

# LINKGPT: 教授大规模语言模型预测缺失链路

Zhongmou He, Jing Zhu, Shengyi Qian, Joyce Chai, Danai Koutra

## 摘要

利用 LLMs 预测图中节点之间缺失的链路面临两个主要挑战：（1）如何有效地将图的结构信息集成到 LLMs 中，这对链路预测性能至关重要；（2）如何克服在训练 LLMs 进行链路预测时的计算瓶颈。为解决这些挑战，提出了 LINKGPT，这是首个为链路预测任务端到端训练的 LLM。为有效提升 LLMs 对图结构的理解能力，本文设计了两阶段的指令微调方法：第一阶段微调对偶编码器、投影器和节点投影器，第二阶段进一步微调 LLMs 以进行链路预测。为提高推理效率，本文引入了检索-重排序方案。实验结果表明，LINKGPT 在多个真实世界图数据集上实现了最先进的性能，并在零样本和少样本学习中展现出优异的泛化能力，超越了现有的基准方法。

**关键词：**大语言模型；链路预测

## 1 引言

利用大规模语言模型（LLMs）进行链路预测（LP）并非易事，面临两个主要挑战。首先，LLMs 主要设计用于处理文本和序列数据，如何将节点级别和对偶结构信息有效集成到 LLMs 中是一个关键问题。其次，在训练 LLMs 进行链路预测时，存在显著的计算瓶颈。链路预测通常需要在推理阶段对每个链接与大量负样本（例如，1,000 个）进行排名，这在与 LLMs 结合时会导致较高的计算成本。如何高效且准确地对候选项进行排序是基于 LLM 的链路预测方法实用性的关键所在。为了解决这些挑战，本文提出了 LINKGPT——首个专门设计用于预测图中缺失链路的端到端训练的 LLM。LINKGPT 由三个主要组件组成：（1）节点编码和对偶编码；（2）两阶段指令微调；（3）推理时的检索-重排序方案。实验结果表明，LINKGPT 不仅在多个数据集上达到了最先进的性能，还在零样本和少样本的上下文学习设置中展现出强大的泛化能力，超越了现有的基准方法。采用设计的检索-重排序方案进行推理时。该工作的主要贡献如下：

- **首个专为链路预测设计的 LLM：**提出了首个链路预测方法，训练 LLMs 在图中预测缺失链路，使 LLMs 能够对结构化数据进行推理。
- **高效且有效的方法：**提出了 LINKGPT 框架，通过指令微调有效地将节点级和对偶结构知识融入 LLMs，并通过检索-重排序方案显著提高推理时间效率。
- **广泛的评估：**在多个真实世界图数据集上进行了广泛实验，展示了 LINKGPT 相较于现有最先进方法的优越性能和快速推理能力。此外，还展示了 LINKGPT 在跨类别和跨领域未见数据集上的零样本泛化能力和少样本上下文学习能力。

## 2 相关工作

### 2.1 链路预测

链路预测的目标是补全图中缺失的链路，应用范围涵盖知识图谱补全、电子商务推荐等领域。尽管启发式算法曾经占据主导地位，近年来，基于图神经网络（GNN）的链路预测方法逐渐兴起。基于

