作为ICLR 2024会议论文发表

TIME-LLM：通过重新编程大型语言模型来预测时间序列



**金明**1 **\*，王士玉**2 **\*，马林涛**2**，褚志轩**2**，张詹姆斯**2**，石晓明**2**，陈**3**品玉，梁**6**宇轩，李**1**元芳，潘**4† **世瑞，温青松**

1莫纳什大学2Ant Group 3IBM Research 4格里菲斯大学5阿里巴巴集团

6

1香港科技大学（广州）

{明。Jin, yuanfang.li}@monash.edu, pin-yu.chen@ibm.com yuxliang@outlook.com, s.pan@griffith.edu.au, qingsongedu@gmail.com {weiming.wsy,lintao.mlt,chuzhixuan.czx,james.z,peter.sxm}@antgroup.com

摘要

时间序列预测在许多现实世界的动态系统中具有重要意义，并得到了广泛的研究。与自然语言处理（NLP）和计算机视觉（CV）不同，单个大型模型可以处理多个任务，时间序列预测模型通常是专门的，需要针对不同的任务和应用进行不同的设计。虽然预训练的基础模型在NLP和CV方面取得了令人印象深刻的进步，但它们在时间序列领域的发展受到数据稀疏性的限制。最近的研究表明，大型语言模型（法学硕士）对复杂的符号序列具有强大的模式识别和推理能力。然而，挑战仍然存在于有效地对齐时间序列数据和自然语言的模式以利用这些能力。在这项工作中，我们提出了TIME-LLM，这是一个重编程框架，可以在主干语言模型保持完整的情况下，将法学硕士用于一般时间序列预测。我们首先用文本原型重新编程输入时间序列，然后将其输入到冻结的LLM中以对齐两种模式。为了增强LLM对时间序列数据进行推理的能力，我们提出了Prompt-as-Prefix (PaP)，它丰富了输入上下文并指导了重新编程的输入补丁的转换。最后对LLM转换后的时间序列patch进行投影以获得预测结果。我们的综合评估表明，TIME-LLM是一个强大的时间序列学习器，优于最先进的专业预测模型。此外，TIME-LLM在few-shot和zero-shot两种学习场景下都表现出色。代码可在https://github.com/KimMeen/Time-LLM上获得

1介绍

时间序列预测是跨越许多现实世界动态系统的关键能力（Jin等人，2023a），其应用范围从需求规划（Leonard, 2001）和库存优化（Li等人，2022）到能源负荷预测（Liu等人，2023a）和气候建模（Schneider & Dickinson, 1974）。每个时间序列预测任务通常需要广泛的领域专业知识和特定于任务的模型设计。这与GPT-3 （Brown et al., 2020）、GPT-4 （OpenAI, 2023）、Llama （Touvron et al., 2023）等基础语言模型形成鲜明对比，后者可以在少量射击甚至零射击设置中在各种NLP任务上表现良好。

预训练的基础模型，如大型语言模型（法学硕士），已经推动了计算机视觉（CV）和自然语言处理（NLP）的快速发展。虽然时间序列建模并没有从同样的重大突破中受益，但法学硕士令人印象深刻的能力激发了它们在时间序列预测中的应用（Jin等人，2023b）。利用法学硕士来推进预测技术有几个理想条件*：通用性。*法学硕士已经证明了少射和零射迁移学习的卓越能力（Brown et al., 2020）。这表明他们的

1

†相应的作者

∗ 平等的贡献

arXiv: 2310.01728 v2 (cs。LG] 29 Jan 2024

作为ICLR 2024会议论文发表

具有跨领域的一般化预测的潜力，无需从头开始对每个任务进行再培训。相比之下，目前的预测方法通常是严格按领域专门化的。*数据的效率。*通过利用预先训练的知识，法学硕士已经显示出仅用少数例子执行新任务的能力。这种数据效率可以在历史数据有限的情况下进行预测。相比之下，目前的方法通常需要丰富的领域内数据。*推理。*法学硕士表现出复杂的推理和模式识别能力(Mirchandani et al., 2023；Wang et al., 2023；Chu et al., 2023)。利用这些技能可以通过利用学习到的更高层次的概念来做出高度精确的预测。现有的非llm方法在很大程度上是统计性的，没有太多的先天推理。*多模式的知识。*随着LLM架构和训练技术的改进，他们在视觉、语音和文本等模式上获得了更多样化的知识（Ma et al., 2023）。利用这些知识可以实现融合不同数据类型的协同预测。传统工具缺乏共同利用多个知识库的方法。*简单的优化。*法学硕士只需接受一次大规模计算训练，就可以应用于预测任务，而无需从头开始学习。优化现有的预测模型通常需要大量的架构搜索和超参数调优（Zhou et al., 2023b）。总之，与当前的专业建模范例相比，法学硕士提供了一条有希望的途径，使时间序列预测更加通用、高效、协同和可访问。因此，将这些强大的模型用于时间序列数据可以释放未开发的巨大潜力。

上述好处的实现取决于时间序列数据和自然语言的模式的有效对齐。然而，这是一项具有挑战性的任务，因为法学硕士操作的是离散令牌，而时间序列数据本质上是连续的。此外，在法学硕士的预训练中，解释时间序列模式的知识和推理能力并不自然存在。因此，解锁法学硕士内部的知识，激活他们以准确、数据高效和任务不可知的方式进行一般时间序列预测的能力，仍然是一个开放的挑战。

在这项工作中，我们提出了TIME-LLM，这是一个重编程框架，用于适应时间序列预测的大型语言模型，同时保持骨干模型的完整性。其核心思想是将*输入时间*序列重新编程为更自然地适合语言模型能力的文本原型表示。为了进一步增强模型对时间序列概念的推理能力，我们引入了提示即前缀*（PaP），这是一种新颖的想法*，可以用额外的上下文丰富输入时间序列，并以自然语言的形式提供任务指令。这提供了关于应用于重编程输入的所需转换的声明性指导。然后对语言模型的输出进行投影，以生成时间序列预测。我们的综合评估表明，当通过这种重编程方法采用时，大型语言模型可以作为有效的少射和零射时间序列学习器，优于专门的预测模型。通过利用法学硕士的推理能力，同时保持模型的完整性，我们的工作指出了在语言和顺序数据任务上都能表现出色的多模态基础模型。我们提出的重编程框架提供了一种可扩展的范式，用于向大型模型注入超出其原始预训练的新功能。我们在这项工作中的主要贡献可以总结如下：

•我们引入了一种新颖的概念，即在不改变预训练的骨干模型的情况下，*重新编程*用于时间序列预测的大型语言模型。通过这样做，我们表明预测可以被视为另一种“语言”任务，可以由现成的LLM有效解决。

•我们提出了一个新的框架，TIME-LLM，它包括将输入时间序列重新编程为LLM更自然的文本原型表示，并使用声明性提示（例如，领域专家知识和任务指令）来增加输入上下文，以指导LLM推理。我们的技术指向在语言和时间序列方面都表现出色的多模态基础模型。

•TIME-LLM在主流预测任务中始终超过最先进的性能，特别是在少射和零射场景中。此外，这种优越的性能是在保持卓越的模型重编程效率的同时实现的。因此，我们的研究是释放法学硕士在时间序列和其他序列数据方面尚未开发的潜力的具体步骤。

2相关工作

**特定于任务的学习。**大多数时间序列预测模型都是为特定的任务和领域（例如，流量预测）精心设计的，并在小规模数据上进行端到端的训练。一个例证见

2

作为ICLR 2024会议论文发表

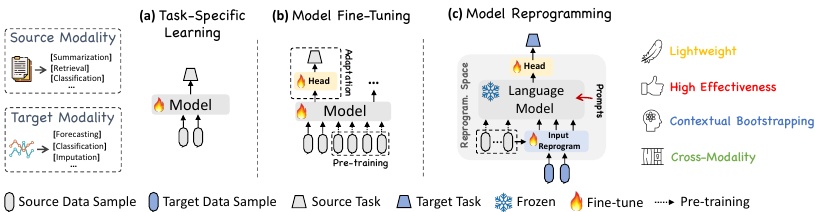


图1：重编程大型语言模型（llm）的示意图，比较**(a)**特定任务学习和**(b)**模型微调。我们的提案调查并演示了**(c)**如何有效地将开源llm重新编程为强大的时间序列学习者，其中开发良好的时间序列预训练模型并不容易获得。

图1 (a)。例如，ARIMA模型是为单变量时间序列预测而设计的（Box等人，2015），LSTM网络是为序列建模而定制的（Hochreiter & Schmidhuber, 1997），时间卷积网络（Bai等人，2018）和变压器（Wen等人，2023）是为处理更长的时间依赖性而开发的。虽然在狭窄的任务上取得了良好的性能，但这些模型缺乏对不同时间序列数据的通用性和泛化性。

**In-modality适应。**CV和NLP的相关研究已经证明了预训练模型的有效性，这些模型可以针对各种下游任务进行微调，而无需从头开始进行昂贵的训练(Devlin等人，2018；Brown et al., 2020；Touvron等人，2023)。受这些成功的启发，最近的研究集中在时间序列预训练模型（TSPTMs）的开发上。其中第一步涉及使用不同策略的时间序列预训练，如监督（Fawaz等人，2018）或自监督学习(张等人，2022b；Deldari等人，2022；张等，2023)。这允许模型学习表示各种输入时间序列。一旦进行了预训练，它就可以在类似的域上进行微调，以学习如何执行特定的任务（Tang et al., 2022）。一个例子如图1(b)所示。TSPTMs的发展利用了NLP和CV中预训练和微调的成功，但由于数据稀疏性，在较小规模上仍然受到限制。

