Appendix: Hybrid Autoencoders for Tabular Data: Leveraging Model-Based Augmentation in Low-Label Settings

Extending the spectral analysis in the main paper (Section 6), we present additional visualizations across more datasets to further examine how the gating mechanisms in TANDEM shape the frequency content of the input. For each dataset, we focus on a single class and compute the unnormalized discrete Fourier transform (NUDFT) over the 50 most variant features. We compare the spectra of the original input x; the neural-gated input $\tilde{x}^{\mathrm{NN}} = x \odot g^{\mathrm{NN}}(x)$, as produced by both TANDEM and SS-AE with gating; and the tree-gated input $\tilde{x}^{\mathrm{OSDT}} = x \odot \bar{g}^{\mathrm{OSDT}}(x)$, where $\bar{g}^{\mathrm{OSDT}}$ is the average gating mask across all trees and depths.

These extended plots confirm the patterns observed in the main paper: namely, that neural gating acts as a strong low-pass filter, suppressing high-frequency components, while tree-based gating preserves more high-frequency variation, which reinforces their complementary inductive biases in shaping representations.

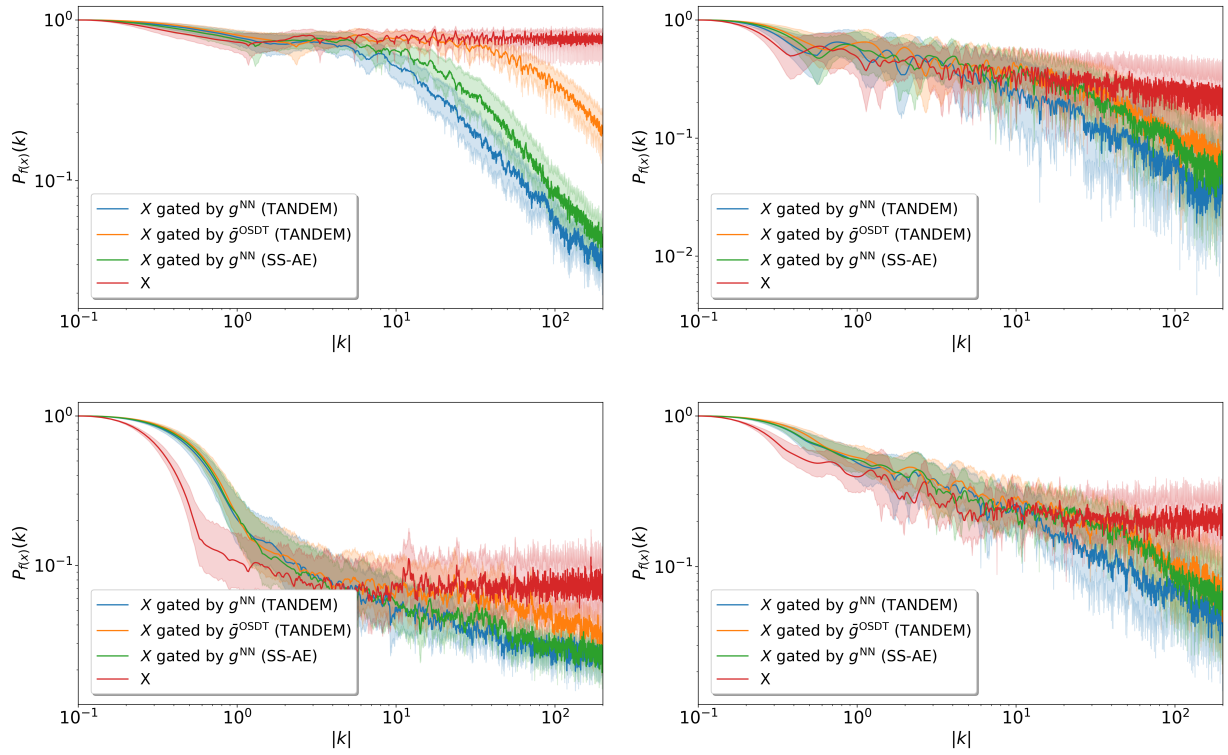

Figure 1: Frequency spectra of gated inputs for the NN and OSDT encoders. Visualized datasets include OG, CP, VO, and CC. NN gating results in stronger suppression of high-frequency components compared to tree-based gating.

A Comparative Performance and Ablation Analysis

This section presents the same tables shown in the main paper, with added standard deviation error bars.

