TabCaps: A Capsule Neural Network for Tabular Data Classification with BoW Routing

Jintai Chen , Kuanlun Liao, Yanwen Fang, Danny Z. Chen, Jian Wu*

Background

Tabular Data

columns = attributes for those observations

Rows = observations

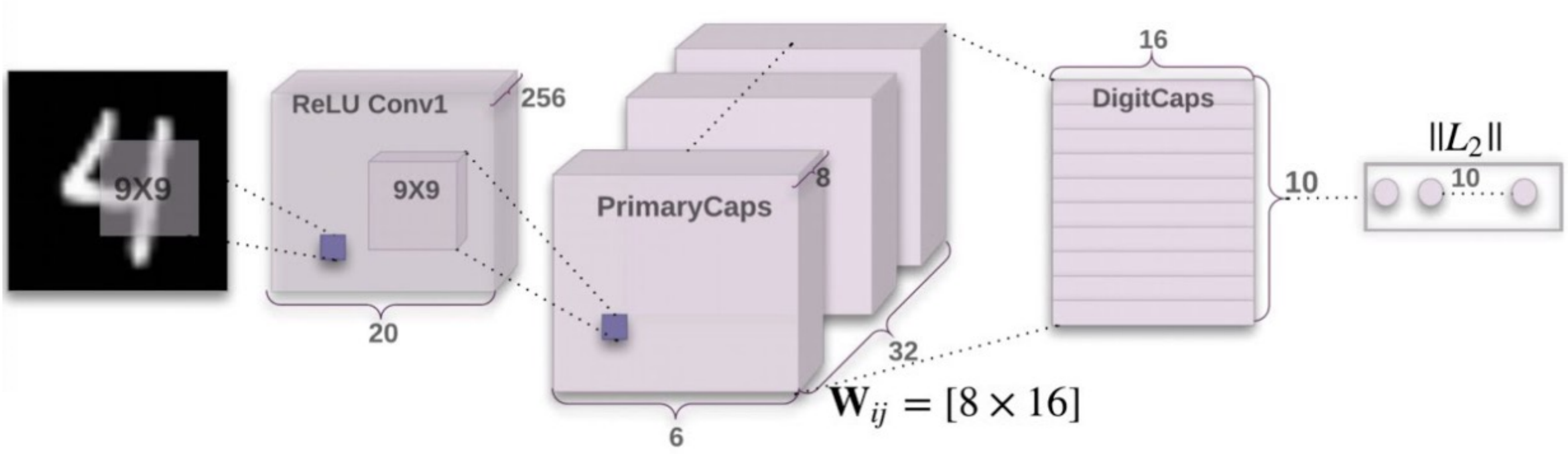
Player	Minutes	Points	Rebounds	Assists
A	41	20	6	5
B	30	29	7	6
C	22	7	7	2
D	26	3	3	9
E	20	19	8	0
F	9	6	14	14
G	14	22	8	3
I	22	36	0	9
J	34	8	1	3

Tabular data are represented by heterogeneous scalar features.

These features are aligned but their relations are unknown.

Mining interactions between heterogeneous features requires a higher sample complexity.

