Learning Graphical Structure of Electronic Health Records with Transformer for Predictive Healthcare

Edward Choi [1] **Mike W. Dusenberry** [1] **Gerardo Flores** [1] **Zhen Xu** [1] **Yujia Li** [2] **Yuan Xue** [1] **Andrew Dai** [1]

Abstract

Effective modeling of electronic health records (EHR) is rapidly becoming an important topic in both academia and industry. A recent study showed that utilizing the graphical structure underlying EHR data improves the performance of prediction tasks such as heart failure diagnosis prediction. However, EHR data do not always contain the complete structural information. Moreover, when it comes to claims data, they do not have any structural information to begin with. Under such circumstances, can we still do better than just treating EHR data as a flat-structured *bag-of-features*? In this paper, we study the possibility of utilizing the implicit structure of EHR by using Transformer for prediction tasks on public electronic health records. Specifically, we make a connection between graph networks and Transformer, then use a variant of Transformer on encounter-based prediction tasks such as medication prediction and masked node prediction. Our model empirically demonstrates superior prediction performance to previous approaches on two publicly available EHR datasets, indicating that it can serve as an effective general-purpose representation learning algorithm for EHR data.

1. Introduction

Large medical records collected by electronic healthcare records (EHR) systems in healthcare organizations enabled deep learning methods to show impressive performance in diverse tasks such as predicting diagnosis (Lipton et al., 2015; Choi et al., 2016a; Rajkomar et al., 2018), learning medical concept representations (Che et al., 2015; Choi et al., 2016d;b), and making interpretable predictions (Choi et al., 2016c; Ma et al., 2017). As diverse as they are, one thing shared by all tasks is the fact that, under the hood, some form of neural network is processing EHR data to learn useful patterns from them. To successfully perform any EHR-related task, it is essential to learn effective representations of various EHR features: diagnosis codes, lab values, encounters, and even patients themselves. EHR data are typically stored in a relational database that can be represented as a hierarchical graph depicted in Figure 1. The common approach for processing EHR data with neural networks has been to treat each encounter as an unordered set of features, or in other words, a *bag of features*. However, the *bag of features* approach completely disregards the graphical structure that reflects the physician's decision process.

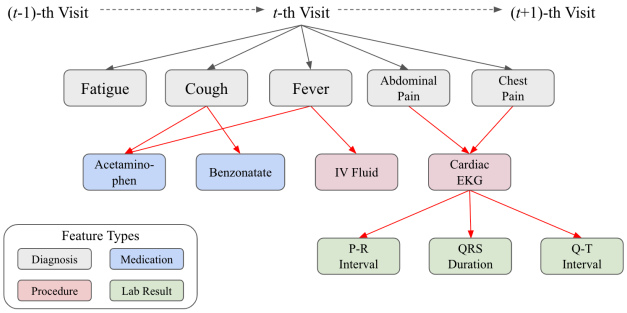

Figure 1. The graphical structure of electronic health records. A single visit consists of multiple types of features, and their connections (red edges) reflect the physician's decision process.

Recently, motivated by this EHR structure, Choi et al. (2018) proposed MiME, a model architecture that reflects EHR's encounter structure, which outperformed various *bag of features* approaches in prediction tasks such as heart failure diagnosis prediction. Their study, however, naturally raises the question: when the dataset does *not* contain structural information (the red edges in Figure 1), can we do better than *bag of features*? This question arises in many occasions, since EHR data do not always contain the complete structural information. Moreover, when it comes to claims data, there is no structural information to begin with. To address this question, we study the possibility of learning the implicit EHR structure using Transformer (Vaswani et al., 2017) on three different tasks based on encounter records.

[1]Google, Mountain View, California, USA [2]DeepMind, London, UK. Correspondence to: Edward Choi <edwardchoi@google.com>.

Presented at the ICML 2019 Workshop on Learning and Reasoning with Graph-Structured Data

Specifically, we test our approach on two publicly available EHR datasets, MIMIC-III (Johnson et al., 2016) and eICU Collaborative Research Database (Pollard et al., 2018), both of which do not contain structural information.