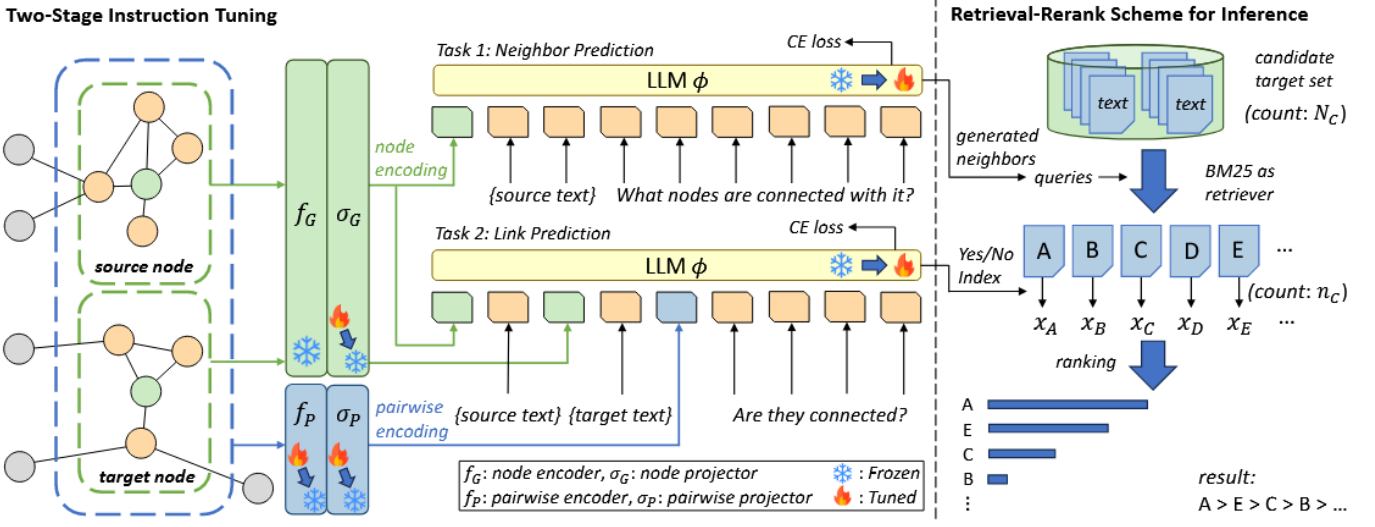


图 1: LINKGPT 框架示意图

GNN 的链路预测方法主要分为两类：基于图自编码器（GAE）的方法和基于封闭子图的方法。尽管 GNN 方法展示了优异的性能，但其在未见过的图数据上的泛化能力仍有限。本研究旨在探索大规模语言模型（LLMs）在链路预测任务中的结构推理能力，并评估其在零样本和少样本设置下的泛化性能。

## 2.2 大规模语言模型在图中的应用

近期，LLMs 在文本属性图（TAGs）上的应用取得了显著进展，旨在利用 LLMs 的强大能力提升图相关任务的性能。主要有两种策略：（1）将 LLMs 作为预测器，直接生成节点分类和链路预测等任务的解决方案；（2）将 LLMs 作为增强器，利用其能力提升较小的 GNN 所学习的表示，以提高效率。LINKGPT 直接预测图中的缺失链路，属于第一类方法。尽管以往研究主要集中于提升 LLMs 对节点标签的预测能力，但认为 LLMs 在图中预测缺失链路的能力更能展示其真正的结构推理能力。

## 2.3 指令微调

指令微调作为一种前沿方法，已被应用于将 LLMs 适应于各种任务和模态。近期研究在视觉-语言模型和多模态学习中探索了指令微调。指令微调亦已被应用于多个基于 LLM 的图任务模型。本研究进一步探讨了指令微调在结构-语义理解中的应用。

# 3 方法

本文复现了 LINKGPT 框架。LINKGPT 利用大型语言模型（LLMs）的能力，有效捕捉图结构和文本语义，从而实现准确且高效的链路预测。该框架由三个主要部分组成：（1）节点编码和成对编码；（2）两阶段指令微调；（3）推理时的检索-重排序方案。具体示意图如图 1 所示。

## 3.1 节点编码和成对编码

为有效捕捉图结构和文本语义，首先引入两个特殊标记：<NODE> 和 <PAIRWISE>。<NODE> 标记表示节点及其邻域，<PAIRWISE> 标记编码两个节点之间的成对关系。邻域感知的节点编码通过对比图-文本预训练（Contrastive Graph-Text Pre-training, CGTP）获得节点嵌入。设  $f_G$  和  $f_T$  分别表示节点编码器和文本编码器，节点表示  $\mathbf{H} \in \mathbb{R}^{N \times d_H}$  和邻域文本表示  $\mathbf{T} \in \mathbb{R}^{N \times 2d_T}$  的获取方式如下：

$$\mathbf{T}' = f_T(\mathcal{X}), \quad \mathbf{T}_v = \text{concat} \left( \mathbf{T}'_v, \frac{1}{|\mathcal{N}_v^1|} \sum_{u \in \mathcal{N}_v^1} \mathbf{T}'_u \right), \quad \mathbf{H} = f_G(\mathcal{V}, \mathcal{E}, \mathcal{X}) \quad (1)$$

