时间序列预测研究全景：方法、挑战与未来方向

报告人 穆莹

报告日期 2025年12月7日

目录

第 1 部分

基础概念与定义

1基础概念与定义

什么是时序序列?

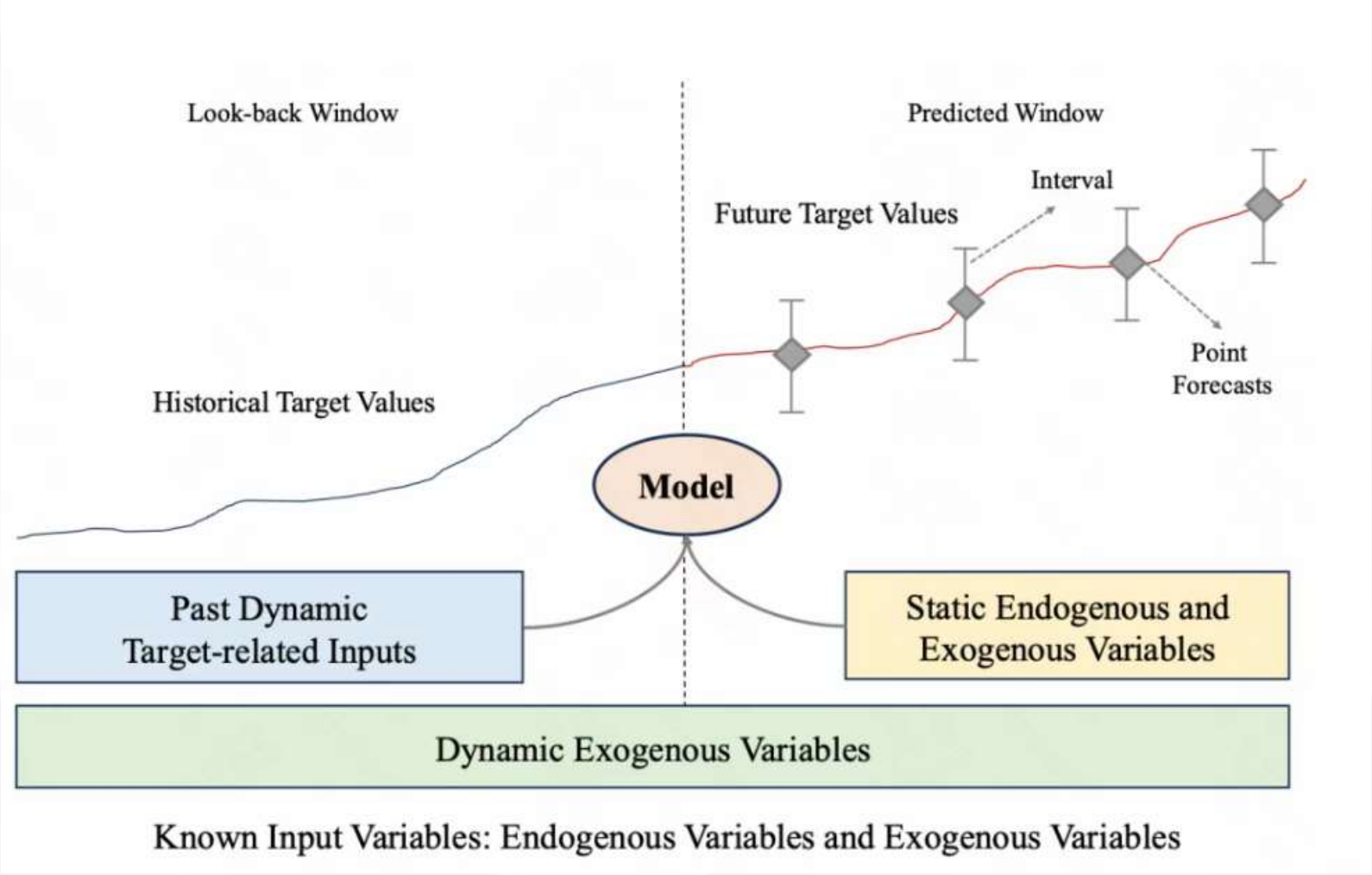
时间序列通常被定义为按时间顺序索引的数据点序列，其中每个观察值是在特定（通常是均匀的）时间间隔内获得的。尽管数据通常在离散的间隔内记录（例如，每小时、每天、每月），但这些观察值所基于的许多实际现象可以被视为连续的，并且原则上在时间和价值上都是无界的。因此，时间序列建模必须同时处理测量的离散性质和过程本身的潜在连续性，并深入建模数据的趋势性、周期性和不规则动态变化。

核心任务：基于历史观测值和可能的协变量，预测目标序列的未来值。
关键组件：

回顾窗口：用作模型输入的历史数据。
预测窗口：模型需要输出的未来数据。
协变量：影响目标序列的外部变量。
任务分类：

单步 vs. 多步预测 单步预测每次仅预测一个未来时间点 多步预测则一次性输出多个未来时间点
单变量 vs. 多变量预测 单变量预测仅依赖目标序列本身；多变量预测则引入多个相关序列或协变量，
迭代 vs. 直接预测策略 迭代策略逐步生成未来值，并将其作为输入用于后续预测；直接策略则为每个未来时间步构建独立的预测机制
点预测 vs. 概率预测 预测结果的性质，时序预测任务可以划分为点预测（非概率）和概率预测，前者为每个时间步生成一个确定性的预测，而后者挖掘数据中的固有不确定性以生成对未来值分布的预测。

1基础概念与定义


Look-back Window
Predicted Window
Interval
Future Target Values
Point
Forecasts
Historical Target Values
Model
Past Dynamic
Target-related Inputs
Static Endogenous and
Exogenous Variables
Dynamic Exogenous Variables
Known Input Variables: Endogenous Variables and Exogenous Variables