In the rest of the paper, we describe the graphical nature of encounter records, and make the connection between graph networks and Transformer. Then we use a variant of Transformer to learn the implicit structure of EHR while performing encounter-based prediction tasks such as medication prediction. In all tasks, Transformer consistently outperformed baseline models, showing its potential to serve as an effective general-purpose representation learning algorithm for EHR data.

2. Related Work

As briefly discussed in the introduction, this work is motivated by three seemingly independent works. *MiME* (Choi et al., 2018) derives the encounter representation in a bottom-up fashion according to the encounter structure. For example in Figure 1, MiME first combines the embedding vectors of lab results with *Cardiac EKG* embedding, which in turn is combined with both *Abdominal Pain* embedding and *Chest Pain* embedding. Then all diagnosis embeddings are pooled together to derive the final visit embedding. By outperforming various bag-of-features models in heart failure prediction and general disease prediction, MiME demonstrated the usefulness of the structural information of hospital encounter records.

Transformer (Vaswani et al., 2017) was proposed for natural language processing, specifically machine translation. It uses a novel method to process sequence data using only attention (Bahdanau et al., 2014), and is recently showing impressive performance in other tasks such as word representation learning (Devlin et al., 2018).

Graph (convolutional) networks encompass various neural network methods to handle graphs such as molecule structures, social networks, or physical experiments. (Kipf & Welling, 2016; Hamilton et al., 2017; Battaglia et al., 2018). In essence, many graph networks can be described as different ways to aggregate a given node's neighbor information, combine it with the given node, and derive the node's latent representation (Xu et al., 2019).

In this work, we focus on the fact that hospital visits inherently are graphs, and try to leverage such characteristic while performing three different encounter-based tasks using Transformer, which can be seen as a generalization of graph networks. In the next section, we first describe the graphical nature of EHR data, then make the connection between Transformer and graph networks.

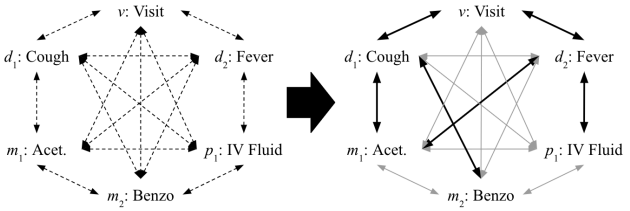

Figure 2. Learning the underlying structure of an encounter. We use Transformer to start from the left, where all nodes are implicitly fully-connected, and arrive at the right, where meaningful connections are described with thicker edges.

3. Method

3.1. Electronic Health Records as a Graph

As depicted in Figure 1, the t-th visit $\mathcal{V}^{(t)}$ starts with the visit node $v^{(t)}$ at the top. Beneath the visit node are diagnosis nodes $d_1^{(t)}, d_2^{(t)}, \ldots, d_{|d^{(t)}|}^{(t)}$, which in turn lead to ordering a set of medications $m_1^{(t)}, m_2^{(t)}, \ldots, m_{|m^{(t)}|}^{(t)}$ and procedures $p_1^{(t)}, p_2^{(t)}, \ldots, p_{|p^{(t)}|}^{(t)}$, where $|d^{(t)}|, |m^{(t)}|, |p^{(t)}|$ respectively denote the number of diagnosis, medication, and procedure codes in $\mathcal{V}^{(t)}$. Some procedures produce lab results $r_1^{(t)}, r_2^{(t)}, \ldots, r_{|r^{(t)}|}^{(t)}$, which may be associated with continuous values (*e.g.* blood pressure) or binary values (*e.g.* positive/negative allergic reaction). Since we focus on a single encounter in this study, we omit the time index t throughout the paper.

If we assume all features d_i, m_i, p_i, r_i[1] can be represented in the same latent space, then we can view an encounter as a graph consisting of $|d| + |m| + |p| + |r|$ nodes with an adjacency matrix $\mathbf{A}$ that describes the connections between the nodes. We use c_i as the collective term to refer to any of d_i, m_i, p_i and r_i for the rest of the paper. Given c_i and $\mathbf{A}$, we can use graph networks or MiME[2] to derive the visit representation $\mathbf{v}$ and use it for downstream tasks such as heart failure prediction. However, if we do not have the structural information $\mathbf{A}$, which is the case in many EHR data and claims data, we typically use feed-forward networks to derive $\mathbf{v}$, which is essentially summing all node representations $\mathbf{c}_i$'s and projecting it to some latent space.

