ICML 2026 · Time Series Forecasting

Baguan-TS

A Sequence-Native In-Context Learning Model for Time Series Forecasting with Covariates

Linxiao Yang · Xue Jiang · Gezheng Xu · Tian Zhou · Min Yang · Zhaoyang Zhu
Linyuan Geng · Zhipeng Zeng · Qiming Chen · Xinyue Gu · Rong Jin · Liang Sun

DAMO Academy, Alibaba Group · Ant Group · Department of Computer Science, Western University

Presenter: Linxiao Yang

Motivation

Two existing paradigms; neither covers raw-sequence ICL with covariates

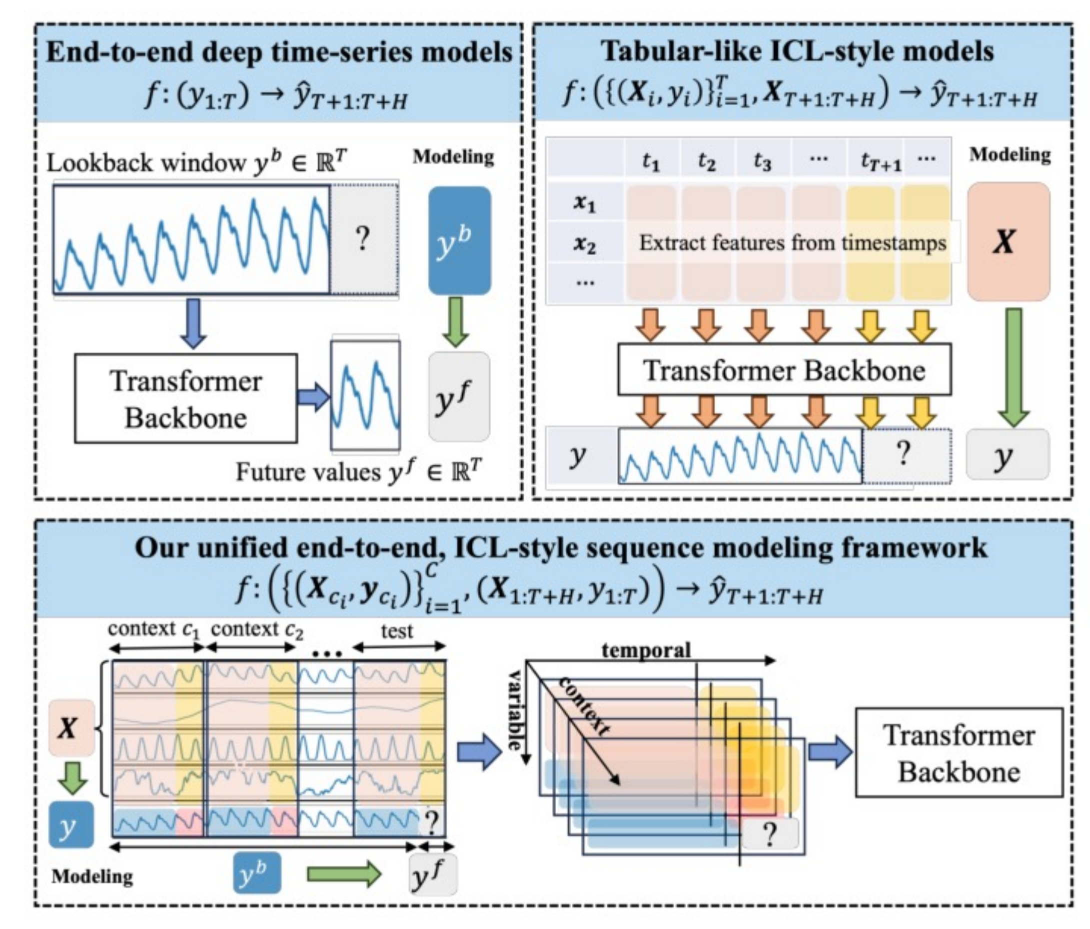

Fig. 1 — Three paradigms for time series forecasting.

- **End-to-end sequence models** — Learn directly from raw histories.
  No inference-time adaptation.
- **Tabular ICL (e.g., TabPFN-TS)** — Gradient-free adaptation.
  Needs hand-crafted features.
- **The gap** — No framework unifies raw-sequence learning with ICL.

Goal: sequence-native ICL on raw multivariate time series.

Contributions

One unified framework, three concrete pieces

01

3D Transformer

Sequence-native ICL with joint attention over
Temporal × Variable × Context.
Also supports a 2D inference mode.

