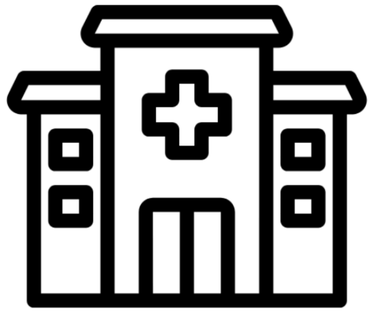
EventFlow: Forecasting Temporal Point Processes with Flow Matching

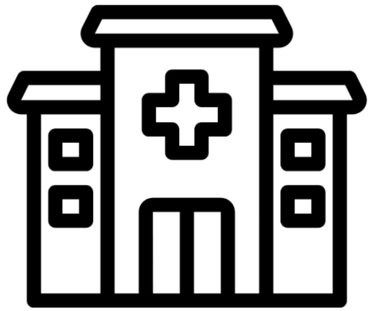
Gavin Kerrigan[13], Kai Nelson[23], Padhraic Smyth[3]

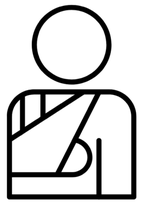
[1]University of Oxford [2]UC Berkeley [3]UC Irvine

Event data occurring at irregular times

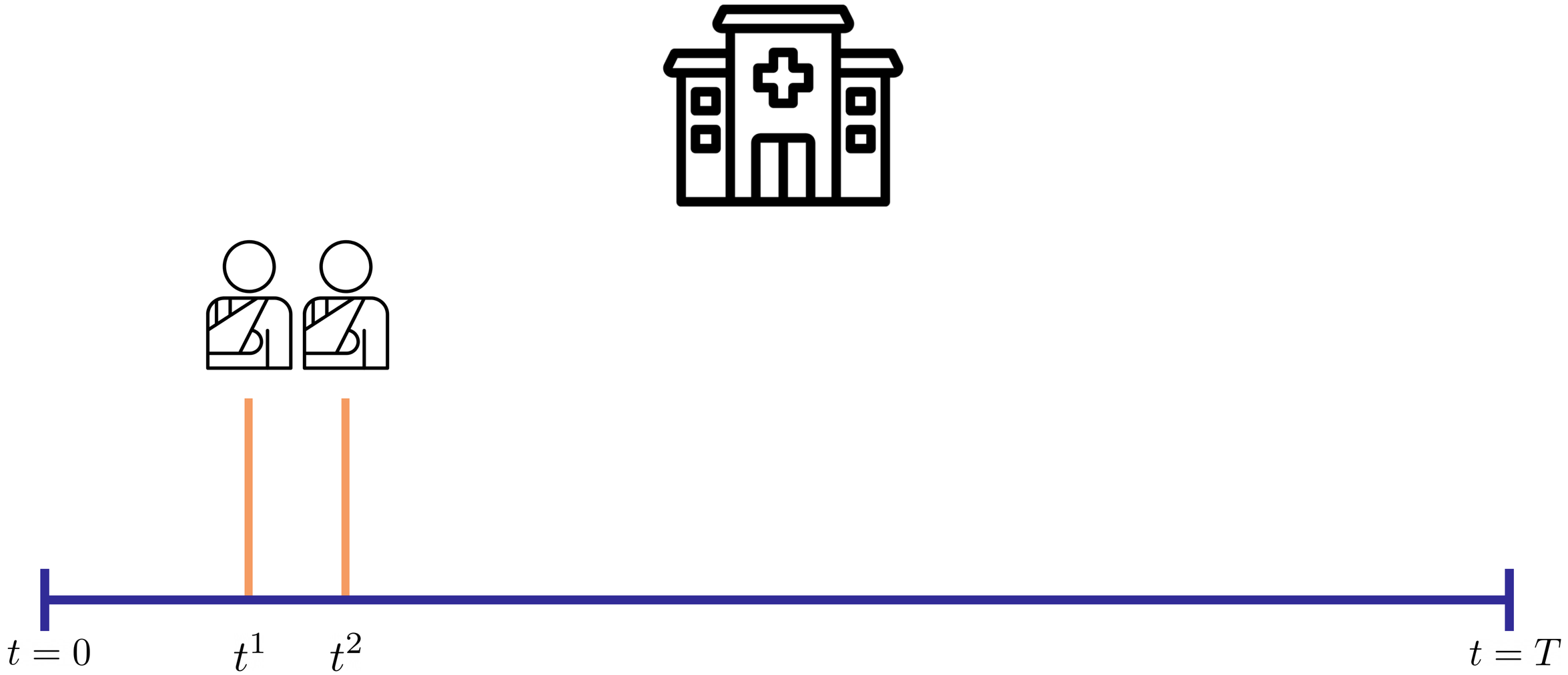


Event data occurring at irregular times

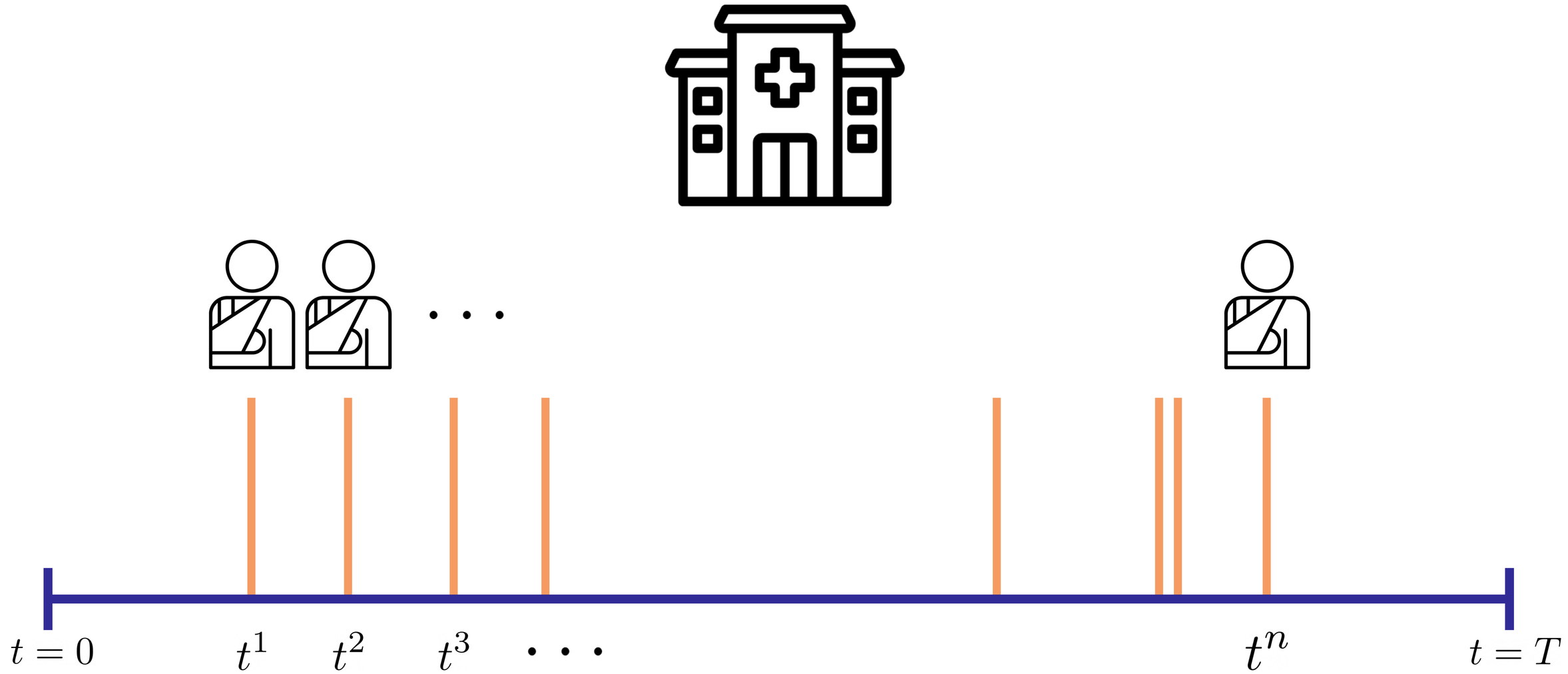




Event data occurring at irregular times

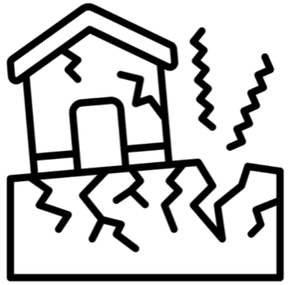


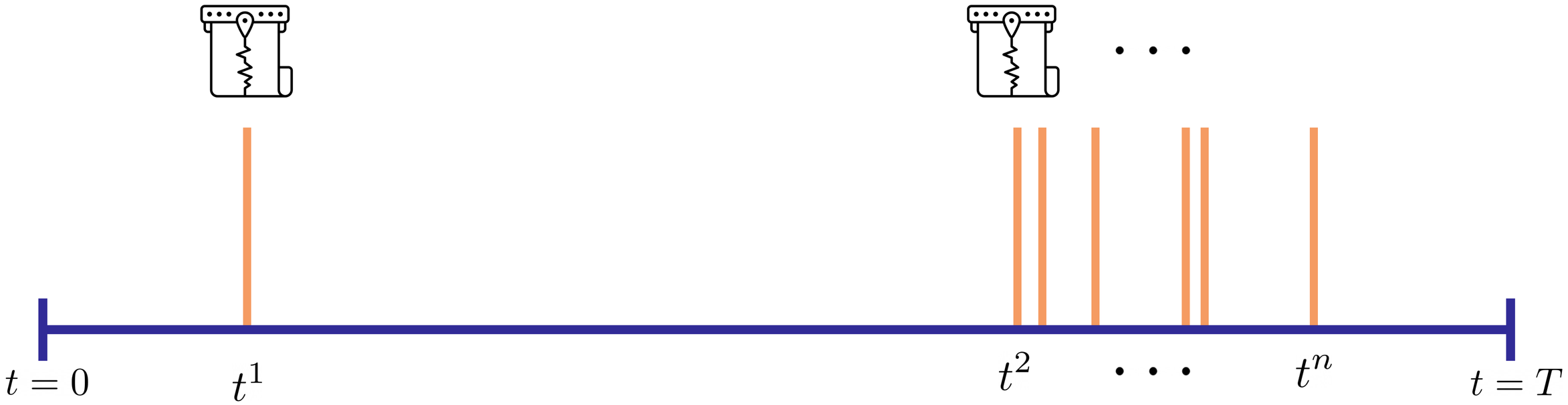
Event data occurring at irregular times



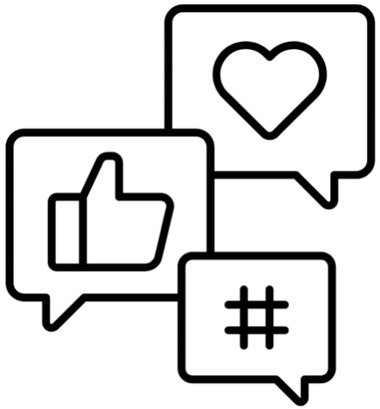
Event data occurring at irregular times

Often involve complex, varied patterns

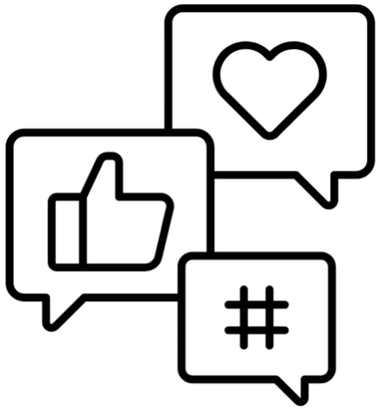




