Learning to Filter with Predictive State Inference Machines

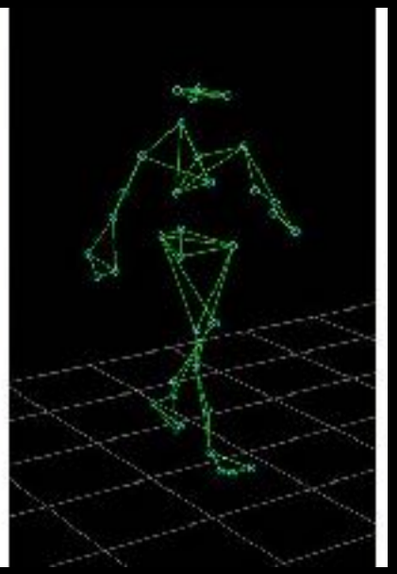
Wen Sun

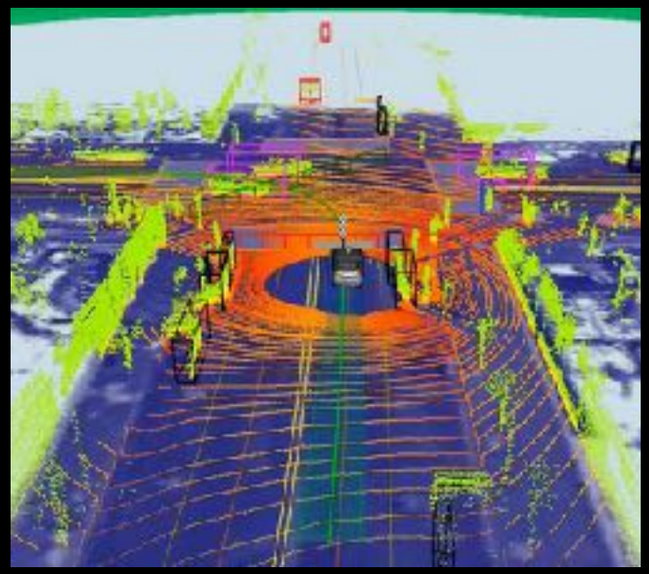
Joint work with Arun Venkatraman, Byron Boots, and Drew Bagnell

Georgia Tech Robotics & Intelligent Machines

Carnegie Mellon University The Robotics Institute

Recursive Bayesian Filtering

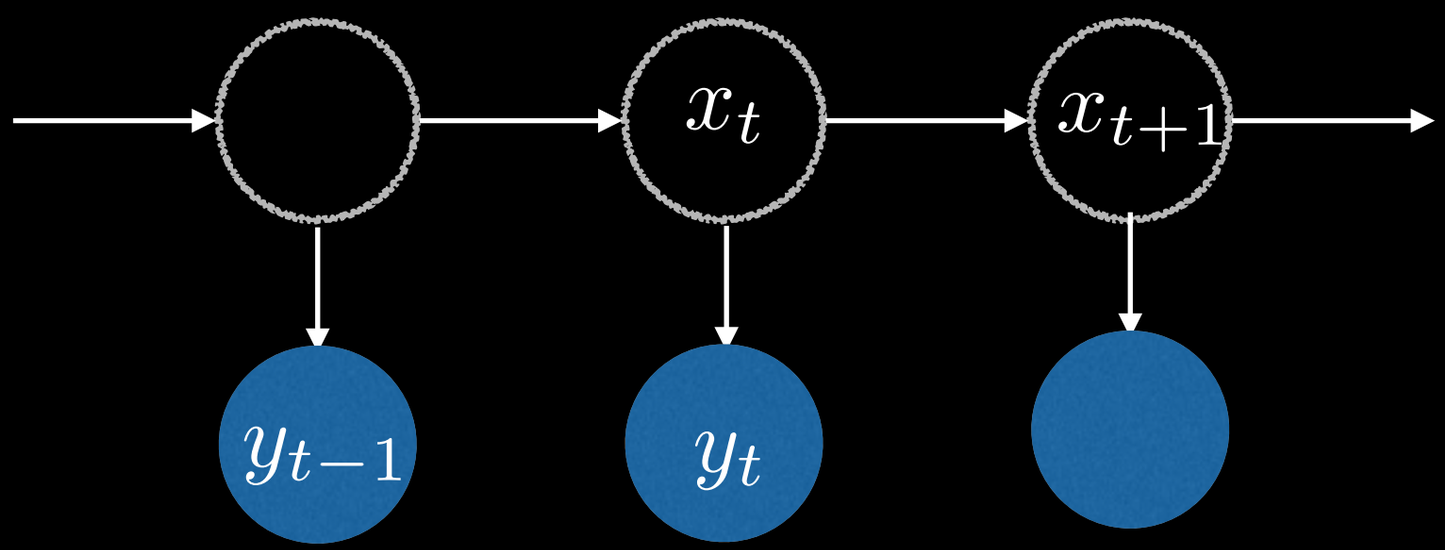




Latent State Space Model

e.g., Hidden Markov Model (HMM)
Linear Dynamical Systems (LDS)

$$x_{t+1} \sim P(X|x_t)$$
$$y_{t+1} \sim P(Y|x_{t+1})$$

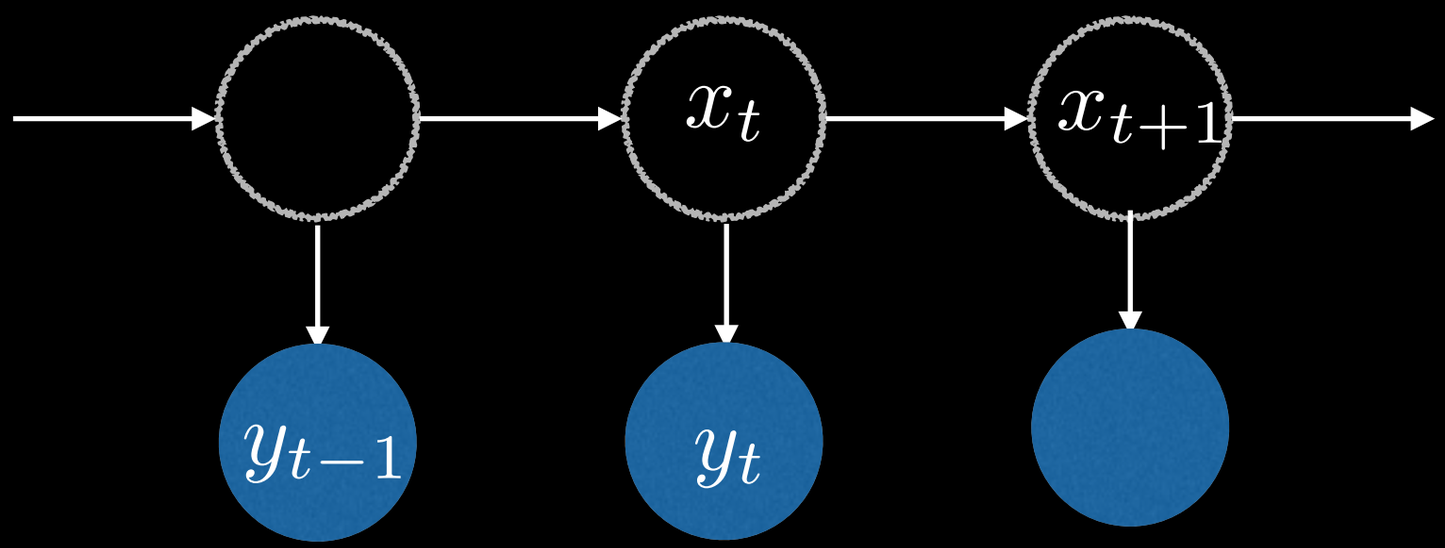


Latent State Space Model

e.g., Hidden Markov Model (HMM)
Linear Dynamical Systems (LDS)

$$x_{t+1} \sim P(X|x_t)$$
$$y_{t+1} \sim P(Y|x_{t+1})$$

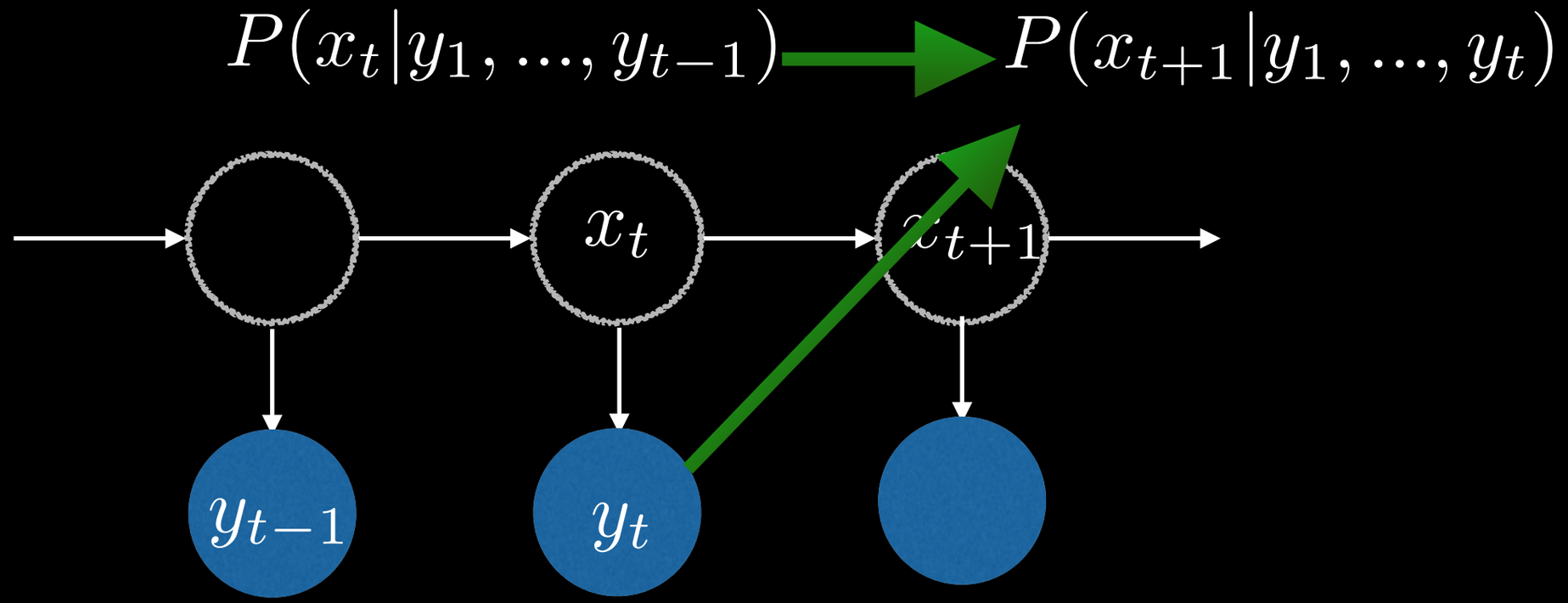
$$P(x_t|y_1, \ldots, y_{t-1})$$



Latent State Space Model

e.g., Hidden Markov Model (HMM)
Linear Dynamical Systems (LDS)

