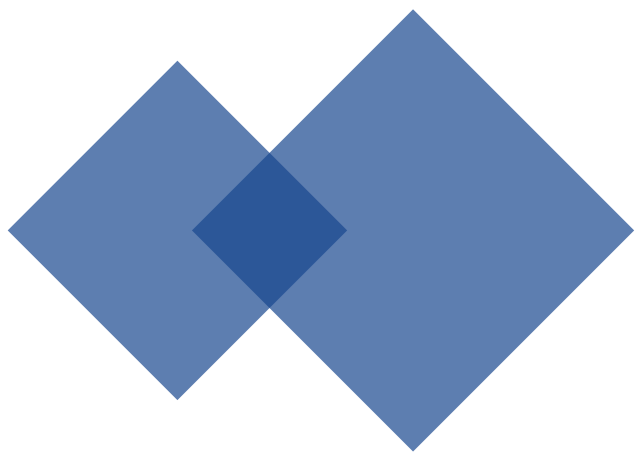


# TimesFM

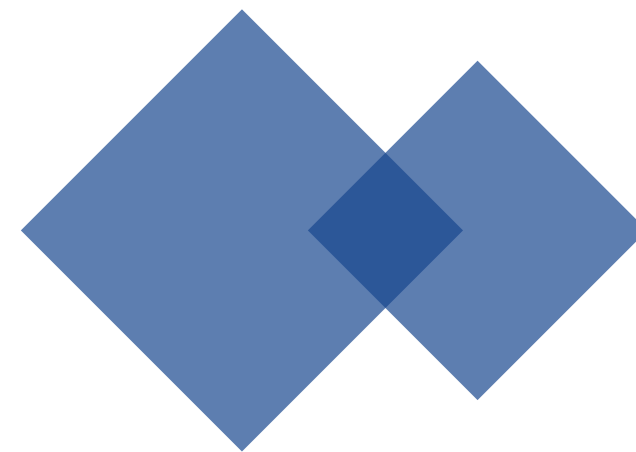
A Decoder-only **F**oundation  
**M**odel For **T**ime-series  
Forecasting

**LLM for TS**  
**From Google Research**



# TimesFM

A Decoder-only  
Foundation **M**odel For  
**T**ime-series Forecasting



24.5.21

Presented by Yyyq



➤ **LLM (Large Language Model) :**

- zero-shot learning
- 强大的泛化能力，不需要对每个任务都进行单独的训练
- 基于预训练数据进行推理，其效果依赖于预训练数据的广度和质量

➤ **LLM for TS:**

- 在时序数据上训练的预训练模型能否学习到时间模式
- 与NLP不同，时间序列没有明确定义的词汇或语法
- 需要支持具有不同历史长度(context)、预测长度(horizon)和时间粒度的预测
- 时间序列数据不容易获得



➤ **TimesFM:**

- 预训练模型  
patched input, decoder-only, attention model
- 预训练数据集  
——大型时序语料库：包含真实数据集 + 合成数据集
- 预测数据集  
对不同领域、不同预测范围和不同时间粒度做zero-shot预测



➤ **TimesFM优势:**

- much smaller  
parameter size— $200M$ , pretraining data size— $O(100B)$  timepoints
- 准确性  
zero-shot性能接近于在不同时序数据集上的完全监督方法的准确性
- 与其他LLMTS (GPT-3 and LLama-2) 相比  
以一小部分成本获得更好的性能



