MET: Masked Encoding for Tabular Data

Kushal Majmundar
Google AI Research Lab,
Bengaluru, India 560016
majak@google.com

Sachin Goyal
Carnegie Mellon University
Pittsburgh, PA 15213
sachingo@andrew.cmu.edu

Praneeth Netrapalli
Google AI Research Lab,
Bengaluru, India 560016
pnetrapalli@google.com

Prateek Jain
Google AI Research Lab,
Bengaluru, India 560016
prajain@google.com

Abstract

This paper proposes *Masked Encoding for Tabular Data (MET)* for learning self-supervised representations from *tabular data*. Tabular self-supervised learning (tabular-SSL) – unlike structured domains like images, audio, text – is more challenging, since each tabular dataset can have a completely different structure among its features (or coordinates), that is hard to identify a priori. MET attempts to circumvent this problem by assuming the following hypothesis: the observed tabular data features come from a latent graphical model and the downstream tasks are significantly easier to solve in the latent space. Based on this hypothesis, MET uses random masking based encoders to learn a positional embedding for each coordinate, which would in turn capture the latent structure between coordinates. Extensive experiments on multiple standard benchmarks for tabular data demonstrate that MET significantly outperforms all the current baselines. For example, on Criteo dataset – a large-scale click prediction dataset – MET achieves as much as 5% improvement over the current state-of-the-art (SOTA) while purely supervised learning based approaches have been able to advance SOTA by at most 1% in the last few years. Furthermore, MET can be $> 20\%$ more accurate than Gradient-boosted decision trees – considered as a SOTA method for the tabular setting – on multiple benchmarks.

1 Introduction

Recently, self-supervised pre-training (SSL) followed by supervised fine-tuning has emerged as the state of the art approach for semi-supervised learning in domains such as natural language processing (NLP) [10], computer vision [7] and speech/audio processing [3]. Given that there is an extensive amount of raw, unlabeled data in various settings such as healthcare, finance, marketing, etc., most of which exist in tabular form, extending SSL to tabular data is an important direction of research.

Broadly speaking, there are two dominant approaches to SSL: (i) reconstruction of masked inputs, and (ii) invariance to certain augmentations/transformations, also known as *contrastive learning*. Several prior works [33, 29] have adopted the second approach of contrastive learning for designing SSL methods for tabular data (tabular-SSL). The underlying structure and semantics of specific domains such as images remain somewhat static, irrespective of the dataset. So, one can design generalizable domain specific augmentations like cropping, rotating, resizing etc. However, tabular data does not have such fixed input vocabulary space (such as pixels in images) and semantic structure, and thus lacks generalizable augmentations across different datasets. Consequently, there are only a limited

Table Representation Learning workshop at NeurIPS 2022

number of augmentations that have been proposed for the tabular setting such as mix-up, adding random (gaussian) noise and selecting subsets of features [33, 29].

In this paper, we begin with the following hypothesis: for any tabular dataset, (i) there is a latent (i.e., unknown/unobserved) graphical model that captures the relations between different coordinates/features, and (ii) classification is easier in the latent space. For example, in the CovType dataset, where the task is to predict the type of forest (e.g., deciduous, alpine etc.) given features such as elevation, soil type and so on, extensive research in mountain and forest science has established that there are very specific relations among different features [5, 2], and leveraging and learning these relations could yield significant improvements in classification accuracy of machine learning models.

Based on this hypothesis, we propose a masking based reconstruction approach for self supervised learning for tabular datasets. More concretely, for every unlabeled data point, we randomly choose a fraction of the coordinates, mask their values, and then train a model to predict values of these masked coordinates using the remaining unmasked coordinates. We use a transformer architecture with learnable (positional) embeddings for each coordinate, which capture the relations between different coordinates. While masked reconstruction task with a transformer architecture has been successfully used for SSL in computer vision [12] and natural language processing [10], to the best of our knowledge, this is the first work to successfully apply this paradigm to tabular datasets.

In particular, we demonstrate through experiments on a simple toy tabular setting, how the position embeddings in a transformer, learned with the masked reconstruction task, can capture the dependency structure across features. Further, on a real world dataset of forest cover type classification (CovType), we indeed show that the most correlated positional embeddings correspond to the features which indeed have meaningful relation between them as corroborated by extensive works in the forest science.

We evaluate the performance of MET through extensive experiments on several tabular datasets spanning a wide range in number of examples, number of classes and difficulty. Our experiments show that MET outperforms current SOTA tabular-SSL methods like DACL [33], SubTab [29], VIME [36], as well as SOTA tabular supervised algorithms such as gradient boosted decision trees (GBDT) [20] on *all* of these datasets, with accuracy improvements up to 5% on some of these datasets. For example, on Criteo – a popular large scale dataset for click through rate prediction with 45 million examples – our algorithm achieves 5% improvement in AUROC over the current SOTA [35]. To put this in context, the SOTA on Criteo has improved by less than 2% over the last six years [18]. Furthermore, on some datasets, MET trained with about 20% of the labelled train-set is as effective as standard supervised learning methods trained with all the labeled points in the train-set.

To summarize, in this paper, we propose MET, which is a masking based reconstruction task with a transformer architecture, as an effective approach for tabular-SSL. Conceptually, through experiments on a toy setting, we show that this approach can learn the relations between different coordinates in the dataset, which helps in downstream classification. Practically, we show through extensive experiments on several popular tabular datasets that MET significantly outperforms all the current SOTA tabular-SSL baselines as well as SOTA supervised approaches.