**交叉模式适应。**在模态内适应的基础上，最近的工作进一步探索了通过多模态微调（Yin等，2023）和模型重编程（Chen, 2022）等技术，将知识从NLP和CV中强大的预训练基础模型转移到时间序列建模。我们的方法与这一类别保持一致；然而，关于时间序列的相关研究有限。Voice2Series （Yang et al., 2021）就是一个例子，它通过将时间序列编辑成适合AM的格式，将声学模型（AM）从语音识别调整到时间序列分类。最近，Chang等人（2023）提出了使用llm进行时间序列预测的LLM4TS。它在LLM上设计了一个两阶段的微调过程——首先是时间序列的监督预训练，然后是特定任务的微调。Zhou等人（2023a）杠杆-在不改变其自注意和前馈层的情况下对预训练的语言模型进行老化。该模型在各种时间序列分析任务上进行微调和评估，并通过从自然语言预训练中转移知识来展示可比或最先进的性能。与这些方法不同的是，我们既不直接编辑输入时间序列，也不微调主干LLM。相反，如图1(c)所示，我们建议用源数据模态重新编程时间序列，并辅以提示，以释放llm作为有效时间序列机器的潜力。

3的方法

我们的模型架构如图2所示。我们专注于重新编程嵌入式可见语言基础模型，如Llama （Touvron等人，2023）和GPT-2 (Radford等人，2019)，用于一般时间序列预测，*而不*需要对骨干模型进行任何微调。具体来说，我们考虑以下问题：给定一系列历史观测X∈N×T r，由跨T个时间步长N个不同的一维变量组成，我们的目标是重新编程一个大型语言模型f（·），以理解输入时间序列，并准确预测未来H个时间步长（用Y表示）的读数

{∈RN×H}，总体目标是最小化均方

基本真理Y与预测之间的误差，即H1 PH h=1 ||Yh−Yh||F。

ˆ

2

我们的方法包含三个主要组成部分：(1)输入变换，(2)预训练和冻结的LLM，以及(3)输出投影。最初，一个多变量时间序列被划分为N个

3

作为ICLR 2024会议论文发表

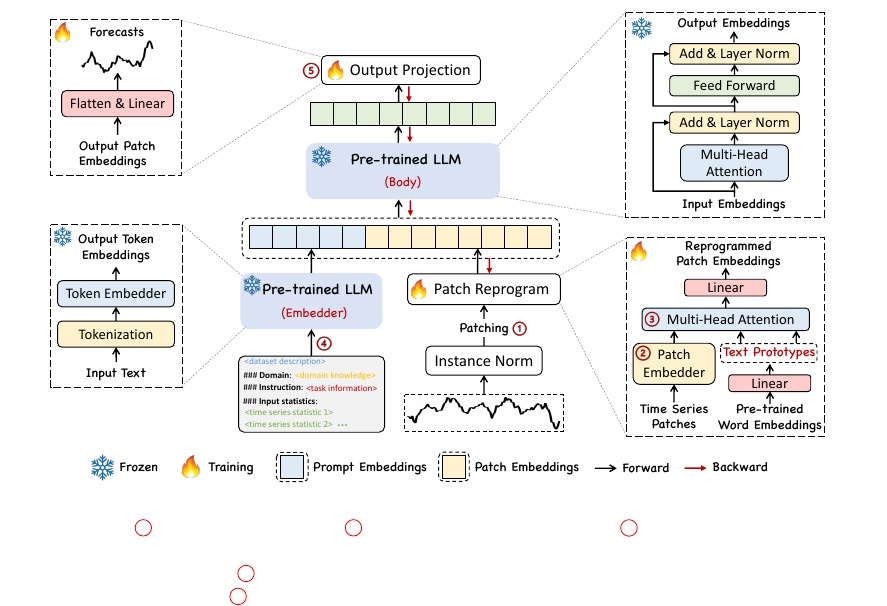


图2:TIME-LLM的模型框架。给定一个输入时间序列，我们首先通过1个补丁和2个自定义嵌入层对其进行标记
  
和嵌入。3然后用压缩文本原型重新编程这些补丁嵌入
  
，以对齐两个模态。为了增强LLM的推理能力，在输入中添加了4个额外
  
的提示前缀，以指导输入补丁的转换。5对LLM的输出patch进行投影以生成预测。

单变量时间序列，随后进行独立处理（Nie等人，2023）。第i个序列表示为X(i) ∈R1×T ，在使用学习到的文本原型重新编程以对齐源和目标模态之前，该序列经历归一化、修补和嵌入。然后，我们通过将LLM与重新编程的补丁一起提示以生成输出表示来增强LLM的时间序列推理能力，这些表示被投影到最终预测Y´(i)∈R1×H。

我们注意到，只有轻量级输入变换和输出投影的参数被更新，而主干语言模型被冻结。与视觉语言和其他多模态语言模型（通常使用配对的跨模态数据进行微调）相比，TIME-LLM是直接优化的，并且只需要一小部分时间序列和几个训练周期就可以随时使用，与从头开始构建大型特定领域模型或对其进行微调相比，它保持了高效率并施加了更少的资源约束。为了进一步减少内存占用，可以无缝集成各种现成的技术（例如，量化）以减小TIME-LLM。

3.1模型结构

**输入嵌入。**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，使其具有零均值和单位标准差，以减轻时间序列分布移位（Kim et al., 2021）。然后，我们将x(i) 划分为几个连续的重叠或不重叠的斑块（Nie等，2023），长度为Lp；因此，输入patch的总数为P =⌊（T−L P） S⌋+ 2，其中S表示水平滑动步幅。潜在的动机有两个方面：(1)通过将本地信息聚合到每个补丁中来更好地保留本地语义信息；(2)作为标记化，形成一个紧凑的输入标记序列，减少计算负担。给定这些补丁XP ∈RP ×L p，我们将它们嵌入为X

(i)

↑(i)∈RP ×dm ， P

采用简单的线性层作为patch embedder来创建维度dm。

**补丁重新编程。**在这里，我们将补丁嵌入重新编程到源数据表示空间中，以对齐时间序列和自然语言的模态，以激活主干的时间序列理解和推理能力。一种常见的做法是学习一种形式的“噪声”，当应用于目标输入样本时，允许预训练的源模型在不需要参数更新的情况下产生所需的目标输出。这对于桥接数据在技术上是可行的

4

作为ICLR 2024年会议论文发表

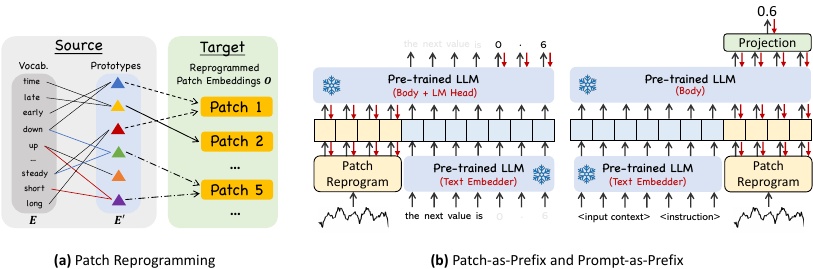


图3:(**a)**补丁重编程和**(b)** patch -as- prefix与Prompt-as-Prefix的示意图。

相同或相似的模式。示例包括重新利用视觉模型来处理跨域图像（Misra等人，2023）或重新编程声学模型来处理时间序列数据（Yang等人，2021）。在这两种情况下，源数据和目标数据之间都存在显式的、可学习的转换，从而允许直接编辑输入样本。然而，时间序列既不能直接编辑，也不能用自然语言无损地描述，这对直接引导LLM来理解时间序列提出了重大挑战，而不需要进行资源密集型的微调。

为了缩小这一差距，我们建议对X进行重新编程

↑(i)使用预训练的词嵌入E∈RV ×D P

在主干中，其中V为词汇表大小。然而，没有先验知识表明哪些源标记是直接相关的。因此，简单地利用E将导致巨大且可能密集的重编程空间。一个简单的解决方案是通过线性探测E（表示为E′∈RV ×D，其中V′≪V）来维持少量文本原型集合。插图见

′

图3(一个)。文本原型学习连接语言线索，例如“short up”（红线）和“steady down”（蓝线），然后将其组合以表示局部补丁信息（例如，“short up then down steady”用于表征patch 5），而不留下语言模型预训练的空间。这种方法是高效的，并且允许自适应地选择相关的源信息。为了实现这一点，我们采用了多头交叉注意层。具体来说，对于每个头部k ={1，···，k}，我们定义查询矩阵Q

(i)

= X

(i)WQ，键矩阵K(i) = E 'WK, P K K K K

和值矩阵Vk = E 'WkV，其中WkQ∈Rdm×d， WK, WkV∈Dk×dR。具体地说,(我)

D为骨干模型的隐维，D =⌊dKm ⌋。然后，我们有对每个注意头中的时间序列补丁进行重编程的操作，定义为：

FORMULA

通过对每个头中的每个Zk (i)∈RP ×d进行聚合，我们得到Z(i)∈RP ×d m。然后将其P ×D (i) 线性投影以使隐藏维度与骨干模型对齐，从而得到O∈R。

**提示符作为前缀。**提示是一种直接而有效的llm特定任务激活方法（Yin et al., 2023）。然而，将时间序列直接翻译成自然语言存在相当大的挑战，既阻碍了指令跟随数据集的创建，也阻碍了在不影响性能的情况下有效利用实时提示（Xue & Salim, 2022）。最近的进展表明，其他数据模式（如图像）可以无缝地集成为提示的前缀，从而促进基于这些输入的有效推理(Tsimpoukelli

Et al., 2021)。受这些发现的启发，并使我们的方法直接适用于现实世界的时间序列，我们提出了另一个问题：*提示符可以作为前缀来丰富输入上下文并指导转换*

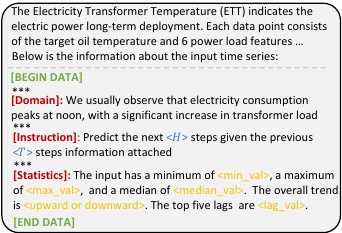


图4：提示示例。<>和<>是特定于任务的配置和计算的输入统计。

*重新编程的时间序列补丁的生成？*我们将这一概念称为提示符作为*前缀*（PaP），并观察到它显著增强了LLM对下游任务的适应性，同时补充了补丁重编程（参见后面的第4.5节）。

