Invited talk for: amazon alexa

Hawkes Process Memory RNN

by: Denis Kazakov [a]
advisor: Michael Mozer [a]
worked with: Rob Lindsey [b]

[a] University of Colorado, Boulder
[b] Imagen Technologies

Outline

1. Motivation:
    1. Inductive bias
    2. Sequence processing domain overview
2. Prerequisites on theory
3. Building the model & intuition
4. Results & analysis

Inductive bias in machine learning

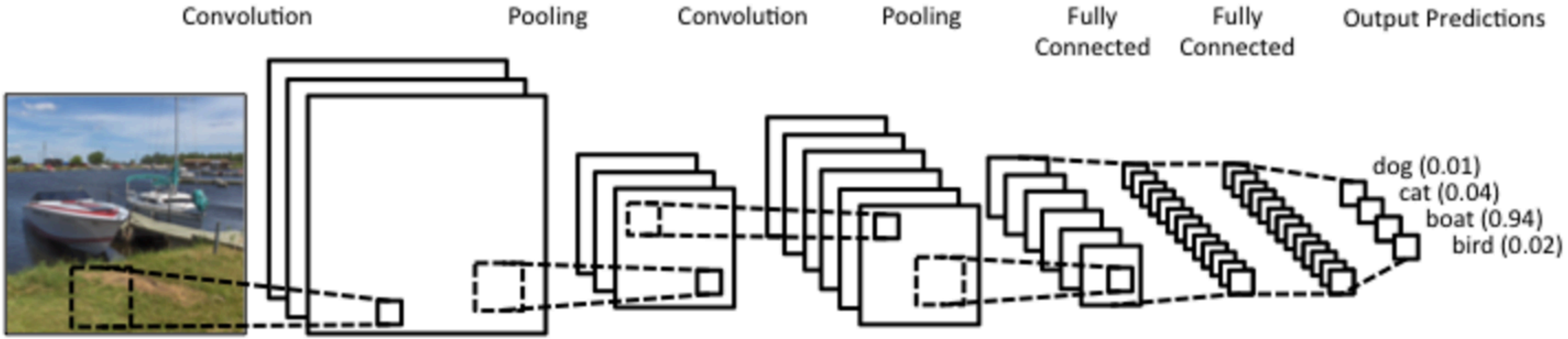
f

Generator

$\hat{f}$

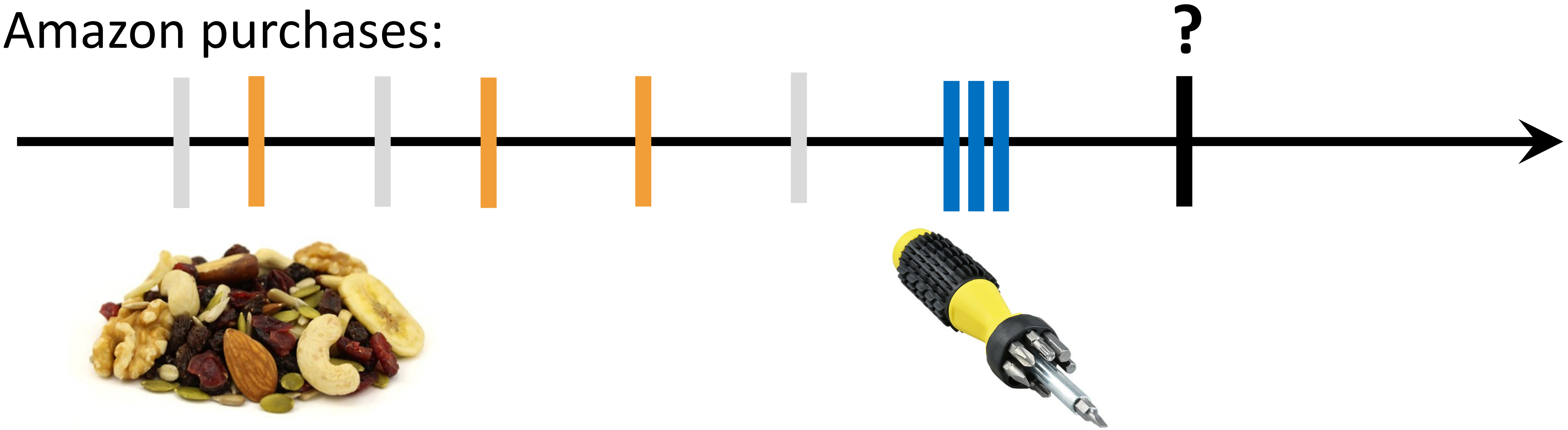
Approximation/Discriminator
with a biased capacity to learn

Inductive bias in machine learning: CNN



Event sequences

Amazon purchases:



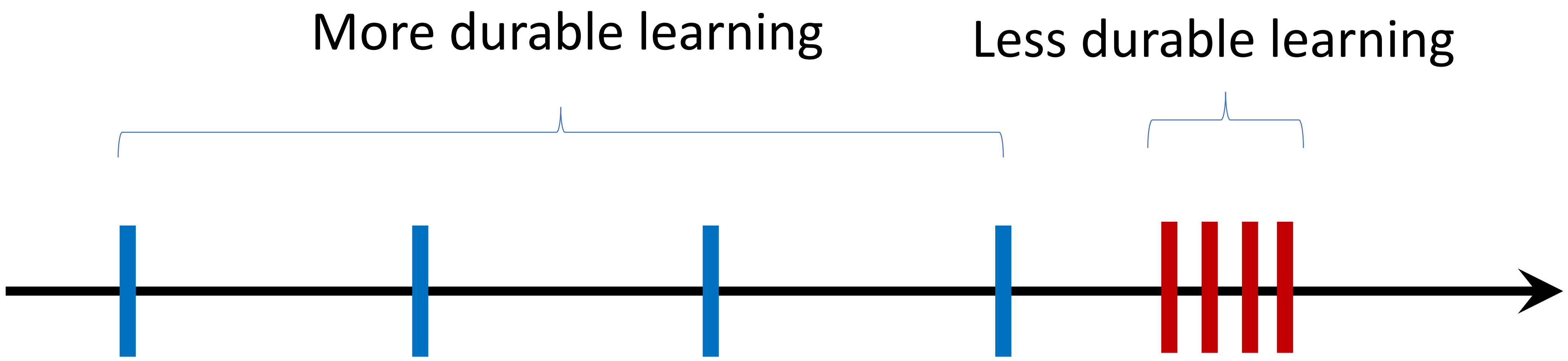
Music selection
Text messaging
Online postings

Inductive bias in machine learning: event sequences

- CNN over time domain (Cui et. al.) – poor scaling to multiple timescales (milliseconds vs days).
- RNN with time as input feature – time is used implicitly, not an inductive bias. Potentially too flexible.
- Probabilistic processes – time built into the model, but poor feature learning ability.

Merge deep learning feature learning ability with probabilistic process's continuous time handling?

Motivation: Time Scales & Human Memory Decay



Kitty – котенок [*kotyonok*]

Point Processes

Homogeneous Poisson Process:

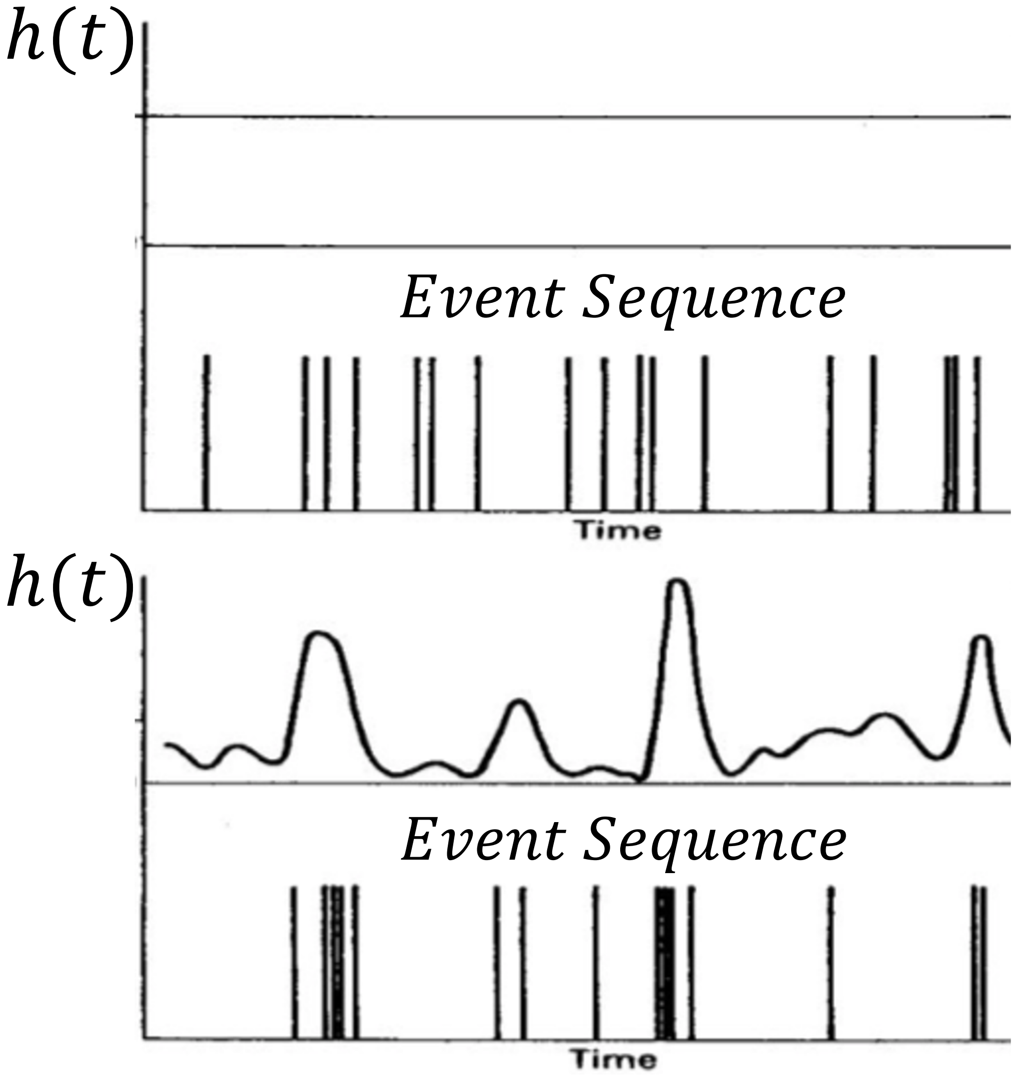
Intensity: $h(t) = \lambda$

Time between arrivals: $X \sim Exp(\lambda)$

Expected number of event: $E[X] = \frac{1}{\lambda}$

Nonhomogeneous Poisson Process:

Intensity is a function of time.