➤ **完全监督模型**

例如FedFormer等

➤ **数据集之间的迁移学习模型**

PatchTST 和N-BEATS

➤ **预训练的基准LLM：GPT-3 和 LLAMA-2**

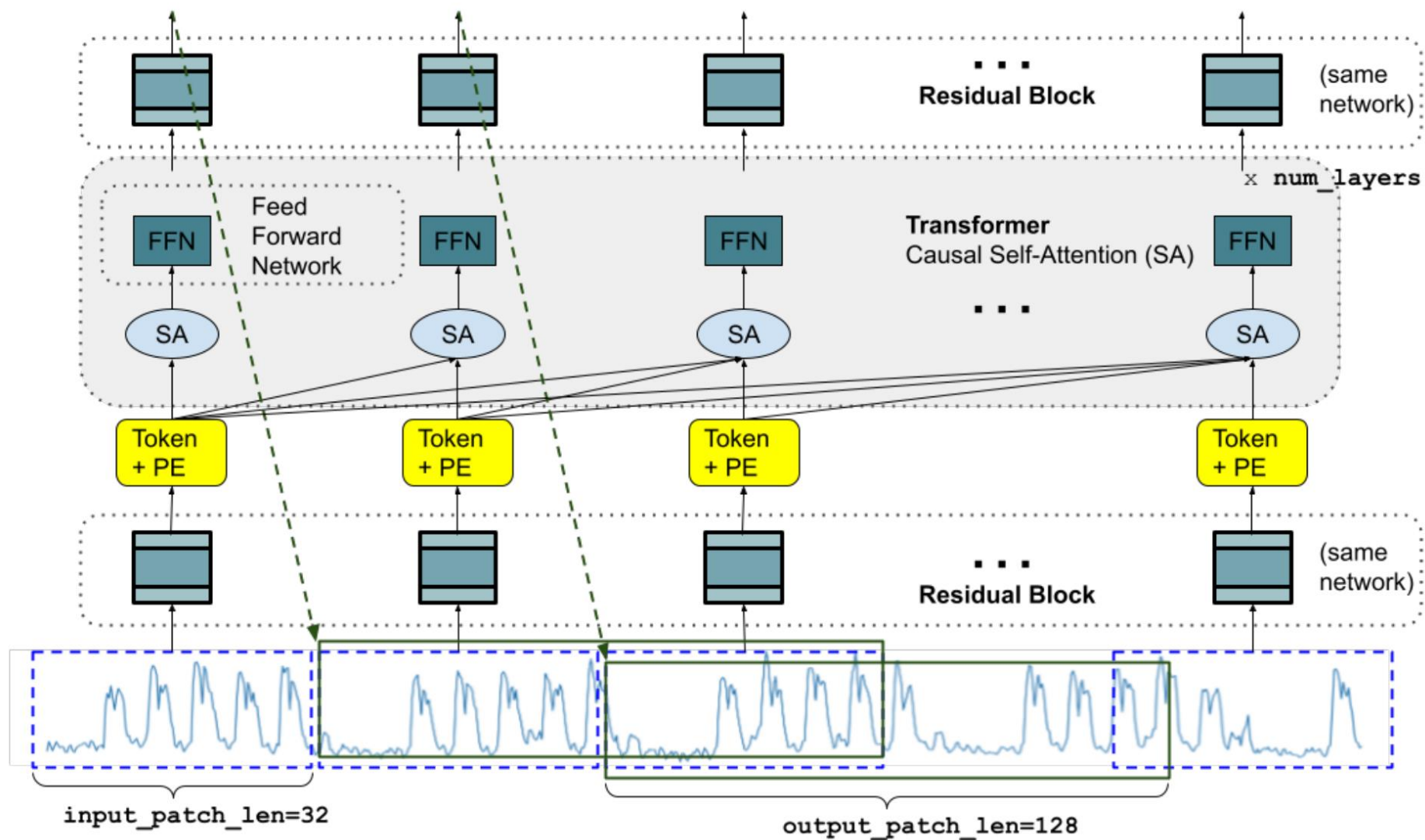
- 例如One fits all（阿里达摩院 nips2023）和 LLM4TS：为时间序列预测任务微调 GPT-2



