计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
[doi: 10.15888/j.cnki.csa.009806] [CSTR: 32024.14.csa.009806]
©中国科学院软件研究所版权所有.
E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

结合因果卷积的非平稳学习倒置 Transformer 的时间序列预测模型①

李子烨, 乔钢柱

(中北大学 计算机科学与技术学院, 太原 030051)
通信作者: 乔钢柱, E-mail: qiaogangzhu@sohu.com

摘　要: 针对当前时间序列预测任务中存在多维特征建模困难、数据非平稳、预测准确性要求高等问题, 提出结合因果卷积的非平稳学习倒置 Transformer 模型. 该模型首先利用倒置嵌入时间序列数据交换注意力机制和前馈神经网络原有功能, 使用注意力机制学习时间序列数据的多元相关性, 前馈神经网络学习时间序列的时间依赖性, 在多维时间序列时间及变量上建模, 增强模型在时间维度和变量间关系的泛化能力, 从而提高模型的可解释性. 然后, 利用序列平稳化模块解决数据非平稳性问题以提高模型的可预测能力. 最后使用结合因果卷积的非平稳学习注意力机制将平稳化模块中消失的关键特征与信息重新引入, 从而提高模型的预测准确性. 与 PatchTST、iTransformer、Crossformer 等多个主流基准模型进行比较, 所提模型在 Exchange 等 4 个数据集上的均方误差平均下降了 6.2%–65.0%. 通过消融实验表明本文的倒置嵌入模块、结合因果卷积的非平稳学习注意力模块能有效提升时间序列预测的准确度.

关键词: 多维时间序列; 倒置嵌入; 序列平稳化; 因果卷积; 非平稳学习

引用格式: 李子烨,乔钢柱.结合因果卷积的非平稳学习倒置 Transformer 的时间序列预测模型.计算机系统应用. http://www.c-s-a.org.cn/1003-3254/9806.html

Time Series Prediction Model Combining Non-stationary Learning Inverted Transformer with Causal Convolution

LI Zi-Ye, QIAO Gang-Zhu

(School of Computer Science and Technology, North University of China, Taiyuan 030051, China)

Abstract: Aiming at the difficulties in multi-dimensional feature modeling, non-stationary data and high prediction accuracy requirements in current time series prediction tasks, a non-stationary learning inverted Transformer model combined with causal convolution is proposed. The model first uses the original functions of the inverted embedding exchange attention mechanism for time series data and feedforward neural network. It employs the attention mechanism to learn the multivariate correlation of time series data and the feedforward neural network to learn the time dependence of the time series. Modeling the time and variables of multi-dimensional time series enhances the generalization ability of the model in terms of time dimension and the relationship between variables. Thus, the interpretability of the model is improved. Then, the sequence stabilization module is used to solve the problem of data non-stationarity to improve the predictability of the model. Finally, the non-stationary learning attention mechanism combined with causal convolution is used to reintroduce the key features and information that vanish in the stabilization module, thereby enhancing the prediction accuracy of the model. Compared with multiple mainstream benchmark models including PatchTST, iTransformer, and Crossformer, the mean square error of the proposed model on four data sets such as Exchange

① 基金项目: 山西省基础研究计划联合资助项目 (太重) (TZLH20230818007)
收稿时间: 2024-09-20; 修改时间: 2024-10-21; 采用时间: 2024-10-30; csa 在线出版时间: 2025-02-25

decreases by 6.2% to 65.0% on average. Ablation experiments show that the inverted embedding module and the non-stationary learning attention module combined with causal convolution can effectively improve the accuracy of time series prediction.

Key words: multidimensional time series; inverted embedding; sequence stabilization; causal convolution; non-stationary learning

时间序列数据是按照不同时间间隔依次、连续产生的数据, 这类数据往往包含丰富且庞杂的信息, 隐含某些客观事物的发展趋势和变化规律, 时间序列数据分析与预测对于揭示事物未来的发展趋势和状态具有重要的现实意义. 现今, 时序预测已被广泛应用于国民经济各个领域中, 如气象降雨预测、交通流量预测、金融预测等.

时间序列预测的准确性往往受到多种因素的影响, 例如在气象领域, 温度的变化往往会受到维度位置、大气环流、海陆分布和地形等众多因素的影响. 多维时间序列预测问题中利用好协变量 (多维时间序列预测中将除了预测变量之外的其他变量统称为协变量) 带来的信息可以提高预测的准确性. 目前 Transformer[1] 是时序预测中使用较广的一种模型.

Transformer 模型一经提出后在自然语言处理[2]和计算机视觉[3]两个领域取得巨大成功. 受这些领域成果的启发, 国内外的学者利用 Transformer 描述成对依赖关系和提取序列特征的强大能力[4], 将 Transformer 应用于时序预测领域中, 极大提高了预测准确性. 但近年来线性模型[5]在时序预测领域中性能远超 Transformer, 使得人们对 Transformer 模型在时序预测领域中的有效性产生了怀疑.

究其原因, 现有的多维时间序列预测方法中, Transformer 模型是将协变量与预测变量一同输入到特征空间中进行预测, 其关注重点仍在时间维度的依赖关系上, 这使得 Transformer 模型预测效果准确性不高且模型解释性差. 因此如何在变量维度建模, 也是多维时间序列预测任务的重点之一.

针对该问题, 本文主要研究工作如下.

(1) 采用序列平稳化模块, 将非平稳时间序列变为平稳时间序列, 提高模型的可预测性.

(2) 采用倒置嵌入时间序列, 交换注意力机制与前馈神经网络的功能, 分别在变量维度与时间维度建模, 提高模型预测能力与可解释性.

(3) 采用结合因果卷积的非平稳学习注意力机制, 将平稳序列失去的重要信息与特征重新引入, 提高模型预测的准确性.

1 相关工作

近年来, 随着深度学习方法的不断突破, 深度学习方法在时间序列预测方面取得了长足的进步. 基于 RNN 的模型[6,7]以自回归方式被应用于序列建模, 利用前一时刻的输出作为后一时刻的输入, 并在每个时刻建立反馈连接, 但循环结构可能会受到建模长期依赖性的影响. 随后, Transformer 被提出并展示了强大的序列建模能力和模型可扩展性, 因此成为时间序列预测的骨干网络. 本文将近几年来基于 Transformer 模型的改进方法总结为图 1, 大致可分为 4 类.

第 1 类修改方式主要修改模型内的模块, 特别是注意力模块, 关注长序列的时间依赖性建模和复杂性优化. 例如 Autoformer[8]将分解块融合成规范结构, 并开发自动相关性机制, 打破时间序列分解的预处理传统, 将其改造为深度学习模型的基本内部块. Informer[9]使用 KL 散度标准扩展了自注意力机制. 然而, 线性预测模型的预测效果超过了第 1 类修改方式的模型.

