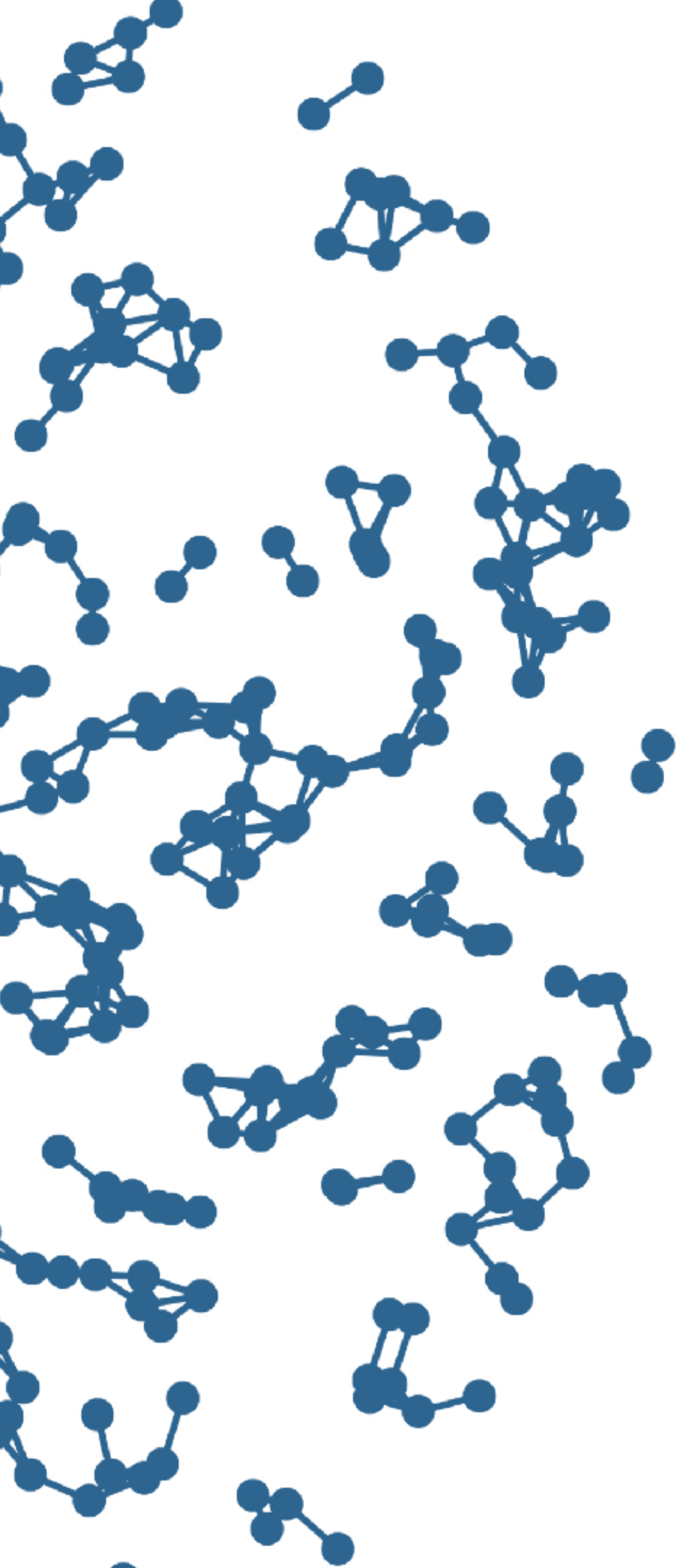


SPATIO-TEMPORAL EVENT MODELING

Lecture Series

COLLABORATIVE INTELLIGENCE

Gerrit Großmann

24.06.2024

github.com/gerritgr/spatio-temporal-lecture

dfki ai
Deutsches Forschungszentrum für Künstliche Intelligenz
German Research Center for Artificial Intelligence

Space and Time

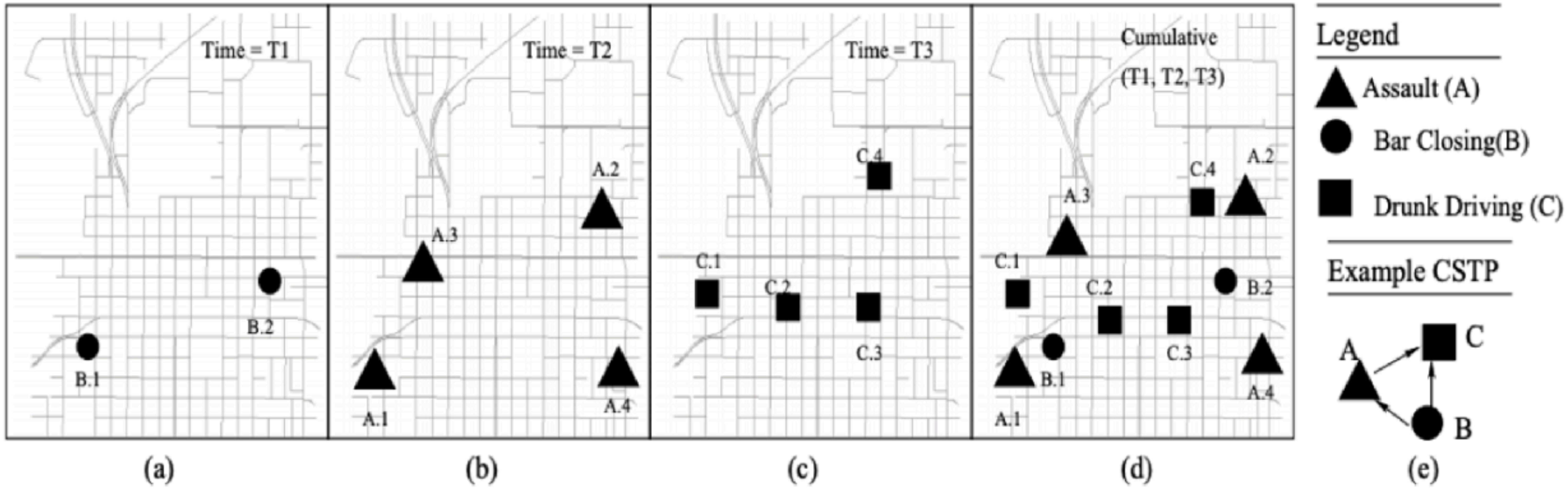
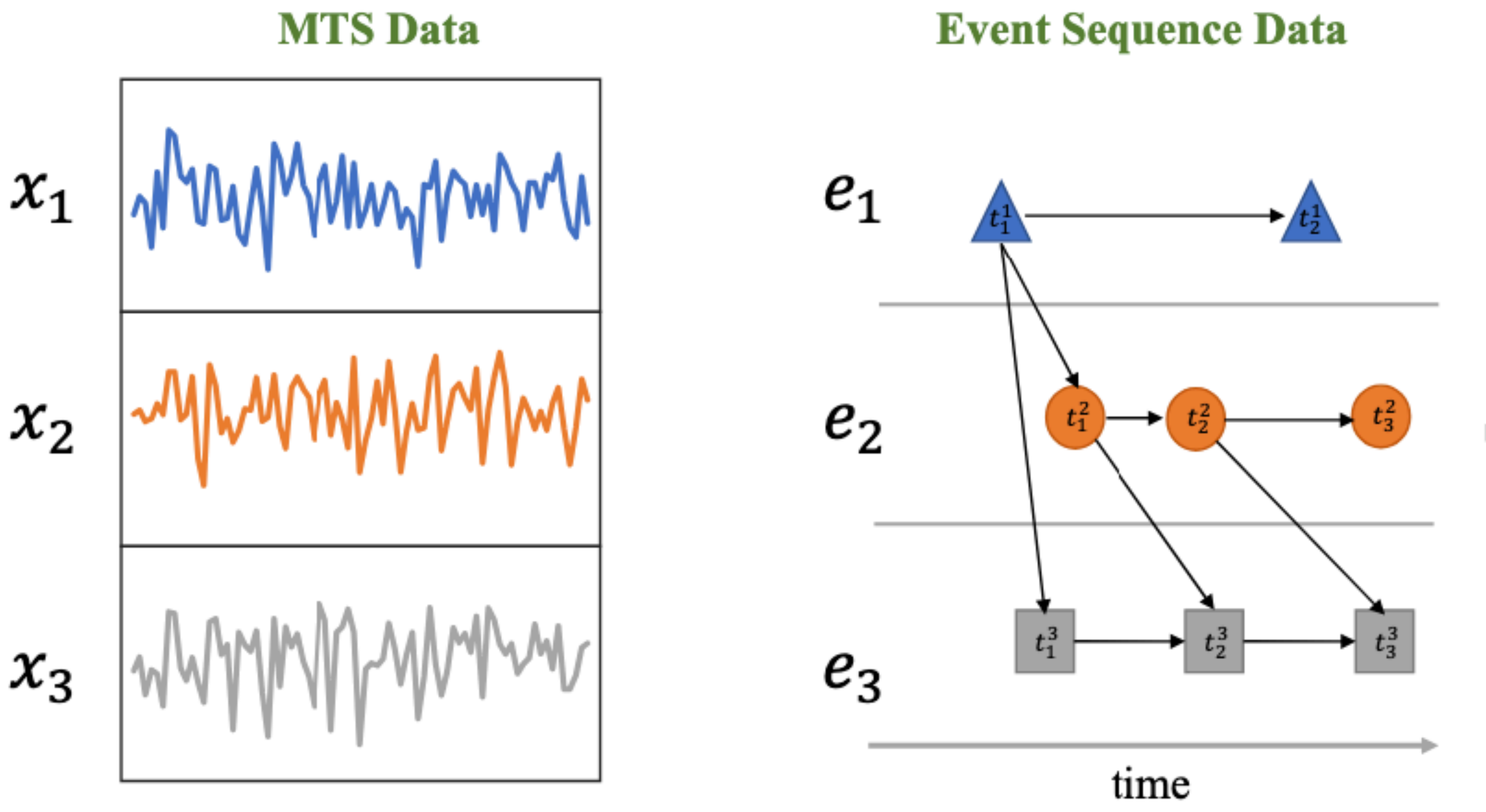


Fig. 2. Illustrative ST Crime Dataset and Cascading ST Pattern. (Best in color)

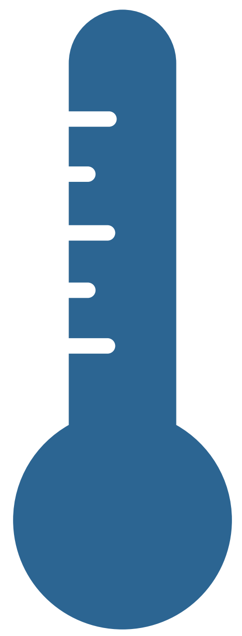
Spatiotemporal Data Mining: A Survey (Sharma et al.)

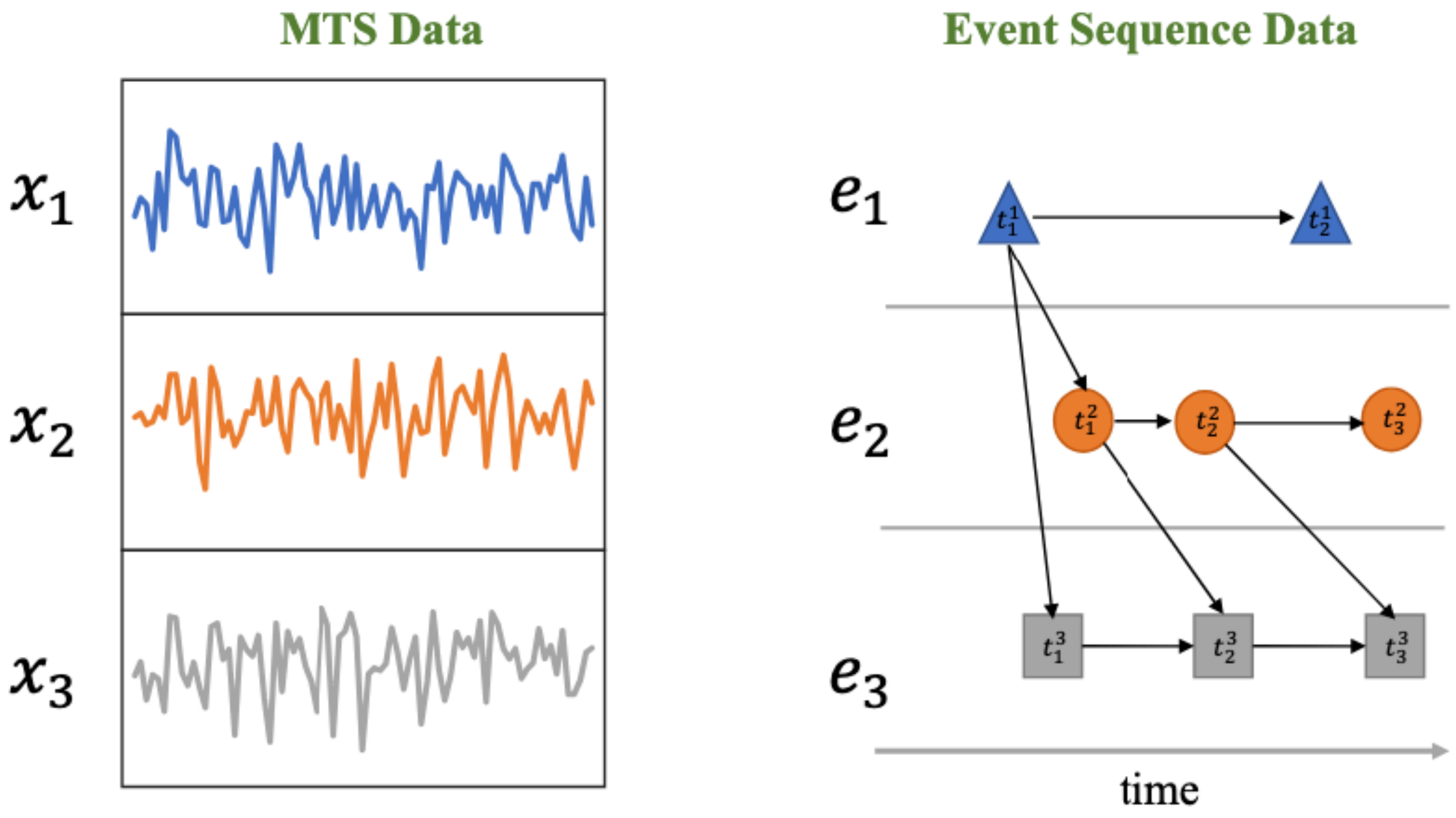
Time-Series vs Event Data



Causal Discovery from Temporal Data: An Overview and New Perspectives (Gong et al.)

Time-Series vs Event Data



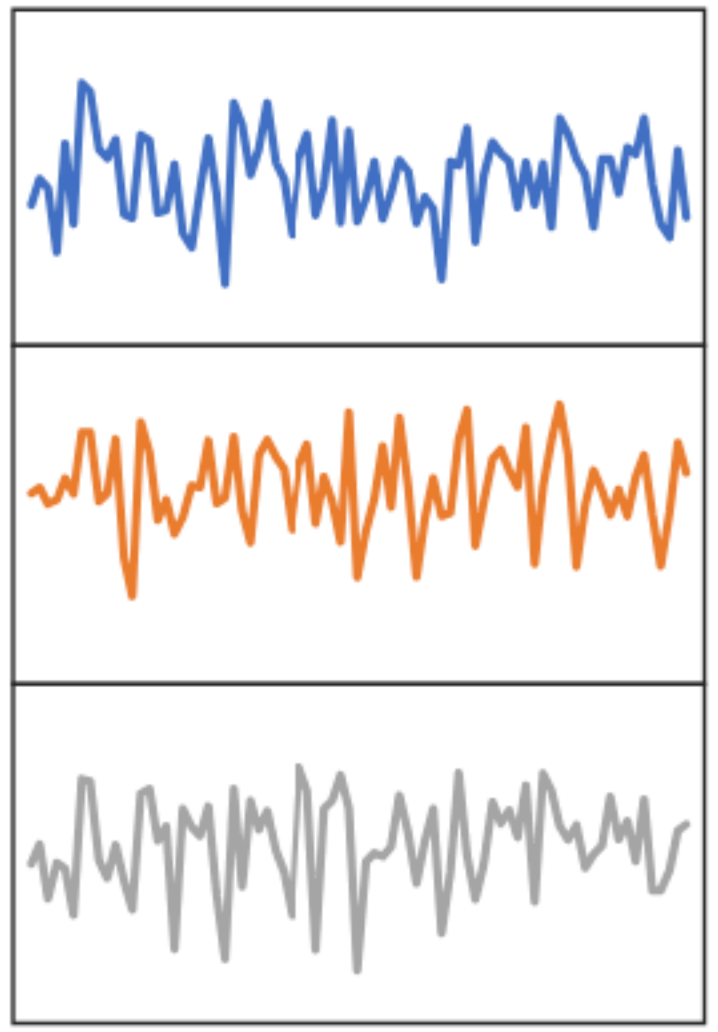


Temperature can be **continuously** monitored at various weather stations.

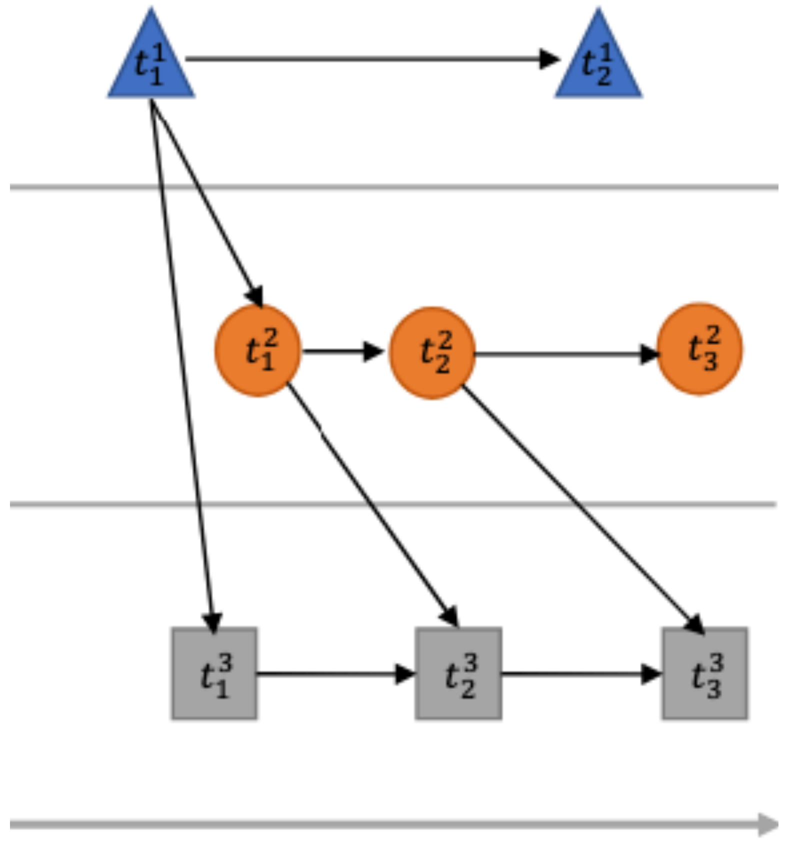
Lightning strike are **events** with specific times and locations.

Causal Discovery from Temporal Data: An Overview and New Perspectives (Gong et al.)

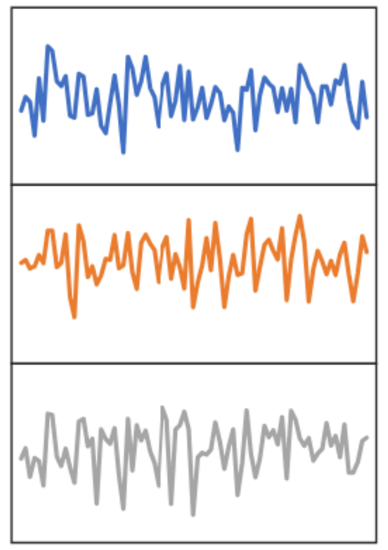
Time-Series vs Event Data

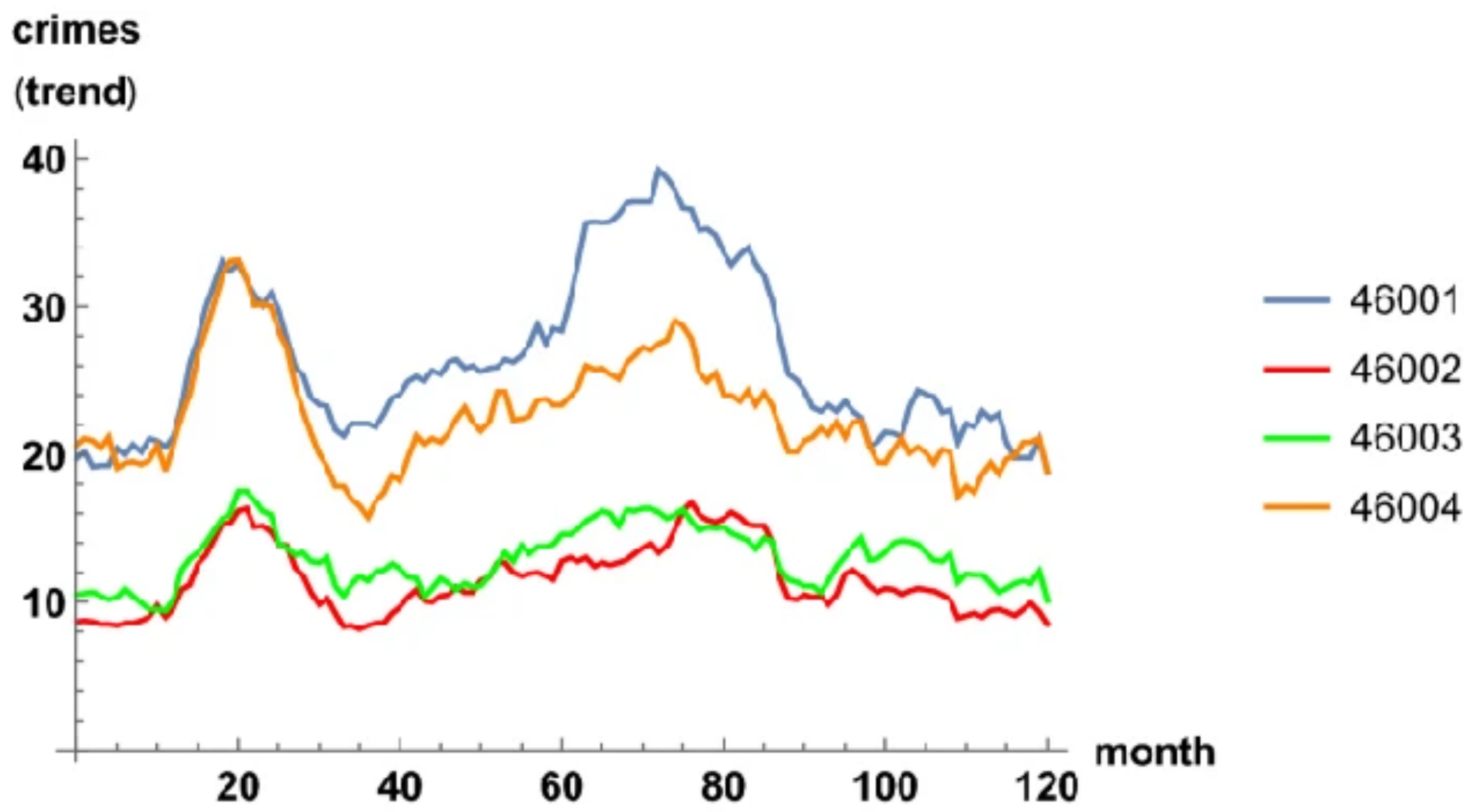


vehicle counts traffic density		accidents traffic jams
energy consumption noise		public gathering infrastructure failure
air quality soil moisture		forest fire oil spills
stock prices trading volume		large trades market crashes

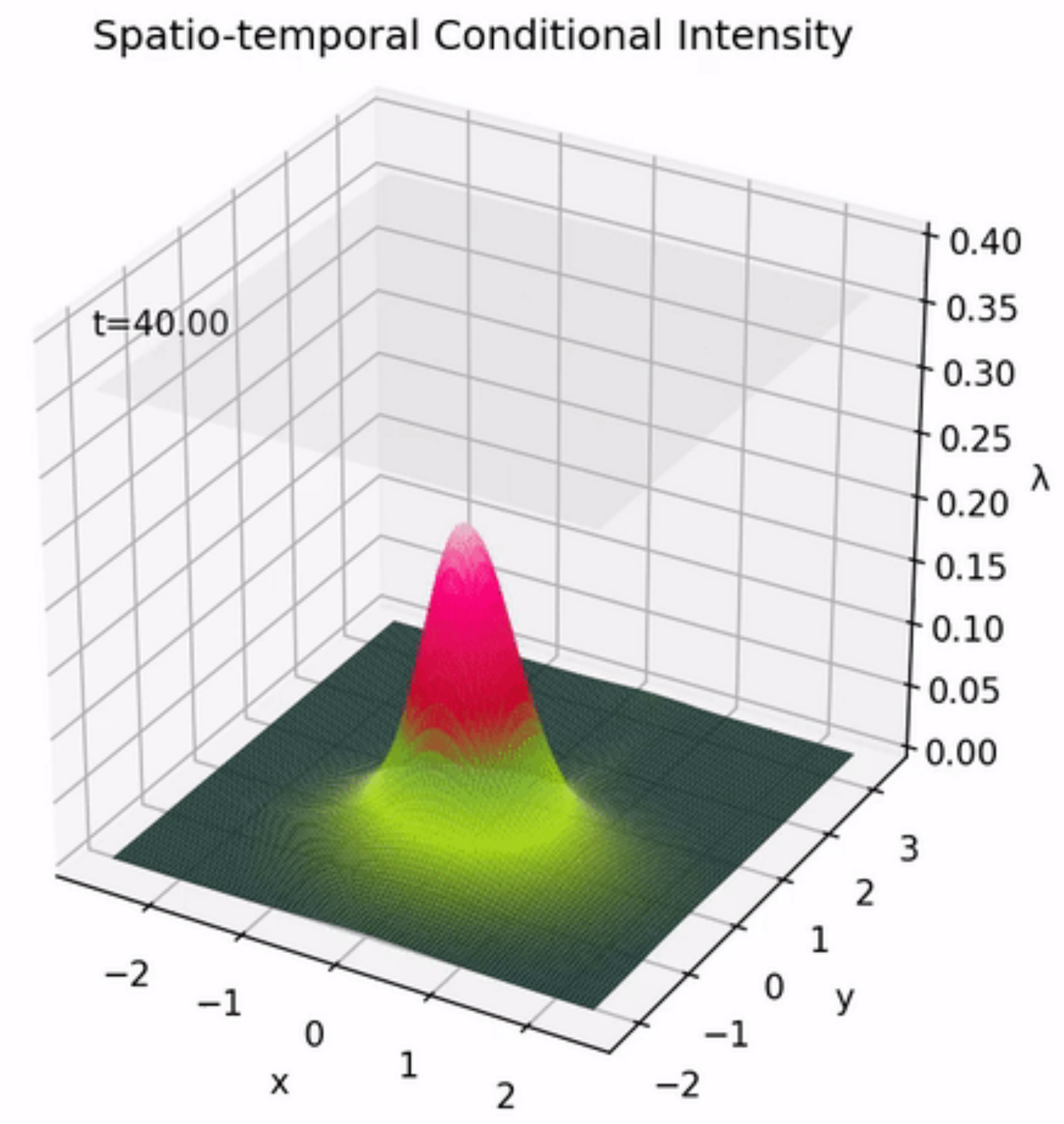


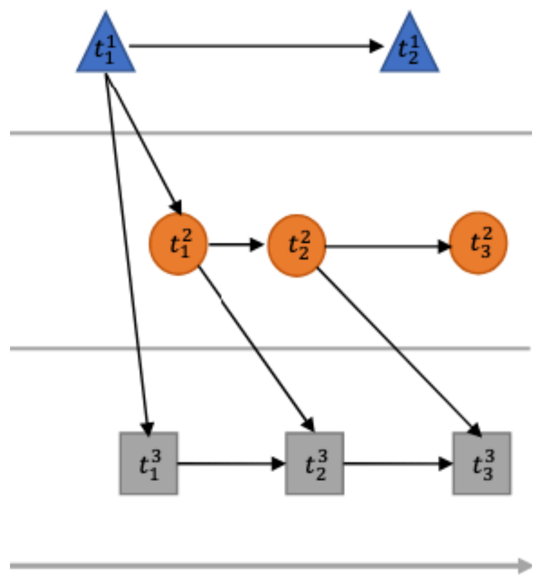
Models





Criminal incidents for different regions in Valencia





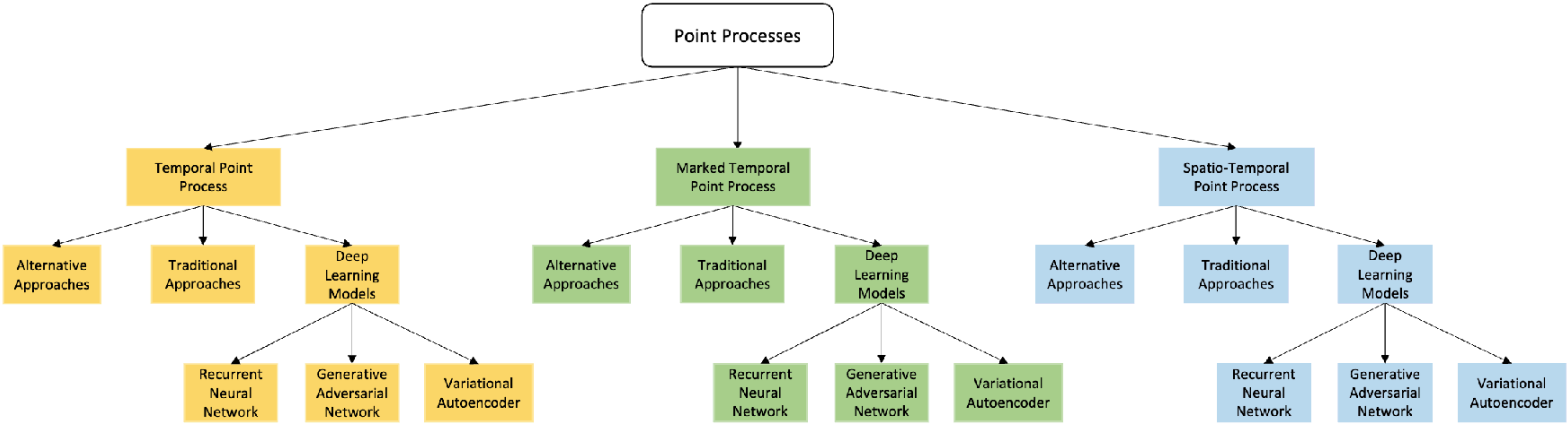
(Stochastic/Partial) **Differential Equations**
Gaussian Processes

(Marked) (Spatio-) **Temporal Point Processes**

Causal Discovery from Temporal Data: An Overview and New Perspectives (Gong et al.)
Spatio-temporal stochastic differential equations for crime incidence modeling (Calatayud et al.)

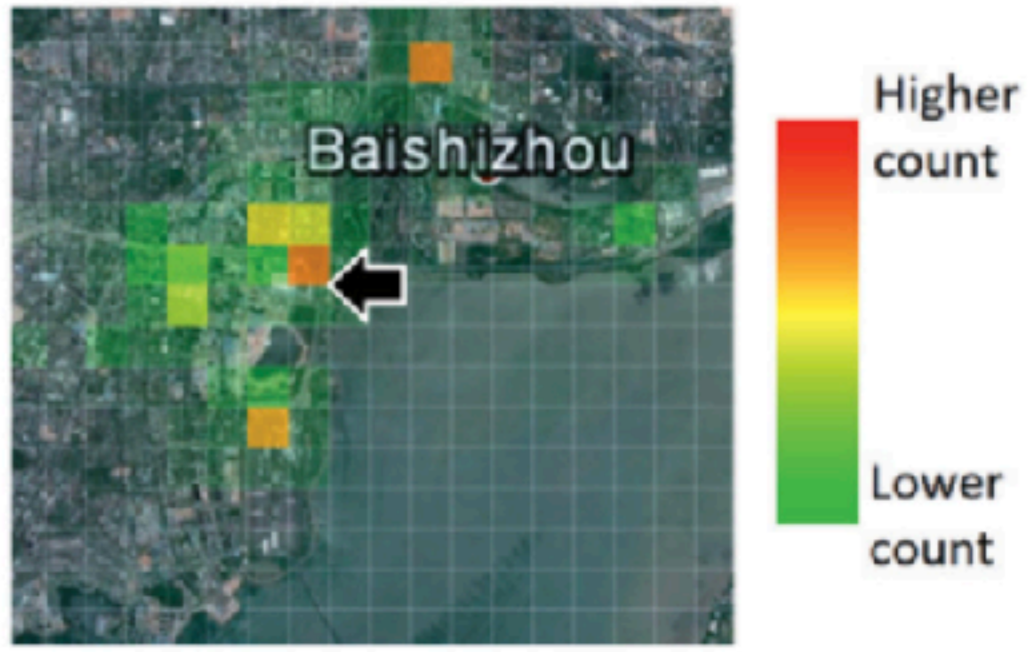
https://github.com/Rose-STL-Lab/DeepSTPP/tree/master

Predictions on Graphs

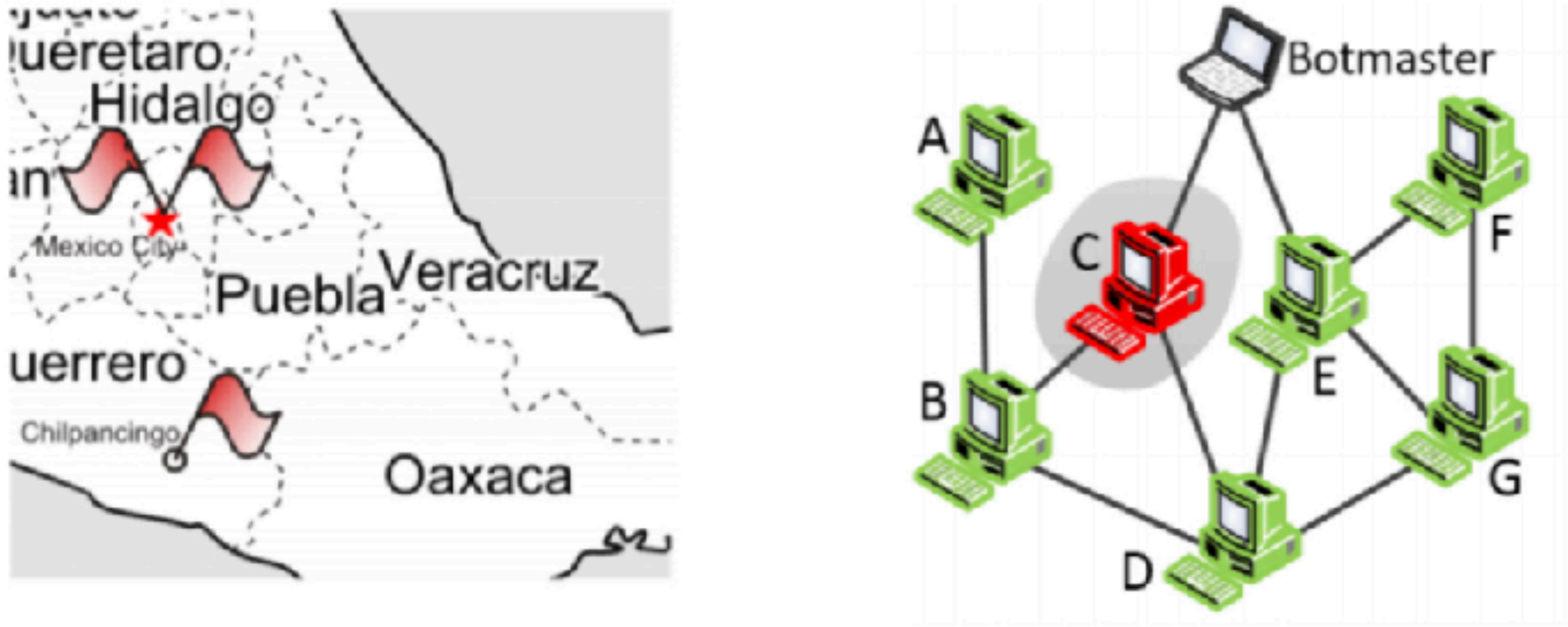


Modeling Events and Interactions through Temporal Processes - A Survey (Liguori et al.)

Notions of Space and Time

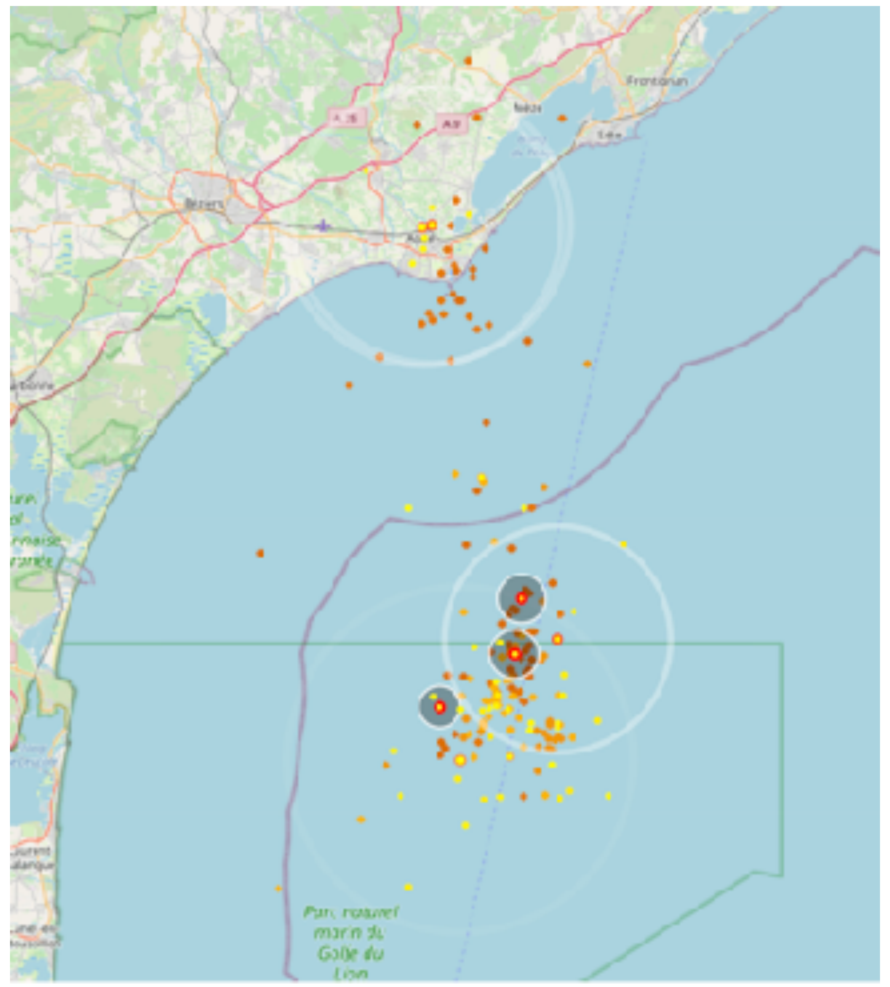


(a) Raster-based event prediction

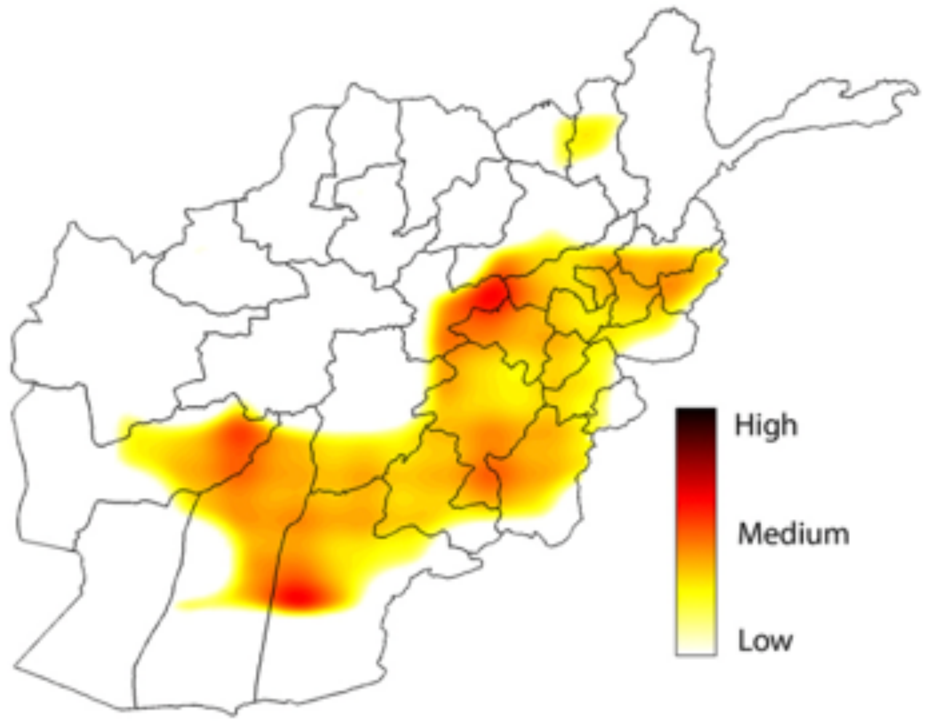


(b) Point-based event prediction in Euclidean and non-Euclidean spaces

What makes spatio-temporal data special?



lightningmaps.org



Point Process Modeling of the Afghan War Diary (Andrew Zammit-Mangion et al.)

Not i.i.d.

Arrow of time (temporal priority principle)

"A cause must precede its effect"

What makes spatio-temporal data special?

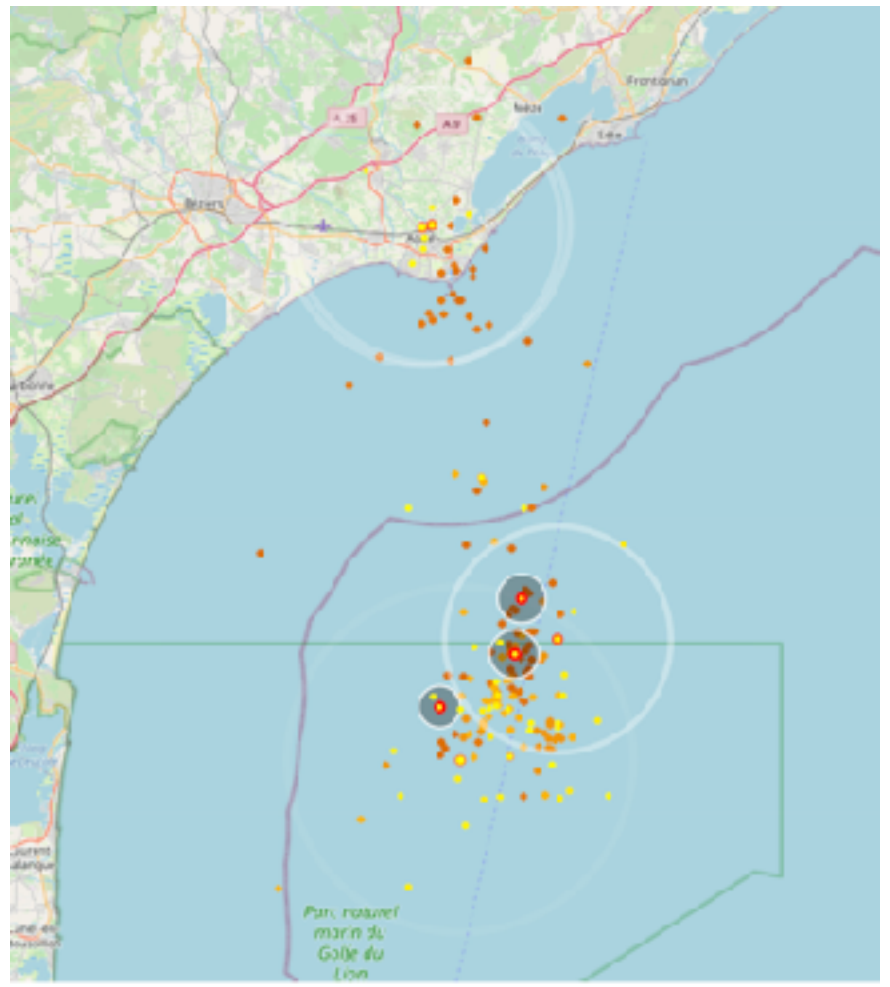
Arrow of time (temporal priority principle)

Caution: Depends on abstraction

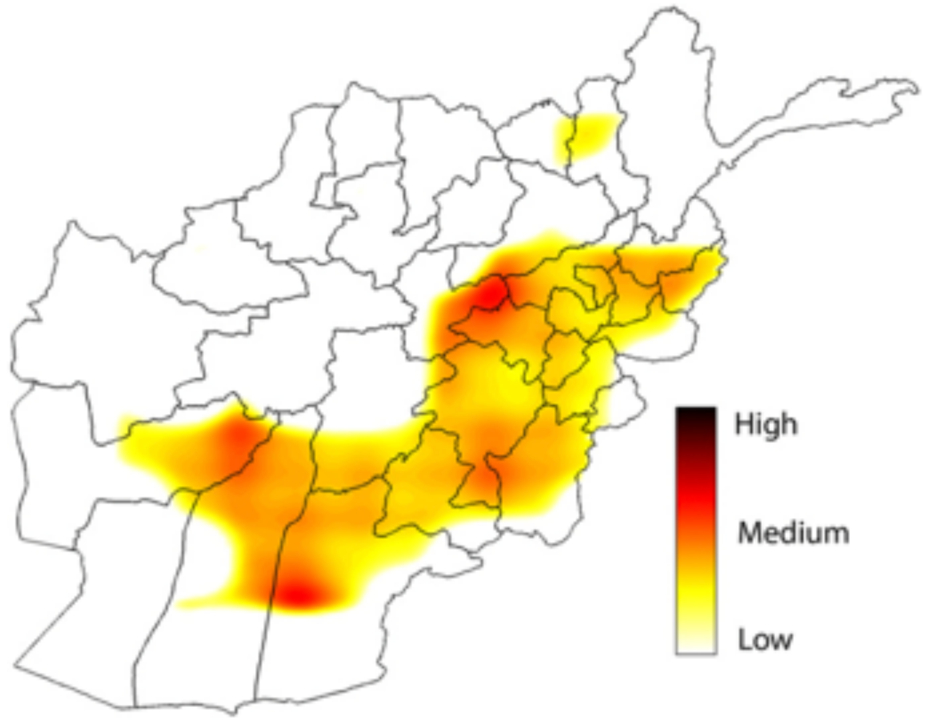
Event A: **rooster crows** precedes (causes?) Event B: **sun rises**

Fix: Consider morning light (caused by the sun rising) as an additional event.

What makes spatio-temporal data special?



lightningmaps.org



Point Process Modeling of the Afghan War Diary (Andrew Zammit-Mangion et al.)

Clustering: Positive correlation in space
Self-Excitement: Positive correlation in time

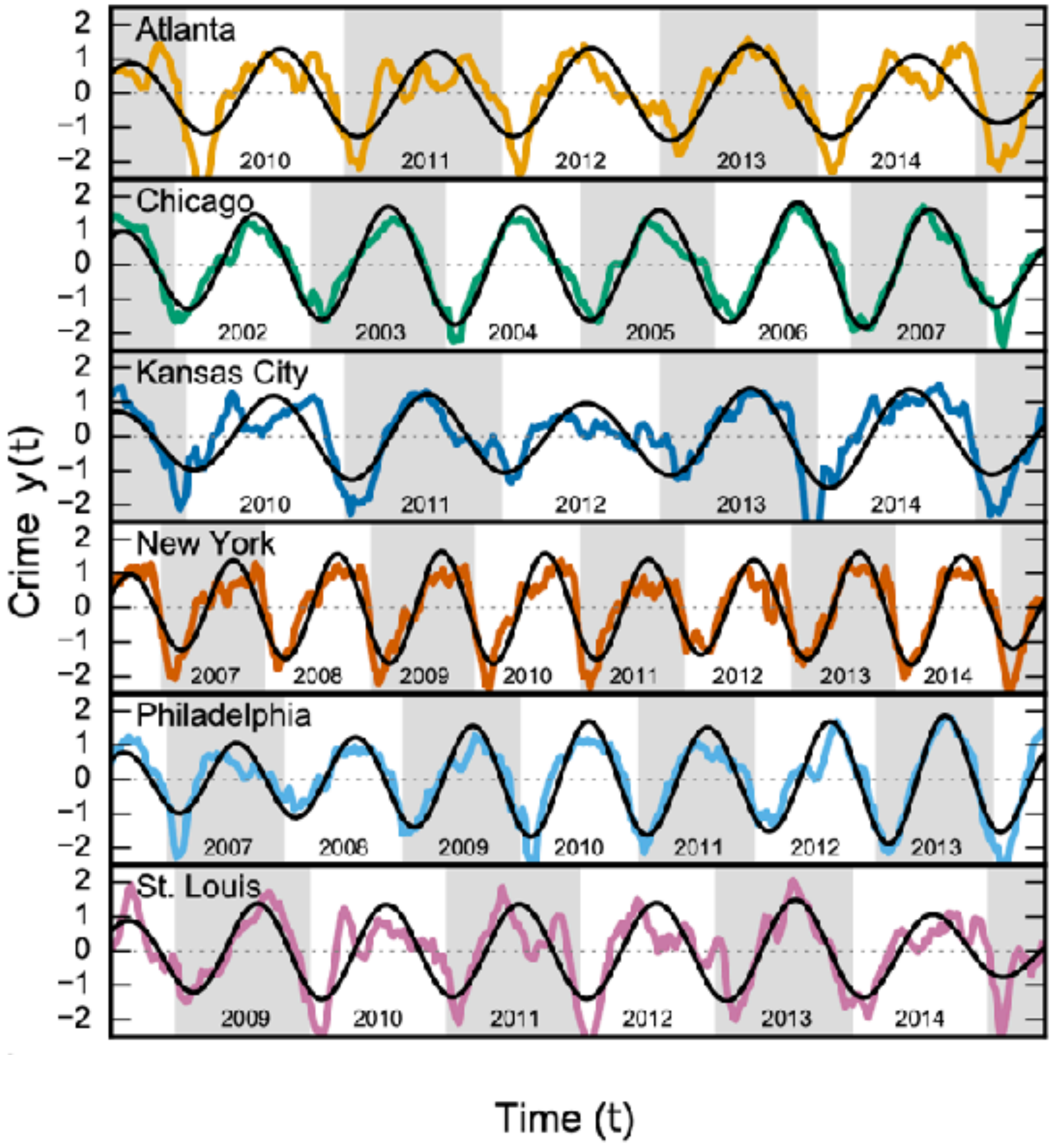
(positive feedback mechanisms, latent cause)

Self-Correction: negative correlation in time

(resource is depleted, environment adapts)

What makes spatio-temporal data special?

