TABPCA: FEATURE-DRIVEN TABULAR DATA AUGMENTATIONS FOR SELF-SUPERVISED LEARNING

Sol A Kim, Ho-Kuen Shin, Sehyun Lee
{ksola,vision,sehyun.lee}@kaist.ac.kr

ABSTRACT

Self-supervised learning has demonstrated remarkable success in real-world environments with a scarcity of labeled data. In this work, we explore the potential of self-supervised learning on tabular data, recognizing that tabular data differs from image and text data in terms of its unique characteristics. Unlike images and text, where randomness-based augmentations can yield invariant views, tabular data may not exhibit the same invariance properties. Therefore, we advocate for feature-dependent augmentations tailored specifically to tabular data, taking into account its distinct properties. Our analysis focuses on investigating the impact of multiple views of tabular data within a contrastive learning framework while incorporating existing data augmentation techniques. Additionally, we propose two novel augmentation schemes, TabPCA and LatentTabPCA, which aim to introduce diverse variations while preserving the original distribution and considering feature independence. By leveraging these tailored augmentation strategies, our goal is to extract robust and meaningful representations from tabular data. To assess the effectiveness of our approaches, we extensively evaluate them on benchmark datasets, directly comparing them with widely adopted self-supervised contrastive augmentations.

1 INTRODUCTION

Self-supervised learning has emerged as a powerful technique for training deep neural networks in the absence of labeled data. This approach has shown great promise in various domains, particularly in computer vision (Chen et al., 2020; He et al., 2020; Zbontar et al., 2021; Grill et al., 2020; Ghiasi et al., 2022), where it has achieved state-of-the-art results on various benchmark datasets. However, self-supervised learning is not limited to vision tasks and has recently gained attention in the context of tabular data.

Tabular data is a crucial component of many real-world applications, including finance, healthcare, manufacturing, and customer relationship management, among others. However, applying self-supervised learning to tabular data poses unique challenges due to its characteristics (Arik & Pfister, 2021). First, tabular data is often characterized by containing independent values among neighbors. Second, tabular data can be noisy, have missing features, or suffer from class imbalance. Third, tabular datasets can have a diverse range of sample sizes and column counts, which can make it challenging to design a single self-supervised learning algorithm that works well across different datasets.

Despite the challenges, some recent works have started exploring the potential of self-supervised learning on tabular data (Arik & Pfister, 2021; Hajiramezanali et al., 2022; Bahri et al., 2022; Somepalli et al., 2021; Majmundar et al., 2022; Yoon et al., 2020). However, still, decision tree-based models have shown promise in effectively modeling tabular data due to their representational efficiency for decision manifolds with approximately hyperplane boundaries, which are commonly present in tabular data (Grinsztajn et al., 2022). Although decision tree-based models are highly interpretable, they still require manual feature engineering and may not perform well in the presence of missing data, noisy data, or imbalanced data. (Dua et al., 2017; IBM, 2019; LeCun et al., 1998; Cortez et al., 2009)

Therefore, self-supervised learning provides an alternative way to extract meaningful features from tabular data without the need for hand-crafted features (Rubachev et al., 2023). To leverage the full
potential of self-supervised learning on tabular data, it is crucial to explore and understand how to create different views for tabular data in a contrastive setting. In this paper, we aim to contribute to this research direction by analyzing the effect of multiple views of tabular data in a contrastive setting, while incorporating existing data augmentation schemes. Then, we propose new data augmentation techniques TabPCA and LatentTabPCA, which aim to introduce diverse variations while preserving the original distribution and considering feature independence. We hypothesize that the optimal augmentation can extract more robust and meaningful representations from tabular data than existing methods. To validate our hypothesis, we will conduct extensive experiments on various benchmark datasets and compare our method with state-of-the-art self-supervised and supervised learning algorithms.

2 RELATED WORK

Data augmentation techniques highly depend on which domain to take. In this section, we briefly explain the methods per specific domain (computer vision, natural language processing, and tabular data).