Dataset	LogReg	DeepTLF	TabM	TabPFN	XGBoost	CatBoost	MLP	SS AE	TANDEM
CP	0.4927 ± 0.0138	0.5438 ± 0.0390	0.5827 ± 0.0283	0.5998 ± 0.0294	0.5822 ± 0.0298	0.5401 ± 0.0114	0.5792 ± 0.0291	0.6505 ± 0.0303	**0.6779 ± 0.0325**
MT	0.7521 ± 0.0252	0.6757 ± 0.0529	0.7720 ± 0.0252	0.8139 ± 0.0216	0.7868 ± 0.0215	0.7929 ± 0.0166	0.7798 ± 0.0213	0.7996 ± 0.0255	**0.8180 ± 0.0228**
OG	0.6228 ± 0.0134	0.3712 ± 0.1913	0.6228 ± 0.0132	0.6514 ± 0.0125	0.6136 ± 0.0115	0.6363 ± 0.0116	0.6321 ± 0.0125	0.6500 ± 0.0118	**0.6870 ± 0.0123**
PW	0.9373 ± 0.0315	0.9281 ± 0.0161	0.9384 ± 0.0152	0.9477 ± 0.0136	0.9353 ± 0.0137	0.9327 ± 0.0296	0.9315 ± 0.0152	0.9353 ± 0.0173	**0.9618 ± 0.0143**
AD	0.7992 ± 0.0221	0.7645 ± 0.0544	0.8115 ± 0.0207	0.8199 ± 0.0200	0.7875 ± 0.0216	0.8019 ± 0.0177	0.8006 ± 0.0260	**0.8202 ± 0.0220**	0.8200 ± 0.0217
ALB	0.6065 ± 0.0265	0.5512 ± 0.0349	0.6196 ± 0.0228	0.6494 ± 0.0324	0.6096 ± 0.0234	0.6354 ± 0.0248	0.5929 ± 0.0347	0.6497 ± 0.0322	**0.7038 ± 0.0283**
BM	**0.8325 ± 0.0155**	0.6306 ± 0.1093	0.8132 ± 0.0292	0.8241 ± 0.0287	0.7991 ± 0.0300	0.8171 ± 0.0246	0.7913 ± 0.0301	0.8141 ± 0.0218	0.8233 ± 0.0178
CO	0.4624 ± 0.0086	0.4881 ± 0.0250	0.5049 ± 0.0172	0.5485 ± 0.0160	0.5326 ± 0.0167	0.4975 ± 0.0111	0.4963 ± 0.0143	0.5007 ± 0.0154	**0.5491 ± 0.0150**
CC	0.6169 ± 0.0249	0.6302 ± 0.0370	0.6218 ± 0.0394	0.6451 ± 0.0374	0.6780 ± 0.0362	0.6690 ± 0.0240	**0.7190 ± 0.0358**	0.6833 ± 0.0248	0.7331 ± 0.0251
EL	0.6617 ± 0.0313	0.6424 ± 0.0274	0.6610 ± 0.0315	**0.7723 ± 0.0279**	0.7668 ± 0.0292	0.7654 ± 0.0249	0.7525 ± 0.0296	0.7429 ± 0.0309	0.6940 ± 0.0240
HE	0.4593 ± 0.0119	—	—	—	0.4590 ± 0.0173	0.4853 ± 0.0111	0.4448 ± 0.0184	0.5250 ± 0.0132	**0.5462 ± 0.0140**
HI	0.5454 ± 0.0257	0.4965 ± 0.0284	0.5532 ± 0.0259	**0.6499 ± 0.0248**	0.6096 ± 0.0266	0.6069 ± 0.0309	0.6035 ± 0.0254	0.6179 ± 0.0326	0.6459 ± 0.0333
JA	0.5105 ± 0.0166	0.4649 ± 0.0292	0.5743 ± 0.0248	**0.5986 ± 0.0254**	0.5409 ± 0.0262	0.5413 ± 0.0159	0.5417 ± 0.0249	0.5168 ± 0.0195	0.5660 ± 0.0212
NU	0.4833 ± 0.0221	0.4932 ± 0.0153	0.5221 ± 0.0144	0.4333 ± 0.0170	0.5308 ± 0.0141	0.5231 ± 0.0177	0.5517 ± 0.0151	0.6092 ± 0.0320	**0.6545 ± 0.0355**
RS	0.6992 ± 0.0186	0.6590 ± 0.0308	**0.7886 ± 0.0272**	0.7554 ± 0.0283	0.7057 ± 0.0289	0.7328 ± 0.0169	0.7243 ± 0.0273	0.7016 ± 0.0253	0.7576 ± 0.0289
VO	0.4624 ± 0.0086	0.3966 ± 0.0340	0.4790 ± 0.0306	0.5082 ± 0.0313	0.4736 ± 0.0304	0.4975 ± 0.0111	**0.5400 ± 0.0298**	0.5154 ± 0.0128	0.5220 ± 0.0138
Mean Accuracy	0.6257	0.5824	0.6577	0.6812	0.6507	0.6612	0.6551	0.6708	**0.6975**
Mean Rank	6.94	8.53	5.13	2.93	5.38	4.81	5.31	3.63	**1.81**

Table A.1: Accuracy results for 8 baseline models and TANDEM, across 19 datasets. Bold indicates the best result per dataset.