2 Related Work

Self-Supervised Learning : Self supervised learning (SSL) has shown promising results not only in the regimes where the labelled training data is scarce but has also shown great empirical success in training large scale models across various domains like Natural Language Processing and Computer Vision. SSL can be broadly classified into two categories : Pretext task based approaches and contrastive learning based approaches. Pretext based SSL approaches solve a "pretext" task like reconstruction from a masked or a noisy input, in order to learn the underlying distribution of the unlabelled data. Wav2Vec[3] efficiently trains a large scale speech-to-text model using masking based reconstruction for learning good speech representations. Similarly, [15, 14] have used masking for learning speech representations. Masked language modelling has been extensively studied in literature [10, 24], and has shown quite promising results. Motivated by the success of masking based approaches in NLP and speech, a recent paper [12] proposed a masked input reconstruction approach for visual representation learning. Prior to this, [19] also proposed masking although in feature space. In this paper, motivated by the constraints in tabular setting along with the success of masking based approaches, we build a purely reconstruction based approach for tabular-SSL.

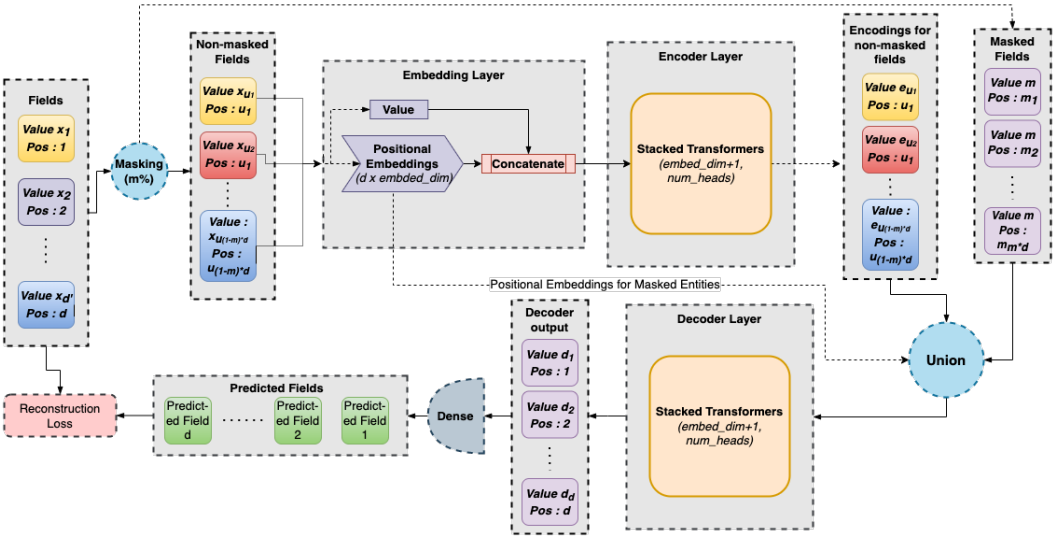

Figure 1: MET Framework for tabular-SSL. Given an input, we mask out a fraction of co-ordinates (features). The masked input is then concatenated with its learnable positional encodings and fed to the transformer based encoder as input. The obtained encoder output (learnt representations) are then passed through the decoder along with the mask token. Recontruction loss is then optimized end-to-end.

A concurrent line of work in SSL learns representations using instance level separation tasks as discussed in [7, 8, 37, 13]. Some other pretext tasks like solving jigsaw puzzles have also been proposed in [21]. However, all these approaches require domain-specific knowledge to create positive-negative sample pairs. Some recent advances towards a domain agnostic approach have also been proposed like [33]. We compare our proposed algorithm MET against such approaches in Section 5.

Adversarial Self supervised learning While adversarial SSL has been explored in the context of contrastive learning [16], it seems to be less explored for reconstruction based SSL. Our method MET proposes a novel framework where we try to find adversarial points in the input manifold which have a high reconstruction loss. [6] have also proposed adversarial learning, although to learn robust pre-trained models. MET instead explores the use of adversarial search over input manifold to learn better separable representations for higher accuracy on downstream classification. [27] proposes to find an adversarial mask which maximizes the distance between the representations of input and its masked (adversarial) counterpart.

Self Supervised Learning for Tabular Data Reconstruction based SSL has been previously explored in SubTab [29], which treats it as a multi-view representation learning problem. They try to learn representations for multiple croppings of the input data and at inference time aggregate the representations of the croppings (multiple-views). Note that MET performs random masking over the input space only at the training time, at inference the representation is given by passing all the co-ordinates through the encoder and hence does not consider the problem as a multi-view representation learning. [36] uses a combination of predicting the masked tokens and reconstruction. Both [36, 29] use gaussian noise addition to the input to prevent the auto-encoder from learning an identity mapping. MET efficiently searches for noise using adversarial search to learn better representations.

Learning Graphical Models : There is a large body of work on unsupervised learning of probabilistic relations between different coordinates/features, as expressed by a graphical model [9, 1, 4, 22, 17]. A popular approach in these works is to predict the value of any single coordinate using all the remaining coordinates. This can however be computationally challenging since this procedure needs to be repeated for every coordinate. The masking based reconstruction task with a random subset of coordinates masked in every step can be seen as a computationally efficient way of doing the same thing. To the best of our knowledge, this is the first work to connect masking based reconstruction approaches for SSL with the work on learning graphical models.

