ExcelFormer:

Can a DNN be a Sure Bet for Tabular Prediction?

ACM SIGKDD 2024

資訊所 P76124752 莊上緣

Outline

Outline

Introduction

- Table data is **ubiquitous** in real-world applications, so we need a powerful, efficient, versatile, and user-friendly table prediction method.
- GDBT(Gradient Boosting Decision Tree) and DNN have been extensively utilized by professional users, they present several **challenges** for **casual users**:

  (i) the dilemma of model selection due to their different dataset preferences

  (ii) the need for heavy hyperparameter searching

Introduction

- Deep tabular model has three main drawbacks.
  - P1 : Lack of rotational variance property
  - P2 : Large data demand
  - P3 : Over-smoothing solution

Introduction

- Deep tabular model has three main drawbacks.
    - **P1 : Lack of rotational variance property**
    - P2 : Large data demand
    - P3 : Over-smoothing solution

Rotational variance means that the system or model’s output changes with the rotation of the input.And the rotational invariance means that output remains the same regardless of how the input is rotated.

DNN is a **rotation-invariant** algorithm, so its sample complexity increases linearly with the number of uninformative features in the worst case. Since tabular data often contains many **uninformative features** created by casual users, decision tree algorithms are more effective for tabular data due to the different meanings of each column.

Introduction

- Deep tabular model has three main drawbacks.
  - P1 : Lack of rotational variance property
  - **P2 : Large data demand (primary obstacle to current deep tabular models)**
  - P3 : Over-smoothing solution

Due to the greater number of layers and parameters in DNNs, they have a larger hypothesis space and require more training data for better performance. In comparisons with GBDT, DNNs sometimes perform better on large tabular datasets but often underperform on smaller datasets.

Introduction

- Deep tabular model has three main drawbacks.
  - P1 : Lack of rotational variance property
  - P2 : Large data demand
  - **P3 : Over-smoothing solution**

DNNs often produce overly smooth solutions, which can hurt performance on datasets with irregular boundaries. Decision trees, however, create clear boundaries by splitting the feature space based on axis thresholds.

Introduction

- The author introduces ExcelFormer with three key features:
  1. A **semi-permeable attention(SPA)** mechanism that filters out low-information features, reducing noise and improving model accuracy (addressing P1).
  2. A custom data augmentation method tailored for tabular data (addressing P2).
  3. An Attentive Feedforward Network that dynamically adjusts feature attention at each layer to enhance model fitting and resolve P3
- ExcelFormer is **user-friendly** and **requires minimal hyperparameter tuning**.

Outline

Supervised Tabular Data Prediction

- In tabular data prediction, **GBDT** methods like XGBoost outperform deep neural networks (DNNs) on **small-scale datasets**.

- Despite efforts to enhance DNNs through **sophisticated modules**, **feature embeddings**, and **self-supervised learning**, they still lag behind GBDTs, especially in smaller datasets.

- Models like XTab and TapPFN show some improvement but are often **dataset-specific** and less efficient on larger datasets.

- Overall, DNNs have not yet surpassed GBDTs in performance for tabular data, particularly on **small-scale datasets**, which remains a challenge.

Mixup and other Data Augmentations

- **Mixup** and its variants often struggle with **irregular target patterns**, making them less suitable for tasks like therapy feasibility prediction.
- **ManifoldMix**, similar to Mixup, and **CutMix** variants work well for **images** but not for tabular data.
- Kadra et al. explored **data augmentations for MLPs on tabular data**, finding them effective **only for limited datasets and requiring extensive testing.**
- This paper introduces **Hid-Mix** and **Feat-Mix**, which avoid Mixup conflicts and enhance ExcelFormer performance.

Outline

Solving (P1) with Semi-Permeable Attention(SPA)

- **Less informative features** require more training samples for DNNs to learn to ignore them, leading to **data inefficiency**.
- We propose incorporating an inductive bias into the self-attention mechanism to selectively restrict the impact of these features, **reducing their overall influence on prediction outcomes**. This is called the **semi-permeable attention module (SPA)**.

$$z' = \text{softmax}\left(\frac{(zW_q)(zW_k)^T \oplus M}{\sqrt{d}}\right)(zW_v)$$

z	input embeddings	$\mathbb{R}^{f\times d}$
z'	output embeddings	$\mathbb{R}^{f\times d}$
W_q, W_k, W_v	learning matrices	$\mathbb{R}^{d\times d}$
M	unoptimizable mask	$\mathbb{R}^{f\times f}$

Solving (P1) with Semi-Permeable Attention(SPA)

- **Less informative features** require more training samples for DNNs to learn to ignore them, leading to **data inefficiency**.
- We propose incorporating an inductive bias into the self-attention mechanism to selectively restrict the impact of these features, **reducing their overall influence on prediction outcomes**. This is called the **semi-permeable attention module (SPA)**.

$$z' = \text{softmax}\left(\frac{(zW_q)(zW_k)^T \oplus M}{\sqrt{d}}\right)(zW_v)$$

z	input embeddings	$\mathbb{R}^{f\times d}$
z'	output embeddings	$\mathbb{R}^{f\times d}$
W_q, W_k, W_v	learning matrices	$\mathbb{R}^{d\times d}$
M	unoptimizable mask	$\mathbb{R}^{f\times f}$

the element at the i-th row and j-th column is defined by:

$$M[i,j] = \begin{cases} -\infty & I(\mathbf{f}_i) > I(\mathbf{f}_j) \\ 0 & I(\mathbf{f}_i) \le I(\mathbf{f}_j) \end{cases}$$

Solving (P1) with Semi-Permeable Attention(SPA)

the element at the *i*-th row and *j*-th column is defined by:

$$M[i,j] = \begin{cases} -\infty & I(\mathbf{f}_i) > I(\mathbf{f}_j) \\ 0 & I(\mathbf{f}_i) \le I(\mathbf{f}_j) \end{cases}$$

The function $I(\cdot)$ represents a measure of feature importance, and we use the “mutual information” metric in this paper.

In this study, we employ **Normalized Mutual Information (NMI)** to assess the importance of various features, as mutual information can capture dependencies between features and targets. We implement NMI using the **sklearn Python package**.

Classification task : "feature_selection.mutual_info_classif" function

Regression task : "feature_selection.mutual_info_regression" function

Solving (P1) with Semi-Permeable Attention(SPA)

the element at the i-th row and j-th column is defined by:

$$M[i,j] = \begin{cases} -\infty & I(\mathbf{f}_i) > I(\mathbf{f}_j) \\ 0 & I(\mathbf{f}_i) \leq I(\mathbf{f}_j) \end{cases}$$

If a feature fi is more informative compared to fj , M [i, j] is set −∞ (we use −10^5 in implementation) and thus the (i, j) grid on the attention map is masked. It prevents the transfer of the embedding of the feature fj to the fi 's embedding.

When we set a value at a certain position to negative infinity, the softmax output corresponding to that position will be close to zero. This is because in the softmax calculation:

$$\mathrm{Softmax}(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \qquad \boxed{\exp^{-\infty} = 0} \qquad z' = \boxed{\mathrm{softmax}}\left(\frac{(zW_q)(zW_k)^T \oplus M}{\sqrt{d}}\right)(zW_v)$$

Thus, by setting a value to negative infinity, we effectively "mask" that position, ensuring it has no influence on the softmax output.

Solving (P1) with Semi-Permeable Attention(SPA)

The SPA mechanism allows only more informative features to propagate information, constraining the impact of less informative ones.

This preserves interaction pathways between features while limiting the influence of noisy features, making the model more data-efficient.

SPA reduces the need for ExcelFormer to learn partial rotational directions, balancing between rotational invariance and variance.

In practice, SPA is extended to a multi-head self-attention version with 32 heads by default.

