SUPPLEMENTARY MATERIALS FOR SUBMISSION 3981: DEEP LINEAR HAWKES PROCESSES

TABLE OF CONTENTS

Appendix A Acronyms and Notation

Appendix B Additional Details on Methods

Appendix C Experimental Configurations and Datasets

Appendix D Additional Experimental Results

A Acronyms and Notation

Table 2: Key notation used repeatedly across this paper.

Symbol	Space	Description
t	$\mathbb{R}_{\geq 0}$	Time
T	$\mathbb{R}_{\geq 0}$	Maximum time in a given sequence's observation window
t_i	$\mathbb{R}_{\geq 0}$	i^{th} time
$t-$	$\mathbb{R}_{\geq 0}$	Subscript minus indicates left-limit
$t+$	$\mathbb{R}_{\geq 0}$	Subscript plus indicates right-limit
k	$\mathcal{M} = \{1, \ldots, K\}$	Event mark
$\mathcal{H}$	$\mathcal{M}^N \times \mathbb{R}_{\geq 0}^N$	Event history for N events
$\mathbf{N}_t$	$\mathbb{Z}_{\geq 0}^K$	Counting process for K marks at time t
λ_t^k	$\mathbb{R}_{>0}$	Intensity of k^{th} mark type at time t
$\boldsymbol{\lambda}_t$	$\mathbb{R}_{\geq 0}^K$	Vector of K mark intensities at time t
λ_t	$\mathbb{R}_{>0}$	Ground/total intensity (sum of mark-specific intensities)
$\mathcal{L}(\cdot)$	$\mathbb{R}$	Log-likelihood of the argument under the model
ν^{k}	$\mathbb{R}_{\geq 0}$	Background intensity for the k^{th} mark
$\boldsymbol{\alpha}$	$\mathbb{R}_{\geq 0}^{K,K}$	(For LHP) Matrix of intensity impulses from each type of mark
$\boldsymbol{\beta}$	$\mathbb{R}_{\geq 0}^{K,K}$	(For LHP) Dynamics matrix of intensity evolution
R	$\mathbb{N}$	Mark embedding rank
P	$\mathbb{N}$	LLH/SSM hidden dimension
$\mathbf{x}_t$	$\mathbb{R}^P$	LLH/SSM hidden state at time t
$\mathbf{x}_0$	$\mathbb{R}^P$	Learned LLH/SSM initial hidden state
H	$\mathbb{N}$	LLH/SSM output dimension
$\mathbf{y}_t$	$\mathbb{R}^H$	LLH/SSM output at time t
$\mathbf{u}_t$	$\mathbb{R}^H$	LLH/SSM input at time t
$\mathbf{A}$	$\mathbb{R}^{P \times P}$	LLH/SSM transition matrix
$\mathbf{B}$	$\mathbb{R}^{P \times H}$	LLH/SSM input matrix
$\mathbf{C}$	$\mathbb{R}^{H \times P}$	LLH/SSM output matrix
$\mathbf{D}$	$\mathbb{R}^{H \times H}$	LLH/SSM passthrough matrix
$\mathbf{E}$	$\mathbb{R}^{P \times R}$	LLH mark embedding matrix ($P \times R$ in low-rank factorization)
L	$\mathbb{N}$	Number of linear recurrences in a DLHP model; model "depth"
$\boldsymbol{\alpha}$	$\mathbb{R}^{R \times K}$	(For DLHP) Mark impulses ($R \times K$ in low-rank factorization)
$\sim$	N/A	Tilde (e.g. $\tilde{\mathbf{B}}$) denotes variable is in the diagonalized eigenbasis
Λ	$\mathbb{C}^{P \times P}$	Matrix of eigenvalues of $\mathbf{A}$; diagonalized dynamics matrix
$\bar{\Lambda}$	$\mathbb{C}^{P \times P}$	Discretized diagonal dynamics matrix
(l)	N/A	Superscript index in parenthesis indicates layer (i.e. $\mathbf{x}$ for layer l)

Table 3: Key acronyms used throughout this paper.

Acronym	Page number	Definition
CNN	6	Convolutional neural network
LHP	1	Linear Hawkes process
LLH	2	Latent linear Hawkes
MTPP	1	Marked temporal point process
RNN	1	Recurrent neural network
SSM	1	(Deep) State-space model
TPP	7	Temporal point process
ZOH	5	Zero-order hold
RMTPP	7	Recurrent marked temporal point process (Du et al., 2016)
NHP	1	Neural Hawkes process (Mei & Eisner, 2017)
SAHP	7	Self-attentive Hawkes process (Zhang et al., 2020)
THP	7	Transformer Hawkes process (Zuo et al., 2020)
AttNHP	7	Attentive neural Hawkes process (Yang et al., 2022)
IFTPP	7	Intensity-free temporal point process (Shchur et al., 2020a)
DLHP	1	Deep linear Hawkes process (ours)

B Additional Details on Methods

B.1 Discretization and Zero Order Hold

The linear recurrence is defined in continuous-time. This mirrors the (M)TPP setting, where event times are not on a fixed intervals. We use the zero-order hold (ZOH) discretization method, to convert the continuous-time linear recurrence into a sequence of closed-form updates, given the integration times, that can also be efficiently computed. We refer the reader to Iserles (2009) for a comprehensive introduction to the ZOH transform.

The main assumption of the ZOH discretization is that the input signal is held constant over the time period being integrated. Under this assumption, it is possible to solve for the dynamics and input matrices that yield the correct state at the end of the integration period. For the LLH dynamics in Eq. (10), when no events occur in (t, t'), this becomes

$$\mathbf{x}_{t'-} = \int_t^{t'} \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t \mathrm{d}t = \overline{\mathbf{A}}\mathbf{x}_t + \overline{\mathbf{A}\mathbf{B}}\mathbf{u}_t \quad \text{assuming} \quad \mathrm{d}\mathbf{u}_t = \mathbf{0} \in [t, t'], \tag{18}$$

where the resulting discretized matrices are

$$\overline{\mathbf{A}} = e^{\mathbf{A}\Delta t}, \quad \overline{\mathbf{A}\mathbf{B}} = \mathbf{A}^{-1}(e^{\mathbf{A}\Delta t} - \mathbf{I})\mathbf{A}\mathbf{B}, \quad \text{where} \quad \Delta t = t' - t. \tag{19}$$

