VALUE-AWARE TRANSFORMERS FOR 1.5D DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse sequential highly-multivariate data of the form characteristic of hospital in-patient investigation and treatment poses a considerable challenge for representation learning. Such data is neither faithfully reducible to 1D nor dense enough to constitute multivariate series. Conventional models compromise their data by requiring these forms at the point of input. Building on contemporary sequence-modelling architectures, we design a *value-aware transformer*, prompting a reconceptualisation of our data as *1.5−dimensional*: a token-value form both respecting its sequential nature and augmenting it with a quantifier. Experiments focused on sequential in-patient laboratory data up to 48hrs after hospital admission show that the value-aware transformer performs favourably versus competitive baselines on in-hospital mortality and length-of-stay prediction within the MIMIC-III dataset.

1 INTRODUCTION

Modelling the sequence of events in electronic health records (EHRs) poses fundamental challenges to contemporary sequence models owing to the complex, heterogeneous nature of the data. Even choosing a good input representation for such data presents difficult trade-offs. Any resolution is highly contextual: the variety of data shapes encountered within the records for different patients and across different sections of their visit is substantial. While there may be some variables, such as patient demographics, that one might reasonably expect all patients to have an entry for, the heterogeneity of structure sets in as soon as the patient's individual health status is more intricately explored. During an admission to hospital, a series of laboratory tests (hereafter referred to as 'labs' for succinctness) may be conducted on a patient: the number of possible types of test might be large (in the hundreds), these may be measured repeatedly, once, or not at all. Hospital stays may vary significantly in length and this variation is not necessarily indicative of severity. Very ill patients may have short survival, or may be artificially sustained on life support in expectation of a recovery that fails to arrive. There are further complications: some tests may have been conducted in advance - attributed to their historical health record; or recorded in a different modality of their health record (e.g. notes, charts, etc.). The vast complexity and heterogeneity of EHR data presents representation learning with fertile ground for innovation.

1.1 RELATED WORK

The wave of excitement set in motion by transformer architectures (Vaswani et al., 2017) quickly propagated to the healthcare world. Researchers seeking to make secondary use of signal lying dormant within EHRs paid particular attention. Much of this signal is locked up in idiosyncratic natural language formats such as clinical notes. Large language models such as BERT (Devlin et al., 2018) have been subjected to rounds of secondary pre-training on medical corpora (Alsentzer et al., 2019), with the aim of attuning them to the nuances of the clinical domain.

A less apparent application of transformer architectures emerges when event sequences in an EHR are conceptualised as a sequence of words. These words can be arranged to constitute a sentence, capturing some of the qualitative aspects of a patient's stay. This approach can be pursued at various levels of granularity, for instance Li et al. (2020) use BERT to model sequences of diagnoses over multiple visits. Applying the same methodology at a finer grain suggests a natural model for within-visit medical event sequences such as labs. But while knowledge of the performance of a lab test conveys information, it is the lab test *together with its value* that is the more complete indicator of the health status of a patient.

Choi et al. (2020) covered some distance towards integrating these forms, modelling the graphical and hierarchical structure of EHRs in an exciting piece of work using graph convolutional transformers. Their method was not designed to model repeat measurements of labs - a common phenomenon within the natural course of medical investigation and treatment. The MIMIC-III lab-events data confirm this: the mean number of labs over the first 48hrs is 142.8, while the mean number of unique labs over the same time period is just 56.1 (see Table 6). With this paper, we intend to derive a principled approach to integrating the values for the sparse sequential data characteristic of EHRs.

In order to objectively compare systems such as those described above, EHR-specific benchmark tasks have been proposed, most based on the MIMIC dataset Johnson et al. (2016). Of particular relevance here are the precisely defined prediction tasks for in-hospital mortality and length-of-stay proposed in Harutyunyan et al. (2019); Purushotham et al. (2018). In related work, and with a view to further increasing reproducibility, Wang et al. (2020) provided a pre-processing pipeline that addresses concerns such as unit conversion, outlier filtering and aggregation that are common in these tasks. Whilst we agree with the spirit of work, the pipeline produces a multivariate input representation that, as we will suggest, is not optimal given the investigative nature of the event sequences.

