TABULAR FOUNDATION MODELS

For Manufacturing

Domain-agnostic in-context learning for manufacturing

Aditya Balu
Data Scientist
Translational AI Center, Iowa State University

What is a Tabular Foundation Model?

The traditional ML loop:

Each new dataset gets a fresh model. Train, tune hyperparameters, validate, deploy. Repeat for every task. Hours to days per dataset.

The foundation model loop:

Pre-train ONCE on millions of synthetic tabular datasets. At inference time, the model conditions on your training rows and predicts test rows in a single forward pass. No fitting, no tuning. Seconds, not hours.

This is in-context learning, the same paradigm GPT uses for text - now applied to tables.

Traditional ML

- Train from scratch per dataset
- Hyperparameter tuning required
- Compute heavy: 5 to 60 minutes
- Bias designed by humans (tree splits, linearity, etc.)

Tabular Foundation Model

- Pre-trained once on synthetic tasks
- Zero hyperparameters at inference
- Inference in seconds via single forward pass
- Bias learned from data (data-specific inductive bias)
- **Works zero-shot across many domains**

Why It Matters: Five Threads

Themes that recur across every result we will see

01

Foundation Models

One pre-trained model, many tasks. The text-to-tables transfer of an entire paradigm.

02

Low-Data Regression

Thrives where deep nets fail: tens to thousands of rows, not millions.

03

Data Imputation Impact

Native tolerance to missing values. No imputation pipeline needed.

04

Domain-Agnostic

Same model, no architectural changes - works on machining, agriculture, biology, finance.

05

Future of Tabular Intelligence

Industry now investing heavily: Kumo, Amazon Mitra, Prior Labs, Fundamental.

These five threads tie together everything in this deck.

TabPFNv2 in One Slide

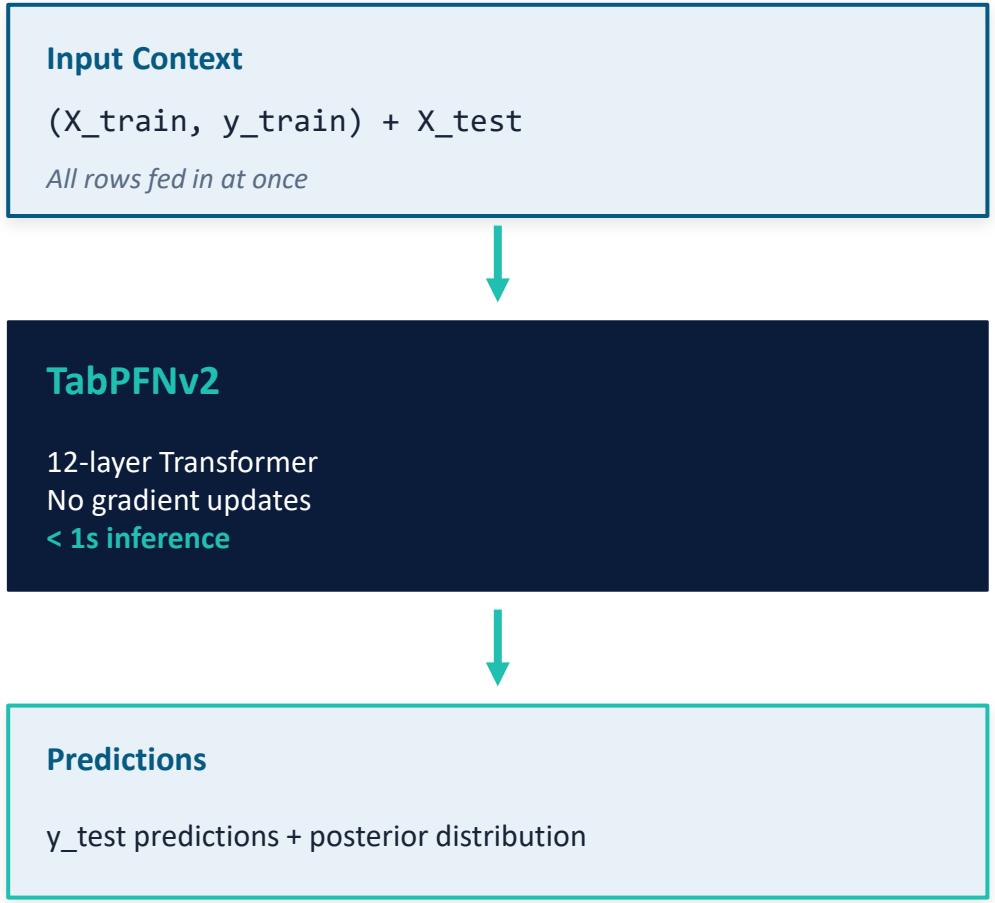
The model behind every result that follows

