NEURAL INFORMATION PROCESSING SYSTEMS

LAMDA Learning And Mining from DatA

Training-Free Generalization on Heterogeneous Tabular Data via Meta-Representation

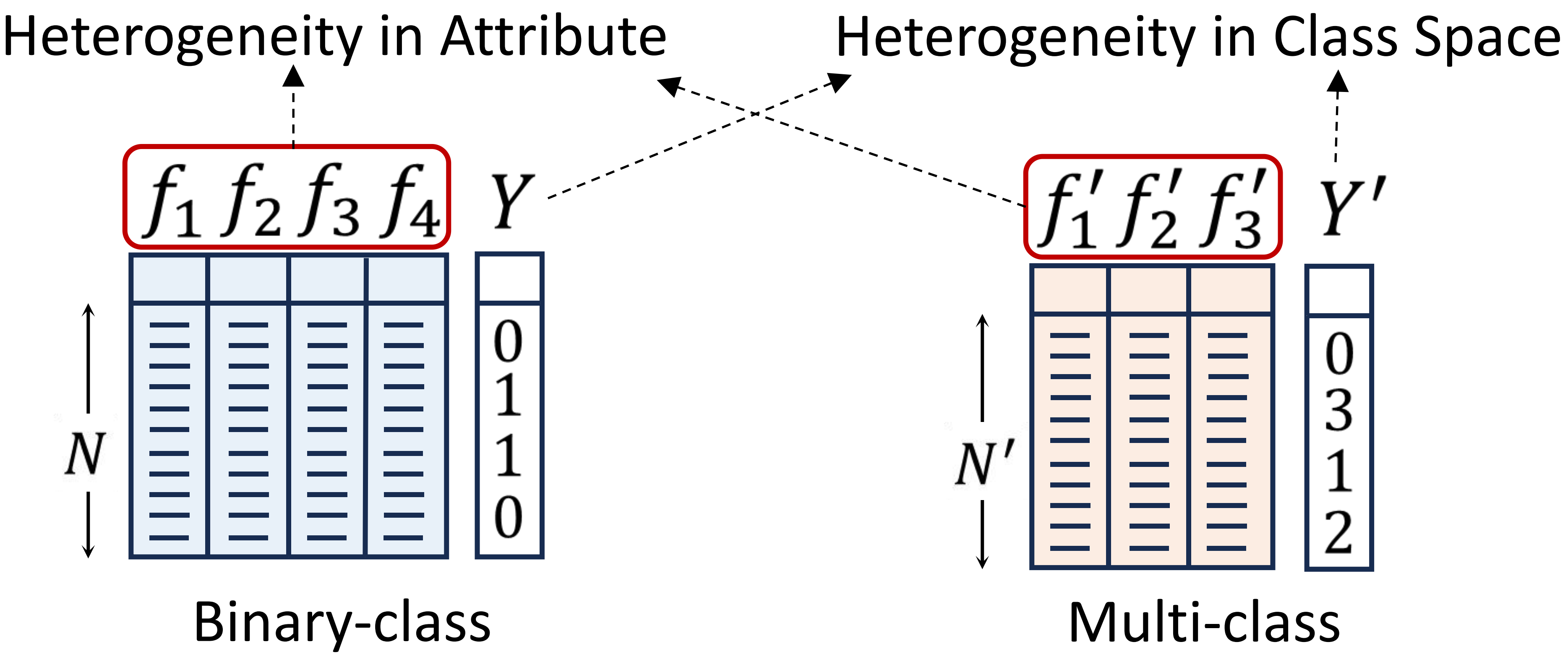
Han-Jia Ye, QiLe Zhou, De-Chuan Zhan

Nanjing University