2 DATA GREATER THAN 1D

Sequences of words such as those found in natural language are neatly represented as 1D data using tokens. Sequences of lab results such as those found early in EHRs are neatly represented as (token, value) pairs, which might suggest that the ideal representation of this data should a multivariate series with one dimension per lab type. To model it in this form would require considerable imputation due to the data's sparsity. However, the presence or missingness of particular labs in a sequence are informative: they are requested by healthcare professionals guided by clinical intuition that the EHR may not capture. By imputing the data, this valuable signal is weakened. Perhaps we can make do with just a 1D form - and if so, how should that 1D be constituted? These are the key questions that a practitioner hoping to model in our data regime faces. As we will discuss in this section, when the data is sparse and highly-multivariate, the choice is not clear *a priori*. For the motivational problem of lab-result sequences there may be only few or a large number of measurements with multiple repeat samples of the same lab - characteristics which may contain important information.

We close the section by proposing a variation on the conventional approaches, which we refer to as "$1.5-$dimensional" data: preserving the sequential nature of the lab orders, together with information about the values, but without the need for full multivariate sequence representations and the imputation issues that would ensue.

2.1 CHARACTERISTICS OF LAB-RESULT SEQUENCES

In-patient lab-result sequences are generated by a process of hospital investigation and treatment which imbue them with particular characteristics:

- Relevance - the labs are pertinent to the health investigation being conducted on the patient. Only a small proportion of the all of the labs available might be obtained during any given visit.
- Repeats - these relevant measurements may be taken once, or repeated many times.
- Pragmatism - difficult or inconvenient labs are not taken unless absolutely necessarily. This leads to a rare-token problem also commonly encountered in natural language.

If we believe that these characteristics contain valuable information, then the way that we represent our data should not compromise them.

2.2 COLLAPSING DATA DOWN TO 1D

A common way to deal with this difficult data dimensionality is to 'collapse' it to 1D. There are two natural ways of doing this: the horizontal collapse and the vertical collapse.

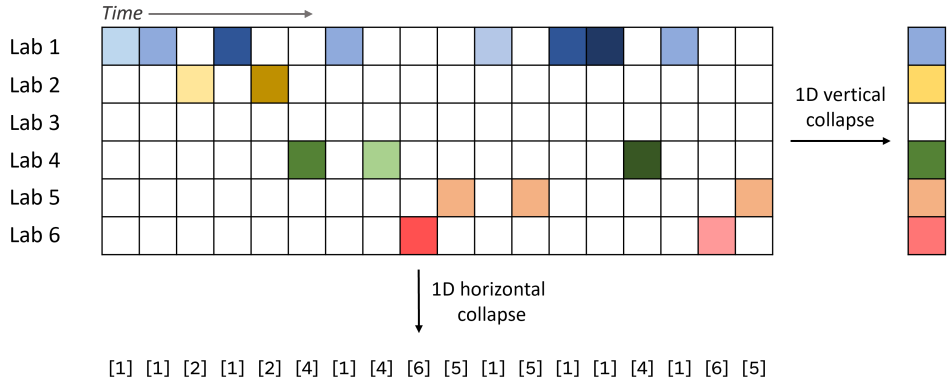

Figure 1: The 1D horizontal data collapse forms a sequence of lab tokens while ignoring their values; a 1D vertical collapse takes a summary statistic (e.g. the mean) for each lab sequence.

Table 1: Comparison of number of inputs required for 1D , 1.5D and multivariate sequence representations, where: l is # variables, n is maximum window size, m is # data entries in a particular sequence.

Data dimensionality	**# Inputs**	**# Imputations**	**Collapsed information**
1D (vertical)	l	$\in [l-m, l-1]$	sequence structure & repeat measures
1D (horizontal)	n	$n-m$	values of measurements
1.5D	$2n$	$2(n-m)$	raw value-scales
Multivariate sequences	$l \cdot n$	$l \cdot n - m$	N/A

Horizontal collapse. The horizontal collapse retains the sequential structure of the data, and removes the value-aspect. One form of the horizontal collapse is to ignore the value from the token-value pair. We then have a sequences of lab orders, and the hope is that there is sufficient information contained in the *presence* of a lab order to inform the underlying state of a patient. This capitalises on recent advancements in NLP and has formed the basis for a number of medical sequence modelling applications as mentioned in Section 1.1.

Vertical collapse. An alternative form of 1D collapse is to collate repeated values of a variable using a summary measure (such as the mean, median, max, most-recent) and then use these summary measures – one per variable – as the input to a model. This representation respects the unique value scales on a per-variable basis, however it compromises the sequential nature of the data and collapses repeated measurements.

