Geodesic flow kernels for semi-supervised learning on mixed-variable Tabular dataset

Yoontae Hwang, Yongjae Lee

AAAI-25

황윤태 (Hwang yoontae)

Yoontae.hwang@eng.ox.ac.uk

Machine Learning Research Group

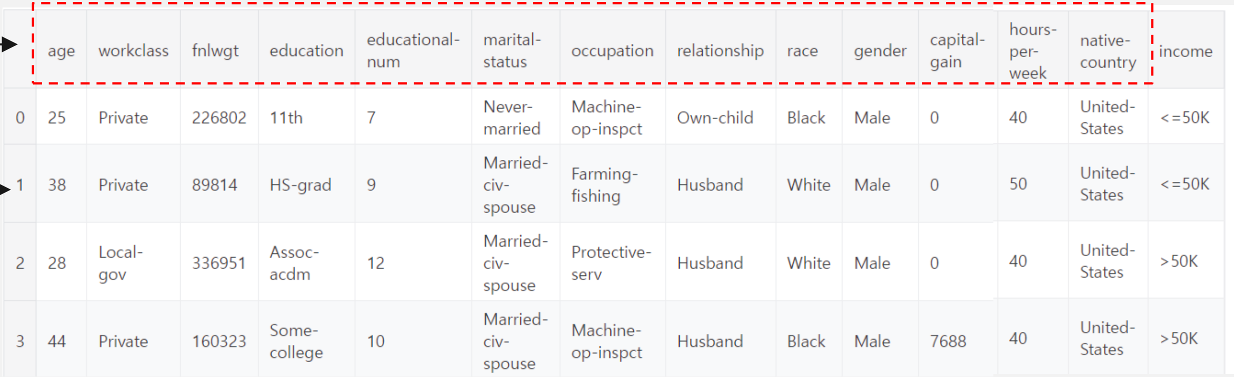
Eagle House, Department of Engineering Science

Tabular dataset

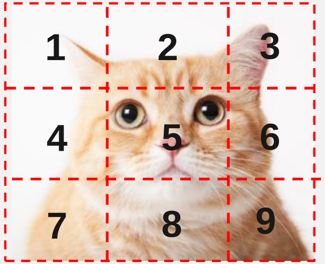
Input features →

Target labels ←

Instance →



	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	hours-per-week	native-country	income
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	40	United-States	<=50K
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	50	United-States	<=50K
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	40	United-States	>50K
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	40	United-States	>50K

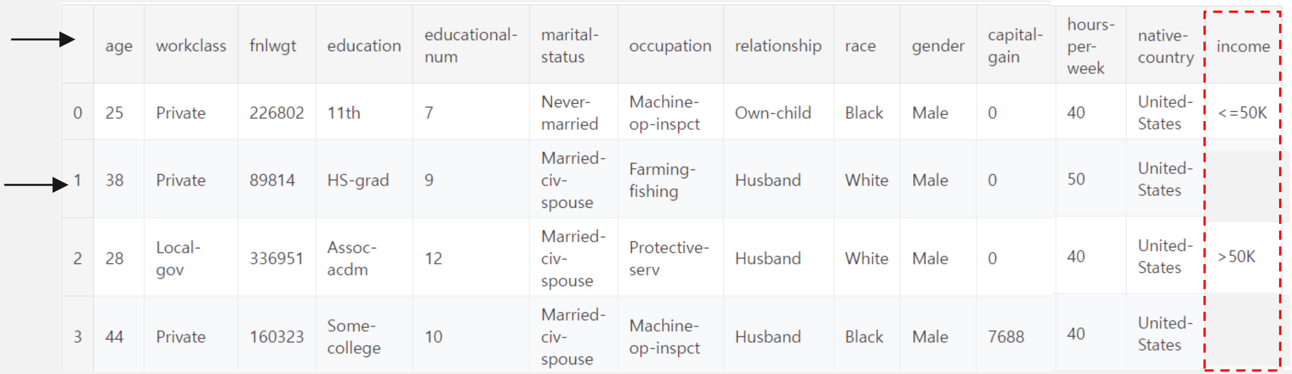


Tabular dataset

Input features

Instance

Target labels



	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	hours-per-week	native-country	income
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	40	United-States	<=50K
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	50	United-States	
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	40	United-States	>50K
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	40	United-States	

Continues variable

Categorical variable

Addressing continuous variable bias in Tabular benchmarks

- We selected **21 datasets** after carefully reviewing more than 4,000 datasets including OpenML (3,953 datasets), AMLB (71 datasets), and Grinsztajn et al.(22 datasets).

- **Preprocessing.** Datasets with more than 30% missing values were excluded. For the remaining datasets, columns with more than 30% missing values were removed. Also, redundant categorical variables, which have only one category, were removed.
- **Variable types.** In order to evaluate tabular models in a more real-world like environment, we selected datasets with both continuous and categorical variables. Surprisingly, around 60% of the entire datasets did not satisfy this condition.
- **Data distribution.** in this study, We assume that data samples are i.i.d. Hence, datasets with certain distributional structure (sequential or temporal) were excluded. Also, we eliminated datasets with too simple distributions, which can be easily predicted with high Accuracy by naïve models. Artificially generated datasets were excluded as well. Lastly, as this study focuses on classification tasks, datasets for regression tasks were not considered.
- **Dataset size.** Most previous studies did not evaluate their models with datasets of different sizes. For more comprehensive evaluation, we selected datasets with different sizes: small-sized (∼10,000 samples), medium-sized (10,000∼100,000), and large-sized (100,000∼).

- **AutoInt** (Song et al., 2019)
  - Designed for learning feature interactions automatically through self-attention layers. Primarily tested in recommendation-like tasks, where categorical embeddings dominate.
- **NODE** (Popov et al., 2020)
  - Trains ensembles of differentiable oblivious decision trees, bridging the gap between tree methods and neural nets. Evaluations have focused on a limited set of (mostly continuous) tabular benchmarks.
- **TabTransformer** (Huang et al., 2020)
  - Uses Transformer layers to capture column-wise relationships, but largely treats all features in a uniform way. Categorical variables are embedded; continuous variables typically pass through simple MLPs.
- **TabNet** (Arik and Pfister, 2021)
  - Employs sequential attention to columns and learns feature selection in a differentiable way. However, it still treats all feature types somewhat uniformly and may struggle with heavily imbalanced categorical data.
- **FT-Transformer** (Gorishniy et al., 2021)
  - Adopts a Transformer-based encoder for tabular data, again applying similar transformations across both categorical and continuous features. Shows good performance on certain benchmarks but often assumes well-behaved or purely continuous inputs.
- **TabPFN** (Hollmann et al., 2023)
  - A probabilistic approach that directly infers posterior predictive distributions for tabular classification. It shows promise but has not been deeply tested on highly mixed or categorical-heavy datasets.
- **GRANDE** (Marton et al., 2024)
  - Focuses on ensembling multiple neural networks for tabular data, achieving robust predictions. Mostly tested under fully supervised conditions and primarily on continuous or balanced scenarios(labels).

Related Works (Tabular DL Models)

Related Works (Self-/Semi-Supervised Learning for Tabular Data)

- **VIME** (Yoon et al., 2020)
  - ➢ One of the first to propose a semi-supervised approach for tabular data, employing a consistency-based loss. Uses the same noise injection strategy across all features, potentially overlooking crucial differences between categorical and continuous attributes.
- **SCARF** (Bahri et al., 2021)
  - ➢ Introduces a self-supervised objective by corrupting a random subset of features and maximizing the agreement between original and corrupted views. However, it does not distinguish between continuous and categorical variables, and tested primarily on continuous-heavy datasets.
- **SubTab** (Ucar et al., 2021)
  - ➢ Splits columns into multiple subsets and uses an autoencoder to reconstruct the data from partial subsets. Again, the corruption does not adapt to different variable types, and evaluation data mostly contained continuous features.

Common Shortcomings

- A prevalent theme is the **lack of tailored handling** for **mixed-variable** settings—continuous + categorical features—in a unified framework.
- Many models are **benchmarked on continuous-heavy data**, which may not reflect real-world tabular datasets containing complex categorical features and imbalances.
- Few methods explicitly address semi-supervised scenarios where labeled data is limited or noisy.

