Deep Contextual Clinical Prediction with Reverse Distillation

Rohan Kodialam,[1] **Rebecca Boiarsky,**[1] **Justin Lim,**[1]
Aditya Sai,[2] **Neil Dixit,**[2] **David Sontag**[1]
[1]MIT CSAIL & IMES
[2]Independence Blue Cross
kodialam@alum.mit.edu, rboiar@mit.edu, justinl@mit.edu,
Aditya.Sai@ibx.com, Neil.Dixit@ibx.com, dsontag@csail.mit.edu

Abstract

Healthcare providers are increasingly using machine learning to predict patient outcomes to make meaningful interventions. However, despite innovations in this area, deep learning models often struggle to match performance of shallow linear models in predicting these outcomes, making it difficult to leverage such techniques in practice. In this work, motivated by the task of clinical prediction from insurance claims, we present a new technique called *reverse distillation* which pretrains deep models by using high-performing linear models for initialization. We make use of the longitudinal structure of insurance claims datasets to develop Self Attention with Reverse Distillation, or SARD, an architecture that utilizes a combination of contextual embedding, temporal embedding and self-attention mechanisms and most critically is trained via reverse distillation. SARD outperforms state-of-the-art methods on multiple clinical prediction outcomes, with ablation studies revealing that reverse distillation is a primary driver of these improvements. Code is available at https://github.com/clinicalml/omop-learn.

Introduction

Machine learning of predictive models on health data is widely used to guide preventative, prophylactic and palliative care. We focus on a subset of electronic medical records that are frequently found as part of health insurance claims or as administrative data in large hospital systems. For each patient, we receive a time series of *visits* – single continuous interactions of a patient with the healthcare system – and *codes* – the medical events occurring during each visit. These codes detail the specialties of visited doctors, diagnoses, procedures, the administration of drugs, and other medical concepts.

Several aspects of these claims data make the machine learning challenge unique from other settings where sequential data is observed (e.g., natural language processing). First, the data is extremely sparse. Second, multiple observations are recorded during a single visit (e.g., diagnoses, procedures, medications) and the vocabulary of medical concepts is often in the tens of thousands. Third, visits correspond to highly irregularly-spaced time series of events, since care is often administered in short bursts punctuated by long gaps. Variable timescales must be simultaneously accounted for, since the time between visits made by a single patient can vary from years to days.

Deep learning suggests a path to improving predictive performance by learning representations of longitudinal health records that capture a patient's medical status and potential future risks. State-of-the-art models in the literature have largely focused on shorter-term prediction over horizons of days or weeks, most notably during a single hospital visit or in the immediate aftermath of a visit (Choi et al. 2017). Approaches to longer-term prediction often rely on manually feature-engineering longitudinal health data into patient state vectors (Razavian et al. 2015; Ahmad et al. 2018; Avati et al. 2018; Miotto et al. 2016), as opposed to training end-to-end from raw longitudinal EHR data. Due to this heuristic approach, these methods cannot fully exploit the temporal nature of EHR data, nor the relationships between clinical concepts.

We introduce Self Attention with Reverse Distillation, or SARD, a self-attention based architecture for longitudinal health data, which uses a self-attention mechanism (Vaswani et al. 2017) to extract meaning from the temporal structure of medical claims and the relationships between clinical concepts. Our architecture is inspired by BEHRT (Li et al. 2020), which recently outperformed previous deep learning algorithms for medical records including RETAIN (Choi et al. 2016b) and Deepr (Nguyen et al. 2016). Building off of BEHRT, our major contribution is our novel pre-training procedure, reverse distillation (RD); our architecture also differs in several other key aspects.

In reverse distillation, we first initialize our model to mimic a performant linear model, and subsequently finetune. We find empirical evidence that reverse distillation acts as an effective way to perform soft feature selection over complex feature spaces, such as multidimensional time-series data. We further establish statistically significant gains against strong baselines in terms of predictive performance for three long-term tasks – predicting the likelihood of a patient dying, requiring surgery, and requiring hospitalization – with clear applications to preventative and palliative healthcare. Our experiments also establish that reverse distillation is a key driver behind these wins, and pave the way for the use of this method in future research.

In summary, we present the following contributions:

- SARD, a transformer architecture which uses an explicit visit representation to better encode claims data. SARD also uses a convolutional prediction head to ingest the outputs of its transformer layers, in contrast to the linear heads used in previous work.
- Reverse distillation, a novel and broadly applicable method of initializing machine learning models using high-performing linear models.
- An introspection analysis of how reverse distillation allows SARD, and deep models in general, to generalize better and make more accurate predictions by effectively regularizing deep models to make good use of features known to be clinically meaningful.

Related Work

Many recent works analyze how deep learning can be applied to clinical prediction (Choi et al. 2016a; Rajkomar et al. 2018; Che et al. 2018; Steinberg et al. 2020; Choi et al. 2016b; Harutyunyan et al. 2019; Gao et al. 2020; Ma et al. 2018; Zhang et al. 2019). Several approaches use recurrent neural networks (RNNs) to ingest medical records, and achieve excellent performance on tasks like predicting in-patient mortality upon hospital admission (Choi et al. 2016a). Further refinements add learned imputation to account for missingness (Che et al. 2018), and improvements in featurizing time by using architectures like bi-directional RNNs (Ma et al. 2017), explicit temporal embeddings (Baytas et al. 2017) and two-level attention mechanisms to find the influence of past visits on a prediction (Choi et al. 2016b; Kwon et al. 2018). Research has also focused on using convolutional neural networks (CNNs) to develop better embeddings of clinical concepts passed into a recurrent model (Ma et al. 2018), and graphically representing the patient-clinician relationship to augment health record data (Zhang et al. 2019). Self-attention has also been used to develop relationships between medical features that have already been collapsed over the temporal dimension using recurrent methods (Ma et al. 2020) and to phenotype patients (Song et al. 2017). More recently, self-attention was used in BERT for EHR, or BEHRT (Li et al. 2020), to simultaneously predict the likelihood of 301 conditions in future patient visits.

