基于 PatchTST 改进的多变量时间序列预测模型

王乐

2024 年 1 月 10 号

摘要

本文深入探讨了 Transformer 在时间序列预测在的应用，特别是在金融市场分析、气象预报和能源消耗等关键领域。文中主要工作是完成对基于 Transformer 模型的变体——PatchTST 模型的复现以及改进方向的思考与实践。该模型通过将时间序列分割成子序列级的“Patch”来有效降低计算和存储复杂性，同时提高长期预测的准确性。在实验部分，本文不仅在标准数据集上验证了 PatchTST 模型的效果，而且还对模型的关键组件进行了创新和优化，改变了 Patch 长度，并通过多头 Patch 设计增强了模型的预测能力。实验结果显示，经过这些改进后的模型在多个预测窗口长度上的性能都有所提升，尤其是在汇率数据集上表现出色。最后，本文还探讨了未来的研究方向，包括如何使模型自适应不同振幅的时间序列，以进一步提高模型的实用性和适应性。

关键词：Patch；通道独立；Transformer

1 引言

随着人工智能和机器学习领域的迅速发展，时间序列预测已成为研究的热点，尤其是在金融市场分析、天气预测、能源消耗等多个领域中的应用。传统的时间序列预测方法，如自回归模型（ARIMA），虽然在某些场景下有效，但在处理大规模、多变量和非线性序列数据时存在限制。近年来，深度学习技术的崛起为这一领域带来了新的解决方案。Transformer 模型，由 Vaswani 等人于 2017 年提出，因其在自然语言处理领域的显著成效而备受关注 [18]。它的核心机制——自注意力（Self-Attention），使得模型能够有效捕捉数据中的长期依赖关系，而本文的研究则是基于 2022 年提出的 Transformer 模型的变体——PatchTST 模型 [12]，该模型是对原始 Transformer 模型的改进，核心在于将时间序列分割成子序列级的“块”（patches），以这些块作为输入 token 用于 Transformer 的输入。这种方法有效地减少了计算和存储的复杂性，同时提高了长期预测的准确性。

2 相关工作

时间序列预测是许多领域的关键组成部分，例如传感器网络监测 [13]、能源和智能电网管理、经济和金融 [22] 以及疾病传播分析 [11]。通常用于时间序列预测任务的经典工具有：ARIMA [1] 通过差分将非平稳过程转变为平稳过程来解决预测问题，使用过滤方法用于序列

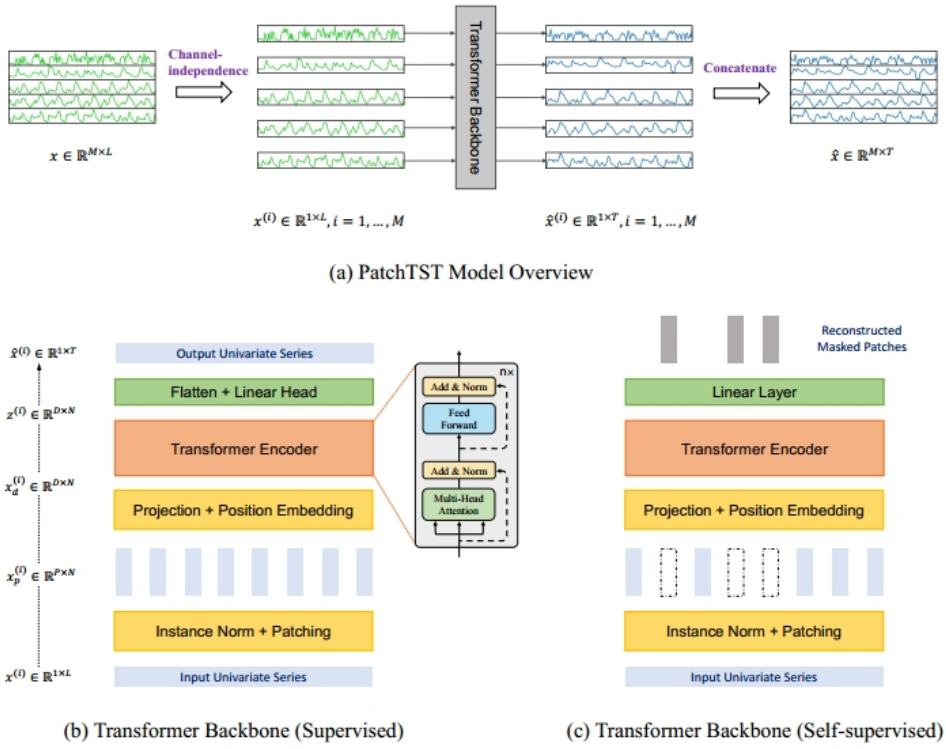
预测 [6]，循环神经网络（RNN）模型对时间序列的时间依赖性进行建模 [14] [10]，DeepAR [15] 结合自回归方法和 RNN 对未来序列的概率分布进行建模，LSTNet [7] 引入具有循环跳跃连接的卷积神经网络（CNN）来捕获短期和长期时间模式，基于注意力的 RNN [17] 引入时间注意力来探索预测的长程依赖性，基于时间卷积网络（TCN）[13] 的工作尝试用因果卷积来建模时间因果关系等等。而近几年对于深度学习的研究中，又有 Transformer 在自然语言处理（NLP）[16]、计算机视觉（CV）[3]、语音 [5] 以及最近的时间序列 [4] 等各个应用领域取得了巨大成功，受益于其注意力机制，可以自动学习序列中元素之间的连接，因此成为顺序建模任务的理想选择。

