# LLM Embeddings Improve Test-time Adaptation to Tabular $Y$ $|X$-Shifts

**摘要**

对于表格数据集，由于缺失变量（即混杂变量），标签与协变量之间的关系发生变化是很常见的。但由于无法在完全新的、未知的领域中进行适应，本文研究了一些可以轻松适应目标领域的模型，即使在只有少量标记样本的情况下，也能进行有效的适应。本文专注于构建更具信息性的表格数据表示，这些表示可以缓解 Y | X 变动，并提出通过将表格数据序列化，然后进行编码，这样可以利用大语言模型（LLMs）中的先验知识。

**关键词：** LLM；适应; 表格数据

## 1 引言

在现实世界的机器学习应用中，尤其是在处理表格数据时，目标领域与源领域之间的分布差异（通常称为分布偏移或迁移问题）往往会导致模型性能的显著下降。具体来说，表格数据中的标签和特征之间的关系可能会因为缺失的变量或隐藏的混杂因素而发生变化，形成所谓的 “Y | X-shifts”。这种问题尤其在表格数据中尤为显著，因为表格数据常常包含许多缺失或隐含的变量，这些变量的分布从源领域到目标领域发生了变化。此外，尽管传统的机器学习方法如线性回归、支持向量机和决策树等已在很多场景中取得了成功，但在面对目标领域和源领域之间显著的分布差异时，它们的表现依然不够理想。另一方面随着大语言模型（LLMs）在多个领域取得了显著的成功，它们作为表格数据的特征编码器，能够有效地捕捉数据中的语义信息，并帮助缓解由于缺失变量或混杂因素引起的 “Y | X-shifts”。大语言模型通过预训练学习到的世界知识，能够生成高效的特征表示，从而使得即使在目标领域样本有限的情况下，也能够有效地减小源领域和目标领域之间的分布差异。通过对这些模型的微调，结合少量标记样本，就能显著提高模型的泛化能力和适应能力。基于这一思路，研究表格数据中 “Y | X-shifts” 问题的有效解决方法，尤其是使用大语言模型嵌入和微调策略，将为机器学习在实际应用中的适应性和鲁棒性提供新的思路。

## 2 相关工作

此部分对课题内容相关的工作进行简要的分类概括与描述，二级标题中的内容为示意，可按照行文内容进行增删与更改，若二级标题无法对描述内容进行概括，可自行增加三级标题，后面内容同样如此，引文的 bib 文件统一粘贴到 **refs.bib** 中并采用如下引用方式。

### 2.1 表格数据与传统模型

表格数据是电子健康记录、金融以及社会与自然科学中常见的数据类型。与图像和文本等其他数据类型不同，梯度提升树 [1,2] 仍然是表格数据领域的最先进技术，甚至相较于专为表格数据设计的神经网络也表现优异。GBDT [3] 因其稳定性和高效性，尤其在面对大规模数据集时，常常作为标准基准。最近的研究表明，GBDT 在应对数据分布偏移时表现出强大的鲁棒性，这是它作为主基准方法的原因之一。此外。在表格数据领域，传统的机器学习方法如决策树、随机森林和支持向量机（SVM）[6,7,9] 等，依然被广泛应用于分类和回归任务中，尤其在数据量较小或特征较少的情况下，传统方法仍然具有较强的竞争力。尽管如此，随着深度学习技术的发展，基于神经网络的方法逐渐在多种应用中获得优势，尤其在高维数据和复杂任务上。

### 2.2 基于大语言模型的表格数据分析

最近，另一类研究开始探索大语言模型（LLM）[4,5,10] 在表格数据分类中的应用。与传统方法不同，这类研究通过使用大语言模型对表格数据进行编码，尝试从中提取高阶的、潜在的特征表达。LLM 的优势在于其能够处理不同格式的输入，自动从复杂的上下文中提取信息，并用于分类任务。目前，关于 LLM 在表格数据分类中的应用，已有三种主要的研究方向：(1) 少样本微调 [5]，即通过少量标注样本对 LLM 进行微调，优化其在特定任务中的表现 (2) 上下文学习 [10]，这种方法不直接训练模型，而是通过为 LLM 提供少量标注样本作为上下文提示来进行推理 (3) 零样本学习 [4]，LLM 通过理解任务描述和数据，直接对未见过的数据进行分类，无需额外的训练样本。然而，尽管 LLM 在处理表格数据上具有一定的优势，现有的研究并没有显式地解决数据分布偏移问题，尤其是零样本学习在目标领域中面临着较大的挑战。具体来说，零样本学习通常难以适应目标域中存在的 Y|X 分布变化，而上下文学习虽然能够利用少量标注目标数据，但往往不能充分利用大量的源域数据 [8]。

