

# SG-XDEAT: Sparsity-Guided Cross-Dimensional and Cross-Encoding Attention with Target-Aware Conditioning in Tabular Learning

Anonymous submission

## Abstract

We propose SG-XDEAT (Sparsity-Guided Cross-Dimensional and Cross-Encoding Attention with Target-Aware Conditioning), a novel framework designed for supervised learning on tabular data. At its core, SG-XDEAT employs a dual-stream encoder that decomposes each input feature into two parallel representations: a raw value stream and a target-conditioned (label-aware) stream. These dual representations are then propagated through a hierarchical stack of attention-based modules. SG-XDEAT integrates three key components: (i) cross-dimensional self-attention, which captures intra-view dependencies among features within each stream; (ii) cross-encoding self-attention, which enables bidirectional interaction between raw and target-aware representations; and (iii) an Adaptive Sparse Self-Attention (ASSA) mechanism, which dynamically suppresses low-utility tokens by driving their attention weights toward zero—thereby mitigating the impact of noise. Empirical results on multiple public benchmarks show consistent gains over strong baselines, confirming that jointly modeling raw and target-aware views—while adaptively filtering noise—yields a more robust deep tabular learner.

## Introduction

Tabular data plays a central role in numerous real-world applications that span domains such as medicine, finance, and transportation (Shwartz-Ziv and Armon 2022; Borisov et al. 2022; Somvanshi et al. 2024; Ye et al. 2024). Despite its prevalence, learning from tabular data poses significant challenges for deep learning models due to its lack of spatial or sequential structure and the presence of heterogeneous feature types (Somvanshi et al. 2024). Consequently, Gradient-Boosted Decision Trees (GBDTs) have long remained the dominant choice for tabular tasks (Shwartz-Ziv and Armon 2022). However, recent years have seen growing interest in deep learning techniques in this domain (Hwang and Song 2023). Attention-based models like FT-Transformer (Gorishniy et al. 2021) capture feature interactions effectively. Graph-based methods such as GANDALF (Joseph and Raj 2022) exploit relational structure but often struggle to scale. Meanwhile, language-model approaches like PTab (Liu, Yang, and Wu 2022) and TabLLM (Hegselmann et al. 2023) convert tabular data into text so they can leverage pretrained language models, delivering strong performance at a high computational cost (Somvanshi et al. 2024).

Recent research has explored the potential benefits of incorporating label information into feature encoding to improve model performance in tabular data (Jiang et al. 2025). For categorical variables, label-aware techniques, such as target encoding (Micci-Barreca 2001) and Generalized Linear Mixed Model (GLMM) (Stroup, Ptukhina, and Garai 2024), outperform one-hot, hashing, and ordinal encoding schemes, particularly when classes are imbalanced or categories are high in cardinality (Pargent et al. 2022). For numerical variables, methods like Piecewise Linear Encoding (PLE) (Gorishniy, Rubachev, and Babenko 2022) split the value range using label-guided thresholds and learn an embedding for each segment, yielding substantial gains for multilayer perceptrons (MLP)- and Transformer-based architectures. Overall, label-conditioned encodings expose informative structure that unsupervised approaches miss.

Deep learning models for tabular data are particularly vulnerable to irrelevant or weakly informative features (McElfresh et al. 2023). Early efforts to address this issue relied primarily on data-driven filtering techniques, such as ranking features by mutual information or correlation and removing the least informative (Li et al. 2017). However, more recent research has shifted toward model-driven approaches that embed sparsity directly into the network architecture. Instead of preprocessing inputs to filter out noise, these methods enable the model to learn which features to disregard during training (Arik and Pfister 2021; Margeloiu et al. 2023). Additionally, some approaches modify activation functions themselves, for example, replacing softmax with rectified-linear variants such as squared ReLU (Zhou et al. 2024) or leaky ReLU (Fiedler 2021), to further enhance robustness against noisy inputs. Overall, this transition from input filtering to architectural adaptation reflects a growing emphasis on making deep learning models inherently resilient to non-informative features.

To address the challenges mentioned above, namely susceptibility to irrelevant features and limited ability to leverage label information, we propose SG-XDEAT, a model that combines supervised feature representations with architectural mechanisms for noise suppression. SG-XDEAT adopts a dual-stream architecture that separately encodes raw inputs and label-aware representations to leverage label information. It incorporates attention modules to capture both cross-feature and cross-encoding dependencies. Furthermore, a

hybrid sparse attention mechanism is introduced to dynamically downweight uninformative features.

We summarize the contributions of our paper as follows.

- **Target-Aware Conditioning** We introduce a tokenization stream with label-guided encodings: PLE for numerical features (Gorishniy, Rubachev, and Babenko 2022) and tree-based encoding (Niculescu-Mizil et al. 2009) for categorical ones.
- **Dual-Path Transformer** We introduce two parallel attention streams: one captures cross-feature interactions, while the other learn cross-encoding dependencies.
- **Adaptive Sparse Self-Attention** We refer to a hybrid attention module that combines softmax-based branch and squared-ReLU-based sparse branch (Zhou et al. 2024) to suppress noisy features while maintaining global context.
- **Comprehensive Evaluation** We also verify the effectiveness and efficiency of SG-XDEAT, effectively bridging the gap between deep models and GBDTs.

