arXiv:2602.13771v1 [cs.MM] 14 Feb 2026

SRA：语义 具备语义关系感知能力的流程图问答

Xinyu Li[1], *Bowei Zou*[2],*Yuchong Chen*[1],*Yifan Fan*[1],*Yu Hong*[1]*

[1]苏州大学计算机科学与技术学院，中国苏州[2]新加坡科技研究局资讯通信研究院，新加坡

摘要

流程图问答（FlowchartQA）是一项多模态任务，旨在根据图形流程图自动回答问题。当前研究将流程图转换为中间语言（如 Graphviz）以进行问答（QA），有效弥合了问题与流程图之间的模态差距。更重要的是，这些中间语言揭示了流程图中节点之间的链接关系，有助于在追踪答案时进行浅层关系推理。然而，现有中间语言仍未能体现复杂的语义／逻辑关系，例如“条件”和“因果关系”关系。这阻碍了对复杂问题进行深层推理。为解决这一问题，我们提出了一种新颖的语义关系感知（SRA）FlowchartQA 方法。该方法利用大语言模型（LLM）检测节点之间的语篇语义关系，从而将基于链接的中间语言升级为基于语义关系的中间语言。此外，我们采用中间语言可控推理过程。在此过程中，我们分析问题意图，以确定推理深度（浅层或深层推理），并选用与之匹配的中间语言。我们在基准数据集 FlowVQA 上进行了实验。测试结果表明，将 Graphviz、Mermaid 和 PlantUML 等不同中间语言升级后，SRA 均能带来广泛的性能提升。

索引词— 多模态，FlowchartQA

1. 引言

流程图是一种结合图示结构与自然语言的多模态数据，可用于说明复杂的过程、工作流和算法。从结构上看，流程图由功能节点和边组成，其中节点包含文本描述，而有向边则连接相关节点。图 1 展示了一个流程图示例。FlowchartQA [1, 2,3, 4] 旨在回答以流程图为条件的问题。由于该任务要求进行跨模态数据分析和语义推理，因此具有挑战性。

FlowchartQA 研究主要呈现出三种趋势，包括 1）端到端跨模态转换 [2, 3, 5],2）非结构化与结构化数据解耦 [6, 7], 以及 3）

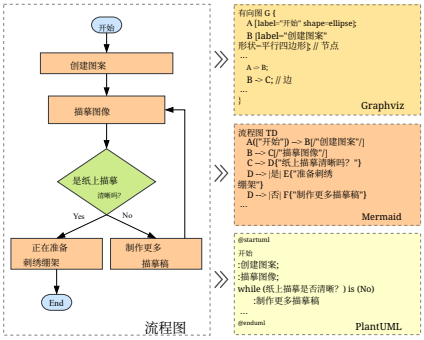


图 1。流程图与中间语言示例。

基于视觉语言模型（VLM）的流程图理解 [8, 9, 10, 11, 12, 13, 14, 15]。本文聚焦于第三种趋势。总体而言，基于 VLM 的方法会将流程图转换为定制的半结构化中间语言，例如 Mermaid[1]，Graphviz[2]，PlantUML[3]。随后，通过提示大语言模型理解中间语言的语义，并据此生成答案。这类方法显著提升了 FlowchartQA 的性能。

然而，在基于视觉语言模型的方法中，中间语言仅描绘节点之间的连接，因此只呈现一种刻板的链接关系——相关性（如图 1 所示）。换言之，它们无法体现节点具体内容之间生动的语义关系（例如因果关系和条件性）。因此，问答模型只能对数据进行浅层推理，而非深层推理。

为解决这一问题，我们提出 SRA，以增强对语义关系的感知能力。它通过用表意语义关系替换链接关系，改进现有的中间语言。我们使用大语言模型，根据节点的具体内容预测每对节点之间的语义关系。所有关系类型均取自典型语言学基准（Penn Discourse TreeBank 2.0 [16]）的话语关系分类体系，而非任意选取。此外，我们设计了一种中间语言可控推理过程，使系统能够根据问题意图，在浅层

* 通讯作者。

[1]Mermaid：https://mermaid.js.org/[2] Graphviz：https://graphviz.org/[3] PlantUML：https://Plantuml.com/