- **预测任务**: 过去C个时间点作为上下文 (C即: 序列1~L) , 预测未来H个时间点

$$f : (\mathbf{y}_{1:L}) \longrightarrow \hat{\mathbf{y}}_{L+1:L+H}.$$

- **LLM for TS**: 学习一个基础模型, map any time-series context to horizon







- { 时间序列: 分patch, 每个patch的长度为p;  $\tilde{\mathbf{y}}_j = \mathbf{y}_{p(j-1)+1:pj}$   
掩码块 (Patch Masking) : 1—mask, 同样以p为patch;  $\tilde{\mathbf{m}}_j = \mathbf{m}_{p(j-1)+1:pj}$
- 线性层: 升维至model\_dim
- 残差块
- 位置编码  $\text{PE}_j$

$$\mathbf{t}_j = \text{InputResidualBlock}(\tilde{\mathbf{y}}_j \odot (1 - \tilde{\mathbf{m}}_j)) + \text{PE}_j$$

➤ **Input Layer:**  $\tilde{\mathbf{m}}_j = \mathbf{m}_{p(j-1)+1:pj}$

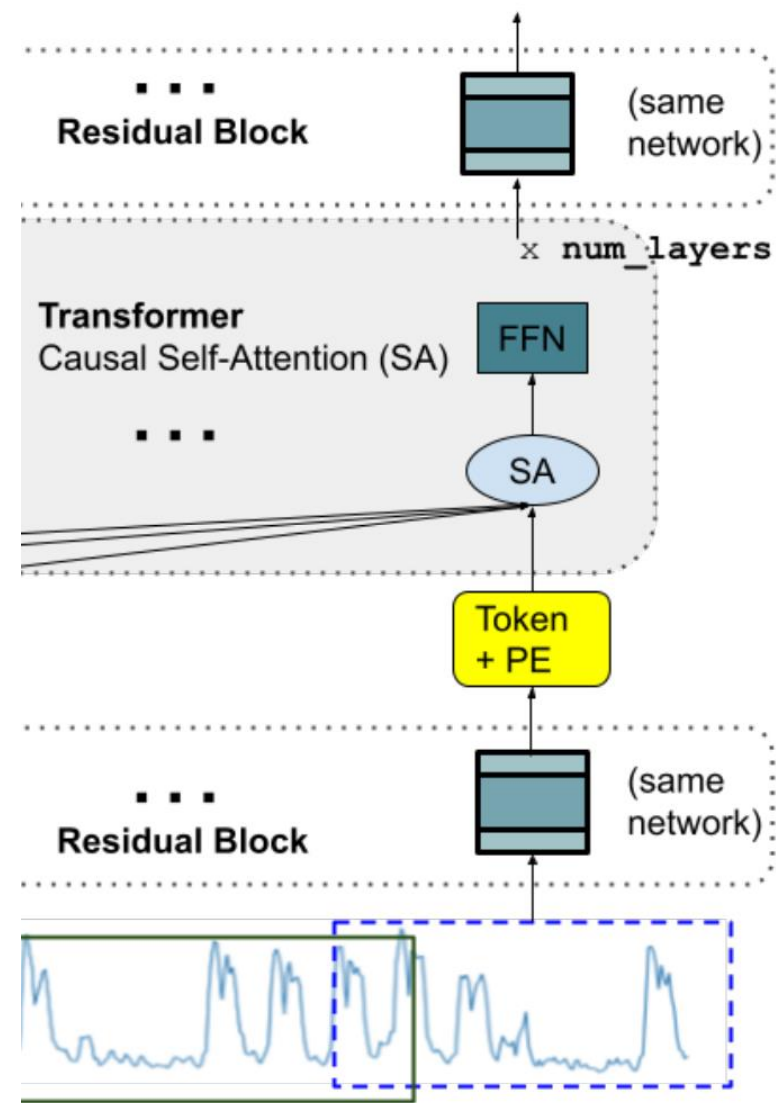
$$\mathbf{t}_j = \text{InputResidualBlock}(\tilde{\mathbf{y}}_j \odot (1 - \tilde{\mathbf{m}}_j)) + \text{PE}_j$$