Dataset	SS AE + Gating	OSDT AE + Gating	TANDEM (no gate)	TANDEM (no LRS + Alignment)	TANDEM
CP	0.6602 ± 0.0318	0.6321 ± 0.0303	0.6740 ± 0.0324	0.6740 ± 0.0324	**0.6779 ± 0.0325**
MT	0.8096 ± 0.0255	0.7639 ± 0.0240	0.8117 ± 0.0214	0.8014 ± 0.0206	**0.8180 ± 0.0228**
OG	0.6740 ± 0.0120	0.6601 ± 0.0356	0.6659 ± 0.0152	0.6671 ± 0.0132	**0.6870 ± 0.0123**
PW	0.9353 ± 0.0181	0.8259 ± 0.0310	0.9464 ± 0.0129	0.9485 ± 0.0141	**0.9618 ± 0.0143**
AD	0.8085 ± 0.0232	0.7688 ± 0.0585	0.8111 ± 0.0223	0.7990 ± 0.0216	**0.8200 ± 0.0217**
ALB	0.6797 ± 0.0320	0.6652 ± 0.0284	0.6763 ± 0.0284	0.6800 ± 0.0277	**0.7038 ± 0.0283**
BM	0.8141 ± 0.0237	0.7552 ± 0.0199	0.8138 ± 0.0214	0.8059 ± 0.0231	**0.8233 ± 0.0178**
CO	0.5207 ± 0.0170	0.5373 ± 0.0499	0.5220 ± 0.0148	0.5164 ± 0.0163	**0.5491 ± 0.0150**
CC	0.6900 ± 0.0284	0.6892 ± 0.0523	**0.7434 ± 0.0254**	0.7188 ± 0.0265	0.7331 ± 0.0251
EL	**0.7429 ± 0.0332**	0.6149 ± 0.0444	0.7134 ± 0.0252	0.7001 ± 0.0276	0.6940 ± 0.0240
HE	0.5350 ± 0.0136	0.4416 ± 0.0656	0.5315 ± 0.0139	0.5193 ± 0.0142	**0.5462 ± 0.0140**
HI	0.6279 ± 0.0330	**0.6621 ± 0.0528**	0.6300 ± 0.0299	0.6350 ± 0.0274	0.6459 ± 0.0333
JA	0.5468 ± 0.0234	0.5012 ± 0.0465	0.5529 ± 0.0175	0.5541 ± 0.0204	**0.5660 ± 0.0212**
NU	0.6092 ± 0.0342	0.6140 ± 0.0393	0.6325 ± 0.0318	0.6377 ± 0.0307	**0.6545 ± 0.0355**
RS	0.7016 ± 0.0260	0.7252 ± 0.0274	0.7311 ± 0.0252	0.7335 ± 0.0270	**0.7576 ± 0.0289**
VO	**0.5254 ± 0.0142**	0.4334 ± 0.0231	0.4741 ± 0.0126	0.4983 ± 0.0140	0.5220 ± 0.0138
Mean Accuracy	0.6801	0.6431	0.6831	0.6768	**0.6975**
Mean Rank	2.75	3.56	2.38	2.88	**1.31**

Table A.2: Accuracy results for TANDEM with encoder and gating variants across 19 datasets. Bold indicates the best result per dataset.

Dolan–Moré curves: Figure 4 in the main paper presents Dolan–Moré performance profiles, which show the proportion of datasets where each model achieves performance within a factor τ of the best-performing model. Unlike rank-based comparisons, these curves capture both accuracy and robustness, providing a more complete view of model consistency across diverse tasks. Higher curves indicate models that maintain strong performance across a larger share of datasets, even if not ranked first.

B Dataset Details

Table B.1 provides an overview of the datasets used in our experiments, including the number of samples, input features, and target classes. This supplements the dataset summary introduced in Section 5.1 of the main paper.

