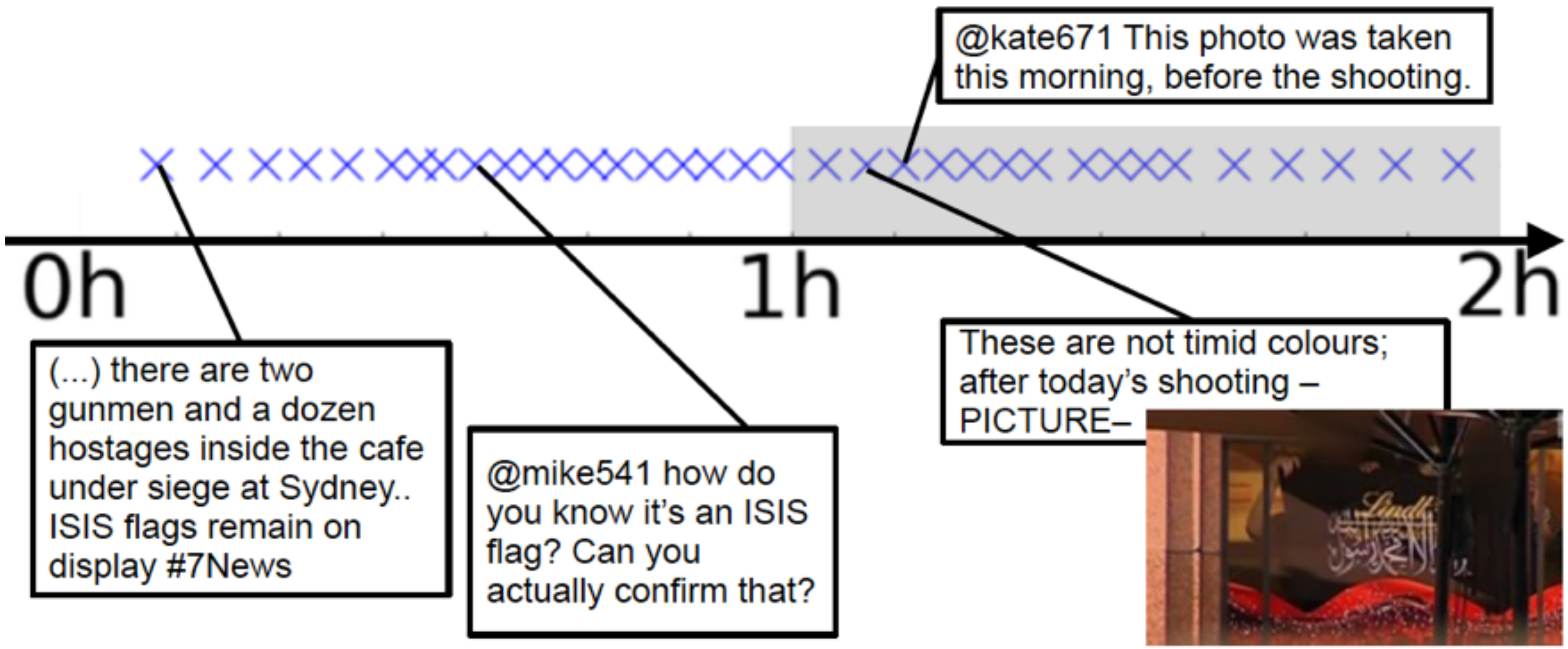
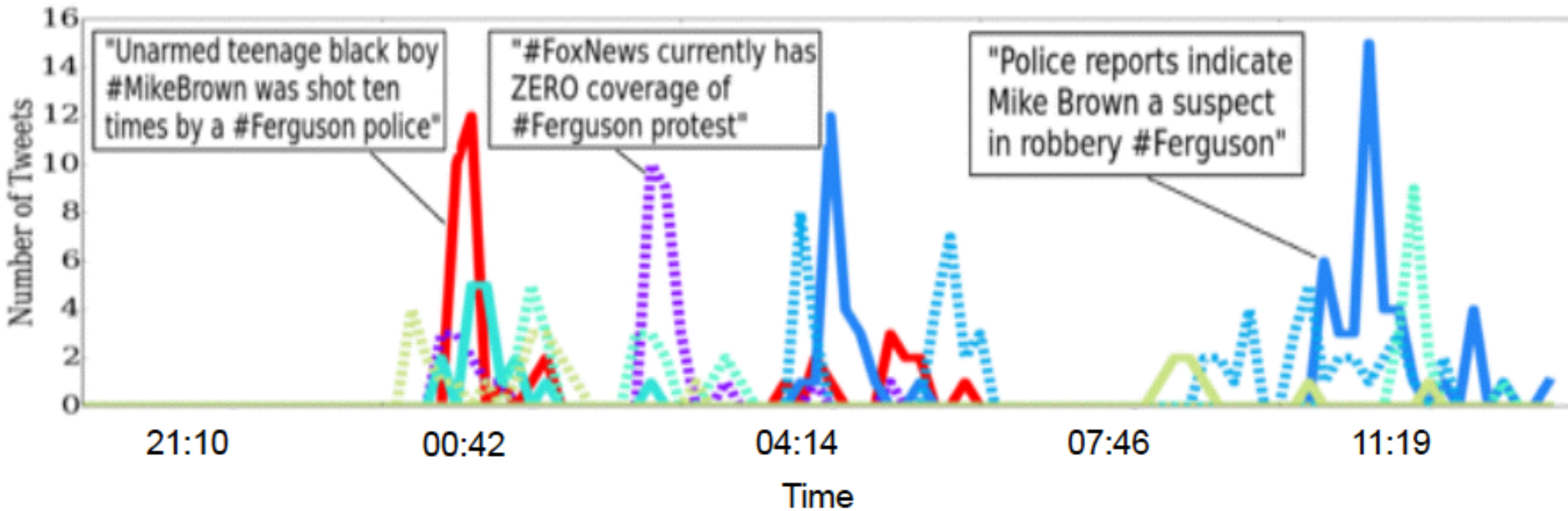
Modeling temporal dynamics in social media

Sydney Seige in 2014



Ferguson Unrest in 2014



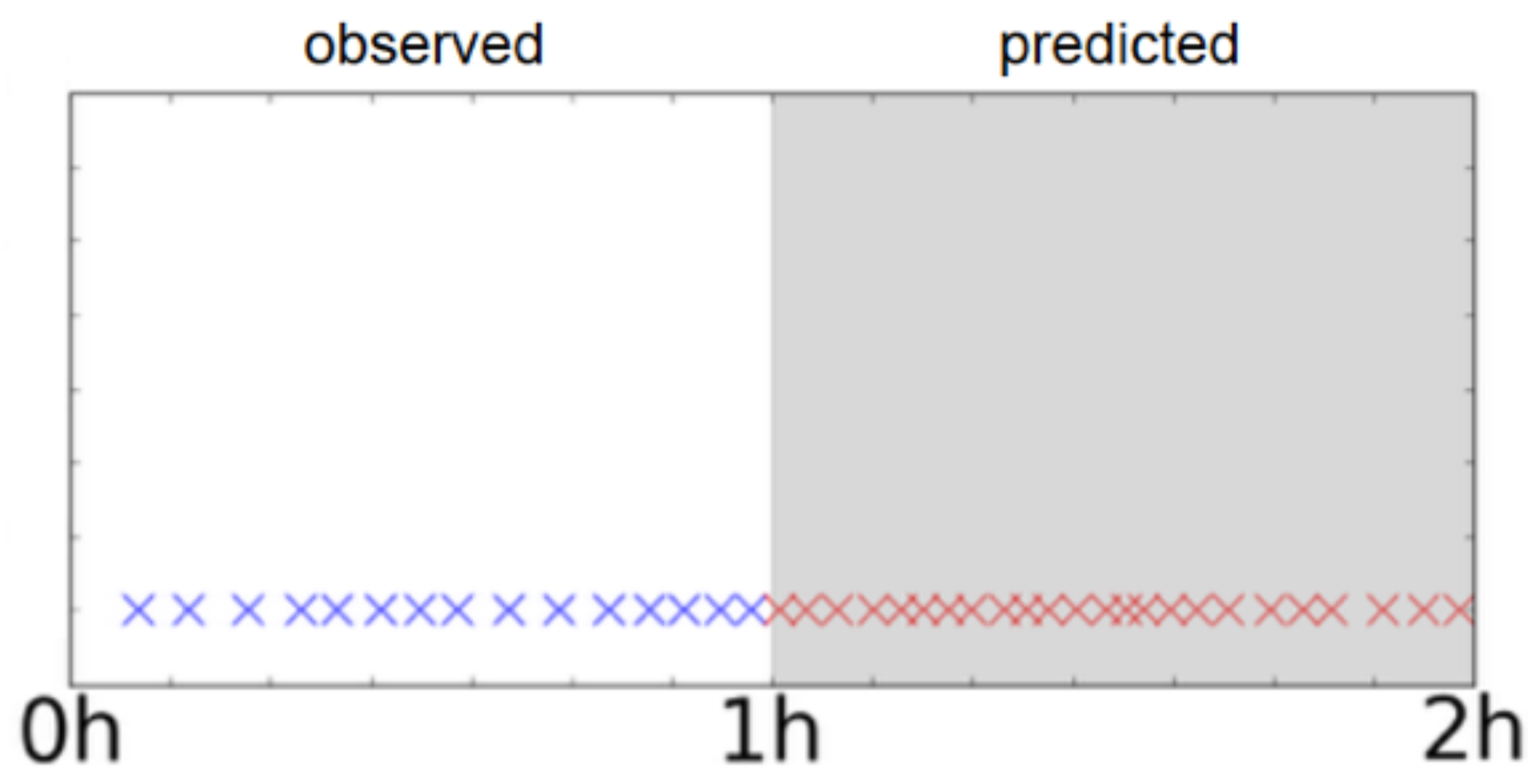
- Model the evolution of memes, activity of users and predicting the popularity of memes.

