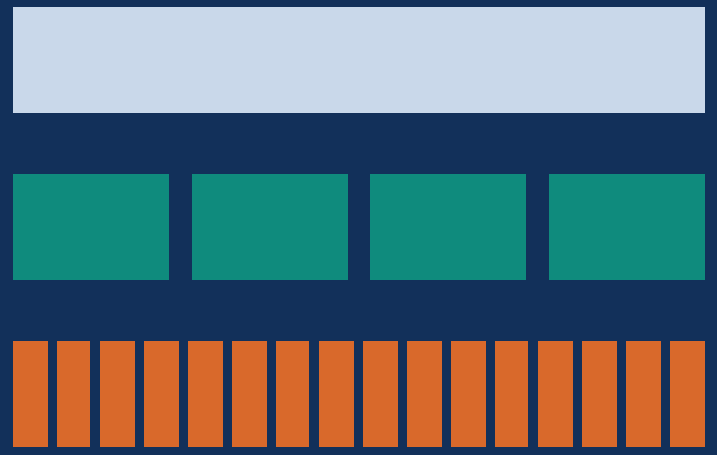
NEURIPS 2026

FracTS

Hierarchical and Autoregressive Time Series Generation

Jiayu Li*, Umair Afzal*, Zilong Zhao,
Milad Abdollahzadeh, Uzair Javaid, Biplab Sikdar

Betterdata AI · University of Illinois Urbana-Champaign · National University of Singapore