第 2 类修改方式更加关注对时间序列的处理, 例如将数据平稳化、保持通道的独立性. 比如 Non-stationary Transformers[10]关注到序列的平稳性问题, 对序列做了平稳化处理. PatchTST[4]结合 Patch 和 Transformer 架构, 利用通道独立性处理多变量时间序列. 这种方式提升了模型性能, 加强了对多个变量的独立性和交互性的关注.

第 3 类修改方式对 Transformer 的模块进行了改进并增加了新的模块形成新的架构, 例如 Crossformer[11]通过维度-段式嵌入技术将时间序列数据转换为二维向量数组, 同时使用两阶段注意力层来高效地捕获跨时间和跨变量的依赖关系.

第 4 类修改方式通过更改模型内各模块的职责, 来提高模型预测性能. 例如 iTransformer[12]将不同变量的原始序列独立嵌入, 交换注意力模块与前馈神经网络的职责, 取得了优秀的预测性能.

从时间序列的本质属性来看, 现实世界中时间序列数据的统计特性和联合分布往往随着时间变化, 这种非平稳性使得时间序列的可预测性相对较差[13,14]. 且 Transformer 模型并没有很好的利用多维时间序列中协变量的有效信息, 故 Transformer 模型预测效果准确性不高且模型解释性差.

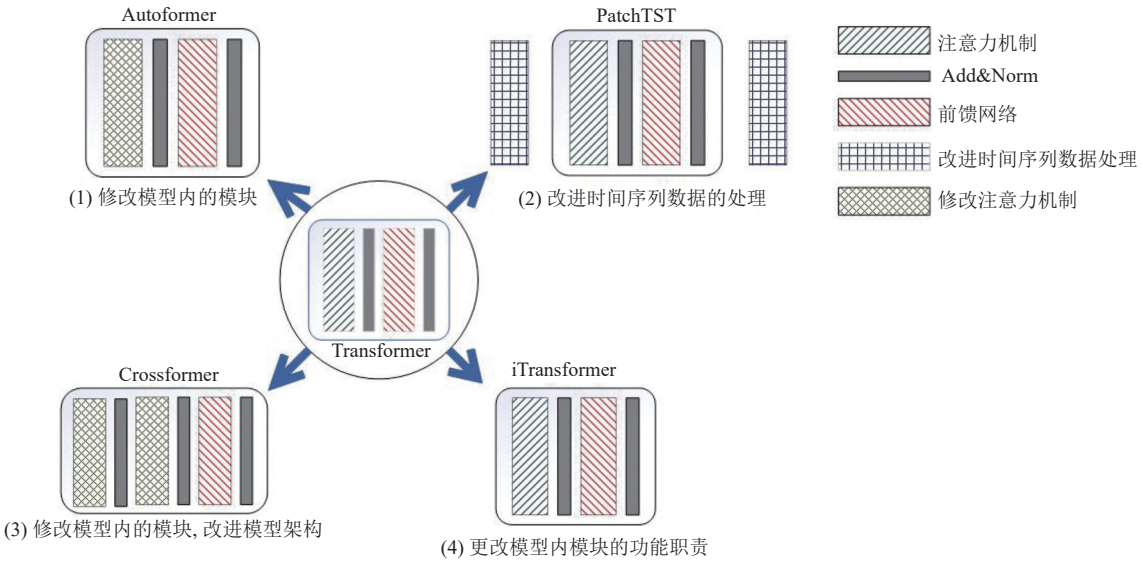

图 1 Transformer 主流修改方式

从 Transformer 模型预测方式来看, 时间序列数据采集时, 由于监视器的延迟, 多维时间序列同一时间步的数据信息可能并非来自同一时间, 具有的物理意义也不同, 且同一时间步的数据拥有的数据信息太过局限. 如图 2 所示, Transformer 模型把同一时间步的变量视作 1 个 token (模型处理的基本数据单位) 进行预测, 这使得模型难以利用有益信息, 消除了多元时间序列变量之间的相关性.

最近提出的 DLinear 模型[5]采用单层线性网络对序列进行建模以用于时间序列预测任务, 并取得了优秀的成果, 表明了线性层擅长学习时间序列具备的时间特征. 这是由于线性层的神经元可以学习到时间序列的内在属性, 例如幅度、周期性, 甚至频率谱. 在 Transformer 模型中前馈神经网络是具有两层线性层的全连接网络, 因此更适合处理时间序列上的时间依赖性.

考虑到上述问题, 将第 2 类修改方式与第 4 类修改方式相结合, 关注数据的本质, 利用平稳化对时间序列进行预处理[15–17], 减弱原始时间序列的非平稳性, 并正确利用 Transformer 架构, 有效提高了模型预测的准确度.

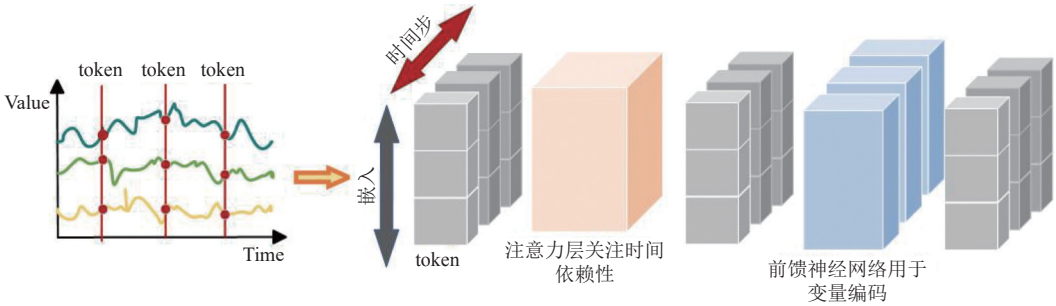

图 2 Transformer 数据嵌入方式

2 模型结构

2.1 问题概述

在多维时间序列预测任务中, 历史观测值 $X=\{x_1,\cdots,x_T\}\in\mathbb{R}^{T\times N}$ 指具有 T 个时间步和 N 个变量的多维序列, 时间序列预测的任务是预测未来的 S 个时间步: $Y=\{x_{T+1},\cdots,x_{T+S}\}\in\mathbb{R}^{S\times N}$. 将 $X_{:,n}$ 表示为第 n 组变量的整个时间序列. 数据采集时, 由于监视器的系统延迟, 同一时间步的数据可能具有不同物理意义, 采集时间可能不对齐且尺度差异大, 强行将它们编码为统一的 token, 这样将消除多变量间的相关性, 并不适合多维时间序列预测任务. 平稳性是时间序列可预测性的一个重要因素. 过去的学术研究中用直接平稳化的方法来减弱序列的非平稳性, 这样虽然获得了更好的可预测性, 却忽略了现实世界中时间序列的固有属性, 导致数据的过度平稳化问题, 模型预测准确率不足.

