**关键问题和技术路线一一对应**

按照技术推导路线的模式，都以股票为例子

1、关键问题：设计适当的时间窗口，确保不同尺度的时序特征能够被重编程并嵌入到 LLM 的输入中。（合适切片+合适embedding）

1、1）多尺度切片 （技术：时间窗口和自适应窗口）

切片和图片的卷积一样，时间窗口就是固定时间间隔，一直切切切，对同一个曲线，用大间隔切一遍，得到一堆曲线段。同样的曲线用小间隔切一遍，得到另一堆。自适应窗口就是波动大的切短一点，波动小的切长一点。

本质和卷积一样，关键两点：1:保留特征（2分钟间隔一直波动的，不能被大间隔抹平了）2:压缩时间序列（好像还有空间序列，不管了）

对应关键问题：确保不同尺度的时序特征（合适切片，不合适干脆多切点）

1、2）多尺度融合（嵌入+融合）

拿到了长长短短的时间序列切片，我们要把它升维成嵌入向量embedding，嵌入的方式有很多，会得到不一样的embedding向量，如线性变换，CGN，CNN，每个效果都不一样，但是《目的都是为了还能和时间扯上关系，还有时间序列的特征（t=1升，t=2降这种）》

融合：把向量拼起来，或者搞个加权平均（和神经网络，transformer里面梯度下降差不多｛我猜的｝反正这里要优化loss，优化权重）

关键问题：重编程并嵌入到 LLM 的输入（就是找到合适的embedding向量）

1、3）上下文重编程（利用llm自注意力机制）

困扰了好久，可能就是PaP。利用大语言模型自己的推理能力？

2、关键问题：时序数据的数值模式、文本模式对齐

2、1）时序转成语义：

法一：预定义的词汇表或符号化规则（把曲线变化描述成通用的语言：上升下降）

法二：领域特定符号表构建（股票上升不是单纯上升，金融界人士故作高深，说是“牛市”）｛本质上，调用了LLM关于金融的专业数据库｝

作用：视频里说是压缩符号表（就是词典），因为原始词典很大，但是金融时间序列用不到那么多。

2、2）语义对齐与重编程

第一步：文本与时序数据对齐

非常关键的一步，我今天上午要再次研究一下。我们有了文本词典，但是时间数据要和文本有对应关系啊，所以相似度计算决定了文本和数据对应的准确度，假如曲线上升，对应“下降”，那么LLM不可能预测准确

作用：对齐文本和时序数据

第二步：上下文语义描述生成

感觉是升级PaP，根据上下文生成对应的prompt，而不是time-LLM单一的prompt。（可以外加一个prompt生成器）

2、3） 跨模态对齐：（我要上午再去翻翻论文呜呜呜）

确保结构化输入，共同作用

3:关键问题：通用的！时间序列预测（感觉像是一个补充）

3、1）

和2、2）类似，PaP升级版，但是作用不是生成复杂prompt，发挥推测潜能。而是让LLM聚焦于关键特征。

3、2）优化loss

**问题清单**

backbone model是什么：特征提取器，就是文中的大模型

使用的大模型：llm-7B 二十多个GB 所以用Bert，作为实验对象。

具体需要调用的文件，会在晚上的具体流程文档中写

思路：利用利用LLM预测时间序列

问题：大模型只能输入文本输出文本

办法：时间序列转化成大模型能理解的toke（文本片段），记作text\*

问题：大模型不能理解test\*

办法1:预处理数据，就是时间序列的点连成曲线，然后用自己的话描述数据，就是把曲线转化成文本

（bug）没有办法描述所有的细节（例如上升——升到哪，升了几秒？）

@总结：这叫多模态，强调数据融合，把图像，时间序列，文本统一成一种形式，来解决问题

办法2:重编程（timeLLM干的）

仍然是时间序列—>文本，但是不是人去写，而是设计小型的神经网络，时序自动转化特征映射token。

@总结：这叫跨模态，指的是知识迁移，把大模型学会的文本知识用到时间序列问题上。（官方一点说。这种方法涉及从一个模态（例如文本）提取信息，并使用这些信息来理解或增强另一个模态（例如图像或声音）的内容）所以本质上来说是跨模态

