Neural Temporal Point Processes Modelling Electronic Health Records

Joseph Enguehard* JOSEPH.ENGUEHARD@BABYLONHEALTH.COM
Babylon Health

Dan Busbridge*
Babylon Health

Adam Bozson
Babylon Health

Claire Woodcock
Babylon Health

Nils Hammerla
Babylon Health

Editors: Emily Alsentzer⊗, Matthew B. A. McDermott⊗, Fabian Falck, Suproteem K. Sarkar, Subhrajit Roy‡, Stephanie L. Hyland‡

Abstract

The modelling of Electronic Health Records (EHRs) has the potential to drive more efficient allocation of healthcare resources, enabling early intervention strategies and advancing personalised healthcare. However, EHRs are challenging to model due to their realisation as noisy, multi-modal data occurring at irregular time intervals. To address their temporal nature, we treat EHRs as samples generated by a Temporal Point Process (TPP), enabling us to model what happened in an event with when it happened in a principled way. We gather and propose neural network parameterisations of TPPs, collectively referred to as Neural TPPs. We perform evaluations on synthetic EHRs as well as on a set of established benchmarks. We show that TPPs significantly outperform their non-TPP counterparts on EHRs. We also show that an assumption of many Neural TPPs, that the class distribution is conditionally independent of time, reduces performance on EHRs. Finally, our proposed attention-based Neural TPP performs favourably compared to existing models, whilst aligning with real world interpretability requirements, an important step towards a component of clinical decision support systems.

1. Introduction

Healthcare systems today are under intense pressure. Costs of care are increasing, resources are constrained, and outcomes are worsening (Topol, 2019). Given these intense challenges, healthcare stands as one of the most promising applications of Machine Learning (ML). In particular, interest in modelling Electronic Health Records

* Equal contribution. Joseph Enguehard and Dan Busbridge led the research, implemented the models and performed the experiments. Adam Bozson created the synthetic EHR datasets and provided insights during project inception. Claire Woodcock strategically guided the project and contributed the broader impact statement. Nils Y. Hammerla provided technical guidance throughout.

(EHRs) has recently increased (Islam et al., 2017; Shickel et al., 2018; Darabi et al., 2019; Li et al., 2019; Rodrigues-Jr et al., 2019). Better EHR modelling at an institutional level could enable more effective allocation of healthcare (Gijsberts et al., 2015). At a clinical level accurate models could provision tools which improve patient outcomes by automating labour intensive administrative tasks (Dhruva et al., 2020) and developing early and optimal intervention strategies for doctors (Komorowski, 2019). However, facets of health evolve at differing rates, and EHRs are typically realised as noisy, multimodal data occurring at irregular time intervals. This makes EHRs difficult to model using common ML methods.

We propose to address the temporal nature of longitudinal EHRs[1] by treating them as samples generated by a Temporal Point Process (TPP), a probabilistic framework which can deal with data occurring at irregular time intervals. Specifically, we use a Neural Network (NN) to approximate its density, which specifies the probability of the next event happening at a given time. We call these models Neural TPPs. We propose jointly modelling times and labels to capture the varying evolution rates within EHRs.

To test this, aware of the need for transparency in research (Pineau et al., 2020) and for sensitive handling of health data (Kalkman et al., 2019), we perform an extensive study on synthetic EHR datasets generated using the open source Synthea simulator (Walonoski et al., 2018). For completeness, we also perform evaluations on benchmark datasets commonly used in the TPP literature. We find that:

- Our proposed Neural TPPs, where labels are jointly modelled with time, significantly outperform those that treat them as conditionally independent; a common simplification in the TPP literature,
- Particularly in the case of TPPs on labelled data, metrics that decompose the performance in terms of both label and temporal accuracy should be reported,
- Some datasets used for benchmarking TPPs are easily solved by a time-independent model. On other datasets, including synthetic EHRs, TPPs outperform their non-TPP counterparts.

Finally, we present a Neural TPP whose attention-based mechanism provides interpretable information without compromising on performance. This is an essential contribution in the development of research for clinical applications. The wider impact of pursuing this direction is detailed in Appendix E.

2. Background

2.1. Temporal point processes

A TPP is a random process that generates a sequence of N events $\mathcal{H} = \{(t_i, \mathcal{M}_i)\}_{i=1}^{N}$ within a given observation window $t_i \in [w_-, w_+]$. Each event consists of a set of labels $\mathcal{M}_i \subseteq \{1, \dots, M\}$ localised at times $t_{i-1} < t_i$. Labels may be independent or mutually exclusive, depending on the task.

A TPP is fully characterised through its conditional intensity $\lambda_m^*(t)$, for $t_{i-1} < t \leq t_i$:

$$\lambda_m^*(t)\, dt = \lambda_m(t|\mathcal{H}_t)\, dt \tag{1}$$

$$= \Pr\left(t_i^m \in [t, t+dt)|\mathcal{H}_t\right), \tag{2}$$

which specifies the probability that a label m occurs in the infinitesimal time interval $[t, t+dt)$ given past events $\mathcal{H}_t = \{t_i^m \in \mathcal{H}|t_i < t\}$.

1. Longitudinal EHRs consist of comprehensive records of patients' clinical experience, usually spanning over several decades. This type of data does not include ICU datasets, such as MIMIC-III (Johnson et al., 2016) which focus on a shorter time frame.

We follow Daley and Vere-Jones (2003) in using $\lambda_m^*(t) = \lambda_m(t|\mathcal{H}_t)$ to indicate $\lambda_m^*(t)$ is conditioned on past events, and t_i^m to refer to a label of type m at time t_i. Given a specified conditional intensity $\lambda_m^*(t)$, the conditional density $p_m^*(t)$ is, for $t_{i-1} < t \leq t_i$,

$$p_m^*(t) = \lambda_m^*(t) \exp\left[-\sum_{n=1}^{M} \Lambda_n^*(t)\right], \quad (3)$$

where $\Lambda_m^*(t)$ is the cumulative intensity:

$$\Lambda_m^*(t) = \Lambda_m(t|\mathcal{H}_t) = \int_{t_{i-1}}^{t} \lambda_m^*(t')\, dt'. \quad (4)$$

In a multi-class setting, where *exactly* one label (displayed as the indicator function $\mathbb{1}$) is present in any event, the log-likelihood of a sequence $\mathcal{H}$ is a form of categorical cross-entropy:

$$\log p_{\text{multi-class}}(\mathcal{H}) = \underbrace{\sum_{m=1}^{M}\sum_{i=1}^{N} \mathbb{1}_{i,m} \log p_m^*(t_i)}_{\text{events at } t_1,\ldots,t_N} - \underbrace{\sum_{m=1}^{M} \int_{t_N}^{w_+} \lambda_m^*(t')\, dt'}_{\text{no events between } t_N \text{ and } w_+}, \quad (5)$$

where $\mathcal{H}_{t_0} = \mathcal{H}_{t_1} = \{\}$, and $t_0 = w_-$ but does not correspond to an event.

In a multi-label setting, where *at least* one label is present in any event, the log-likelihood of a sequence $\mathcal{H}$ is a form of binary cross-entropy:

$$\log p_{\text{multi-label}}(\mathcal{H}) = \log p_{\text{multi-class}}(\mathcal{H}) + \sum_{m=1}^{M}\sum_{i=1}^{N} (1 - \mathbb{1}_{i,m}) \log\left[1 - p_m^*(t_i)\right]. \quad (6)$$

This setting should be especially useful to model EHRs, as a single medical consultation usually includes various events, such as diagnoses or prescriptions, all happening at the same time.

2.2. Neural temporal point processes

Encoder-decoder architectures have proven effective for Natural Language Processing (NLP) (Cho et al., 2014; Kiros et al., 2015; Hill et al., 2016; Vaswani et al., 2017). Existing Neural TPPs also exhibit this structure: the encoder creates event representations based only on information about other events; the decoder takes these representations and the decoding time to produce a new representation. The output of the decoder produces the conditional intensity and conditional cumulative intensity at that decoding time. For more detail, see Figure 1.

3. Previous work

While a Neural TPP encoder can be readily chosen from existing sequence models such as Recurrent Neural Networks (RNNs), choosing a decoder is much less straightforward due to the integral in Equation (4). Existing work can be categorised based on the relationship between the conditional intensity $\lambda^*(t)$ and conditional cumulative intensity $\Lambda^*(t)$. We focus on three approaches:

- Closed form likelihood: the conditional density $p^*(t)$ is closed form,
- Analytic conditional intensity: only $\lambda^*(t)$ is closed form, $\Lambda^*(t)$ is estimated numerically,
- Analytic cumulative conditional intensity: only $\Lambda^*(t)$ is closed form, $\lambda^*(t)$ is computed through differentiation.

3.1. Closed form likelihood

The closed form likelihood approach implies that the contribution from each event to the likelihood $p_m^*(t) = \lambda_m^*(t) \exp[-\sum_{n=1}^{M} \Lambda_n^*(t)]$ is closed form.

A well-known example is the Hawkes process (Hawkes, 1971; Nickel and Le, 2020),

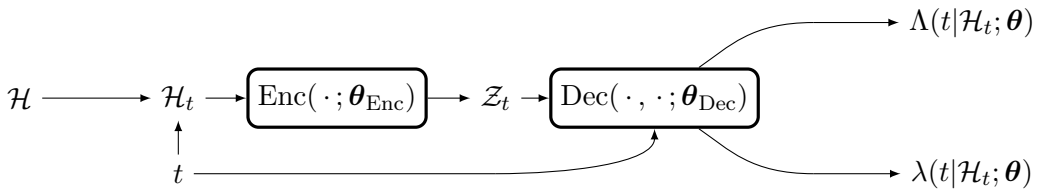


Figure 1: Encoder/decoder architecture of Neural TPPs. Given a query time t, the sequence $\mathcal{H}$ is filtered to the events $\mathcal{H}_t$ in the past of t. The encoder maps $\mathcal{H}_t$ to continuous representations $\mathcal{Z}_t = \{\boldsymbol{z}_i\}_{i=1}^{|\mathcal{H}_t|} = \text{Enc}(\mathcal{H}_t; \boldsymbol{\theta}_{\text{Enc}})$. Each $\boldsymbol{z}_i$ can be considered as a contextualised representation for the event at t_i. Given $\mathcal{Z}_t$ and t, the decoder outputs $\text{Dec}(t, \mathcal{Z}_t; \boldsymbol{\theta}_{\text{Dec}}) \in \mathbb{R}^M$ that the conditional intensity and conditional cumulative intensity are derived from without any learnable parameters.

which models the conditional intensity as

$$\lambda_m^*(t) = \mu_m + \sum_{n=1}^{M} \alpha_{m,n} \sum_{i:t_i^n<t} \exp[-\beta_{m,n}(t-t_i^n)]$$

with learnable parameters $\boldsymbol{\mu} \in \mathbb{R}_{>0}^M$, $\boldsymbol{\alpha} \in \mathbb{R}_{\geq 0}^{M\times M}$, and $\boldsymbol{\beta} \in \mathbb{R}^{M\times M}$. The closed form of $\Lambda_m^*(t;\boldsymbol{\theta})$ comes from the simple exponential linear t-dependence of $\lambda_m^*(t;\boldsymbol{\theta})$, which limits the model flexibility.

Du et al. (2016) leveraged the same closed form, conditioning an exponential linear decoder on the output of a RNN encoder (RMTPP). As with the Hawkes process, the model can only model exponential dependence in time. Additionally, it assumes labels are conditionally independent of time given a history, which we will show is a limiting assumption in domains like modelling EHRs.

An alternative to taking $\lambda^*(t)$ and $\Lambda^*(t)$ closed form is to take $p^*(t)$ closed form. Shchur et al. (2020) directly approximate the conditional density using a log-normal mixture (LNM). With a sufficiently large mixture, $p^*(t)$ can approximate any conditional density and is a more flexible model than the RMTPP. However, as in Du et al. (2016), labels are modelled conditionally independent of decoding time.

