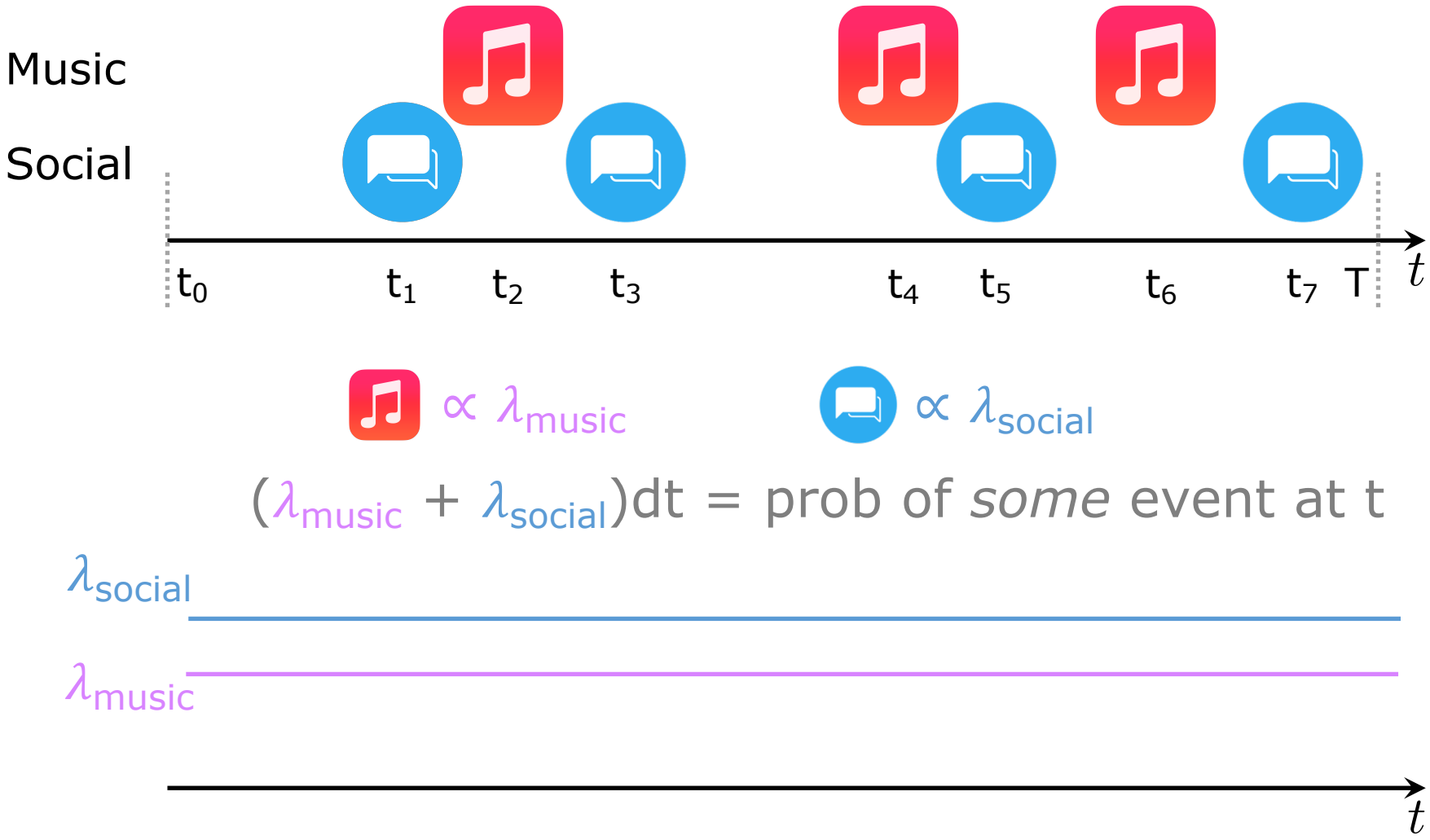
http://bburl/tpp-slides-p3
http://bburl/tpp-lab-p3

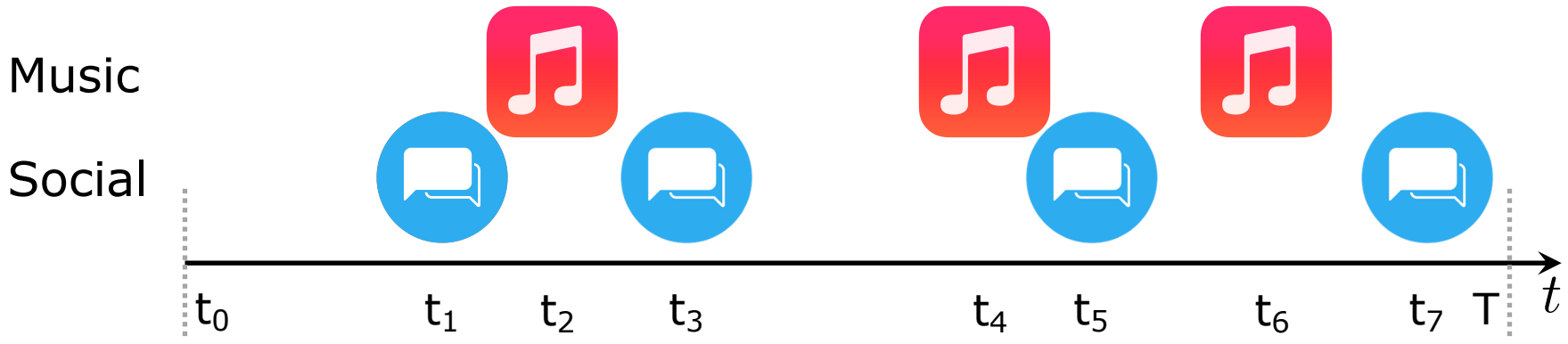
Modeling Irregular Time Series

Hongyuan Mei
2018—2021 Bloomberg Data Science Ph.D. Fellow
Johns Hopkins University

Multivariate Time Series



Multivariate Time Series: MLE



$$\lambda_{social} \exp(-\lambda (t_1 - t_0))$$
$$\times \lambda_{music} \exp(-\lambda (t_2 - t_1))$$
$$\times \lambda_{social} \exp(-\lambda (t_3 - t_2))$$
$$\dots \dots$$
$$= \lambda_{social}^4 \lambda_{music}^3 \exp(-\lambda (T - t_0))$$

$$(1-\lambda dt) \dots (1-\lambda dt)$$
$$\lambda = \lambda_{music} + \lambda_{social}$$

$$\lambda_{social} = 4 / (T - t_0)$$
$$\lambda_{music} = 3 / (T - t_0)$$

# of events of *that* type

Multivariate Time Series: MBR

Music

Social

$\lambda_{social} = 4 / (T - t_0)$

$\lambda_{music} = 3 / (T - t_0)$

t_0 s t

time

$t_{pred} = \int_s^\infty t\, p(t)\, dt \quad p(t) = \lambda \exp(-\lambda (t - s))$

$\lambda = \lambda_{music} + \lambda_{social} = 7 / (T - t_0)$

type?

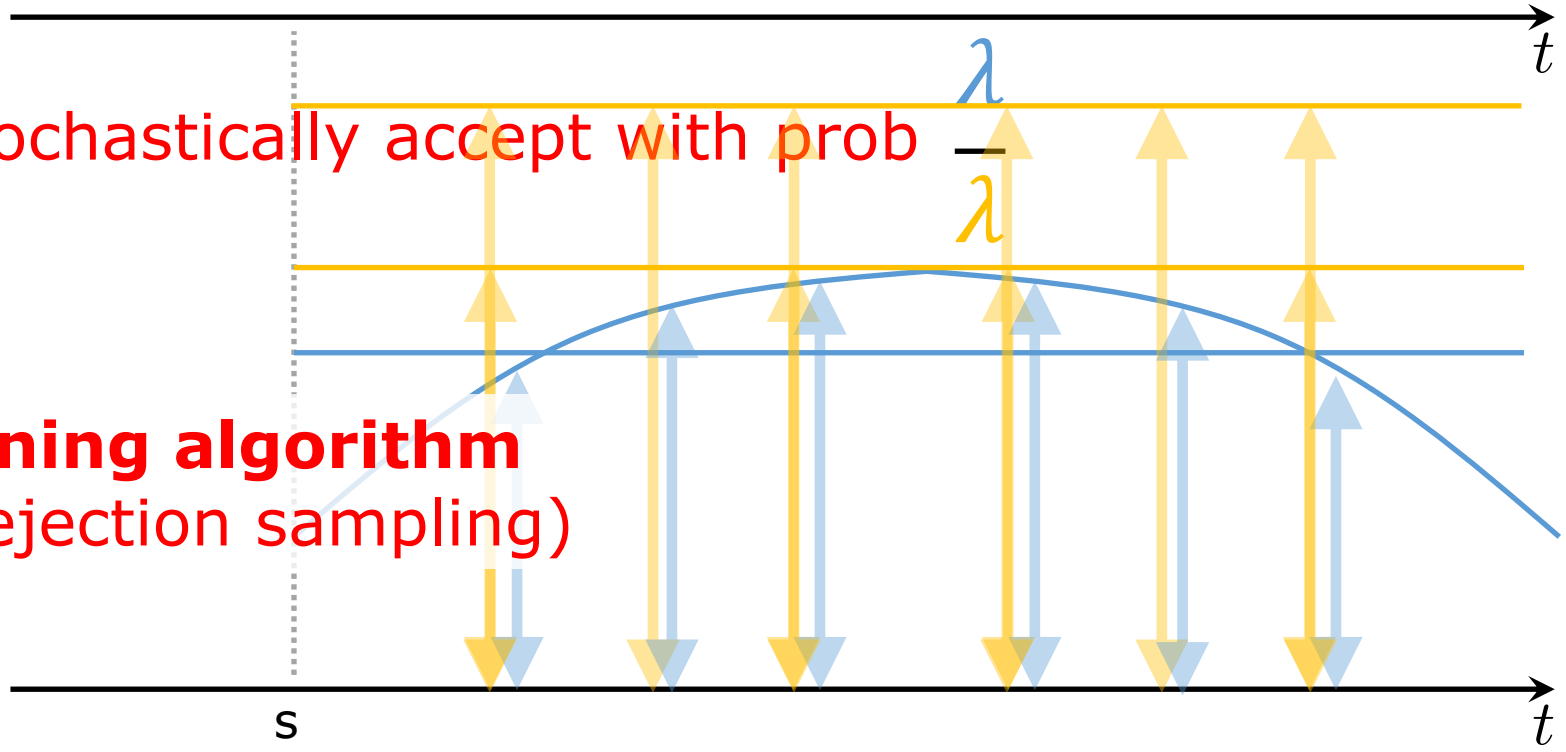
$k = \text{argmax}_k\, p(k \mid t) = \text{argmax}_k\, \lambda_k$

$\text{argmin}_k \sum_j p(j \mid t)\, \mathbb{I}(k \neq j) \quad j, k \in \{\text{music}, \text{social}\}$

choose type k that makes $\sum_{j \neq k} p(j \mid t)$ small

When intensities are not constant...

F^{-1} too complicated!!!

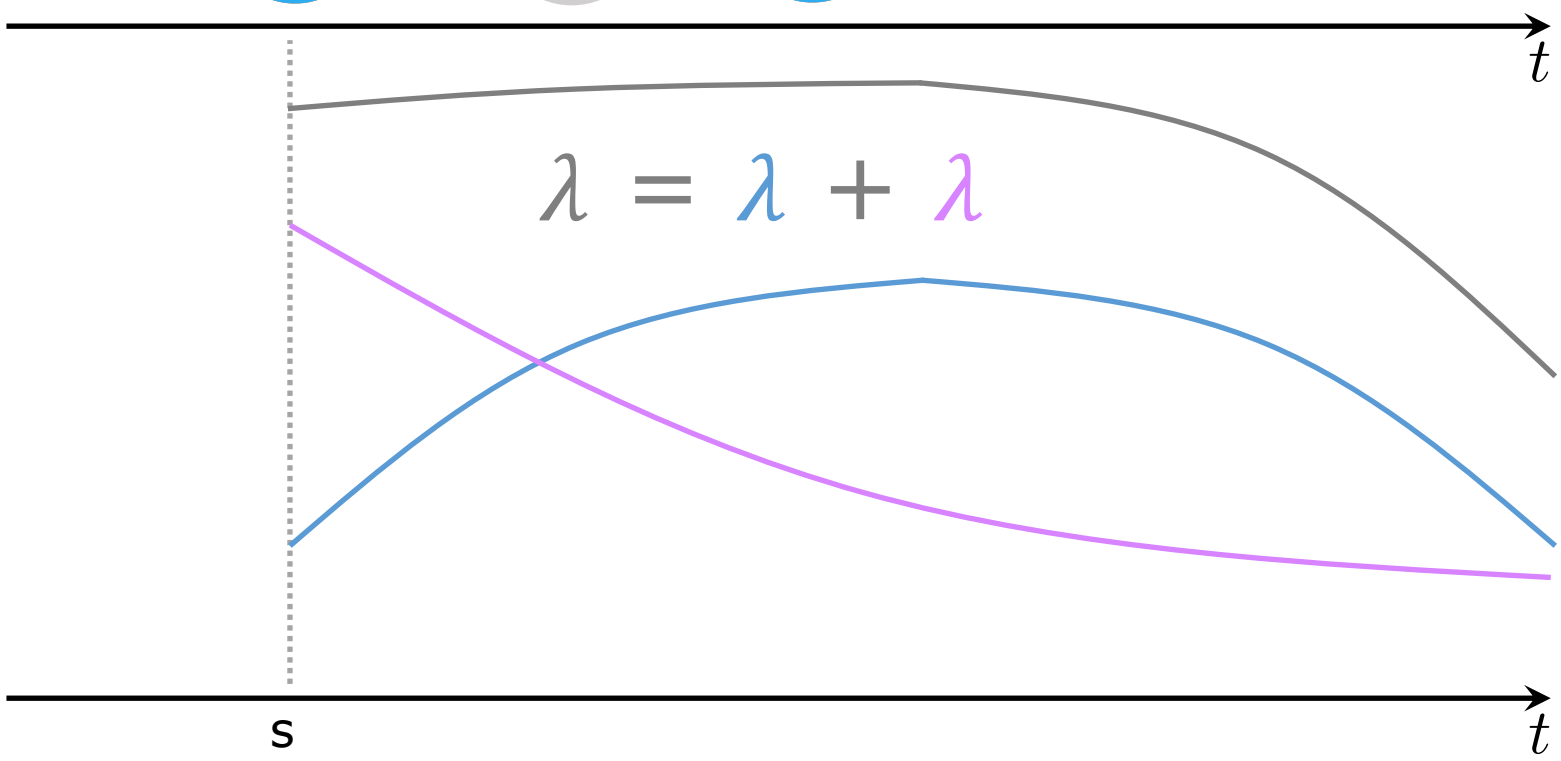


thinning algorithm
(c.f. rejection sampling)

Different types *race* against one another

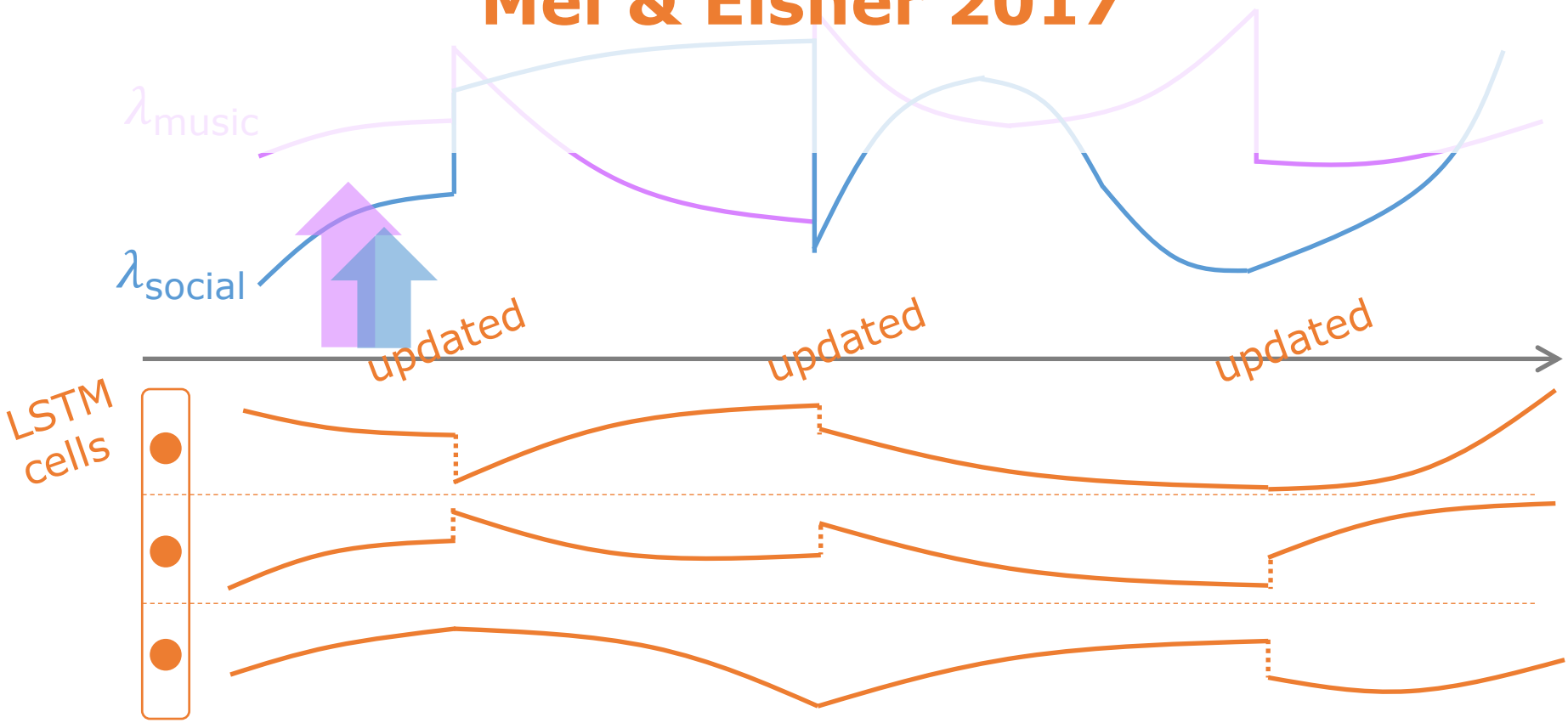
draw an event for each type

keep the earliest



Let LSTM state *evolve* with time

Neural Hawkes Process (NHP)
Mei & Eisner 2017



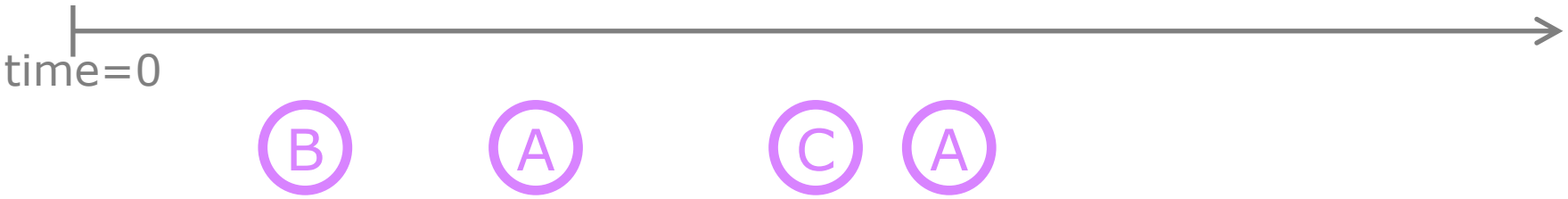
Lecture Structure

- Lecture-1: concepts
  - Intensity, point process, MLE, thinning, …
- Lecture-2: fancy models
  - Hawkes, NHP, Neural ODE, …
- Lecture-3: training + advanced topics
  - MLE & NCE
  - Large event space + domain knowledge

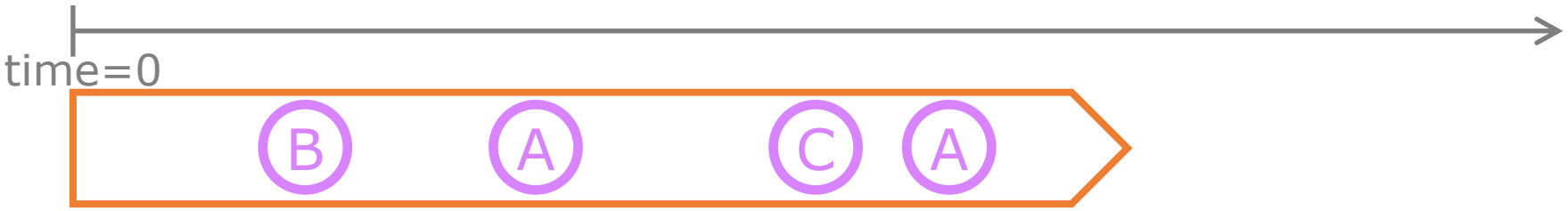
MLE: Max log prob of *data*

MLE: Max log prob of *data*

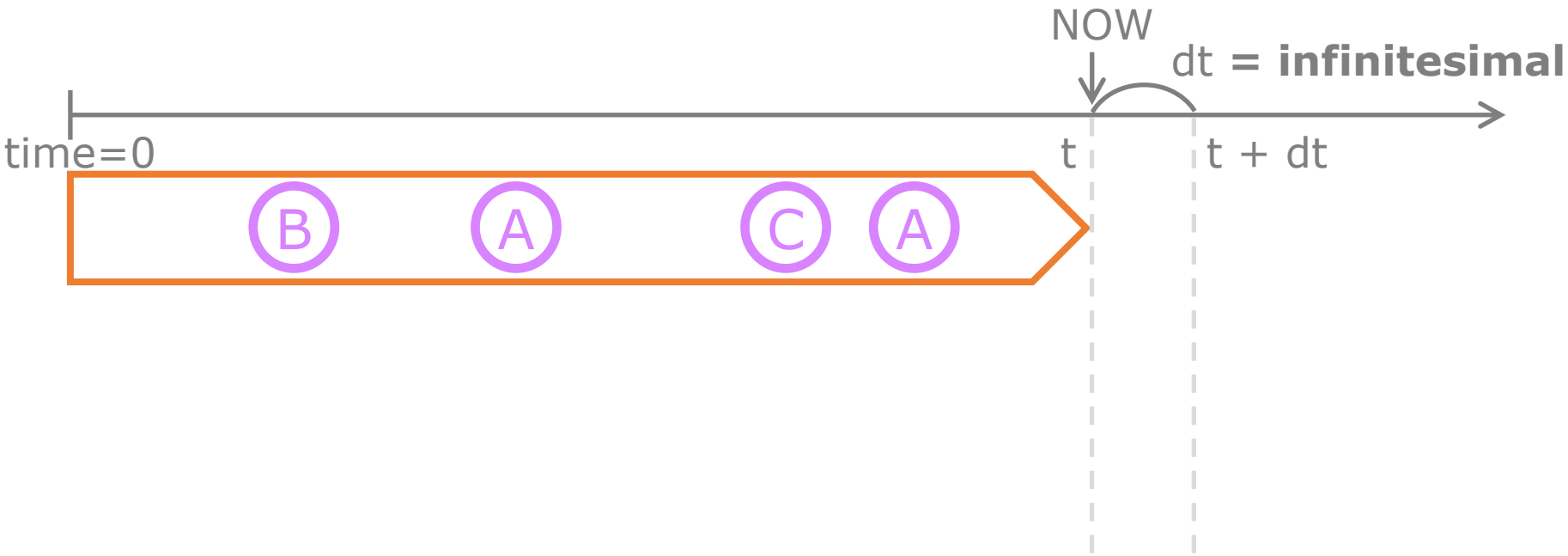
MLE: Max log prob of *data*



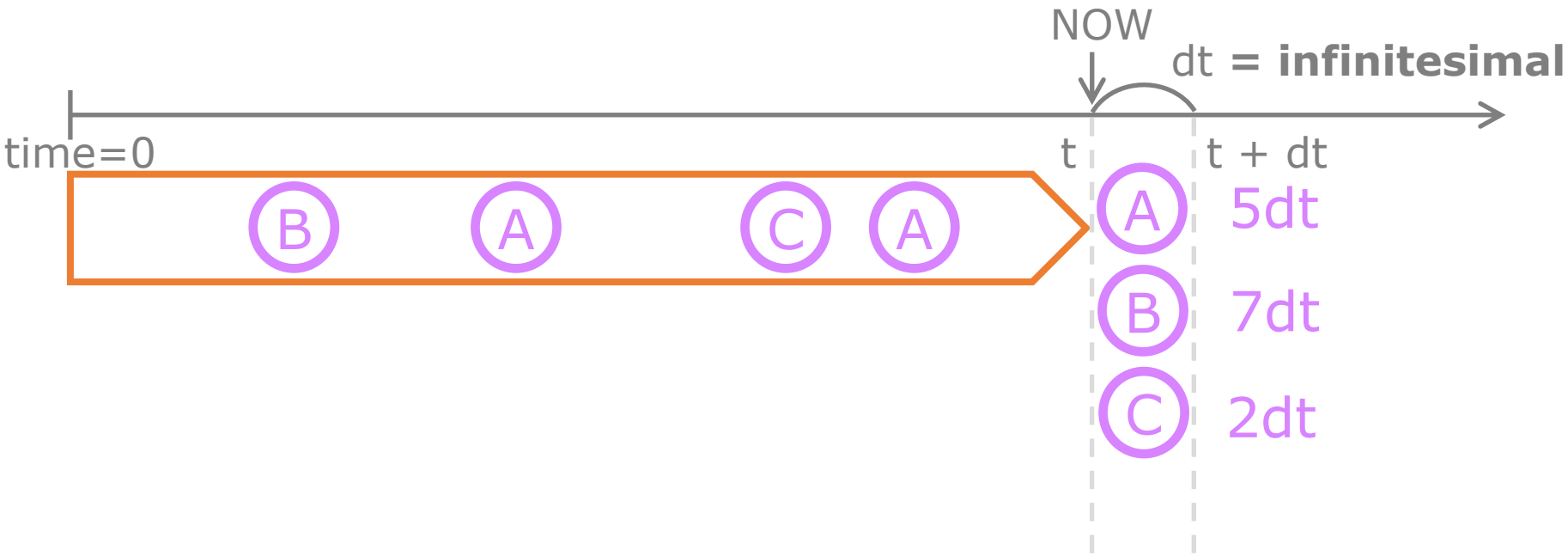
MLE: Max log prob of *data*



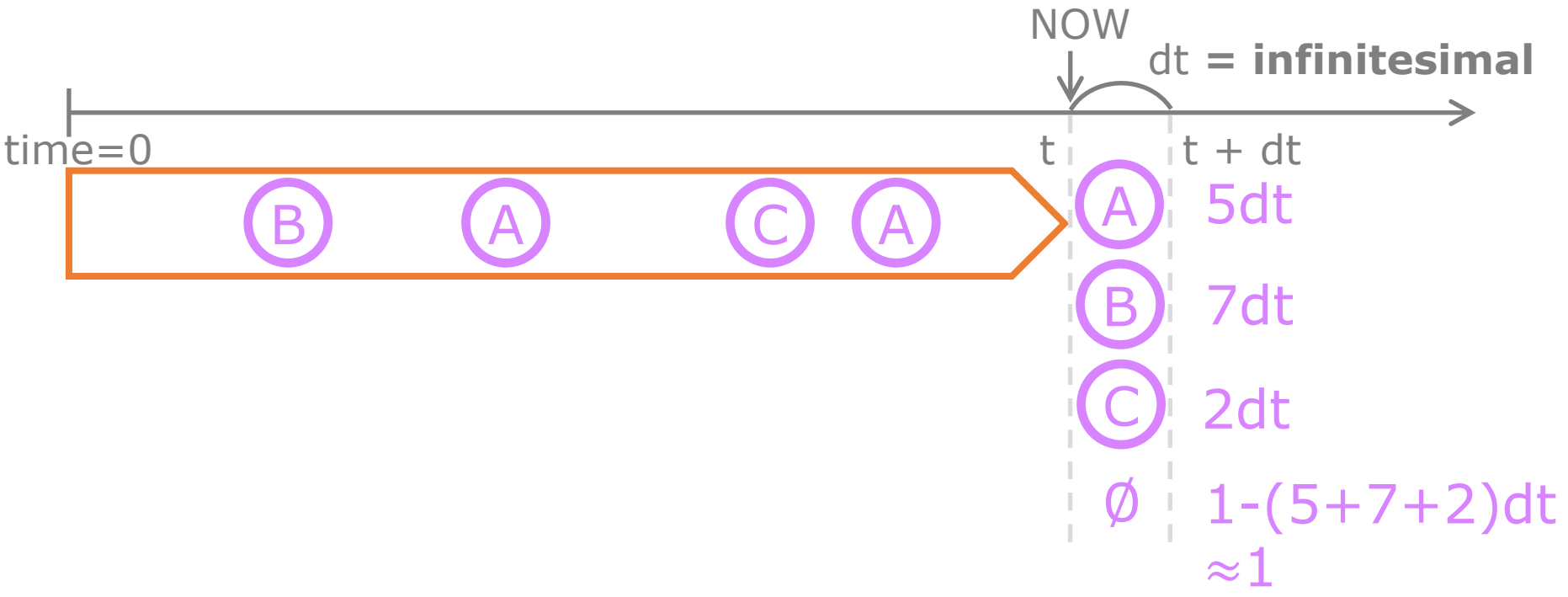
MLE: Max log prob of *data*



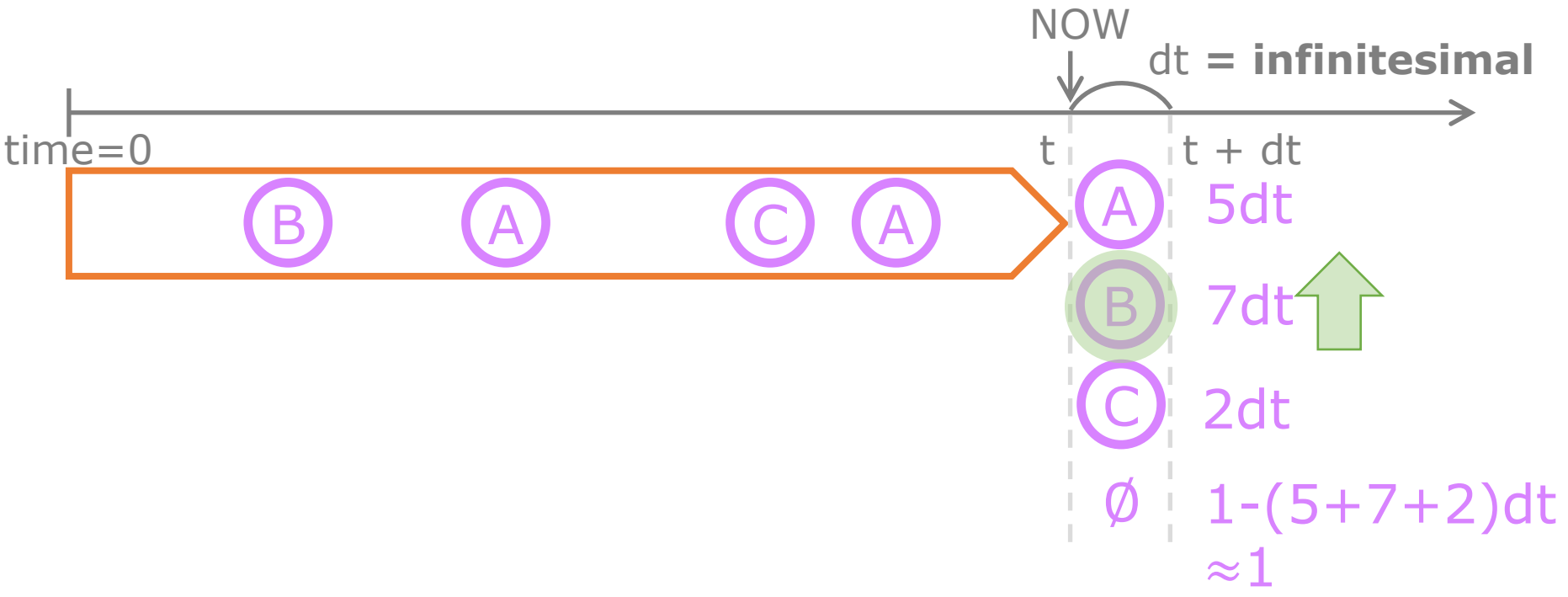
MLE: Max log prob of *data*



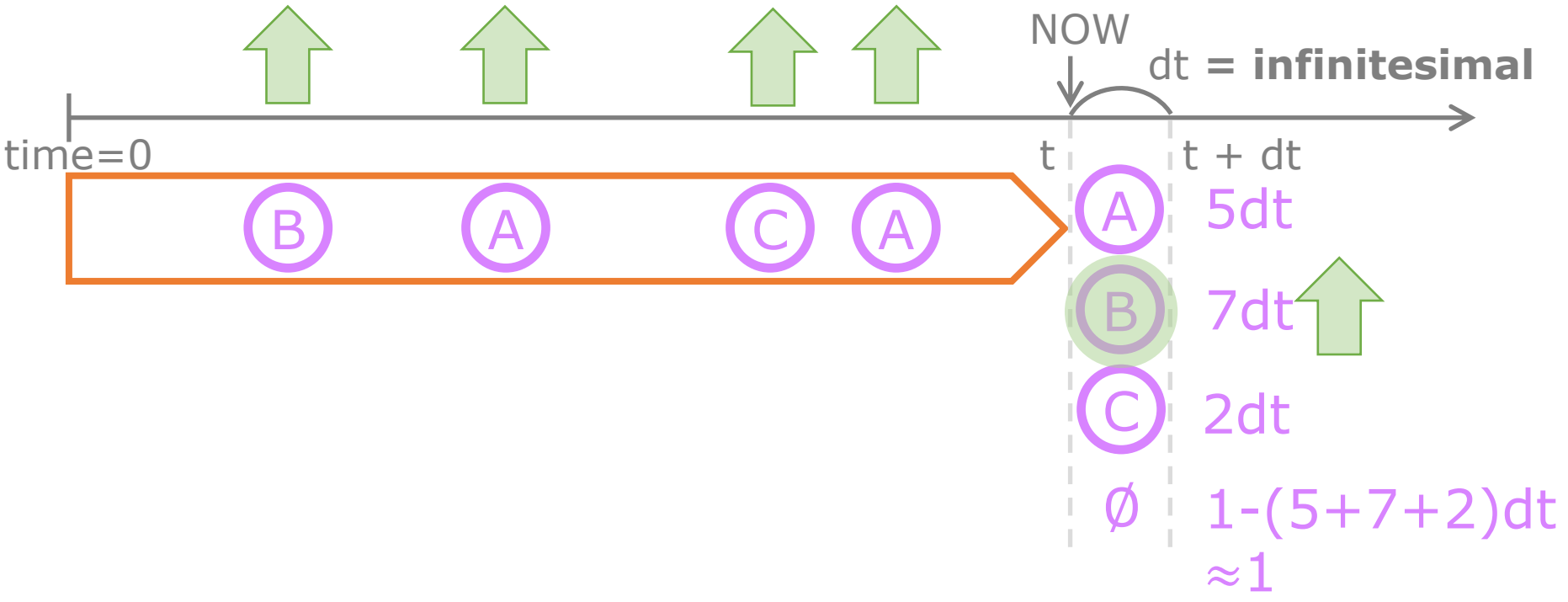
MLE: Max log prob of *data*



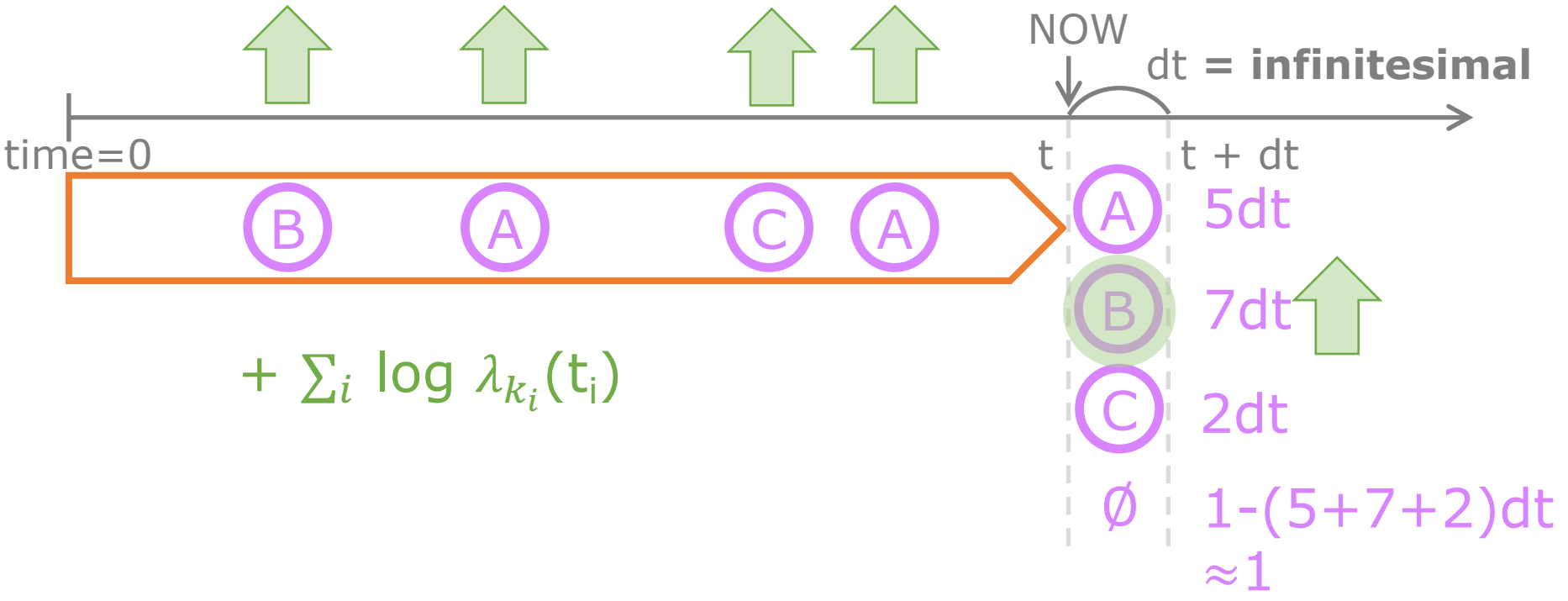
MLE: Max log prob of *data*



MLE: Max log prob of *data*

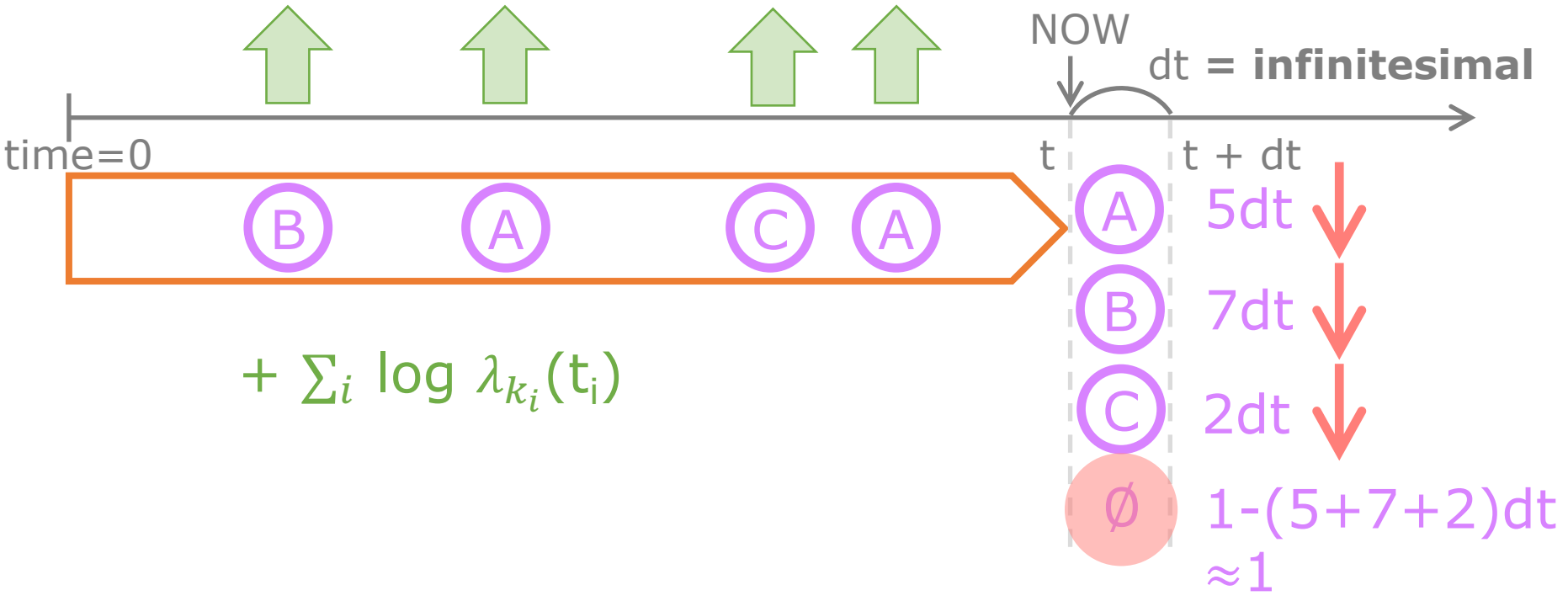


MLE: Max log prob of *data*



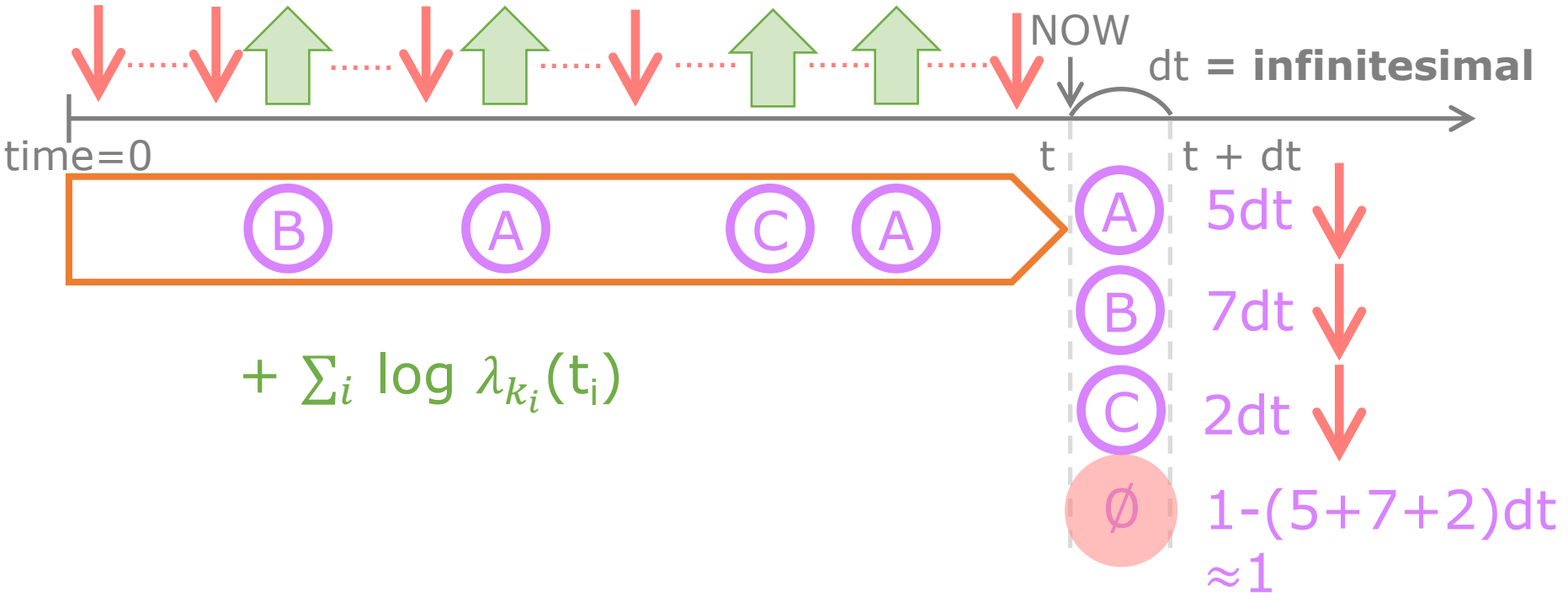
$+ \sum_i \log \lambda_{k_i}(t_i)$

MLE: Max log prob of *data*



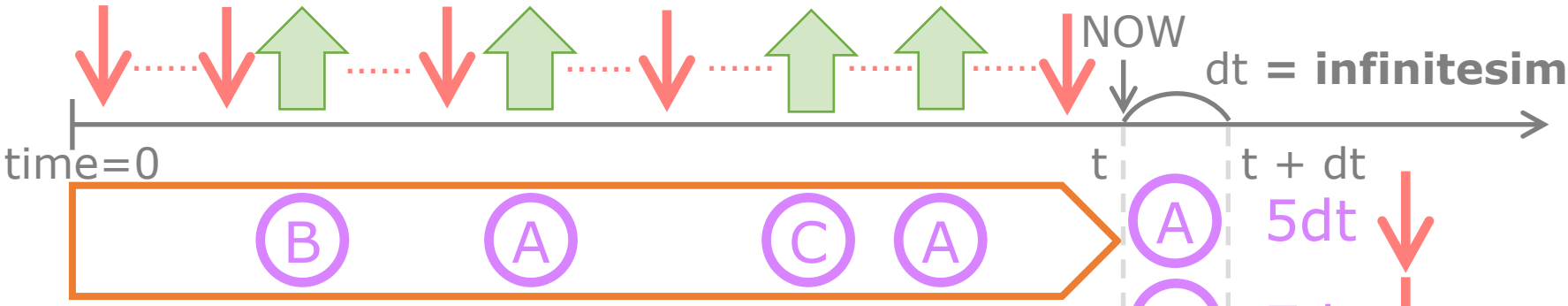
$+ \sum_i \log \lambda_{k_i}(t_i)$

MLE: Max log prob of *data*



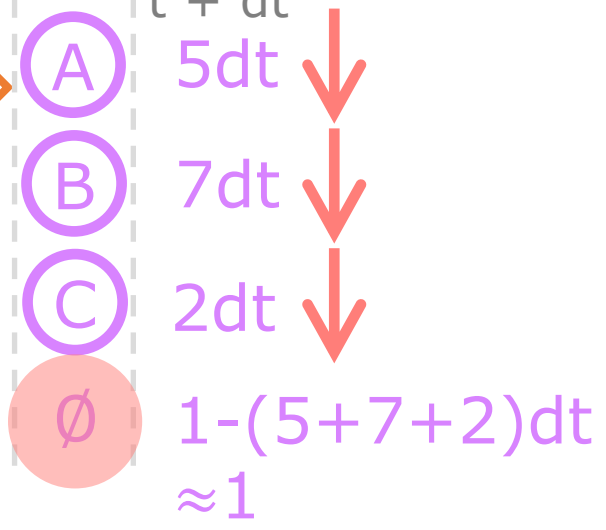
$+ \sum_i \log \lambda_{k_i}(t_i)$

MLE: Max log prob of *data*

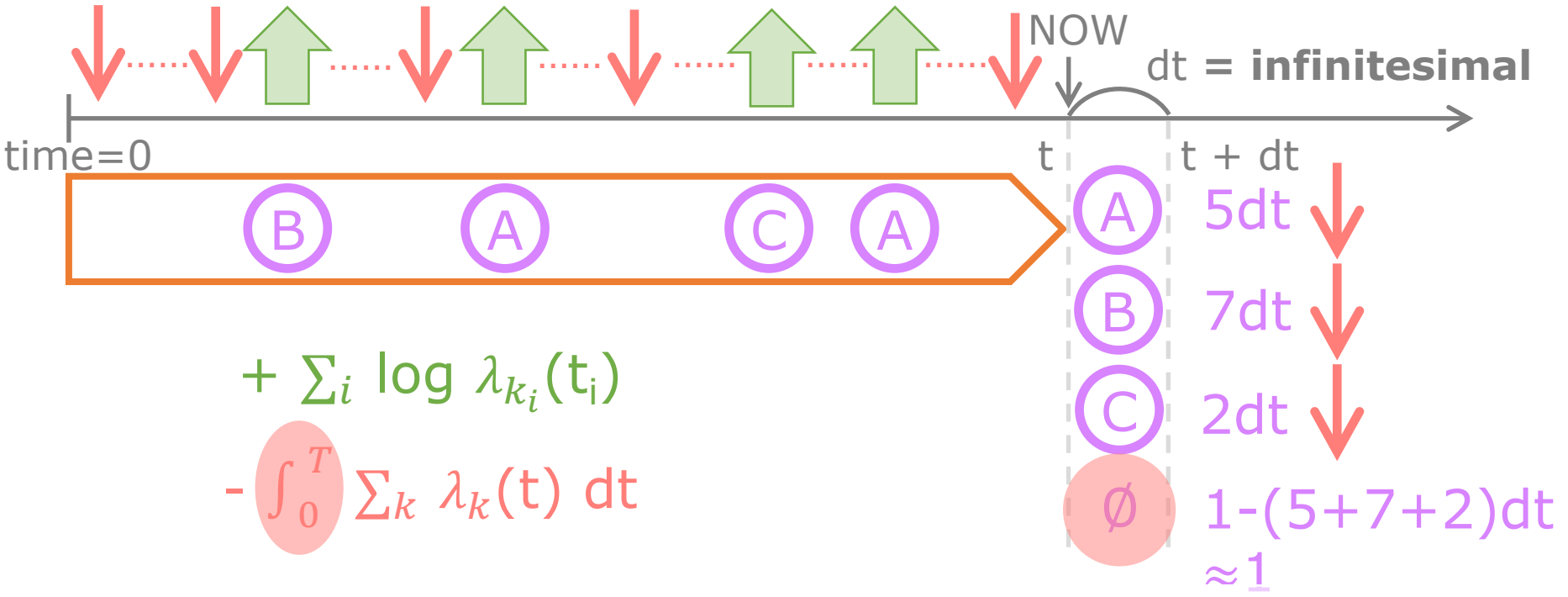


$$+ \sum_i \log \lambda_{k_i}(t_i)$$

$$- \int_0^T \sum_k \lambda_k(t)\, dt$$

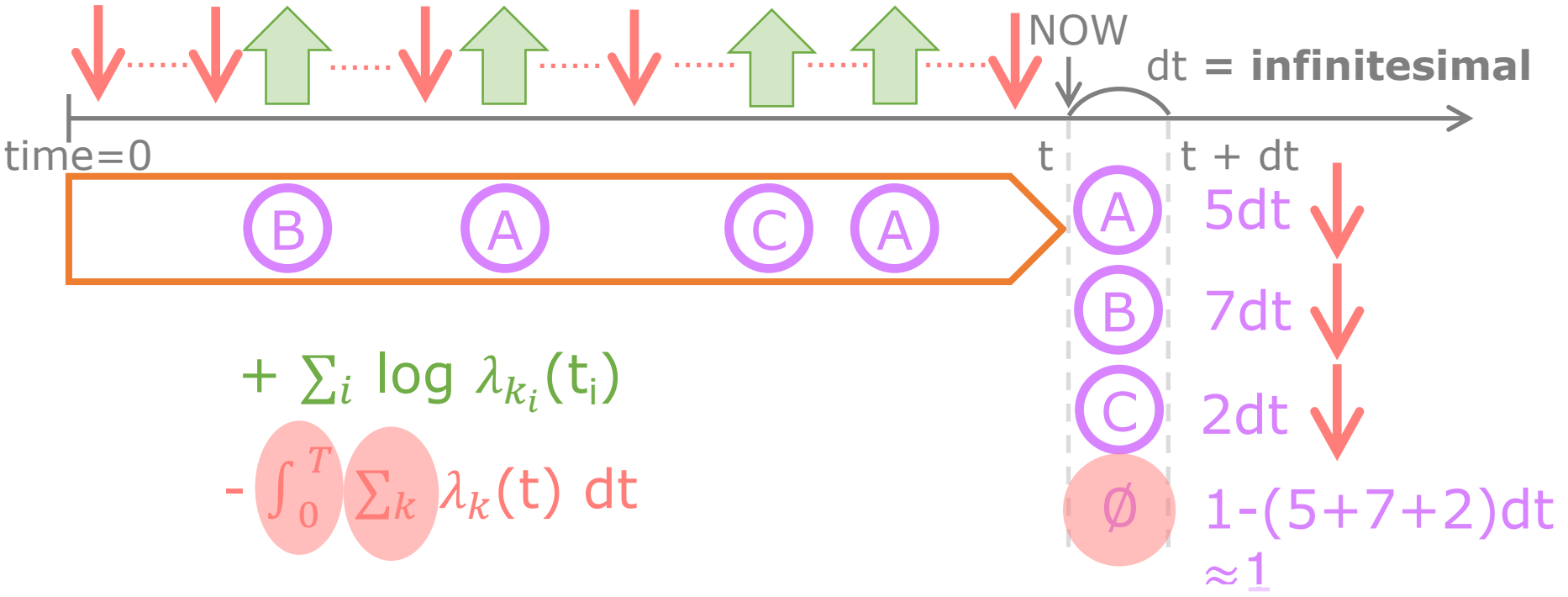


MLE: Max log prob of *data*



Integrate over <u>infinitely</u> many *non*-events!

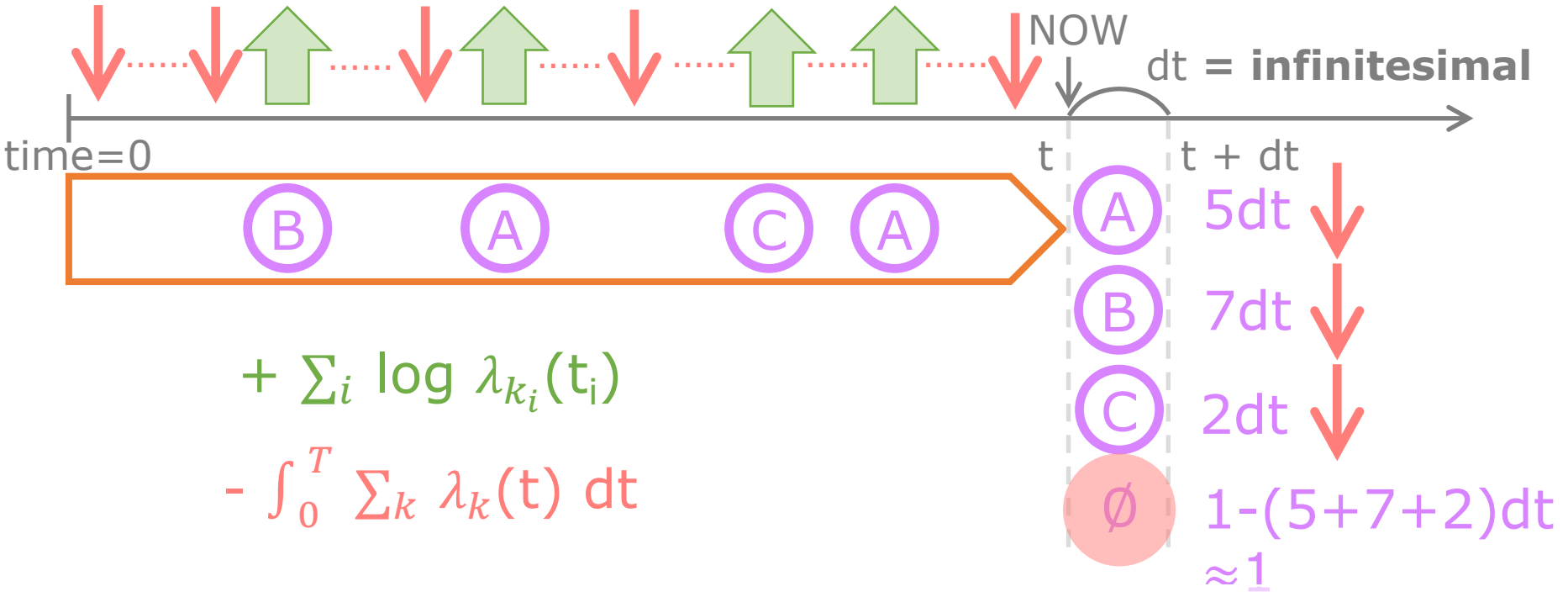
MLE: Max log prob of *data*



Integrate over <u>infinitely</u> many *non*-events!

Loop over <u>all</u> event types!

MLE: Max log prob of *data*



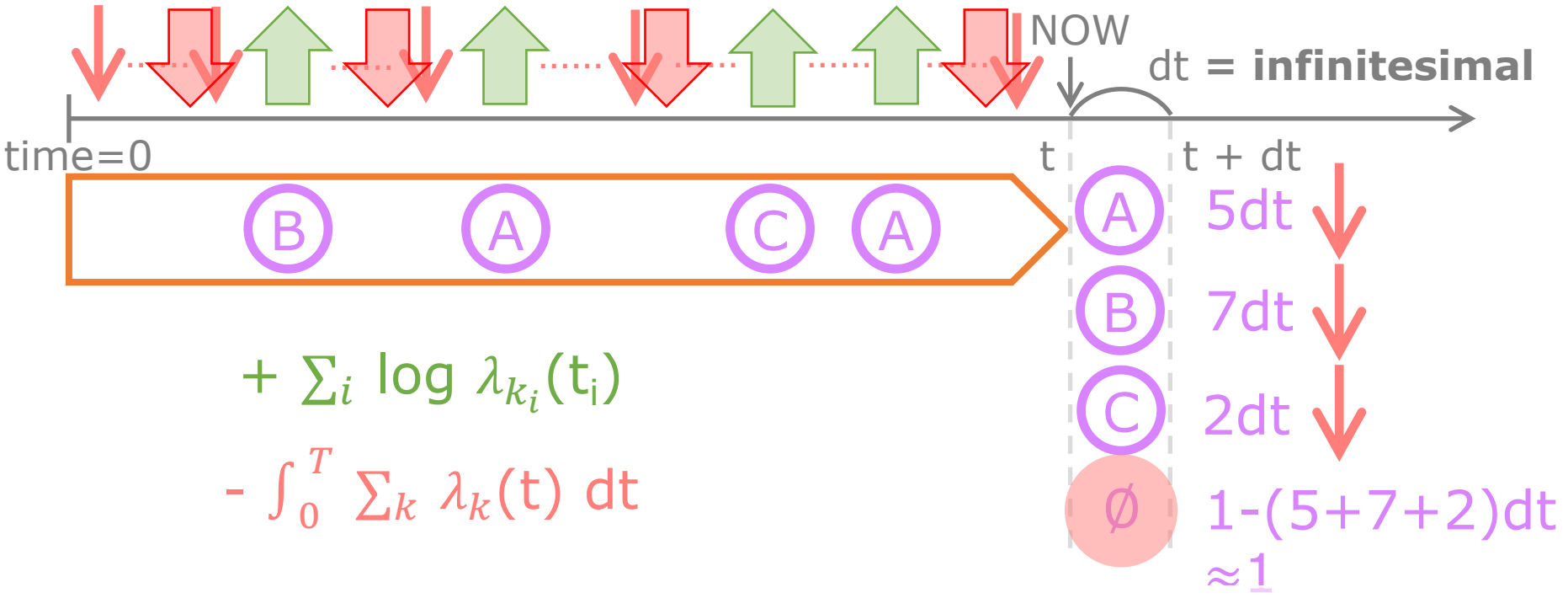
$+ \sum_i \log \lambda_{k_i}(t_i)$

$- \int_0^T \sum_k \lambda_k(t)\, dt$

Integrate over <u>infinitely</u> many *non*-events!

Loop over <u>all</u> event types!

MLE: Max log prob of *data*



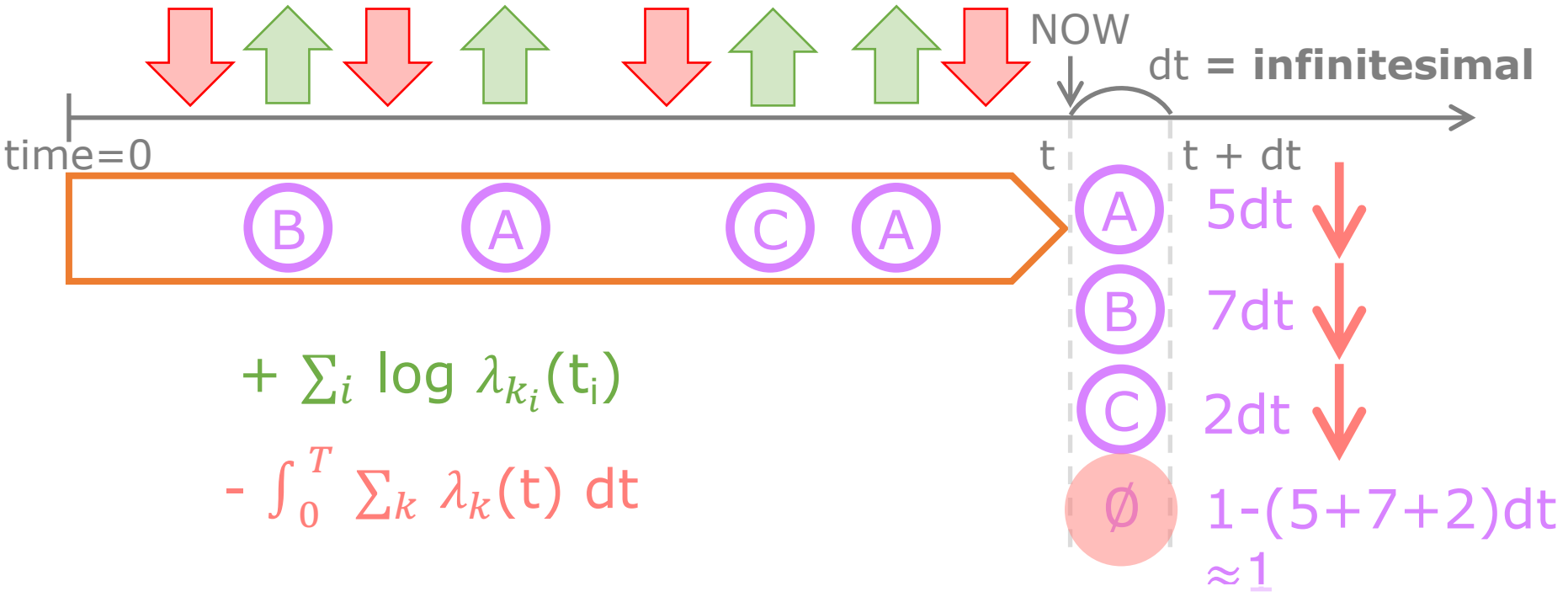
$+ \sum_i \log \lambda_{k_i}(t_i)$

$- \int_0^T \sum_k \lambda_k(t)\, dt$

Integrate over <u>infinitely</u> many *non*-events!

Loop over <u>all</u> event types!

MLE: Max log prob of *data*



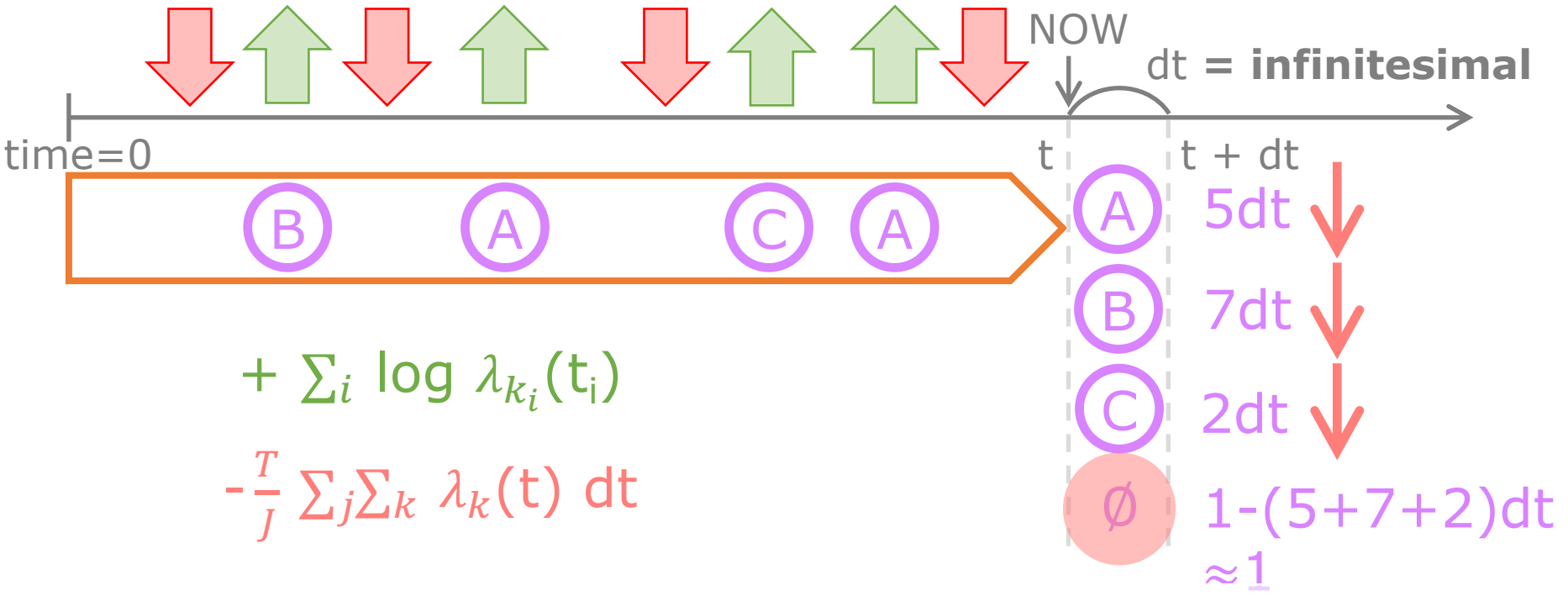
$+ \sum_i \log \lambda_{k_i}(t_i)$

$- \int_0^T \sum_k \lambda_k(t)\, dt$

Integrate over <u>infinitely</u> many *non*-events!

Loop over <u>all</u> event types!