➤ **Stacked Transformer:**  $\dot{\mathbf{m}}_j, \min\{\mathbf{m}_{p(j-1)+1:pj}\}$

$$\mathbf{o}_j = \text{StackedTransformer}((\mathbf{t}_1, \dot{\mathbf{m}}_1), \dots, (\mathbf{t}_j, \dot{\mathbf{m}}_j)),$$

➤ **Output Layer:** 预测h步

$$\hat{\mathbf{y}}_{pj+1:pj+h} = \text{OutputResidualBlock}(\mathbf{o}_j).$$





### ➤ 指导原则

- 给定一个输入patch, 模型利用所有过去patch来预测下一个patch
- 允许预测输出patch比输入patch长

### ➤ 一种中间策略

- 直接预测全视界 (dlinear) 比多步自回归解码的精度更高
- Zero-shot learning: 视界长度先验未知



➤ 指导原则

- 给定一个输入patch, 模型利用所有过去patch来预测下一个patch
- 允许预测输出patch比输入patch长

➤ 一种中间策略

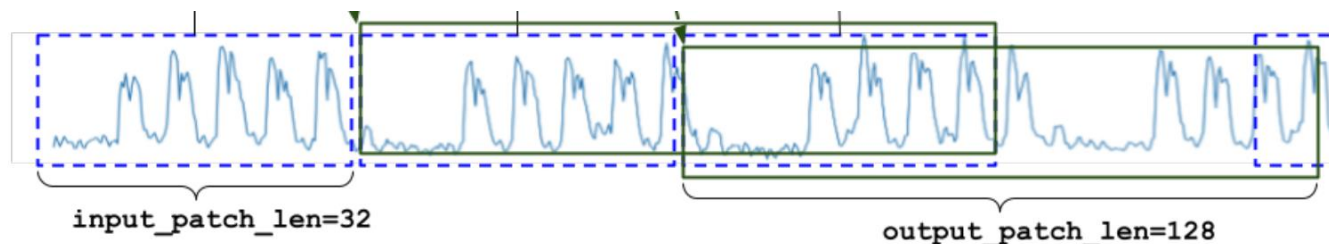
例: 输入 patch = 32, 输出 patch = 128

- 训练过程:  $32 \rightarrow 128$ ,  $32+32 \rightarrow 65 \sim 192$ ,  $32+32+32 \rightarrow 97 \sim 224$ , .....
- 推理过程:

假设需要256预测256,

模型先生成256~128,

再把生成的输出合并为输入, 得到385~512





➤ mini-batch gradient descent

- 每次参数更新使用一个mini-batch (batchsize = 32~256) 的训练样本

➤ 随机掩码采样：对于batch中的每个时间序列

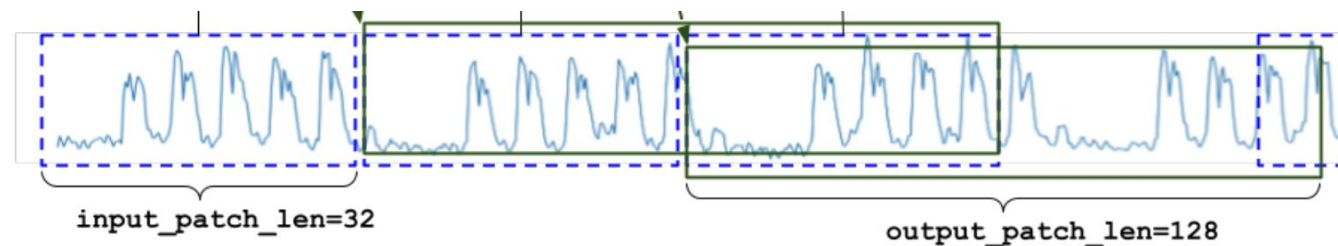
- 设置掩码：屏蔽掉第一个输入块的一部分

$$\mathbf{m}_{1:r} = 1, r \text{ between } 0 \text{ and } p - 1$$

- 目的：覆盖所有上下文长度（历史序列）

例：L = 512, p = 32

r = 4: 从28=32-4个时间点开始预测



### ➤ 合成数据：3M条序列，每条2048个timepoints

组合器：ARMA过程，季节模式(余弦，正弦)，趋势(线性、指数)和阶跃函数

### ➤ 真实世界数据源

#### • M4: 100k

由年度、季度、月度和其他（每周、每日和每小时）数据的时间序列组成，数据覆盖经济、金融、商业、社会、环境等

#### • ETT、ECL、weather、traffic

交通和电力：长期预测数据集，分别有>800和>300个时间序列，每个时间序列都有数万个时间点。

### ➤ Wiki Pageviews: 300B timepoints

2012年1月~2023年11月，所有维基媒体页面的浏览量

### ➤ Google Trends: 0.5B timepoints

2007年~2022年，谷歌搜索兴趣，22k个头部查询；

Table 1: Composition of TimesFM pretraining dataset.

Dataset	Granularity	# Time series	# Time points
Synthetic		3,000,000	6,144,000,000
Electricity	Hourly	321	8,443,584
Traffic	Hourly	862	15,122,928
Weather [ZZP <sup>+</sup> 21]	10 Min	42	2,213,232
Favorita Sales	Daily	111,840	139,179,538
LibCity [WJJ <sup>+</sup> 23]	15 Min	6,159	34,253,622
M4 hourly	Hourly	414	353,500
M4 daily	Daily	4,227	9,964,658
M4 monthly	Monthly	48,000	10,382,411
M4 quarterly	Quarterly	24,000	2,214,108
M4 yearly	Yearly	22,739	840,644
Wiki hourly	Hourly	5,608,693	239,110,787,496
Wiki daily	Daily	68,448,204	115,143,501,240
Wiki weekly	Weekly	66,579,850	16,414,251,948
Wiki monthly	Monthly	63,151,306	3,789,760,907
Trends hourly	Hourly	22,435	393,043,680
Trends daily	Daily	22,435	122,921,365
Trends weekly	Weekly	22,435	16,585,438
Trends monthly	Monthly	22,435	3,821,760

### ➤ 数据集混合

- Hourly, Daily: 最大context长度512
- Weekly: 最大256
- Monthly: 最大64

### ➤ 可逆归一化

- 根据第一个patch的均值和标准差缩放

Table 1: Composition of TimesFM pretraining dataset.

