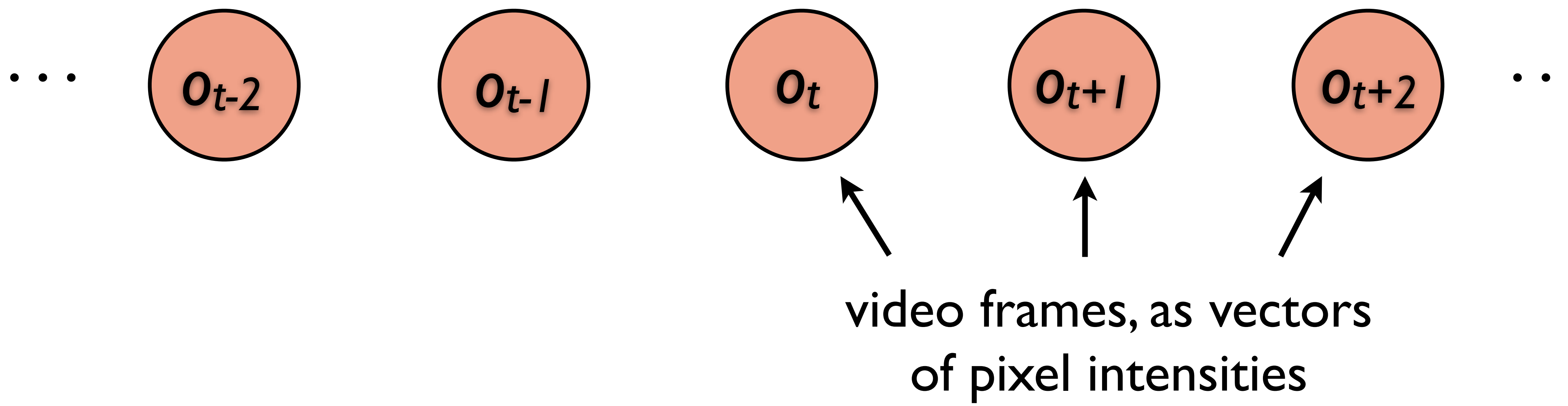
Spectral learning algorithms for dynamical systems

Geoff Gordon
http://www.cs.cmu.edu/~ggordon/
Machine Learning Department
Carnegie Mellon University

joint work with Byron Boots, Sajid Siddiqi, Le Song, Alex Smola

What's out there?

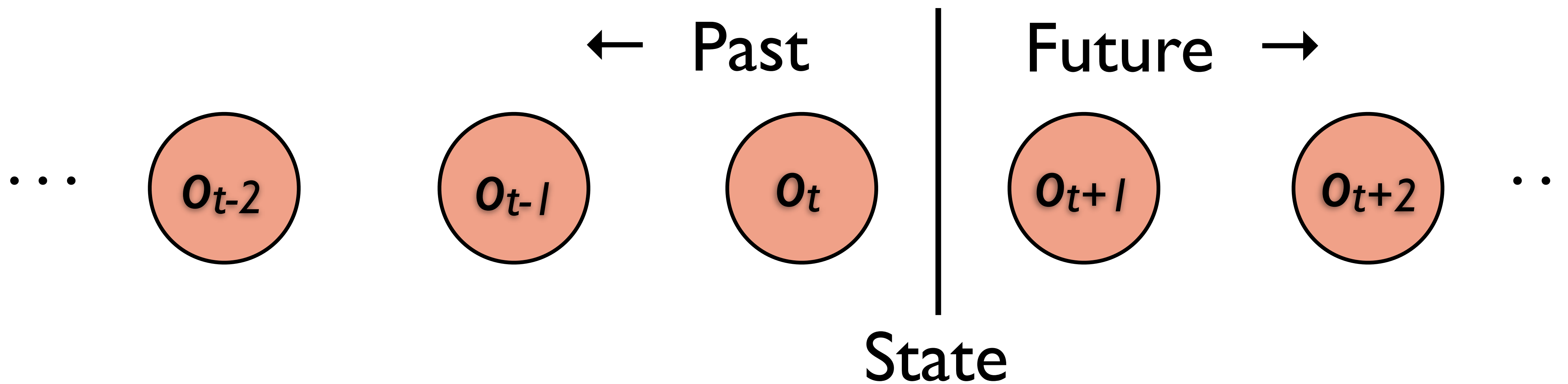


What's out there?

… o_{t-2} o_{t-1} o_t o_{t+1} o_{t+2} …

video frames, as vectors
of pixel intensities

A dynamical system



Given past observations from a partially observable system

Predict future observations

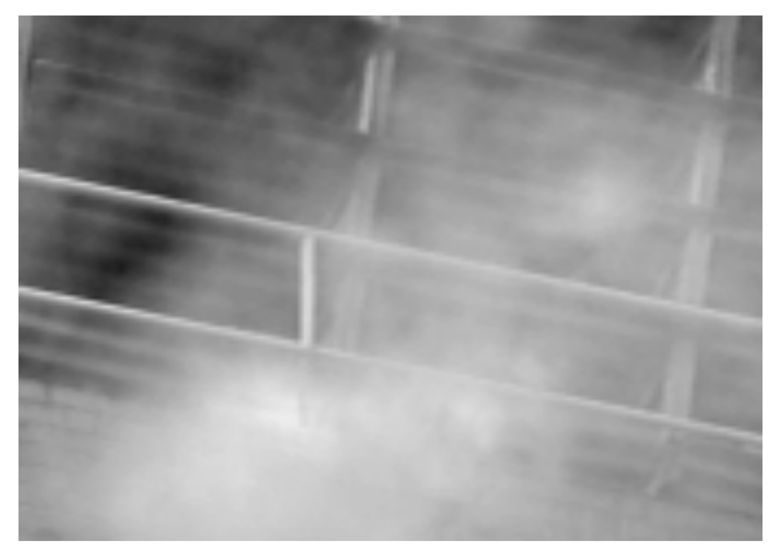
A dynamical system

← Past | Future →

… o_{t-2} o_{t-1} o_t | o_{t+1} o_{t+2} …

State

Given past observations from a partially observable system



Predict future observations

This talk

- General class of models for dynamical systems
- Fast, statistically consistent learning algorithm
  - no local optima
- Includes many well-known models & algorithms as special cases
- Also includes new models & algorithms that give state-of-the-art performance on interesting tasks

Includes models

- hidden Markov model
  - n-grams, regexes, k-order HMMs
- PSR, OOM, multiplicity automaton, RR-HMM
- LTI system
  - Kalman filter, AR, ARMA
- Kernel versions and manifold versions of above

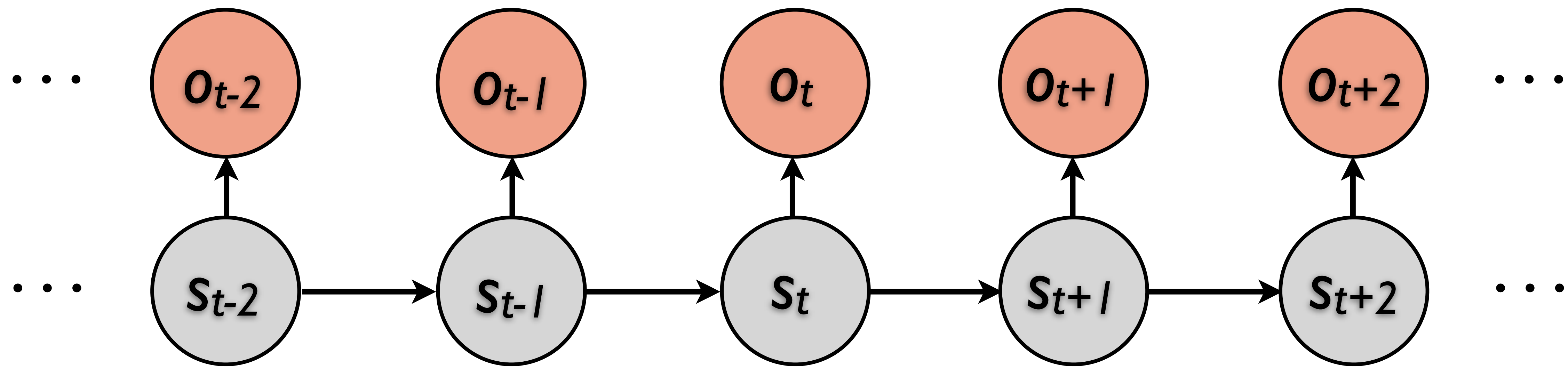
Includes algorithms

- Subspace identification for LTI systems
  - recent extensions to HMMs, RR-HMMs
- Tomasi-Kanade structure from motion
- Principal components analysis (e.g., eigenfaces); Laplacian eigenmaps
- New algorithms for learning PSRs, OOMs, etc.
- A new way to reduce noise in manifold learning
- A new range-only SLAM algorithm

Interesting applications

- Structure from motion
- Simultaneous localization and mapping
  - Range-only SLAM
  - “SLAM” from inertial sensors
  - very simple vision-based SLAM (so far)
- Video textures
- Opponent modeling, option pricing, audio event classification, …

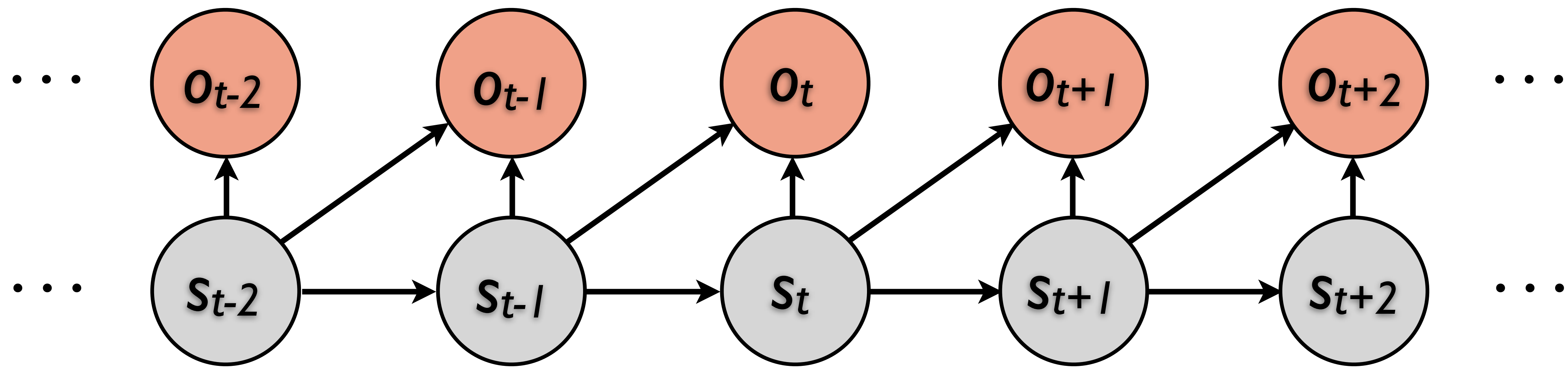
Bayes filter (HMM, Kalman, etc.)



Belief $P_t(s_t) = P(s_t \mid o_{1:t})$

- Goal: given o_t, update $P_{t-1}(s_{t-1})$ to $P_t(s_t)$
- ***Extend***: $P_{t-1}(s_{t-1}, o_t, s_t) = P_{t-1}(s_{t-1})\ P(s_t \mid s_{t-1})\ P(o_t \mid s_t)$
- ***Marginalize***: $P_{t-1}(o_t, s_t) = \int P_{t-1}(s_{t-1}, o_t, s_t)\, ds_{t-1}$
- ***Condition***: $P_t(s_t) = P_{t-1}(o_t, s_t) \,/\, P_{t-1}(o_t)$

Bayes filter (HMM, Kalman, etc.)



Belief $P_t(s_t) = P(s_t \mid o_{1:t})$

- Goal: given o_t, update $P_{t-1}(s_{t-1})$ to $P_t(s_t)$
- ***Extend***: $P_{t-1}(s_{t-1}, o_t, s_t) = P_{t-1}(s_{t-1})\, P(s_t \mid s_{t-1})\, P(o_t \mid s_t)$
- ***Marginalize***: $P_{t-1}(o_t, s_t) = \int P_{t-1}(s_{t-1}, o_t, s_t)\, ds_{t-1}$
- ***Condition***: $P_t(s_t) = P_{t-1}(o_t, s_t) \,/\, P_{t-1}(o_t)$

A common form for Bayes filters

- Find covariances as (linear) fns of previous state
  - $\Sigma_o(s_{t-1}) = E(\varphi(o_t)\,\varphi(o_t)^T \mid s_{t-1})$
  - $\Sigma_{so}(s_{t-1}) = E(s_t\,\varphi(o_t) \mid s_{t-1})$

    ***nb**: uncentered covars;* s_t & $\varphi(o_t)$ *include constant*
- Linear regression to get current state
  - $s_t = \Sigma_{so}\,\Sigma_o^{-1}\,\varphi(o_t)$
- Exact if discrete (HMM, PSR), Gaussian (Kalman, AR), RKHS w/ characteristic kernel [Fukumizu et al.]
- Approximates many more

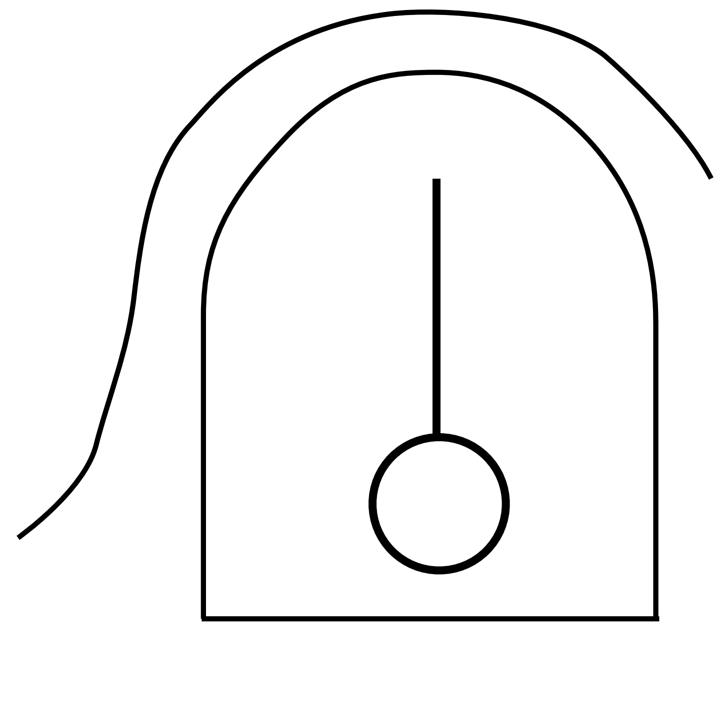
Why this form?

