DynaTab: Dynamic Feature Ordering as Neural Rewiring for High-Dimensional Tabular Data

Al Zadid Sultan Bin Habib[1], **Gianfranco Doretto**[2], **Donald A. Adjeroh**[3]

[1, 3]Lane Dept. of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506, USA
[2]Sci. Computing and Imaging Inst. & Dept. of Biomedical Informatics, The Univ. of Utah, Salt Lake City, UT 84112, USA
[1]ah00069@mix.wvu.edu, [2]doretto@utah.edu, [3]donald.adjeroh@mail.wvu.edu

Abstract

High-dimensional tabular data lacks a natural feature order, limiting the applicability of permutation-sensitive deep learning models. We propose DynaTab, a dynamic feature ordering-enabled architecture inspired by neural rewiring. We introduce a lightweight criterion that predicts when feature permutation will benefit a dataset by quantifying its intrinsic complexity. DynaTab dynamically reorders features via a neural rewiring algorithm and processes them through a compact, dynamic order-aware combination of separate learned positional embedding, importance-based gating, and masked attention layers, compatible with any sequence-sensitive backbone. Trained end-to-end with bespoke dynamic feature ordering (DFO) and dispersion losses, DynaTab achieves statistically significant gains, particularly on high-dimensional datasets, where it is benchmarked against 45 state-of-the-art baselines across 36 different real-world tabular datasets. Our results position DynaTab as a compelling new paradigm for high-dimensional tabular deep learning.

Code — https://github.com/zadid6pretam/DynaTab

Introduction

Developing end-to-end deep learning models for high-dimensional tabular data remains challenging due to its lack of inherent structure, unlike image or text modalities. Traditional architectures such as convolutional neural networks (CNNs) and transformers perform well in vision and natural language processing (NLP) tasks, but often underperform on tabular tasks, especially with high-dimensional data (Ruiz et al. 2024). Tree ensembles are strong baselines; deep tabular models improve with imbalance-aware training (Tabasum et al. 2025). Models like TabNet (Arik and Pfister 2021), TabTransformer (Huang et al. 2020), and FT-Transformer (Gorishniy et al. 2021) attempt to address these issues, yet robust solutions are still limited. TabSeq (Habib et al. 2024) introduced feature ordering to impose structure, but its fixed order cannot generalize across datasets. Mambular (Thielmann et al. 2024) highlighted the sensitivity of model behavior to random input orderings. This issue is also seen in large language models (LLMs), where sequence affects predictions in reasoning and question answering (QA) tasks (Guan et al. 2025). In parallel, methods such as ProtoGate (Jiang et al. 2024) and Deep Neural Pursuit (DNP) (Liu et al. 2017) explored data-driven feature selection for High-Dimensional Low-Sample Size (HDLSS) contexts. These developments motivate the Column Permutation Problem (CPP) as a core challenge in tabular deep learning (Lima, Santos, and Carvalho 2024) for high-dimensional datasets, supporting dynamic feature ordering with order-aware fusion as a promising solution and foundation of DynaTab.

Neuroplasticity, the brain's capacity to reorganize and adapt its neural connections (Pham et al. 2023; Schmidgall et al. 2024; Nwadiugwu 2020) provides a powerful model for developing flexible approaches in deep learning, such as DynaTab's dynamic feature ordering. Through synaptic plasticity (Bliss and Collingridge 1993; Kandel 2001), the brain strengthens or weakens connections based on stimulus relevance, allowing neurons to rewire in response to learning, experience, or recovery (Song, Miller, and Abbott 2000; Zenke, Gerstner, and Ganguli 2017). Similarly, structural plasticity (Holtmaat and Svoboda 2009; Lamprecht and LeDoux 2004; Chklovskii, Mel, and Svoboda 2004), the formation and pruning of connections, enables the brain to optimize processing by reconfiguring networks for specific tasks. Mechanisms like Long-Term Potentiation (LTP) and Long-Term Depression (LTD) regulate synaptic strength (Malenka and Bear 2004), reinforcing frequently co-activated neural pathways to enhance memory and learning. Principles from Hebbian learning (Hebb 2005) and Spike-Timing-Dependent Plasticity (STDP) (Sjöström and Gerstner 2010) further refine this adaptability by adjusting connection strength based on the timing of neuronal spikes, optimizing synaptic efficiency.

Feature ordering has long history in pattern recognition and is central to Incremental Attribute Learning (IAL), where features arrive sequentially and must be ranked before training (Wang and Guan 2013). Unlike set-based treatments that assume order invariance (Zaheer et al. 2017), column order in practice shapes redundancy exposure and impacts our ability to capture dependencies. Empirical studies show that Fisher/correlation/entropy ranks reduce interference and error compared to unordered baselines (Wang et al. 2015c,b), motivating learned, task-aware ordering in machine learning (Wang et al. 2015a, 2014). Recent work highlights that tabular models can be brittle to column per-

 `pip install dynatab`.

mutations and thus enforce order-agnostic (permutation-invariant) representations to neutralize the impact of feature ordering (Eremeev et al. 2025; Brahmavar, Li, and Oliva 2025). In deep tabular learning, Mambular (Thielmann et al. 2024) highlighted the importance of feature ordering, and TabSeq (Habib et al. 2024) introduced an explicit ordering algorithm. ROTATOR-LLM (Wang et al. 2025) investigates feature ordering for LLM-based tabular inference. TabICL (Qu et al. 2025) makes predictions over multiple different column permutations and then combines these results to restore the desired invariance. In many real-world HDLSS tasks, even simple models, such as MLP and Lasso, can outperform advanced methods (see ProtoGate (Jiang et al. 2024)). This observation underscores that selection alone is not sufficient in $n \ll m$ regimes (here n=number of samples, m=number of features), and that the permutation itself should be a learnable object. We therefore pose the CPP as a core challenge for deep tabular learning for high-dimensional settings to learn a data-driven feature order that reduces redundancy, exposes long-range dependencies, and provides sequence structure to downstream modules. Practically, this can be approached with attention-based pointer mechanisms and graph-aware extensions that generate permutations while encoding relational structure (Vinyals, Fortunato, and Jaitly 2015; Yang et al. 2022; Veličković et al. 2020). Analogous reordering ideas appear in hyperspectral compression schemes that reorder bands to expose correlations (Tate 1997; Jain and Adjeroh 2007) and in graph/-communication layouts that minimize pairwise interaction costs (Wang, Xu, and Lisser 2014).

Inspired by these adaptive processes, we propose DynaTab, a novel deep learning model specifically designed to optimize feature ordering for high-dimensional complex tabular data. This model introduces a dynamic feature ordering approach inspired by neural rewiring principles, enabling adaptive feature arrangement that responds to data complexity. DynaTab emulates neural rewiring (Merzenich and Jenkins 1993; Pascual-Leone et al. 2005; Draganski et al. 2004) through dynamic feature ordering, adjusting feature connectivity and relevance in response to data complexity. Just as local plasticity (Govindarajan, Kelleher, and Tonegawa 2006; Yasuda 2017) allows clusters of neurons to adapt and improve task-specific performance, DynaTab reorders feature graphs based on node centrality to optimize local relationships, creating a context-aware, flexible approach to enhance learning on complex, high-dimensional tabular data. This biologically inspired model adaptation allows tabular deep learning models to mimic the brain's adaptability in recognizing patterns and making decisions, underscoring dynamic feature ordering as an effective tool for handling heterogeneous datasets. Our key contributions are as follows:

1. We predict when feature ordering offers major gains by quantifying dataset complexity and ordering sensitivity.
2. We propose dynamic feature ordering (DFO) inspired by neural rewiring, adaptively reordering columns to optimize inter-feature connectivity per dataset.
3. We introduce DynaTab, an end-to-end model that integrates an order-aware fusion of positional embeddings, importance gating, and dynamic masked attention with a plug-and-play sequential backbone (e.g., Transformer, DAE, LSTM, or Mamba), trained jointly under DFO and dispersion losses.
4. We introduce a taxonomy for tabular datasets based on their sample size and feature dimensionality, and thus empirically group tabular datasets into 5 categories. Then, we benchmark DynaTab against 45 baselines on 36 tabular datasets of different kinds, using Friedman tests with Critical Difference (CD) diagrams and Wilcoxon Holm post hoc analysis to demonstrate significant, consistent improvements in classification and regression.

These contributions together tackle varying feature relevance and dataset complexity, setting a new benchmark for adaptive feature ordering and end-to-end model optimization in tabular deep learning.

Related Work

Deep tabular models face challenges due to the lack of spatial or sequential structure. Early architectures such as TabNet (Arik and Pfister 2021), TabTransformer (Huang et al. 2020), and FT-Transformer (Gorishniy et al. 2021) employ attention-based mechanisms to model feature interactions. Tree-based models like NODE (Popov, Morozov, and Babenko 2020) and probabilistic transformers like TabPFN (Hollmann et al. 2023) achieve strong performance but face scalability issues ($\leq$1000 samples). Ensemble models like XGBoost, LightGBM, and CatBoost remain competitive baselines in many scenarios.

Ordering-based models directly motivate our approach. TabSeq (Habib et al. 2024) sequences features by relevance; Mambular (Thielmann et al. 2024) studies column permutations with state-space layers; ProtoGate (Jiang et al. 2024) uses prototype-guided selection in HDLSS settings. Pointer Networks (Vinyals, Fortunato, and Jaitly 2015) and extensions like GPN (Yang et al. 2022) explore differentiable output permutations. Models such as TANGOS (Jeffares et al. 2023), NDTF (Kontschieder et al. 2015), and TabPFN v2 (Hollmann et al. 2025) introduce selectors, hybrid modules, or generative pretraining. Sequential backbones are increasingly explored. MambaTab (Ahamed and Cheng 2024) and MambAttention (Thielmann and Samiee 2024), Mamba-based variants use state-space modeling; TabulaRNN (Thielmann and Samiee 2024), Trompt (Chen et al. 2023), and ModernNCA (Ye et al. 2025) apply recurrence, memory, or attention pooling to structured inputs. ResNet-style MLPs (Gorishniy et al. 2021), TabR (Gorishniy et al. 2024), and TabM (Gorishniy, Kotelnikov, and Babenko 2025) stack lightweight modules with residual connections for improved generalization. TabICL (Qu et al. 2025) uses RoPE and combines column permutations to restore invariance.

Instance-wise feature selection complements our ordering approach. Models like INVASE (Yoon, Jordon, and Van der Schaar 2018), STG (Yamada et al. 2020), and LSPIN/LLSPIN (Yang, Lindenbaum, and Kluger 2022) dynamically gate features per sample, supporting interpretabil-

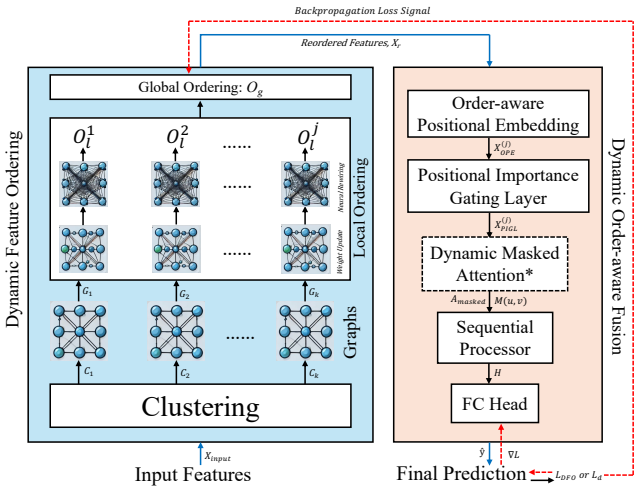


Figure 1: End-to-end DynaTab. Left (light blue): Dynamic Feature Ordering produces X_r (feature graphs drawn using GPT-4o (Achiam et al. 2023)). Right (light peach): Order-aware Fusion (OPE + PIGL), DMA$*$, and a sequential backbone. Solid arrows: data flow; dashed red arrows: gradient flow. $*$ DMA applies only to attention-based backbones.

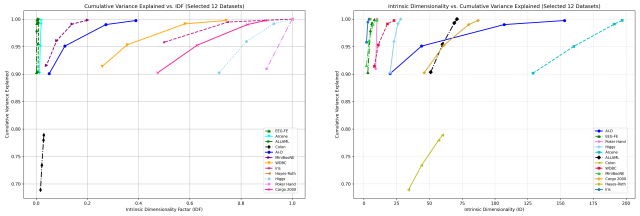
ity. Broader developments inform our context. PLATO (Ruiz et al. 2024) proposes a graph-based meta-learning framework for tabular transfer learning. TabDDPM (Kotelnikov et al. 2023) introduces diffusion models for high-fidelity tabular synthesis. TabReD (Rubachev et al. 2025) presents temporally-split benchmarks, highlighting gaps between gradient boosted decision trees (GBDTs) and deep models. T2G-Former (Yan et al. 2023) and HYTREL (Chen et al. 2024) use graph-augmented transformers to model feature relationships and support structure-aware learning. These works underscore the ongoing need for flexible, data-adaptive solutions. RKNN-FS (Li, Harner, and Adjeroh 2011) presented a feature selection-based model for HDLSS problems, building on a proposed random k-nearest neighbor (RKNN) algorithm.

Our DynaTab framework (Figure 1) introduces dynamic feature ordering inspired by neural rewiring. It integrates an order-aware fusion block (OPE, PIGL, DMA) with sequential backbones (Transformer, LSTM, DAE, or Mamba), trained end-to-end using DFO and Dispersion losses. This enables context-sensitive learning across classification, regression, and coherence tasks on complex tabular datasets.

Methodology

Feature Ordering - When to Use?

The Intrinsic Dimensionality Factor (IDF) estimates the benefit of feature ordering by comparing a dataset's intrinsic dimensionality i.e., the minimal number of features capturing core variability (Levina and Bickel 2004) to its total feature count. IDF is defined as the ratio of intrinsic to total feature count, $\text{IDF} = n_{\text{intrinsic}}/n_{\text{total}}$, where $n_{\text{intrinsic}}$ is estimated using methods such as PCA. The dataset's complexity is then



(a) Cumulative Variance vs. IDF (b) ID vs. Cumulative Variance

Figure 2: Relationship between cumulative variance and intrinsic dimensionality across 12 selected datasets. See supplementary material for details on all 36 datasets used.

measured as:

$$\text{Complexity Score} = \frac{\text{Cumulative Variance at IDF}}{\text{IDF}^s} \quad (1)$$

where s is a tunable sensitivity parameter. The Feature Ordering Effectiveness (FOE) quantifies the gain from ordering:

$$\text{FOE} = \frac{\psi}{(\text{AUC} \times \text{IDF})^s} \quad (2)$$

Here, ψ is a dataset-specific scaling factor, and AUC is the area under the IDF-variance curve. The loss function below optimizes ψ by minimizing the gap between the computed FOE and the target value (set to 1):

$$\text{Loss}(\psi) = \left(\frac{\psi}{(\text{AUC})^s} - 1 \right)^2 \quad (3)$$

Setting $s = 2$ introduces quadratic sensitivity (Hinton and Salakhutdinov 2006), increasing the penalty for low-variance datasets. AUC is estimated using the trapezoidal rule (Hanley and McNeil 1982) for efficient integration of discrete IDF–variance pairs. While ψ and AUC vary across datasets, FOE maintains an inverse relationship with IDF (Equation 2), making it a reliable proxy for ordering effectiveness.

$$\text{FOE} \propto \frac{1}{\text{IDF}} \quad \text{(for fixed } \psi \text{ and AUC)} \quad (4)$$

Although ψ and AUC vary, FOE consistently decreases with increasing IDF (Equation 2). A lower IDF indicates higher redundancy, making feature ordering more effective. In contrast, higher IDF reflects better-structured datasets with limited gain from ordering. Thus, the success probability P_{success} is inversely related to IDF (Eq. 5) which yields in Eq. 6.

$$\text{FOE} \propto \frac{1}{\text{IDF}} \propto P_{\text{success}} \quad (5)$$

$$P_{\text{success}} = 1 - \text{IDF} = 1 - \frac{n_{\text{intrinsic}}}{n_{\text{total}}} \quad (6)$$

As intrinsic dimensionality approaches the total, P_{success} drops; lower intrinsic dimensionality suggests greater ordering benefit (Equations 5, 6). Figure 2 (see also Supplementary Material) shows the cumulative variance–IDF relationship. Figure 2a illustrates that datasets with low IDF reach high cumulative variance quickly, indicating redundancy. Figure 2b presents absolute intrinsic dimensionality across datasets. See the supplementary material for FOE-based dataset rankings and further analysis.

Dynamic Feature Ordering as Neural Rewiring

Feature ordering aims to find an optimal feature arrangement within and across clusters that minimizes disorganization (Habib et al. 2024). For a dataset $X \in \mathbb{R}^{n\times m}$, we define graphs $\{G_1, \ldots, G_k\}$ for k clusters, where each $G_j = (V_j, E_j)$ represents cluster j, with vertices $u \in V_j$ as features and edges $(u, v) \in E_j$ weighted by relationship strength $w(u, v)$. The optimal local ordering $\mathcal{O}_{\text{l}}^j$ minimizes intra-cluster dispersion:

$$F_j(\mathcal{O}_{\text{l}}^j) = \arg\min_{\mathcal{O}_{\text{l}}^j} \sum_{(u,v)\in E_j} w(u, v) \cdot |\mathcal{O}_{\text{l}}^j(u) - \mathcal{O}_{\text{l}}^j(v)| \quad (7)$$

where $\mathcal{O}_{\text{l}}^j(u)$ is the position of feature u in $\mathcal{O}_{\text{l}}^j$. The global ordering $\mathcal{O}_{\text{g}}$ aggregates local orderings to minimize a weighted global dispersion cost (Eq. 8) and the optimal global ordering is then obtained (Eq. 9).

$$G(\mathcal{O}_{\text{g}}) = \sum_{j=1}^{k} \alpha_j \cdot F_j(\mathcal{O}_{\text{l}}^j), \quad \sum_{j=1}^{k} \alpha_j = 1 \quad (8)$$

$$\mathcal{O}_{\text{g}}^* = \arg\min_{\mathcal{O}_{\text{g}}} G(\mathcal{O}_{\text{g}}) \quad (9)$$

Local Ordering with Neural Rewiring: Local ordering minimizes intra-cluster dispersion by optimizing $F_j(\mathcal{O}_{\text{l}}^j)$ (Equation 7). To guide this process, we compute centrality scores e.g., degree, betweenness, or eigenvector centrality (Freeman 1978; Bonacich 1972) which quantify a feature's influence in the graph. High-centrality features are prioritized by strengthening their connections and pruning weaker ones, simulating structural plasticity (Holtmaat and Svoboda 2009; Lamprecht and LeDoux 2004). Edge weights are then updated based on feature centrality, producing an optimized feature graph and revised ordering $\mathcal{O}_{\text{l}}^j$, in line with synaptic plasticity (Bliss and Collingridge 1993; Kandel 2001). Inspired by Hebbian learning (Hebb 2005), we model edge weight updates as:

$$w_{uv}^{(t+1)} = w_{uv}^{(t)} + \Delta w_{uv} \quad (10)$$

where Δw_{uv} reflects centrality-driven adjustment:

$$\Delta w_{uv} = \lambda C(u)C(v) - \epsilon w_{uv}^{(t)} \quad (11)$$

Here, λ is the learning rate, ϵ a regularization term, and $C(u), C(v)$ the centrality scores of features u, v respectively. This promotes connections between influential features while suppressing less relevant ones (Bullmore and Sporns 2009; Bassett and Bullmore 2006), enhancing information flow and structural coherence. As in neuroplasticity (Kandel et al. 2000), this rewiring reinforces key connections to minimize dispersion (Equation 7) and align clusters into a cohesive global order (Equation 9).

Local Ordering and Quality Maximization: The rewiring process aims to maximize a local quality metric Q, halting when further updates yield minimal improvement. At each step t, edge weights are updated as:

$$w_{uv}^{(t+1)} = w_{uv}^{(t)} + \Delta w_{uv} \quad \text{such that} \quad Q(\mathcal{O}_{\text{l}}^{(t+1)}) \geq Q(\mathcal{O}_{\text{l}}^{(t)}) \quad (12)$$

Iterations stop when the quality change drops below a threshold ϵ, indicating convergence:

$$\left|\frac{\partial Q}{\partial \mathcal{O}_{\text{l}}^{(t)}}\right| = |Q(\mathcal{O}_{\text{l}}^{(t+1)}) - Q(\mathcal{O}_{\text{l}}^{(t)})| < \epsilon \quad (13)$$

The threshold ϵ controls early stopping and defines when additional rewiring becomes non-beneficial.

Edge Pruning and Rewiring: Edges with weights below a threshold θ are pruned:

$$w_{uv}^{(t+1)} = 0 \quad \text{if} \quad w_{uv}^{(t)} < \theta \quad (14)$$

New connections are then formed between high-centrality nodes u', v' as:

$$w_{u'v'}^{(t+1)} = \lambda C(u')C(v') - \epsilon w_{u'v'}^{(t)} \quad (15)$$

This rewiring promotes high-centrality features within each local ordering $\mathcal{O}_{\text{l}}^j$, guiding the final global ordering $\mathcal{O}_{\text{g}}$.

Quality Metric Q: We define a general metric Q to evaluate how well a feature ordering aligns with pairwise relationships captured by an edge metric $\zeta(u, v)$, which may represent KL divergence, Euclidean or Manhattan distance, variance difference, or correlation-based dissimilarity (see ablation studies in supplementary). In the feature graph G_j for cluster C_j, each edge $(u, v) \in E_j$ is weighted by $\zeta(u, v)$. The quality of a local ordering $\mathcal{O}_{\text{l}}^j$ is:

$$Q(\mathcal{O}_{\text{l}}^j) = \sum_{(u,v)\in E_j} \zeta(u, v) \cdot d_{\mathcal{O}_{\text{l}}^j}(u, v) \quad (16)$$

$$d_{\mathcal{O}_{\text{l}}^j}(u, v) = \left|\mathcal{O}_{\text{l}}^j(u) - \mathcal{O}_{\text{l}}^j(v)\right| \quad (17)$$

This encourages similar features (low $\zeta(u, v)$) to appear closer in the ordering. The system iteratively rewires the graph and updates $\mathcal{O}_{\text{l}}^j$ to reduce $Q(\mathcal{O}_{\text{l}}^j)$, accepting changes that lower total dissimilarity. Algorithm 1 outlines the neural rewiring procedure for local ordering.

Global Ordering: The global ordering $\mathcal{O}_{\text{g}}$ integrates local cluster orderings to minimize total feature dispersion and improve model performance (Equation 9). It is defined as:

$$\mathcal{O}_{\text{g}} = \arg\min_{\mathcal{O}} \sum_{j=1}^{k} \alpha_j \cdot F_j(\mathcal{O}_{\text{l}}^j)$$

where $F_j(\mathcal{O}_{\text{l}}^j)$ measures local feature dispersion (Kostal, Lansky, and Pokora 2013; Th. Gries 2021), and α_j is the weight for cluster j, derived from inter-centroid distances. This formulation encourages cohesion across clusters by emphasizing more densely connected regions of the feature space.

Local dispersion for cluster j is computed as:

$$D(\mathcal{O}_{\text{l}}^j) = \sum_{(u,v)\in E_j} w(u, v) \cdot |\mathcal{O}_{\text{l}}^j(u) - \mathcal{O}_{\text{l}}^j(v)| \quad (18)$$

where $w(u, v)$ captures the proximity importance between features u and v, reinforcing related features being placed closer together (see also Equation 7). An alternate view defines dispersion of feature j under ordering $\mathcal{O}_{\text{l}}$ as $D(\mathcal{O}_{\text{l}}^j) = \sum_{k=1}^{d} W_{jk}$ where W_{jk} is the entry of the adjacency (distance) matrix. Reducing this dispersion directly reduces feature variance as $\text{Var}_j(X_u) \propto D(\mathcal{O}_{\text{l}}^j)$. Algorithm 2 summarizes the complete dynamic feature ordering process.

Algorithm 1: Neural Rewiring for Local Ordering

Input: Feature graph $G_j = (V_j, E_j)$, quality metric Q, mutation probability p, tolerance ϵ, pruning threshold θ
Output: Rewired local feature ordering $\mathcal{O}_{\text{l}}^j$
Initialize $\mathcal{O}_{\text{l}}^j = \arg\min_{\mathcal{O}} \sum_{(u,v)\in E_j} w(u,v) \cdot |\mathcal{O}_{\text{l}}^j(u) - \mathcal{O}_{\text{l}}^j(v)|$
Calculate centrality $C(v)$ for each node $v \in V_j$
while stopping criteria not met **do**
    Compute current quality $Q_{\text{c}} = Q(\mathcal{O}_{\text{l}}^j)$
    for each edge $(u, v) \in E_j$ **do**
        if $w_{uv}^{(t)} < \theta$ **then**
            Prune edge (u, v) and rewire based on centrality of u' and v': $w_{u'v'}^{(t+1)} = \lambda C(u')C(v') - \epsilon \cdot w_{u'v'}^{(t)}$
        end if
    end for
    Update $\mathcal{O}_{\text{l}}^{j,(t+1)}$ based on rewired graph
    if $|\Delta Q| < \epsilon$ **then**
        Stop rewiring
    end if
end while
Return $\mathcal{O}_{\text{l}}^j$

Dynamic Order-aware Fusion

The DynaTab backbone includes a specialized Dynamic Order-aware Fusion (DOF) module that processes globally reordered features through an order-sensitive pipeline. To mitigate permutation invariance and capture ordering effects, DOF incorporates three components before the sequential processor: Order-aware Positional Embedding (OPE), Positional-Importance Gating Layer (PIGL), and Dynamic Masked Attention (DMA), inspired by attention mechanisms, biological gating, and sequence modeling. DOF maintains the inductive bias from DFO and supports robust representation learning via a flexible sequential processor, instantiated as a Transformer, Denoising Autoencoder (DAE), LSTM, hybrid DAE-MHA-LSTM, or Mamba.

Order-aware Positional Embedding (OPE): Inspired by positional encoding in Transformers (Vaswani et al. 2017), OPE injects dynamic positional information into globally reordered tabular features. Given a reordered sequence $X_r \in \mathbb{R}^m$ from the global ordering $\mathcal{O}_{\text{g}}$, each feature $X_r^{(f)}$ is augmented with a learned embedding based on its position:

$$X_{\text{OPE}}^{(f)} = X_r^{(f)} + PE(\mathcal{O}_{\text{g}}(f)), \quad PE(\mathcal{O}_{\text{g}}(f)) \in \mathbb{R}^d \quad (19)$$

Here, $PE(\mathcal{O}_{\text{g}}(f))$ denotes the positional embedding for feature f, and d is the embedding dimension. Unlike fixed sinusoidal encodings, OPE learns embeddings that adapt to feature permutations, preserving semantic structure and enhancing downstream performance.

