



Carnegie Mellon

School of Computer Science

Cut-And-Stitch: Efficient Parallel Learning of Linear Dynamical Systems on SMPs

Lei Li, Wenjie Fu, Fan Guo,
Todd C. Mowry, Christos Faloutsos
Computer Science Department
Carnegie Mellon University
leili@cs.cmu.edu

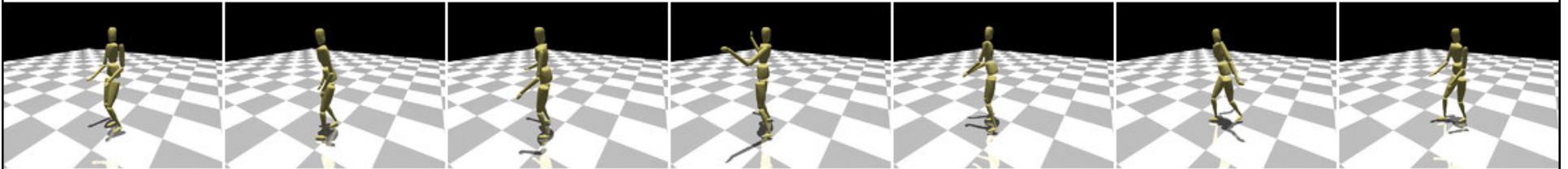


Outline

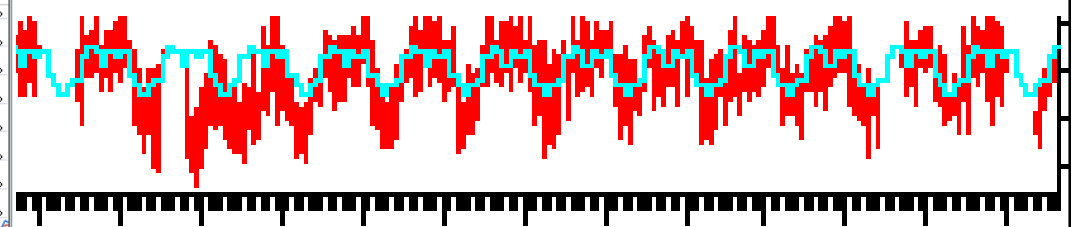
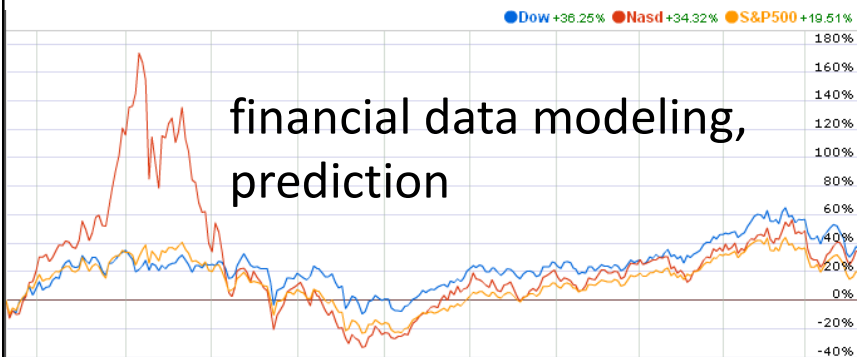
- Motivation
- Background
- Proposed Method: Cut-And-Stitch
- Experiments
- Conclusion



Motivation



Motion Capture, Human Motion Modeling



- Common tool: Linear Dynamical System (LDS)
- Challenge?
 - Learning LDS is slow

Need For Speed



Related Work

- Parallel mining of closed sequential patterns, [KDD 05, [S. Cong, J. Han, D. Padua](#)]
- Mining terabytes of data for frequent itemsets, [PPoPP 07, [G. Buehrer, S. Parthasarathy, S. Tatikonda, T. Kurc, J. Saltz](#)]
- Parallelize the discovery of frequent patterns in large graphs, [IPDPS 07, [S. Reinhardt, G. Karypis](#)]
- Parallel algorithms for SVM
 - Cascade SVM, [NIPS 04, [H. Graf, E. Cosatto, L. Bottou, I. Durdanovic, V. Vapnik](#)]
 - Parallel Mixture of SVMs, [NIPS 02, [R. Collobert, S. Bengio, Y. Bengio](#)]
 - PSVM (open src), [NIPS 07, [E. Chang, K. Zhu, H. Wang, H. Bai, J. Li, Z. Qiu, H. Cui](#)]
- Multicore Map-Reduce approach, [NIPS 06, [C. Chu, S. Kim, Y. Lin, Y. Yu, G. Bradski, A. Ng, K. Olukotun](#)]



Problem Definition

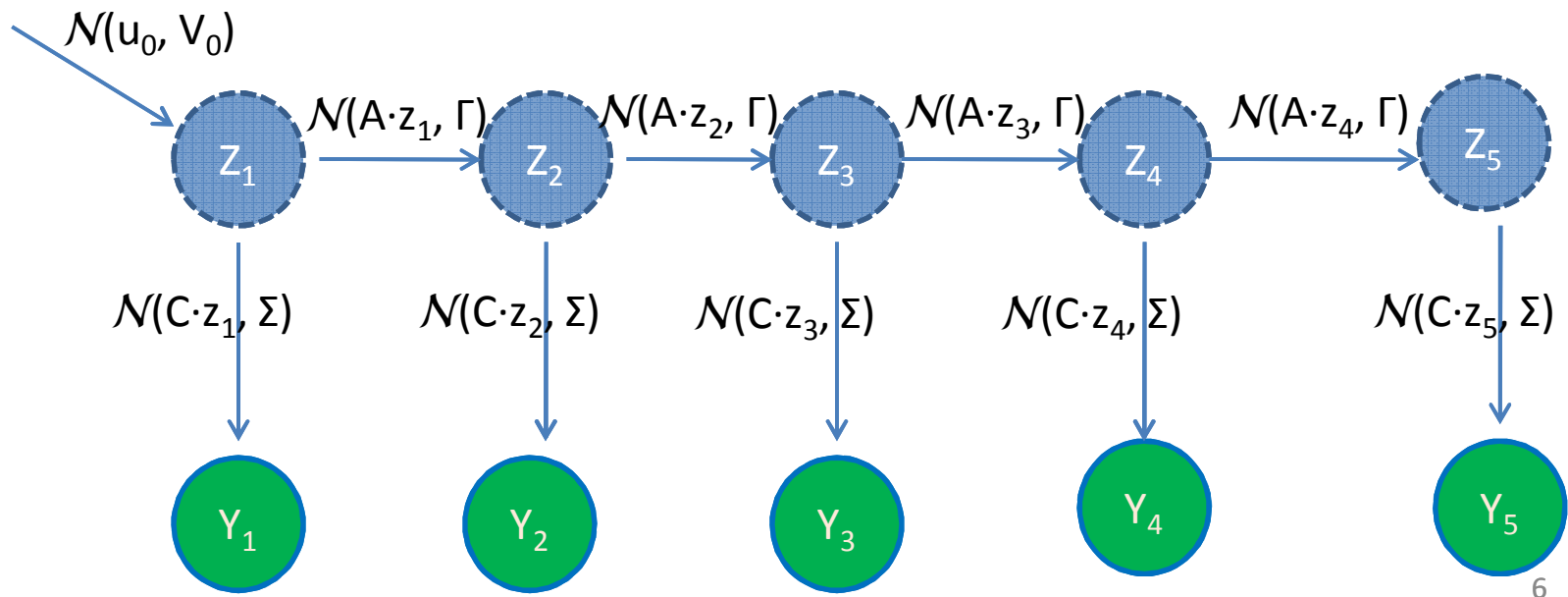
- Problem:
 - **Given** a sequence of data, **find** the best model parameters for Linear Dynamical System
- Traditional Method:
 - *Maximum Likelihood* Estimation via *Expectation-Maximization*(EM) algorithm
- Objective:
 - **Parallelize** the learning algorithm
- Assumption:
 - *shared memory architecture*



Linear Dynamical System

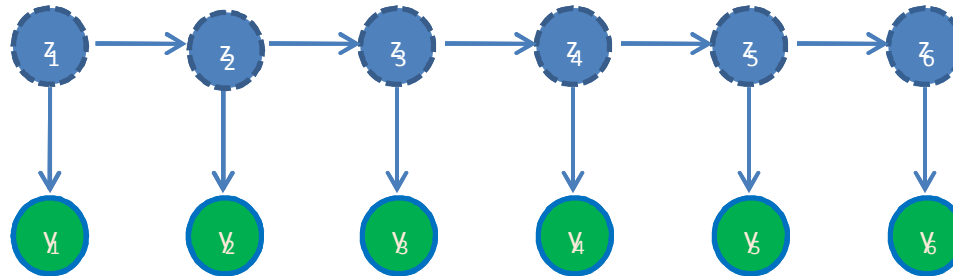
aka. Kalman Filter

- Parameters: $\theta = (u_0, V_0, A, \Gamma, C, \Sigma)$
- Observation: $Y_1 \dots Y_n$
- Hidden variables: $Z_1 \dots Z_n$



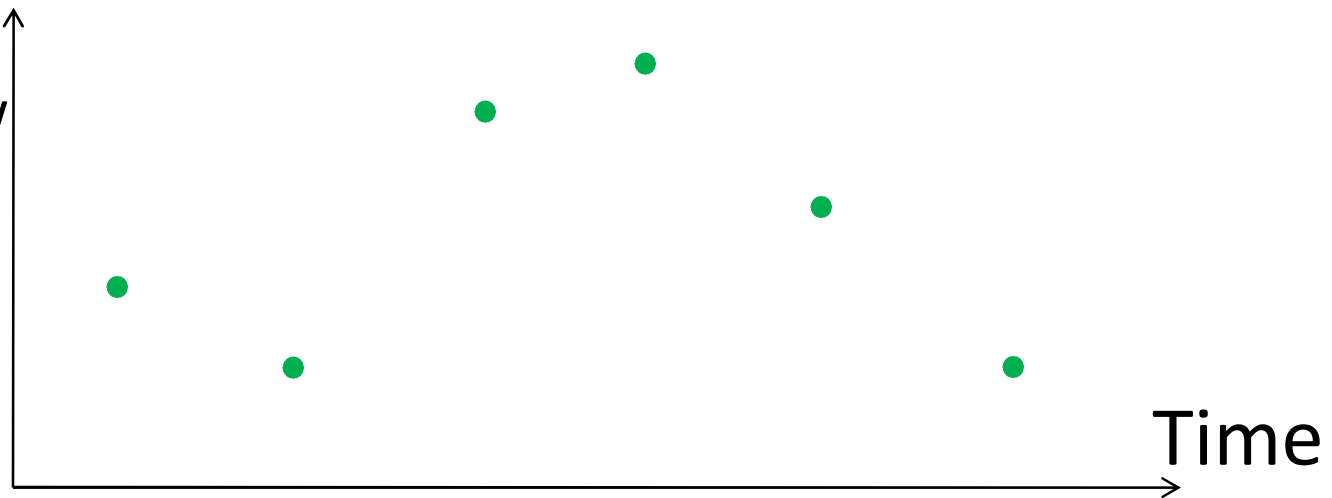


Example



given positions, estimate dynamics (i.e. params)

Position
of left elbow

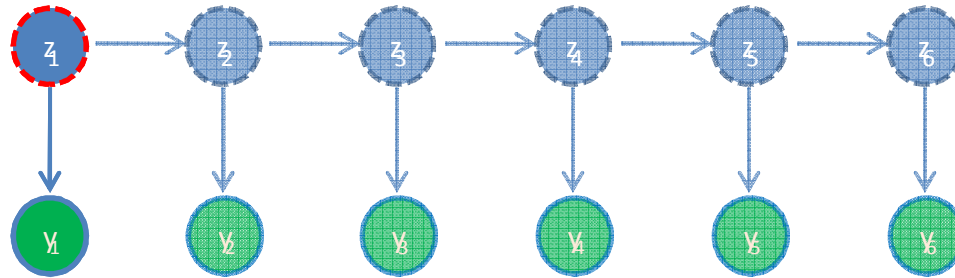




Traditional: How to learn LDS?