近年来，有大量的工作试图应用 Transformer 模型来预测长期时间序列，像 LogTrans [8] 使用具有 LogSparse 设计的卷积自关注层来捕获局部信息并降低空间复杂度；Informer [20] 提出了一种带有蒸馏技术的 ProbSparse 自关注，以有效地提取最重要的关键字；Autoformer [19] 借鉴了传统时间序列分析方法的分解和自相关思想；FEDformer [21] 采用傅里叶增强结构获得线性复杂度；Pyraformer [9] 采用具有尺度间和尺度内连接的金字塔式注意力模块，同样具有线性复杂度等等。这些模型大多侧重于设计新的机制来降低原有注意机制的复杂性，从而达到预测长度较长情况下更好的预测效果。然而，大多数模型使用逐点关注，忽略了 patch 的重要性。LogTrans [8] 避免了键和查询之间逐点的点积，但其值仍然基于单个时间步长，Autoformer [19] 使用自相关来获得补丁级连接，但它是一种手工设计，不包括补丁内的所有语义信息，Triformer [2] 提出了 patch 关注，但其目的是通过使用伪时间戳作为 patch 内的查询来降低复杂性，因此它既没有将 patch 作为输入单元，也没有揭示其背后的语义重要性。

3 本文方法

3.1 本文方法概述

本文提出了一种有效的基于变压器的多元时间序列预测和自监督表示学习模型设计。它基于两个关键组件:(i) 将时间序列分割成子序列级补丁，作为 Transformer 的输入令牌;(ii) 通道独立，其中每个通道包含单个单变量时间序列，该序列在所有序列中共享相同的嵌入和 Transformer 权重。Patch 设计有以下三方面的好处：在嵌入中保留了局部语义信息; 在相同的回望窗口下，注意图的计算量和内存使用量呈二次减少; 而且该模型可以关注更长的历史。将该模型应用于时序预测中，则是考虑以下问题: 给定一个多变量时间序列样本集合，具有回溯窗口 $L : (x_1, ..., x_L)$，其中每个 x_t 在时间步长 t 是一个维数 M 的向量，我们想要预测未来 t 的值 $(x_{L+1}, ..., x_{L+T})$ 我们的 PatchTST 如图1所示所示，其中模型使用了普通的 Transformer 编码器作为其核心架构。



(a) PatchTST Model Overview

(b) Transformer Backbone (Supervised)

(c) Transformer Backbone (Self-supervised)

图 1. PatchTST 架构。(a) 将多变量时间序列数据分成不同的通道。它们共享相同的 Transformer 主干，但是前向进程是独立的。(b) 通过实例归一化算子对每个通道单变量序列进行分割。这些补丁被用作 Transformer 输入令牌。(c) 使用 PatchTST 的掩码自监督表示学习，其中随机选择补丁并将其设置为零。该模型将重建被掩盖的斑块。