其中,  $\mathbf{T}'_v \in \mathbb{R}^{d_T}$  为节点  $v$  的原始文本编码。本文采用 GraphFormers<sup>[1]</sup>作为节点编码器  $f_G$ , 该编码器将文本编码与图聚合融合到统一的迭代工作流程中, 并通过图-文本对比预训练进行优化。使用 BERT<sup>[2]</sup>作为文本编码器  $f_T$ 。对比损失函数  $\mathcal{L}$  的计算公式如下:

$$\Gamma = \hat{\mathbf{H}}\hat{\mathbf{T}}^T/\tau, \quad \mathcal{L} = (\text{CE}(\Gamma, \mathbf{y}) + \text{CE}(\Gamma^T, \mathbf{y}))/2 \quad (2)$$

其中,  $\hat{\mathbf{T}} = \text{norm}(\mathbf{T})$ ,  $\hat{\mathbf{H}} = \text{norm}(\mathbf{H})$ ,  $\tau$  为温度系数, 标签  $\mathbf{y}$  设为  $(0, 1, \dots, N-1)^T$ 。学习到的节点编码  $\mathbf{H}$  通过对齐投影器  $\sigma_G$  映射到 LLM 的语义空间。成对编码为捕捉节点间的成对关系, 采用 LPFormer。对于节点  $a$  和  $b$ , LPFormer 学习一个成对编码  $f_P(a, b)$  来表示它们之间的关系:

$$f_P(a, b) = \sum_{u \in \mathcal{V}(a, b)} w(a, b, u) \odot h(a, b, u) \quad (3)$$

其中,  $w(a, b, u)$  和  $h(a, b, u)$  分别表示节点  $u$  相对于节点  $a$  和  $b$  关系的重要性和编码,  $\mathcal{V}(a, b)$  表示可能对关系  $a$  和  $b$  有重要作用的节点集合。  $f_P(a, b)$  同样通过对齐投影器  $\sigma_P$  映射到 LLM 的语义空间。需要注意的是, 成对编码器未经过预训练。

### 3.2 两阶段指令微调

采用两阶段的指令微调方法来训练 LINKGPT, 用于链路预测和邻居预测任务。这些任务的指令模板如图 1所示。

**提示设计 (Prompt design):** 对于邻居预测任务, 模型根据节点的编码和文本, 直接生成该节点邻居的文本。此任务的设计旨在进一步对齐节点编码与其邻域, 并使模型能够直接生成某节点的邻居, 从而加速推理过程。

对于链路预测任务, 模型需要根据节点的文本、节点编码以及它们之间的成对编码, 预测两个节点是否相连。每条指令包含一个源节点和多个候选目标节点, 其中正样本和负样本的数量相等。此设置使模型能够在上下文中比较当前候选节点与其他候选节点, 增强模型在推理时的上下文学习能力。

**训练计划 (Training schedule):** 采用两阶段的微调策略。在第一阶段, 重点训练成对编码器以及将特殊编码与预训练 LLM 的词嵌入对齐。此阶段, LLM 保持冻结, 仅训练与编码相关的模块, 包括成对编码器和两个对齐投影器。在第二阶段, 冻结编码相关模块, 通过 LoRA 微调 LLM, 以提升模型对编码的理解能力。

### 3.3 检索-重排序推理机制

在链路预测的推理阶段, 需要对每个源节点  $s$  的候选目标节点集合  $C = \{t_1, t_2, \dots, t_{N_C}\}$  进行排序。然而, 直接对候选集合  $C$  中的所有节点使用 LLM 进行排序可能会导致较大的计算开销, 尤其当  $C$  较大时。为解决此问题, 提出了一种检索-重排序方案, 通过高效缩小候选集合的范围, 再对最终排序应用 LLM。