While no decode-time dependent Neural TPPs methods have been applied to modelling EHRs, Weiss and Page (2013) and Lian et al. (2015) apply conditional Poisson processes to various EHRs datasets, Islam et al. (2017) combines a conditional Poisson process with a Gaussian distribution to predict in hospital mortality using ICU data, and Zhang et al. (2020) uses an individual heterogeneous conditional Poisson model on longitudinal EHRs for adverse drug reaction discovery.

3.2. Analytic conditional intensity

The analytic conditional intensity approach approximates the conditional intensity with a NN whose output is positive

$$\lambda_m^*(t;\boldsymbol{\theta}) = \text{Dec}(t, \mathcal{Z}_t; \boldsymbol{\theta}_{\text{Dec}})_m \in \mathbb{R}_{\geq 0},$$

and whose t-integral must be approximated numerically.

The positivity constraint is satisfied by requiring the final activation function to be positive. Early Neural TPPs employed exponential activation (Du et al., 2016). Recently, the scaled softplus activation

$$\sigma_+(x_m) = s_m \log(1 + \exp(x_m/s_m)) \in \mathbb{R}_{>0}$$

with learnable $\boldsymbol{s} \in \mathbb{R}^M$ has gained popularity (Mei and Eisner, 2017). Ultimately, writing

a neural approximator for $\lambda_m^*(t)$ is relatively simple as there is no constraint on the NN architecture itself. Of course, this does not mean it is easy to train.

To approximate the integral, Monte Carlo (MC) estimation can be employed given a sampling strategy (Mei and Eisner, 2017). Zhu et al. (2020) also applies this strategy for spatio-temporal point process, which jointly models times and labels, but only uses an embedding layer as an encoder.

3.3. Analytic conditional cumulative intensity

The analytic conditional cumulative intensity approach approximates the conditional cumulative intensity with a NN whose output is positive

$$\Lambda_m^*(t;\boldsymbol{\theta}) = \text{Dec}(t, \mathcal{Z}_t; \boldsymbol{\theta}_{\text{Dec}})_m \in \mathbb{R}_{\geq 0}$$

and whose derivative is positive and approximates the conditional intensity

$$\lambda_m^*(t;\boldsymbol{\theta}) = d\text{Dec}(t, \mathcal{Z}_t; \boldsymbol{\theta}_{\text{Dec}})_m / dt \in \mathbb{R}_{\geq 0}$$

This derivative can be computed using backpropagation. While the cumulative intensity approach removes the variance induced by MC estimation, the monotonicity entails specific constraints on the NN.

Omi et al. (2019) uses a Multi-Layer Perceptron (MLP) with positive weights in order to model a monotonic decoder. It also uses tanh activation functions, and a softplus final activation function to ensure a positive output. However, this model does not handle labels, and has been criticised by Shchur et al. (2020) for violating $\lim_{t\to+\infty} \Lambda_m^*(t;\boldsymbol{\theta}) = +\infty$.

4. Proposed models

Our goal is to model the joint distribution of EHRs which contain long-term, non-sequential dependencies. This makes the successful application of RNNs challenging due to their recency bias (Bahdanau et al., 2016; Ravfogel et al., 2019). Additionally, RNNs do not meet the explainability standard required for real-world application; although there are methods for interpreting RNN outputs (Samek and Müller, 2019), when deployed in healthcare applications, models should be interpretable without relying on auxiliary models (Rudin, 2019).

To address these challenges, we propose the use of attention. This is capable of dealing with long range context (Liu et al., 2018), and benefits some model explainability[2] without requiring auxiliary models. In this section we develop the necessary building blocks to define attention-based Neural TPPs that directly model either the conditional intensity $\lambda_m^*(t)$ or the conditional cumulative intensity $\Lambda_m^*(t)$, alongside several benchmark Neural TPPs. In total, we present 2 encoders (Section 4.1) and 7 decoders (Section 4.2) whose combination yields the 14 NeuralTPPs we evaluate[3].

4.1. Encoders

As in RMTPPs (Du et al., 2016) we use a Gated Recurrent Unit (GRU) encoder (**GRU**). We also evaluate a continuous-time self-attention encoder (Vaswani et al., 2017; Xiong et al., 2020) (**SA**). Precise forms are presented in Appendix B.1.

4.2. Decoders

Precise forms for all decoders are presented in Appendix B.2.

2. This intepretability may be limited to the importance of events in a patient record (Serrano and Smith, 2019), which without auxiliary models provides much clinical utility. For example, rapidly provisioning important information about a patient to a doctor to facilitate greater focus on patient need (Overhage and McCallie, 2020).
3. Our implementation of these models, as well as the experimental setup can be found at https://github.com/babylonhealth/neuralTPPs.

4.2.1. Closed form likelihood

We take two decoders from the literature: a conditional Poisson process (**CP**), a **RMTPP** (Du et al., 2016), and a log-normal mixture (**LNM**) (Shchur et al., 2020).

4.2.2. Analytic conditional intensity

To model a positive $\lambda_m^*(t)$ we follow Mei and Eisner (2017): we apply a softplus activation to the final layer, and use MC estimation with a single, uniformly distributed sample per time interval.

Using this strategy, we propose a decoder based on a MLP (**MLP-MC**), and an attention mechanism (**Attn-MC**).

4.2.3. Analytic conditional cumulative intensity

Finally, we consider modelling $\Lambda_m^*(t) = \text{Decoder}(t, \mathcal{Z}_t; \boldsymbol{\theta})_m$. This is more difficult as the decoder must satisfy four properties:

1. $\Lambda_m^*(t) > 0$,
2. $\Lambda_m^*(t_i) = 0$,
3. $\lim_{t\to\infty} \Lambda_m^*(t) = \infty$,
4. $d\Lambda_m^*(t)/dt > 0$.

1. is solved using a final layer positive activation. 2. is solved by parameterising the decoder as $\text{Decoder}(t, \mathcal{Z}_t; \boldsymbol{\theta}) = f(t, \mathcal{Z}_t; \boldsymbol{\theta}) - f(t_i, \mathcal{Z}_t; \boldsymbol{\theta})$. 3. can be satisfied in two ways. The approach we take is to add a Poisson term to the conditional intensity (see Section 4.3). An alternative is to use a non-saturating activation function, which we investigate in Appendix A.

4. is the most challenging. If f represents a L-layer NN, where the output of each layer f_i is fed into the next f_{i+1}, we can write $f(t) = (f_L \circ \cdots \circ f_1)(t)$. Providing each step produces an output that is a monotonic in its input, i.e. $df_i/df_{i-1} \geq 0$ and $df_1/dt \geq 0$, then $f(t)$ is a monotonic function of t.

Given that softmax, layer normalisation, trigonometric encoding, and projection are inconsistent with monotonicity, many modifications were required in order to use MLPs and attention networks for cumulative modelling. Given the the importance of t-derivatives, we use the adaptive Gumbel activation (Farhadi et al., 2019), allowing learnable adaption of first derivatives. All proposed modifications are presented in Appendix A.

Using these modifications, we propose two cumulative decoders: a MLP (**MLP-CM**) and an attention mechanism (**Attn-CM**), which handle labels and address the criticisms of Shchur et al. (2020).

4.3. Base intensity

To every model[4] we add a Poisson process[5]

$$\boldsymbol{\lambda}^*_{\text{Total}}(t) = \alpha_1 \boldsymbol{\mu} + \alpha_2 \boldsymbol{\lambda}^*(t) \tag{7}$$

where $\boldsymbol{\alpha} = \text{softmax}(\boldsymbol{a})$, with $\boldsymbol{a}$ learnable.

We found that initialising $\boldsymbol{a}$ such that $\alpha_1 \sim e^3 \alpha_2$, i.e. starting the combined TPP as mostly Poisson, aided convergence.

5. Evaluation

5.1. Datasets

Dataset statistics are summarised in Table 1.

Hawkes Processes This synthetic data allows us to have access to a theoretically infinite amount of data. Moreover, as we know the intensity function, we were able to compare each of our modelled intensity against the true one. We designed one dataset consisting of two independent processes, **Hawkes (independent)**, and a second one allowing interactions between two processes, **Hawkes (dependent)**.

4. We do not add a Poisson term to the conditional Poisson process. Moreover, the original RMTPP does not include this Poisson term, however we include it to provide a fairer comparison.
5. In the TPP literature, the Poisson term is often referred to as the exogenous intensity, and the time-dependent piece the endogenous impact.

Table 1: Properties of datasets used for evaluation.

Dataset	# classes	Task type	# events	Avg. length	Size			
					Train	Valid	Test	Batch
Hawkes (ind.)	2	Multi-class	457,788	19	16,384	4,096	4,096	512
Hawkes (dep.)	2	Multi-class	607,512	25	16,384	4,096	4,096	512
MIMIC-II	75	Multi-class	2,419	4	585	NA	65	65
Stack Overflow	22	Multi-class	480,413	72	5,307	NA	1,326	32
Retweets	3	Multi-label	2,087,866	104	16,000	2,000	2,000	256
Ear infection	39	Multi-label	14,810	2	8,179	1,022	1,023	512
Synthea	357	Multi-label	496,625	43	10,524	585	585	64

Baselines We also compared our models on datasets commonly used to evaluate TPPs. We use: **MIMIC-II**[6], a medical dataset of clinical visits to Intensive Care Units, **Stack Overflow**, which classifies users on this question answering website, and **Retweets**, which consists in streams of retweet events by different types of Twitter users. Further details about the datasets and their preprocessing can be found in Du et al. (2016); Mei and Eisner (2017).

Synthetic EHRs We used the Synthea simulator (Walonoski et al., 2018) which generates patient-level EHRs using human expert curated Markov processes.

We created a dataset utilising all of Synthea's modules, **Synthea (Full)**, as well as a dataset generated solely from the ear infection module **Synthea (Ear Infection)**[7]. For each task, we generated approximately 10,000 patients, and we kept 10% of the data for validation and testing, using 5 different folds randomly chosen. Moreover, to optimise GPU utilisation, we truncated sequences larger than 400 events, which amounts to around 0.4% of the dataset.

We filtered out all events except conditions and medications. For each patient, we transformed event times by subtracting their birth date, then multiplying by 10^{-5}. When provided, their death date is used as the end of the observation window w_+, otherwise we use the latest event in their history.

5.2. Hyperparameters

Encoder and decoder were set to one layer each. The size of each layer was 8 for the small datasets (Hawkes and MIMIC-II), 32 for the medium ones (Ear infection, Stack Overflow and Retweets), and 64 for Synthea. Batch size adjusted per-dataset due to GPU memory availability (see Table 1). The Adam optimizer (Kingma and Ba, 2017) was used with the Noam learning rate scheduler (Vaswani et al., 2017). The scheduler had a maximum learning rate value of 0.01 and 10 epochs of warming-up, allowing the Poisson component to be adjusted before tuning the temporal piece. All our models were trained using Maximum Likelihood Estimation (MLE), and early stopping was employed with a patience of 100 epochs.

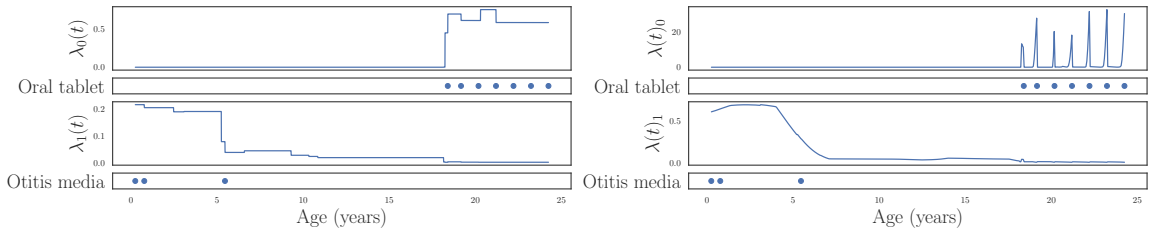
5.3. Metrics

Models were evaluated using a weighted F1 metric for multi-class datasets, and a weighted ROC-AUC metric for the multi-label ones. This enabled us to assess a model's ability predict the label(s) of the next event given the time it occurred and the previous events.