具体来说，该模型的工作流程如图2：

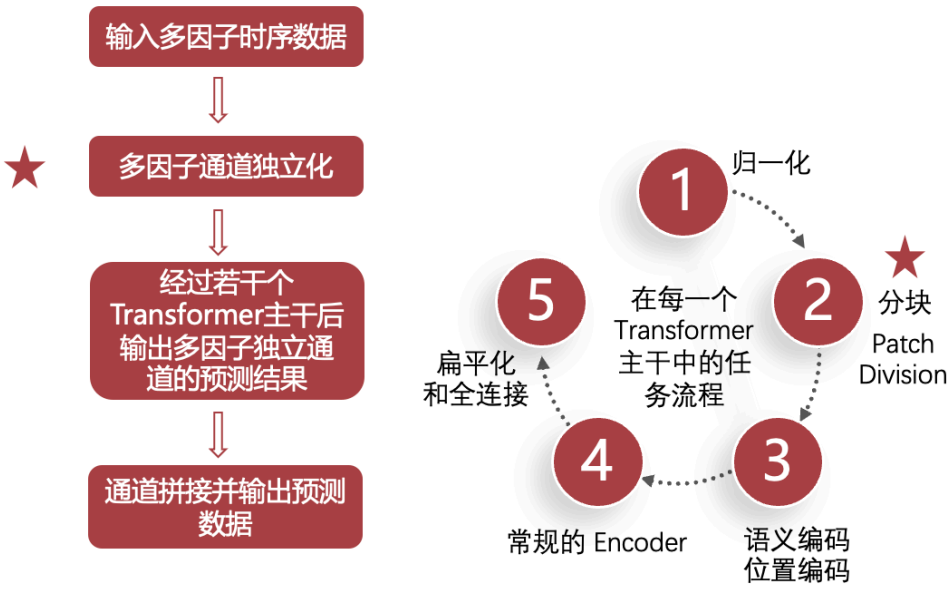


图 2. PatchTST 工作流程

3.2 模型结构

向前过程（Forward Process）：我们表示从时间索引 1 开始的长度为 L 的第 i 个单变量序列为 $x_{1:L}^{(i)} = (x_1^{(i)}, \ldots, x_L^{(i)})$，其中 $i = 1, \ldots, M$. 输入 $(x_1, \ldots, x_L)$ 被分割成 M 个单变量序列 $x^{(i)} \in \mathbb{R}^{1\times L}$, 其中每个序列分别根据我们的频道独立设置。然后 Transformer 骨干网将提供预测结果：$\mathbf{x}^{(i)} = \left(x_{L+1}^{(i)}, \ldots, x_{L+T}^{(i)}\right) \in \mathbb{R}^{1\times T}$

时间块（Patch）：首先将每个输入单变量时间序列 $x(i)$ 划分为可以重叠或不重叠的 patch。将 patch 长度记为 P，两个连续 patch 之间的不重叠区域记为 S，则补片过程将生成一个 patch 序列 $\mathbf{x}_p^{(i)} \in \mathbb{R}^{P\times N}$，其中 N 为 patch 个数，$N = \left\lfloor \frac{L-P}{S} \right\rfloor + 2.$。在这里，我们将最后一个值 $x_p^{(i)} \in \mathbb{R}$ 的 S 个重复数填充到原始序列的末尾，然后进行修补。通过使用 patch，输入 token 的数量可以从 L 减少到大约 L/S。这意味着注意力图的内存使用量和计算复杂度以 s 的倍数二次降低。因此，在训练时间和 GPU 内存的约束下，patch 设计可以让模型看到更长的历史序列，从而显著提高预测性能。