## Related Work

**Traditional Methods** Gradient Boosting Decision Trees (GBDT), including XGBoost (Chen and Guestrin 2016), LightGBM (Ke et al. 2017), and CatBoost (Prokhorenkova et al. 2018), are widely used for tabular data due to their ability to handle mixed feature types, missing values, and non-linear patterns. These models offer strong predictive performance and remain the default choice in many structured data applications. In particular, CatBoost (Prokhorenkova et al. 2018) demonstrates state-of-the-art performance across various benchmark datasets, consistently outperforming both XGBoost (Chen and Guestrin 2016) and LightGBM (Ke et al. 2017) in terms of accuracy and stability. Despite their impressive performance, GBDT models are prone to overfitting, where deep trees tend to memorize the training data, including noise and irrelevant features, leading to poor generalization on unseen data. (Costa and Pedreira 2023)

**Deep Learning Models** Early work on deep learning for structured data relied on MLPs, yet these simple feed-forward networks rarely matched the performance of GBDT. Researchers then imported design principles from computer vision: residual MLPs in the style of ResNet, when carefully tuned, proved unexpectedly competitive (Gorishniy et al. 2021). Attention-based architectures soon followed—AutoInt (Song et al. 2019) replaces hand-crafted feature crosses with multi-head self-attention capable of capturing high-order interactions, while DCN-V2 (Wang et al. 2021) augments explicit low-rank cross layers to satisfy the latency and scale requirements of industrial ranking systems. A parallel line of research grafts tree logic into neural networks: Neural Oblivious Decision Ensembles (NODE) (Popov, Morozov, and Babenko 2019) integrate ensembles of oblivious decision trees within a fully differentiable scaffold, closing much of the gap to GBDTs while preserving interpretability. The latest advance, FT-Transformer (Gorishniy et al. 2021), pairs a feature tokenizer with a pre-norm Transformer encoder and has become the standard backbone in tabular learning.

**Target Aware Encoding** Despite a variety of encoding strategies, traditional methods such as one-hot and ordinal encoding (Bird et al. 2014) do not incorporate any supervision from the target label. To address this, recent work has introduced label-aware encodings, particularly for categorical features (Stroup, Ptukhina, and Garai 2024; Larionov 2020; Zeng 2014). For example, regularized target encoders like the M-estimate and S-shrink variants (Micci-Barreca 2001) assign each category a smoothed estimate of the target mean, while DecisionTreeEncoder (Niculescu-Mizil et al. 2009) maps inputs to class probabilities based on leaf nodes of shallow trees. Similar ideas extend to numerical features: Piecewise Linear Encoding with Target guidance (PLE-T) (Gorishniy, Rubachev, and Babenko 2022) uses supervised splits to construct interval-based embeddings aligned with label distribution.

However, these methods treat raw and target-aware representations as separate alternatives and do not model their interaction. In contrast, our approach introduces a dual-stream architecture that processes both representations in parallel and captures dependencies between them.

**Noise-Robust for Deep Learning Models** Recent work has increasingly focused on integrating noise suppression directly into the model architecture. For example, ExcelFormer (Chen et al. 2023) introduces a semi-permeable attention mask, built from pre-computed feature importance, that stops weak columns from sending information to stronger ones. The Leaky Gate applies a per-feature linear layer followed by a Leaky ReLU, down-weighting low-value inputs on the fly while making each feature’s impact transparent (Fiedler 2021). In computer vision, Adaptive Sparse Self-Attention (Zhou et al. 2024) adds a parallel squared-ReLU branch that zeros small scores while a standard softmax branch preserves global context.

Deep tabular models are often susceptible to irrelevant or weakly informative features (McElfresh et al. 2023), which can undermine both robustness and performance. To address this issue, we integrate the Adaptive Sparse Self-Attention (Zhou et al. 2024), originally developed for vision tasks. By combining the squared-ReLU and softmax branches, our model learns to suppress low-utility attention scores, reducing the influence of noisy inputs without the need for manual feature selection.

**Dual-Path Transformer** In multivariate time-series forecasting, it is increasingly common to divide attention along separate time and variable axes. For example, CrossFormer (Zhang and Yan 2023) embeds each series as a time-by-variable grid and applies a Two-Stage Attention (TSA) block: Stage 1 attends along time, Stage 2 along variables, capturing cross-time and cross-dimension patterns. TimeXer (Wang et al. 2024) builds on a similar idea, using self-attention within variable axes and cross-attention to fuse information from variable and time axes.

In our tabular setup, each feature has two parallel representations after target-aware encoding: the raw value and its label-guided version. These two representations carry different types of signals, and capturing their dependencies is not straightforward. We discard a design such as CrossFormer (Zhang and Yan 2023), which applies attention in a

fixed order. Instead, we use a dual-path structure: one path attends across features within a view (i.e., within the raw value or within the label-guided version), while the other models cross-view interactions. This approach avoids arbitrary ordering and allows the model to better integrate the information from both representations.

## Methodology

Figure 1 presents the overall architecture of **SG-XDEAT**. The raw input features are first processed by the **Dual-Stream Embedding** block, which generates two parallel streams: a *raw stream* that retains the original feature values, and a *target-aware stream* that encodes label-informed representations. Next, the **Token Extension** step adds global learnable tokens to both streams, enabling the model to incorporate instance-level contextual information. These enriched token streams are then passed through a **Dual-Path Transformer** stack. In each layer, (i) *Cross-Encoding Self-Attention* aligns the raw and target-aware views of each feature, while (ii) *Cross-Dimension Self-Attention*, equipped with an **Adaptive Sparse Self-Attention** module, facilitates information exchange across features while filtering out less informative tokens. After the transformer layers, the global tokens from both streams are mean-pooled and merged via the **Token Aggregation** module before being passed to the final classifier. Collectively, these components integrate raw and label-guided information, reduce noise, and produce a rich representation for tabular classification or regression. The following sections describe each key module in detail.