2.3 Imputing for Multivariate Sequences

An enticing candidate for the representation of (token, value) pairs is as a collection of sequences: one sequence per token. This retains each variable's independent value-scale and the innate temporally-ordered nature of the data. However, the sparsity of our specific data regime is problematic, affecting its suitability. The usual, but de-emphasised, approach is to conduct a program of imputation, be it: sequential forward-filling of data; per-variable mean imputation; or the use of sentinel values - the choice of which will necessarily depend on the variable in question (so as not to inappropriately assign a sentinel value that is commonly attained by *true* data points).

Imputation is widely regarded as a necessary evil in such situations. Not all missingness is made equal and even the most advanced general-purpose imputation schemes rely on assumptions which are unrealistic in a healthcare setting. Due to the high-risks associated with the healthcare setting unrealistic imputation can have serious implications.

2.4 1.5D DATA

Both the 1D collapse and the multivariate sequence approaches introduce their own issues or simplifications for sparse sequential highly-multivariate data. We propose to interpolate these data regimes and use an alternative data format: ([TOK], [VAL]), where [TOK] is the name of the variable and where [VAL] is its value, which has been transformed and discretised in such a way that it is meaningfully comparable across all variables. This:

- retains the sequential nature of the data: it is a 1D sequence of tokens.
- removes the need for considerable imputation (cf. Table 1)
- reduces the value dimension to a common scale, contextualised by its token.

For numeric labs values this can be achieved via *quantilisation*: conditional population statistics for the lab results can be used to attribute a category to each of the values, depending on which quantile the value lies in. If there are no prior distributional statistics available, they can be extracted from the training set data (for simplicity and generality we took this approach in our analysis). For categorical labs we can encode them in any manner which seems natural as long as it is consistent. Where no category can be associated, a sentinel value may be used - this may be necessary in the case where a result is not recorded, or the lab itself is not seen within the training set.

For our experiments we assigned the following value tokens:

- [XLOW]: the value lies below the 10% percentile
- [LOW] : the value lies between the 10% and 25% percentiles
- [MID] : the value lies between the 25% and 75% percentiles
- [HIGH] : the value lies between the 75% and 90% percentiles
- [XHIGH] : the values lies above the 90% percentile.

The token [UNK] was assigned if the lab did not feature in the train set, or if there was no numeric lab value associated with the lab.

This specific encoding captures the fundamental quality of the values, but should not be read as definitive. The [VAL] tokens should be set in context by their associated [TOK]. In particular, it is important to note that the sequence of [VAL]s, while plausibly carrying some autoregressive signal, are not in-of-themselves readily interpretable as the sequential data of a single variable. They are a value-augmentation of a $1-$dimensional sequence of tokens hence the apt moniker "1.5D data".

3 VALUE-AWARE TRANSFORMERS

Transformer architectures in their various guises have shown exceptional performance on tokenised sequence modelling tasks, but they lack a principled approach for learning (variable, value) sequences. The 1.5D data form lays a foundation for doing so. We outline guiding principles which we believe are important for utilising this input form based of the discussion in Section 2.4, then design a neat and minimalistic transformer architecture for generating it.

3.1 DESIGN PRINCIPLES

1. Tokens are primary. First, there were tokens, and then came their values.

Principle 1 makes a statement about precedence. The sequence of values [VAL], by themselves, are both bereft of meaning and without unambiguous autoregressive interpretation. To benefit from these qualities the model must contextualise them using their tokens.

2. Values should be loosely-coupled to their token.

Principle 2 is most easily defined by contrast with a *strong-coupling* where each token [TOK] and its quantifier, the value [VAL], is explicitly concatenated during preprocessing to form a new token [TOK−VAL]. Strong-coupling drops a fundamental symmetry present in the raw data: the fact that

the pairs ([TOK], [VAL1]) and ([TOK], [VAL2]) are of the same variable would now need to be learnt by the model if using this representation. This concatenation approach also scales poorly with the number of tokens and values, exacerbating already present token-sparsity concerns. At the other extreme there is *no-coupling*. No-coupling is to treat the tokens and values as completely separate entities. Modelling them in this way both causes the 1.5D form to lose its meaning and violates principle 1.

3. Values should have the opportunity to influence the autoregressive learning of the token sequence.

While the core autoregressive properties of the 1.5D sequence lie amongst the tokens, there may be useful information in their associated values which we should aim to capitalise on.