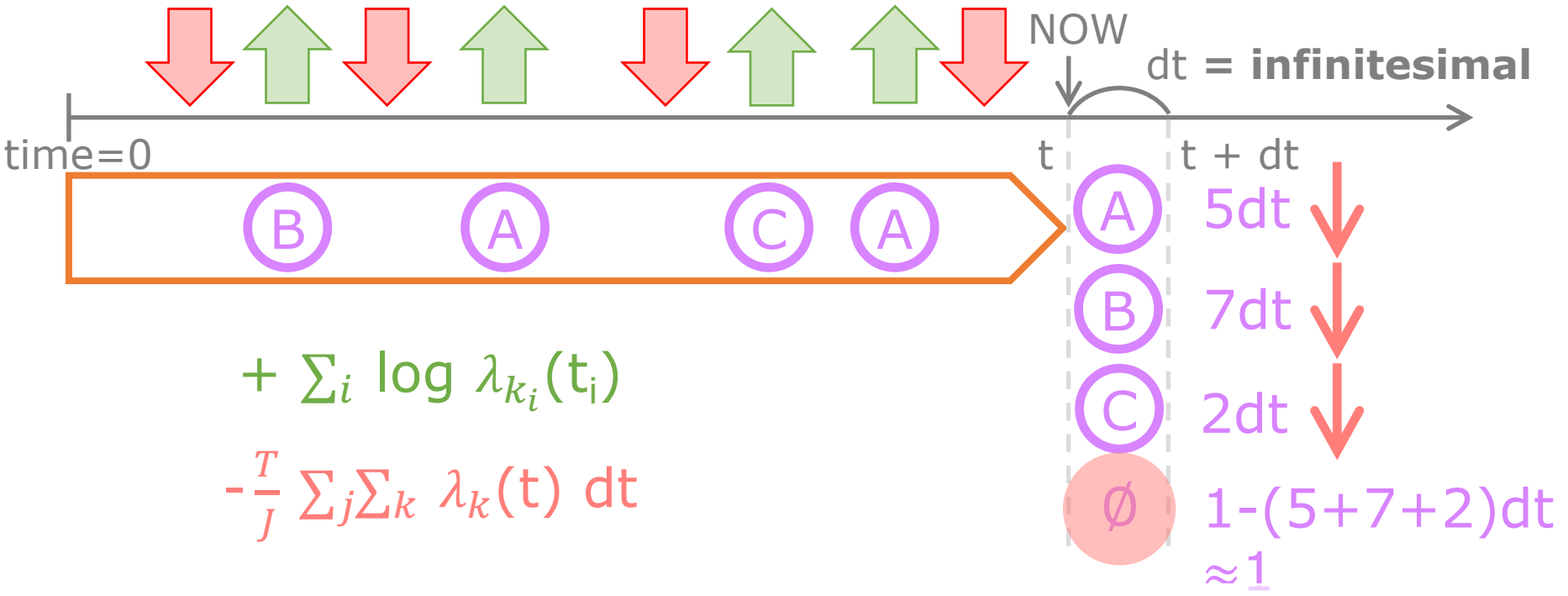
MLE: Max log prob of *data*



Integrate over <u>infinitely</u> many *non*-events!

Loop over <u>all</u> event types!

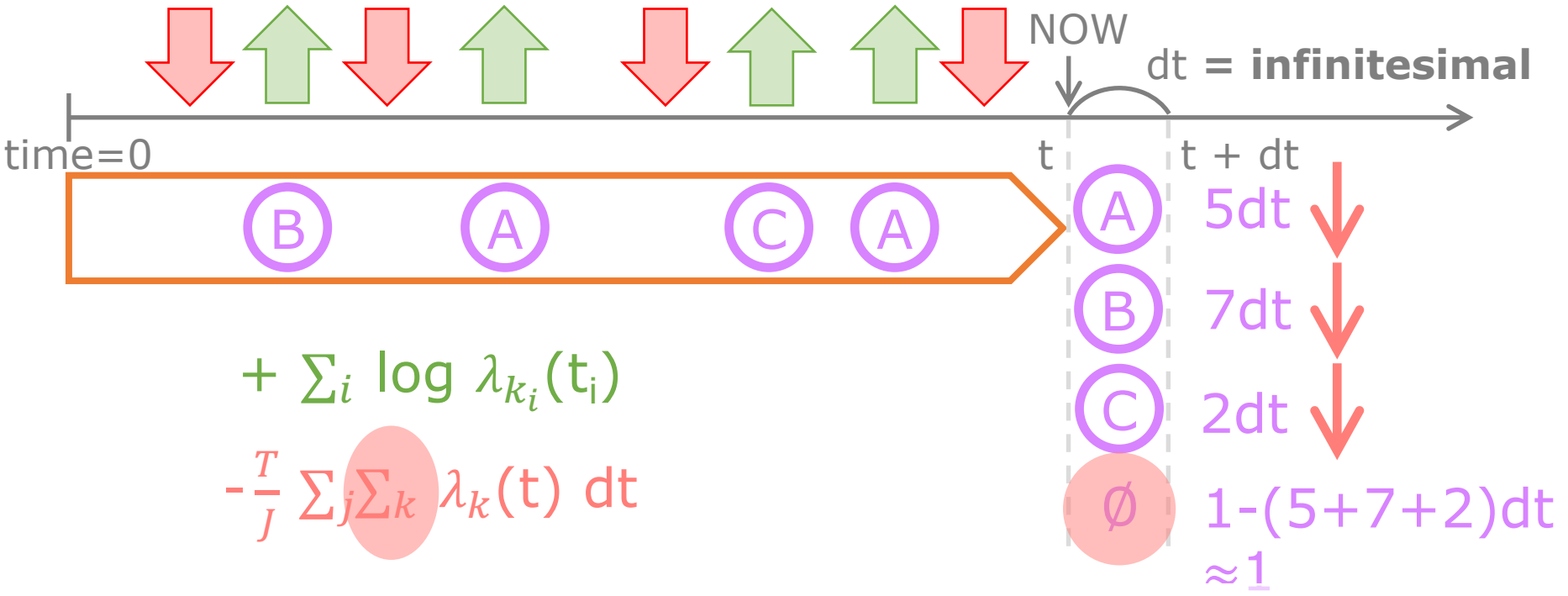
MLE: Max log prob of *data*



Approx by sampling

Loop over <u>all</u> event types!

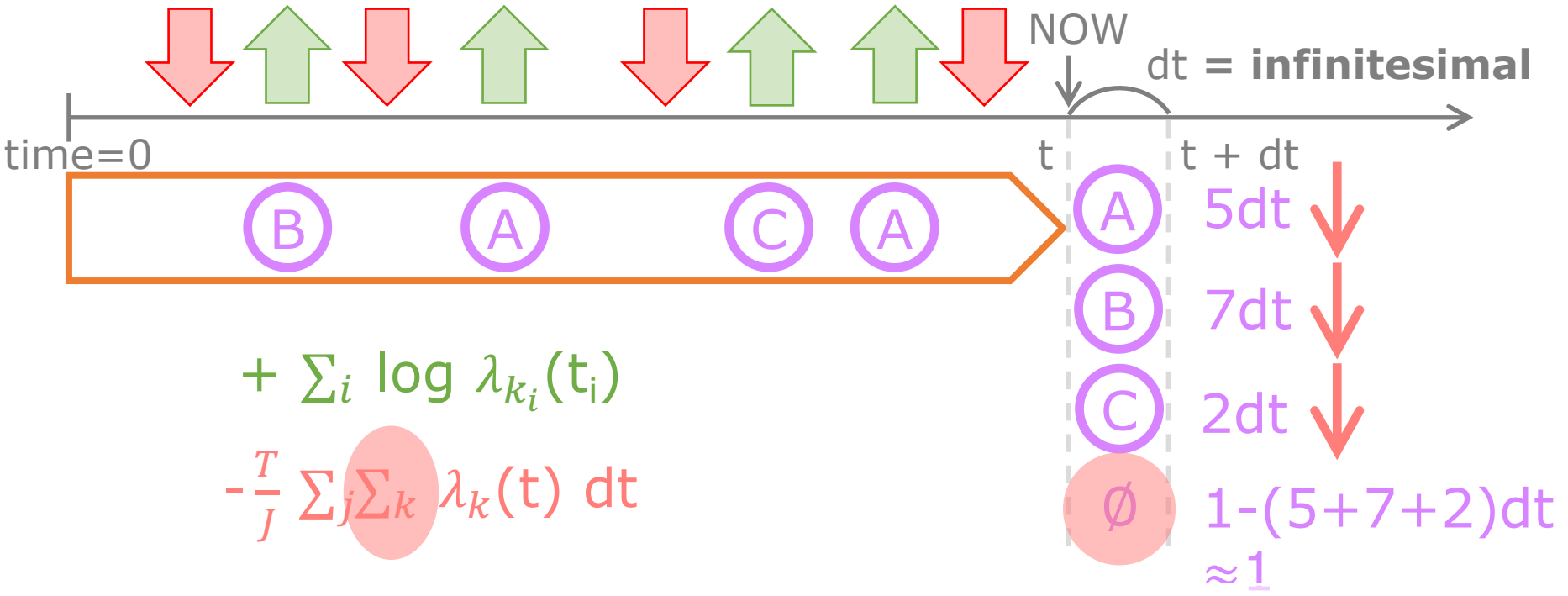
MLE: Max log prob of *data*



Approx by sampling

Loop over <u>all</u> event types!

MLE: Max log prob of *data*



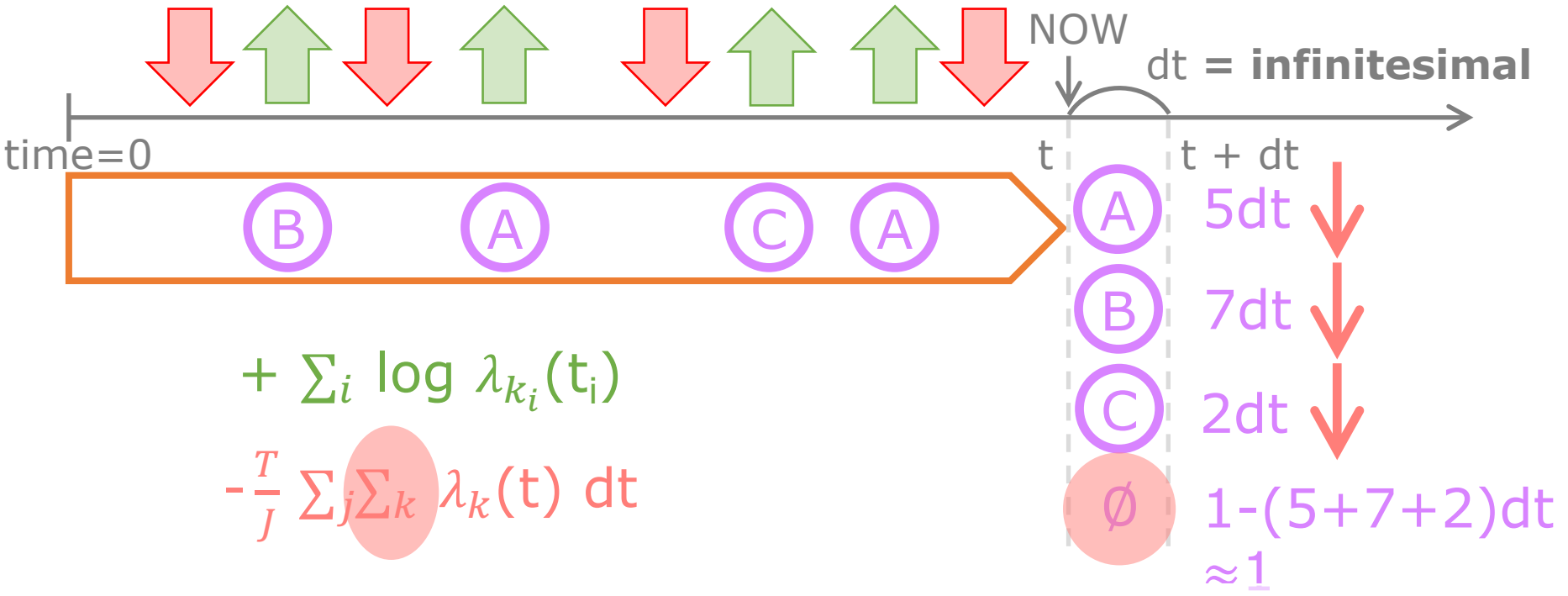
Approx by sampling

Still **Loop over <u>all</u> event types!**

Any Questions?

http://bburl/tpp-slides-p3

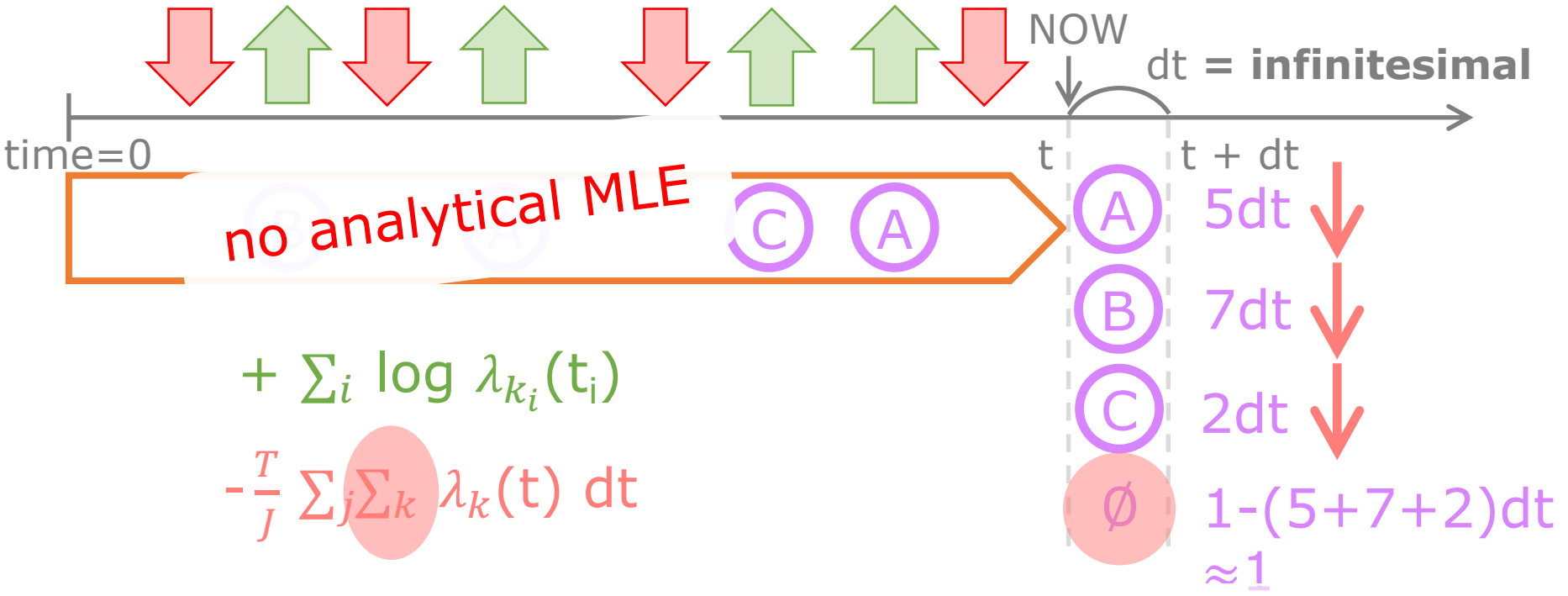
MLE: Max log prob of *data*



Approx by sampling

Still **Loop over <u>all</u> event types!**

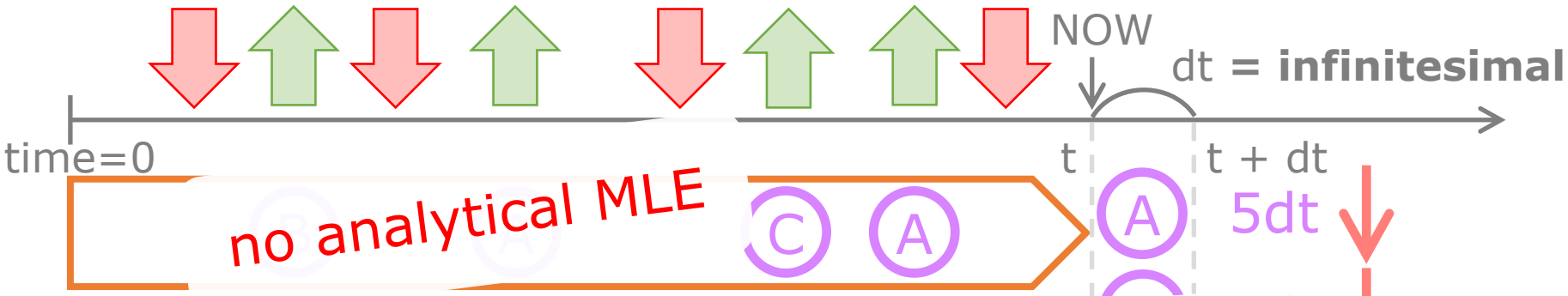
MLE: Max log prob of *data*



Approx by sampling

Still **Loop over <u>all</u> event types!**

MLE: Max log prob of *data*



Stochastic Gradient Descent

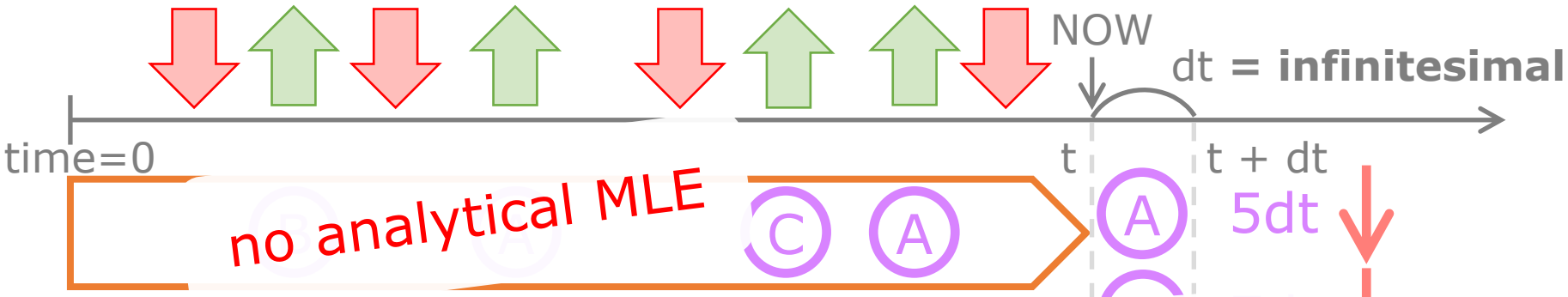
while not converge:

sequence ~ data corpus

S Still Loop over all event types!

$\theta \leftarrow \theta + \nabla_\theta$ log-likelihood

MLE: Max log prob of *data*



Stochastic Gradient Descent

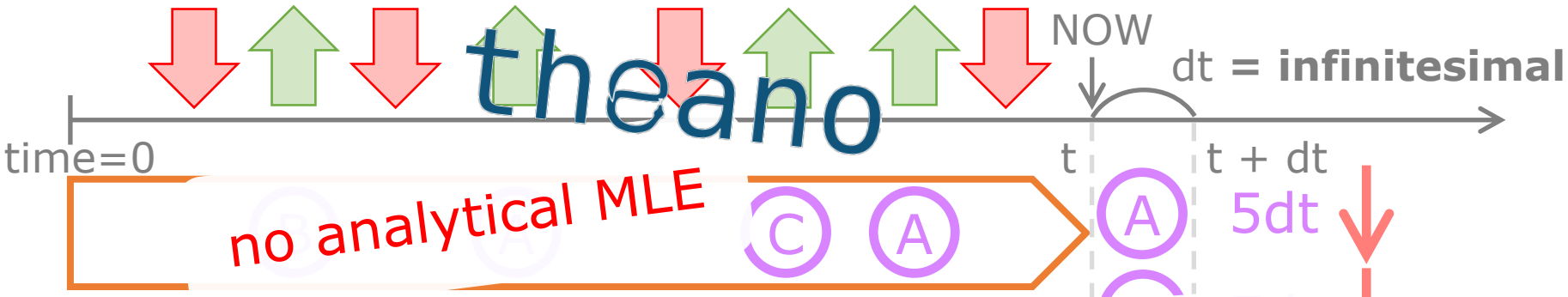
while not converge:

sequence ~ data corpus

S *Still* Loop over all event types!

$\theta \leftarrow \theta + \nabla_\theta$ log-likelihood ?

MLE: Max log prob of *data*



Stochastic Gradient Descent

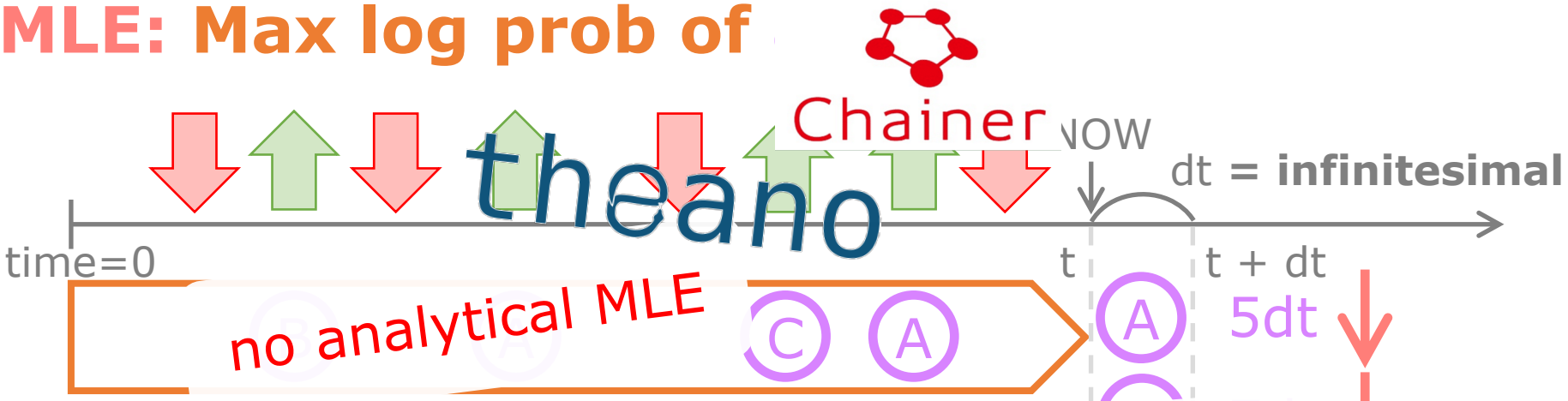
while not converge:

sequence ~ data corpus

S ~~Still Loop over all event types!~~

$\theta \leftarrow \theta + \nabla_\theta$ log-likelihood ?

MLE: Max log prob of



Stochastic Gradient Descent

while not converge:

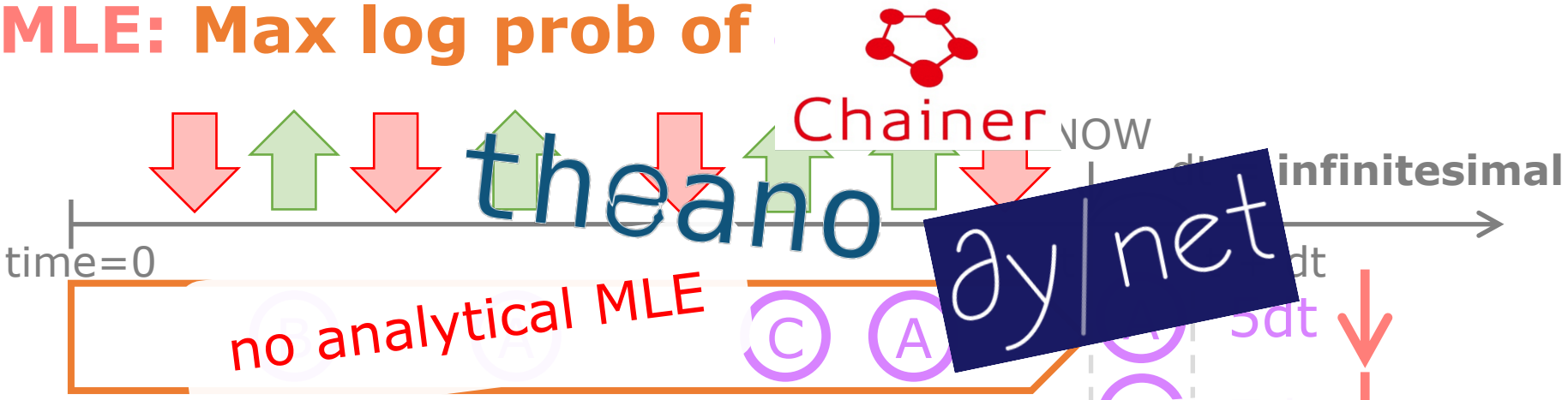
sequence ~ data corpus

S

$\theta \leftarrow \theta + \nabla_\theta$ log-likelihood

?

MLE: Max log prob of



Stochastic Gradient Descent

while not converge:

sequence ~ data corpus

S

$\theta \leftarrow \theta + \nabla_\theta$ log-likelihood

?

MLE: Max log prob of

Stochastic Gradient Descent

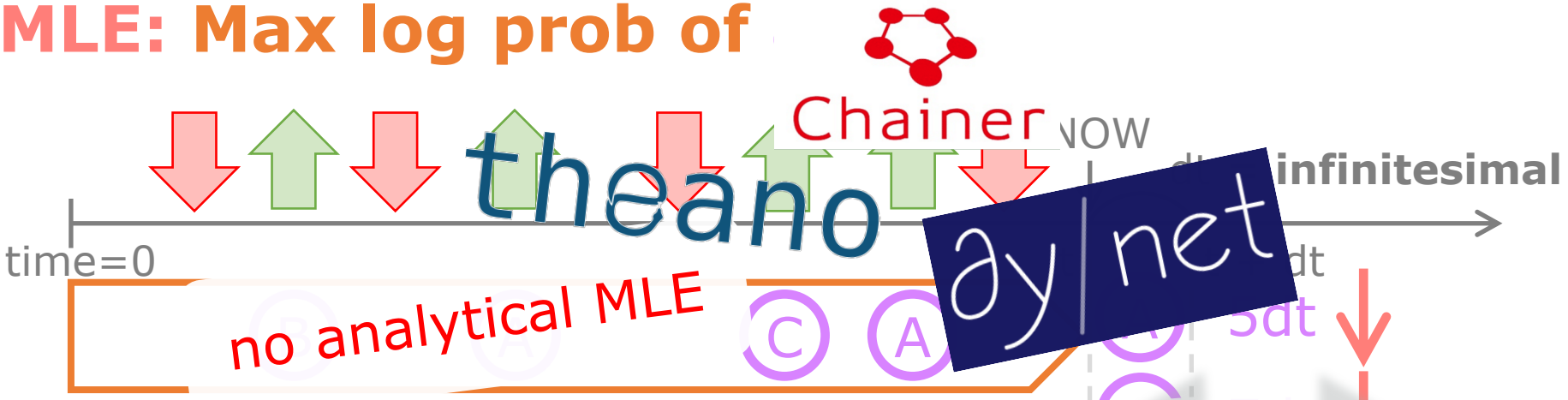
while not converge:

sequence ~ data corpus

S

$\theta \leftarrow \theta + \nabla_\theta$ log-likelihood ?

MLE: Max log prob of



Stochastic Gradient Descent

while not converge:

sequence ~ data corpus

S

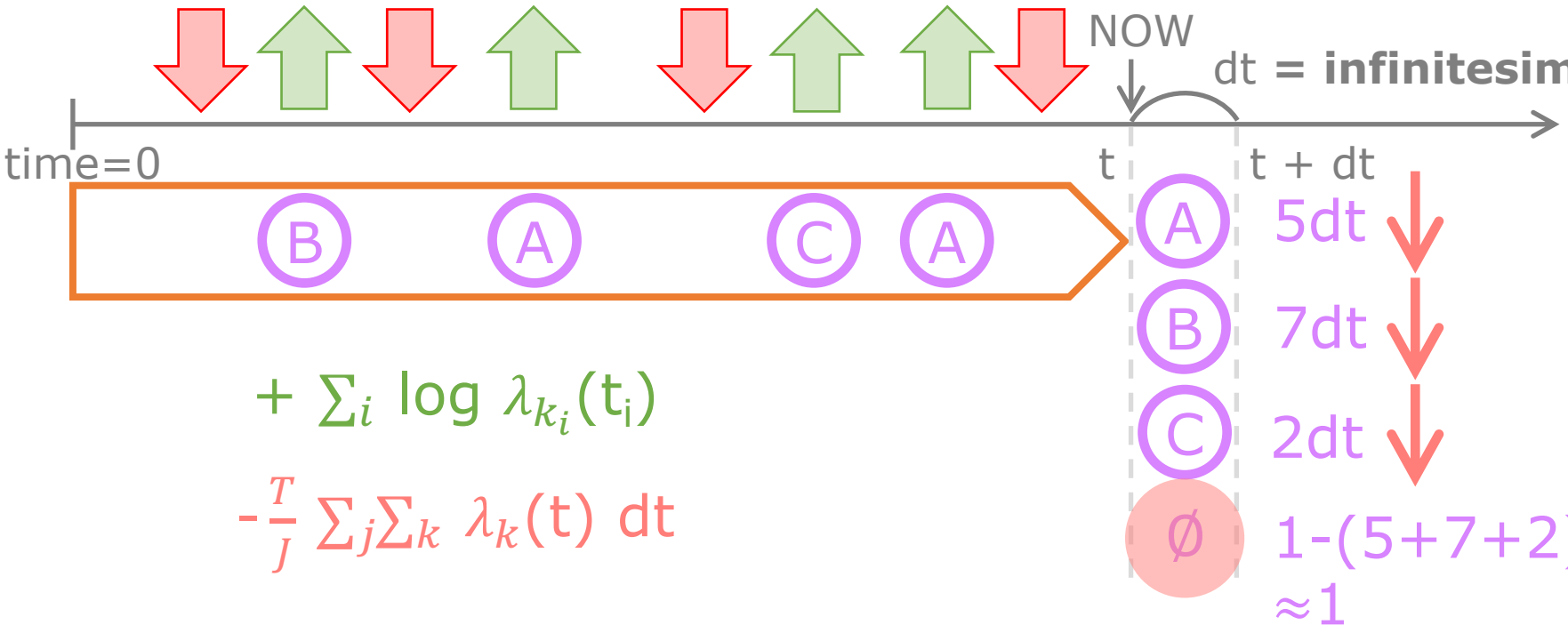
$\theta \leftarrow \theta + \nabla_\theta$ log likelihood

The Incredible PYTORCH

Let’s write some code

http://bburl/tpp-lab-p3

MLE: Max log prob of *data*

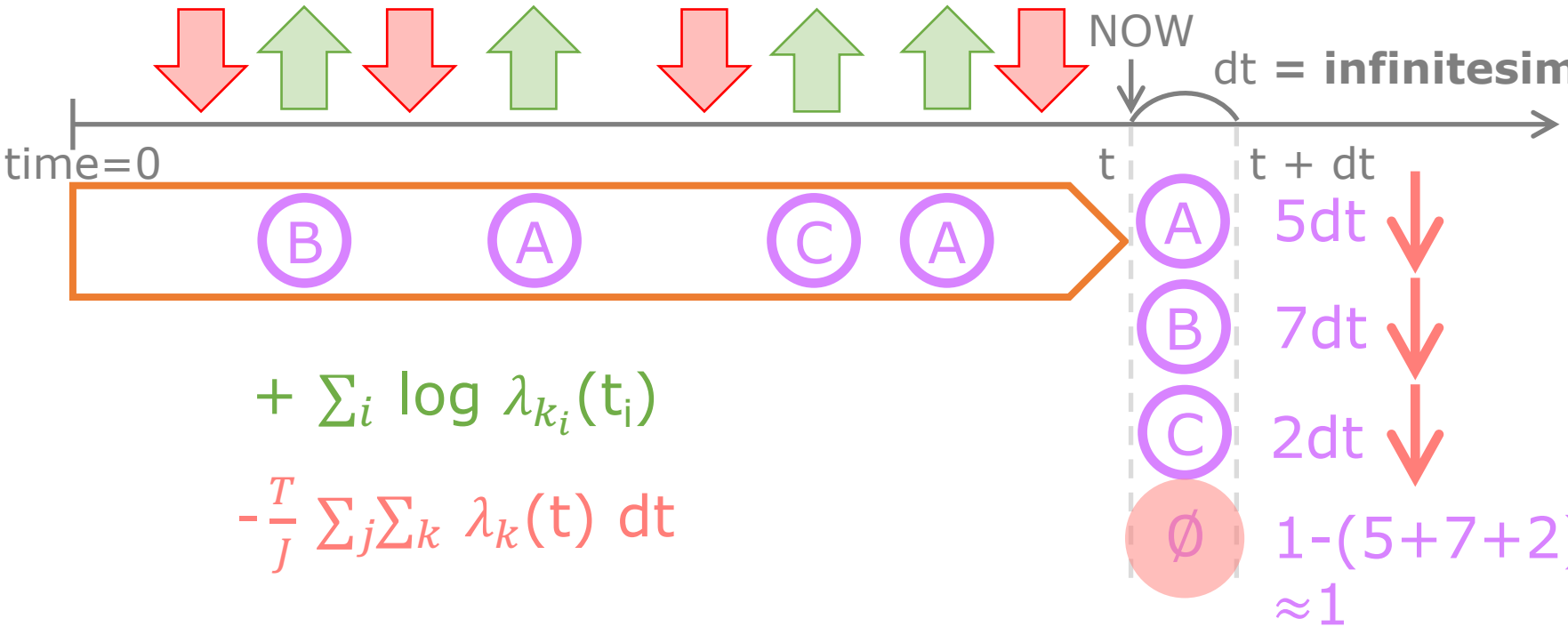


$+ \sum_i \log \lambda_{k_i}(t_i)$

$-\frac{T}{J} \sum_j \sum_k \lambda_k(t)\, dt$

Loop over <u>all</u> event types!

MLE: Max log prob of *data*



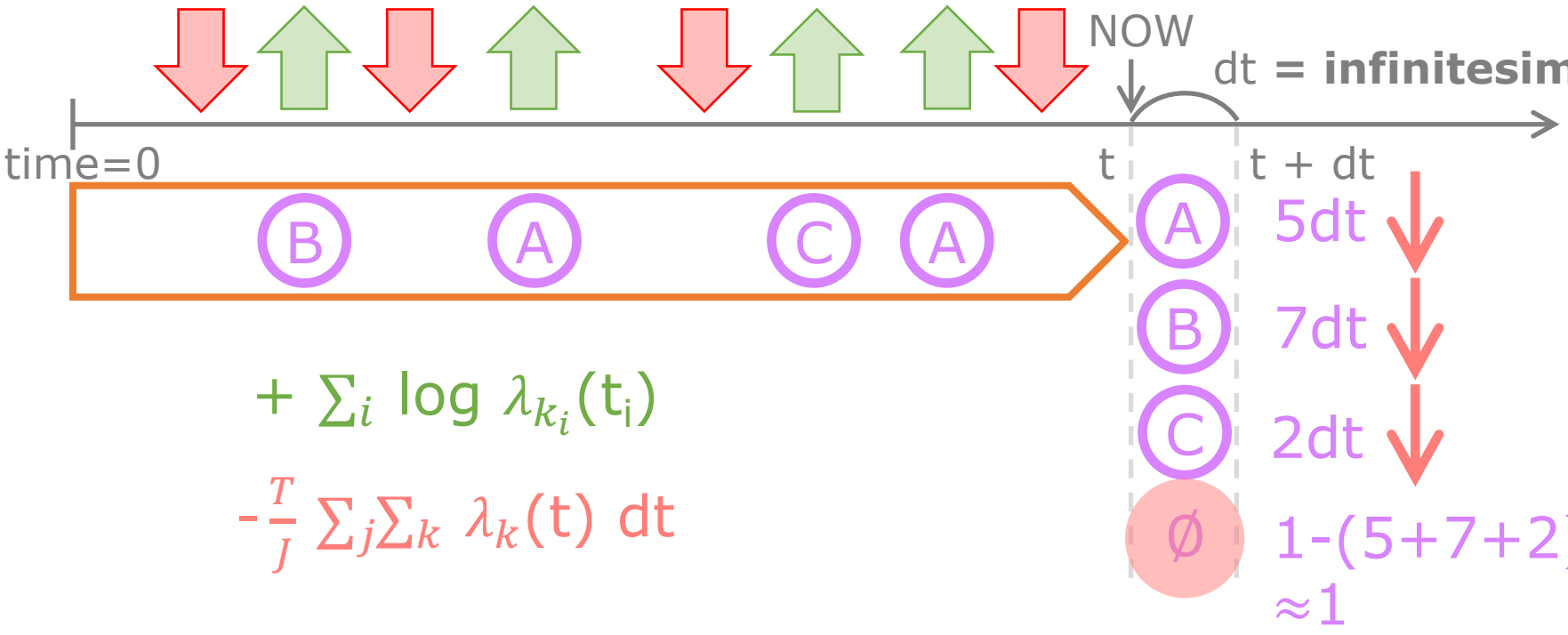
$+ \sum_i \log \lambda_{k_i}(t_i)$

$-\frac{T}{J} \sum_j \sum_k \lambda_k(t)\, dt$

= 50000?

Loop over <u>all</u> event types!

MLE: Max log prob of *data*



$+ \sum_i \log \lambda_{k_i}(t_i)$

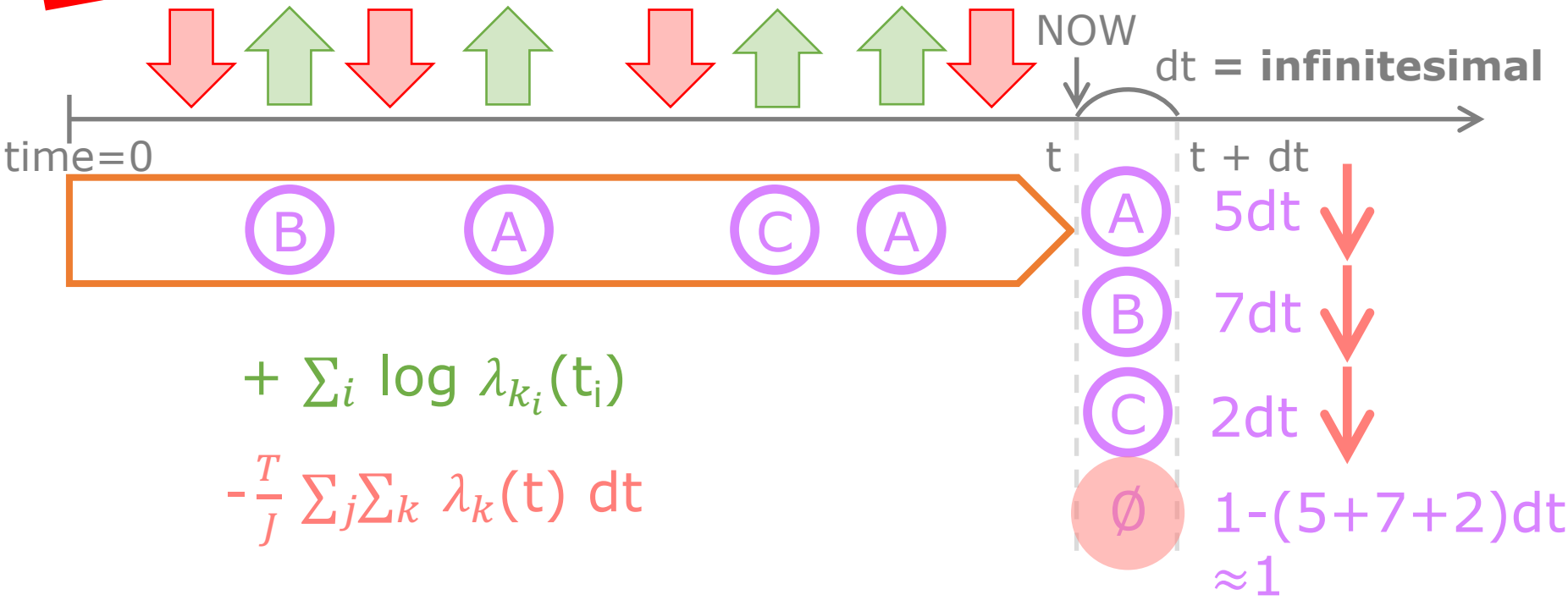
$-\frac{T}{J} \sum_j \sum_k \lambda_k(t)\, dt$

SLOW!

= 50000?

Loop over <u>all</u> event types!

~~MLE: Max log prob of *data*~~



$+ \sum_i \log \lambda_{k_i}(t_i)$

$-\frac{T}{J} \sum_j \sum_k \lambda_k(t)\, dt$

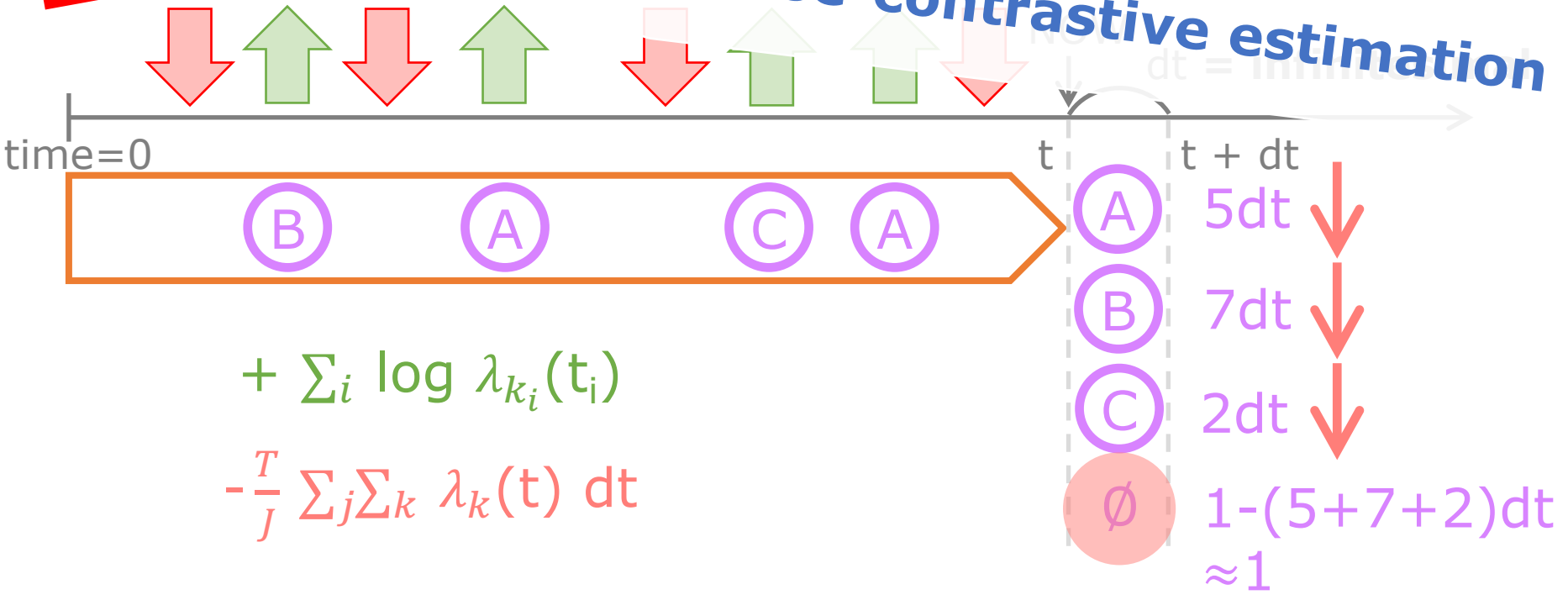
SLOW!

= 50000?

Loop over <u>all</u> event types!

~~MLE: Max log prob of data~~

alternative: noise-contrastive estimation



$+ \sum_i \log \lambda_{k_i}(t_i)$

$-\frac{T}{J} \sum_j \sum_k \lambda_k(t)\, dt$

SLOW!

= 50000?

Loop over <u>all</u> event types!

NCE: Max log prob of *correct discrimination*

NCE: Max log prob of *correct discrimination*

NCE: Max log prob of *correct discrimination*

log p(A)

NCE: Max log prob of *correct discrimination*

~~$\log p(\cdot)$~~

A

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

A

A

Is it Real or Noise?

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

A

A

Is it Real or Noise?

A

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

Is it Real or Noise?

p(A)

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

Is it Real or Noise?

p(A)

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

Is it Real or Noise?

p(A)

q(A)

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

A

A

Is it Real or Noise?

A p(A)

A q(A)

q = a cheaper model

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

A

A $\log \dfrac{p(A)}{p(A) + q(A)}$

Is it Real or Noise?

A $p(A)$

A $q(A)$

q = a cheaper model

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

A

A $\log \dfrac{p(A)}{p(A) + q(A)}$

Is it Real or Noise?

A $p(A)$ 

A $q(A)$

q = a cheaper model

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

A

A $\log \dfrac{p(A)}{p(A) + q(A)}$

B

Is it Real or Noise?

A $p(A)$ 

A $q(A)$

q = a cheaper model

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

A A $\log \frac{p(A)}{p(A) + q(A)}$ B B

Is it Real or Noise?

A $p(A)$ 

A $q(A)$

q = a cheaper model

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

A A $\log \frac{p(A)}{p(A) + q(A)}$ B B

Is it Real or Noise?

A p(A) B

A q(A)

q = a cheaper model

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

A A $\log \frac{p(A)}{p(A) + q(A)}$ B B

Is it Real or Noise?

A p(A)  B p(B)

A q(A)

q = a cheaper model

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

A A $\log \frac{p(A)}{p(A) + q(A)}$ B B

Is it Real or Noise?

A p(A)  B p(B)

A q(A) B

q = a cheaper model

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

A A $\log \frac{p(A)}{p(A) + q(A)}$ B B

Is it Real or Noise?

A p(A)  B p(B)

A q(A) B q(B)

q = a cheaper model

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

A A $\log \frac{p(A)}{p(A) + q(A)}$

B B $\log \frac{q(B)}{p(B) + q(B)}$

Is it Real or Noise?

A p(A) 

B p(B)

A q(A)

B q(B)

q = a cheaper model

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

A A $\log \frac{p(A)}{p(A) + q(A)}$

B B $\log \frac{q(B)}{p(B) + q(B)}$

Is it Real or Noise?

A p(A) ↑

A q(A)

B p(B) ↓

B q(B)

q = a cheaper model

NCE: Max log prob of *correct discrimination*

~~log p(A)~~

Binary-Classification Version
Gutmann & Hyvarinen 2010

A A $\log \frac{p(A)}{p(A) + q(A)}$

B B $\log \frac{q(B)}{p(B) + q(B)}$

Is it Real or Noise?

A p(A) ↑

B p(B) ↓

A q(A)

B q(B)

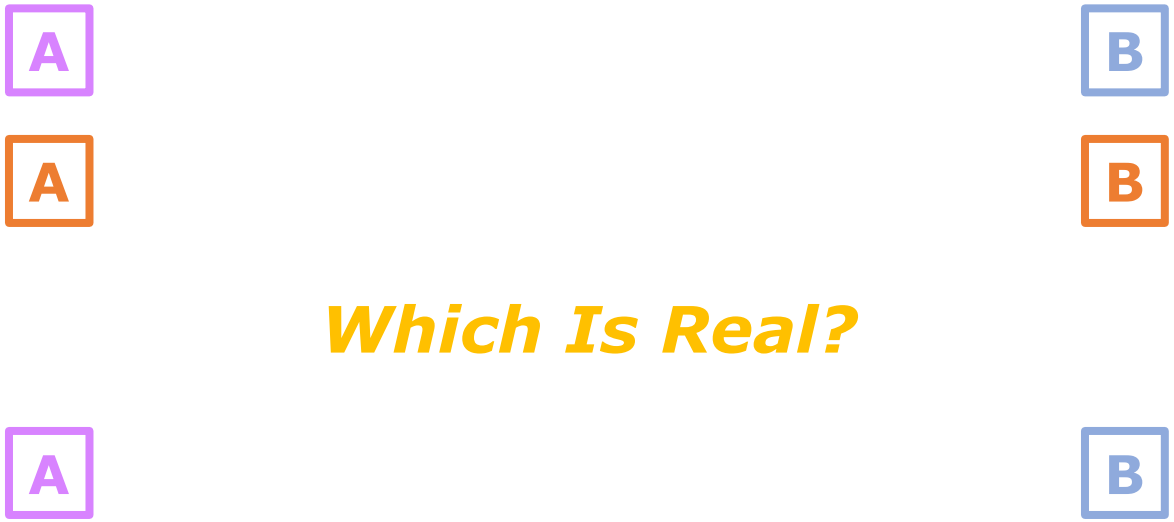
q = a cheaper model

NCE: Max log prob of *correct discrimination*

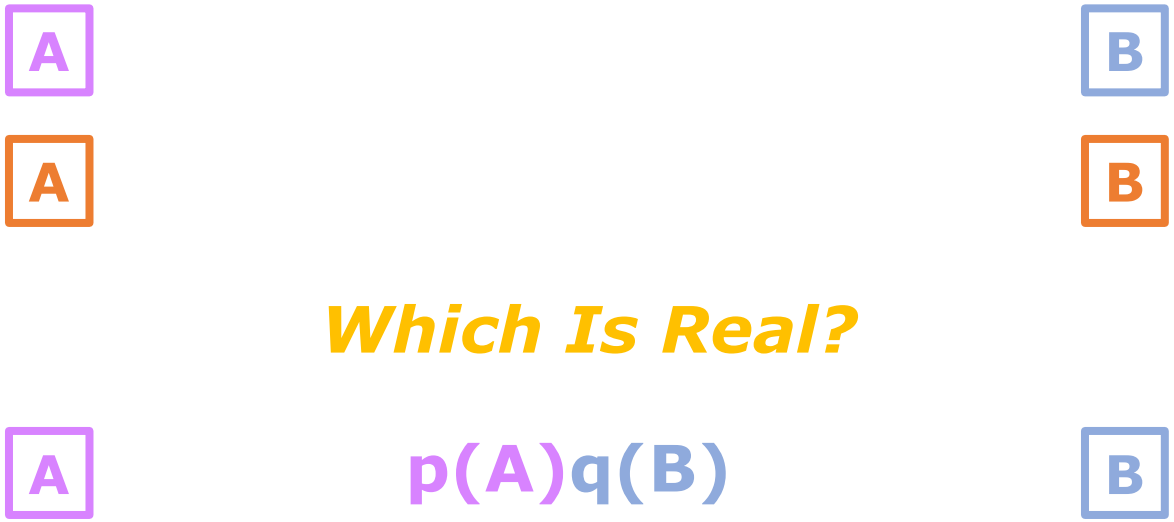
NCE: Max log prob of *correct discrimination*

Which Is Real?

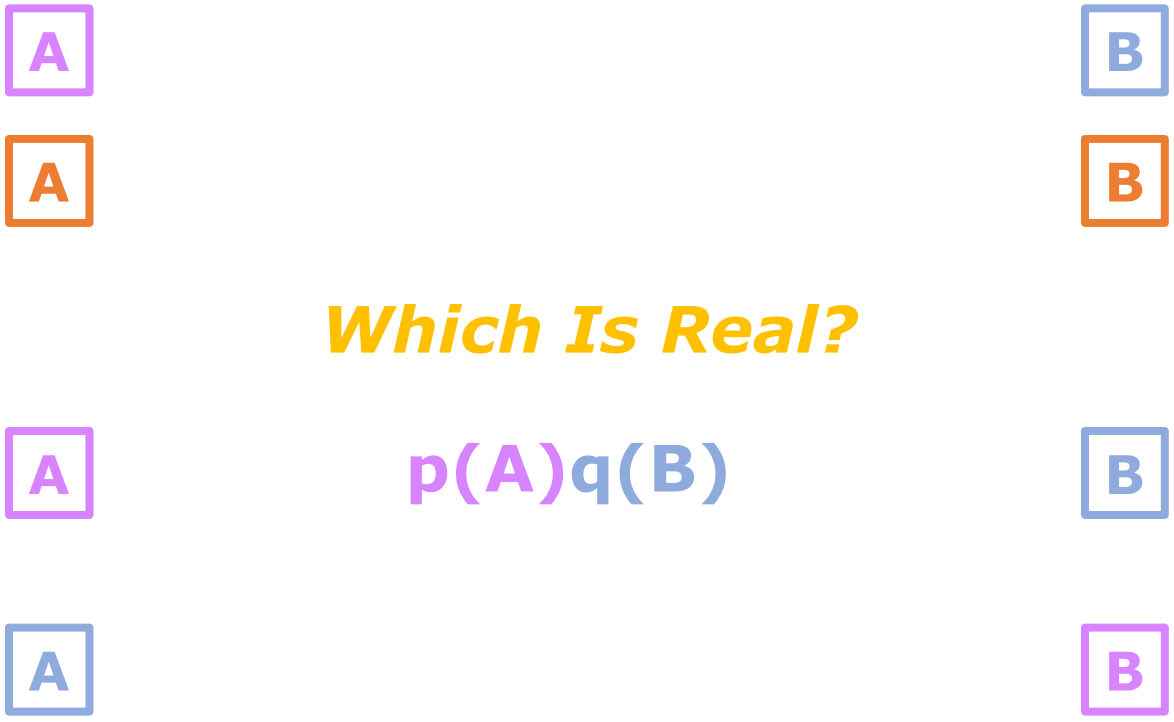
NCE: Max log prob of *correct discrimination*



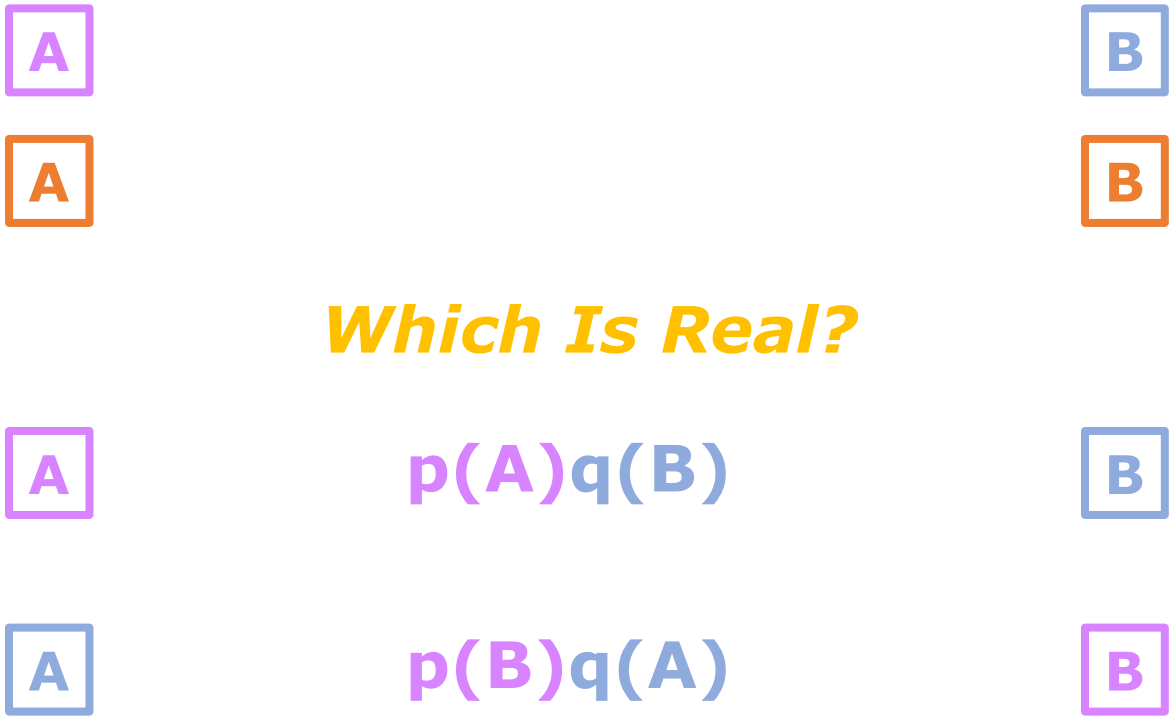
NCE: Max log prob of *correct discrimination*



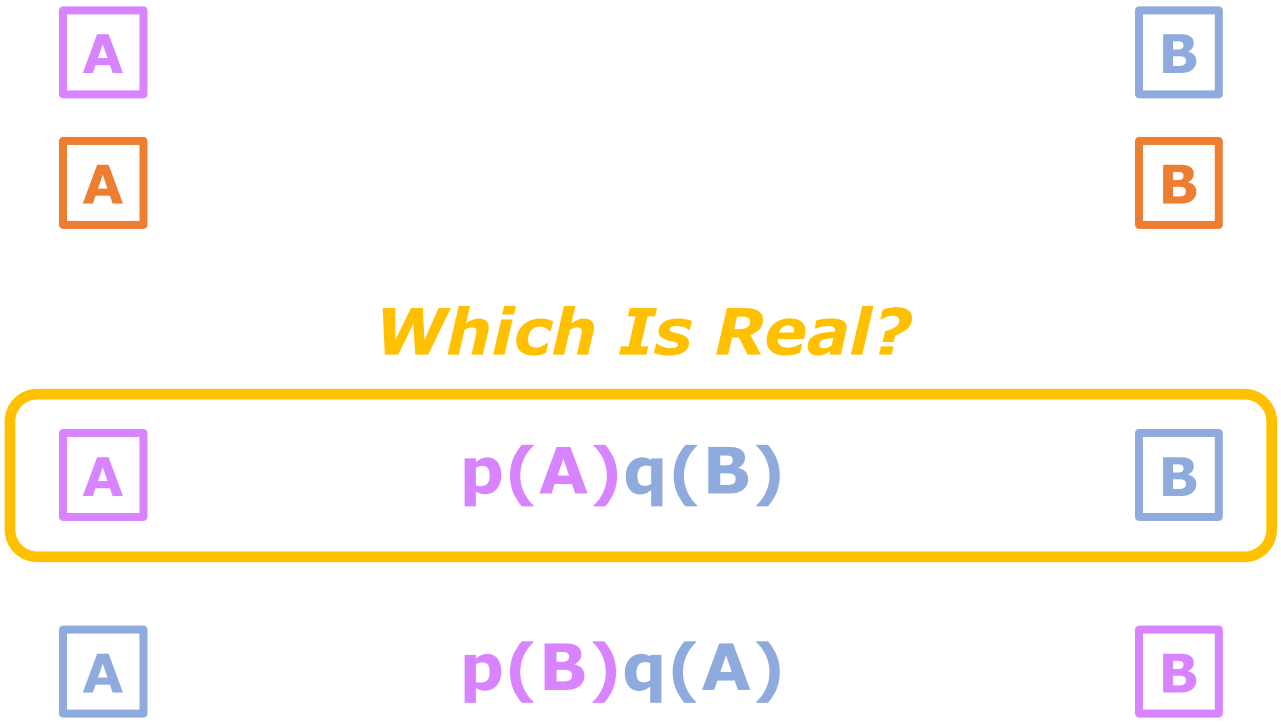
NCE: Max log prob of *correct discrimination*



NCE: Max log prob of *correct discrimination*



NCE: Max log prob of *correct discrimination*



NCE: Max log prob of *correct discrimination*

$$\log \frac{p(A)q(B)}{p(A)q(B) + p(B)q(A)}$$

A A B B

Which Is Real?

A p(A)q(B) B

A p(B)q(A) B

NCE: Max log prob of *correct discrimination*

A A $\log \frac{p(A)q(B)}{p(A)q(B) + p(B)q(A)}$ B B

p(A) ↑

Which Is Real?

A p(A)q(B) B

A p(B)q(A) B

NCE: Max log prob of *correct discrimination*

$$\log \frac{p(A)q(B)}{p(A)q(B) + p(B)q(A)}$$

A A B B

Which Is Real?

A p(A)q(B) B

A p(B)q(A) B

p(A) ↑

p(B) ↓

NCE: Max log prob of *correct discrimination*

Ranking Version
Ma & Collins 2018

A A $\log \frac{p(A)q(B)}{p(A)q(B) + p(B)q(A)}$ B B

p(A) ↑

p(B) ↓

Which Is Real?

A p(A)q(B) B

A p(B)q(A) B

NCE: Max log prob of *correct discrimination*

Consistency under *weaker* assumptions

Ranking Version Ma & Collins 2018

A A $\log \frac{p(A)q(B)}{p(A)q(B) + p(B)q(A)}$ B B

p(A) ↑

p(B) ↓

Which Is Real?