3.2. Transformer and Graph Networks

Even without the structure information $\mathbf{A}$, it is unreasonable to treat $\mathcal{V}$ as a bag of nodes c_i, because obviously physicians must have made some decisions when making diagnosis and ordering medications and procedures. The question is how

[1] If we bucketize the continuous values associated with r_i, we can treat r_i as a discrete feature like d_i, m_i, p_i.

[2] MiME is in fact, a special form of graph networks with residual connections.

to utilize the underlying structure without explicit $\mathbf{A}$. One way to view this problem is to assume that all nodes c_i in $\mathcal{V}$ are implicitly fully-connected, and try to figure out which connections are stronger than the other as depicted in Figure 2. In this work, we propose that Transformer is a suitable algorithm to learn the underlying encounter structure. To elaborate, we draw a comparison between two cases:

- Case A: We know $\mathbf{A}$, hence we can use a graph embedding algorithm, specifically Graph Isomorphism Network (GIN) (Xu et al., 2019) which can be expressed as

$$\mathbf{C}^{(j)} = \text{MLP}^{(j)}(\hat{\mathbf{D}}^{-1}\hat{\mathbf{A}}\mathbf{C}^{(j-1)}\mathbf{W}^{(j)}), \quad (1)$$

  where $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\hat{\mathbf{D}}$[3] is the diagonal node degree matrix of $\hat{\mathbf{A}}$, $\mathbf{C}^{(j)}$ is the node embeddings of the j-th convolution, and $\mathbf{W}^{(j)}$ is the trainable parameters of the j-th convolution. $\text{MLP}^{(j)}$ is a multi-layer perceptron of the j-th convolution with its own trainable parameters.

- Case B: We do not know $\mathbf{A}$, hence we use Transformer, specifically the encoder with a single-head attention, which can be formulated as

$$\mathbf{C}^{(j)} = \text{MLP}^{(j)}(\text{softmax}(\frac{\mathbf{Q}^{(j)}\mathbf{K}^{(j)\top}}{\sqrt{d}})\mathbf{V}^{(j)}), \quad (2)$$

  where $\mathbf{Q}^{(j)} = \mathbf{C}^{(j-1)}\mathbf{W}_Q^{(j)}$, $\mathbf{K}^{(j)} = \mathbf{C}^{(j-1)}\mathbf{W}_K^{(j)}$, $\mathbf{V}^{(j)} = \mathbf{C}^{(j-1)}\mathbf{W}_V^{(j)}$, and d is the column size of $\mathbf{W}_K^{(j)}$. $\mathbf{W}_Q^{(j)}, \mathbf{W}_K^{(j)}, \mathbf{W}_V^{(j)}$ are trainable parameters of the j-th Transformer block. Note that positional encoding using sine and cosine functions is not required, since features in an encounter are underordered.

Given Eq. 1 and Eq. 2, we can readily see that there is a correspondence between the normalized adjacency matrix $\hat{\mathbf{D}}^{-1}\hat{\mathbf{A}}$ and the attention map $\text{softmax}(\frac{\mathbf{Q}^{(j)}\mathbf{K}^{(j)T}}{\sqrt{d}})$, and between the node embeddings $\mathbf{C}^{(j-1)}\mathbf{W}^{(j)}$ and the value vectors $\mathbf{C}^{(j-1)}\mathbf{W}_V^{(j)}$. In fact, GIN can be seen as a special case of Transformer, where the attention mechanism is replaced with the known, fixed adjacency matrix. Conversely, Transformer can be seen as a graph embedding algorithm that assumes fully-connected nodes and learns the connection strengths during training. Given this connection, it seems natural to use Transformer as an algorithm to learn the underlying structure of visits. In our experiments, we use a variant of Transformer that is more suitable for handling EHR data. The details of the modification are described in the Appendix A.

[3]The original GIN does not use the normalizer $\hat{\mathbf{D}}^{-1}$ to improve model expressiveness on multi-set graphs, but we include $\hat{\mathbf{D}}^{-1}$ to make the comparison with Transformer clearer. Moreover, encounter records are not multi-set.