3 Preliminaries

In this section, we formalize the general task of self-supervised representation learning and introduce all the notations required for laying out the proposed approach MET formally.

Notation: We use $x_i \in \mathbb{R}^d$ to denote an example and x_i^j to denote the j^{th} coordinate (or feature) of x_i. Every coordinate in x_i i.e. $x_i^j \in \mathbb{R}$ can be either a categorical or a non-categorical value, without being explicitly specified. Let S_i denote the set of masked co-ordinates for x_i, and let $x_i^{S_i}$ denote the masked input, i.e. $x_i^{S_i}$ consists of only those x_i^j such that $j \in [d] \setminus S_i$. S_i is chosen randomly for every sample in every training iteration, and the number of masked coordinates i.e. $|S_i|$ is dictated by the masking ratio hyperparameter.

Self-Supervised Representation Learning: Consider access to a corpus of unlabelled dataset given by $\mathcal{D}_u = \{x_i\}_{i=1}^{N_u}$ where each datapoint $x_i \in \mathbb{R}^d$. The general goal of self-supervised learning is to learn a parameterized mapping $f_\theta : \mathbb{R}^d \to \mathbb{R}^m$ between the input x_i and its representation $f_\theta(x_i) \in \mathbb{R}^m$, such that the representations are well suited for a downstream task as described next.

Evaluation of learned representations: In this paper, we evaluate the quality of learned representations through accuracy on a downstream classification task. More concretely, we have access to a labelled training dataset $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N_{\text{train}}}$ where $y_i \in \mathbb{R}^k$ and each (x_i, y_i) is drawn independently and identically (i.i.d.) from some underlying distribution $\mathcal{D}$ on ${}^d\times^k$. The task is to learn a classifier $c_\phi : \mathbb{R}^d \to \mathbb{R}^k$ which minimizes $_{(x,y)\sim\mathcal{D}}\left[\ell(c_\phi(x), y)\right]$, where ℓ is a loss function such as $0-1$ loss or cross entropy loss etc. Given the learned representations $f_\theta \in^m$, we train a shallow classifier $g_\mu : \mathbb{R}^m \to \mathbb{R}^k$ (we use a 2-hidden layer MLP in our default setting) and use the resulting accuracy to evaluate the quality of learned representations f_θ.

4 Method

As described in the previous section, our goal is to learn a parameterized mapping $f_\theta : \mathbb{R}^d \to \mathbb{R}^m$ between the input x_i and its representation $f_\theta(x_i) \in \mathbb{R}^m$. Motivated by the intuition that there exists a latent structure over different coordinates and that classification is easier in this space, we learn the latent structure through masked reconstruction. More concretely, we have an encoder represented by $f_\theta : \mathbb{R}^d \to \mathbb{R}^m$ and a decoder represented by $h_\phi : \mathbb{R}^m \to \mathbb{R}^d$. The task of the encoder is to take a noisy version of input example x_i, e.g., where some coordinates of x_i are masked, and reconstruct the entire example x_i. More formally, as introduced in the previous section, let S_i denote the set of masked coordinates for x_i, and $x_i^{S_i}$ denote the masked input, then this approach can be written as minimization of the following function:

$$\mathcal{L}_{\text{rec}}(\theta, \phi) = \sum_{i=1}^{N_u} \|x_i - h_\phi(f_\theta(x_i^{S_i}))\|_2^2. \tag{1}$$

For downstream evaluation task, we discard the decoder h_ϕ and only use the representations computed by f_θ. At an empirical level, the high level approach of masked reconstruction (also known as denoising autoencoder) (1) was first proposed in the seminal paper [34], and subsequently instantiated for various domains such as text, speech and images, which have required several domain specific insights including architectures of h_ϕ and f_θ, which coordinates to mask etc. ([10, 30, 12]). In parallel, there was a long line of theoretical work on learning graphical models from data starting with [9]. The state of the art approaches for this again use a similar masked reconstruction task: recover the value of a particular coordinate, given the values of all the remaining coordinates. Interpreting the masked reconstruction task (1) through the lens of learning graphical models, we observe that (1) has the ability to learn the underlying latent structure among different coordinates, thereby making the classification task easier.

Encoder-Decoder Architecture and the Framework: Given that transformers [31] explicitly try to capture the relation between different coordinates through positional embeddings and attention mechanism, they are a natural choice for encoder f_θ as well as decoder h_ϕ. Below we describe in detail the whole framework i.e. input to the transformer encoder, the input to the transformer decoder and how we obtain the reconstructed input.