Geometric transformations (e.g. cropping, rotation, translation, and clipping), color space transformations are domain-specific augmentation for image data. (Shorten & Khoshgoftaar, 2019) Based on the simple transformations, machine learning on computer vision tasks including self-supervised learning have been developed through various researches of data augmentations; mixing images (Hendrycks et al., 2019; Summers & Dinneen, 2019; Lee et al., 2020; Yun et al., 2019), random erasing (Singh et al., 2018; Zhong et al., 2020), feature space augmentations (Hendrycks et al., 2021; Verma et al., 2019), and GAN-based synthetic augmentations (Frid-Adar et al., 2018; Zhu et al., 2017; Shaham et al., 2019).

Different to the image data, in NLP, the meaning of a sentence could be changed by minor changes of words. The techniques do not change the meaning are synonym replacement (Zhang et al., 2015), text surface transformation (Coulombe, 2018), word embeddings replacement (Wang & Yang, 2015; Kobayashi, 2018; Luque, 2019), and masked language models (Devlin et al., 2018; Liu et al., 2019; Lan et al., 2019). From the augmentation techniques on computer vision tasks, noising (Xie et al., 2017), random inserting/deleting/swapping (Wei & Zou, 2019), mixing (Guo et al., 2019) augmentations are proposed. There exists not only the aggregations of character-wise or word-wise changes but also the generation of the whole sentence (Edunov et al., 2018; Kafle et al., 2017).

As mentioned above, significant progress has been made in generating multiple views of input data for self-supervised learning in the image domain. However, applying these methods directly to tabular data is not straightforward due to its unique characteristics. Recent research has been actively exploring the potential of self-supervised learning on tabular data. (Somepalli et al., 2021) adopts CutMix (Yun et al., 2019) in the real space of tabular data and applies Mixup (Zhang et al., 2017) in the latent space to perform contrastive pre-training. (Ucar et al., 2021) preprocesses the input features of the table by dividing them into multiple overlapped subsets, inspired by image cropping. Plus, for randomly selected cells, 3 types of noise are added, including Gaussian noise, overwriting by the values of other samples in the same column, and zeroing out. (Yoon et al., 2020) proposes a pretext task of estimating mask vectors after adding column-wise swap noise as an augmentation method. (Arik & Pfister, 2021) mimics tree-based model structures and uses multiplicative sparse masks for unsupervised representation learning. (Hollmann et al., 2022) shuffles columns and uses permuted class labels. (Bahri et al., 2022) proposes a method of contrastive learning with randomly corrupting by drawing from the marginal distribution of each feature. (Wang & Sun, 2022) utilizes overlapped samples as positive samples and non-overlapped samples as negative samples in the real-world table data that is partially overlapped by each other. (Lee et al., 2022) provides self-supervision by reconstructing the original input using only a part of the input features and estimating a gate vector that defines which features to select.
Figure 1: Data Augmentations for generating diverse views. We introduce TabPCA as a novel augmentation method for tabular data, aiming to create different views. Considering the distinct properties of each feature, it is rational to apply feature-dependent perturbation strategies rather than random perturbations. Our approach involves using two samples, the original and augmented samples, to form positive pairs (excluding Subset). After embedding the sample in the latent space, MixUp augmentation is applied. LatentTabPCA is a variant of TabPCA applied after embedding.

3 Self-supervised Learning for Tabular Data

3.1 Tabular Data Augmentations

Tabular data, which is widely utilized in various fields for storing and organizing data, also faces the challenge of a scarcity of labeled data, which is crucial for effective modeling, despite the abundance of data in many cases. Moreover, due to the unique nature of tabular data, annotating unlabeled data proves to be a daunting task even for human experts, unlike in the domains of images and language. Consequently, finding ways to address the shortage of labeled data becomes paramount, making the learning of suitable representations for the tabular domain through self-supervised learning all the more important. While contrastive learning has been the primary focus of research for pretraining tabular data, recent studies have explored masked encoder approaches as well. However, the characteristics of tabular data, where each cell value is independent and unrelated to its neighboring values, pose challenges in intuitively applying data augmentation methods that provide different views, as commonly done in contrastive learning. In contrast, the image domain benefits from widely used techniques such as rotation, color jitter, resize, and random crop, which not only preserve semantic information but also significantly enhance the model’s comprehension of contextual information in images.