编码器（Transformer Encoder）：使用一个普通的变压器编码器，将观察到的信号映射到潜在表征。通过可训练的线性投影 $\mathbf{W}_p \in \mathbb{R}^{D\times P}$ 将 patch 映射到 D 维的 Transformer 潜在空间，并使用可学习的加性位置编码 $W_{pos} \in \mathbb{R}^{D\times N}$ 来监控 patch 的时间顺序：$\tilde{x}^{(i)} = W_p x_p^{(i)} + W_{pos}$，其中 $\tilde{x}^{(i)} \in \mathbb{R}^{D\times N}$ 表示将被馈入图 1 中的 Transformer 编码器的输入。然后每个头 $h = 1, \ldots, H$ 在多头关注中将它们转化为查询矩阵 $Q_h^{(i)} = (\tilde{x}^{(i)})^T W_h^Q$，键矩阵 $K_h^{(i)} = (\tilde{x}^{(i)})^T W_h^K$，以及值矩阵 $V_h^{(i)} = (\tilde{x}^{(i)})^T W_h^V$，其中 $W_h^Q, W_h^K, W_h^V \in \mathbb{R}^{D_k\times D}$。在使用缩放点积后用于获取注意力输出 $O_h^{(i)} \in \mathbb{R}^{D\times N}$，通过如下计算：

$$(O_h^{(i)})^T = \text{Attention}(Q_h^{(i)}, K_h^{(i)}, V_h^{(i)}) = \text{Softmax}\left(\frac{(Q_h^{(i)})^T (K_h^{(i)})^T}{\sqrt{D_k}}\right) V_h^{(i)}$$

损失函数（Loss Function）：我们选择使用 MSE 损失来衡量预测与实际情况之间的差异。每个通道的损失被收集并在 M 个时间序列上平均，以得到总体目标损失:

$$L = \frac{1}{M} \sum_{i=1}^{M} \left\| \hat{x}_{L+1:L+T}^{(i)} - x_{L+1:L+T}^{(i)} \right\|^2$$

实例标准化（Instance Normalization）该技术用以帮助减轻训练数据和测试数据之间的分布偏移效应。它简单地将每个时间序列实例 x (i) 归一化，平均值为零，单位标准差为零。本质上，我们在修补之前对每个 x (i) 进行归一化，并将平均值和偏差添加回输出预测中。

4 复现细节

4.1 与已有开源代码对比

在复现过程中，我参考了由清华大学团队提供的时序预测库中对 PatchTST 模型的复现代码，在此基础上进行了相应数据集的复现工作以及后续的改进等。根据原论文以及复现代码，得到 PaTchTST 的算法步骤见算法1：

在此基础上，我首先进行了数据集的迁移——汇率（exchange）数据集，在此数据集上分别得到了 PatchTST 模型下的预测精度与其他基于 Transformer 的 SOTA 时序预测模型的预测精度，对比分析实验结果。其次，考虑到原文中所用的 patch 长度固定为 16，但是从实际

Algorithm 1 PatchTST Model Pseudocode

Input: Multivariate time series $X \in \mathbb{R}^{M \times T}$
Output: Forecasted time series $\hat{Y} \in \mathbb{R}^{M \times T}$
procedure PatchTST(X)
    $X' \leftarrow \text{InstanceNorm}(X)$
    $P \leftarrow \text{Patchify}(X')$
    $Z^0 \leftarrow \text{PositionEmbedding}(P)$
    for $i = 1$ to M **do**
        $Z^i \leftarrow \text{TransformerEncoder}(Z^{i-1})$
        $Z^i \leftarrow Z^i + \text{FeedForward}(Z^i)$
    end for
    $Z^M \leftarrow \text{Flatten}(Z^M)$
    $\hat{Y} \leftarrow \text{Linear}(Z^M)$
    return $\hat{Y}$
end procedure

应用的角度出发，我们知道对于不同的时间序列数据，数据可能呈现的振幅是不一样的，因此在原代码的基础上我对 patch 长度进行了更改，对比分析了不同 patch 长度下 PatchTST 模型的预测精度。最后，我将上述实验中预测效果更佳的几个 patch 长度进行混合，作为多头的 patch 分别输入对应的 transformer 架构中，最后进行拼接得到最终的预测结果，并思考未来可以改进的方向。