Background

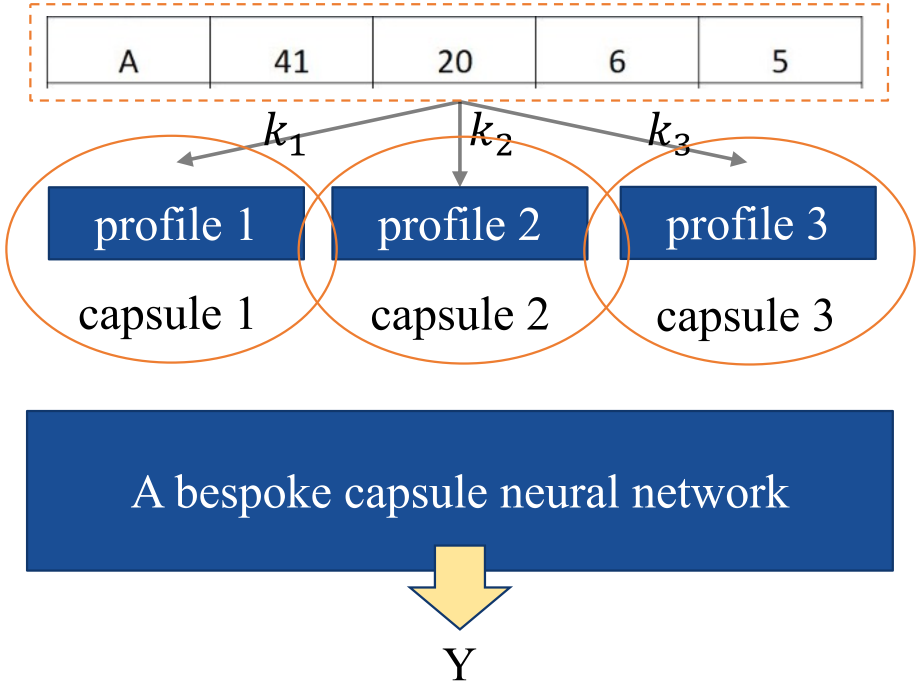


Capsule Neural Networks uses "capsules" to package scalar features as units.

The capsule features represent more concrete semantics.

Since mining feature relations is complex on tabular data, how about packaging them together and conducting no interactions?

IDEA: Use Capsule and Conduct NO Feature Interactions



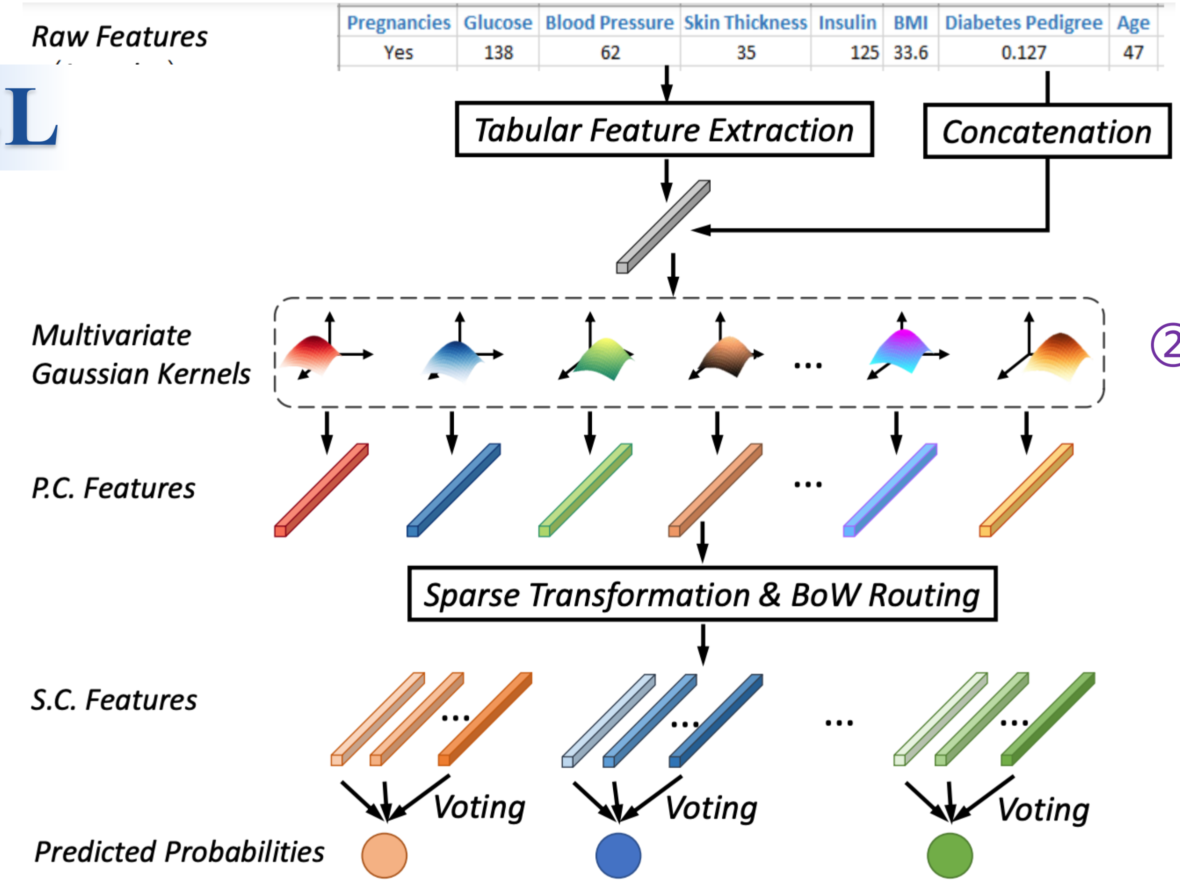
Each function *k* feature-wisely transforms a sample into a vector.

