Deep Representation Learning for Tabular Data

Han-Jia Ye
Nanjing University
yehj@lamda.nju.edu.cn

Jun-Peng Jiang
Nanjing University
jiangjp@lamda.nju.edu.cn

Machine Learning

Machine learning has been applied in various fields successfully.

Features & Label Samples	F1
Sample_1	$v_{1,1}$
Sample_2	$v_{2,1}$
Sample_3	$v_{3,1}$
.	.
.	
.	
Sample_n	

Classification and regression

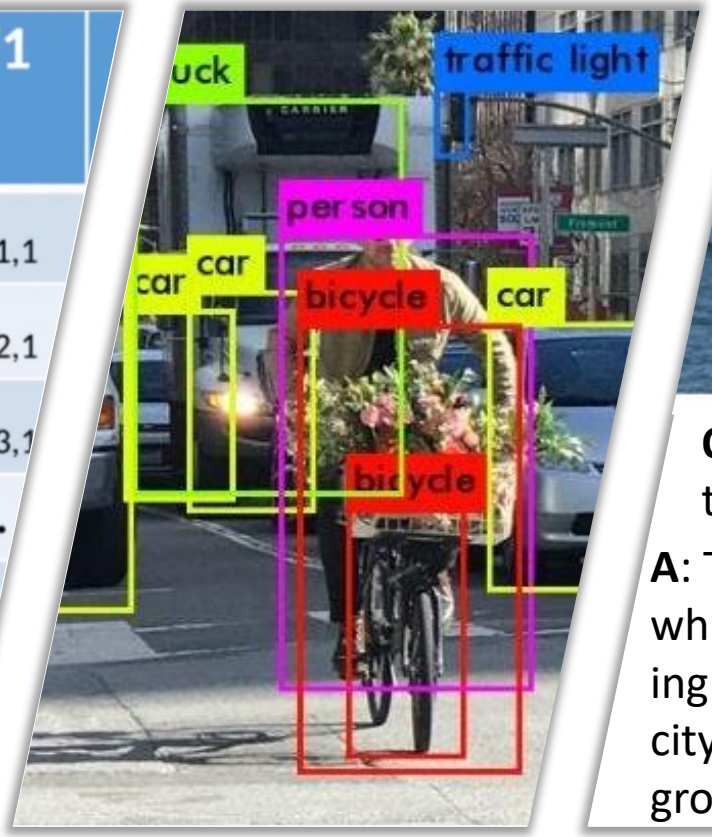


Image recognition and detection

Q: could you depict the picture

A: The image shows a white motor yacht cruising on the water with a city skyline in the background.

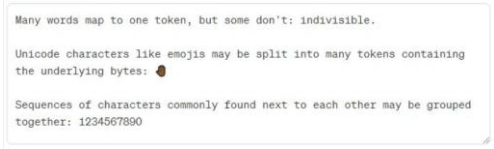
Vision-language understanding

Tiny potato kings: wearing intricate crowns, seated grand thrones, overlook a vast kingdom made of potato farms and village

Image/text generation

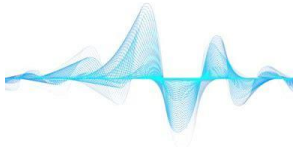
The Basic Process of Machine Learning

Text



Image

Audio



Video

Input data

Machine Learning Algorithm

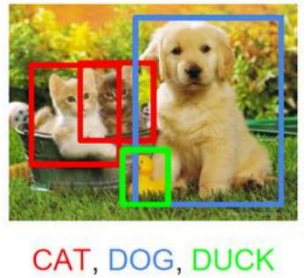
Output model

Recognition

Classification, image recognition

Detection

Object detection



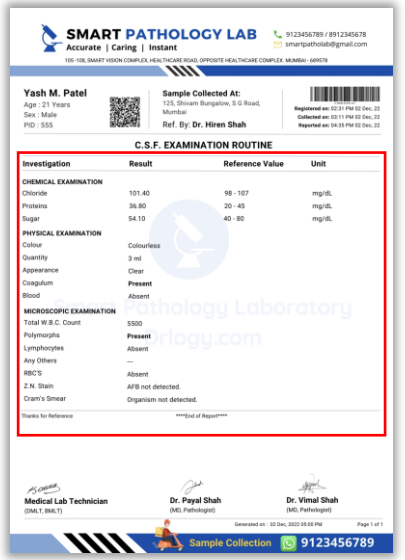
Generation

Image, text QA

Data in the “Table” Format

- Tabular data is prevalent across diverse domains in machine learning.
  - CTR prediction [Yan et al., 2014; Juan et al., 2016]
  - Healthcare [Hassan et al., 2020]
  - Medical analysis [Schwartz et al., 2007; Subasi, 2012]
  - E-commerce [Nederstigt et al., 2014]
  - AI for Science [Tibshirani et al., 2002; Ivanciuc et al., 2007]
  - …
- Famous tabular data sources

	1996		1991		growth in share
	% of national turnover	no. of stores	% of national turnover	no. of stores	(% points)
Austria	17	568	14	530	3
Belgium/Luxembourg	25	762	18	587	7
Denmark	20	739	15	544	5
Finland	12	820	10	760	2
France	7	1940	1	436	6
Germany	30	12130	24	8290	6
Greece	n.a.	n.a.	n.a.	n.a.	n.a.
Ireland	n.a.	n.a.	n.a.	n.a.	n.a.
Italy	10	2360	..	60	10
Netherlands	13	607	10	482	3
Portugal	9	314	2	30	7
Spain	9	2315	5	1180	4
Sweden	11	305	6	166	5
UK	11	1440	6	1129	5



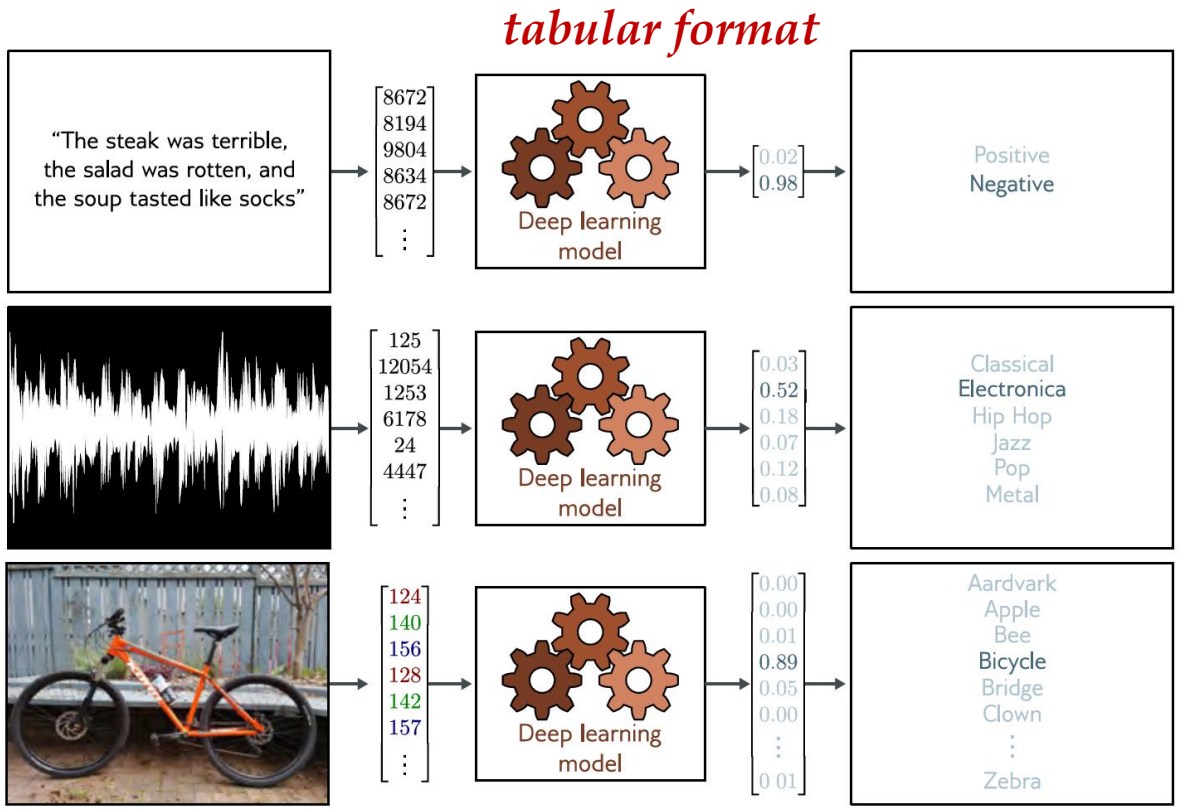
UCI Machine Learning Repository

OpenML

kaggle

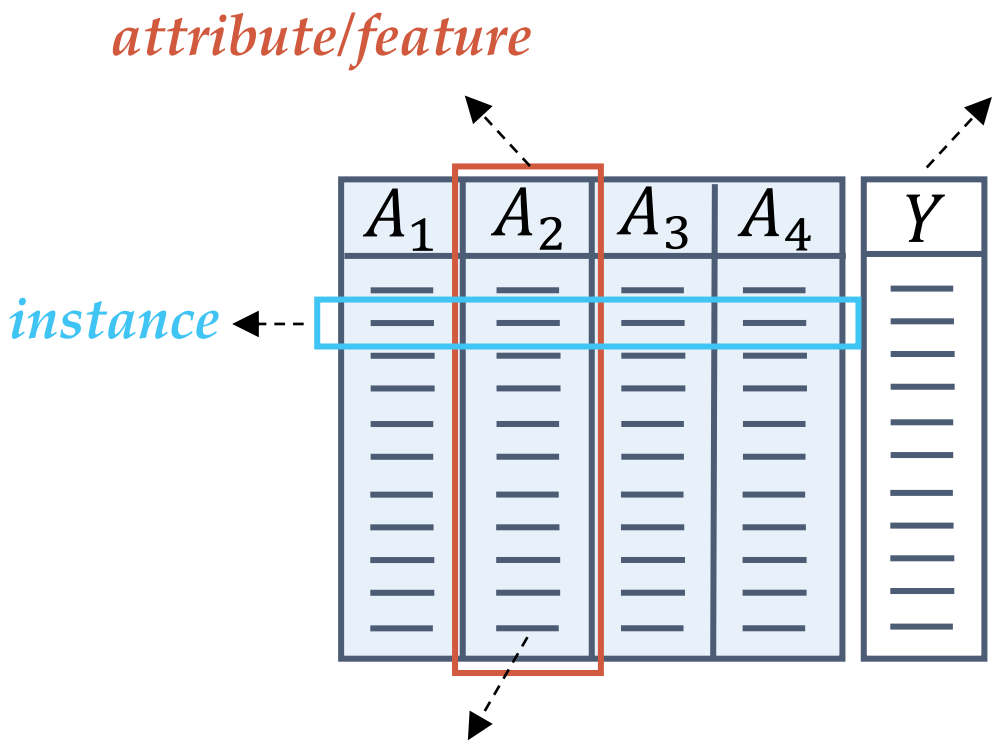
Data in the "Table" Format

Different types of data are "vectorized" into numerical data before further processings.



[Simon J.D. Prince, 2023]

Vectorized data organized via the tabular form.



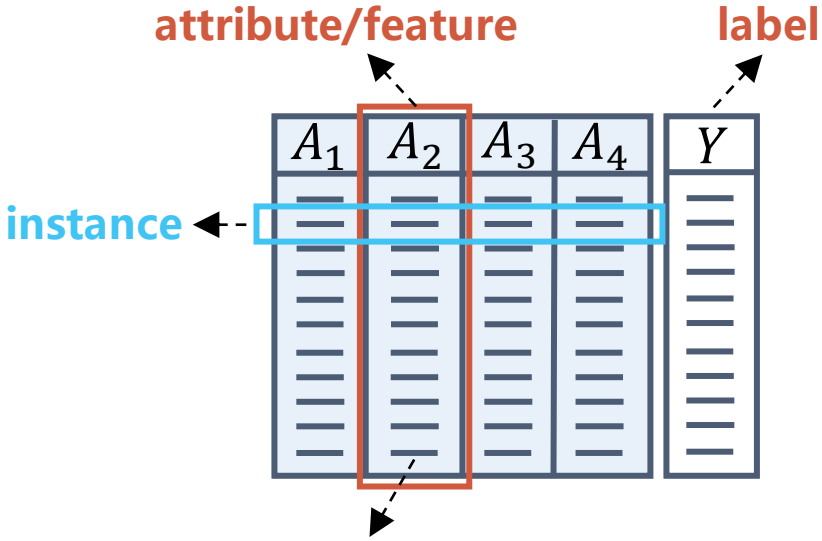
numerical attribute: e.g., 1, 6.7, 1024

categorical attribute: e.g., small, middle, large

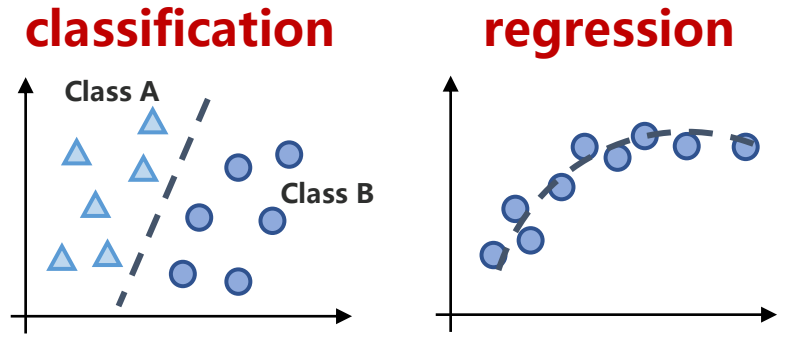
Learning with Tabular Data

- Given a training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$,
  - N instances (rows) and d columns (attributes)
  - $\boldsymbol{x}_i \in \mathbb{R}^d$, with ***categorical*** and ***numerical*** features/attributes
  - $y_i \in \{0,1\}$ for binary classification, $y_i \in \{1, \ldots, C\}$ for C-way classification, and $y_i \in \mathbb{R}$ for regression
- The goal is to learn a mapping f. Given an unseen instance $\boldsymbol{x}^*$,

$$\hat{y}^* = f(\boldsymbol{x}^*, \mathcal{D})$$



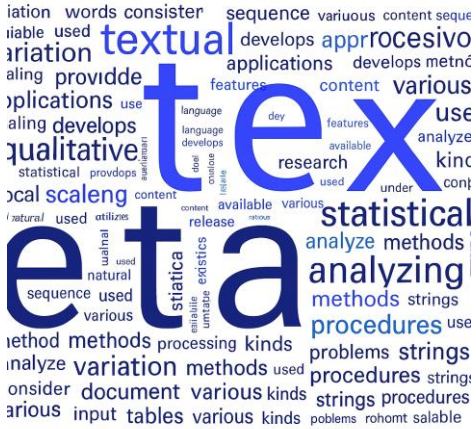
Learning with tabular data is one of the fundamental task in machine learning.



Unstructured data vs. Structured Data

Structured data is an important data source for AI models.

Unstructured data



- Implicit semantics
- High-dimensional raw signals
- Strong transferability via foundation models

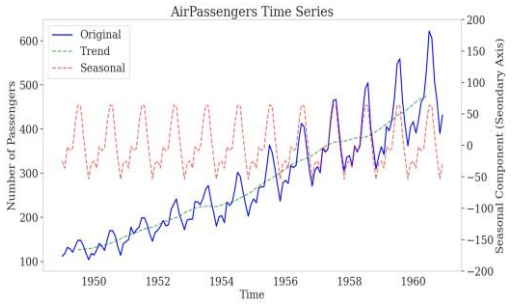
Structured data

Player	Minutes	Points	Rebounds	Assists
A	41	20	6	5
B	30	29	7	6
C	22	7	7	2
D	26	3	3	9
E	20	19	8	0
F	9	6	14	14
G	14	22	8	3
I	22	36	0	9
J	34	8	1	3

We can also introduce “tabular data” as one of the representative structured modalities.

Structured data

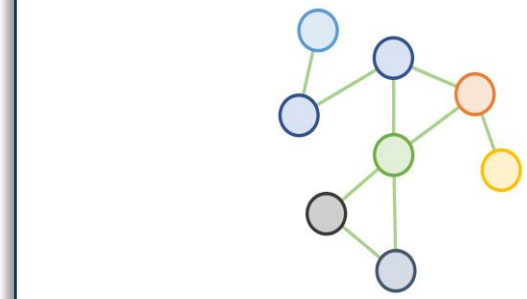
Various representative forms of structured data.



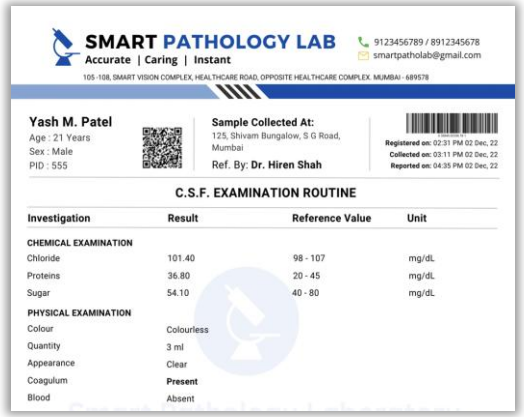
Time Series

Features & Label Samples	F1	F2	F3	F4	Label
Sample_1	$v_{1,1}$	$v_{1,2}$	$v_{1,3}$	$v_{1,4}$	L_1
Sample_2	$v_{2,1}$	$v_{2,2}$	$v_{2,3}$	$v_{2,4}$	L_2
Sample_3	$v_{3,1}$	$v_{3,2}$	$v_{3,3}$	$v_{3,4}$	L_3
.	.				.
.		.			.
.			.		.
Sample_n	$v_{n,1}$	$v_{n,2}$	$v_{n,3}$	$v_{n,4}$	L_n

Table



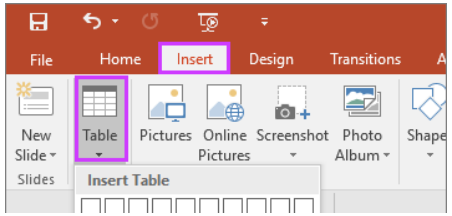
Graph

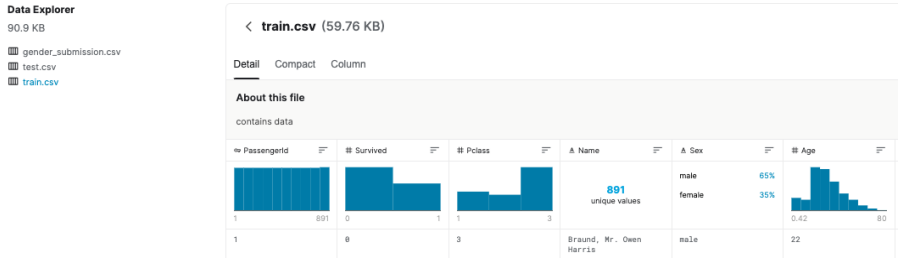


Tabular Data

Among them, tabular data is the most common form of structured data.