6. Although this dataset contains EHRs, they are not longitudinal. We only include it as a known TPP benchmark.
7. Ear infection was selected for a simpler benchmark as the generation process demonstrated temporal dependence

Table 2: Evaluation on multi-class tasks. On both tables, (†) indicates a model newly presented in this work. Best performance and performances whose confidence intervals overlap the best are boldened.

Encoder GRU	MIMIC-II		Stack Overflow		Hawkes (ind.)	Hawkes (dep.)
Decoder	F1 score	NLL/time	F1 score	NLL/time	NLL/time	NLL/time
CP	**.691 (.083)**	6.78 (1.99)	.325 (.004)	.553 (.003)	.623 (.002)	.774 (.004)
RMTPP	.215 (.12)	11.4 (3.57)	.284 (.005)	.592 (.006)	.610 (.004)	.731 (.004)
LNM	**.705 (.17)**	6.33 (.37)	.314 (.003)	.548 (.004)	**.605 (.001)**	**.727 (.001)**
MLP-CM†	.166 (.083)	12.2 (3.57)	.305 (.016)	.573 (.004)	.609 (.005)	**.728 (.003)**
MLP-MC†	**.665 (.050)**	9.45 (3.87)	**.333 (.008)**	**.540 (.006)**	**.605 (.001)**	**.727 (.001)**
Attn-CM†	.189 (.10)	10.4 (2.34)	.286 (.017)	.589 (.012)	.613 (.003)	.732 (.002)
Attn-MC†	**.641 (.131)**	**5.27 (2.05)**	.314 (.006)	.560 (.005)	.612 (.005)	.729 (.002)
Encoder SA						
CP	**.686 (.067)**	7.26 (1.44)	.326 (.003)	.555 (.003)	.622 (.001)	.774 (.002)
RMTPP	**.709 (.076)**	**4.24 (2.66)**	.288 (.002)	.592 (.002)	.609 (.005)	.734 (.002)
LNM	**.632 (.20)**	7.76 (2.42)	.305 (.003)	.561 (.006)	**.607 (.005)**	.730 (.004)
MLP-CM†	.159 (.088)	12.7 (3.45)	.292 (.003)	.634 (.042)	.611 (.005)	.733 (.002)
MLP-MC†	.498 (.150)	10.82 (2.13)	.327 (.016)	.557 (.015)	**.606 (.002)**	.730 (.001)
Attn-CM†	.186 (.082)	11.4 (3.90)	.269 (.011)	.665 (.047)	.614 (.001)	.733 (.001)
Attn-MC†	**.648 (.098)**	**4.61 (2.49)**	**.342 (.006)**	**.543 (.005)**	**.605 (.001)**	**.728 (.001)**



(*a*) GRU-Conditional-Poisson (*b*) GRU-Attn-MC

Figure 2: Intensity functions on several labels applied to the same EHR. We can see that the intensity of the GRU-Attn-MC decreases more smoothly than the GRU-CP on the Otitis media label and is able to spike on the recurring medicine prescription. This enables the GRU-Attn-MC to achieve a lower NLL than the GRU-CP.

Negative Log Likelihood (NLL) normalised by time[8] (NLL/time) was also used to evaluate models. This metric enables us to assess a model's ability to predict the label(s) of the next event, as well as when that event will happen given the previous events.

We want to stress that neither of these metrics is a perfect measure of performance in isolation. Indeed, F1 and ROC-AUC reward a model for accurate predictions given an event instance, but do not penalise a model for inaccurate predictions in the inter-event interval. This happens in Figure 2.

Alternatively, the NLL is prone to overfit: if a Neural TPP accurately guesses when an event will happen, it has the incentive to endlessly increase the intensity at that time, reducing the overall NLL without improving

8. Time normalisation makes the metric meaningful over sequences defined on different time intervals.

Table 3: Evaluation on multi-label tasks.

Encoder GRU	Retweets		Synthea (Ear infection)		Synthea (Full)	
Decoder	ROC-AUC	NLL/time	ROC-AUC	NLL/time	ROC-AUC	NLL/time
CP	**.611 (.001)**	5.71 (1.02)	.786 (.015)	37.6 (7.76)	**.850 (.014)**	89.6 (3.15)
RMTPP	.532 (.003)	-9.05 (1.40)	.672 (.013)	36.7 (4.25)	.616 (.043)	62.2 (6.63)
LNM	.521 (.010)	**-10.9 (2.11)**	.765 (.008)	**26.7 (5.35)**	.770 (.010)	**9.49 (6.37)**
MLP-CM†	.533 (.001)	**-10.4 (.301)**	.742 (.050)	30.4 (7.03)	.692 (.051)	34.2 (14.5)
MLP-MC†	.535 (.001)	**-10.3 (.186)**	.838 (.024)	**29.2 (9.09)**	.789 (.024)	**15.7 (9.57)**
Attn-CM†	.526 (.001)	**-10.0 (.535)**	.799 (.028)	**27.1 (4.95)**	.508 (.000)	80.9 (2.51)
Attn-MC†	.532 (.001)	-9.14 (.358)	**.857 (.005)**	**25.1 (4.45)**	.822 (.006)	21.9 (2.05)
Encoder SA						
CP	.608 (.001)	4.85 (.021)	.792 (.009)	38.6 (8.96)	.825 (.020)	91.2 (4.24)
RMTPP	.535 (.001)	-8.70 (.087)	.675 (.068)	42.9 (9.01)	.593 (.029)	67.8 (6.13)
LNM	.528 (.007)	**-10.9 (1.29)**	.767 (.007)	**26.2 (5.51)**	.760 (.014)	27.5 (27.2)
MLP-CM†	.512 (.001)	-8.93 (.180)	.684 (.094)	33.8 (15.3)	.589 (.123)	62.8 (30.8)
MLP-MC†	.533 (.001)	-9.21 (.174)	**.849 (.016)**	**24.9 (5.10)**	.796 (.007)	18.5 (1.20)
Attn-CM†	.517 (.001)	-8.54 (.168)	.697 (.059)	31.4 (6.90)	.503 (.004)	85.0 (3.01)
Attn-MC†	.534 (.001)	-9.48 (.188)	**.855 (.010)**	**24.5 (4.56)**	.805 (.007)	27.1 (4.77)

the resulting model meaningfully. Ideally, a model should perform well on both metrics.

6. Results

We report our results on multi-class tasks (Table 2) and on multi-labels tasks (Table 3).

First, we note that the GRU-CP has a competitive NLL and F1 score on the MIMIC-II and StackOverflow datasets. As the the CP decoder is decode time-independent, the result indicates that MIMIC-II and StackOverflow may not be appropriate for benchmarking TPPs. We suggest caution for future work using these datasets. All other datasets appear suitable. We focus here on EHRs, and discuss the Hawkes and Retweets results in Appendix C.

On EHRs, we see that modelling time and labels jointly yields better results than assuming them independent given the history. Indeed, most of our models outperform the RMTPP on all metrics, and, while the LNM is the best in terms of NLL, it achieves a significantly lower ROC-AUC score than all of our models except those using an Attn-CM decoder. Overall, our MC-based models outperform the CP in terms of NLL/time, and the LNM in terms of ROC-AUC.

Despite Attn-MC decoders performing strongly across all baseline datasets, as well as the Synthea (Ear Infection), this performance gap is not as wide on Synthea (Full). Studies have found Synthea to be medically representative (Walonoski et al., 2018; Chen et al., 2019). However, it is possible that noise in Synthea (Full) may be obfuscating the temporal progression of diseases (non-clinical investigation of the records showed 36.5% received a diagnosis of prediabetes after being diagnosed with diabetes). Applying Neural TPP to genuine EHRs would be a fruitful line of future enquiry.

Although the cumulative-based models have theoretical benefits compared with their Monte Carlo counterparts, they are in practice harder to train. This may explain their relatively poor performance on datasets such as on MIMIC-II and Synthea (Full), and the high variance between F1/ROC-AUC scores on these datasets.

7. Conclusion

In this work we gathered and proposed several neural network parameterisations of TPPs, evaluating them on synthetic EHRs, as well as common benchmarks.

Given the significant out-performance of our models on synthetic EHRs, labels should be modelled jointly with time, rather than be treated as conditionally independent; a common simplification in the TPP literature. We also note that common TPP metrics are not good performance indicators on their own. For a fair comparison we recommend using multiple metrics, each capturing distinct performance characteristics.

By employing a simple test that checks a dataset's validity for benchmarking TPPs, we demonstrated potential issues within several widely-used benchmark datasets. We recommend caution when using those datasets, and that our test be run as a sanity check of any new evaluation task.

Finally, we have demonstrated that attention-based TPPs appear to transmit pertinent EHR information and perform favourably compared to existing models. This is an exciting line for further enquiry in EHR modelling, where human interpretability is essential. Future work should apply these models to real EHR data to investigate medical relevance.

Acknowledgements

We are grateful to Sheldon Hall, Jeremie Vallee and Max Wasylow for assistance with experimental infrastructure, and to Kristian Boda, Sunir Gohil, Kostis Gourgoulias, Claudia Schulz and Vitalii Zhelezniak for fruitful comments and discussion.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*, May 2016.

Ramnath Balasubramanian, Ari Libarikian, and Doug McElhaney. Insurance 2030–The impact of AI on the future of insurance. https://www.mckinsey.com/industries/financial-services/our-insights/insurance-2030-the-impact-of-ai-on-the-future-of-insurance, 2018.

Karen L. Calderone. The influence of gender on the frequency of pain and sedative medication administered to postoperative patients. *Sex Roles*, 23(11):713–725, December 1990. ISSN 1573-2762. doi: 10/d87j9w.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*, pages 1721–1730, Sydney, NSW, Australia, 2015. ACM Press. ISBN 978-1-4503-3664-2. doi: 10/gftgxk.

Junqiao Chen, David Chun, Milesh Patel, Epson Chiang, and Jesse James. The validity of synthetic clinical data: a validation study of a leading synthetic data generator (synthea) using clinical quality measures. *BMC Medical Informatics and Decision Making*, 19(1), March 2019. doi: 10.1186/s12911-019-0793-0. URL https://doi.org/10.1186/s12911-019-0793-0.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv:1406.1078 [cs, stat]*, September 2014.

Daryl J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*. Springer, New York, 2nd ed edition, 2003.

ISBN 978-0-387-95541-4 978-0-387-21337-8 978-0-387-49835-5.

Sajad Darabi, Mohammad Kachuee, and Majid Sarrafzadeh. Unsupervised Representation for EHR Signals and Codes as Patient Status Vector. *arXiv:1910.01803 [cs, stat]*, October 2019.

Cláudia de Freitas and Graham Martin. Inclusive public participation in health: Policy, practice and theoretical contributions to promote the involvement of marginalised groups in healthcare. *Social Science & Medicine (1982)*, 135:31–39, June 2015. ISSN 1873-5347. doi: 10/f7gh39.

Sanket S. Dhruva, Joseph S. Ross, Joseph G. Akar, Brittany Caldwell, Karla Childers, Wing Chow, Laura Ciaccio, Paul Coplan, Jun Dong, Hayley J. Dykhoff, Stephen Johnston, Todd Kellogg, Cynthia Long, Peter A. Noseworthy, Kurt Roberts, Anindita Saha, Andrew Yoo, and Nilay D. Shah. Aggregating multiple real-world data sources using a patient-centered health-data-sharing platform. *npj Digital Medicine*, 3(1), April 2020. doi: 10.1038/s41746-020-0265-z. URL https://doi.org/10.1038/s41746-020-0265-z.

