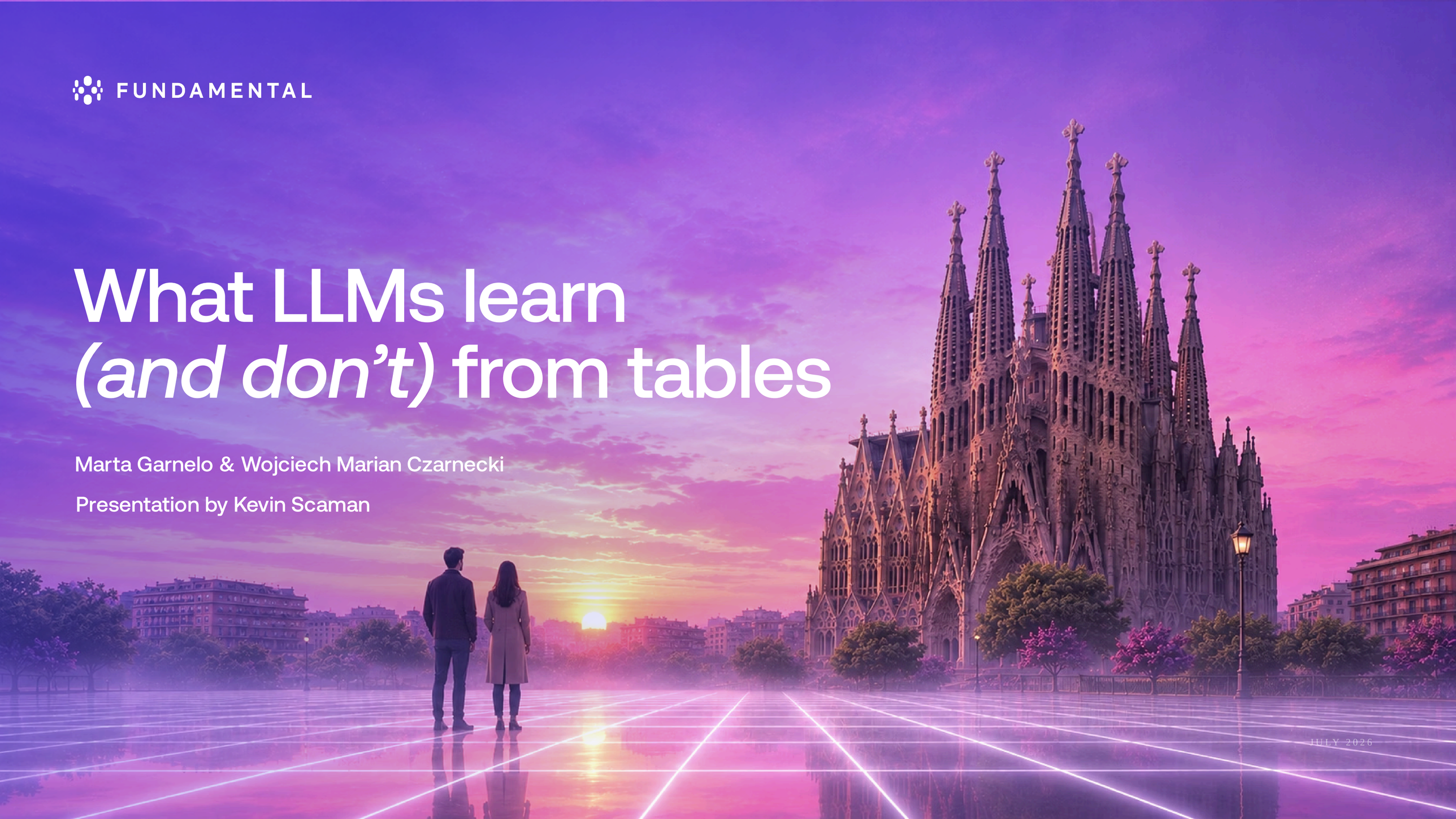
FUNDAMENTAL

What LLMs learn *(and don't)* from tables

Marta Garnelo & Wojciech Marian Czarnecki

Presentation by Kevin Scaman



JULY 2026

Should we *fear* general-purpose LLMs?

In a few months, will a prediction problem just be a matter of asking a frontier LLM, ***politely***, for the answer?

They took over writing, code, and agents. Tabular prediction runs finance, healthcare, and operations. The premise of our whole field is that it needs its own models. Has anyone checked?

One prompt, context in, predictions out.