The ZOH does not affect the output or passthrough matrices $\mathbf{C}$ and $\mathbf{D}$. To compute the matrices $\overline{\mathbf{A}}$ and $\overline{\mathbf{A}\mathbf{B}}$ however requires computing a matrix exponential and a matrix inverse. However, Smith et al. (2022) avoid this by diagonalizing the system (also avoiding a dense matrix-matrix multiplication in the parallel scan). The diagonalized dynamics and input matrices are denoted $\mathbf{\Lambda}$ (a diagonal matrix) and $\mathbf{\Lambda}\tilde{\mathbf{B}}$ respectively. In this case, Eq. (19) reduces to

$$\overline{\mathbf{A}} = e^{\mathbf{\Lambda}\Delta t}, \tag{20}$$

$$\overline{\mathbf{A}\mathbf{B}} = \mathbf{\Lambda}^{-1}(e^{\mathbf{\Lambda}\Delta t} - \mathbf{I})\mathbf{\Lambda}\tilde{\mathbf{B}} \tag{21}$$

$$= (e^{\mathbf{\Lambda}\Delta t} - \mathbf{I})\tilde{\mathbf{B}} \quad \text{(diagonal matrices commute)} \tag{22}$$

where $e^{\mathbf{\Lambda}\Delta t}$ is trivially computable as the exponential of the leading diagonal of $\mathbf{\Lambda}\Delta t$. These operations are embarrassingly parallelizable across the sequence length and state dimension given the desired evaluation times.

To contextualize, suppose an event occurs at time t, Eq. (22) allows us to exactly (under the constant-input assumption) efficiently evaluate the linear recurrence at subsequent times t'. We use this extensively in the DLHP to efficiently evaluate the recurrence (and hence the intensity) at the irregularly-spaced event times and times used to compute the integral term.

It should be noted the discretization was done to compute a left-limit $\mathbf{x}_{t'-}$ from a previous right-limit $\mathbf{x}_t$. Should an event not occur at t', then the left- and right-limits agree and $\mathbf{x}_{t'-} = \mathbf{x}_{t'+} = \mathbf{x}_{t'}$. If an event does occur at time t' with mark k, then the left-limit $\mathbf{x}_{t'-}$ can be incremented by $\tilde{\mathbf{E}}\boldsymbol{\alpha}_k$ to compute $\mathbf{x}_{t'+} = \mathbf{x}_{t'}$. This increment from left- to right-limit is exact and leverages no discretization assumption.

B.2 Interpretation for Input-Dependent Dynamics

Consider the input-dependent recurrence for an LLH layer, as defined in Eq. (17):

$$\mathrm{d}\tilde{\mathbf{x}}_t := \mathbf{\Lambda}_i\tilde{\mathbf{x}}_{t-}\mathrm{d}t + \mathbf{\Lambda}_i\tilde{\mathbf{B}}\mathbf{u}_{t-}\mathrm{d}t + \tilde{\mathbf{E}}\boldsymbol{\alpha}\mathrm{d}\mathbf{N}_t \tag{23}$$

for $t \in (t_i, t_{i+1}]$ where $\mathbf{\Lambda}_i := \mathrm{diag}(\Delta_i)\mathbf{\Lambda}$ with the input-dependent factor defined as $\Delta_i := \mathrm{softplus}(\mathbf{W}'\mathbf{u}_{t_i} + \mathbf{b}') \in \mathbb{R}^P_{>0}$. This factor can be thought of as the input-dependent relative-time scale for the dynamics. To see this, we first note that for vectors $\mathbf{p}, \mathbf{q} \in \mathbb{R}^d$, the following holds true: $\mathrm{diag}(\mathbf{p})\mathbf{q} = \mathbf{p} \odot \mathbf{q} = \mathbf{q} \odot \mathbf{p}$ where $\odot$ is the Hadamard or element-wise product. It then follows that

$$\mathrm{d}\tilde{\mathbf{x}}_t := \mathbf{\Lambda}_i\tilde{\mathbf{x}}_{t-}\mathrm{d}t + \mathbf{\Lambda}_i\tilde{\mathbf{B}}\mathbf{u}_{t-}\mathrm{d}t + \tilde{\mathbf{E}}\boldsymbol{\alpha}\mathrm{d}\mathbf{N}_t \tag{24}$$

$$= \mathbf{\Lambda}_i(\tilde{\mathbf{x}}_{t-} + \tilde{\mathbf{B}}\mathbf{u}_{t-})\mathrm{d}t + \tilde{\mathbf{E}}\boldsymbol{\alpha}\mathrm{d}\mathbf{N}_t \tag{25}$$

$$= \mathrm{diag}(\Delta_i)\mathbf{\Lambda}(\tilde{\mathbf{x}}_{t-} + \tilde{\mathbf{B}}\mathbf{u}_{t-})\mathrm{d}t + \tilde{\mathbf{E}}\boldsymbol{\alpha}\mathrm{d}\mathbf{N}_t \tag{26}$$

$$= [\mathbf{\Lambda}(\tilde{\mathbf{x}}_{t-} + \tilde{\mathbf{B}}\mathbf{u}_{t-})] \odot (\Delta_i\mathrm{d}t) + \tilde{\mathbf{E}}\boldsymbol{\alpha}\mathrm{d}\mathbf{N}_t. \tag{27}$$

As shown, the positive vector Δ_i can be thought of as changing the relative time-scale for each channel in the hidden state $\tilde{\mathbf{x}}$. Large values of Δ_i will act as if time is passing quickly, encouraging the state to converge to the steady-state sooner. Conversely, smaller values will make time pass more slowly causing the model to retain the influence that prior events have on future ones (for that specific channel in $\tilde{\mathbf{x}}$ at least).

B.3 Forwards and Backwards Zero Order Hold Discretization

In Section 3.3 we highlighted that the ZOH discretization is exact when $\mathbf{u}_t$ is held constant over the integration window. This raises a unique design question for DLHPs: what constant value should $\mathbf{u}_t$ take on when evolving $\mathbf{x}$ from time t to t'? For the first layer of the model, the input is zero by construction, so there is no choice to be made—in fact, since $\mathbf{u}$ is constant for the first layer the updates are exact. However, the input is non-zero at deeper layers, and, crucially, varies over the integration period.