### Dual-Stream Embedding

Each input sample is processed through two parallel embedding pathways. The *raw stream*  $\mathcal{R}$  preserves the original values of all features, while the *target-aware stream*  $\mathcal{T}$  transforms features based on the label information.

In the target-aware stream, categorical features are encoded using a **DecisionTreeEncoder** (Niculescu-Mizil et al. 2009): a shallow decision tree is trained for each feature using the target label, and the class probability at the leaf node replaces the raw category. For numerical features, we use **PLE-T** (Gorishniy, Rubachev, and Babenko 2022), which fits a single-feature decision tree to the target. The resulting leaf bins define discrete intervals, and the corresponding bin index is mapped to a learnable embedding vector. In the raw stream, feature values are preserved without modification.

Both streams are then passed through a tokenizer  $E(\cdot)$ , which projects each feature into a  $d$ -dimensional embedding space. This yields two embedding matrices:

$$\begin{aligned}\mathbf{R} &= E(\mathcal{R}) \in \mathbb{R}^{F \times d} \quad (\text{Raw Embeddings}) \\ \mathbf{T} &= E(\mathcal{T}) \in \mathbb{R}^{F \times d} \quad (\text{Target-Aware Embeddings})\end{aligned}$$

where  $F$  is the number of features and  $d$  is the embedding dimension. The raw and target-aware embeddings are then concatenated along a new stream dimension to form  $\mathbf{D}$ , integrating both streams for Token Extension module:

$$\mathbf{D} = \text{Concat}(\mathbf{T}, \mathbf{R}) \in \mathbb{R}^{2 \times F \times d}$$

### Token Extension

We introduce two types of learnable global tokens. The *dimension-level tokens*  $\mathbf{g}_{\text{dim}} \in \mathbb{R}^{2 \times d}$  are prepended to the raw and target-aware representations to model cross-feature dependencies. The *encoding-level tokens*  $\mathbf{g}_{\text{enc}} \in \mathbb{R}^{F \times d}$  capture interactions between the two encoding views. The resulting extended representations are:

$$\begin{aligned}\mathbf{D}_{\text{dim}} &= [\mathbf{g}_{\text{dim}}; \mathbf{D}] \in \mathbb{R}^{2 \times (F+1) \times d} \\ \mathbf{D}_{\text{enc}} &= [\mathbf{g}_{\text{enc}}; \mathbf{D}] \in \mathbb{R}^{3 \times F \times d}\end{aligned}$$

Here,  $\mathbf{D}_{\text{dim}}$  appends  $\mathbf{g}_{\text{dim}}$  along the feature axis, while  $\mathbf{D}_{\text{enc}}$  includes  $\mathbf{g}_{\text{enc}}$  as another token group. These augmentations enable models to capture both cross-feature patterns and cross-view relations.

### Dual-Path Transformer

The Dual-Path Transformer comprises two attention modules—*Cross-Encoding Self-Attention* and *Cross-Dimension Self-Attention*—that extract complementary information from  $\mathbf{D}_{\text{dim}}$  and  $\mathbf{D}_{\text{enc}}$ .

**Cross-Encoding Self-Attention** This module operates on the encoding-level token group  $\mathbf{D}_{\text{enc}} \in \mathbb{R}^{3 \times F \times d}$ , which includes global encoding tokens, raw feature tokens, and target-aware tokens. To enable feature-wise attention, the tensor is first reshaped to  $\mathbb{R}^{F \times 3 \times d}$  to treat each feature independently. A standard multi-head self-attention block is applied independently to each of the  $F$  features to model interactions among the three token types. For a given feature, the attention is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V,$$

where the query, key, and value matrices are computed:  $Q = \mathbf{D}_{\text{enc}}W_Q$ ,  $K = \mathbf{D}_{\text{enc}}W_K$ , and  $V = \mathbf{D}_{\text{enc}}W_V$ , with  $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$  being learnable parameters.

This setup enables the model to learn encoding-level relations for each feature and use the global token to capture interactions between the raw and target-aware representations.

**Cross-Dimension Self-Attention** This module builds on the *Adaptive Sparse Self-Attention* mechanism, enabling the model to focus on informative features while filtering out less relevant ones. It operates on dimension-level tokens  $\mathbf{D}_{\text{dim}} \in \mathbb{R}^{2 \times (F+1) \times d}$ , which consist of raw and target-aware token groups, each prepended with a global dimension token. To capture cross-feature dependencies, the raw and target-aware representations are processed with attention applied across the feature dimension within each group.