$$x_{t+1} \sim P(X|x_t)$$
$$y_{t+1} \sim P(Y|x_{t+1})$$



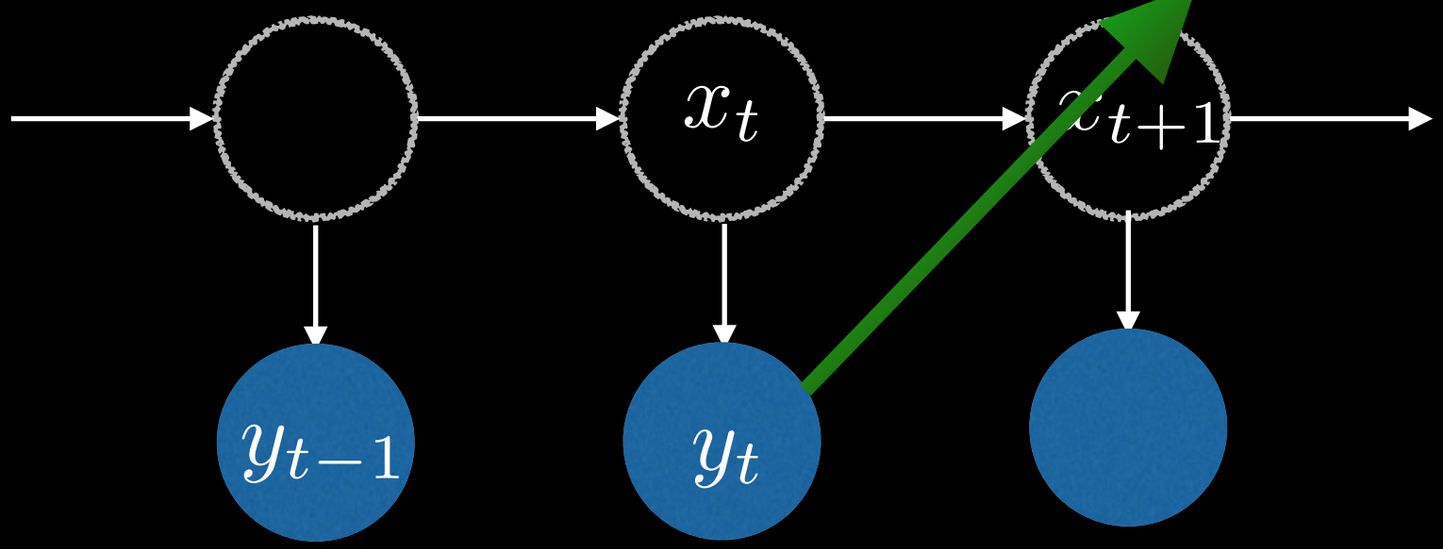
Latent State Space Model

e.g., Hidden Markov Model (HMM)
Linear Dynamical Systems (LDS)

$$x_{t+}$$  $$|x_t)$$

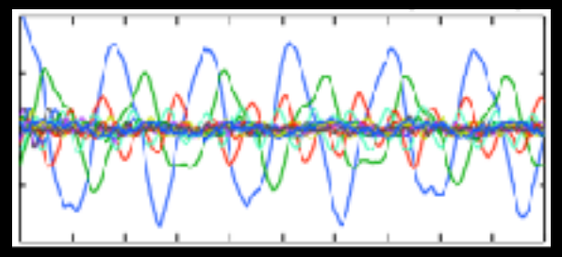
$$y_{t+1}$$ $$_{t+1})$$

$$P(x_t|y_1,\ldots,y_{t-1}) \rightarrow P(x_{t+1}|y_1,\ldots,y_t)$$



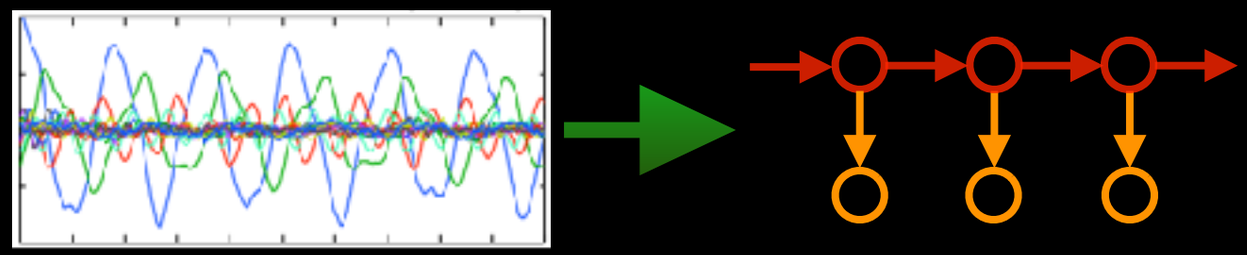
Classic Approach

Classic Approach



Sequences of observation

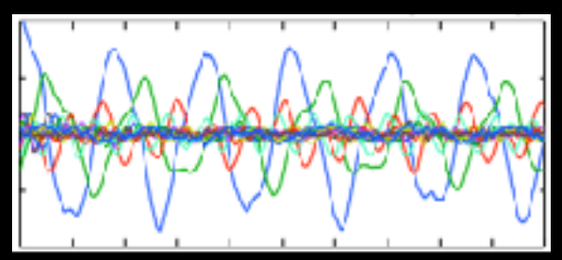
Classic Approach



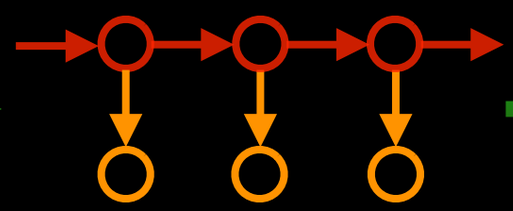
Sequences of observation

Hypothesize a model (e.g., LDS)

Classic Approach



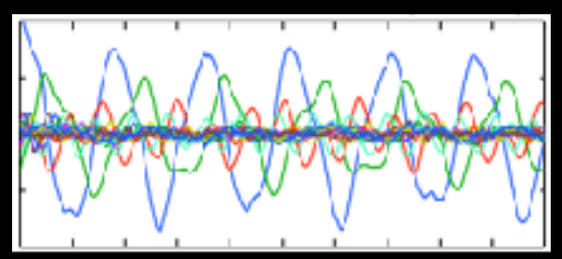
Sequences of observation



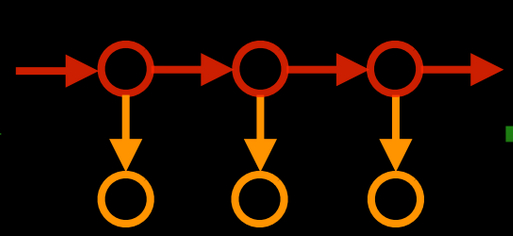
Hypothesize a model (e.g., LDS)

Learning (e.g., EM)

Classic Approach



Sequences of observation



Hypothesize a model (e.g., LDS)

Learning (e.g., EM)

$\hat{\theta}$

Learned parameters (e.g., MLE)

$$\|\hat{\theta} - \theta^*\| \neq 0$$

e.g., local optimal (EM),
finite data (Spectral methods)

$$\|\hat{\theta} - \theta^*\| \neq 0$$

e.g., local optimal (EM),
finite data (Spectral methods)

Model Mismatch

$$\|\hat{\theta} - \theta^*\| \neq 0$$

e.g., local optimal (EM),
finite data (Spectral methods)

Model Mismatch

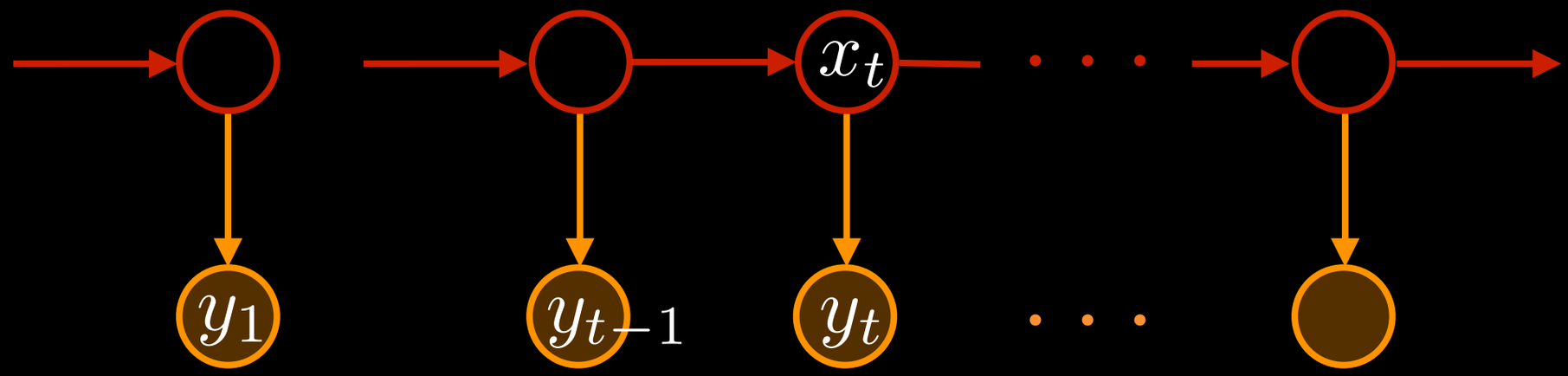
System identification generally does not guarantee filtering or prediction performance

Goal:

Directly learn the filter procedure to predict from history to future observations.

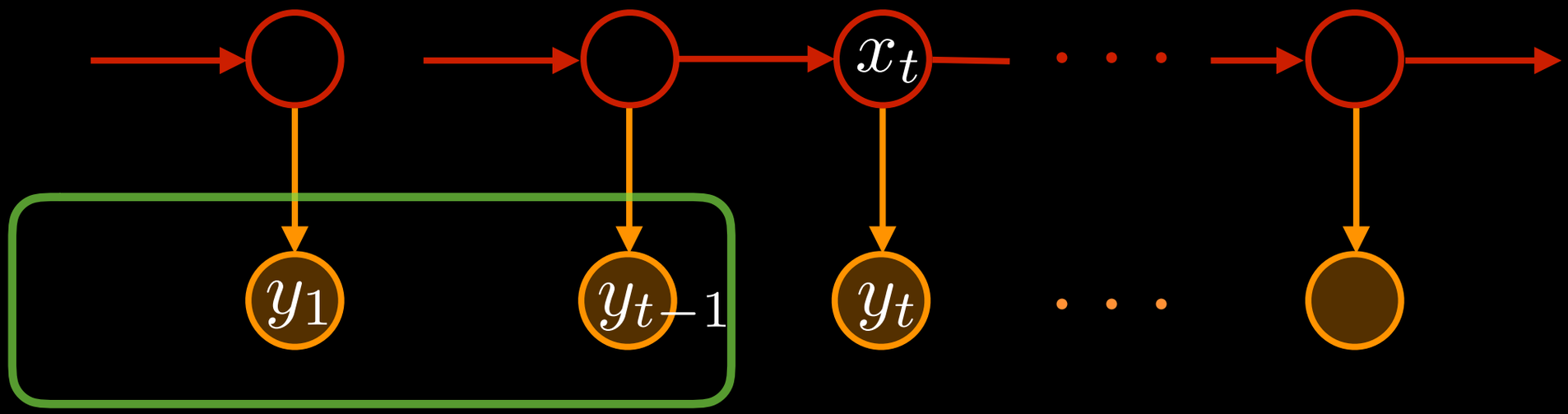
- Similar to Recurrent Neural Networks
- Performance guarantees on prediction errors
- Good practical performance