Periodicity and Seasonality:



Spatio-temporal variations in the urban rhythm: the travelling waves of crime (Oliveira et al.)

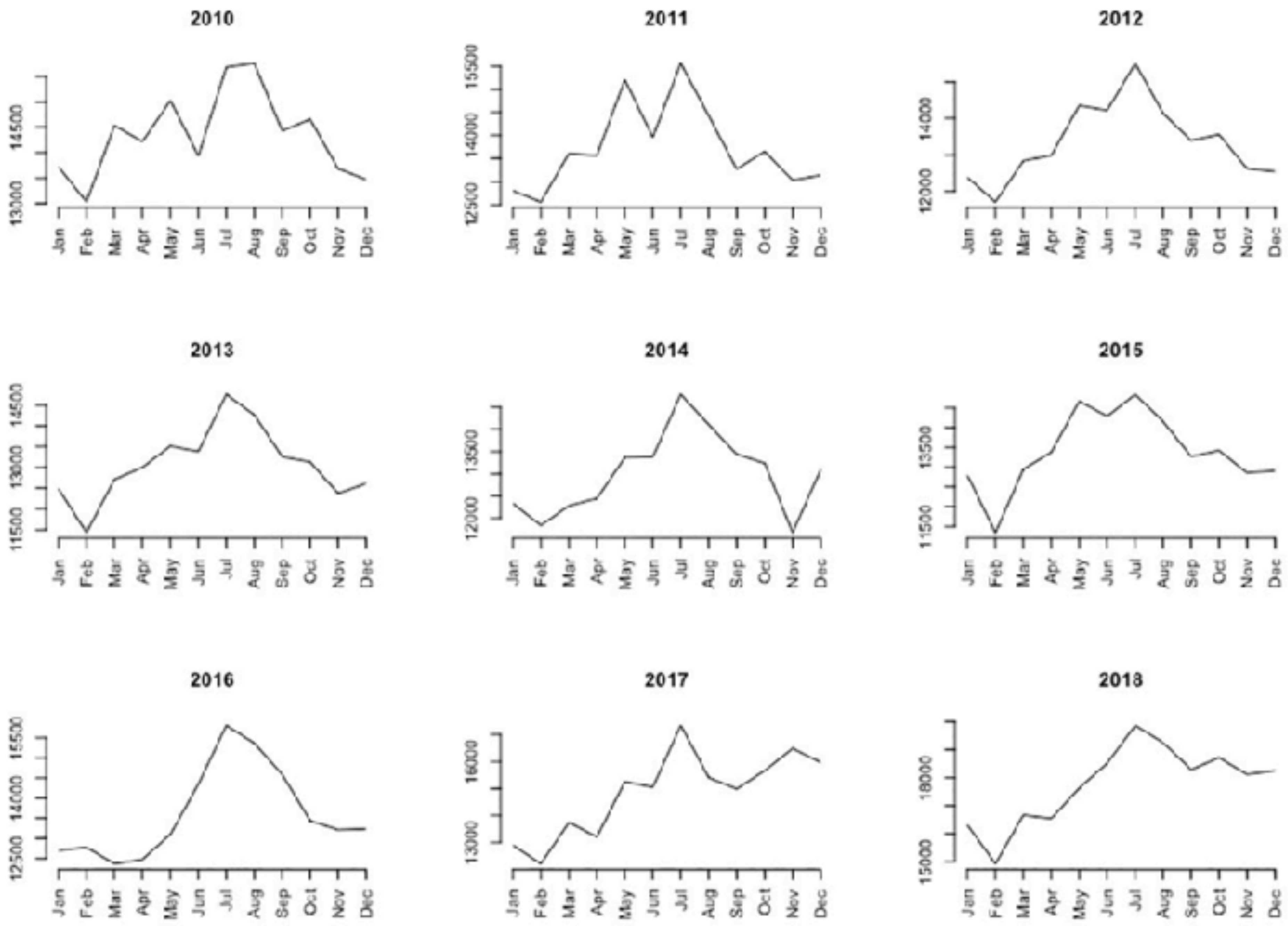
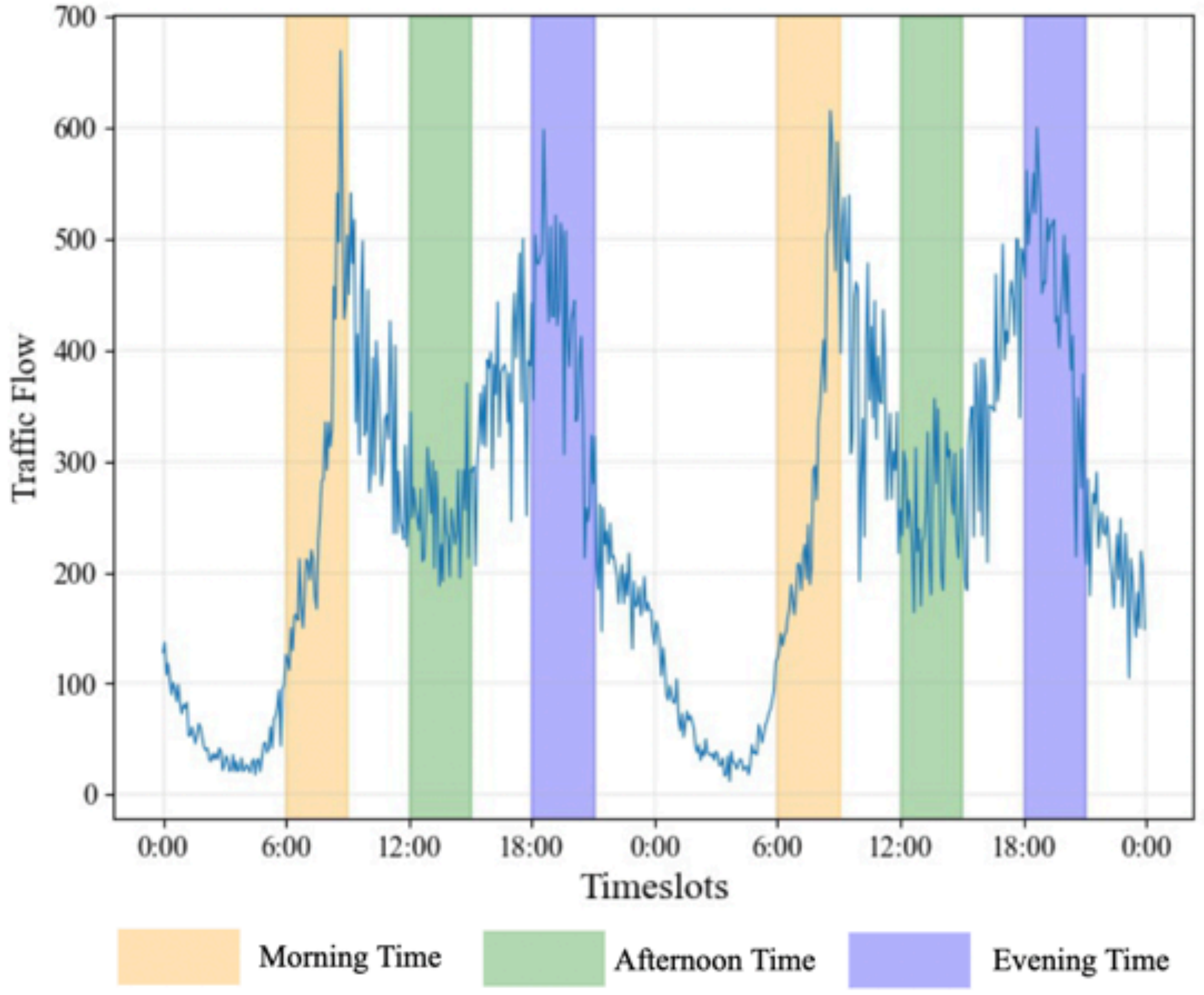


Fig 6. Seasonal trend from January to December of the monthly total of crimes in Barcelona, by year.

The effect of seasonality in predicting the level of crime. A spatial perspective (Delgado et al.)

What makes spatio-temporal data special?

Periodicity and Seasonality:



STG4Traffic: A Survey and Benchmark of Spatial-Temporal Graph Neural Networks for Traffic Prediction (Luo et al.)

Self-Exciting

Self-Correcting

Does a robbery make it more or less likely that another robbery will occur the next day?

- Signal to criminals
- Acts of retaliation

https://comic-denkblase.de/70-jahre-panzerknacker

- Vigilance
- Police presence
- Community engagement

Self-Exciting

Self-Correcting

The flashing of a firefly

Suicide

Crime

Earthquake

A goal in a soccer match

Terrorist attack

Large jump in the stock price of a company

Neuron firing

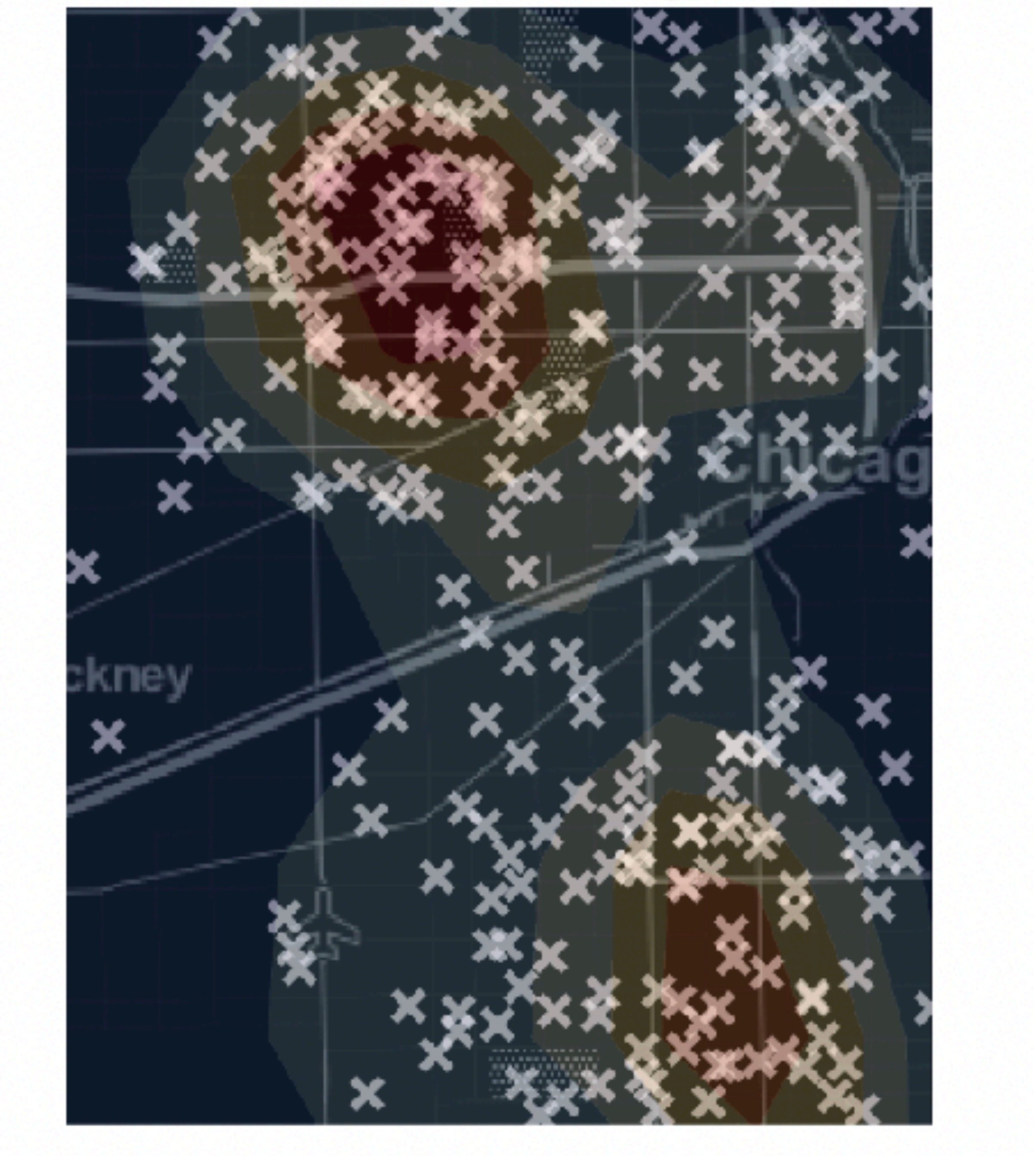
A person coughs

Rainfall

Traffic congestion

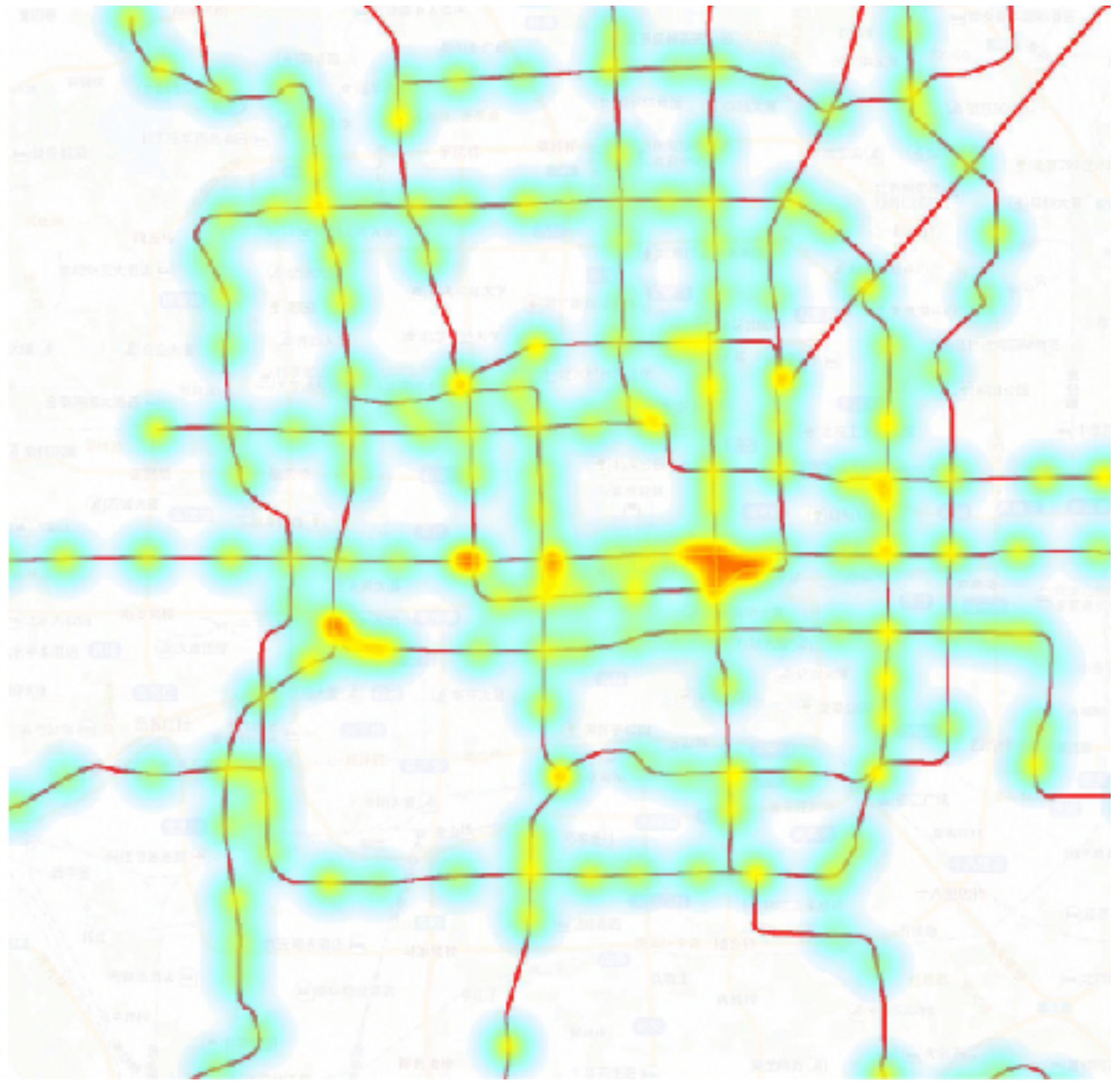
Applications

Prediction:



Crime

Deep Mixture Point Processes: Spatio-temporal Event Prediction with Rich Contextual Information (Okawa et al.)



Traffic

GSTNet: Global Spatial-Temporal Network for Traffic Flow Prediction (Fang et al.)

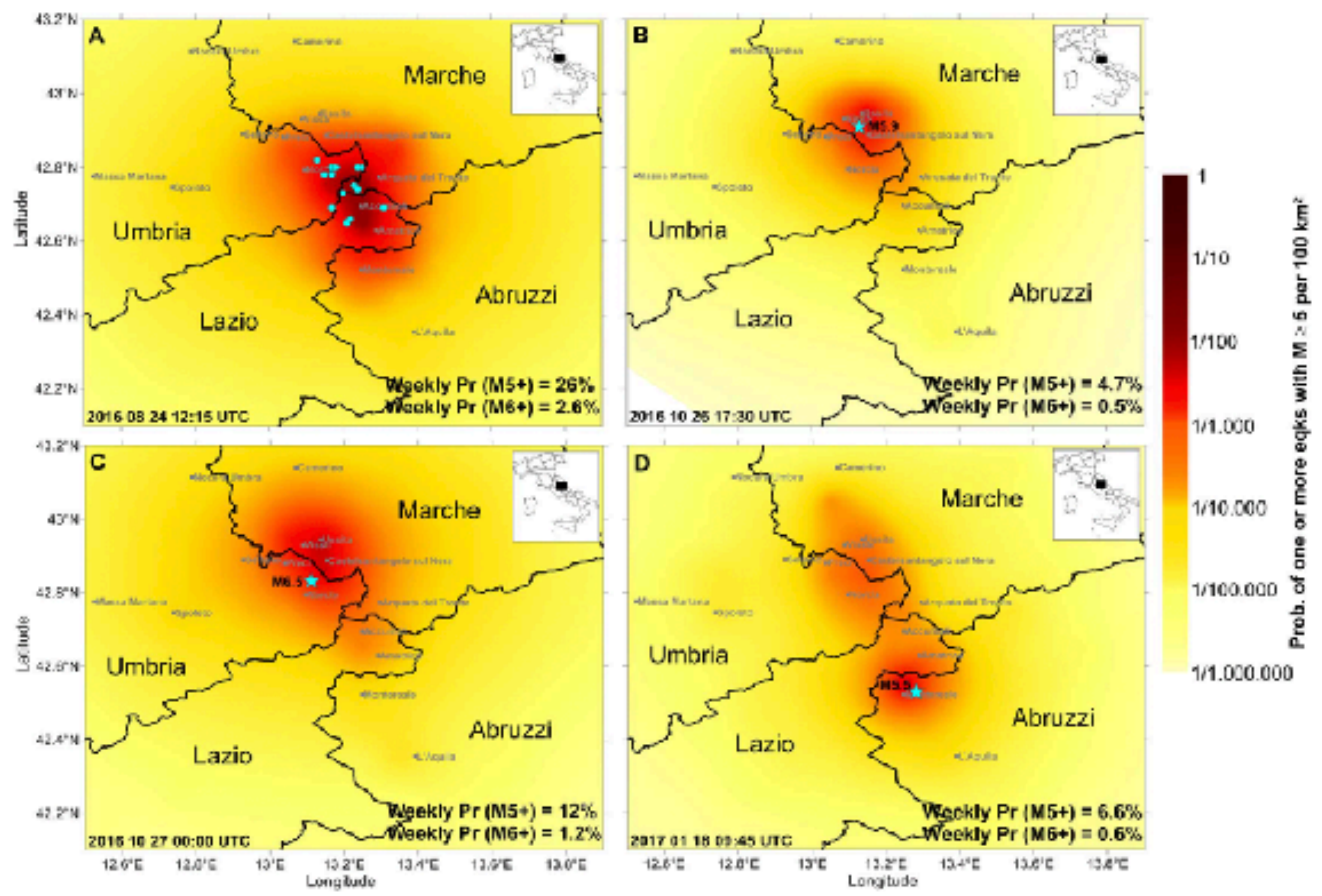


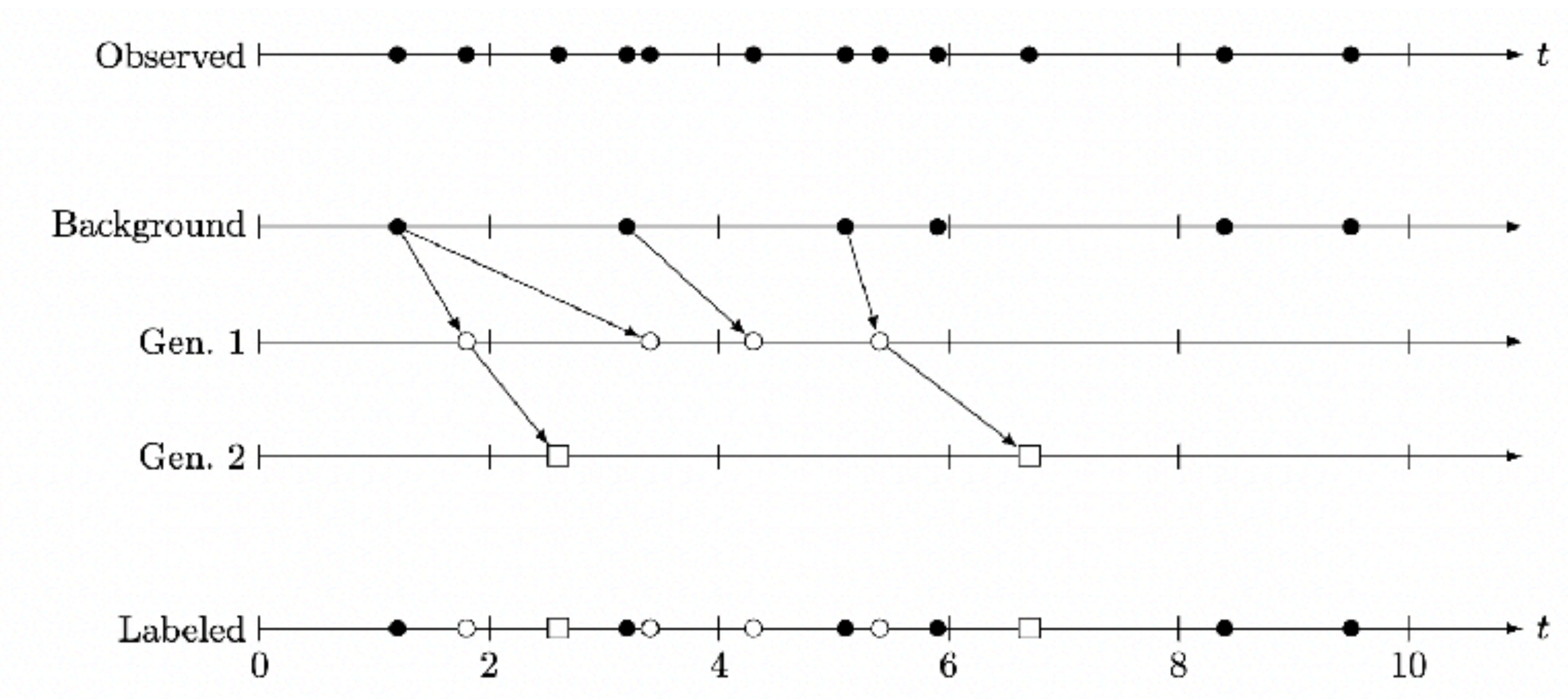
Fig. 2. Some examples of weekly forecasts (the number of the forecasts are reported on Table 1). (A) Forecast number 3, a few hours after the Amatrice earthquake and the M3.5+ earthquakes (blue-green circles) that occurred in the forecasting time window. **(B)** Forecast number 15, before the M3.9 earthquake (blue-green star) that occurred on October 26. **(C)** Forecast number 18, before the Norcia M6.5 earthquake (blue-green star) that occurred on October 30. **(D)** Forecast number 35, before the Campotosto M5.5 earthquake (blue-green star) that occurred on January 18.

Earthquakes

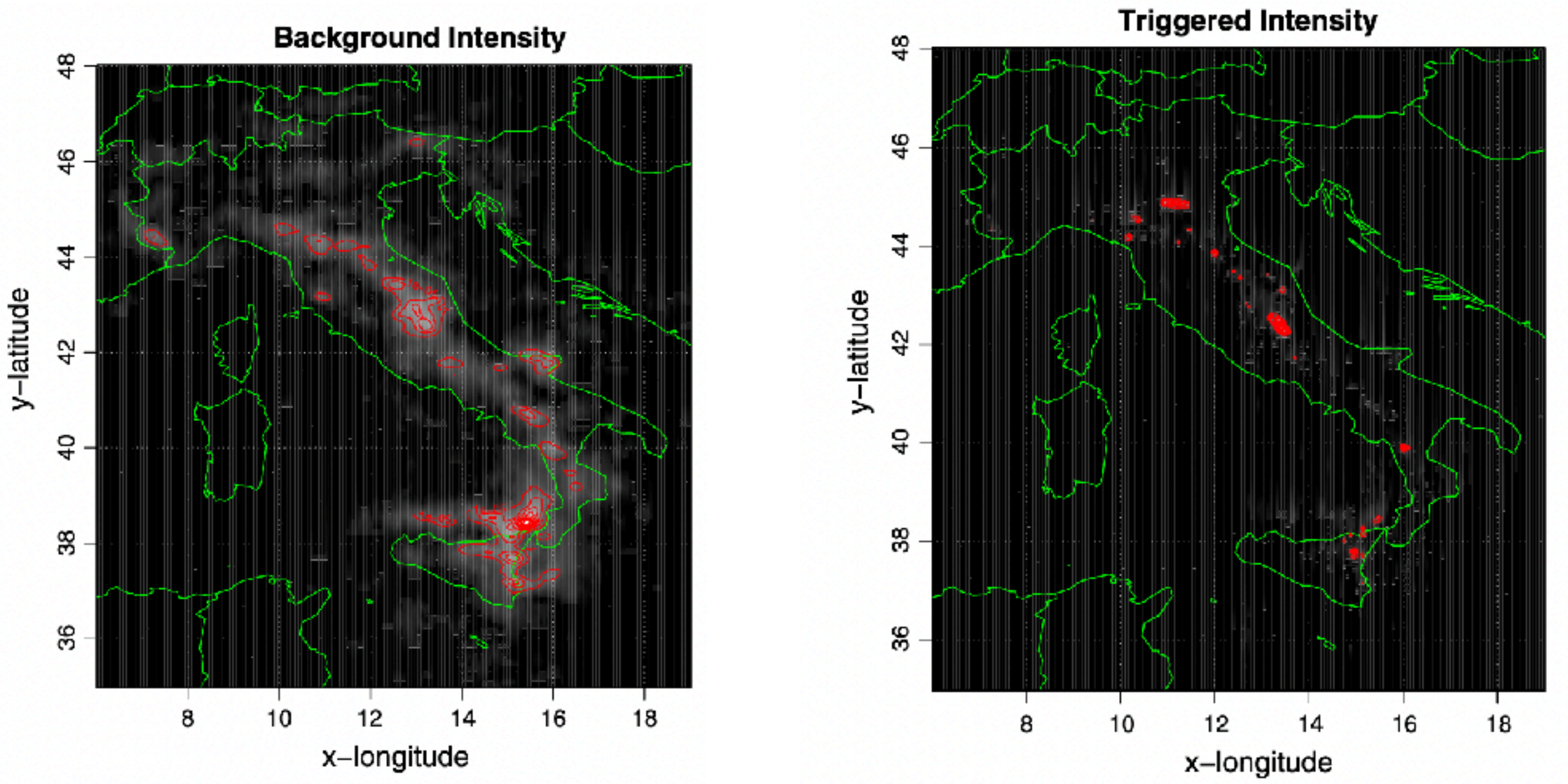
Earthquake forecasting during the complexAmatrice-Norcia seismic sequence (Marzocchi et al.)

Applications

De-clustering:



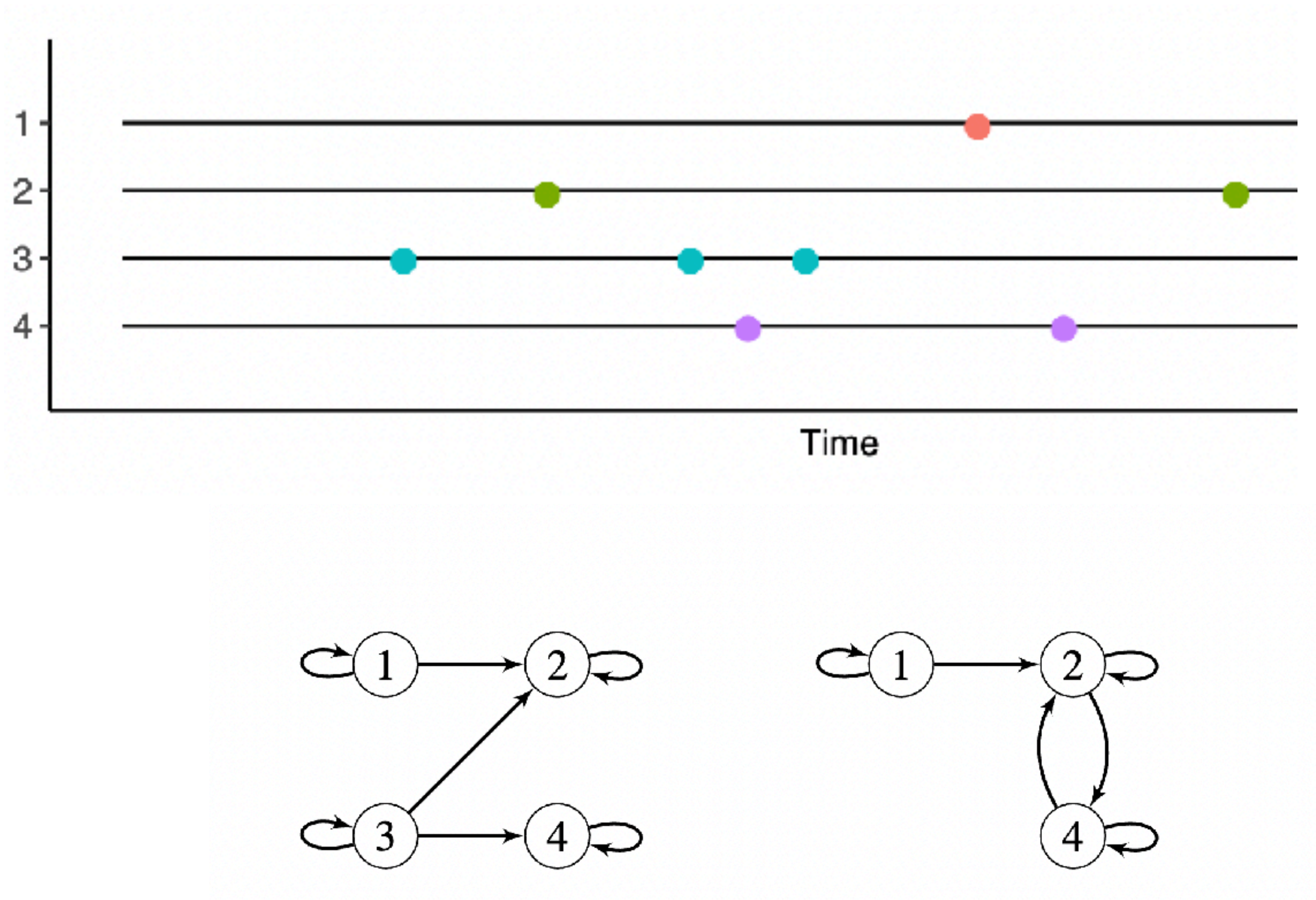
A Review of Self-Exciting Spatio-Temporal Point Processes and Their Applications (Reinhart)



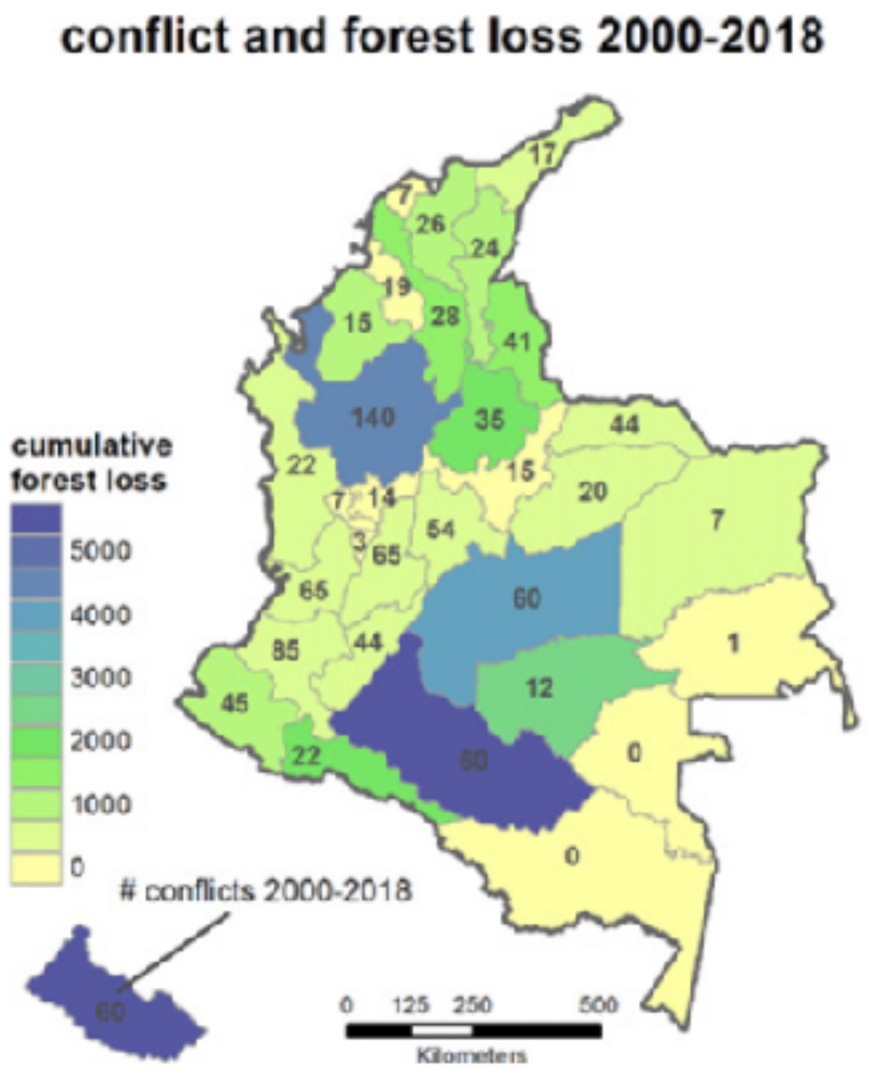
Alternated estimation in semi-parametric space-time branching-type point processes with application to seismic catalogs (Adelfio et al.)

Applications

Causal Discovery:



Causal screening in dynamical systems (Mogensen)



Toward Causal Inference for Spatio-Temporal Data: Conflict and Forest Loss in Colombia (Christiansen et al.)

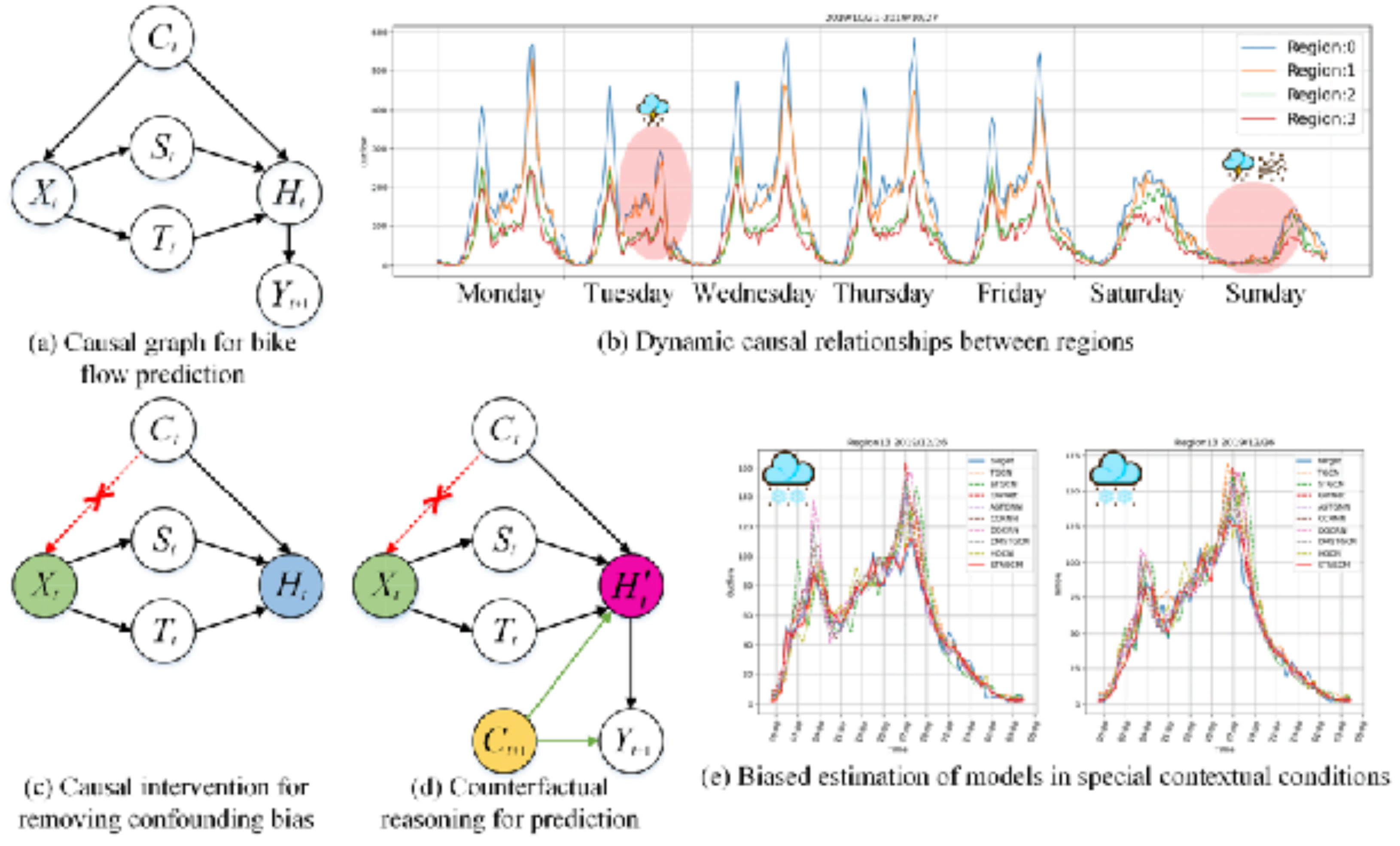
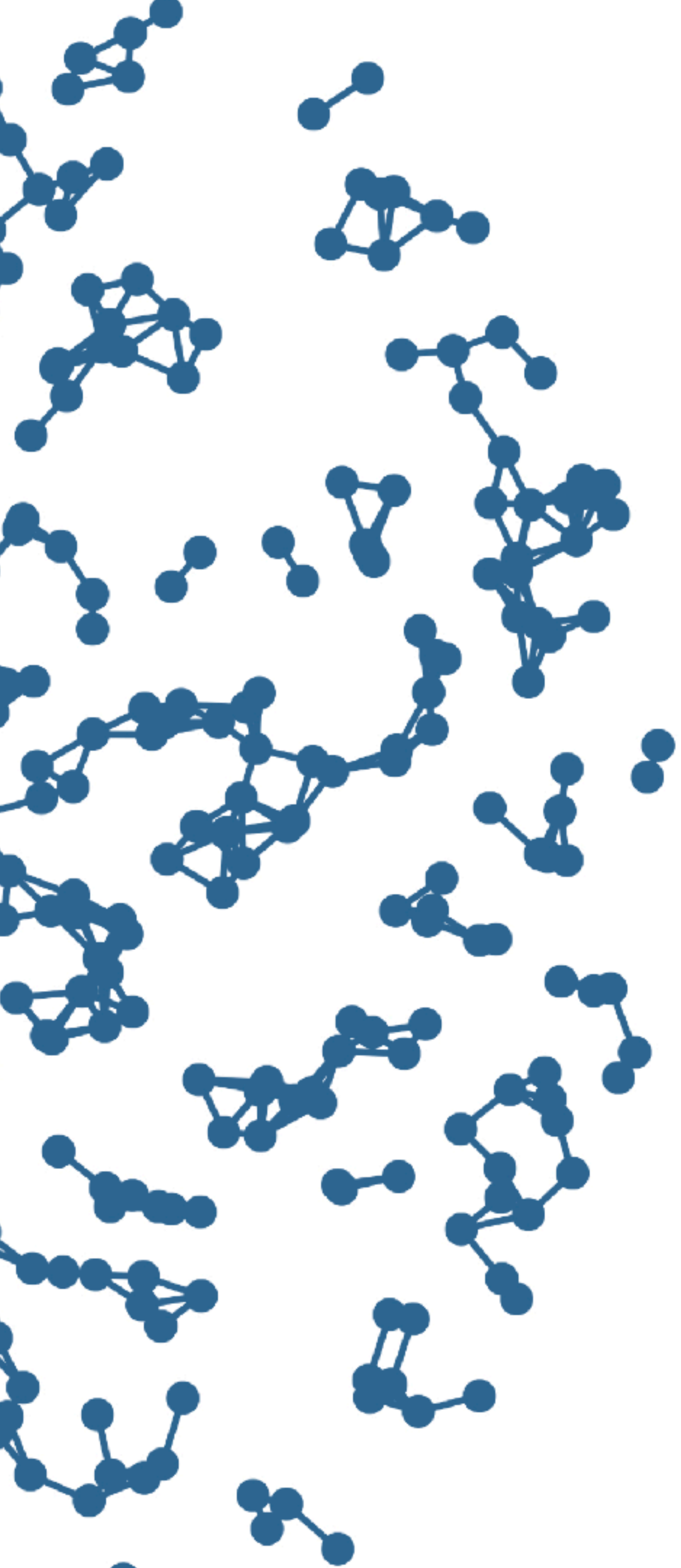


Figure 1: The change process of bike flow from the perspective of causality.

Spatio-temporal Neural Structural Causal Models for Bike Flow Prediction (Deng et al.)

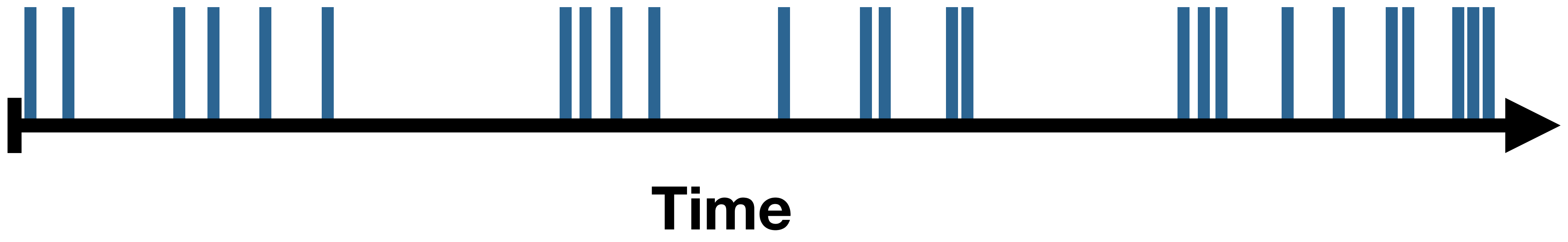


Part I

TEMPORAL POINT PROCESSES

Models of TPPs

Spike train of a single neuron



https://mark-kramer.github.io/Case-Studies-Python/08.html

Event sequence: $H = [t_0, t_1, \ldots, t_n]$ ($t_i < t_{i+1}$ and $t_i \in \mathbb{R}_{\geq 0}$).

E.g., $\mathrm{H} = [0.1, 0.5, 3.4, 4.2]$

t_i is called the event time, timestamp, or spike time

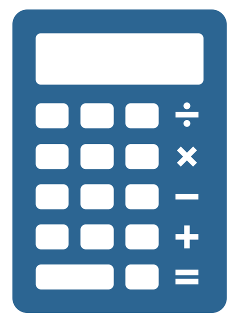
Inter-event time or waiting time: $t_{i+1} - t_i$

H_t is the sequence up to a time horizon t.

E.g., $H_4 = [0.1,\ 0.5,\ 3.4]$

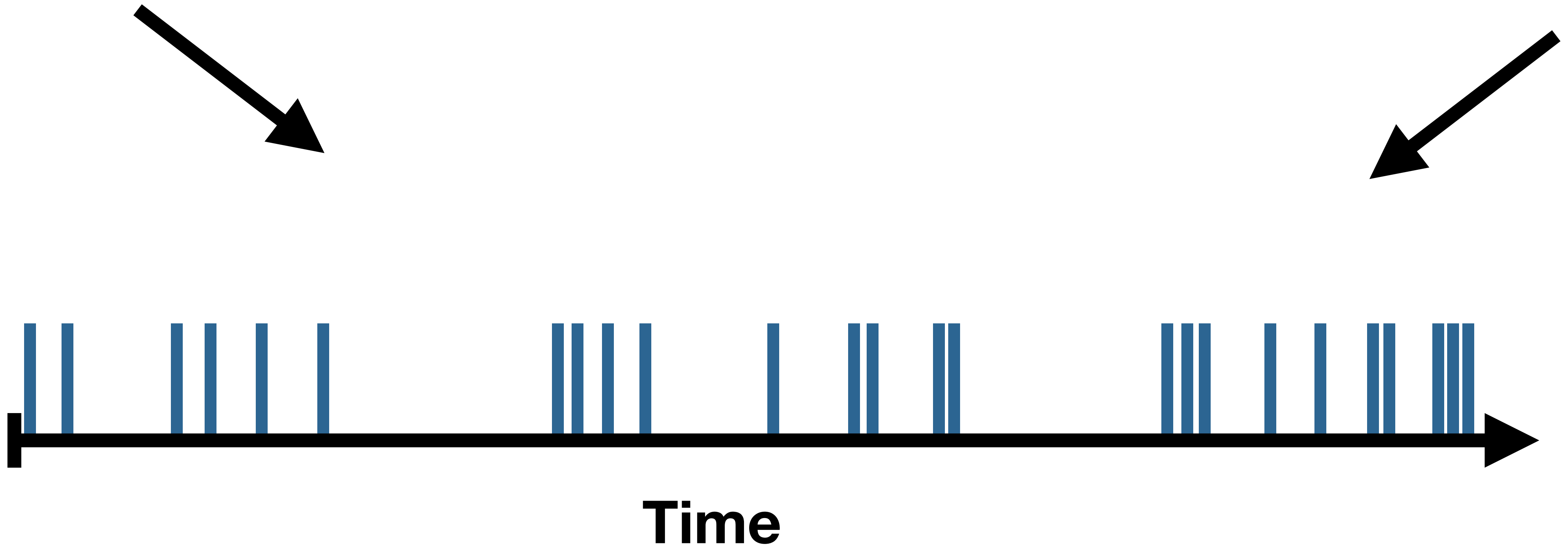
Models of TPPs

How do we specify TPPs?



