ReConTab: Regularized Contrastive Representation Learning for Tabular Data

Suiyao Chen *
University of South Florida
Tampa, FL 33620
suiyaochen@usf.edu

Jing Wu *
University of Illinois at Urbana-Champaign
Champaign, IL 61820
jingwu6@illinois.edu

Naira Hovakimyan
University of Illinois at Urbana-Champaign
Champaign, IL 61820
nhovakim@illinois.edu

Handong Yao
University of Georgia
Athens, GA 30602
handong.yao@uga.edu

Abstract

Representation learning stands as one of the critical machine learning techniques across various domains. Through the acquisition of high-quality features, pre-trained embeddings significantly reduce input space redundancy, benefiting downstream pattern recognition tasks such as classification, regression, or detection. Nonetheless, in the domain of tabular data, feature engineering and selection still heavily rely on manual intervention, leading to time-consuming processes and necessitating domain expertise. In response to this challenge, we introduce ReConTab, a deep automatic representation learning framework with regularized contrastive learning. Agnostic to any type of modeling task, ReConTab constructs an asymmetric autoencoder based on the same raw features from model inputs, producing low-dimensional representative embeddings. Specifically, regularization techniques are applied for raw feature selection. Meanwhile, ReConTab leverages contrastive learning to distill the most pertinent information for downstream tasks. Experiments conducted on extensive real-world datasets substantiate the framework's capacity to yield substantial and robust performance improvements. Furthermore, we empirically demonstrate that pre-trained embeddings can seamlessly integrate as easily adaptable features, enhancing the performance of various traditional methods such as XGBoost and Random Forest.

1 Introduction

In the last decade, representation learning has made remarkable strides in fields like computer vision and natural language processing, revolutionizing the way we extract valuable insights from image and text data. However, several critical industries, including healthcare[57, 11, 12], manufacturing[6, 13, 66, 10], agriculture[47, 77, 64] and various engineering fields[80, 70, 15, 9], still heavily rely on structured tabular data. Researchers traditionally leverage domain expertise for feature selection[21], model refinement[68, 67] and uncertainty quantification[14, 71, 69]. With high-quality hand-crafted features, it is typically believed that traditional tree-based models are able to automatically capture the feature importance and interactions, without additional tuning.

While manual feature engineering has been effective, it comes with challenges. The process of crafting high-quality features for tabular data is labor-intensive and lacks the guarantee of optimal

---

*These authors contributed equally to this work.

performance. Feature selection often requires iterative experimentation, making it resource and time-intensive. In tackle these challenges, recent research endeavors have sought to harness the potential of deep representation learning for more efficient feature engineering in tabular data.

However, tabular data presents unique hurdles that have hindered its integration with the remarkable success of deep learning in other domains. In contrast to text data, where tokens are inherently discrete, or images, where pixels exhibit spatial correlations, tabular data encompasses a diverse mix of continuous, categorical, and ordinal values. These values can exhibit complex interdependencies and correlations, adding layers of complexity to the modeling process. Moreover, unlike the structured nature of images or the sequential nature of text, tabular data lacks inherent positional information to capture the intrinsic meanings or learn explicit representations.

In this paper, we proposed ReConTab, a transformer-based framework to automatically generate high-quality embeddings as features for classification model improvement. Our framework consists of an asymmetric autoencoder (AE) architecture, which is able to extract the most critical information from raw features to provide substantial performance improvement and robustness for downstream classification tasks. Moreover, ReConTab can be effectively trained in both self- and semi-supervised modes. This adaptability ensures the model to perform well across various training scenarios, irrespective of the availability of labeled data. The contributions are summarized as follows:

- We proposed a transformer-based automatic feature engineering framework, which is agnostic to modeling tasks, with scalability and adaptability.
- We designed a novel AE architecture with regularization and contrastive learning for an enhanced feature learning process.
- We conducted a comprehensive empirical study on various public datasets that demonstrates the superiority of the proposed work in performance lift and robustness.
- We demonstrated that representative embeddings extracted from raw features can serve as readily applicable features, seamlessly augmenting the performance of various conventional classification models such as logistic regression and tree-based models, etc.

2 Related Work

2.1 Classical Models

Various traditional machine-learning methods have been developed for tabular data classification and regression tasks. When it comes to modeling linear relationships, Logistic Regression (LR) [73] and Generalized Linear Models (GLM) [32] are the prominent choices. For those seeking tree-based models, Decision Trees (DT) [8] are popular options. Additionally, there are various ensemble methods based on DT, such as XGBoost [16], Random Forest [7], CatBoost [56], and LightGBM [40]. These ensemble methods are widely embraced in the industry for their ability to model complex non-linear relationships, enhance interpretability, and handle various feature types, including null values and categorical features.

2.2 Deep Learning Models

The current research landscape has a prominent trend focusing on applying deep learning techniques to tabular data. This movement has given rise to diverse neural architectures, each of which is designed to enhance performance within tabular data domain. These architectures can be broadly classified into several categories [6, 27]. Firstly, there are supervised methods that harness the power of neural networks, including well-known models like ResNet [35], SNN [44], AutoInt [62], and DCN-V2 [72], to improve the handling of tabular data. Secondly, there exist hybrid approaches that seamlessly integrate decision trees with neural networks, resulting in end-to-end training. This category includes innovative techniques like NODE [53], GrowNet [3], TabNN [42], and DeepGBM [41]. Thirdly, transformer-based methods have emerged, allowing models to learn from attention-spanning features and data points. Notable examples in this class include TabNet [1], TabTransformer [36], and FT-Transformer [27]. Lastly, representation learning methods are gaining prominence, emphasizing effective information extraction through self- and semi-supervised learning techniques. Noteworthy models in this realm encompass VIME [82], SCARF [4], and SAINT [61]. These approaches align seamlessly with the growing emphasis on representation learning in the field.

