# Self Supervised Pre-training for Large Scale Tabular Data

**Sharad Chitlangia, Anand Muralidhar, Rajat Agarwal**
Amazon Ads
{chitshar,anandmur,agrajat}@amazon.com

## Abstract

In this paper, we tackle the problem of self supervised pre-training of deep neural networks for large scale tabular data in online advertising. Self supervised learning has recently been very effective for pre-training representations in domains such as vision, natural language processing, etc. But unlike these, designing self supervised learning tasks for tabular data is inherently challenging. Tabular data can consist of various types of data with high cardinality and range of feature values especially in a large scale real world setting. To that end, we propose a self supervised pre-training strategy that utilizes Manifold Mixup to produce data augmentations for tabular data and perform reconstruction on these augmentations using noise contrastive estimation and mean absolute error losses, both of which are particularly suitable for large scale tabular data. We demonstrate its efficacy by evaluating on the problem of click fraud detection on ads to obtain a 9% relative improvement on robot detection metrics over a supervised learning baseline and 4% over a contrastive learning experiment.

## 1 Introduction

Unsupervised deep learning in the recent years has seen massive advancements. In particular, there has been a burgeoning of works centered around self supervised learning (SSL) across domains such as vision [3], natural language processing [6], etc. One of the foundations spurting this growth is the ever increasing sizes of datasets such as ImageNet [5], Microsoft COCO [13], superglue [20], etc. Compared to the typical supervised learning framework, SSL boasts largely of pre-training representations on these datasets followed by fine-tuning to learn task specific representations.

Tabular data is ubiquitous and there has been some interest recently towards porting successful frameworks such as self supervised pre-training from other domains to tabular datasets. In this paper, we present a reconstruction based SSL framework for tabular data that is specifically designed to handle large scale tabular data. We utilize a data augmentation strategy using Manifold Mixup [17] that is generic and useable with all forms of tabular data. Manifold Mixup interpolates linearly between hidden state representations to produce perturbed datapoints in the hidden state manifold, that are in the vicinity of the original hidden state representations. The network is tasked to recover the original input from these perturbed datapoints. For instance, in case of a categorical input feature, the network learns to classify the correct category among other randomly sampled categories using noise contrastive estimation. We consider the scenario of online advertising wherein ads are displayed when a user visits a website. In such domains, features such as customer ID have very high cardinality and frequency based features have a very range of values. There is prevalence of robotic traffic that click on such ads. We evaluate our approach on large scale ads data on the problem of click fraud detection while comparing it to a supervised learning baseline and a contrastive SSL experiment to show its efficacy.

Accepted at Table Representation Learning Workshop at NeurIPS 2022

The rest of the paper is as follows: Section 2 summarizes and reviews related work, Section 3 describes our baseline and the proposed method, Section 4 defines the evaluation metrics and details the results of our approach, and we conclude in Section 5.

## 2 Related Work

**Self Supervised Learning** Self Supervised Learning (SSL) is a form of unsupervised learning where a model learns representations by utilizing invariant data transformations, commonly known as "pretext" tasks. SSL methods for vision [3] use image augmentations to create pairs of noisy views of the same image followed by contrastive learning with random negative samples. Another variant of loss commonly used in SSL is reconstruction loss and it is similar to denoising autoencoders [19] Corruption of data is central to SSL and various methods are used to create noisy versions of the original input such as data augmentation [3], introduction of noise [18], etc.

**Tabular data** Recently there has been some progress towards modelling tabular data with deep learning especially with introduction of transformer based models such as TabTransformers [10], TaBERT [21], etc. In the context of SSL, research on tabular data has witnessed some works like SubTab [16], VIME [22], SCARF [1]. SubTab performs reconstruction of input from a subset of its features in an autoencoder with the assumption that there exists some overlap in features. VIME [22] proposed pretext tasks to estimate mask vectors from corrupted data samples but these were not designed to handle embeddings for categorical features. SAINT [15] proposes several improvements to TabTransformers including *CutMix* on input features but utilizes cross-entropy loss for reconstruction of categorical features which is not scalable with the number of classes (categories).