Temporal dynamics : Predicting popularity

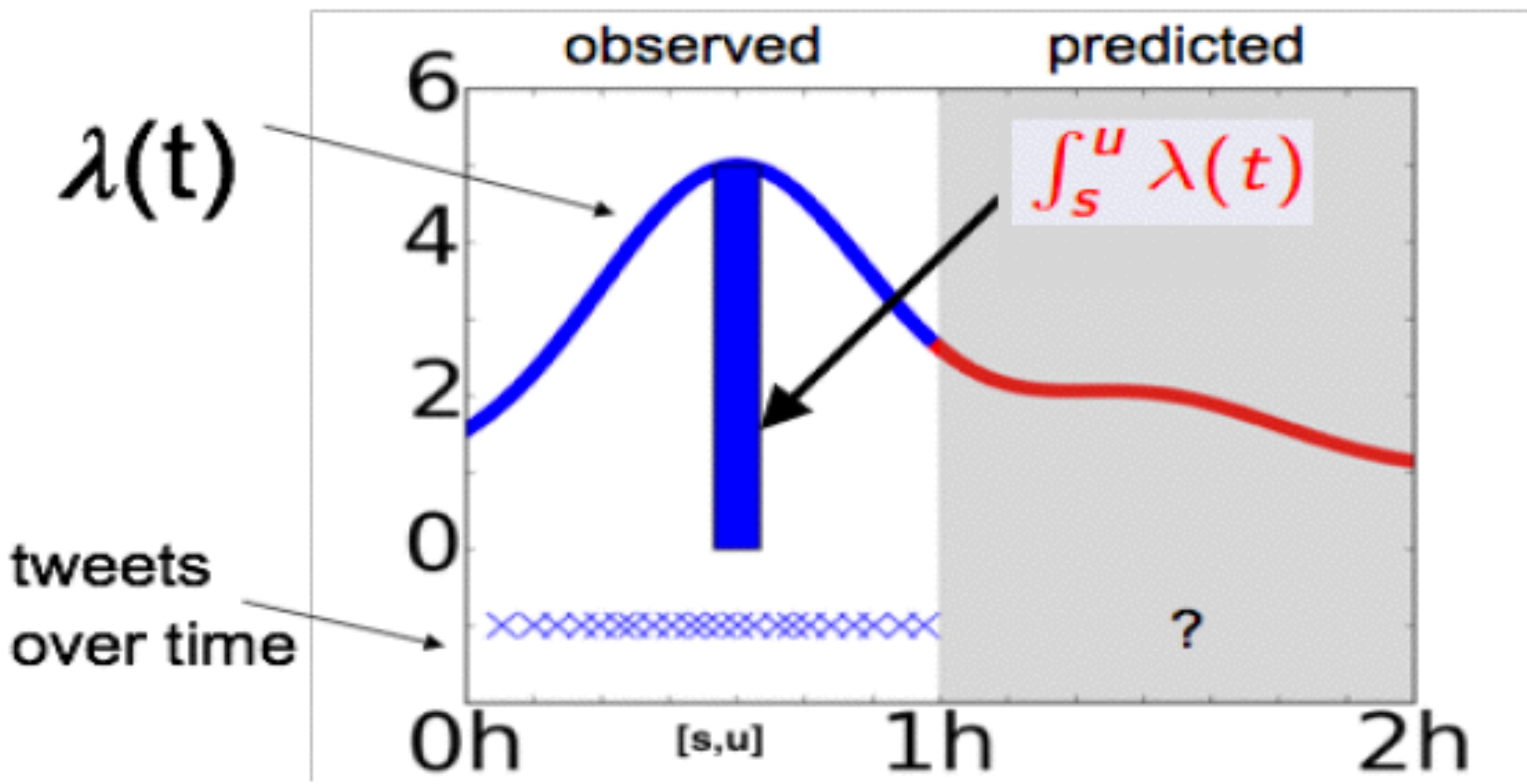
- Riots

- Politics

- Business

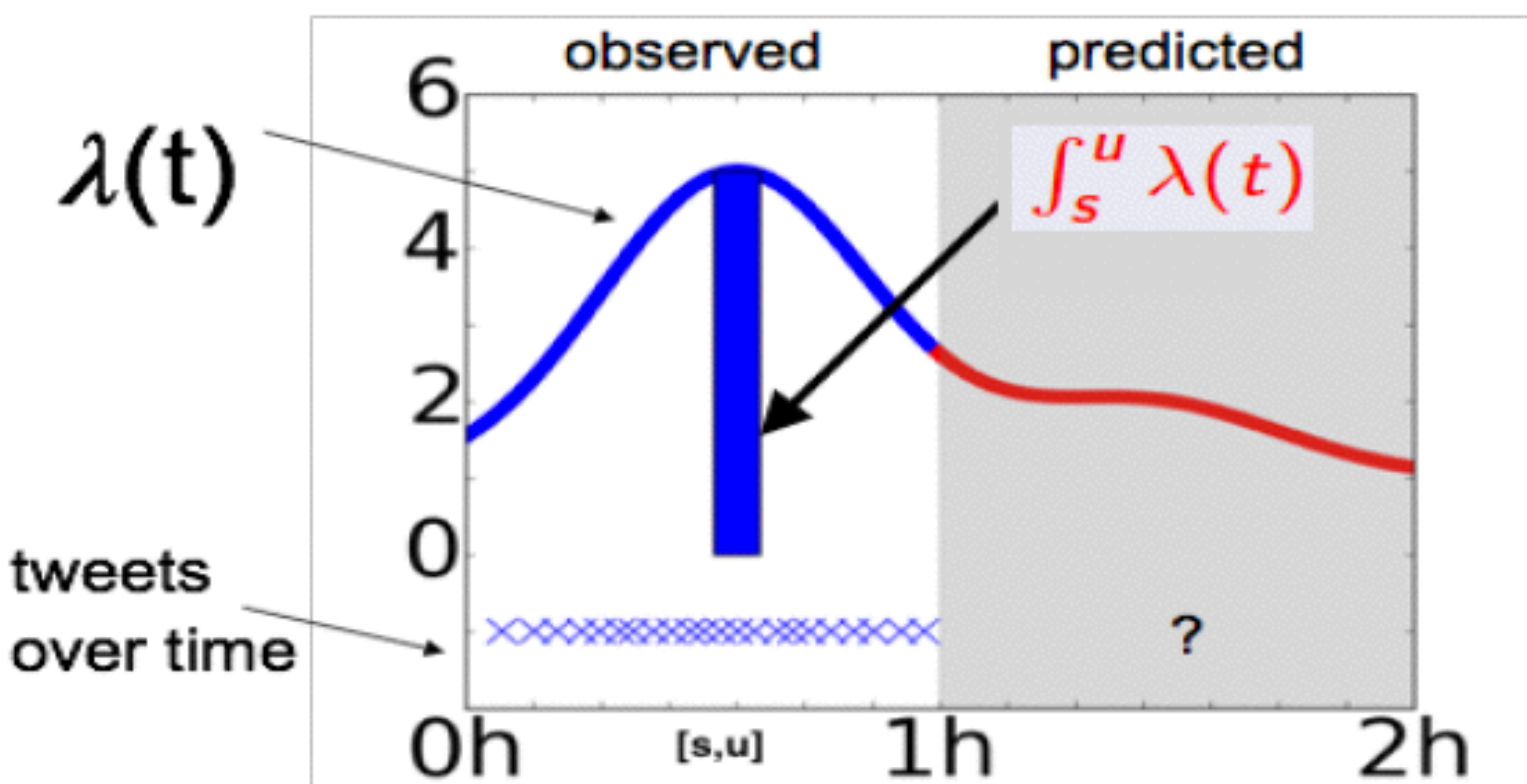


Modelling the occurrence of tweets over time : A point process approach

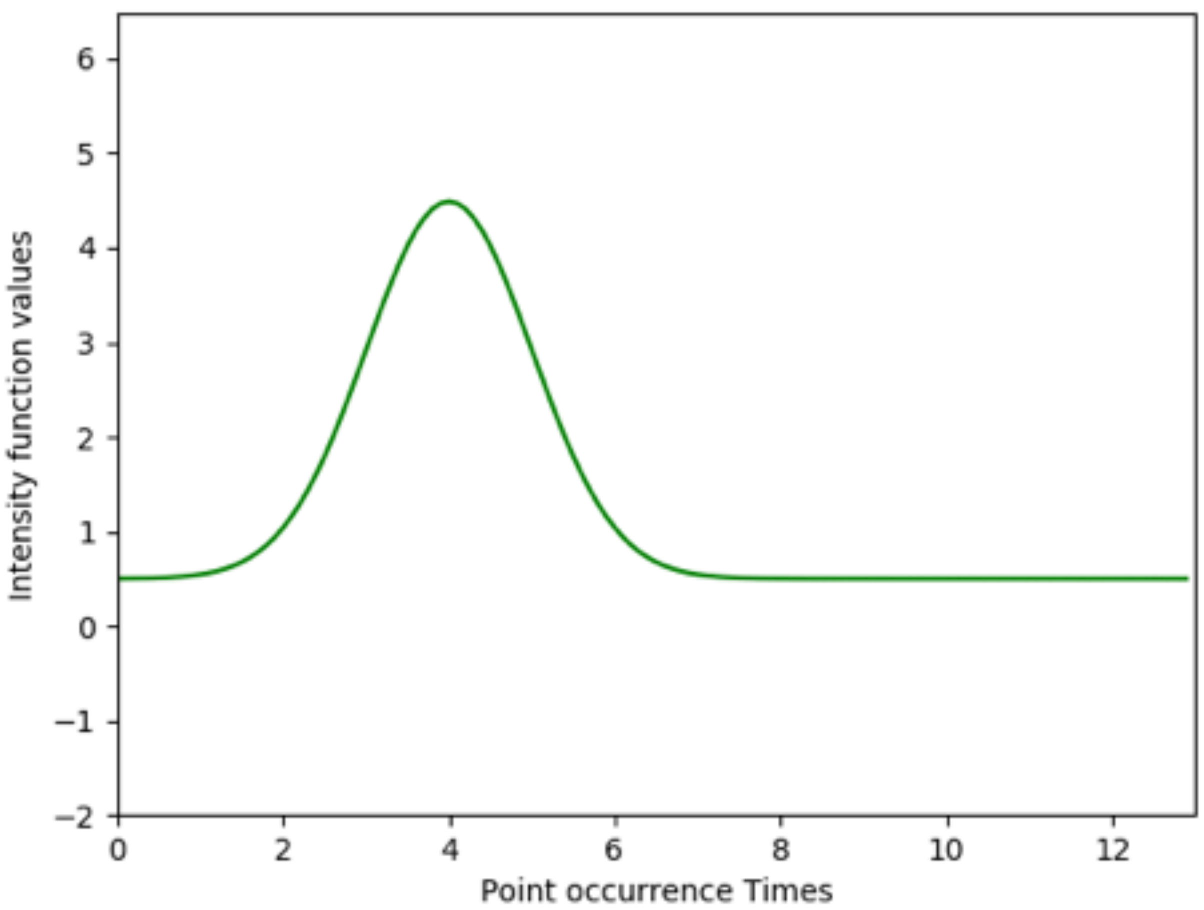


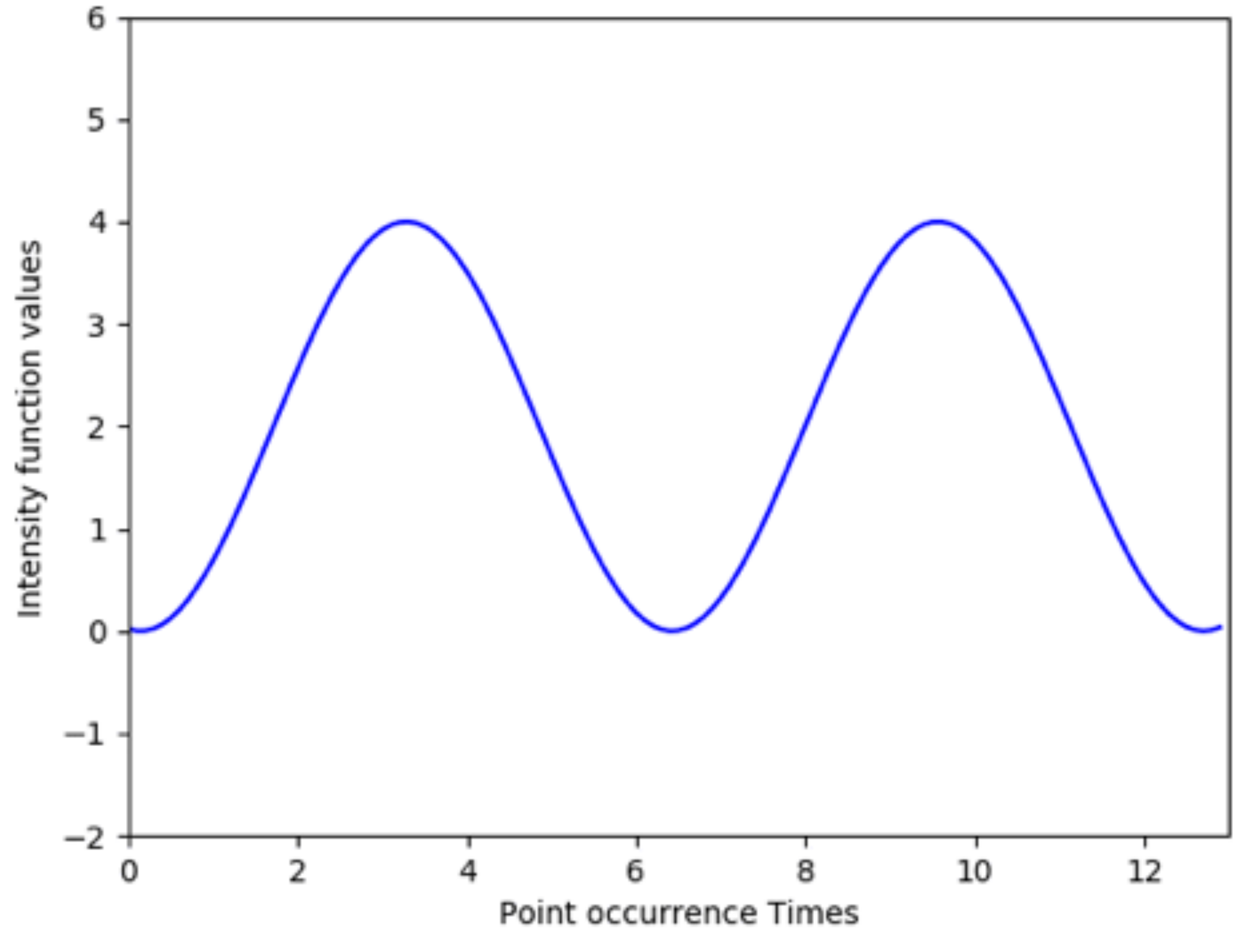
Modelling the occurrence of tweets over time : A point process approach

Likelihood $\lambda(u) \times \exp(-\int_s^u \lambda(t)dt)$
inst. prob. × survival probability



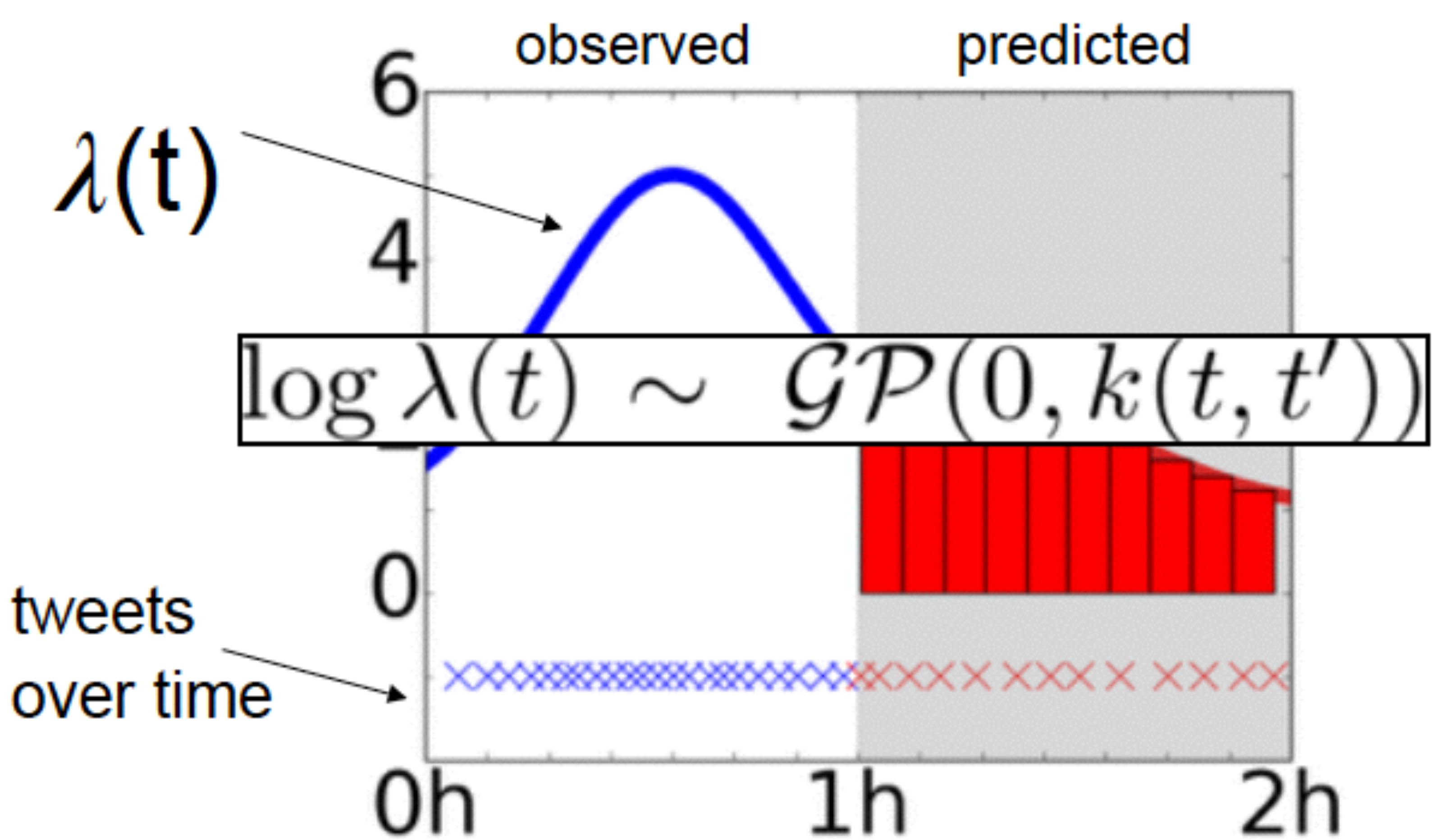
Sampling points from a point process





Modelling Tweet occurrence : Log-Gaussian Cox Process

- Doubly stochastic point process.
- Bayesian non-parametric approach to learning the intensity function.
- Log-Intensity function is sampled from a Gaussian process prior.



$$\log \lambda(t) \sim \mathcal{GP}(0, k(t, t'))$$

Bayesian Learning

$$P(\text{hypothesis}|\text{data}) = \frac{P(\text{data}|\text{hypothesis})P(\text{hypothesis})}{P(\text{data})}$$

Rev'd Thomas Bayes (1702–1761)

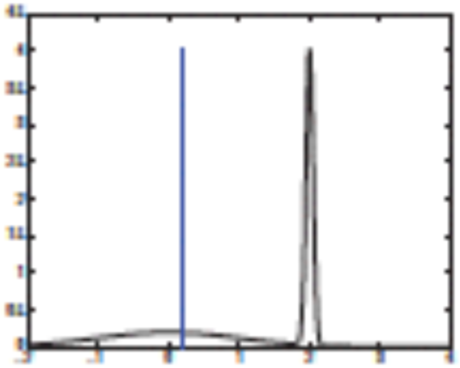
- Bayes rule tells us how to do inference about hypotheses from data.
- Learning and prediction can be seen as forms of inference.

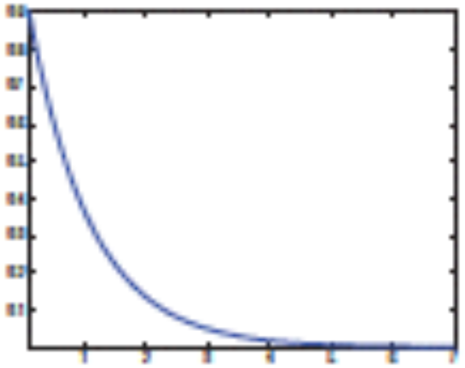
$$P(\theta|\mathcal{D}, m) = \frac{P(\mathcal{D}|\theta, m)P(\theta|m)}{P(\mathcal{D}|m)}$$

$P(\mathcal{D}\|\theta, m)$	likelihood of parameters θ in model m
$P(\theta\|m)$	prior probability of θ
$P(\theta\|\mathcal{D}, m)$	posterior of θ given data $\mathcal{D}$

Prediction:

$$P(x|\mathcal{D}, m) = \int P(x|\theta, \mathcal{D}, m)P(\theta|\mathcal{D}, m)d\theta$$





Model Comparison:

$$P(m|\mathcal{D}) = \frac{P(\mathcal{D}|m)P(m)}{P(\mathcal{D})}$$

$$P(\mathcal{D}|m) = \int P(\mathcal{D}|\theta, m)P(\theta|m)\, d\theta$$

Supervised learning : Regression

Regression

Given Data $\mathbf{D} = (\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, $\mathbf{x}_i \in \mathcal{X} \subset \mathcal{R}^P$, $y_i \in \mathcal{Y} \subset \mathcal{R}$

- Learn $f : \mathcal{X} \to \mathcal{Y}$.
- Bayesian approach : Allows to encode prior belief over functions
- Parametric model : $f(\mathbf{x}) = \mathbf{w}.\mathbf{x}$

$$p(\mathbf{w}|\mathbf{D}) \propto p(\mathbf{D}|\mathbf{w}) \; p(\mathbf{w})$$

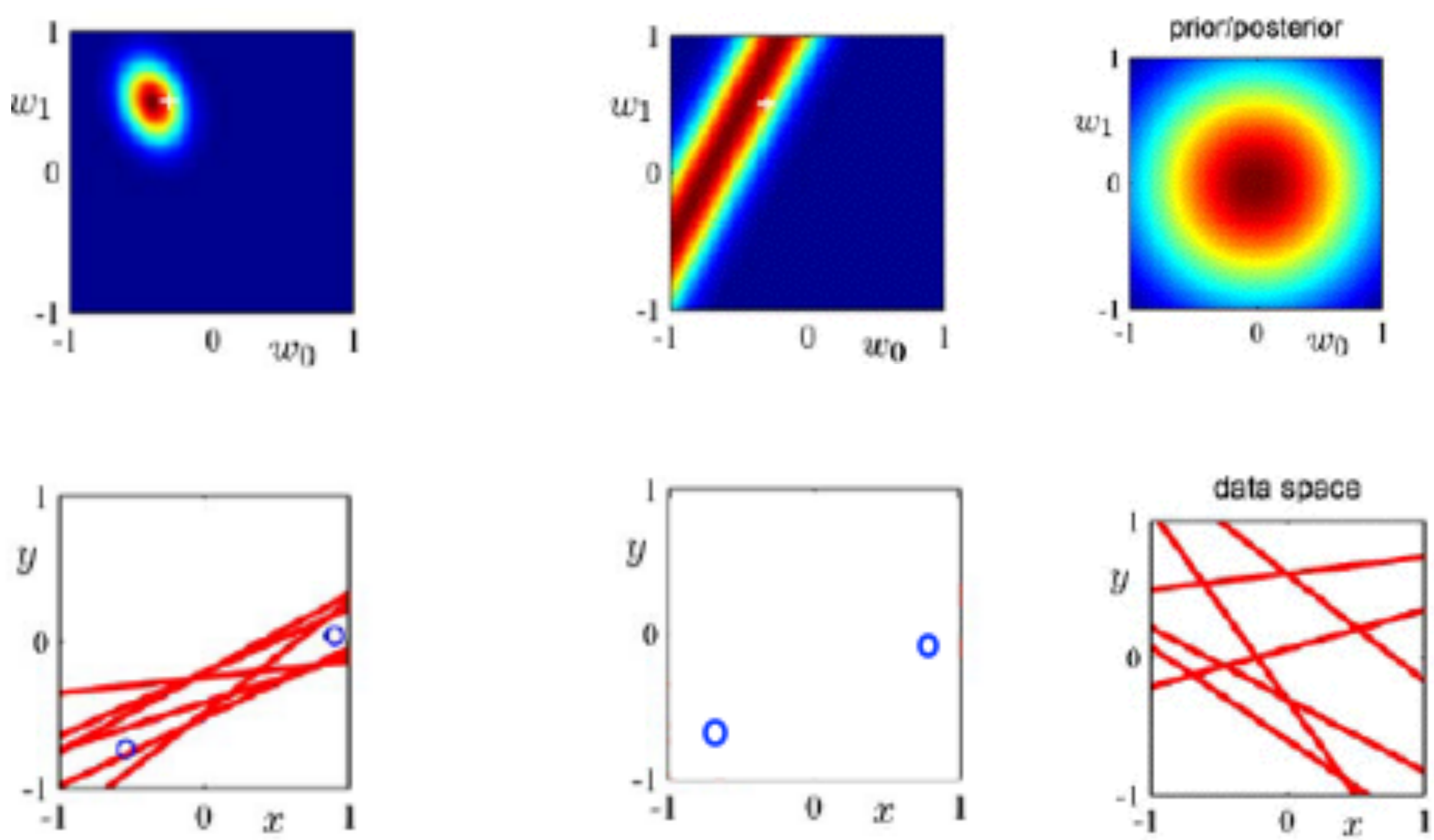
Posterior $\propto$ Likelihood Prior

Bayesian linear Regression

$$p(\mathbf{w}|D) \propto p(D|\mathbf{w}) \; p(\mathbf{w})$$

Posterior ∝ Likelihood Prior

Posterior **Likelihood** **Prior**

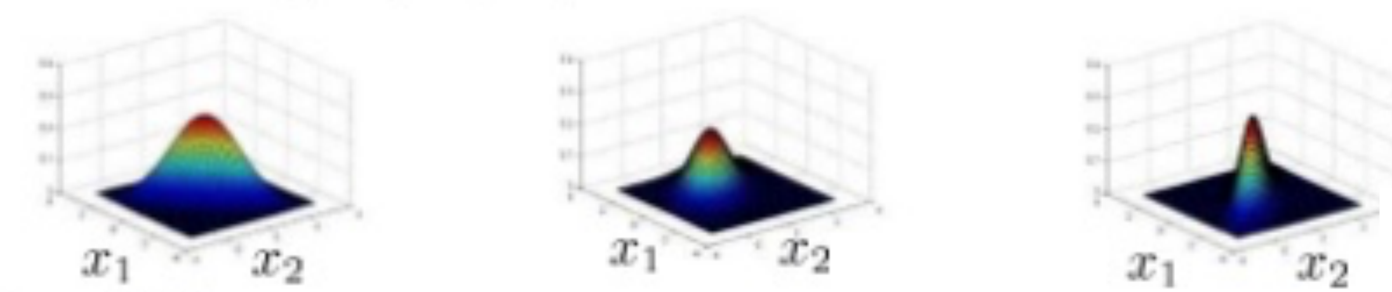


Gaussian distribution to Gaussian Process

- Functions are infinite dimensional objects.
- Gaussian processes define distribution over functions.

$$p(\mathbf{f}|\mathbf{D}) \propto p(\mathbf{D}|\mathbf{f}) \quad p(\mathbf{f})$$