THE PROMPT · VERBATIM

"Predict values of each element in x_test, based on labeled data provided in x_train and y_train. Do not include any reasoning, comments or anything. Output ONLY a list of integers (0 or 1) of length exactly 30."

Model	*A frontier LLM (Claude Opus 4.6), queried directly*
In context	*120 labeled rows as CSV text*
To predict	*30 unseen rows*
Repetitions	*30 runs, accuracy logged*
Baselines	*Random forest (depth 4) · 1-NN · logistic regression*

No tools, no code execution, no agent. The purest inference regime. Anything layered on top measures the harness; we want to measure *the model.*

Let's start easy with *Iris*.

Iris is arguably the simplest real open-source dataset there is: three species of flower, four measurements per row.

Rows	150
Columns	4
Classes	3

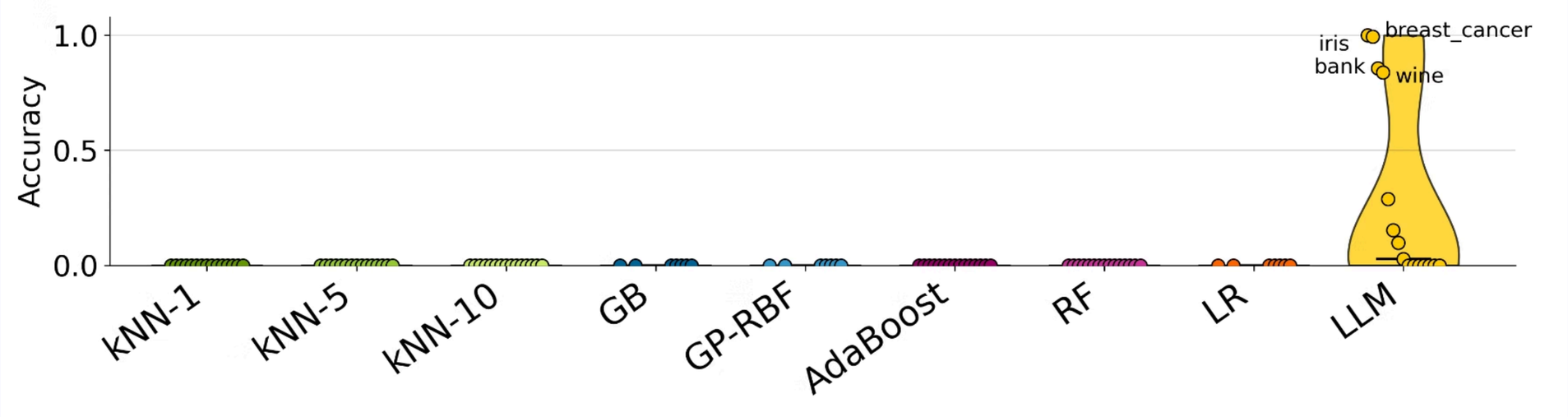
IRIS · ACCURACY OVER 30 RUNS

It works!

FIRST REACTION The LLM edges out every baseline. Maybe we *can* just ask.

Very good at benchmark toy problems... *Actually, too good*.

We train on two labels and test on the remaining one, so a genuine in-context learner must score zero against the originals. Baselines do: the LLM recovers the original labels of *breast cancer* and *iris* almost perfectly. It has memorised the task.



