

FLARe: Forecasting by Learning Anticipated Representations

Surya Teja Devarakonda* SURYATEJADEV@CS.UMASS.EDU *Department of Computer Science
University of Massachusetts
Amherst, MA, USA*

Joie Yeahuay Wu* YEAHUAYWU@CS.UMASS.EDU *Department of Computer Science
University of Massachusetts
Amherst, MA, USA*

Yi Ren Fung YFUNG@CS.UMASS.EDU *Department of Computer Science
University of Massachusetts
Amherst, MA, USA*

Madalina Fiterau MFITERAU@CS.UMASS.EDU *Department of Computer Science
University of Massachusetts
Amherst, MA, USA*

Abstract

Computational models that forecast the progression of Alzheimer’s disease at the patient level are extremely useful tools for identifying high risk cohorts for early intervention and treatment planning. The state-of-the-art work in this area proposes models that forecast by using latent representations extracted from the longitudinal data across multiple modalities, including volumetric information extracted from medical scans and demographic info. These models incorporate the time horizon, which is the amount of time between the last recorded visit and the future visit, by directly concatenating a representation of it to the latent data representation. In this paper, we present a model which generates a sequence of latent representations of the patient status across the time horizon, providing more informative modeling of the temporal relationships between the patient’s history and future visits. Our proposed model outperforms the baseline in terms of forecasting accuracy and F1 score.

1. Introduction

Longitudinal medical datasets commonly contain biomarker information and medical scans spanning multiple years for thousands of patients. Recently, there has been an increased interest in building deep learning models, which have the benefit of being able to extract relevant features from datasets without expert knowledge, to forecast a patient’s disease progression. If effective, such models have the potential to become extremely useful tools for identifying high risk cohorts for early intervention and treatment planning.

Clinical Relevance With the increase in the world’s aging population, dementia is a critical threat to health around the world. Currently, it is estimated that 24 million people world wide suffer from the condition and this number is projected to double every 20 years (Reitz et al., 2011). Alzheimer’s disease is one of the most prevalent forms of dementia.

* Authors contributed equally

Diagnosis of Alzheimer’s takes into account many factors such as performance on cognitive screens, inheritance of high-risk biomarkers, and assessment of MRI scans (Neugroschl and Wang, 2011). Deep learning models that can leverage signals from these varying factors in order to accurately forecast the progression of the disease would greatly benefit doctors in identifying patients that are at a high risk of developing Alzheimer’s.

Technical Significance A wide range of models using deep learning to forecast longitudinal signals have been proposed in the literature. These approaches can be generalized as follows: they project the inputs collected from patient history into a common space using some fully-connected layers and then send the projected representations to a sequence learning model such as an RNN. The time horizon τ , which is the interval between the last observed status and the predicted status, is concatenated to the output of the RNN and sent to a classifier which assigns a disease stage label to the patient, τ time steps into the future.

These techniques do not take advantage of a temporal structure of a patient’s historical information. By simply concatenating τ to the output of the RNN, the models are not taking temporal information into account when forecasting the visit that occurs τ time steps into the future. In this case, τ is an additive term which signals the magnitude of the time horizon to the classifier without much of an effect on the overall prediction. Thus, the resulting model is partially agnostic with respect to the temporal correlations that occur across the time horizon. Additionally, with these methods, the irregular time intervals between consecutive visits are not incorporated into the model at any stage in the training process, resulting in models which operate under the assumption that the visits in all sequences are spaced uniformly in time.

To address these problems, we propose FLARe: Forecasting by Learning Anticipated Representations, a generative model which naturally incorporates τ into the prediction pipeline with the ability to "impute" representations of missing visits. FLARe draws inspiration from language modeling (LM), which is one of the most widely researched areas in natural language processing. One of the key challenges in LM is to come up with models that can generate sentences or paragraphs of any language. This is typically done by training a model to sequentially generate the next word, given a history of generated words. Such models are trained by optimizing the loss obtained by aggregating the difference between the predicted next word and the actual next word in each sentence in the training data. This approach operates under the assumption that every language has constraints on the ordering of words in a sentence. The model tries to learn what the most probable words are, given the present word or the history of words.

