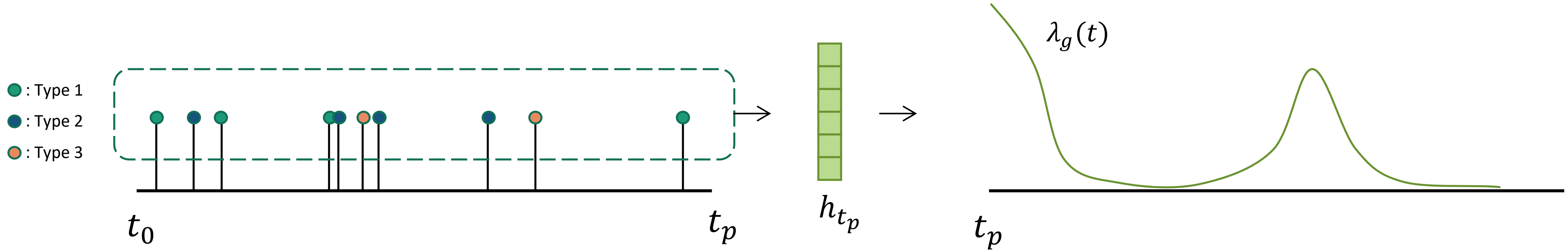
Decoupled Marked Temporal Point Process using Neural ODEs

Yujee Song, Lee Dong Hyun, Rui Meng, Won Hwa Kim

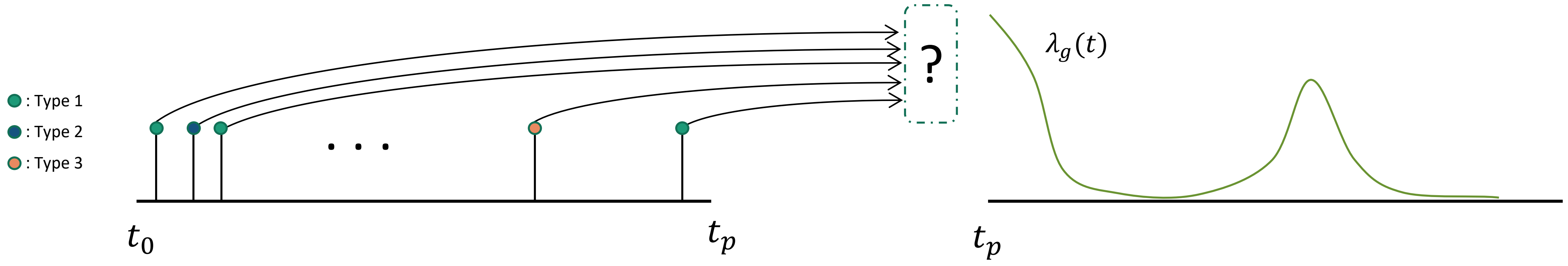
POSTECH
POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

amazon

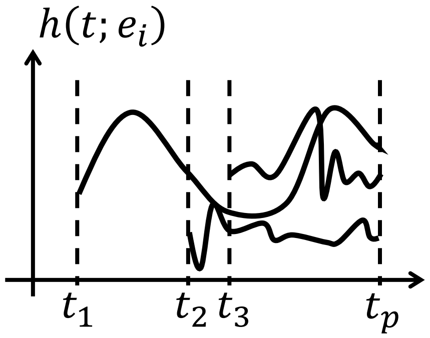
Neural Marked Temporal Point Process: very successful



But how does an individual event contribute…?