Interaction Attenuated Initialization

Our interaction attenuated initialization scheme is built upon the commonly used He’s initialization or Xavier initialization, by **rescaling the variance of an initialized weight w with γ ($\gamma \to 0+$)** while **keeping the expectation at 0**

$$\mathrm{Var}(w) = \gamma \mathrm{Var}_{\mathrm{prev}}(w)$$

Interaction Attenuated Initialization

Our interaction attenuated initialization scheme is built upon the commonly used He's initialization or Xavier initialization, by **rescaling the variance of an initialized weight w with γ ($\gamma \to 0+$)** while **keeping the expectation at 0**

$$\boxed{\mathrm{Var}(w) = \gamma \mathrm{Var}_{\mathrm{prev}}(w)}$$

To reduce the impacts of SPA, we apply Eq. (3) to all the parameters in the SPA module. Thus, **ExcelFormer works like a non-rotationally invariant model initially**

During training, weights are adjusted based on data feedback, gradually increasing SPA's impact. This allows the model to strengthen important feature interactions as it learns from the data.

Solving (P3) with GLU layer

Decision trees excel in handling irregular tabular feature-target relationships by using multiple thresholds for splitting. In contrast, existing Transformers use a two-layer MLP as their FFN, which has limited non-linear fitting capability.

To improve this, we replace the standard FFN with a Gated Linear Unit (GLU) layer, using "tanh" activation instead of "sigmoid" for better optimization.

$$z' = \tanh\left(\mathtt{Linear}_1(z)\right) \odot \mathtt{Linear}_2(z)$$

Please note that both the vanilla FFN and GLU employ two fully connection layers
(FFN is defined by **z' = Linear1 (ReLU(Linear2 (z))**)), resulting in **similar computational costs!!** 🤩

Solving (P3) with GLU layer

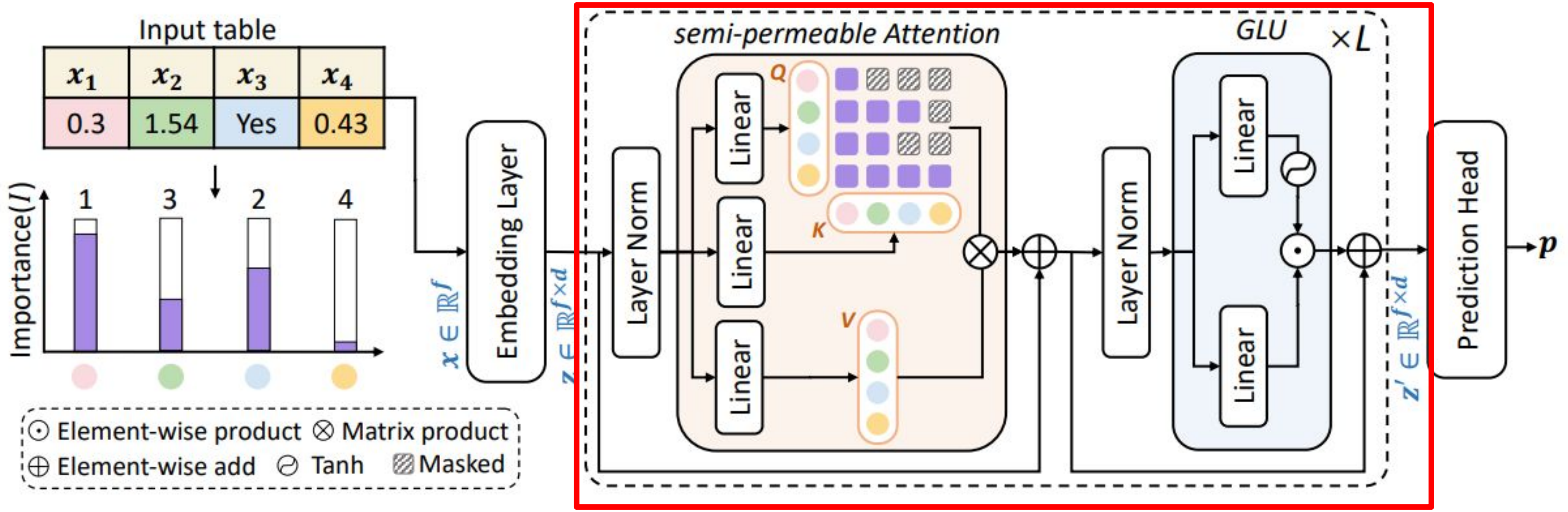
Advantages of GLU:

- **Enhanced Non-Linearity:** GLU's gating mechanism allows for capturing complex non-linear relationships more flexibly.
- **Controlled Information Flow:** GLU selectively passes information, improving robustness by handling noise and redundant features better.
- **Efficient Computation:** GLU maintains similar computational costs to standard FFNs but provides stronger representational power.

In ExcelFormer, GLU replaces the traditional two-layer FFN to improve fitting irregular feature-target relationships, and using tanh instead of sigmoid in GLU further optimizes training.

Solving (P3) with GLU layer

The SPA and GLU modules are alternately stacked to form the core structure of the ExcelFormer model



The rest part of ExcelFormer

Feature Pre-processing: Numerical features are normalized using quantile transformation, while categorical features are encoded numerically with CatBoost Encoder from Sklearn, similar to previous methods (e.g., FT-Transformer).

Prediction Head: The prediction head consists of two fully connected layers applied to the top transformer block output.

$$p = \phi(\texttt{Linear}_d(\text{P-ReLU}(\texttt{Linear}_f(z^{(L)}))))$$

For multiclassification task, ϕ indicates softmax.

For regression and binary classification tasks, ϕ is sigmoid.

Outline

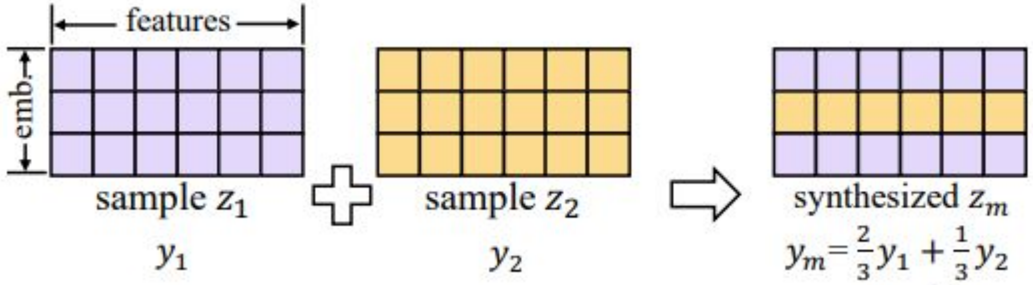
SOLVING (P2) WITH DATA AUGMENTATION

A straightforward approach to tackle data insufficiency is to create training data.Mixup is effective for regularizing DNNs in computer vision by promoting linearity between samples, but it doesn't perform well on tabular data due to the mismatch between linear assumptions and irregular target functions. To address this, we introduce two Mixup variants, **Hid-Mix** and **Feat-Mix**, to better handle these conflicts in tabular data.

HID-MIX

Hid-Mix is applied to token-level embeddings after processing by the embedding layer. It randomly exchanges embedding dimensions between two samples.

$$\begin{cases} z_{\mathrm{m}}^{(0)} = S_H \odot z_1^{(0)} + (\mathbb{1}_H - S_H) \odot z_2^{(0)} \\ y_{\mathrm{m}} = \lambda_H y_1 + (1 - \lambda_H) y_2, \end{cases}$$



(a) An example of HID-MIX (on feature embedding), $\lambda_H = \frac{2}{3}$

HID-MIX

$$z(0)_1 = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \qquad y1 = 0$$

$$z(0)_2 = \begin{pmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{pmatrix} \qquad y2 = 1$$

HID-MIX

$$z(0)_1 = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

$$y1 = 0$$

$$z(0)_2 = \begin{pmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{pmatrix}$$

$$y2 = 1$$

Assume that λH sampled from the Beta distribution is 0.5.

Construct the sh vector and SH matrix (assuming the first and third elements are randomly chosen to be 1):

$$sh = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}$$

$$SH = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

HID-MIX

$$z(0)_1 = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \qquad y1 = 0$$

$$z(0)_2 = \begin{pmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{pmatrix} \qquad y2 = 1$$

Assume that λH sampled from the Beta distribution is 0.5.