When making predictions with horizons of months or years, the state-of-the-art is often still simple, linear models with carefully chosen features (Bellamy, Celi, and Beam 2020; Razavian et al. 2015; Ahmad et al. 2018). Recent work exploring deep-learning based approaches to long-term clinical prediction train neural networks directly on features constructed using hand-picked time windows and summary statistics (Avati et al. 2018) or use denoising autoencoders to pre-process this type of data (Miotto et al. 2016), and do not necessarily beat strong linear baselines (Rajkomar et al. 2018, Supplemental Table 1). Critically, many of these models rely on manual feature-engineering to create representations of the time-series data that forms a patient's medical record rather than learning this structure in tandem with the task at hand.

SARD Model Architecture

Our model builds upon self-attention architectures (Vaswani et al. 2017), most recently applied in the clinical domain by the BEHRT model. SARD differs from BEHRT in several important ways. Firstly, SARD operates on visit embeddings which summarize a patient's medical events in that visit in a single input, while BEHRT encodes each diagnosis separately in a sequence, using separators to indicate the boundaries of each visit. This allows SARD to include significantly more data from a patient's history with the same computational efficiency. Secondly, SARD uses a convolutional prediction head applied to all transformed visit embeddings, while BEHRT uses dense layers applied to a single transformer output. Furthermore, BEHRT was demonstrated on a feature dimension of 301 condition codes, which did not include medications and procedures; in this paper, we apply SARD on a much larger set of 37,004 codes, spanning conditions, medications, procedures, and physician specialty.

We use a set encoding approach to address the challenge of sparsity and the need to represent a set of data observed at each visit, and a self-attention based architecture to allow any visit's embedding to interact with another visit embedding through $O(1)$ layers, thus ensuring that we can capture temporal information and dependencies. An overview of the architecture is provided in Figure 1.

We denote the set of visits made by a patient i by $\mathcal{V}_i$, and represent this patient's jth visit by V_j^i. We further denote the time of visit V_j^i by t_j^i and the set of codes assigned during visit V_j^i with $C_j^i \subseteq \mathcal{C}$.

Input Embedding: We adapt the method of Choi, Chiu, and Sontag (2016) to generate an initial concept embedding map $\phi : \mathcal{C} \to \mathbb{R}^{d_e}$, learned only using data in the training window to prevent label leakage. The vector representation $\psi(V_j^i) \in \mathbb{R}^{d_e}$ of each visit is calculated as $\psi(V_j^i) = \sum_{c \in C_j^i} \phi(c)$, providing invariance to permutations of the codes. This is similar to the Deep Sets paradigm, with nonlinearity provided by the embedding ϕ and downstream components of our architecture (Zaheer et al. 2017).

Temporal Embedding: SARD does not explicitly encode the order of events, and visits do not occur in regular intervals. We embed the time of each visit into $\mathbb{R}^{d_e}$ using sinusoidal embeddings (Vaswani et al. 2017), and generate a *temporal embedding* $\tau(V_j^i) = \sin(\tilde{t}_j^i \omega) || \cos(\tilde{t}_j^i \omega)$, where $\tilde{t}_j^i = \min(365, T_A - t_j^i)$ and T_A represents the prediction date. This allows us to measure time relative to the prediction date. We found that clipping these relative time differences at one year increased performance – this design choice effectively groups together all longer-term dependencies. Note that we denote concatenation with $||$, ω is a length $d_e/2$ vector of frequencies in geometric progression from 10^{-5} to 1, and $\sin$ and $\cos$ are applied element-wise.

Self-Attention: We add $\psi(V_j^i)$ and $\tau(V_j^i)$ to create final encodings that represent the content and timing of visits. To contextualize visits in a patient's overall history we use multi-headed self-attention (Vaswani et al. 2017) with $L = 2$ self-attention blocks and $H = 2$ heads. For efficiency, we truncate to the $n_v = 512$ most recent visits, and

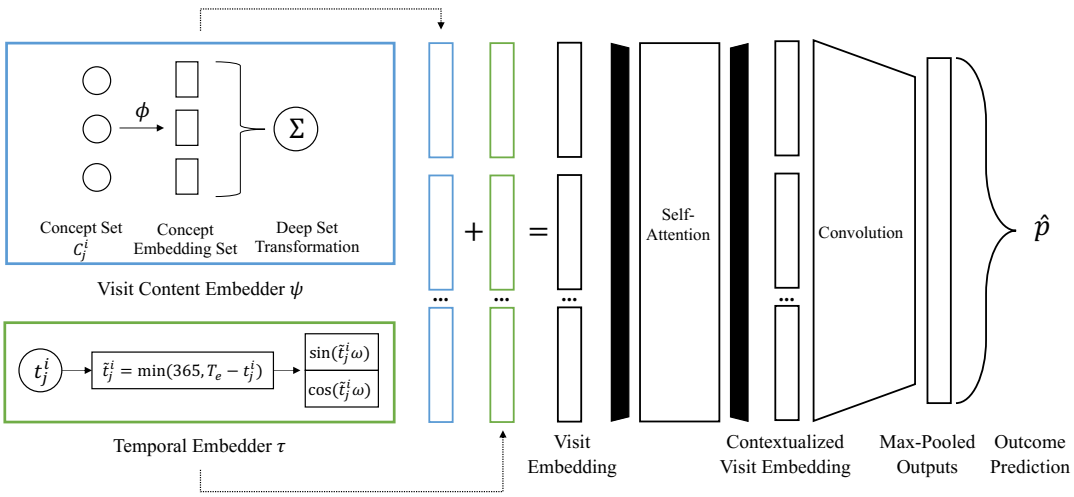

Figure 1: SARD Architecture for Longitudinal Claims Data

add padding for patients with less than n_v visits, but use a masking mechanism to only allow non-pad visits to attend to each other. We apply dropout with probability $\rho_d^t = 0.05$ after each self-attention block to prevent overfitting. This approach allows any visit to attend to any other, so longer-range dependencies of clinical interest can be learned.