GFTab (Geodesic Flow Kernel on Mixed-variable Tabular data)

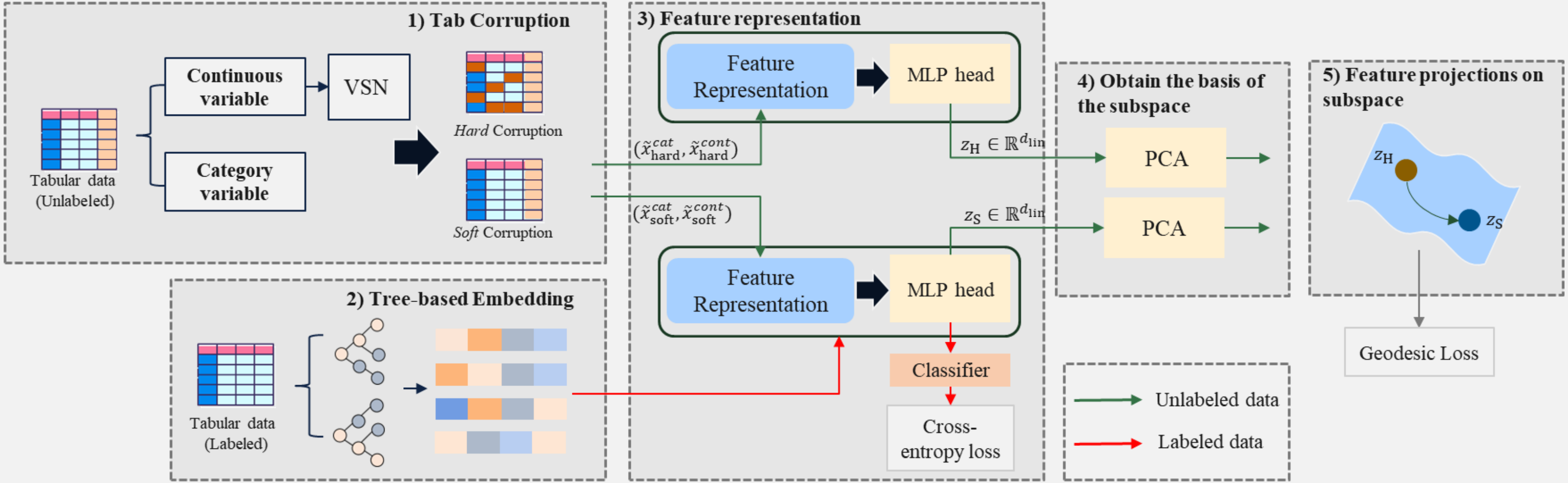


Figure 1 The proposed model (GKSMT) is a semi-supervised learning framework specifically designed for handling tabular data.

- The model proposed in this study aims to obtain **an invariant representation** by considering both ***continuous*** and ***categorical*** variables through a (1) corruption method and (2) geodesic kernel flow, reflecting the basic structure of the data from the corrupted representation.
- This technique is particularly useful for dealing with diverse types of data, ensuring that the model can effectively learn from and represent both numerical (continuous) and non-numerical (categorical) elements.

Methodology : Overview

- We have N_l labeld samples $D_l = \{x^i, y^i\}_{i=1}^{N_l} \subseteq \mathbb{R}^{M+1}$ and N_u un-labeled samples $D_u = \{x^i\}_{i=1}^{N_u} \subseteq \mathbb{R}^M,$ where $N_u \gg N_l$.
- Here, $x^i \subseteq \mathbb{R}^M$ is an example, $y^i \subseteq \mathbb{R}$ is the label of x^i.
- In this work, x^i can be decomposed into continuous variable $x^{i,cont} \in \mathbb{R}^{M_{cont}}$ and categorical variable $x^{i,cat} \in \mathbb{R}^{M_{cat}}$.

Corruption method – continuous variable

Let P_m^{cont} be the uniform distribution $X_m^{cont} = \{x_m^{i,cont} : x^{i,cont} \in X^{cont}\}$, where $x_m^{i,cont}$ denotes the m-th coordinate of $x^{i,cont}$.

- $x^{i,cont} = \mathrm{VSN}(\mathrm{x}^{i,cont})$ where VSN is Variable Selection Network.

Here, we define permutation matrix, **P** size $d^{\text{cont}} \times d^{\text{cont}}$.

- $\tilde{x}^{i,cont}_{\text{soft}} = \lambda x^{i,cont} + (1\text{-}\lambda)\hat{x}^{i,cont}\mathbf{P}$
- $\tilde{x}^{i,cont}_{\text{hard}} = (1-\lambda)x^{i,cont} + \lambda\hat{x}^{i,cont}\mathbf{P}$ Where $\hat{x}^{i,cont} = \left[\hat{x}_m^{i,cont}\right]_{m=1}^{M_{cont}}$ and $\hat{x}_m^{i,cont} \sim P_m^{cont}$

Methodology : Corruption method (continuous variable)

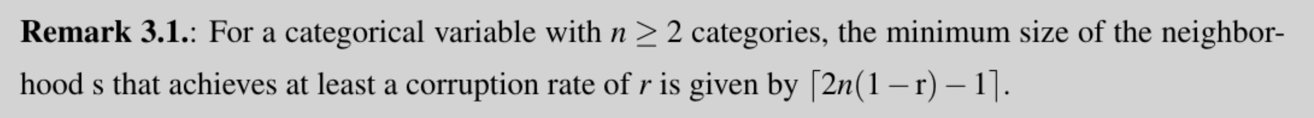
- We have N_l labeld samples $D_l = \{x^i, y^i\}_{i=1}^{N_l} \subseteq \mathbb{R}^{M+1}$ and N_u un-labeled samples $D_u = \{x^i\}_{i=1}^{N_u} \subseteq \mathbb{R}^M$, where $N_u \gg N_l$.
- Here, $x^i \subseteq \mathbb{R}^M$ is an example, $y^i \subseteq \mathbb{R}$ is the label of x^i.
- In this work, x^i can be decomposed into continuous variable $x^{i,cont} \in \mathbb{R}^{M_{cont}}$ and categorical variable $x^{i,cat} \in \mathbb{R}^{M_{cat}}$.

Feature corruption – Categorical variable

Let $k = [k_1, \ldots, k_{M_{cat}}]^T$ be the mask vector where k_m's are independently sampled from over the set $\mathbb{Z} \cap [-s_m, s_m] \backslash \{0\}$ with equal probability. Where s_m is size of the neighborhood at k_m. We can find the size of neighborhood for which the corruption rate is $r\%$ greater for a categorical variable via **Remark 1**.

- $\tilde{x}^{cat}_{\text{soft}} = x^{cat} + \text{nh}_r(k)$
- $\tilde{x}^{cat}_{\text{hard}} = x^{cat} + \text{nh}_{s=1}(k)$

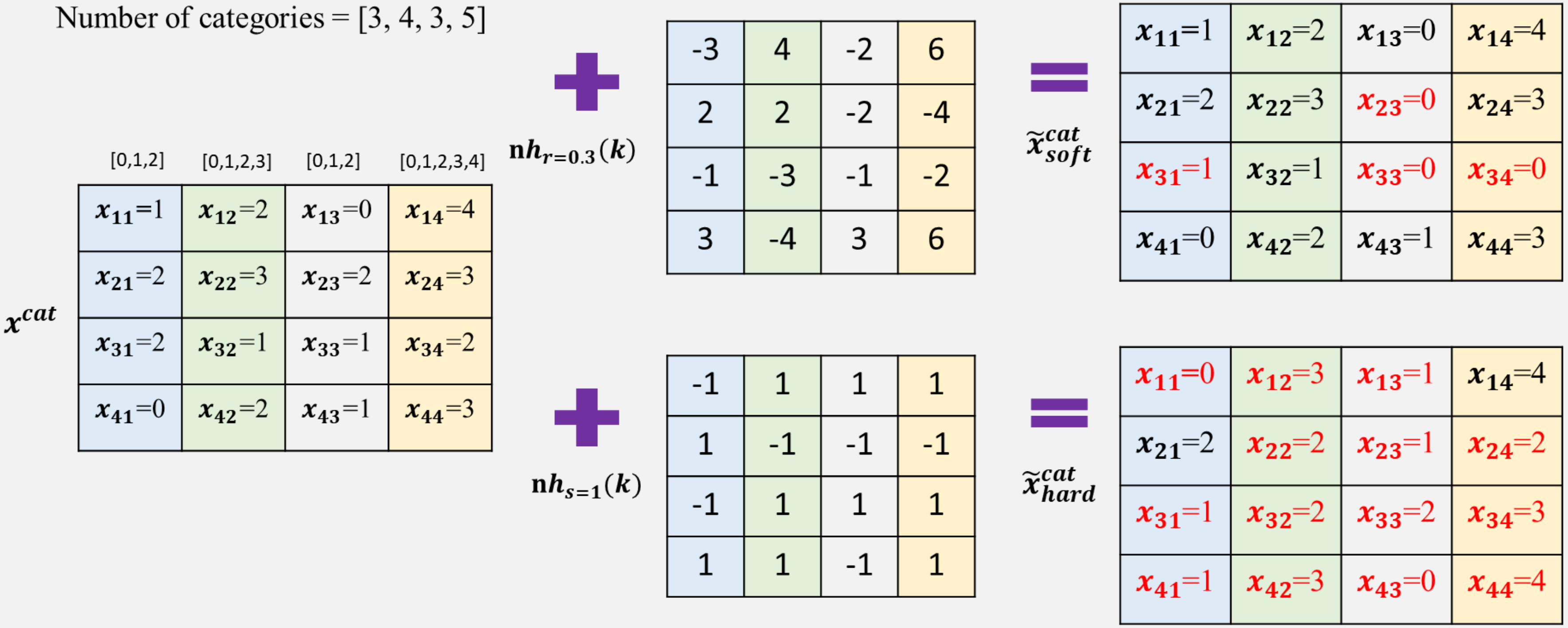
Notice that if the corrupted value $\tilde{x}^{i,cat}_r$ falls outside the range of the variable, the mask k_m is set to zero.



Remark 3.1.: For a categorical variable with $n \geq 2$ categories, the minimum size of the neighborhood s that achieves at least a corruption rate of r is given by $\lceil 2n(1-\mathrm{r})-1 \rceil$.

