STab: Self-supervised Learning for Tabular Data

Ehsan Hajiramezanali*, **Nathaniel Diamant, Gabriele Scalia, Max W. Shen**

Genentech
hajiramezanali.ehsan@gene.com

Abstract

Self-supervised learning has drawn recent interest for learning generalizable, transferable and robust representations from unlabeled tabular data. Unfortunately, unlike its image and language counterparts which have unique spatial or semantic structure information, it is difficult to design an effective augmentation method generically beneficial to downstream tasks in the tabular setting, owing to its lack of common structure and diverse nature. On the other hand, most existing augmentation methods are domain-specific (such as rotation in vision, token masking for NLP, and edge dropping for graphs), making them less effective for real-world tabular data. This significantly limits tabular self-supervised learning and hinders progress in this domain. Aiming to fill this crucial gap, we propose **STab**, an augmentation-free self-supervised representation learning based on stochastic regularization techniques that does not rely on negative pairs, to capture highly heterogeneous and non-structured information in tabular data. Our experiments show that **STab** achieves state-of-the-art performance compared to existing contrastive and pretext task self-supervised methods.

1 Introduction

Human learning in the real world builds mental representations that are robust to different views or distortions of an identity. With this in mind, when designing algorithms that imitate the human learning process, we seek a multi-view learning model that can learn representations invariant to a family of viewing conditions. Contrastive learning between multiple views of the data often fits such a description well by bringing two views of the same scene together in the representation space, while pushing those of different scenes (*negative samples*) apart (Tian et al., 2020a). However, their performance critically depends on the choice of input augmentations. In addition, these methods rely on memory banks, large batch sizes, or customized mining strategies to retrieve the negative pairs (Grill et al., 2020). Although there has been a range of approaches that broadly tackle the issue of contrastive representation learning in the vision and natural language domains, they fall short of proposing a complete range of augmentation methods applicable across domains and in particular, ones that can be applied to the tabular setting. More specifically, the augmentation steps to generate views or corruptions are mostly domain-specific (e.g. cropping, rotation, color transformation in vision, token masking in NLP, node/edge dropping in graph), making them less effective in the tabular data commonly used in many fields such as healthcare, advertisement, finance, etc. (Yoon et al., 2020; Ucar et al., 2021; Bahri et al., 2021).

We, therefore, seek a well-designed augmentation-free self-supervised representation learning to capture highly heterogeneous and non-structured information in tabular data in the vein of (Li et al., 2022; Gao et al., 2021; Verma et al., 2021). More specifically, instead of applying augmentations over the input samples to make different views of data for contrastive learning, we propose to apply

*Corresponding author

NeurIPS 2022 Workshop on Table Representation Learning.

augmentations to every layer of encoders. Note that this can be seen as a stochastic regularization technique rather than an augmentation method. As a result, the proposed **Self-supervised learning for Tabular data (STab)** relies on two (or multiple) weight-sharing neural networks with different regularizations applied to a single input. By exploiting the stop-gradient operation technique (Chen and He, 2021), the proposed weight-sharing networks can model invariance with respect to more complicated regularizations while it will not converge to an undesired trivial solution.

2 Related Works

Most self-supervised methods for representation learning can be categorized as either auxiliary handcrafted prediction tasks or contrastive tasks. Many of these approaches are appropriate only for computer vision and natural language. In particular, surrogate classes prediction, rotation degree predictions, colorization (Larsson et al., 2016), relative patches prediction (Doersch et al., 2015; Doersch and Zisserman, 2017), image de-noising (Laine et al., 2019), image jigsaw puzzle (Noroozi and Favaro, 2016), and next/previous words predictions (Devlin et al., 2018), have been shown to be useful pretext tasks. Yet, even with suitable architectures, these methods are often outperformed by contrastive approaches which avoid a costly generation step in pixel space. These methods bring the representations of different views of the same image closer (positive pairs) and spread representations of views from different images (negative pairs) apart. For example, contrastive predictive coding (Oord et al., 2018), contrastive multi-view coding (Tian et al., 2020b), and SimCLR (Chen et al., 2020) are pioneer works in this regard.

