Thank you for the insightful comments from all reviewers. Those are very helpful to improve our submission.

Additional baselines [R1-A1] We agree that [TaBERT & TabNet] and VIME have common concepts - the pretext task for self-supervised learning (self-SL) is recovering masked data. However, there are two major differences as well: (1) VIME utilizes another pretext task (binary mask vector estimation) for self-SL, (2) VIME utilizes the imputed data as augmented samples for semi-SL. We include TaBERT and TabNet as additional baselines and the brief results (for Table 2) are: [0.8653, 0.9463, 0.8127] and [0.8637, 0.9487, 0.8114] which are consistently worse than VIME.
Extra ablation study [R1-A2] These results are already presented in Section 8 in Appendix (see Table 4). The combination of two pretext tasks is consistently better than only using one of those two pretext tasks.
Correlation structure of tabular data [R1-A3] The spatial correlations between pixels in images or the sequential correlations between words in text data are well-known and consistent across different datasets. By contrast, the correlation structure among features in tabular data is unknown and varies across different datasets. In other words, there is no *common* correlation structure in tabular data.
Novelties [R1-A4] The design of VIME is dedicated to tabular data. The pretext tasks we use here mark a departure from those used previously on image and text data. The main novelties of the VIME framework are (1) novel pretext task(s) for tabular data (the combination of two pretext tasks) in Section 4.1 and (2) novel data augmentation for tabular data in Section 4.2. We will clarify these novelties in the revised manuscript and tone down the semi-SL part.
Alternative augmentation [R1-A5] Thank you for the suggestion, though we would note that augmentation methods for tabular data are not standardized. Note also that we included "MixUp" results as an additional augmentation model in the manuscript (and it underperformed VIME in all experimental settings). We have since performed an additional experiment in which we add Gaussian noise to the original data and treat this as the augmented sample. Briefly, as compared with results in Table 2, performances with Gaussian noise augmentations are 0.8627, 0.9481, 0.8253 with Income, MNIST, and Blog datasets, respectively. These results are consistently worse than the performances of VIME.
Minor issues [R1-A6] We will fix those typos and improve Figure 1 and 2 in the revised manuscript.

Categorical variables [R2-A1] We agree that categorical variable issues in tabular data is critical. In practice and experiments, we change Eq (6) to cross-entropy loss for categorical features to properly handle them. We will clarify this. Most of the tabular datasets used in our paper include categorical variables. For instance, all the variables in genomic datasets are categorical. Also, two clinical datasets include various categorical features (see Table 2 in Appendix).
PCA on Genomics [R2-A2] We performed extra experiments using PCA + ElasticNet and PCA + Linear on genomics data (as suggested). Unfortunately, the performance of PCA + ElasticNet and of PCA + Linear are consistently and significantly worse than original ElasticNet and Linear models in terms of MSE. As explained in **R2-A1**, all the variables in genomic data are categorical features; and usually, applying PCA on categorical variables is not recommended.
Data normalization [R2-A3] Yes, we did. We first use MinMaxScaler to normalize the data between 0 and 1 (it can be also checked in the submitted codes (data-loader.py)); then, we train the self and semi-supervised models. We also tried StandardScaler for data normalization (mean=0, std=1); and the performances were similar with MinMaxScaler.
Clarification [R2-A4] In the revised paper, we will clarify the meaning of "Variants of VIME" in the captions.

Novelties [R3-A1] We acknowledge that self/semi-supervised learning is well-studied in the image and language domains. However, as shown in various results in the manuscript (e.g., Table 2) and Appendix (e.g., Table 5), the state-of-the-art self/semi-supervised learning models for image and language domains (such as SimCLR) underperform VIME in the tabular setting. This demonstrates that new self/semi-supervised learning models are necessary for tabular domain (we also highlight our novelties in **R1-A4**). Note that VIME consistently outperforms Gaussian noise based models in various settings not only shown in the manuscript (results with DAE baseline) but also proved in **R1-A5**.
Masking the data [R3-A2] The first term $(\mathbf{m} \cdot \bar{\mathbf{x}})$ consists precisely of the shuffled samples (with $\mathbf{m}$ determining which features will be shuffled for the given data sample $\mathbf{x}$. Essentially the masked samples are partially masked (as is the typical meaning of *masking*) - some of the sample consists of truly observed features, and the remainder is masked by the *shuffling* you refer to. In this case, the marginal distributions of the corrupted samples are valid but conditional distributions are invalid. Therefore, the encoder must consider the values of other components to estimate whether that component is corrupted. In other words, the encoder should learn the *joint distributions* (which is the main objective of self-supervised learning).
Masking with Gaussian [R3-A3] If we sample $\mathbf{m}$ with Gaussian distribution (instead of Bernoulli distribution), the prediction performances for Income, MNIST, and Blog datasets are 0.8427, 0.9439, 0.8127 which are consistently worse than VIME (See Table 2). Note that the performance degradation is significant with datasets including categorical variables (Income and Blog). This is because, with Gaussian noise, the encoder can easily identify which feature is corrupted if the corrupted features are categorical variables. Please see **R2-A1** and **R1-A5** for more details.
Clarification [R3-A4] To clarify the "Self-SL only" variant, it can be interpreted as $\beta = 0$. More specifically, we first train the encoder via self-supervised learning. Then, we train the predictive model with loss function (in Eq 7) with $\beta = 0$ (only utilizing the labeled data). We will clarify this in the revised manuscript.