02

Y-space RBfcst

Feature-agnostic, episode-local calibration via target-history retrieval — no hand-crafted features.

03

Context-Overfitting

Mitigates output oversmoothing in ICL by balancing denoising and selection.

SOTA on fev-bench-cov (30 covariate-aware tasks): highest win rate, −4.8 % SQL vs. TabPFN-TS.

Method 1 — 3D Transformer

Axis-factorized attention over (Context, Time, Variable)

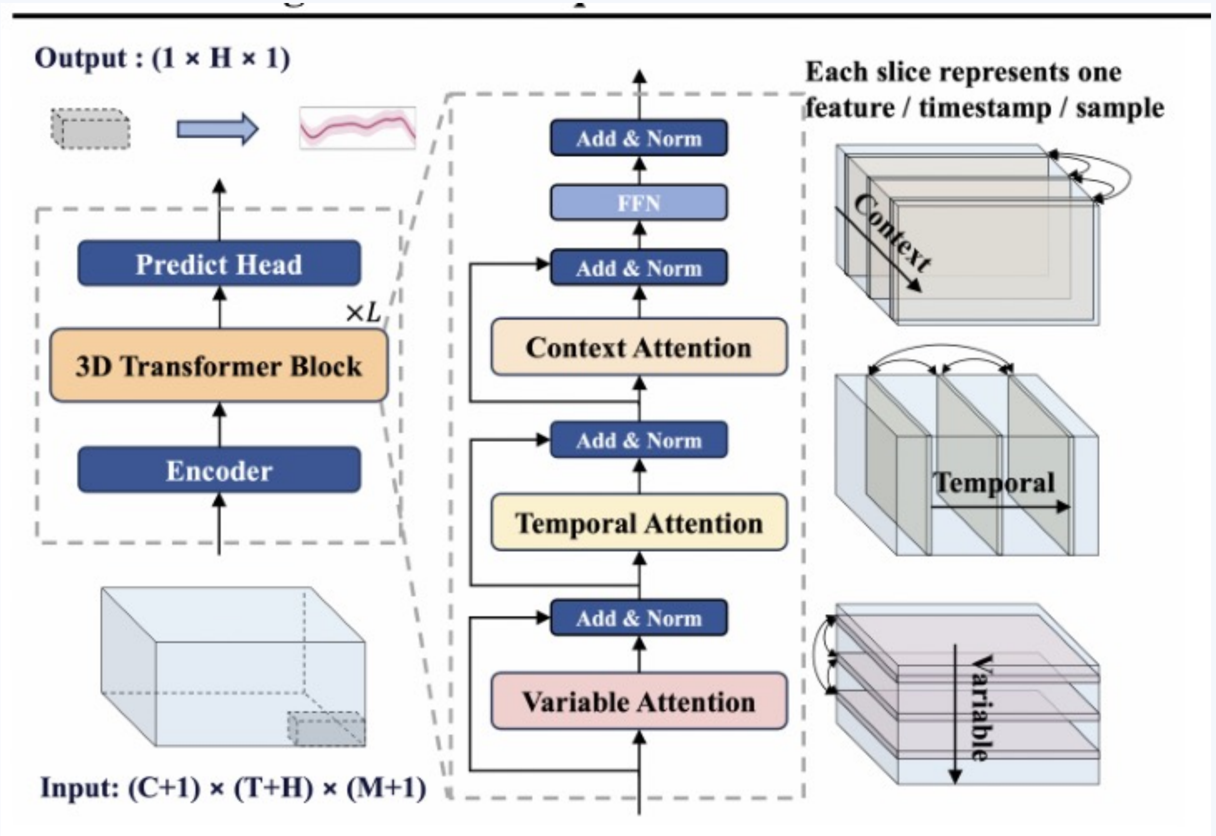

Fig. 3 — Overall architecture of Baguan-TS.

How it works

- **Input tensor** — $Y \in \mathbb{R}^{(C+1)\times(T+H)\times(M+1)}$; patch + random Fourier features.
- **3D block — temporal attn** — MHA with RoPE for relative time and periodicity.
- **3D block — variable attn** — MHA with learnable variable embeddings.
- **3D block — context attn** — MHA across episodes; no positional encoding.

Why this design

- **Avoids prohibitive cost** — Full 3D attention is $O((CSM)^2)$ — infeasible. Factorization is the only way to train this at scale.
- **Keeps the 3D structure as a prior** — Flattening loses the (time, variable, context) semantics; axis-wise attention preserves it.
- **Same model serves 2D & 3D** — Setting S=1 yields a 2D mode for free — useful when history is unreliable.