第 2 部分

十大核心挑战

2十大核心挑战

1.数据噪声与异常干扰
2.不规则采样与缺失值
3.长期依赖建模困难
4.多变量间复杂相关性建模
5.外生变量建模机制不足
6.数据分布漂移影响泛化
7.复杂趋势与季节性模式建模
8.多尺度结构融合困难
9.高计算开销限制落地
10.跨场景泛化与迁移能力薄弱

2十大核心挑战

2.1 数据噪声与异常干扰

时间序列数据常受到传感器误差、人为干预或自然异常事件影响，产生噪声与异常点，严重干扰模型学习。尽管已有大量填补与检测算法，但主流研究多基于理想化干净数据，真实应用场景中仍需提升模型的容错与纠错能力。

2.2 不规则采样与缺失值问题

现实采集场景中由于传输丢包、设备故障或经济成本约束，时间序列往往呈现不规则采样或关键值缺失。如何重构这些数据片段，并充分利用时间间隔信息，成为处理异质时间序列的关键技术问题。

2.3 长期依赖建模困难

长序列预测面临信息稀释、误差积累与依赖衰减等挑战，常规模型难以稳定捕捉远距离信息。需要设计具备更强记忆能力或结构重用机制的模型结构，以突破长期预测的性能瓶颈。

2.4 多变量相关性建模复杂

多变量序列存在潜在的非线性、动态因果关系，变量间的交互往往缺乏明确先验信息。如何在数据驱动条件下学习跨通道依赖关系，是提高预测准确性的核心路径。

2.5 外生变量建模机制不足

预测结果常受外部因素影响，例如节假日、天气、政策等，这些变量具有异步性与非线性作用。目前仍缺乏统一框架来识别关键外因、动态建模其作用，并与目标序列联动优化。

2十大核心挑战

2.6 数据分布漂移阻碍泛化能力

时间序列往往非平稳，不同时间段的统计特性可能发生剧烈变化。分布漂移不仅削弱模型的泛化能力，还可能导致训练数据与预测目标严重不一致，是构建健壮模型必须应对的问题。

2.7 趋势与季节性模式建模复杂

多周期叠加、突变结构、非线性趋势使得传统的趋势/季节性建模方法难以准确表达结构特性。需要更具表达能力的分解方法与数据增强策略辅助模型捕捉多级周期模式。

2.8 多尺度结构融合困难

时间序列存在局部扰动与全局趋势的多层次结构，不同粒度之间的平衡与融合直接影响预测结果。如何在建模过程中有效整合多尺度特征，是提升模型泛化与准确性的关键手段。

2.9 高计算开销限制落地效率

任务维度提升与序列延长导致预测模型计算复杂度剧增，严重影响部署与实时推理效率。在保持预测准确性的同时压缩模型结构，是工业应用中尤为关键的优化目标。

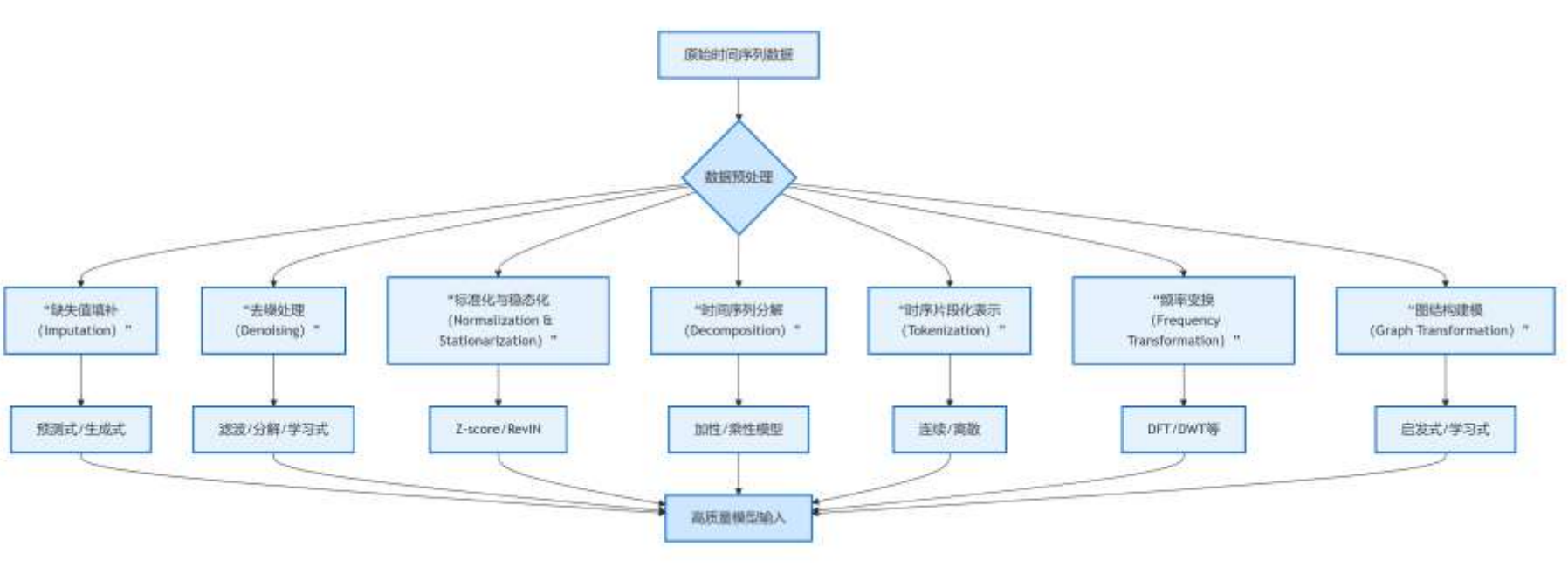
2.10 跨场景泛化与迁移能力薄弱

时间序列缺乏通用语义单元，不同任务之间变量含义、结构特性差异巨大，限制了模型的迁移与重用能力。构建可跨任务泛化的基础模型仍是当前研究的重大挑战之一。

第 3 部分

数据预处理

3数据预处理


原始时间序列数据
数据预处理
"缺失值填补
(Imputation)"
"去噪处理
(Denoising)"
"标准化与稳态化
(Normalization &
Stationarization)"
"时间序列分解
(Decomposition)"
"时序片段化表示
(Tokenization)"
"频率变换
(Frequency
Transformation)"
"图结构建模
(Graph Transformation)"
预测式/生成式
滤波/分解/学习式
Z-score/RevIN
加性/乘性模型
连续/离散
DFT/DWT等
启发式/学习式
高质量模型输入