Sequential Learning (EM)



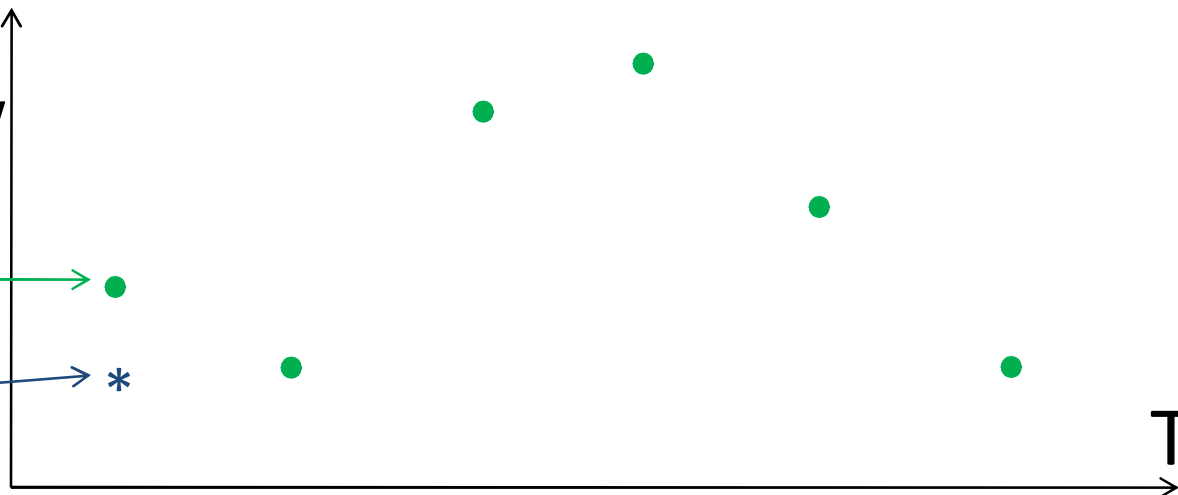
Compute $P(z_1 | y_1)$

Position
of left elbow

Measured → ●

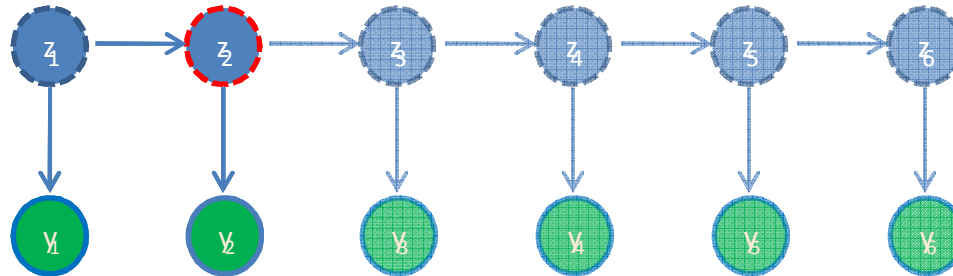
Estimated → *

Time





Sequential Learning (EM)



From $P(z_1 | y_1) \rightarrow$ Compute $P(z_2 | y_1, y_2)$

Position
of left elbow

Measured \rightarrow ●

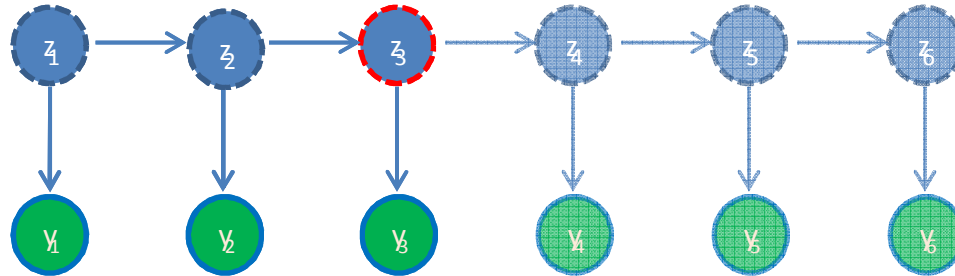
Estimated \rightarrow *

Intuition: z_2 may be close to z_1

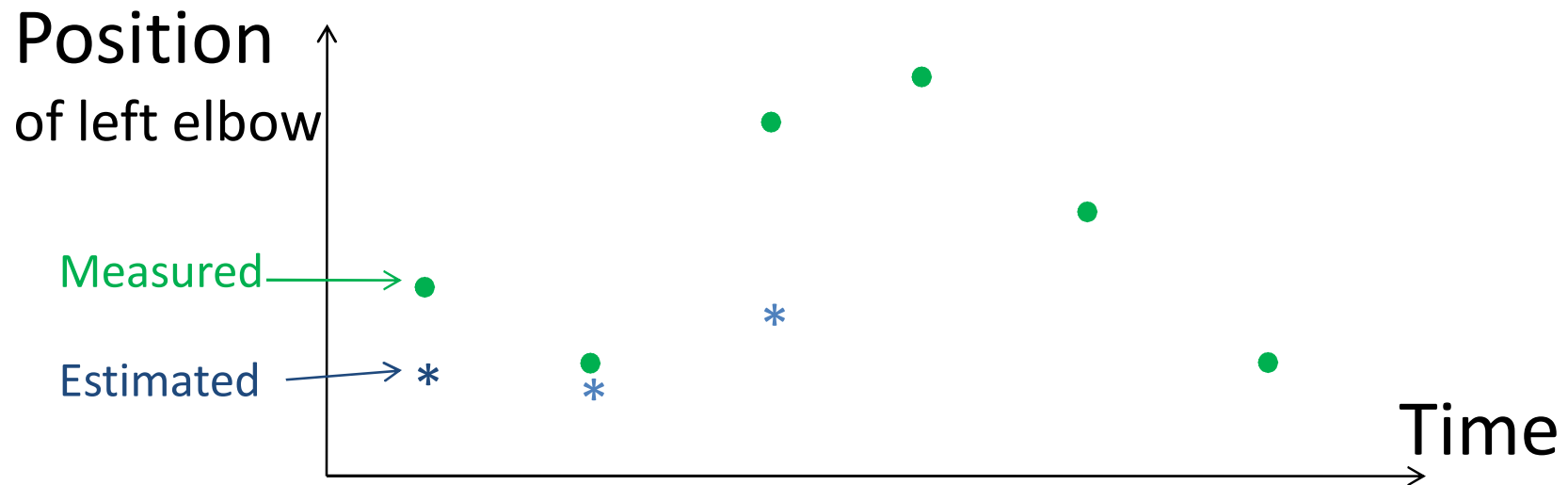
Time \rightarrow



Sequential Learning (EM)

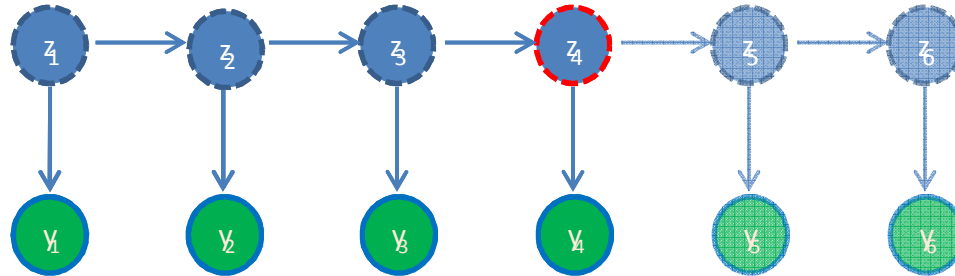


From $P(z_2 | y_1, y_2) \rightarrow$ Compute $P(z_3 | y_1, y_2, y_3)$

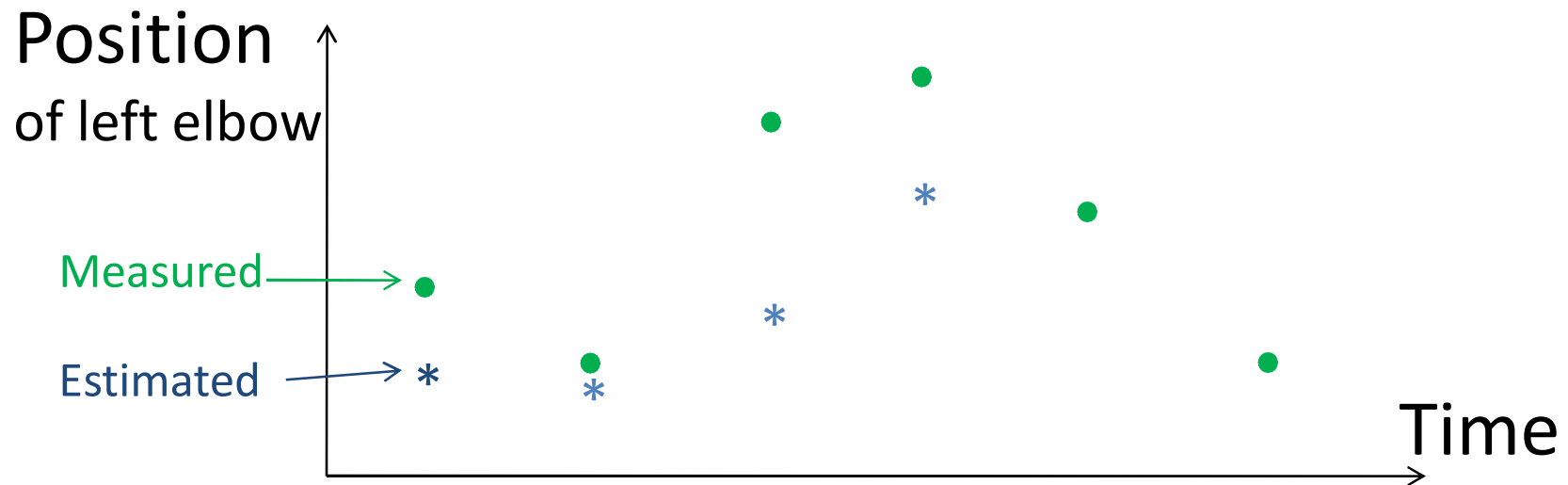




Sequential Learning (EM)

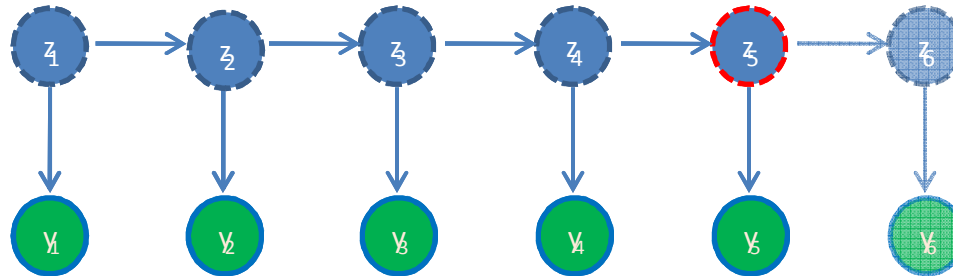


From $P(z_3 | y_1, y_2, y_3) \rightarrow$ Compute $P(z_4 | y_1, y_2, y_3, y_4)$

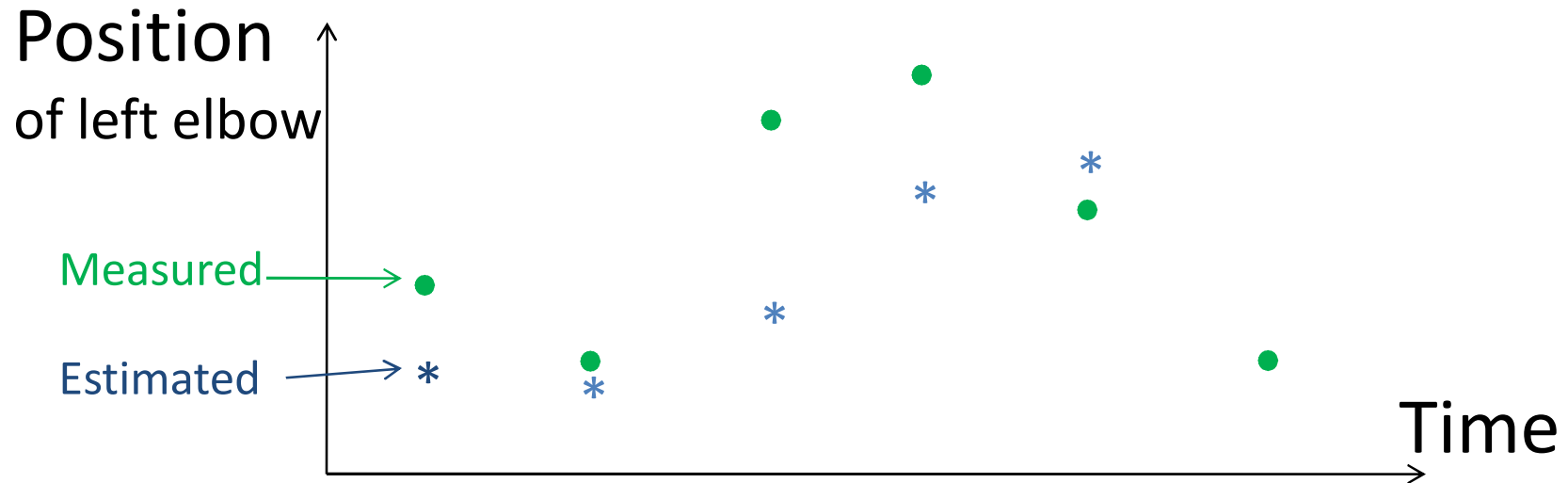




Sequential Learning (EM)