Ezio di Nucci. Should we be afraid of medical AI? *Journal of Medical Ethics*, 45(8):556–558, August 2019. ISSN 0306-6800, 1473-4257. doi: 10/ggsqsf.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent Marked Temporal Point Processes: Embedding Event History to Vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pages 1555–1564, San Francisco, California, USA, 2016. ACM Press. ISBN 978-1-4503-4232-2. doi: 10/ggntmn.

Farnoush Farhadi, Vahid Partovi Nia, and Andrea Lodi. Activation Adaptation in Neural Networks. *arXiv:1901.09849 [cs, stat]*, September 2019.

Michael A. Gara, Shula Minsky, Steven M Silverstein, Theresa Miskimen, and Stephen M. Strakowski. A Naturalistic Study of Racial Disparities in Diagnoses at an Outpatient Behavioral Health Clinic. *Psychiatric Services*, 70(2):130–134, December 2018. ISSN 1075-2730. doi: 10/drvz.

Crystel M. Gijsberts, Karlijn A. Groenewegen, Imo E. Hoefer, Marinus J. C. Eijkemans, Folkert W. Asselbergs, Todd J. Anderson, Annie R. Britton, Jacqueline M. Dekker, Gunnar Engström, Greg W. Evans, Jacqueline de Graaf, Diederick E. Grobbee, Bo Hedblad, Suzanne Holewijn, Ai Ikeda, Kazuo Kitagawa, Akihiko Kitamura, Dominique P. V. de Kleijn, Eva M. Lonn, Matthias W. Lorenz, Ellisiv B. Mathiesen, Giel Nijpels, Shuhei Okazaki, Daniel H. O'Leary, Gerard Pasterkamp, Sanne A. E. Peters, Joseph F. Polak, Jacqueline F. Price, Christine Robertson, Christopher M. Rembold, Maria Rosvall, Tatjana Rundek, Jukka T. Salonen, Matthias Sitzer, Coen D. A. Stehouwer, Michiel L. Bots, and Hester M. den Ruijter. Race/Ethnic Differences in the Associations of the Framingham Risk Factors with Carotid IMT and Cardiovascular Events. *PloS One*, 10(7):e0132321, 2015. ISSN 1932-6203. doi: 10/f3nj9f.

Mark L. Graber. The incidence of diagnostic error in medicine. *BMJ Quality & Safety*, 22(Suppl 2):ii21–ii27, October 2013. ISSN 2044-5415, 2044-5423. doi: 10/f5db55.

Felix Greaves, Indra Joshi, Mark Campbell, Samantha Roberts, Neelam Patel, and John Powell. What is an appropriate level of evidence for a digital health intervention? *The Lancet*, 392(10165):2665–2667, December 2018. ISSN 01406736. doi: 10/gfpfnn.

Ben Green and Yiling Chen. Disparate Interactions: An Algorithm-in-the-Loop Analysis of Fairness in Risk Assessments. In *Proceedings of the Conference on Fairness, Accountability, and Transparency - FAT* '19*, pages 90–99, Atlanta, GA, USA, 2019. ACM Press. ISBN 978-1-4503-6125-5. doi: 10/gftmqh.

Alan G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971. ISSN 0006-3444, 1464-3510. doi: 10/fj27m2.

William H. Herman and Robert M. Cohen. Racial and Ethnic Differences in the Relationship between HbA1c and Blood Glucose: Implications for the Diagnosis of Diabetes. *The Journal of Clinical Endocrinology & Metabolism*, 97(4):1067–1072, April 2012. ISSN 0021-972X, 1945-7197. doi: 10.1210/jc.2011-1894.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning Distributed Representations of Sentences from Unlabelled Data. *arXiv:1602.03483 [cs]*, February 2016.

Andreas Holzinger, Chris Biemann, Constantinos S. Pattichis, and Douglas B. Kell. What do we need to build explainable AI systems for the medical domain? *arXiv:1712.09923 [cs, stat]*, December 2017.

Kazi T. Islam, Christian R. Shelton, Juan I. Casse, and Randall Wetzel. Marked Point Process for Severity of Illness Assessment. In Finale Doshi-Velez, Jim Fackler, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 2nd Machine Learning for Healthcare Conference*, volume 68 of *Proceedings of Machine Learning Research*, pages 255–270, Boston, Massachusetts, August 2017. PMLR.

Gaurav Jetley and He Zhang. Electronic health records in IS research: Quality issues, essential thresholds and remedial actions. *Decision Support Systems*, 126:113137, November 2019. ISSN 0167-9236. doi: 10/ggfwd5.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

Tamara Jugov and Lea Ypi. Structural Injustice, Epistemic Opacity, and the Responsibilities of the Oppressed. *Journal of Social Philosophy*, 50(1):7–27, 2019. ISSN 1467-9833. doi: 10/ggx48b.

Shona Kalkman, Menno Mostert, Christoph Gerlinger, Johannes J. M. van Delden, and Ghislaine J. M. W. van Thiel. Responsible data sharing in international health research: A systematic review of principles and norms. *BMC Medical Ethics*, 20(1):21, March 2019. ISSN 1472-6939. doi: 10/ggx47x.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-Thought Vectors. In C. Cortes, N. D.

Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3294–3302. Curran Associates, Inc., 2015.

E. Kleinpeter. Four Ethical Issues of "E-Health". *IRBM*, 38(5):245–249, October 2017. ISSN 1959-0318. doi: 10/ggx479.

Matthieu Komorowski. Clinical management of sepsis can be improved by artificial intelligence: yes. *Intensive Care Medicine*, 46(2):375–377, December 2019. doi: 10.1007/s00134-019-05898-2. URL https://doi.org/10.1007/s00134-019-05898-2.

Yikuan Li, Shishir Rao, Jose Roberto Ayala Solares, Abdelaali Hassaine, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. BEHRT: Transformer for Electronic Health Records. *arXiv:1907.09538 [cs, stat]*, July 2019.

Li Shanshan, Fonarow Gregg C., Mukamal Kenneth J., Liang Li, Schulte Phillip J., Smith Eric E., DeVore Adam, Hernandez Adrian F., Peterson Eric D., and Bhatt Deepak L. Sex and Race/Ethnicity–Related Disparities in Care and Outcomes After Hospitalization for Coronary Artery Disease Among Older Adults. *Circulation: Cardiovascular Quality and Outcomes*, 9(2_suppl_1):S36–S44, February 2016. doi: 10/ggx476.

Wenzhao Lian, Ricardo Henao, Vinayak Rao, Joseph Lucas, and Lawrence Carin. A multitask point process predictive model. In *International Conference on Machine Learning*, pages 2030–2038, 2015.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.

Hongyuan Mei and Jason M Eisner. The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6754–6764. Curran Associates, Inc., 2017.

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, February 2019. ISSN 00043702. doi: 10/gfwcxw.

Brent Daniel Mittelstadt, Patrick Allo, Mariarosaria Taddeo, Sandra Wachter, and Luciano Floridi. The ethics of algorithms: Mapping the debate. *Big Data & Society*, 3(2):205395171667967, December 2016. ISSN 2053-9517, 2053-9517. doi: 10/gcdx92.

Jessica Morley and Luciano Floridi. Enabling digital health companionship is better than empowerment. *The Lancet Digital Health*, 1(4):e155–e156, August 2019a. ISSN 2589-7500. doi: 10/ggx478.

Jessica Morley and Luciano Floridi. The Limits of Empowerment: How to Reframe the Role of mHealth Tools in the Healthcare Ecosystem. *Science and Engineering Ethics*, June 2019b. ISSN 1471-5546. doi: 10/gf3w62.

Jessica Morley and Luciano Floridi. An ethically mindful approach to AI for health care. *The Lancet*, 395(10220):254–255, January 2020. ISSN 01406736. doi: 10/ggx592.

Jessica Morley, Caio Machado, Christopher Burr, Josh Cowls, Mariarosaria Taddeo, and Luciano Floridi. The Debate on the Ethics of AI in Health Care: A Reconstruction and Critical Review. SSRN

Scholarly Paper ID 3486518, Social Science Research Network, Rochester, NY, November 2019.

Maximilian Nickel and Matthew Le. Learning Multivariate Hawkes Processes at Scale. *arXiv:2002.12501 [cs, stat]*, February 2020.

Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, October 2019. ISSN 0036-8075, 1095-9203. doi: 10/ggchqk.

Office for National Statistics. Avoidable mortality in the UK, 2017.

Takahiro Omi, naonori ueda, and Kazuyuki Aihara. Fully Neural Network based Model for General Temporal Point Processes. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d\ textquotesingle Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 2122–2132. Curran Associates, Inc., 2019.

J. Marc Overhage and David McCallie. Physician time spent using the electronic health record during outpatient encounters. *Annals of Internal Medicine*, 172 (3):169, January 2020. doi: 10.7326/m18-3684. URL https://doi.org/10.7326/m18-3684.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché-Buc, Emily Fox, and Hugo Larochelle. Improving Reproducibility in Machine Learning Research (A Report from the NeurIPS 2019 Reproducibility Program). *arXiv:2003.12206 [cs, stat]*, April 2020.

Wullianallur Raghupathi and Viju Raghupathi. An Empirical Study of Chronic Diseases in the United States: A Visual Analytics Approach to Public Health. *International Journal of Environmental Research and Public Health*, 15(3), March 2018. ISSN 1661-7827. doi: 10/gdc634.

Shauli Ravfogel, Yoav Goldberg, and Tal Linzen. Studying the Inductive Biases of RNNs with Synthetic Variations of Natural Languages. *arXiv:1903.06400 [cs]*, March 2019.

Jose F. Rodrigues-Jr, Gabriel Spadon, Bruno Brandoli, and Sihem Amer-Yahia. Patient trajectory prediction in the Mimic-III dataset, challenges and pitfalls. *arXiv:1909.04605 [cs, stat]*, September 2019.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5): 206–215, May 2019. ISSN 2522-5839. doi: 10/gf4tp9.

Wojciech Samek and Klaus-Robert Müller. Towards explainable artificial intelligence. In *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 5–22. Springer, 2019.

Sofia Serrano and Noah A. Smith. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1282. URL https://www.aclweb.org/anthology/P19-1282.

Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. Intensity-Free Learning of Temporal Point Processes. *arXiv:1909.12127 [cs, stat]*, January 2020.

Benjamin Shickel, Patrick Tighe, Azra Bihorac, and Parisa Rashidi. Deep EHR:

A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis. *IEEE Journal of Biomedical and Health Informatics*, 22(5): 1589–1604, September 2018. ISSN 2168-2194, 2168-2208. doi: 10.1109/JBHI.2017.2767063.

Joseph Sill. Monotonic Networks. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 661–667. MIT Press, 1998.

Harini Suresh and John V. Guttag. A Framework for Understanding Unintended Consequences of Machine Learning. *arXiv:1901.10002 [cs, stat]*, February 2020.

Eric J. Topol. High-performance medicine: The convergence of human and artificial intelligence. *Nature Medicine*, 25(1):44–56, January 2019. ISSN 1546-170X. doi: 10/gfsvzn.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv:1706.03762 [cs]*, December 2017.

Jason Walonoski, Mark Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kudakwashe Dube, Thomas Gallagher, and Scott McLachlan. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association*, 25(3):230–238, March 2018. ISSN 1067-5027. doi: 10/gb4mkg.

Jeremy C Weiss and David Page. Forest-based point process for event prediction from electronic health records. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 547–562. Springer, 2013.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On Layer Normalization in the Transformer Architecture. *arXiv:2002.04745 [cs, stat]*, February 2020.

Wei Zhang, Zhaobin Kuang, Peggy Peissig, and David Page. Adverse drug reaction discovery from electronic health records with deep neural networks. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 30–39, 2020.

Shixiang Zhu, Minghe Zhang, Ruyi Ding, and Yao Xie. Deep Attention Spatio-Temporal Point Processes. *arXiv:2002.07281 [cs, stat]*, February 2020.