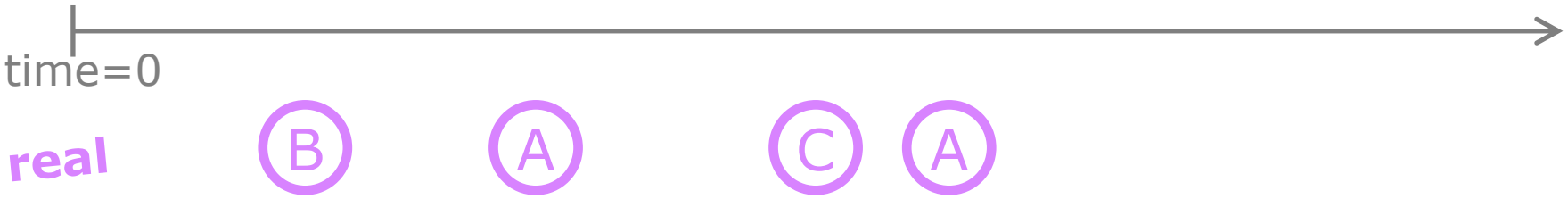
A p(A)q(B) B

A p(B)q(A) B

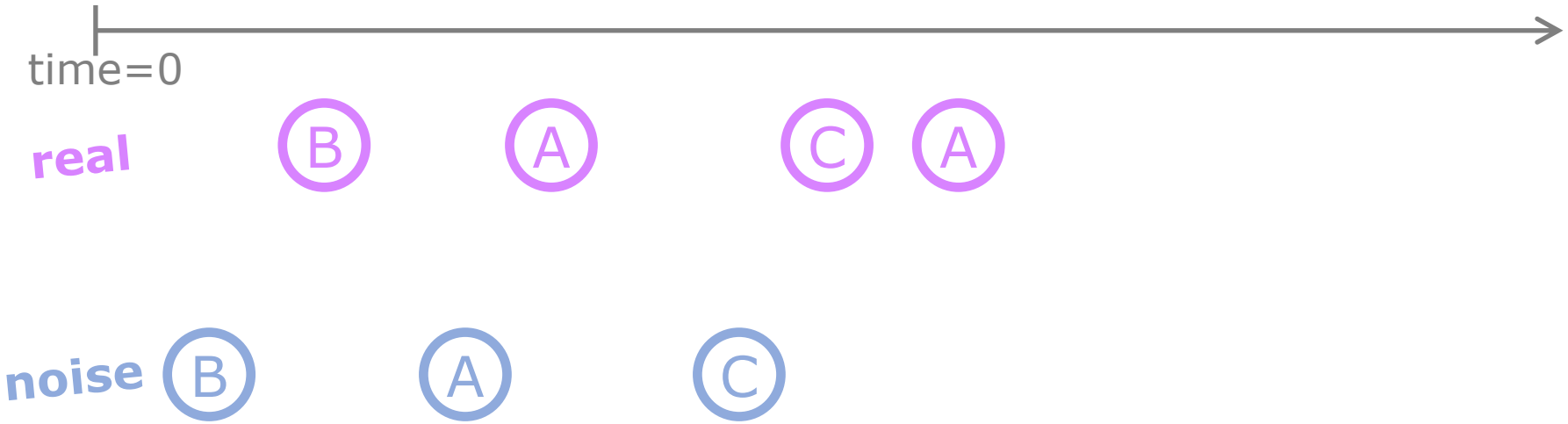
Any Questions?

http://bburl/tpp-slides-p3

NCE: Max log prob of *correct discrimination*

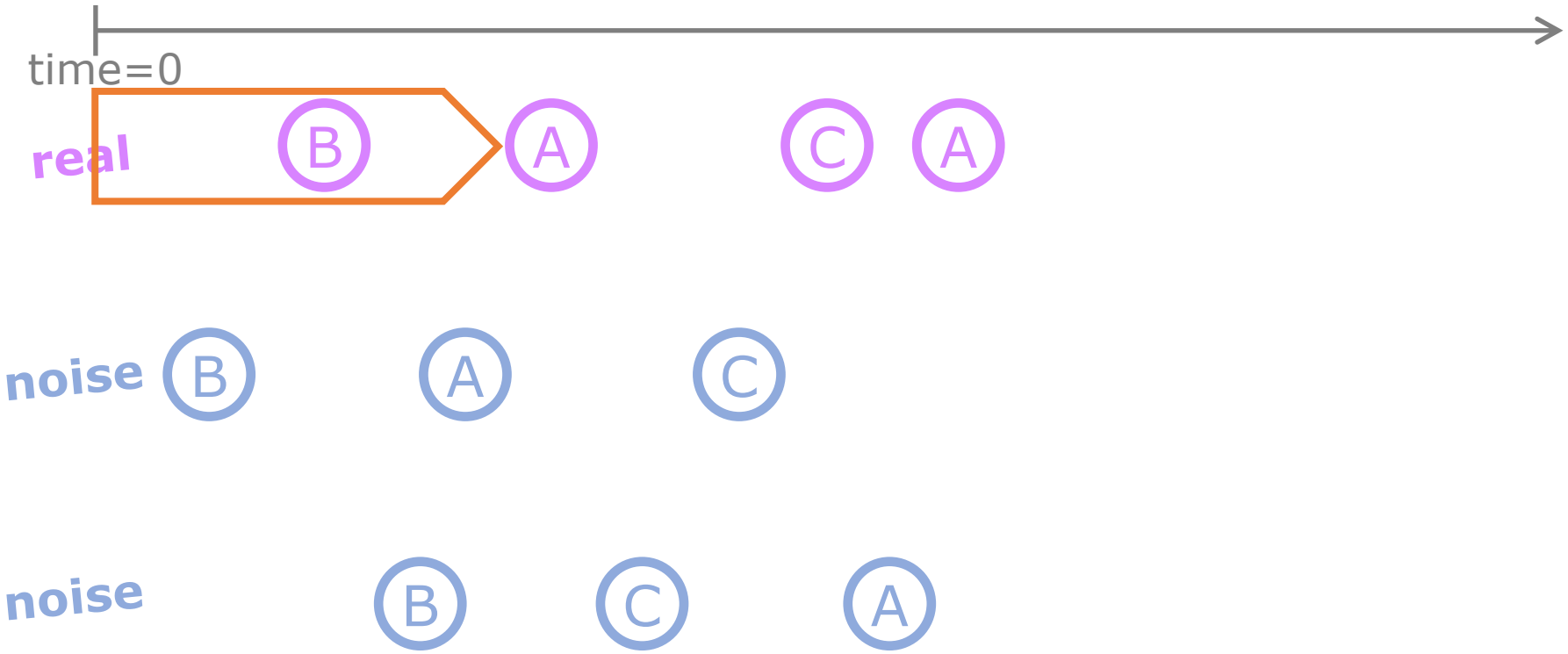


NCE: Max log prob of *correct discrimination*

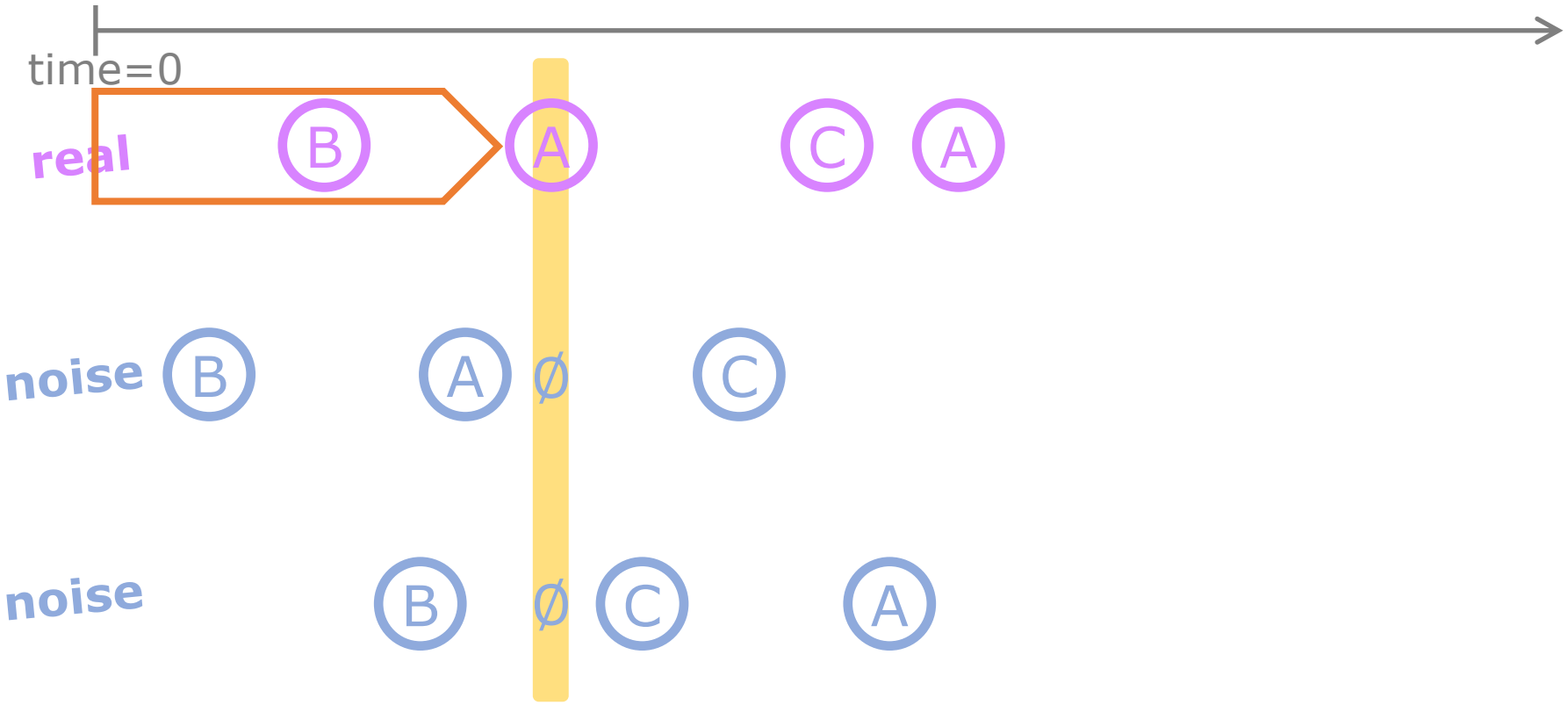


NCE: Max log prob of *correct discrimination*

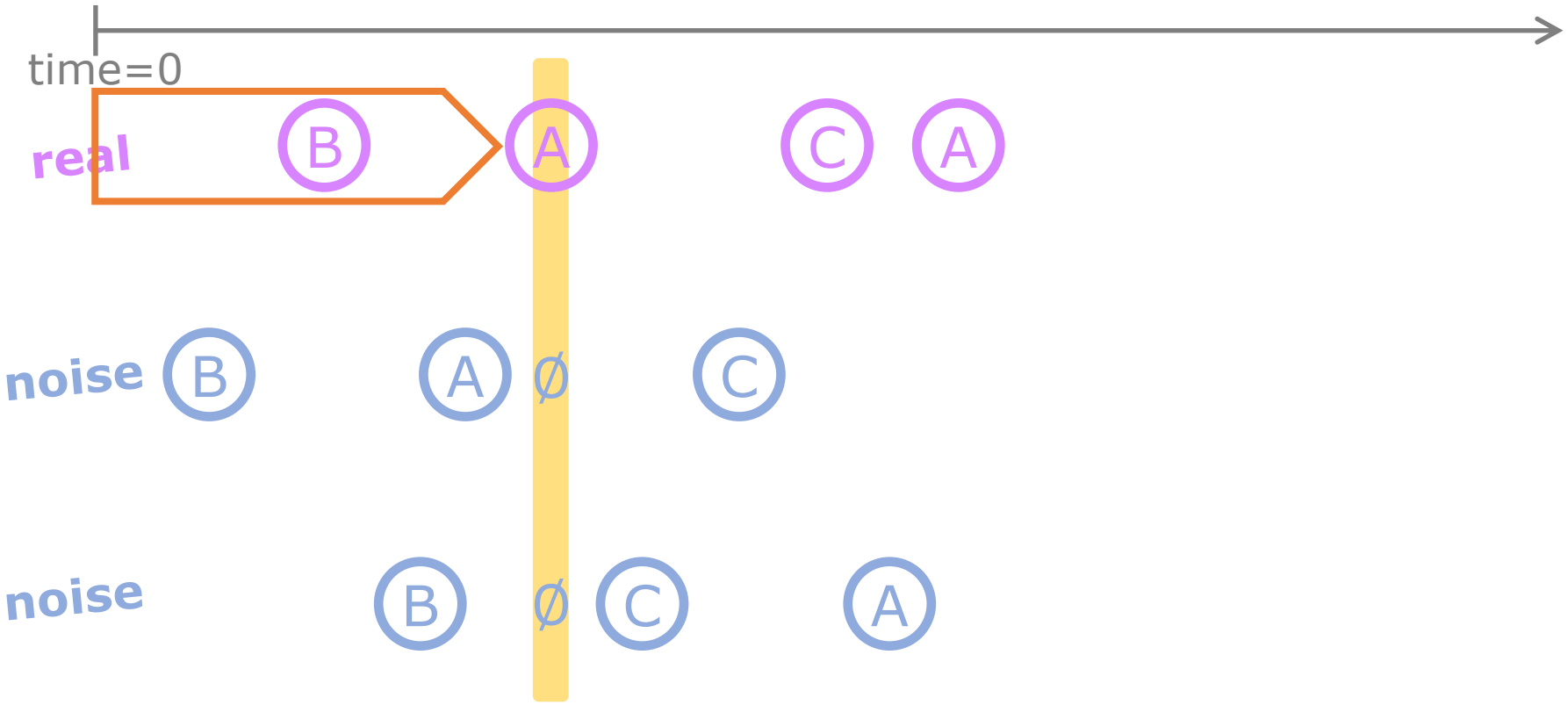
NCE: Max log prob of *correct discrimination*



NCE: Max log prob of *correct discrimination*

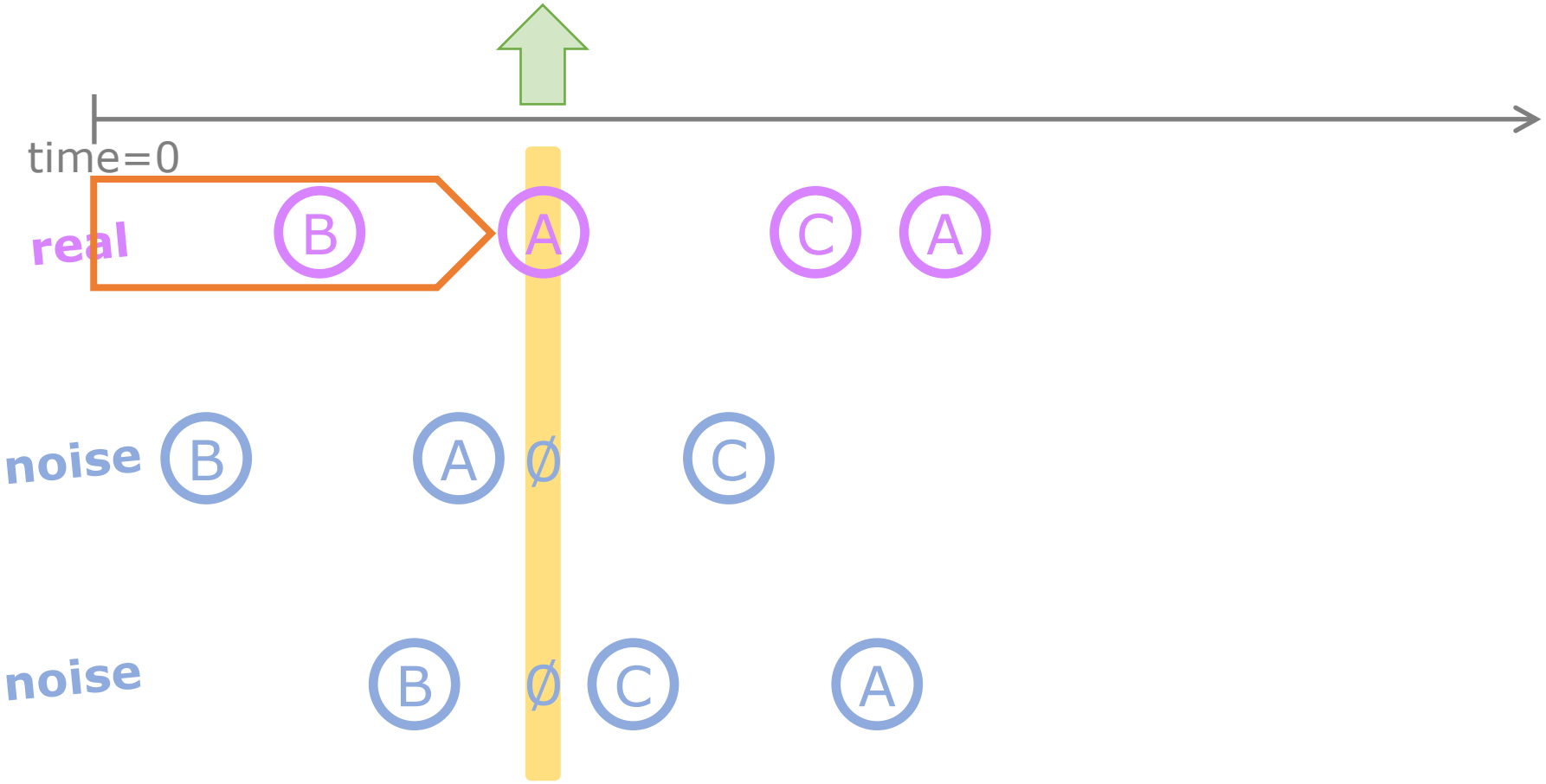


NCE: Max log prob of *correct discrimination*



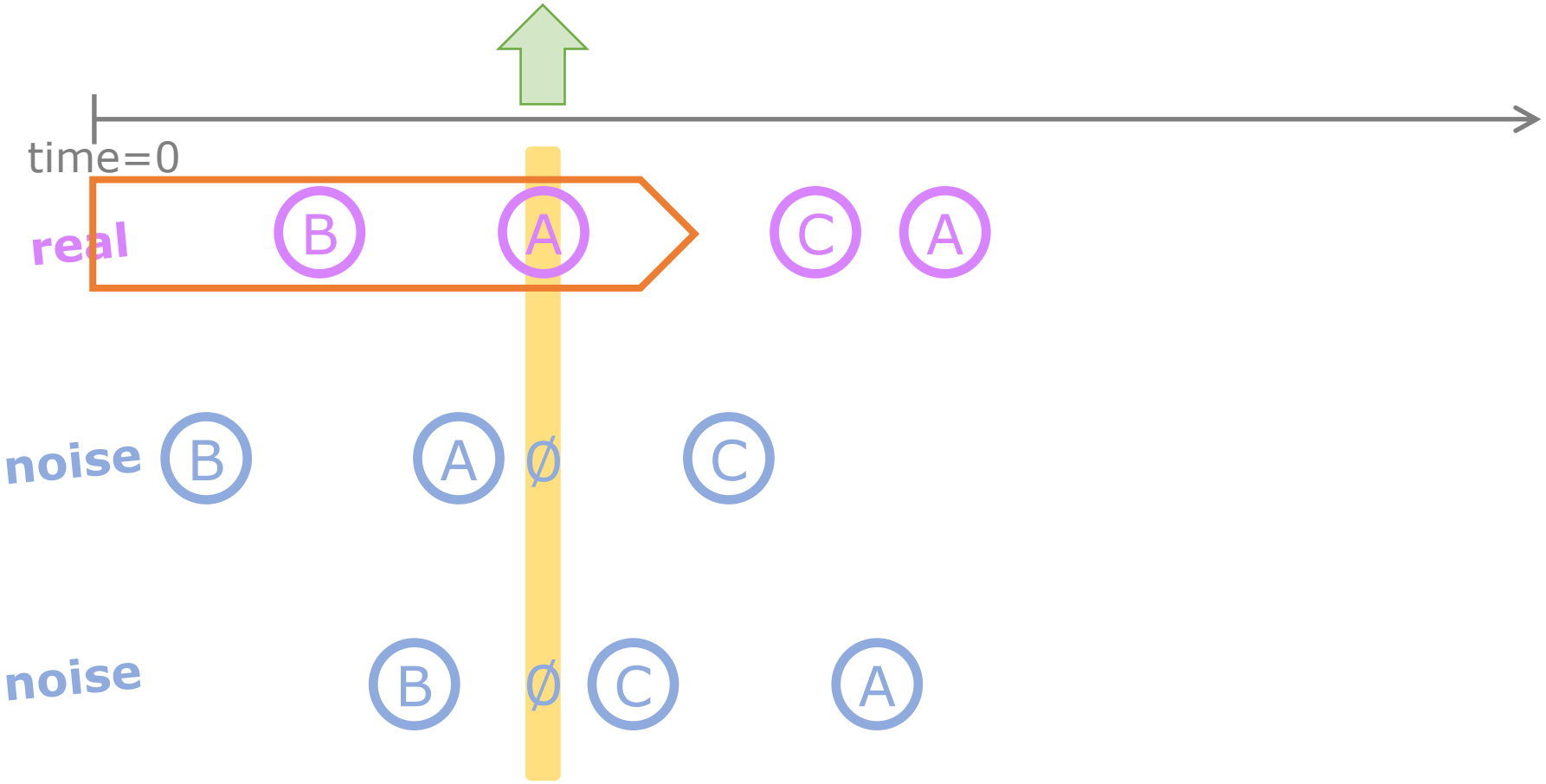
Which Is Real?

NCE: Max log prob of *correct discrimination*



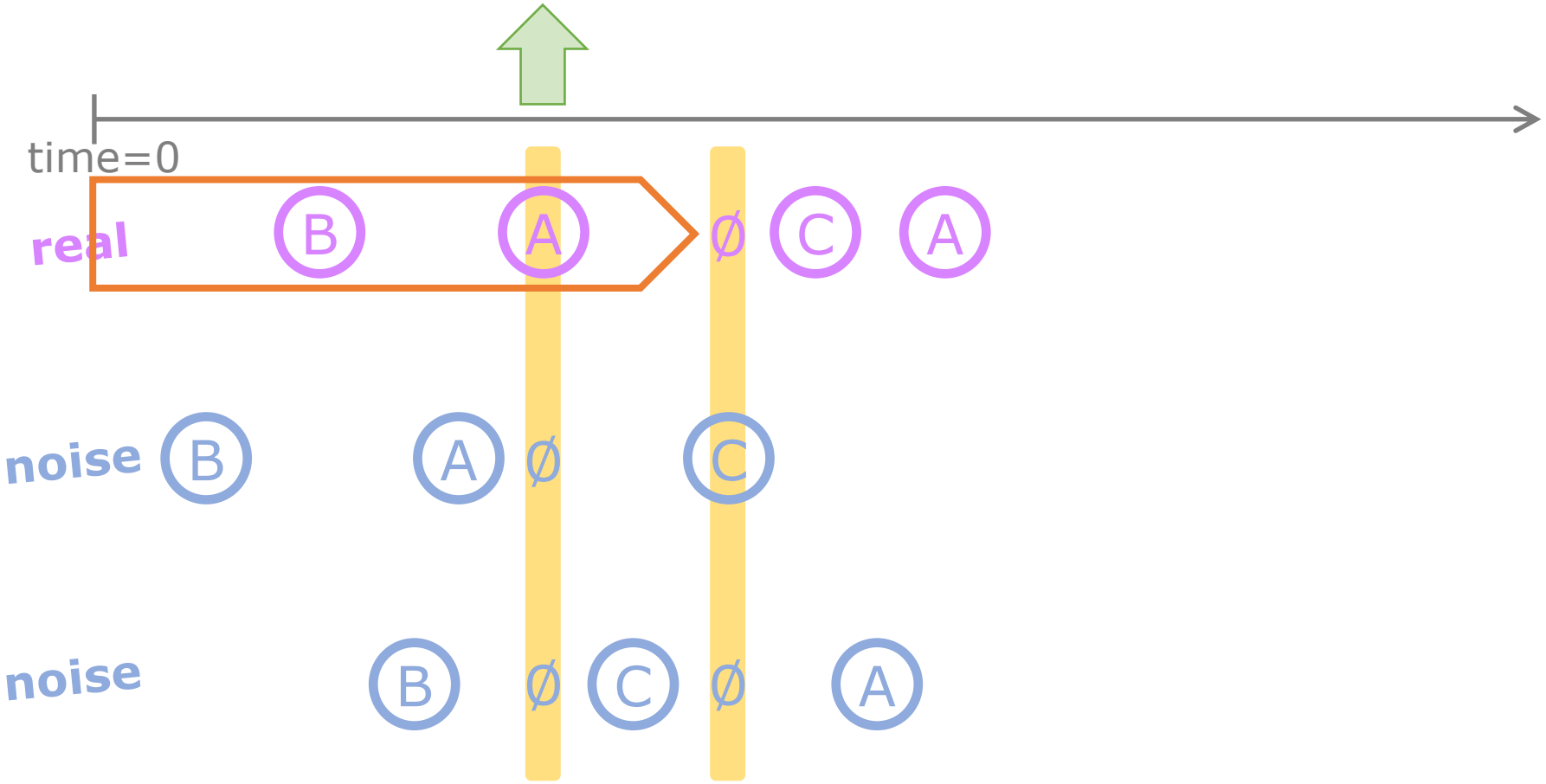
Which Is Real?

NCE: Max log prob of *correct discrimination*



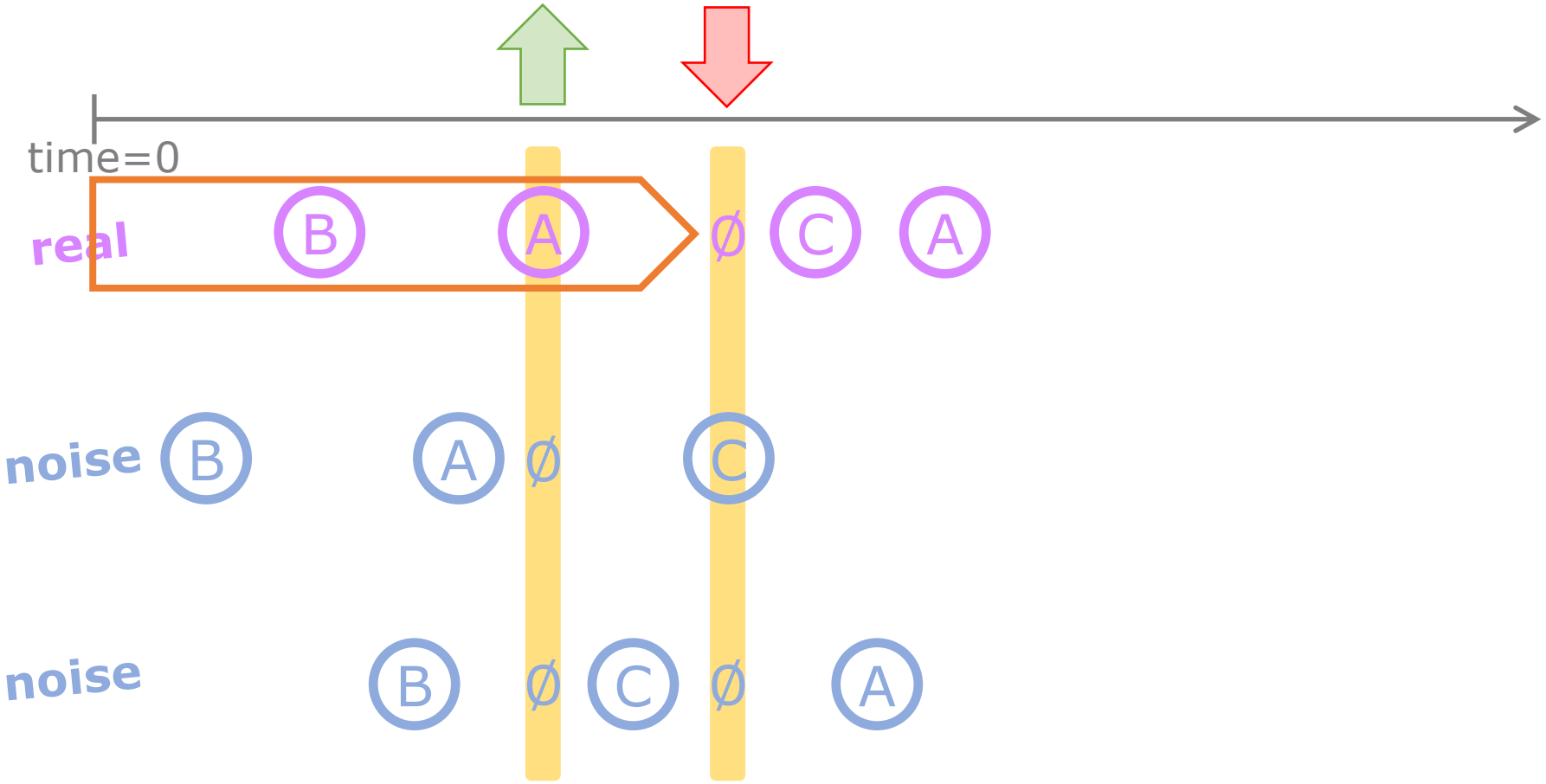
Which Is Real?

NCE: Max log prob of *correct discrimination*



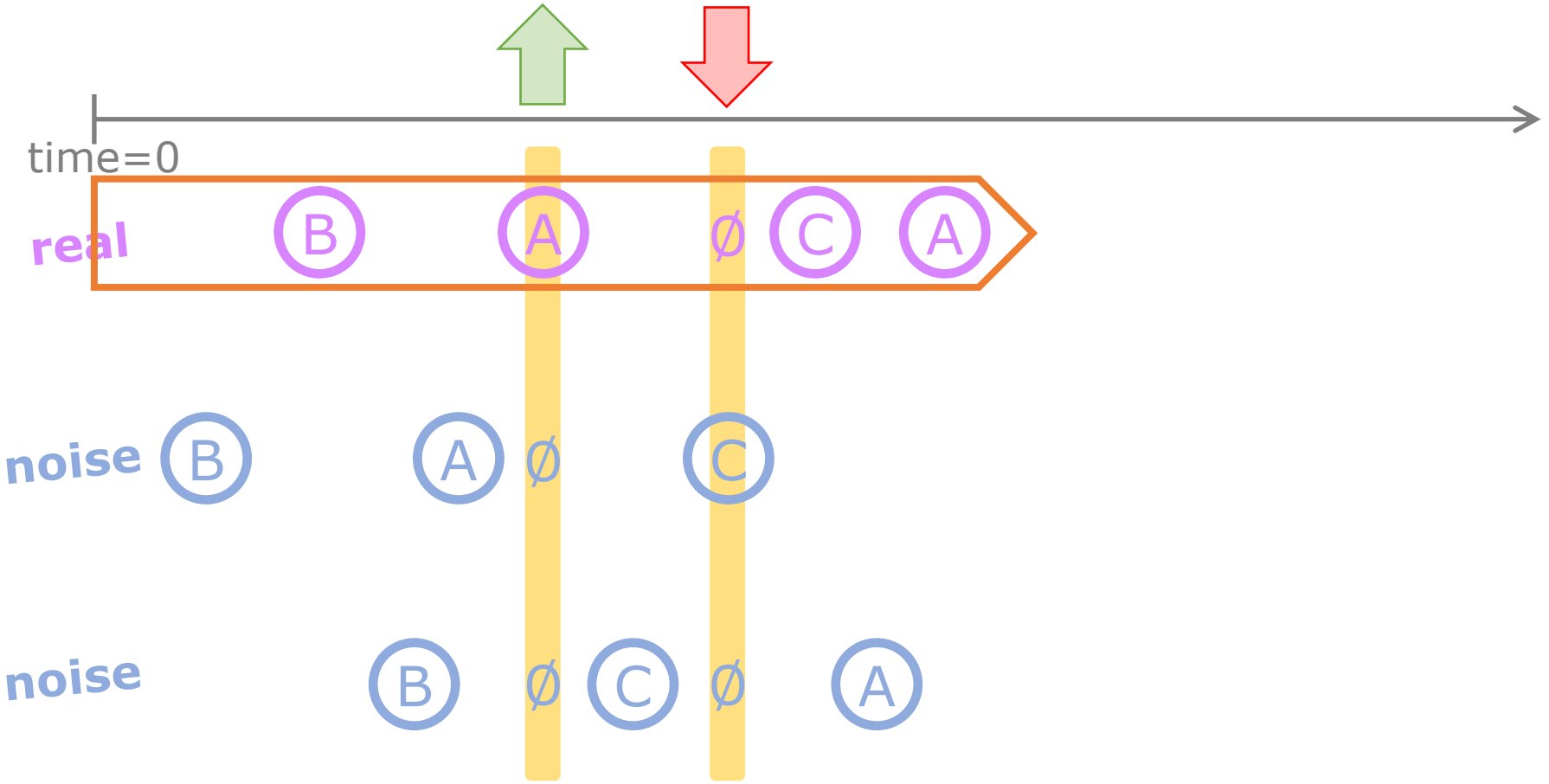
Which Is Real?

NCE: Max log prob of *correct discrimination*



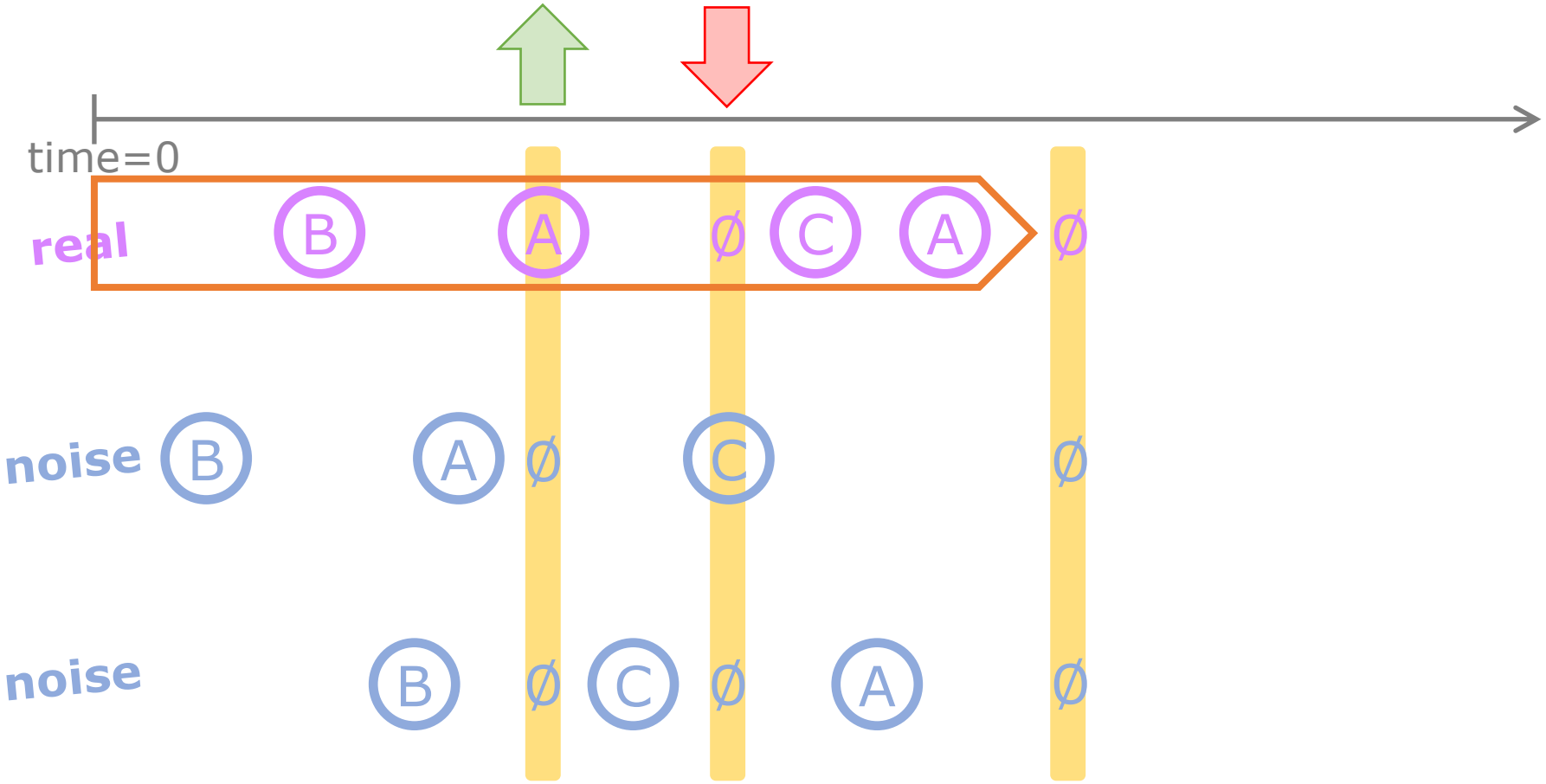
Which Is Real?

NCE: Max log prob of *correct discrimination*



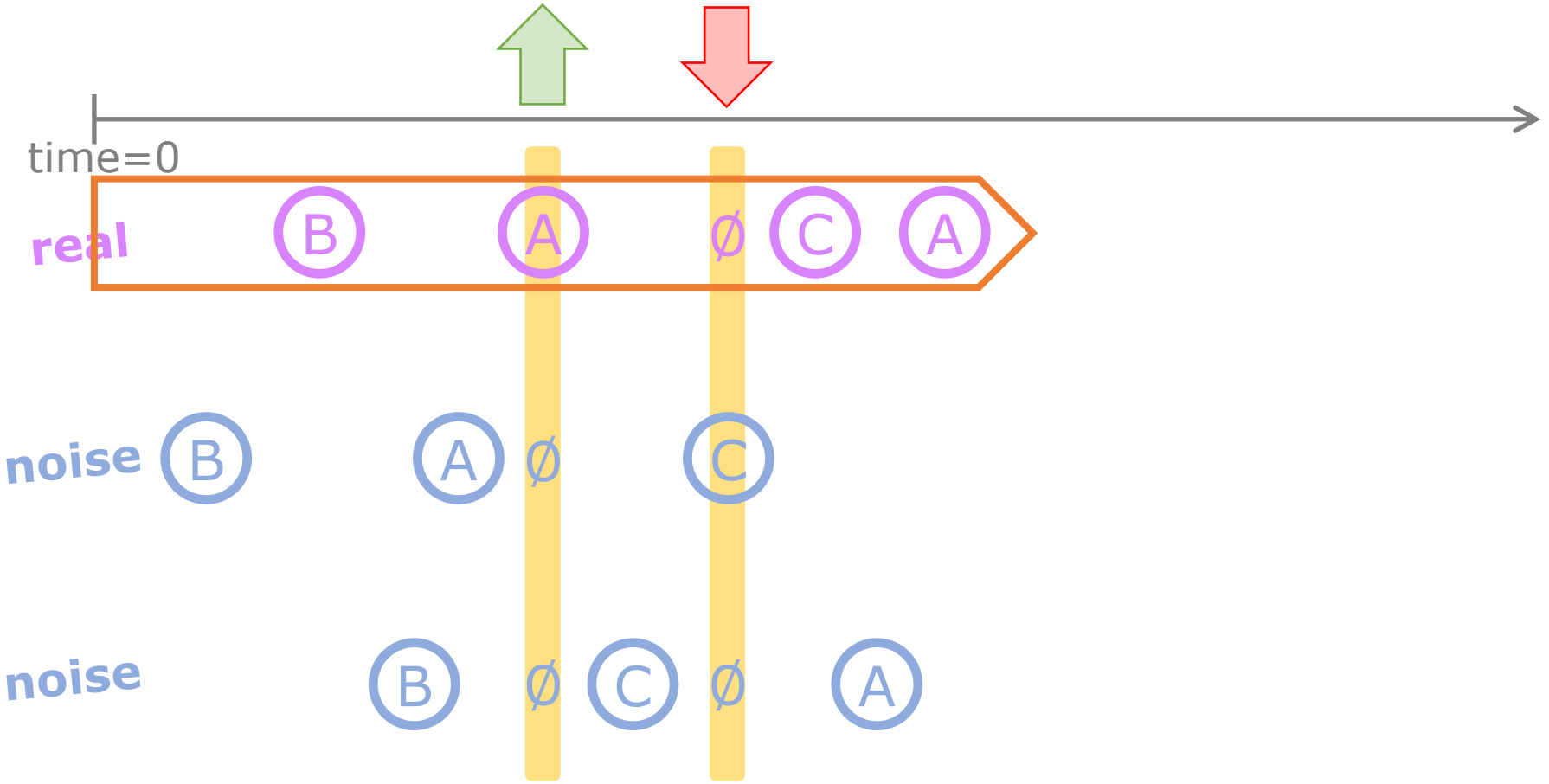
Which Is Real?

NCE: Max log prob of *correct discrimination*



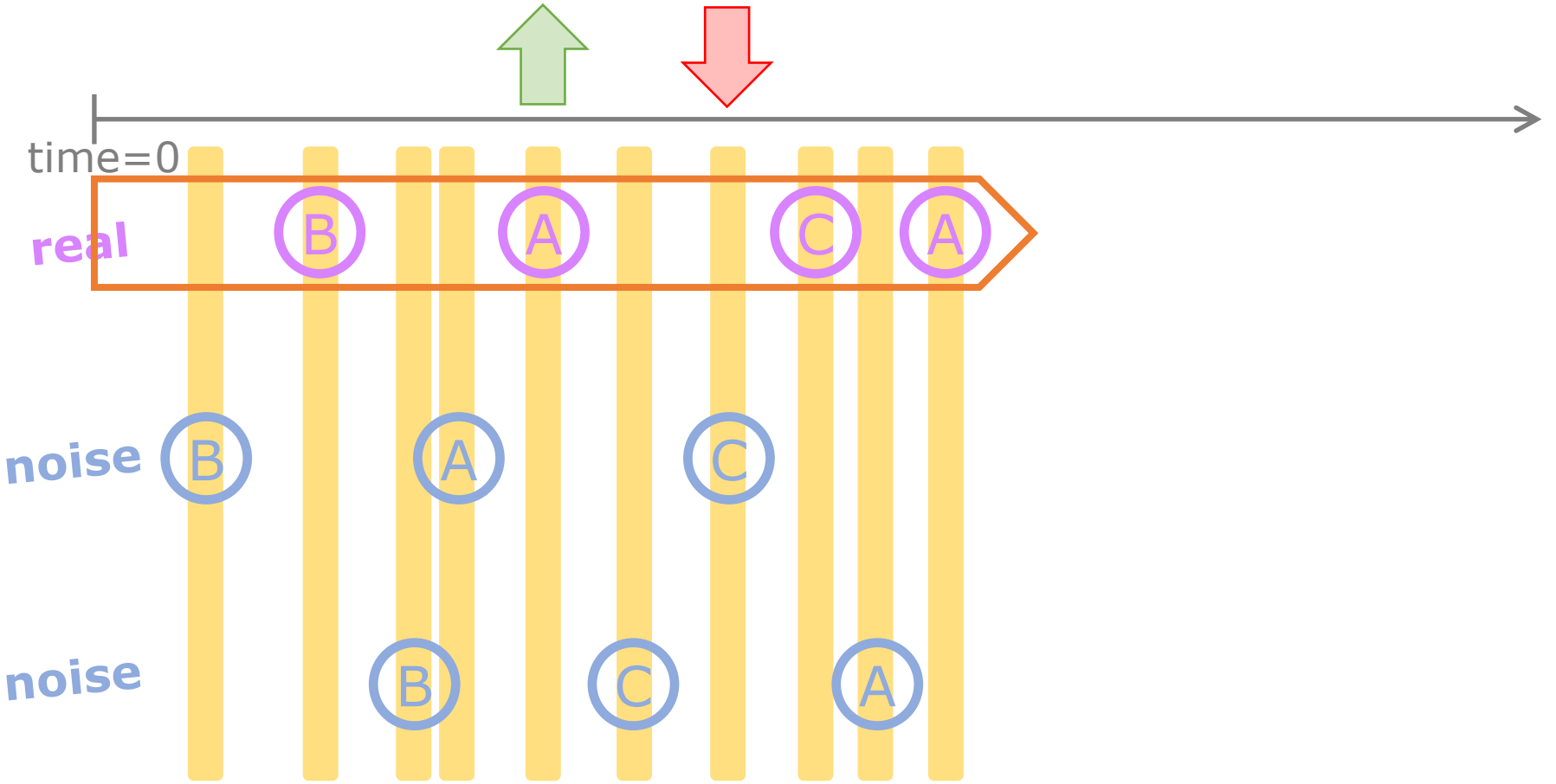
Which Is Real?

NCE: Max log prob of *correct discrimination*



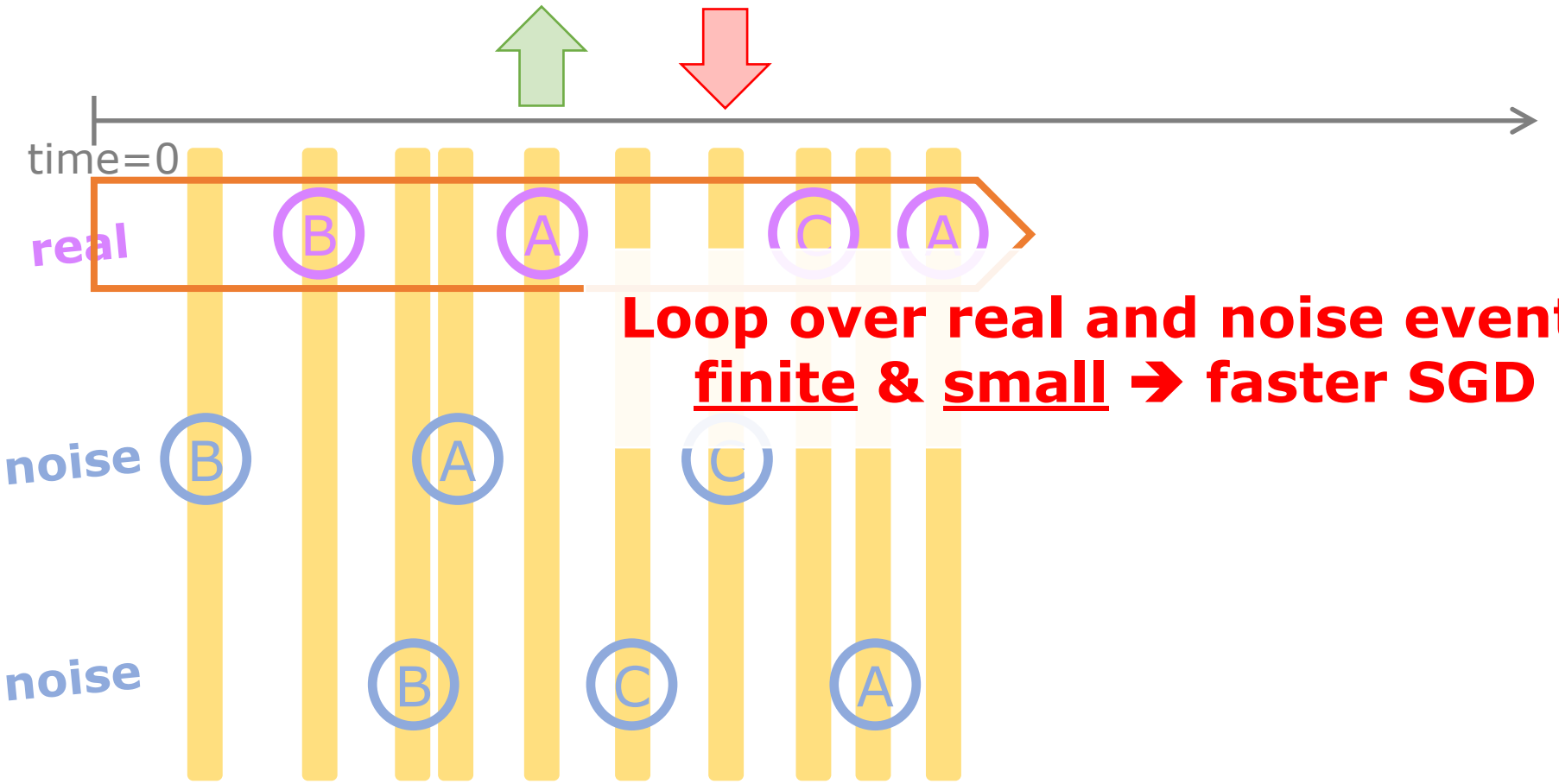
Which Is Real?

NCE: Max log prob of *correct discrimination*



Which Is Real?

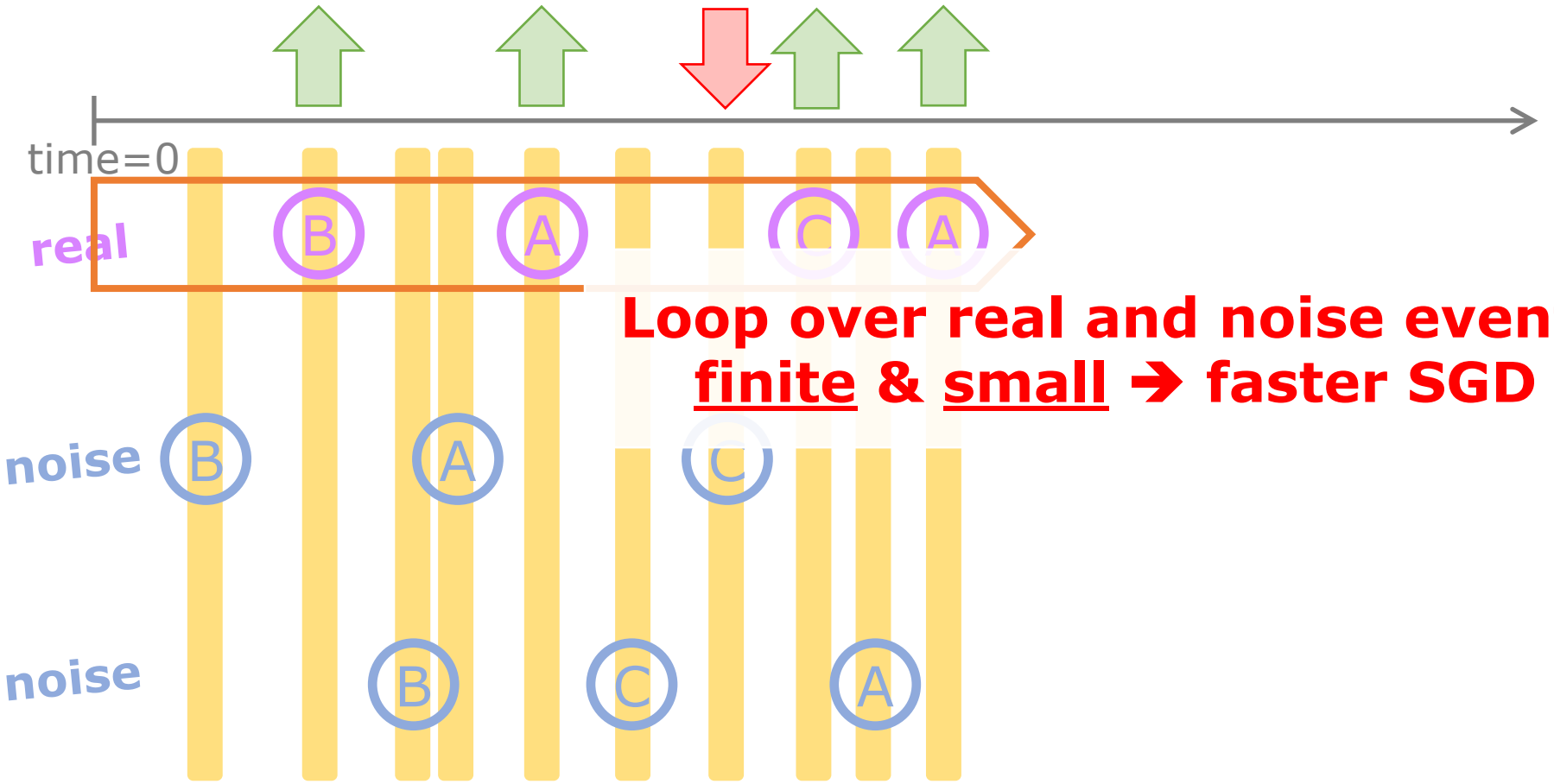
NCE: Max log prob of *correct discrimination*



Loop over real and noise events
finite & small ➔ faster SGD

Which Is Real?

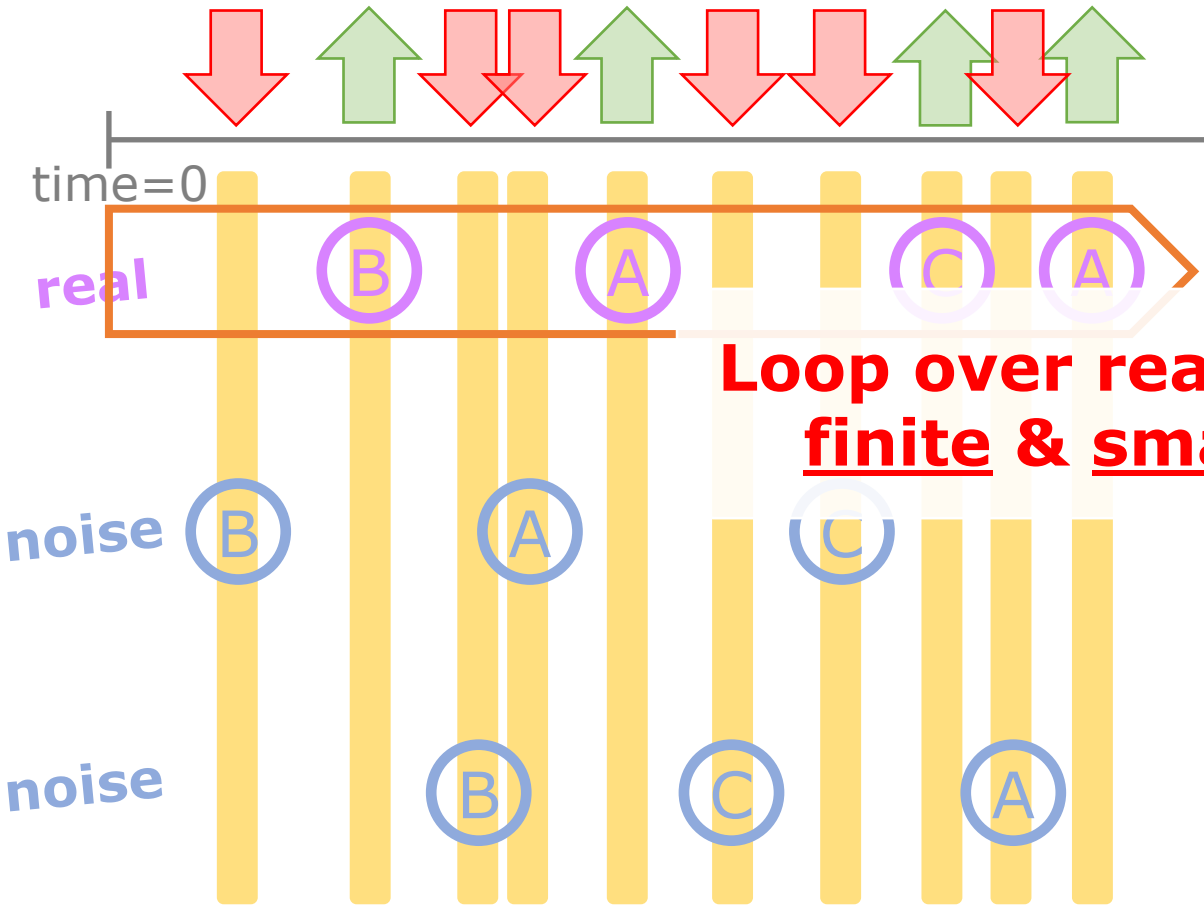
NCE: Max log prob of *correct discrimination*



Loop over real and noise events
finite & small ➔ faster SGD

Which Is Real?

NCE: Max log prob of *correct discrimination*



Loop over real and noise events
finite & small ➔ faster SGD

Which Is Real?

Let’s write some code

http://bburl/tpp-lab-p3

NCE vs MLE: consistent & efficient & faster

NCE vs MLE: consistent & efficient & faster

MLE NCE

NCE vs MLE: consistent & efficient & faster

	MLE	NCE
consistent?		

NCE vs MLE: consistent & efficient & faster

	MLE	NCE
consistent?	yes	yes

NCE vs MLE: consistent & efficient & faster

	MLE	NCE
consistent?	yes	yes
sample-efficient?		

NCE vs MLE: consistent & efficient & faster

	MLE	NCE
consistent?	yes	yes
sample-efficient?	yes	(yes)

NCE vs MLE: consistent & efficient & faster

	MLE	NCE
consistent?	yes	yes
sample-efficient?	yes	(yes)
sampling for …		

NCE vs MLE: consistent & efficient & faster

	MLE	NCE
consistent?	yes	yes
sample-efficient?	yes	(yes)
sampling for …	integration	noise

NCE vs MLE: consistent & efficient & faster

	MLE	NCE
consistent?	yes	yes
sample-efficient?	yes	(yes)
sampling for …	integration	noise
fewer func evals?		

NCE vs MLE: consistent & efficient & faster

	MLE	NCE
consistent?	yes	yes
sample-efficient?	yes	(yes)
sampling for …	integration	noise
fewer func evals?		yes

NCE vs MLE: consistent & efficient & faster

	MLE	NCE
consistent?	yes	yes
sample-efficient?	yes	(yes)
sampling for …	integration	noise
fewer func evals?		yes
lower runtime?		

http://bburl/tpp-slides-p3
http://bburl/tpp-lab-p3

NCE vs MLE: consistent & efficient & faster

	MLE	NCE
consistent?	yes	yes
sample-efficient?	yes	(yes)
sampling for …	integration	noise
fewer func evals?		yes
lower runtime?		yes

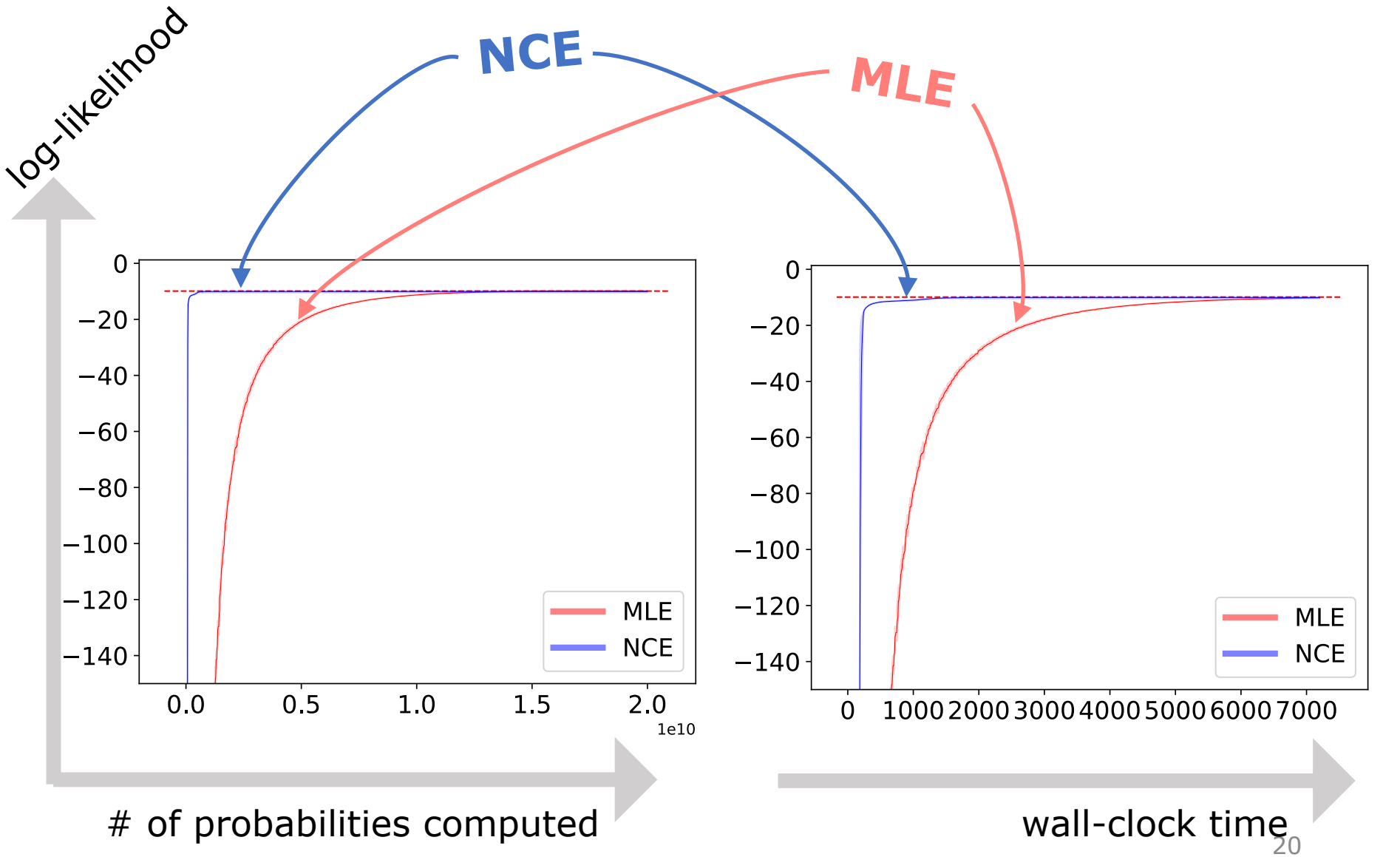
NCE vs MLE: consistent & efficient & faster

	MLE	NCE
consistent?	yes	yes
sample-efficient?	yes	(yes)
sampling for …	integration	noise
fewer func evals?		yes
lower runtime?		yes
better learning curve?		

NCE vs MLE: consistent & efficient & faster

	MLE	NCE
consistent?	yes	yes
sample-efficient?	yes	(yes)
sampling for …	integration	noise
fewer func evals?		yes
lower runtime?		yes
better learning curve?		yes

NCE vs MLE: what it typically looks like



NCE: More in Mei et al. NeurIPS 2020

NCE: More in Mei et al. NeurIPS 2020

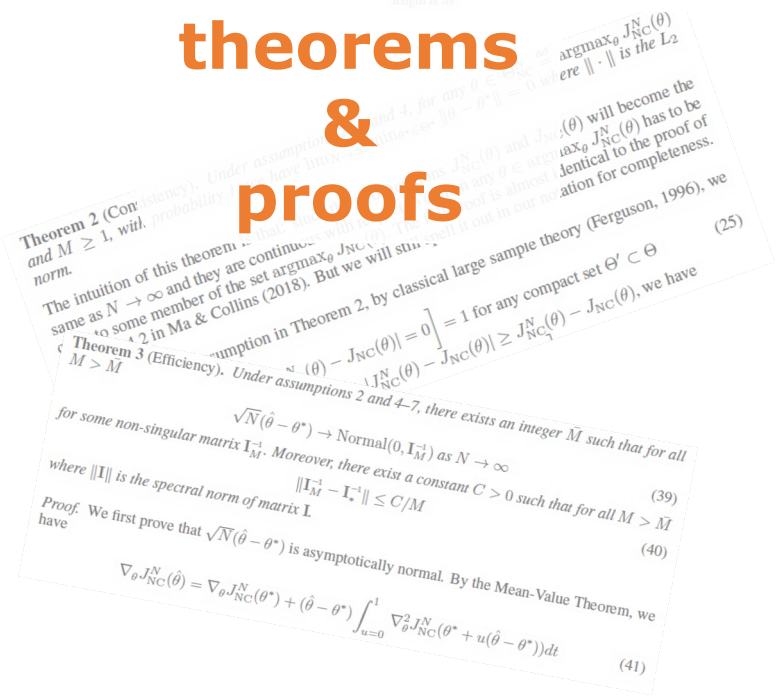
Theorem 1 (Optimality). *Under assumptions 1 and 2, $\theta \in \operatorname{argmax}_\theta J_{\mathrm{NC}}(\theta)$ if and only if $p_\theta = p^\star$.*

We first need to highlight the key insight that $H_\theta(k, t, x^0_{[0,t)})$ in equation (20) is the negative cross-entropy between the following two discrete distributions over $\{\varnothing, 1, \ldots, K\}$:

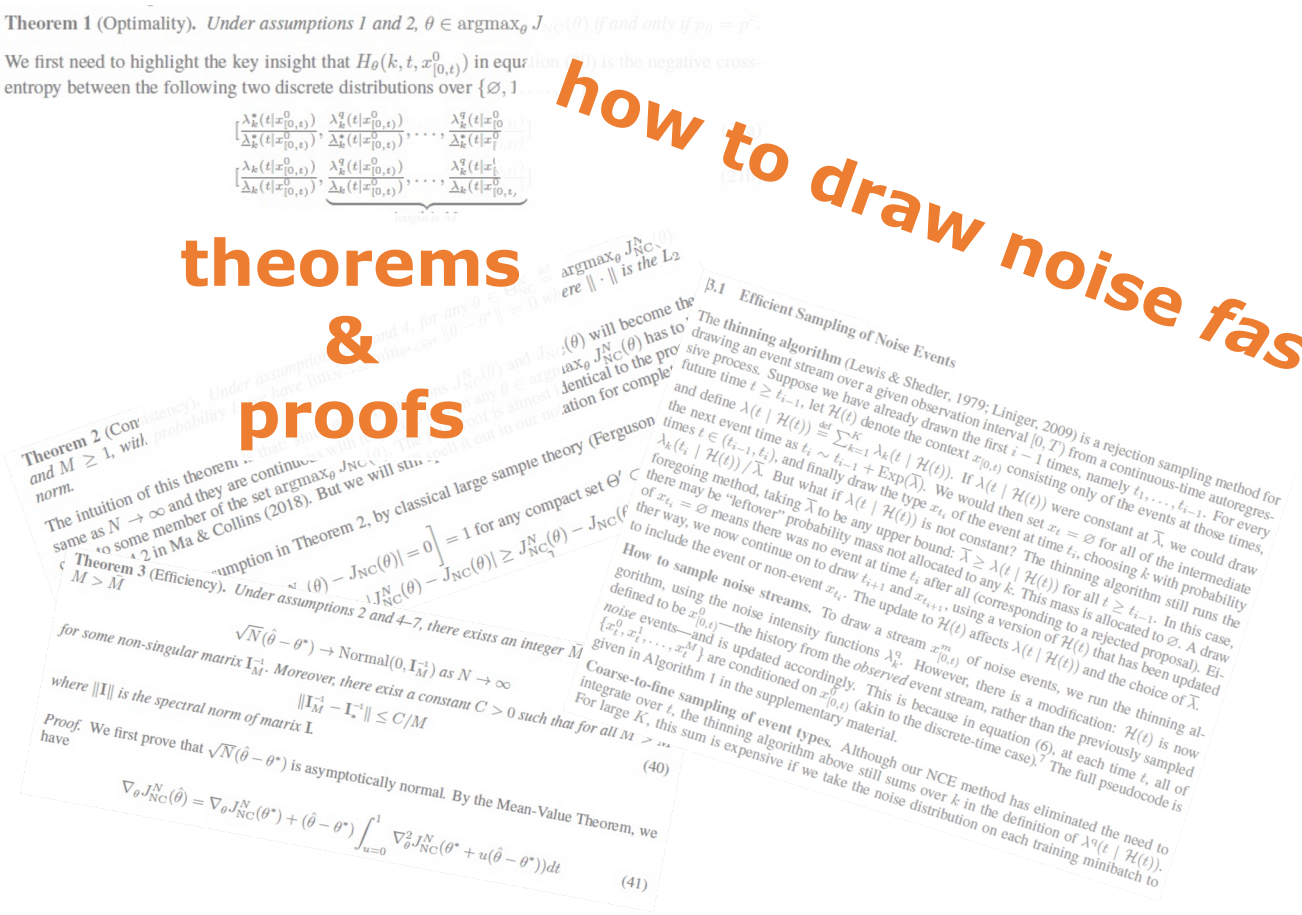
$$[\frac{\lambda^\star_k(t|x^0_{[0,t)})}{\Delta^\star_k(t|x^0_{[0,t)})}, \frac{\lambda^q_k(t|x^0_{[0,t)})}{\Delta^\star_k(t|x^0_{[0,t)})}, \ldots, \frac{\lambda^q_k(t|x^0_{[0,t)})}{\Delta^\star_k(t|x^0_{[0,t)})}] \tag{21a}$$

$$[\frac{\lambda_k(t|x^0_{[0,t)})}{\Delta_k(t|x^0_{[0,t)})}, \underbrace{\frac{\lambda^q_k(t|x^0_{[0,t)})}{\Delta_k(t|x^0_{[0,t)})}, \ldots, \frac{\lambda^q_k(t|x^0_{[0,t)})}{\Delta_k(t|x^0_{[0,t)})}}] \tag{21b}$$

theorems & proofs



NCE: More in Mei et al. NeurIPS 2020



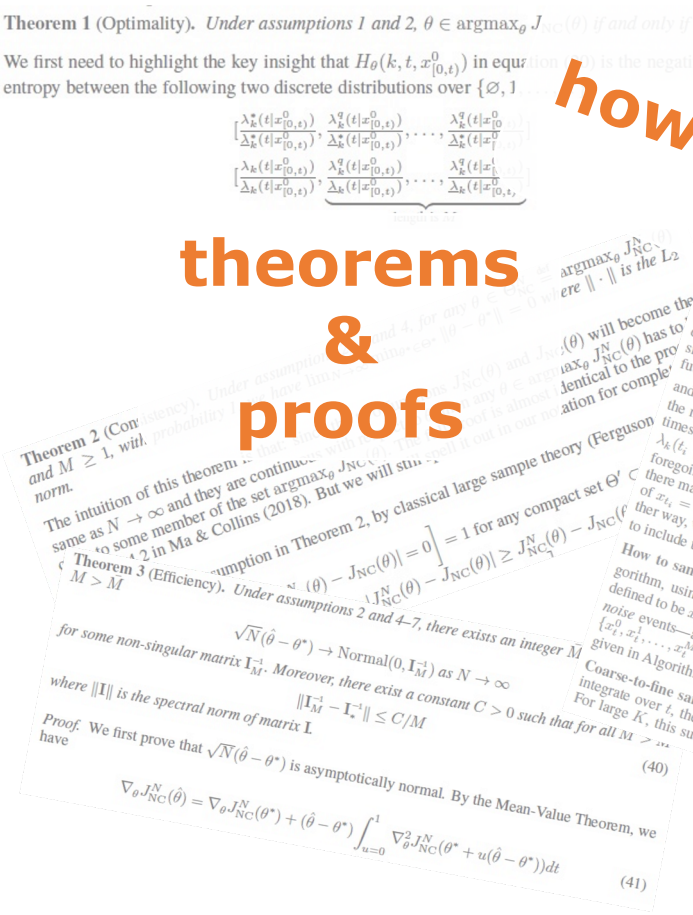
http://bburl/tpp-slides-p3
http://bburl/tpp-lab-p3

NCE: More in Mei et al. NeurIPS 2020

how to draw noise

theorems & proofs

more results & analysis



More Topics

More Topics

Domain Knowledge

KownEvolve: Trivedi et al. ICML 2017

DyRep: Trivedi et al. ICLR 2019

More Topics

Domain Knowledge
KownEvolve: Trivedi et al. ICML 2017
DyRep: Trivedi et al. ICLR 2019

Continuous-Time Reinforcement Learning
Policy-based: Upadhyay et al. NeurIPS 2018

More Topics

Domain Knowledge
KownEvolve: Trivedi et al. ICML 2017
DyRep: Trivedi et al. ICLR 2019

Continuous-Time Reinforcement Learning
Policy-based: Upadhyay et al. NeurIPS 2018

Granger Causality over event types?
CAUSE: attribution methods, Zhang et al. ICML 2020

More Topics

Domain Knowledge
KownEvolve: Trivedi et al. ICML 2017
DyRep: Trivedi et al. ICLR 2019

