Accurate Imputation and Efficient Data Acquisition with Transformer-based VAEs

Sarah Lewis*
Microsoft Research, Cambridge, UK

Tatiana Matejovicova*
Microsoft Research, Cambridge, UK

Angus Lamb
Microsoft Research, Cambridge, UK

Yordan Zaykov
Microsoft Research, Cambridge, UK

Yingzhen Li
Imperial College, London, UK

Miltiadis Allamanis
Microsoft Research, Cambridge, UK

Cheng Zhang
Microsoft Research, Cambridge, UK

*** contributed equally**

Abstract

Predicting missing values in tabular data, with uncertainty, is an essential task by itself as well as for downstream tasks such as personalised data acquisition. It is not clear whether state-of-the-art deep generative models for these tasks are well equipped to model the complex relationships that may exist between different features, especially when the subset of observed data are treated as a set. In this work we propose new attention-based models for estimating the joint conditional distribution of randomly missing values in mixed-type tabular data. The models improve on the state-of-the-art Partial Variational Autoencoder (Ma et al., 2018) on a range of imputation and information acquisition tasks.

1 Introduction

Real-world data is often incomplete, which indicates a crucial need for extending the applicability of machine learning models beyond complete and homogeneous datasets. Consider asking a doctor for a disease diagnosis based on a patient's medical records. The doctor attempts a diagnosis based on available information, but importantly, they also express their uncertainty regarding missing information that is relevant to the diagnosis. Introducing such desired behaviour to machine learning models requires the ability to handle incomplete data with many potential patterns of partial observations.

State-of-the-art approaches use the partial variational autoencoder (PVAE) to learn the underlying data distribution from mixed-type tabular data which contains randomly missing values (Ma et al., 2018, 2020; Mattei and Frellsen, 2018). This enables downstream tasks such as *personalised acquisition* of relevant information in order to resolve uncertainty in the model's predictions. Though successful in handling partially observed data with simple structures, the PVAE's architecture is less suited to modelling complex interactions between features. This is due to the use of a simple set-processing network Zaheer et al. (2017); Qi et al. (2017) which separately encodes each observed feature and then simply aggregates them, e.g. by taking the mean. To address this, we improve the PVAE by introducing multi-head attention (Vaswani et al., 2017) to handle the interactions between features.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

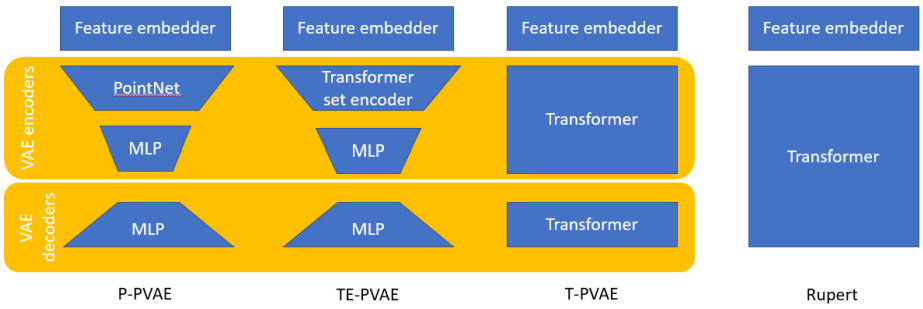

Figure 1: Basic structure of the models implemented. Computation flows from top (inputs) to bottom (outputs) of the diagrams.

This is non-trivial due to multiple modeling choices, and we research different modeling choices with experimental analysis.

In particular, in this paper, we introduce attention-based set-processing modules to the PVAE model for probabilistic missing value prediction. We introduce a batch transformer VAE architecture which generates coherent samples in a single forward pass. This provides a significant speed-up when compared with auto-regressive sampling. We also present a simple and effective transformer architecture, which we call Rupert, for mixed-type tabular data with missing values, and we present a procedure for personalised data acquisition using Rupert. We perform experiments on multiple datasets and show that our new models improve on PVAE's performance, both in terms of raw imputation accuracy, and on a downstream personalised data acquisition task.