- If Bayes filter takes (approximately) the above form, can design a simple ***spectral*** algorithm to identify the system
- Intuitions:
  - predictive state
  - rank bottleneck
  - observable representation

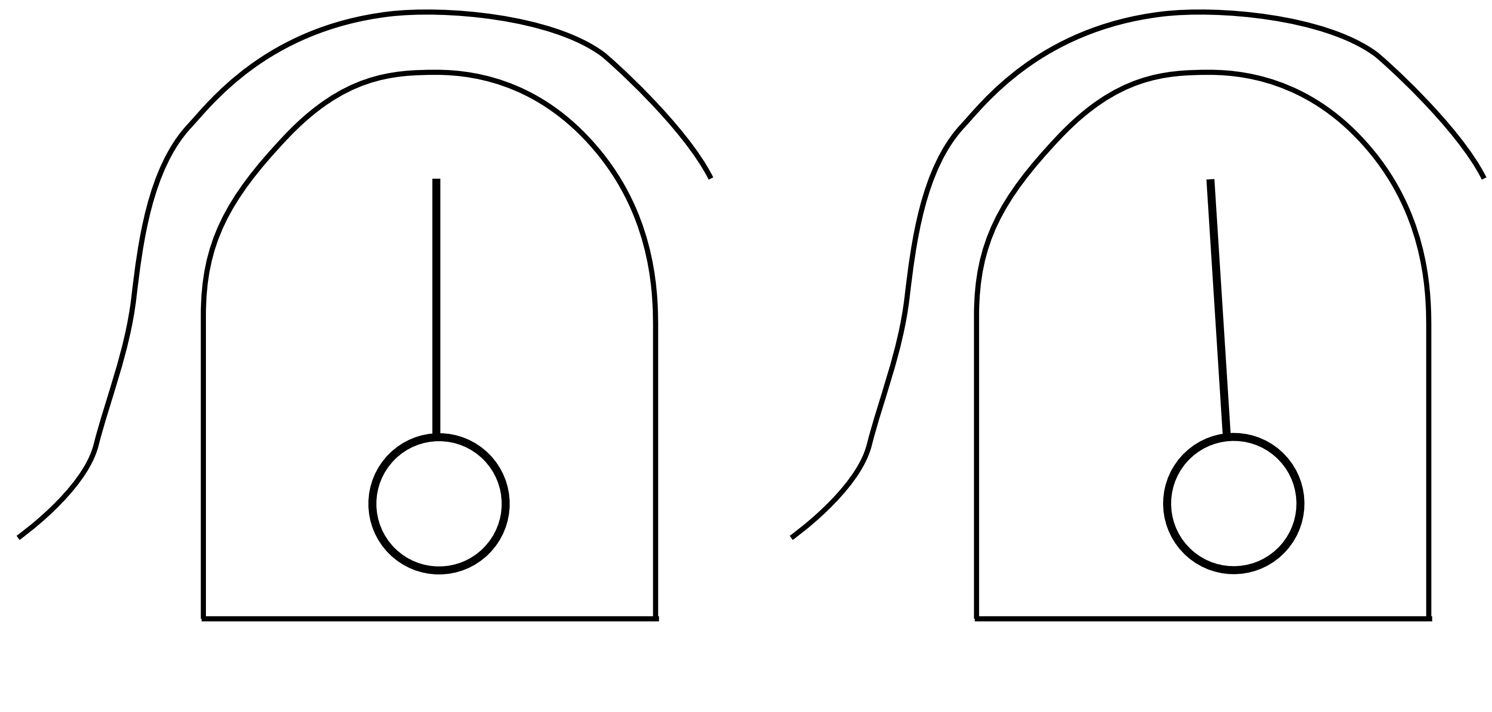
Predictive state

- If Bayes filter takes above form, we can use a vector of predictions of observables as our state
  - $E(\varphi(o_{t+k}) \mid s_t)$ = linear fn of s_t
  - for big enough k, $E([\varphi(o_{t+1}) \ \ldots \ \varphi(o_{t+k})] \mid s_t)$ = ***invertible*** linear fn of s_t
  - so, take $E([\varphi(o_{t+1}) \ \ldots \ \varphi(o_{t+k})] \mid s_t)$ as our state

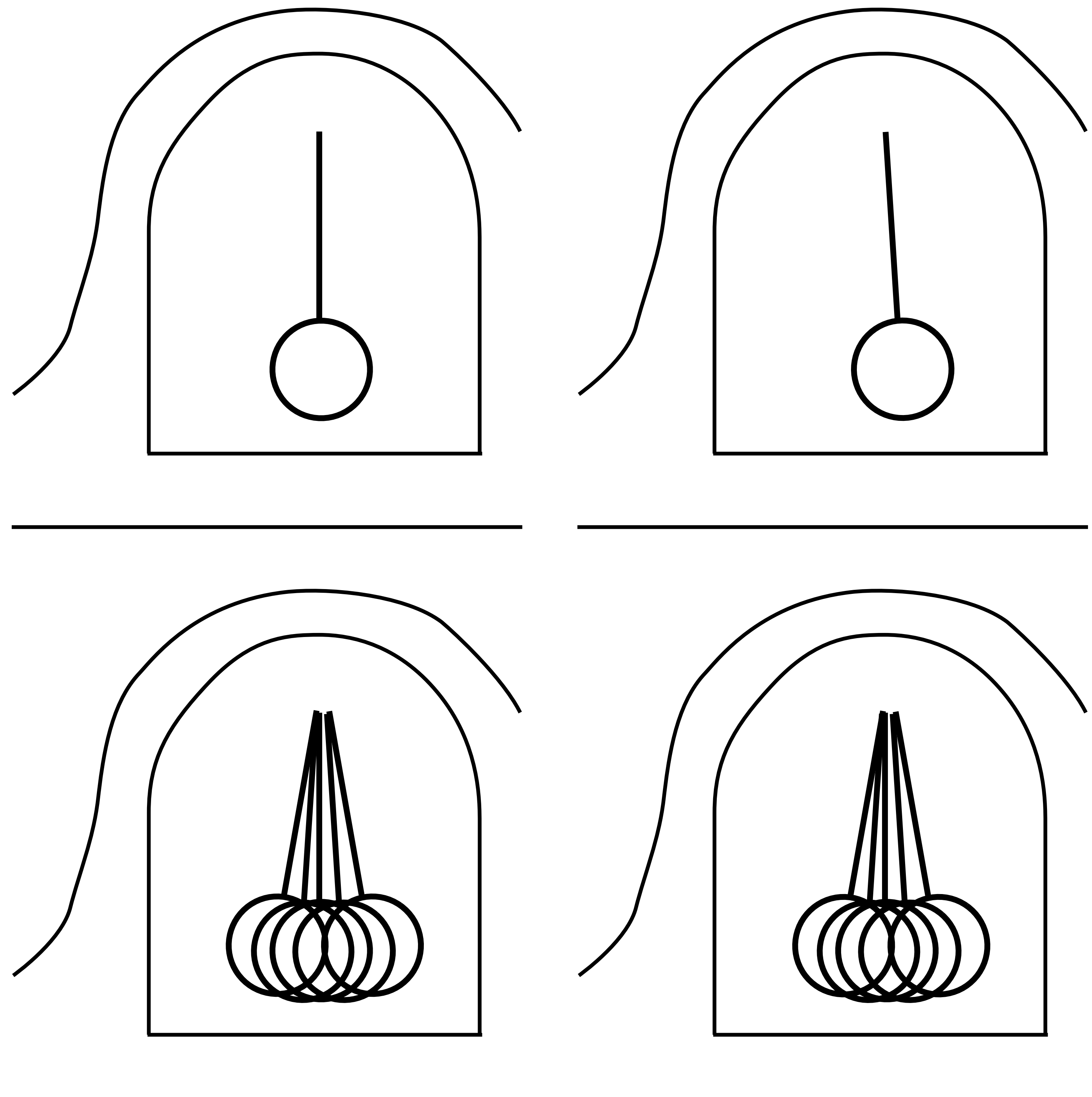
Predictive state: example



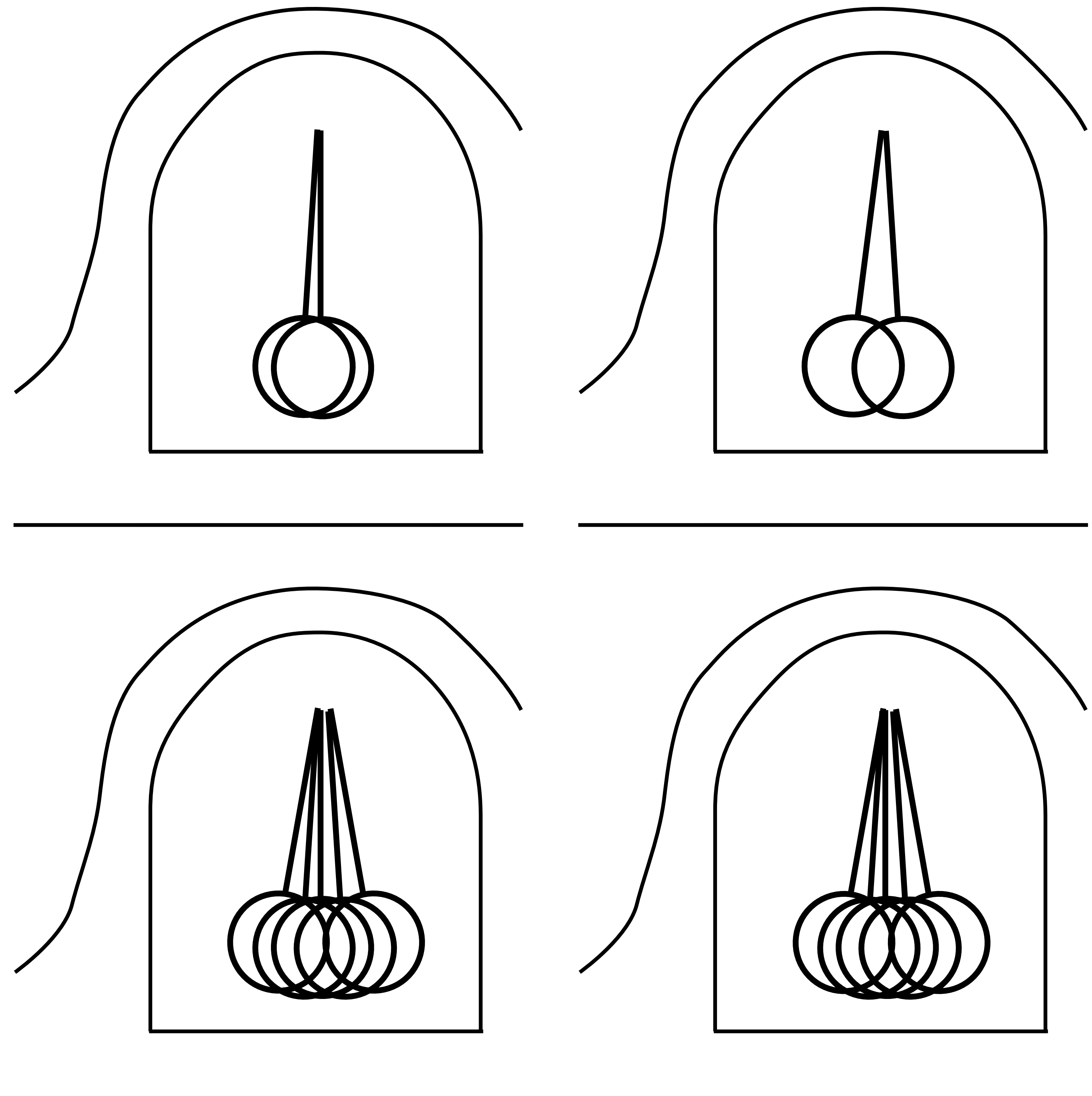
Predictive state: example



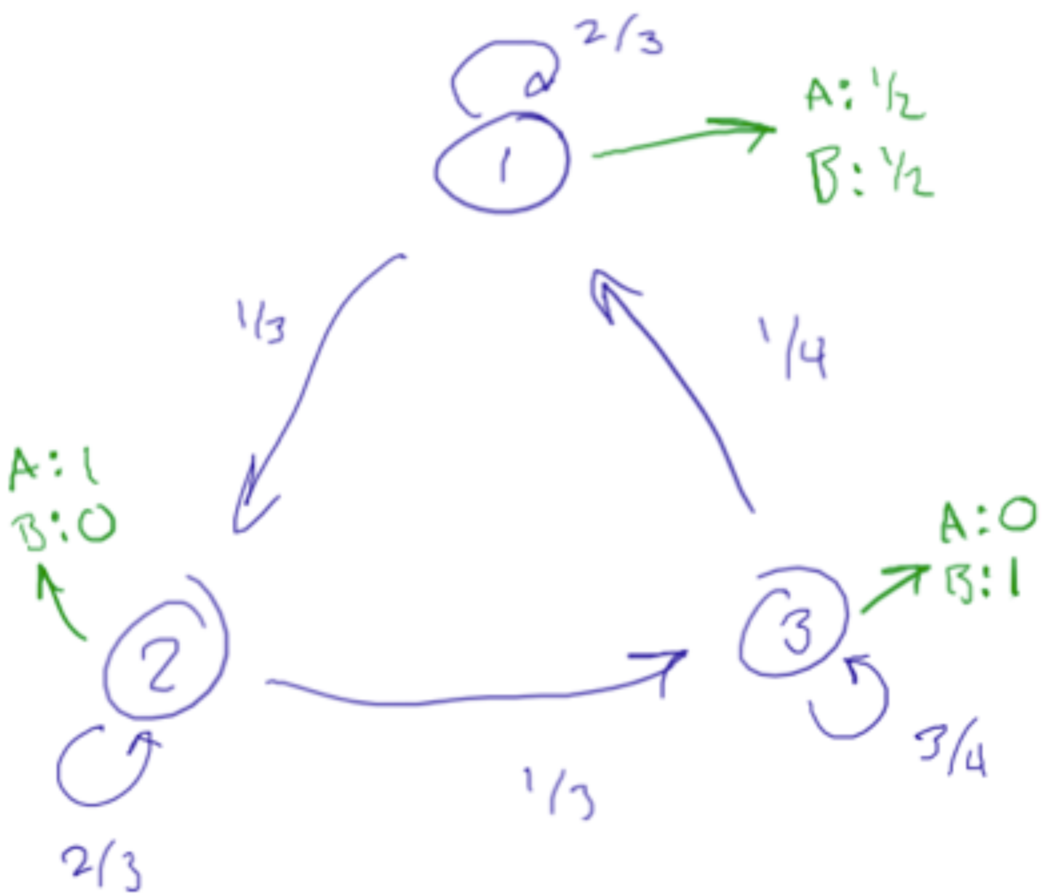
Predictive state: example



Predictive state: example

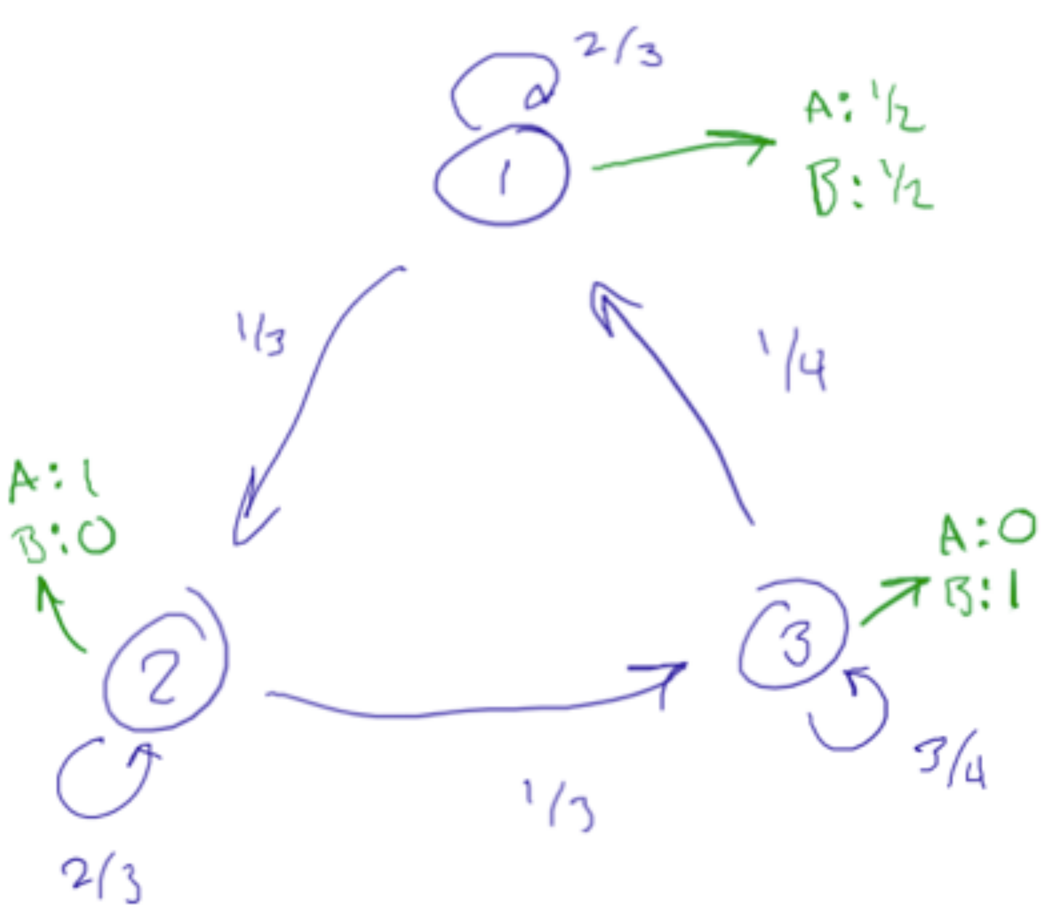


Predictive state: minimal example



- For big enough k, $E([\varphi(o_{t+1}) \ \ldots \ \varphi(o_{t+k})] \mid s_t) = Ws_t$ ***invertible*** linear fn (if system ***observable***)