Continuous-Time Reinforcement Learning
Policy-based: Upadhyay et al. NeurIPS 2018

Granger Causality over event types?
CAUSE: attribution methods, Zhang et al. ICML 2020

Variational Autoencoder (VAE)
NHP as decoder: Boyd et al. NeurIPS 2020

Neural Datalog Through Time

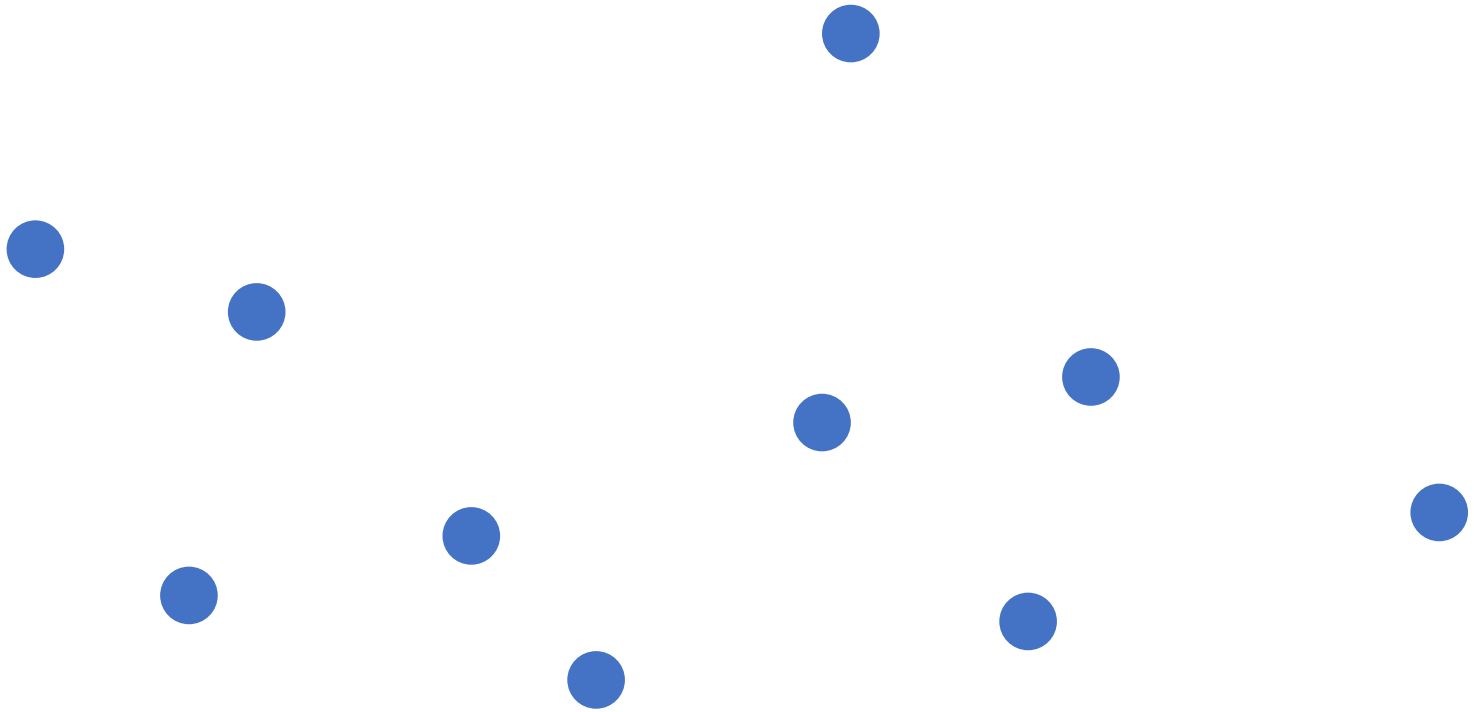
Hongyuan Mei[1], Guanghui Qin[1], Minjie Xu[2], Jason Eisner[1]

[1]Johns Hopkins University

[2]Bloomberg

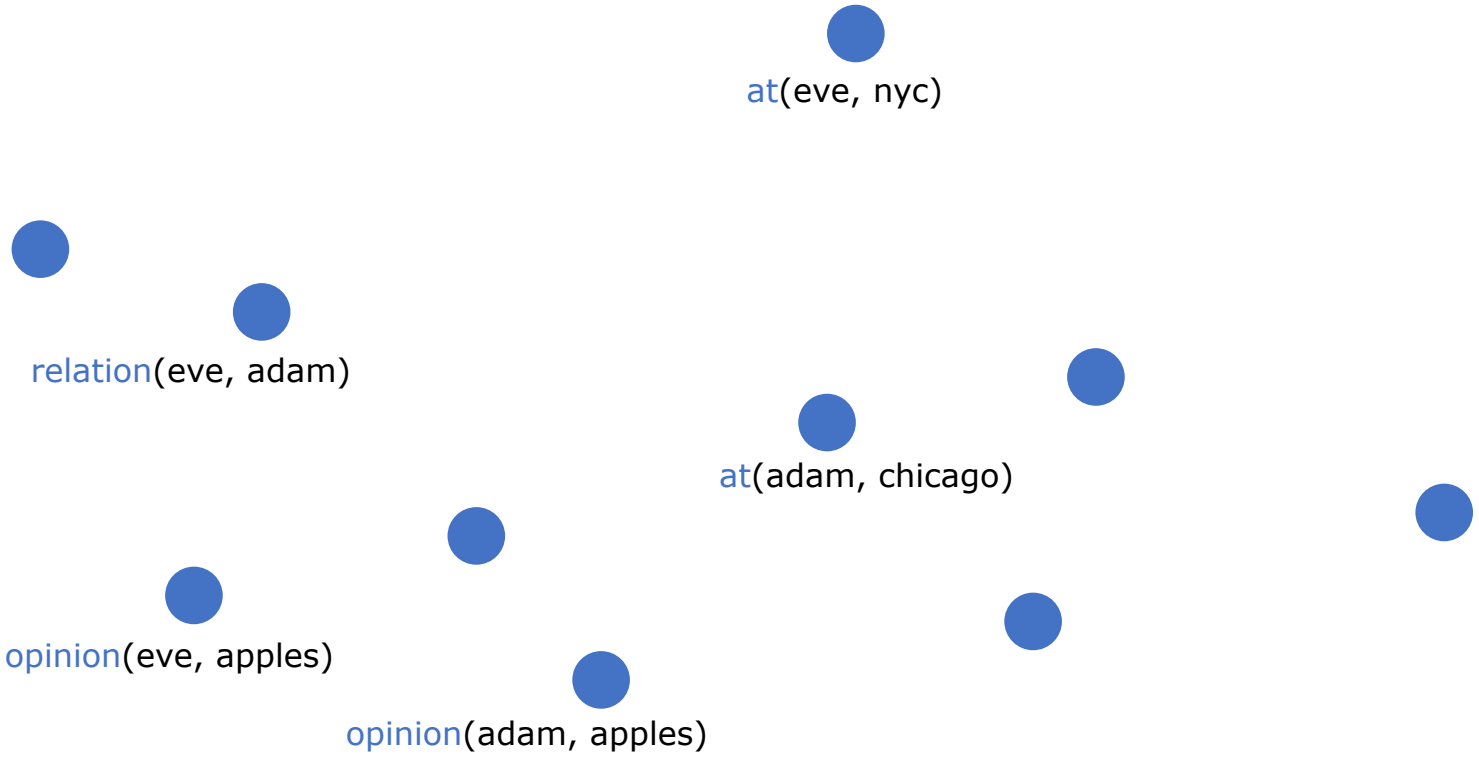
http://bburl/tpp-slides-p3
http://bburl/tpp-lab-p3

Model How a Database Changes Over Time



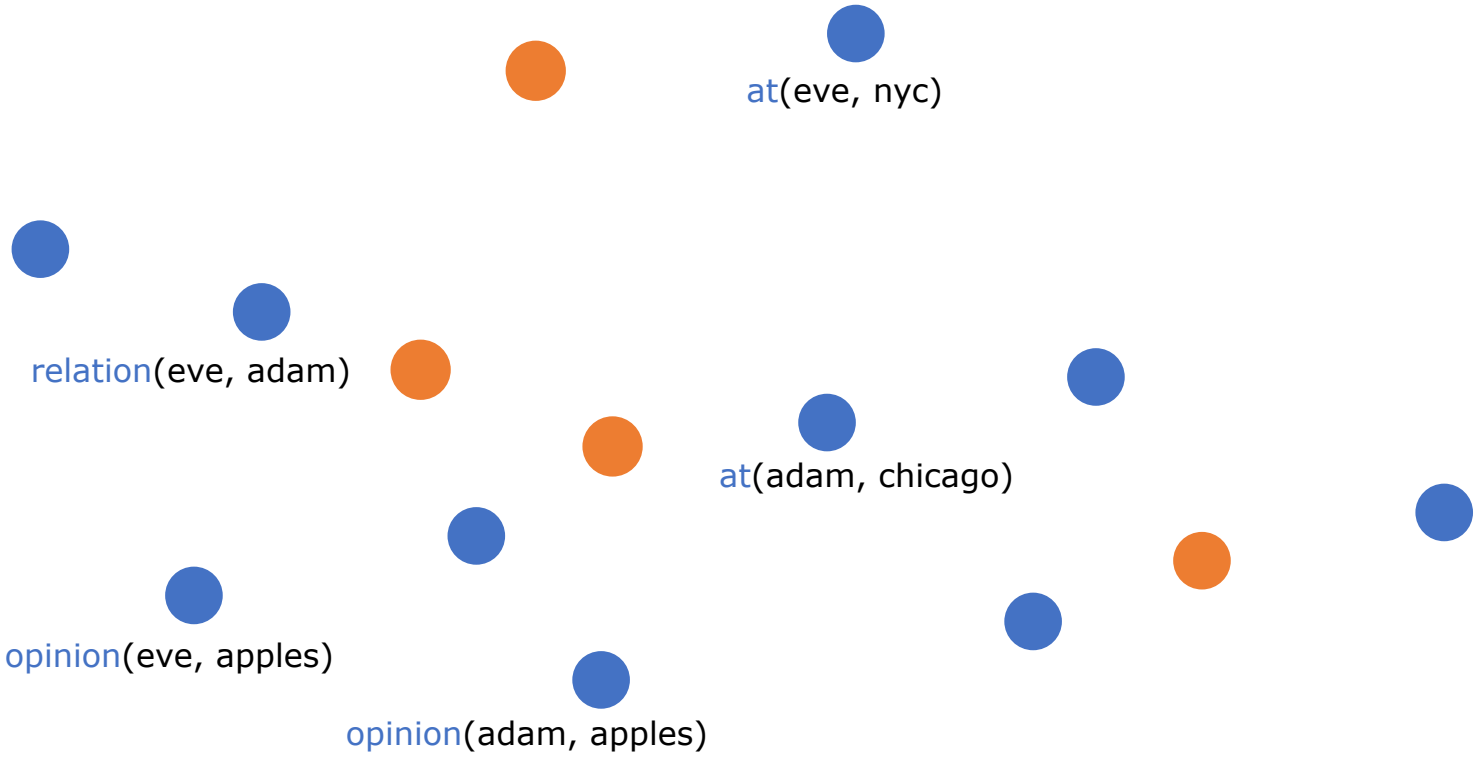
Model How a Database Changes Over Time

200,000 facts right now



Model How a Database Changes Over Time

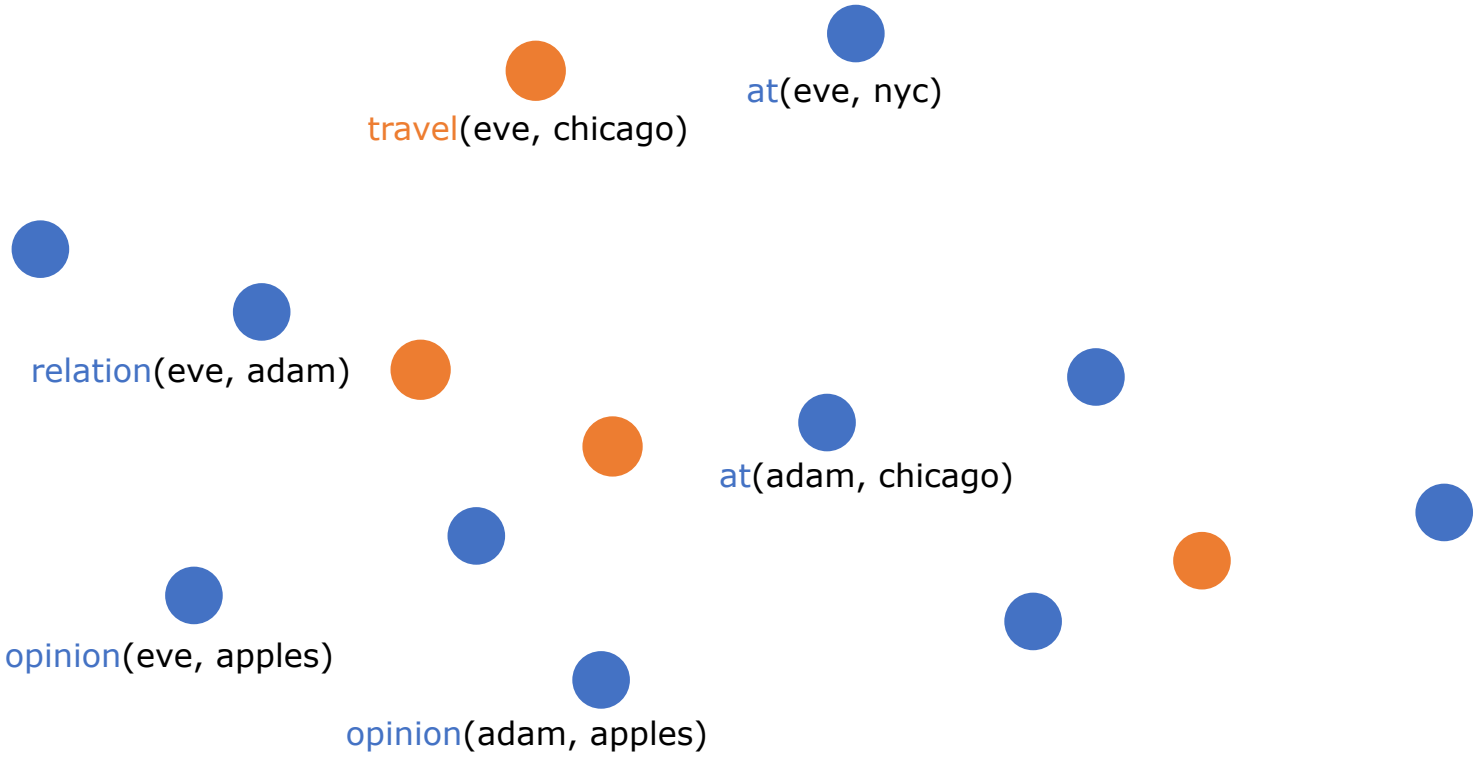
200,000 facts right now



Model How a Database Changes Over Time

200,000 facts right now

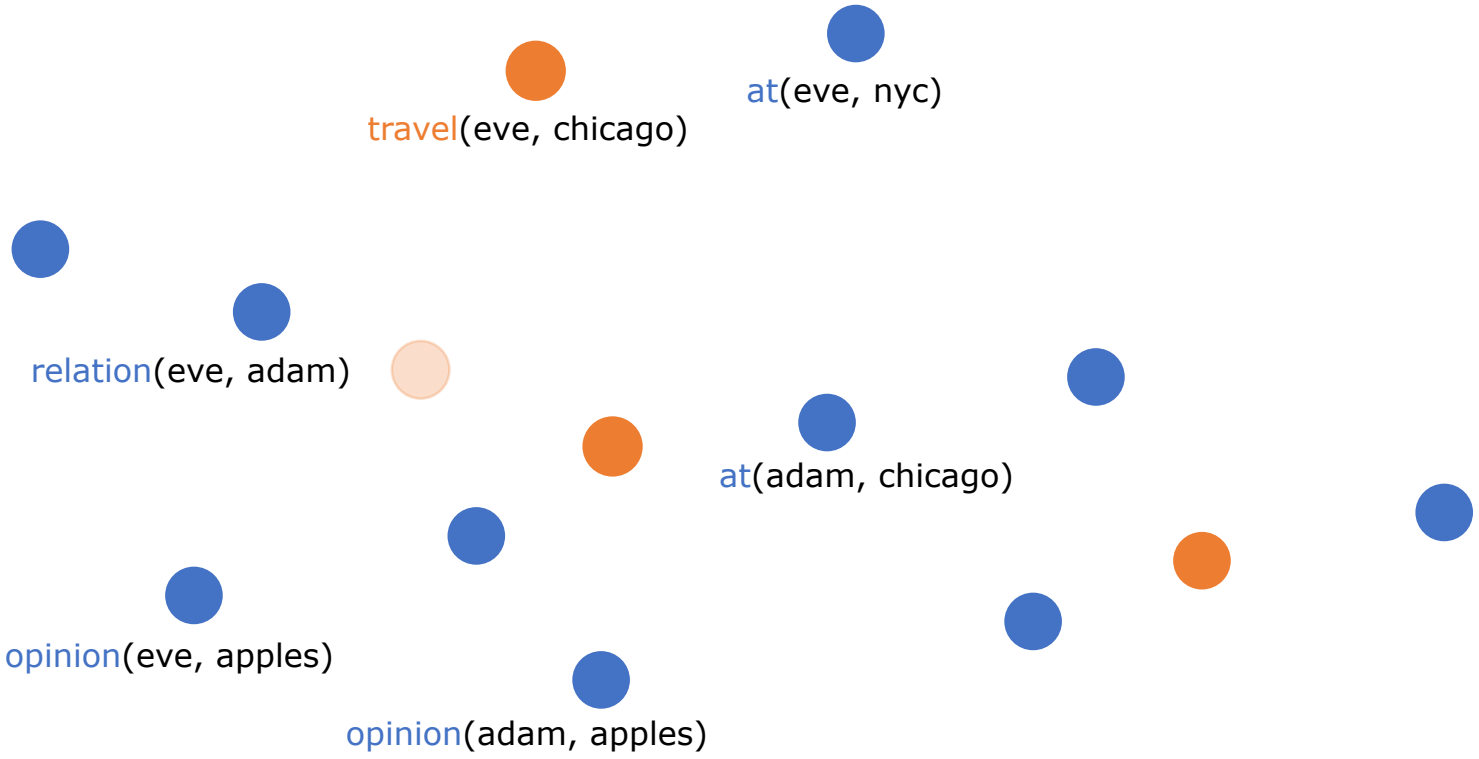
50,000 possible events right now



Model How a Database Changes Over Time

200,000 facts right now

50,000 possible events right now

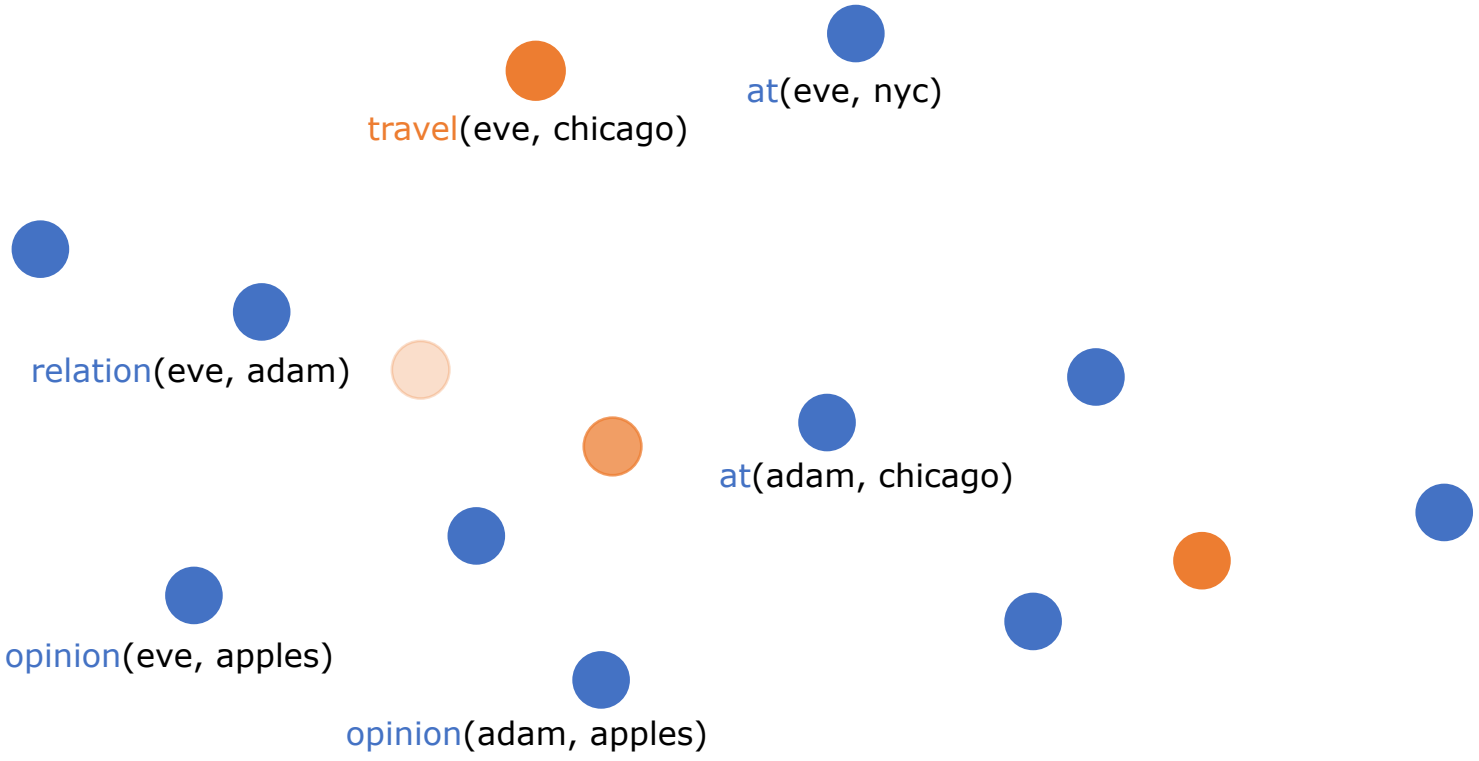


Model How a Database Changes Over Time

200,000 facts right now

50,000 possible events right now

little language
to specify a generative model
of event sequences

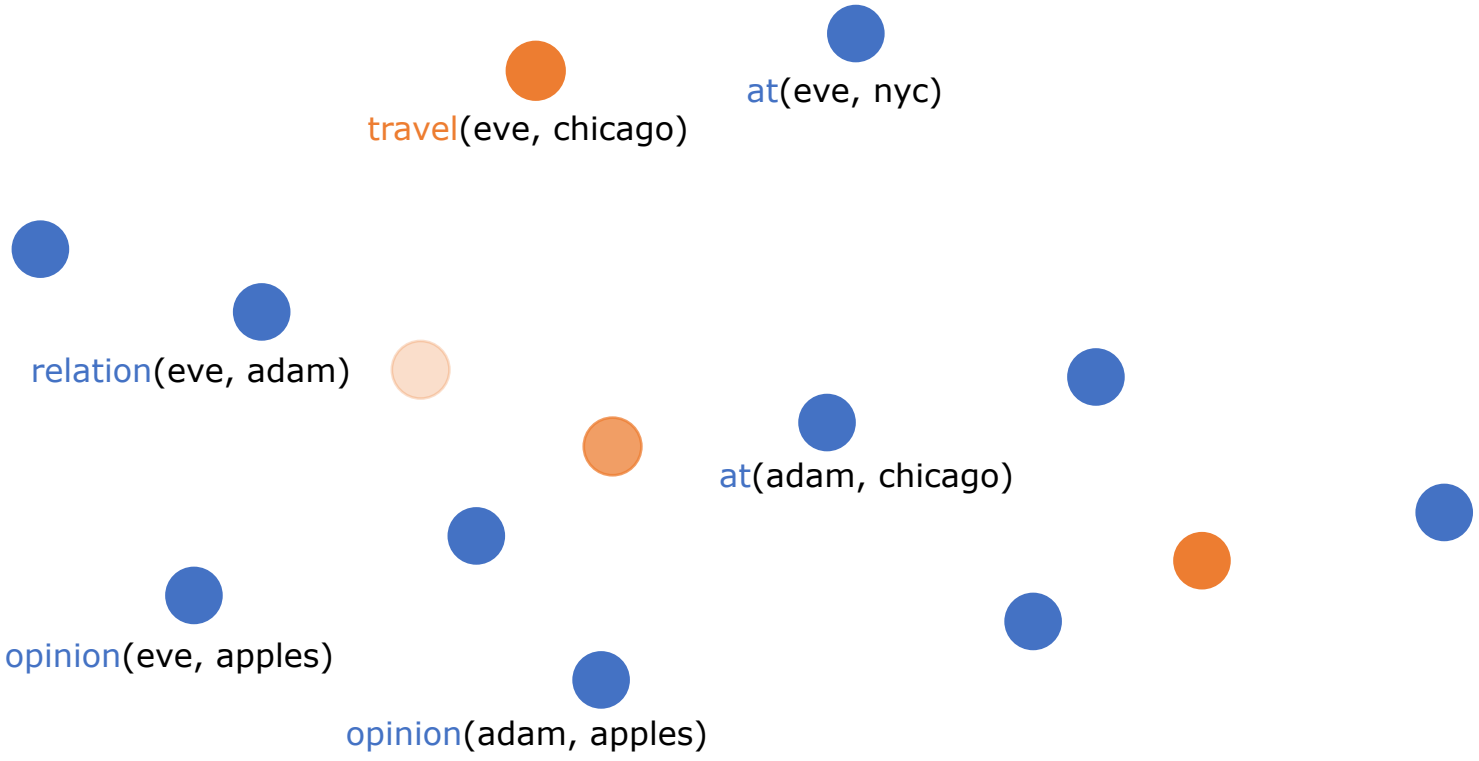


Model How a Database Changes Over Time

200,000 facts right now

50,000 possible events right now

little language
to specify a generative model
of event sequences

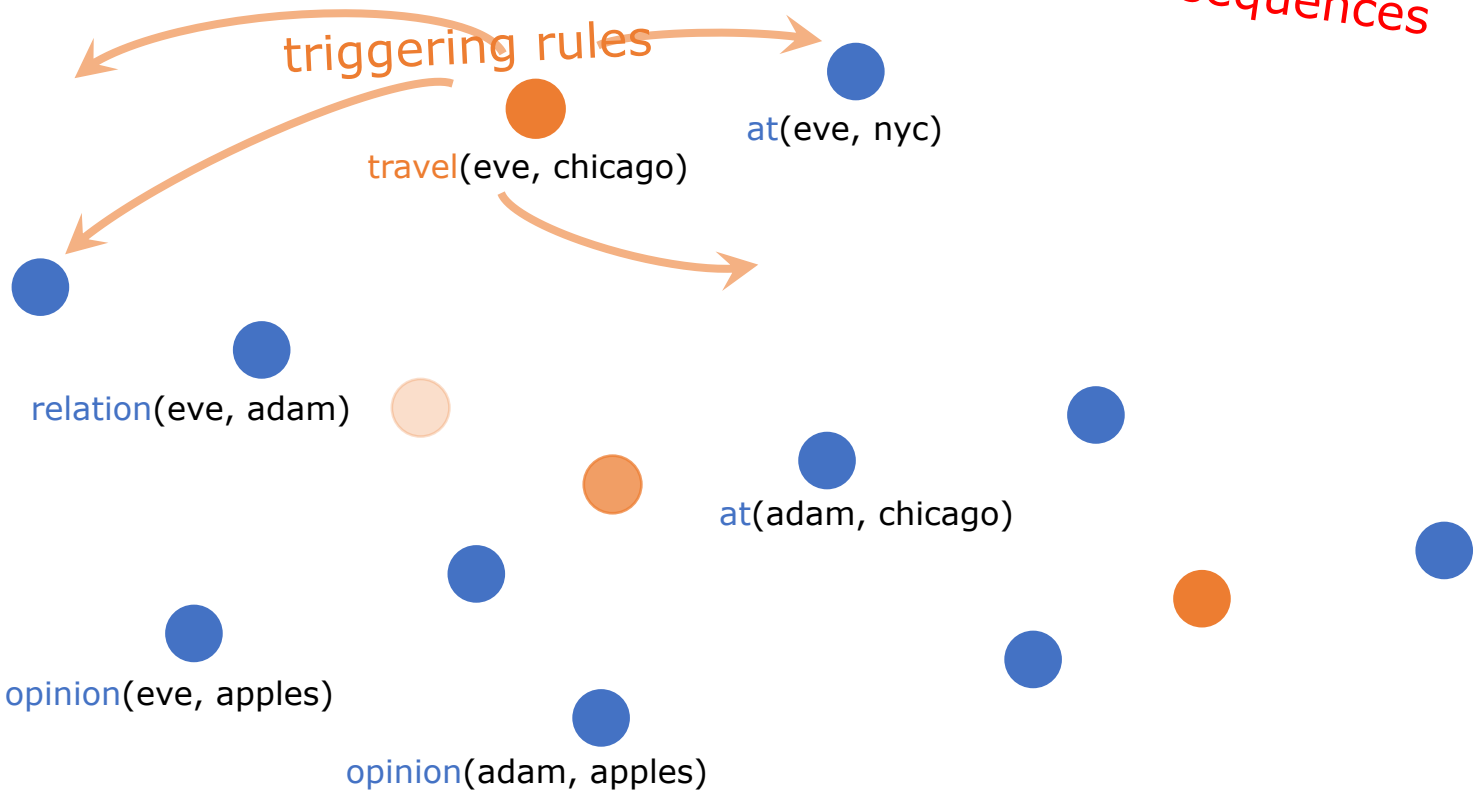


Model How a Database Changes Over Time

200,000 facts right now

50,000 possible events right now

little language
to specify a generative model
of event sequences

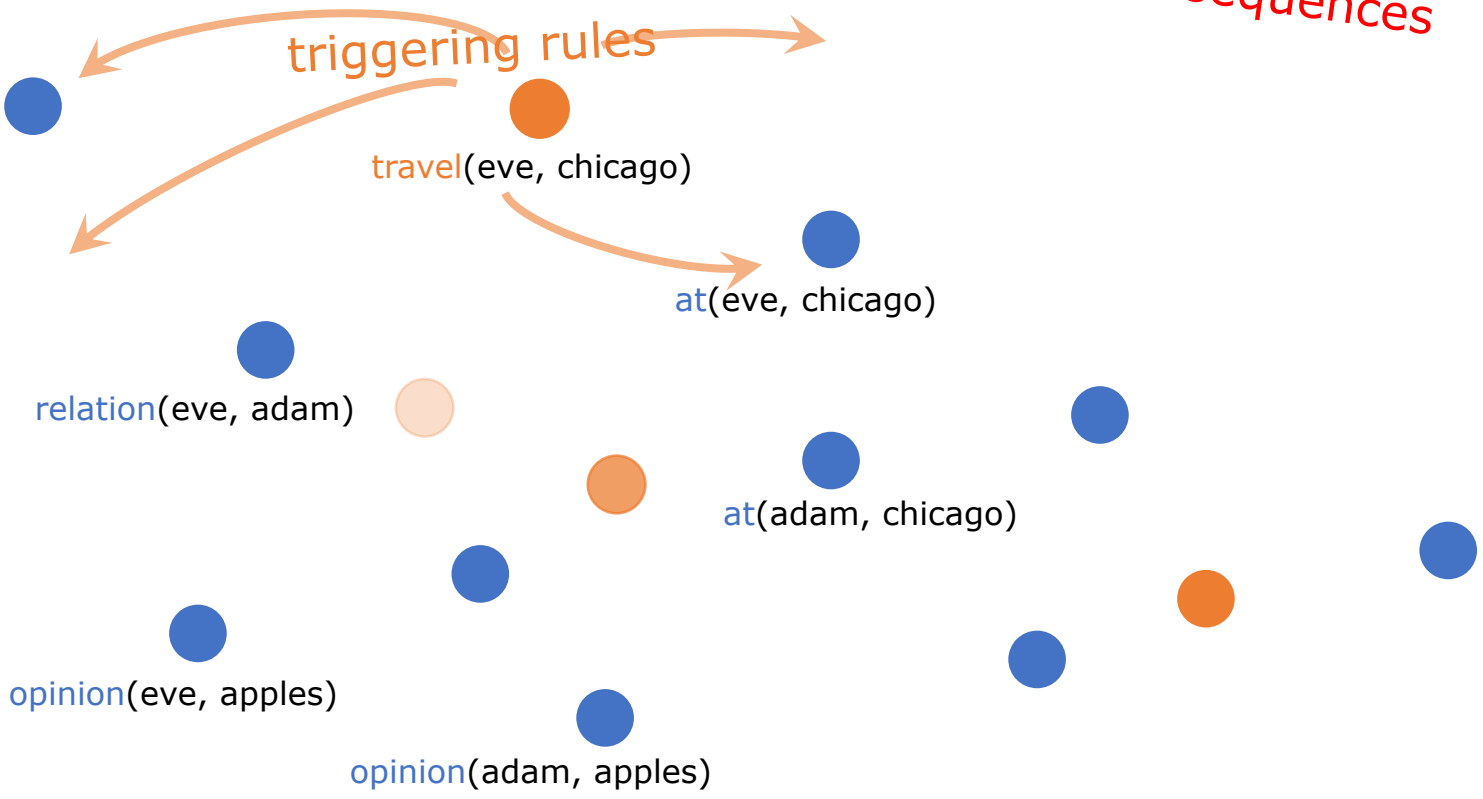


Model How a Database Changes Over Time

200,000 facts right now

50,000 possible events right now

little language
to specify a generative model
of event sequences

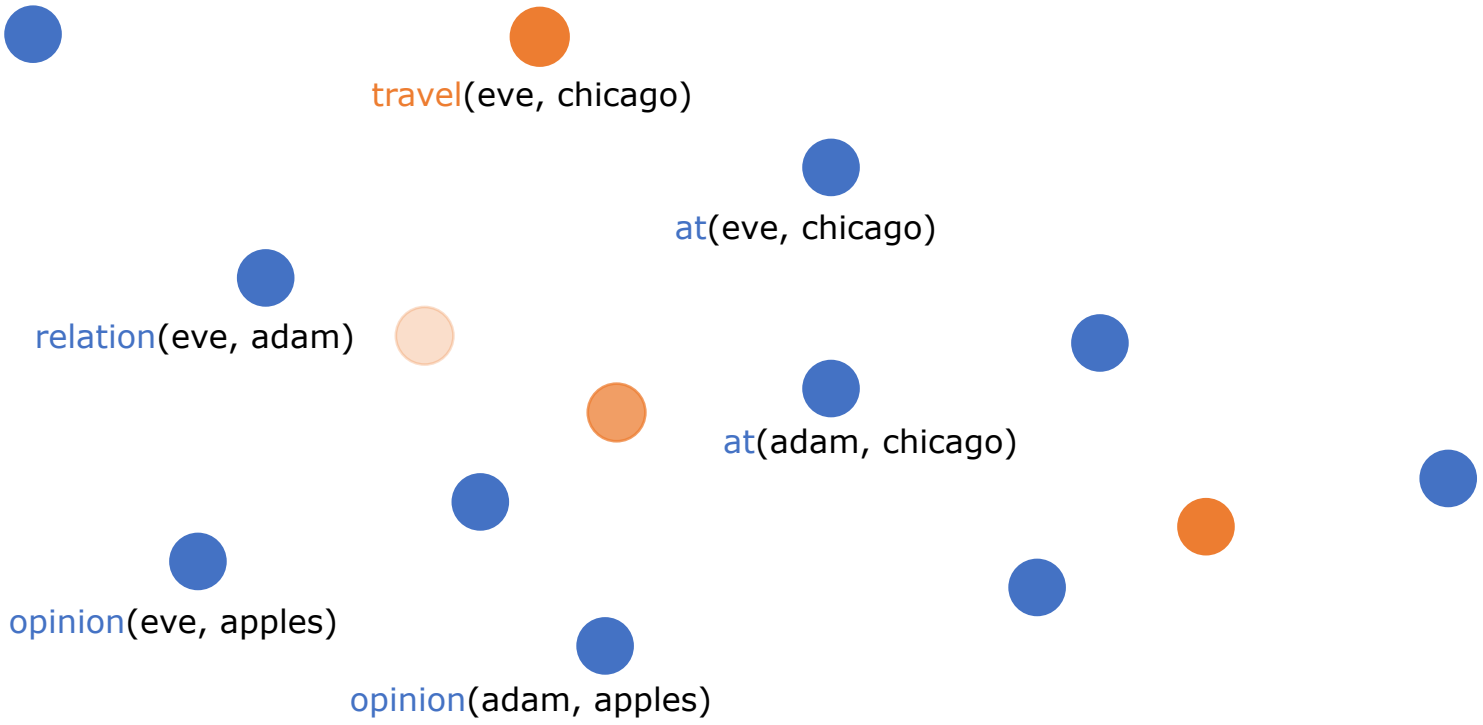


Model How a Database Changes Over Time

200,000 facts right now

50,000 possible events right now

little language
to specify a generative model
of event sequences

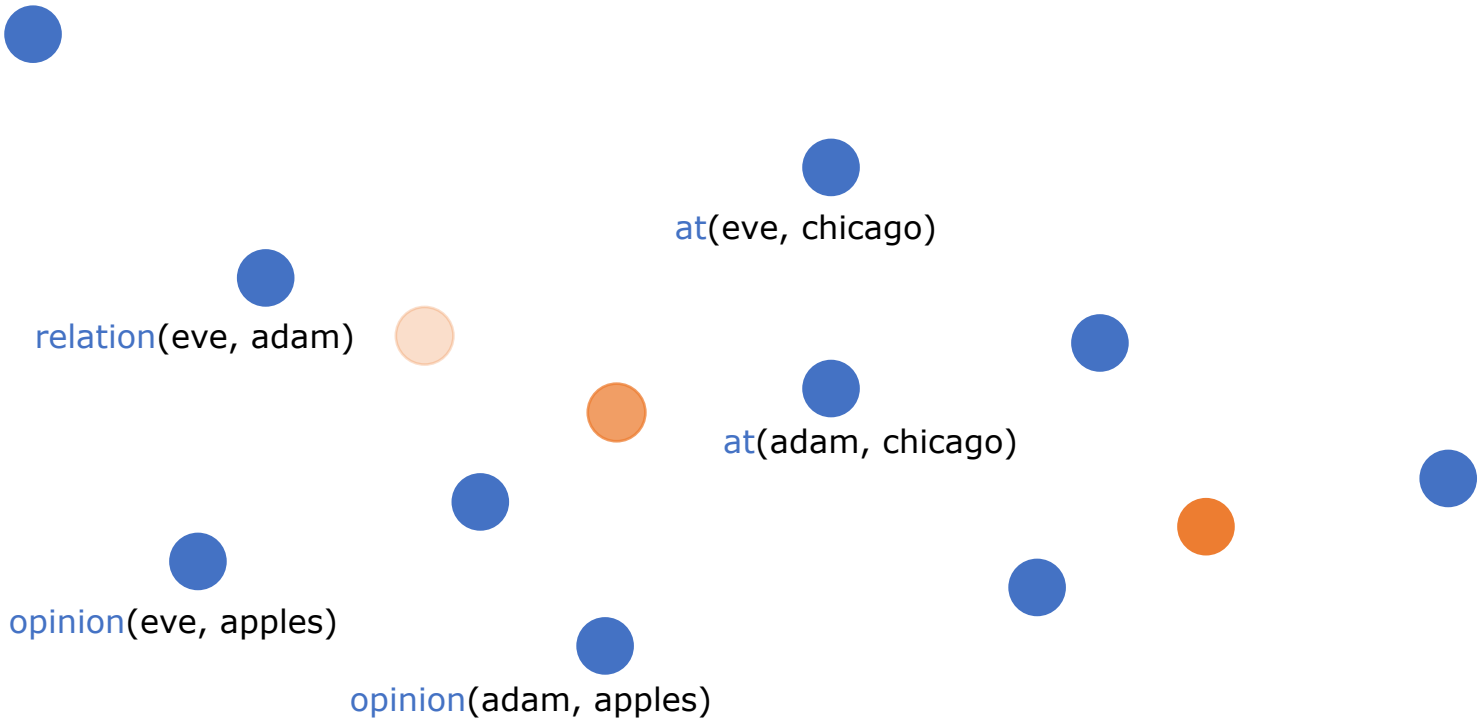


Model How a Database Changes Over Time

200,000 facts right now

50,000 possible events right now

little language
to specify a generative model
of event sequences

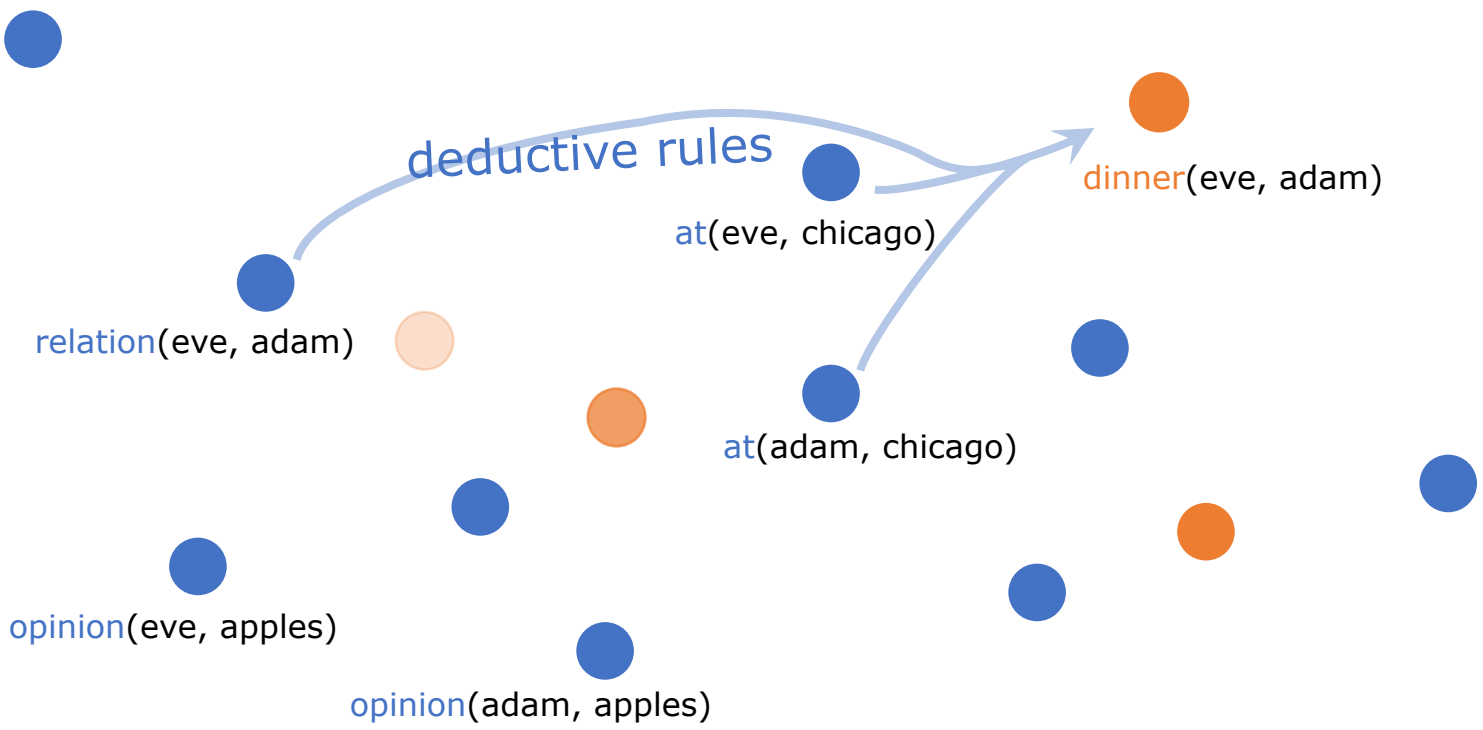


Model How a Database Changes Over Time

200,000 facts right now

50,000 possible events right now

little language
to specify a generative model
of event sequences

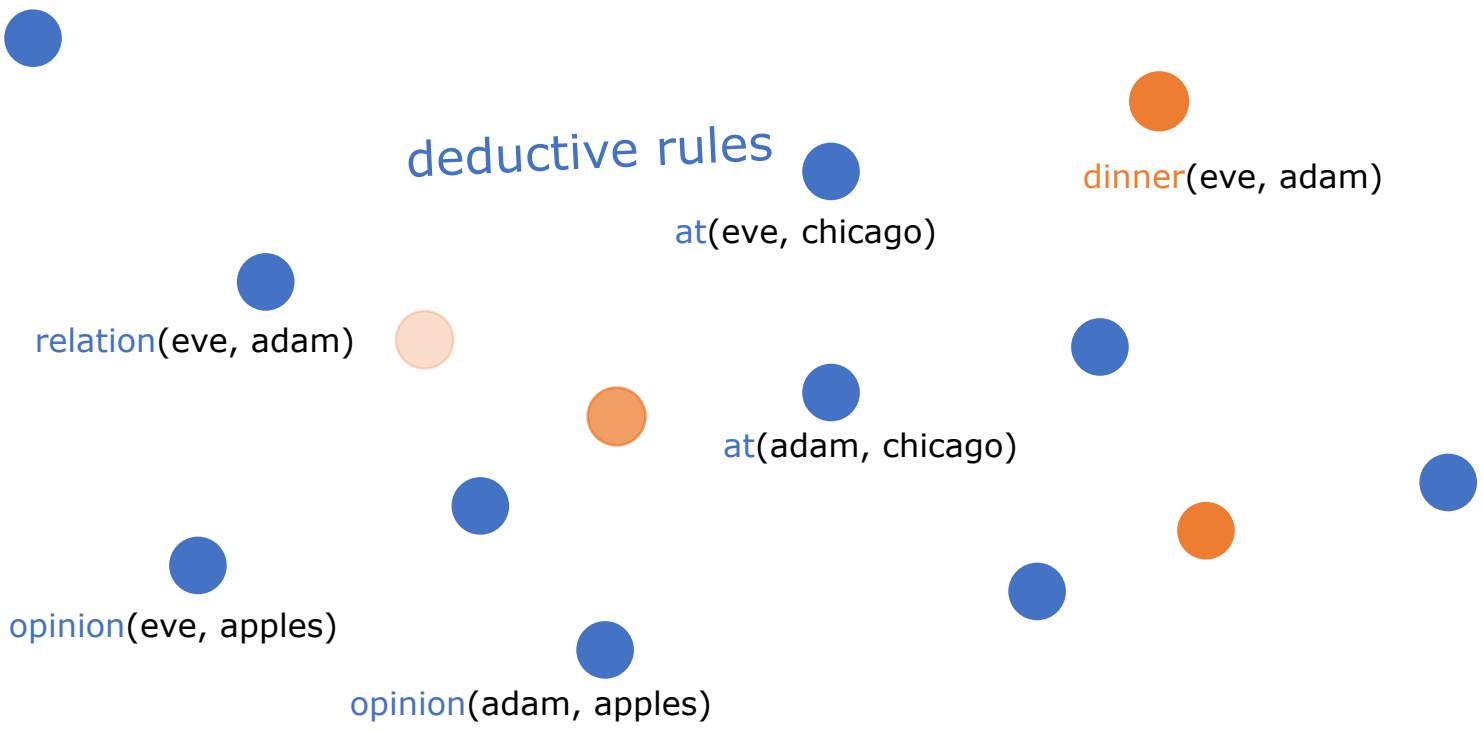


Model How a Database Changes Over Time

200,000 facts right now

50,000 possible events right now

little language
to specify a generative model
of event sequences

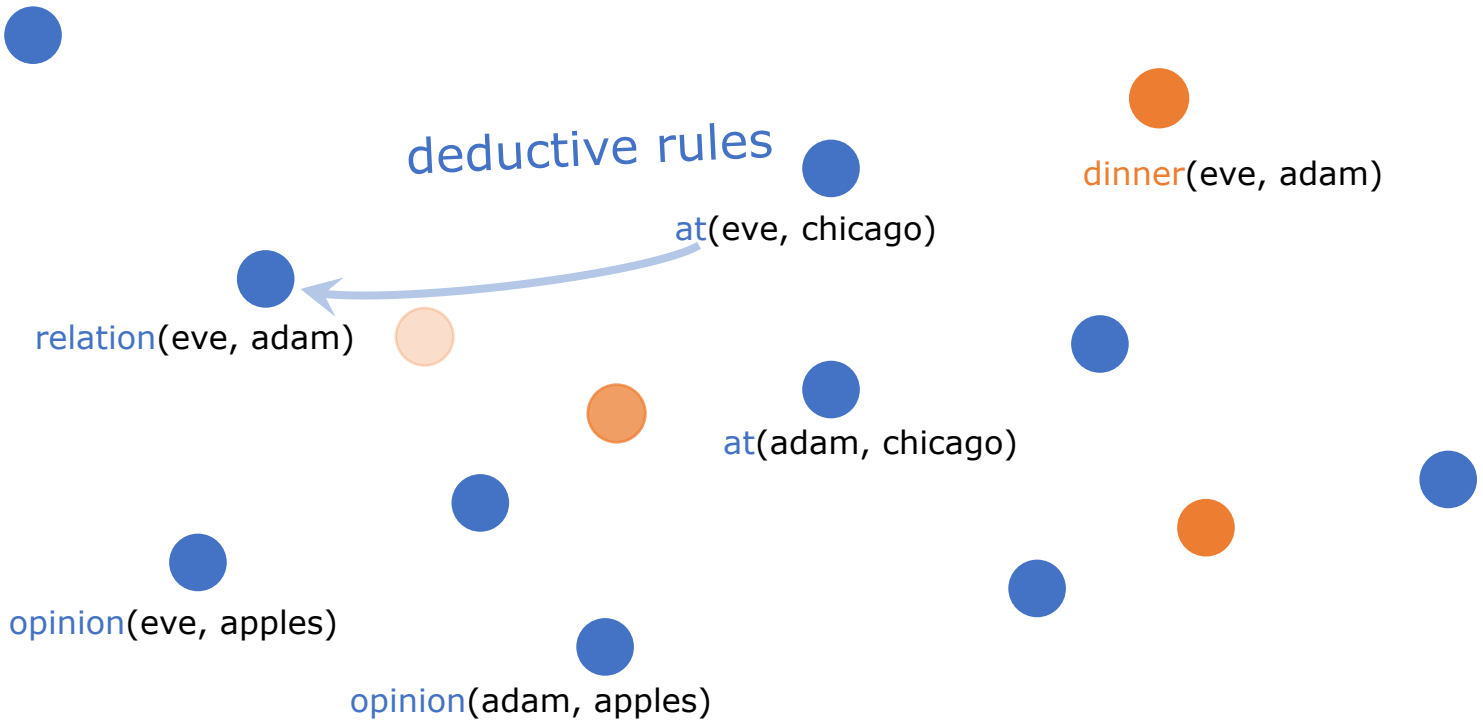


Model How a Database Changes Over Time

200,000 facts right now

50,000 possible events right now

little language
to specify a generative model
of event sequences

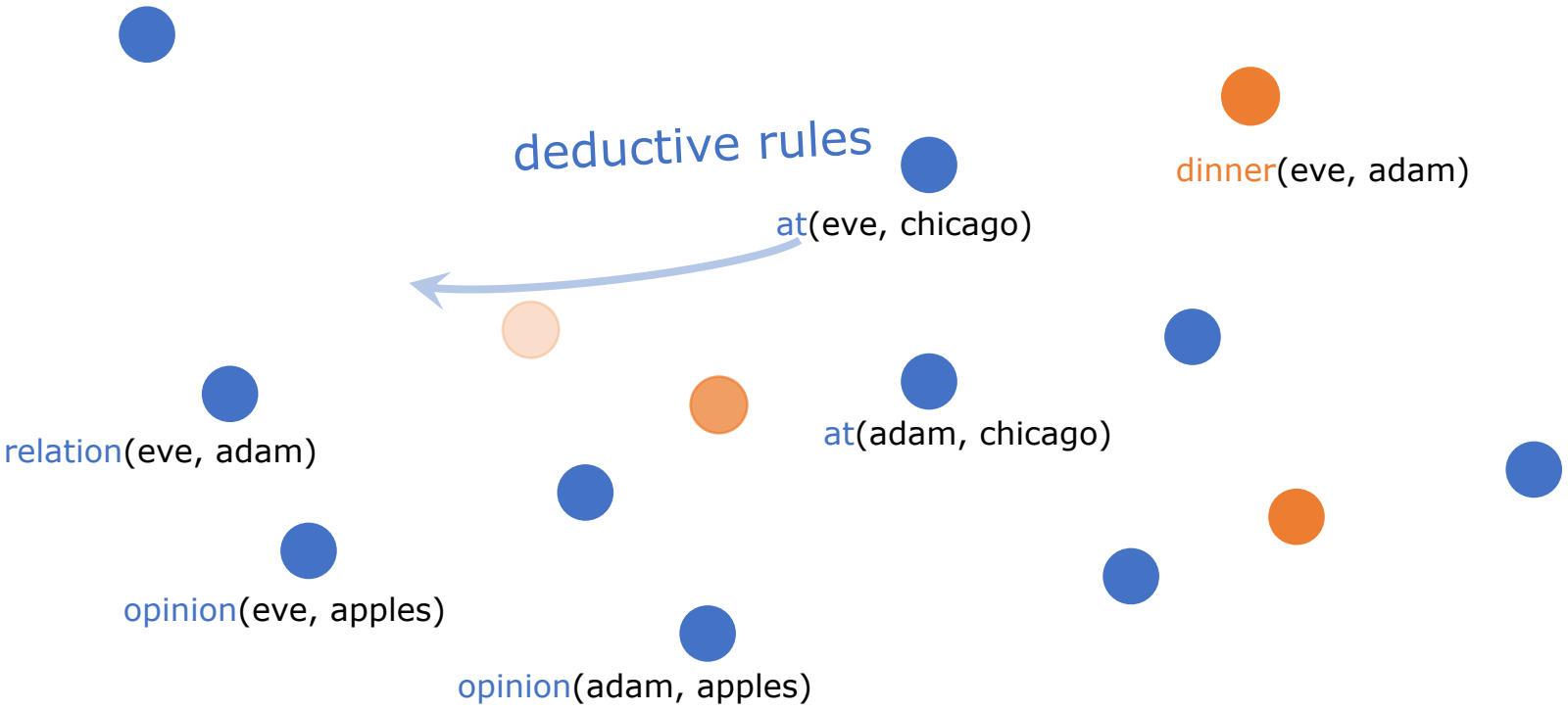


Model How a Database Changes Over Time

200,000 facts right now

50,000 possible events right now

little language
to specify a generative model
of event sequences

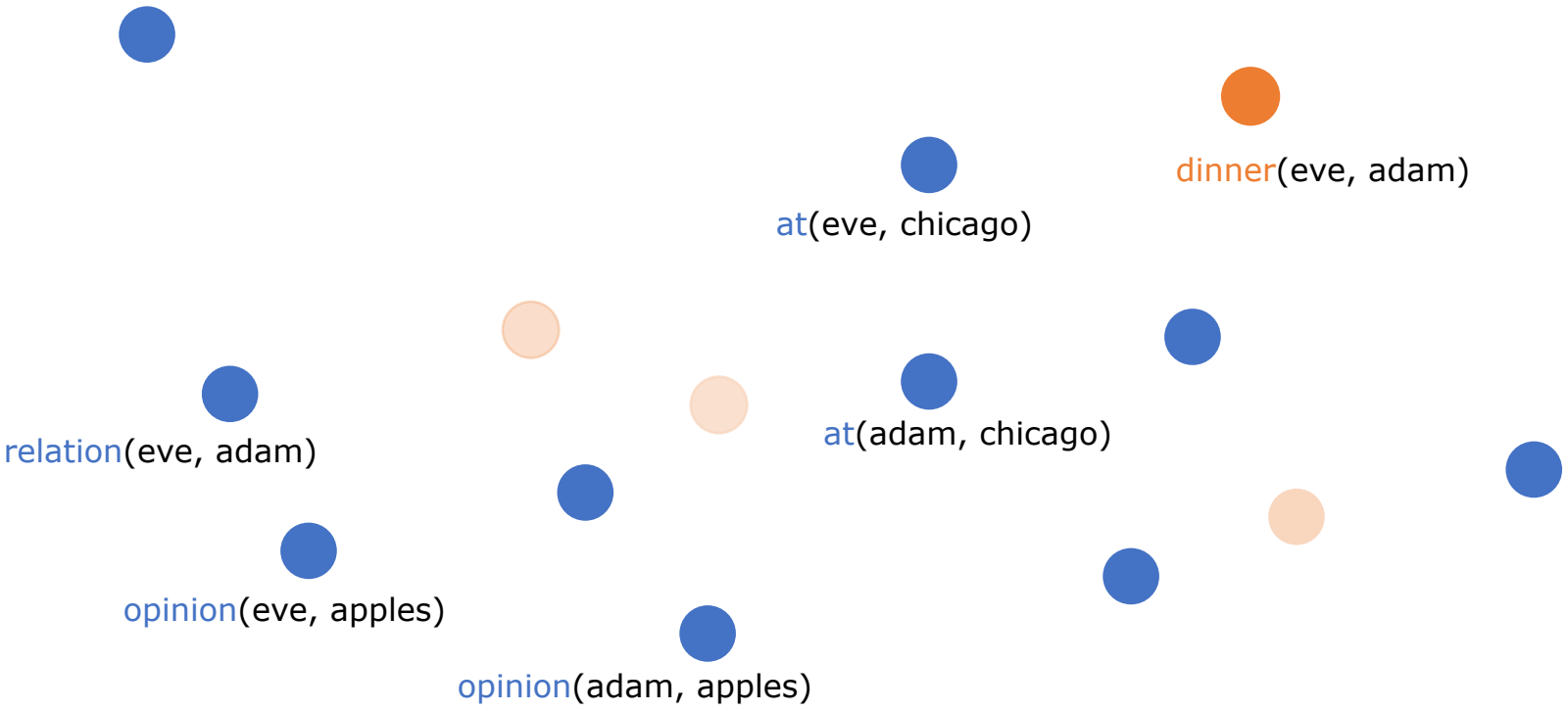


Model How a Database Changes Over Time

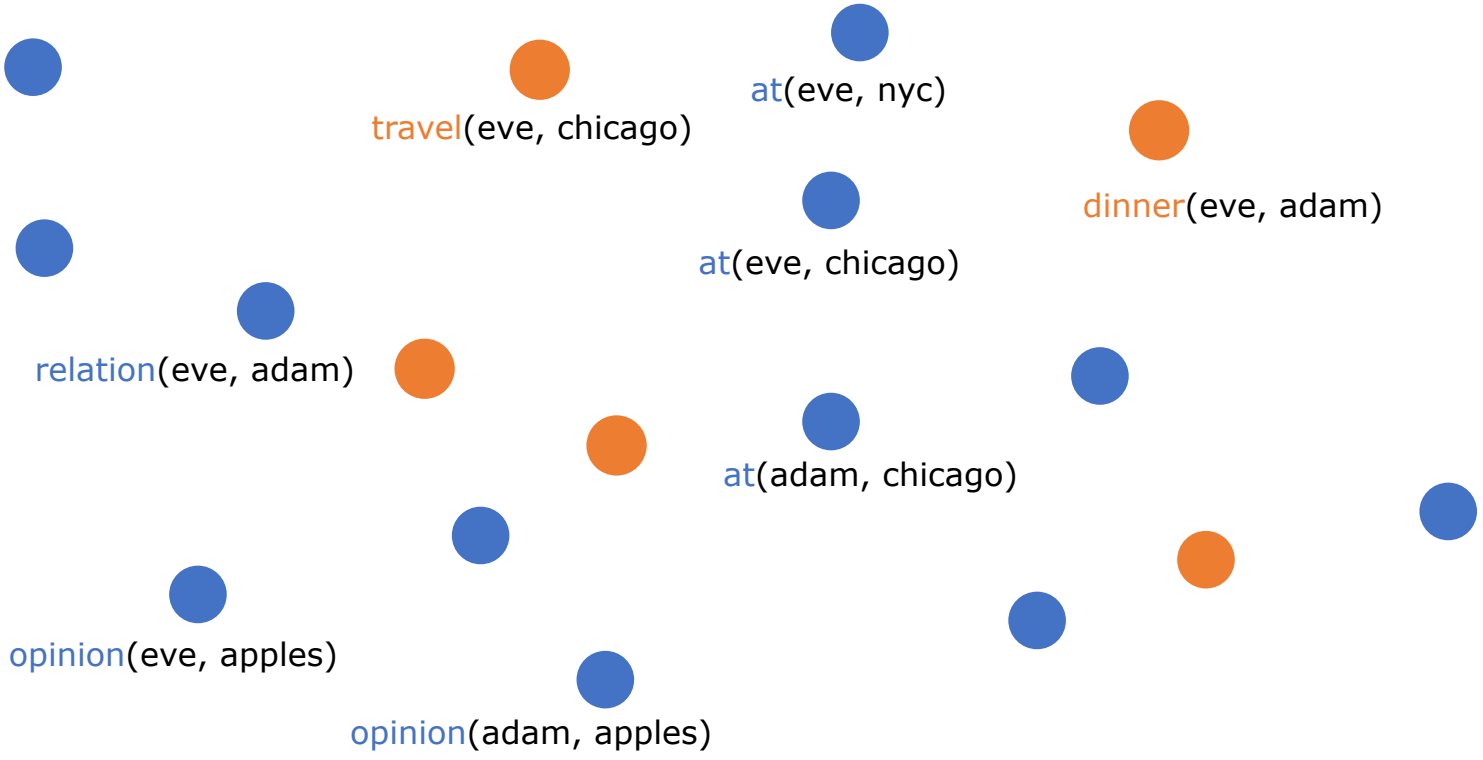
200,000 facts right now

50,000 possible events right now

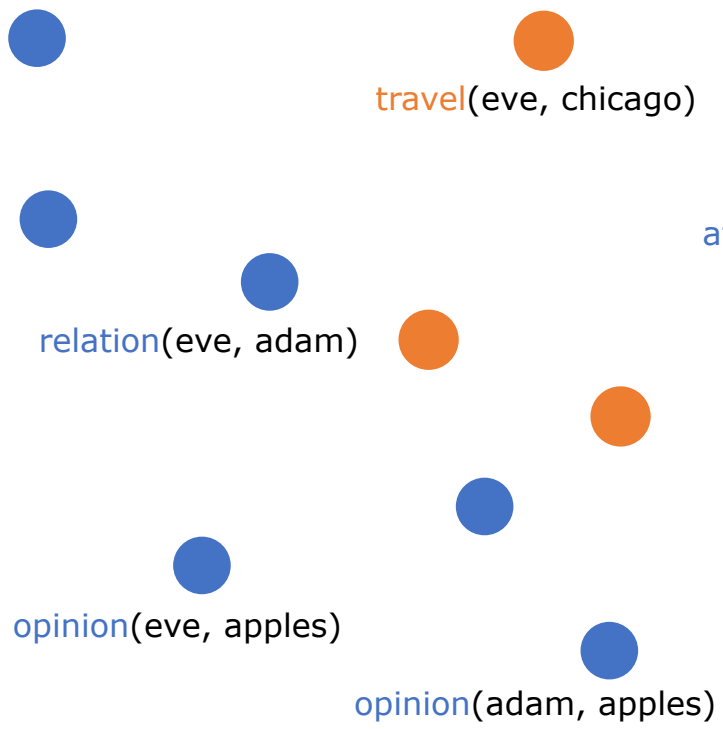
little language
to specify a generative model
of event sequences



Deductive Rules, Triggering Rules



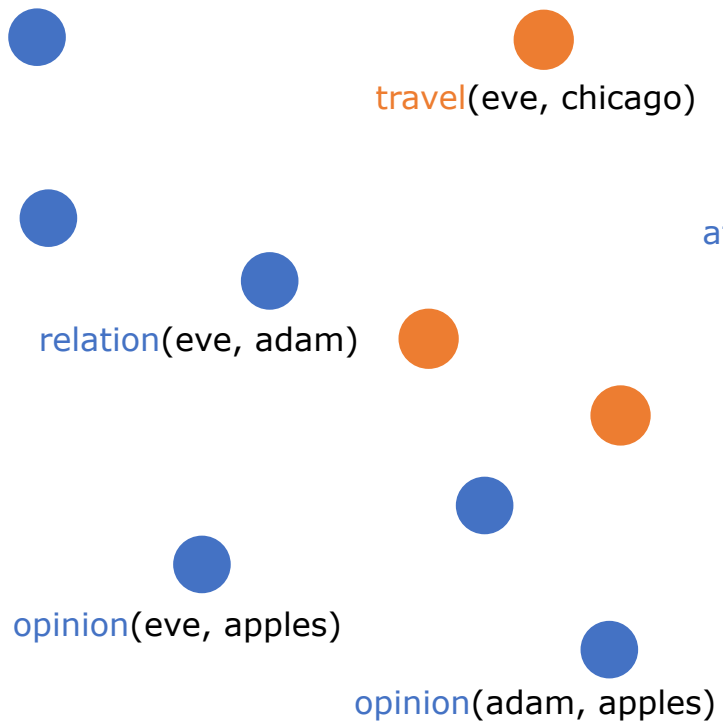
Deductive Rules, Triggering Rules




```
relation(X, Y)
    :- opinion(X, U), opinion(Y, U).

travel(X, P)
    :- relation(X, Y), at(Y, P).

!at(X, Q)
    ← travel(X, P), at(X, Q), P != Q.

at(X, P)
    ← travel(X, P).

dinner(X, Y)
    :- relation(X, Y), at(X, P), at(Y, P).

relation(X, Y)
    ← dinner(X, Y).
```

Deductive Rules, Triggering Rules

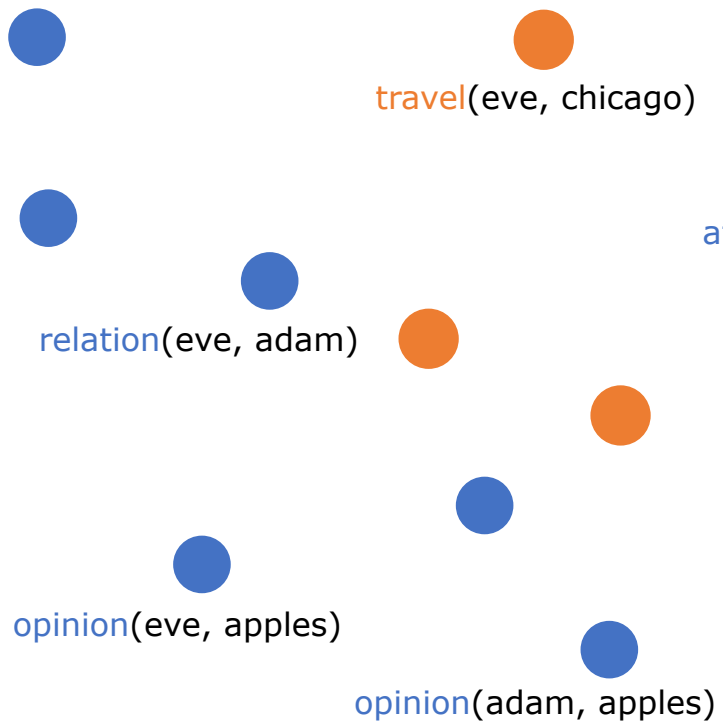
logic!




```
relation(X, Y)
    :- opinion(X, U), opinion(Y, U).

travel(X, P)
    :- relation(X, Y), at(Y, P).

!at(X, Q)
    ← travel(X, P), at(X, Q), P != Q.

at(X, P)
    ← travel(X, P).

dinner(X, Y)
    :- relation(X, Y), at(X, P), at(Y, P).

relation(X, Y)
    ← dinner(X, Y).
```

Deductive Rules, Triggering Rules

which facts are in the database



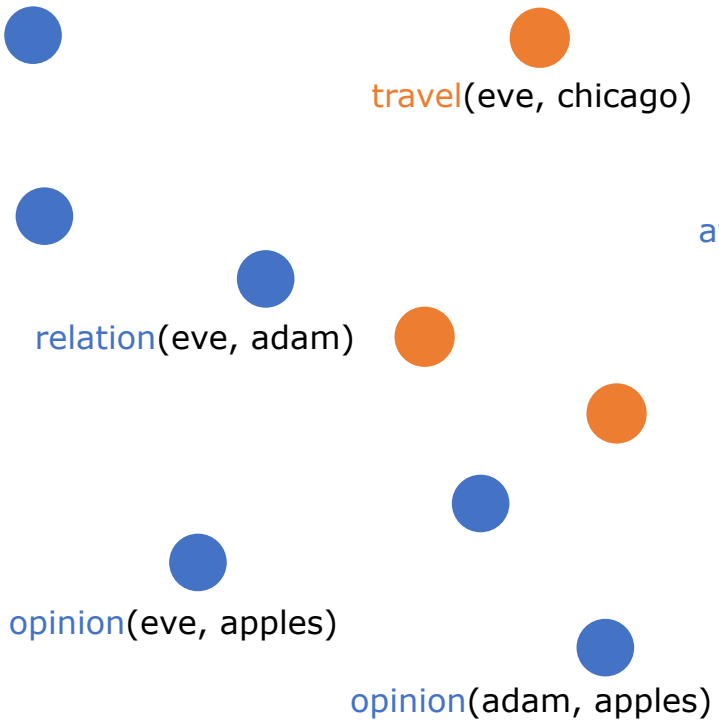
logic!

```
relation(X, Y)
    :- opinion(X, U), opinion(Y, U).

travel(X, P)
    :- relation(X, Y), at(Y, P).

!at(X, Q)
    ← travel(X, P), at(X, Q), P != Q.

at(X, P)
    ← travel(X, P).

dinner(X, Y)
    :- relation(X, Y), at(X, P), at(Y, P).

relation(X, Y)
    ← dinner(X, Y).
```

Deductive Rules, Triggering Rules

which facts are in the database
define a trainable neural architecture

logic!