Positional Importance Gating Layer (PIGL): Inspired by biological gating and attention-free networks (Veness et al. 2021), PIGL modulates each feature based on an ordering-derived importance score γ_f, computed from metrics like dispersion or ordering weights. A sigmoid-activated gate controls the contribution of each feature:

$$X_{\text{PIGL}}^{(f)} = \sigma(W_g \gamma_f + b_g) \cdot X_{\text{OPE}}^{(f)} \quad (20)$$

where σ is the sigmoid function, and W_g, b_g are learnable gating parameters. This operation is permutation-invariant yet guided by γ_f, reinforcing salient features and suppressing less informative ones.

Dynamic Masked Attention (DMA): Inspired by directional masking in Transformers (Shaw, Uszkoreit, and Vaswani 2018), DMA enforces attention flow consistent with the global feature order. In attention-based architectures (e.g., Transformer, DAE-MHA-LSTM), a directional mask $M \in \mathbb{R}^{m\times m}$ is constructed using the global ordering $\mathcal{O}_{\text{g}}$. For features f_p and f_q, the mask is defined as:

$$M_{\text{SW}}(f_p, f_q) = \begin{cases} 0, & \text{if } \mathcal{O}_g(f_p) \le \mathcal{O}_g(f_q) \\ & \text{and } |p-q| \le w, \\ -\infty, & \text{otherwise.} \end{cases} \quad (21)$$

This mask is added to attention logits to block reverse attention:

$$A_{\text{DMA}} = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M_{\text{SW}}\right) V \quad (22)$$

where Q, K, V are query, key, and value matrices, and d_k is the key dimension. We use a full causal mask ($w = m$), so each feature can attend to all earlier ones. DMA enforces order-aware directionality in feature interactions.

Sequential Processor and FC Head: The processor operates on reordered inputs $X \in \mathbb{R}^{n\times m\times d_{\text{model}}}$, using any backbone (Transformer (Vaswani et al. 2017), DAE (Vincent et al. 2008), LSTM (Hochreiter and Schmidhuber 1997), DAE-MHA-LSTM, or Mamba (Gu and Dao 2024)) to produce a pooled output $\bar{h}$, followed by a lightweight fully connected head for prediction. Training uses class-weighted DFO or dispersion loss, enabling effective modeling even when $m \gg n$.

DFO Loss Function: The DFO loss combines a prediction loss with penalties for local feature smoothness and

Algorithm 2: Dynamic Feature Ordering

Input: Dataset X, clustering algorithm, number of clusters k, edge metric, sorting order, tolerance ϵ, mutation probability p
Output: Global feature ordering $\mathcal{O}_{\text{g}}$
Cluster data (e.g., KMeans) with clusters C_j and centroids μ_j: $c_i = \arg\min_j \|x_i - \mu_j\|^2$, $\mu_j = \frac{1}{|C_j|}\sum_{x_i\in C_j} x_i$
Build feature graph $G_j = (V_j, E_j)$ for each cluster C_j
for each cluster C_j **do**
    Apply Neural Rewiring to get local ordering $\mathcal{O}_{\text{l}}^j$
    Mutate ordering with probability p if $|\Delta Q| < \epsilon$
end for
Compute cluster importance α_j such that $\sum_{j=1}^k \alpha_j = 1$
Integrate local orderings into global ordering: $\mathcal{O}_{\text{g}} = \arg\min_{\mathcal{O}_{\text{g}}} \sum_{j=1}^k \alpha_j \cdot F_j(\mathcal{O}_{\text{l}}^j)$
Return $\mathcal{O}_{\text{g}}$

global ordering coherence. The default prediction term is binary cross-entropy (BCE) for binary classification:

$$L_{\text{b}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(y_i^{\text{pred}}) + (1-y_i)\log(1-y_i^{\text{pred}})\right] \tag{23}$$

For multi-class classification, BCE is replaced by categorical cross-entropy (CCE) (Goodfellow 2016), and for regression tasks, mean squared error (MSE) is used. The feature dispersion penalty encourages smooth transitions across reordered features:

$$P_d(X_r) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M-1}\left|X_r^{(i,j)} - X_r^{(i,j+1)}\right| \tag{24}$$

$$P_{\text{g}}(\mathcal{O}_{\text{g}}) = \frac{1}{|V|}\sum_{(u,v)\in V}\alpha_{u,v}\cdot d(\mathcal{O}_{\text{g}}(u), \mathcal{O}_{\text{g}}(v)) \tag{25}$$

$$\alpha_{u,v} = \frac{1}{\|c_u - c_v\| + \epsilon} \tag{26}$$

Global ordering coherence is encouraged via centroid-proximity weighting (Eq. 25). The full DFO loss balances these components:

$$L_{\text{DFO}} = \lambda_{\text{d}} P_d(X_r) + \lambda_{\text{g}} P_{\text{g}}(\mathcal{O}_{\text{g}}) + (1-\lambda_{\text{d}}-\lambda_{\text{g}})L_{\text{b}} \tag{27}$$

Dispersion Loss Function: A simplified version excludes the global penalty:

$$L_{\text{d}} = \lambda_{\text{reg}} P_d(X_r) + (1-\lambda_{\text{reg}})L_{\text{b}} \tag{28}$$

Customized loss functions enable end-to-end training that jointly promotes ordering quality, coherence, and prediction accuracy. This reinforces dynamic feature ordering as a biologically inspired mechanism akin to neural rewiring.

Experimental Results

When Feature Ordering Matters: While Deep Sets (Zaheer et al. 2017) handles unordered data (e.g., point clouds, chemoinformatics), we focus on high-dimensional tabular data where column order impacts learning. Optimal permutations reduce redundancy, reveal feature dependencies, and improve performance. This is especially relevant for high-dimensional biological profiles (e.g., gene expression), electroencephalograms (EEGs) and sensor streams, remote sensing, climate data, and multimodal or engineered feature tables, typical domains marked by sparsity, redundancy, and latent structure.

Categorization Rules: Let n be the number of samples, m the number of features, and $\rho = \frac{m}{n}$ the feature-to-sample ratio. Each dataset is assigned to one of five categories based on empirical ρ ranges:

HDLSS: $m > 1000,\ n < 1000,\ \rho > 2.$
HDHSS: $m > 1000,\ n > 10^4,\ 0.005 < \rho \leq 2.$
LDHSS: $m \leq 100,\ n > 10^4,\ \rho \leq 0.01.$
LDLSS: $m \leq 100,\ n \leq 1000,\ \rho \leq 0.05.$
MixedRegime: otherwise.

Datasets: We evaluate DynaTab on 36 datasets across five structural regimes defined by sample size and dimensionality. The HDLSS (high dimensionality, low sample size) group includes 8 biological datasets (e.g., Arcene, Colon, GLI-85, SMK_CAN_187, etc.) downloaded from (Li et al. 2018). The HDHSS (high dimensionality, high sample size) group includes 6 image datasets (e.g., HAM10000, MNIST, DeepLesion, etc.). We use ResNet-50 embeddings for the images. The LDLSS (low dimensionality, low sample size) group includes 6 low-dimensional datasets (e.g., Iris, Pima Indian, Glass), while the LDHSS (low dimensionality, high sample size) includes 5 large-sample datasets (e.g., Higgs, Adult Census, MiniBooNE). The MixedRegime category spans 8 varied datasets from clinical, geospatial, and chemical domains e.g. ADNI, MOF, EEG-FE (Bird et al. 2019), CNAE-9, AI-D (Ohlsson et al. 2020), EEG-PD (Park 2021), Water. We also evaluate on 3 regression datasets: DrivFace (HDLSS), Cargo, and GHG (MixedRegime). Full dataset sources and details are provided in the supplementary.

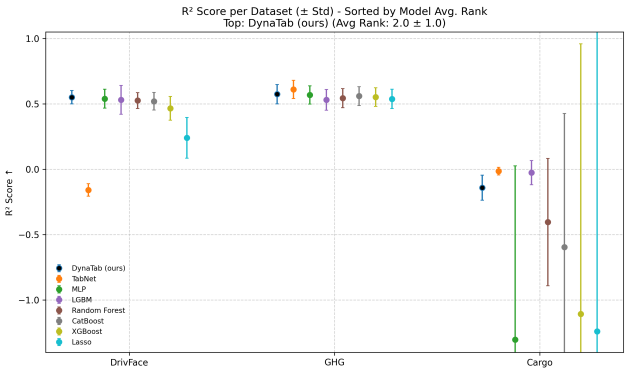


Figure 3: R^2 score ($\pm$ std) across three regression datasets, with models ranked by their avg. performance. Our proposed DynaTab achieves the best avg. rank across datasets.

Baseline Models: We benchmark DynaTab against a diverse set of models: classical (Naive Bayes, KNN, SVM, Decision Tree, Lasso, RF), ensembles (AdaBoost, GBM, XGBoost, LightGBM, CatBoost), deep models (MLP, CNN, DeepFM, DCN), probabilistic/tabular models (TabPFN, TabPFN v2, NODE, ENODE, NDTF, TANGOS, TabSeq, ProtoGate, TabNet, TabTransformer, FT-Transformer, SAINT, AutoInt), and others (ResNet Tabular, ModernNCA, TabR, Mamba-variants, TabM, Trompt, TabulaRNN, LSPIN, INVASE, STG).

Evaluation Metrics: We report 5-fold cross-validation accuracy (mean $\pm$ std) for all models, using 5×5 nested cross-validation for HDLSS and LDLSS datasets (matching the ProtoGate's protocol (Jiang et al. 2024) for HDLSS), and standard 5-fold cross-validation for HDHSS, LDHSS, and MixedRegime datasets. To compare methods across datasets, we compute average rank (Friedman 1937) over the benchmarks and select the top 10 for further analysis for each dataset categories. We report Critical Difference (CD) diagrams (Demšar 2006a) and perform Friedman-Nemenyi (Nemenyi 1963) and Wilcoxon-Holm tests (Demšar 2006b) (significance level $\alpha = 0.05$) to as-

HDLSS						
Dataset	1. DynaTab (Ours)	2. Lasso	3. ProtoGate†	4. MLP	5. TabulaRNN‡	6. LGBM§
GLI-85	85.96 ± 5.77	85.88 ± 4.71	82.48 ± 5.68	85.41 ± 8.00	79.68 ± 6.68	85.88 ± 11.53
SMK_CAN_187	61.31 ± 3.37	61.19 ± 13.72	60.16 ± 5.10	59.05 ± 7.44	60.02 ± 3.18	58.85 ± 10.14
ALLAML	92.31 ± 5.77	87.24 ± 3.39	86.12 ± 3.34	89.98 ± 9.17	88.92 ± 2.02	85.81 ± 5.67
Prostate-GE	90.91 ± 8.91	91.18 ± 6.39	90.58 ± 5.72	89.20 ± 6.07	90.50 ± 6.00	91.38 ± 5.71
Arcene	83.00 ± 6.71	81.00 ± 3.39	81.50 ± 5.10	78.40 ± 4.05	81.50 ± 5.10	80.50 ± 5.79
TOX-171	88.71 ± 3.53	91.86 ± 6.03	92.34 ± 5.67	94.48 ± 4.28	85.80 ± 4.70	81.98 ± 6.25
Colon	85.71 ± 8.91	79.40 ± 10.18	83.95 ± 9.82	83.95 ± 9.80	84.20 ± 6.50	76.60 ± 11.67
Lung	92.75 ± 1.28	94.47 ± 4.39	93.44 ± 6.37	96.47 ± 2.69	90.50 ± 4.80	93.42 ± 5.91
Avg. Rank ($\mu \pm \sigma$)	2.63 ± 2.26	5.13 ± 4.29	5.50 ± 3.74	5.88 ± 4.22	8.00 ± 5.86	9.00 ± 6.19
HDHSS						
Dataset	1. DynaTab (Ours)	2. LSPIN$^{\|}$	3. MLP	4. LGBM§	5. TabNet*	6. LLSPIN$^{\#}$
HAM10000^{+}	84.48 ± 0.38	81.37 ± 1.13	80.05 ± 0.95	80.33 ± 1.33	73.51 ± 3.62	79.60 ± 1.19
DeepLesion$^{+\dagger}$	94.48 ± 0.25	93.14 ± 0.29	92.93 ± 4.06	94.73 ± 2.84	93.02 ± 0.20	92.95 ± 0.63
MNIST$^{+\dagger}$	96.68 ± 0.16	96.77 ± 0.40	96.20 ± 0.76	95.42 ± 0.43	96.88 ± 0.38	96.36 ± 0.34
Fashion MNIST$^{+\dagger}$	88.52 ± 0.21	87.46 ± 0.53	87.85 ± 0.65	87.25 ± 0.68	87.99 ± 0.68	87.89 ± 0.29
CIFAR-10$^{+\dagger}$	88.58 ± 0.45	87.17 ± 1.02	88.10 ± 0.62	87.20 ± 0.75	87.25 ± 0.63	86.59 ± 0.63
Dog vs Cat$^{+\dagger}$	99.20 ± 0.15	99.25 ± 0.13	99.22 ± 0.15	99.12 ± 0.15	99.09 ± 0.13	99.19 ± 0.29
Avg. Rank ($\mu \pm \sigma$)	2.50 ± 2.35	4.67 ± 2.80	5.67 ± 3.01	6.33 ± 3.67	6.83 ± 4.62	7.33 ± 3.01

Table 1: Mean accuracy ($\mu \pm \sigma$) on HDLSS (top) and HDHSS (bottom) datasets and average rank ($\mu \pm \sigma$) of the top-6 models in each regime. Rankings are based on cross-dataset performance. "+" indicates ResNet50-extracted features; "†" denotes 11K subsample. Model symbols: ProtoGate† (Jiang et al. 2024), TabulaRNN‡, (Thielmann and Samiee 2024), LGBM§ (Ke et al. 2017), LSPIN$^{\|}$ (Yang, Lindenbaum, and Kluger 2022), LLSPIN$^{\#}$ (Yang, Lindenbaum, and Kluger 2022), TabNet* (Arik and Pfister 2021). See supplementary material for the full results with all baselines.

sess pairwise significance, visualized via heatmaps.

Implementation Details: We tuned all models with Optuna (Akiba et al. 2019) for 150 trials per dataset following standard tabular deep learning practice of Gorishniy et al. (Gorishniy, Kotelnikov, and Babenko 2025; Gorishniy et al. 2024; Gorishniy, Rubachev, and Babenko 2022; Gorishniy et al. 2021) and evaluated using task-specific cross-validation. Experiments ran on PyTorch 2.4.1+cu121 with AMP across two systems: (1) Windows (Intel, 64GB RAM, RTX 2000 Ada, 16GB VRAM) and (2) TITAN cluster (x86_64, 188GB RAM, TITAN RTX, 24GB VRAM). DDP/DP mitigated out of memory (OOM) issues in transformer models. Full configurations, sources, and DynaTab's optimal hyperparameters are provided in the supplement.

Computational Complexity: DynaTab performs end-to-end training on dynamically reordered features by integrating clustering, graph construction, and rewiring. Feature ordering is computed once per training loop via KMeans ($\mathcal{O}(nkd)$), edge evaluation ($\mathcal{O}(mk^2)$), and centrality-based rewiring ($\mathcal{O}(km^2)$). The reordered input is processed per epoch through the DOF backbone (OPE, PIGL, DMA, and a sequential processor) at cost $\mathcal{O}(nmd)$. Space complexity is $\mathcal{O}(nmd + m^2)$, dominated by the backbone and graph storage. Overall, DynaTab scales linearly with number of samples n, and quadratically with number of features m. (See Table 2). It also supports optional dimensionality reduction for further improved efficiency on high-dimensional data.

Component	**Time Complexity**	**Space Complexity**
KMeans Clustering	$\mathcal{O}(nkd)$	$\mathcal{O}(nd + kd)$
Edge Metric Computation	$\mathcal{O}(mk^2)$	$\mathcal{O}(m^2)$
Graph Rewiring (per cluster)	$\mathcal{O}(km^2)$	$\mathcal{O}(m^2)$
DOF Sequential Backbone	$\mathcal{O}(nmd)$	$\mathcal{O}(nmd)$
Total (DynaTab)	$\mathcal{O}(nkd + mk^2 + km^2 + nmd)$	$\mathcal{O}(nmd + m^2)$

Table 2: Time and space complexity of DynaTab with integrated feature ordering.

Performance Summary: As shown in Table 1 and Figure 3, DynaTab achieves strong performance across diverse data regimes. On eight HDLSS datasets, it ranks best or second-best in most tasks, with the lowest average rank (2.63 ± 2.26). Similarly, on six HDHSS datasets, it ranks first or second on all but one task (average rank of 2.50 ± 2.35), showing robustness in extreme high-dimensional settings. Notably, Lasso and MLP remain highly competitive across several datasets in HDLSS and HDHSS settings, which is consistent with the findings of ProtoGate (Jiang et al. 2024)

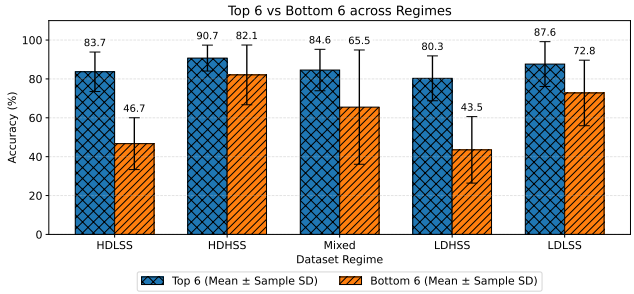


Figure 4: Top-6 vs. Bottom-6 mean accuracies across dataset regimes (HDLSS, HDHSS, Mixed, LDHSS, LDLSS). Error bars show sample standard deviation across the aggregated cells in each group.

for HDLSS that highlight the strength of these baselines relative to heavier architectures. In MixedRegime datasets, DynaTab ranks first on five of eight tasks, yielding the best average rank (6.00 ± 8.72), outperforming methods like LSPIN, TabTransformer, and REAL-X. On regression datasets, it achieves the best average rank (2.00 ± 1.00), ranking first on DrivFace and second on GHG, and third on Cargo. However, on LDHSS and LDLSS datasets, tree-based models such as CatBoost, XGBoost, and GBM outperform DynaTab, which ranks tenth and fourteenth respectively, reflecting its limitations in low-dimensional regimes. Full results and comparisons for all datasets against 45 baselines are provided in the supplementary material.

Regime-level Performance Patterns: Figure 4 summarizes cross-regime performance and shows that high dimensional regimes, particularly HDLSS, remain under-served in tabular deep learning. In HDLSS, the Top-6 average accuracy is 83.7% while the Bottom-6 drops to 46.7% (gap $\approx$ 37.0 points), the largest gap among all regimes. LDHSS exhibits a similar gap (80.3% vs. 43.5%, $\Delta \approx 36.8$), whereas HDHSS narrows to $\approx$ 8.6 points and LDLSS to $\approx$ 14.9 points. The Bottom-6 groups also show larger variability, for example sample SD $= 29.4$ in MixedRegime and $= 17.1$ in LDHSS, indicating instability and hyperparameter brittleness. These patterns are consistent with models struggling when $m \gg n$, motivating capacity-controlled, structure-aware, and scarcity-tuned training methods.

Statistical Significance and Limitations: On HDLSS datasets, DynaTab achieved the best average rank (2.12), outperforming Lasso (3.06), ProtoGate (3.50), and MLP (3.56), while TabulaRNN (4.19) and LGBM (4.56) ranked lower. A Nemenyi test with critical difference (CD = 2.40 at $\alpha = 0.05$) showed only TabulaRNN differed significantly from DynaTab (Holm-adjusted $p < 0.05$). The Friedman test ($\chi^2 = 8.5$, $p = 0.13$) was not significant, suggesting no global performance difference across models. Despite strong results on high-dimensional data, DynaTab faces memory constraints when scaling to large sample sizes (e.g., $>$ 100K), especially during ordering and Transformer-based sequential processing. To address this, we adopt Mamba for efficiency, though DP/DDP may still exacerbate OOM issues in HDHSS settings. Additionally, DynaTab underperforms on low-dimensional datasets, where classical models often suffice. See supplementary material for statistical plots and additional regime-wise analysis.

Backbone	Acc (%) $\pm$ std	AUC $\pm$ std
DAE-MHA-LSTM	80.91 ± 6.76	0.8257 ± 0.0820
Transformer	83.40 ± 5.10	0.8470 ± 0.0740
DAE	81.44 ± 1.04	0.8445 ± 0.0420
LSTM	79.09 ± 3.96	0.8242 ± 0.0589
Mamba	82.18 ± 2.86	0.8315 ± 0.0801

Table 3: DynaTab with different backbones as sequential processor on AI-D (case 5) dataset.

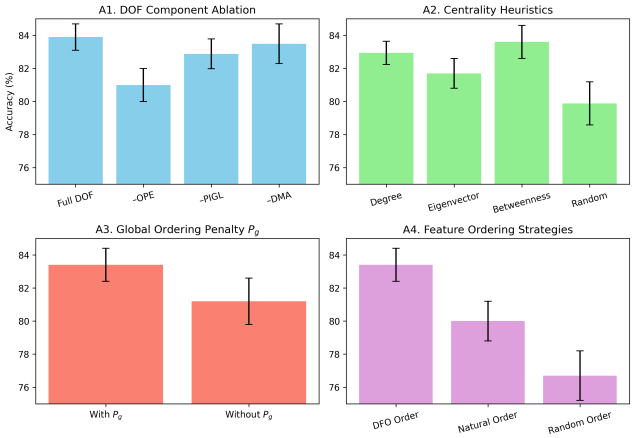


Figure 5: Key ablations on fusion, ordering, and rewiring strategies in DynaTab (See supplementary for more ablation).

Ablation Studies: We run targeted ablations to assess the contribution of core DynaTab components (Figure 5). Removing any part of the DOF fusion module (OPE, PIGL, or DMA) reduces performance (A1), noting their complementary roles. Centrality-based rewiring (A2) consistently outperforms random, validating our structural prior. Disabling the global penalty P_g (A3) leads to clear accuracy drop. Finally, our learned DFO ordering (A4) surpasses natural and random feature orders, confirming the impact of ordering-aware processing. See Table 3 for backbone-specific performance, and supplementary for further ablations.

Conclusion

We presented DynaTab, a neural rewiring-based model that dynamically orders features based on intrinsic structure and fuses them with sequence-aware backbones. A theoretical criterion using IDF and FOE guides when ordering is beneficial. DynaTab rewires local feature graphs into a global sequence, improving inductive bias and performance across 36 tabular datasets. It gains strong results on high-dimensional complex and mixed-regime tasks, with statistically significant gains (Friedman, CD, Wilcoxon-Holm). Limitations include lower gains on low-dimensional or extreme sample-size regimes. Future work includes scaling, efficient backbones, and extensions to multi-view and continual learning.

Acknowledgment

This work was supported in part by grants from the US National Science Foundation (Award #1920920, #2125872, and #2223793).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

Ahamed, M. A.; and Cheng, Q. 2024. MambaTab: A Plug-and-Play Model for Learning Tabular Data. In *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, 369–375. IEEE.

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.

Bassett, D. S.; and Bullmore, E. 2006. Small-World Brain Networks. *The Neuroscientist*, 12(6): 512–523.

Bird, J. J.; Ekart, A.; Buckingham, C. D.; and Faria, D. R. 2019. Mental Emotional Sentiment Classification with an EEG-Based Brain-Machine Interface. In *Proceedings of the International Conference on Digital Image and Signal Processing (DISP'19)*.

Bliss, T. V.; and Collingridge, G. L. 1993. A Synaptic Model of Memory: Long-Term Potentiation in the Hippocampus. *Nature*, 361(6407): 31–39.

Bonacich, P. 1972. Factoring and Weighting Approaches to Status Scores and Clique Identification. *Journal of Mathematical Sociology*, 2(1): 113–120.

Brahmavar, S. B.; Li, Y.; and Oliva, J. 2025. Towards Universal Neural Inference. *arXiv preprint arXiv:2508.09100*.

Bullmore, E.; and Sporns, O. 2009. Complex Brain Networks: Graph Theoretical Analysis of Structural and Functional Systems. *Nature Reviews Neuroscience*, 10(3): 186–198.

Chen, J.; Liao, K.; Wan, Y.; Chen, D. Z.; and Wu, J. 2022. DANets: Deep Abstract Networks for Tabular Data Classification and Regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 3930–3938.

Chen, J.; Song, L.; Wainwright, M.; and Jordan, M. 2018. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. In *International Conference on Machine Learning*, 883–892. PMLR.

Chen, K.-Y.; Chiang, P.-H.; Chou, H.-R.; Chen, T.-W.; and Chang, T.-H. 2023. Trompt: Towards A Better Deep Neural Network for Tabular Data. In *International Conference on Machine Learning*, 4392–4434. PMLR.

Chen, P.; Sarkar, S.; Lausen, L.; Srinivasan, B.; Zha, S.; Huang, R.; and Karypis, G. 2024. HYTREL: Hypergraph-Enhanced Tabular Data Representation Learning. *Advances in Neural Information Processing Systems*, 36.

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. ACM.

Chklovskii, D. B.; Mel, B.; and Svoboda, K. 2004. Cortical Rewiring and Information Storage. *Nature*, 431(7010): 782–788.

Demšar, J. 2006a. Statistical Comparisons of Classifiers Over Multiple Data Sets. *Journal of Machine Learning Research*, 7(Jan): 1–30.

Demšar, J. 2006b. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7: 1–30.

Draganski, B.; Gaser, C.; Busch, V.; Schuierer, G.; Bogdahn, U.; and May, A. 2004. Changes in Grey Matter Induced by Training. *Nature*, 427(6972): 311–312.

Eremeev, D.; Bazhenov, G.; Platonov, O.; Babenko, A.; and Prokhorenkova, L. 2025. Turning Tabular Foundation Models into Graph Foundation Models. In *NeurIPS 2025 New Perspectives in Graph Machine Learning Workshop*.

Freeman, L. C. 1978. Centrality in Social Networks: Conceptual Clarification. *Social networks*, 1(3): 215–239.

Freund, Y.; and Schapire, R. E. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1): 119–139.

Friedman, J. H. 2001. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, 29(5): 1189–1232.

Friedman, M. 1937. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, 32(200): 675–701.

Goodfellow, I. 2016. Deep Learning.