Predictive state: minimal example



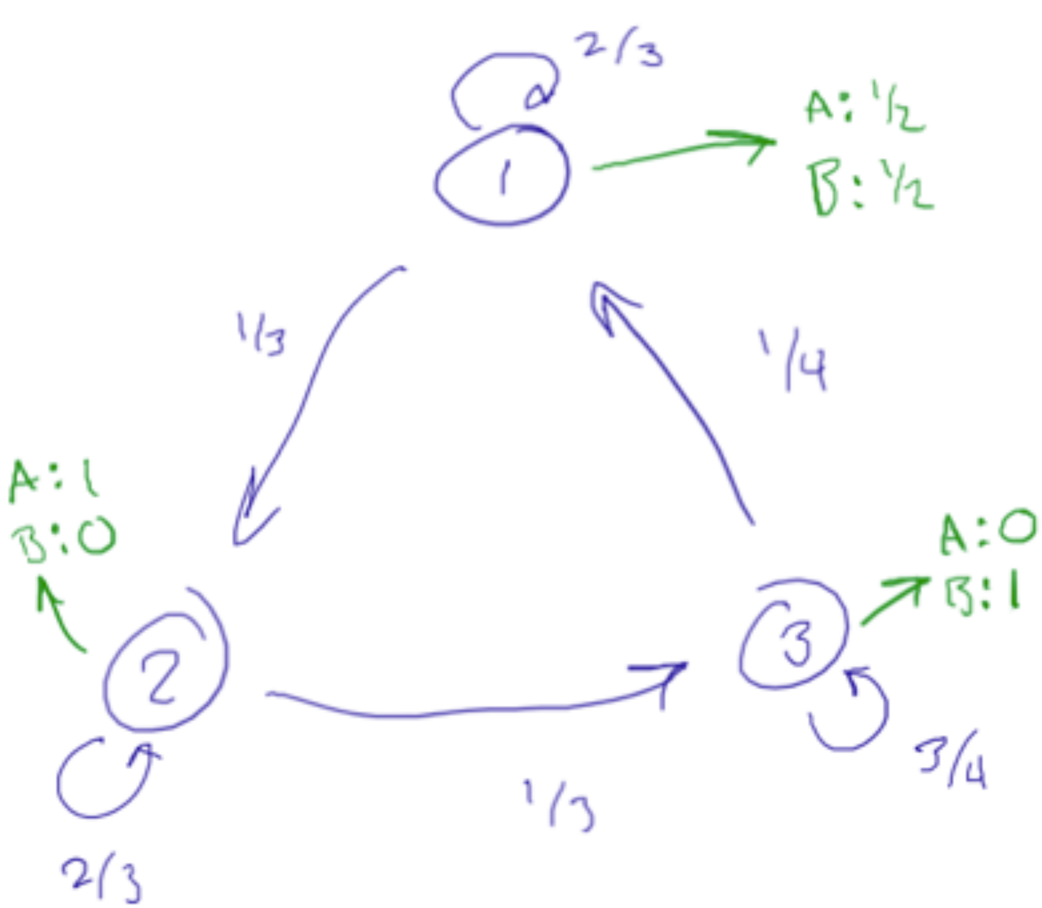
$$\mathbb{P}(s_t \mid s_{t-1}) = \begin{bmatrix} \frac{2}{3} & 0 & \frac{1}{4} \\ \frac{1}{3} & \frac{2}{3} & 0 \\ 0 & \frac{1}{3} & \frac{3}{4} \end{bmatrix} = T$$

$$\mathbb{P}(o_t \mid s_{t-1}) = \begin{bmatrix} \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{bmatrix} = O$$

$$OT = \begin{bmatrix} \frac{2}{3} & \frac{2}{3} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{7}{8} \end{bmatrix}$$

- For big enough k, E([φ(o$_{t+1}$) … φ(o$_{t+k}$)] | s$_t$) = Ws$_t$ ***invertible*** linear fn (if system ***observable***)

Predictive state: minimal example



$$\mathbb{P}(s_t \mid s_{t-1}) = \begin{bmatrix} \frac{2}{3} & 0 & \frac{1}{4} \\ \frac{1}{3} & \frac{2}{3} & 0 \\ 0 & \frac{1}{3} & \frac{3}{4} \end{bmatrix} = T$$

$$\mathbb{P}(o_t \mid s_{t-1}) = \begin{bmatrix} \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{bmatrix} = O$$

$$OT = \begin{bmatrix} \frac{2}{3} & \frac{2}{3} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{7}{8} \end{bmatrix}$$

W

- For big enough k, $E([\varphi(o_{t+1}) \ \ldots \ \varphi(o_{t+k})] \mid s_t) = W s_t$ ***invertible*** linear fn (if system ***observable***)

Predictive state: summary

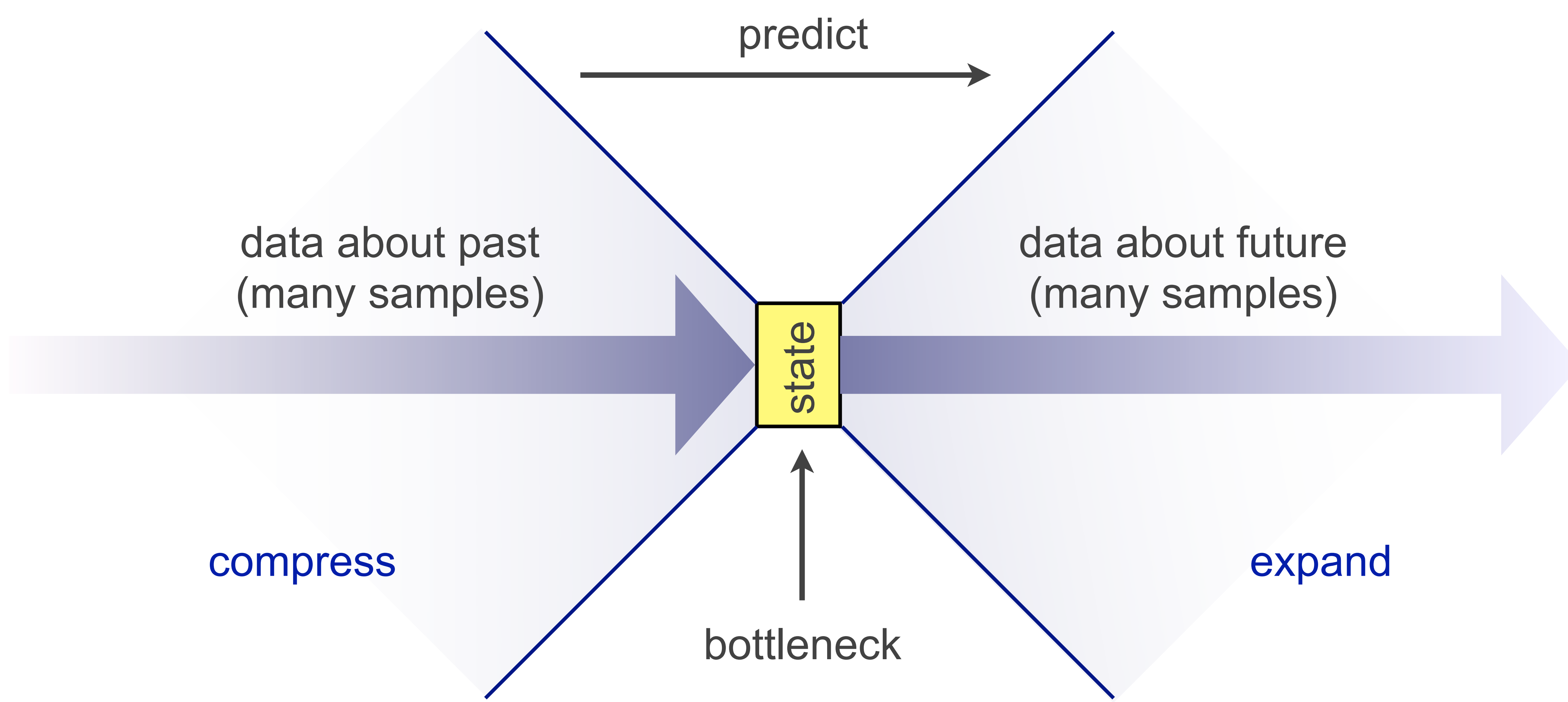
- $E([\varphi(o_{t+1}) \ldots \varphi(o_{t+k})] \mid s_t)$ is our state rep'n
  - interpretable
  - observable—a natural target for learning

Predictive state: summary

- $E([\varphi(o_{t+1}) \ldots \varphi(o_{t+k})] \mid s_t)$ is our state rep'n
    - interpretable
    - observable—a natural target for learning

- ***To find s_t, predict $\varphi(o_{t+1}) \ldots \varphi(o_{t+k})$ from $o_{1:t}$***

Rank bottleneck



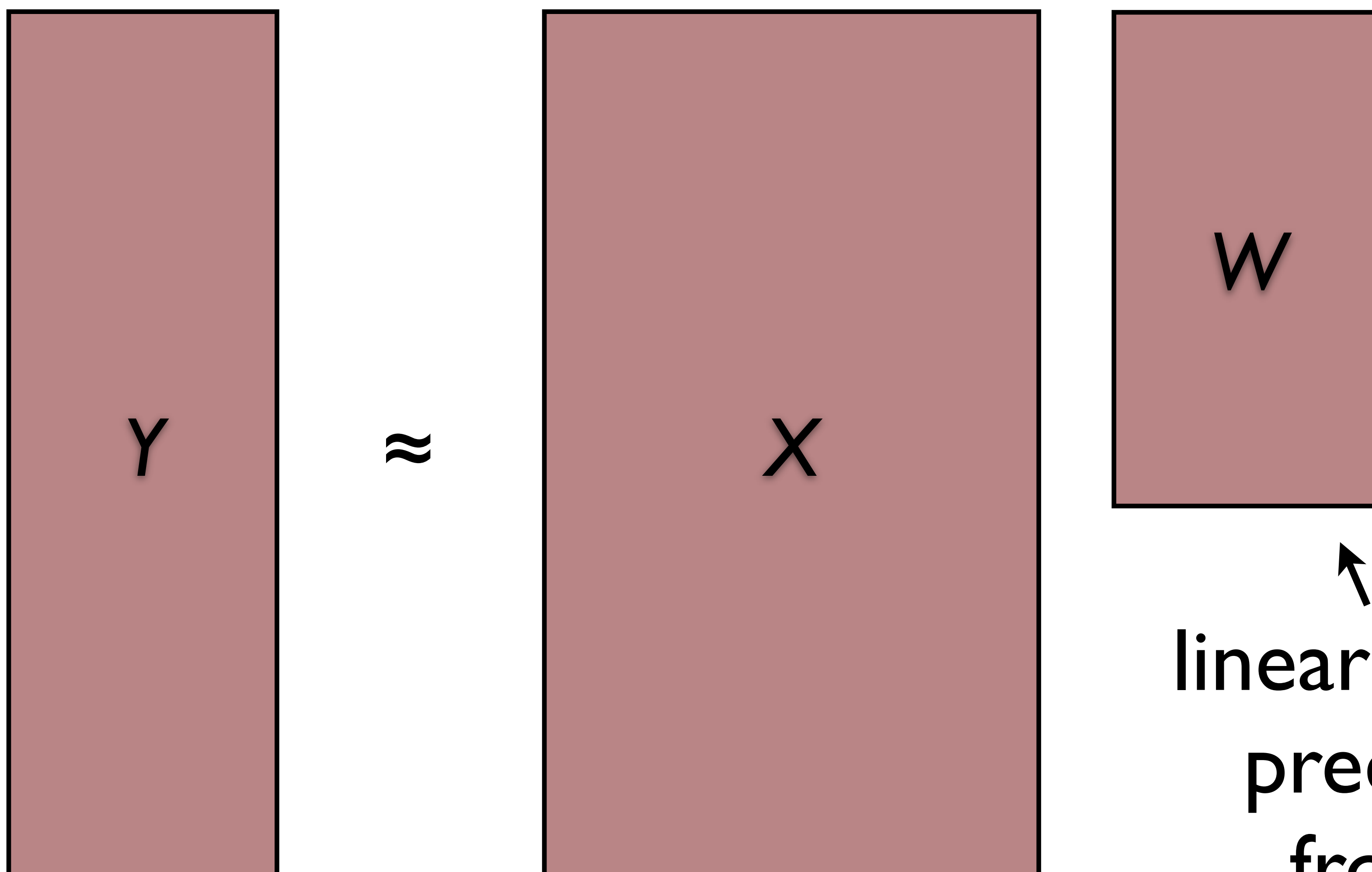
Can find best rank-k bottleneck via matrix factorization ⇒ ***spectral*** method