Each layer of each head performs three affine transformations on the input embeddings, which for the first layer are $\psi(V_j^i) + \tau(V_j^i)$ for each visit V_j^i. These transformations produce vectors k_j^i, q_j^i and v_j^i respectively. We find the contextualized embedding of visit V_j^i by computing raw attention weights $w_{j\ell}^i = q_j^i \cdot k_\ell^i / \sqrt{d_e}$, normalizing via softmax to $\tilde{w}_{j\ell}^i = \left(\sum_{r=1}^{n_v} e^{w_{jr}^i}\right)^{-1} e^{w_{j\ell}^i}$, and taking the weighted sum $\sum_{\ell=1}^{n_v} \tilde{w}_{j\ell}^i v_\ell^i$. This process is then repeated at each layer using the contextualized embeddings as inputs, and residual connections are used between layers. The outputs of each head are concatenated to create final, contextualized visit representations $\tilde{\psi}(V_j^i)$.

Convolutional Prediction Head: The prediction head returns an estimated probability of the target event using the outputs of the self-attention mechanism. We do so by creating K convolutional kernels of size $d_e \times 1$. Then, each kernel extracts a feature from the non-pad contextualized visit embeddings by first calculating a cross-correlation versus each $\tilde{\psi}(V_j^i)$, then using a max-pooling operation to select the highest of these cross-correlations. Concatenating these outputs gives a length-K real vector of extracted features. To obtain a predicted probability $\hat{p}(i)$ for each patient, we apply a sigmoid nonlinearity to this vector, take the dot product of the transformed components with a learned vector of weights, and apply another sigmoid nonlinearity to obtain a final prediction probability.

Learning with Reverse Distillation

Reverse distillation is a novel method by which we initialize a deep model using a linear proxy. We consider a binary prediction model $f_\theta : \mathcal{X} \to [0,1]$ parametrized by θ which maps from a domain $\mathcal{X}$ of data to a probability value, and a linear model $g_w : \mathcal{X} \to [0,1]$ defined by $g_w(x) = \sigma(w^T \xi(x))$, where σ is the sigmoid function and ξ is a fixed *feature engineering* transformation $\xi : \mathcal{X} \to \mathbb{R}^d$ based on heuristic domain knowledge.

While f_θ may be a large, highly-parametrized model, g_w may perform better on prediction tasks for several reasons, including the ability to select features and avoid overfitting through regularization of w, and the quality of the transformation ξ. As such, we initialize f_θ to mimic the outputs of g_w in order to benefit from the structure and performance of the linear model while allowing for further data-driven improvements.

We interpret predictions $f_\theta(x)$ (resp $g_w(x)$) as indicating that the distribution of the label for data point x is $\mathbf{B}(f_\theta(x))$ (resp $\mathbf{B}(g_w(x))$), where $\mathbf{B}(p)$ indicates a Bernoulli distribution with success parameter p. We perform reverse distillation by pre-training our deep model to optimize over θ a loss function defined by

$$\begin{aligned} \ell_{\text{RD}}(x) &= -p_c g_w(x) \log f_\theta(x) \\ &\quad -(1 - g_w(x)) \log(1 - f_\theta(x)). \end{aligned} \tag{1}$$

This algorithm is inspired by the standard knowledge distillation paradigm (Hinton, Vinyals, and Dean 2015), in which a simpler model is trained to mimic a complex model. To fine-tune f_θ, we make use of both the true label $y(x) \in \{0,1\}$ and the prediction $g_w(x)$, combining a cross-entropy loss versus the true label

$$\ell_{\text{CE}}(x) = -p_c y(x) \log f_\theta(x) \tag{2}$$
$$-(1 - y(x)) \log(1 - f_\theta(x)) \tag{3}$$

and the reverse distillation loss ℓ_{RD}, to get a loss function

$$\ell_{\text{tune}}(x) = \ell_{CE}(x) + \alpha \ell_{RD}(x). \tag{4}$$

We include a class weighting term p_c equal to the ratio between the number of negative and positive training data

points to encourage higher recall in our trained model, and a hyperparameter α to represent the weight placed on differences between $g_w(x)$ and $f_\theta(x)$. We note that cross-validation over α always selected 0 in our experiments, meaning that reverse distillation was only needed for initializing the model.

Training Procedure for SARD. We next describe our procedure for training a SARD model with reverse distillation. All training is performed end-to-end, including the initial embedding ϕ of clinical concepts. We reverse distill from a highly L_1-regularized logistic regression model. As the logistic regression's predictions tend to be well-calibrated (Niculescu-Mizil and Caruana 2005), we interpret its output as a distribution over outcomes. While hand-engineered features are often created for specific tasks in the clinical domain, we opt for a more general formulation. Inspired by prior work in high-performance linear models for clinical prediction (Razavian et al. 2015), we construct features by aggregating codes over different temporal windows, and thus we refer to this model as a *windowed* linear model. Given a time interval $W = [t_s, t_e]$, we find the feature vector corresponding to this interval for patient i by finding the subset of visits $\mathcal{V}_i(W) = \{V_j^i \in \mathcal{V}_i | t_j^i \in W\}$ and subsequently finding the set of codes $\mathcal{C}_i(W) = \bigcup_{V_j^i \in \mathcal{V}_i(W)} C_j^i$. We find that performance was optimized by using a multi-hot vector $f_i(W)$ of size $|\mathcal{C}|$ as the feature engineering transformation ψ to map these sets of codes to real-valued vectors, with the element corresponding to concept $c \in \mathcal{C}$ set equal to 1 if $c \in \mathcal{C}_i(W)$ and 0 otherwise.

To capture the longitudinal nature of claims data, we use multiple windows simultaneously as features. We establish a list $\mathcal{W}_C$ of candidate windows, each of which has an end time equal to the prediction date and start times ranging from 15 to ∞ days before the prediction date, as shown in Appendix Table 5. We selected the $n_W = 5$ best windows from all $\binom{|\mathcal{W}_C|}{n_W}$ unique window choices by comparing validation performances.

Theoretical Analysis

We note that a deep model and a linear model making the same classifications are not necessarily learning the same classification boundary. We investigate if the self-attention model actually replicates the linear model's classification function.