Construct the sh vector and SH matrix (assuming the first and third elements are randomly chosen to be 1):

$$sh = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}$$

$$SH = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

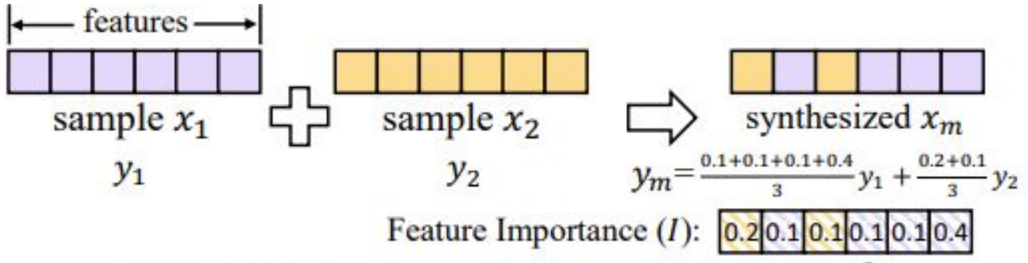
$$\begin{aligned} z(0)_m &= SH \odot z(0)_1 + (1H - SH) \odot z(0)_2 \\ &= \begin{pmatrix} 1 & 0 & 3 \\ 4 & 0 & 6 \\ 7 & 0 & 9 \end{pmatrix} + \begin{pmatrix} 0 & 8 & 0 \\ 0 & 5 & 0 \\ 0 & 2 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 8 & 3 \\ 4 & 5 & 6 \\ 7 & 2 & 9 \end{pmatrix} \end{aligned}$$

$$ym = 0.5 \times 0 + 0.5 \times 1 = 0.5$$

FEAT-MIX

Unlike Hid-Mix, Feat-Mix creates new samples by swapping features between two randomly selected samples, while generating labels based on feature importance(Using the same method as in SPA for computing feature importance).

$$\begin{cases} x_m = \mathbf{s}_F \odot x_1 + (\mathbb{1}_F - \mathbf{s}_F) \odot x_2, \\ y_m = \Lambda y_1 + (1 - \Lambda) y_2, \end{cases} \qquad \Lambda = \frac{\sum_{\mathbf{s}_F^{(i)}=1} I(\mathbf{f}_i)}{\sum_{i=1}^{f} I(\mathbf{f}_i)},$$


features
sample x_1
y_1
sample x_2
y_2
synthesized x_m
$y_m = \frac{0.1+0.1+0.1+0.4}{3} y_1 + \frac{0.2+0.1}{3} y_2$
Feature Importance (I):
0.2
0.1
0.1
0.1
0.1
0.4

(b) An example of FEAT-MIX (on input data), $\lambda_F = \frac{2}{3}$

FEAT-MIX

$x1 = [0.1, 0.5, 0.3]$ 、 $y1 = 1$

$x2 = [0.4, 0.2, 0.7]$ 、 $y2 = 0$

$$s_F = [1, 0, 1]$$

FEAT-MIX

$$x1 = [0.1, 0.5, 0.3] \text{，} y1 = 1$$

$$x2 = [0.4, 0.2, 0.7] \text{，} y2 = 0$$

$$s_F = [1, 0, 1] \rightarrow x_m = [0.1, 0.2, 0.3]$$

FEAT-MIX

$$x1 = [0.1, 0.5, 0.3] ，y1 = 1$$
$$x2 = [0.4, 0.2, 0.7] ，y2 = 0$$

$$s_F = [1, 0, 1]$$

$$x_m = [0.1, 0.2, 0.3]$$

Assume that feature importances is $[0.3, 0.4, 0.3] \longrightarrow$

$$\Lambda = \frac{0.3\times1+0.3\times1}{0.3+0.4+0.3} = \frac{0.6}{1} = 0.6$$

FEAT-MIX

$x1 = [0.1, 0.5, 0.3]$, $y1 = 1$

$x2 = [0.4, 0.2, 0.7]$, $y2 = 0$

$s_F = [1, 0, 1]$

$x_m = [0.1, 0.2, 0.3]$

Assume that feature importances is $[0.3, 0.4, 0.3]$

$\Lambda = \frac{0.3\times1+0.3\times1}{0.3+0.4+0.3} = \frac{0.6}{1} = 0.6$

$y_m = 0.6 \times 1 + (1 - 0.6) \times 0 = 0.6$

FEAT-MIX

Feat-Mix creates new samples by swapping features between two random samples and generating labels based on feature importance. Without feature importance, it resembles CutMix, but Feat-Mix is more robust due to the presence of uninformative features. It promotes fewer feature interactions, aligning with the Interaction Attenuated Initialization and enhancing ExcelFormer's non-rotational invariance.

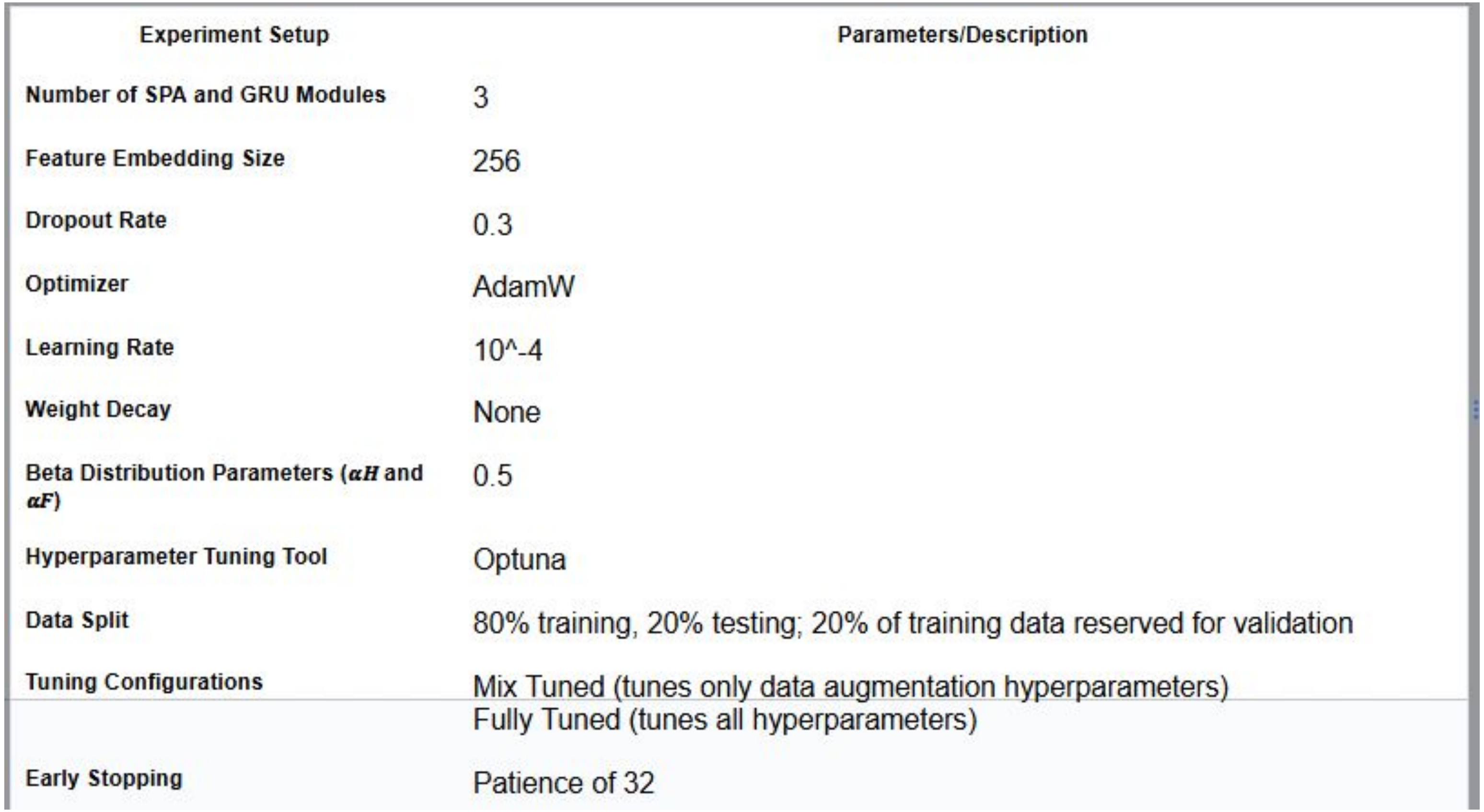
Outline

Training and Loss Function

ExcelFormer supports both classification and regression tasks. During training, data augmentation can be applied with either Hid-Mix or Feat-Mix, or both successively. However, testing suggests that using only one method might be more effective for some datasets. **Cross-entropy loss is used for classification, while mean square error loss is used for regression.**

