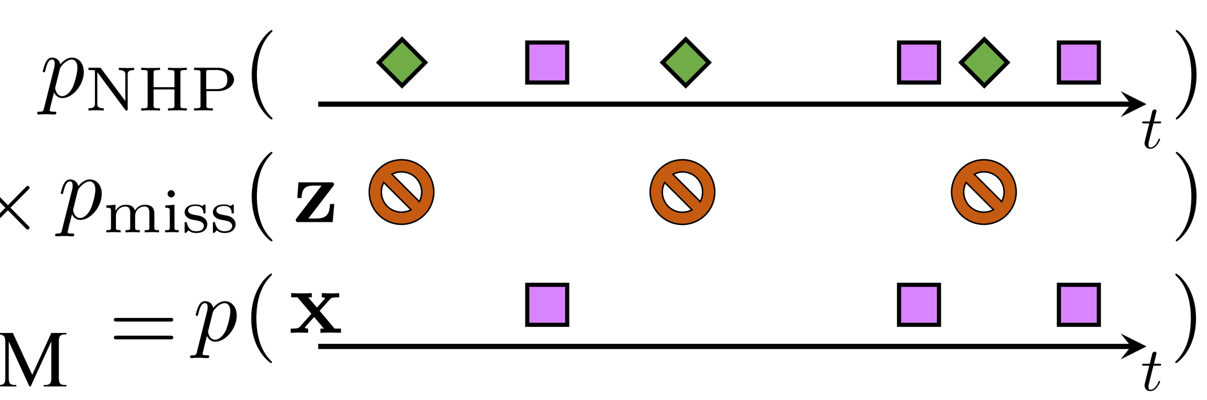
Neural Hawkes Particle Smoothing:

Imputing Missing Events in Continuous-Time Event Streams

Hongyuan Mei[1], Guanghui Qin[2], Jason Eisner[1]
[1]Center for Language and Speech Processing, Department of Computer Science, Johns Hopkins University, USA
[2]Department of Physics, Peking University, China

Overview

Neural Hawkes process (NHP: Mei & Eisner, NeurIPS 2017)
Missingness mechanism that determines missing events $\mathbf{z}$
$p(\mathbf{z} \mid \mathbf{x})$: **What / When / How-Many missing events?**
Why? Impute past to predict future; train with Monte Carlo EM

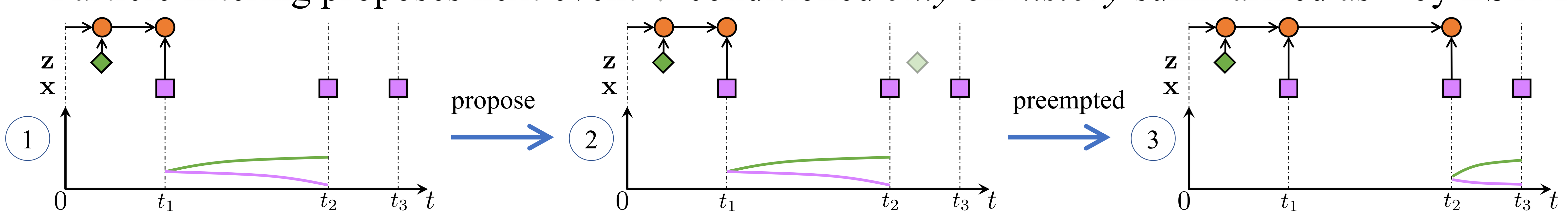


Sequential Monte Carlo

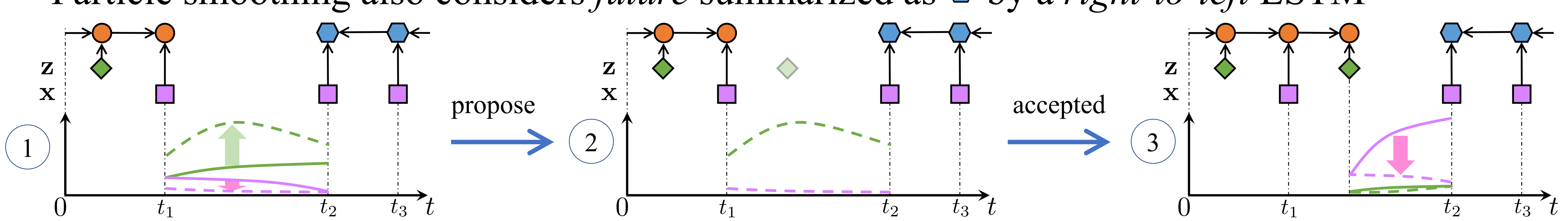
Draw $\mathbf{z}_1, \ldots, \mathbf{z}_M$ from a proposal distribution $q(\mathbf{z} \mid \mathbf{x})$ and weight them $w \propto p(\mathbf{z} \mid \mathbf{x}) / q(\mathbf{z} \mid \mathbf{x})$

