Appendix A

Dataset Description

The datasets differ in feature types: some are purely numerical, others entirely categorical, and some are mixed. The selected datasets include 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). Among these, AD, KD, HI, and GE are retrieved from the OpenML platform, while CA is accessed via the scikit-learn library.

California Housing (Scikit-learn) The California Housing dataset (Pace and Barry 1997) pertains to houses found in a given California district and some summary statistics on them based on 1990 census data. The final data contained 20,640 observations on 8 numerical variables. The dependent variable is median house value.

Adult (OpenML ID=1590) The Adult dataset (Kohavi et al. 1996) is a benchmark dataset commonly used for classification tasks in machine learning. It contains 48,842 instances with 14 features, which include 8 categorical and 6 integer types. The goal is to predict whether an individual's annual income exceeds 50,000, based on census information such as age, education level, occupation, and marital status.

KDD Internet Usage (OpenML ID=981) The KDD Internet Usage dataset (Kehoe and Pitkow 1996), originally compiled by the Georgia Tech Research Corporation as part of the GVU's WWW User Surveys, provides detailed information on users' internet usage patterns and demographic characteristics. The dataset contains 10,120 instances and consists exclusively of 68 categorical features. The objective of this task is to predict which users are likely to pay for internet access at work.

HIGGS (OpenML ID=23512) The HIGGS Small dataset (Baldi, Sadowski, and Whiteson 2014) is a physics dataset generated via Monte Carlo simulations, containing approximately 98,000 samples. It includes 28 numerical features, divided into 21 low-level and 7 high-level features. The low-level features represent kinematic properties directly measured by particle detectors. In contrast, the high-level features are derived from the low-level ones using domain knowledge, designed by physicists to enhance discrimination between signal and background events. The dataset is used for classifying particle collision events.

Gesture (OpenML ID=4538) The dataset (Madeo, Lima, and Peres 2013) contains frame-based gesture phase segmentation data from 7 videos, each with approximately 1,400 to 2,700 frames. For each frame, two types of motion features are provided: 3D positions of the hands, wrists, head, and spine (from raw files, 18 features), and velocity/acceleration of the hands and wrists (from processed files, 32 features). Combined, each frame yields up to 50 numeric features with a class label, enabling per-frame gesture phase prediction.

Appendix B

Model Hyperparameter Configuration (Optuna)

For each model, we perform hyperparameter optimization using Optuna (Akiba et al. 2019). The search space follows the settings used in tabular deep learning benchmarks (Gorishniy et al. 2021).

XGBoost

Implementation: We use the official `xgboost` library (Chen and Guestrin 2016). The following parameters are fixed throughout all experiments:

- `booster = "gbtree"`
- `early_stopping_rounds = 50`
- `n_estimators = 2000`

The remaining hyperparameters are tuned according to the search space defined in Table 1.

Parameter	Distribution
Max depth	UniformInt[3, 10]
Min child weight	LogUniform[1e-8, 1e5]
Subsample	Uniform[0.5, 1]
Learning rate	LogUniform[1e-5, 1]
Col sample by level	Uniform[0.5, 1]
Col sample by tree	Uniform[0.5, 1]
Gamma	LogUniform[1e-8, 1e2]
Lambda	LogUniform[1e-8, 1e2]
Alpha	LogUniform[1e-8, 1e2]
# Iterations	100

Table 1: XGBoost hyperparameter search space.

Deep Learning Models

Implementation: We implemented all deep learning models using the official codebase from Gorishniy et al. (Gorishniy et al. 2021), which serves as the foundation for many recent benchmarks on tabular data. The models include:

- MLP (Table 2) (Gorishniy et al. 2021)
- ResNet (Table 3) (Gorishniy et al. 2021)
- DCN V2 (Table 4) (Wang et al. 2021)
- AutoInt (Table 5) (Song et al. 2019)
- FT-Transformer (Table 6) (Gorishniy et al. 2021)

Each corresponding table presents the hyperparameter search space used for tuning these models. The hyperparameter configuration for our proposed method, SG-XDEAT, is provided separately in Table 7.

Parameter	Distribution
# Layers	UniformInt[1, 8]
Layer size	UniformInt[1, 512]
Dropout	Uniform[0, 0.5]
Learning rate	LogUniform[1e-5, 1e-2]
Weight decay	LogUniform[1e-6, 1e-3]
Category embedding size	UniformInt[64, 512]
# Iterations	100

Table 2: MLP hyperparameter search space.

Parameter	Distribution
# Layers	UniformInt[1, 8]
Layer size	UniformInt[64, 512]
Hidden factor	Uniform[1, 4]
Hidden dropout	Uniform[0, 0.5]
Residual dropout	Uniform[0, 0.5]
Learning rate	LogUniform[1e-5, 1e-2]
Weight decay	LogUniform[1e-6, 1e-3]
Category embedding size	UniformInt[64, 512]
# Iterations	100

Table 3: ResNet hyperparameter search space.