In the tabular setting, Yoon et al. (2020) applies a de-noising autoencoder with a classifier attached to its representation layer. The corrupted input data through a random binary mask network is fed to the encoder. While the decoder tries to re-construct the uncorrupted original input similar to a de-noising autoencoder, its classifier predicts the mask. However, this approach might not work well in very high-dimensional, small and noisy data sets since the model might easily become over-parameterized and be prone to overfitting to the data. Apart from that, training a classifier in this setting can be challenging since it needs to predict a very high dimensional, sparse, and imbalanced binary mask, similar to the problems observed when training a model on an imbalanced, binary dataset (Ucar et al., 2021). Verma et al. (2021) apply mixup to pairs of data and intermediate layer representations to create positive pairs for an InfoNCE loss. They find performance improvements in many domains including a tabular benchmark created by flattening and permuting image data. Verma et al. (2021) do not explore using stop gradient and so inherit the limitations of InfoNCE based approaches including reliance on large batch size and quadratic computational cost scaling in batch size.

Recently, Ucar et al. (2021) introduce SubTab, a new framework that turns the task of learning from tabular data into a multi-view representation learning problem by dividing the input features into multiple subsets. The paper argues that reconstructing the data from the subset of its features rather than its corrupted version in an autoencoder setting can better capture its underlying latent representation. In SubTab framework, the joint representation is learned through aggregating of latent representations of the subsets, similar to the collaborative inference. In addition to the aforementioned pretext task, one can add contrastive loss to its objective by comparing the pairs of projections from all subsets. However, using contrastive, and/or distance losses requires the combinations of projections, which makes the computational complexity quadratic during training and limits the number of subsets the model can use to divide the data. In addition, the reconstruction step is not suitable for high-dimensional data which is the case in most life science applications. Lastly, SCARF (Bahri et al., 2021), a contrastive learning based model, extends the existing input augmentations in the structured domains, and generates negative pair for a given input by selecting a random subset of its features and replacing them by random draws from the features' respective empirical marginal distributions. However, its applicability is questionable for high dimensional tabular data and might collapse to a trivial solution due to the high heterogeneity in this domain.

3 Method

STab takes an unlabeled tabular sample $\boldsymbol{x} \in \mathbb{R}^M$ as input. The input sample is then processed by two encoder multi-layer perceptron (MLP) networks f_1 and f_2. While the weight parameters of the encoders are shared, they have two different stochastic regularizations. In addition, a projection head g, which is a MLP, transforms the output of one encoder and matches it to the output of the other

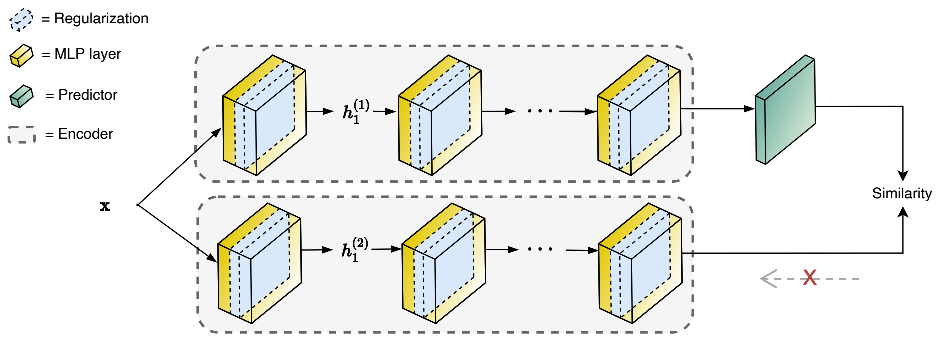

Figure 1: A schematic illustration of the proposed augmentation-free self-supervised learning for tabular data.