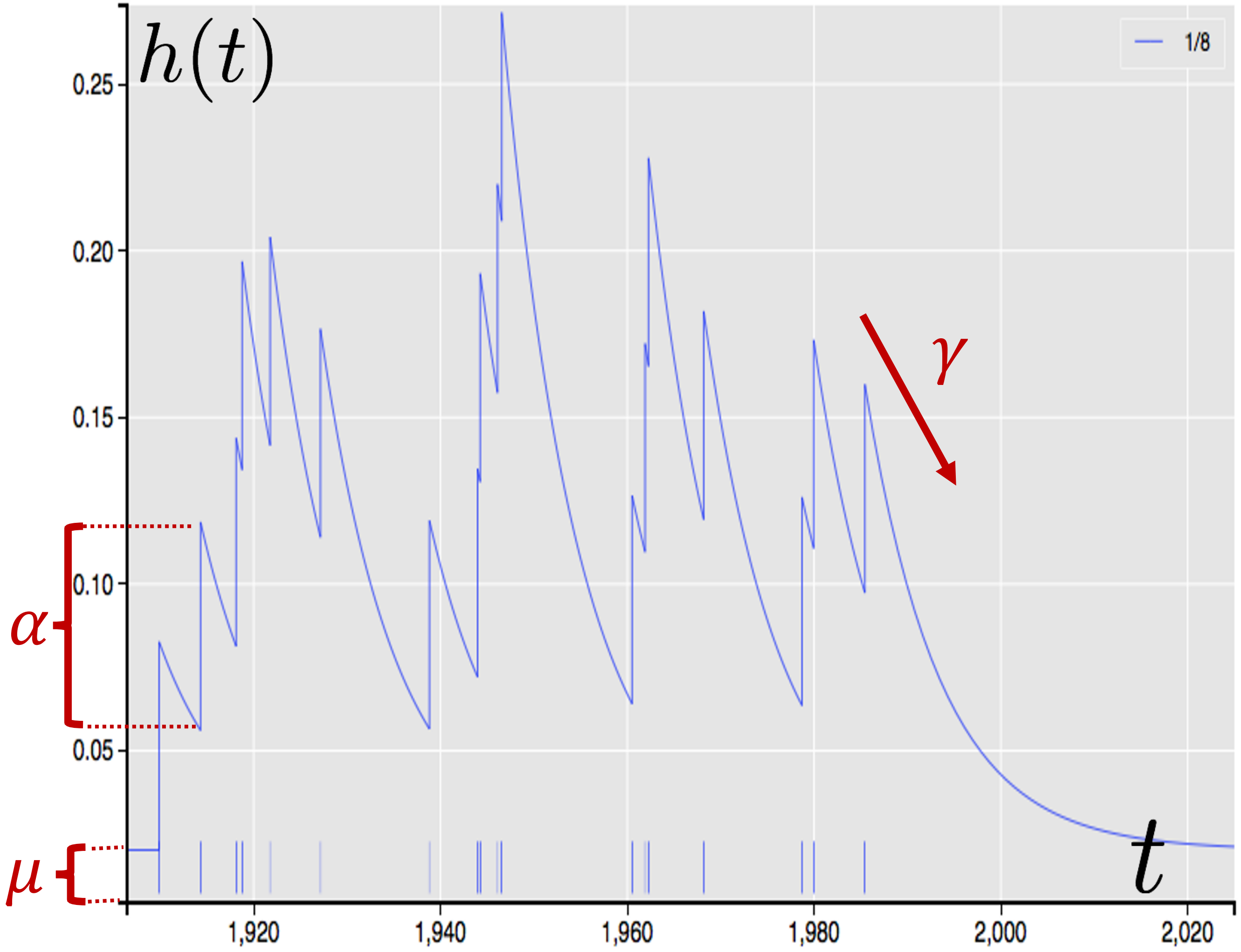


Hawkes Process

A point process … with a twist:
Self excitatory, conditional intensity function with an exponential decay:

$$h(t) = \mu + \alpha \sum_{t_j < t} e^{-\gamma(t - t_j)}$$

μ – *baseline intensity*
α – *"jump" rate*
γ – *decay rate*



Expectation of the Intensity

$$h(t) = \mu + \alpha \sum_{t_j < t} e^{-\gamma(t - t_j)}$$

μ – *baseline intensity*

α – *"jump" rate*

γ – *decay rate*

$$\lim_{t \to \infty} \mathbf{E}[h(t)] = \frac{\mu}{1 - \frac{\alpha}{\gamma}}$$

Takeaway: given μ, for $i, j \in N$: if $\frac{\alpha_i}{\gamma_i} = \frac{\alpha_j}{\gamma_j} = const$,

