Reproducing Context-aware Health Event Prediction via Transition Functions on Dynamic Disease Graphs (Lu et al., 2022) CS598 DL4H Spring 2023

Shiyu (Sherry) Li and Wei-Lun (Will) Tsai
{shiyuli2, wltsai2}@illinois.edu

Group ID: 53
Paper ID: 28
Presentation link: https://youtu.be/K6ZzJMSb7as
Code links: https://github.com/willtsai/dlh-sp23-team53
Juptyer Notebooks (extra credit): chet.ipynb; cgl.ipynb

1 Introduction

In this report, we will focus on our reproduction study for *Context-aware Health Event Prediction via Transition Functions on Dynamic Disease Graphs* (Lu et al., 2022). This paper propose a new deep learning model called Chet (**c**ontext-aware **h**ealth **e**vent prediction via **t**ransition functions on dynamic disease graphs) that leverages the relationship between diseases and how they develop over time to predict future outcomes and diagnoses. Existing research on deep learning models for classification and prediction of diseases based on longitudinal EHR data have modeled disease diagnoses as independent events in their respective visits. However, intuition and data indicate that there are in fact hidden patterns within the combinations of disease diagnoses that may be useful for predicting future outcomes for patients, but yet have not been leveraged in existing best-in-class healthcare deep learning models. The Chet model is able to learn how diagnosed diseases develop over the course of each patient's doctor visits and then utilize this learned disease combination context to predict future outcomes and diagnoses. The most innovative part of the approach is the design to include both disease combinational information and the dynamic scheme of disease into the model. To include disease combinational information, the paper constructed a weighted disease combination based on the entire longitudinal EHR data globally and also a disease subgraph based on the specific visit locally. To include dynamic scheme of diseases, the paper utilized a disease-level temporal learning with multiple diagnosis roles and corresponding transition functions to extract historical contexts.

2 Scope of reproducibility

In our reproduction study, we will use the same methodology as proposed by the authors for data selection, cleaning, and preprocessing. Specifically, we will join the MIMIC-III (Johnson et al., 2023b) and MIMIC-IV (Johnson et al., 2023a) datasets along the same overlapping time ranges that the authors describe and split training/validation/test sets randomly using the same ratios they used. We will build the diagnosis graphs and calculate the adjacency matrices for their corresponding subgraphs using the same methodology described by the authors. We will train the model and at least one listed baseline model for diagnosis prediction and heart failure prediction respectively and compare the performance.

2.1 Addressed claims from the original paper

- By utilizing disease combinational information and the dynamic scheme of diseases, the Chet model has higher accuracy for diagnosis prediction and heart failure prediction than CGL (Lu et al., 2021), which is the most performant of the baseline models.

- The designed global disease graph and visit subgraphs can integrate global and local context from disease combinations to inform the deep learning model, so the Chet model has higher accuracy for diagnosis prediction and heart failure prediction than $Chet_{d-}$ model where dynamic part of GNN is removed in Chet.

- The Chet model has higher accuracy for diagnosis prediction and heart failure prediction than $Chet_{att-}$ model where attention from the final patient embedding layer is removed in Chet. This is an additional ablation study we proposed.

3 Methodology

In this section, we demonstrate the details of the model used in the original paper, our implementation approach as well as necessary computational resource.

3.1 Model descriptions

The Chet model can be decomposed into three layers: graph layer, transition layer and embedding layer.

3.1.1 Graph Layer

The first layer is a dynamic graph learning layer to extract both local and global contexts for diagnosis and neighbors in visit t using a memory-efficient calculation:

$$Z_D^t = m^t \odot (M + A(m^t \odot M) + A(n^t \odot N)) \quad (1)$$

$$Z_N^t = n^t \odot (N + A(n^t \odot N) + A(m^t \odot M)) \quad (2)$$

Where M,N represent embedding matrices for diagnoses and neighbors, A is the static adjacency matrix, m^t and n^t represent diagnoses code and neighbors code in t visit, Z_D^t is aggregated diagnosis local context and diagnosis global context and Z_N^t is aggregated neighbor global context. Finally, the GNN outputs are calculated with a fully connected layer with LeakyReLU as the activation function from Z_D^t and Z_N^t.