Algorithm 1: MET : Masked Encoding Tabular data

Input : Unlabelled data $\mathcal{D}_u = \{x_i\}_{i=1}^{N_u}$, masking ratio $m \in [0, 1]$, Encoder f_θ, Decoder h_ϕ, projection radius ϵ, weight of adversarial loss λ

for $iteration = 0, 1, ...\ N-1$ **do**

  for $x_i \in \mathcal{D}_u$ **do**

    $S_i \subset [d]$ s.t. $|S_i| = m * d$ /* Random subset of coordinates to mask */

    $\hat{x}_i = h_\phi(f_\theta(x_i^{S_i}))$ /* Try to reconstruct from masked input. */

    $\mathcal{L}_{\text{rec}}^{\text{std}} = \|x_i - \hat{x}_i\|_2^2$ /* Standard reconstruction loss. */

    $h \sim \mathcal{N}(0, \mathcal{I}_d)/\sqrt{d}$ /* Initialize adversarial perturbation. */

    for *steps in* $1, 2, \ldots$ *adv_steps* **do**

      /* Find adversarial perturbation h to maximize reconstruction loss using gradient ascent. */

      $\hat{x}_i = h_\phi(f_\theta((x_i + h)^{S_i}))$

      $\mathcal{L}_{\text{rec}}(h) = \|x_i - \hat{x}_i\|_2^2$

      $h = h + \eta \frac{\nabla_h \mathcal{L}_{\text{rec}}}{\|\nabla_h \mathcal{L}_{\text{rec}}\|}$

      $h = \frac{h}{\|h\|}\alpha$ where $\alpha = \|h\| \cdot \mathbb{1}[\|h\| < \epsilon] + \epsilon \cdot \mathbb{1}[\|h\| \geq \epsilon]$

    $\hat{x}_i = h_\phi(f_\theta((x_i + h)^{S_i}))$

    $\mathcal{L}_{\text{rec}}^{\text{adv}} = \|x_i - \hat{x}_i\|_2^2$ /* Adversarial reconstruction loss. */

    $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}}^{\text{std}} + \lambda \cdot \mathcal{L}_{\text{rec}}^{\text{adv}}$ /* Final loss is a sum of standard and adversarial reconstruction losses. */

    $(\theta, \phi) = (\theta, \phi) - \eta(\nabla_\theta \mathcal{L}_{\text{total}}, \nabla_\phi \mathcal{L}_{\text{total}})$ /* Gradient descent on θ and ϕ. */

- Given $x_i \in \mathbb{R}^d$ and the set of masked coordinates S_i, let $z_i^j \in \mathbb{R}^{1+e}$ denotes the input embedding corresponding to the j^{th} coordinate of x_i, where $j \in [d] \setminus S_i$. z_i^j is constructed by concatenation of $pe_j \in^e$, a learnable positional encoding corresponding to the j^{th} coordinate and $x_i^j \in \mathbb{R}$, the value of j^{th} coordinate in x_i.
- Given $x_i \in \mathbb{R}^d$ and the set of masked coordinates S_i, then $z_i \in \mathbb{R}^{|[d]\setminus S_i| \times (1+e)}$ denotes the input corresponding to the masked input $x_i^{S_i} \in \mathbb{R}^{|[d]\setminus S_i|}$ to the transformer encoder (f_θ).
- The output of the transformer encoder (also the learnt representations) is given by $w_i \in^{|[d]\setminus S_i| \times (1+e)}$, such that $w_i = f_\theta(z_i)$.
- Let $v_i \in^{S_i \times (1+e)}$ denote the representation for the masked coordinates, constructed by concatenation of pe_j with a learnable mask parameter $u \in , \forall j \in S_i$.
- The reconstructed input $\hat{x}_i \in^d$ is then given by $\hat{x}_i = h_\phi([w_i, v_i])$, where h_ϕ denotes the transformer decoder.

Adversarial loss: In the context of supervised learning, several papers have demonstrated that adversarial training can yield more robust features [28] that are better for transfer learning [26]. While an adversarial loss function has been observed to encourage learning of robust features in contrastive SSL [6, 16], to the best of our knowledge, it does not seem to have been explored in the context of masked autoencoders. In this work, we demonstrate that an adversarial version of the reconstruction loss works better than standard reconstruction loss. More concretely, the adversarial reconstruction loss is given by:

$$\mathcal{L}_{\text{rec}}^{\text{adv}}(\theta, \phi) = \sum_{i=1}^{N_u} \max_{\delta : \|\delta\| \leq \epsilon} \|x_i - h_\phi(f_\theta(x_i^{S_i} + \delta))\|_2^2. \tag{2}$$

In this paper, we constrain the adversarial noise δ in an ϵ radius $L2$ norm ball around the input data point x_i, where ϵ is chosen from a grid-search in $\{2, 4, 6, 10, 12, 14\}$.

Finally, MET minimizes the sum of loss functions given in (1) and (2) :

$$\mathcal{L}_{\text{rec}}^{\text{MET}} = \mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{rec}}^{\text{adv}}. \tag{3}$$

We fix $\lambda = 1$ in all our experiments. The overall algorithm for MET, which minimizes (3) is given in Algorithm 1. For consistency of notation, we present a non-batch version of the algorithm.

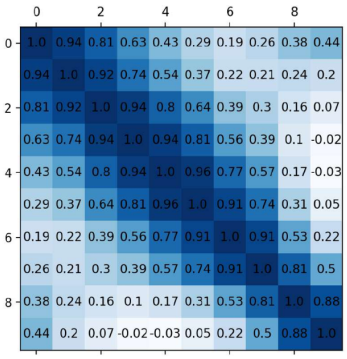

Figure 2: Cosine similarity between the learnt position embeddings

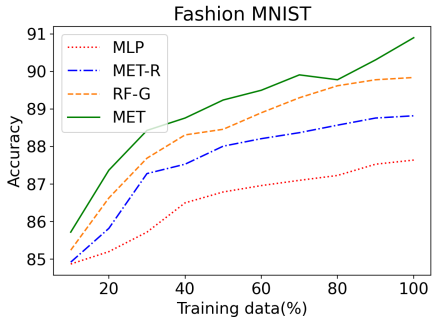

Figure 3: Downstream accuracy as a fraction of labeled data.