Common forms of tabular data






```
In [2]:
train_data = pd.read_csv("/kaggle/input/titanic/train.csv")
train_data.head()
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S

Some structured data tasks can be transformed into tabular tasks

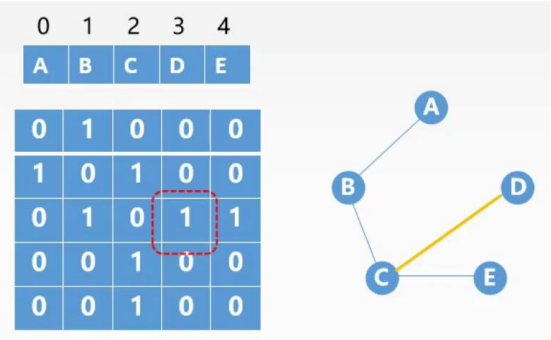
	AAPL.High	AAPL.Low	AAPL.Close
2008-09-16	20.4	18.9	20.0
2008-09-17	19.8	18.3	18.3
2008-09-18	19.3	17.2	19.2
2008-09-19	20.6	19.5	20.1
2008-09-22	20.0	18.7	18.7
2008-09-23	19.4	18.1	18.1
2008-09-24	18.7	17.9	18.4
2008-09-25	19.3	18.4	18.8
2008-09-26	18.5	17.6	18.3
2008-09-29	17.1	14.4	15.0
2008-09-30	16.4	15.2	16.2
2008-10-01	16.1	15.3	15.6
2008-10-02	15.5	14.3	14.3

Showing 1 to 13 of 2518 rows.

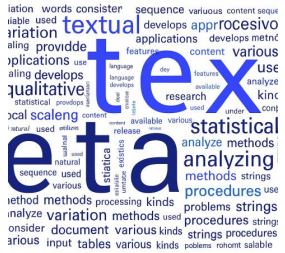
Start Previous Next End Find row

The table records the time series.

The adjacency matrix of the graph.



The Challenges of Tabular Data

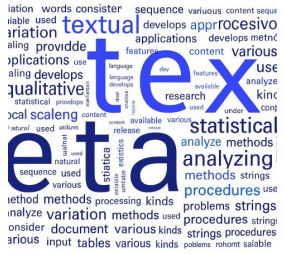


Player	Minutes	Points	Rebounds	Assists
A	41	20	6	5
B	30	29	7	6
C	22	7	7	2
D	26	3	3	9
E	20	19	8	0
F	9	6	14	14
G	14	22	8	3
I	22	36	0	9
J	34	8	1	3

	Image	Text	Table
Visualization	Pixels are visible and understandable to humans	The sequence is readable and understandable to humans	It is difficult to observe directly after **exceeding 2 to 3 dimensions**
Feature type	Continuous	Discrete	**Mixture**
Feature space	Pixels are colors, and their features have the same origin	Shared word dictionary, with same feature origin	**The meanings of different dimensions vary greatly**
Data source	“naturally” generated	“naturally” generated	Artificial construction, relying on task design

https://calvinmccarter.substack.com/p/the-idiosyncrasies-of-tabular-data

Comparison: Table vs. image vs. text



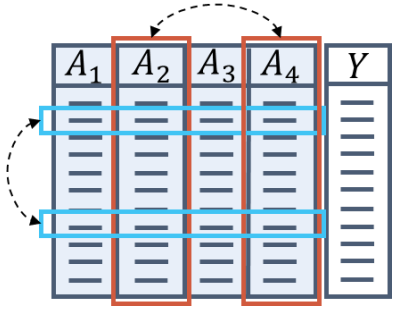
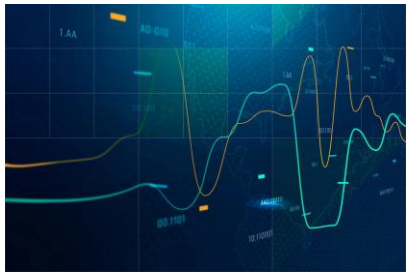
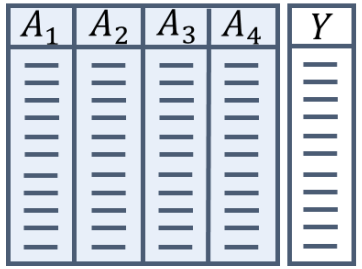


	Image	Text	Table
Inner dimension	a spatial manifold structure	a semantic sequential structure	**no linear manifold structure**
Internal sequence	spatial order (pixel neighborhood)	natural order (semantic sequence)	**no natural order and the features are independent**
Robustness	the disturbance of small pixels is not obvious	synonyms can be replaced, the order can be fine-tuned, and with robustness	**highly sensitive** to small disturbances of key features
Missing value	relatively rare	relatively rare	**widespread,** and the absence is meaningful

Comparison: Table vs. Time series

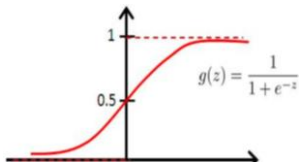




	Time series	Table
Feature space	strong homogeneity (usually continuous values)	mixed type of features
Internal sequence	a natural sequence (equal interval time sequence relationship), and the characteristics evolve over time	no natural order and the features are independent
Robustness	minor disturbances do not affect the overall trend	highly sensitive to small disturbances of key features
Correlation	a strong correlation between the time adjacent points. Different channels also have correlations	The correlation between features is weak or unpredictable

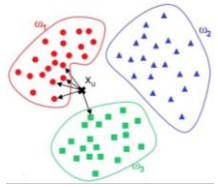
Classical Tabular Methods

Deep learning is developing rapidly in CV and NLP



Logistic Regression

Before 1900



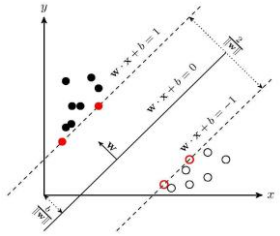
KNN

1951

Birth of AI

Perceptron & MLP

1957

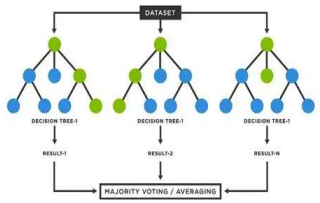


Kernel SVM

1990s

Random Forest

2002



Design models from various perspectives

XGBoost

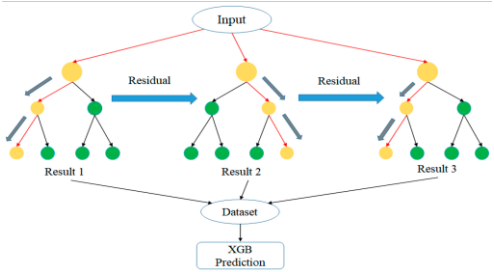
2016

LightGBM

2017

CatBoost

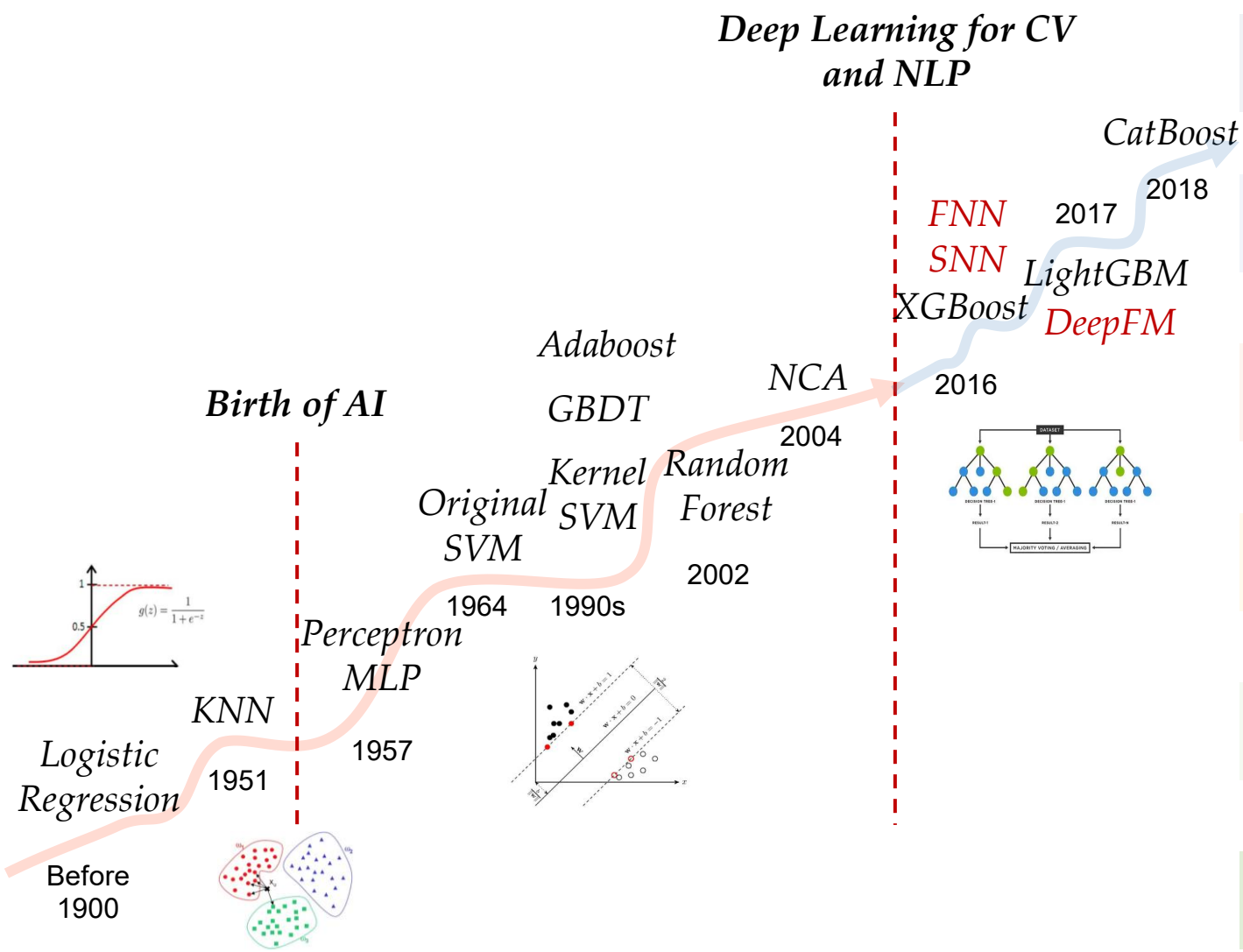
2018



Further improve through ensemble learning

Can deep learning technologies be applied to structured tabular data prediction tasks?

Tabular Methods are Constantly Evolving



MLP Variants

SNN [NIPS'17], MLP[NeurIPS'21], MLP-PLR [NeurIPS'22]

Special Designed DNNs

DCN v2 [NIPS'17], DANets [NeurIPS'21], TabCaps [NeurIPS'22]

Tree-mimic DNNs

NODE [ICLR'20], GrowNet [CoRR'21], TabNet [AAAI'24]

Transformer-Variants

AutoInt [CIKM'19], TabTransformer [CoRR'20], FT-T [NeurIPS'21]

Objective-Based

TANGOS [ICLR'23], SwithTab [AAAI'24], PTaRL [ICLR'24]

Context-Based

TabPFN [ICLR'23], TabR [ICLR'24]

Schedule of the Tutorial

Tutorial Part 1 (by Han-Jia)

Tabular prediction with deep representation learning,

From basic to foundation models

Tutorial Part 2 (by Jun-Peng)

Tabular data learning with large language models,

From prediction to other tasks

Slides of the tutorial

Outline

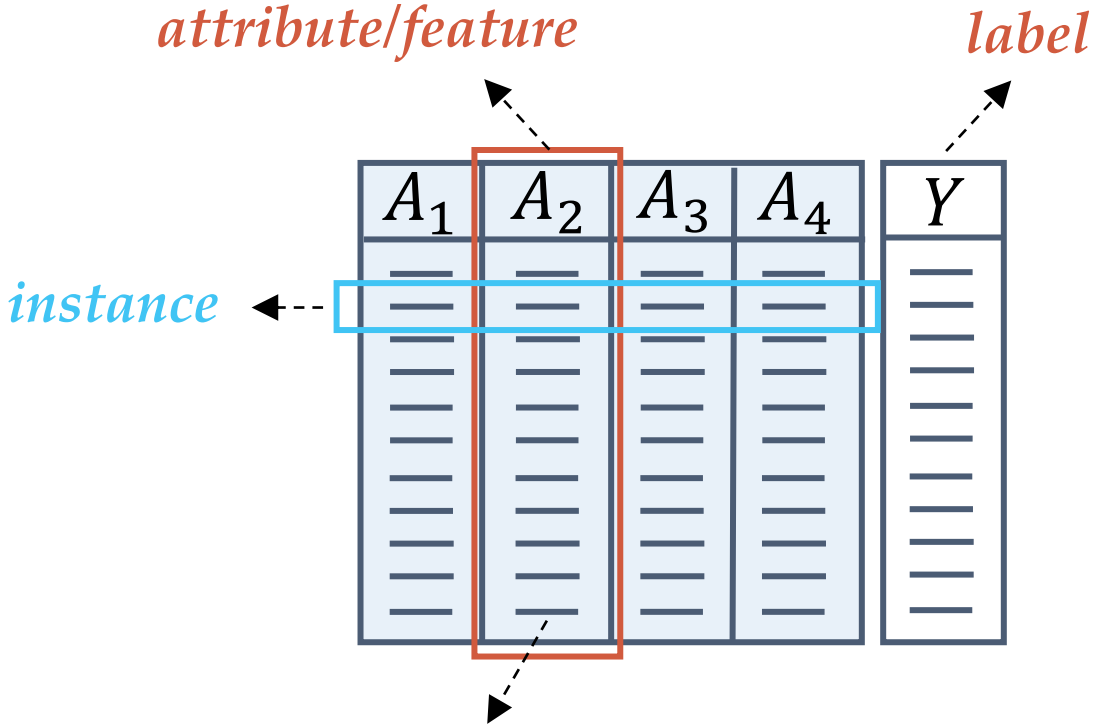
- Introduction to tabular data tasks and evaluation methods
- From classic tabular models to deep tabular models
- From specialized tabular models to general tabular models
- The preliminary evaluation and analyses of the tabular prediction models

Outline

- Introduction to tabular data tasks and evaluation methods
- From classic tabular models to deep tabular models
- From specialized tabular models to general tabular models
- The preliminary evaluation and analyses of the tabular prediction models

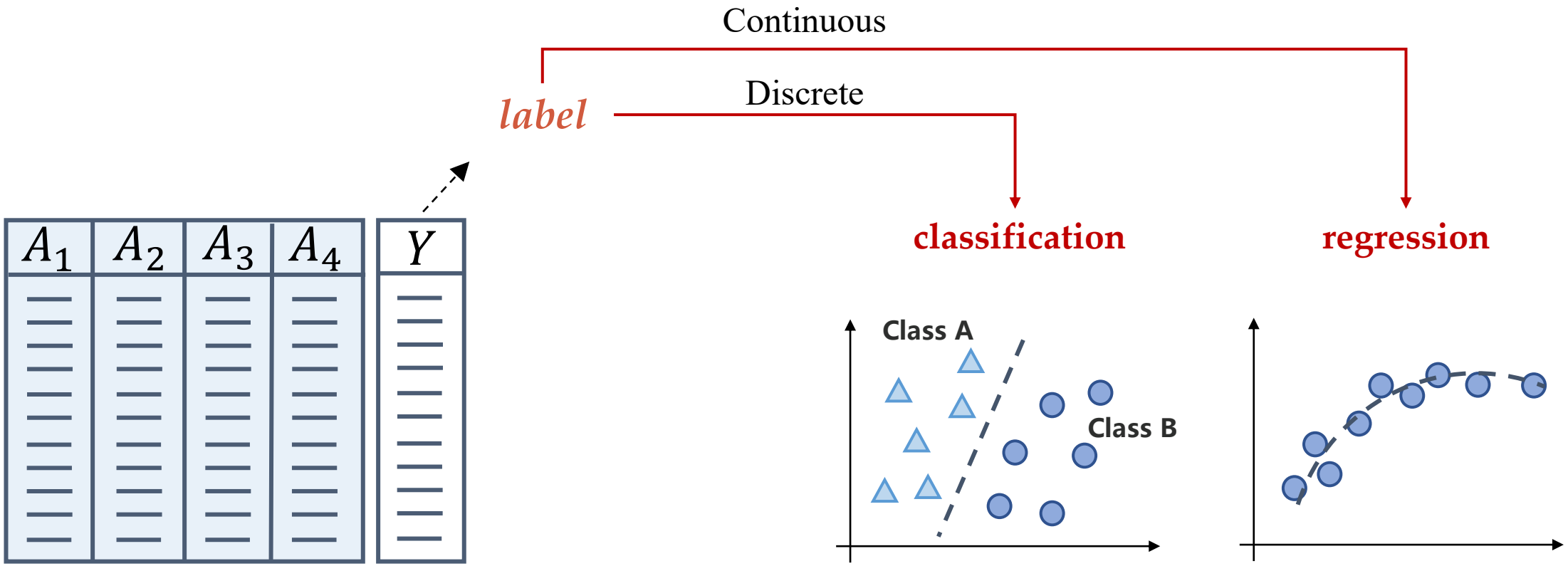
Tabular Data

- Tabular data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$,
- N instances (rows), d attributes (columns).
- Instance $\boldsymbol{x}_i \in \mathbb{R}^d$, with numerical and categorical attributes.
- Label $y_i \in \{0,1\}$ means binary classification task, $y_i \in \{1, \ldots, C\}$ means multi-class classification task, $y_i \in \mathbb{R}$ means regression task.



Tabular Data Task

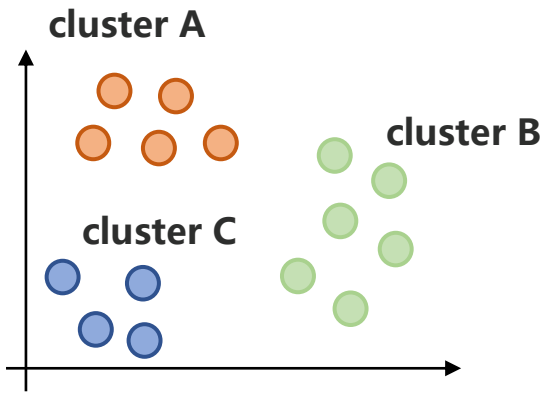
Typical tabular data (prediction) tasks are categorized by the label form.



Tabular Data Task

Other tabular data (prediction) tasks.

Clustering



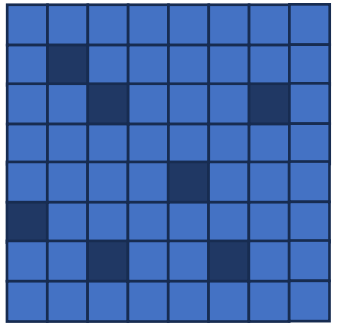
Group the unlabeled samples

Missing value completion/generation

Age	Sex	Income
25	-	<= 50K
37	F	-

Fill in the missing attributes

Anomaly detection



Identify outliers that are different from the main sample