$$H_{D,N}^t = \text{LeakyReLU}(Z_{D,N}^t W) \in \mathcal{R}^{d \times s'} \quad (3)$$

3.1.2 Transition Layer

The Transition (second) Layer is to learn the disease development schemes, it takes the vector of diagnosis codes m^t per visit as input and partitions it into three disjoint vectors: (1) persistent diseases m_p^t representing diagnoses in visit t that are also present in visit $t-1$, (2) emerging neighbors m_{en}^t representing diagnoses in visit t that are neighbors in visit $t-1$, (3) emerging unrelated diseases m_{eu}^t representing diagnoses in visit t that are unrelated diseases in visit $t-1$. The layer is composed of three transition functions corresponding to each partition of m^t and are designed to extract historical context from previous visits to compute the hidden values. The transition function for calculating the hidden values for both m_{en}^t and m_{eu}^t is a scaled dot-product attention (Vaswani et al., 2017):

$$\text{Attn}(\text{Q, K, V}) = \text{soft max}\left(\frac{QW_q(KW_k)^T}{\sqrt{a}}\right) VW_v \quad (4)$$

Where a is the attention size, W_q, W_k, W_v are the weight matrices. For h_{en}^t, Q and K are the hidden neighbor embeddings H_N^{t-1}. For h_{eu}^t, Q and K are the universal embeddings of unrelated diseases R. For both h_{en}^t and h_{eu}^t, V is the diagnosis embeddings H_D^t. The transition function for calculating the hidden values for m_p^t is a modified gated recurrent unit (M-GRU) (Cho et al., 2014):

$$h_p^t = M\text{-}GRU(m_p^t \odot H_D^t, h_{en}^t, h_{eu}^t, h_p^{t-1}) \quad (5)$$

Finally, to calculate the visit embedding v^t, we apply max pooling to the transition output of the three partitions, which are all contained in h_p^t:

$$v^t = \mathtt{max\_pooling}(h_p^t). \quad (6)$$

3.1.3 Embedding Layer

The third layer is an embedding layer with a location-based attention to calculate the final hidden representation of all visits embeddings.

$$\alpha = \text{softmax}([v^1, v^2, \ldots, v^T] W_\alpha) \in \mathcal{R}^T \quad (7)$$

$$o = \alpha [v^1, v^2, \ldots, v^T]^T \in \mathcal{R}^p \quad (8)$$

Where W_α is a context vector for attention, α is the attention score for visits and o represents the final patient embedding.

3.2 Data descriptions

For our reproduction study, we will use the MIMIC-III (Johnson et al., 2023b) and MIMIC-IV (Johnson et al., 2023a) datasets downloaded from PhysioNet (Goldberger et al., 2000) for training/validation/testing, same as the original paper. In MIMIC-III data, there are 7493 patients in total from 2001 to 2012 with an average of 2.6 visits per patient and an average of 13.06 diagnose codes per visit. We randomly split the data into training set, validation set and test set with a size of 6000, 493 and 1000 respectively. In MIMIC-IV data, there are 10000 patients in total from 2013 to 2019 with an average of 3.79 visits per patient and an average of 13.51 diagnose codes per visit. We split the data into training set, validation set and test set with a size of 8000, 1000 and 1000 respectively.

3.3 Hyperparameters

The authors had randomly initialized model parameters and tuned on the validation dataset to arrive at the optimal values described in their paper. Thus, we set most of the hyperparameters to the same values as the original paper for our reproduction experiments: $batchsize = 32, hiddensize = 150, dropoutrate = 0.45$ (diagnosis prediction), and $dropoutrate = 0.0$ (heart failure prediction). The only hyperparameters we modified were the number of epochs and learning rate, the latter of which the authors had set as a step function change at specific epochs. Given that we ran just a fraction of the epochs (20 instead of 200), we had to adjust the learning rates accordingly:

- $LR_{diag} = 0.01$ for epochs 1 through 14, $1e-3$ for epochs 15 through 17, and $1e-5$ for epochs 18 through 20
- $LR_{hf} = 0.01$ for epochs 1 through 1, $1e-3$ for epochs 2 through 2, $1e-4$ for epochs 3 through 3, and $1e-$ 4 through 20