Table 1. Statistics of the MIMIC-III and eICU

	MIMIC-III	eICU
# of patients	30,556	166,355
# of visit	39,638	200,859
# of unique features	14,446	7,751
- # of diagnosis codes	6,007	3,827
- # of medication codes	3,804	1,285
- # of procedure codes	1,872	1,330
- # of bucketized lab results	2,763	1,309
Avg. # of diagnosis per visit	13.24	3.57
Avg. # of medication per visit	38.65	14.16
Avg. # of procedures per visit	5.60	2.89
Avg. # of lab results per visit	72.26	37.58

Note that Transformer's self-attention has been used in previous works for learning relations between features in settings other than text. Graph Attention Networks (Vaswani et al., 2017) applied attention on top of the adjacency matrix to learn non-static edge weights, and (Wang et al., 2018) used self-attention to capture non-local dependencies in images. Although our work also relies on attention, our interest lies in the connection between Transformer and graph networks, and whether Transformer can be an effective tool to capture the underlying graphical structure of EHR data even when the structural information is missing, thus improving encounter-based prediction tasks.

4. Experiments

4.1. Datasets

We conduct all of our experiments using two publicly available EHR datasets. MIMIC-III consists of ICU records collected at Beth Israel Deaconess Medical Center between 2001 and 2012, and eICU consists of ICU records collected from multiple sites in United States between 2014 and 2015. From the encounter records, medication/procedure orders and lab results, we extracted diagnosis codes, medication codes, procedure codes, and lab values. As mentioned in the introduction, both datasets do not contain structural information. For example, we know that *cough* and *acetaminophen* in Figure 1 occur in the same visit, but do not know if *acetaminophen* was prescribed due to *cough*. Table 1 summarizes the data statistics.

4.2. Baseline Models

- **shallow**: Each feature c_i in a visit $\mathcal{V}$ is converted to a latent representation $\mathbf{c}_i$ using multi-layer feedforward networks. Then the visit representation $\mathbf{v}$ is obtained by simply summing all $\mathbf{c}_i$'s. We use layer normalization

(Ba et al., 2016), drop-out (Srivastava et al., 2014) and residual connections (He et al., 2016) between layers.

- **deep**: We use multiple feedforward layers (including layer normalization, drop-out and residual connections) on top of **shallow** to increase the expressivity. Note that Zaheer et al. (2017) theoretically describes that this model is sufficient to obtain the optimal representation of a set of items (*i.e.,* a visit consisting of multiple features).

4.3. Prediction Tasks

In order to evaluate the model's capacity to leverage the implicit encounter structure, we use prediction tasks based on a single encounter, rather than a sequence of encounters, which was the experiment setup in Choi et al. (2018). Specifically, we test the models on the following tasks.

- **Mortality prediction**: Given an encounter record, we train models to learn the visit embedding $\mathbf{v}$ (*i.e.,* graph-level representation) to predict patient death during the ICU admission, *i.e.,* a binary prediction.
- **Medications prediction**: Given an encounter record with all medication codes removed, we train models to learn the visit representation $\mathbf{v}$ to predict all medication codes, *i.e.,* a multi-label prediction with the output dimension the size of the medication vocabulary.
- **Masked diagnosis code prediction**: Given an encounter record, we mask a random diagnosis code d_i. We train Transformer to learn the embedding of the masked code to predict its identity, *i.e.* a multi-class prediction. For **shallow** and **deep**, we use the visit embedding $\mathbf{v}$ as a proxy for the masked code representation.

Training details and hyperparameter settings are described in Appendix B.

4.4. Prediction Performance

Table 2. Mortality prediction performance

Dataset	Model	Validation AUCPR	Validation AUROC	Test AUCPR	Test AUORC
MIMIC-III	shallow	0.8564	0.9731	0.8138	0.9615
	deep	0.8590	0.9727	0.8172	0.9619
	Transformer	0.8934	0.9813	0.8685	0.9769
eICU	shallow	0.6677	0.9264	0.6728	0.9231
	deep	0.6747	0.9334	0.6638	0.9293
	Transformer	0.6847	0.9345	0.6879	0.9335

Table 2, Table 3, and Table 4 respectively show mortality prediction performance, medication prediction performance, and masked diagnosis code prediction performance for all models on two datasets.

