In-context Time Series Predictor

Jiecheng Lu Yan Sun Shihao Yang

Georgia Institute of Technology

Introduction

Recent large language models (LLMs) exhibit strong in-context learning abilities without parameter updates. To leverage this for time series forecasting (TSF), we propose representing TSF tasks directly as sequences of (lookback, future) token pairs, aligning naturally with in-context learning mechanisms. Our method avoids overfitting, enhances efficiency, and outperforms existing Transformer-based models in full-data, few-shot, and zero-shot forecasting scenarios.

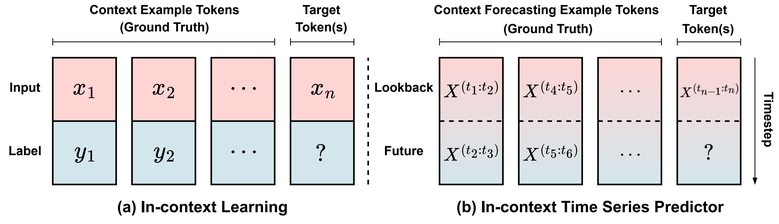

Figure 1. Overview of in-context TSF learning in our setup.

TSF Transformers in an ICL view

Time Series Forecasting Consider a time series $X \in \mathbb{R}^{C\times L}$, composed of historical data $X_I \in \mathbb{R}^{C\times L_I}$ and future data $X_P \in \mathbb{R}^{C\times L_P}$, where $L = L_I + L_P$. The forecasting task aims to find a predictor: $f : \mathbb{R}^{C\times L_I} \to \mathbb{R}^{C\times L_P}, \quad \widehat{X}_P = f(X_I)$.

Transformer Architecture We use a pre-norm Transformer with input tokens $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N] \in \mathbb{R}^{D\times N}$. Each Transformer layer $\texttt{TF}_k$ computes:

$$\mathbf{Z}_k = \mathbf{Z}_{k-1} + \texttt{Attn}_k(\texttt{LN}(\mathbf{Z}_{k-1})) + \texttt{LN}(\texttt{FFN}_k(\mathbf{Z}_{k-1} + \texttt{Attn}_k(\texttt{LN}(\mathbf{Z}_{k-1})))),$$

with the final output given by $\mathbf{Z}_K = \texttt{TF}(\mathbf{Z}_0)$, where $\mathbf{Z}_0 = \mathbf{Z}$.

ICL Given datapoints $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, ICL predicts target $\mathbf{y}_i$ from context examples $\{(\mathbf{x}_j, \mathbf{y}_j)\}_{j<i}$ and current input $\mathbf{x}_i$. The input token format for the i-th datapoint is:

$$\mathbf{Z}^{(a+b)\times i} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_{i-1} & \mathbf{p}_i \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{i-1} & \mathbf{x}_i \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_{i-1} & \mathbf{o} \end{bmatrix},$$

where $\mathbf{p}.$ are position embeddings, $\mathbf{o}$ is a 0-placeholder. Predictions are as:

$$\hat{\mathbf{Z}} = \mathbf{W}_{\text{out}}\texttt{TF}(\mathbf{W}_{\text{in}}\mathbf{Z} + \mathbf{b}_{\text{in}}) + \mathbf{b}_{\text{out}},$$

with linear projection parameters $\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{out}}$, and biases $\mathbf{b}_{\text{in}}, \mathbf{b}_{\text{out}}$.

Temporal-wise Transformer Constructs tokens from multiple series along temporal dimensions:

$$\mathbf{Z}^{C\times L} = \begin{bmatrix} \mathbf{P}_{1:L_I} & \mathbf{P}_{L_I+1:L_I+L_P} \\ X_I & \mathbf{O} \end{bmatrix}.$$

Each token represents multi-channel values at timesteps, learning mappings $f_t^* : \mathbf{p}_t \to [X_1^{(t)}, \ldots, X_C^{(t)}]^\top$, with overfitting on weak inter-channel dependencies.