Two attention branches are then applied in parallel:

- **Softmax Branch:** This branch applies scaled dot-product attention, capturing fine-grained dependencies:

$$\text{Emb}_{\text{soft}} = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V$$

- **Squared ReLU Branch:** To enhance sparsity and suppress noisy signals, this branch replaces the softmax with

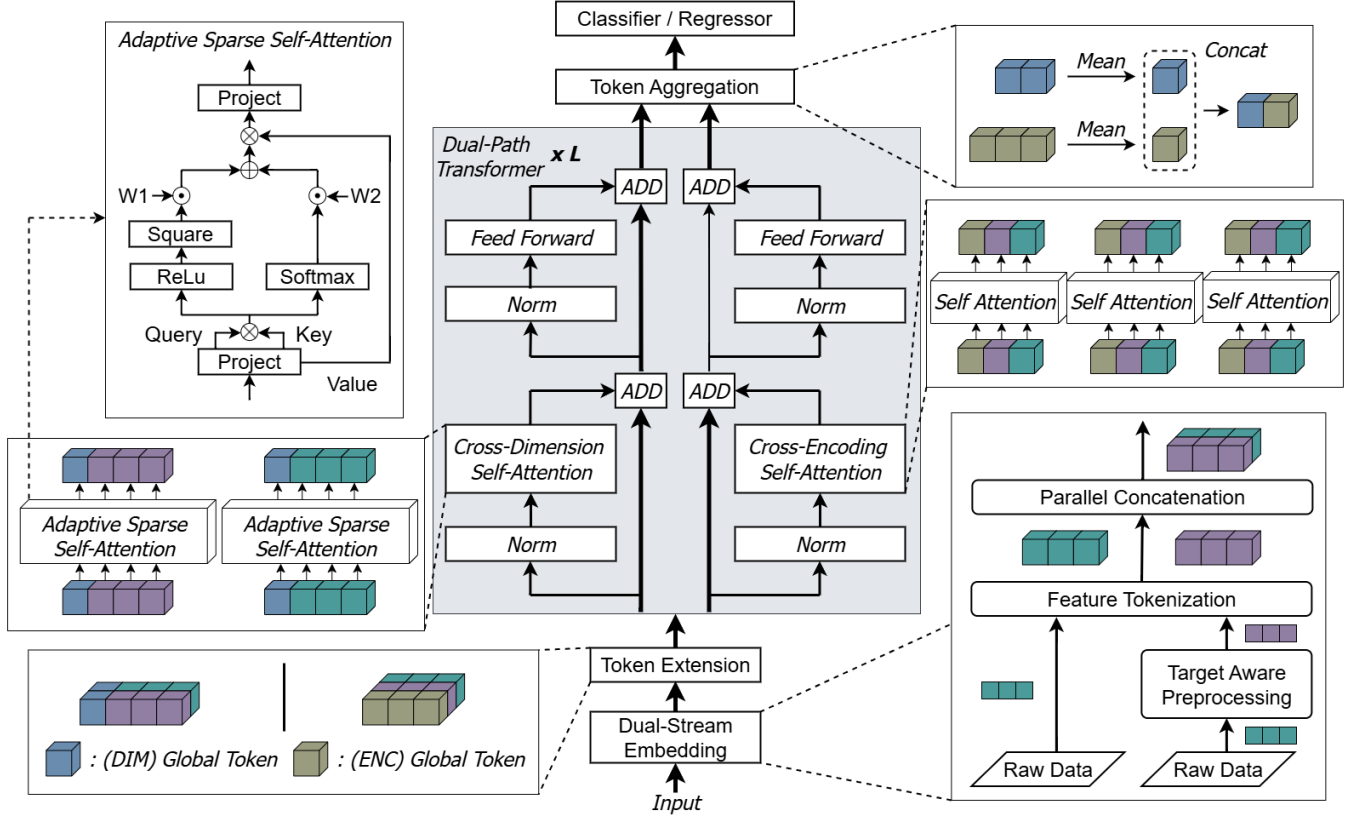


Figure 1: The overview of SG-XDEAT.  $L$  is the layer number.

a squared ReLU activation:

$$\mathbf{Emb}_{\text{relu}} = \left[ \text{ReLU} \left( \frac{QK^\top}{\sqrt{d}} \right) \right]^2 V$$

Given the input tokens  $\mathbf{D}_{\text{dim}} \in \mathbb{R}^{(F+1) \times d}$  from one view, we compute the query, key, and value matrices:

$$Q = \mathbf{D}_{\text{dim}} W_Q, \quad K = \mathbf{D}_{\text{dim}} W_K, \quad V = \mathbf{D}_{\text{dim}} W_V$$

where  $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$  are learnable parameters.

The outputs from both branches are combined using learned weights. Let  $a_1, a_2 \in \mathbb{R}$  be scalar parameters (initialized to 1), and define the normalized weights  $w_1, w_2$  using a softmax over  $(a_1, a_2)$ :

$$w_n = \frac{e^{a_n}}{e^{a_1} + e^{a_2}}, \quad n = \{1, 2\}$$

The final attention output is computed as a weighted sum:

$$\mathbf{D}_{\text{dim}} = w_1 \cdot \mathbf{Emb}_{\text{soft}} + w_2 \cdot \mathbf{Emb}_{\text{relu}}$$

This approach enables the model to weigh standard attention against a denoising pathway, enhancing the relevance of feature interactions.

### Token Aggregation

The model performs mean pooling independently over the global tokens, and the pooled vectors are then concatenated

to form the final representation:

$$\mathbf{g}_{\text{dim}}^{\text{mean}} = \text{Mean}(\mathbf{g}_{\text{dim}}) \in \mathbb{R}^d, \quad \mathbf{g}_{\text{enc}}^{\text{mean}} = \text{Mean}(\mathbf{g}_{\text{enc}}) \in \mathbb{R}^d$$

$$\mathbf{z} = \text{Concat}(\mathbf{g}_{\text{dim}}^{\text{mean}}, \mathbf{g}_{\text{enc}}^{\text{mean}}) \in \mathbb{R}^{2d}$$

This fused vector  $\mathbf{z}$  is subsequently passed to the classifier.