问题：cv，nlp都有通用模型，但是时间序列中子任务（长时间序列预测，异常检测，补全，短时间序列预测）他们用的模型都不一样，我们要找到通用的模型。

办法：技术路线的第三点

优化：PaP（要解决的任务目标+时间序列数据转成的文本）

问题：具体的预测数据

办法：使用投影层，就是大模型的输出层不用了，自己训练一个全连接层。

最后回顾一下整个流程

提示词+时序（转成token）—>特征（LLM懂的）—>llm处理—>特征output—>全连接层输出具体时间序列

INTRODUCTION

时间序列预测是许多现实世界动态系统中的关键能力，应用范围包括需求规划、库存优化、能源负荷预测和气候建模等。

时间序列预测任务通常需要深厚的领域专业知识和特定的模型设计，与可以执行多种自然语言处理任务的通用语言模型（如GPT-3、GPT-4、Llama）形成鲜明对比。

预训练的基础模型，如大型语言模型（LLMs），在计算机视觉和自然语言处理领域推动了快速进步。

尽管时间序列模型没有享受到同样的突破性进展，但LLMs的强大能力激发了其在时间序列预测中的应用。

利用LLMs提升预测技术有几个期望特点：泛化能力，LLMs在少量样本或零样本迁移学习上表现出色，这可能意味着它们在不重新训练的情况下具有跨领域泛化的潜力。

数据效率：LLMs能够高效利用数据。

时间序列数据与自然语言模态的有效对齐是实现上述好处的关键，但这是一个挑战，因为LLMs处理离散令牌，而时间序列数据本质上是连续的。

将LLMs的知识应用于准确、数据高效和任务无关的时间序列预测仍是一个开放性挑战。

本研究提出了TIME-LLM，一个重新编程框架，用于将大型语言模型适应于时间序列预测，同时保持基础模型不变。

TIME-LLM的核心思想是将时间序列重新编程为更适合语言模型能力的文本原型表示。

为了增强模型对时间序列概念的理解，引入了Prompt-as-Prefix（PaP）方法，通过在输入时间序列中增加额外上下文和任务指导来丰富信息。

通过这种重新编程方法，大型语言模型可以成为有效的少量样本和零样本时间序列学习者，优于专门的预测模型。

本研究的工作指出了向多模态基础模型发展的方向，这些模型可以在语言和序列数据任务上都表现出色。

提出的重新编程框架为赋予大型模型新能力提供了一种可扩展的范式。

RELATED WORK

任务特定学习：

大多数时间序列预测模型针对特定任务和领域设计，并在小规模数据上端到端训练。

例如，ARIMA模型用于单变量时间序列预测，LSTM网络用于序列建模，时间卷积网络和变换器用于处理更长时间依赖性。

模态内适应：

计算机视觉和自然语言处理领域的研究表明，预训练模型可以通过微调适应各种下游任务，无需从头开始训练。

时间序列预训练模型（TSPTMs）的开发受到启发，这些模型首先通过监督或自监督学习进行预训练，然后微调以学习特定任务。

跨模态适应：

基于模态内适应，最近的研究探讨了将NLP和CV中强大的预训练基础模型的知识转移到时间序列建模。

例如，Voice2Series将语音识别中的声学模型适应到时间序列分类，利用大规模语音数据训练的模型快速适应时间序列。

最近的研究进展：

Chang等（2023年）提出LLM4TS，使用大型语言模型进行时间序列预测，采用两阶段微调过程。

Zhou等（2023年a）利用预训练语言模型进行微调，并在各种时间序列分析任务上展示性能。

我们的方法：

与上述方法不同，我们既不直接编辑输入时间序列，也不微调基础LLM。

我们提出通过源数据模态重编程并结合提示，发挥LLM作为有效时间序列处理工具的潜力。

METHODOLOGY

我们的研究关注于重编程一个嵌入-可见的语言基础模型，如Llama和GPT-2，用于通用的时间序列预测，且不需要对基础模型进行微调。

研究问题具体为：给定一个包含N个不同1维变量在T个时间步骤上的历史观测序列X，目标是重编程大型语言模型以理解和准确预测H个未来时间步骤的读数Ŷ，目标是减小预测值与真实值之间的均方误差。