{yehj, zhouql, zhandc}@lamda.nju.edu.cn

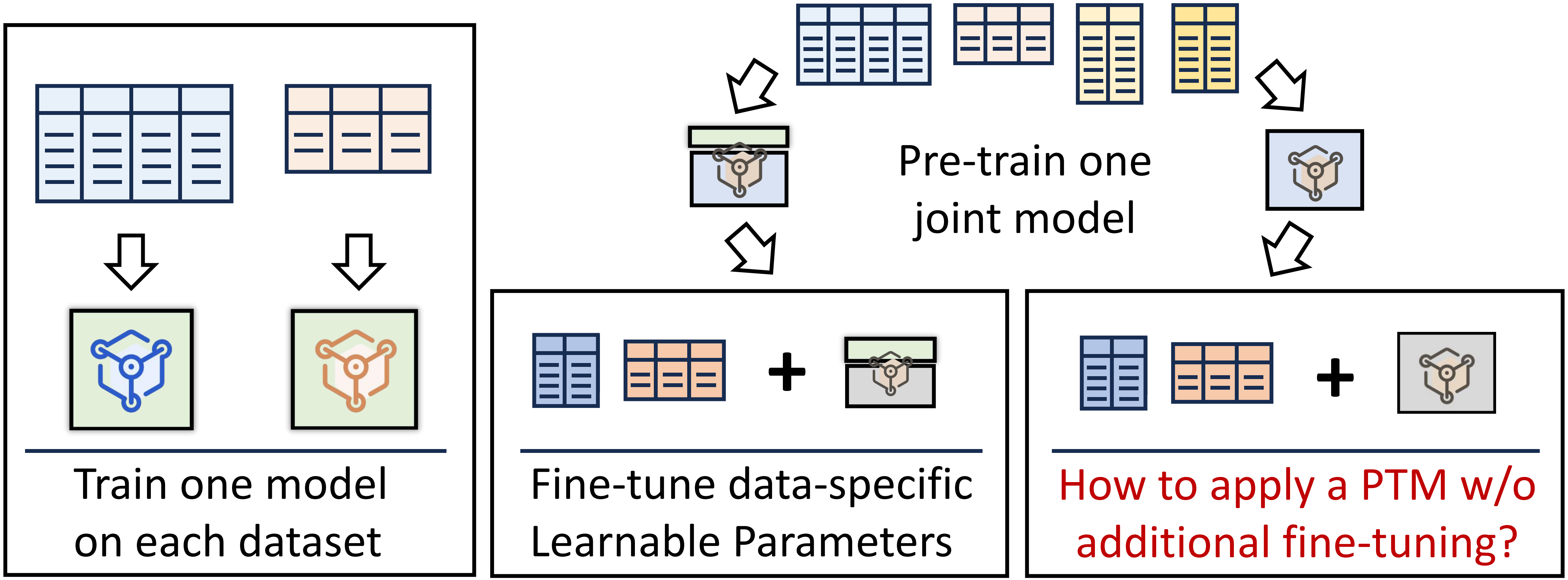
Table Representation Learning Workshop at NeurIPS 2023

Background: Learning with tabular data



The inherent heterogeneities across different tabular datasets hinder the effective sharing of knowledge.