Architecture

- 12-layer transformer encoder
- Self-attention across rows (samples), not columns
- Train and test rows processed in a single forward pass
- Context window: up to 10,000 rows, 500 features

How it was trained

- Pre-trained on millions of synthetic datasets generated from structural causal models
- Each synthetic task: random DAG -> sample features and labels -> present as in-context task
- Result: a general-purpose tabular learner baked into the weights



Hollmann et al., Nature 2025

SECTION 2

Manufacturing Machining Data

Surface roughness prediction in CNC turning

Manufacturing: Turning Dataset

CNC surface roughness prediction - 28 samples, 4 process parameters

About the Data

A classic CNC turning experimental dataset. 28 single-pass turning experiments measuring surface roughness under varied process parameters.

Inputs (4 process parameters)

- v - cutting velocity (m/min)
- f - feed rate (mm/rev)
- d - depth of cut (mm)
- r - nose radius (mm)

Target: surface roughness Ra (nm)

R^2 score

0.938

with just 22 training rows

RMSE: 0.29 um
MAE: 0.21 um
MAPE: 8.8%

Robustness check

0.891

R^2 at 14/14 split (50/50)

Holds up even when half the data is held out - the prior is doing real work, not memorization.

Why this is the strongest low-data argument

Most regressors collapse below 50 samples.
TabPFN sees 22 rows of (v, f, d, r) -> Ra and predicts the held-out 6 rows with 8.8% MAPE.

Zero feature engineering. Zero hyperparameter tuning. The synthetic prior already knows what physics-style tabular regression looks like.

SECTION 3

Agricultural Yield Prediction

Three datasets, three regimes - published at AAAI 2026 AgriAI Workshop

Three Datasets, Three Regimes

Each one stresses TabPFN in a different way

US Soybean

86,101

samples

LARGE / COMPLETE

Daily weather aggregated to 61 statistical features. No missing values. Single crop, single country, deep history.

Global Multi-Crop

28,242

samples

DIVERSE / COMPLETE

Annual climate + management indicators across 101 countries and 13 crops. Categorical-heavy, very heterogeneous.