2.3 Self- and Semi-supervised Representation Learning

In computer vision, deep representation learning methodologies have emerged as potent tools, capitalizing on self- and semi-supervised training paradigms [45, 23, 46]. These methodologies exhibit a dichotomy, falling into two distinct categories of innovation. The first category of deep representation learning methods is rooted in generative models, particularly autoencoders [43]. A striking exemplar within this genre is the Masked AutoEncoder (MAE) architecture introduced by [33]. MAE features an asymmetric encoder-decoder architecture purposefully crafted for the extraction of embeddings from images. Impressively, the framework demonstrates the capability to capture spatiotemporal information [24] and extends seamlessly to various domains such as 3D space [39] and multiple scales [59]. Notably, akin masking strategies, prevalent in the Natural Language Processing (NLP) community [22], have also been transposed into the tabular data landscape [1, 36, 81]. Furthermore, VIME [82] presents a method reminiscent of MAE in the tabular data context. VIME perturbs and encodes each data sample within the feature space through the involvement of two estimators. Subsequently, these estimators use decoders to reconstruct both a binary mask and the original, uncorrupted data samples, demonstrating versatility in information extraction.

The second category predominantly revolves around the contrastive learning paradigm and strategically employs data augmentation techniques. Within this domain, prominent models harnessed momentum-update strategies [34, 18, 75, 74], embraced the concept of large batch sizes [17], incorporated stop-gradient operations [19], spatiotemporal information[76], or even introduced an online network tasked with predicting the output of a target network [28]. Notably, these concepts, initially designed for image data, have gracefully transcended into the arena of tabular data. An exemplar of such adaptation is found in SCARF [4], which ingeniously incorporates the principles of SimCLR [17] to pre-train the encoder. This pre-training procedure employs a subset of feature corruption as a pivotal data augmentation method. Furthermore, the work of [61] exemplifies a contrastive framework tailored to tabular data, introducing SAINT, computing both column- and row-wise attentions.

2.4 Regularization

Regularization techniques, pivotal in machine learning and statistical modeling, mitigate overfitting and enhance generalization by introducing penalty terms into the loss function. Early approaches such as Ridge Regression, which applies L2 regularization to linear models [20], and Lasso Regression [50], which implements L1 regularization, paved the way for modern regularization methods. The Elastic Net [83] combines these approaches to strike a balance between feature selection and coefficient shrinkage, while Dropout [63] and Batch Normalization [38] cater specifically to neural networks, fostering robust and generalized representations. Other techniques like early stopping [55] and weight decay [49] further complement the regularization arsenal. Bayesian approaches introduce probabilistic frameworks, such as Bayesian regression [5] and Gaussian Processes [58], integrating prior beliefs and data likelihood. Recent trends encompass adversarial training [51] to enhance model robustness and graph regularization techniques [25] for graph-based data modeling tasks. As machine learning continues to advance, regularization remains vital for model generalization and robustness.

3 Method

In this section, we present ReConTab, our comprehensive approach for tabular data self- and semi-supervised representation learning. First, we outline the process of the regularization method. Second, we formulate the feature corruption process. The self-supervised training process is illustrated in the third sub-section, without knowing the task labels. The fourth sub-section elucidates our novel semi-supervised training method, wherein we leverage labels for contrastive learning. Finally, we expound on our utilization of pre-trained encoders and embeddings to improve downstream tasks.

3.1 Regularization

We apply regularization [21, 78] on the input layer by introducing a penalty term $\lambda \|\boldsymbol{W}\|_p$ into the loss function, where $\boldsymbol{W}$ represents the input weights, λ is the regularization parameter and p is the specific norm for the penalty. The idea behind is to prevent similar features from weighing too much in loss objective and to learn more robust representation, especially when highly correlated features

are present. For example, if we can reconstruct features A, B, and C with only feature A, then B and C should be assigned less weights.

3.2 Feature Corruption

It's common for the generative-based representation approach to use data augmentation techniques to generate robust feature embeddings. One of the most promising approaches is feature corruption, which has also been used in this paper to enhance our model's performance. Considering the original dataset $\mathcal{X} \subseteq \mathbb{R}^M$, given any tabular data point x_i, we have its j-th feature as x_{i_j}, where $x_i = (x_{i_1}, x_{i_2}, ..., x_{i_M}), j \subseteq M$, with M representing the dimension of features and i denoting the sample index. In our approach, for each sample, we stochastically select t features from the pool of M features and replace them with corrupted features denoted as c. To elaborate, we generate c from the distribution $\widehat{\mathcal{X}}i_j$, where $\widehat{\mathcal{X}}i_j$ represents the uniform distribution over $\mathcal{X}_{i_j} = \left\{ x_{i_j} : x_i \in \mathcal{X} \right\}$.