3数据预处理

3.1 缺失值填补

问题根源：传感器故障、通信中断等导致数据出现局部或结构性缺失。
严重性：缺失值不仅导致信息丢失，更会破坏时序连续性，严重误导模型。
方法分类：
预测式方法：将填补视为一个序列预测问题，使用如RNN、Transformer等模型直接**预测缺失片段的值**。
生成式方法：引入生成机制（如GAN、扩散模型），学习完整数据的分布，从而生成可能的多条缺失片段。这种方法能更好地量化不确定性。

3.2 去噪处理

问题根源：设备误差、环境扰动引入高频噪声。
目的：让模型专注于长期趋势和关键变化点，提高预测的稳定性和泛化能力。
方法分类：
经典滤波方法：如移动平均、高低通滤波，简单有效，适用于平滑局部震荡。
分解型方法：如小波变换、经验模态分解，将序列拆解为不同尺度的信号成分，从而分离出噪声。
学习型方法：训练神经网络（如自编码器、扩散模型），学习从有噪数据到干净数据的映射函数，能力更强。

3.3 标准化与稳态化

问题根源：不同变量间存在量纲和尺度差异；序列本身存在非平稳性（统计特性随时间变化）。
目的：提升模型训练的数值稳定性，加速收敛，并缓解分布漂移问题。
方法分类：
标准化：如Z-score（归一化为均值为0，标准差为1）、Min-Max（缩放到[0,1]区间），将变量统一到可比尺度。
样本归一化：如RevIN，对每个样本独立进行归一化，并在模型输出后反向操作，能有效减少样本间的非平稳性影响。
动态归一化方法：如DAIN、SAN，引入可学习参数，根据输入序列的上下文自适应地进行归一化。

3.4 时间序列分解

核心思想：将序列视为由趋势（Trend）、季节性（Seasonality）和残差（Residual）等成分组合而成。分别建模这些结构性成分，可以提升预测的准确性和可解释性。
方法分类：
加性模型：认为序列是各成分相加的结果（如 $Y = T + S + R$）。方法包括STL、EMD等，结构清晰，适用面广。
乘性模型：认为序列是各成分相乘的结果（如 $Y = T \times S \times R$）。方法包括VMD等，更适合周期成分强度与趋势值相关的数据（如金融数据）。

3数据预处理

3.5 时序片段化表示

背景：借鉴NLP和CV领域的成功经验，将连续序列转换为离散的"Token"，是构建强大时序模型（尤其是Transformer类）的重要基础。
方法分类：
连续Tokenization：
Point-wise：每个时间点作为一个Token。简单但信息稀疏。
Patch-wise：将连续的时间段打包成一个Patch（如PatchTST）。能学习局部上下文，提升建模质量和效率。
离散Tokenization：
向量量化：使用VQ-VAE等技术将连续值映射到一个有限的离散字典中。
统计分桶：按值域分位数等进行离散化。
优势：便于与语言模型等多模态模型融合，利于构建大模型。

3.6 频率变换

核心思想：时间序列的周期性和局部变化在频域中往往有更清晰的表现。
目的：增强模型对周期、节律的捕捉能力。
常用方法：
DFT/DCT：适用于分析平稳数据中的主导频率成分。
STFT：通过滑窗进行局部傅里叶变换，适合分析整体不平稳但局部平稳的信号。
DWT：具有优秀的时频局部化能力，能同时捕捉趋势和突变点，在金融、生理信号分析中应用广泛。

3.7 图结构建模

适用场景：专门用于多变量时间序列。
核心思想：多变量间的依赖关系是复杂的拓扑结构（图），而非简单的堆叠。通过图神经网络（GCN, GAT, STGCN）可以同时学习变量间的空间依赖和时间上的演化。
图构建方法：
启发式图构建：基于先验知识（如地理位置）、统计指标（如Pearson相关系数、DTW动态时间规整距离）来构建静态图。
学习式图建模：使用神经网络动态学习变量间的关系图，结构可微、可自适应优化。
动态图/时变图：建模变量间关系随时间变化的动态过程，更符合现实系统的演化。

第 4 部分

模型方法全景

四、 模型方法全景

文章回顾了从传统方法到前沿深度学习的各类预测模型：

- 统计建模方法：如ARIMA、指数平滑、状态空间模型，形式简洁、理论严谨。
- 传统机器学习方法：如SVR、GBDT，在中小规模数据上表现优异。
- 深度学习方法：
  - RNN/LSTM/GRU：擅长捕捉序列依赖。
  - CNN/TCN：高效提取局部模式。
  - Transformer：通过自注意力机制建模长期依赖，并衍生出多种为时序优化的变体。
  - MLP：结构简单，通过创新设计焕发新生。
- 生成模型：如VAE、GAN、扩散模型，用于概率预测和生成高质量序列。

4模型方法全景

一、统计建模方法
这类方法基于严格的统计假设，模型形式通常有明确的数学表达式，解释性强。
自回归与滑动平均模型族
AR (自回归模型)： 用自身历史值的线性组合进行预测。
MA (滑动平均模型)： 用过去误差的线性组合进行预测。
ARMA (自回归滑动平均模型)： 结合AR和MA，适用于平稳时间序列。
ARIMA (差分整合移动平均自回归模型)： 在ARMA基础上引入差分，处理非平稳序列。
SARIMA (季节性ARIMA)： 扩展ARIMA，增加了对季节性规律的建模。
指数平滑方法
SES (简单指数平滑)： 适用于无趋势、无季节性的序列。
Holt方法： 在SES基础上增加趋势项。
Holt-Winters方法： 在Holt方法基础上再增加季节性项。
状态空间模型
Kalman Filter (卡尔曼滤波)： 在线性高斯假设下，对系统状态进行最优估计。
HMM (隐马尔可夫模型)： 状态空间离散化，适合建模状态跳变的系统。
高斯混合模型 (GMM)： 用多个高斯分布的加权组合来拟合复杂数据分布，用于多模态预测和异常检测