EU-27 Regional

8,656

samples

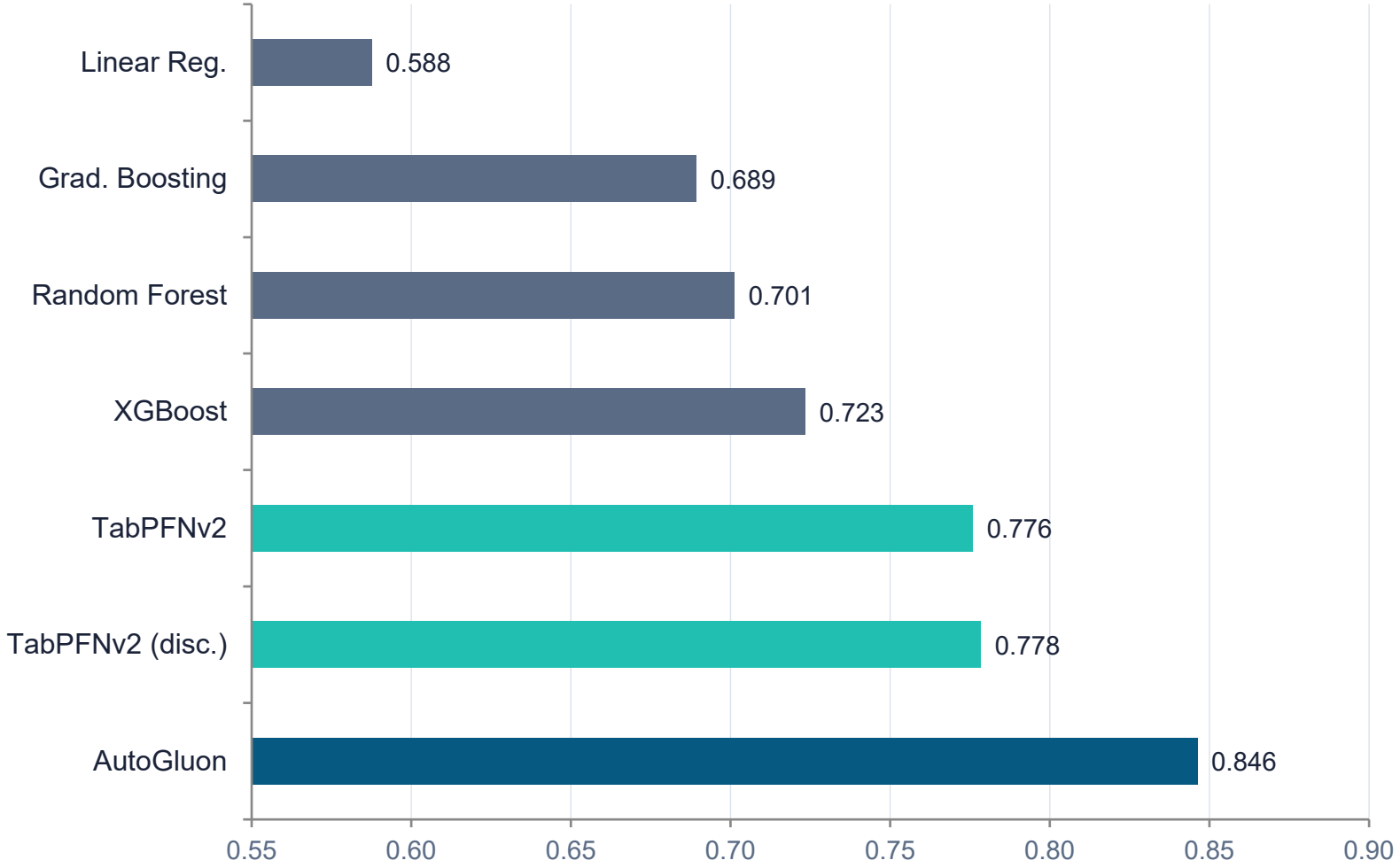
SMALL / MISSING

20 crops across 27 EU countries at the NUTS-2 sub-national level. 5-13% missing values per feature.

Source: Khaki & Wang (2019), FAOSTAT (2023), Eurostat NUTS-2

US Soybean (86K samples) - AutoGluon Wins

Where lots of clean data is available, deep ensembling pays off



Read it like this

AutoGluon takes the win

- **AutoGluon: R^2 = 0.846**
- TabPFNv2: R^2 = 0.776 - 0.778
- Gap: 8.8 percentage points