4.2 数据集介绍

本文在在 8 个流行的数据集上评估了文章所提出的 PatchTST 的性能，包括天气、交通、电力、ILI 和 4 个 ETT 数据集 (ETTh1、ETTh2、ETTm1、ETTm2)。这些数据集已被广泛用于基准测试，并公开提供 (Wu et al .，2021)。此外在后续的实验中本文还将该模型迁移到汇率数据集上做了进一步的研究与分析。

表1总结了这些数据集的统计数据，分别是天气、交通和电力等大型数据集，它们有更多的时间序列，因此结果会比其他较小的数据集更稳定，更不容易过度拟合。

表 1. 数据集介绍

数据集	Weather	Traffic	Electricity	ILI	ETH1	ETH2	ETTm1	ETTm2	Exchange
特征数	21	862	321	7	7	7	7	7	8
时间步长	52696	17544	26304	966	17420	17420	69680	69680	3280

Weather：2020 年每十分钟记录一次的 21 个天气指标;
Traffic：包含 2015 年-2016 年旧金山高速公路传感器每小时记录的道路占用率;
Electricity：从 2012 年到 2014 年收集的 321 个客户每小时电力消耗;
ILI：包括了 2002 年至 2021 年美国疾病控制和预防中心每周的流感患者比率;

ETT（H、M）：包含 2016 年至 2018 年的 7 种石油和电力变压器的负载特征（小时级、分钟级）；

Exchange：收集了 1990 年到 2016 年 8 个国家的汇率。

4.3 基线选择以及变量设置

基线设置：本文选择基于 Transformer 的 SOTA 模型，包括 FEDformer (Zhou 等人，2022)、Autoformer (Wu 等人，2021)、Informer (Zhou 等人，2021)、Pyraformer (Liu 等人，2022)、LogTrans (Li 等人，2019) 和最近的非基于变压器的模型 DLinear (Zeng 等人，2022) 作为对比基线。所有模型都遵循与原始论文相同的实验设置，ILI 数据集的预测长度 $T \in [24, 36, 48, 60]$，其他数据集的预测长度 $T \in [96, 192, 336, 720]$。我们收集了 Zeng 等人 (2022) 的基线结果，对于基于变压器的模型，默认回溯窗口 $\mathrm{L} = 96$，对于 DLinear 模型，默认回溯窗口 $\mathrm{L} = 336$。但为了避免低估基线，我们还对六个不同的回看窗口 $L \in [24, 48, 96, 192, 336, 720]$ 运行 FEDformer, Autoformer 和 Informer，并始终选择最佳结果来创建强基线。本文计算多元时间序列预测的 MSE（均方误差）和 MAE（平均绝对误差）作为指标。

变量设置：本文在进行实验过程中的参数设置见表2:

表 2. 变量设置

参数名称	值
dropout	0.1
patch 大小	8/16/32/64
stride	8
encoder 块个数	8
decoder 块个数	8
学习率	0.0001
epochs	10
批量处理大小	48
预测时间长度	96/192/336/720

4.4 创新点及未来工作

Patch 长度：更改了原文中 Patch 的固定长度，原文中只给定了 Patch=16 时的情况，且并未说明 Patch 的选择方法，因此本文中尝试了多种 Patch 长度，以图找到更适合的 Patch 对时间序列进行更合理的分块。

多头 Patch：改变 Patch 长度后，分析不同的 Patch 在不同长度的预测窗口下的预测精度，将表现较好的几个 Patch 长度作为多头 Patch，将时间序列划分为不同长度的子序列，分别进入到后续的 Transformer 架构中进行处理，再经过展平、拼接、全连接等操作得到最终的预测结果。

未来改进方向：由于不同时间序列的振幅不同，因此固定模型的 Patch 本就使模型效果受限，因此后续工作中我将重点研究如何使模型自识别不同振幅的时间序列（像小波变换、傅里叶变换等识别序列频率的方法），自动挑选相应的振幅作为模型的 patch 输入，以提高模型的实用性与适用性。

5 实验结果分析

表3中展示了复现的原文几个基准数据集在不同长度的预测窗口下得到的以 MSE 和 MAE 为评价指标的实验结果：

表 3. 基准数据集的实验结果

数据集	Weather		ECL		ETTh1		Traffic	
预测窗口	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	0.174	0.215	0.181	0.270	0.378	0.396	0.458	0.294
192	0.221	0.257	0.187	0.276	0.422	0.425	0.464	0.295
336	0.280	0.297	0.203	0.291	0.462	0.448	0.482	0.304
720	0.357	0.347	0.245	0.325	0.498	0.483	0.515	0.321