Each capsule learns a profile (the vector) of the sample.

Conduct no feature interactions and directly learning the semantics at data level,

so we call it Data-Level Learning.

MODEL



① Add some features to obtain more features

② $$u_i = k_i(x; \mu_i, \Sigma_i) = \frac{\exp(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i))}{(2\pi)^{m/2}|\Sigma_i|}$$

③ $$\hat{u}_{j|i} = u_i W'_{j|i}\ ,\ W'_{j|i} = entmax_\alpha(W_{j|i}),$$
$$u_j = Routing(\hat{u}_{j|i}).$$

④ $$l_k = \sum_{j \in G_k} ||v_j||_2 / ||G_k||$$
20% of v_j are dropped out in training

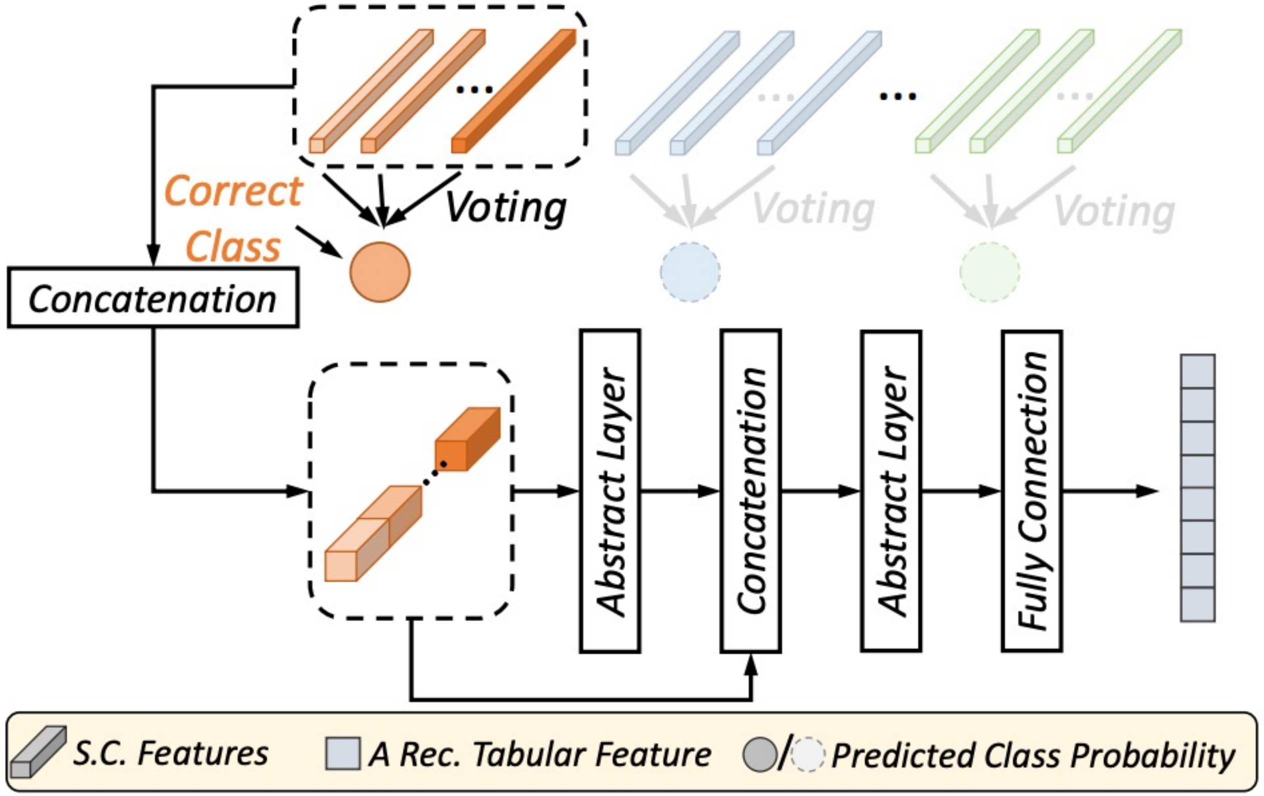
① feature extension by Abstract Layer or MLP (automatic feature engineering)