方法包含三个主要部分：(1) 输入变换，(2) 预训练并冻结的大型语言模型(LLM)，(3) 输出投影。

多变量时间序列首先被分割成N个单变量时间序列，然后独立处理。

通过添加额外的提示前缀增强LLM的时间序列推理能力，以指导输入补丁的转换。

只有轻量级的输入变换和输出投影的参数被更新，而基础语言模型保持冻结。

与视觉-语言和其他多模态语言模型相比，TIME-LLM可以直接优化，仅需要少量时间序列数据和几个训练周期即可使用，效率高且资源约束较少。

为了进一步减少内存占用，可以无缝集成各种现成的技术（如量化）来压缩TIME-LLM。

输入嵌入时，每个输入通道X(i)首先通过可逆实例归一化(RevIN)单独归一化，以减轻时间序列分布偏移。

输入通道被划分为多个连续的重叠或非重叠补丁，以形成紧凑的输入令牌序列，减少计算负担，同时保留局部语义信息。

MAIN RESULTS

TIME-LLM 在多个基准和设置中始终优于最新的预测方法，尤其在少量样本和零样本场景下表现更为突出。

与一系列最新的模型进行了对比，包括一种针对时间序列分析微调语言模型的研究（Zhou et al., 2023a）。

为了确保公平比较，所有基线模型都遵循（Wu et al., 2022）的实验配置，并使用统一的评估流程。

默认使用 Llama-7B（Touvron et al., 2023）作为基础模型，除非有其他说明。

与最新的时间序列模型进行比较，包括一系列基于 Transformer 的方法和一些近期竞争性模型。

在短期预测中，进一步与 N-HiTS (2023b) 和 N-BEATS (2019) 进行比较。

在四个 ETT 数据集上进行评估，这些数据集被广泛用于基准测试长期预测模型。

设置输入时间序列长度为 512，使用四种不同的预测范围 H ∈ {96, 192, 336, 720}。

评估指标包括均方误差（MSE）和平均绝对误差（MAE）。

TIME-LLM 在大多数情况下优于所有基线模型，并且在多数情况下优势显著。

与近期的工作 GPT4TS (Zhou et al., 2023a) 对比，TIME-LLM 表现尤为突出，平均性能提高了 13% 和 15%。

与最新的特定任务 Transformer 模型 PatchTST 对比，TIME-LLM 通过重新编程最小的 Llama 实现了平均 MSE 降低 2%。

在其他模型（如 DLinear）相比，TIME-LLM 的改进也很明显，超过了 10%。

在 M4 基准上进行了测试，该基准包含不同采样频率的市场数据。

在少量样本学习能力的评估中，TIME-LLM 在 10% 和 5% 的少样本学习场景下明显优于所有基线方法。

TIME-LLM 和 GPT4TS 均持续优于其他竞争性基线，进一步凸显了语言模型作为高效时间序列处理工具的潜力。

CONCLUSION AND FUTURE WORK

TIME-LLM通过将时间序列数据重编程为更适合LLM（大型语言模型）的文本原型，并通过Prompt-as-Prefix提供自然语言指导，显示出在时间序列预测中适应冻结的大型语言模型的潜力。

评估显示，经过适配的LLM可以超越专业的专家模型，表明它们作为有效时间序列机器的潜力。

研究结果提供了一种新见解，即时间序列预测可以被视作另一种“语言”任务，通过Time-LLM框架，使用现成的LLM可以达到最先进的性能。

进一步研究应探索最优的重编程表示，通过持续预训练丰富LLM的显式时间序列知识，构建能够跨模态联合推理的多模态模型，并将重编程框架应用于赋予LLM额外新能力。

相关工作扩展聚焦于任务特定学习，特别是与本研究进行比较的最相关模型。

近期工作通过整合信号处理原理（如贴片、指数平滑、分解和频率分析）改进Transformer，用于时间序列预测。

PatchTST模型通过将时间序列分割成贴片作为输入令牌，保留局部语义，减少注意力的计算/内存需求，并允许更长的历史记录，提高了长期预测的准确性。

ETSformer模型将指数平滑原理融入Transformer注意力中，提高了准确性和效率。

FEDformer模型结合了Transformer和季节性趋势分解，通过频率增强实现长期预测，提供了比标准Transformer更好的性能和效率。

Autoformer模型使用基于自相关的分解架构，为复杂序列提供渐进式分解能力，在效率和准确性上超过自注意力。

虽然这些方法在效率和质量上优于原始的Transformer，但它们大多为特定领域的狭窄预测任务设计和优化，通常在小的、特定领域的数据集上进行端到端训练。

这些专业模型在其目标任务上表现强，但在实际世界遇到的多样化时间序列数据上牺牲了通用性和适用性。

为了推进时间序列预测，需要更灵活、广泛适用的模型，它们能够在不进行大量重训练的情况下适应新数据分布和任务。

理想的模型应该学习能够跨领域传递知识的时间序列表示。

开发这种广泛能力预测模型仍然是一个开放性的挑战。