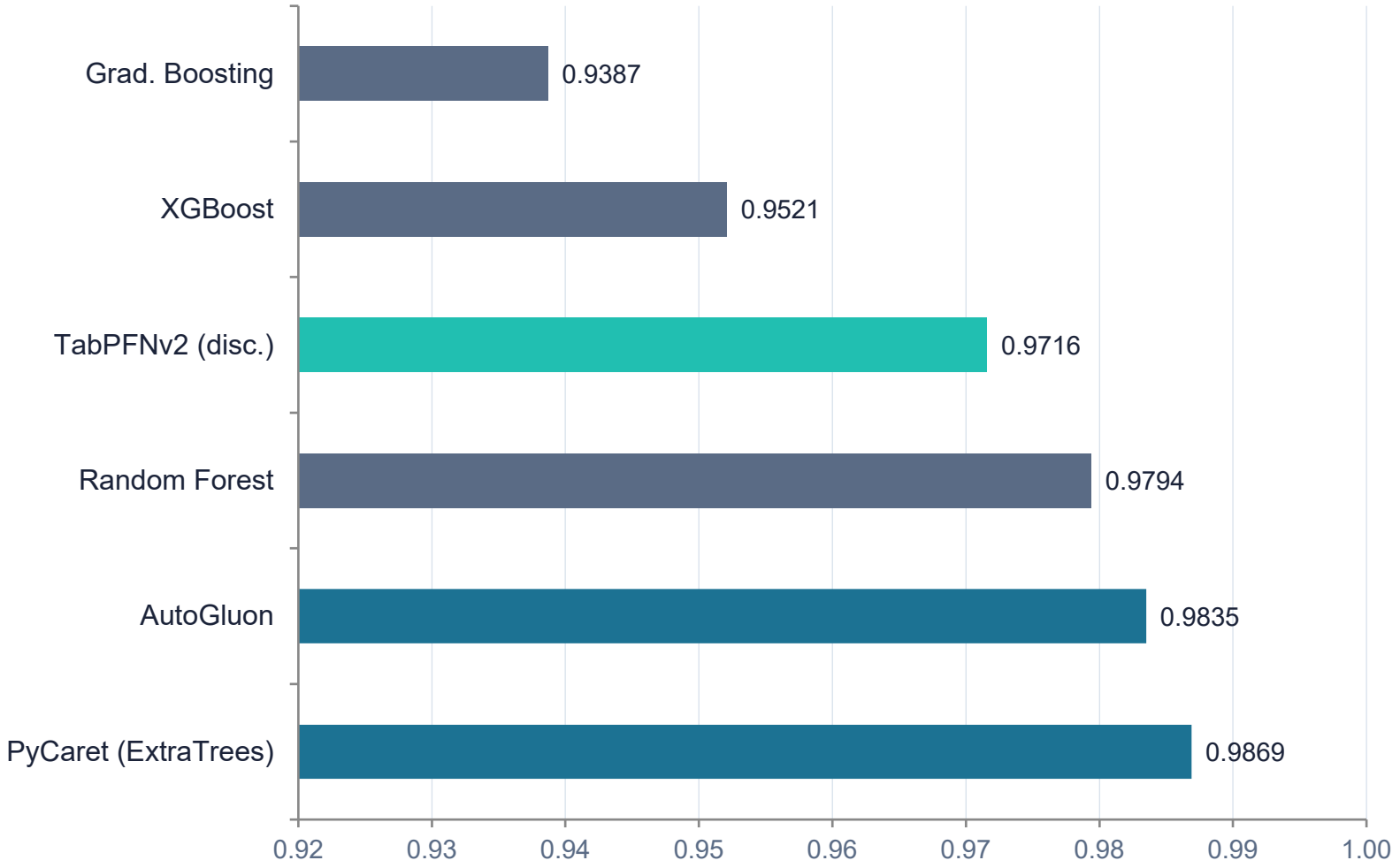
Why?

AutoGluon uses ALL 86K rows for deep ensembling. TabPFN's 10K context window leaves 88% of the data on the table.

Top features: temp variability (0.31), precipitation (0.24), latitude (0.18)