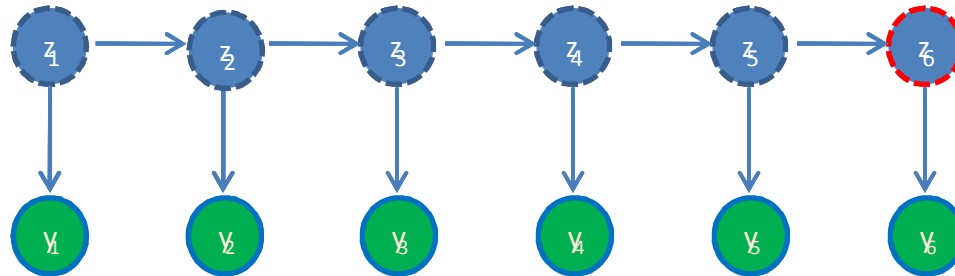


From $P(z_4 | y_1, y_2, y_3, y_4) \rightarrow$ Compute $P(z_5 | y_1, y_2, y_3, y_4, y_5)$

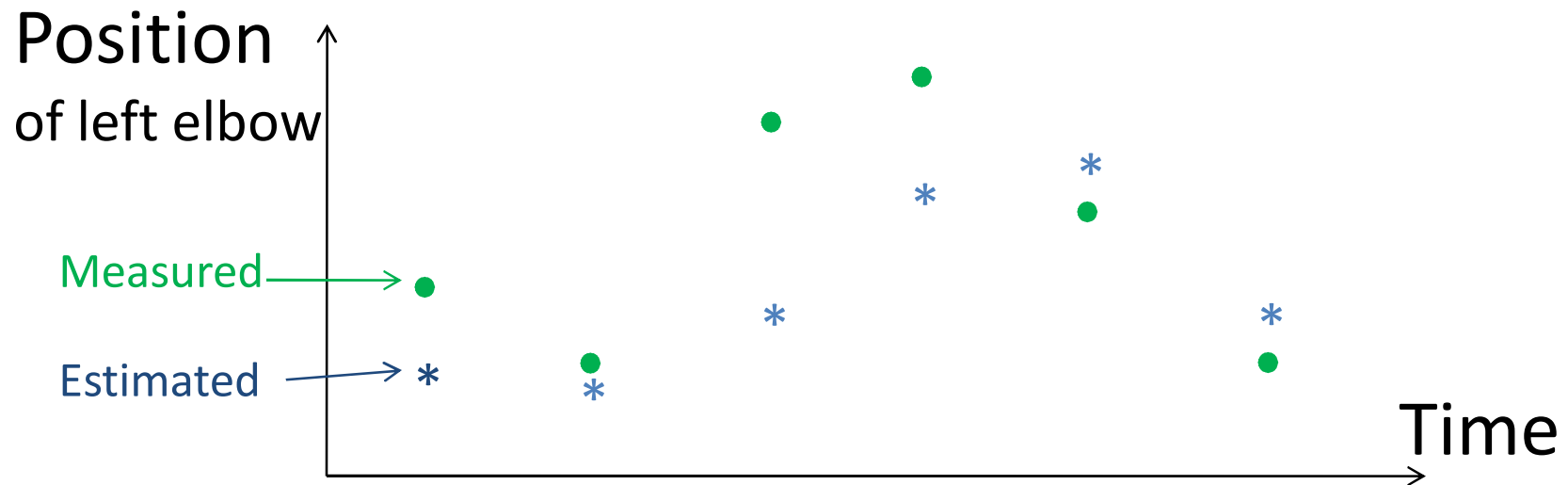




Sequential Learning (EM)

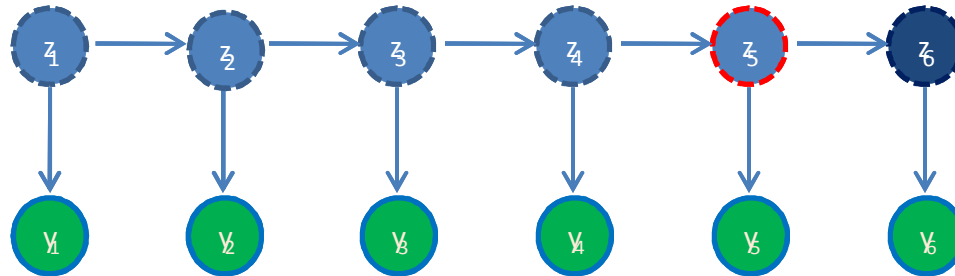


From $P(z_5 | y_1, y_2, y_3, y_4, y_5) \rightarrow$ Compute $P(z_6 | y_1, y_2, y_3, y_4, y_5, y_6)$





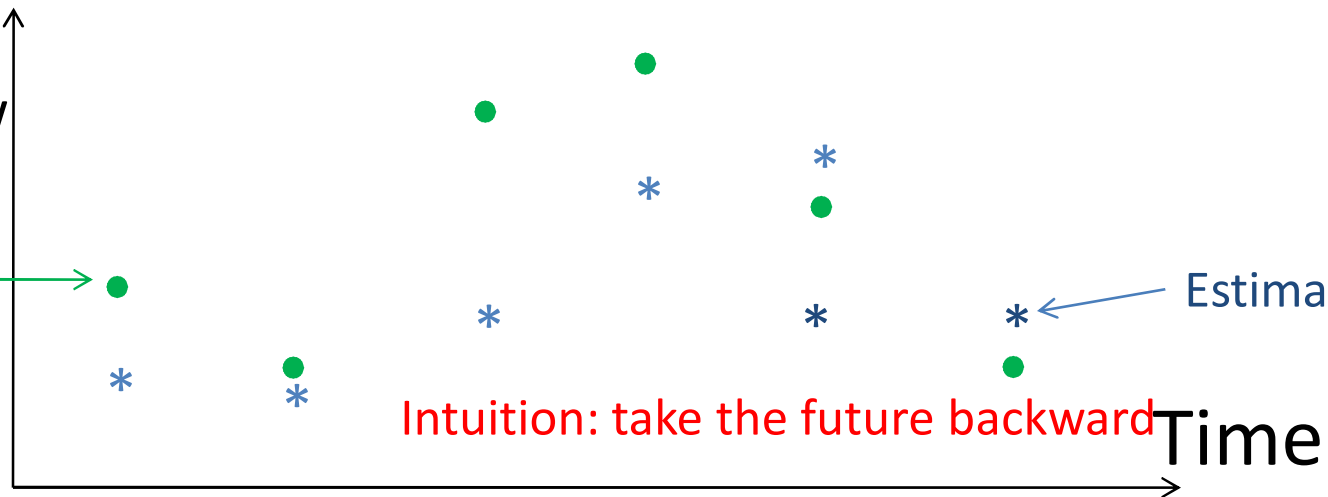
Sequential Learning (EM)



From $P(z_6 | y_1, y_2, y_3, y_4, y_5, y_6) \rightarrow$ Compute $P(z_5 | y_1, y_2, y_3, y_4, y_5, y_6)$

Position
of left elbow

Measured

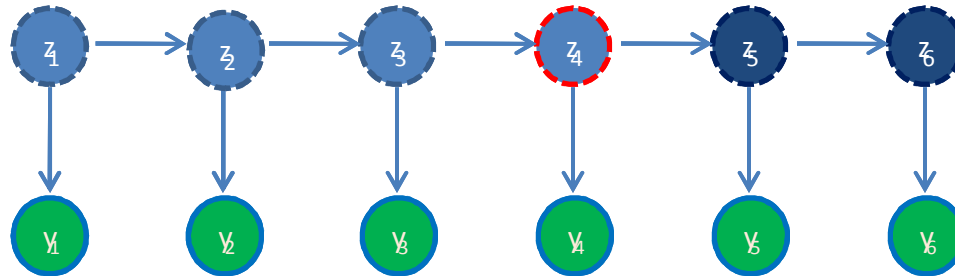


Intuition: take the future backward

Time



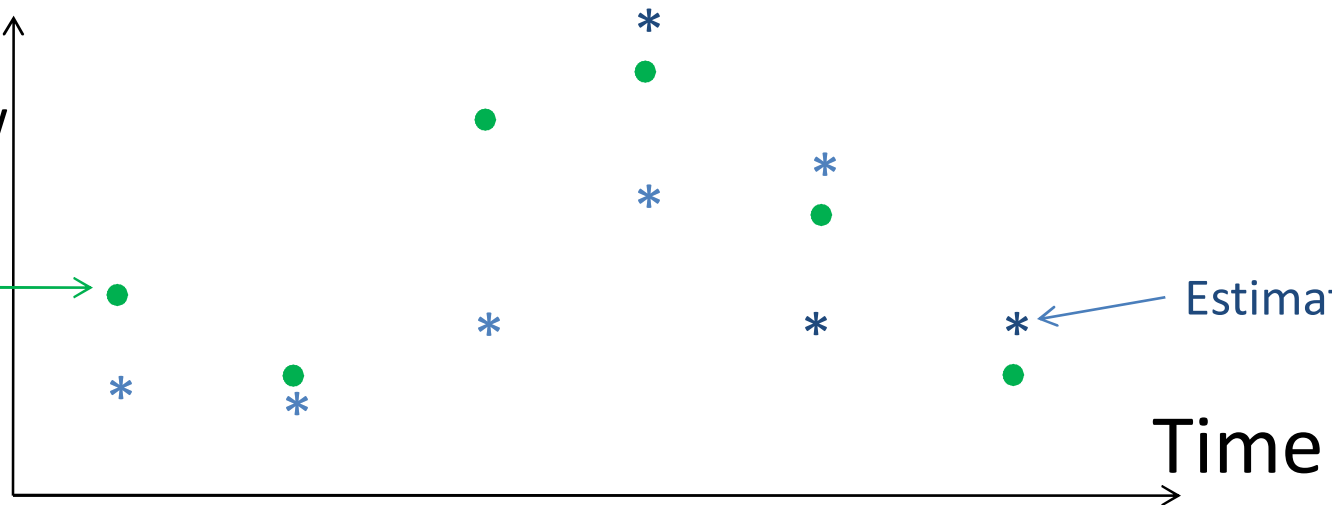
Sequential Learning (EM)



From $P(z_6 | y_1, y_2, y_3, y_4, y_5, y_6) \rightarrow$ Compute $P(z_4 | y_1, y_2, y_3, y_4, y_5, y_6)$

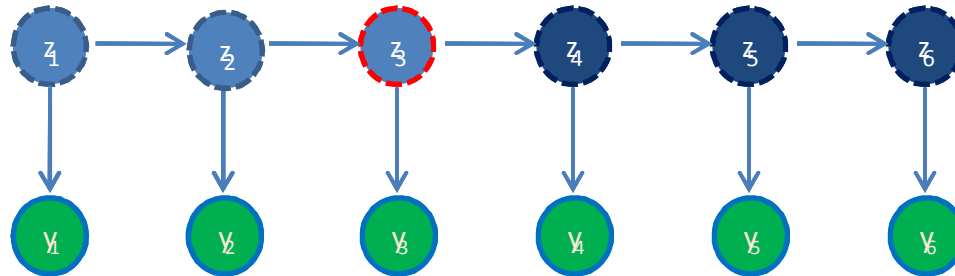
Position
of left elbow

Measured





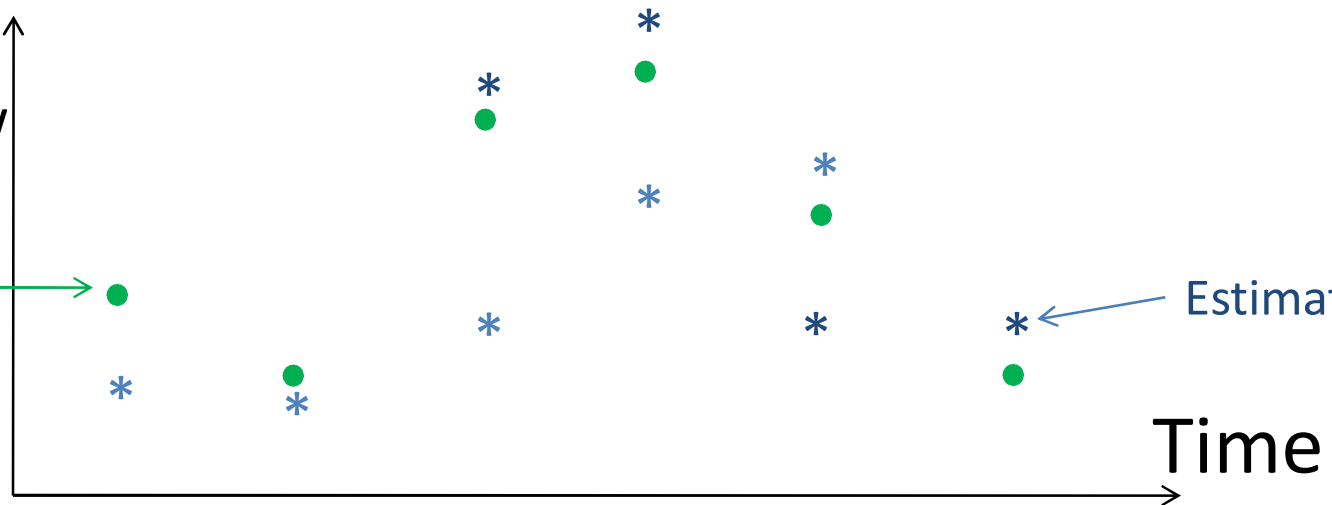
Sequential Learning (EM)