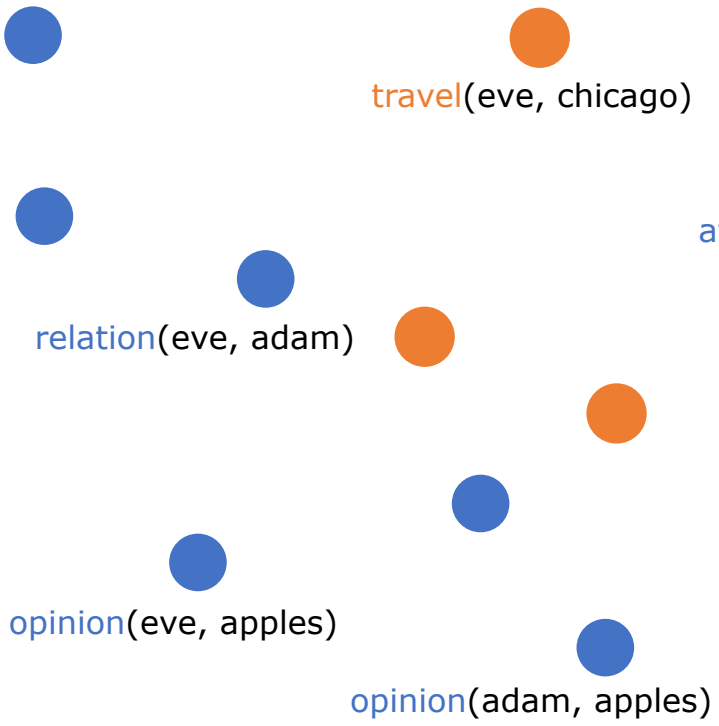



```
relation(X, Y)
    :- opinion(X, U), opinion(Y, U).

travel(X, P)
    :- relation(X, Y), at(Y, P).

!at(X, Q)
    ← travel(X, P), at(X, Q), P != Q.

at(X, P)
    ← travel(X, P).

dinner(X, Y)
    :- relation(X, Y), at(X, P), at(Y, P).

relation(X, Y)
    ← dinner(X, Y).
```

Deductive Rules, Triggering Rules

which facts are in the database
define a trainable neural architecture
that computes embeddings of the facts

logic!

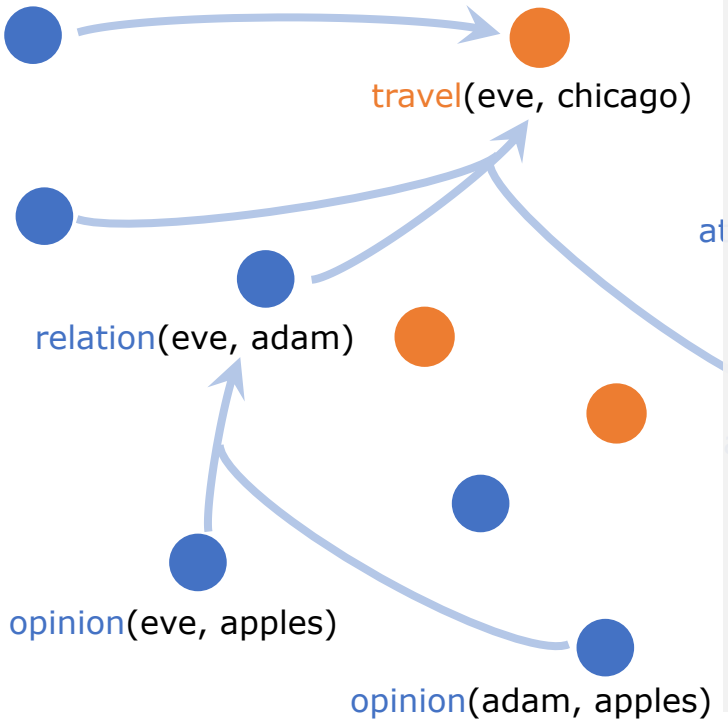



```
relation(X, Y)
    :- opinion(X, U), opinion(Y, U).

travel(X, P)
    :- relation(X, Y), at(Y, P).

!at(X, Q)
    ← travel(X, P), at(X, Q), P != Q.

at(X, P)
    ← travel(X, P).

dinner(X, Y)
    :- relation(X, Y), at(X, P), at(Y, P).

relation(X, Y)
    ← dinner(X, Y).
```

Deductive Rules, Triggering Rules

which facts are in the database
define a trainable neural architecture
that computes embeddings of the facts

logic!

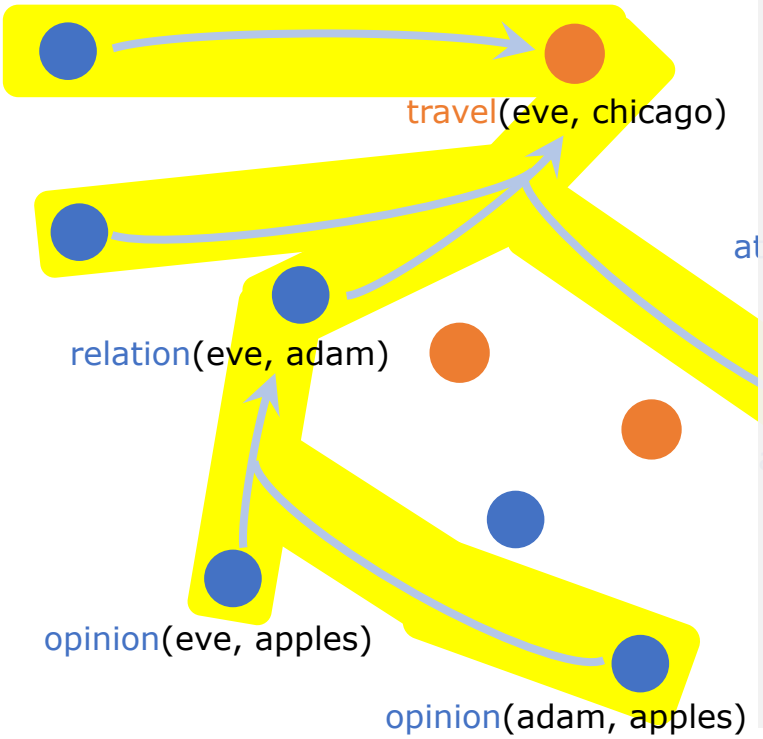



```
relation(X, Y)
    :- opinion(X, U), opinion(Y, U).

travel(X, P)
    :- relation(X, Y), at(Y, P).

!at(X, Q)
    ← travel(X, P), at(X, Q), P != Q.

at(X, P)
    ← travel(X, P).

dinner(X, Y)
    :- relation(X, Y), at(X, P), at(Y, P).

relation(X, Y)
    ← dinner(X, Y).
```

Deductive Rules, Triggering Rules

which facts are in the database
define a trainable neural architecture
that computes embeddings of the facts

logic!

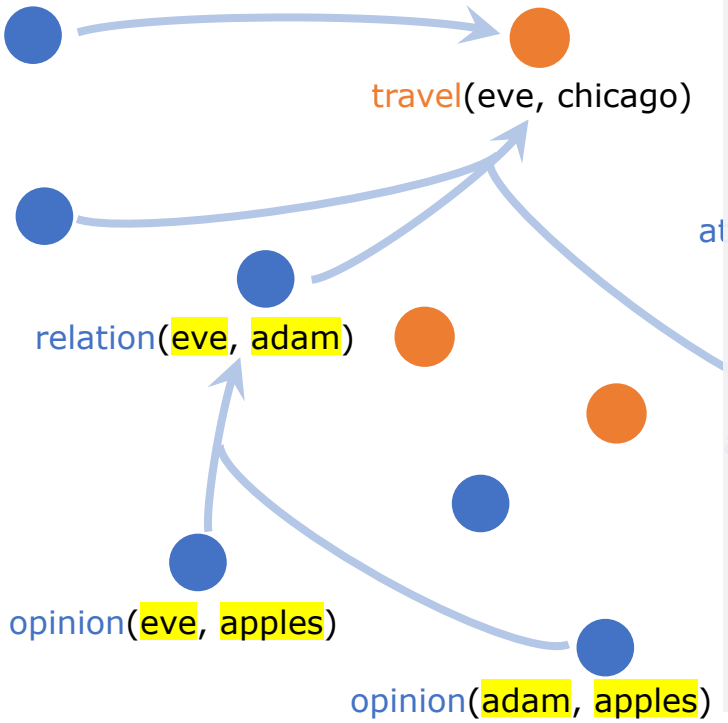



```
relation(X, Y)
    :- opinion(X, U), opinion(Y, U).

travel(X, P)
    :- relation(X, Y), at(Y, P).

!at(X, Q)
    ← travel(X, P), at(X, Q), P != Q.

at(X, P)
    ← travel(X, P).

dinner(X, Y)
    :- relation(X, Y), at(X, P), at(Y, P).

relation(X, Y)
    ← dinner(X, Y).
```

Deductive Rules, Triggering Rules

which facts are in the database
define a trainable neural architecture
that computes embeddings of the facts

logic!




```
relation(X, Y)
    :- opinion(X, U), opinion(Y, U).

travel(X, P)
    :- relation(X, Y), at(Y, P).

!at(X, Q)
    ← travel(X, P), at(X, Q), P != Q.

at(X, P)
    ← travel(X, P).

dinner(X, Y)
    :- relation(X, Y), at(X, P), at(Y, P).

relation(X, Y)
    ← dinner(X, Y).
```

Datalog ➔ Neural Datalog Through Time

Datalog ➔ Neural Datalog Through Time

deductive rule

new fact :- old fact $_1$, old fact $_2$, …

Datalog ➔ Neural Datalog Through Time

deductive rule

add to database

new fact :- old fact $_1$, old fact $_2$, …

Datalog ➔ Neural Datalog Through Time

deductive rule

add to database if

new fact :- old fact $_1$, old fact $_2$, …

Datalog ➔ Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, ...

Datalog → Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, ...

likes(X, U),

Datalog ➔ Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, ...

likes(X, U), likes(Y, U)

Datalog ➔ Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, ...

:- likes(X, U), likes(Y, U)

Datalog ➔ Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, ...

compatible(X, Y) :- likes(X, U), likes(Y, U)

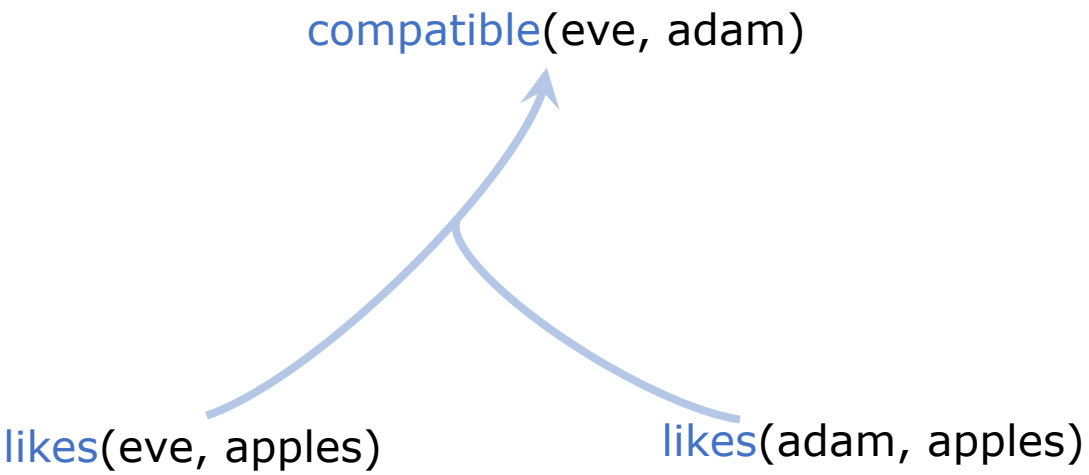
Datalog ➔ Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, ...

compatible(X, Y) :- likes(X, U), likes(Y, U)



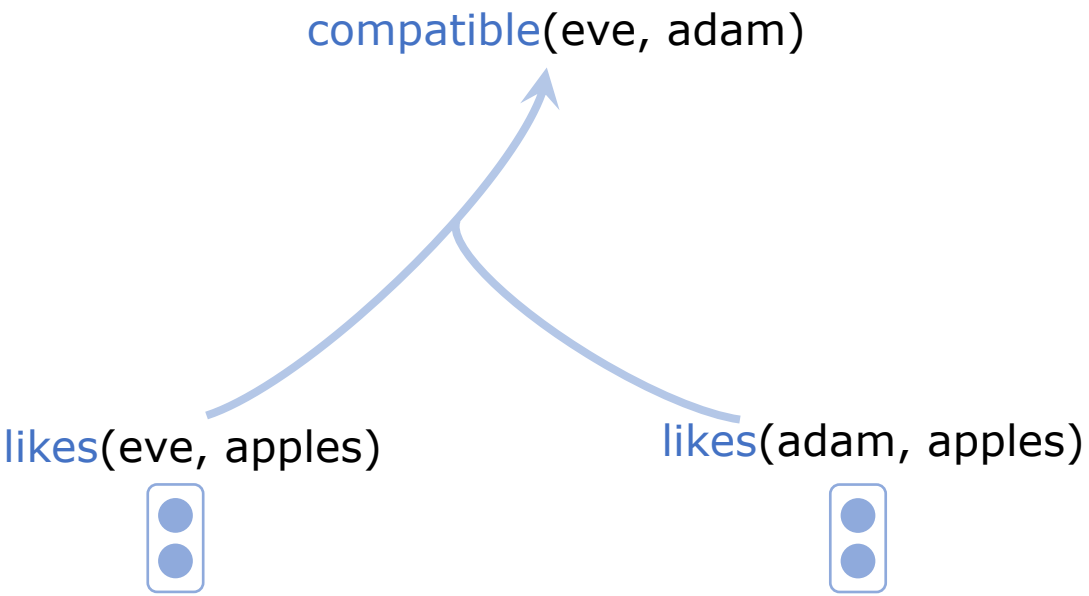
Datalog ➔ Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, ...

compatible(X, Y) :- likes(X, U), likes(Y, U)



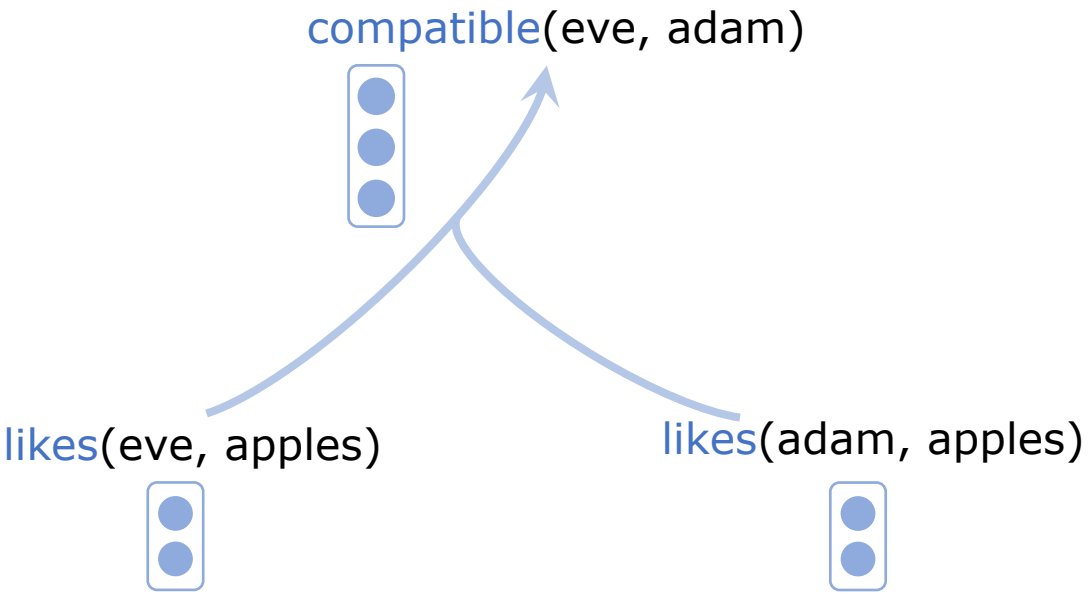
Datalog ➔ Neural Datalog Through Time

deductive rule

add to database | if | these are in database

new fact :- old fact $_1$, old fact $_2$, ...

compatible(X, Y) :- likes(X, U), likes(Y, U)



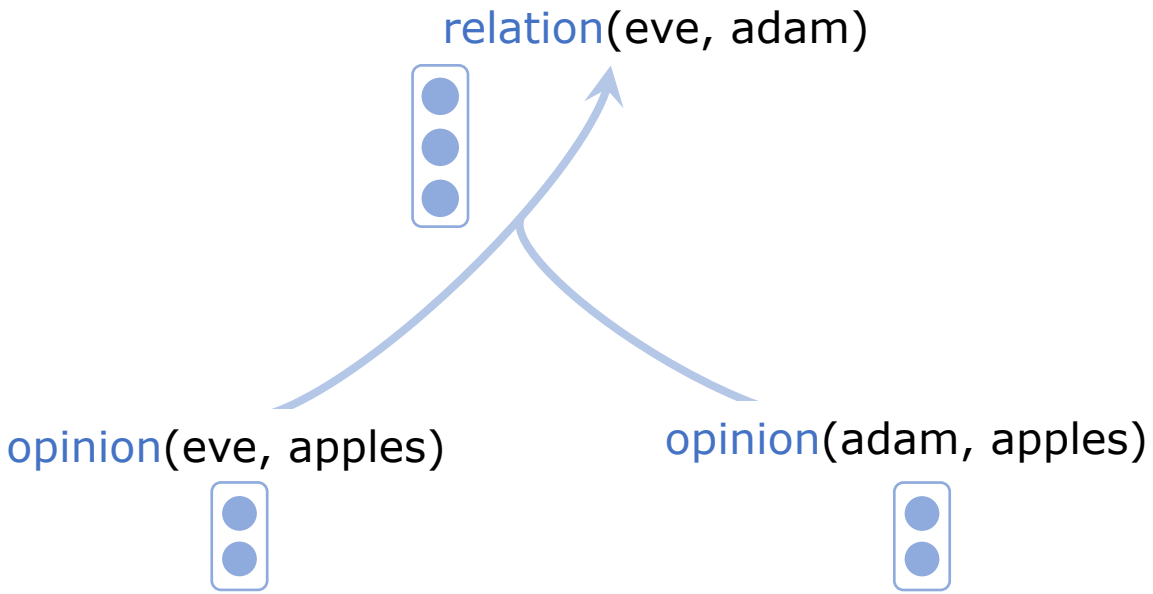
Datalog ➔ Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, ...

compatible(X, Y) :- likes(X, U), likes(Y, U)



Datalog → Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, …

triggering rule

new fact ← event, old fact $_1$, old fact $_2$, …

Datalog → Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, …

triggering rule

when

new fact ← event, old fact $_1$, old fact $_2$, …

Datalog → Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, …

triggering rule

when this happens

new fact ← event, old fact $_1$, old fact $_2$, …

Datalog → Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, …

triggering rule

when this happens while these are in database

new fact ← event, old fact $_1$, old fact $_2$, …

Datalog → Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, …

triggering rule

add to database when this happens while these are in database

new fact ← event, old fact $_1$, old fact $_2$, …

Datalog → Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, …

triggering rule

add to database when this happens while these are in database

new fact ← event, old fact $_1$, old fact $_2$, …

! old fact ← event, old fact $_1$, old fact $_2$, …

Datalog ➔ Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, …

triggering rule

add to database when this happens while these are in database

new fact ← event, old fact $_1$, old fact $_2$, …

when

! old fact ← event, old fact $_1$, old fact $_2$, …

Datalog ➔ Neural Datalog Through Time

deductive rule

add to database — if — these are in database

new fact :- old fact $_1$, old fact $_2$, …

triggering rule

add to database — when this happens — while these are in database

new fact ← event, old fact $_1$, old fact $_2$, …

when this happens

! old fact ← event, old fact $_1$, old fact $_2$, …

Datalog ➔ Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, …

triggering rule

add to database when this happens while these are in database

new fact ← event, old fact $_1$, old fact $_2$, …

when this happens while these are in database

! old fact ← event, old fact $_1$, old fact $_2$, …

Datalog ➔ Neural Datalog Through Time

deductive rule

add to database if these are in database

new fact :- old fact $_1$, old fact $_2$, …

triggering rule

add to database when this happens while these are in database

new fact ← event, old fact $_1$, old fact $_2$, …

delete when this happens while these are in database

! old fact ← event, old fact $_1$, old fact $_2$, …

Computing the Embeddings

Computing the Embeddings

relation(eve, adam)

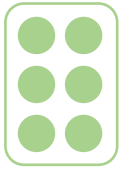
Computing the Embeddings

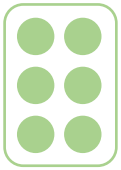
relation(X, Y) :- opinion(X, U), opinion(Y, U)

relation(eve, adam)

Computing the Embeddings

relation(X, Y) :- opinion(X, U), opinion(Y, U)

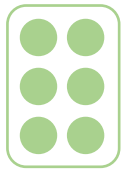


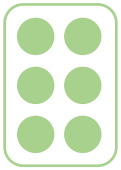


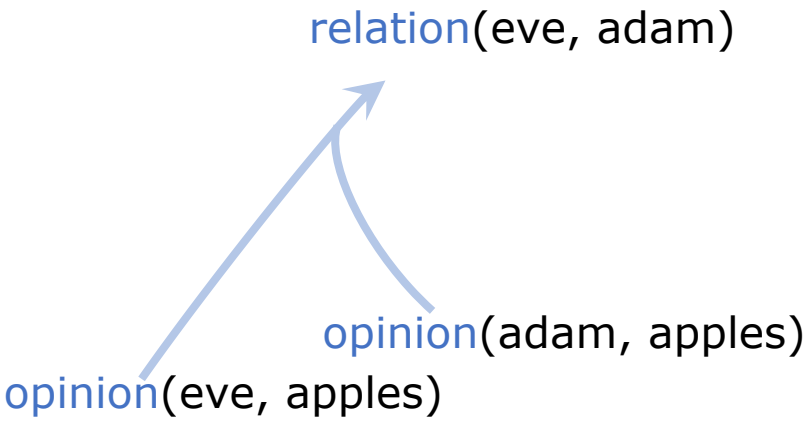
relation(eve, adam)

Computing the Embeddings

relation(X, Y) :- opinion(X, U), opinion(Y, U)

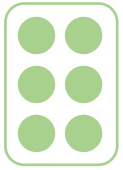


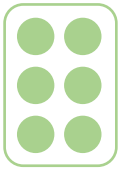


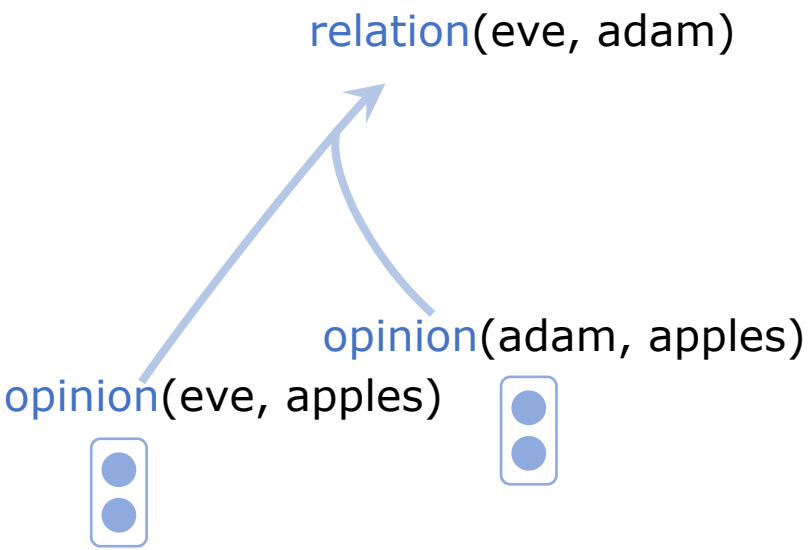


Computing the Embeddings

relation(X, Y) :- opinion(X, U), opinion(Y, U)

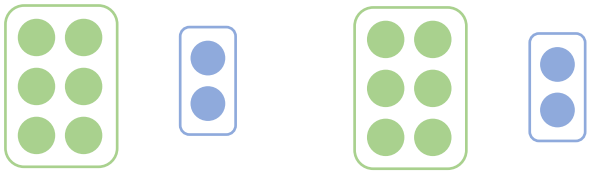


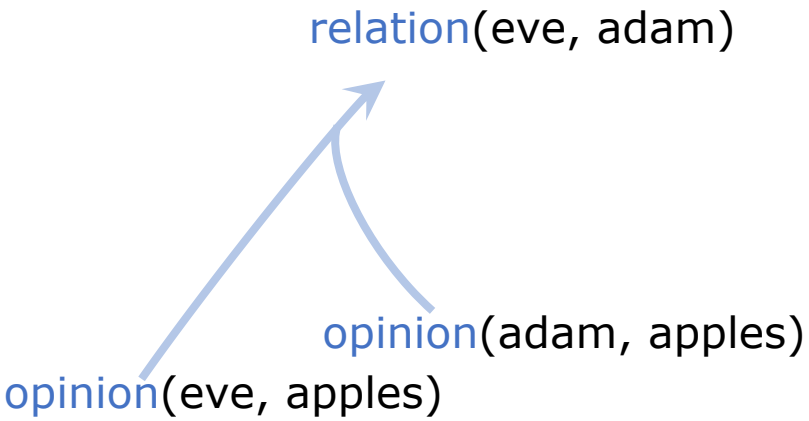




Computing the Embeddings

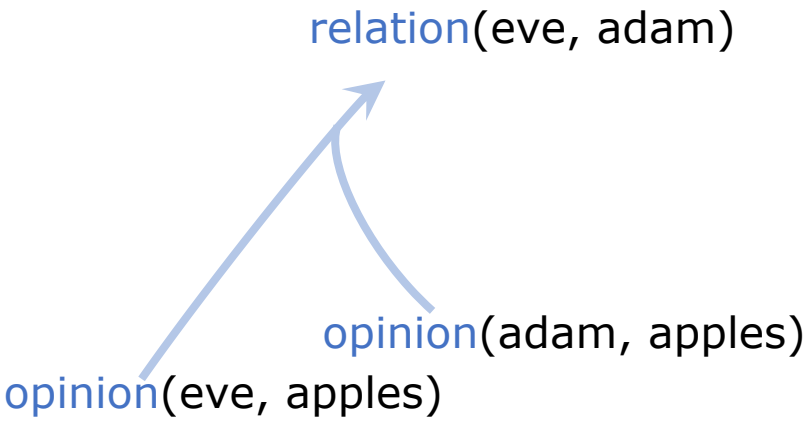
relation(X, Y) :- opinion(X, U), opinion(Y, U)





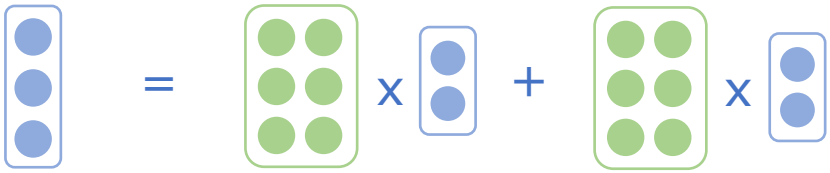
Computing the Embeddings

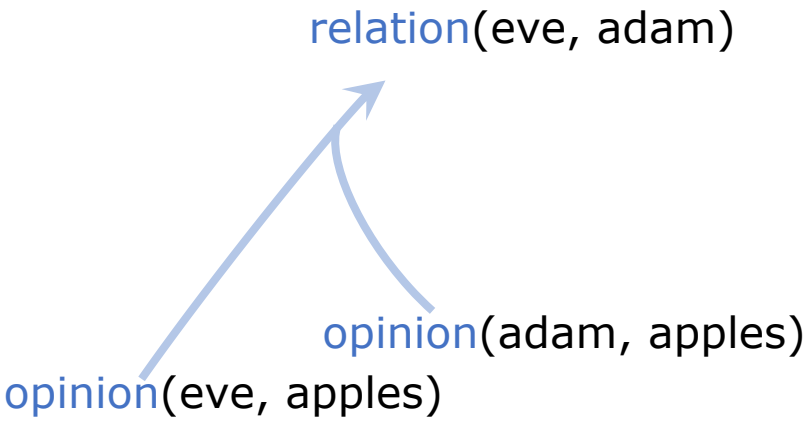
relation(X, Y) :- opinion(X, U), opinion(Y, U)



Computing the Embeddings

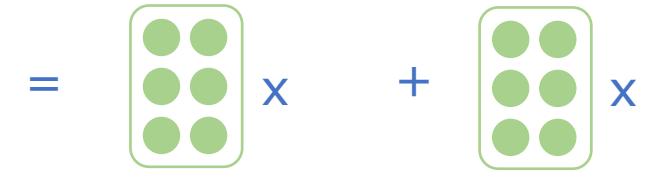
relation(X, Y) :- opinion(X, U), opinion(Y, U)

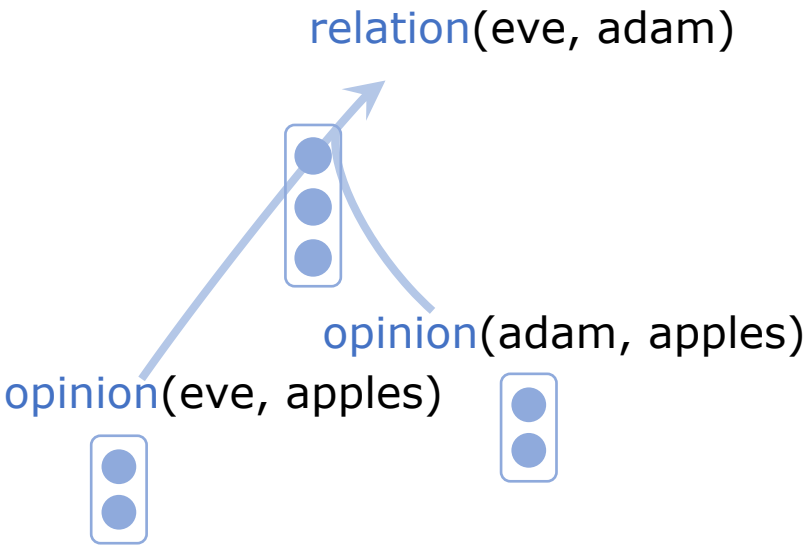




Computing the Embeddings

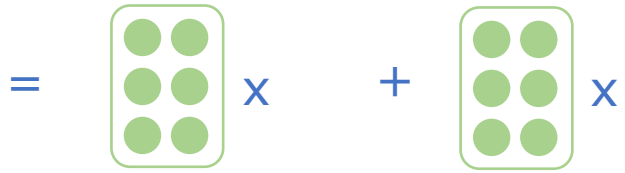
relation(X, Y) :- opinion(X, U), opinion(Y, U)

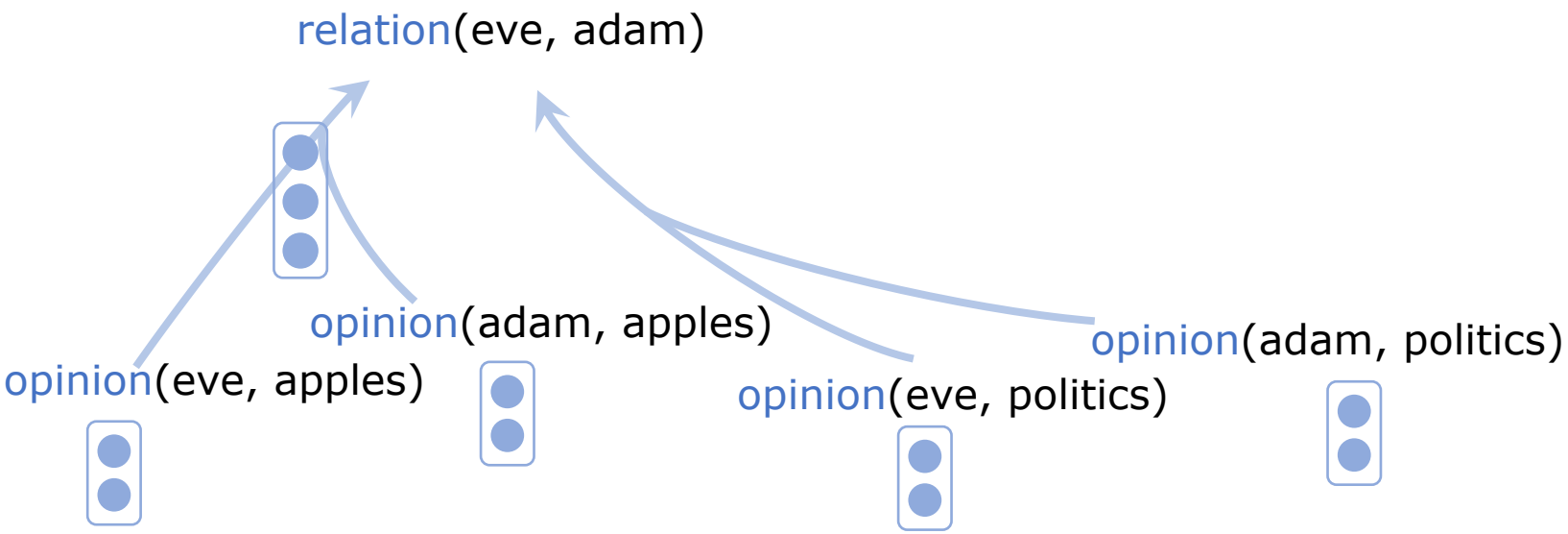




Computing the Embeddings

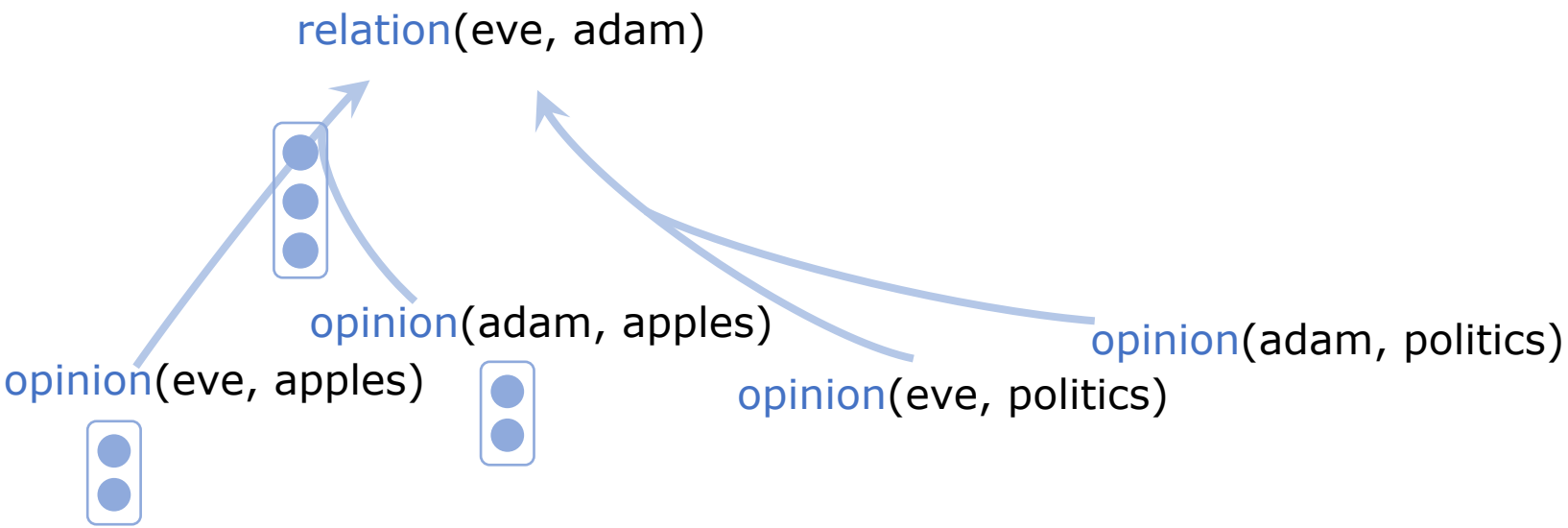
relation(X, Y) :- opinion(X, U), opinion(Y, U)





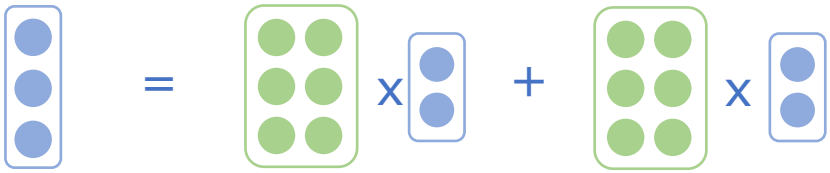
Computing the Embeddings

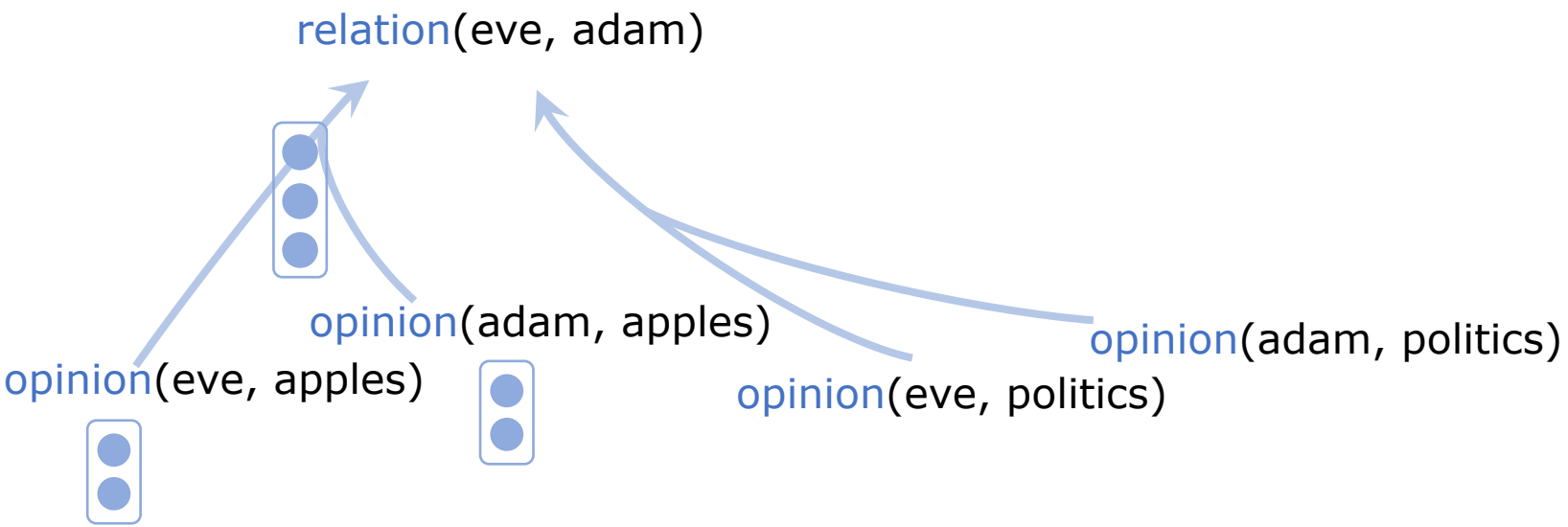
relation(X, Y) :- opinion(X, U), opinion(Y, U)



Computing the Embeddings

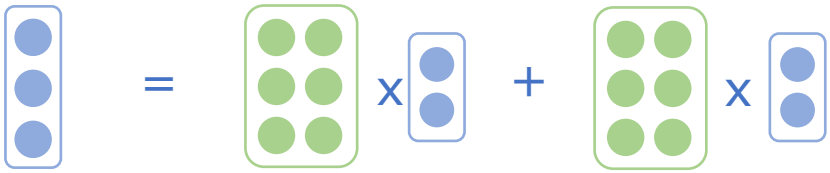
relation(X, Y) :- opinion(X, U), opinion(Y, U)





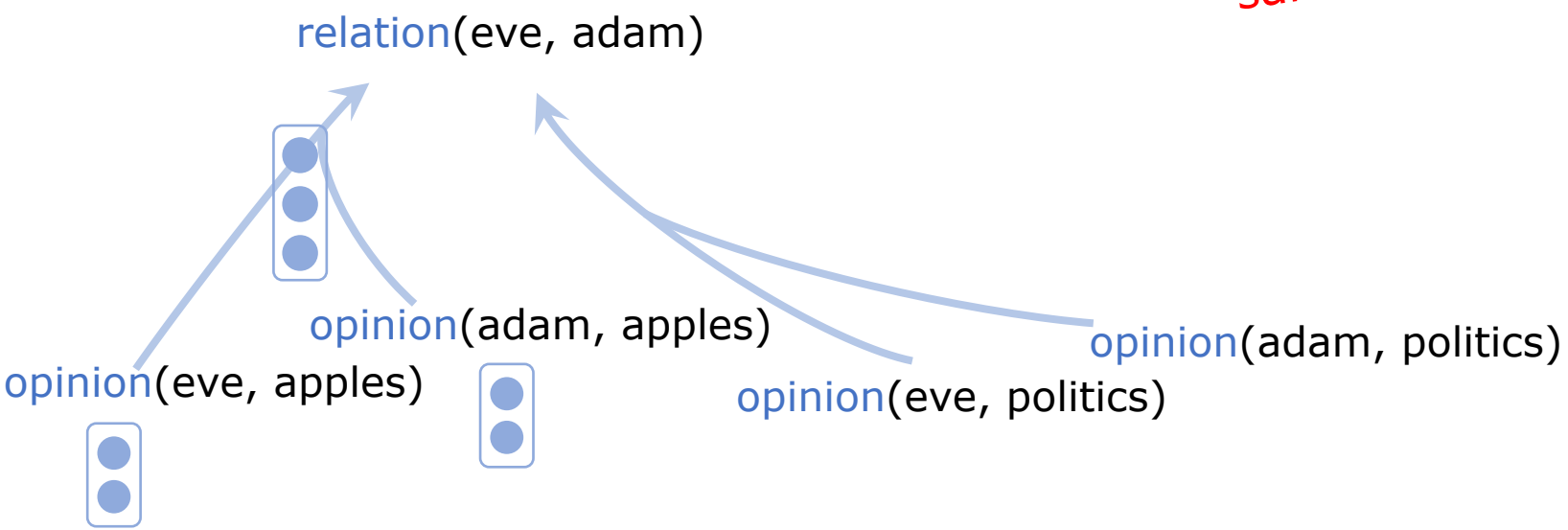
Computing the Embeddings

relation(X, Y) :- opinion(X, U), opinion(Y, U)



different inputs

same params

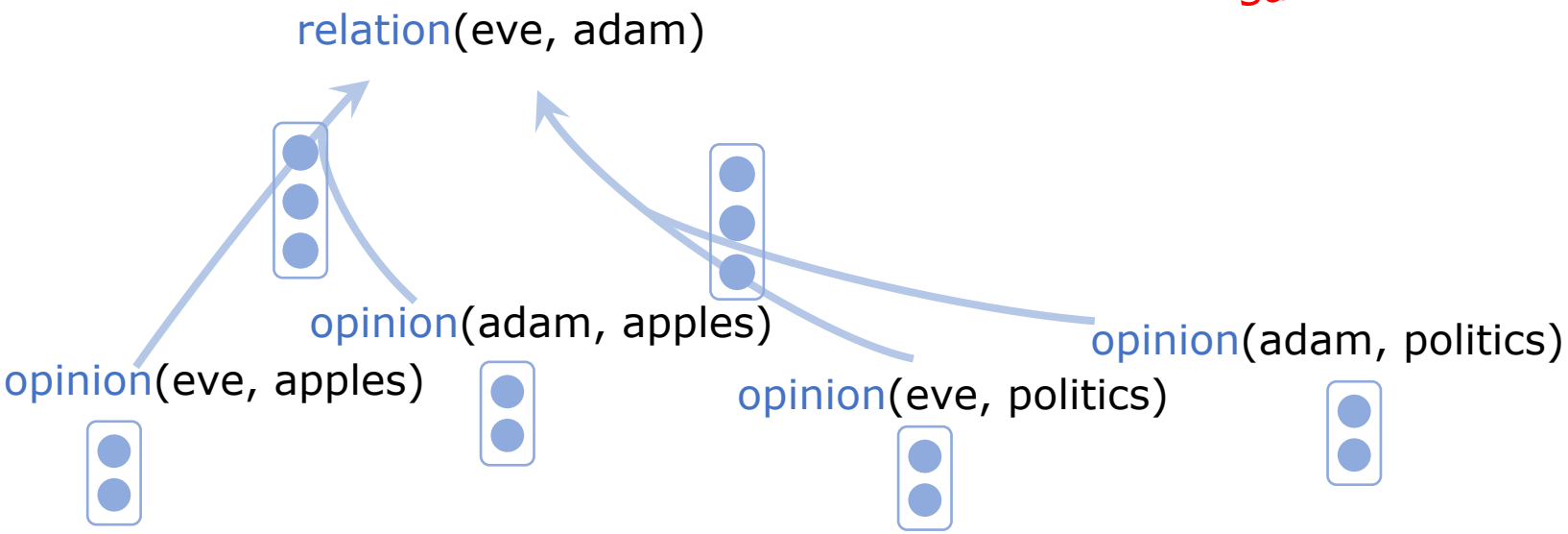


Computing the Embeddings

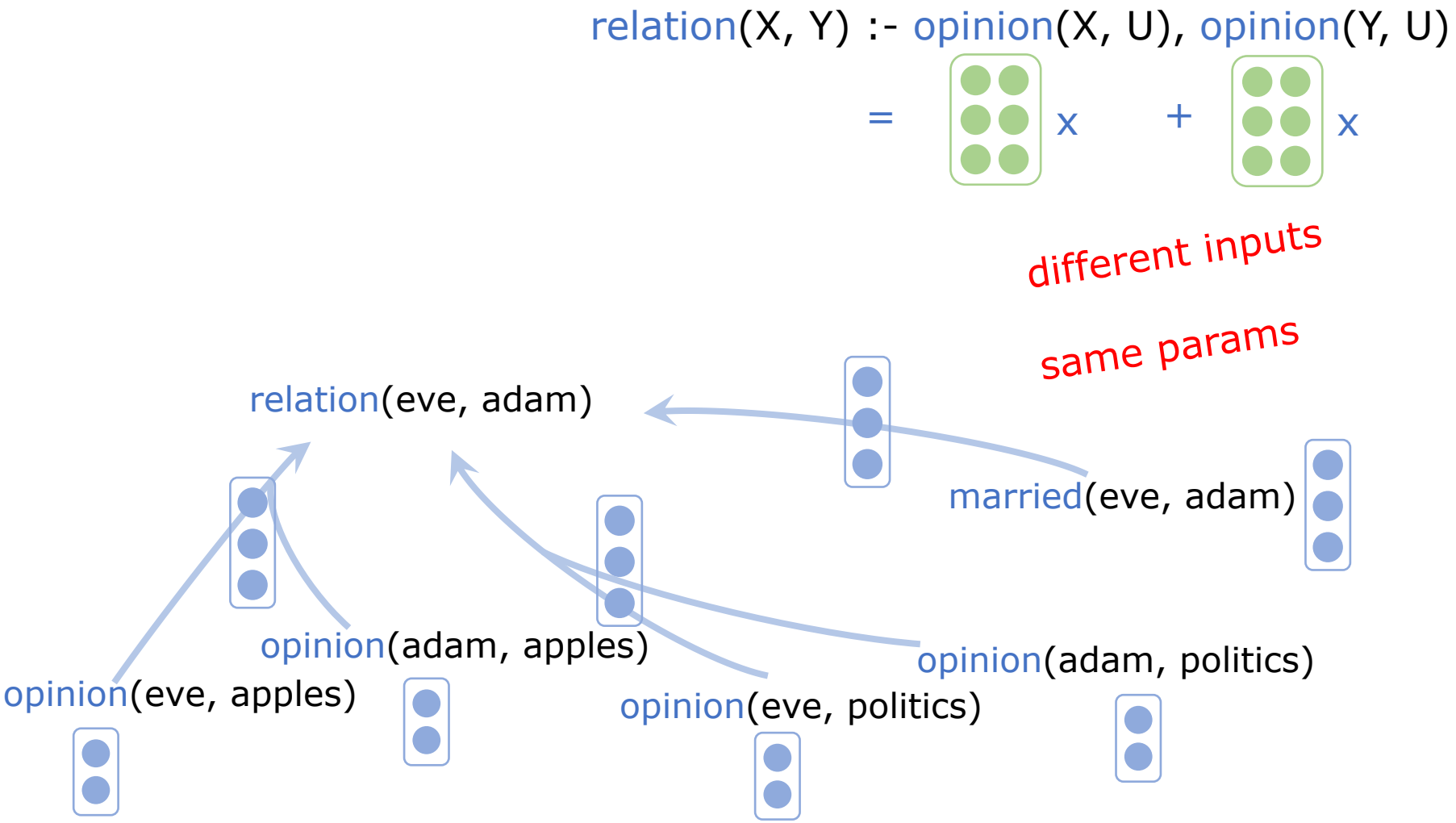
relation(X, Y) :- opinion(X, U), opinion(Y, U)

different inputs

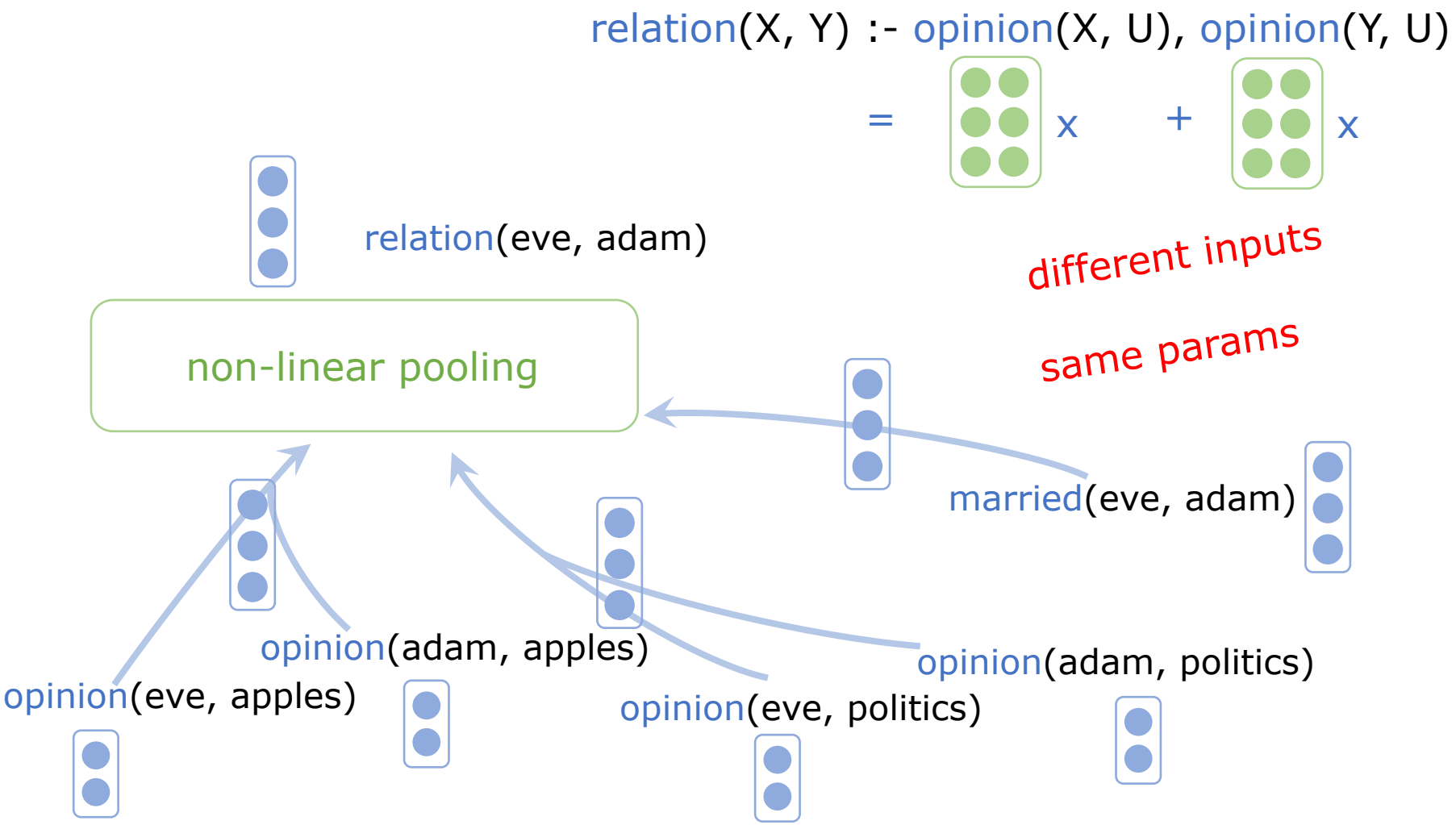
same params



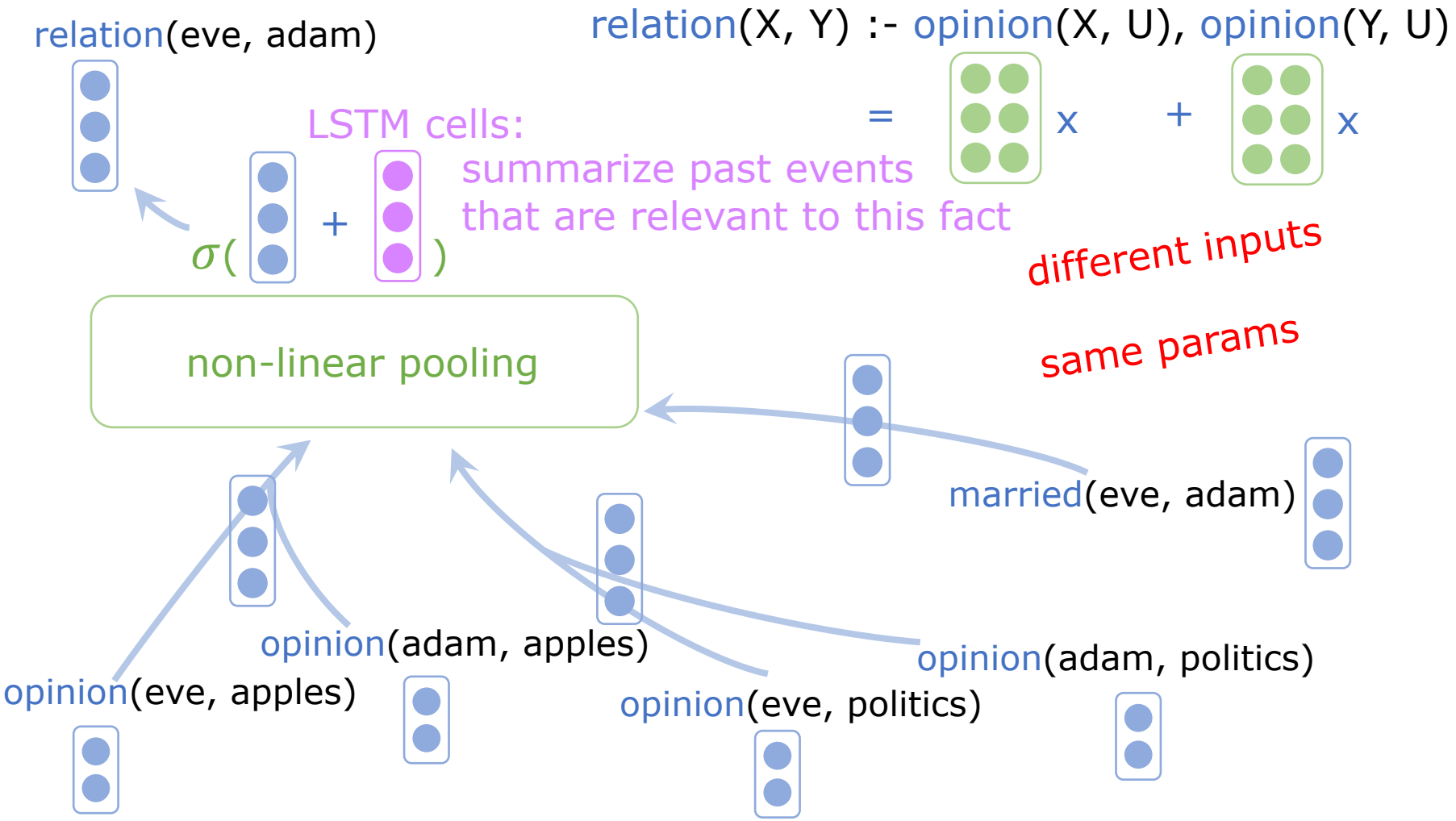
Computing the Embeddings



Computing the Embeddings



Computing the Embeddings



Computing Embeddings & Probabilities

Computing Embeddings & Probabilities

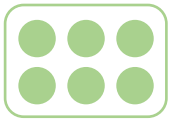
...

relation(eve, adam)

relation(eve, cain)

Computing Embeddings & Probabilities

travel(X, P) :- relation(X, Y), at(Y, P).



...

relation(eve, adam)

relation(eve, cain)

Computing Embeddings & Probabilities

travel(X, P) :- relation(X, Y), at(Y, P).

...

...

relation(eve, adam)

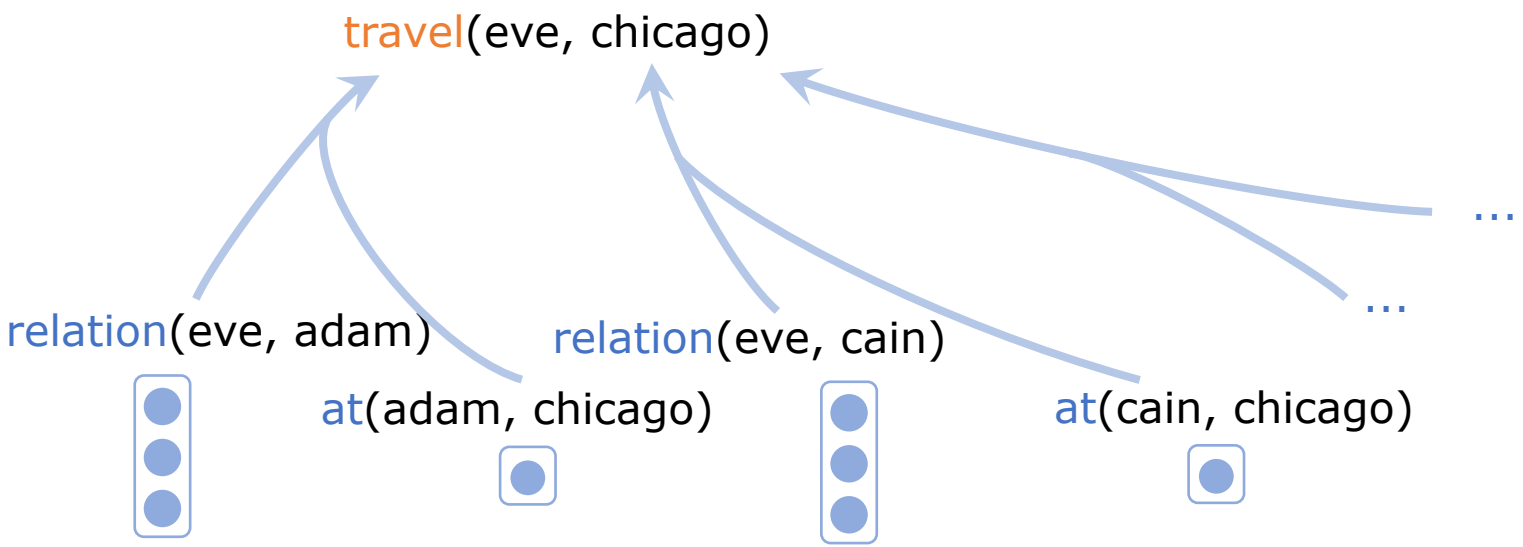
at(adam, chicago)

relation(eve, cain)

at(cain, chicago)

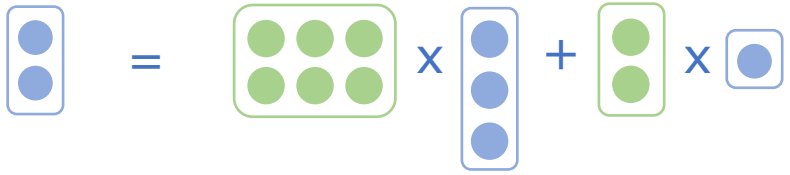
Computing Embeddings & Probabilities

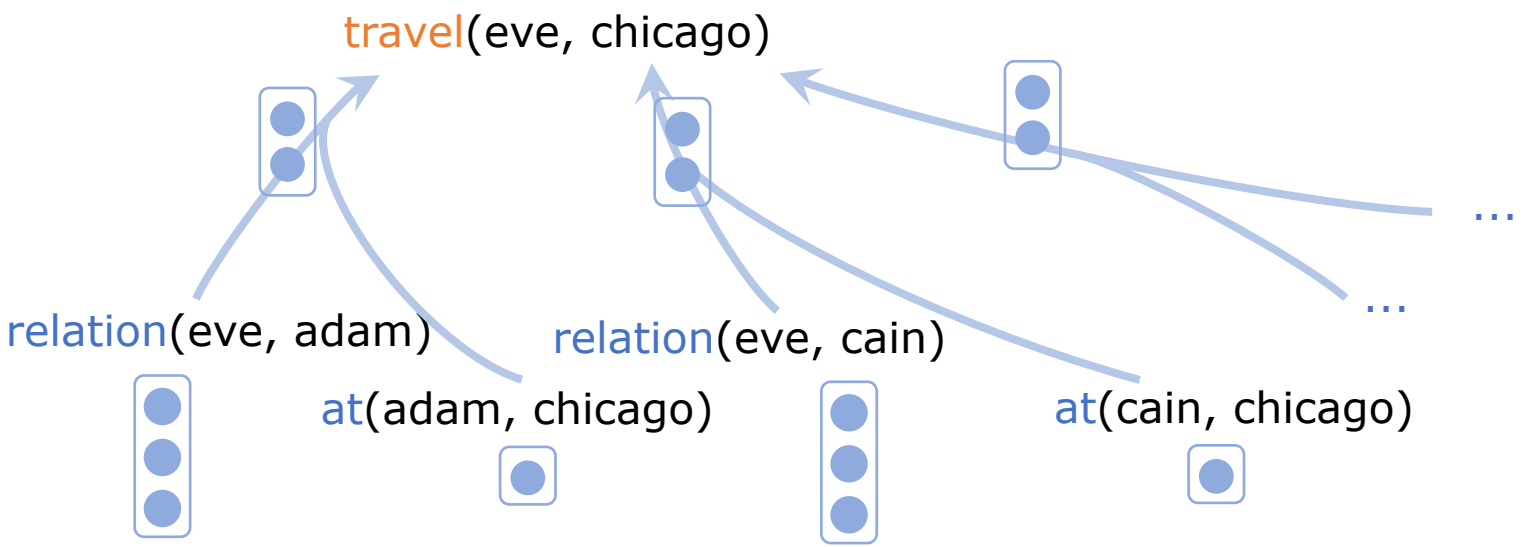
travel(X, P) :- relation(X, Y), at(Y, P).