Method 2 — Y-space Retrieval-Based Forecasting

How we build each episode's context — and why target-space wins

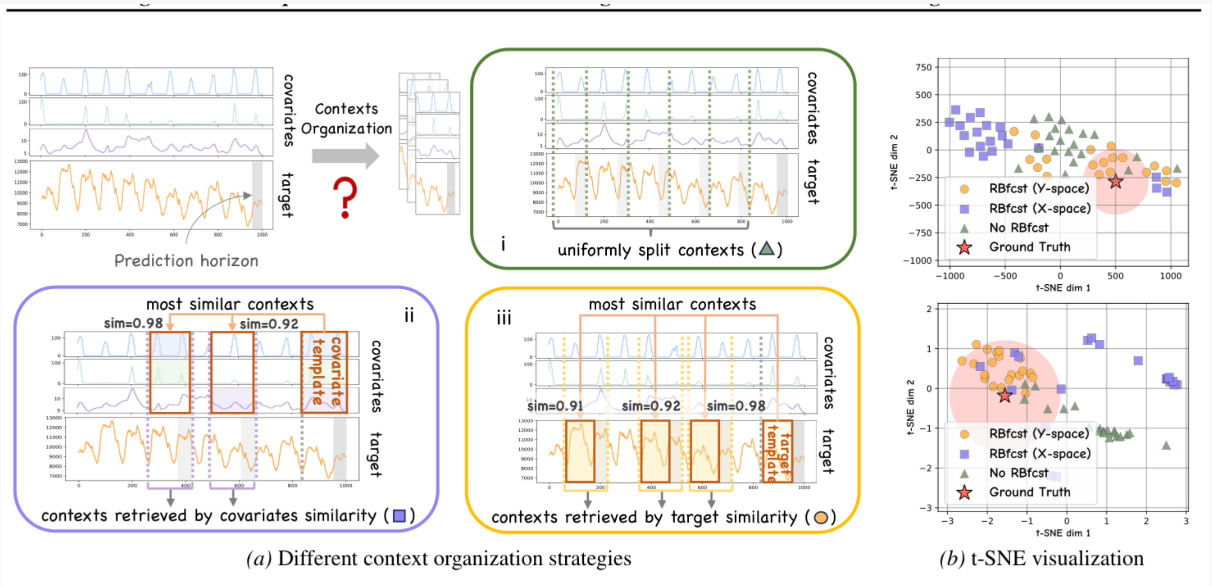

Fig. 4 — Context organization & t-SNE of prediction horizons.

How it works

- **Query** — Use the target lookback y^b as the query.
- **Nearest-neighbor lookup** — Retrieve K_ctx most-similar T-length windows from history.
- **Form episode** — Each retrieved window + its future = one context slice.
- **Training-time noise** — Retrieve 2K_ctx then random-sample K_ctx → robust to imperfect retrieval.

Why this design

- **Targets actually capture the task** — Covariates alone (X-space) miss the regime; the target trajectory summarizes all drivers.
- **Pulls episodes toward the true horizon** — t-SNE shows Y-space retrievals cluster around the ground-truth horizon — local & informative.
- **Removes the feature-engineering bottleneck** — Works across domains without designing X-space distances per task.