Predictive State Representation



[Singh et al., 2004, Hefty et al. 2015]

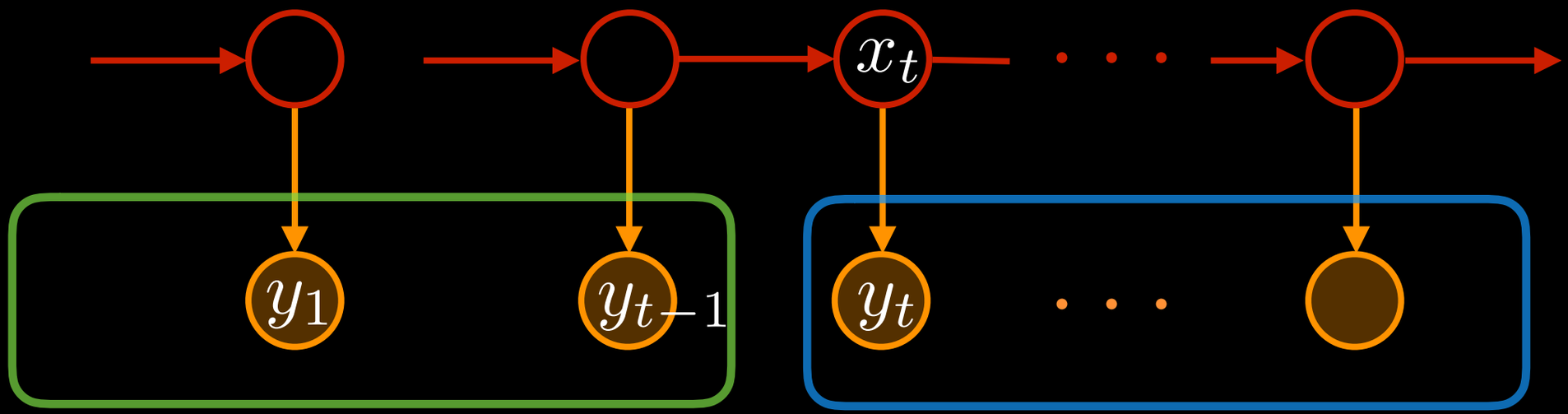
Predictive State Representation



$$h_{t-1} = [y_1, \ldots, y_{t-1}]$$

[Singh et al., 2004, Hefty et al. 2015]

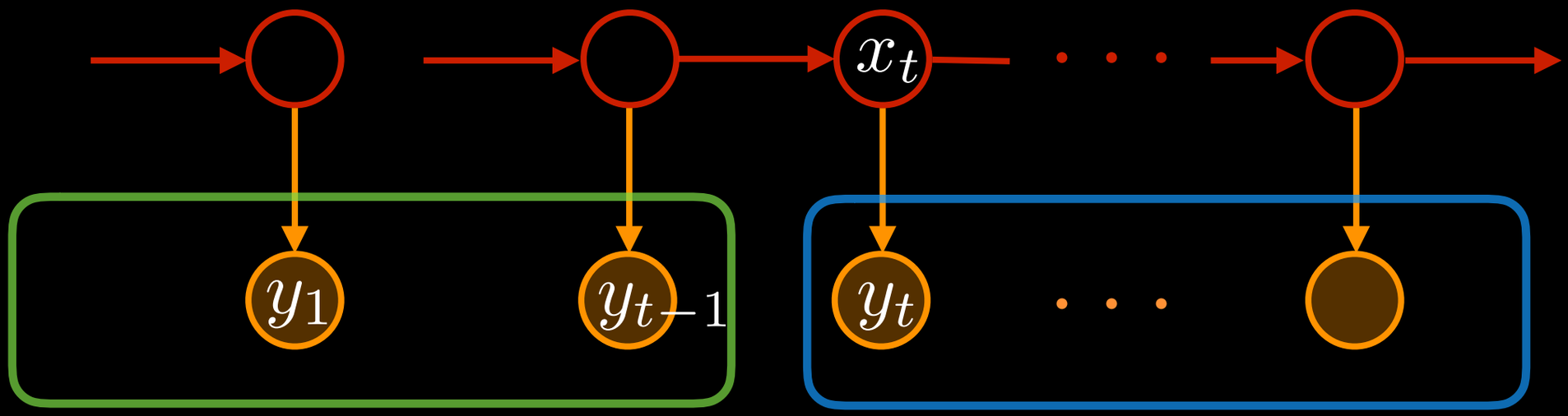
Predictive State Representation



$$h_{t-1} = [y_1, \ldots, y_{t-1}] \qquad f_t = [y_t, y_{t+1}, \ldots, y_{t+k-1}]$$

[Singh et al., 2004, Hefty et al. 2015]

Predictive State Representation



$h_{t-1} = [y_1, \ldots, y_{t-1}]$

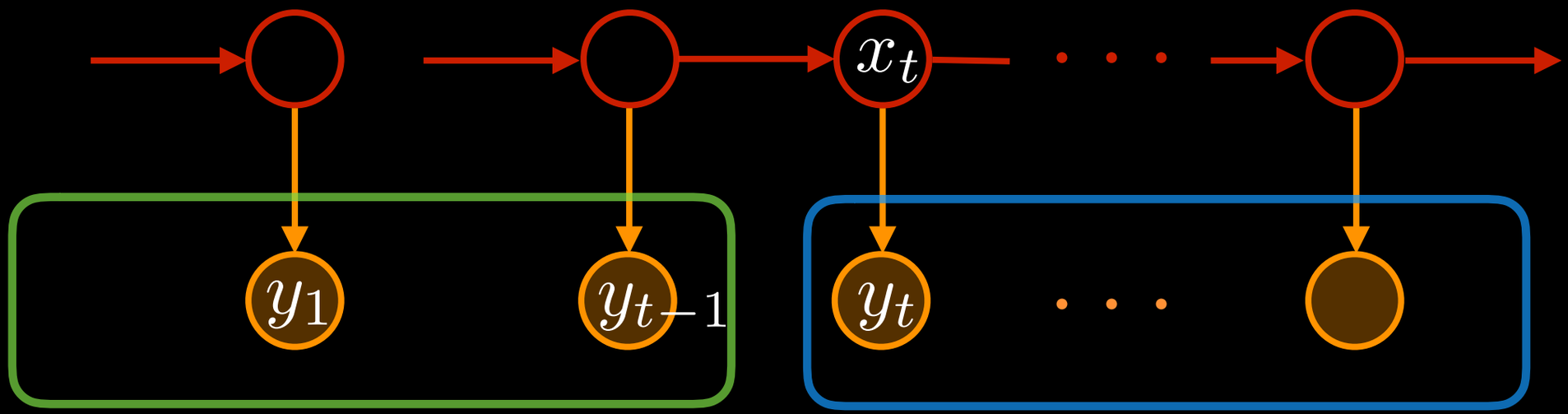
$f_t = [y_t, y_{t+1}, \ldots, y_{t+k-1}]$

$P(x_t|h_{t-1}) \Leftrightarrow P(f_t|h_{t-1})$ K-observability

[Kalman 1963, Hsu et al. 2012]

[Singh et al., 2004, Hefty et al. 2015]

Predictive State Representation



$$h_{t-1} = [y_1, \ldots, y_{t-1}] \qquad f_t = [y_t, y_{t+1}, \ldots, y_{t+k-1}]$$

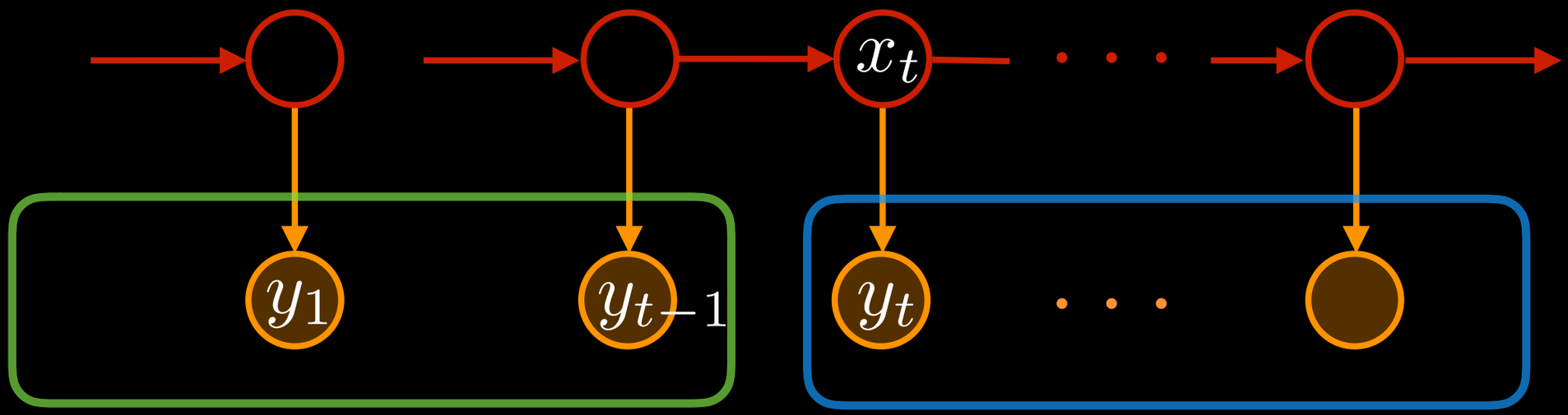
$$P(x_t|h_{t-1}) \Leftrightarrow P(f_t|h_{t-1})$$ K-observability

[Kalman 1963, Hsu et al. 2012]

$$P(f_t|h_{t-1}) \Leftrightarrow \mathbb{E}[\phi(f_t)|h_{t-1}]$$

[Singh et al., 2004, Hefty et al. 2015]

Predictive State Representation



$$h_{t-1} = [y_1, \ldots, y_{t-1}] \qquad f_t = [y_t, y_{t+1}, \ldots, y_{t+k-1}]$$

$$P(x_t|h_{t-1}) \Leftrightarrow P(f_t|h_{t-1})$$ K-observability

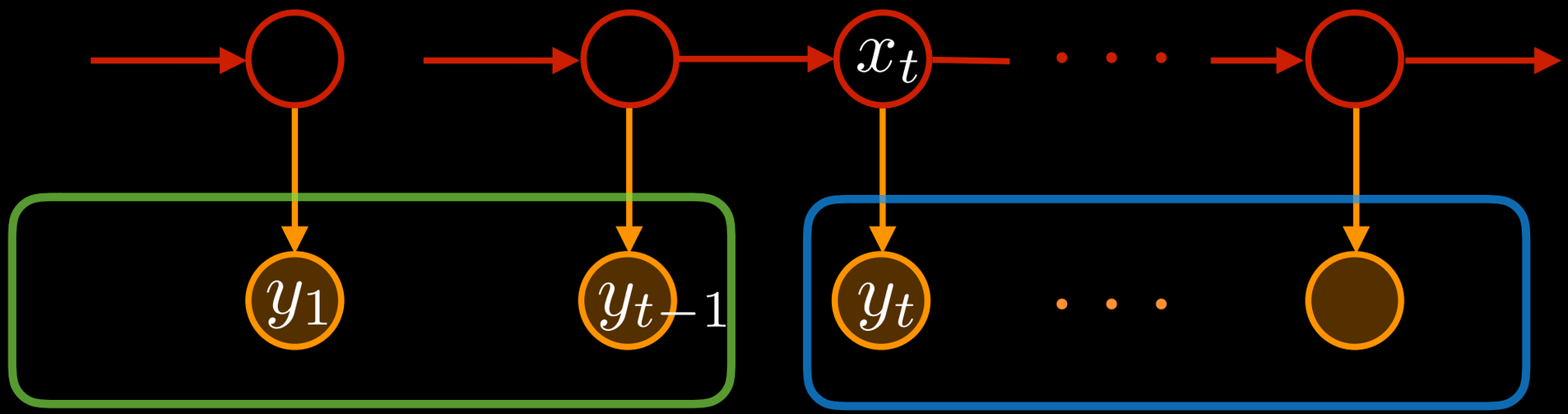
[Kalman 1963, Hsu et al. 2012]

$$P(f_t|h_{t-1}) \Leftrightarrow \mathbb{E}[\phi(f_t)|h_{t-1}]$$

Sufficient Features

[Singh et al., 2004, Hefty et al. 2015]