表4中展示了在汇率数据集上 PatchTST 模型与 Informer、Autoformer、FEDformer、LightTS 这几个 SOTA 模型以以 MSE 和 MAE 为评价指标的实验结果：

表 4. 迁移数据集的实验结果对比

数据集	预测窗口	**PatchTST**		**Informer**		**Autoformer**		**FEDformer**		**Dlinear**		**LightTS**	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange rate	96	**0.089**	**0.206**	0.823	0.731	0.152	0.306	0.180	0.324	0.110	0.266	0.215	0.370
	192	**0.177**	**0.299**	0.808	0.745	0.268	0.408	0.392	0.476	0.218	0.376	0.442	0.533
	336	**0.299**	**0.396**	1.533	1.040	0.484	0.544	0.562	0.573	0.387	0.497	0.758	0.690
	720	0.883	0.705	3.878	1.778	1.151	0.835	1.438	0.934	**0.713**	**0.662**	0.928	0.764

从表4中可以看出 PatchTST 模型的预测精度显著优于其他 SOTA 模型，即在汇率数据集上的迁移也取得了十分客观的预测精度提升效果。

下面更改 Patch 的长度 P，得到 P=8，P=16，P=32，P=64 时模型以 MSE 和 MAE 为评价指标的实验结果见表5。

表5可以看出，当 Patch 的长度为 8、16、64 时的预测精度在不同长度的预测窗口上有了一定的提升，因此下面选择这三个 Patch 长度作为多头 Patch 的输入，图3中展示了多头 Patch 输入的具体过程，即分别用这三个 Patch 长度处理时间序列，再经过经典的 Transformer 层，通过展开、拼接后，经全连接层得到最终预测结果。

表 5. 不同 Patch 长度下结果对比

数据集	预测窗口	P=8		P=16		P=32		P=64	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange rate	96	0.087	0.203	0.089	0.206	0.088	0.206	**0.082**	**0.198**
	192	**0.177**	**0.299**	**0.177**	**0.299**	0.182	0.304	0.186	0.305
	336	0.302	0.398	**0.299**	**0.396**	0.326	0.412	0.332	0.417
	720	0.889	0.711	0.883	0.705	0.907	0.716	**0.872**	**0.703**

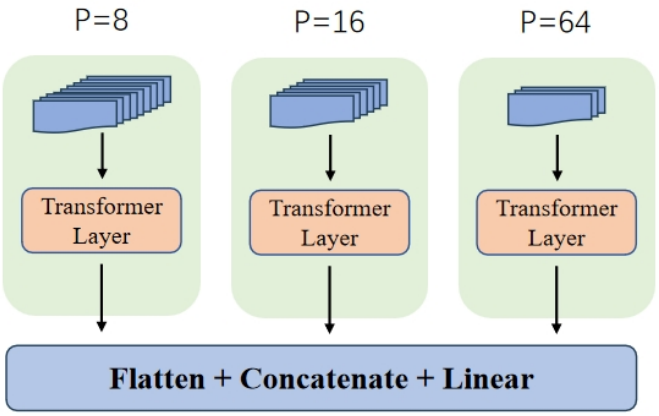


图 3. PatchTST 工作流程

得到的实验结果与原文选择的固定 Patch 长度所得的实验结果的对比见表6：

表 6. 不同 Patch 长度下结果对比

数据集	预测窗口	Multi-Patch		P=16	
		MSE	MAE	MSE	MAE
Exchange rate	96	**0.087**	**0.203**	0.089	0.206
	192	**0.176**	**0.299**	0.177	0.299
	336	0.305	0.401	**0.299**	**0.396**
	720	**0.869**	**0.699**	0.883	0.705