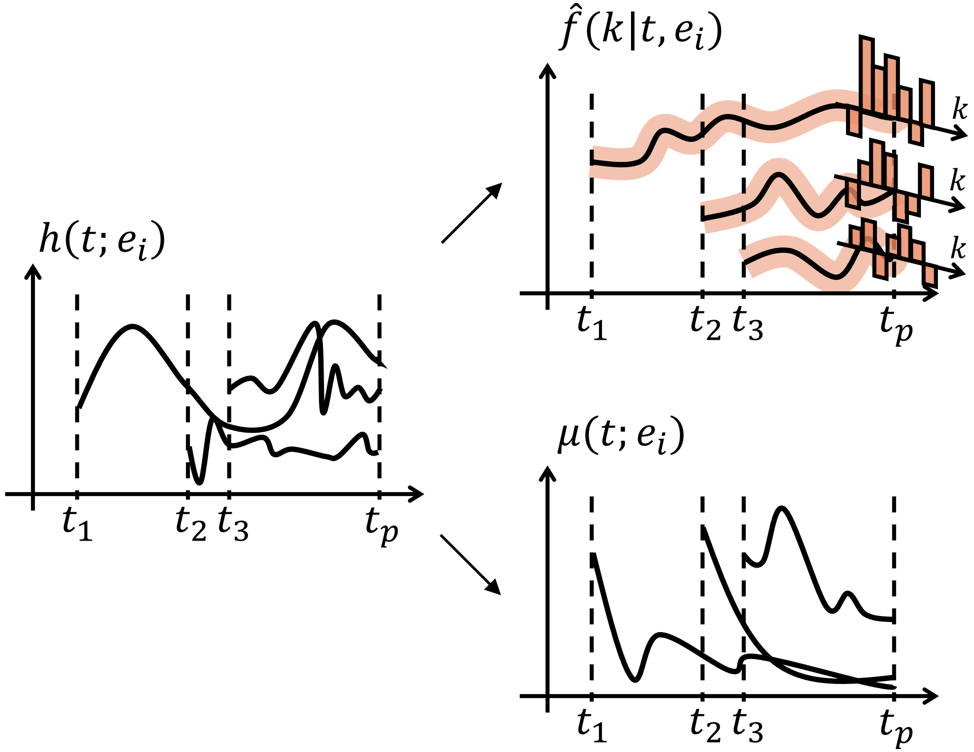


Dec-ODE



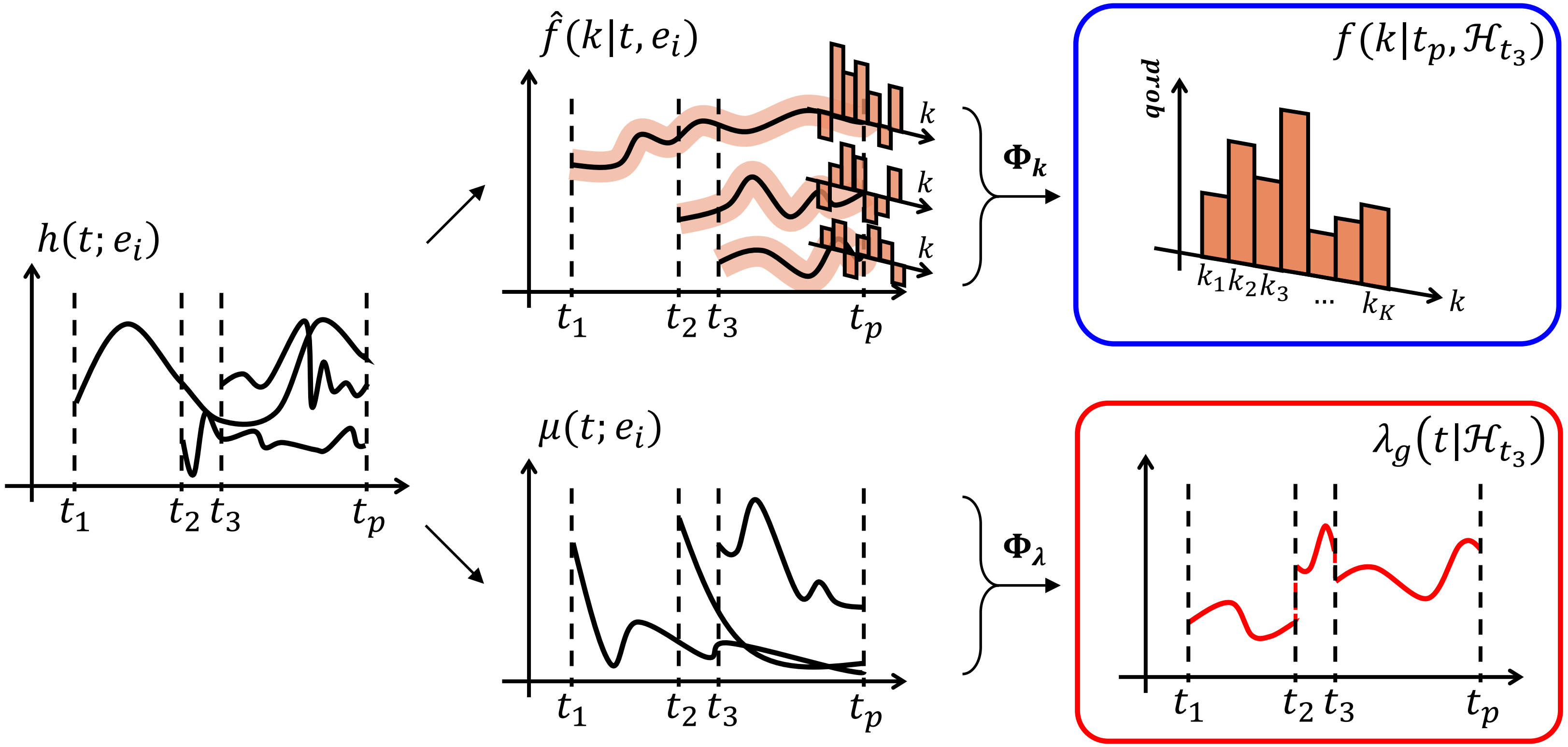
$$
\begin{aligned}
dh(t;e_i) &= \gamma(h(t;e_i), t, k_i; \theta)dt \\
h(t;e_i) &= h(t_i;e_i) + \int_{t_i}^{t} \gamma(h(s;e_i), s, k_i; \theta)ds \\
&= W_e(k_i) + \int_{t_i}^{t} \gamma(h(s;e_i), s, k_i; \theta)ds
\end{aligned}
$$