在检索阶段, 的目标是快速识别一个更小的候选目标节点子集  $n_C \ll N_C$ , 这些节点最有可能与源节点连接。受 GPT4Rec<sup>[3]</sup>启发, 提示 LLM 直接生成与源节点潜在邻居相关的文本。这些生成的文本随后被用作查询, 通过 BM25 算法<sup>[4]</sup>从候选集合  $C$  中搜索相关节点。为了确保生成文本的多样性, 采用带有多重惩罚的束搜索方法。此外, 应用基于距离的分组策略, 从源节点的 2 跳邻居  $N_s^{2'}$  中选取  $\beta n_C$  个节点, 以及从图中其余部分  $V \setminus N_s^{2'}$  中选取  $(1 - \beta)n_C$  个具有最高 BM25 分数的节点。在实验中, 设置  $\beta = 0.65$ 。

在重排序阶段，使用 LLM 对检索到的  $n_C$  个候选节点进行排序，得到最终的链路预测结果。为了量化源节点  $s$  与候选目标节点  $t_i$  之间存在边的概率，引入了 “Yes/No index”：

$$\text{Yes/No index} = \frac{p(\text{“Yes”} \mid \text{context})}{p(\text{“Yes”} \mid \text{context}) + p(\text{“No”} \mid \text{context})} \quad (4)$$

其中，概率通过对 LLM 的 logits 应用 softmax 函数计算得到。

为了利用 LLM 的上下文学习能力，在提示的上下文中包含源节点的邻居，其中正样本从训练集中的源节点邻居中随机选择，负样本从  $V \setminus N_s'$  中随机选择。由于所有候选目标节点共享相同的源节点上下文和示例，通过使用键值对的共享来降低推理时间复杂度。

## 4 复现细节

### 4.1 数据集

模型在四个数据集上进行了训练和评估：Amazon-Sports、Amazon-Clothing、MAG-Geology 和 MAG-Math。前两个为电商网络数据集，每个节点代表亚马逊上的一个商品，节点之间的边表示商品被频繁一起购买。后两个为学术网络数据集，每个节点代表一篇论文，节点之间的边表示一篇论文引用了另一篇论文。为降低 LLM 在链路预测推理阶段的计算开销，所有数据集被随机下采样至 20,000 个节点。

### 4.2 任务

使用 LINKGPT 完成两个任务：

1. **链路预测**：根据候选目标节点与源节点之间的连接概率对  $N_C$  个候选节点进行排序。为了更清晰地评估模型的排序能力，此实验未采用检索阶段。
2. **邻居预测**：模型根据提示直接生成相关节点的文本，然后将其用于从  $N_C$  池中检索  $N_C$  个候选目标节点。需要注意的是，链路预测是主要任务，而邻居预测的设计旨在提高推理过程中的效率。

### 4.3 评估指标

在链路预测任务中，使用了 MRR 和 Hits@1 这两个指标，它们是文献中最常用的评估指标之一。MRR 被定义为正目标节点的平均倒数排名，Hits@1 表示正目标节点在所有候选节点中排名第一的比例。在邻居预测任务中，生成的文本被用作检索的查询，性能通过整个检索-重排序过程的 MRR 和运行时间进行评估。如果正目标节点未被检索到，其排名将被视为无穷大。基线模型包括三类竞争模型：

1. 传统基于 GAE 的模型，包括 GCN<sup>[5]</sup>、GraphSAGE<sup>[6]</sup>和 GATv2<sup>[7]</sup>。
2. 利用成对信息的模型，包括 SEAL<sup>[8]</sup>、BUDDY<sup>[9]</sup>和 LPFormer<sup>[10]</sup>。
3. 基于 LLM 的模型，如的 LINKGPT，包括 vanilla LLAMA2-7B、GraphGPT<sup>[11]</sup>和 LLaGA<sup>[12]</sup>。