4.1 Analysis on Toy Dataset

As explained previously in Section 1, our proposed approach MET relies on capturing and learning the latent graphical model that defines the relation between various features (coordinates) of the tabular dataset. We first show that MET is indeed able to learn this latent graphical model by presenting some interesting results on a 10-dimensional toy dataset.

Let each datapoint $x_i \in \mathbb{R}^{10}$ be sampled from a linear graphical model as follows (recall that $x_i^j \in \mathbb{R}$ denotes the j^{th} coordinate of x_i) :

$$x_i^0 \sim \mathcal{N}(0, 1) \tag{4}$$

$$x_i^j = x_i^{j-1} + n_j, \forall j \in [1, 10]; \text{ where } n_j \sim \mathcal{N}(0, 1) \tag{5}$$

Further, for the downstream binary classification task, let the label $y_i = \mathbb{1}[|n_1| \geq \Phi^{-1}(0.5)]$ where Φ^{-1} denotes the inverse gaussian CDF.

First, on the downstream task, our proposed approach MET (99.87 AUROC) outpeforms all the baselines, most competitive being Gradient Boosted Decision Trees (97.86 AUROC) and Gaussian featurization (see Section 5.1) which gets 98.07 AUROC. But more importantly, MET is able to identify the relation between the consecutive coordinates, as demonstrated by the high cosine similarity scores for the consecutive coordinates in Figure 2. Note that MET learns the reconstruction task and hence it is able to learn the whole underlying graphical model. Hence, it can potentially perform various downstream tasks with high accuracy, although only the first two coordinates might have been relevant for the binary classification in this particular case. More detailed results for other baselines along with AUROC and accuracy numbers can be found in the Appendix.

Further, in the next section, we show that similar trends even hold on real world tabular datasets, where MET is able to identify the set of features with meaningful relation.

5 Experiments

Here, we empirically evaluate our proposed approach MET against tabular-SSL methods, along with other classical baselines as described in subsection 5.1. We experiment with common tabular dataset benchmarks like the permuted MNIST, permuted FashionMNIST and permuted CIFAR-10. Further, we work with two other common tabular datasets from the UCI machine learning repository [11] : Forest CovType and Adult Income which are described in Section 5.3. TODO saching once results are finalized

5.1 Baselines and Existing Methods

We compare MET with the following baselines:

- VIME [36] : A SSL approach for tabular dataset, which uses a combination of masked token prediction and reconstruction loss. Note that MET proposes masked input reconstruction.

- SubTab [29] : SubTab views SSL as a multi-view representation learning problem, where representations from multiple croppings are aggregated at test time.
- DACL [33] : A domain agnostic contrastive learning baseline which uses mixup as an augmentation. We specifically use DACL+ which uses geometric mean based mixup.
- MLP : We also compare against this natural baseline, wherein we train a MLP over the raw tabular data (and not the learnt representations) with the available labeled samples.
- Random Forest (RF): We train a random forest with 100 decision trees over the raw tabular data using the available labeled samples. We choose maximum depth and minimum samples in a leaf node by performing a grid-search over $\{2, 5, 10, 20, auto\}$ and $\{1, 2, 5\}$ respectively.
- Gradient Boosted Decision Trees (GBDT): We train a gradient boosted decision tree over raw tabular data, choosing the parameters by grid search over maximum depth in $\{2, 5, 10, 20, auto\}$, minimum samples in leaf node over $\{1, 2, 5\}$ and learning rate in $\{0.1, 0.01, 0.001, 0.0001\}$.
- Random Featurization (MET-R): To check the effectiveness of the learnt representations, we compare against fine-tuning an MLP over the representations from a random encoder i.e. a randomly initialized and fixed transformer. This is denoted by MET-R.
- Random Gaussian Featurization (RF-G) : Here, we compute standard random kitchen sink [25] style features, i.e., $\phi(x) = Rx$ is the embedding of point $x \in \mathbb{R}^d$. Random features are known to be asymptotically an accurate approximation of the RBF kernel, which in turn is known to be a highly accurate and in fact, a "universal" classifier for tabular data. Note that we fix embedding dimension of RF-G to be same as that of MET.

5.2 Implementation Details

We use transformers [32] as the backbone for both the encoder and the decoder. Embedding dimension for the encoder and the decoder is chosen from a gridsearch in $\{64, 100, 128\}$, feedforward dimension from $\{64, 100, 128\}$, encoder and decoder depth from $\{1, 3, 6\}$ and the number of heads from $\{1, 2, 3\}$. The weight for adversarial reconstruction loss λ in MET (see Algorithm 1) is set to 1. We share the exact hyper-parameters for all the experiments in the appendix. All the experiments have been performed on a cluster of Tesla P100 GPUs.

5.3 Datasets

- MNIST: Permuted, Normalized and Flattened version of standard MNIST dataset.
- FMNIST: Permuted, Normalized and Flattened version of standard Fashion-MNIST dataset.
- CIFAR-10: Permuted, Normalized and Flattened version of standard CIFAR-10 dataset.
- CoverType: Forest CoverType(*CoverType*) is a UCI dataset where the task is to predict seven different types of forest cover type from cartographic variables of a 30x30 meter cell.
- Income: Adult Income(*Income*) is a UCI dataset where the binary prediction task is to determine whether a person makes over $50K a year based on census data.
- Obesity: Obesity is a relatively small dataset of 253 samples with 465 features representing the human gut metagen-omic samples of the obesity cohort available publicly.
- Criteo: Criteo consists of 45M samples, each with 39 features pertaining to display ads collected for a week at CriteoLabs for click-through-rate(CTR) prediction.