Evaluation of Tabular Models

Evaluation criteria for a single tabular dataset.

Classification Task

- **Accuracy:** Measure the proportion of samples whose predictions are correct among the total samples

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- **F1:** The harmonic average of precision and recall, used to balance false positives and false negatives

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recal}}$$

- **AUC:** Evaluate the model's ability to distinguish different categories at all possible thresholds

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\text{dFPR}$$

Regression Task

- **MSE:** Measure the average of the squares of the differences between the predicted value and the true value

$$\text{MSE} = \frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2$$

- **MAE:** The average of the absolute values that measure the difference between the predicted value and the true value is more robust and less affected by outliers

$$\text{MAE} = \frac{1}{n}\sum_{i=0}^{n}|y_i - \hat{y}_i|$$

- $\boldsymbol{R^2}$: The ability of a model to interpret data changes is measured. The closer the value is to 1, the better the fitting effect of the model

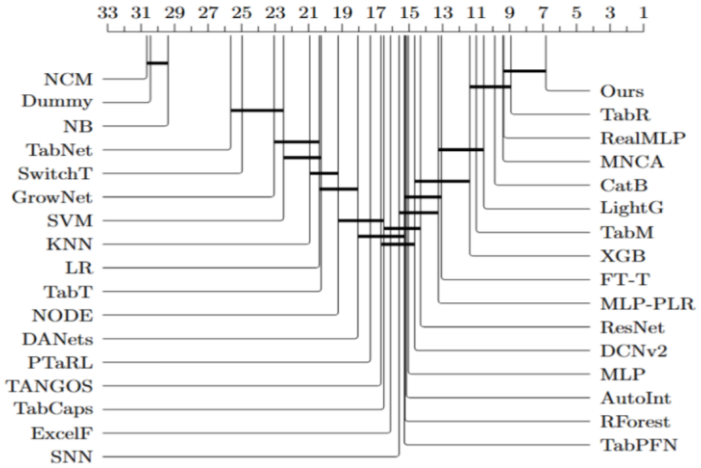
$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$$

Evaluation of Tabular Models

Evaluation criteria for multiple tabular datasets.

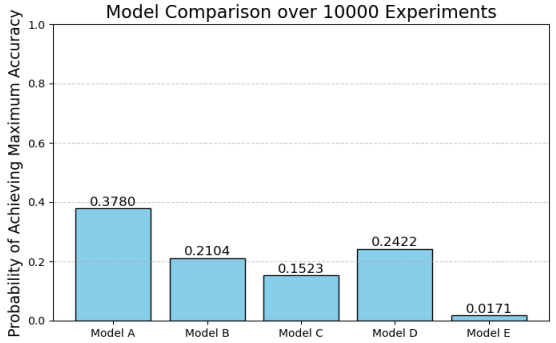
Average Rank

For each dataset, all methods are sorted, and the average ranking of all methods across all datasets is calculated. Generally, this is presented in conjunction with tests such as Wilcoxon-Holm



PAMA (Probability of Achieving the Maximum Accuracy)

The probability that a certain model becomes the one with the highest accuracy among R experimental comparisons of M models and algorithms



$$\mathrm{PAMA}(m) = \frac{1}{R}\sum_{jr=0}^{R} 1\left[m = \arg\max_{j \in \mathrm{M}} \mathrm{Acc}_{j,r}\right]$$

Evaluation of Tabular Models

Evaluation criteria for multiple tabular datasets.

SGM (Shifted Geometric Mean error)

By translating and geometrically averaging the errors, the influence of extreme values is reduced, and the prediction errors are measured more robustly

$$\mathrm{SGM} = \exp\left(\frac{1}{n}\sum_{i=0}^{n} ln(|y_i - \hat{y}_i| + \epsilon)\right) - \epsilon$$

NRMSE (Normalized Root Mean Squared Error)

Normalize the RMSE to make data of different dimensions comparable and measure the deviation between the predicted value and the true value

$$\mathrm{NRMSE} = \frac{\sqrt{\frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2}}{y_{max} - y_{min}}$$

Relative Improvement

It is a very common metric in the comparison of machine learning and experimental results, used to measure how much a method has improved compared to the baseline

Method	Adult	Default	Shoppers	Magic	Beijing	News	Average
SMOTE	3.28±0.29	8.41±0.38	3.56±0.22	3.16±0.41	2.39±0.35	5.38±0.76	4.36
CTGAN	20.23±1.20	26.95±0.93	13.08±0.16	7.00±0.19	22.95±0.08	5.37±0.05	15.93
TVAE	14.15±0.88	19.50±0.95	18.67±0.38	5.82±0.49	18.01±0.08	6.17±0.09	13.72
GOGGLE	45.29	21.94	23.90	9.47	45.94	23.19	28.28
GReaT	17.59±0.22	70.02±0.12	45.16±0.18	10.23±0.40	59.60±0.55	OOM	44.24
STaSy	14.51±0.25	5.96±0.26	8.49±0.15	6.61±0.53	8.00±0.10	3.07±0.04	7.77
CoDi	22.49±0.08	68.41±0.05	17.78±0.11	6.53±0.25	7.07±0.15	11.10±0.01	22.23
TabDDPM	3.01±0.25	4.89±0.10	6.61±0.16	1.70±0.22	2.71±0.09	13.16±0.11	5.34
TabSyn	**1.54±0.27**	**2.05±0.12**	**2.07±0.21**	**1.06±0.31**	**2.24±0.28**	**1.44±0.03**	**1.73**
Improve.	**48.8% ↓**	**58.1% ↓**	**68.7% ↓**	**37.6% ↓**	**17.3% ↓**	**53.1% ↓**	**67.6% ↓**

$$\mathrm{RI} = \frac{M_{new} - M_{base}}{M_{base}}$$

Outline

- Introduction to tabular data tasks and evaluation methods
- From classic tabular models to deep tabular models
- From specialized tabular models to general tabular models
- The preliminary evaluation and analyses of the tabular prediction models

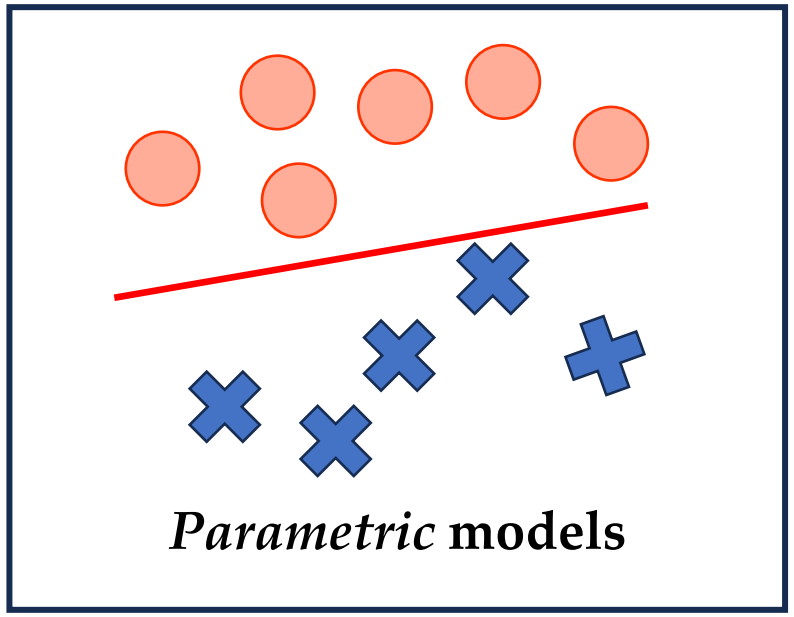
The Basic Learning Form of Tabular Data

Given a tabular dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, learn a mapping f which predicts an unseen instance $\boldsymbol{x}_i$.

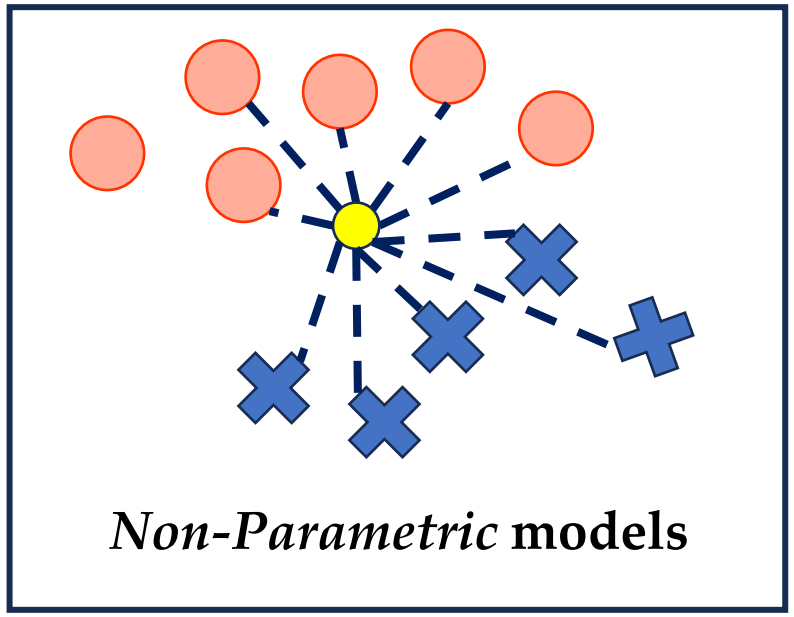
loss function

$$\min_{f} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}} \ell\big(y, \hat{y}_i = f(\boldsymbol{x}_i)\big) + \Omega(f)$$

Regularization



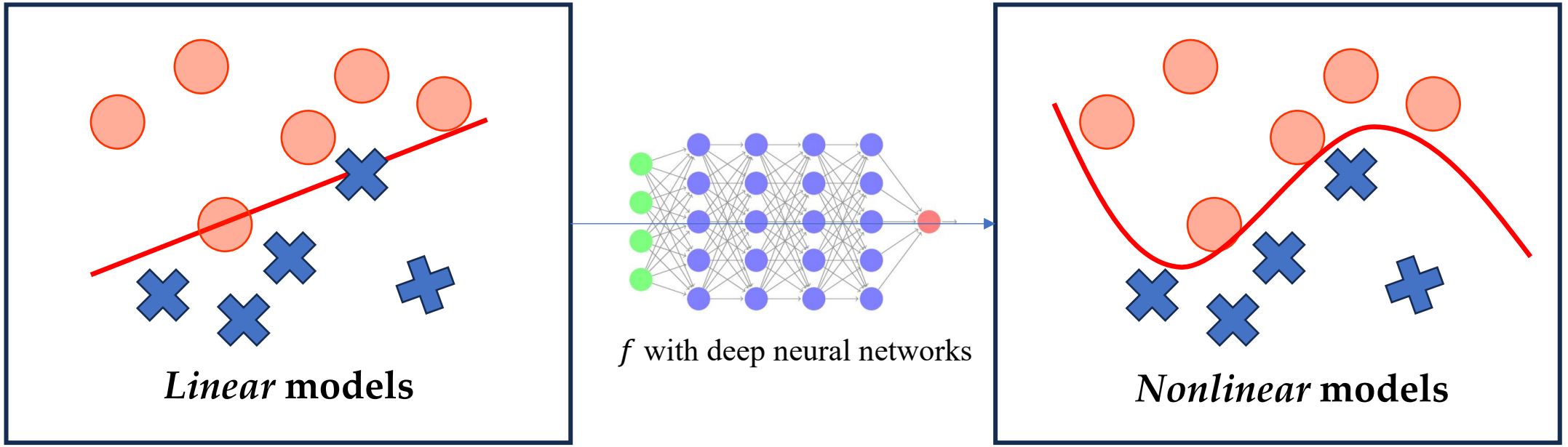
Parametric models



Non-Parametric models

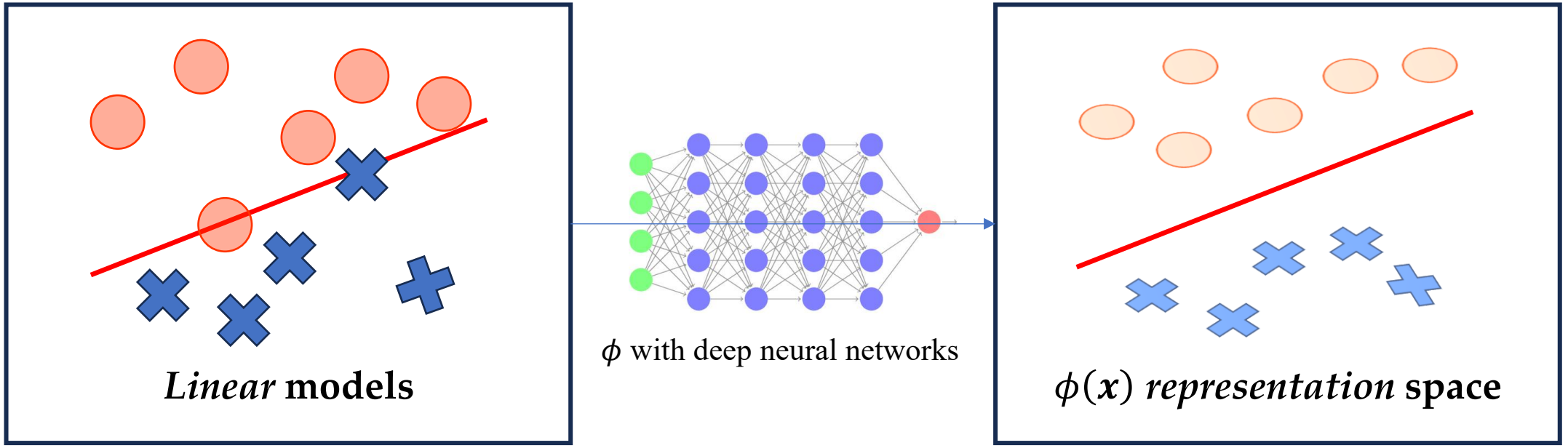
Tabular Prediction with Deep Representation Learning

The basic idea of deep learning for tabular data.



Tabular Prediction with Deep Representation Learning

The basic idea of deep learning for tabular data.



$$f(\boldsymbol{x}) = W^{\top}\phi(\boldsymbol{x}) + b$$

representation space mapping

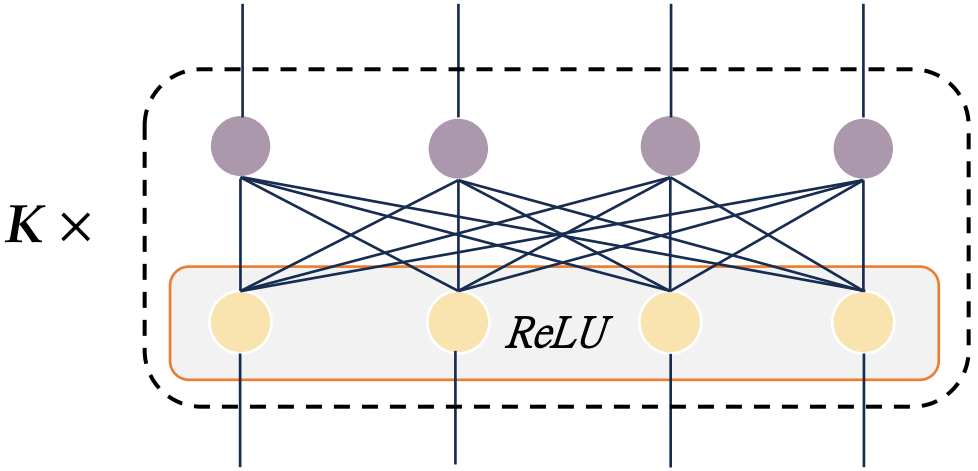
Learning high-quality ***feature representations*** for tabular data in the representation space can achieve good predictions with simple models. There have been related studies in the early days in the fields of recommendation systems and CTR [Zhang et al., ECIR'16] [Song et al., CIKM'19]

Rethinking Deep Tabular Prediction Models

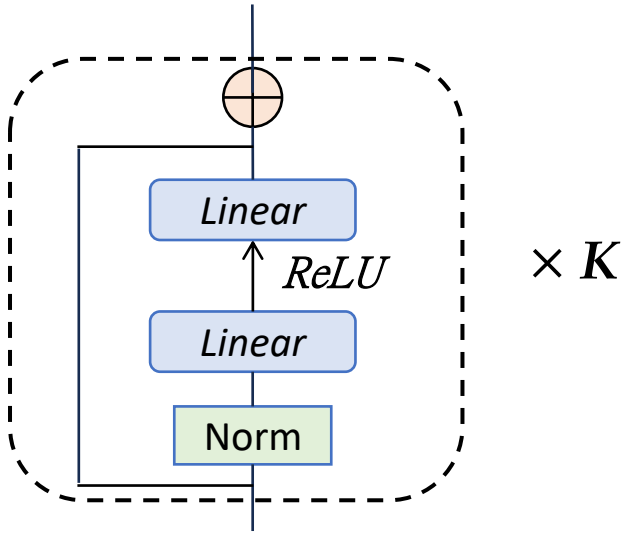
Starting from the most classic neural network model MLP, consider the new design of MLP with modern blocks.

$$\text{MLP}(x) = \text{Linear}\,(\text{MLPBlock}\,(\ldots(\text{MLPBlock}(x))))$$

$$\text{MLPBlock}(x) = \text{Dropout}(\text{ReLU}(\text{Linear}(x)))$$



Multilayer Perceptron (MLP)

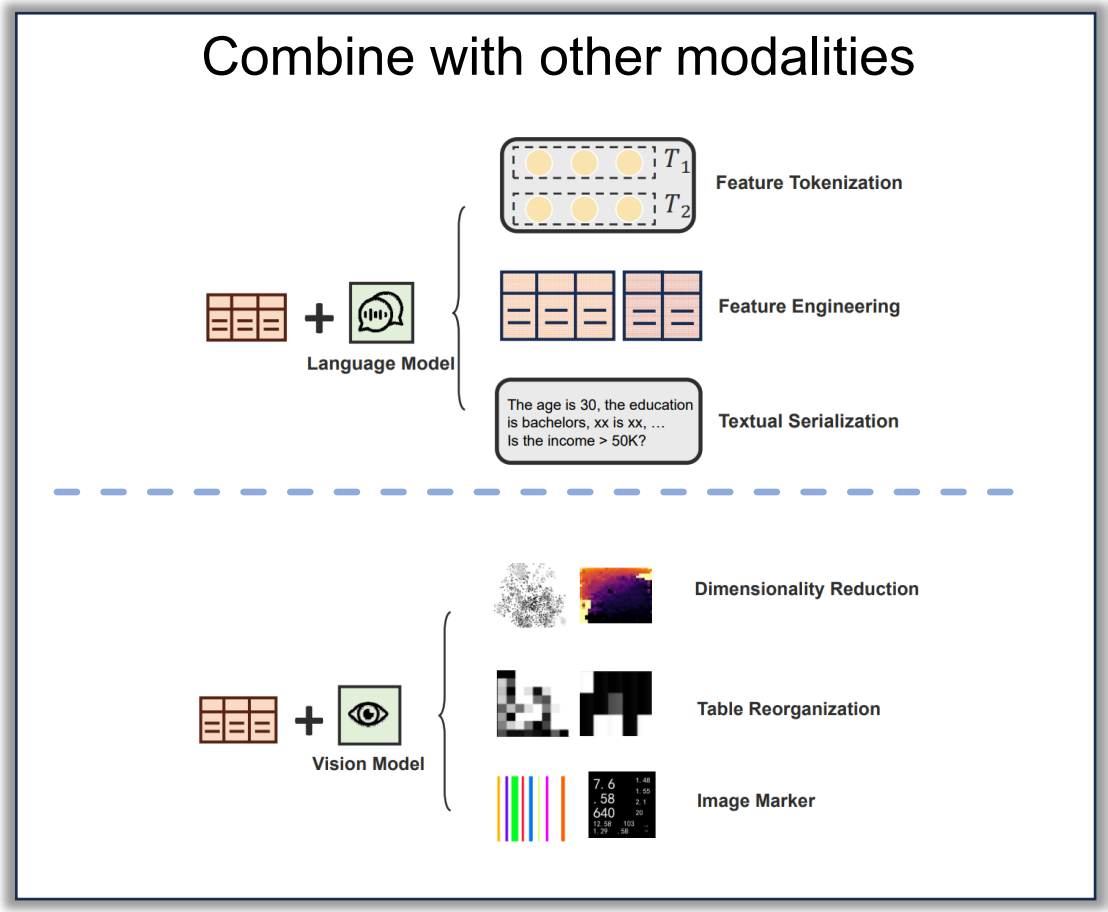


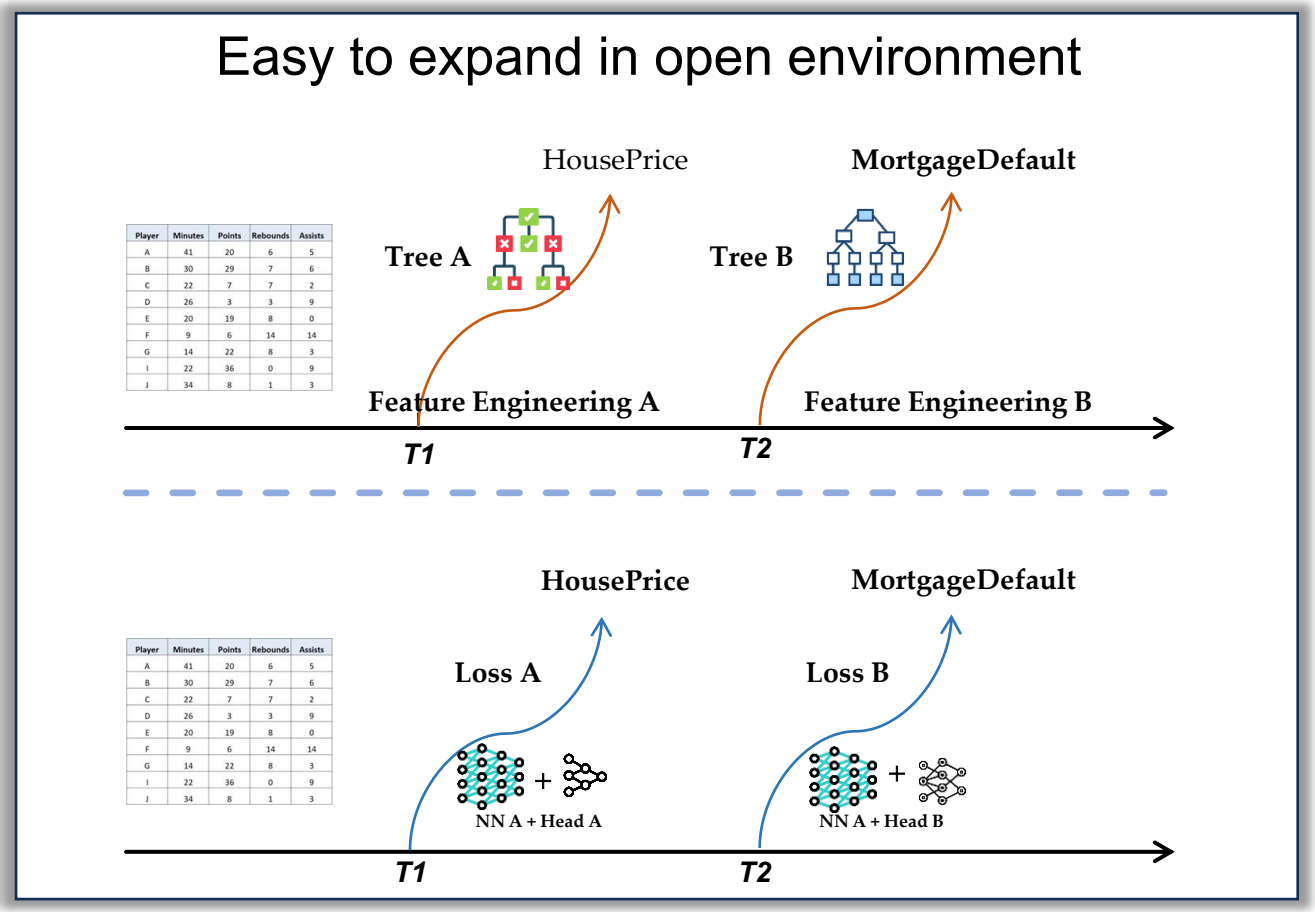
ResNet

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, Artem Babenko. ***Revisiting Deep Learning Models for Tabular Data***. NeurIPS 2021.
Arlind Kadra, Marius Lindauer, Frank Hutter, Josif Grabocka. ***Well-tuned Simple Nets Excel on Tabular Datasets***. NeurIPS 2021.