### Classifier / Regressor

The aggregated vector  $\mathbf{z} \in \mathbb{R}^{2d}$  is passed to the classifier:

$$\hat{y} = \text{Linear}(\text{ReLU}(\text{LayerNorm}(\mathbf{z}))).$$

## Experiments

### Experimental Setup

**Dataset** To evaluate the generalizability of our model, we use a set of benchmark datasets covering regression, binary classification, and multiclass classification tasks. The datasets vary in feature composition—some are only numerical, others entirely categorical, and some include both. The selected datasets are Gesture Phase (GE) (Madeo, Lima, and Peres 2013), KDD Internet Usage (KD) (Kehoe and Pitkow 1996), Adult (AD) (Kohavi et al. 1996), California Housing (CA) (Pace and Barry 1997), and Higgs Small (HI) (Baldi, Sadowski, and Whiteson 2014). Detailed statistics and configuration are provided in Table 1 and Appendix A.

Name	Abbr	#Train	#Val	#Test	#Num	#Cat	Task Type	Batch Size
Gesture Phase	GE	6,318	1,580	1,975	32	0	Multiclass	128
KDD Internet Usage	KD	6,468	1,618	2,022	0	68	Binclass	128
Adult	AD	26,048	6,513	16,281	6	8	Binclass	256
California Housing	CA	13,209	3,303	4,128	8	0	Regression	256
Higgs Small	HI	62,752	15,688	19,610	28	0	Binclass	512

Table 1: Dataset statistics

Datasets	CA	GE	AD	KD	HI	–
Metrics	RMSE ↓	Accuracy ↑				Rank ± std
XGBoost	<b>0.451 (1.0)</b>	<b>0.685 (1.0)</b>	0.871 (2.0)	0.902 (3.0)	0.727 (4.0)	2.2 ± 1.17
MLP	0.499 (7.0)	0.651 (5.0)	0.858 (6.0)	0.892 (7.0)	0.725 (7.0)	6.4 ± 0.80
Resnet	0.489 (5.0)	0.657 (4.0)	0.852 (7.0)	0.894 (6.0)	<b>0.734 (1.0)</b>	4.6 ± 2.06
DCN-V2	0.488 (4.0)	0.634 (6.0)	0.859 (4.5)	0.899 (4.0)	0.726 (5.5)	4.8 ± 0.81
AutoInt	0.490 (6.0)	0.602 (7.0)	0.859 (4.5)	0.898 (5.0)	0.726 (5.5)	5.6 ± 0.86
FT-Transformer	0.472 (3.0)	0.677 (2.0)	0.861 (3.0)	<b>0.903 (1.5)</b>	0.732 (2.5)	2.4 ± 0.58
SG-XDEAT	0.454 (2.0)	0.675 (3.0)	<b>0.872 (1.0)</b>	<b>0.903 (1.5)</b>	0.732 (2.5)	<b>2.0 ± 0.71</b>

Table 2: Comparison of performance across various benchmark models. Numbers in parentheses denote ranks of performance, and the **best** and second-best results are highlighted.

**Baseline** We evaluated our approach against a range of baselines commonly used for tabular data. Traditional methods include gradient boosting models such as **XG-Boost** (Chen and Guestrin 2016). Deep learning baselines include a standard **MLP** (Gorishniy et al. 2021) and several advanced architectures: **ResNet** (Gorishniy et al. 2021), **DCN-V2** (Wang et al. 2021), **AutoInt** (Song et al. 2019), and **FT-Transformer** (Gorishniy et al. 2021). This selection ensures a comprehensive comparison across both conventional and modern learning paradigms.

**Implementation Details** All experiments were implemented in Python 3.10 using PyTorch 2.5.1 and executed on an NVIDIA RTX 4090 GPU with 24GB of memory.

All deep learning models were evaluated using the same preprocessing: quantile transformation for input features (via Scikit-learn). In addition, regression targets were standardized across all methods.

All models were trained using the AdamW optimizer, along with a cosine annealing scheduler. A linear warm-up strategy was applied during the first 10 epochs, and each model was trained for a maximum of 200 epochs. Early stopping was employed with a patience of 15 epochs based on validation performance.

The hyperparameters of the models were tuned individually for each dataset using a multi-objective optimization strategy based on the Optuna framework (Akiba et al. 2019). The optimization process aimed to balance performance on both the training and validation datasets to ensure good generalization. To promote reproducibility and consistent results, we set the random seed to 42 during the search process. The search ranges and grids used for hyperparameter tuning are provided in Appendix B.

For classification tasks, accuracy was used as the evaluation metric, while for regression tasks, root mean squared error (RMSE) was used. Once the optimal configuration was found, each model was retrained 15 times (using random seeds from 0 to 14), and the final results were reported as the average over these runs.