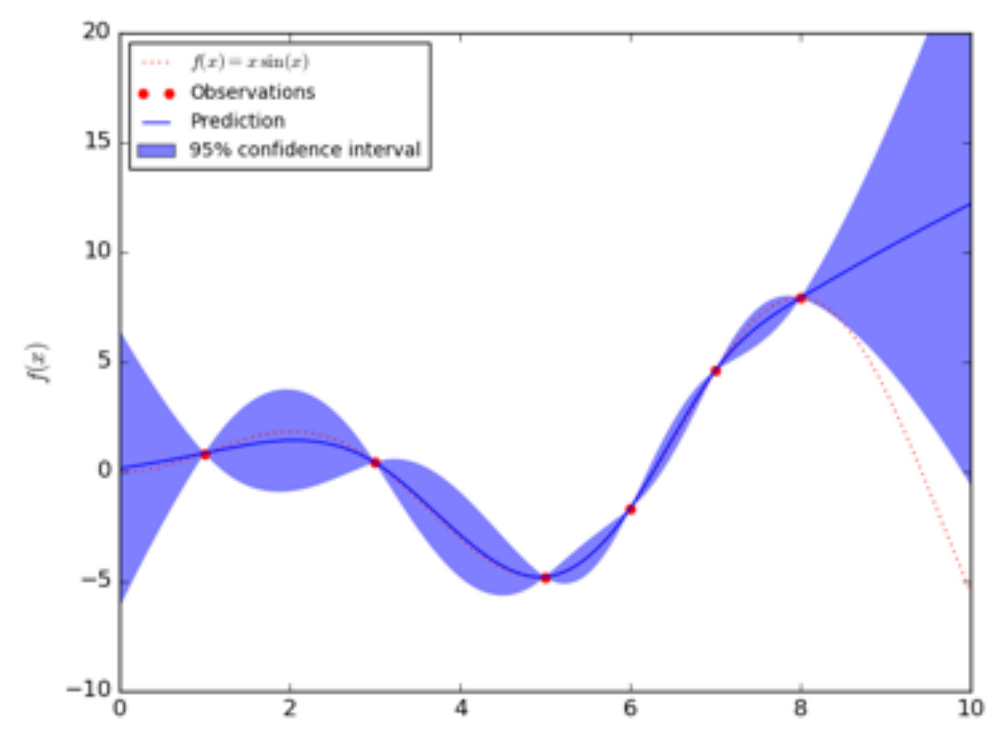
$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$



mean function

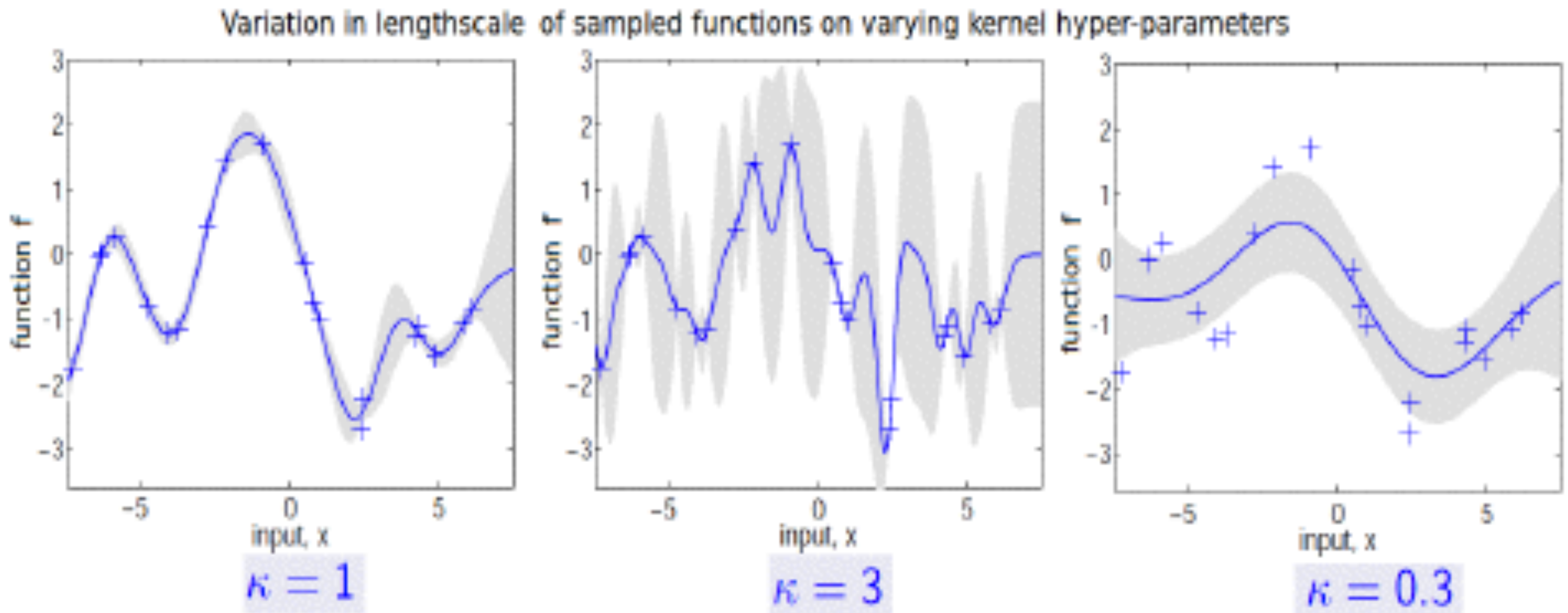
covariance function

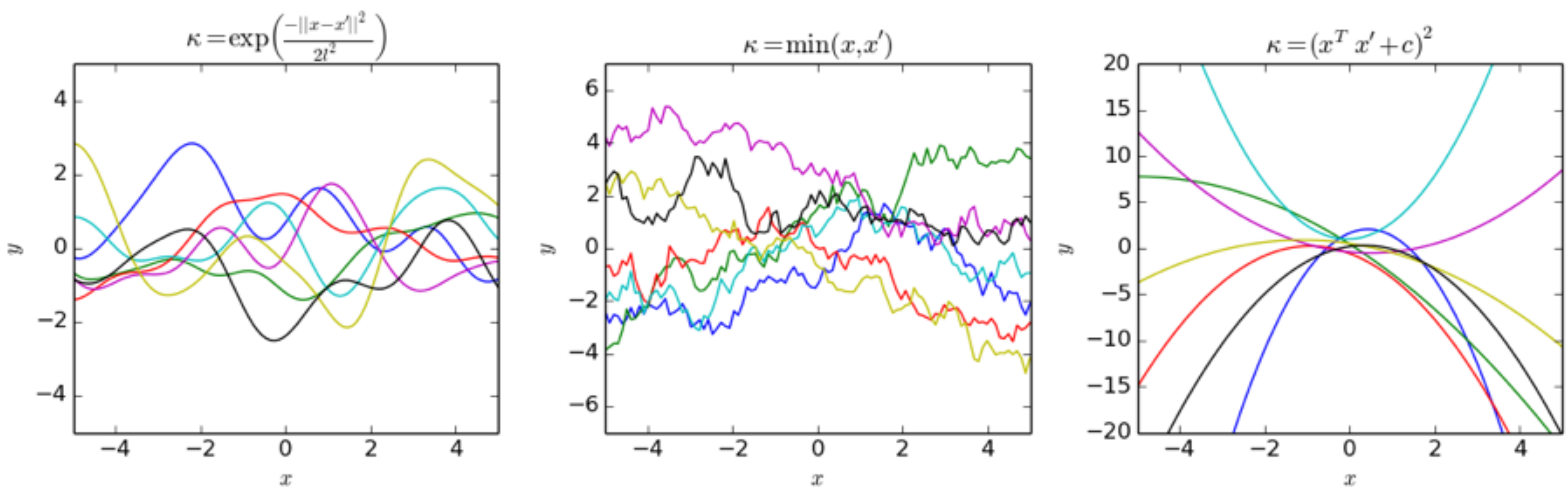
$$f(\mathrm{x}) \sim GP(m(\mathrm{x}), K(\mathrm{x},\mathrm{x}'))$$



Gaussian process Kernels

$$cov(f(\mathbf{x}_i), f(\mathbf{x}_j)) = K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\kappa}{2}||\mathbf{x}_i - \mathbf{x}_j||^2)$$





GP Posterior from GP Prior

Regression : Output is real and scalar, $y \in \mathcal{R}$, $y = f(\mathbf{x}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$

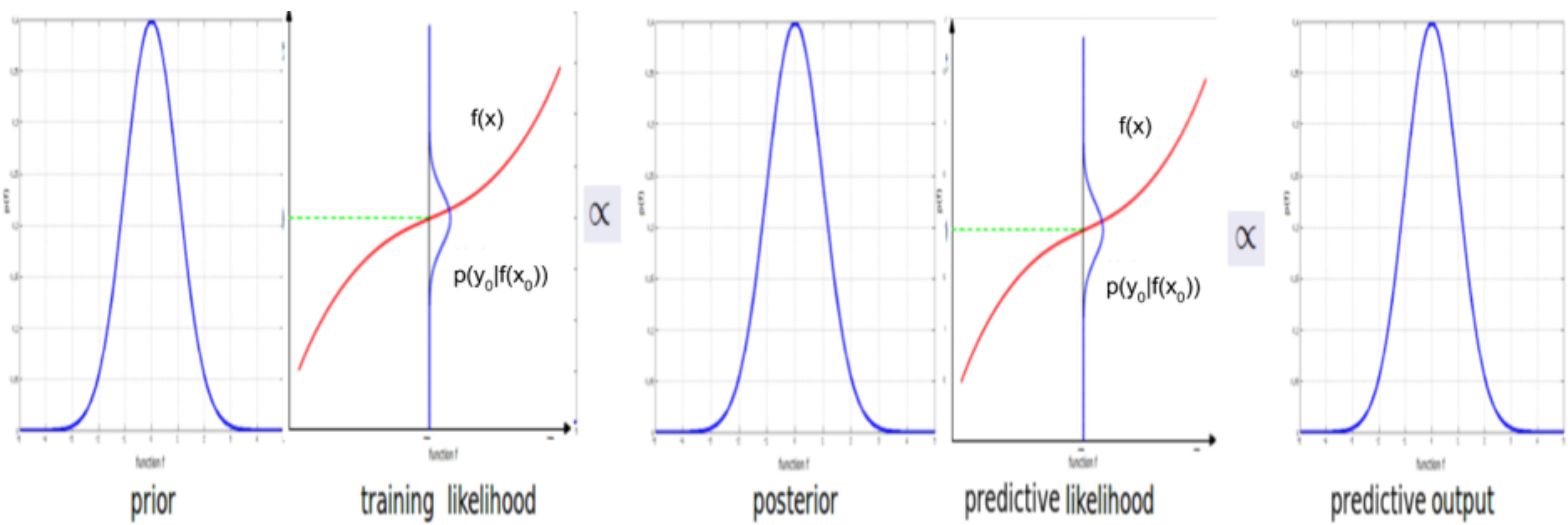
$f(\mathbf{X}) \sim \mathcal{N}(0, K(\mathbf{X}, \mathbf{X}))$ Gaussian

$p(\mathbf{f}|\mathbf{X}, \mathbf{y}) = \frac{1}{p(\mathbf{y}|\mathbf{X})} p(\mathbf{f}|\mathbf{X}) p(\mathbf{y}|\mathbf{f}, \mathbf{X})$ Gaussian

$p(y_i|f(\mathbf{x}_i)) = \mathcal{N}(y_i; f(\mathbf{x}_i), \sigma_n^2)$ Gaussian

$p(\mathbf{f}_*|\mathbf{X}, \mathbf{X}_*, \mathbf{y}) = \int p(\mathbf{f}_*|\mathbf{f}, \mathbf{X}, \mathbf{X}_*) p(\mathbf{f}|\mathbf{X}, \mathbf{y}) d\mathbf{f}$ Gaussian

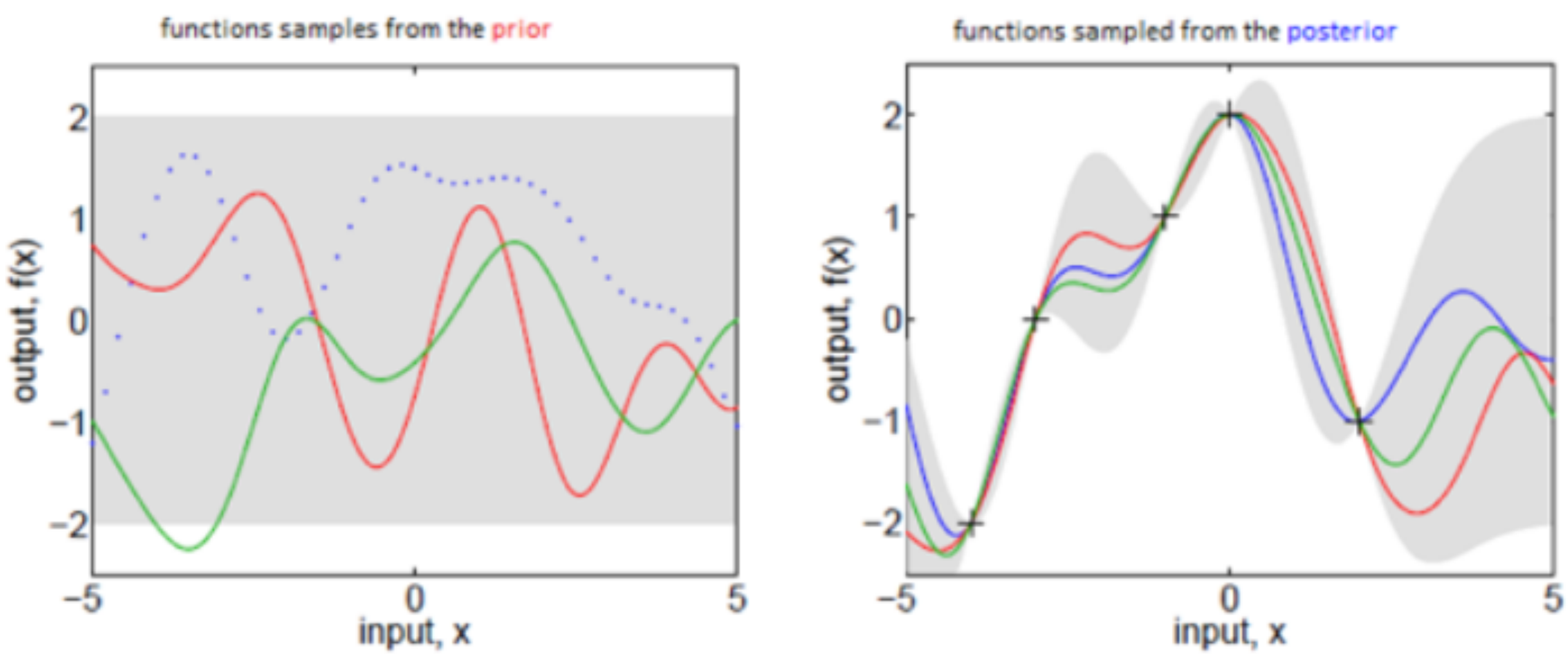
$p(\mathbf{y}_*|\mathbf{X}, \mathbf{X}_*, \mathbf{y}) = \int p(\mathbf{y}_*|\mathbf{f}_*) p(\mathbf{f}_*|\mathbf{X}, \mathbf{X}_*, \mathbf{y}) d\mathbf{f}_*$ Gaussian



Gaussian Process Prediction and Learning

$$\mathbf{f}_*|X,\mathbf{y},X_* \sim \mathcal{N}(\bar{\mathbf{f}}_*, \mathrm{cov}(\mathbf{f}_*)), \quad \text{where}$$
$$\bar{\mathbf{f}}_* \triangleq \mathbb{E}[\mathbf{f}_*|X,\mathbf{y},X_*] = K(X_*,X)[K(X,X)+\sigma_n^2 I]^{-1}\mathbf{y},$$
$$\mathrm{cov}(\mathbf{f}_*) = K(X_*,X_*) - K(X_*,X)[K(X,X)+\sigma_n^2 I]^{-1}K(X,X_*).$$

$$\bar{f}(\mathbf{x}_*) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}_*)$$
$$\alpha = (K+\sigma_n^2 I)^{-1}\mathbf{y}.$$



- Learn model parameters $\boldsymbol{\theta} = (\kappa, \sigma_n^2)$ by maximizing $p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta}) = \int p(\mathbf{y}|\mathbf{f},\boldsymbol{\theta})p(\mathbf{f}|\mathbf{X},\boldsymbol{\theta})d\mathbf{f}$

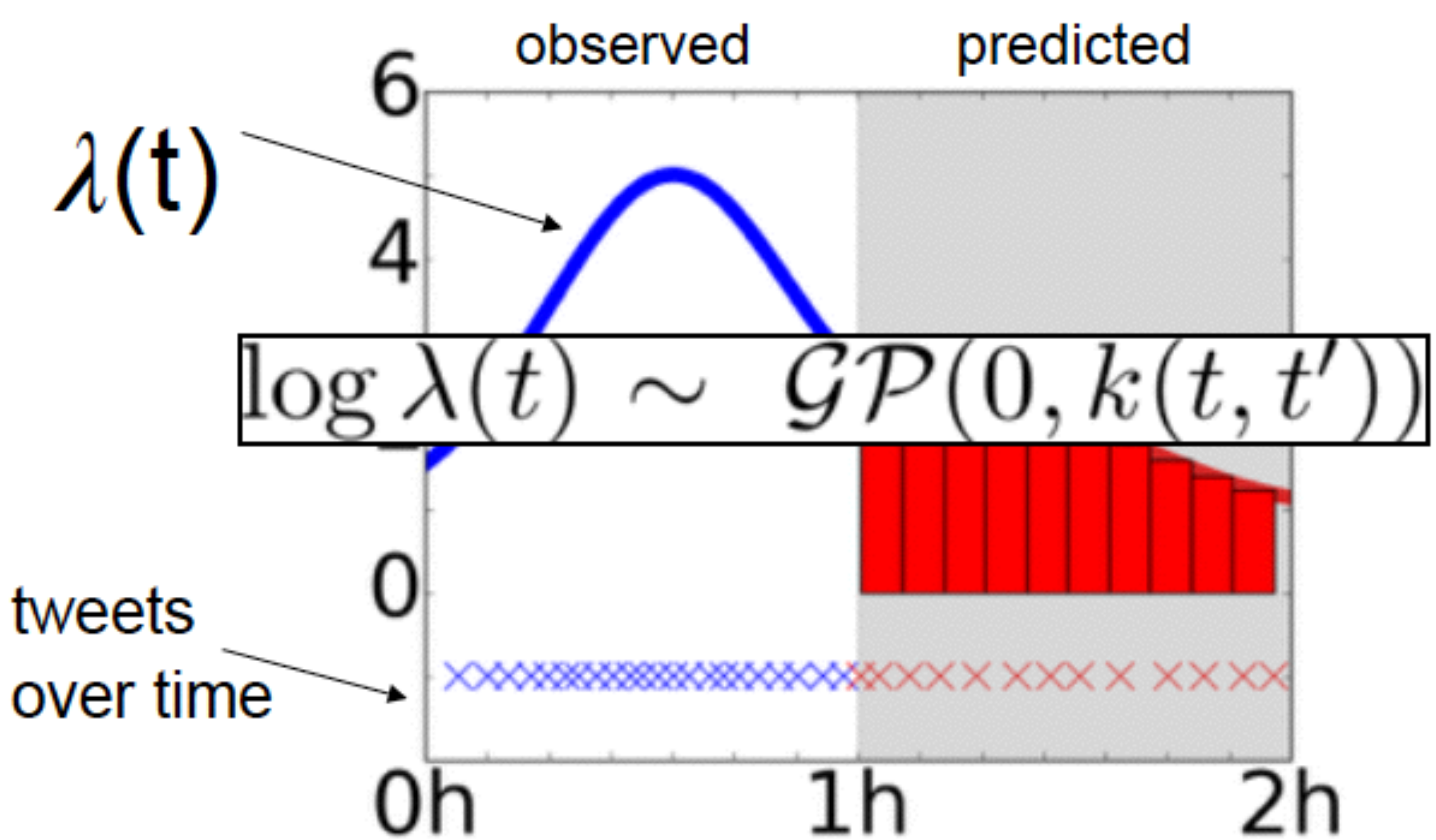
$$p(\mathbf{f}|\mathbf{X},\mathbf{y}) = \frac{1}{p(\mathbf{y}|\mathbf{X})}p(\mathbf{f}|\mathbf{X})p(\mathbf{y}|\mathbf{f},\mathbf{X})$$

Marginal Likelihood

Gaussian

Modelling Tweet occurrence : Log-Gaussian Cox Process

- Useful when the form of the intensity function is unknown
- Not sufficient data to learn the form
- Model the evolution of memes



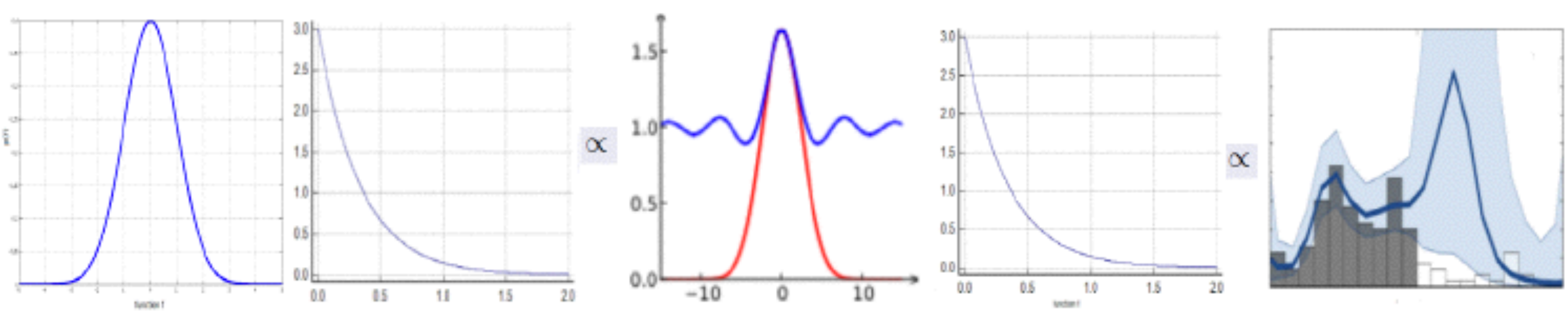
$$\log \lambda(t) \sim \mathcal{GP}(0, k(t, t'))$$

Modelling Twitter Dynamics

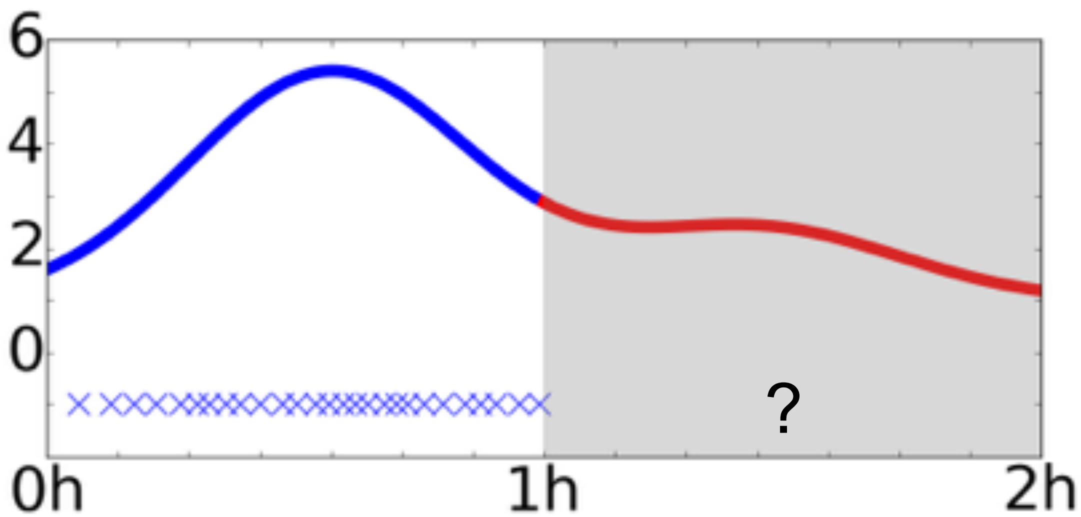
- Twitter data containing information tweet time, text, and meme category $\{d_n = (t_n, \boldsymbol{W}_n, m_n)\}_{n=1}^{N}$.