Tabular Prediction with Deep Learning

The advantages of deep tabular prediction models.

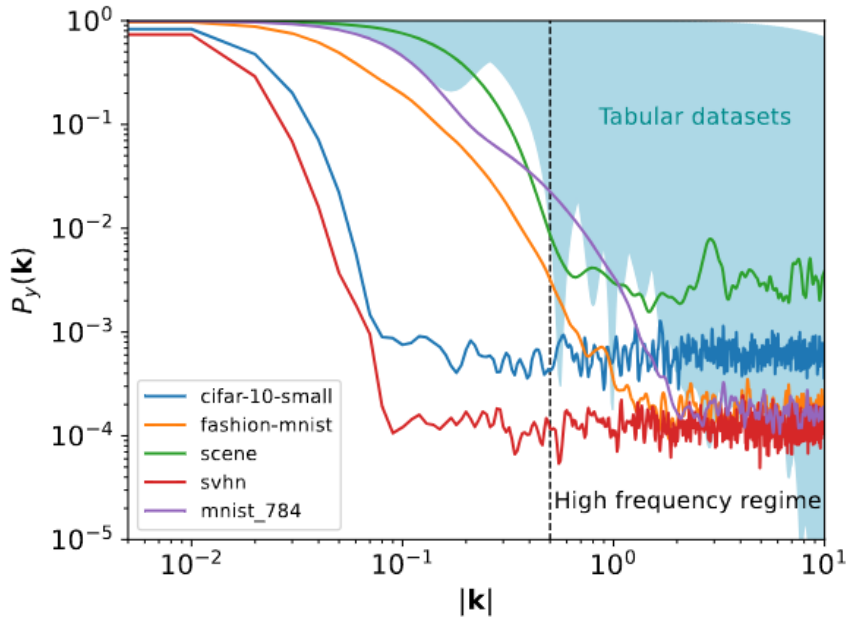




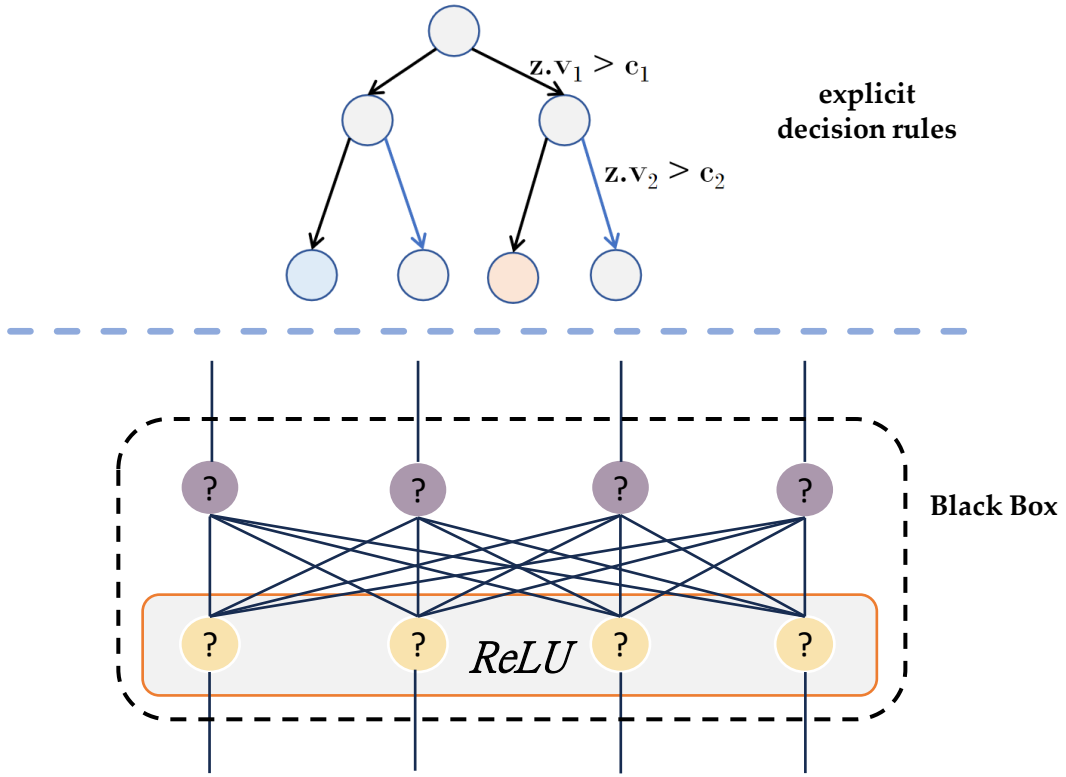
Tree Models vs. Deep Models

Does the tree model still have an advantage?

More advantages in processing *high-frequency* data



Better interpretability

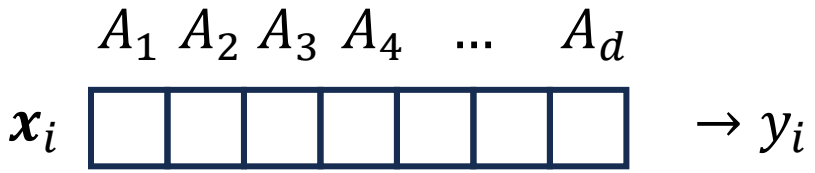


Ege Beyazit, Jonathan Kozaczuk, Bo Li, Vanessa Wallace, Bilal Fadlallah. ***An Inductive Bias for Tabular Deep Learning***. NeurIPS 2023.

A Rough Categorization of Deep Tabular Models

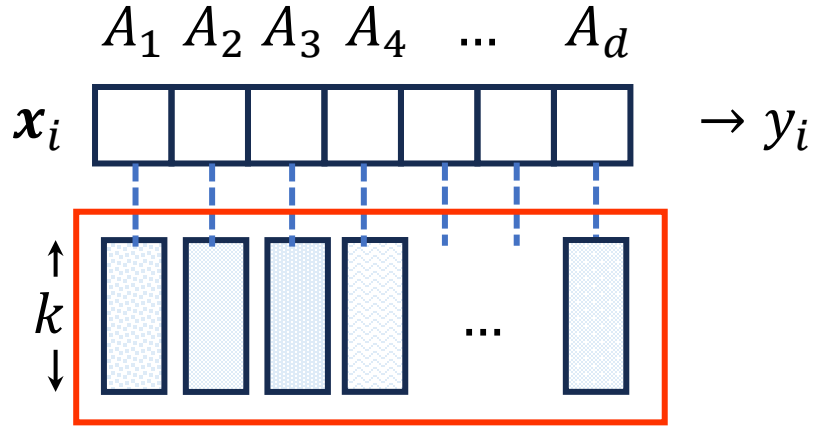
Learning from **raw data**:

- The training set is in the form of $N \times d$



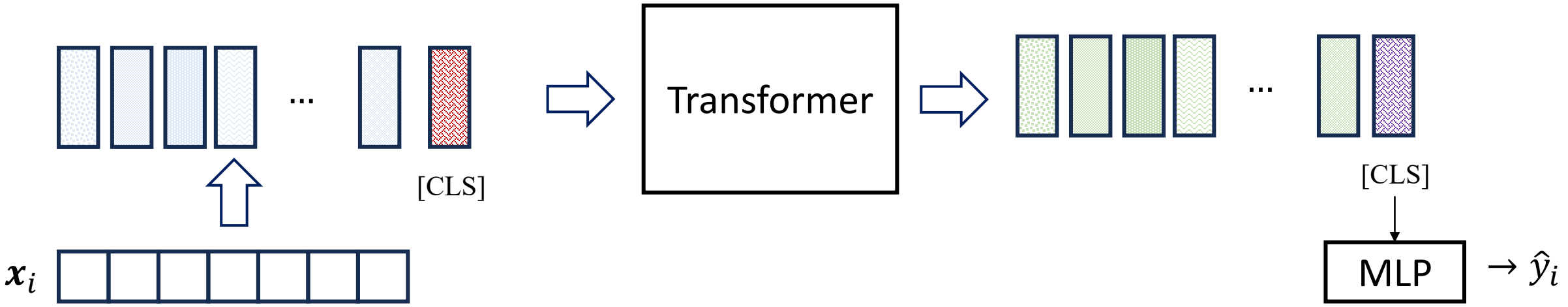
Tokenize the features:

- Categorical feature: Each attribute value is assigned a learnable token
- Numerical feature: Multiply the attribute value by a publicly learnable token
- The training set is in the form of $N \times d \times k$



FT-Transformer (FT-T)

- Tokenize instance x_i, transformed into a collection of d k-dimension tokens
- All tokens are input into a Transformer and concatenated with a [CLS] token. The corresponding input is then predicted through MLP



FT-T learns the interaction and fusion between features more flexibly through the self-attention mechanism. It introduces bias for each feature in the tokenizer and introduce [CLS] tokens to aggregate global information.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, Artem Babenko. ***Revisiting Deep Learning Models for Tabular Data***. NeurIPS 2021.

Feature Encoding

- Numerical feature encoding is not merely a “dimensional upgrade”, but rather a readjustment of feature scale and correlation in the latent space
- Through a set of learnable parameters $c_1 \ldots c_k$, periodic embedding mapping values to vectors through sine and cosine functions

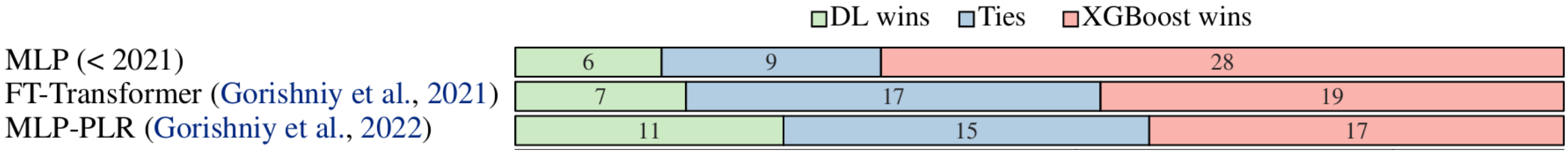
$$\text{Periodic}(x) = \text{concat}[\sin(v), \cos(v)], v = [2\pi c_1 x, \ldots, 2\pi c_k x]$$

- PLR embedding adds Linear and ReLU layers after it:

$$\text{PLR}(x) = \text{ReLU}(\text{Linear}(\text{Periodic}(x)))$$

- Learn richer semantic representations for numerical values by mapping numerical scalars to vectors with values between [-1,1]

The performance of MLP combined with PLR (MLP-PLR) can be significantly improved, even surpassing FT-T.



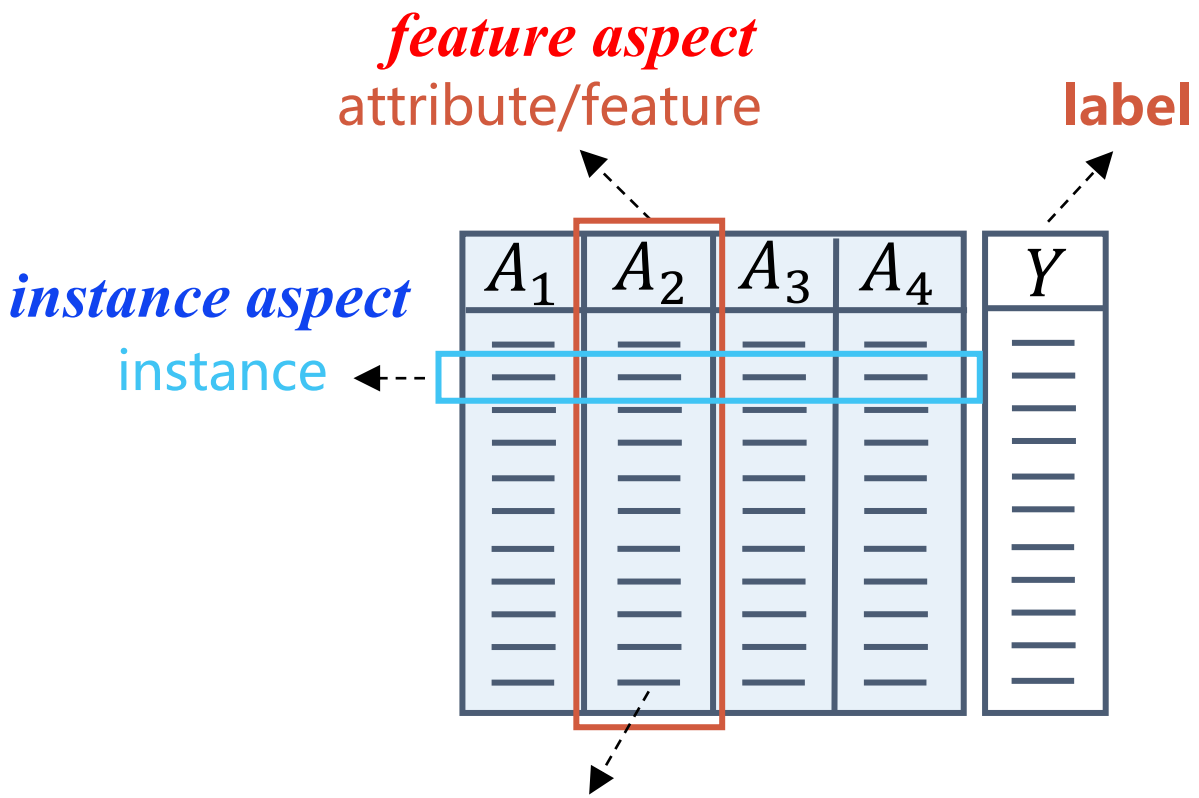
Yury Gorishniy, Ivan Rubachev, Artem Babenko. ***On embeddings for numerical features in tabular deep learning***. in NeurIPS 2022.

Outline

- Introduction to tabular data tasks and evaluation methods
- From classic tabular models to deep tabular models
- From specialized tabular models to general tabular models
- The preliminary evaluation and analyses of the tabular prediction models

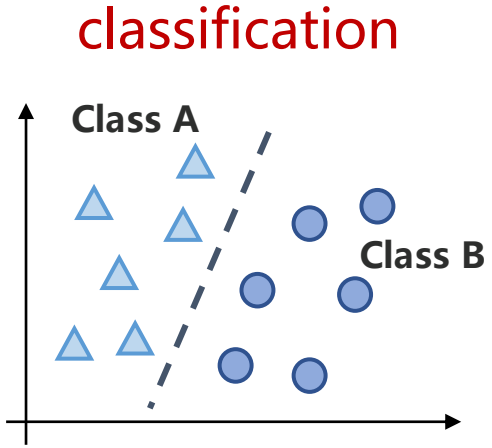
The Taxonomy of Tabular Models

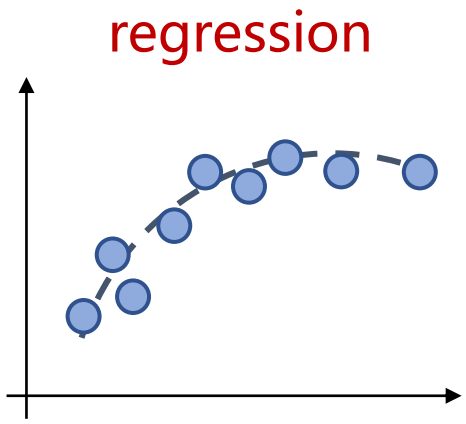
Categorize deep tabular methods via several learning factors.



numerical attribute: *e.g., 1, 6.7, 1024*

categorical attribute: *e.g., small, middle, large*





Object aspect

$$\min_f \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}} \ell(y,\ \hat{y}_i = f(\boldsymbol{x}_i)) + \Omega(f)$$

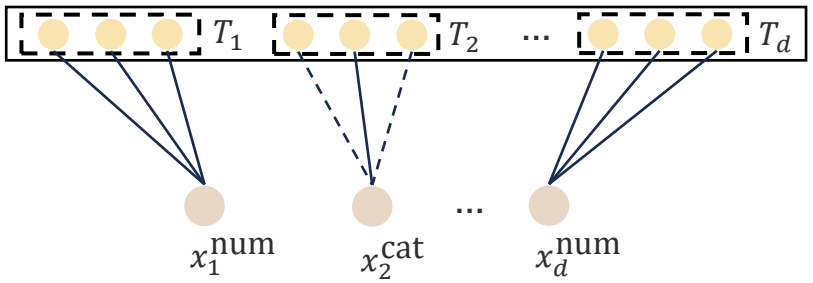
$$\boldsymbol{L}_{\mathrm{CE}} + \boldsymbol{L}_{\boldsymbol{regular}}$$

$$\boldsymbol{L}_{\mathrm{MSE}} + \boldsymbol{L}_{\boldsymbol{regular}}$$

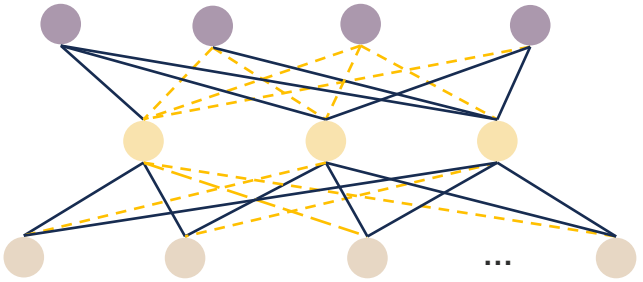
Taxonomy of Specialized Methods

From feature aspect, we can further categorize methods via how they process features.

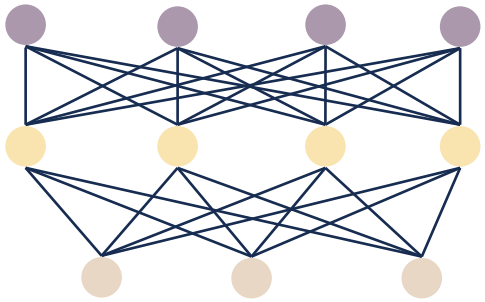
feature encoding



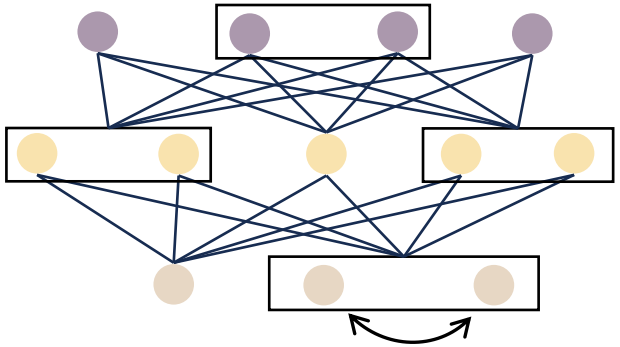
feature selection



feature projection



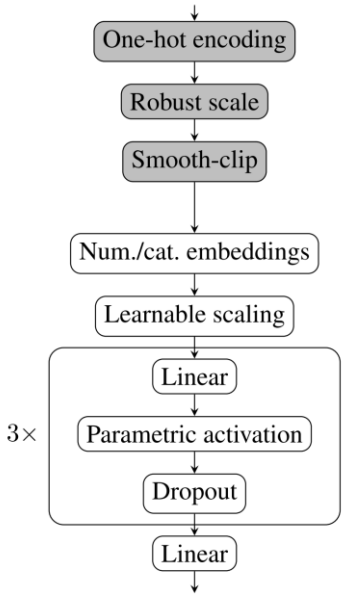
feature interaction



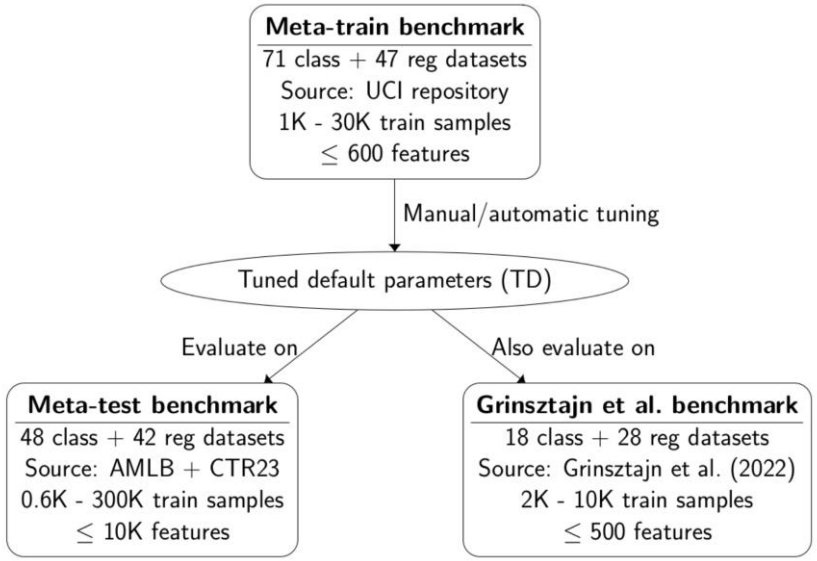
Jun-Peng Jiang, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, Han-Jia Ye. ***Representation Learning for Tabular Data: A Comprehensive Survey***. TPAMI 2026

RealMLP

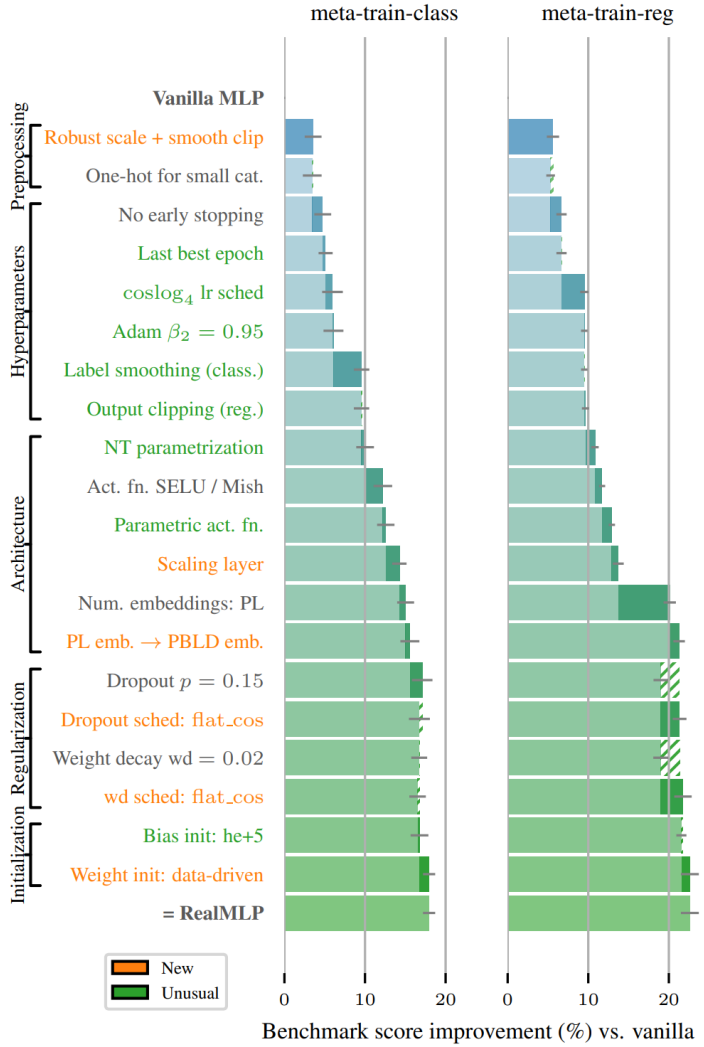
- A “better by default” model which can achieve performance comparable to the optimal parameter tuning results without hyperparameter search
- Improve traditional MLP from multiple perspectives such as structure, training, and hyperparameters
- Find a set of good default hyperparameters through meta-tuning



Architecture



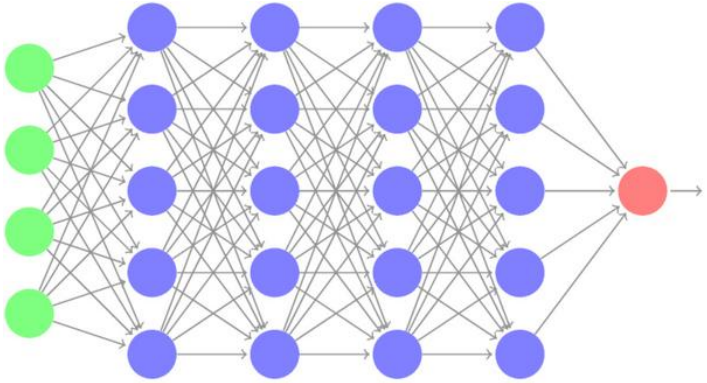
Meta-tuning



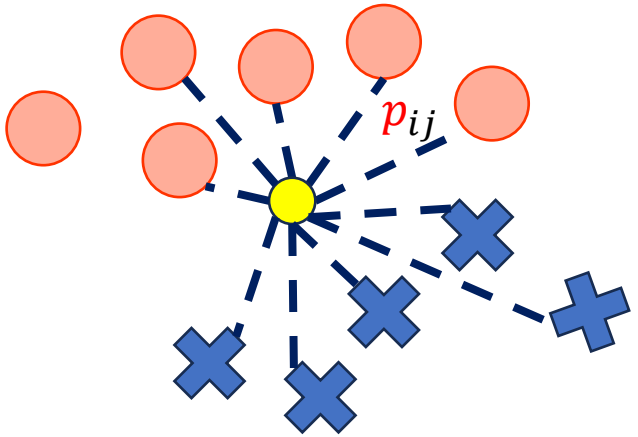
Improvements with tricks

David Holzmüller, Leo Grinsztajn, Ingo Steinwart. ***Better by default: Strong pre-tuned MLPs and boosted trees on tabular data***. NeurIPS 2024.

A Baseline for Specialized Model



Multilayer Perceptron (MLP)



Neighborhood Component Analysis (NCA)

[Goldberger et al., NIPS'04]

*Draw inspiration from traditional **parametric** methods*

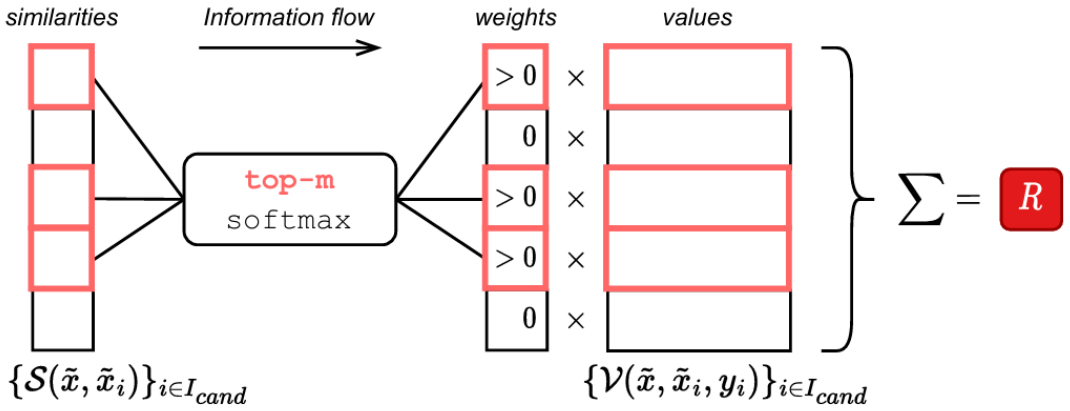
MLP (2021), MLP-PLR (2022), RealMLP (2024), …

*Could we draw inspiration from traditional **non-parametric** methods?*

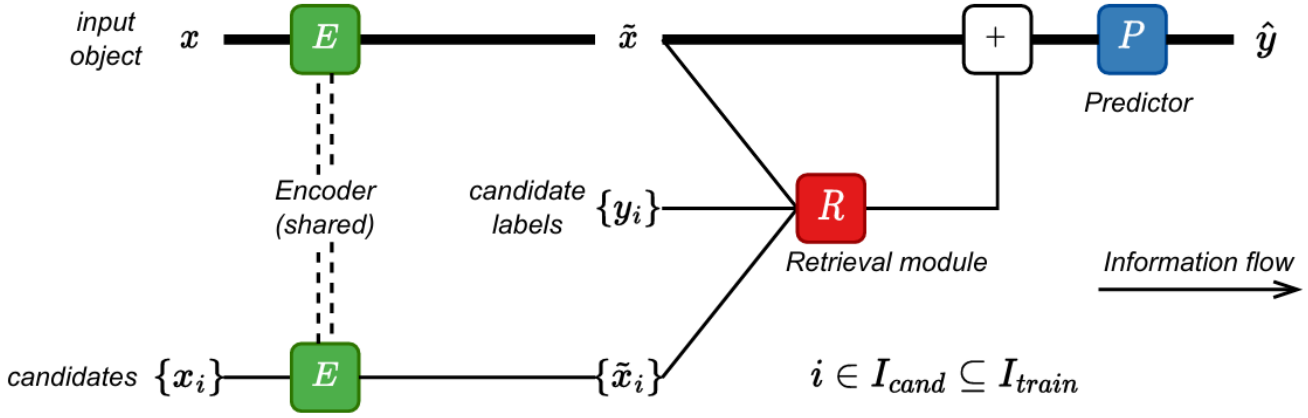
TabR

- TabR retrieves the m nearest neighbors with the highest similarity through the attention structure
- Calculate the contribution of the nearest neighbors, add it to the current sample features through residuals, and then make predictions through the prediction module (similar to FFN)

TabR achieves the use of other sample information to assist in prediction through a retrieval mechanism similar to attention, but the computational cost is relatively high.



The process of TabR retrieval and neighbor fusion



The overall process of TabR

Yury Gorishniy, Ivan Rubachev, Nikolay Kartashev, Daniil Shlenskii, Akim Kotelnikov, Artem Babenko. ***TabR: Tabular deep learning meets nearest neighbors***. ICLR 2024.

ModernNCA (MNCA)

- The learning objective: soft-NN