Gorishniy, Y.; Kotelnikov, A.; and Babenko, A. 2025. TabM: Advancing Tabular Deep Learning with Parameter-Efficient Ensembling. In *The Thirteenth International Conference on Learning Representations*.

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.; Kartashev, N.; Shlenskii, D.; Kotelnikov, A.; and Babenko, A. 2024. TabR: Tabular Deep Learning Meets Nearest Neighbors. In *The Twelfth International Conference on Learning Representations*.

Gorishniy, Y.; Rubachev, I.; Khrulkov, V.; and Babenko, A. 2021. Revisiting Deep Learning Models for Tabular Data. In *Advances in Neural Information Processing Systems*, volume 34, 18932–18943.

Govindarajan, A.; Kelleher, R. J.; and Tonegawa, S. 2006. A Clustered Plasticity Model of Long-Term Memory Engrams. *Nature Reviews Neuroscience*, 7(7): 575–583.

Gu, A.; and Dao, T. 2024. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *First Conference on Language Modeling*.

Guan, B.; Rezagholizadeh, M.; Roosta, T. G.; and Passban, P. 2025. The Order Effect: Investigating Prompt Sensitivity to Input Order in LLMs. In *First International KDD Workshop on Prompt Optimization, 2025*.

Guo, H.; Tang, R.; Ye, Y.; Li, Z.; and He, X. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization.

Habib, A. Z. S. B.; Wang, K.; Hartley, M.-A.; Doretto, G.; and Adjeroh, D. A. 2024. TabSeq: A Framework for Deep Learning on Tabular Data via Sequential Ordering. In *International Conference on Pattern Recognition*, 418–434. Springer.

Hanley, J. A.; and McNeil, B. J. 1982. The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology*, 143(1): 29–36.

Hebb, D. O. 2005. *The Organization of Behavior: A Neuropsychological Theory*. Psychology Press.

Hinton, G. E.; and Salakhutdinov, R. R. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786): 504–507.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Hollmann, N.; Müller, S.; Eggensperger, K.; and Hutter, F. 2023. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. In *The Eleventh International Conference on Learning Representations*.

Hollmann, N.; Müller, S.; Purucker, L.; Krishnakumar, A.; Körfer, M.; Hoo, S. B.; Schirrmeister, R. T.; and Hutter, F. 2025. Accurate Predictions on Small Data with a Tabular Foundation Model. *Nature*, 637(8045): 319–326.

Holtmaat, A.; and Svoboda, K. 2009. Experience-Dependent Structural Synaptic Plasticity in the Mammalian Brain. *Nature Reviews Neuroscience*, 10(9): 647–658.

Huang, X.; Khetan, A.; Cvitkovic, M.; and Karnin, Z. 2020. TabTransformer: Tabular Data Modeling Using Contextual Embeddings. *arXiv preprint arXiv:2012.06678*.

Jain, S. K.; and Adjeroh, D. A. 2007. Edge-based Prediction for Lossless Compression of Hyperspectral Images. In *2007 Data Compression Conference (DCC'07)*, 153–162. IEEE.

Jeffares, A.; Liu, T.; Crabbé, J.; Imrie, F.; and van der Schaar, M. 2023. TANGOS: Regularizing Tabular Neural Networks through Gradient Orthogonalization and Specialization. In *The Eleventh International Conference on Learning Representations*.

Jethani, N.; Sudarshan, M.; Aphinyanaphongs, Y.; and Ranganath, R. 2021. Have We Learned to Explain?: How Interpretability Methods Can Learn to Encode Predictions in Their Interpretations. In *International Conference on Artificial Intelligence and Statistics*, 1459–1467. PMLR.

Jiang, X.; Margeloiu, A.; Simidjievski, N.; and Jamnik, M. 2024. ProtoGate: Prototype-based Neural Networks with Global-to-local Feature Selection for Tabular Biomedical Data. In *International Conference on Machine Learning*, 21844–21878. PMLR.

Joseph, M. 2021. PyTorch Tabular: A Framework for Deep Learning with Tabular Data. arXiv:2104.13638.

Kandel, E. R. 2001. The Molecular Biology of Memory Storage: A Dialogue between Genes and Synapses. *Science*, 294(5544): 1030–1038.

Kandel, E. R.; Schwartz, J. H.; Jessell, T. M.; Siegelbaum, S.; Hudspeth, A. J.; Mack, S.; et al. 2000. *Principles of Neural Science*, volume 4. McGraw-hill New York.

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.

Kontschieder, P.; Fiterau, M.; Criminisi, A.; and Bulo, S. R. 2015. Deep Neural Decision Forests. In *Proceedings of the IEEE International Conference on Computer Vision*, 1467–1475.

Kostal, L.; Lansky, P.; and Pokora, O. 2013. Measures of Statistical Dispersion Based on Shannon and Fisher Information Concepts. *Information Sciences*, 235: 214–223.

Kotelnikov, A.; Baranchuk, D.; Rubachev, I.; and Babenko, A. 2023. TabDDPM: Modelling Tabular Data with Diffusion Models. In *International Conference on Machine Learning*, 17564–17579. PMLR.

Lamprecht, R.; and LeDoux, J. 2004. Structural Plasticity and Memory. *Nature Reviews Neuroscience*, 5(1): 45–54.

Levina, E.; and Bickel, P. 2004. Maximum Likelihood Estimation of Intrinsic Dimension. *Advances in Neural Information Processing Systems*, 17.

Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R. P.; Tang, J.; and Liu, H. 2018. Datasets — Feature Selection @ ASU. [Online; accessed 2025-07-25].

Li, S.; Harner, E. J.; and Adjeroh, D. A. 2011. Random KNN Feature Selection-A Fast and Stable Alternative to Random Forests. *BMC Bioinformatics*, 12(1): 450.

Lima, J. R.; Santos, V. G. M.; and Carvalho, M. A. M. 2024. Δ-Evaluation Function for Column Permutation Problems. *arXiv preprint arXiv:2409.04926*.

Liu, B.; Wei, Y.; Zhang, Y.; and Yang, Q. 2017. Deep Neural Networks for High Dimension, Low Sample Size Data. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2287–2293.

Malenka, R. C.; and Bear, M. F. 2004. LTP and LTD: An Embarrassment of Riches. *Neuron*, 44(1): 5–21.

Merzenich, M. M.; and Jenkins, W. M. 1993. Reorganization of Cortical Representations of the Hand Following Alterations of Skin Inputs Induced by Nerve Injury, Skin Island Transfers, and Experience. *Journal of Hand Therapy*, 6(2): 89–104.

Nemenyi, P. B. 1963. *Distribution-Free Multiple Comparisons*. Princeton University.

Nwadiugwu, M. C. 2020. Neural Networks, Artificial Intelligence and the Computational Brain. *arXiv preprint arXiv:2101.08635*.

Ohlsson, M.; Hellmark, T.; Bengtsson, A. A.; Theander, E.; Turesson, C.; Klint, C.; Wingren, C.; and Ekstrand, A. I.

2020. Proteomic Data Analysis for Differential Profiling of the Autoimmune Diseases SLE, RA, SS, and ANCA-Associated Vasculitis. *Journal of Proteome Research*, 20(2): 1252–1260.

OpenTabular Contributors. 2025. deeptab: Tabular Deep Learning Made Simple. https://github.com/OpenTabular/DeepTab. [Online; accessed 2025-07-05].

Park, S. M. 2021. EEG Machine Learning. https://osf.io/8bsvr/. Identifying Psychiatric Disorders Using Machine-Learning (Dataset).

Pascual-Leone, A.; Amedi, A.; Fregni, F.; and Merabet, L. B. 2005. The Plastic Human Brain Cortex. *Annu. Rev. Neurosci.*, 28(1): 377–401.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine Learning In Python. *The Journal of Machine Learning Research*, 12: 2825–2830.

Pham, M. D.; D'Angiulli, A.; Dehnavi, M. M.; and Chhabra, R. 2023. From Brain Models to Robotic Embodied Cognition: How Does Biological Plausibility Inform Neuromorphic Systems? *Brain Sciences*, 13(9): 1316.

Popov, S.; Morozov, S.; and Babenko, A. 2020. Neural Oblivious Decision Ensembles for Deep Learning on Tabular Data. In *International Conference on Learning Representations*.

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.

Qu, J.; Holzmüller, D.; Varoquaux, G.; and Le Morvan, M. 2025. TabICL: A Tabular Foundation Model for In-Context Learning on Large Data. In *International Conference on Machine Learning*, 50817–50847. PMLR.

Rubachev, I.; Kartashev, N.; Gorishniy, Y.; and Babenko, A. 2025. TabReD: Analyzing Pitfalls and Filling the Gaps in Tabular Deep Learning Benchmarks. In *The Thirteenth International Conference on Learning Representations*.

Ruiz, C.; Ren, H.; Huang, K.; and Leskovec, J. 2024. High Dimensional, Tabular Deep Learning with an Auxiliary Knowledge Graph. *Advances in Neural Information Processing Systems*, 36.

Schmidgall, S.; Ziaei, R.; Achterberg, J.; Kirsch, L.; Hajiseyedrazi, S.; and Eshraghian, J. 2024. Brain-Inspired Learning in Artificial Neural Networks: A Review. *APL Machine Learning*, 2(2).

Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-Attention with Relative Position Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 464–468.

Shwartz-Ziv, R.; and Armon, A. 2022. Tabular Data: Deep Learning Is Not All You Need. *Information Fusion*, 81: 84–90.

Sjöström, J.; and Gerstner, W. 2010. Spike-Timing Dependent Plasticity. *Scholarpedia*, 5(2): 1362.

Somepalli, G.; Schwarzschild, A.; Goldblum, M.; Bruss, C. B.; and Goldstein, T. 2022. SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training. In *NeurIPS 2022 First Table Representation Workshop*.

Song, S.; Miller, K. D.; and Abbott, L. F. 2000. Competitive Hebbian Learning through Spike-Timing-Dependent Synaptic Plasticity. *Nature Neuroscience*, 3(9): 919–926.

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.

Tabasum, M.; Tasnim, T.; Islam, M. E.; and Habib, A. Z. S. B. 2025. Tabular Deep Learning vs Classical Machine Learning for Urban Land Cover Classification. In *NeurIPS 2025 Muslims in ML Workshop*.

Tate, S. R. 1997. Band Ordering in Lossless Compression of Multispectral Images. *IEEE Transactions on Computers*, 46(4): 477–483.

Th. Gries, S. 2021. Analyzing Dispersion. In *A Practical Handbook of Corpus Linguistics*, 99–118. Springer.

Thielmann, A. F.; Kumar, M.; Weisser, C.; Reuter, A.; Säfken, B.; and Samiee, S. 2024. Mambular: A Sequential Model for Tabular Deep Learning. *arXiv preprint arXiv:2408.06291*.

Thielmann, A. F.; and Samiee, S. 2024. On the Efficiency of NLP-Inspired Methods for Tabular Deep Learning. In *NeurIPS Efficient Natural Language and Speech Processing Workshop*, 532–539. PMLR.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention Is All You Need. *Advances in Neural Information Processing Systems*, 30.

Veličković, P.; Buesing, L.; Overlan, M.; Pascanu, R.; Vinyals, O.; and Blundell, C. 2020. Pointer Graph Networks. *Advances in Neural Information Processing Systems*, 33: 2232–2244.

Veness, J.; Lattimore, T.; Budden, D.; Bhoopchand, A.; Mattern, C.; Grabska-Barwinska, A.; Sezener, E.; Wang, J.; Toth, P.; Schmitt, S.; et al. 2021. Gated Linear Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 10015–10023.

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and Composing Robust Features with Denoising Autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, 1096–1103.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer Networks. *Advances in Neural Information Processing Systems*, 28.

Wang, C.; Xu, C.; and Lisser, A. 2014. Bandwidth Minimization Problem. In *MOSIM 2014, 10ème Conférence Francophone de Modélisation, Optimisation et Simulation*.

Wang, G.; Chen, Y.; Chen, H.; Fan, X.; Wang, J.; Li, X.; Hu, M.; Chang, C.-Y.; and Hu, X. 2025. Advancing Table Understanding of Large Language Models via Feature Reordering. *ACM SIGKDD Explorations Newsletter*, 27(1): 112–123.

Wang, R.; Fu, B.; Fu, G.; and Wang, M. 2017. Deep & Cross Network for Ad Click Predictions. In *Proceedings of the ADKDD'17*, 1–7. ACM.

Wang, T.; and Guan, S.-U. 2013. Feature Ordering for Neural Incremental Attribute Learning Based on Fisher's Linear Discriminant. In *2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 2, 507–510. IEEE.

Wang, T.; Guan, S.-U.; Ma, J.; and Liu, F. 2015a. Linear Feature Sensibility for Output Partitioning in Ordered Neural Incremental Attribute Learning. In *International Conference on Intelligent Science and Big Data Engineering*, 373–383. Springer.

Wang, T.; Guan, S.-U.; Man, K. L.; Park, J. H.; and Hsu, H.-H. 2015b. Output Effect Evaluation Based on Input Features in Neural Incremental Attribute Learning for Better Classification Performance. *Symmetry*, 7(1): 53–66.

Wang, T.; Guan, S.-U.; Man, K. L.; and Ting, T. 2014. EEG Eye State Identification Using Incremental Attribute Learning with Time-Series Classification. *Mathematical Problems in Engineering*, 2014(1): 365101.

Wang, T.; Zhu, X.; Guan, S.-U.; Man, K. L.; and Ting, T. 2015c. Regression Based on Neural Incremental Attribute Learning with Correlation-based Feature Ordering. In *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 109–113. IEEE.

Yamada, Y.; Lindenbaum, O.; Negahban, S.; and Kluger, Y. 2020. Feature Selection Using Stochastic Gates. In *International Conference on Machine Learning*, 10648–10659. PMLR.

Yan, J.; Chen, J.; Wu, Y.; Chen, D. Z.; and Wu, J. 2023. T2G-Former: Organizing Tabular Features into Relation Graphs Promotes Heterogeneous Feature Interaction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 10720–10728.

Yang, J.; Lindenbaum, O.; and Kluger, Y. 2022. Locally Sparse Neural Networks for Tabular Biomedical Data. In *International Conference on Machine Learning*, 25123–25153. PMLR.

Yang, T.; Wang, Y.; Yue, Z.; Yang, Y.; Tong, Y.; and Bai, J. 2022. Graph Pointer Neural Networks. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 36, 8832–8839.

Yasuda, R. 2017. Biophysics of Synaptic Plasticity: Basic Mechanisms Toward Complex Computation. *Neuroscience Research*, 116: 3–9.

Ye, H.-J.; Yin, H.-H.; Zhan, D.-C.; and Chao, W.-L. 2025. Modern Neighborhood Components Analysis: A Deep Tabular Baseline Two Decades Later. In *The Thirteenth International Conference on Learning Representations*.

Yoon, J.; Jordon, J.; and Van der Schaar, M. 2018. INVASE: Instance-Wise Variable Selection Using Neural Networks. In *International Conference on Learning Representations*.

Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Poczos, B.; Salakhutdinov, R. R.; and Smola, A. J. 2017. Deep Sets. *Advances in Neural Information Processing Systems*, 30.

Zenke, F.; Gerstner, W.; and Ganguli, S. 2017. The Temporal Paradox of Hebbian Learning and Homeostatic Plasticity. *Current Opinion in Neurobiology*, 43: 166–176.

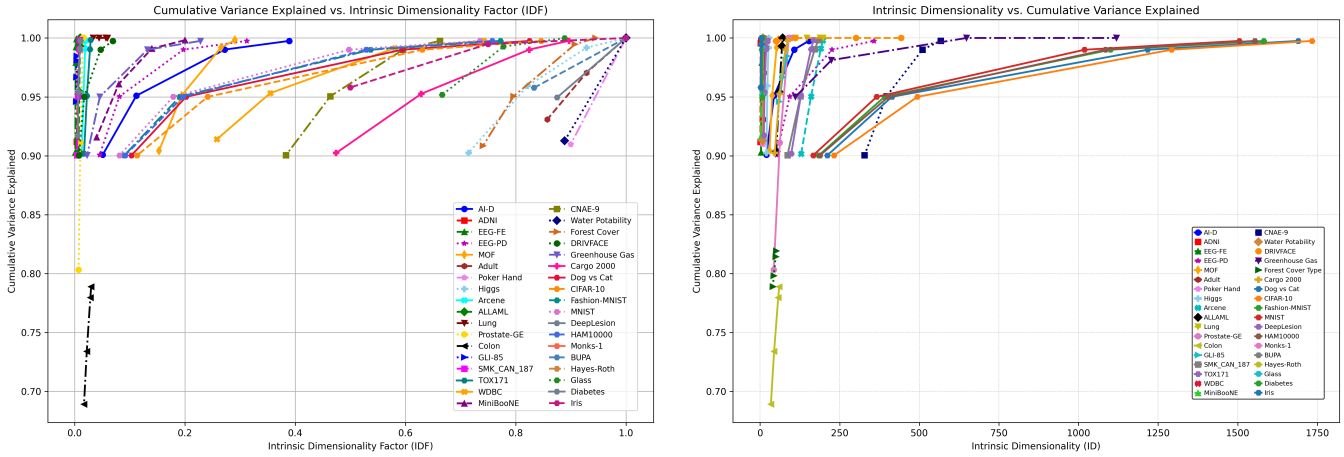
Supplementary Materials

DynaTab: Dynamic Feature Ordering as Neural Rewiring for High-Dimensional Tabular Data

This supplementary document supports our main paper *DynaTab: Dynamic Feature Ordering as Neural Rewiring for High-Dimensional Tabular Data* (Proceedings, AAAI 2026 First International Workshop on Neuro for AI & AI for Neuro: Towards Multi-Modal Natural Intelligence). Specifically, it includes:

- Extended Analysis of Feature Ordering - When to Use?
- Extended Analysis of Datasets
- Baseline Details and Hyperparameters for Selected Models
- DynaTab Hyperparameters
- Discussion on HDLSS Benchmark Results
- Discussion on HDHSS Benchmark Results
- Discussion on Mixed Regime Benchmark Results
- Discussion on LDHSS Benchmark Results
- Discussion on LDLSS Benchmark Results
- Extended Analysis for Statistical Significance
- Extended Analysis on Ablation Studies

Extended Analysis of Feature Ordering - When to Use?



(a) Cumulative Variance vs. IDF

(b) ID vs. Cumulative Variance

Figure A.1: Relationship between cumulative variance and intrinsic dimensionality across all 36 datasets.

This section synthesizes the empirical evidence from the full suite of datasets to answer the core question: under what conditions does feature ordering meaningfully improve performance? We leverage four complementary perspectives: (1) intrinsic dimensionality required to retain varying cumulative variance (Table A.7), (2) success probability as a function of intrinsic dimensionality factor (IDF) (Figure A.3 and the transposed view in Figure A.1), (3) the compounded effectiveness summarized by the FOE-based ranking (Table A.1), and (4) detailed head-to-head contrasts for illustrative pairs (Figure A.2b). Together, they reveal nuanced structure about data compactness, complexity, and amenability to feature ordering.

Intrinsic Dimensionality and Compactness

Table A.7 enumerates the number of principal components needed to achieve four cumulative variance thresholds (99.75%, 99%, 95%, 90%) for each dataset. Datasets such as ADNI and EEG-FE stand out immediately: even at the stringent 99.75% level, their intrinsic dimensionality is tiny (3 and 8 components, respectively), indicating extremely compact representations. This is visually corroborated in Figure A.2a (Cumulative Variance vs Number of Components) and in the alternate projection in Figure A.1b, where their curves spike to near-total variance with minimal components. Such sharp concentration suggests that the variance is dominated by a few coherent directions, making them favorable candidates for interventions that rely

Rank	Dataset	FOE	Complexity	ψ^*	Mean IDF	AUC	Rank	Dataset	FOE	Complexity	ψ^*	Mean IDF	AUC
1	EEG-FE	214326.59	651305.76	3.63e-06	0.00216	0.00191	19	HAM10000	6.46	107.26	0.41792	0.39352	0.6465
2	GLI-85	90256.19	130481.38	1.08e-06	0.00332	0.00104	20	Fashion-MNIST	6.33	111.53	0.43986	0.39758	0.6632
3	SMK_CAN_187	19818.83	48663.27	2.14e-05	0.00710	0.00463	21	Dog vs Cat	5.38	84.82	0.49220	0.43115	0.7016
4	ADNI	17290.68	63064.61	5.50e-05	0.00760	0.00742	22	CIFAR-10	4.77	70.16	0.50455	0.45789	0.7103
5	DeepLesion	14996.21	27611.93	2.65e-05	0.00817	-0.00514	23	WDBC	4.27	13.72	0.22119	0.48387	0.4703
6	ALLAML	12898.16	17653.29	7.18e-06	0.00880	0.00268	24	CNAE-9	3.61	6.13	0.07229	0.52629	0.2689
7	Prostate-GE	6633.18	15459.78	7.32e-05	0.01228	0.00856	25	Iris	2.12	3.83	0.24351	0.68750	0.4935
8	Arcene	3490.53	5418.36	4.19e-05	0.01693	0.00648	26	Cargo 2000	2.01	4.01	0.16437	0.70619	0.4054
9	TOX171	1722.34	3102.51	1.35e-04	0.02410	0.01162	27	Glass	1.78	2.14	0.04784	0.75000	0.2187
10	Colon	1683.01	2384.56	9.32e-05	0.02438	0.00965	28	Forest	1.39	1758.23	0.04024	0.84722	0.2006
11	DRIVFACE	800.17	1.66	0.03841	0.03535	0.1960	29	Higgs	1.33	1.77	0.07584	0.86607	0.2754
12	Lung	423.18	890.30	6.90e-04	0.04861	0.02627	30	BUPA	1.19	1.38	0.02662	0.91667	0.1631
13	Greenhouse	87.12	14749.63	0.00358	0.10714	0.05986	31	Adult	1.12	1.27	0.01912	0.94643	0.1383
14	MiniBooNE	75.61	572.35	0.02426	0.11500	0.1558	32	Diabetes	1.07	1.24	0.01485	0.96875	0.1218
15	EEG-PD	39.54	437.50	0.06769	0.15903	0.2602	33	Water Potability	1.06	1.16	0.01129	0.97222	0.1063
16	AI-D	23.54	347.82	0.10789	0.20611	0.3285	34	Poker	1.05	1.12	0.00912	0.97500	0.0955
17	MOF	19.62	38.60	0.01759	0.22577	0.1326	35	Monks-1	undefined	1.00	1.0	1.00000	0.0000
18	MNIST	7.18	135.41	0.40324	0.37317	0.6350	36	Hayes-Roth	undefined	1.00	1.0	1.00000	0.0000

Table A.1: Ranking of datasets based on FOE score (* = optimized value). "undefined" means FOE was infinite and treated as bottom in ranking.

Parameter	Value	Parameter	Value
Num Clusters	12	Backbone	Transformer
Order	Descending	Mutation Prob	0.2
Tolerance	0.021	Loss Mode	DFO
λ_{disp}	0.4	λ_{global}	0.3
Learning Rate	0.001	d_model	256
nhead	4	Num Layers	3
Window Size	64	Metric	euclidean

Table A.2: Optuna-tuned parameters for DynaTab on the AI-D (Case 5) dataset.

Parameter	Value	Parameter	Value
Num Clusters	2	Backbone	Transformer
Order	Ascending	Mutation Prob	0.0
Tolerance	0.001	Loss Mode	DFO
λ_{disp}	0.0	λ_{global}	0.0
Learning Rate	0.001	d_model	128
nhead	8	Num Layers	2
Window Size	32	Metric	manhattan

Table A.3: Optuna-tuned parameters for DynaTab on the Pima Indian dataset.

on identifying and preserving key directions, such as feature ordering. In contrast, datasets such as Dog vs. Cat, CIFAR-10, Fashion-MNIST, and HAM10000 require hundreds to over a thousand components to reach high-variance thresholds, reflecting diffuse information spread across many directions. Their cumulative variance curves rise much more gradually (Figure A.2a), signifying that there is no small subspace capturing the bulk of the variability. This inherently limits how concisely their structure can be exploited, making effective feature ordering more challenging unless the ordering can reliably isolate the few most informative features among many. GLI-85, updated in the success probability table to have very high success probabilities (e.g., 0.9962 at 99.75%), also shows moderately low intrinsic dimensionality (84 components for 99.75%), situating it in a middle regime: not as concentrated as ADNI/EEG-FE, but much more so than the image and high-variability datasets. Its performance will be better understood in later subsections when juxtaposing intrinsic compactness with ordering effectiveness.

Intrinsic Dimensionality Factor (IDF) and Success Probability

The Intrinsic Dimensionality Factor (IDF), defined as the ratio of intrinsic dimensionality to the true (ambient) dimensionality, normalizes these counts and allows cross-dataset comparison. Figure A.3 shows success probability of feature ordering as a function of IDF for each dataset; Figure A.1 offers a transposed, multi-view layout that emphasizes per-dataset trends across the same axis space. A consistent and informative pattern emerges: datasets with very low IDF (i.e., compact intrinsic representations like ADNI, EEG-FE, GLI-85, ALLAML, and DeepLesion) maintain high success probability across a wide range of variance thresholds. Their curves in Figure A.3 are relatively flat and positioned near the top, indicating that even as more variance is required (which typically increases intrinsic dimensionality and hence IDF), the degradation in success probability is marginal. This suggests that when variance is concentrated in a few directions, feature ordering can reliably prioritize those directions early, yielding high success even if one expands the retained subspace slightly. By contrast, datasets with large IDF such as CIFAR-10, Dog vs Cat, Fashion-MNIST, and Adult exhibit steep declines in success probability as IDF increases. Their curves in Figure A.1 exhibit a clear decreasing trend with IDF (see Fig. A.1(b)): as higher intrinsic dimensionality (thus larger IDF) is needed to explain the same variance, the success probability of feature ordering drops rapidly. This indicates that

Parameter	Value	Parameter	Value
Num Clusters	20	Backbone	Mamba
Order	Descending	Mutation Prob	0.2
Tolerance	0.021	Loss Mode	DFO
λ_{disp}	0.1	λ_{global}	0.1
Learning Rate	0.0001	d_model	128
Num Layers	3	Dropout	0.4
Metric	manhattan		

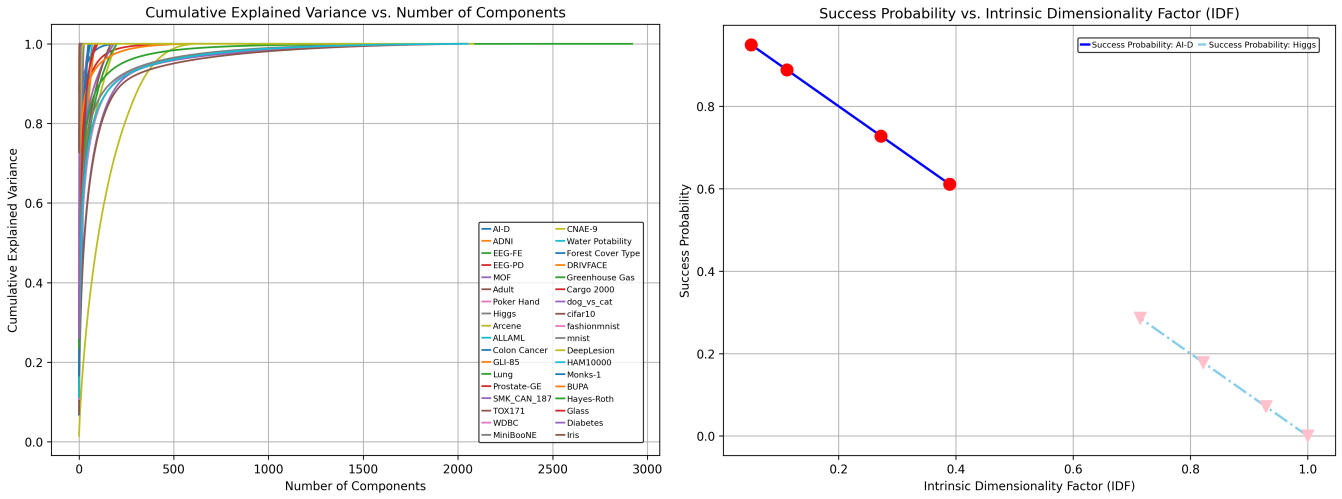
Table A.4: Optuna-tuned parameters for DynaTab on the Arcene dataset.