ACCURACY AGAINST ORIGINAL LABELS · PERMUTED-LABEL CONTEXT

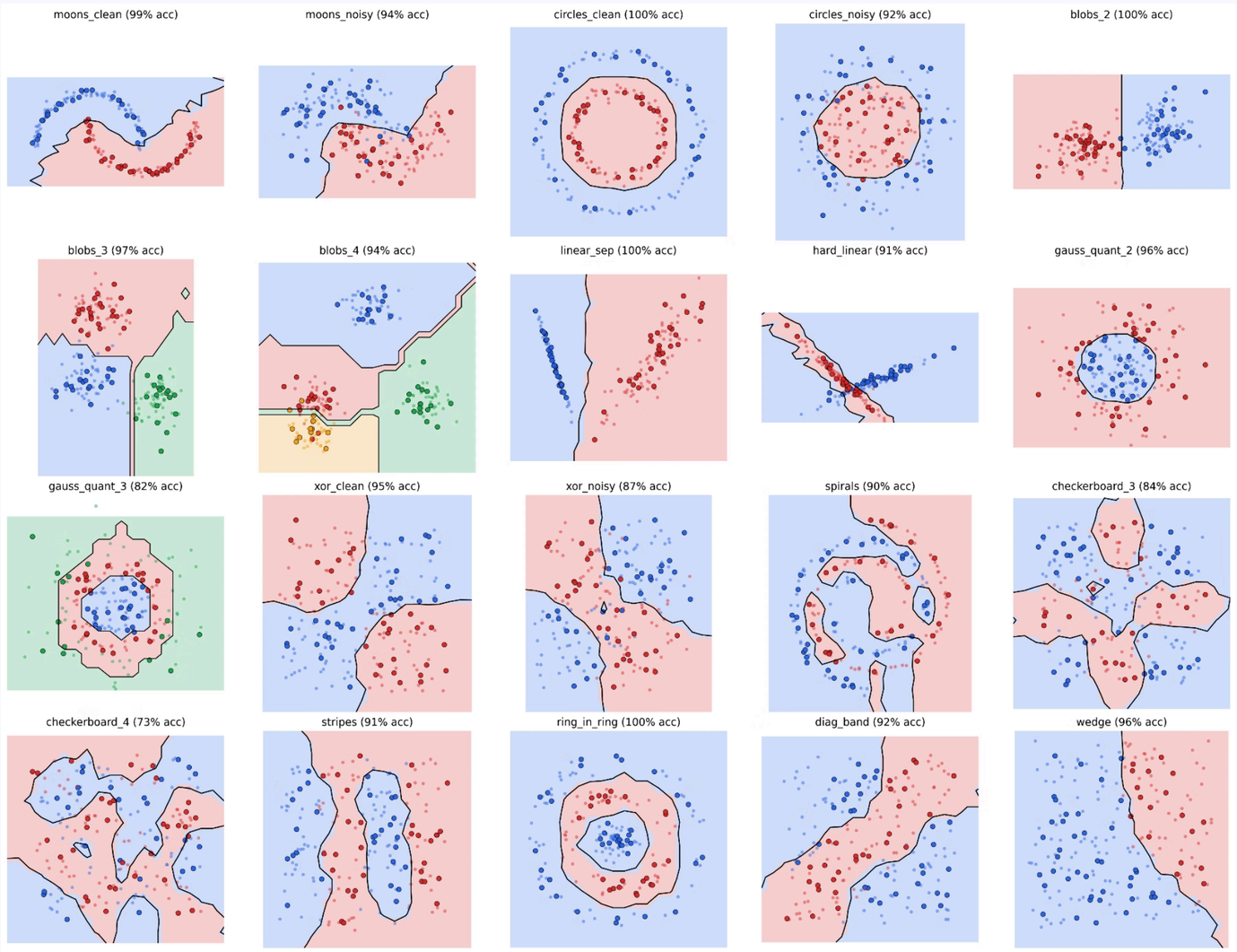
These datasets are in the training corpus, *labels included*.

In two dimensions, *it works*.

We map the LLM's prediction frontier on twenty synthetic 2-D tasks: 60 labeled points in the prompt, labels for a 20×20 grid out.

The boundaries are reasonable (moons, circles, XOR, rings) with *13 of 20 tasks above 90% accuracy*.

Note the shape of the frontiers: local and patchy, not smooth. Remember that.