$$\hat{y}_i = \sum_{(\boldsymbol{x}_j, y_j) \in \mathcal{D}} p_{ij}\, y_j = \sum_{(\boldsymbol{x}_j, y_j) \in \mathcal{D}} \frac{\exp\left(-\mathrm{dist}^2\left(\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)\right)\right)}{\sum_{(\boldsymbol{x}_l, y_l) \in \mathcal{D}, \boldsymbol{x}_l \neq \boldsymbol{x}_i} \exp\left(-\mathrm{dist}^2\left(\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_l)\right)\right)} y_j$$

- Minimize the sum of negative log probability. Use soft-NN for prediction.

Architecture

- Set ϕ as MLP, and a single block is implemented by [Gorishniy et al., NIPS'21]
- One-hot encoding for categorical features
- PLR (lite) encoding for numerical features

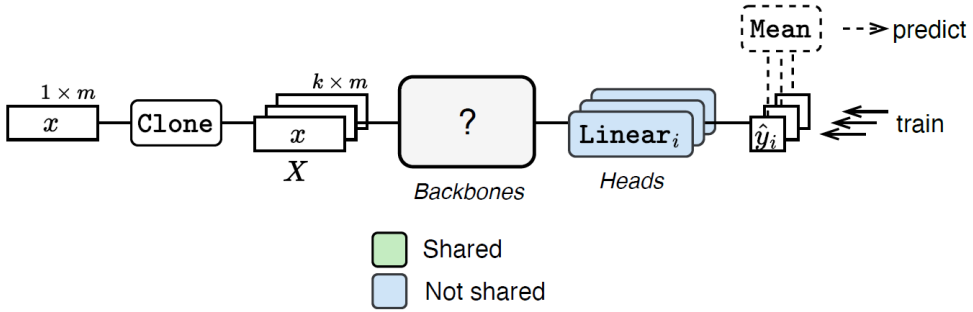
Acceleration

Sample *a subset of training set* in a mini-batch to act as the neighbor candidates, while use *the whole training set* during the inference.

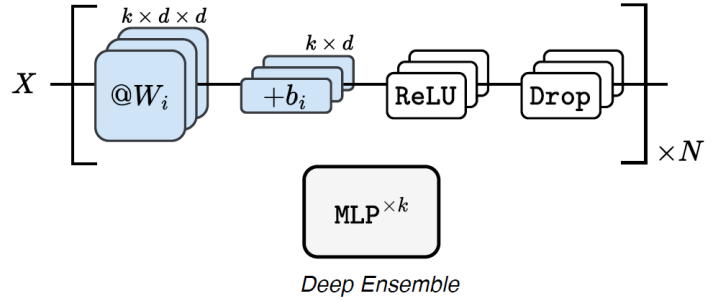
Han-Jia Ye, Huai-Hong Yin, De-Chuan Zhan, Wei-Lun Chao. ***Revisiting Nearest Neighbor for Tabular Data: A Deep Tabular Baseline Two Decades Later***. ICLR 2025.

TabM

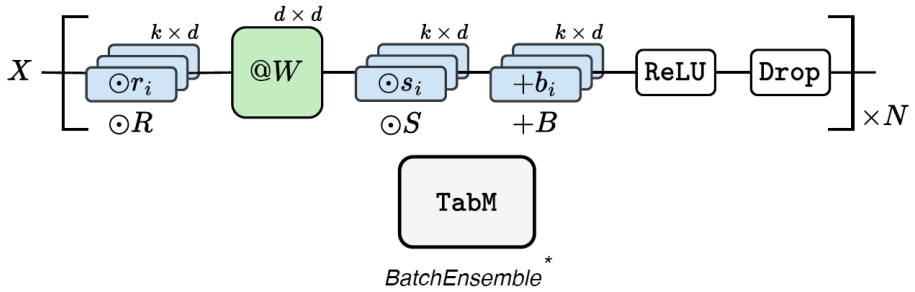
The MLP model was parameter-efficiently integrated through BatchEnsemble [Wen, Tran, Ba. ICLR'20]



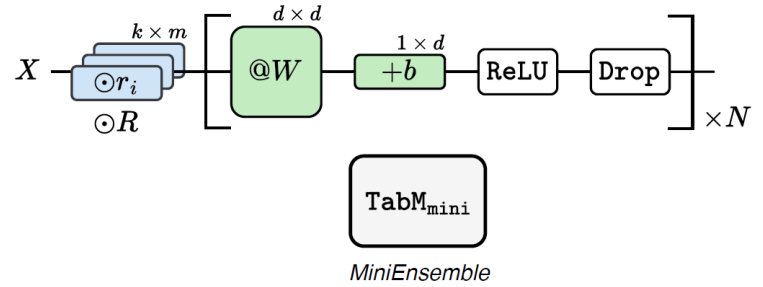
General integration framework, the input feature x is copied into k copied, each for different MLP heads, and the output is independently generated; When making predictions, the average of all outputs is taken, and during training, each sub-model BP independently.



Classic integration, each subnetwork is completely independent (k independent MLPs). During training, different representations are learned respectively, and the final prediction is the average of the outputs of these k networks



BatchEnsemble shares the backbone parameters and only injects a small number of "multiplicative additive adapters" R,S,B at each layer to achieve lightweight diversity.

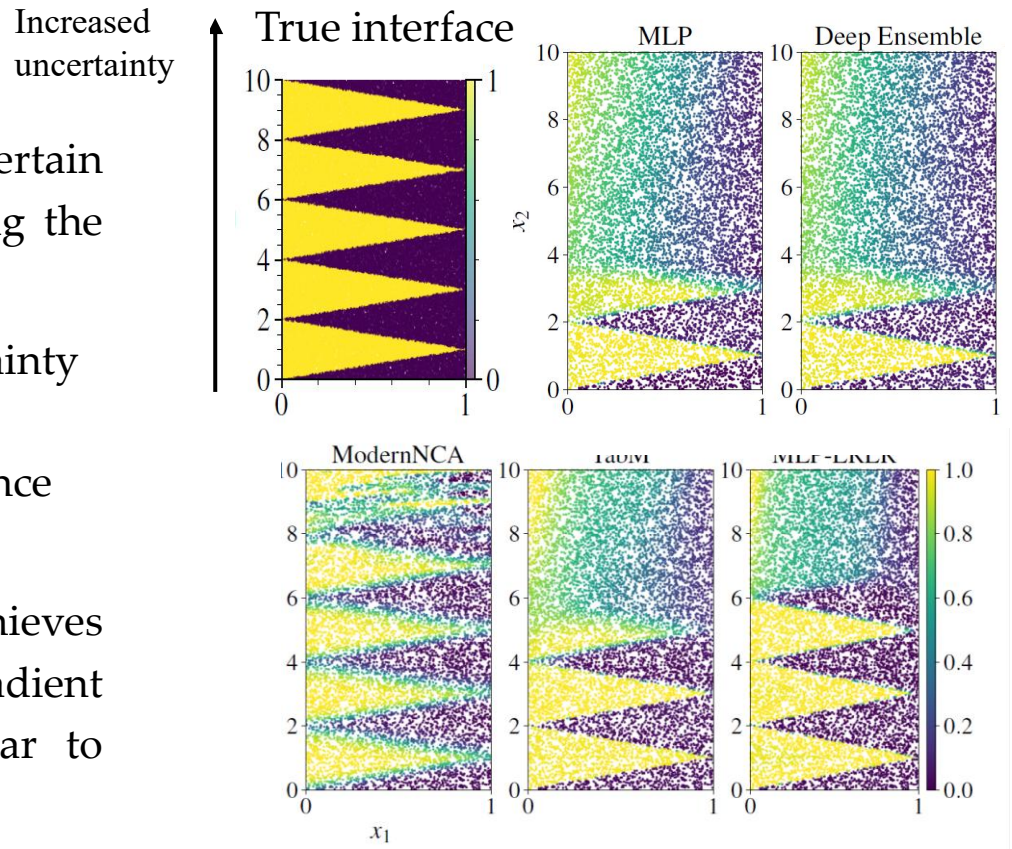


Further simplify BatchEnsemble, retaining only the first adapter R and removing S and B.

Yury Gorishniy, Akim Kotelnikov, Artem Babenko. ***TabM: Advancing Tabular Deep Learning with Parameter-Efficient Ensembling***. ICLR 2025.

Possible Reasons for the Improvement of DNNs

- Introducing “data uncertainty” as a key factor in explaining the performance of deep models, explaining the effects of numerical encoding (such as PLR), nearest neighbor methods (such as MNCA), and efficient integration (such as TabM), and demonstrating that their essence is to enhance the model's ability to cope with data uncertainty



- **PLR (*input smoothing*):** By reparameterization, the highly uncertain regions in the input space are made “locally smoothed”, enabling the model to learn to be more robust in these regions;
- **MNCA (*output smoothing*):** In the prediction stage, a “local uncertainty smoothing operator” was introduced to automatically perform neighborhood averaging on high-noise samples, reducing the variance caused by data uncertainty;
- **TabM (*gradient smoothing*):** Efficient parameter integration achieves implicit regularization of uncertain samples through the “gradient averaging” mechanism of sharing backbone parameters (similar to gradient noise filters).

Nikolay Kartashev, Ivan Rubachev, Artem Babenko. ***Unveiling the Role of Data Uncertainty in Tabular Deep Learning***. CoRR 2025.

RFM

Main observation: Neural Feature Matrices (NFM) is proportion to AGOP

Average Gradient Outer Product

$$W_i^{(\ell)\top} W_i^{(\ell)} \propto \left(\frac{1}{m} \sum_{i=1}^{m} \left(\sum_{j=1}^{k} \nabla_{u_{ij}^{(\ell)}} \hat{f}\left(x^{(p)}\right) \nabla_{u_{ij}^{(\ell)}} \hat{f}\left(x^{(p)}\right)^T \right) \right)^{\alpha}$$

l is the layer index, m is the number of training examples, k is the number of input neurons for a certain input example at the l-th layer

- Recursive Feature Machines (RFM) mainly adopts the form of dual linear regression. By setting the $\boldsymbol{M}_1 = \boldsymbol{I}$, it iterates the following two steps:

Update f

$$f_t(\boldsymbol{x}) = \kappa(\boldsymbol{M}_t\boldsymbol{x}, \boldsymbol{X}\boldsymbol{M}_t)\boldsymbol{\alpha}_t$$

$$\boldsymbol{\alpha}_t = [\kappa(\boldsymbol{X}\boldsymbol{M}_t, \boldsymbol{X}\boldsymbol{M}_t) + \lambda \boldsymbol{I}]^{-1}\boldsymbol{y}$$

Update $\boldsymbol{M}$

$$\boldsymbol{M}_{t+1} = [\mathrm{AGOP}(f_t(\boldsymbol{M}_t\boldsymbol{x}), \boldsymbol{X})]^c$$

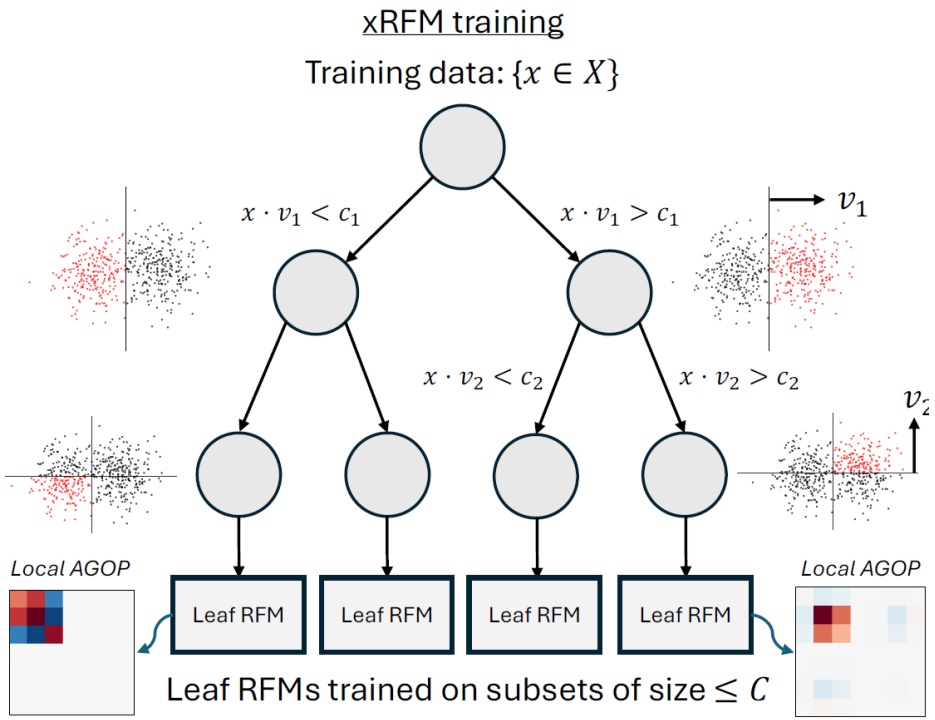
$$c \in \left\{\frac{1}{4}, \frac{1}{2}\right\}$$

Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, Mikhail Belkin. ***Mechanism for feature learning in neural networks and backpropagation-free machine learning models***. Science 2024.

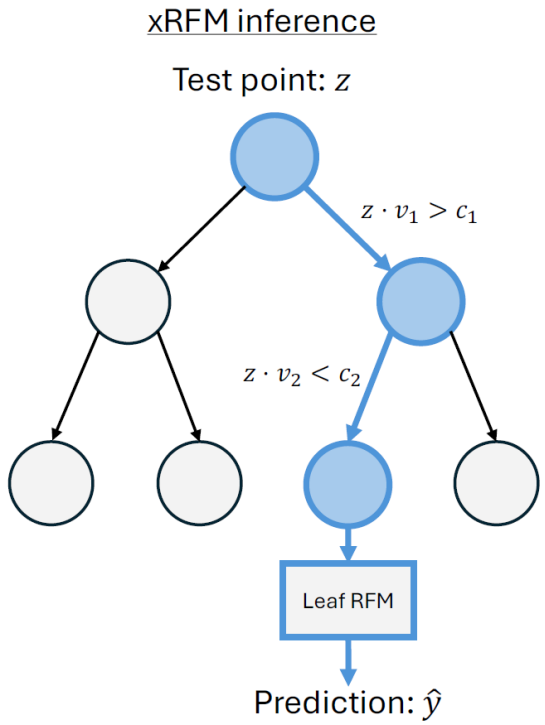
xRFM

- xRFM is an extension of RFM, where additional kernel function is used, and a tree-like architecture is used to apply RFM in a local manner

training phase



test phase



Daniel Beaglehole, David Holzmuller, Adityanarayanan Radhakrishnan Mikhail Belkin. ***xRFM: Accurate, scalable, and interpretable feature learning models for tabular data***. CoRR'25.

Tabular Foundation Models

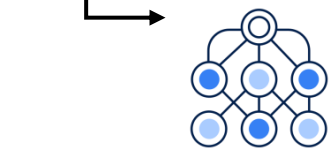
Language Foundation Models

Vision Foundation Models

Video Foundation Models

Mileage	Year	Brand	Condition	Price
45000	2018	BMW	Excellent	18500
82000	2015	Ford	Good	12000
120000	2012	Honda	Fair	7500
25000	2020	BMW	Excellent	?

Income	Credit	Status
25	Low	Denied
51	High	Approved
38	Medium	?



Price= 32,000

Status= Approved

How to deal with heterogeneous tabular prediction tasks (with varying size)

Pre-training Methods for Tabular Data

- Explore whether pre-training techniques in the fields of images and text can assist in tabular prediction tasks

Pre-training objective

Pre-training on unsupervised data and further fine-tuning on supervised data [1-3].