## Experimental Results

**Performance Comparison** Table 2 presents a comprehensive comparison of benchmark models across five tabular datasets. The proposed model, SG-XDEAT, achieves state-of-the-art performance on the Adult (AD) dataset with an accuracy of 0.872, and obtains the second-best RMSE on the California Housing (CA) dataset (0.454)—a result very close to the best-performing XGBoost (0.451), while substantially outperforming all other deep learning architectures. It also demonstrates competitive results on the remaining datasets, including a joint-best accuracy on the KDD Internet Usage (KD) dataset (0.903). These results are consistent with the findings reported in the FT-Transformer research, where XGBoost is shown to perform strongly on certain datasets, and FT-Transformer also exhibits competitive performance. Among all deep learning-based methods, SG-XDEAT clearly stands out in most datasets and achieves the best average rank ( $2.0 \pm 0.71$ ), underscoring its robustness and strong generalizability. More detailed, one-sided Wilcoxon Test to statistically confirm these findings and further evaluate the significance of differences between SG-XDEAT and other models will be provided in Appendix C.

**Ablation Analysis of Architectural Components** To evaluate the contributions of the proposed architectural modules, we conducted an ablation study comparing three vari-

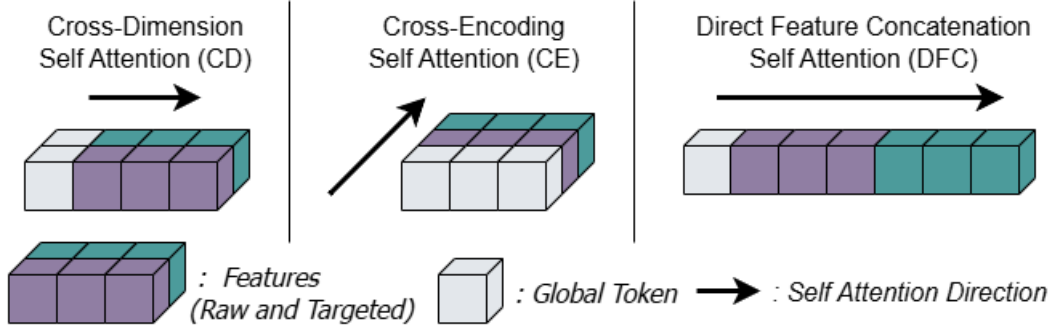


Figure 2: Illustration of architectural variants used in the ablation study.

Datasets	CA	GE	AD	KD	HI
Metrics	RMSE ↓	Accuracy ↑			
DFC	0.480	0.649	0.869	0.902	<b>0.732</b>
CD	0.463	<b>0.678</b>	0.871	<b>0.903</b>	<b>0.732</b>
CE	0.459	0.626	<b>0.872</b>	0.899	0.727
CD + CE	<b>0.454</b>	0.675	<b>0.872</b>	<b>0.903</b>	<b>0.732</b>

Table 3: Ablation results for architectural components (visualized in Figure 2). **CD** = Cross-Dimension, **CE** = Cross-Encoding, **DFC** = Direct Feature Concatenation. **Best** results are highlighted.

ants of our model: Cross-Dimension Self Attention (CD), Cross-Encoding Self Attention (CE), and their combination (CD+CE). Additionally, we include Direct Feature Concatenation (DFC) as a baseline strategy, which represents the most common approach—applying standard self-attention to concatenated feature representations. It is important to note that Cross-Dimension Self Attention (CD) in this ablation does not incorporate the adaptive sparsity mechanism used in the SG-XDEAT model. The architectural designs of these variants are illustrated in Figure 2.

As shown in Table 3, the combined configuration CD+CE consistently achieves the best performance across most datasets, including the lowest RMSE on CA (0.454) and the highest accuracy on AD (0.872), KD (0.903), and HI (0.732), demonstrating the complementary nature of the two mechanisms. Individually, both CD and CE contribute to performance improvements: CD, which focuses on feature-level fusion, is especially effective on GE, KD, and HI, while CE—which emphasizes encoding-level fusion—achieves the highest accuracy on AD and also performs best on CA. These observations indicate that different datasets benefit from different forms of interaction modeling, and thus, jointly considering both feature-level and encoding-level fusion is crucial for achieving robust and generalizable performance across diverse tabular tasks.

Furthermore, the baseline DFC configuration, which lacks explicit modeling of both feature-wise and encoding-wise dependencies, performs worse than CD+CE across most

Datasets	CA	GE	AD	KD	HI
Metrics	RMSE ↓	Accuracy ↑			
Raw	0.483	<u>0.665</u>	0.859	<b>0.903</b>	<b>0.732</b>
Targeted	<u>0.479</u>	0.655	0.863	0.888	<b>0.732</b>
DFC	0.480	0.649	<u>0.869</u>	0.902	<b>0.732</b>
CD + CE	<b>0.454</b>	<b>0.675</b>	<b>0.872</b>	<b>0.903</b>	<b>0.732</b>

Table 4: Performance comparison between different input strategies. The **Raw** setting uses original features only, while the **Targeted** variant incorporates label-dependent encoding. The **DFC** and **CD+CE** methods are visualized in Figure 2. **Best** and second-best results are highlighted.

datasets—particularly on CA and GE. These results demonstrate that both CD and CE play an important role in improving model performance, leading to a more effective and robust architecture overall.

**Analysis of Different Input Strategies** To evaluate the impact of target-aware information on input representations, we compare two distinct streams: the raw stream, which uses the original input features, and the target-aware stream, which integrates target information. As shown in Table 4, the effectiveness of each stream varies across datasets. The raw stream performs better on GE and KD, suggesting that preserving the original feature semantics is advantageous for these cases. On the other hand, the target-aware stream outperforms on CA and AD, demonstrating that incorporating label-aware context can improve discriminative power. For the HI dataset, both streams show similar results.