encoder. Denoting the two output vectors by $\boldsymbol{y}_1 = g(f_1(\boldsymbol{x}))$ and $\boldsymbol{z}_2 = f_2(\boldsymbol{x})$, we use the negative cosine distance as a measure of similarity:

$$\mathcal{D}(\boldsymbol{y}_1, \boldsymbol{z}_2) = -\frac{\boldsymbol{y}_1}{||\boldsymbol{y}_1||_2} \cdot \frac{\boldsymbol{z}_2}{||\boldsymbol{z}_2||_2}, \tag{1}$$

where $||\cdot||_2$ is l_2-norm. We optimize the following symmetric loss:

$$\mathcal{L} = \frac{1}{2}\mathcal{D}(\boldsymbol{y}_1, \boldsymbol{z}_2) + \frac{1}{2}\mathcal{D}(\boldsymbol{y}_2, \boldsymbol{z}_1). \tag{2}$$

To avoid converging to trivial solution, similar to Chen and He (2021), we need to make sure the encoder f_2 receives no gradient from $\boldsymbol{z}_2$ in the first term, but it receives gradients from $\boldsymbol{y}_2$ in the second term (and vice versa for f_1). Figure 1 depicts an overview of STab.

In order to regularize each encoder, we impose dynamic sparsity within the model. Specifically, similar to DropConnect (Wan et al., 2013), the fully-connected layers become a sparsely connected layer in which the connections are chosen at random during the training. Please note that this is different from considering the weights of the linear layer to be a fixed sparse matrix during training. Let's denote the output of hidden layers for view i by $\mathbb{H}^{(i)} = \{\boldsymbol{h}_j^{(i)}\}_{j=0}^{L}$ with $\boldsymbol{h}_0^{(1)} = \boldsymbol{h}_0^{(2)} = \boldsymbol{x}$ being the input data and L as the number of layers. For each layer of the encoders, the output is given as:

$$\boldsymbol{h}_j^{(i)} = \sigma\left((\boldsymbol{M}_j^{(i)} \odot \boldsymbol{W}_j)\, \boldsymbol{h}_{j-1}^{(i)}\right), \quad \text{for} \quad i = 1, 2 \tag{3}$$

where $\boldsymbol{M}_j^{(i)}$ is a binary matrix encoding the connection information for f_i and $\boldsymbol{M}_{j,mn}^{(i)} \sim \text{Bernoulli}(p_j^{(i)})$, $\boldsymbol{W}$ is the shared weight parameters across encoders, and σ is a non-linear activation function. Note that each element of the mask $\boldsymbol{M}_j^{(i)}$, i.e. $\boldsymbol{M}_{j,mn}^{(i)}$, is drawn independently for each sample during training, essentially instantiating a different connectivity for each sample seen.

3.1 Expectation-Maximization Interpretation

From another point of view, STab solves two underlying sub-problems based on two sets of implicit variables. Let's introduce a new set of variables as η and write the loss function as:

$$\mathcal{L}(\theta, \eta) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{M}}\left[||\mathcal{F}(\boldsymbol{x}; \theta, \boldsymbol{M}) - \eta_x||_2^2\right], \tag{4}$$

where $\mathcal{F}$ maps input data $\boldsymbol{x}$ to an output through a sequence of operations given the parameters $\theta = \{W_j\}_{j=1}^{L}$ and randomly drawn masks $\boldsymbol{M}$. The expectation is over the distribution of both tabular inputs and masks, and $||\cdot||_2^2$ is due to the aforementioned cosine similarity loss. Please note that η is not necessarily an output of a network, rather it is the argument of an optimization problem. With this formulation, we need to solve $\min_{\theta,\eta} \mathcal{L}(\theta, \eta)$. To solve such an optimization, we can alternate between solving these two sub-problems:

$$\theta_t \longleftarrow \arg\min_{\theta} \mathcal{L}(\theta, \eta_{t-1}), \qquad \eta_t \longleftarrow \arg\min_{\eta} \mathcal{L}(\theta_t, \eta). \tag{5}$$

Table 1: Performance of our STab and baselines in terms of classification accuracy (in %). * In STab w/ DropOut we used DropOut to mask weight instead of DropConnect.