Global Multi-Crop (28K) - PyCaret Edges Out

Diverse but complete data: tree ensembles still excel



Tight race

All advanced methods clear R^2 = 0.97

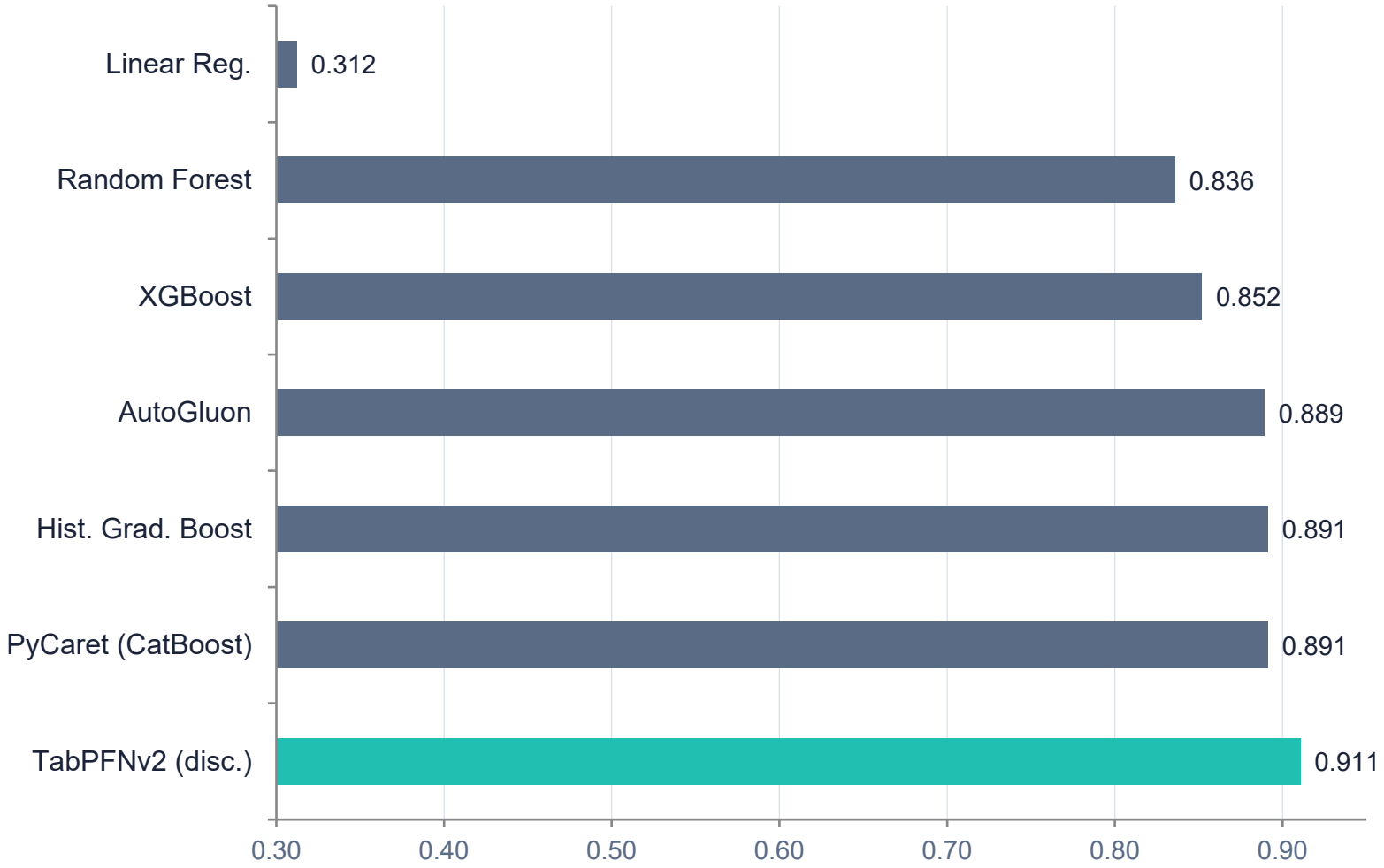
TabPFN sees only 35% of the data (10K context limit) and still posts R^2 = 0.972 - within 1.5 points of the leader.

Compute side

- PyCaret: 21.8s
- AutoGluon: 118.3s
- **TabPFN: < 1s**

EU-27 Regional (8.6K, 5-13% missing) - TabPFN Wins

Section 3 / The headline result of the AAAI paper



+2.18

percentage points R^2

TabPFN over the next-best method (PyCaret + CatBoost)

Why it wins here

- Full 8.6K dataset fits the 10K context window
- Native missing-value handling (no imputation pipeline)
- Pre-training on synthetic missing-data patterns transfers directly

This is the data-imputation impact thread.