2 Methods

Problem formulation We focus on missing value prediction. Let $\mathbf{x} = [x_1, ..., x_{|I|}]$ be a set of random variables with joint probability density $p(\mathbf{x})$. For any set $A \subset I$, we write $\mathbf{x}_A$ for the subset of variables $x_i, i \in A$. We consider problems where a subset of variables $\mathbf{x_O}$ are observed (i.e., their values are given as input to the model) while the remainder $\mathbf{x}_U, U = I \backslash O$ are unobserved. For images, $\mathbf{x}$ is an image and I is the set of pixel locations. For tabular data, $\mathbf{x}$ is a row in a table and I is the set of columns. Our models will be used to make inferences about the unobserved values $\mathbf{x_U}$ given observed values $\mathbf{x_O}$. After training, we use the models to predict randomly masked values in held-out test data and measure their accuracy. We also test our models on the downstream *personalised data acquisition* task, in which the models are used to select which variables to observe in order to reduce uncertainty about a given target variable. We perform the active learning tasks using the same method as in (Ma et al., 2018), with a small modification for the one non-VAE model, Rupert.

2.1 Models

Using pointnet based PVAE (P-PVAE) as baseline, we propose transformer encoder PVAE (TE-PVAE) and full transformer PVAE (T-PVAE) as our novel contribution with Rupert as novel reference. Each of the models that we implement takes as input some partially observed data $\mathbf{x}_O, O \subset I$, and outputs a probability distribution for each x_i. For continuous variables, the models output a mean and variance of a Gaussian distribution, and for categorical variables the output is passed through a softmax to obtain a discrete probability distribution over class labels. Figure 1 shows the architectures of the models. A detailed diagram can be found in the appendix in Figure 3.

Our T-PVAE and TE-PVAE models, described below, follow the design of *variational autoencoders* (VAE) (Kingma and Welling, 2013). They assume a latent variable model $p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$ where $p(\mathbf{z})$ is the prior of the latent variable $\mathbf{z}$ and $p(\mathbf{x}|\mathbf{z})$ is parametrized by a decoder network. An encoder network parametrizes $q(\mathbf{z}|\mathbf{x})$ which approximates $p(\mathbf{z}|\mathbf{x})$. Both networks are trained by maximizing an *evidence lower bound* (ELBO).

PointNet PVAE (P-PVAE) (Ma et al., 2018) This baseline PVAE uses PointNet (Qi et al., 2017) as the *set encoder* which encodes partial observations, each of which may contain values for different

numbers of features, to a fixed-size *set encoding* vector. PointNet transforms individual input features (points) independently, then aggregates them using a simple sum or max aggregation.

Transformer Encoder PVAE (TE-PVAE) We improve the encoder of the PVAE first. Improved architectures such as the Set Transformer (Lee et al., 2019) have shown superior performance over PointNet in expressing complicated permutation-invariant functions. The main innovation of Set Transformer is the use of multi-head attention to transform elements of the input set before pooling. In TE-PVAE, we replace the PointNet with a Set Transformer, which contains several set attention blocks (SABs), followed by pooling by multihead attention (PMA). The SAB and PMA are described in Lee et al. (2019), and details of the multi-head attention block are in the appendix in Figure 4.

Transformer PVAE (T-PVAE) While TE-PVAE has a powerful encoder, it retains the simple multilayer perceptron decoder of P-PVAE, which is a potential bottleneck. To address this, in T-PVAE both the encoder and the decoder are composed of multiple SABs. The latent dimension (the dimension of $\mathbf{z}$) is equal to the total number of features (i.e., for tabular data, the number of columns) times the transformer embedding dimension. The mean and variance of $q(\mathbf{z}|\mathbf{x})$ are read out using a linear layer from the outputs of the transformer encoder.

Rupert To evaluate the importance of using latent variables to handle partially observed data, we also implemented a simple transformer model to directly estimate $p(x_i|\mathbf{x}_O), i \in U$, which we call Rupert. The model consists of a stack of SABs between a feature embedder and readout layer borrowed from the PVAE. Thus, it is similar to BERT (Devlin et al., 2018), but with minimal adaptations so that — like its namesake — it will ingest inputs of various types indiscriminately. Rupert is not a VAE and does not explicitly represent dependencies between unseen features, but one could sample from $p(\mathbf{x}_U \mid \mathbf{x}_O)$ autoregressively, by predicting one feature at a time and adding the already-predicted features to the input of Rupert at each step.

2.2 Implementation

Code is available at https://github.com/microsoft/project-azua.