Log-Gaussian Cox Processes (LGCP)

- Prior intensity $\lambda_m(t)$: $\log \lambda_m(t) = f_m(t) \sim \mathcal{GP}(0, k_m(t, t'))$ (ensure positivity)
- Likelihood : $\prod_{j=1}^{N^m} \lambda_m(t_m) \exp(-\int \lambda_m(t) dt)$ not Gaussian !
- Posterior over latent function $f_m(t)$ obtained using Laplace approximation
- Predictive intensity $\lambda_m(t_*^m | D^m) = \int \exp\left(f_m(t_*^m)\right) p\left(f_m(t_*^m) | D^m\right) df_m(t_*^m)$



Modelling Twitter dynamics

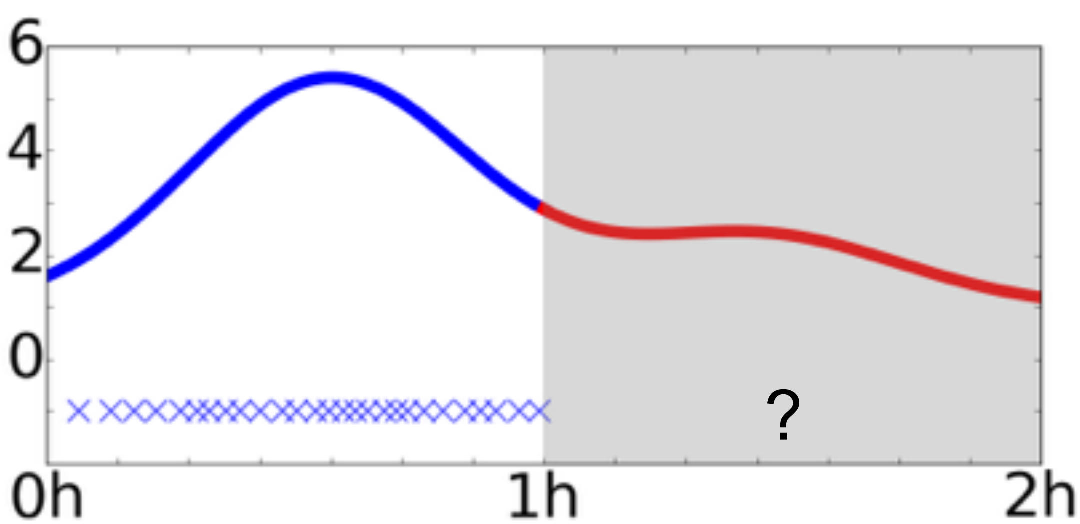


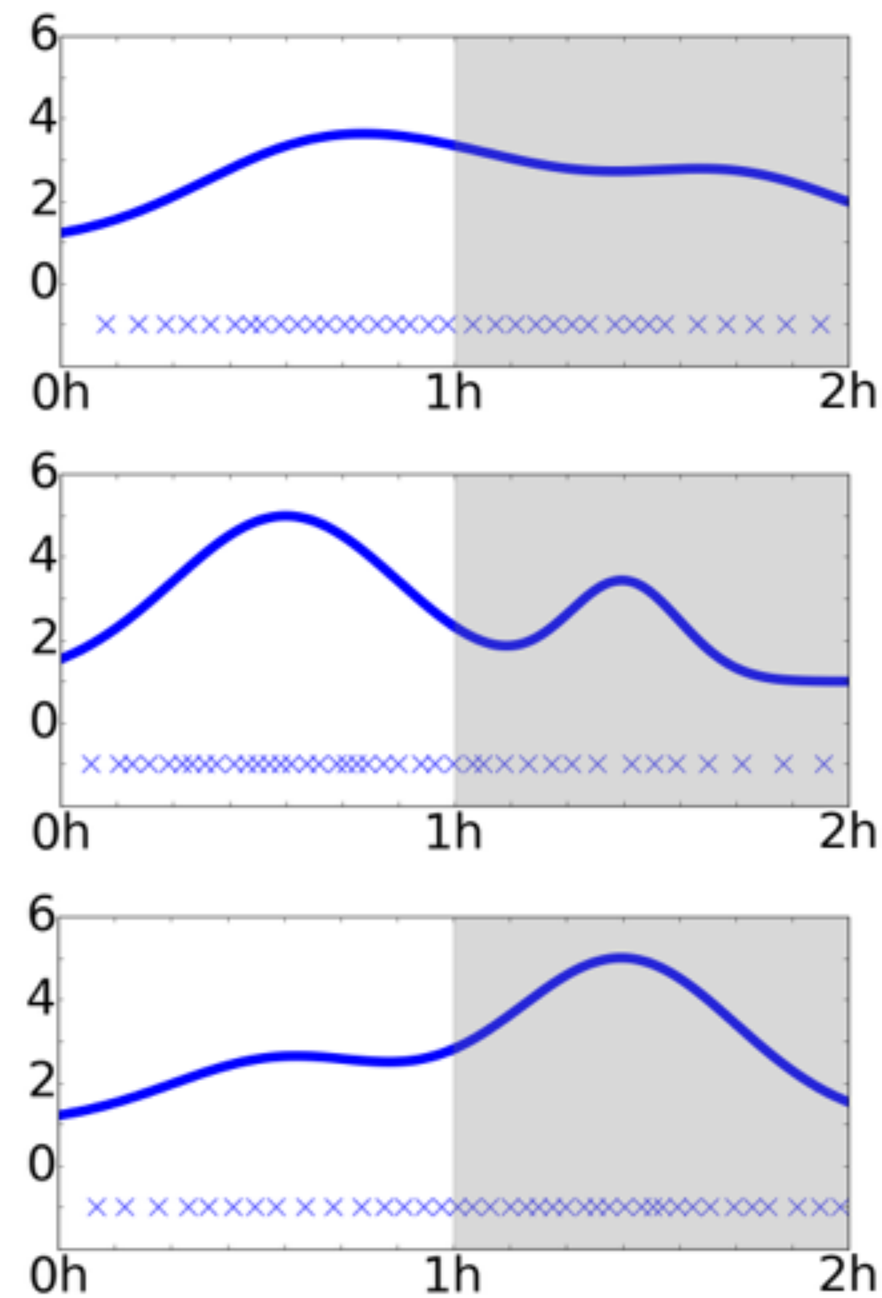
Problem: Small cascades

A hint for a solution:

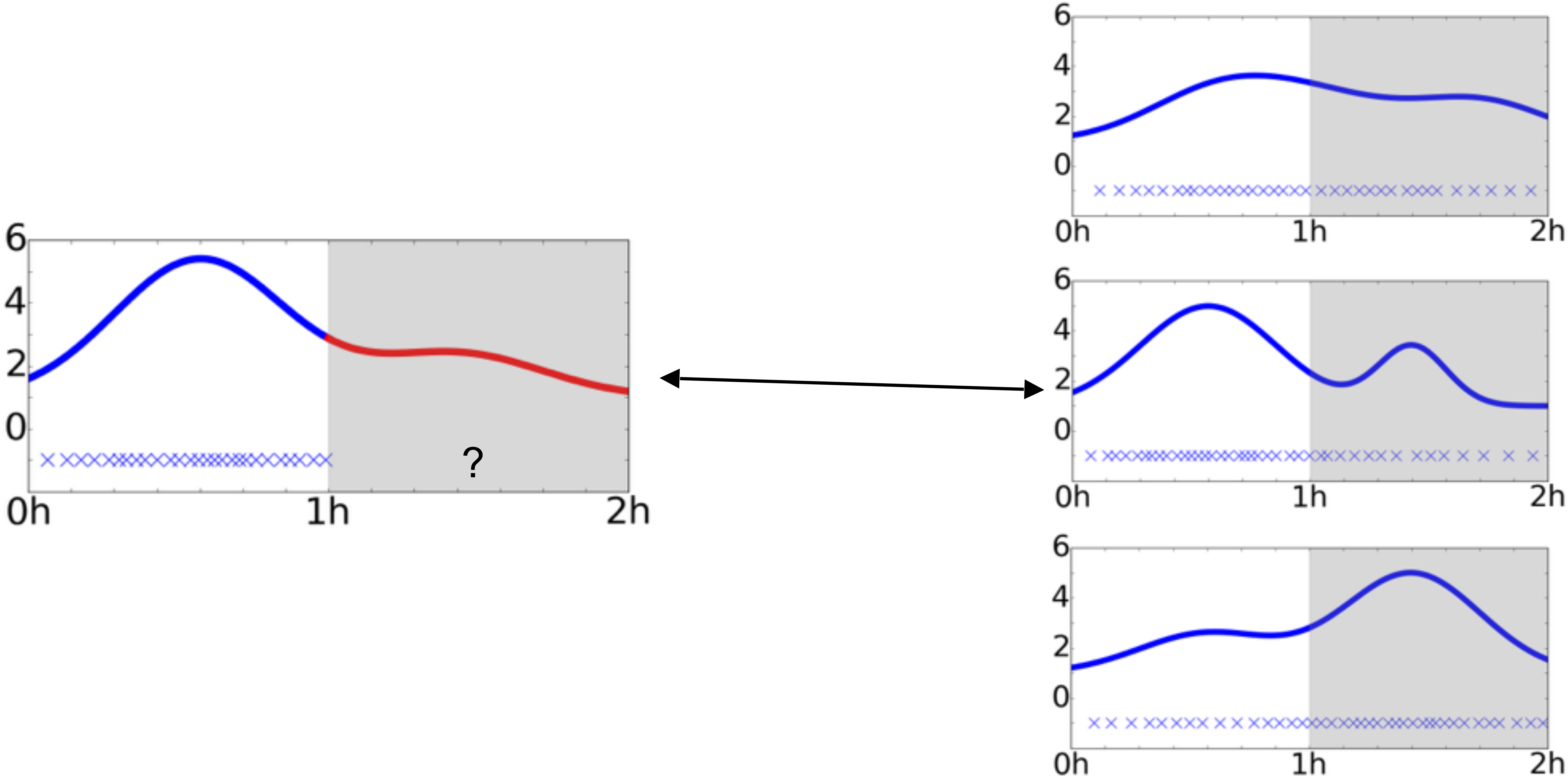
Similarity across temporal patterns across memes