4模型方法全景

二、传统机器学习方法
这类方法依赖于数据和特征工程，能捕捉非线性关系，比统计模型更灵活。

支持向量回归 (SVR)： 通过核函数在高维空间拟合数据，处理非线性问题。

回归树与集成模型

RF (随机森林)： 通过Bagging集成多棵树。

GBDT (梯度提升决策树)： 通过Boosting集成多棵树。

XGBoost / LightGBM： GBDT的高效、高性能实现。

K近邻回归 (KNN)： 基于相似的历史片段进行预测，适用于有重复模式的序列

4模型方法全景

三、深度学习方法

利用深层神经网络自动学习特征和复杂模式，尤其擅长处理大规模和高维数据。

循环神经网络 (RNN)

LSTM (长短期记忆网络)：引入门控机制，解决长期依赖问题。

GRU (门控循环单元)：LSTM的变体，结构更简单。

DeepAR：基于LSTM的概率预测模型。

SegRNN：采用通道解耦策略的GRU模型。

卷积神经网络 (CNN)

TCN (时序卷积网络)：使用因果卷积和空洞卷积的CNN。

LSTNet：融合CNN和RNN的模型。

SCINet：基于序列拆解与交互的CNN模型。

TimesNet：融合时间卷积与多尺度块设计的CNN模型。

Transformer模型

Autoformer：引入序列分解机制的Transformer。

FEDformer：引入频率增强机制的Transformer。

PatchTST：将时间序列分成片段进行处理的Transformer。

Non-stationary Transformer：引入可学习归一化层以适配非平稳数据。

多层感知机 (MLP)

N-HiTS：引入多尺度采样的MLP模型。

TimeMixer：引入时间滤波器的MLP模型。

SparseTSF / FITS：探索极限参数压缩的MLP模型。

4模型方法全景

四、生成模型
这类模型专注于学习数据的内在分布，可用于概率预测、数据生成和不确定性量化。

变分自编码器 (VAE)
TimeVAE： 引入时间条件向量的VAE。
HyVAE： 支持多模态协同生成的VAE。
生成对抗网络 (GAN)
TimeGAN： 结合监督信号以保留时间顺序的GAN。
Curb-GAN： 在电力和交通领域应用的GAN。
流模型 (Flow-based Models)
FM-TS： 结合频率域结构的流模型。
CFMTS： 使用耦合变换处理高维序列的流模型。
扩散模型 (Diffusion Models)
TSDiff： 通过自我引导提高效率和短期准确性的扩散模型。
LDT： 利用潜在扩散模型，在降水预报等任务中表现优异

4模型方法全景

那生成模型和深度学习方法有什么区别呢?

一、核心定义与范畴的差异

1.深度学习方法（狭义，即判别式时序预测模型）
这里的 “深度学习方法” 特指你之前学习的判别式时序预测模型（RNN/LSTM、CNN/TCN、Transformer/PatchTST、MLP/N-HiTS 等），其范畴是以深度学习架构为基础，直接学习 “输入历史时序→输出未来时序” 的映射关系的模型，属于深度学习技术在时序预测领域的 “直接预测类” 应用。
2.生成模型
生成模型是以学习数据内在概率分布为核心的模型（包括 VAE、GAN、流模型、扩散模型等），它既可以用深度学习架构（如用 LSTM 做 VAE 的编码器、用 Transformer 做扩散模型的去噪器）实现，也可基于传统概率模型构建；其核心不是直接预测未来值，而是先掌握数据的分布规律，再基于分布完成预测、生成等任务。
简言之：生成模型是按 “建模目标” 划分的类别，深度学习方法是按 “技术架构” 划分的类别，二者存在交叉（如 TimeGAN 是深度学习架构的生成模型，PatchTST 是深度学习架构的判别式模型）。

二、核心建模逻辑的本质区别
这是二者最核心的差异，可通过 “船舶污染物排放预测” 的具体建模过程直观理解：

1.深度学习判别式模型的逻辑：“拟合映射，直接预测”以 LSTM 预测船舶排放为例：
输入：船舶过去 10 小时的航速、燃油硫含量、排放数据；

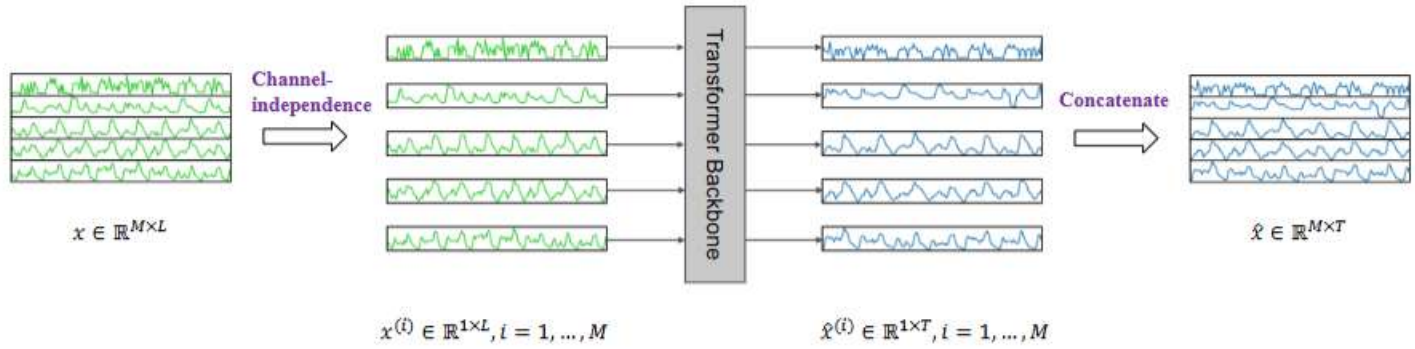
建模：LSTM 通过门控机制学习 “历史数据→下 1 小时排放量” 的非线性映射关系，训练过程是最小化 “预测排放量与真实排放量的误差”；