## 3 本文方法

### 3.1 本文方法概述

此部分对本文将要复现的工作本文的方法如图 1所示，主要可以分为三个部分，即嵌入获取部分。额外信息获取部分。训练和适应部分。

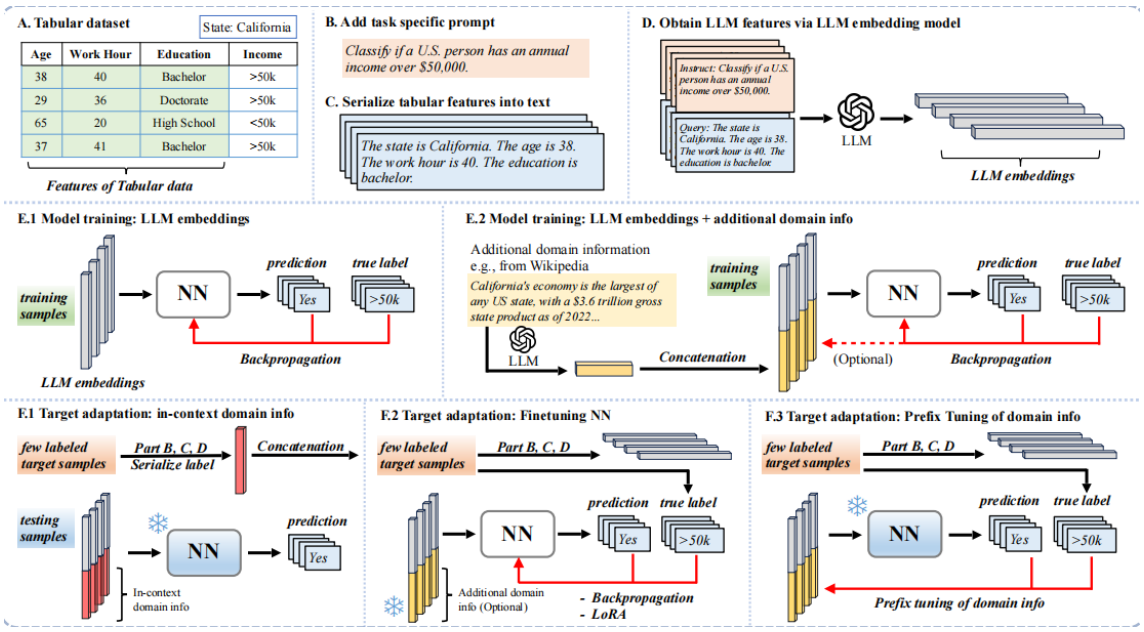


图 1. 方法示意图

### 3.2 LLM 嵌入获取

将表格数据转换为 LLM 嵌入，其中核心思想是将每个样本序列化为 LLM 能够处理的自然语言格式。那么怎么样序列化，这方面已有大量的研究，但本文考虑一种简单的序列化，即使用简单的文本模板并附加任务描述。

以收入预测问题为例，考虑一个简单的任务描述："判断 2018 年美国工作的成年人年收入是否超过 50000 美元。" 然后使用简单的序列化模板，列举所有特征，格式为："[特征名称]是 [值]"。采用这种序列化方法，我们使用编码器模型 e5-Mistral-7B-Instruct 生成 LLM 嵌入。形式上，编码器将样本 X 的序列化 Serialize(X) 作为输入，并输出相应的嵌入 $\Phi(X)$，即：

$$X \xrightarrow{\text{序列化}} \text{Serialize}(X) \xrightarrow{\text{e5-Mistral-7B-Instruct}} \Phi(X) \tag{1}$$

由于 e5-Mistral-7B-Instruct 要求输入数据按照如下模板进行格式化：Instruct: 分类任务的描述 Query: 数据的描述所以本文将任务描述放在 "Instruct" 部分，使用序列化模板来格式化表格数据得到的训练，放在 "Query" 部分。图 1中的 A-D 部分给出了一个示例。

### 3.3 额外的领域信息

使用 LLM 嵌入的另一个优势是它们能够整合额外的领域信息或先验知识，记作 C。融入领域特定的信息有助于解决 Y|X 分布偏移问题，并提高目标域的泛化能力。本文提出了一种简单而有效的方法来将领域知识集成到表格预测中。与将领域信息与序列化的表格数据结合并生成单一 LLM 嵌入不同，本文为领域知识和序列化表格数据分别生成 LLM 嵌入，然后将它们连接起来。该方法的优点有两个：(a) 尽管领域信息可能包含比序列化表格特征更多的文字，但这种连接方法确保了二者的 1:1 平衡，防止生成一个过于关注较长领域信息的单一嵌入；这里的原因是大模型生成嵌入，无论多长，最后的嵌入都是 4096 维。(b) 通过将表格特征与领域信息分离，我们可以高效地更新领域信息，而无需重新生成整个数据集的所有嵌入。