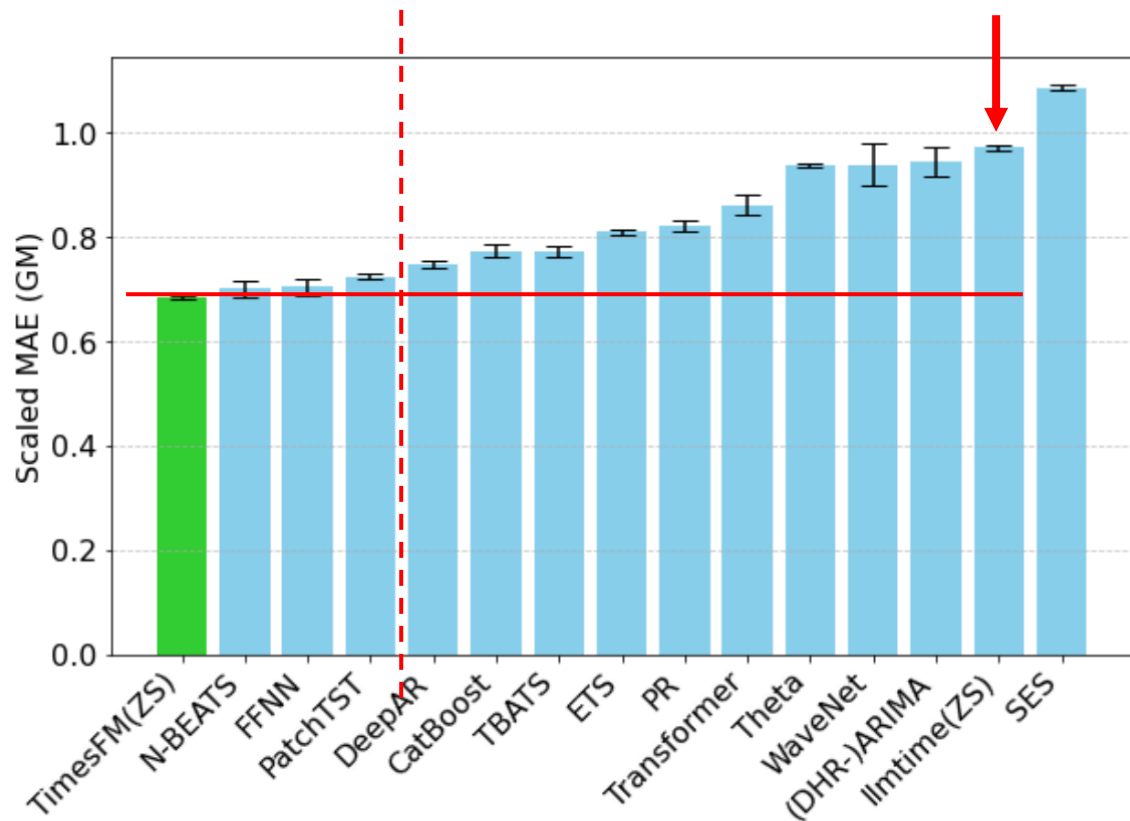
Dataset	Granularity	# Time series	# Time points
Synthetic		3,000,000	6,144,000,000
Electricity	Hourly	321	8,443,584
Traffic	Hourly	862	15,122,928
Weather [ZZP <sup>+</sup> 21]	10 Min	42	2,213,232
Favorita Sales	Daily	111,840	139,179,538
LibCity [WJJ <sup>+</sup> 23]	15 Min	6,159	34,253,622
M4 hourly	Hourly	414	353,500
M4 daily	Daily	4,227	9,964,658
M4 monthly	Monthly	48,000	10,382,411
M4 quarterly	Quarterly	24,000	2,214,108
M4 yearly	Yearly	22,739	840,644
Wiki hourly	Hourly	5,608,693	239,110,787,496
Wiki daily	Daily	68,448,204	115,143,501,240
Wiki weekly	Weekly	66,579,850	16,414,251,948
Wiki monthly	Monthly	63,151,306	3,789,760,907
Trends hourly	Hourly	22,435	393,043,680
Trends daily	Daily	22,435	122,921,365
Trends weekly	Weekly	22,435	16,585,438
Trends monthly	Monthly	22,435	3,821,760