In FLARe, we follow a similar intuition by operating under the assumption that there are constraints on how much the 3D MRI scans and cognitive tests of a given patient can vary within 1 time point – which corresponds to 6 months. It follows from these assumptions that we should be able to train a model that can sequentially predict the feature vector of the patient in the next time step, given his/her feature vector in the present time step as well as the complete history until that point. Furthermore, we can do this by minimizing the loss obtained by aggregating the differences in the predicted and actual values of the next feature vector. As a byproduct of this sequence generation approach, our proposed model can also robustly handle patient trajectories which contain missing visits by 'imputing' their

learned representations. All of these factors together give FLARe a heightened ability to model the temporal relationship between a patient’s medical history and their future health status, resulting in better disease stage forecasting accuracy.

We tested our proposed model on the data of 1652 patients from the publicly available ADNI (Alzheimer’s Disease Neuroimaging Initiative) dataset using volumetric information extracted from MRI scans along with demographic information and cognitive test scores as input features. Our results show that our proposed model outperforms baseline models in terms of forecasting performance, while providing more balanced predictions across disease classes. We provide detailed analysis of the performance of our model over different time horizons τ and time steps T used for forecasting.

2. Related Work

Disease progression modeling has been an important topic in the field of healthcare analytics. Existing work in this area has been applied towards the development of early prevention and treatment methods. [Ito et al. \(2010\)](#) develop a progression model based on data from literature in order to measure longitudinal changes in cognitive test scores of Alzheimer’s patients. [De Winter et al. \(2006\)](#) propose a progression model for Type 2 Diabetes that aims to identify the effects of various treatments on diabetes-related biomarkers. There has also been much previous work in joint modeling, which is the sharing of parameters between two submodels: one for longitudinal outcomes and another for time-to-event outcomes [Hickey et al. \(2016\)](#) in order to jointly forecast the two. However, the aforementioned models require medical domain knowledge of the pathologies of the diseases being modeled and thus must be modified for use on different diseases.

Recently, there has been increased interest in disease progression modeling in the machine learning community. Such approaches attempt to model the trajectory of the disease using statistical and machine learning techniques on observational data acquired from medical records. A variety of methods have been deployed for the task including Markov Jump Models ([Wang et al., 2014](#)), Gaussian Processes ([Schulam and Arora, 2016](#)), and Functional Clustering ([Yao et al., 2005](#)), ([Halilaj et al., 2018](#)). Some approaches such as [Sukkar et al. \(2012\)](#) propose models which identify more fine grained trajectory types outside of the standard clinical stages and explore the correlations between multiple longitudinally collected measurements. [Zhou et al. \(2012\)](#) approach the problem by treating it as a multi-task regression, where the objective is to jointly model the trajectory of multiple longitudinal outcomes and biomarkers.

Even more recently, deep learning approaches to the problem have experienced increasing popularity due to their ability to learn features from the dataset without the need for domain knowledge. [Fiterau et al. \(2017\)](#) use hybrid CNN and LSTM layers which leverage information from structured covariates to predict the cartilage degeneration of patients with osteoarthritis, six years into the future. [Bhagwat et al. \(2018\)](#) use a Siamese network to learn a difference representation between two visits in a patient’s history in order to classify the patient’s trajectory into hierarchically clustered classes. Both [Choi et al. \(2016\)](#) and [Lim and van der Schaar \(2018\)](#) treat the disease progression as a slowly evolving point process and use temporal deep learning models such as RNNs to jointly model event occurrences and doctor diagnosis although the former directly predict the time-to-event occurrence and ICU-

9 codes of patients while the later predict the parameters of time-to-event and longitudinal measurement distributions.

3. Methodology

3.1. Model Description

Let $\mathbf{X}_{traj} = [\mathbf{x}_{v_1}, \mathbf{x}_{v_2}, \dots, \mathbf{x}_{v_L}] = \mathbf{x}_{v_1:v_L}$ denote the trajectory of a patient where $v_1 < v_2 < v_3 < \dots < v_L$ and \mathbf{x}_{v_t} are the input features from some patient at visit t . Note that $v_1, v_2, \dots, v_L \in \mathbb{N}$ and do not have to be consecutive. However, we can assume that they are without loss of generality.

Our goal is to predict the disease stage of the patient at visit $v_{L+\tau}$ where $\tau \in \mathbb{N}$.

