·冯·诺伊曼，赫尔曼·戈德斯廷、格蕾丝·霍珀，John Backus、早期 FORTRAN 和 ALGOL 团体，以及许多人	[3, 14, 78, 87]	[6, 7, 56, 67, 106, 167]
Python	Guido van Rossum		Python 文档和 Python 增强提案（例如, [2, 57, 138, 154]）；口头历tory [117, 147]; van Rossum 1991年左右的 USENET 帖子；van Rossum 的 “权力移交”公告 [145]
JavaScript ES6	Brendan Eich 等人		ECMAScript 标准；TC39 倡构想；Eich 的博客 [42, 43]
Markdown 和 CommonMark	John Gruber、Aaron Swartz；CommonMark 团队成员包括 John MacFarlane、Martin Woodward、Jeff Atwood		[5, 38, 60]
微手势符号	Adrien Chaffangeon Caillet、Alix Goguey 和 Laurence Nigay		[30]
化学分子式以及后来的结构式式	Jöns Jakob Berzelius、G. N. Lewis，奥古斯特·凯库勒等人	Klein [76], [64, 157]	[15, 16, 72, 73, 82]
Stokoe符号体系	威廉·斯托科	[97]	[133]
UML图	布奇、兰博和雅各布森（“三友”）	[115, 150]	[21]
“神经网”图表	麦卡洛克和皮茨；Kleene		[75, 100]
数学记号	自古代以来的各种	Cajori [26]	
杂耍记谱法	Claude Shannon 等人，主要出现在业余爱好者语境中	[81]	[13, 128, 139, 152]
折线图和散点图	William Playfair（有争议地）	[132]	[116]

表1：Su 所考察记法、其发明者以及参考的一手和二手资料概览 **s.**