Series-wise Transformer Transposes inputs, making each token represent an individual series:

$$\mathbf{Z}^{L\times C} = \begin{bmatrix} \mathbf{P}_{1:C} \\ X_I^\top \\ \mathbf{O} \end{bmatrix}.$$

Each token captures temporal dependencies within a single series, learning mappings $f_j^* : X_j^{(1:L_I)} \to X_j^{(L_I+1:L_I+L_P)}$. However, it lacks ground truth context examples, limiting generalization and few-shot performance.

TSF Transformers in an ICL view

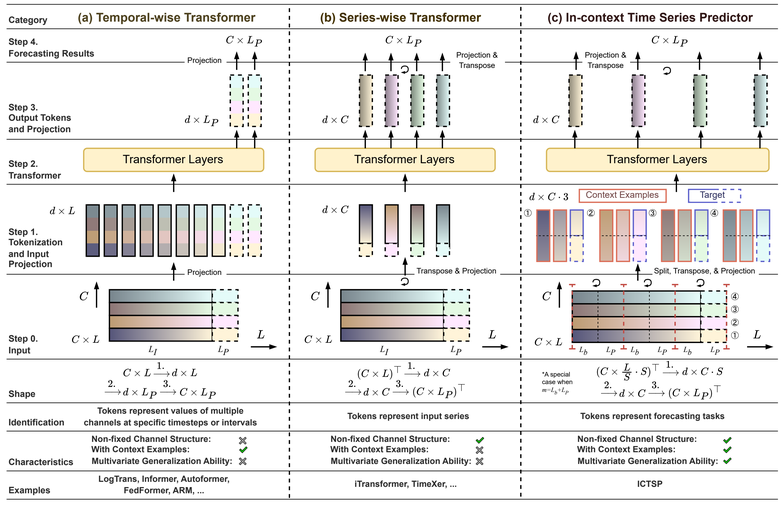

Figure 2. Architecture comparison among the three main TSF Transformer structures.

In-context Time Series Predictor (ICTSP) Uses forecasting tasks directly as tokens, constructing context examples from lookback windows:

$$\mathbf{Z}^{(L_b+L_P)\times(N+1)C} = [\mathbf{H}_1, \ldots, \mathbf{H}_C], \quad \text{with each} \quad \mathbf{H}_j = \begin{bmatrix} \mathbf{p}_{j,1} & X_j^{(1:L_b+L_P)} \\ \vdots & \vdots \\ \mathbf{p}_{j,N} & X_j^{(L_I-L_P-L_b+1:L_I)} \\ \mathbf{p}_{j,\text{target}} & [X_j^{(L_I-L_b+1:L_I)}, \mathbf{o}] \end{bmatrix}^\top.$$

ICTSP fully utilizes the Transformer's ICL capabilities, generalizes flexibly across datasets, and exhibits robust zero-shot learning performance.

Adaptive Model Reduction ICTSP adaptively reduces complexity according to dataset characteristics: (i) for weak temporal dependencies, it simplifies to linear or shallow MLP predictors via ICL; (ii) for unstable series patterns, it reduces to a Series-wise Transformer using only local lookbacks; (iii) if inter-series context is negligible, it simplifies to univariate MLP, effectively handling noisy data.

Solving Key Issues of TSF Transformers

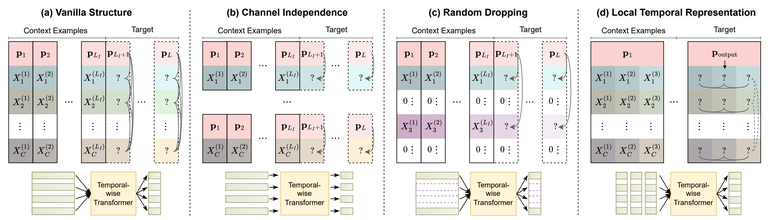

Figure 3. Previous solutions of Temporal-wise Transformers' overfitting issue from an ICL view.

Permutation Invariance Temporal-wise Transformers are negatively affected by permutation invariance, weakening positional embeddings when channel structures are stable. ICTSP circumvents this by representing tokens as independent forecasting tasks, making token order irrelevant and aligning effectively with Transformer characteristics.