$$\lim_{t \to \infty} E[h_i(t)] = \lim_{t \to \infty} E\left[h_j(t)\right]$$

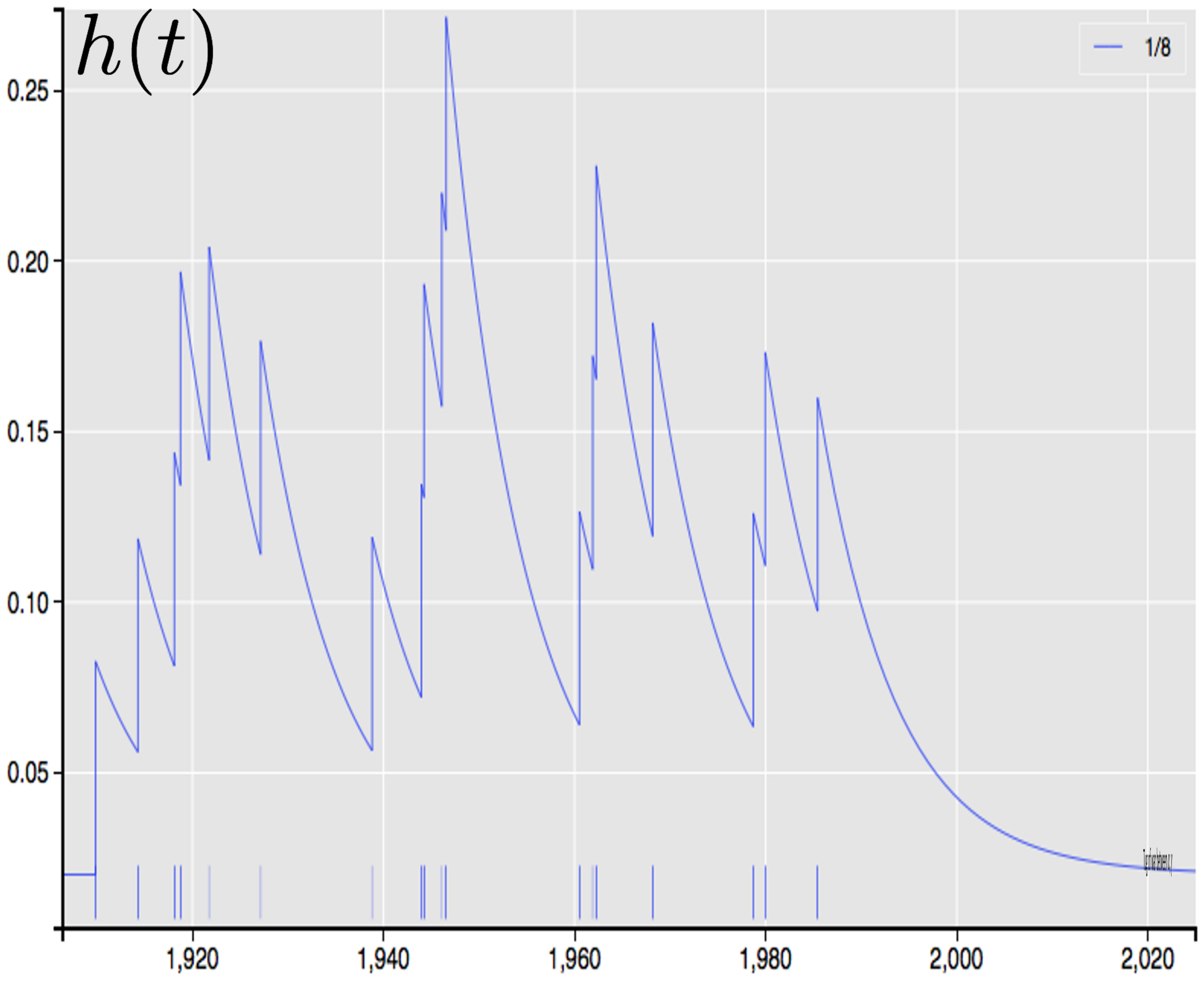
Hawkes Process Divergence

$$h(t) = \mu + \alpha \sum_{t_j < t} e^{-\gamma(t - t_j)}$$

Tug of war between α, γ

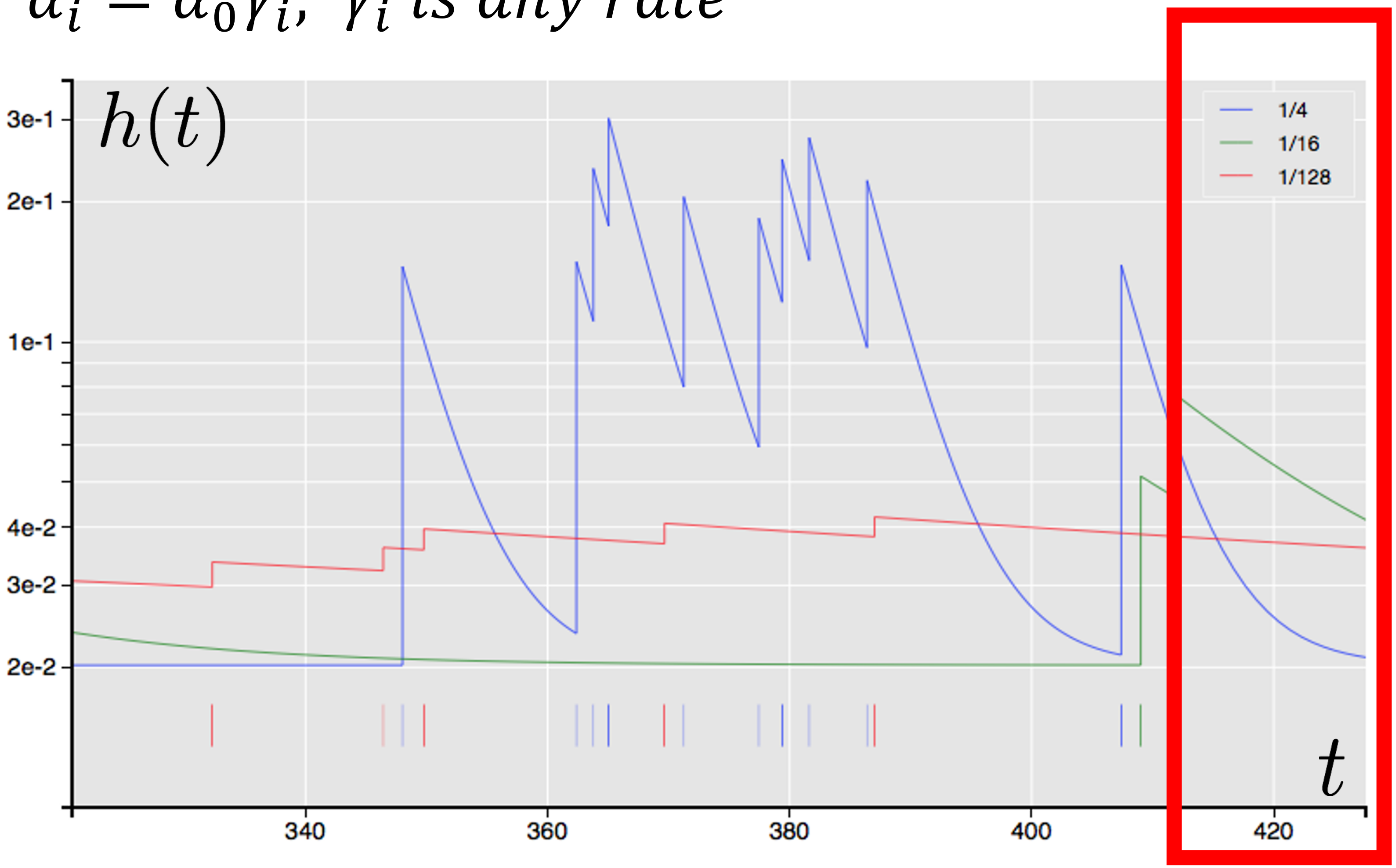
Therefore, force $\boldsymbol{\alpha} < \boldsymbol{\gamma}$.

(derived from conditional expectation formula)



Controlling a Hawkes Process

- $\alpha < \gamma$ or $\alpha_i = \alpha_0 \gamma_i,\ \gamma_i$ *is any rate*



Exact Simulation of Hawkes Process

Conditional intensity function: $$h(t) = \mu + \alpha \sum_{t_j < t} e^{-\gamma(t - t_j)}$$

1. Initialize:

   $h_0 = \mu, \quad t_0 = 0 \quad , \Delta t_k \equiv t_k - t_{k-1}$

2. Decay the intensity with each event:

   $$h_k = \underbrace{\mu + e^{-\gamma \Delta t_k}(h_{k-1} - \mu)}_{H_k(\Delta_{t_k})} + \alpha\gamma x_k, \quad \text{where } x_k = \begin{cases} 1, & \text{event occurs} \\ 0, & \text{else} \end{cases}$$

3. Probability of the next event x_k occurring after the current time - t_{k-1} within the time window of Δt.

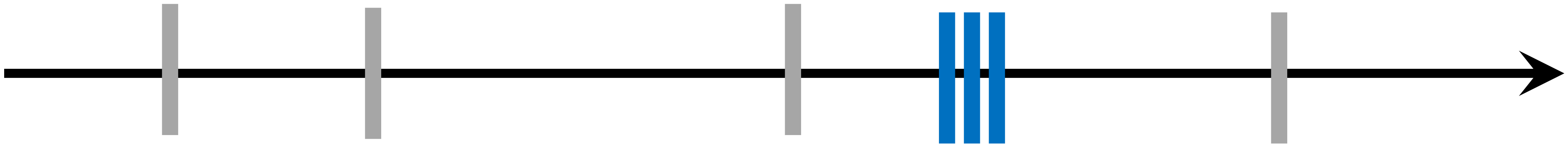
$$P(t_k \leq t_{k-1} + \Delta t | t_{1:k-1}) = 1 - \overbrace{P(t_k > t_{k-1} + \Delta t | t_{1:k-1})}^{Z_k(\Delta_t)} = 1 - e^{-\int_0^{\Delta t} h_{k-1} dt} \tag{5}$$

$$= 1 - e^{-\frac{(h_{k-1} - \mu)(1 - e^{-\gamma \Delta t})}{\gamma} - \mu \Delta t}$$

Scale Inference for Hawkes Process

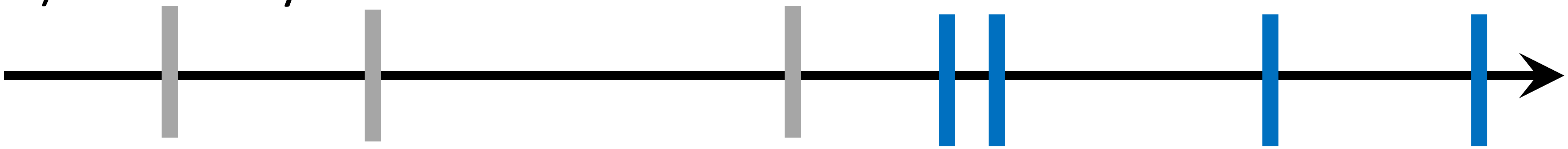
In music. You heard a catchy song, then:

1) Going on a binge immediately and forget about it



OR

2) Discover your new favorite artist to listen for weeks on end



Scale Inference for Hawkes Process

- Approximate with discrete values on a log-scale: $\gamma_i \in [\gamma_1, \gamma_2, \ldots, \gamma_S]$
- Simulate S Hawkes processes

$h_{0,i} = \mu.$

$history_i \equiv (x_i, t_i)$ defines the events and their respective times.

$P(\gamma_i) = \frac{1}{S}$ - initial belief is uniform across all γ's.