### 4.4 实现细节

选用 LLAMA2-7B 作为框架的基础模型。在每个指令微调阶段，链路预测和邻居预测任务分别进行 1 轮训练。此外，为了公平比较，使用 bert-base-uncased<sup>[13]</sup>作为 LINKGPT 和所有基于 GNN 的基线模型的节点原始嵌入。对于基于 LLM 的模型，采用其原始论文中描述的嵌入方法。

```

> Prompt: This is a product on Amazon:
1 <node_start><node>
2 text: Frogg Toggs 647484036325 Chilly Sport Cooling Towel, 33" Length x 6-1/2" Width,
3
4 This is another product:
5 <node_start><node><pairwise_start><pairwise>
6 text: Frogg Toggs 647484036301 Chilly Sport Cooling Towel, 33" Length x 6-1/2" Width, Gray
7 Is this product also bought by the same user?
8 Answer: Yes
9
10 This is another product:
11 <node_start><node><pairwise_start><pairwise>
12 text: Pearl Izumi Men's Pro Barrier Arm Warmer
13 Is this product also bought by the same user?
14 Answer: No
15
16 This is another product:
17 <node_start><node><pairwise_start><pairwise>
18 text: Lucky Scooter HIC Shim System
19 Is this product also bought by the same user?
20 Answer: No
21
22 This is another product:
23 <node_start><node><pairwise_start><pairwise>
24 text: Frogg Toggs 647484036332 Chilly Sport Cooling Towel, 33" Length x 6-1/2" Width, Lime Green
25 Is this product also bought by the same user?
26 Answer: Yes
27
28 This is another product:
29 <node_start><node><pairwise_start><pairwise>
30 text: The Scout Hunting Slingshot
31 Is this product also bought by the same user?
32 Answer: No
33 Prediction: No
34 Yes Logit: 23.5312, No Logit: 29.4531
35 Yes Probability: 0.0027, No Probability: 0.9971

```

图 3: 测试结果示例

#### 4.5 实验环境搭建

实验所用环境配置如下表 1 所示：在实验环境搭建过程中，需要手动下载部分库的特定版本，但

库	版本
Python	3.9
PyTorch	1.13.1
CUDA	11.7
Transformers 库	4.41.2
DGL	2.0.0+cu116

表 1: 使用库的版本

作者未明确指出各库的版本。通过在 GitHub 平台上向作者提问后，问题得以解决。

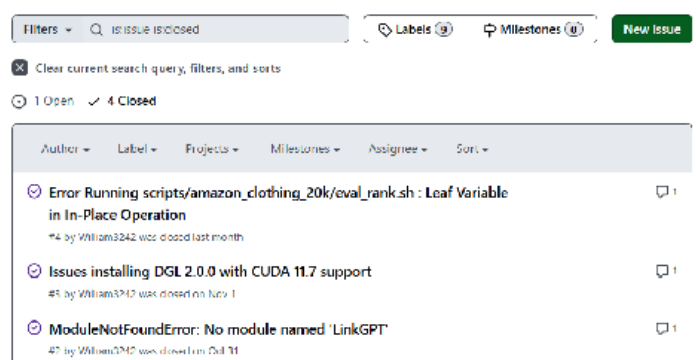


图 2: 复现过程在 GitHub 提出的问题

**Link Prediction:**  
This is the source node. <NODE> Text: {text}.  
This is another node. <NODE> <PAIRWISE> Text: {text}. Is this node connected with the source node? Answer: {Yes/No}  
This is another node. <NODE> <PAIRWISE> Text: {text}. Is this node connected with the source node? Answer: {Yes/No}  
...  
**Neighbor Prediction:**  
This is the source node. <NODE> Text: {text}. What nodes are connected with it?  
Answer: Text: {text}. Text: {text}. ...