We let $\mathbf{x}_{v_t} = \mathbf{i}_{v_t} \oplus \mathbf{s}_{v_t} \oplus \mathbf{c}_{v_t}$ where $\mathbf{i}_{v_t}, \mathbf{s}_{v_t}, \mathbf{c}_{v_t}$ denote volumetric information, demographic information, and cognitive test scores respectively, where \oplus is concatenation.

First, we use three separate multilayer perceptrons, ϕ_i, ϕ_s, ϕ_c , one for each category of input features to encode our input features into a common latent space. After we extract the representations, we concatenate them:

$$\phi_{i,s,c}(\mathbf{x}_{v_t}) = \phi_i(\mathbf{i}_{v_t}) \oplus \phi_s(\mathbf{s}_{v_t}) \oplus \phi_c(\mathbf{c}_{v_t}) = \mathbf{f}_{v_t}$$

We do this for all $\mathbf{x}_{v_t} \in \mathbf{X}_{traj}$ resulting in:

$$\phi(\mathbf{X}_{traj}) = \mathbf{f}_{v_1:v_L}$$

Then, the sequence $\mathbf{f}_{v_1:v_L}$ is sent to an RNN which provides hidden layer outputs for each input:

$$\mathbf{RNN}[\mathbf{f}_{v_1:v_L}] = \mathbf{h}_{v_1:v_L}$$

Each entry is used as input to another MLP, which we refer to as the feature prediction network, $\rho(\cdot)$. The purpose of $\rho(\cdot)$ is to take the hidden layer output \mathbf{h}_{v_L} and reconstruct the latent representation of $\mathbf{x}_{v_{L+1}}$ that is generated by $\phi(\cdot)$. More generally:

$$\rho(\mathbf{h}_{v_t}) = \hat{\mathbf{f}}_{v_{t+1}}, \quad t \in \mathbb{N}$$

$\hat{\mathbf{f}}_{v_{L+1}}$ is the reconstructed latent representation of a patient’s disease progression at visit $v_L + 1$. We create an auxiliary loss term $\mathcal{L}_{aux}(\mathbf{f}_{v_{L+1}}, \hat{\mathbf{f}}_{v_{L+1}})$, which is the mean squared error between the reconstructed representations and the learned representations of the visit data that we have available.

At this stage, we have dealt with all of the available visit data in the trajectory. If the trajectory only consisted of one visit, say $\mathbf{X}_{traj} = \mathbf{x}_{v_1}$, we would skip the RNN and take the representation \mathbf{f}_{v_1} and send it through ρ resulting in $\rho(\mathbf{f}_{v_1}) = \hat{\mathbf{f}}_{v_1+1}$. Then $classifier(\hat{\mathbf{f}}_{v_1+1})$ is the disease progression prediction of the patient at time $v_1 + 1$. Otherwise, in the general case where we have more than one visit we have this chain of events:

$$\hat{\mathbf{f}}_{v_{L+1}} \rightarrow \mathbf{RNN}[\cdot] \rightarrow \hat{\mathbf{h}}_{v_{L+1}} \rightarrow \rho(\cdot) \rightarrow \hat{\mathbf{f}}_{v_{L+2}} \rightarrow \dots$$

We continue iteratively generating the sequence of representations for the datapoints between our last available visit v_L and the visit we want to forecast $v_L + \tau$ until we reach

$\hat{h}_{v_L+\tau}$. Finally, we take $classifier(\hat{h}_{v_L+\tau})$ to be the disease stage forecast of the patient at time $v_L + \tau$. The disease stage of the patient can be placed in three categories: Cognitively Normal (CN), Mild Cognitive Impairment (MCI), or Alzheimer’s Disease (AD). If the trajectory is not continuous, i.e v_1, v_2, \dots, v_L are not consecutive, we can use ρ to "impute" any missing representation. During training, we backpropagate on both cross entropy loss, \mathcal{L}_{cel} and an auxiliary loss \mathcal{L}_{aux} . For this paper, we let the auxiliary loss to be the mean squared error between the forecasting feature vector and the ground truth feature vector although it would also be plausible to use the cross-entropy loss from labels predicted from the forecasted feature vectors. The objective function for a batch of N training samples is given in equation (1). The diagrams of FLARE and the baseline model are illustrated in Figure 1.