Dec-ODE



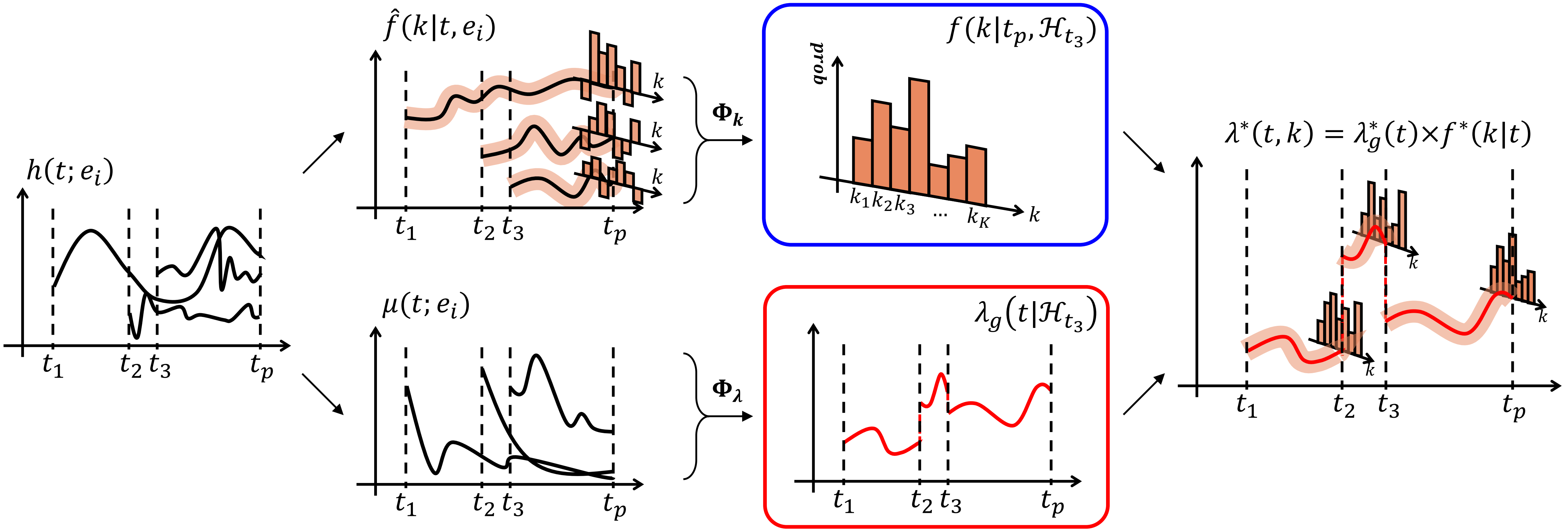
$$\lambda^*(t,k) = \lambda_g^*(t) f^*(k|t) = \frac{f(t|\mathcal{H}_t) f^*(k|t)}{1 - F(t|\mathcal{H}_t)} = \frac{f(t,k|\mathcal{H}_t)}{1 - F(t|\mathcal{H}_t)}$$