图 4: 提示模板

## 5 实验结果分析

### 5.1 案例分析

如图 4 所示，这是微调过程及评估测试过程中所用的提示模板。利用该提示模板得到的一个测试结果如图 3 所示。在该案例中，通过链路预测任务，评估模型对用户购买行为的预测能力。具体而言，输入为一组产品描述，模型的目标是判断这些产品是否由同一用户购买。实验数据包含多种产品的文本描述，以及模型对这些产品间关联的预测结果。

模型在处理高度相似的产品（例如，同品牌和相似规格的毛巾）时，能够准确识别这些产品是否由同一用户购买。例如，对于两个产品“Frogg Toggs Chilly Sport Cooling Towel”（不同颜色或尺寸），模型预测为“Yes”，表明它们可能被同一用户购买，预测结果与真实情况一致。

对于明显不同类别的产品（例如，毛巾与自行车护臂或滑板零件），模型正确预测为“No”，表明其能够有效区分类别特征，避免错误关联。然而，在一些更复杂的场景中，模型的预测可能受到概率分布的影响。例如，某些边缘案例中，尽管概率较低（如最后一项产品“The Scout Hunting Slingshot”的预测概率为 0.0027），模型仍然能够输出正确的结果“否”。

这说明，模型在相似产品间（如同一品牌的不同颜色产品）能够准确捕捉关联，展示了良好的文本特征提取能力；在不同类别的产品之间（如毛巾与护臂或滑板零件）表现出较强的区分能力，证明模型能够理解产品语义上的显著差异。此外，在低概率情况下，模型仍然能够做出合理预测，表明其对置信度的处理具有一定的鲁棒性。

### 5.2 LINKGPT 性能表现优异

首先，比较了 LINKGPT 与最先进的基线模型在链路预测任务中的性能。候选节点的数量  $N_C$  在所有数据集中均设置为 150。结果如表 2 所示。LINKGPT 在链路预测任务上显著优于所有基线模型，在所有四个数据集上均实现了最先进的性能（在 MRR 和 Hits@1 指标方面）。这凸显了 LINKGPT 在链路预测中有效利用节点级、成对级和文本信息的能力。

尽管 GraphGPT 的表现相对较差，因为其设计主要用于节点分类任务，但 vanilla LLAMA2 尽管未专门为该任务训练，仍表现出合理的性能，表明预训练语言模型在利用 TAGs 的文本属性方面具有巨大潜力。LLaGA 通过同时利用节点级结构信息和文本信息，优于大多数基线模型，但仍落后于 LINKGPT，这表明有效建模成对信息在链路预测中的重要性。虽然 GATv2 由于其动态注意机制也实现了令人印象深刻的性能，但总体而言，能够捕获成对信息的模型表现更为优异。

复现结果对比表明，无论是论文原文中的结果，还是本次复现的结果，LINKGPT 在链路预测任

务上均显著优于所有基线模型。

表 2: 基线模型与 LINKGPT 在四个数据集上的性能表现

数据集	Amazon-Sports		Amazon-Clothing		MAG-Math		MAG-Geology	
	MRR	Hits@1	MRR	Hits@1	MRR	Hits@1	MRR	Hits@1
GCN	70.44	60.17	68.18	60.00	51.35	40.11	45.80	33.14
GraphSAGE	77.60	68.43	81.17	71.61	50.97	36.77	44.17	28.41
GATv2	81.44	72.96	87.83	81.56	65.65	55.27	51.59	37.95
SEAL	76.79	69.58	82.14	75.41	61.79	56.34	58.05	50.07
BUDDY	81.33	74.00	83.89	76.52	58.15	48.30	54.95	45.32
LPFormer	69.94	62.61	65.99	56.29	47.97	42.19	43.10	35.08
LLaMA2	40.81	30.88	30.22	22.49	22.92	13.54	21.86	13.47
GraphGPT	14.82	5.96	32.29	14.28	12.43	4.39	9.78	2.62
LLaGA	83.41	75.40	84.49	77.54	74.25	63.34	62.19	49.80
LINKGPT (本文方法)	<b>87.07</b>	<b>79.56</b>	<b>90.18</b>	<b>84.82</b>	<b>81.03</b>	<b>71.01</b>	<b>75.43</b>	<b>64.57</b>
复现结果	86.58	78.51	85.79	79.06	81.03	71.47	75.62	64.76

5.3 LINKGPT 泛化性能优异

在两种设置下评估了 LINKGPT 的泛化能力：零样本泛化（zero-shot generalization）和少样本上下文学习（few-shot in-context learning）。