Parameter	Value	Parameter	Value
Num Clusters	5	Backbone	Transformer
Order	Descending	Mutation Prob	0.2
Tolerance	0.021	Loss Mode	DFO
λ_{disp}	0.1	λ_{global}	0.1
Learning Rate	0.0001	d_model	128
nhead	8	Num Layers	2
Window Size	32	Metric	manhattan

Table A.5: Optuna-tuned parameters for DynaTab on the MiniBooNE dataset.

Parameter	Value	Parameter	Value
Num Clusters	20	Backbone	Mamba
Order	Descending	Mutation Prob	0.2
Tolerance	0.021	Loss Mode	DFO
λ_{disp}	0.1	λ_{global}	0.1
Learning Rate	5e-5	d_model	128
Num Layers	3	Dropout	0.4
Batch Size	64	Accum. Steps	2
Clip Norm	1.0	Metric	manhattan

Table A.6: Optuna-tuned parameters for DynaTab on the HAM10000 dataset.



(a) (b)

Figure A.2: Additional analyses: (a) cumulative variance vs. number of components across datasets; (b) head-to-head success probability of feature ordering vs. IDF for AI-D and Higgs.

when variance is dispersed across many directions, so that no small subset dominates, ordering becomes a weaker proxy for true feature importance, and early selected features are more likely to miss critical structure, leading to lower success. Some datasets occupy intermediate regimes (e.g., MOF, EEG-PD, WDBC): their curves show moderate slopes and partial retention of ordering benefit. Their performance depends on the variance threshold at looser thresholds (e.g., 90%), ordering regains more utility because the effective IDF is smaller, but at tighter ones, ordering becomes less reliable.

Success Probability at Intrinsic Dimensionality and the Effect of Compactness

Table A.8 (with the corrected GLI-85 row) reports success probability evaluated exactly at the intrinsic dimensionality for each variance level. This isolates the performance when the ordering is tasked to capture the minimal subspace necessary for the desired variance, making it a clean reflection of whether the ordering aligns with the true dominant directions. Datasets like EEG-FE, ALLAML, SMK_CAN_187, Prostate-GE, and DeepLesion achieve extremely high success probabilities (often exceeding 0.99) even at the most stringent 99.75% threshold, reinforcing that their intrinsic structure is both compact and ordered in a way that the selected features are strongly informative. GLI-85's high success across all variants (0.9962 to 0.9973) further solidifies it as a dataset where feature ordering captures the key dimensions almost perfectly. On the lower end, datasets such as Monks-1, Hayes-Roth, and some of the small tabular datasets (e.g., BUPA, Iris at certain thresholds) have either zero or very low success probabilities. Their extreme values (e.g., zero for Monks-1 and Hayes-Roth at most thresholds) point to pathological regimes where either variance is too uniformly spread relative to dimension (making ordering noisy), or the notion of ordering does not meaningfully separate signal from noise given the dataset structure. For example, Monks-1 has intrinsic dimensionality equal to its ambient dimension (IDF=1) across thresholds, leaving virtually no ordering leverage. Intermediate datasets like AI-D, EEG-PD, and MOF show moderate success suggesting that while there is some alignment between ordering and informative directions, it is not sufficiently sharp to be a panacea, especially at higher variance retention (e.g., 99.75%).

Feature Ordering Effectiveness (FOE) and Dataset Complexity

Table A.1 aggregates the above findings into a composite ranking based on the FOE score, along with auxiliary metrics: complexity, optimal ψ^* (ordering parameter), mean IDF, and AUC. The FOE score encapsulates the net benefit of ordering relative to complexity and variance retention behavior. Top-ranked datasets (EEG-FE, GLI-85, SMK_CAN_187, ADNI, DeepLesion, ALLAML) share the empirical fingerprints identified earlier: low mean IDF, high compactness, and correspondingly high success probabilities. Their high complexity values despite compactness reveal a subtle point: complexity here is not merely about dimensional spread but about nuanced structure (e.g., subtle patterns, correlations, biological signal) that makes feature ordering valuable. High FOE alongside nontrivial complexity suggests ordering both uncovers and leverages structured signal that might otherwise be obscured in naive representations. Mid-tier datasets (e.g., Colon, TOX171, DRIVFACE, Lung) exhibit a mix of moderate mean IDF and non-negligible complexity. Their FOE scores, while lower, are still positive, indicating that ordering provides benefit, albeit less spectacular and more sensitive to the choice of parameters (reflected in ψ^* variability). Lower-ranked or "undefined" datasets (such as Monks-1 and Hayes-Roth) expose the limits of the approach. Their FOE is infinite/undefined because the ordering fails to improve over the trivial baseline in any meaningful way, and their mean IDF is extreme (close to 1), implying that signal is not concentrated in a way that ordering can exploit. In combination with low effective complexity (relative to how FOE is computed), this yields minimal (or no) gain, underlining a boundary regime where feature ordering should either be deprioritized or rethought.

Pairwise and Head-to-Head Illustrations

Figure A.2b provides a focused head-to-head comparison, e.g., between AI-D and Higgs. The contrast is instructive: AI-D, despite having a moderate IDF, shows a steeper decline in success probability as the IDF increases, whereas in the extreme case of Higgs (which has very low compactness IDF near 1 for high thresholds), the challenge is that ordering has very little room to improve, and success probabilities collapse accordingly. Such pairwise views help clarify that two datasets with similar ambient sizes may behave drastically differently depending on intrinsic structure; thus, global heuristics (like "larger dimension means harder") are insufficient without the nuanced lens of IDF and ordering alignment.

Synthesis and Practical Implications

Bringing the threads together, the following practical insights emerge:

1. **High Benefit Regime:** Datasets with low mean IDF and high FOE (e.g., EEG-FE, ADNI, GLI-85, ALLAML, DeepLesion) are ideal candidates for feature ordering. They combine compact variance concentration with structured complexity, meaning ordering can rapidly isolate informative features and yield large relative gains.
2. **Conditional Benefit Regime:** Datasets with moderate IDF and FOE (e.g., Colon, TOX171, DRIVFACE, Lung) benefit, but performance depends on tuning (optimal ψ^*) and variance target. For looser variance thresholds (lower intrinsic dimensionality), ordering is more effective; for tighter ones, the window narrows.
3. **Low/No Benefit Regime:** Datasets exhibiting high IDF (close to 1), low FOE, or pathological characteristics (e.g., Monks-1, Hayes-Roth, and small tabular sets whose success probability remains nearly constant and low across variance thresholds) provide little to no benefit from feature ordering in the current formulation.
4. **Interpretability of Ordering:** The consistent alignment between compact intrinsic structure and success probability indicates that feature ordering is implicitly discovering principal directions (or strong proxies thereof) when those directions dominate variance. Failure cases often arise when variance is diffuse, meaning ordering needs auxiliary signals (e.g., supervised guidance or domain priors) to differentiate among many weakly informative features.
5. **Role of Complexity:** High dataset complexity does not preclude success; rather, it often coexists with high ordering benefit when that complexity is organized (structured) rather than random. Thus, complexity metrics should be interpreted jointly with IDF and FOE to decide whether effort on feature ordering is warranted.

Recommendations for Usage

- **Pre-screening:** Compute intrinsic dimensionality and IDF at the desired variance level. If IDF is low (e.g., < 0.1) and success probability at that point is empirically high (or inferred from similar datasets), prioritize feature ordering in pipelines.
- **Adaptive Thresholding:** For datasets with moderate IDF, consider relaxing variance retention targets to reduce IDF and improve ordering reliability trading off a bit of variance for more stable ordering gains.
- **Fallback Strategies:** For datasets in the low-benefit regime, either avoid feature ordering or augment it with domain-specific feature importance signals, since naive ordering is unlikely to yield robust improvements.
- **Hyperparameter Tuning:** Use the ψ^* (optimal ordering parameter) profile from Table A.1 as starting points when adapting to new but similar datasets; small deviations in structure can cause the effective ordering scale to shift.

Dataset	99.75%	99%	95%	90%	Dataset	99.75%	99%	95%	90%
AI-D	153	107	44	20	TOX171	168	160	128	98
ADNI	3	2	2	1	DRIVFACE	443	301	111	50
EEG-FE	8	6	5	3	Dog vs Cat	1690	1217	414	211
EEG-PD	358	226	93	52	CIFAR-10	1733	1292	494	232
MOF	57	52	38	30	Fashion-MNIST	1582	1100	391	184
WDBC	23	18	11	8	MNIST	1505	1019	366	167
CNAE-9	567	510	397	328	DeepLesion	23	19	14	12
Water Potability	9	9	9	8	HAM10000	1554	1085	403	188
Forest Cover Type	51	49	43	40	Monks-1	6	6	6	6
Greenhouse Gas	1119	648	224	111	BUPA	6	6	5	5
Cargo 2000	87	80	61	46	Hayes-Roth	5	5	5	5
Arcene	197	191	160	129	Glass	8	7	6	6
ALLAML	71	69	60	51	Diabetes	8	8	8	7
Colon Cancer	60	57	44	34	Iris	4	3	2	2
GLI-85	84	81	71	60	Water Potability	9	9	9	8
Lung	198	188	147	111	Forest Cover Type	51	49	43	40
Prostate-GE	98	90	62	43	Greenhouse Gas	1119	648	224	111
SMK_CAN_187	183	172	127	86	Cargo 2000	87	80	61	46

Table A.7: Intrinsic dimensionality (number of components) required to reach various cumulative variance thresholds for all datasets.

Caveats and Future Directions

While the current analysis unifies variance concentration, ordering alignment, and complexity, it remains largely empirical. Future work could formalize the theoretical link between ordering heuristics and the spectral properties of the data covariance in regimes of diffuse variance. Additionally, developing hybrid orderings that incorporate supervised signals or nonlinear embeddings may recover utility for datasets currently in the low-benefit regime.

We showcase the success probability of feature ordering methods for each dataset at different variance levels. Datasets such as EEG-FE and ADNI, which have lower intrinsic dimensionality and fewer required components (as seen in Figure A.3), achieve higher success probabilities consistently across all variance levels. In contrast, datasets like Poker Hand and Adult show poor success probabilities, particularly at higher variance levels, indicating limited benefits from feature ordering for these datasets. Figure A.1 presents the relationship between success probability and the Intrinsic Dimensionality Factor (IDF) for different datasets. Each subplot corresponds to a specific dataset, highlighting the decline in success probability with increasing IDF. Notably, datasets with compact representations, such as ADNI and EEG-FE, exhibit a shallow decline, maintaining high success probabilities even at larger IDFs. Conversely, datasets like Poker Hand and Adult demonstrate a steep decline, signifying their lower amenability to feature ordering. We provide a comprehensive ranking of datasets based on Feature Ordering Effectiveness (FOE), alongside their parameter namely, Optimal ψ, Mean IDF, AUC, and overall dataset complexity score. The complexity score measures the inherent complexity of each dataset, offering insights into the challenge solved by feature ordering. Datasets such as EEG-FE and ADNI demonstrate both high FOE and high complexity scores, indicating that while they are amenable to feature ordering, their specialized and focused nature presents unique challenges. Conversely, datasets like Poker Hand and Adult exhibit low complexity scores and FOE values, reflecting their limited potential for improvement with feature ordering.

Extended Analysis of Datasets

We provide detailed analysis of all datasets used in this study. The 36 datasets span various regimes: HDLSS, HDHSS, LDHSS, LDLSS, and MixedRegime. Table B.1 summarizes their properties, including sample and feature counts, class distribution, FOE score, AUC, and feature-to-sample ratio.

MixedRegime datasets

We denote a dataset as MixedRegime when it does not satisfy the threshold conditions of the HDLSS/HDHSS/LDHSS/LDLSS categories (i.e., it falls into the "otherwise" case in our empirical stratification based on m, n, and $\rho = \frac{m}{n}$). AI-D comprises 239 samples with 393 proteomic features. It includes measurements from patients with autoimmune diseases and healthy controls. To generate five binary classification tasks, samples were grouped into one-vs-rest splits. For example, Case 5 distinguishes SS from the other three diseases (SLE, RA, SV). Despite moderate dimensionality, the dataset shows FOE of 23.5 and a feature-to-sample ratio of 1.64, suggesting that reordering aids in uncovering relevant signals. ADNI contains 177 samples with 263 imaging-derived features. Three-way classification targets the stages of Alzheimer's progression (0=Normal, 1=Mild Cognitive Impairment, 2=AD). The dataset demonstrates a very high FOE score of 1.73e4 and a low AUC of 0.0074, indicating that

Dataset	99.75%	99%	95%	90%	Dataset	99.75%	99%	95%	90%
AI-D	0.6107	0.7277	0.8880	0.9491	TOX171	0.9708	0.9722	0.9777	0.9830
ADNI	0.9886	0.9924	0.9924	0.9962	DRIVFACE	0.9308	0.9530	0.9827	0.9922
EEG-FE	0.9969	0.9976	0.9980	0.9988	Dog vs Cat	0.1748	0.4058	0.7979	0.8970
EEG-PD	0.6876	0.8028	0.9188	0.9546	CIFAR-10	0.1538	0.3691	0.7588	0.8867
MOF	0.7092	0.7347	0.8061	0.8469	Fashion-MNIST	0.2275	0.4629	0.8091	0.9102
WDBC	0.2581	0.4194	0.6452	0.7419	MNIST	0.2651	0.5024	0.8213	0.9185
CNAE-9	0.3376	0.4042	0.5362	0.6168	DeepLesion	0.9890	0.9909	0.9933	0.9942
Water Potability	0.0000	0.0000	0.0000	0.1111	HAM10000	0.2427	0.4712	0.8036	0.9084
Forest Cover Type	0.0556	0.0926	0.2037	0.2593	MiniBooNE	0.8000	0.8600	0.9200	0.9600
Greenhouse Gas	0.7719	0.8679	0.9543	0.9774	Adult	0.0000	0.0000	0.0714	0.1429
Cargo 2000	0.1031	0.1753	0.3711	0.5258	Poker Hand	0.0000	0.0000	0.0000	0.1000
Arcene	0.9803	0.9809	0.9840	0.9871	Higgs	0.0000	0.0714	0.1786	0.2857
ALLAML	0.9900	0.9903	0.9916	0.9928	Monks-1	0.0000	0.0000	0.0000	0.0000
Colon Cancer	0.9700	0.9715	0.9780	0.9830	BUPA	0.0000	0.0000	0.1667	0.1667
GLI-85	0.9962	0.9964	0.9968	0.9973	Hayes-Roth	0.0000	0.0000	0.0000	0.0000
Lung	0.9402	0.9432	0.9556	0.9665	Glass	0.1111	0.2222	0.3333	0.3333
Prostate-GE	0.9836	0.9849	0.9896	0.9928	Diabetes	0.0000	0.0000	0.0000	0.1250
SMK_CAN_187	0.9908	0.9914	0.9936	0.9957	Iris	0.0000	0.2500	0.5000	0.5000

Table A.8: Success probability at various variance levels for each dataset, evaluated at the corresponding intrinsic dimensionality.

meaningful structure exists but is difficult to capture without feature ordering. EEG-FE (2132 samples, 2549 features) and EEG-PD (941 samples, 1146 features) contain EEG-derived features for emotion and psychiatric disorder classification, respectively. Both datasets have $\rho > 1$ and FOE scores of 2.14e5 and 39.5, supporting the importance of reordering in brain signal modeling. MOF includes 61 ICU patient records with 196 features. Despite its small size, the dataset yields a FOE of 19.6 and a high ratio ($\rho = 3.21$), indicating concentrated variance in a subset of features. Additional MixedRegime datasets include WDBC, CNAE-9, Water Potability, Cargo 2000, and Greenhouse Gas. These datasets show diverse FOE scores (ranging from 1.06 to 87.1), making them ideal for evaluating the generalization ability of feature ordering methods across moderate-dimensional tabular domains.

HDLSS datasets

This group includes highly imbalanced, small-sample datasets with very high feature counts, where the feature-to-sample ratio exceeds 2. Arcene (200 samples, 10000 features), ALLAML (72, 7129), Colon Cancer (62, 2000), GLI-85 (85, 22283), Lung (203, 3312), Prostate-GE (102, 5966), SMK_CAN_187 (187, 19993), and TOX171 (171, 5748) are classic gene expression or microarray datasets. All have FOE scores well above 1000, confirming that most variance lies in a small subset of features. These datasets are well-suited for studying dynamic ordering under extreme sample-scarcity conditions. DrivFace, with 606 samples and 6400 image-derived features, also falls in this group. The regression task (facial angle prediction) benefits from ordering, as shown by its FOE score of 800 and an AUC of 0.196.

HDHSS datasets

This regime includes high-dimensional datasets with large sample counts and moderate ratios (typically $\rho \approx 0.18$). All are derived from images and use deep feature embeddings. DeepLesion, HAM10000, MNIST, Fashion-MNIST, CIFAR-10, and Dog vs Cat are each subsampled to 11000 samples and 2048 features. Their FOE scores range from 4.77 to 1.5e4. High AUCs (above 0.63 for all datasets and up to 0.994 for DeepLesion) indicate that ordering improves predictive performance in settings with moderately redundant visual features.

LDHSS datasets

LDHSS datasets have large sample sizes and low-dimensional feature sets. These include Adult (32561, 14), Poker Hand (25010, 10), Higgs (8.9 million, 28), Forest Cover (581012, 54), and MiniBooNE (130064, 50). All have extremely low feature-to-sample ratios (below 0.001) and FOE values under 76. In these cases, variance is evenly distributed across all features, and feature ordering shows minimal effectiveness.

LDLSS datasets

This group includes traditional low-dimensional, small-sample datasets such as Iris, Pima Indian Diabetes, Monks-1, Glass, Hayes-Roth, and BUPA Liver. FOE values range from 1.07 to 2.12, with two datasets (Monks-1 and Hayes-Roth) yielding undefined FOE due to degenerate eigenvalue spectra in PCA. These datasets are not suited for dynamic ordering, but serve as sanity checks and efficient benchmarks.

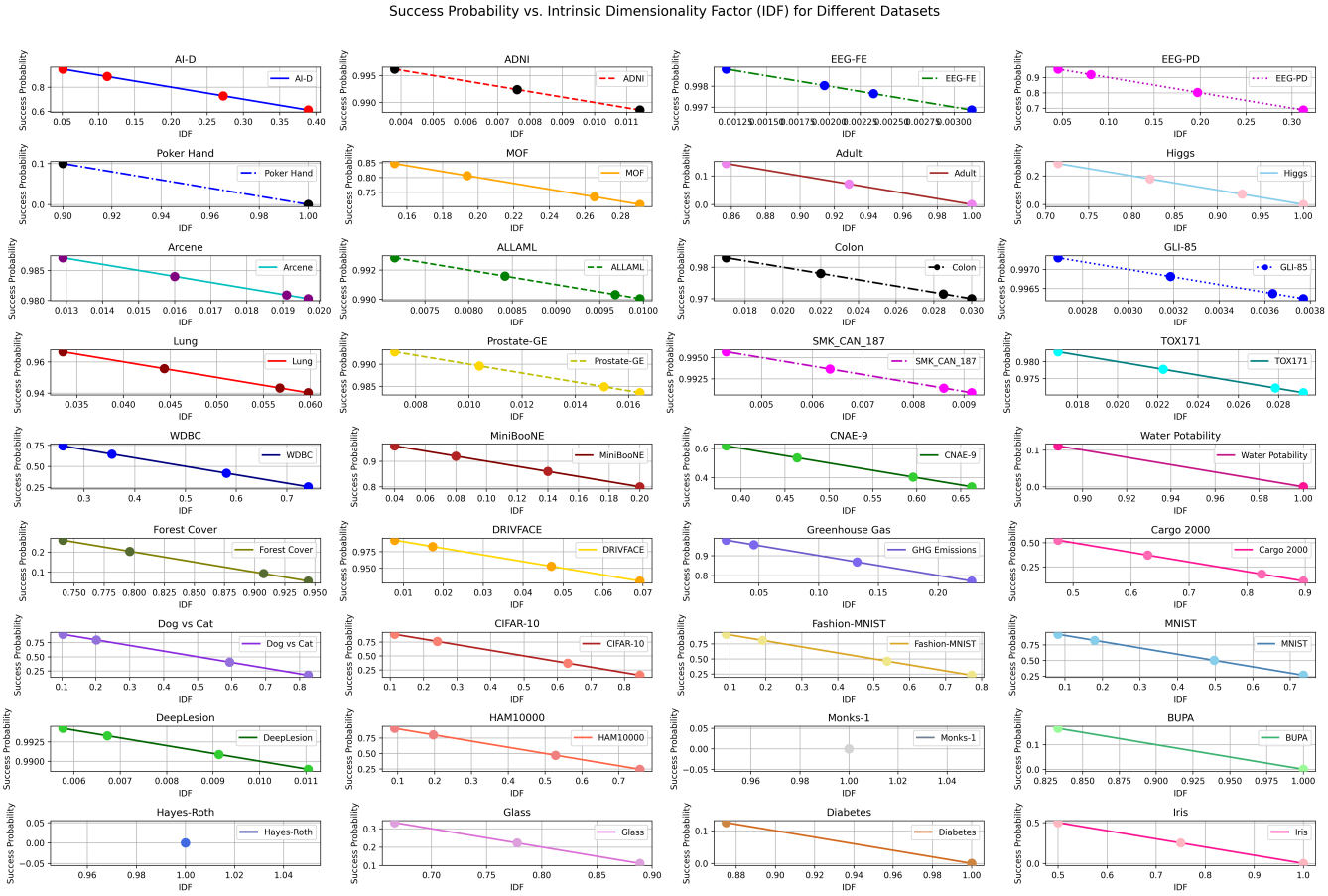


Figure A.3: Success probability of Feature Ordering vs IDF for Different Datasets.

Additional FOE-based analysis

We analyzed 36 datasets to evaluate the effectiveness of feature ordering. High dimensional datasets showed strong behavior but lower dimensional datasets showed poor performance. Specifically, Adult, Poker Hand, and Higgs showed poor FOE scores (1.12, 1.05, and 1.33, respectively) and slow variance convergence, making them less suitable for reordering-based improvements. Their low cumulative variance and high intrinsic dimensionality (IDF) confirm that information is broadly distributed. These datasets serve as control baselines where feature ordering is expected to offer limited value. Table B.1 summarize the key statistics across all dataset types, supporting the categorization and selection strategies used throughout the experiments.

Baseline Details and Hyperparameters for Selected Models

We provide detailed commentary on all evaluated models in this study. The models span classical machine learning, deep neural networks, tree ensembles, attention-based transformers, interpretable models, and recent innovations in tabular deep learning. Each model is contextualized by its design motivations, architecture, and empirical strengths or limitations.

Naive Bayes (Pedregosa et al. 2011) A generative probabilistic model that assumes independence between features given the class label. Despite its simplicity, it performs well in high-bias domains, particularly text classification. Its closed-form parameter estimation and fast inference make it useful as a baseline in low-resource settings.

KNN (Pedregosa et al. 2011) An instance-based learner that classifies data based on the majority class among the k nearest neighbors in feature space. While KNN is highly interpretable and non-parametric, it suffers from the curse of dimensionality and lacks scalability to large datasets.

SVM (Pedregosa et al. 2011) Support Vector Machines maximize the margin between class boundaries using kernel methods to transform the data into higher-dimensional feature spaces. SVMs offer strong generalization but require careful kernel and regularization tuning, and do not scale efficiently with large sample sizes.