5

作为ICLR 2024年会议论文发表

图3(b)展示了两种提示方法。在*Patch-as-Prefix*中，语言模型被提示预测时间序列中的后续值，以自然语言表达。这种方法遇到了一定的限制：(1)语言模型在没有外部工具的帮助下处理高精度数字时通常表现出较低的敏感性，从而在准确解决长期实际预测任务方面提出了重大挑战；(2)考虑到不同的语言模型是在不同的语料库上进行预训练的，并且在精确和高效地生成高精度数字时可能采用不同的标记化类型，因此需要对不同的语言模型进行复杂的、定制化的后处理。这导致预测以不同的自然语言格式表示，例如[' 0 ‘,’。‘, ’ 6 ', ' 1 ']和[' 0 ‘,’。‘, ’ 61 ']，表示小数0.61。

另一方面，提示符作为前缀巧妙地避免了这些限制。在实践中，我们确定了构建有效提示的三个关键组件：(1)数据集上下文，(2)任务指令和(3)输入统计。提示示例如图4所示。数据集上下文为LLM提供了关于输入时间序列的基本背景信息，这些信息通常在不同的领域中表现出不同的特征。任务指令作为LLM在特定任务的补丁嵌入转换中的关键指南。我们还用额外的关键统计数据（如趋势和滞后）来丰富输入时间序列，以促进模式识别和推理。

**输出投影。**如图2所示，通过冻结的LLM对提示符和补丁嵌入(i) o进行打包和前馈后，我们丢弃前缀部分并获得输出表示。接下来，我们对它们进行平坦化和线性投影，从而得到最终的预测Y (i)。

4个主要结果

TIME-LLM在多个基准和设置中，特别是在少射和零射场景中，始终比最先进的预测方法表现出色。我们将我们的方法与大量最新模型进行了比较，包括最近的一项研究，该研究对时间序列分析的语言模型进行了微调（Zhou et al., 2023a）。为了确保公平的比较，我们坚持（Wu et al., 2023）中所有基线的实验配置，并使用统一的评估管道1。除非另有说明，否则我们使用Llama-7B （Touvron et al., 2023）作为默认主干。

**基线。**我们与SOTA时间序列模型进行了比较，如果适用，我们引用了（Zhou等人，2023a）的性能。我们的基线包括一系列基于变压器的方法：PatchTST (2023), ESTformer(2022)，非平稳变压器（2022），FEDformer (2022), Autoformer (2021), Informer（2021）和Reformer（2020）。我们还选择了一组最近的竞争模型，包括GPT4TS （2023a）、LLMTime（2023）、DLinear（2023）、TimesNet（2023）和LightTS （2022a）。在短期预测中，我们进一步将我们的模型与N-HiTS （2023b）和N-BEATS（2020）进行比较。更多细节见附录A。

4.1长期预测

**设置。**我们对ETTh1、ETTh2、ETTm1、ETTm2、Weather、Electricity （ECL）、Traffic和ILI进行了评估，这些指标已被广泛用于长期预测模型的基准测试（Wu et al., 2023）。具体实现和数据集见附录b。输入时间序列长度T设为512，我们使用四个不同的预测视界H∈{96,192,336,720}。评价指标包括均方误差（MSE）和平均绝对误差（MAE）。

**结果。**我们的简要结果如表1所示，其中TIME-LLM在大多数情况下都优于所有基线，并且对大多数情况都明显优于基线。与GPT4TS （Zhou等人，2023a）的比较特别值得注意。GPT4TS是最近的一项工作，涉及对主干语言模型进行微调。我们注意到，与**GPT4TS**和TimesNet相比，平均性能分别提高了12%和**20%**。与SOTA特定任务的Transformer模型PatchTST相比，通过对最小的Llama进行重新编程，TIME-LLM实现了平均MSE降低1.4%。相对于其他模型，例如DLinear，我们的改进也很明显，超过了12%。

4.2短期预测

**设置。**我们选择M4基准（Makridakis et al., 2018）作为测试平台，该基准包含不同采样频率下的营销数据集合。在附录b中提供了更多的细节。在这种情况下的预测范围相对较小，在[6,48]中。输入长度

6

1 https://github.com/thuml/Time-Series-Library

在ICLR 2024上作为会议论文发表

表1：长期预测结果。所有结果均为四个不同预测视界的平均值：ILI的H∈{24,36,48,60}，其他的H∈{96,192,336,720}。值越低，表示性能越好。**红色**：最好，蓝色：次好。我们的完整结果见附录D。

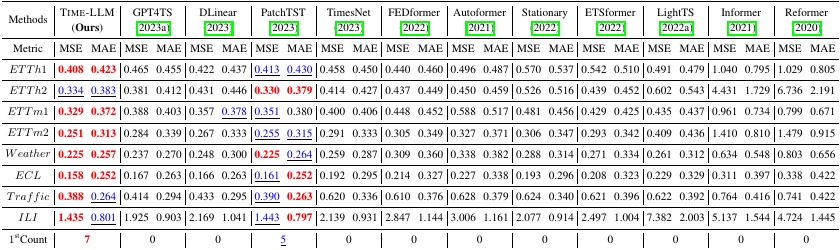
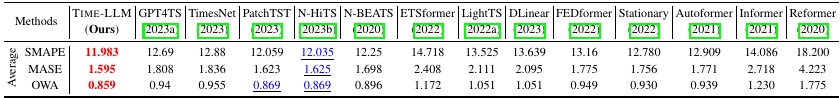


表2：对M4的短期时间序列预测结果。预测视界在[6,48]中，给出的三行是不同采样间隔下所有数据集的加权平均。值越低表示性能越好。**红色**：最好，蓝色：次好。更多结果见附录D。



是预测范围的两倍。评价指标为对称平均绝对百分比误差（SMAPE）、平均绝对缩放误差（MSAE）和整体加权平均（OWA）。

**结果。**我们在所有方法中统一种子的简要结果见表2。TIME-LLM始终超过所有基线，比GPT4TS高出8.7%**。**即使与SOTA模型、N-HiTS （Challu et al., 2023b）、w.r.t. MASE和OWA相比，TIME-LLM仍然具有竞争力。

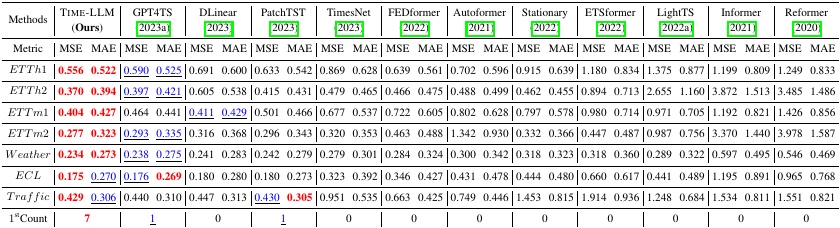
4.3 few-shot预测

**设置。**法学硕士最近展示了非凡的少镜头学习能力（Liu et al., 2023b）。在本节中，我们评估重新编程的LLM在预测任务中是否保留了这种能力。为了公平比较，我们坚持（Zhou等人，2023a）中的设置，并且我们在训练数据有限的情况下（即≤前10%的训练时间步长）进行评估。

**结果。**我们简短的10%和5%的少数次学习结果分别在表3和表4中。TIME- LLM明显优于所有基线方法，我们将此归因于我们重新编程的LLM中成功的知识激活。有趣的是，我们的方法和GPT4TS都不断超越其他竞争基准，进一步强调了语言模型作为熟练时间序列机器的潜在实力。

在10%的少次学习范围内，我们的方法与GPT4TS相比实现了**5%**的MSE降低，而无需对LLM进行任何微调。与最近的SOTA模型有关

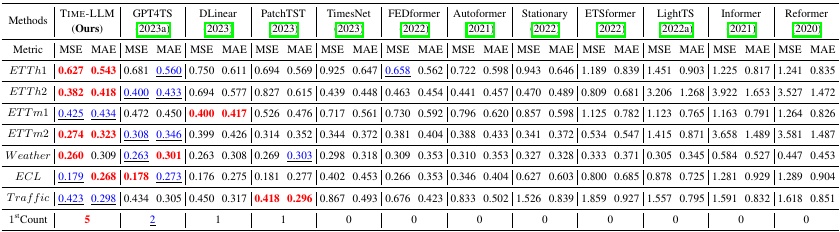
表3:10%训练数据上的少射学习。我们在表1中使用相同的协议。所有结果从四个不同的预测视界取平均值：H∈{96,192,336,720}。我们的全部结果见附录E。



7

作为ICLR 2024年会议论文发表

表4:5%训练数据下的少射学习。我们在表1中使用相同的协议。所有结果从四个不同的预测视界取平均值：H∈{96,192,336,720}。我们的全部结果见附录E。

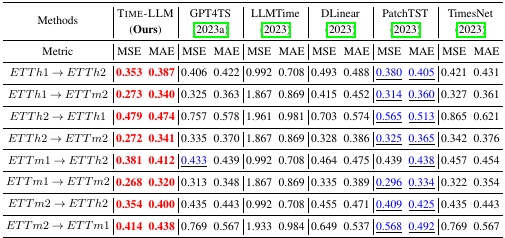


如PatchTST， DLinear和TimesNet，我们的平均增强超过**8%**，**12%**和**33%** w.r.t.MSE。在5%的少数次学习场景中，我们可以看到类似的趋势，其中我们对GPT4TS的平均进步超过**5%**。与PatchTST、DLinear和TimesNet相比，TIME-LLM的平均改进幅度超过20%。

4.4零概率预测

**设置。**除了几次学习之外，法学硕士还具有作为有效的零次推理器的潜力（Kojima et al., 2022）。在本节中，我们在跨域适应的框架内评估了重新编程的LLM的零次学习能力。具体来说，我们检查模型在另一个数据集♠上优化时在数据集上的表现如何，而模型没有