3.2 TAILORING A VALUE-AWARE TRANSFORMER

The temporal nature of our sequence data make a decoder-only style transformer a compelling candidate architecture. Large decoder-only models such a GPT-3 (Brown et al., 2020) have demonstrated astonishing performance, especially for natural language generation, and while data generation is not the primary use-case here, the future-masking inductive bias is an appropriate one.

We highlight three core components that adapt a plain decoder-only transformer to the 1.5D data.

Embeddings. To respect principle 1 and ensure loose-coupling of the ([TOK], [VAL]) pairs we use separate embedding layers for the tokens and the values. With our application in mind, we map the tokens to $\mathbb{R}^{100}$ and the values to $\mathbb{R}^{5}$. These two embeddings are then concatenated. Only after this point is a positional embedding added.

Attention. The loosely-coupled embedding concatenation means that information about the values is included within the embeddings that are propagated through the attention layers. This is key so that principle 3 is respected. We use a now established bag of tricks with the normalise-attend-feedforward (with residuals) structure for the attention layers constituting our value-aware transformer.

Output layers. The choice of autoregressive prediction head for the transformer is another crucial design consideration. Both values and tokens should be predicted, and how this is done has a significant effect on the overall learning of the model because of its proximity to the loss calculation.

Splitting the final attention layer's output embedding into two parts proved to work better than sharing the entire embedding by some margin. In respect of principle 1, the lion's share is dedicated to next-token prediction and the remaining part of the embedding used for next-value prediction. Dedicated token-prediction and value-prediction layers are then attached to their respective segment of the embedding. To truly respect the priority of tokens, the value-prediction layer is made conditional on the predicted token, explicitly including it (whether correct or incorrect) as an input feature to be used for value-prediction.

Finally, we control the relative weight given to the cross-entropy losses for the tokens $\mathcal{L}_{\texttt{[TOK]}}$ and the values $\mathcal{L}_{\texttt{[VAL]}}$ by using a hyperparameter $\gamma \in [0, 1]$.

$$\mathcal{L} = \gamma \mathcal{L}_{\texttt{[TOK]}} + (1 - \gamma) \mathcal{L}_{\texttt{[VAL]}} \tag{1}$$

3.2.1 PRE-TRAINING

With the core building blocks in place we move to pre-training. With a view to later finetuning tasks we include one type of special token: an [EOS] token for the end-of-sequence. The [EOS] has a distinguished role as the token upon whose output embedding we perform classification or regression. There is a different [EOS] for the tokens ($\in \mathbb{R}^{100}$) and for the values ($\in \mathbb{R}^{5}$).[1]. The last special token we include is a [PAD] - to fill up empty slots in the sequence. We ignore the [PAD] class when calculating the cross-entropy losses in equation 1.

[1] These are always concatenated together during the input embedding stage.

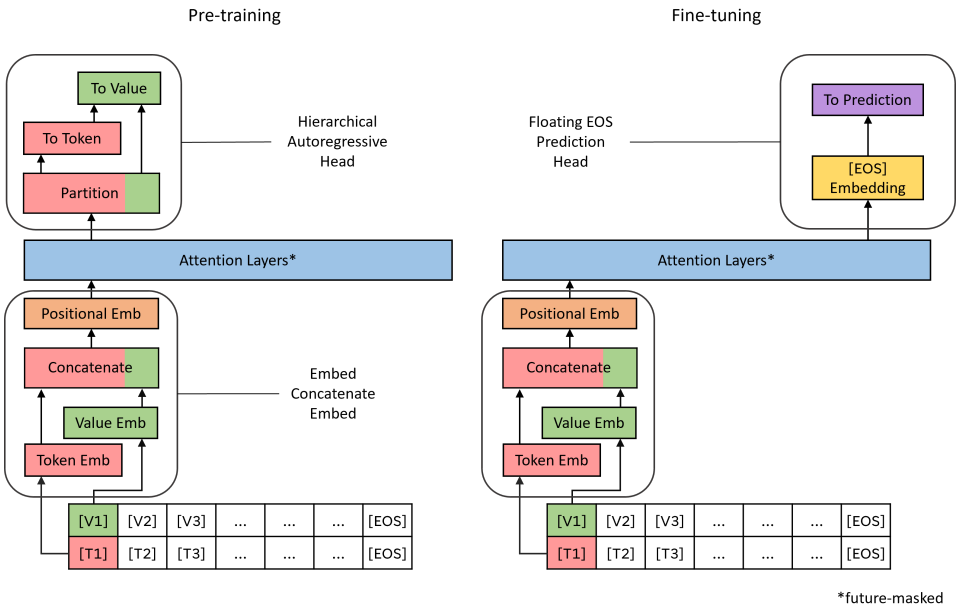

Figure 2: The architectures for pre-training and fine-tuning a value-aware transformer

We choose to set the hyperparameter γ nearer to 1 than to 0, both in respect of principle 1 and because our number of tokens ($\approx$ 750) is considerably larger than the number of quantilised values ($\approx$ 10), hence the baseline chance of correctly predicting a `[VAL]` is much higher than that of predicting a `[TOK]`. Learning rates across the order of magnitude 1e-4 to 1e-5 were consistently performant when used with gradient-clipping.