## 06



## 实验：实验结果——Monash

包含30个不同数据集的集合，涵盖了从分钟到年的时间粒度和金融、需求预测、天气、交通等多个领域。

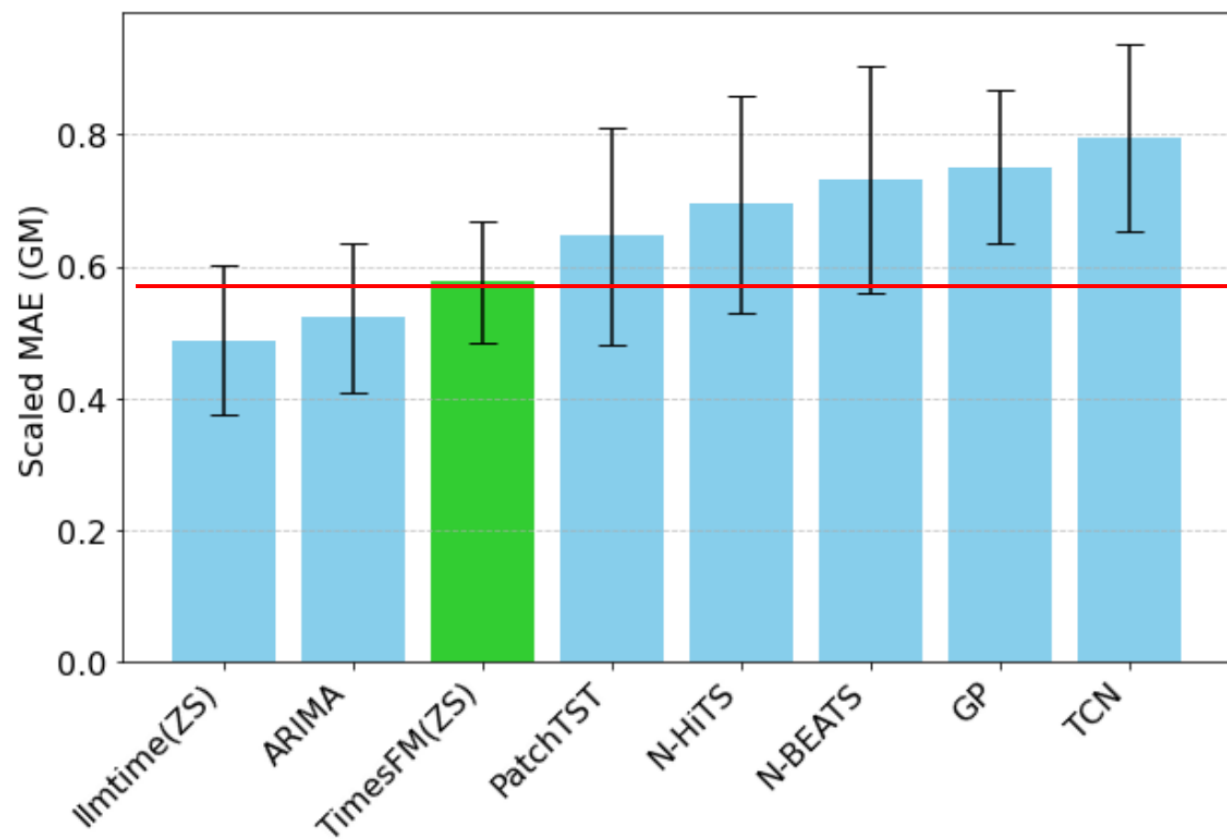


(a) Monash Archive [GBW<sup>+</sup>21]