- Contrastive learning: InforNCE loss
- Reconstruction: Predict the damaged features;
- Mask Prediction: Predict which columns will be masked.

Data augmentation methods

Referring to graph and text, design an augmentation method for structured tabular data [1-3].

- CutMix or Mixup. The original input/embedding layer randomly exchanges some features:

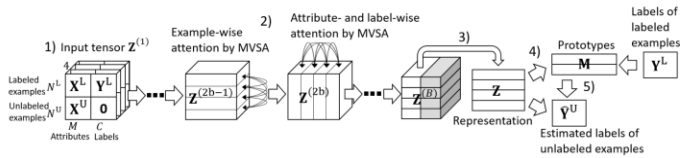
$$x_i' = x_i \odot m + x_a \odot (1 - m)$$

- Random Feature Corruption

$$\tilde{x}_{ij} = \begin{cases} x_{ij}, & j \notin I_i \\ v, v \sim p(x_j), & j \in I_i \end{cases}$$

Model architecture design

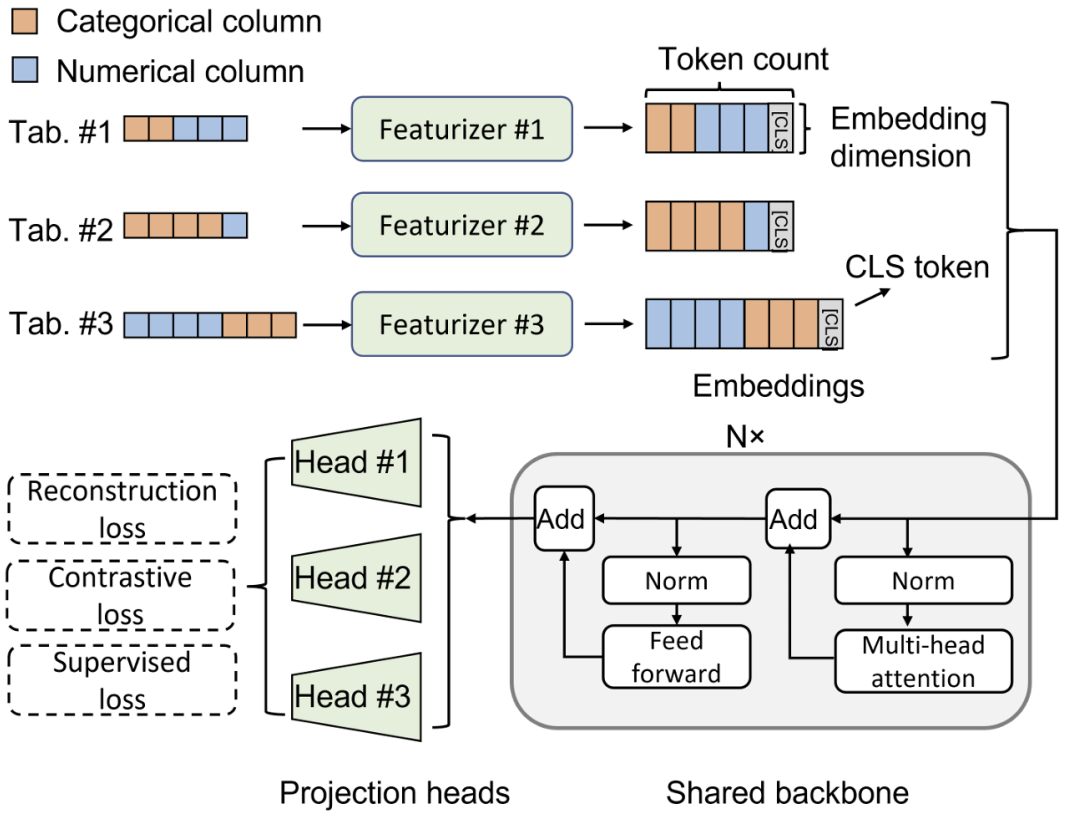
By using the row-column attention mechanism [1], the correlation between samples and attributes can be enhanced, and even the limitations of heterogeneous attribute spaces can be broken through [4].



[1] Gowthami Somepalli et al., ***SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training***. CoRR 2021.
[2] Ivan Rubachev et al., ***Revisiting Pretraining Objectives for Tabular Deep Learning***. CoRR 2022.
[3] Dara Bahri et al., ***Scarf Self-Supervised Contrastive Learning using Random Feature Corruption***. ICLR 2022.
[4] Tomoharu Iwata, Atsutoshi Kumagai. ***Meta-learning of semi-supervised learning from tasks with heterogeneous attribute spaces***. CoRR 2023.

XTab

Based on the architecture of FT-T. XTab learns a sharable Transformer across heterogeneous tabular tasks, wher task-specific tokenizer and head are tuned for each task.



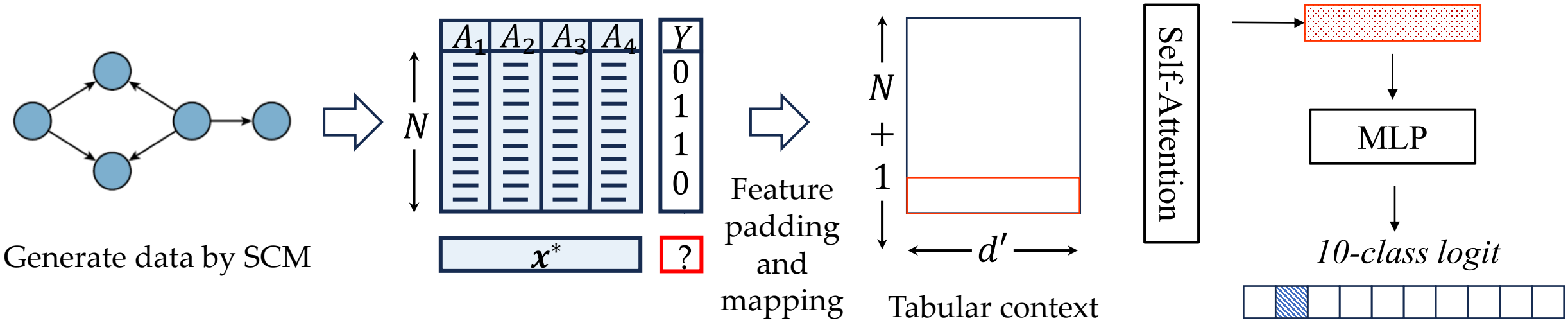
- The learned transformer may capture the shared experience to aggregate information among features across tabular tasks
- Task-specific fine-tuning still requires a certain amount of instances given a target task

Could we share all parameters across tabular tasks, and deploy the TFM without fine-tuning?

Bingzhao Zhu, Xingjian Shi, Nick Erickson, Mu Li, George Karypis, Mahsa Shoaran. ***XTab: Cross-table Pretraining for Tabular Transformers***. ICML 2023.

Tabular Foundation Model: TabPFN

The pre-training and prediction process of TabPFN:



- TabPFN v1 uses SCM to synthesize a large dataset for pre-training;
- For small-scale data, it can only handle classification problems within 10 categories, 3000 samples and 100 features;
- All datasets are uniformly padded with zeros to 100 dimensions, and each sample is regarded as a token in the token sequence;
- After passing through 12 layers of Transformer Encoder, the model completes the prediction of the test samples through in-context learning;
- The model does not require parameter adjustment. Similar to the nearest neighbor method, the training process is not displayed.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, Frank Hutter. ***TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second***. ICLR 2023.

“Bias-variance” analysis of PFN

The relative percentage change (%) of bias and variance with respect to TabPFN; A negative value indicates a decline.

Metric	Dataset	BETA	BETA-S	KnnPFN	LocalPFN	TabPFN-Finetune	TabPFN-Bagging
Bias	Adult	**-4.40**	-3.60	-0.77	-1.68	-2.69	+1.73
	Bank	**-1.65**	-1.19	-1.33	-1.47	-1.05	+0.53
Variance	Adult	**-16.23**	+46.99	+4.99	+18.52	+39.27	-9.91
	Bank	**-36.57**	+27.12	+27.49	+2.06	+10.35	-9.32

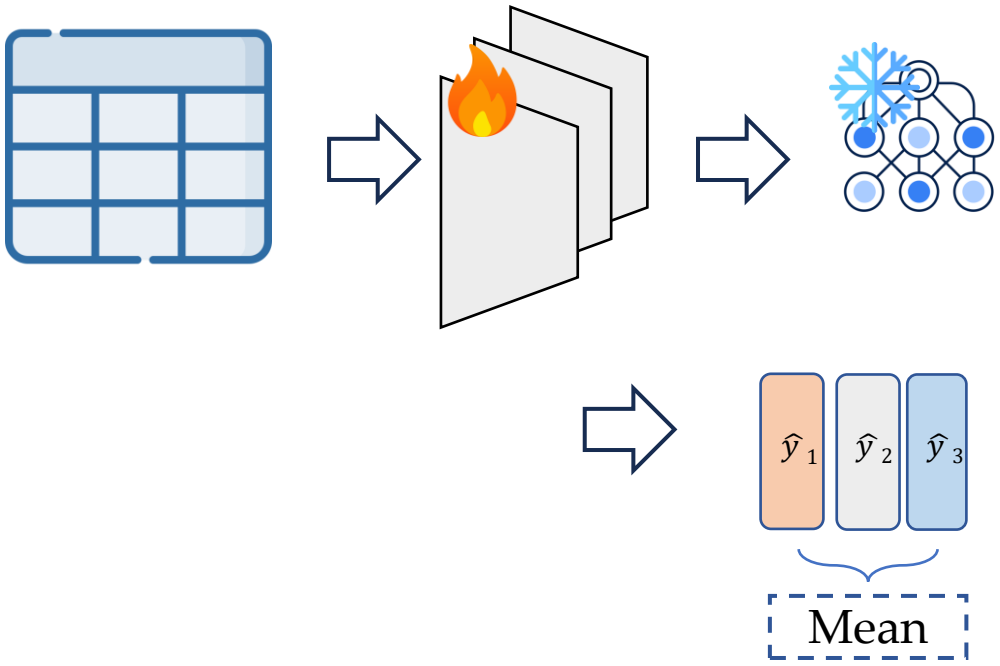
	BETA (Ours)	TabPFN	TuneTables	TabForestPFN	LocaLPFN	MixturePFN
Scales to Large Datasets	✓	✗	✓	✗	✓	✓
Handles High-Dimensional Data	✓	✗	✗	✗	✗	✗
Adapts to More Than 10 Classes	✓	✗	✓	✗	✗	✗
Reduces Bias	✓	✗	✓	✓	✓	✓
Reduces Variance	✓	✗	✗	✗	✗	✗
No Additional Inference Cost	✓	✓	✓	✓	✗	✗
Fine-Tuning	lightweight encoder	✗	prompt & backbone	backbone	backbone	adapter

Si-Yang Liu, Han-Jia Ye. ***TabPFN Unleashed: A Scalable and Effective Solution to Tabular Classification Problems***. ICML 2025.

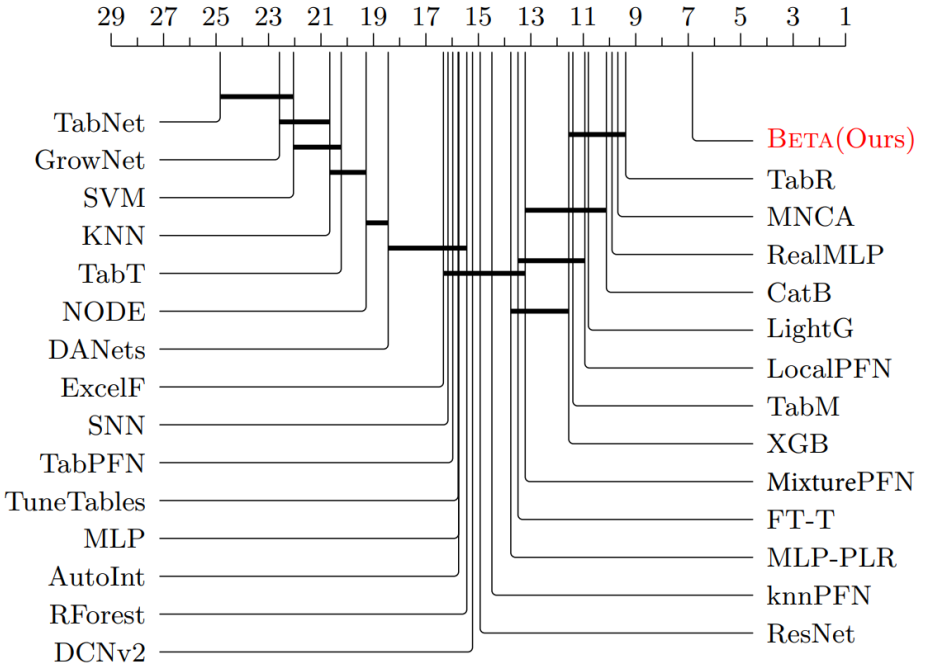
Efficient Adaptation of TabPFN

BETA: **B**agging and **E**ncoder-based Fine-tuning for **T**abPFN **A**daptation

- Fine-tune multiple encoders with batch ensemble
- Bagging at the inference stage



Results on classification datasets



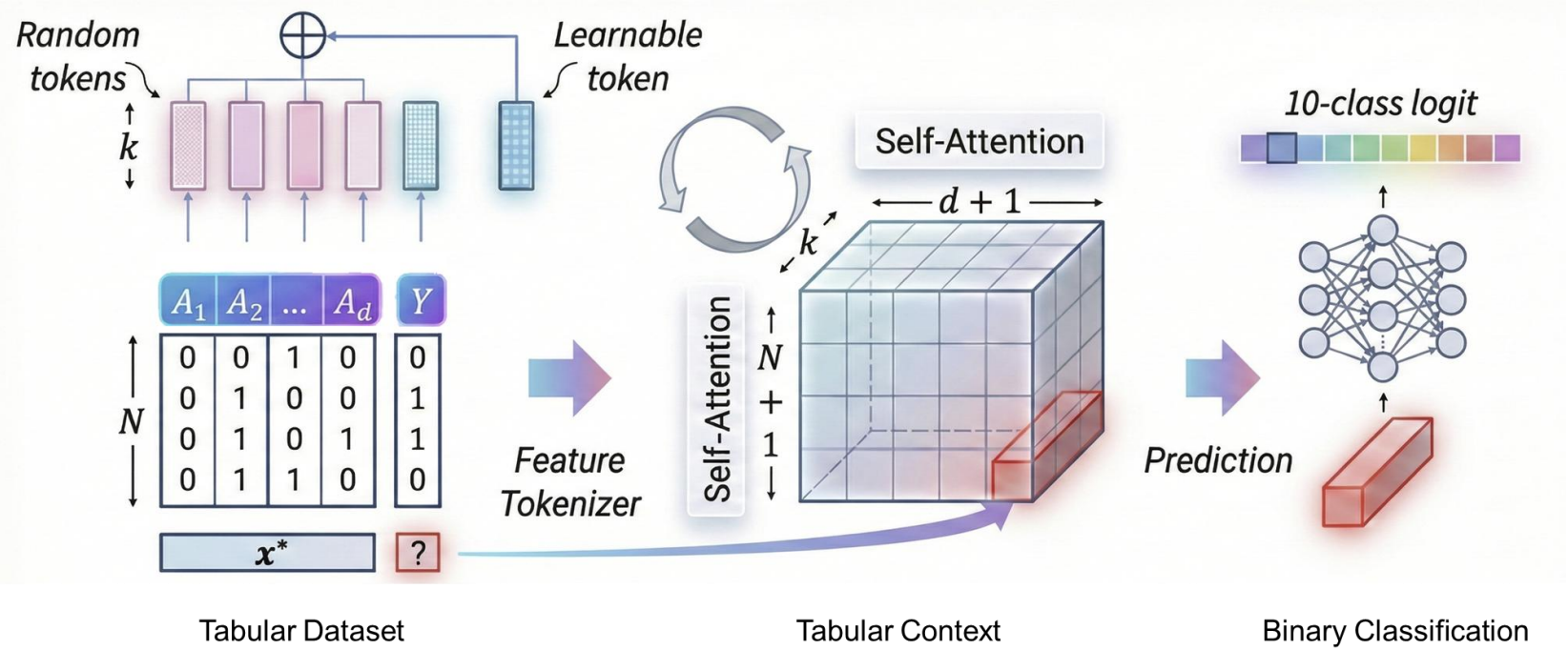
Results on benchmark with datasets < 10 classes

Si-Yang Liu, Han-Jia Ye. ***TabPFN Unleashed: A Scalable and Effective Solution to Tabular Classification Problems***. ICML 2025.

TabPFN v2

The pre-training and prediction process of TabPFN v2:

- TabPFN v2: Introducing tree-based SCM enriches the data generation methods;
- Be capable of handling classification and regression problems simultaneously; The sample size is expanded to 10,000, and the feature dimension is expanded to 500



The units of each dataset are mapped to a unified dimension, and the model adopts ***an alternating row-column attention*** approach

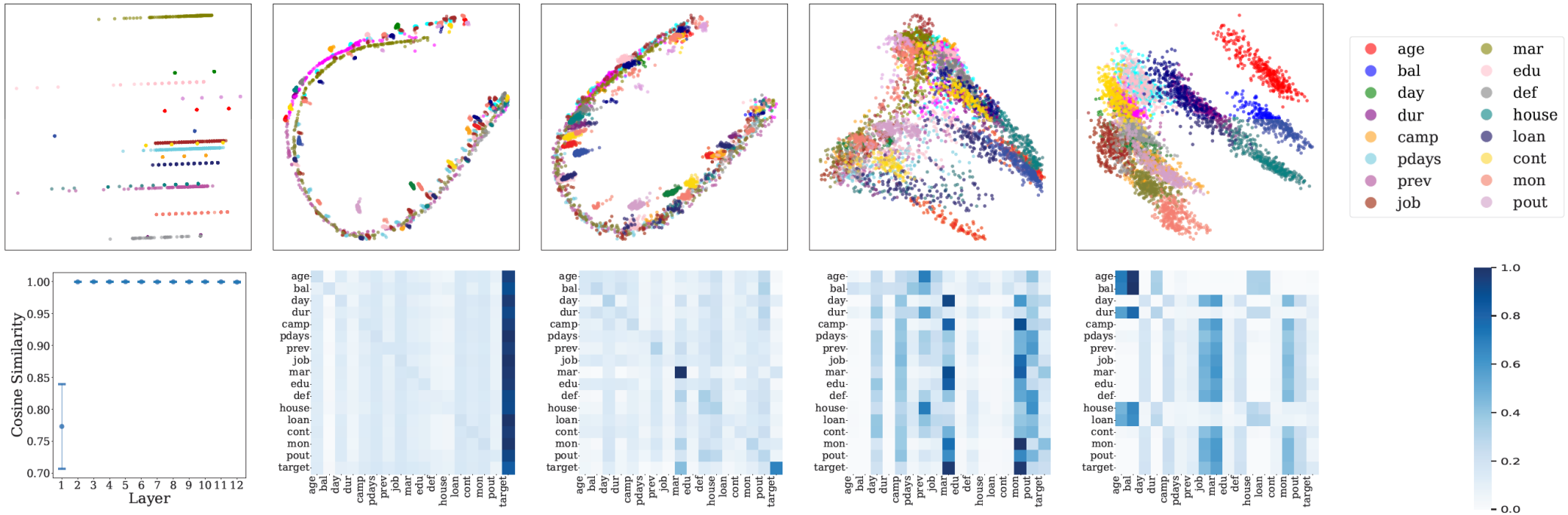
Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, Frank Hutter.
Accurate predictions on small data with a tabular foundation model. Nature 2025.

Random Token Analysis of TabPFN v2

- TabPFN v2 uses randomized tokenization (shared vector + attribute-specific perturbations) to handle heterogeneity, avoiding predefined semantics or task-specific tokens for direct pre-trained application

$$\left[x_i^1 \cdot \boldsymbol{u} + \boldsymbol{r}_1, \ldots, x_i^d \cdot \boldsymbol{u} + \boldsymbol{r}_d, \tilde{\boldsymbol{y}}_i\right] \in \mathbb{R}^{k \times (d+1)}$$

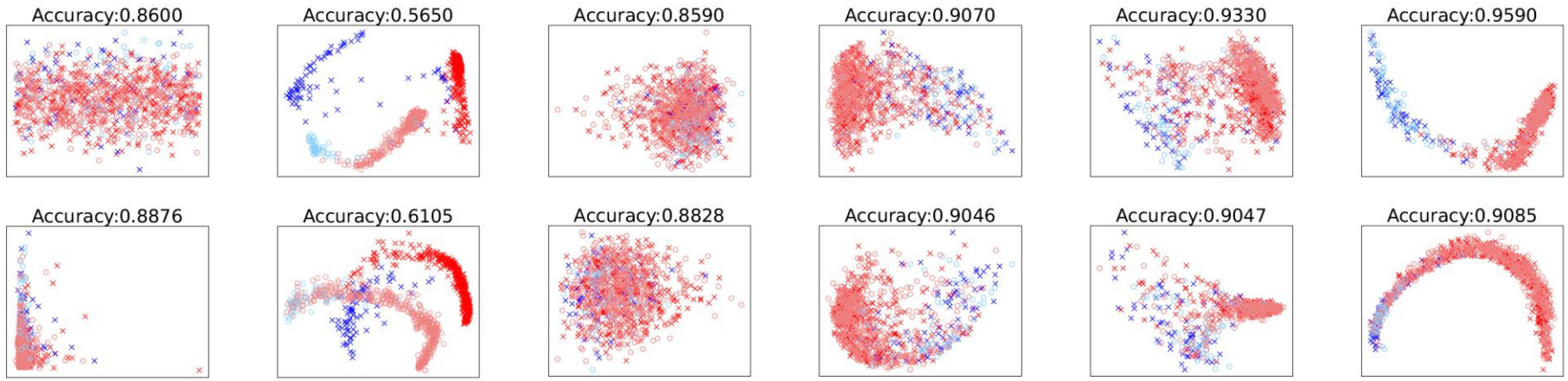
- Despite initial randomness, it infers meaningful attribute relationships via in-context learning, though it leverages statistical dataset structure rather than fixed semantics.



(a) (b) Layer 3 (c) Layer 6 (d) Layer 9 (e) Layer 12 (f) Legend

TabPFN v2 as an Embedding Extractor

- TabPFN v2 simplifies the feature distribution among categories
- Due to the different roles of labels in training and testing, the embedding of samples is difficult to extract directly