Predictive State Representation



$$h_{t-1} = [y_1, \ldots, y_{t-1}] \qquad f_t = [y_t, y_{t+1}, \ldots, y_{t+k-1}]$$

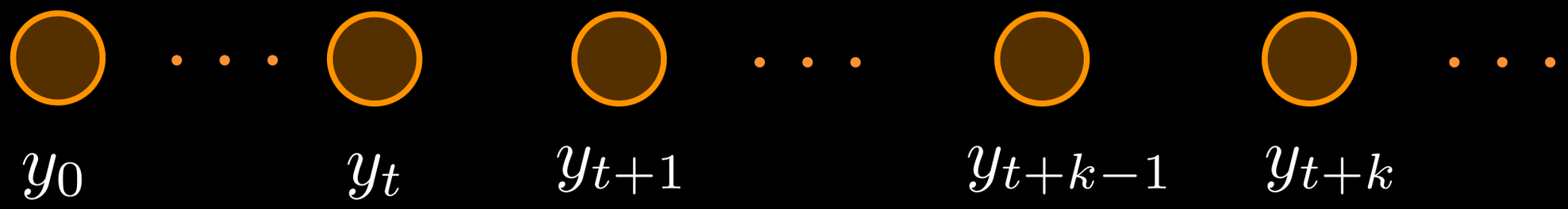
$$P(x_t|h_{t-1}) \Leftrightarrow P(f_t|h_{t-1})$$ K-observability

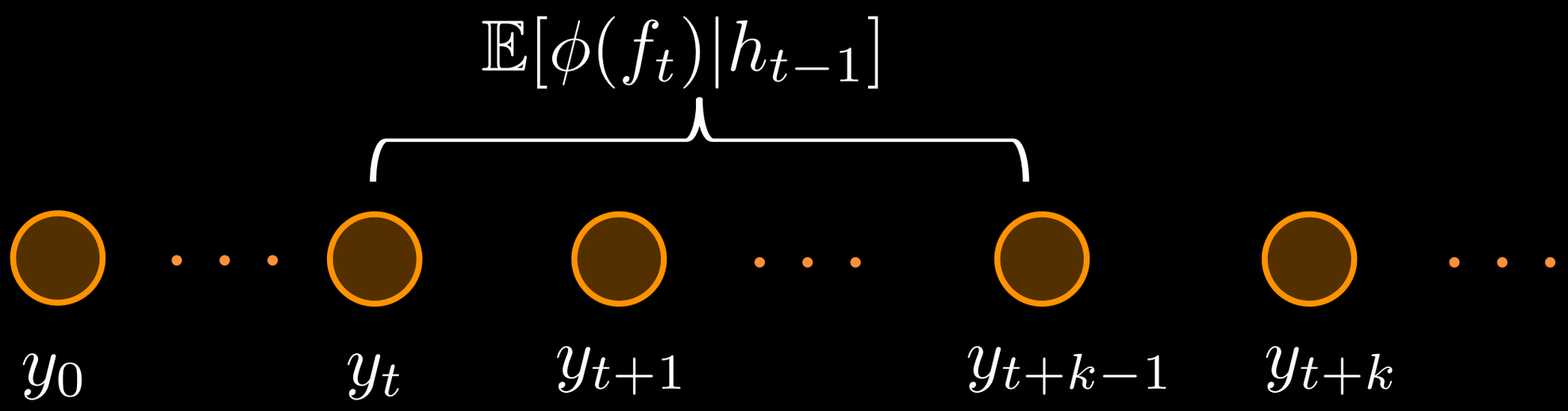
[Kalman 1963, Hsu et al. 2012]

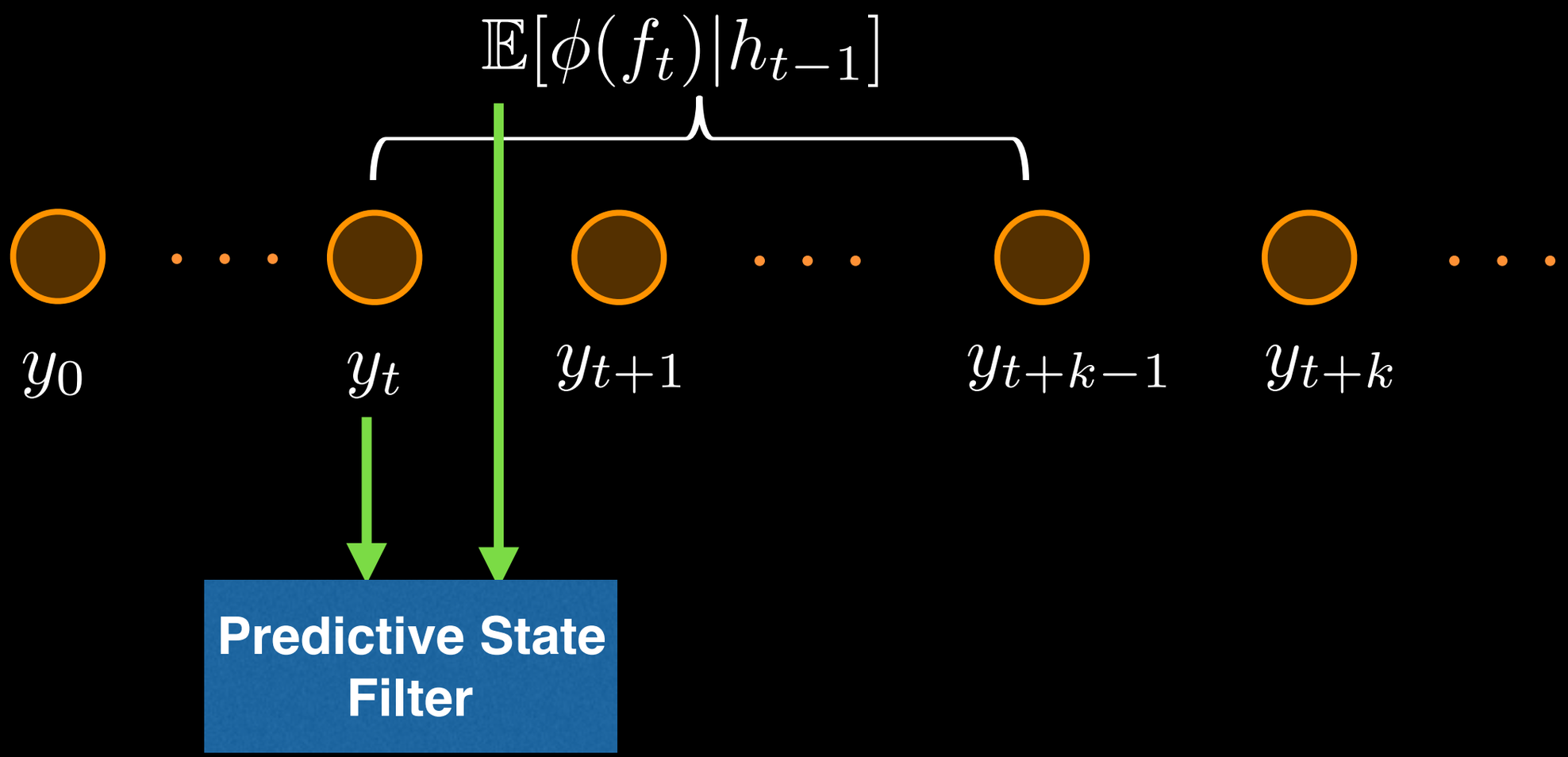
$$P(f_t|h_{t-1}) \Leftrightarrow \mathbb{E}[\phi(f_t)|h_{t-1}]$$