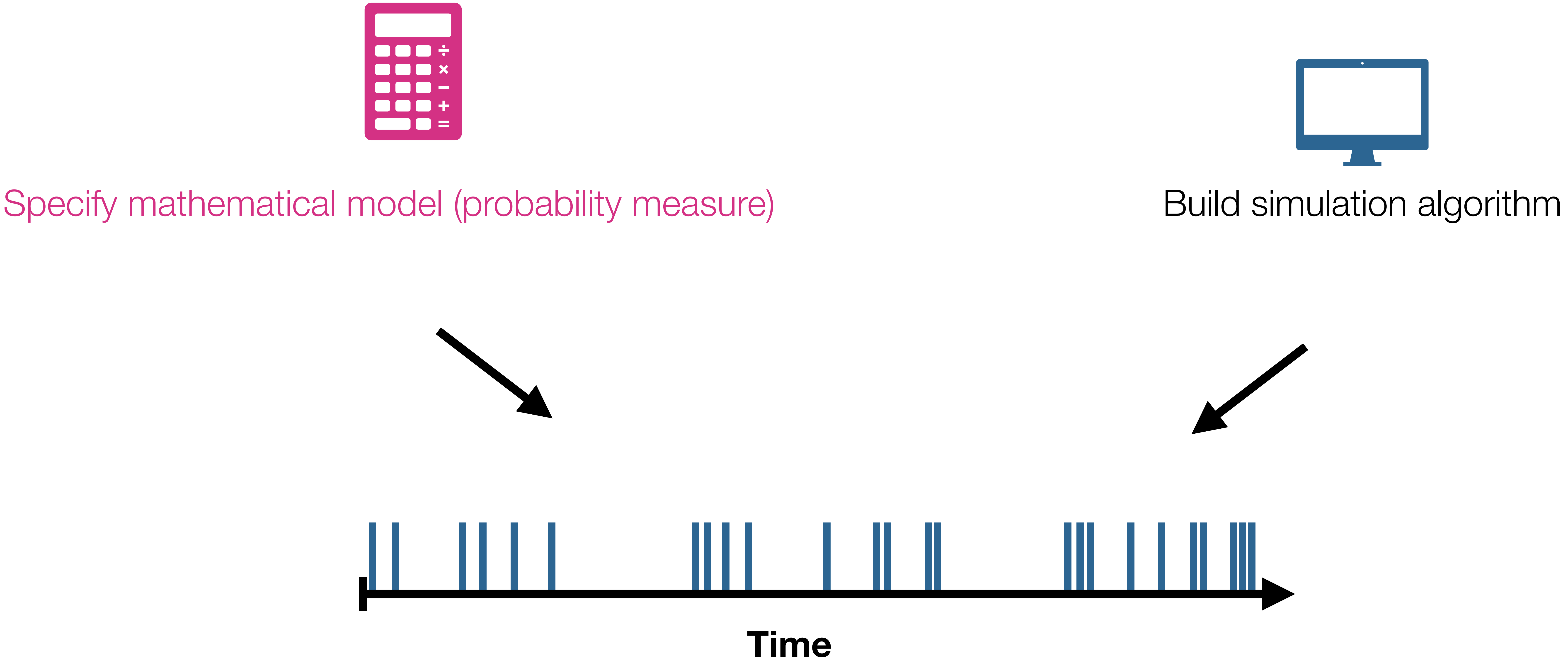
Specify mathematical model (probability measure)

Build simulation algorithm



Models of TPPs

How do we specify TPPs?



Models of TPPs

How do we specify TPPs?

Use **statistical model** or **neural network** to specify.

Build likelihood function for all (finite) sequences of fixed horizon:

$f(\cdot)$ integrates to 1 over the space of all sequences.

$$f^{pred}(\cdot)$$

The distribution over the **next event** given the history.

$$[2.6,\ 3.2] \longrightarrow \qquad \longrightarrow 2.5$$

$$[0.1,\ 0.5,\ 3.4] \longrightarrow \qquad \longrightarrow 0.0001$$

$$f(\cdot)$$

$$[0.5,\ 2.5,\ 2.7,\ 3.8] \longrightarrow \qquad \longrightarrow 10.6$$

$$[1.1,\ 2.0,\ 3.7,\ 3.9] \longrightarrow \qquad \longrightarrow 4.2$$

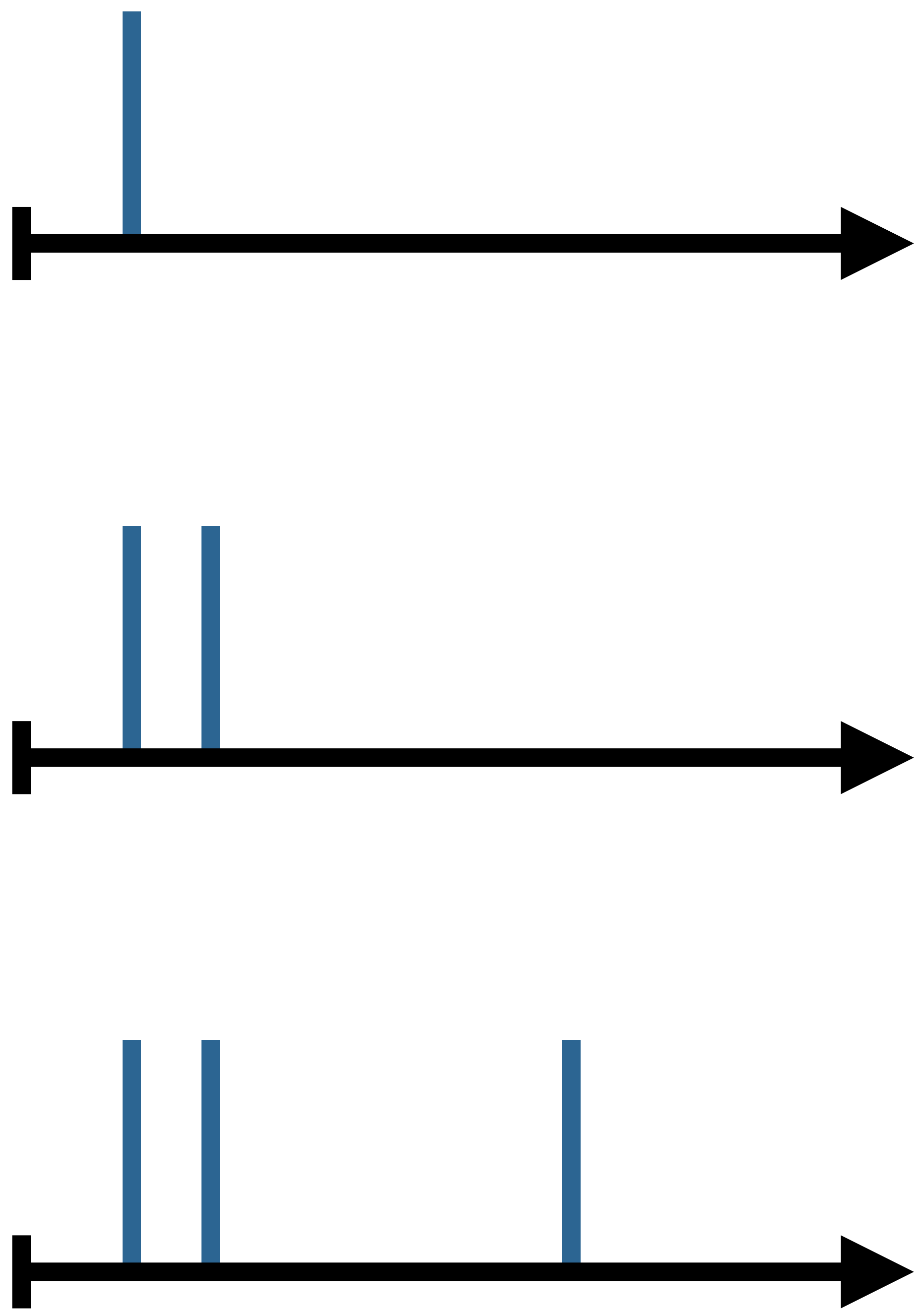
Models of TPPs

$H_0 = []$

$H_{0.1} = [0.1]$

$H_{0.2} = [0.1, 0.2]$

$H_{1.3} = [0.1, 0.2, 1.3]$

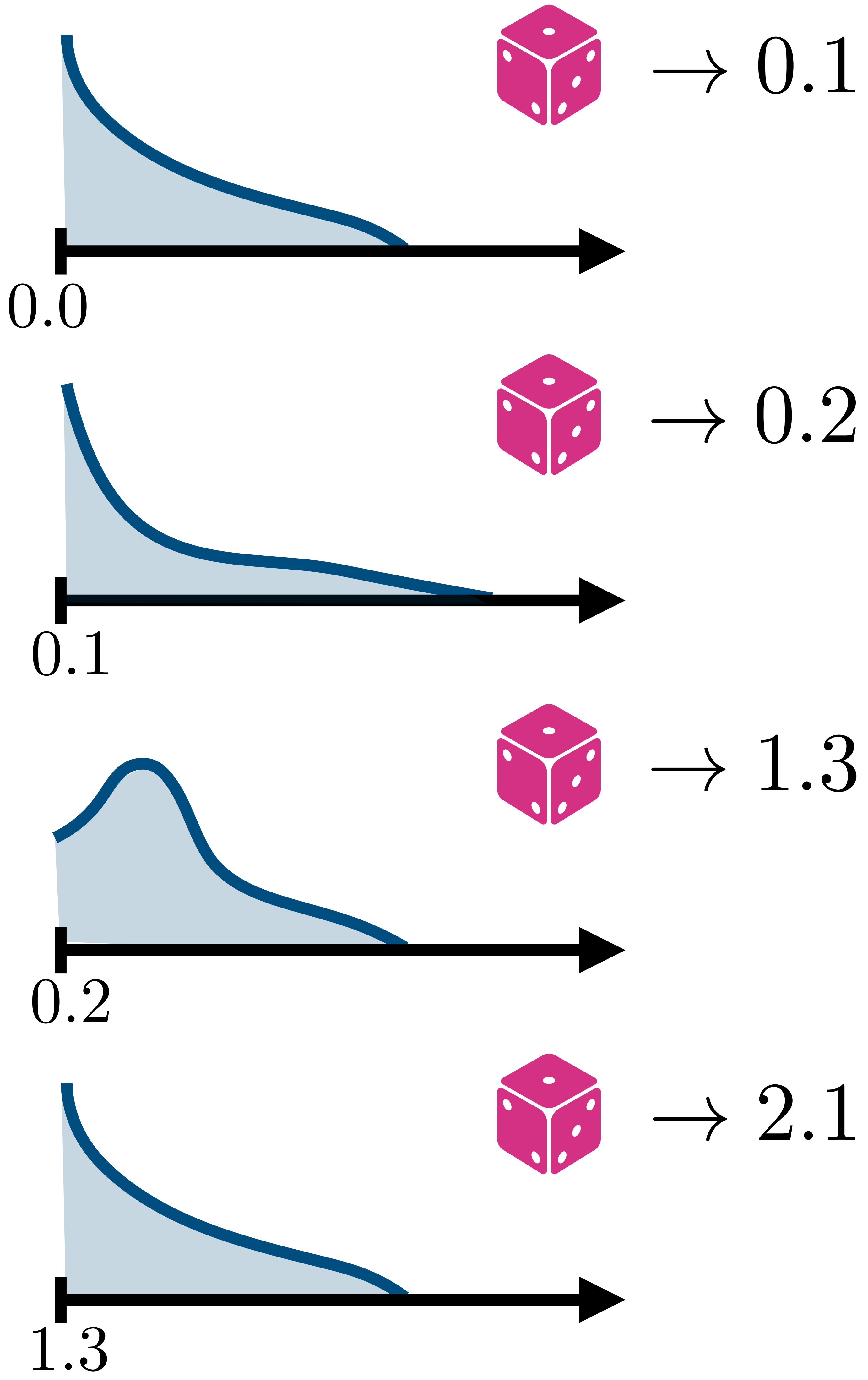


$f^{pred}(\cdot \mid H_0) \rightarrow$

$f^{pred}(\cdot \mid H_{0.1}) \rightarrow$

$f^{pred}(\cdot \mid H_{0.2}) \rightarrow$

$f^{pred}(\cdot \mid H_{1.3}) \rightarrow$



Models of TPPs

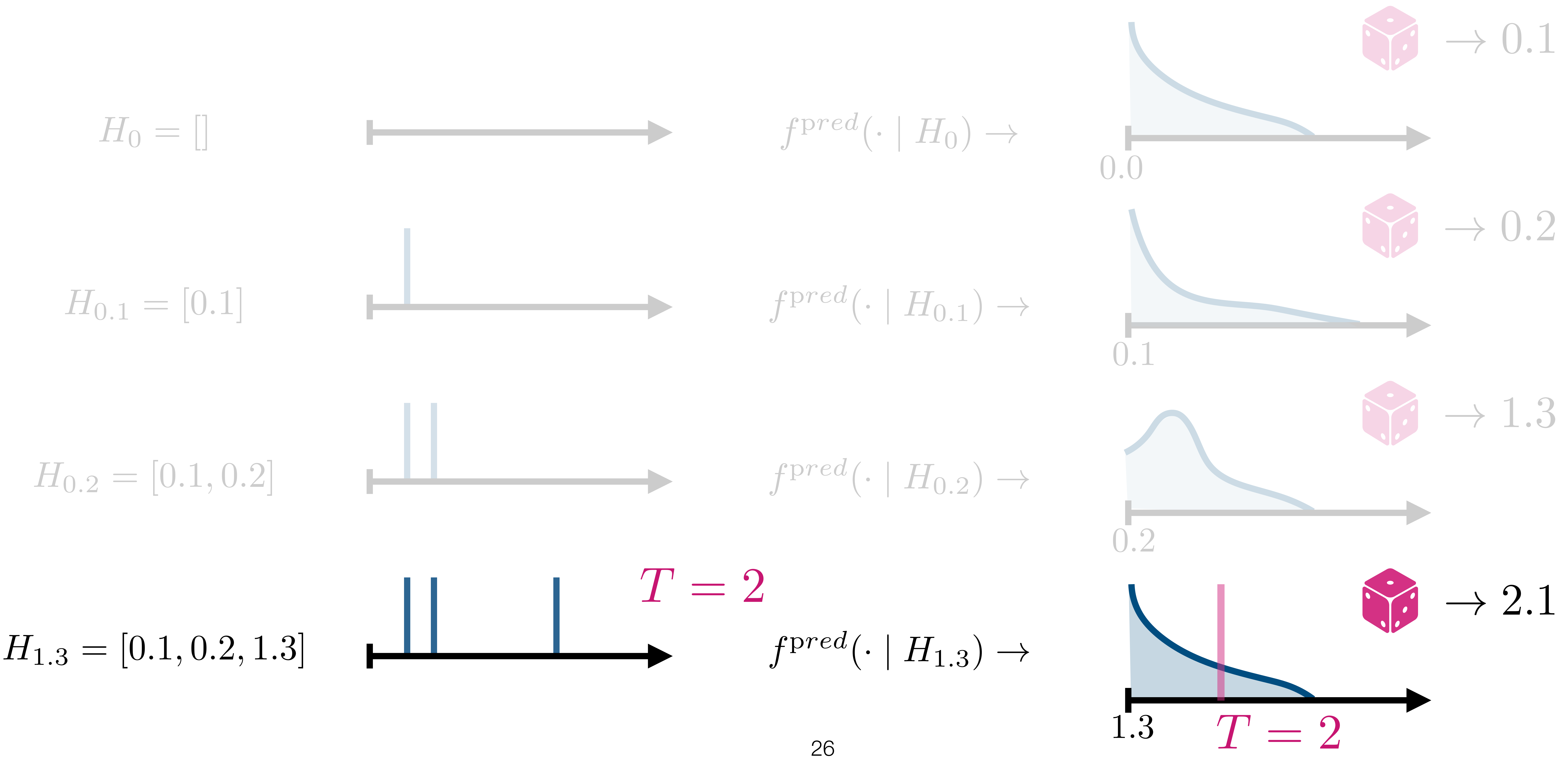
Likelihood of event sequence H.

represents the probability that no events occur from the last observed event time t_n up to the time horizon T.

$$f(H) = \left(\prod_{i=1}^{n} f^{\text{pred}}(t_{i+1} \mid H_{t_i}) \right) \left(1 - F^{\text{pred}}(T \mid H_{t_n})\right) \ .$$

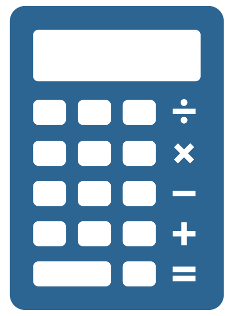
represents the probability density of the next event occurring at time t_{i+1} given the history of all previous events up to time t_i.

Models of TPPs

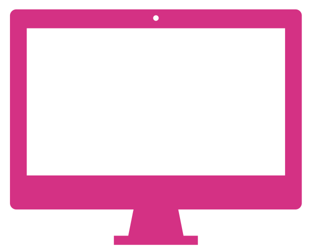


Models of TPPs

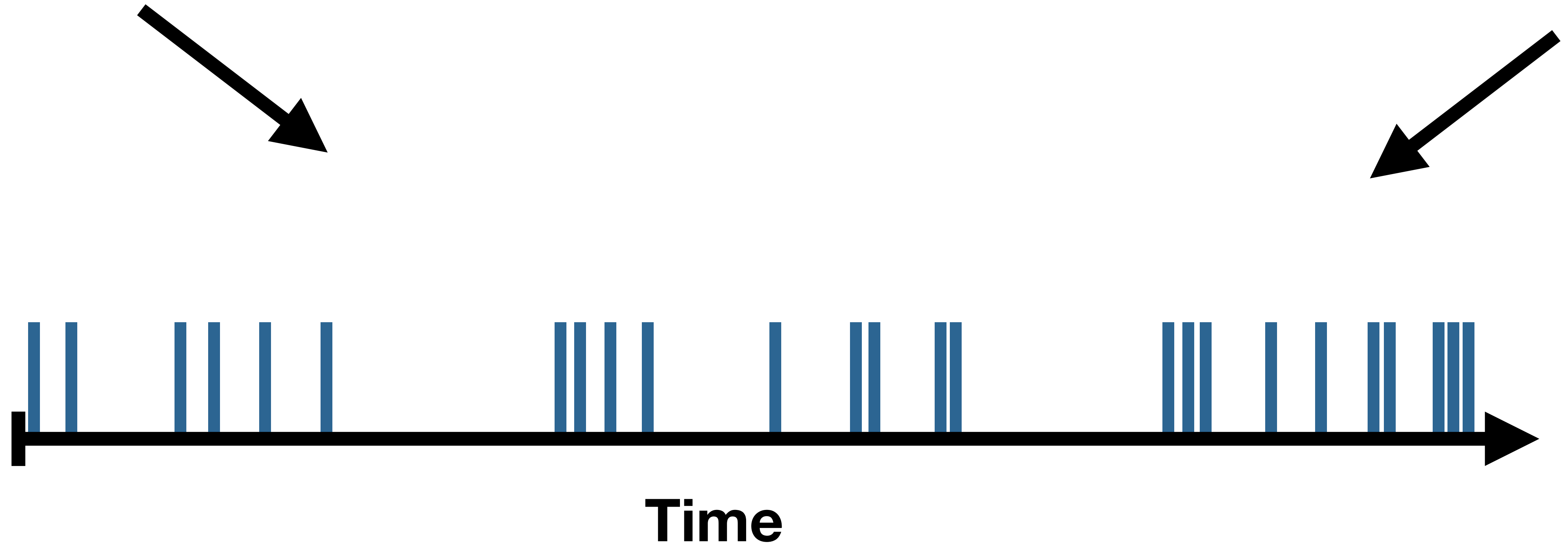
How do we specify TPPs?



Specify mathematical model (probability measure)



Build simulation algorithm

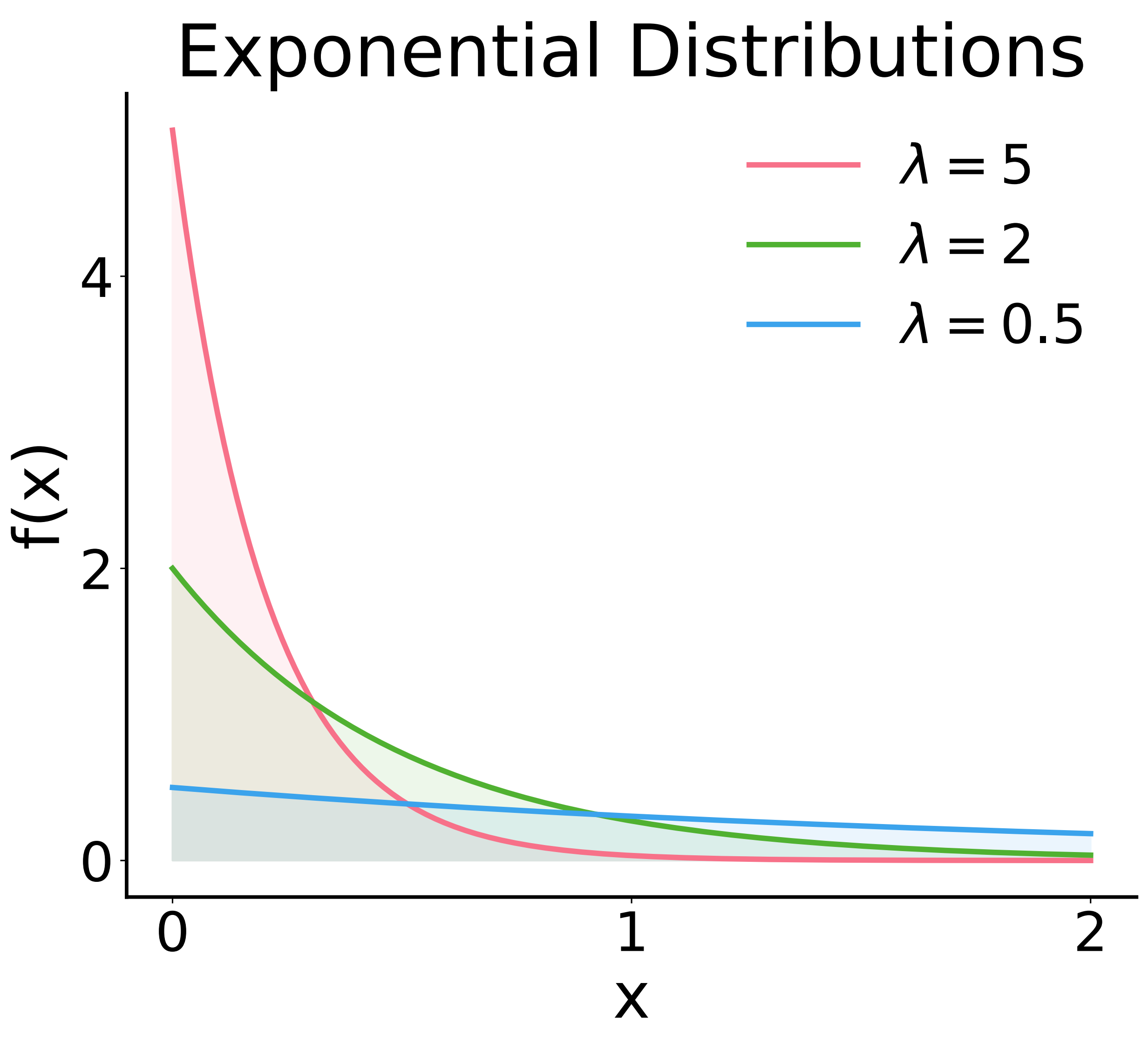


Simulating TPPs

Homogenous TPP: Inter-event times follow **exponential** distribution

Heterogeneous TPP: Inter-event times follow **arbitrary** distribution

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0, \\ 0 & x < 0 \end{cases}$$



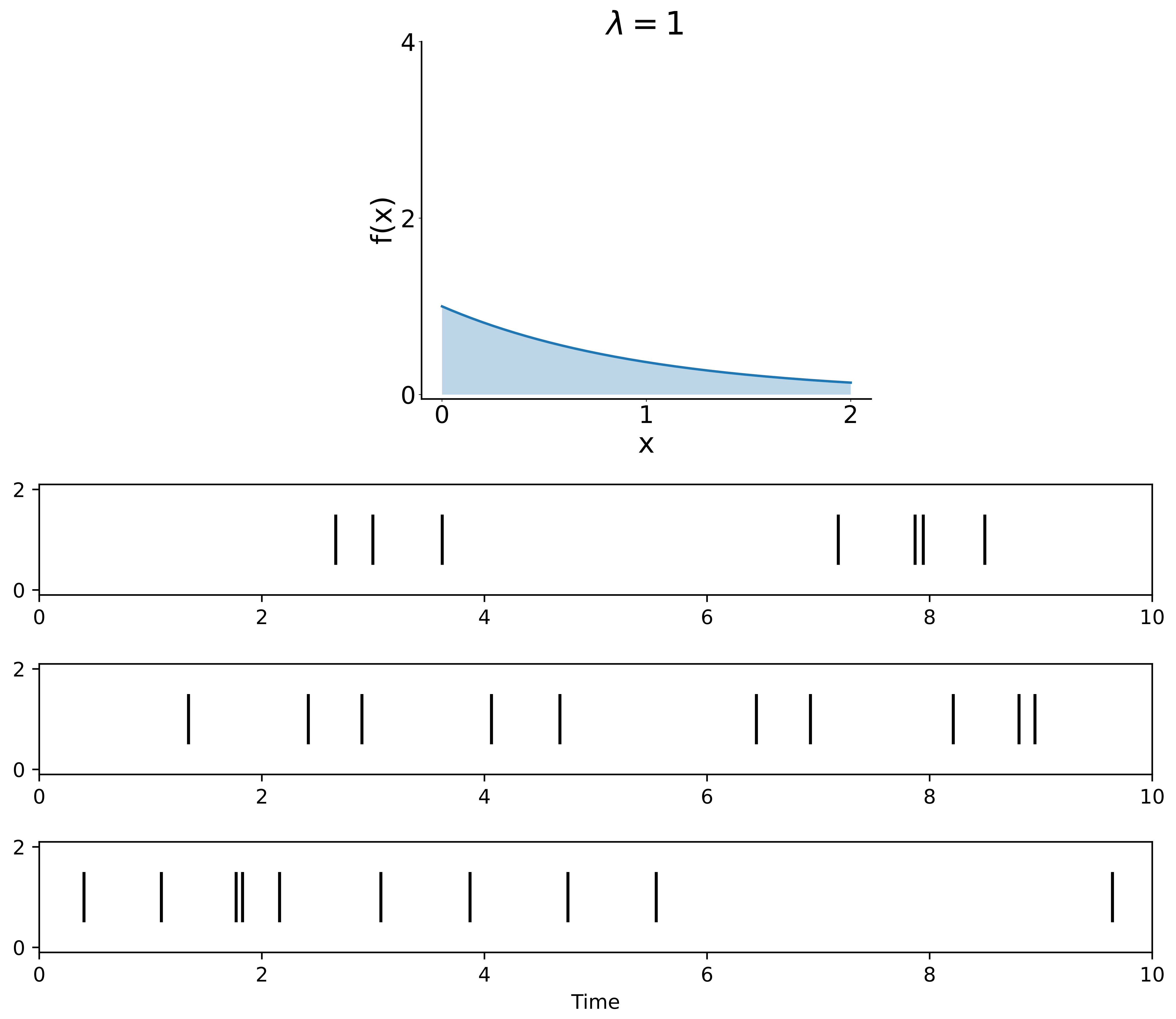
$H_{0.2} = [0.1, 0.2]$

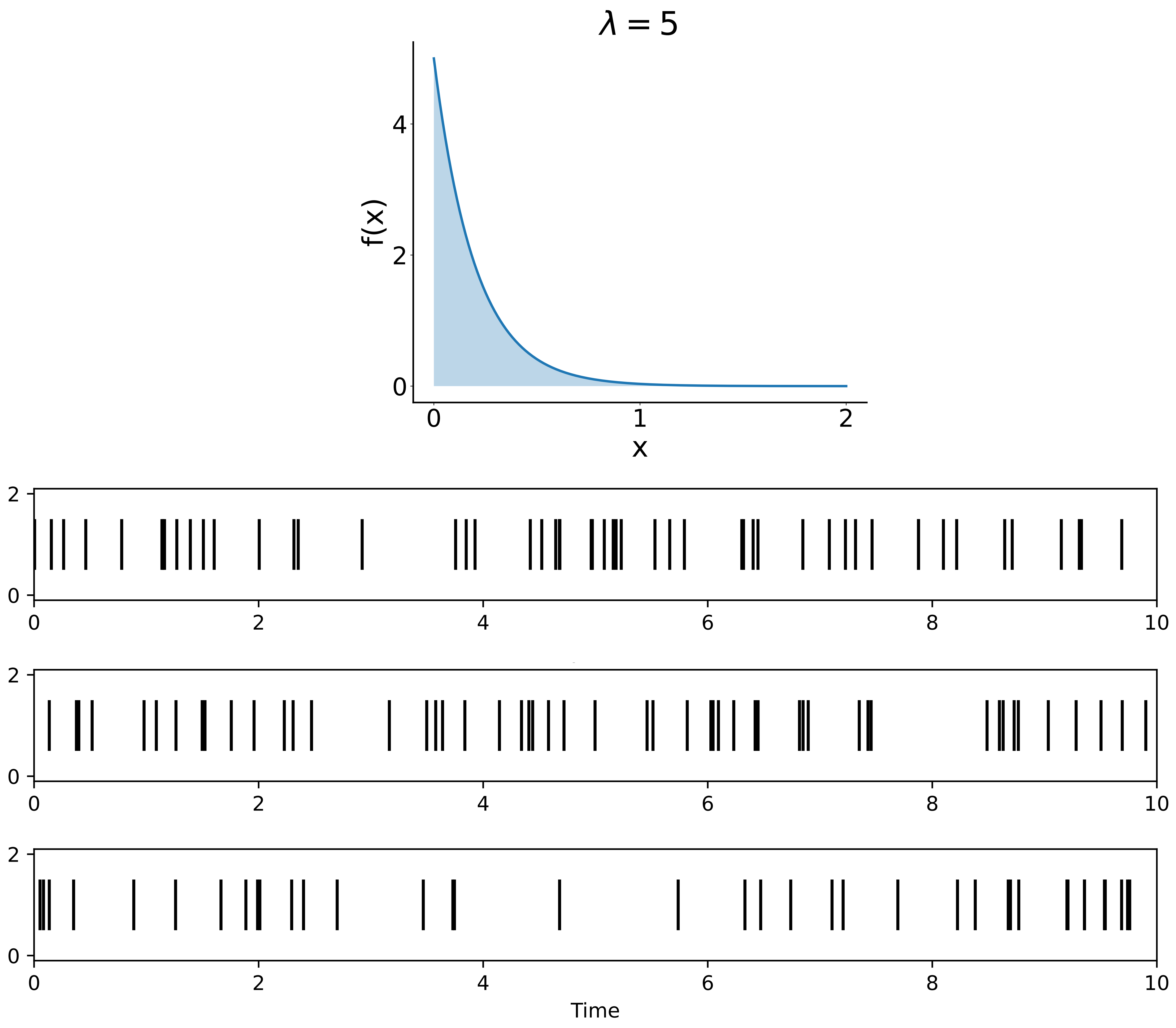
$f^{\text{pred}}(\cdot \mid H_{0.2}) \rightarrow$

$\rightarrow 1.3$

Simulating TPPs

Homogenous TPP: Inter-event times follow **exponential** distribution





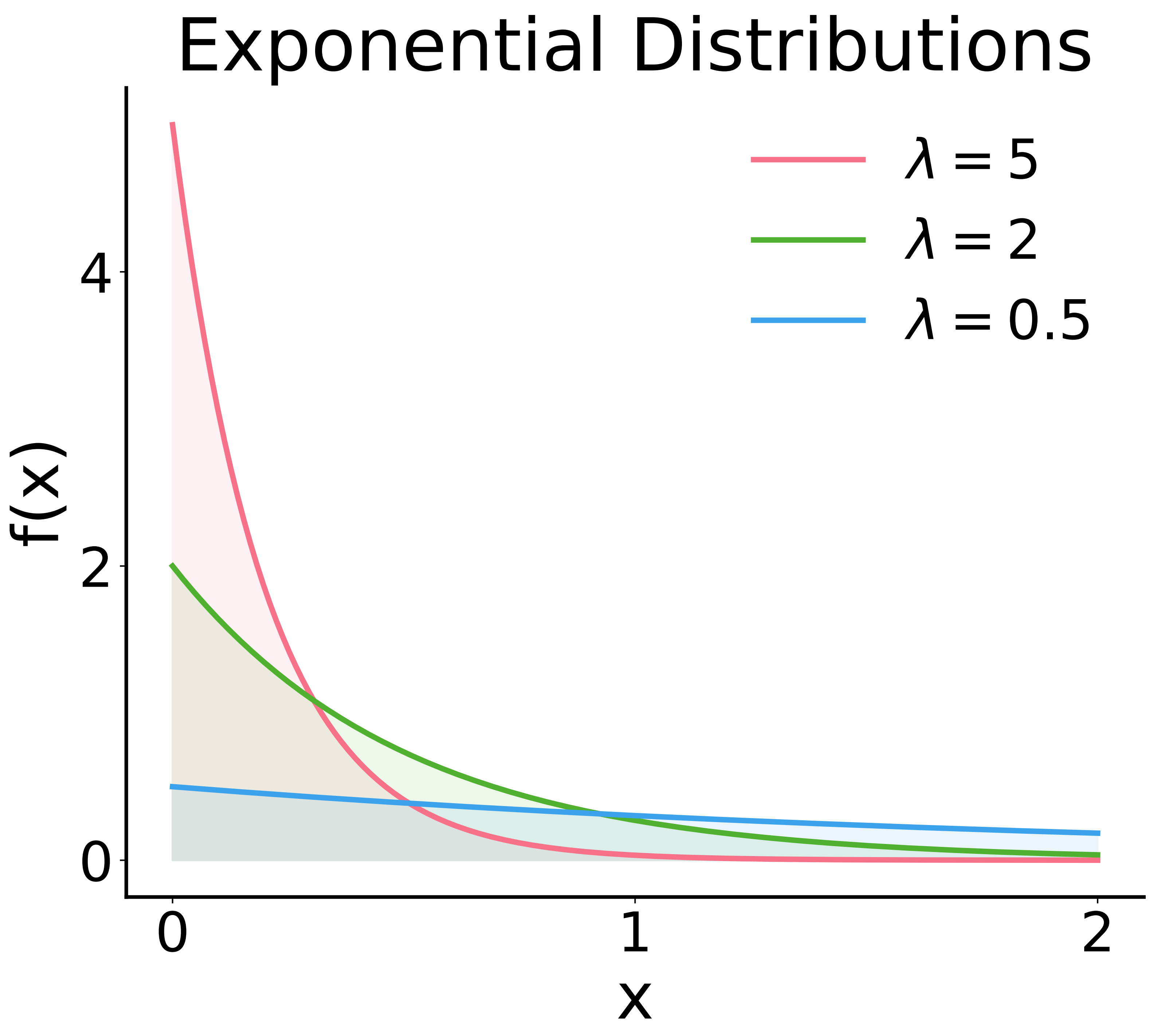
The Exponential Distribution

How do we sample from the exponential distribution?

- Use our favorite library
  - ➡ e.g., numpy.random.exponential
- Inverse transform
  - ➡ Sample U uniformly at random in [0,1]

$$X = -\frac{1}{\lambda}\ln(U)$$

- Step-wise
  - ➡ Based on time discretization



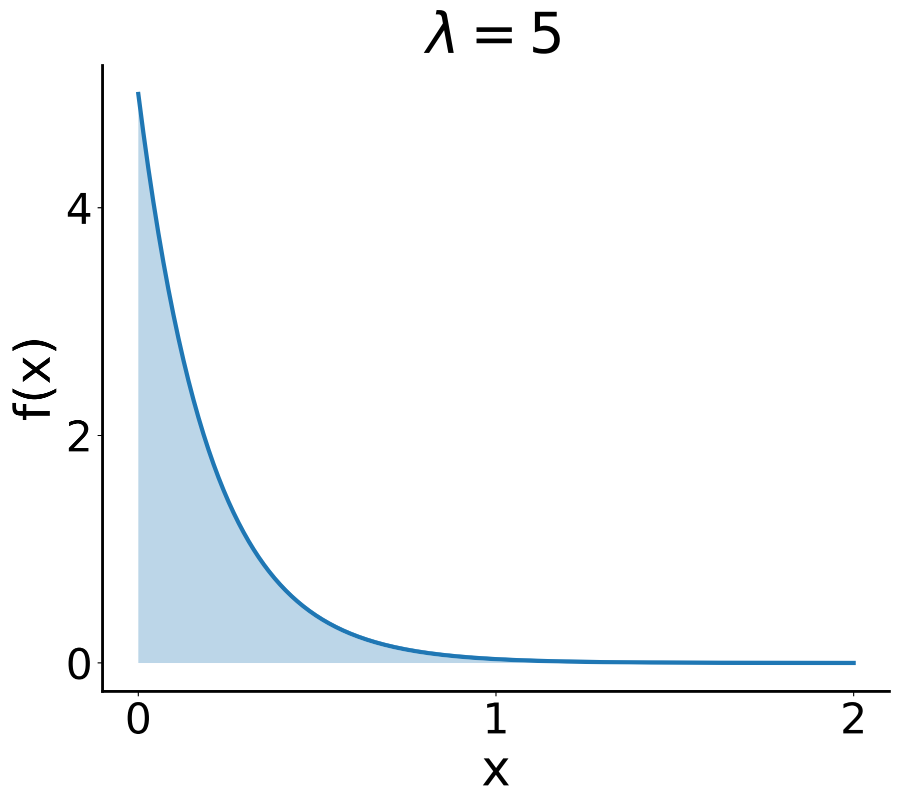
The Exponential Distribution

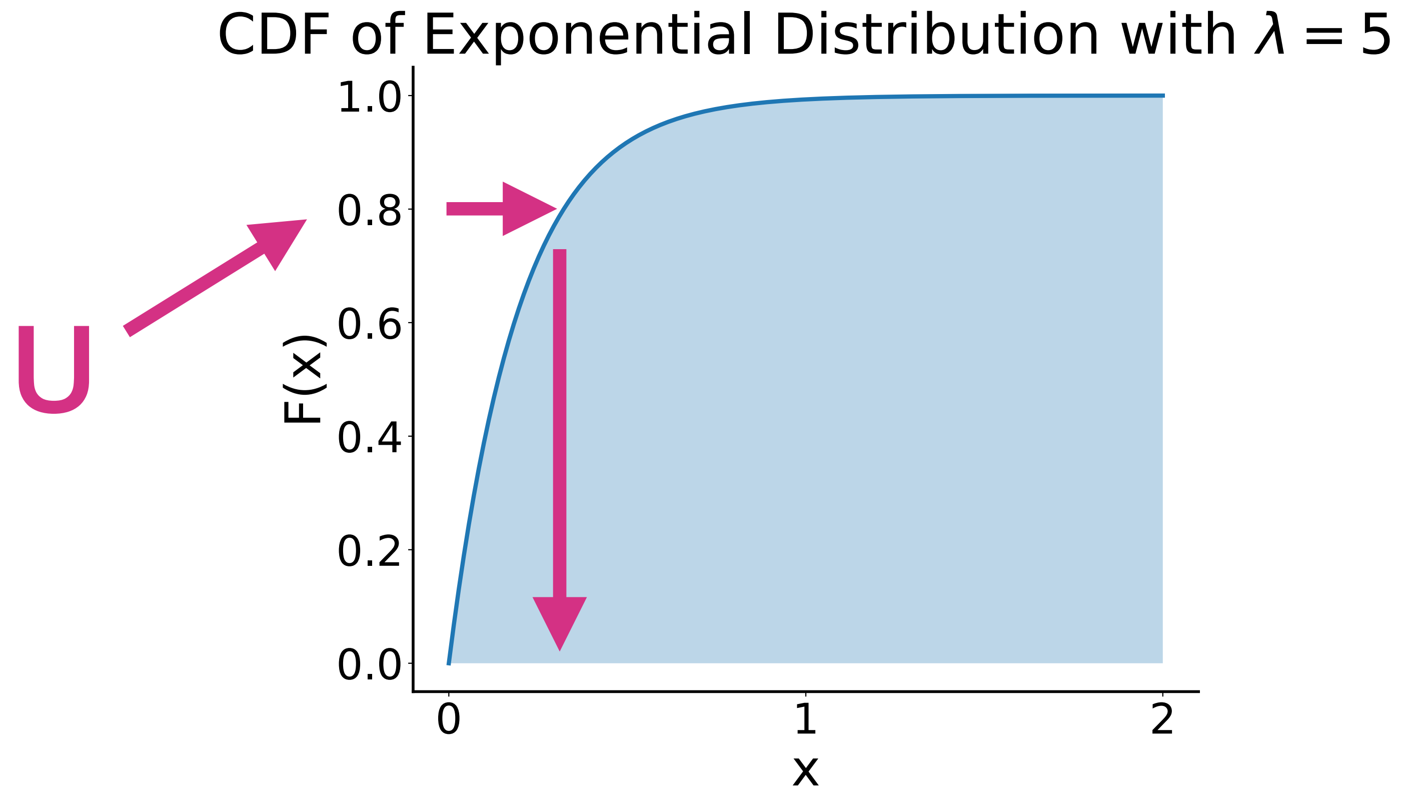
How do we sample from the exponential distribution?

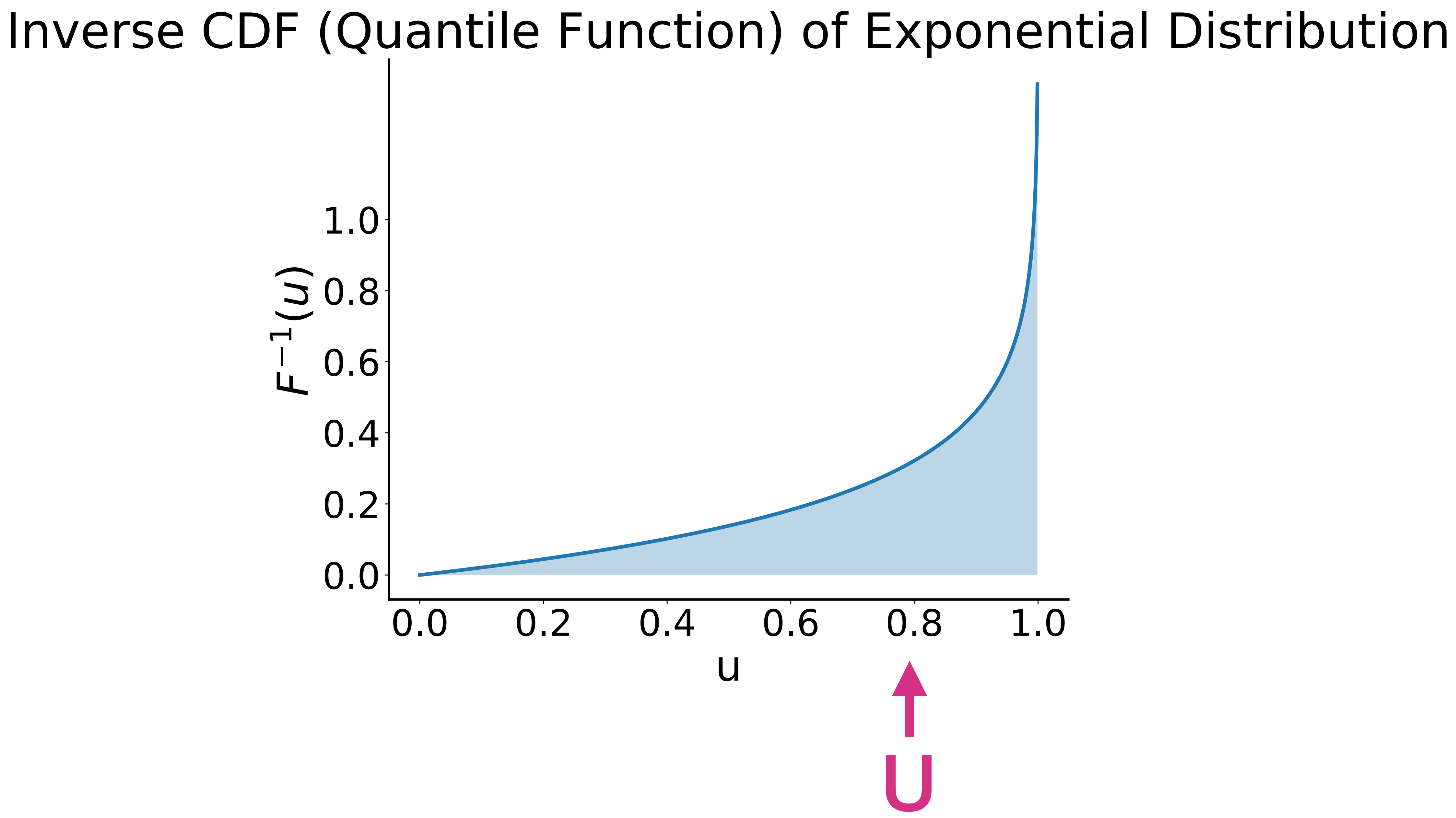
- Use our favorite library
  - ➡ e.g., numpy.random.exponential
- Inverse transform
  - ➡ Sample U uniformly at random in [0,1]

$$X = -\frac{1}{\lambda}\ln(U)$$

- Step-wise
  - ➡ Based on time discretization







The Exponential Distribution

How do we sample from the exponential distribution?

- Use our favorite library
  - ➡ e.g., numpy.random.exponential

- Inverse transform
  - ➡ Sample U uniformly at random in [0,1]

$$X = -\frac{1}{\lambda}\ln(U)$$

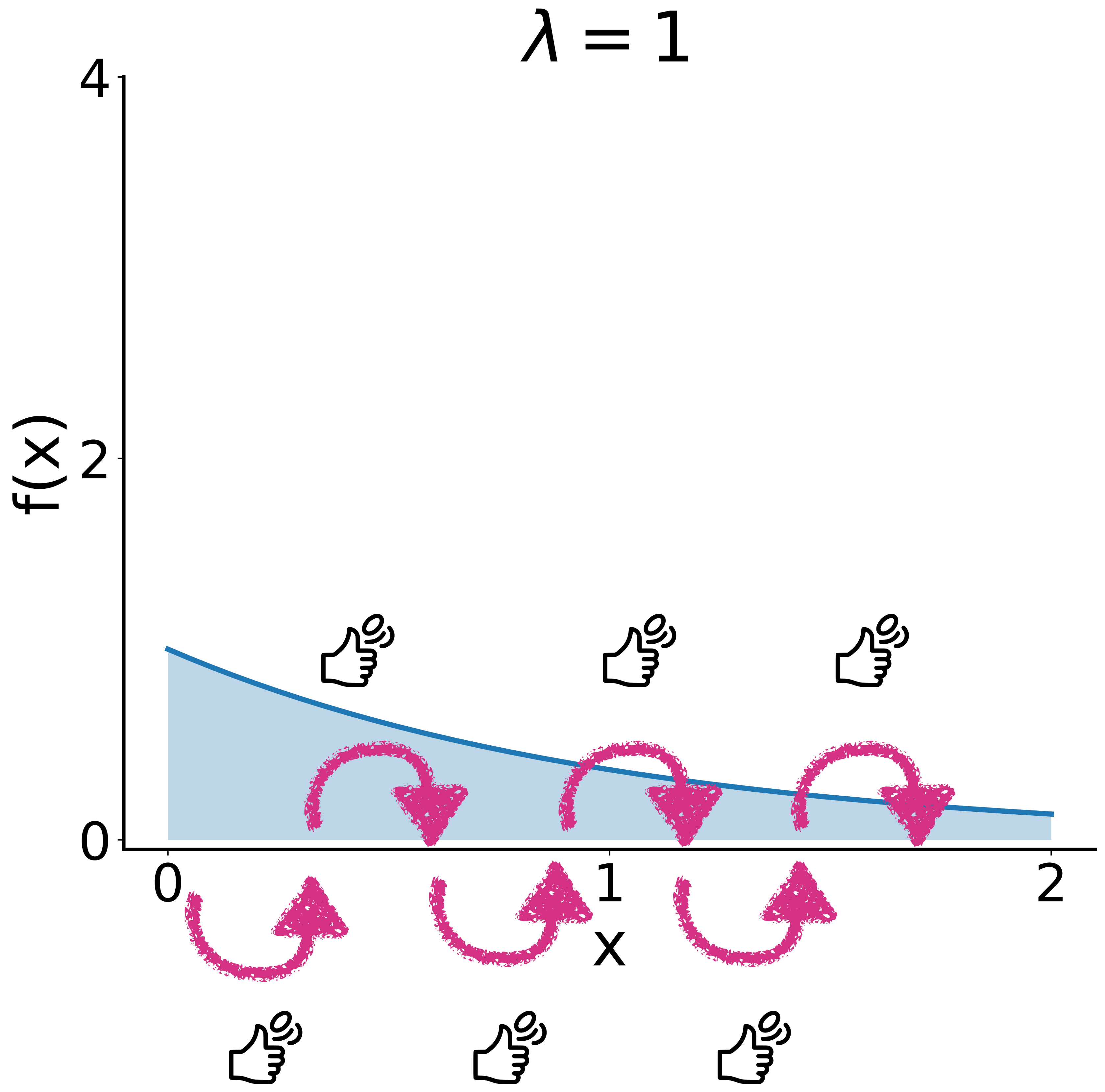
- Step-wise
  - ➡ Based on time discretization

The Exponential Distribution

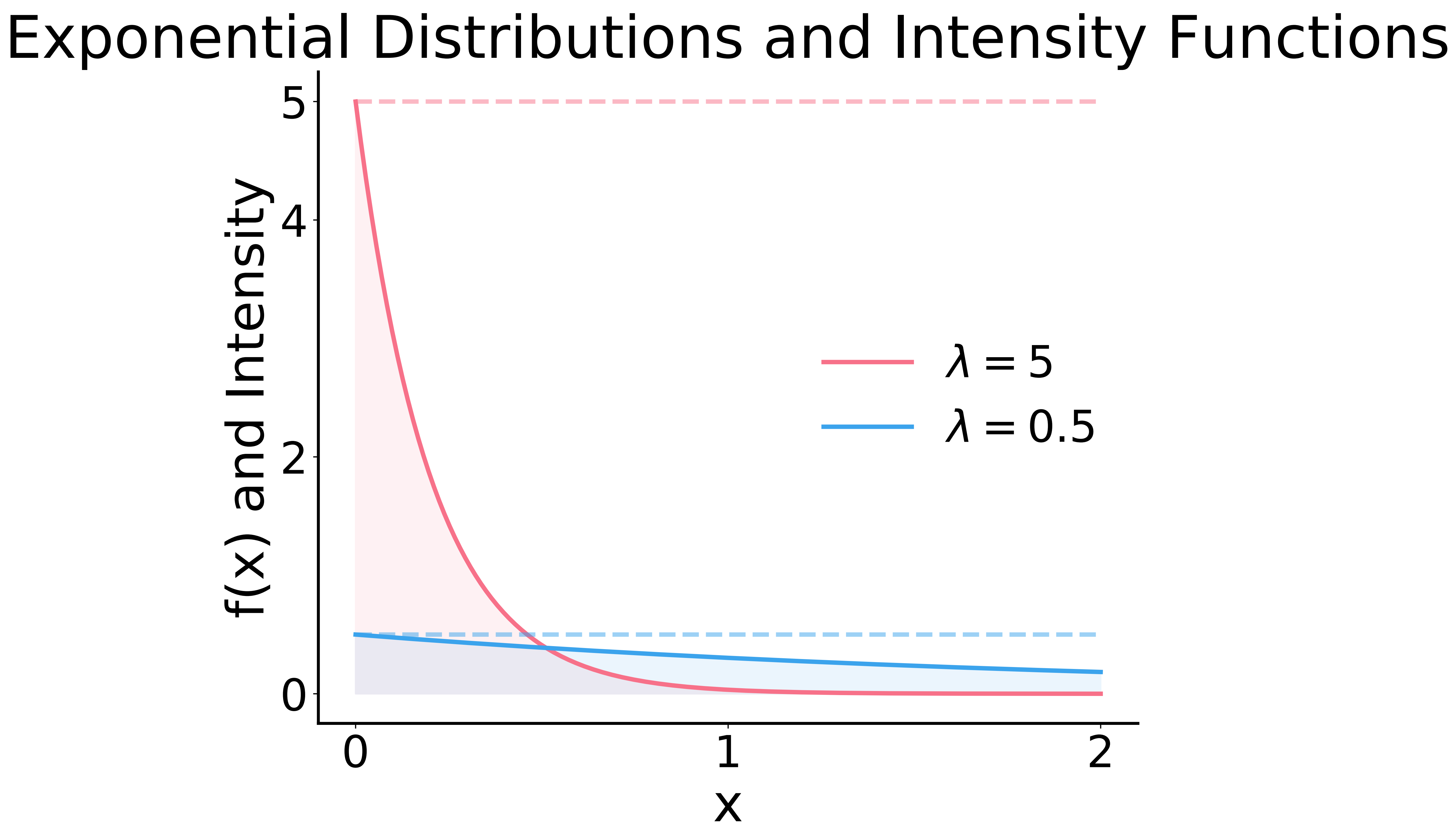
1. **Discretize Time:** Choose a small time step Δt.
2. **Biased Coin:** At each time step, use a biased coin to decide whether an event occurs. The probability p of the event occurring in each time step is $p = \lambda \Delta t$.
3. **Simulate Until Event:** Continue this process until the event occurs.