3.3 Self-supervised Learning

Self-supervised learning of ReConTab aims to learn informative representations from unlabeled data (Algorithm 1). For each of the two data samples, x_1 and x_2, we apply input weights and add feature corruption to obtain corrupted data. Then we encode the corrupted data using an encoder, f, resulting in two features, z_1 and z_2. The decoder d will decode the learned embeddings to reconstruct $\hat{x}_1$ and $\hat{x}_2$ respectively, from where we can define the reconstruction loss $\mathcal{L}_{\text{reconstruction}}$ for two samples x^1 and x^2 as the mean squared error (MSE) between input features and reconstructions, shown as:

$$\mathcal{L}_{\text{reconstruction}} = \frac{1}{M} \sum_{j=1}^{M} (x_{1_j} - \hat{x}_{1_j})^2 + \frac{1}{M} \sum_{j=1}^{M} (x_{2_j} - \hat{x}_{2_j})^2. \tag{1}$$

Algorithm 1 Self-supervised learning

Require: unlabeled data $\mathcal{X} \subseteq \mathbb{R}^M$, batch size B, encoder f, decoder d, mean squared error (MSE), input weights $\boldsymbol{W} \subseteq \mathbb{R}^M$, regularization parameter λ and specific norm for penalty p.
  for two sampled mini-batch $\left\{ x_i^1, y_i^1 \right\}_{i=1}^{B} \subseteq \{\mathcal{X}, \mathcal{Y}\}$ and $\left\{ x_i^2, y_i^2 \right\}_{i=1}^{B} \subseteq \{\mathcal{X}, \mathcal{Y}\}$ **do**
    for each sample x_i^1 and x_i^2,
    apply input weights $x_i^1 = x_i^1 \boldsymbol{W}, x_i^2 = x_i^2 \boldsymbol{W}$, for $i \in [B]$
    apply feature corruption, define the corrupted feature as: $\breve{x}_i^1$ and $\breve{x}_i^2$, for $i \in [B]$
    data encoding:
      $z_i^1 = f(\breve{x}_i^1), z_i^2 = f(\breve{x}_i^2)$, for $i \in [B]$
    data reconstruction:
      $\hat{x}_i^1 = d(z_i^1), \hat{x}_i^2 = d(z_i^2)$, for $i \in [B]$
    define reconstruction loss $\mathcal{L}_{\text{reconstruction}} = \text{MSE}(x_i^1, \hat{x}_i^1) + \text{MSE}(x_i^2, \hat{x}_i^2)$
    define penalty as $\lambda \|\boldsymbol{W}\|_p$
    update encoder f and decoder d to minimize $\mathcal{L}_{\text{reconstruction}}$ and $\lambda \|\boldsymbol{W}\|_p$ using RMSProp.
  end for

Therefore, the loss function for self-supervised learning can be defined as:

$$\mathcal{L}_{\text{self}} = \mathcal{L}_{\text{reconstruction}} + \lambda \|\boldsymbol{W}\|_p, \tag{2}$$

3.4 Semi-supervised Learning

We further improve the pre-training process through semi-supervised learning to take advantage of labeled data, as shown in Figure 1. In self-supervised learning, we only compute the MSE between reconstructed data and original data as the reconstruction loss $\mathcal{L}_{\text{reconstruction}}$. With labels introduced, we can pose additional constraints to the encoded embeddings z_1 and z_2. One is for label prediction to compute the prediction loss (illustrated by classification loss $\mathcal{L}_{\text{classification}}$ through the context). To be specific, z_1 and z_2 are fed to the same multi-layer perceptron (MLP) that maps from the embedding space to the label space. We can also define the cross-entropy loss for classification task as:

$$\mathcal{L}_{\text{classification}} = - \left(y_1 \log(\hat{y}_1) + y_2 \log(\hat{y}_2) \right), \tag{3}$$

where $\hat{y}_1$ and $\hat{y}_2$ are predicted labels computing a MLP, i.e., $\hat{y}_1 = \text{MLP}(z_1)$ and $\hat{y}_2 = \text{MLP}(z_2)$.

3.4.1 Contrastive Loss

We further introduce the contrastive loss $\mathcal{L}_{\text{contrastive}}$ in the loss function by forming contrastive pairs (z_1, z_2) of embeddings in the bottleneck layer with respect to the classification labels (y_1, y_2). With this constraint, the model is enforced to maximize the similarity between embeddings with the same label and minimize the similarity between embeddings with different labels, thus capturing the discriminative features for the classification labels and better aligning with downstream tasks. Algorithm 2 formally defines the contrastive loss in the proposed model, which is a variation from the original contrastive learning [31] and relevant to these extensions [17, 65, 26].