We must therefore decide how to select a $\mathbf{u}$ value over the integration period. This should be a value in (or function of) $\{\mathbf{u}_s \mid s \in [t, t')\}$. Note this is because the value at t', $\mathbf{u}_{t'}$, *cannot* be incorporated as this would cause a data leakage in our model; while values prior to t would discard the most recent mark. For this work, we explore two natural choices: (i) the input value at the beginning of the interval, $\mathbf{u}_t$, and (ii) the left-limit at the end of the interval, $\mathbf{u}_{t'-}$. We illustrate the backwards variant in Fig. 2, where in the rightmost panel, we use the $\mathbf{u}_{t^*}$ values at each layer, as opposed to $\mathbf{u}_{t_3}$. We refer to these options as *forwards* and *backwards* ZOH, respectively. All experiments in the main paper utilize backwards ZOH.

It is not obvious *a priori* which one of these modes is more performant. We therefore conducted an ablation experiment in Table 9. We see that there is little difference between the two methods. We also note that models are learned through this discretization, and so this decision does not mean that a model is "incorrectly discretized" one way or the other, but instead they define subtlety different families of models. Theoretical and empirical investigation of the interpretations of this choice is an interesting area of investigation going forwards, extending the ablations we present in Table 9.

C EXPERIMENTAL CONFIGURATIONS AND DATASETS

C.1 TRAINING DETAILS & HYPERPARAMETER CONFIGURATIONS

We apply a grid search for all models on all datasets for hyperparameter tuning. We use a default batch size of 256 for training. For models/datasets that require more memory (e.g. large mark space or long sequences), we reduce the batch size and keep them as consistent as possible among all the models on each dataset. We use the Adam stochastic gradient optimizer (Kingma & Ba, 2015), with a learning rate of 0.01 and a linear warm-up schedule over the first 1% iterations, followed by a cosine decay. Initial experiments showed this setting generally worked well across different models and datasets leads to convergence within 300 epochs. We also clip the gradient norm to have a max norm of 1 for training stability. We use Monte-Carlo samples to estimate the integral in log-likelihood, where we use 10 Monte-Carlo points per event during training.

On the five `EasyTPP` benchmark datasets and MIMIC-II that are smaller in their scales, we choose an extended grid based on the architecture reported in the `EasyTPP` paper. Specifically, we search over hidden states size $h = \{16, 32, 64, 128, 256\}$ for RMTPP, $h = \{32, 64, 128\}$ for NHP, and $h = \{16, 32, 64\}$ for IFTPP. For SAHP, THP, and AttNHP, we searched over all combinations of number of $L = \{1, 2, 3\}$, hidden state size = $\{16, 32, 64, 128\}$, and number of heads = $\{1, 2, 4\}$. Finally, for DLHP, we considered combinations for number of layers = $\{1, 2, 3, 4\}$, $p = \{16, 32, 64, 128\}$ and $h = \{16, 32, 64, 256\}$. We fixed the activation function as GeLU (Hendrycks & Gimpel, 2016) and apply post norm with layer norm (Ba, 2016). We fix the dropout as 0.1 for DLHP on the five core benchmark datasets, and add dropout = $\{0, 0.1\}$ to the grid search for the other three datasets. Due to the scale of Last.fm and EHRShot datasets, we perform a smaller search over architectures that roughly match the parameter counts for all models at three levels: 25k, 50k, 200k, and choose the model with the best validation results. AttNHP has expensive memory requirements that tends to have smaller batch sizes than other models. We were unable to train any AttNHP on EHRShot. The final model architectures used are reported in Table 4a and Table 4b. These configurations are also included in the supplementary code we include.

Table 4: Model architectures for the experiments presented in Table 1

(a) Model architectures for the five `EasyTPP` benchmark datasets.

Model	Amazon	Retweet	Taxi	Taobao	StackOverflow
RMTPP	$h = 128$	$h = 16$	$h = 128$	$h = 16$	$h = 256$
NHP	$h = 128$	$h = 64$	$h = 128$	$h = 128$	$h = 64$
SAHP	$h = 32, l = 2, \text{heads} = 2$	$h = 32, l = 3, \text{heads} = 4$	$h = 16, l = 2, \text{heads} = 4$	$h = 32, l = 1, \text{heads} = 1$	$h = 64, l = 1, \text{heads} = 1$
THP	$h = 32, l = 2, \text{heads} = 4$	$h = 16, l = 3, \text{heads} = 4$	$h = 128, l = 1, \text{heads} = 4$	$h = 64, l = 1, \text{heads} = 1$	$h = 16, l = 2, \text{heads} = 4$
AttNHP	$h = 64, t = 16, l = 2, \text{heads} = 4$	$h = 16, t = 16, l = 2, \text{heads} = 4$	$h = 16, t = 16, l = 3, \text{heads} = 4$	$h = 32, t = 16, l = 3, \text{heads} = 4$	$h = 32, t = 16, l = 2, \text{heads} = 4$
IFTPP	$h = 64$	$h = 64$	$h = 32$	$h = 64$	$h = 32$
DLHP	$h = 64, p = 128, l = 2$	$h = 128, p = 128, l = 2$	$h = 128, p = 16, l = 4$	$h = 32, p = 16, l = 4$	$h = 32, p = 32, l = 3$

(b) Model architectures for the additional three benchmark datasets.

Model	Last.fm	MIMIC-II	EHRShot
RMTPP	$h = 256$	$h = 128$	$h = 16$
NHP	$h = 112$	$h = 128$	$h = 80$
SAHP	$h = 136, l = 2, \text{heads} = 4$	$h = 64, l = 2, \text{heads} = 4$	$h = 8, l = 2, \text{heads} = 4$
THP	$h = 48, l = 2, \text{heads} = 4$	$h = 32, l = 3, \text{heads} = 4$	$h = 32, l = 2, \text{heads} = 4$
AttNHP	$h = 28, t = 16, l = 2, \text{heads} = 4$	$h = 64, t = 16, l = 3, \text{heads} = 2$	OOM
IFTPP	$h = 48$	$h = 256$	$h = 16$
DLHP	$h = 144, p = 16, l = 2$	$h = 256, p = 64, l = 2$	$h = 128, p = 32, l = 2$

C.2 DATASET STATISTICS

We report the statistics of all eight datasets we used in Table 5. We used the `HuggingFace` version of the five `EasyTPP` datasets. For all datasets, we further ensure the MTPP modeling assumptions are satisfied that no more than two events occur at the same time (i.e. inter-arrival time is strictly positive), and event times do not lie on grid points that are effectively discrete-time events. Dataset descriptions and pre-processing details are provided in Appendix C.3.