Dataset	n	m	#Cls	Distribution / Note	Target	Category	Eval	FOE	AUC	ρ	Justification
AI-D	239	393	2	0=167, 1=72	Status	MixedRegime	5–Fold CV	23.5	0.3280	1.64	$m \leq 1000,\ n \leq 1000,\ \rho > 0.05$
ADNI	177	263	3	0=33, 1=36, 2=108	AD123	MixedRegime	5–Fold CV	1.73e+04	0.0074	1.49	$m \leq 1000,\ n \leq 1000,\ \rho > 0.05$
EEG-FE	2132	2549	3	0=708, 1=716, 2=708	label	MixedRegime	5–Fold CV	2.14e+05	0.0019	1.20	$m > 1000,\ n < 10^4,\ \rho > 0.05$
EEG-PD	941	1146	2	0=95, 1=846	main.disorder	MixedRegime	5–Fold CV	39.5	0.2600	1.22	$m > 1000,\ n < 1000,\ \rho > 0.05$
MOF	61	196	2	0=26, 1=35	Label	MixedRegime	5–Fold CV	19.6	0.1330	3.21	$m \leq 1000,\ n < 1000,\ \rho > 0.05$
Adult	32561	14	2	0=24720, 1=7841	income	LDHSS	5×5 CV	1.12	0.138	0.00043	$m \leq 100,\ n > 10^4,\ \rho \leq 0.01$
Poker Hand	25010	10	9	0=12493, 1=10599,…	Label	LDHSS	5×5 CV	1.05	0.0950	0.00040	$m \leq 100,\ n > 10^4,\ \rho \leq 0.01$
Higgs	8978568	28	2	0=4 221 016, 1=4 757 552	regression	LDHSS	5×5 CV	1.33	0.275	0.00000	$m \leq 100,\ n > 10^4,\ \rho \leq 0.01$
Arcene	200	10000	2	0=112, 1=88	Label	HDLSS	5×5 CV	3.49e+03	0.0065	50.00	$m > 1000,\ n < 1000,\ \rho > 2$
ALLAML	72	7129	2	0=47, 1=25	Label	HDLSS	5×5 CV	1.29e+04	0.0027	99.01	$m > 1000,\ n < 1000,\ \rho > 2$
Colon Cancer	62	2000	2	0=40, 1=22	Label	HDLSS	5×5 CV	1.68e+03	0.0097	32.26	$m > 1000,\ n < 1000,\ \rho > 2$
GLI-85	85	22283	2	0=26, 1=59	Label	HDLSS	5×5 CV	9.03e+04	0.0010	262.15	$m > 1000,\ n < 1000,\ \rho > 2$
Lung	203	3312	5	0=139, 1=17, 2=21,…	Label	HDLSS	5×5 CV	423.2	0.0260	16.32	$m > 1000,\ n < 1000,\ \rho > 2$
Prostate-GE	102	5966	2	0=50, 1=52	Label	HDLSS	5×5 CV	6.63e+03	0.0086	58.49	$m > 1000,\ n < 1000,\ \rho > 2$
SMK_CAN_187	187	19993	2	0=90, 1=97	Label	HDLSS	5×5 CV	1.98e+04	0.0046	106.91	$m > 1000,\ n < 1000,\ \rho > 2$
TOX171	171	5748	4	0=45, 1=45, 2=39, 3=42	Label	HDLSS	5×5 CV	1.72e+03	0.0116	33.61	$m > 1000,\ n < 1000,\ \rho > 2$
WDBC	569	31	2	0=357, 1=212	diagnosis	MixedRegime	5×5 CV	4.27	0.470	0.055	$m \leq 100,\ n < 1000,\ \rho > 0.05$
MiniBooNE	130064	50	2	0=93 565, 1=36 499	label	LDHSS	5×5 CV	75.61	0.156	0.00038	$m \leq 100,\ n > 10^4,\ \rho \leq 0.01$
CNAE-9	1080	856	9	0…8 each 120	label	MixedRegime	5–Fold CV	3.61	0.269	0.79	$m < 10^4,\ n < 10^4,\ \rho > 0.05$
Water Potability	3276	9	2	0=1998, 1=1278	Potability	MixedRegime	5–Fold CV	1.06	0.106	0.0028	$m \leq 100,\ n < 10^4,\ \rho \leq 0.05$
Forest Cover Type	581012	54	7	0=211 840, 1=283 301,…	Cover_Type	LDHSS	5×5 CV	1.39	0.021	0.00009	$m \leq 100,\ n > 10^4,\ \rho \leq 0.01$
DrivFace	606	6400	–	Face-angle prediction	angle	HDLSS	5×5 CV	800.2	0.196	10.56	$m > 1000,\ n < 1000,\ \rho > 2$
Greenhouse Gas	2920	4905	–	GHG concentrations…	regression	MixedRegime	5–Fold CV	87.1	0.060	1.68	$m < 10^4,\ n < 10^4,\ \rho > 0.05$
Cargo 2000	3943	97	–	actual delivery time	o_dlv_e	MixedRegime	5–Fold CV	2.01	0.405	0.025	$m \leq 100,\ n < 10^4,\ \rho \leq 0.05$
HAM10000	10015	2052	7	0=327, 1=514, 2=109,…	dx	HDHSS	5–Fold CV	6.46	0.647	0.205	$m > 1000,\ n > 10^4,\ 0.005 < \rho \leq 2$
DeepLesion	11000	2082	3	0=9343, 1=807, 2=850	Train_Val_Test	HDHSS	5–Fold CV	1.50e+04	0.994	0.189	$m > 1000,\ n > 10^4,\ 0.005 < \rho \leq 2$
MNIST+	11000	2048	10	0=1017,…,9=1053	label	HDHSS	5–Fold CV	7.18	0.635	0.186	$m > 1000,\ n > 10^4,\ 0.005 < \rho \leq 2$
Fashion-MNIST+	11000	2048	10	0=1096,…,9=1087	label	HDHSS	5–Fold CV	6.33	0.663	0.186	$m > 1000,\ n > 10^4,\ 0.005 < \rho \leq 2$
CIFAR-10+	11000	2048	10	0=1100,…,9=1090	label	HDHSS	5–Fold CV	4.77	0.710	0.186	$m > 1000,\ n > 10^4,\ 0.005 < \rho \leq 2$
Dog vs Cat+	11000	2048	2	0=5553, 1=5447	label	HDHSS	5–Fold CV	5.38	0.702	0.186	$m > 1000,\ n > 10^4,\ 0.005 < \rho \leq 2$
Iris	150	4	3	0=50, 1=50, 2=50	class	LDLSS	5×5 CV	2.12	0.494	0.027	$m \leq 100,\ n < 1000,\ \rho \leq 0.05$
Pima Indian Diabetes	768	8	2	0=500, 1=268	Outcome	LDLSS	5×5 CV	1.07	0.122	0.010	$m \leq 100,\ n < 1000,\ \rho \leq 0.05$
Monks-1	432	6	2	0=216, 1=216	label	LDLSS	5×5 CV	undefined	0.000	0.014	$m \leq 100,\ n < 1000,\ \rho \leq 0.05$
Glass	214	9	6	0=70, 1=76,…	Glass_Type	LDLSS	5×5 CV	1.78	0.219	0.042	$m \leq 100,\ n < 1000,\ \rho \leq 0.05$
Hayes-Roth	132	5	3	0=51, 1=51, 2=30	class	LDLSS	5×5 CV	undefined	0.000	0.038	$m \leq 100,\ n < 1000,\ \rho \leq 0.05$
BUPA Liver	345	6	2	0=257, 1=88	label	LDLSS	5×5 CV	1.19	0.163	0.017	$m \leq 100,\ n < 1000,\ \rho \leq 0.05$

Table B.1: Summary of all 36 datasets: sample count n, feature count m, number of classes (#Cls; "–" for regression), class distribution (or regression note), target column, category, evaluation strategy, FOE score, dataset AUC, feature-to-sample ratio $\rho = m/n$, and rule-based justification. Datasets marked "+" are 11 k–sample subsamples of the full originals.

HDLSS: $m > 1000, n < 1000, \rho > 2$; HDHSS: $m > 1000, n > 10^4, 0.005 < \rho \leq 2$; LDHSS: $m \leq 100, n > 10^4, \rho \leq 0.01$; LDLSS: $m \leq 100, n \leq 1000, \rho \leq 0.05$; MixedRegime: otherwise.

↓ Metric/Cluster Size →	7	9	12	15
Variance	0.5647	0.5546	0.5648	0.7531
Correlation	0.3054	0.5138	0.5019	0.7196
Euclidean	0.3906	0.3851	0.3937	0.3960
Manhattan	0.4791	0.5163	0.8151	1.6975
KL Divergence	107.0911	137.7239	183.8071	229.2274

Table B.2: Runtime (seconds) for GPU-enabled similarity metrics in DynaTab's dynamic feature ordering. *Note:* KL divergence times are GPU-based. For 5 clusters, GPU time was 76.89s; CPU time was 25284.89s (7.02 hours).

Decision Tree (Pedregosa et al. 2011) A non-linear model that partitions the input space using axis-aligned splits. Trees are interpretable and handle feature interactions well but can overfit small or noisy datasets. Pruning and regularization are essential for generalization.

Lasso (Pedregosa et al. 2011) A linear model with L1 regularization that induces sparsity in the learned coefficients, enabling embedded feature selection. Lasso is robust to multicollinearity and is widely used in biomedical and financial modeling where interpretability is critical.

MLP (Pedregosa et al. 2011) Multi-Layer Perceptrons consist of fully connected layers with nonlinear activations. MLPs are powerful function approximators but can struggle with tabular data due to lack of inductive bias and sensitivity to feature scaling and distribution shifts.

1-D CNN (Shwartz-Ziv and Armon 2022) One-dimensional convolutional neural networks model local spatial dependencies across feature dimensions. Suitable for time series or ordered feature settings, they can capture local motifs but require fixed input ordering.

Random Forest (Pedregosa et al. 2011) An ensemble of decorrelated decision trees trained via bootstrap aggregation. Random forests reduce variance and are robust to overfitting, offering strong baselines for tabular tasks without requiring extensive tuning.

Dataset	Source URL	Dataset	Source URL
AI-D (Case 5)	https://figshare.com/s/3bd3848a28ef6e7ae9a9?file=15050525	CNAE9	https://archive.ics.uci.edu/dataset/233/cnae+9
ADNI (AD123)	https://adni.loni.usc.edu	Water Potability	https://www.kaggle.com/datasets/adityakadiwal/water-potability
EEG-FE	https://www.kaggle.com/datasets/birdy654/eeg-brainwave-dataset-feeling-emotions	Forest Cover Type	https://archive.ics.uci.edu/dataset/31/covertype
EEG-PD	https://osf.io/8bsvr	DrivFace	https://archive.ics.uci.edu/dataset/378/drivface
MOF	https://www.kaggle.com/datasets/andrewmvd/multi-organ-failure-prediction	Greenhouse Gas	https://archive.ics.uci.edu/dataset/328/greenhouse+gas+observing+network
Adult	https://archive.ics.uci.edu/dataset/2/adult	Cargo 2000	https://archive.ics.uci.edu/dataset/382/cargo+2000+freight+tracking+and+tracing
Poker Hand	https://archive.ics.uci.edu/dataset/158/poker+hand	HAM10000	https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000
Higgs	https://archive.ics.uci.edu/dataset/280/higgs	DeepLesion+	https://www.kaggle.com/datasets/kmader/nih-deeplesion-subset
Arcene	https://jundongl.github.io/scikit-feature/datasets.html	MNIST+	https://www.kaggle.com/datasets/hojjatk/mnist-dataset
ALLAML	https://jundongl.github.io/scikit-feature/datasets.html	Fashion MNIST+	https://www.kaggle.com/datasets/zalando-research/fashionmnist
Colon	https://jundongl.github.io/scikit-feature/datasets.html	CIFAR-10+	https://www.kaggle.com/datasets/pankrzysiu/cifar10-python
GLI-85	https://jundongl.github.io/scikit-feature/datasets.html	Dog vs Cat+	https://www.kaggle.com/datasets/kunalgupta2616/dog-vs-cat-images-data
Lung	https://jundongl.github.io/scikit-feature/datasets.html	Iris	https://archive.ics.uci.edu/dataset/53/iris
Prostate-GE	https://jundongl.github.io/scikit-feature/datasets.html	Pima Indians Diabetes	https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database
SMK_CAN_187	https://jundongl.github.io/scikit-feature/datasets.html	Monks-1	https://archive.ics.uci.edu/dataset/70/monk+s+problems
TOX171	https://jundongl.github.io/scikit-feature/datasets.html	Glass	https://archive.ics.uci.edu/dataset/42/glass+identification
WDBC	https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data	Hayes-Roth	https://archive.ics.uci.edu/dataset/44/hayes+roth
MiniBooNE	https://archive.ics.uci.edu/dataset/199/miniboone+particle+identification	BUPA Liver	https://archive.ics.uci.edu/dataset/60/liver+disorders

Table B.3: Source URLs for all datasets used in this work. Datasets marked with "+" are 11 k–sample subsamples of the full originals.

AdaBoost (Freund and Schapire 1997) A boosting framework that sequentially fits weak learners, often shallow trees, on reweighted samples to emphasize hard examples. AdaBoost performs well with clean data but is sensitive to label noise.

GBM (Friedman 2001) Gradient Boosting Machines build additive models by fitting learners to residuals. They are expressive and support arbitrary differentiable loss functions, but are prone to overfitting without careful tuning.

LGBM (Ke et al. 2017) LightGBM improves GBM efficiency via histogram binning, leaf-wise tree growth, and optimized data structures. It supports categorical features natively and achieves state-of-the-art results on large tabular datasets.

XGBoost (Chen and Guestrin 2016) An efficient and scalable GBM variant that includes regularization, sparsity-aware split finding, and weighted quantile sketching. It has become a dominant baseline for structured data problems.

CatBoost (Prokhorenkova et al. 2018) Designed for datasets with categorical variables, CatBoost uses ordered boosting and target permutation to reduce overfitting and leakage. It consistently outperforms other gradient boosting methods in categorical-heavy settings.

TabNet (Arik and Pfister 2021) Combines attentive feature selection and sequential decision steps using sparse masks. TabNet processes input in multiple steps, attending to different features at each, and is inherently interpretable through learned masks.

TabTransformer (Huang et al. 2020) Applies transformer blocks to tokenized categorical variables alongside continuous inputs. It captures contextual dependencies among features using multi-head self-attention and has shown strong performance when large-scale pretraining is used.

FT-Transformer (Gorishniy et al. 2021) A simplified version of TabTransformer using Fourier features and numerical embeddings, offering a lightweight yet expressive model for mixed-type tabular data.

TabSeq (Habib et al. 2024) Learns an optimal ordering of features and models them sequentially using transformer or recurrent backbones. By treating features as tokens, TabSeq aligns with the inductive biases of language models and benefits from position-aware processing.

TANGOS (Jeffares et al. 2023) Regularizes neural networks by enforcing orthogonality and specialization in gradient directions. It encourages disentangled representations across layers and improves robustness to noisy features.

TabPFN (Hollmann et al. 2023) A pretrained transformer trained to solve tabular classification tasks in a single forward pass. It uses meta-learning across millions of synthetic tasks, providing zero-shot generalization without task-specific fine-tuning.

NODE (Popov, Morozov, and Babenko 2020) Neural Oblivious Decision Ensembles replace classical decision trees with differentiable counterparts. Each ensemble member applies fixed feature splits and learned leaf weights, enabling end-to-end training and strong inductive bias.

SAINT (Somepalli et al. 2022) Enhances tabular transformers with contrastive pretraining, row-wise attention, and augmentation strategies. It improves generalization, particularly in low-label or imbalanced settings.

DeepFM (Guo et al. 2017) Merges factorization machines and MLPs to model both low- and high-order feature interactions. It is popular in recommender systems and tabular CTR prediction.

DCN (Wang et al. 2017) The Deep & Cross Network explicitly models feature crosses alongside deep representations. Its residual stacking of cross layers allows learning both memorization and generalization patterns.

AutoInt (Song et al. 2019) Uses self-attention to learn feature interactions of arbitrary order, replacing manual feature engineering with neural interaction modeling.

TabPFN v2 (Hollmann et al. 2025) An improved version of TabPFN with better calibration, extended to uncertainty-aware prediction and out-of-distribution detection.

TabR (Gorishniy et al. 2024) Incorporates nearest-neighbor retrieval into tabular deep learning, enabling semi-parametric prediction with robustness to rare feature patterns.

ProtoGate (Jiang et al. 2024) Combines prototype learning with sparse feature selection. It assigns samples to learned prototypes while gating features both globally and locally, providing a balance between accuracy and interpretability.

LSPIN (Yang, Lindenbaum, and Kluger 2022) Promotes locally sparse activations using adaptive gating and structural priors. It is designed for biomedical data with redundant or irrelevant features.

LLSPIN (Yang, Lindenbaum, and Kluger 2022) A hierarchical extension of LSPIN that enforces layered sparsity across neural representations, supporting multi-resolution abstraction.

INVASE (Yoon, Jordon, and Van der Schaar 2018) Employs a policy-based reinforcement learning framework to perform instance-wise feature selection. It learns to select the minimal feature subset needed for accurate predictions.

L2X (Chen et al. 2018) Maximizes the mutual information between selected features and output predictions. It enables post-hoc explanations via variational approximations of the information bottleneck objective.

Mambular (Thielmann et al. 2024) Adopts Mamba blocks to model feature sequences with state-space layers. It efficiently captures long-range dependencies and outperforms transformers in sequence modeling benchmarks.

DANets (Chen et al. 2022) Introduces a family of abstract residual networks tailored for tabular tasks. It uses deep skip connections and normalized embeddings for categorical variables.

STG (Yamada et al. 2020) Uses a differentiable relaxation of Bernoulli masks to perform sparse feature selection. The stochastic gates are optimized jointly with the predictive loss.

REAL-X (Jethani et al. 2021) Trains an auxiliary explanation model to mirror the predictive model's output. It offers faithful and consistent explanations through a reconstruction objective.

TabM (Gorishniy, Kotelnikov, and Babenko 2025) Combines multiple parameter-efficient heads with shared encoders. It reduces variance through implicit ensembling while preserving training efficiency.

ModernNCA (Ye et al. 2025) Revisits neighborhood components analysis using deep networks and contrastive objectives, showing competitive results in tabular domains.

Trompt (Chen et al. 2023) Combines permutation-invariant encoders with supervised contrastive loss. It addresses tabular data irregularities through modular architectural design.

TabulaRNN (Thielmann and Samiee 2024) Applies recurrent neural networks across feature dimensions using dynamic positional encoding. It learns to exploit sequential dependencies among features.

MambAttention (Thielmann and Samiee 2024) Merges Mamba-style state-space layers with global self-attention, enabling both local and global context modeling.

MambaTab (Ahamed and Cheng 2024) A plug-and-play tabular model using selective state-space mechanisms inspired by Mamba. It provides linear complexity while capturing long-range dependencies.

NDTF (Kontschieder et al. 2015) Neural Decision Tree Forests integrate soft decision rules into end-to-end differentiable ensembles. They approximate classical trees while being fully trainable.

ENODE (OpenTabular Contributors 2025) An extended version of NODE with residual connections and deeper ensemble structures. It improves expressivity and robustness.

ResNetTabular (OpenTabular Contributors 2025) Adapts residual networks to tabular input by stacking fully connected layers with skip connections. It helps stabilize deeper networks and accelerates convergence.

CategoryEmbedding (Joseph 2021) Learns task-specific embeddings for categorical variables, often combined with transformers or residual architectures to improve semantic learning.

Model	Hyperparameters (key = value)
Lasso	`model__C` = 0.19318275305768362 (from scikit-learn pipeline; note: key name indicates a linear model with C regularization).
Random Forest	`model__n_estimators` = 143; `model__max_depth` = 14.
LightGBM (LGBM)	`model__n_estimators` = 141; `model__learning_rate` = 0.14610823896610248.
CatBoost	`model__iterations` = 85; `model__learning_rate` = 0.2961190633942953.
MLP (scikit-learn)	Pipeline: `StandardScaler` → `MLPClassifier`. `hidden_layer_sizes` = (100,), `activation` = relu, `solver` = adam, `alpha` = 1e-4, `max_iter` = 1000, `random_state` = 42.
LLSPIN	`pred_hidden_dims` = [200, 200], `gate_hidden_dims` = [100, 100], `a` = 1.0, `sigma` = 0.5, λ (`lam`) = 0.1, $\gamma_1 = 0.1$, $\gamma_2 = 0.1$, `activation_pred` = relu, `activation_gate` = relu, `use_batchnorm` = True, `optimizer` = Adam(`lr`=1e-3), `epochs` = 20, `batch_size` = 32 (train), 64 (val).
ProtoGate	`lr` = 1e-3, `feature_selection` = True, `protogate_a` = 0.2, `protogate_sigma` = 0.5, `protogate_lam_local` = 0.1, `protogate_lam_global` = 0.01, `protogate_activation_gating` = relu, `protogate_gating_hidden_layer_list` = [100, 100], `protogate_init_std` = 0.02, `pred_k` = 5, `pred_coef` = 1.0, `sorting_tau` = 1.0, `optimizer` = Adam, `max_epochs` = 100, `batch_size` = $\lvert X_{\text{train}}\rvert$ (full batch).
STG	`task_type` = classification, `hidden_dims` = [100, 100], `activation` = relu, `optimizer` = Adam, `learning_rate` = 1e-3, `batch_size` = $\lvert X_{\text{train}}\rvert$ (full batch), `feature_selection` = True, `sigma` = 0.5, `lam` = 0.1, `nr_epochs` = 100, `device` = cpu.
TabulaRNN	Training: `max_epochs` = 50, `lr` = 1e-3, `patience` = 10; Architecture: defaults from `DefaultTabulaRNNConfig` (no overrides provided in the snippet).

Table C.1: Arcene dataset: selected model hyperparameters after Optuna tuning.

Selected Baseline Hyperparameters. Table C.1 compiles the concrete settings used for Arcene dataset, pairing Optuna-selected values with the exact training configurations extracted from the provided code for some of the models. Among the classical/GBDT baselines, Lasso favored a relatively strong regularizer (C=0.193), Random Forest settled on a moderately deep ensemble (143 trees, max depth 14), LightGBM chose 141 estimators with a comparatively aggressive learning rate (0.1461), and CatBoost used 85 iterations with an even higher rate (0.2961), consistent with short, fast schedules on HDLSS data. Neural and selection-style baselines follow other pattern: the MLP pipeline standardizes inputs and trains a single-hidden-layer ReLU MLP (100 units, Adam, α=10^{-4}, `max_iter` =1000, seed 42); LLSPIN uses gating $[100, 100]$ and predictor $[200, 200]$ MLPs with ReLU, a=1.0, σ=0.5, λ=0.1, γ_1=γ_2=0.1, Adam (10^{-3}), 20 epochs; ProtoGate enables feature selection with ReLU gating $[100, 100]$, a=0.2, σ=0.5, λ_{local}=0.1, λ_{global}=0.01, k=5, τ=1, Adam (10^{-3}), 100 epochs, full batch; STG is trained as a classifier with ReLU $[100, 100]$, Adam (10^{-3}), σ=0.5, λ=0.1, 100 epochs, full batch on CPU; and TabulaRNN relies on its default architecture with a 50-epoch schedule (lr 10^{-3}, patience 10). Together, these settings document the exact capacity, regularization, and optimization choices underpinning the Arcene experiments.

DynaTab Hyperparameters

To illustrate the diversity of optimal configurations across data regimes, we report Optuna-tuned hyperparameters for DynaTab on five representative datasets spanning each data category. The Optuna-tuned hyperparameters across five representative datasets (Tables A.2-A.6) reveal how DynaTab adapts its architectural and algorithmic configuration based on dataset characteristics, particularly feature dimensionality, sample size, and the presence of noise or redundancy. For AI-D (Case 5), a MixedRegime dataset, the selected configuration emphasizes model expressivity with a larger embedding dimension ($d_{model} = 256$) and moderate sparsity controls ($\lambda_{\text{disp}} = 0.4$, $\lambda_{\text{global}} = 0.3$). The use of 12 clusters and a *descending* order suggests that the model benefits from attending to more informative features early in the sequence. The Transformer backbone, with 4 attention heads and a window size of 64, allows sufficient capacity for localized feature interactions. In contrast, the Pima Indian dataset from the LDLSS category (few features and low sample size) favors minimalistic configuration: only 2 clusters, no feature mutation, and no sparsity regularization. This aligns with its low complexity; the model avoids overfitting by using a shallow Transformer (2 layers, $d_{model} = 128$) and lower attention granularity (window size 32). The ascending feature order indicates that gradually introducing features improves stability when information density is uniformly distributed. The Arcene dataset, a classic HDLSS benchmark (high-dimensional, low-sample size), requires aggressive feature compression and selective attention. DynaTab uses 20 clusters with the *Mamba* backbone, favoring state-space modeling over full attention. Higher mutation probability (0.2) and mild sparsity ($\lambda_{\text{disp}} = \lambda_{\text{global}} = 0.1$) are beneficial here, promoting adaptive pruning of noisy dimensions. The learning rate is reduced to 0.0001 to stabilize training under high variance. For the MiniBooNE dataset (LDHSS), which has moderate feature size but a large sample count, the optimal configuration resembles that of Arcene but uses the Transformer backbone. While the feature space is not as high-dimensional, the large sample size allows deeper exploration of feature relevance. The 5 clusters

Model	Source URL	Model	Source URL
Naive Bayes	https://scikit-learn.org/stable/supervised_learning.html	AutoInt	https://github.com/OpenTabular/DeepTab
KNN	https://scikit-learn.org/stable/supervised_learning.html	TabPFN v2	https://github.com/PriorLabs/tabpfn-extensions
SVM	https://scikit-learn.org/stable/supervised_learning.html	TabR	https://github.com/OpenTabular/DeepTab
Decision Tree	https://scikit-learn.org/stable/supervised_learning.html	ProtoGate	https://github.com/SilenceX12138/ProtoGate
Lasso	https://scikit-learn.org/stable/supervised_learning.html	LSPIN	https://github.com/jcyang34/lspin
MLP	https://scikit-learn.org/stable/supervised_learning.html	LLSPIN	https://github.com/jcyang34/lspin
1-D CNN	https://github.com/harryjdavies/Python1D_CNNs	INVASE	https://github.com/vanderschaarlab/INVASE
Random Forest	https://scikit-learn.org/stable/supervised_learning.html	L2X	https://github.com/Jianbo-Lab/L2X
AdaBoost	https://scikit-learn.org/stable/supervised_learning.html	Mambular	https://github.com/OpenTabular/DeepTab
GBM	https://scikit-learn.org/stable/supervised_learning.html	DANets	https://github.com/manujosephv/pytorch_tabular
LGBM	https://github.com/microsoft/LightGBM	STG	https://github.com/runopti/stg
XGBoost	https://github.com/dmlc/xgboost	REAL-X	https://github.com/rajesh-lab/realx
CatBoost	https://github.com/catboost/catboost	TabM	https://github.com/OpenTabular/DeepTab
TabNet	https://github.com/dreamquark-ai/tabnet	ModernNCA	https://github.com/OpenTabular/DeepTab
TabTransformer	https://github.com/lucidrains/tab-transformer-pytorch	Trompt	https://github.com/OpenTabular/DeepTab
FT-Transformer	https://github.com/lucidrains/tab-transformer-pytorch	TabulaRNN	https://github.com/OpenTabular/DeepTab
TabSeq	https://github.com/zadid6pretam/TabSeq	MambAttention	https://github.com/OpenTabular/DeepTab
TANGOS	https://github.com/OpenTabular/DeepTabular	MambaTab	https://github.com/OpenTabular/DeepTab
TabPFN	https://github.com/PriorLabs/TabPFN	NDTF	https://github.com/OpenTabular/DeepTab
NODE	https://github.com/OpenTabular/DeepTab	ENODE	https://github.com/OpenTabular/DeepTab
SAINT	https://github.com/OpenTabular/DeepTab	ResNetTabular	https://github.com/OpenTabular/DeepTab
DeepFM	https://github.com/shenweichen/DeepCTR-Torch	CategoryEmbedding	https://github.com/manujosephv/pytorch_tabular
DCN	https://github.com/shenweichen/DeepCTR-Torch		

Table C.2: List of models and their source URLs.

and use of descending ordering again suggest the presence of strongly informative features that should be prioritized early. Finally, the HAM10000 dataset, representing the HDHSS regime, combines both high dimensionality and high sample count. DynaTab adopts a robust configuration with a Mamba backbone, larger batch size, accumulation steps, and gradient clipping all techniques that stabilize training on large-scale datasets. Like Arcene, 20 clusters are used, but the optimizer reduces the learning rate further to 5×10^{-5}, reflecting the need for precise, low-variance updates on large and expressive models. Overall, these results demonstrate that DynaTab's design space is both flexible and responsive: it adapts its architecture (Transformer vs. Mamba), attention structure (number of clusters, window size), and regularization strategy (mutation, λ values) according to the statistical regime of the data. This adaptability is key to its competitive performance across diverse tabular benchmarks.