$$\mathcal{L} = \underbrace{\sum_{n=1}^N -y_n \log(\text{softmax}(\hat{y}_n))}_{\mathcal{L}_{cel}} + \alpha \underbrace{\sum_{n=1}^N \text{MSE}(\hat{\mathbf{f}}_n, \mathbf{f}_n)}_{\mathcal{L}_{aux}} \quad (1)$$

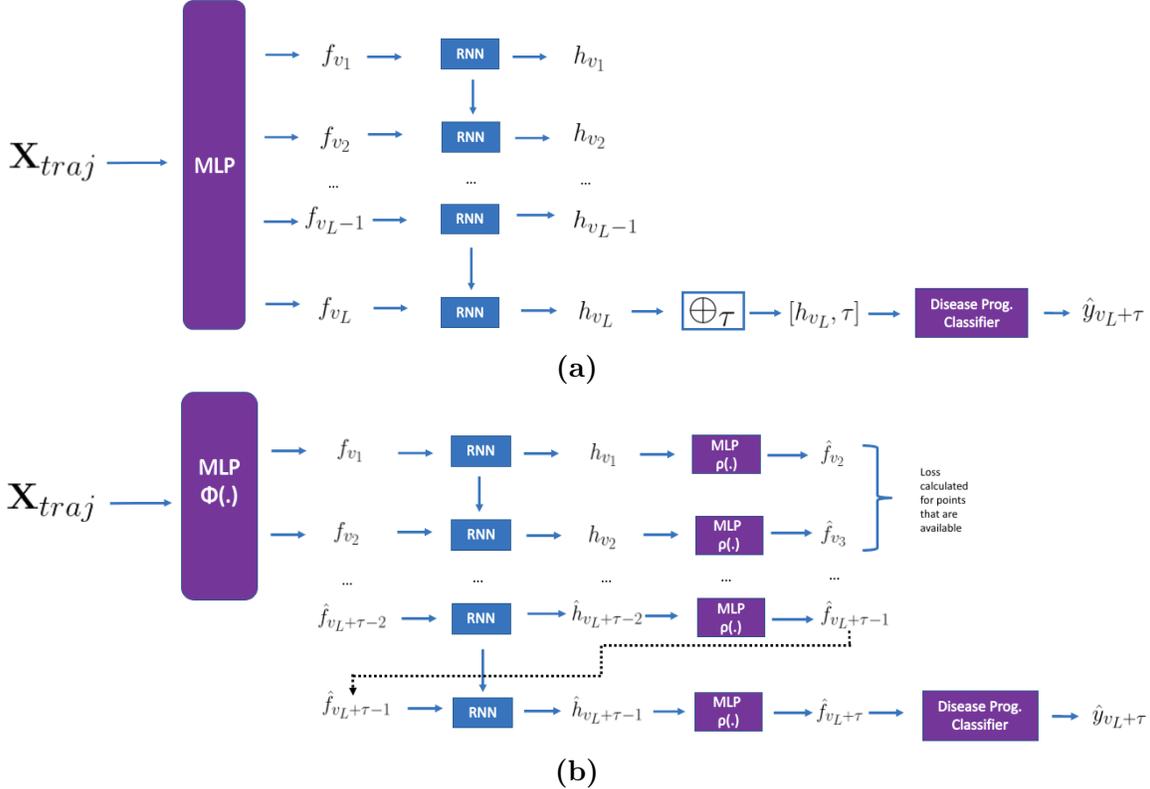


Figure 1: Diagrams illustrating the structure of the baseline (a) and FLARE (b)

4. Experiments

4.1. Dataset

We used the Alzheimer’s Disease Neuroimaging Initiative (ADNI) ¹ dataset for our experiments. The ADNI dataset contains clinical information and biomarkers taken from 2104 patients during visits that occur every 6 months spanning over five years. The dataset contains 1907 metrics recorded from the patient during each visit although it is common for many entries to be missing, and the patterns of missingness to be inconsistent across patients.

Subjects are given a diagnostic label of Cognitively Normal (CN), Mild Cognitive Impairment (MCI), and Alzheimer’s Disease (AD) during each visit. For more detailed information, the class breakdown and the number of disease stage transitions are available in Appendix ??.