3.4 Implementation

In our reproduction implementation, we built Python notebooks chet.ipynb for reproducing the main Chet model and cgl.ipynb for reproducing the baseline CGL model. The notebooks contain complete model flows, including hyperparameters setting, data preprocessing, data loading, model building, model training and evaluation. We reused the author's code in data preprocessing. Our major coding efforts went into model rebuilding and the training/validation flow. In training/validation part, we built our own training and validation method to streamline the training, validation and test process while reusing the existing schedulers and metrics. In model rebuilding part, we tried to follow closely with the model structure and all the equations in the paper step by step.

3.5 Computational requirements

All code are implemented with Python and PyTorch. For additional package and version details, please refer to requirements.txt. Initially, we trained the model using our local machine with 16GB memory and Apple M1 PRO chip, and it took around 40 minutes for one epoch and around 14 hours in total for a combined MIMIC-III/MIMIC-IV training set for both diagnosis and heart failure prediction tasks. In order to unblock ourselves from the computational constraints we have with the CPUs on our local machine, we explored three different computational resources:

- We explored with Google Colab to utilize their free standard GPUs(NVIDIA T4 Tensor Core GPUs). This worked pretty well for us and it took around 8 minutes for one epoch and around 2 hours in total for a combined MIMIC-III/MIMIC-IV training set for both diagnosis and heart failure prediction tasks.
- We tried to explore with Microsoft Azure but we were not able to use GPU-optimized virtual machines from Microsoft Azure due to hardware availability limitations against our computing credits.
- We tried to use the available GPUs(16 cores) on one of our local machines. Although we were able to get the model training to run on our local Apple M1 Pro GPU, the training time actually grew along with unexplainably high computed losses at each epoch. We believe this is due to the fact that the M1 Pro GPU support on PyTorch may be buggy (e.g. this issue) and suspect that the the sequential throughout issues for RNN described in the PyTorch forum (Turner, 2022) affected our executions.

4 Results

We evaluated prediction performance for our models against the test datasets. Our reproduction experiment results are from training each of our reproduced Chet models once with 20 epochs instead of the 200 epochs originally used in the paper. Our reproduced Chet model performance directionally aligns with that of the original paper across both diagnosis and heart failure prediction tasks. We make the same observation as the original paper that MIMIC-IV trained models outperform MIMIC-III trained models in terms of AUC and F1 score. Our experiment results indicate that our work supports the claims from the original paper that Chet outperforms the most performant baseline model (CGL), with our results falling within the same order of magnitude as the original paper for Chet model performance improvement over CGL. Interestingly, the number of parameters in our reproduction of Chet model were about 0.9M and 0.3M smaller than the originals for diagnosis and

Models	MIMIC-III				MIMIC-IV			
	w-F1	**R@10**	**R@20**	**# Params**	**w-F1**	**R@10**	**R@20**	**# Params**
Orig CGL	21.92	26.64	36.72	1.53M	25.41	28.52	37.15	1.83M
Orig Chet	22.63	28.64	37.87	2.12M	26.35	30.28	38.69	2.59M
Repro CGL	20.69	24.91	34.69	1.51M	23.53	26.97	36.21	1.81M
Repro Chet	**21.45**	**26.56**	**36.33**	**1.22M**	**24.47**	**28.54**	**37.50**	**1.49MM**

Table 1: Diagnosis prediction results on MIMIC-III and MIMIC-IV using w-F1 (%) and R@k (%).

Models	MIMIC-III			MIMIC-IV		
	AUC	**F1**	**# Params**	**AUC**	**F1**	**# Params**
Orig CGL	84.19	71.77	0.55M	89.05	69.36	0.60M
Orig Chet	86.14	73.08	0.68M	90.83	71.14	0.88M
Repro CGL	82.33	69.08	0.53M	93.14	73.29	0.60M
Repro Chet	**85.09**	**71.99**	**0.47M**	**92.96**	**74.37**	**0.58M**

Table 2: Heart failure prediction results on MIMIC-III and MIMIC-IV using AUC (%) and F1 (%).

heart failure prediction tasks, respectfully, which suggests that our Chet model has reduced space complexity.