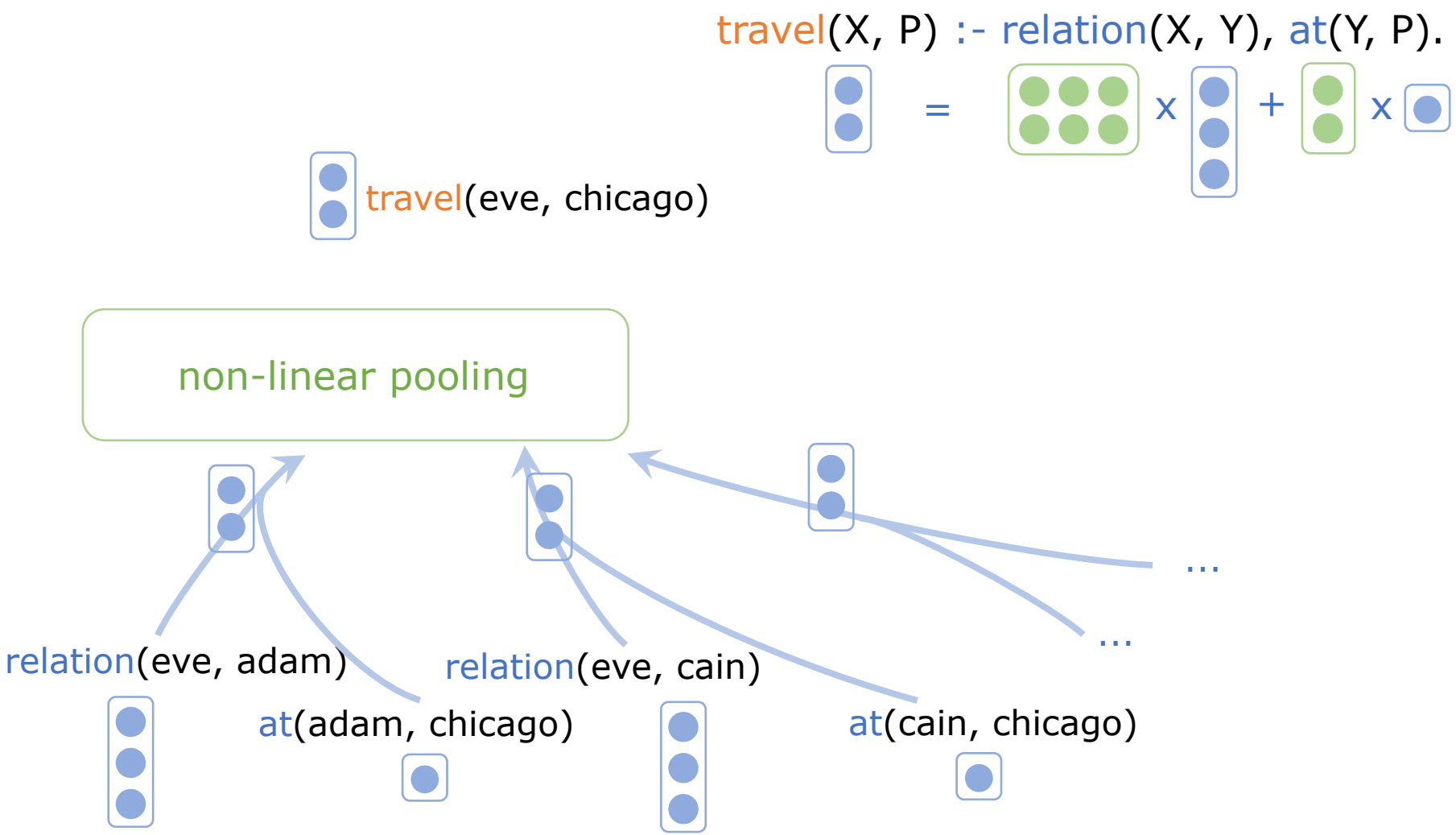
Computing Embeddings & Probabilities

travel(X, P) :- relation(X, Y), at(Y, P).

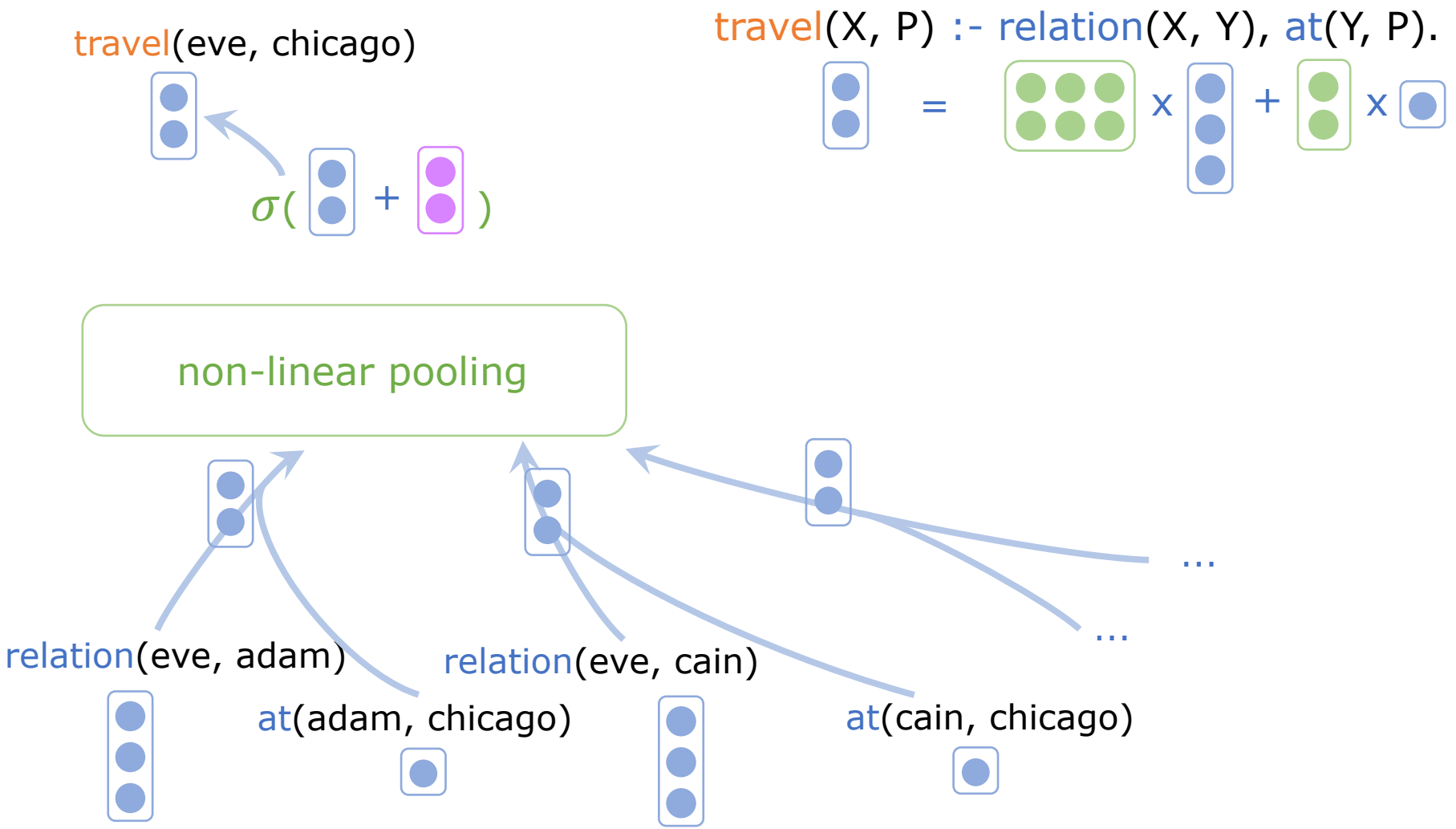




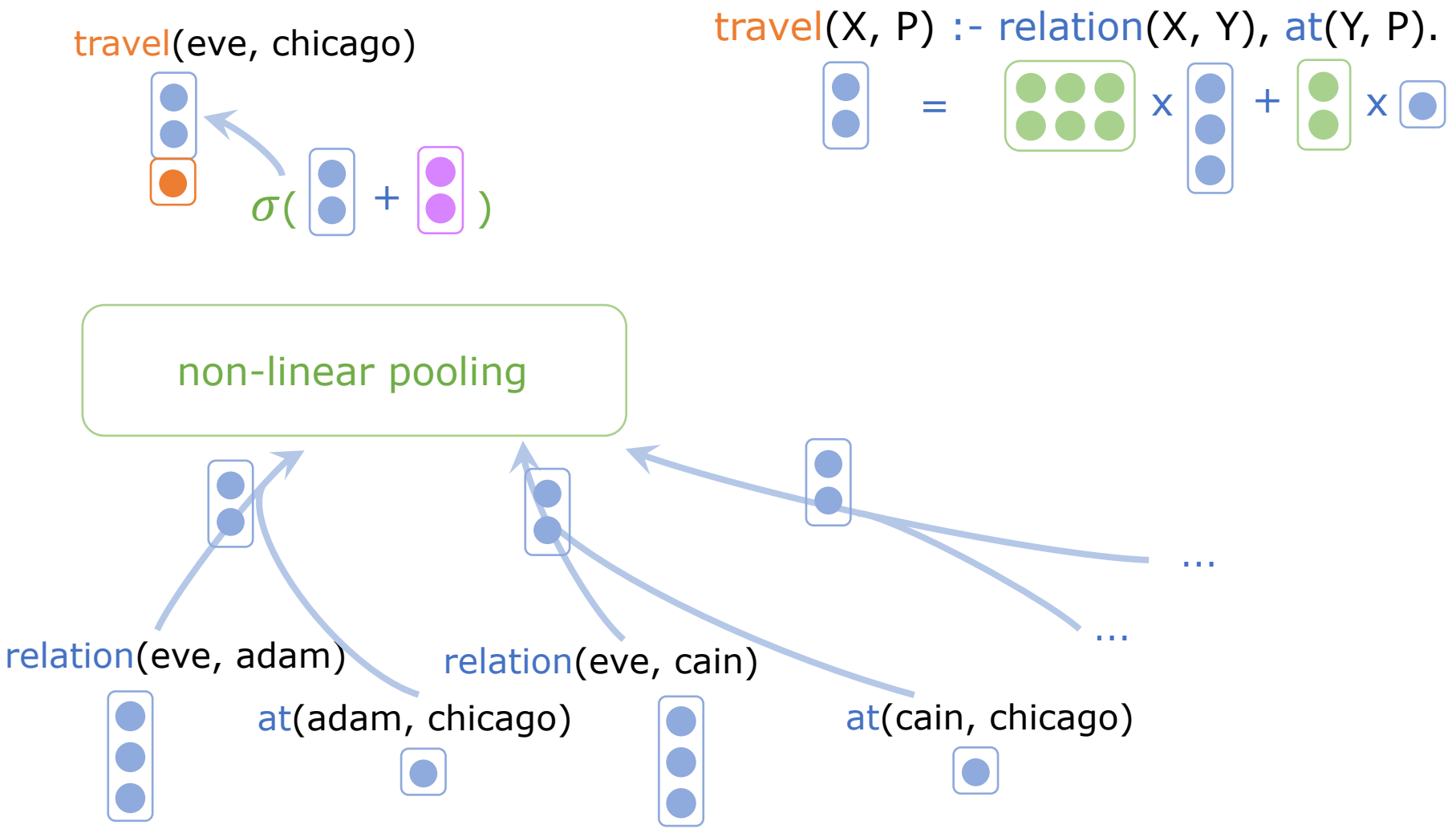
Computing Embeddings & Probabilities



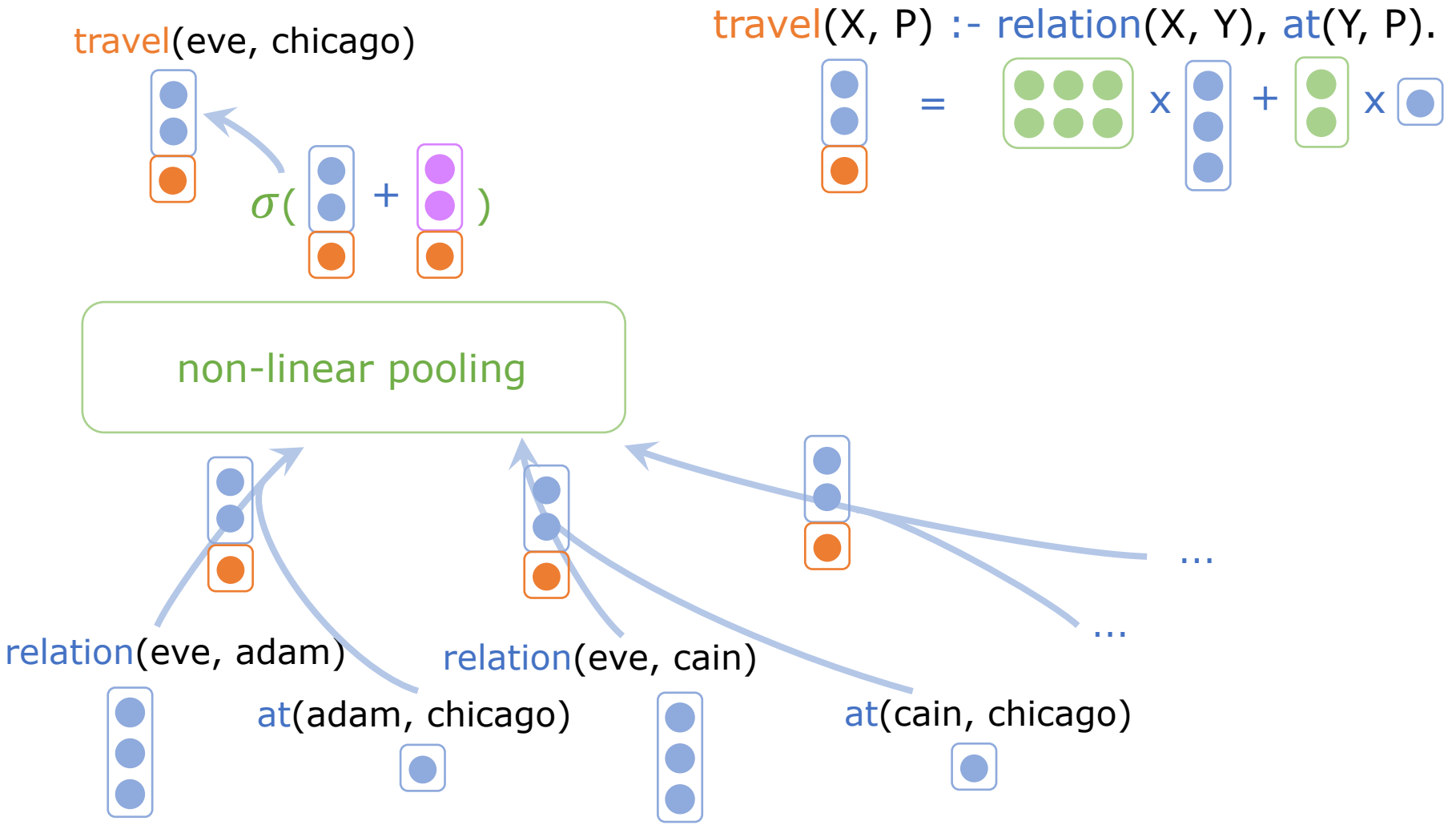
Computing Embeddings & Probabilities



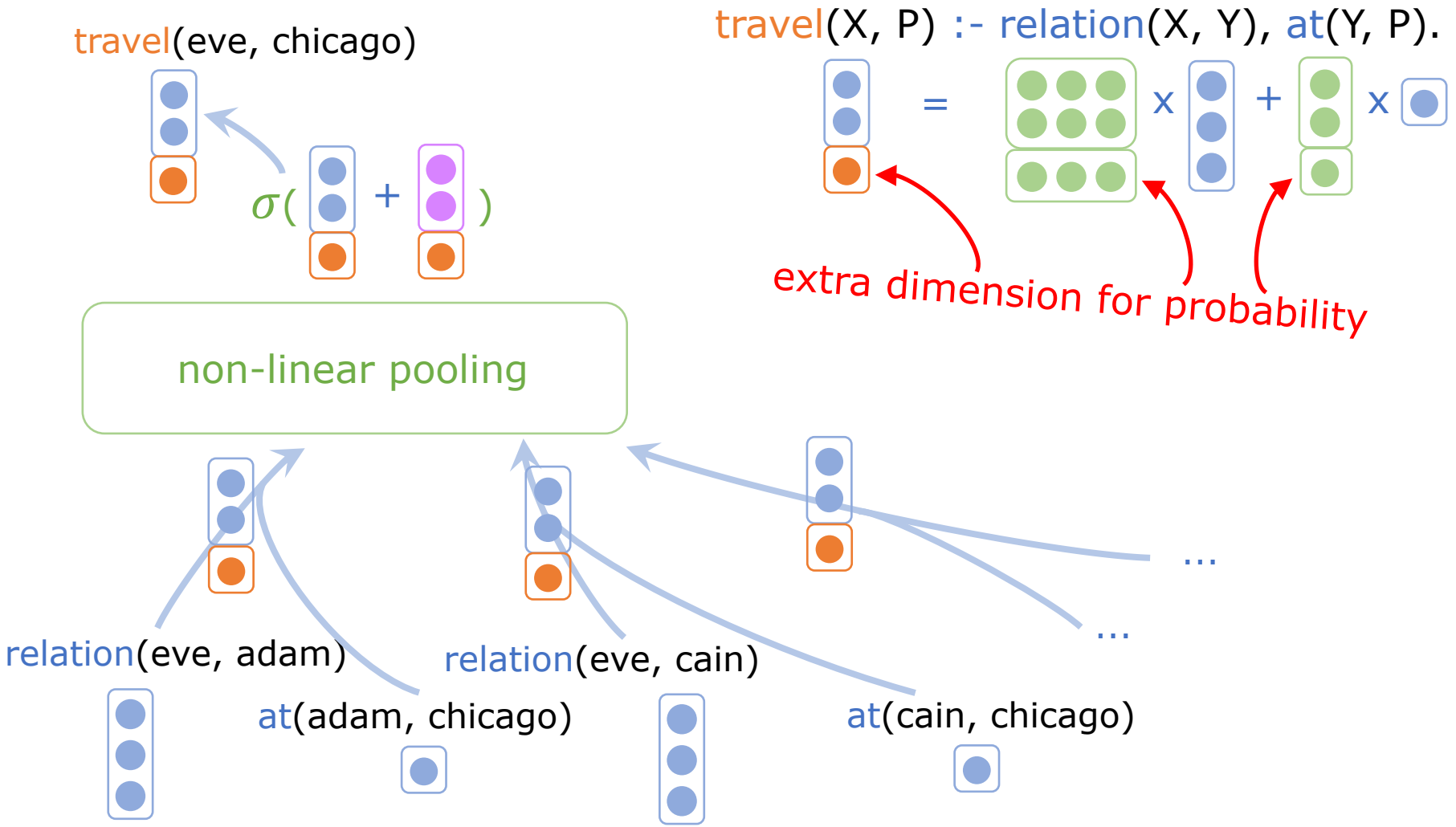
Computing Embeddings & Probabilities



Computing Embeddings & Probabilities

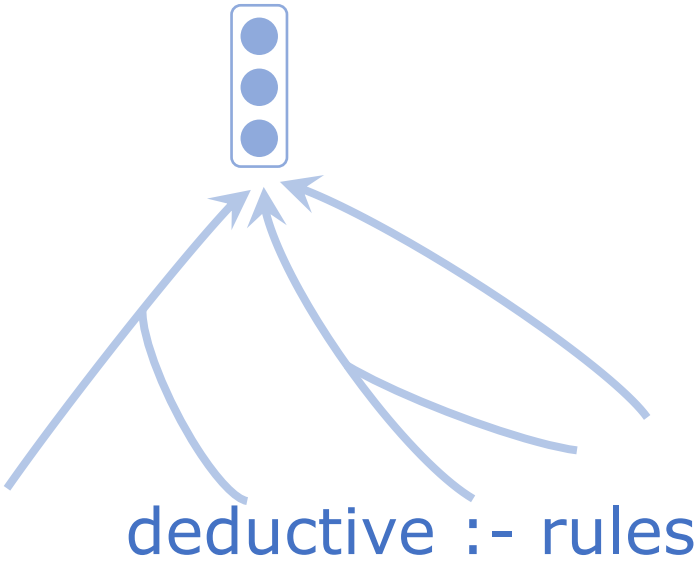


Computing Embeddings & Probabilities

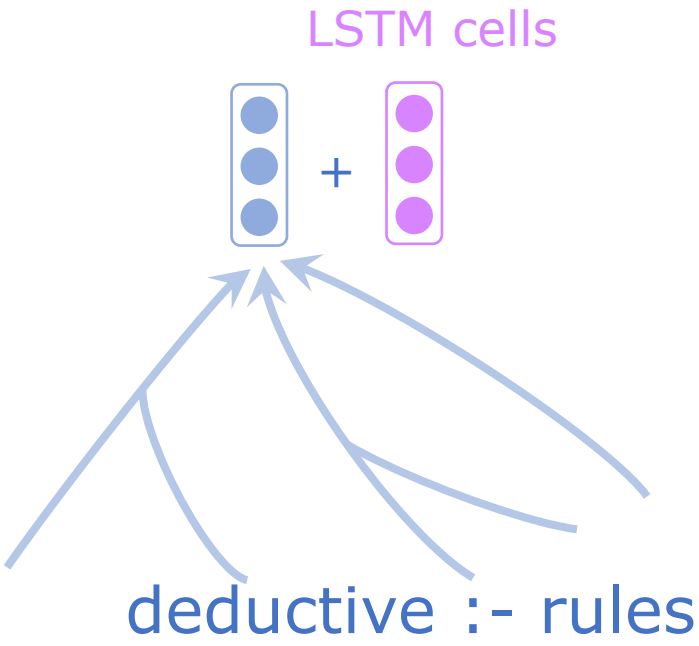


Rules ➔ Deep Recurrent Neural Net

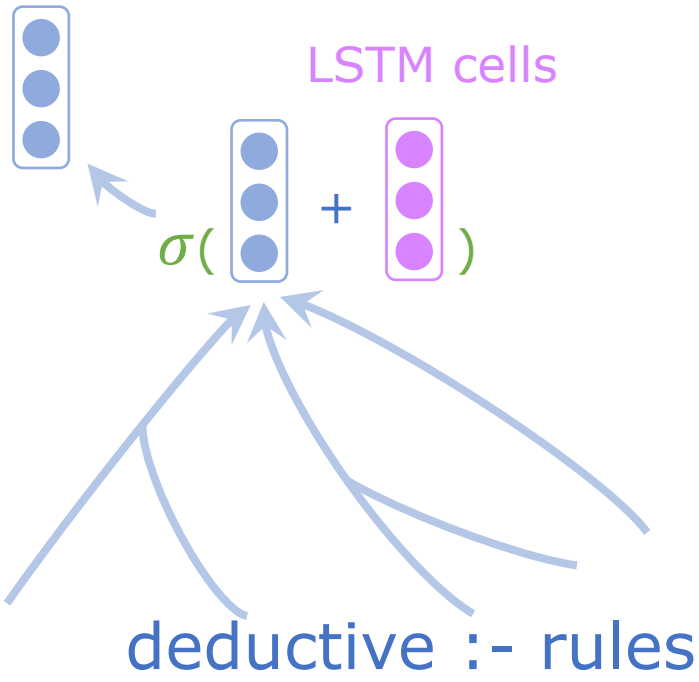
Rules ➔ Deep Recurrent Neural Net



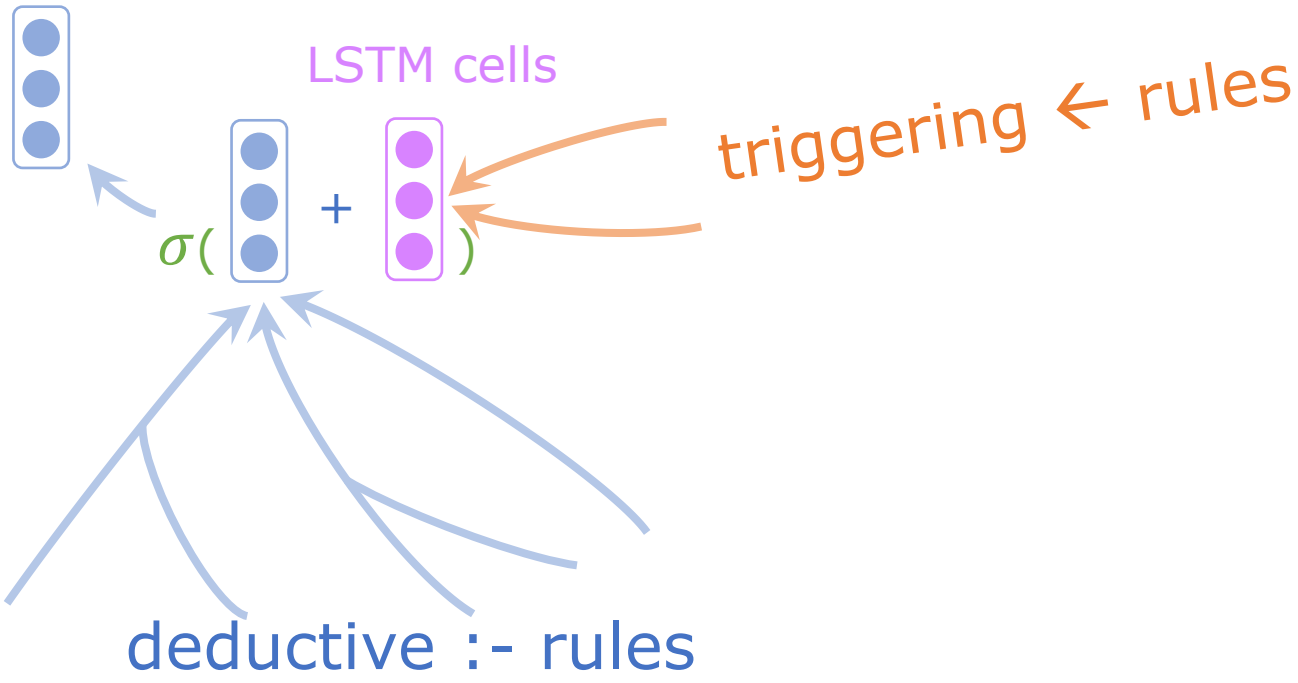
Rules ➔ Deep Recurrent Neural Net



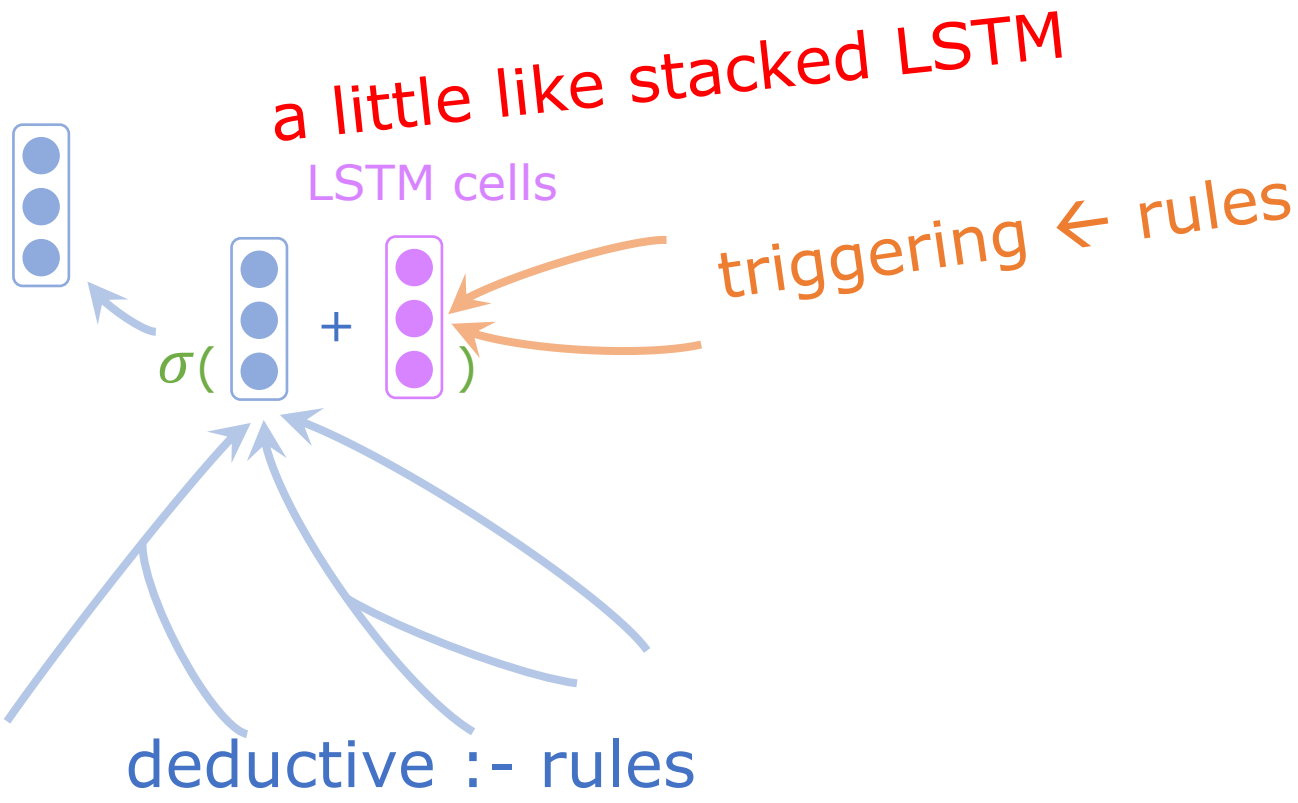
Rules ➔ Deep Recurrent Neural Net



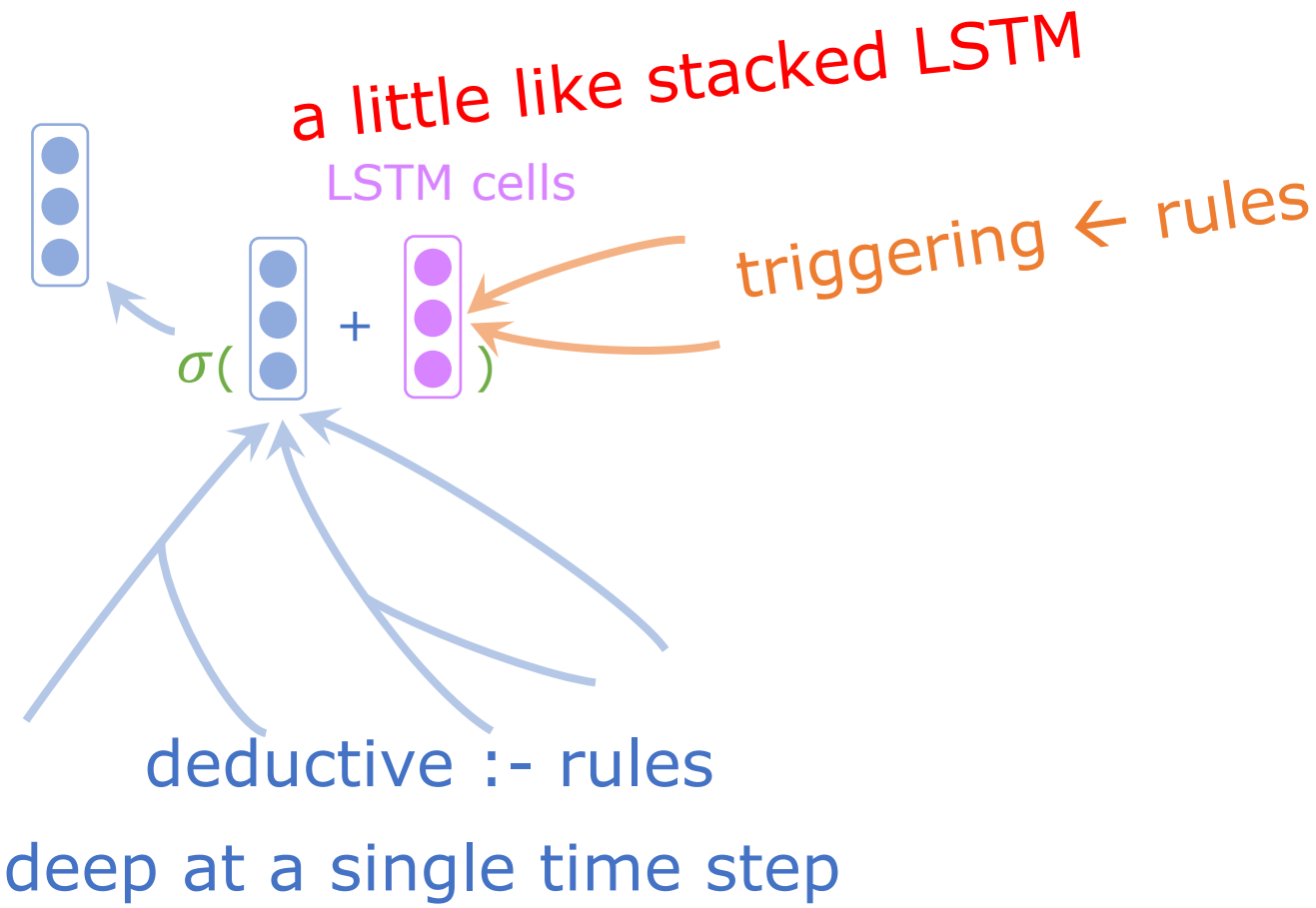
Rules ➔ Deep Recurrent Neural Net



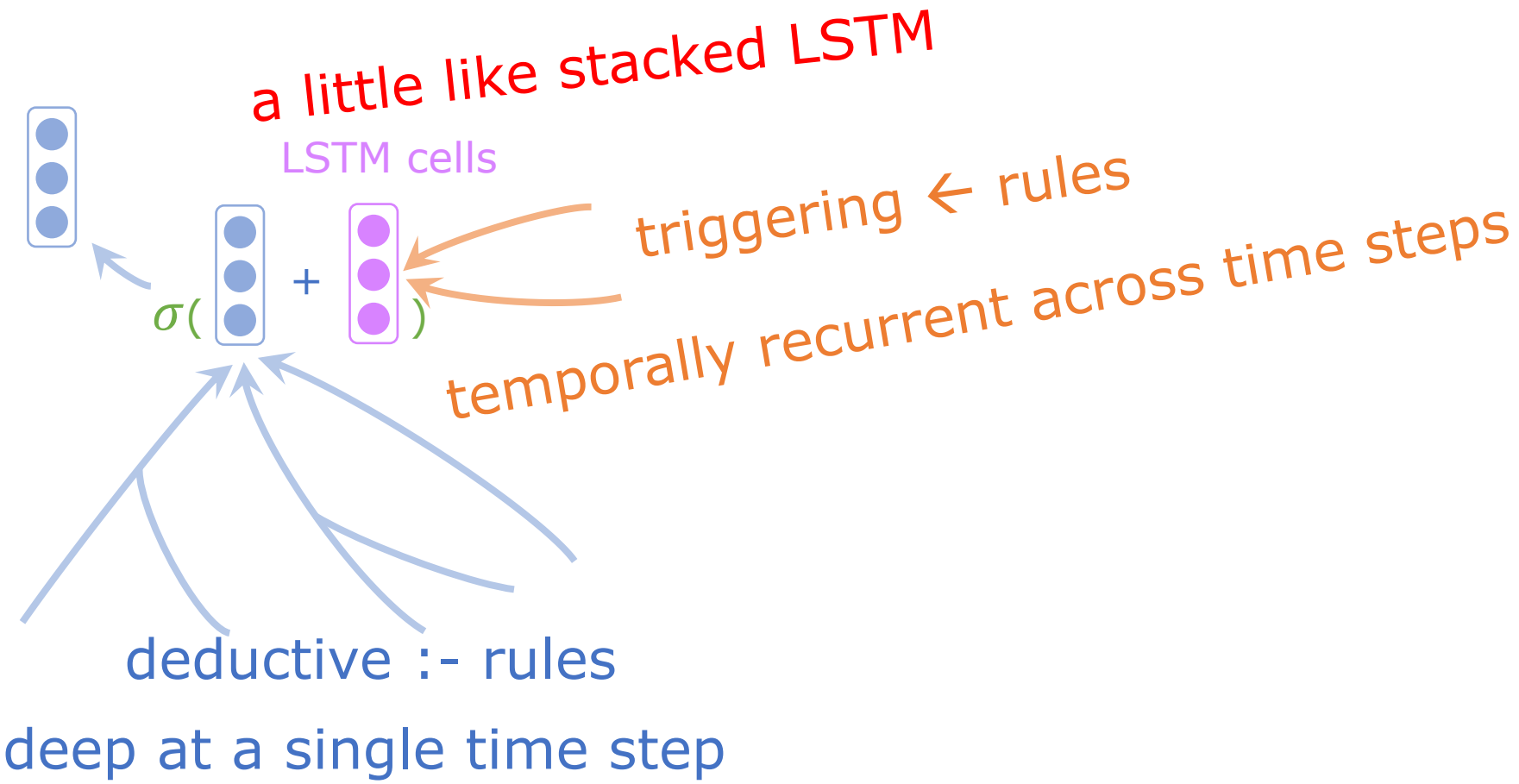
Rules → Deep Recurrent Neural Net



Rules → Deep Recurrent Neural Net



Rules → Deep Recurrent Neural Net



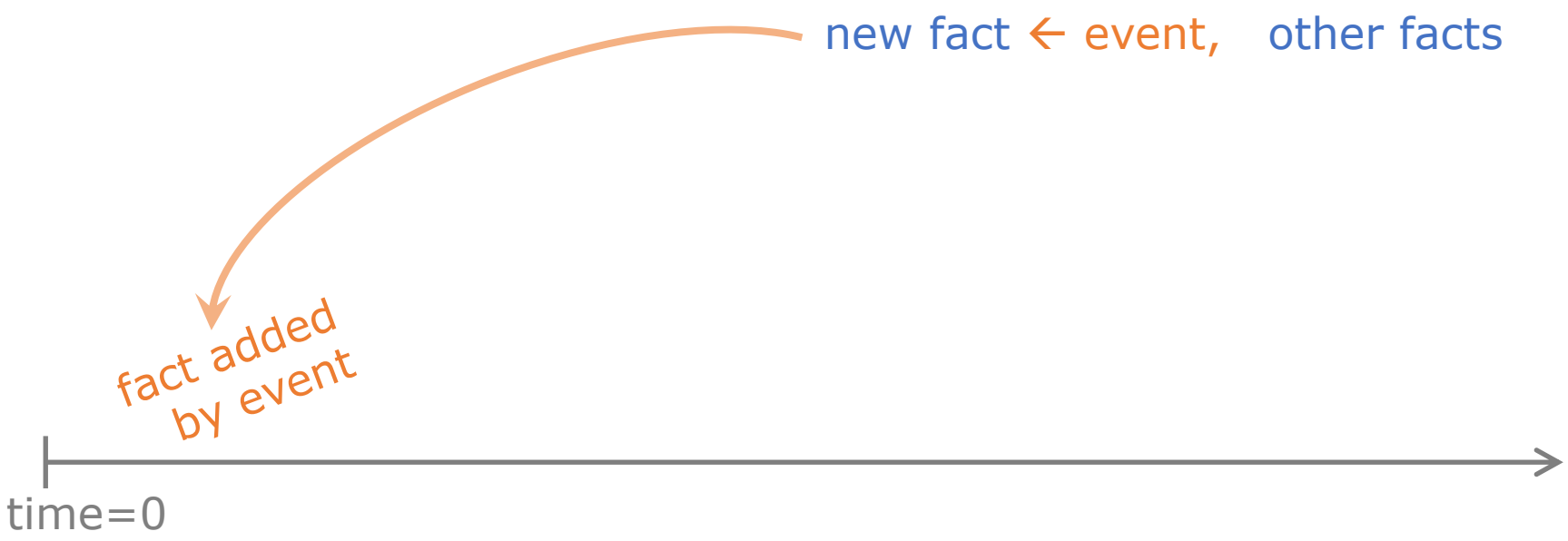
Life Story of a Fact

Life Story of a Fact

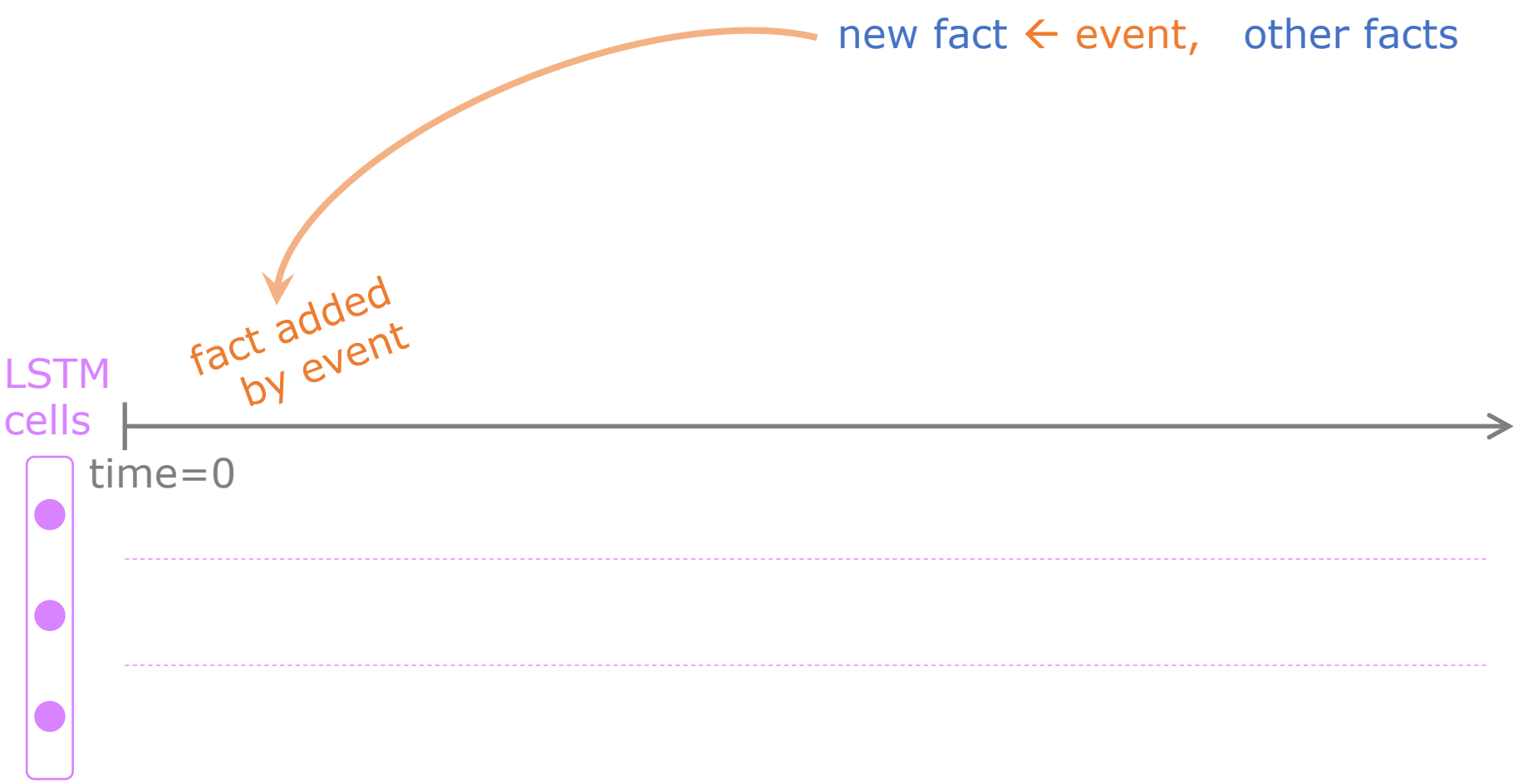
Life Story of a Fact

new fact ← event, other facts

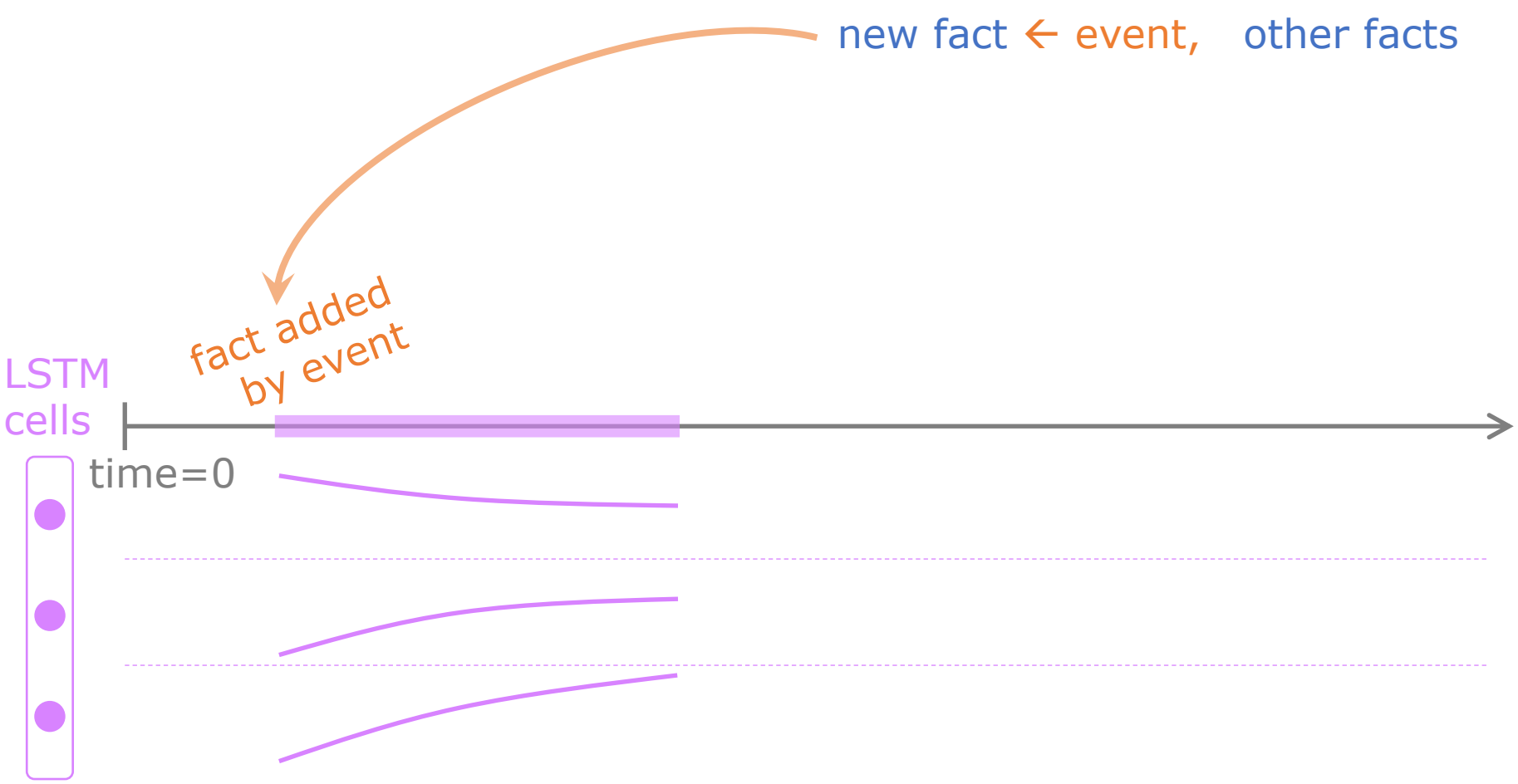
Life Story of a Fact



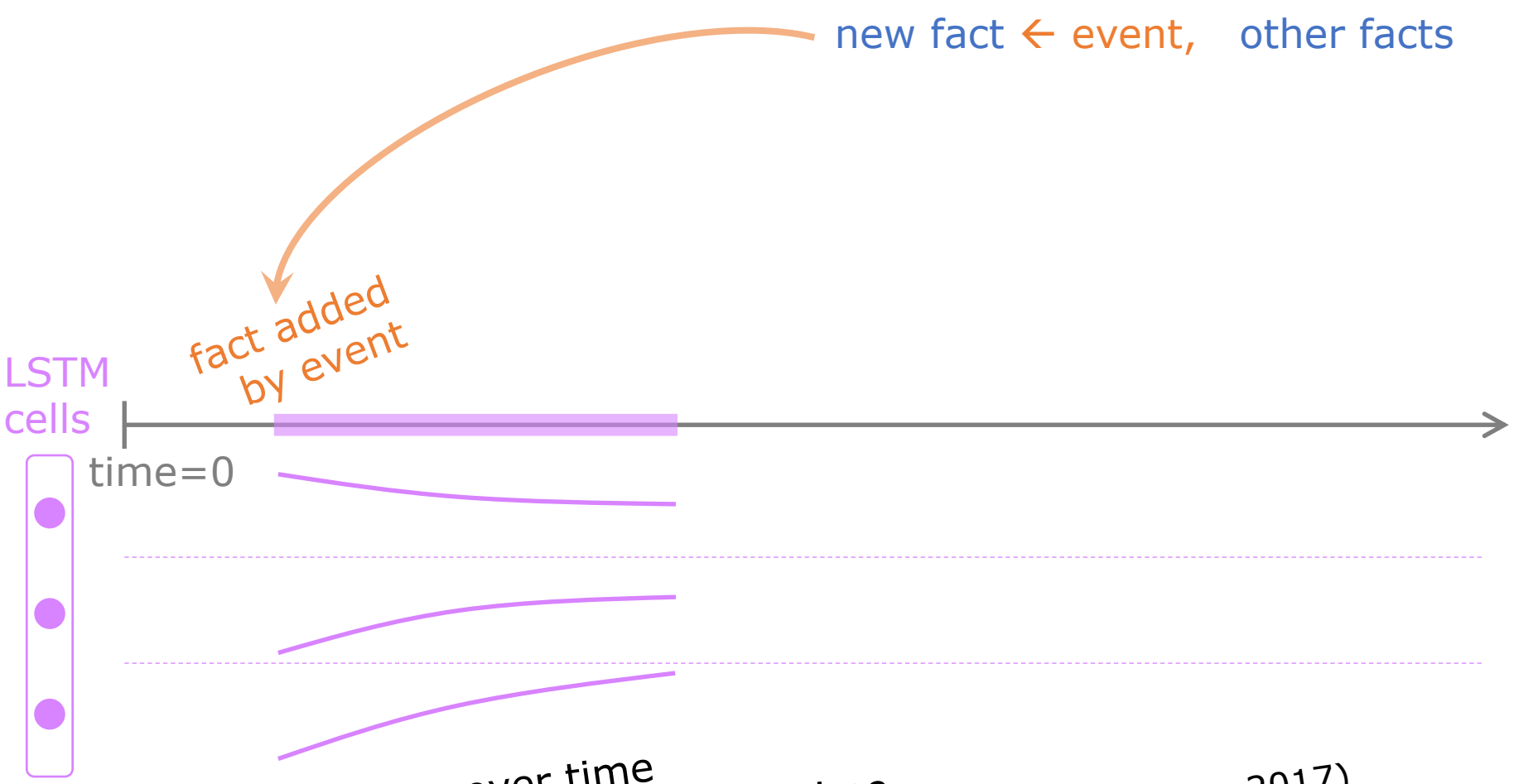
Life Story of a Fact



Life Story of a Fact

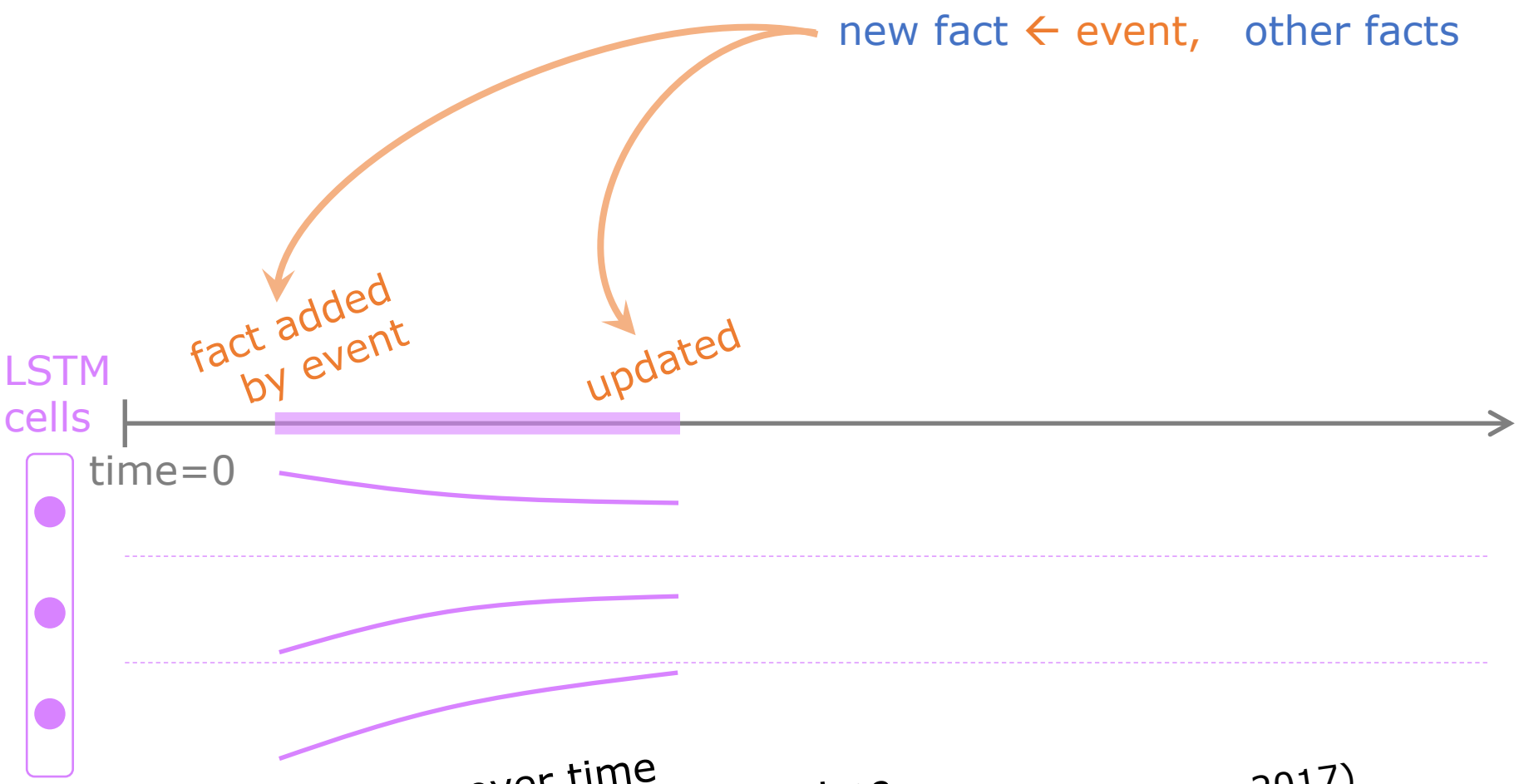


Life Story of a Fact



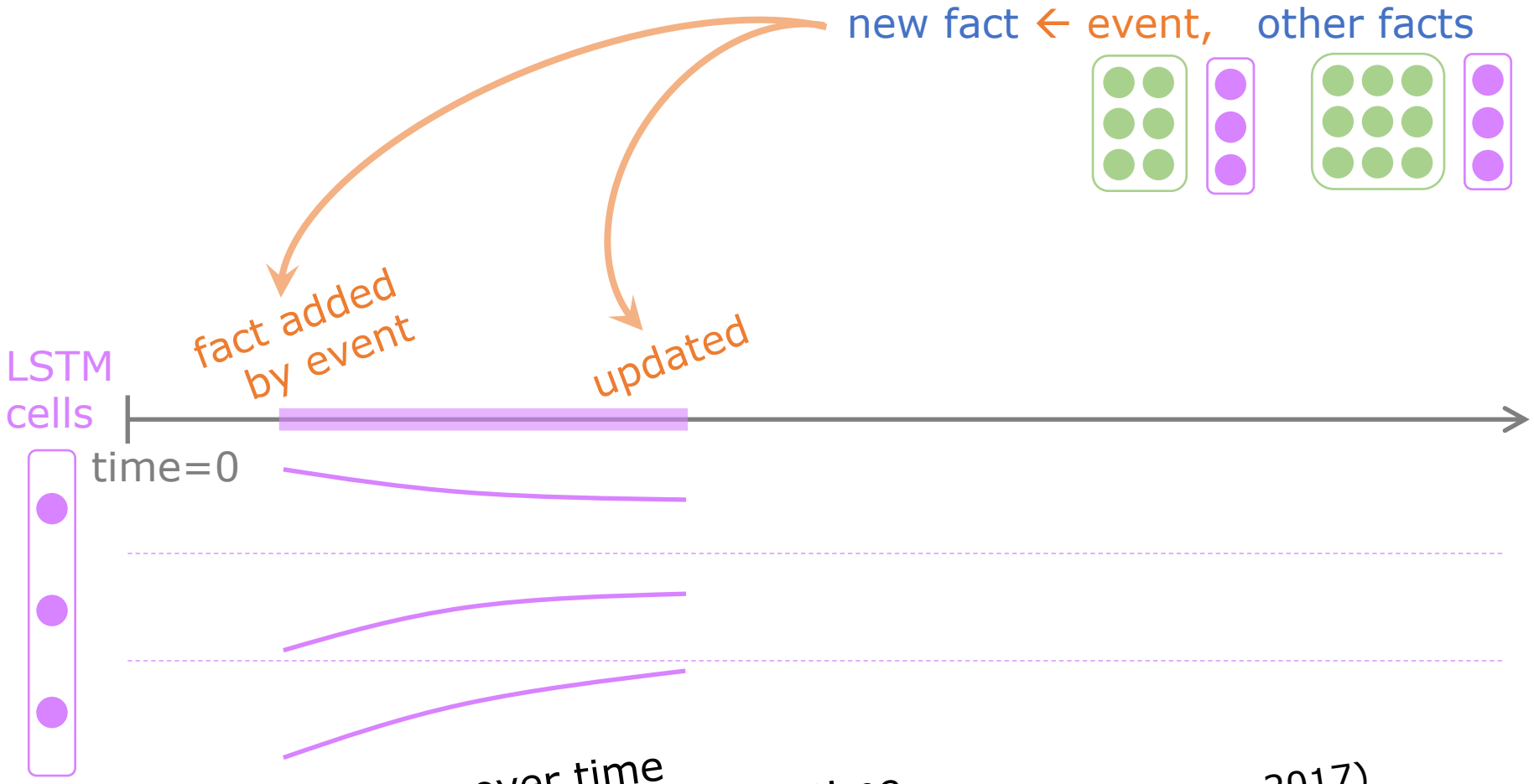
embedding changes over time
probability changes over time
≈ neural Hawkes process (Mei & Eisner, 2017)

Life Story of a Fact

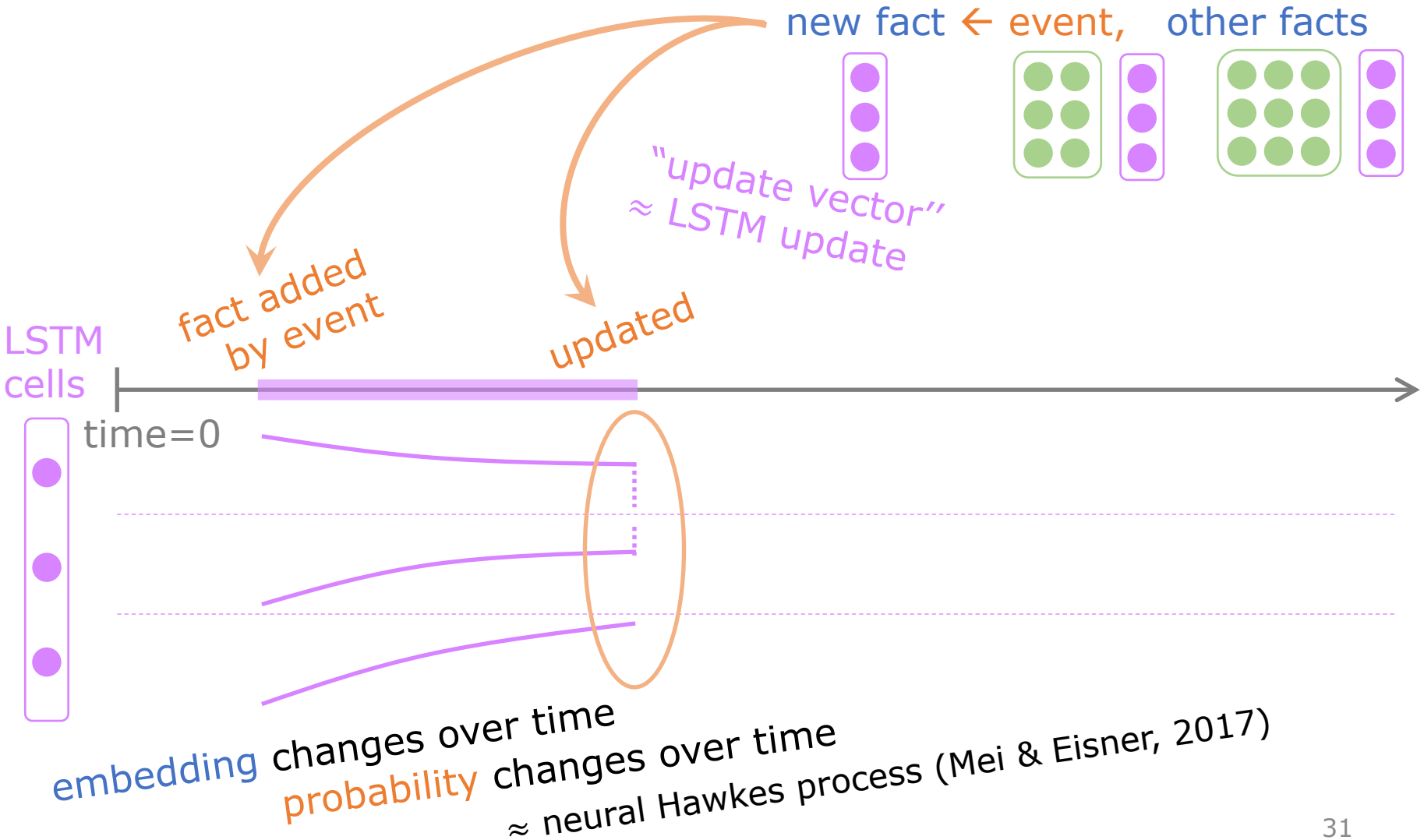


embedding changes over time
probability changes over time
≈ neural Hawkes process (Mei & Eisner, 2017)