Table 3. Medication prediction performance

Dataset	Model	Validation AUCPR	Validation AUROC	Test AUCPR	Test AUORC
MIMIC-III	shallow	0.6318	0.9856	0.6253	0.9856
	deep	0.6447	0.9869	0.6402	0.9867
	Transformer	0.6571	0.9875	0.6514	0.9874
eICU	shallow	0.1229	0.7922	0.1235	0.7910
	deep	0.1550	0.8614	0.1567	0.8626
	Transformer	0.2499	0.9397	0.2515	0.9408

Table 4. Masked diagnosis code prediction performance

Dataset	Model	Validation Accuracy	Test Accuracy
MIMIC-III	shallow	0.0608	0.0603
	deep	0.0587	0.0341
	Transformer	0.0869	0.0934
eICU	shallow	0.3488	0.3417
	deep	0.3549	0.3547
	Transformer	0.3647	0.3581

In both mortality prediction and medication prediction, all models show stronger performance on MIMIC-III than eICU, which is reasonable given that MIMIC-III's encounter records are more dense, *i.e.* there are more features per encounter on average as shown by Table 1. In masked code prediction, on the other hand, all models show stronger performance on eICU, naturally because eICU's diagnosis code vocabulary is smaller. Furthermore, as MIMIC-III's encounter has more diagnosis codes on average, it is more likely for the models to predict one of the visible diagnosis codes.

As all three tables show, Transformer outperforms baseline models on all three tasks on two different datasets. This empirical evidence strongly indicates Transformer's suitability to be used as a general-purpose EHR encounter modeling framework. Finally, Transformer significantly outperforms baseline models on some tasks and datasets, while the perforamnce improvement is not as dramatic for other tasks and datasets, which requires further investigation in the future. We discuss Transformer's attention behavior in Appendix C.

5. Conclusion

Learning effective patterns from raw EHR data is an essential step for improving the performance of many downstream prediction tasks. In this paper, we addressed that the previous state-of-the-art method required the complete encounter structure information, and proposed Transformer is a suitable method for learning the underlying structure when the structure information is unknown. The experiments demonstrated that Transformer outperformed various baseline models on all three tasks on two publicly available EHR datasets. In the future, we plan to apply Transformer on patient-level tasks such as heart failure diagnosis prediction or unplanned emergency admission prediction, while working on improving the attention mechanism to learn more medically meaningful patterns.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.

Ba, J., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. Relational inductive biases, deep learning, and graph networks. *arXiv:1806.01261*, 2018.

Che, Z., Kale, D., Li, W., Bahadori, M. T., and Liu, Y. Deep computational phenotyping. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 507–516. ACM, 2015.

Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., and Sun, J. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*, pp. 301–318, 2016a.

Choi, E., Bahadori, M. T., Searles, E., Coffey, C., Thompson, M., Bost, J., Tejedor-Sojo, J., and Sun, J. Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1495–1504. ACM, 2016b.

Choi, E., Bahadori, M. T., Sun, J., Kulas, J., Schuetz, A., and Stewart, W. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*, pp. 3504–3512, 2016c.

Choi, E., Xiao, C., Stewart, W., and Sun, J. Mime: Multi-level medical embedding of electronic health records for predictive healthcare. In *NeurIPS*, pp. 4552–4562, 2018.

Choi, Y., Chiu, C. Y.-I., and Sontag, D. Learning low-dimensional representations of medical concepts. *AMIA Summits on Translational Science Proceedings*, 2016:41, 2016d.

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

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Lipton, Z. C., Kale, D. C., Elkan, C., and Wetzel, R. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

Ma, F., Chitta, R., Zhou, J., You, Q., Sun, T., and Gao, J. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1903–1911. ACM, 2017.

Pollard, T. J., Johnson, A. E., Raffa, J. D., Celi, L. A., Mark, R. G., and Badawi, O. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5, 2018.

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., Liu, P. J., Liu, X., Marcus, J., Sun, M., et al. Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine*, 1(1):18, 2018.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NIPS*, pp. 5998–6008, 2017.

Wang, X., Girshick, R., Gupta, A., and He, K. Non-local neural networks. In *CVPR*, pp. 7794–7803, 2018.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *ICLR*, 2019.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. In *Advances in neural information processing systems*, pp. 3391–3401, 2017.