In addition to undergoing a cranial MRI scan, patients have a variety of other metrics recorded during each visit including real-valued bio-markers that measure atrophy, molecular processes, and protein levels of a patient. These bio-markers include PET scans, Cerebral Spinal Fluid (CSF) measures, and volumetric measures extracted from MRI scans using FreeSurfer segmentation software. Additionally, patients take up to seven cognitive tests such as Alzheimer’s Disease Assessment Scale-Cognitive 13 Item Scale (ADAS-Cog13) and MMSE (Mini Mental State Examination) during each visit as a measure of their cognitive ability. Demographic information such as age, sex, and weight, along with risk factors such as presence of the APOE-4 gene are also recorded.

A list of patient IDs we used for training and testing will be made available in our code repository.

4.2. Experimental Setting

We shuffled the training data and performed an 80/20 split across 1652 patients. To address the class imbalance within our dataset, we used a weighted Cross Entropy Loss based on the class proportions. A list of hyperparameters we used for both models is available in Appendix C.

We chose 697 features from the ADNI dataset. 692 of them were volumetric measures taken from the segmentation of MRI scans, 4 were cognitive test scores, and 3 were demographic information. A detailed list of the features used will be made available in the code repository since the entire list is too long to put in this submission.

In order to create more training examples for our model, we generate multiple samples from a given patient trajectory \mathbf{X}_{traj} . We set two sampling parameters: T , which is the number of points used for prediction and τ , the time horizon. For example, let $\mathbf{X}_{traj} = [\mathbf{x}_{v_1}, \mathbf{x}_{v_2}, \mathbf{x}_{v_3}, \mathbf{x}_{v_4}]$. If we are sampling subtrajectories for $T = 2$ for $\tau = 1$, we would have: $[\mathbf{x}_{v_1}, \mathbf{x}_{v_2}]$ and $[\mathbf{x}_{v_2}, \mathbf{x}_{v_3}]$ where the disease stage of \mathbf{x}_{v_3} and \mathbf{x}_{v_4} would be the labels for each sampled trajectory respectively.

For the ADNI dataset, each unit of τ is 6 months. For each patient trajectory in a training batch, our model samples data for every possible value T and τ . We made the decision to avoid sampling for $T = 1$ since we wanted to use a sequence based model.

1. www.adni.loni.usc.edu

	Number of Patients	Augmented Trajectories
Training	1321	21520
Testing	331	5533

Table 1: Number of patients used for training and testing along with the number of trajectories augmented from them

Since the lengths of the trajectories ranged from 2 to 4 and thus did not exhibit enough range to be high variance, we decided to train our model on batches which contained trajectories of the same length, as opposed to using the common NLP technique of padding a batch of variable length sequences to be the same length. To select which trajectory length to load into a batch, we randomly sampled the trajectory length from a uniform distribution. We found that our results were sensitive to the method of loading the batches, but still resulted in models that supported the conclusion of our original experiments. Results for both dataloading approaches are available in the results section.²

Subsampling from our training data exposes our model to more trajectories of varying time horizons and points used for prediction. We split the patients into training and test sets before sampling so that samples generated for one patient will only be encountered in the training set. For our test set, we perform the same augmentation routine on the test sample. Table 1 illustrates the number of patients we used for training and testing, along with the total number of trajectories we sample from each set:

4.3. Architecture Selection

In order to determine the architecture for the initial MLP ϕ , we chose the architecture that performed the best on disease stage classification. This way, we could ensure that the architecture for ϕ is capable of extracting informative representations from the training data. Detailed descriptions of the model architectures will be made available in the code repository.

4.4. Candidate Models

We compare the performance of FLARe against the baseline model RNN-Concat, which is a reimplementaion of current forecasting models. It concatenates the time horizon τ to the hidden layer output of the RNN. We keep the model architectures the same between FLARe and RNN-Concat in order to control against the slight performance variations that may occur when implementing different architectures.