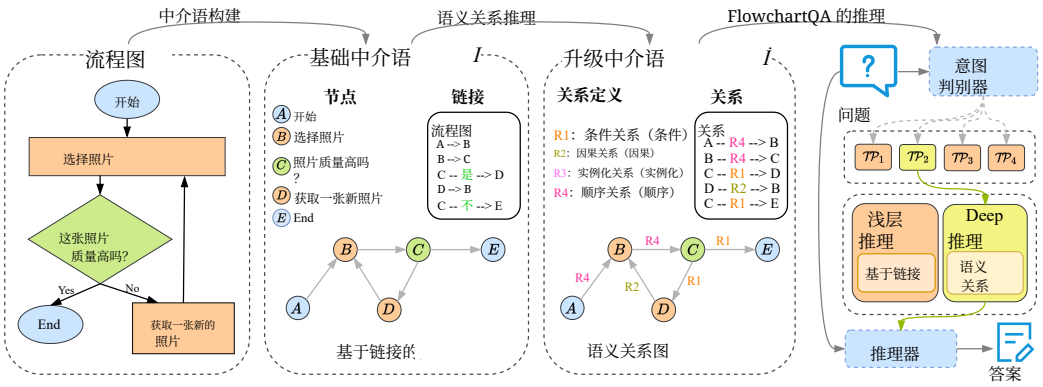


图 2。基于 SRA 的 FlowchartQA 架构。 $\mathcal{TP}_1$-$\mathcal{TP}_4$ 表示对应不同意图的不同问题类型。

推理（基于链接关系）与深层推理（基于语义关系）之间动态切换，以完成问答。

我们在 FlowVQA [2] 上进行了实验。测试结果证明了 SRA 的可行性：在采用不同中间语言以及使用不同规模的大语言模型作为骨干网络时，SRA 均能带来广泛的改进。在最佳情况下，其准确率约为 71.3%。

2. 方法

2.1. 基于 SRA 的 FlowchartQA 架构 e

我们通过 SRA 增强基线 FlowchartQA 模型。其架构如图 2 所示。基线模型 [8] 将流程图转换为中间语言 $\mathcal{I}$，并执行浅层推理 $f_s(*)$，以预测问题 Q的答案 $\mathcal{A}$。其形式化表示为 $\mathcal{A} = f_s(\mathcal{I}, \mathcal{Q})$。SRA 将中间语言 $\mathcal{I}$ 升级为具备语义关系感知能力的版本 $\mathcal{I}$，并执行深层推理 $f_d(*)$来预测答案。其形式化表示为 $\mathcal{A} = f_d(\mathcal{I}, \mathcal{Q})$。在 SRA 中，支持中间语言可控推理：根据问题意图，可以执行浅层推理 $f_s(*)$ 或深层推理 $f_d(*)$。实现细节如下。

2.2. 基础中间语言与浅层推理

基线方法在基于链接的中间语言上执行 FlowchartQA。具体而言，我们使用 TextFlow [8] 来分解流程图，将其划分为节点对。在分解过程中，TextFlow 中的视觉语言模型将流程图的视觉图像转换为可分析的结构化数据（例如 Mermaid 数据 [8]），其中展示了流程图中的所有节点、边和标签。所考虑的标签仅限于边上的原生符号“是”和“否”（如有），如图 2 所示。我们将此类结构化数据用作基础中介语 $\mathcal{I}$，它只揭示链接关系和简单的“与或非”逻辑。在实验中，我们评估了不同格式的结构化数据，包括 Mermaid 和 Graphviz [8]。

表1。语义关系分类。

条件性（定义）：节点 O_B 描述一个结果这取决于节点 O_A中提出的条件。 示例： O_A– 完成作业？ O_B– 打破关系 O_A 和 O_B之间– 条件性
因果关系（定义）：节点 O_A 描述导致节点 O_B中结果的因果关系或动作。 示例： O_A–获取一张新照片 O_B–选择照片 O_A 与 O_B之间的关系– 因果关系
实例化（定义）：节点 O_B 提供一个具体的实例或案例，用以说明节点 O_A中的一般概念。 示例： O_A– 柑橘类水果 O_B– 橙子和葡萄柚 O_A 与 O_B之间的关系– 实例化
时序关系（定义）：节点 O_B 描述一个事件，该事件在节点 O_A之后按时间顺序发生。 示例： O_A– 混合面粉和水 O_B– 揉捏混合物-成面团 O_A 和 O_B之间的关系 – 时序关系