coarse → fine

* Equal contribution

Why time series generation is still hard

Step-by-step autoregression fits time series, but it…

Lacks diversity

Near-deterministic next-step prediction gives low-variance samples

Accumulates error

Small per-step mistakes compound over long horizons

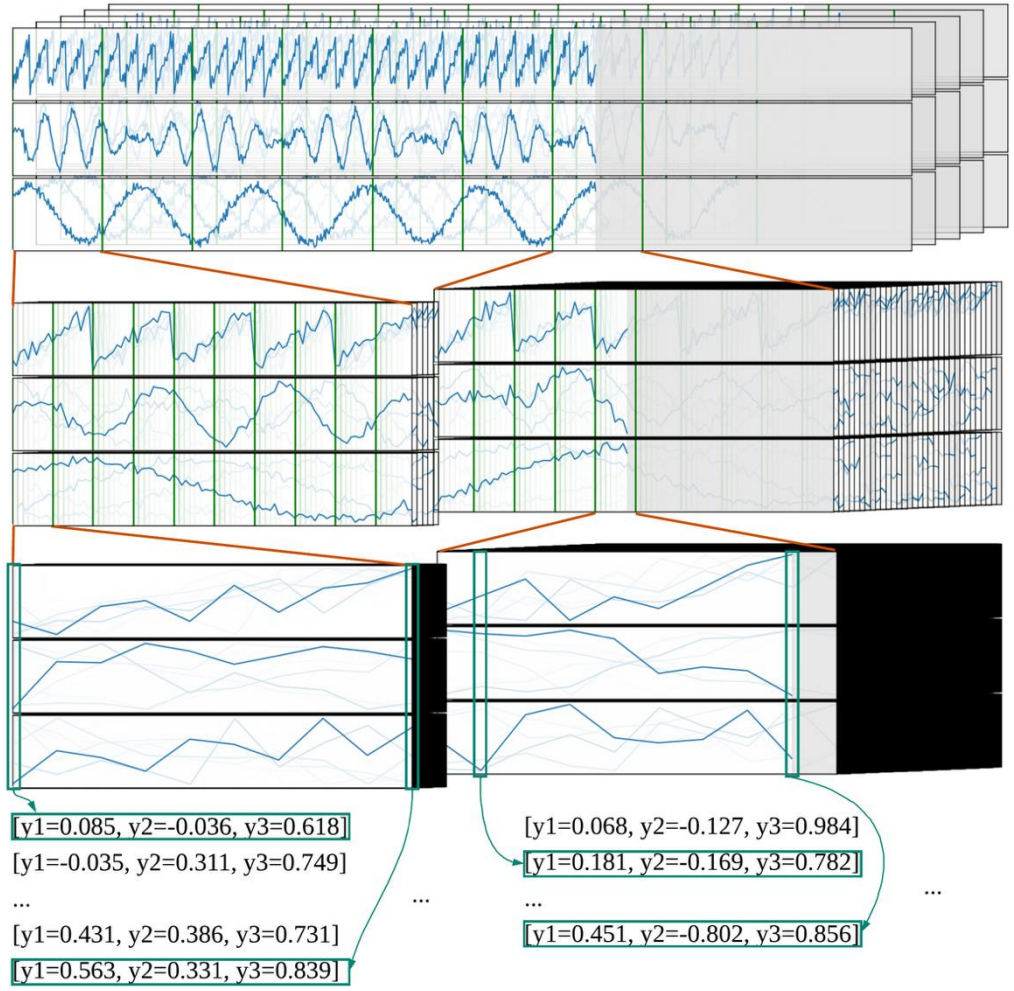
Misses global structure

Long-range dependencies and the overall trajectory get lost

…and real data breaks common assumptions



Key idea: time series are fractal



A long series splits into shorter series, level by level

1 **Coarse → fine autoregression**

Each level's transformer generates patches in order; the next level expands every patch (patch size 20, GPT-2 small per level).

2 **Last level = the variables**

At length 1, generate the M variables of one time step, like colour channels of a pixel.

3 **Global skip connections**

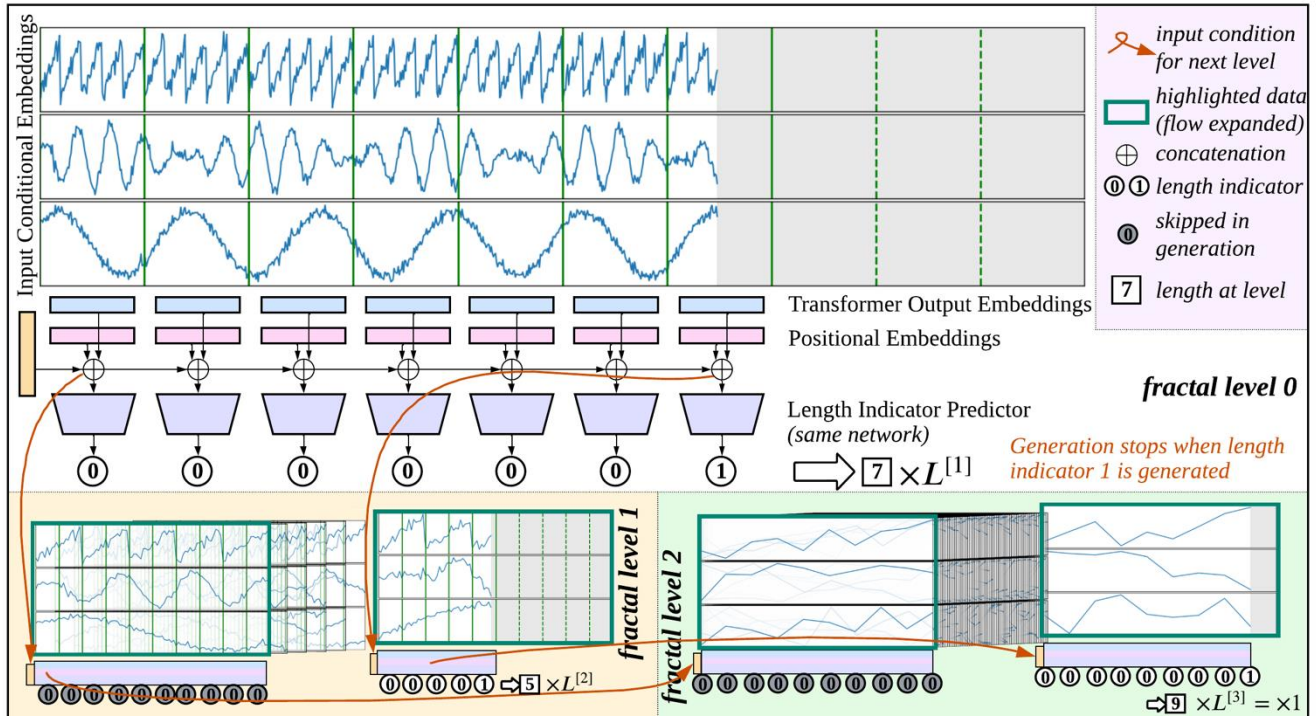
Pass the global condition straight down to deeper levels so long-range context isn't lost.