From $P(z_4 | y_1, y_2, y_3, y_4, y_5, y_6) \rightarrow$ Compute $P(z_3 | y_1, y_2, y_3, y_4, y_5, y_6)$

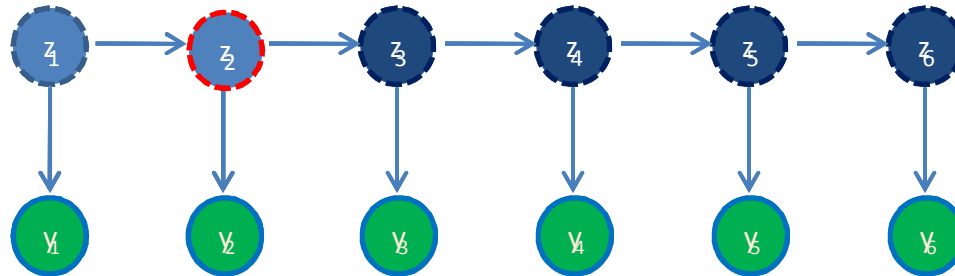
Position
of left elbow

Measured





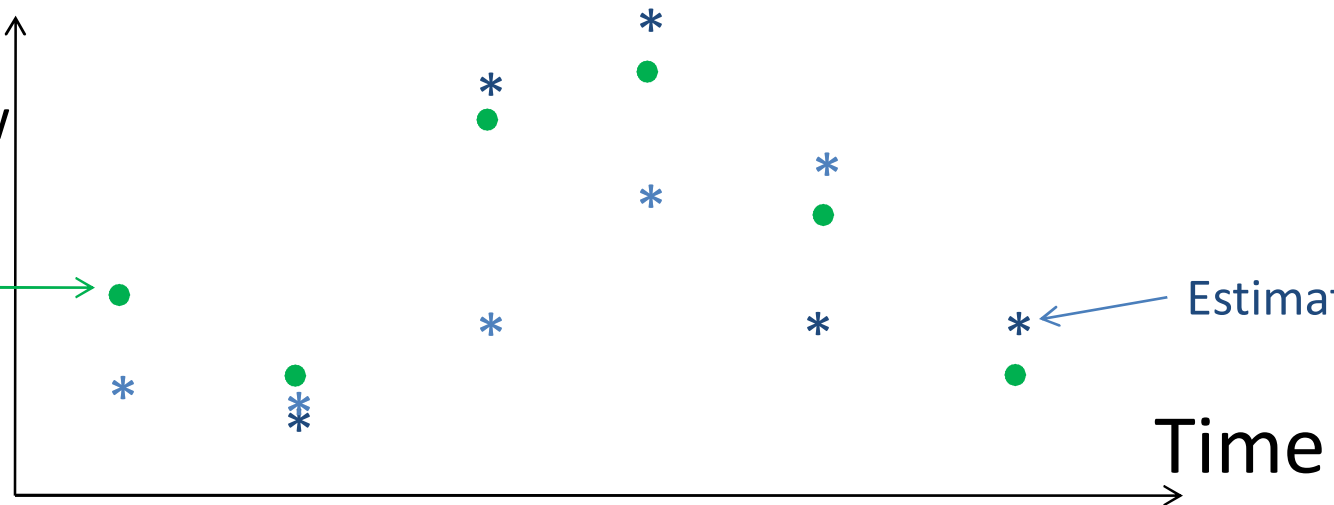
Sequential Learning (EM)



From $P(z_3 | y_1, y_2, y_3, y_4, y_5, y_6) \rightarrow$ Compute $P(z_2 | y_1, y_2, y_3, y_4, y_5, y_6)$

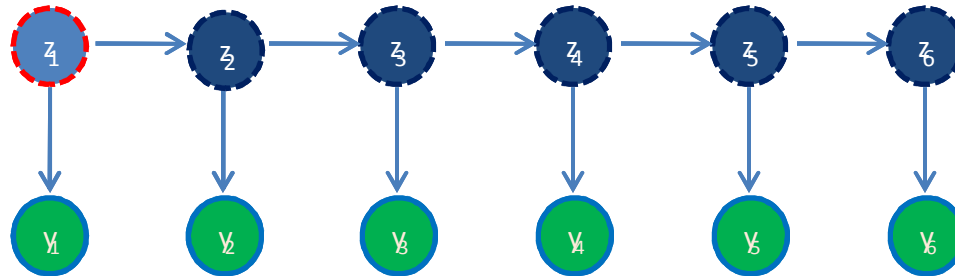
Position
of left elbow

Measured





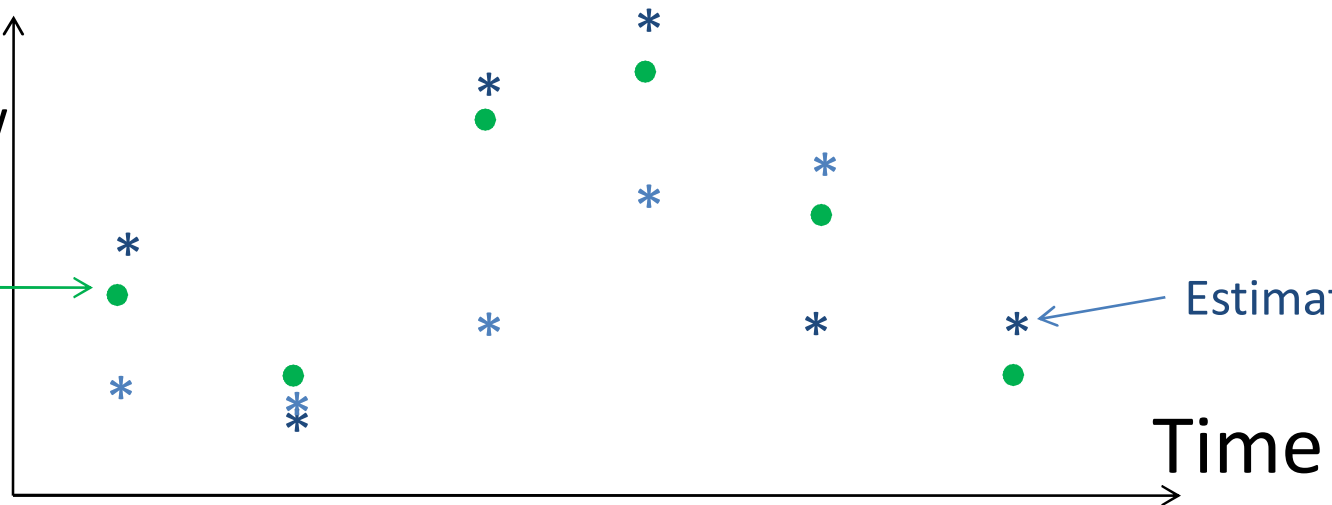
Sequential Learning (EM)



From $P(z_2 | y_1, y_2, y_3, y_4, y_5, y_6) \rightarrow$ Compute $P(z_1 | y_1, y_2, y_3, y_4, y_5, y_6)$

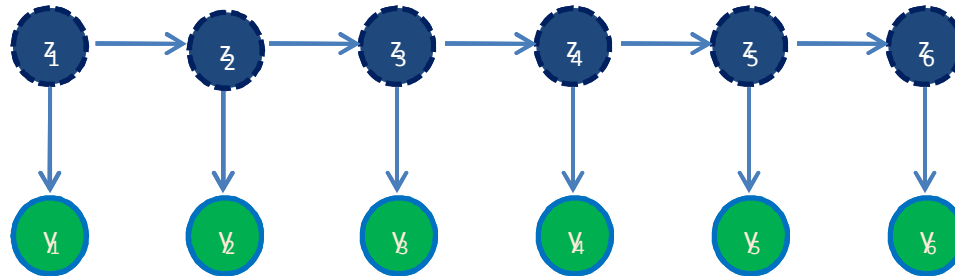
Position
of left elbow

Measured





Sequential Learning (EM)



From all posterior $z_1, z_2, z_3, z_4, z_5, z_6$

$P(z_1 | y_1, y_2, y_3, y_4, y_5, y_6), P(z_2 | y_1, y_2, y_3, y_4, y_5, y_6) \dots$

Compute sufficient statistics

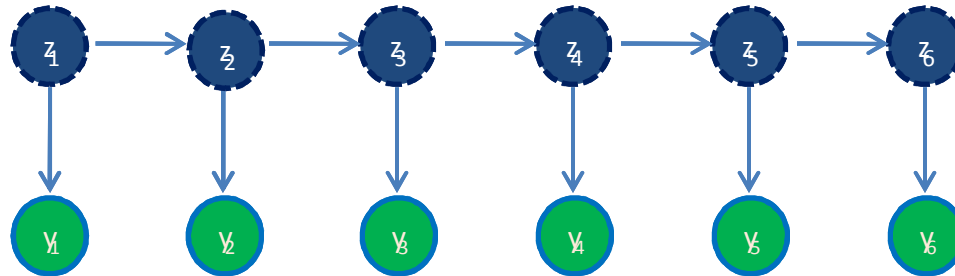
$$E[z_i]$$

$$E[z_i z_i']$$

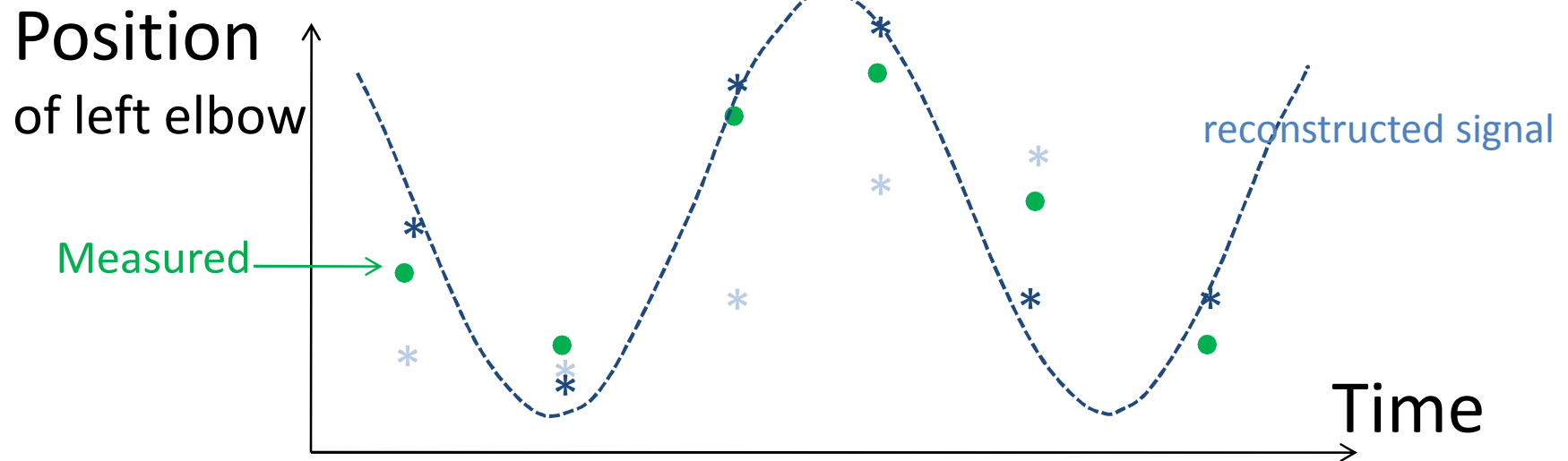
$$E[z_{i-1} z_i']$$



Sequential Learning (EM)

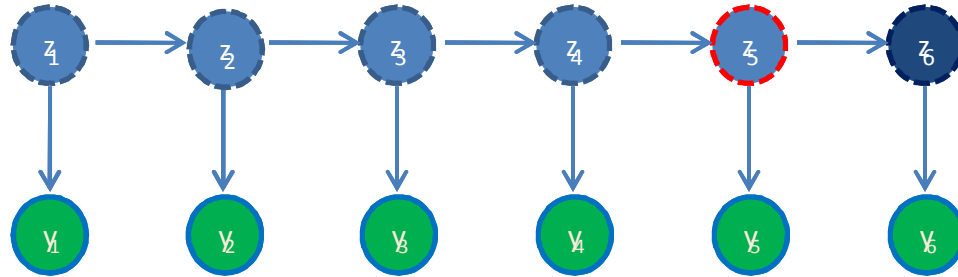


with sufficient statistics, compute $\text{argmax}_{\theta} \leftarrow \text{likelihood}(\theta)$