A. Modification to Transformer

Instead of using the original Transformer, we made the following changes to make it more suitable to handle EHR data.

- Our task is not sequence-to-sequence. Therefore we only use the encoding side of Transformer.
- As mentioned earlier, we do not use positional encoding, since features in a visit do not for a sequence.
- We used the same $\mathbf{W}_Q$ and $\mathbf{W}_K$ in all Transformer blocks. This is to regularize the optimization process such that Transformer's attention map (*i.e.* adjacency matrix in graph networks) does not differ too much in each block.
- Instead of concatenating the outputs from multiple attention heads, we use element-wise summation. Considering how graph networks aggregate a given node's neighbor information, summing the outputs of multiple attention heads corresponds to summing the node embeddings that are learned from slightly different adjacency matrices.
- Instead of using 6 Transformer blocks, we use 3 blocks, which is sufficient for the top visit node v to attend to the bottom lab values r_i.

B. Training Details

All experiments were conducted with two datasets, MIMIC-III and eICU. We divided the datasets into a training set, a validation set, and a test set in 8:1:1 ratio. All models were trained with Adam (Kingma & Ba, 2014) on the training set, and performance was evaluated against the validation set to select the final model. Final performance was evaluated against the test set. We used the minibatch of size 32, and trained all models for 300,000 iterations (*i.e.* minibatch updates). All models were implemented in TensorFlow 1.13 (Abadi et al., 2016), and trained with a system equipped Nvidia P100's.

Tunable hyperparameters for **shallow** and **deep** baseline model are as follows:

- Adam learning rate
- Drop-out rate between layers
- Feature embedding size

shallow used 6 feedforward layers and **deep** used 3 feedforward layers each before and after summing the embeddings. The number of layers were chosen to match the number of trainable parameters of Transformer.

Tunable hyperparameters for Transformer are as follows:

- Adam learning rate
- Drop-out rate between layers
- Number of attention heads
- Feature embedding size

Hyperparameters were searched via bayesian optimization with Gaussian Process for 48-hour wall clock time.

C. Attention Behavior

Using the Transformer model trained for masked code prediction on eICU, we chose a random encounter record from the test set to study the attention behavior. Figures 3, 4, and 5 respectively show the attention map from the first, second and the third Transformer block, given an encounter with 28 features including the visit node v. In the figures, features that start with 'D' are diagnosis codes, 'P' procedure codes, 'M' medication codes, and 'L' lab values. The attention maps are from the perspective of the diagnosis code *D_cardiovascular|chest pain, ashd|hyperlipidemia* (*i.e.* heart disease), which has the red background.

As can be seen from the figures, the heart disease diagnosis code is attending uniformly to other features in the first Transformer block, as the features haven't observed one another yet. However, in the second and the third Transformer block, the heart disease diagnosis code is selectively attending to relevant features. For example in Figure 4, the heart disease diagnosis code is almost exclusively connected to *P_pulmonary|ventilation and oxygenation|mechanical ventilation*, which is a procedure typically ordered for patients with respiratory/cardiovascular disorders.

It is noteworthy that even with the modification to the trainable parameters, Transformer learned to generate attention maps of different distributions in each block, which can be seen by the difference in Figure 4 and Figure 5. Further analysis on the attention behavior and improved modification of Transformer are required as future work in order to guide Transformer to learn medically meaningful edges.