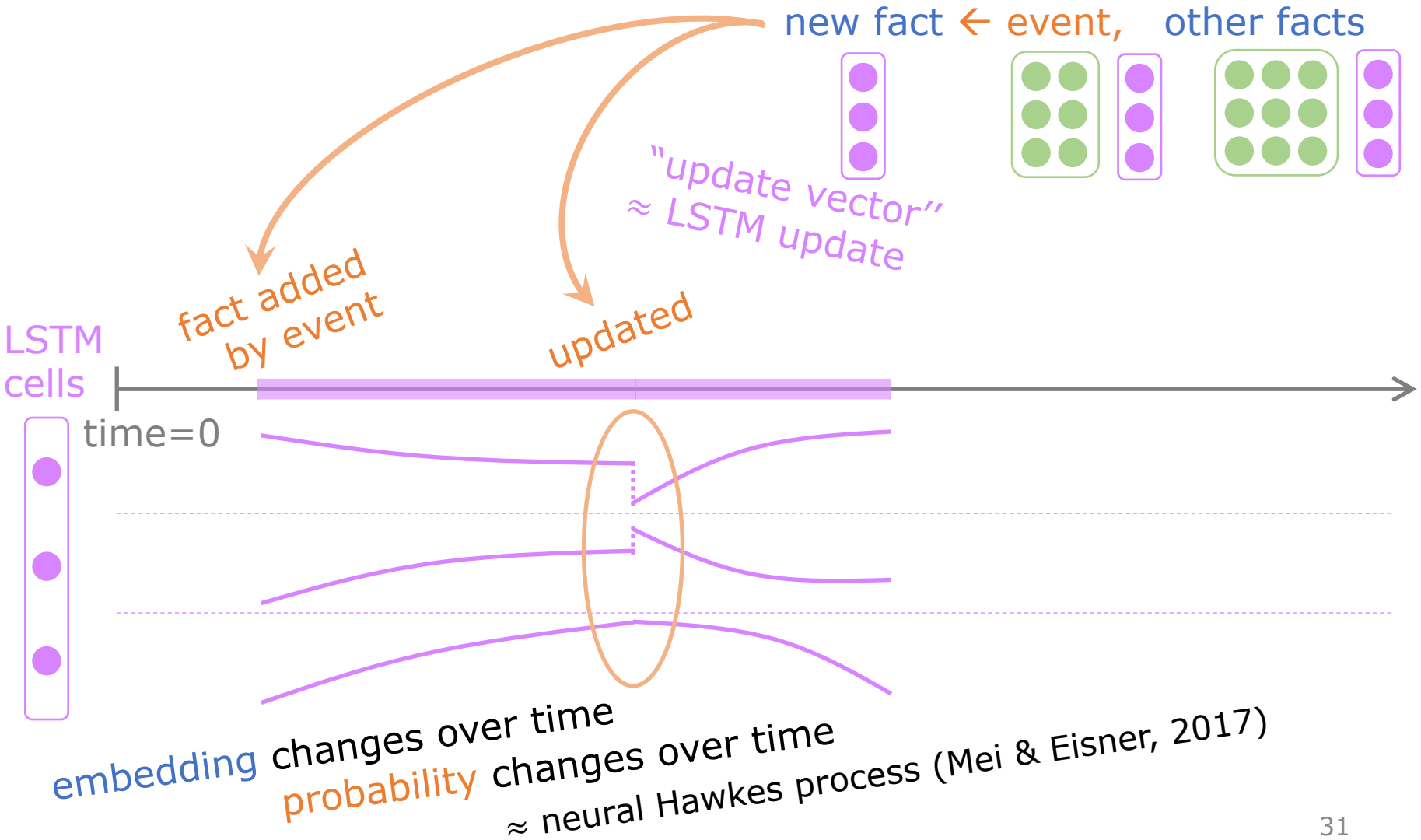
Life Story of a Fact



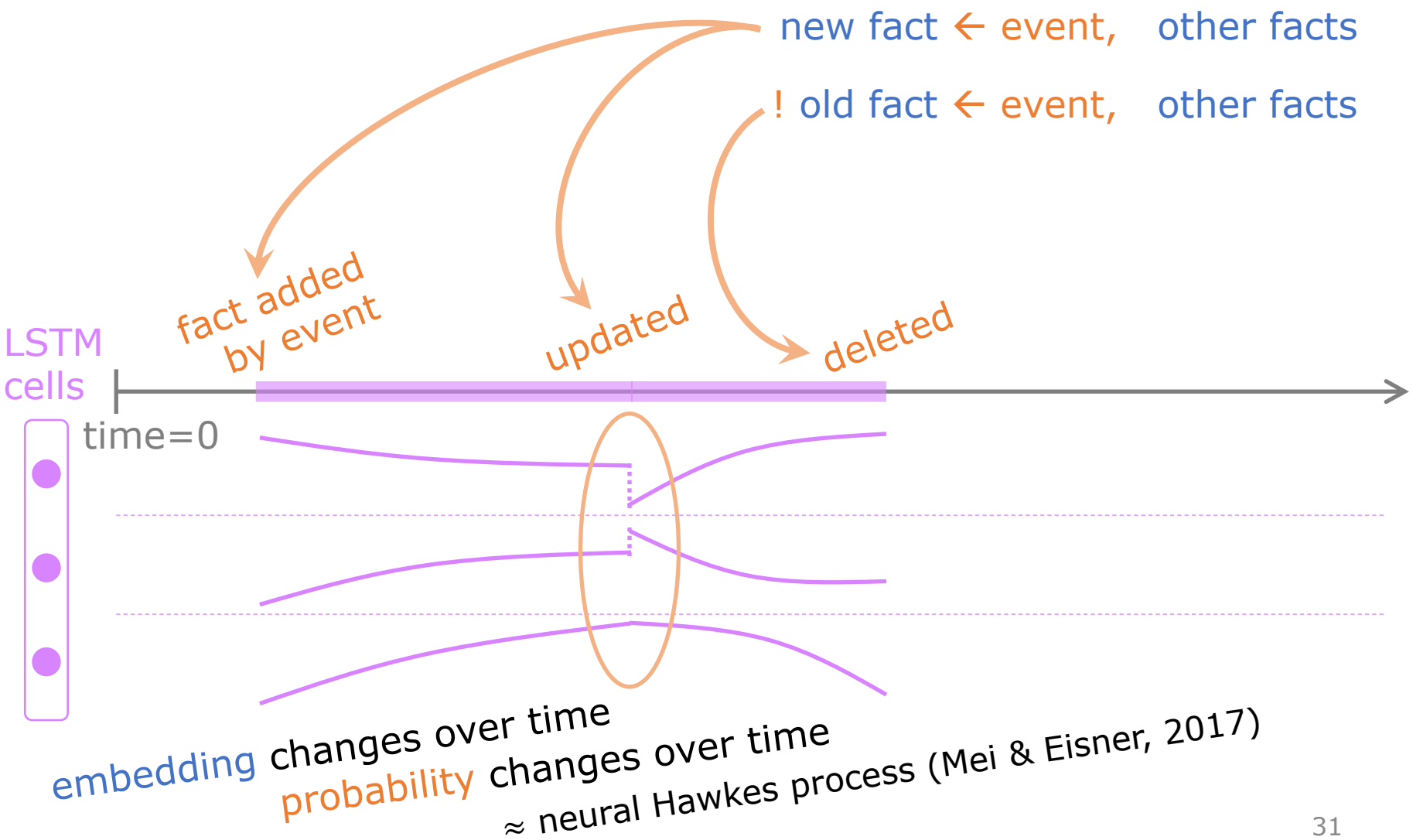
Life Story of a Fact



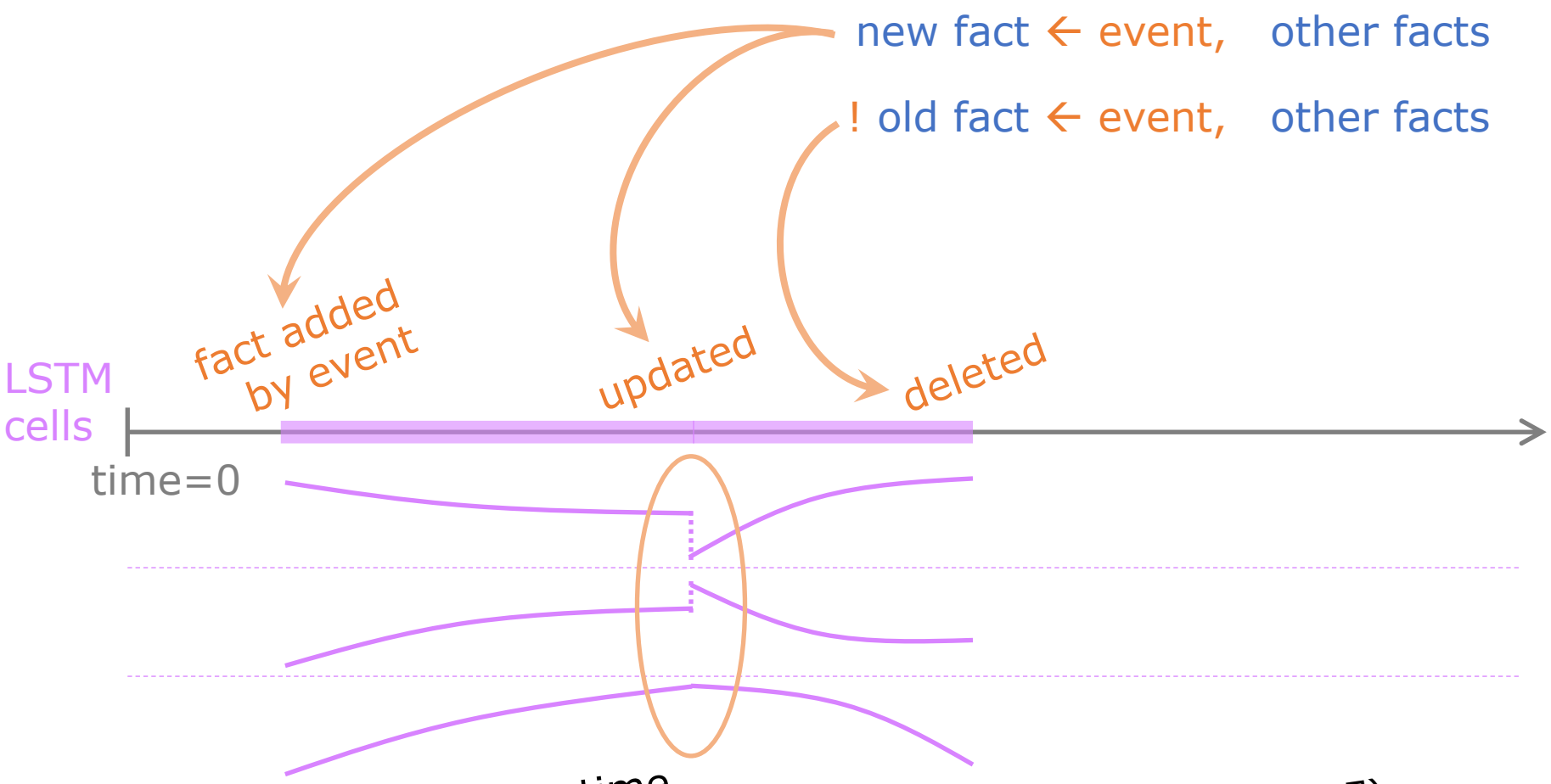
Life Story of a Fact



Life Story of a Fact

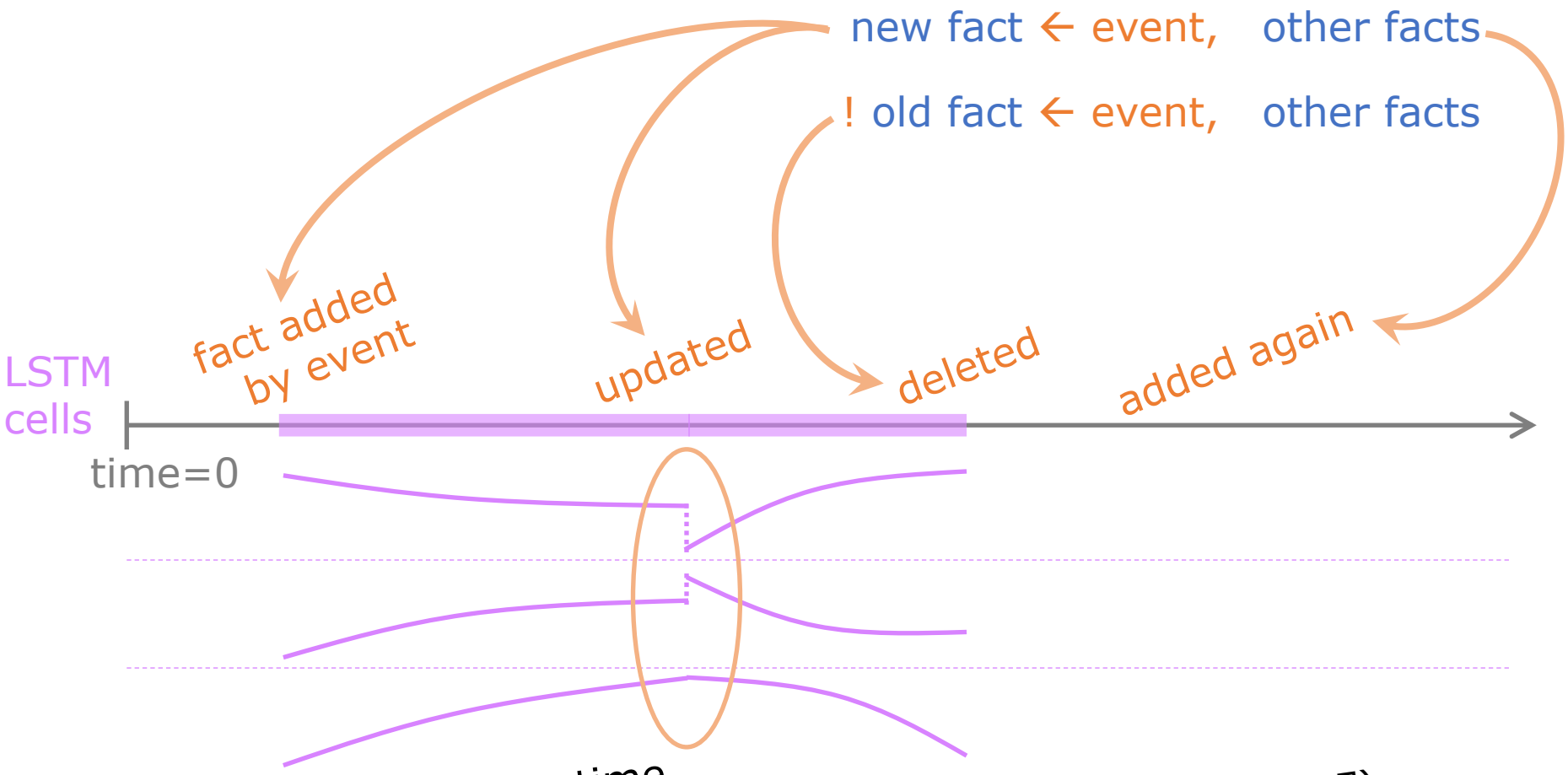


Life Story of a Fact



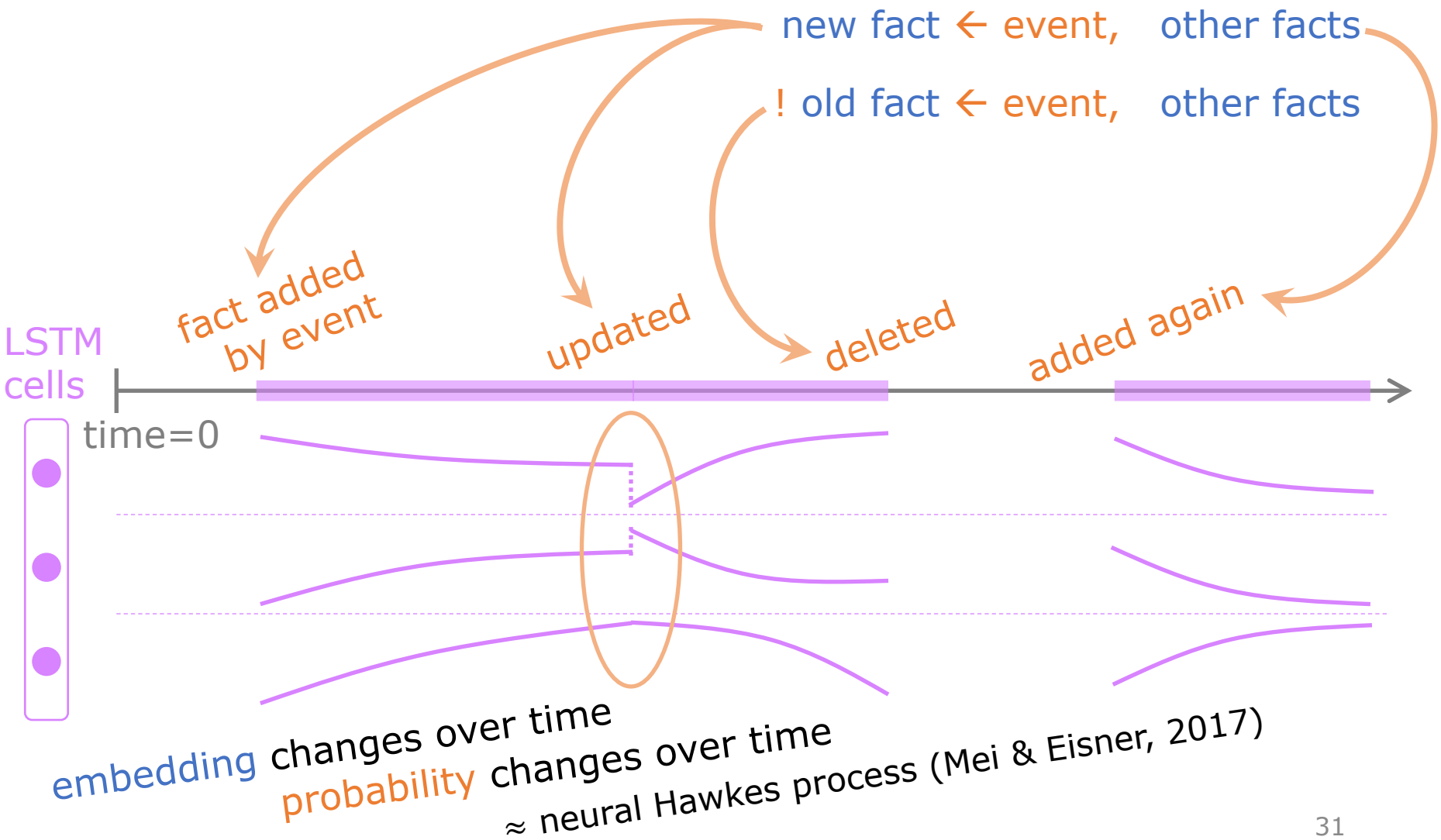
embedding changes over time
probability changes over time
≈ neural Hawkes process (Mei & Eisner, 2017)

Life Story of a Fact



embedding changes over time
probability changes over time
≈ neural Hawkes process (Mei & Eisner, 2017)

Life Story of a Fact



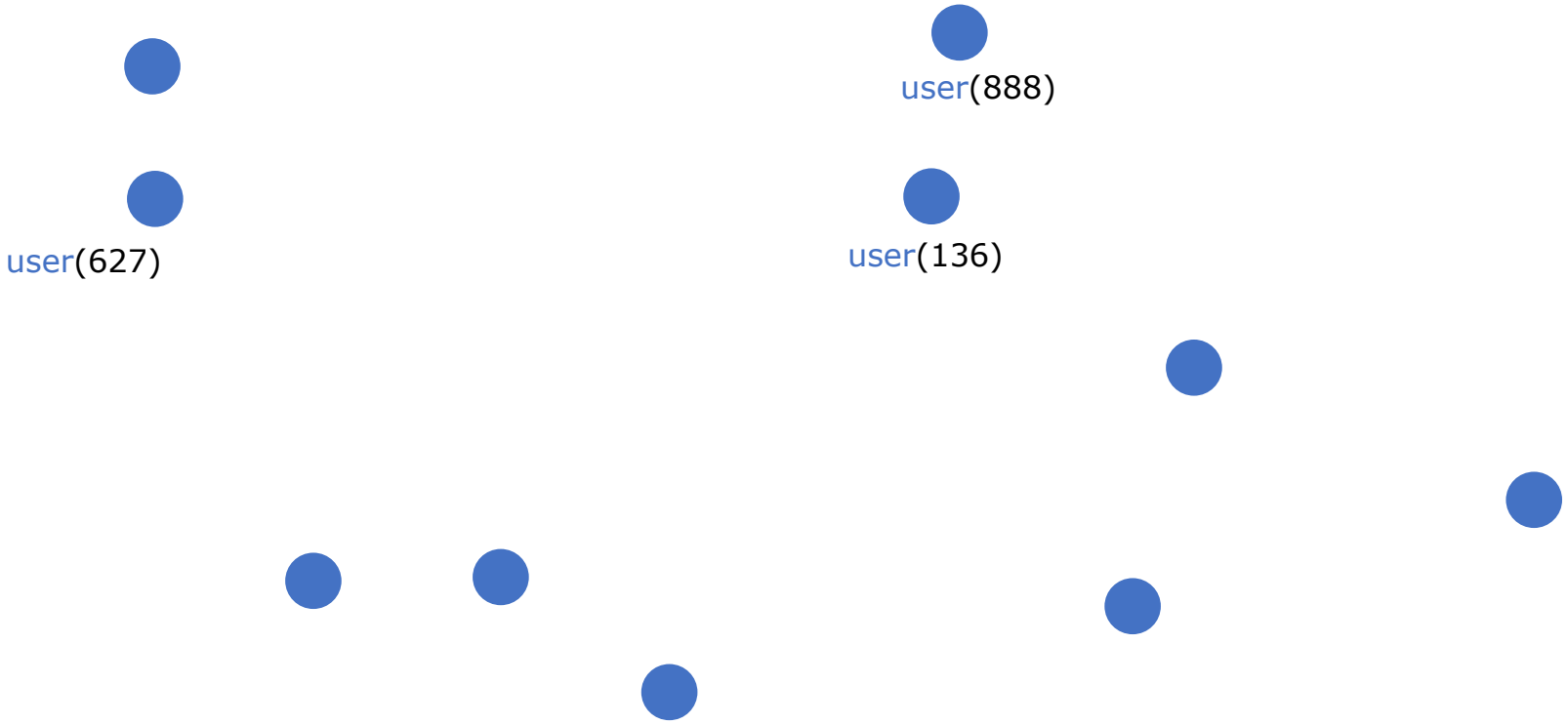
Experiment: Users watch TV programs

collaborative filtering problem with timing
who watches what and when?

Experiment: Users watch TV programs

collaborative filtering problem with timing
who watches what and when?

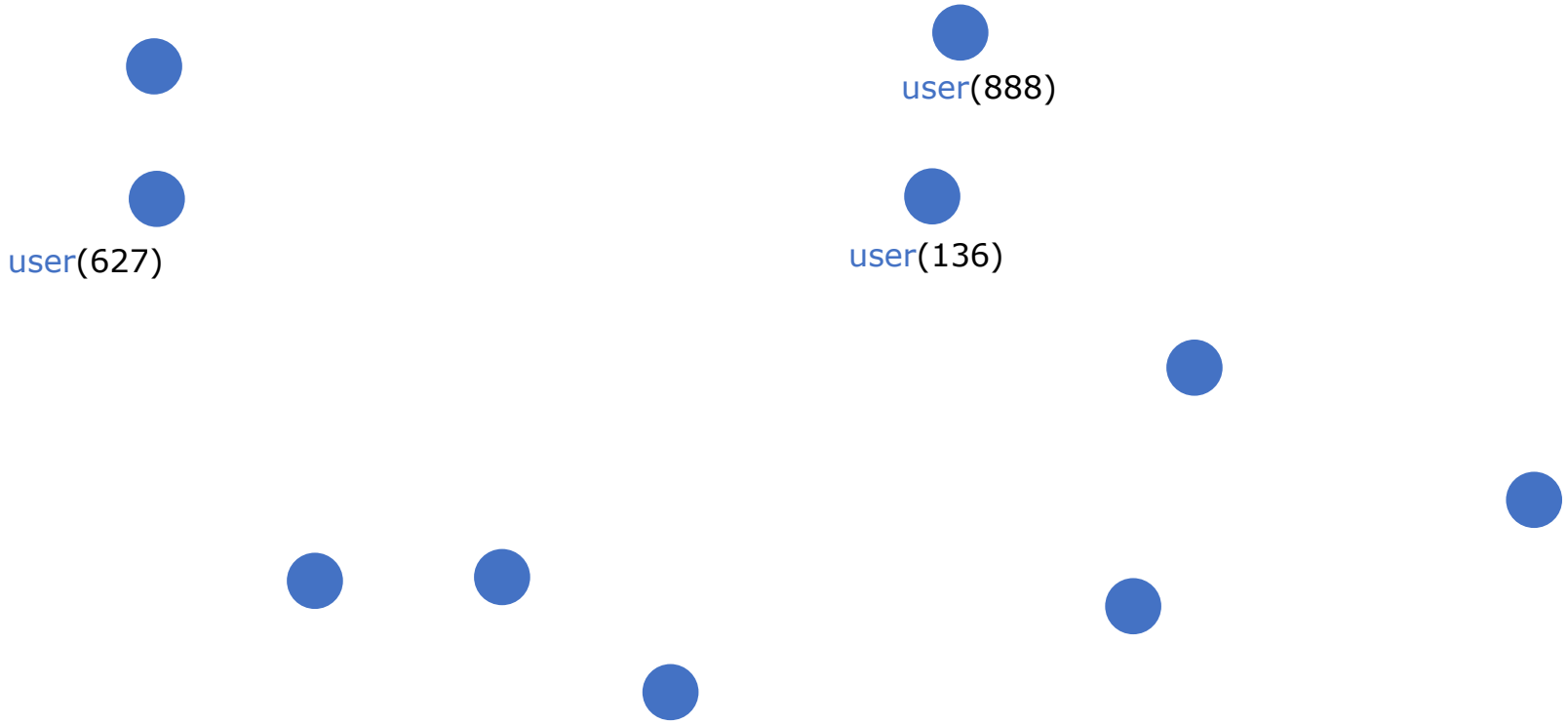
1000 users



Experiment: Users watch TV programs

collaborative filtering problem with timing
who watches what and when?

1000 users 49 TV programs to be released

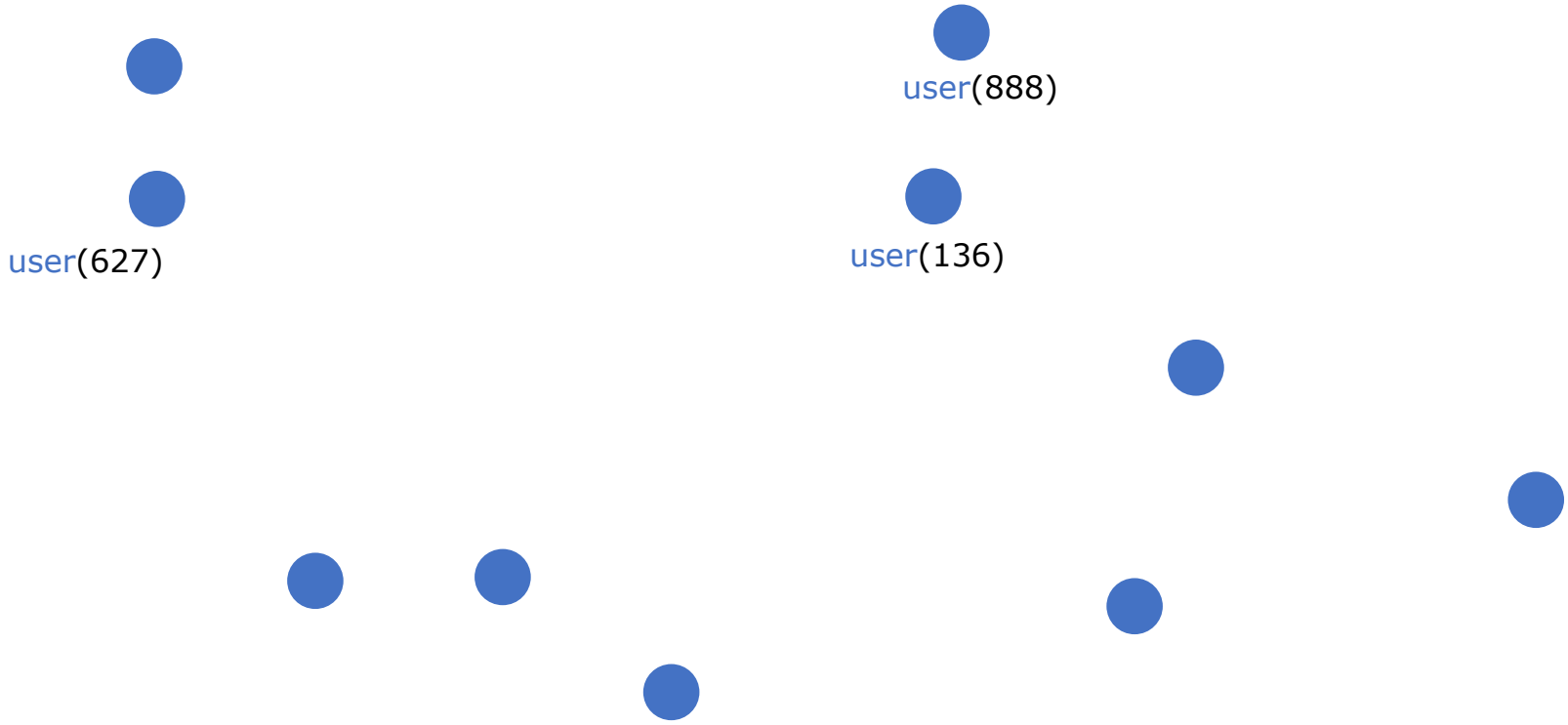


Experiment: Users watch TV programs

collaborative filtering problem with timing
who watches what and when?

1000 users 49 TV programs to be released

49000 possible watch events



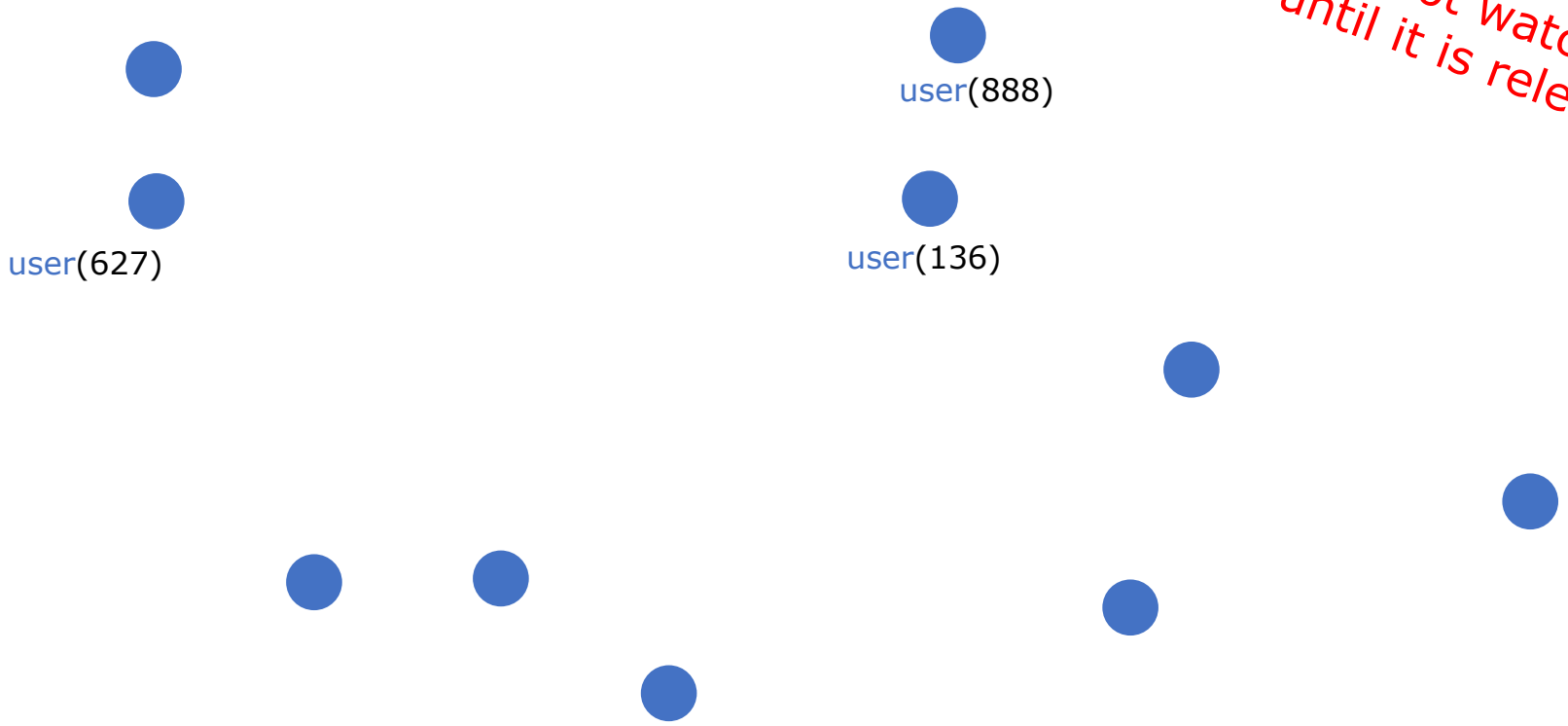
Experiment: Users watch TV programs

collaborative filtering problem with timing
who watches what and when?

1000 users 49 TV programs to be released

49000 possible watch events

can not watch it
until it is released

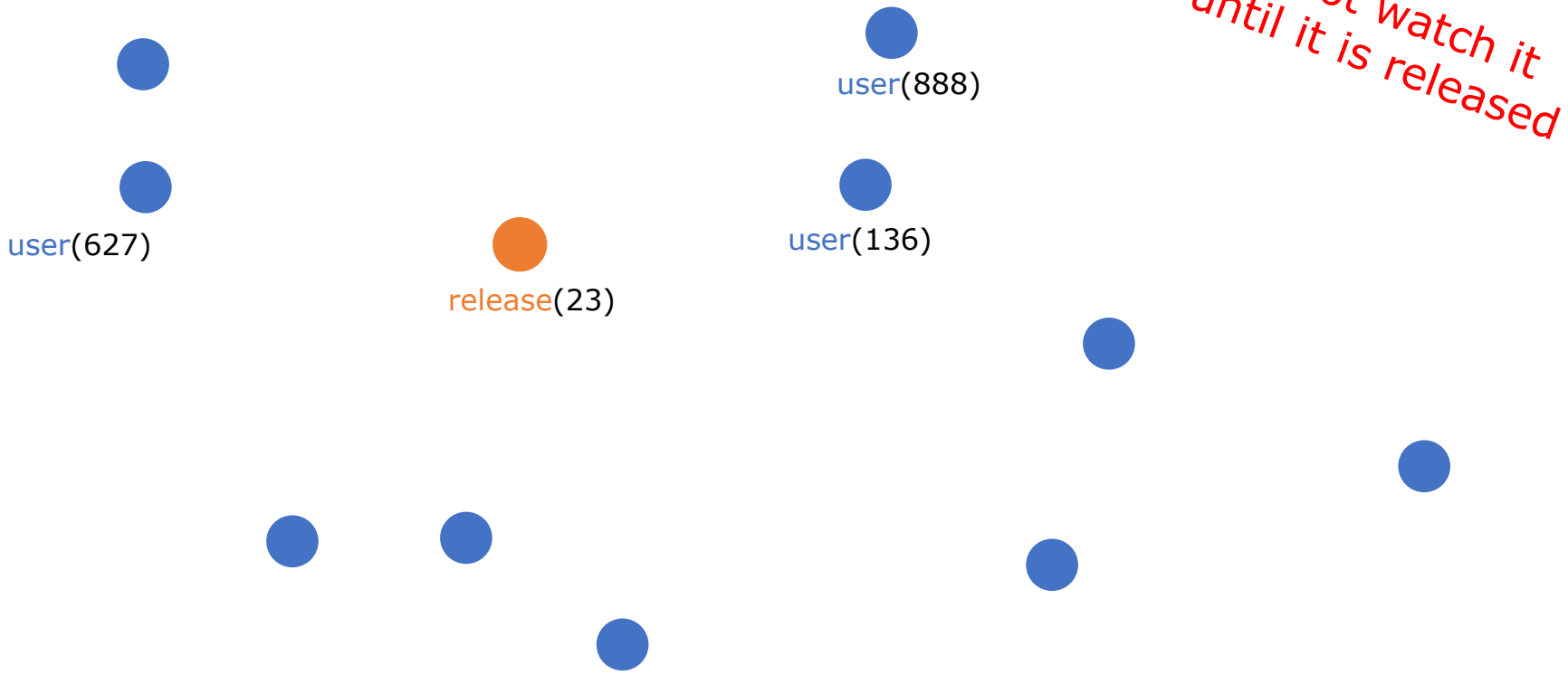


Experiment: Users watch TV programs

collaborative filtering problem with timing
who watches what and when?

1000 users 49 TV programs to be released

49000 possible watch events



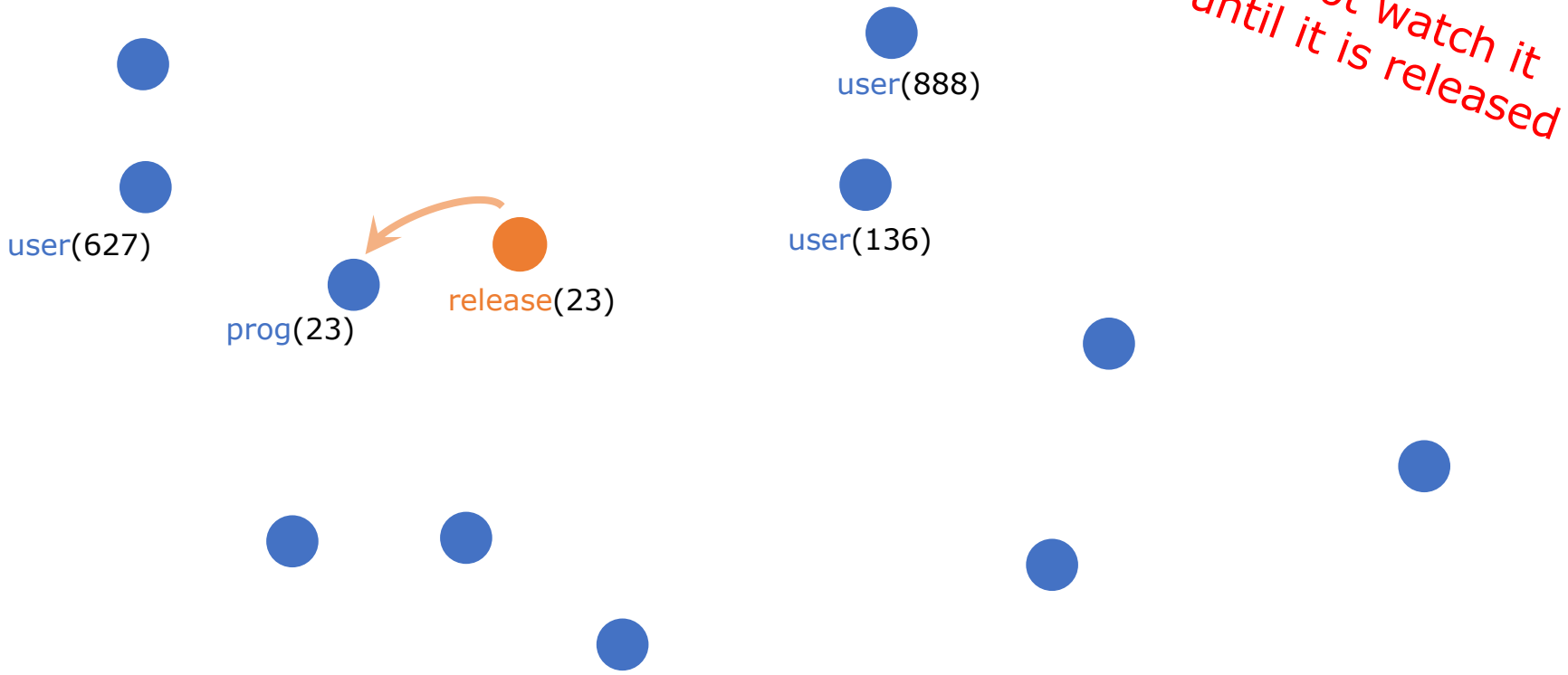
can not watch it
until it is released

Experiment: Users watch TV programs

collaborative filtering problem with timing
who watches what and when?

1000 users 49 TV programs to be released

49000 possible watch events

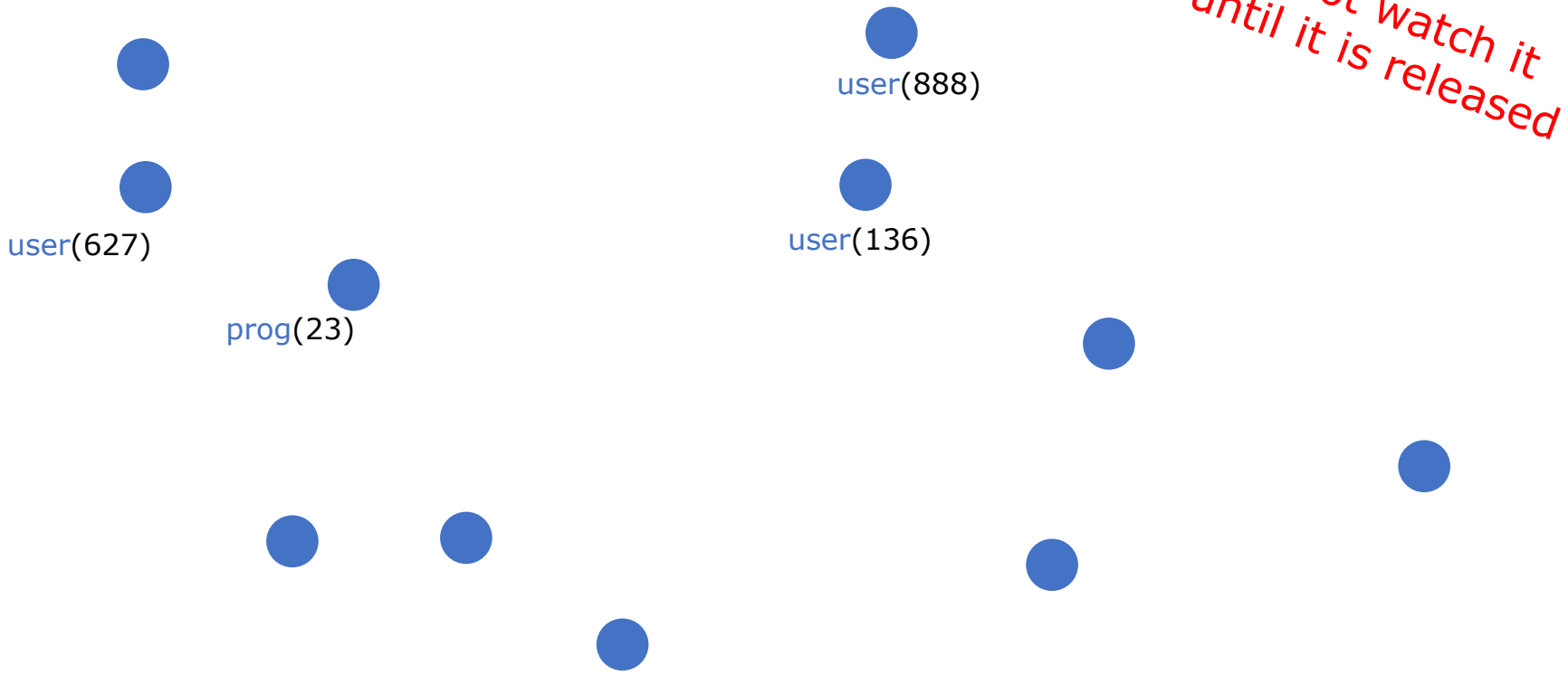


Experiment: Users watch TV programs

collaborative filtering problem with timing
who watches what and when?

1000 users 49 TV programs to be released

49000 possible watch events



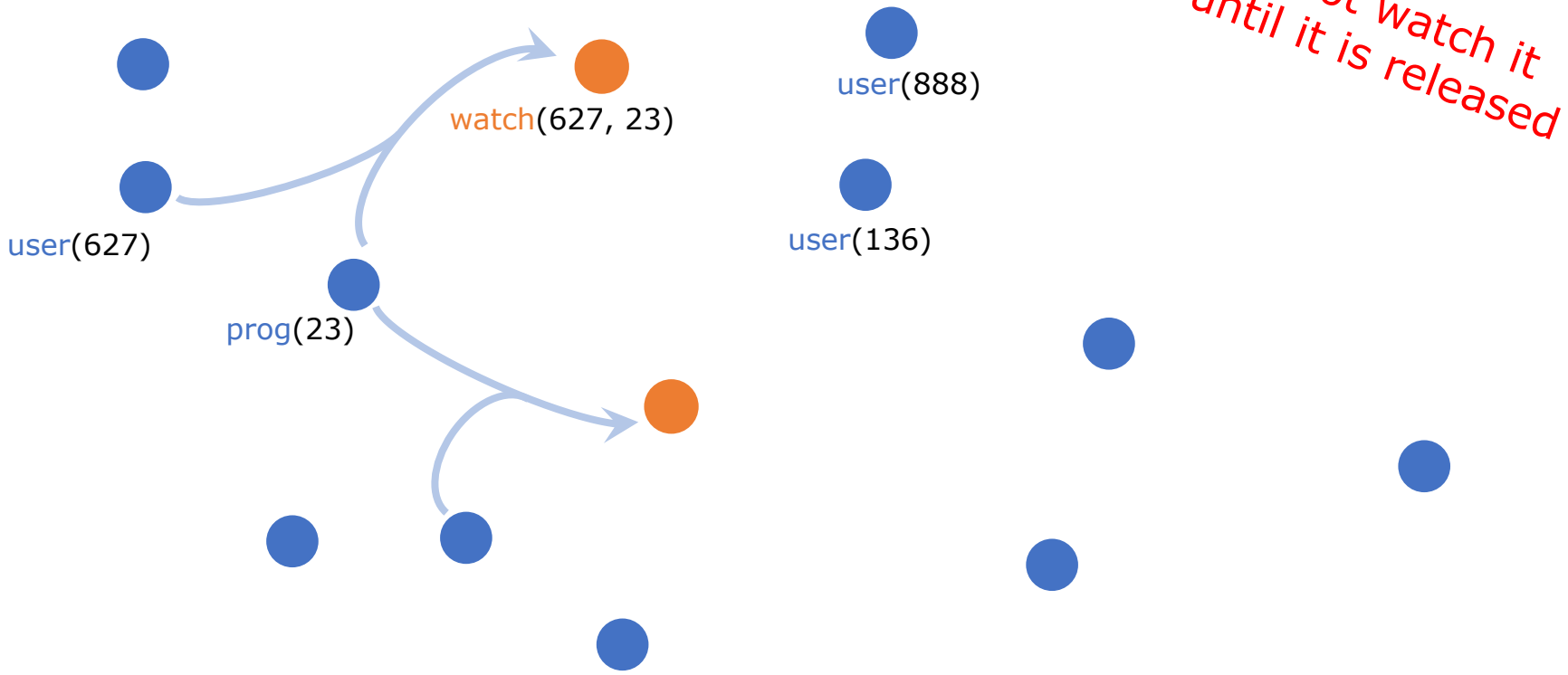
can not watch it
until it is released

Experiment: Users watch TV programs

collaborative filtering problem with timing
who watches what and when?

1000 users 49 TV programs to be released

49000 possible watch events



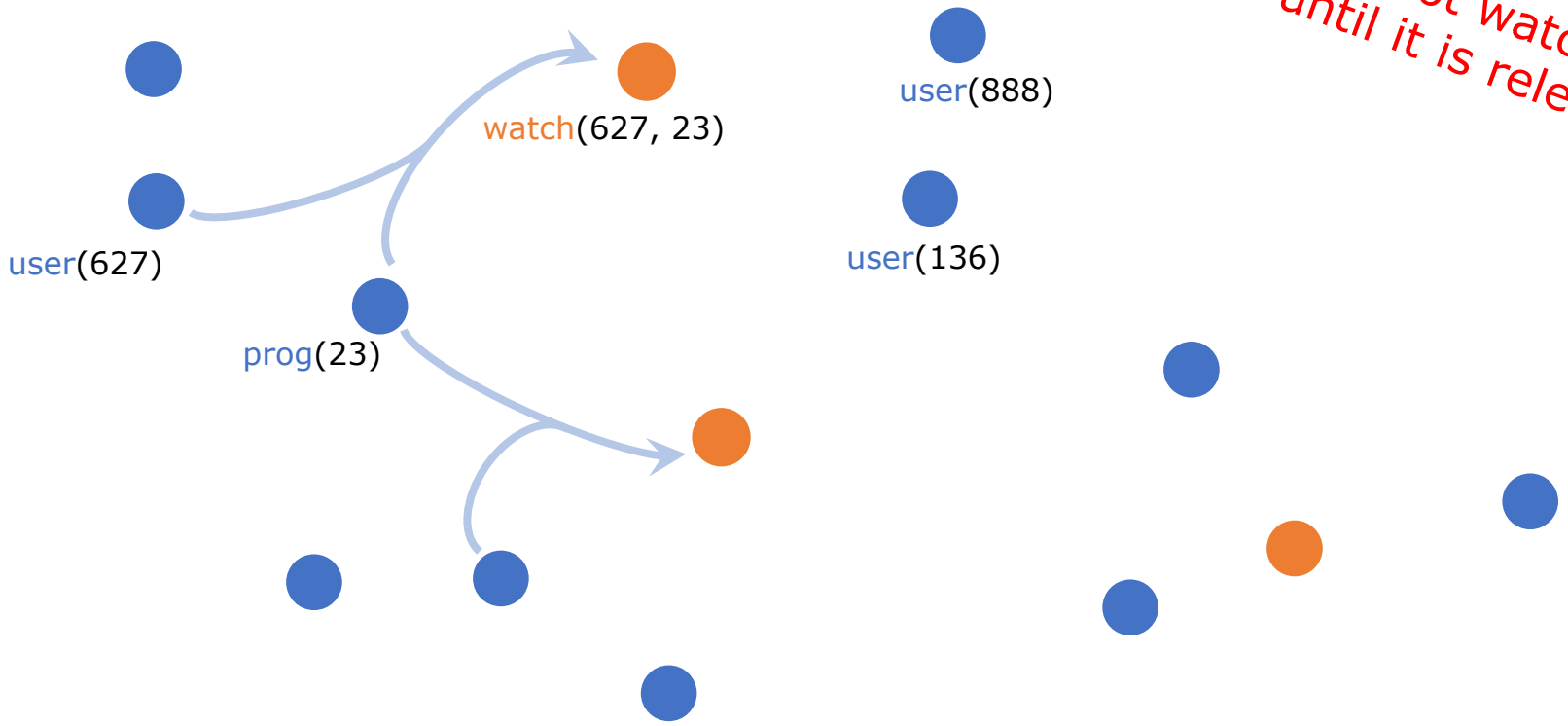
can not watch it
until it is released

Experiment: Users watch TV programs

collaborative filtering problem with timing
who watches what and when?

1000 users 49 TV programs to be released

49000 possible watch events



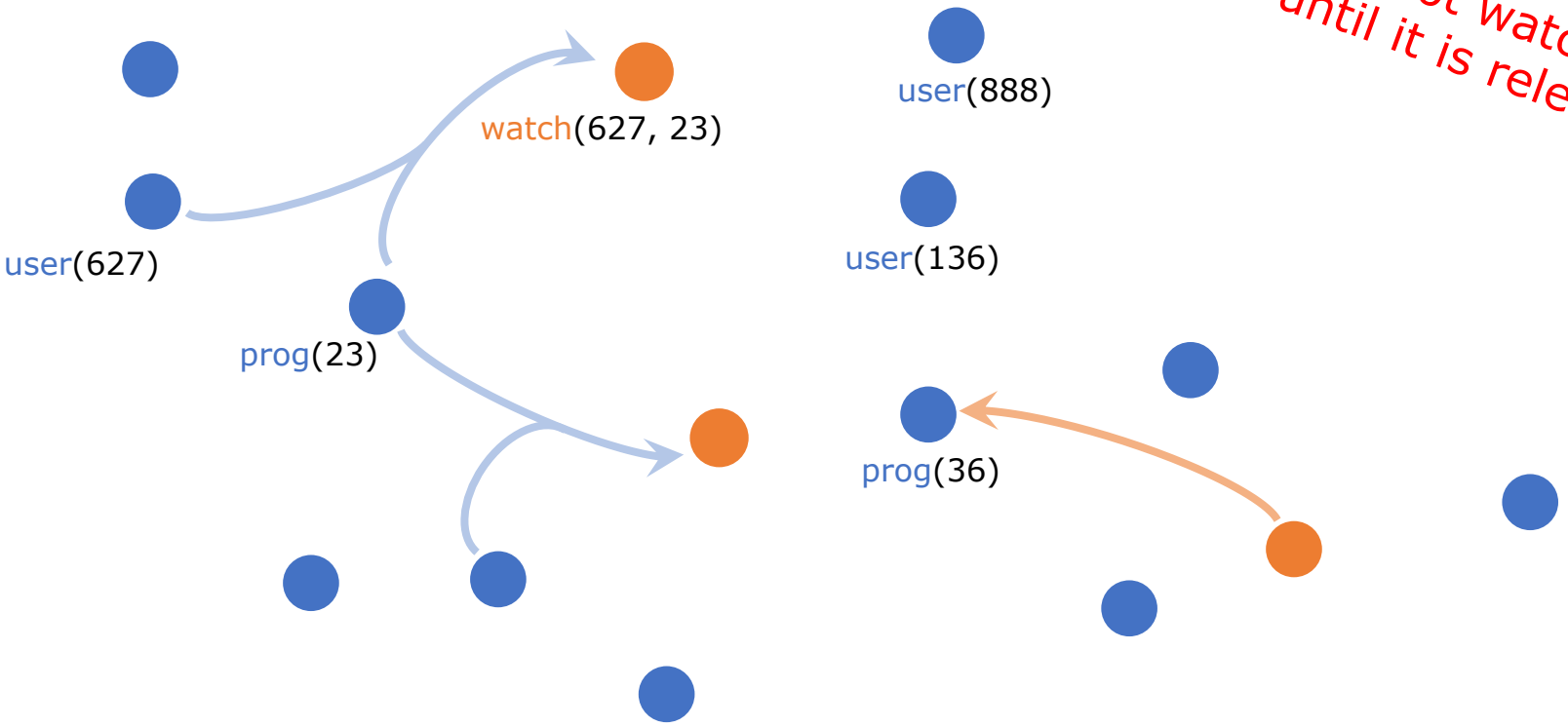
can not watch it
until it is released

Experiment: Users watch TV programs

collaborative filtering problem with timing
who watches what and when?

1000 users 49 TV programs to be released

49000 possible watch events



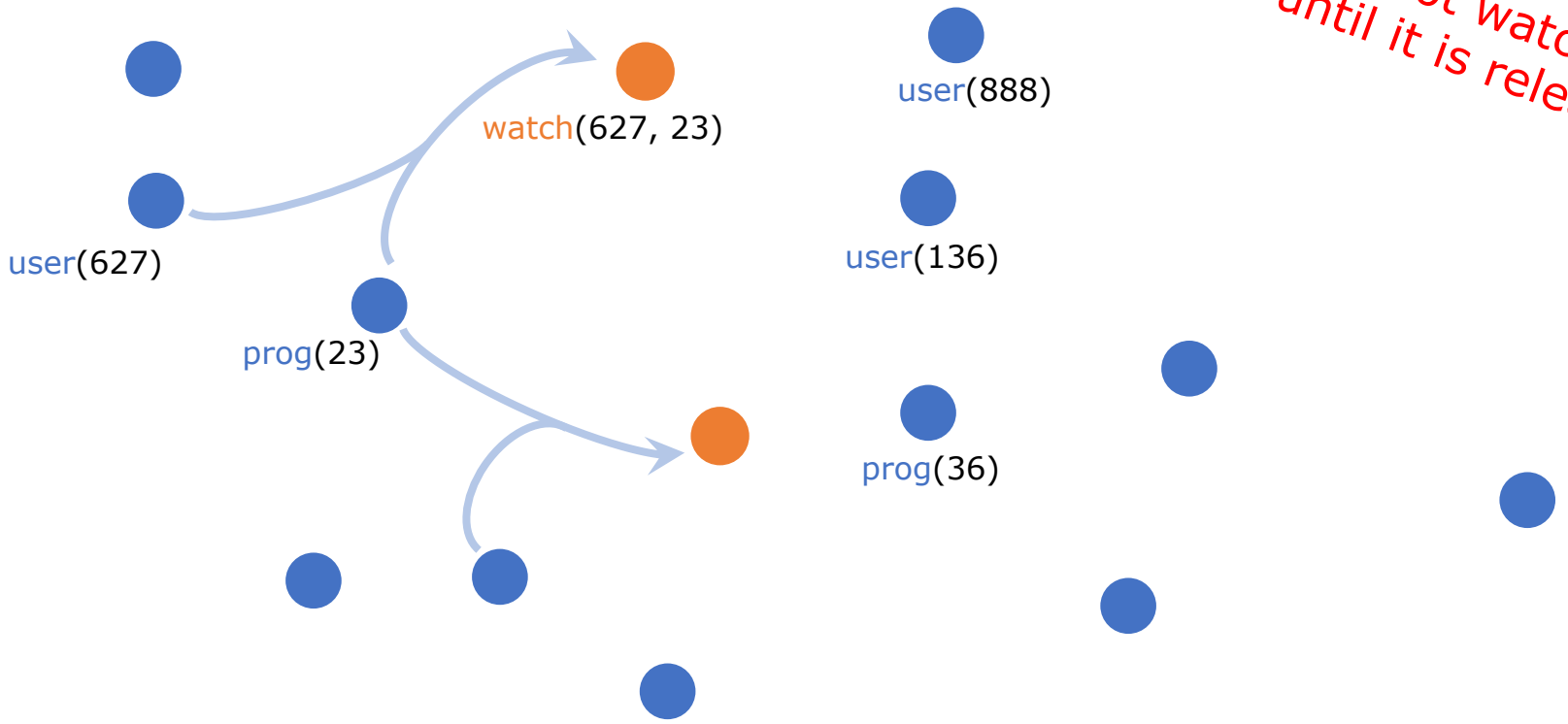
can not watch it
until it is released

Experiment: Users watch TV programs

collaborative filtering problem with timing
who watches what and when?

1000 users 49 TV programs to be released

49000 possible watch events



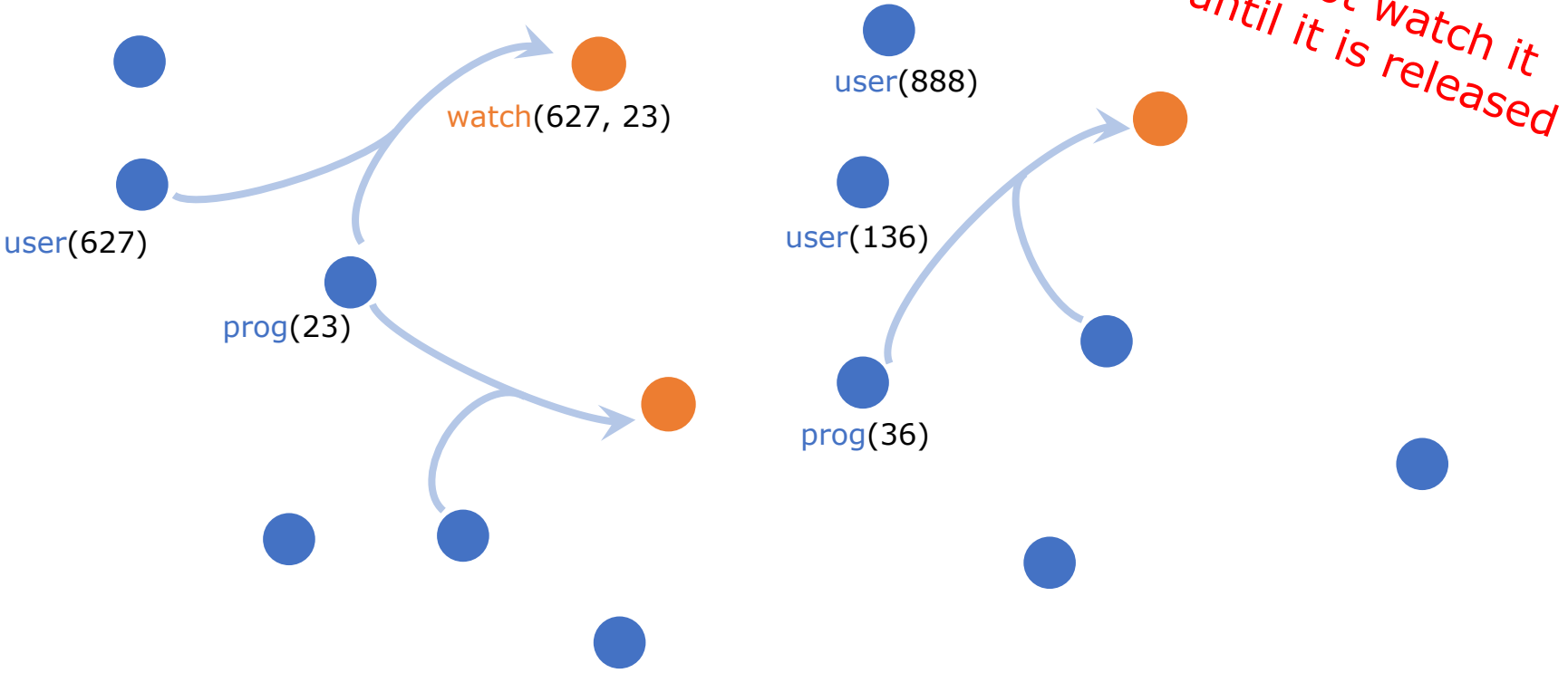
can not watch it
until it is released

Experiment: Users watch TV programs

collaborative filtering problem with timing
who watches what and when?