Using reference memes

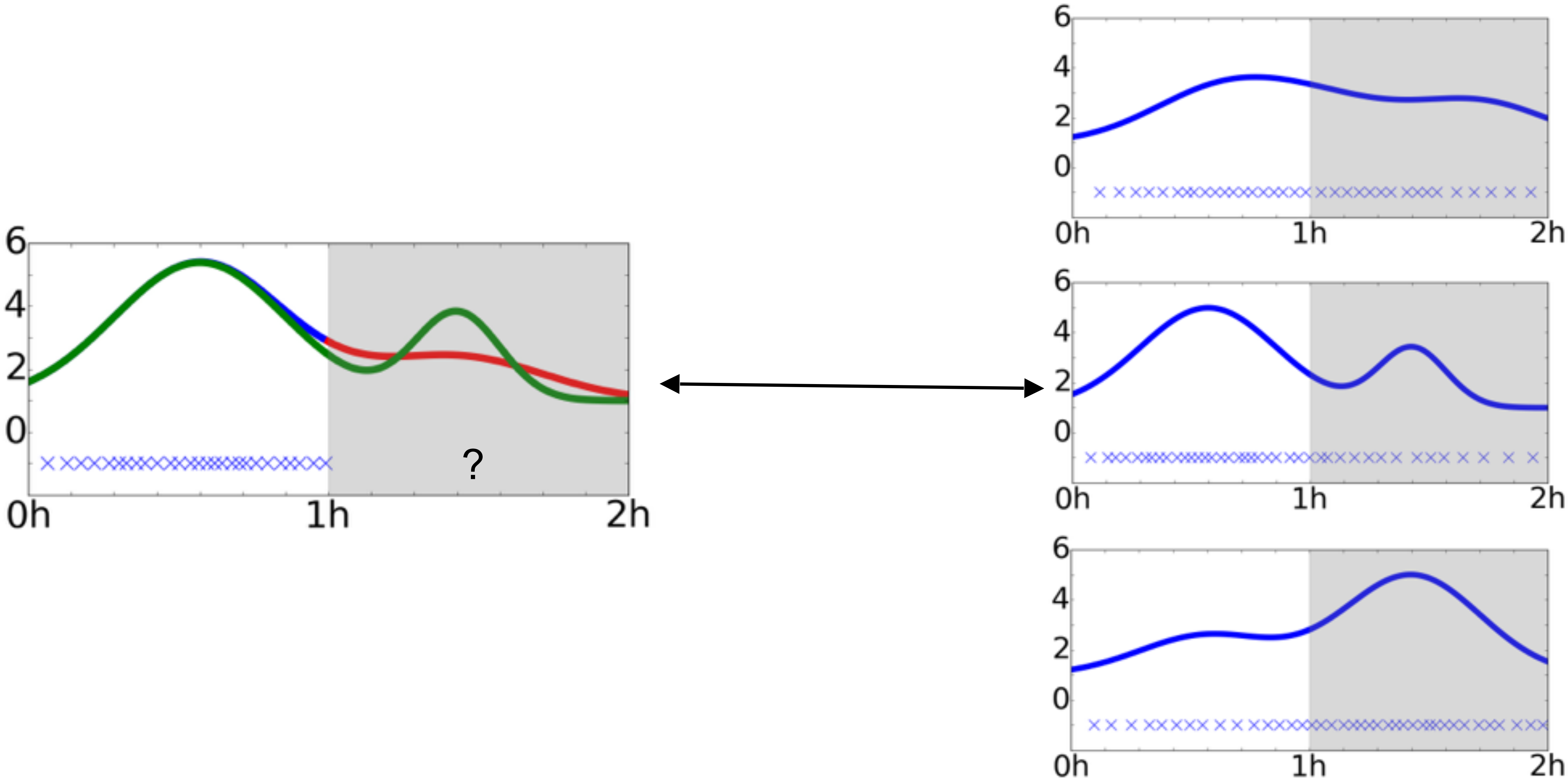




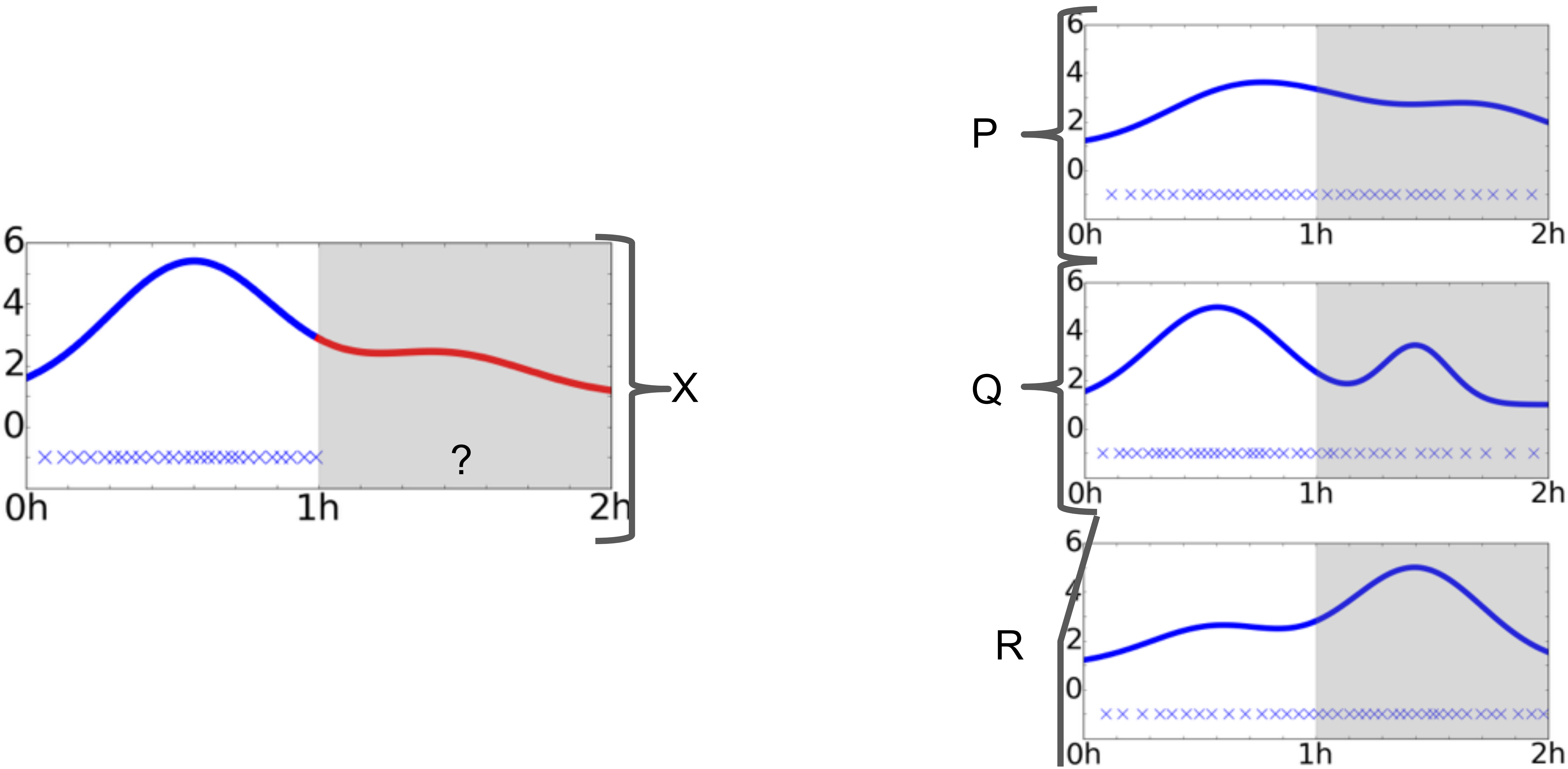
Using reference memes



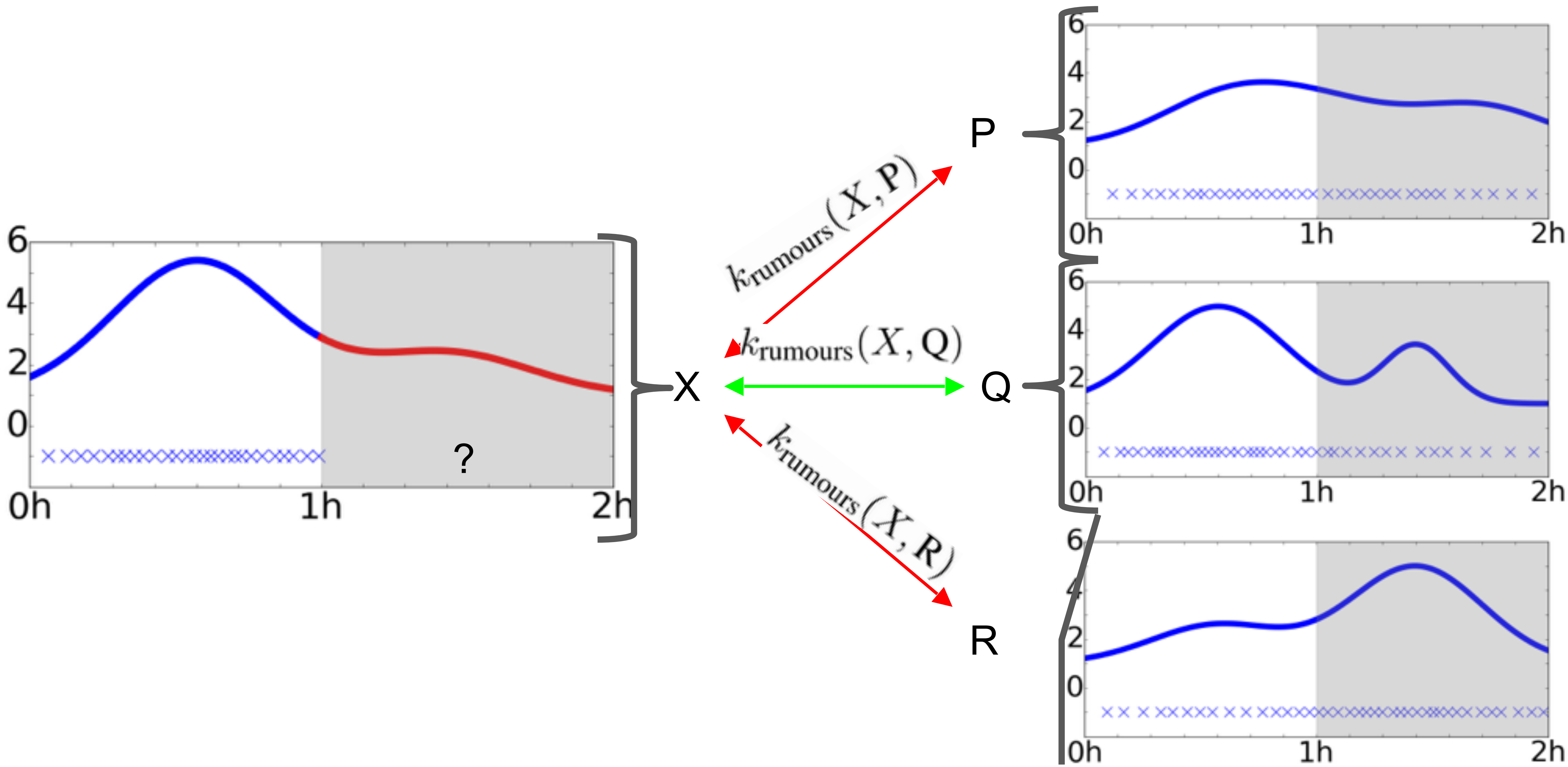
Using reference memes



Using reference memes

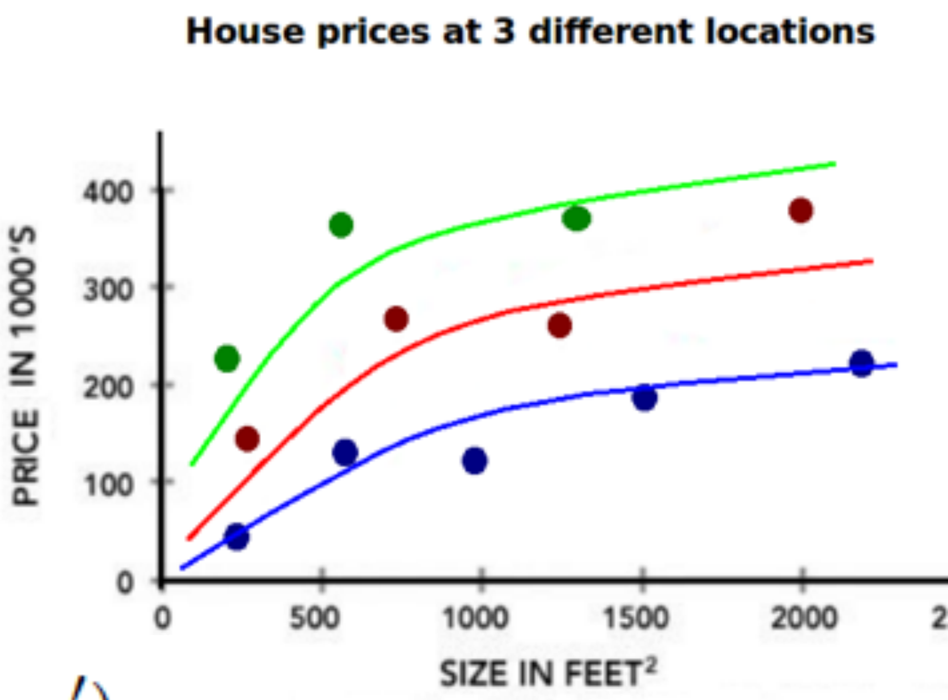


Using reference memes



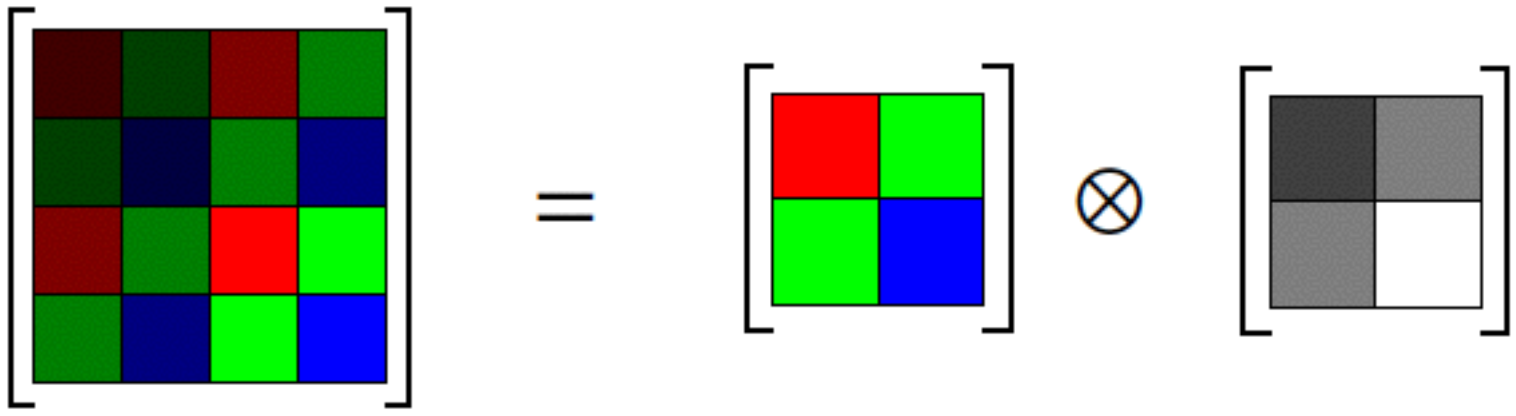
Multi-task learning

- Several related tasks sharing a common data representation
- Use multi-task Gaussian processes
  - Captures similarities between the tasks using kernels
  - Uses information from other tasks to make predictions



$$y_{il} \sim \mathcal{N}(f_l(\mathbf{x}_i), \sigma_l^2), \qquad \langle f_l(\mathbf{x}) f_k(\mathbf{x}') \rangle = K^f_{lk} k^x(\mathbf{x}, \mathbf{x}')$$

$$\bar{f}_l(\mathbf{x}_*) = (\mathbf{k}_l^f \otimes \mathbf{k}_*^x)^T \Sigma^{-1} \mathbf{y} \qquad \Sigma = K^f \otimes K^x + D \otimes I$$

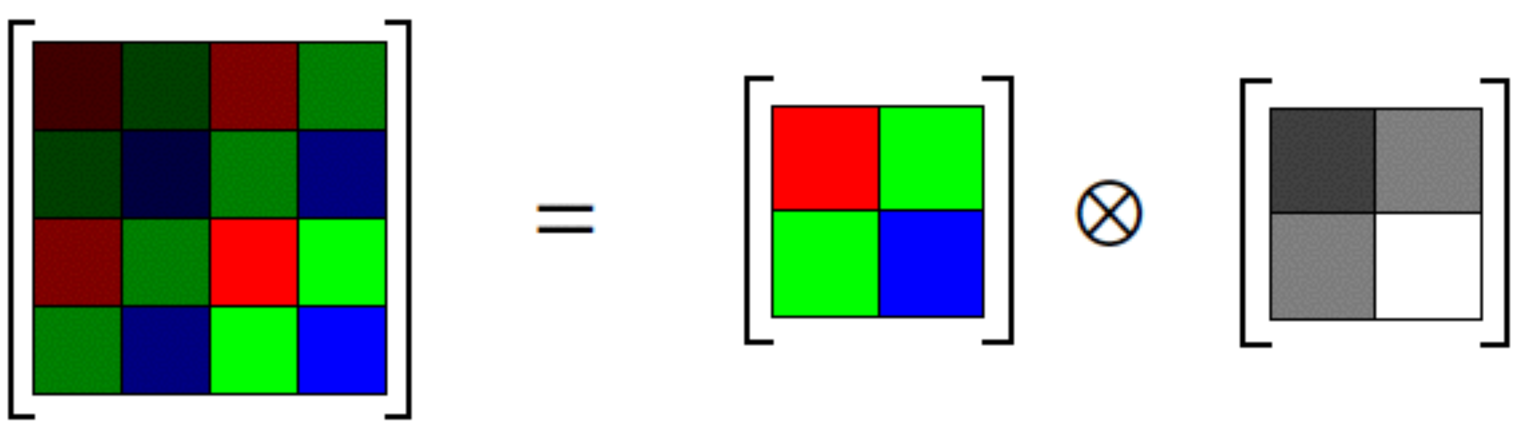


Multi-task learning

- Several related tasks sharing a common data representation
- Use multi-task Gaussian processes
  - Captures similarities between the memes using kernels

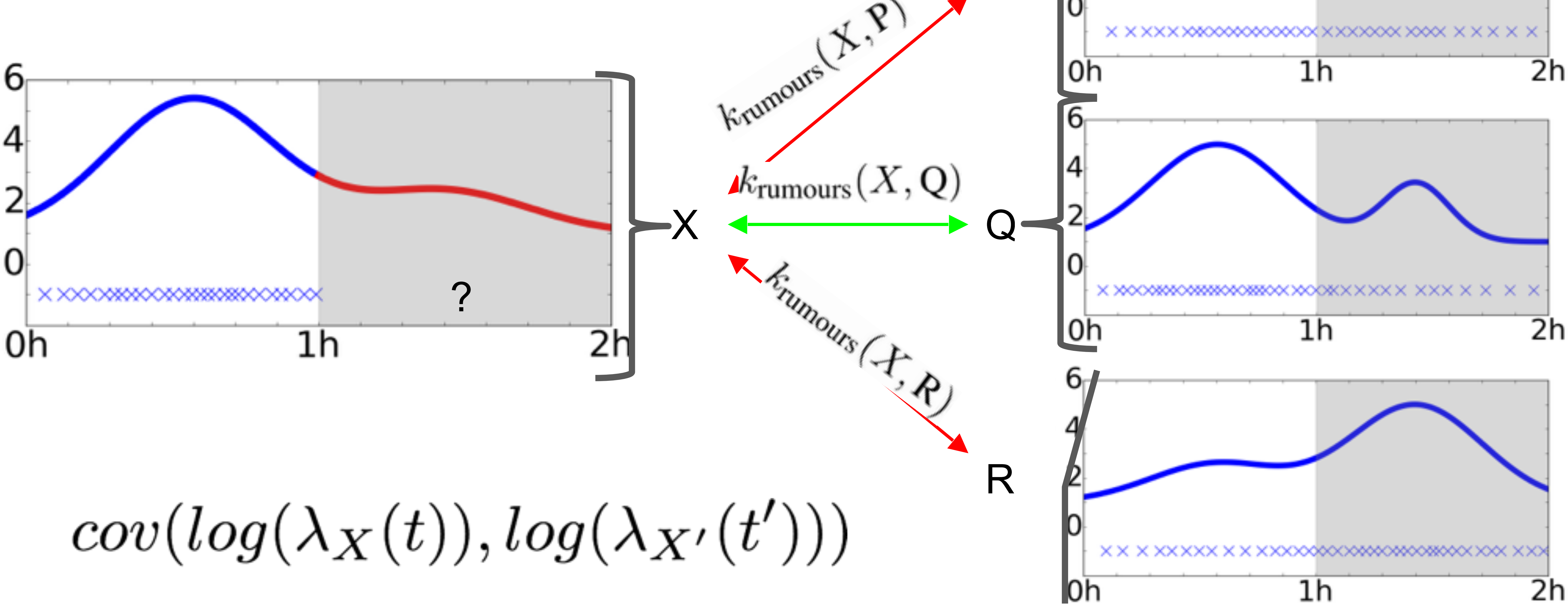
$$k_{\text{time x rumours}}((t, X), (t', X')) = k_{\text{time}}(t, t') \times k_{\text{rumours}}(X, X')$$

$$\begin{aligned} k_{\text{time x corr}}((t, i), (t', i')) &= \\ k_{\text{time}}(t, t') \times k_{\text{corr}}(i, i') &= \\ k_{\text{time}}(t, t') \times B_{ii'} & \end{aligned}$$



Multitask learning of meme intensities

$$k_{\text{time x rumours}}((t, X), (t', X')) = k_{\text{time}}(t, t') \times k_{\text{rumours}}(X, X')$$

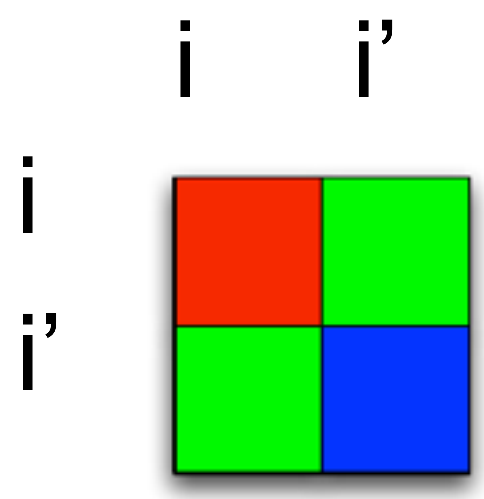


$$cov(log(\lambda_X(t)), log(\lambda_{X'}(t')))$$

Using reference rumours: correlations

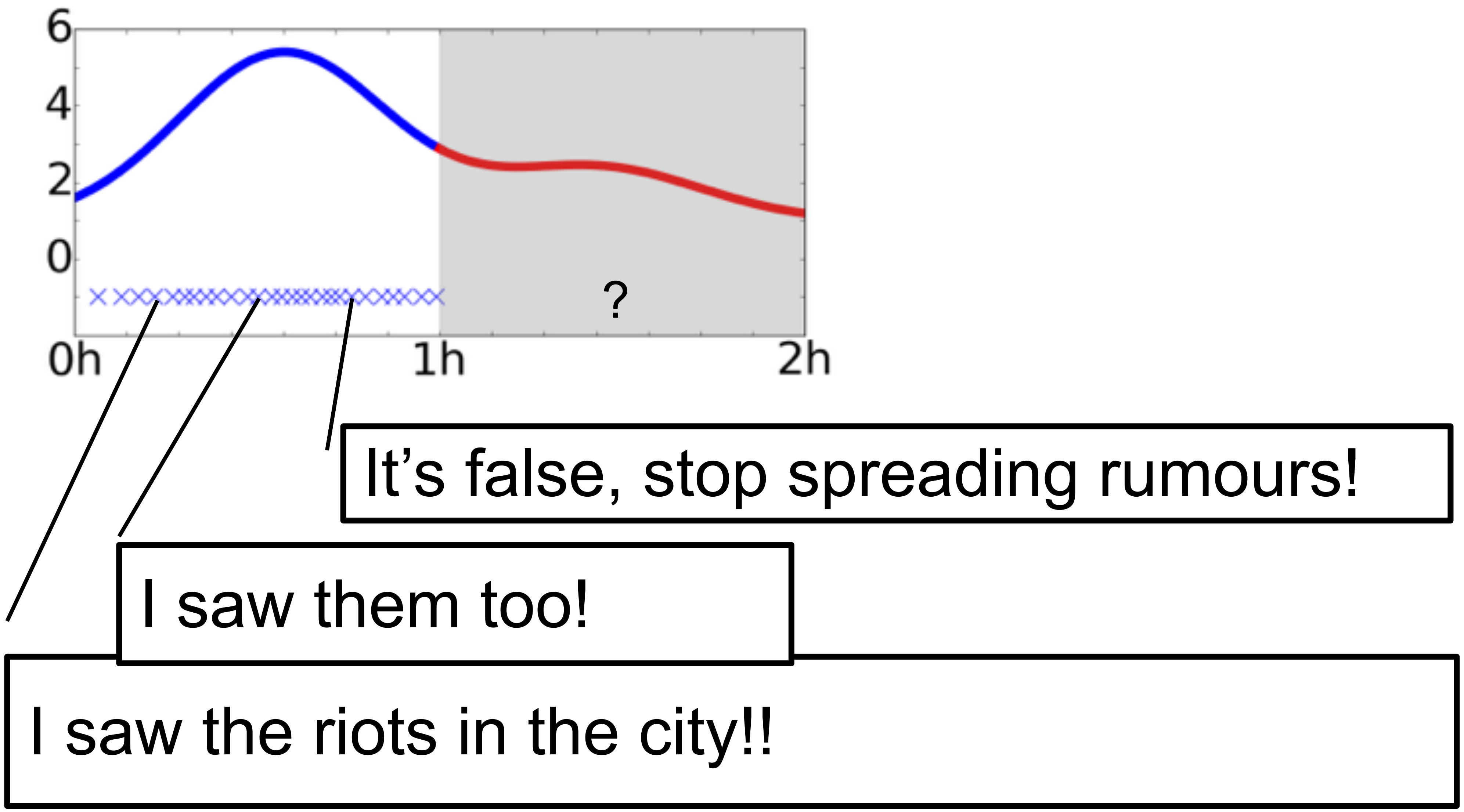
$$
\begin{aligned}
k_{\text{time x corr}}((t,i),(t',i')) = \\
k_{\text{time}}(t,t') \times k_{\text{corr}}(i,i') = \\
k_{\text{time}}(t,t') \times B_{ii'}
\end{aligned}
$$