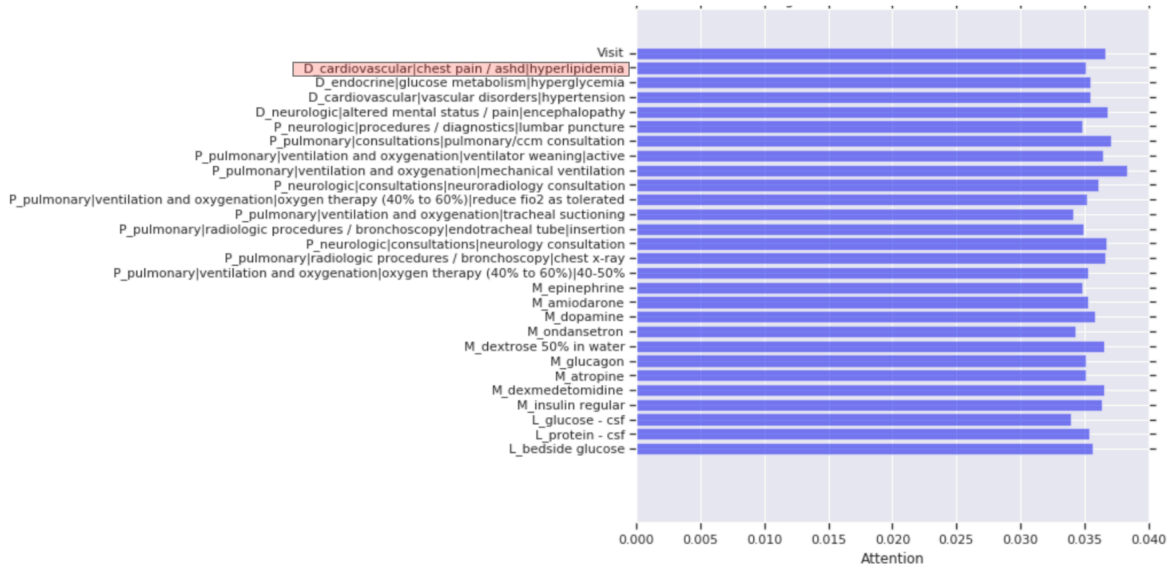

Figure 3. Attention map of the first Transformer block. Code starting with 'D' are diagnosis codes, 'P' procedure codes, 'M' medication codes, 'L' lab values. The diagnosis code with the red background is attending to the other features.

Figure 4. Attention map of the second Transformer block. Code starting with 'D' are diagnosis codes, 'P' procedure codes, 'M' medication codes, 'L' lab values. The diagnosis code with the red background is attending to the other features.

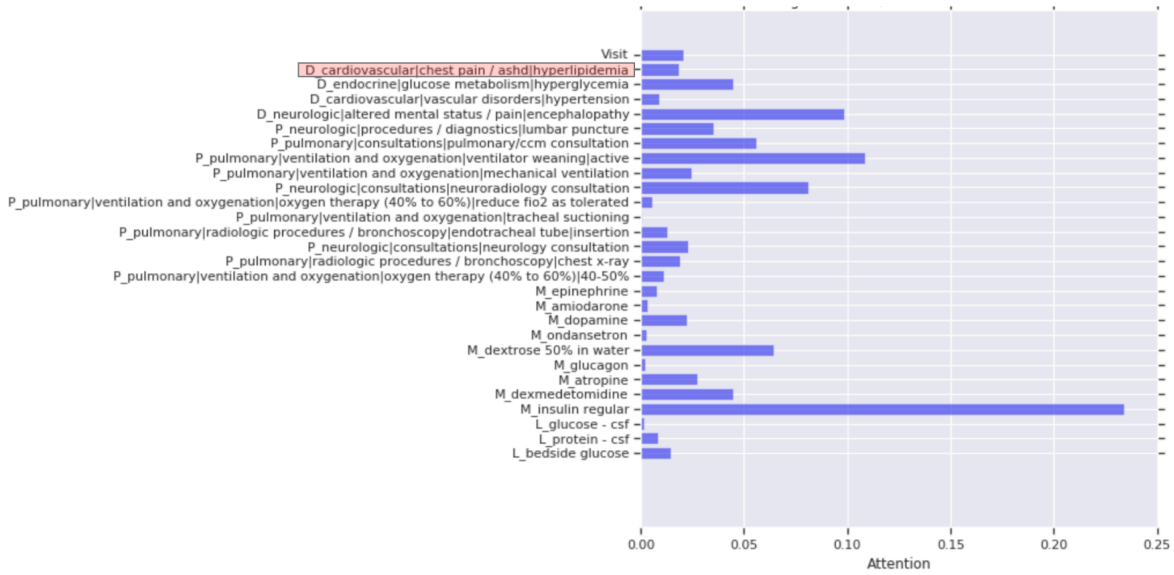

Figure 5. Attention map of the third Transformer block. Code starting with 'D' are diagnosis codes, 'P' procedure codes, 'M' medication codes, 'L' lab values. The diagnosis code with the red background is attending to the other features.