Table 5: Statistics of the eight datasets we experiment with.

Dataset	K	Number of Events			Sequence Length			Number of Sequences		
		Train	Valid	Test	Min	Max	Mean	Train	Valid	Test
Amazon	16	288,377	40,995	84,048	14	94	44.8	6,454	922	1,851
Retweet	3	2,176,116	215,521	218,465	50	264	108.8	20,000	2,000	2,000
Taxi	10	51,584	7,404	14,820	36	38	37.0	1,400	200	400
Taobao	17	73,483	11,472	28,455	28	64	56.7	1,300	200	500
StackOverflow	22	90,497	25,762	26,518	41	101	64.8	1,401	401	401
Last.fm	120	1,534,738	344,542	336,676	6	501	207.2	7,488	1,604	1,604
MIMIC-II	75	9,619	1,253	1,223	2	33	3.7	2600	325	325
EHRShot	668	759,141	165,237	170,147	5	3,955	177.0	4,329	927	927

C.3 Dataset Pre-processing

We use the default train/validation/test splits for `EasyTPP` benchmark datasets. For MIMIC-II, we copy Du et al. (2016) and keep the 325 test sequences in the test split, and further split the 2,935 training sequences into 2,600 for training and 325 for validation. In our pre-processed datasets, Last.fm and EHRShot, we randomly partition into subsets containing 70%, 15%, 15% of all sequences for training/validation/test respectively. We provide a high-level description of all the datasets we used, followed by our pre-processing procedure of Last.fm and EHRShot in more detail. Note that for datasets that contain concurrent events or effectively discrete times, we apply a small amount of jittering to ensure no modeling assumptions are violated in the MTPP framework.

Amazon (Ni et al., 2019) contains user product reviews where product categories are considered as marks. **Retweet** (Zhao et al., 2015) predicts the popularity of a retweet cascade, where the event type is decided by if the retweet comes from users with "small", "medium", or "large" influences, measured by number of followers (Mei & Eisner, 2017). **Taxi** data (Whong, 2014) uses data from the pickups and dropoffs of New York taxi and the marks are defined as discrete locations. **Taobao** (Xue et al., 2022) describes the viewing patterns of users on an e-commerce site, where item categories are considered as marks. **StackOverflow** contains the badges (defined as marks) awarded to users on a question-answering website. Finally, **MIMIC-II** (Saeed et al., 2002) records different diseases (used as marks) during hospital visits of patients. We add a small amount of noise to the MIMIC-II event times so that events do not lie on a fixed grid. Both StackOverflow and MIMIC-II datasets were first pre-processed by Du et al. (2016).

Last.fm Celma Herrada et al. (2009); McFee et al. (2012) records 992 users' music listening habits that has been widely used in MTPP literature (Kumar et al., 2019; Boyd et al., 2020; Bosser & Taieb, 2023). Mark types are defined as the genres of a song, and each event is a play of a particular genre. Each sequence represents the monthly listening behavior of each user, with sequence lengths between 5 and 500. If the song is associated with multiple genres we select a random one of the genres, resulting in a total of 120 different marks.

EHRShot Wornow et al. (2023) is a newly proposed large dataset of longitudinal de-identified patient medical records, and has rich information such as hospital visits, procedures, and measurements. We introduce an MTPP dataset derived from EHRShot, where medical services and procedures are treated as marks, as identified by *Current Procedural Terminology* (CPT-4) codes. Each patient defines an event sequence, and we retain only CPT-4 codes with at least 100 occurrences in the dataset. For the $< 1\%$ events of events where there are more than 10 codes at a single timestamp, we retain the top 10 codes with the most frequencies and discard the rest. We then add a small amount of random noise to the event time to ensure they are not overlapping. This process ensures we still satisfy the MTPP framework, and can reasonably instead compute top-10 accuracy for the next mark prediction. Other work has considered extending the MTPP framework to consider simultaneous event occurrence (Chang et al., 2024). Then we standardize each sequence to start and end with start and end of a sequence events. Note that we do not score these events. Event times are normalized to be in hours. We discard sequences that have less than 5 events and a single timestamp. This leads to the final version of our dataset to have 668 marks, and the sequence lengths range from 5 to 3955 events, reflecting patient histories that can span multiple years. We include the notebook used for compiling the data we use from the original EHRShot data in the supplementary code submission.

D ADDITIONAL EXPERIMENTAL RESULTS

D.1 FULL RESULTS ON BENCHMARK DATASETS

We provide the full log-likelihood results and corresponding plots in Table 6 and Fig. 5 respectively, where we decompose the likelihood into time and mark likelihoods. The improvement of our DLHP model is mainly driven by better modeling of time, though we also often obtain best- or second-best predictive performance on marks from the next event prediction accuracy results conditioned on true event time in Table 7. In all predictive metrics, our model ranks the best averaged over all of the datasets.

In aggregate, our model achieves a 1.416 per-event likelihood ratio between itself and the next best method across all datasets (a 41.6% improvement in likelihood). This is calculated by computing the mean log-likelihood ratio across all datasets and then exponentiating. Doing so is equivalent to taking the geometric mean across likelihood ratios.

Table 6: Complete per-event log-likelihood (higher is better) results on the held-out test for the eight benchmark datasets we consider. In Table 6a we show the full log-likelihood. We then decompose this log-likelihood into the log-likelihood of the event time in Table 6b, and the time-conditional log-likelihood of the mark type in Table 6c. OOM indicates out of memory. We highlight the best-performing model in bold and underline the second-best. We also report the average rank of models across datasets as a summary metric (lower is better). DLHP is consistently the best or second best-performing model across all datasets.

(a) Full log-likelihood results (equal to the summation of Table 6b and Table 6c). Extended version of Table 1.