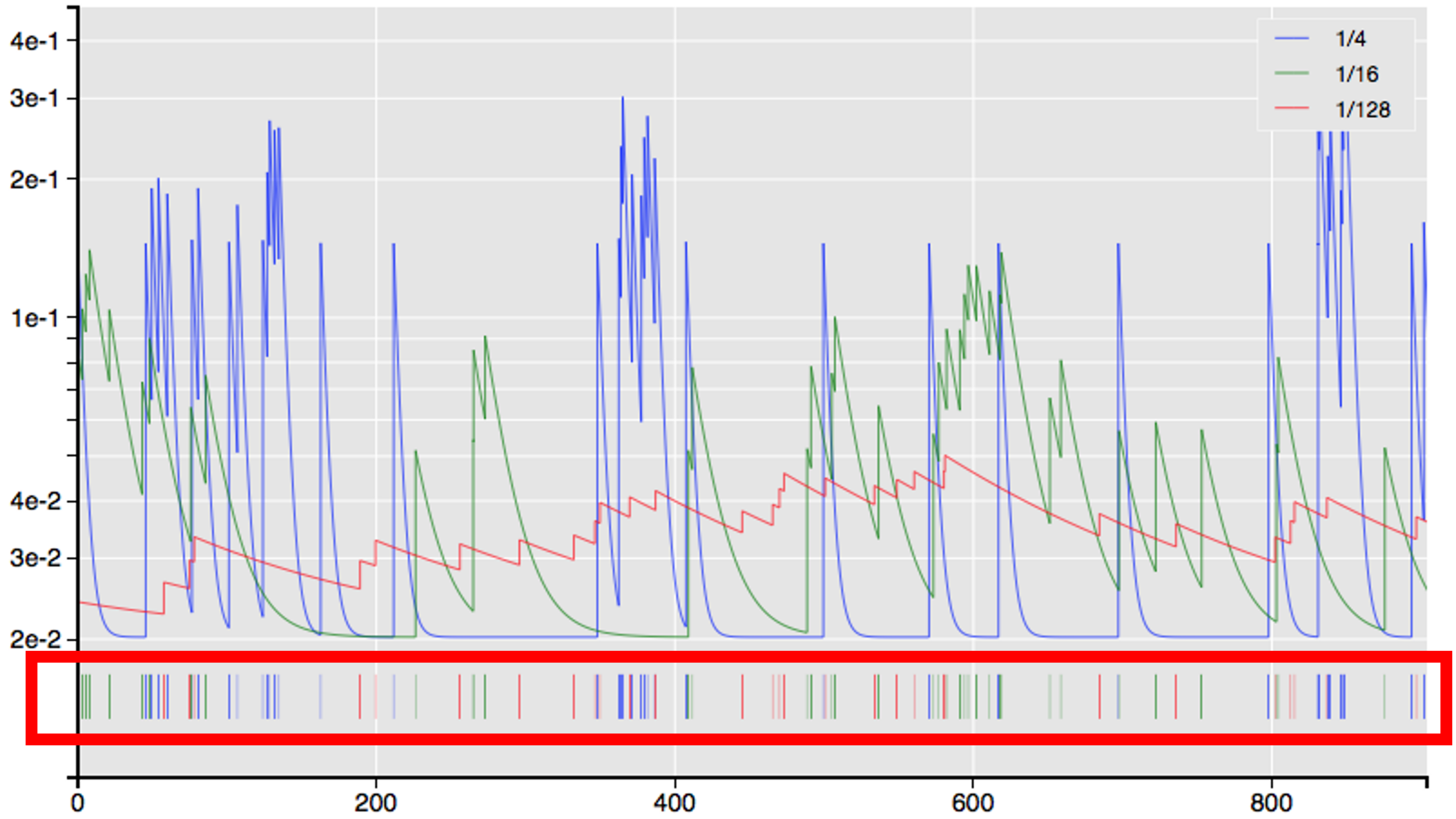
$$C_{k,i} \equiv P(\gamma_i | history_{1:k})$$

$$H_{k,i}(\Delta t_k) \equiv \mu + e^{-\gamma_i \Delta t_k}(h_{k-1,i} - \mu)$$

$$Z_{k,i}(\Delta t) \equiv P(t_k \geq t_{k-1} + \Delta t | t_{1:k-1})$$

$$P(\gamma_i | history_{1:k}) \sim P(history_k | history_{1:k-1}, \gamma_i) P(\gamma_i | history_{1:k-1})$$
$$\sim H_{k,i}(\Delta t_k)^{x_k} Z_{k,i}(\Delta\ t_k) C_{k-1,i}$$