2.2 模型结构

结合因果卷积的非平稳学习倒置 Transformer 的模型结构如图 3 所示, 该模型基于 Transformer 的编码器架构实现, 仅利用 Transformer 编码器来专注于特征学习和多元序列的自适应关联. 从平稳性出发提出两个互补模块: 序列平稳化模块和结合因果卷积的非平稳学习注意力机制模块. 其中序列平稳化模块用于处理非平稳时间序列使其平稳化; 结合因果卷积的非平稳学习注意力机制模块则将平稳化导致消失的重要非平稳信息重新引入. 通过转置嵌入时间序列, 利用结合因果卷积的非平稳学习注意力机制来关注变量之间的依赖性, 并由前馈神经网络处理数据时间上的依赖性. 通过这些模块设计, 结合因果卷积的非平稳学习倒置 Transformer 提高了模型的预测准确性.

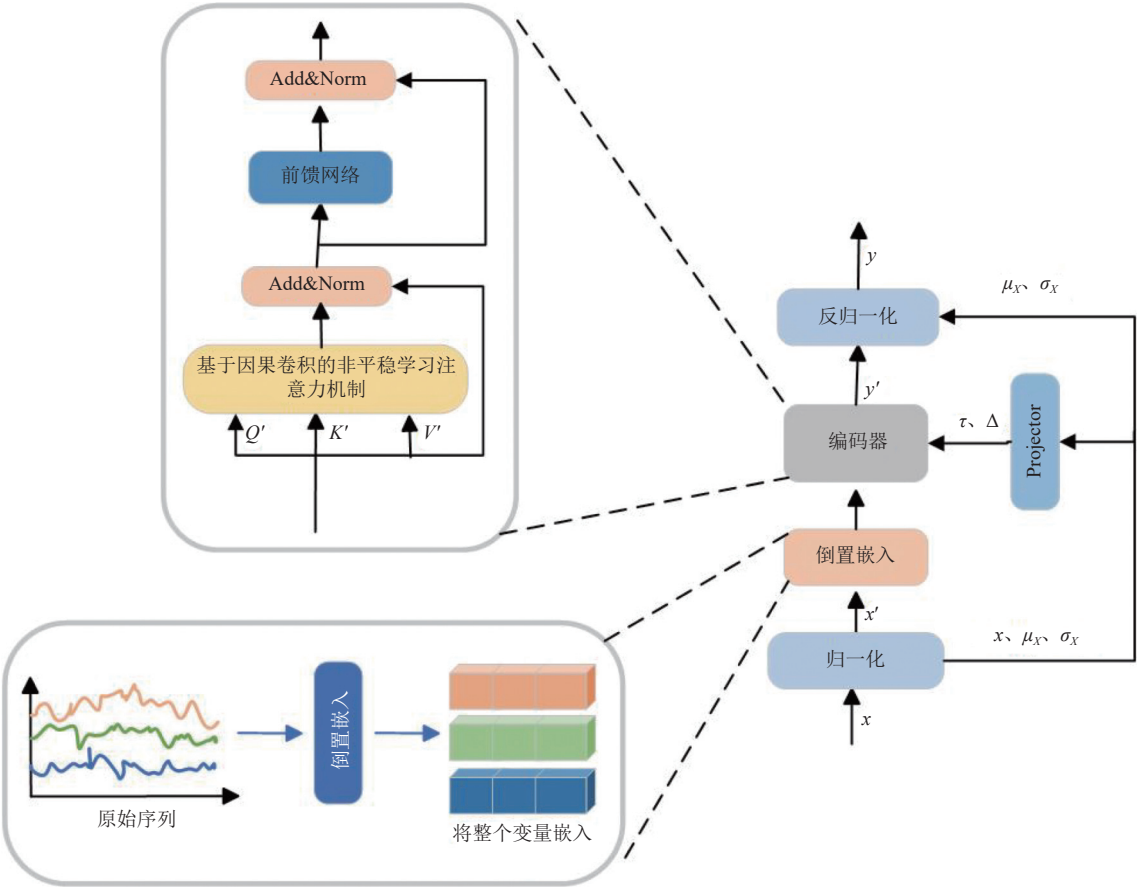

图 3 模型结构

2.2.1 倒置嵌入

倒置嵌入是将时间序列整个输入转置, 即将每个变量的序列整体作为嵌入 token, 删掉原本的位置嵌入. 这样的操作丰富了输入数据所携带的信息, 有利于全局建模, 适合多维时间序列预测任务.

基于以上考虑, 在结合因果卷积的非平稳学习倒

置 Transformer 模型中, 基于过去时间序列$X_{:,n}$预测每个特定变量$\hat{Y}_{:,n}$的未来序列的过程可以简单表述为式 (1)–式 (3):

$$h_n^0 = Embedding(X_{:,n}) \tag{1}$$

$$H^{l+1} = TrmBlock(H^l),\ l = 0,\cdots,L-1 \tag{2}$$

$$\hat{Y}_{:,n} = Projection(h_n^L) \tag{3}$$

其中, $H = \{h_1,\cdots,h_N\} \in \mathbb{R}^{N\times D}$包含$N$个维度为$D$的嵌入 token. 嵌入以及投影均由多层感知器实现. 通过倒置, 将每个变量的整个时间序列独立地嵌入到模型中, 嵌入的 token 聚合了序列的全局表示, 这些表示更加以变量为中心, 通过自注意力相互交互更好地学习变量之间的相关性, 再由前馈神经网络处理时间维度的依赖性. 如图 4 所示.

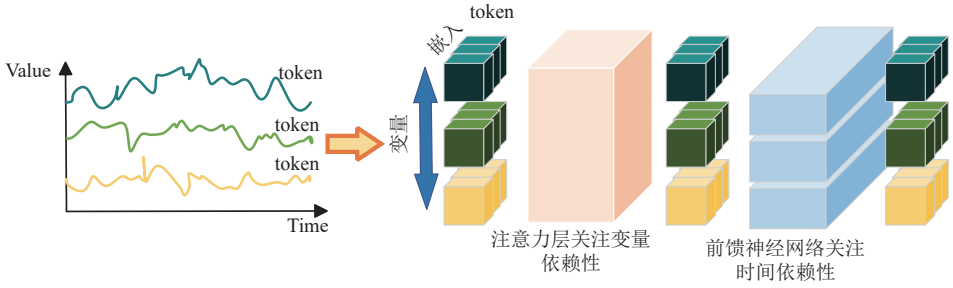

图 4 倒置嵌入模块

将序列转置嵌入, 使 Transformer 原有的结构功能发生改变. 形成 token 的是同一变量的整条序列. 这样嵌入的序列具备足够的模型容量来提取在历史观测数据和未来预测中共享的时间序列特征, 通过万能逼近定理[18]可知前馈神经网络可以近似任何函数, 利用线性网络更适合处理时间序列上的时间依赖性特点, 使用前馈神经网络提取复杂的特征表示来描述时间序列.