We have extensively reviewed and discussed approximately 30 augmentation methods that have been employed in the image domain, from the perspective of tabular data. As a result of our analysis, we have discovered that principal component analysis (PCA) based perturbation, known as FancyPCA and originally proposed in AlexNet [Krizhevsky et al., 2012], has the potential to introduce diverse changes to the original tabular data while preserving its original distribution and considering the interdependence of features for perturbation.

3.2 TabPCA for Tabular Data

We propose a novel PCA based tabular data perturbation method, called TabPCA for utilization in contrastive learning. Firstly, we find the principal components of the training tabular dataset. Then, for each row sample, we draw a random value from a normal distribution and multiply them by the eigenvalues. These values are added as noise to the original data to obtain perturbed samples.
Considering the tabular dataset as a $N \times M$ matrix, perturbed $n_{th}$ row data $x'_n$:

$$x'_n = x_n + [v_1, ..., v_M][\alpha_n \lambda_1, ..., \alpha_n \lambda_M]^T, \quad n = 1, ..., N,$$

where $v_m$ and $\lambda_m$ are $m_{th}$ eigenvector and eigenvalue of the $M \times M$ covariance matrix of tabular dataset. alpha value is drawn for every sample and is applied to all column.

TabPCA performs a dataset-specific principal component analysis, utilizing the eigenvalues and eigenvectors to proportionally perturb the data and obtain augmented samples. This approach offers the advantage of preserving not only the characteristics of the original features but also the relationships among different features, ensuring the retention of important details.

3.3 LatentTabPCA for Categorical Data

Tabular data values can be classified into categorical and numerical types based on their form. However, PCA is basically applicable to features with continuous values, making it unsuitable for directly applying TabPCA to categorical values. In this paper, we propose a novel approach called LatentTabPCA to address this issue by transforming both categorical and numerical values into a latent space and performing PCA and augmentation on the continuous embedding values. However, since the trainable embeddings are not fixed and are learned its training, the eigenvalues, and eigenvectors need to be calculated through PCA at each epoch. Moreover, considering the embedding dimension for each feature adds computational complexity, which can increase quadratically. Therefore, in our LatentTabPCA experiments, we limit the embedding dimension to 8. The perturbed latent valuables of categorical and continuous values for $i_{th}$ embedding dimension are

$$p'_\text{categ} = p^\text{categ} + [v_{l1}, ..., v_{LM}][\alpha_{nl}\lambda_{l1}, ..., \alpha_{nl}\lambda_{LM}]^T, \quad n = 1, ..., N, \quad l = 1, ..., L$$

$$p'_\text{cont} = p^\text{cont} + [v_{l1}, ..., v_{LM}][\alpha_{nl}\lambda_{l1}, ..., \alpha_{nl}\lambda_{LM}]^T, \quad n = 1, ..., N, \quad l = 1, ..., L$$

where $L$ is the dimension of embeddings.

4 Pre-Training & Fine-Tuning

Self-supervised contrastive learning has demonstrated remarkable success in the vision and language domains by training models to be invariant to various label-preserving "views" of data (Chen et al., 2020; Grill et al., 2020; Bordes et al., 2021). Building upon the achievements of contrastive learning in vision and text domains, researchers have begun exploring the application of these techniques to the tabular domain (Somepalli et al., 2021; Yoon et al., 2020; Ucar et al., 2021; Wang & Sun, 2022). However, previous studies have mainly focused on augmentations based on randomness, such as Gaussian noise, swapping, partitioning (subsets), CutMix, and MixUp. It is important to note that each feature (column) in tabular data has different characteristics, including varying variances, unlike image and text data, which often exhibit universal characteristics across the feature dimension. We argue that it is beneficial to consider feature-specific characteristics in tabular data, leading us to propose the use of TabPCA and LatentTabPCA augmentations.