本质：是“输入→输出” 的函数拟合，不关心排放数据整体的概率分布，只关注预测值的精准度。
2.生成模型的逻辑：“学习分布，按需采样”以 TimeVAE 预测船舶排放为例：
输入：大量船舶的历史排放及工况数据；

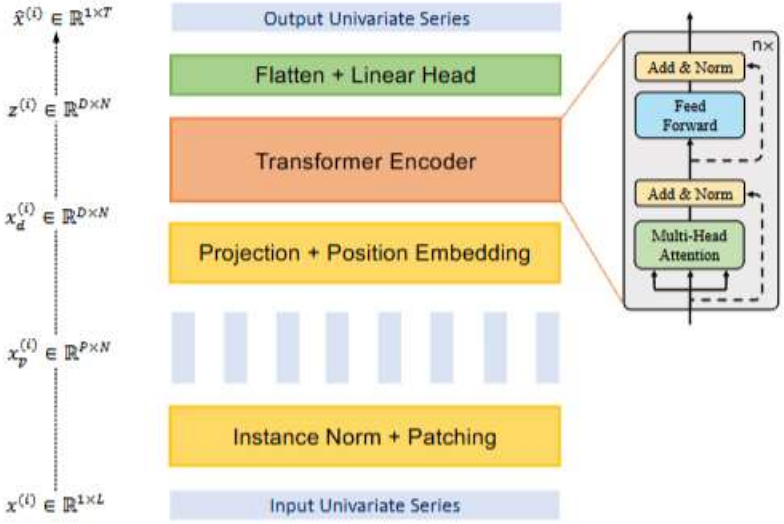
建模：TimeVAE 先通过编码器将排放数据映射为潜在概率分布（如正态分布），再通过解码器从分布中采样生成新的排放序列，训练过程是最小化 “重构误差 + 分布一致性误差”；

本质：是“数据→分布→新数据” 的概率建模 ，先掌握 “什么样的排放数据是合理的”，再基于分布生成 / 预测数据，而非直接拟合输入输出映射。

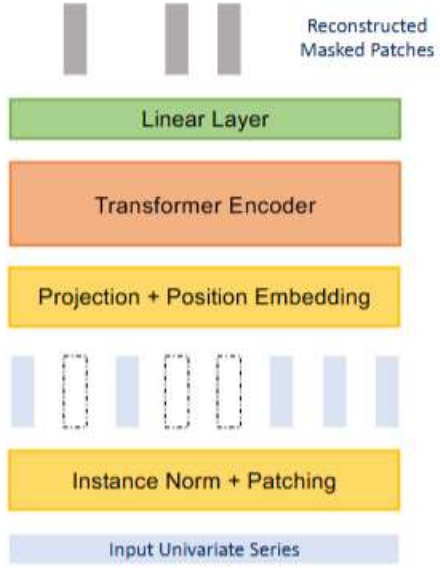
4模型方法全景



(a) PatchTST Model Overview



(b) Transformer Backbone (Supervised)



(c) Transformer Backbone (Self-supervised)

4模型方法全景

2. 两大核心 trick

（1）Channel Independence（通道独立）
操作：把多变量时序的每个变量拆开，单独以 “单变量时序” 的形式进 Transformer。

关键疑问：是不是每个变量都要训练一个 Transformer？
→ **答案**：不用！所有变量共享同一个 Transformer，只是把 “变量数” 和 “batch_size” 合并成同一维度（看源码：把n_vars和bs拼在一起），这样注意力就不会跨变量计算，实现 “通道独立”（类似注意力不能跨 batch 计算）。

（2）Patching（分块）
操作：把单变量的长时序拆成多个短 “子序列分块”（比如把长度为 L 的序列拆成 N 个长度为 P 的子序列），再把分块作为 Transformer 的输入 token。
好处：保留局部时序信息（比 “单个时间点” 当 token 更合理），同时减少输入 token 数，降低 Transformer 的计算复杂度。

3. 数据维度的处理（解决 “多变量 + 分块” 的维度问题）

假设输入是(batch_size, n_vars, L)（batch 数、变量数、序列长度）：

先对每个变量做 Patching，得到(batch_size, n_vars, patch_num, patch_len)；【分块数量、单个 patch 的长度】