**Online Advertising** Online advertising has large amounts of tabular data and problems studied in this domain such as click-through rate (CTR) prediction [4], conversion rate (CVR) prediction [14], contextual advertising [2], attribution modelling[7] and ad fraud detection [12] have typically utilized supervised learning. In this paper, we are concerned with the problem of ad fraud detection where the goal is to invalidate robotic clicks. Some of the available solutions include lists of entities (e.g. IP address, User Agent, etc) that are deemed entirely robotic. This leads to a very low precision solution as an entity will see a mixture of human and robotic traffic. Additionally, since these lists are static, they also suffer from low recall during emergent attacks. These reasons warrant the need to build sophisticated models that learn robotic activity patterns and are able to invalidate traffic at a very high precision and recall. Ad fraud detection also presents an ideal setting for testing of SSL techniques because there are no high quality labels that identify bots. Additionally, the domain is naturally adversarial and continuously evolving which requires algorithms to be highly robust.

## 3 Method

### 3.1 Baseline Supervised Model

We consider a website that displays ads to users and construct a variety of features for each ad click. These features provide the model with a discriminative ability to identify robots that click on ads from humans and include:

- **Frequency and velocity counters**: Volume and rate of clicks from a user computed over various time periods ranging from few minutes to hours. These features are critical for identifying emergent robotic attacks that involve a sudden burst of clicks from a robotic entity.
- **User-entity counters**: Distinct sessions and users from an IP address in the recent past. We also compute the corresponding maximum values seen in an hour. These features help to identify IP addresses that correspond to gateways of enterprises that may have many users behind them who perform a large number of ad clicks. We do not want to inadvertently mark such IPs as robotic.
- **Time of click**: Hour-of-day and day-of-week mapped to points on the unit circle. Human activity follows the regular diurnal and weekly activity patterns but robotic activity may not conform to it and these features help to consider activity patterns over time. Also mapping these features to the unit circle ensures a smooth transition from the last hour (or day) to the first one and the model does not see a sudden jump in the feature value.

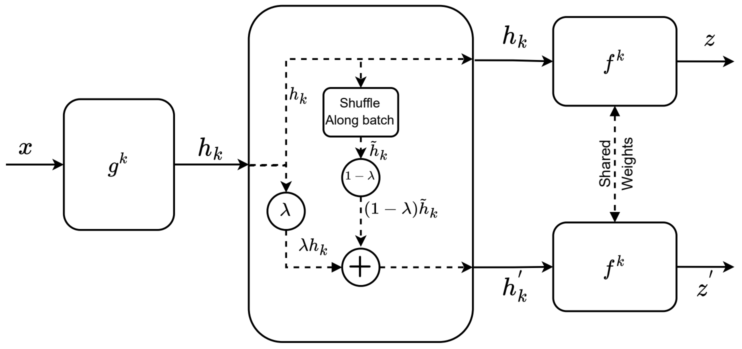


Figure 1: Manifold Mixup used to produce data augmentations

- **Logged-in**: Boolean feature that captures if the user was logged-in to the account. To be logged in requires a valid email address and/or phone number and requires more sophistication from a robot.

These features are largely numerical in nature. As these are counter based features, they have a large range of values.

**Embeddings** The features described so far cannot "remember" and learn about behavior of a robotic entity across clicks due to lack of any entity identifiers as features. Towards remedying this drawback, we consider embeddings for customer ID, user agent, IP 3-octet and features from the raw HTTP request made by a client to the website. Instead of directly learning embeddings for these categorical features as part of supervised learning, we pretrain the embeddings and later use them in the supervised model as non-trainable parameters. The rationale behind pre-training is two-fold: 1) Lack of good quality labels can lead to undesirable behavior learning 2) Overfitting to a few data points due to the high cardinality of these features ($\gg 10^3$). All of these features have a long tailed distribution. To tackle this, an out of vocabulary embedding is created that groups sparse feature values together. The sparsity is dependent on the number of clicks and the threshold is a hyperparameter.