4.1 Diagnosis Prediction Results

Model performance results for the diagnosis prediction task from our preliminary experiments are summarized in Table 1. Compared to the 200-epoch Chet model performance results from the original paper, our reproduced 20-epoch Chet model performed only approximately 93-95% as well in terms of F1 score, R@10, and R@20; across both MIMIC-III and MIMIC-IV datasets. We observe similar patterns for the original vs. reproduced CGL model performance. Thus, our experiment results for the diagnosis prediction task confirm the original paper's claim that Chet outperforms CGL, and by the same order of magnitude.

4.2 Heart Failure Prediction Results

Model performance results for the heart failure prediction task from our preliminary experiments are summarized in Table 2. Compared to the 200-epoch Chet model performance results from the original paper, our reproduced 20-epoch Chet model performed about 99% as well as the original in terms of AUC and F1 score for the MIMIC-III dataset. Surprisingly, our Chet model outperformed the original Chet model by a magnitude of 2% in terms of AUC and 5% in terms of F1 score for the MIMIC-IV dataset. We observe similar patterns for the original vs. reproduced CGL model performance. We suspect that the higher prediction performance can be attributed to the fact that the heart failure prediction task is a much easier task than the diagnosis prediction task. Intuitively, predicting a general diagnosis is much more ambiguous and involves more complexity than predicting a single specific condition such as heart failure. Indeed, we observe in the original paper that the prediction performance for heart failure is much better than that of diagnosis prediction, across all baseline models and Chet, for both MIMIC-III and MIMIC-IV datasets. Our repro Chet models outperform our repro CGL models in all scenarios, with the lone exception of AUC for MIMIC-IV, where CGL and Chet have similar performance (CGL slightly outperforms by 0.19%). Thus, we conclude our experiment results for the heart failure prediction task confirm the original paper's claim that Chet outperforms CGL by the same order of magnitude as the original paper, with the exception of when measured by AUC on MIMIC-IV data. We suspect that this might be due to random noise or fluctuations in the model training runs, which the original paper normalizes by averaging over 5 training runs while we opted to train each model just once given limitations in compute capacity.

4.3 Ablation studies

To help us better understand the effectiveness of the design of each module in Chet, we also conducted two ablations studies with two variants of Chet:

- $Chet_{d-}$: This ablation study is also conducted in the original paper, and we followed the similar approach. We removed the dy-

Models	Diagnosis			Heart failure	
	w-F1	R@10	R@20	AUC	F1
$Chet_{att-}$	21.52	25.91	35.48	85.36	70.50
$Chet_{d-}$	19.82	25.05	34.72	85.37	69.64
Repro Chet	21.45	26.56	36.33	85.09	71.99
Original Chet	22.63	28.64	37.87	86.14	73.08

Table 3: Diagnosis prediction and heart failure prediction for Chet variants on the MIMIC-III dataset

namic part of GNN in Chet and instead of using dynamic subgraphs, we used a universal embedding matrix for all diseases and only used the global combination graph for the aggregation of diagnoses and neighbors.

- $Chet_{att-}$: This ablation study is a new one we proposed for us to better understand the effectiveness of attentions. In the final embedding layer of the model, we removed the location-based attention applied after max pooling.

Table 3 shows the result of diagnosis and heart failure prediction for original Chet, reproduced Chet, $Chet_{att-}$ and $Chet_{d-}$ on MIMIC-III dataset. We noticed that $Chet_{d-}$(removing the dynamic part of GNN) has significant drop on both F1 scores for diagnosis and heart failure predictions and recall for diagnosis predictions which validates the effectiveness of dynamic learning for the combination graph. However, we can see that $Chet_{d-}$ actually has a slightly higher AUC score than the reproduced Chet, there can be two main reasons: 1) The dynamic learning plays a more important role in diagnosis prediction than heart failure predictions. 2) The slightly higher AUC of $Chet_{d-}$ model is possibly an one-time error introduced because we only trained $Chet_{d-}$ model once with 20 epochs. In terms of $Chet_{att-}$ model, we can see the model has significantly lower recall for diagnosis prediction and significantly lower f1 for heart failure prediction compared to the reproduced Chet but the f1 score for diagnosis prediction and AUC for heart failure prediction are slightly higher than the reproduced Chet. And also $Chet_{att-}$ performs better than $Chet_{d-}$. From this result, we can also draw two conclusions: 1) Removing the final attention in the embedding layer retains the general structure of Chet so the effectiveness of attention is lower than the dynamic GNN. 2) The slightly higher AUC of $Chet_{att-}$ model is possibly an one-time error introduced because we only trained $Chet_{att-}$ model once with 20 epochs.