Task: One model that generalizes on heterogeneous tabular datasets



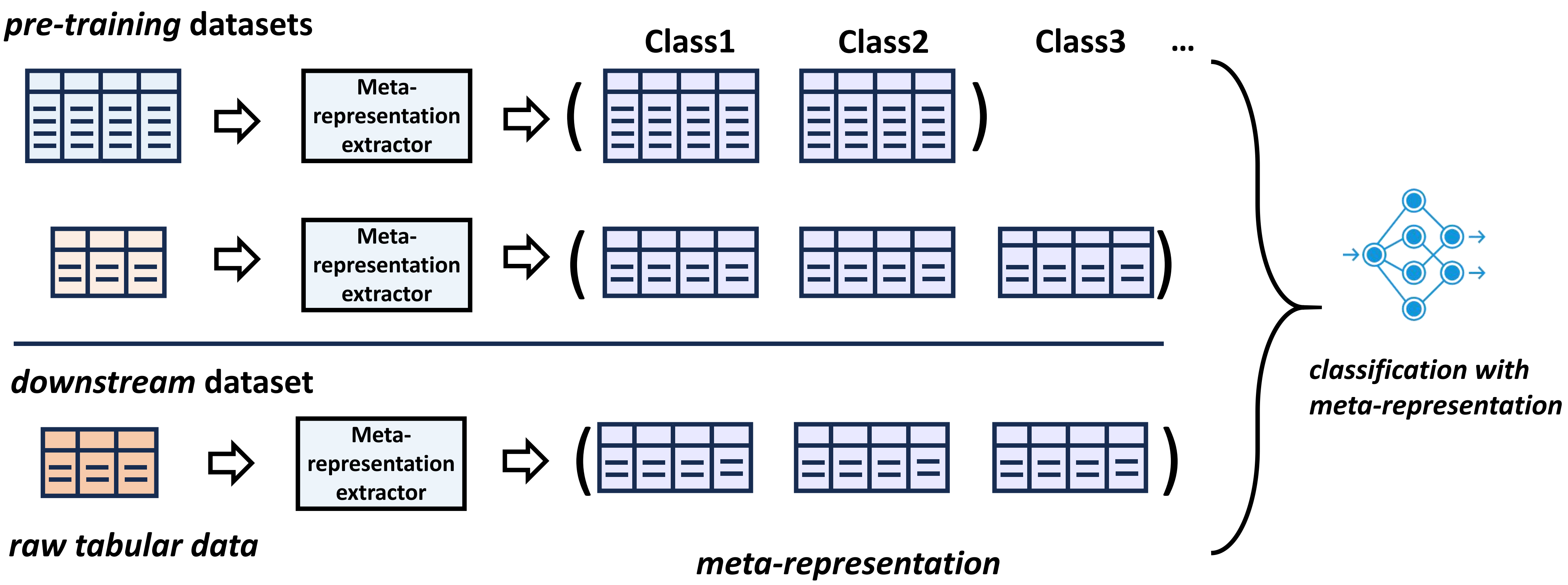
Possible Solutions

- TabPFN[1] : enable model to work on datasets with different numbers of features by zero-padding.
- LLM[2,3] : assumes the existence of attribute names, each instance could be transformed into a text.
- Dimension-invariant transformation[4,5] : transform raw data of different dimensions into consistent dimension.

[1] Tabpfn: A transformer that solves small tabular classification problems in a second. In ICLR, 2023.
[2] Tabllm: few-shot classification of tabular data with large language models. In AISTATS, 2023.
[3] Anypredict: Foundation model for tabular prediction. CoRR, 2023.
[4] Meta-learning from tasks with heterogeneous attribute spaces. In NeurIPS, 2020.
[5] Distribution embedding networks for generalization from a diverse set of classification tasks. TMLR, 2022.

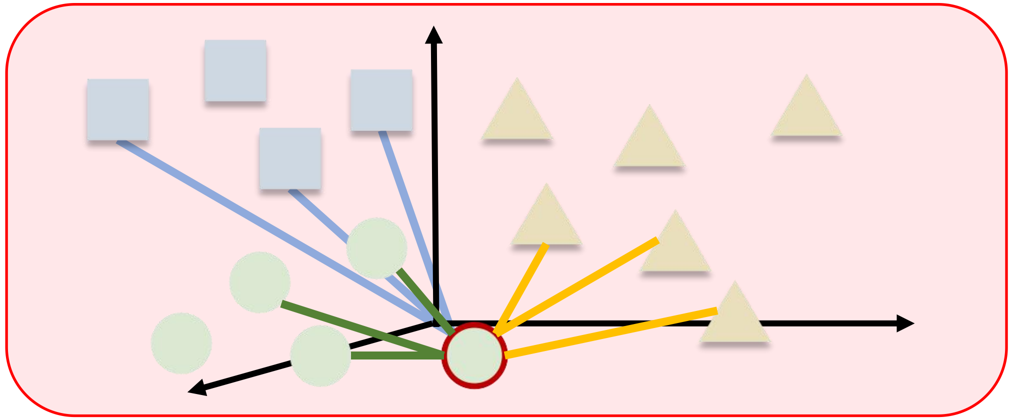
Our Solution

- Meta representation: standardize diverse datasets so that a joint deep neural network can be applied.
- Transforms any instance, irrespective of its original dimensionality, into a set of K-dimensional vectors, one for each of the C classes.