本文探索了三种领域信息来源：Wikipedia、GPT-4 和标注的目标样本。由于本文的实验主要集中在社会经济因素上，就选择了 Wikipedi 上的每个美国州的“经济”数据作为领域信息 C。对于 GPT-4，我们提示它为每个州的预测任务提供相关的背景知识作为 C。对于标注的目标样本，我们将 32 个来自相关领域的标注样本序列化作为先验知识 C。在获得领域信息 C 后，我们使用 e5-Mistral-7B-Instruct 生成 C 的 LLM 嵌入。然后将这个嵌入与表格数据的 LLM 嵌入连接后，作为输入传递给后端神经网络模型（NN）。这种方法可以让我们能够只需要生成一次数据集的 LLM 嵌入，并根据需要将其与来自不同提示的嵌入连接，增强在 Y|X 分布偏移下的泛化能力。

### 3.4 模型训练与目标适应

#### 3.4.1 模型训练

对于后端模型，本文在表格特征和 LLM 嵌入的基础上，使用简单的神经网络（NN）分类器进行表格数据分类。该 NN 是一个简单的前馈神经网络，包含多个隐藏层、dropout 层和 ReLU 激活函数。当通过嵌入层加入额外的领域信息时，相同的嵌入会应用于来自同一领域的所有样本。然后将 LLM 嵌入与领域信息的嵌入连接在一起。这个连接后的向量会传递到隐藏层、dropout 层和 ReLU 激活函数中。对于所有神经网络，最后一层是一个输出维度为 2 的线性层，之后是一个 softmax 层用于二分类。训练时，我们使用交叉熵作为损失函数，批量大小为 128，并使用 Adam 优化器。

#### 3.4.2 目标适应

即使结合了 LLM 嵌入和领域信息，模型仍可能遇到 Y|X 分布偏移。在实践中，通常会有来自目标域的小量样本，这些样本可以帮助更好地将模型适应到目标域。然后基于源域训练的模型，探索了四种主要的目标适应方法：上下文领域信息、全参数微调、低秩适应（LoRA）和前缀调优（Prefix Tuning）用于领域信息的调整。

上下文领域信息：保持训练好的模型冻结，只更新领域信息，将其从源域训练中的标注样本的自然语言描述切换为推理阶段目标域的描述。

全参数微调：对整个神经网络进行微调，使用目标样本。

低秩适应（LoRA）：在每个线性层引入一个低秩适应层，通过加入两个较小的矩阵 A 和 B，它们的秩为 16，仅微调这些 LoRA 参数，保持模型的其他部分不变。

前缀调优：初始的领域信息嵌入作为进一步微调的起点。在训练过程中，NN 和源域的领域信息嵌入同时训练。对于目标适应，将领域信息嵌入从源域切换到目标域。NN 保持冻结，仅使用目标域的样本更新目标域的领域信息嵌入。

## 4 复现细节

### 4.1 与已有开源代码对比

参考原作这给出的 github 连接 https://github.com/namkoong-lab/LLM-Tabular-Shifts。先按照作者的源代码运行，成功跑出论文结果。然后这篇文章研究是的不是表格数据的适应吗，我就尝试把他扩展到 graph 领域，进行了相关方面的研究。首先确定了选择文本图数据，

然后按照原文的方法进行序列化，然后后面流程一样，但这样发现效果一般，所以就对其进行优化，本文的序列化只关注于自身的特征，而 graph，一般需要考虑其他结点的相关性，所以我选择把一阶邻居的特征也加入进去，按照一下形式，它的邻居的特征是，然后再进行序列化。此外，本文使用的 nn 模型，将其换为图的 gcn 模型，这就我的复现及部分改进思路。