Finding a compact predictive state

$$Y \approx X \quad W$$

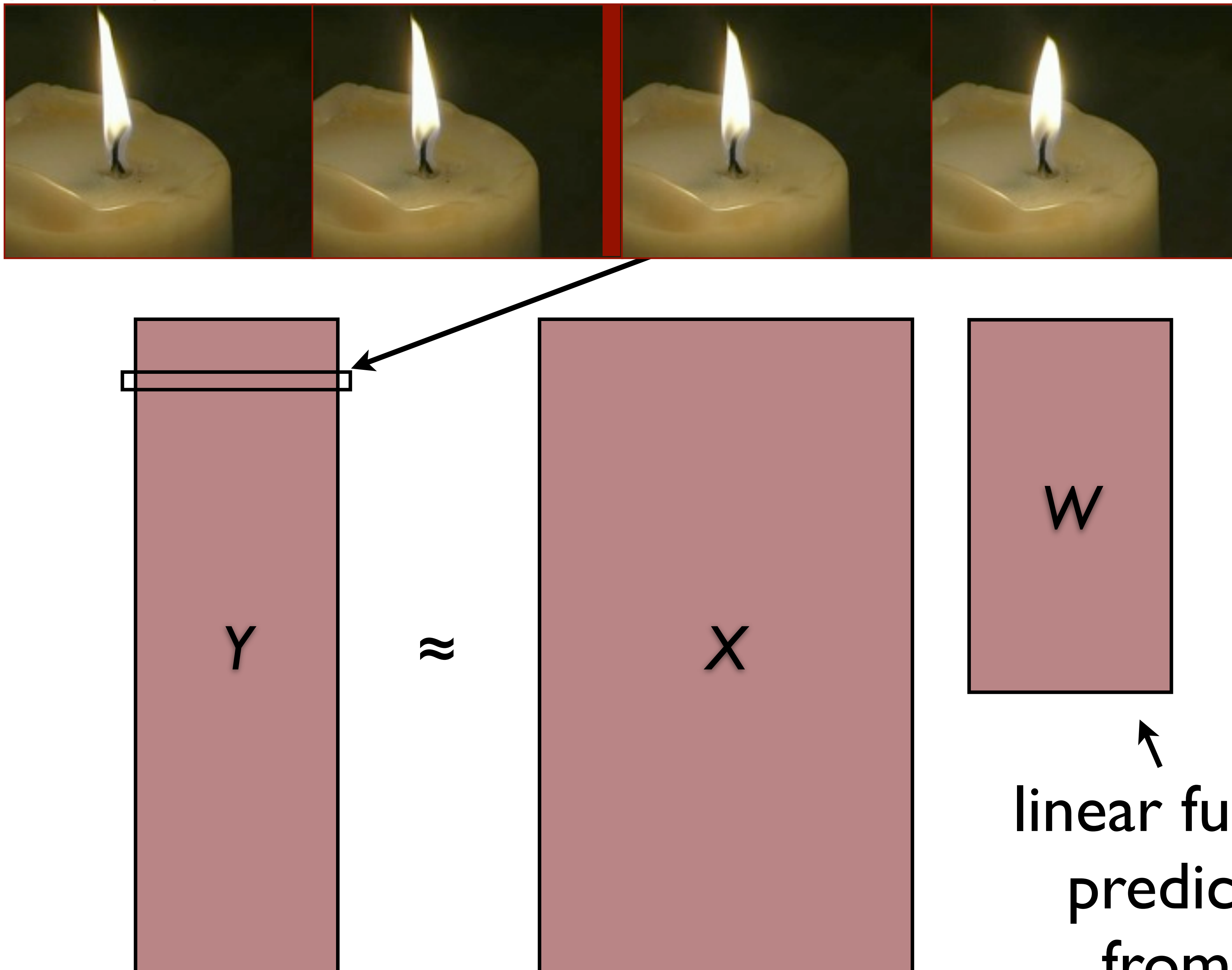
linear function
predicts Y_t
from X_t

Finding a compact predictive state



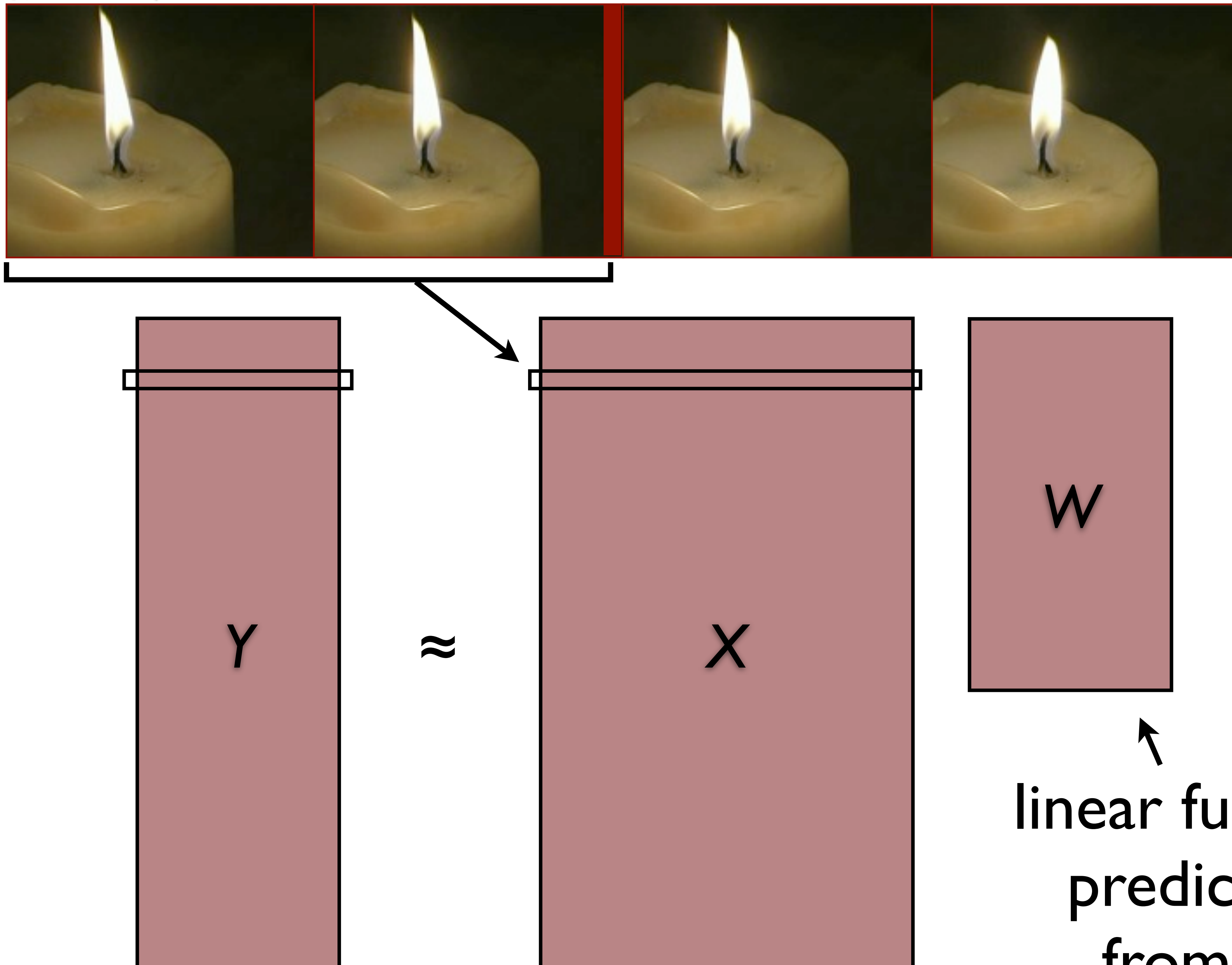
linear function predicts Y_t from X_t

Finding a compact predictive state



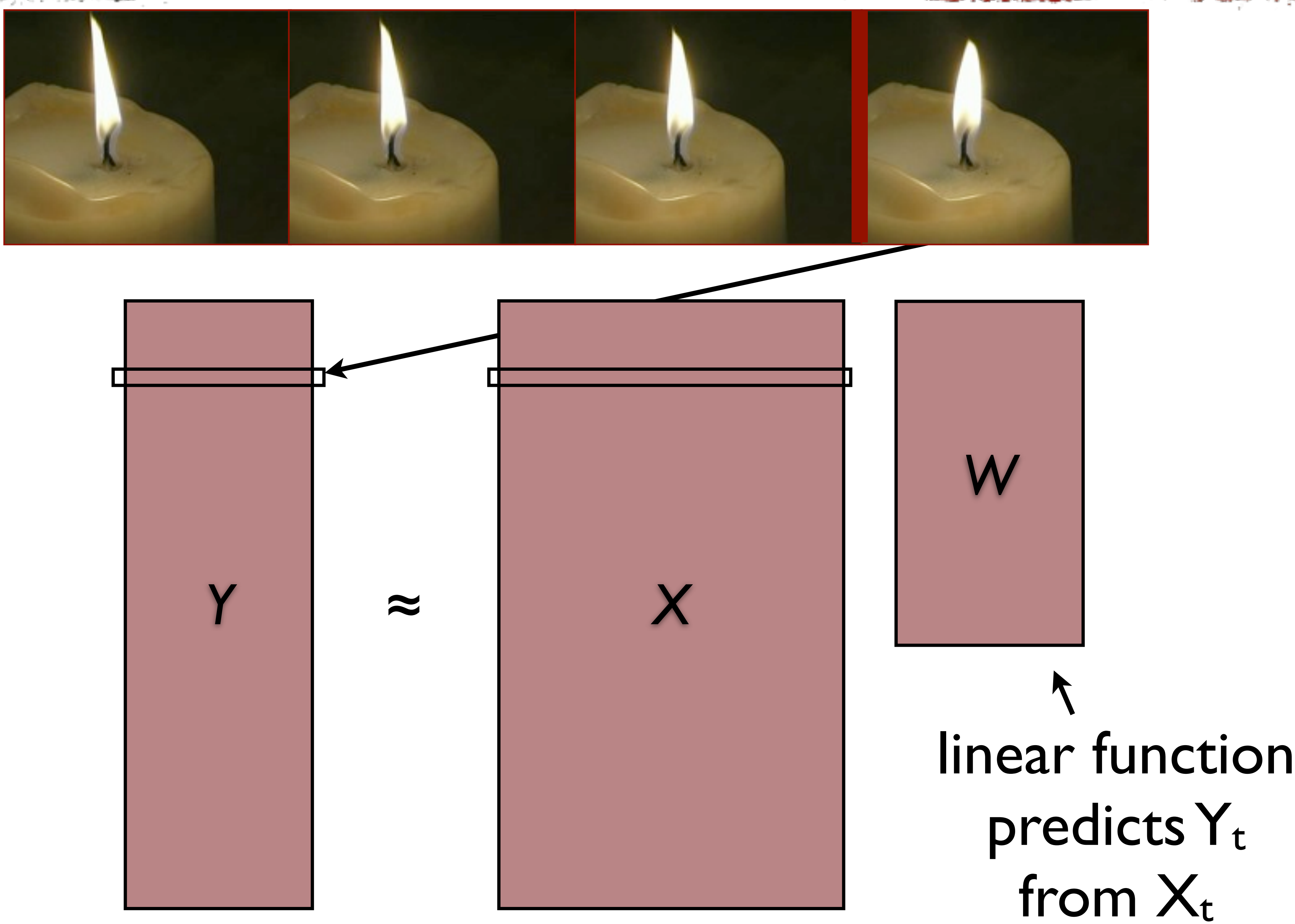
linear function predicts Y_t from X_t

Finding a compact predictive state

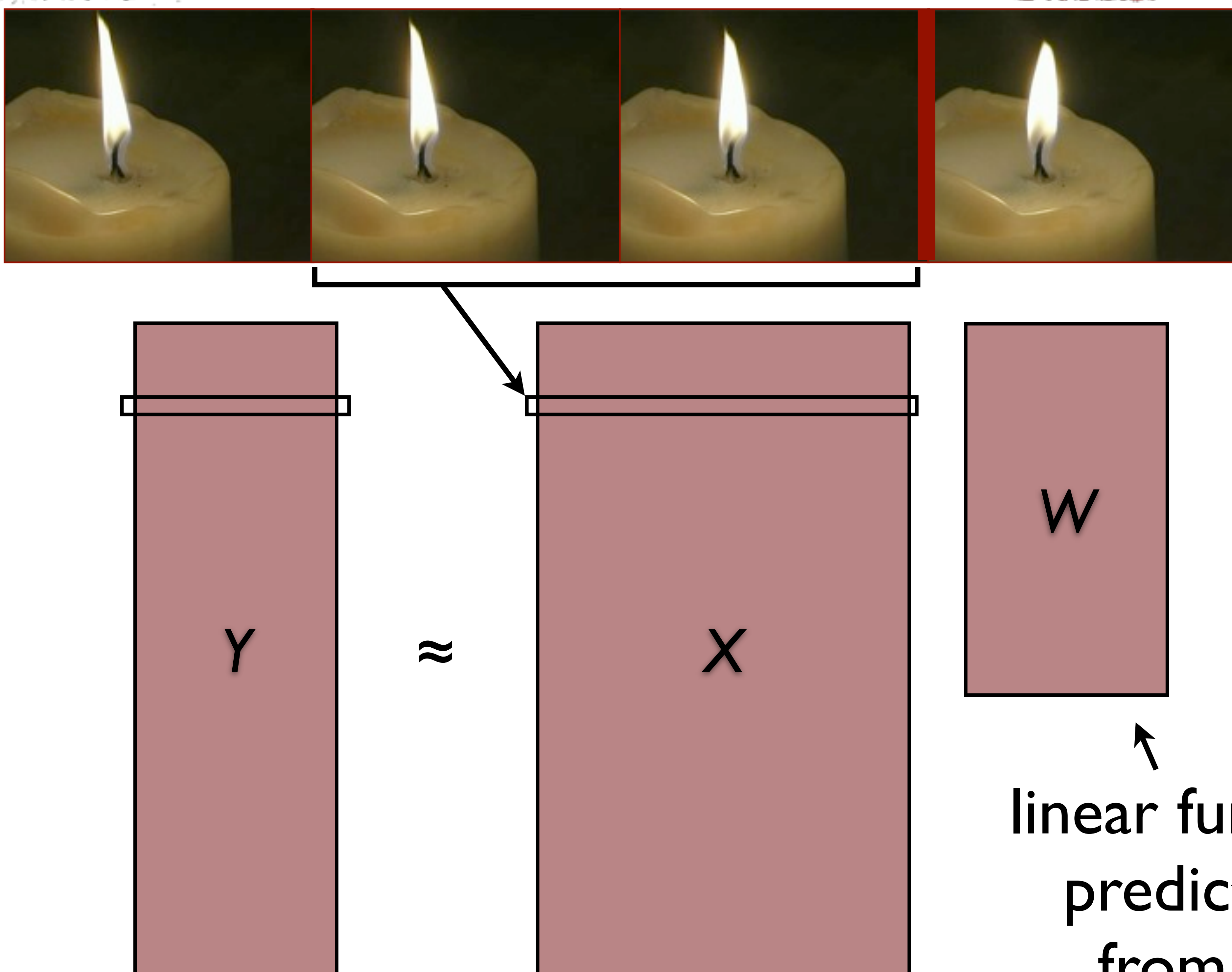


linear function predicts Y_t from X_t

Finding a compact predictive state

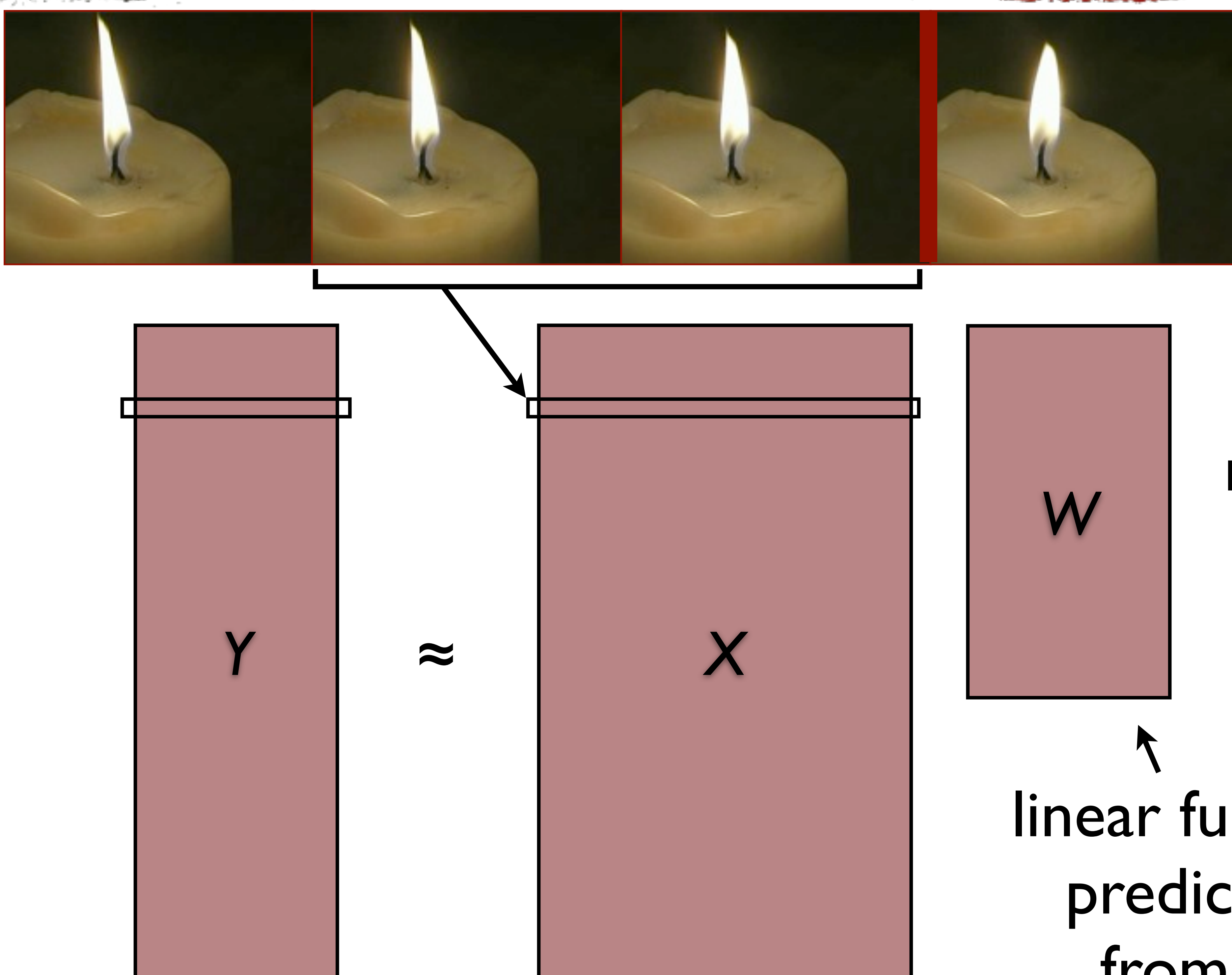


Finding a compact predictive state



linear function predicts Y_t from X_t

Finding a compact predictive state

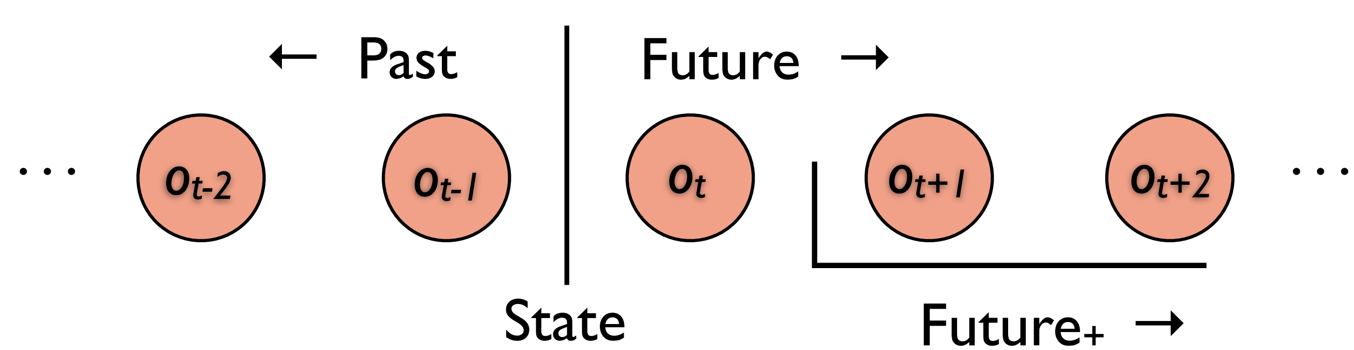


rank(W) ≤ k

linear function
predicts Y_t
from X_t

Observable representation

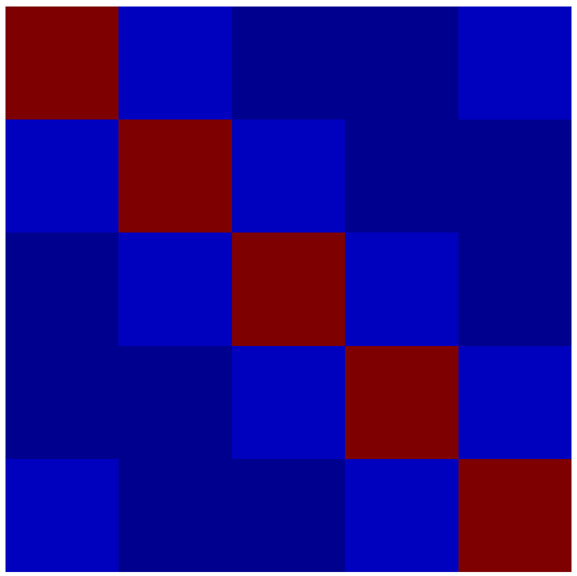
- Now that we have a compact predictive state, need to estimate fns $\Sigma_o(s_t)$ and $\Sigma_{so}(s_t)$