在 Transformer 中会将时间序列同一时间点的不同变量看作一个 token, 因此在 Transformer 中自注意力机制是用来促进时间之间的依赖. 而转置过后, 则是将整个序列看作 1 个 token, 将同一变量不同时间点上的数据视为一个整体, 自注意力机制通过全面提取每个时间序列$H = \{h_1,\cdots,h_N\} \in \mathbb{R}^{N\times D}$的表示, 采用线性投影来获取 query、key 和 value 的值. 因此自注意力机制可以促进不同变量之间的依赖性, 相关性高的变量之间的V的权重更大, 这一机制对多变量时间序列变量之间的关系进行建模, 提升了模型的可解释性.

2.2.2 序列平稳化

非平稳时间序列使深度学习模型的预测任务变得十分棘手, 因为深度学习模型很难学习统计数据变化的数据特征 (均值和标准差). RevIN[15]将具有可学习仿射参数的实例归一化应用于每个输入, 使得每个序列遵循相似的分布, 再将统计数据恢复到相应的输出. 首先利用归一化模块处理均值和标准差变化引起的非平稳序列, 最后利用反归一化模块将模型输出转换回原始数据相应的输出. 通过滑动窗口对时间维度进行归一化. 对于每个输入序列$X = [x_1, x_2,\cdots,x_T]^{\mathrm{T}} \in \mathbb{R}^{T\times N}$, 通过归一化得到$X' = [x'_1, x'_2,\cdots,x'_T]^{\mathrm{T}} \in \mathbb{R}^{T\times N}$, 其中$T$和$N$分别表示序列长度和变量个数. 归一化模块的数学表示如式 (4):

$$\begin{cases} \mu_X = \dfrac{1}{T}\displaystyle\sum_{i=1}^{T} x_i \\ \sigma_X^2 = \dfrac{1}{T}\displaystyle\sum_{i=1}^{T}(x_i-\mu_X)^2 \\ x'_i = \dfrac{1}{\sigma_X}\odot(x_i-\mu_X) \end{cases} \tag{4}$$

其中, $\mu_X, \sigma_X \in \mathbb{R}^{N\times 1}$, $1/\sigma_X$表示逐元素除法, $\odot$表示逐元素乘积. 因所有序列都标准化为正态分布, 所以减少了由于不一致的测量引起的差异, 使模型的输入数据分布更加稳定.

在模型H预测出长度为 L 的预测值后, 对模型输出的$y' = \left[y'_1, y'_2,\cdots,y'_L\right]^{\mathrm{T}} \in \mathbb{R}^{L\times N}$再采用反归一化, 得到$\hat{y} = [\hat{y}_1, \hat{y}_2,\cdots,\hat{y}_L]^{\mathrm{T}} \in \mathbb{R}^{L\times N}$作为最终的预测结果. 反归一化模块的数学表示如式 (5):

$$y' = H(x'),\ \hat{y}_i = \sigma_X \odot \left(y'_i + \mu_X\right) \tag{5}$$

通过这两个模块, 模型将接收到平稳的输入, 且更容易泛化, 从而有利于现实世界中的序列预测.

2.3 结合因果卷积的非平稳学习注意力

通过反归一化模块可以将每个经过归一化模块的

时间序列都显式地恢复到相应的预测结果, 但仅通过反归一化无法完全恢复原始序列的非平稳性. 假设两个不同的时间序列x_1, x_2 ($x_2 = \alpha x_1 + \beta$), 这两个时间序列具有相同的均值与方差, 通过归一化模块后会生成相同的平稳输入x', 这就会导致这两个序列将获得相同的关注权重, 但原来的序列由于存在非平稳信息可能获得的关注权重并不同. 非平稳时间序列经过平稳化所得的序列与原始数据相比遵循更相似的分布. 这使得模型有可能产生过度平稳且无规律的输出, 这种破坏效应发生在深层模型内部, 尤其是在注意力的计算中. 为防止序列平稳化引起的过度平稳化问题, 提出引入非平稳信息的注意力机制, 通过它可以近似原始序列获得的注意力.

过度平稳化问题是固有的非平稳性信息消失而引起的, 这使得模型无法捕获序列的正确的关注权重. 从Self-attention 的公式来分析:

$$Attn(Q,K,V) = Softmax\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{6}$$

其中, Q、K、$V \in \mathbb{R}^{N\times d_k}$ 分别是维度为d_k长度为N的query、key 和 value. 过去的方法是对每个时间序列变量进行归一化, 以避免某个变量占主导地位, 可以假设序列x的每个变量具有相同的方差, 时间序列经过归一化模块后, 模型将会接收到平稳输入$x' = (x - \mathbf{1}\mu_x^{\mathrm{T}})/\sigma_X$, 其中$\mathbf{1} \in \mathbb{R}^{N\times 1}$. 如果没有序列平稳化, 自注意力中$Softmax(\cdot)$的输入应该是$QK^{\mathrm{T}}/\sqrt{d_k}$, 经过序列平稳化后注意力层将得到$Q' = \left[f\left(X_1'\right),\cdots,f\left(X_N'\right)\right] = (Q - \mathbf{1}\mu)/\sigma$与$K'$, 故此时$Softmax(\cdot)$的输入为$Q'K'^{\mathrm{T}}/\sqrt{d_k}$.

$$Q'K'^{\mathrm{T}} = \frac{1}{\sigma_X^2}\left(QK^{\mathrm{T}} - \mathbf{1}\left(\mu_Q^{\mathrm{T}}K^{\mathrm{T}}\right) - (Q\mu_K)\mathbf{1}^{\mathrm{T}} + \mathbf{1}\left(\mu_Q^{\mathrm{T}}\mu_K\right)\mathbf{1}^{\mathrm{T}}\right) \tag{7}$$

$$Softmax\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right) = Softmax\left(\begin{array}{l}\dfrac{\sigma_X^2 Q'K'^{\mathrm{T}} + \mathbf{1}\left(\mu_Q^{\mathrm{T}}K^{\mathrm{T}}\right)}{\sqrt{d_k}} + \\ \dfrac{(Q\mu_K)\mathbf{1}^{\mathrm{T}} - \mathbf{1}\left(\mu_Q^{\mathrm{T}}\mu_K\right)\mathbf{1}^{\mathrm{T}}}{\sqrt{d_k}}\end{array}\right) \tag{8}$$

根据$Softmax$算子的平移不变性, 上述公式可以简化为式 (9):

$$Softmax\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right) = Softmax\left(\frac{\sigma_X^2 Q'K'^{\mathrm{T}} + \mathbf{1}\left(\mu_Q^{\mathrm{T}}K^{\mathrm{T}}\right)}{\sqrt{d_k}}\right) \tag{9}$$

序列经过平稳化可能会丢失序列上的部分重要信息, 在经过注意力模块时所得到的注意力权重会受到影响. 为了恢复对非平稳序列的原始注意力, 尝试将消失的非平稳信息带回到其中计算. 根据式 (9), 假设正标量$\tau = \sigma_X^2$、平移向量$\Delta = K\mu_O$为非平稳因子, 但是严格的线性特性很难适用于深度模型, 为了获得真实非平稳因子, 完成对式 (9) 的深度学习实现, 我们应用多层感知器作为投影仪, 从非平稳化x的统计数据μ_X、σ_X中分别学习非平稳因子τ、Δ, 以恢复原始的注意力权重. 非平稳学习注意力机制公式如式 (10)、式 (11):

$$\log\tau = MLP(\sigma_X, x),\ \Delta = MLP(\mu_X, x) \tag{10}$$

$$Attn(Q',K',V',\tau,\Delta) = Softmax\left(\frac{\tau Q'K'^{\mathrm{T}} + \mathbf{1}\Delta}{\sqrt{d_k}}\right)V' \tag{11}$$

其中, 非平稳因子τ和Δ由所有层的非平稳学习注意力共享. 非平稳学习注意力机制从平稳序列Q', K'和非平稳序列x、μ_X、σ_X中学习依赖性, 因此, 它既可以保持平稳序列的可预测性, 又可以学习到原始序列间固有的依赖性.