### 4.2 实验环境搭建

使用 python3.8+torch 库搭建，在 A100 显卡上运行。

### 4.3 创新点

将本文的表格数据的适应迁移到图领域上，对图领域上的分布漂移进行了研究。

## 5 实验设置与结果分析

### 5.1 实验设置

原文数据集：使用从美国 ACS PUMS 数据衍生的 ACS 数据集的 ACS 收入数据集，目标是预测个人的个人收入是否超过 50K 图数据集：Citation3 涉及由 ArnetMiner 提供的三个引用数据集，这些数据集来自不同的来源和时间段。具体而言，ACMv9（A）、Citationv1（C）、DBLPv7（D）分别来自 ACM（2000-2010 年间）、Microsoft Academic Graph（2008 年之前）和 DBLP 数据库（2004-2008 年）。然后，每篇论文根据其研究主题被分类为五个类别（即 DB、AI、CV、IS 和 Networking）。这些分布变化既来自时间层面，也来自领域层面。我选择对 A 到 C 数据集的适应进行研究。

### 5.2 实验结果

表 1. ACSmobility 实验结果比较，评价指标 ACC（准确率）

| **ACSmobility** | **svm** | **nn** | **nn 微调 nn** | **nn 微调 emb** |
|---|---|---|---|---|
| 原文 | 0.56 | 0.4725 | 0.57 | 0.58 |
| LLama-7b | 0.573 | **0.4632** | 0.563 | 0.584 |

总的来看复现工作完成的不错，和论文所给工作差距并不大。

表 2. 图领域实验结果

| **Citation** | **gnn** | **微调 gnn** | **微调 embedding** | **微调 lora** |
|---|---|---|---|---|
| ACC | 0.43 | 0.54 | 0.63 | 0.62 |

从实验结果来看，本文所提出一些适应方法存在一定的作用，均比直接使用 gnn 进行分类的效果好。但感觉提升也不算太大，具有很多可优化的地方。

## 6 总结与展望

我首先参考原文作者提供的代码与实验流程，成功复现了论文中针对表格数据的分布漂移适应方法，包括序列化、特征嵌入生成、域外迁移以及适应策略等关键步骤。这验证了原方法的可行性，并为迁移至图领域奠定了基础。然后尝试探索该方法在图领域的适用性，我先对图数据中的节点特征进行了序列化，然后将节点本身的特征与一阶邻居节点的特征整合后，输入 LLM，生成嵌入表示。进一步，我使用图神经网络（GCN）替代原方法中的神经网络模块，完成了针对图数据的分布漂移任务的训练和适应。但仍然有一些优化空间。首先，我的序列化方法仅整合了一阶邻居的信息，而在更复杂的图数据中，多阶邻居或全局信息可能对任务有重要影响。但由于序列化的文本长度限制以及计算开销，未能在本文中进一步深入探索。此外使用 GCN 作为图神经网络模型，其表示能力有限，尤其在处理复杂图结构和高阶邻居信息时可能不足。更先进的图模型（如 GAT、GraphSAGE 等）或结合预训练图模型的方法尚未纳入研究范围。另一方面图领域中的分布漂移形式更加复杂（如拓扑结构漂移、节点属性漂移等），而本文的实验设计主要基于节点属性漂移，未全面覆盖其他漂移类型，这可能限制了方法的适用性。针对这些不足，可以在未来进行进一步的优化。如可以尝试设计更高效的序列化方案，例如结合更高阶邻居的信息或通过图嵌入技术直接生成节点特征的摘要，同时保持输入文本的简洁性和有效性，及引入更先进的图神经网络（如 GraphSAGE、GAT、GraphTransformer 等）以增强模型的泛化能力。此外，可探索将 LLM 与预训练图模型结合，充分利用两者的表示能力。最后针对图数据中的多种分布漂移形式（如结构漂移、边权重漂移等），设计更灵活的适应策略，并评估不同策略的效果。同时，可以研究如何通过少量标注样本更高效地适应目标域。

## 参考文献


[1] JH FRĠEDMAN. Stochastic gradient boosting. computational statistics and data analysis. 2002.

[2] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[3] Josh Gardner, Zoran Popovic, and Ludwig Schmidt. Subgroup robustness grows on trees: An empirical baseline investigation. *Advances in Neural Information Processing Systems*, 35:9939–9954, 2022.

[4] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*, pages 5549–5581. PMLR, 2023.

[5] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

[6] Arlind Kadra, Marius Lindauer, Frank Hutter, and Josif Grabocka. Well-tuned simple nets excel on tabular datasets. *Advances in neural information processing systems*, 34:23928–23941, 2021.

[7] Liran Katzir, Gal Elidan, and Ran El-Yaniv. Net-dnf: Effective deep modeling of tabular data. In *International conference on learning representations*, 2020.

[8] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. 2023. *URL https://arxiv. org/abs/2303.15361*.

[9] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

[10] Dylan Slack and Sameer Singh. Tablet: Learning from instructions for tabular data. *arXiv preprint arXiv:2304.13188*, 2023.