Dec-ODE



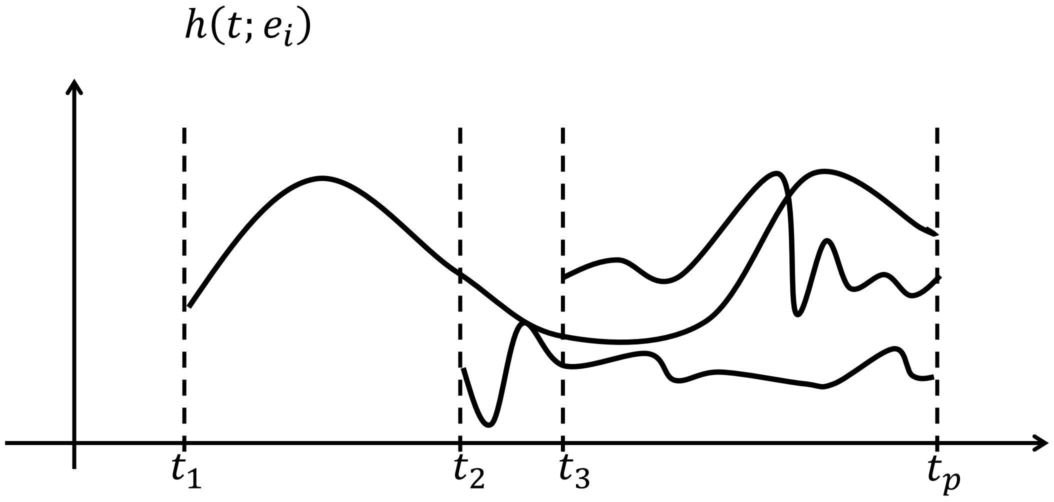
$$\ln f(t,k) = \ln\left[\prod_{i=1}^{\mathcal{N}_g(t_N)} \lambda_g^*(t_i)\right]\left[\prod_{i=1}^{\mathcal{N}_g(t_N)} f^*(k_i|t_i)\right]\exp\left(-\int_0^{t_N}\lambda_g^*(u)du\right)$$
$$= \underbrace{\sum_{i=1}^{\mathcal{N}_g(t_N)} \ln \lambda_g^*(t_i) - \int_0^{t_N}\lambda_g^*(u)du}_{\ln L_\lambda} + \underbrace{\sum_{i=1}^{\mathcal{N}_g(t_N)} \ln f^*(k_i|t_i)}_{\ln L_k}$$