Social Media

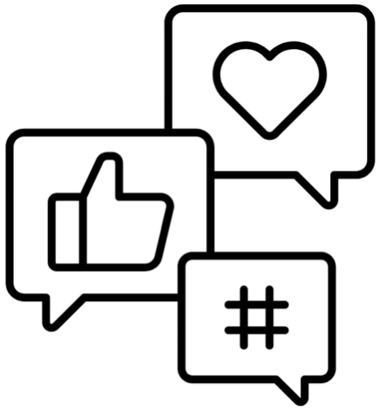


Social Media



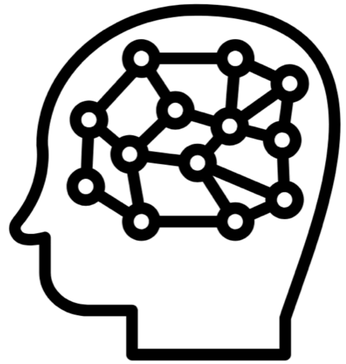
Consumer Behaviour

Social Media



Consumer Behaviour

Brain Activity

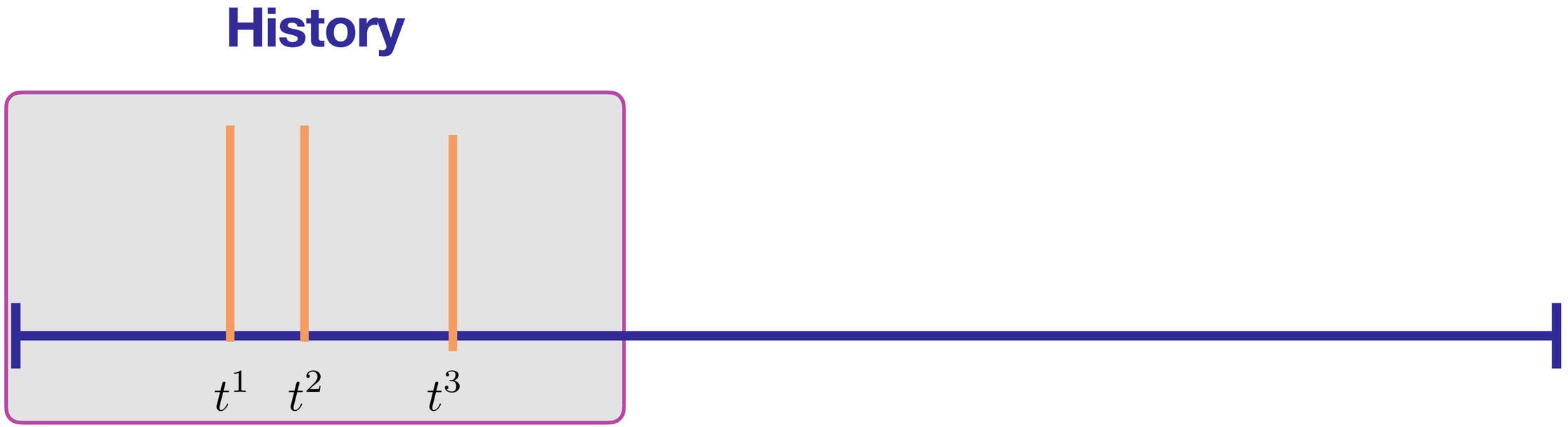


Key Challenge: Forecasting

When will future events occur?

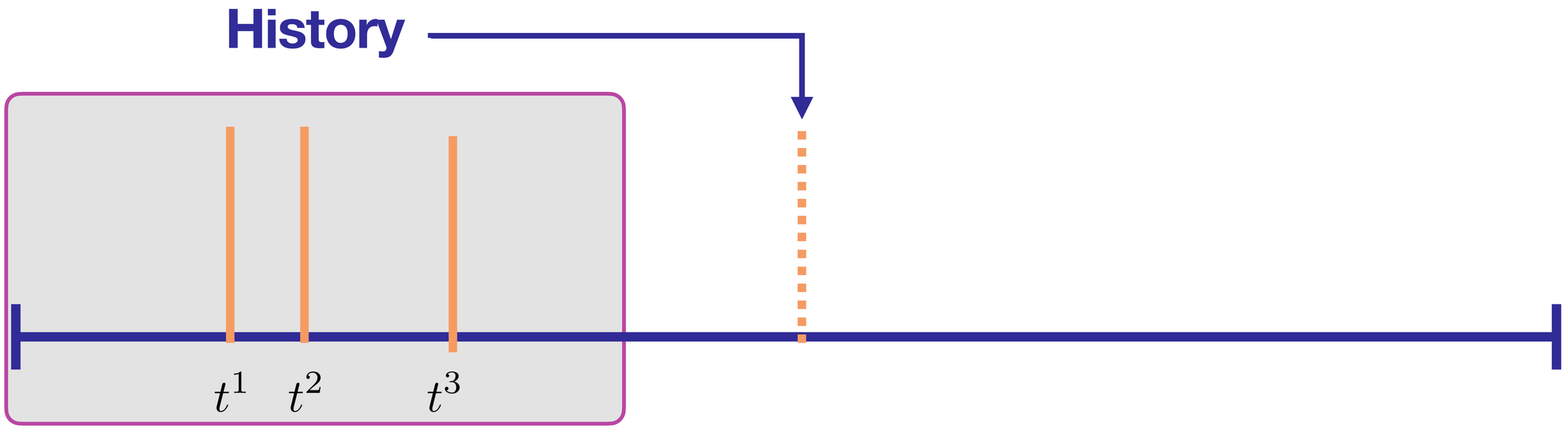
Key Challenge: Forecasting

Standard approaches are largely autoregressive



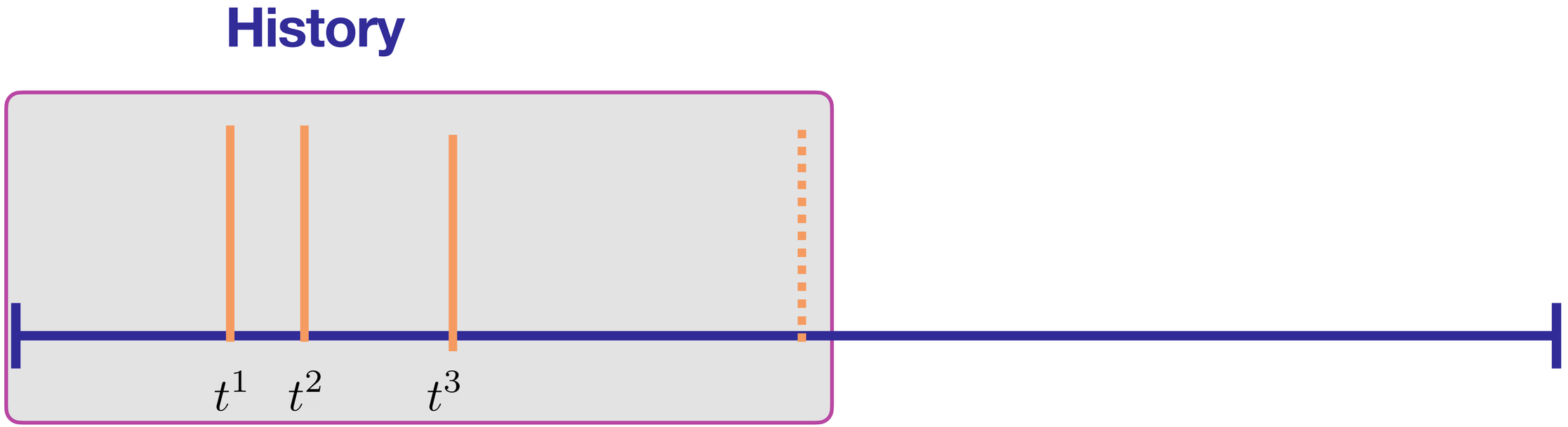
Key Challenge: Forecasting

Standard approaches are largely autoregressive



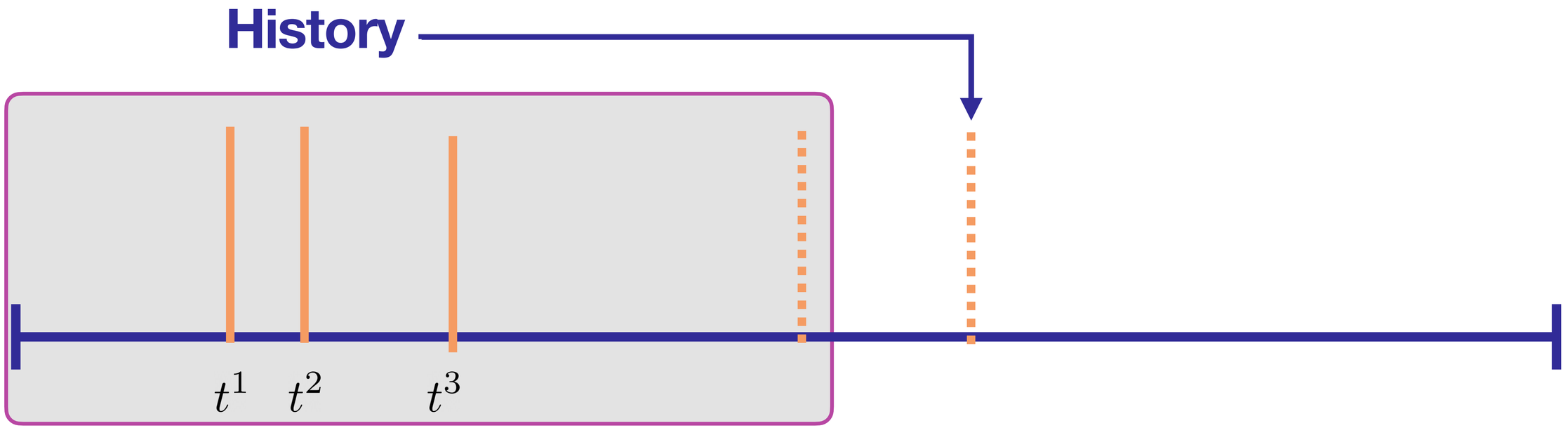
Key Challenge: Forecasting

Standard approaches are largely autoregressive



Key Challenge: Forecasting

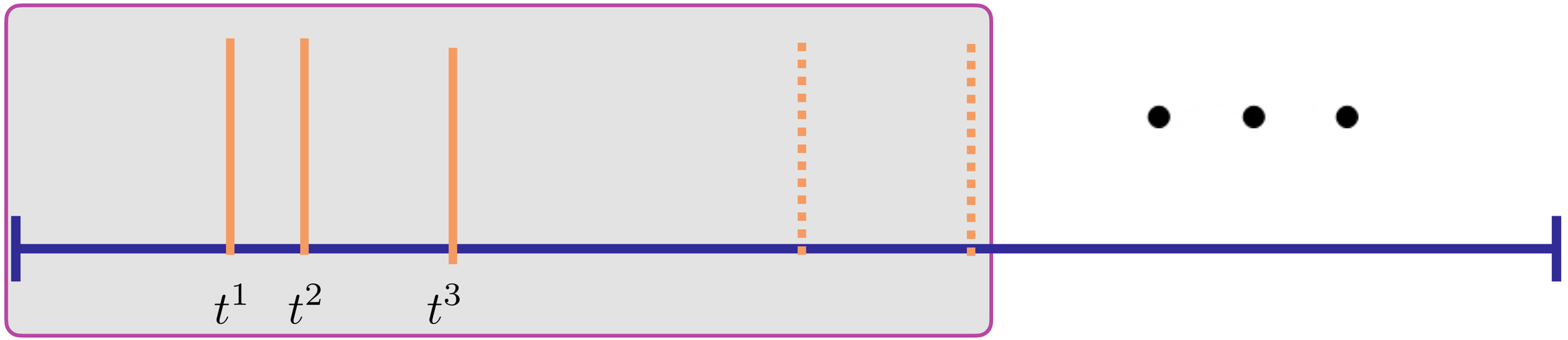
Standard approaches are largely autoregressive