Discussion on HDLSS Benchmark Results

Table E.1 summarize the performance of 45 models across eight benchmark HDLSS datasets. DynaTab achieves the best average rank (2.62 ± 2.26), outperforming both classical statistical methods and recent deep learning architectures, which underscores its robustness in high-dimensional, low-sample-size settings. Among traditional baselines, Lasso (5.12 ± 4.29) and KNN (15.88 ± 8.63) performed reasonably well, demonstrating that sparsity-inducing and instance-based learners still hold merit under extreme sample constraints. Tree-based ensembles e.g. Random Forest (11.38 ± 5.73), AdaBoost (16.12 ± 8.95), LGBM (9.00 ± 6.19), and CatBoost (10.62 ± 6.05) also maintained competitive accuracy, though with greater variability across datasets. Among neural-network methods, ProtoGate (5.50 ± 3.74), MLP (5.88 ± 4.22), and TabulaRNN (8.00 ± 5.86) emerged as strong contenders; notably, ProtoGate's prototype-based regularization appears particularly beneficial in the low-sample regime. In contrast, attention-heavy transformers such as SAINT (27.50 ± 6.02), FT-Transformer (27.62 ± 7.82), and TabTransformer (38.75 ± 5.28) underperformed significantly likely owing to overparameterization and their reliance on large datasets to learn stable representations. Tabular-specific architectures (NODE: 31.75 ± 5.52, TabNet: 32.38 ± 6.07, AutoInt: 27.75 ± 5.50) and feature-selection methods (L2X: 32.75 ± 7.74, INVASE: 19.38 ± 7.67) also showed inconsistent results, indicating limited scalability to HDLSS contexts. Sequence-inspired Mamba variants e.g. MambaTab (21.12 ± 9.28), Mambular (23.12 ± 12.70), and MambAttention (24.00 ± 8.96) yielded mixed performance, revealing the challenge of applying temporal-style processing to static tabular data. More recent proposals such as TabSeq (22.75 ± 11.00), Trompt (34.12 ± 5.41), and ModernNCA (38.50 ± 8.09) did not surpass simpler baselines, emphasizing the importance of architectural frugality and inductive bias over sheer model complexity in the HDLSS regime. Overall, DynaTab's superior average rank and low standard deviation highlight its consistent generalization across diverse HDLSS benchmarks. Its differentiable feature-ordering mechanism and dynamic fusion module effectively adapt to extreme data scarcity, validating our design choices for state-of-the-art performance under high-dimensional, low-sample constraints.

Discussion on HDHSS Benchmarks Results

Table F.1 presents the performance of 22 models across six high-dimensional, high-sample-size (HDHSS) datasets, including hybrid tabular-image benchmarks where image embeddings are combined with structured metadata. The results reveal several key trends in model performance, highlighting the strengths of recent methods and limitations of classical baselines in handling complex, high-dimensional data regimes. DynaTab achieves the highest average rank (2.50 ± 2.35), outperforming all 21 baselines across nearly all datasets. Its consistently high accuracy from 84.48% on HAM10000 to 99.20% on Dog vs Cat demonstrates robust generalization in diverse HDHSS conditions. Particularly notable is its ability to outperform both deep learning and gradient-boosting methods, suggesting that the model's dynamic feature selection and order-aware processing effectively leverage the high-dimensional structure while maintaining efficiency. Among the top contenders, LSPIN ranks second (4.67 ± 2.80), performing well across all datasets, especially on CIFAR-10 and Dog vs Cat+#. It slightly outperforms DynaTab on MNIST and matches performance on the binary Dog vs Cat+# task, but lags in datasets like HAM10000. MLP (5.67 ± 3.01) also performs competitively, likely due to its scalability and expressiveness, though it exhibits higher variance, particularly on DeepLesion. LGBM and TabNet are ranked fourth and fifth, respectively. While LGBM achieves strong performance on DeepLesion (94.73%) and maintains good consistency, it is surpassed by DynaTab and LSPIN on almost all datasets. TabNet performs well on image-rich datasets like MNIST+# and Fashion-MNIST, where its attentive architecture helps, but suffers from instability on HAM10000 (73.51%). Models like TabR, GBM, and XGBoost cluster in the mid-tier rankings, all achieving solid but not state-of-the-art performance. This suggests that while ensemble-based methods still offer competitive baselines, they struggle to match newer methods that adapt dynamically to feature distributions. LLSPIN (rank: 7.33) underperforms relative to its full-rank counterpart LSPIN, highlighting the limitations of low-rank approximations in very high-dimensional regimes. Similarly, STG and Lasso, both sparse-feature selection methods, demonstrate stable but non-optimal performance, indicating that static sparsity mechanisms may be insufficient in HDHSS contexts with mixed feature modalities. Conventional models such as KNN, SVM, and Random Forest rank lower despite reasonable performance on individual datasets. For instance, SVM achieves 99.00% on Dog vs Cat, but this does not generalize well across datasets with higher feature dimensionality and noise. Tree-based methods like Decision Tree and AdaBoost struggle significantly, with Decision Tree scoring just 68.24% on HAM10000 and a sharp drop on MNIST (74.86%), highlighting their sensitivity to high-dimensional noise and lack of feature interaction modeling. Naive Bayes and CategoryEmbedding rank at the bottom (19.67), failing to handle complex input distributions and heterogeneous feature spaces. Naive Bayes performs especially poorly on DeepLesion (18.55%), suggesting its independence assumption breaks under correlated, embedded inputs. Similarly, CategoryEmbedding shows weak generalization, particularly on CIFAR-10 and DeepLesion. While models like TabNet, TabM, TabR, and TabSeq span a wide range of ranks, only DynaTab shows consistent improvements across all HDHSS datasets. TabSeq, for instance, performs relatively well on MNIST and Dog vs Cat, but fails on DeepLesion+# (78.82%) and CIFAR-10 (85.42%), yielding an average rank of 18.33. This underscores the difficulty of building generalized tabular deep learning architectures that scale to high-dimensional settings without task-specific tuning. These results highlight the importance of dynamic, adaptive mechanisms such as DynaTab's order-aware fusion and feature gating in modeling HDHSS data. While traditional models and even modern tabular deep learning methods exhibit strong dataset-specific performance, they often fail to generalize across different modalities and dimensionality regimes. DynaTab's consistently top-tier performance validates the efficacy of neural rewiring and selective attention in high-dimensional learning.

Discussion on Mixed Regime Benchmarks

Table G.1 summarizes the performance of 46 models across eight Mixed Regime (MR) datasets, which feature varying levels of dimensionality, modality, and sample size. These benchmarks are particularly challenging due to their heterogeneity, making them a rigorous test for model generalization. The results demonstrate significant performance variance across models, highlighting the importance of adaptable architectures in mixed-regime settings. DynaTab obtains the best overall rank (6.00 ± 8.72), consistently delivering strong performance across diverse datasets. It ranks first on EEG-FE (98.10 ± 0.56) and WDBC (98.14 ± 2.05), and performs competitively on all others, including noisy low-signal datasets like Water Potability (62.56 ± 2.31). This robustness reflects the strength of DynaTab's dynamic processing mechanisms, which effectively reconcile feature heterogeneity and task complexity without manual tuning. LSPIN ranks second overall (8.50 ± 4.17), maintaining high accuracy across all datasets, particularly on AI-D (85.10 ± 4.30), EEG-FE (97.42 ± 0.32), and WDBC (97.33 ± 2.52). However, it lags slightly behind DynaTab on tasks involving fine-grained inter-modality fusion, such as CNAE-9 and MOF. TabTransformer (10.38 ± 10.42) also performs well, especially on ADNI and CNAE-9, but suffers from higher variance across tasks, indicating sensitivity to domain shifts. STG, REAL-X, and MLP form a strong cluster of sparse or fully-connected models with competitive performance. For instance, MLP scores 97.46 ± 0.58 on EEG-FE and 96.84 ± 1.42 on WDBC, confirming that deep but simple architectures can remain competitive in mixed regimes when sufficiently tuned. REAL-X and STG similarly offer strong results on MOF, EEG-FE, and ADNI, although their feature selection heuristics may lead to suboptimal generalization in lower-signal datasets like Water Potability. TabPFN v2 (13.38 ± 19.59) delivers outstanding accuracy on the six datasets it was evaluated on peaking at 97.68 ± 1.78 on WDBC and 94.81 ± 1.26 on CNAE-9 but was not run on EEG-FE or EEG-PD due to runtime or memory limitations. The original TabPFN was only evaluated on WDBC (95.79 ± 1.93) and, by default, was assigned the worst rank on all other datasets, resulting in the lowest overall average rank (45.00 ± 0.00). Wherever '—' appears

in Table G.1, it reflects that no prediction was made rather than a failed outcome. LLSPIN performs consistently well across all datasets, though it scores slightly lower on Water Potability (59.63 ± 2.04), which impacts its average rank (14.13 ± 9.09). Ensemble methods such as CatBoost, Random Forest, and LGBM continue to be strong contenders in MR benchmarks. CatBoost achieves 96.14 ± 1.63 on WDBC and 88.15 ± 0.57 on EEG-PD, but its performance on CNAE-9 is comparatively weaker. Random Forest and LGBM also score highly on EEG-FE and EEG-PD but fall short on tasks with noisy or high-cardinality features. SVM and TabulaRNN perform similarly, with SVM slightly outperforming on ADNI and EEG-PD. Both models show reduced variance and strong single-dataset scores but fail to generalize uniformly, resulting in mid-tier rankings. Other notable methods like FT-Transformer and TabM show reliable results on EEG and clinical datasets but experience moderate degradation on MOF and CNAE-9, suggesting that fixed-token attention or memory-based mechanisms are insufficient to fully adapt to cross-domain variation. Methods such as TabSeq, MambaTab, and SAINT offer stable mid-level performance. TabSeq and MambaTab score well on EEG-related tasks, while SAINT performs decently across all benchmarks but lags on Water Potability. These outcomes suggest a degree of architectural strength, but also sensitivity to dataset-specific biases. Toward the bottom, a group of models including ModernNCA, TabNet, and INVASE struggle to maintain stability. TabNet, for example, performs strongly on EEG-FE (93.22 ± 0.88), but falls sharply to 57.06 ± 3.31 on Water Potability. ModernNCA and INVASE show similar inconsistencies, indicating difficulty generalizing in the presence of noise or multi-modal features. The weakest-performing models include DeepFM, DCN, ProtoGate, and TabPFN (original). ProtoGate in particular underperforms across all reported datasets, scoring 64.50 ± 6.40 on AI-D and only 52.73 ± 2.51 on Water Potability. These results suggest that fixed, low-capacity, or overly constrained models are ill-equipped to handle the heterogeneity of MR benchmarks. In summary, DynaTab again demonstrates robust and consistent superiority in this mixed regime setting. Its ability to adapt dynamically to diverse dataset structures and modality combinations makes it especially well-suited for real-world applications where prior assumptions about feature distributions and task structure may not hold. The broad variation in model performance further emphasizes the need for architectures that are both expressive and adaptable across domains.

Discussion on LDHSS Benchmarks Results

Table H.1 reports the performance of models across five low-dimensional, high-sample-size (LDHSS) datasets. These datasets emphasize large sample sizes and relatively moderate input dimensionality, placing importance on training stability, generalization under data imbalance, and computational efficiency. Tree-based ensemble methods dominate the top ranks. CatBoost achieves the highest average rank (1.60 ± 0.55), consistently outperforming other models across all datasets. It scores particularly well on MiniBooNE (94.19 ± 0.29), Adult (87.32 ± 0.37), and Poker Hand (80.53 ± 0.91), reflecting its strong inductive bias and robustness to tabular structure. XGBoost follows closely with strong performance across the board, achieving near-identical results to CatBoost on MiniBooNE and Adult, although slightly weaker on Poker Hand (71.98 ± 0.72). Random Forest and LGBM also rank among the top five, with LGBM outperforming on Higgs (72.45 ± 0.33), while Random Forest leads on Covertype (89.29 ± 0.17). However, Random Forest underperforms on Poker Hand (60.23 ± 0.44), which impacts its average rank. MLP is the top deep learning-based model in this regime, showing consistent performance across all datasets with no catastrophic failures, scoring as high as 93.34 ± 0.32 on MiniBooNE. GBM and TabR also demonstrate strong overall utility, with TabR achieving the highest score on Adult (87.50 ± 0.45), though its high standard deviation on Covertype suggests less stability in larger feature spaces. TabSeq and TabTransformer, representing early-stage and transformer-based tabular models respectively, achieve mid-range ranks. While TabSeq performs reliably on Adult and Higgs, it struggles on Covertype and Poker Hand. TabTransformer suffers from relatively high variance on MiniBooNE (86.48 ± 3.42), which penalizes its rank. SVM achieves competitive scores on Adult and MiniBooNE but is limited by relatively lower accuracy on Covertype and Poker Hand. DynaTab performs solidly on MiniBooNE (91.23 ± 0.80) and Higgs (71.93 ± 0.35), but underperforms on Covertype (68.80 ± 2.12) and Poker Hand (52.91 ± 3.70), resulting in a mid-tier average rank (11.40 ± 5.90). This reflects a trade-off between expressiveness and training dynamics when scaled to very large datasets with shallow feature representations. TabNet, CategoryEmbedding, and TabM also fall into the mid-tier group. TabNet exhibits stable but non-leading results, while CategoryEmbedding and TabM struggle on datasets with complex distributions, such as Covertype and Poker Hand. KNN performs surprisingly well on Covertype (85.15 ± 0.25), but shows a steep drop on Higgs (62.02 ± 0.23), which lowers its overall rank. Transformer-based and neural feature selection models like FT-Transformer, SAINT, AutoInt, and Mambular show mixed results. FT-Transformer performs decently on MiniBooNE and Adult but lags on Higgs. SAINT and AutoInt display inconsistent behavior, with relatively poor generalization on Poker Hand and Covertype. Mambular and 1-D CNN models are even less competitive, with both methods showing significant drop-offs across most tasks. ModernNCA, DCN, LLSPIN, and MambaTab struggle with the scale and shallowness of LDHSS data, reflecting their architectural biases toward high-dimensional feature spaces. DeepFM, TANGOS, DANets, and NDTF exhibit similarly poor performance, often failing to surpass simpler baselines. Notably, models designed for feature selection such as REAL-X, L2X, and INVASE perform poorly in this setting, as their complexity does not translate well to LDHSS distributions. The bottom ranks are dominated by TabulaRNN, ProtoGate, and various deep hybrid models. Naive Bayes and Decision Tree perform poorly on datasets with high cardinality or class imbalance, such as Covertype and Poker Hand. ProtoGate and INVASE rank lowest, indicating their limited applicability to large-sample, shallow-feature tabular domains. Overall, the LDHSS benchmarks strongly favor traditional gradient-boosted ensembles and robust shallow models. While deep learning models like DynaTab and MLP show competitive behavior on some datasets, their rank is hindered by underperformance on datasets that require highly optimized memory or input encoding strategies. These

results reaffirm the dominance of ensemble models in low-dimensional settings, while highlighting the importance of future work on scaling deep learning methods to such regimes without sacrificing stability or sample efficiency.

Discussion on LDLSS Benchmarks Results

Table I.1 presents the performance of 46 models across six low-dimensional, low-sample-size (LDLSS) datasets, a regime where statistical robustness, overfitting control, and generalization from sparse supervision are crucial. The results exhibit large variance across models, reflecting the difficulty of learning meaningful representations from small datasets with limited feature complexity. TabPFN v2 ranks highest overall (4.50 ± 5.09), achieving perfect or near-perfect scores on Monks-1 and Liver Disorder, and competitive performance on all other datasets. Its pretrained zero-shot prior enables strong few-shot generalization without the need for extensive tuning. CatBoost also performs exceptionally well (5.33 ± 6.41), nearly matching TabPFN v2 across all tasks, particularly on Iris (95.33 ± 3.39), Glass (81.32 ± 4.62), and the two datasets with perfect performance. Tree ensembles such as GBM and Random Forest maintain strong average rankings. GBM achieves 99.31 ± 1.38 on Monks-1 and 100.00 ± 0.00 on Liver Disorder, while Random Forest performs well across the board, particularly on Hayes-Roth and Glass. However, both models are susceptible to performance fluctuations on very small datasets like Pima Indian and Hayes-Roth. TabulaRNN, AutoInt, FT-Transformer, NODE, and TabR form a strong group of neural and attention-based models adapted to tabular structure. Despite moderate variance, these models show competitive performance on most datasets. TabulaRNN and AutoInt both achieve 100.00 ± 0.00 on Monks-1 and Liver Disorder, while maintaining reasonable generalization elsewhere. FT-Transformer exhibits relatively high variance, especially on Hayes-Roth (77.27 ± 13.51), but still ranks in the top third. DynaTab ranks mid-tier overall (14.17 ± 12.92), performing solidly on Hayes-Roth (84.29 ± 5.35) and Liver Disorder, but slightly underperforming on Iris and Pima Indian. While its architecture is optimized for high-dimensional and mixed-regime inputs, its inductive biases are less tailored to small tabular datasets, where simpler models or pretrained solutions excel. XGBoost, TabPFN (original), and Mambular also show strong individual results but slightly lower average ranks due to inconsistencies on certain datasets. TabPFN performs well on all but Monks-1, where it slightly lags behind v2. Mambular, despite achieving perfect scores on two datasets, underperforms on Hayes-Roth (58.26 ± 13.87), reducing its overall standing. Methods like MambAttention, LSPIN, ResNetTabular, Decision Tree, and LLSPIN show stable but non-leading performance. Many of these models are not explicitly optimized for low-sample training and may overfit or underfit, especially on highly noisy or class-imbalanced datasets. TabM, INVASE, and SVM display a similar trend strong accuracy on specific datasets but large drop-offs elsewhere. Mid- to lower-ranked models, including CategoryEmbedding, TabSeq, SAINT, 1-D CNN, MLP, and MambaTab, tend to suffer from higher variance across tasks. While most achieve high scores on Monks-1 and Liver Disorder, they often struggle on Pima Indian or Glass, which contain more challenging class structures and fewer informative features. Models such as ModernNCA, TabNet, Lasso, TabTransformer, and STG demonstrate modest performance, with moderate to high instability across datasets. These models often rely on larger training sets or complex feature interactions that are difficult to capture in LDLSS scenarios. The weakest-performing models include DeepFM, DANets, NDTF, DCN, KNN, ENODE, Naive Bayes, TANGOS, L2X, and ProtoGate. ProtoGate performs especially poorly (40.50 ± 7.71), with accuracy as low as 31.30 ± 6.40 on Glass and 65.51 ± 4.41 on Liver Disorder, suggesting that complex selection mechanisms or constrained model capacity hinder generalization in this regime. In summary, the LDLSS setting clearly favors pretrained models like TabPFN v2 and robust tree ensembles such as CatBoost and GBM. Simpler architectures and models with few parameters tend to generalize better than deep neural networks when data is scarce. While DynaTab remains competitive, its design favors more complex data scenarios. These results highlight the unique challenges of the LDLSS regime and the need for models that can generalize reliably from limited supervision.

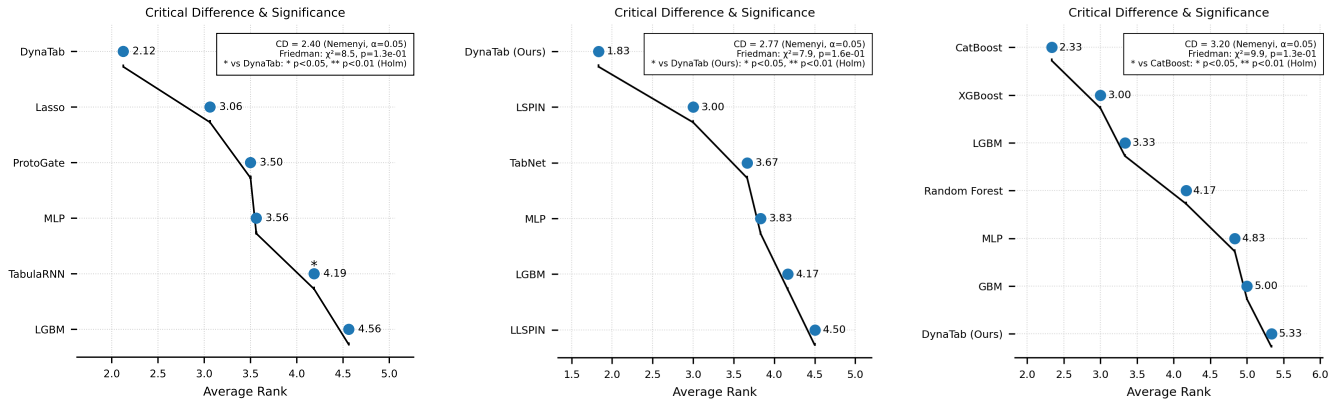
Extended Analysis for Statistical Significance

Statistical Significance for HDLSS

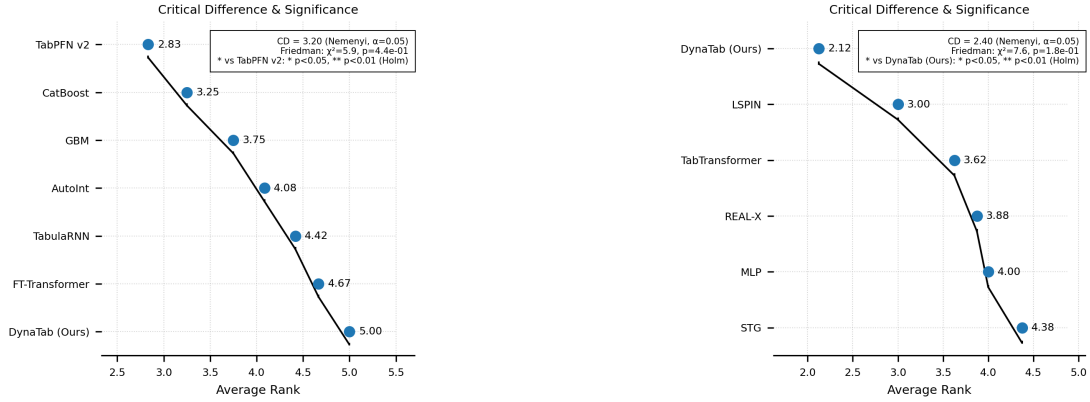
We evaluated top six models on HDLSS datasets using average rank analysis (Fif. J.1a). DynaTab achieved the best overall performance with the lowest average rank (2.12), followed by Lasso (3.06), ProtoGate (3.50), and MLP (3.56). TabulaRNN (4.19) and LGBM (4.56) performed worse in comparison. A critical difference (CD) of 2.40 was computed using the Nemenyi test at $\alpha = 0.05$. Models connected by horizontal lines fall within this CD and thus do not differ significantly. Pairwise comparisons with DynaTab, corrected using the Holm method, revealed that only TabulaRNN was significantly worse ($p < 0.05$), denoted by an asterisk. The overall Friedman test statistic ($\chi^2 = 8.5$, $p = 0.13$) was not significant, indicating no strong evidence to reject the null hypothesis of equal performance across models (see supplementary material for details on other dataset categories and figures).

Statistical Significance for LDHSS

We evaluated the top six models on LDHSS datasets using average rank analysis (Figure J.1c). DynaTab scored the lowest average rank (5.33), slightly trailing behind GBM (5.00) and MLP (4.83). CatBoost ranked first overall (2.33), followed by XGBoost (3.00) and LGBM (3.33). A critical difference (CD) of 3.20 was computed using the Nemenyi test at $\alpha = 0.05$. Several models, including DynaTab and GBM, fall within this CD range, indicating no statistically significant difference in performance.



(a) HDLSS (b) HDHSS (c) LDHSS



(d) LDLSS (e) Mixed Regime

Figure J.1: Critical difference diagrams for different dataset regimes (top-6 ranks). Non-significant ties (Nemenyi) and pairwise significance (Wilcoxon–Holm) are indicated.

Pairwise Holm-corrected Wilcoxon tests confirmed that none of the models were significantly worse than CatBoost at the 0.05 level. The Friedman test yielded $\chi^2 = 9.9$ with $p = 0.13$, indicating no statistically significant difference in overall rankings.

Statistical Significance for LDLSS

On LDLSS datasets (Figure J.1d), DynaTab obtained the highest average rank (5.00), while TabPFN v2 achieved the lowest (2.83), followed closely by CatBoost (3.25). GBM (3.75) and AutoInt (4.08) formed the mid-cluster. A critical difference (CD) of 3.20 was used for significance testing with $\alpha = 0.05$. According to the Nemenyi test, no model pairs exceeded this CD, indicating statistical similarity. Pairwise Wilcoxon tests against TabPFN v2, adjusted via the Holm method, revealed no significant differences at $p < 0.05$. The overall Friedman test was also non-significant ($\chi^2 = 5.9$, $p = 0.44$), reinforcing the lack of strong statistical separation among models.

Statistical Significance for Mixed Regime

For mixed-regime datasets (Figure J.1e), DynaTab achieved the best average rank (2.12), followed by LSPIN (3.00) and TabTransformer (3.62). STG, REAL-X, and MLP clustered near the bottom, with ranks ranging from 3.88 to 4.38. The Nemenyi test yielded a CD of 2.40 at $\alpha = 0.05$, under which only STG was significantly worse than DynaTab in pairwise comparison. Holm-corrected Wilcoxon tests confirmed statistical significance between DynaTab and STG ($p < 0.05$). The Friedman test statistic was $\chi^2 = 7.6$ with $p = 0.18$, indicating no overall significant difference, but notable relative advantages for DynaTab.