- Insight: parameters are now ***observable***: the problem is just to estimate some covariances from data
    - to get Σ_o, regress $(\varphi(o) \times \varphi(o)) \leftarrow$ state
    - to get Σ_{so}, regress $(\varphi(o) \times \text{future}_+) \leftarrow$ state

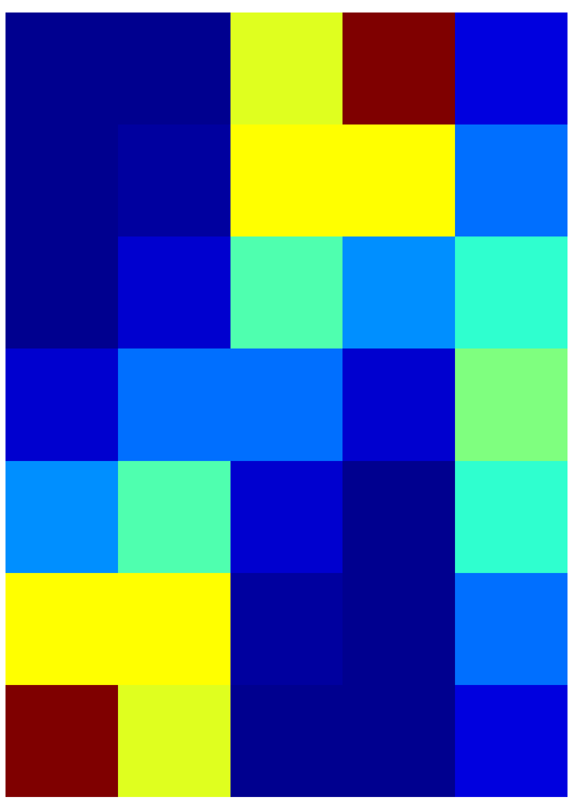


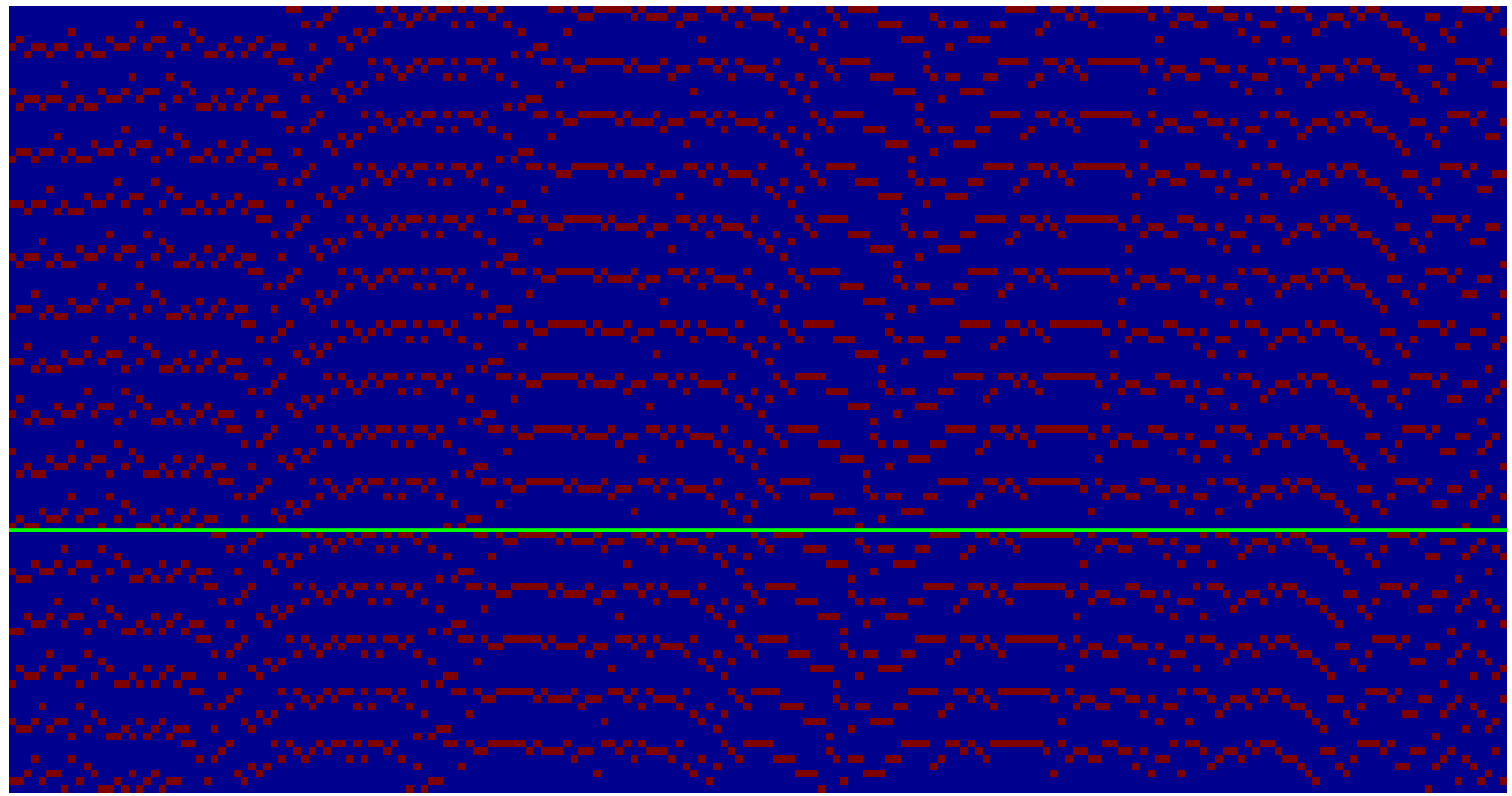
Algorithm

- Find compact predictive state: regress future ← past
  - use any features of past/future, or a kernel, or even a learned kernel (manifold case)
  - constrain rank of prediction weight matrix
- Extract model: regress
  - $(o \times \text{future}_+) \leftarrow$ [predictive state]
  - $(o \times o) \leftarrow$ [predictive state]
  - future_+: use same rank-constrained basis as for predictive state

Does it work?





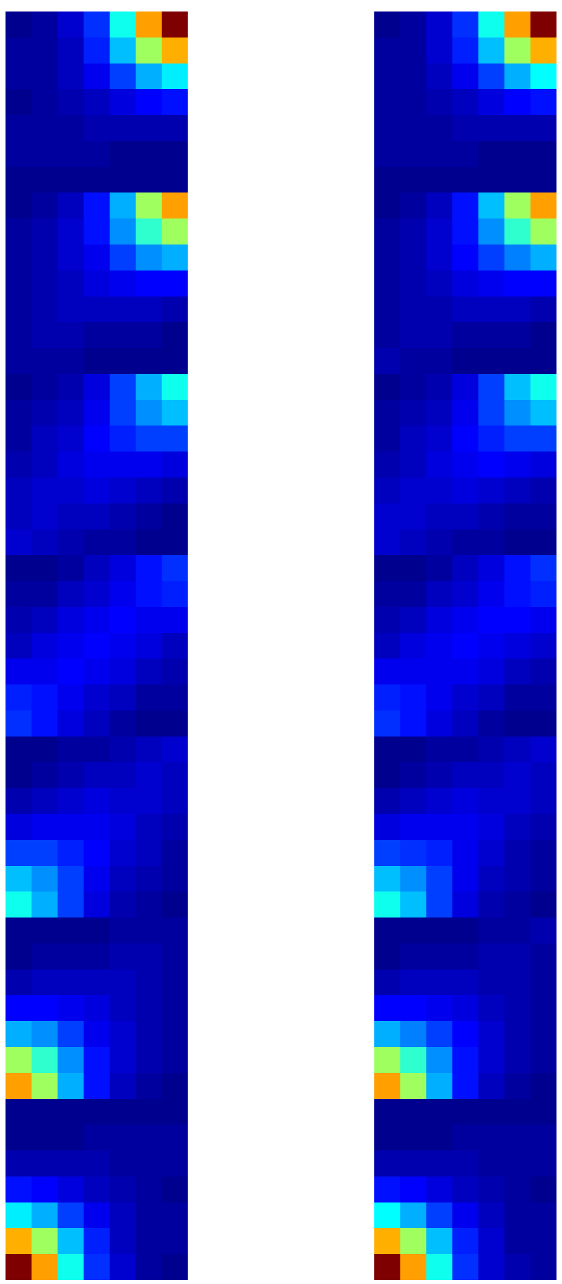


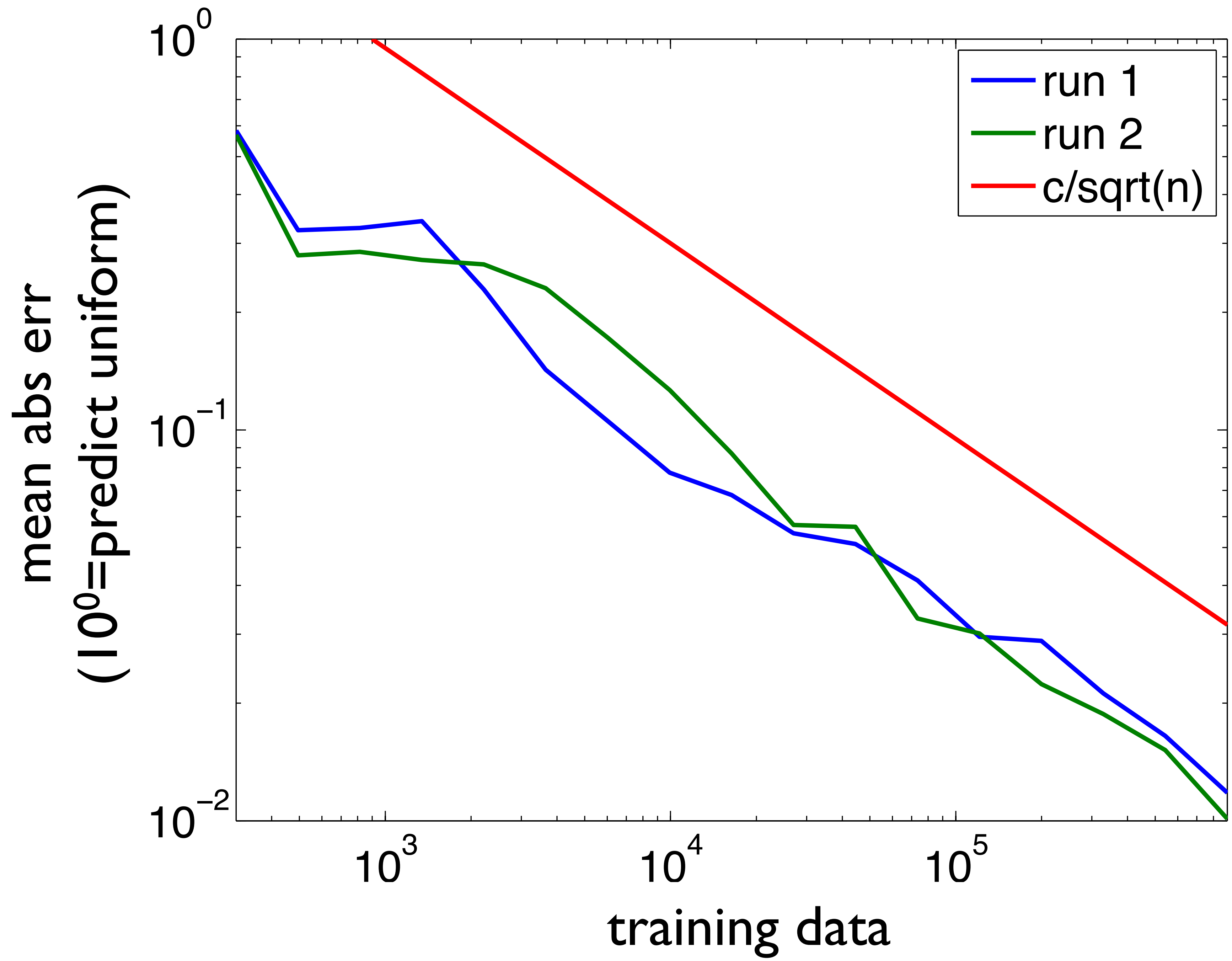
- Simple HMM: 5 states, 7 observations
- N=300 thru N=900k

Does it work?