Algorithm 2 Contrastive Loss for Semi-supervised learning

Require: data embeddings $\mathcal{Z}$ from unlabeled data $\mathcal{X} \subseteq \mathbb{R}^M$, binary labels $\mathcal{Y} \subseteq \mathbb{R}$, batch size B, encoder f, decoder d, contrastive loss margin m, distance function $D(\cdot)$

  for two sampled mini-batch $\left\{z_i^1, y_i^1\right\}_{i=1}^{B} \subseteq \{\mathcal{X}, \mathcal{Y}\}$ and $\left\{z_i^2, y_i^2\right\}_{i=1}^{B} \subseteq \{\mathcal{X}, \mathcal{Y}\}$ **do**
    for each sample embedding z_i^1 and z_i^2,
    define contrastive loss:
    for $i = 1$ to B **do**
      if $y_i^1 = y_i^2$ **then**
        $y_i^{12} = 1$ for the pair (z_i^1, z_i^2) // z_i^1 is deemed similar to z_i^2
      else
        $y_i^{12} = 0$ for the pair (z_i^1, z_i^2) // z_i^1 is deemed dissimilar to z_i^2
      end if
      $d_i = D(z_i^1, z_i^2)$ // calculate the distance of two embeddings in the pair
      $c_i = (y_i^{12})\frac{1}{2}d_i^2 + (1 - y_i^{12})\frac{1}{2}\max(0, m - d_i)^2$ // calculate the contrastive loss of the pair
    end for
    $\mathcal{L}_{contrastive} = \frac{1}{B}\sum c_i$
    update encoder f and decoder d to minimize $\mathcal{L}_{contrastive}$ using RMSProp.
  end for

During the optimization stage, we combine the two additional losses with the self-supervised learning loss $\mathcal{L}_{\text{self}}$ and define the semi-supervised learning loss function $\mathcal{L}_{\text{semi}}$ as follows:

$$\mathcal{L}_{\text{semi}} = \mathcal{L}_{\text{self}} + \alpha * \mathcal{L}_{\text{classification}} + \beta * \mathcal{L}_{\text{contrastive}}, \tag{4}$$

where α and β are used to seek balance among multiple losses and set to 1 as default, respectively.
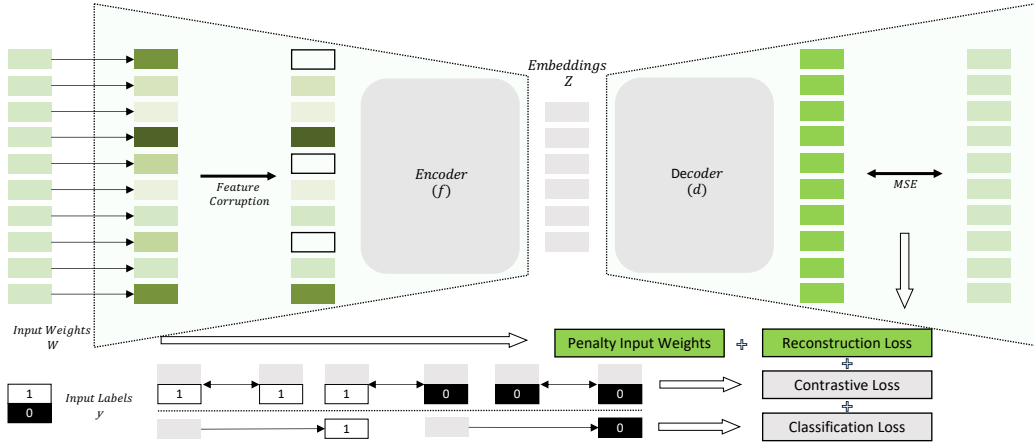

Figure 1: Proposed AE architecture with contrastive loss and input weights regularization

3.5 Downstream Fine-Tuning

Drawing inspiration from established representation learning paradigms [34, 18, 17, 4], we embrace an end-to-end fine-tuning strategy for the pre-trained encoder f from ReConTab, utilizing the

complete labeled dataset. This approach entails the seamless integration of the encoder with an additional linear layer, thereby granting the flexibility to unlock and adapt all its parameters to align with the specific requirements of downstream supervised tasks. Additionally, we can harness the potential of the salient feature s as a versatile plug-and-play embedding. Through the fusion of z with its original counterpart x, we construct enriched data points. This innovative approach serves to amplify inherent data characteristics, thereby assisting in the establishment of distinct decision boundaries. As a result, we anticipate notable enhancements in classification tasks when employing the concatenated features as the input for conventional models like Random Forest or LightGBM.

4 Experiments and Results

In this section, we present the results of our extensive experiments conducted on diverse public datasets to highlight the effectiveness of our proposed method, ReConTab. This section is structured into two parts for clarity and comprehensiveness. In the first part, we provide essential details regarding the experiments. This includes information about the public datasets for experiments, the preprocessing steps applied to these datasets, the architecture of our models, and specific training procedures. This transparency ensures the reproducibility of our findings.

In the second part, we assess the performance of ReConTab through various empirical studies. We conduct a thorough comparison between ReConTab and mainstream deep learning methods as well as traditional methods. Meanwhile, we showcase the versatility of ReConTab using it as an automatic feature engineering tool. Specifically, we demonstrate how ReConTab can enhance the performance of traditional models such as XGBoost, Random Forest, and LightGBM by seamlessly integrating its salient features as plug-and-play embeddings, as shown in Figure 2. This strategy simplifies the feature engineering process and eliminates additional complexity in traditional models training.
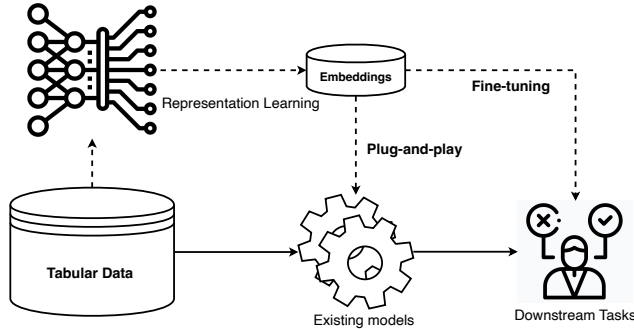

Figure 2: Illustration of usages of pre-trained encoders and embeddings. 1) The first option could be to fine-tune the pre-trained encoder directly for downstream tasks. This option usually achieves the optimal results but needs additional computation. 2) The second option is to concatenate the pre-trained embeddings with the original datasets, which requires no additional training and computation but still benefits the downstream tasks with considerable improvements in evaluation metrics.