These findings emphasize that different datasets respond uniquely to raw and target-aware representations. However, the **DFC** method, which directly combines the raw and target-aware streams, fails to capture the dependencies between them effectively. In contrast, **CD+CE** can capture both cross-view and cross-feature interactions and ultimately improve performance.

**Effectiveness of Adaptive Sparse Self-Attention** To assess the robustness of Adaptive Sparse Self-Attention (ASSA) under varying levels of feature redundancy, we con-



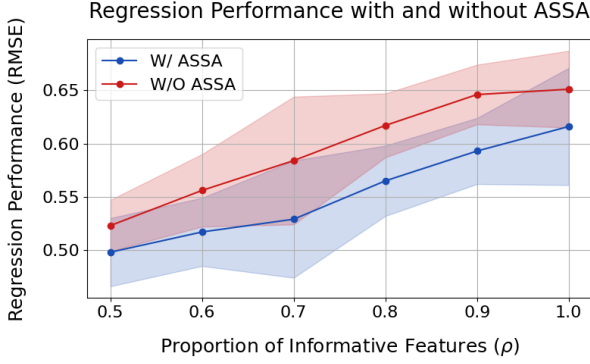


Figure 3: Regression performance (RMSE) on synthetic datasets with and without ASSA. A lower  $\rho$  implies a higher proportion of irrelevant (noisy) features. Lower RMSE indicates better predictive performance.

struct a synthetic regression benchmark in which only a controlled subset of the input features is informative. Each input sample is represented as  $x \in \mathbb{R}^d$  with  $d = 100$ , and a proportion  $\rho \in \{0.5, 0.6, \dots, 1.0\}$  of the features contribute to the target. The number of informative dimensions is defined as  $d_{\text{useful}} = \lfloor \rho \cdot d \rfloor$ , and the informative subvector is given by  $x_{\text{useful}} \in \mathbb{R}^{d_{\text{useful}}}$ , consisting of the first  $d_{\text{useful}}$  elements of  $x$ . The remaining  $(1 - \rho) \cdot d$  features are uninformative and serve as distractors. We focus on  $\rho \geq 0.5$  to maintain a meaningful signal-to-noise ratio, enabling attention-based models to effectively exploit informative features and better reflect practical scenarios.

The target value  $y \in \mathbb{R}$  is generated by applying a fixed, randomly initialized multi-layer perceptron (MLP) to the informative subvector:

$$y = f_{\text{MLP}}(x_{\text{useful}}).$$

Subsequently, standard normalization is applied:

$$y \leftarrow \frac{y - \mu_y}{\sigma_y},$$

where  $\mu_y$  and  $\sigma_y$  denote the mean and standard deviation of the target values computed from the training set.

This setup enables us to assess whether ASSA can suppress attention to irrelevant features and focus on informative ones. Details of the synthetic dataset generation and model architecture settings are provided in Appendix C.

Figure 3 illustrates the impact of ASSA across varying levels of feature redundancy. Across all settings of  $\rho$ , models equipped with ASSA consistently achieve lower RMSE compared to their counterparts without ASSA. This suggests that the adaptive sparsity mechanism enables the model to focus on relevant inputs even in the presence of substantial noise, leading to more robust and accurate predictions. Interestingly, even when  $\rho = 1$ , meaning that all features are informative, the model with ASSA still outperforms the model without it. This may be attributed to ASSA’s ability to better prioritize and focus on the most relevant features, even in scenarios where all features contribute useful information.

Therefore, ASSA’s mechanism of enforcing sparse attention might still enhance model performance, even when there is no explicit need to filter out irrelevant features.

## Conclusion

We introduced SG-XDEAT, a sparsity-guided attention framework designed for deep learning on tabular data. By capturing both cross-feature and cross-encoding dependencies through a dual-path attention design, SG-XDEAT effectively leverages raw and label-informed representations. The incorporation of an adaptive sparse attention mechanism further improves robustness by suppressing noisy or low-relevance signals. Empirical results across a range of benchmarks show that SG-XDEAT consistently outperforms strong baselines, helping close the gap between deep models and gradient-boosted decision trees. These results underscore the benefit of integrating label-aware encoding with structured attention for tabular prediction tasks.

## Acknowledgments

This research was supported by the National Science and Technology Council of Taiwan (grant nos. NSTC 114-2221-E-A49 -061) and the Higher Education Sprout Project of National Yang Ming Chiao Tung University and the Ministry of Education, Taiwan (grant no. CGMH-NYCU-114-CORPG2P0072). The funders had no role in the study design and procedures; data collection, management, analysis, and interpretation; manuscript preparation, review, and approval; or the decision to submit the manuscript for publication.

## References

- Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; and Koyama, M. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2623–2631.
- Arik, S. Ö.; and Pfister, T. 2021. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 6679–6687.
- Baldi, P.; Sadowski, P.; and Whiteson, D. 2014. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5(1): 4308.
- Bird, C. M.; Berens, S. C.; Horner, A. J.; and Franklin, A. 2014. Categorical encoding of color in the brain. *Proceedings of the National Academy of Sciences*, 111(12): 4590–4595.
- Borisov, V.; Leemann, T.; Seßler, K.; Haug, J.; Pawelczyk, M.; and Kasneci, G. 2022. Deep neural networks and tabular data: A survey. *IEEE transactions on neural networks and learning systems*, 35(6): 7499–7519.
- Chen, J.; Yan, J.; Chen, Q.; Chen, D. Z.; Wu, J.; and Sun, J. 2023. Excelformer: A neural network surpassing gbdt on tabular data. *arXiv preprint arXiv:2301.02819*.
- Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd*

*international conference on knowledge discovery and data mining*, 785–794.