Model True

P(o1, o2, o3)

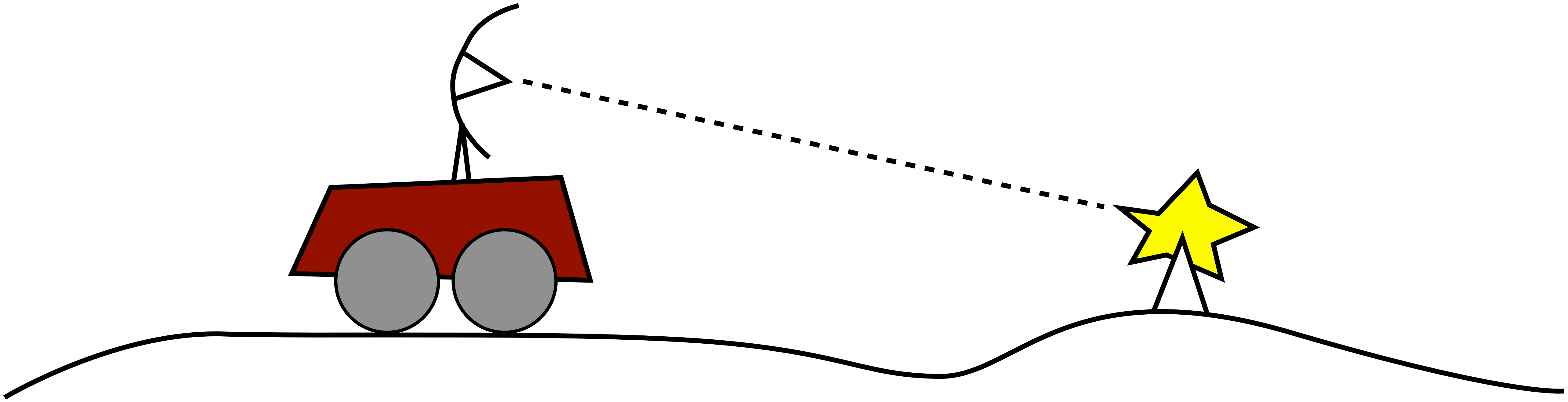




Discussion

- ***Impossibility***—learning DFA in poly time = breaking crypto primitives [Kearns & Valiant 89]
  - so, clearly, we can't always be statistically efficient
  - but, see McDonald [11], HKZ [09], us [09]: convergence depends on ***mixing rate***
- ***Nonlinearity***—Bayes filter update is highly nonlinear in state (matrix inverse), even though we use a *linear* regression to identify the model
  - nonlinearity is essential for expressiveness

Range-only SLAM



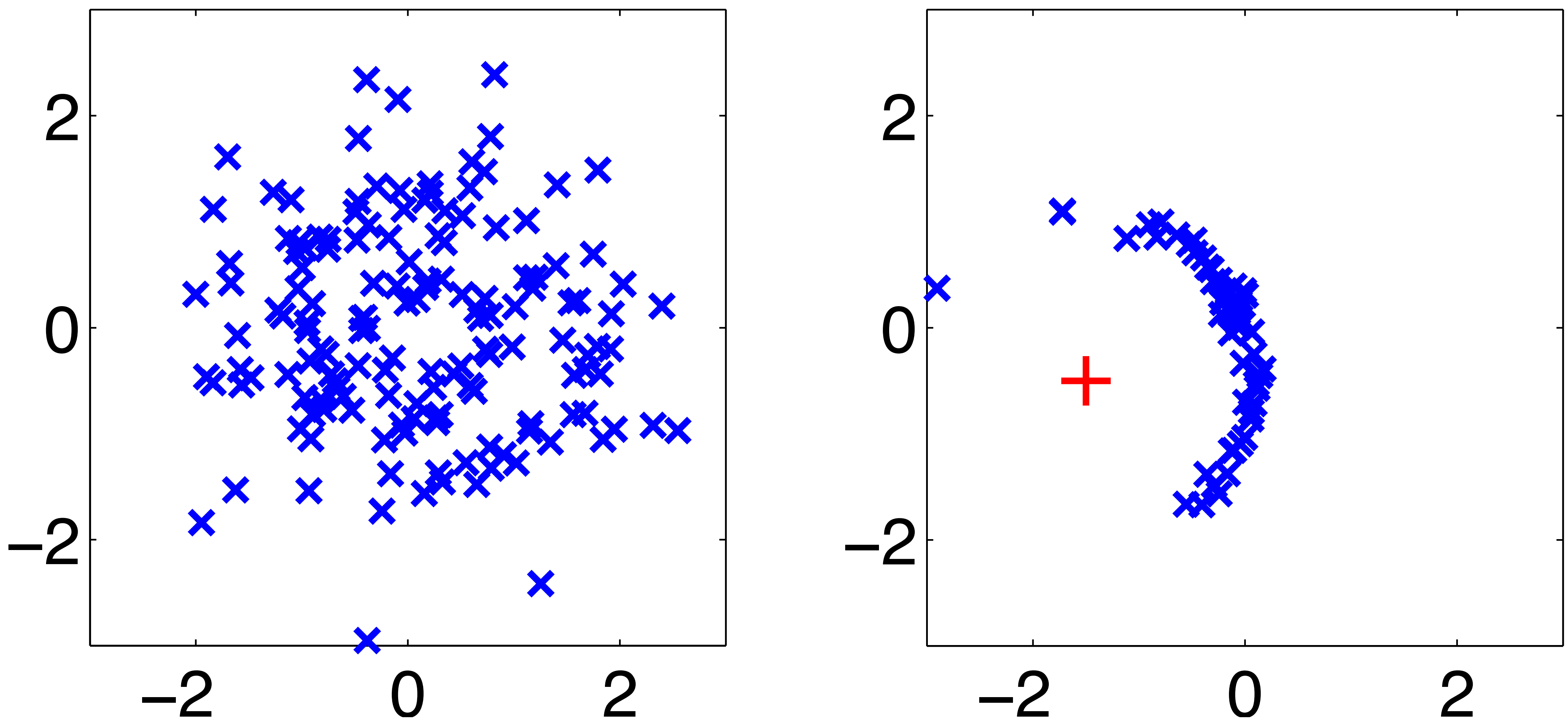
- Robot measures distance from current position to fixed beacons (e.g., time of flight or signal strength)
  - may also have odometry
- Goal: recover robot path, landmark locations

Typical solution: EKF

Bayes filter vs. EKF

Prior

After 1 range measurement to known landmark



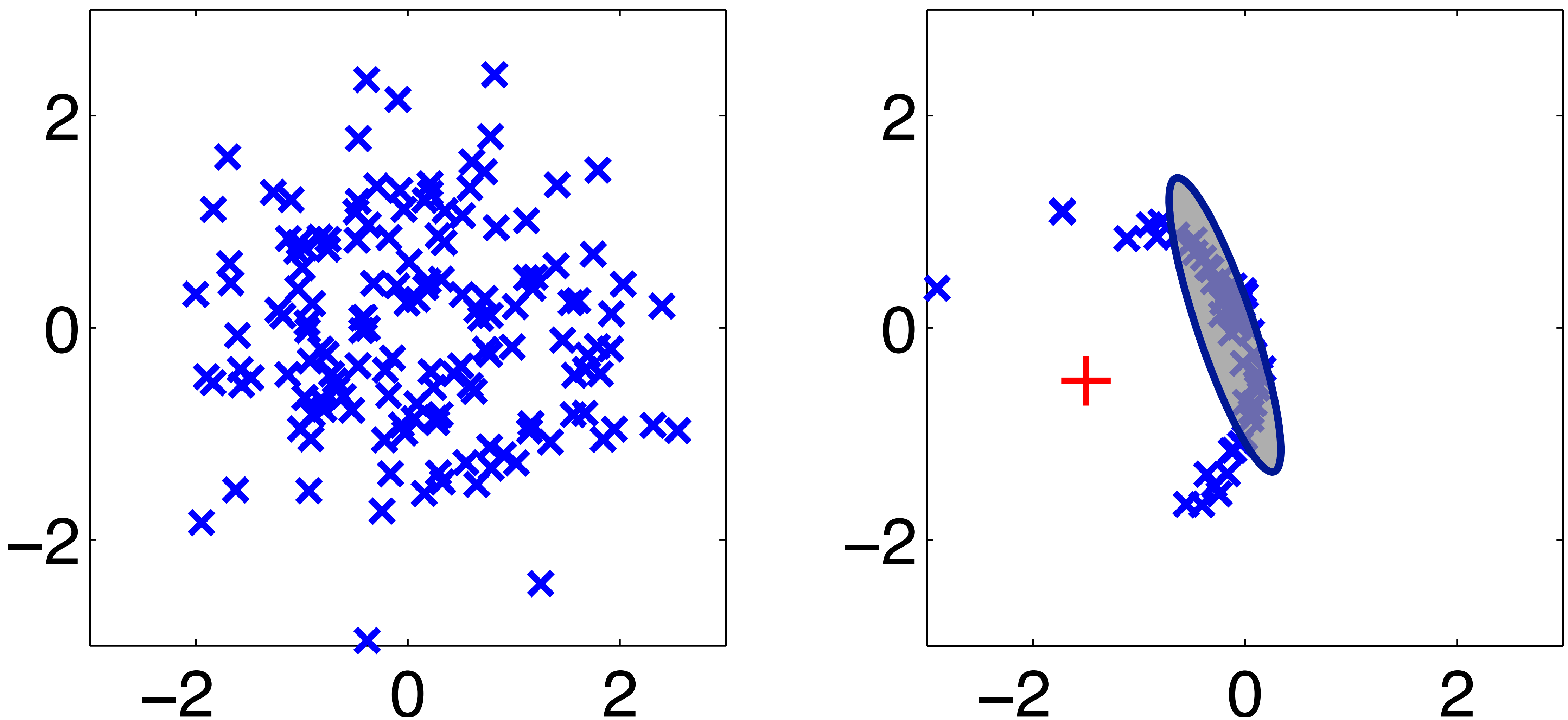
- Problem: linear/Gaussian approximation very bad

Typical solution: EKF

Bayes filter vs. EKF

Prior

After 1 range measurement to known landmark

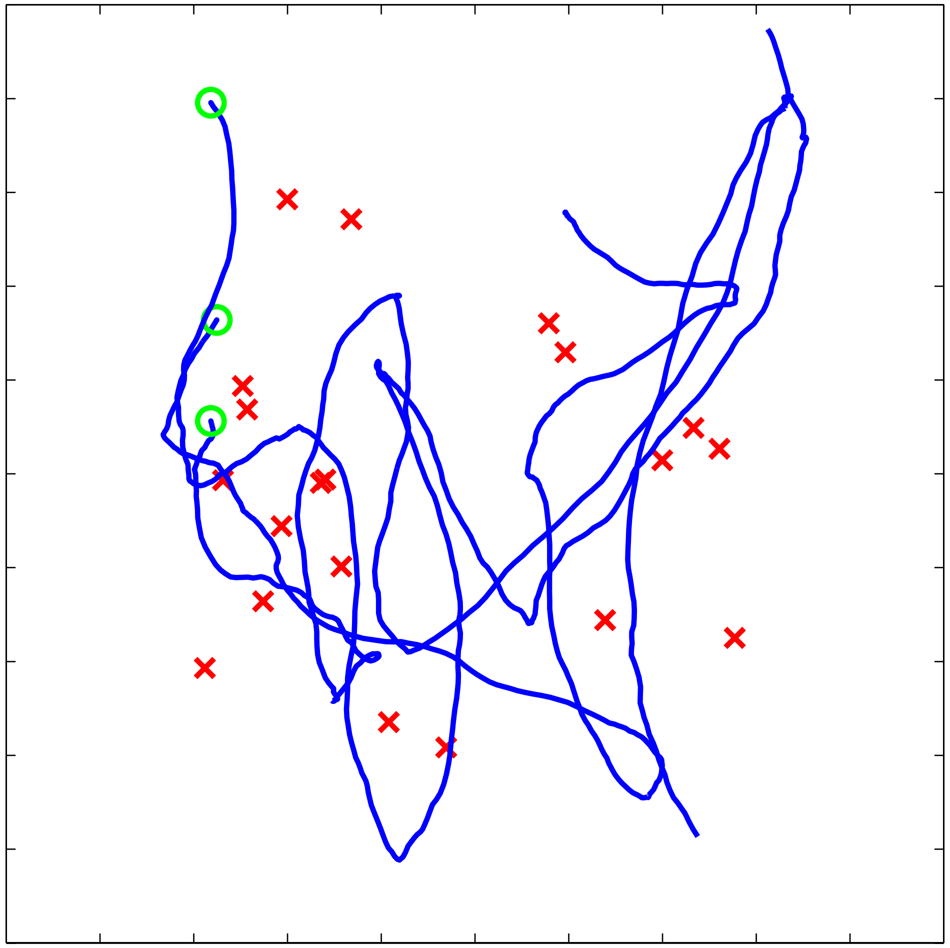


- Problem: linear/Gaussian approximation very bad

Spectral solution (simple version)

$$
\begin{array}{cccc}
 & \textit{landmark 1, time step 2} & & \\
\end{array}
$$

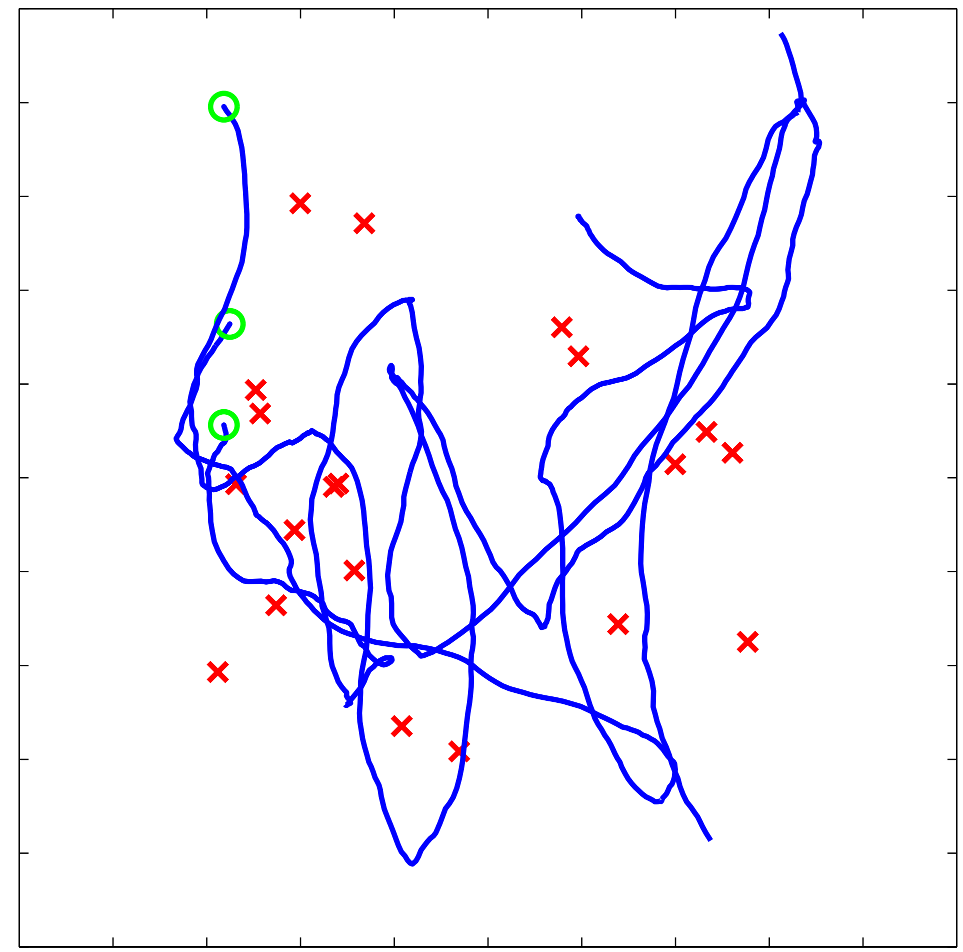
$$
\begin{bmatrix}
d_{11}^2 & \boxed{d_{12}^2} & \cdots & d_{1T}^2 \\
d_{21}^2 & d_{22}^2 & \cdots & d_{2T}^2 \\
\vdots & \vdots & \vdots & \vdots \\
d_{N1}^2 & d_{N2}^2 & \cdots & d_{NT}^2
\end{bmatrix}
$$



Spectral solution (simple version)

landmark 1, time step 2

$$1/2 \begin{bmatrix} d_{11}^2 & \boxed{d_{12}^2} & \cdots & d_{1T}^2 \\ d_{21}^2 & d_{22}^2 & \cdots & d_{2T}^2 \\ \vdots & \vdots & \vdots & \vdots \\ d_{N1}^2 & d_{N2}^2 & \cdots & d_{NT}^2 \end{bmatrix}$$