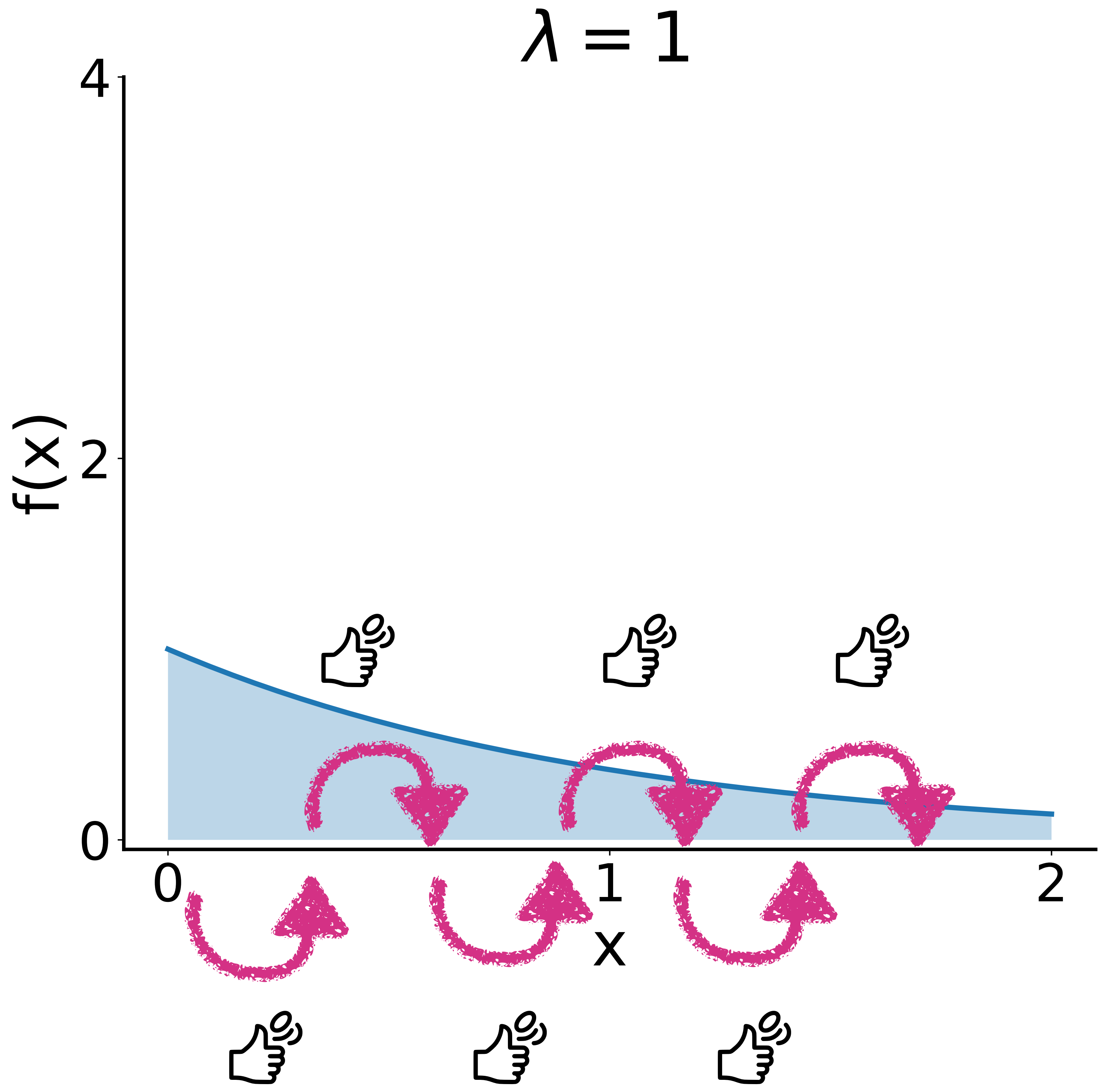
The intensity (in a homogeneous TPP) is constant

$$\lambda = \lambda(t) = \lim_{\Delta t \to 0} \frac{P(\text{Event in } [t, t + \Delta t))}{\Delta t}$$



The Exponential Distribution





$$\lambda = \lambda(t) = \lim_{\Delta t \to 0} \frac{P(\text{Event in } [t, t + \Delta t))}{\Delta t}$$

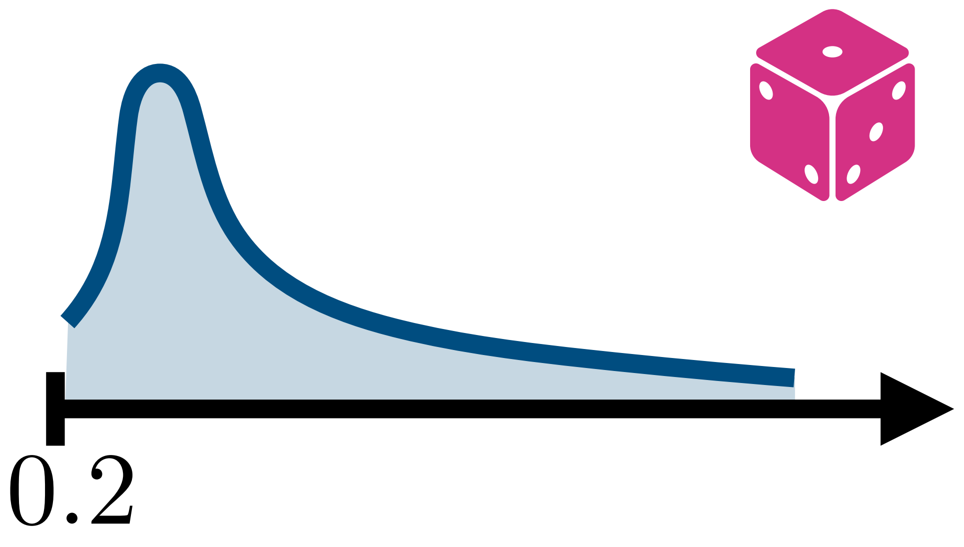
Simulating TPPs

Homogenous TPP: Inter-event times follow exponential distribution

Heterogeneous TPP: Inter-event times follow arbitrary distribution

$H_{0.2} = [0.1, 0.2]$

$f^{pred}(\cdot \mid H_{0.2}) \rightarrow$



$\rightarrow 1.3$

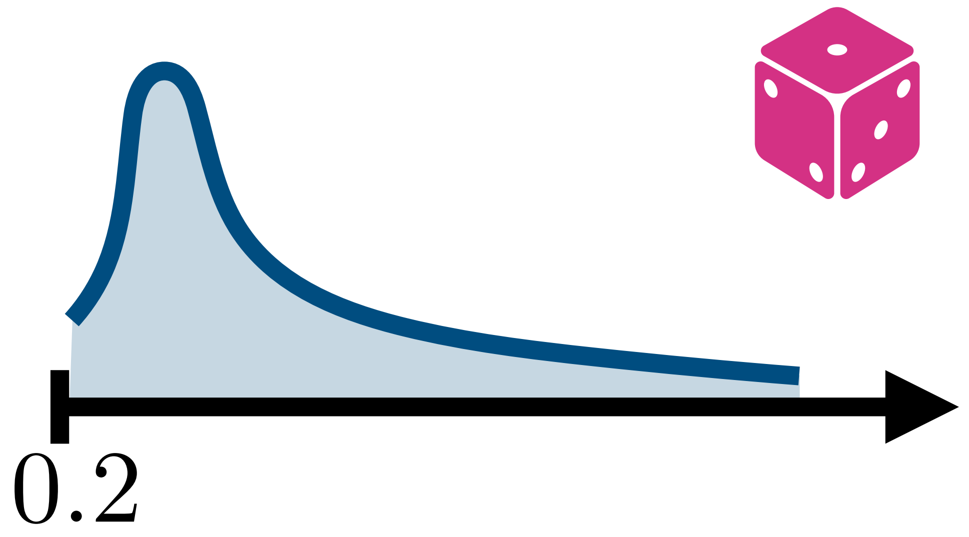
Goal:

- Allow arbitrary inter-event time distributions.
- Enable the event time distribution to depend on the history, allowing us to model effects such as periodicity, self-excitation, and self-correction.

Simulating TPPs

$H_{0.2} = [0.1, 0.2]$

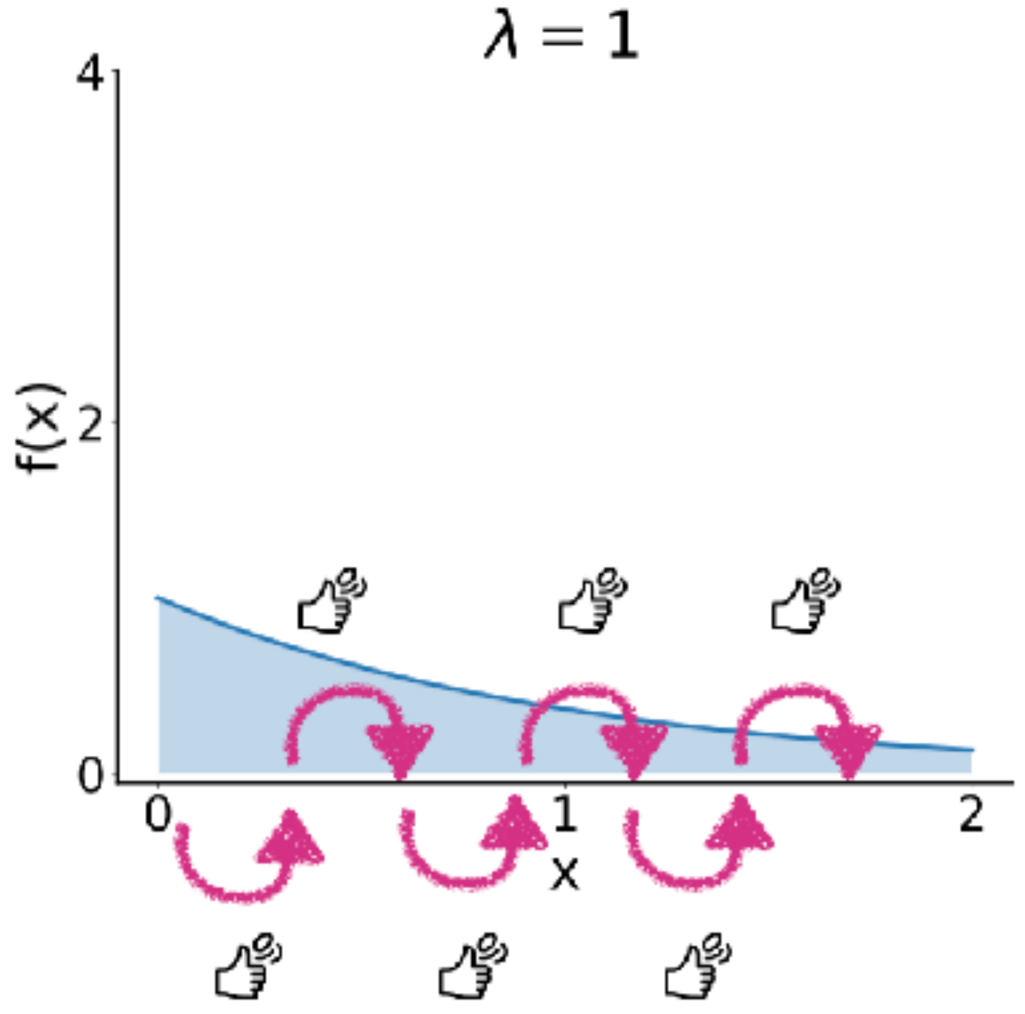
$f^{pred}(\cdot \mid H_{0.2}) \rightarrow$



$\rightarrow 1.3$

Goal:

- Allow arbitrary inter-event time distributions.
- Enable the event time distribution to depend on the history, allowing us to model effects such as periodicity, self-excitation, and self-correction.



The probability of an event happening changes over time

Method:

Construct a time and history dependent intensity function:

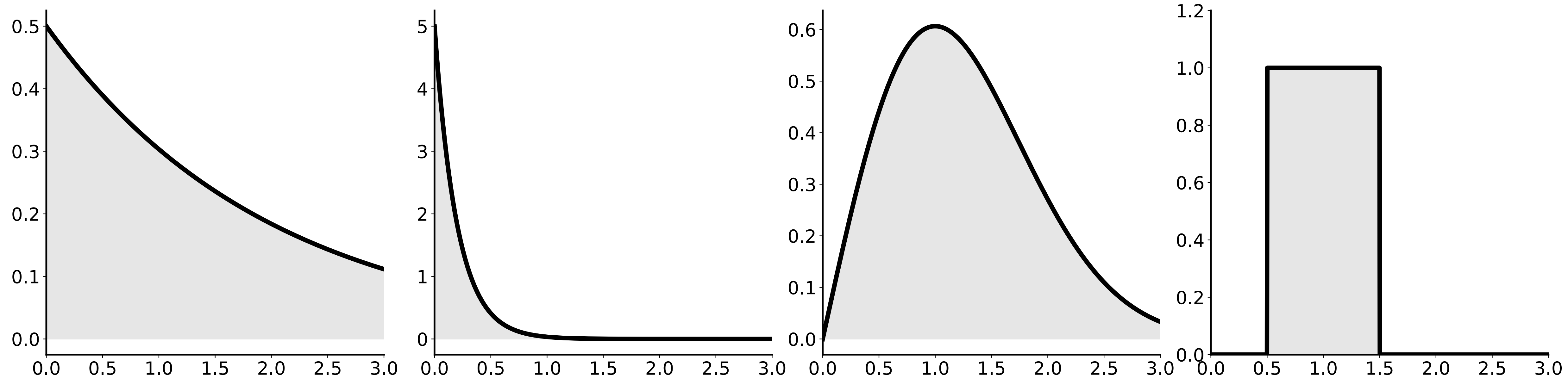
$$\lambda : \mathbb{R}_{\geq 0} \times \mathcal{H} \rightarrow \mathbb{R}_{\geq 0}$$

$$\lambda(t, H_t) = \lim_{\Delta t \rightarrow 0} \frac{P(\text{Event in } [t, t + \Delta t) \mid H_t)}{\Delta t}$$

Intensity Function

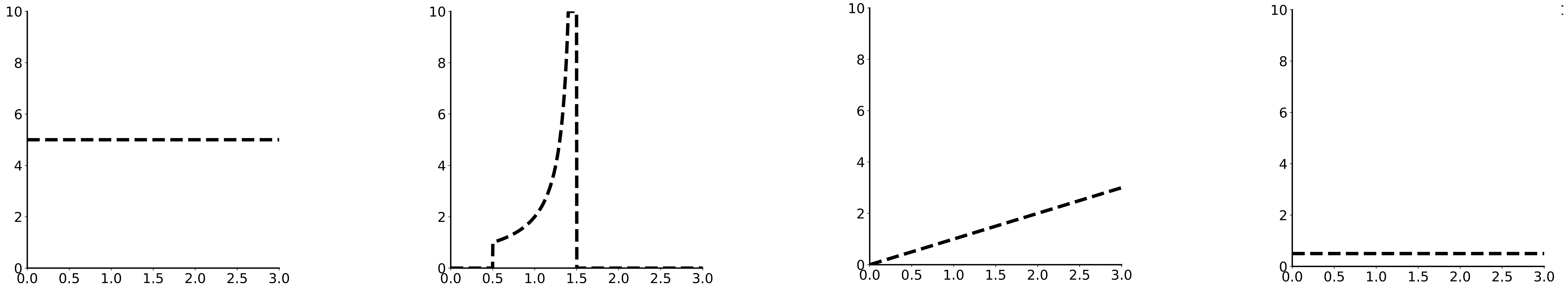
Predictive distribution:

PDF over inter-event times.



Intensity Function:

Instantaneous rate of
event happening at *t.*



Questions:

Which PDF corresponds to each intensity function

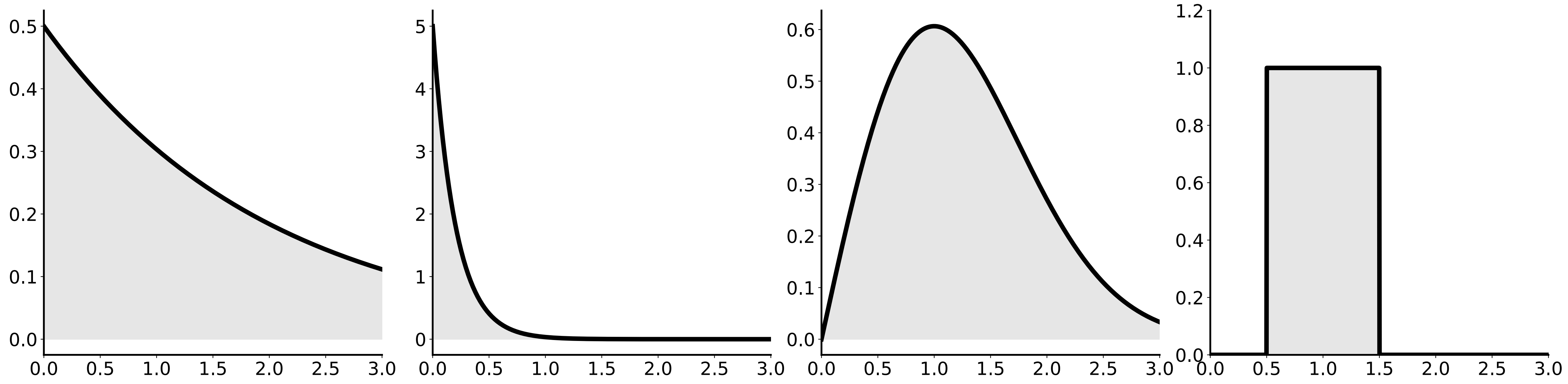
What are the properties of intensity functions?

How can we convert them into each other?

Intensity Function

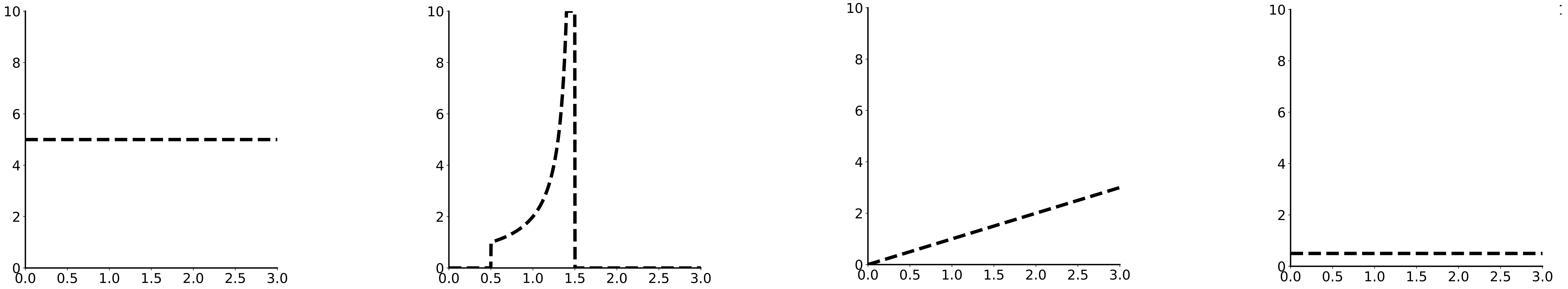
Predictive distribution:

PDF over inter-event times.



Intensity Function:

Instantaneous rate of
event happening at *t.*



Questions:

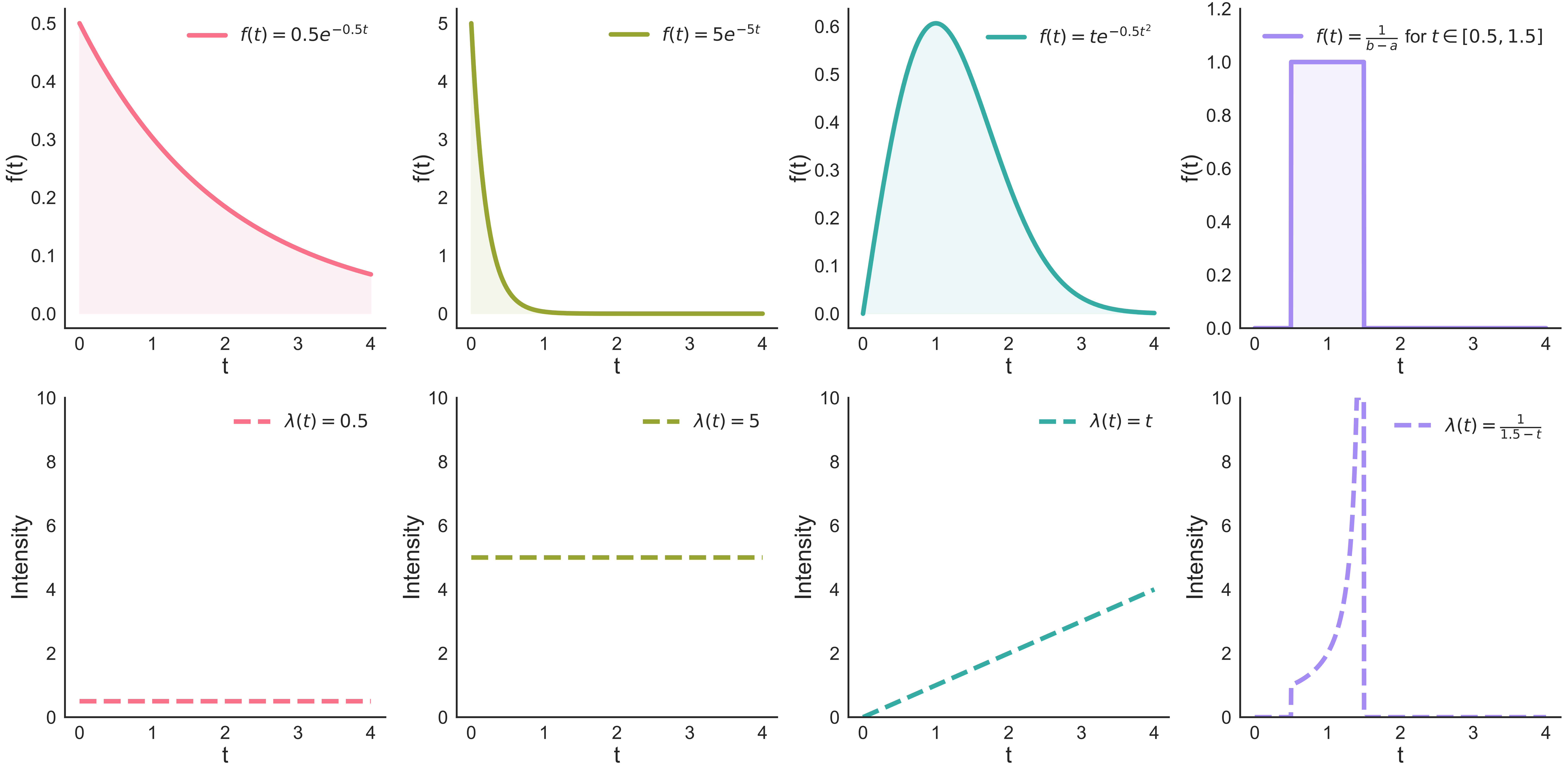
Which PDF corresponds to each intensity function

What are the properties of intensity functions?

How can we convert them into each other?

The probability of an event happening changes over time

Intensity Function



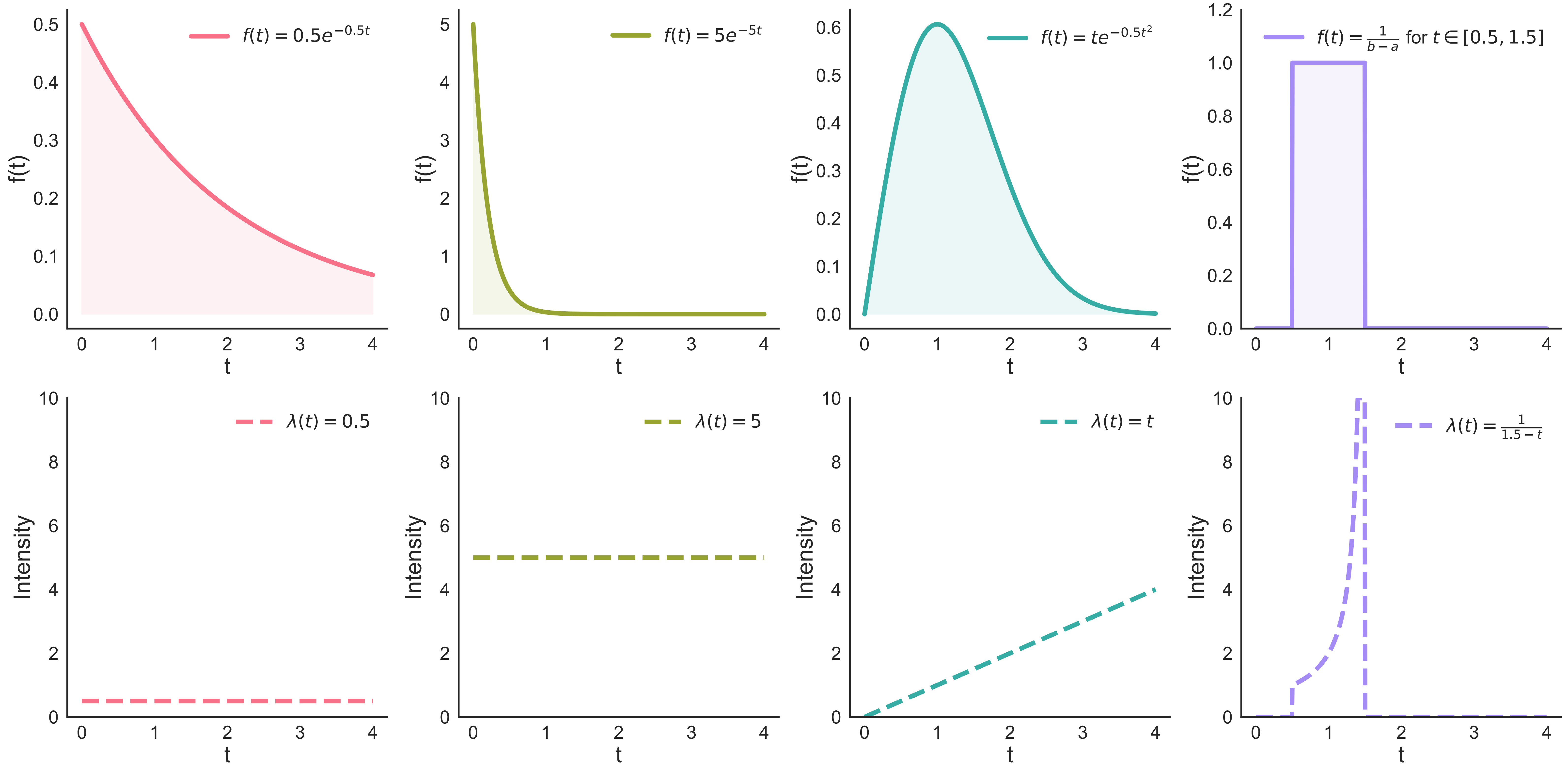
Intensity Function

Predictive distribution:

PDF over inter-event times.

Intensity Function:

Instantaneous rate of
event happening at *t.*



Questions:

Which PDF corresponds to each intensity function

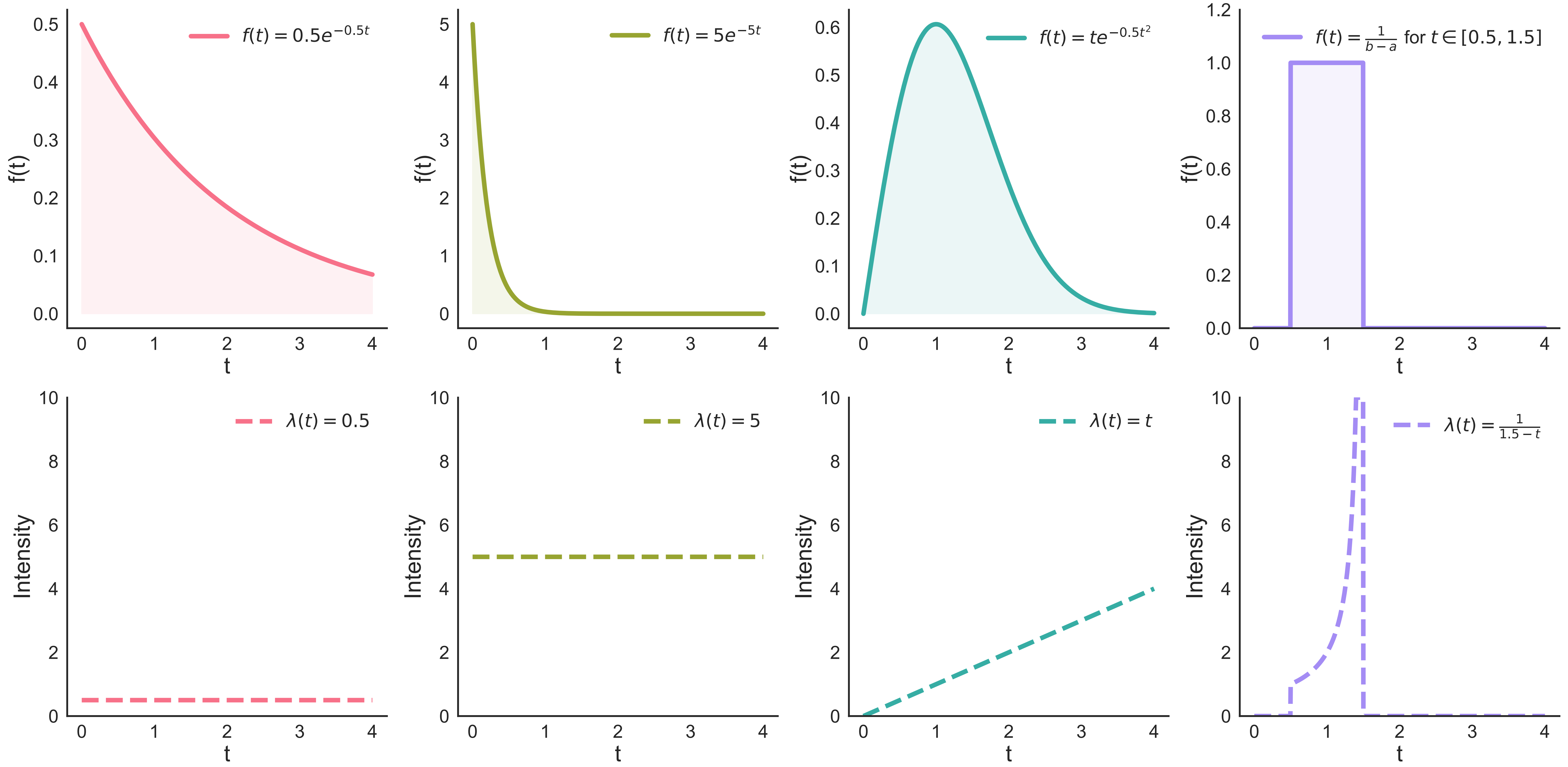
What are the properties of intensity functions?

How can we convert them into each other?

Intensity Function

Predictive distribution:

PDF over inter-event times.



Intensity Function:

Instantaneous rate of event happening at *t.*

Questions:

Which PDF corresponds to each intensity function

What are the properties of intensity functions?

How can we convert them into each other?

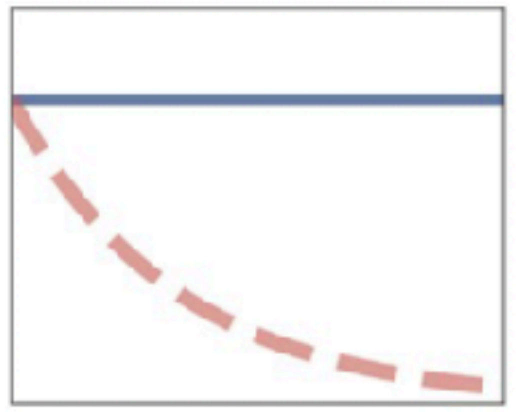
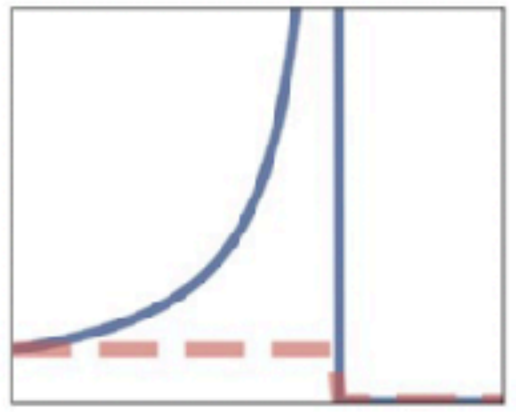
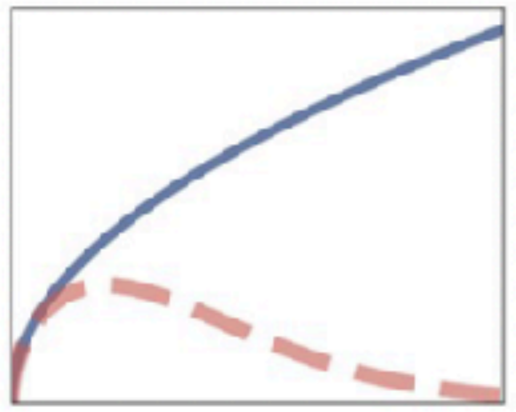
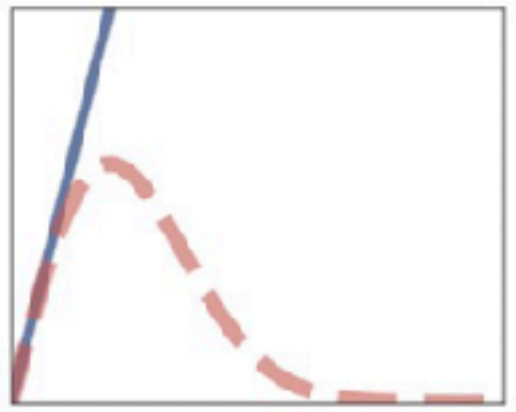
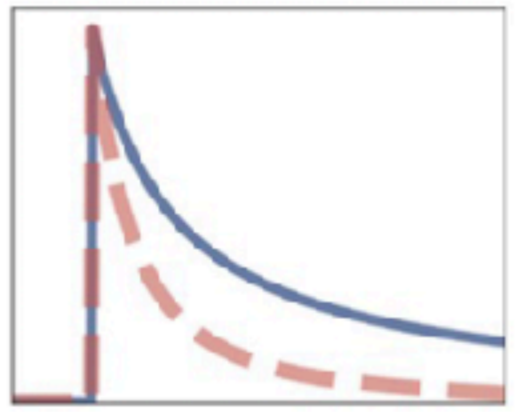
Intensity Function

Predictive distribution:

PDF over inter-event times.

Intensity Function:

Instantaneous rate of event happening at *t.*

	Exponential	Uniform	Weibull	Rayleigh	Power law
y-axis: γ (dashed), λ (solid) **x-axis:** time					
Parameters	$\lambda \in \mathbb{R}_{>0}$	$a, b \in \mathbb{R}_{>0}$ $a < b$	$c, u \in \mathbb{R}_{>0}$	$\sigma \in \mathbb{R}_{>0}$	$\alpha, t_{\min} \in \mathbb{R}_{>0}$ $\alpha > 1$
Intensity $\lambda(t)$	λ	$\frac{\mathbb{1}_{t\in[a,b]}}{1-\frac{t}{b-a}}$	$cu(tu)^{c-1}$	$\frac{t}{\sigma^2}$	$\mathbb{1}_{t\geq t_{\min}} \frac{\alpha-1}{t}$
PDF $\gamma(t)$	$\lambda e^{-\lambda t}$	$\mathbb{1}_{t\in[a,b]}$	$cu(tu)^{c-1}e^{-(tu)^c}$	$\frac{t}{\sigma^2}e^{-\frac{t^2}{2\sigma^2}}$	$\mathbb{1}_{t\geq t_{\min}} \frac{\alpha-1}{t_{\min}} \left(\frac{t}{t_{\min}}\right)^{-\alpha}$

Efficient simulation of non-Markovian dynamics on complex networks (Großmann et al.)

Questions:

Which PDF corresponds to each intensity function

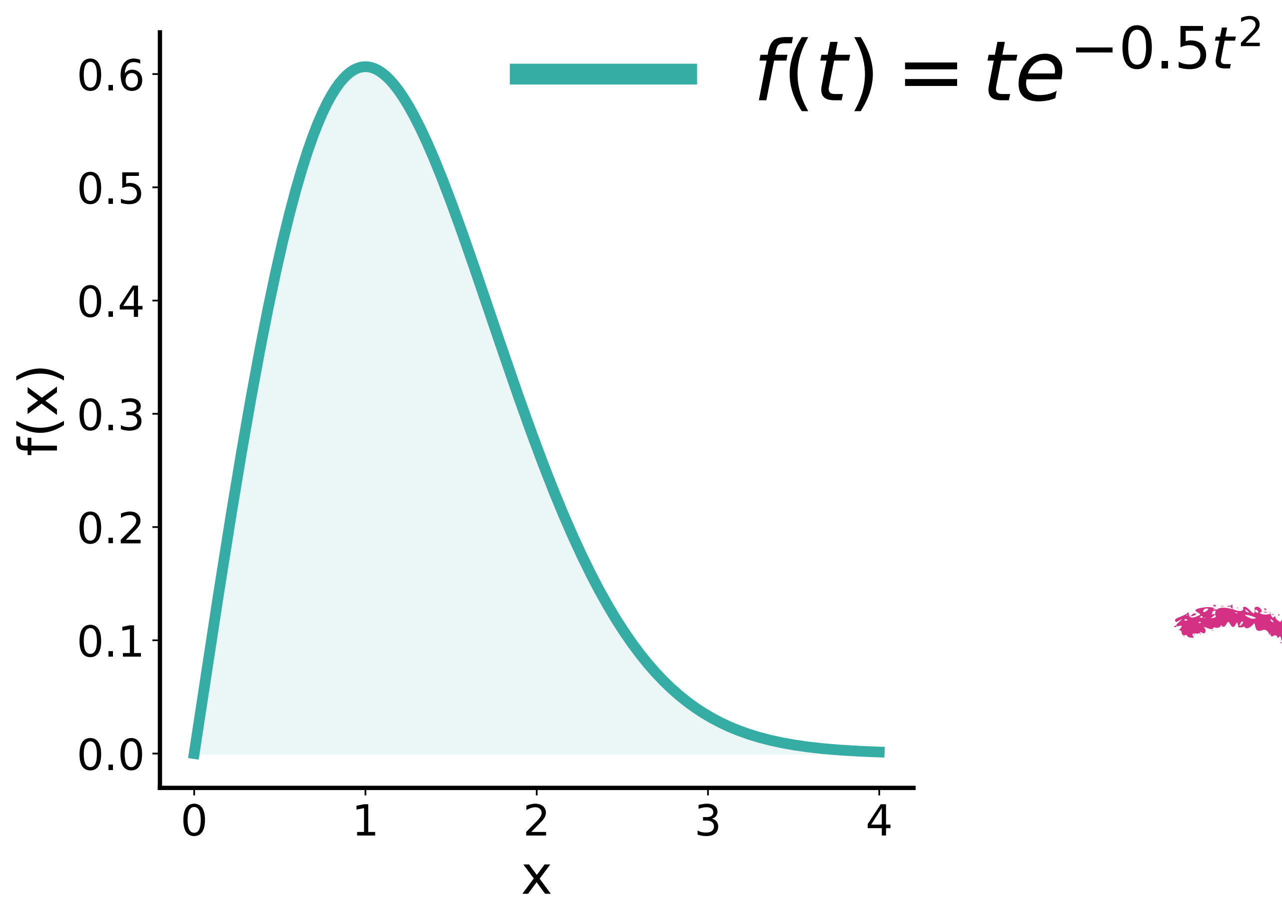
What are the properties of intensity functions?

How can we convert them into each other?

Intensity Function

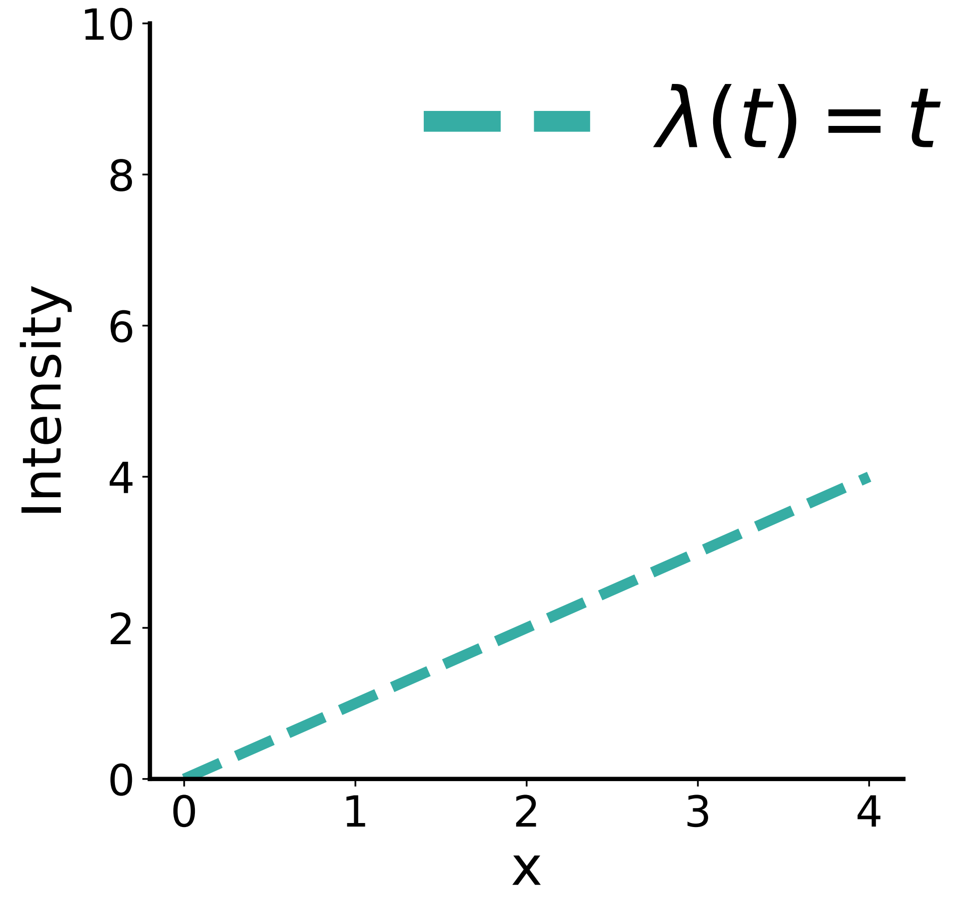
Predictive distribution:

PDF over inter-event times.



Intensity Function:

Instantaneous rate of
event happening at *t.*



$$\lambda(t) = \frac{f(t)}{\mathrm{P(No\ event\ until\ t)}} = \frac{f(t)}{1 - F(t)}$$

$$F(t) = \int_0^t f(s)\, ds$$

Questions:

Which PDF corresponds to each intensity function

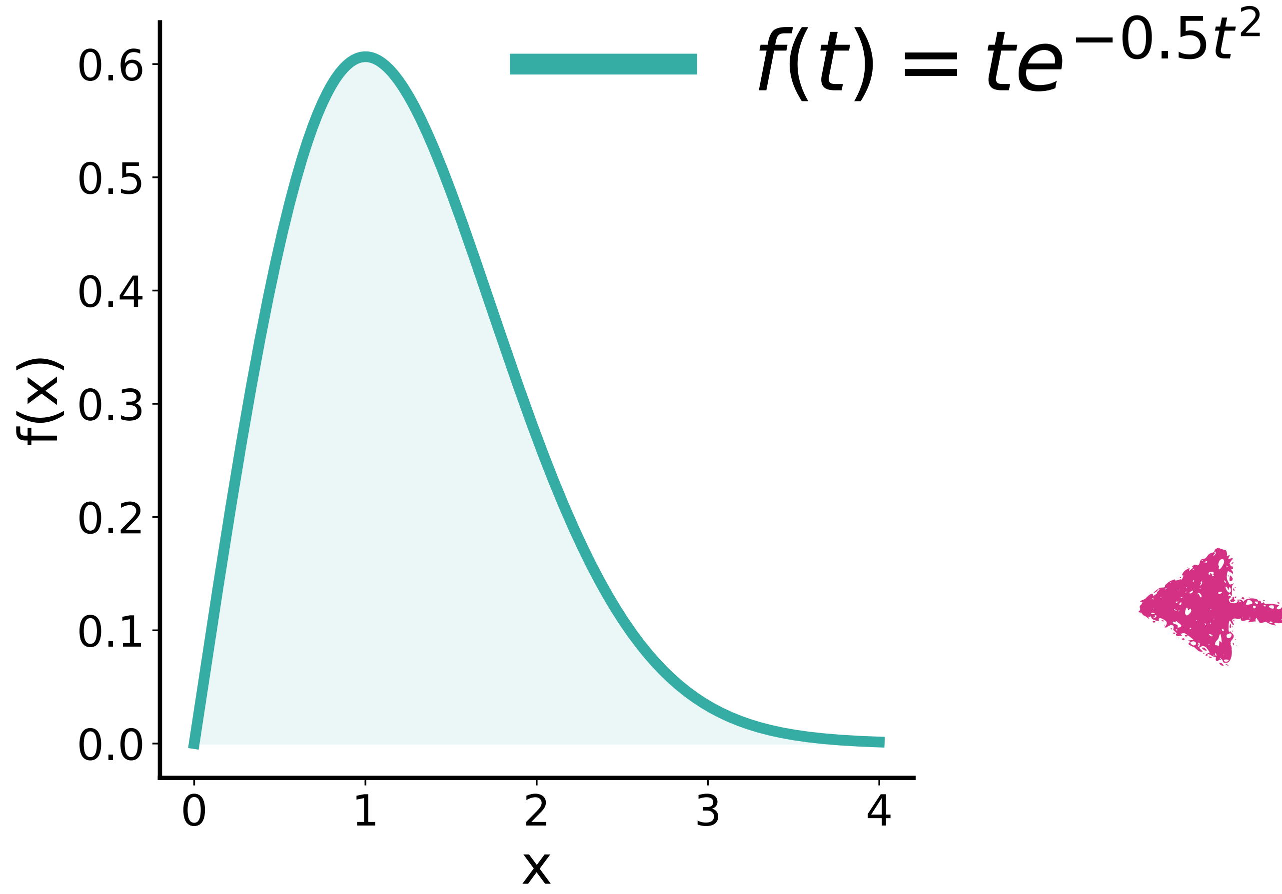
What are the properties of intensity functions?

How can we convert them into each other?