3.2.2 Fine-tuning

Fine-tuning a value-aware transformer for a classification task consists of replacing the autoregressive output layers with a classification head, directly harnessing the embeddings output from the top attention layer. There is a small but important difference in approach from common bidirectional language models. There the first token is often denoted as `[CLS]` and is imbued with a special classification function with a classifier head being attached to it. Because we are using a decoder-style network, classification on a start-of-sequence token would be practically useless - our value-aware transformer has no contextual understanding at this point. In our setting classification is instead done on the `[EOS]` token: we use a floating classification head which attaches itself at this position, taking as its input the embedding of `[EOS]`.

The classifier we used was a straight-forward one-hidden-layer feed-forward neural net with the hidden layer chosen to be of dimension 100 and with an output layer to the number of classes relevant for the classification problem. For regression tasks we use exactly the same set up except with a single class output and a further non-linearity. Dropout of probability 0.5 at the hidden layer combined with learning rates between the orders of magnitude 1e-4 to 1e-5 worked well across our problems[2], especially when combined with exponential scheduler decay at a rate of ≈ 0.9. With this set-up, after just a few epochs, the value-aware transformer was at peak performance.

4 Experiments

MIMIC-III ('Medical Information Mart for Intensive Care')[3] is a core resource for machine-learning researchers pursuing the advancement of healthcare. It is a large publicly accessible anonymised database comprising comprehensive information relating to patients admitted to the critical care

[2]These are highly-imbalanced prediction problems - more details in Section 4.

[3]Available at `https://physionet.org/content/mimiciii/1.4/`

unit at a large US hospital. The types of data structure encountered within it are exemplar of the electronic health records of healthcare systems across the world.

4.1 MIMIC-III LAB-EVENTS

Lab-event sequences occurring early within an admission to hospital embody the sparse sequential highly-multivariate data characteristics described above, making the MIMIC-III lab-events an excellent benchmark. We focus our experiments on the lab-events data which is available up until 48 hours after admission, pre-training a value-aware transformer to learn its autoregressive structure. We then tackle three challenging downstream problems by fine-tuning the value-aware transformer. The first and second are classification tasks: in-hospital mortality prediction within 3 days, and within 7 days; the third is a regression task: predicting the remaining-length-of-stay (LOS) at 48 hours.

4.1.1 DATA INCLUSION & PRE-PROCESSING

For a visit to meet our inclusion criteria it must have lasted at least 48 hours from admission to discharge. It must have at least ten lab-events recorded and death must not occur within the 48 hours. Those visits that met these criteria were then randomly split on the patient's id into Train, Dev. and Test sets with the proportions 0.8 / 0.1 / 0.1. Some patients have multiple visits so it was important to partition based on the patient so as not to introduce a leak. The partition statistics are detailed in the Appendix A, table 6.

Minimal processing of the data was performed so as not to collapse exploitable signal within the raw data. The visits in the Train set were used to extract per-lab quantile distribution following the procedure outlined in Section 2.4. From these derived distributions, we formatted each lab-event in the Train, Dev. and Test sets as $(\texttt{[LAB]}, \texttt{[VAL]})$, with $\texttt{[VAL]} \in \{\texttt{[XLOW]}, \texttt{[LOW]}, \texttt{[MID]}, \texttt{[HIGH]}, \texttt{[XHIGH]}, \texttt{[UNK]}\}$. Finally, with visits represented as sequences of token-value pairs, we append a special end-of-sequence token `[EOS]` to mark the 48 hour point. The `[TOK]` vocabulary was formed of 753 labs, which together with the two special tokens `[EOS]` and `[PAD]` brings the size to 755. The `[VAL]` vocabulary was formed of the 6 tokens detailed above, plus the two special tokens `[EOS]` and `[PAD]`, totalling 8 values.