表5:Zero-shot学习结果。**红色**：最好，蓝色：第二好。附录E显示了我们的详细结果。



遇到来自数据集的任何数据样本与few-shot学习类似，我们使用长期预测协议，并利用ETT数据集对各种跨域场景进行评估。

**结果。**我们的简要结果见表5。TIME-LLM的性能始终大大优于最具竞争力的基准，**w.r.t.**超过14.2%，在MSE降低方面排名第二。考虑到少数镜头的结果，我们观察到，在数据稀缺的情况下，重新编程LLM往往会产生明显更好的结果。例如，我们在10%少射预测、5%少射预测和零射预测中的总体误差减少w.r.t. GPT4TS逐渐增加：7.7%、**8.4%和**22%**。即使**对LLMTime（该领域最新的方法，其主干LLM的大小与之相当（7B））进行**基准测试**，TIME-LLM也显示出超过75%的显著改进。我们将此归因于我们的方法在执行时间序列任务时以资源高效的方式更好地激活LLM的知识转移和推理能力。

4.5模型分析

**语言模型变体。**我们比较了两个具有不同容量的代表性主干（表6中的A.1-4）。我们的结果表明，在LLM重编程后，缩放律仍然保留。我们默认采用全容量的Llama-7B，其性能明显优于其1/4容量的变体(**A.2**；包括前8层Transformer)的性能高出**14.5%**。GPT-2 （**A.3**）的平均MSE降低**了14.7%，略优于其变体GPT-2 (6) (A.4**) 2.7%。

**交叉模式对齐。**我们在表6中的结果表明，去除补丁重编程或Prompt-as-Prefix都会损害LLM重编程中的知识转移，从而实现有效的时间序列预测。在没有表示对齐**（B.1）**的情况下，我们观察到**9.2%**的显着平均性能下降，在少数射击任务中变得更加明显（超过**17%**）。在TIME-LLM中，提示行为是利用LLM理解输入和任务的能力的关键因素。这一成分的消失（B.2）分别**导致**标准预测任务和少量预测任务的8%**和**19%以上的**退化**。我们发现删除输入统计数据（C.1）伤害最大，**导致**MSE平均增加10.2%。这是一个——

8

作为ICLR 2024年会议论文发表

表6:ETTh1和ETTm1在预测提前96步和192步时的消退（报告的MSE）。**红色**：最佳。

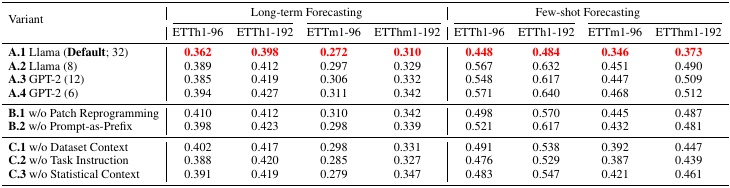
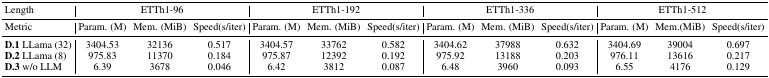


表7:TIME-LLM对ETTh1预测不同步长的效率分析。



作为外部知识的预期可以通过提示自然地纳入，以促进学习和推理。此外，为LLM提供明确的任务指令和输入上下文（例如，数据集标题）也是有益的(即，**C.2**和**C.1**；引出率分别超过**7.7%**和**9.6%**)。

**重新解释。**我们在图5中提供了一个用100个文本原型重新编程48个时间序列补丁的ETTh1案例研究。前4个子图可视化了从随机初始化(a)到良好优化(d)的重编程空间**优化**。我们发现只有一小部分原型（列）参与了子图(e)中输入补丁（行）的重编程。此外，补丁通过不同的原型组合经历了不同的表示。这表明：(1)文本原型学会了总结语言线索，并且选择了少数

与表示局部时间序列补丁中的信息高度相关，我们通过

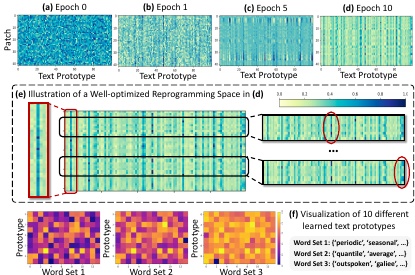


图5：补丁重编程的展示。

在子图**(f)**中随机选择10个。我们的结果表明，与描述时间序列属性的单词（即单词集1和2）具有高度相关性；(2) patch通常具有不同的底层语义，需要不同的原型来表示。

**重编程效率。**表7给出了有和没有主干LLM的TIME-LLM的总体效率分析。我们提出的重编程网络本身（D.3）在激活LLM的时间序列预测能力方面**是轻量级**的(即少于660万个可训练参数*；*仅占Llama-7B**中**总参数的0.2%左右)，TIME-LLM的总体效率实际上受到杠杆化主干（例如，D.1和D.2）的限制。即使与参数高效的微调方法（例如，QLoRA (Dettmers等人，2023)）相比，这在平衡任务性能和效率方面也是有利的。

5结论和未来的工作

通过将时间序列数据重新编程为对法学硕士来说更自然的文本原型，并通过Prompt-as-Prefix提供自然语言指导来增强推理，TIME-LLM在将冻结的大型语言模型用于时间序列预测方面显示出了希望。评估表明，经过调整的法学硕士可以胜过专门的专家模型，表明它们有潜力成为有效的时间序列机器。我们的研究结果还提供了一种新颖的见解，即时间序列预测可以被视为另一种“语言”任务，可以由现成的LLM解决，通过我们的time -LLM框架实现最先进的性能。进一步的研究应该探索最佳的重编程表示，通过持续的预训练，用明确的时间序列知识丰富法学硕士，并建立跨时间序列、自然语言和其他模式的联合推理的多模态模型。此外，还应考虑应用重编程框架为法学硕士提供更广泛的时间序列分析能力或其他新能力。

9

作为ICLR 2024年会议论文发表

参考文献

白少杰，J Zico Kolter和Vladlen Koltun。序列建模的通用卷积和循环网络的经验评价。*arXiv预印本arXiv:1803.01271*, 2018。

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, Greta M Ljung。*时间序列分析：预测与控制*。John Wiley & Sons, 2015。

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry， Amanda Askell等人。语言模型是几次学习。*神经信息处理系统进展*，33:1877-1901,2020。

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza， Max Mergenthaler和Artur Dubrawski。N-hits：用于时间序列预测的神经层次插值。*AAAI人工智能会议论文集*，2023a。

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski。Nhits：用于时间序列预测的神经层次插值。*《AAAI人工智能会议论文集》*，第37卷，第6989-6997页，2023b。

Chang Ching, Peng Wen-Chih, and Tien-Fu Chen。Llm4ts：使用预训练的llms进行时间序列预测的两阶段微调。*arXiv预印本arXiv:2308.08469*, 2023。

Pin-Yu陈。模型重编程：资源高效的跨领域机器学习。*arXiv预印本arXiv:2202.10629*, 2022。

褚志轩，郝红艳，欧阳昕，王艳，王艳，沈悦，顾金杰，崔青，李龙飞，薛思乔，等。利用大型语言模型进行预训练的推荐系统。*arXiv预印本arXiv: 2308.10837*,2023。

Shohreh Deldari，郝红艳薛思乔，Aaqib Saeed，何嘉远，Daniel V Smith, Flora D Salim。Beyond just vision：多模态和时态数据上的自监督表示学习综述。*arXiv预印本arXiv:2206.02353*, 2022。

蒂姆·Dettmers，阿蒂多罗·帕格诺尼，阿里·霍尔茨曼和卢克·泽特勒莫耶。Qlora：量子化llms的高效微调。*《神经信息处理系统进展》*，2023。

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova。Bert：深度双向变形器的语言理解预训练。*《计算语言学协会北美分会2019年会议论文集：人类语言技术》*，2018年。

哈桑·伊斯梅尔·法瓦兹，日尔曼·弗赖斯捷，乔纳森·韦伯，拉哈桑·伊杜姆哈尔和皮埃尔-阿兰·穆勒。时间序列分类的迁移学习。*IEEE大数据国际会议*，第1367-1376页。IEEE, 2018。

Nate Gruver, Marc Anton Finzi, Shikai Qiu, and Andrew Gordon Wilson。大型语言模型是零射击时间序列预报员。*《神经信息处理系统进展》*，2023。

Julien Herzen, Francesco Lassig, Samuele Giuliano Piazzetta, Thomas Neuer, Leo Tafti, Guillaume Raille, Thomas Van Pottelbergh, Marek Pasieka, Andrzej Skrodzki， Nicolas Huguenin等。Darts：时间序列的友好型现代机器学习。*机器学习研究学报，23*(1):5442-5447,2022。

Sepp Hochreiter和J¨urgen Schmidhuber。长短期记忆。*神经计算*，9(8):1735-1780,1997。

金明，柯焕仪，温青松，Daniele Zambon, Cesare Alippi, Geoffrey I Webb， Irwin King和潘石瑞。时间序列图神经网络研究综述：预测、分类、imputation和异常检测。*arXiv预印本arXiv:2307.03759*, 2023a。

金明，温青松，梁宇轩，张，薛思乔，薛思乔王，詹姆斯张，王怡，陈海峰，李小丽等。时间序列和时空数据的大模型：综述与展望。*arXiv预印本arXiv: 2310.10196*,2023b。

10

作为ICLR 2024年会议论文发表

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, Jaegul Choo。针对分布移位的精确时间序列预测的可逆实例归一化。在*国际学习表征会议*上，2021。

Diederik P. Kingma和Jimmy Ba。Adam：一种随机优化的方法。*国际学习表征会议*，2015。

Nikita Kitaev， Łukasz Kaiser, and Anselm Levskaya。改革家：高效的变压器。*《学习表征国际会议》*，2020。

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo，和Yusuke Iwasawa。大型语言模型是零射击推理器。*神经信息处理系统进展*，35:22199-22213,2022。

迈克尔·伦纳德。需求规划的促销分析和预测：一种实用的时间序列方法。*与展品*，1,2001。

Na李，Donald M Arnold, Douglas G Down, Rebecca Barty, John Blake, Fei Chiang, Tom Courtney, Marianne Waito， Rick Trifunov和Nancy M Heddle。通过整合机器学习、统计建模和库存优化，从需求预测到红细胞的库存订购决策。*输血*，62(1):87-99,2022。

刘恒波，马自清，杨林晓，周田，夏锐，王一，温青松，孙良。萨迪：一种极端事件下电力负荷预测的自适应分解可解释框架。*IEEE声学、语音和信号处理国际会议*，2023a。