$B =$



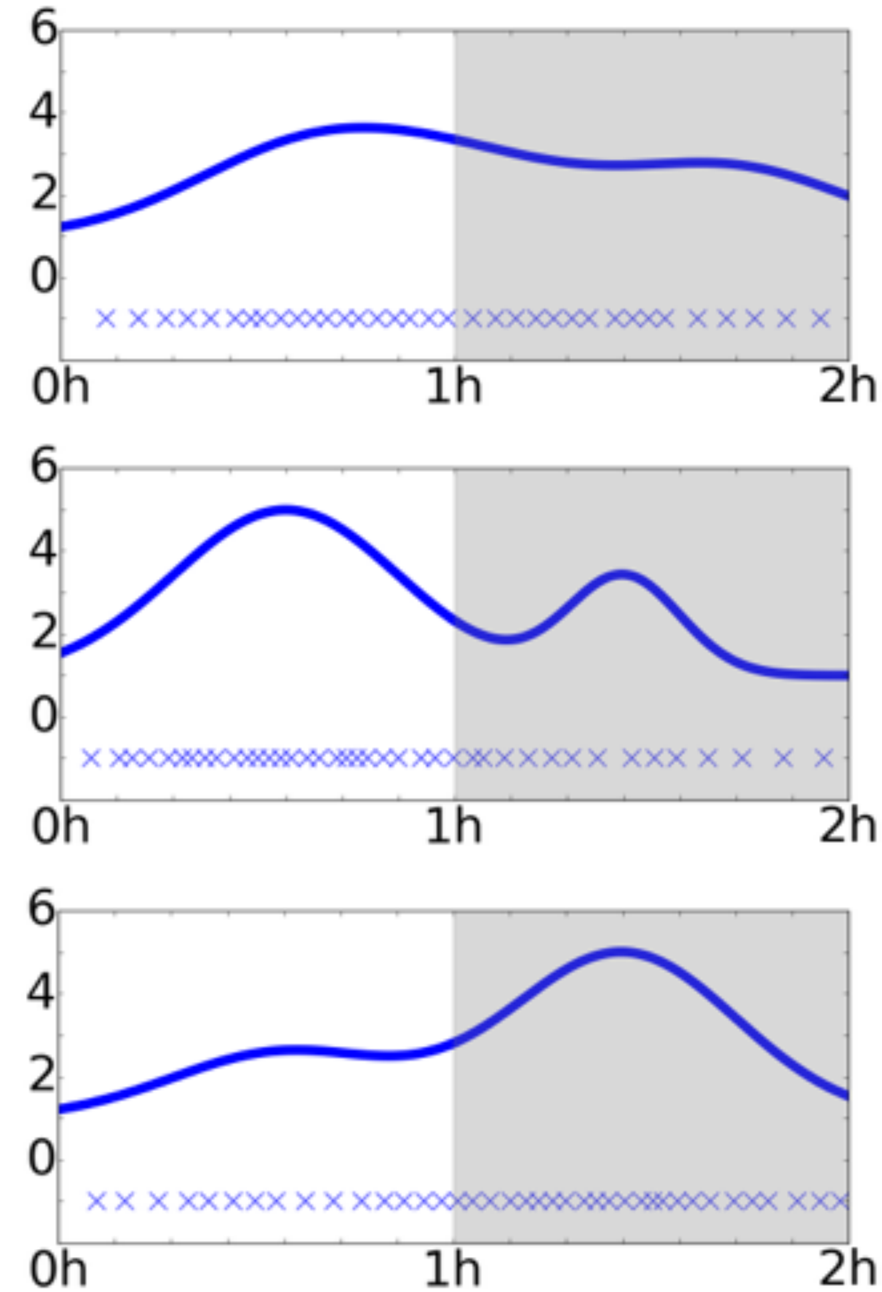
Treat B as hyper parameters and
Learn it by maximising the marginal likelihood

Using reference meme: text



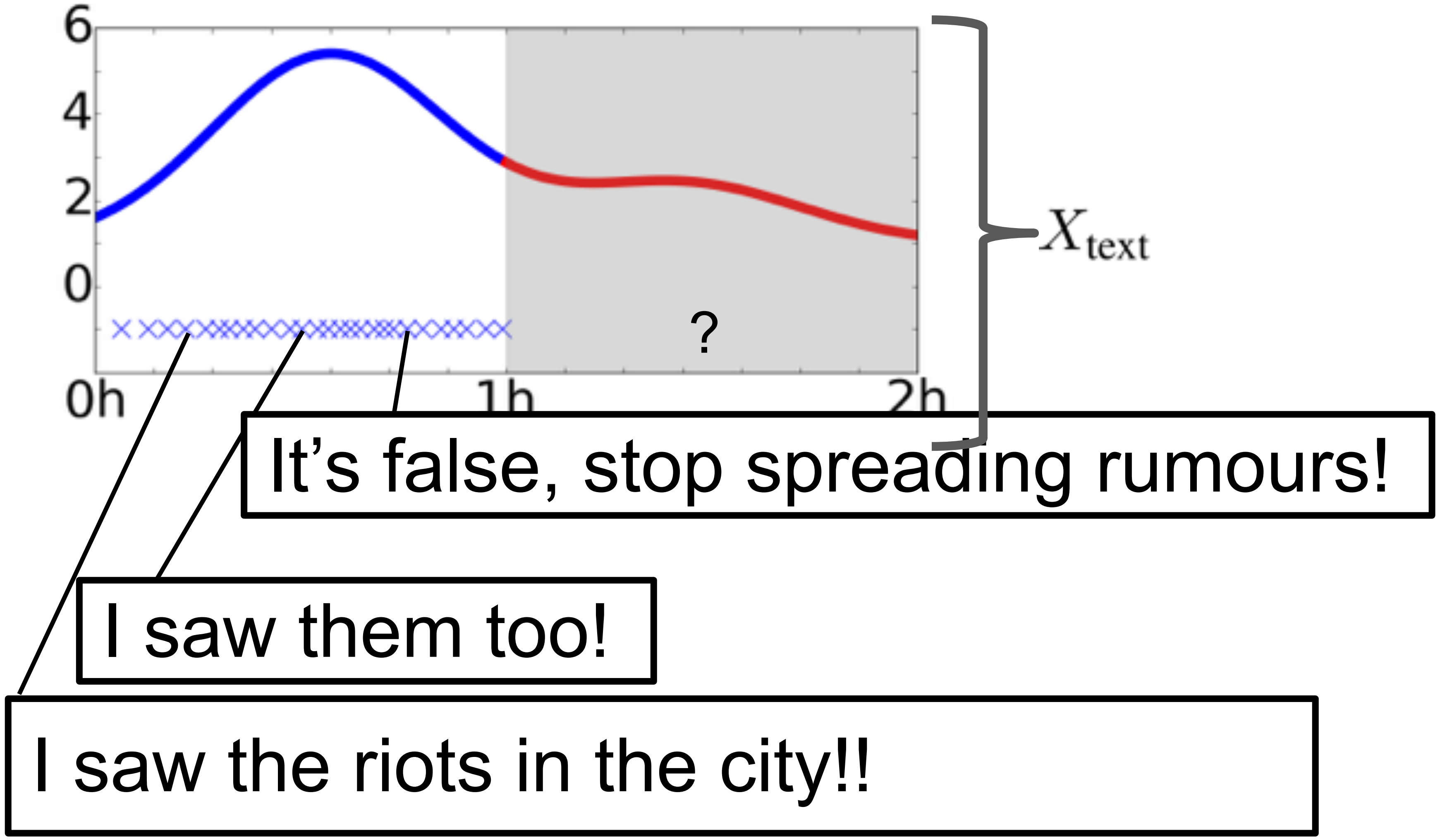
Using reference meme: text

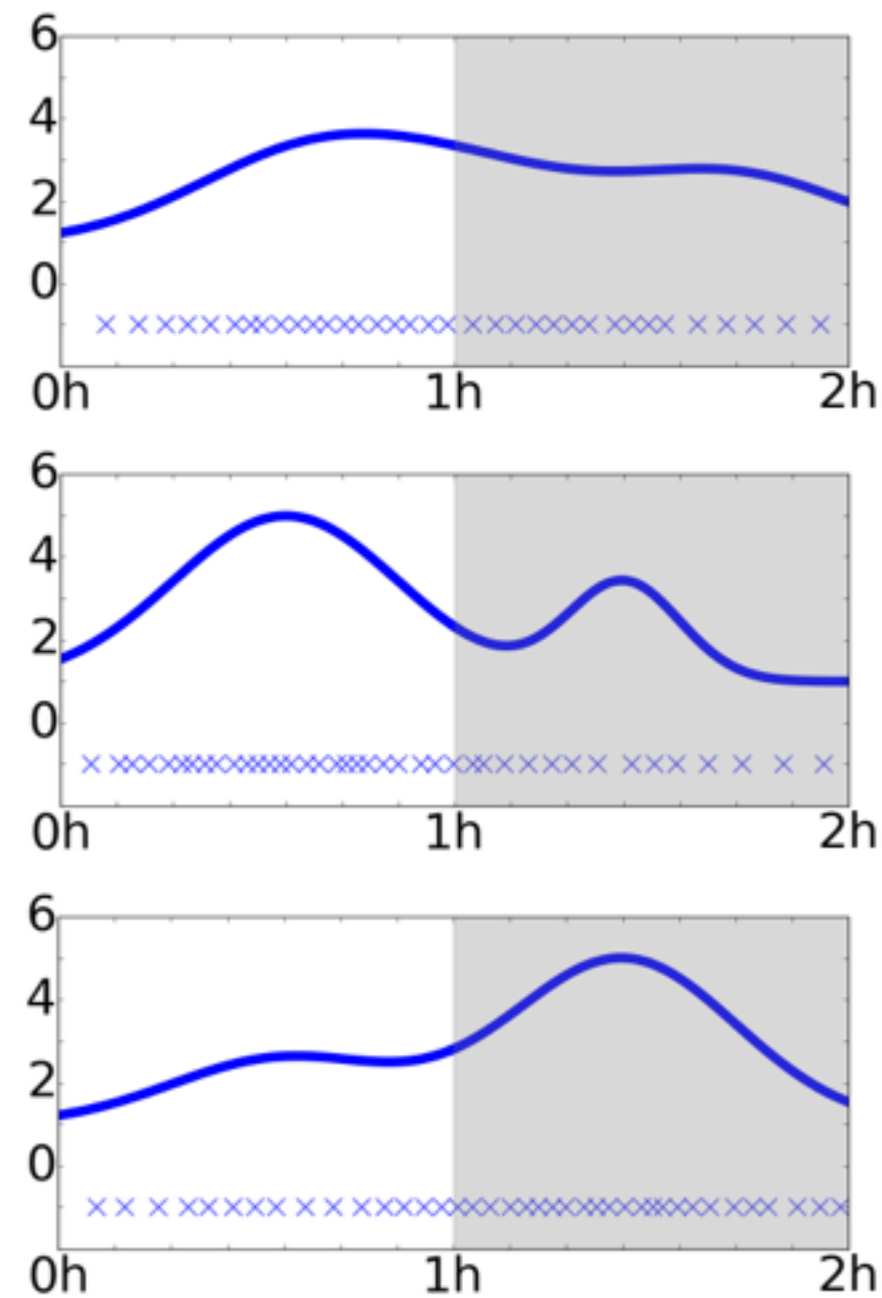
Memes with similar textual content exhibit similar temporal behaviour



Using reference meme: text

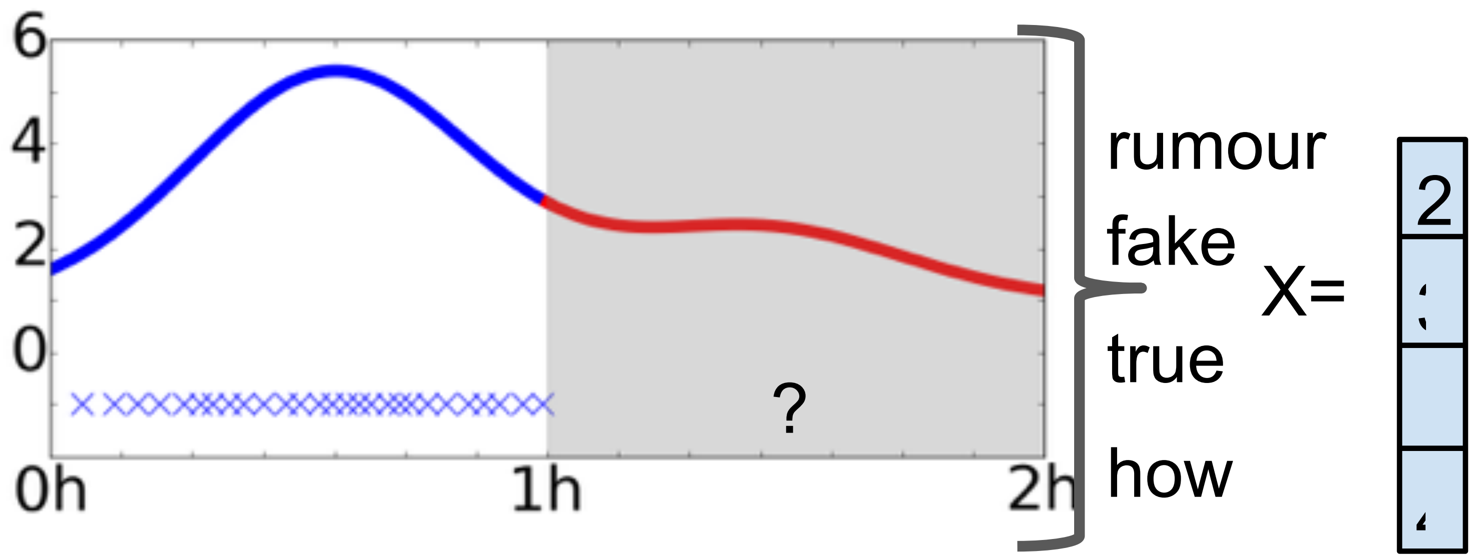
$$k_{\text{time x text}}((t, X_{\text{text}}), (t', X'_{\text{text}})) = k_{\text{time}}(t, t') \times k_{\text{text}}(X_{\text{text}}, X'_{\text{text}})$$

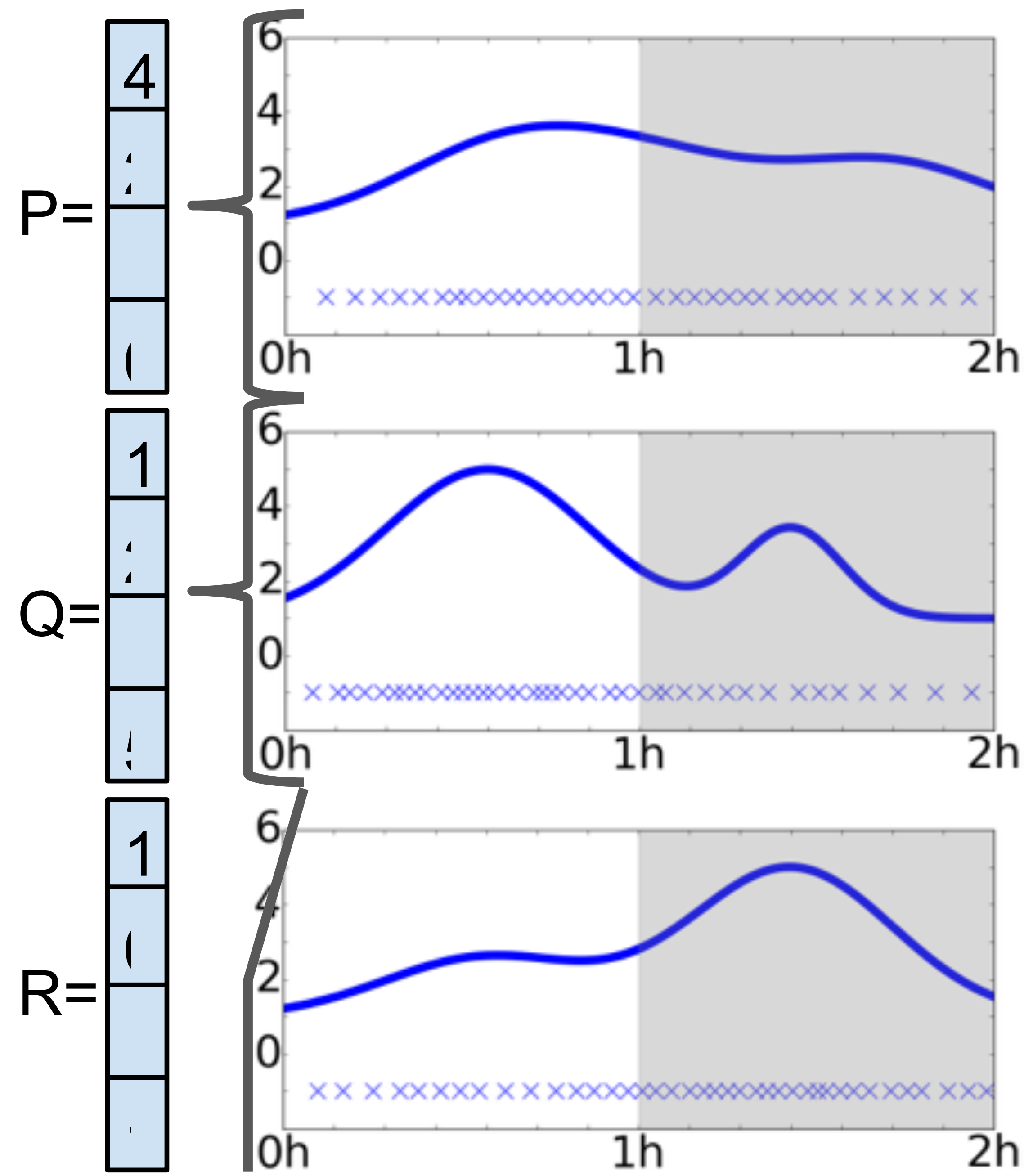




Using reference rumours: text

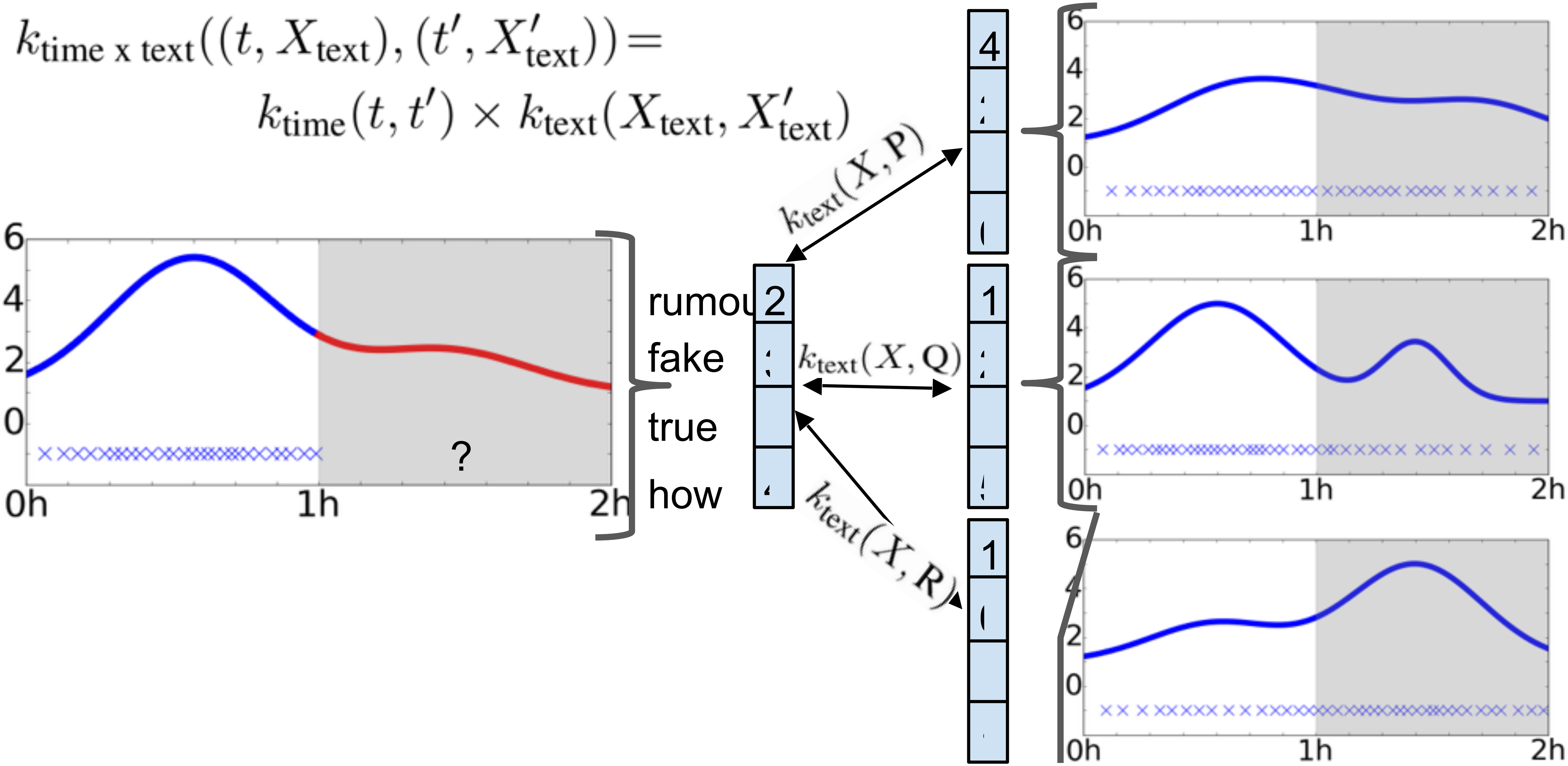
$$k_{\text{time x text}}((t, X_{\text{text}}), (t', X'_{\text{text}})) = k_{\text{time}}(t, t') \times k_{\text{text}}(X_{\text{text}}, X'_{\text{text}})$$