We find that it is possible to construct a set of weights such that SARD and a windowed logistic regression model have identical outputs for all inputs:

Lemma 1. *In the limit $d_e \to \infty, K \to \infty$ and for an appropriate choice of ω, SARD can identically replicate a windowed linear model.*

The proof can be found in the Appendix. The crux of the argument is that we can express a filter of the form $[[t_j^i < T]]$ for any T as a linear combination of the elements $\tau(V_j^i) = sin\left(t_j^i \omega\right) || cos\left(t_j^i \omega\right)$, with weights determined as Fourier series coefficients. This allows SARD to replicate the windowed feature vectors of the linear model. We note that this lemma holds even with a single self-attention layer.

This result increases our confidence in our choice of architecture and its ability to generalize and improve beyond a linear model. For example, windows of the form $[[t_j^i < T]]$ implied by the linear model might be inferior to a more complex filter in the time domain. However, such filters can be learned by SARD. While the existence of this set of weights does not mean that SARD will converge to these exact weights after reverse distillation, it does highlight one possible mechanism for ensuring that the deep and linear models generalize in the same way.

Interpretability via Network Dissection

We next introduce a technique to investigate whether and how reverse distillation surfaces features of the windowed linear baseline. We utilize the Network Dissection global interpretability framework of Bau et al. (2018) to compare the outputs of the penultimate layer of SARD networks to the linear baseline's features. Our goal is to match the latent features which are inputted to the final prediction head in the deep model to the interpretable features of the linear model, as a means of both understanding which linear features are preserved using reverse distillation, as well as to aid in interpreting the deep model features which are ultimately used in prediction. To do this "matching," we binarize the penultimate layer of the deep model by taking the sign of each output, and then calculate the Matthew's Correlation Coefficient (MCC) of each output with each windowed linear baseline feature, across all people in the test set.

Experiments

We evaluate our approach using a de-identified dataset of $121,593$ Medicare Advantage patients provided by a large health insurer in the United States. This data is mapped into the Observational Medical Outcomes Partnership (OMOP) common data model (CDM) version 6 (Hripcsak et al. 2015). OMOP provides a normalized concept vocabulary, and although our dataset is not public, hundreds of health institutions with data in an OMOP CDM can use our code out-of-the-box to reproduce results on local datasets[1]. We also investigate the properties of reverse distillation through experimentation on synthetic data.

Baselines. We compare to several baselines. First, we compare to the windowed L_1-regularized logistic regression model (Razavian et al. 2015) described earlier in the context of reverse distillation. Second, we compare to two of the previous state-of-the-art deep learning models for similar tasks: RETAIN (Choi et al. 2016b; Kwon et al. 2018), a recurrent architecture with attention, and BEHRT (Li et al. 2020), the transformer-based architecture which served as the jumping off point for our model. Third, we compare to our own self-attention-based model trained without reverse distillation.

To build a BEHRT model in our data setting, we use a self-attention architecture to ingest sequences of medical codes (as in the original BEHRT model) instead of aggregated sequences of entire visits (as in SARD). This model is very similar to BEHRT, with some minor differences. Specifically, we omit the use of SEP tokens and age embeddings.

[1] https://github.com/clinicalml/omop-learn

Due to the computational constraints imposed on both the SARD and BEHRT models, it was generally not possible to include significantly more than one year of data for a given patient, rendering a per-code age embedding superfluous. For the same computational reasons, we omit the SEP token to allow more actual codes to be embedded per patient. In our initial experimentation, we found no gains from using a masked language model to pretrain transformer architectures (including both BEHRT and SARD) in our setting. We instead used the method of Choi, Chiu, and Sontag (2016) for initialization in all cases. We further discuss our choice of baselines in the Appendix.

We train using a single NVIDIA k80 GPU. Our algorithms are implemented in Python 3.6 and use the PyTorch autograd library (Paszke et al. 2019). We train our deep models using an ADAM optimizer (Kingma and Ba 2014) with the hyperparameter settings of $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-9}$ and a learning rate of $\eta = 2 \times 10^{-4}$. A batch size of 500 patients was used for ADAM updates.

Prediction Tasks. We consider three tasks important for predictive healthcare:

1: The *End of Life (EoL)* prediction task: we estimate patient mortality over a six-month window. This task is key to proactively providing palliative care to patients.

2: The *Surgical Procedure (Surgery)* prediction task: we predict if a patient will require any surgical procedure in a six-month window. If so, an appropriate, intervention can be taken early on.

3: The *Likelihood of Hospitalization (LoH)* prediction task: we estimate if a patient will require inpatient hospitalization in a six-month window. This allows for early interventions that could mitigate the need for hospitalization.

We split the $121,593$ patients into training, validation, and test sets of size $82,955$, $19,319$, and $19,319$ respectively. Data was collected up to the end of the calendar year 2016, and outcomes measured between April and September of 2017 – patients who had an outcome in the three-month gap between the end of data collection and the outcome measurement were excluded from the dataset. We denote the set of all OMOP concepts used in the dataset by $\mathcal{C}$, which in our case contained $|\mathcal{C}| = 37,004$ codes. All models are trained using the SARD architecture, using reverse distillation with early stopping for both pre-training and fine-tuning. SARD models are trained with $d_e = 300$ and $K = 10$; we found that validation performance did not increase with larger embedding sizes or number of convolutional kernels. Early stopping and the selection of the hyperparameters as outlined in Appendix Table 5 are performed using the validation set, and the parameters that maximized validation ROC-AUC are used to evaluate performance on the test set.

Our metric for measuring the performance is the area under the receiver-operator curve (ROC-AUC), i.e. the area under a plot of the true positive rate of the model as a function of false positive rate. An equivalent interpretation is the probability that the model gives a higher score to a random positive-outcome patient than a random negative-outcome patient. Thus, ROC-AUC is a good proxy for the application of choosing which patients should receive early interventions. While in class-balanced problems metrics like accuracy are useful, and in cases of extreme class imbalance metrics like AUC-PRC may provide insights, our metric is meaningful across a wide variety of class imbalances that may occur in the clinical domain. Indeed, our class balances range from 1.8% for EoL, to 8.5% for LoH, to 57.8% for Surgery. Nevertheless, for completeness, we also provide an AUC-PRC comparison in the Appendix, and find that SARD continues to outperform baselines.