A more detailed explanation of the datasets is available under appendixA.2

5.4 Downstream Classification

In this section, we compare MET against various baselines as mentioned in subsection 5.1. We compare the downstream classification accuracy of various representations. Specifically, we train an MLP over the learnt representations using all the available labeled dataset samples; see Section 5.2.

Multi-Class Classification Benchmarks : Table 1 compares accuracy of MET with downstream classification against tabular-SSL methods and supervised learning baselines on multi-class classification benchmarks. Both MET (adversarial noise + masking) and MET-S (only masking) outperform

Table 1: Downstream classification accuracy on four common multi-class tabular datasets, comparing MET against various baselines. Table 2 shows additional results on new datasets for binary classification setting. MET uses adversarial training + masking for reconstruction based self-supervised-learning whereas MET-S is an ablation where only masking is used. MET outperforms the baselines across the all the datasets.

Type	Methods	FMNIST	CIFAR10	MNIST	CovType
Supervised Baseline	MLP	87.62%	16.50%	96.95%	65.47%
	RF	88.43%	42.73%	97.62%	71.37%
	GBDT	88.71%	43.43%	**100**%	72.96%
	RF-G	89.84%	29.32%	97.65%	71.57%
	MET-R	88.84%	28.94%	97.44%	69.68%
Self-Supervised Methods	VIME	80.36%	34.00%	95.77%	62.80%
	DACL+	81.40%	39.70%	91.40%	64.23%
	SubTab	87.59%	39.34%	98.31%	42.36%
Our Method	MET-S	90.94%	**48.00%**	99.01%	74.11%
	MET	**91.36%**	47.82%	99.19%	**76.71%**

Table 2: Downstream classification accuracy and auroc scores on three common tabular datasets with binary classification task, comparing MET against various baselines. MET outperforms all the baselines even on the AUROC metric.

Datasets	Metric	MLP	RF	GBDT	RF-G	MET-R	VIME	SubTab	MET
Obesity	Accuracy	62.4	65.99	64.4	58.79	51.875	59.23	67.48	**76.88**
	AUROC	52.3	64.36	64.4	54.45	53.2	57.27	64.92	**71.84**
Income	Accuracy	84.36	85.88	86.01	85.59	75.51	86	84.43	**86.25**
	AUROC	89.39	91.53	92.5	90.09	83.48	89.01	88.95	**93.85**
Criteo	Accuracy	74.28	74.11	57.97	74.62	76.21	74.2	73.02	**78.49**
	AUROC	79.82	77.57	78.77	80.32	79.17	74.28	76.57	**86.17**

the baselines across all the datasets. For example, on the permuted Fashion MNIST dataset, MET achieves an accuracy of 91.36%, outperforming all the other tabular SSL baselines like SubTab (87.59%). Similarly, on CovType, MET is about 10% more accurate DACL+, and in fact about 34% more accurate than SubTab, perhaps due to lack of semantics in neighbouring columns in the dataset. Overall, we observe that MET gives an average improvement of 3.2% accuracy compared to the nearest competitive baseline, establishing MET as a new state-of-the-art approach for self supervised learning on tabular data.

Here, we would like to make two key points.
a) Note that using the same embedding dimensions, MET is able to give up to 18% more accurate classifier than RF-G embeddings. This is interesting because RF-G embeddings are also able to capture non-linear features in the data; in fact, it can approximate RBF kernel itself. But, due to self-supervised training with the entire unlabeled data, MET can capture the data manifold more accurately, while RF-G are completely *independent* of the data distribution. This indicates the importance of further investigation of data-distribution based (random) embedding methods.
b) Here, we use a modified and much harder version of CovType, where the key categorical features like soil type are represented by their category index, instead of one-hot vectors. This immediately imposes an ordering on categories which is incorrect, and hence the representation learning method has to somehow learn to embed such coordinates in something similar to one-hot vectors, which in absence of additional domain information is challenging. Naturally, methods like MLP struggle on this dataset, but MET is able to get a reasonable accuracy which is 11% higher than MLP.

Binary Classification Benchmarks : Table 2 presents results on three real-world binary classification tabular datasets. Consider Criteo, which is a large scale (45 million training sample) click prediction dataset. MET seems to scale seamlessly to these large scale datasets, outperforming all the baselines, giving an AUROC of 86.37 compared to MET-R (83.25 AUROC) and GBDT which has just 74.47 AUROC. Similar trends hold on Income and Obesity datasets.

Accuracy with a fraction of labeled training data: Next we compare our proposed algorithm MET against the baselines when only a fraction of labeled data is used for the supervised training.

Specifically, we vary the fraction of labeled data used for training the downstream classifier from 20% to 100% and compare the obtained downstream classification accuracies with the baselines. Figure 3 shows the variation of accuracy with the fraction of labeled data for MET, comparing against the baselines like gaussian random featurization (RF-G), learning MLP directly over raw features and a random encoder (MET-R). We observe that MET outperforms the baselines for all the choices of fraction of labeled data used for supervised learning.