Sequential Learning (EM)

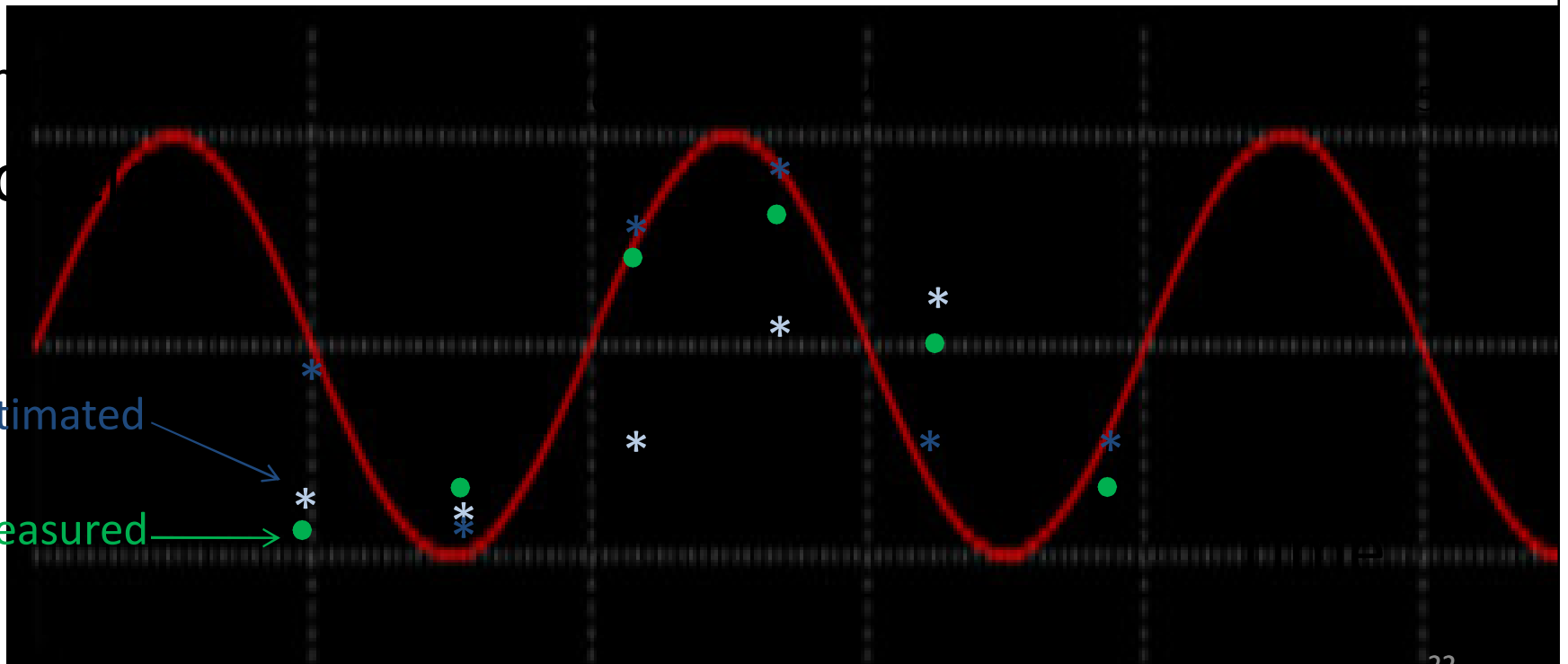


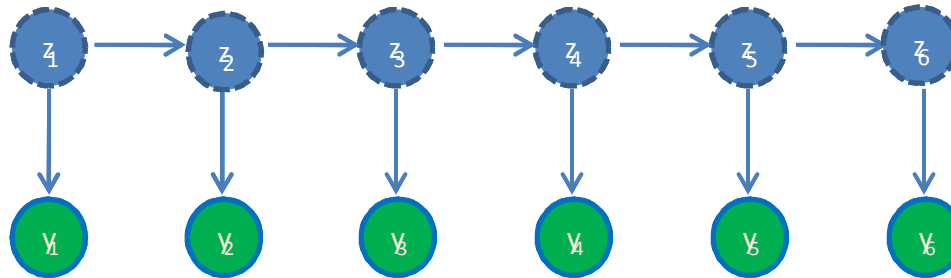
From

PO

Estimated

Measured





Speed Bottleneck:
sequential computation of posterior

How to *parallelize* it?

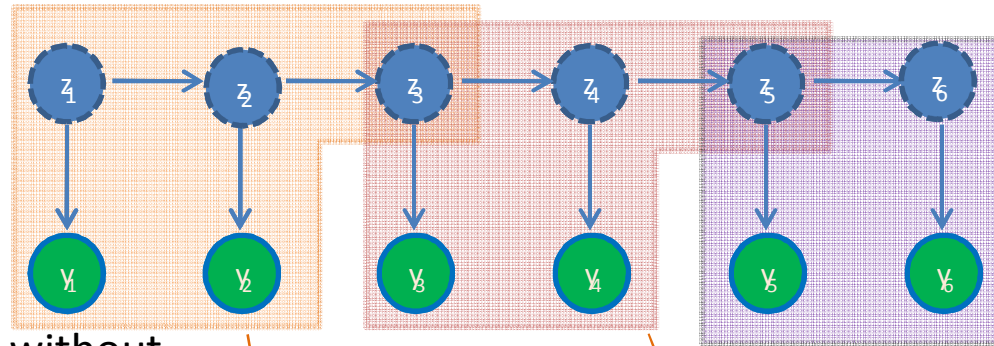


“Leap of faith”

start computation without feedback from
previous node (cut),
and reconcile later (stitch)

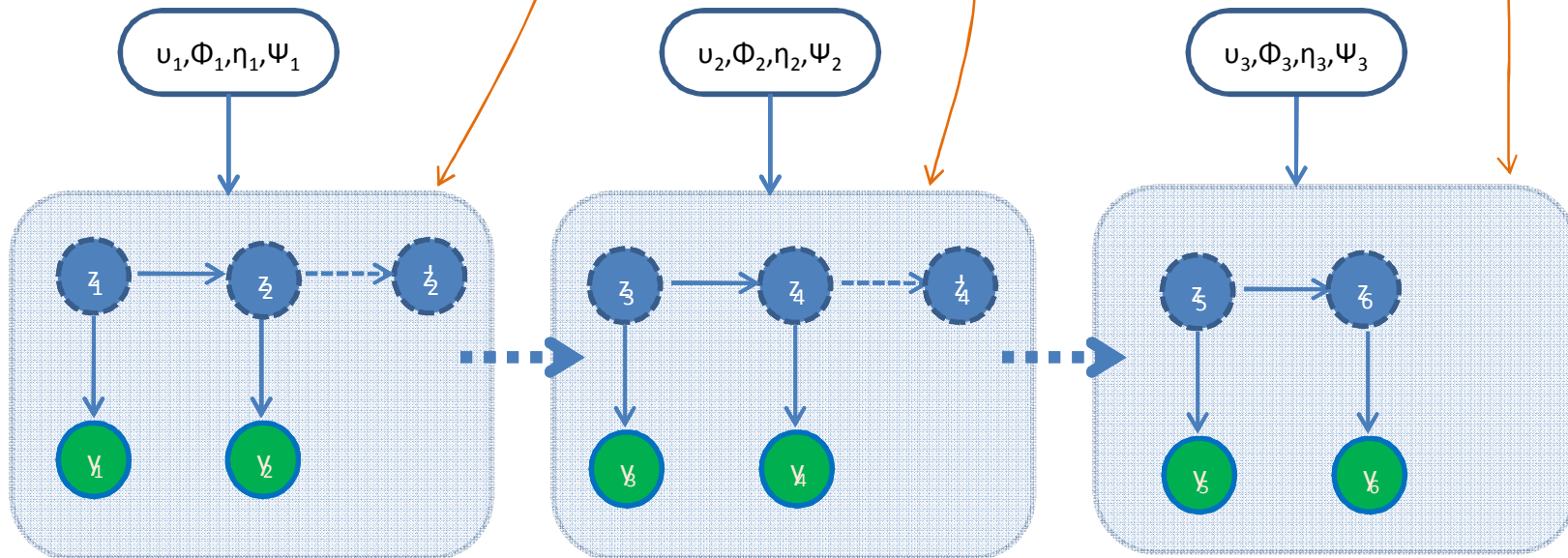


Proposed Method: Cut-And-Stitch



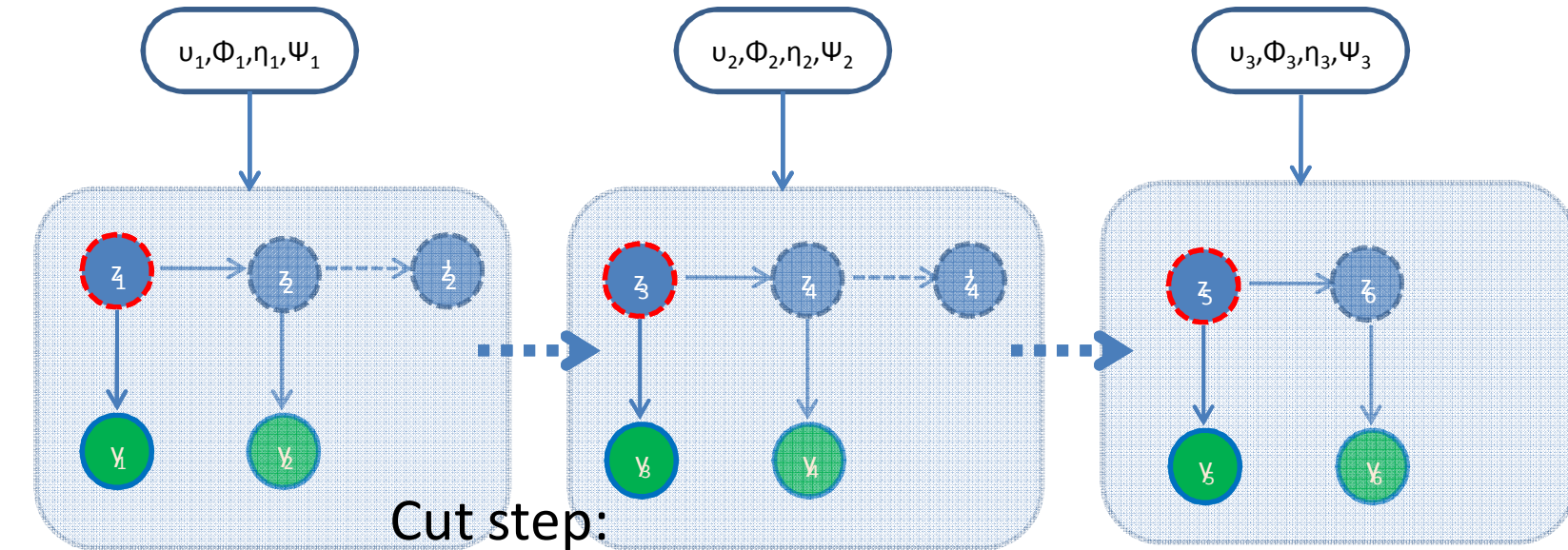
start computation without feedback from previous node (cut)

reconcile later (stitch)





Cut-And-Stitch



Cut step:

Estimate posteriors (E) $P(z_1 | y_1), P(z_3 | y_3), P(z_5 | y_5)$

Position of left elbow

Measured → ●

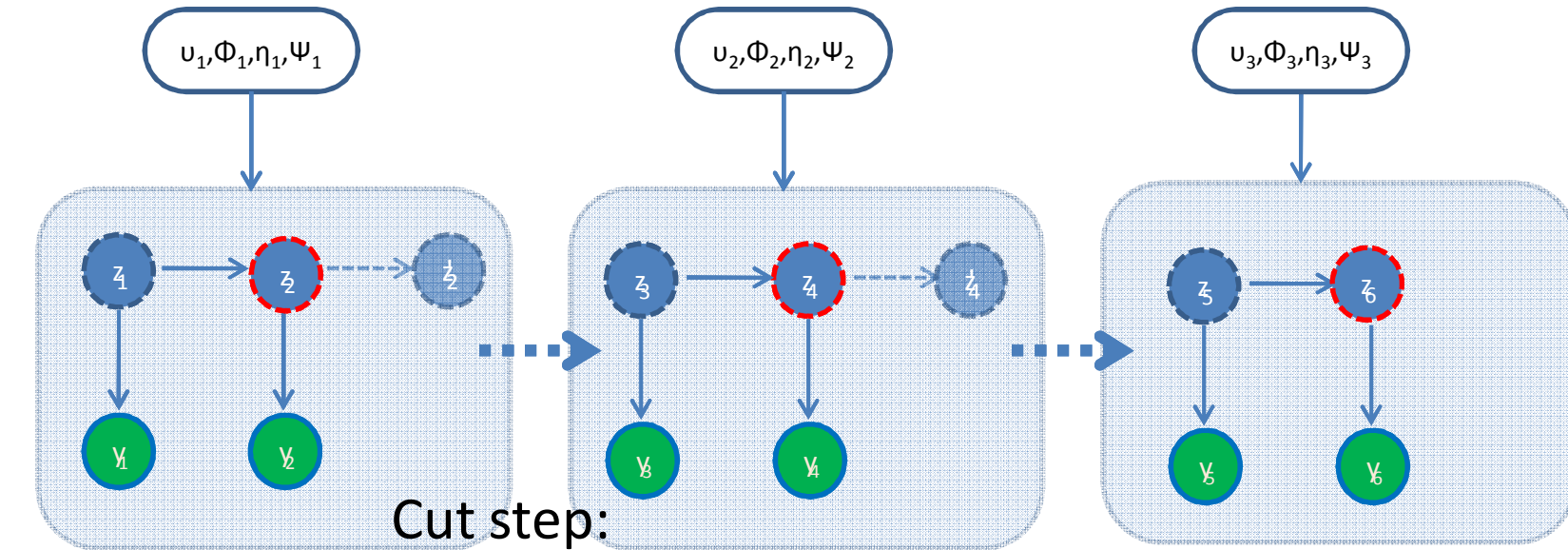
Estimated → *

Intuition: compute all three at once

Time →



Cut-And-Stitch



Cut step:

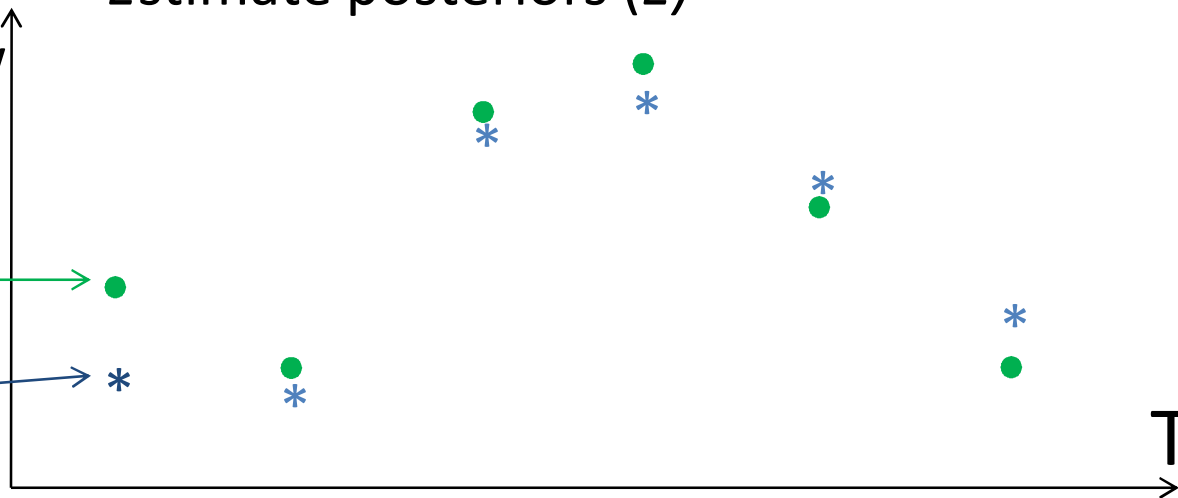
Estimate posteriors (E)

Position
of left elbow

Measured → ●

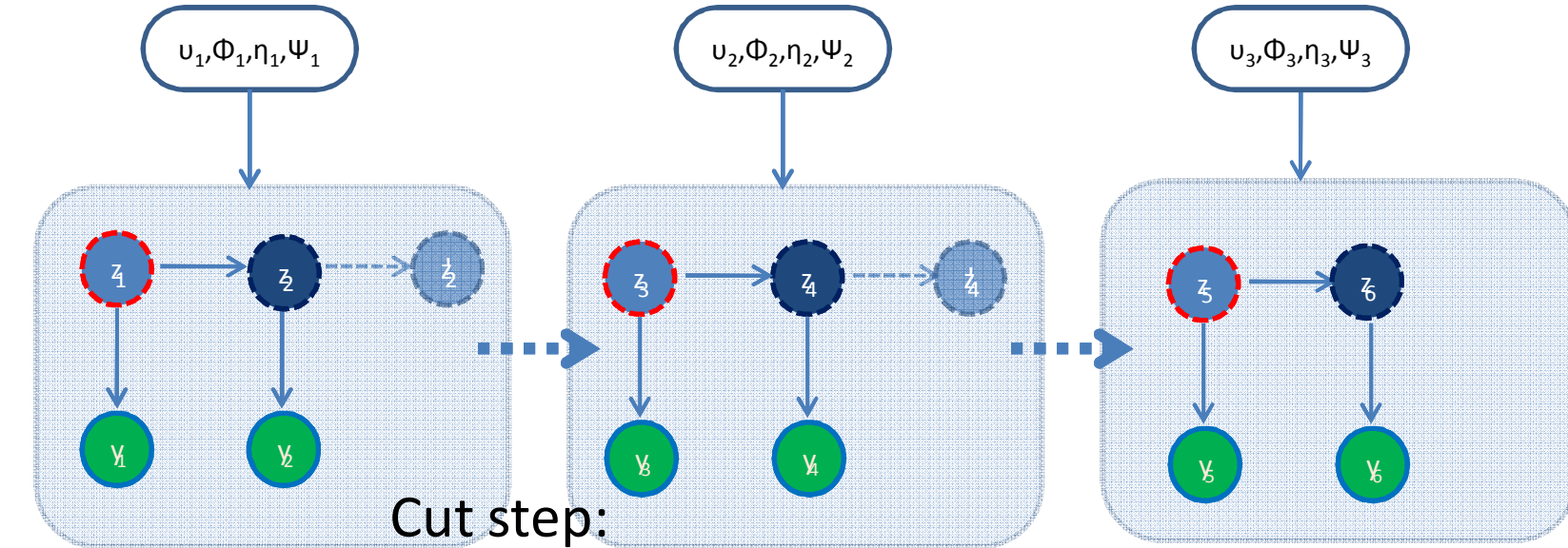
Estimated → *

Time





Cut-And-Stitch



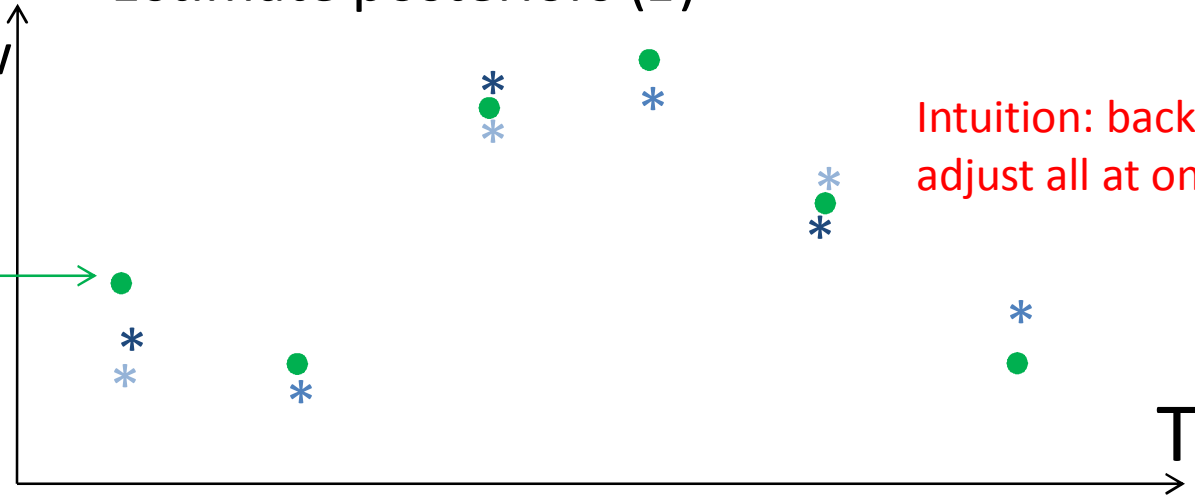
Cut step:
Estimate posteriors (E)

Position
of left elbow

Measured

Intuition: backward
adjust all at once

Time





Cut-And-Stitch

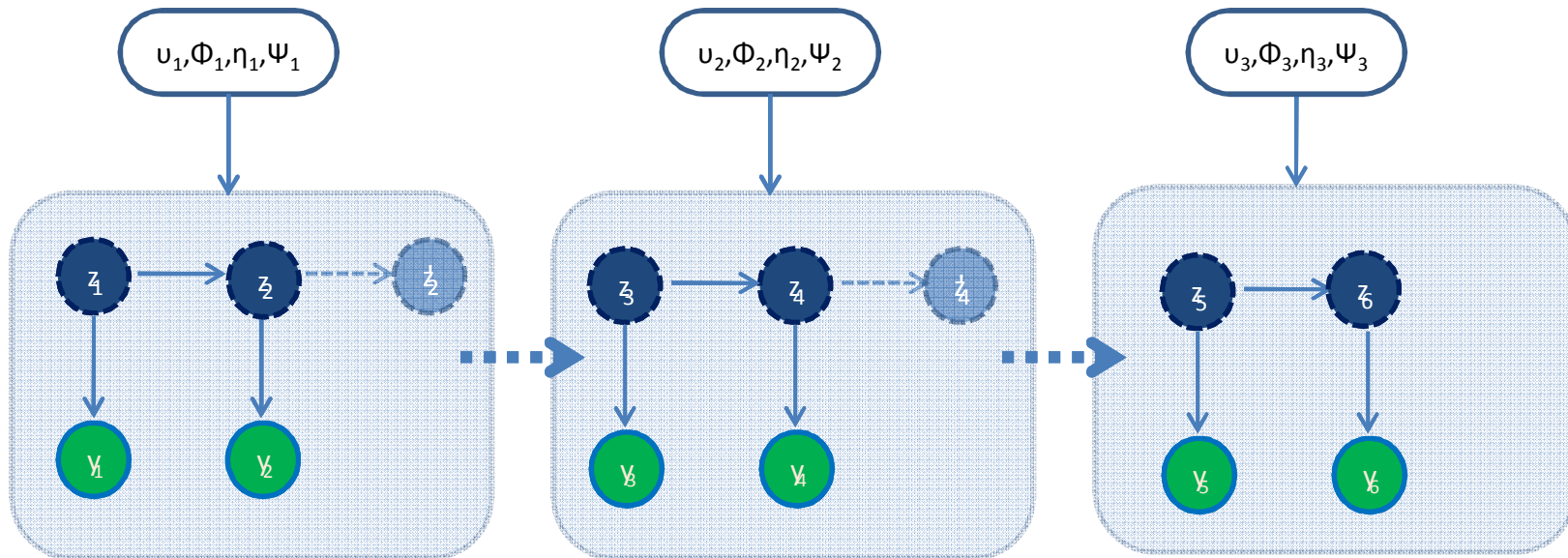
Stitch step:

Collect sufficient statistics for each block (C)

$$E[z_i]$$

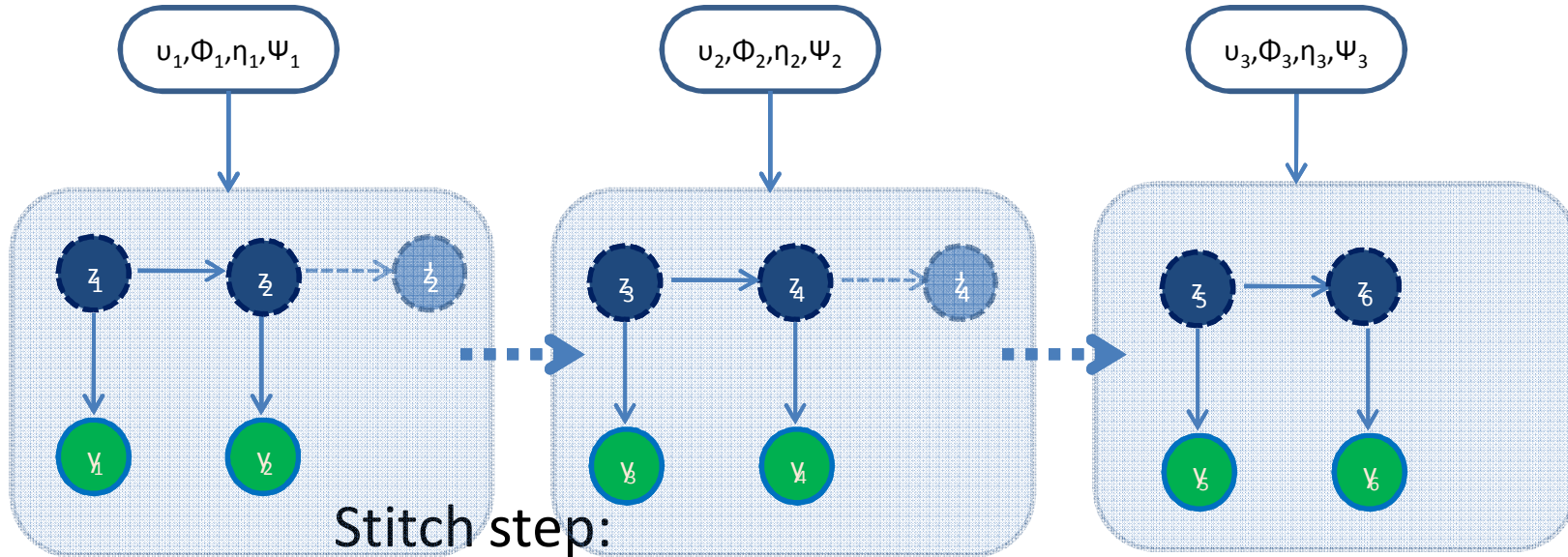
$$E[z_i z_i']$$

$$E[z_{i-1} z_i']$$





Cut-And-Stitch



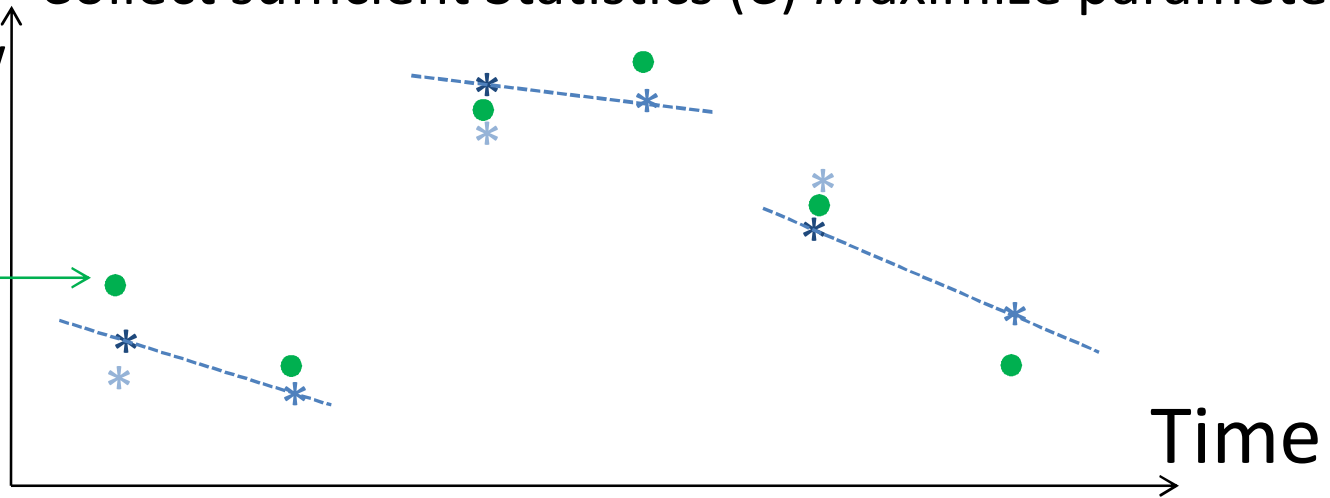
Stitch step:

Position

of left elbow

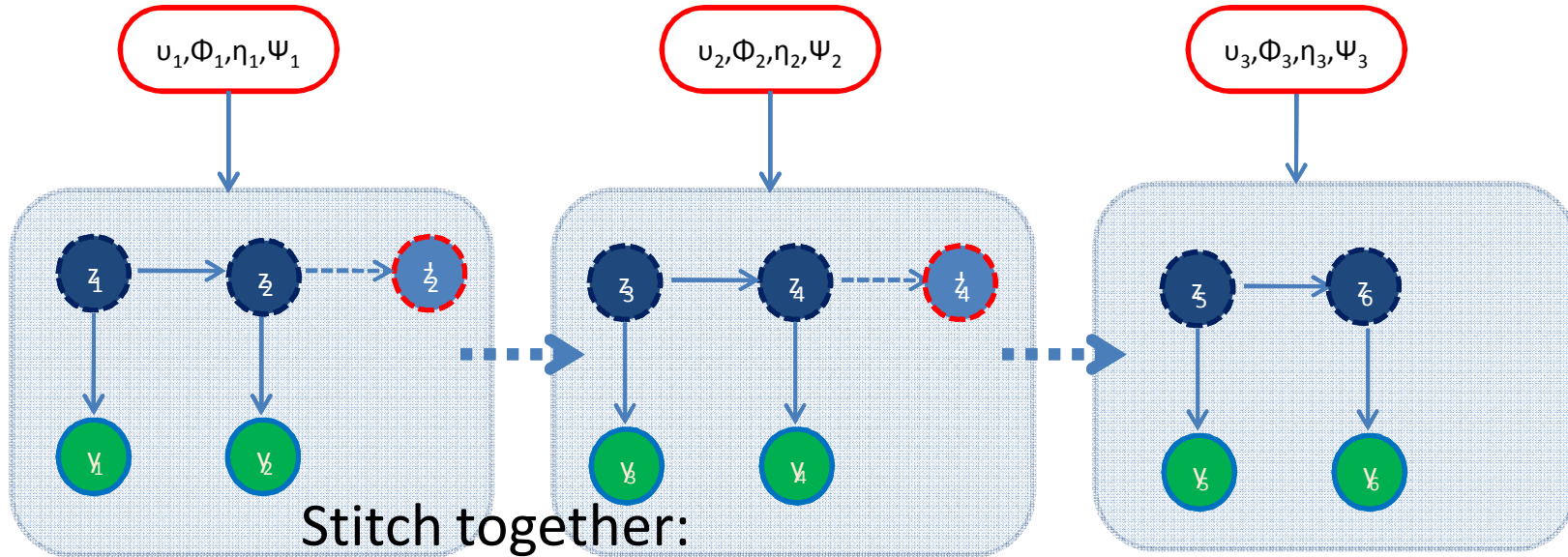
Collect sufficient Statistics (C) Maximize parameters (M)

Measured





Cut-And-Stitch



Position
of left elbow

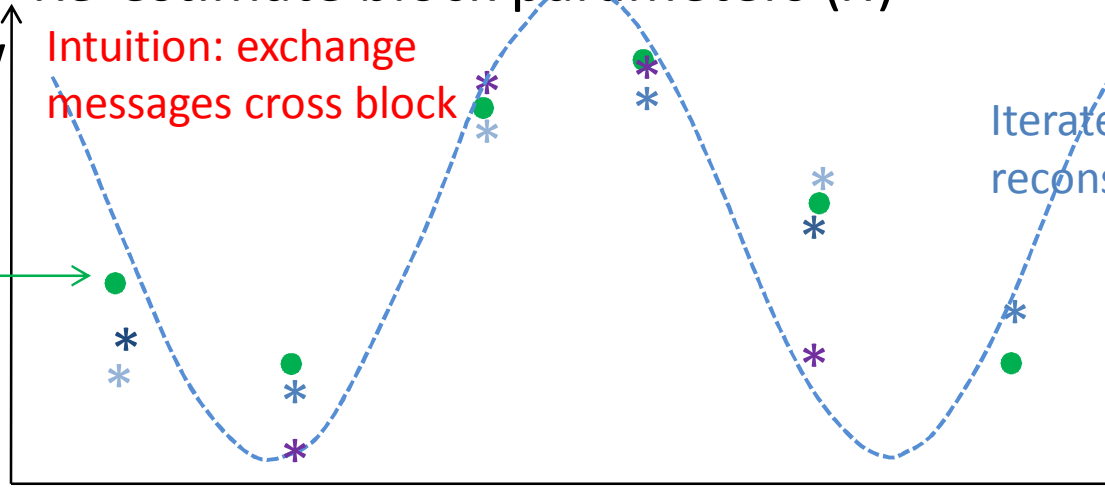
Re-estimate block parameters (R)

Intuition: exchange
messages cross block

Iterate...
reconstructed signal

Measured

Time





Experiments

Q1: How much speed up can we get?

Q2: How good is the reconstruction accuracy?

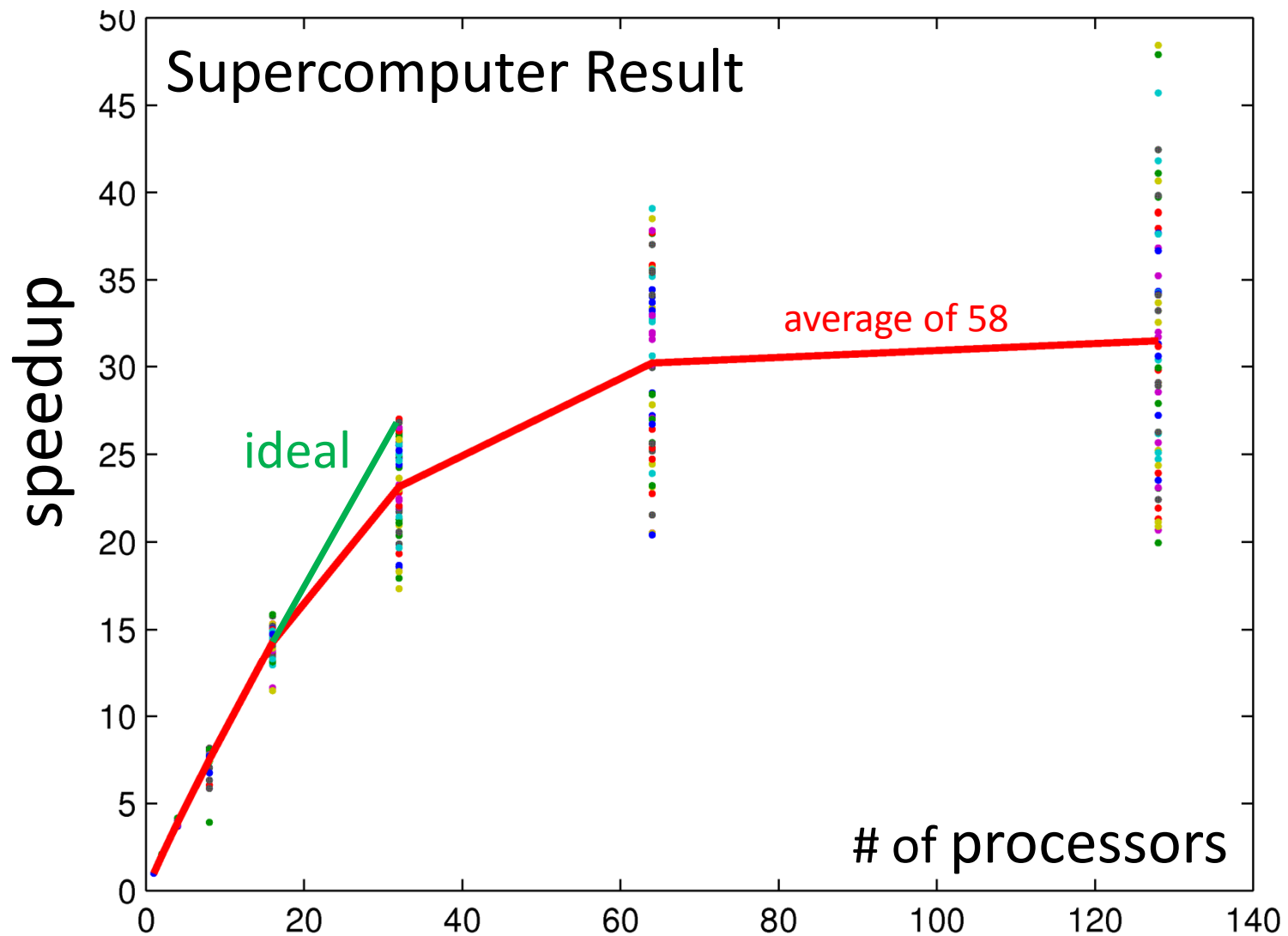


Experiments

- Dataset:
 - 58 human motion sequences, 200 – 500 frames
 - Each frame with 93 bone positions in body local coordinates
 - <http://mocap.cs.cmu.edu>
- Setup:
 - Supercomputer: SGI Altix system, distributed shared memory architecture
 - Multi-core desktop: 4 Intel Xeon cores, shared memory
- Task:
 - Learn the dynamics, hidden variables and reconstruct motion