Details about Generating augmentations Crafting invariance transforms for tabular data presents a challenge. In our study, we adopt traditional noise-adding techniques and other techniques that have been used in previous works (Somepalli et al., 2021; Yoon et al., 2020), along with our novel approaches. Gaussian noise involves adding small white noise to each value in the sample. Swap noise randomly swaps the features within a sample (row). Subset partitioning divides the data sample into subsets based on a specified number and overlap percentage. CutMix augments the masked part of a sample by mixing it with other samples in the batch. MixUp, similar to CutMix, involves blending the values of different rows within the same batch at a certain ratio to obtain an augmented view. We also explore the combined approach, CutMix+MixUp. Additionally, we propose our novel methods, TabPCA and LatentTabPCA, which are described in detail in Sections 3.2 and 3.3.

Network architecture We basically follow the architecture from (Somepalli et al., 2021), which is a transformer-based model. After obtaining the original embeddings $p_i$ and augmented embeddings
Figure 2: Self-supervised pretraining workflow. After generating diverse views of the data, we obtain representation vectors denoted as $r$. The utilization of VICReg loss and denoising techniques is crucial in our self-supervised pretraining approach, as motivated by previous works (Somepalli et al., 2021; Bardes et al., 2021). For the MixUp and LatentTabPCA augmentations, the sample undergoes augmentation after passing through the embedding layer. In the case of Subset, the number of samples compared is determined by the hyperparameter values of the number of subsets and overlap percentage.

$p'_i$, we feed them through the network and then through two projection heads. Each projection head is composed of a multi-layer perceptron (MLP) with a single hidden layer and a rectified linear unit (ReLU) activation function. This approach of using a projection head to reduce dimensionality before calculating the contrastive loss.

**Objective functions** Utilizing the power of contrastive learning and masking modeling, we employ two objective functions in our approach. The contrastive objective function comprises two components: a covariance term and a variance term. For the denoising objective function, we employ mean squared error (MSE) loss for continuous features and a combination of subset augmentation and cross-entropy loss for categorical features. This is achieved by feeding the representation $r'_i$ into an additional projection head.

$$L(\cdot) = \frac{1}{d} \sum_{j=1}^{d} \max(0, \gamma - S(z'_j, \epsilon)) + \frac{1}{n-1} \sum_{i=1}^{n}(z_i - \bar{z})(z_i - \bar{z})^T + \frac{1}{n} \sum_{i=1}^{n}(x''_i - x_i)^2$$ (4)

**Fine-tuning** After pre-training the network on all unlabeled data, we proceed to finetune the model for the target prediction task using the labeled samples. For each data point $x_i$, we learn the contextual embedding $r_i$. In the case of subset augmentation, we aggregate the representations of subsets by taking their average. In the final prediction step, we extract the embedding corresponding to the [CLS] token and pass it through a simple MLP with a single hidden layer activated by ReLU to obtain the final output. For classification tasks, we evaluate the model using cross-entropy loss on the output probabilities, while for regression tasks, we employ root mean squared error as the evaluation metric.

5 Experimental Evaluations

**Datasets** To assess the robustness and meaningfulness of the learned representations, we conduct experiments on diverse datasets (Table 1) with distinct characteristics, such as numerical data, categorical data, and mixed datasets. These datasets encompass 3 binary classification tasks, 2 multiclass classification tasks, and 1 regression task. They have been previously used to evaluate competing methods such as TabNet, MET, VIME, and SAINT (Arik & Pfister, 2021; Majmundar et al., 2022; Yoon et al., 2020; Somepalli et al., 2021). The datasets vary in size, ranging from 200 to 495,141 samples, and in feature count, ranging from 8 to 784 features. They consist of both categorical and
Table 1: Multiple datasets to cover diverse characteristics of tabular data: the size of the dataset, number of feature dimensions, types of feature, and properties