Using reference rumours: text

$$k_{\text{time x text}}((t, X_{\text{text}}), (t', X'_{\text{text}})) = k_{\text{time}}(t, t') \times k_{\text{text}}(X_{\text{text}}, X'_{\text{text}})$$



Using reference rumours: text

$$k_{\text{time x text}}((t, X_{\text{text}}), (t', X'_{\text{text}})) = k_{\text{time}}(t, t') \times k_{\text{text}}(X_{\text{text}}, X'_{\text{text}})$$

$$k_{\text{text}}(X_{\text{text}}, X'_{\text{text}}) = b + c \frac{X_{\text{text}}^T X'_{\text{text}}}{\|X_{\text{text}}\| \|X'_{\text{text}}\|}$$

- Text representation:
  - Brown clusters learnt on a large scale Twitter corpus

Using reference rumours: summary

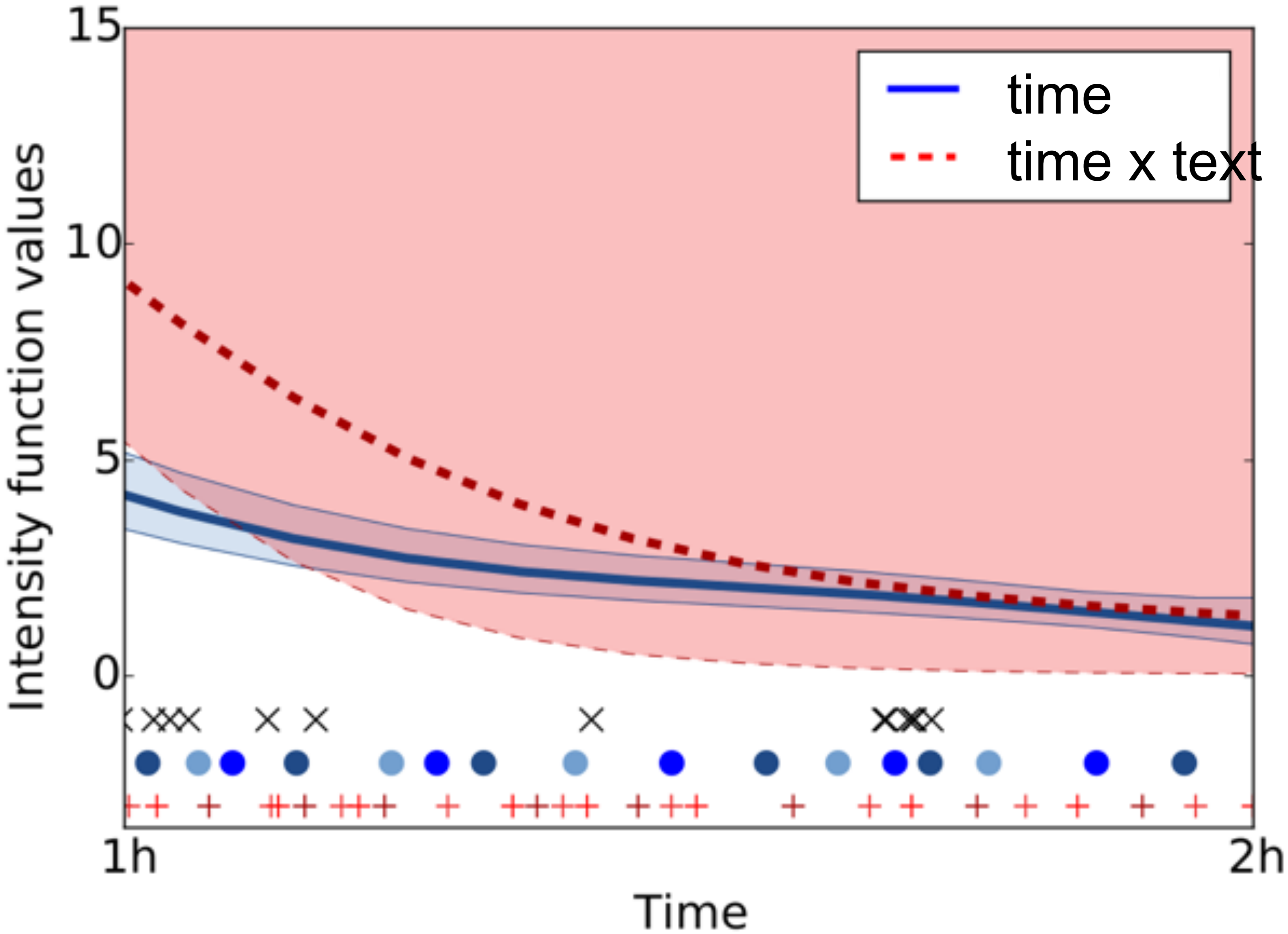
Instead of having only time as input, also use a rumour representation.

$$k_{\text{time x rumours}}((t, X), (t', X')) = k_{\text{time}}(t, t') \times k_{\text{rumours}}(X, X')$$

$$k_{\text{time x corr}}((t, i), (t', i')) = k_{\text{time}}(t, t') \times k_{\text{corr}}(i, i')$$

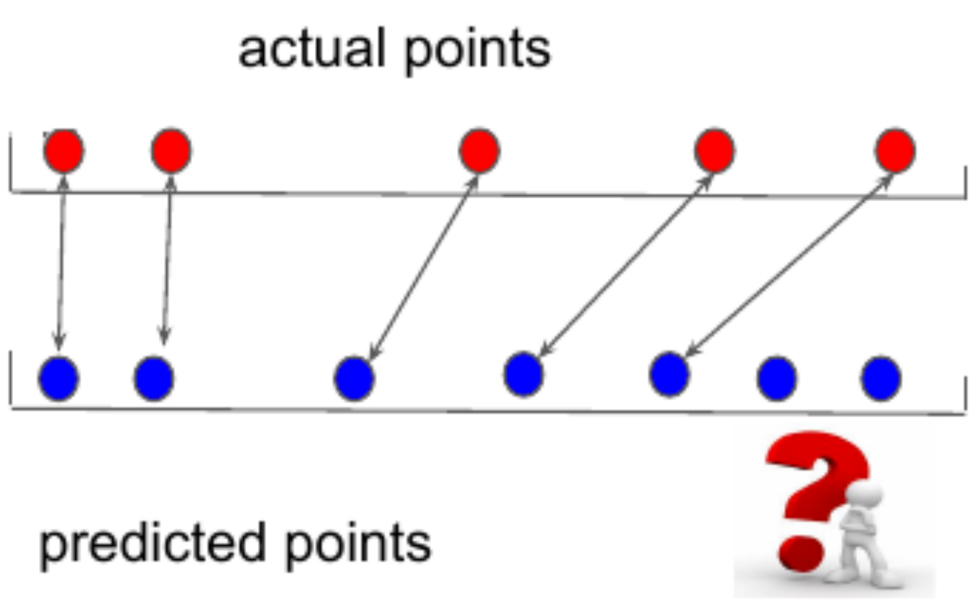
$$k_{\text{time x text}}((t, X_{\text{text}}), (t', X'_{\text{text}})) = k_{\text{time}}(t, t') \times k_{\text{text}}(X_{\text{text}}, X'_{\text{text}})$$

Predicting tweet arrival times

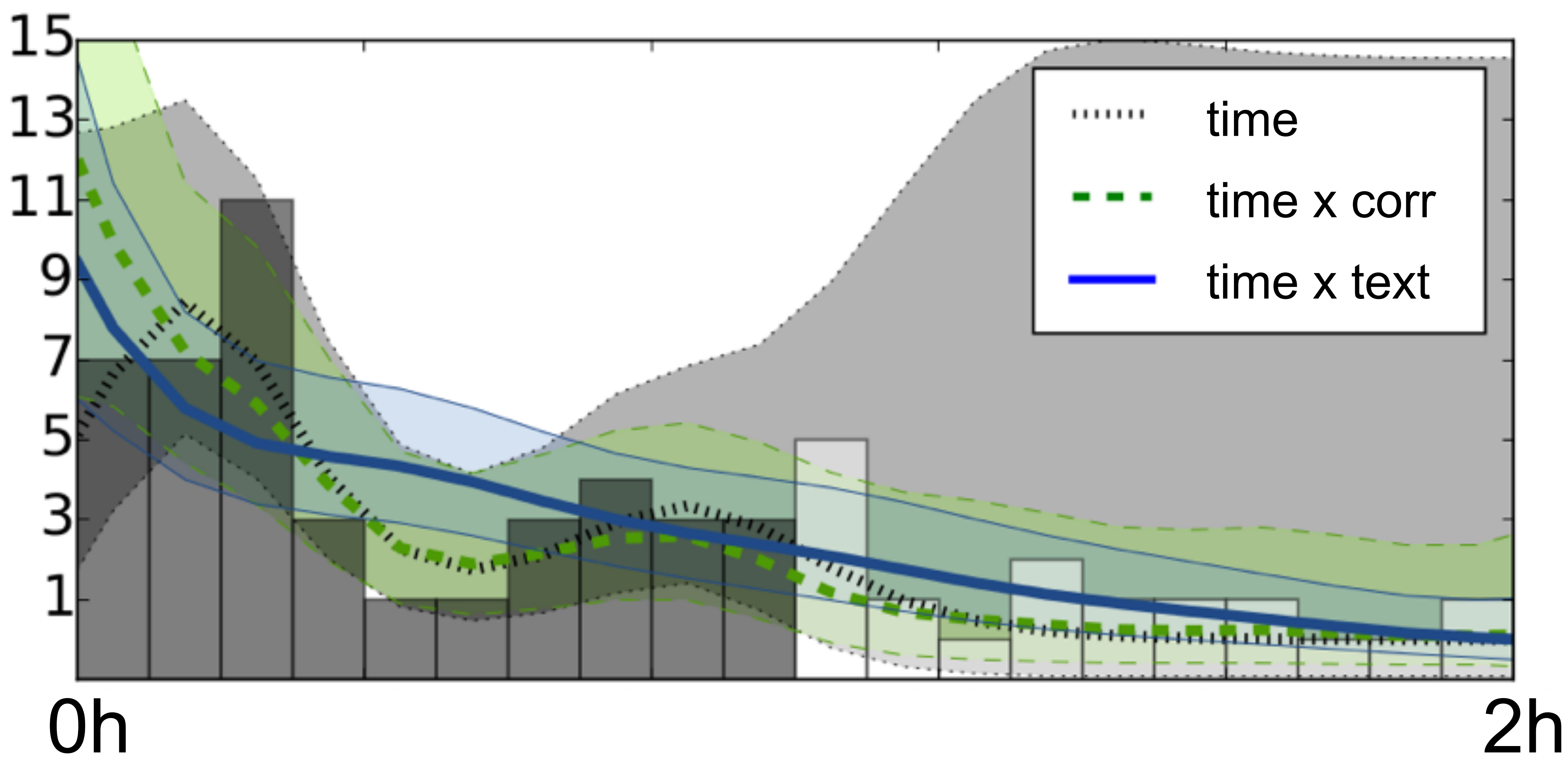


Results

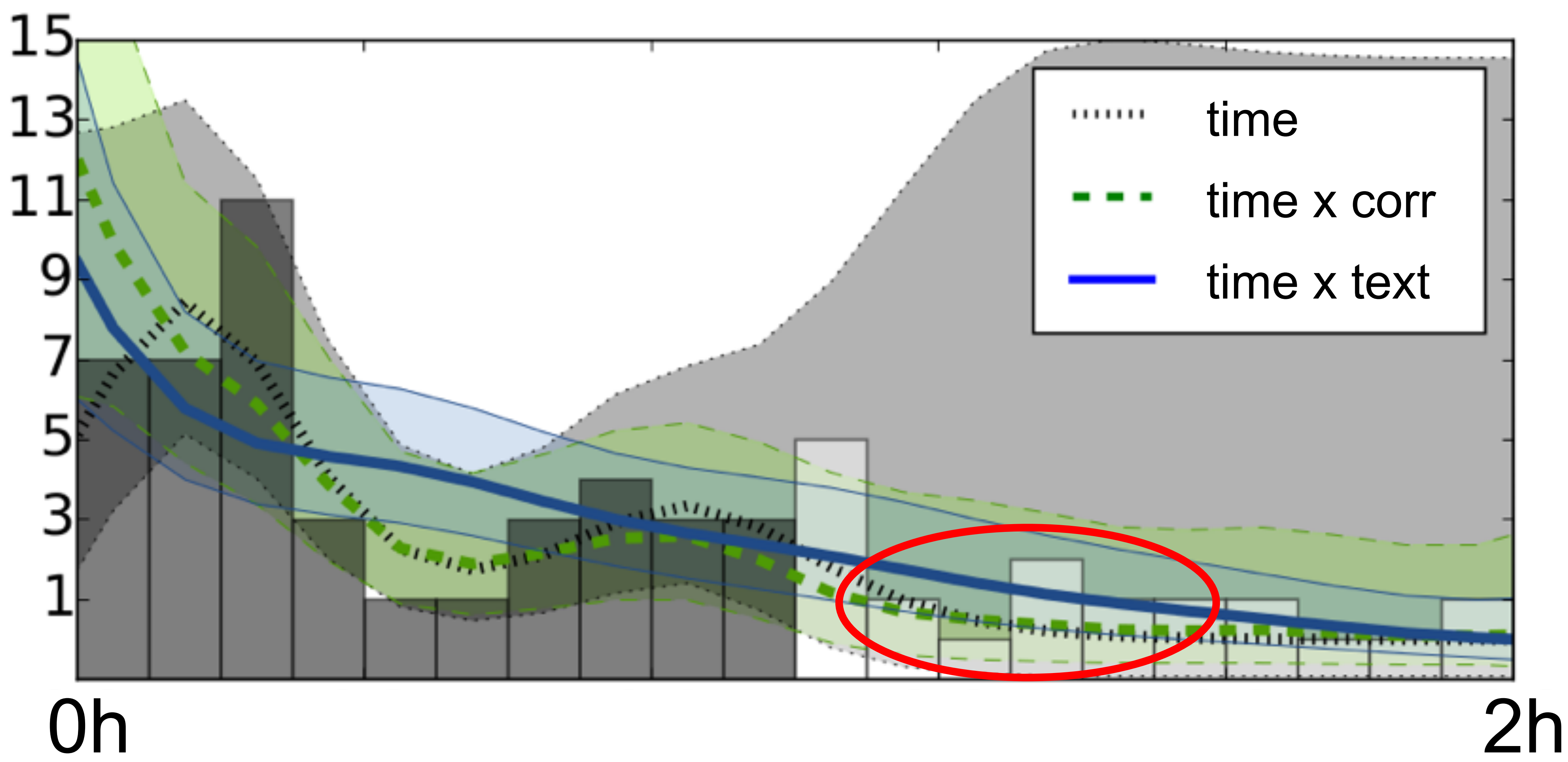
method	ARMSE	PRMSE
GPLIN	20.60±22.01⋆	1279.78±903.90⋆
HPP	21.85±22.82⋆	431.4±96.5⋆
HP	15.94±18.20	363.70±59.01⋆
LGCP	13.31±14.28	261.26±92.97⋆
LGCP Pooled	19.18±20.36⋆	183.25±102.20⋆
LGCPTXT	15.52±18.79	154.05±115.70