Cross-Dataset View: When to Use What

The decision rule from three regimes of agricultural data

Large + Complete

USE

AutoML (AutoGluon)

Soybean: $R^2 = 0.846$

Deep ensembling exploits all available rows

Diverse + Complete

USE

AutoML or TFM

Global: $R^2 \approx 0.97$ (both)

Both viable - choose by speed (TFM 100x faster)

Small + Missing

USE

Tabular Foundation Model

EU-27: $R^2 = 0.911$

Native missingness + small data = TFM territory

The bigger picture: tabular foundation models do not replace AutoML. They define a new dominant region - small, messy, time-pressed problems where they win on both accuracy and inference speed.

SECTION 4

Vehicle Sensor Data

Sensor signal regression on vehicle data

Fast-TrAC Vehicle Sensor Dataset

Vehicle telemetry- sensor signal regression

What we predict

An internal sensor measurement (GroundTruth) from vehicle telemetry. Each row is a sensor reading aggregated to a vehicle unit.

Features (8 inputs)

Eight on-vehicle sensor signals captured during operation. Aggregated per UnitID into a tabular regression problem.

Why this dataset is interesting

Heterogeneous vehicle units, real sensor noise, no obvious linear structure - a real industrial regression problem where tree ensembles have historically dominated.

Dataset at a Glance

68,229

raw rows of sensor data

1,339

unique vehicle UnitIDs (after aggregation)