Costa, V. G.; and Pedreira, C. E. 2023. Recent advances in decision trees: an updated survey. *Artificial Intelligence Review*, 56(5): 4765–4800.

Fiedler, J. 2021. Simple modifications to improve tabular neural networks. *arXiv preprint arXiv:2108.03214*.

Gorishniy, Y.; Rubachev, I.; and Babenko, A. 2022. On embeddings for numerical features in tabular deep learning. *Advances in Neural Information Processing Systems*, 35: 24991–25004.

Gorishniy, Y.; Rubachev, I.; Khrulkov, V.; and Babenko, A. 2021. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*, 34: 18932–18943.

Hegselmann, S.; Buendia, A.; Lang, H.; Agrawal, M.; Jiang, X.; and Sontag, D. 2023. Tabllm: Few-shot classification of tabular data with large language models. In *International conference on artificial intelligence and statistics*, 5549–5581. PMLR.

Hwang, Y.; and Song, J. 2023. Recent deep learning methods for tabular data. *Communications for Statistical Applications and Methods*, 30(2): 215–226.

Jiang, J.-P.; Liu, S.-Y.; Cai, H.-R.; Zhou, Q.; and Ye, H.-J. 2025. Representation learning for tabular data: A comprehensive survey. *arXiv preprint arXiv:2504.16109*.

Joseph, M.; and Raj, H. 2022. GANDALF: gated adaptive network for deep automated learning of features. *arXiv preprint arXiv:2207.08548*.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Kehoe, C. M.; and Pitkow, J. E. 1996. Surveying the territory: GVU’s five WWW user surveys. *The World Wide Web Journal*, 1(3): 77–84.

Kohavi, R.; et al. 1996. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, 202–207.

Larionov, M. 2020. Sampling techniques in bayesian target encoding. *arXiv preprint arXiv:2006.01317*.

Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R. P.; Tang, J.; and Liu, H. 2017. Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6): 1–45.

Liu, G.; Yang, J.; and Wu, L. 2022. Ptab: Using the pre-trained language model for modeling tabular data. *arXiv preprint arXiv:2209.08060*.

Madeo, R. C.; Lima, C. A.; and Peres, S. M. 2013. Gesture unit segmentation using support vector machines: segmenting gestures from rest positions. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 46–52.

Margeloiu, A.; Simidjievski, N.; Lio, P.; and Jamnik, M. 2023. Weight predictor network with feature selection for small sample tabular biomedical data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 9081–9089.

McElfresh, D.; Khandagale, S.; Valverde, J.; Prasad, C. V.; Ramakrishnan, G.; Goldblum, M.; and White, C. 2023. When do neural nets outperform boosted trees on tabular data? *Advances in Neural Information Processing Systems*, 36: 76336–76369.

Micci-Barreca, D. 2001. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *ACM SIGKDD explorations newsletter*, 3(1): 27–32.

Niculescu-Mizil, A.; Perlich, C.; Swirszcz, G.; Sindhwani, V.; Liu, Y.; Melville, P.; Wang, D.; Xiao, J.; Hu, J.; Singh, M.; et al. 2009. Winning the KDD cup orange challenge with ensemble selection. In *KDD-Cup 2009 competition*, 23–34. PMLR.

Pace, R. K.; and Barry, R. 1997. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3): 291–297.

Pargent, F.; Pfisterer, F.; Thomas, J.; and Bischl, B. 2022. Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. *Computational Statistics*, 37(5): 2671–2692.

Popov, S.; Morozov, S.; and Babenko, A. 2019. Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312*.

Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A. V.; and Gulin, A. 2018. CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

Shwartz-Ziv, R.; and Armon, A. 2022. Tabular data: Deep learning is not all you need. *Information Fusion*, 81: 84–90.

Somvanshi, S.; Das, S.; Javed, S. A.; Antariksa, G.; and Hos-sain, A. 2024. A survey on deep tabular learning. *arXiv preprint arXiv:2410.12034*.

Song, W.; Shi, C.; Xiao, Z.; Duan, Z.; Xu, Y.; Zhang, M.; and Tang, J. 2019. AutoInt: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 1161–1170.

Stroup, W. W.; Ptukhina, M.; and Garai, J. 2024. *Generalized linear mixed models: modern concepts, methods and applications*. Chapman and Hall/CRC.

Wang, R.; Shivanna, R.; Cheng, D.; Jain, S.; Lin, D.; Hong, L.; and Chi, E. 2021. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the web conference 2021*, 1785–1797.

Wang, Y.; Wu, H.; Dong, J.; Qin, G.; Zhang, H.; Liu, Y.; Qiu, Y.; Wang, J.; and Long, M. 2024. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37: 469–498.

Ye, H.-J.; Liu, S.-Y.; Cai, H.-R.; Zhou, Q.-L.; and Zhan, D.-C. 2024. A closer look at deep learning methods on tabular datasets. *arXiv preprint arXiv:2407.00956*.

Zeng, G. 2014. A necessary condition for a good binning algorithm in credit scoring. *Applied Mathematical Sciences*, 8(65): 3229–3242.



Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*.

Zhou, S.; Chen, D.; Pan, J.; Shi, J.; and Yang, J. 2024. Adapt or perish: Adaptive sparse transformer with attentive feature refinement for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2952–2963.