因果卷积的核心是因果性约束, 因果卷积只允许当前时间窗口之前的信息流入到当前时间窗口, 不允许未来的信息影响当前的预测. 这种约束保证了模型对时间序列数据具有因果关系的建模能力, 同时也有助于模型减少过拟合风险. 将因果卷积与非平稳学习注意力相结合, 利用因果卷积来提取局部特征, 在这个过程中只考虑当前时间步之前的信息, 确保只有过去的信息流入当前时间步, 引入因果关系, 减少模型对序列中不必要信息的关注, 帮助非平稳学习注意力机制提取更真实的特征来更好的学习时间序列变量之间的依赖性, 这样的结合能够提高模型的可解释性, 使模型更加稳定, 增强模型的泛化能力.

3 实验

3.1 数据集

本文采用多个公开数据集, 在 ETT 数据集中的 ETTm1 数据集和 ETTm2 数据集、Weather 数据集和 Exchange 数据集上进行实验.

ETT 数据集中包含 2 个电力变压器站点的数据, ETTm1 与 ETTm2 数据集分别来自我国同一个省的两个不同地区的站点, 按每 15 min 的采样频率记录了从 2016–2018 年共 69 680 条数据记录.

Weather 数据集是由马克斯普朗克生物地球化学

研究所气象站记录的 2020 年气象数据, 包括温度、气压、湿度、风向等共 21 个天气指标, 本文选取了其中按 10 min 采样频率所采集的 52 696 条数据进行实验.

Exchange 数据集记录了 1990–2016 年 8 个国家的每日汇率数据. 本文选取了 1990–2010 年中采集的 7588 条数据进行实验.

3.2 实验设置

本文选取 iTransformer[12]、Crossformer[11]等 10 个模型作为对比实验基准模型. 其中 iTransformer 模型更改模型内模块的作用, 有效提高了模型的预测准确性. Crossformer 模型利用两阶段注意力机制同时关注时间维度与变量之间的依赖性, 提高了模型预测效果并增加了模型的可解释性.

本文模型及对比模型均使用 PyTorch 框架实现, 使用 AMD Ryzen 7 5800H with Radeon Graphics 核处理器, NVIDIA 4090 Ti GPU 用于加速网络模型的训练和测试. 详细数据集描述如表 1 所示, 其中数据集设置大小中的 3 个数据分别表示训练集, 验证集, 测试集. 为了保证真实场景中信息不泄露且加速模型收敛, 使用训练集的均值和标准差进行归一化操作. 所有模型训练均使用 L2 loss 作为损失函数训练, Adam 优化器进行参数优化, 初始学习率设置为{0.001, 0.0005, 0.0001}, 批量大小统一设置为 32, 训练周期数固定为 10.

3.3 评价指标

所有实验使用均方误差和平均绝对误差作为评价指标来衡量预测值与真实值之间的差异, 数值越小代表预测效果越好. 评价指标表达式如式 (12)、式 (13):

$$MSE = \frac{1}{M}\sum_{i=1}^{M}(y_i - \hat{y}_i)^2 \tag{12}$$

$$MAE = \frac{1}{M}\sum_{i=1}^{M}|y_i - \hat{y}_i| \tag{13}$$

其中, M为预测序列长度, y_i和$\hat{y}_i$分别对应第i时刻的真实值和预测值.

表 1 详细数据集描述

数据集	变量数量	预测长度	数据集设置大小	数据集信息
Exchange	8	{96, 192, 336, 720}	(5 120, 665, 1 422)	国家每日汇率
Weather	21	{96, 192, 336, 720}	(36 696, 5 175, 10 444)	天气
ETTm1	7	{96, 192, 336, 720}	(34 369, 11 425, 11 425)	电力
ETTm2	7	{96, 192, 336, 720}	(34 369, 11 425, 11 425)	电力

3.4 实验设计与结果分析

本文针对不同数据集固定输入长度, 其大小至少包含一个周期项, 分别进行不同输出长度的性能测试. 计算验证集中所有批次的 *MSE* 取平均值选取最优模型, 并最终在测试集上进行测试, 具体输入长度, 预测长度以及实验结果如表 2 所示.

表 2 本文模型与其他模型在不同数据集下的预测结果

模型	数据集	Exchange		Weather		ETTm1		ETTm2		4个数据集指标平均对比结果 (%)	
	预测长度	*MSE*	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*
本文模型	96	0.089	0.218	0.171	0.214	0.330	0.372	0.171	0.257	+0	+0
	192	0.166	0.301	0.222	0.259	0.375	0.393	0.233	0.302	+0	+0
	336	0.274	0.386	0.276	0.298	0.407	0.415	0.295	0.334	+0	+0
	720	0.503	0.544	0.344	0.346	0.468	0.451	0.403	0.397	+0	+0
	Avg	**0.258**	**0.362**	**0.253**	**0.279**	**0.395**	**0.408**	**0.275**	**0.322**	+0	+0
iTransformer[12]	96	0.098	0.226	0.174	0.218	0.347	0.378	0.182	0.265	+5.5	+2.5
	192	0.188	0.321	0.224	0.261	0.381	0.397	0.254	0.313	+5.6	+2.9
	336	0.329	0.417	0.278	0.302	0.419	0.425	0.317	0.352	+6.8	+4.1
	720	0.869	0.704	0.358	0.349	0.486	0.462	0.425	0.417	+13.7	+7.7
	Avg	0.371	0.417	0.258	0.282	0.408	0.416	0.295	0.337	+7.9	+4.3
PatchTST[4]	96	0.089	0.221	0.177	0.218	0.331	0.376	0.177	0.260	+1.8	+1.4
	192	0.179	0.301	0.225	0.259	0.378	0.396	0.244	0.303	+3.5	+0.3
	336	0.342	0.422	0.280	0.298	0.412	0.423	0.304	0.342	+6.4	+3.2
	720	0.922	0.717	0.357	0.349	0.469	0.452	0.416	0.406	+13.1	+6.9
	Avg	0.383	0.415	0.260	0.281	0.398	0.412	0.285	0.328	+6.2	+2.9
Crossformer[11]	96	0.234	0.356	0.164	0.232	0.358	0.398	0.287	0.382	+26.5	+21.4
	192	0.439	0.505	0.227	0.301	0.505	0.500	0.316	0.398	+29.1	+24.9
	336	0.685	0.807	0.270	0.329	0.546	0.531	0.433	0.458	+28.8	+27.6
	720	0.826	0.828	0.366	0.398	0.626	0.633	0.600	0.527	+25.8	+25.2
	Avg	0.546	0.624	0.257	0.315	0.509	0.516	0.409	0.441	+27.5	+24.8