4.2 EXPERIMENT 1: PRE-TRAINING

In the first experiment we investigated whether incorporating values assists in the prediction of tokens by comparing to a plain transformer architecture of the same base architecture. We initialised them with the same hyperparameters (learning rate 5e-5) and captured their respective token cross-entropy losses $\mathcal{L}_{\texttt{[TOK]}}$. We chose the input sequence length to be 250 for each of the models, since this sequence length choice captures the entirety of 90% of the visits in the Train set (cf. Appendix A, Table 6). For the value-aware transformer we set the loss hyperparameter $\gamma = 0.9$ to put greater weight on the learning of the tokens and to counteract the value's greater at-random probability of being correctly predicted.

With the competition set up, we let each run until it hit an early-stopping threshold of 7 on the Dev. set. The results in Table 2 demonstrate a promising improvement over the transformer, with better average next token prediction for both the small (depth 4) and large (depth 6) versions of the model. Comparing the two depth 6 models, the value-aware transformer provides a 3.2% boost over its transformer sibling in the average probability of correct token prediction, whilst also generating values.

The successful outcome of experiment 1 demonstrates that there is both valuable signal in the 1.5D representation, and that the value-aware transformer is able to exploit it. The second and third experiments test whether we can channel its superior generative modelling into a performance gain on downstream tasks.

4.3 EXPERIMENT 2: IN-HOSPITAL MORTALITY PREDICTION

In-hospital mortality is a challenging clinical outcome to predict because of its highly-imbalanced nature: thankfully a minority of the patients admitted to hospital die. Just 0.84% of visits in our

Table 2: Pre-training performance: value-aware transformer vs. a transformer on Dev. set

Model	Data dim.	$\mathcal{L}_{[\texttt{TOK}]}$	$p_{[\texttt{TOK}]}$	$\mathcal{L}_{[\texttt{VAL}]}$	$p_{[\texttt{VAL}]}$
Transformer (depth 4)	1	1.331	0.2642	–	–
VA-transformer (depth 4)	1.5	1.322	0.2674	1.347	0.2600
Transformer (depth 6)	1	1.319	0.2666	–	–
VA-transformer (depth 6)	1.5	**1.291**	**0.2750**	**1.332**	**0.2639**

Table 3: In-hospital mortality $\leq$ 3 days classification performance on Test set.

Model	Data dim.	Mortality $\leq$ 3 days	
		Balanced-accuracy	ROC-AUC
FFNN values (mean)	1 v	0.577 ± 0.044	0.610 ± 0.033
FFNN quants (mean, one-hot)	1 v	0.546 ± 0.064	0.736 ± 0.014
FFNN quants (mean, integer)	1 v	0.603 ± 0.028	0.707 ± 0.030
Transformer (depth 4)	1 h	0.666 ± 0.013	0.761 ± 0.007
Transformer (depth 6)	1 h	0.663 ± 0.020	0.752 ± 0.013
VA-Transformer (depth 4)	1.5	$\mathbf{0.725 \pm 0.039}$	$\mathbf{0.825 \pm 0.020}$
VA-Transformer (depth 6)	1.5	0.716 ± 0.025	0.813 ± 0.012

Train set result in death within 3 days of admission. Accuracy is a poor metric of success here - we can attain 99.16% by always predicting that there is no mortality event. Instead we optimise for balanced accuracy, using a Train set propensity-weighted binary cross-entropy loss. We consider two targets: mortality $\leq$ 3 days (which has Train set propensity of 0.0084), and mortality $>$ 7 days (whose Train set propensity is 0.0496).

We compare the performance of a fixed-specification classifier head (a one-hidden layer feed-forward neural network with hidden dimension 100) when trained on various types of 1D vertical collapse (listed as 'FFNN'), with the use of the same classifier head for the finetuning of the plain transformer model (using the 1D horizontal representation) and the value-aware transformer (using the 1.5D representation). Each is trained with an early stopping threshold of 3.

To determine a competitive setting for the FFNN baselines we ran a hyperparameter search against the Dev. set. The all-round best-performing hyperparameters from the search were then chosen and 5 separate training runs were performed for each model, checkpointed at the epoch with the best Dev. set loss. For the transformers and value-aware transformers, we finetuned the 4 pre-trained models from Experiment 1 (Section) five times each from scratch, checkpointing them at the epoch with the lowest Dev. set loss. The 5 runs of each competitor were then loaded from their checkpoints and evaluated against Test set[4], with the mean $\pm$ standard deviation of key metrics reported in Table 3 and Table 4.