Statistical Significance for HDHSS

On HDHSS benchmarks (Figure J.1b), DynaTab again led with the lowest average rank (1.83), followed by LSPIN (3.00), TabNet (3.67), and MLP (3.83). LGBM and LLSPIN placed slightly lower with ranks above 4. A critical difference of 2.77 was calculated via the Nemenyi test ($\alpha = 0.05$). Several models fall outside this CD range compared to DynaTab. Pairwise

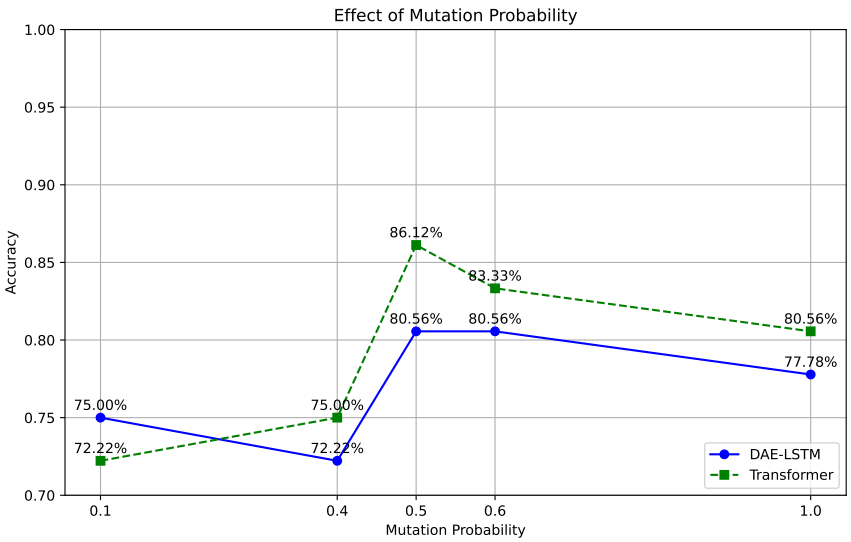


Figure K.1: Effect of mutation probability on performance.

Wilcoxon–Holm tests confirmed that LGBM and LLSPIN were significantly worse than DynaTab at $p < 0.05$. The Friedman test result ($\chi^2 = 7.9, p = 0.16$) indicates moderate but not statistically significant evidence of rank differences across models.

Extended Analysis on Ablation Studies

We conduct controlled ablation studies on DynaTab using the AI-D Case 5 dataset to assess the impact of core design components. Each setting is evaluated on both Transformer and DAE-MHA-LSTM backbones.

Mutation Probability. Figure K.1 illustrates that a mutation probability of 0.5 is optimal, yielding 86.12% for Transformer and 80.56% for DAE-MHA-LSTM. Lower probabilities (e.g., 0.1) under-explore feature permutations, while higher values (e.g., 1.0) destabilize learning.

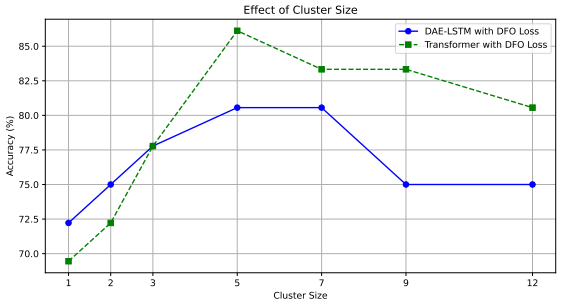
Cluster Size. As shown in Figure K.2a, a cluster size of 5 yields the highest accuracy across both backbones, balancing feature grouping granularity and model generalization. Performance drops for very small or large clusters, confirming the necessity of moderate grouping for efficient relational encoding.

Loss Functions. Figure K.2b shows that our proposed Dispersion Loss and DFO Loss outperform traditional objectives. For example, with the Transformer backbone, Dispersion Loss achieves 86.12% accuracy versus 80.56% with traditional loss. These custom objectives enhance ordered feature coherence and stabilize learning.

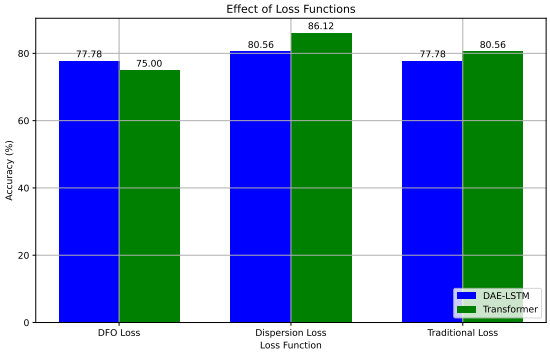
Edge Metrics without Rewiring. In Figure K.2c, KL divergence emerges as the most effective edge metric, reaching 86.12% accuracy on the Transformer backbone. Other metrics like Bhattacharyya and Total Variation achieve close but lower performance. Simpler metrics (e.g., Euclidean, Hellinger) underperform, indicating the benefit of information-theoretic divergence for capturing inter-feature relations.

Tolerance and Sorting Order. As seen in Figure K.2d, lower tolerance values (e.g., 0.03) consistently yield better accuracy across both orderings and backbones. Ascending order slightly outperforms descending in most settings. These results suggest that stricter dispersion thresholds help converge toward stable and informative feature orders.

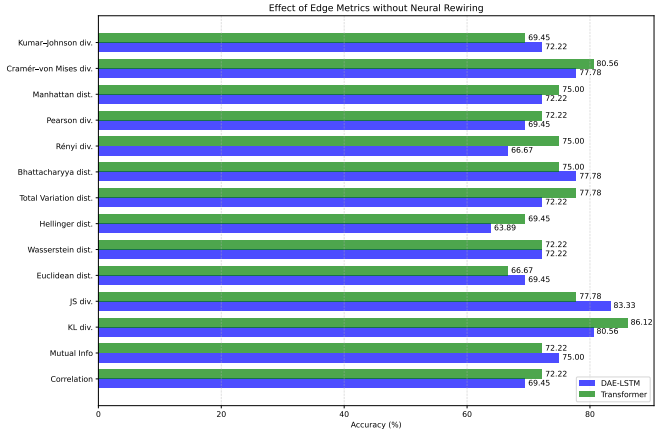
Computational Time for Dynamic Feature Ordering We benchmark the runtime of different GPU-enabled similarity metrics used in DynaTab's dynamic feature ordering module. Table B.2 reports the execution time (in seconds) for cluster sizes 7, 9, 12, and 15 on the AI-D Case-5 dataset (316×393 matrix). Traditional metrics such as variance and correlation are efficient, completing in under a second across all settings. In contrast, KL divergence, while the most effective (see Figure K.2c), is computationally expensive even with GPU support. KL divergence takes 107 seconds at cluster size 7 and grows to 229 seconds at size 15. On CPU, the same operation requires over 7 hours. This highlights the tradeoff between metric precision and computational cost in DynaTab's feature ordering.



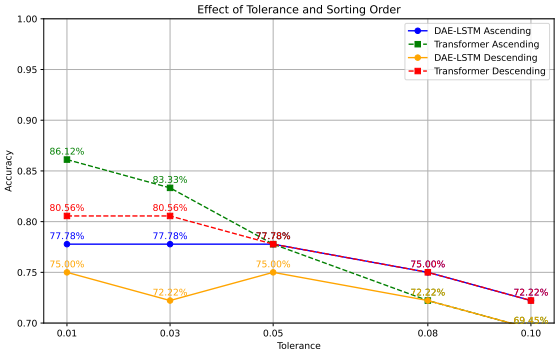
(a)



(b)



(c)



(d)

Figure K.2: Ablations: (a) effect of cluster size on accuracy (cluster size 5 is optimal); (b) comparison of loss functions (Dispersion and DFO outperform traditional losses); (c) accuracy under different edge metrics without neural rewiring (KL divergence is most effective); (d) impact of tolerance threshold and sorting order (lower tolerance and ascending order are more robust).

Model	GLI-85	SMK_CAN_187	ALLAML	Prostate-GE	Arcene	TOX-171	Colon	Lung	Rank (Avg ± SD)
DynaTab (Ours)	85.96 ± 5.77	61.31 ± 3.37	92.31 ± 5.77	90.91 ± 8.91	83.00 ± 6.71	88.71 ± 3.53	85.71 ± 8.91	92.75 ± 1.28	2.62 ± 2.26
Lasso	85.88 ± 4.71	61.19 ± 13.72	87.24 ± 3.39	91.18 ± 6.39	81.00 ± 3.39	91.86 ± 6.03	79.40 ± 10.18	94.47 ± 4.39	5.12 ± 4.29
ProtoGate	82.48 ± 5.68	60.16 ± 5.10	86.12 ± 3.34	90.58 ± 5.72	81.50 ± 5.10	92.34 ± 5.67	83.95 ± 9.82	93.44 ± 6.37	5.50 ± 3.74
MLP	85.41 ± 8.00	59.05 ± 7.44	89.98 ± 9.17	89.20 ± 6.07	78.40 ± 4.05	94.48 ± 4.28	83.95 ± 9.80	96.47 ± 2.69	5.88 ± 4.22
TabulaRNN	79.68 ± 6.68	60.02 ± 3.18	88.92 ± 2.02	90.50 ± 6.00	81.50 ± 5.10	85.80 ± 4.70	84.20 ± 6.50	90.50 ± 4.80	8.00 ± 5.86
LGBM	85.88 ± 11.53	58.85 ± 10.14	85.81 ± 5.67	91.38 ± 5.71	80.50 ± 5.79	81.98 ± 6.25	76.60 ± 11.67	93.42 ± 5.91	9.00 ± 6.19
CatBoost	84.71 ± 12.11	58.28 ± 12.16	91.71 ± 8.22	90.24 ± 6.87	81.00 ± 2.00	81.95 ± 7.47	72.65 ± 10.12	91.57 ± 5.74	10.62 ± 6.05
STG	82.48 ± 4.56	57.25 ± 8.82	86.08 ± 5.60	89.38 ± 5.85	74.40 ± 6.90	87.95 ± 5.01	79.55 ± 10.53	93.30 ± 6.28	11.25 ± 4.83
Random Forest	85.88 ± 8.80	58.29 ± 10.61	85.71 ± 5.71	90.38 ± 7.31	74.00 ± 1.22	79.78 ± 7.10	80.05 ± 10.37	91.73 ± 6.61	11.38 ± 5.73
LLSPIN	84.42 ± 7.12	61.16 ± 7.92	88.12 ± 1.26	88.71 ± 5.98	80.80 ± 4.90	81.67 ± 9.01	79.35 ± 7.74	70.10 ± 12.31	12.75 ± 8.63
REAL-X	83.24 ± 5.56	56.48 ± 4.90	84.16 ± 5.68	86.75 ± 6.68	77.30 ± 6.10	90.79 ± 4.75	76.75 ± 12.21	93.27 ± 4.32	12.75 ± 5.47
LSPIN	83.48 ± 6.62	58.92 ± 6.78	84.46 ± 3.36	87.75 ± 6.74	78.60 ± 5.80	83.47 ± 8.59	81.30 ± 7.97	76.92 ± 9.38	13.00 ± 5.01
TabR	81.42 ± 6.64	58.46 ± 6.68	80.84 ± 2.24	84.50 ± 8.00	75.85 ± 6.50	86.50 ± 5.00	80.75 ± 8.40	86.70 ± 6.40	14.25 ± 3.99
KNN	83.53 ± 5.76	52.05 ± 13.13	79.05 ± 8.02	78.78 ± 9.20	82.50 ± 5.00	83.86 ± 7.07	71.65 ± 12.03	91.06 ± 5.41	15.88 ± 8.63
AdaBoost	85.88 ± 7.97	58.28 ± 9.55	84.38 ± 8.32	89.19 ± 4.94	75.50 ± 5.34	57.85 ± 9.01	78.97 ± 10.96	78.32 ± 1.93	16.12 ± 8.95
XGBoost	77.06 ± 7.80	55.59 ± 6.34	90.38 ± 9.39	82.55 ± 10.22	81.50 ± 4.06	70.13 ± 7.85	72.60 ± 12.59	86.61 ± 8.72	16.88 ± 8.43
SVM	85.88 ± 2.88	61.09 ± 11.78	83.14 ± 13.37	85.75 ± 6.63	77.00 ± 2.92	66.75 ± 7.86	70.75 ± 13.93	72.77 ± 8.33	17.00 ± 10.45
GBM	80.00 ± 8.80	58.31 ± 9.72	82.10 ± 6.67	82.24 ± 5.34	82.00 ± 3.32	53.85 ± 15.31	77.31 ± 14.43	91.59 ± 2.52	17.12 ± 9.70
INVASE	80.14 ± 4.56	36.42 ± 4.46	78.90 ± 2.26	88.00 ± 6.50	71.20 ± 7.80	79.94 ± 6.60	75.40 ± 10.10	91.22 ± 6.16	19.38 ± 7.67
Naive Bayes	82.35 ± 3.72	59.32 ± 14.95	88.57 ± 2.86	60.86 ± 14.63	53.50 ± 8.31	58.49 ± 8.14	83.85 ± 10.56	84.24 ± 3.99	19.38 ± 12.85
Decision Tree	78.82 ± 9.56	56.63 ± 6.94	75.14 ± 10.18	82.33 ± 5.13	73.50 ± 5.39	45.68 ± 8.70	83.85 ± 9.15	85.16 ± 5.58	21.00 ± 9.32
MambaTab	80.16 ± 6.64	54.98 ± 9.20	68.16 ± 4.90	58.52 ± 15.60	64.00 ± 9.60	82.30 ± 6.00	80.75 ± 8.40	87.85 ± 6.00	21.12 ± 9.28
TabSeq	75.29 ± 9.98	65.16 ± 7.50	77.28 ± 14.49	65.24 ± 10.76	65.30 ± 6.45	47.95 ± 7.27	72.00 ± 11.24	86.81 ± 3.98	22.75 ± 11.00
Mambular	46.92 ± 4.56	32.90 ± 15.12	60.80 ± 5.77	81.12 ± 8.02	69.65 ± 8.20	84.95 ± 5.20	83.55 ± 7.10	89.90 ± 5.10	23.12 ± 12.70
MambAttention	76.18 ± 7.78	52.18 ± 4.46	70.16 ± 6.28	61.99 ± 14.45	49.90 ± 12.90	80.90 ± 6.40	81.90 ± 8.00	78.00 ± 9.10	24.00 ± 8.96
SAINT	78.56 ± 9.36	50.34 ± 12.16	52.92 ± 14.15	61.99 ± 14.45	57.10 ± 11.20	75.10 ± 8.20	67.60 ± 13.10	78.00 ± 9.10	27.50 ± 6.02
FT-Transformer	52.46 ± 8.92	56.45 ± 9.68	56.12 ± 12.10	85.00 ± 7.00	52.30 ± 12.30	79.45 ± 6.90	69.25 ± 12.50	67.30 ± 12.20	27.62 ± 7.82
TabM	60.46 ± 4.42	42.90 ± 6.36	66.67 ± 3.34	75.03 ± 10.08	66.10 ± 9.10	70.20 ± 9.60	65.80 ± 13.70	74.70 ± 10.10	27.62 ± 3.78
AutoInt	48.44 ± 5.65	49.68 ± 9.80	58.34 ± 9.78	83.47 ± 7.53	67.90 ± 8.70	66.80 ± 10.60	65.80 ± 13.70	74.70 ± 10.10	27.75 ± 5.50
CategoryEmbedding	69.14 ± 9.46	59.16 ± 6.68	78.12 ± 6.90	45.01 ± 19.50	54.80 ± 11.80	57.80 ± 13.30	60.50 ± 15.50	72.95 ± 10.60	28.12 ± 9.98
ResNetTabular	66.67 ± 12.16	52.16 ± 4.46	75.10 ± 3.30	48.43 ± 18.70	42.60 ± 14.70	63.30 ± 11.70	64.10 ± 14.30	74.70 ± 10.10	30.62 ± 6.14
1-D CNN	56.92 ± 5.62	40.92 ± 10.42	60.42 ± 9.92	70.00 ± 12.00	54.80 ± 11.80	71.85 ± 9.10	58.70 ± 16.10	63.50 ± 13.30	30.75 ± 4.46
NODE	64.92 ± 12.34	46.58 ± 10.45	76.92 ± 10.08	58.52 ± 15.60	59.40 ± 10.70	65.10 ± 11.10	54.95 ± 17.40	55.10 ± 15.70	31.75 ± 5.52
TabNet	55.29 ± 10.26	48.67 ± 2.17	63.89 ± 4.17	66.55 ± 15.33	50.00 ± 7.55	41.68 ± 9.03	56.75 ± 15.20	80.14 ± 12.23	32.38 ± 6.07
L2X	56.92 ± 3.58	45.68 ± 9.78	76.28 ± 2.36	61.78 ± 13.69	72.95 ± 7.30	31.72 ± 9.11	57.60 ± 13.48	50.02 ± 14.26	32.75 ± 7.74
Trompt	43.96 ± 12.16	46.65 ± 12.12	46.12 ± 4.46	69.08 ± 12.20	40.10 ± 15.30	66.80 ± 10.60	58.70 ± 16.10	61.55 ± 13.90	34.12 ± 5.41
DCN	42.86 ± 8.84	34.92 ± 14.10	43.78 ± 16.10	85.02 ± 7.05	61.75 ± 10.10	51.90 ± 15.00	48.30 ± 19.20	57.25 ± 15.10	35.12 ± 8.85
DANets	35.90 ± 6.78	29.16 ± 16.10	56.98 ± 4.48	78.58 ± 9.10	30.30 ± 17.80	59.60 ± 12.80	50.60 ± 18.60	76.35 ± 9.60	35.50 ± 7.45
TANGOS	64.84 ± 10.42	32.68 ± 6.80	50.12 ± 8.86	65.47 ± 13.30	27.90 ± 18.40	68.50 ± 10.10	38.20 ± 21.70	48.10 ± 17.50	36.75 ± 7.67
DeepFM	43.28 ± 7.82	36.12 ± 12.26	48.16 ± 12.56	45.01 ± 19.50	64.00 ± 9.60	55.90 ± 13.90	56.90 ± 16.80	59.40 ± 14.50	36.75 ± 5.12
ENODE	58.92 ± 13.12	26.12 ± 5.68	71.16 ± 4.46	52.05 ± 17.95	35.20 ± 16.50	45.40 ± 16.80	52.80 ± 18.00	50.45 ± 16.90	37.62 ± 5.68
ModernNCA	27.68 ± 14.10	31.16 ± 11.67	33.47 ± 10.24	71.95 ± 11.12	32.80 ± 17.10	68.50 ± 10.10	40.85 ± 21.10	43.10 ± 18.80	38.50 ± 8.09
TabTransformer	47.77 ± 17.71	50.26 ± 8.56	53.70 ± 15.05	51.01 ± 10.64	48.20 ± 7.57	23.66 ± 7.84	46.44 ± 16.84	21.01 ± 12.53	38.75 ± 5.28
NDTF	26.14 ± 16.10	36.18 ± 4.48	43.14 ± 6.94	54.98 ± 16.80	37.70 ± 15.90	43.10 ± 17.30	48.30 ± 19.20	63.50 ± 13.30	39.75 ± 3.20

Table E.1: Performance (mean ± std) of all models across eight HDLSS datasets. Models are sorted by average rank.

Model	HAM10000+	DeepLesion+#	MNIST+#	Fashion MNIST+#	CIFAR-10	Dog vs Cat+#	Rank (Avg ± SD)
DynaTab (Ours)	84.48 ± 0.38	94.48 ± 0.25	96.68 ± 0.16	88.52 ± 0.21	88.58 ± 0.45	99.20 ± 0.15	2.50 ± 2.35
LSPIN	81.37 ± 1.13	93.14 ± 0.29	96.77 ± 0.40	87.46 ± 0.53	87.17 ± 1.02	99.25 ± 0.13	4.67 ± 2.80
MLP	80.05 ± 0.95	92.93 ± 4.06	96.20 ± 0.76	87.85 ± 0.65	88.10 ± 0.62	99.22 ± 0.15	5.67 ± 3.01
LGBM	80.33 ± 1.33	94.73 ± 2.84	95.42 ± 0.43	87.25 ± 0.68	87.20 ± 0.75	99.12 ± 0.15	6.33 ± 3.67
TabNet	73.51 ± 3.62	93.02 ± 0.20	96.88 ± 0.38	87.99 ± 0.68	87.25 ± 0.63	99.09 ± 0.13	6.83 ± 4.62
LLSPIN	80.07 ± 1.12	90.49 ± 1.95	96.62 ± 0.20	87.46 ± 0.52	86.98 ± 0.78	99.11 ± 0.17	7.33 ± 3.01
TabR	73.53 ± 2.66	94.32 ± 1.29	96.58 ± 0.37	87.64 ± 0.39	84.60 ± 1.60	98.93 ± 0.13	7.83 ± 6.27
GBM	78.52 ± 1.36	94.55 ± 0.23	95.65 ± 0.52	87.53 ± 0.66	88.17 ± 0.71	99.25 ± 0.15	8.00 ± 3.58
XGBoost	77.42 ± 0.58	93.78 ± 3.64	95.95 ± 0.62	87.84 ± 0.83	87.20 ± 0.67	99.25 ± 0.15	8.17 ± 2.86
STG	78.86 ± 1.10	91.30 ± 5.24	95.87 ± 0.48	87.72 ± 0.50	86.12 ± 0.68	99.11 ± 0.11	9.17 ± 4.22
Lasso	78.55 ± 1.13	91.94 ± 3.62	96.09 ± 0.55	86.79 ± 0.28	87.67 ± 0.73	98.95 ± 0.16	10.33 ± 5.92
SVM	77.61 ± 0.97	92.13 ± 4.55	95.50 ± 0.37	86.48 ± 0.68	87.67 ± 1.04	99.00 ± 0.08	10.83 ± 4.88
TabM	77.95 ± 1.80	95.07 ± 0.24	95.28 ± 0.64	86.66 ± 0.61	82.40 ± 1.41	98.52 ± 0.17	12.50 ± 5.32
CatBoost	80.01 ± 1.26	92.45 ± 0.23	95.52 ± 0.51	87.39 ± 0.69	84.78 ± 0.79	98.98 ± 0.09	12.67 ± 4.55
KNN	73.37 ± 1.10	89.78 ± 2.69	90.95 ± 0.54	84.26 ± 0.62	83.60 ± 0.88	99.25 ± 0.15	13.67 ± 5.61
1-D CNN	78.42 ± 0.96	87.19 ± 0.30	95.43 ± 0.73	86.50 ± 0.37	81.42 ± 1.44	98.90 ± 0.15	14.67 ± 3.56
Random Forest	78.68 ± 0.99	86.86 ± 0.91	94.87 ± 0.53	86.85 ± 0.35	80.60 ± 0.82	99.15 ± 0.14	16.17 ± 2.79
Decision Tree	68.24 ± 0.49	93.78 ± 5.50	74.86 ± 0.70	70.31 ± 1.05	63.48 ± 1.48	97.70 ± 0.09	18.17 ± 5.53
TabSeq	76.70 ± 1.51	78.82 ± 0.52	95.83 ± 0.42	87.36 ± 0.57	85.42 ± 0.52	98.77 ± 0.17	18.33 ± 2.34
AdaBoost	74.60 ± 2.22	84.99 ± 0.52	95.16 ± 0.44	86.62 ± 0.84	75.10 ± 1.55	98.69 ± 0.14	19.00 ± 3.52
Naive Bayes	56.40 ± 1.03	18.55 ± 2.23	79.74 ± 0.85	79.59 ± 0.86	83.23 ± 1.10	97.88 ± 0.36	19.67 ± 2.25
CategoryEmbedding	76.73 ± 1.65	82.19 ± 0.75	93.02 ± 0.62	84.77 ± 0.59	70.47 ± 1.42	98.45 ± 0.18	19.67 ± 3.39

Table F.1: Performance (mean ± std) of all models across six HDHSS datasets. Models are sorted by average rank.