Model \ Task Name	EoL	Surgery	LoH
L_1-reg. logistic regression (Razavian et al. 2015)	83.4	79.2	73.1
RETAIN (Choi et al. 2016b)	82.2	79.8	72.5
BEHRT (Li et al. 2020)	83.1	80.3	71.2
BEHRT + RD	83.7	81.1	73.7
SARD (no RD)	85.0	82.7	72.7
SARD	**85.6**	**83.1**	**74.3**

Table 1: AUC-ROC Scores on Test Set. + RD indicates that reverse distillation is used for pre-training. Increases in AUC-ROC for SARD are significant versus the closest baseline in all cases (paired z-test, $p < .005$).

Main Results

As seen in Table 1, our model outperforms all baselines for each of the example tasks. Increases in AUC-ROC are significant versus the closest baseline in all cases (paired z-test, $p < .005$) (DeLong, DeLong, and Clarke-Pearson 1988). Notably, while the SARD model has the absolute highest performance, RD pre-training still offers improvement to the BEHRT baseline; through ablation studies, we show that RD similarly improves performance across additional, varied architecture choices. In the next section, we explore the nuances of how SARD extracts clinical narratives, and qualitatively find that SARD is able to use a patient's entire medical history to contextualize visits, whereas the high-performing linear models are not able to make these connections.

Ablation Studies. We empirically test the design decisions made in our SARD Model Architecture section via ablation studies. These studies validate our architecture choices, as ablation of both SARD's self-attention mechanism and its convolutional prediction head lead to performance decreases.

As seen in Figure 1, the SARD architecture naturally splits into modular parts, the two most important of which - the transformer and the prediction head - we investigate via ablation:

- **Self-attention**: A key aspect of our work is its use of a self-attention architecture as a tool to ingest time-series of embedded clinical data. Until recently, RNN-based approaches (Choi et al. 2016a,b; Ma et al. 2017) have been the state-of-the-art, and as such we developed an ablation study in which we replace our architecture with a unidirectional recurrent GRU-cell network, leaving the rest of

the network unchanged. This GRU-cell network used input dimension $d_e = 300$ and hidden dimension $d_e = 300$. In Table 2, the row `RNN (no RD)` corresponds to this ablated model trained from a random initialization, and `RNN + RD` to the ablated model trained using the same reverse distillation procedure used in SARD.

To ensure that our ablation fairly compared recurrent and self-attention based approaches, we preserved all other architectural elements including the visit-level input embeddings, use of temporal embeddings (fixed-frequency sinusoidal time embeddings led to the best performance), and the prediction head to aggregate the final visit representations, which here operates on the hidden states of each element of the last layer of the RNN. We found the prediction head's aggregation to be more performant and serve as a more apt comparison than the standard recurrent technique of simply predicting from the hidden state of the last element of the last layer of the RNN. This design choice helps mitigate the fact that older visits may be 'forgotten' by the RNN, by allowing these visits to directly influence the inputs of the prediction head. We find that the self-attention architecture is competitive with the RNN, so long as the RNN is also trained with reverse distillation. An important finding is that reverse distillation can also be used to successfully train highly-performant recurrent models, further validating the usefulness of this method and indicating that it can be used more generally.

We performed a similar ablation in which we replaced the the self-attention layers with the identity, to further evaluate the value of explicitly contextualizing visits. In Table 2, the row `Identity (no RD)` corresponds to this ablated model trained from a random initialization, and `Identity + RD` to this model trained using the same reverse distillation procedure used in SARD. We find that self-attention and recurrent architectures improve performance on our surgery task, but have less of an impact on our other two tasks; why this is requires further investigation. Furthermore, the strong performance of our identity ablation speaks to the strength of our convolutional prediction head, a design choice that likely contributes to our improvement over the previous state of the art, BEHRT.

- **Prediction Head**: We also ablate our convolutional prediction head by replacing it with a naive alternative which simply sums the contextualized vector representations of all visits to obtain a vector $\sum_j \tilde{\psi}(V_j^i)$ representing the entire history of patient i. This summed vector, which will have dimension d_e, is then passed into a single linear layer with sigmoid activation to make a final prediction. We use input embedding, sinusoidal time embedding and a self attention mechanism identical to those of the SARD model described in our SARD Model Architecture section.

  We find that SARD's convolutional prediction head gives performance increases when compared with this simpler alternative. Even in this regime, we again find that reverse distillation allows models to be more performant. In Table 2, the row `Summing Head (no RD)` corresponds to this ablated model trained from a random initialization, and `Summing Head + RD` to this model trained using the same reverse distillation procedure used in SARD.

Design Choice \ Task Name	EoL	Surgery	LoH
SARD	85.6	83.1	74.3
SARD (no RD)	85.0	82.7	72.7
Replace Self-Attention with:			
RNN + RD	85.5	82.8	74.1
RNN (no RD)	84.3	82.3	72.6
Identity + RD	85.3	81.6	74.1
Identity (no RD)	84.3	79.9	73.2
Replace Convolutional Prediction Head with:			
Summing Head + RD	84.2	82.4	74.2
Summing Head (no RD)	83.1	81.6	72.0

Table 2: Ablation Study Results. + RD indicates that reverse distillation is used for pre-training

As seen in Table 2, our design choices perform as well as or better than alternatives. Importantly, our ablation studies highlight that in addition to architectural innovations, reverse distillation is a key driver in SARD's performance gains, and more generally in performance gains across diverse architectures. Indeed, the smallest difference in ablated performance was observed when SARD's self-attention architecture was replaced with a recurrent equivalent, but reverse distillation was still used for pre-training, indicating reverse distillation's universal applicability.

Model Introspection

In healthcare applications, it is critical to understand and interpret how models make predictions. In this section we employ a local, or per-prediction, method of introspecting on the SARD model; specifically, we examine which visits are most influential in the prediction head for a given individual, and how those visits leverage self-attention to contextualize. Our primary goal in this analysis is to introspect on the SARD model to better understand how its self-attention architecture leverages and transforms our input features to make improved predictions, and we note that further work would be needed before using such interpretation methods to justify clinical decisions.