Appendix A. Positive monotonic building blocks

As discussed in Section 4.2.3, if $f(t)$ is given by a L-layer NN, where the output of each layer f_i is fed into the next f_{i+1}, then we can write $f(t) = (f_L \circ \cdots \circ f_2 \circ f_1)(t)$. Application of the chain rule

$$\frac{df(t)}{dt} = \frac{df_L}{df_{L-1}} \frac{df_{L-1}}{df_{L-2}} \cdots \frac{df_2}{df_1} \frac{df_1}{dt} \tag{8}$$

shows that a sufficient solution to achieving $df(t)/dt \geq 0$ is to enforce each step of processing to be a monotonic function of its input, i.e. $df_i/df_{i-1} \geq 0$ and $df_1/dt \geq 0$ implies $f(t)$ is a monotonic function of t.

As well as the monotonicity constraint, as we are interested in intensities that can decay as well as increase, it needs to be possible

that the second derivative of $f(t)$ can be negative

$$\frac{d\lambda(t|\mathcal{H}_t)}{dt} \sim \frac{d^2 f(t|\mathcal{H}_t)}{dt^2} < 0 \tag{9}$$

for some history $\mathcal{H}_t$.

Here we collect all of the modifications we employ to ensure our architectures comply with this restriction.

A.1. Linear projections

Linear projections are a core component of dense layers, and can be made monotonic by requiring every element of the projection matrix is at least 0

$$f(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x}, \tag{10}$$

$$df(\boldsymbol{x})_i/dx_j = W_{i,j} \geq 0 \; \forall \; i, j. \tag{11}$$

One solution to this problem is to parameterise the $\boldsymbol{W}$ as a positive g transformation of some auxiliary parameters $\boldsymbol{V}$ such that $\boldsymbol{W} = g(\boldsymbol{V}) \geq 0$. Taking g as Rectified Linear Unit (ReLU) leads to issue during training: any weight which reaches zero will become frozen no longer be updated by the network, harming convergence properties. We also experimented using softplus and sigmoid activations. These have the issue where the auxiliary weights $\boldsymbol{V}$ can be pushed arbitrarily negative and, when the network needs them again, training needs to pull them very far back for little change in $\boldsymbol{W}$, again harming convergence. This approach is also burdened with additional computational cost.

The most effective approach, which we employ in our models, involves modifying the forward pass such that any weights below some small $\epsilon > 0$ are set to ϵ. This ϵ can be thought of as a lower bound on the derivative, has no computational overheads, and none of the issues discussed in the approaches above. In our experiments we take $\epsilon = 10^{-30}$.

A.2. Dense activation functions

ReLU The widely used ReLU activation clearly satisfies the monotonicity constraint, however

$$\frac{d\,\mathrm{ReLU}(x)}{dx} = \begin{cases} 0 & \text{if} \quad x \leq 0 \\ 1 & \text{if} \quad x > 0, \end{cases}$$

$$\frac{d^2\,\mathrm{ReLU}(x)}{dx^2} = 0.$$

As the second derivative is 0 then

$$\frac{d\lambda^*_m(t)}{dt} = 0 \tag{12}$$

which means that any cumulative model using the ReLU activation is equivalent to the conditional Poisson process $\boldsymbol{\lambda}(\tau|\mathcal{H}_t) = \boldsymbol{\mu}(\mathcal{H}_t)$ introduced in Section 4.2.3. We cannot use ReLU in a cumulative model.

Tanh Omi et al. (2019) proposed to use the tanh activation function which doesn't have the constraints of ReLU:

$$\frac{d\tanh(x)}{dx} = \mathrm{sech}^2(x) \in (0, 1),$$

$$\frac{d^2\tanh(x)}{dx^2} = 2\tanh(x)\,\mathrm{sech}^2(x) \in (c_-, c_+)$$

where $c_\pm = \log(2 \pm \sqrt{2})/2$, so tanh meets our requirements of a positive first derivative and a non-zero second derivative.

Gumbel An alternative to tanh is the adaptive Gumbel activation introduced in Farhadi et al. (2019):

$$\sigma(x_m) = 1 - [1 + s_m \exp(x_m)]^{-\frac{1}{s_m}} \tag{13}$$

where $\forall m : s_m \in \mathbb{R}_{>0}$, and m is a dimension/activation index. For brevity, we will refer to this activation function as the Gumbel activation, and while its analytic properties we will drop the dimension index m, however, since the activation is applied element-wise, the analytic properties discussed directly transfer to the vector application case.

Its first and second derivatives match our positivity and negative requirements

$$\frac{d\sigma(x)}{dx} = \exp(x) \times [1 + s\exp(x)]^{-\frac{s+1}{s}} \in (0, 1/e), \tag{14}$$

$$\frac{d^2\sigma(x)}{dx^2} = \exp(x)\,[1 - \exp(x)] \times [1 + s\exp(x)]^{-\frac{2s+1}{s}} \in (c_-, c_+), \tag{15}$$

where $c_\pm = (\pm\sqrt{5}-2)\exp[(\pm\sqrt{5}-3)/2]$. The Gumbel activation shares many of the limiting properties of the tanh activation.

$$\lim_{x\to-\infty} \sigma(x) = 0, \tag{16}$$

$$\lim_{x\to\infty} \sigma(x) = 1, \tag{17}$$

$$\lim_{x\to\pm\infty} \frac{d\sigma(x)}{dx} = \lim_{x\to\pm\infty} \frac{d^2\sigma(x)}{dx^2} = 0. \tag{18}$$

For our purposes, the key advantage of the Gumbel over tanh are the learnable parameters s_m. These parameters control the magnitude of the gradient through equation Equation (14) and the magnitude of the second derivative through Equation (15). The maximum value of the first derivative is obtained at $x = 0$, which corresponds to the mode of the Gumbel distribution, and for any value of s

$$\max_x \frac{d\sigma(x)}{dx} = (1 + s)^{-\frac{2s+1}{s}}. \tag{19}$$

By sending $s \to 0$ we get the largest maximum gradient of $1/e$ which occurs in a short window in x, and by sending $s \to \infty$ we get the smallest maximum gradient of 0, which occurs for all x. This allows the activation function to control its sensitivity changes in the input, and allows the NN to be selective in where it wants fine-grained control over its first and second derivatives gradient (i.e. produce therefore have an output that changes slowly over a large range in the input values), and where it needs the gradient to be very large in a small region of the input.

These properties are extremely beneficial for cumulative modelling, and we employ the Gumbel activation in all of these models.

Gumbel-Softplus Although tanh and adaptive Gumbel meet our positivity and negativity requirements, they share a further issue raised by Shchur et al. (2020) in that their saturation ($\lim_{x\to\infty}\sigma(x) = 1$) does not allow the property $\lim_{t\to\infty}\Lambda^*_m(t) = \infty$, leading to an ill-defined joint probability density $p^*_m(t)$. To solve this, we introduce the Gumbel-Softplus activation: [9]

$$\sigma(x_m) = \text{gumbel}(x_m)\,[1 + \text{softplus}(x_m)]\,, \tag{20}$$

where gumbel is defined in Equation (13), and softplus is the parametric softplus function:

$$\text{softplus}(x_m) = \frac{\log\left(1 + s_m\,\exp(x_m)\right)}{s_m} \tag{21}$$

This activation function has the property $\lim_{t\to\infty}\sigma(t) = \infty$, in addition to satisfying the positivity and negativity constraints.

A.3. Attention activation functions

In addition to the activation functions used in the dense layers of NNs, the attention block requires another type of activation. This is indicated in Equation (63), as the attention logits $E_{i,j}$ are passed through a function g to produce the attention coefficients $\alpha_{i,j}$. Generally, this activation function is the Softmax function (see Equation (64)), however, this function is not monotonic:

$$\frac{\partial\,\text{softmax}(x_i)}{\partial x_j} = \text{softmax}(x_i) * [\delta_{ij} - \text{softmax}(x_j)] < 0\ \forall\ i \neq j, \tag{22}$$

9. This modification could equally be applied to the tanh activation.

where δ_{ij} is the Kronecker delta function.

Consequently, we chose to use the sigmoid activation function instead of the softmax when the modelled function required to be monotonic. Indeed, the sigmoid function is monotonic:

$$\frac{\partial \sigma(x)}{\partial x} = \sigma(x)\left[1 - \sigma(x)\right] > 0. \tag{23}$$

The softmax activation function has the nice property of making the attention coefficient sum to one, therefore forcing the network to attend to a limited number of points. This can make the interpretation of the attention coefficient easier. However, using a softmax could potentially lead to a decrease of performance if the points shouldn't strictly compete for contribution to the intensity under the generating process. As a result, we believe that choosing a sigmoid or a softmax is not straightforward. We use sigmoid for cumulative approximators, and softmax everywhere else.

A.4. Layer normalisation

The final layer of the Transformer requiring modification issue is layer normalisation, a layer which dramatically improves the convergence speed of these models.

Modifying this layer requires realising that any layer requiring the division by the sum of activations (for example, even L_2 normalisation) will result in a negative derivative occurring somewhere. Indeed, this is the underlying reason for the negative elements of the softmax Jacobian in Equation (22).

It follows that any kind of normalisation cannot explicitly depend on the current set of activations. Taking inspiration from batch normalisation, we construct an exponential moving average form of Layer normalisation during training. When performing a forward pass, the means and standard deviations are taken from these moving averages and are treated as constants. After the forward pass, the means and standard deviations are then updated in a similar fashion to batch normalisation. Finally, taking the gain parameter positive as discussed in Appendix A.1 results in a monotonic form of layer normalisation.

We employ this form of layer normalisation in the cumulative self-attention decoder (SACM), otherwise we use the standard form.

A.5. Encoding

Temporal Encoding for monotonic approximators The temporal embedding of Equation (65) is not monotonic in t and therefore cannot be used in any conditional cumulative approximator. (Vaswani et al., 2017) noted that a learnable temporal encoding had similar performance to the one presented in Equation (66). In order to model the conditional cumulative intensity $\Lambda^*_m(t)$, we will instead use a MLP

$$\begin{aligned}\text{ParametricTemporal}(t) = \\ \text{MLP}(t; \boldsymbol{\theta}_{\geq \epsilon}) \in \mathbb{R}^{d_{\text{Model}}},\end{aligned} \tag{24}$$

where $\boldsymbol{\theta}_{\geq \epsilon}$ indicates that all projection matrices have positive values in all entries. Biases may be negative. If we choose monotonic activation functions for the MLP, then it is a monotonic function approximator (Sill, 1998).

The d_{Model} dimensional temporal encoding of an event at t_i with labels $\mathcal{M}_i$ is then

$$\boldsymbol{x}_i = \text{TemporalEncode}(t_i, \mathcal{M}_i) \tag{25}$$

$$= \boldsymbol{v}_i(\mathcal{M}_i)\sqrt{d_{\text{Model}}} + \text{ParametricTemporal}(t_i). \tag{26}$$

A.6. Double backward trick

When using a marked neural network based on the cumulative intensity, one issue comes from the fact that neural networks accumulate gradients. Specifically, the way that

autograd is implemented in commonly used deep learning frameworks means that it is not possible to compute $\frac{\partial}{\partial t}\Lambda^*_m(t)$ for a single m. The value of the derivative will always be accompanied by the derivatives of all m due to gradient accumulation, i.e. one can only compute the sum $\sum_{m=1}^{M}\frac{\partial}{\partial t}\Lambda^*_m(t)$ but not an individual $\frac{\partial}{\partial t}\Lambda^*_m(t)$.

A way of solving this issue is to split $\Lambda^*_m(t)$ into individual M components, and to compute the gradient for each of them. However, this method is not applicable to high values of M due to computational overhead. Another way to solve this issue is to use the *double backward trick* which allows to compute this gradient without having to split the cumulative intensity.