使用大语言模型 Qwen-2.5 [17] 推断答案 $\mathcal{A}$，并将问题 $\mathcal{Q}$ 和中间语言 $\mathcal{I}$ 打包到查询中作为输入。Qwen-2.5 在零样本设置下运行，且没有提供明确指令来引导模型生成特定任务的思维链（CoT）。因此，Qwen-2.5 基于浅层的节点级关系知识（即链接和简单逻辑），按常规方式进行关系推理。

2.3. 基于语义关系的跨语言 e

2.3.1. 关系分类体系

我们为中间语言升级构建了一套定制的语义关系分类体系。该分类体系包含四种关系类型，包括条件、因果关系、实例化关系和顺序关系。表1列出了它们的定义和示例。所有这些类型均源自宾夕法尼亚话语树库2.0（PDTB v2）完整的话语关系分类体系 [16]，该分类体系最初用于自动分析句子级文本之间的语义关系。

请注意，PDTB实际上包含4个主要类别，共11种语义关系类型。本研究禁用了其中大多数类型（上述四种除外）。原因有二：1）被禁用的关系与流程图节点之间的交互逻辑不兼容；2）因此，它们会成为升级后的中间语言中的噪声。我们使用Deepseek-R1 [18] 验证兼容性。用于验证的节点对有1K组，均从训练集中随机选取。

2.3.2. 语义关系识别与升级

我们的 SRA 根据上述关系分类体系构建了一种具备语义关系感知能力的中间语言$\mathcal{I}$。具体做法是升级基础中间语言 $\mathcal{I}$，将每对节点之间的链接关系标签替换为其语义关系标签。关系识别用于标签重新标注。

具体而言，对于$\mathcal{I}$中的每对节点，我们使用 Deepseek- R1 [18] 识别它们之间的语义关系。标签重新标注同步进行。在关系识别过程中，我们使用特定任务指令引导 Deepseek-R1 进行两阶段思维链（CoT）推理。首先，引导其读取节点内容，并据此分析它们之间的语义关系。其次，根据分析结果，从分类体系中确定最匹配的关系标签。所有关系定义（表1）均与基础中间语言 $\mathcal{I}$一同提供在输入查询中。输出限定为节点 d 与关系 r组成的三元组 $\mathcal{T}$，即 $\mathcal{T}_{ij}$=$\{n_i, r_{ij}, n_j\}$。

2.3.3. 深层推理

使用升级后的语言 $\mathcal{I}$，提示 Qwen-2.5 执行深层推理，并在查询中提供更深层知识。具体而言，提供以关系为中心的三元组 $\mathcal{T}$，以展示多样化的关联，同时由分类体系阐释关系的具体含义。与浅层推理类似，深层推理采用零样本设置，也省略了思维链。不过，这里使用了严格的约束，要求 Qwen-2.5 在生成答案之前充分理解三元组 $\mathcal{T}$、分类体系和问题。

2.4. 中间语言可控推理

FlowchartQA 处理不同类型的问题。其中一些问题较为简单，无需或几乎无需通过推理来解决复杂关系。例如，在基准 FlowVQA[2], 的四种问题类型（$\mathcal{TP}_{1-4}$）中，只有“应用场景”（$\mathcal{TP}_2$）这一类型严重依赖深层关系分析（例如动作的条件关系或时序关系）。相比之下，“拓扑”（$\mathcal{TP}_4$）这一类型只关注结构信息（例如，参见问题“图表中有多少个节点？”），与关系推理毫无关系。

对这些直接问题应用深层推理（第 2.3.3 节）会造成不必要的计算资源和时间消耗。因此，我们开发了一种可控中间语言的推理方法。该方法使用基于大语言模型的判别器来确定问题类型，并输出“简单”或“复杂”问题标签。对于简单问题，该方法会调用大语言模型，使用基本中间语言进行浅层推理 $\mathcal{I}$（第 2.2 节）。对于复杂问题，则使用升级中间语言进行深层推理 $\dot{\mathcal{I}}$（第 2.3.3 节）。在实现过程中，使用 LLaMA-3（8B） [19]作为判别器。该判别器在从 FlowVQA 训练集中随机选取的 2K 个实例上进行微调。