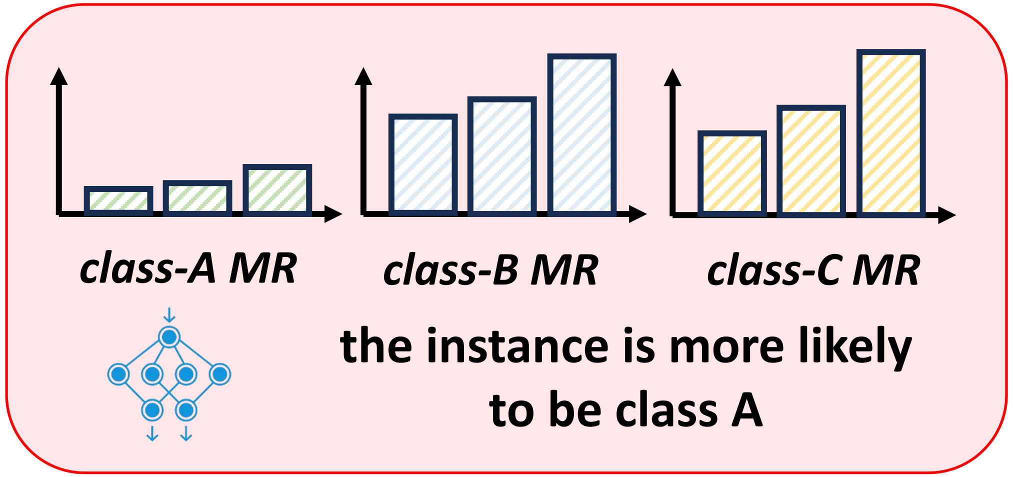


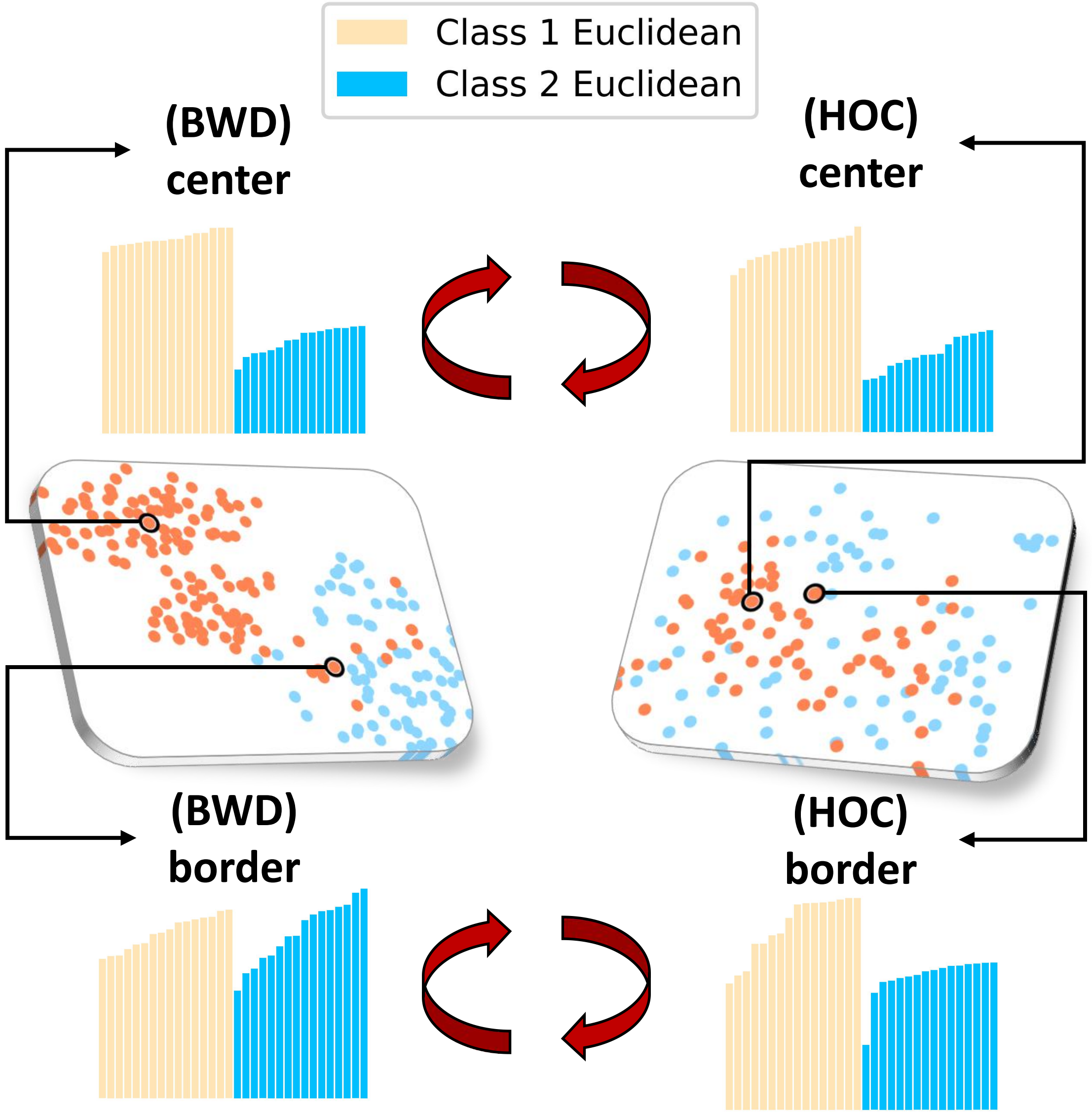
Meta-representation

Extract class-specific prototypes.



Calculate the distance to those prototypes,
Sort and select the K smallest.