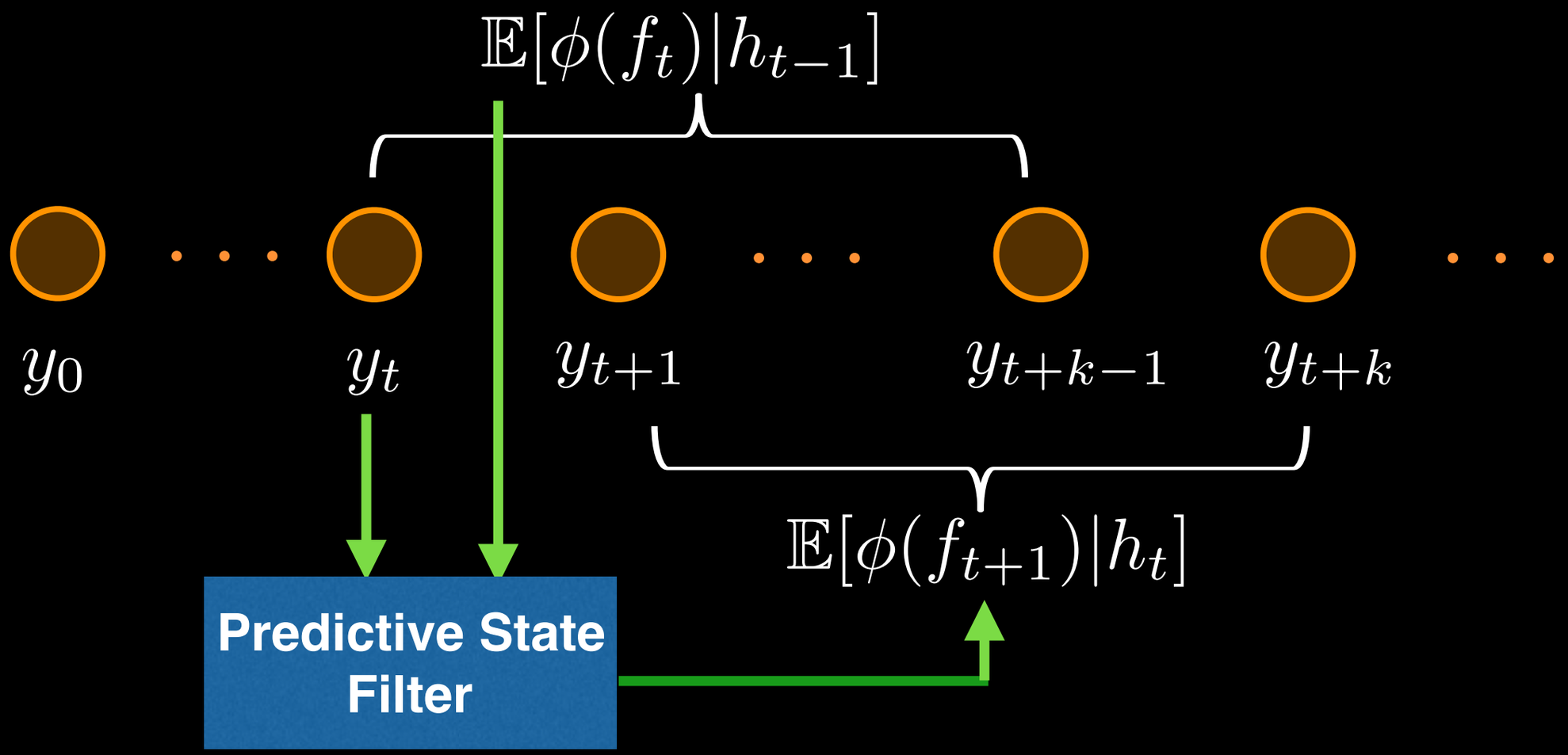
Sufficient Features

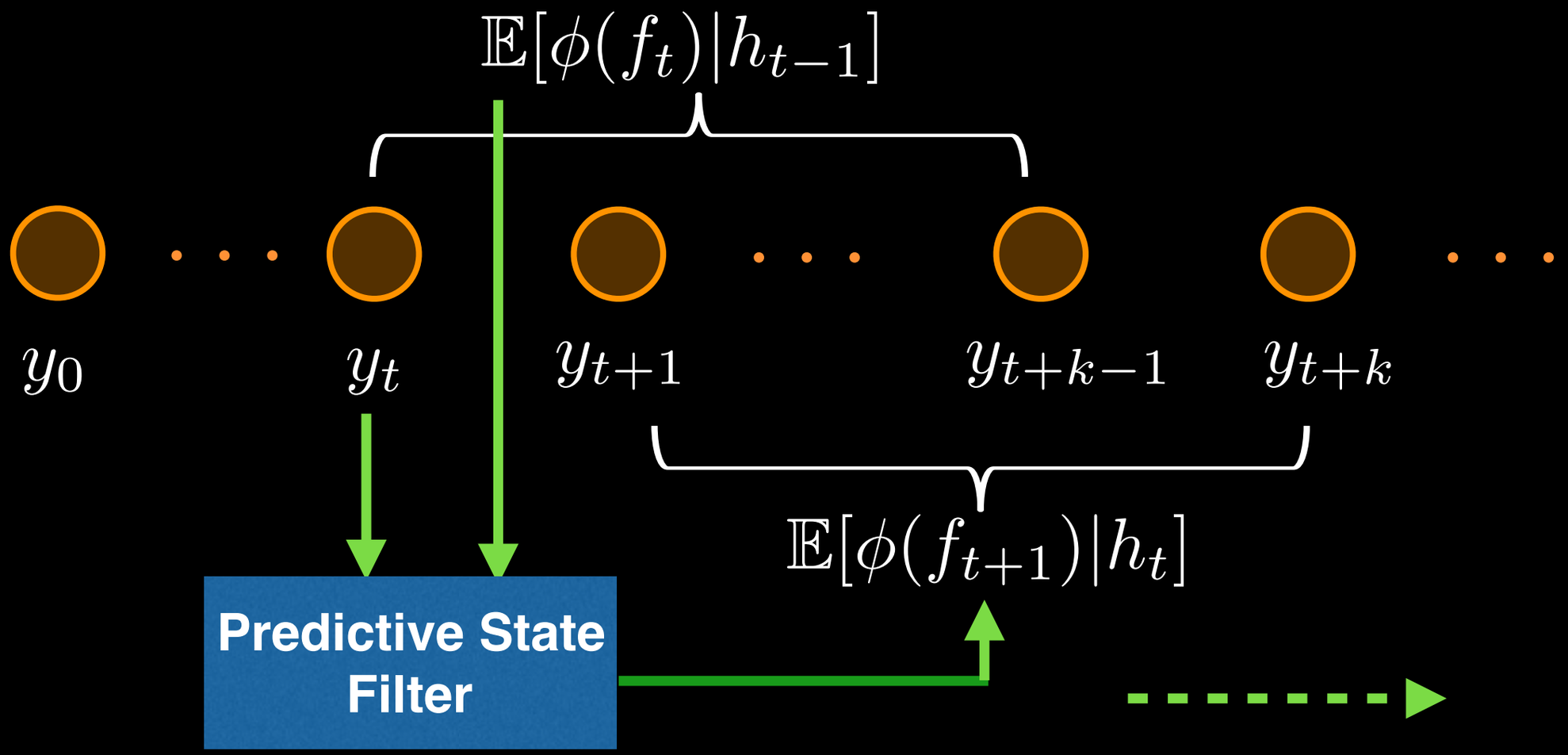
[Singh et al., 2004, Hefty et al. 2015]

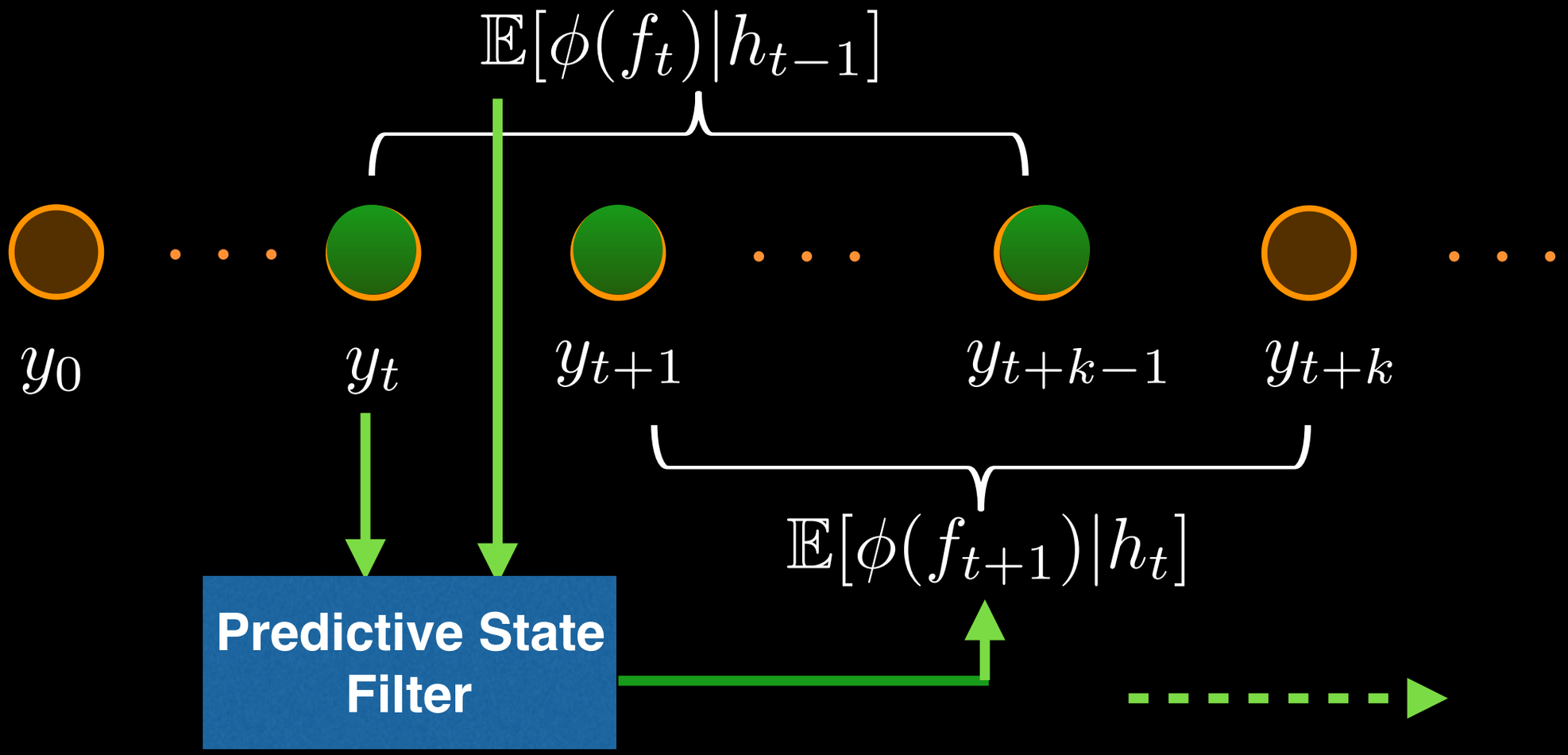

y_0
y_t
y_{t+1}
y_{t+k-1}
y_{t+k}

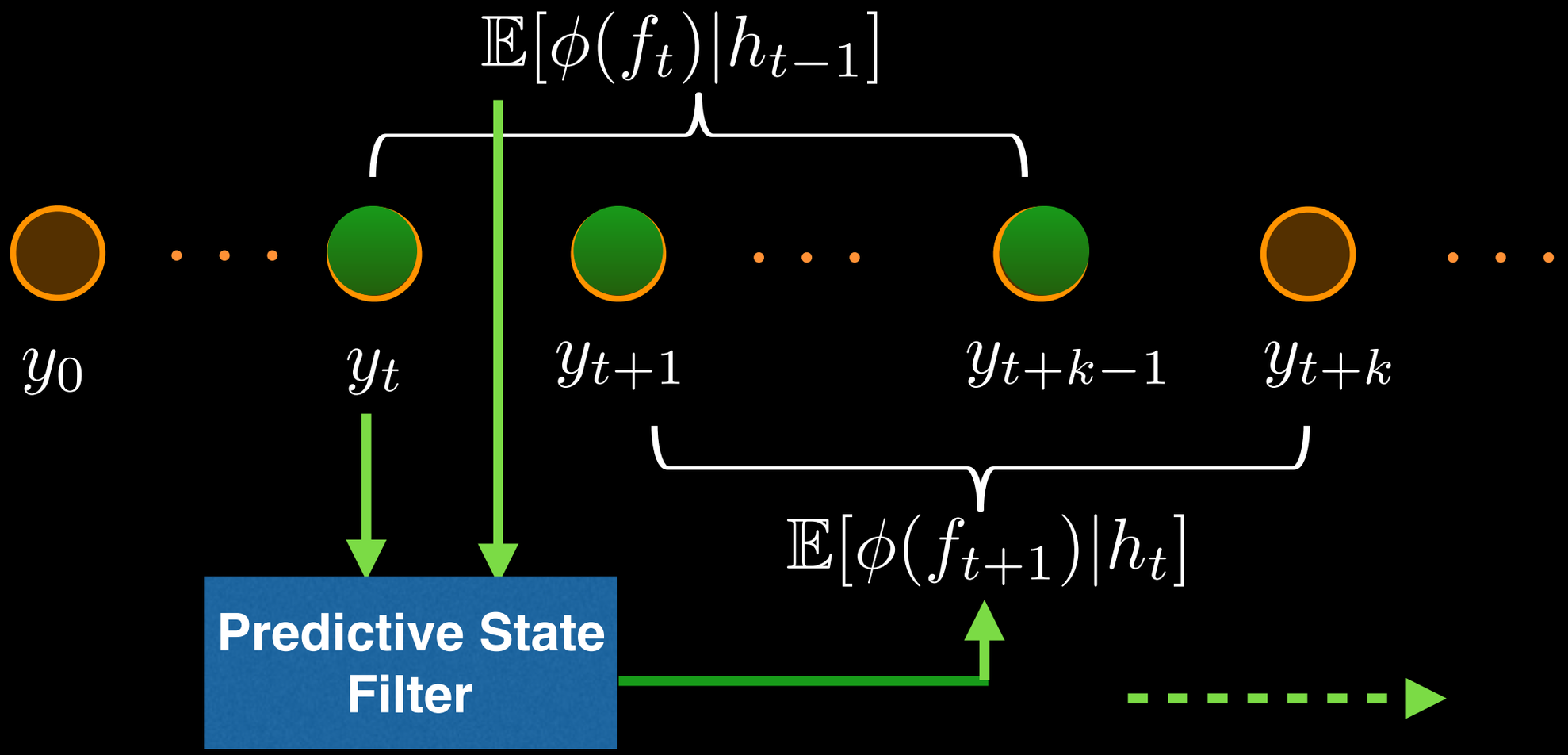

$\mathbb{E}[\phi(f_t)|h_{t-1}]$
y_0
y_t
y_{t+1}
y_{t+k-1}
y_{t+k}


$\mathbb{E}[\phi(f_t)|h_{t-1}]$
y_0
y_t
y_{t+1}
y_{t+k-1}
y_{t+k}
Predictive State Filter