Concatenated vs averaged embeddings: Next, we try two approaches for the task of getting good tabular representations from co-ordinate level representations:

- **Concatenation** : Concatenating co-ordinate level representations learnt by MET to represent tabular level representations.
- **Averaging** : Taking an average of representations learnt by MET over all co-ordinates to represent tabular level representations.

We try above mentioned approaches for FMNIST and CovType. For both datasets, concatenation significantly outperforms averaging. For CovType, concatenation and averaging obtain 74.11% and 61.87% accuracies respectively and for FMNIST, they obtain 90.94% and 88.64% respectively.

Effect of adversarial reconstruction: Next, we analyze the effect of using gaussian noise along with masking for reconstruction based SSL on tabular datasets. In Table 1, observe that MET always outperforms or matches MET-S (without adversarial noise), and in some cases like CovType gives upto 2.5% improvement compared to the non-adversarial counterpart.

6 Conclusion and Limitations

In this paper, we proposed a *purely reconstruction based* SSL algorithm, MET, for representation learning on tabular datasets. The two key ideas in MET are (i) use a concatenation of representations for all features instead of averaging, and (ii) use adversarial reconstruction loss in addition to the standard loss. Through experiments on five tabular datasets, we showed that MET achieves a new SOTA result for downstream classification on these datasets, improving over previous contrastive based approaches by 3.2% on average.

While reconstruction based SSL has been shown to learn powerful representations across various domains such as text [10], vision [12] as well as tabular (this paper), a thorough understanding of *why* reconstruction loss promotes linearly separable representations is missing in the literature. In this paper, we took a step forward by proposing a simple dataset, that can act as a test bed for answering this question. We showed empirically that while this dataset is not linearly separable in the input space, it became linearly separable using MET representations. We believe that a mathematical proof of this phenomenon could shed light on *why* and *how* reconstruction based approaches learn useful representations. We also demonstrated that using a concatenation of representations of all features/coordinates gives substantially better results than pooling of all token level representations as done in vision [12] and text [10]. This is intuitive as the average representation of all tokens is not specifically trained to be useful for the downstream task. At the same time, concatenation significantly increases the representation dimension as well as the complexity of the downstream finetuning model. Hence, it can exacerbate the risk of overfitting. For tabular-SSL, our results ruled out this case. However, a thorough investigation of this aspect is also an interesting direction for future work.

References

[1] Francis Bach and Michael Jordan. Learning graphical models with mercer kernels. *Advances in Neural Information Processing Systems*, 15, 2002.

[2] David Badía, Alberto Ruiz, Antonio Girona, Clara Martí, José Casanova, Paloma Ibarra, and Raquel Zufiaurre. The influence of elevation on soil properties and forest litter in the siliceous moncayo massif, sw europe. *Journal of Mountain Science*, 13(12):2155–2169, 2016.

[3] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *CoRR*, abs/2006.11477, 2020.

[4] Guy Bresler, Elchanan Mossel, and Allan Sly. Reconstruction of markov random fields from samples: Some observations and algorithms. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 343–356. Springer, 2008.

[5] Filipe X Catry, Francisco C Rego, Fernando L Bação, and Francisco Moreira. Modeling and mapping wildfire ignition risk in portugal. *International Journal of Wildland Fire*, 18(8):921–931, 2009.

[6] Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 699–708, 2020.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020.

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020.

[9] CKCN Chow and Cong Liu. Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory*, 14(3):462–467, 1968.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[11] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. *CoRR*, abs/2111.06377, 2021.

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2019.

[14] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units, 2021.

[15] Dongwei Jiang, Xiaoning Lei, Wubo Li, Ne Luo, Yuxuan Hu, Wei Zou, and Xiangang Li. Improving transformer-based speech recognition using unsupervised pre-training, 2019.

[16] Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:2983–2994, 2020.

[17] Adam Klivans and Raghu Meka. Learning graphical models using multiplicative weights. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 343–354. IEEE, 2017.

[18] PaperswithCode Criteo Leaderboard. Leader board for Criteo. `https://paperswithcode.com/sota/click-through-rate-prediction-on-criteo`, 2022. [Online; accessed 22-September-2022].

[19] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, and Jinqiao Wang. Mst: Masked self-supervised transformer for visual representation, 2021.

[20] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Boosting algorithms as gradient descent. *Advances in neural information processing systems*, 12, 1999.

[21] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations, 2019.

[22] Praneeth Netrapalli, Siddhartha Banerjee, Sujay Sanghavi, and Sanjay Shakkottai. Greedy learning of markov network structure. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1295–1302. IEEE, 2010.

[23] Min Oh and Liqing Zhang. Deepmicro: deep representation learning for disease prediction based on microbiome data. *Scientific Reports*, 10, 04 2020.

[24] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[25] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Neurips*, 2008.

[26] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020.

[27] Yuge Shi, N. Siddharth, Philip H. S. Torr, and Adam R. Kosiorek. Adversarial masking for self-supervised learning, 2022.

[28] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[29] 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.

[30] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pages arXiv–1807, 2018.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[33] Vikas Verma, Minh-Thang Luong, Kenji Kawaguchi, Hieu Pham, and Quoc V. Le. Towards domain-agnostic contrastive learning. *CoRR*, abs/2011.04419, 2020.

[34] 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*, pages 1096–1103, 2008.

[35] Zhiqiang Wang, Qingyun She, and Junlin Zhang. Masknet: Introducing feature-wise multiplication to ctr ranking models by instance-guided mask. *ArXiv*, abs/2102.07619, 2021.