TabPTM on Classification Tasks

- An adaptive metric compatible with heterogeneous tasks:

$$\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_{k=1}^{d} w_k \cdot \left|\boldsymbol{x}_{ik} - \boldsymbol{x}_{jk}\right|^p\right)^{\frac{1}{p}}, \qquad w_k = \mathrm{normalize}\big(\mathrm{MI}(\boldsymbol{X}_{:k}, \boldsymbol{Y})\big)$$

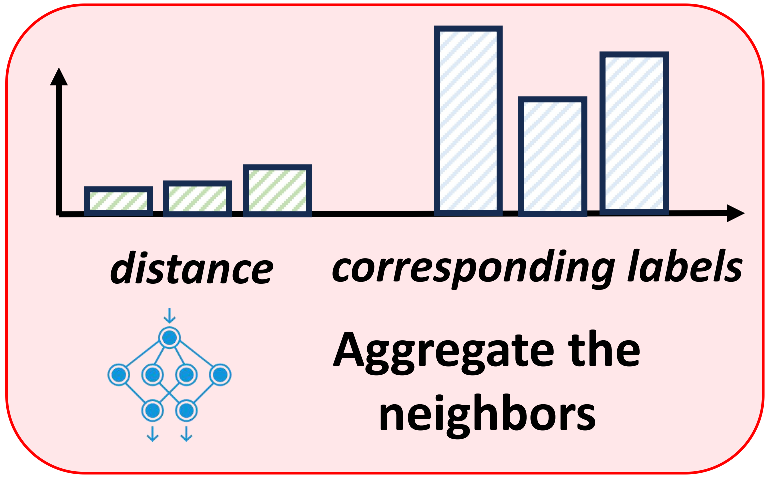
- Meta representation:

$$\phi_c(\boldsymbol{x}_i) = \left[\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_1), \ldots, \mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j), \ldots, \mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_K)\right] \in \mathbb{R}^K$$

- The architecture: outputs the class-wise classification scores
  - One MLP outputs the score. (TabPTM)
  - One MLP outputs class-wise representation, combined with Transformer. (TabPTM †)

$$[s(\boldsymbol{x}_i)_1, \ldots, s(\boldsymbol{x}_i)_C] = \mathrm{Transformer}\Big(\big[\mathbf{MLP}\big(\phi_1(\boldsymbol{x}_i)\big), \ldots, \mathbf{MLP}\big(\phi_C(\boldsymbol{x}_i)\big)\big]\Big)$$