② Gaussian kernels as function *k*

③ Transformation and Routing for selective capsule-feature-fusion

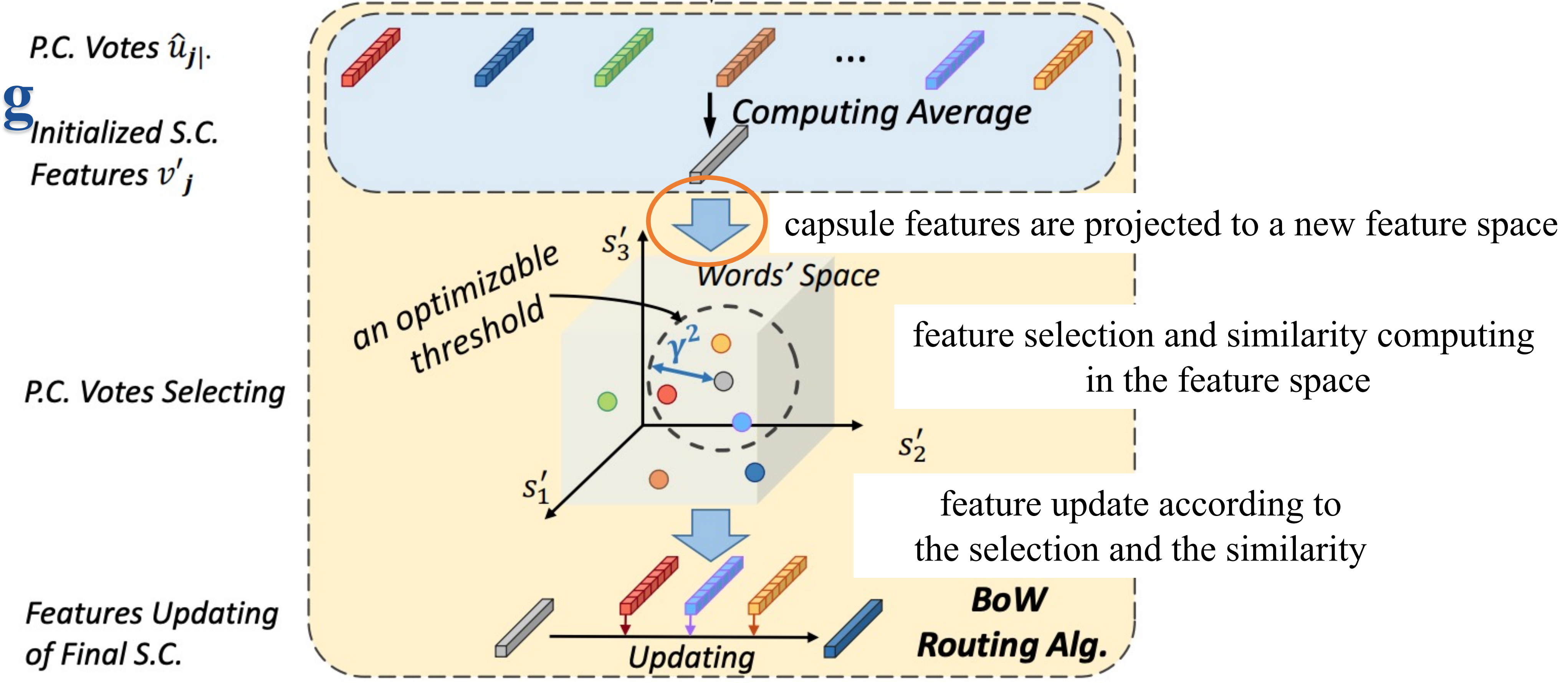
④ voting for prediction results (TIPS: dropout is helpful!)