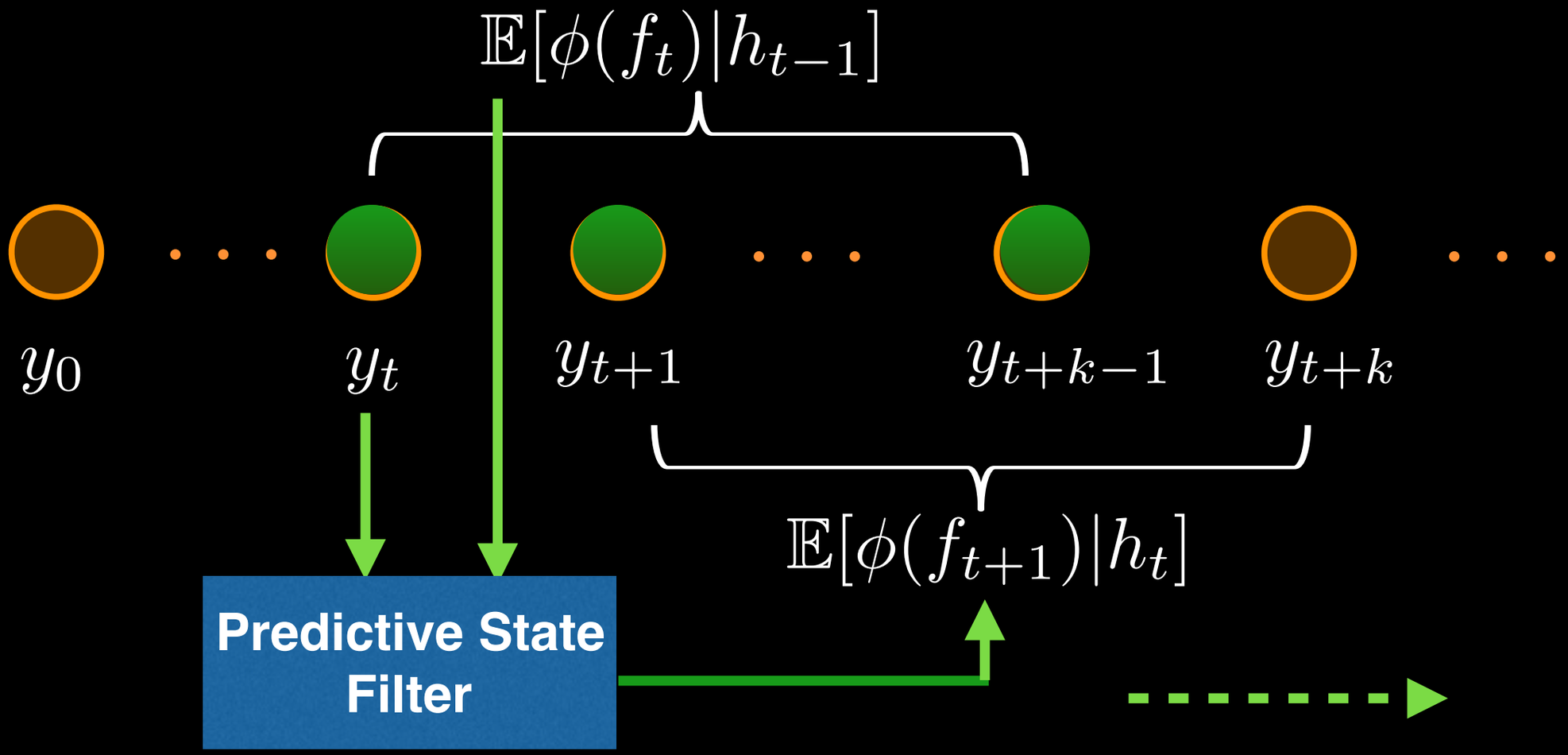

$\mathbb{E}[\phi(f_t)|h_{t-1}]$
y_0
y_t
y_{t+1}
y_{t+k-1}
y_{t+k}
$\mathbb{E}[\phi(f_{t+1})|h_t]$
Predictive State Filter


$\mathbb{E}[\phi(f_t)|h_{t-1}]$
y_0
y_t
y_{t+1}
y_{t+k-1}
y_{t+k}
$\mathbb{E}[\phi(f_{t+1})|h_t]$
Predictive State Filter


$\mathbb{E}[\phi(f_t)|h_{t-1}]$
y_0
y_t
y_{t+1}
y_{t+k-1}
y_{t+k}
$\mathbb{E}[\phi(f_{t+1})|h_t]$
Predictive State Filter