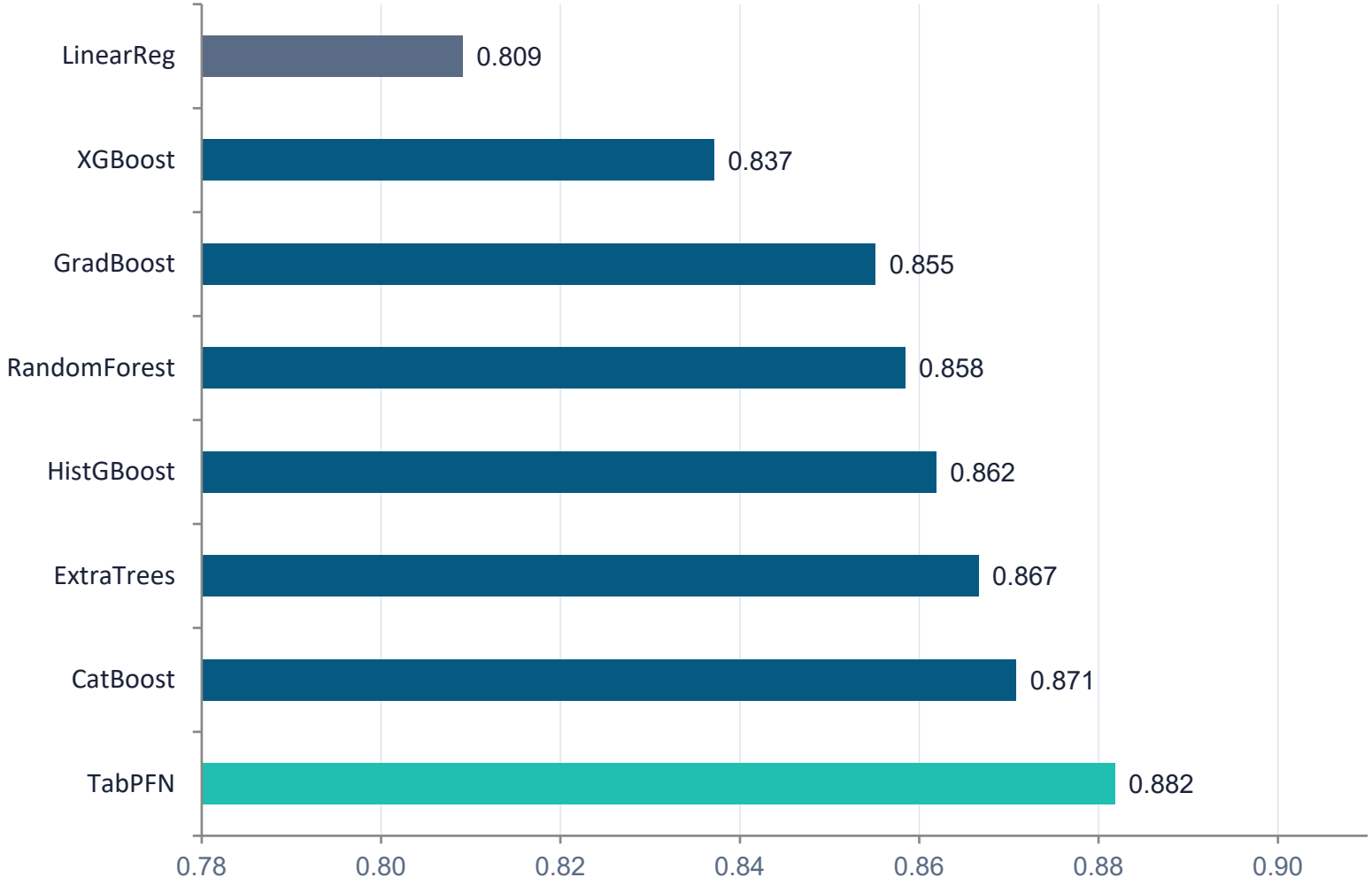
8

sensor input features per unit

Target: GroundTruth sensor measurement

TabPFN vs Classical ML on Fast-TrAC

50/50 split, R^2 score on Vehicle sensor data



0.8818

TabPFN R^2

Beats CatBoost (0.871) and every gradient boosting method

Takeaway

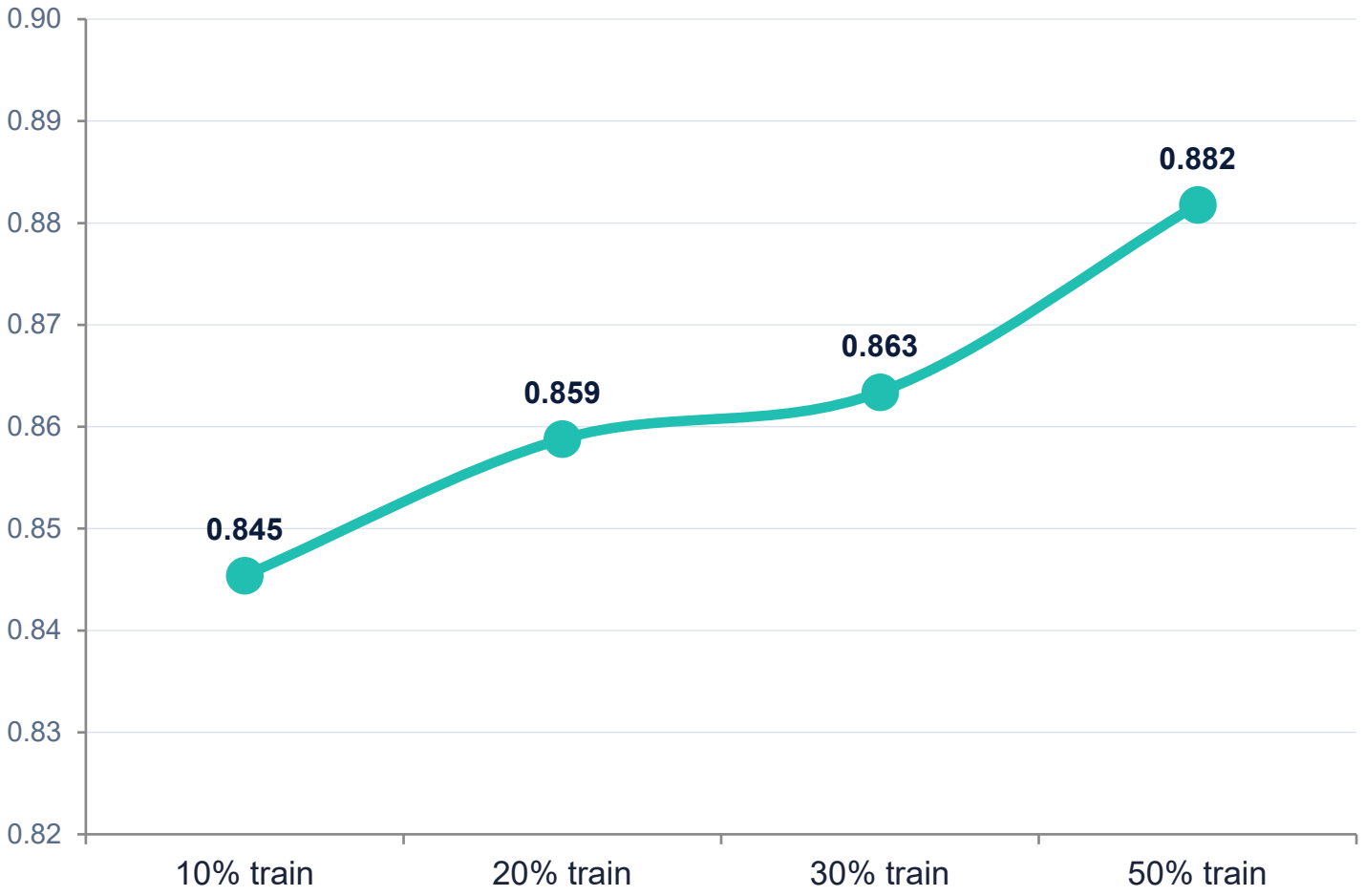
Zero hyperparameter tuning. Zero feature engineering. One forward pass.