Decoder



The corresponding capsule features of correct class are stacked for reconstruction.

BoW-Routing



Why the BoW Routing requires no iterations?

Previous CapsNets for images capture some unknown object parts in initializing capsule features and need routing-by-agreement. However, our data-level learning learns concrete profiles of the entire data and thus we believe that our routing does not need agreement.

Experiments

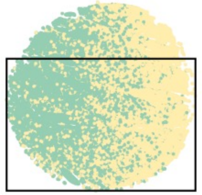
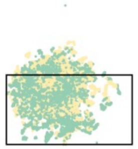
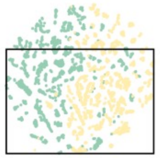
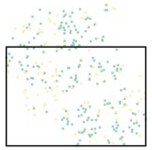
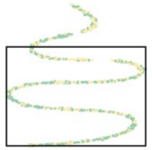
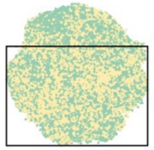
Table 1: **Classification Performances.** The best and second best performances of deep learning approaches are respectively marked in **bold** and underlined. Note that the reported log-loss values (the lower the better) are with a 100× factor. The model size (# param.) and inference speed (fps) are on the *Diabetes* dataset. The performances are reported as "mean$\pm$std".

Method	Click	Diabetes	EEG	Gas	Heart	Hill	Higgs	Epsilon	# param.	fps
XGboost	62.253$\pm$0.02	14.338$\pm$0.03	14.117$\pm$0.02	2.087$\pm$0.06	32.371$\pm$0.04	69.049$\pm$1e-3	53.158$\pm$0.01	26.748$\pm$1e-3	–	–
Catboost	64.273$\pm$0.08	14.777$\pm$0.07	18.423$\pm$0.12	2.064$\pm$0.05	30.043$\pm$0.14	69.174$\pm$0.06	53.273$\pm$0.05	27.228$\pm$2e-3	–	–
TabNet	62.303$\pm$0.03	17.964$\pm$0.04	45.340$\pm$0.04	4.647$\pm$0.04	44.967$\pm$0.01	87.804$\pm$0.07	54.668$\pm$0.03	26.743$\pm$0.02	3.4M	73.1
Net-DNF	67.633$\pm$0.02	13.767$\pm$0.02	17.386$\pm$0.01	**1.229**$\pm$0.04	55.371$\pm$0.03	15.787$\pm$0.03	53.417$\pm$0.02	27.122$\pm$0.03	8.5M	175.2