Outline

Experiments(Experimental Setup**)**

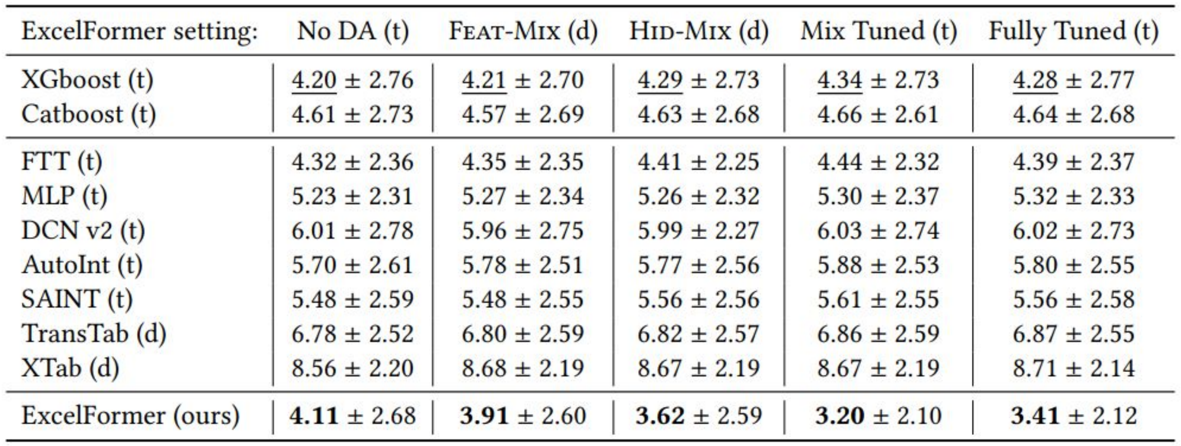


Experiment Setup	Parameters/Description
Number of SPA and GRU Modules	3
Feature Embedding Size	256
Dropout Rate	0.3
Optimizer	AdamW
Learning Rate	10^-4
Weight Decay	None
Beta Distribution Parameters (αH and αF)	0.5
Hyperparameter Tuning Tool	Optuna
Data Split	80% training, 20% testing; 20% of training data reserved for validation
Tuning Configurations	Mix Tuned (tunes only data augmentation hyperparameters)
	Fully Tuned (tunes all hyperparameters)
Early Stopping	Patience of 32

Experiments(Experimental Setup**)**

Datasets	96 small datasets (<10,000 samples) 21 large datasets (>10,000 samples)
Compared Models	XGBoost, CatBoost, FT-Transformer (FTT), SAINT, MLP, DCN v2, AutoInt, TapPFN,TransTab, XTab (pre-trained)
Compared Models Settings	XGBoost and CatBoost tuned to 4096 trees and 500iterations FT-Transformer, SAINT, TapFPN tuned as per original papers
Evaluation Metrics	Classification: AUC, Accuracy (ACC) Regression: Negative RMSE (nRMSE)

Performances on Small Datasets



ExcelFormer setting:	No DA (t)	Feat-Mix (d)	Hid-Mix (d)	Mix Tuned (t)	Fully Tuned (t)
XGboost (t)	<u>4.20</u> ± 2.76	<u>4.21</u> ± 2.70	<u>4.29</u> ± 2.73	<u>4.34</u> ± 2.73	<u>4.28</u> ± 2.77
Catboost (t)	4.61 ± 2.73	4.57 ± 2.69	4.63 ± 2.68	4.66 ± 2.61	4.64 ± 2.68
FTT (t)	4.32 ± 2.36	4.35 ± 2.35	4.41 ± 2.25	4.44 ± 2.32	4.39 ± 2.37
MLP (t)	5.23 ± 2.31	5.27 ± 2.34	5.26 ± 2.32	5.30 ± 2.37	5.32 ± 2.33
DCN v2 (t)	6.01 ± 2.78	5.96 ± 2.75	5.99 ± 2.27	6.03 ± 2.74	6.02 ± 2.73
AutoInt (t)	5.70 ± 2.61	5.78 ± 2.51	5.77 ± 2.56	5.88 ± 2.53	5.80 ± 2.55
SAINT (t)	5.48 ± 2.59	5.48 ± 2.55	5.56 ± 2.56	5.61 ± 2.55	5.56 ± 2.58
TransTab (d)	6.78 ± 2.52	6.80 ± 2.59	6.82 ± 2.57	6.86 ± 2.59	6.87 ± 2.55
XTab (d)	8.56 ± 2.20	8.68 ± 2.19	8.67 ± 2.19	8.67 ± 2.19	8.71 ± 2.14
ExcelFormer (ours)	**4.11** ± 2.68	**3.91** ± 2.60	**3.62** ± 2.59	**3.20** ± 2.10	**3.41** ± 2.12

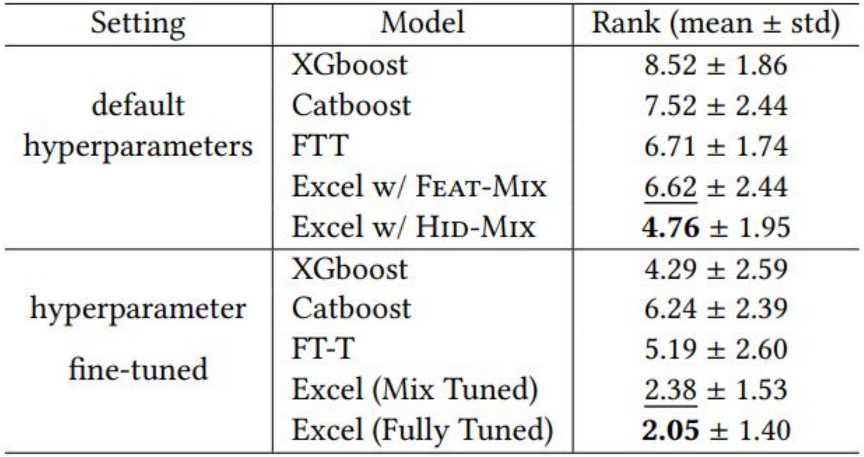
1.ExcelFormer’s Superiority:ExcelFormer consistently outperforms other models on small datasets,regardless of hyperparameter tuning.

2.Hid-Mix vs. Feat-Mix:ExcelFormer with Hid-Mix slightly outperforms Feat-Mix;performance improves further with hyperparameter tuning.

3.Hyperparameter Tuning:Tuning reduces performance rank variability,leading to more stable results.Interestingly,”Mix-Tuned” configuration performs better than “Fullt Tuned” under the same iterations, likely due to more efficient data augmentation tuning.

4.Architecture Effectiveness:Even without Feat-Mix and Hid-mix,ExcelFormer outperforms existing models,highlighting its superior architecture.

