Mixture of In-Context Prompters for Tabular PFNs

Derek Xu[1], Olcay Cirit[2], Reza Asadi[2], Yizhou Sun[1], Wei Wang[1]

[1]University of California, Los Angeles
[2]Uber Technologies Inc

Paper QR

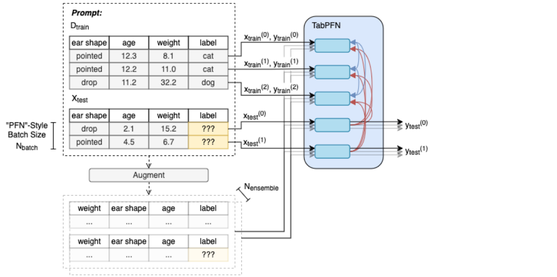
Summary

*"We are the first to propose Sparse Mixture of In-Context Prompters (MiCP) and Context-Aware finetuning (CaPFN), which extends **tabular in-context learning** to **larger datasets**.*

- *Existing tabular in-context learning are capped by their limited context size (typically 3000 training samples).*
- *MiCP extends the context length by **splitting larger datasets into local clusters** using K-Means*
- *CaPFN proposes **a novel finetuning strategy to learn in-context experts** for each cluster by **bootstrapping new contexts** for each cluster via K-Nearest Neighbors.*
- *Our approach achieves **state-of-the-art results** on the TabZilla tabular learning benchmark.*

Background: Prior Fitted Networks

Pretrain a transformer on data-generating prior. During inference, pass downstream dataset (≤3K samples) into the PFN model.



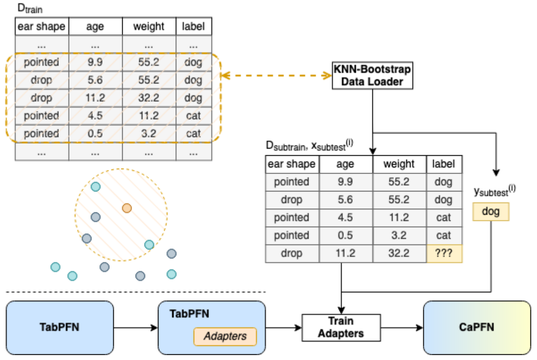
Our Proposal: Mixture of Prior Fitted Networks

Divide and Conquer local clusters of tabular data with PFN

1) Cluster downstream training data into smaller clusters.
2) Down-sample each cluster into a fixed context size.
3) Finetune specialized experts on each cluster of training data.

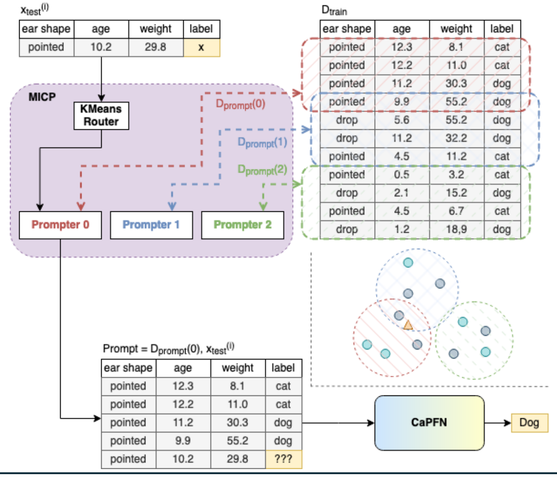
Finetuning: Context-Aware PFN (CaPFN)

Finetune PFN by bootstrapping KNN clusters around the K-Means centroids found by Mixture of Prompters to create PFN experts.



Inference: Mixture of In-Context Prompters (MiCP)

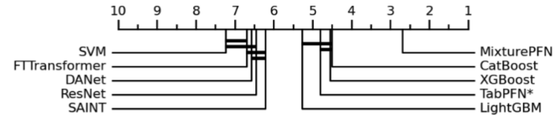
Compute K-Means centroids to define input space of each expert. During inference, route individual samples and corresponding cluster (≤3K samples) to individual PFN experts.



On TabZillA Benchmark...