Key Challenge: Forecasting

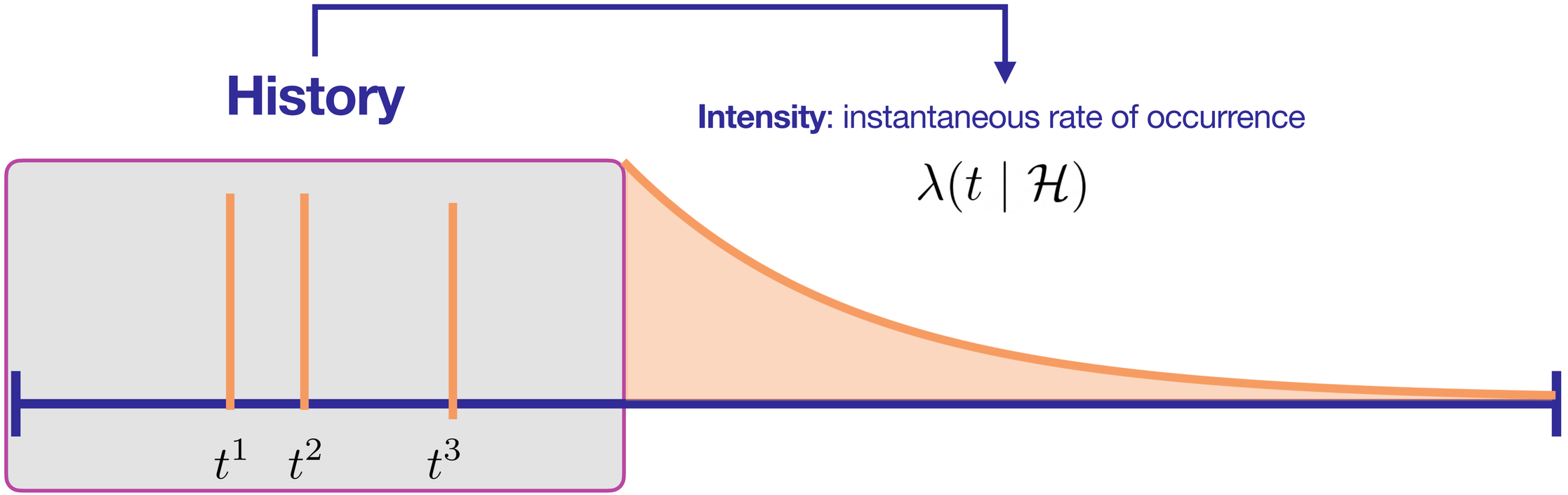
Standard approaches are largely autoregressive

History



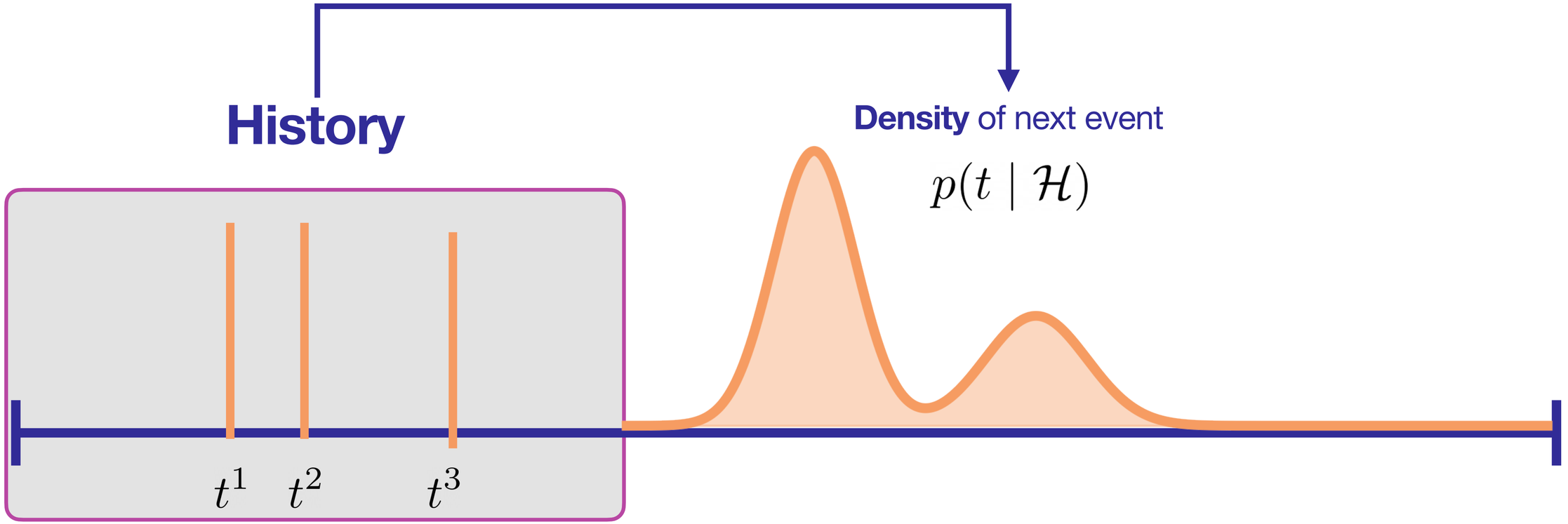
Key Challenge: Forecasting

Standard approaches are largely autoregressive



Key Challenge: Forecasting

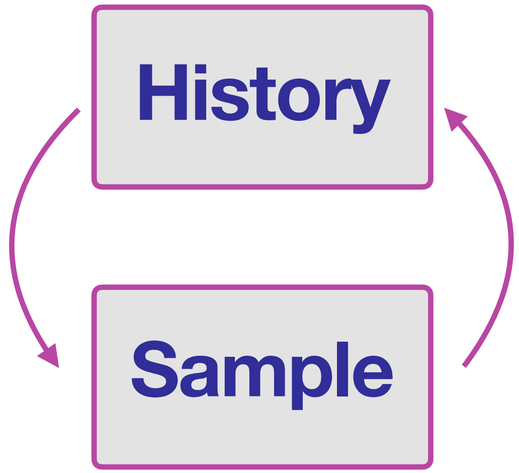
Standard approaches are largely autoregressive



Limitations of Autoregressive Models

Reliance on simple or hand-crafted forms for the intensity/density

- Requirement: able to evaluate likelihood to train
- More expressive choices quickly become expensive



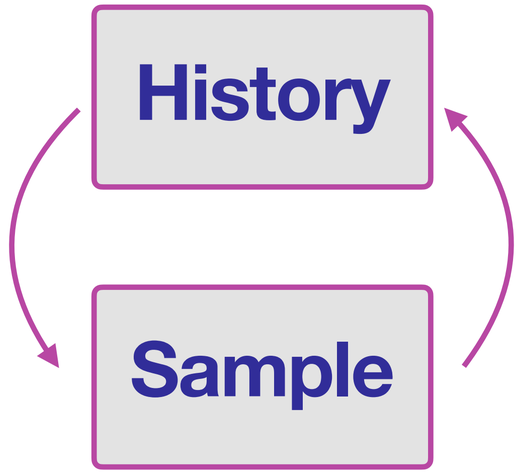
Limitations of Autoregressive Models

Reliance on simple or hand-crafted forms for the intensity/density

- Requirement: able to evaluate likelihood to train
- More expressive choices quickly become expensive

Forecast and observed events treated equally

- Errors can cascade and accumulate
- "Myopic" — model cannot correct early errors at later steps



Limitations of Autoregressive Models

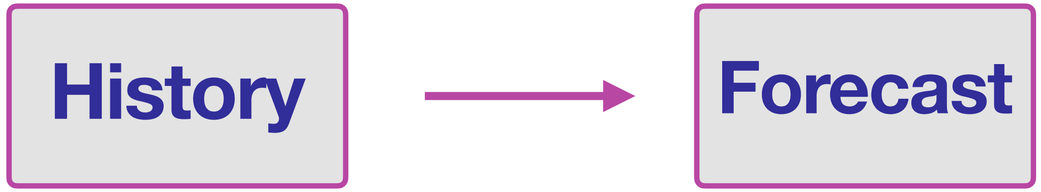
Reliance on simple or hand-crafted forms for the intensity/density

- Requirement: able to evaluate likelihood to train
- More expressive choices quickly become expensive

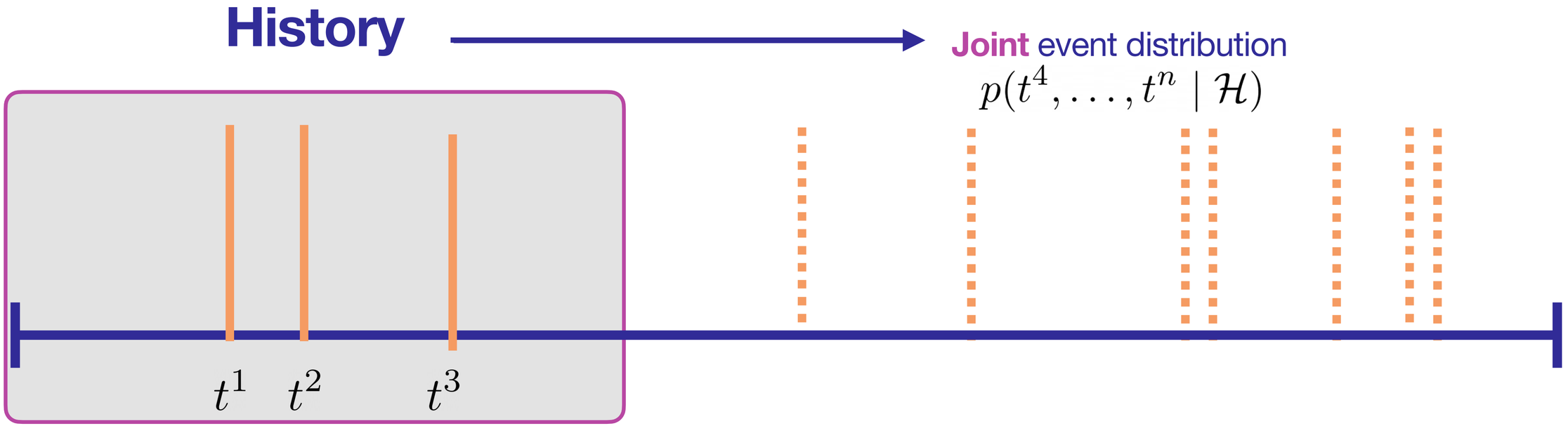
Forecast and observed events treated equally

- Errors can cascade and accumulate
- "Myopic" — model cannot correct early errors at later steps