Parameter	Distribution
# Cross layers	UniformInt[1, 8]
# Hidden layers	UniformInt[1, 8]
Layer size	UniformInt[64, 512]
Hidden dropout	Uniform[0, 0.5]
Cross dropout	Uniform[0, 0.5]
Learning rate	LogUniform[1e-5, 1e-2]
Weight decay	LogUniform[1e-6, 1e-3]
Category embedding size	UniformInt[64, 512]
# Iterations	100

Table 4: DCN V2 hyperparameter search space.

Parameter	Distribution
# Layers	UniformInt[1, 6]
Feature embedding size	UniformInt[8, 64]
Residual dropout	Uniform[0.0, 0.2]
Attention dropout	Uniform[0.0, 0.5]
Learning rate	LogUniform[1e-5, 1e-3]
Weight decay	LogUniform[1e-6, 1e-3]
# Iterations	100

Table 5: AutoInt hyperparameter search space.

Parameter	Distribution
# Layers	UniformInt[1, 4]
Feature embedding size	UniformInt[64, 512]
Residual dropout	Uniform[0, 0.2]
Attention dropout	Uniform[0, 0.5]
FFN dropout	Uniform[0, 0.5]
FFN factor	Uniform[2/3, 8/3]
Learning rate	LogUniform[1e-5, 1e-3]
Weight decay	LogUniform[1e-6, 1e-3]
# Iterations	100

Table 6: FT-Transformer hyperparameter search space.

Parameter	Distribution
# Layers	UniformInt[1, 4]
Feature embedding size	UniformInt[64, 512]
Residual dropout	Uniform[0, 0.2]
Attention dropout	Uniform[0, 0.5]
FFN dropout	Uniform[0, 0.5]
FFN factor	Uniform[2/3, 8/3]
Learning rate	LogUniform[1e-5, 1e-3]
Weight decay	LogUniform[1e-6, 1e-3]
Min samples leaf	UniformInt[1, 128]
Min impurity decrease	LogUniform[1e-9, 0.01]
# Iterations	100

Table 7: SG-XDEAT hyperparameter search space.

Appendix C

Results for all algorithms on all datasets

Table 8 summarizes the performance of various benchmark models across five datasets. To assess the statistical significance of performance differences, we apply the one-sided Wilcoxon test (Wilcoxon 1945) with a significance level of $\alpha = 0.05$ with Bonferroni correction. This provides a rigorous measure of whether improvements are consistent and not due to random variation.

Results for Architectural Ablation Analysis

We present ablation results in Table 9 to evaluate the contribution of each architectural component in SG-XDEAT. Specifically, we compare the design (CD + CE) against two partial variants—CD-only and CE-only—as well as a baseline that directly concatenates feature embeddings (DFC) without modeling any structured attention. The full model consistently achieves comparable results, demonstrating the importance of modeling both types of dependencies.

Results for Different Input Strategies

Table 10 reports the performance of models using either raw features or target-aware encodings exclusively. Across most datasets, neither stream consistently outperforms the other,

Datasets	CA	GE	AD	KD	HI
Metrics	RMSE↓	Accuracy↑			
XGBoost	**0.451±0.009**	**0.685±0.009**	**0.871±0.003**	**0.902±0.005**	0.727±0.003
MLP	0.499±0.008	0.651±0.012	0.858±0.002	0.892±0.007	0.725±0.003
Resnet	0.489±0.007	0.657±0.008	0.852±0.003	0.894±0.006	**0.734±0.003**
DCN-V2	0.488±0.009	0.634±0.013	0.859±0.002	0.899±0.005	0.726±0.003
AutoInt	0.490±0.009	0.602±0.015	0.859±0.002	0.898±0.004	0.726±0.003
FT-Transformer	0.472±0.009	**0.677±0.009**	0.861±0.002	**0.903±0.005**	**0.732±0.002**
SG-XDEAT	**0.454±0.008**	0.675±0.012	**0.872±0.003**	**0.903±0.003**	**0.732±0.002**

Table 8: Comparison of performance across various benchmark models. Performance is reported as mean ± standard deviation. The best result and those not statistically different from it ($p \geq 0.0083$) are shown in **bold**.