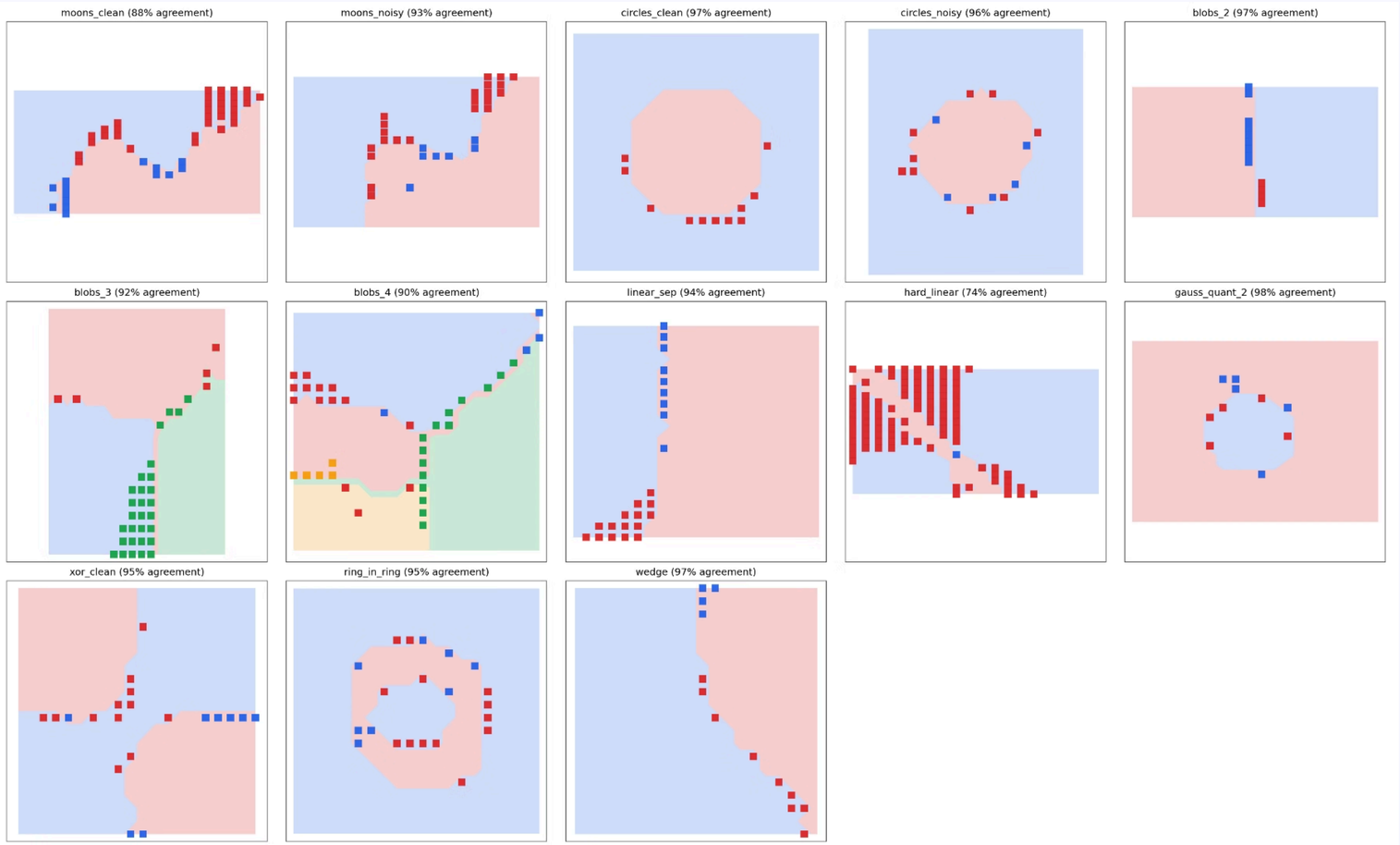


LLM PREDICTION FRONTIER · 20 TWO-DIMENSIONAL TASKS

The LLM's frontier is *1-NN's frontier*.

We overlay the LLM's grid predictions with a 1-nearest-neighbour classifier fit on the same 60 points. Shaded regions agree; the colored squares are the only disagreements.

1-NN matches *92.7% of grid cells* on tasks the LLM solves well: closer than any of 252 baselines.