Scale Inference for Hawkes Process



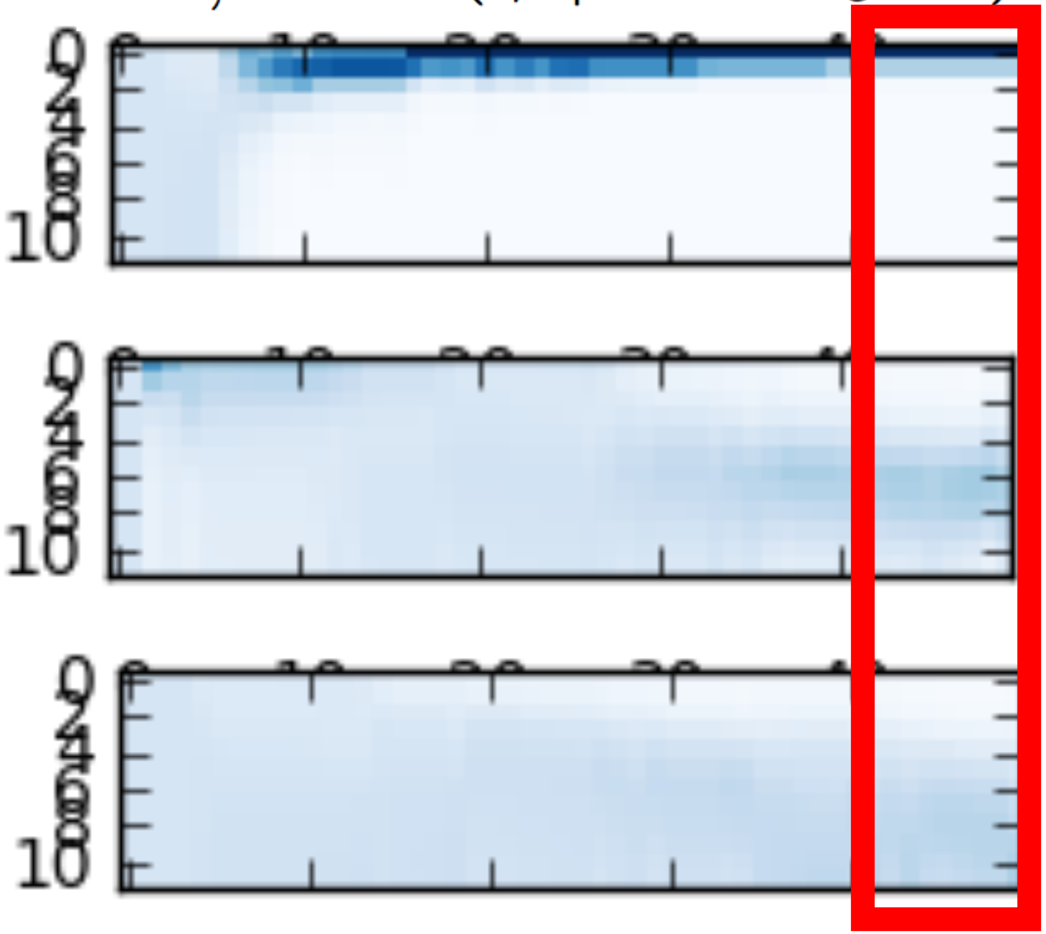
Scale Inference for Hawkes Process

$$C_{k,i} \equiv P(\gamma_i | history_{1:k})$$

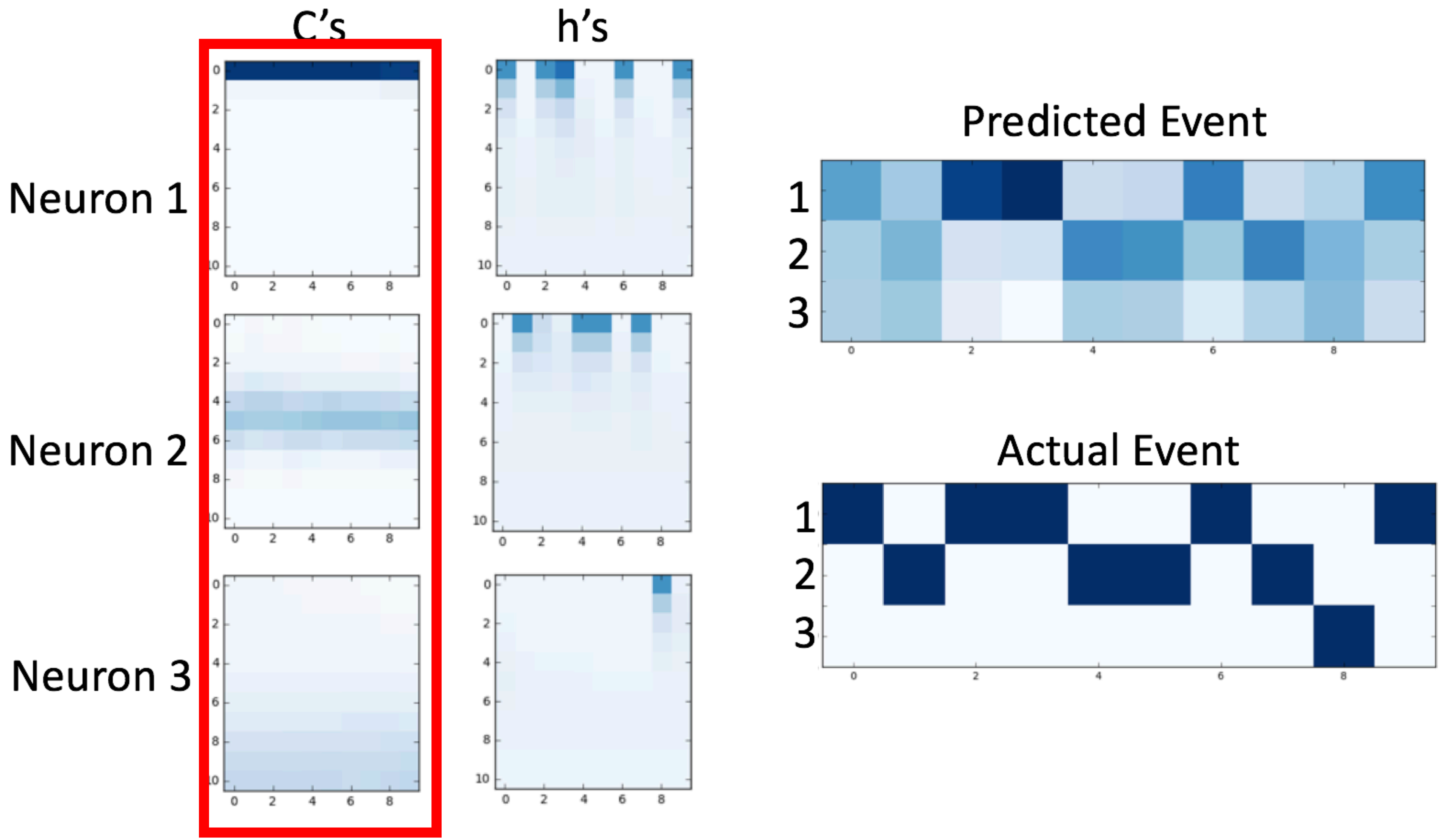
Hawkes Process 3

Hawkes Process 2

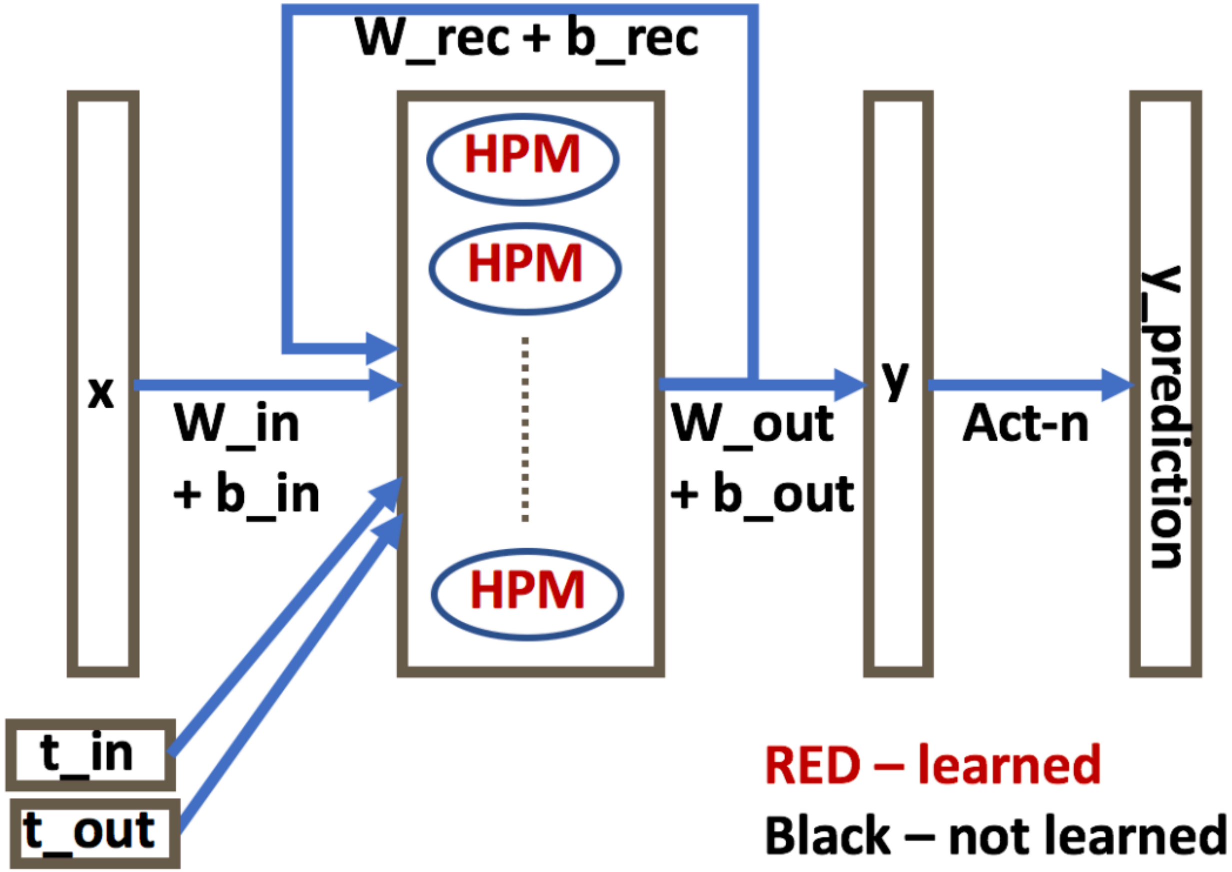
Hawkes Process 1



Scale Inference for Hawkes Process



HPM Model (Plain Hawkes process or “1-to-1”)



Where:

x - one-hot embedding of the sequence element

$P(x) = W_{in}x$ - input into HPM cells,

$W_{in}, W_{out} = IdentityMatrix$,

$W_{rec}, b_{rec}, b_{out}, bin = Zeros$,

Act-n = normalization of output.

What event happened…?

Don’t know timescales -> Infer them
Don’t know if an event happened -> ?

$$h_{k,i} = \mu + e^{-\gamma_i \Delta t_k}(h_{k-1,i} - \mu) + \alpha\gamma_i x_k \quad \text{,where } x_k = \begin{cases} 1, & \text{event occurs} \\ 0, & \text{else} \end{cases}$$

$$P(\gamma_i | history_{1:k}) \sim H_{k,i}(\Delta t_k)^{x_k} Z_{k,i}(\Delta\ t_k) C_{k-1,i}$$

Marginalize over Event probability

$$h_{k,i} = \mu + e^{-\gamma_i \Delta t_k}(h_{k-1,i} - \mu) + \alpha\gamma_i P(x_k)$$

$$P(\gamma_i | history_{1:k}) \sim \sum_{x_k \in \{0,1\}} P(x_k) H_{k,i}(\Delta t_k)^{x_k} Z_{k,i}(\Delta\ t_k) C_{k-1,i}$$

HPM Model Formulation

1. **Initialize:** $\gamma_i \in [\gamma_1, \gamma_2, ..., \gamma_S]$, $h_{0,i} = \mu$, $c_{0,i} = \frac{1}{S}$

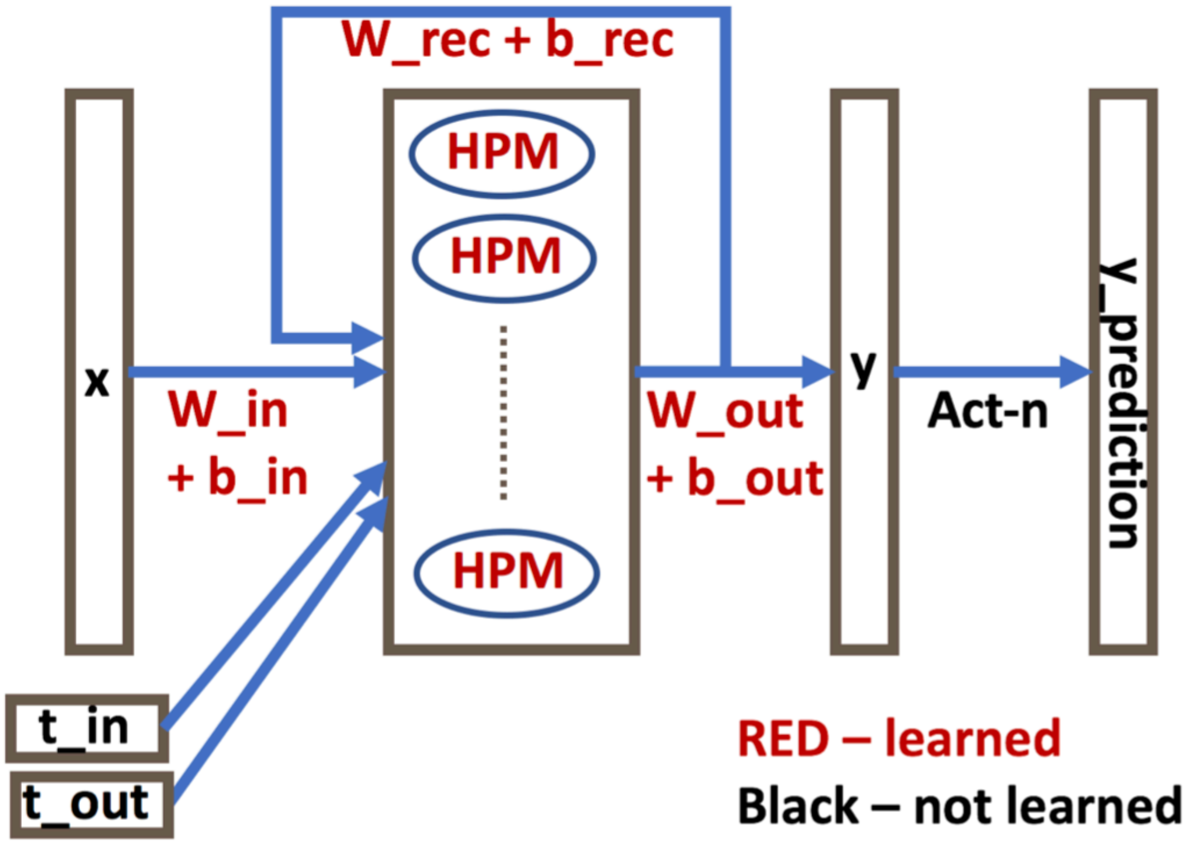
2. **Event occurrence:**
   $P(x_k) = f(input_k)$