4.1 Preliminaries for Experiments

4.1.1 Public Datasets

We evaluate the performance of ReConTab on a standard benchmark from [61], including Bank (BK) [52], Blastchar (BC) [37], Arrhythmia (AT) [48], Arcene (AR) [2], Shoppers (SH) [60], Volkert (VO) [30] and MNIST (MN) [79]. Five of the datasets focus on binary classification, and two of them focus on multi-class classification tasks. Importantly, the datasets employed in our experiments exhibit significant diversity. They encompass a wide range of characteristics, including varying sample sizes, ranging from 200 to 495,141 samples, and feature dimensions spanning from 8 to 784, encompassing both categorical and numerical features. Among these datasets, some exhibit missing data, while others are complete, and there is a mix of well-balanced datasets as well as those presenting highly skewed class distributions. This diversity allows us to comprehensively evaluate the performance and robustness of our proposed approach across a spectrum of real-world data scenarios.

6

4.1.2 Preprocessing of Datasets

To handle categorical features, we employ a backward difference encoder as described in [54]. Addressing the issue of missing data, we take a two-step approach. Initially, we remove any features that lack values across all samples. Subsequently, for the remaining missing values, we apply distinct imputation strategies based on the feature type. Numerical features are imputed using the mean value, while categorical features are filled with the most frequent category observed within the dataset. Moreover, we ensure data uniformity by employing a min-max scaler for dataset scaling. In cases involving image-based data, we flatten the images into vectors, treating them akin to tabular data. This approach aligns with established practices found in prior works such as [82] and [61].

4.1.3 Model Architectures

The ReConTab model architecture features a transformer-based shared network with three layers and two attention heads. This architecture is tailored for processing input data with a dimensionality determined by the shape of the training dataset. Additionally, the decoder remains a one-layer network with a sigmoid activation function. In the downstream fine-tuning stage, we add a linear layer after the encoder f to accommodate classification or regression tasks as needed.

4.1.4 Training Details

The ReConTab model is trained with a batch size of 128 over 1000 epochs, employing a learning rate of 0.0001. Gaussian masking is applied to the input data with a masking ratio of 0.3. The model's output dimension is set to half of the input data dimension. A contrastive loss with a margin of 2 is used during training, along with L2 normalization. Additionally, a regularization coefficient of 0.01 is applied to introduce a penalty term based on the L2 norm of the standard deviation of the Gaussian mask. During training, data is divided into two batches, and various loss components, including feature reconstruction loss, classification loss, contrastive loss, and regularization penalty, are computed to guide the optimization process. These training configurations ensure effective representation learning while controlling model behavior.

4.1.5 Metrics

Given that the majority of the tasks in our analysis involve binary classification, we employ the AUROC (Area Under the Receiver Operating Characteristic curve) as our primary metric for assessing performance. AUROC effectively quantifies the model's ability to distinguish between the two classes in the dataset. However, for the two multi-class datasets, VO and MN, we utilize accuracy on the test set as the metric for comparing performance.

4.2 Results on the Benchmarks

We show performance comparisons using chosen datasets and present the summarized results in Table 1. These results encompass evaluations employing both traditional models and more recent deep-learning techniques. In the majority of cases, ReConTab exhibits remarkable improvements, outperforming all baseline methods and reaffirming its superiority across a range of datasets and scenarios. However, it is important to note that, on BK, SH, and VO datasets, ReConTab achieved suboptimal results when compared to the best models. This observation aligns with previous research conclusions that the tabular domain presents unique challenges, with no single method universally excelling [27]. Nonetheless, ReConTab still gives the best performance over all of the deep-learning-based models and the second-best results over all of the methods. Meanwhile, this outcome warrants further investigation to uncover the specific factors contributing to this variation in performance.

4.3 Results as Plug-and-Play Embeddings

As previously mentioned, ReConTab has learned features that can significantly impact the decision boundaries in classification tasks. In the plug-and-play setting from Figure 2, our experimental results demonstrate the immense value of integrating these salient features with the original data as additional features. To be more specific, the performance of traditional models obtains relatively marginal improvement with only distilled features, as shown in the light gray columns of Table 1. While the improvement is relatively modest, it aligns with our expectations. The potential absence