Model	**Per-Event Log-Likelihood, $\mathcal{L}_{\text{Total}}$ (nats)**								**Avg. Ranking**
	Amazon	Retweet	Taxi	Taobao	StackOverflow	Last.fm	MIMIC-II	EHRShot	
RMTPP	-2.137	-7.169	0.347	1.006	-2.403	-1.776	-0.480	-8.035	6.38
NHP	0.205	**-6.346**	0.516	1.163	-2.243	-0.578	0.076	-3.907	3.13
SAHP	-2.040	-6.704	0.372	1.201	-2.283	-1.500	-0.773	-6.845	5.13
THP	-2.098	-6.652	0.374	0.791	-2.331	-1.716	-0.587	-7.183	5.63
AttNHP	0.608	-6.459	0.499	1.278	-2.179	-0.558	-0.244	OOM	2.86
IFTPP	0.493	-10.339	0.454	**1.335**	-2.224	**-0.472**	0.299	-6.424	3.00
DLHP (Ours)	**0.765**	-6.367	**0.528**	1.332	**-2.165**	-0.496	**1.231**	**-2.189**	**1.38**

(b) Per-event log-likelihood of the event times (higher is better).

Model	**Next Event Time Log-Likelihood, $\mathcal{L}_{\text{Time}}$ (nats)**								**Avg. Ranking**
	Amazon	Retweet	Taxi	Taobao	StackOverflow	Last.fm	MIMIC-II	EHRShot	
RMTPP	0.010	-6.231	0.622	2.427	-0.780	0.259	-0.182	-1.888	5.88
NHP	2.196	**-5.583**	0.728	2.579	-0.703	1.196	0.240	-0.758	3.38
SAHP	0.173	-5.895	0.681	2.612	-0.681	0.600	-0.298	-1.779	4.63
THP	-0.070	-5.867	0.623	2.242	-0.769	0.220	-0.277	-1.890	6.00
AttNHP	2.545	-5.688	0.724	2.665	-0.681	1.213	-0.017	OOM	3.14
IFTPP	2.482	-9.494	0.736	2.730	-0.660	1.290	0.536	-2.642	3.25
DLHP	**2.638**	-5.600	**0.738**	**2.742**	**-0.636**	**1.294**	**1.345**	**0.723**	**1.13**

(c) Per event log-likelihood of mark type conditioned on the arrival time (higher is better).

Model	**Per-Event Next Mark Log-Likelihood, $\mathcal{L}_{\text{Mark}}$ (nats)**								**Avg. Ranking**
	Amazon	Retweet	Taxi	Taobao	StackOverflow	Last.fm	MIMIC-II	EHRShot	
RMTPP	-2.148	-0.939	-0.275	-1.421	-1.623	-2.035	-0.298	-6.147	6.00
NHP	-1.992	**-0.764**	-0.212	-1.416	-1.540	-1.774	-0.164	-3.149	2.75
SAHP	-2.213	-0.809	-0.308	-1.411	-1.602	-2.100	-0.475	-5.066	5.88
THP	-2.028	-0.786	-0.249	-1.451	-1.563	-1.936	-0.310	-5.294	5.00
AttNHP	-1.938	-0.771	-0.225	**-1.387**	**-1.498**	-1.771	-0.227	OOM	2.14
IFTPP	-1.989	-0.845	-0.282	-1.395	-1.565	**-1.763**	-0.237	-3.782	3.75
DLHP	**-1.873**	-0.767	**-0.209**	-1.410	-1.529	-1.790	**-0.114**	**-2.912**	**1.88**

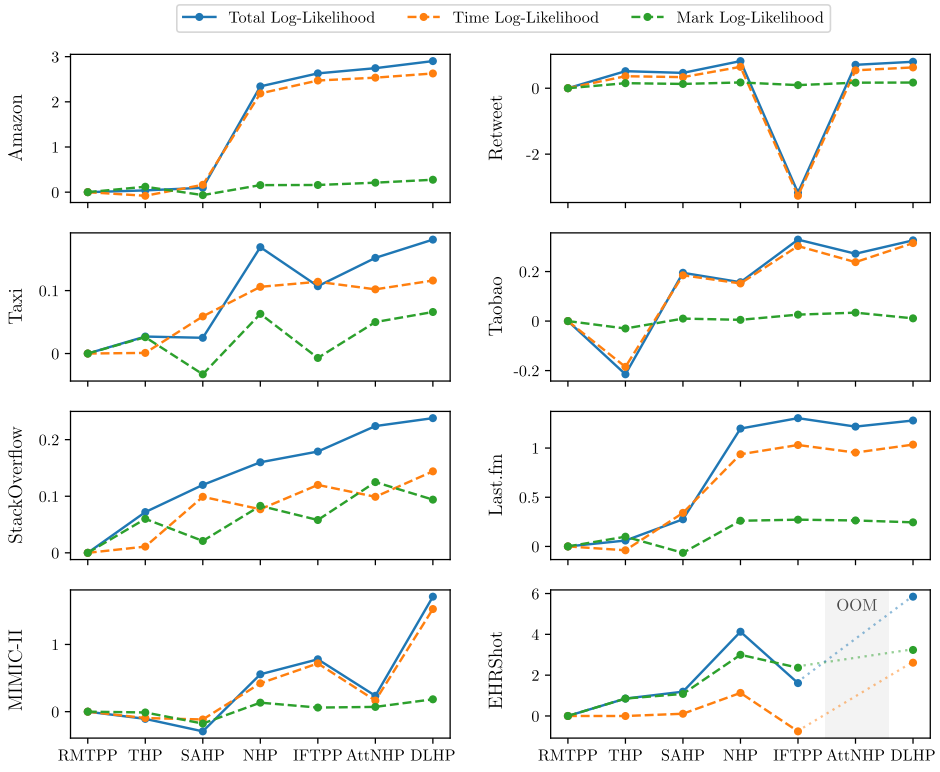

Figure 5: Visualization of $\mathcal{L}_{\text{Total}}$ decomposed into $\mathcal{L}_{\text{Time}}$ and $\mathcal{L}_{\text{Mark}}$ for all models and all datasets relative to RMTPP, as discussed in Section 5.2. The improvement of DLHP is mainly driven by better modeling of $\mathcal{L}_{\text{Time}}$.

Table 7: Next event prediction accuracy (reported as a percentage, ↑ is better) conditioned on the true event time. We report top 1 accuracy for all datasets except for top 10 accuracy for EHRShot, due to the pre-processing procedure described in Appendix C.3. We **bold** the **best** result per dataset, and underline the runner-up.