Training We train all three PVAE models in the same way. We randomly mask out some of the observed features in each training example and maximise the ELBO:

$$\log p(\mathbf{x}_A, \mathbf{x}_B|z) - D_{KL}\left(q(\mathbf{z}|\mathbf{x}_A)||p(\mathbf{z})\right),$$

where A denotes unmasked features whose values are input to the model, B denotes the randomly masked features, and $A \cup B = O$. This is slightly different from the ELBO used in Ma et al. (2018), where the likelihood term is just $\log p(\mathbf{x}_A|z)$. We found that this modification, which directly rewards imputation during training, improves imputation performance at test time. Whereas (Ma et al., 2018) maximises an ELBO for $\mathbf{x}_A$, our modification means we now maximise an ELBO for $\mathbf{x}_O = \mathbf{x}_{A\cup B}$, subject to a constraint that the inference network must evaluate $q(\mathbf{z}|\mathbf{x})$ with values of $\mathbf{x}_B$ masked out, i.e. with some dropout.

We train Rupert by minimizing $-\sum_{i\in U} \log p(x_i|\mathbf{x}_O)$, similar to masked language model pre-training.

Preprocessing and feature embedding As in Ma et al. (2018), categorical variables are one-hot encoded and continuous variables are normalised before passing them into any of the models. We then concatenate each value with a learned *feature embedding*. For transformer models, the feature embedding plays a similar role to the *positional encoding* of Vaswani et al. (2017).

Masking In our setting, different numbers of features may be observed in each example within a minibatch, and we handle this using masking. For TE-PVAE, we use a mask in the multihead attention functions to enforce that the multihead attention does not attend to the masked keys/values. However, masked values are still passed as queries in the SABs, and it is only after the PMA block that we have outputs that do not depend at all on the masked values. In T-PVAE and Rupert there is no PMA and so we handle masking differently. In these models, the feature embedder transforms an observed value x_i for feature i to $(x_i, \mathbf{e_i}, 1)$ (where $\mathbf{e_i}$ is the learned feature embedding), and an unobserved value to $(0, \mathbf{e_i}, 0)$. No masking is used the multihead attention functions.

	P-PVAE	TE-PVAE	T-PVAE	Rupert
Boston	0.17	0.17	0.17	**0.16**
Concrete	0.18	**0.14**	0.17	0.16
Eedi	0.282	0.279	**0.274**	0.275
Energy	0.22	**0.18**	0.23	0.20
Iris	0.59	0.45	**0.32**	0.39
Kin8nm	0.27	0.26	0.28	**0.25**
MNIST	0.10	**0.083**	0.26	0.25
Wine	0.17	0.17	0.17	**0.15**
Yacht	0.24	0.18	0.24	**0.17**

Table 1: Imputation performance of our models. For most datasets, we show performance in terms of normalised RMSE. The exception is the Eedi dataset, where all variables are categorical, and we show misclassification rate instead.

Scaling to large numbers of features. Standard self-attention has complexity $\mathcal{O}(N^2)$ in the number of input features N, which becomes prohibitive for some datasets e.g., image datasets where each pixel is a feature. For such datasets, we replace the SAB with the induced set attention block (ISAB) of Lee et al. (2019) to reduce the computational and memory complexity of self-attention from quadratic to linear.

Model applications The aforementioned models can be used both for imputing missing data, and for personalised data acquisition similar to (Ma et al., 2018).

To estimate a missing variable x_i using a PVAE-based model, we take multiple samples $\mathbf{z}_1, ..., \mathbf{z}_k \sim q(\mathbf{z}|\mathbf{x}_O)$, and then report the mean (if continuous) or mode (if categorical) of the equally weighted mixture of $p(x_i|\mathbf{z}_1), ..., p(x_i|\mathbf{z}_k)$. For Rupert, we report the mode of $p(x_i|\mathbf{x}_O)$.

For personalised data acquisition using TE-PVAE and T-PVAE, we can use the same objective and inference method as PVAE Ma et al. (2018). Rupert does not support the same objective as it is not a probabilistic method. Thus, we introduce predictive variance reduction as an objective for Rupert for the same task. The algorithm can be found in the appendix.

3 Evaluation

Datasets We evaluate our models on UCI regression datasets (Dua and Graff, 2017), MNIST (LeCun et al., 1998), as well as data from the online education platform Eedi which consists of students' answers to hundreds of different multiple-choice maths questions (Wang et al., 2020). We treat the Eedi data as a table with one row per student and one column per question.