Example: stochastically impute a taxi's pick-up events ◆ given its observed drop-off events ■. Below shows one sequential step, which determines the next event after ■ at time t_1 ---either an unobserved event at time $\in (t_1, t_2)$ or the next observed event at t_2.

- Particle filtering proposes next event ◇ conditioned *only* on *history* summarized as ● by LSTM



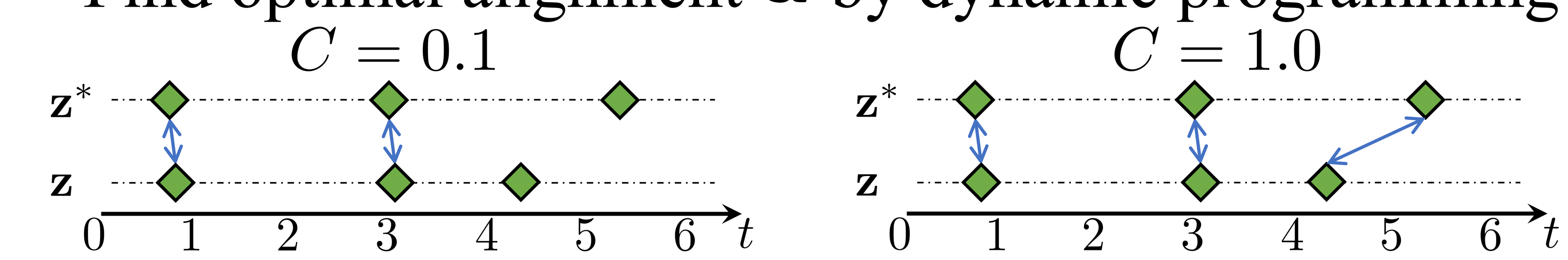
- Particle smoothing also considers *future* summarized as ⬢ by a *right-to-left* LSTM



Minimum Bayes Risk Decoding

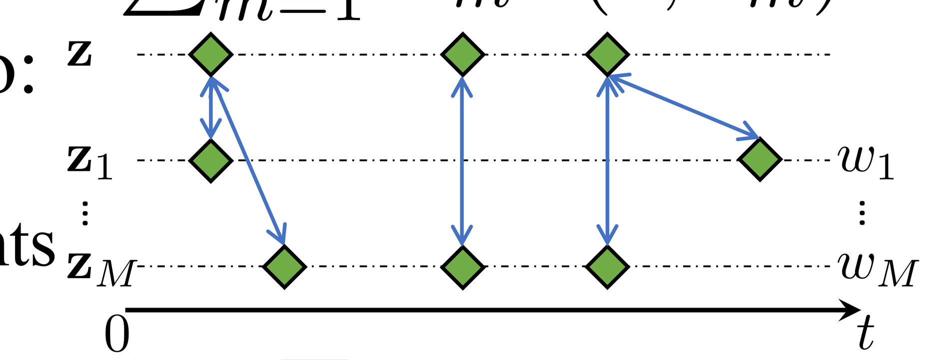
Define optimal transport distance $L(\mathbf{z}, \mathbf{z}^*)$

- Aligning two events in $\mathbf{z}$ and $\mathbf{z}^*$ has cost $|t - t^*|$
- An unaligned event in $\mathbf{z}$ or $\mathbf{z}^*$ has cost C
- Find optimal alignment $\mathbf{a}$ by dynamic programming