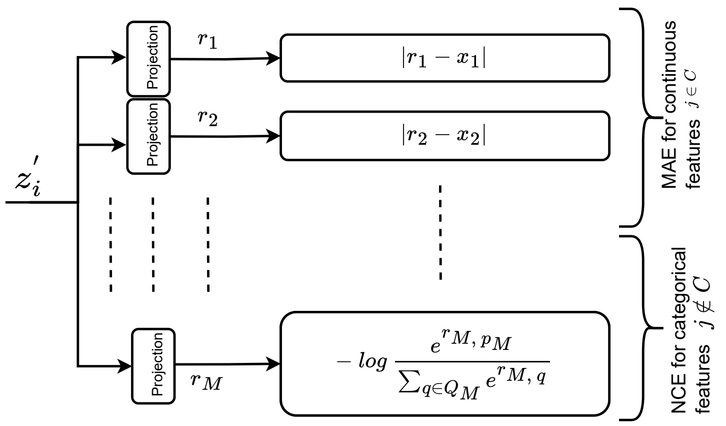
The features are used to train a DNN to solve a classification problem. The labels used for the classification problem act as proxy for humans [1] and are based on clicks that led to a purchase. The network has three fully-connected hidden layers with ReLU activation with the final layer connected to a sigmoid. Since the dataset is imbalanced with very few human (0) labels, we weigh the $i$-th training sample by $w_i = C/N_i$, where $N_i$ is total clicks in $(\text{hour}_i; \text{day}_i; \text{logged-in}_i; \text{label}_i)$ bucket and $C$ is a constant added for numerical stability. The network is optimized to minimize weighted binary cross-entropy loss. The network trains on all ad clicks in a week for one epoch. Next, we describe the methodology to create noisy samples for the pre-training task used in SSL.

## 3.2 Self Supervised Pretraining of Embeddings

**Data Augmentation** In tabular data, it is not straightfoward to create two views of the same data point as in vision. Techniques such as MixUp [24], CutMix[23], Feature Corruption [1], etc., that perform mixing directly on the input are not suitable for our usecase, as unlike images, our data is not very high dimensional. Hence, mixing inputs directly can lead to high mismatch and information loss. We leverage ideas from Manifold Mixup[17] for creating noisy representations from the latent representations of hidden states. Consider a DNN with an architecture same as Section 3.1 except that the output sigmoid layer is replaced with a latent representation of length equal to the number of input features [2]. Let's represent this network as $f(x) = f^k(g^k(x))$, where $g^k$ denotes portion of the neural network mapping the input $x$ to the hidden representation at layer $k$ and $f^k$ maps the hidden representation to the output. Let $k$ be randomly sampled among the three hidden layers of the network. Let $z = f(x)$ and choose $\tilde{x}$ randomly from the same training batch. We define the mixed representation in the $k$-th layer as: $h_k^{'} = \lambda \cdot g^k(x) + (1-\lambda) \cdot g^k(\tilde{x})$,. This is passed through rest of the network to get a noisy latent representation $z^{'} = f^k(h_k^{'})$ with $\lambda = 0.75$ to control the mixing.

[1]It is important to note here that not all non-human activity necessitates to robot activity.

[2]this is to form a 1 to 1 mapping between the inputs and outputs for reconstruction



$z_i'$
Projection
Projection
Projection
$r_1$
$r_2$
$r_M$
$|r_1 - x_1|$
$|r_2 - x_2|$
$-\,log\,\frac{e^{r_{M},\,p_M}}{\sum_{q\in Q_M} e^{r_{M},\,q}}$
MAE for continuous features $j \in C$
NCE for categorical features $j \notin C$

Figure 2: Reconstruction loss utilizing mean absolute error (MAE) for continuous features and noise contrastive estimation (NCE) for categorical features

Unlike isometric corruption with Gaussian noise, these perturbations are directed towards hidden representations of other data points. Figure 1 visually depicts Manifold Mixup.