1000 users 49 TV programs to be released

49000 possible watch events

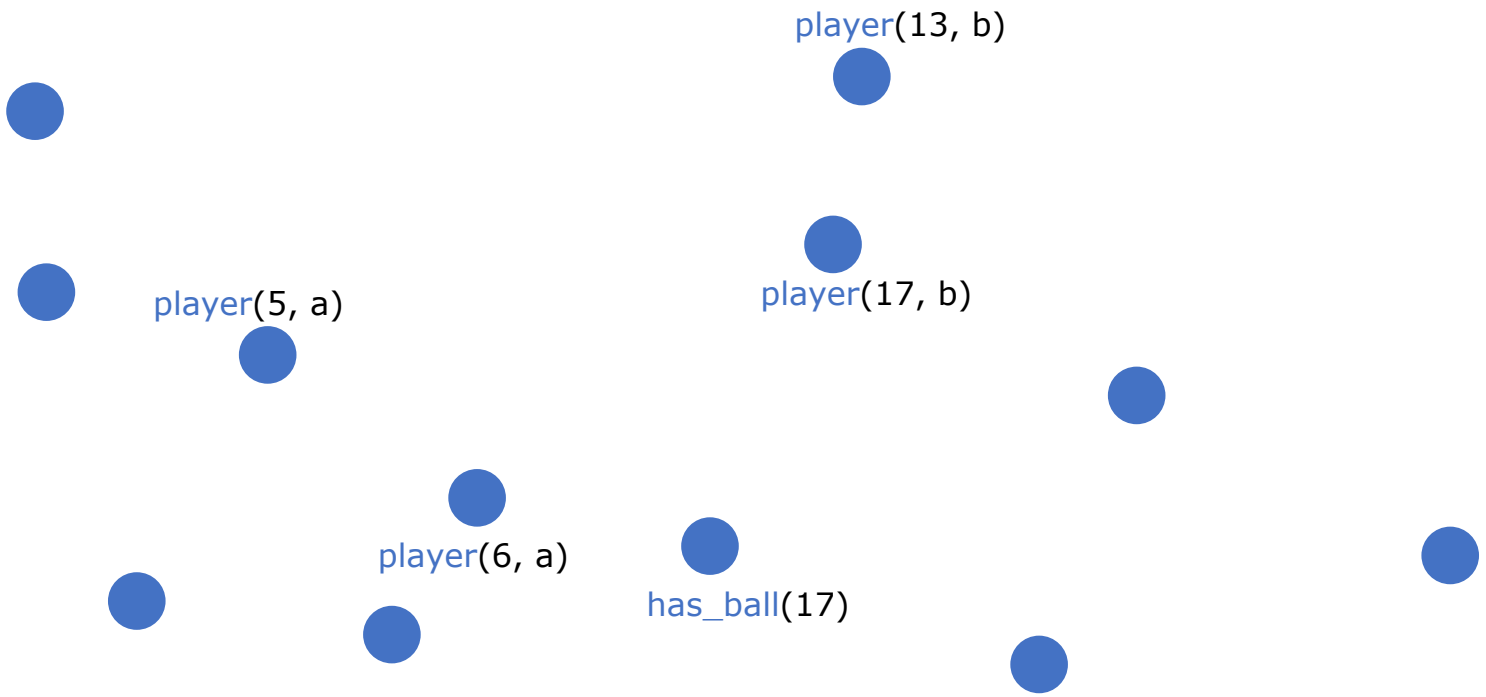


Experiment: Robots Kick/Pass/Steal

Experiment: Robots Kick/Pass/Steal

22 robot soccer players

player(Number, Team)

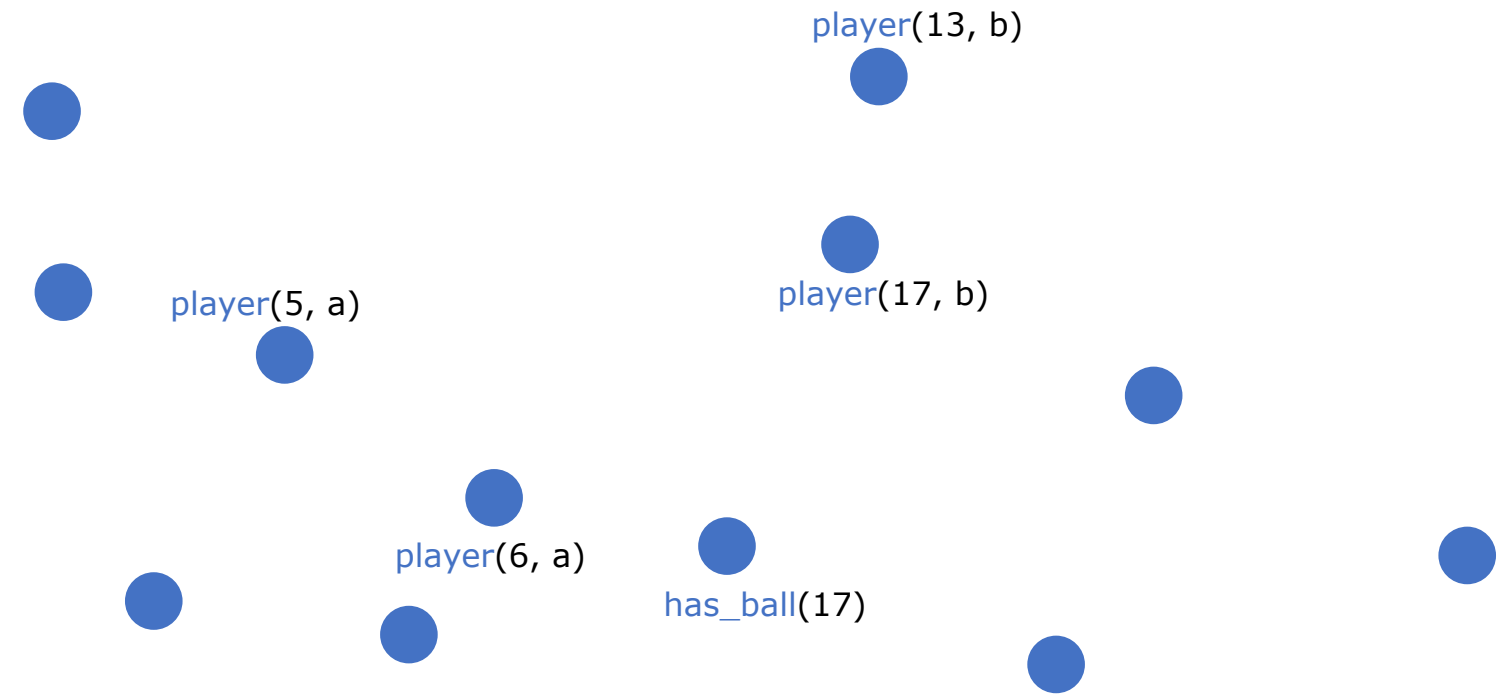


Experiment: Robots Kick/Pass/Steal

kick

22 robot soccer players

player(Number, Team)

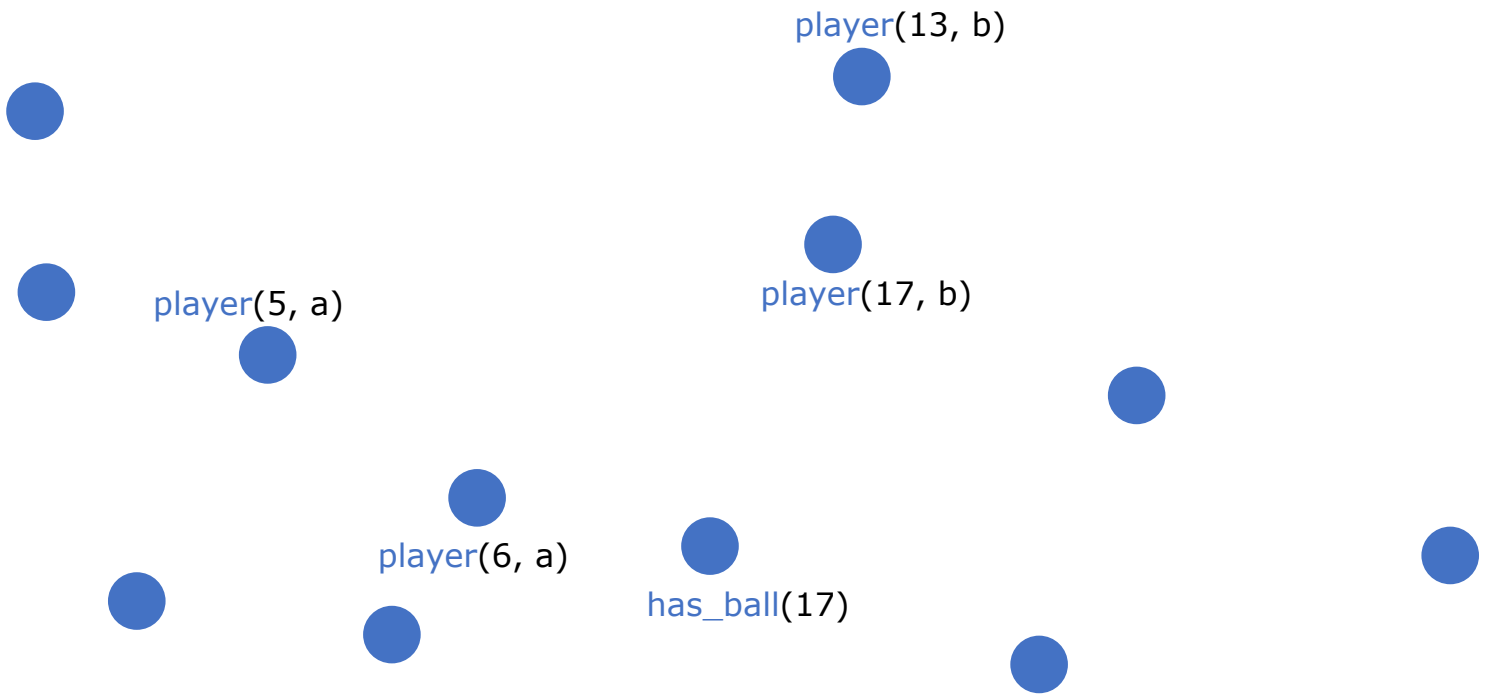


Experiment: Robots Kick/Pass/Steal

kick only if has ball

22 robot soccer players

player(Number, Team)



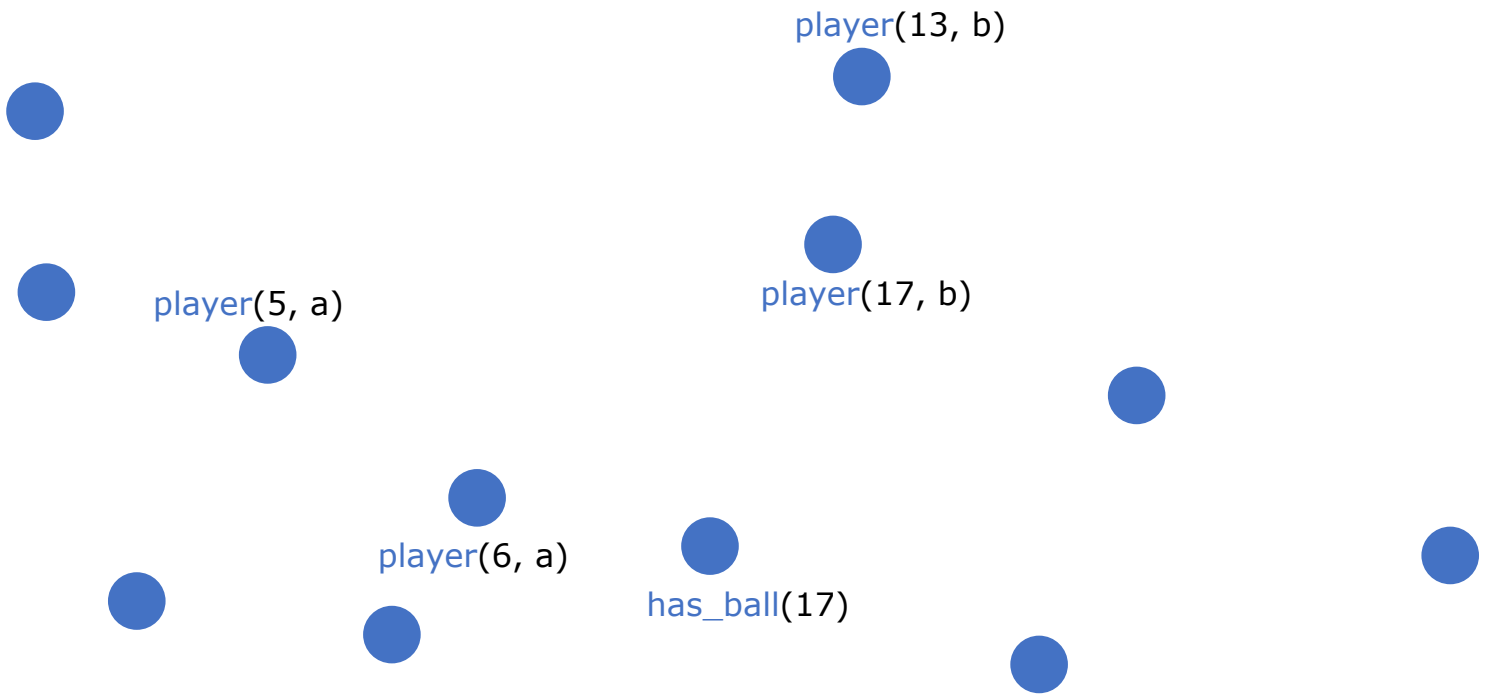
Experiment: Robots Kick/Pass/Steal

22 robot soccer players

player(Number, Team)

kick only if has ball

pass



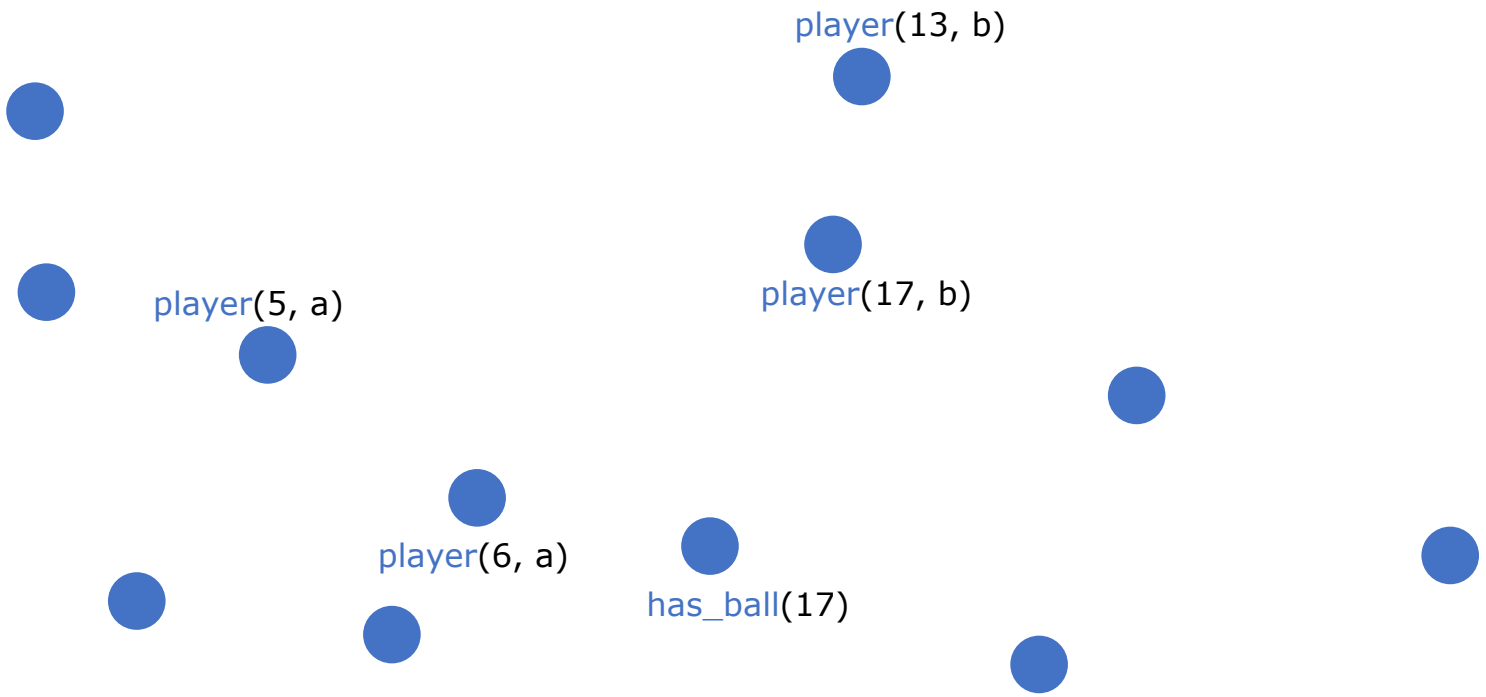
Experiment: Robots Kick/Pass/Steal

22 robot soccer players

player(Number, Team)

kick only if has ball

pass only to a teammate



Experiment: Robots Kick/Pass/Steal

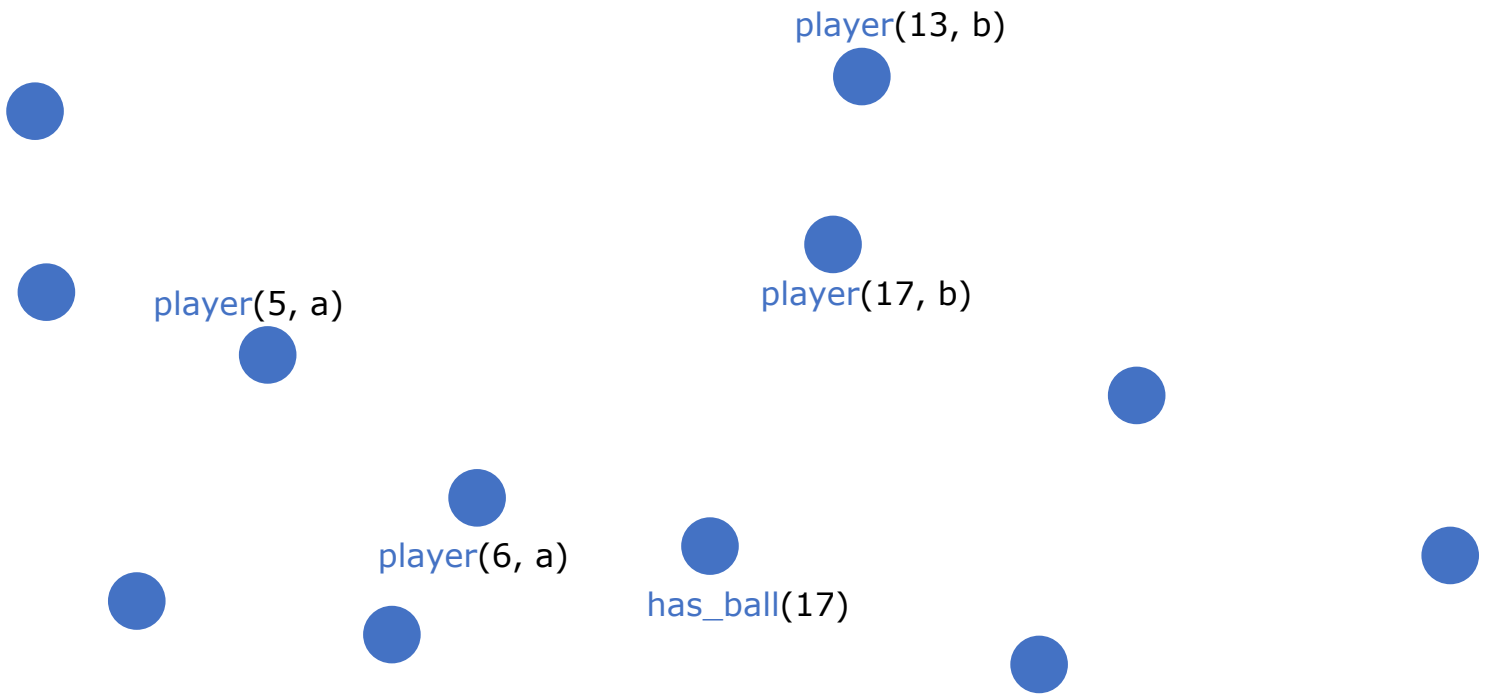
22 robot soccer players

player(Number, Team)

kick only if has ball

pass only to a teammate

steal

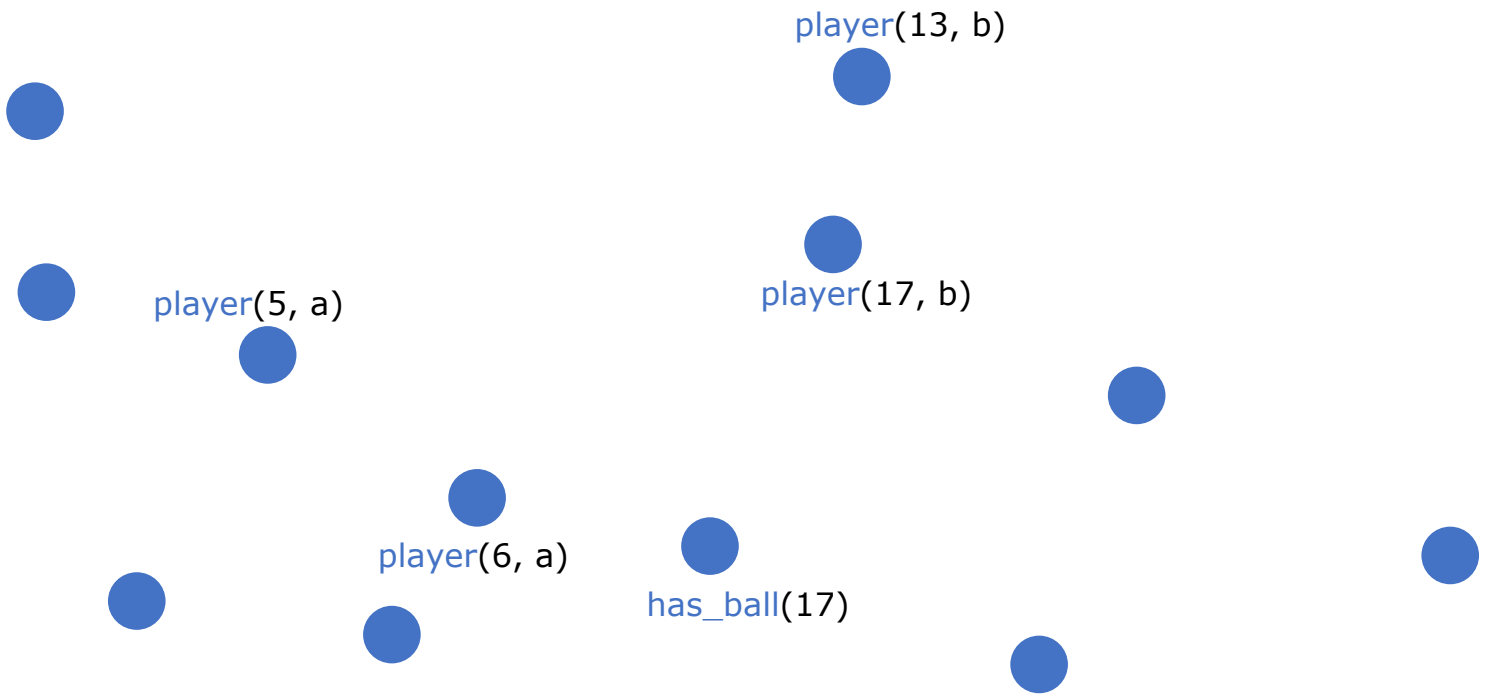


Experiment: Robots Kick/Pass/Steal

22 robot soccer players

player(Number, Team)

kick	only if has ball
pass	only to a teammate
steal	only from opponent

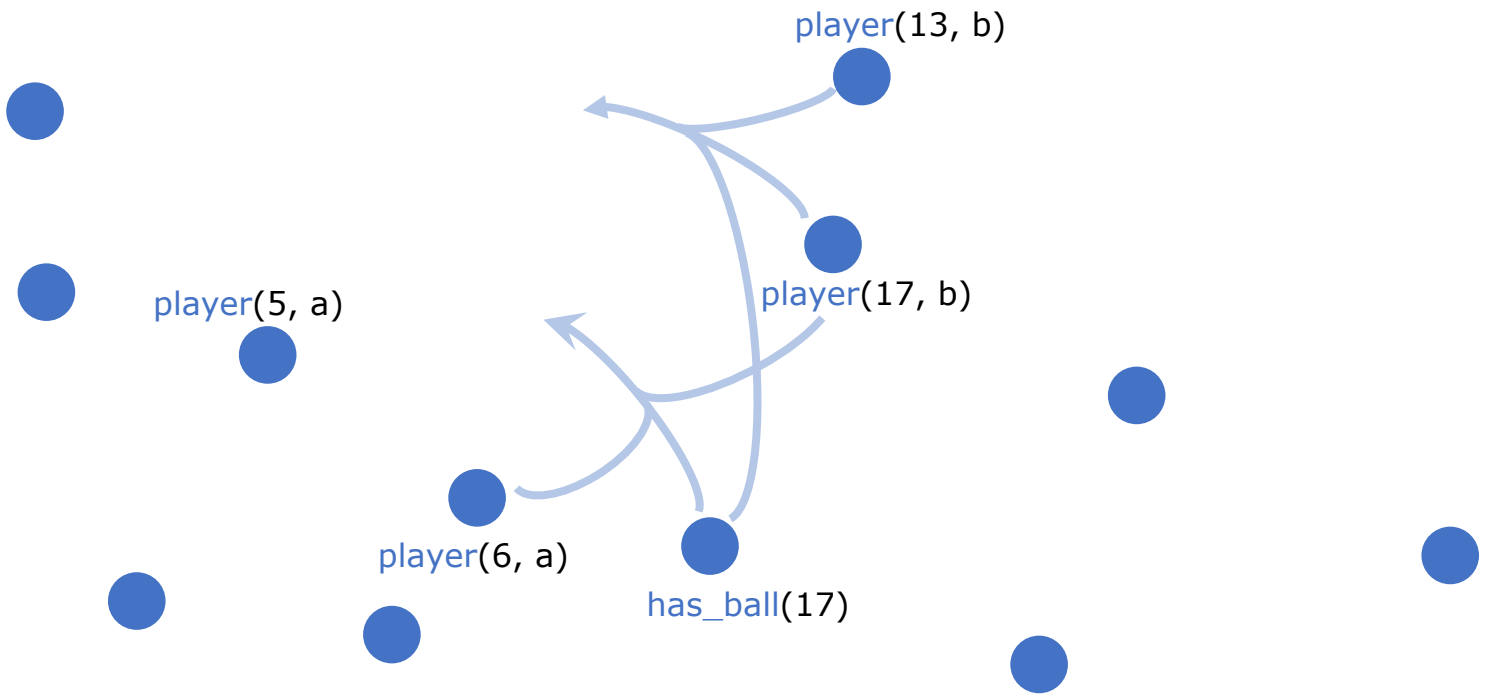


Experiment: Robots Kick/Pass/Steal

22 robot soccer players

player(Number, Team)

kick	only if has ball
pass	only to a teammate
steal	only from opponent

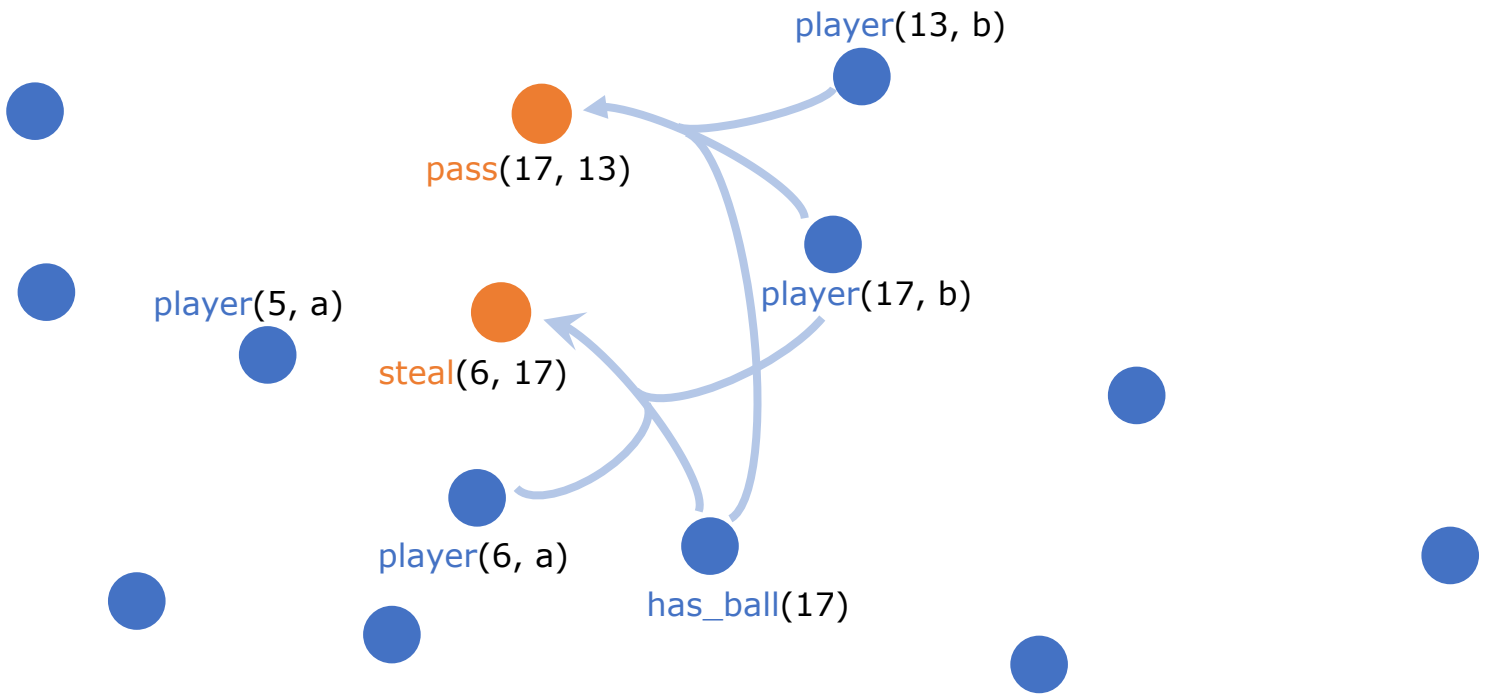


Experiment: Robots Kick/Pass/Steal

22 robot soccer players

player(Number, Team)

kick	only if has ball
pass	only to a teammate
steal	only from opponent

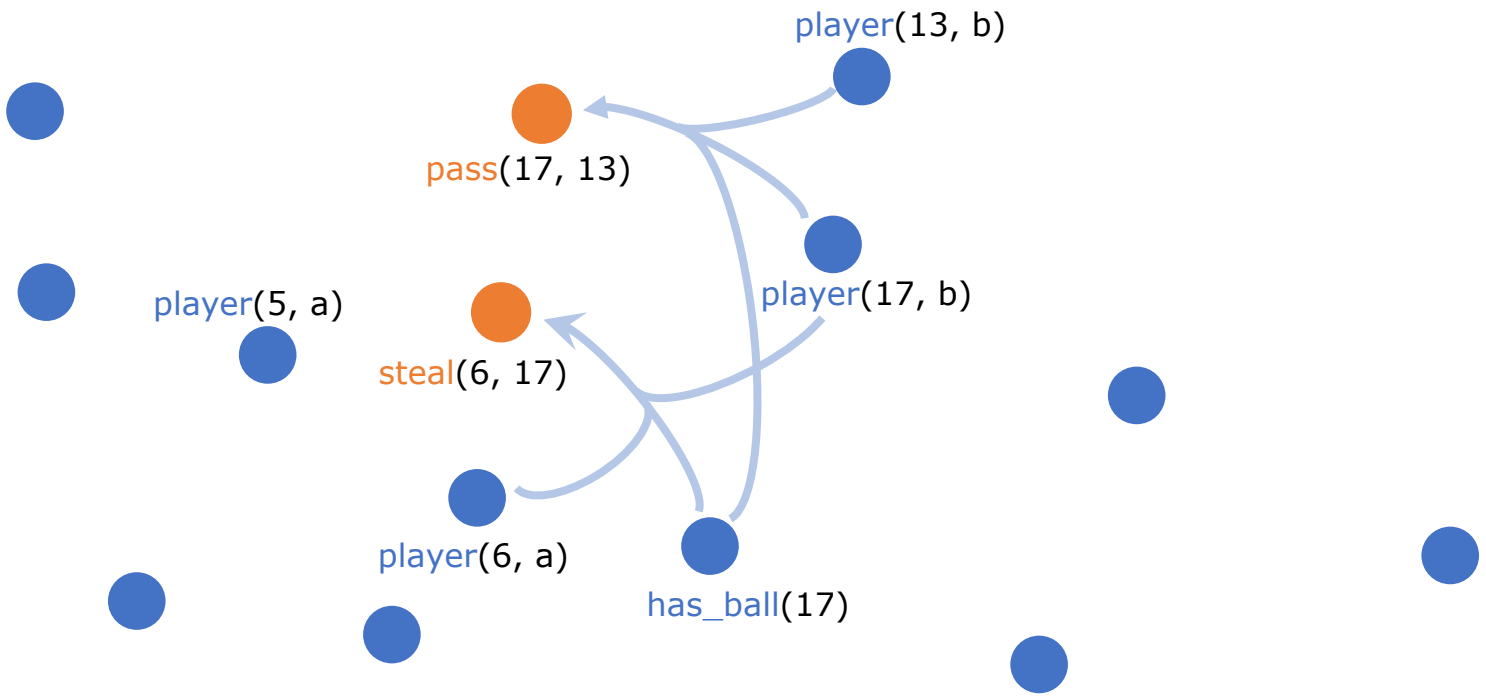


Experiment: Robots Kick/Pass/Steal

22 robot soccer players

player(Number, Team)

kick	only if has ball
pass	only to a teammate
steal	only from opponent

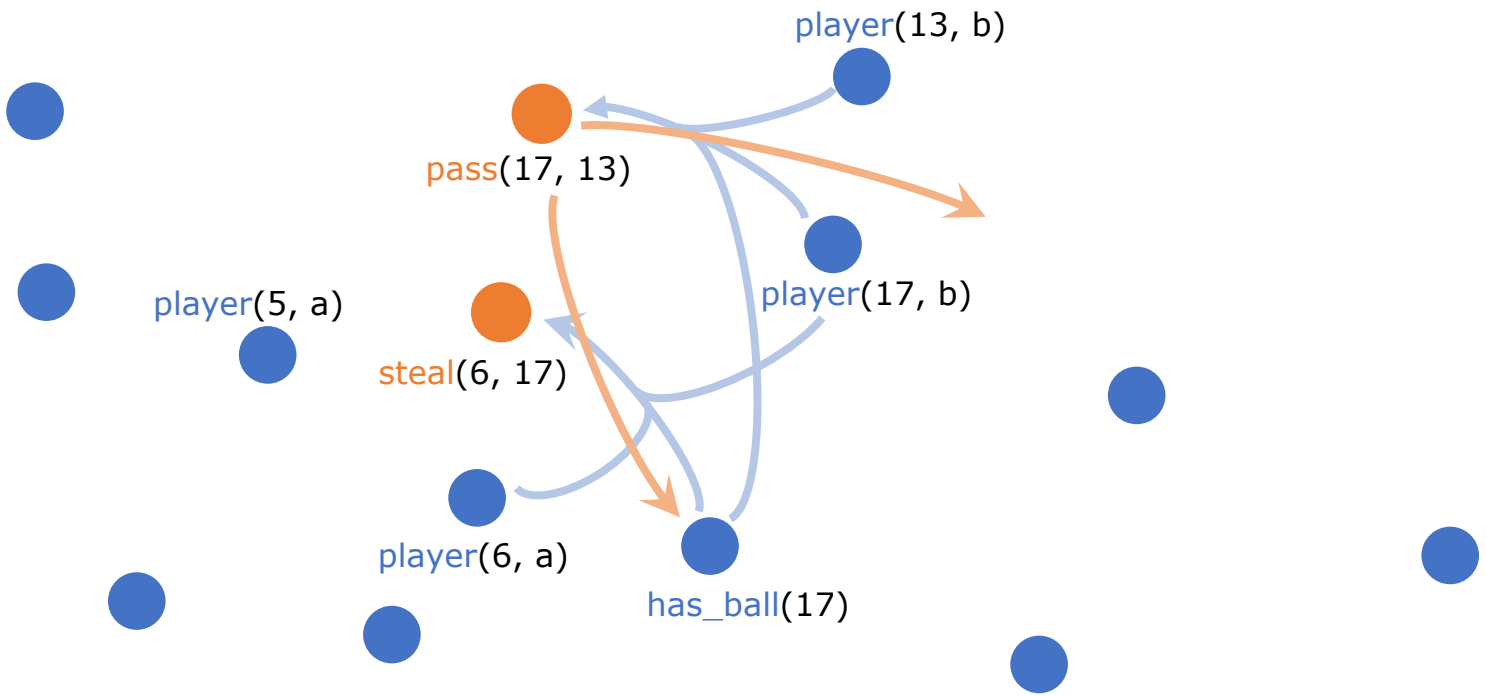


Experiment: Robots Kick/Pass/Steal

22 robot soccer players

player(Number, Team)

kick	only if has ball
pass	only to a teammate
steal	only from opponent

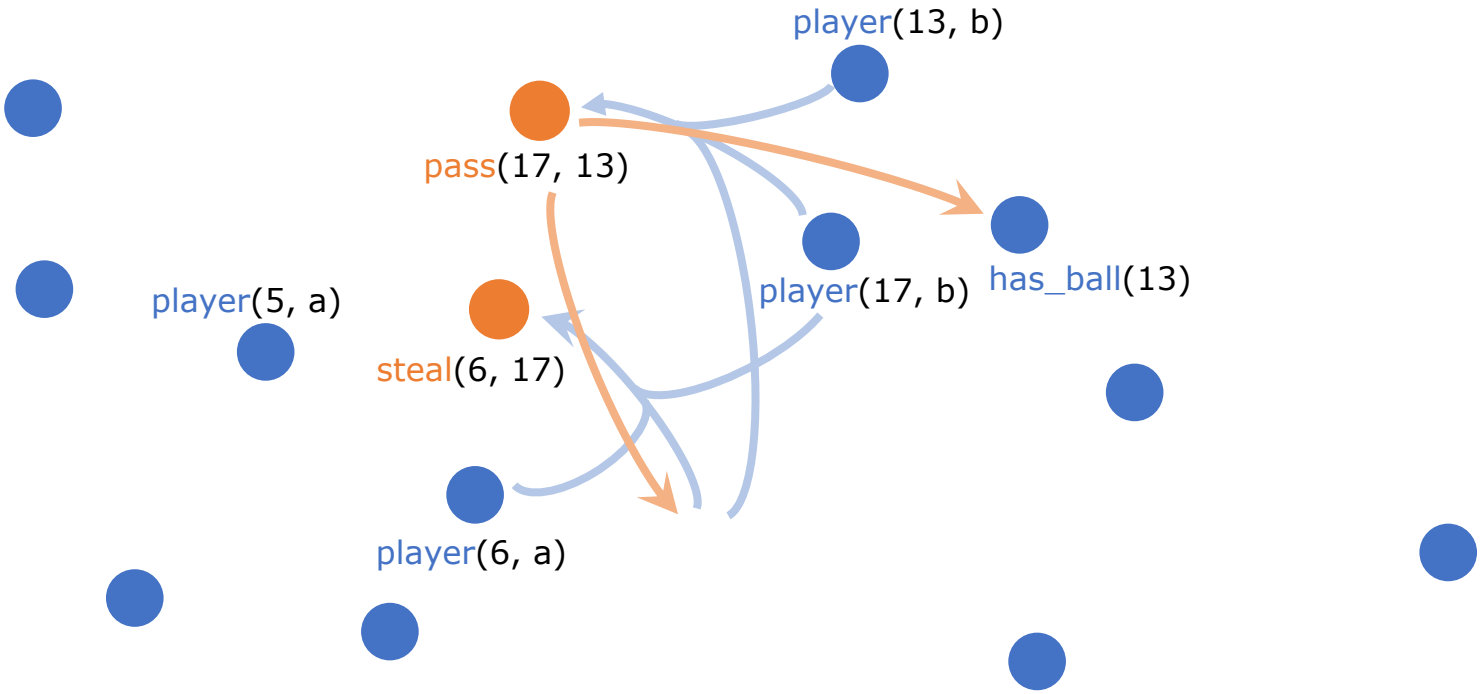


Experiment: Robots Kick/Pass/Steal

22 robot soccer players

player(Number, Team)

kick	only if has ball
pass	only to a teammate
steal	only from opponent

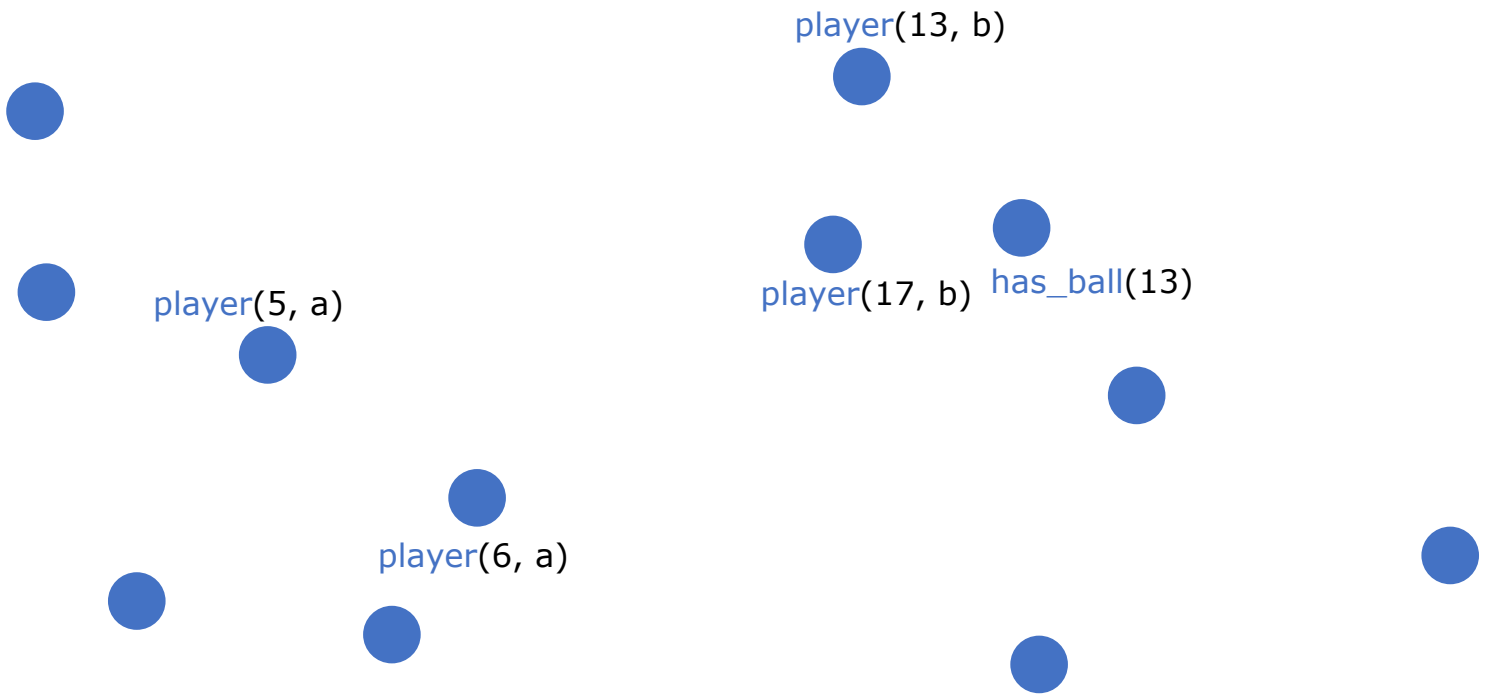


Experiment: Robots Kick/Pass/Steal

22 robot soccer players

player(Number, Team)

kick	only if has ball
pass	only to a teammate
steal	only from opponent

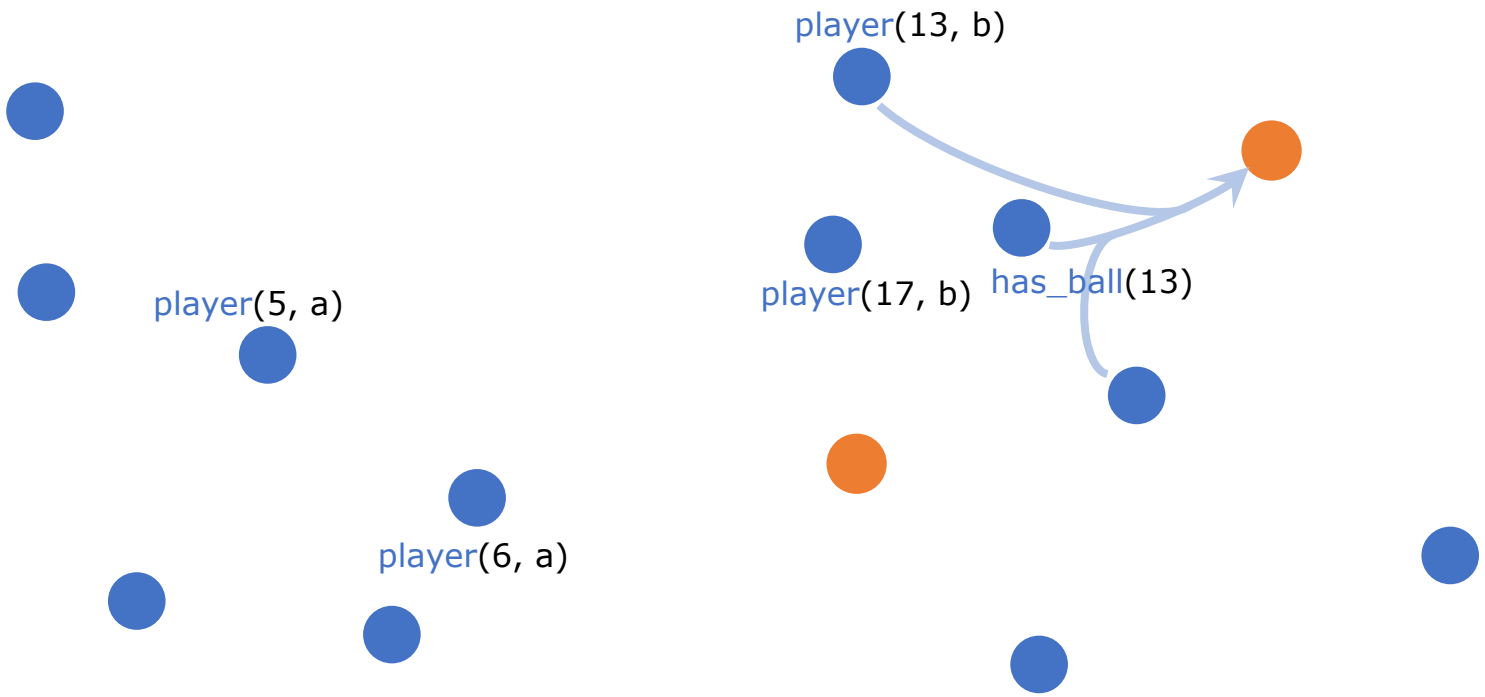


Experiment: Robots Kick/Pass/Steal

22 robot soccer players

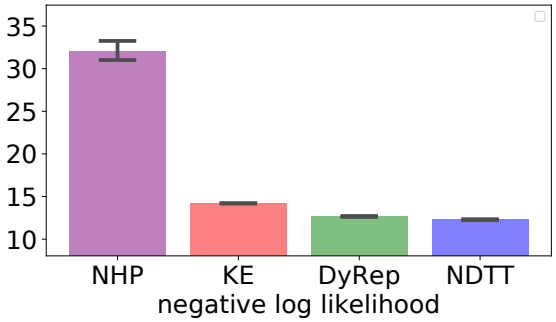
player(Number, Team)

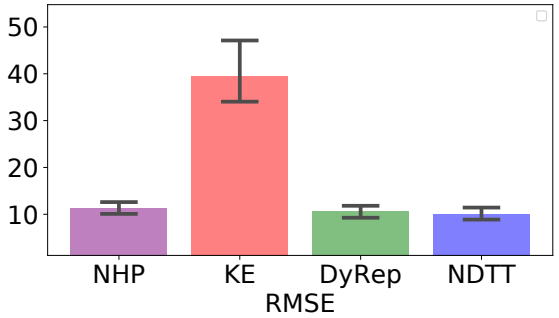
kick	only if has ball
pass	only to a teammate
steal	only from opponent

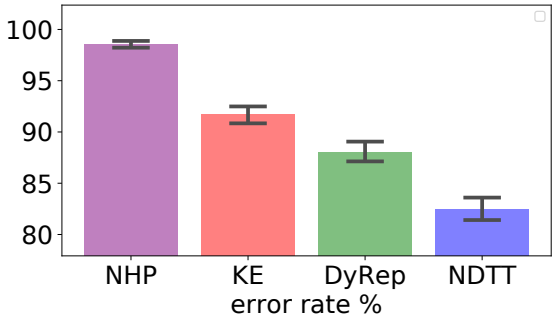


Results: NDTT > Competitors

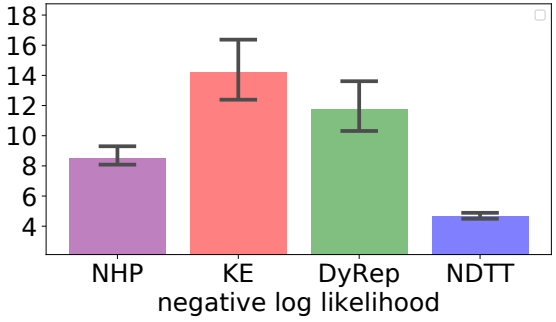
3 error metrics (in 3 columns): smaller is better

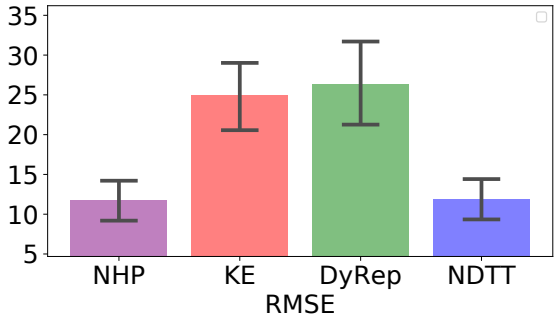


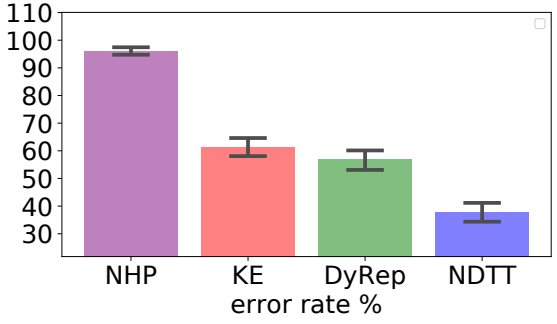




users watch TV programs

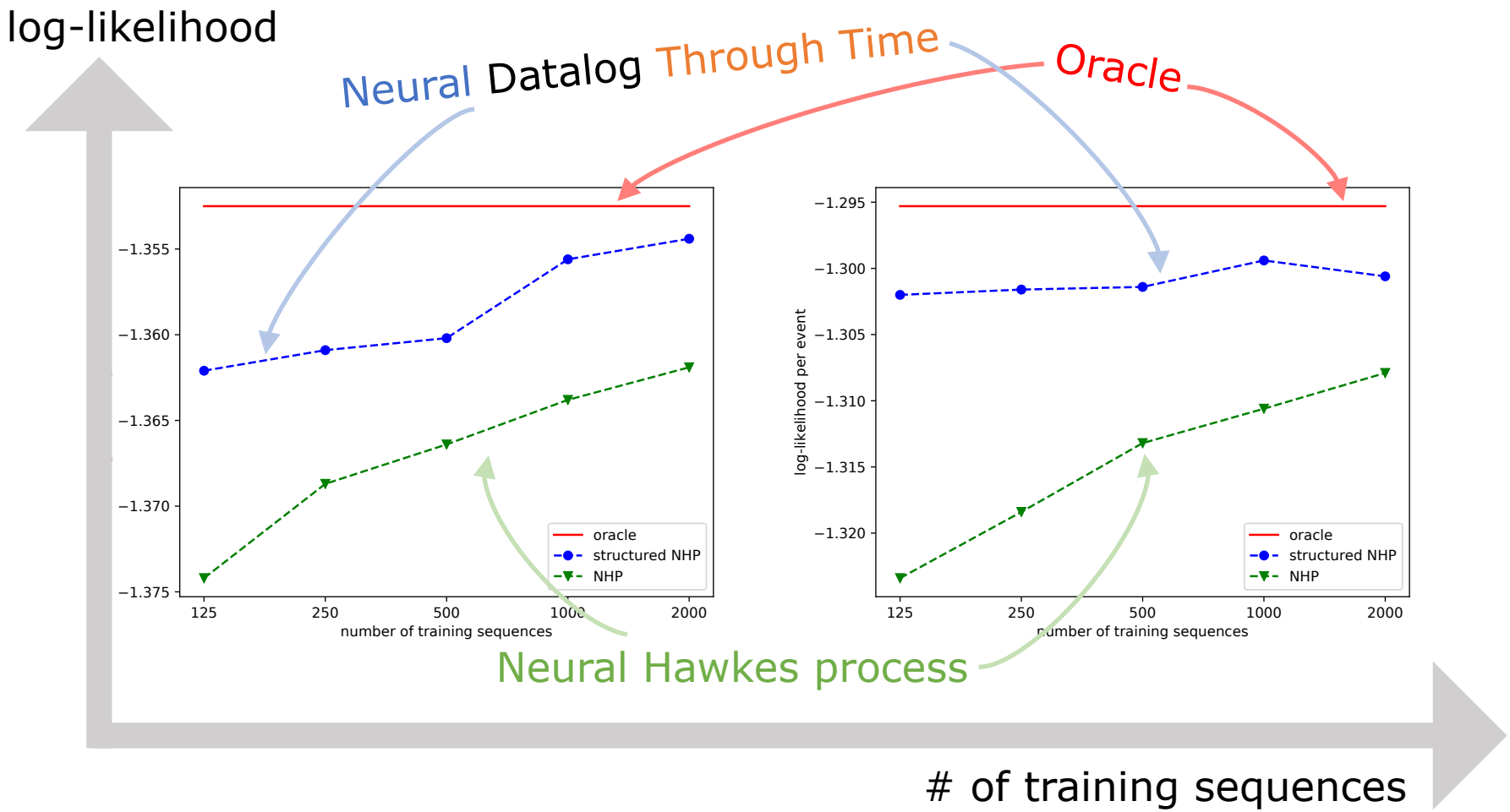






robots kick/pass/steal soccer ball

Good Generalization with Less Data



Summary: Deep Recurrent Net

Summary: Deep Recurrent Net

e.g., RNN
LSTM

discrete-time

Summary: Deep Recurrent Net

e.g., RNN discrete-time
LSTM
neural Hawkes process continuous-time

Summary: Deep Recurrent Net

hidden
system
state

e.g., RNN discrete-time
LSTM
neural Hawkes process continuous-time

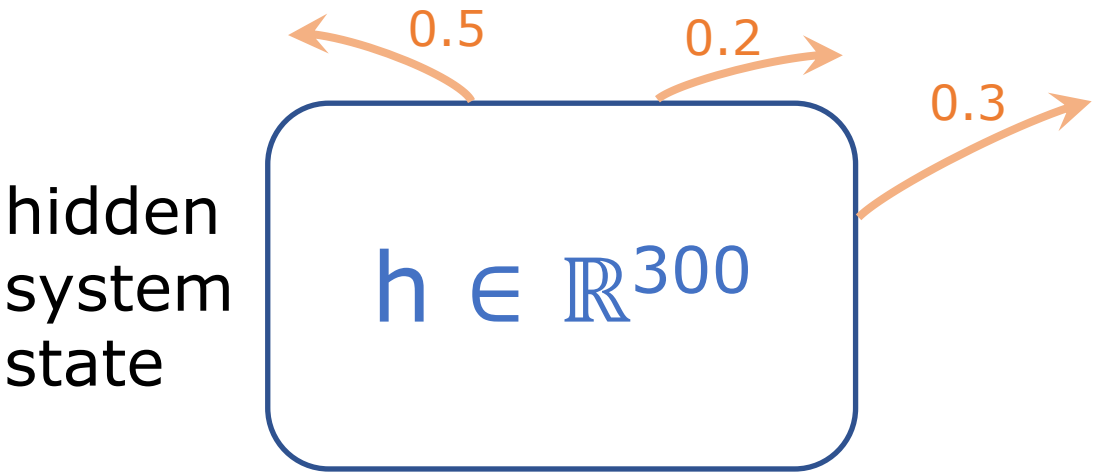
Summary: Deep Recurrent Net

hidden system state

$h \in \mathbb{R}^{300}$

e.g., RNN discrete-time
LSTM
neural Hawkes process continuous-time

Summary: Deep Recurrent Net

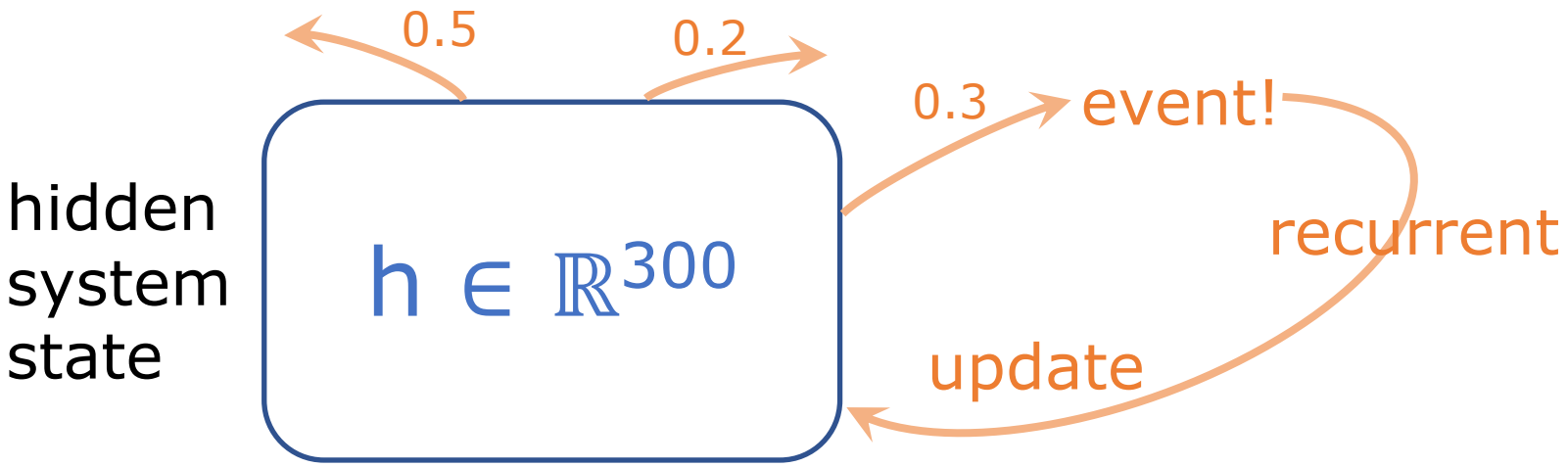


e.g., RNN discrete-time
LSTM
neural Hawkes process continuous-time

Summary: Deep Recurrent Net

e.g., RNN discrete-time
LSTM
neural Hawkes process continuous-time

Summary: Deep Recurrent Net



e.g., RNN discrete-time
LSTM
neural Hawkes process continuous-time

Summary: Logic ➔ Deep Recurrent Net

hidden system state

Summary: Logic ➔ Deep Recurrent Net

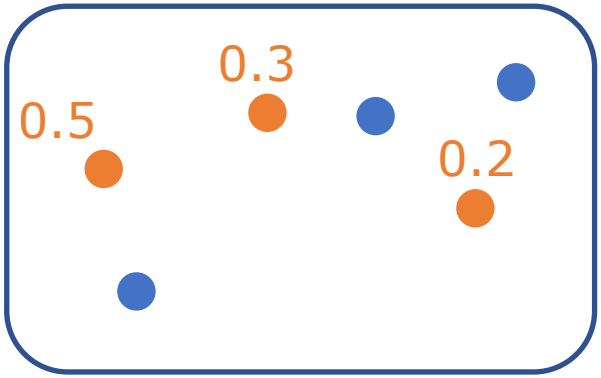
distributed

hidden
system
state

Summary: Logic ➔ Deep Recurrent Net

distributed

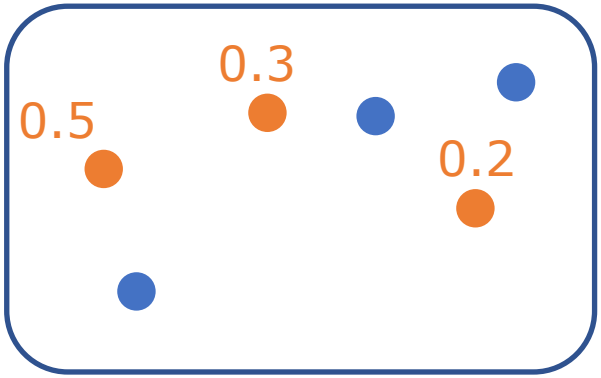
hidden
system
state



Summary: Logic ➔ Deep Recurrent Net

distributed

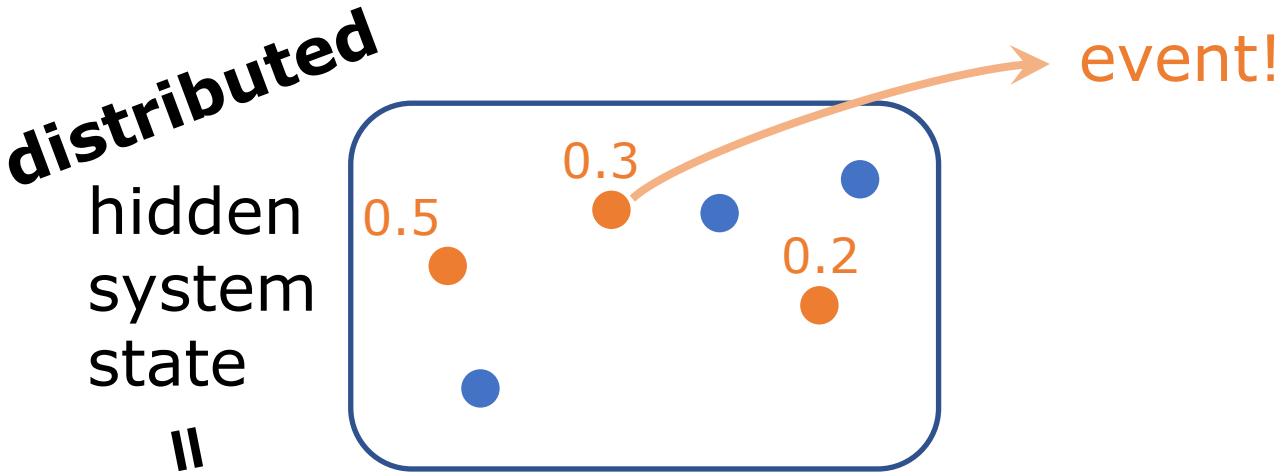
hidden
system
state



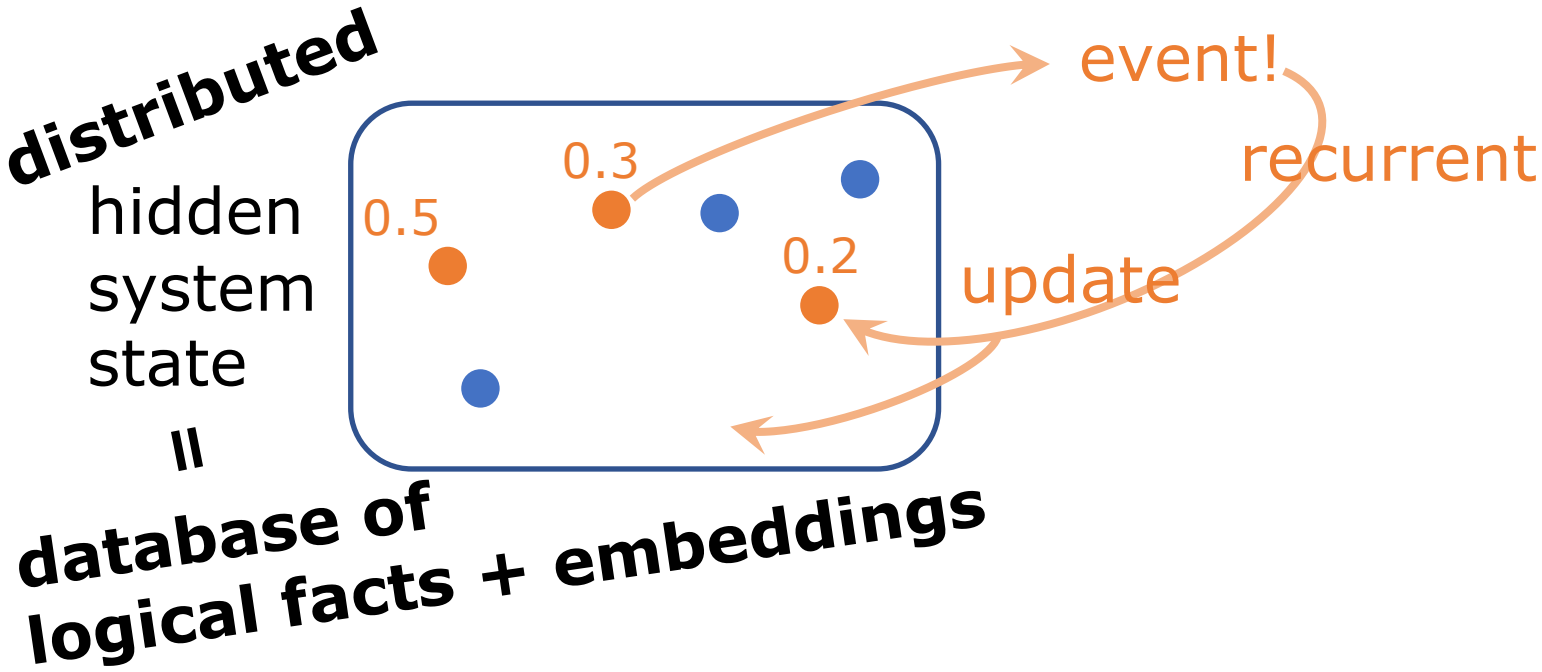
II

database of
logical facts + embeddings

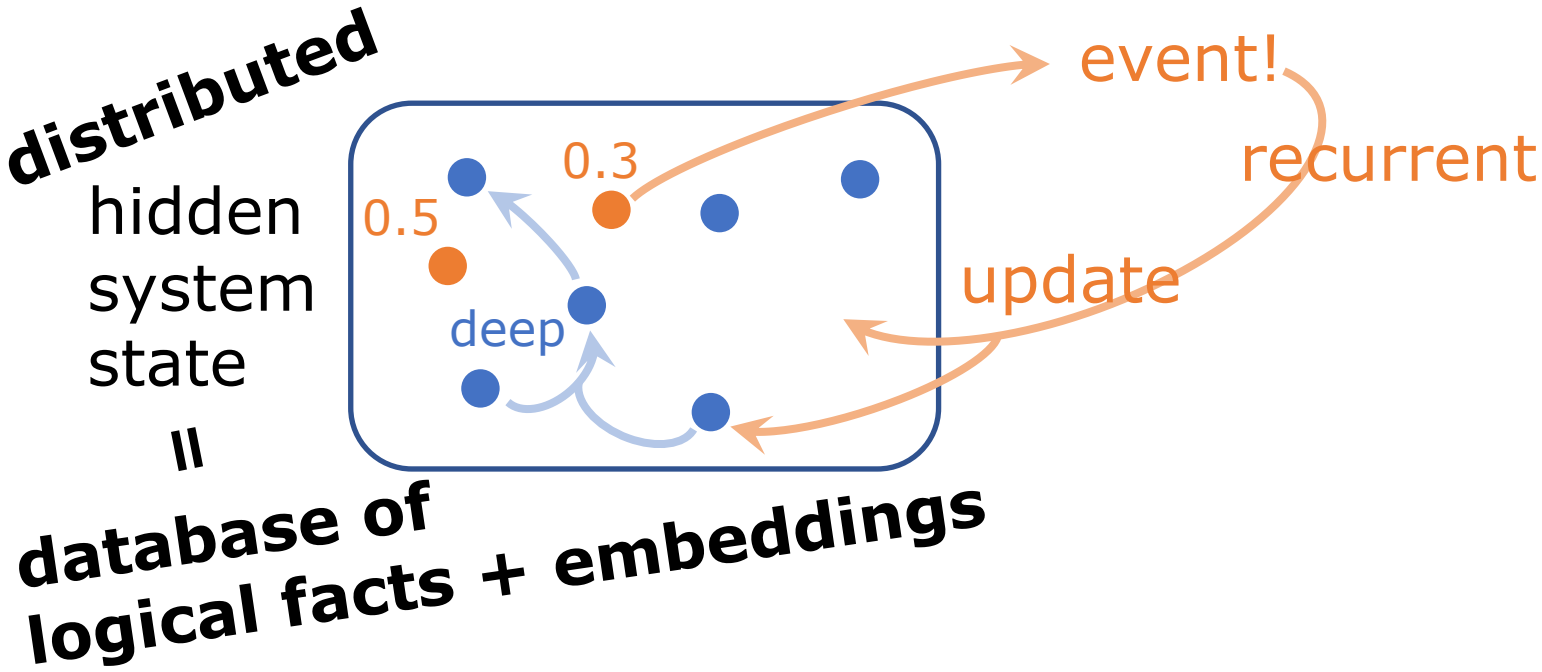
Summary: Logic ➔ Deep Recurrent Net



Summary: Logic ➔ Deep Recurrent Net

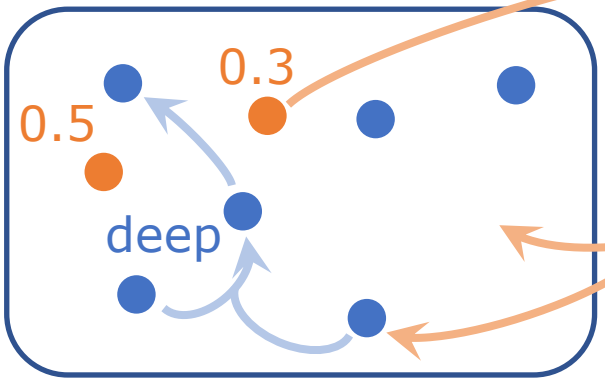


Summary: Logic ➔ Deep Recurrent Net



Summary: Logic ➔ Deep Recurrent Net

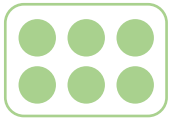
distributed
hidden
system
state
=
database of
logical facts + embeddings



event!
recurrent
update

update derived from
small rule set +
learned low-dim matrices

new fact :- old fact $_1$, …



new fact ← event, …

! old fact ← event, …

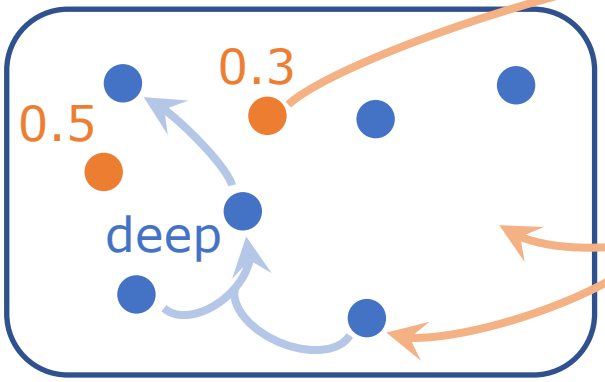
Summary: Logic → Deep Recurrent Net

distributed

hidden
system
state

=

database of logical facts + embeddings



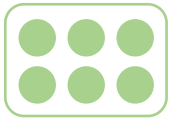
event!

recurrent

update

update derived from small rule set + learned low-dim matrices

new fact :- old fact $_1$, …



new fact ← event, …

! old fact ← event, …

try our code in your domain!

http://bburl/tpp-slides-p3
http://bburl/tpp-lab-p3

Neural Datalog Through Time

Bloomberg PhD Fellowship
Songyun Duan
Yujie Zha
Karan Uppal
NSF
ICLR reviewers
Rakshit Trivedi
ICML reviewers
MARCC
Hongteng Xu
JHU-CLSP
Argo Lab
NVIDIA

Thank You

Any Questions?

Thank You

http://bburl/tpp-slides-p3

http://bburl/tpp-lab-p3