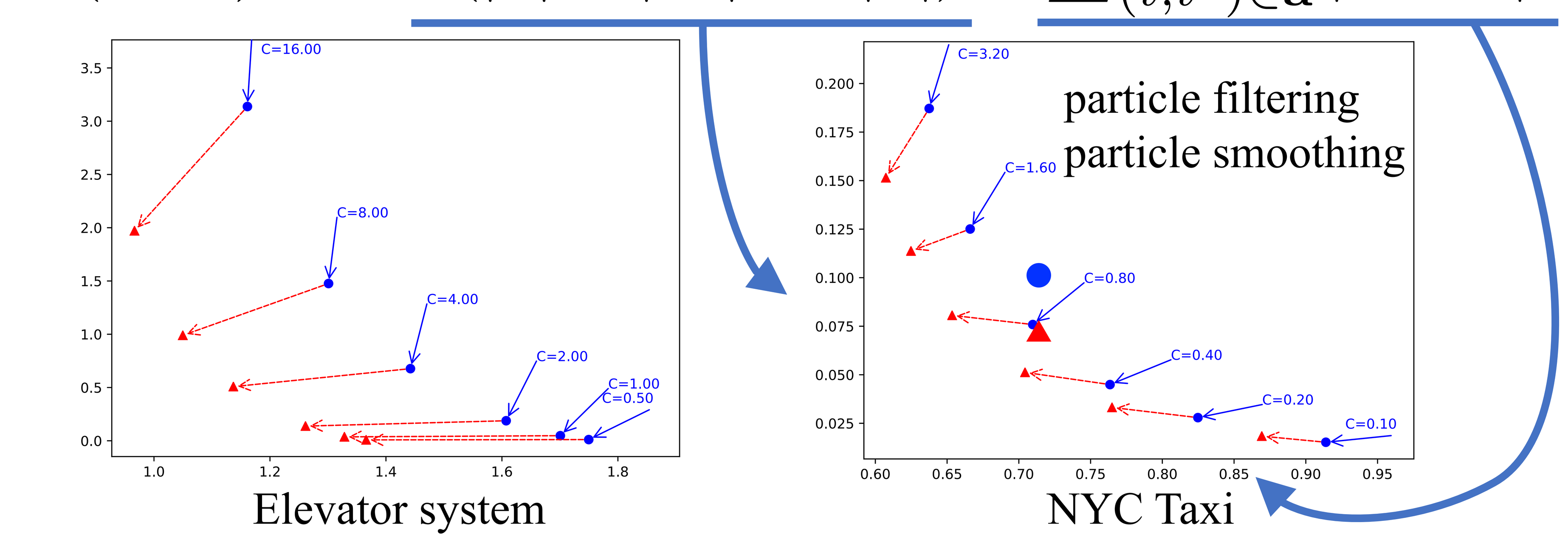


Seek $\mathbf{z}$ with small expected loss $\sum_{m=1}^{M} w_m L(\mathbf{z}, \mathbf{z}_m)$

- Until $\mathbf{z}$ does not change, do:
  - Align $\mathbf{z}$ to all particles
  - Move, delete and insert events



$$L(\mathbf{z}, \mathbf{z}^*) = C \cdot (|\mathbf{z}| + |\mathbf{z}^*| - 2|\mathbf{a}|) + \sum_{(t,t^*) \in \mathbf{a}} |t - t^*|$$



Finding: for each C, actual improvement → is always in the positive direction of the steepest improvement →

Training the Proposal Distribution (only for particle smoothing)

Minimize $\beta \underbrace{\mathrm{KL}(p \| q)}_{\text{inclusive}} + (1 - \beta) \underbrace{\mathrm{KL}(q \| p)}_{\text{exclusive}}$ between $q(\mathbf{z} \mid \mathbf{x})$ and $p(\mathbf{z} \mid \mathbf{x})$

- p includes missingness mechanism: don't propose what you know won't be missing!
- Inclusive KL: learn to propose every $\mathbf{z}$ that is probable under $p(\mathbf{z} \mid \mathbf{x})$
- Exclusive KL: learn to avoid proposing any $\mathbf{z}$ that is not probable under $p(\mathbf{z} \mid \mathbf{x})$

Does particle smoothing help (vs. filtering)?

Each point is a single gold seq, showing $\log q$ of proposing it under the two methods

Datasets:

- 10 synthetic (left)
- Elevator (mid)
- NYC taxi (right)