Dataset (Acronym)	#Samples (N)	#Features (F)	#Classes (C)
Click_prediction_small (CP)	9200	27	2
MagicTelescope (MT)	19020	11	2
Otto-Group-Product-Classification-Challenge (OG)	16400	93	5
PhishingWebsites (PW)	9200	30	2
adult (AD)	9200	14	2
albert_categorical (ALB)	9200	25	2
bank-marketing (BM)	45200	16	2
covertype (CO)	283300	54	6
default-of-credit-card-clients_categorical (CC)	5200	23	2
electricity (EL)	19240	11	2
helena (HE)	36260	27	13
higgs (HI)	470080	28	2
jannis (JA)	83600	54	4
numerai28.6 (NU)	9200	119	2
road-safety_categorical (RS)	363240	67	2
volkret (VO)	14800	181	8
pol (PO)	15000	48	2
california (CA)	20634	8	2
eye_movements (EM)	10935	27	3

Table B.1: Dataset statistics: number of samples (N), features (F), and target classes (C).

C Experimental Setup

We follow the training and evaluation protocol described in Section 5.2 of the main paper. All experiments use 2,000 unlabeled and 400 labeled examples per class, with 100 pretraining epochs and batch size 128. During supervised training, encoders were frozen for the first 25 epochs and then fine-tuned for an additional 25 epochs. Gating modules were frozen throughout.

All experiments were conducted on a local machine equipped with an Intel i7 CPU, 16GB RAM, and an NVIDIA RTX 3060 GPU (12GB).

D Model Architectures and Training

This section outlines the architectural components used in our experiments, following the design described in Sections 4.1 and 4.3 of the main paper. The final embedding dimensionality is determined by the depth parameter L of the OSDT encoder, which was selected separately for each dataset based on validation performance.

- **Neural Encoder:** 4-layer MLP with BatchNorm and Leaky ReLU activations. Hidden dimensions are chosen to match the embedding size dictated by the OSDT encoder.
- **OSDT Encoder:** Ensemble of oblivious soft decision trees with fixed depth L; each tree outputs a soft assignment over 2^L leaves. The mean-aggregated output defines the embedding dimension.
- **Neural Decoder:** Mirrors the architecture and dimensionality of the neural encoder.
- **Gating Network:** 2-layer MLP with tanh activation and hard-sigmoid output; used for per-sample feature selection.
- **Fine-tuning MLP:** A single-layer fully connected classifier trained on top of the encoder for downstream supervised evaluation.

E Hyperparameter Tuning

Table E.1 summarizes the hyperparameter search space used in Optuna-based tuning (50 trials per model, per dataset). This supports the configuration described in Section 5.2 of the main paper. Gating components, when applicable, were tuned separately.

Table E.1: Hyperparameter search space for all models. Each model was optimized using Optuna with 50 trials. Gating components (when applicable) were tuned separately.

Model	Hyperparameter Search Space	Notes
TabNet	learning rate: $[10^{-4}, 10^{-1}]$ mask type: {sparsemax, entmax} scheduler step size: [5–20], gamma: [0.8–0.95] batch size: {512, 1024, 2048} virtual batch size: {64, 128, 256} max epochs: [10–100], patience: [5–20]	Pretrained with TabNetPretrainer before supervised fine-tuning
MLP	learning rate: $[10^{-4}, 10^{-2}]$ (log scale) hidden sizes: [64–256] batch size: {32, 64, 128} epochs: [10–50]	4-layer MLP with ReLU activations
XGBoost	max depth: [3–10] gamma: $[10^{-8}, 1]$ subsample, colsample bytree: $[0.5, 1.0]$ reg alpha, reg lambda: $[10^{-8}, 10]$	`use_label_encoder=False`
CatBoost	depth: [2–10] learning rate: $[10^{-3}, 0.3]$ iterations: [100–300]	`verbose=0`
Logistic Regression	C: $[10^{-4}, 10]$ (log-uniform)	`max_iter=5000`
DeepTFL	n estimators: [10–100] max depth: [2–7] dropout: [0.0–0.3] n layers: [1–3]	Internal tree-based layers
TabM	n blocks: [2–4] d block: [64–256] dropout: [0.0–0.3] k: [8–64] learning rate: $[10^{-4}, 5 \times 10^{-3}]$ (log)	–
TabPFN	n estimators: {1, 2, 4, 8} softmax temperature: $[0.5, 1.0]$ balance probabilities: {True, False} average before softmax: {True, False}	Pretrained; limited tuning allowed
SS-AE	learning rate: $[10^{-4}, 10^{-2}]$ weight decay: $[10^{-5}, 10^{-2}]$ depth: [3–10]	Encoder uses powers-of-two hidden sizes. Gating tuned separately.
TANDEM	learning rate: $[10^{-4}, 10^{-2}]$ weight decay: $[10^{-5}, 10^{-2}]$ depth: [3–10] num trees: [2–16] gating optimizer, activation, learning rate	Shared decoder. Gating network tuned separately.
Gating Network	hidden size: [32–128] learning rate: $[10^{-4}, 10^{-2}]$ weight decay: $[10^{-5}, 10^{-2}]$	Used in SS-AE and TANDEM for per-sample masking.