Model	Next Mark Accuracy (%)								Avg. Ranking
	Amazon	Retweet	Taxi	Taobao	StackOverflow	Last.fm	MIMIC-II	EHRShot (Top 10)	
RMTPP	30.96	50.36	91.37	60.93	46.46	52.51	92.20	34.09	5.63
NHP	39.23	**61.47**	92.82	**61.58**	47.03	56.43	94.32	71.85	1.88
SAHP	32.03	59.18	92.23	60.78	46.46	52.84	84.52	32.56	5.63
THP	34.63	60.17	91.59	60.00	46.64	53.28	90.98	45.47	5.13
AttNHP	38.55	60.92	92.60	61.24	**48.33**	56.18	91.98	OOM	3.00
IFTPP	35.75	49.08	91.71	60.93	45.69	**56.44**	93.43	60.60	4.25
DLHP	**40.66**	61.33	**93.05**	61.06	47.45	56.26	**96.55**	**75.45**	**1.75**

D.2 Full Results for Synthetic Poisson Experiments

We present the full results in Fig. 6 for all models regarding the synthetic experiments discussed in Section 5.1. All models are trained until convergence using a set of 5,000 generated sequences, where we use 20 Monte Carlo points per event to estimate the integral of log-likelihood during training to accommodate the sparsity of events. We used small models so they do not overfit; model architecture and parameter counts are reported in Table 8. We plot the background intensity conditioned on empty sequences using 1,000 equidistant grid points between the start and end points. Our model is the only one that perfectly recovers the underlying ground truth intensity, while also using the fewest parameters.

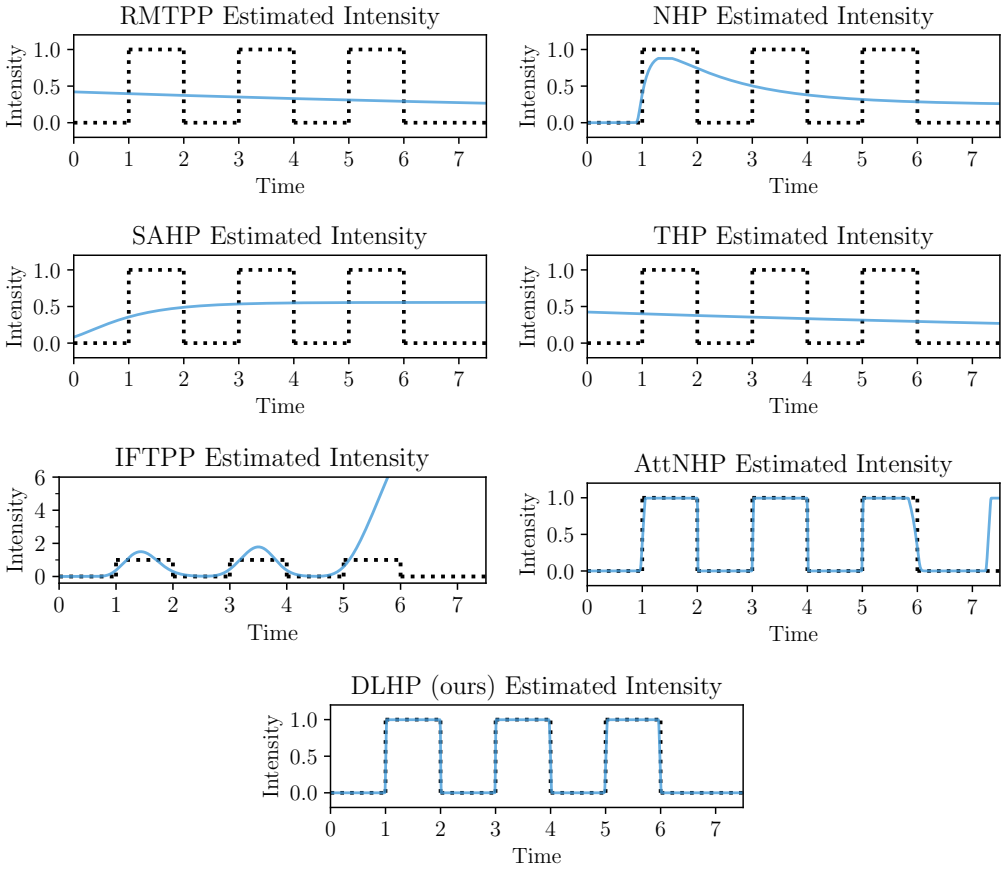

Figure 6: Results for all baseline models for the synthetic Poisson experiment introduced in Section 5.1. The estimated intensity (blue lines) conditioned on an empty sequence are plotted against the ground truth (dotted black lines).

Table 8: Model architectures and corresponding parameter counts for synthetic Poisson experiments.

Model	Architecture	# Parameters
RMTPP	$h = 16$	627
NHP	$h = 8$	1010
SAHP	$h = 16, l = 2, \text{heads} = 4$	1738
THP	$h = 16, l = 2, \text{heads} = 4$	1684
AttNHP	$h = 8, t = 2, l = 2, \text{heads} = 2$	1178
IFTPP	$h = 16$	1899
DLHP	$h = 4, p = 4, l = 2$	178

D.3 Ablation for Different DLHP Variants

We perform an ablation study of different model variants that we proposed on all datasets and summarize the results in Table 9. We train EHRShot using 10% of its training data because larger dataset scale requires more training time (but use the original validation and test sets for model selection and reporting results). Forward and backward discretization are very close in performance, with backwards discretization having a slight edge. Models that are input-dependent achieve better performance on most datasets, although on certain datasets input dependence appears to harm performance. It is an interesting direction for future work to explore theoretically and empirically when each of these variants is best. We select backward discretization with input dependence for the results in the main paper.

Table 9: Ablation for different model variants log-likelihood (LL). ID stands for input-dependent, see Section 3.4. Backward and Forward respectively refer to using $\mathbf{u}_{t_{i-1}}$ and $\mathbf{u}_{t_i-}$ (i.e. the previous right limit or current left limit), see Appendix B.3.