Dec-ODE

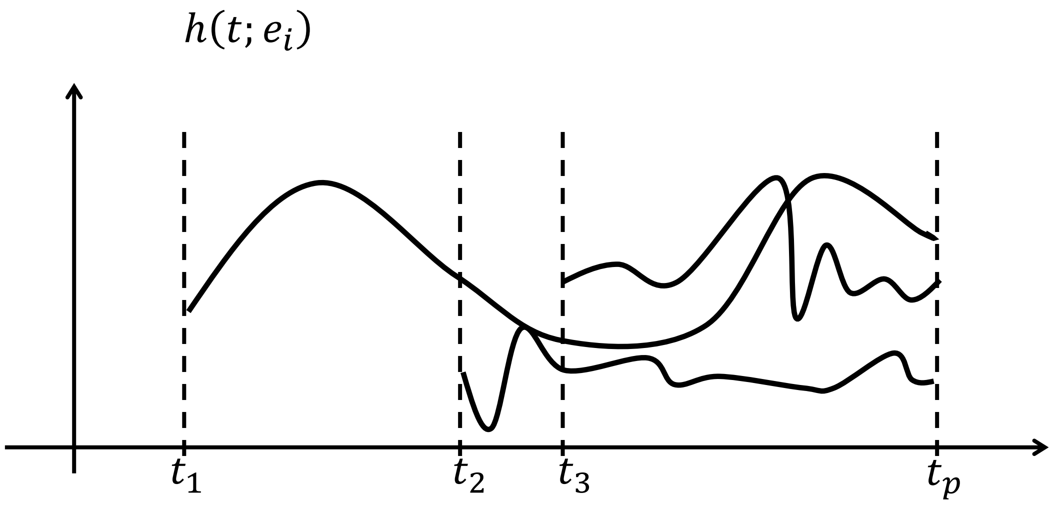


$h(t;e_i)$
t_1 t_2 t_3 t_p
$\hat{f}(k|t,e_i)$
k
$\mu(t;e_i)$
Φ_k
Φ_λ
$f(k|t_p,\mathcal{H}_{t_3})$
prob
$k_1 k_2 k_3$ … k_K
$\lambda_g(t|\mathcal{H}_{t_3})$
$\lambda^*(t,k) = \lambda_g^*(t) \times f^*(k|t)$

Efficient training



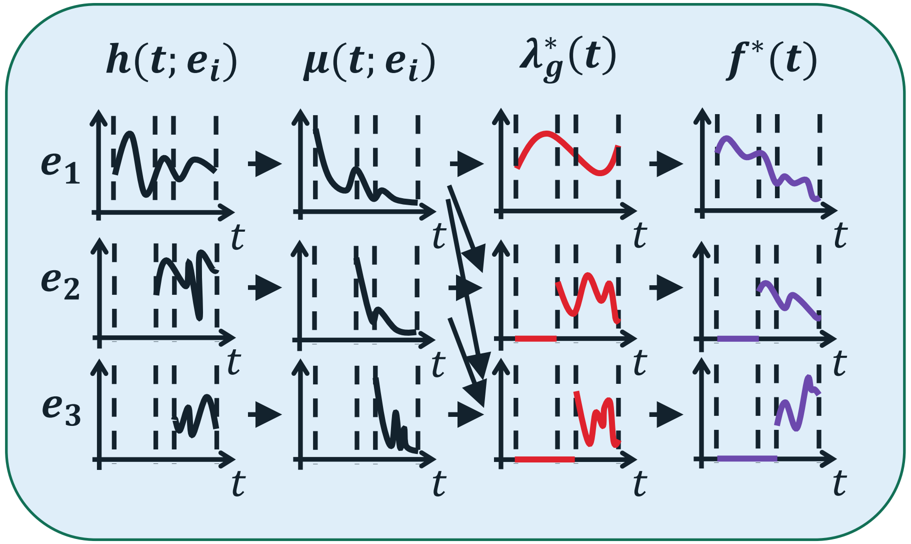
Efficient training



$$\frac{d}{dt}\mathbf{h}(\mathbf{t}) = \frac{d}{dt}\begin{bmatrix} h(t; e_0) \\ \vdots \\ h(t; e_i) \end{bmatrix} = \begin{bmatrix} \gamma(h(t; e_0), t, k_0; \theta) \\ \vdots \\ \gamma(h(t; e_i), t, k_i; \theta) \end{bmatrix}$$

Efficient estimation

$$
\frac{\partial}{\partial t}\begin{bmatrix} \mathbf{h}(t) \\ \Lambda_g(t|\mathcal{H}_{t_i}) \\ F(t|\mathcal{H}_{t_i}) \\ \mathbb{E}[t] \end{bmatrix} = \begin{bmatrix} \gamma(\mathbf{h}(t), t, \mathbf{k}; \theta) \\ \lambda_g(t|\mathcal{H}_{t_i}) \\ f(t|\mathcal{H}_{t_i}) \\ t \cdot f(t|\mathcal{H}_{t_i}) \end{bmatrix} = \begin{bmatrix} \gamma(\mathbf{h}(t), t, \mathbf{k}; \theta) \\ \Phi_\lambda(g_\mu(\mathbf{h}(t))) \\ \lambda_g(t|\mathcal{H}_{t_i}) \cdot \exp(\Lambda_g(t_{i-1}|\mathcal{H}_{t_i}) - \Lambda_g(t|\mathcal{H}_{t_i})) \\ t \cdot f(t|\mathcal{H}_{t_i}) \end{bmatrix}
$$