**零样本泛化**)：在数据集  $A$  上训练 LINKGPT，并在另一个数据集  $B$  上评估，而不进行微调或在提示中提供额外的样本。结果汇总于表 3。Sports→Clothing 和 Math→Geology 属于跨类别泛化（cross-category generalization），因为这两个数据集来自同一领域（例如电商网络或学术网络）。Sports→Math 和 Math→Sports 则属于跨领域泛化（cross-domain generalization），因为这两个数据集来自不同领域。复现结果对比表明，无论是论文原文中的结果，还是本次复现的结果，LINKGPT 在链路预测任务上的零样本泛化能力均显著优于所有基线模型。

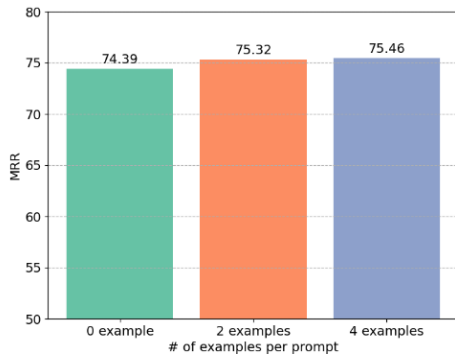
所提出的 LINKGPT 模型展现出最强的泛化能力，在所有设置中均优于其他基线模型。LINKGPT 的优势不仅源于其强大的 LLM 架构，使其具备出色的文本特征泛化能力，还由于其通过节点和成对编码有效捕获链路形成的结构共性。表中利用成对信息的模型均展示了良好的泛化能力，但由于无法有效迁移文本特征的知识，其性能落后于 LINKGPT。此外，尽管 LLaGA 也具备 LLM 的文本泛化能力，但由于未能明确和有效地捕获成对结构信息，其性能低于 LINKGPT。

表 3: 在链路预测任务上的零样本泛化能力

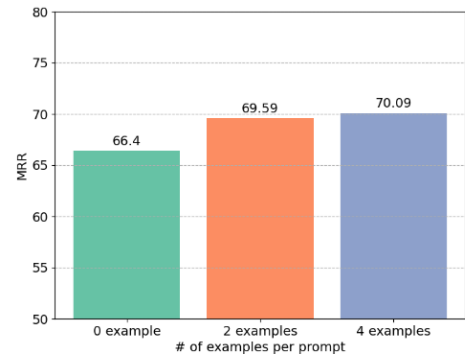
数据集	跨类别泛化				跨领域泛化			
	Sports→Clothing		Math→Geology		Sports→Math		Math→Sports	
	MRR	Hits@1	MRR	Hits@1	MRR	Hits@1	MRR	Hits@1
GraphSAGE	51.88	36.43	8.11	2.06	19.27	8.88	27.64	14.52
GATv2	67.01	55.36	21.41	13.54	36.87	26.86	49.68	37.21
BUDDY	81.23	72.01	24.45	13.35	26.43	15.73	42.39	21.22
LPFormer	68.26	59.64	42.70	38.06	48.36	43.75	56.16	44.36
LLaGA	78.62	68.90	23.20	13.13	46.82	12.97	53.67	40.46
LINKGPT (本文方法)	<b>85.74</b>	<b>79.01</b>	<b>71.58</b>	<b>60.42</b>	<b>67.50</b>	<b>55.27</b>	<b>69.38</b>	<b>55.40</b>
复现结果	84.48	77.51	71.09	59.06	65.03	54.47	75.62	64.76

**少样本上下文学习**：还评估了 LINKGPT 在测试时通过少量样本进行泛化的能力。其他基线模型由于其架构或提示设计的限制，无法实现上下文内学习。结果如图 5 所示。当在同一数据集上进行训练和评估时，由于模型已见过相关信息，上下文学习并未显著提升性能。