EventFlow: directly predict the full forecast



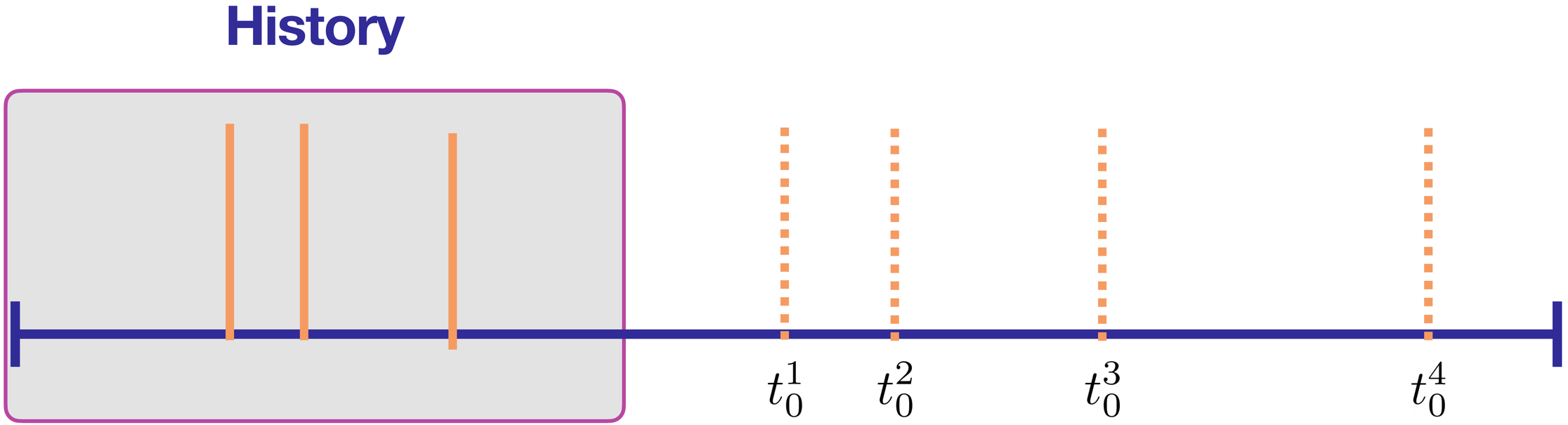
EventFlow: Forecasting Event Sequences



EventFlow: Forecasting Event Sequences

[1] Initialise a forecast randomly

$$t_0^i \sim q(\cdot) \qquad i = 1, 2, \ldots, n$$



EventFlow: Forecasting Event Sequences

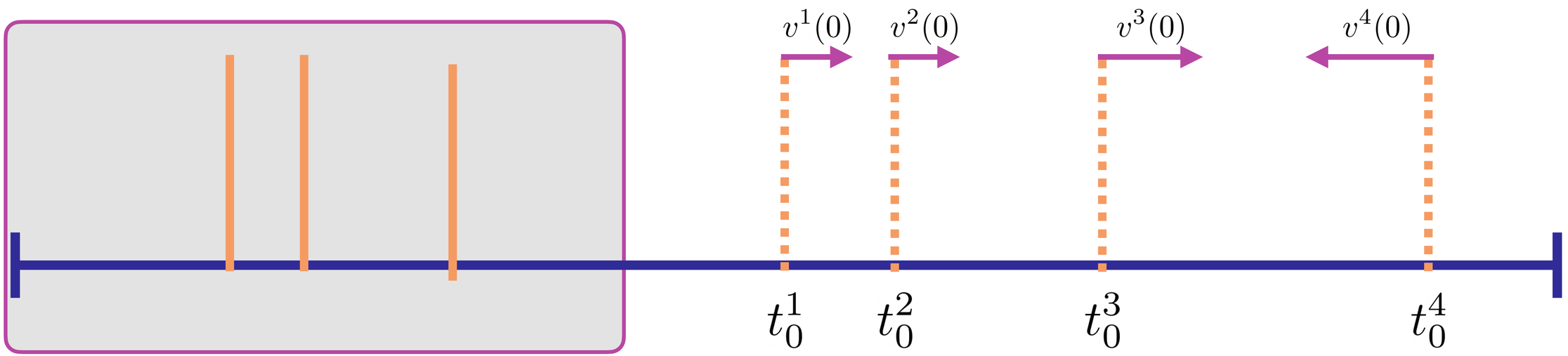
[1] Initialise a forecast randomly

$$t_0^i \sim q(\cdot) \qquad i = 1, 2, \ldots, n$$

[2] Give each event a velocity

$$v^i(s, \mathbf{t}_s) \qquad s \in [0, 1]$$

- Depends on history and current forecast state



EventFlow: Forecasting Event Sequences

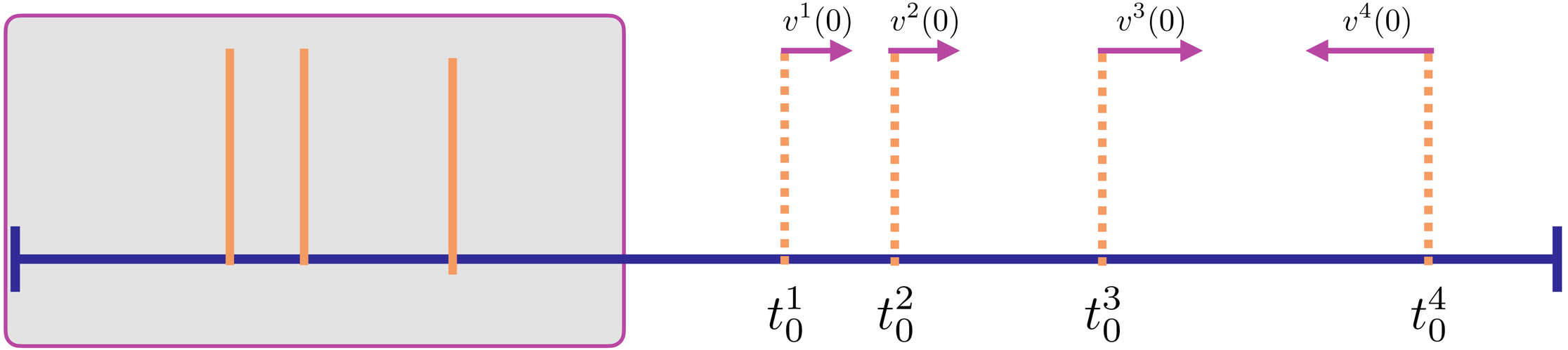
[1] Initialise a forecast randomly $t_0^i \sim q(\cdot) \qquad i = 1, 2, \ldots, n$

[2] Give each event a velocity $v^i(s, \mathbf{t}_s) \qquad s \in [0, 1]$

- Depends on history and current forecast state

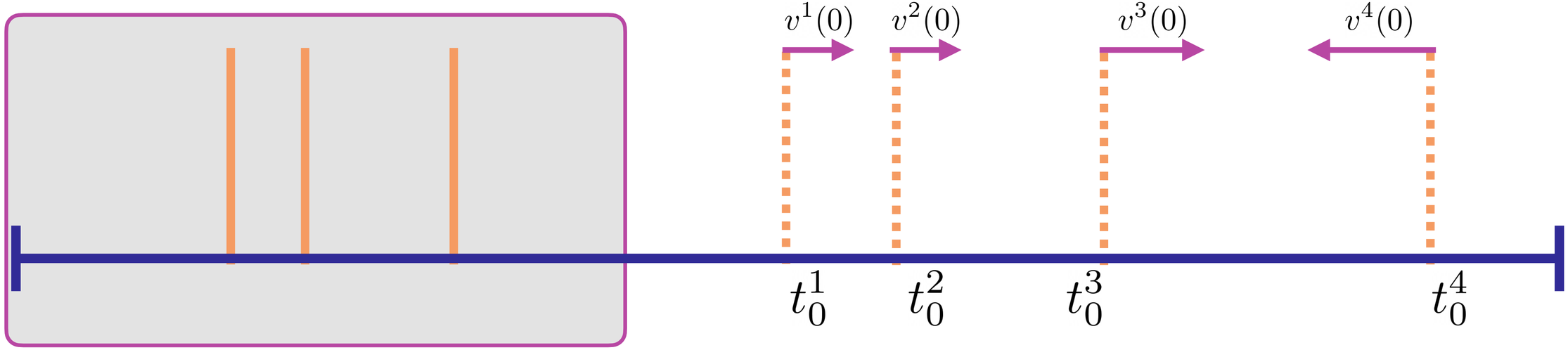
[3] Flow each event according to its velocity $dt_s^i = v^i(s, \mathbf{t}_s)ds$