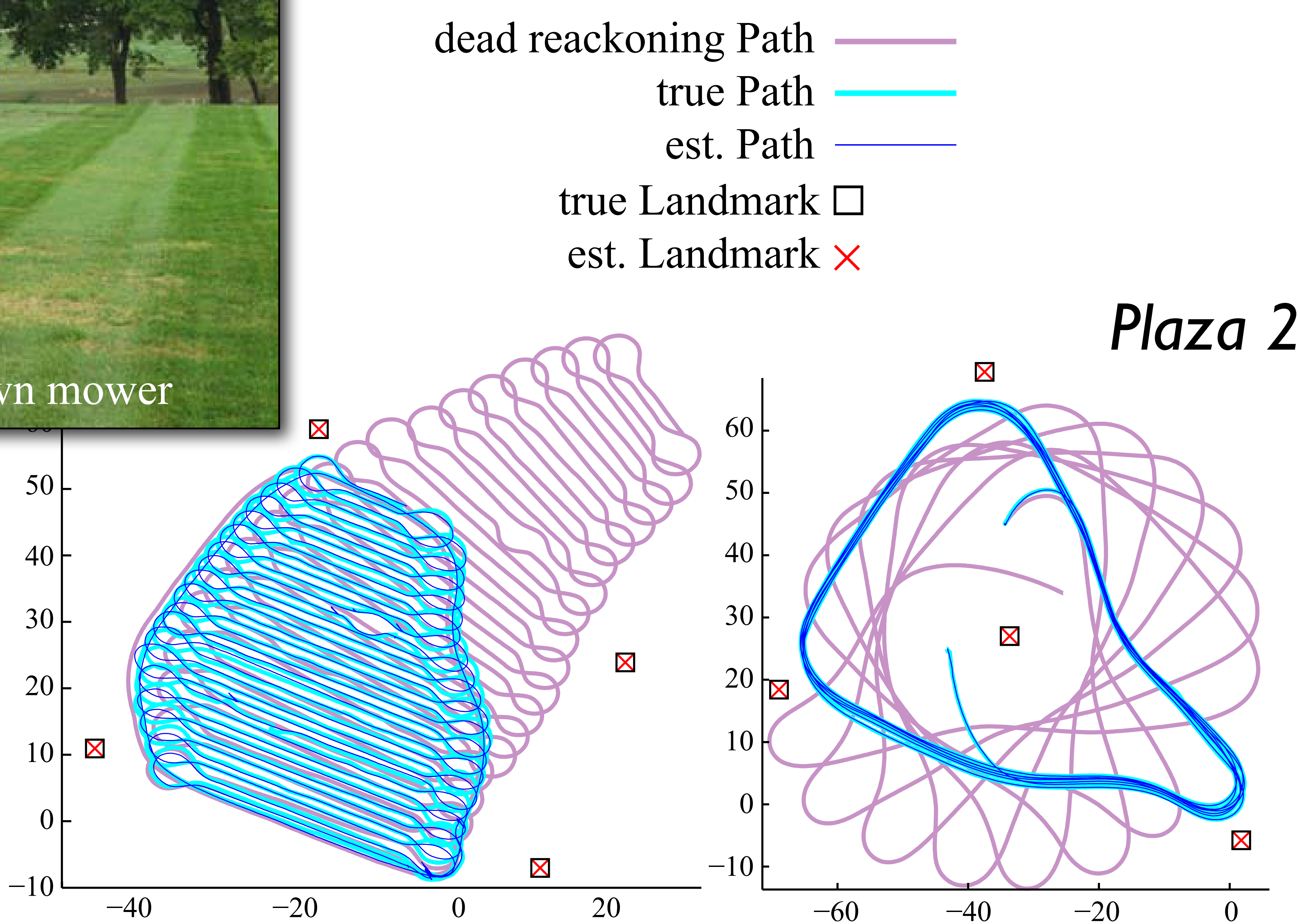


landmark position x, y

$$= \begin{bmatrix} 1 & x_1 & y_1 & (x_1^2 + y_1^2)/2 \\ 1 & x_2 & y_2 & (x_2^2 + y_2^2)/2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & y_N & (x_N^2 + y_N^2)/2 \end{bmatrix} \begin{bmatrix} (u_1^2 + v_1^2)/2 & (u_2^2 + v_2^2)/2 & \cdots & (u_T^2 + v_T^2)/2 \\ -u_1 & -u_2 & \cdots & u_T \\ -v_1 & -v_2 & \cdots & v_T \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

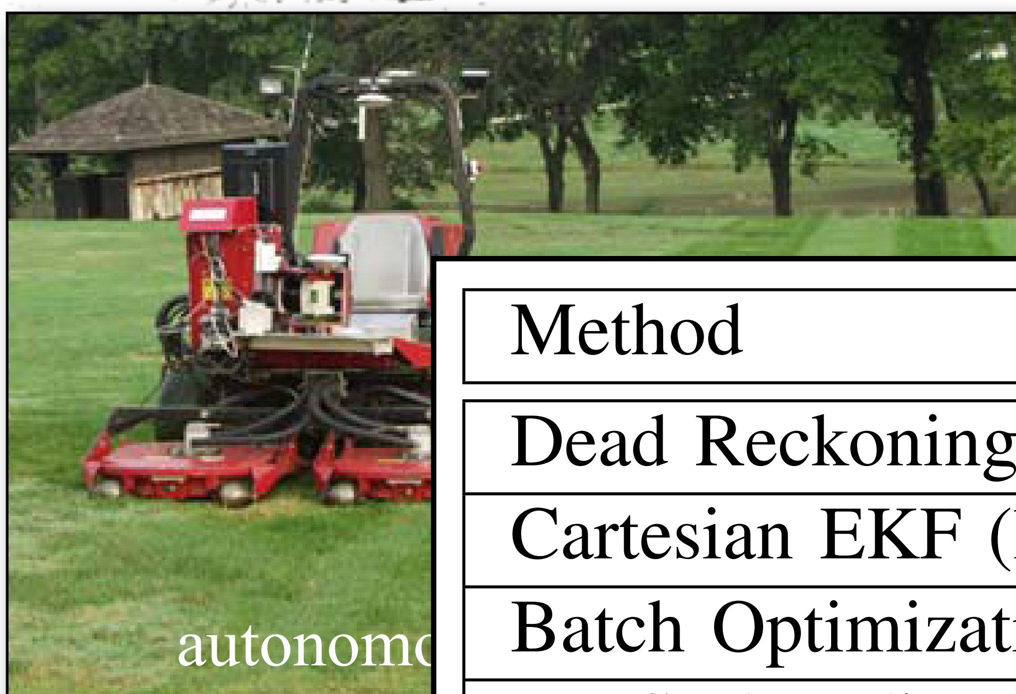
robot position u, v

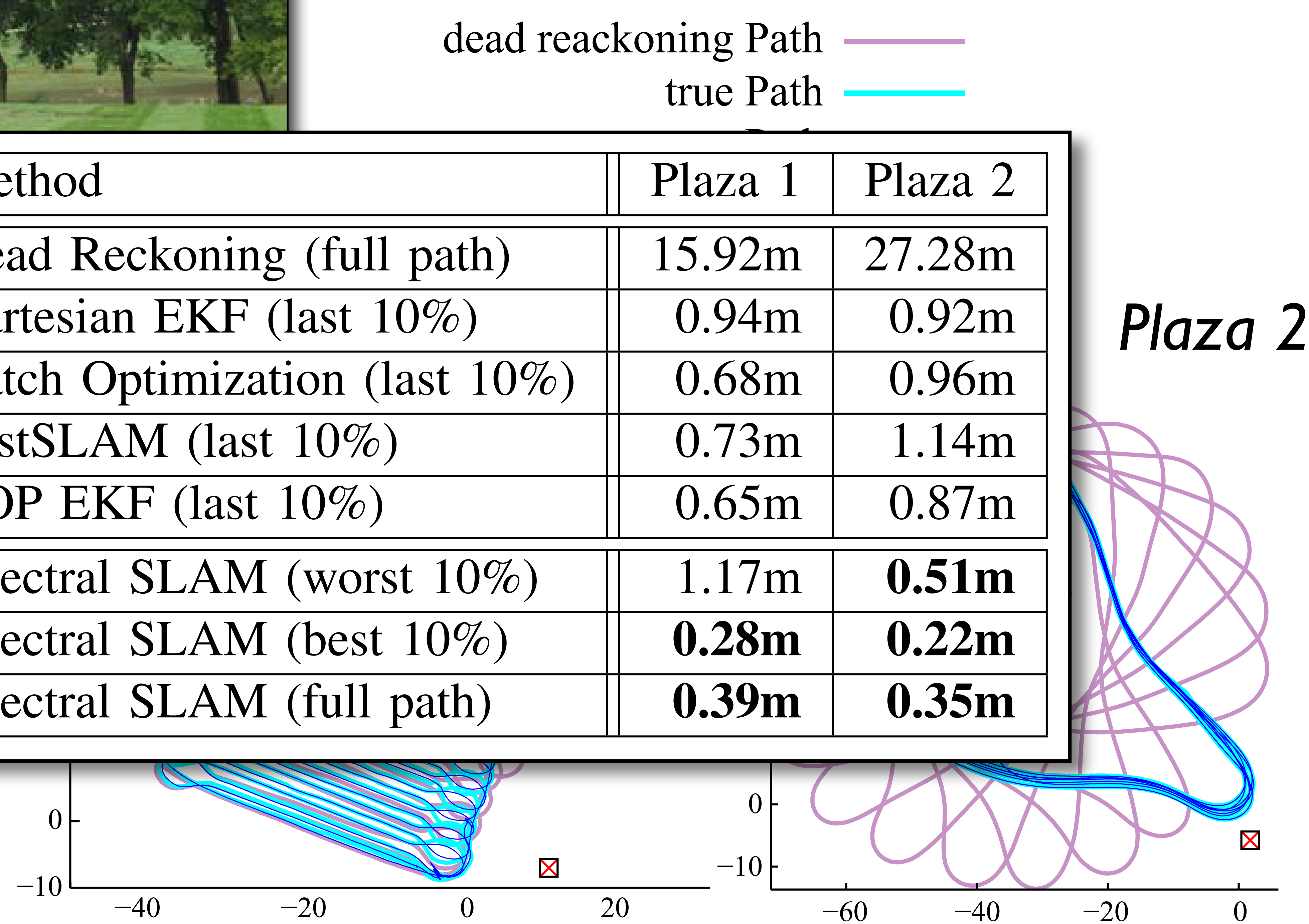
Experiments (full version)



[Djugash & Singh, 2008]

Experiments (full version)

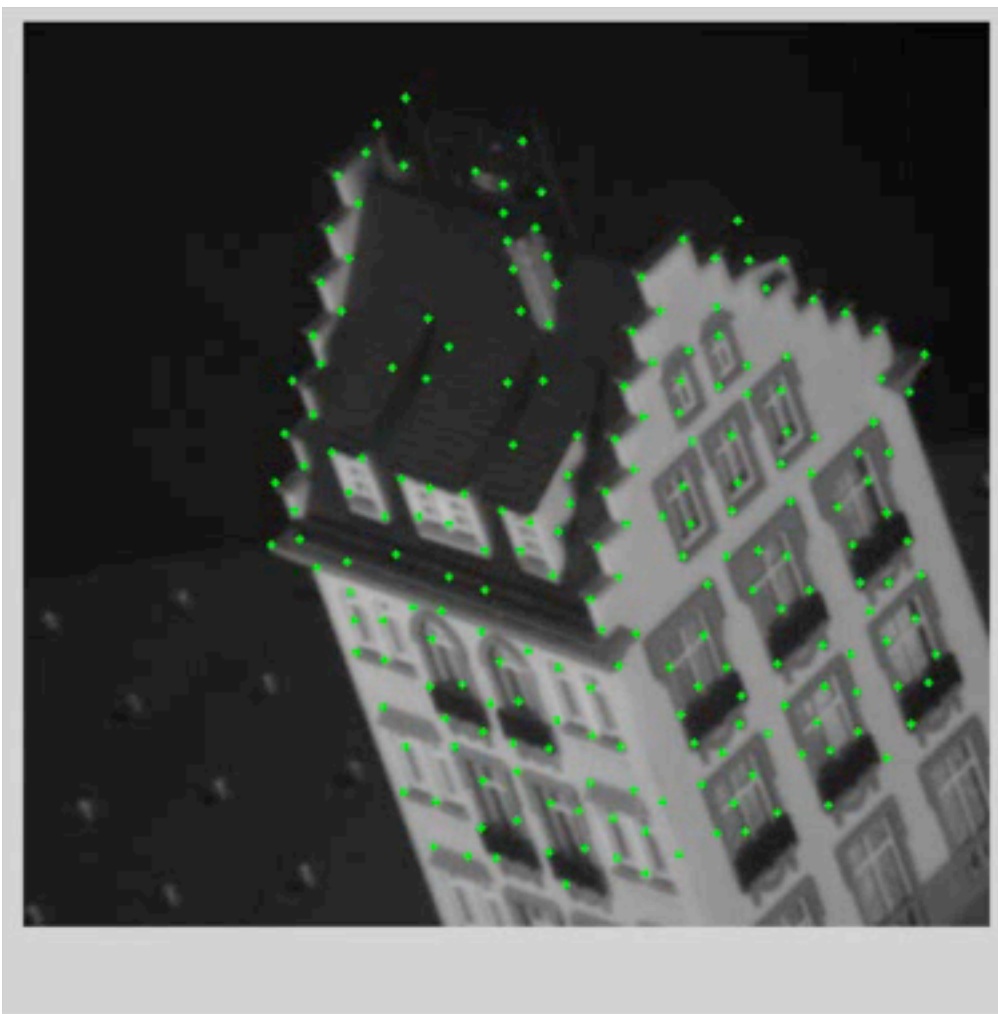


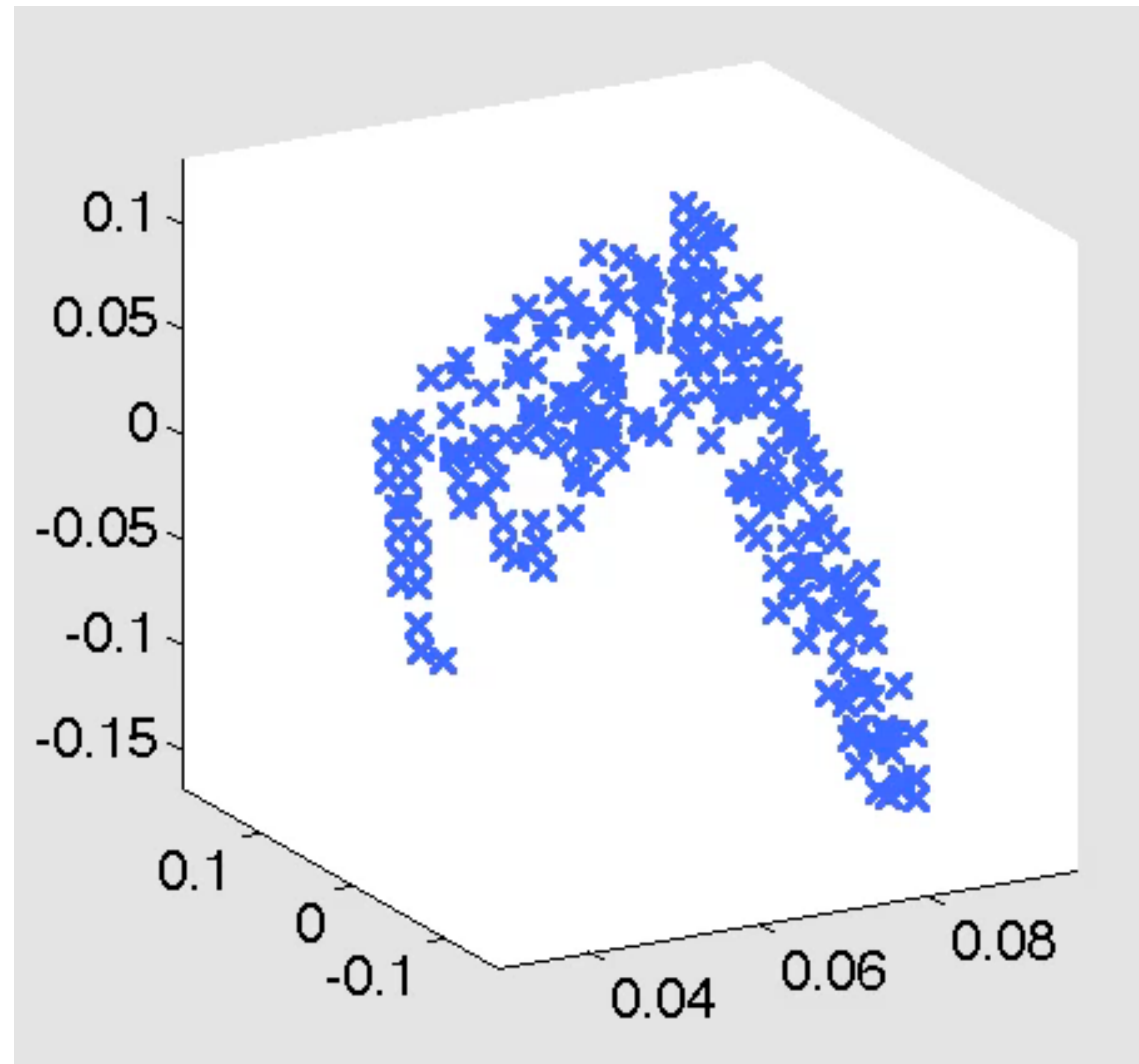


Method	Plaza 1	Plaza 2
Dead Reckoning (full path)	15.92m	27.28m
Cartesian EKF (last 10%)	0.94m	0.92m
Batch Optimization (last 10%)	0.68m	0.96m
FastSLAM (last 10%)	0.73m	1.14m
ROP EKF (last 10%)	0.65m	0.87m
Spectral SLAM (worst 10%)	1.17m	**0.51m**
Spectral SLAM (best 10%)	**0.28m**	**0.22m**
Spectral SLAM (full path)	**0.39m**	**0.35m**