4.5. Results

Table 2 contains the classification accuracy, precision, recall, and F1-scores of the models we tested using the random batch loading approach. We found that FLARe outperforms RNN-Concat across all the evaluation metrics. Additionally, we observe that while the baseline has unbalanced performance across disease stages, often predicting MCI stage patients as

2. Link to code used to run experiments is available at <https://github.com/Information-Fusion-Lab-Umass/flare/tree/legacy>

	Accuracy	Precision	Recall	F1 score
RNN-Concat	.603	.607	.610	.609
FLARe	.659	.670	.660	.665

Table 2: Results for Alzheimer’s disease stage forecasting

AD or NC, our model provides more balance between the classes. Confusion matrices for RNN-Concat and FLARe are provided in Table 3 and Table 4 to illustrate the difference in class balance between models.

True \ Predicted	Predicted		
	NL	MCI	AD
NL	0.65	0.19	0.16
MCI	0.30	0.40	0.30
AD	0.14	0.08	0.78

Table 3: Average confusion matrix obtained across all possible values of (T, τ) using RNN-Concat

True \ Predicted	Predicted		
	NL	MCI	AD
NL	0.66	0.23	0.11
MCI	0.19	0.71	0.10
AD	0.13	0.26	0.61

Table 4: Average confusion matrix obtained across all possible values of (T, τ) using FLARe

To analyze our proposed model’s change in performance across different levels of data availability and forecasting horizons, we partition the testing set into buckets where each bucket corresponds to an ordered pair (T, τ) : the number of points used for prediction and the forecasting horizon. In Table 5 and Table 6, we provide the F1 score of RNN-Concat and FLARe for each bucket. Since our results vary with the batch loading approach used, we provide the table of results for two batch loading schemes we tried. In the first one, we randomly sample the sequence length from a uniform distribution and emit a batch of that sequence length. In Table 6, we keep a Dataloader for each sequence length and iterate through each of them during an epoch. We can make the following observations from the tables:

- FLARe is consistently on par or better than RNN-Concat across all buckets of the partitioned test set.
- The F1-scores of FLARe improve as the number of available visits increases. This is expected since the model has more data to make a prediction about the future visit representation.

	6 months	12 months	18 months
2 visits	Concat: .7334 FLARe: .7152	Concat: .5268 FLARe: .6272	Concat: .5468 FLARe: .6433
3 visits	Concat: .6183 FLARe: .7486	Concat: .5995 FLARe: .7442	N/A
4 visits	Concat: .7494 FLARe: .7957	N/A	N/A

Table 5: F1 scores of Alzheimer’s disease stage forecasting over each bucket of the partitioned test set using our random batch loading approach.

	6 months	12 months	18 months
2 visits	Concat: .5454 FLARe: .8399	Concat: .4978 FLARe: .8425	Concat: .3895 FLARe: .8208
3 visits	Concat: .7436 FLARe: .8674	Concat: .7542 FLARe: .8134	N/A
4 visits	Concat: .7172 FLARe: .8729	N/A	N/A

Table 6: F1 scores of Alzheimer’s disease stage forecasting over each bucket of the partitioned test set using pytorch’s built-in DataLoader class. Model parameters are determined by using the weights from the epoch with the lowest training loss. Notice how the inclusion of the built-in Dataloader increases FLARe’s performance by a huge margin across all buckets, while Concat’s performance increases in some buckets and decreases in others when compared to the results in Table 5.

5. Conclusion

In this paper, we present a novel approach called FLARe for disease trajectory forecasting using multimodal longitudinal data. FLARe uses a feature prediction module to anticipate the learned representations of the sequence of visits leading up to the future visit, given the representations of the history of visits, instead of directly concatenating some representation of the forecasting horizon τ to a latent representation of the medical history of the patient. The main reason why FLARe is so effective is because it has the capacity to model a more descriptive temporal relationship between a patient’s medical history and their future health status. Also, FLARe has an inherent robustness towards missing data, as it is trained to learn representations that can be used to impute the missing data points. Our performance analysis over the partitioned test set serves to illustrate this point. We observed that generally as the number of visits used for prediction increases, FLARe consistently has a better, or on par, F1 score across all time horizons when compared to the baseline model of RNN-Concat.