Methodology : Corruption method (categorical variable)

Number of categories = [3, 4, 3, 5]



x^{cat}

[0,1,2]	[0,1,2,3]	[0,1,2]	[0,1,2,3,4]
x_{11}=1	x_{12}=2	x_{13}=0	x_{14}=4
x_{21}=2	x_{22}=3	x_{23}=2	x_{24}=3
x_{31}=2	x_{32}=1	x_{33}=1	x_{34}=2
x_{41}=0	x_{42}=2	x_{43}=1	x_{44}=3

+

$\mathbf{n}h_{r=0.3}(k)$

-3	4	-2	6
2	2	-2	-4
-1	-3	-1	-2
3	-4	3	6

=

$\tilde{x}^{cat}_{soft}$

x_{11}=1	x_{12}=2	x_{13}=0	x_{14}=4
x_{21}=2	x_{22}=3	x_{23}=0	x_{24}=3
x_{31}=1	x_{32}=1	x_{33}=0	x_{34}=0
x_{41}=0	x_{42}=2	x_{43}=1	x_{44}=3

+

$\mathbf{n}h_{s=1}(k)$

-1	1	1	1
1	-1	-1	-1
-1	1	1	1
1	1	-1	1

=

$\tilde{x}^{cat}_{hard}$

x_{11}=0	x_{12}=3	x_{13}=1	x_{14}=4
x_{21}=2	x_{22}=2	x_{23}=1	x_{24}=2
x_{31}=1	x_{32}=2	x_{33}=2	x_{34}=3
x_{41}=1	x_{42}=3	x_{43}=0	x_{44}=4

Methodology : Corruption method (categorical variable)

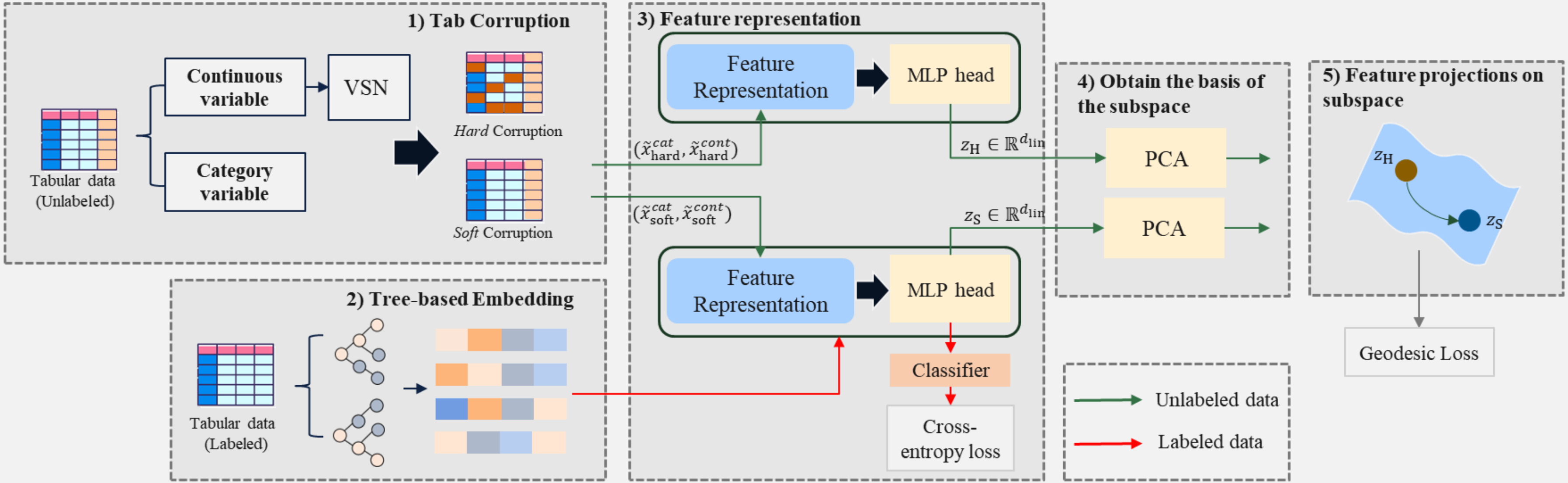


Figure 1 The proposed model (GKSMT) is a semi-supervised learning framework specifically designed for handling tabular data.

B

Obtain the basis of the subspace

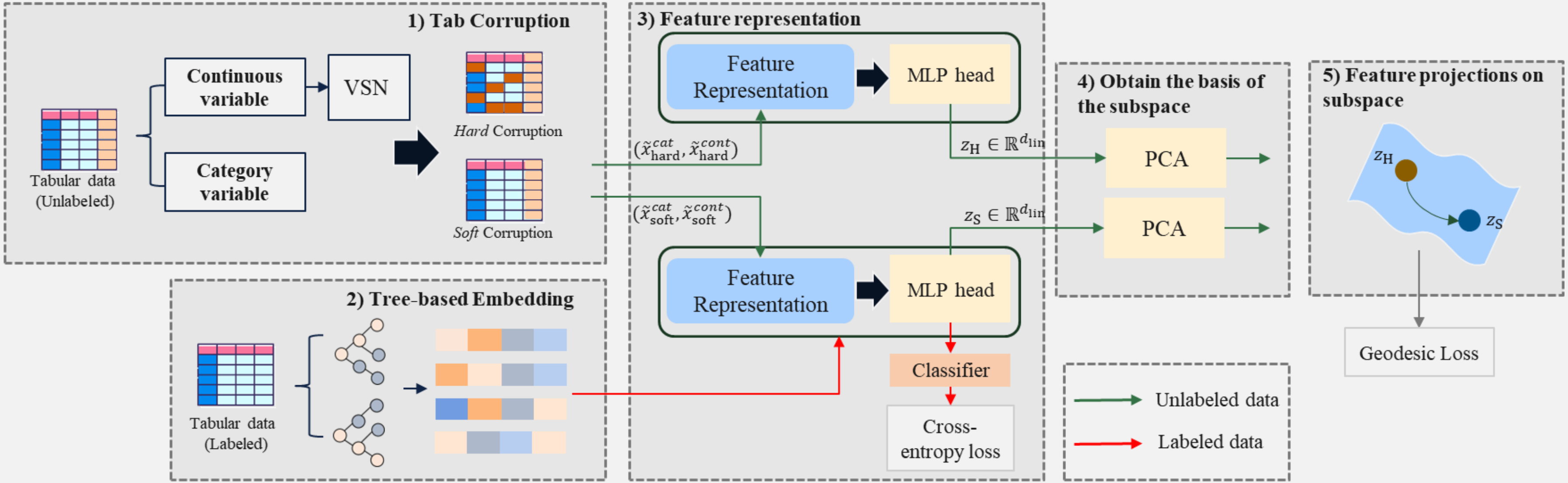
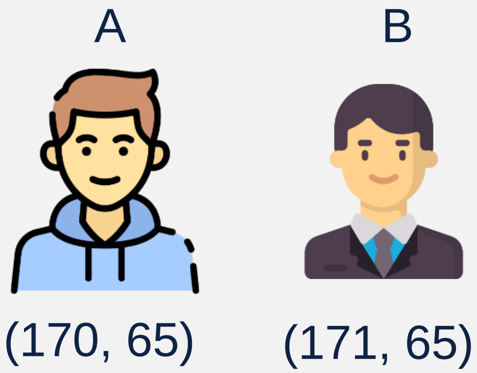


Figure 1 The proposed model (GKSMT) is a semi-supervised learning framework specifically designed for handling tabular data.

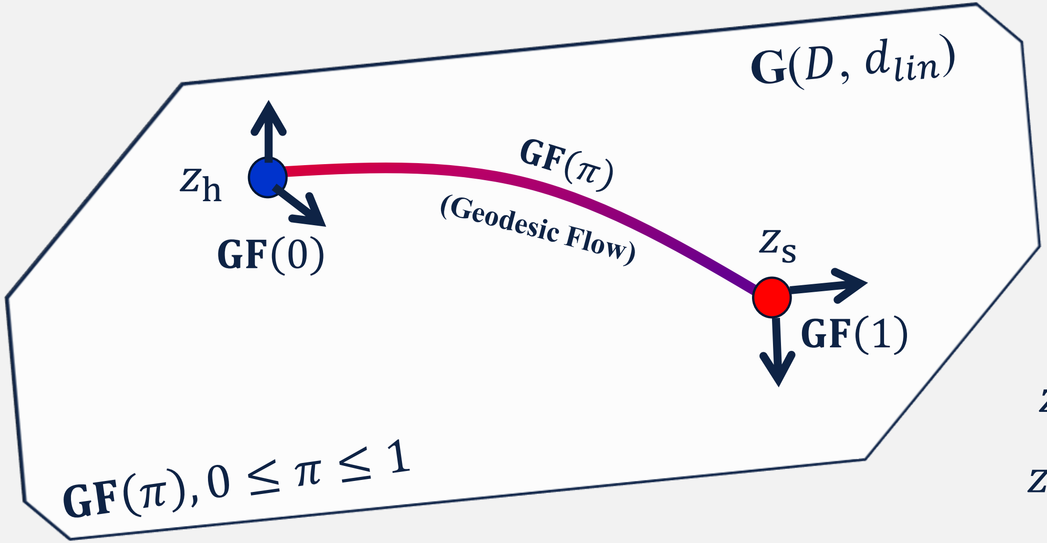


Obtain the basis of the subspace

A.1. Construct Geodesic Flow

We use the notion of subspaces to incorporate a collection of features derived from both soft and hard representation. To do this, we model the by low-dimensional subspace with basis $\boldsymbol{P} \in \mathbb{R}^{d_{lin} \times D}$.