Intensity Function

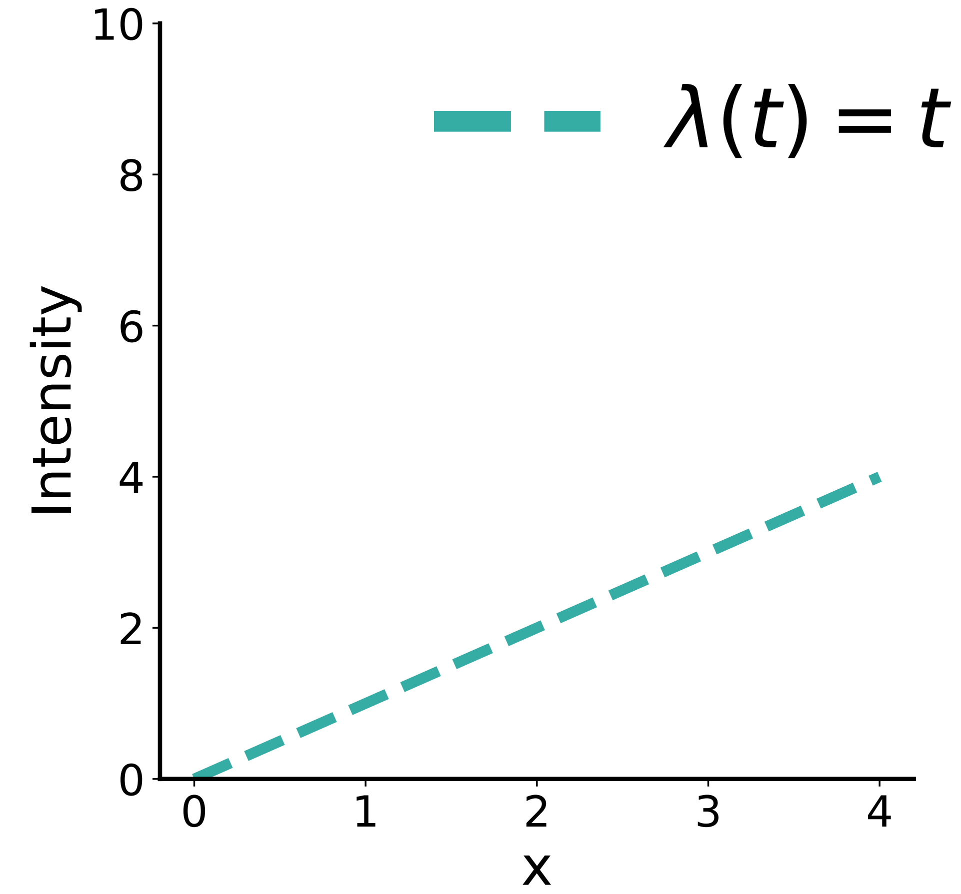
Predictive distribution:

PDF over inter-event times.



Intensity Function:

Instantaneous rate of event happening at *t.*



$$f(t) = \lambda(t) e^{-\int_0^t \lambda(s)\, ds}$$

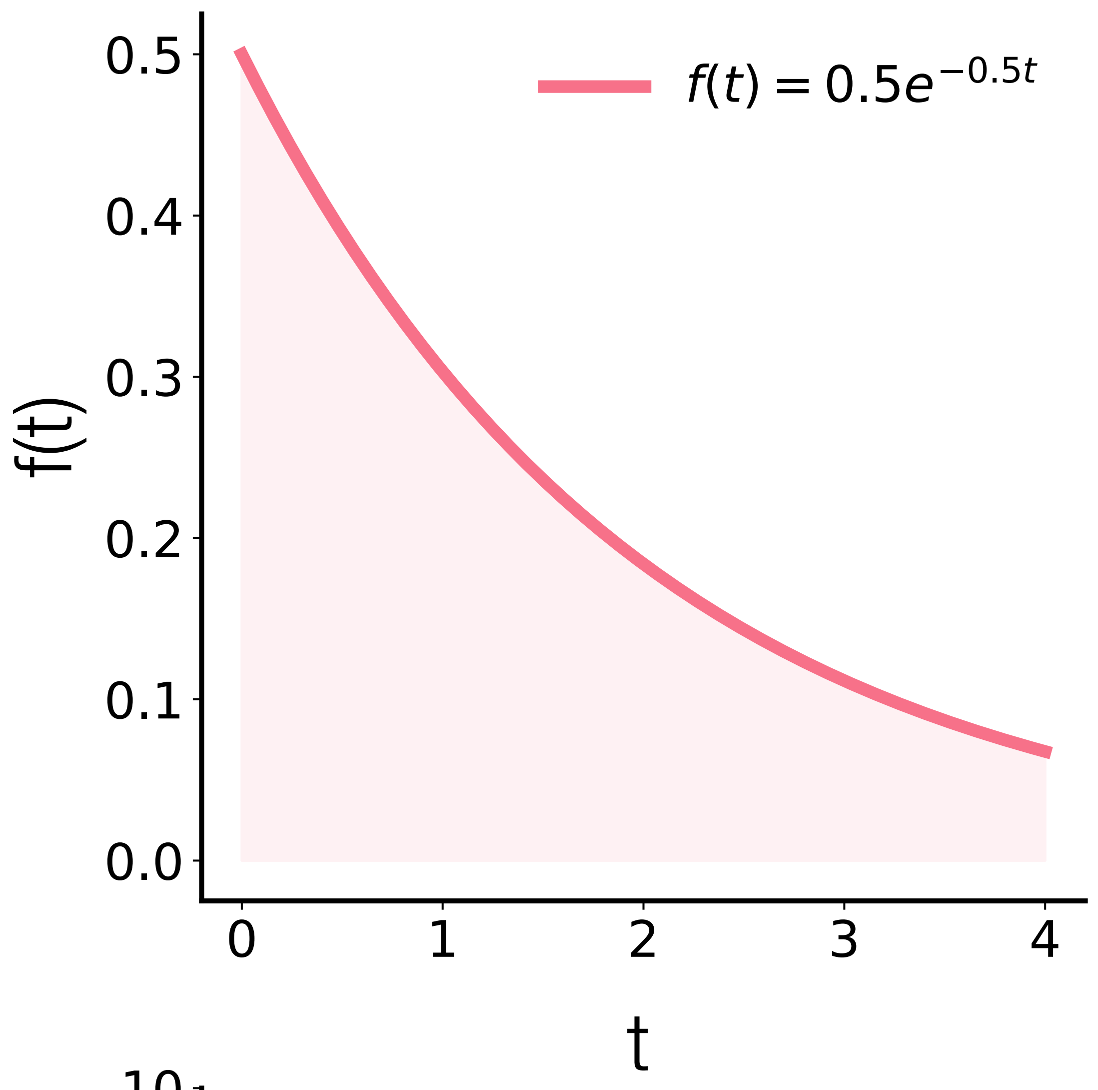
Questions:

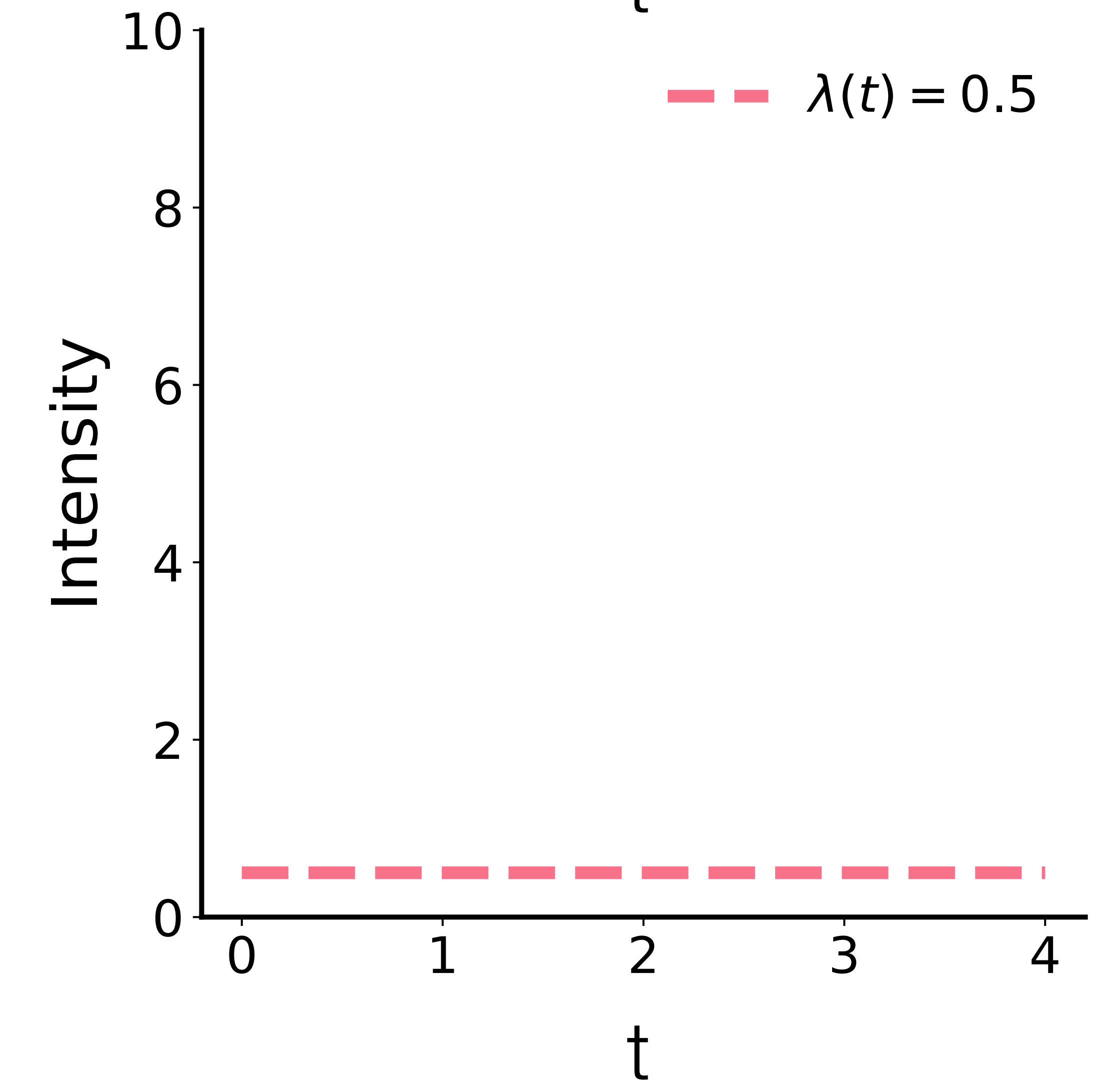
Which PDF corresponds to each intensity function

What are the properties of intensity functions?

How can we convert them into each other?

Intensity Function





$$f(t) = \lambda(t) e^{-\int_0^t \lambda(s)\, ds}$$

$$\lambda(t) = \frac{f(t)}{\text{P(No event until t)}} = \frac{f(t)}{1 - F(t)}$$

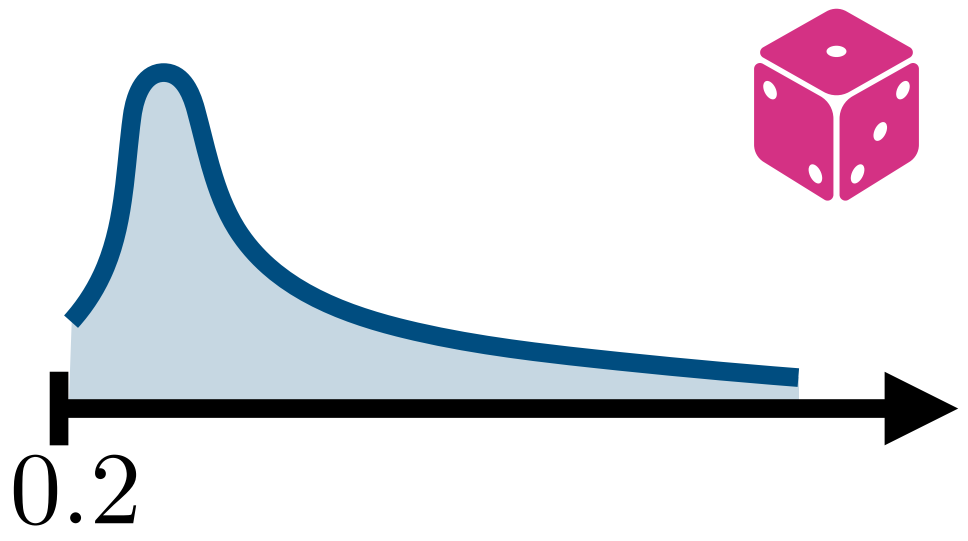
$$F(t) = \int_0^t f(s)\, ds$$

- $e^{-\int_0^t \lambda(s)\, ds}$ is called the *exponential decay*. It is the probability that no event has occurred up to time t.
- $1 - F(t)$ is called the *survival function*. This term represents the probability that the event has not occurred by time t.

Simulating TPPs

$H_{0.2} = [0.1, 0.2]$

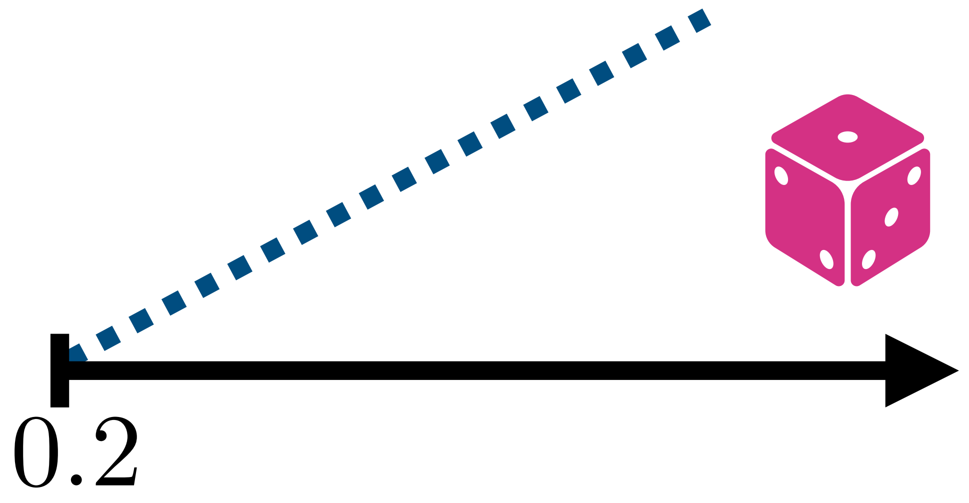
$f^{\mathrm{pred}}(\cdot \mid H_{0.2}) \to$



$\to 1.3$

$H_{0.2} = [0.1, 0.2]$

$\lambda(t, H_{0.2}) \to$

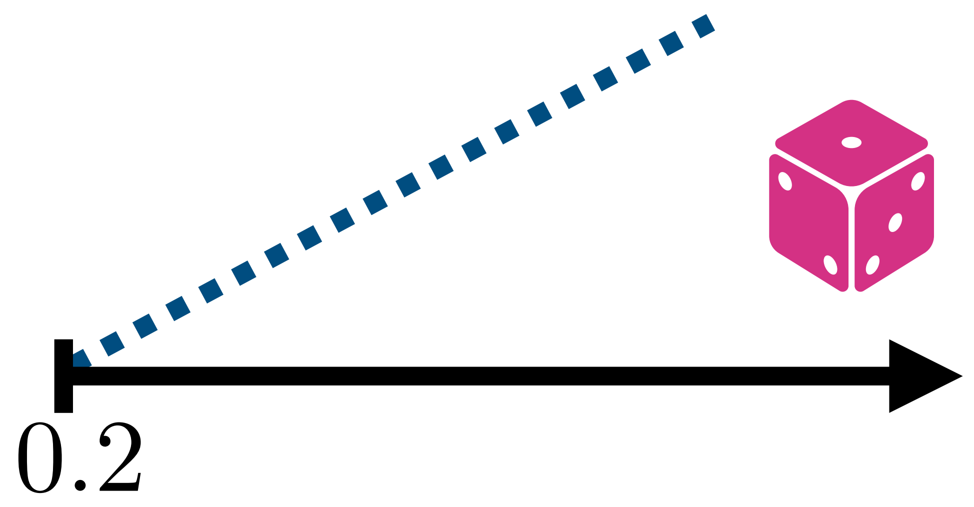


$\to 1.3$

Simulating TPPs

$H_{0.2} = [0.1, 0.2]$

$\lambda(t, H_{0.2}) \rightarrow$



$\rightarrow 1.3$

How do we sample from $\lambda(\cdot \mid H)$?

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{P(\text{Event in } [t, t + \Delta t))}{\Delta t}$$

Simulation Algorithm:

Initialize $H = []$
Until horizon is reached:

- Sample next event time t_i using $\lambda(\cdot \mid H)$ ($t_i >$ max(H))
- Add t_i to H

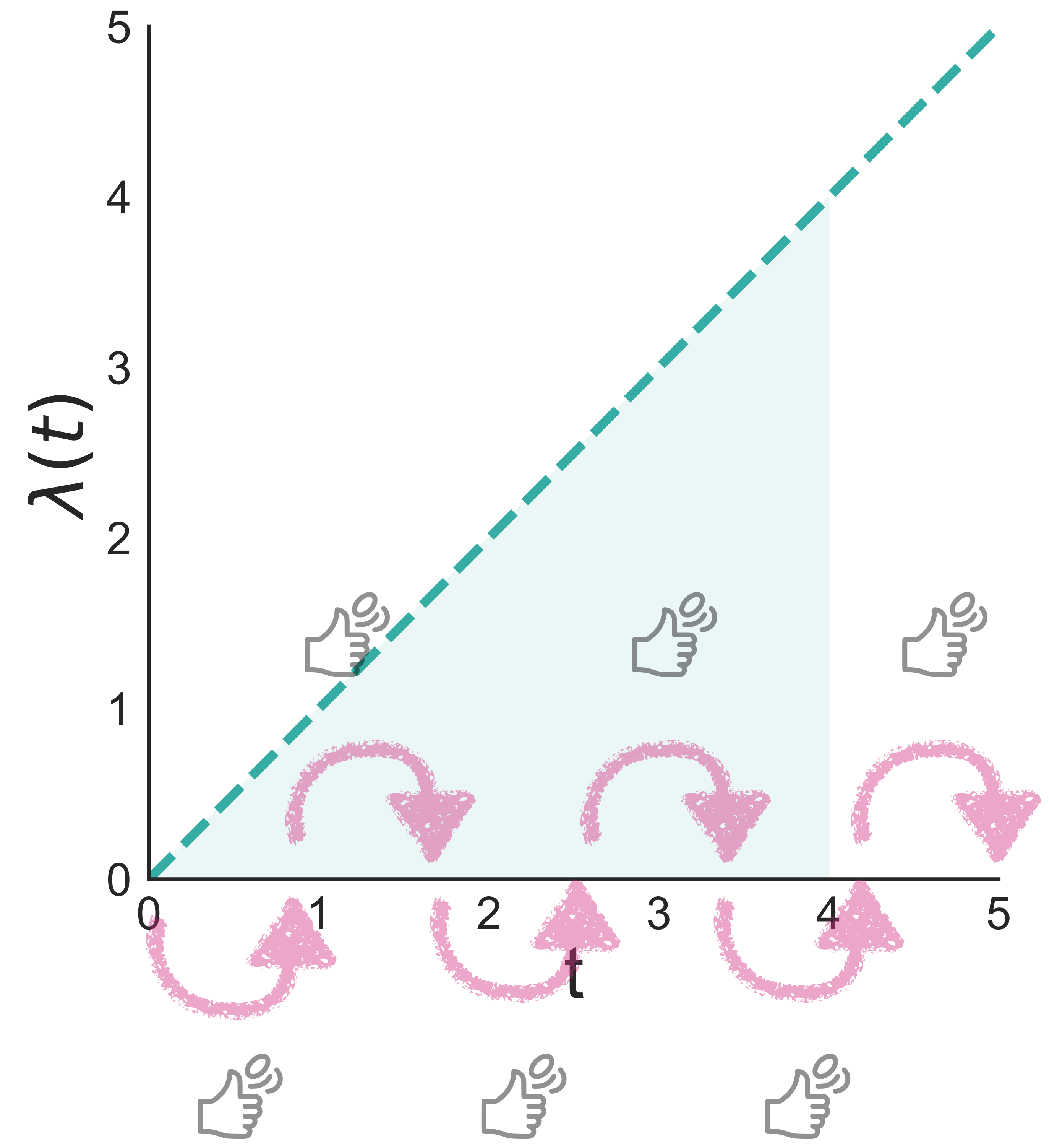
Simulating TPPs

Initialize $H = []$
Until horizon is reached:

- Sample next event time t_i using $\lambda(\cdot \mid H)$ $(t_i > \max(H))$ How do we sample from $\lambda(\cdot \mid H)$?
- Add t_i to H

Method 1:

1. **Discretize Time:** Choose a small time step Δt.
2. **Biased Coin:** At each time step t, use a biased coin to decide whether an event occurs. The probability p_t of the event occurring in each time step is given by $p_t = \lambda(t)\Delta t$.
3. **Simulate Until Event:** Continue this process until an event occurs.



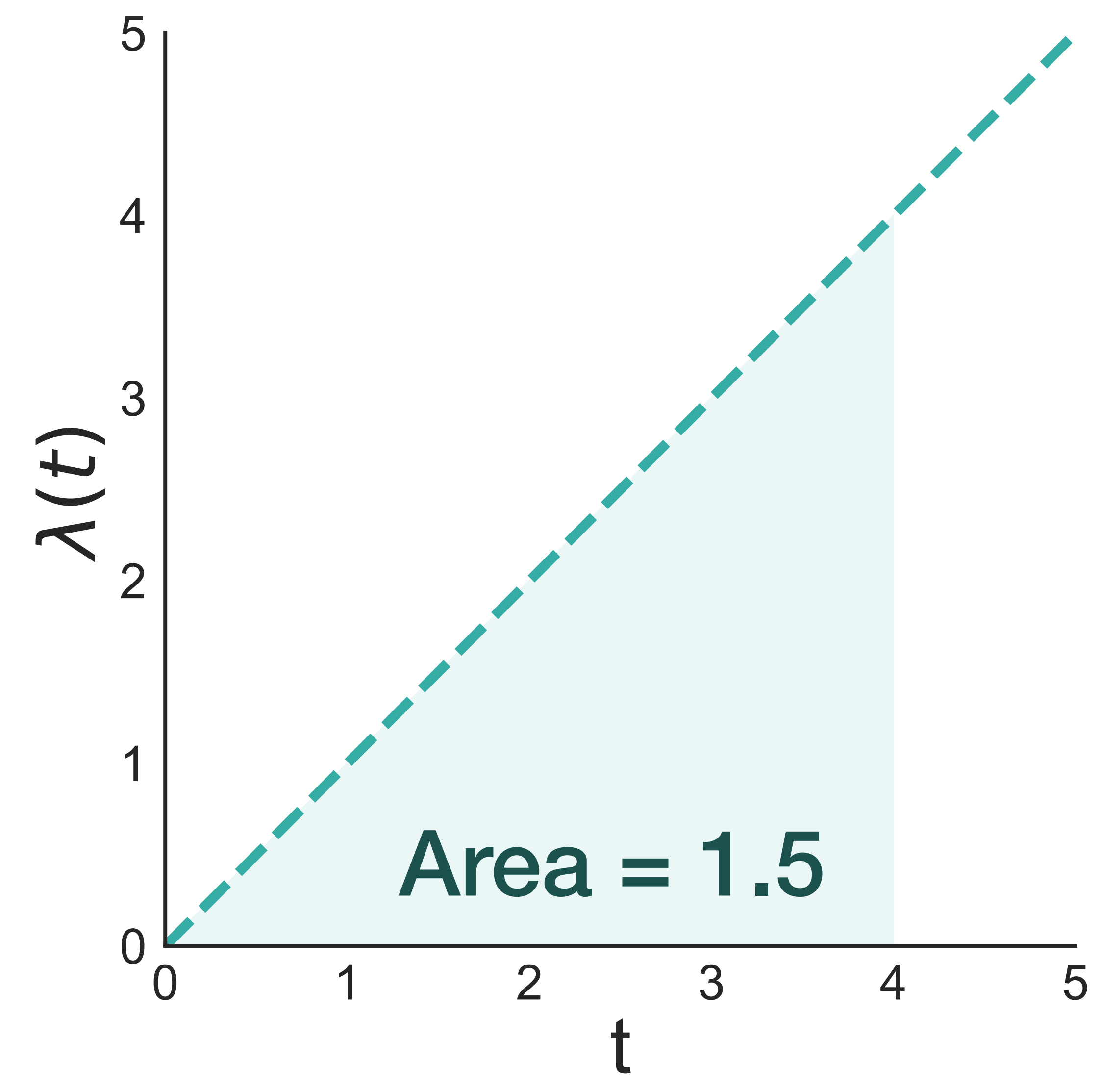
Simulating TPPs

Initialize $H = []$
Until horizon is reached:

- Sample next event time t_i using $\lambda(\cdot \mid H)$ $(t_i > \max(H))$ How do we sample from $\lambda(\cdot \mid H)$?
- Add t_i to H

Method 2:

1. **Discretize Time:** Choose Δt.
2. **Draw a Random Variate:** Draw an exponentially distributed random variate E with $\lambda = 1$.
3. **Numerical Integration:** Integrate $\lambda(t)$ over time until the area under the curve of the intensity equals E. Mathematically, find the smallest T such that:
$$\int_0^T \lambda(t)\, dt = E.$$
4. **Event Occurrence:** The time T at which this condition is first met is the time at which the event occurs.



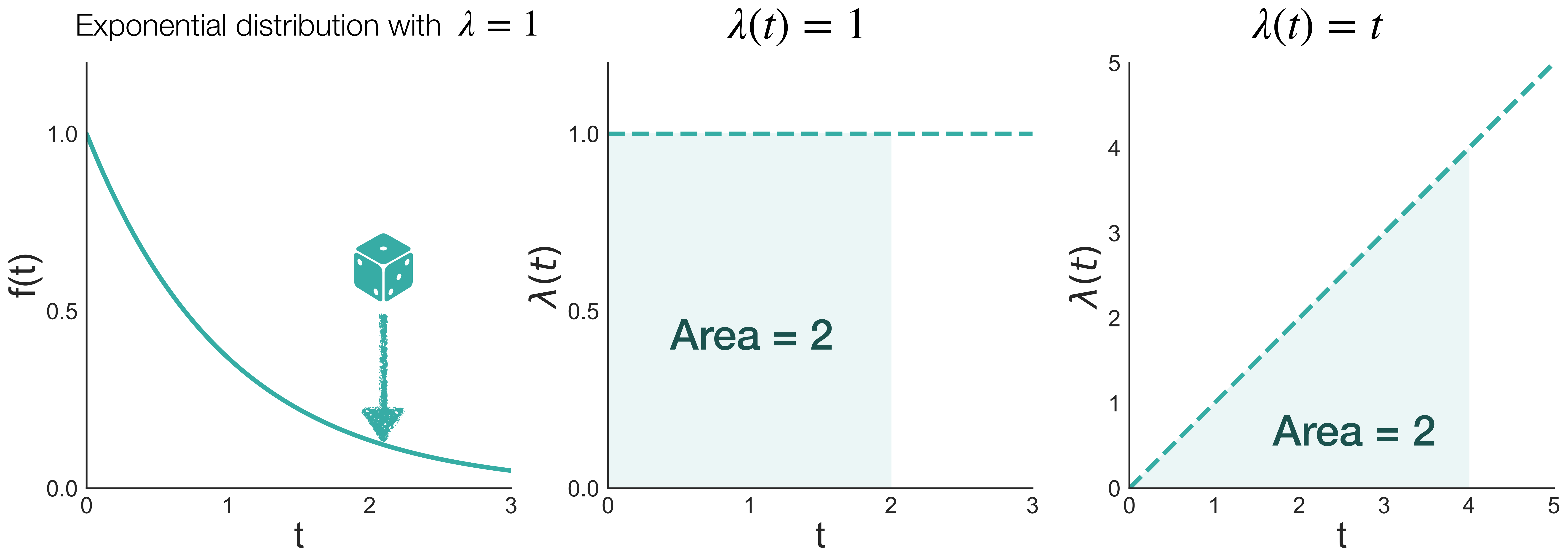
$$E = 1.5 \implies T = 3$$

Simulating TPPs

Initialize $H = []$
Until horizon is reached:

- Sample next event time t_i using $\lambda(\cdot \mid H)$ $(t_i > \max(H))$ How do we sample from $\lambda(\cdot \mid H)$?
- Add t_i to H

Method 2:



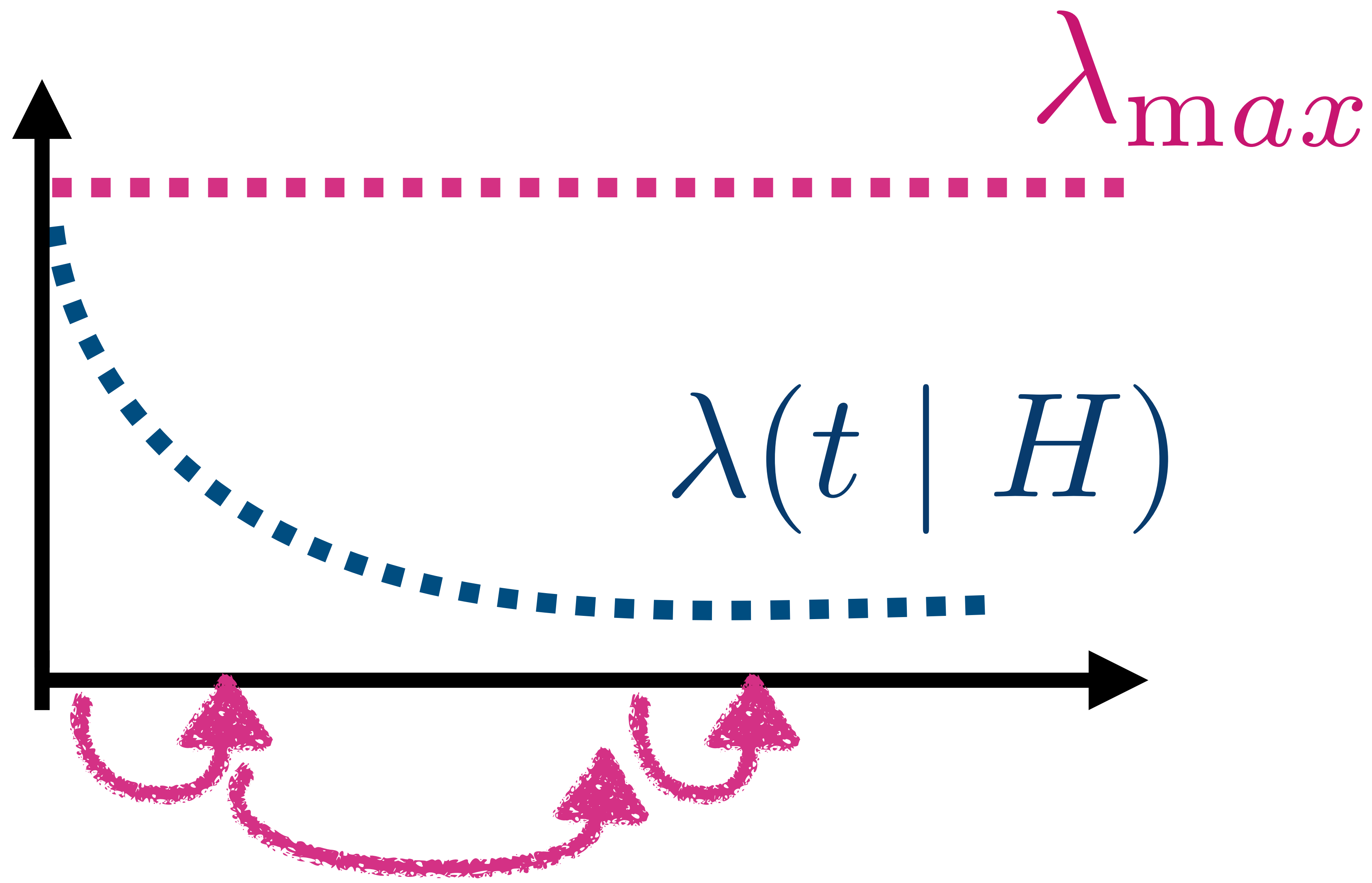
Simulating TPPs

Initialize $H = []$
Until horizon is reached:

- Sample next event time t_i using $\lambda(\cdot \mid H)$ $(t_i > \max(H))$ How do we sample from $\lambda(\cdot \mid H)$?
- Add t_i to H

Method 3:

1. **Determine Maximum Intensity:** Determine an upper bound of the intensity function $\lambda_{\max}$.
2. **Initialize Time:** Set $t = 0$.
3. **Sample Waiting Time:** Draw an exponentially distributed random variate E with rate parameter $\lambda_{\max}$.
4. **Update Time:** Update the time: $t = t + E$.
5. **Acceptance Test:** Generate another uniform random number V in the interval $[0, 1]$. If $V \leq \frac{\lambda(t|H)}{\lambda_{\max}}$, accept the event at time t. Otherwise, reject the event and go back to step 3.



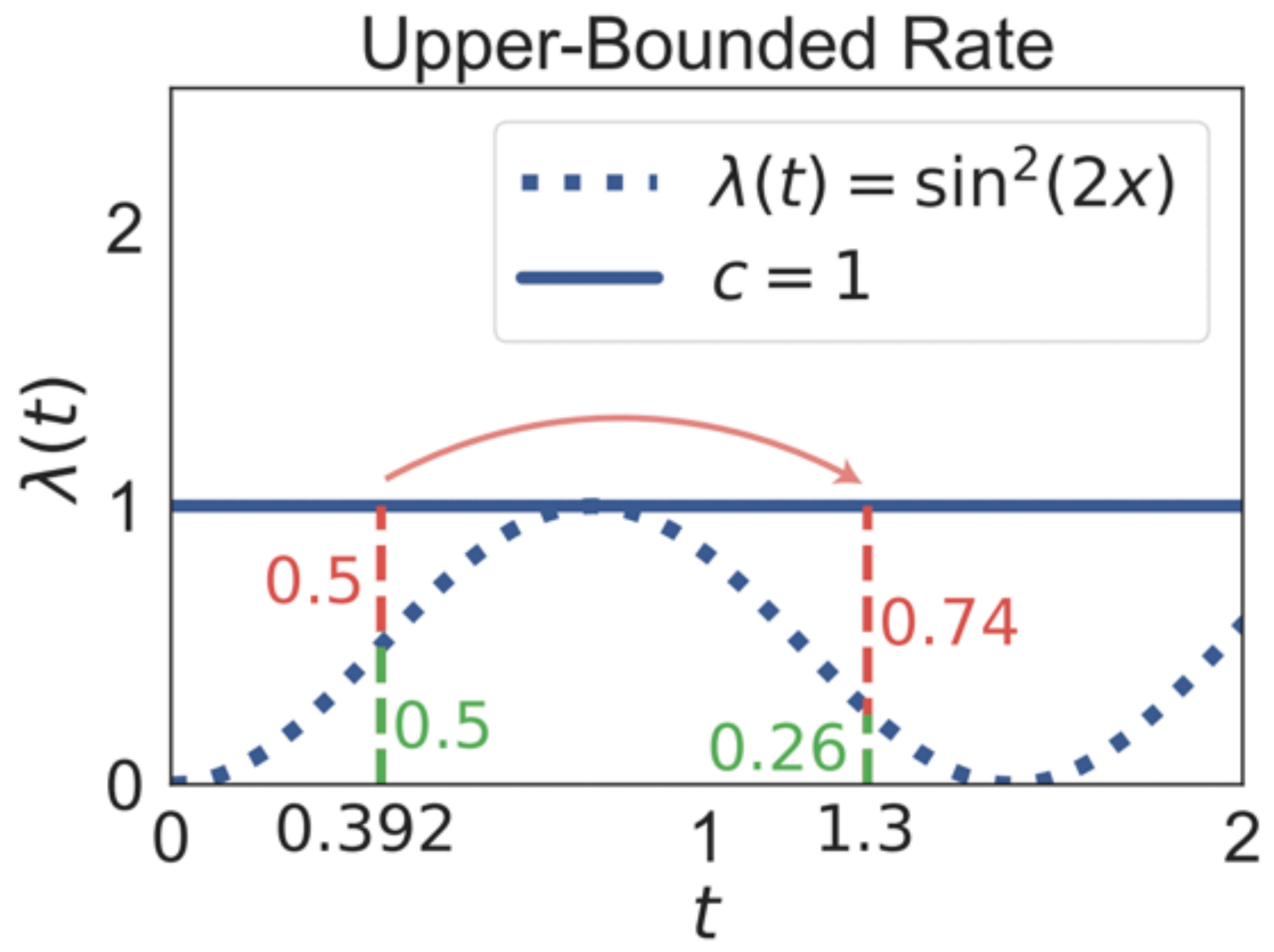
Simulating TPPs

Initialize $H = []$
Until horizon is reached:

- Sample next event time t_i using $\lambda(\cdot \mid H)$ $(t_i > \max(H))$ How do we sample from $\lambda(\cdot \mid H)$?
- Add t_i to H

Method 3:

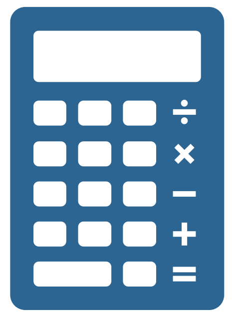
1. **Determine Maximum Intensity:** Determine an upper bound of the intensity function $\lambda_{\max}$.
2. **Initialize Time:** Set $t = 0$.
3. **Sample Waiting Time:** Draw an exponentially distributed random variate E with rate parameter $\lambda_{\max}$.
4. **Update Time:** Update the time: $t = t + E$.
5. **Acceptance Test:** Generate another uniform random number V in the interval $[0, 1]$. If $V \leq \frac{\lambda(t|H)}{\lambda_{\max}}$, accept the event at time t. Otherwise, reject the event and go back to step 3.



Efficient simulation of non-Markovian dynamics on complex networks (Großmann et al.)

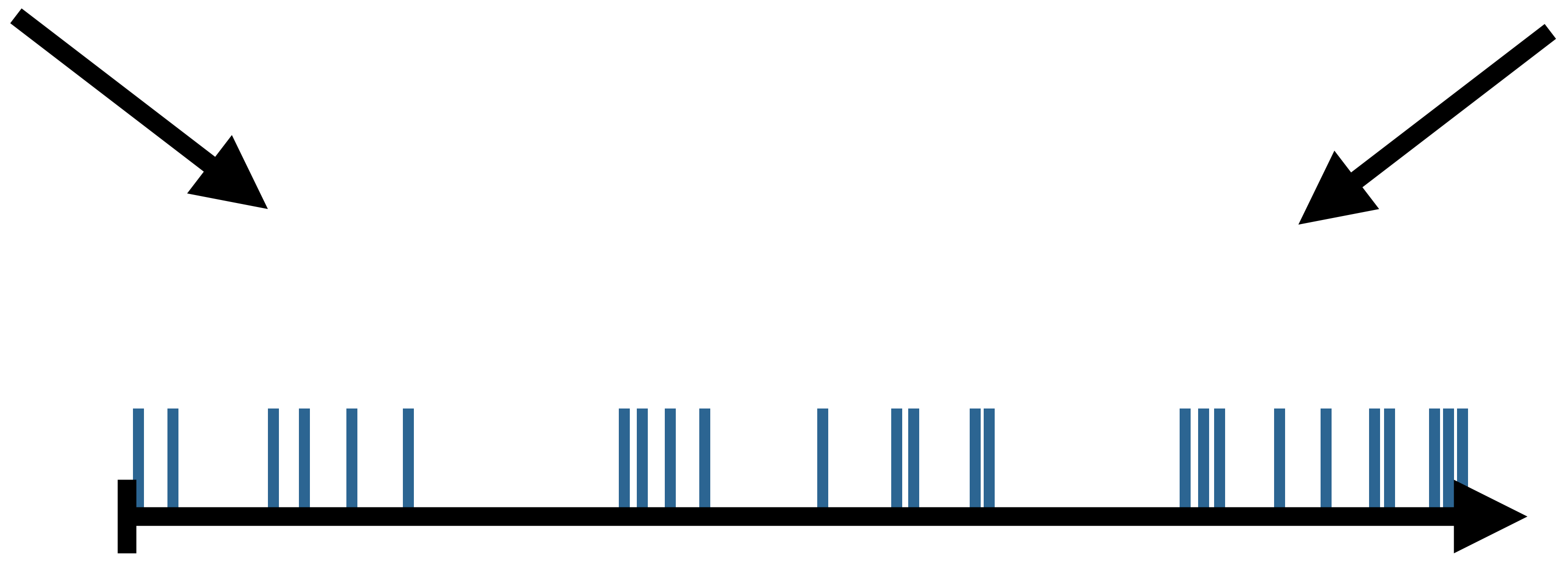
Models of TPPs

How do we specify TPPs?



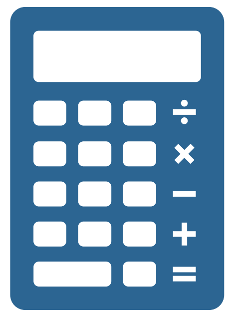
Specify mathematical model (probability measure)

Build simulation algorithm



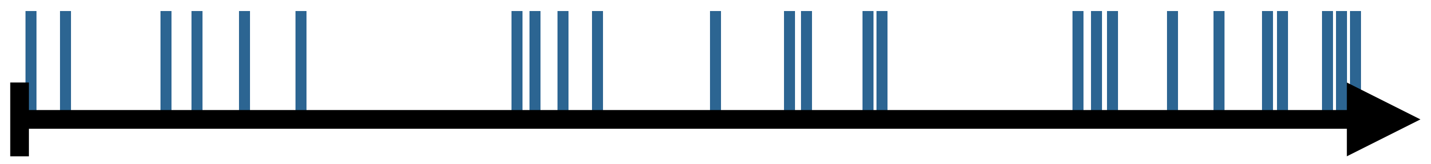
Models of TPPs

How do we specify TPPs?



Specify mathematical model (probability measure)

Build simulation algorithm



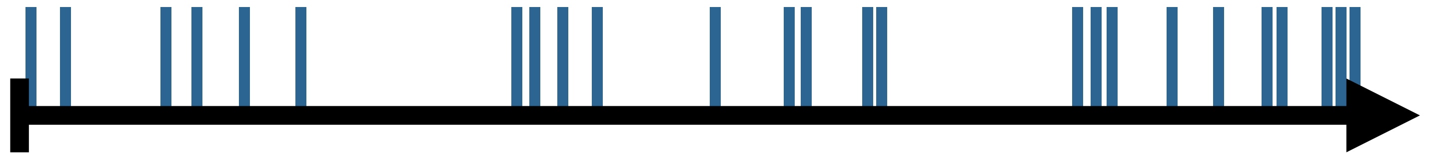
$$f(H) = \left(\prod_{i=1}^{n} f^{\mathrm{pred}}(t_{i+1} \mid H_{t_i}) \right) \left(1 - F^{\mathrm{pred}}(T \mid H_{t_n})\right) \ .$$

Models of TPPs

How do we specify TPPs?

Specify mathematical model (probability measure)

Build simulation algorithm

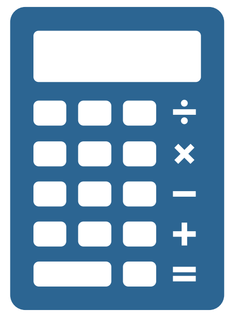


$$f(H) = \left(\prod_{i=1}^{n} f^{\text{pred}}(t_{i+1} \mid H_{t_i}) \right) \left(1 - F^{\text{pred}}(T \mid H_{t_n})\right) \ .$$

1. **Determine Maximum Intensity:** Determine an upper bound of the intensity function $\lambda_{\max}$.
2. **Initialize Time:** Set $t = 0$.
3. **Sample Waiting Time:** Draw an exponentially distributed random variate E with rate parameter $\lambda_{\max}$.
4. **Update Time:** Update the time: $t = t + E$.
5. **Acceptance Test:** Generate another uniform random number V in the interval $[0, 1]$. If $V \leq \frac{\lambda(t|H)}{\lambda_{\max}}$, accept the event at time t. Otherwise, reject the event and go back to step 3.

Models of TPPs

How do we specify TPPs?



Specify mathematical model (probability measure)

Build simulation algorithm

How do we actually construct $\lambda(\cdot \mid H)$?

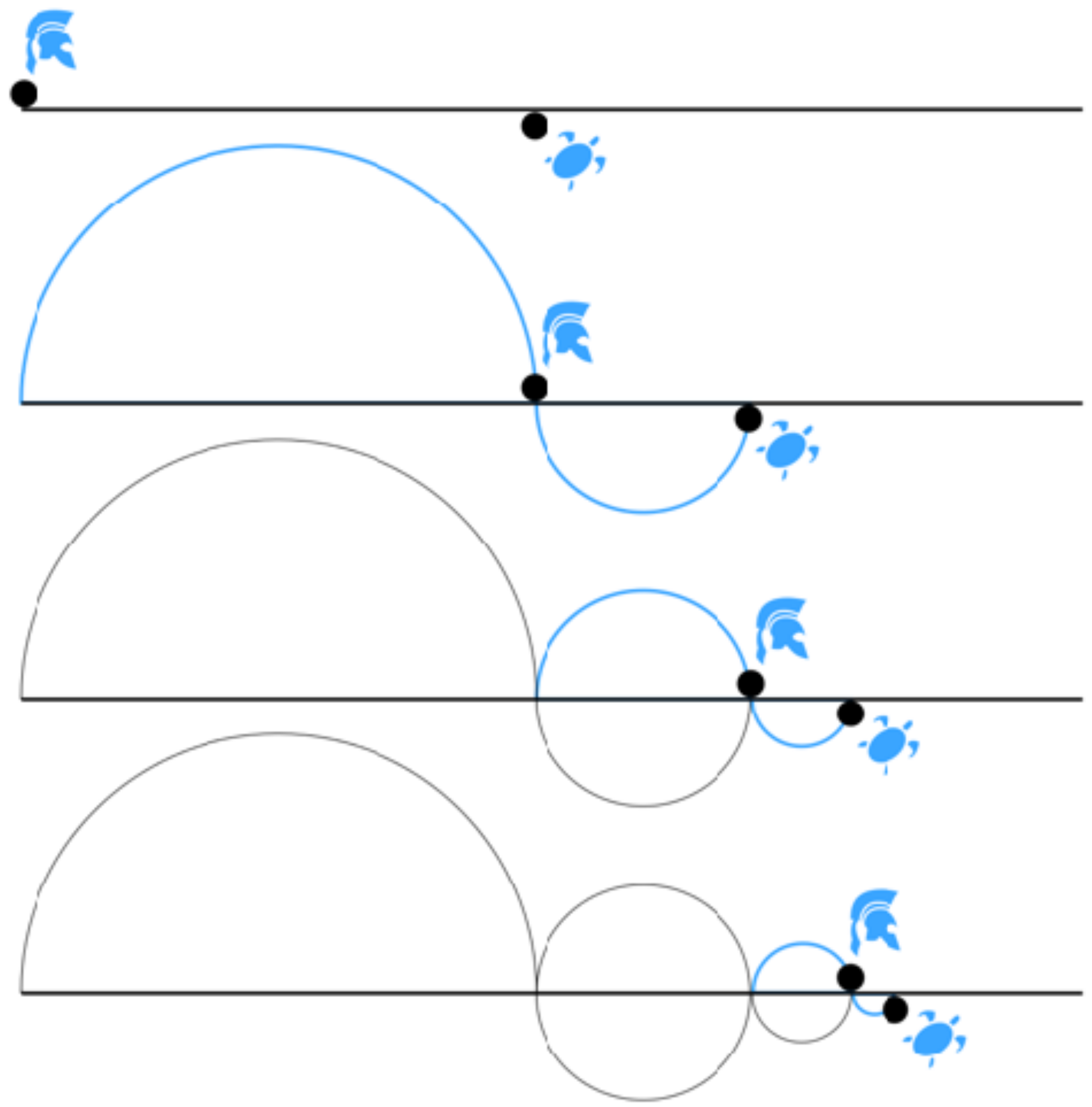
$$f(H) = \left(\prod_{i=1}^{n} f^{\text{pred}}(t_{i+1} \mid H_{t_i})\right)\left(1 - F^{\text{pred}}(T \mid H_{t_n})\right) .$$

1. **Determine Maximum Intensity:** Determine an upper bound of the intensity function $\lambda_{\max}$.
2. **Initialize Time:** Set $t = 0$.
3. **Sample Waiting Time:** Draw an exponentially distributed random variate E with rate parameter $\lambda_{\max}$.
4. **Update Time:** Update the time: $t = t + E$.
5. **Acceptance Test:** Generate another uniform random number V in the interval $[0, 1]$. If $V \leq \frac{\lambda(t|H)}{\lambda_{\max}}$, accept the event at time t. Otherwise, reject the event and go back to step 3.

Explosive Behavior

Can you specify a $\lambda(\cdot \mid H)$ such that the number of events converges to infinity in a finite amount of time?

- If H_t is empty: Converge to infinity within $[0, 0.5]$.
- If H_t is non-empty:
  - Define $t_{\max}$ to be the maximum of H_t (the most recent event).
  - Let the intensity function convert to infinity within $(H_t - t_{\max})/2$.



Zeno behavior (Zeno of Elea) refers to the paradoxical situation where an infinite number of events occur in a finite amount of time.

Hawkes Processes

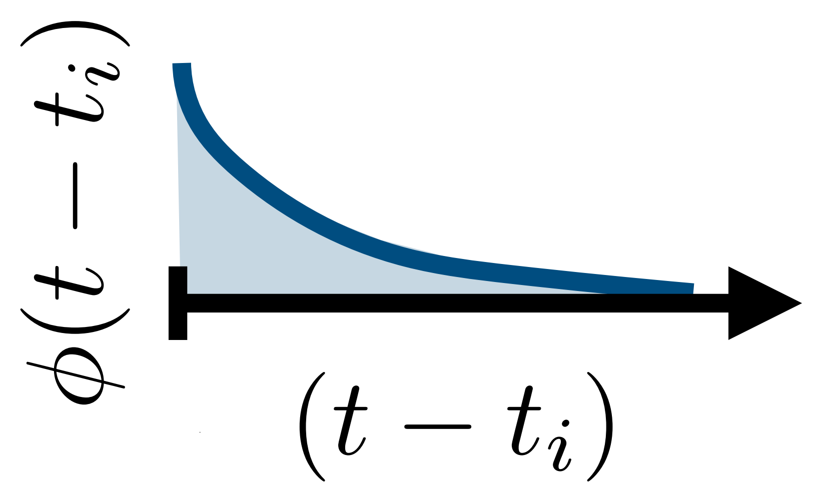
Q: How do we actually construct $\lambda(\,\cdot \mid H)$?

A: To create self-exciting dynamics: Hawkes processes

Sum over all past events H_t

$$\lambda(t, H_t) = \mu + \sum_{i:t_i<t} \phi(t - t_i)$$

Background rate:
responsible for spontaneous events



Kernel:
The impact of each event decreases the further it is in the past.