Method	Income	Gesture	Robot	Theorem
Raw features	82.28±0.08	46.93±1.05	68.46±1.34	46.96±0.1
VIME-self	82.43±0.16	46.08±0.37	74.23±1.21	44.99±0.9
SubTab	83.97±0.31	52.03±0.98	88.21±0.72	50.81±0.76
SCARF	83.96±0.23	52.28±1.04	83.51±0.86	51.06±1.09
STab w/ DropOut *	81.37±1.13	51.81±0.95	81.28±0.85	48.88±1.22
STab	**84.53 ±0.11**	**53.08 ±0.91**	**88.40 ±0.82**	**55.06 ±0.28**

Since the gradient does not back-propagate to η_{t-1}, we can use SGD or Adam to solve the first sub-problem. For the second one, one can minimize the following expectation:

$$\mathbb{E}_{\boldsymbol{M}} \left[||\mathcal{F}(\boldsymbol{x}; \theta_t, \boldsymbol{M}) - \eta_x||_2^2 \right], \tag{6}$$

that can be re-written as $\eta_{x,\,t} \longleftarrow \mathbb{E}_{\boldsymbol{M}} \left[\mathcal{F}(\boldsymbol{x}; \theta_t, \boldsymbol{M})\right]$. This means η_x can be calculated from the average representation of input over the distribution of mask regularization. Let's consider $\boldsymbol{M}^{'}$ as a single sampling of the mask, we will have:

$$\eta_{x,\,t} \longleftarrow \mathcal{F}(\boldsymbol{x}; \theta_t, \boldsymbol{M}^{'}) \tag{7}$$

$$\theta_{t+1} \longleftarrow \arg\min_\theta \mathbb{E}_{\boldsymbol{x},\boldsymbol{M}} \left[||\mathcal{F}(\boldsymbol{x}; \theta, \boldsymbol{M}) - \mathcal{F}(\boldsymbol{x}; \theta_t, \boldsymbol{M}^{'})||_2^2 \right]. \tag{8}$$

Therefore, θ_t will be a constant and $\boldsymbol{M}^{'}$ can be seen as the second regularization mask due to its random nature. The predictor g is expected to minimize $\mathbb{E}_{\boldsymbol{z}} \left[||g(\boldsymbol{z}_1) - \boldsymbol{z}_2||_2^2 \right]$, where the optimal solution for each input sample will be $g(\boldsymbol{z}_1) = \mathbb{E}_{\boldsymbol{z}} \left[\boldsymbol{z}_2\right] = \mathbb{E}_{\boldsymbol{M}} \left[\mathcal{F}(\boldsymbol{x}; \theta, \boldsymbol{M})\right]$.

4 Results and Conclusions

To demonstrate the effectiveness of the proposed framework, we have conducted experiments on a diverse set of tabular datasets including UCI adult income (Income) (Kohavi et al., 1996), Gesture Phase Segmentation (Gesture) (Madeo et al., 2013), Wall Robot (Robot) (Freire et al., 2009), and First Order Theorem Proving (Theorem) (Bridge et al., 2014). Note that the last three datasets belong to OpenML-CC18 datasets (Bischl et al., 2017; Bahri et al., 2021). We compare our STab with existing self-supervised learning SOTA methods for tabular data. VIME-self (Yoon et al., 2020) and SubTab (Ucar et al., 2021) can be categorized as an autoencoder-based model, while SCARF is a contrastive model based on InfoNCE loss.

Following the experimental setting in Bahri et al. (2021), all encoders are four-layer [256, 256, 256, 256] dimensional fully-connected NN while the projection head is a two-layer [256, 256] dimensional fully-connected NN. We train SubTab by using two subsets with zero overlaps and using only reconstruction loss as suggested in the paper. For all models, we train and evaluate them with 10 different random seeds. Evaluation of these models is done by training a logistic regression model using the embeddings of the training set (i.e. 80% of the data), and by testing it using the embeddings of the test set (20% of the data). Similar to SCARF, we use ReLU as activation functions for all experiments. Please note that we follow SubTab for the preprocessing of each dataset.