DEF 1 (Grassmannian) Grassmannian $\mathbf{G}(D, d_{lin})$, which is the collection of all D-dimensional linear subspaces $\mathbb{R}^{d_{lin} \times D}$, is a smooth Riemannian manifold. Also, an element $\boldsymbol{P}$ of $\mathbf{G}(D, d_{lin})$ can be specified by a basis. That is $d_{lin} \times D$ matrix with orthogonal columns.



$$z_{\mathrm{s}} = f_{\mathrm{enc}}(\tilde{x}_{\mathrm{soft}}^{\mathrm{cat}}, \tilde{x}_{\mathrm{soft}}^{\mathrm{cont}})$$

$$z_{\mathrm{h}} = f_{\mathrm{enc}}(\tilde{x}_{\mathrm{hard}}^{\mathrm{cat}}, \tilde{x}_{\mathrm{hard}}^{\mathrm{cont}})$$

DEF 2. (Geodesic flow) Let $\boldsymbol{P_s}$ and $\boldsymbol{P_h}$ denote the sets of basis vector for the subspaces corresponding to the soft and hard representations, respectively. The geodesic flow between $\boldsymbol{P_s}$ and $\boldsymbol{P_h}$ denoted $\mathbf{GF} : \pi \in [0,1] \rightarrow \mathbf{GF}(\pi) \in \mathbf{G}(\mathrm{D}, \mathrm{d}_{\boldsymbol{lin}})$. Then we can rewrite the geodesic flow:

$$\mathbf{GF}(\pi)=[\boldsymbol{P}_h\ \boldsymbol{R}_{\boldsymbol{h}}]\begin{bmatrix}\boldsymbol{U}_1\boldsymbol{\Gamma}(\pi) & 0\\ 0 & -\boldsymbol{U}_2\ \boldsymbol{\Sigma}(\pi)\end{bmatrix} = \boldsymbol{P}_h\boldsymbol{U}_1\boldsymbol{\Gamma}(\pi) - \boldsymbol{R}_h\boldsymbol{U}_2\boldsymbol{\Sigma}(\pi)$$

(1)

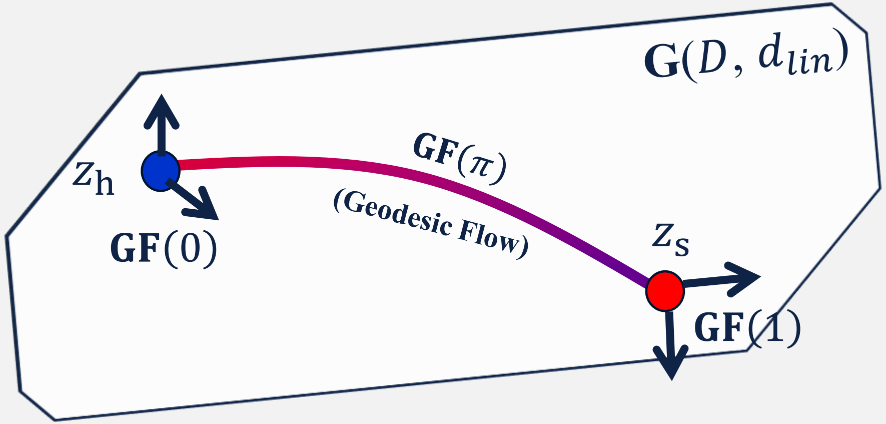
Here, the $\boldsymbol{R}_h \in \mathbb{R}^{d_{lin}\times(d_{lin}-D)}$ is the orthogonal complement of $\boldsymbol{P}_h$, that is $\boldsymbol{R}_h^\top \boldsymbol{P}_h$=0, and the $\boldsymbol{\Gamma}$ and $\boldsymbol{\Sigma}$ diagonal matrices. Also, The $\boldsymbol{U}_1 \in \mathbb{R}^{D\times D}$ and $\boldsymbol{U}_2 \in \mathbb{R}^{(D-d_{lin})\times D}$ are orthonormal matrices.

DEF 3. (Geodesic Flow Kernel) Let $\boldsymbol{z_h}$=$\boldsymbol{f}_{enc}\left(\tilde{x}_{\mathrm{soft}}^{cat}, \tilde{x}_{\mathrm{soft}}^{cont}\right)$ and $\boldsymbol{z_s} = \boldsymbol{f}_{enc}(\tilde{x}_{\mathrm{Hard}}^{cat}, \tilde{x}_{\mathrm{Hard}}^{cont})$ are the encoded representation, respectively. Then, the geodesic flow kernel is defined as :

$$z_h^\top \boldsymbol{A}\, z_s = \int_0^1 (\mathbf{GF}\,(\pi)^\top z_h)^\top (\mathbf{GF}(\pi)^\top z_s)\; d\pi$$

Where $\boldsymbol{A} = \int_0^1 \mathbf{GF}(\pi)\mathbf{GF}(\pi)^\top d\pi.$

Geodesic Flow Kernel



$$< x_h, x_h >= \int_0^1 \left(\mathrm{GF}(\pi)^\mathsf{T} z_s\right)^\mathsf{T} \left(\mathrm{GF}(\pi)^\mathsf{T} z_h\right) d\pi = z_s^\mathsf{T} A z_h$$

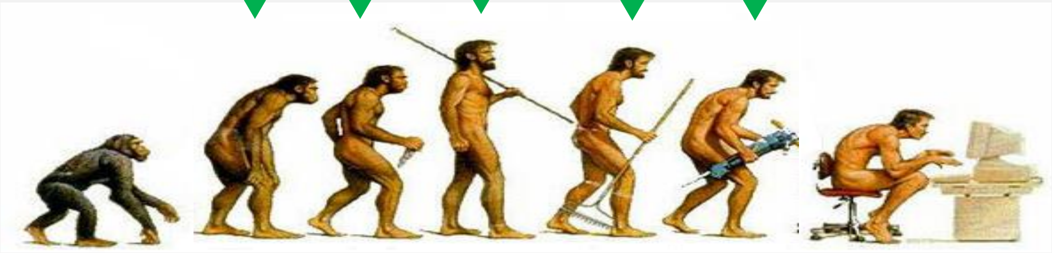
Notice 1 : that we can calculate A in closed-form. (A is a d by d psd.)

Notice 2 : A is the matrix that defines the manifold structure between features of two vector.

$$x_s = \boxed{[\mathrm{GF}(0)^\mathsf{T} z_s,} \dots, \mathrm{GF}(\pi)^\mathsf{T} z_s, \dots \boxed{, \mathrm{GF}(1)^\mathsf{T} z_s]}$$
$$x_h = \boxed{[\mathrm{GF}(0)^\mathsf{T} z_h,} \dots, \mathrm{GF}(\pi)^\mathsf{T} z_h, \dots \boxed{, \mathrm{GF}(1)^\mathsf{T} z_h]}$$

More similar to hard $\mathbf{GF}(0)$ → $\mathbf{GF}(1)$ More similar to soft

Geodesic Flow Example



Geodesic loss

$$Loss = 1 - \frac{z_s^T A z_h}{||A^{0.5} z_s|| \, ||A^{0.5} z_h||}$$

Geodesic Flow Kernel

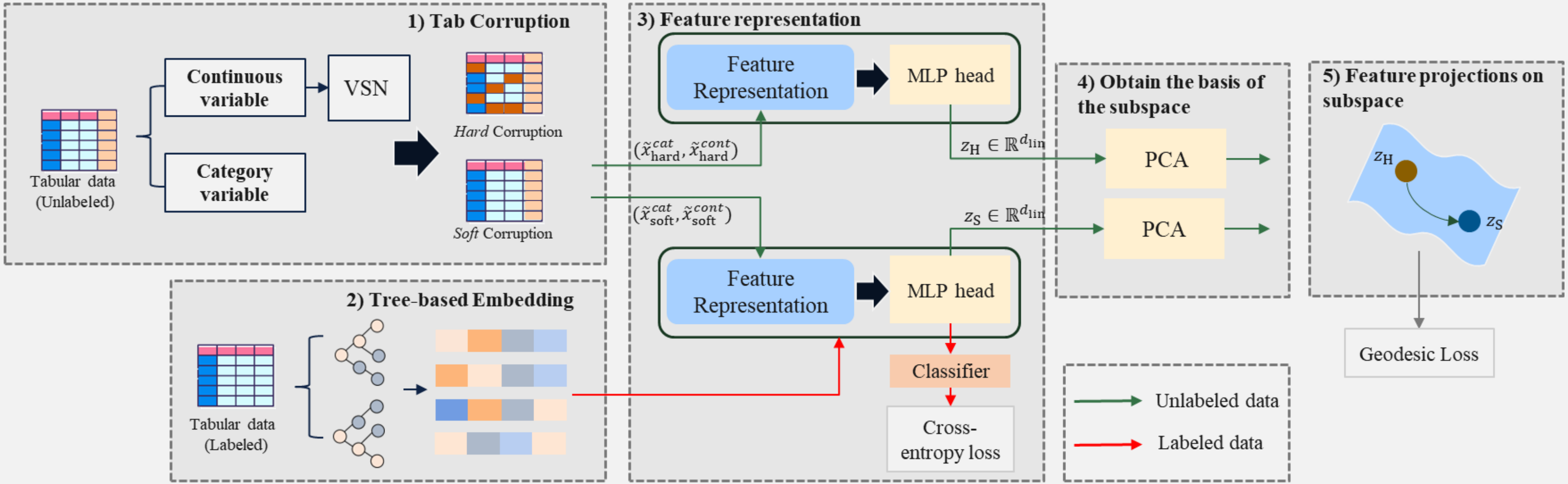


Figure 1 The proposed model (GKSMT) is a semi-supervised learning framework specifically designed for handling tabular data.