(a) MAG-Geology



(b) Amazon-Sports → MAG-Geology

图 5: 少样本上下文学习的性能表现

## 6 总结与展望

在本报告中，深入探讨了大规模语言模型在链路预测中的应用及其潜在局限性。尽管 LLMs 在推理和生成任务中展现出强大的能力，但“幻觉”问题，即模型生成看似合理但实际错误的信息，仍然是亟待解决的挑战。特别是在链路预测任务中，虚假链路的生成不仅可能误导模型的推断，还可能影响基于图数据的决策制定，带来严重后果。

为减轻幻觉的影响，未来研究可以从以下几个方向展开。首先，应用检索增强生成（Retrieval-Augmented Generation, RAG）技术，能够将外部知识引入 LLM 的生成过程，为模型提供更多事实基础，避免基于不准确信息的推理。这一方法能有效减轻模型的错误生成，增强其可靠性。此外，结合显式知识库或结构化数据，也是减少幻觉发生的重要手段。通过这些知识库作为额外输入，模型可以更准确地获取真实世界的知识，进一步提高预测精度。

另一个可行的改进是实施输出过滤机制或置信度阈值，以检测并剔除可能含有幻觉的输出。该方法通过对模型生成的结果进行可信度评估，筛选出更为可靠的链路预测结果，避免误导性预测的产生。

总体而言，尽管大规模语言模型在链路预测领域具有巨大潜力，但解决幻觉问题仍是提升模型应用效果的关键。未来研究可在多模态信息融合、结构化知识结合以及可信度评估机制等方面深入探索，以期为实际应用中的可靠性和准确性提供更有力的保障。

## 参考文献

- [1] YANG J, LIU Z, XIAO S, et al. GraphFormers: GNN-nested Transformers for Representation Learning on Textual Graph[C/OL]. // RANZATO M, BEYGELZIMER A, DAUPHIN Y, et al. Advances in Neural Information Processing Systems: vol. 34. Curran Associates, Inc., 2021: 28798-28810. [https://proceedings.neurips.cc/paper\\_files/paper/2021/file/f18a6d1cde4b205199de8729a6637b42-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2021/file/f18a6d1cde4b205199de8729a6637b42-Paper.pdf).
- [2] DEVLIN J, CHANG M, LEE K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J/OL]. CoRR, 2018, abs/1810.04805. arXiv: 1810.04805. <http://arxiv.org/abs/1810.04805>.
- [3] LI J, ZHANG W, WANG T, et al. GPT4Rec: A generative framework for personalized recommendation and user interests interpretation[J]. arXiv preprint arXiv:2304.03879, 2023.



- [4] ROBERTSON S, ZARAGOZA H, et al. The probabilistic relevance framework: BM25 and beyond[J]. *Foundations and Trends® in Information Retrieval*, 2009, 3(4): 333-389.
- [5] KIPF T N, WELING M. Semi-supervised classification with graph convolutional networks[J]. *arXiv preprint arXiv:1609.02907*, 2016.
- [6] HAMILTON W, YING Z, LESKOVEC J. Inductive representation learning on large graphs[J]. *Advances in neural information processing systems*, 2017, 30.
- [7] BRODY S, ALON U, YAHAV E. How attentive are graph attention networks?[J]. *arXiv preprint arXiv:2105.14491*, 2021.
- [8] ZHANG M, CHEN Y. Link prediction based on graph neural networks[J]. *Advances in neural information processing systems*, 2018, 31.
- [9] CHAMBERLAIN B P, SHIROBOKOV S, ROSSI E, et al. Graph neural networks for link prediction with subgraph sketching[J]. *arXiv preprint arXiv:2209.15486*, 2022.
- [10] SHOMER H, MA Y, MAO H, et al. Lpformer: An adaptive graph transformer for link prediction[C]// *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2024: 2686-2698.
- [11] TANG J, YANG Y, WEI W, et al. Graphgpt: Graph instruction tuning for large language models[C]// *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2024: 491-500.
- [12] CHEN R, ZHAO T, JAISWAL A, et al. Llaga: Large language and graph assistant[J]. *arXiv preprint arXiv:2402.08170*, 2024.
- [13] DEVLIN J. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. *arXiv preprint arXiv:1810.04805*, 2018.