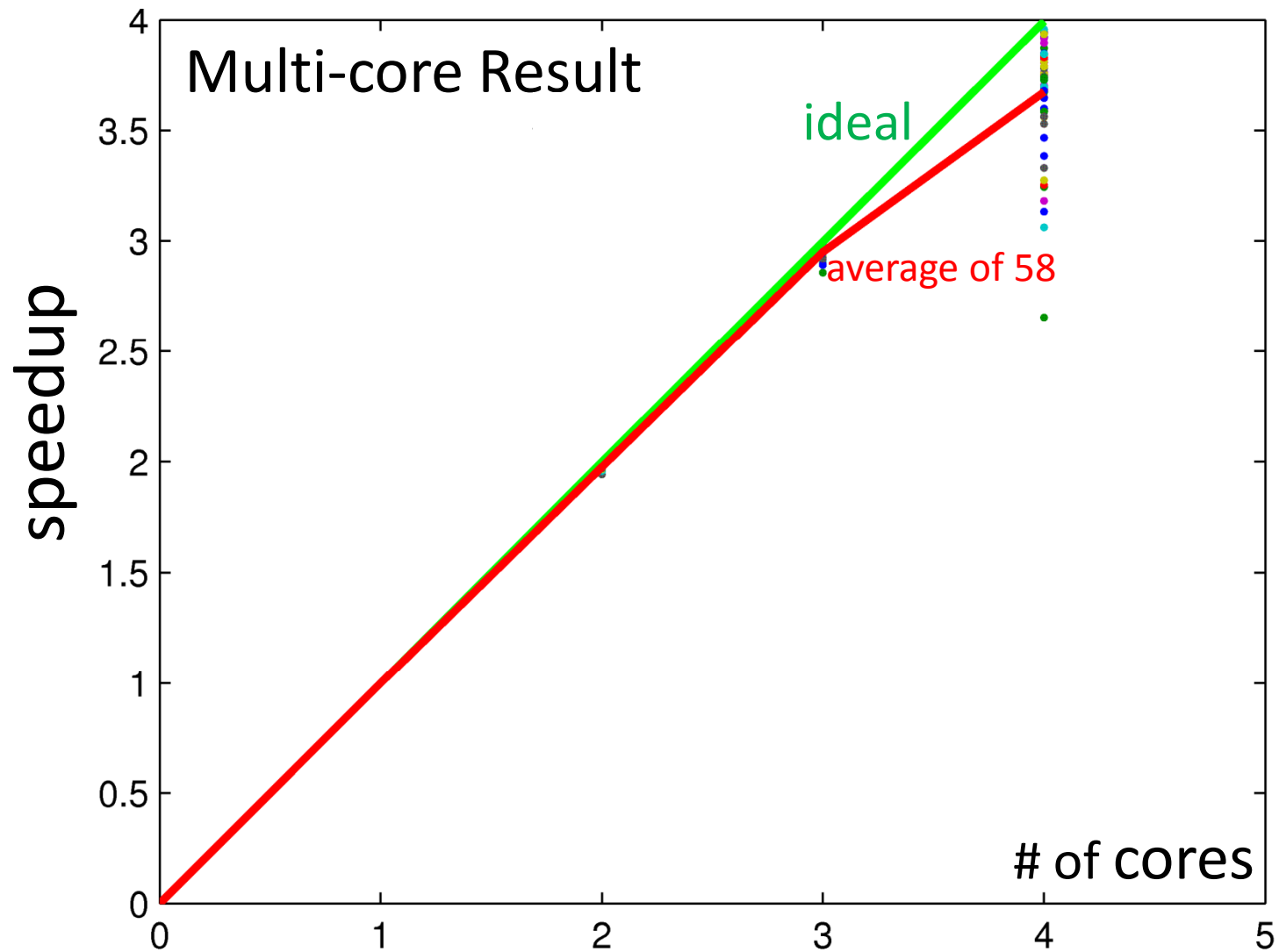


Q1: How much speed up?



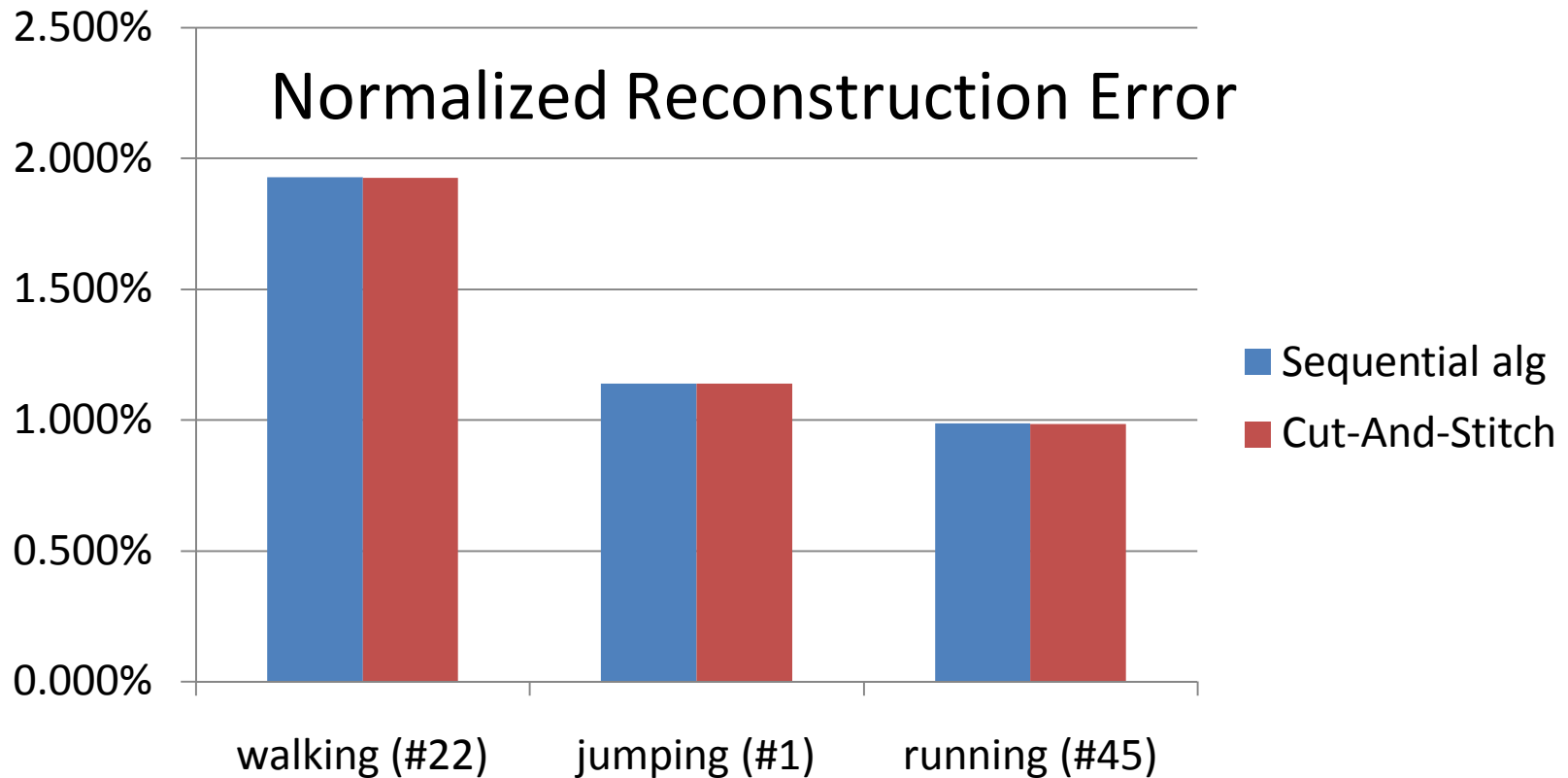


Q1: How much speed up?





Q2: How good?



Result: ~ IDENTICAL accuracy



Conclusion & Contributions

General approximate parallel learning algorithm for LDS

- Near linear speed up
- Accuracy (NRE): \sim identical to sequential learning
- Easily extended to HMM and other chain Markovian models

Software (C++ w. openMP) and datasets:

www.cs.cmu.edu/~leili/paralearn

Thank you

- Questions?

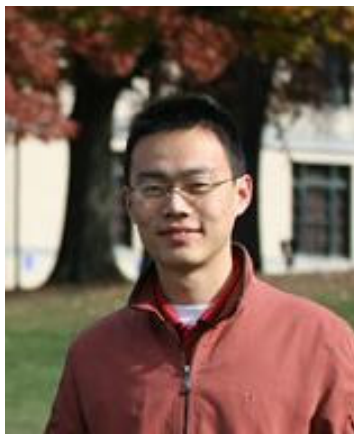
WELCOME TO POSTER TONIGHT

- Contact info: leili@cs.cmu.edu



Christos Faloutsos

Wenjie Fu



Fan Guo

Lei Li

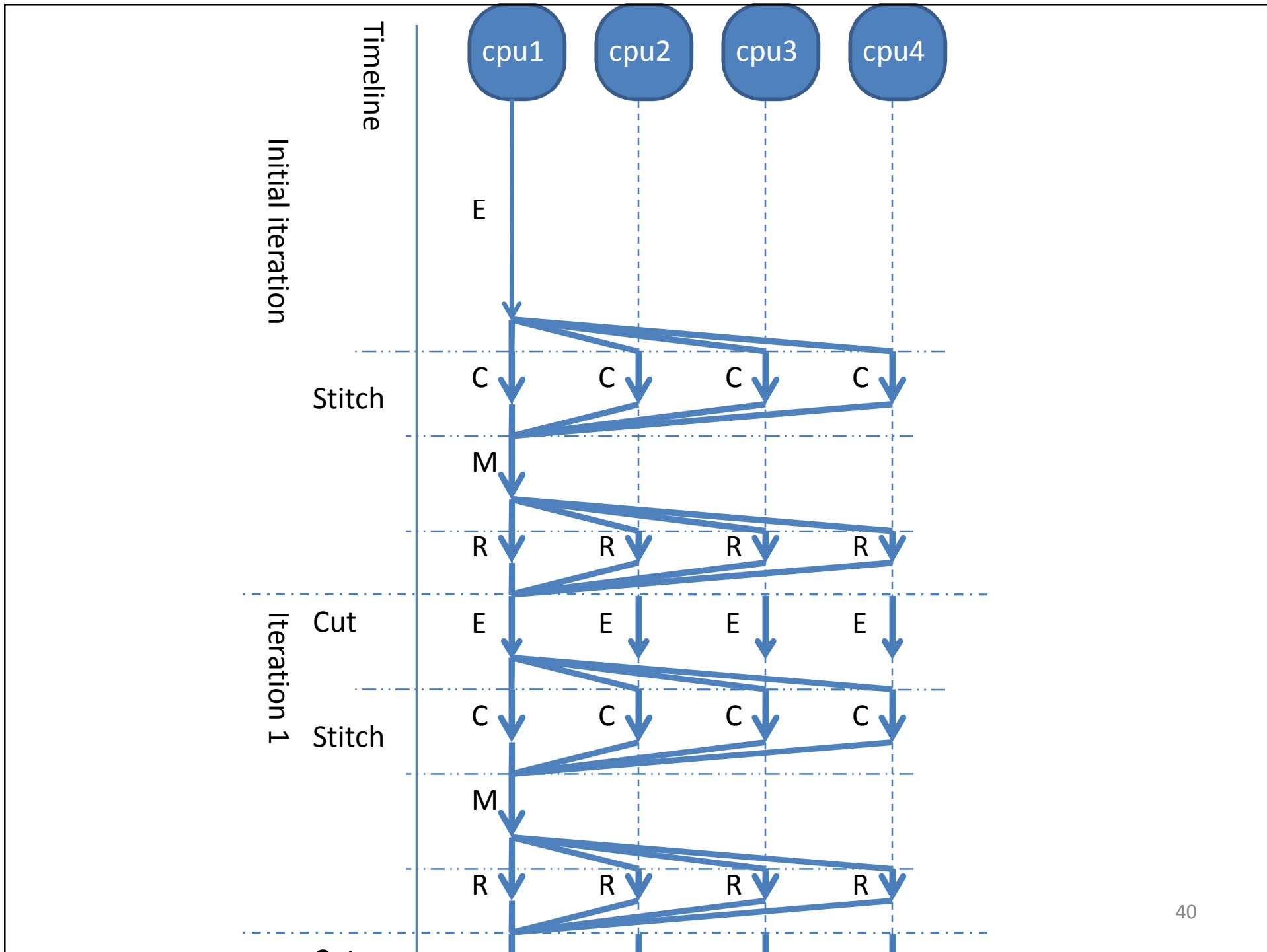


Todd Mowry



Promising Extensions

- Extension
 - HMM
 - other Markov models (*similar graphical model*)
- Open Problem:
 - Can prove the error bound?





Approximate Parallel Learning