History

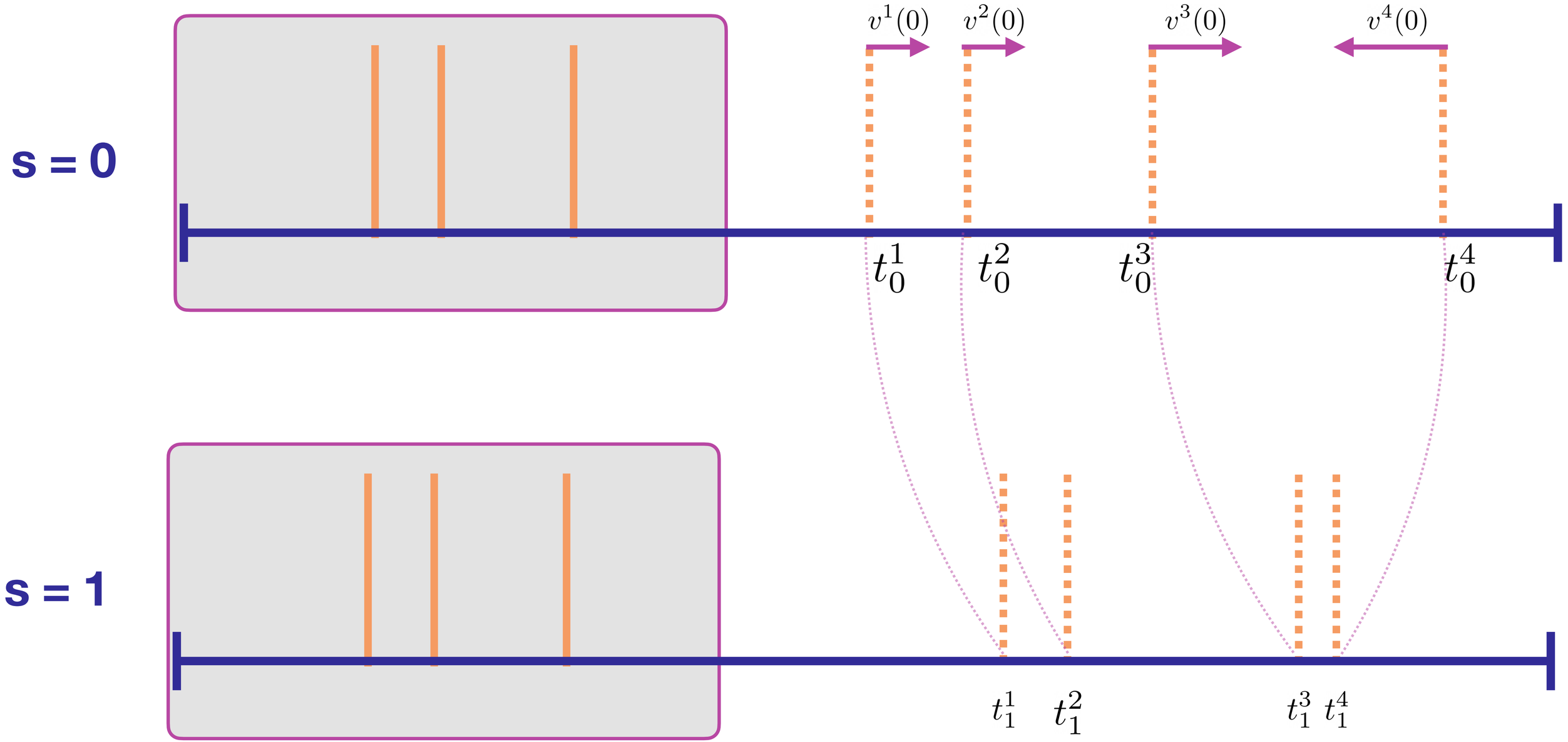


History

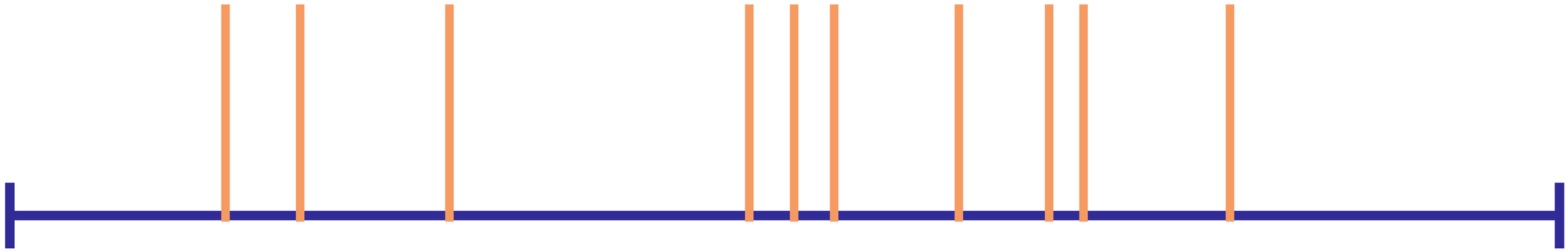
s = 0



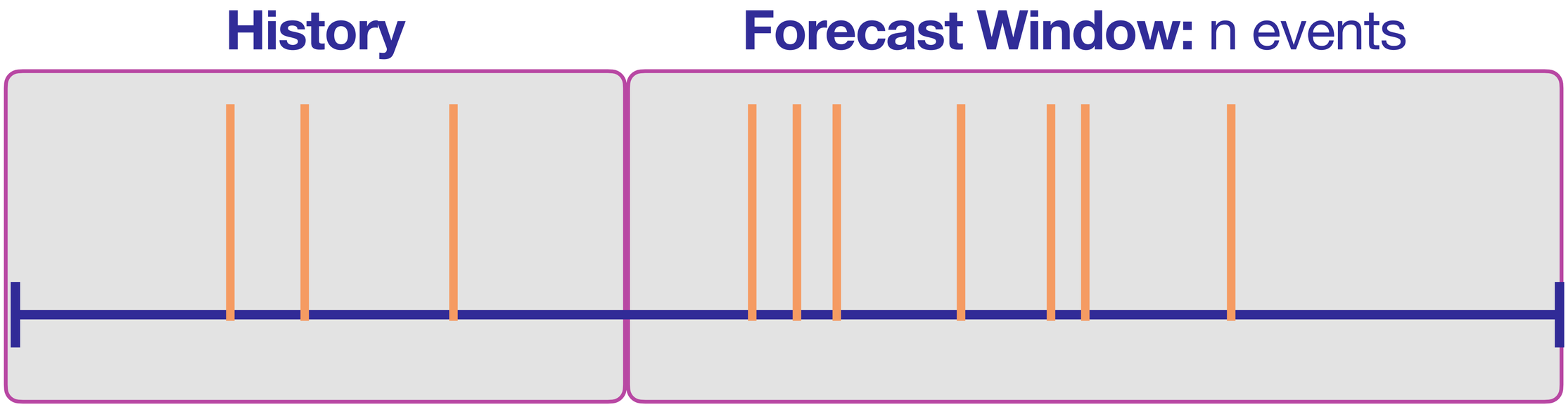
History



Training EventFlow



Training EventFlow



Training EventFlow

Inspired by Flow Matching [Lipman et al. 2023; Tong et al. 2024]

[1] Sample n random initial events $t_0^i \sim q(\cdot) \qquad i = 1, 2, \ldots, n$

Training EventFlow

Inspired by Flow Matching [Lipman et al. 2023; Tong et al. 2024]

[1] Sample n random initial events $t_0^i \sim q(\cdot) \qquad i = 1, 2, \ldots, n$

[2] Sort and pair with real events t_1^i

Training EventFlow

Inspired by Flow Matching [Lipman et al. 2023; Tong et al. 2024]

[1] Sample n random initial events $t_0^i \sim q(\cdot) \qquad i = 1, 2, \ldots, n$

[2] Sort and pair with real events t_1^i

[3] Sample flow time uniformly and interpolate $t_s^i = (1 - s)t_0^i + st_1^i$

Training EventFlow

Inspired by Flow Matching [Lipman et al. 2023; Tong et al. 2024]

[1] Sample n random initial events $t_0^i \sim q(\cdot) \qquad i = 1, 2, \ldots, n$

[2] Sort and pair with real events t_1^i

[3] Sample flow time uniformly and interpolate $t_s^i = (1-s)t_0^i + st_1^i$

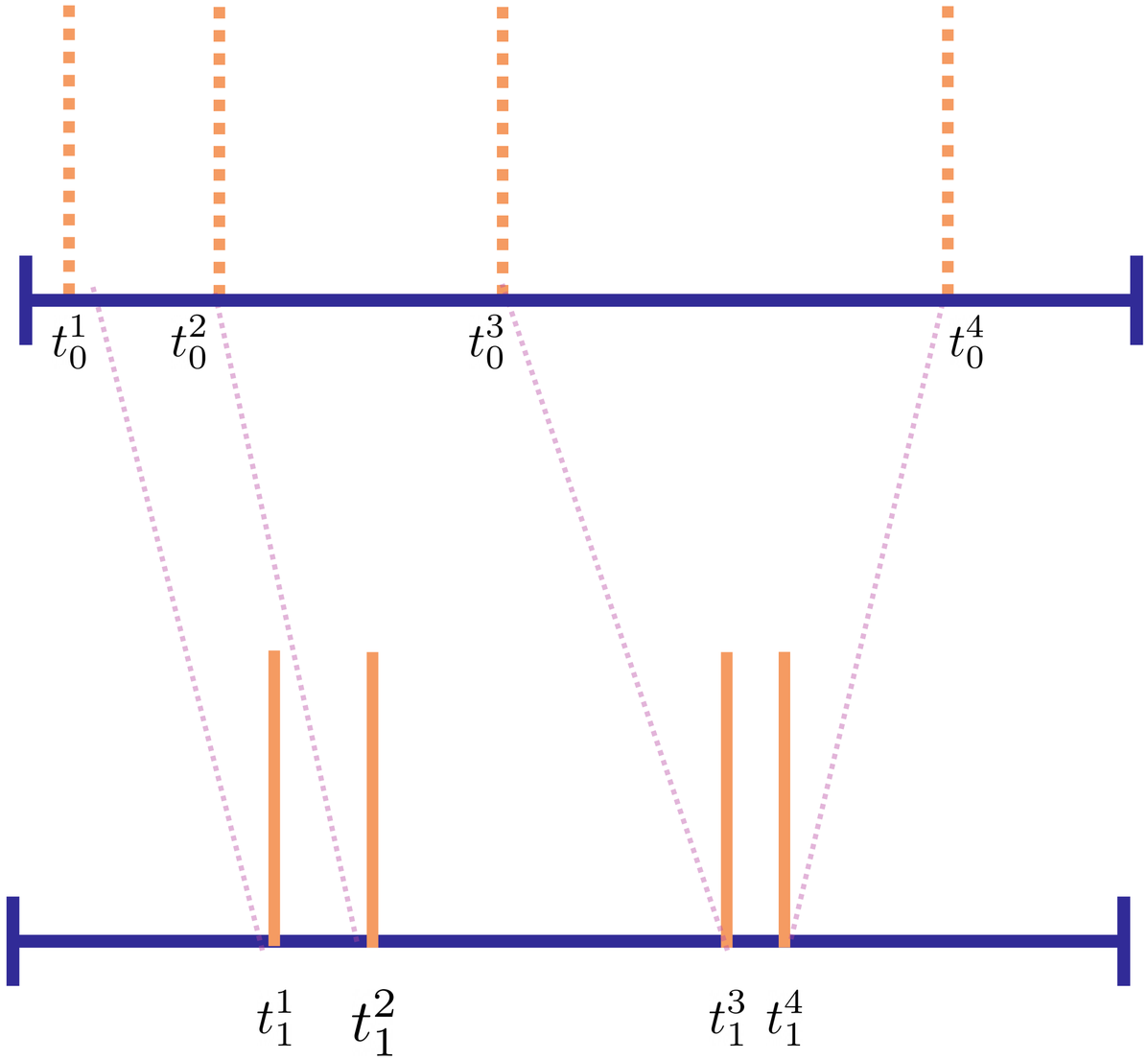
[4] Use noisy version to predict straight-line velocity

$$\sum_i |v_\theta^i(s, \mathbf{t}_s + \epsilon \mid \mathcal{H}) - (t_1^i - t_0^i)|^2 \qquad \epsilon \sim \mathcal{N}(0, \sigma^2 I)$$

Why does this work?