$$\mathcal{L}_{\text{GFTab}} = \mathcal{L}_{\text{sim}} + \beta \mathcal{L}_{\text{ce}}$$

Methodology

Experiment Settings

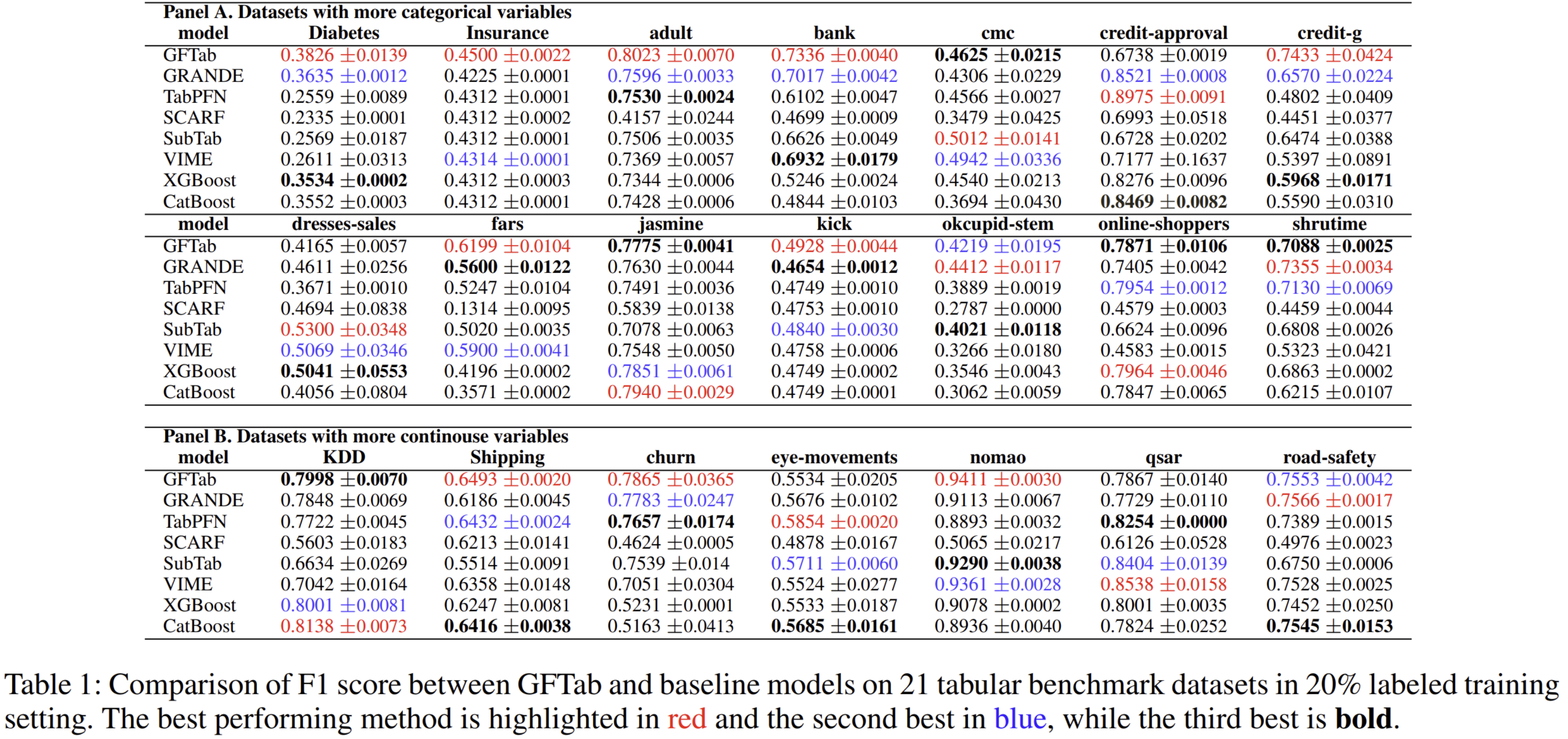
Baseline model

GFTab	(**Ours**)	GRANDE	(ICLR, 2024)
SCARF	(ICLR 2022 Spotlight)	TabPFN	(ICLR 2023)
SubTab	(NeurIPS 2021)	XGBoost	
VIME	(NeurIPS 2020)	Caboost	

For XGBoost and CatBoost, we optimized hyperparameters using Optuna, conducting 250 experiments for each model with a search space consistent with GRANDE. Deep learning models used settings from their respective papers for finding hyper-parameters, while SCARF and VIME had arbitrarily set ranges detailed in the appendix.

Experiment setting

All experiments (GFTab and all baseline models) were **repeated three times**.
All the models were trained with **labels for 10% and 20%** of the entire data samples.
Additionally, there was an experiment conducted with **20% label noise** introduced only in the training data to assess the models' robustness to noisy labels.



Panel A. Datasets with more categorical variables

model	Diabetes	Insurance	adult	bank	cmc	credit-approval	credit-g
GFTab	0.3826 ±0.0139	0.4500 ±0.0022	0.8023 ±0.0070	0.7336 ±0.0040	**0.4625 ±0.0215**	0.6738 ±0.0019	0.7433 ±0.0424
GRANDE	0.3635 ±0.0012	0.4225 ±0.0001	0.7596 ±0.0033	0.7017 ±0.0042	0.4306 ±0.0229	0.8521 ±0.0008	0.6570 ±0.0224
TabPFN	0.2559 ±0.0089	0.4312 ±0.0001	**0.7530 ±0.0024**	0.6102 ±0.0047	0.4566 ±0.0027	0.8975 ±0.0091	0.4802 ±0.0409
SCARF	0.2335 ±0.0001	0.4312 ±0.0002	0.4157 ±0.0244	0.4699 ±0.0009	0.3479 ±0.0425	0.6993 ±0.0518	0.4451 ±0.0377
SubTab	0.2569 ±0.0187	0.4312 ±0.0001	0.7506 ±0.0035	0.6626 ±0.0049	0.5012 ±0.0141	0.6728 ±0.0202	0.6474 ±0.0388
VIME	0.2611 ±0.0313	0.4314 ±0.0001	0.7369 ±0.0057	**0.6932 ±0.0179**	0.4942 ±0.0336	0.7177 ±0.1637	0.5397 ±0.0891
XGBoost	**0.3534 ±0.0002**	0.4312 ±0.0003	0.7344 ±0.0006	0.5246 ±0.0024	0.4540 ±0.0213	0.8276 ±0.0096	**0.5968 ±0.0171**
CatBoost	0.3552 ±0.0003	0.4312 ±0.0001	0.7428 ±0.0006	0.4844 ±0.0103	0.3694 ±0.0430	**0.8469 ±0.0082**	0.5590 ±0.0310