从表6可以得知，经过多头 patch 输入后，模型的预测精度在大部分长度的预测窗口得到了一定的提升，可以认为本文的改进是有效的。

6 总结与展望

本文针对当前时间序列预测的研究热潮，特别是其在金融市场分析、气象预报、能源消耗等关键领域的应用，进行了深入的探讨和研究。尽管传统的时间序列预测方法（如 ARIMA）在特定情况下有效，但面对复杂多变的大规模非线性序列数据时，却显得力不从心。深度学习的兴起，尤其是 Transformer 模型及其变体 PatchTST 的提出，为这一挑战提供了新的解决思路。本文基于 PatchTST 模型，提出了一种有效的多元时间序列预测和自监督表示学习

方法，通过将时间序列分割成子序列级的补丁并实现通道独立处理，显著降低了注意力计算的复杂度，同时增强了模型对长期依赖关系的捕捉能力。

在对相关工作的回顾中，本文不仅总结了经典方法，还细致考察了基于深度学习的各种尝试，尤其是 Transformer 在多个领域的成功应用，以及各种模型在降低复杂度、增强长期预测能力方面的创新点。本文在复现过程中利用了 PatchTST 的优势，并在实验中对不同的 Patch 长度进行了尝试，发现合适的 Patch 长度对提高预测精度至关重要。此外，多头 Patch 的引入进一步增强了模型对不同时间尺度依赖性的理解，从而改善了预测性能。

展望未来，本文认为自动调整 Patch 长度以匹配不同时间序列振幅的策略将是一个有意义的研究方向。类似于小波变换和傅里叶变换在频率识别中的应用，自适应 Patch 选择机制可以根据时间序列的固有特性动态调整，进一步提升模型在各种实际应用场景中的适用性和精确度，实现真正智能的时间序列预测模型，满足日益增长的实际需求。

在实验结果的分析中，本文选择复现的模型在多个基准数据集上都显示出了优越的性能，尤其是在汇率数据集上的应用表现，证实了所提出方法的有效性。而改进后的模型不仅在预测精度上有所提升，更在理解不同时间序列结构上展示了其独到的优势。

综上所述，本文对 PatchTST 模型的复现及改进过程，在理论和实践两个层面上都取得了显著的成果。在未来的工作中，我将继续探索自动 Patch 长度调整机制，并在更多实际应用中验证模型的泛化能力和实用价值。

参考文献

[1] George EP Box and Gwilym M Jenkins. Some recent advances in forecasting and control. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 1968.

[2] Razvan-Gabriel Cirstea, Chenjuan Guo, Bin Yang, Tung Kieu, Xuanyi Dong, and Shirui Pan. Triformer: Triangular, variable-specific attentions for long sequence multivariate time series forecasting. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 1994–2001, 2022.

[3] Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*, 2021.

[4] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al. A comparative study on transformer vs rnn in speech applications. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 449–456. IEEE, 2019.

[5] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 2021.

[6] Richard Kurle, Syama Sundar Rangapuram, Emmanuel de Bézenac, Stephan Günnemann, and Jan Gasthaus. Deep rao-blackwellised particle filters for time series forecasting. In *Advances in Neural Information Processing Systems*, 2020.

[7] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018.

[8] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[9] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2022.

[10] Danielle C Maddix, Yuyang Wang, and Alex Smola. Deep factors with gaussian processes for forecasting. *arXiv preprint arXiv:1812.00098*, 2018.

[11] Yasuko Matsubara, Yasushi Sakurai, Willem G. van Panhuis, and Christos Faloutsos. Funnel: automatic mining of spatially coevolving epidemics. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 105–114. ACM, 2014.

[12] Yuchen Nie, Ngoc Hoang Nguyen, Panuwat Sinthong, et al. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2022.

[13] Spiros Papadimitriou and Philip S. Yu. Optimal multi-scale patterns in time series streams. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 647–658. ACM, 2006.

[14] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. In *Advances in Neural Information Processing Systems*, 2018.

[15] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 2020.

[16] Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. In *Advances in Neural Information Processing Systems*, 2019.

[17] Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, 2019.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[19] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 2021.

[20] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115, 2021.

[21] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proc. 39th International Conference on Machine Learning*, 2022.

[22] Yunyue Zhu and Dennis E. Shasha. Statstream: Statistical monitoring of thousands of data streams in real time. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 358–369, 2002.