Dataset	Model variant	LL	Arrival time LL	Mark LL conditioned on time
Amazon	Forward	0.705	2.617	-1.912
	Forward + ID	0.748	2.634	-1.886
	Backward	0.740	2.640	-1.899
	Backward + ID	**0.765**	2.638	-1.873
Retweet	Forward	-6.405	-5.625	-0.780
	Forward + ID	-6.370	-5.602	-0.767
	Backward	-6.398	-5.618	-0.780
	Backward + ID	**-6.367**	-5.600	-0.767
Taxi	Forward	0.473	0.697	-0.224
	Forward + ID	0.525	0.733	-0.208
	Backward	0.477	0.705	-0.228
	Backward + ID	**0.528**	0.738	-0.209
Taobao	Forward	1.207	2.643	-1.435
	Forward + ID	**1.332**	2.742	-1.410
	Backward	1.215	2.648	-1.432
	Backward + ID	**1.332**	2.742	-1.410
StackOverflow	Forward	-2.249	-0.676	-1.572
	Forward + ID	-2.174	-0.644	-1.530
	Backward	-2.225	-0.679	-1.547
	Backward + ID	**-2.165**	-0.636	-1.529
Last.fm	Forward	**-0.463**	1.309	-1.772
	Forward + ID	-0.477	1.302	-1.779
	Backward	-0.474	1.303	-1.777
	Backward + ID	-0.496	1.294	-1.790
MIMIC-II	Forward	0.555	0.847	-0.292
	Forward + ID	**1.319**	1.405	-0.086
	Backward	0.322	0.601	-0.279
	Backward + ID	1.231	1.345	-0.114
EHRShot (10%)	Forward	-3.885	0.105	-3.990
	Forward + ID	**-3.848**	-0.021	-3.827
	Backward	-4.571	-0.432	-4.139
	Backward + ID	-4.684	-0.641	-4.043

D.4 Model Calibration

To further probe the models, we evaluate the calibration metrics of MTPPs that are proposed in literature (Bosser & Taieb, 2023), which has a different focus than log-likelihood-based evaluation. On a high level, calibration describes how well the uncertainty in the model is reflected in the observed data. However, a model can achieve perfect calibration by predicting the marginal distribution, so better calibration *does not* necessarily transform into better predictive performance. We therefore present these metrics as a secondary metric (secondary to log-likelihood per Daley & Vere-Jones (2003)) for investigating the performance of different models. We provide summarized statistics for both probabilistic calibration error (PCE) for time calibration and expected calibration error (ECE) for mark calibration in Table 10, and visualize the calibration curves in Figs. 7 and 8. From our results, all MTPP models are well-calibrated on most of the datasets, especially on mark predictions.

Table 10: Calibration results for the models and datasets tests.

(a) Probabilistic calibration error (PCE) for time calibration in percentage.

Model	**Probabilistic Calibration Error (PCE)**							
	Amazon	Retweet	Taxi	Taobao	StackOverflow	Last.fm	MIMIC-II	EHRShot
RMTPP	13.70	4.20	3.55	10.18	1.91	11.55	3.85	13.31
NHP	7.57	0.15	0.27	7.38	1.77	4.77	6.05	8.22
SAHP	10.86	9.75	1.73	2.88	1.14	10.89	2.79	15.05
THP	12.28	5.71	3.32	16.32	2.10	10.90	1.21	14.55
AttNHP	6.20	1.26	0.96	3.17	1.52	1.57	4.66	OOM
IFTPP	1.74	23.93	0.44	0.61	0.50	0.30	2.19	17.66
DLHP	3.47	0.40	0.13	2.05	0.60	1.18	8.94	12.47

(b) Expected calibration error (ECE) for mark calibration in percentage.

Model	**Expected Calibration Error (ECE)**							
	Amazon	Retweet	Taxi	Taobao	StackOverflow	Last.fm	MIMIC-II	EHRShot
RMTPP	6.41	5.89	2.62	1.60	1.36	2.44	1.97	9.22
NHP	6.75	0.33	0.81	4.40	1.02	4.10	1.92	2.84
SAHP	8.36	4.74	6.96	3.00	1.12	8.55	5.77	11.09
THP	2.02	1.20	1.74	6.48	0.77	2.67	1.81	11.42
AttNHP	2.88	0.39	0.44	2.52	1.21	0.50	2.79	OOM
IFTPP	0.37	0.58	0.41	1.49	1.48	0.59	1.40	2.01
DLHP	1.00	0.72	0.46	1.66	2.01	0.74	2.34	1.19

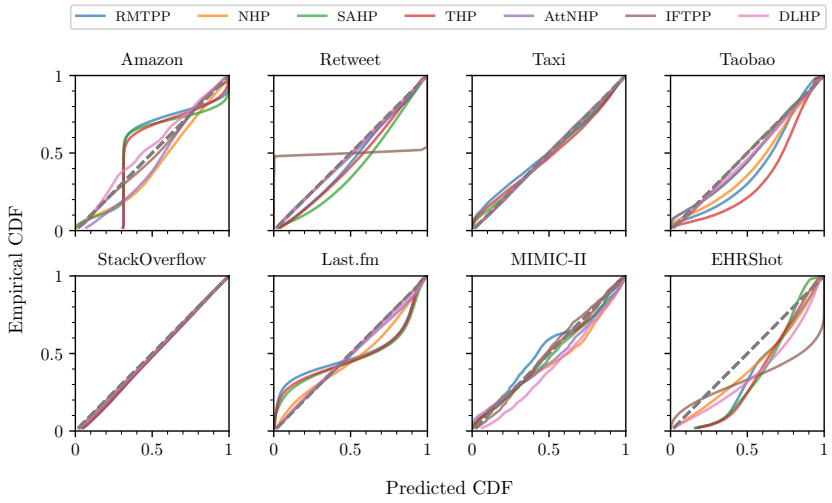

Figure 7: Reliability diagram for predicted inter-arrival time for each model on all datasets. Diagonal dashed lines refer to perfect calibration.