把 “变量数” 和 “batch_size” 合并，变成(batch_size*n_vars, patch_num, patch_len)；【通道数量】

经过投影 + 位置编码后，以(batch_size*n_vars, patch_num, d_model)的形状进 Transformer 编码器；

输出后再把维度拆回(batch_size, n_vars, ...)，最终拼接所有变量的结果。

4模型方法全景

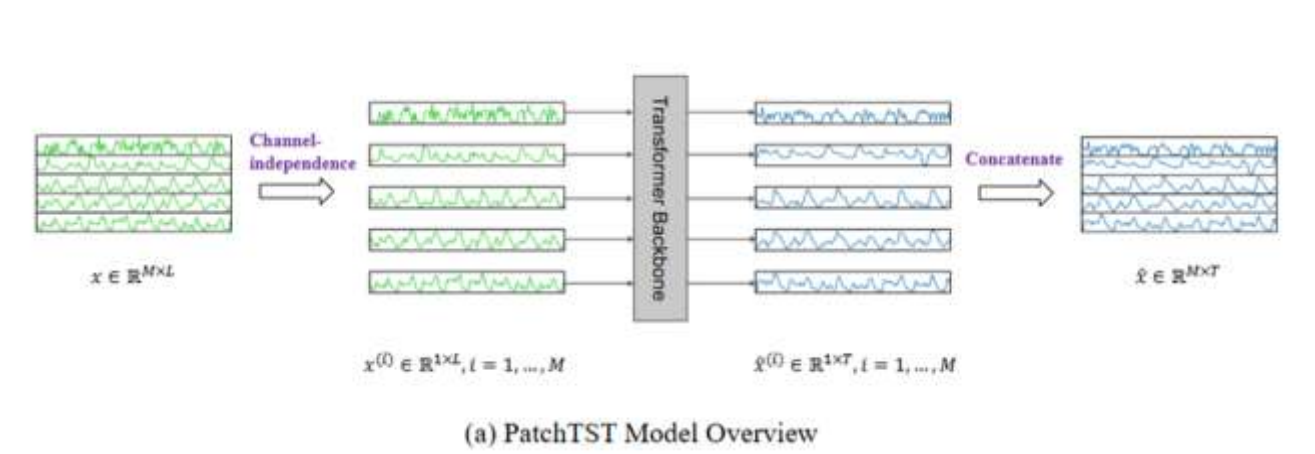
Transformer 编码器的「原始标准输入形状」
标准 Transformer 编码器（不管是 NLP 还是时序任务）的核心输入形状是：(batch_size, seq_len, d_model)
batch_size：批次大小（一次喂给模型的样本数）；
seq_len：序列长度（输入的 token 数量，NLP 里是单词数，时序里原本是时间步数）；
d_model：每个 token 的特征维度（模型潜空间维度，比如 128/256）。

2. 分块（Patching）：「降维」（降的是 seq_len 维度）
以单变量时序为例，输入原本是 (batch_size, 1, L)（L=336 时间步）：
分块后：拆成 (batch_size, 1, patch_num, patch_len)（比如 patch_len=16，patch_num=21）；
投影后：每个 patch 映射到 d_model 维，变成 (batch_size, 1, patch_num, d_model)；
最终输入编码器前：挤压掉 “1（单变量）” 维度，变成 (batch_size, patch_num, d_model)。
核心变化：seq_len 从 336（时间步）降到 21（patch 数）——**seq_len 维度降维**，计算量从 $O(336^2)$ 降到 $O(21^2)$。

3. 通道独立（Channel Independence）：「维度重组，非升 / 降维」
针对多变量时序（比如 n_vars=8 个变量），原始输入是 (batch_size, n_vars, L)：
分块后先变成 (batch_size, n_vars, patch_num, patch_len)；
维度合并：把 batch_size 和 n_vars 拼在一起，变成 (batch_size*n_vars, patch_num, patch_len)；
投影后：(batch_size*n_vars, patch_num, d_model)（这就是喂给编码器的形状）。

4模型方法全景

数据预处理 trick + 标准 Transformer 编码器 + 简单输出层



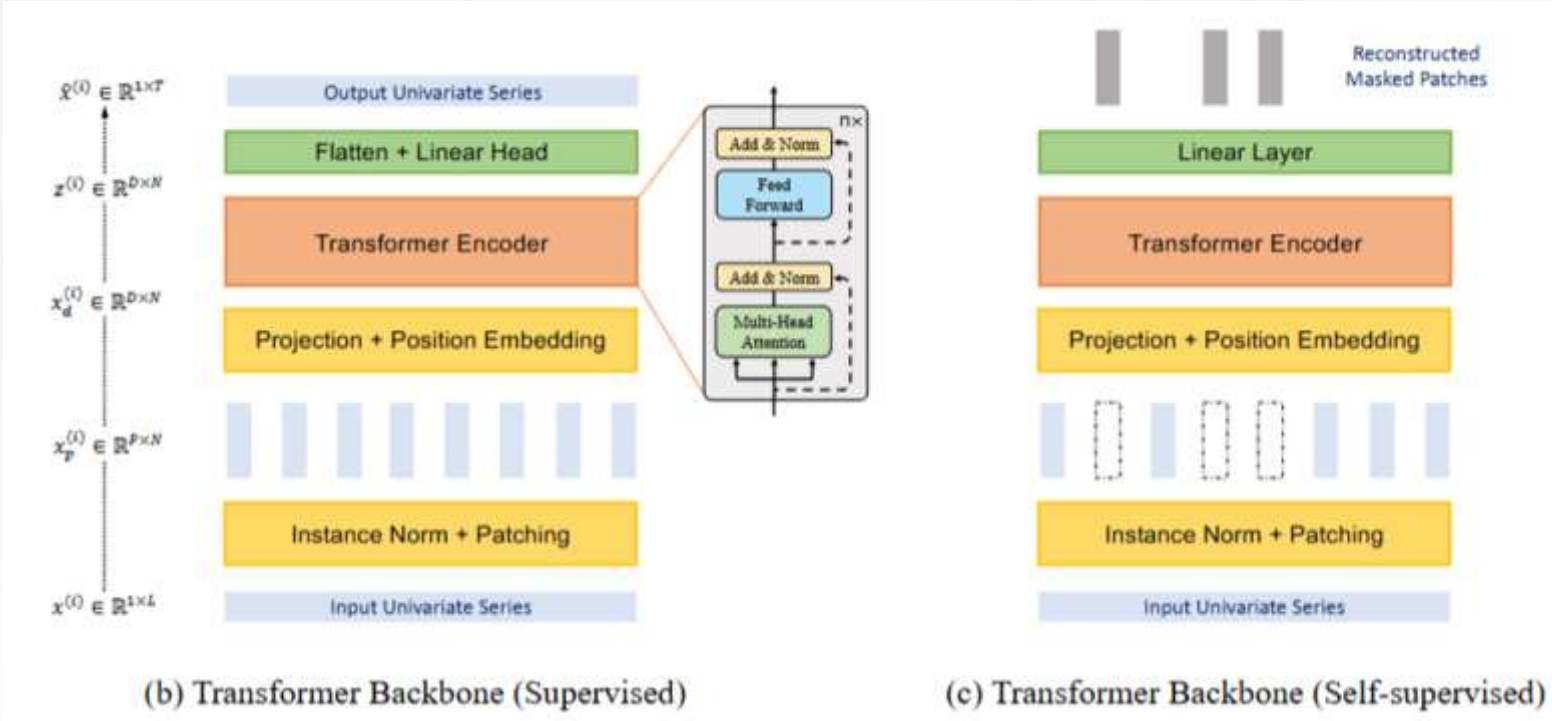
(a) PatchTST Model Overview

1. 什么是「跨变量计算注意力」？（传统做法的问题）
假设输入是（batch_size=2, n_vars=8, L=336）（2 个样本，8 个变量，336 时间步）：