包含8个包含季节性和加性+乘性趋势的单变量数据集。



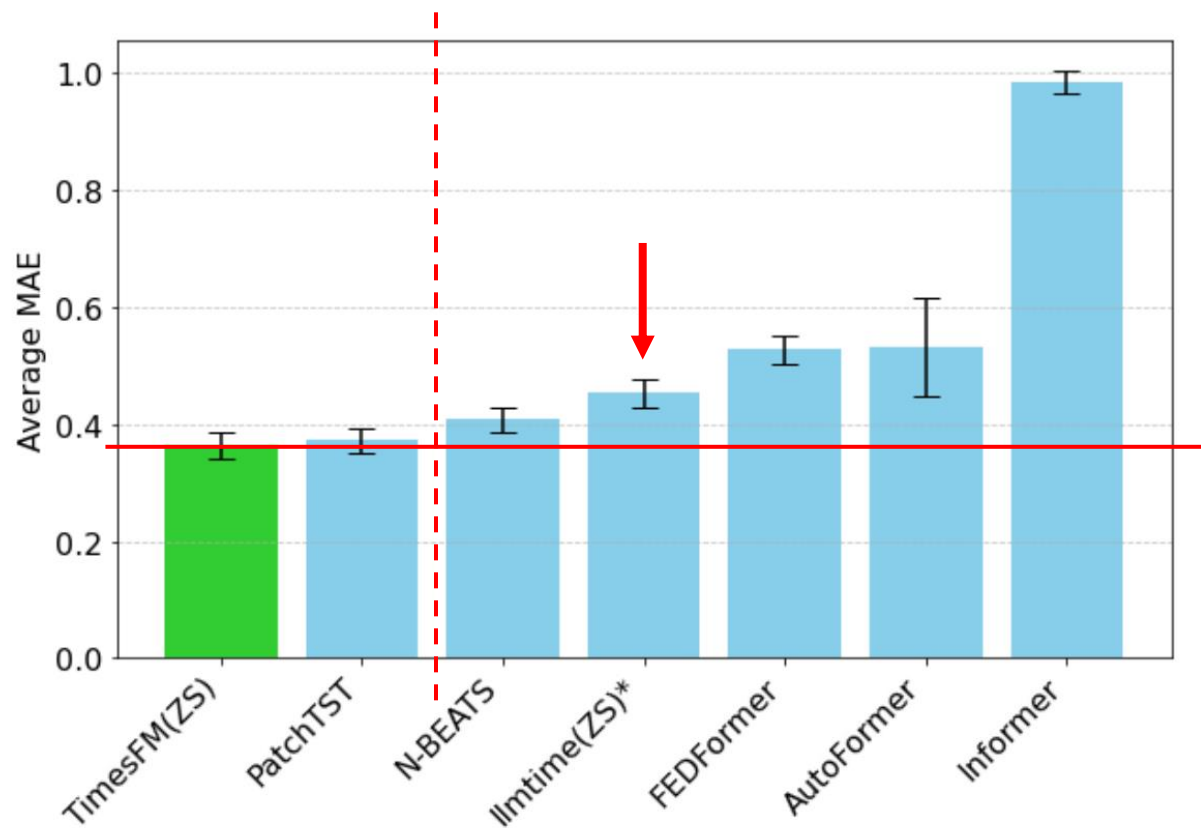
(b) Darts [HLP<sup>+</sup> 22]

06



## 实验：实验结果——Informer

512预测96和192



(c) ETT (Horizons 96 and 192) [ZZP<sup>+</sup> 21]



10%数据进行微调

Dataset		TimesFM(FT)	GPT4TS(FT)	DLinear	PatchTST	TimeNet	FEDFormer	Autoformer
ETTh1	96	0.398	0.456	0.495	0.485	0.628	0.499	0.552
	192	0.424	0.516	0.538	0.524	0.593	0.555	0.598
	336	0.436	0.535	0.622	0.550	0.648	0.574	0.619
	720	0.445	0.591	0.743	0.610	0.641	0.614	0.616
	Avg	<b>0.426</b>	0.525	0.600	0.542	0.628	0.561	0.596
ETTh2	96	0.356	0.374	0.411	0.389	0.409	0.416	0.451
	192	0.400	0.411	0.519	0.414	0.467	0.474	0.477
	336	0.428	0.433	0.572	0.441	0.494	0.501	0.543
	720	0.457	0.464	0.648	0.480	0.491	0.509	0.523
	Avg	<b>0.410</b>	0.421	0.538	0.431	0.465	0.475	0.499
ETTm1	96	0.345	0.404	0.392	0.419	0.501	0.518	0.614
	192	0.374	0.423	0.412	0.434	0.528	0.546	0.592
	336	0.397	0.439	0.434	0.454	0.568	0.775	0.677
	720	0.436	0.498	0.477	0.556	0.549	0.579	0.630
	Avg	<b>0.388</b>	0.441	0.429	0.466	0.537	0.605	0.628
ETTm2	96	0.263	0.269	0.303	0.274	0.285	0.399	0.454
	192	0.309	0.309	0.345	0.317	0.323	0.379	0.691
	336	0.349	0.346	0.385	0.353	0.353	0.559	1.407
	720	0.415	0.417	0.440	0.427	0.449	0.614	1.166
	Avg	<b>0.334</b>	0.335	0.368	0.343	0.353	0.488	0.930



# 谢谢观看

MANY THANKS !



24.5.21