刘欣，Daniel McDuff, Geza Kovacs, Isaac Galatzer-Levy, Jacob Sunshine，詹杰宁，金明哲，廖顺，Paolo Di Achille和Shwetak Patel。大型语言模型是少数几个镜头的健康学习者。*arXiv预印本arXiv:2305.15525*, 2023b。

刘勇，吴海旭，，陈海峰，龙明生。非平稳变压器：探索时间序列预测中的平稳性。*神经信息处理系统进展*，35:9881-9893,2022。

马子阳，吴文，郑志生，郭一伟，陈茜，张，陈茜。利用语音ptm、文本llm和情感tts进行语音情感识别。*arXiv预印本arXiv:2309.10294*, 2023。

Spyros Makridakis和Michele Hibon。m3竞赛：结果、结论和影响。*国际预测学报*，16(4):451-476,2000。

Spyros Makridakis， Evangelos Spiliotis和Vassilios Assimakopoulos。m4竞赛：结果、发现、结论和前进之路。*国际预测杂志*，34(4):802-808,2018。

Igor Melnyk, Vijil Chenthamarakshan, Pin-Yu Chen, Payel Das, Amit Dhurandhar， Inkit Padhi和Devleena Das。抗体序列填充的预训练语言模型重编程。*机器学习国际会议*，2023。

Suvir Mirchandani, Fei Xia, Pete Florence, Danny Driess, Montserrat Gonzalez Arenas, Kanishka Rao, Dorsa Sadigh, Andy Zeng，等。作为通用模式机器的大型语言模型。*《第七届机器人学习年会论文集》*，2023年。

Diganta Misra, Agam Goyal, Bharat Runwal, Pin Yu Chen。约束下的重编程：重新审视彩票的高效可靠可转移性。*arXiv预印本arXiv:2308.14969*, 2023。

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong，和Jayant Kalagnanam。一个时间序列抵得上64个字：用变压器进行长期预测。In *International Conference on Learning Representations*, 2023。

OpenAI。Gpt-4技术报告，2023年。

11

作为ICLR 2024年会议论文发表

Boris N Oreshkin、Dmitri Carpov、Nicolas Chapados和Yoshua Bengio。N-beats：可解释时间序列预测的神经基础展开分析。在*国际学习表征会议*上，2020。

亚当·帕斯克、山姆·格罗斯、弗朗西斯科·马萨、亚当·勒勒、詹姆斯·布拉德伯里、格雷戈里·查南、特雷弗·基林、泽明·林、娜塔莉亚·吉梅尔斯海因、卢卡·安提加等人。Pytorch：命令式、高性能的深度学习库。*《神经信息处理系统进展》*，2019年32期。

Alec Radford， Jeffrey吴海旭，Rewon Child, David Luan, Dario Amodei， Ilya Sutskever等。语言模型是无监督的多任务学习者。*OpenAI博客*，1(8):9,2019。

Stephen H Schneider和Robert E Dickinson。气候建模。*地球物理学评论*，12(3):447-493,1974。

唐义宏，瞿敖，周德华，林兆宏，王世成，马伟。领域对抗性时空网络：跨城市短期交通预测的可转移框架。在*第31届ACM国际信息与知识管理会议论文集中*，pp. 1905-1915, 2022。

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timoth´ee Lacroix, Baptiste Rozi 'ere, Naman Goyal, Eric Hambro， Faisal Azhar等。Llama:开放、高效的基础语言模型。*arXiv预印本arXiv:2302.13971*, 2023。

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, Felix Hill。使用冻结语言模型进行多模态少镜头学习。*神经信息处理系统进展*，34(4):200 - 212,2021。

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez， Łukasz Kaiser，和Illia Polosukhin。注意力就是你所需的一切。*神经信息处理系统进展*，2017年第30期。

Ria Vinod, Pin-Yu Chen, Payel Das。分子表征学习的重编程语言模型。*神经信息处理系统年会上*，2020。

王建民，褚志轩，欧阳昕，王建民，郝红艳，沈悦，顾金杰，薛四桥，詹姆斯张，崔青，等。用大型语言模型推理图增强推荐系统。*arXiv预印本arXiv:2308.10835*, 2023。

温青松，周田，张，陈卫奇，马自清，闫俊驰，孙良。《时间序列中的变形金刚：调查》。*人工智能国际联合会议*，2023。

Gerald Woo, Chenghao Liu, Doyen Sahoo， Akshat Kumar和Steven Hoi。Etsformer：用于时间序列预测的指数平滑变压器。*arXiv预印本arXiv: 2202.01381*,2022。

吴海旭，徐杰辉，王建民，龙明生。自耦器：用于长期序列预测的自相关分解变压器。*神经信息处理系统进展*，34(4):22419 - 22430,2021。

吴海旭，胡腾阁，刘勇，周航，王建民，龙明生。Timesnet：用于一般时间序列分析的时间二维变异建模。In *International Conference on Learning Representations*, 2023。

郝红艳薛四桥，Flora D Salim。基于提示的时间序列预测：一个新的任务和数据集。*arXiv预印本arXiv:2210.08964*, 2022。

赵汉，杨哈克，蔡云云，陈pinyu。声音2系列：重编程时间序列分类声学模型。*机器学习国际会议*，11808-11819页。PMLR, 2021年。

尹树康，付朝友，赵思瑞，李科，孙兴，徐彤，陈恩红。多模态大型语言模型综述。*arXiv预印本arXiv:2306.13549*, 2023。

12

作为ICLR 2024年会议论文发表

曾爱玲，陈木溪，张以卓，徐强。变形金刚对时间序列预测有效吗？摘自*AAAI人工智能会议论文集*，第37卷，pp. 11121-11128, 2023年。

张以卓，温青松，张以卓，蔡荣耀，金荣，刘勇，张以卓，梁宇轩，庞冠松，宋东进，等。时间序列分析的自监督学习：分类、进展与展望。*arXiv预印本arXiv:2306.10125*, 2023。

张以卓、张以卓、曹伟、边江、易晓涵、郑顺、李健。少即是多：基于轻采样导向mlp结构的快速多元时间序列预测。*arXiv预印本arXiv:2207.01186*, 2022a。

张以卓，赵自远，Theodoros Tsiligkaridis, Marinka Zitnik。基于时频一致性的时间序列自监督对比预训练。*神经信息处理系统进展*，35 (3):3988 - 4003,2022b。

周浩毅，张以卓，彭杰琪，张以卓，李建新，熊辉，张以卓。Informer: Beyond efficient transformer用于长序列时间序列预测。*《AAAI人工智能会议论文集》*，第35卷，第11106-11115页，2021年。

周田，马自清，温青松，王雪，孙良，金荣。Fedformer：用于长期序列预测的频率增强分解变压器。*《机器学习国际会议》*，pp. 27268-27286。PMLR, 2022年。

周，牛培松，王雪，孙亮，金荣。一刀切：预训练lm的Power一般时间序列分析。*神经信息处理系统进展*，36,2023a。

周，褚志轩，阮一佳，金戈，黄宇晨，李胜。ptse：一种用于概率时间序列预测的多模型集成方法。在第*32届国际人工智能联合会议*上，2023b。

13

作为ICLR 2024年会议论文发表

更相关的工作

**特定于任务的学习。**我们提供了特定任务学习相关工作的延伸，特别关注我们进行比较的最相关的模型。最近的工作通过结合信号处理原理，如修补、指数平滑、分解和频率分析，改进了Transformer （Vaswani et al., 2017）的时间序列预测。例如，PatchTST （Nie et al., 2023）将时间序列分割成补丁，作为Transformer的输入令牌。这保留了局部语义，减少了注意力的计算/内存，并允许更长的历史。与其他Transformer模型相比，它提高了长期预测的准确性。它在自监督预训练和迁移学习方面也取得了优异的表现。ETSformer （Woo et al., 2022）将指数平滑原理融入到Transformer注意力中，以提高准确性和效率。它使用指数平滑注意和频率注意来取代标准的自我注意。FEDformer（周等人，2022）将Transformer与季节趋势分解相结合。分解捕获全局概况，而Transformer捕获详细结构。它还使用频率增强进行长期预测。这提供了比标准Transformer更好的性能和效率。Autoformer （Wu et al., 2021）使用具有自相关的分解架构，为复杂序列提供渐进式分解能力。自相关是基于序列周期性设计的，用于进行依赖项发现和表示聚合。它在效率和准确性上都优于自关注。

虽然这些方法与香草Transformer相比提高了效率和准确性，但它们大多是针对特定领域内的狭窄预测任务而设计和优化的。这些模型通常在小的、特定领域的数据集上进行端到端训练。虽然在目标任务上实现了强大的性能，但这种专门的模型牺牲了在现实世界中遇到的各种时间序列数据范围内的通用性和泛化性。狭窄的焦点限制了它们对新数据集和任务的适用性。为了推进时间序列预测，需要更灵活、更广泛适用的模型，这些模型可以适应新的数据分布和任务，而无需进行大量的再培训。理想的模型将学习跨领域传递知识的鲁棒时间序列表示。开发如此广泛的预测模型仍然是一个开放的挑战。根据我们对之前相关工作的讨论，最近的研究已经开始通过预训练和架构创新来探索模型的通用性。然而，要实现我们在本研究中推进的真正通用的预测系统，还需要进一步的努力。

**交叉模式适应。**我们提供了跨模态适应相关工作的扩展概述，特别关注时间序列和其他数据模态的模型重编程的最新进展。模型重编程是一种资源高效的跨领域学习方法，它涉及从一个领域（源）调整一个开发良好的预训练模型来解决不同领域（目标）中的任务，而不需要模型微调，即使这些领域明显不同，正如Chen（2022）所指出的那样。在时间序列数据的背景下，Voice2Series （Yang et al., 2021）通过转换时间序列来拟合模型并将输出重新映射到新标签，从而使语音识别中的声学模型适应时间序列分类。类似地，LLMTime （Gruver等人，2023）将LLM用于零间隔时间序列预测，重点是对主干LLM的输入时间序列进行有效的标记化，然后自回归地生成预测。与这些方法不同的是，TIME-LLM不直接编辑输入时间序列。相反，它建议用源数据模式和提示重新编程时间序列，以释放LLM在标准、少射和零射场景中作为多功能预测器的全部潜力。该领域的其他著名作品（主要是在生物学领域）包括R2DL （Vinod等人，2020）和ReproBert (Melnyk等人，2023)，它们使用词性层序对氨基酸进行重编程。我们的补丁重编程方法的一个关键区别在于，与氨基酸的完整集合不同，时间序列补丁不会形成一个完整的集合。因此，我们建议优化一小部分文本原型及其到时间序列补丁的映射，而不是直接优化两个完备集（如词汇表和氨基酸）之间的大型转换矩阵。