表2。测试集上的比较结果（准确率. %）。

骨干网络	方法	总体	独立测试集			
			$\mathcal{TP}_1$	$\mathcal{TP}_2$	$\mathcal{TP}_3$	$\mathcal{TP}_4$
GPT-4o	图像	63.7	69.5	65.6	68.8	57.9
Qwen-2.5 (72B)	Graphviz	68.7	79.8	63.6	72.5	64.2
	+ SRA	**71.3**	83.5	73.1	72.0	64.4
	Mermaid	69.5	83.8	68.1	73.1	62.0
	+ SRA	70.1	84.5	72.9	71.7	61.2
	PlantUML	57.3	79.8	63.6	70.2	38.1
	+ SRA	60.0	82.0	72.9	68.8	39.6
Qwen-2.5（7B）	Graphviz	59.4	78.8	59.3	54.6	52.0
	+ SRA	61.0	81.3	65.1	54.3	52.0
	Mermaid	60.6	80.0	60.3	56.4	53.1
	+ SRA	**62.0**	80.8	66.1	56.7	53.2
	PlantUML	53.0	82.8	57.8	57.8	34.4
	+ SRA	54.7	83.3	65.6	57.5	34.5

3. 实验

3.1. 数据集与评估

我们在 FlowVQA [2]上进行实验。FlowVQA 中有 12,938 个实例可用于训练，另有 9,475 个实例用于测试。为便于比较，我们遵循先前工作 [8] 的做法，使用 1,978 个实例构建测试集，这些实例是来自 197 个流程图的问答对。准确率（准确率）被用作评估指标。此外，通过将预测结果与真实数据进行比较，使用 GPT-4o 确定正例和负例 [2, 3, 8]。这被视为一种用于正确性验证的语义匹配方法。

3.2. 主要结果

我们将 SRA 与基于图像的 FlowchartQA [8], 进行比较，其中 GPT-4o 用于图像转文本。我们还与三种基于中间语言的基线方法进行比较。它们均使用 Qwen-2.5（7B/72B）作为骨干网络，但基于不同的中间语言推断答案，包括 *Graphviz*、*Mermaid* [20] 和 *Plan-tuml* [21]。请注意，Graphviz 和 Mermaid 用作基于链接的结构化数据（第 2.2 节）。Plantuml 是一种半结构化语言，其表现形式是包含执行逻辑的程序。

表2展示了比较结果，包括在准确率方面的综合表现，以及在**整体**测试集上的表现

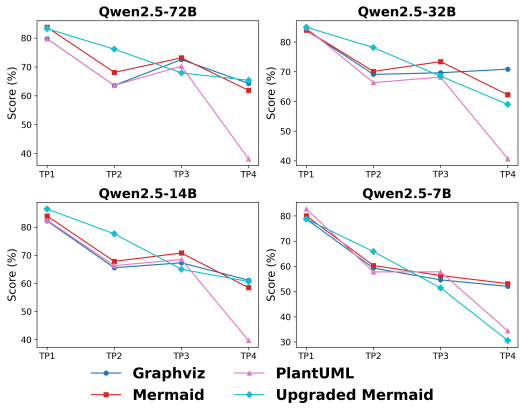


图3. 多语言单向验证。

以及不同问题类型子集上的局部表现。这些类型包括事实检索($\mathcal{TP}_1$)、应用场景($\mathcal{TP}_2$)、流程引用($\mathcal{TP}_3$)和拓扑($\mathcal{TP}_4$)。从表2可以看出，无论采用哪种骨干网络，SRA的综合表现都优于基线方法。另一个发现是，SRA在可解释性问题($\mathcal{TP}_2$)上取得了最显著的提升。与*Graphviz*相比，最高提升可达9.5%。不过，SRA在其他三种类型上的表现也具有可比性。

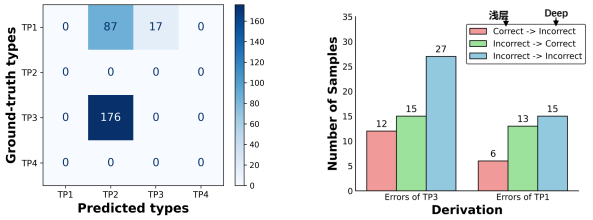
3.3. 可控跨语言设置的必要性

SRA 在上述四种问题类型（$\mathcal{TP}_{1-4}$）上的局部表现（见表 2）似乎证明了中间语言可控推理的必要性。然而，这种验证是间接的，因为控制是由自动判别器执行的。这促使我们进一步考察伪正样本是否可能带来良好的结果。