Hawkes Processes

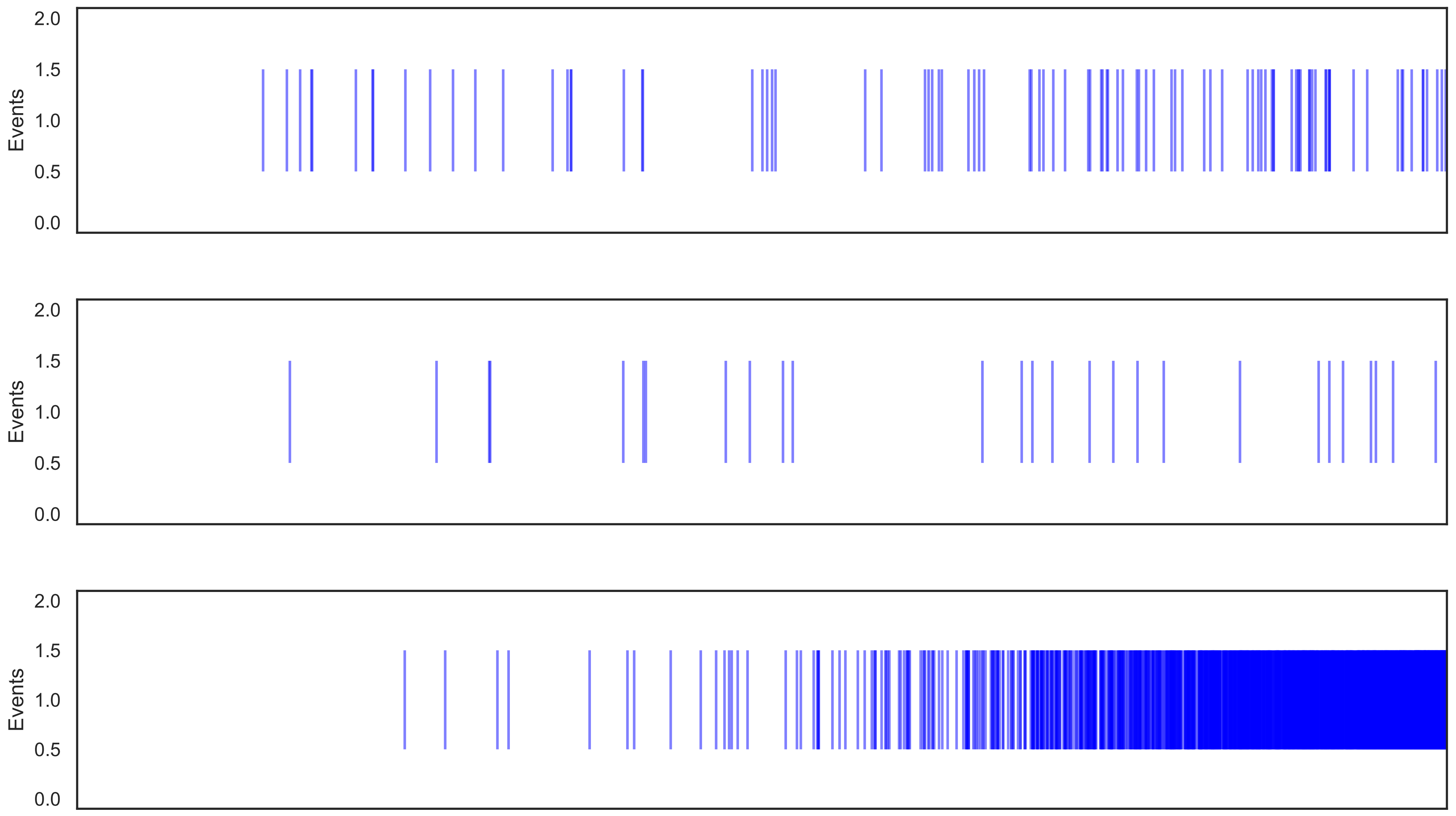
Q: How do we actually construct $\lambda(\cdot \mid H)$?

A: To create self-exciting dynamics: Hawkes processes

$$\lambda(t, H_t) = \mu + \sum_{i:t_i<t} \phi(t - t_i)$$

$$\phi(t) = \alpha e^{-\beta t}$$

```python
def hawkes_intensity(t, events, mu=0.25, alpha=0.7, beta=0.5):
    intensity = mu
    for t_i in events:
        if t_i < t:
            intensity += alpha * np.exp(-beta * (t - t_i))
    return intensity
```



Hawkes Processes

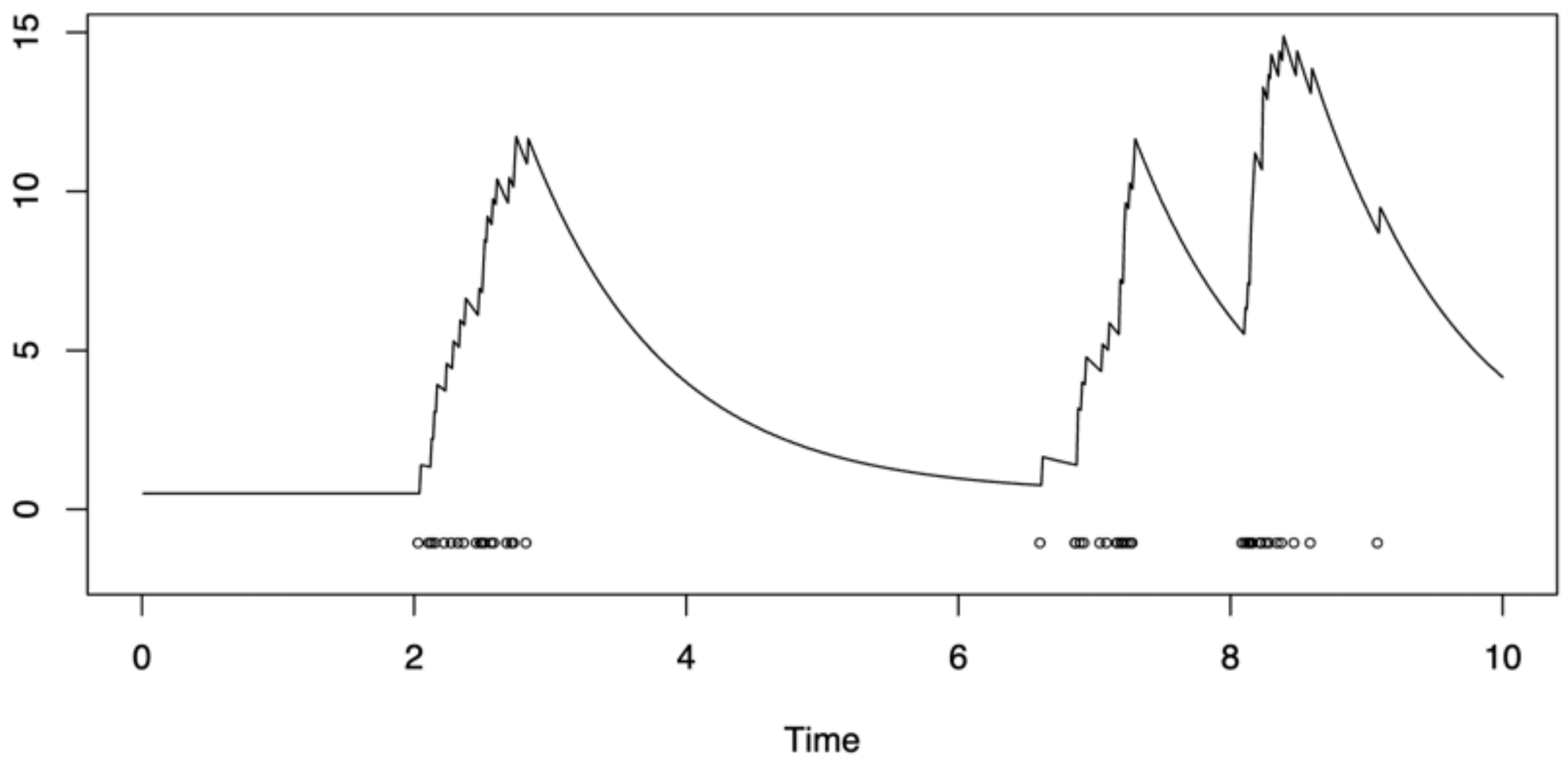
Q: How do we actually construct $\lambda(\cdot \mid H)$?

A: To create self-exciting dynamics: Hawkes processes

$$\lambda(t, H_t) = \mu + \sum_{i:t_i<t} \phi(t - t_i)$$

$$\phi(t) = \alpha e^{-\beta t}$$

```python
def hawkes_intensity(t, events, mu=0.25, alpha=0.7, beta=0.5):
    intensity = mu
    for t_i in events:
        if t_i < t:
            intensity += alpha * np.exp(-beta * (t - t_i))
    return intensity
```



Lecture Notes: Temporal Point Processes and the Conditional Intensity Function (Rasmussen et al.)

Self-Correcting Processes

Self-Exciting

$$\lambda(t, H_t) = \mu + \sum_{i:t_i<t} \phi(t - t_i)$$

Self-Correcting

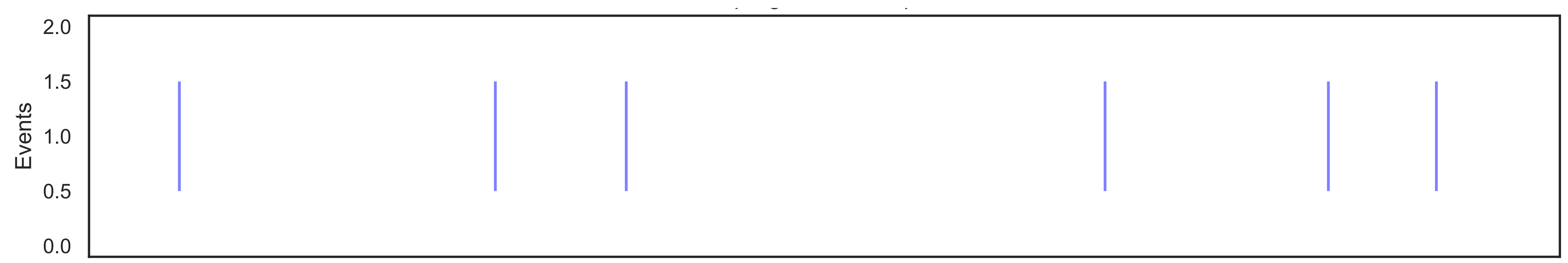
$$\lambda(t, H_t) = \exp\left(\mu t - \sum_{i:t_i<t} \alpha\right)$$

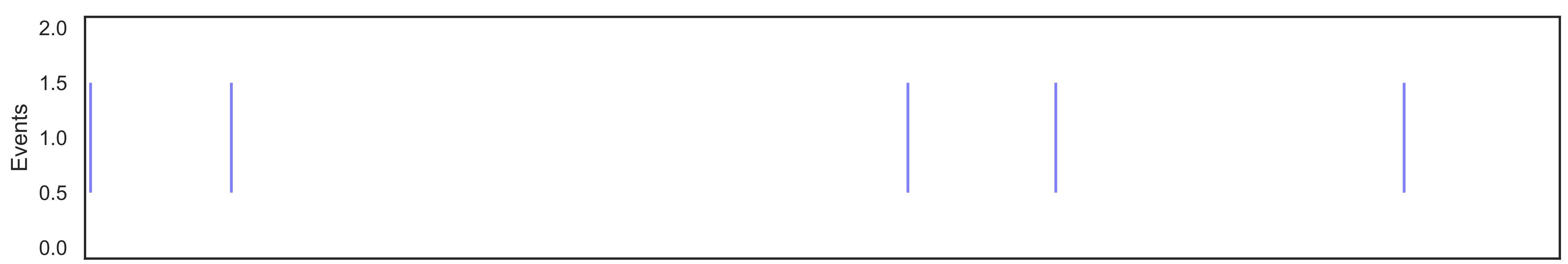
Background rate:

Increases constantly.

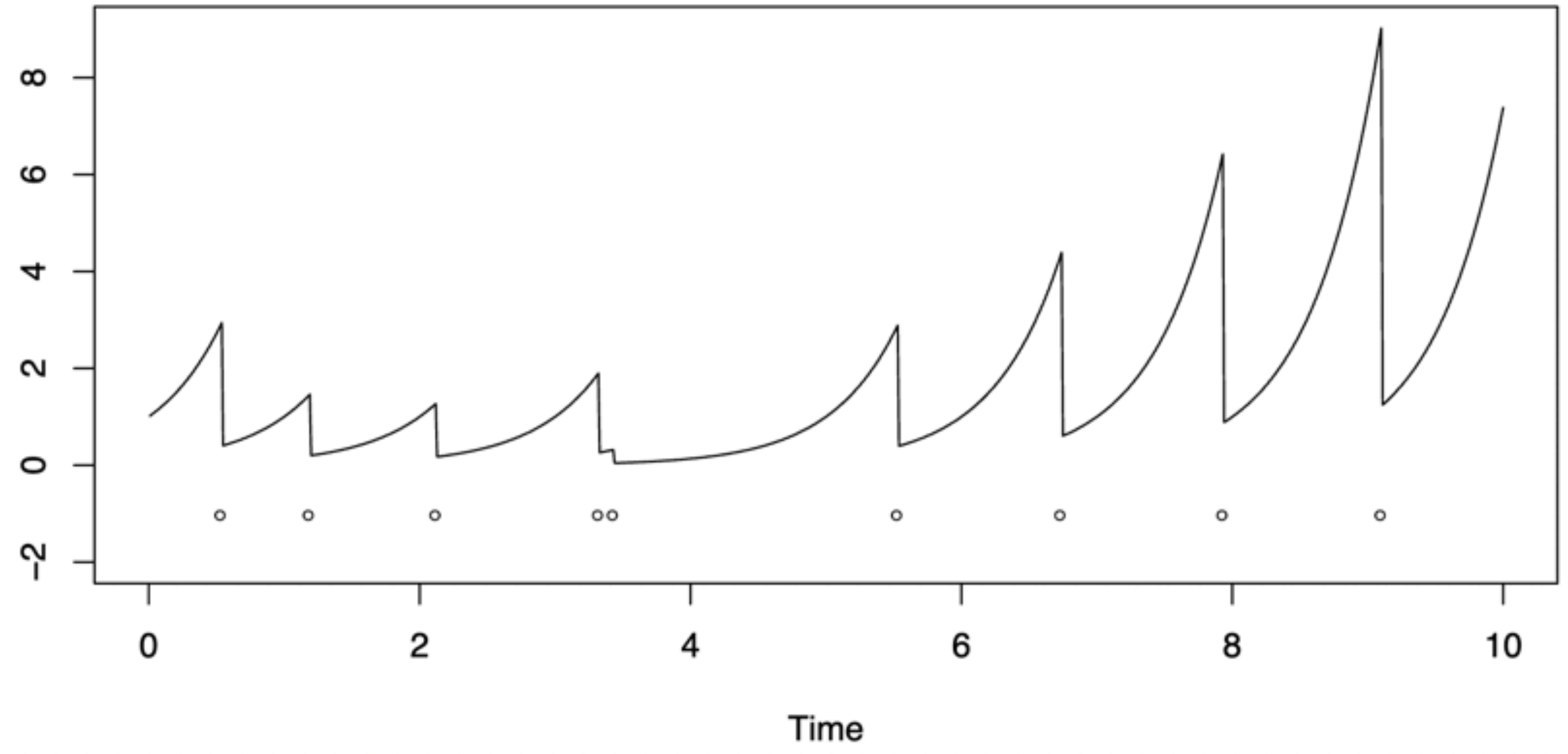
Damping:

Decreases intensity after each event.





Self-Correcting Processes



Lecture Notes: Temporal Point Processes and the Conditional Intensity Function (Rasmussen et al.)

Self-Correcting

$$\lambda(t, H_t) = \exp\left(\mu t - \sum_{i:t_i<t} \alpha\right)$$

Background rate:

Increases constantly.

Damping:

Decreases intensity after each event.

Maximum Likelihood

Self-Exciting

$$\lambda(t, H_t) = \mu + \sum_{i:t_i<t} \phi(t - t_i)$$

Self-Correcting

$$\lambda(t, H_t) = \exp\left(\mu t - \sum_{i:t_i<t} \alpha\right)$$

Assume you have a given event sequence and want to infer $\boldsymbol{\mu, \alpha, \beta,}$ etc.

Closed form solution for homogeneous TPPs: $L(\lambda \mid H) = \lambda^n \exp(-\lambda T) \implies \hat{\lambda} = \frac{n}{T}$

The general likelihood can only be solved numerically: $L(\lambda(\cdot) \mid H) = \left(\prod_{i=1}^{n} \lambda(t_i \mid H_{t_i})\right) \exp\left(-\int_0^T \lambda(t \mid H_t)\, dt\right)$

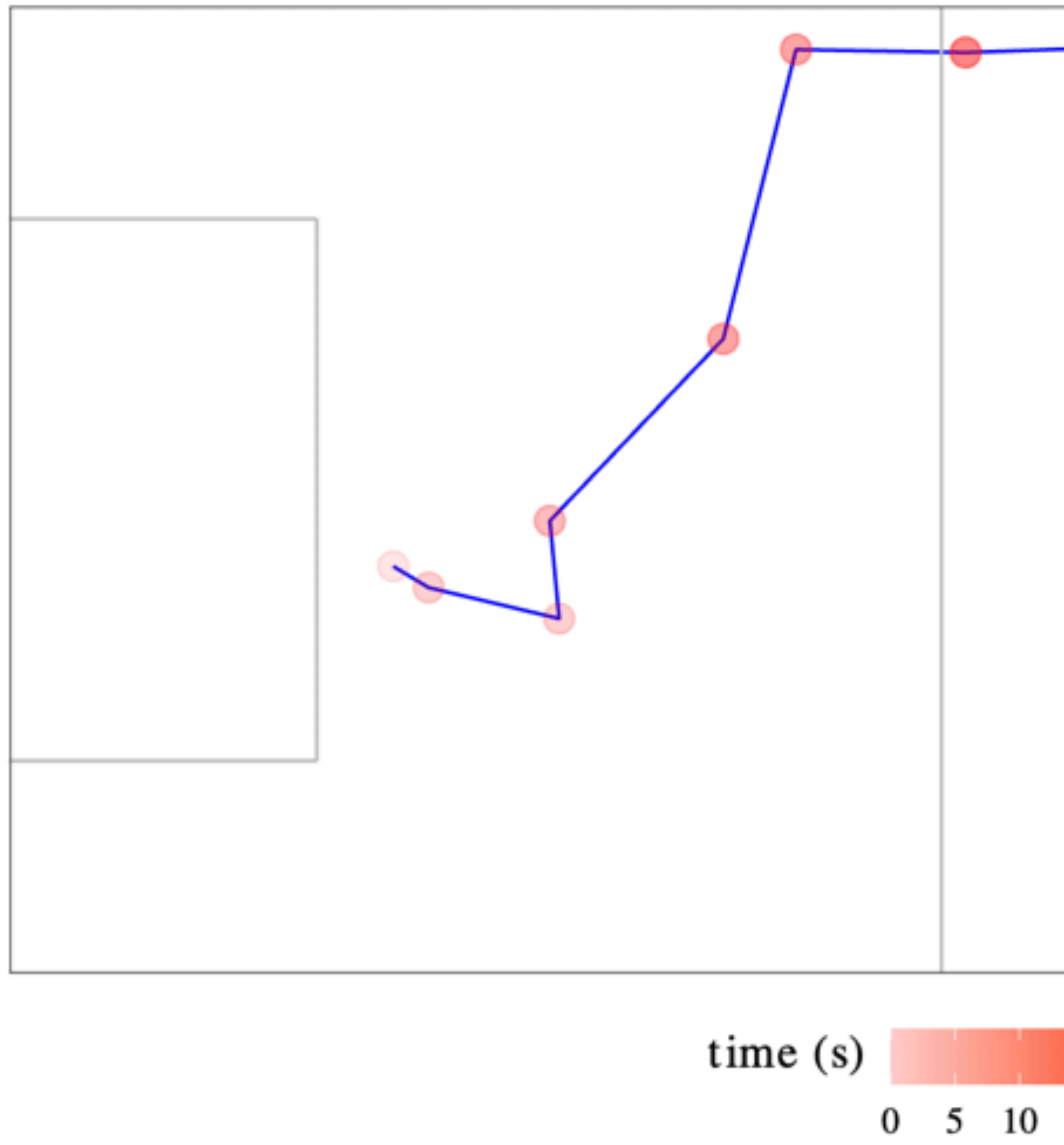
Part II

MARKED TEMPORAL POINT PROCESSES

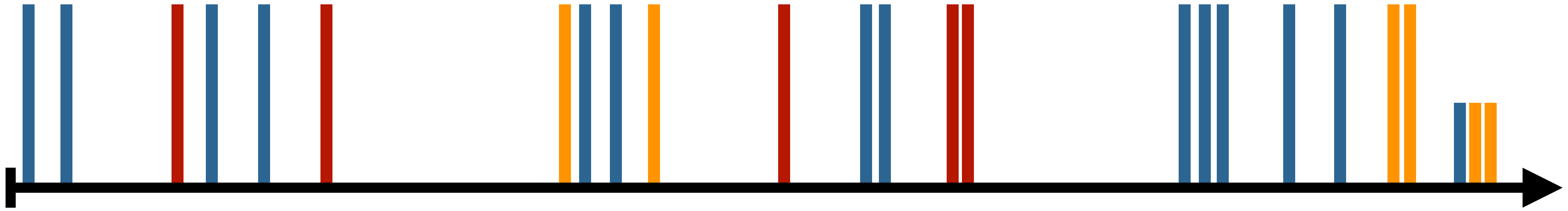
event type	frequency	event
Pass	376924	SavedS
BallRecovery	36908	Save
Clearance	25462	Corne
Tackle	14581	Missec
TakeOn	13607	Offsid
BallTouch	13517	Claim
Aerial	12871	Goal
Interception	10422	Punch
Dispossessed	8897	ShotO
Foul	8238	Smoth
KeeperPickup	5208	CrossN

Marked TPPs

- Each event time t_i is associated with a mark $m_i \in \mathcal{M}$.
- Event sequence: $H = [(t_0, m_0), (t_1, m_1), \ldots, (t_n, m_n)]$.
- Intensity functions: $\{\lambda_m(t, H_t)\}_{m \in \mathcal{M}}$, where $\lambda_m(t, H_t)$ is the intensity for events with mark m, given history H_t.
- All intensity functions run in parallel to generate a sequence.



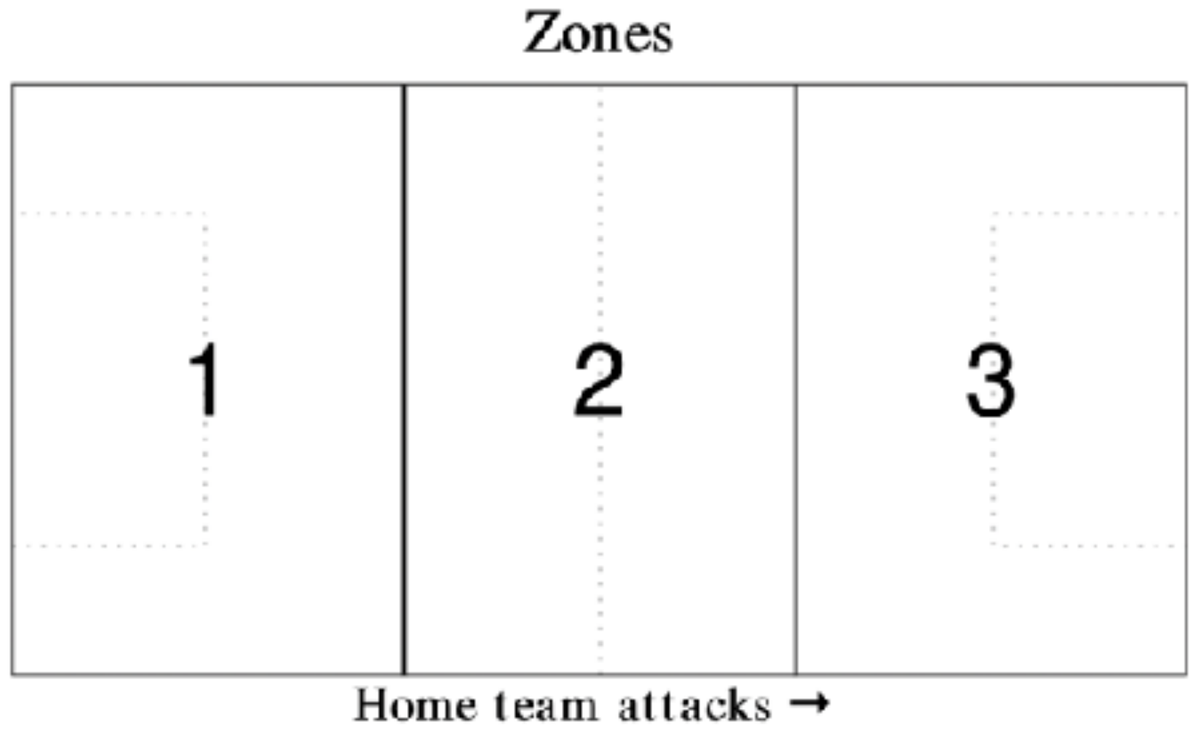
Three marks: blue, red, yellow:

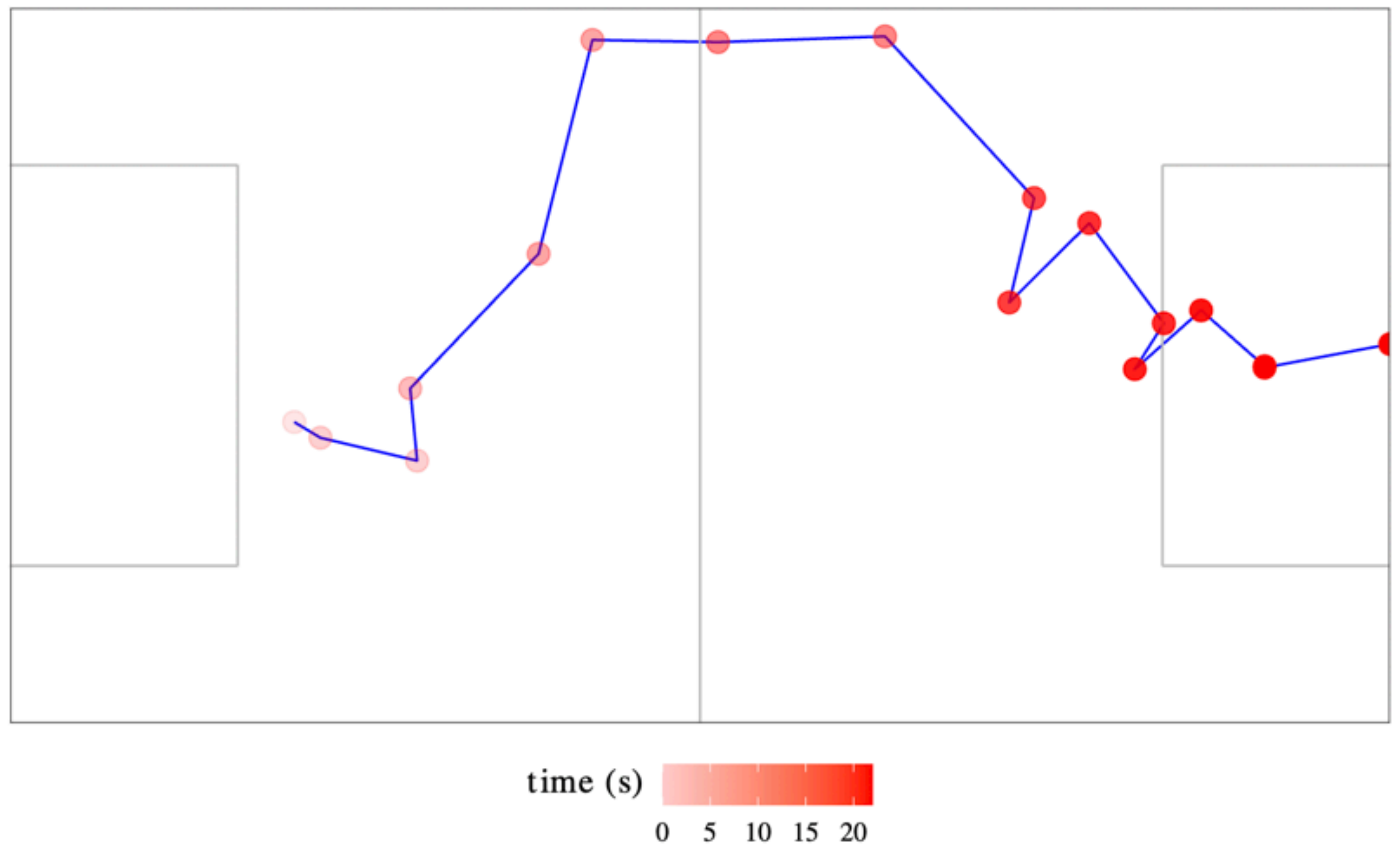


Marked TPPs

Table 2: Frequencies of the 22 distinct types of touch-ball events in the data.

event type	frequency	event type	frequency
Pass	376924	SavedShot	4971
BallRecovery	36908	Save	4910
Clearance	25462	CornerAwarded	4100
Tackle	14581	MissedShots	4076
TakeOn	13607	OffsidePass	1582
BallTouch	13517	Claim	1181
Aerial	12871	Goal	1052
Interception	10422	Punch	380
Dispossessed	8897	ShotOnPost	187
Foul	8238	Smother	122
KeeperPickup	5208	CrossNotClaimed	81

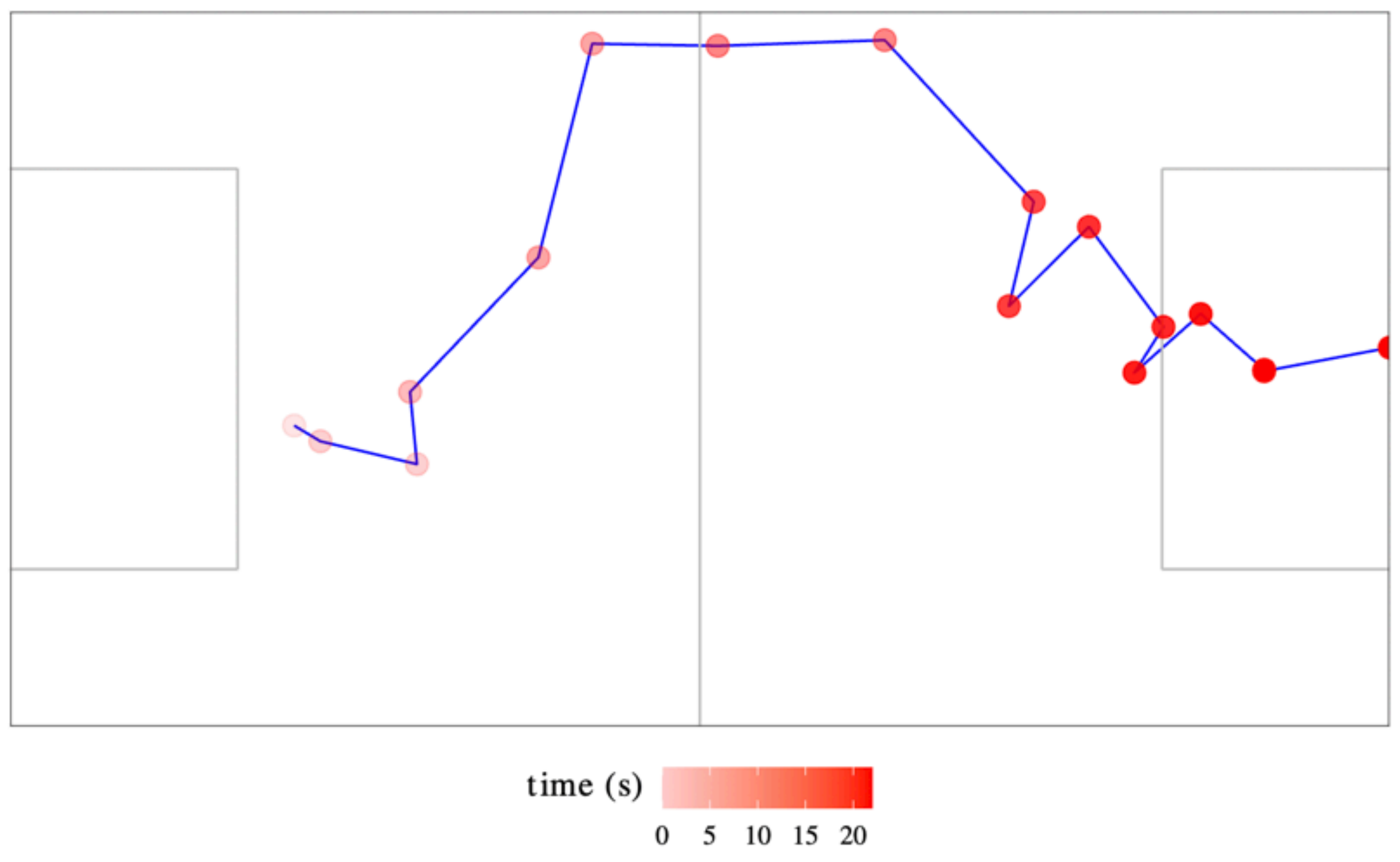


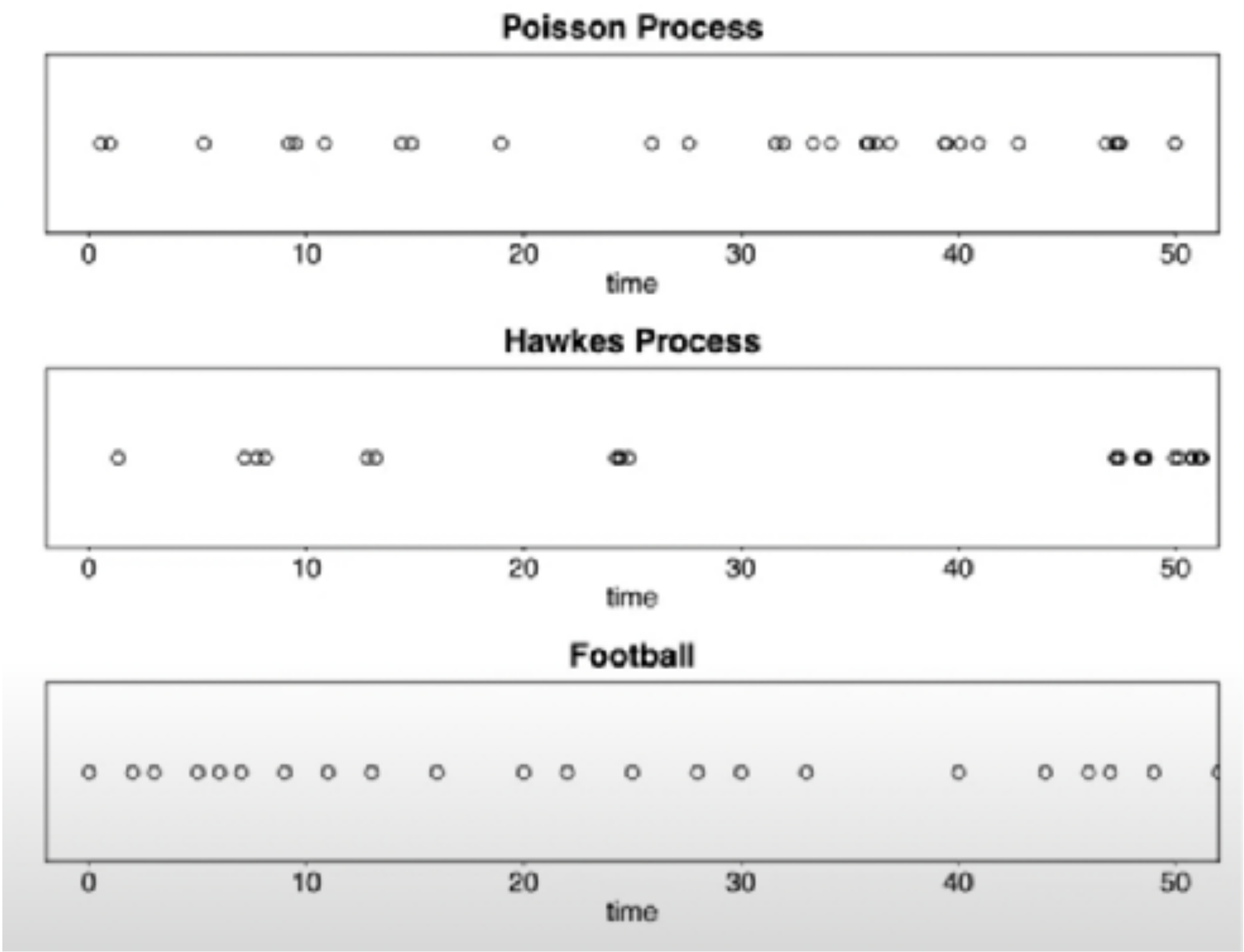


i	id	period	team_id	time (t_i)	zone (z_i)	mark (m_i)
1	101	1	1	0	2	18
2	101	1	1	1	2	19
3	101	1	2	3	1	8
4	101	1	1	6	3	16
5	101	1	1	8	3	18
6	101	1	1	15	2	18
7	101	1	1	16	1	19
8	101	1	2	19	1	12

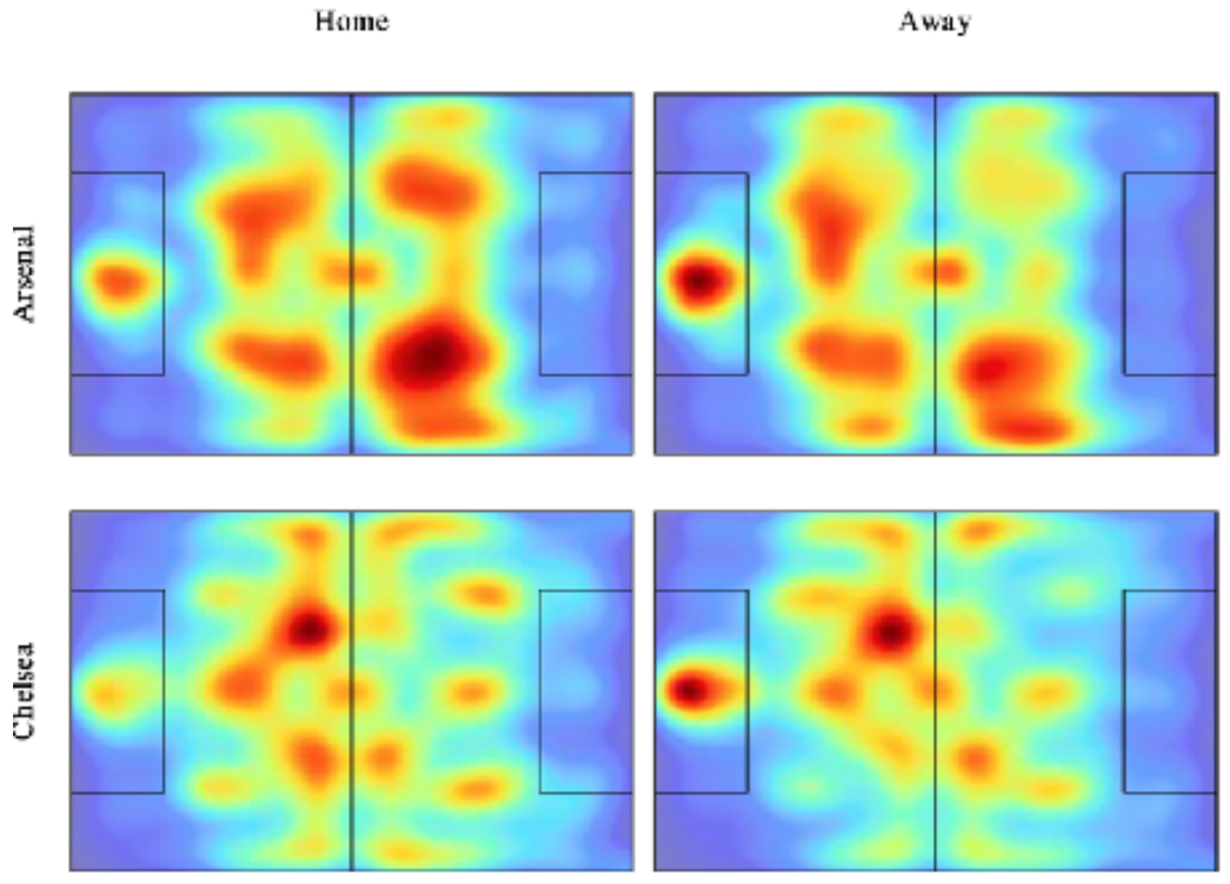
Flexible marked spatio-temporal point processes with applications to event sequences from association football (Narayanan)

Marked TPPs





youtube.com/watch?v=TJTki8Ym5d0&t=1639s&ab_channel=RoyalStatSoc



Density of ball-touches is shifted to the right in their home games

- identify player and team strength
- predict event probabilities
- contribution of background process
- identify home advantage
- explore startegies

Flexible marked spatio-temporal point processes with applications to event sequences from association football (Narayanan)

Part III

NEURAL TEMPORAL POINT PROCESSES

Neural Temporal Point Processes

Self-Exciting

$$\lambda(t, H_t) = \mu + \sum_{i:t_i<t} \phi(t - t_i)$$

Neural

?

Statistical models of TPPs only capture basic patterns. Neural network models can learn **arbitrary dynamics**.

Self-Correcting

$$\lambda(t, H_t) = \exp\left(\mu t - \sum_{i:t_i<t} \alpha\right)$$

Model **intensity** function with neural network:

$$\lambda(t, H_t) = \mathrm{NN}(t, H_t)$$

- Numerical integration is expensive
- NN architecture needs to process event sequence of arbitrary length.

Model **predictive distribution** with neural network:

$$f^{\mathrm{pred}}(t, H_t) = \mathrm{NN}(t, H_t)$$

- Predict valid PDF
- PDF representation must support both efficient sampling and evaluation.
- NN architecture needs to process event sequence of arbitrary length.

Neural Temporal Point Processes

Self-Exciting

$$\lambda(t, H_t) = \mu + \sum_{i:t_i<t} \phi(t - t_i)$$

Neural

?

Statistical models of TPPs only capture basic patterns. Neural network models can learn **arbitrary dynamics**.

Self-Correcting

$$\lambda(t, H_t) = \exp\left(\mu t - \sum_{i:t_i<t} \alpha\right)$$

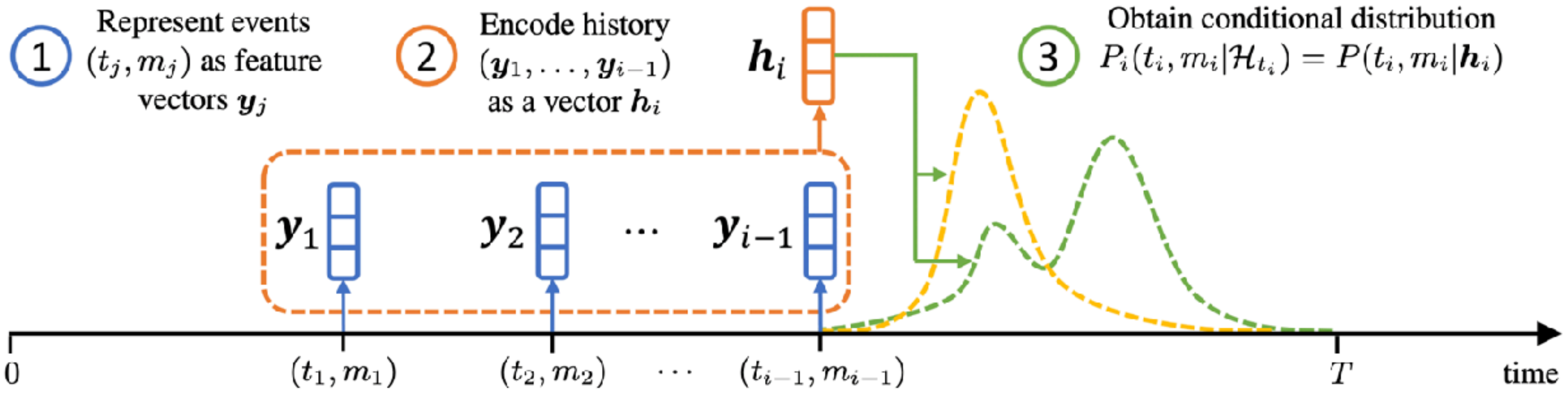
Model **intensity** function with neural network: $\lambda(t, H_t) = \mathrm{NN}(t, H_t)$

- Numerical integration is expensive
- NN architecture needs to process event sequence of arbitrary length.

Model **predictive distribution** with neural network: $f^{\text{pred}}(t, H_t) = \mathrm{NN}(t, H_t)$

- Predict valid PDF
- PDF representation must support both efficient sampling and evaluation.
- NN architecture needs to process event sequence of arbitrary length.

Neural Temporal Point Processes



Neural Temporal Point Processes: A Review (Shchur et al.)

1. Use **MLP** to map event (t_i, m_i) to feature.
2. Use **RNN** to aggregate all event features into single history feature h_i .
3. Use **MLP** to map h_i to parameters θ of a parametric PDF $f_\theta(t)$.
    - Predict λ, k of a Weibull distribution
    - Predict α, d, p of a generalized gamma distribution

Neural Temporal Point Processes

Self-Exciting

$$\lambda(t, H_t) = \mu + \sum_{i:t_i<t} \phi(t - t_i)$$

Neural

?

Statistical models of TPPs only capture basic patterns. Neural network models can learn **arbitrary dynamics**.

Self-Correcting

$$\lambda(t, H_t) = \exp\left(\mu t - \sum_{i:t_i<t} \alpha\right)$$

Model **intensity** function with neural network:

$$\lambda(t, H_t) = \mathrm{NN}(t, H_t)$$

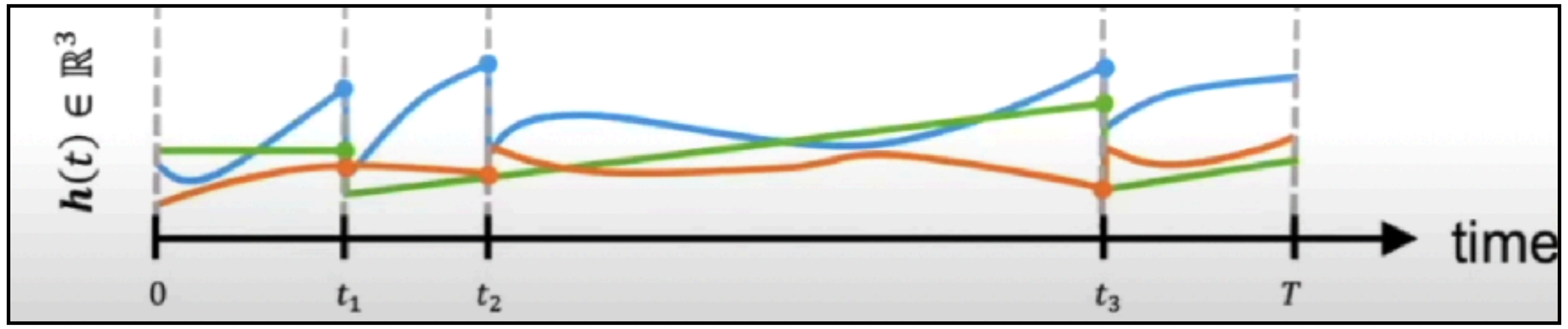
- Numerical integration is expensive
- NN architecture needs to process event sequence of arbitrary length.

Model **predictive distribution** with neural network:

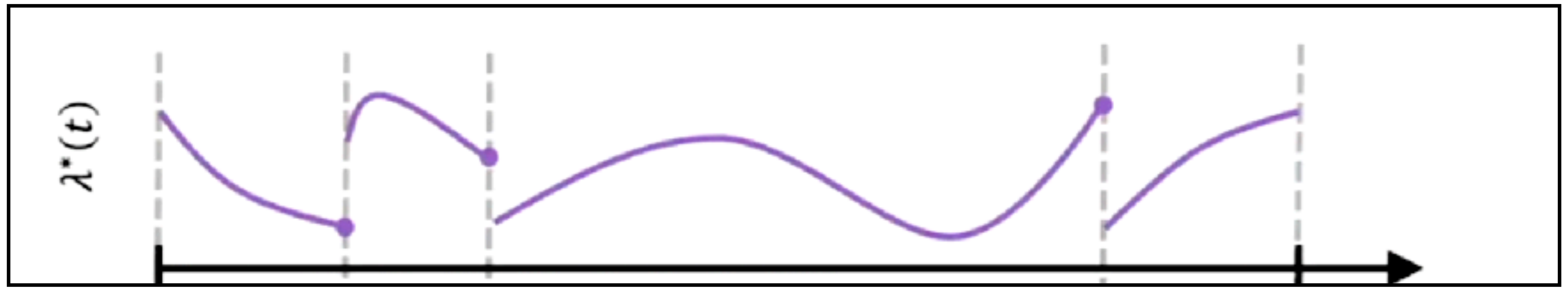
$$f^{pred}(t, H_t) = \mathrm{NN}(t, H_t)$$

- Predict valid PDF
- PDF representation must support both efficient sampling and evaluation.
- NN architecture needs to process event sequence of arbitrary length.

Neural Temporal Point Processes



youtube.com/watch?v=J7qH7i0EyfU&t=749s&ab_channel=TUM-DAML



1. Learn the latent feature $h(t)$ for each point in time.
   Here: $h(t) \in \mathbb{R}^3$.
2. An **ODE** specifies how $h(t)$ evolves when no event is happening.
3. An **MLP** specifies how $h(t)$ jumps when an event occurs.
4. Another **MLP** converts $h(t)$ to the intensity value $\lambda(t, H_t)$.