- 传统做法：先把 n_vars 和 L 合并，变成（2, 8*336, d_model），再进编码器；
- 注意力计算：模型会计算「变量 1 的时间步 1」和「变量 8 的时间步 336」之间的注意力（跨变量 + 跨时间）；
- 问题：不同变量的物理意义 / 量纲可能完全不同（比如变量 1 是温度，变量 8 是湿度），强行计算跨变量注意力会让模型学混规律（比如把温度的周期和湿度的趋势错误关联）。

4模型方法全景

数据预处理 trick + 标准 Transformer 编码器 + 简单输出层



(b) Transformer Backbone (Supervised)

(c) Transformer Backbone (Self-supervised)

4模型方法全景

通道独立就是个不建模变量间关系

Transformer 确实**无法计算跨变量的注意力**，也就是 “不主动建模变量间的关联”——这是通道独立的核心，（没有复杂的跨变量建模，只是 “回避” 了这个问题）。

PatchTST 在**输出阶段**会把所有变量的预测结果拼接，最终损失是 “所有变量的误差平均”，模型会通过 “共享 Transformer 权重” 间接学习变量间的共性规律（比如所有变量都有日周期，权重会捕捉这种通用模式），但确实没有 “显式建模变量间的依赖”（比如温度→用电量的因果关系）。

通道独立≈“对每个变量单独训练一个相同结构的 Transformer，最后合并结果”

4模型方法全景

patch 其实就是个卷积

卷积是 “滑动窗口 + 局部聚合”（用固定窗口扫过序列 / 图像，提取局部特征）；PatchTST 的 patch 分块也是 “滑动窗口（步幅 S）+ 局部聚合（把窗口内 P 个时间步合并成 1 个 patch）”，从操作形式上看，两者确实是 “换汤不换药” 的局部特征提取逻辑。

卷积是 “参数共享的线性变换 + 固定感受野”（窗口内的权重是固定的，只能捕捉预设的局部规律）；而 PatchTST 的 patch 只是 “数据预处理步骤”—— 分块后并没有卷积操作，而是把 patch 直接送 Transformer 做

操作环节	卷积（式子）	PatchTST 分块（式子）	核心共性
滑动窗口	kernel_size=16, stride=8 → 输出 42 个单元	patch_len=16, stride=8 → 输出 42 个 patch	窗口大小 / 步幅一致，输出单元数一致
局部聚合	(32,1,336) → (32,1,42)（加权聚合）	(32,1,336) → (32,1,42,16) → (32,42,128)（合并聚合）	把局部 16 个时间步聚合成 1 个单元
最终目标	提取局部时序特征	提取局部时序特征（为 Transformer 做输入）	都是为了抓局部规律

卷积是 “加权聚合（带参数）”，PatchTST 是 “合并聚合（无参数）”—— 操作形式（滑动 + 聚合）

第 5 部分

学习范式演进

5学习范式演进

五、 学习范式演进

探讨了提升模型可迁移性、可信度和性能的新兴学习范式：

•自监督学习：通过对比学习、掩码自编码、自回归预训练等方式，从无标签数据中学习通用表示。
•领域自适应：解决分布外泛化问题，使模型能适应新领域。
•基于语言模型的预测：探索利用大语言模型的语义理解和推理能力进行时序预测，包括参数调优和零样本提示两种路径。
•时间序列基座模型：旨在构建大规模预训练、具备强大跨领域泛化能力的基础模型。
•可解释性与可信预测：结合因果发现和物理引导神经网络，提升模型的透明度和可靠性。
•鲁棒性：研究模型对抗攻击的防御机制。
•隐私保护：通过联邦学习等框架，在保护数据隐私的前提下进行模型训练。

5学习范式演进

5.1 自监督学习

核心目标：在不依赖大量昂贵标注数据的情况下，通过从无标签数据本身挖掘监督信号，学习强大、通用的时间序列表示。

动机：标注时间序列数据成本高，且手工设计的特征往往泛化能力差。自监督学习能充分利用海量无标签数据，为下游预测任务提供一个良好的模型初始化或特征提取器。

主要技术路径：

对比学习：

思路：通过数据增强（如裁剪、遮蔽、抖动）构造“正样本对”（同一序列的不同视角）和“负样本对”（不同序列），训练模型学习到区分性表示，使相似序列的表征在空间中靠近，不相似的远离。