3. **Update time-scale posterior**
   $C_{k,i} = \sum_{x_k \in \{0,1\}} P(x_k) \frac{H_{k,i}(\Delta t_k)^{x_k} Z_{k,i}(\Delta\ t_k) C_{k-1,i}}{\sum_j H_{k,j}(\Delta t_k)^{x_k} Z_{k,j}(\Delta\ t_k) C_{k-1,j}}$

4. **Update intensity:**
   $h_{k,i} = H_{k,i}(\Delta t_k) + \alpha \gamma_i P(x_k)$

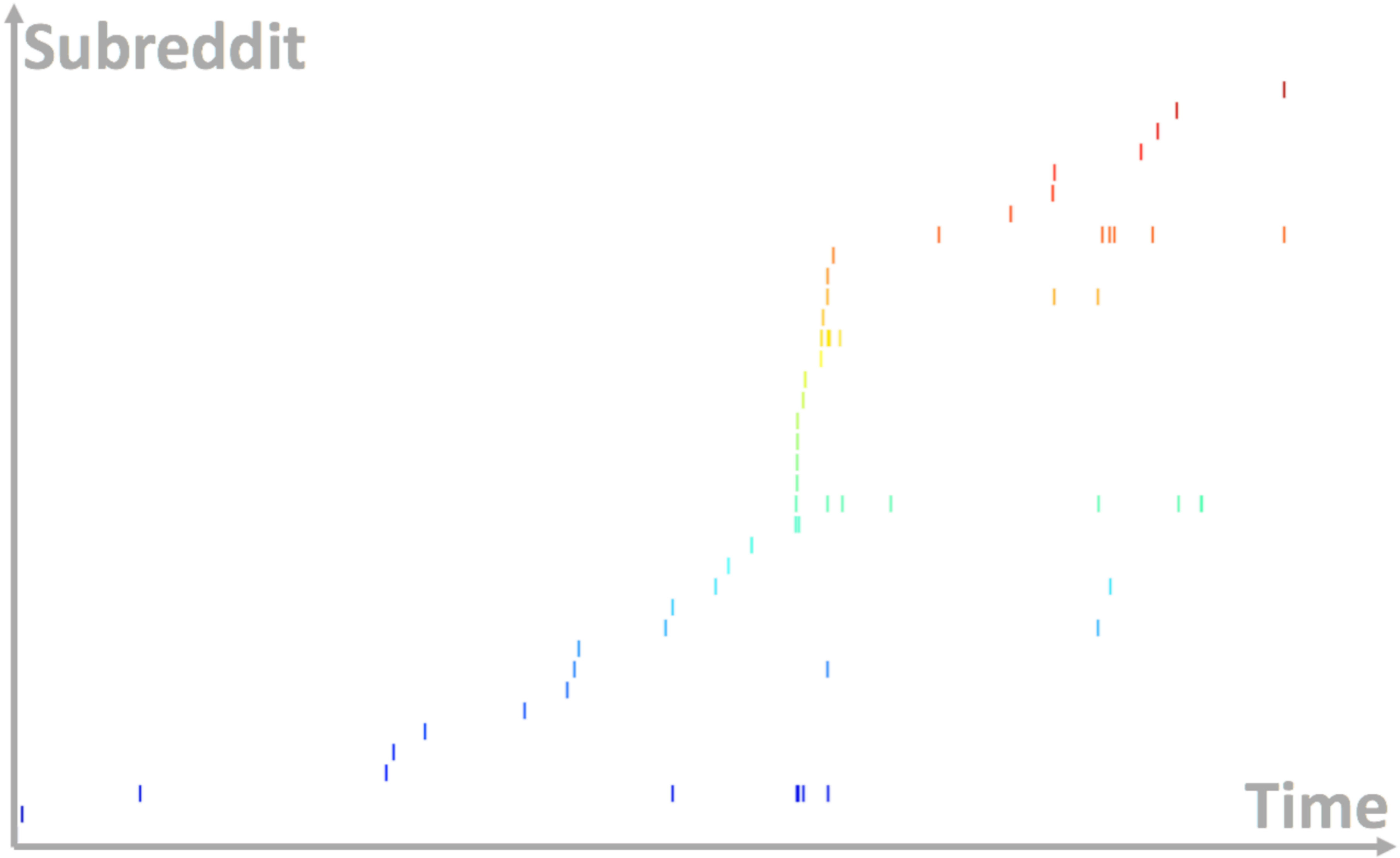
5. **Cell's output to predict event at Δt_{k+1} and for recurrent information for next step:**
   $y_k(\Delta t_{k+1}) = \sum_{i \in S} C_{k,i} Z_{k+1,i}(\Delta t_{k+1})$

HPM Model ("1-to-all")



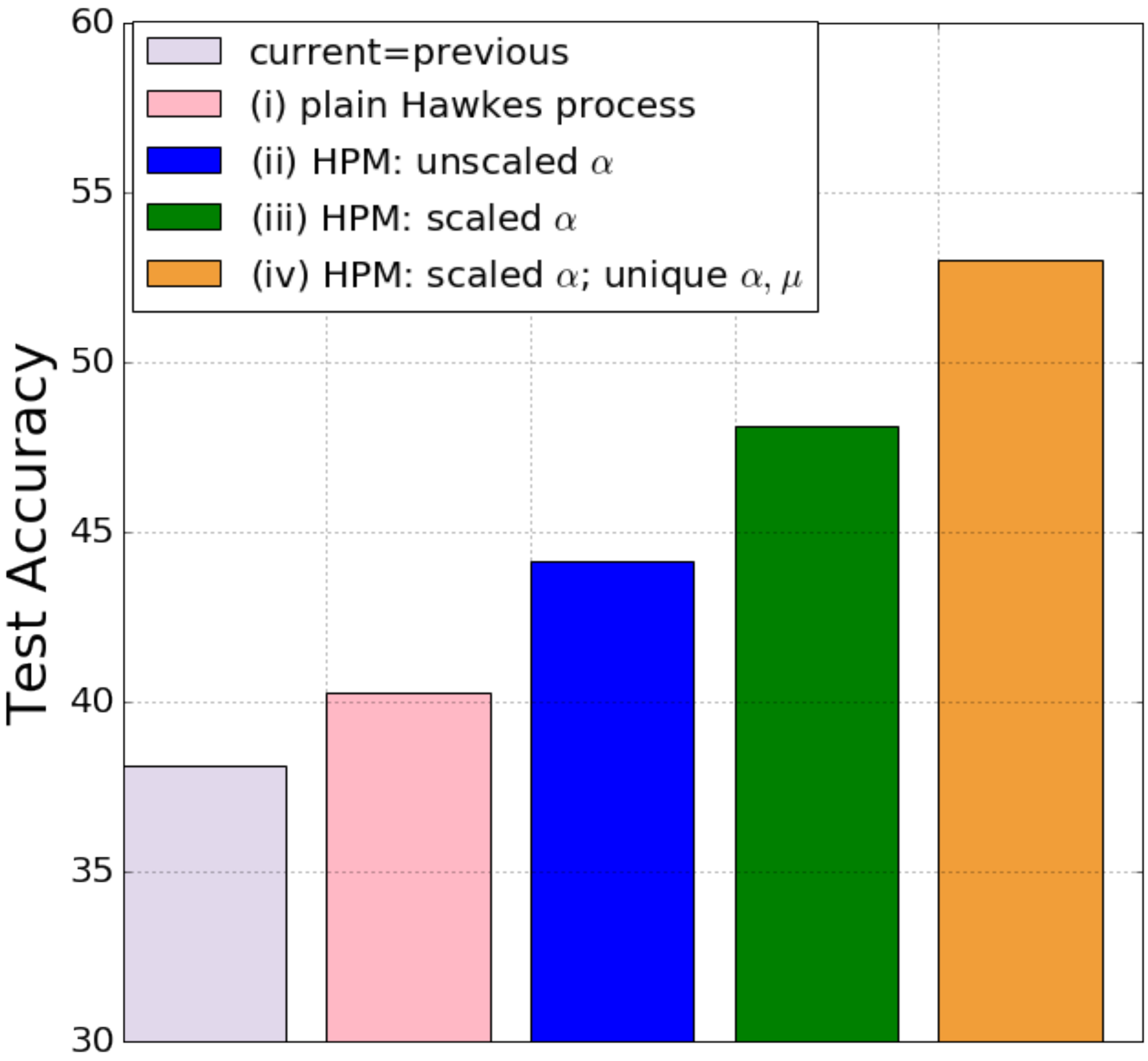
Where:
x - one-hot embedding of the sequence element,
$P(x) = W_{in}x$ - input into HPM cells,
$W_{in}, W_{out}, W_{rec}, b_{in}, b_{out}, brec$ - Normal Distributions,
Act-n = softmax of output.