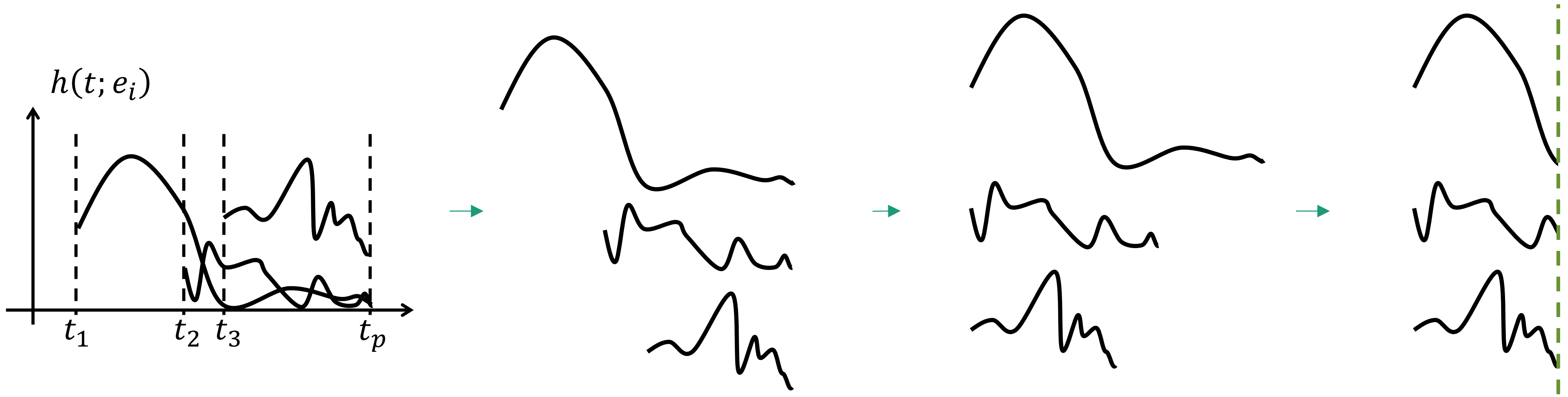


Linear Dec-ODE: simple implementation

$$\Phi_\lambda(\mu(t;e_0),\cdots,\mu(t;e_n)) = \sum_{e_i\in\mathcal{H}_{t_{n+1}}} \text{softplus}(\mu(t;e_i))$$

$$\Phi_k(\hat{f}(k|t,e_0),\cdots,\hat{f}(k|t,e_n)) = \text{softmax}\Bigg(\sum_{e_i\in\mathcal{H}_{t_{n+1}}} \hat{f}(k|t,e_i)\Bigg)$$

Linear Dec-ODE: training scheme



Linear Dec-ODE: training scheme (cont.)

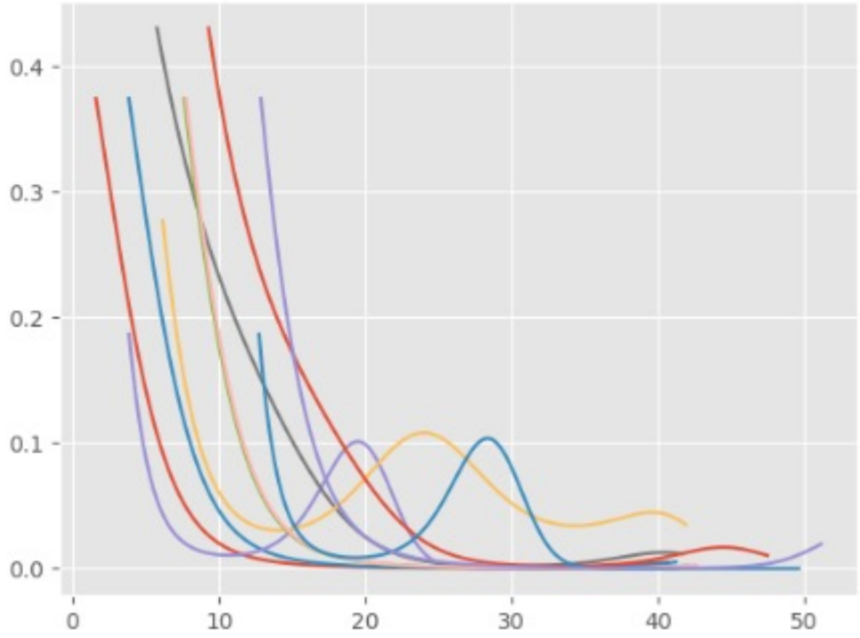
$$\Lambda_g^*(t) = \int_0^t \lambda_g^*(s)ds = \int_0^t \sum_{e_i \in \mathcal{H}_t} \text{softplus}(\mu(s; e_i))ds$$
$$= \sum_{e_i \in \mathcal{H}_t} \int_{t_i}^t \text{softplus}(\mu(s; e_i))ds$$