4.4 Experiment 3: LOS prediction

To round off our experiments we choose a regression task and length-of-stay is another core problem faced by caregivers. The accurate prediction of LOS enables more efficient delivery of care. When the hospital is under strain or run inefficiently there is a negative impact on patient outcomes.

We use the same prediction network as for experiment 2 the only differences being a single-class output and a final soft-plus non-linearity, for the reason that the remaining-LOS cannot be non-negative. We compare this regression network's performance when the input takes the form of the 1D vertically collapsed data with a finetuned transformer trained on the 1D horizontally collapsed data and a value-aware transformer trained on our 1.5D representation.

[4]The Dev. set performance is detailed in Appendix A, Tables 8 & 9.

Table 4: In-hospital mortality > 7 days classification performance on Test set.

Model	**Data dim.**	**Mortality > 7 days**	
		Balanced-accuracy	ROC-AUC
FFNN values (mean)	1 v	0.624 ± 0.030	0.661 ± 0.054
FFNN quants (mean, one-hot)	1 v	0.704 ± 0.005	0.779 ± 0.015
FFNN quants (mean, integer)	1 v	0.686 ± 0.007	0.772 ± 0.006
Transformer (depth 4)	1 h	0.695 ± 0.006	0.777 ± 0.002
Transformer (depth 6)	1 h	0.699 ± 0.010	0.776 ± 0.006
VA-Transformer (depth 4)	1.5	$\mathbf{0.723 \pm 0.016}$	$\mathbf{0.808 \pm 0.003}$
VA-Transformer (depth 6)	1.5	$\mathbf{0.724 \pm 0.005}$	$\mathbf{0.810 \pm 0.003}$

Table 5: Remaining length-of-stay regression performance on Test set.

Model	**Data dim.**	**Remaining LOS at 48hrs**	
		MSE	R^2
FFNN raw values (mean)	1 v	158.6 ± 6.6	0.091 ± 0.038
FFNN quants (mean, one-hot)	1 v	131.5 ± 0.9	0.246 ± 0.005
FFNN quants (mean, integer)	1 v	131.6 ± 1.3	0.246 ± 0.007
Transformer (depth 4)	1 h	132.4 ± 1.7	0.241 ± 0.010
Transformer (depth 6)	1 h	131.5 ± 1.2	0.246 ± 0.007
VA-Transformer (depth 4)	1.5	$\mathbf{126.9 \pm 1.4}$	$\mathbf{0.273 \pm 0.008}$
VA-Transformer (depth 6)	1.5	$\mathbf{125.2 \pm 3.0}$	$\mathbf{0.282 \pm 0.017}$

Both the depth 4 and the depth 6 value-aware transformer surpass the competitor models by a healthy margin, on the LOS regression task, providing further compelling evidence that additional information of worth is retained in the 1.5D input representation and that the value-aware transformer is able to convert it into superior downstream performance.

5 Conclusion

This paper is devoted to sparse sequential highly-multivariate data - a form which is found throughout the healthcare domain and whose value is difficult to unlock. We have analysed conventional approaches to modelling this data using representation learning methods, highlighting the 1D vertical and horizontal collapses.

Offering a fresh perspective, we have reconceptualised the raw data as 1.5-dimensional and presented a token-value form which retains both sequential and value structure without forcing considerable imputation before the point of input to a model. We have outlined principles for the appropriate modelling of this data form, and based on these principles, made targeted adaptations to a transformer architecture, efficiently tailoring it to the generation of 1.5D data.

In a series of experiments on the MIMIC-III lab-events dataset we have demonstrated both the appropriateness of the 1.5D data representation and the suitability of the *value-aware transformer* by clearly surpassing competitive baselines both as an autoregressive model, and on the challenging real-world in-hospital mortality (classification) and length-of-stay (regression) prediction tasks.

Reproducibility Statement

The codebase supporting this work has been made available with the submission. Included within it are: the scripts used to pre-process the data; a va_transformers package; the scripts used for the pre-training, fine-tuning and baseline experiments. The pre-processing follows the procedure outlined Section 4.1.1, with the MIMIC-III dataset available at `https://physionet.org/content/mimiciii/1.4/`.