To determine which visits are most influential to a prediction, we introspect directly on our convolutional prediction head. In notating this introspection, we suppress indices corresponding to batches (i.e. patients), as the introspection will be ultimately performed at the level of a single individual.

Recall that the prediction head convolves K kernels of size $d_e \times 1$ with the final contextualized visit representations, then uses a max-pooling operation to return the maximum cross-correlation between the kernel and any individual contextualized visit. For the k^{th} of these K kernels, denote this maximum cross-correlation value by χ_k, and the maximizing visit by ν_i. Let w_k denote the weight given to the output from the k^{th} kernel in the final linear layer mapping to a prediction. We assign a score of $s(V_j) = \sum_k [[V_j = \nu_k]] w_k \sigma(\chi_k)$ to visit V_j, where σ represents the sigmoid

nonlinearity applied after max-pooling. This metric represents the total importance of visit V_j by summing all of its possible contributions to the final prediction.

We use these introspection techniques in the Appendix to interpret the case of a $\geq$90 year-old female patient whose death was predicted with high probability (71.1%) by SARD, but missed by our baseline windowed linear model (5.4% probability of death). Using the total importance metric described above, we can find the most predictive visits for our case study patient in SARD. We present her top four visits, which include visits from 2011, 2015 and 2016 in which the patient chiefly experienced cardiovascular diseases and their complications, in Appendix Table 7.

We then seek to understand how each visit is contextualized by examining its attention weights in SARD's self-attention layers. For example, in our case study, we examine the visits attended to most strongly by the patient's top visit; we include these results in Appendix Table 8 and visualize the attention weights from her top visit in Appendix Figure 5. We find that while this patient's top visit occurred in 2016 and included detection of a myocardial infarction along with other cardiovascular disease, her top visit strongly attends to a cluster of visits in 2011. By carefully analyzing these visits, we find that during the 2011 visits, the patient experienced other manifestations of atherosclerotic vascular disease. We conjecture that these continued, albeit more minor, cardiovascular issues over the years provide context for the 2016 visit, and ultimately augment the risk of death associated with the events of the 2016 visit.

More generally, introspecting on the SARD model reveals that its self-attention mechanism leverages important contextual information from throughout a patient's history to gain a nuanced understanding of which parts of the medical timeline are most important for prediction. Thus, the deep model is able to make better predictions than simpler baselines when it is necessary to interpret an entire clinical narrative. In particular, in cases where SARD outperforms linear baselines, patients have significantly *more* data, as measured by the patient's total number of visits, than in cases where the linear baseline outperforms (Mann-Whitney U test, $p < .05$).

Model Performance Across Subpopulations. For any clinical machine learning model, it is important to introspect on and be aware of differential performance across different groups of patients. We evaluate SARD's performance across a diverse range of patient clinical categories. We consider subpopulations defined by Clinical Classifications Software Refined (CCSR) codes[2], and place a patient in a subpopulation if they experience at least three occurrences of a related condition within two years of prediction time. For the LoH task, Figure 2 shows the positive predicted value (PPV, computed at a sensitivity of 0.5 for each category and model) of SARD trained with and without RD across the 189 CCSR categories with at least 10 positive outcomes in the associated subpopulations. In addition to improvements in overall AUC, we find that SARD trained with RD outperforms a SARD model trained without RD in 147 out of 189 categories, spanning many diverse subpopulations, such as patients with immunity disorders and neonatal disorders.

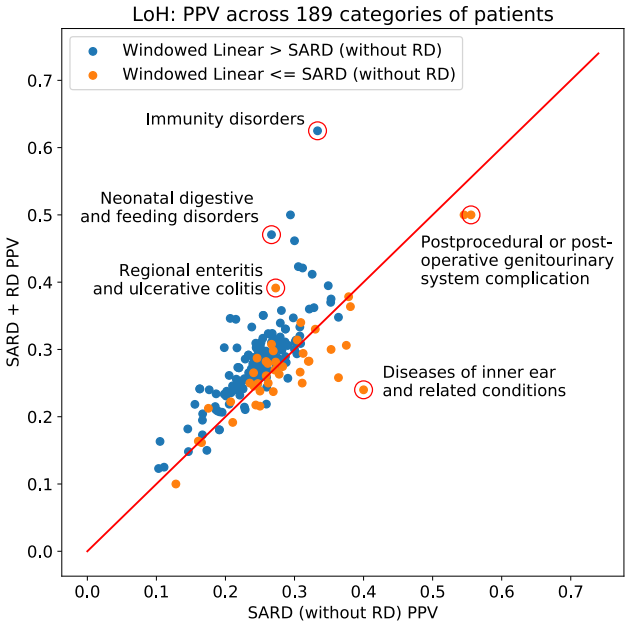

Figure 2: PPV for SARD with vs. without RD across subpopulations in the LoH task. Each point represents a patient category.

Figure 2 also indicates whether the windowed linear baseline performed better than SARD without RD on each subpopulations, in terms of PPV. We find that for almost all categories where SARD outperforms SARD without RD, the linear baseline also outperforms. This corroborates our understanding that the success of SARD's unique pre-training procedure emanates from its ability to capture performant aspects of the linear baseline.

Analyses of Reverse Distillation

We empirically validate that the SARD model for the End of Life task after reverse distillation (but before fine-tuning) generalizes in the same way as a linear model by analyzing the predictions made by both models on a held-out validation set. As seen in Figure 3, we find a Spearman correlation of 0.897 between the logit outputs of the two models on held-out data[3]. This indicates that even for unseen patients, the models make similar predictions. Thus, the reverse-distilled deep model does indeed mimic the linear model, not just memorize its outputs at certain points.

Reverse distillation is further analyzed via experiments on synthetic data in the Appendix. We find performance gains through reverse distillation for classification problems where data are poorly separated, or where only a small fraction of features are relevant, both properties of our prediction tasks.

[2]CCSR for ICD-10-CM Diagnoses was developed as part of the Healthcare Cost and Utilization Project (HCUP), www.hcup-us.ahrq.gov/toolssoftware/ccsr/dxccsr.jsp.