Method	Condorcet Statistics #Votes↑	#Wins↑	#Ties	#Losses↓	All Algo. Mean ± Std Rank↓
MixturePFN	**524**	**19**	0	**0**	**2.350 ± 1.824**
XGBoost	500	18	0	1	5.500 ± 4.621
CatBoost	474	17	0	2	4.900 ± 4.158
SAINT	408	16	0	3	8.300 ± 5.367
TabPFN*	381	13	1	5	4.550 ± 2.747
LightGBM	373	14	1	4	9.150 ± 4.351
DANet	312	14	0	5	9.050 ± 3.369
FTTransformer	294	12	0	7	8.600 ± 3.541
ResNet	286	11	0	8	8.400 ± 3.262
SVM	285	9	0	10	11.300 ± 4.766
STG	284	10	0	9	11.900 ± 4.549
RandomForest	247	7	0	12	11.600 ± 4.443
NODE	243	7	0	12	13.350 ± 3.410
MLP-rtdl	227	5	0	14	10.800 ± 5.046
TabNet	210	5	0	14	13.550 ± 5.296
LinearModel	202	3	1	15	12.400 ± 4.652
MLP	191	5	1	13	13.700 ± 3.621
VIME	134	2	0	17	15.350 ± 3.851
DecisionTree	114	1	0	18	16.800 ± 3.881
KNN	74	0	0	19	18.450 ± 1.936

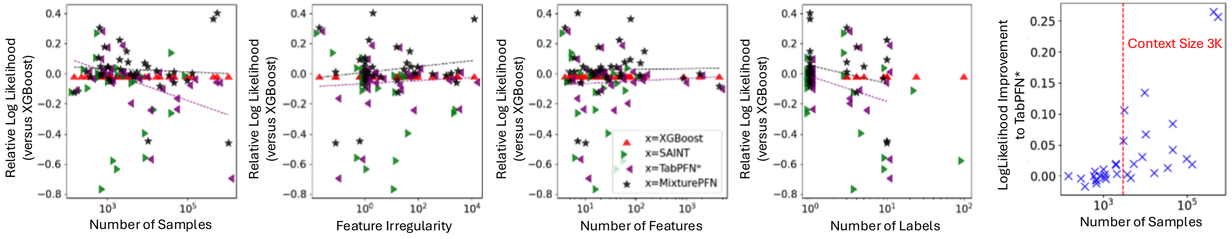
State of the Art Results



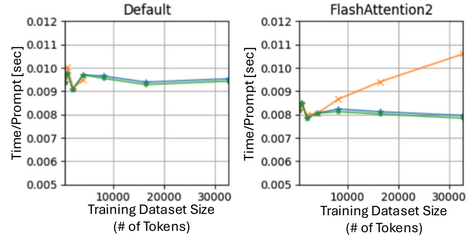
Ablation Study

Method	Mean ± Std Rank↓	Median Rank↓	Min Rank↓	Max Rank↓
MixturePFN	**2.75 ± 1.94**	**2**	**1**	9
CatBoost	4.70 ±3.16	6	**1**	9
MixturePFN (CaPFN w. Full FT)	4.85 ±2.03	4.5	2	9
XGBoost	4.95 ±3.17	6	**1**	10
MixturePFN (-CaPFN)	5.10 ±1.12	5	3	8
MixturePFN (-CaPFN-MICP) = TabPFN*	5.30 ±1.59	5	2	9
MLP-rtdl	7.25 ±2.79	8	**1**	10
MLP	8.85 ±0.99	9	7	10

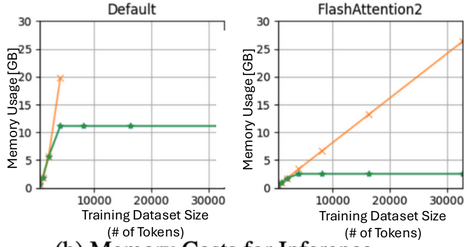
Scalability Studies



(a) N_{train} w.r.t. baselines (b) Kurtosis w.r.t. baselines (c) #Feat w.r.t. baselines (d) #Class w.r.t. baselines (a) N_{train} w.r.t. TabPFN*



(a) Runtime Costs for Inference



(b) Memory Costs for Inference