Error scale (Thm 4.1)

Flat AR: $\Omega(\delta \cdot L)$ FracTS: $\Omega(\delta \cdot I \cdot K)$

e.g. L = 8,000, I = 20, K = 3 → I·K = 60. Memory O(BL(I+d)) (Thm 4.2)

Built for messy, real-world data



Each level predicts an end-of-sequence flag; generation stops at the first 1.

One numeric value becomes: Δ prev. step · K-means bin · raw value

Variable length

EOS-style length indicator; empty padded patches are skipped in training.

Mixed types

Categoricals are embedded; numerics get a sampled bin plus the exact value, adding diversity.

Trends

Difference to the previous step is added as an auxiliary feature.

Static conditions

Attributes plus summary statistics, compressed by a Perceiver-style module.

Large gains, especially in downstream utility

≈100%

lower predictive-utility error

unconditional

70%

lower correlation error

unconditional

75–81%

lower predictive-utility error

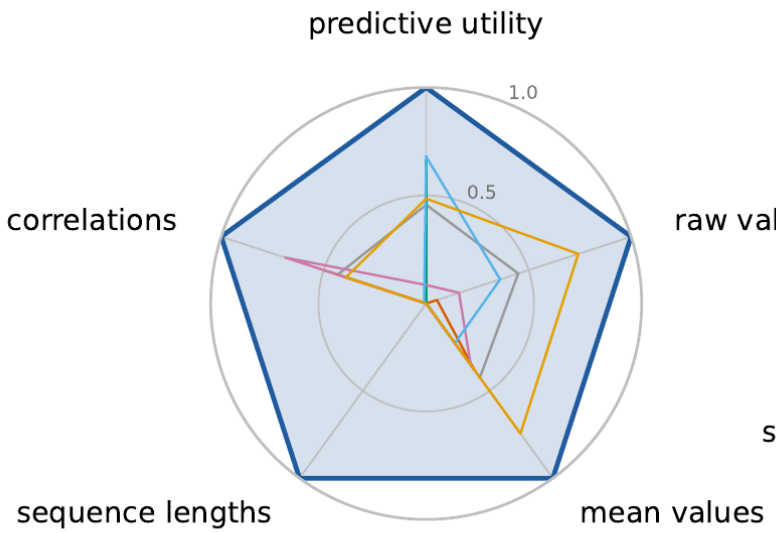
conditional

38%

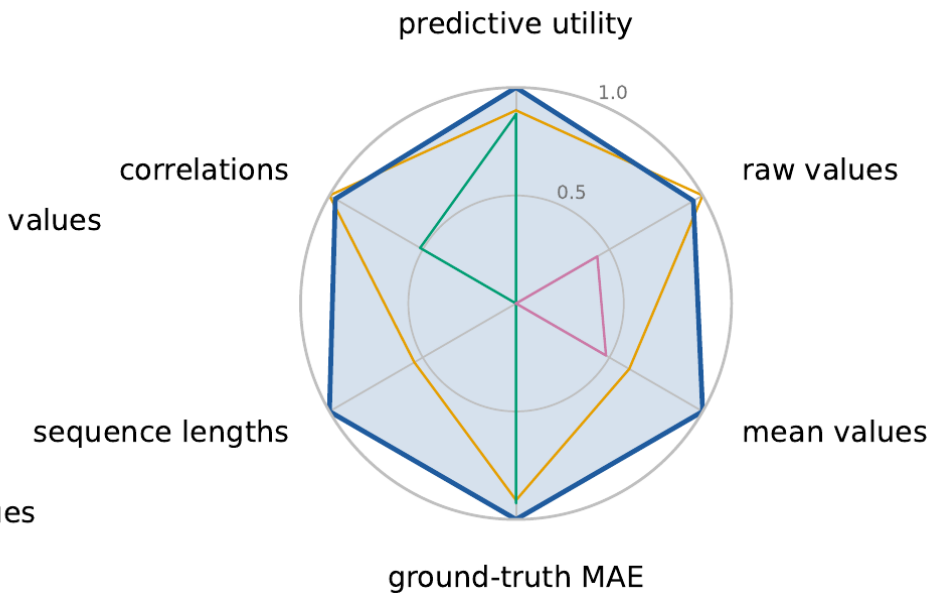
lower error vs. ground truth

simulated, conditional

Unconditional · 3 datasets · 6 baselines



Conditional · 2 datasets · 3 baselines



Bold blue = FracTS

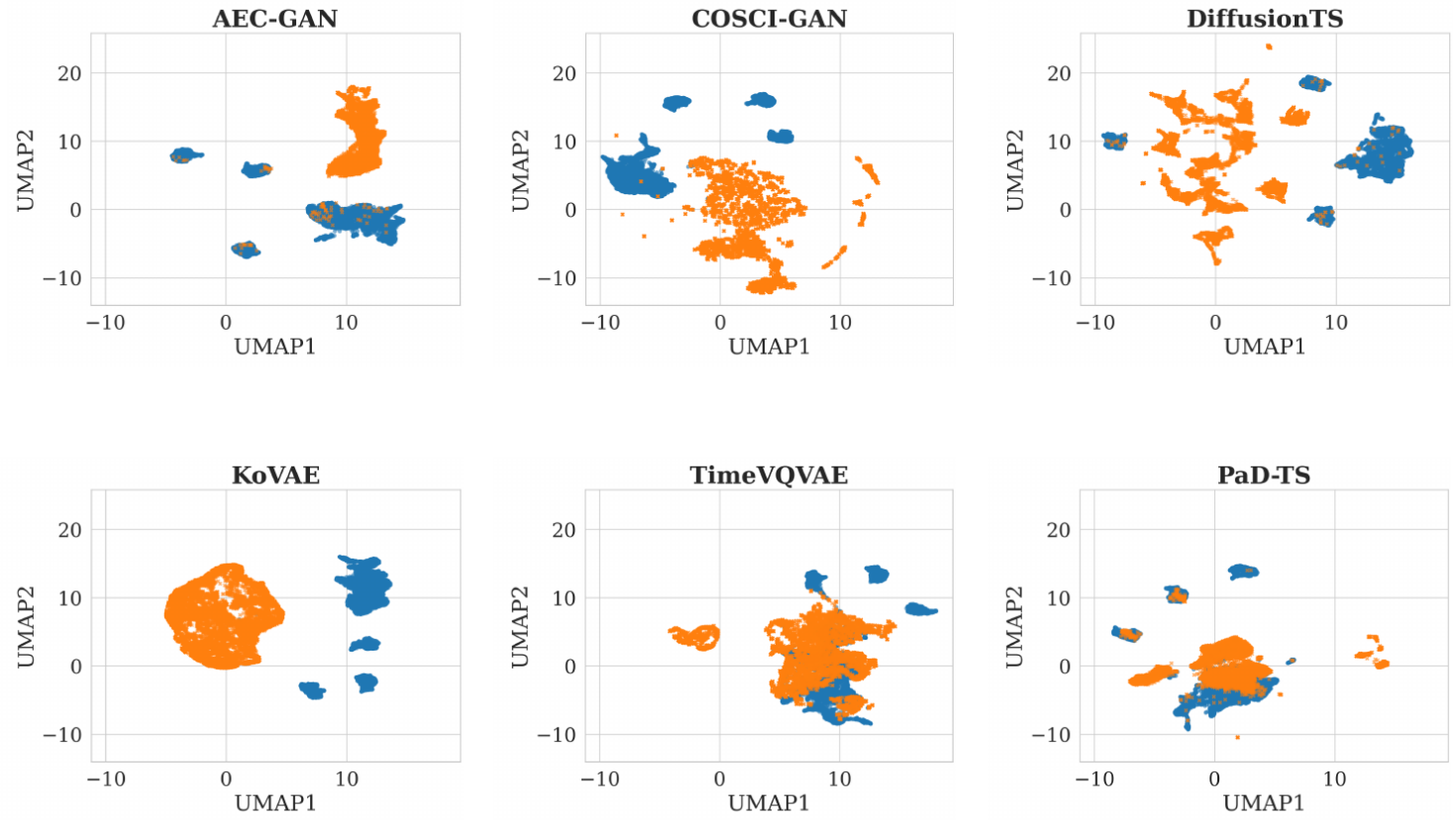
Thin lines = baselines

Scores rescaled per axis; larger is better

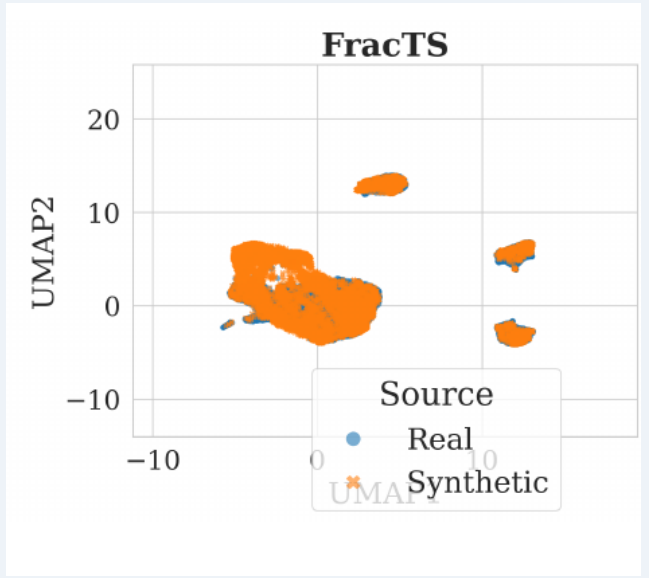
Relative to the best baseline, on dataset-averaged scores