This trick is based on the following: we define a $B \times L \times M$ (Batch size $\times$ Sequence length $\times$ # classes) tensor $\boldsymbol{a}$ filled with zeros, and compute the Jacobian vector product

$$\mathbf{jvp} = \left(\frac{\partial}{\partial t}\Lambda^*_m(t)\right)^T \boldsymbol{a} \in \mathbb{R}^{B\times L}. \tag{27}$$

We then define second tensor $\boldsymbol{b}$ of shape $B \times L \times M$, filled with ones, and we compute the following Jacobian vector product:

$$\left(\frac{\partial}{\partial \boldsymbol{a}}\mathbf{jvp}\right)^T = \left\{\frac{\partial}{\partial \boldsymbol{a}}\left[\left(\frac{\partial}{\partial t}\Lambda^*_m(t)\right)^T \mathbf{a}\right]\right\}^T \tag{28}$$

$$\left(\frac{\partial}{\partial \boldsymbol{a}}\mathbf{jvp}\right)^T \boldsymbol{b} = \left(\frac{\partial}{\partial t}\Lambda^*_m(t)\right)^T \boldsymbol{b} \tag{29}$$

$$= \frac{\partial}{\partial t}\Lambda^*_m(t) \in \mathbb{R}^{B\times L\times M} \tag{30}$$

This recovers the element-wise derivative of the cumulative intensity function as desired, with the number of derivative operations performed being independent of the number of labels M.

A.7. Modelling the log cumulative intensity

We also investigated an alternative which is to let the decoder directly approximate $\log \Lambda^*_m(t)$. We have then:

$$\frac{\partial \log \Lambda^*_m(t)}{\partial t} = \frac{\lambda^*_m(t)}{\Lambda^*_m(t)}, \tag{31}$$

$$\log \lambda^*_m(t) = \log \Lambda^*_m(t) + \log\left(\frac{\partial \log \Lambda^*_m(t)}{\partial t}\right). \tag{32}$$

While we didn't use this method to produce our results, we implemented it in our codebase. The advantage of this method is that the log intensity is directly modelled by the network, and its disadvantage is that the subtraction of the exponential terms to compute $\Lambda^*_m(t) - \Lambda^*_m(t_i)$ can lead to numerical instability.

Appendix B. Taxonomy

We ran our experiments using 2 encoders: SA and GRU, and 7 decoders: CP, RMTPP, LNM, MLP-CM, MLP-MC, Attn-CM, Attn-MM. We combined these encoders and decoders to form models, which are defined by linking an encoder with a decoder: for instance, GRU-MLP-CM denotes a GRU encoder with a MLP-CM (for cumulative) decoder. We present here these different components.

For the implementations of these models, as well as instances trained on the tasks in our setup, please refer to code base supplied in the supplementary material.

Label embeddings Although Neural TPP encoders differ in how they encode temporal information, they share a label embedding step. Given labels $m \in \mathcal{M}_i$ localised at time t_i, the D_{Emb} dimensional

embedding $\boldsymbol{v}_i$ is

$$\boldsymbol{v}_i = f_{\text{pool}}\left(\mathcal{W}_i = \{\boldsymbol{w}^{(m)} | m \in \mathcal{M}_i\}\right) \in \mathbb{R}^{D_{\text{Emb}}} \tag{33}$$

where $\boldsymbol{w}^{(m)}$ is the learnable embedding for class m, and $f_{\text{pool}}(\mathcal{W})$ is a pooling function, e.g. mean pooling: $f_{\text{pool}}(\mathcal{W}) = \sum_{\boldsymbol{w}\in\mathcal{W}} \boldsymbol{w}/|\mathcal{W}|$, or max pooling: $f_{\text{pool}}(\mathcal{W}) = \oplus_{\alpha=1}^{D_{\text{Emb}}} \max(\{w_\alpha | \boldsymbol{w} \in \mathcal{W}\})$. [10]

B.1. Encoders

GRU We follow the default equations of the GRU NN:

$$\begin{aligned}\boldsymbol{r}_i &= \sigma(\boldsymbol{W}^{(1)}\boldsymbol{x}_i + \boldsymbol{b}^{(1)} + \\ &\qquad \boldsymbol{W}^{(2)}\boldsymbol{h}_{(i-1)} + \boldsymbol{b}^{(2)}) \end{aligned} \tag{34}$$

$$\begin{aligned}\boldsymbol{z}_i &= \sigma(\boldsymbol{W}^{(3)}\boldsymbol{x}_t + \boldsymbol{b}^{(3)} + \\ &\qquad \boldsymbol{W}^{(4)}\boldsymbol{h}_{(i-1)} + \boldsymbol{b}^{(4)}) \end{aligned} \tag{35}$$

$$\begin{aligned}\boldsymbol{n}_i &= \tanh(\boldsymbol{W}^{(5)}\boldsymbol{x}_i + \boldsymbol{b}^{(5)} + \\ &\qquad \boldsymbol{r}_i \circ (\boldsymbol{W}^{(6)}\boldsymbol{h}_{(i-1)} + \boldsymbol{b}^{(6)})) \end{aligned} \tag{36}$$

$$\boldsymbol{h}_i = (1 - \boldsymbol{z}_i) \circ \boldsymbol{n}_i + \boldsymbol{z}_i \circ \boldsymbol{h}_{(i-1)}, \tag{37}$$

where $\boldsymbol{h}_t$, $\boldsymbol{r}_t$, $\boldsymbol{z}_t$ and $\boldsymbol{n}_t$ are the hidden state, and the reset, update and new gates, respectively, at time t. $\boldsymbol{x}_t$ is the input at time t, defined by Equation (65). σ designs the sigmoid function and $\circ$ is the Hadamard product.

Self-attention (SA) We follow Equation (63) and Equation (64) to form our attention block. The keys, queries and values are linear projections of the input: $\boldsymbol{q} = \boldsymbol{W}_q\boldsymbol{x}$, $\boldsymbol{k} = \boldsymbol{W}_k\boldsymbol{x}$, $\boldsymbol{v} = \boldsymbol{W}_v\boldsymbol{x}$, with the input $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \}$, with each x_i defined by Equation (66). We also apply a Layer Normalisation layer before each attention and feedforward block as in Xiong et al. (2020):

$$\mathcal{Q}' = \text{LayerNorm}(\mathcal{Q}), \tag{38}$$

$$\mathcal{Q}' = \text{MultiHeadAttn}(\mathcal{Q}', \mathcal{V}) \tag{39}$$

$$\mathcal{Q}' = \mathcal{Q}' + \mathcal{Q} \tag{40}$$

$$\mathcal{Z} = \text{LayerNorm}(\mathcal{Q}') \tag{41}$$

$$\mathcal{Z} = \{\boldsymbol{W}^{(2)}\text{ReLU}(\boldsymbol{W}^{(1)}\boldsymbol{z}_i + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}\} \tag{42}$$

$$\mathcal{Z} = \mathcal{Z} + \mathcal{Q}', \tag{43}$$

where MultiHeadAttn is defined by Equation (67). We summarise these equations by:

$$\mathcal{Z} = \text{Attn}(\mathcal{Q} = \mathcal{X}, \mathcal{V} = \mathcal{X}) \equiv \text{SA}(\mathcal{X}). \tag{44}$$

B.2. Decoders

For every decoder, we use when appropriate the same MLP, composed of two layers: one from D_{Hid} to D_{Hid}, and another from D_{Hid} to M, where D_{Hid} is either 8, 32 and 64 depending on the dataset, and M is the number of marks.

We also define the following terms for every decoder: t, a query time, $\boldsymbol{z}_t \equiv \boldsymbol{z}_{|\mathcal{H}_t|}$, the representation for the latest event in $\mathcal{H}_t$, i.e. the past event closest in time to t, and t_i, the time of the previous event. In addition, we define $\boldsymbol{q}_t$, the representation of the query time t. For the cumulative models, $\boldsymbol{q}_t = \text{ParametricTemporal}(t)$, and for the Monte-Carlo models, $\boldsymbol{q}_t = \text{Temporal}(t)$, following Equation (24) and Equation (65), respectively.

Conditional Poisson (CP) The conditional Poisson decoder returns:

$$\lambda_m^*(t) = \text{MLP}(\boldsymbol{z}_t), \tag{45}$$

$$\Lambda_m^*(t) = \text{MLP}(\boldsymbol{z}_t)(t - t_i), \tag{46}$$

where the MLP is the same in both equations.

10. In the multi-class setting, only one label appears at each time t_i, and so $\boldsymbol{v}_i$ is directly the embedding for that label, and pooling has no effect.

RMTPP Given the same elements as for the conditional Poisson, the RMTPP returns:

$$\lambda_m^*(t) = \exp\left[\boldsymbol{W}^{(1)}\boldsymbol{z}_t + w^{(2)}(t-t_i) + \boldsymbol{b}^{(1)}\right]_m \tag{47}$$

$$\Lambda_m^*(t) = \frac{1}{w^{(2)}}\Big[\exp(\boldsymbol{W}^{(1)}\boldsymbol{z}_t + \boldsymbol{b}^{(1)}) - \tag{48}$$

$$\exp(\boldsymbol{W}^{(1)}\boldsymbol{z}_t + w^{(2)}(t-t_i) + \boldsymbol{b}^{(1)})\Big]_m \tag{49}$$

where $\boldsymbol{W}^{(1)} \in \mathbb{R}^{D_{\text{Hid}} \times M}$, $w^{(2)} \in \mathbb{R}$, and $\boldsymbol{b}^{(1)} \in \mathbb{R}^M$, are learnable parameters.

Log-normal mixture (LNM) Following Shchur et al. (2020), the log-normal mixture model returns:

$$\bar{p}^*(t) = \sum_{k=1}^{K} w_k \frac{1}{(t-t_i)\,\sigma_k\sqrt{2\pi}} \exp\left[-\frac{(\log(t-t_i)-\mu_k)^2}{2\sigma_k^2}\right], \tag{50}$$

where $\bar{p}_m^*(t) = \bar{p}^*(t)\,\rho_m(\mathcal{H}_t)$, and $\boldsymbol{w}, \boldsymbol{\sigma}, \boldsymbol{\mu} \in \mathbb{R}^K$ are mixture weights, distribution means and standard deviations, and are outputs of the encoder. These parameters are defined by:

$$\boldsymbol{w} = \text{softmax}(\boldsymbol{W}^{(1)}\boldsymbol{z}_t + \boldsymbol{b}^{(1)}), \tag{51}$$

$$\boldsymbol{\sigma} = \exp(\boldsymbol{W}^{(2)}\boldsymbol{z}_t + \boldsymbol{b}^{(2)}), \tag{52}$$

$$\boldsymbol{\mu} = \boldsymbol{W}^{(3)}\boldsymbol{z}_t + \boldsymbol{b}^{(3)}. \tag{53}$$

MLP Cumulative (MLP-CM) The MLP-CM returns:

$$\lambda_m^*(t) = \frac{\partial \Lambda_m^*(t)}{\partial t}, \tag{54}$$

$$\Lambda_m^*(t) = \text{MLP}([\boldsymbol{q}_t, \boldsymbol{z}_t]; \boldsymbol{\theta}_{\geq \epsilon}), \tag{55}$$

where the square brackets indicate a concatenation.

MLP Monte Carlo (MLP-MC) The MLP-MC returns:

$$\lambda_m^*(t) = \text{MLP}([\boldsymbol{q}_t, \boldsymbol{z}_t]), \tag{56}$$

$$\Lambda_m^*(t) = \text{MC}\left[\lambda_m^*(t'), t', t_i, t\right], \tag{57}$$

where MC represents the estimation of the integral $\int_{t_i}^{t} \lambda_m^*(t')\,dt'$ using a Monte-Carlo sampling method.