Performances on Large Datasets



Setting	Model	Rank (mean ± std)
default hyperparameters	XGboost	8.52 ± 1.86
	Catboost	7.52 ± 2.44
	FTT	6.71 ± 1.74
	Excel w/ Feat-Mix	6.62 ± 2.44
	Excel w/ Hid-Mix	**4.76** ± 1.95
hyperparameter fine-tuned	XGboost	4.29 ± 2.59
	Catboost	6.24 ± 2.39
	FT-T	5.19 ± 2.60
	Excel (Mix Tuned)	2.38 ± 1.53
	Excel (Fully Tuned)	**2.05** ± 1.40

1.Model Comparison:ExcelFormer is compared with three SOTA models(XGBoost,CatBoost and FTT),excluding others due to inferior performance and high computational load.

2.Evaluation Setting:Each model is evaluated with default hyperparameters and dataset-adaptive fine-tuning.

3.Performance Results:Excelformer outperforms previous models in both setting, as shown in table.

4.Hid-Mix Performance:ExcelFormer with Hid-Mix performs comparably to prior models with hyperparameter tuning,consistent with small dataset findings.

5.Hyperparameter Tuning Conclusion:Unlike small datasets,fully tuned ExcelFormer outperforms the Mix Tuned version on large datasets.

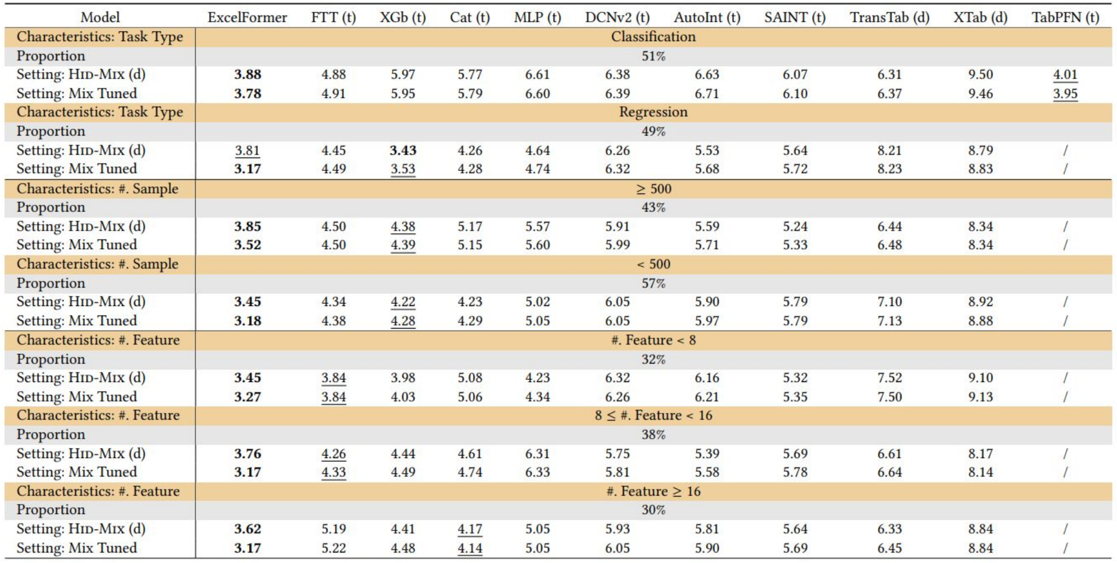
Tasks

1.Model Performance:ExcelFormer performs well on both GBDT-favored small datasets and DNN-favored large datasets,addressing DNN’s existing drawbacks in tabular prediction.

2.Hyperparameter Setting:Even with default hyperparameter tuned competitors on small datasets and perform comparably on large ones.This makes it a strong solution for non-experts and a top choice for professionals,as hyperparameter-tuned ExcelFormer excels in various tabular prediction tasks.

Can ExcelFormer be a sure bet solution across various types of datasets?

Table 3: Performance evaluation within several dataset subgroups. Performance rank (↓) within the datasets are reported. The best scores are in bold and the runners-up are underlined. "(d)": default hyperparameters; "(t)": finely tuned hyperparameters.



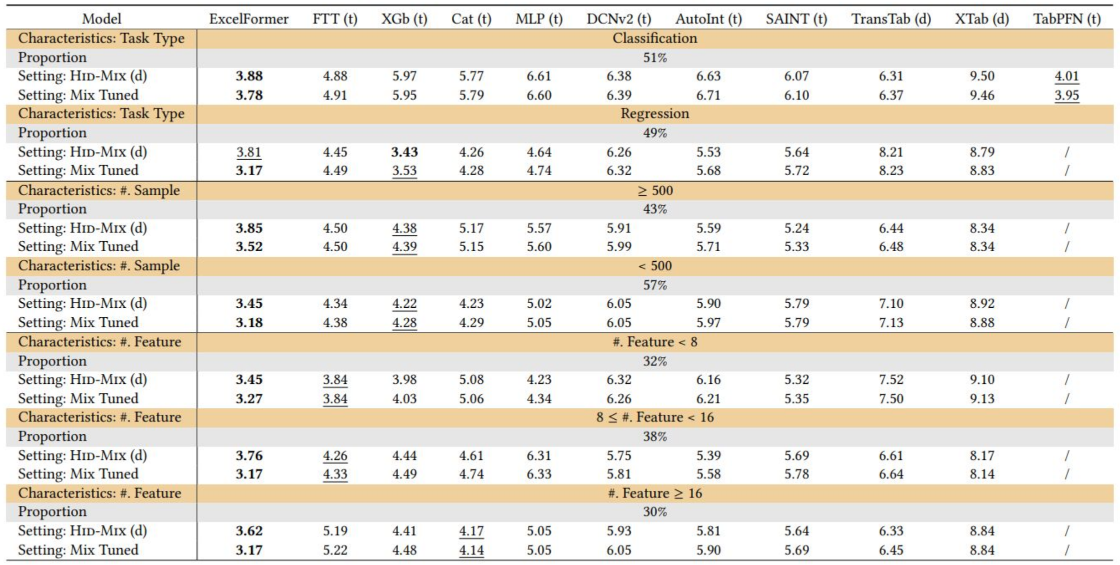
Model	ExcelFormer	FTT (t)	XGb (t)	Cat (t)	MLP (t)	DCNv2 (t)	AutoInt (t)	SAINT (t)	TransTab (d)	XTab (d)	TabPFN (t)
Characteristics: Task Type	Classification										
Proportion	51%										
Setting: Hid-Mix (d)	**3.88**	4.88	5.97	5.77	6.61	6.38	6.63	6.07	6.31	9.50	<u>4.01</u>
Setting: Mix Tuned	**3.78**	4.91	5.95	5.79	6.60	6.39	6.71	6.10	6.37	9.46	<u>3.95</u>
Characteristics: Task Type	Regression										
Proportion	49%										
Setting: Hid-Mix (d)	<u>3.81</u>	4.45	**3.43**	4.26	4.64	6.26	5.53	5.64	8.21	8.79	/
Setting: Mix Tuned	**3.17**	4.49	<u>3.53</u>	4.28	4.74	6.32	5.68	5.72	8.23	8.83	/
Characteristics: #. Sample	≥ 500										
Proportion	43%										
Setting: Hid-Mix (d)	**3.85**	4.50	<u>4.38</u>	5.17	5.57	5.91	5.59	5.24	6.44	8.34	/
Setting: Mix Tuned	**3.52**	4.50	<u>4.39</u>	5.15	5.60	5.99	5.71	5.33	6.48	8.34	/
Characteristics: #. Sample	< 500										
Proportion	57%										
Setting: Hid-Mix (d)	**3.45**	4.34	<u>4.22</u>	4.23	5.02	6.05	5.90	5.79	7.10	8.92	/
Setting: Mix Tuned	**3.18**	4.38	<u>4.28</u>	4.29	5.05	6.05	5.97	5.79	7.13	8.88	/
Characteristics: #. Feature	#. Feature < 8										
Proportion	32%										
Setting: Hid-Mix (d)	**3.45**	<u>3.84</u>	3.98	5.08	4.23	6.32	6.16	5.32	7.52	9.10	/
Setting: Mix Tuned	**3.27**	<u>3.84</u>	4.03	5.06	4.34	6.26	6.21	5.35	7.50	9.13	/
Characteristics: #. Feature	8 ≤ #. Feature < 16										
Proportion	38%										
Setting: Hid-Mix (d)	**3.76**	<u>4.26</u>	4.44	4.61	6.31	5.75	5.39	5.69	6.61	8.17	/
Setting: Mix Tuned	**3.17**	<u>4.33</u>	4.49	4.74	6.33	5.81	5.58	5.78	6.64	8.14	/
Characteristics: #. Feature	#. Feature ≥ 16										
Proportion	30%										
Setting: Hid-Mix (d)	**3.62**	5.19	4.41	<u>4.17</u>	5.05	5.93	5.81	5.64	6.33	8.84	/
Setting: Mix Tuned	**3.17**	5.22	4.48	<u>4.14</u>	5.05	6.05	5.90	5.69	6.45	8.84	/

Comprehensive Evaluation:

To ensure ExcelFormer is a reliable solution,datasets are divided into subgroups based on task type,dataset size,and number of features,and its performance is examined in each subgroup.