Synthetic samples cover the real distribution

Baselines



FracTS



Blue = real, orange = synthetic (Weather dataset).

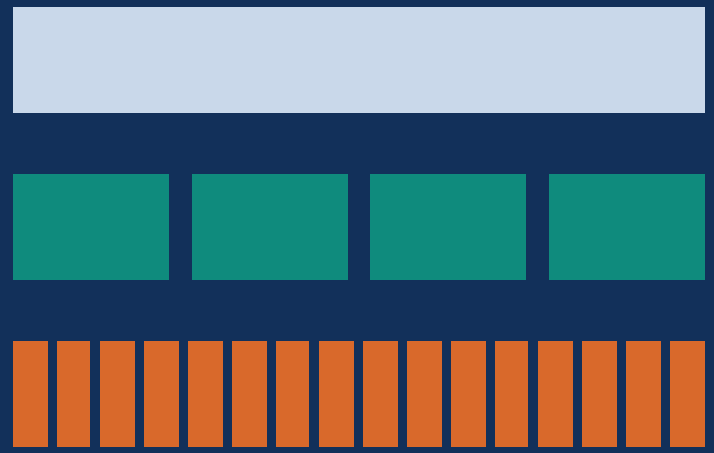
Baselines collapse into one blob or miss clusters; FracTS lands on all of them.

Takeaways

1. **Time series are fractal**
   FracTS generates them coarse to fine with hierarchical autoregression.

2. **Real data, natively**
   Variable lengths, mixed types and static conditions, with no fixed-length input.

3. **Big gains where it matters**
   Best on all unconditional metrics; predictive-utility error cut by 75–100%.



Thank you!

github.com/BetterdataLabs/fracts

jiayul11@illinois.edu · zilong@betterdata.ai