**Pretext Task** For computing the reconstruction loss, we reconstruct the original input features from the noisy latent representation of each click. There is a 1-1 mapping between elements in the latent representation and input features and we pass each element in the latent output through an additional projection layer of 5 hidden nodes to get the reconstruction of each feature. In case of continuous features, the reconstruction is a scalar and we compute the mean absolute error with respect to the original value. Using the standard mean squared error for reconstruction loss like in [15] leads to a poor performance due to the large dynamic range of some features. In case of categorical features, we can compute the cross entropy loss at output of the projection layer but this is not feasible due to a large vocabulary size ($\gg 10^3$) for these features. Hence we approximate the reconstruction loss for categorical features with noise contrastive estimation (NCE)[8]. NCE approximates the cross entropy loss by reducing it to a binary classification problem between choosing the true category among other randomly sampled values.

Let a click be mapped to $M$ input features with $x = (x_j)_{j=1}^{M}$ and let the indices corresponding to continuous features lie in the set $\mathcal{C}$. We denote the $j$-th reconstructed continuous feature as $r_j$. Consider a categorical feature $x_j, j \notin \mathcal{C}$ and let $\{r_{j,p}\}_p$ denote the outputs of the projection layer where $p$ ranges over all possible values of $x_j$. Since it is not tractable to calculate $\{r_{j,p}\}_p$ for large cardinality of $p$, we compute $r_{j,p_j}$ where $p_j$ is the correct index for the $j$-th categorical feature and $\{r_{j,q}\}_{q\in\mathcal{Q}_j}$ where $\mathcal{Q}_j$ is a set of randomly chosen negative category indices for $x_j$. The reconstruction loss for $x$ is defined as:

$$\mathcal{L}_R = \frac{1}{N}\sum_x \Bigg( \underbrace{\sum_{j\in\mathcal{C}} |r_j - x_j|}_{\text{MAE}} - \underbrace{\sum_{j\notin\mathcal{C}} \log \frac{e^{r_{j,p_j}}}{\sum_{q\in\mathcal{Q}_j} e^{r_{j,q}}}}_{\text{NCE}} \Bigg) \tag{1}$$

## 4 Evaluation

### 4.1 Metrics

Due to lack of confident labels on robotic clicks, we cannot reliably measure standard metrics for a classification model like precision and recall. Hence we use other metrics that are closely aligned with the business impact, and also allow us to reliably compare models.

**Invalidation Rate (IVR)** This is defined as the fraction of clicks marked as robotic by the algorithm, i.e., IVR $= (\text{ROBOTIC CLICKS})/(\text{TOTAL CLICKS})$. IVR is indicative of recall of the model, but it cannot be viewed in isolation since a poorly-performing algorithm might incorrectly invalidate a significant volume of human clicks. Hence we always measure IVR in conjunction with False Positive Rate (defined next) to ensure that we measure recall without reducing the precision of detection.

Table 1: Comparing models at a particular FPR (%)

| Model | IVR % | Robotic Coverage % |
|---|---|---|
| **SSL Combined** | $1.09x$ | $1.07y$ |
| SSL Reconstruction | $1.09x$ | $y$ |
| SSL Contrastive | $1.05x$ | $0.96y$ |
| Supervised Baseline | $x$ | $y$ |