Dataset size	45211			7043			452			200			12330			58310			518012		
Feature size	16			20			226			783			17			147			54		
Dataset	**BK**			**BC**			**AT**			**AR**			**SH**			**VO★**			**MN★**		
Raw Feature (x)	✓		✓	✓		✓	✓		✓	✓		✓	✓		✓	✓		✓	✓		✓
Distilled Feature (s)		✓	✓		✓	✓		✓	✓		✓	✓		✓	✓		✓	✓		✓	✓
Logistic Reg.	0.907	0.907	0.909	0.892	0.892	0.895	0.862	0.864	0.866	0.916	0.914	0.918	0.870	0.871	0.873	0.539	0.540	0.543	0.899	0.902	0.905
Random Forest	0.891	0.892	0.894	0.879	0.880	0.884	0.850	0.856	0.861	0.809	0.809	0.811	0.929	0.928	0.930	0.663	0.665	0.669	0.938	0.938	0.942
XGboost	0.929	0.928	0.930	0.906	0.903	0.906	0.870	0.871	0.883	0.824	0.822	0.826	0.925	0.925	0.927	**0.690**	0.690	0.692	0.958	0.959	0.963
LightGBM	**0.939**	0.933	0.939	0.910	0.909	0.912	0.887	0.888	0.907	0.821	0.822	0.825	**0.932**	0.933	0.936	0.679	0.680	0.682	0.952	0.953	0.954
CatBoost	0.925	0.928	0.932	0.912	0.910	0.914	0.879	0.880	0.889	0.825	0.827	0.833	0.931	0.932	0.935	0.664	0.665	0.670	0.956	0.958	0.968
MLP	0.915	0.919	0.920	0.892	0.893	0.898	0.902	0.904	0.908	0.903	0.904	0.904	0.887	0.887	0.890	0.631	0.631	0.636	0.939	0.940	0.940
VIME	0.766	-	-	0.510	-	-	0.653	-	-	0.610	-	-	0.744	-	-	0.623	-	-	0.958	-	-
TabNet	0.918	-	-	0.796	-	-	0.521	-	-	0.541	-	-	0.914	-	-	0.568	-	-	**0.968**	-	-
TabTransformer	0.913	-	-	0.817	-	-	0.700	-	-	0.868	-	-	0.927	-	-	0.580	-	-	0.887	-	-
ReConTab(Self-Sup.)	0.908	-	-	0.898	-	-	0.873	-	-	0.887	-	-	0.920	-	-	0.619	-	-	0.956	-	-
ReConTab(Semi-Sup.)	0.929	-	-	**0.913**	-	-	**0.907**	-	-	**0.918**	-	-	0.931	-	-	0.680	-	-	**0.968**	-	-

" − " indicates the experiments are not applicable for the corresponding methods to demonstrate the benefits of plug-and-play embeddings.

Table 1: Comparison of different methods on the classification tasks. For each method and dataset, we report three categories 1) raw features only, 2) salient features only, 3) plug-and-play features. The best results are shown in **Bold**, second-best results are Underlined. Columns added with ★ are multi-class classification tasks, reporting accuracy. The other results of binary classification tasks are evaluated with AUROC.

Ratio	0.0	0.1	0.2	0.3	0.4	0.5	0.6
BK	0.918	0.920	0.928	**0.929**	0.922	0.917	0.881
BC	0.889	0.897	0.906	**0.913**	0.910	0.901	0.896
AT	0.889	0.894	0.901	**0.905**	0.903	0.890	0.884
AR	0.904	0.911	0.913	**0.918**	0.915	0.909	0.901
SH	0.902	0.914	0.924	**0.931**	0.920	0.909	0.904
VO★	0.667	0.674	0.676	0.680	**0.681**	0.670	0.663
MN★	0.935	0.942	0.951	**0.959**	**0.959**	0.941	0.932

Table 2: Ablation of corruption ratio. Columns added with ★ are multi-class classification tasks, reporting their accuracy. The other results of binary classification tasks are evaluated with AUC.

of original information in this scenario results in a less substantial performance boost. Larger gains without fine-tuning come from the concatenation of original and distilled features. Notably, this integration enhances the performance of every method, leading to improvements in evaluation metrics (e.g., AUROC) across various datasets, as shown in the dark gray columns of Table 1.

4.4 Ablation Studies

In this section dedicated to ablation studies, we delve into the crucial components of ReConTab, assessing the significance of the parameter, i.e., feature corruption rate. Our analysis encompasses all the datasets listed in Table 1, employing consistent data preprocessing and optimization strategies throughout the experiments. In Table 2, we thoroughly examine the most advantageous feature corruption ratio. After extensive analysis, we find that the optimal corruption ratio is approximately 0.3. Therefore, we've adopted this value as the default for all previously reported experiments. However, it's important to emphasize that this chosen ratio may not always be the best fit for every dataset. Additionally, we've noticed interesting patterns in the datasets themselves. Datasets with more complex features, like VO or MN, tend to benefit from larger corruption ratios because they often contain redundant features. This observation aligns with previous research discussed in [29] regarding tabular data. On the flip side, for datasets with simpler, lower-dimensional features like BC, using smaller corruption ratios in our experiments might lead to better results.

5 Conclusion

As we observe the evolution of potent representation learning techniques tailored for different types of data from computer vision and natural language processing, we embark on a journey to extend their remarkable performance into new domains, such as tabular data. Drawing inspiration from related endeavors that address this challenge from the vantage points of contrastive learning and generative modeling, we present ReConTab — an innovative self- and semi-supervised framework designed for representation learning and feature distillation. The features learned through ReConTab exhibit superior performance in downstream tasks, obviating the need for extensive exploration of hand-crafted features. Furthermore, these features manifest as discernible, low-dimensional representations that seamlessly enhance the capabilities of various traditional models. We hold a strong conviction that this research marks a pivotal milestone in the pursuit of more representative, efficient, and structured representations for tabular data.