[3]Recall that the logit corresponding to an output probability p is $\log(p/(1-p))$

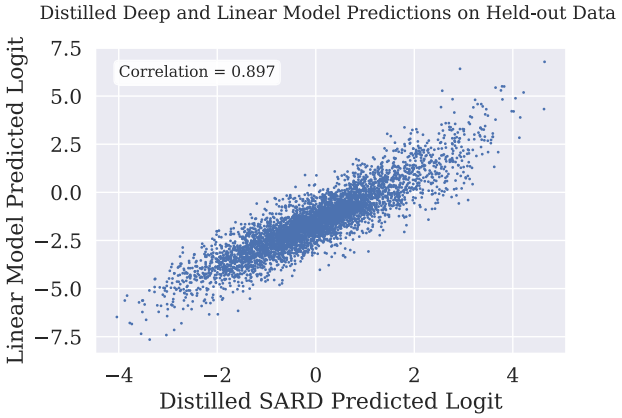

Figure 3: Comparison of Predictions on Held-out Data by Reverse Distilled and Linear Models

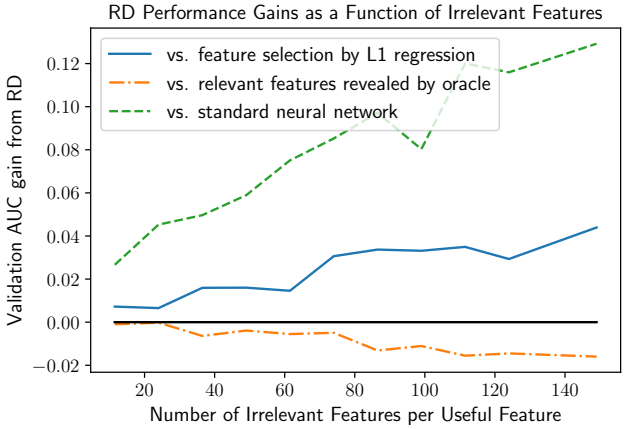

Figure 4: Reverse distillation AUC gains on synthetic data, as a function of sparsity of useful features

The ability of reverse distillation to enhance performance in synthetic scenarios with this property is shown in Figure 4, where we additionally compare to alternative feature-selection methods.

These experiments support that in addition to generalizing in the same way as an underlying linear model, a deep model trained via reverse distillation learns a soft version of the feature-selection performed by a regularized linear model. This is especially interesting in the case of multi-dimensional time-series data, where a simpler feature selection algorithm is not applicable. Indeed, in the case of longitudinal data, we would need to select a temporal context per feature, not just the features themselves. A naive approach of limiting SARD to the features selected by the windowed linear baseline in any time window results in no performance gains versus the baseline.

Network Dissection: We present the results of our Network Dissection approach for intepretability. We summarize the findings of our correlation analysis as follows: for each neuron in SARD's penultimate layer, we "match" it to the single linear model feature with which it had the highest MCC correlation; the linear model we refer to is the L1-regularized windowed logistic regression used for pre-training, and we only include features which have non-zero coefficients. In Table 3 we report the total number of unique linear features which "matched" at least one of the latent features in the penultimate layer of each deep model.

Model \ Task Name	EoL	Surgery	LoH
Total # of Non-Zero Linear Features	106	2000	1009
SARD (no RD)	41 (39%)	52 (3%)	43 (4%)
RD Only	71 (67%)	86 (4%)	161 (16%)
SARD	71 (67%)	69 (3%)	144 (14%)

Table 3: Number (percentage) of unique linear model features represented by the final latent layer in the following model variants: SARD trained without RD pre-training (SARD (no RD)), SARD paused after pre-training (RD Only), and SARD with pre-training and fine-tuning (SARD).

Unsurprisingly, we observe that the penultimate layers of our SARD networks trained *without* RD pre-training do not capture a high fraction of the linear model's feature set. After RD pre-training, a much higher fraction of the linear model's features are represented by the penultimate layers of the deep models, and they remain so even after fine-tuning, highlighting RD's ability to effectively regularize even a fine-tuned model to make use of features known to be clinically meaningful. This helps explain the performance gains driven by reverse distillation seen in our experiments.

To better understand the impact of RD at the neuron level, we provide examples of top correlations for penultimate layer neurons trained with different SARD variants on the EoL task in the Appendix (see Tables 9 and 10). For example, when training the network with RD, *before* fine-tuning, we find a neuron with correlation .487 with the linear model feature "Hearing loss", .414 with "Dementia", and .4 with "Alzheimer's disease" (for all three, the ∞-time window). After fine-tuning, the same neuron has correlation .403 with "Hearing loss", .32 with "Dementia", and .312 with "Subsequent hospital care", keeping the same broad interpretation although with a new emphasis on hospitalization. By contrast, none of the top 10,000 correlations for SARD trained *without* RD include a neuron correlated with the linear model feature for "Hearing loss."

Discussion

We showed in Table 1 that two of the previous state-of-the-art deep models for longitudinal health data (Choi et al. 2016b; Li et al. 2020) do not outperform a well-tuned linear model with windowed features, consistent with previously reported results (Rajkomar et al. 2018, Supplemental Table 1). When trained without reverse distillation, our new archi-

tecture, SARD, achieves substantial wins in two of the tasks, yet also performs worse than the linear model on the third. However, when the models are pre-trained using reverse distillation, all of the architectures outperform the linear model, with SARD obtaining the best performance. Reverse distillation is just one successful method by which self-attention based predictive models can be initialized. Although we did not observe an advantage in our dataset, possibly because of the small number of individuals relative to the large vocabulary, Li et al. (2020) demonstrated the use of masked language models as an unsupervised pre-training method for transformer-based models.

We hypothesize that reverse distillation will be of utility in other applications of deep learning with limited data where strong shallow models already exist. For example, within healthcare, interpretation of ECG waveforms (e.g. to predict atrial fibrillation) with deep models could be pre-trained with reverse distillation using linear models on easily derived clinical features such as R-R intervals (Teijeiro et al. 2018). Beyond healthcare, text classification in under-resourced languages without pre-trained language models might benefit from reverse distillation using linear models with bag-of-words features.