Method 3 — Context-Overfitting Strategy

How we stop ICL from oversmoothing periodic spikes

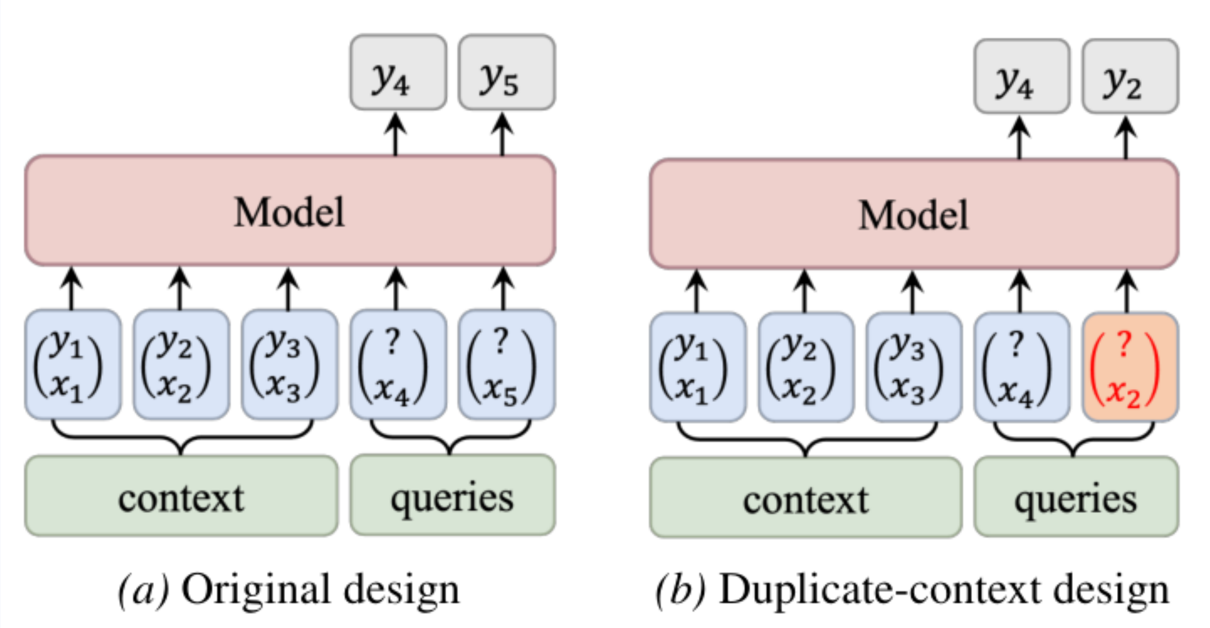

Fig. 5 — Original design vs. duplicate-context design.

How it works

- **Duplicate-context query** — Copy a short motif from one context slice into the query.
- **Self-retrieval objective** — Train the model to identify the matching context and reconstruct its target.
- **Encourage template matching** — Concentrates attention on the few examples that actually match.

Why this design

- **Forces the model to USE the context** — Plain ICL ignores spike templates in context and outputs smooth trends; duplication makes context-use unavoidable.
- **Restores periodic spikes** — Without it, the model treats spikes as noise; with it, identical patterns in context get faithfully reproduced.
- **Sharpens attention on what matters** — Drives attention from diffuse averaging to precise template matching — exactly what ICL should be doing.

Main Results — fev-bench-cov

30 covariate-aware tasks; SOTA on average SQL & MASE

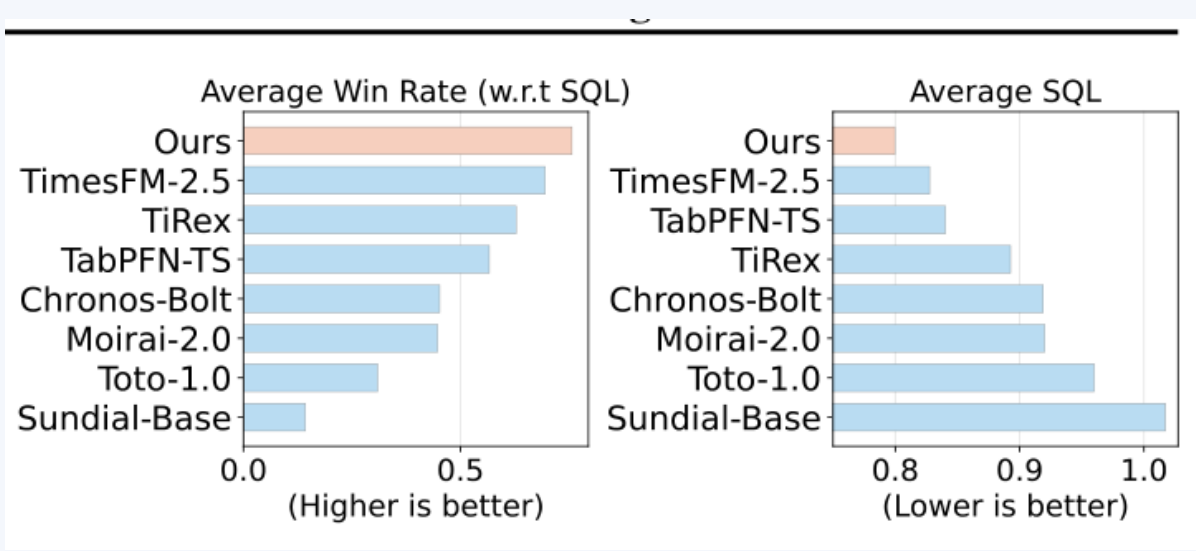

Fig. 2 — Average win rate & SQL across 8 baselines.