References

[1] Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6679–6687, 2021.

[2] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

[3] Sarkhan Badirli, Xuanqing Liu, Zhengming Xing, Avradeep Bhowmik, Khoa Doan, and Sathiya S Keerthi. Gradient boosting neural networks: Grownet. *arXiv preprint arXiv:2002.07971*, 2020.

[4] Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. Scarf: Self-supervised contrastive learning using random feature corruption. *arXiv preprint arXiv:2106.15147*, 2021.

[5] Christopher M Bishop, Michael E Tipping, et al. Bayesian regression and classification. *Nato Science Series sub Series III Computer And Systems Sciences*, 190:267–288, 2003.

[6] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[7] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[8] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[9] Chang Che, Bo Liu, Shulin Li, Jiaxin Huang, and Hao Hu. Deep learning for precise robot position prediction in logistics. *Journal of Theory and Practice of Engineering Science*, 3(10):36–41, Oct. 2023.

[10] Suiyao Chen. *Some Recent Advances in Design of Bayesian Binomial Reliability Demonstration Tests*. PhD thesis, University of South Florida, 2020.

[11] Suiyao Chen, William D Kearns, James L Fozard, and Mingyang Li. Personalized fall risk assessment for long-term care services improvement. In *2017 Annual Reliability and Maintainability Symposium (RAMS)*, pages 1–7. IEEE, 2017.

[12] Suiyao Chen, Nan Kong, Xuxue Sun, Hongdao Meng, and Mingyang Li. Claims data-driven modeling of hospital time-to-readmission risk with latent heterogeneity. *Health care management science*, 22:156–179, 2019.

[13] Suiyao Chen, Lu Lu, and Mingyang Li. Multi-state reliability demonstration tests. *Quality Engineering*, 29(3):431–445, 2017.

[14] Suiyao Chen, Lu Lu, Yisha Xiang, Qing Lu, and Mingyang Li. A data heterogeneity modeling and quantification approach for field pre-assessment of chloride-induced corrosion in aging infrastructures. *Reliability Engineering & System Safety*, 171:123–135, 2018.

[15] Suiyao Chen, Lu Lu, Qiong Zhang, and Mingyang Li. Optimal binomial reliability demonstration tests design under acceptance decision uncertainty. *Quality Engineering*, 32(3):492–508, 2020.

[16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[17] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[18] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[19] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[20] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. L2 regularization for learning kernels. *arXiv preprint arXiv:1205.2653*, 2012.

[21] Ian Covert, Uygar Sumbul, and Su-In Lee. Deep unsupervised feature selection. ' ', 2019.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[23] Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M Hospedales. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 39(3):42–62, 2022.

[24] Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. Masked autoencoders as spatiotemporal learners. *Advances in neural information processing systems*, 35:35946–35958, 2022.

[25] Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Transactions on Knowledge and Data Engineering*, 33(6):2493–2504, 2019.

[26] Zhabiz Gharibshah and Xingquan Zhu. Local contrastive feature learning for tabular data. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3963–3967, 2022.

[27] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943, 2021.

[28] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[29] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in Neural Information Processing Systems*, 35:507–520, 2022.

[30] Isabelle Guyon, Lisheng Sun-Hosoya, Marc Boullé, Hugo Jair Escalante, Sergio Escalera, Zhengying Liu, Damir Jajetic, Bisakha Ray, Mehreen Saeed, Michéle Sebag, Alexander Statnikov, WeiWei Tu, and Evelyne Viegas. Analysis of the automl challenge series 2015-2018. In *AutoML*, Springer series on Challenges in Machine Learning, 2019.

[31] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[32] Trevor J Hastie and Daryl Pregibon. Generalized linear models. In *Statistical models in S*, pages 195–247. Routledge, 2017.

[33] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[34] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[36] Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020.

[37] IBM. Telco customer churn (11.1.3+), 2019.

[38] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[39] Jincen Jiang, Xuequan Lu, Lizhi Zhao, Richard Dazeley, and Meili Wang. Masked autoencoders in 3d point cloud representation learning. *arXiv preprint arXiv:2207.01545*, 2022.

[40] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[41] Guolin Ke, Zhenhui Xu, Jia Zhang, Jiang Bian, and Tie-Yan Liu. Deepgbm: A deep learning framework distilled by gbdt for online prediction tasks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 384–394, 2019.

[42] Guolin Ke, Jia Zhang, Zhenhui Xu, Jiang Bian, and Tie-Yan Liu. Tabnn: A universal neural network solution for tabular data. ' ', 2018.

[43] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[44] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *Advances in neural information processing systems*, 30, 2017.

[45] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1920–1929, 2019.

[46] Xin Li, Yuhong Guo, and Dale Schuurmans. Semi-supervised zero-shot classification with label representation learning. In *Proceedings of the IEEE international conference on computer vision*, pages 4211–4219, 2015.

[47] Konstantinos G Liakos, Patrizia Busato, Dimitrios Moshou, Simon Pearson, and Dionysis Bochtis. Machine learning in agriculture: A review. *Sensors*, 18(8):2674, 2018.