NODE	63.206$\pm$0.05	45.951$\pm$0.03	47.654$\pm$0.04	38.774$\pm$0.04	46.541$\pm$0.04	69.220$\pm$0.04	61.864$\pm$0.08	27.838$\pm$0.54	13.4M	145.2
FT-Transformer	70.487$\pm$0.02	12.382$\pm$0.03	**7.446**$\pm$0.06	2.258$\pm$0.04	**27.547**$\pm$0.05	20.084$\pm$0.03	53.310$\pm$0.02	25.958$\pm$0.85	9.3M	284.7
DANet-24	73.708$\pm$0.02	13.338$\pm$0.02	9.301$\pm$0.04	2.171$\pm$0.02	49.643$\pm$0.04	24.763$\pm$0.03	**53.033**$\pm$0.01	26.431$\pm$0.01	5.5M	54.9
FCNN w/ mixup	63.863$\pm$0.07	12.715$\pm$0.05	9.572$\pm$0.07	2.083$\pm$0.06	36.742$\pm$0.02	56.005$\pm$0.05	56.787$\pm$0.04	27.467$\pm$0.03	0.7M	594.3
FCNN w/ lasso	87.005$\pm$0.17	41.071$\pm$0.75	31.852$\pm$0.05	4.141$\pm$0.06	44.881$\pm$0.06	69.302$\pm$0.01	132.102$\pm$0.07	32.282$\pm$0.02	0.7M	568.8
Vector CapsNet	64.135$\pm$0.05	52.635$\pm$0.03	53.587$\pm$0.06	161.547$\pm$0.03	58.516$\pm$0.04	51.591$\pm$0.02	62.654$\pm$0.02	54.252$\pm$0.02	0.4M	318.5
TabCaps (Ours)	**62.054**$\pm$0.04	**12.043**$\pm$0.03	8.130$\pm$0.05	2.013$\pm$0.03	34.047$\pm$0.02	**14.301**$\pm$0.04	53.776$\pm$0.03	**25.821**$\pm$0.02	0.2M	501.1

The performances are competitive to or even better than other approach that conducts complex feature interactions.

Experiments

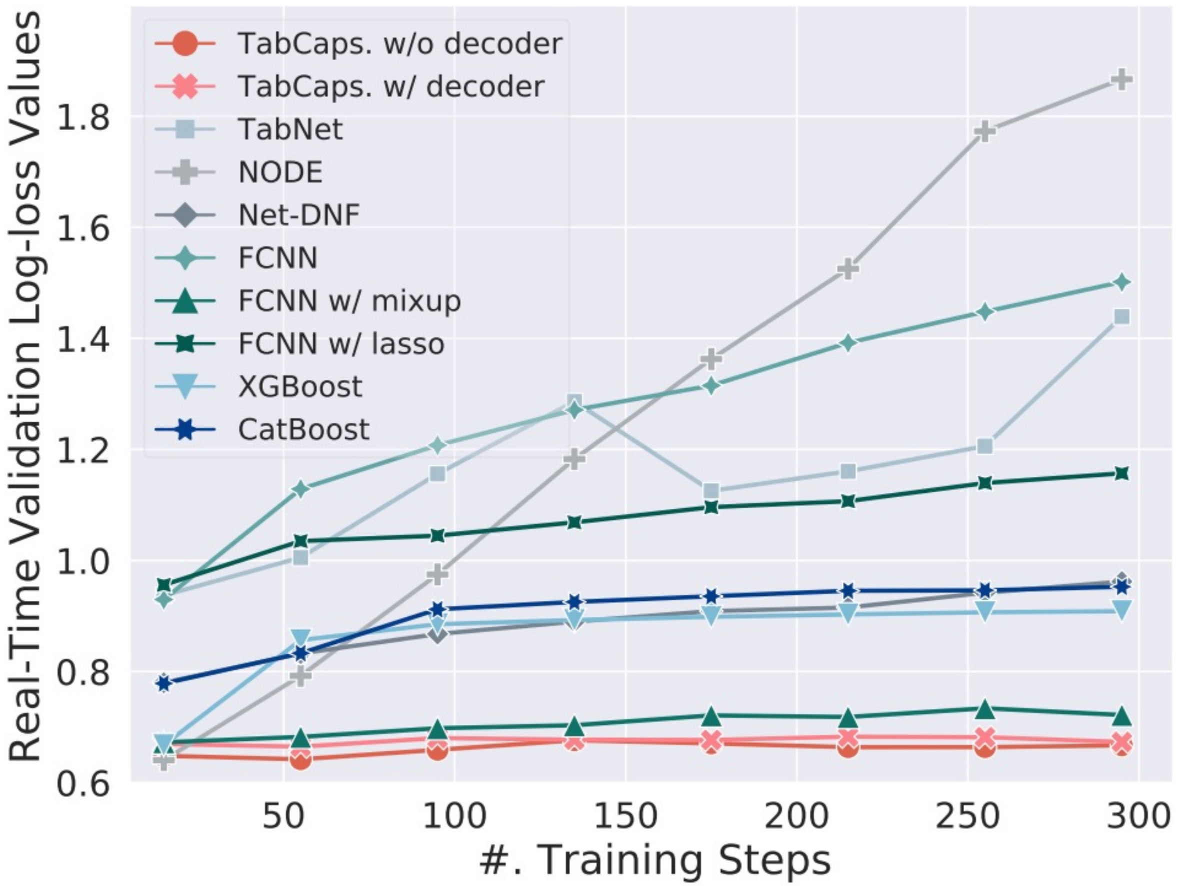
Table 2: **Extreme generalization performances.** The best and second best performances of deep learning approaches are respectively marked in **bold** and <u>underlined</u>. Note that the reported log-loss values (the lower the better) are with a 100× factor. The *Epsilon* dataset is not included due to its extremely high computation complexity in conducting t-SNE projection.