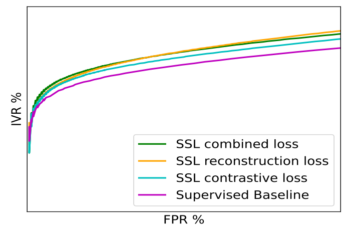


Figure 3: IVR-FPR curves of models

**False Positive Rate (FPR)** This is defined as the fraction of human clicks invalidated by the algorithm, i.e., FPR $= \frac{\text{HUMAN CLICKS MARKED AS ROBOTIC}}{\text{TOTAL HUMAN CLICKS}}$. But we cannot directly measure FPR since we do not have confident labels for human clicks. We consider ad clicks that led to a purchase as a proxy for human clicks and measure a proxy metric FPR$_{\text{PROXY}}$ defined as $\frac{\text{PURCHASING AD CLICKS MARKED AS ROBOTIC}}{\text{TOTAL PURCHASING AD CLICKS}}$. In rest of the paper FPR refers to FPR$_{\text{PROXY}}$, while noting that there are two underlying assumptions. First is that all purchasing clicks are human and second is that purchasing clicks are a representative sample of human clicks. We have detected robots that make purchases but these are small in number and do not substantially change the FPR.

**Robotic coverage** We define a heuristic rule to identify traffic that has a high likelihood to be robotic. If a session makes more than $\mathcal{K}$ ad clicks in a hour, then it is likely to be robotic. We choose $\mathcal{K}$ to be a large number so that humans are less likely to click on ads at that rate. We define *Robotic coverage* as the percentage of such clicks that were marked as robotic by the model. This metric can also be considered as an indicator of recall for a model.

### 4.2 Results

We benchmark the performance of our reconstruction based SSL technique against the supervised learning baseline and a contrastive learning experiment similar to [1, 15]. Except for the final loss function, all details in the contrastive learning experiment remain the same as the reconstruction based SSL framework. The positive sample is produced via Manifold Mixup with a high mixing coefficient (similar to how it is done for the reconstruction loss) and the negative samples for the contrastive loss are randomly picked from the current batch. The output of the first set of projection layers ($z_i$) is used for the contrastive loss computation. A more detailed description and visual depiction of the contrastive learning experiment is given in Section A.1. We skip comparison with replaced token detection (RTD) as proposed in TabTransformers [10] as the task is trivial to solve in our domain due to the large cardinality of categorical features and its inability to work with real valued inputs. We also skip data augmentation techniques that perform mixing/cropping directly on the inputs such as from SubTab [16], SCARF [1] and VIME [22] due to aforementioned reasons in Section 3.2.

Figure 3 shows the IVR-FPR curves of the models. A model with higher IVR at particular FPR means that the model is able to invalidate more traffic at that FPR and is hence likely to catch more bots than a model with a lower IVR at the same FPR. *Supervised Baseline* is the model that directly learns embeddings in a supervised setting. *SSL contrastive loss* and *SSL reconstruction loss* are the models where SSL is used to pretrain embeddings using contrastive and reconstruction losses respectively.

Self supervised pre-training of embeddings improves performance when compared to the the supervised baseline model. The performance of reconstruction based self supervised pretraining exceeds the contrastive learning based self supervised pretraining. We experiment with a combined formulation where the latent representation of all features is passed to both the reconstruction and contrastive losses. We observe that embeddings from this experiment have the highest robotic coverage while maintaining a similar performance on the IVR-FPR curve as the reconstruction based SSL model approach. This suggests that there is very less overshadowing effect between the contrastive and reconstruction losses and using both losses results in the best performing model.

### 4.3 Conclusion

We propose a methodology for tackling self-supervised pre-training for large scale tabular data. Using Manifold Mixup ensures that we are able to utilize this framework for any form of tabular data. Noise contrastive estimation (NCE) and mean absolute error (MAE) losses enable tackling the problem of large cardinality of features in categorical data and large range of values in numerical data. We expect our approach to be generally useful to all tabular data practitioners but particularly useful for domains with large scale tabular data such as Recommender System, Advertising and healthcare. For future work, we aim to experiment with hard negative mining for both contrastive and reconstruction losses, since this crucial for applications in vision [9, 11] and we expect a similar behavior in the tabular data setting.