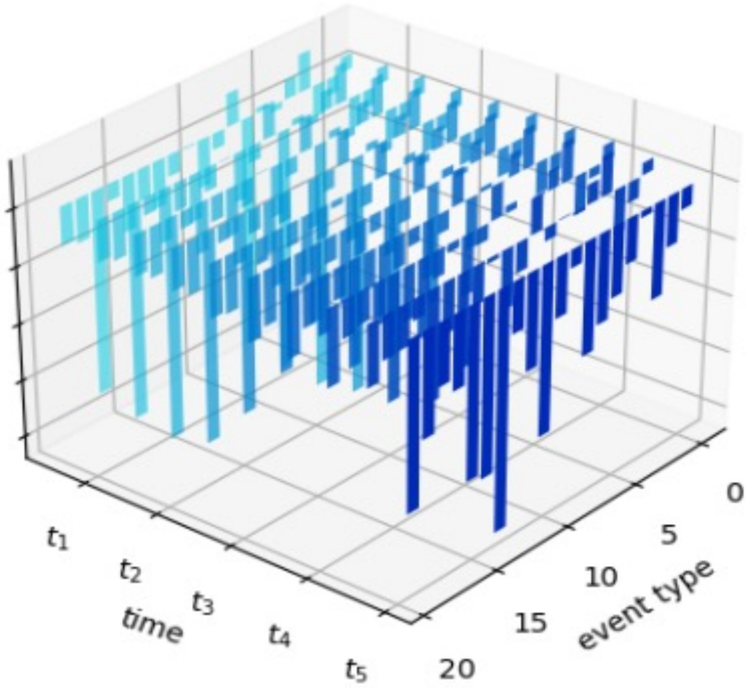
Results: benchmark dataset

	RMSE					ACC					NLL				
	RMTPP	THP	IFL	ANHP	**Dec-ODE**	RMTPP	THP	IFL	ANHP	**Dec-ODE**	RMTPP	THP	IFL	ANHP	**Dec-ODE**
MOOC	0.473 (0.012)	0.476 (0.010)	0.501 (0.012)	0.470 (0.019)	**0.467 (0.012)**	20.98 (0.29)	24.49 (0.22)	32.30 (1.30)	31.53 (0.20)	**42.08 (0.44)**	−0.315 (0.031)	0.733 (0.047)	**-2.895 (0.031)**	−2.632 (0.043)	−2.289 (0.191)
Reddit	0.953 (0.016)	6.151 (0.195)	1.040 (0.017)	1.149 (0.010)	**0.934 (0.017)**	29.67 (1.19)	60.72 (0.08)	48.91 (1.27)	**63.45 (0.16)**	62.32 (0.11)	3.559 (0.070)	2.335 (0.031)	2.188 (0.088)	**1.203 (0.068)**	1.367 (0.126)
Retweet	0.990 (0.016)	1.055 (0.015)	1.012 (0.018)	1.663 (0.014)	**0.985 (0.016)**	51.72 (0.33)	**60.68 (0.11)**	55.35 (0.19)	59.72 (0.11)	60.17 (0.23)	−2.180 (0.025)	−2.597 (0.016)	−2.672 (0.023)	**-3.134 (0.019)**	−2.897 (0.030)
MIMIC-II	0.859 (0.093)	> 10 (0.114)	1.005 (0.121)	0.933 (0.088)	**0.810 (0.173)**	78.20 (5.00)	**85.98 (2.56)**	80.49 (5.20)	84.30 (2.78)	85.06 (3.65)	1.167 (0.150)	5.657 (0.304)	**0.939 (0.139)**	1.025 (0.155)	1.354 (0.413)
Stack Overflow	**1.017 (0.011)**	1.057 (0.011)	1.340 (0.013)	1.052 (0.011)	1.018 (0.011)	53.95 (0.32)	53.83 (0.18)	53.00 (0.35)	**56.80 (0.18)**	55.58 (0.29)	2.156 (0.022)	2.318 (0.022)	2.314 (0.020)	**1.873 (0.017)**	2.063 (0.016)

Comparison with the state of the art methods on 5 popular real-life datasets. Results with boldface and underline represent the best and the second-best results, respectively.