Message Passing

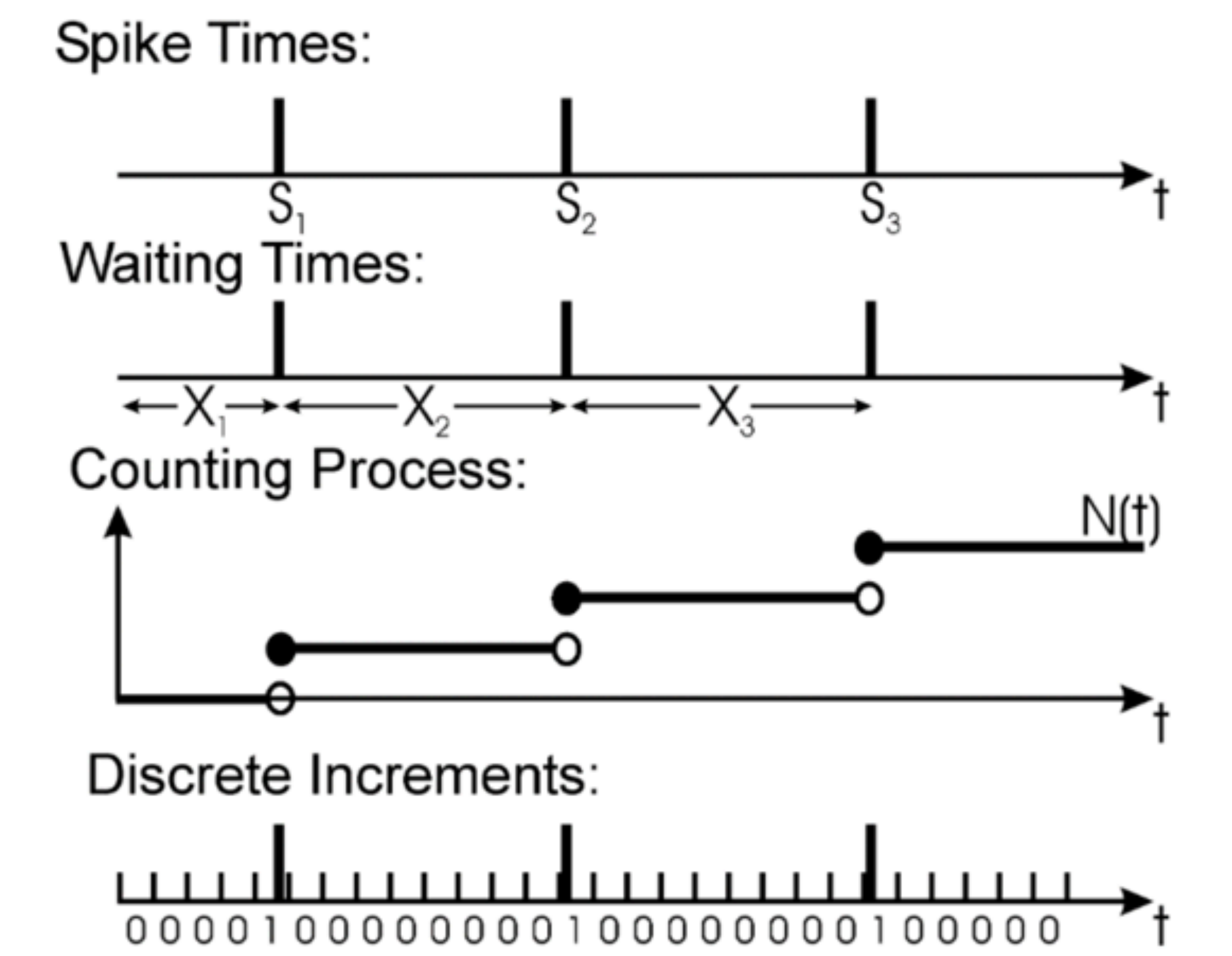
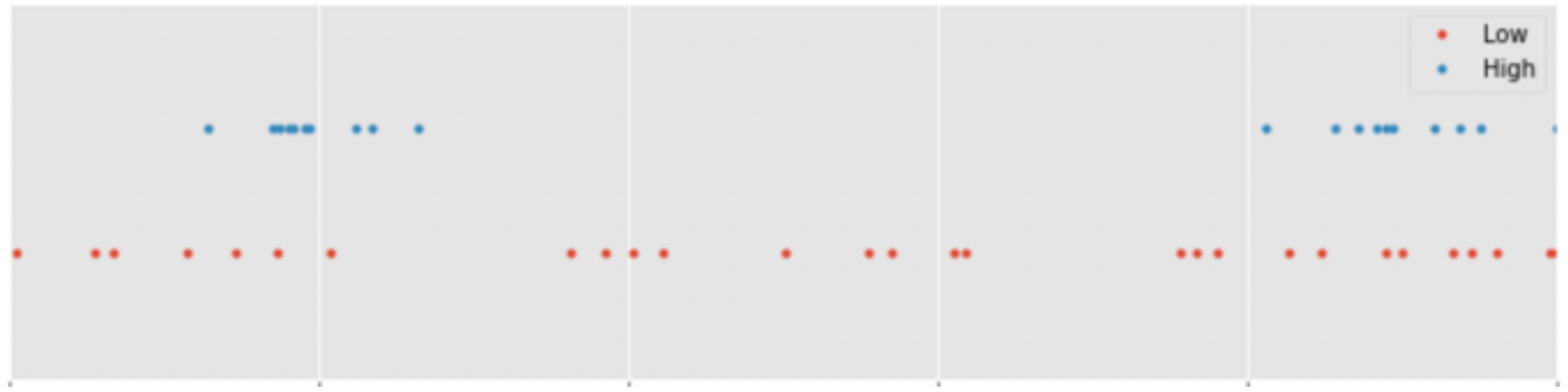
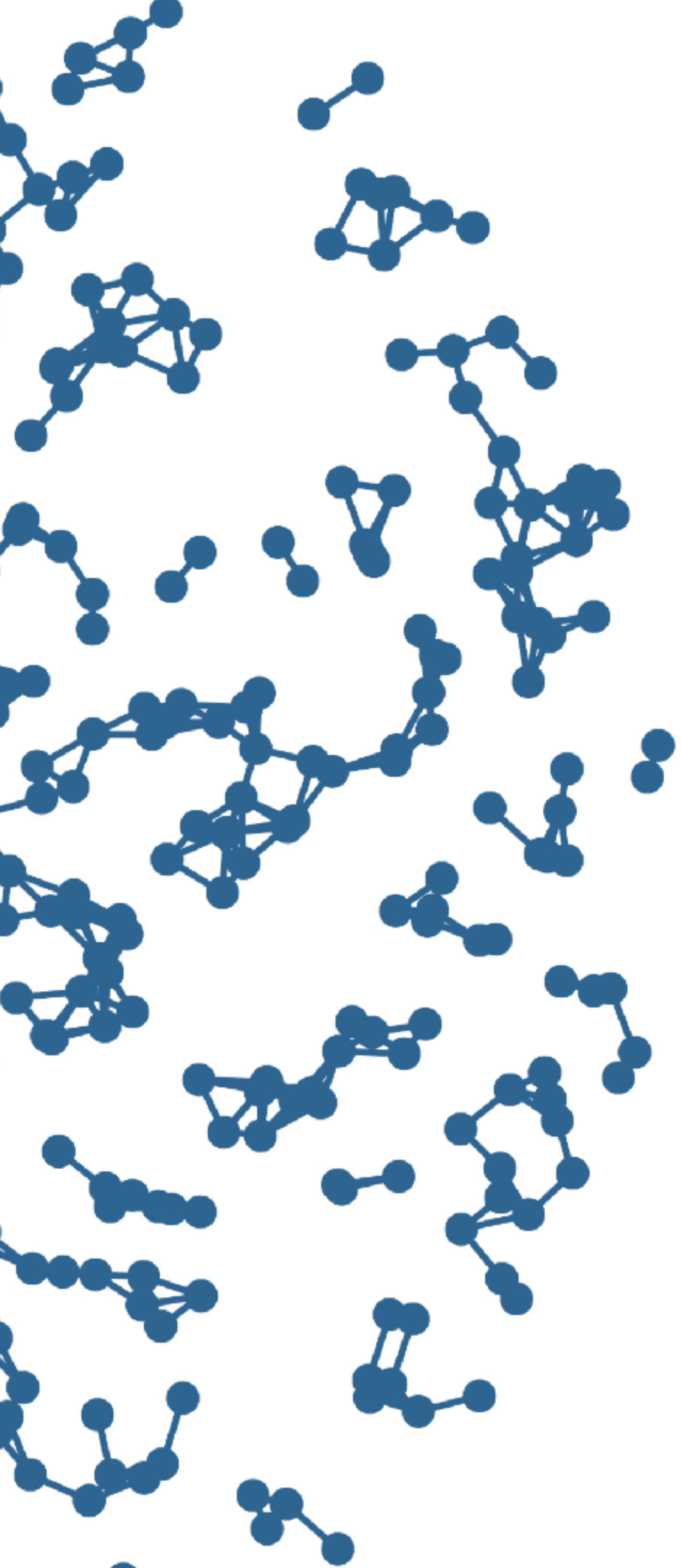


Figure 5. Multiple specifications for point process data.



https://mark-kramer.github.io/Case-Studies-Python/08.html

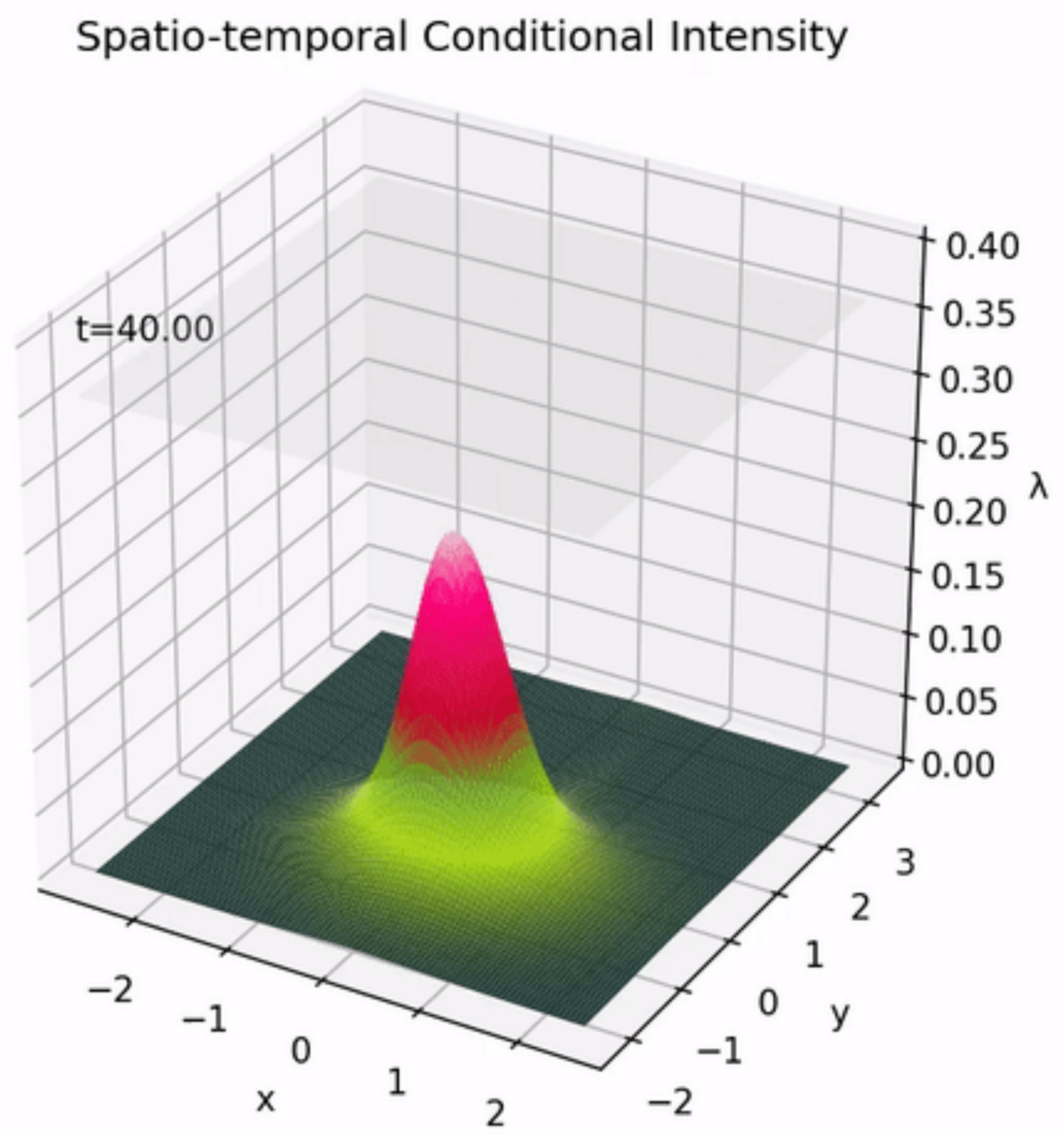
http://www.stat.columbia.edu/~liam/teaching/neurostat-spr11/uri-eden-point-process-notes.pdf



Part IV

SPATIO-TEMPORAL POINT PROCESSES

STPP



github.com/Rose-STL-Lab/DeepSTPP

- $H = [(t_1, s_1), (t_2, s_2), (t_3, s_3), \ldots]$
- t_i denotes the event time and $s_i \in \mathcal{S} \subset \mathbb{R}^2$ the location.
- E.g., $H = \big[\big(1.3,\ (0.3, 0.8)\big),\ \big(4.2,\ (0.6, 0.2)\big), \ldots\big]$

How does the intensity function look like?

Reminder **TPP**

$$\lambda(t, H_t) = \lim_{\Delta t \to 0} \frac{P(\text{Event in } [t, t + \Delta t) \mid H_t)}{\Delta t}$$

STPP

- $H = [(t_1, s_1), (t_2, s_2), (t_3, s_3), \ldots]$
- t_i denotes the event time and $s_i \in \mathcal{S} \subset \mathbb{R}^2$ the location.
- E.g., $H = [\big(1.3,\ (0.3, 0.8)\big),\ \big(4.2,\ (0.6, 0.2)\big), \ldots]$

Reminder **TPP**

$$\lambda(t, H_t) = \lim_{\Delta t \to 0} \frac{P(\text{Event in } [t, t + \Delta t) \mid H_t)}{\Delta t}$$

$$\lambda(s, t \mid H_t) := \lim_{\Delta s \to 0, \Delta t \to 0}$$

STPP

- $H = [(t_1, s_1), (t_2, s_2), (t_3, s_3), \ldots]$
- t_i denotes the event time and $s_i \in \mathcal{S} \subset \mathbb{R}^2$ the location.
- E.g., $H = [\big(1.3,\ (0.3, 0.8)\big),\ \big(4.2,\ (0.6, 0.2)\big), \ldots]$

Reminder **TPP**

$$\lambda(t, H_t) = \lim_{\Delta t \to 0} \frac{P(\text{Event in } [t, t + \Delta t) \mid H_t)}{\Delta t}$$

$$\lambda(s, t \mid H_t) := \lim_{\Delta s \to 0, \Delta t \to 0} \frac{P(\text{Event in } B(s, \Delta s) \times [t, t + \Delta t) | H_t)}{|B(s, \Delta s)| \Delta t}\ .$$

STPP

- $H = [(t_1, s_1), (t_2, s_2), (t_3, s_3), \dots]$
- t_i denotes the event time and $s_i \in \mathcal{S} \subset \mathbb{R}^2$ the location.
- E.g., $H = \big[\big(1.3,\ (0.3, 0.8)\big),\ \big(4.2,\ (0.6, 0.2)\big), \dots\big]$

Probability of an event in a very small unit of space and time

Reminder **TPP**

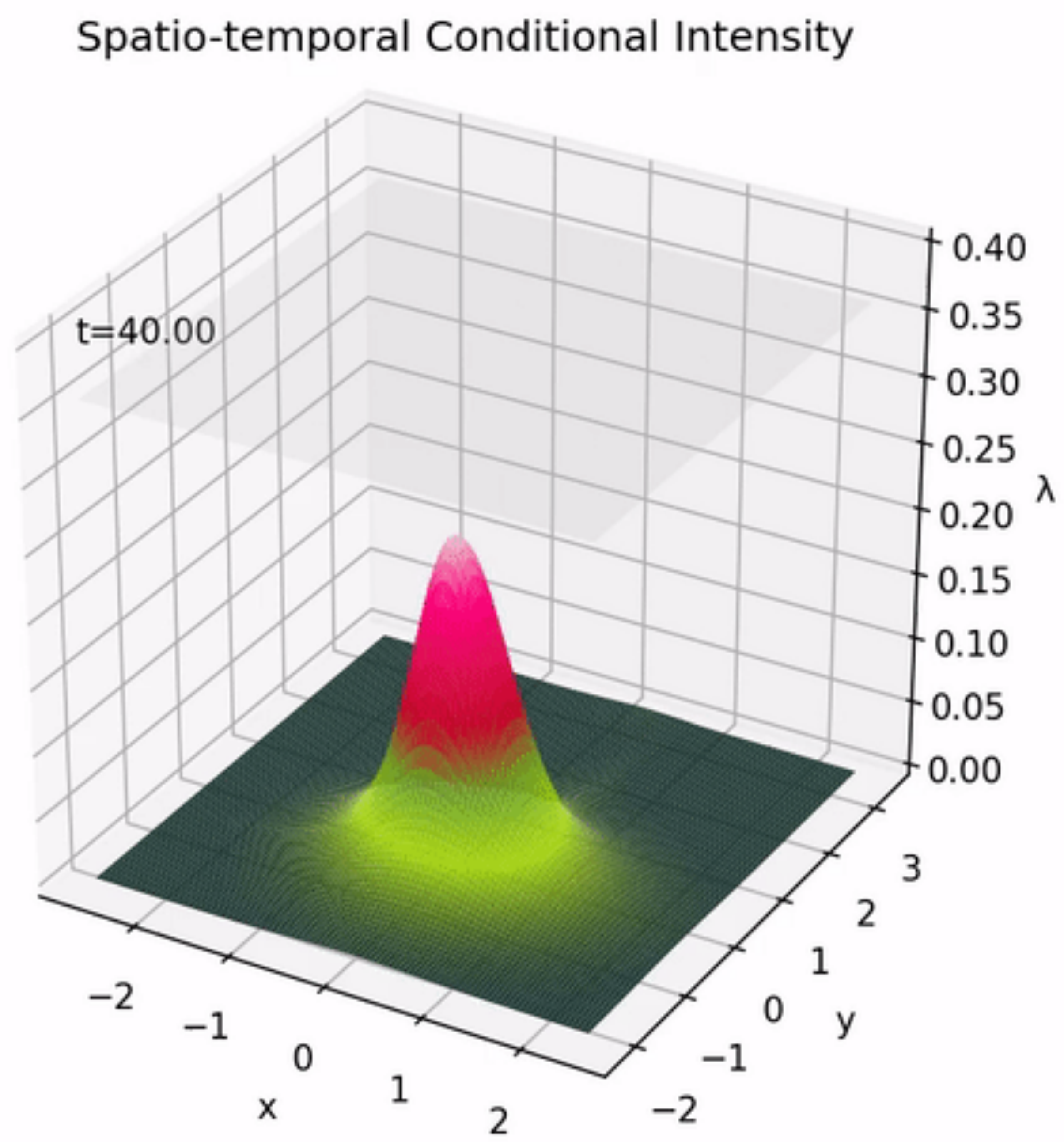
$$\lambda(t, H_t) = \lim_{\Delta t \to 0} \frac{P(\text{Event in } [t, t + \Delta t) \mid H_t)}{\Delta t}$$

$$\lambda(s, t \mid H_t) := \lim_{\Delta s \to 0, \Delta t \to 0} \frac{P(\text{Event in } B(s, \Delta s) \times [t, t + \Delta t) | H_t)}{|B(s, \Delta s)| \Delta t} .$$

A 2D-ball (disk) around s with radius Δs

Very small time interval

STPP



github.com/Rose-STL-Lab/DeepSTPP

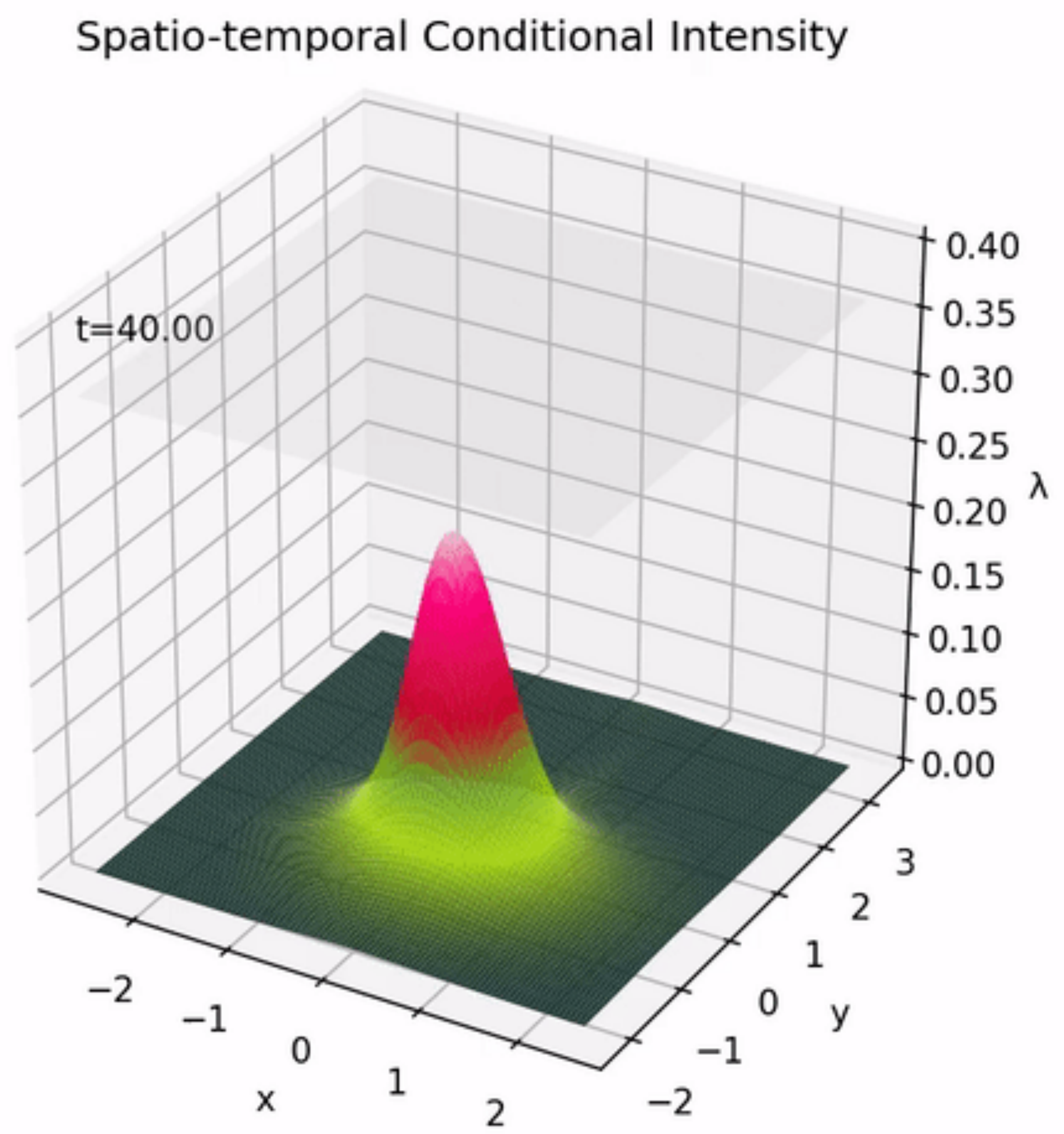
- $H = [(t_1, s_1), (t_2, s_2), (t_3, s_3), \ldots]$
- t_i denotes the event time and $s_i \in \mathcal{S} \subset \mathbb{R}^2$ the location.
- E.g., $H = \big[\big(1.3,\ (0.3, 0.8)\big),\ \big(4.2,\ (0.6, 0.2)\big), \ldots\big]$

How do we specify an intensity?

Reminder **TPP**

$$\lambda(t, H_t) = \mu + \sum_{i:t_i<t} \phi(t - t_i)$$

STPP



github.com/Rose-STL-Lab/DeepSTPP

- $H = [(t_1, s_1), (t_2, s_2), (t_3, s_3), \ldots]$
- t_i denotes the event time and $s_i \in \mathcal{S} \subset \mathbb{R}^2$ the location.
- E.g., $H = \big[\big(1.3,\ (0.3, 0.8)\big),\ \big(4.2,\ (0.6, 0.2)\big), \ldots\big]$

How do we specify an intensity?

$$\lambda(s, t, H_t) = \mu g_0(s) + \sum_{(t_i, s_i) \in H_t} g_1(t, t_i) g_2(s, s_i)$$

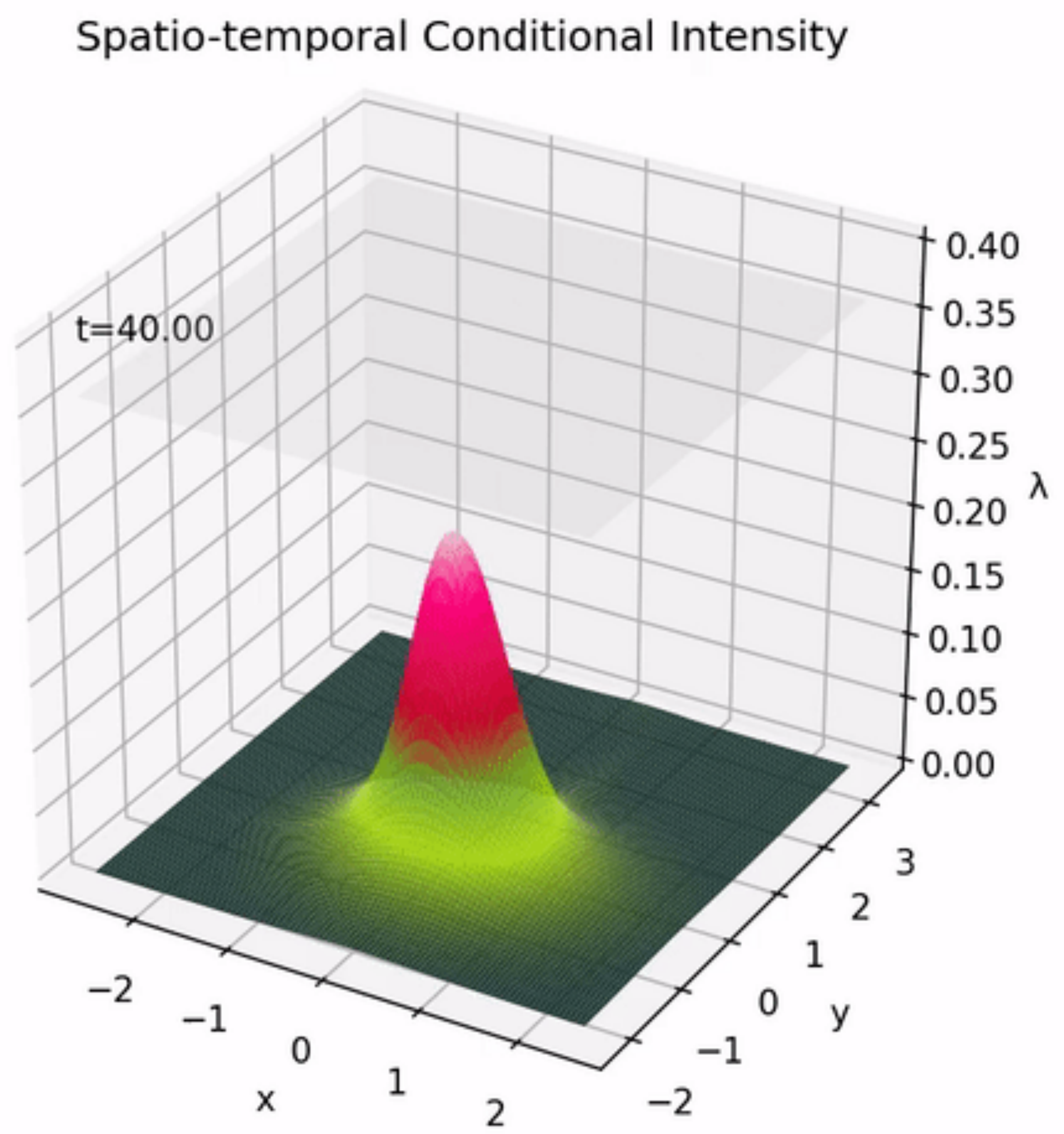
with

$$g_1(t, t_i) = \alpha \exp(-\beta(t - t_i))$$

and

$g_2(s, s_i)$ being a Normal distribution.

STPP



github.com/Rose-STL-Lab/DeepSTPP

- $H = [(t_1, s_1), (t_2, s_2), (t_3, s_3), \dots]$
- t_i denotes the event time and $s_i \in \mathcal{S} \subset \mathbb{R}^2$ the location.
- E.g., $H = \big[\big(1.3,\ (0.3, 0.8)\big),\ \big(4.2,\ (0.6, 0.2)\big), \dots\big]$
- $\lambda(s, t, H_t) = \mu g_0(s) + \sum_{(t_i, s_i) \in H_t} g_1(t, t_i) g_2(s, s_i)$

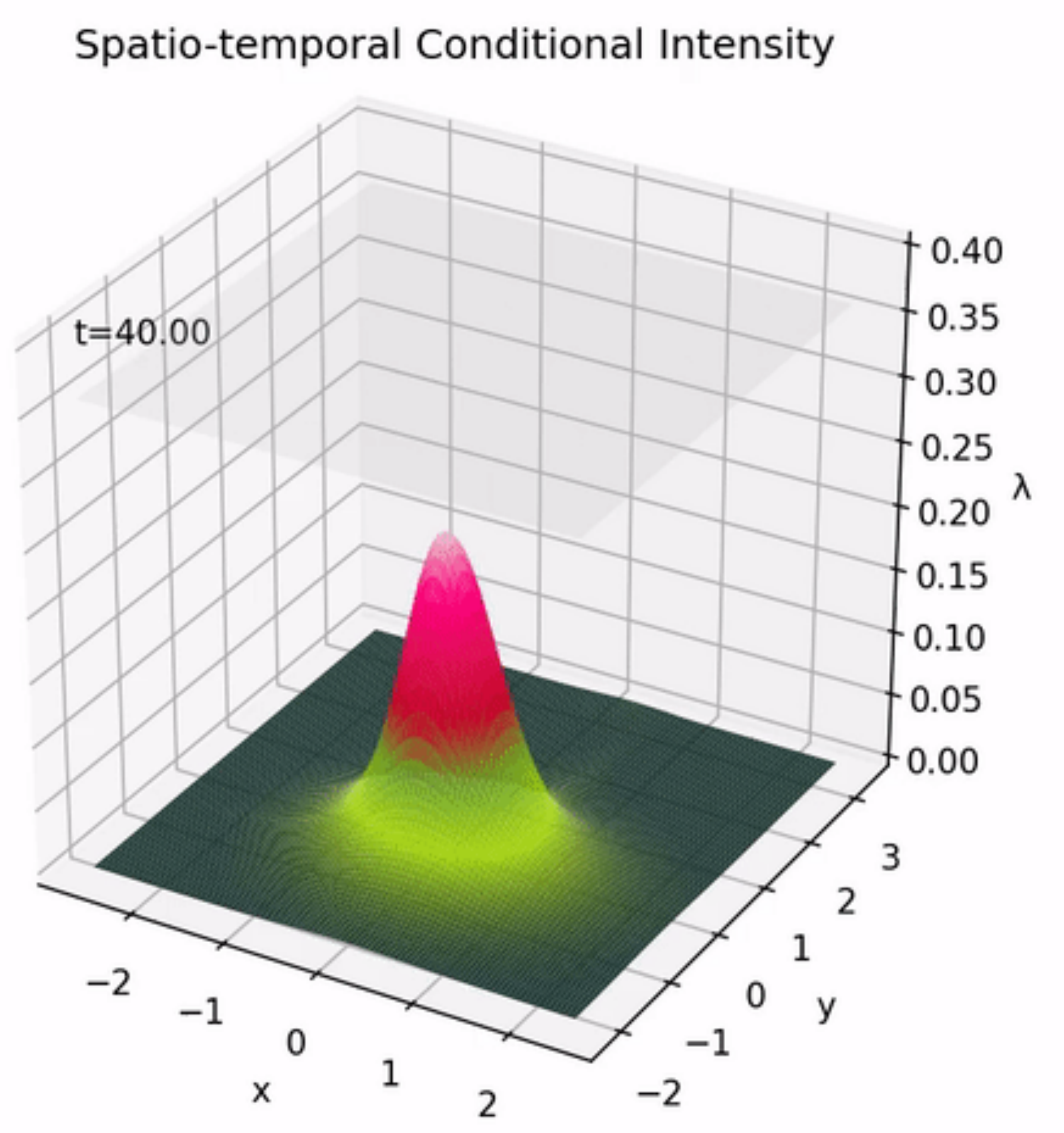
How do we simulate STPPs?

Reminder **TPP**

Initialize $H = []$
Until horizon is reached:

- Sample next event time t_i using $\lambda(\,\cdot \mid H)$ $\quad (t_i > \max(H))$
- Add t_i to H

STPP



github.com/Rose-STL-Lab/DeepSTPP

- $H = [(t_1, s_1), (t_2, s_2), (t_3, s_3), \ldots]$
- t_i denotes the event time and $s_i \in \mathcal{S} \subset \mathbb{R}^2$ the location.
- E.g., $H = \left[\left(1.3,\ (0.3, 0.8)\right),\ \left(4.2,\ (0.6, 0.2)\right), \ldots\right]$
- $\lambda(s, t, H_t) = \mu g_0(s) + \sum_{(t_i, s_i) \in H_t} g_1(t, t_i) g_2(s, s_i)$

How do we simulate STPPs?

Reminder **TPP**

But how?

Initialize $H = []$
Until horizon is reached:

- Sample next event time t_i using $\lambda(\cdot \mid H)$ $\quad (t_i > \max(H))$
- Add t_i to H

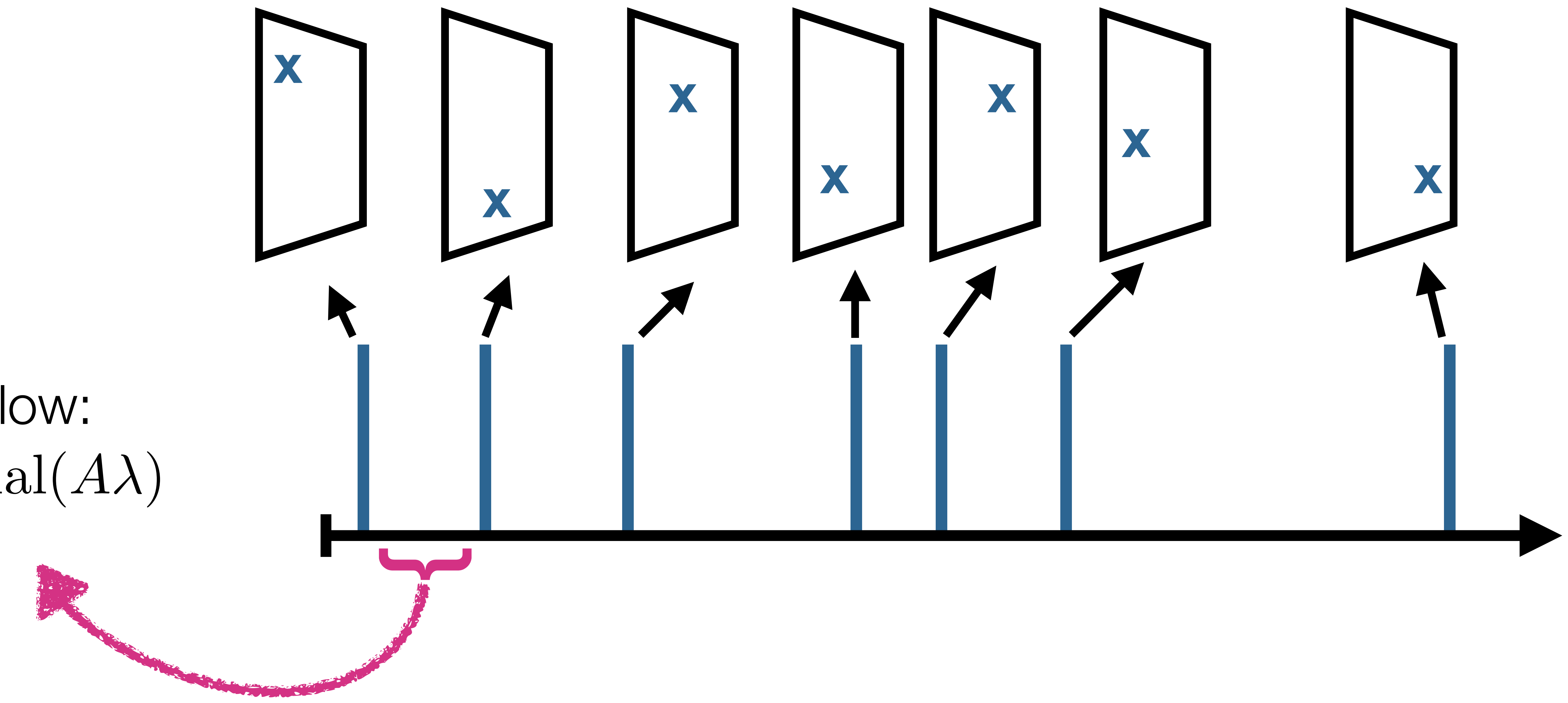
Simulating Homogeneous STPPs

Homogeneous STPP Assume we want to simulate on a space $S = [0,1] \times [0,1]$ with area $A = 1$ from time $t = 0$ to 1

1. Simulate event times with rate $\lambda \cdot A$, where λ is the rate and A is the area of the space (typically also 1).

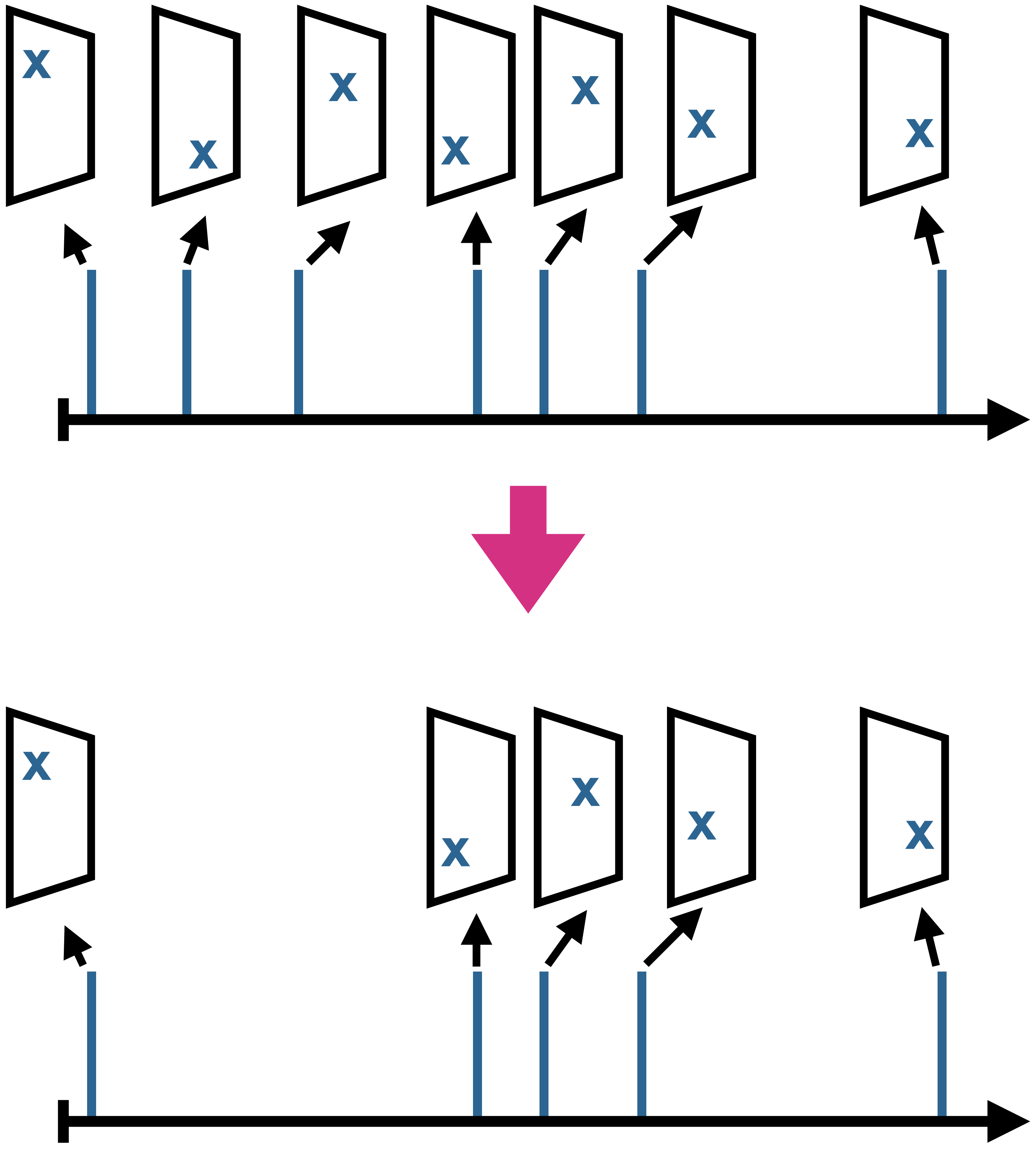
2. For each event time, place the event uniformly over the space.

Inter-event times follow:
$t_{i+1} - t_i \sim \mathrm{Exponential}(A\lambda)$

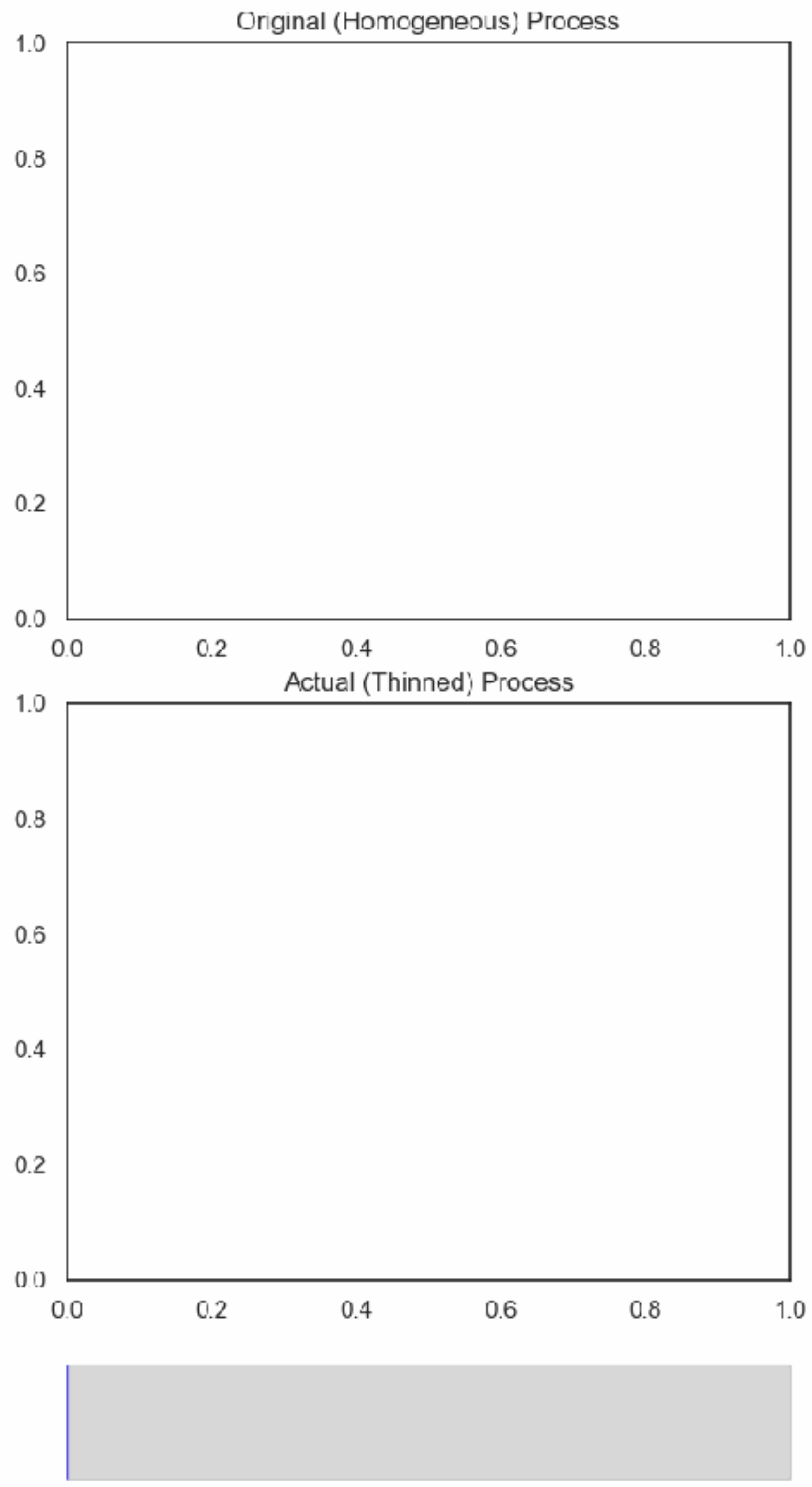


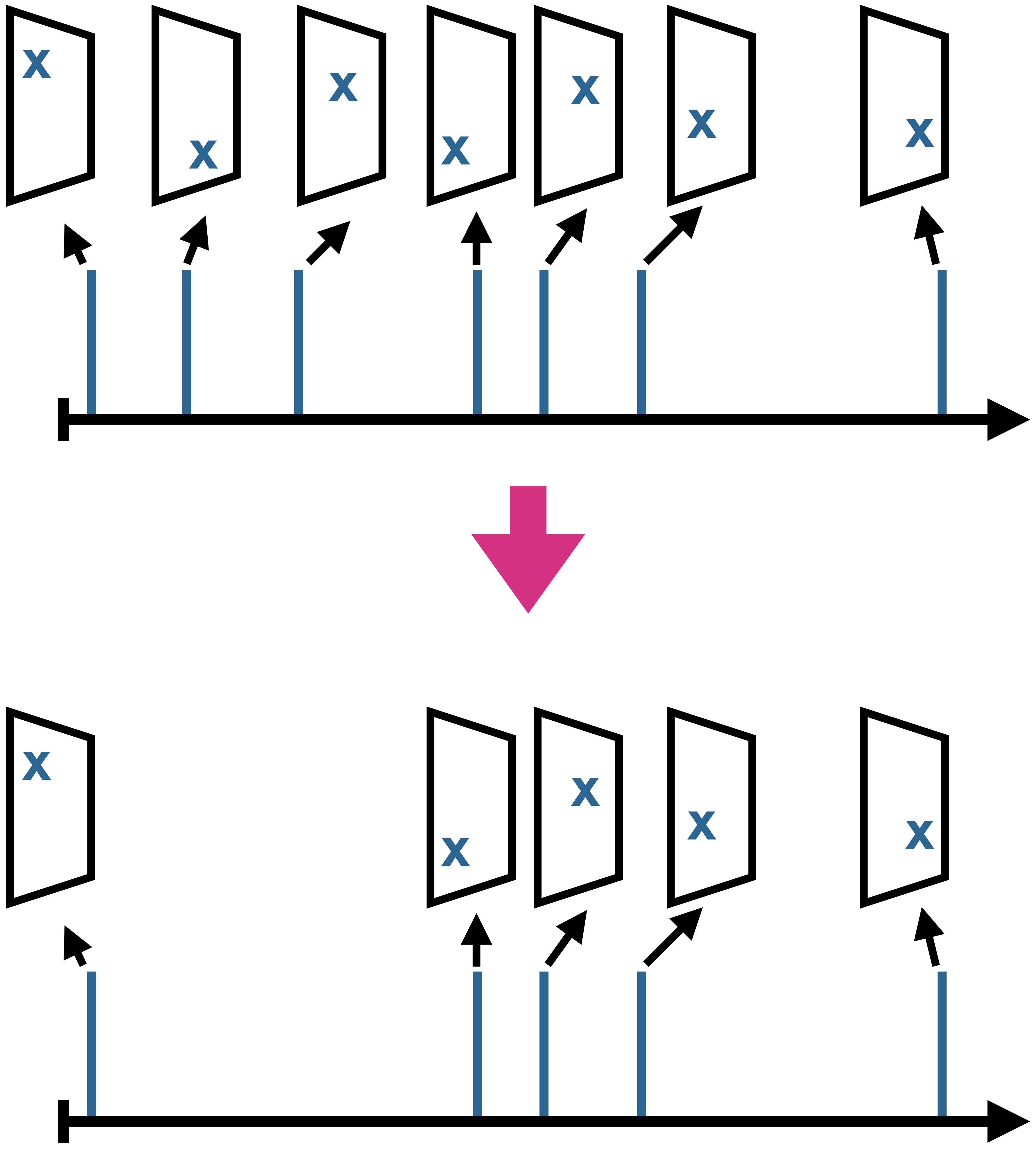
Simulating Heterogeneous STPPs

1. Generate a homogeneous STPP with a rate $\lambda_{\max}$ that upper bounds the actual rate.
2. For each event (from left to right):
   (a) Compute the actual intensity $\lambda(s, t \mid H_t)$.
   (b) Reject the event with probability $1 - \frac{\lambda(s,t|H_t)}{\lambda_{\max}}$.