Metrics To evaluate imputation performance, we randomly mask out 30% of variables in each example. We produce a point estimate for each missing value. To combine errors for categorical and continuous variables into a single metric, we use the discrete metric for categorical values and report normalised RMSE over all variables.

We evaluate performance on personalised data acquisition using *area under the information curve* (AUIC) as in Ma et al. (2018). The information curve is the graph of normalised RMSE in the target variable versus the number of variables queried.

Results Table 1 summarises imputation performance of the presented models. On all datasets, P-PVAE is outperformed by one of the transformer-based models. On MNIST, TE-PVAE outperforms P-PVAE.

Table 2 summarises data acquisition performance for the UCI datasets. In most cases, when using EDDI, TE-PVAE gives the best results. Rupert (with the variance-based algorithm) gives the best results overall, probably because the objective used to select features aligns with the RMSE-based metric more directly.

	P-PVAE		TE-PVAE		T-PVAE		Rupert	
	EDDI	R	EDDI	R	EDDI	R	Var	R
Boston	2.03	2.04	**<u>1.46</u>**	1.67	1.58	1.80	1.59	1.75
Concrete	1.48	1.59	**1.45**	1.52	1.65	1.67	<u>1.20</u>	1.45
Energy	1.18	1.18	**0.91**	1.00	1.35	1.51	<u>0.85</u>	1.12
Iris	2.80	2.80	2.73	2.73	**2.10**	1.90	<u>1.87</u>	<u>1.87</u>
Kin8nm	**1.28**	1.39	1.31	1.31	1.65	1.65	<u>1.06</u>	1.28
Wine	2.14	2.78	**1.42**	1.42	1.83	2.40	<u>1.03</u>	1.81
Yacht	0.94	1.31	0.63	0.63	**0.57**	1.06	<u>0.42</u>	0.98

Table 2: Personalised data acquisition, evaluated on UCI datasets. EDDI: information theoretical objective from Ma et al. (2018); R: random query order; Var: Algorithm 1 using variance objective. Best results using EDDI and PVAE are in **bold** and best results using any algorithm and model are <u>underlined</u>.

4 Conclusion

We presented new models for imputing missing values by using attention to propagate information between features. Our new models give improved performance in both imputation and downstream tasks, on multiple datasets. Our results also raise research questions regarding methods for downstream tasks, and we will implement variance-based data acquisition for PVAE-based models to investigate.

References

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., and Teh, Y. W. (2019). Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pages 3744–3753. PMLR.

Ma, C., Tschiatschek, S., Palla, K., Hernández-Lobato, J. M., Nowozin, S., and Zhang, C. (2018). Eddi: Efficient dynamic discovery of high-value information with partial vae. *arXiv preprint arXiv:1809.11142*.

Ma, C., Tschiatschek, S., Turner, R., Hernández-Lobato, J. M., and Zhang, C. (2020). Vaem: a deep generative model for heterogeneous mixed type data. *Advances in Neural Information Processing Systems*, 33.

Mattei, P.-A. and Frellsen, J. (2018). missiwae: Deep generative modelling and imputation of incomplete data. *arXiv preprint arXiv:1812.02633*.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Wang, Z., Lamb, A., Saveliev, E., Cameron, P., Zaykov, Y., Hernández-Lobato, J. M., Turner, R. E., Baraniuk, R. G., Barton, C., Jones, S. P., et al. (2020). Instructions and guide for diagnostic questions: The neurips 2020 education challenge. *arXiv preprint arXiv:2007.12061*.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R., and Smola, A. (2017). Deep sets. *arXiv preprint arXiv:1703.06114*.