Can ExcelFormer be a sure bet solution across various types of datasets?

Table 3: Performance evaluation within several dataset subgroups. Performance rank (↓) within the datasets are reported. The best scores are in bold and the runners-up are underlined. "(d)": default hyperparameters; "(t)": finely tuned hyperparameters.



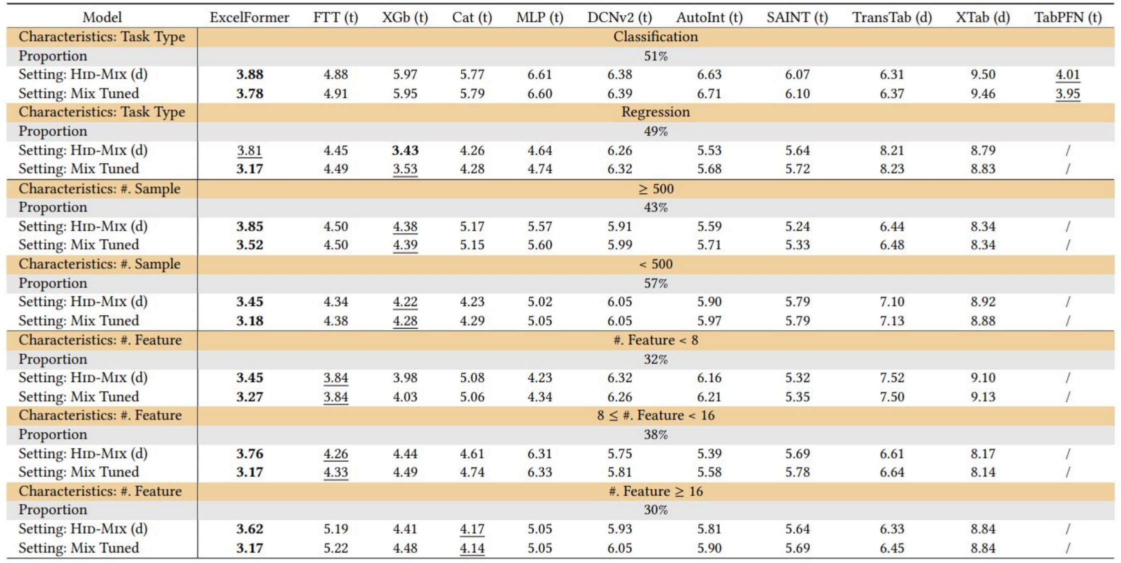
Model	ExcelFormer	FTT (t)	XGb (t)	Cat (t)	MLP (t)	DCNv2 (t)	AutoInt (t)	SAINT (t)	TransTab (d)	XTab (d)	TabPFN (t)
Characteristics: Task Type	Classification										
Proportion	51%										
Setting: HID-MIX (d)	**3.88**	4.88	5.97	5.77	6.61	6.38	6.63	6.07	6.31	9.50	4.01
Setting: Mix Tuned	**3.78**	4.91	5.95	5.79	6.60	6.39	6.71	6.10	6.37	9.46	3.95
Characteristics: Task Type	Regression										
Proportion	49%										
Setting: HID-MIX (d)	3.81	4.45	**3.43**	4.26	4.64	6.26	5.53	5.64	8.21	8.79	/
Setting: Mix Tuned	**3.17**	4.49	3.53	4.28	4.74	6.32	5.68	5.72	8.23	8.83	/
Characteristics: #. Sample	≥ 500										
Proportion	43%										
Setting: HID-MIX (d)	**3.85**	4.50	4.38	5.17	5.57	5.91	5.59	5.24	6.44	8.34	/
Setting: Mix Tuned	**3.52**	4.50	4.39	5.15	5.60	5.99	5.71	5.33	6.48	8.34	/
Characteristics: #. Sample	< 500										
Proportion	57%										
Setting: HID-MIX (d)	**3.45**	4.34	4.22	4.23	5.02	6.05	5.90	5.79	7.10	8.92	/
Setting: Mix Tuned	**3.18**	4.38	4.28	4.29	5.05	6.05	5.97	5.79	7.13	8.88	/
Characteristics: #. Feature	#. Feature < 8										
Proportion	32%										
Setting: HID-MIX (d)	**3.45**	3.84	3.98	5.08	4.23	6.32	6.16	5.32	7.52	9.10	/
Setting: Mix Tuned	**3.27**	3.84	4.03	5.06	4.34	6.26	6.21	5.35	7.50	9.13	/
Characteristics: #. Feature	8 ≤ #. Feature < 16										
Proportion	38%										
Setting: HID-MIX (d)	**3.76**	4.26	4.44	4.61	6.31	5.75	5.39	5.69	6.61	8.17	/
Setting: Mix Tuned	**3.17**	4.33	4.49	4.74	6.33	5.81	5.58	5.78	6.64	8.14	/
Characteristics: #. Feature	#. Feature ≥ 16										
Proportion	30%										
Setting: HID-MIX (d)	**3.62**	5.19	4.41	4.17	5.05	5.93	5.81	5.64	6.33	8.84	/
Setting: Mix Tuned	**3.17**	5.22	4.48	4.14	5.05	6.05	5.90	5.69	6.45	8.84	/

Configuration Settings:

Two configurations are tested: Hid-Mix(default) and Mix-Tuned,with all existing models undergoing hyperparameter fine-tuning.

Can ExcelFormer be a sure bet solution across various types of datasets?

Table 3: Performance evaluation within several dataset subgroups. Performance rank (↓) within the datasets are reported. The best scores are in bold and the runners-up are underlined. "(d)": default hyperparameters; "(t)": finely tuned hyperparameters.