Method	**Click**	**Diabetes**	**EEG**	**Gas**	**Heart**	**Hill**	**Higgs**
Training-Test Split							
XGboost	66.070±0.03	65.886±0.09	70.654±0.02	31.504±0.04	35.650±0.01	69.657±0.09	54.557±0.04
Catboost	63.925±0.04	68.819±0.06	68.799±0.04	18.864±0.04	35.207±0.08	69.162±0.03	54.632±0.07
TabNet	115.907±0.11	225.22±0.08	79.666±0.07	158.618±0.03	44.967±0.06	89.114±0.08	55.763±0.11
Net-DNF	67.625±0.02	<u>58.792</u>±0.05	68.261±0.04	15.124±0.03	55.371±0.07	48.301±0.04	55.738±0.06
NODE	<u>63.839</u>±0.04	67.021±0.04	68.357±0.04	57.698±0.06	46.541±0.03	69.771±0.10	61.870±0.03
FT-Transformer	78.431±0.11	59.283±0.04	68.278±0.07	**6.416**±0.06	**26.132**±0.05	66.972±0.05	**53.970**±0.10
DANet-24	74.401±0.02	59.736±0.06	69.021±0.03	10.395±0.01	49.643±0.02	<u>37.976</u>±0.04	<u>54.182</u>±0.01
FCNN mixup	66.052±0.05	60.262±0.04	68.850±0.08	25.102±0.03	35.674±0.17	67.126±1e-3	55.847±0.01
FCNN lasso	106.123±3e-3	67.082±0.04	93.170±0.04	61.310±0.02	76.854±0.03	75.853±0.02	106.580±0.06
Vector CapsNet	64.724±0.05	66.009±0.02	<u>67.845</u>±0.04	163.193±0.04	60.848±0.04	64.743±0.09	62.791±0.02
TabCaps (Ours)	**63.355**±0.04	**58.409**±0.02	**67.471**±0.01	<u>8.750</u>±0.06	<u>34.503</u>±0.05	**17.887**±0.04	54.707±0.07

We biasedly split train and test sets to inspect the generalization capability.

TabCaps performs well!

Experiments



We observe that overfitting often occurs on the Click data.

We demonstrate the model's ability to resist overfitting through comparison.

THANKS

Thank you for listening!