A Appendix

Algorithm 1 Data acquisition using variance objective

Require: set I of observable variables, target variable $t \notin I$
  Observed variables $O \leftarrow$ initially observed variables $\subset I$
  while $O \neq I$ (or some other stopping criterion) **do**
    for $i \in I \backslash O$ **do**
      Sample n values $x_i^1, ..., x_i^n$ from $p(x_i|\mathbf{x}_O)$
      $r_i \leftarrow$ mean over k of $-\mathrm{Var}_{x_t \sim p(x_t|\mathbf{x}_O, x_i^k)}(x_t)$
    $O \leftarrow O \cup \mathrm{argmax}_i r_i$

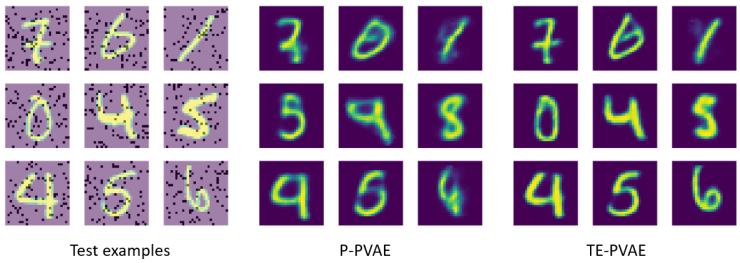

Figure 2: Inpainting images from the MNIST dataset. On the left are test images. The dark pixels are the ones whose values are given to the model; here we have masked out all but 10% of pixels. On the right are sampled completions of the images from models P-PVAE and TE-PVAE. Each completion is generated using a single forward pass through the model.

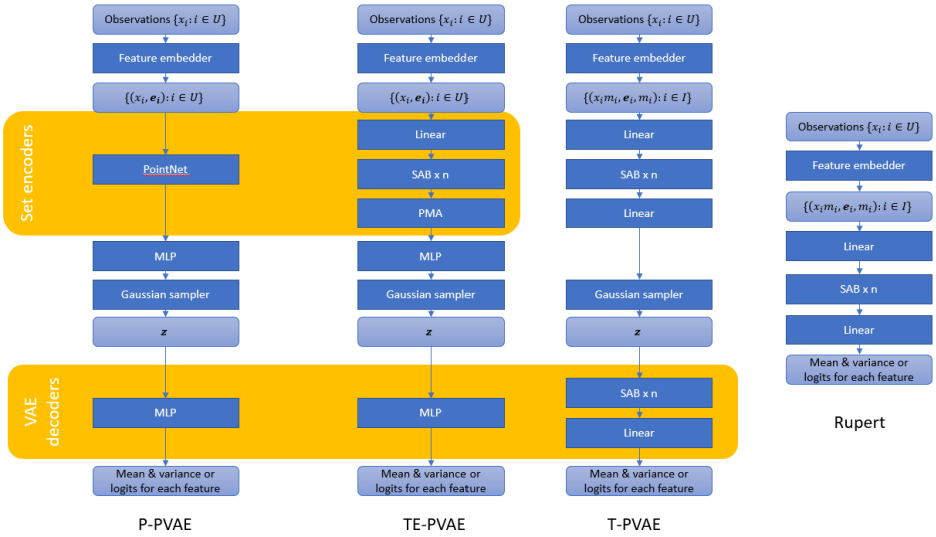

Figure 3: Detailed architecture of the models that we implemented. SAB: set attention block, MLP: multi-layer perceptron, PMA: pooling by multi-headed attention. $\mathbf{m}$ denotes the *mask* indicating which features of $\mathbf{x}$ are observed: $m_i = 0$ for unobserved $i \in U$ and $m_i = 1$ for observed $i \in O$. Pale boxes are data; dark boxes are functions.

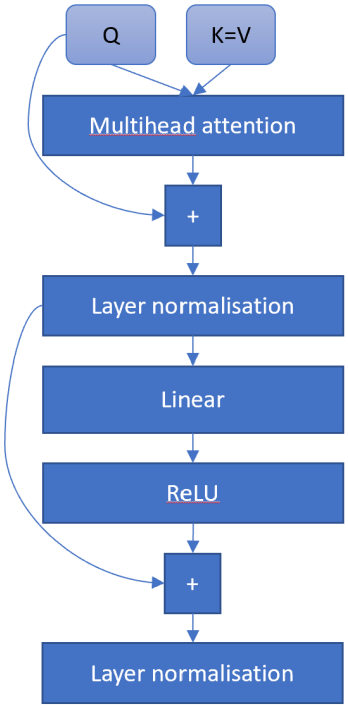

Figure 4: The multi-head attention block (MAB) used in our models. The MAB includes standard dot-product multi-head attention, layer normalisation, a single linear transformation, and a nonlinearity. Only the multi-head attention mixes information between different features $i \in I$: the linear transformation and layer normalisation act over the transformer embedding dimension.