## References

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

[2] Andrei Broder, Marcus Fontoura, Vanja Josifovski, and Lance Riedel. A semantic approach to contextual advertising. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 559–566, 2007.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

[4] Talking Data. Talkingdata adtracking fraud detection challenge. https://ailab.criteo.com/download-criteo-1tb-click-logs-dataset/, 2013.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

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

[7] Diemert Eustache, Meynet Julien, Pierre Galland, and Damien Lefortier. Attribution modeling increases efficiency of bidding in display advertising. In *Proceedings of the AdKDD and TargetAd Workshop, KDD, Halifax, NS, Canada, August, 14, 2017*, page To appear. ACM, 2017.

[8] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[9] 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.

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

[11] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning, 2020.

[12] Criteo AI Labe. Click-through rate (ctr) prediction. Kaggle, April 2018.

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014.

[14] Quan Lu, Shengjun Pan, Liang Wang, Junwei Pan, Fengdan Wan, and Hongxia Yang. A practical framework of conversion rate prediction for online display advertising. In *Proceedings of the ADKDD'17*, ADKDD'17, New York, NY, USA, 2017. Association for Computing Machinery.

[15] Gowthami Somepalli, Avi Schwarzschild, Micah Goldblum, C. Bayan Bruss, and Tom Goldstein. SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training, 2022.

[16] Talip Ucar, Ehsan Hajiramezanali, and Lindsay Edwards. Subtab: Subsetting features of tabular data for self-supervised representation learning, 2021.

[17] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6438–6447. PMLR, 09–15 Jun 2019.

[18] Vikas Verma, Thang Luong, Kenji Kawaguchi, Hieu Pham, and Quoc Le. Towards domain-agnostic contrastive learning. In *International Conference on Machine Learning*, pages 10530–10541. PMLR, 2021.

[19] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery.

[20] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

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

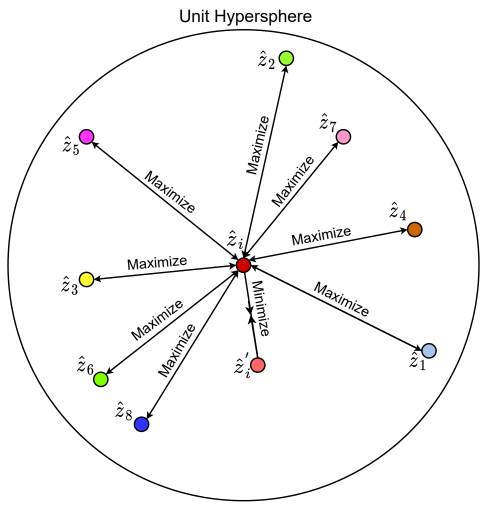
[22] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. Vime: Extending the success of self- and semi-supervised learning to tabular domain. In *NeurIPS*, 2020.

[23] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019.

[24] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2017.

## A Appendix

### A.1 Losses



(a) Contrastive Loss

Figure 4: Self Supervised Learning Setup on top of DNN

**Contrastive Loss** Figure 4a depicts the contrastive loss where $\hat{z}_i$ is the $L2$-normalized output corresponding to $x_i$ post application of $f^k(g^k(\cdot))$, $\hat{z}_i^{'}$ is the $L2$-normalized output corresponding to $h_i^{'}$ post application of $f^k(\cdot)$, and $z_j$ with $j \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ are the normalized outputs of randomly sampled datapoints from the current batch.

In this case, distance between the latent representation of an input and the noisy version described above is minimized while distance between representations of two dissimilar data points is maximized.

Consider a batch of size $N$ with inputs $\{x_i\}_{i=1}^N$ and latent representations $z_i = f(x_i), \forall i$. For each input $x_i$, we also compute a noisy representation $z_i^{'}$ as described previously. The contrastive loss is defined as

$$\mathcal{L}_{\text{C}} \;:=\; -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\Big(z_i^T z_i^{'}/(\|z_i\|\|z_i^{'}\|T)\Big)}{\sum_{j=1}^{N} \exp\big(z_i^T z_j/(\|z_i\|\|z_j\|T)\big)}, \tag{2}$$

where we choose the temperature $T$ to be $0.2$ in our experiments.