B实验细节

责任的实现

我们主要遵循（Wu等人，2023）中的实验配置，在https://github.com/thuml/Time-Series-Library for的统一评估管道中跨越所有基线

14

作为ICLR 2024年会议论文发表

公平的比较。除非另有说明，否则我们使用Llama-7B （Touvron et al., 2023）作为默认骨干模型。我们所有的实验都重复了三次，并报告了平均结果。我们的模型实现是在PyTorch （Paszke et al., 2019）上进行的，所有实验都在NVIDIA A100-80G gpu上进行。我们详细的模型配置见附录B.4，我们的代码可以在https://github.com/KimMeen/Time-LLM上获得。

**技术细节。**我们在三个方面提供了TIME-LLM的额外技术细节：(1)文本原型的学习，(2)用于提示的时间序列趋势和滞后的计算，以及(3)输出投影的实现。为了从RE∈V ×D中识别出一小组文本原型E’∈V ×D，我们学习R了R一个矩阵W∈V ×V作为中介。来描述

′

′

用自然语言描述整体时间序列趋势，我们计算连续时间步长之间的差值之和。大于0的和表示上升趋势，而小于0的和表示下降趋势。此外，我们计算了时间序列的前5个滞后，通过使用快速傅立叶变换计算自相关并选择具有最高相关值的5个滞后来识别。在我们通过冻结的LLM打包和前馈提示和补丁嵌入O(i) ∈P ×D r之后，我们丢弃前缀部分并获得输出表示，表示为O ~ i∈RP ×D。随后，我们遵循PatchTST （Nie et al., 2023）并将O ~ i平坦化为长度为P ×D的1D张量，然后将其线性投影为Y

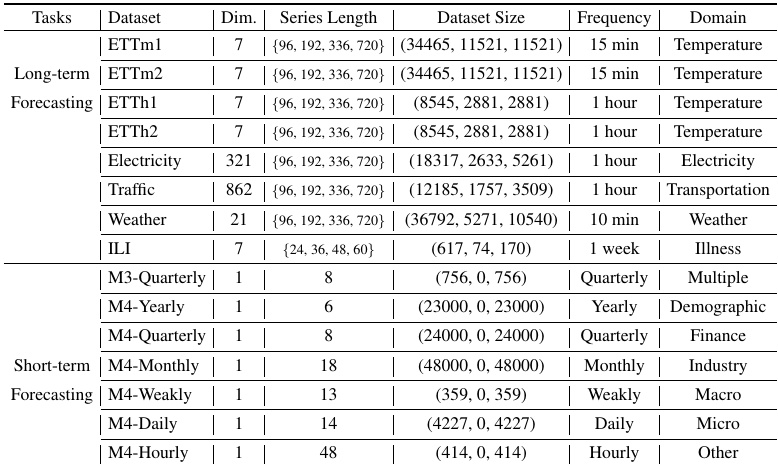
ˆ∈R 。

我

B.2数据集细节

数据集统计汇总如表8所示。我们在已建立的8个不同基准上评估了长期预测性能，包括4个ETT数据集（Zhou et al., 2021）（即ETTh1、ETTh2、ETTm1和ETTm2）、天气、电力、交通和ILI （Wu et al., 2023）。此外，我们在M4基准（Makri- dakis等人，2018）和M3基准（Makridakis & Hibon, 2000）的季度数据集上评估了短期预测的性能。

表8：数据集统计来自（Wu et al., 2023）。维度表示时间序列（即通道）的数量，数据集大小按（训练、验证、测试）进行组织。



电力变压器温度(ETT；反映长期电力部署的指标)基准由来自中国两个县的两年数据组成，并被细分为四个不同的数据集，每个数据集的采样率不同：ETTh1和ETTh2以1小时为采样周期，ETTm1和ETTm2以15分钟为采样周期。ETT数据集中的每个条目包括六个功率负载特征和一个目标变量，称为“油温”。电力数据集包括321个客户的用电量记录，以1小时的采样率测量。Weather数据集包括一年的记录

15

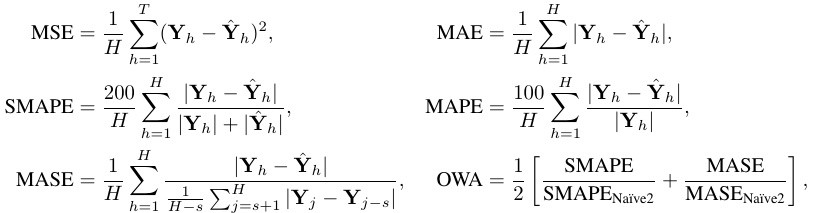
作为ICLR 2024会议论文发表

来自位于德国的21个气象站，采样率为10分钟。Traffic数据集包括高速公路系统的占用率数据，这些数据来自加利福尼亚州的862个传感器，采样率为1小时。流感样疾病（ILI）数据集包含患有严重流感并伴有并发症的患者记录。

M4基准包括100K个时间序列，从商业、金融和经济预测中常见的各个领域积累而来。这些时间序列被划分为六个不同的数据集，每个数据集的采样频率从每年到每小时不等。M3- quarterly数据集包括M3基准中的756个季度采样时间序列。这些序列被分为五个不同的领域：人口、微观、宏观、工业和金融。

B.3评价指标

对于评估指标，我们利用均方误差（MSE）和平均绝对误差（MAE）进行长期预测。在M4基准的短期预测中，我们采用了N-BEATS中的对称平均绝对百分比误差（SMAPE）、平均绝对缩放误差（MASE）和整体加权平均（OWA） （Oreshkin et al., 2020）。请注意，OWA是M4竞赛中使用的特定度量。这些指标的计算方法如下：

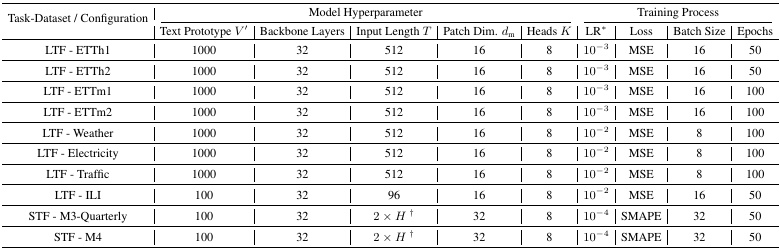


其中s为时间序列数据的周期性。H表示数据点的数量（在我们的例子中，即预测视界）。h Y和Y´h 是h∈{1，···，h}的第h个基本真理和预测。

B.4模型配置

相对于不同的任务和数据集，我们的模型配置在表9中进行了整合。默认情况下，所有实验都使用Adam优化器（Kingma & Ba, 2015）。具体来说，对于短期和长期预测任务，文本原型V '的数量分别保持在100和1000不变。我们满负荷使用Llama-7B模型，将所有任务的骨干模型层数保持在32层作为标准。术语输入长度T表示原始输入时间序列数据中存在的时间步长数。Patch维数m 表示重编程之前嵌入时间序列Patch的隐藏维数。最后，头像K与用于补丁重编程的多头交叉注意相关。在表9最右边的四列中，我们详细介绍了与模型训练相关的配置。

表9:TIME-LLM实验配置的概述。“LTF”和“STF”分别表示长期和短期预测。



†H表示M4和M3数据集的预测视界。\* LR表示初始学习率。

16

作为ICLR 2024会议论文发表

C超参数灵敏度

我们针对TIME-LLM中四个重要的超参数，即骨干模型层数、文本原型数V、时间序列输入长度T、′补丁重编程交叉注意头数k，进行了超参数敏感性分析，相关结果如图6所示。从我们的分析中，我们得出以下观察结果：(1)主干LLM中Transformer层数与TIME-LLM的性能之间存在正相关关系，证实了LLM重编程后的标度规律保持不变；(2)一般来说，获取更多的文本原型会增强性能。我们假设，有限数量的原型V可能′会在聚合语言线索时产生噪声，从而阻碍对表征输入时间序列补丁所必需的高度代表性原型的有效学习；(3)输入时间长度T与预测精度表现出直接关系，在预测扩展视界时尤其明显。这一观察结果是合乎逻辑的，与传统的时间序列模型是一致的；(4)在输入补丁重编程过程中增加注意头的数量被证明是有利的。