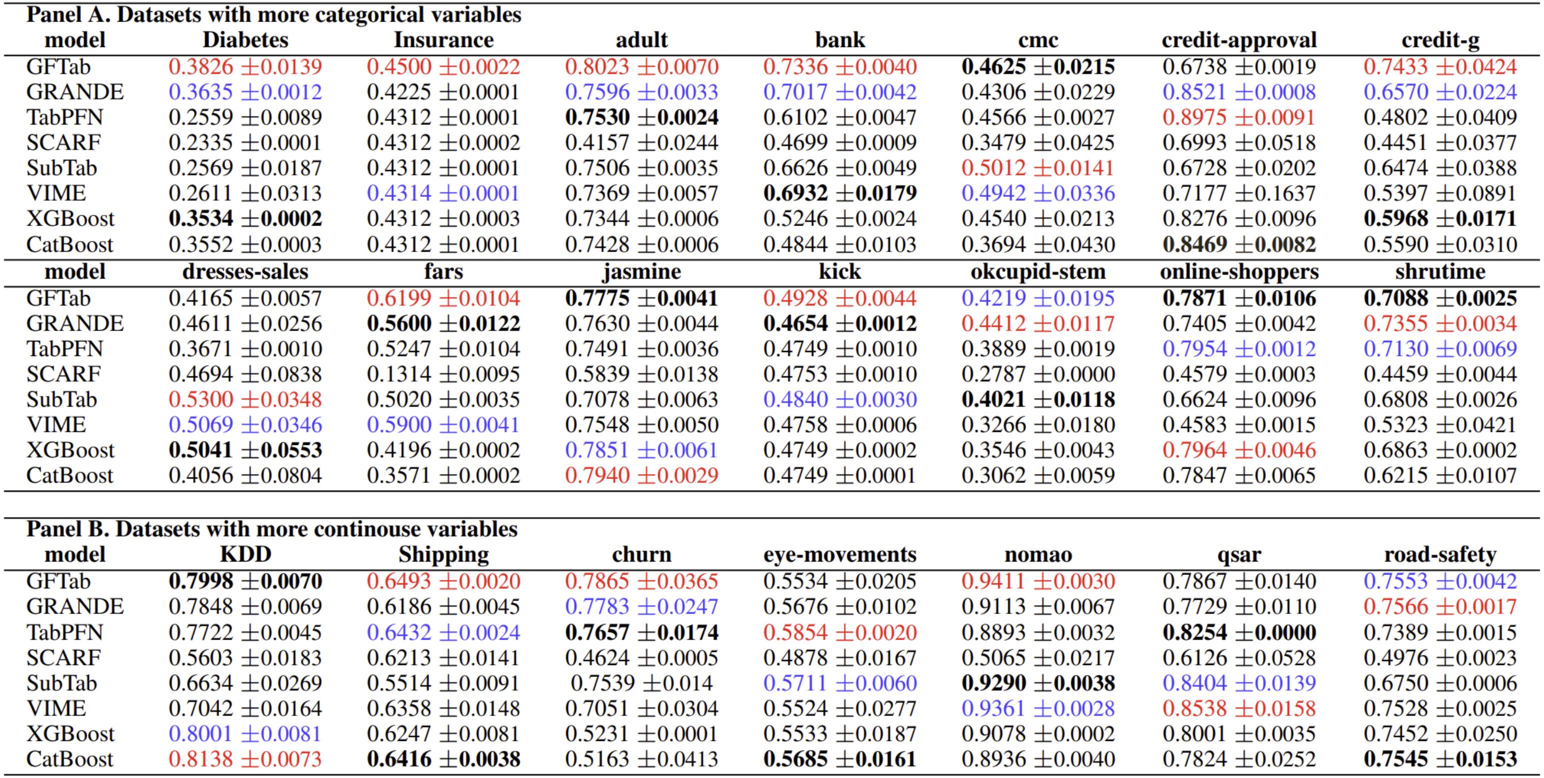
model	dresses-sales	fars	jasmine	kick	okcupid-stem	online-shoppers	shrutime
GFTab	0.4165 ±0.0057	0.6199 ±0.0104	**0.7775 ±0.0041**	0.4928 ±0.0044	0.4219 ±0.0195	**0.7871 ±0.0106**	**0.7088 ±0.0025**
GRANDE	0.4611 ±0.0256	**0.5600 ±0.0122**	0.7630 ±0.0044	**0.4654 ±0.0012**	0.4412 ±0.0117	0.7405 ±0.0042	0.7355 ±0.0034
TabPFN	0.3671 ±0.0010	0.5247 ±0.0104	0.7491 ±0.0036	0.4749 ±0.0010	0.3889 ±0.0019	0.7954 ±0.0012	0.7130 ±0.0069
SCARF	0.4694 ±0.0838	0.1314 ±0.0095	0.5839 ±0.0138	0.4753 ±0.0010	0.2787 ±0.0000	0.4579 ±0.0003	0.4459 ±0.0044
SubTab	0.5300 ±0.0348	0.5020 ±0.0035	0.7078 ±0.0063	0.4840 ±0.0030	**0.4021 ±0.0118**	0.6624 ±0.0096	0.6808 ±0.0026
VIME	0.5069 ±0.0346	0.5900 ±0.0041	0.7548 ±0.0050	0.4758 ±0.0006	0.3266 ±0.0180	0.4583 ±0.0015	0.5323 ±0.0421
XGBoost	**0.5041 ±0.0553**	0.4196 ±0.0002	0.7851 ±0.0061	0.4749 ±0.0002	0.3546 ±0.0043	0.7964 ±0.0046	0.6863 ±0.0002
CatBoost	0.4056 ±0.0804	0.3571 ±0.0002	0.7940 ±0.0029	0.4749 ±0.0001	0.3062 ±0.0059	0.7847 ±0.0065	0.6215 ±0.0107

Panel B. Datasets with more continouse variables

model	KDD	Shipping	churn	eye-movements	nomao	qsar	road-safety
GFTab	**0.7998 ±0.0070**	0.6493 ±0.0020	0.7865 ±0.0365	0.5534 ±0.0205	0.9411 ±0.0030	0.7867 ±0.0140	0.7553 ±0.0042
GRANDE	0.7848 ±0.0069	0.6186 ±0.0045	0.7783 ±0.0247	0.5676 ±0.0102	0.9113 ±0.0067	0.7729 ±0.0110	0.7566 ±0.0017
TabPFN	0.7722 ±0.0045	0.6432 ±0.0024	**0.7657 ±0.0174**	0.5854 ±0.0020	0.8893 ±0.0032	**0.8254 ±0.0000**	0.7389 ±0.0015
SCARF	0.5603 ±0.0183	0.6213 ±0.0141	0.4624 ±0.0005	0.4878 ±0.0167	0.5065 ±0.0217	0.6126 ±0.0528	0.4976 ±0.0023
SubTab	0.6634 ±0.0269	0.5514 ±0.0091	0.7539 ±0.014	0.5711 ±0.0060	**0.9290 ±0.0038**	0.8404 ±0.0139	0.6750 ±0.0006
VIME	0.7042 ±0.0164	0.6358 ±0.0148	0.7051 ±0.0304	0.5524 ±0.0277	0.9361 ±0.0028	0.8538 ±0.0158	0.7528 ±0.0025
XGBoost	0.8001 ±0.0081	0.6247 ±0.0081	0.5231 ±0.0001	0.5533 ±0.0187	0.9078 ±0.0002	0.8001 ±0.0035	0.7452 ±0.0250
CatBoost	0.8138 ±0.0073	**0.6416 ±0.0038**	0.5163 ±0.0413	**0.5685 ±0.0161**	0.8936 ±0.0040	0.7824 ±0.0252	**0.7545 ±0.0153**

Table 1: Comparison of F1 score between GFTab and baseline models on 21 tabular benchmark datasets in 20% labeled training setting. The best performing method is highlighted in red and the second best in blue, while the third best is **bold**.

Is GFTab really effective for tabular datasets? (without label noise)



Panel A. Datasets with more categorical variables

model	Diabetes	Insurance	adult	bank	cmc	credit-approval	credit-g
GFTab	0.3826 ±0.0139	0.4500 ±0.0022	0.8023 ±0.0070	0.7336 ±0.0040	**0.4625 ±0.0215**	0.6738 ±0.0019	0.7433 ±0.0424
GRANDE	0.3635 ±0.0012	0.4225 ±0.0001	0.7596 ±0.0033	0.7017 ±0.0042	0.4306 ±0.0229	0.8521 ±0.0008	0.6570 ±0.0224
TabPFN	0.2559 ±0.0089	0.4312 ±0.0001	**0.7530 ±0.0024**	0.6102 ±0.0047	0.4566 ±0.0027	0.8975 ±0.0091	0.4802 ±0.0409
SCARF	0.2335 ±0.0001	0.4312 ±0.0002	0.4157 ±0.0244	0.4699 ±0.0009	0.3479 ±0.0425	0.6993 ±0.0518	0.4451 ±0.0377
SubTab	0.2569 ±0.0187	0.4312 ±0.0001	0.7506 ±0.0035	0.6626 ±0.0049	0.5012 ±0.0141	0.6728 ±0.0202	0.6474 ±0.0388
VIME	0.2611 ±0.0313	0.4314 ±0.0001	0.7369 ±0.0057	**0.6932 ±0.0179**	0.4942 ±0.0336	0.7177 ±0.1637	0.5397 ±0.0891
XGBoost	**0.3534 ±0.0002**	0.4312 ±0.0003	0.7344 ±0.0006	0.5246 ±0.0024	0.4540 ±0.0213	0.8276 ±0.0096	**0.5968 ±0.0171**
CatBoost	0.3552 ±0.0003	0.4312 ±0.0001	0.7428 ±0.0006	0.4844 ±0.0103	0.3694 ±0.0430	**0.8469 ±0.0082**	0.5590 ±0.0310

model	dresses-sales	fars	jasmine	kick	okcupid-stem	online-shoppers	shrutime
GFTab	0.4165 ±0.0057	0.6199 ±0.0104	**0.7775 ±0.0041**	0.4928 ±0.0044	0.4219 ±0.0195	**0.7871 ±0.0106**	**0.7088 ±0.0025**
GRANDE	0.4611 ±0.0256	**0.5600 ±0.0122**	0.7630 ±0.0044	**0.4654 ±0.0012**	0.4412 ±0.0117	0.7405 ±0.0042	0.7355 ±0.0034
TabPFN	0.3671 ±0.0010	0.5247 ±0.0104	0.7491 ±0.0036	0.4749 ±0.0010	0.3889 ±0.0019	0.7954 ±0.0012	0.7130 ±0.0069
SCARF	0.4694 ±0.0838	0.1314 ±0.0095	0.5839 ±0.0138	0.4753 ±0.0010	0.2787 ±0.0000	0.4579 ±0.0003	0.4459 ±0.0044
SubTab	0.5300 ±0.0348	0.5020 ±0.0035	0.7078 ±0.0063	0.4840 ±0.0030	**0.4021 ±0.0118**	0.6624 ±0.0096	0.6808 ±0.0026
VIME	0.5069 ±0.0346	0.5900 ±0.0041	0.7548 ±0.0050	0.4758 ±0.0006	0.3266 ±0.0180	0.4583 ±0.0015	0.5323 ±0.0421
XGBoost	**0.5041 ±0.0553**	0.4196 ±0.0002	0.7851 ±0.0061	0.4749 ±0.0002	0.3546 ±0.0043	0.7964 ±0.0046	0.6863 ±0.0002
CatBoost	0.4056 ±0.0804	0.3571 ±0.0002	0.7940 ±0.0029	0.4749 ±0.0001	0.3062 ±0.0059	0.7847 ±0.0065	0.6215 ±0.0107