为获取直接证据，我们禁用了中间语言可控推理模块。在此基础上，针对单一问题类型分别独立运行 FlowchartQA，并使用不同的中间语言（Graphviz、Mermaid、PlantUML 和升级版 Mermaid）。由此可以进行多语言单向验证。图3展示了各问题类型的验证结果，其中使用了不同版本的 Qwen-2.5。可以观察到，对于 $\mathcal{TP}_2$，采用升级后的中间语言总能带来显著更高的性能（图3中的蓝线）。相反，对于其他三种树形问题类型，得到的性能相近或不稳定；第2.4节将它们归类为直接问题。验证结果直接证明了可控跨语言设置的必要性。

3.4 判别器错误分析

SRA 中的问题类型判别是一项中间操作，在中间语言——



(a) 错误数量　　(b) 错误分析

图4。错误分析（使用 Qwen-2.5-72B 和 Graphviz）。

可控推理过程之前进行。因此，它产生的错误问题类型会误导后续的浅层或深层推理。实际上，一项单独的实验表明，判别器在测试集上的准确率为 85.8%。14.2% 的错误所带来的负面影响不容忽视。

为清楚了解其影响，我们考察了四种问题类型的错误分布，并将测试集中的所有实例纳入分析。图4(a)展示了错误分布，其中每个热图块表示错误数量。结果表明，主要错误来自 $\mathcal{TP}_1$ 和 $\mathcal{TP}_3$ 问题，其中一些被误判为 $\mathcal{TP}_2$。考虑到在可控过程中，$\mathcal{TP}_2$ 通过深层推理处理，我们进一步验证了深层推理对这些误判问题的准确率，并将其与浅层推理进行比较。图4(b)中的子图(b)展示了比较结果。结果表明，一些误判问题确实能被浅层推理正确回答，却被深层推理错误回答（红色柱状图）。相比之下，反向情况更为明显（绿色柱状图）。这证实了深层推理具备一定的容错能力。

3.5 大语言模型对推理的影响

基于大语言模型的语义关系生成是跨语言升级的核心。我们验证了两个大语言模型（DeepSeek-R1 和 GPT-4o）的可靠性，并对随机选取的 500 个样本进行了人工检查。结果表明，DeepSeek-R1 的表现优于 GPT-4o（准确率分别为 86.2% 和 80.0%）。

4. 结论

为增强FlowchartQA，我们提出利用语义关系升级结构化中间语言，旨在提高推理深度。此外，我们开发了一种跨语言可控推理方法，可针对不同问题类型选择性地调用浅层推理与深层推理。实验证明了我们方法的有效性。未来，我们将研究自包含知识增强方法，检测图中的隐性知识以实现全面理解，例如主题和多跳隐式关系。

5. 参考文献

[1] Simon Tannert、Marcelo G. Feighelstein、Jasmina Bogojeska、Joseph Shtok、Assaf Arbelle、Peter W. J. Staar、Anika Schumann、Jonas Kuhn和Leonid Karlinsky, “FlowchartQA：首个面向流程图推理的大规模基准”，载于 *LIMO*，2023，第34–46页。

[2]Shubhankar Singh、Purvi Chaurasia、Yerram Varun、Pranshu Pandya、Vatsal Gupta、Vivek Gupta 和 Dan Roth，“FlowVQA：利用流程图映射视觉问答中的多模态逻辑”，发表于*ACL*，2024年，第1330–1350页。

[3] Huitong Pan、Qi Zhang、Cornelia Caragea、Eduard C. Dragut 和 Longin Jan Latecki，“Flowlearn：评估大型视觉语言模型对流程图的理解能力”，发表于*ECAI*，2024年，第73–80页。

[4] Enming Zhang、Ruobing Yao、Huanyong Liu、Junhui Yu 和 Jiale Wang，“对多模态大型语言模型流程图理解能力的首次多维度评估”，发表于*ArXiv*，2024年。