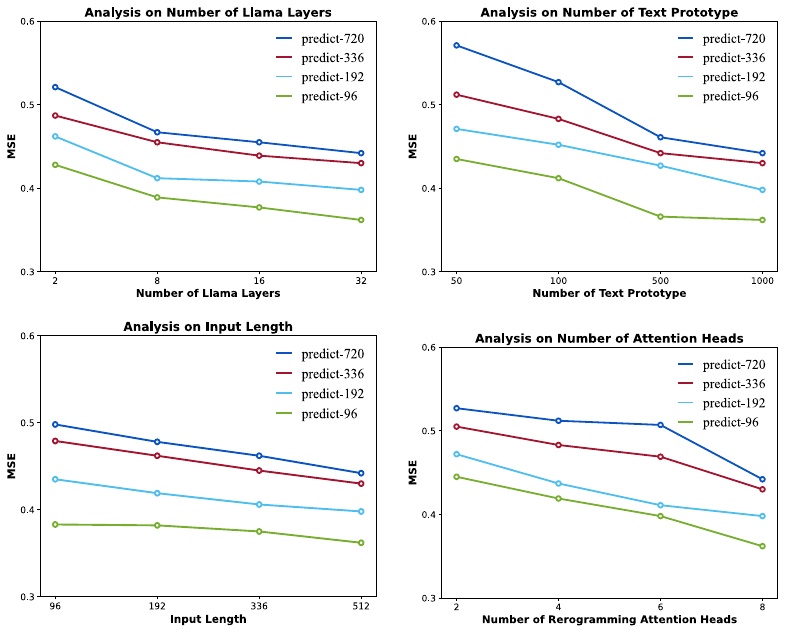


图6:ETTh1数据集上的超参数灵敏度分析。

D长期和短期预测

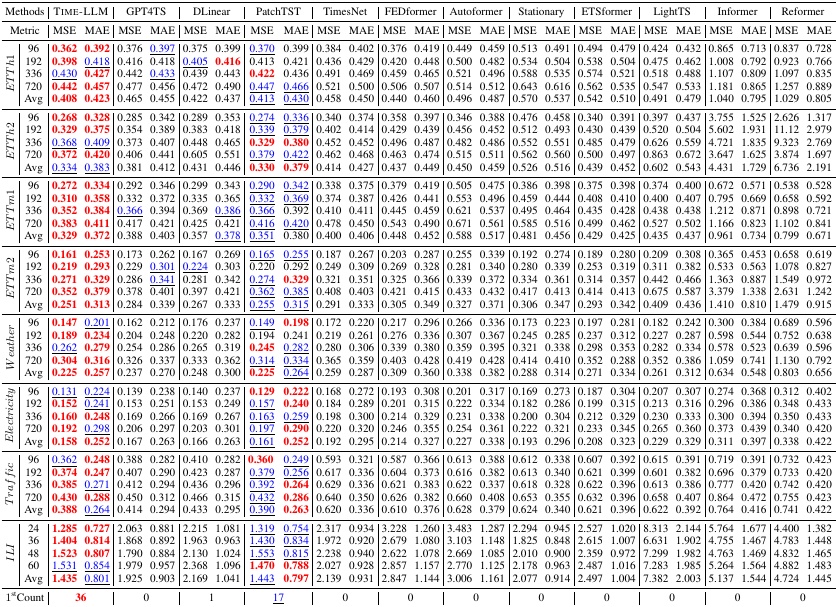
D.1长期预测

通过对最小的Llama模型进行重新编程，同时保持其完整性，TIME-LLM在8个时间序列基准测试的40个实例中的36个实例中实现了SOTA性能。这强调了llm作为稳健可靠的时间序列预测者的巨大潜力。此外，我们将所提出的方法与表11中其他完善的基线进行了基准测试。这种比较包括三种著名的统计方法（AutoARIMA， AutoTheta和AutoETS） （Herzen等人，2022）和两个最近的时间序列模型，N-HiTS （Challu等人，2023b）和N-BEATS （Ore- shkin等人，2020）。值得注意的是，TIME-LLM在所有情况下都保证了SOTA性能，在**MSE和MAE方面**比第二好的结果高出22%和16%。

17

作为ICLR 2024会议论文发表

表10：完整的长期预测结果。我们设置ILI的预测视界H∈{24,36,48,60}，其他的则为{96,192,336,720}。值越低，表示性能越好。**红色**：最好，蓝色：次好。



D.2短期预测

我们对短期预测的完整结果见表12。在大多数情况下，TIME-LLM始终优于大多数基线模型。值得注意的是，我们大大超过了GPT4TS（例如，**总体**8.7%,**m4 -** annual 13.4%, m4 - hour， **M4-Daily和M4**-Weekly平均21.5%），以及TimesNet（例如，总体10%，m4 - annual 14.1%, m4 - hour， M4-Daily和M4-**Weekly**平均**30.1%**）。与最近最先进的预测模型，N-HiTS和PatchTST相比，TIME-LLM在主干LLM上没有任何参数更新的情况下表现出可比或更好的性能。

此外，我们将TIME-LLM与M3-Quarterly数据集上表现最好的模型进行了比较分析，结果见表13。我们提供了额外的指标，即MRAE和MAPE，以及M3竞赛中使用的默认SMAPE。在该数据集上，与TimesNet和PatchTST相比，TIME-LLM的性能达到了相当的水平，远远优于GPT4TS，在**SMAPE、MRAE和**MAPE上分别降低了23%、**35%**和26%以上。

E few-shot和zero-shot预测

E.1次少射预测

我们在few-shot预测任务中的完整结果详见表14和表15。在10%的少次学习范围内，TIME-LLM在35个案例中的32个案例中确保了SOTA性能，跨越了7个不同的时间序列基准。我们的方法的优势在5%的少射情况下变得更加明显，在32例中有21例实现了SOTA结果。我们将此归因于我们重新编程的LLM中成功的知识激活。

18

作为ICLR 2024会议论文发表

表11：在长期预测任务中与其他基线的额外比较。我们设置ILI的预测视界H∈{24,36,48,60}，其他的则为{96,192,336,720}。值越低，表示性能越好。**红色**：最好，蓝色：次好。

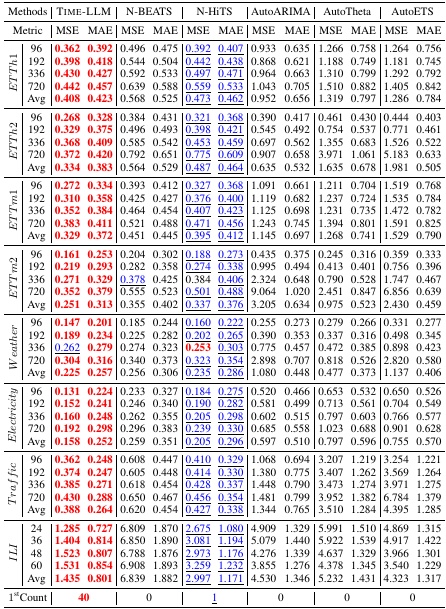
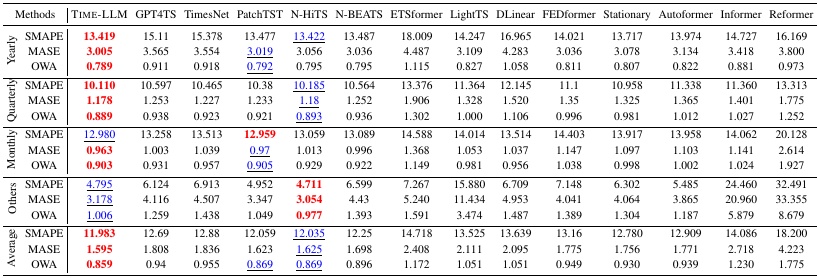


表12：全短期时间序列预测结果。预测视界为[6,48]，后三行为不同采样间隔下所有数据集的加权平均。值越低表示性能越好。**红色**：最好，蓝色：次好。



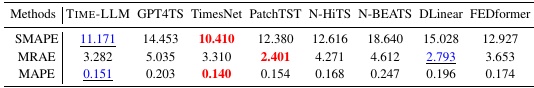
E.2零概率预测

表16总结了零次预测的完整结果。TIME-LLM在零射击自适应方面明显优于6种最具竞争力的时间序列模型。总体而言，我们观察到所有基线的**MSE和**MAE平均降低了23.5%和**12.4%**。我们的改进在典型的跨域场景（例如，ETTh2→ETTh1和ETTm2→ETTm1）上持续显著，**平均w.r.t、MSE**和MAE分别超过20.8%和11.3%。值得注意的是，与LLMTime （Gruver等人，2023）相比，TIME-LLM表现出卓越的性能提升，LLMTime采用了类似大小的主干LLM (7B)，是利用llm的最新努力

19

作为ICLR 2024年会议论文发表

表13:M3 （Quarterly）上额外的短期时间序列预测结果。预测视界为8。值越低表示性能越好。**红色**：最好，蓝色：次好。



零概率时间序列预测。我们将这一成功归因于我们的重编程框架在执行时间序列任务时，能够以资源高效的方式更好地激活LLM的知识转移和推理能力。

F消融研究

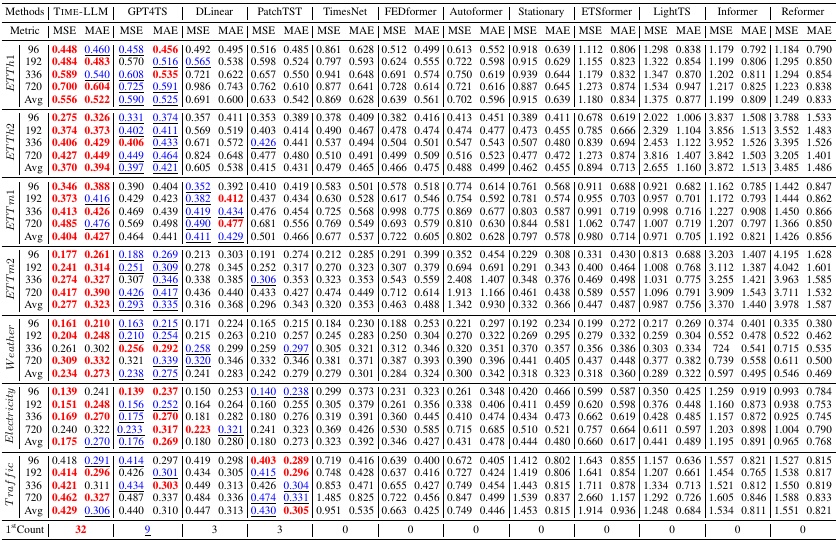
全烧蚀结果见表17。我们还比较了重编程和微调（与QLoRA Dettmers等人（2023））协议下的模型性能。我们的结果表明，与QLoRA变体**（a .5）**相比，我们的方法的性能平均**提高了19%**。

G与模型微调的效率比较

**设置。**我们比较了模型微调（与QLoRA Dettmers等人（2023））的效率和本节中我们提出的模型重编程与两个不同的主干，即1/4容量的Llama（前8个Transformer层）和全容量。在这里，我们坚持ETTh1上的长期预测协议来预测两个不同的步骤（即本例中的96和336）。对于评估指标，我们报告可训练参数的总数（以百万计），GPU内存（以兆字节计）和运行时间（每次迭代秒）。