Interpolants define a path of sequences

$$\mathbf{t}_s = (1-s)\mathbf{t}_0 + s\mathbf{t}_1$$

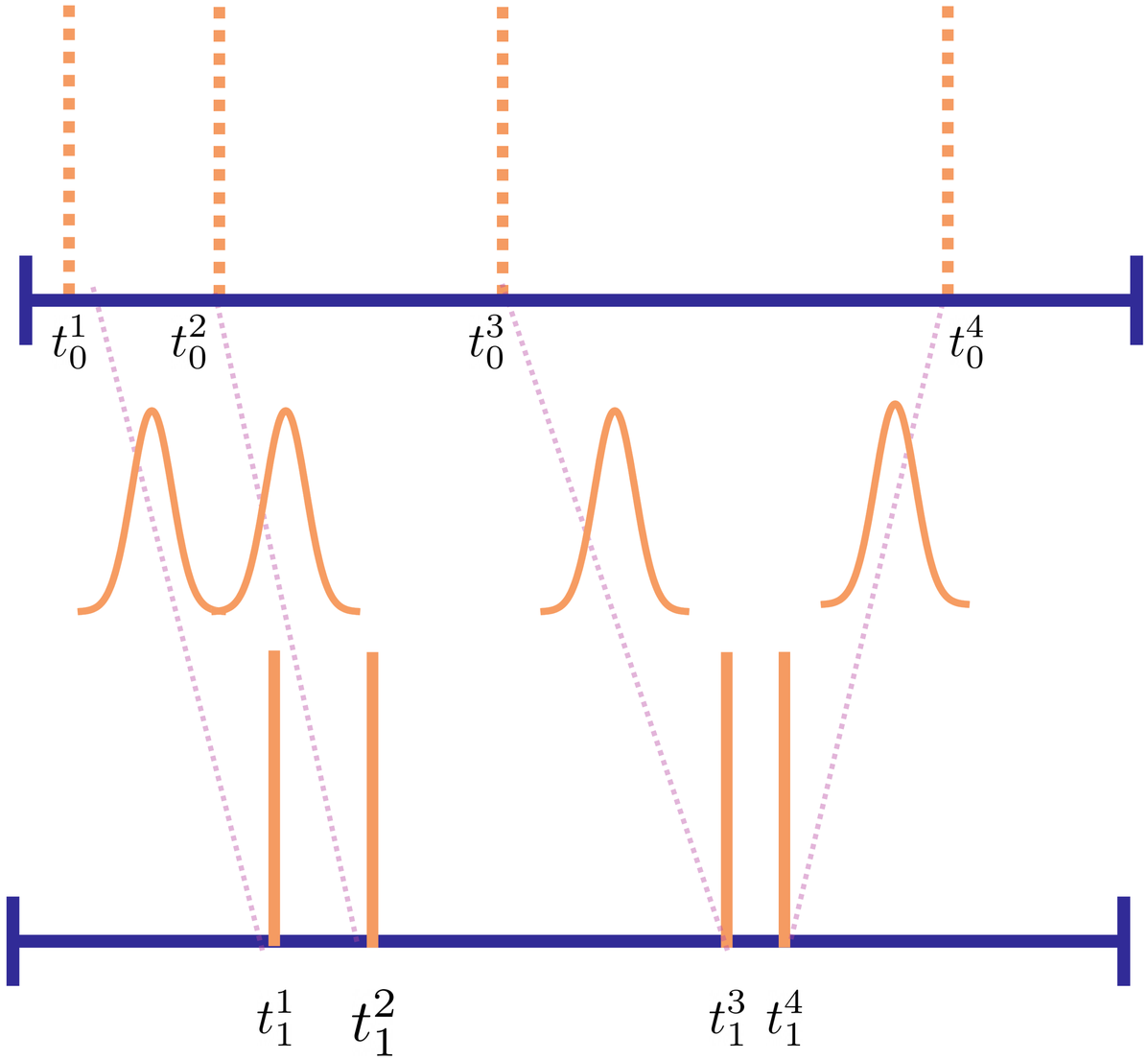


Why does this work?

Noising defines a path of distributions

$$\mathbf{t}_s = (1-s)\mathbf{t}_0 + s\mathbf{t}_1$$

$$p_s(\cdot \mid \mathbf{t}_0, \mathbf{t}_1) = \mathcal{N}(\mathbf{t}_s, \sigma^2 I)$$



Why does this work?

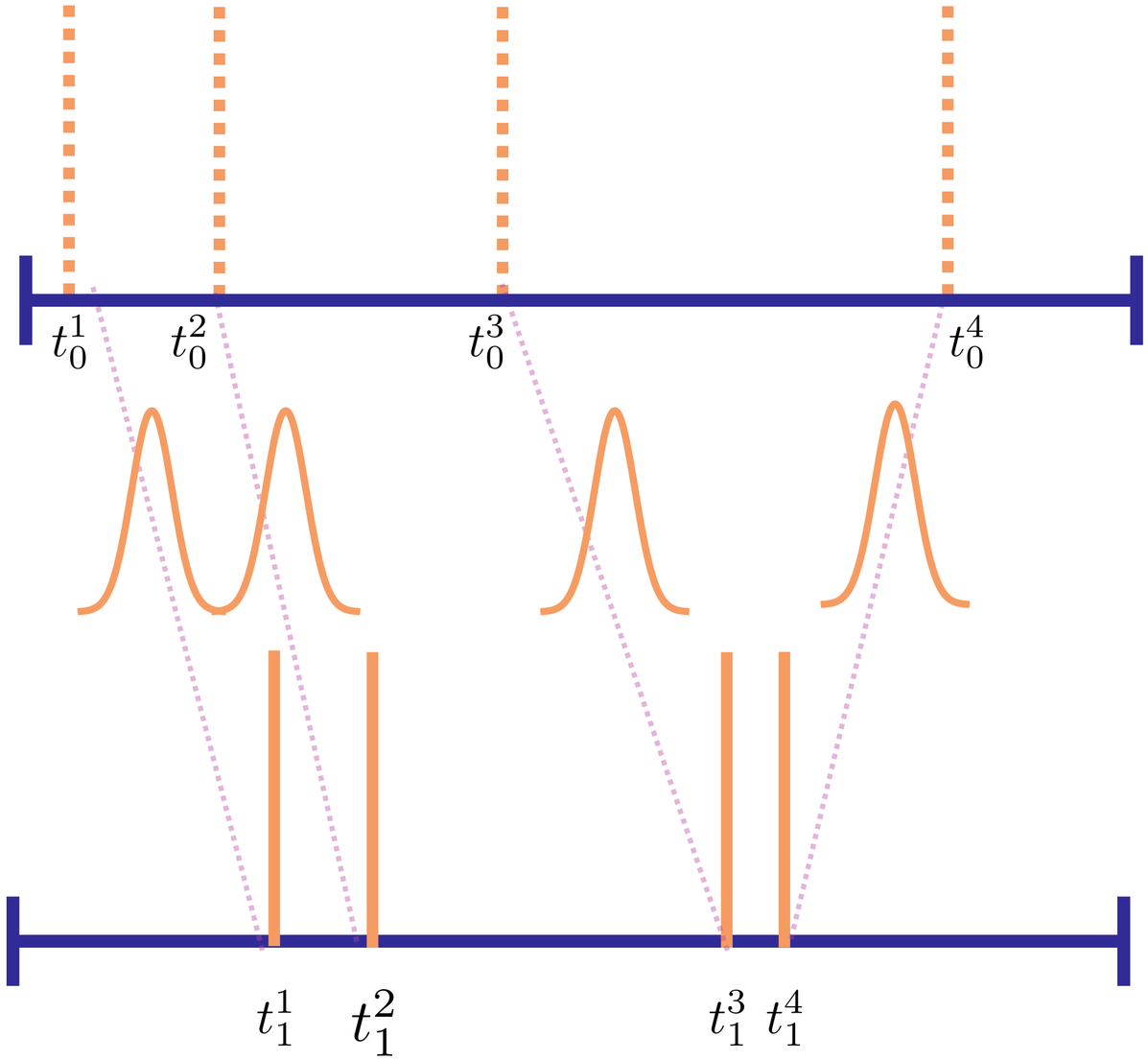
Noising defines a path of distributions

$$\mathbf{t}_s = (1-s)\mathbf{t}_0 + s\mathbf{t}_1$$

$$p_s(\cdot \mid \mathbf{t}_0, \mathbf{t}_1) = \mathcal{N}(\mathbf{t}_s, \sigma^2 I)$$

Marginalising gives a path of distributions
Random sequences —> Data Sequences

$$p_s(\cdot) = \int p_s(\cdot \mid \mathbf{t}_0, \mathbf{t}_1) d\pi(\mathbf{t}_0, \mathbf{t}_1)$$



Why does this work?

Marginal path of distributions
Interpolates random and data distributions

$$p_s(\cdot) = \int p_s(\cdot \mid \mathbf{t}_0, \mathbf{t}_1) d\pi(\mathbf{t}_0, \mathbf{t}_1)$$

Generated by the intractable vector field

$$v(s, \mathbf{t}) = \mathbb{E}_{\mathbf{t}_0, \mathbf{t}_1}[\mathbf{t}_1 - \mathbf{t}_0 \mid \mathbf{t_s} = t]$$

Why does this work?

Marginal path of distributions
Interpolates random and data distributions

$$p_s(\cdot) = \int p_s(\cdot \mid \mathbf{t}_0, \mathbf{t}_1) d\pi(\mathbf{t}_0, \mathbf{t}_1)$$

Generated by the intractable vector field

$$v(s, \mathbf{t}) = \mathbb{E}_{\mathbf{t}_0, \mathbf{t}_1}[\mathbf{t}_1 - \mathbf{t}_0 \mid \mathbf{t_s} = t]$$

Regressing on the marginal velocity is equivalent to regressing on the conditionals:

$$\mathbb{E}|v_\theta(s, \mathbf{t}_s + \epsilon) - (\mathbf{t}_1 - \mathbf{t}_0)|^2$$

EventFlow fixes the number of events throughout generation

How do we handle variable-length forecasts?

EventFlow fixes the number of events throughout generation

How do we handle variable-length forecasts?

Just predict the event count distribution!

EventFlow fixes the number of events throughout generation

How do we handle variable-length forecasts?

Just predict the event count distribution!

A strength of the method:

- Greatly simplifies the generative process

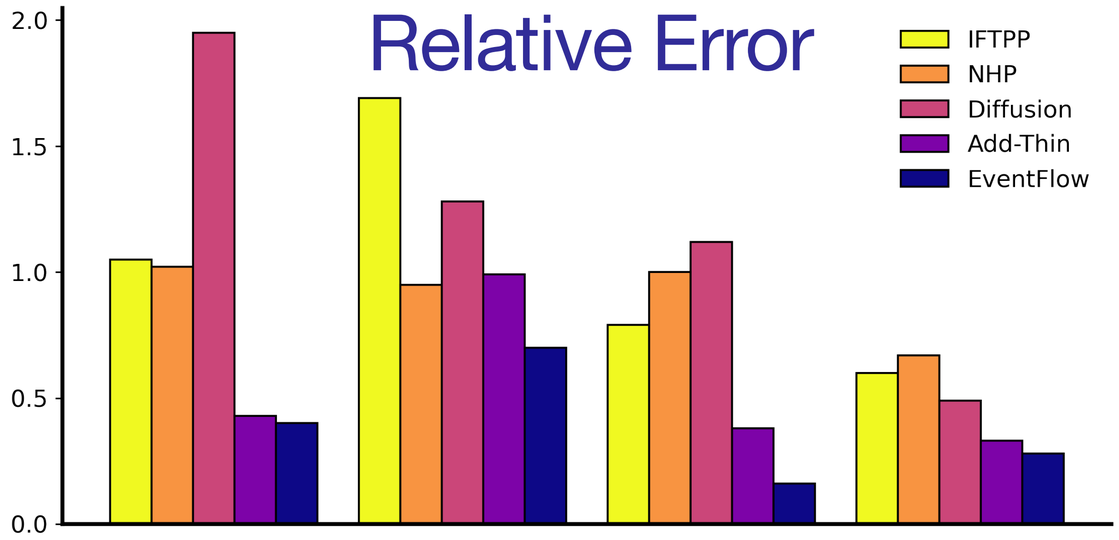
EventFlow fixes the number of events throughout generation

How do we handle variable-length forecasts?

Just predict the event count distribution!

A strength of the method:

- Greatly simplifies the generative process
- Autoregressive learn this implicitly; we directly train for it



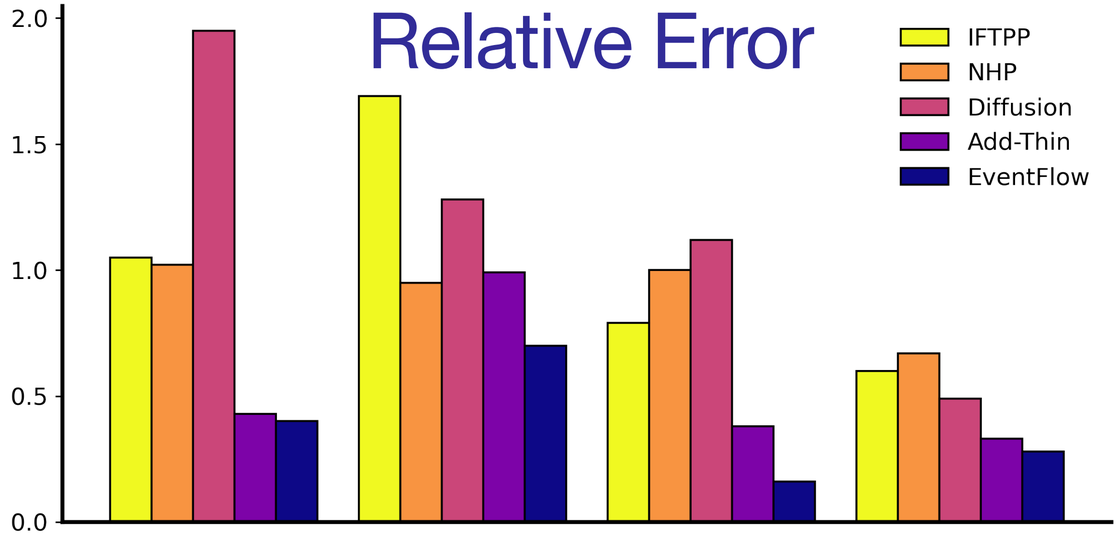
EventFlow fixes the number of events throughout generation

How do we handle variable-length forecasts?