[48] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[49] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[50] Lukas Meier, Sara Van De Geer, and Peter Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 70(1):53–71, 2008.

[51] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[52] Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

[53] Sergei Popov, Stanislav Morozov, and Artem Babenko. Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312*, 2019.

[54] Kedar Potdar, Taher S Pardawala, and Chinmay D Pai. A comparative study of categorical variable encoding techniques for neural network classifiers. *International journal of computer applications*, 175(4):7–9, 2017.

[55] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 2002.

[56] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[57] Adnan Qayyum, Junaid Qadir, Muhammad Bilal, and Ala Al-Fuqaha. Secure and robust machine learning for healthcare: A survey. *IEEE Reviews in Biomedical Engineering*, 14:156–180, 2020.

[58] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

[59] Colorado J Reed, Ritwik Gupta, Shufan Li, Sarah Brockman, Christopher Funk, Brian Clipp, Salvatore Candido, Matt Uyttendaele, and Trevor Darrell. Scale-mae: A scale-aware masked autoencoder for multiscale geospatial representation learning. *arXiv preprint arXiv:2212.14532*, 2022.

[60] C Okan Sakar, S Olcay Polat, Mete Katircioglu, and Yomi Kastro. Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and lstm recurrent neural networks. *Neural Computing and Applications*, 31:6893–6908, 2019.

[61] Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C Bayan Bruss, and Tom Goldstein. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*, 2021.

[62] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1161–1170, 2019.

[63] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[64] Ran Tao, Pan Zhao, Jing Wu, Nicolas F Martin, Matthew T Harrison, Carla Ferreira, Zahra Kalantari, and Naira Hovakimyan. Optimizing crop management with reinforcement learning and imitation learning. *arXiv preprint arXiv:2209.09991*, 2022.

[65] Shuting Tao, Peng Peng, and Hongwei Wang. Supervised contrastive learning with tpe-based bayesian optimization of tabular data for imbalanced learning. *arXiv preprint arXiv:2210.10824*, 2022.

[66] Bingjie Wang, Lu Lu, Suiyao Chen, and Mingyang Li. Optimal test design for reliability demonstration under multi-stage acceptance uncertainties. *Quality Engineering*, 0(0):1–14, 2023.

[67] Chen Wang, Xu Wu, and Tomasz Kozlowski. Sensitivity and uncertainty analysis of trace physical model parameters based on psbt benchmark using gaussian process emulator. *Proc. 17th Int. Topl. Mtg. Nuclear Reactor Thermal Hydraulics (NURETH-17)*, pages 3–8, 2017.

[68] Chen Wang, Xu Wu, and Tomasz Kozlowski. Surrogate-based bayesian calibration of thermal-hydraulics models based on psbt time-dependent benchmark data. In *Proc. ANS Best Estimate Plus Uncertainty International Conference, Real Collegio, Lucca, Italy*, 2018.

[69] Chen Wang, Xu Wu, and Tomasz Kozlowski. Gaussian process–based inverse uncertainty quantification for trace physical model parameters using steady-state psbt benchmark. *Nuclear Science and Engineering*, 193(1-2):100–114, 2019.

[70] Chen Wang, Xu Wu, and Tomasz Kozlowski. Inverse uncertainty quantification by hierarchical bayesian inference for trace physical model parameters based on bfbt benchmark. *Proceedings of NURETH-2019, Portland, Oregon, USA*, 2019.

[71] Chen Wang, Xu Wu, and Tomasz Kozlowski. Inverse uncertainty quantification by hierarchical bayesian modeling and application in nuclear system thermal-hydraulics codes. *arXiv preprint arXiv:2305.16622*, 2023.

[72] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the web conference 2021*, pages 1785–1797, 2021.

[73] Raymond E Wright. Logistic regression. ' ', 1995.

[74] Jing Wu, Jennifer Hobbs, and Naira Hovakimyan. Hallucination improves the performance of unsupervised visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16132–16143, 2023.

[75] Jing Wu, Naira Hovakimyan, and Jennifer Hobbs. Genco: An auxiliary generator from contrastive learning for enhanced few-shot learning in remote sensing. *arXiv preprint arXiv:2307.14612*, 2023.

[76] Jing Wu, David Pichler, Daniel Marley, David Wilson, Naira Hovakimyan, and Jennifer Hobbs. Extended agriculture-vision: An extension of a large aerial image dataset for agricultural pattern analysis. *arXiv preprint arXiv:2303.02460*, 2023.

[77] Jing Wu, Ran Tao, Pan Zhao, Nicolas F Martin, and Naira Hovakimyan. Optimizing nitrogen management with deep reinforcement learning and crop simulations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1712–1720, 2022.

[78] Yao Wu, Donghua Zhu, and Xuefeng Wang. Contrastive learning enhanced deep neural network with serial regularization for high-dimensional tabular data. *Expert Systems with Applications*, 228:120243, 2023.

[79] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[80] Decheng Xu, Shujie Hu, Dainan Zhang, Yongqiang Xiong, Yu Yang, and Yong Ran. Importance of sporopollenin structure and accessibility in the sorption of phenanthrene by biota spores and pollens. *Environmental science & technology*, 53(24):14285–14295, 2019.

[81] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. Tabert: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*, 2020.

[82] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33:11033–11043, 2020.

[83] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320, 2005.