Model	AI-D (Case 5)	ADNI (AD123)	MOF	EEG-FE	EEG-PD	WDBC	CNAE-9	Water Potability	Rank (Avg ± SD)
DynaTab (Ours)	83.40 ± 5.10	86.12 ± 5.40	85.71 ± 1.87	98.10 ± 0.56	90.14 ± 0.24	98.14 ± 2.05	84.87 ± 3.36	62.56 ± 2.31	6.00 ± 8.72
LSPIN	85.10 ± 4.30	86.20 ± 5.40	83.29 ± 2.42	97.42 ± 0.32	89.85 ± 0.24	97.33 ± 2.52	80.51 ± 4.52	62.67 ± 2.12	8.50 ± 4.17
TabTransformer	83.10 ± 5.00	85.37 ± 5.13	80.78 ± 1.32	95.66 ± 0.58	86.82 ± 0.86	94.52 ± 3.23	84.02 ± 3.58	60.21 ± 3.69	10.38 ± 10.42
STG	81.70 ± 6.40	85.51 ± 4.49	84.07 ± 2.62	97.50 ± 0.66	89.39 ± 0.51	95.79 ± 2.48	77.85 ± 3.48	62.52 ± 2.48	12.25 ± 4.71
REAL-X	81.30 ± 5.20	83.75 ± 4.60	83.64 ± 2.93	96.74 ± 0.44	88.13 ± 0.94	97.54 ± 1.42	75.49 ± 2.70	61.40 ± 2.74	12.63 ± 8.60
MLP	81.90 ± 5.40	84.16 ± 5.51	84.21 ± 1.76	97.46 ± 0.58	89.25 ± 0.45	96.84 ± 1.42	80.96 ± 3.00	60.51 ± 1.96	13.13 ± 6.56
TabPFN v2	80.33 ± 2.81	80.76 ± 4.34	81.92 ± 6.33	—	—	98.07 ± 0.86	94.81 ± 1.26	69.29 ± 1.96	13.63 ± 20.05
LLSPIN	84.50 ± 3.70	84.83 ± 4.42	83.36 ± 1.78	95.82 ± 0.60	89.24 ± 0.34	95.79 ± 1.93	77.62 ± 3.28	59.63 ± 2.04	14.13 ± 9.09
CatBoost	82.30 ± 4.60	85.41 ± 5.10	84.07 ± 1.49	96.36 ± 0.52	88.15 ± 0.57	96.14 ± 1.63	77.48 ± 3.20	60.33 ± 2.09	14.63 ± 8.25
Lasso	81.30 ± 5.30	85.51 ± 4.49	82.21 ± 1.42	97.30 ± 0.60	88.23 ± 0.69	97.02 ± 1.51	72.59 ± 4.84	60.97 ± 2.33	15.25 ± 10.98
L2X	78.90 ± 5.70	81.64 ± 6.59	83.71 ± 2.64	94.94 ± 0.84	87.11 ± 0.49	97.37 ± 1.60	71.60 ± 2.82	61.59 ± 2.48	15.38 ± 11.10
Random Forest	82.50 ± 5.30	82.48 ± 4.45	84.21 ± 2.47	97.42 ± 0.24	88.88 ± 0.45	94.91 ± 2.84	76.20 ± 3.22	61.09 ± 2.23	16.25 ± 9.48
SVM	77.70 ± 5.40	86.32 ± 4.22	81.21 ± 1.21	97.18 ± 0.38	89.09 ± 0.40	96.49 ± 1.60	77.62 ± 2.98	59.24 ± 2.51	16.88 ± 11.78
TabulaRNN	81.80 ± 4.90	82.24 ± 5.37	83.57 ± 1.85	96.02 ± 0.60	84.52 ± 0.92	93.51 ± 1.70	78.89 ± 3.66	60.52 ± 2.29	16.88 ± 12.65
LGBM	82.50 ± 4.30	83.30 ± 5.52	82.93 ± 2.00	97.46 ± 0.40	89.15 ± 0.59	95.96 ± 2.84	73.10 ± 3.84	60.91 ± 2.21	17.88 ± 6.20
GBM	80.70 ± 5.00	82.77 ± 6.14	84.21 ± 2.85	97.02 ± 0.46	88.72 ± 0.59	95.44 ± 2.40	75.32 ± 3.48	60.79 ± 2.36	18.00 ± 6.41
XGBoost	81.70 ± 5.00	85.51 ± 4.49	82.64 ± 1.66	97.14 ± 0.36	88.68 ± 0.48	94.56 ± 2.39	75.44 ± 3.24	60.85 ± 2.43	18.25 ± 9.13
TabR	79.60 ± 4.80	82.97 ± 5.47	83.71 ± 1.85	96.64 ± 0.54	85.91 ± 0.51	95.09 ± 1.57	76.34 ± 4.00	60.09 ± 2.39	19.13 ± 12.68
FT-Transformer	81.30 ± 5.70	85.17 ± 4.78	81.86 ± 1.77	96.50 ± 0.58	84.65 ± 1.23	94.03 ± 3.20	77.48 ± 3.62	60.00 ± 2.23	19.50 ± 12.92
TabM	81.30 ± 5.00	84.02 ± 4.76	83.00 ± 1.48	95.82 ± 0.78	85.75 ± 0.82	95.44 ± 1.42	78.23 ± 2.92	59.18 ± 2.46	19.75 ± 13.55
AdaBoost	82.30 ± 5.40	80.59 ± 4.57	83.71 ± 1.51	96.82 ± 0.28	88.48 ± 0.51	95.61 ± 2.52	73.56 ± 3.62	60.24 ± 2.46	20.25 ± 9.02
MambaTab	80.60 ± 4.90	82.28 ± 4.52	83.43 ± 2.20	95.50 ± 0.88	86.48 ± 0.82	94.56 ± 2.50	75.71 ± 3.84	59.24 ± 2.59	21.00 ± 12.11
KNN	75.40 ± 5.00	83.40 ± 4.45	80.71 ± 1.40	96.88 ± 0.44	88.32 ± 0.77	95.26 ± 2.69	76.20 ± 3.12	59.27 ± 2.46	21.38 ± 7.65
TabSeq	80.60 ± 5.20	82.19 ± 5.74	83.14 ± 2.51	95.72 ± 0.82	86.51 ± 0.92	93.68 ± 2.30	77.76 ± 3.48	58.55 ± 2.31	21.38 ± 13.29
Mambular	79.30 ± 5.60	84.21 ± 4.22	81.71 ± 1.85	94.66 ± 0.72	86.48 ± 0.80	93.33 ± 2.60	75.62 ± 3.26	59.18 ± 2.33	21.50 ± 11.55
SAINT	81.50 ± 5.40	80.98 ± 5.06	82.71 ± 1.48	95.58 ± 0.74	85.49 ± 0.70	94.21 ± 2.03	77.67 ± 3.78	58.18 ± 2.36	22.25 ± 11.51
MambAttention	79.90 ± 4.90	83.60 ± 5.44	81.43 ± 1.51	95.14 ± 0.62	85.05 ± 1.23	92.63 ± 2.42	75.39 ± 3.68	59.24 ± 2.14	22.88 ± 9.30
INVASE	77.50 ± 4.80	80.93 ± 4.80	79.71 ± 1.48	95.40 ± 0.58	86.45 ± 0.77	95.79 ± 1.93	73.93 ± 2.26	57.70 ± 2.00	25.38 ± 11.53
TabNet	79.10 ± 5.20	82.38 ± 5.44	83.50 ± 1.93	93.22 ± 0.88	85.01 ± 0.82	92.98 ± 2.60	77.25 ± 3.72	57.06 ± 3.31	26.63 ± 14.72
ModernNCA	76.90 ± 5.10	82.09 ± 6.12	79.21 ± 1.67	94.94 ± 0.68	84.68 ± 0.88	91.05 ± 1.42	75.90 ± 3.12	56.76 ± 2.80	26.63 ± 16.73
Decision Tree	75.10 ± 5.40	83.11 ± 4.76	79.00 ± 1.51	95.84 ± 0.60	87.66 ± 0.42	94.74 ± 2.33	67.04 ± 4.16	58.61 ± 2.66	27.50 ± 10.31
CategoryEmbedding	79.50 ± 5.80	81.15 ± 4.16	82.57 ± 1.76	93.42 ± 1.10	84.07 ± 1.32	93.51 ± 1.60	76.76 ± 3.42	56.36 ± 2.59	27.75 ± 10.63
1-D CNN	77.10 ± 6.00	81.78 ± 4.22	82.79 ± 1.49	93.06 ± 0.78	84.87 ± 0.86	93.68 ± 1.70	76.02 ± 3.52	56.33 ± 2.68	28.63 ± 8.18
AutoInt	78.90 ± 5.20	83.11 ± 5.51	82.14 ± 1.87	94.10 ± 0.80	82.66 ± 1.00	92.81 ± 2.60	75.85 ± 3.62	56.64 ± 2.48	29.75 ± 10.86
Naive Bayes	74.40 ± 5.70	85.41 ± 4.49	74.07 ± 1.53	94.60 ± 0.82	85.17 ± 0.88	94.74 ± 2.42	64.32 ± 4.84	59.52 ± 2.21	31.13 ± 7.24
DANets	78.90 ± 5.30	80.44 ± 6.14	80.64 ± 1.64	92.66 ± 1.02	82.89 ± 1.20	90.88 ± 2.52	75.76 ± 3.54	55.94 ± 2.77	32.25 ± 12.43
NDTF	77.10 ± 5.60	79.60 ± 4.65	82.36 ± 1.55	92.22 ± 0.90	82.21 ± 1.16	92.63 ± 2.12	74.74 ± 3.88	55.36 ± 3.14	32.88 ± 10.84
ResNetTabular	75.10 ± 6.00	80.78 ± 4.88	81.93 ± 1.17	91.06 ± 0.84	83.52 ± 0.57	91.93 ± 2.30	74.83 ± 3.64	56.18 ± 2.62	34.13 ± 4.55
TANGOS	72.30 ± 5.90	81.49 ± 4.76	78.64 ± 1.48	92.66 ± 1.00	83.33 ± 0.69	92.11 ± 1.93	73.56 ± 3.42	56.24 ± 2.51	34.25 ± 5.52
NODE	73.10 ± 5.40	81.20 ± 5.51	81.93 ± 1.72	91.30 ± 0.88	83.20 ± 0.77	92.28 ± 1.60	74.69 ± 3.42	56.09 ± 2.95	34.38 ± 4.66
Trompt	73.10 ± 5.40	80.78 ± 4.88	80.21 ± 1.51	90.68 ± 1.50	83.49 ± 0.77	91.40 ± 2.39	75.16 ± 3.60	55.91 ± 2.62	35.00 ± 5.42
ENODE	72.30 ± 5.70	79.84 ± 5.06	81.14 ± 1.64	89.42 ± 0.84	82.50 ± 0.88	91.23 ± 3.37	74.42 ± 3.96	55.73 ± 2.68	36.38 ± 4.53
DeepFM	76.90 ± 5.60	82.58 ± 5.52	79.50 ± 1.67	91.84 ± 0.78	79.70 ± 1.62	90.35 ± 2.84	71.60 ± 3.64	55.64 ± 2.59	37.50 ± 12.29
DCN	70.50 ± 6.00	80.54 ± 5.06	77.86 ± 1.49	88.20 ± 1.56	82.80 ± 0.92	88.95 ± 2.42	72.68 ± 3.52	55.76 ± 3.08	39.88 ± 5.19
TabPFN	—	—	—	—	—	97.19 ± 1.02	—	—	41.50 ± 12.73
ProtoGate	64.50 ± 6.40	63.14 ± 5.51	70.14 ± 1.12	83.70 ± 1.22	74.40 ± 1.45	79.12 ± 3.37	55.90 ± 5.28	52.73 ± 2.51	44.88 ± 0.64

Table G.1: Performance (mean ± std) of all models across eight Mixed Regime (MR) datasets. Models are sorted by average rank.

Model	MiniBooNE*	Covertype*	Adult	Poker Hand	Higgs*	Rank (Avg ± SD)
CatBoost	94.19 ± 0.29	87.72 ± 0.14	87.32 ± 0.37	80.53 ± 0.91	72.73 ± 0.48	1.60 ± 0.55
XGBoost	94.21 ± 0.19	85.47 ± 0.12	86.99 ± 0.48	71.98 ± 0.72	71.98 ± 0.38	2.60 ± 1.14
Random Forest	93.37 ± 0.26	89.29 ± 0.17	85.74 ± 0.43	60.23 ± 0.44	71.75 ± 0.36	4.60 ± 2.30
LGBM	93.93 ± 0.22	84.94 ± 0.21	87.21 ± 0.38	52.51 ± 4.02	72.45 ± 0.33	5.20 ± 4.49
MLP	93.34 ± 0.32	83.88 ± 0.40	84.87 ± 0.49	63.44 ± 2.47	71.26 ± 0.35	6.80 ± 3.35
GBM	92.74 ± 0.28	77.15 ± 0.28	86.76 ± 0.37	60.39 ± 0.75	71.06 ± 0.49	7.60 ± 3.51
TabR	90.50 ± 0.30	81.85 ± 6.50	87.50 ± 0.45	54.80 ± 1.10	69.50 ± 0.40	8.40 ± 4.72
TabSeq	90.28 ± 0.20	79.93 ± 0.10	84.98 ± 0.60	54.74 ± 0.49	70.63 ± 0.29	10.40 ± 1.14
TabTransformer	86.48 ± 3.42	82.46 ± 0.48	85.00 ± 0.51	55.68 ± 1.25	70.47 ± 0.25	11.00 ± 5.39
SVM	88.95 ± 0.24	76.51 ± 0.25	85.12 ± 0.44	55.14 ± 0.43	69.73 ± 0.25	11.40 ± 3.21
DynaTab (Ours)	91.23 ± 0.80	68.80 ± 2.12	83.26 ± 0.53	52.91 ± 3.70	71.93 ± 0.35	11.40 ± 5.90
TabNet	85.32 ± 2.09	81.86 ± 0.62	84.80 ± 0.47	54.62 ± 0.76	71.77 ± 0.46	11.80 ± 6.38
CategoryEmbedding	89.75 ± 0.40	69.50 ± 2.30	83.40 ± 0.40	50.75 ± 1.60	71.26 ± 0.75	13.40 ± 4.04
TabM	87.90 ± 0.55	68.80 ± 2.30	86.80 ± 0.42	52.10 ± 1.40	70.22 ± 1.24	13.60 ± 5.59
KNN	88.83 ± 0.28	85.15 ± 0.25	83.07 ± 0.35	50.81 ± 0.61	62.02 ± 0.23	15.00 ± 6.63
FT-Transformer	88.60 ± 0.50	66.40 ± 2.55	83.00 ± 0.62	53.50 ± 1.25	68.10 ± 0.45	17.00 ± 4.74
Decision Tree	88.93 ± 0.16	84.64 ± 0.22	81.38 ± 0.54	48.99 ± 2.06	62.76 ± 0.30	17.20 ± 6.91
Lasso	85.63 ± 0.32	72.49 ± 0.49	83.65 ± 0.51	49.96 ± 6.05	64.10 ± 0.30	17.80 ± 3.11
AdaBoost	90.68 ± 0.18	61.07 ± 1.63	85.30 ± 0.25	47.10 ± 1.50	67.64 ± 0.42	17.80 ± 10.59
NODE	90.10 ± 0.35	80.90 ± 5.90	78.30 ± 1.00	40.55 ± 3.95	66.75 ± 0.50	19.80 ± 9.65
SAINT	83.30 ± 1.20	65.10 ± 2.85	82.60 ± 0.65	52.10 ± 1.40	65.20 ± 0.55	20.20 ± 4.82
Trompt	80.75 ± 1.70	67.95 ± 2.90	81.30 ± 0.75	50.75 ± 1.60	60.90 ± 0.70	22.20 ± 3.63
AutoInt	89.20 ± 0.45	60.65 ± 4.05	80.20 ± 0.85	43.60 ± 3.10	63.80 ± 0.60	24.00 ± 7.71
Mambular	79.20 ± 2.00	69.50 ± 2.25	81.80 ± 0.72	50.75 ± 1.60	41.75 ± 2.75	24.20 ± 8.87
1-D CNN	88.60 ± 0.50	67.60 ± 2.30	69.20 ± 2.55	43.60 ± 3.10	60.90 ± 0.70	25.40 ± 7.57
ResNetTabular	84.50 ± 1.00	64.55 ± 4.45	80.80 ± 0.80	49.30 ± 1.85	46.30 ± 2.00	25.60 ± 4.22
MambaTab	72.25 ± 3.60	68.10 ± 2.75	79.60 ± 0.90	48.00 ± 2.10	56.10 ± 0.95	26.40 ± 6.02
ModernNCA	70.30 ± 4.10	61.20 ± 6.50	82.20 ± 0.62	45.10 ± 2.75	50.50 ± 1.45	28.60 ± 6.19
DCN	80.75 ± 1.70	53.45 ± 6.50	76.80 ± 1.20	48.00 ± 2.10	46.30 ± 2.00	30.60 ± 6.35
LLSPIN	80.75 ± 1.70	51.35 ± 7.30	79.60 ± 0.90	37.40 ± 5.00	50.50 ± 1.45	31.40 ± 6.11
MambAttention	61.50 ± 6.60	65.90 ± 3.30	74.30 ± 1.60	40.55 ± 3.95	48.45 ± 1.70	31.60 ± 5.08
DeepFM	82.10 ± 1.40	62.20 ± 3.60	70.40 ± 2.30	35.70 ± 5.60	44.10 ± 2.35	31.80 ± 4.66
TANGOS	75.90 ± 2.70	57.25 ± 5.15	74.30 ± 1.60	42.10 ± 3.50	56.10 ± 0.95	32.20 ± 4.49
Naive Bayes	28.34 ± 0.04	9.27 ± 0.16	81.16 ± 0.73	46.92 ± 1.53	60.32 ± 0.23	32.60 ± 10.43
NDTF	77.60 ± 2.30	67.10 ± 3.20	56.52 ± 8.68	34.00 ± 6.25	44.10 ± 2.35	32.80 ± 5.97
LSPIN	77.60 ± 2.30	44.10 ± 10.10	78.30 ± 1.00	35.70 ± 5.60	48.45 ± 1.70	33.80 ± 5.63
TabulaRNN	56.40 ± 8.10	64.55 ± 4.45	75.20 ± 1.45	39.00 ± 4.45	33.60 ± 4.30	34.00 ± 5.15
DANets	82.10 ± 1.40	60.45 ± 6.90	71.50 ± 2.10	30.50 ± 9.30	36.50 ± 3.75	34.40 ± 5.90
STG	75.90 ± 2.70	62.00 ± 5.95	56.52 ± 5.66	32.30 ± 7.00	24.00 ± 6.40	36.20 ± 4.27
ENODE	80.75 ± 1.70	60.10 ± 6.42	46.92 ± 12.10	30.01 ± 12.90	30.50 ± 4.90	36.80 ± 5.89
REAL-X	53.70 ± 8.90	60.45 ± 6.90	36.48 ± 9.82	31.50 ± 7.60	27.20 ± 5.60	39.40 ± 3.29
L2X	56.40 ± 8.10	60.10 ± 5.80	46.92 ± 6.88	31.10 ± 7.60	20.50 ± 8.70	39.60 ± 2.19
INVASE	66.10 ± 5.30	55.40 ± 5.80	34.68 ± 10.10	26.35 ± 6.45	20.20 ± 9.50	41.40 ± 2.70
ProtoGate	47.75 ± 10.50	46.70 ± 9.10	34.56 ± 12.10	30.00 ± 15.30	20.00 ± 10.30	43.20 ± 0.84

Table H.1: Performance (mean ± std) of all models across five LDHSS datasets. Models are sorted by average rank. (* = 30K subsamples taken from original dataset).

Model	Iris	Pima Indian	Glass	Hayes-Roth	Monks-1	Liver Disorder	Rank (Avg ± SD)
TabPFN v2	97.33 ± 3.27	76.17 ± 1.92	80.82 ± 3.56	81.00 ± 7.70	100.00 ± 0.00	100.00 ± 0.00	4.50 ± 5.09
CatBoost	95.33 ± 3.39	77.60 ± 1.91	81.32 ± 4.62	80.26 ± 8.60	100.00 ± 0.00	100.00 ± 0.00	5.33 ± 6.41
GBM	95.33 ± 3.39	76.17 ± 1.45	75.70 ± 1.82	82.56 ± 5.15	99.31 ± 1.38	100.00 ± 0.00	9.50 ± 6.19
TabulaRNN	96.00 ± 3.89	75.78 ± 3.62	65.39 ± 8.24	80.97 ± 10.01	100.00 ± 0.00	100.00 ± 0.00	9.67 ± 8.89
AutoInt	95.33 ± 3.39	76.04 ± 2.07	62.62 ± 1.23	82.59 ± 8.00	100.00 ± 0.00	100.00 ± 0.00	11.00 ± 10.64
FT-Transformer	94.67 ± 5.42	77.86 ± 1.99	75.68 ± 3.32	77.27 ± 13.51	95.86 ± 8.28	100.00 ± 0.00	12.00 ± 10.77
NODE	95.33 ± 4.52	76.56 ± 2.23	71.48 ± 5.44	65.07 ± 14.41	99.77 ± 0.47	100.00 ± 0.00	12.33 ± 7.74
TabR	95.33 ± 4.52	75.65 ± 1.92	72.43 ± 4.53	60.51 ± 7.51	100.00 ± 0.00	100.00 ± 0.00	12.50 ± 10.91
Random Forest	96.00 ± 3.89	74.63 ± 1.19	78.51 ± 3.08	82.59 ± 3.83	98.85 ± 1.26	100.00 ± 0.00	13.00 ± 17.24
LGBM	94.67 ± 4.00	75.51 ± 2.49	78.49 ± 3.48	67.38 ± 10.24	100.00 ± 0.00	100.00 ± 0.00	13.00 ± 12.17
Trompt	96.00 ± 3.89	75.78 ± 2.81	63.10 ± 7.76	82.54 ± 6.66	97.46 ± 2.12	100.00 ± 0.00	13.00 ± 9.76
XGBoost	94.00 ± 3.27	73.17 ± 2.81	75.23 ± 4.82	83.33 ± 6.63	100.00 ± 0.00	100.00 ± 0.00	13.17 ± 16.40
TabPFN	95.33 ± 4.52	77.21 ± 1.77	71.94 ± 4.08	81.85 ± 11.47	98.16 ± 1.87	99.71 ± 0.58	13.17 ± 6.74
DynaTab (Ours)	94.07 ± 6.02	75.84 ± 2.11	75.01 ± 7.14	84.29 ± 5.35	85.45 ± 0.19	100.00 ± 0.00	14.17 ± 12.92
Mambular	95.33 ± 3.39	75.65 ± 2.04	64.95 ± 3.62	58.26 ± 13.87	100.00 ± 0.00	100.00 ± 0.00	15.00 ± 11.73
MambAttention	95.33 ± 3.39	74.08 ± 4.18	61.16 ± 6.56	79.57 ± 8.67	100.00 ± 0.00	100.00 ± 0.00	15.33 ± 13.53
LSPIN	96.00 ± 3.89	74.61 ± 2.32	65.88 ± 2.46	77.98 ± 6.27	100.00 ± 0.00	99.42 ± 0.71	15.83 ± 10.76
ResNetTabular	96.00 ± 3.89	75.12 ± 3.14	57.01 ± 5.40	83.30 ± 8.92	88.64 ± 9.56	100.00 ± 0.00	16.00 ± 14.06
Decision Tree	95.33 ± 3.39	69.52 ± 4.18	69.62 ± 3.90	83.33 ± 10.19	92.36 ± 6.73	100.00 ± 0.00	16.17 ± 14.37
LLSPIN	94.00 ± 4.89	76.17 ± 1.96	70.54 ± 7.07	72.82 ± 8.49	100.00 ± 0.00	99.71 ± 0.58	16.50 ± 10.25
TabM	96.67 ± 4.22	73.94 ± 5.04	48.54 ± 8.75	62.62 ± 18.69	100.00 ± 0.00	100.00 ± 0.00	16.67 ± 16.86
INVASE	95.33 ± 4.52	74.61 ± 2.36	69.14 ± 5.11	75.78 ± 9.63	100.00 ± 0.00	98.55 ± 1.83	17.50 ± 9.81
SVM	96.00 ± 3.89	76.17 ± 1.73	68.69 ± 3.12	78.01 ± 6.67	91.43 ± 2.03	96.52 ± 1.48	18.00 ± 9.74
CategoryEmbedding	94.00 ± 6.53	76.04 ± 1.73	62.13 ± 5.71	69.62 ± 10.29	100.00 ± 0.00	100.00 ± 0.00	18.17 ± 15.20
TabSeq	94.00 ± 4.89	77.34 ± 2.55	65.42 ± 2.52	65.85 ± 12.59	99.31 ± 1.38	100.00 ± 0.00	18.50 ± 14.76
SAINT	94.00 ± 3.27	75.78 ± 1.68	68.18 ± 2.86	58.20 ± 11.42	99.77 ± 0.47	100.00 ± 0.00	18.83 ± 13.32
1-D CNN	94.00 ± 4.89	76.43 ± 2.16	57.48 ± 8.42	62.82 ± 9.23	100.00 ± 0.00	100.00 ± 0.00	19.33 ± 16.20
MLP	93.33 ± 5.42	76.82 ± 1.78	67.26 ± 6.56	58.93 ± 18.38	100.00 ± 0.00	100.00 ± 0.00	19.50 ± 17.10
MambaTab	92.67 ± 6.11	77.21 ± 1.92	57.94 ± 4.82	65.16 ± 21.39	100.00 ± 0.00	100.00 ± 0.00	20.00 ± 17.66
ModernNCA	90.67 ± 4.99	74.34 ± 2.16	75.68 ± 4.42	68.65 ± 13.51	94.01 ± 8.19	100.00 ± 0.00	20.83 ± 11.40
TabNet	93.33 ± 4.00	75.65 ± 2.48	66.82 ± 2.73	62.58 ± 7.16	93.54 ± 9.17	100.00 ± 0.00	21.17 ± 13.17
Lasso	94.67 ± 4.00	77.47 ± 1.87	59.32 ± 6.02	58.91 ± 4.62	100.00 ± 0.00	97.10 ± 0.78	22.00 ± 15.15
TabTransformer	93.33 ± 4.00	74.08 ± 4.02	66.82 ± 4.22	61.09 ± 17.69	93.08 ± 6.42	100.00 ± 0.00	22.17 ± 13.42
STG	92.00 ± 4.89	76.56 ± 2.23	61.66 ± 8.42	64.43 ± 10.42	97.92 ± 1.81	99.71 ± 0.58	22.50 ± 13.97
AdaBoost	94.00 ± 4.89	77.21 ± 2.04	75.20 ± 4.40	48.37 ± 10.52	86.35 ± 7.17	96.52 ± 2.77	23.17 ± 14.36
REAL-X	92.00 ± 8.00	76.95 ± 2.36	68.18 ± 5.84	62.56 ± 7.82	88.65 ± 12.46	97.68 ± 2.23	24.00 ± 11.01
DeepFM	87.33 ± 6.11	71.48 ± 2.74	69.61 ± 5.48	62.59 ± 6.76	100.00 ± 0.00	100.00 ± 0.00	25.00 ± 14.60
DANets	92.67 ± 4.00	75.91 ± 3.31	63.54 ± 6.16	62.59 ± 7.80	78.94 ± 10.90	99.42 ± 0.71	26.17 ± 11.64
NDTF	90.00 ± 9.80	73.95 ± 2.47	60.26 ± 3.42	63.67 ± 11.11	94.01 ± 8.19	100.00 ± 0.00	26.83 ± 14.50
DCN	93.33 ± 4.00	74.60 ± 2.44	56.08 ± 3.62	57.41 ± 9.18	94.93 ± 5.87	99.42 ± 0.71	27.00 ± 15.52
KNN	97.33 ± 2.49	72.78 ± 1.96	64.43 ± 6.86	47.72 ± 7.72	83.33 ± 5.96	92.75 ± 2.43	28.00 ± 14.28
ENODE	90.00 ± 5.66	75.39 ± 2.40	51.45 ± 6.02	59.38 ± 9.03	81.02 ± 21.62	99.42 ± 0.71	30.00 ± 13.88
Naive Bayes	94.67 ± 4.00	75.52 ± 3.38	44.85 ± 11.48	66.75 ± 8.33	66.66 ± 2.50	93.62 ± 2.53	31.33 ± 8.16
TANGOS	86.00 ± 6.53	75.52 ± 2.24	62.15 ± 4.66	60.43 ± 7.62	69.21 ± 17.67	96.52 ± 0.45	31.50 ± 9.85
L2X	85.33 ± 5.07	71.74 ± 2.88	56.53 ± 6.42	60.49 ± 2.90	75.46 ± 9.32	99.42 ± 0.71	34.67 ± 10.56
ProtoGate	78.67 ± 7.02	58.45 ± 2.39	31.30 ± 6.40	55.15 ± 14.55	75.00 ± 16.14	65.51 ± 4.41	40.50 ± 7.71

Table I.1: Performance (mean ± std) of all 46 models across six LDLSS datasets. Models are sorted by average rank.