Model	ExcelFormer	FTT (t)	XGb (t)	Cat (t)	MLP (t)	DCNv2 (t)	AutoInt (t)	SAINT (t)	TransTab (d)	XTab (d)	TabPFN (t)
Characteristics: Task Type	Classification										
Proportion	51%										
Setting: HID-MIX (d)	**3.88**	4.88	5.97	5.77	6.61	6.38	6.63	6.07	6.31	9.50	<u>4.01</u>
Setting: Mix Tuned	**3.78**	4.91	5.95	5.79	6.60	6.39	6.71	6.10	6.37	9.46	<u>3.95</u>
Characteristics: Task Type	Regression										
Proportion	49%										
Setting: HID-MIX (d)	<u>3.81</u>	4.45	**3.43**	4.26	4.64	6.26	5.53	5.64	8.21	8.79	/
Setting: Mix Tuned	**3.17**	4.49	<u>3.53</u>	4.28	4.74	6.32	5.68	5.72	8.23	8.83	/
Characteristics: #. Sample	≥ 500										
Proportion	43%										
Setting: HID-MIX (d)	**3.85**	4.50	<u>4.38</u>	5.17	5.57	5.91	5.59	5.24	6.44	8.34	/
Setting: Mix Tuned	**3.52**	4.50	<u>4.39</u>	5.15	5.60	5.99	5.71	5.33	6.48	8.34	/
Characteristics: #. Sample	< 500										
Proportion	57%										
Setting: HID-MIX (d)	**3.45**	4.34	<u>4.22</u>	4.23	5.02	6.05	5.90	5.79	7.10	8.92	/
Setting: Mix Tuned	**3.18**	4.38	<u>4.28</u>	4.29	5.05	6.05	5.97	5.79	7.13	8.88	/
Characteristics: #. Feature	#. Feature < 8										
Proportion	32%										
Setting: HID-MIX (d)	**3.45**	<u>3.84</u>	3.98	5.08	4.23	6.32	6.16	5.32	7.52	9.10	/
Setting: Mix Tuned	**3.27**	<u>3.84</u>	4.03	5.06	4.34	6.26	6.21	5.35	7.50	9.13	/
Characteristics: #. Feature	8 ≤ #. Feature < 16										
Proportion	38%										
Setting: HID-MIX (d)	**3.76**	<u>4.26</u>	4.44	4.61	6.31	5.75	5.39	5.69	6.61	8.17	/
Setting: Mix Tuned	**3.17**	<u>4.33</u>	4.49	4.74	6.33	5.81	5.58	5.78	6.64	8.14	/
Characteristics: #. Feature	#. Feature ≥ 16										
Proportion	30%										
Setting: HID-MIX (d)	**3.62**	5.19	4.41	<u>4.17</u>	5.05	5.93	5.81	5.64	6.33	8.84	/
Setting: Mix Tuned	**3.17**	5.22	4.48	<u>4.14</u>	5.05	6.05	5.90	5.69	6.45	8.84	/

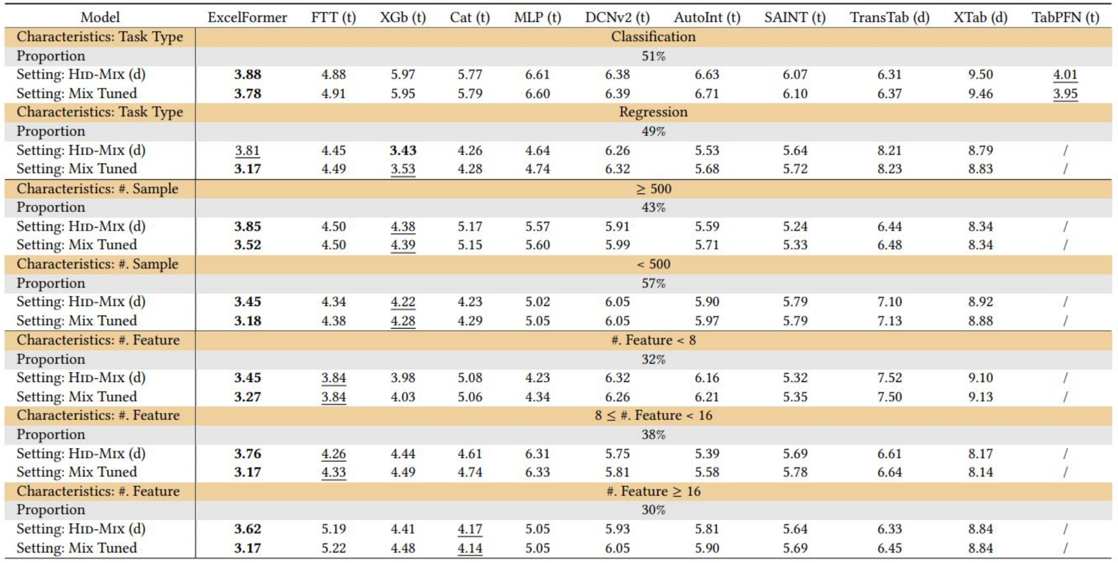
Performance Results:

As shown in Table, ExcelFormer performs best in all subgroups except regression tasks,where it slightly lags behind hyperparameter-tuned XGBoost.

Mix Tuned ExcelFormer outperforms other models in all subgroups, indicating no strong dataset type preference.

Can ExcelFormer be a sure bet solution across various types of datasets?

Table 3: Performance evaluation within several dataset subgroups. Performance rank (↓) within the datasets are reported. The best scores are in bold and the runners-up are underlined. "(d)": default hyperparameters; "(t)": finely tuned hyperparameters.



Model	ExcelFormer	FTT (t)	XGb (t)	Cat (t)	MLP (t)	DCNv2 (t)	AutoInt (t)	SAINT (t)	TransTab (d)	XTab (d)	TabPFN (t)
Characteristics: Task Type	Classification										
Proportion	51%										
Setting: Hid-Mix (d)	**3.88**	4.88	5.97	5.77	6.61	6.38	6.63	6.07	6.31	9.50	4.01
Setting: Mix Tuned	**3.78**	4.91	5.95	5.79	6.60	6.39	6.71	6.10	6.37	9.46	3.95
Characteristics: Task Type	Regression										
Proportion	49%										
Setting: Hid-Mix (d)	3.81	4.45	**3.43**	4.26	4.64	6.26	5.53	5.64	8.21	8.79	/
Setting: Mix Tuned	**3.17**	4.49	3.53	4.28	4.74	6.32	5.68	5.72	8.23	8.83	/
Characteristics: #. Sample	≥ 500										
Proportion	43%										
Setting: Hid-Mix (d)	**3.85**	4.50	4.38	5.17	5.57	5.91	5.59	5.24	6.44	8.34	/
Setting: Mix Tuned	**3.52**	4.50	4.39	5.15	5.60	5.99	5.71	5.33	6.48	8.34	/
Characteristics: #. Sample	< 500										
Proportion	57%										
Setting: Hid-Mix (d)	**3.45**	4.34	4.22	4.23	5.02	6.05	5.90	5.79	7.10	8.92	/
Setting: Mix Tuned	**3.18**	4.38	4.28	4.29	5.05	6.05	5.97	5.79	7.13	8.88	/
Characteristics: #. Feature	#. Feature < 8										
Proportion	32%										
Setting: Hid-Mix (d)	**3.45**	3.84	3.98	5.08	4.23	6.32	6.16	5.32	7.52	9.10	/
Setting: Mix Tuned	**3.27**	3.84	4.03	5.06	4.34	6.26	6.21	5.35	7.50	9.13	/
Characteristics: #. Feature	8 ≤ #. Feature < 16										
Proportion	38%										
Setting: Hid-Mix (d)	**3.76**	4.26	4.44	4.61	6.31	5.75	5.39	5.69	6.61	8.17	/
Setting: Mix Tuned	**3.17**	4.33	4.49	4.74	6.33	5.81	5.58	5.78	6.64	8.14	/
Characteristics: #. Feature	#. Feature ≥ 16										
Proportion	30%										
Setting: Hid-Mix (d)	**3.62**	5.19	4.41	4.17	5.05	5.93	5.81	5.64	6.33	8.84	/
Setting: Mix Tuned	**3.17**	5.22	4.48	4.14	5.05	6.05	5.90	5.69	6.45	8.84	/

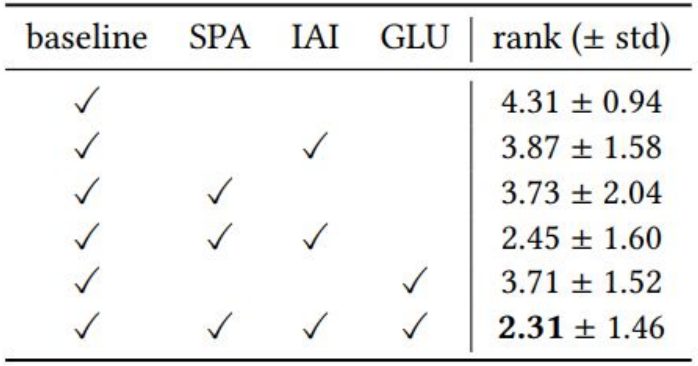
Summary:

What changes does ExcelFormer bring to tabular data prediction? Besides ExcelFormer,TapFPN,FTT,XGBoost,and CatBoost hold runner-up positions in different subgroups.Notably,TapFPN is only for classification tasks,CatBoost excels with numerous features, and FTT performs well within fewer than 16 features.