- A leave-one-fold-out strategy can be used to extract the embedding



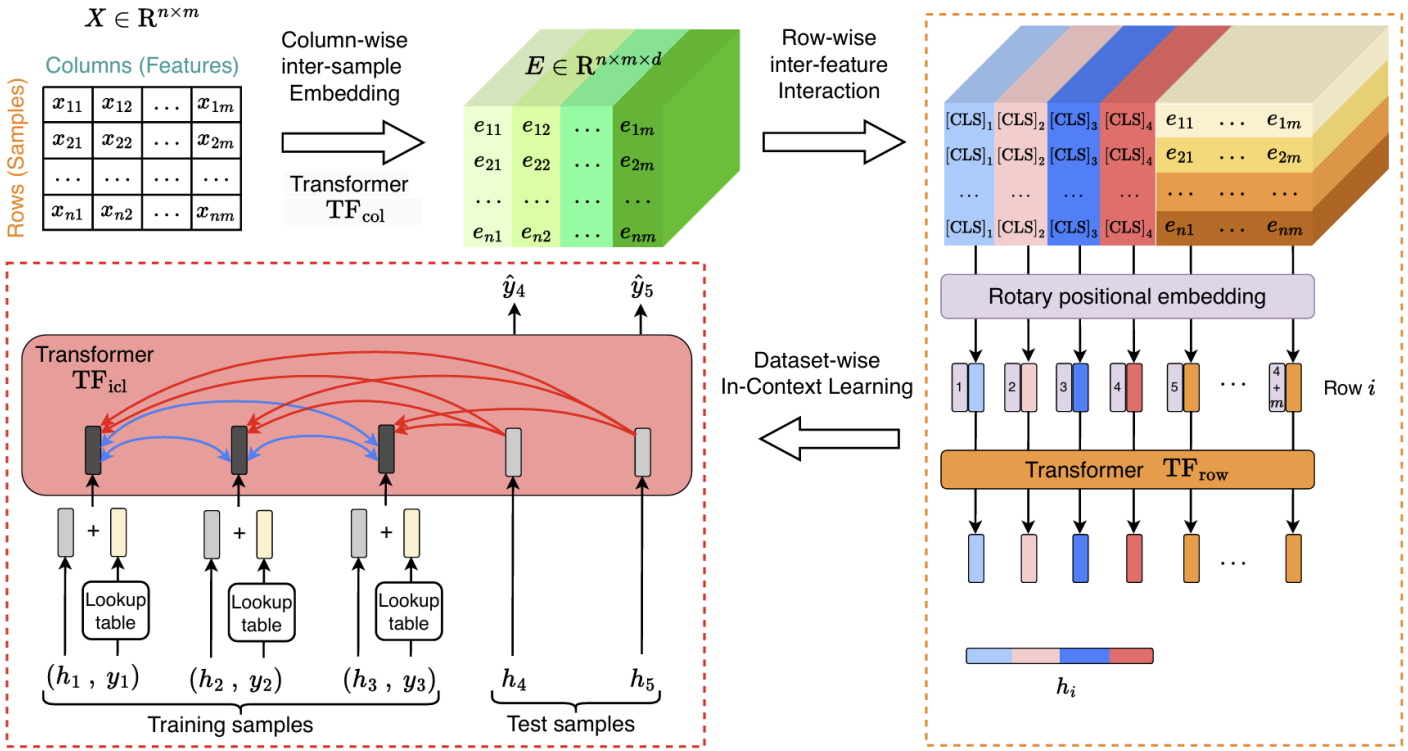
(a) Raw Feature (b) Vanilla (c) Layer 3 (d) Layer 6 (e) Layer 9 (f) Layer 12

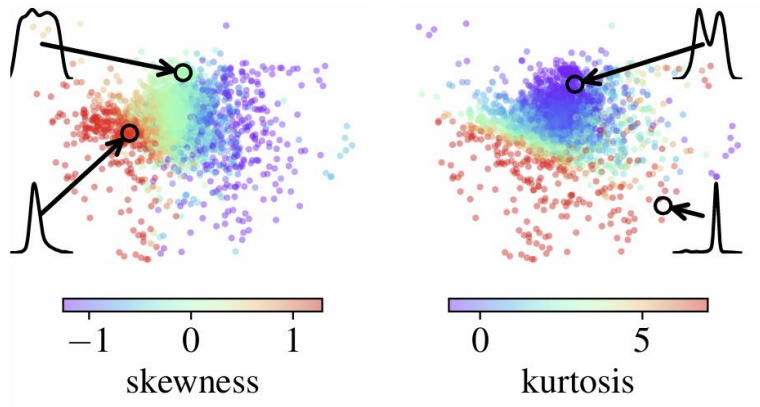
↓	TabPFN v2	Vanilla	Layer 6	Layer 9	Layer 12	Combined
Rank	2.69	5.97	4.28	4.00	2.12	**1.94**

Comparison of the average rank of TabPFN v2 and the linear classifier trained based on extracting embeddings on 29 classification datasets

TabICL

- Instead of using column/row-wise transformer alternatively, TabICL proposes a column-then-row manner to apply the transformer
- A feature-wise set transformer is used to extract feature information and then a row-wise transformer is used to align the instance embeddings with labels. RoPE is used for feature positional encoding



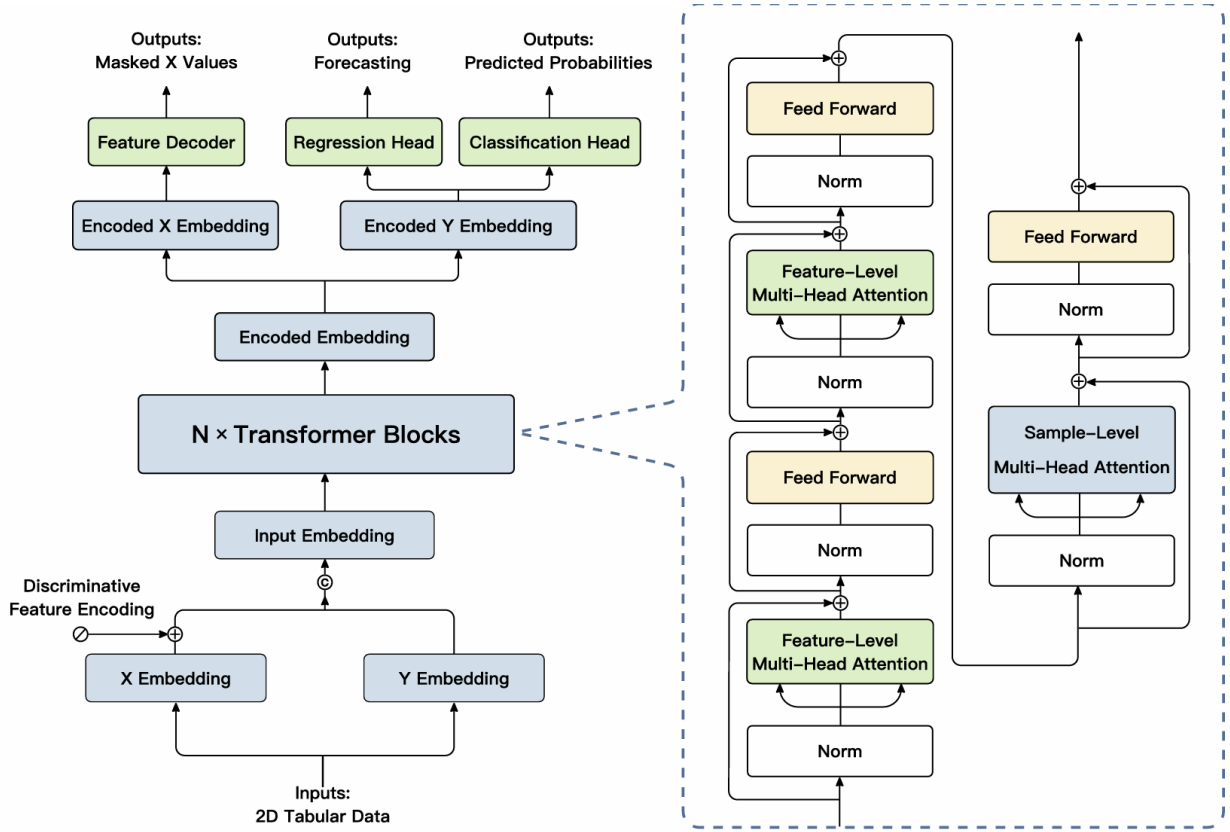


Learned column wise embeddings encode statistical properties of feature distributions.

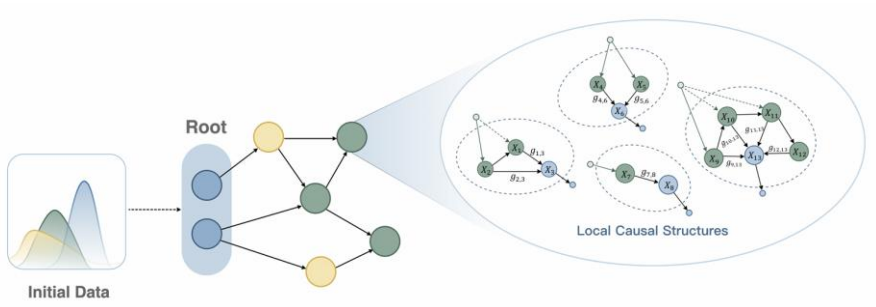
Jingang Qu, David Holzmüller, Gaël Varoquaux, Marine Le Morvan. ***TabICL: A Tabular Foundation Model for In-Context Learning on Large Data***. ICML 2025.

LimiX

LimiX instantiate Large Structured-data Models (LDMs) and mainly follows the architecture of TabPFN v2, where alternative attentions are used to encoder row/column-wise information.



Masked training objective is used together with classification/regression loss.



Hierarchical SCMs are applied to generate synthetic datasets.

Xingxuan Zhang et al. ***LimiX: Unleashing Structured-Data Modeling Capability for Generalist Intelligence***. CoRR'25.

Outline

- Introduction to tabular data tasks and evaluation methods
- From classic tabular models to deep tabular models
- From specialized tabular models to general tabular models
- The preliminary evaluation and analyses of the tabular prediction models

Tabular Benchmarks

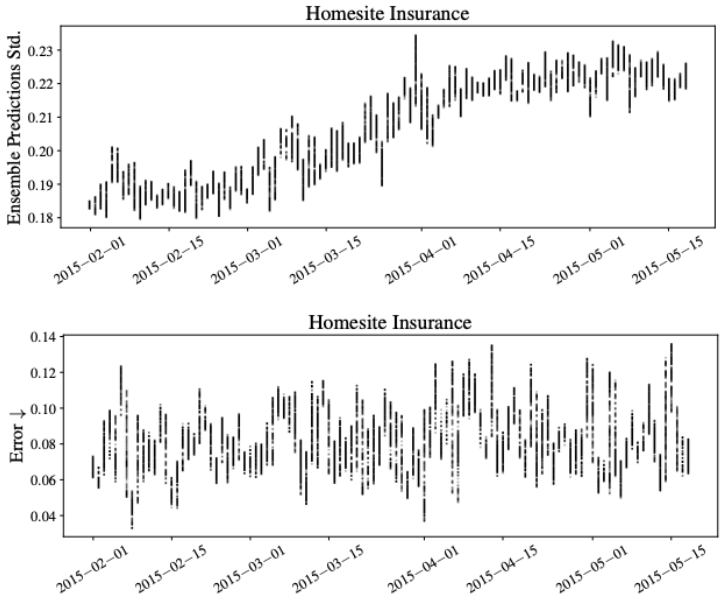
Benchmark	Datasets	Methods	Remarks
[1]	11	11	Only 11 datasets with large amounts of data.
[2]	45	7	Only a few methods and the attributes of the dataset are limited.
TabZilla	176	19	Evaluations were conducted on more difficult and other datasets (only classification).
TALENT	300	35+	A large number of common-size tabular datasets and a special set of datasets in TALENT-extension.

[1] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, Artem Babenko. ***Revisiting deep learning models for tabular data***. NeurIPS 2021
[2] Léo Grinsztajn, Edouard Oyallon, Gaël Varoquaux. ***Why do tree-based models still outperform deep learning on typical tabular data***? NeurIPS 2022
Han-Jia Ye, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, De-Chuan Zhan. ***A Closer Look at Deep Learning Methods on Tabular Data***. CoRR 2024

Other Benchmarks

Benchmarks with distribution shift:

- TabReD took into account the **changes in the distribution** of the dataset during training and testing



- The way industrial data is collected will affect the final prediction results

Benchmarks with semantics:

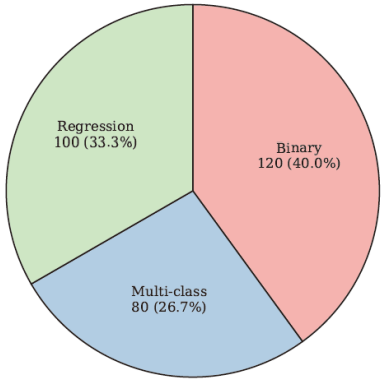
- Attribute name, task description…
- TabLib has collected 627M tables and 867M text tokens introducing these tables [Eggert et al., CoRR'23]
- CM2 proposed the OpenTabs dataset pre-trained for cross-tables, which contains semantic information about the column names of a large number of large-scale tables [Ye et al., WWW'24]
- T4 took into account the difficult-to-understand statistics and tables containing personal identity information in TabLib [Gardner et al., NeurIPS'24]

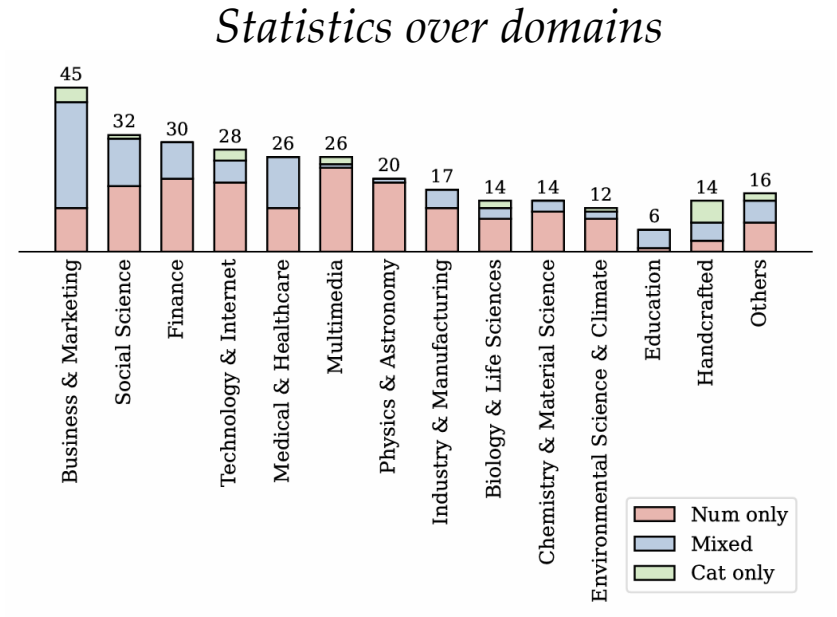
Ivan Rubachev, Nikolay Kartashev, Yury Gorishniy, Artem Babenko. ***TabReD: Analyzing Pitfalls and Filling the Gaps in Tabular Deep Learning Benchmarks***. ICLR 2025

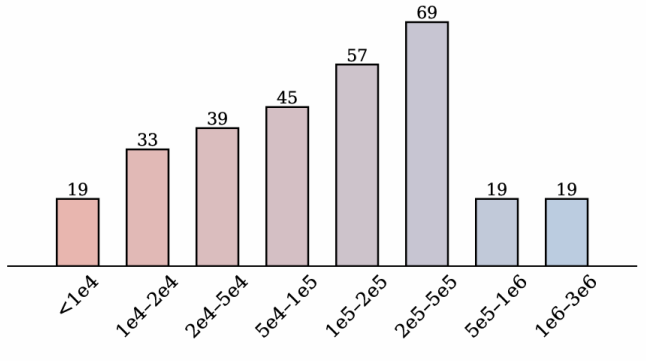
TALENT Benchmark

Evaluations on 300 tabular datasets

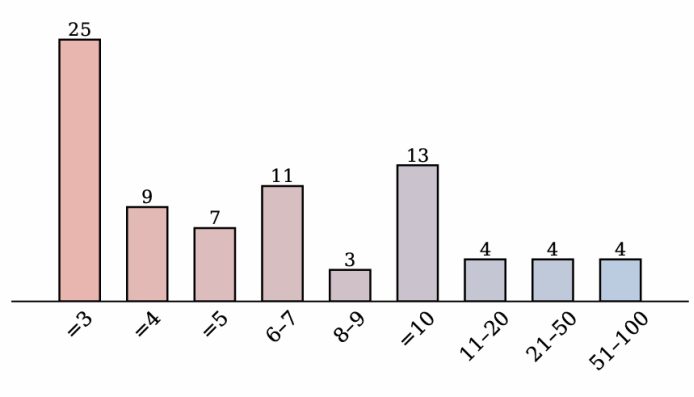
- Binary Classification: 120
- Multi-Class Classification: 80
- Regression: 100







Statistics over data sizes



Statistics of classes in multi-class tasks

Han-Jia Ye, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, De-Chuan Zhan. ***A Closer Look at Deep Learning Methods on Tabular Datasets***. CoRR 2024.

Tabular Benchmarks

In different benchmarks, there are generally different ways of data splitting and selection of hyperparameters

Data Split		
Split Method	**TabArena**	**TALENT**
Split the fixed data set and conduct multiple evaluations	-	• 64% train 16% validate 20% test • Take the average value after running 15 times with different seeds
Cross validation	• For datasets with a sample size of less than 2,500, repeat the 3-fold cross-validation 10 times. • Repeat 3 times for other datasets	-
Hyperparameters		
TabArena	• Except for some models, conduct a default parameter evaluation for the remaining models and then perform 200 random parameter searches • Limit the evaluation time for a group of parameters to no more than one hour	
TALENT	• Without considering the time limit, search for 100 times in the parameter space and take the optimal parameter	

Nick Erickson, et al. ***Tabarena: A living benchmark for machine learning on tabular data***. NeurIPS 2025.

Tabular Toolbox

LAMDA
Learning And Mining from DatA

TALENT

Arxiv 2407.04057 | 中文解读 | TALENT | Stars 723 | last commit september
PYTORCH 2.0.1 | PYTHON 3.10 | docs passing | license MIT

TALENT: A Tabular Analytics and Learning Toolbox

[Paper] [中文解读] [Docs]

https://github.com/LAMDA-Tabular/TALENT

- Unified interface
- Customizable methods

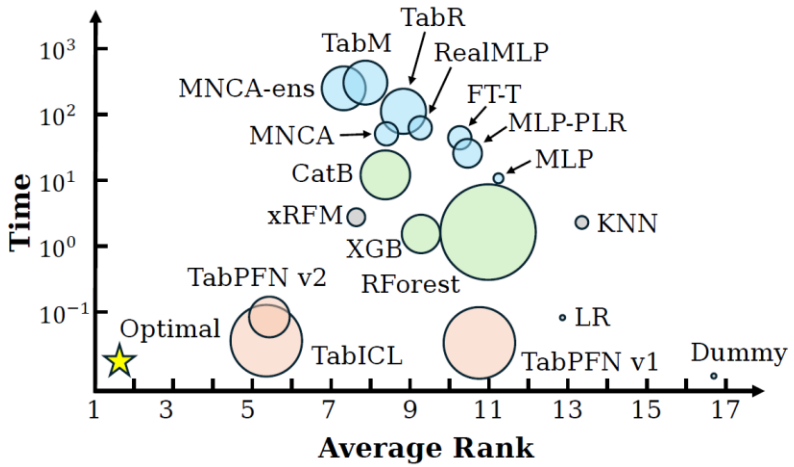
It includes over 35 deep learning methods (including new ones such as NeurIPS'25/ICLR'25/ICML'25).

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort et al. ***Scikit-learn: Machine Learning in Python***. JMLR 2011.
Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, Han-Jia Ye. ***TALENT: A Tabular Analytics and Learning Toolbox***. JMLR 2025.

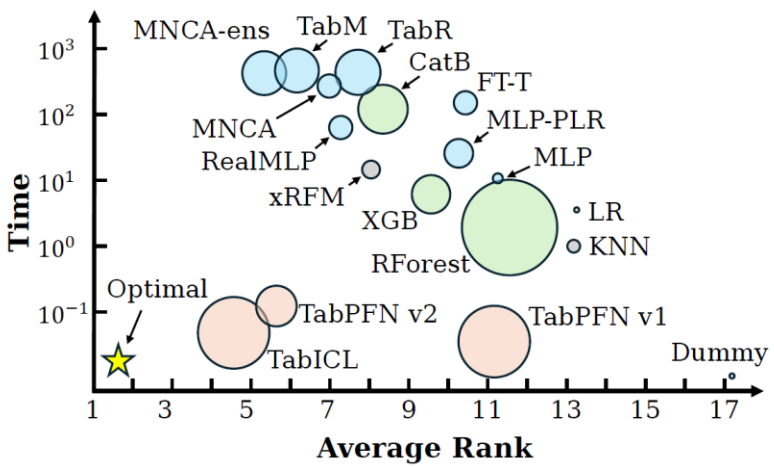
Evaluation Results

Evaluation results on 300 datasets: Average Rank vs. Time cost vs. model size

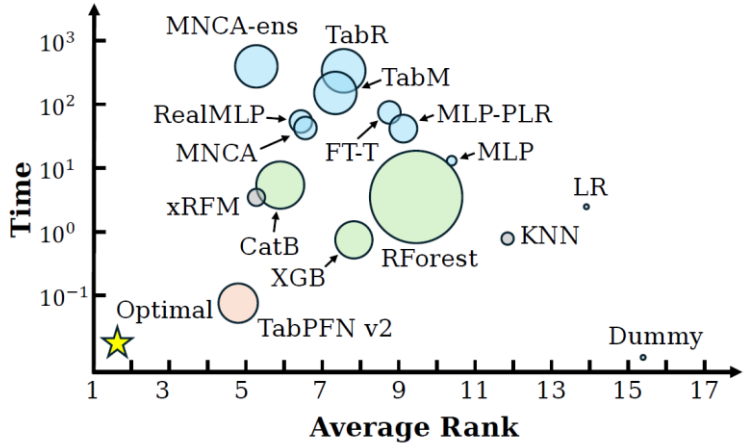
- Deep learning methods can be on par with or even surpass those of tree model methods
- Compared with training models separately for each task, pre-trained models have stronger capabilities (and lower costs)
- Ensemble learning methods can further enhance capabilities



Binary Classification



Multi-Class Classification



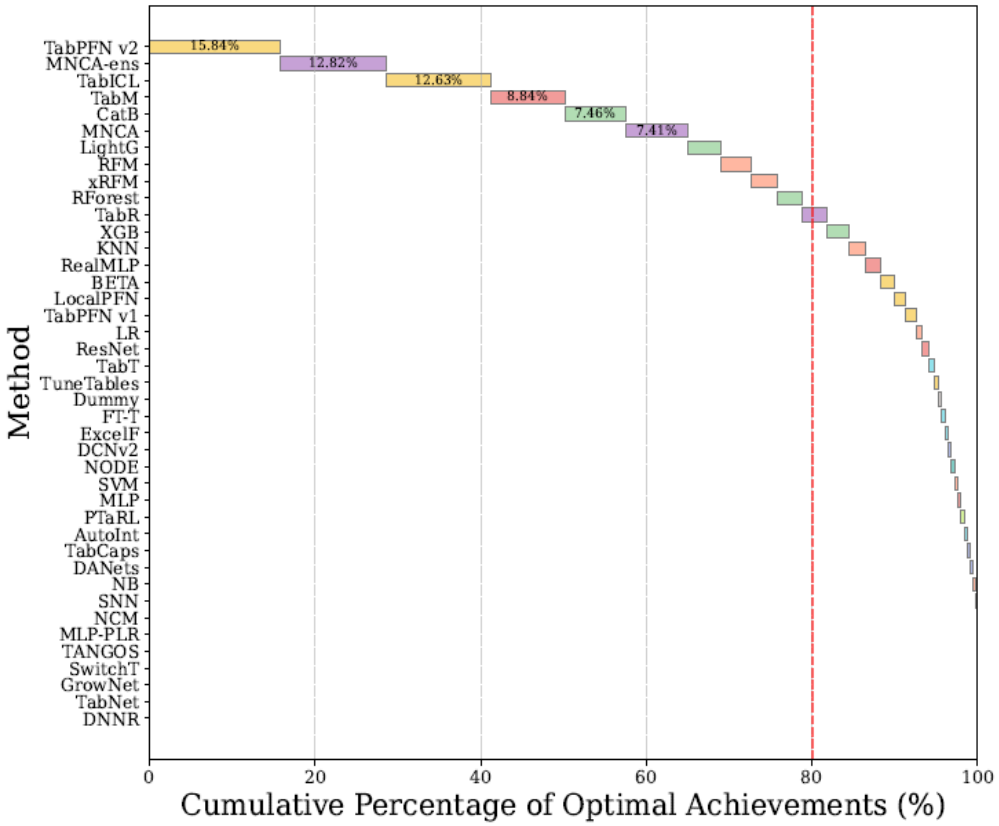
Regression

Han-Jia Ye, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, De-Chuan Zhan. ***A Closer Look at Deep Learning on Tabular Data***. CoRR 2024.

Evaluation Results

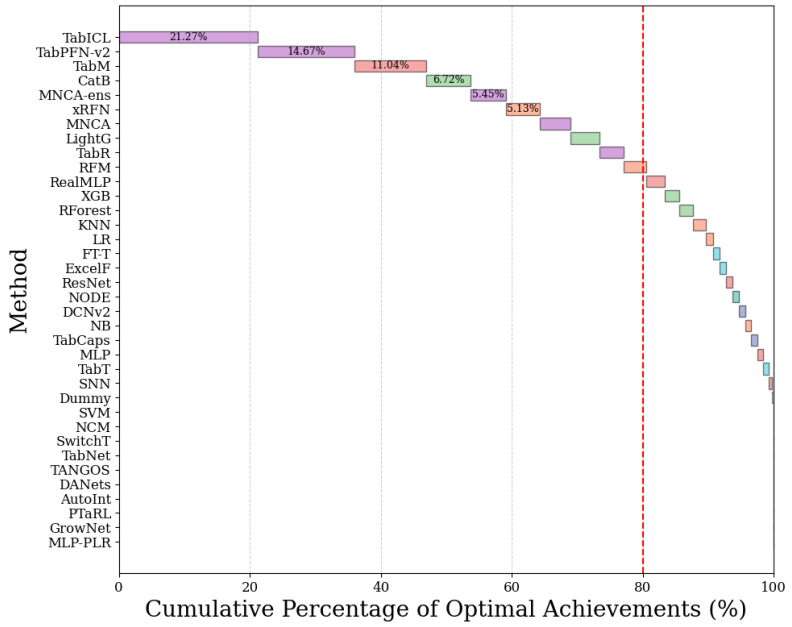
PAMA results

- Pre-trained models have strong capabilities
- Classical ML methods still have stable performance on some tasks
- Some deep learning methods combined with Ensemble can achieve better performance
- A practical candidate set for tabular prediction: *TabICL, TabPFN v2, MNCA-ens, MNCA, CatBoost, LightGBM, TabM,* classical methods such as LR and KNN may be better sometimes

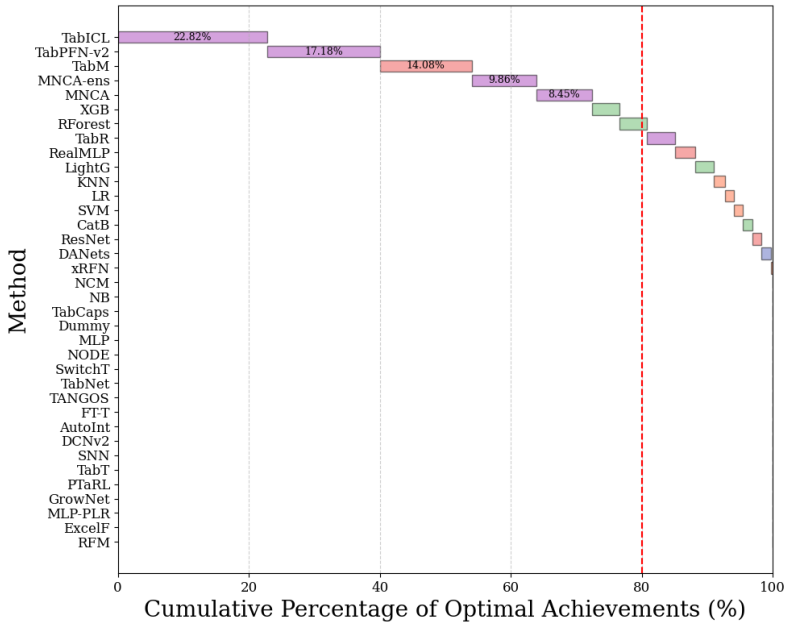


Results: PAMA

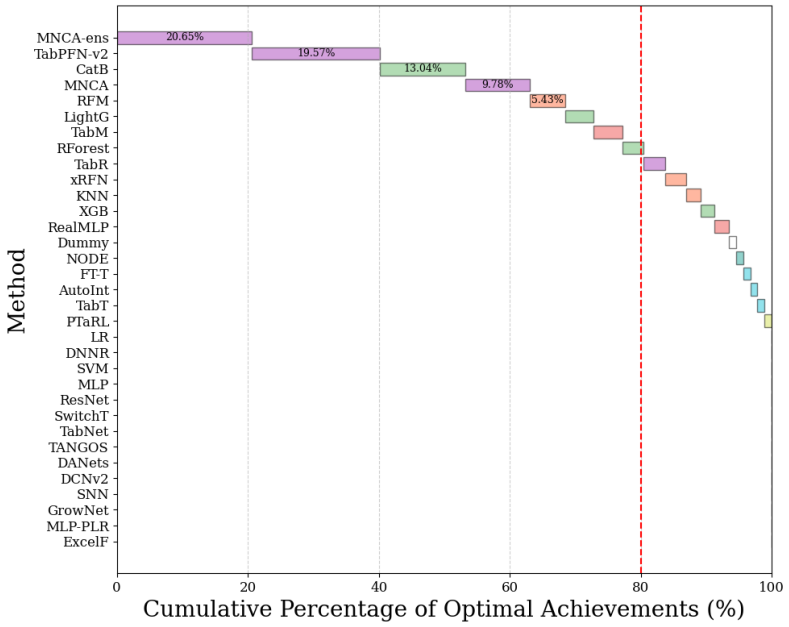
Updated PAMA results with tabular foundation models.



Binary Classification



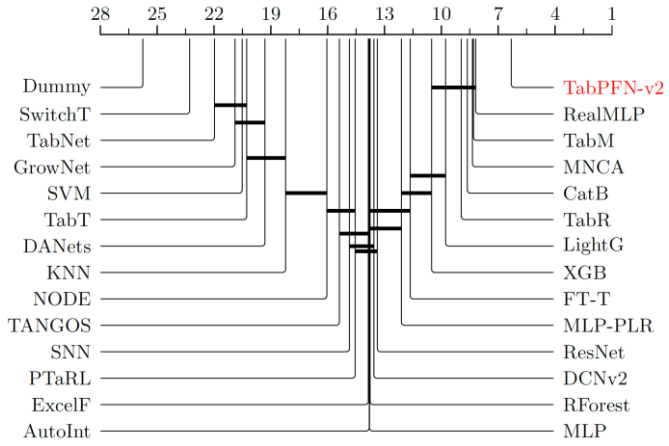
Multi-Class Classification

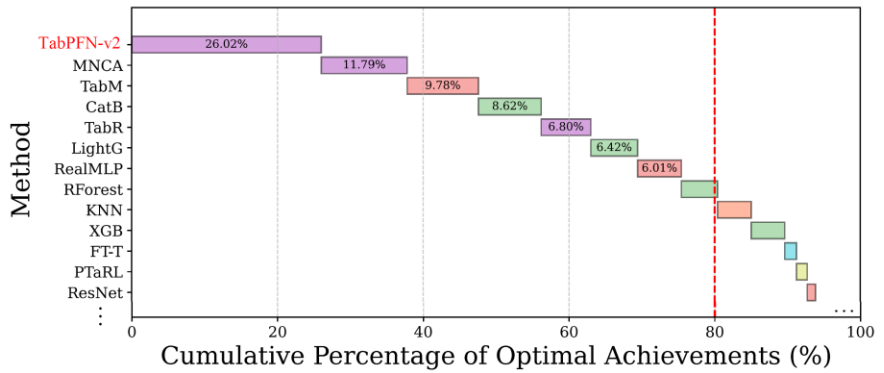


Regression

Capability Analysis of TabPFN v2

- TabPFN v2 performs outstandingly on medium and small-scale datasets
- However, it is difficult to handle high-dimensional, multi-class and large-scale tasks





The Wilcoxon-Holm test (significance level 0.05) was conducted on 273 medium and small-scale datasets, and TabPFN v2 was significantly superior to other current methods.

↓	TabPFN v2	CatB	MNCA	R-MLP	LR
High-Dim	3.36	2.82	4.41	2.14	2.27
Large-Scale	3.97	1.89	2.27	1.94	4.47
>10 classes	3.33	2.75	3.17	1.42	4.33

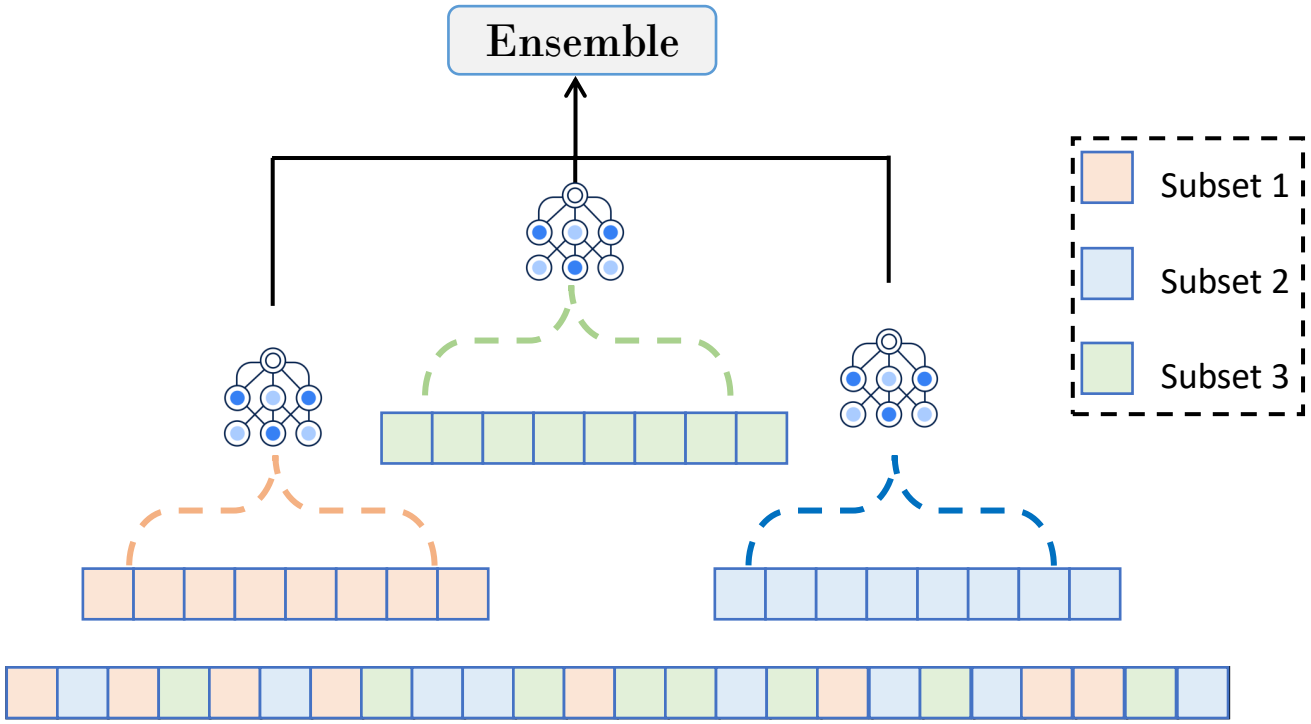
Average rank of TabPFN v2 and representative baseline methods on the following datasets (the lower the value, the better) :

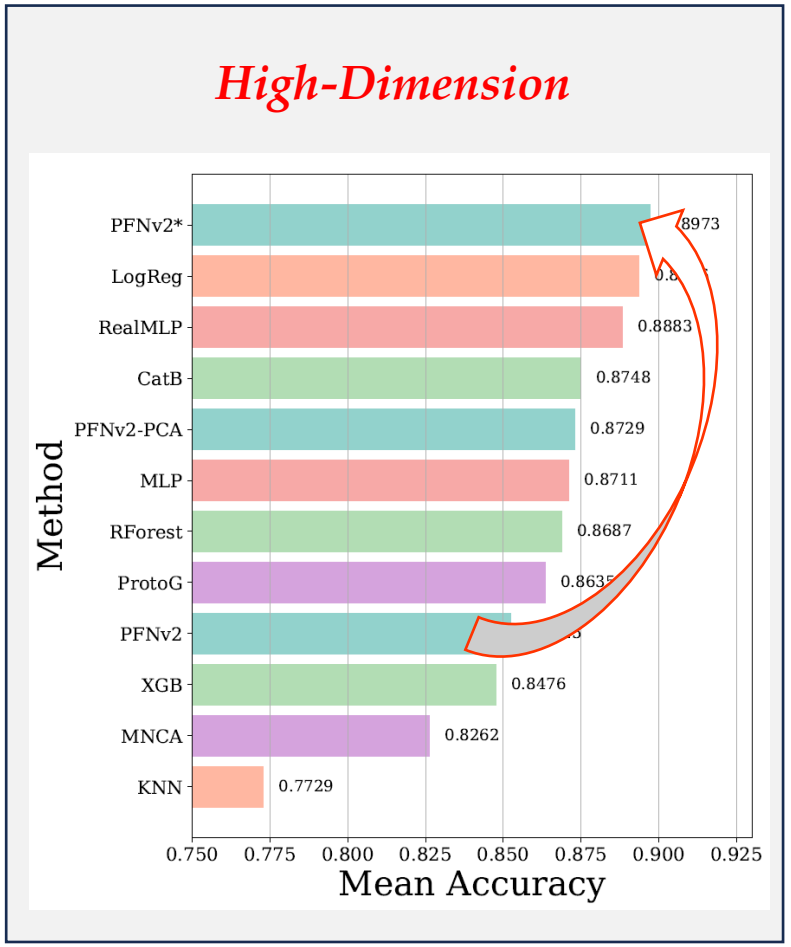
- 18 high-dimensional datasets ($d \geq 2000$)
- 18 large-scale tasks ($Nd \geq 1{,}000{,}000$)
- 12 datasets with classes more than 10

Han-Jia Ye, Si-Yang Liu, Wei-Lun Chao. ***A Closer Look at TabPFN v2: Understanding Its Strengths and Extending Its Capabilities***. NeurIPS 2025.

Test-time Divide-and-Conquer of TFM

- Subsample original dimensions and transform the prediction task to a low-dimension one
- Evaluation on TALENT-extension benchmark

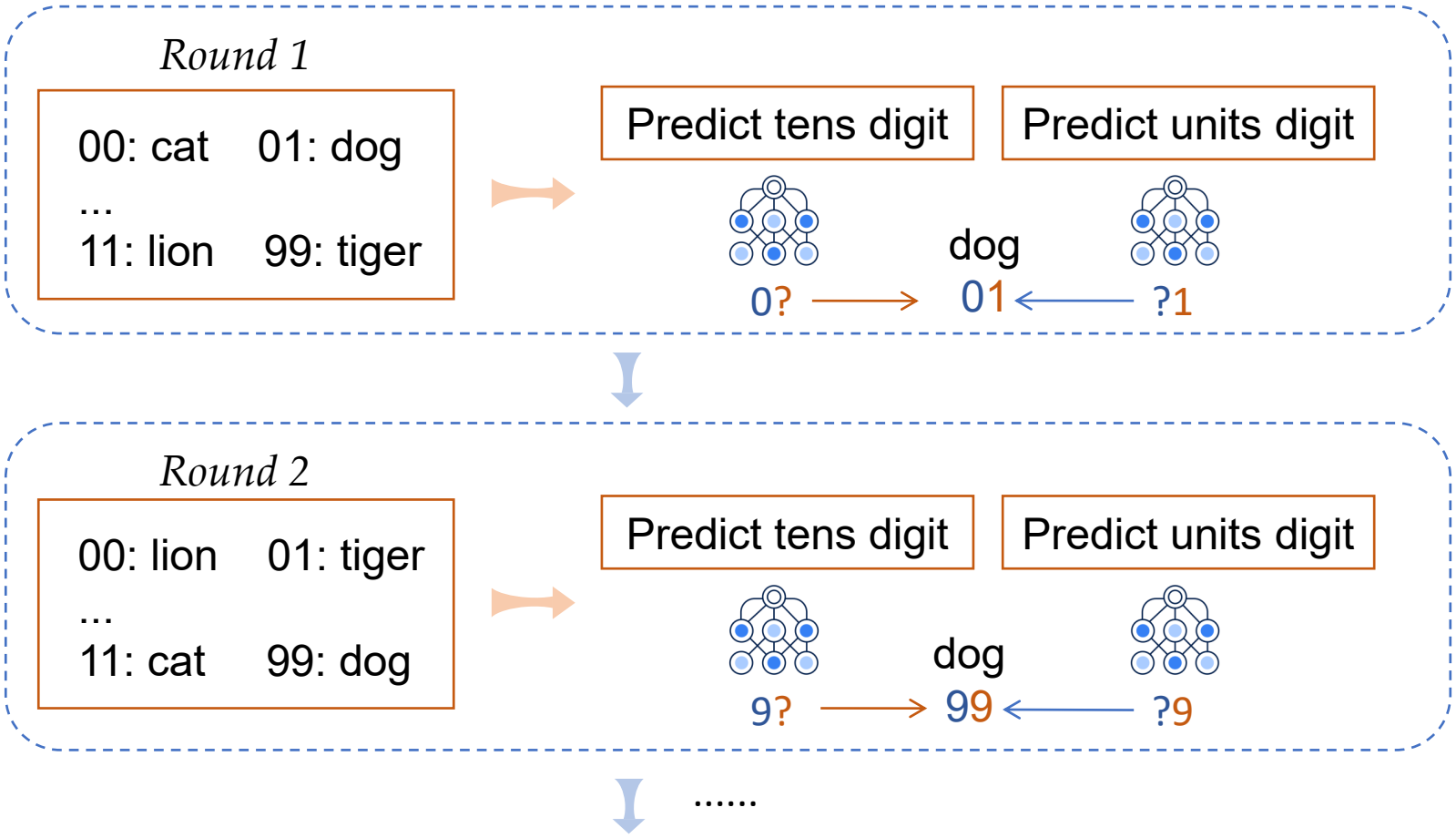


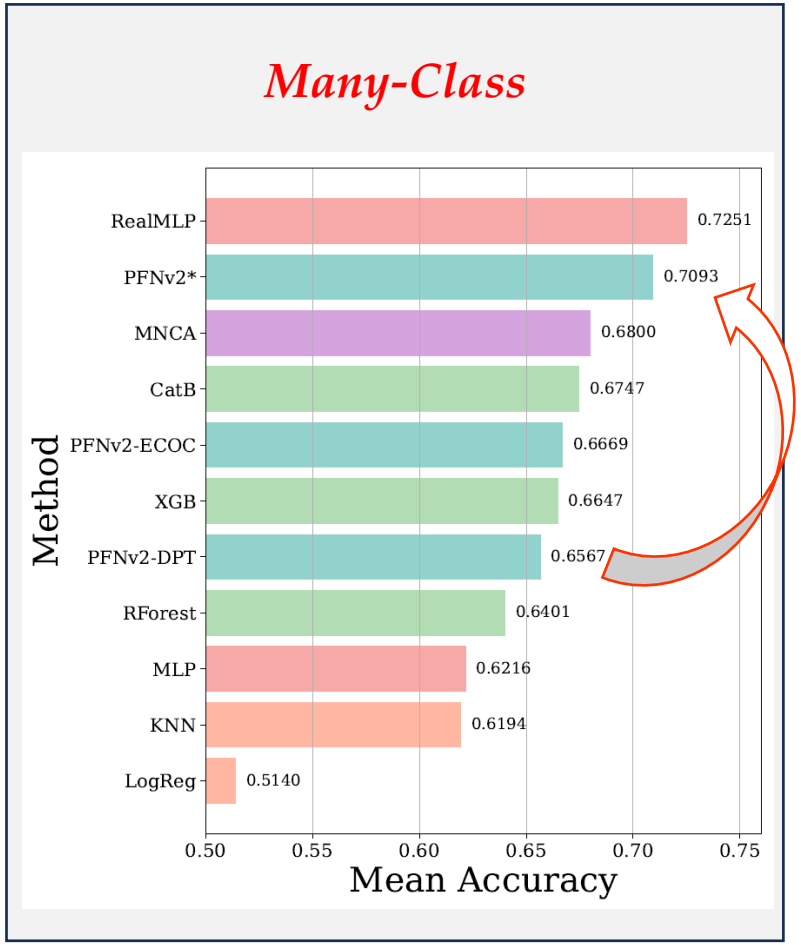


Han-Jia Ye, Si-Yang Liu, Wei-Lun Chao. ***A Closer Look at TabPFN v2: Understanding Its Strengths and Extending Its Capabilities***. NeurIPS 2025.

Test-time Divide-and-Conquer of TFM

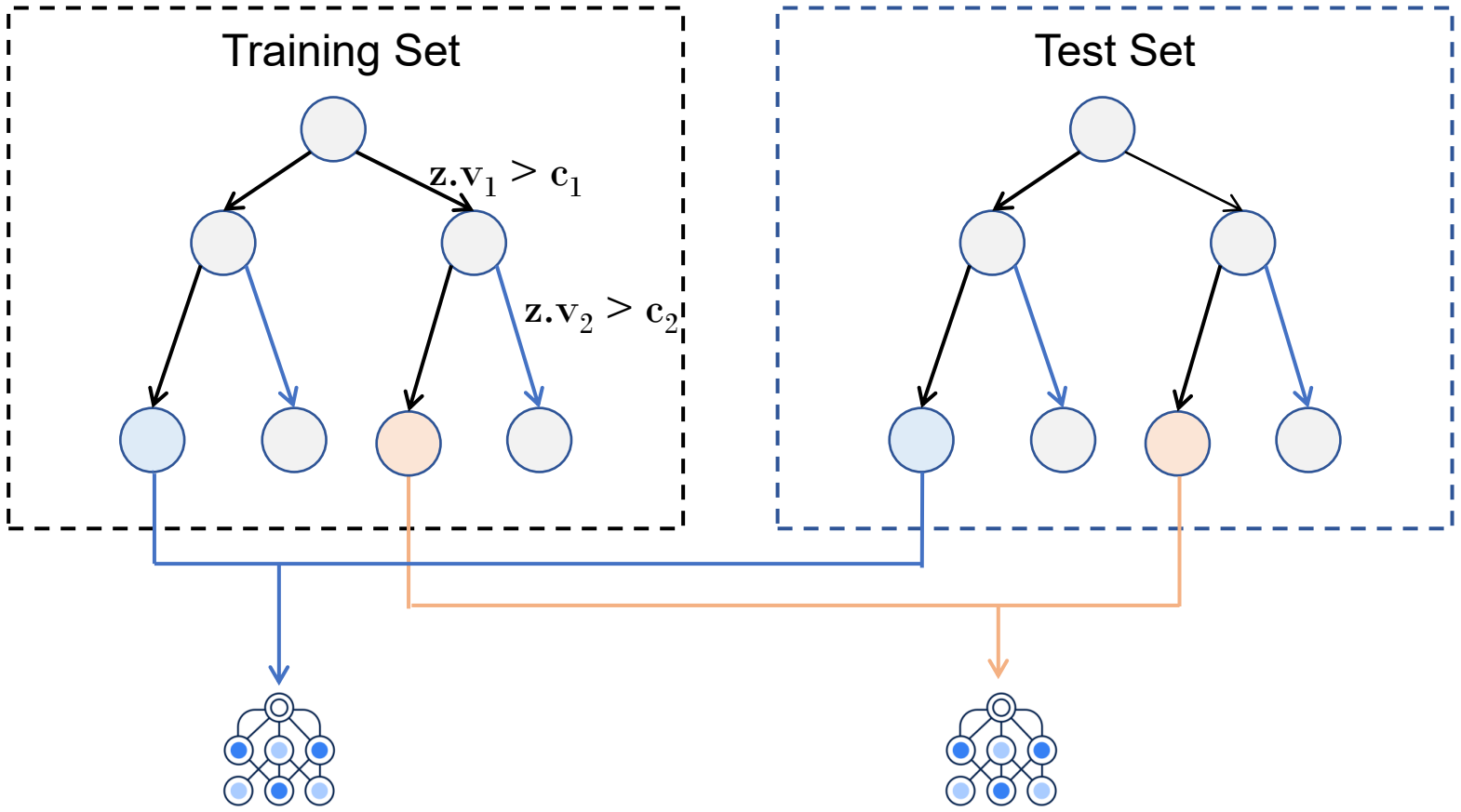
- When there are many classes, we propose a decimal encoding approach that decomposes multi-class problems into multiple 10-class subproblems

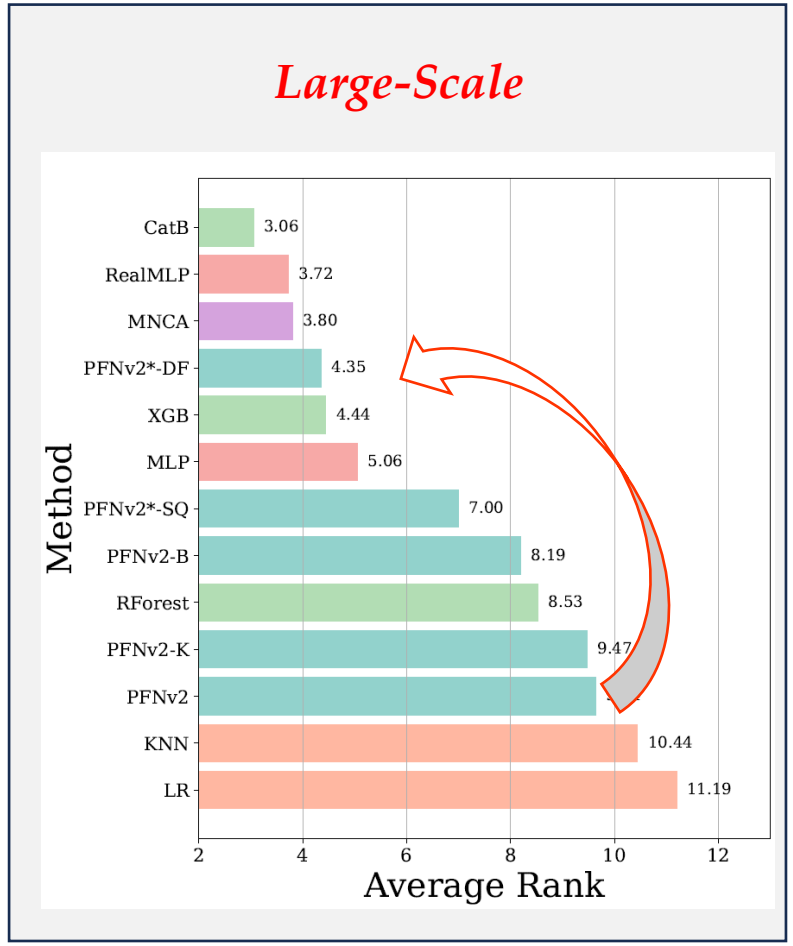




Test-time Divide-and-Conquer of TFM

- Partition the very large dataset into different regions with a shallow decision tree
- Apply TabPFN v2 over each leaf node

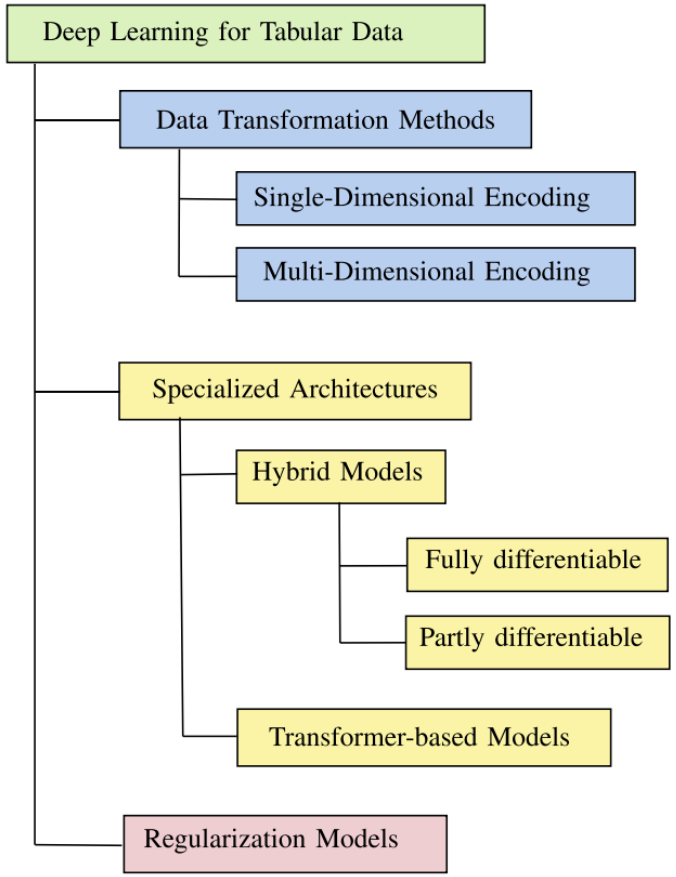




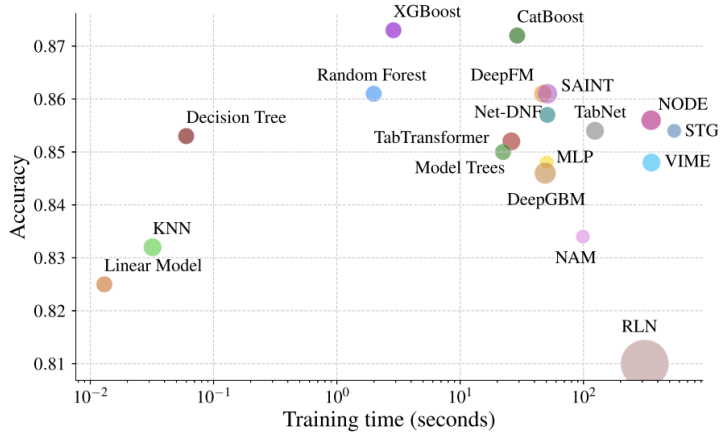
Tabular Data Survey

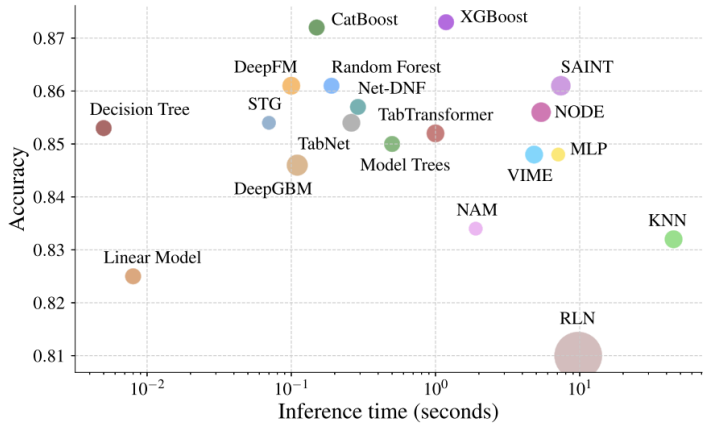
Deep Neural Networks and Tabular Data: A Survey

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci



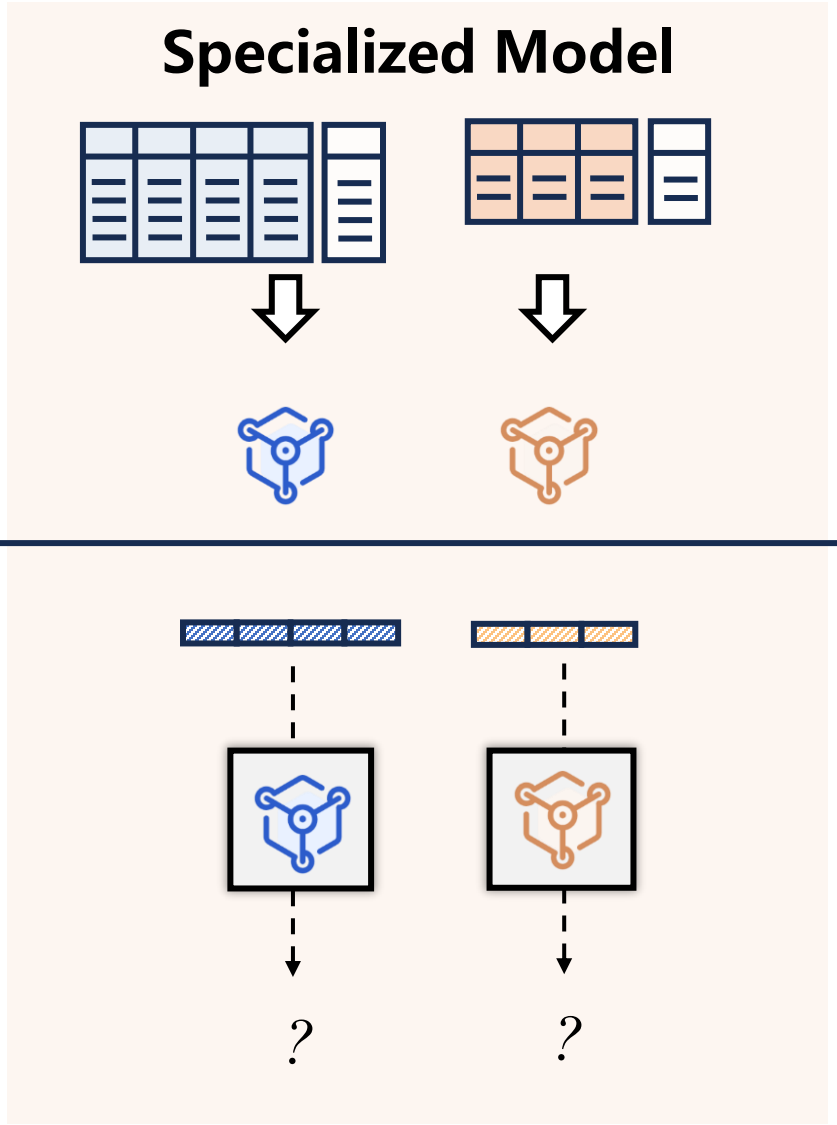
- The existing methods are mainly classified from the perspectives of data, structure and regularization
- Provide an empirical comparison between traditional machine learning methods and 11 deep learning methods
- The internal division within each category is relatively simple, *ignoring* the tabular method with semantic information





Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, Gjergji Kasneci. ***Deep neural networks and tabular data: A survey***. TNNLS 2021

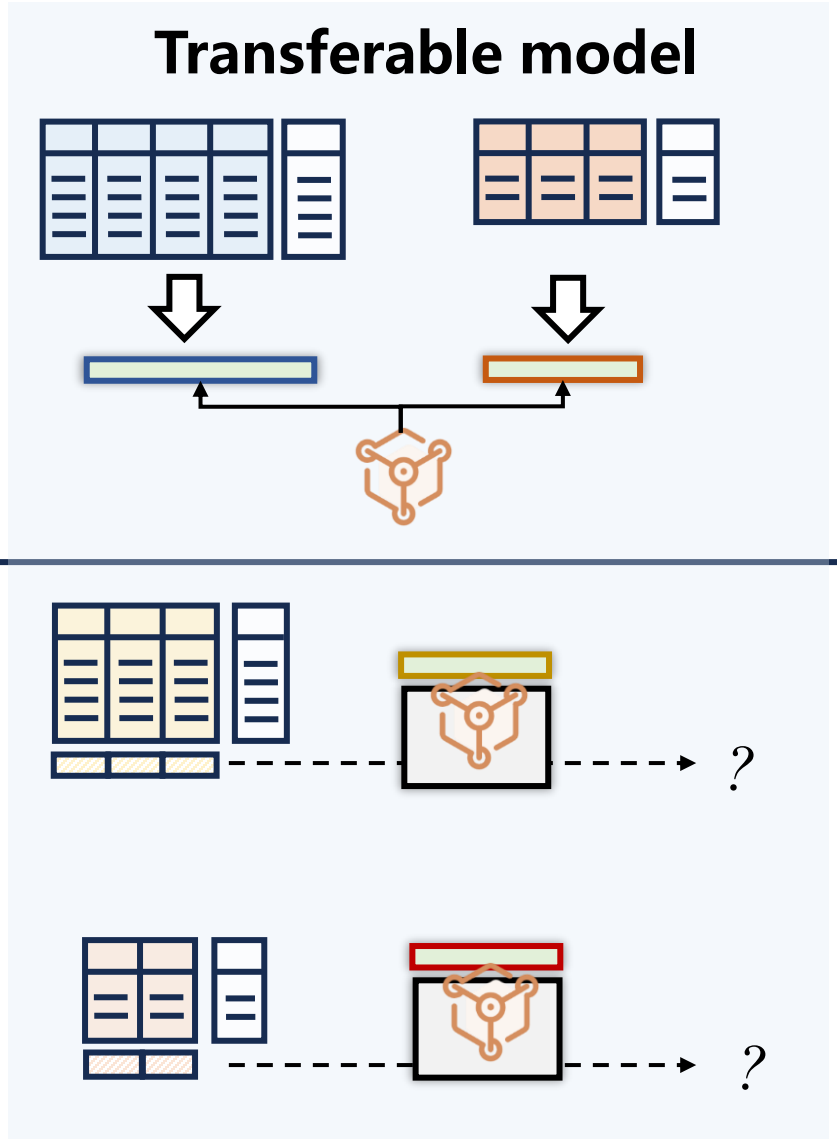
Single-task Tabular Data Model



Train and test the corresponding model on each tabular data, dataset $\mathcal{D}_1$ for model f_1, dataset $\mathcal{D}_2$ for model f_2 ……

The training of models for different tasks is generally independent to each other, and the models show **task/domain specificity**

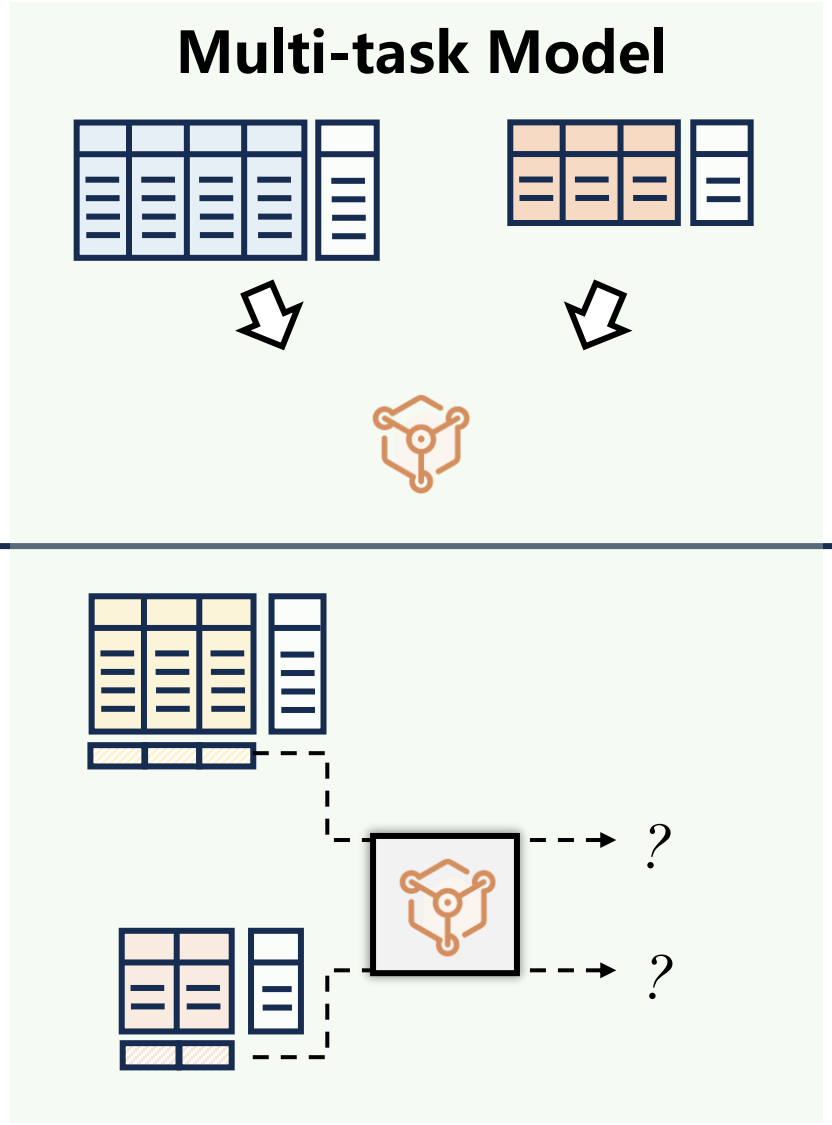
Cross-task Tabular Data Model



Pre-trained on a single or multiple table tasks, light fine-tuning can quickly adapt to downstream tasks

The pre-trained model provides domain knowledge for downstream tasks, and the model shows **task/domain transferability**

General Tabular Model

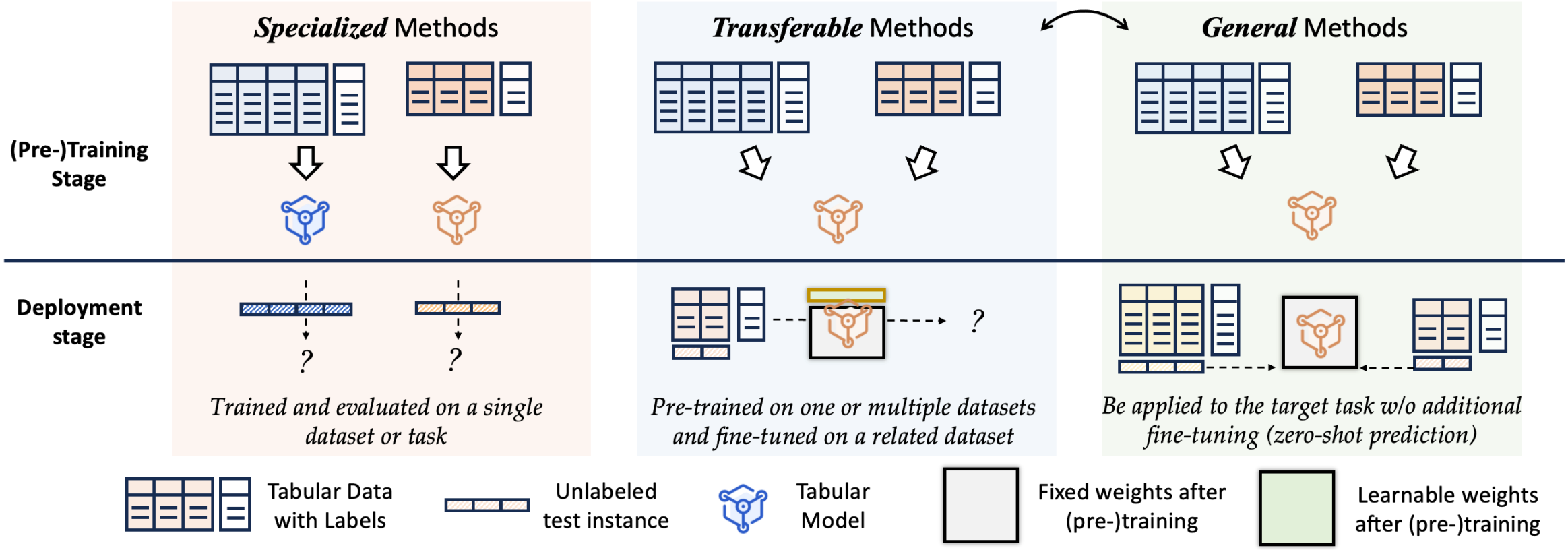


Pre-trained on multiple tabular tasks to obtain a “general” model that can be directly adapted to downstream tasks without fine-tuning

Learning the patterns among heterogeneous data from a large number of tasks, the model shows **task/domain universality**

Tabular Data Survey

- From the perspective of model generalization ability, the current methods are classified, and the table methods with semantic understanding capabilities in the LLM era are comprehensively considered
- The historical background of tabular data, the opportunities and challenges of deep tabular learning, the division of methods and multi-faceted expansions were discussed in detail



Jun-Peng Jiang, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, Han-Jia Ye. ***Representation Learning for Tabular Data: A Comprehensive Survey***. TPAMI 2026

Thank you

Q&A

For more discussions, please contact yehj@nju.edu.cn

Tutorial Slides

Tabular Toolbox

Tabular Benchmark

Tabular Survey