Attention Cumulative (Attn-CM) For the Attn-CM and Attn-MC, the attention block differs from the SA encoder: the queries $\boldsymbol{W}_q\boldsymbol{q}$ are defined from the query representations, and the keys $\boldsymbol{W}_k\boldsymbol{z}$ and values $\boldsymbol{W}_v\boldsymbol{z}$ are defined from the encoder representations.

The Attn-CM returns:

$$\lambda_m^*(t) = \frac{\partial \Lambda_m^*(t)}{\partial t}, \tag{58}$$

$$\Lambda_m^*(t) = \text{Attn}(\mathcal{Q} = \{\boldsymbol{q}_t\}, \mathcal{V} = \mathcal{Z}), \tag{59}$$

where Transformer refers to Equation (44), with $\boldsymbol{x} = \boldsymbol{q}$ as input. Moreover, the components of this Attention are modified following Appendix A. In particular the ReLU activation is replaced by a Gumbel.

Attention Monte Carlo (Attn-MC) The Attn-MC returns:

$$\lambda_m^*(t) = \text{Attn}(\mathcal{Q} = \{\boldsymbol{q}_t\}, \mathcal{V} = \mathcal{Z}), \tag{60}$$

$$\Lambda_m^*(t) = \text{MC}\left[\lambda_m^*(t'), t', t_i, t\right], \tag{61}$$

where MC represents the estimation of the integral $\int_{t_i}^{t} \lambda_m^*(t')\,dt'$ using a Monte-Carlo sampling method.

Appendix C. Additional results

C.1. Hawkes datasets

The parameters of our Hawkes datasets are:

$$\begin{aligned}
&\text{Independent}: \\
&\mu = \begin{bmatrix} 0.1 \\ 0.05 \end{bmatrix}, \\
&\alpha = \begin{bmatrix} 0.2 & 0.0 \\ 0.0 & 0.4 \end{bmatrix}, \\
&\beta = \begin{bmatrix} 1.0 & 1.0 \\ 1.0 & 1.0 \end{bmatrix}, \\
&\text{Dependent}: \\
&\mu = \begin{bmatrix} 0.1 \\ 0.05 \end{bmatrix}, \\
&\alpha = \begin{bmatrix} 0.2 & 0.1 \\ 0.2 & 0.3 \end{bmatrix}, \\
&\beta = \begin{bmatrix} 1.0 & 1.0 \\ 1.0 & 2.0 \end{bmatrix}.
\end{aligned} \tag{62}$$

C.2. Hawkes results

A useful property of these datasets is that we can visually compare the modelled intensities with the intensity of the underlying generating process. On each datasets, Hawkes (independent) and Hawkes (dependent), all models perform similarly, with the exception of the conditional Poisson. We present results for the SA-MLP-MC on Figure 3.

C.3. Retweets results

On the Retweets dataset, while the conditional Poisson model significantly outperforms TPP models in terms of ROC-AUC, the opposite is true in terms of NLL/time. This is probably due to the fact that TPPs can model the intensity decay that occurs when a tweet is not retweeted over a certain period of time. We compare on Figure 4 the intensity functions of a SA-CP and and a SA-MLP-MC.

C.4. Attention coefficients

Figure 5 shows an example of attention coefficients for a sequence of Synthea (Full).

Appendix D. Transformer architecture

Attention block The key component of the Transformer is attention. This computes contextualised representations $\boldsymbol{q}_i' = \text{Attention}(\boldsymbol{q}_i, \{\boldsymbol{k}_j\}, \{\boldsymbol{v}_j\}) \in \mathbb{R}^{d_{\text{Model}}}$ of queries $\boldsymbol{q}_i \in \mathbb{R}^{d_{\text{Model}}}$ from linear combinations of values $\boldsymbol{v}_j \in \mathbb{R}^{d_{\text{Model}}}$ whose magnitude of contribution is governed by keys $\boldsymbol{k}_j \in \mathbb{R}^{d_{\text{Model}}}$

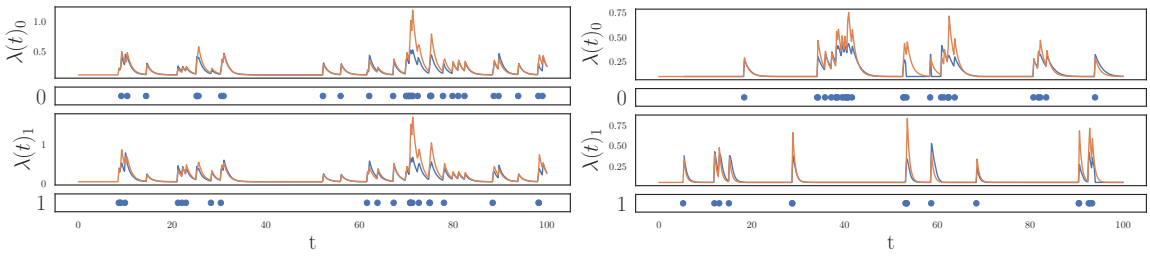
$$\begin{aligned}
\text{Attention}(\boldsymbol{q}_i, \{\boldsymbol{k}_j\}, \{\boldsymbol{v}_j\}) &= \sum_j \alpha_{i,j}\, \boldsymbol{v}_j, \\
\alpha_{i,j} &= g\left(E_{i,j}\right), \\
E_{i,j} &= \frac{\boldsymbol{q}_i^T \boldsymbol{k}_j}{\sqrt{d_k}}
\end{aligned} \tag{63}$$

where $\alpha_{i,j}$ are the attention coefficients, $E_{i,j}$ are the attention logits, and g is an activation function that is usually taken to be the softmax

$$\underset{j}{\text{softmax}}\left(E_{i,j}\right) = \frac{\exp(E_{i,j})}{\sum_k \exp(E_{i,k})}. \tag{64}$$

Masking The masking in our encoder is such that events only attend to themselves and events that happened strictly before them in time. The masking in the decoder is such that the query only attends to events occurring strictly before the query time.

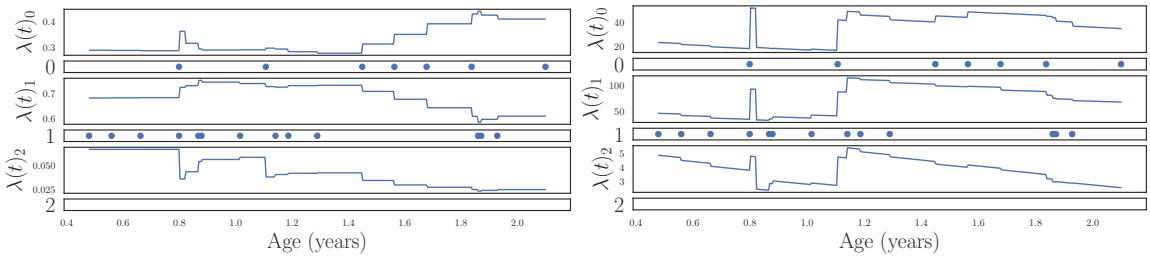
Temporal Encoding In the original transformer (Vaswani et al., 2017), absolute positional information is encoded into $\boldsymbol{q}_i$, $\boldsymbol{k}_j$ and $\boldsymbol{v}_j$. This was achieved by producing positional embeddings where the embeddings of relative positions were linearly related $\boldsymbol{x}(i) = \boldsymbol{R}(i-j)\,\boldsymbol{x}(j)$ for some rotation matrix $\boldsymbol{R}(i-j)$. Our temporal embedding



(*a*) SA-MLP-MC on Hawkes (dependent)

(*b*) SA-MLP-MC on Hawkes (independent)

Figure 3: Intensity functions on the Hawkes datasets. The orange line represents the true intensity of the sequence, while the blue line represents the intensity modelled by the SA-MLP-MC.



(*a*) SA-CP on Retweets data

(*b*) SA-MLP-MC on Retweets data

Figure 4: Intensity functions of a SA-CP and a SA-MLP-MC on the Retweets datasets. The MLP-MC is able to model the time decay of the intensity, contrary to the CP.

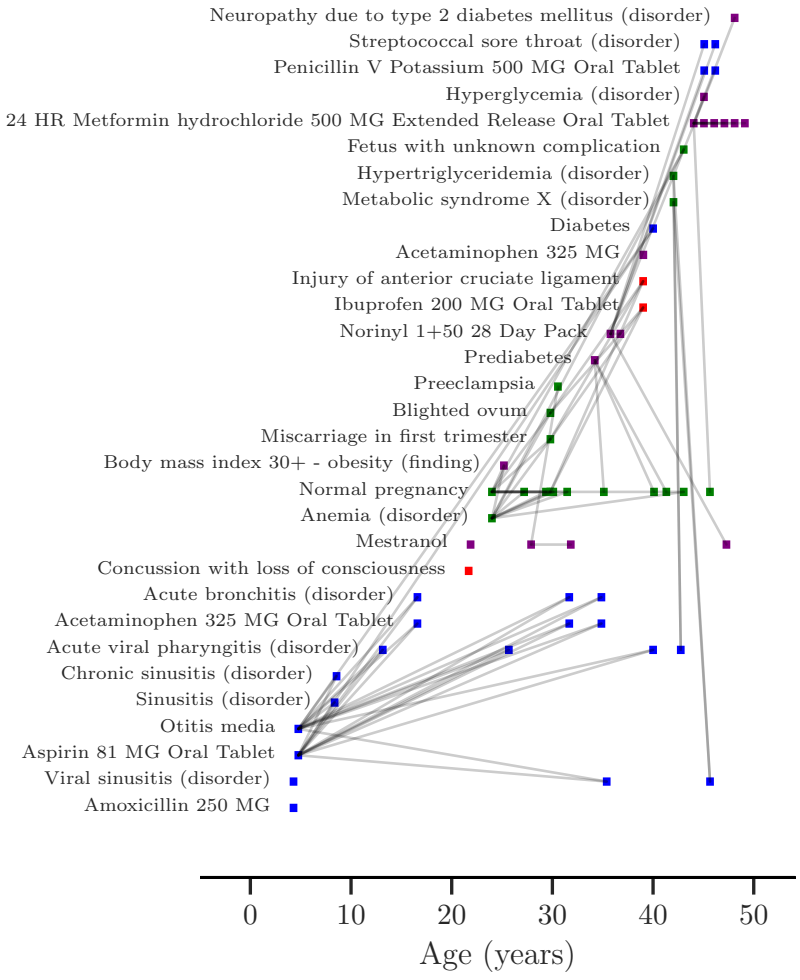
method generalises this to the continuous domain

$$\text{Temporal}(t) = \bigoplus_{k=0}^{d_{\text{Model}}/2-1} \sin(\alpha_k t) \oplus \cos(\alpha_k t) \in \mathbb{R}^{d_{\text{Model}}}, \tag{65}$$

where α_k is proportional to $10000^{-2k/d_{\text{Model}}}$, and β is a temporal rescaling parameter and plays the role of setting the shortest time scale the model is sensitive to. In practice we estimate $\hat{\beta} = \mathbb{E}[(w_+ - w_-)/N]$ from the training set, so that a TPP with smaller average gaps between events is modelled at a higher granularity by the encoding. We experimented with having β a learnable parameter of the model, however, this made the model extremely difficult to optimise. Note that $\hat{\beta} = 1$ for a language model. In addition, β does not change the relative frequency of the rotational subspaces in the temporal embedding from the form in Vaswani et al. (2017). The encoding with temporal information of an event at t_i with labels $\mathcal{M}_i$ is then

$$\boldsymbol{x}_i = \text{TemporalEncode}(t_i, \mathcal{M}_i) = \boldsymbol{v}_i(\mathcal{M}_i)\sqrt{d_{\text{Model}}} + \text{Temporal}(t_i) \in \mathbb{R}^{d_{\text{Model}}}. \tag{66}$$

Figure 5: Encoder attention coefficients for an EHR extracted from the Synthea (Full) dataset. Line thickness corresponds to attention strength, with values below 0.5 are omitted for clarity. As stressed in Serrano and Smith (2019), "while attention is by no means a fail-safe indicator", it is still able to "noisily predicts input components' overall importance to a model".