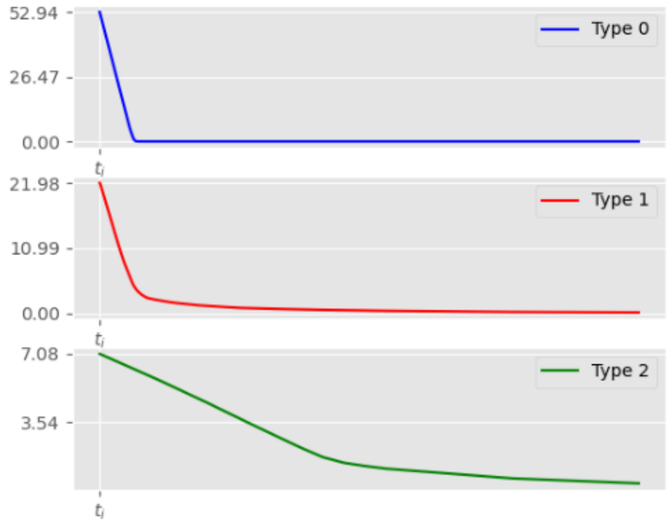
Results: visualization



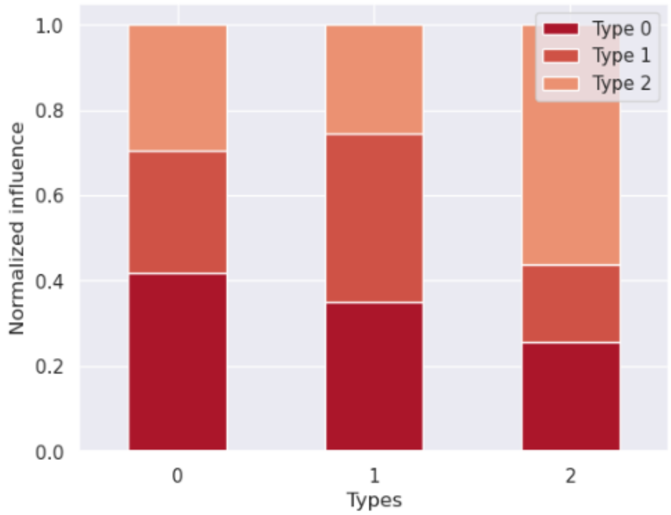


Complex dynamics in both the temporal and mark aspects trained using Stack Overflow dataset.

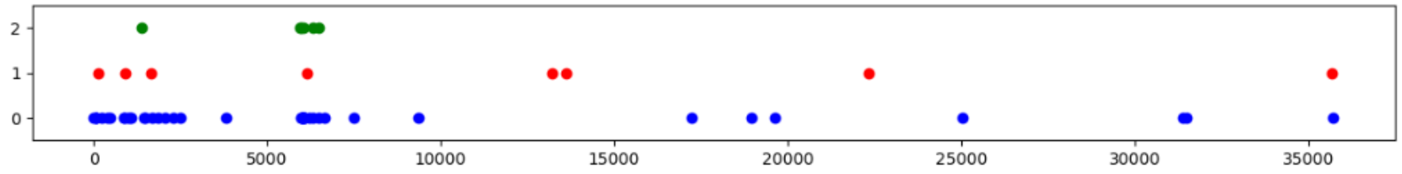
Results: interpretability



(a) Influence function $\mu(t)$ conditioned on different event types plotted on the same time range.



(b) Averaged proportion of influence from different marks on specific marks.



(c) Visualization of events, i.e., marks across time, affecting each other. Blue(0): users with small followers, Red(1): users with medium followers, Green(2): users with large followers.

Results: efficiency

StackOverflow			MIMIC-II			Retweet			Reddit		
parallel	Sequential	Ratio	parallel	Sequential	Ratio	parallel	Sequential	Ratio	parallel	Sequential	Ratio
15.0	57.7	0.26	2.9	6.5	0.47	16.8	67.6	0.25	15.5	78.7	0.20

Training efficiency comparison in different datasets with the average time taken for an iteration (sec/iter) as the metric.
Ratio shows that the iteration time at least reduces in half.