表 2 本文模型与其他模型在不同数据集下的预测结果 (续)

模型	数据集	Exchange		Weather		ETTm1		ETTm2		4个数据集指标平均对比结果 (%)	
	预测长度	*MSE*	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*
DLinear[5]	96	0.081	0.203	0.196	0.255	0.344	0.371	0.185	0.281	+3.6	+4.2
	192	0.157	0.293	0.237	0.296	0.381	0.395	0.279	0.356	+4.7	+6.4
	336	0.305	0.414	0.283	0.335	0.413	0.414	0.357	0.409	+7.9	+8.9
	720	0.643	0.601	0.345	0.381	0.471	0.457	0.536	0.510	+11.9	+10.5
	Avg	0.297	0.378	0.265	0.317	0.402	0.409	0.339	0.389	+7.0	+7.5
TimesNet[19]	96	0.108	0.235	0.172	0.220	0.347	0.383	0.182	0.267	+7.3	+4.1
	192	0.199	0.327	0.219	0.261	0.382	0.400	0.254	0.308	+6.3	+3.1
	336	0.376	0.449	0.280	0.306	0.412	0.419	0.321	0.349	+9.5	+5.5
	720	0.954	0.744	0.365	0.359	0.485	0.457	0.422	0.405	+15.2	+8.4
	Avg	0.409	0.439	0.259	0.287	0.407	0.415	0.295	0.332	+9.6	+5.3
FEDformer[20]	96	0.148	0.278	0.217	0.296	0.377	0.415	0.203	0.287	+22.3	+17.5
	192	0.271	0.380	0.276	0.336	0.436	0.499	0.269	0.328	+21.4	+18.2
	336	0.460	0.500	0.339	0.380	0.469	0.471	0.325	0.366	+20.4	+16.3
	720	1.195	0.841	0.403	0.428	0.501	0.486	0.421	0.415	+20.9	+16.5
	Avg	0.519	0.499	0.309	0.360	0.446	0.468	0.305	0.349	+21.2	+17.1
Nonstationary[10]	96	0.122	0.249	0.186	0.233	0.433	0.421	0.296	0.333	+25.3	+13.8
	192	0.240	0.353	0.253	0.293	0.536	0.463	0.442	0.391	+30.1	+16.1
	336	0.462	0.506	0.339	0.346	0.532	0.476	0.478	0.419	+30.2	+17.7
	720	1.283	0.835	0.382	0.374	0.628	0.525	0.579	0.481	+31.7	+18.5
	Avg	0.527	0.486	0.290	0.312	0.532	0.471	0.449	0.406	+29.3	+16.5
Autoformer[8]	96	0.197	0.323	0.266	0.336	0.510	0.492	0.255	0.339	+39.7	+29.3
	192	0.300	0.369	0.307	0.367	0.514	0.495	0.281	0.340	+29.1	+19.9
	336	0.509	0.524	0.359	0.395	0.521	0.497	0.339	0.372	+26.0	+19.4
	720	1.447	0.941	0.419	0.428	0.527	0.493	0.422	0.419	+24.7	+18.8
	Avg	0.613	0.539	0.338	0.382	0.518	0.494	0.324	0.368	+29.9	+21.9
Pyraformer[21]	96	0.852	0.780	0.354	0.392	0.543	0.510	0.409	0.488	+59.7	+47.9
	192	0.993	0.858	0.673	0.597	0.557	0.537	0.673	0.461	+62.1	+45.7
	336	1.240	0.958	0.634	0.592	0.754	0.655	1.210	0.846	+64.0	+51.6
	720	1.711	1.093	0.942	0.723	0.908	0.724	4.044	1.526	+68.1	+53.5
	Avg	1.119	0.922	0.651	0.576	0.691	0.607	1.584	0.831	+63.5	+49.7
Informer[9]	96	0.847	0.752	0.300	0.384	0.626	0.560	0.365	0.453	+58.2	+48.0
	192	1.204	0.895	0.598	0.544	0.725	0.619	0.533	0.563	+63.4	+50.4
	336	1.672	1.036	0.592	0.578	1.005	0.741	1.363	0.887	+68.7	+54.4
	720	2.478	1.310	0.723	1.059	1.133	0.845	3.379	1.388	+69.7	+60.9
	Avg	1.551	0.998	0.553	0.641	0.872	0.691	1.410	0.823	+65.0	+53.4

实验结果如图 5 所示, 横坐标表示数据集预测长度, 纵坐标表示 *MSE* 数值. 实验结果表明在 Exchange、Weather、ETTm1、ETTm2 这 4 个数据集中本文模型的 *MSE* 和 *MAE* 评价指标均为最优. 根据表 2, 在 Exchange 数据集上, 本文模型与最优基准模型 DLinear 相比, 本文模型的 *MSE* 值平均降低了 13.1%, 与其他 9 个模型相比, 本文模型的 *MSE* 值平均降低了 30.5%–83.4%, 且本文模型在长期预测性能上明显优于其他模型. 在 Weather 数据集上, 本文模型与 CrossFormer 在预测性能上表现相当, iTransformer、PatchTST、TimesNet、DLinear 模性表现次之, 虽然本文模型与上述 4 个模型在短期预测性能相当, 但在长序列预测中展现了自己的优势. 对比其他模型, 本文模型的 *MSE* 值平均降低了 13.4%–61.1%. 在 ETTm1 数据集上, 本文模型与最优基准模型 PatchTST 在长期预测上性能表现相当, 其 *MSE* 值降低了 0.3%–1.3%. iTransformer 模型、TimesNet 模型与 DLinear 模型表现次之. CrossFormer 模型虽在短期预测上表现较好, 但对于长期趋势的捕捉能力较差导致其在长期预测上的性能下降较多. 与其他 5 个模型对比, 本文模型的 *MSE* 值平均降低了 23.7%–54.7%, 在 ETTm2 数据集上, 本文模型与最优基准模型 PatchTST 相比, 其 *MSE* 值和 *MAE* 值平均降低了 3.5% 和 1.8%. 但本文模型在预测长度为 720 时 *MSE* 值为 0.403, 而 PatchTST 模型的 *MSE* 值为 0.416, 因此本文模型在长序列预测性能上更优秀. 与其他 9 个模型对比, 本文模型的 *MSE* 值平均降低了 6.7%–82.6%. 总体

而言, 与同样关注时间与变量建模的 Crossformer 相比, 本文模型的建模方式在长期预测上明显更好, 这表明来自不同多元变量未对齐的时间 token 的交互会给预测带来不必要的噪声且前馈神经网络更适合在时间维度建模. 与同样更换模型内模块功能的 iTransformer 相比, 本文模型中使用了结合因果卷积的非平稳学习注意力模块能够更关注原始序列的特征学习, 使得总体性能更加优秀.