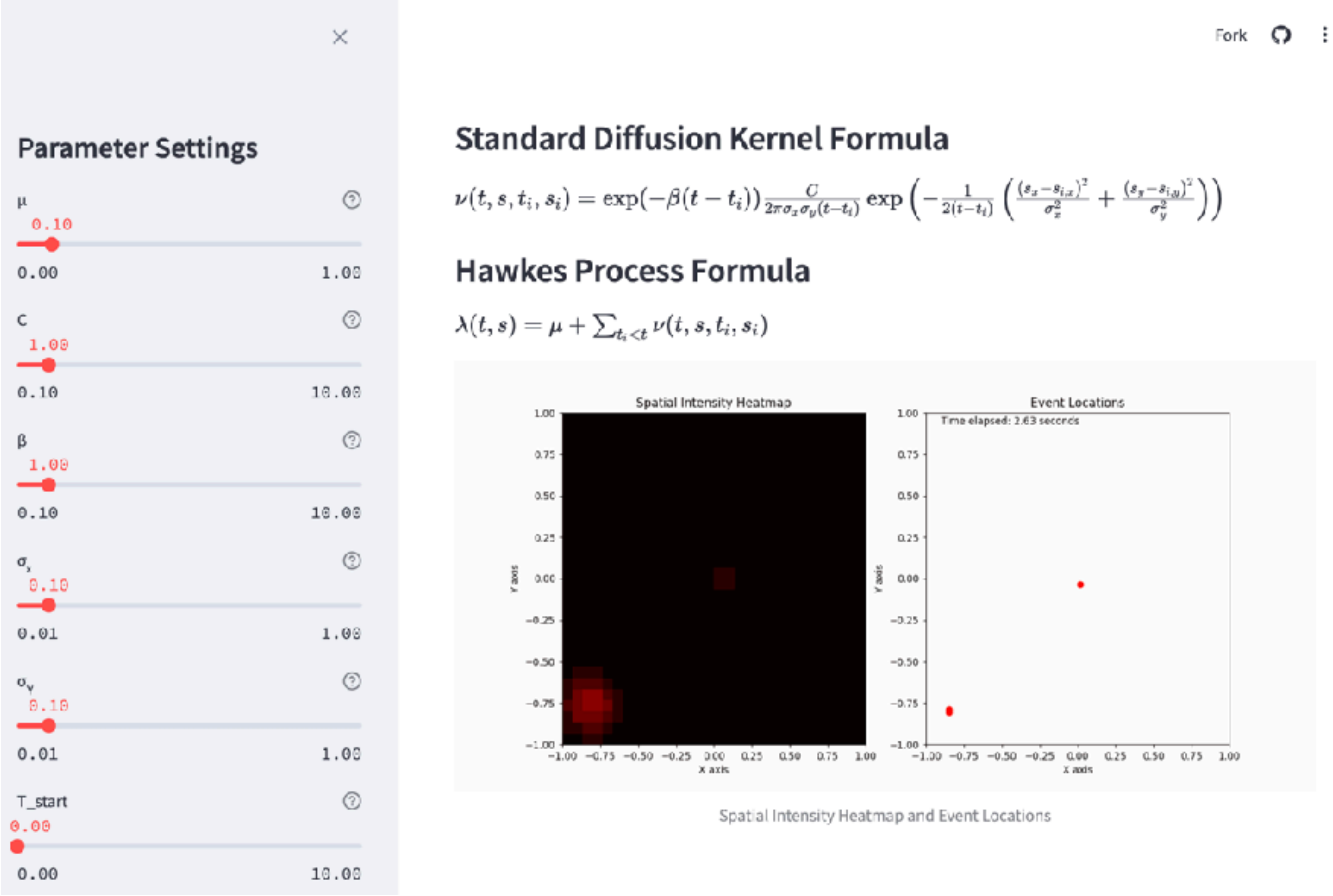


Simulating Heterogeneous STPPs



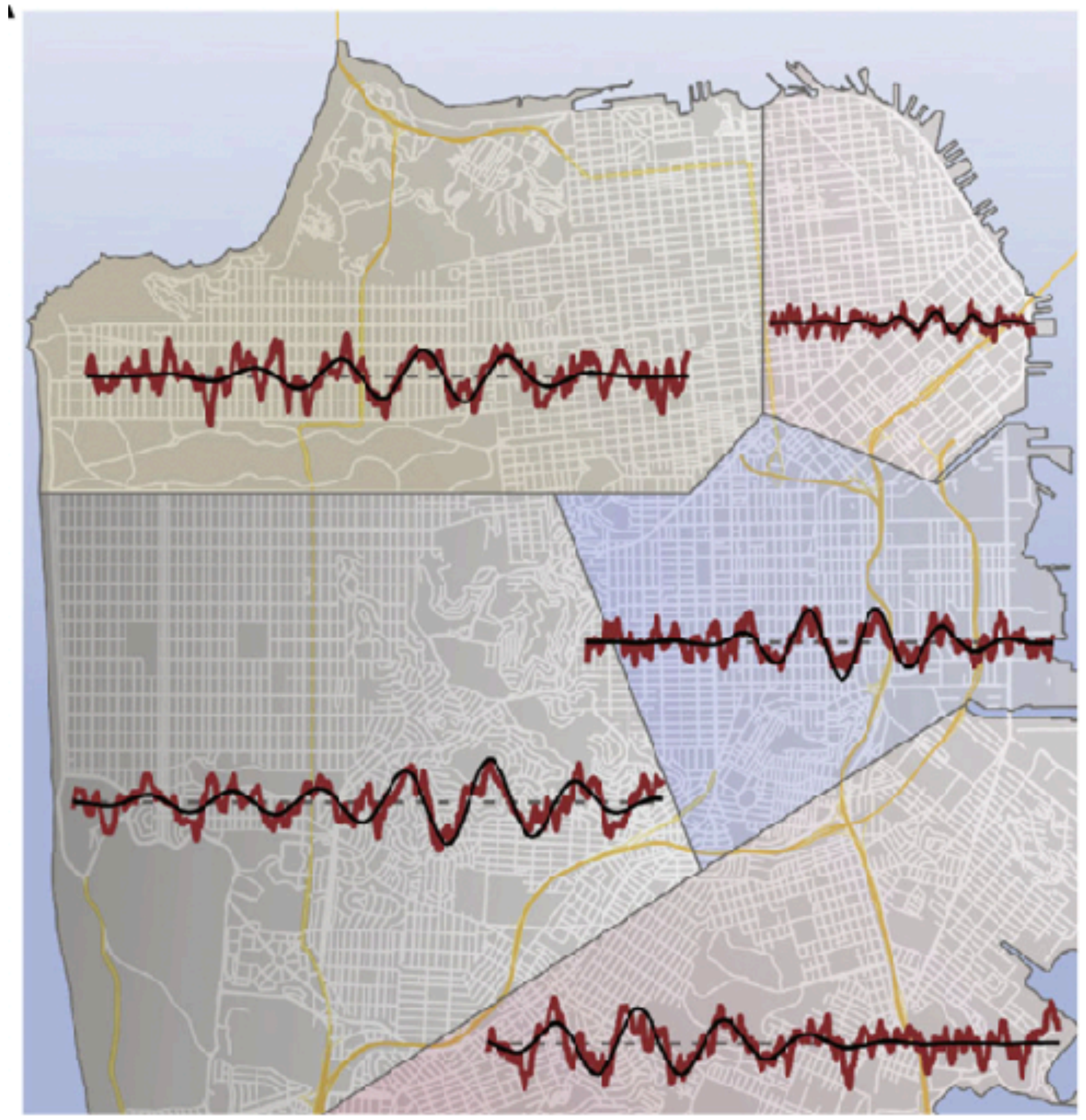


Simulating Heterogeneous STPPs



stpp-simulator.streamlit.app

Applications



Spatio-temporal variations in the urban rhythm: the travelling waves of crime (Oliveira)

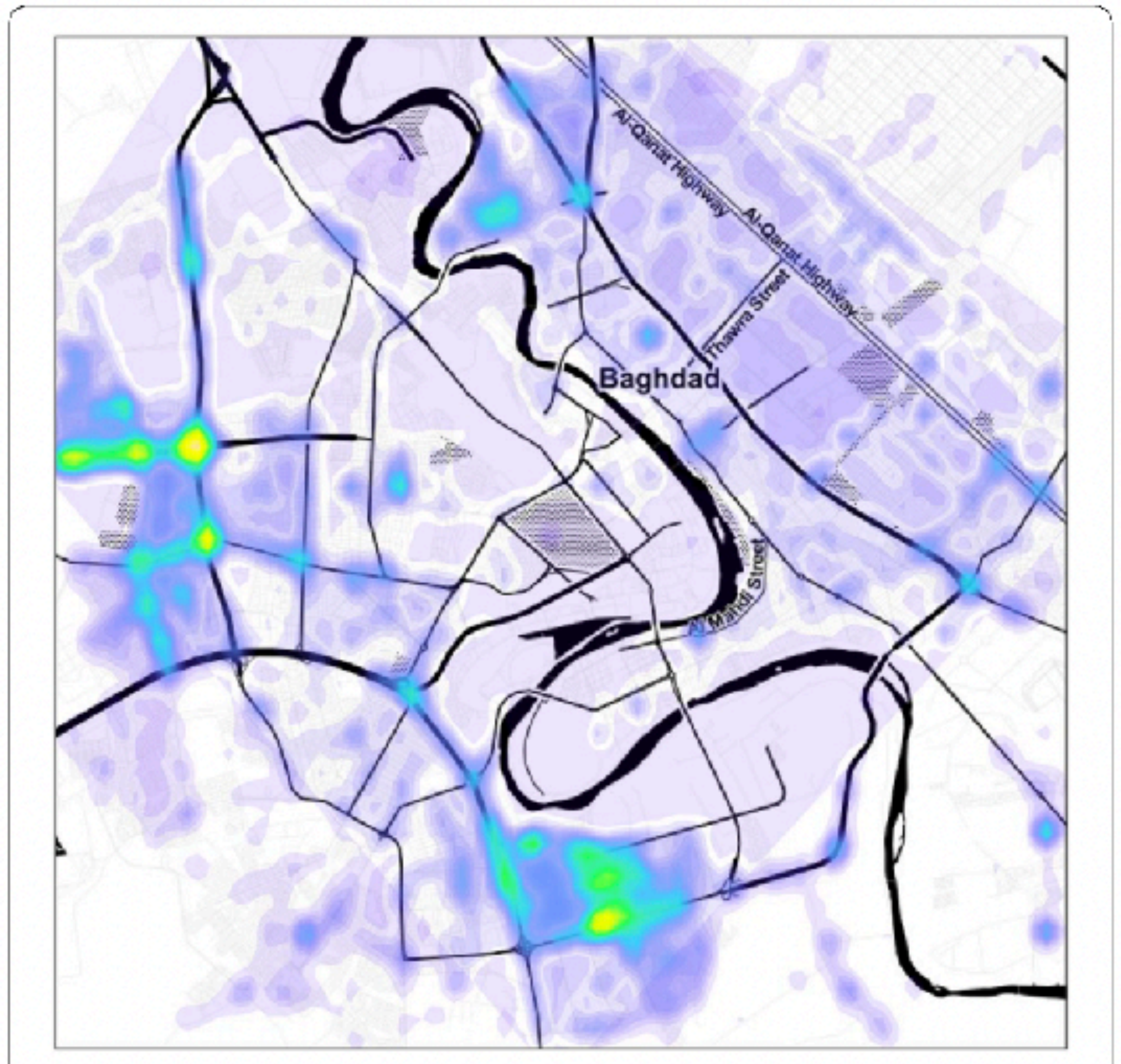
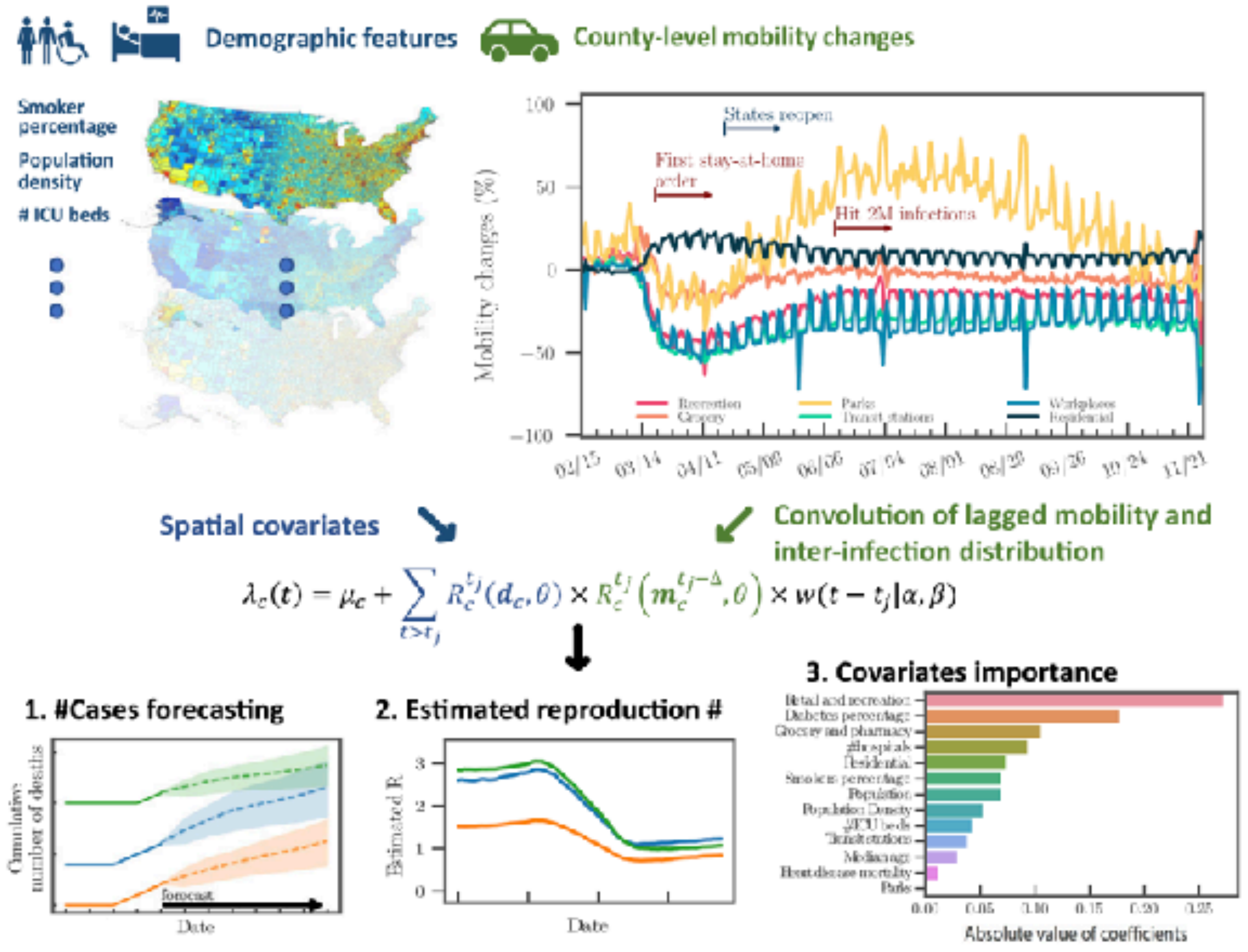


Fig. 1 IED attack hotspots in Baghdad from 2004 to 2009

Learning to rank spatio-temporal event hotspots (Mohler et al.)



Hawkes process modeling of COVID-19 with mobility leading indicators and spatial covariates (Chiang et al.)

Part V

NEURAL STTPS

Neural STTPs

How do we generalize this towards STPPs?

Model **intensity** function with neural network: $\lambda(t, H_t) = \mathrm{NN}(t, H_t)$

- Numerical integration is expensive
- NN architecture needs to process event sequence of arbitrary length.

Model **predictive distribution** with neural network: $f^{pred}(t, H_t) = \mathrm{NN}(t, H_t)$

- Predict valid PDF
- PDF representation must support both efficient sampling and evaluation.
- NN architecture needs to process event sequence of arbitrary length.

Neural STTPs

How do we generalize this towards STPPs?

Model **intensity** function with neural network: $\lambda(t, H_t) = \mathrm{NN}(t, H_t)$

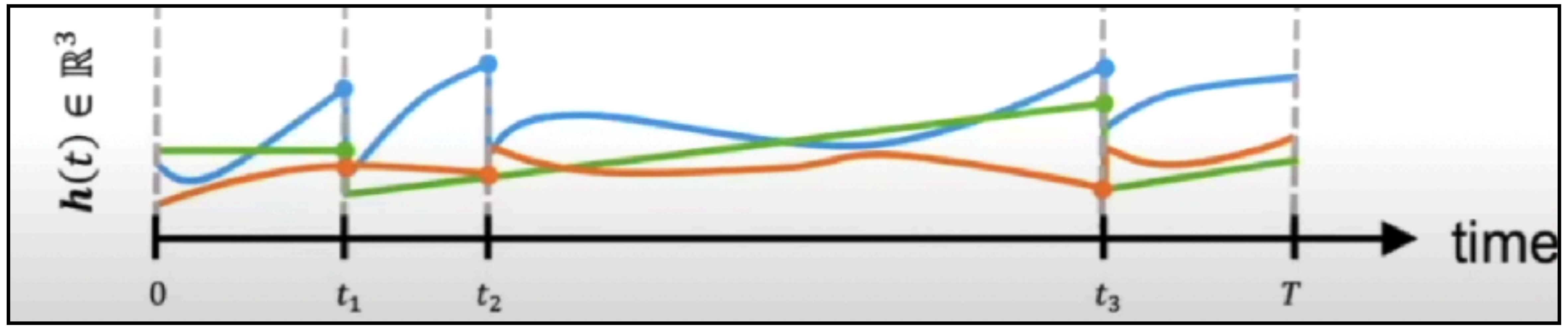
- Numerical integration is expensive
- NN architecture needs to process event sequence of arbitrary length.

Model **predictive distribution** with neural network: $f^{\mathrm{pred}}(t, H_t) = \mathrm{NN}(t, H_t)$

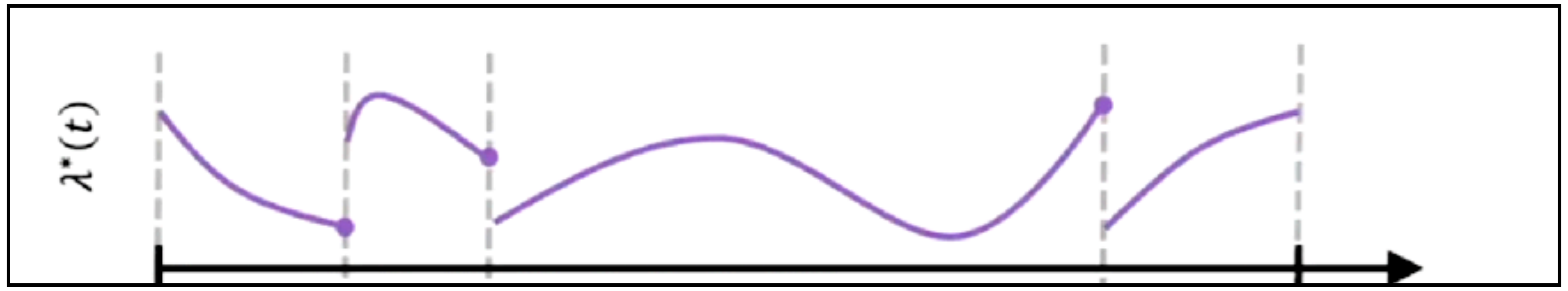
- Predict valid PDF
- PDF representation must support both efficient sampling and evaluation.
- NN architecture needs to process event sequence of arbitrary length.

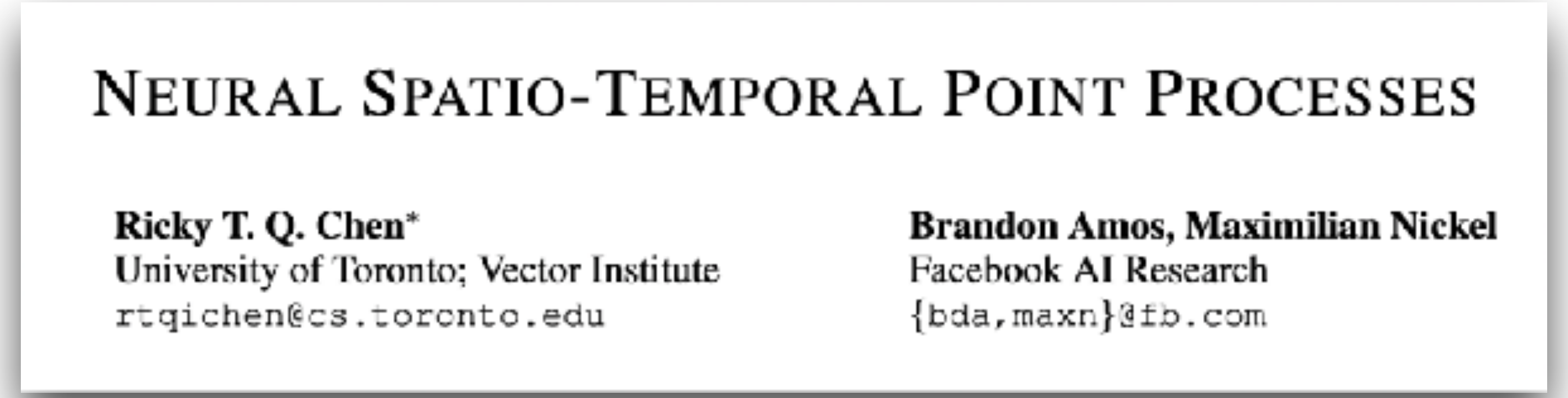
Neural STTPs

$$\lambda(t, \mathbf{s} \mid H_t) = \lambda(t \mid H_t) \cdot p(\mathbf{s} \mid H_t)$$



youtube.com/watch?v=J7qH7i0EyfU&t=749s&ab_channel=TUM-DAML




NEURAL SPATIO-TEMPORAL POINT PROCESSES

Ricky T. Q. Chen*
University of Toronto; Vector Institute
rtqichen@cs.toronto.edu

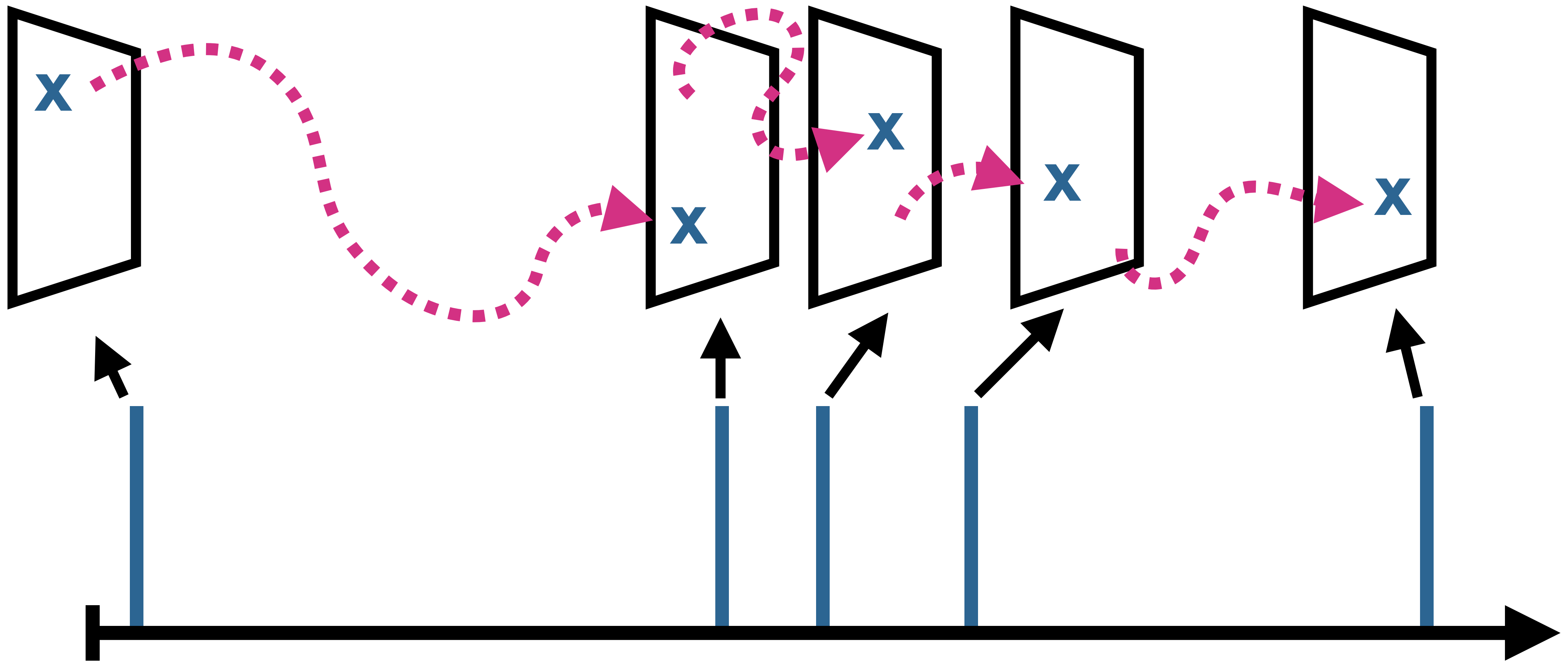
Brandon Amos, Maximilian Nickel
Facebook AI Research
{bda,maxn}@fb.com

Specify **ODE** that moves current position

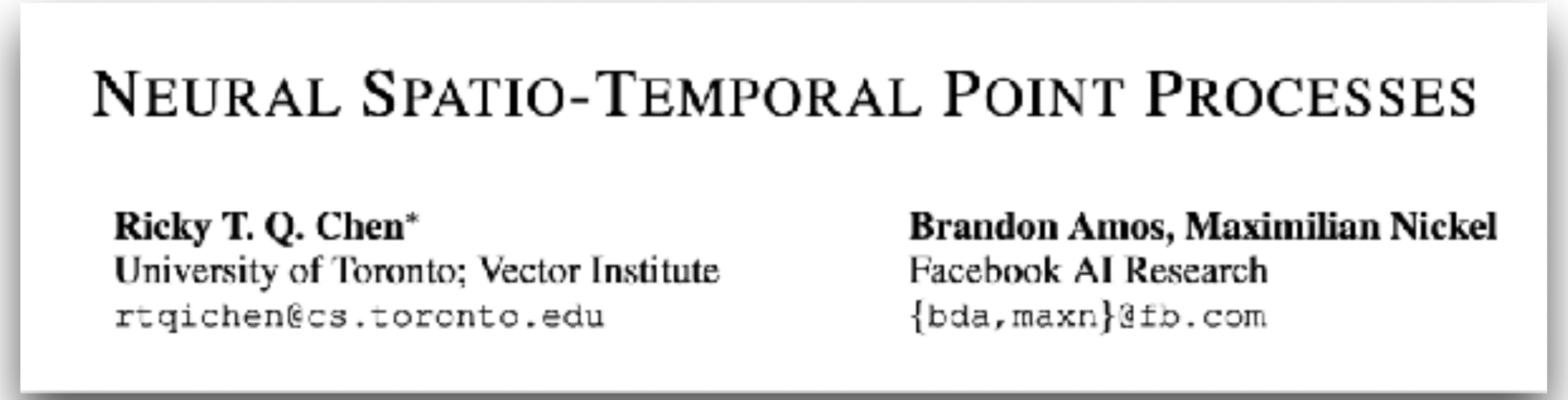
$$\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0)$$

$$\frac{d\boldsymbol{x}_t}{dt} = f_x(t, \boldsymbol{x}_t, \boldsymbol{h}_t)$$

$$\lim_{\varepsilon \to 0} \boldsymbol{x}_{t_i+\varepsilon} = g_x(t_i, \boldsymbol{x}_{t_i}, \boldsymbol{h}_{t_i})$$



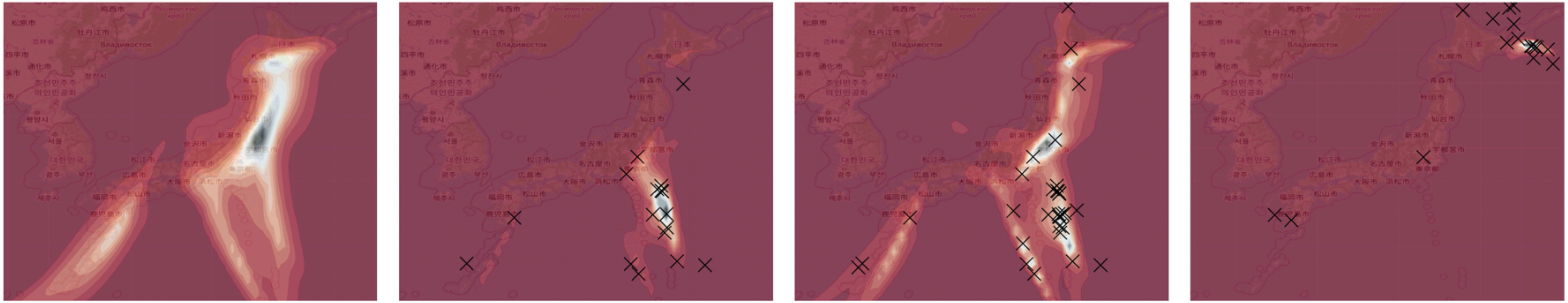
Neural STTPs



NEURAL SPATIO-TEMPORAL POINT PROCESSES

Ricky T. Q. Chen*
University of Toronto; Vector Institute
rtqichen@cs.toronto.edu

Brandon Amos, Maximilian Nickel
Facebook AI Research
{bda,maxn}@fb.com



Neural STTPs

How do we generalize this towards STPPs?

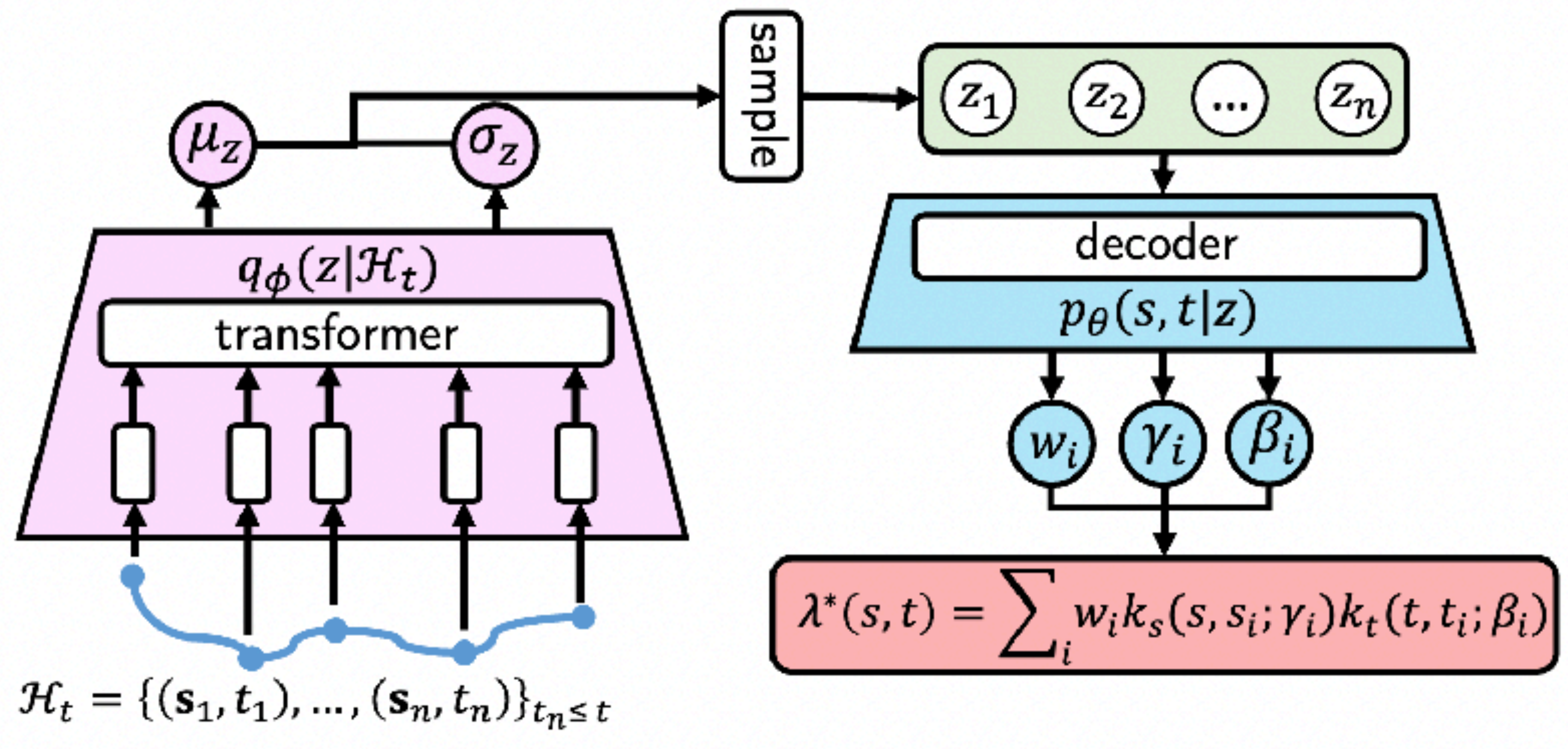
Model **intensity** function with neural network: $\lambda(t, H_t) = \mathrm{NN}(t, H_t)$

- Numerical integration is expensive
- NN architecture needs to process event sequence of arbitrary length.

Model **predictive distribution** with neural network: $f^{\mathrm{pred}}(t, H_t) = \mathrm{NN}(t, H_t)$

- Predict valid PDF
- PDF representation must support both efficient sampling and evaluation.
- NN architecture needs to process event sequence of arbitrary length.

Neural STTPs



Find parametric form of the intensity function:

$$\lambda(s, t|H_t) = \sum_{i=1}^{n} w_i k_s(s, s_i; \gamma_i) k_t(t, t_i; \beta_i)$$

For each event i in H_t we learn w_i, γ_i, β_i

Neural Point Process for Learning Spatiotemporal Event Dynamics

Zihao Zhou ZIZ244@UCSD.EDU
UC San Diego
Xingyi Yang XYANG@U.NUS.EDU
National University of Singapore
Ryan A. Rossi RYROSSI@ADOBE.COM
Adobe Research
Handong Zhao HAZHAO@ADOBE.COM
Adobe Research
Rose Yu ROSEYU@UCSD.EDU
UC San Diego

Define kernels:

normalization constant

$$k_s(s, s_i) = \alpha^{-1} \exp(-\gamma_i \|s - s_i\|)$$

$$k_t(t, t_i) = \exp(-\beta_i \|t - t_i\|)$$

Neural STTPs with Context

"Human activities are largely influenced by environmental features, i.e., weather, geographical characteristics and traffic conditions. These features must be considered to accurately predict future events."

511NY GET CONNECTED TO GO TRAFFIC, TRAVEL AND TRANSIT INFORMATION

511ny.org

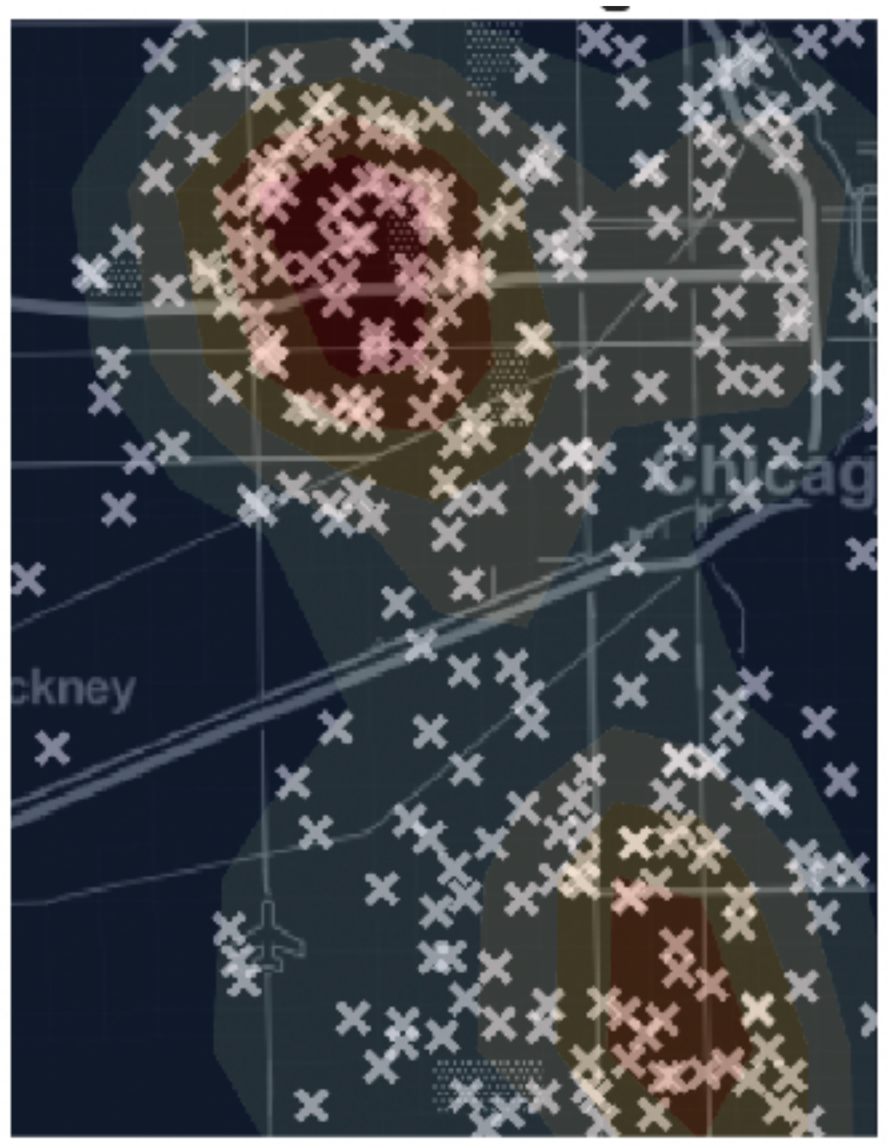
street construction works, sports events, musical concerts

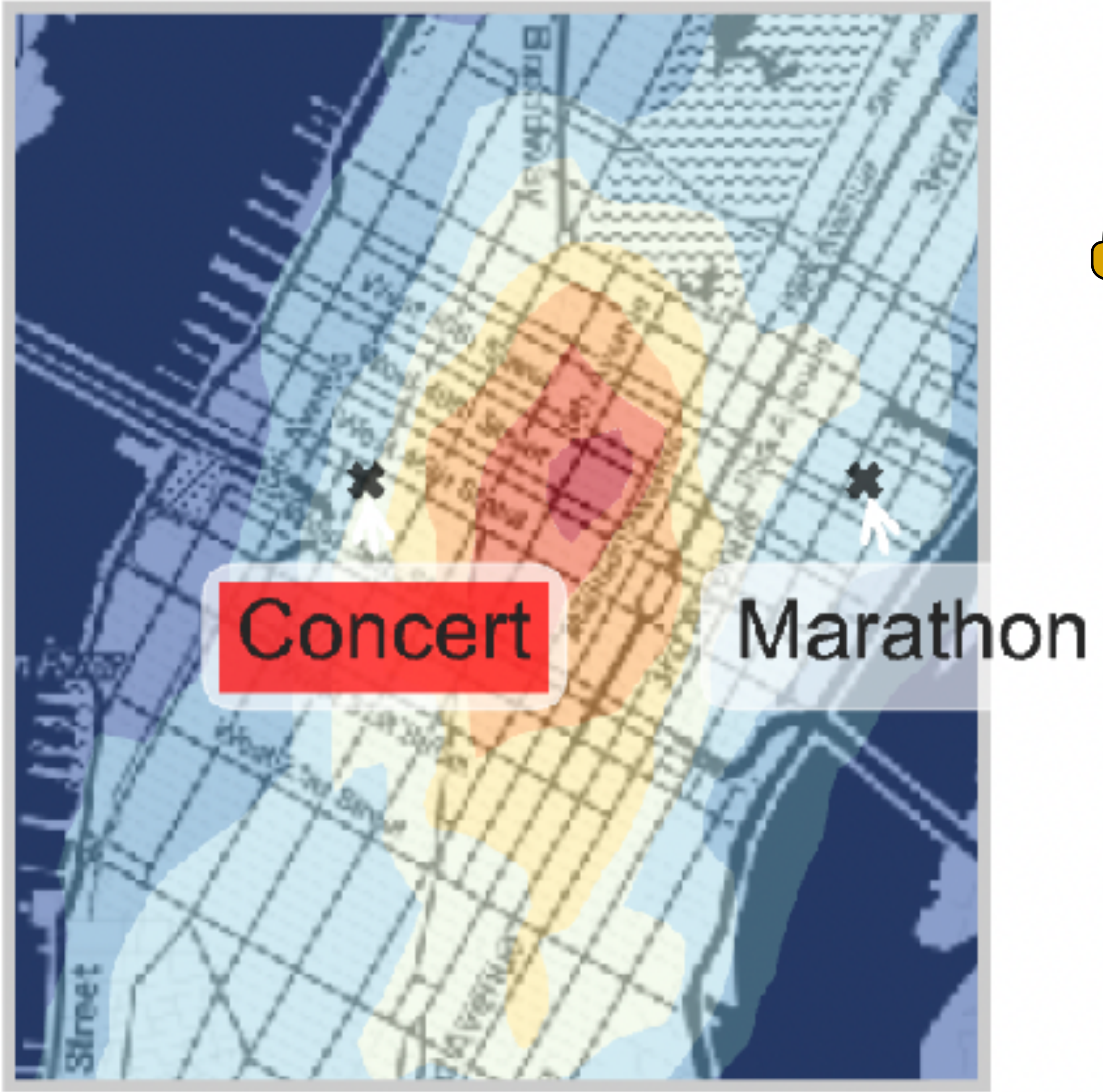
H_t

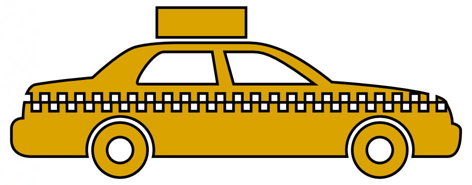
Fusion NN

OpenStreetMap

process maps as image attention network



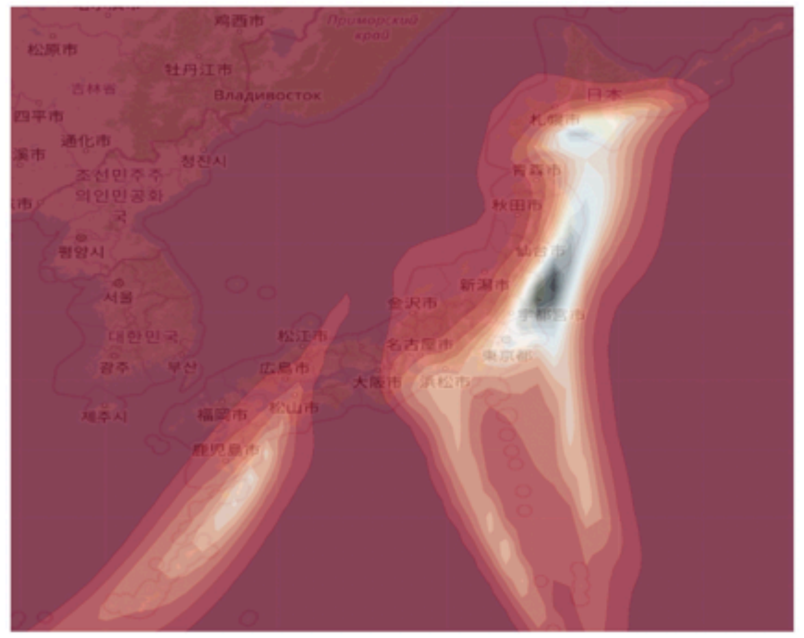




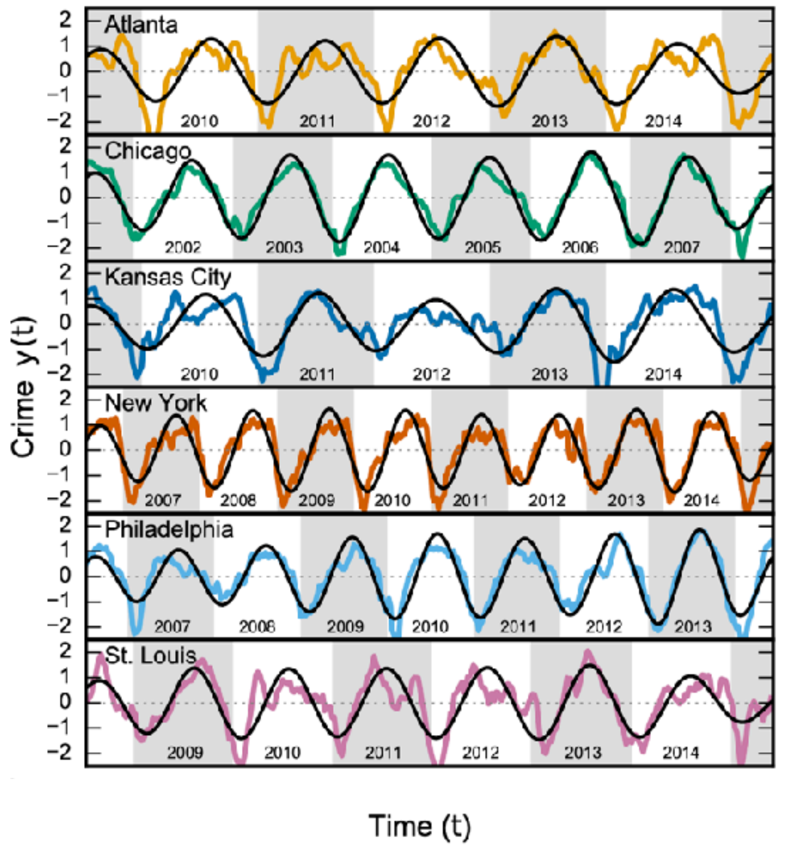
Deep Mixture Point Processes: Spatio-temporal Event Prediction with Rich Contextual Information

Maya Okawa[1], Tomoharu Iwata[2], Takeshi Kurashima[1], Yusuke Tanaka[1], Hiroyuki Toda[1] and Naonori Ueda[2]

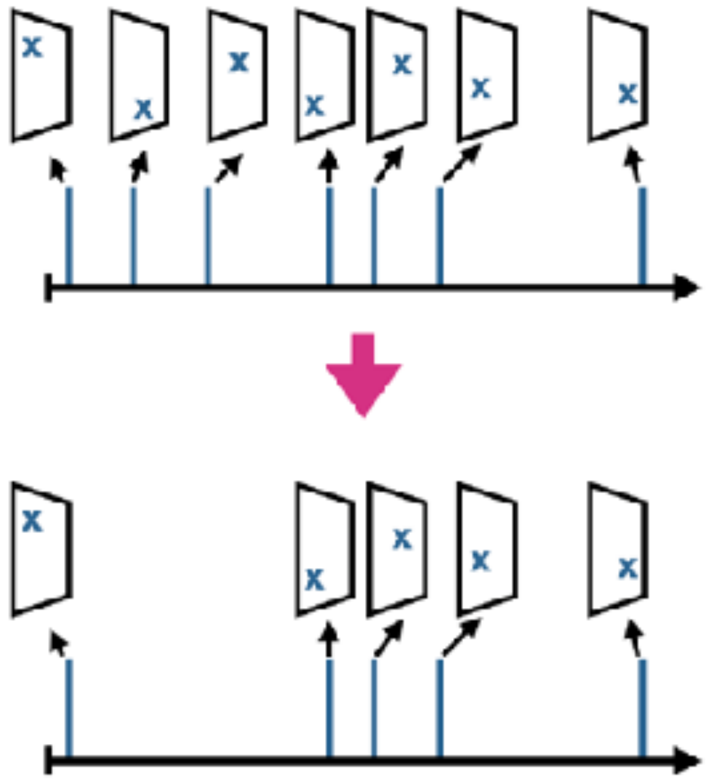
Recap



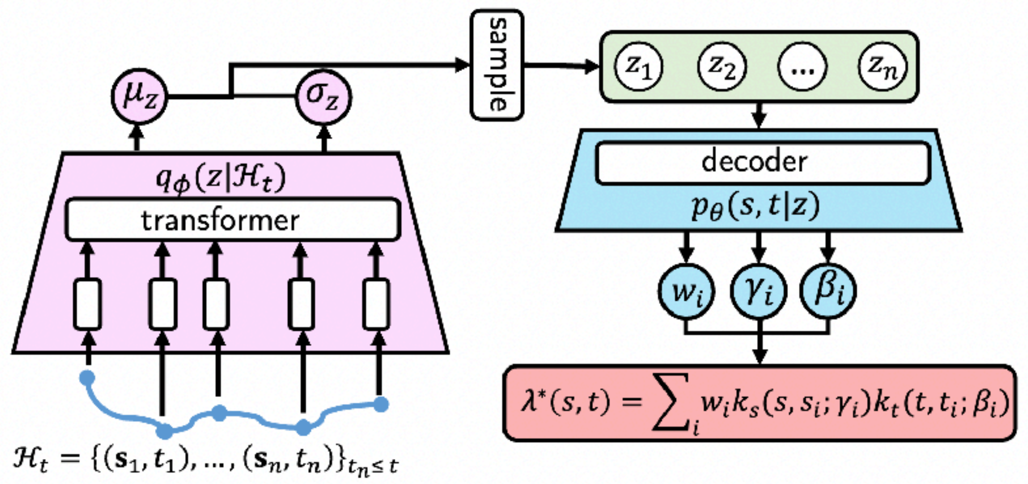
Spatio-temporal data is everywhere.



Statistical models capture patterns like bustiness, self-excitation, and self-correction



Simulation with the thinning method is easy and fast.



Neural models capture arbitrary dynamics and include contextual information.

Slides, code, and lecture notes available at github.com/gerritgr/spatio-temporal-lecture