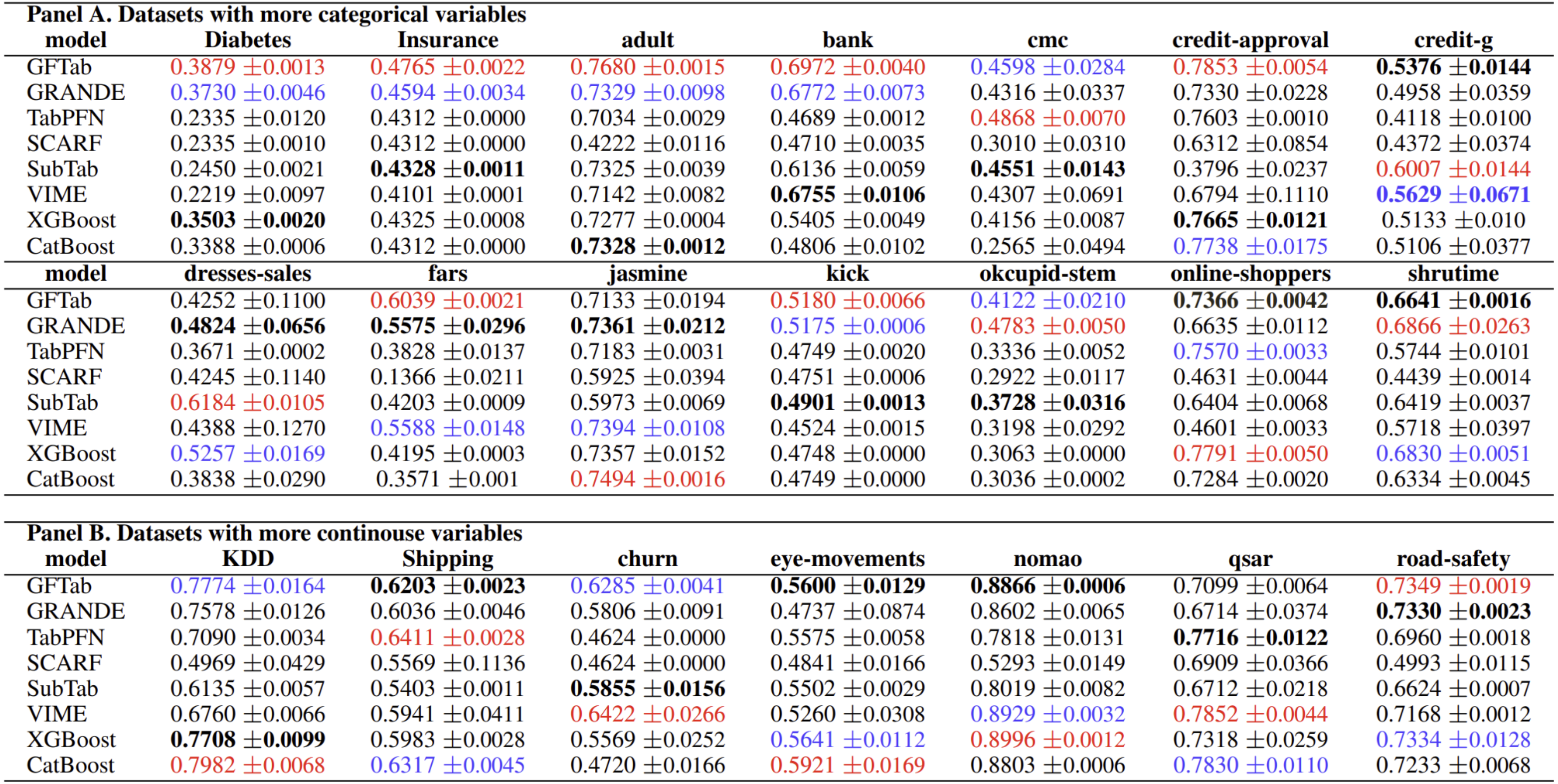
Panel B. Datasets with more continouse variables

model	KDD	Shipping	churn	eye-movements	nomao	qsar	road-safety
GFTab	**0.7998 ±0.0070**	0.6493 ±0.0020	0.7865 ±0.0365	0.5534 ±0.0205	0.9411 ±0.0030	0.7867 ±0.0140	0.7553 ±0.0042
GRANDE	0.7848 ±0.0069	0.6186 ±0.0045	0.7783 ±0.0247	0.5676 ±0.0102	0.9113 ±0.0067	0.7729 ±0.0110	0.7566 ±0.0017
TabPFN	0.7722 ±0.0045	0.6432 ±0.0024	**0.7657 ±0.0174**	0.5854 ±0.0020	0.8893 ±0.0032	**0.8254 ±0.0000**	0.7389 ±0.0015
SCARF	0.5603 ±0.0183	0.6213 ±0.0141	0.4624 ±0.0005	0.4878 ±0.0167	0.5065 ±0.0217	0.6126 ±0.0528	0.4976 ±0.0023
SubTab	0.6634 ±0.0269	0.5514 ±0.0091	0.7539 ±0.014	0.5711 ±0.0060	**0.9290 ±0.0038**	0.8404 ±0.0139	0.6750 ±0.0006
VIME	0.7042 ±0.0164	0.6358 ±0.0148	0.7051 ±0.0304	0.5524 ±0.0277	0.9361 ±0.0028	0.8538 ±0.0158	0.7528 ±0.0025
XGBoost	0.8001 ±0.0081	0.6247 ±0.0081	0.5231 ±0.0001	0.5533 ±0.0187	0.9078 ±0.0002	0.8001 ±0.0035	0.7452 ±0.0250
CatBoost	0.8138 ±0.0073	**0.6416 ±0.0038**	0.5163 ±0.0413	**0.5685 ±0.0161**	0.8936 ±0.0040	0.7824 ±0.0252	**0.7545 ±0.0153**

Table 1: Comparison of F1 score between GFTab and baseline models on 21 tabular benchmark datasets in 20% labeled training setting. The best performing method is highlighted in red and the second best in blue, while the third best is **bold**.

There is no one-size-fits-all solution for all tabular datasets

Is GFTab really effective for tabular datasets? **(without label noise)**



Panel A. Datasets with more categorical variables

model	Diabetes	Insurance	adult	bank	cmc	credit-approval	credit-g
GFTab	0.3879 ±0.0013	0.4765 ±0.0022	0.7680 ±0.0015	0.6972 ±0.0040	0.4598 ±0.0284	0.7853 ±0.0054	**0.5376 ±0.0144**
GRANDE	0.3730 ±0.0046	0.4594 ±0.0034	0.7329 ±0.0098	0.6772 ±0.0073	0.4316 ±0.0337	0.7330 ±0.0228	0.4958 ±0.0359
TabPFN	0.2335 ±0.0120	0.4312 ±0.0000	0.7034 ±0.0029	0.4689 ±0.0012	0.4868 ±0.0070	0.7603 ±0.0010	0.4118 ±0.0100
SCARF	0.2335 ±0.0010	0.4312 ±0.0000	0.4222 ±0.0116	0.4710 ±0.0035	0.3010 ±0.0310	0.6312 ±0.0854	0.4372 ±0.0374
SubTab	0.2450 ±0.0021	**0.4328 ±0.0011**	0.7325 ±0.0039	0.6136 ±0.0059	**0.4551 ±0.0143**	0.3796 ±0.0237	0.6007 ±0.0144
VIME	0.2219 ±0.0097	0.4101 ±0.0001	0.7142 ±0.0082	**0.6755 ±0.0106**	0.4307 ±0.0691	0.6794 ±0.1110	**0.5629 ±0.0671**
XGBoost	**0.3503 ±0.0020**	0.4325 ±0.0008	0.7277 ±0.0004	0.5405 ±0.0049	0.4156 ±0.0087	**0.7665 ±0.0121**	0.5133 ±0.010
CatBoost	0.3388 ±0.0006	0.4312 ±0.0000	**0.7328 ±0.0012**	0.4806 ±0.0102	0.2565 ±0.0494	0.7738 ±0.0175	0.5106 ±0.0377