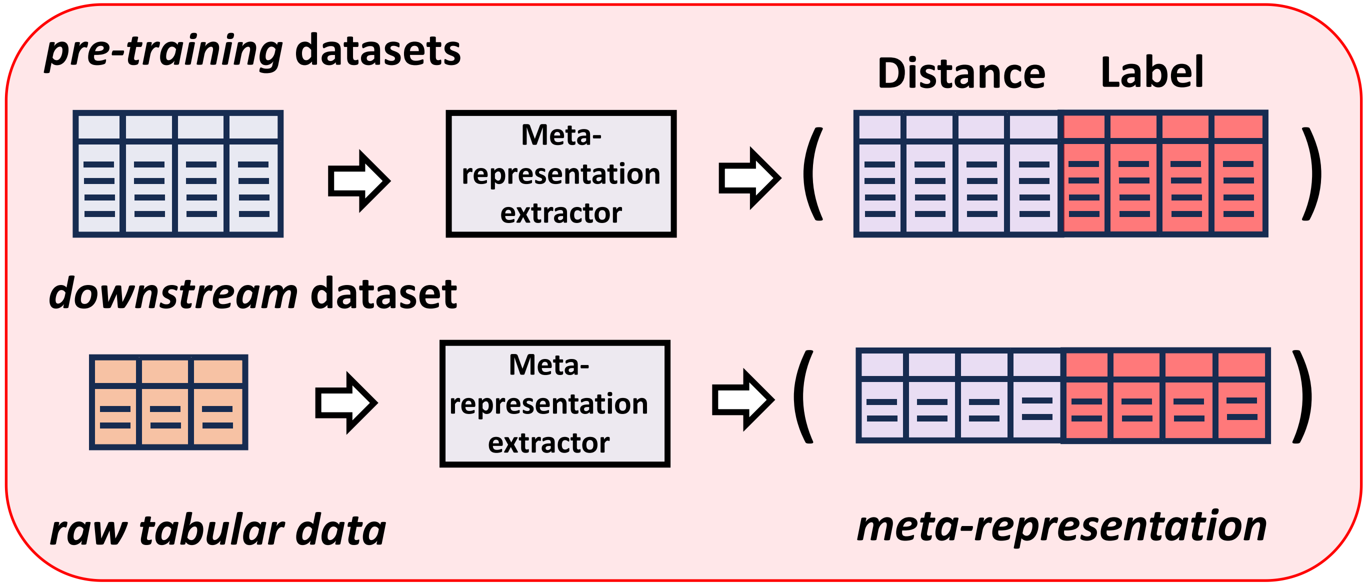
TabPTM on Regression Tasks



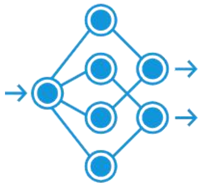
Modify the meta representation as the concatenation of distance with neighbors as well as corresponding labels.

$$\phi(\boldsymbol{x}_i) = \left[\mathrm{dist}(\boldsymbol{x}_i,\boldsymbol{x}_1),\dots,\mathrm{dist}(\boldsymbol{x}_i,\boldsymbol{x}_j),\dots,\mathrm{dist}(\boldsymbol{x}_i,\boldsymbol{x}_K), y_1,\dots,y_j,\dots y_K\right] \in \mathbb{R}^{2K}.$$

Meta-representation (for regression)



The idea could be extended to the regression scenario.



Prediction with meta-representation

Experiments

	SVM	XGBoost	MLP	FT-T	TabCaps	DANets	TabPFN	XTab	DEN	TabPTM	TabPTM†
BC	67.24	68.10	64.48	65.17	67.93	67.59	67.59	66.55	63.62	**68.79**	67.93
BW	97.14	97.23	96.64	97.07	96.36	97.64	97.14	97.50	96.71	**99.29**	98.57
BWD	97.37	96.23	96.32	97.26	97.02	**97.64**	97.15	96.14	94.74	95.61	96.49
ECD	77.78	78.89	77.41	75.19	79.63	82.96	77.78	83.07	78.89	84.07	**85.19**
HC	52.46	53.11	51.64	52.30	52.30	53.77	**53.44**	48.36	51.80	51.80	51.94
HH	81.36	83.05	82.88	78.64	81.36	**83.39**	81.02	83.22	78.47	79.66	80.51
HV	30.00	34.50	34.50	29.25	34.00	35.00	30.00	32.00	28.75	**36.50**	33.00
HOC	85.14	**87.84**	82.57	83.51	83.24	79.05	83.78	71.49	66.89	85.68	86.08
MAM	81.87	83.94	82.23	84.77	83.99	83.32	**84.61**	83.89	75.39	82.38	83.16
SPE	67.92	63.40	68.68	68.87	68.30	63.58	**70.94**	70.00	64.34	70.19	70.00
MEAN	73.82	74.63	73.74	73.20	74.41	74.39	74.35	73.22	69.96	**75.40**	75.29

	SVM	XGBoost	MLP	FT-T	TabCaps	DANets	XTab	DEN	TabPTM	TabPTM†
churn	85.25	**85.99**	85.66	85.92	85.61	85.34	85.60	72.48	85.45	85.45
crowd	42.00	**47.17**	43.53	39.80	45.83	46.57	42.73	35.13	44.47	44.97
eye	56.35	**72.36**	60.98	62.87	58.15	57.93	56.55	43.04	61.94	62.22
htru	97.96	**98.11**	98.09	98.09	98.03	97.94	98.05	94.18	97.94	97.96
jm1	81.21	**81.62**	81.03	81.87	80.90	80.82	81.03	80.49	77.53	80.88
satellite	99.31	**99.41**	99.29	99.15	99.03	99.06	99.13	99.03	97.52	99.25
MEAN	77.01	**80.78**	78.10	76.54	77.93	77.94	77.18	70.73	77.48	78.46
Time (s)	2.3×10^2	1.8×10^3	8.7×10^3	4.7×10^3	5.4×10^2	8.0×10^3	3.2×10^2	7.7×10^2	**5.7**	6.2

Discussion and Conclusion

- Utilize meta-representations to **reduce attribute heterogeneity** and enable the pre-training of a **joint model** over tabular datasets.
- Explore how to make predictions based on the meta-representations, and the pre-trained TabPTM is capable of **generalizing to unseen tabular datasets without additional training**.
- Meta-representation is validated as an effective way for tabular **classification and regression**. TabPTM shows promising capabilities in generalizing to unseen datasets.
- A trade off between Specialized model and Generalized model.