[36] 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.

[37] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction, 2021.

A Appendix

A.1 Hyper-Parameters

In this section, we share the exact hyper-parameters of MET for replicating the results on all the five tabular datasets. Note that Encoder Depth refers to the number of transformer layers in the encoder stack and Decoder Depth refers to the number of transformer layers in the decoder stack. Adversarial Learning Rate (lr_adv) refers to the learning rate used for gradient ascent in adversarial loop and Learning Rate(lr) refers to the learning rate for gradient descent on reconstruction loss. We perform a grid-search for Embedding dimension(e) and Feed-forward dimension(fw) in $\{64, 100, 128\}$, Number of Heads in the transformer architecture in $\{1, 2, 3\}$, encoder and decoder depth in $\{1, 3, 6\}$, Learning Rate(lr) in $\{1e^{-1}, 1e^{-2}, 1e^{-3}, 1e^{-4}, 1e^{-5}\}$, masking percentage($m$) in $\{30, 50, 70, 80, 90\}$, adversarial steps(adv_steps) in $1, 2, 4$, radius of L2-norm ball(ϵ) in $\{2, 6, 10, 12, 14\}$ and learning rate of gradient ascent step(lr_adv) in $\{0.1, 0.01\}$. The optimal set of hyper-parameters for various datasets obtained are mentioned in Tables3 and 4. Note that all the ablation studies are conducted using these parameters and MET-S.

Table 3: We share the exact hyper-parameters for replicating the results with MET.

	Embedding Dimension(e)	Feed-forward Dimension(fw)	Number of Heads	Encoder Depth	Decoder Depth
Fashion MNIST	64	64	1	6	1
CIFAR10	100	64	2	3	3
MNIST	64	64	1	6	1
CovType	100	64	1	1	1
Adult Income	64	64	1	3	6

Table 4: We share the exact hyper-parameters for replicating the results with MET.

	lr	Masking Percentage(m)	Adversarial Steps(adv_steps)	L2 Norm Ball Radius(ϵ)	Adversarial lr (lr_adv)
Fashion MNIST	$1e^{-5}$	70	2	2	$1e^{-2}$
CIFAR10	$1e^{-4}$	70	3	14	$1e^{-2}$
MNIST	$1e^{-4}$	70	2	12	$1e^{-2}$
CovType	$1e^{-4}$	50	5	4	$1e^{-1}$
Adult Income	$1e^{-3}$	80	1	6	$1e^{-1}$

A.2 Datasets

MNIST[1]: The MNIST dataset of handwritten digits consists of 28x28 dimensional images, which are then flattened to get 784 coordinates in tabular form. The classification task consists of ten classes, one for each digit. A split of 60,000 entries as the train set and 10,000 entries as the test set is used as per the split for the original dataset.
FMNIST: Fashion-MNIST(*FMNIST*) is a dataset of Zalando's article images consisting of 28x28 dimensional images and is proposed as a more challenging replacement dataset for the MNIST dataset. The data is flattened to get 784 coordinates in tabular form and has ten classes. A split of 60,000 entries as the train set and 10,000 entries as the test set is used as per the split for the original dataset.
CIFAR-10: The CIFAR-10 dataset contains 60,000 color images each of size 32x32 belonging to ten different classes with 6,000 images of each class. We flatten it to get 3072 coordinates in tabular form and use a split of 50,000 entries as the train set and 10,000 entries as the test set is used as per the split for the original dataset.
CoverType: Forest CoverType(*CoverType*) is a UCI dataset where the task is to predict forest cover type only from cartographic variables of a 30x30 meter cell, as determined from US Forest Service Region 2's resource information system. The data is not scaled and contains binary columns of data for qualitative independent variables: wilderness areas and soil types. It is a 7 class classification

[1]The data is normalized and shuffled for all datasets.

problem and consists of 54 features out of which one-hot vectors of wilderness area and soil type make up for 44 features. We replace them as two features by taking an argmax over the one-hot vectors and it reduces to 12 features in a tabular form. This makes the problem harder since the categorical features are now represented as integers instead of one-hot representations[2]. A split of 11,340 entries as train set and 565,892 entries as test set is used as per the split for the original dataset.
Income: Adult Income(*Income*) is a UCI dataset where the prediction task is to determine whether a person makes over $50K a year based on census data. It consists of a mix of six continuous and eight categorical fields. Similar to CoverType dataset, we use integers instead of one hot representation for the categorical features and get 14 features in a tabular form . A split of 30,162 entries as train set and 15,060 as test set is used as per the split for the original dataset.
Obesity: Obesity[23] is made of human gut metagen-omic samples of obesity cohort available publicly. Each sample has 465 features representing them and we use a binary classification task of predicting whether the sample is non-obese or obese. All 465 features are continious. The features are normalized and shuffled before feeding it to the model. It is a relatively small dataset of 253 samples, hence we use a 90-10 train-test split with a 10-fold cross validation for evaluation purposes.
Criteo: Display advertising is a billion dollar industry and an important use case of machine learning. Criteo consists of one-week data from CriteoLabs for click-through-rate(*CTR*) prediction summing up to 45M samples with 39 features each. Out of the 39 features, 26 are categorical, some of which have as many as 5M distinct values in the form of anonymized string and the other 13 are real-valued fields.

[2]Consequently, our accuracy numbers are not directly comparable to standard results on this dataset.