TabPFN edges out every classical method on a real vehicle sensor dataset, including the specialist gradient boosting libraries.

RMSE: 7.09 | MAE: 4.68 | MAPE: 17.4%

Low-Data Regression: TabPFN Holds Up

Train fraction vs R^2 on Fast-TrAC vehicle sensor data



The headline

With only 10% training data, TabPFN still hits R^2 = 0.845.

Why this matters

Vehicle sensor datasets are often small per-unit. Sensors are expensive, on-vehicle experiments are slow, and labelled examples are scarce.

Classical models need lots of data plus tuning to compete. TabPFN gets there from the prior alone.

From 10% to 50% data, TabPFN gains only 4 R^2 points - the foundation already does most of the work.

Industry Is Moving Fast

The space of tabular foundation models in 2025-2026

Kumo.ai

KumoRFM (Relational Foundation Model)

- First foundation model for relational, multi-table data
- Built on a Relational Graph Transformer
- Outperforms feature engineering by 2-8% across 30 tasks (RelBench)
- Used at DoorDash, Snowflake, Reddit

Amazon AWS

Mitra + Fundamental NEXUS partnership

- Mitra: AWS' own TFM, shipped inside AutoGluon 1.4
- Pretrained on a curated mixture of synthetic priors
- SOTA on TabRepo, TabZilla, TabArena vs TabPFNv2 and TabICL
- Strategic AWS partnership with Fundamental (NEXUS) for enterprise tabular AI

Other major players

The space is crowded and growing

- Prior Labs (TabPFNv2 / TabPFN-2.5) - Germany
- Layer6 AI (TabDPT) - Canada
- Fundamental (NEXUS, $255M Series A) - USA
- NeuralkAI, ContextTab (SAP), LimiX, Orion - and counting

THE FUTURE

Tabular Intelligence Is the Next Wave

One paradigm, many domains

Same TabPFN, no tuning - wins on Alahmari machining, EU-27 agriculture, and vehicle sensors. The domain-agnostic claim holds in practice.

Low-data is the new high-data

From 10% training data on Fast-TrAC to 8.6K rows on EU-27, TabPFN is dominant precisely where deep nets and big AutoML pipelines stumble.

Missing values become a feature, not a bug

Native imputation handling lets TabPFN turn the messiest real-world data into a competitive advantage.

Industry is investing - hard

Kumo, Amazon Quick, Prior Labs, Fundamental ($255M). Tables are the last untouched modality, and the race is on.