Dataset



HPM Variants

$$h(t) = \mu + \alpha \sum_{t_j < t} e^{-\gamma(t - t_j)}$$



LSTM

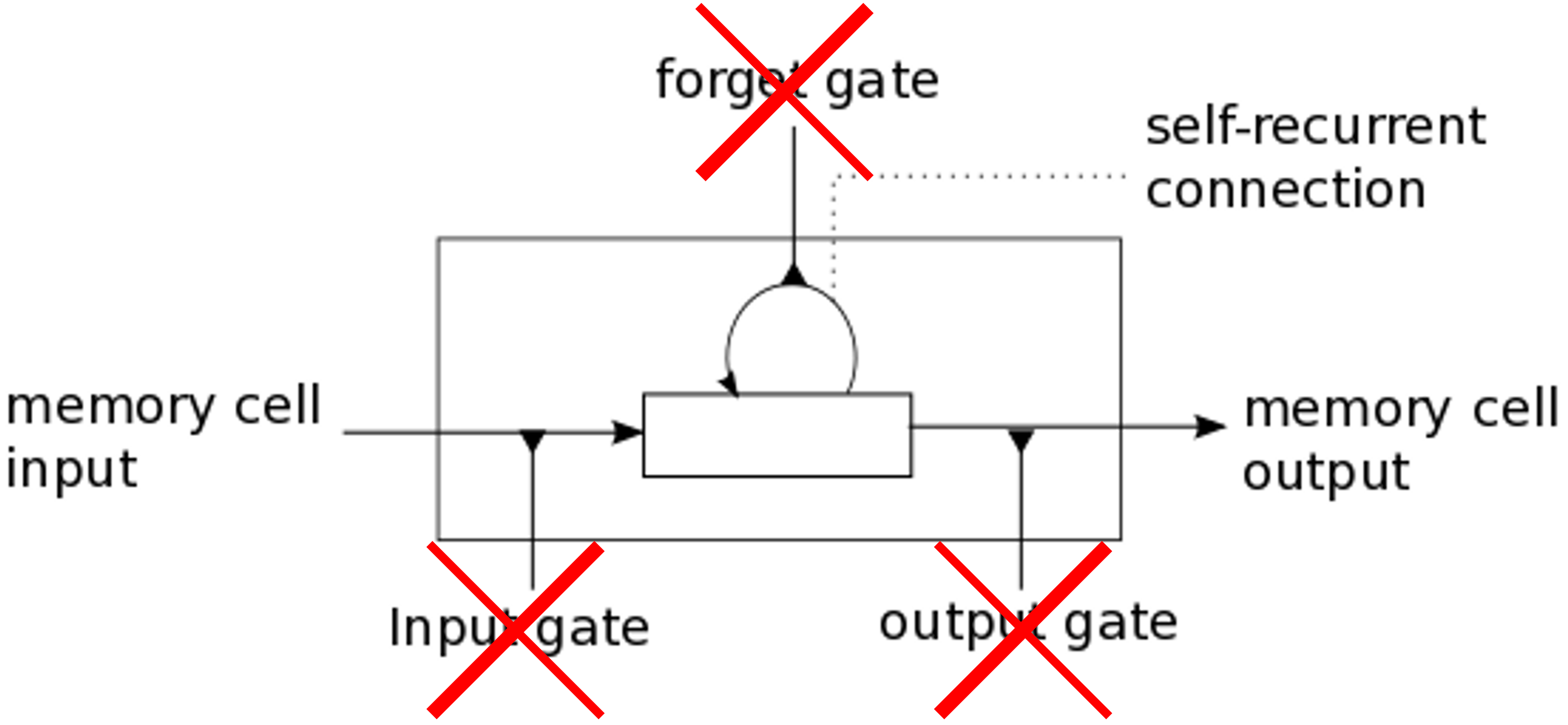
3. **Update time-scale posterior (using (10))**

$$C_{k,i} = \sum_{x_k \in \{0,1\}} P(x_k) \frac{H_{k,i}(\Delta t_k)^{x_k} Z_{k,i}(\Delta\ t_k) C_{k-1,i}}{\sum_j H_{k,j}(\Delta t_k)^{x_k} Z_{k,j}(\Delta\ t_k) C_{k-1,j}}$$

4. **Update intensity:**

$$h_{k,i} = H_{k,i}(\Delta t_k) + \alpha \gamma_i P(x_k)$$

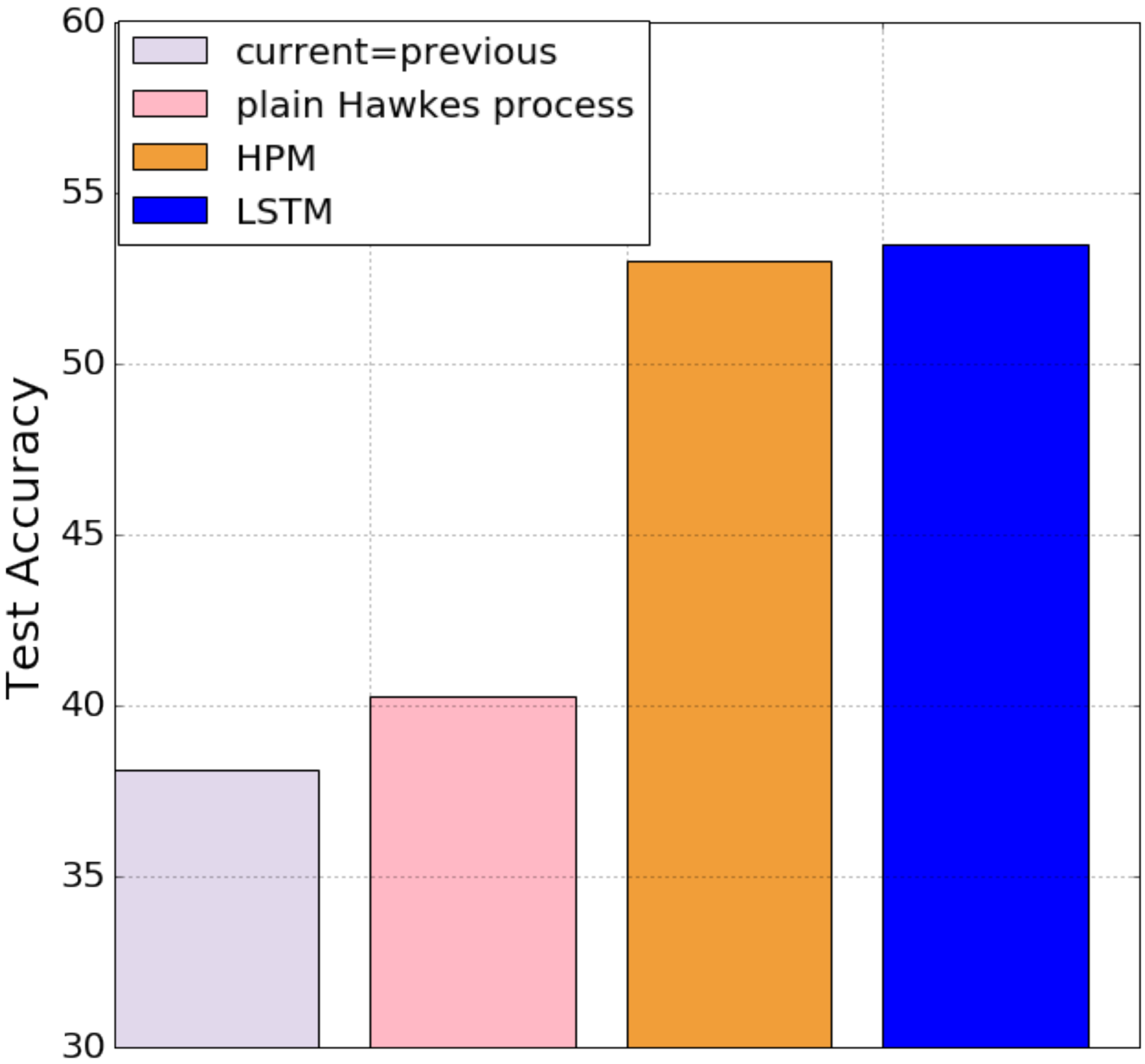
HPM:



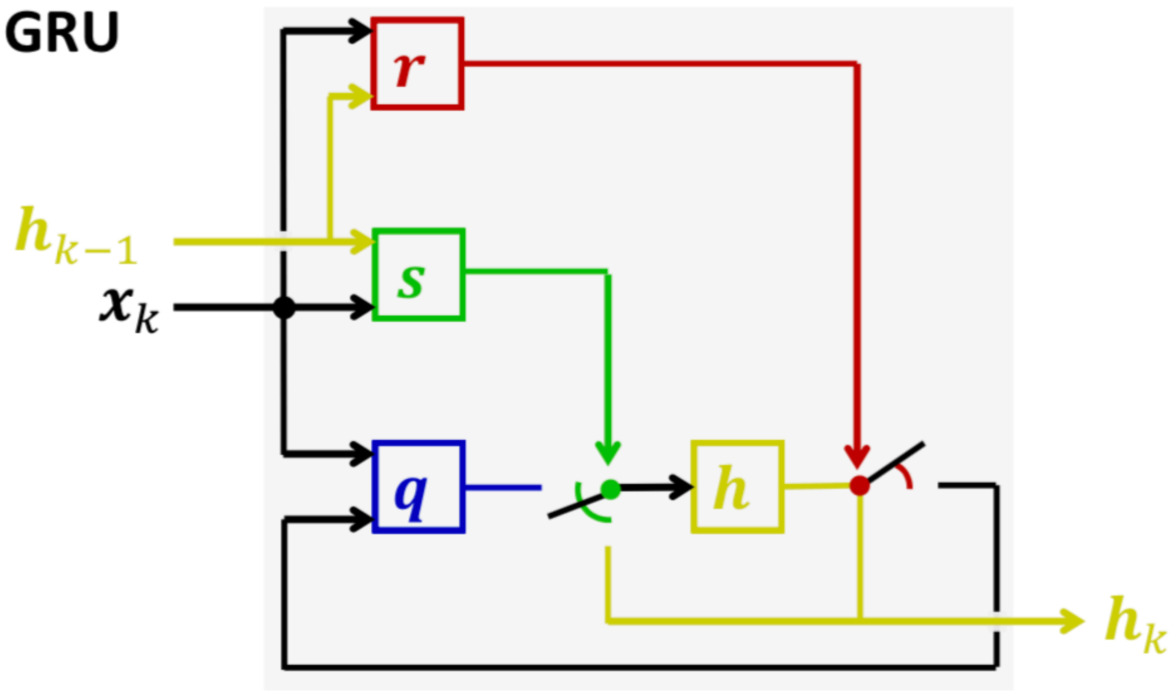
HPM vs. LSTM

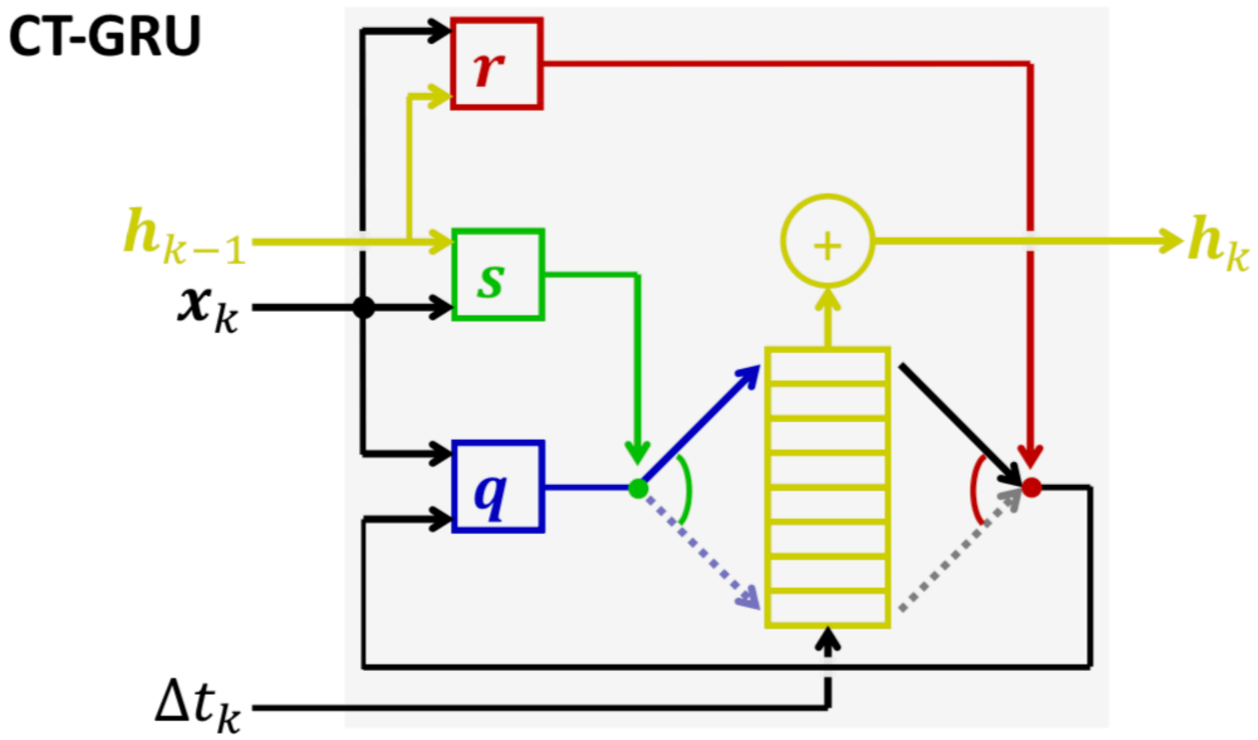
- For LSTM – time information is just another input.
- For HPM – time information is part of its operating memory.

HPM vs. LSTM


current=previous
plain Hawkes process
HPM
LSTM
Test Accuracy
60
55
50
45
40
35
30

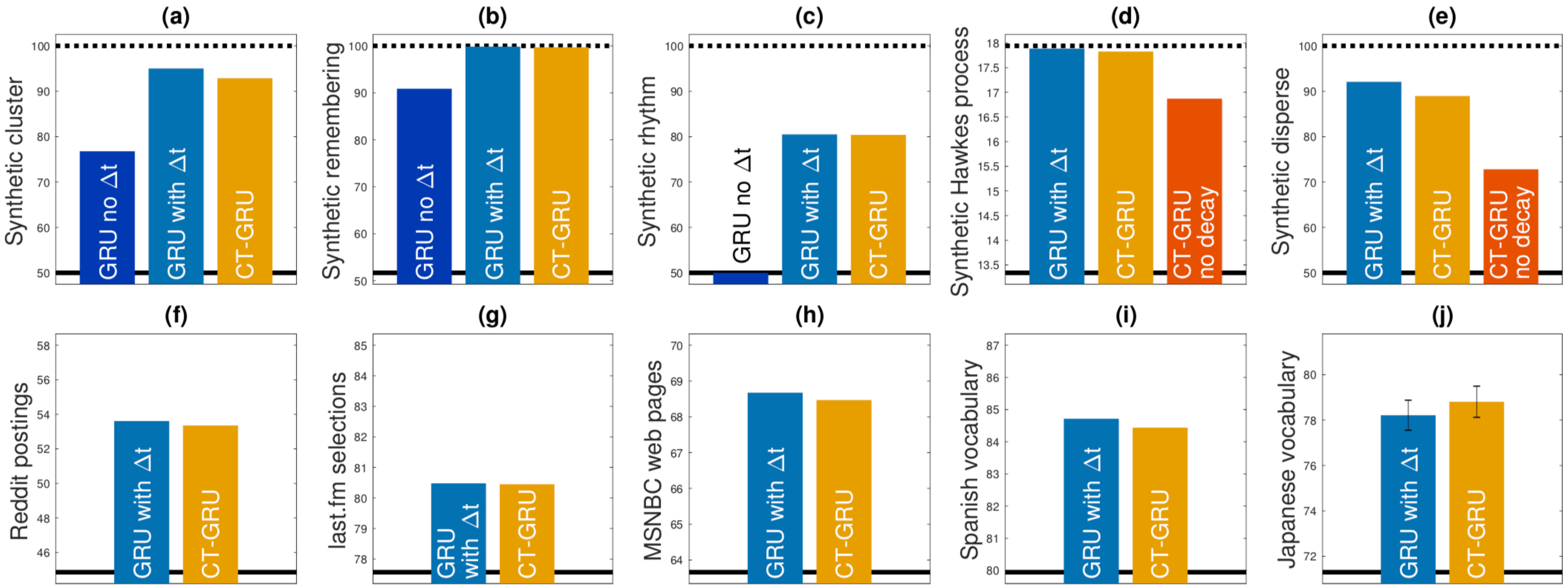
Continuous Time – GRU (CT-GRU)





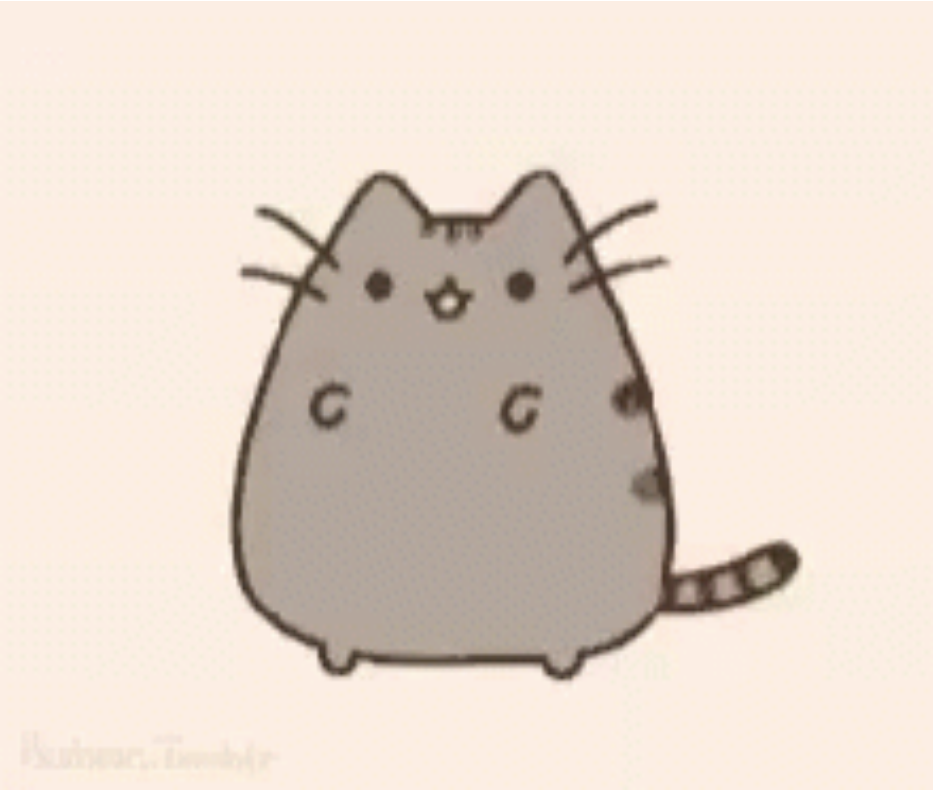
- Same decay mechanism as HPM.
- Same multiscale inference, but no longer Bayesian.

CT-GRU (explicit time) vs. GRU (implicit time)



What could be happening?

1) GRU/LSTM are so robust that the cells can always <u>implicitly</u> learns how to work with time information. Whereas, HPM just learns the same information <u>explicitly</u>.
2) We are not giving tasks where time information is complex enough.



Fin