However,ExcelFormer shows strong performance across all dataset types,proving its status as a reliable solution for tabular

Ablation Study

Table 4: Additive study for the *semi-permeable attention* (SPA) and *interaction-attenuated initialization* (IAI), and the GLU based attentive module. No data augmentation.



baseline	SPA	IAI	GLU	rank (± std)
✓				4.31 ± 0.94
✓		✓		3.87 ± 1.58
✓	✓			3.73 ± 2.04
✓	✓	✓		2.45 ± 1.60
✓			✓	3.71 ± 1.52
✓	✓	✓	✓	**2.31** ± 1.46

Baseline model:The baseline model uses ExcelFormer with typical Kaiming initialization and a MLP-based FFN.

Impact of Architecture Design:SPA and IAI each improve baseline performance, with better results when used together.GLU also significantly enhances baseline performance,indicating these designs are suitable for tabular data prediction.

Best Results:ExcelFormer achieves the best results when all components(SPA, IAI,GLU) are used together.

Ablation Study

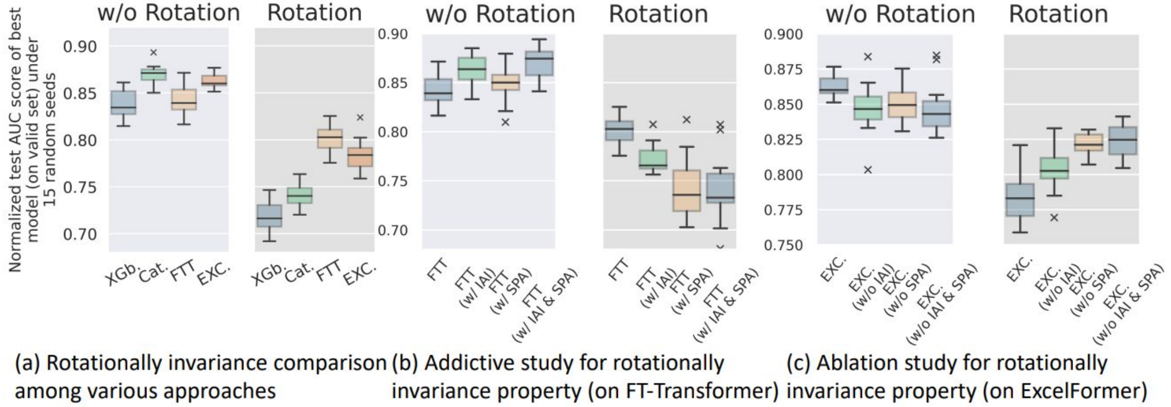


Figure 5: Model performances under random dataset rotations. Test accuracy scores have been normalized across datasets, and the boxes represent the distribution of scores across 20 random seeds. XGb.: XGboost, Cat.: Catboost, EXC.: ExcelFormer without data augmentation.

Rotational Variance in DNNs:

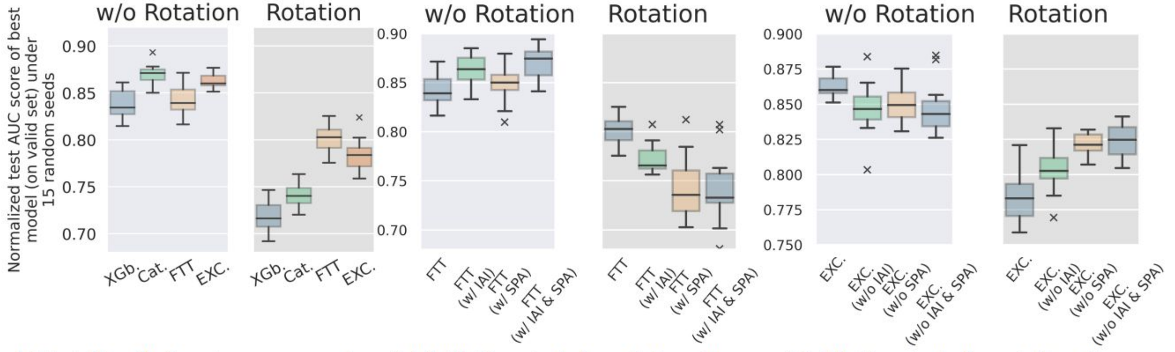
The main paper highlights that DNNs lack rotational variance,making them less efficient for tabular prediction tasks.This limitation motivated the proposal of SPA and IAI.

Evaluation of ExcelFormer:

The performance of ExcelFormer,with SPA and IAI,is assessed without data augmentation by randomly rotating datasets.The evaluation uses binary classification datasets with numerical features and fewer than 300 samples.

Uninformative features are added using Gaussian noise.

Ablation Study



(a) Rotationally invariance comparison among various approaches (b) Addictive study for rotationally invariance property (on FT-Transformer) (c) Ablation study for rotationally invariance property (on ExcelFormer)

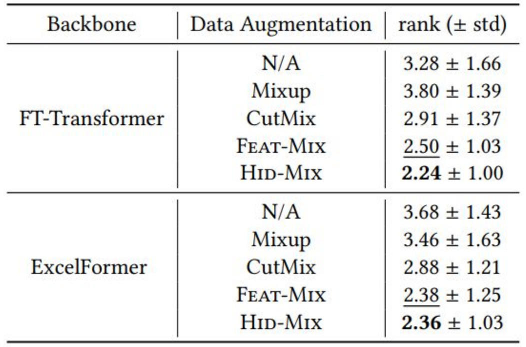
Figure 5: Model performances under random dataset rotations. Test accuracy scores have been normalized across datasets, and the boxes represent the distribution of scores across 20 random seeds. XGb.: XGboost, Cat.: Catboost, EXC.: ExcelFormer without data augmentation.

Configuration Settings:

Two configurations are tested: Hid-Mix(default) and Mix-Tuned,with all existing models undergoing hyperparameter fine-tuning.

Ablation Study

Table 5: Comparison among several data augmentation approaches: Mixup, CutMix, Feat-Mix, and Hid-Mix on FT-Transformer and ExcelFormer. The performance ranks are computed separately for different backbones.



Backbone	Data Augmentation	rank (± std)
FT-Transformer	N/A	3.28 ± 1.66
	Mixup	3.80 ± 1.39
	CutMix	2.91 ± 1.37
	Feat-Mix	2.50 ± 1.03
	Hid-Mix	**2.24** ± 1.00
ExcelFormer	N/A	3.68 ± 1.43
	Mixup	3.46 ± 1.63
	CutMix	2.88 ± 1.21
	Feat-Mix	2.38 ± 1.25
	Hid-Mix	**2.36** ± 1.03

Mixup Effect:

Mixup has minimal impact on performance and can sometimes be detrimental due to interpolation errors or overly smooth solution.

CutMix Performance:

CutMix consistently outperforms Mixup,approaching Feat-Mix performance but slightly inferior.

Importance of Feat-Mix:

Without considering feature importance,Feat-Mix may regress to Cutmix.Calculating feature importance is crucial to mitigate the impact of uninformative features,common in tabular datasets.

Hid-Mix Performance:Hid-Mix performs the best among all data augmentation techniques.

Outline

Conclusion

Novel Approach: Introduces ExcelFormer to address three key limitations of DNNs for tabular data.

Key Innovations:

- Semi-permeable attention (SPA) module.
- Interaction attenuated initialization.
- GLU-based feedforward network (FFN).
- Data augmentation techniques: Hid-Mix and Feat-Mix.

Performance: Outperforms existing GBM and DNN models without hyperparameter tuning.

Conclusion: ExcelFormer is a robust solution for tabular data, effective and user-friendly for both novices and experts.

謝謝大家~