LLM × 1-NN AGREEMENT PER TASK · SQUARES MARK DISAGREEING CELLS

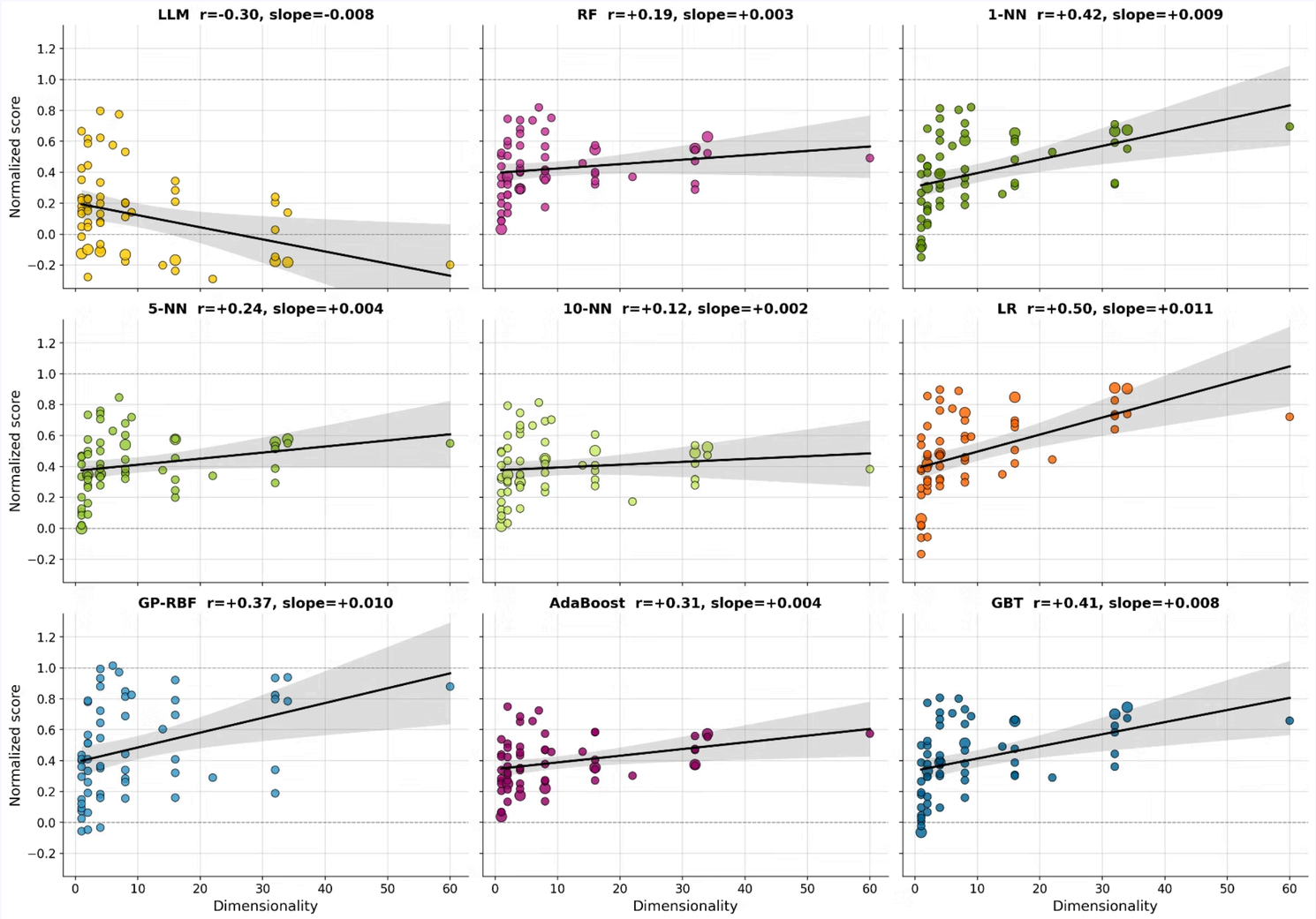
Where it works, the LLM predicts like a *very local nearest neighbour*.

Add columns, and only the LLM *collapses*.

Random projections of 12 datasets sweep the dimension while preserving the information. Baselines stay flat or improve.

The LLM is the only method of nine whose score falls as columns are added: r = −0.30 — at or below majority guessing by d ≳ 30.

A 2-column table is nearly a *sequence*.
A 30-column table is not.



NORMALISED SCORE VS. DIMENSIONALITY · 9 METHODS · 12 DATASETS

LLMs mainly rely on *memorisation* to make predictions, and *lose performance* as the number of columns increases.

Open-source benchmark scores are *contamination*, not capability.

Good at narrow sequence-like tables and behaves like *1-NN + noise*. Collapses as dimensions grow.

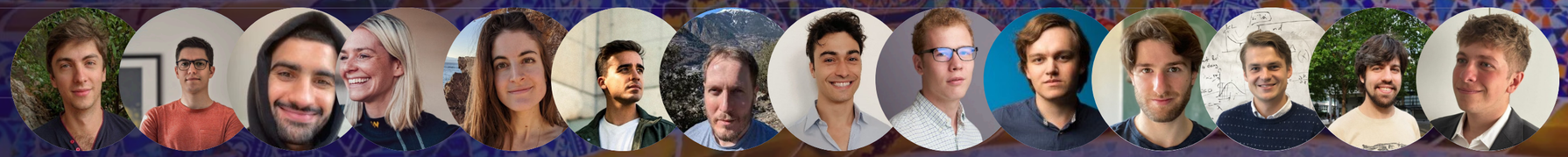
At least for the time being, tables need models *built for tables*.

FUNDAMENTAL

BARCELONA

Fundamental - *join us!*

We build NEXUS, a foundation model *native to tables*. From Barcelona.



SEE THE FUTURE IN YOUR DATA

fundamental.tech/careers