REFERENCES

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*, 2019.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Edward Choi, Zhen Xu, Yujia Li, Michael Dusenberry, Gerardo Flores, Emily Xue, and Andrew Dai. Learning the graphical structure of electronic health records with graph convolutional transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 606–613, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1):96, 2019. ISSN 2052-4463. doi: 10.1038/s41597-019-0103-9. URL https://doi.org/10.1038/s41597-019-0103-9.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Yikuan Li, Shishir Rao, Jose Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. Behrt: transformer for electronic health records. *Scientific reports*, 10(1):1–12, 2020.

Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. Benchmarking deep learning models on large healthcare datasets. *Journal of Biomedical Informatics*, 83:112–134, 2018. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2018.04.007. URL https://www.sciencedirect.com/science/article/pii/S1532046418300716.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Shirly Wang, Matthew BA McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C Hughes, and Tristan Naumann. Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pp. 222–235, 2020.

A APPENDIX

Table 6: MIMIC-III labs data statistics after preprocessing

Partition	**Train**	**Validation**	**Test**
# Patients	32989	4124	4124
# Visits	41403	5164	5072
# Labs/visit mean	142.8	142.3	143.8
# Labs/visit median	127.0	128.0	129.0
# Labs/visit 90%	248.0	245.7	250.0
Remaining LOS mean (hrs)	219.3	227.7	226.8
Remaining LOS median (hrs)	125.6	127.3	130.4
Mortality $\leq$ 3 days	0.00841	0.00833	0.00848
Mortality $>$ 7 days	0.04959	0.04493	0.05205

Table 7: Remaining length-of-stay regression performance on the Dev. set

Model	**Data dim.**	**Remaining LOS at 48hrs**	
		MSE	R^2
FFNN values (mean)	1 v	158.9 $\pm$ 3.7	0.106 $\pm$ 0.021
FFNN quants (mean, one-hot)	1 v	131.2 $\pm$ 1.0	0.262 $\pm$ 0.006
FFNN quants (mean, integer)	1 v	131.1 $\pm$ 1.3	0.262 $\pm$ 0.007
Transformer (depth 4)	1 h	130.1 $\pm$ 1.2	0.268 $\pm$ 0.007
Transformer (depth 6)	1 h	129.6 $\pm$ 1.0	0.271 $\pm$ 0.006
VA-Transformer (depth 4)	1.5	**124.3** $\pm$ **1.9**	**0.300** $\pm$ **0.011**
VA-Transformer (depth 6)	1.5	**123.3** $\pm$ **2.2**	**0.306** $\pm$ **0.012**

Table 8: In-hospital mortality $\leq$ 3 days classification performance on the Dev. set

Model	**Data dim.**	**Mortality $\leq$ 3 days**	
		Balanced-accuracy	**ROC-AUC**
FFNN values (mean)	1 v	0.557 $\pm$ 0.049	0.638 $\pm$ 0.024
FFNN quants (mean, one-hot)	1 v	0.556 $\pm$ 0.077	0.784 $\pm$ 0.008
FFNN quants (mean, integer)	1 v	0.604 $\pm$ 0.037	0.732 $\pm$ 0.008
Transformer (depth 4)	1 h	0.698 $\pm$ 0.020	0.800 $\pm$ 0.012
Transformer (depth 6)	1 h	0.694 $\pm$ 0.013	0.797 $\pm$ 0.009
VA-Transformer (depth 4)	1.5	0.737 $\pm$ 0.049	**0.845** $\pm$ **0.019**
VA-Transformer (depth 6)	1.5	0.726 $\pm$ 0.020	**0.831** $\pm$ **0.014**

Table 9: In-hospital mortality $>$ 7 days classification performance on the Dev. set

Model	**Data dim.**	**Mortality $>$ 7 days**	
		Balanced-accuracy	**ROC-AUC**
FFNN values (mean)	1 v	0.616 $\pm$ 0.041	0.649 $\pm$ 0.055
FFNN quants (mean, one-hot)	1 v	0.696 $\pm$ 0.007	0.784 $\pm$ 0.015
FFNN quants (mean, integer)	1 v	0.691 $\pm$ 0.008	0.779 $\pm$ 0.003
Transformer (depth 4)	1 h	0.685 $\pm$ 0.004	0.773 $\pm$ 0.004
Transformer (depth 6)	1 h	0.687 $\pm$ 0.008	0.772 $\pm$ 0.003
VA-Transformer (depth 4)	1.5	**0.727** $\pm$ **0.005**	**0.817** $\pm$ **0.006**
VA-Transformer (depth 6)	1.5	**0.731** $\pm$ **0.007**	**0.816** $\pm$ **0.004**