[Djugash & Singh, 2008]

Structure from motion





[Tomasi & Kanade, 1992]

Video textures

- Kalman filter works for some ***video textures***
  - steam grate example above
  - fountain:

observation = raw pixels (vector of reals over time)

Video textures, redux

Original Kalman Filter PSR

both models: 10 latent dimensions

Video textures, redux

Original

Kalman Filter

PSR

both models: 10 latent dimensions

Learning in the loop: option pricing

- Price a financial derivative: "psychic call"
  - holder gets to say "I bought call 100 days ago"
  - underlying stock follows Black Scholes (unknown parameters)

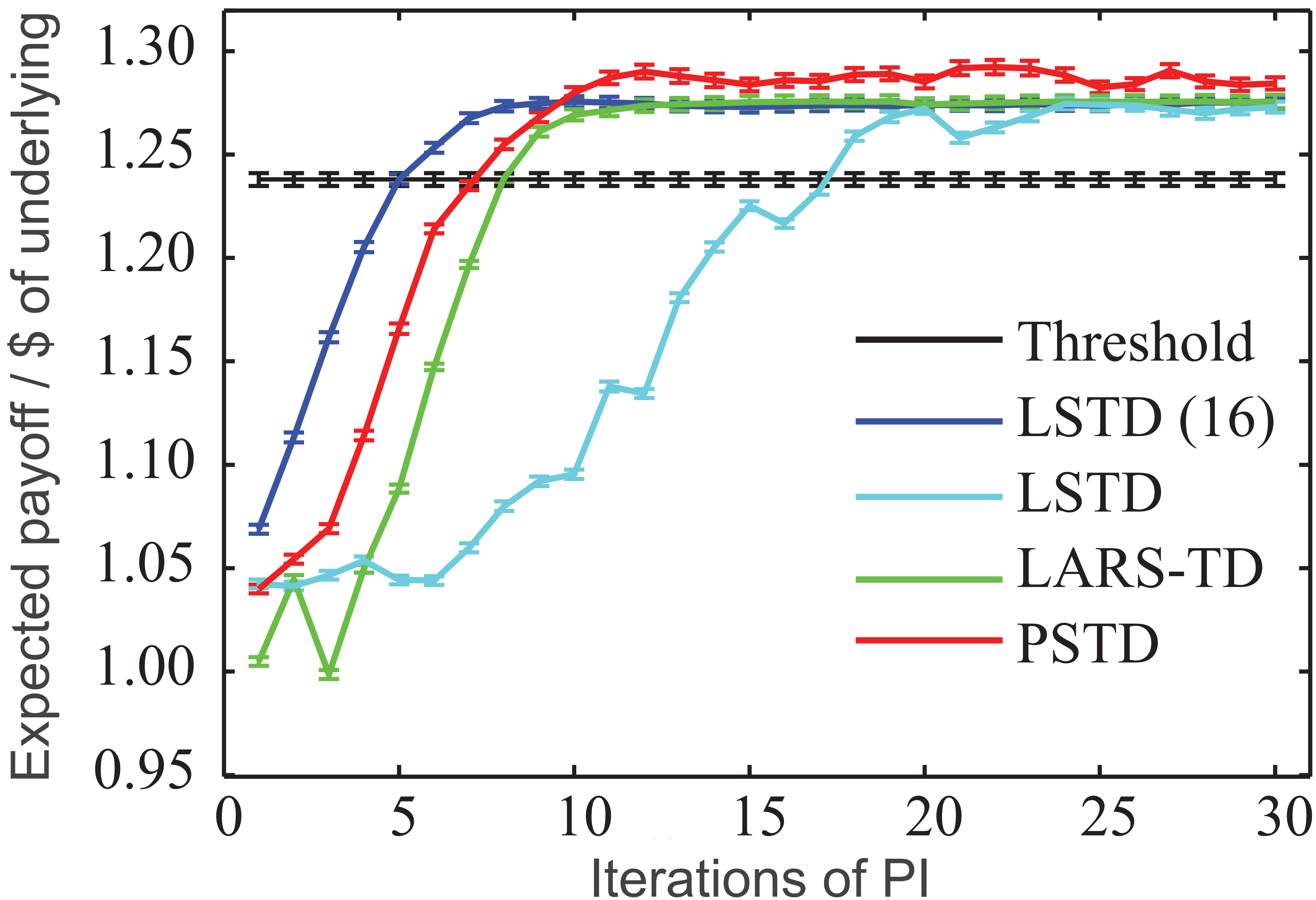
Option pricing

- Solution [Van Roy et al.]: use policy iteration
  - 16 hand-picked features (e.g., poly · history)
  - initialize policy arbitrarily
  - least-squares temporal differences (LSTD) to estimate value function
  - policy := greedy; repeat

Option pricing

- Better solution: spectral SSID inside policy iteration
  - 16 original features from Van Roy et al.
  - 204 additional "low-originality" features
    - e.g., linear fns of price history of underlying
  - SSID picks best 16-d dynamics to explain feature evolution
  - solve for value function in closed form

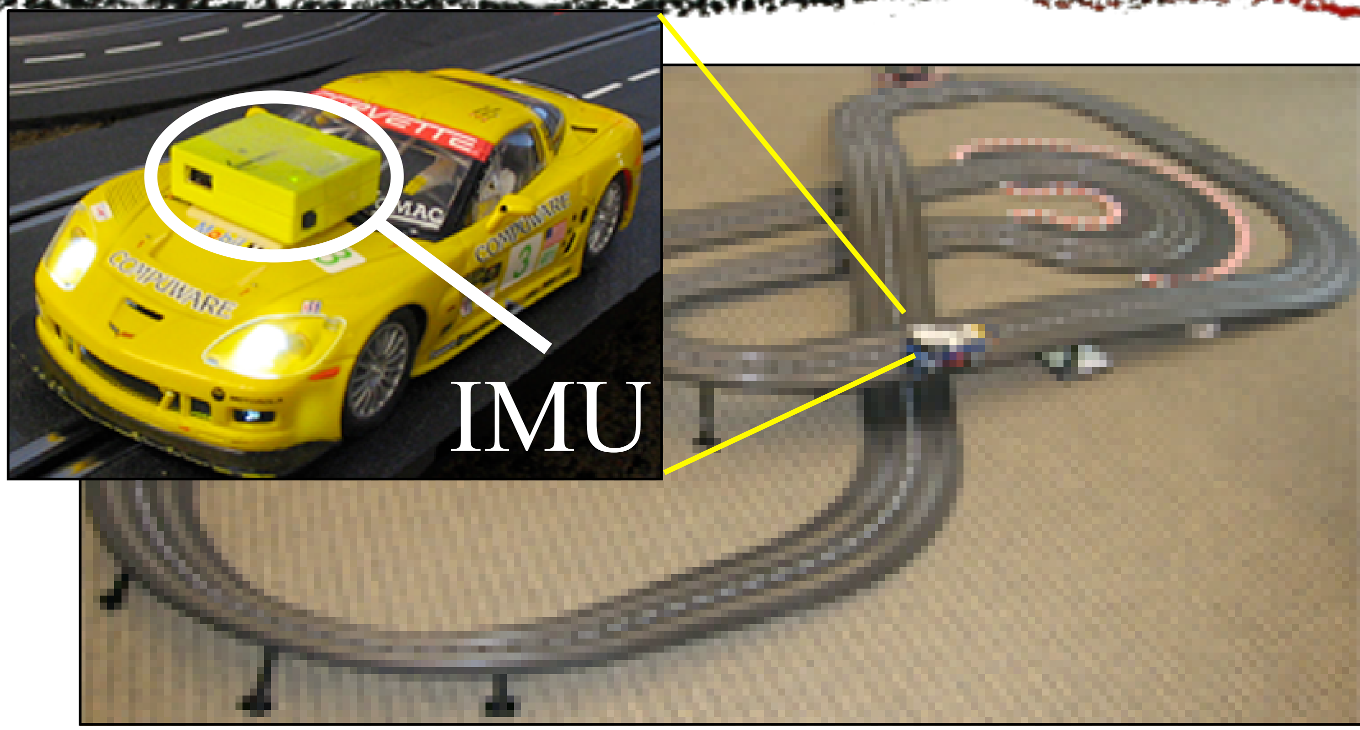
Policy iteration w/ spectral learning

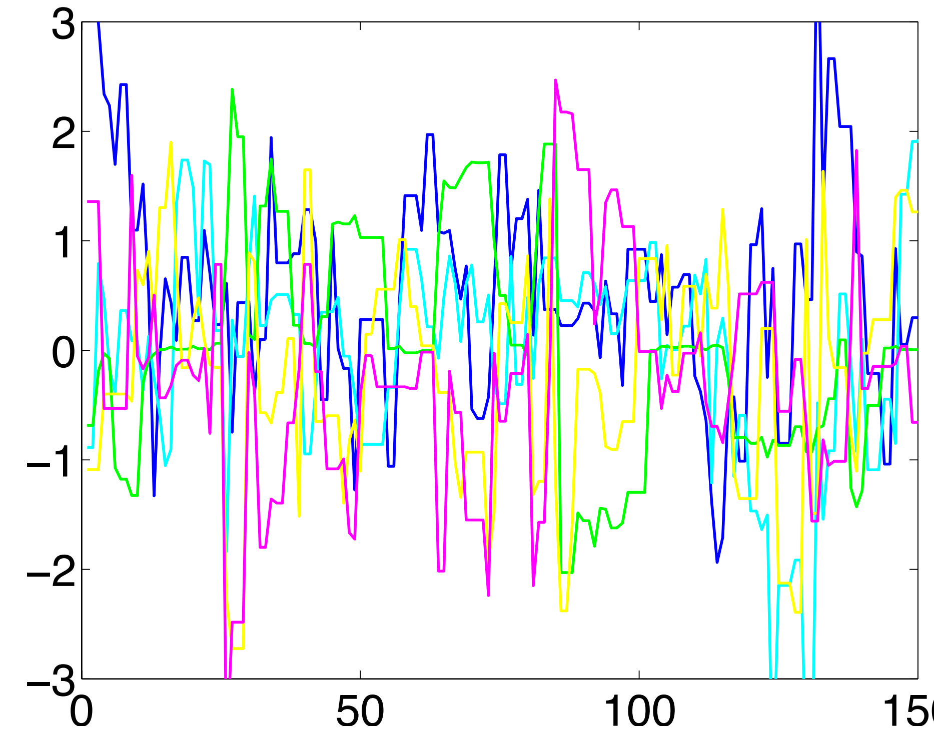


0.82¢/$ better than best competitor; 1.33¢/$ better than best previously published

data: 1,000,000 time steps

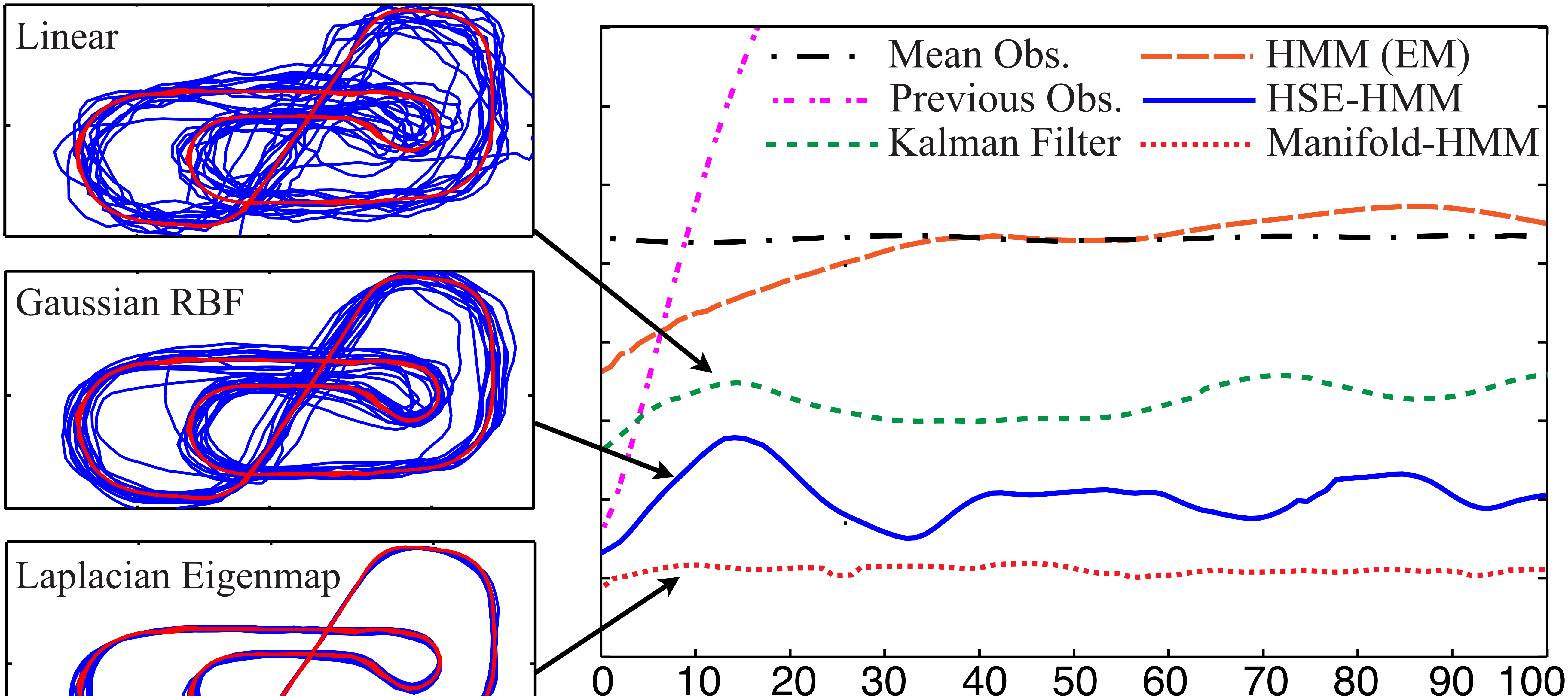
Slot-car IMU data





[Ko & Fox, 09; Boots et al., 10]

Kalman < HSE-HMM < manifold



comparison of kernels

RMS prediction error (XY position) v. horizon

Algorithm intuition

- Use separate one-manifold learners to estimate structure in past, future
  - result: noisy Gram matrices G_X, G_Y
  - "signal" is correlated between G_X, G_Y
  - "noise" is independent
- Look at eigensystem of $G_X G_Y$
  - suppresses noise, leaves signal

Summary

- General class of models for dynamical systems
- Fast, statistically consistent learning method
- Includes many well-known models & algorithms as special cases
  - HMM, Kalman filter, n-gram, PSR, kernel versions
  - SfM, subspace ID, Kalman video textures
- Also includes new models & algorithms that give state-of-the-art performance on interesting tasks
  - range-only SLAM, PSR video textures, HSE-HMM

Papers

- B. Boots and G. *An Online Spectral Learning Algorithm for Partially Observable Nonlinear Dynamical Systems*. AAAI, 2011.
- B. Boots and G. *Predictive state temporal difference learning*. NIPS, 2010.
- B. Boots, S. M. Siddiqi, and G. *Closing the learning-planning loop with predictive state representations*. RSS, 2010.
- L. Song, B. Boots, S. M. Siddiqi, G., and A. J. Smola. *Hilbert space embeddings of hidden Markov models*. ICML, 2010. (Best paper)