Datasets	CA	GE	AD	KD	HI
Metrics	RMSE↓	Accuracy↑			
DFC	0.480±0.010	0.649±0.015	0.869±0.003	0.902±0.005	**0.732±0.003**
CD	0.463±0.010	**0.678±0.008**	0.871±0.003	**0.903±0.005**	**0.732±0.003**
CE	0.459±0.011	0.626±0.013	**0.872±0.003**	0.899±0.007	0.727±0.003
CD + CE	**0.454±0.008**	0.675±0.012	**0.872±0.003**	**0.903±0.003**	**0.732±0.002**

Table 9: Ablation results for architectural components. Values are reported as mean ± standard deviation. **CD** = Cross-Dimension, **CE** = Cross-Encoding, **DFC** = Direct Feature Concatenation. **Best** results are highlighted.

suggesting that each captures distinct yet complementary information. However, the DFC method, which directly combines the raw and target-aware streams, fails to capture the dependencies between them effectively. These results motivate the dual-stream design adopted in SG-XDEAT, which aims to integrate both perspectives more effectively.

Setup for Adaptive Sparse Self-Attention

This experiment aims to evaluate whether incorporating Adaptive Sparse Self-Attention (ASSA) into a pre-norm Transformer improves model robustness in the presence of noisy or redundant features. To this end, we construct a controlled synthetic regression benchmark where the degree of feature redundancy is systematically varied. Table 12 reports the test RMSE for pre-norm Transformers with and without ASSA across different values of ρ. The results demonstrate that models equipped with ASSA consistently achieve lower RMSEs confirming its effectiveness in suppressing noisy features and enhancing predictive robustness.

Synthetic Benchmark Construction We construct a synthetic regression benchmark to test whether incorporating Adaptive Sparse Self-Attention (ASSA) into a pre-norm Transformer improves performance compared to using standard softmax attention only. Each input is a 100-dimensional vector, with only a fraction $\rho \in 0.5, 0.6, \ldots, 1.0$ containing useful signal. For each ρ and random seed (10 in total), we generate 64,000 training, 16,000 validation, and 20,000 test samples. Targets are computed using a fixed, randomly initialized 4-layer MLP with ReLU activations between layers, which processes only the informative features. The same MLP is used across all splits for a given seed and ρ. Target values are standardized to zero mean and unit variance.

This setup produces 60 datasets (10 seeds × 6 ρ values), allowing for fine-grained evaluation of model behavior under increasing feature sparsity. It provides a clear testbed for assessing whether ASSA-equipped Transformers are more robust to noisy or redundant features compared to their softmax-based counterparts.

Implementation Details In this setup, we do not apply any target-aware encoding. Instead, raw input features are directly tokenized and fed into a standard Transformer backbone. The full configuration is summarized in Table 11. Training is conducted for up to 200 epochs with a batch size of 512. Early stopping is employed based on validation performance, using a patience of 10 epochs.

Parameter	Setup
# Layers	3
# Heads	8
Feature embedding size	192
Feature hidden size	256
Residual dropout	0.0
FFN dropout	0.1
Attention dropout	0.2
(Optimizer, LR)	(AdamW, 1e-3)

Table 11: Transformer Configuration for ASSA Experiments

Datasets	CA	GE	AD	KD	HI
Metrics	RMSE↓	Accuracy↑			
Raw	0.483±0.008	0.665±0.013	0.859±0.002	**0.903±0.005**	**0.732±0.003**
Targeted	0.479±0.011	0.655±0.015	0.863±0.005	0.888±0.007	**0.732±0.002**
DFC	0.480±0.010	0.649±0.015	0.869±0.003	0.902±0.005	**0.732±0.003**
CD + CE	**0.454±0.008**	**0.675±0.012**	**0.872±0.003**	**0.903±0.003**	**0.732±0.002**

Table 10: Performance comparison between different input strategies. The *Raw* setting uses original features only, while the *Targeted* variant incorporates label-dependent encodings. The DFC and CD+CE methods are described in Figure 2 (Main Text). Values are reported as mean ± standard deviation. The overall best results are shown in **bold**, while underlined values indicate the better performance between Raw and Targeted for each dataset.

Setting	$\rho = 0.5$	$\rho = 0.6$	$\rho = 0.7$	$\rho = 0.8$	$\rho = 0.9$	$\rho = 1.0$
W/ ASSA	**0.498±0.032**	**0.517±0.032**	**0.529±0.055**	**0.565±0.033**	**0.593±0.031**	**0.616±0.055**
W/O ASSA	0.523±0.024	0.556±0.034	0.584±0.060	0.617±0.030	0.646±0.028	0.651±0.036

Table 12: Regression performance (RMSE) on synthetic datasets with varying proportions of informative features ρ. A lower ρ implies a higher proportion of irrelevant (noisy) features. Values are reported as mean ± standard deviation.

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.

Baldi, P.; Sadowski, P.; and Whiteson, D. 2014. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5(1): 4308.

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.

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.

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.

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.

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

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.

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.

Wilcoxon, F. 1945. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6): 80–83.