Table 1 shows the proposed STab outperforms all the baselines. This proves that the stochastic regularization techniques used in STab is a more effective approach for modeling invariance than random augmentation over the input such as the one in SCARF (Bahri et al., 2021). Comparing DropConnect with DropOut in STab framework is demonstrating that DropConnect, which generalizes Dropout (Gal and Ghahramani, 2016) to the entire connectivity structure of a fully connected neural network layer, is a more powerful augmentation in the tabular settings and critical to achieve better performance compared to other baselines. A different interpretation of STab is through the lens of stochastic functions. It is well-known that neural networks with stochastic regularization are random functions (Gal and Ghahramani, 2016). We can interpret STab as siamese model with two different stochastic functions as encoders. Different stochasticities in the encoders produces different views of data. One possible avenue for future works is employing other classes of stochastic functions such as neural processes as encoder. Another avenue for further improvements is learning the drop rates of binary masks throughout a hierarchical Bayesian model or bi-level optimization which leads to a more flexible and versatile model.

References

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020a.

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.

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.

Talip Ucar, Ehsan Hajiramezanali, and Lindsay Edwards. Subtab: Subsetting features of tabular data for self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34, 2021.

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

Jiangmeng Li, Wenwen Qiang, Changwen Zheng, Bing Su, and Hui Xiong. MetAug: Contrastive Learning via Meta Feature Augmentation. In *Proceedings of the 39th International Conference on Machine Learning*, pages 12964–12978. PMLR, June 2022. URL https://proceedings.mlr.press/v162/li22r.html.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.552. URL https://aclanthology.org/2021.emnlp-main.552.

Vikas Verma, Thang Luong, Kenji Kawaguchi, Hieu Pham, and Quoc Le. Towards Domain-Agnostic Contrastive Learning. In *Proceedings of the 38th International Conference on Machine Learning*, pages 10530–10541. PMLR, July 2021. URL https://proceedings.mlr.press/v139/verma21a.html.

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.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593. Springer, 2016.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2051–2060, 2017.

Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. *Advances in Neural Information Processing Systems*, 32, 2019.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

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.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer, 2020b.

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.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066. PMLR, 2013.

Ron Kohavi et al. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pages 202–207, 1996.

Renata CB Madeo, Clodoaldo AM Lima, and Sarajane M Peres. Gesture unit segmentation using support vector machines: segmenting gestures from rest positions. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 46–52, 2013.

Ananda L Freire, Guilherme A Barreto, Marcus Veloso, and Antonio T Varela. Short-term memory mechanisms in neural network learning of robot navigation tasks: A case study. In *2009 6th Latin American Robotics Symposium (LARS 2009)*, pages 1–6. IEEE, 2009.

James P Bridge, Sean B Holden, and Lawrence C Paulson. Machine learning for first-order theorem proving. *Journal of automated reasoning*, 53(2):141–172, 2014.

Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Frank Hutter, Michel Lang, Rafael G Mantovani, Jan N van Rijn, and Joaquin Vanschoren. Openml benchmarking suites. *arXiv preprint arXiv:1708.03731*, 2017.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

A Appendix

A.1 More related works

A few recent works have explored applying augmentations to the model weights instead of the data. MetAug (Li et al., 2022) applies meta-learned augmentations to the output of the encoder portion of their neural networks. The augmentations are learned to balance instance discrimination and a regularization term. MetAug focuses on one layer of the neural network and the results are focused on natural image data, whereas STab augments multiple layers and is intended for non-structured data. Similar to STab, the sentence embedding method SimSCE (Gao et al., 2021) uses dropout on model weights rather than input augmentations. SimSCE outperforms input-augmentation baselines and is simple to implement. STab builds on this approach by removing the need for negative views by using stop gradient. STab also replaces DropOut with DropConnect, which we found led to better results (Table 1).