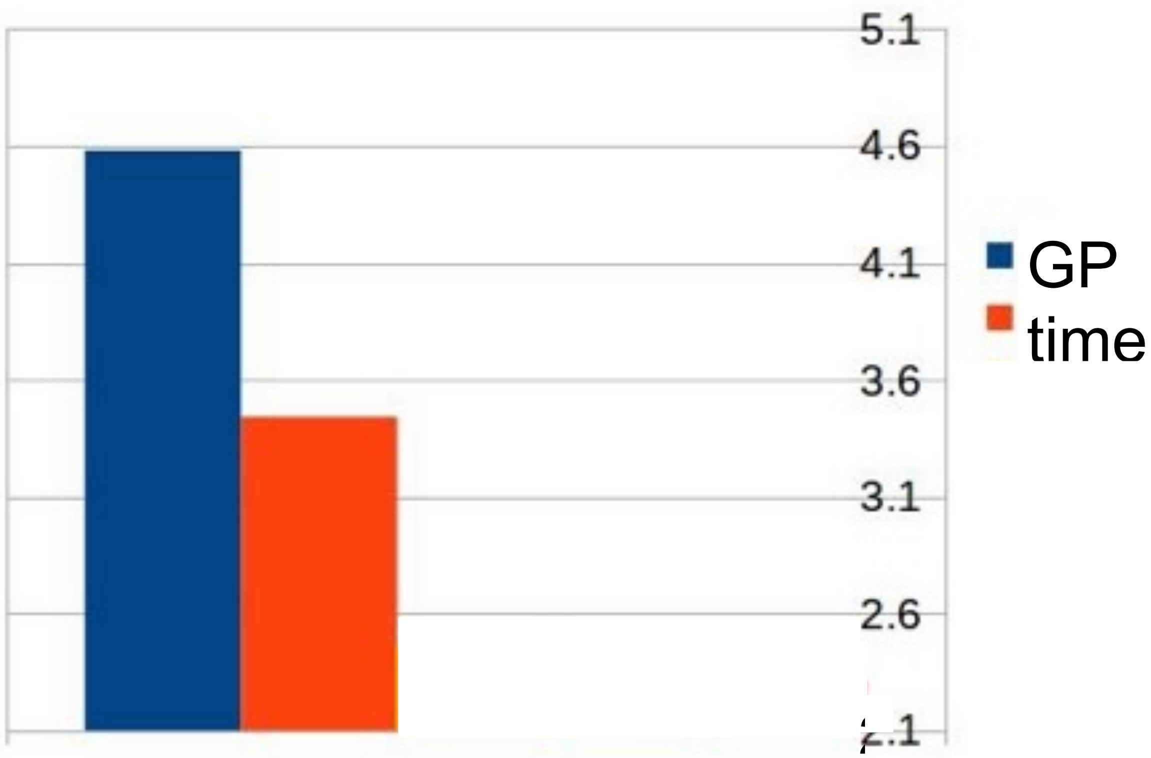
Results



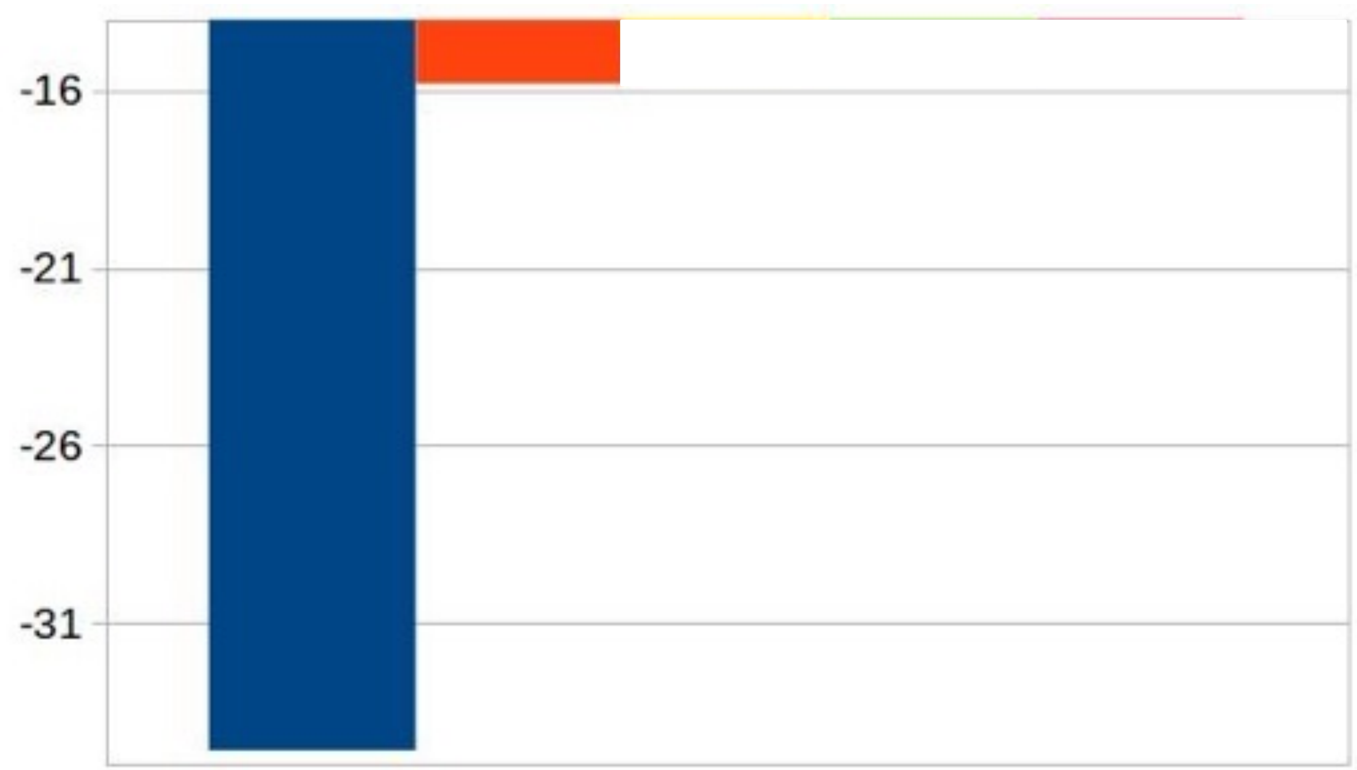
Results



Results

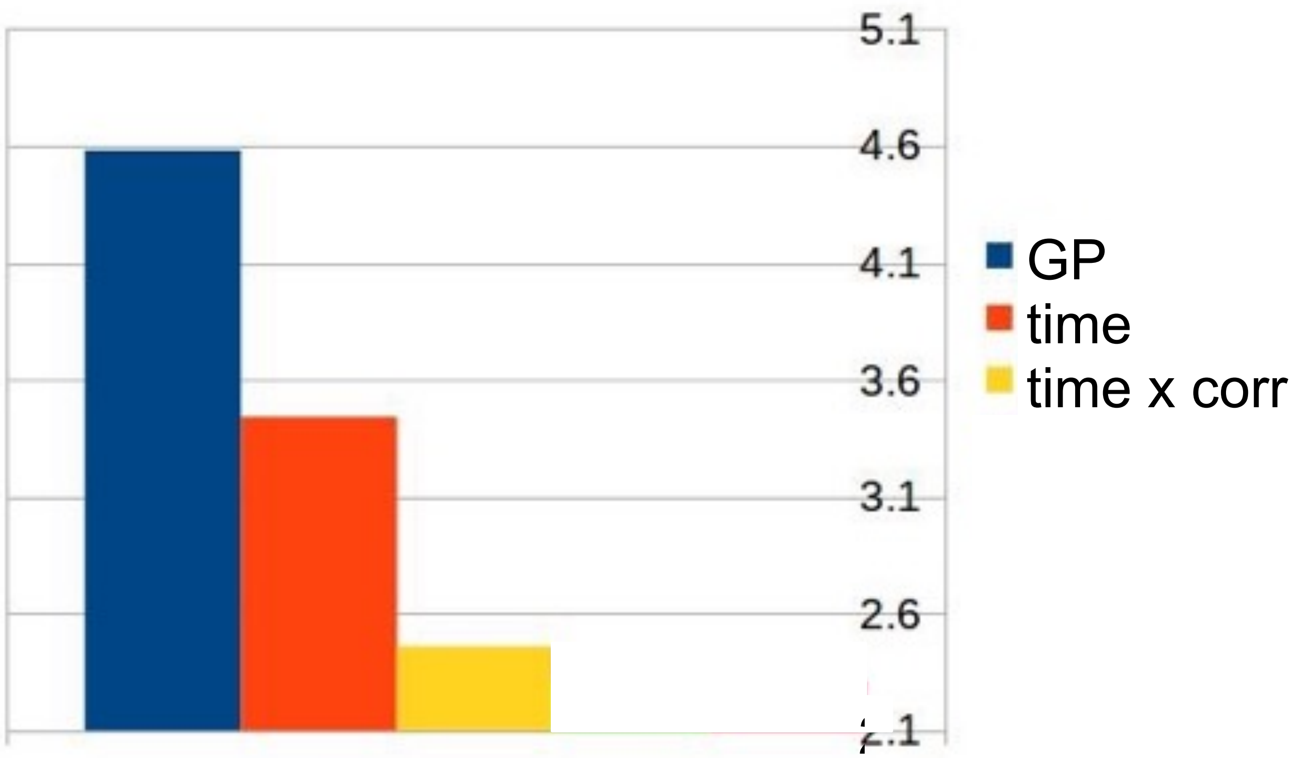


MSE

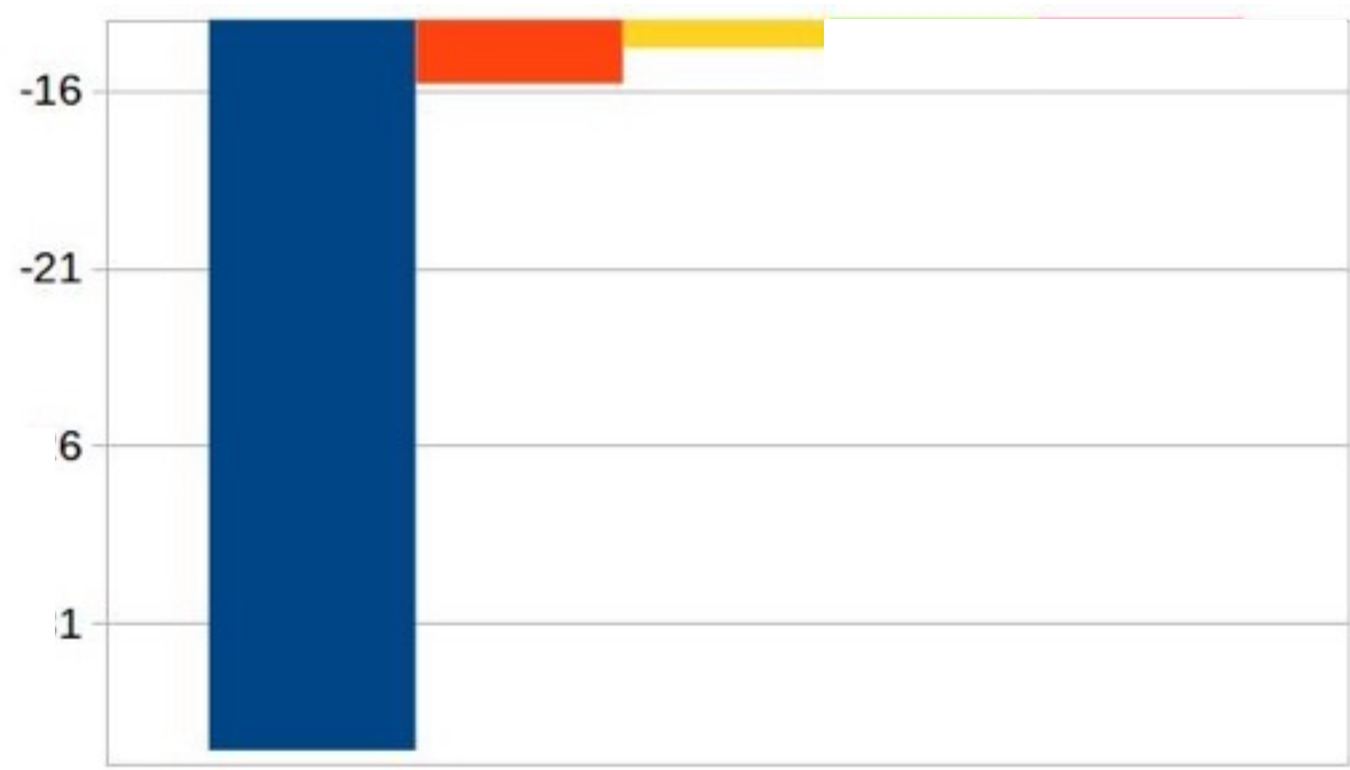


Log
Likelihood

Results

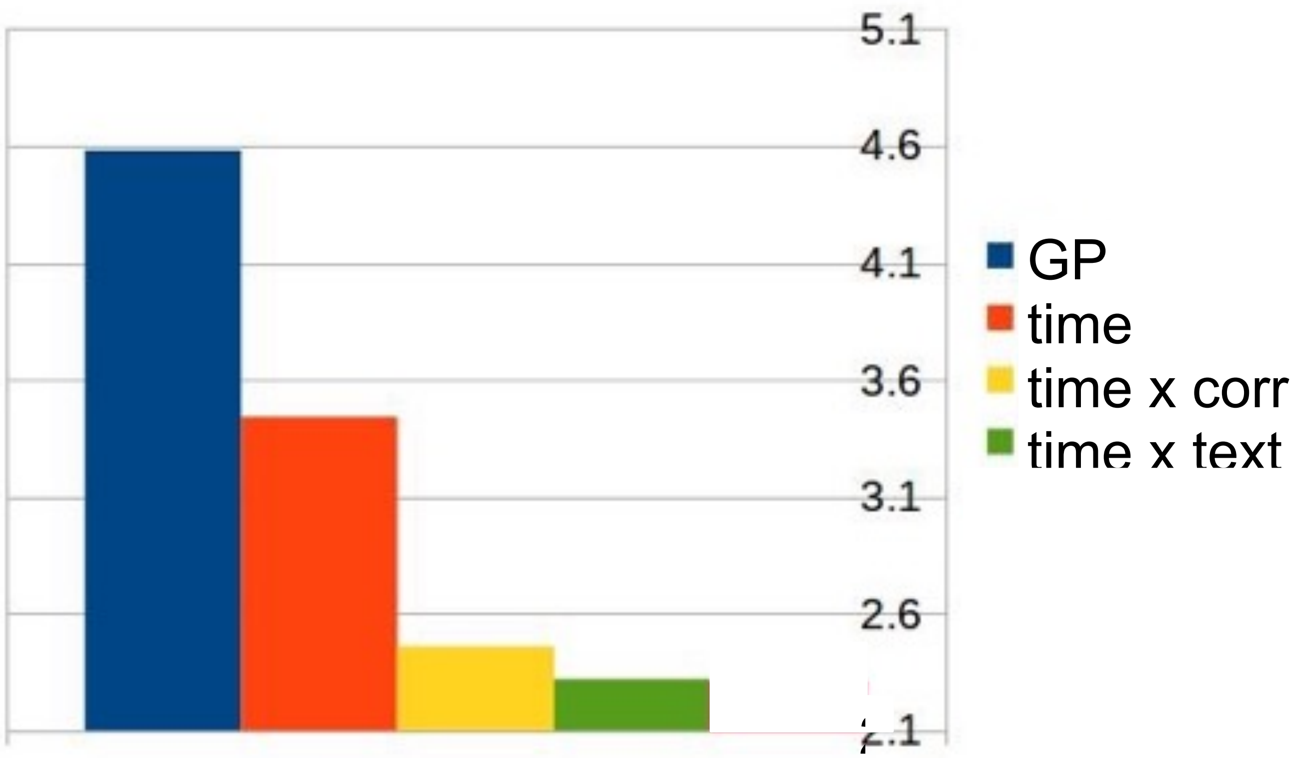


MSE

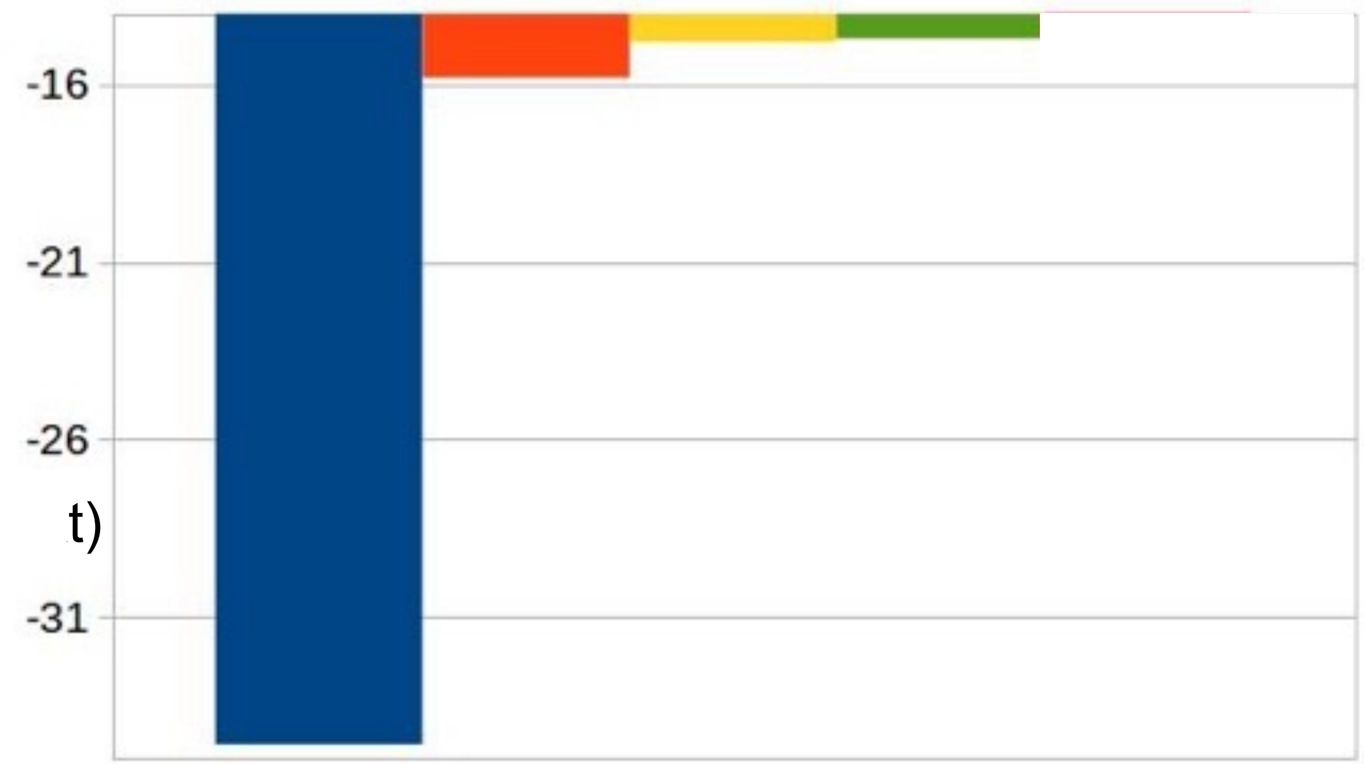


Log
Likelihood

Results

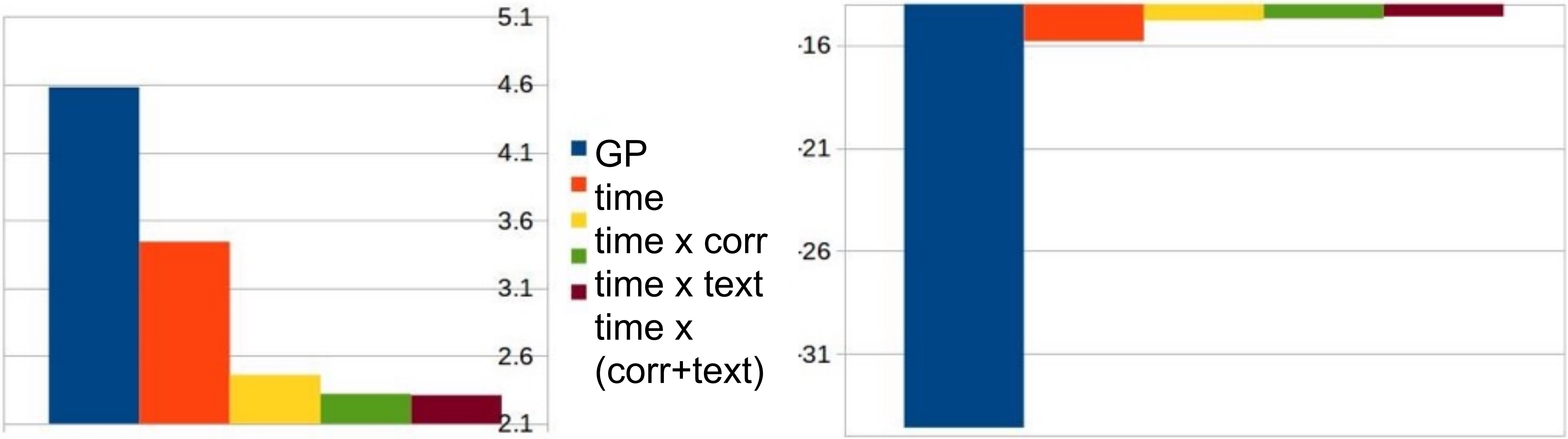


MSE



Log
Likelihood

Results



MSE

Log
Likelihood

References

- Rasmussen and Williams, Gaussian Processes for Machine Learning, 2006

- M. Lukasik, P.K. Srijith, T. Cohn and K. Bontcheva. Modeling Tweet Arrival Times using Log-Gaussian Cox Processes. EMNLP, 2015.

- E. V. Bonilla, K. M. A. Chai, C. K. I. Williams, "Multi-task Gaussian process prediction", NIPS, 2008.

- Point process modelling of rumour dynamics in social media
- Michal Lukasik, Trevor Cohn and Kalina Bontcheva., ACL 2015