[5] Pan Lu、Liang Qiu、Jiaqi Chen、Tanglin Xia、Yizhou Zhao、Wei Zhang、Zhou Yu、Xiaodan Liang 和 Song-Chun Zhu，“Iconqa：用于抽象图表理解和视觉语言推理的新基准”，发表于*NeurIPS*，2021年。

[6] Abdul Arbaz、Heng Fan、Junhua Ding、Meikang Qiu 和 Yunhe Feng，《Genflowchart：使用生成式人工智能解析和理解流程图》，载于 *KSEM*，2024年，第 99–111 页。

[7] Takamitsu Omasa、Ryo Koshihara 和 Masumi Morishige，《Arrow-guidedvlm：通过箭头方向编码增强流程图理解》，载于 *ArXiv*，2025年。

[8]Junyi Ye、Ankan Dash、Wenpeng Yin 和 Guiling Wang，《超越端到端视觉语言模型：利用中间文本表示实现卓越的流程图理解》，载于 *NAACL*，2025年，第 3534–3548 页。

[9] Peng Wang、Shuai Bai、Sinan Tan、Shijie Wang、Zhi-hao Fan、Jinze Bai、Keqin Chen、Xuejing Liu、Jialin Wang、WenbinGe 等，《Qwen2-vl：增强视觉语言模型在任意分辨率下对世界的感知能力》，载于 *ArXiv*，2024年。

[10] Haotian Liu、Chunyuan Li、Yuheng Li 和 Yong Jae Lee，《通过视觉指令微调改进基线》，发表于计算机视觉与模式识别会议，2024年，第 26286–26296 页。

[11] Wenyi Hong、Weihan Wang、QingsongLv、JiazhengXu、Wenmeng Yu、Junhui Ji、Yan Wang、Zihan Wang、Yux- iao Dong、Ming Ding 等人，“Cogagent：面向图形用户界面智能体的视觉语言模型”，发表于计算机视觉与模式识别会议，2024年，第14281–14290 页。

[12] Xiaoyi Dong、Pan Zhang、Yuhang Zang、Yuhang Cao、Bin Wang、Linke Ouyang、Xilin Wei、Songyang Zhang、Haodong Duan、Maosong Cao 等人，“Internlm-xcomposer2：掌握视觉语言大模型中的自由形式图文组合与理解”，发表于 *ArXiv*，2024年。

[13] Gemini 团队、Rohan Anil、Sebastian Borgeaud、Jean-Baptiste Alayrac、Jiahui Yu、Radu Soricut、Johan Schalkwyk、Andrew M Dai、Anja Hauth、Katie Millican 等，《Gemini：一系列能力卓越的多模态模型》，发表于 *ArXiv*，2023。

[14]Anthropic.，《推出 Claude 3.5 Sonnet，2024》，https://www.anthropic.com/news/claude-3-5-sonnet，2024 年。

[15] Aaron Hurst、Adam Lerer、Adam P Goucher、Adam Perelman、Aditya Ramesh、AidanClark、AJ Ostrow、Akila Welihinda、Alan Hayes、Alec Radford 等，《GPT-4o 系统卡》，发表于 *ArXiv*，2024年。

[16] Rashmi Prasad、Nikhil Dinesh、Alan Lee、Eleni Miltsakaki、Livio Robaldo、Aravind K. Joshi 和 Bonnie L. Webber，《宾夕法尼亚话语树库2.0》，发表于语言资源与评测会议，2008年。

[17] Qwen 团队，《Qwen2.5 技术报告》，发表于 *ArXiv*，2024年。

[18] Daya Guo、Dejian Yang、Haowei Zhang、Junxiao Song、Ruoyu Zhang、Runxin Xu、Qihao Zhu、Shirong Ma、Peiyi Wang、Xiao Bi 等，《DeepSeek-R1：通过强化学习激励大语言模型的推理能力》，发表于 *ArXiv*，2025年。

[19] Abhimanyu Dubey、Abhinav Jauhri、Abhinav Pandey、Abhishek Kadian、Ahmad Al-Dahle、Aiesha Letman、Akhil Mathur、Alan Schelten、Amy Yang、Angela Fan 等，《Llama 3 模型家族》，发表于 *ArXiv*，2024年。

[20] Jiaye Wang，《引导大语言模型生成计算机可解析内容》，发表于 *ArXiv*，2024年。

[21] Felix H¨arer，《面向大语言模型的概念模型解释器》，发表于*ER*，2023。