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Task</th>
<th>Data type</th>
<th># Features</th>
<th># Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wine Quality</td>
<td>Numerical regression</td>
<td>Numeric</td>
<td>11</td>
<td>6497</td>
</tr>
<tr>
<td>Bank-marketing</td>
<td>Binary classification</td>
<td>Categ. + Numeric</td>
<td>9+7</td>
<td>45,211</td>
</tr>
<tr>
<td>BlastChar</td>
<td>Binary classification</td>
<td>Categ. + Numeric</td>
<td>17+3</td>
<td>7,043</td>
</tr>
<tr>
<td>Credit</td>
<td>Binary classification</td>
<td>Numeric, Imbalance</td>
<td>29</td>
<td>284,807</td>
</tr>
<tr>
<td>Volkert</td>
<td>Multiclass classification</td>
<td>Numeric</td>
<td>147</td>
<td>58,310</td>
</tr>
<tr>
<td>Eye movement</td>
<td>Multiclass classification</td>
<td>Categ. + Numeric</td>
<td>7+20</td>
<td>10,936</td>
</tr>
</tbody>
</table>

continuous features, with some datasets containing missing data. Additionally, the class distributions range from well-balanced to highly skewed. The datasets are publicly available from OpenML (Vanschoren et al., 2013). As a preprocessing step, continuous features are Z-normalized, and categorical features are label encoded before passing the data to the embedding layer.

**Baselines** In this study, our primary focus is to explore the impact of different views of tabular data in contrastive self-supervised learning on generating robust and meaningful representations. To investigate this research direction, we adopt the contrastive learning method VICReg as our baseline (Bardes et al., 2021). We consider various techniques to create multiple views of tabular data, including CutMix, Mixup, partitioning (cropping) with or without noise, and corruption.

**Metrics** Due to the prevalence of binary classification tasks in our analysis, we employ the area under the receiver operating characteristic curve (AUROC) as the primary metric to assess the performance of our models. AUROC provides a measure of how effectively the model distinguishes between the two classes within the dataset. For the two multiclass datasets, Volkert and Eye Movements, we select a one-versus-rest (OVR) strategy to approximate the multiclass problem as a binary classification task. For all experiments, we verify the results using the mean (with standard deviation) of five trials.

**Visualization** To compare augmented high-dimensional features, we bring a non-linear dimensionality reduction algorithm t-distributed stochastic neighbor embedding (t-SNE) (van der Maaten & Hinton, 2008). By measuring the pairwise distances in the high-dimensional and low-dimensional spaces using different probability distributions, the intra-cluster detail is more optimally visualized. tSNE optimizes over a set number of iterations, using gradient descent with Kullback-Leibler divergence as the cost function. It allows the visualization of the underlying local structure of high-dimensional data in 2 or 3 dimensions, allowing the results to be plotted in a simple scatter plot.

Table 2: Experimental results on benchmark datasets. The Wine Quality dataset is evaluated using the RMSE metric, where lower values indicate better performance. For the remaining datasets, AUROC is used as the evaluation metric, with higher values indicating better performance. In the case of multiclass classification tasks, the one-versus-rest (OVR) strategy is employed to approximate the multiclass problem as a binary classification task. For all experiments, we verify the results using the mean (with standard deviation) of five trials.