3.5 消融实验

为验证本文提出的结合因果卷积的非平稳学习倒置 Transformer 模型中倒置嵌入模块和结合因果卷积的非平稳学习注意力模块的有效性, 分别设置了 3 个对比模型, 消融实验的结果如表 3.

Model-1: 仅将时间序列倒置嵌入模块更改为正常嵌入的方式, 恢复位置编码.

Model-2: 仅将结合因果卷积的非平稳学习注意力模块替换为普通的自注意力模块.

Model-3: 仅将结合因果卷积的非平稳学习注意力模块替换为非平稳学习注意力模块

在 Exchange 数据集上 3 个对比模型的 *MSE* 值分别上升了 36.2%、30.4% 和 14.8%; 在 Weather 数据集上分别上升了 9.9%、1.9% 和 2.6%; 在 ETTm1 数据集上分别上升了 20.8%、3.1% 和 8.6%; 在 ETTm2 数据集上分别上升了 22.9%、6.5% 和 13.5%. 结果表明本文所引入的模块能有效地提升时间序列的长期预测的准确性. 图 6 为本文模型在 4 个数据集上与消融实验模型在 *MSE* 指标上的对比结果, 进一步表明本文所提出的采用倒置嵌入时间序列、将因果卷积与注意力机制相结合, 同时将非平稳信息引入注意力机制的方法, 能够有效提升模型准确度.

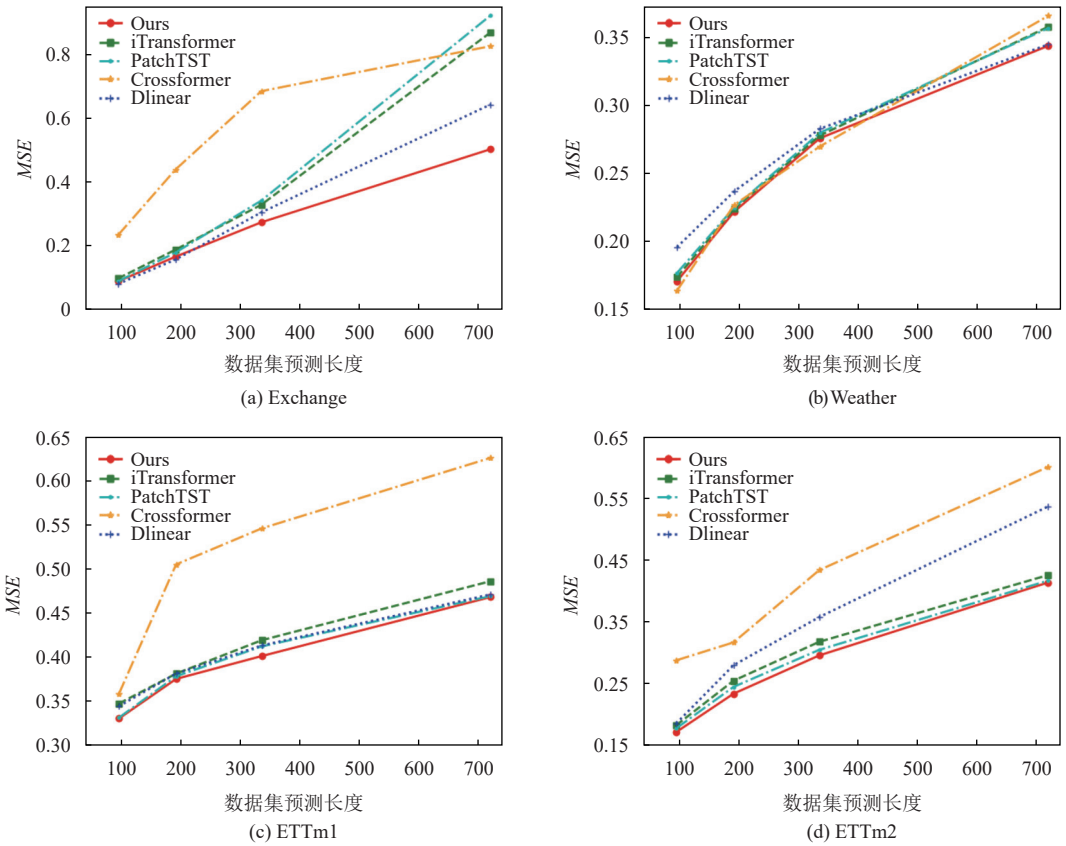

图 5 5 个模型在不同数据集上的测试结果

4 结论与展望

本文中针对时间序列的非平稳性采取了平稳化方式, 为深度模型提供更稳定的数据分布, 提高了数据的可预测性. 针对模型预测过程中平稳化导致的重要信息缺失问题, 采用结合因果卷积的非平稳学习注意力机制, 学习原始非平稳序列的重要信息, 获得更为准确的注意力权重. 针对模型的正确职责与时间维度与变量维度建模问题, 采用倒置的方式嵌入时间序列, 利用

注意力机制学习变量之间的依赖性, 利用前馈神经网络学习时间序列时间维度上的依赖性, 使模型对时间维度以及变量维度更好的建模, 提高了模型的预测准确性. 多维时间序列的变量之间存在一定的依赖性以及因果关系, 因此协变量会影响预测变量的结果. 本文所提模型虽然探索了一定的依赖以及因果关系, 但其学习的因果关系并不是真正的因果关系. 为了更好地学习协变量与预测变量的依赖性以及因果性, 需要设计更有效的网络来挖掘其中的关系, 后续将针对该问题继续展开研究.