Just predict the event count distribution!

A strength of the method:

- Greatly simplifies the generative process
- Autoregressive learn this implicitly; we directly train for it
- Gives explicit control
  - What would the forecast be if n was large/small?



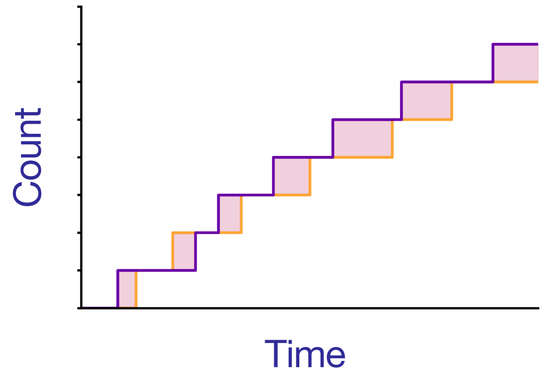
Experiments: Forecasting Real-World Event Sequences

Distance between sequences of **different lengths**:

[Xiao et al. 2017]

$$\mathbf{t} = \left(t^1, t^2, \ldots, t^n\right) \quad \hat{\mathbf{t}} = \left(\hat{t}^1, \hat{t}^2, \ldots, \hat{t}^m\right) \quad m > n$$

$$d(\mathbf{t}, \hat{\mathbf{t}}) = \sum_{k=1}^{n} \left|t^k - \hat{t}^k\right| + \sum_{k=n+1}^{m} \left(T - \hat{t}^k\right)$$



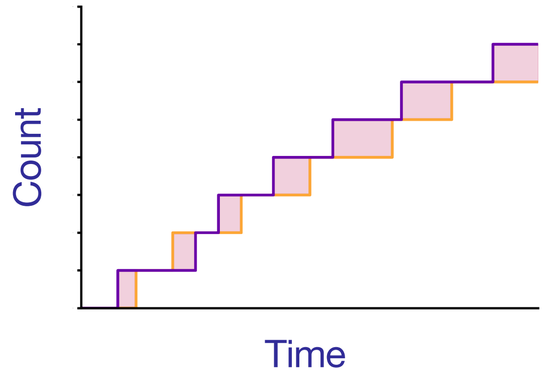
Experiments: Forecasting Real-World Event Sequences

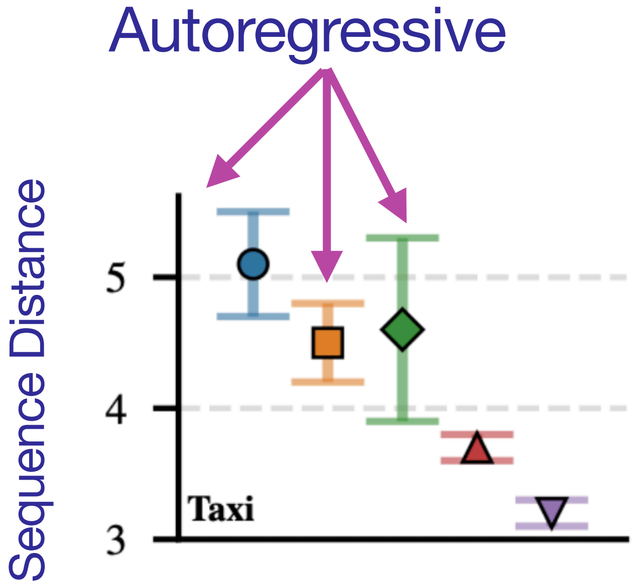
Distance between sequences of **different lengths**:

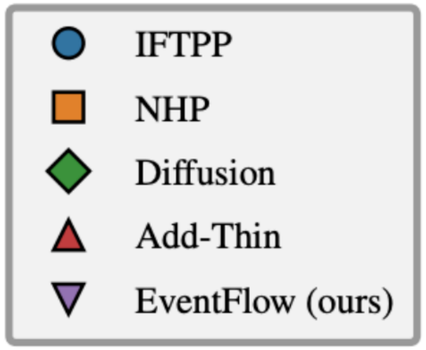
[Xiao et al. 2017]

$$\mathbf{t} = \left(t^1, t^2, \ldots, t^n\right) \quad \hat{\mathbf{t}} = \left(\hat{t}^1, \hat{t}^2, \ldots, \hat{t}^m\right) \quad m > n$$

$$d(\mathbf{t}, \hat{\mathbf{t}}) = \sum_{k=1}^{n} \left|t^k - \hat{t}^k\right| + \sum_{k=n+1}^{m} \left(T - \hat{t}^k\right)$$







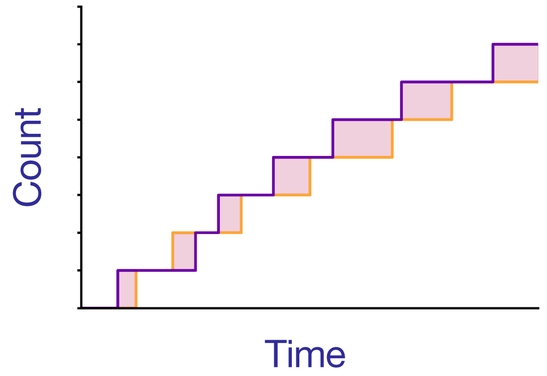
Experiments: Forecasting Real-World Event Sequences

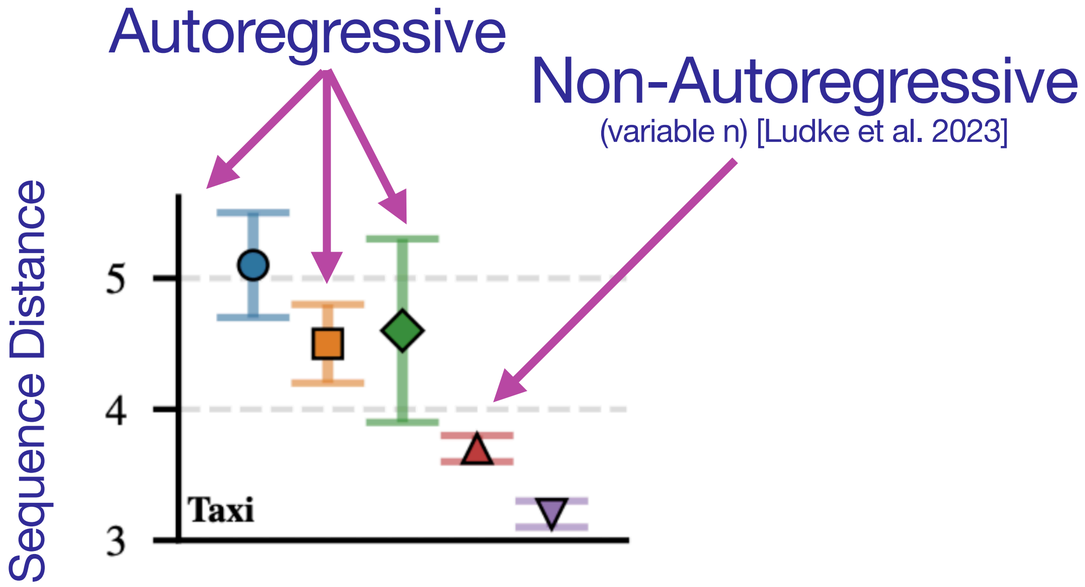
Distance between sequences of **different lengths**:

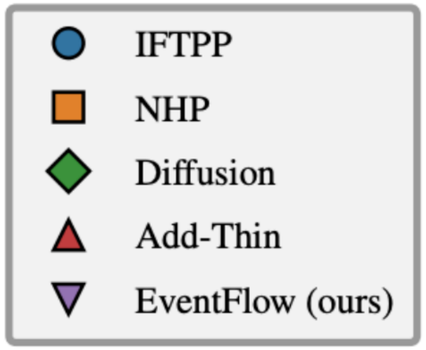
[Xiao et al. 2017]

$$\mathbf{t} = \left(t^1, t^2, \ldots, t^n\right) \quad \hat{\mathbf{t}} = \left(\hat{t}^1, \hat{t}^2, \ldots, \hat{t}^m\right) \quad m > n$$

$$d(\mathbf{t}, \hat{\mathbf{t}}) = \sum_{k=1}^{n} \left|t^k - \hat{t}^k\right| + \sum_{k=n+1}^{m} \left(T - \hat{t}^k\right)$$







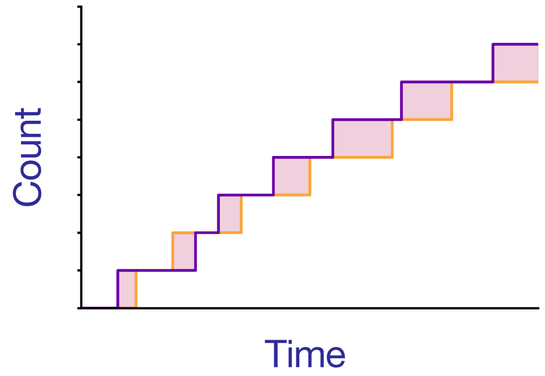
Experiments: Forecasting Real-World Event Sequences

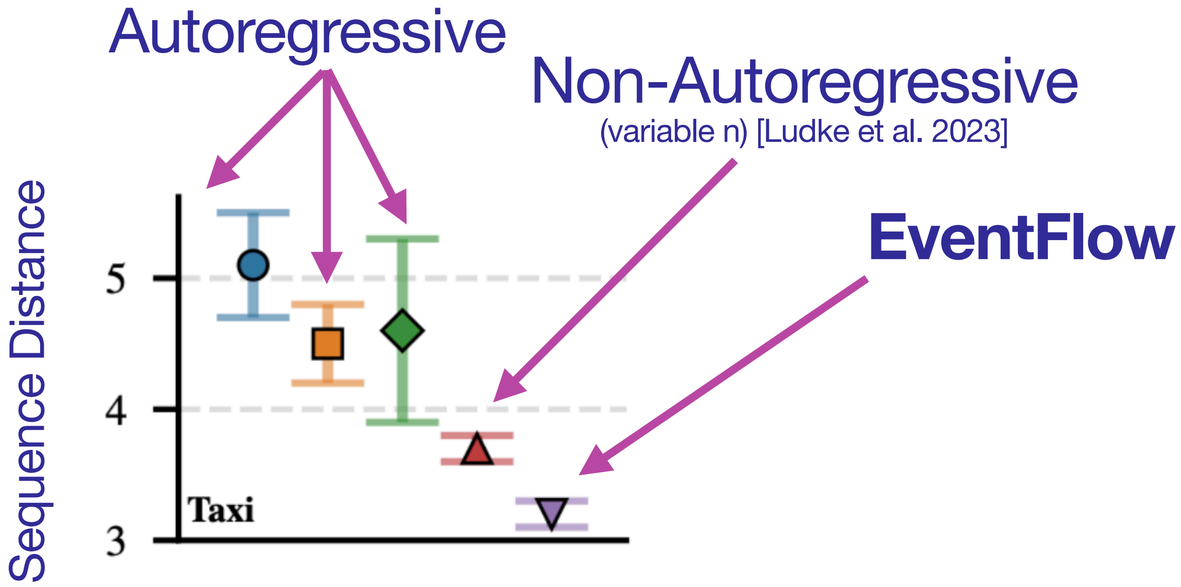
Distance between sequences of **different lengths**:

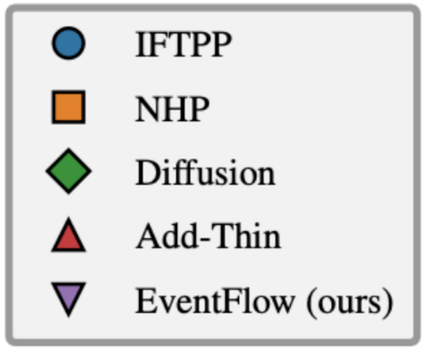
[Xiao et al. 2017]

$$\mathbf{t} = \left(t^1, t^2, \ldots, t^n\right) \quad \hat{\mathbf{t}} = \left(\hat{t}^1, \hat{t}^2, \ldots, \hat{t}^m\right) \quad m > n$$

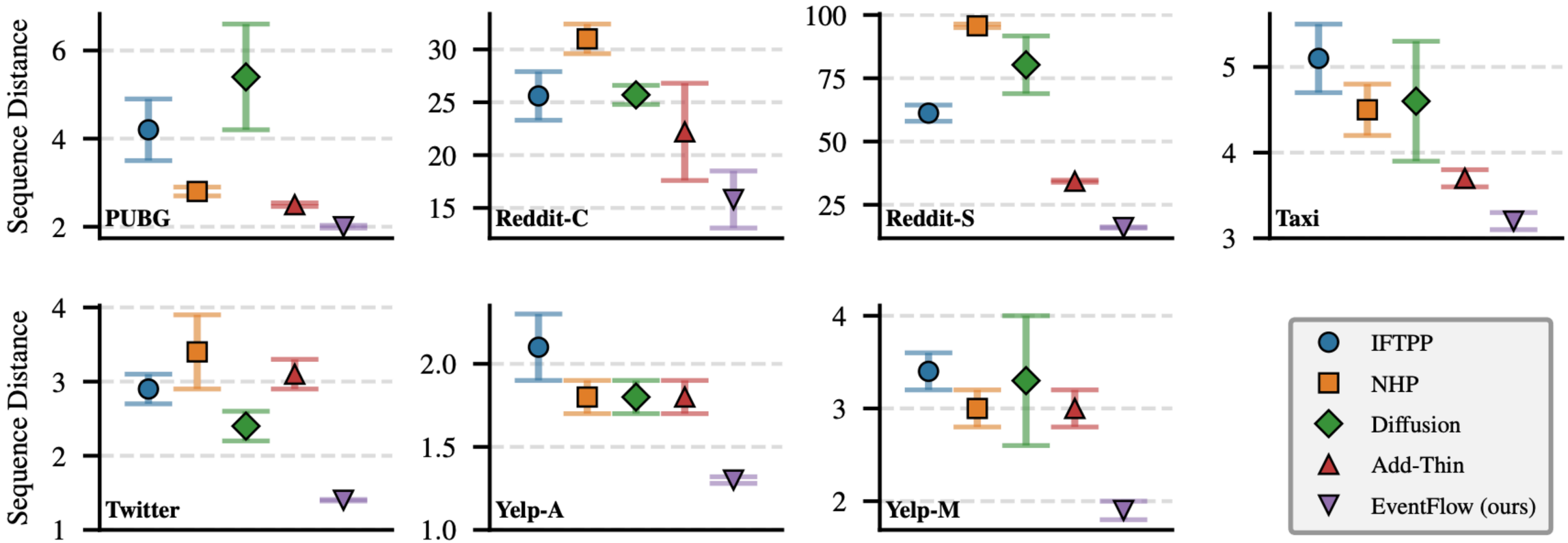
$$d(\mathbf{t}, \hat{\mathbf{t}}) = \sum_{k=1}^{n} \left|t^k - \hat{t}^k\right| + \sum_{k=n+1}^{m} \left(T - \hat{t}^k\right)$$







Experiments: Forecasting Real-World Event Sequences



Isn't this expensive?
Sampling requires solving a neural ODE

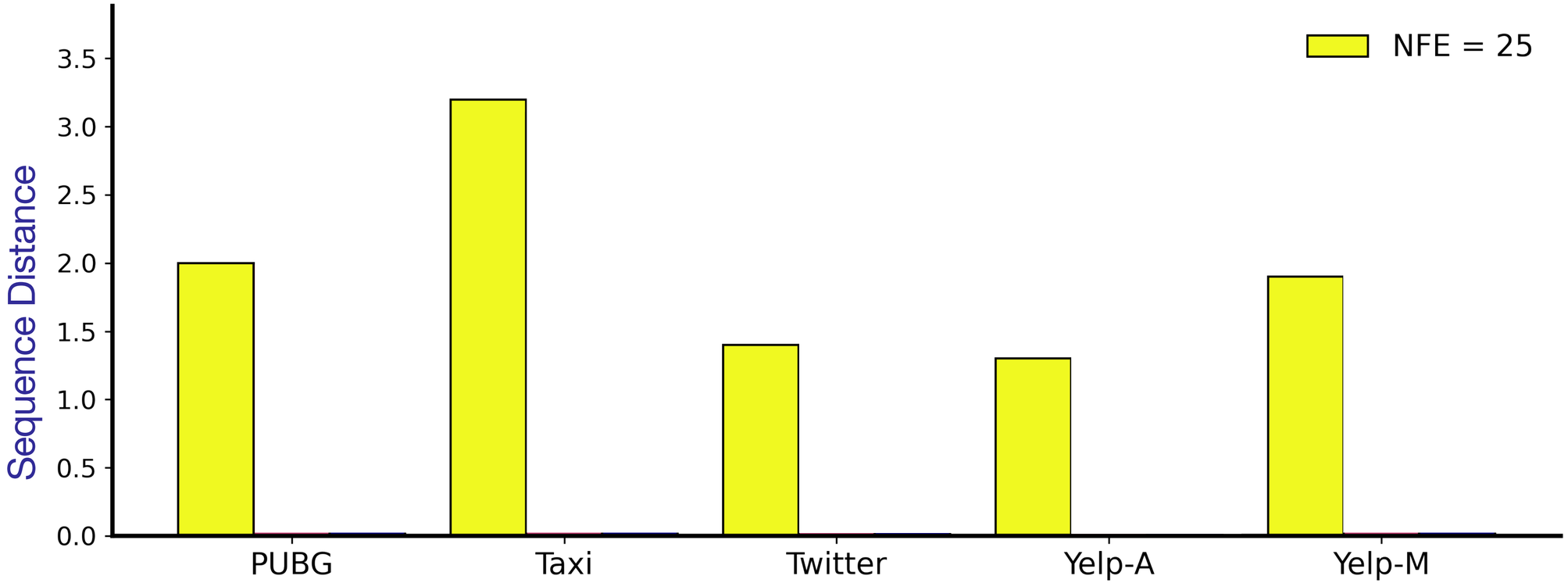
For n = 0, 1, … N:

$$t_{s_{n+1}} = v_\theta(s_n, \mathbf{t}_{s_n})\Delta s + t_{s_n}$$

Isn’t this expensive?
Sampling requires solving a neural ODE

For n = 0, 1, … N:

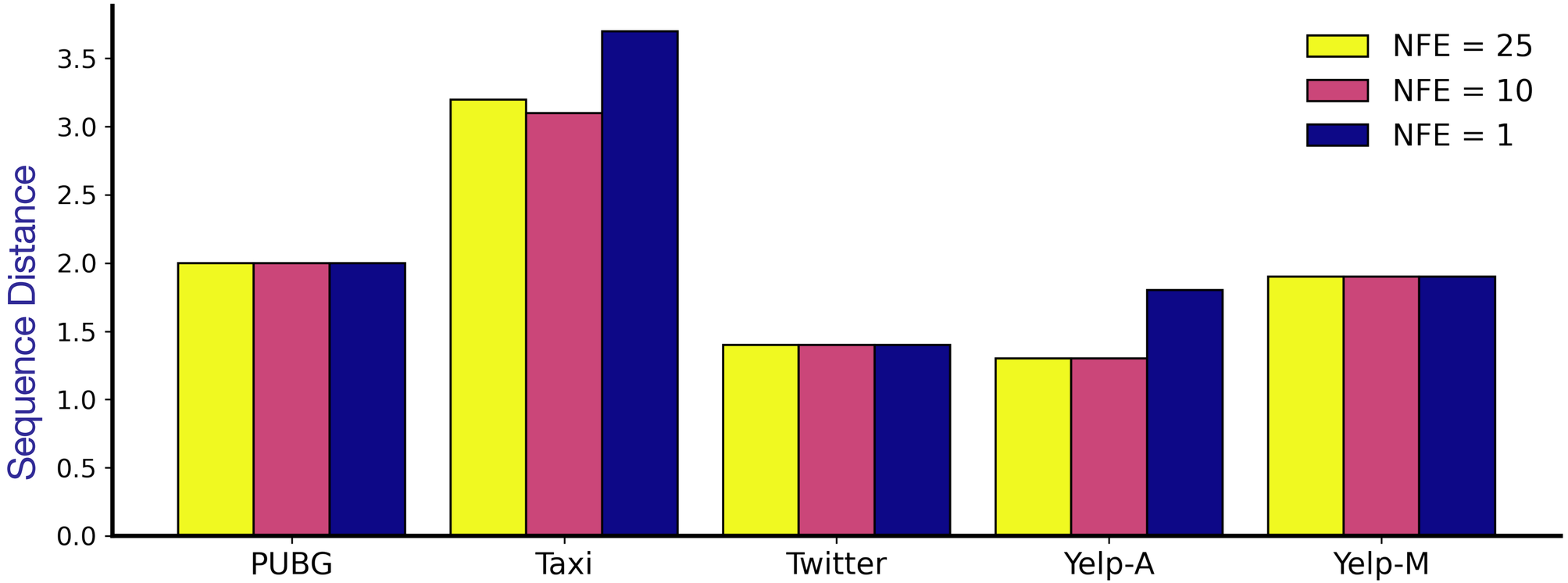
$$t_{s_{n+1}} = v_\theta(s_n, \mathbf{t}_{s_n})\Delta s + t_{s_n}$$



Isn't this expensive?
Sampling requires solving a neural ODE

For n = 0, 1, … N:

$$t_{s_{n+1}} = v_\theta(s_n, \mathbf{t}_{s_n})\Delta s + t_{s_n}$$



EventFlow is a simple and highly effective method for predicting irregular event sequences

Thank you!

EventFlow: Forecasting Temporal Point Processes with Flow Matching

Gavin Kerrigan[13], Kai Nelson[23], Padhraic Smyth[3]

[1]University of Oxford [2]UC Berkeley [3]UC Irvine

Poster #164