代表性工作：TS2Vec（通用框架）、TNC、TS-TCC。

掩码自编码器：

思路：随机遮蔽输入序列的一部分，然后训练模型（通常是Transformer）来重建被遮蔽的部分。这个过程迫使模型学习序列的内结构和依赖关系。

代表性工作：

原始空间：TST。

离散潜在空间：TimeMAE、CrossTimeNet（便于跨领域预训练）。

流行空间：SimMTM（通过加权聚合实现更结构化的语义建模）。

自回归式预训练：

思路：借鉴语言模型的思路，使用历史数据预测未来值，从而直接学习序列的动态演化规律。

代表性工作：TimeDART（结合扩散模型）、Timer（统一序列格式，采用GPT风格架构）。

5学习范式演进

5.2 领域自适应

核心目标：解决分布外泛化问题，让在“**源领域**”上训练的模型能够有效地适应并应用于数据分布不同的“**目标领域**”。

挑战：时间序列中普遍存在的分布漂移是模型在实际部署中性能下降的主要原因。

方法：

SLARDA：通过自监督对比预训练学习迁移性强的表示，并同时考虑时间动态和特征对齐。

FOIL：在存在未观测混淆变量的情况下，通过设计替代损失函数来学习更稳定的因果表示。

5学习范式演进

5.3 基于语言模型的预测方法

核心思想：利用大语言模型在预训练中获得的世界知识、语义理解和推理能力，来处理时间序列预测任务。

实现方式：通常将时间序列数据（通过投影、分桶、离散化等方式）转换为LLM能够理解的“伪文本”序列，然后利用LLM的生成或理解能力进行预测。

两种主要技术路径：

参数调优型：

对预训练好的LLM进行微调，使其适配时序数据。

创新点多样：

预测方式：自回归（AutoTimes） vs. 一步预测（FPT）。【先预测下 1 小时的排放量，再用这个预测值结合历史数据】

训练范式：引入自监督预训练（CrossTimeNet）。【先打基础再学专项】

文本融合：结合结构化数据与文本描述（TEMPO）。

参数更新：全参数微调 vs. 高效参数微调如 LoRA（LLM4TS）。

输入表示：嵌入表示（Chronos, UniTime） vs. 时序-文本特征融合（TimeLLM）。

零调优型：

不更新LLM参数，完全通过提示工程来引导模型完成任务。

方法：

Prompt设计：LLMTime（将背景转为文本），LSTPrompt（引入“思考间歇”）。

上下文增强：采用检索增强生成（RAG），如 TimeRAF、TimeRAG，从历史中寻找相似序列作为提示的一部分。

智能体协作：TESSA 利用多个智能体分工合作来解析和推理时序数据。

挑战：目前零样本能力有限，部分任务效果接近随机，表明LLM的“时间序列推理”能力仍处于早期阶段。

5学习范式演进

5.4 时间序列基座模型
核心目标：构建在大规模、跨领域时序数据上预训练的基础模型，使其具备强大的零样本/少样本泛化能力，即“**一个模型解决所有问题**”。
驱动力：借鉴NLP和CV中“基础模型”的成功，旨在改变时序预测领域模型碎片化的现状。
设计理念：趋向于“简洁通用”，避免对特定任务的过拟合。
代表性工作：
Moirai：统一架构，在包含270亿时间步的LOTSA数据集上预训练。
Timer：统一任务格式，采用GPT式自回归结构。
关键支撑：
大规模数据集（如LOTSA）。
统一建模框架。
利用合成数据（如TimesFM用了20%的合成数据）提升泛化能力。
这标志着时序研究正迈向 “以数据为中心” 的新范式。

5学习范式演进

5.5 可解释性与可信预测
核心目标：让模型不仅能预测，还能解释“为什么”这么预测，增加模型的透明度和可信度。

两大主流方向：

因果发现：
旨在揭示变量之间潜在的因果关系，而不仅仅是相关关系。
方法从传统的Granger因果分析、动态贝叶斯网络，发展到与深度学习结合的非线性方法。
有助于识别真正的驱动因素，提升模型的泛化性和可解释性。

物理引导神经网络：
将已知的物理定律（如微分方程、守恒律）作为约束嵌入到神经网络的结构或损失函数中。
确保预测结果不仅拟合数据，而且符合物理规律。
在数据稀疏或噪声大的科学计算、工程领域尤为重要。

5学习范式演进

5.6 鲁棒性

核心目标：保护模型免受对抗性攻击的干扰，确保其在恶意环境下的稳定性和可靠性。

威胁：

对抗攻击：对输入施加精心构造的、人眼难以察觉的微小扰动，就能导致模型产生严重错误预测。

后门攻击：在训练数据中植入“触发器”，一旦在预测时遇到该触发器，模型就会输出特定错误结果（如BACKTIME所揭示）。

防御机制：

对抗训练、随机平滑。

[随机平滑：给模型的输入数据加一点可控的“小噪声”，让模型习惯数据的轻微波动，就算遇到攻击带来的小扰动，也不会大幅偏离正确预测；

对抗训练：提前在训练数据里混入一些“轻度攻击样本”，让模型在学习阶段就见过“坑”，等实际遇到攻击时，就能识别并抵御，同时还能保住原本的预测精度。]

RDAT：结合强化学习和自蒸馏，专门提升交通预测模型的鲁棒性。[主动学习识别攻击]

5学习范式演进

5.7 隐私保护

核心目标：在联合多方数据训练强大模型的同时，保护原始数据的隐私，不泄露给其他参与方或中央服务器。

主流解决方案：联邦学习。

[“数据不动模型动”：各参与方用本地数据训练本地模型，仅分享模型参数（或梯度）等中间结果，由中心节点聚合更新，反复迭代后得到性能接近集中式训练（用所有数据）的模型，既保护数据隐私，又能利用多源数据提升模型泛化能力。]

代表性工作：

CNFGNN：结合图神经网络与联邦学习，建模跨节点的时空依赖而不共享数据。

MetePFL：引入提示学习和图模型，处理不同客户端数据的异质性问题。

Time-FFM：构建了融合LLM的联邦时序基础模型，实现隐私保护下的个性化预测。

第 6 部分

未来趋势

3未来趋势

论文最后展望了时间序列预测领域未来的发展方向：

时间序列基础模型：构建能够统一处理不同领域、频率和任务的通用基础模型，并探索其扩展规律和多模态融合。

可信时间序列预测：深度融合可解释AI和隐私保护技术，构建既准确又安全、可靠、透明的预测系统。

新型建模范式：引入AutoML、物理引导神经网络等“跨界”方法，推动建模范式的革命。

全面评估体系：建立覆盖多分布、多维度指标的评估基准，推动模型从追求“跑分”到全面发展“认知”能力。

这篇综述系统性地勾勒了时间序列预测领域的技术全貌，从基础定义、现实挑战、关键技术（预处理与模型），到前沿范式（自监督、大模型、可信AI），并最终指向未来的智能化发展趋势。它指出，该领域正从专注于单一模型结构设计，迈向构建通用、可信、高效的下一代预测系统的范式革命。

谢谢！