Model	SQL	MASE	WAPE	WQL
Chronos-Bolt	0.9190	1.1233	0.2702	0.2181
Moirai-2.0	0.9202	1.1233	0.2750	0.2226
Sundial-Base	1.0172	1.1449	0.2813	0.2498
TimesFM-2.5	0.8276	1.0122	0.2592	0.2095
TabPFN-TS	0.8404	1.0430	0.2015	0.1643
TiRex	0.8927	1.1081	0.2753	0.2233
Toto-1.0	0.9600	1.1755	0.2737	0.2248
Baguan-TS (ours)	**0.7997**	**0.9857**	**0.2173**	**0.1724**

- **Best SQL & MASE** — among 8 strong TS foundation models.
- **Highest win rate (SQL)** — across 30 covariate-aware tasks.
- **−4.8% SQL vs. TabPFN-TS** — the only covariate-aware baseline.

Ablations & Robustness

What actually matters — and how it holds up under noise

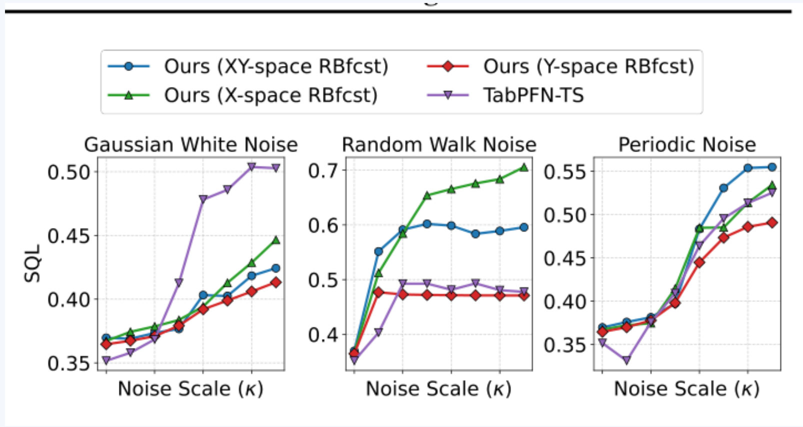

Fig. 8 — Robustness to input noise (Gaussian / random walk / periodic).

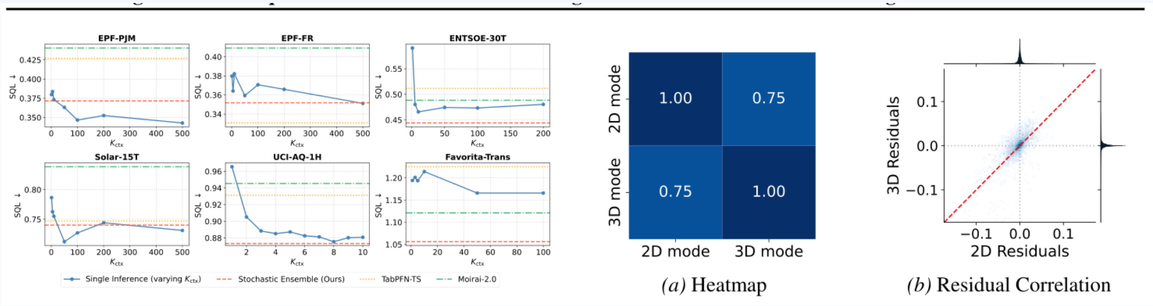

Fig. 9 — Sensitivity to context size K_ctx.

RBfcst

Y > XY > X > none.
Largest gain on entsoe-1H.

Noisy contexts

Stable across 0–83 % noise.
Ensemble further smooths.

K_ctx

Non-monotonic; default 50.
Too few → unstable; too many → noisy.

2D ⊕ 3D ensemble

Complementary residuals.
Improves SQL & calibration.

Conclusion

Baguan-TS unifies raw-sequence representation learning with in-context learning for time series.

- **Sequence-native ICL** — 3D Transformer attends jointly over time, variables, and context.
- **Feature-agnostic calibration** — Y-space RBfcst provides local, episode-specific correction.
- **Stable ICL training** — Context-overfitting strategy prevents output oversmoothing.
- **Strong empirical evidence** — SOTA on fev-bench-cov and 27 real-world energy/retail datasets.

Takeaway: ICL can be sequence-native — no hand-crafted features required.

Thank you

Questions & Discussion

Linxiao Yang · yanglinxiao0929@hotmail.com

Baguan-TS · ICML 2026