5 Discussion

From the result of the reproduction study, we are able to confirm the main claim that by utilizing disease combinational information and the dynamic scheme of diseases, the Chet model has higher accuracy for both diagnosis prediction and hearth failure prediction than CGL (Lu et al., 2021), which is the most performant of the baseline models. Moreover, we noticed that our reproduction study has fewer parameters than the original paper, and we suspect that in our implementation of the graph layer embeddings we ended up reducing the number of parameters in each model and thus reducing overall space complexity of the computations. In our additional ablation studies, we also learned that 1)The attention in the final embedding layer is less effective than the dynamic GNN part for model accuracy; 2) The dynamic learning plays a more important role in diagnosis prediction than heart failure predictions. However, due to time and computational constraints, we were not able to reproduce all baseline models in the paper and we used only 20 epochs for all our model training for one time which may have led to some errors.

5.1 What was easy

- MIMIC data is readily available so we have no issue downloading, accessing or using the data.

- The original codebase is well organized with clear instructions in readme file, so it is easy to run the author's code locally for our initial investigation and also easier to understand the code structure

- The CGL baseline model codebase is available and with clear instruction so it saves us a lot of efforts when reproducing the baseline model performance.

- In the original paper, the author already applied subgraphs' adjacency matrix calculation in graph layer so we don't need work with large and sparse matrix, which significantly improved computational efficiency for us.

- Google collab provides free GPU and has great integration with github which makes it easier for us to use and improves our model training speed by 5 times compared to initial CPUs on our local machine.

5.2 What was difficult

- Computational complexity is still the biggest problem that we have. Initially with limited computational resource, it was really expensive(time wise) to train the model. And it also took us significant time to try to explore other possible computational resources.

- The model implementation in the author's code has some difference from the original paper especially in the transition layer which brought some confusion to us.

- Baseline model (CGL) more difficult to reproduce than expected - needed to retrofit the model and data preprocessing to ensure we could train on common dataset. Specifically, we needed to produce an additional patient-code adjacency graph that was required for CGL.

5.3 Recommendations for reproducibility

- More descriptive names for variables in codebase to better align with the equations listed in the paper.

- Data preprocessing is quite complicated so more code documentation on the various functions used will be really helpful.

6 Communication with original authors

We communicated via email with one of the original authors, Chang Lu, who is also the owner of the original Chet and CGL experiment code repos. We learned through our correspondence that they had experimented with different LRs and observed where the F1 scores began to degrade in their 200-epoch runs and lowered the LR at those respective epochs. We conducted the same experiments to determine the optimal LR step function scheduler for our own 20-epoch training runs. Additionally, we confirmed with them the needed modifications we had to make on the preprocessing code for CGL to exclude clinical notes from the input data so that Chet and CGL can both be trained on the same cuts of MIMIC-IV data for fair comparison.

References

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. 2000. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220. Circulation Electronic Pages: http://circ.ahajournals.org/content/101/23/e215.full PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. 2023a. Mimic-iv.

Alistair Johnson, Tom Pollard, and Roger Mark. 2023b. Mimic-iii clinical database.

Chang Lu, Tian Han, and Yue Ning. 2022. Context-aware health event prediction via transition functions on dynamic disease graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4):4567–4574.

Chang Lu, Chandan K Reddy, Prithwish Chakraborty, Samantha Kleinberg, and Yue Ning. 2021. Collaborative graph learning with auxiliary text for temporal event prediction in healthcare. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3529–3535. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Philip Turner. 2022. Sequential throughput of gpu execution. `https://discuss.pytorch.org/t/sequential-throughput-of-gpu-execution/156303`. Accessed on May 5th, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008.