<table>
<thead>
<tr>
<th>Method</th>
<th>Wine Quality</th>
<th>Bank-marketing</th>
<th>BlastChar</th>
<th>Creditcard</th>
<th>Volkert</th>
<th>Eye Movements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>0.694 (0.015)</td>
<td>0.933 (0.007)</td>
<td>0.847 (0.011)</td>
<td>0.976 (0.012)</td>
<td>0.394 (0.008)</td>
<td>0.523 (0.015)</td>
</tr>
<tr>
<td>Swap</td>
<td>0.695 (0.013)</td>
<td>0.936 (0.004)</td>
<td>0.843 (0.009)</td>
<td>0.976 (0.020)</td>
<td>0.385 (0.002)</td>
<td>0.523 (0.010)</td>
</tr>
<tr>
<td>MixUp</td>
<td>0.783 (0.050)</td>
<td>0.942 (0.001)</td>
<td>0.849 (0.009)</td>
<td>0.971 (0.016)</td>
<td>0.403 (0.022)</td>
<td>0.529 (0.010)</td>
</tr>
<tr>
<td>CutMix</td>
<td>0.698 (0.015)</td>
<td>0.932 (0.002)</td>
<td>0.844 (0.008)</td>
<td>0.973 (0.014)</td>
<td>0.398 (0.011)</td>
<td>0.531 (0.018)</td>
</tr>
<tr>
<td>CutMix+MixUp</td>
<td>0.748 (0.033)</td>
<td>0.943 (0.003)</td>
<td>0.839 (0.009)</td>
<td>0.972 (0.009)</td>
<td>0.411 (0.016)</td>
<td>0.532 (0.017)</td>
</tr>
<tr>
<td>Subset</td>
<td>0.696 (0.018)</td>
<td>0.939 (0.006)</td>
<td>0.850 (0.002)</td>
<td>0.946 (0.004)</td>
<td>0.419 (0.012)</td>
<td>0.517 (0.007)</td>
</tr>
<tr>
<td>TabPCA</td>
<td>0.666 (0.003)</td>
<td>0.946 (0.008)</td>
<td>0.851 (0.002)</td>
<td>0.965 (0.005)</td>
<td>0.388 (0.016)</td>
<td>0.534 (0.010)</td>
</tr>
<tr>
<td>LatentTabPCA</td>
<td>0.727 (0.020)</td>
<td>0.941 (0.007)</td>
<td>0.849 (0.008)</td>
<td>0.964 (0.004)</td>
<td>0.431 (0.016)</td>
<td>0.539 (0.007)</td>
</tr>
</tbody>
</table>

**Results** As shown in Table 2, TabPCA and LatentTabPCA exhibited superior performance compared to conventional augmentations on three out of the six datasets evaluated. These proposed methods demonstrated their effectiveness in enhancing the representations of tabular data. The results provide evidence that TabPCA and LatentTabPCA offer enhanced robustness and meaningfulness
compared to existing augmentation techniques. This finding emphasizes the importance of incorporating feature-dependent augmentations specifically designed for tabular data within the framework of self-supervised learning.

![Figure 3](image1.png)

**Figure 3**: The result of dimensionality reduction of features using t-SNE on Volkert dataset which has ten classes. We extract features from VICReg with TabPCA + fine-tuned model for (a) and only the downstream task training without pretraining for (b).

![Figure 4](image2.png)

**Figure 4**: Two dimensional visualization of t-SNE on eye movements dataset which has three classes. We extract features from VICReg with LatentTabPCA + fine-tuned model for (a) and from VICReg with Gaussian noise + fine-tuned model (b).

Before comparing the trained features among augmentations with the visualization, we first check the effect of contrastive pretraining. In Figure 3, pretraining + fine-tuned model 3a has a higher inter-cluster distance than the downstream task training without pretraining 3b, especially about class 1 and class 4. Then, we compare t-SNE between features from VICReg with LatentTabPCA and Gaussian noise as in Figure 4. The trained features with LatentTabPCA augmentation 4a show a higher inter-cluster distance and the lower intra-cluster distance than the trained features with Gaussian noise augmentation 4b. Also, it shows some conflicts on class 0 in 4b which doesn’t appear in 4a. These observations support that our proposed tabular augmentations TabPCA and LatentTabPCA could successfully augment the data while considering the feature independence.

### 6 Conclusion

In this paper, we aimed to overcome the scarcity of labeled data in the tabular data domain by exploring augmentation methods for contrastive learning. The existing augmentation methods used in image and language domains can distort the context and inherent meaning of tabular data. In contrast, we propose TabPCA and LatentTabPCA, which are specifically tailored for the tabular domain. We evaluate their performance on various tasks such as binary classification, multi-class classification, and regression using openML public tabular datasets. The key idea is to consider the distribution of features specific to each tabular dataset when perturbing tabular data for contrastive learning. For categorical values, we propose LatentTabPCA, which performs principal component analysis in the latent space and applies perturbation. The augmentation methods we proposed con-
sider the distribution of features or latent spaces in the input tabular data, demonstrating excellent versatility and enabling their application in various tabular data solutions.

REFERENCES