**结果。**我们的结果见表18。我们发现，与远程QLoRA的参数有效微调（PEFT）相比，模型重编程显著提高了效率

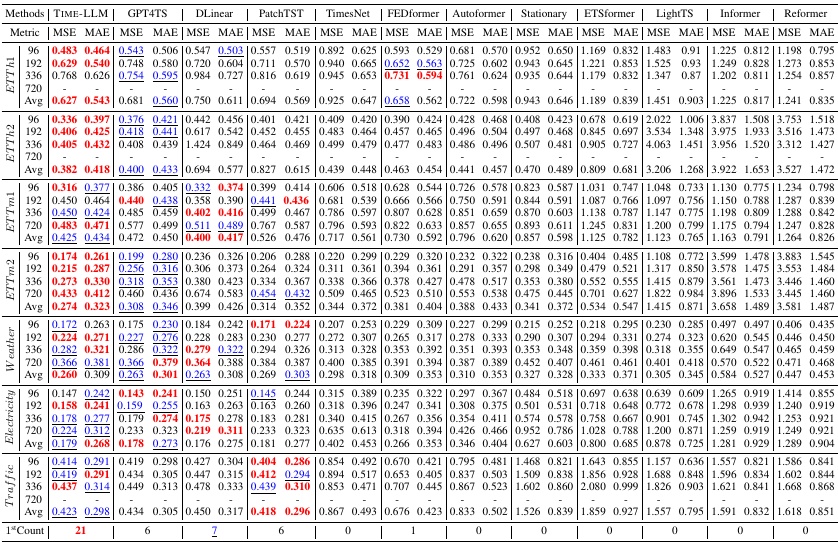
表14:10%训练数据上完整的few-shot学习结果。我们使用与表1相同的协议。



20

作为ICLR 2024会议论文发表

表15:5%训练数据上完整的few-shot学习结果。我们使用与表1相同的协议。“-”表示5%的时间序列不足以构成训练集。



根据可训练参数的总数、GPU内存开销和训练速度来预测任务。在数量上，四种场景下的可训练参数平均减少**71.2%**，导致内存消耗减少**23.1%**，训练速度提高**25.3%**。

H误差条

所有的实验都进行了三次，我们在这里给出了我们的模型和亚军模型的标准差。我们的方法与第二好的方法PatchTST （Nie等人，2023）在长期预测任务上的比较见表19。在此表中，四个ETT数据集的平均MSE和MAE均已报告，并附有标准差。此外，表20将我们的方法与第二好的方法N-HiTS （Challu et al., 2023a）的有效性进行了对比，采用不同的M4数据集进行比较。

我可视化

在本部分中，我们将TIME-LLM在各种场景下的预测结果与最先进的代表性方法(例如GPT4TS （Zhou等人，2023a), PatchTST （Nie等人，2023）和Autoformer (Wu等人，2021)）进行了可视化比较，以证明TIME-LLM的优越性能。

在图7和图8中，将各种方法的长期（input-96-predict-96）和短期（input-36-predict-36）预测结果与地面真实情况进行了比较。在这里，TIME-LLM展示了与GPT4TS， PatchTST和经典的基于变压器的方法Autoformer相比，其预测精度明显优于前者。

我们还提供了在少量射击和零射击场景下预测结果的视觉比较，如图9和图10所示。我们坚持长期（input-96-predict-96）预测设置

21

作为ICLR 2024会议论文发表

表16:ETT数据集上的全零次学习结果。值越低表示性能越好。**红色**：最好，蓝色：次好。

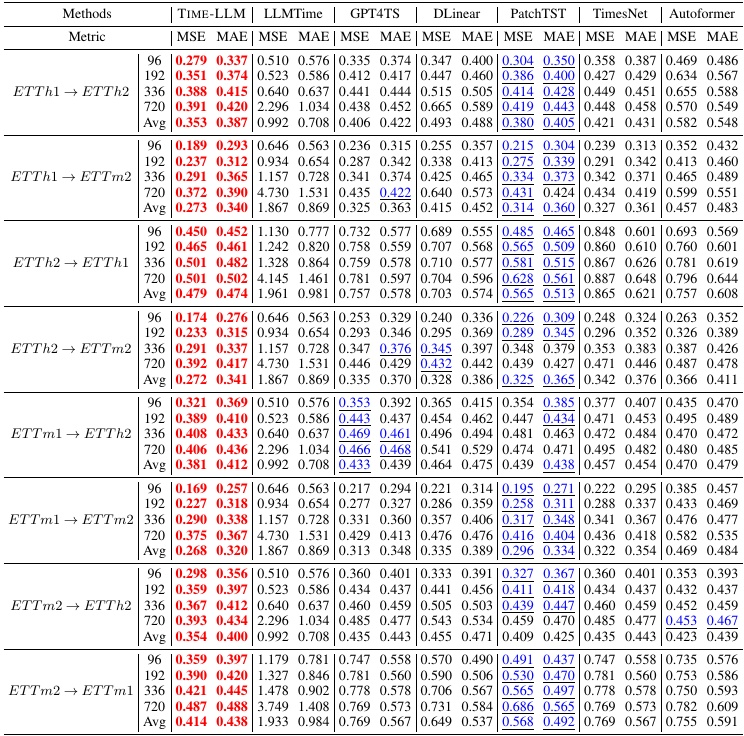
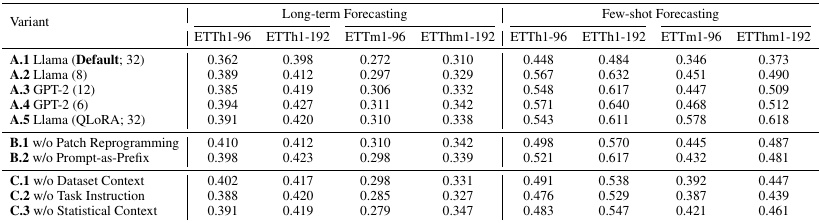


表17:ETTh1和ETTm1的完全消融预测提前96步和192步（MSE报告）。



在两种情况下。TIME-LLM在使用有限数据进行预测方面表现出显著的优势——与GPT4TS相比，这一事实变得尤为突出。

22

作为ICLR 2024会议论文发表

表18：在ETTh1数据集上使用QLoRA （Dettmers et al., 2023）进行模型重编程和参数高效微调（PEFT）预测两个不同步骤的效率比较。



表19：我们的方法和次优方法（PatchTST）在所有时间序列数据集上用于长期预测的标准差。

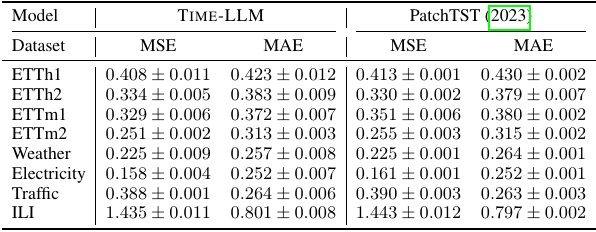
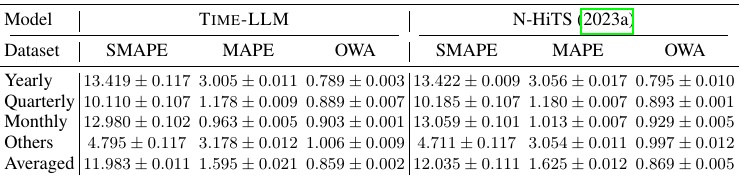


表20：我们的TIME-LLM和次优方法（N-HiTS）在M4数据集上用于短期预测的标准差。



23

作为ICLR 2024会议论文发表

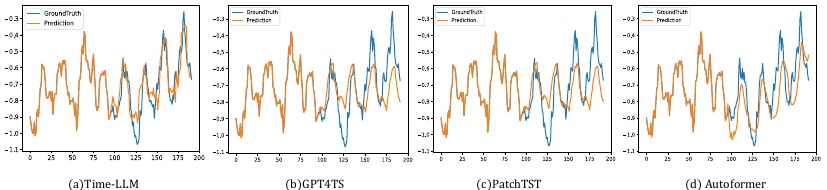


图7：在input-96-predict- 96设置下，不同模型对ETTh1的长期预测案例。蓝线为基础事实，橙线为模型预测。

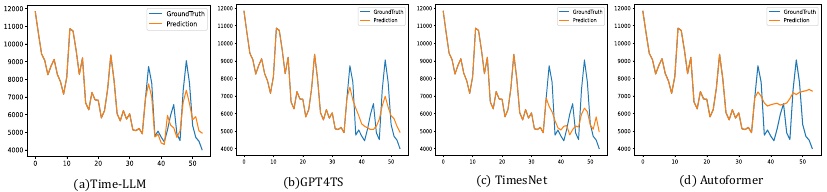


图8：在input-36-predict- 18设置下，不同模型对M4数据集的短期预测。

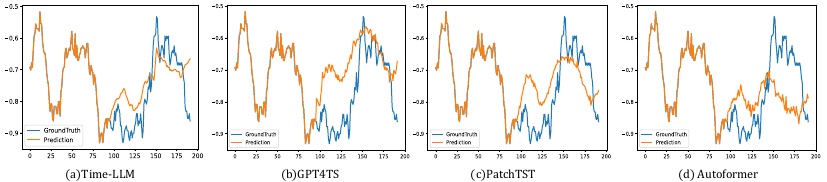


图9：在input-96-predict- 96设置下，不同模型对ETTm1的少量预测案例。蓝线为基础事实，橙线为模型预测。

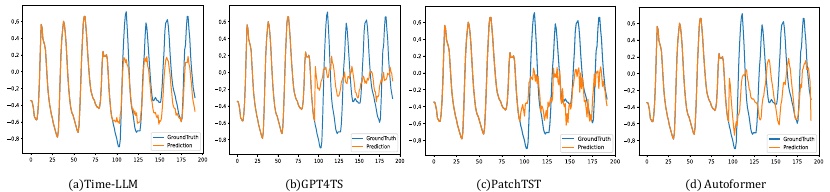


图10:input- 96-predict-96设置下，不同模型从ETTh1→ETTh2的零次预测案例蓝线为基础真相，橙线为模型预测。

24