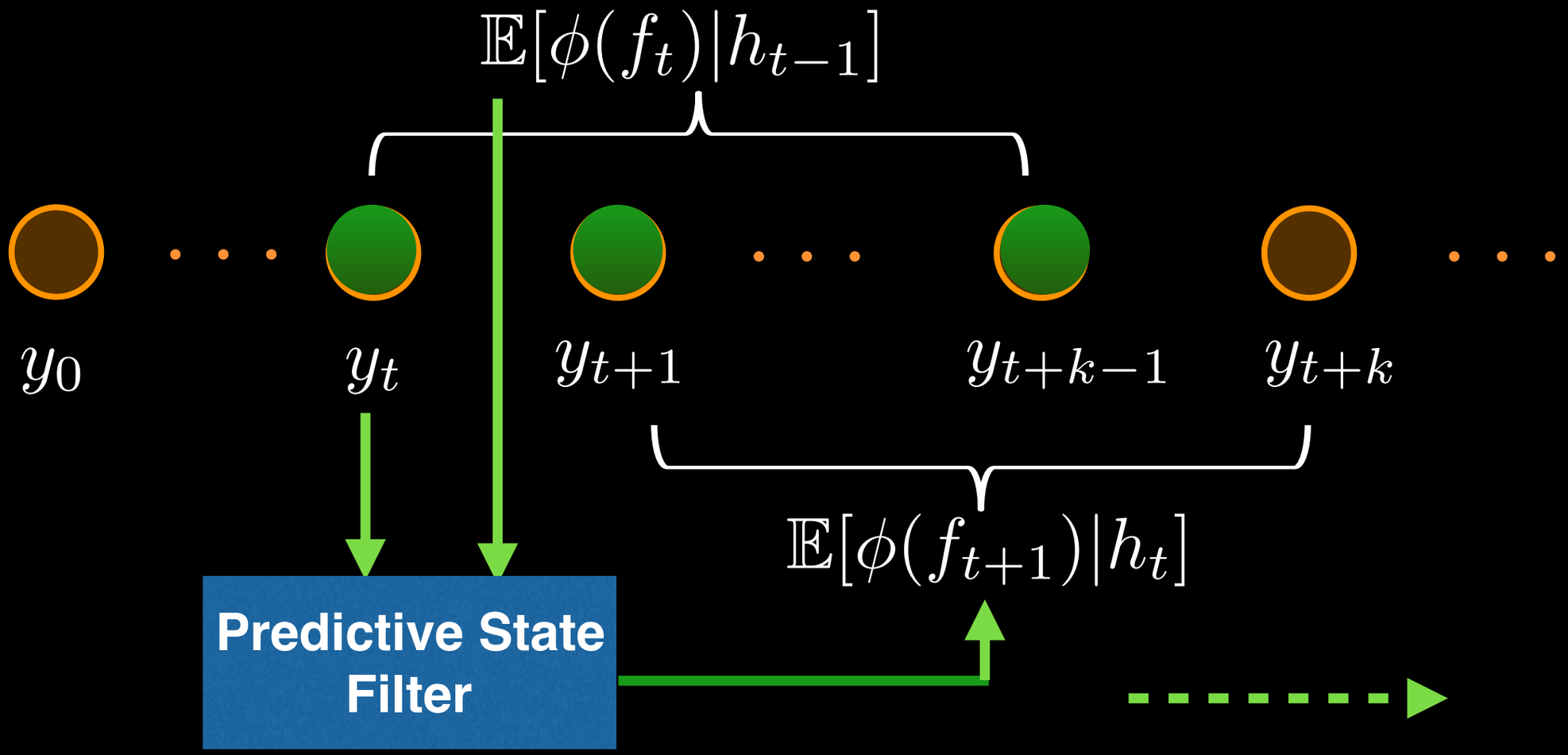


$$\|\mathbb{E}[\phi(f_t)|h_{t-1}] - \phi(\tilde{f}_t)\|^2$$



$$\|\mathbb{E}[\phi(f_t)|h_{t-1}] - \phi(\tilde{f}_t)\|^2$$

Moment Matching



$$\sum_{t=1}^{T} \|\mathbb{E}[\phi(f_t)|h_{t-1}] - \phi(\tilde{f}_t)\|^2$$

Moment Matching

$\mathbb{E}[\phi(f_t)|h_{t-1}]$

y_0 y_t y_{t+1} y_{t+k-1} y_{t+k}

$\mathbb{E}[\phi(f_{t+1})|h_t]$

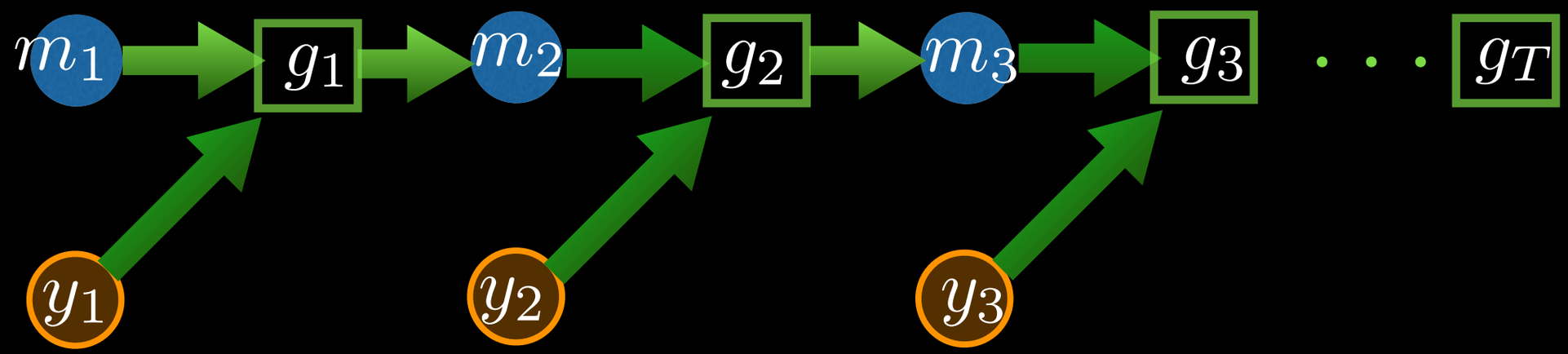
Predictive State Filter

$$\sum_{t=1}^{T} \|\mathbb{E}[\phi(f_t)|h_{t-1}] - \phi(\tilde{f}_t)\|^2$$

Moment Matching

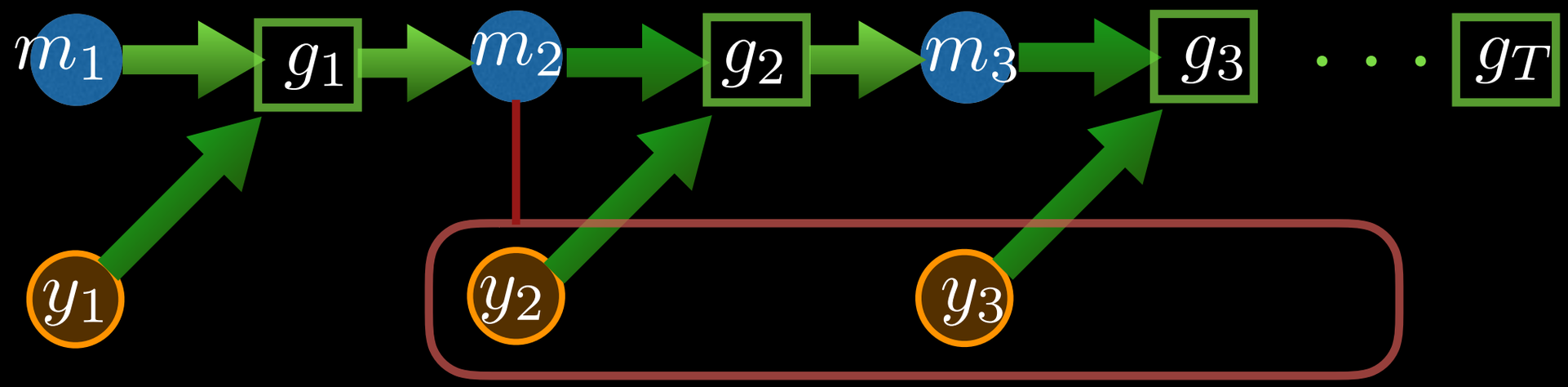
Learning Non-stationary Filters

$$m_t = \mathbb{E}[\phi(f_t)|h_{t-1}]$$



Learning Non-stationary Filters

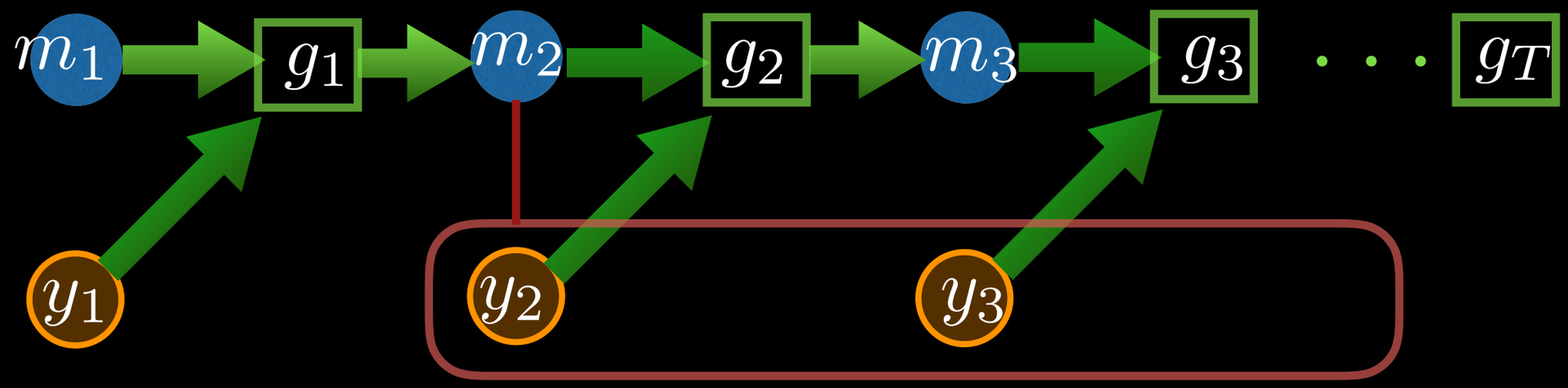
$$m_t = \mathbb{E}[\phi(f_t)|h_{t-1}]$$



Supervision f_2 for m_2

Learning Non-stationary Filters

$$m_t = \mathbb{E}[\phi(f_t)|h_{t-1}]$$



Supervision f_2 for m_2

$$\min_{g_1, g_2, \ldots, g_T} \mathbb{E}_\tau \sum_{t=1}^{T} \|g_t(m_t, y_t) - \phi(f_{t+1})\|^2$$

$$s.t. \;\; m_{t+1} = g_t(m_t, y_t), \forall t$$

Learning Non-stationary Filters

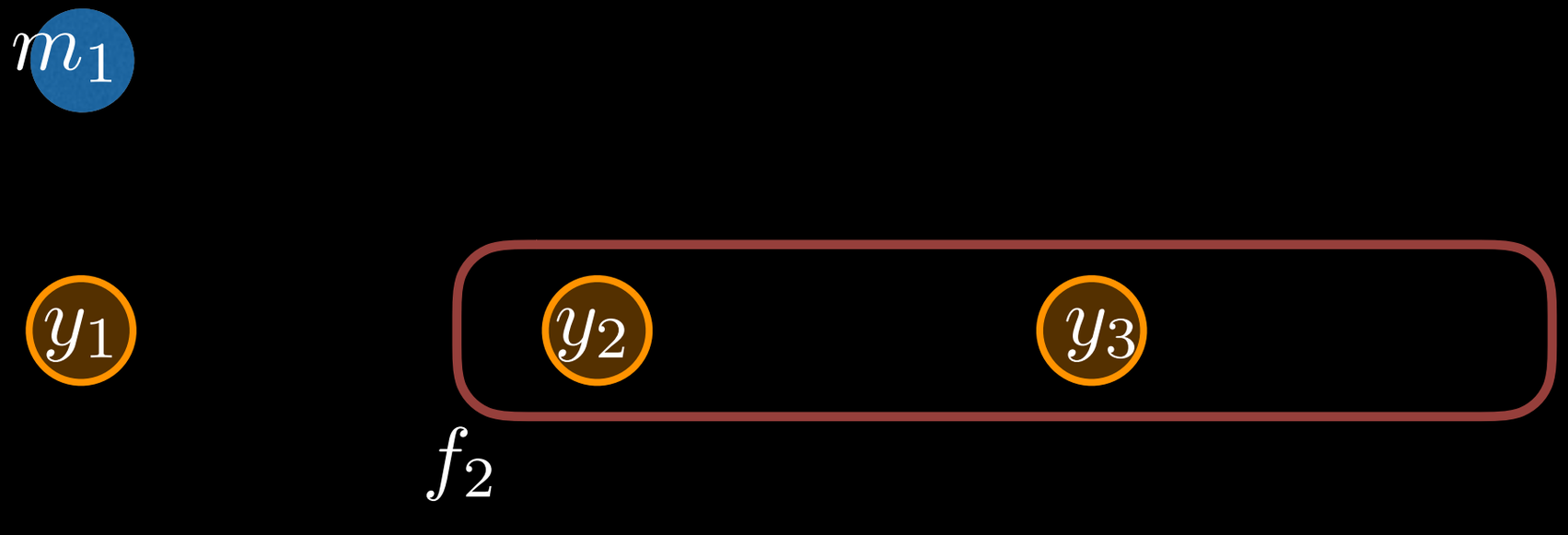
Forward Training [Ross & Bagnell 2010]

m_1

y_1 y_2 y_3

Learning Non-stationary Filters

Forward Training [Ross & Bagnell 2010]



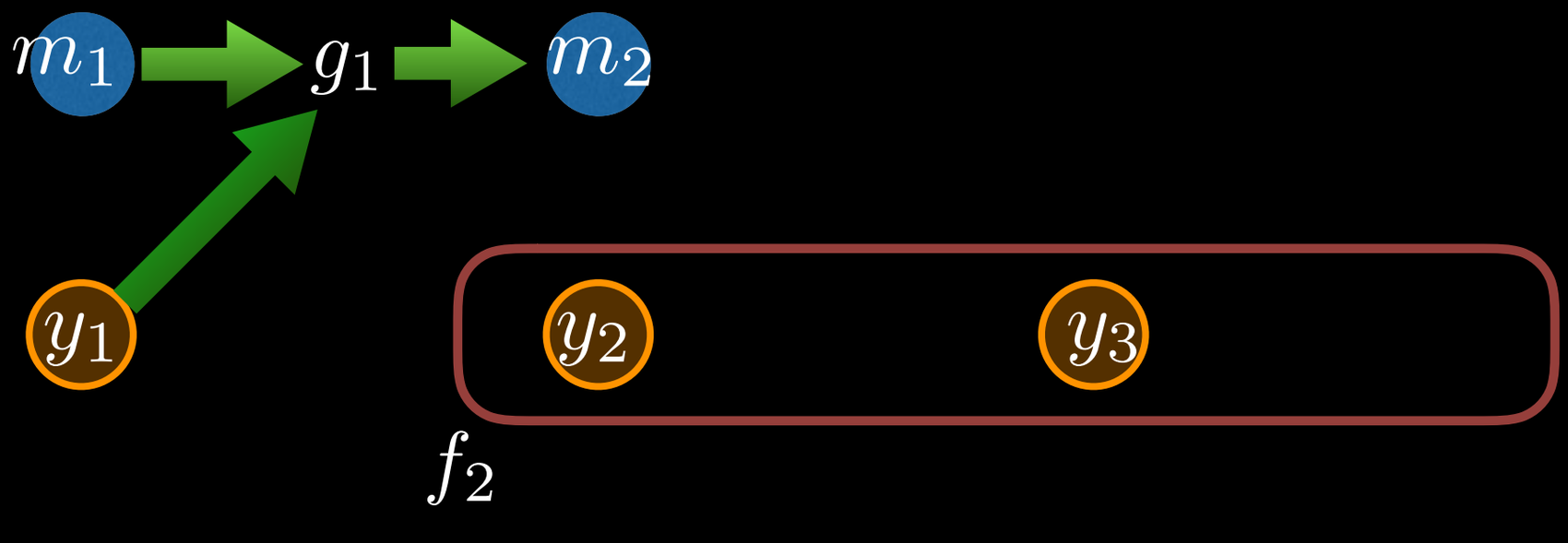
Learning Non-stationary Filters

Forward Training [Ross & Bagnell 2010]