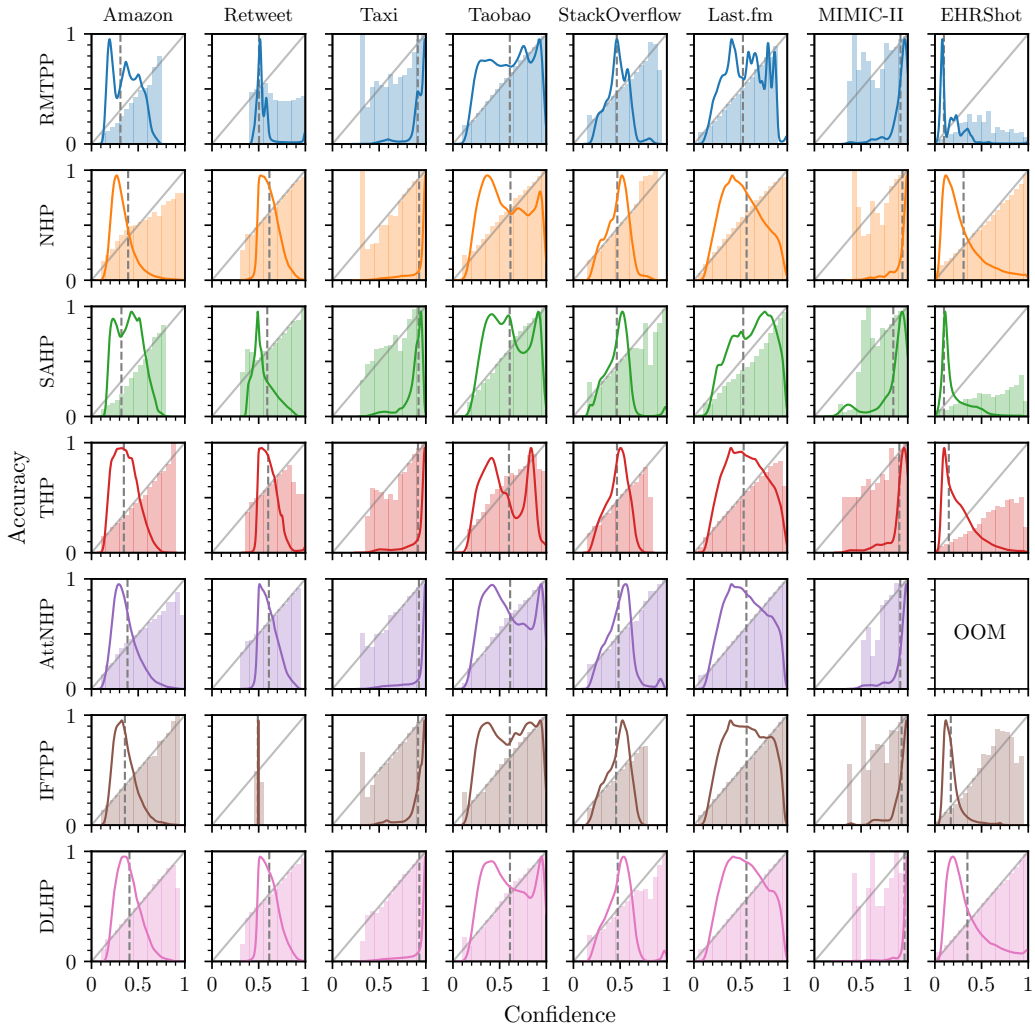

Figure 8: Reliability diagram for mark prediction of all models and all datasets. The x-axis specifies the confidence of model estimates grouped into 20 bins, and the y-axis of the bar plot is the model accuracy within that bin. The diagonal lines represent perfect calibration. The solid curves depict the distribution of confidences, and do not share the y-axis. The grey dashed lines indicate the overall prediction accuracy of the model for the next event conditioned on true event time.

Finally, in Figs. 9 and 10 we plot the log-likelihood of time and mark respectively, versus their corresponding calibration results, to provide an overall view of the performances of different models. Our DLHP model consistently achieves higher log-likelihood while maintaining good calibration on both time and mark components on most datasets.

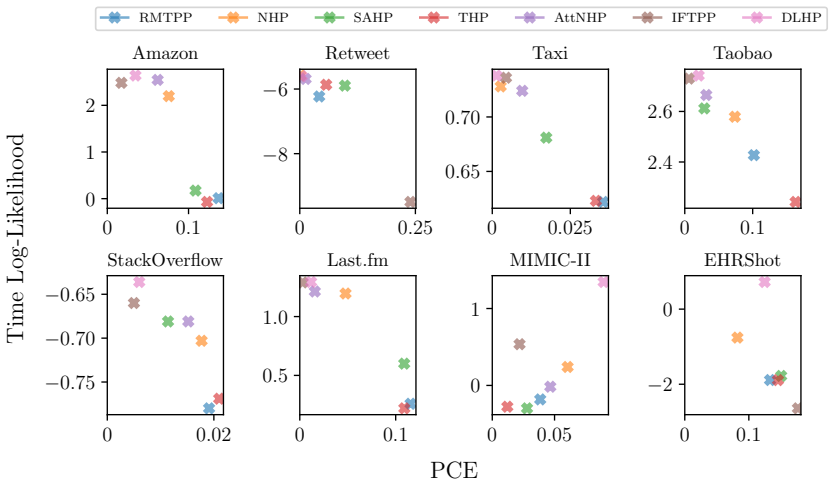

Figure 9: Log-likelihood of time vs. PCE for all models grouped by datasets. Higher log-likelihood and lower PCE are better (i.e. top left corner).

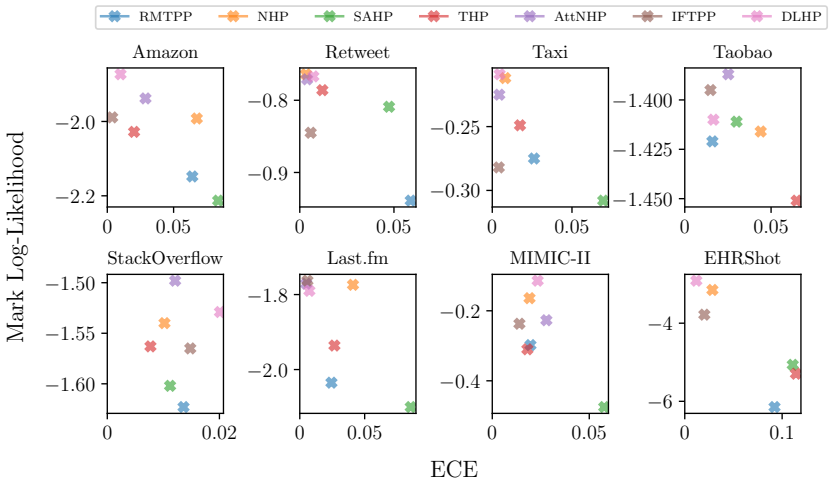

Figure 10: Log-likelihood of mark vs. ECE for all models grouped by datasets. Higher log-likelihood and lower ECE are better (i.e. top left corner).