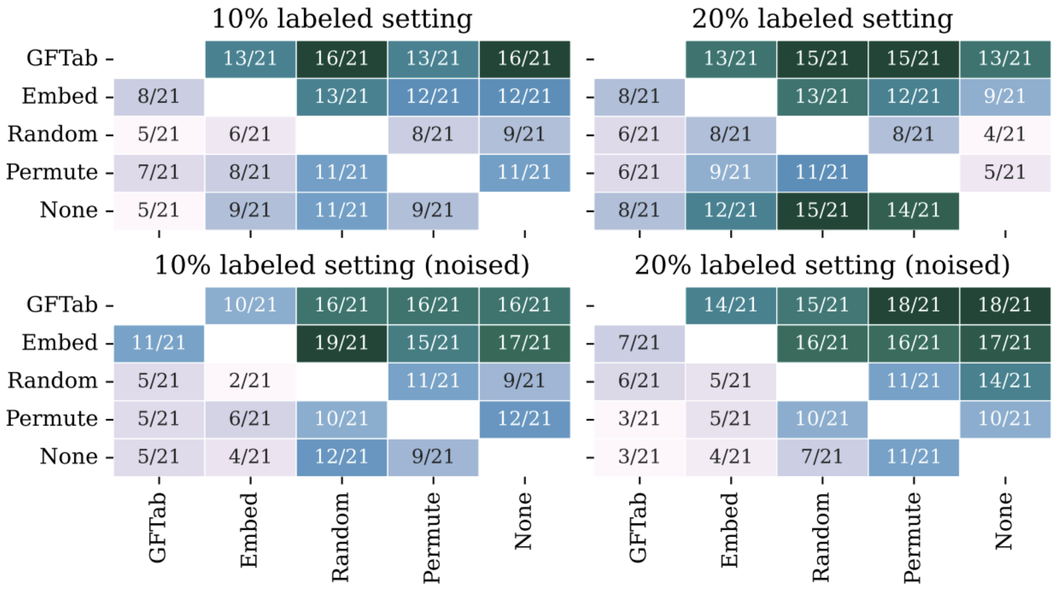
model	dresses-sales	fars	jasmine	kick	okcupid-stem	online-shoppers	shrutime
GFTab	0.4252 ±0.1100	0.6039 ±0.0021	0.7133 ±0.0194	0.5180 ±0.0066	0.4122 ±0.0210	**0.7366 ±0.0042**	**0.6641 ±0.0016**
GRANDE	**0.4824 ±0.0656**	**0.5575 ±0.0296**	**0.7361 ±0.0212**	0.5175 ±0.0006	0.4783 ±0.0050	0.6635 ±0.0112	0.6866 ±0.0263
TabPFN	0.3671 ±0.0002	0.3828 ±0.0137	0.7183 ±0.0031	0.4749 ±0.0020	0.3336 ±0.0052	0.7570 ±0.0033	0.5744 ±0.0101
SCARF	0.4245 ±0.1140	0.1366 ±0.0211	0.5925 ±0.0394	0.4751 ±0.0006	0.2922 ±0.0117	0.4631 ±0.0044	0.4439 ±0.0014
SubTab	0.6184 ±0.0105	0.4203 ±0.0009	0.5973 ±0.0069	**0.4901 ±0.0013**	**0.3728 ±0.0316**	0.6404 ±0.0068	0.6419 ±0.0037
VIME	0.4388 ±0.1270	0.5588 ±0.0148	0.7394 ±0.0108	0.4524 ±0.0015	0.3198 ±0.0292	0.4601 ±0.0033	0.5718 ±0.0397
XGBoost	0.5257 ±0.0169	0.4195 ±0.0003	0.7357 ±0.0152	0.4748 ±0.0000	0.3063 ±0.0000	0.7791 ±0.0050	0.6830 ±0.0051
CatBoost	0.3838 ±0.0290	0.3571 ±0.001	0.7494 ±0.0016	0.4749 ±0.0000	0.3036 ±0.0002	0.7284 ±0.0020	0.6334 ±0.0045

Panel B. Datasets with more continouse variables

model	KDD	Shipping	churn	eye-movements	nomao	qsar	road-safety
GFTab	0.7774 ±0.0164	**0.6203 ±0.0023**	0.6285 ±0.0041	**0.5600 ±0.0129**	**0.8866 ±0.0006**	0.7099 ±0.0064	0.7349 ±0.0019
GRANDE	0.7578 ±0.0126	0.6036 ±0.0046	0.5806 ±0.0091	0.4737 ±0.0874	0.8602 ±0.0065	0.6714 ±0.0374	**0.7330 ±0.0023**
TabPFN	0.7090 ±0.0034	0.6411 ±0.0028	0.4624 ±0.0000	0.5575 ±0.0058	0.7818 ±0.0131	**0.7716 ±0.0122**	0.6960 ±0.0018
SCARF	0.4969 ±0.0429	0.5569 ±0.1136	0.4624 ±0.0000	0.4841 ±0.0166	0.5293 ±0.0149	0.6909 ±0.0366	0.4993 ±0.0115
SubTab	0.6135 ±0.0057	0.5403 ±0.0011	**0.5855 ±0.0156**	0.5502 ±0.0029	0.8019 ±0.0082	0.6712 ±0.0218	0.6624 ±0.0007
VIME	0.6760 ±0.0066	0.5941 ±0.0411	0.6422 ±0.0266	0.5260 ±0.0308	0.8929 ±0.0032	0.7852 ±0.0044	0.7168 ±0.0012
XGBoost	**0.7708 ±0.0099**	0.5983 ±0.0028	0.5569 ±0.0252	0.5641 ±0.0112	0.8996 ±0.0012	0.7318 ±0.0259	0.7334 ±0.0128
CatBoost	0.7982 ±0.0068	0.6317 ±0.0045	0.4720 ±0.0166	0.5921 ±0.0169	0.8803 ±0.0006	0.7830 ±0.0110	0.7233 ±0.0068

Table 2: Comparison of F1 score between GFTab and baseline models on 21 tabular benchmark datasets in 20% labeled training with 20% label noise. The best performing method is highlighted in red and the second best in blue, while the third best is **bold**.

Is GFTab really effective for tabular datasets? (with label noise)



10% labeled setting

	GFTab	Embed	Random	Permute	None
GFTab		13/21	16/21	13/21	16/21
Embed	8/21		13/21	12/21	12/21
Random	5/21	6/21		8/21	9/21
Permute	7/21	8/21	11/21		11/21
None	5/21	9/21	11/21	9/21	

20% labeled setting

	GFTab	Embed	Random	Permute	None
GFTab		13/21	15/21	15/21	13/21
Embed	8/21		13/21	12/21	9/21
Random	6/21	8/21		8/21	4/21
Permute	6/21	9/21	11/21		5/21
None	8/21	12/21	15/21	14/21	

10% labeled setting (noised)

	GFTab	Embed	Random	Permute	None
GFTab		10/21	16/21	16/21	16/21
Embed	11/21		19/21	15/21	17/21
Random	5/21	2/21		11/21	9/21
Permute	5/21	6/21	10/21		12/21
None	5/21	4/21	12/21	9/21	

20% labeled setting (noised)

	GFTab	Embed	Random	Permute	None
GFTab		14/21	15/21	18/21	18/21
Embed	7/21		16/21	16/21	17/21
Random	6/21	5/21		11/21	14/21
Permute	3/21	5/21	10/21		10/21
None	3/21	4/21	7/21	11/21	

Figure 2: Win matrices for different categorical variable corruption methods.

How to corrupt categorical variables effectively?

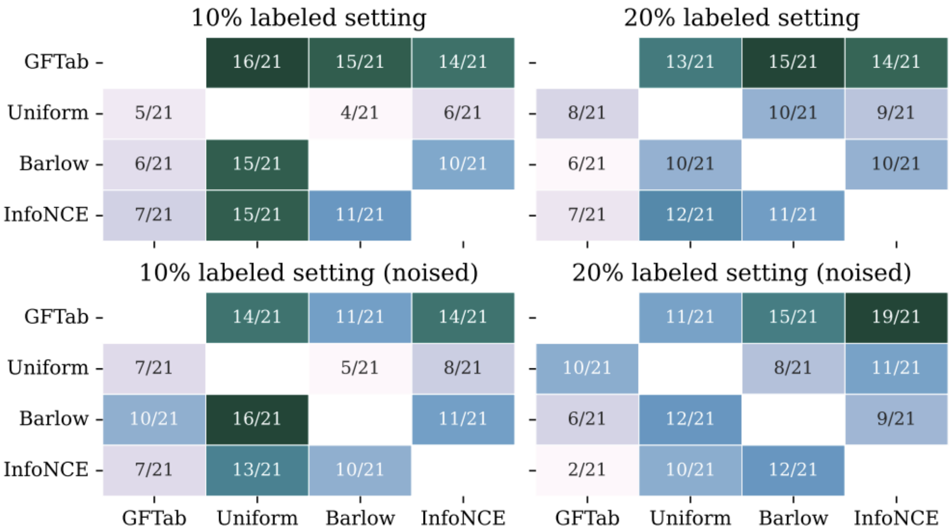


Figure 3: Win matrix between GFTab with three different similarity losses.

Is geodesic flow useful for tabular datasets?

Proposed GFTab: A semi-supervised framework designed for mixed-type tabular data (continuous + categorical).

Key Contribution

- *Variable-Specific Corruption*: Tailored noise injection for continuous vs. categorical variables.
- *Geodesic Flow Kernel*: Smoothly measures similarity across corrupted data subspaces.
- *Tree-Based Embedding*: Leverages hierarchical relationships from labeled data.

Experimental Results

- Outperforms existing ML/DL baselines under limited labeled data and noisy label settings.
- Robust across diverse datasets with both categorical-dominant and continuous-dominant features.

Conclusion

Thank you