Timestep Mixing and Overfitting Temporal-wise Transformers overfit by incorrectly emphasizing channel dependencies at single timesteps. Previous solutions partially address this by enhancing temporal focus but lack fundamental resolution. ICTSP fundamentally resolves this by using forecasting tasks as tokens with explicit context examples, naturally guiding the Transformer to model temporal and meaningful series relationships.

Channel Structure Restriction Temporal-wise Transformers require fixed input series structures, limiting adaptability. Series-wise Transformers offer flexibility but embed series-specific features, reducing transferability. ICTSP removes such restrictions entirely, using context examples and minimal embeddings, enabling strong zero-shot performance across diverse multivariate datasets.

Experiment Results

We evaluate ICTSP comprehensively under full-data, few-shot, and zero-shot settings on widely-used TSF benchmarks (ETTs, Traffic, Electricity, Weather). We compare ICTSP against state-of-the-art methods, including LLM-based (Time-LLM, GPT4TS, LLMTime), Temporal-wise Transformers (PatchTST, FEDformer, Autoformer, Informer), Series-wise Transformers (iTransformer), CNN-based (TimesNet), and simpler methods (DLinear, Last-value Repeat), reporting average test MSE across datasets.

Table 1. Full-data, few-shot, and zero-shot TSF results. Averaged test set MSE ranking on each dataset is reported. The best and second-best results are in bold and underlined, respectively.

Models	**ICTSP**	Time-LLM	GPT4TS	iTransformer	PatchTST	FEDformer	Autoformer	Informer	DLinear	Repeat
Rank (Full)	**1.36**	<u>1.93</u>	4.29	5.64	2.79	6.86	7.89	9.43	5.00	9.50
#Rank1	**18**	<u>8</u>	0	0	2	0	0	0	0	0
Models	**ICTSP**	Time-LLM	GPT4TS		PatchTST	FEDformer	Autoformer	Informer	TimesNet	DLinear	Repeat
Rank (10%)	**1.57**	<u>1.89</u>	3.21		4.07	6.39	7.46	9.43	6.68	4.93	9.21
#Rank1 (10%)	**15**	<u>10</u>	1		1	0	0	0	0	0	0
Rank (5%)	**1.56**	<u>2.36</u>	3.48		3.96	6.04	7.08	9.44	6.80	4.84	9.24
#Rank1 (5%)	**12**	<u>7</u>	1		2	0	0	0	0	4	0
Models	**ICTSP**	Time-LLM	LLMTime		GPT4TS	PatchTST	Autoformer	TimesNet	DLinear	Repeat	
Rank (0%)	**1.13**	<u>1.88</u>	8.67		4.54	3.29	6.21	5.75	5.33	7.88	
#Rank1	**21**	<u>3</u>	0		0	0	0	0	0	0	

Analysis of TSF Transformer Characteristics From the ICL view, Temporal-wise Transformers excel at modeling inter-series dependencies, while Series-wise Transformers focus on intra-series patterns. On the synthetic "Multi" dataset with strong inter-series shifts, the Temporal-wise Transformer performs well; Series-wise fails to capture the dependency, while ICTSP excels due to its shifting-context design. On real-world datasets with weak inter-series dependencies (e.g., ETTm2), ICTSP outperforms both, showing strong adaptability across different dependency structures.

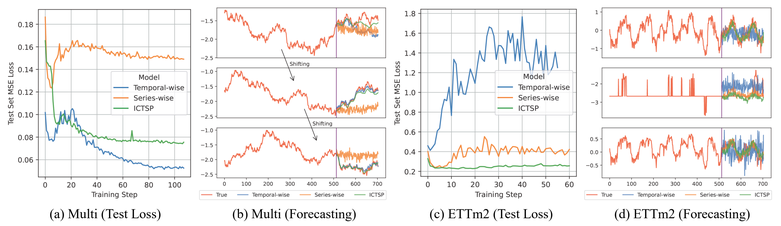

Figure 4. Comparison of the 3 architectures. The first 3 series of Multi and ETTm2 are visualized.

jlu414@gatech.edu | shihao.yang@isye.gatech.edu