We showed in Lemma 1 that our transformer architecture with temporal embeddings can represent a windowed linear model. However, that does not imply that gradient descent will learn a function that is equivalent to the linear model used within pre-training – the objective is nonconvex and, even with infinite training data, there will be many equivalently good solutions. Nonetheless, we showed in Figure 3 that the function learned by the deep model closely mirrors the function learned by the linear model on held-out data. A possible theoretical explanation might be found in recent work on convergence of stochastic gradient descent in over-parameterized deep models, coupled with the realization that pre-training is attempting to fit a particularly simple concept class, a linear model (Allen-Zhu, Li, and Song 2019).

Acknowledgements

This work was supported by Independence Blue Cross and would not have been possible without the advice and support of Aaron Smith-McLallen, Ravi Chawla, Kyle Armstrong, Luogang Wei, and Jim Denyer. The Tesla K80s used for this research were donated by the NVIDIA Corporation.

References

Ahmad, M. A.; Eckert, C.; McKelvey, G.; Zolfagar, K.; Zahid, A.; and Teredesai, A. 2018. Death vs. data science: predicting end of life. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Allen-Zhu, Z.; Li, Y.; and Song, Z. 2019. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, 242–252. PMLR.

Avati, A.; Jung, K.; Harman, S.; Downing, L.; Ng, A.; and Shah, N. H. 2018. Improving palliative care with deep learning. *BMC medical informatics and decision making* 18(4): 122.

Bau, D.; Zhu, J.-Y.; Strobelt, H.; Zhou, B.; Tenenbaum, J. B.; Freeman, W. T.; and Torralba, A. 2018. Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597* .

Baytas, I. M.; Xiao, C.; Zhang, X.; Wang, F.; Jain, A. K.; and Zhou, J. 2017. Patient subtyping via time-aware lstm networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 65–74.

Bellamy, D.; Celi, L.; and Beam, A. L. 2020. Evaluating Progress on Machine Learning for Longitudinal Electronic Healthcare Data. *arXiv preprint arXiv:2010.01149* .

Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; and Liu, Y. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports* 8(1): 1–12.

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

Choi, E.; Bahadori, M. T.; Song, L.; Stewart, W. F.; and Sun, J. 2017. GRAM: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 787–795.

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

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

DeLong, E. R.; DeLong, D. M.; and Clarke-Pearson, D. L. 1988. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics* 837–845.

Gao, J.; Xiao, C.; Wang, Y.; Tang, W.; Glass, L. M.; and Sun, J. 2020. StageNet: Stage-Aware Neural Networks for Health Risk Prediction. In *Proceedings of The Web Conference 2020*, 530–540.

Harutyunyan, H.; Khachatrian, H.; Kale, D. C.; Ver Steeg, G.; and Galstyan, A. 2019. Multitask learning and benchmarking with clinical time series data. *Scientific data* 6(1): 1–18.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* .

Hripcsak, G.; Duke, J. D.; Shah, N. H.; Reich, C. G.; Huser, V.; Schuemie, M. J.; Suchard, M. A.; Park, R. W.; Wong, I. C. K.; Rijnbeek, P. R.; et al. 2015. Observational Health Data Sciences and Informatics (OHDSI): opportunities for observational researchers. *Studies in health technology and informatics* 216: 574.

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

Kwon, B. C.; Choi, M.-J.; Kim, J. T.; Choi, E.; Kim, Y. B.; Kwon, S.; Sun, J.; and Choo, J. 2018. Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records. *IEEE transactions on visualization and computer graphics* 25(1): 299–309.

Li, Y.; Rao, S.; Solares, J. R. A.; Hassaine, A.; Ramakrishnan, R.; Canoy, D.; Zhu, Y.; Rahimi, K.; and Salimi-Khorshidi, G. 2020. BeHRt: transformer for electronic Health Records. *Scientific Reports* 10(1): 1–12.

Ma, F.; Chitta, R.; Zhou, J.; You, Q.; Sun, T.; and Gao, J. 2017. 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*, 1903–1911.

Ma, F.; Wang, Y.; Xiao, H.; Yuan, Y.; Chitta, R.; Zhou, J.; and Gao, J. 2018. A general framework for diagnosis prediction via incorporating medical code descriptions. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 1070–1075. IEEE.

Ma, L.; Zhang, C.; Wang, Y.; Ruan, W.; Wang, J.; Tang, W.; Ma, X.; Gao, X.; and Gao, J. 2020. Concare: Personalized clinical feature embedding via capturing the healthcare context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 833–840.

Miotto, R.; Li, L.; Kidd, B. A.; and Dudley, J. T. 2016. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports* 6(1): 1–10.

Nguyen, P.; Tran, T.; Wickramasinghe, N.; and Venkatesh, S. 2016. Deepr: a convolutional net for medical records. *IEEE journal of biomedical and health informatics* 21(1): 22–30.

Niculescu-Mizil, A.; and Caruana, R. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, 625–632.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 8024–8035. Curran Associates, Inc. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

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

Razavian, N.; Blecker, S.; Schmidt, A. M.; Smith-McLallen, A.; Nigam, S.; and Sontag, D. 2015. Population-level prediction of type 2 diabetes from claims data and analysis of risk factors. *Big Data* 3(4): 277–287.

Song, H.; Rajan, D.; Thiagarajan, J. J.; and Spanias, A. 2017. Attend and diagnose: Clinical time series analysis using attention models. *arXiv preprint arXiv:1711.03905* .

Steinberg, E.; Jung, K.; Fries, J. A.; Corbin, C. K.; Pfohl, S. R.; and Shah, N. H. 2020. Language Models Are An Effective Patient Representation Learning Technique For Electronic Health Record Data. *arXiv preprint arXiv:2001.05295* .

Teijeiro, T.; García, C. A.; Castro, D.; and Félix, P. 2018. Abductive reasoning as a basis to reproduce expert criteria in ECG atrial fibrillation identification. *Physiological Measurement* 39(8): 084006. ISSN 1361-6579.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

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

Zhang, F.; Wu, T.; Wang, Y.; Cai, Y.; Xiao, C.; Zhao, E.; Glass, L.; and Sun, J. 2019. Predicting treatment initiation from clinical time series data via graph-augmented time-sensitive model. *arXiv preprint arXiv:1907.01099* .