表 3 消融实验结果

数据集	模型	本文模型		Model-1		Model-2		Model-3	
	预测长度	*MSE*	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*
Exchange	96	0.089	0.218	0.117	0.234	0.096	0.224	0.112	0.231
	192	0.166	0.301	0.231	0.348	0.188	0.321	0.191	0.314
	336	0.274	0.386	0.452	0.484	0.329	0.417	0.285	0.396
	720	0.503	0.544	0.821	0.672	0.869	0.704	0.627	0.607
	Avg	**0.258**	**0.362**	0.405	0.434	0.371	0.416	0.303	0.387
Weather	96	0.171	0.214	0.183	0.229	0.171	0.218	0.176	0.216
	192	0.222	0.259	0.248	0.286	0.224	0.261	0.226	0.261
	336	0.276	0.298	0.334	0.346	0.288	0.305	0.279	0.280
	720	0.344	0.346	0.362	0.361	0.351	0.252	0.361	0.350
	Avg	**0.253**	**0.279**	0.281	0.305	0.258	0.259	0.260	0.276
ETTm1	96	0.330	0.372	0.422	0.418	0.347	0.378	0.341	0.374
	192	0.375	0.393	0.429	0.424	0.381	0.397	0.398	0.404
	336	0.407	0.415	0.525	0.471	0.419	0.425	0.446	0.432
	720	0.468	0.451	0.618	0.519	0.486	0.462	0.546	0.504
	Avg	**0.395**	**0.407**	0.499	0.458	0.408	0.415	0.432	0.428
ETTm2	96	0.171	0.257	0.210	0.288	0.182	0.265	0.209	0.297
	192	0.233	0.302	0.306	0.338	0.254	0.313	0.287	0.351
	336	0.295	0.334	0.395	0.389	0.317	0.352	0.303	0.343
	720	0.403	0.397	0.517	0.544	0.425	0.417	0.476	0.433
	Avg	**0.275**	**0.322**	0.357	0.389	0.294	0.336	0.318	0.356

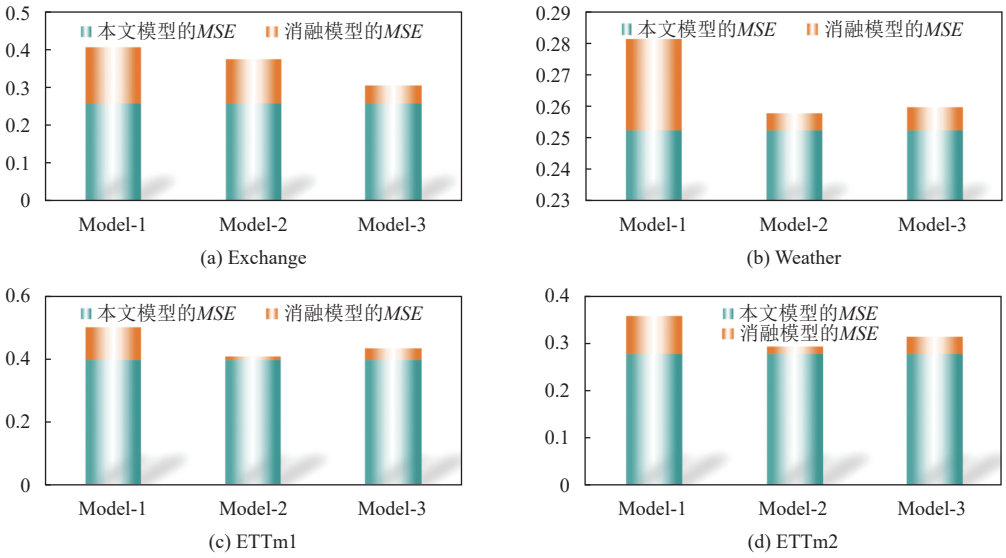

图 6 消融实验结果对比

参考文献

1 Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010.

2 Brown TB, Mann B, Ryder N, *et al*. Language models are

few-shot learners. Proceedings of the 34th International Conference on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 159.

3 Dosovitskiy A, Beyer L, Kolesnikov A, *et al*. An image is worth 16x16 words: Transformers for image recognition at scale. Proceedings of the 9th International Conference on Learning Representations. OpenReview.net, 2021 .

4 Nie YQ, Nguyen NH, Sinthong P, *et al*. A time series is worth 64 words: Long-term forecasting with Transformers. Proceedings of the 11th International Conference on Learning Representations. OpenReview.net, 2023.

5 Li Z, Qi SY, Li YD, *et al*. Revisiting long-term time series forecasting: An investigation on linear mapping. arXiv:2305.10721, 2023.

6 Maddix DC, Wang YY, Smola A. Deep factors with Gaussian processes for forecasting. arXiv:1812.00098, 2018.

7 Rangapuram SS, Seeger M, Gasthaus J, *et al*. Deep state space models for time series forecasting. Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montréal: Curran Associates Inc., 2018. 7796–7805.

8 Wu HX, Xu JH, Wang JM, *et al*. Autoformer: Decomposition Transformers with auto-correlation for long-term series forecasting. Proceedings of the 35th International Conference on Neural Information Processing Systems. Curran Associates Inc., 2021. 1717.

9 Zhou HY, Zhang SH, Peng JQ, *et al*. Informer: Beyond efficient Transformer for long sequence time-series forecasting. Proceedings of the 35th Conference on Artificial Intelligence. AAAI Press, 2021. 11106–11115.

10 Liu Y, Wu HX, Wang JM, *et al*. Non-stationary Transformers: Exploring the stationarity in time series forecasting. Proceedings of the 36th International Conference on Neural Information Processing Systems. New Orleans, 2022. 9881–9893.

11 Zhang YH, Yan JC. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. Proceedings of the 11th International Conference on Learning Representations. OpenReview.net, 2023.

12 Liu Y, Hu TG, Zhang HR, *et al*. iTransformer: Inverted Transformers are effective for time series forecasting. Proceedings of the 12th International Conference on Learning Representations. OpenReview.net, 2024.

13 Anderson OD, Kendall M. Time-series. 2nd ed., The Statistician, 1976, 25(4): 308–310.

14 Hyndman RJ, Athanasopoulos G. Forecasting: Principles and Practice. 2nd ed., Melbourne: OTexts, 2018.

15 Kim T, Kim J, Tae Y, *et al*. Reversible instance normalization for accurate time-series forecasting against distribution shift. Proceedings of the 10th International Conference on Learning Representations. OpenReview.net, 2022.

16 Ogasawara E, Martinez LC, De Oliveira D, *et al*. Adaptive normalization: A novel data normalization approach for non-stationary time series. Proceedings of the 2010 International Joint Conference on Neural Networks. Barcelona: IEEE, 2010. 1–8.

17 Passalis N, Tefas A, Kanniainen J, *et al*. Deep adaptive input normalization for time series forecasting. IEEE Transactions on Neural Networks and Learning Systems, 2020, 31(9): 3760–3765. [doi: 10.1109/TNNLS.2019.2944933]

18 Hornik K. Approximation capabilities of multilayer feedforward networks. Neural Networks, 1991, 4(2): 251–257. [doi: 10.1016/0893-6080(91)90009-T]

19 Wu HX, Hu TG, Liu Y, *et al*. TimesNet: Temporal 2D-variation modeling for general time series analysis. Proceedings of the 11th International Conference on Learning Representations. OpenReview.net, 2023.

20 Zhou T, Ma ZQ, Wen QS, *et al*. FEDformer: Frequency enhanced decomposed Transformer for long-term series forecasting. Proceedings of the 39th International Conference on Machine Learning. Baltimore: PMLR, 2022. 27268–27286.

21 Liu SZ, Yu H, Liao C, *et al*. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. Proceedings of the 10th International Conference on Learning Representations. OpenReview.net, 2022.

(校对责编: 王欣欣)