Multi-head attention Multi-head attention produces $\boldsymbol{q}_i'$ using H parallel attention layers (heads) in order to jointly attend to information from different subspaces at different positions

$$\text{MultiHead}(\boldsymbol{q}_i, \{\boldsymbol{k}_j\}, \{\boldsymbol{v}_j\}) = \boldsymbol{W}^{(o)} \left[\bigoplus_{h=1}^{H} \text{Attn}\Big(\boldsymbol{W}_h^{(q)} \boldsymbol{q}_i, \{\boldsymbol{W}_h^{(k)} \boldsymbol{k}_j\}, \{\boldsymbol{W}_h^{(v)} \boldsymbol{v}_j\} \Big) \right], \quad (67)$$

where $\boldsymbol{W}_h^{(q)}, \boldsymbol{W}_h^{(k)} \in \mathbb{R}^{d_k \times d_{\text{Model}}}$, $\boldsymbol{W}_h^{(v)} \in \mathbb{R}^{d_v \times d_{\text{Model}}}$, and $\boldsymbol{W}^{(o)} \in \mathbb{R}^{d_{\text{Model}} \times h\, d_v}$ are learnable projections.

Appendix E. Broader impact statement

E.1. Overview

In this paper, we demonstrate that Temporal Point Processes perform favourably in modelling Electronic Health Records. We embarked on this project aware that as high impact systems, Clinical Decision Support tools must ultimately provide explanations (Holzinger et al., 2017) to ensure clinician adoption (Miller, 2019) and provide accountability (Mittelstadt et al., 2016) whilst being mindful of the call for researchers to build interpretable ML models (Rudin, 2019). Our results demonstrate that our attention-based model performs favourably against less interpretable models. As attention could carry medically interpretable information our innovation contributes towards the development of medically useful AI tools.

Aware of both the need for transparency in the research community (Pineau et al., 2020) and for sensitive handling of health data (Kalkman et al., 2019), we opted to use open source synthetic EHRs. We provide all trained models, benchmarked datasets and a high-quality deterministic codebase to allow others to easily implement and benchmark their own models. We also make the important contribution of highlighting existing benchmark datasets as inappropriate; continuing to develop temporal models using these datasets may slow development of useful technologies or cause poor outputs if applied to EHR data.

Turning to the potential impact of our technology, we begin by briefly discussing the benefits temporal EHR models could bring to healthcare before, with no less objectivity, analysing harms which could occur, raising questions for further reflection.

E.2. The potential benefits of EHR modelling

Healthcare systems today are under intense pressure. Chronic disease prevalence is rising (Raghupathi and Raghupathi, 2018), costs of care are increasing, resources are constrained and outcomes are worsening (Topol, 2019). 23% of UK deaths in 2017 were classed as avoidable (Office for National Statistics, 2017), clinician diagnostic error is estimated at 10-15% (Graber, 2013) and EHR data suffers from poor coding and incompleteness which affects downstream tasks (Jetley and Zhang, 2019). Our research contributes techniques which could be used to:

- More accurately impute missing EHR data.
- Identify likely misdiagnoses.
- Predict future health outcomes.
- Identify higher risk patients for successful health interventions.
- Design optimal care pathways.
- Optimise resource management.
- Identify previously undiscovered links between health conditions.

Clearly, implemented well, temporal EHR modelling could provide immense benefit to our society. These benefits outweigh possible harms on the provision that they are well mitigated. Hence, we devote the rest of the discussion to evaluating potential harms.

E.3. Potential Harms of EHR Modelling

There are a number of ways in which the temporal modelling of EHRs could create so-

cietal harm. To examine these risks, we evaluate them by topic, framed as a series of case studies.

E.3.1. Poor Health Outcomes Through Data Bias

Bias can mean many things when discussing the application of AI to healthcare. As such, we would encourage researchers to use Suresh and Guttag (2020)'s framework to better identify and address the specific challenges that arise when modelling health data. All are relevant to our technology, but we will specifically focus on issues arising from historical, representation and aggregation biases. Importantly, we anticipate these biases would be more easily identified due to the interpretability of our attention-based technique.

Scenario Temporal EHR modelling is deployed to predict future health having been trained on historical health records.

Societal harm Societal groups who have historically been discriminated against continue to receive below standard healthcare.

Examples of historical biases Historically, the majority of medical trials have been conducted on white males (Morley et al., 2019) providing greater health information on this population group. We also know that clinicians are fallible to human biases. For example: women are less likely than men to receive optimal care despite being more likely to present with hypertension and heart failure (Li Shanshan et al., 2016); They are often not diagnosed with diseases due to human bias (Suresh and Guttag, 2020) and are less likely than men to be given painkillers when they report they are in pain (Calderone, 1990); African Americans more likely to be misdiagnosed with schizophrenia than white patients (Gara et al., 2018).

Societal harm Societal groups who have historically been discriminated against continue to receive below standard healthcare.

Examples of historical biases Historically, the majority of medical trials have been conducted on white males (Morley et al., 2019) providing greater health information on this population group. We also know that clinicians are fallible to human biases. For example: women are less likely than men to receive optimal care despite being more likely to present with hypertension and heart failure (Li Shanshan et al., 2016); They are often not diagnosed with diseases due to human bias (Suresh and Guttag, 2020) and are less likely than men to be given painkillers when they report they are in pain (Calderone, 1990); African Americans more likely to be misdiagnosed with schizophrenia than white patients (Gara et al., 2018).

* * *

Scenario Temporal EHR modelling is deployed to predict future health and develop care plans. One model is used for a diverse population group.

Societal harm Demographics' varying symptom patterns and care plan needs are not accounted for, resulting in below optimal health outcomes for the majority.

Examples of aggregation biases It has been shown that the risk factors for carotid artery disease differ by ethnicity (Gijsberts et al., 2015) yet prevention plans are developed on data from almost exclusively a white population. A common measurement for diabetes widely used for diagnosis and monitoring differs in value across ethnicities and genders (Herman and Cohen, 2012).

* * *

Scenario Researchers select proxies to represent metrics within their model without considering broader socioeconomic factors.

Societal harm Societal groups are subject to allocation biases where they do not receive the same standard of healthcare compared to advantaged groups.

Example of poor proxy selection (Obermeyer et al., 2019) analysed a model in current commercial use which identifies high risk patients for care interventions used medical costs as a proxy for illness. It was found that black people were much sicker than white people with the same risk score as historical and socioeconomic factors mean that black people do not receive parity in health care and treatments.

* * *

Scenario Biased models are deployed as clinician-in-the-loop with the intention of using a human to protect against model biases.

Societal harm The clinician cannot redress model biases, causing the above mentioned harms to persist.

Example of human-in-the-loop failure Humans are subject to confirmation bias (Green and Chen, 2019) and (Obermeyer et al., 2019) found that whilst clinicians can redress some disparity in care allocation, they do not adjust the balance to that of a fair algorithm.

* * *

Recommendations:

- Researchers must employ caution when using proxies such as "diagnosed with condition X" equals "has condition X".
- Models must be evaluated against the intended recipient populations.
- We call on all institutions deploying temporal EHR models to assess fair performance in the target population, and, on policy makers to provide benchmarked patient datasets on which fair and safe performance of models must be demonstrated before deployment.

Questions for reflection:

- How do we best formulate the problems temporal EHR modelling is trying to solve to prevent discriminatory harms?
- If temporal EHR modelling was deployed, what techniques would be used to protect society from the reinforcement of structural inequalities and structural injustices (Jugov and Ypi, 2019)?
- How do you construct adequate benchmarking datasets to test the suitability of a temporal EHR model for a given population?
- What evidence is needed to demonstrate the clinical effectiveness of a temporal model (Greaves et al., 2018)?

E.3.2. Poor care provision through lack of contextual awareness

Scenario A temporal model is trained in a perfectly unbiased way and is deployed for use.

Societal harm As the model is not aware of emergent health outcomes, information provided is out of sync with modern medical knowledge. Potentially resulting in suboptimal care plan recommendations and predictions.

Example of lack of contextual awareness Consider training on a cohort of 100-year-old patients. They lived through the second world war, ate rations, never took a contraceptive pill and were retiring as computers became commonplace. Individuals approaching old age in the future will likely have very different health trajectories. By their nature, modelling of EHR data will be

context blind (Caruana et al., 2015). For example, if a new drug is found to prevent the development of Type 2 diabetes, model predictions will be out of sync with patient outcomes, it would view the prescription of this better drug as an error.

* * *

Questions for reflection:

- How would we develop temporal EHR models to correct dynamically for non-stationary processes (changing medical knowledge)?
- How do we introduce contextual awareness to temporal EHR models?

E.3.3. Restrictions in health access

Scenario A perfect model is deployed to be used by health insurers to identify high cost patients. Those who can be assisted by interventions are helped. The model identifies a set of individuals who are unavoidably costly, due to health issues beyond their control.

Societal harm This may cause a shift in the economic model of the society in which the EHR model is deployed (Balasubramanian et al., 2018).

Examples of economic impact Temporal models will make increasingly accurate predictions about how much each individual's healthcare will cost over a lifetime. In countries where healthcare is not universal insurers and providers will adjust their actuarial models, moving away from pooled risk healthcare costs. Health insurers increase premiums to an unfeasibly high amount for costly individuals, preventing their access to healthcare and pricing models change for those who are able to access care. Governments are forced to determine whether to use social security to provide universal healthcare. If it is not the divide between those who can access healthcare and those who can't potentially widens with ripple effects across the broader economy.

* * *

Questions for reflection:

- How do we as a society determine how to carry the cost of healthcare provision?
- What level of risk granularity should health insurance companies be privy to?

E.3.4. Changes to the Doctor / Patient relationship and agency

Scenario Temporal EHR modelling causes a paradigm shift towards solely data-centred medicine. Reducing a patient solely to their datapoints. Theoretically, the model can predict a patient's death.

Societal harm Agency is removed from both the patient and the doctor. Knowledge of death raises existential questions and changes individuals' behaviours.

Example of solely data-centred medicine The doctor is left to evaluate symptoms solely on datapoints (Kleinpeter, 2017). A patient's ability to fully express what it means to inhabit their body and feel illness (or not) is removed. They may feel unable to decline or deviate from recommended treatment plans, impacting the patient's perception of bodily autonomy.

* * *

Scenario Temporal EHR models are deployed in consumer health and wellness applications with the current narrative of empowering an individual to take control of their health (Morley and Floridi, 2019a).

Societal harm The burden of disease prevention is shifted to the individual who has proportionally little control over macro health issues, tantamount to victim blaming (Morley and Floridi, 2019a).

Example of empowerment ineffectiveness Many factors of health are not solvable with applications. For example, those on low incomes can't afford healthy foods and other socioeconomic factors prevent individuals from accessing healthcare (de Freitas and Martin, 2015).

* * *

Recommendations:

- Policy makers should contemplate how far into the future various institutions should be permitted to predict peoples' health.
- We encourage future researchers to view consumer applications that use temporal EHR modelling as digital companions (Morley and Floridi, 2019b).

Questions for reflection:

- How many years in advance should predictive health technologies be able to predict?
- How much clinical decision-making should we be delegating to AI-Health solutions (di Nucci, 2019)?
- How can we develop temporal EHR models to account for the socioeconomic and historical factors which contribute to poor health outcomes?
- How do we incorporate into temporal EHR modelling an ethical focus on the end user, and their expectations, demands, needs, and rights (Morley and Floridi, 2020)?

E.4. Wider reaching applications

The temporal modelling of irregular data could be applied to many domains. For example, education and personalised learning, crime and predictive policing, social media interactions and more. Each of these applications raises questions around the societal impact of improved temporal modelling. We hope that the above analysis provides stimulation when developing its beneficial usage.