References

Nikhil Bhagwat, Joseph D Viviano, Aristotle N Voineskos, M Mallar Chakravarty, Alzheimer’s Disease Neuroimaging Initiative, et al. Modeling and prediction of clinical

- symptom trajectories in Alzheimer’s disease using longitudinal data. *PLoS computational biology*, 14(9):e1006376, 2018.
- Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor AI: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference 2016*, pages 301–318, 2016.
- Willem De Winter, Joost DeJongh, Teun Post, Bart Ploeger, Richard Urquhart, Ian Moules, David Eckland, and Meindert Danhof. A mechanism-based disease progression model for comparison of long-term effects of pioglitazone, metformin and gliclazide on disease processes underlying type 2 diabetes mellitus. *Journal of pharmacokinetics and pharmacodynamics*, 33(3):313–343, 2006.
- Madalina Fiterau, Suvrat Bhooshan, Jason Fries, Charles Bournhonesque, Jennifer Hicks, Eni Halilaj, Christopher Ré, and Scott Delp. Shortfuse: Biomedical time series representations in the presence of structured information. *Proceedings of 2nd Machine Learning for Healthcare Conference*, 2017.
- Eni Halilaj, Ya Le, Jennifer L Hicks, Trevor J Hastie, and Scott L Delp. Modeling and predicting osteoarthritis progression: data from the osteoarthritis initiative. *Osteoarthritis and cartilage*, 26(12):1643–1650, 2018.
- Graeme L Hickey, Pete Philipson, Andrea Jorgensen, and Ruwanthi Kolamunnage-Dona. Joint modelling of time-to-event and multivariate longitudinal outcomes: recent developments and issues. *BMC medical research methodology*, 16(1):117, 2016.
- Kaori Ito, Sima Ahadiéh, Brian Corrigan, Jonathan French, Terence Fullerton, Thomas Tensfeldt, Alzheimer’s Disease Working Group, et al. Disease progression meta-analysis model in Alzheimer’s disease. *Alzheimer’s & Dementia*, 6(1):39–53, 2010.
- Bryan Lim and Mihaela van der Schaar. Disease-atlas: Navigating disease trajectories with deep learning. *Proceedings of the 3rd Machine Learning for Healthcare Conference*, 2018.
- Judith Neugroschl and Sophia Wang. Alzheimer’s disease: diagnosis and treatment across the spectrum of disease severity. *Mount Sinai Journal of Medicine: A Journal of Translational and Personalized Medicine*, 78(4):596–612, 2011.
- Christiane Reitz, Carol Brayne, and Richard Mayeux. Epidemiology of alzheimer disease. *Nature Reviews Neurology*, 7(3):137, 2011.
- Peter Schulam and Raman Arora. Disease trajectory maps. In *Advances in Neural Information Processing Systems*, pages 4709–4717, 2016.
- Rafid Sukkar, Elyse Katz, Yanwei Zhang, David Raunig, and Bradley T Wyman. Disease progression modeling using hidden Markov models. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2845–2848. IEEE, 2012.

Xiang Wang, David Sontag, and Fei Wang. Unsupervised learning of disease progression models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 85–94. ACM, 2014.

Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association*, 100(470):577–590, 2005.

Jiayu Zhou, Jun Liu, Vaibhav A Narayan, and Jieping Ye. Modeling disease progression via fused sparse group lasso. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1095–1103. ACM, 2012.

Appendix A: Transitions and Disease Stage Counts in Experiment Data

Table 7: Disease Stage Counts

	Number of Patients
Cognitively Normal (CN)	805
Mild Cognitive Impairment (MCI)	536
Alzheimer’s Disease (AD)	317

Table 8: Transition Counts

	Number of Transitions
CN to MCI	66
MCI to AD	318
CN to AD	2

Appendix B: Number of Trajectories Used Per Partition

Table 9: Breakdown of number of samples in each partition of the training set

	6 months	12 months	18 months	24 months
2 visits	2954	2079	1294	N/A
3 visits	1982	1227	N/A	N/A
4 visits	1165	N/A	N/A	N/A

Table 10: Breakdown of number of samples in each partition of the testing set

	6 months	12 months	18 months	24 months
2 visits	762	536	332	N/A
3 visits	514	317	N/A	N/A
4 visits	301	N/A	N/A	N/A

Appendix C: Hyperparameters

	Class Weights	Learning Rate	Optimizer	Epochs Total
Hyperparameters	[CN:1,MCI:1.3,AD:2]	.001	ADAM	10000