$$g_1 = \arg\min_{g \in \mathcal{G}} \sum_{\tau} \|g(m_1, y_1) - \phi(f_2)\|^2$$

Learning Non-stationary Filters

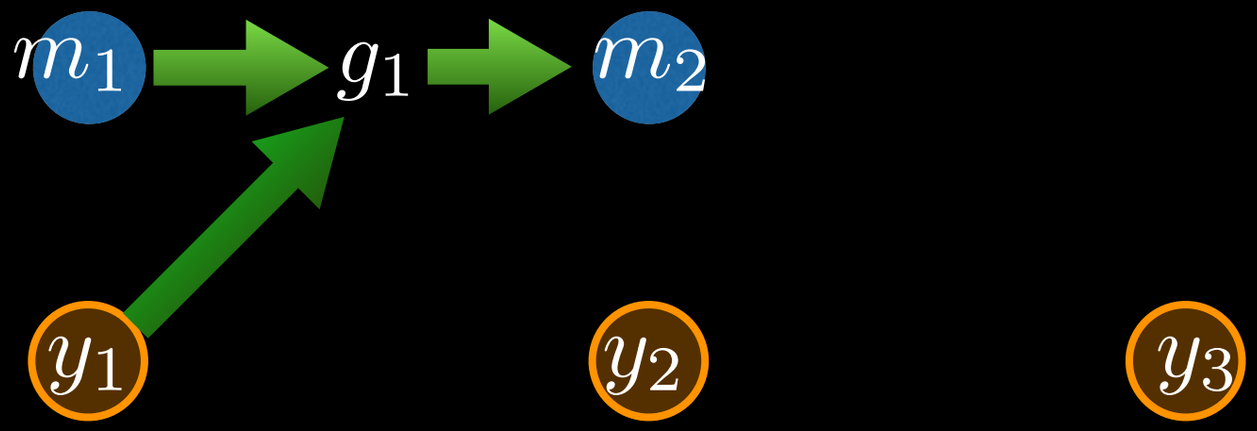
Forward Training [Ross & Bagnell 2010]



$$g_1 = \arg\min_{g \in \mathcal{G}} \sum_{\tau} \|g(m_1, y_1) - \phi(f_2)\|^2$$

Learning Non-stationary Filters

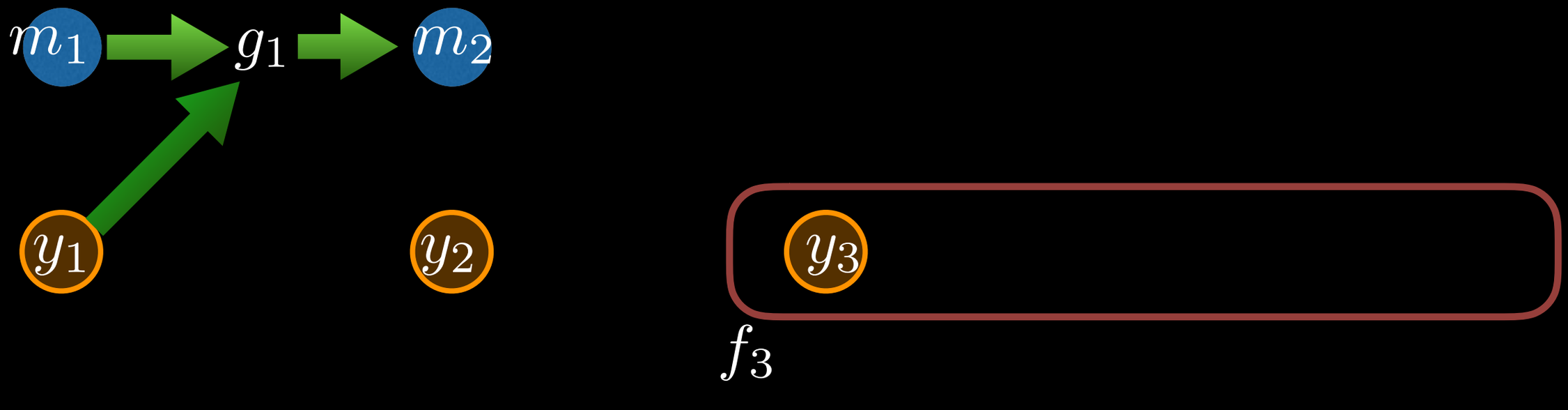
Forward Training [Ross & Bagnell 2010]



$$g_1 = \arg\min_{g \in \mathcal{G}} \sum_{\tau} \|g(m_1, y_1) - \phi(f_2)\|^2$$

Learning Non-stationary Filters

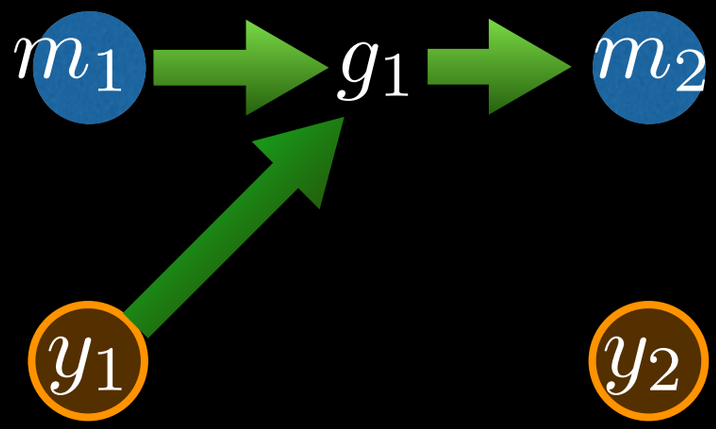
Forward Training [Ross & Bagnell 2010]



$$g_1 = \arg\min_{g \in \mathcal{G}} \sum_{\tau} \|g(m_1, y_1) - \phi(f_2)\|^2$$

Learning Non-stationary Filters

Forward Training [Ross & Bagnell 2010]

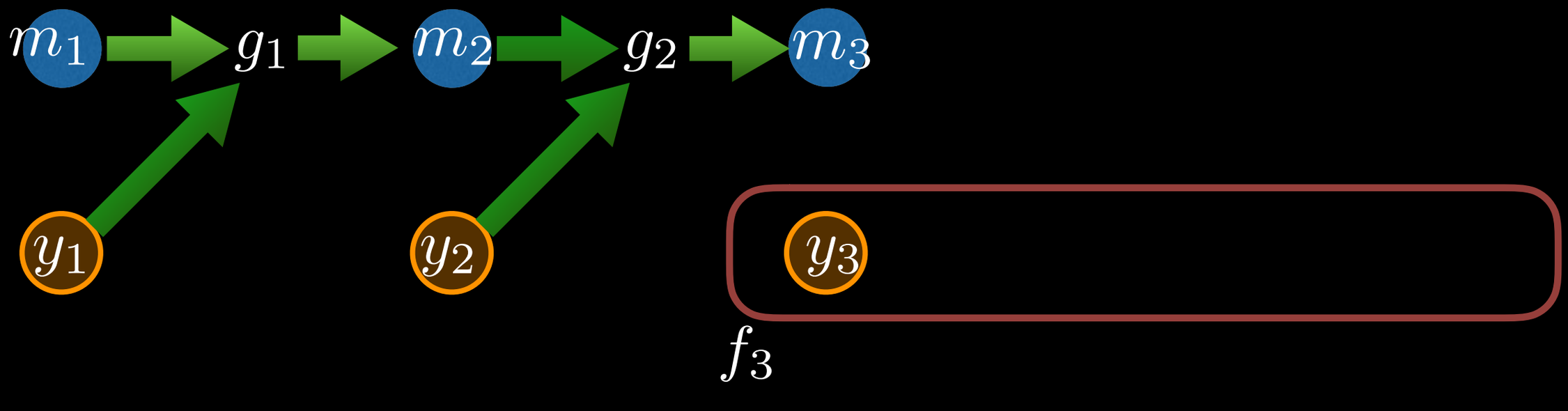


$$g_1 = \arg\min_{g \in \mathcal{G}} \sum_{\tau} \|g(m_1, y_1) - \phi(f_2)\|^2$$

$$g_2 = \arg\min_{g \in \mathcal{G}} \sum_{\tau} \|g(m_2, y_2) - \phi(f_3)\|^2$$

Learning Non-stationary Filters

Forward Training [Ross & Bagnell 2010]

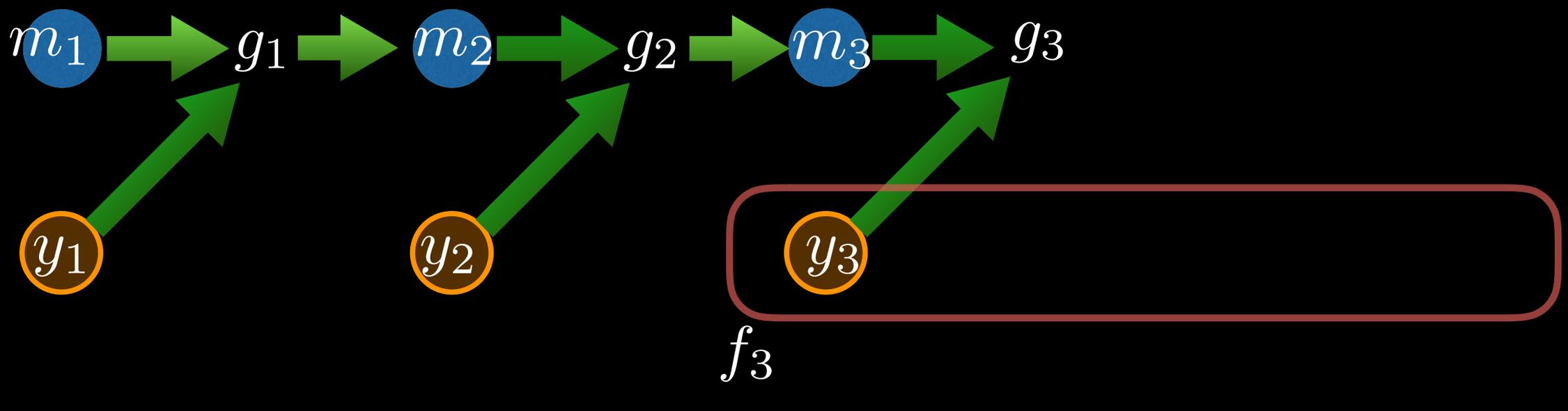


$$g_1 = \arg\min_{g \in \mathcal{G}} \sum_{\tau} \|g(m_1, y_1) - \phi(f_2)\|^2$$

$$g_2 = \arg\min_{g \in \mathcal{G}} \sum_{\tau} \|g(m_2, y_2) - \phi(f_3)\|^2$$

Learning Non-stationary Filters

Forward Training [Ross & Bagnell 2010]



$$g_1 = \arg\min_{g \in \mathcal{G}} \sum_{\tau} \|g(m_1, y_1) - \phi(f_2)\|^2$$

$$g_2 = \arg\min_{g \in \mathcal{G}} \sum_{\tau} \|g(m_2, y_2) - \phi(f_3)\|^2$$

Learning Non-stationary Filters

Performance Guarantee

Learning Non-stationary Filters

Performance Guarantee

$$\mathbb{E}_{\tau\sim\mathcal{D}_\tau}\Big[\frac{1}{T}\sum_{t=1}^{T}\|g_t(\hat{m}_t^\tau, x_t^\tau) - \phi(f_{t+1}^\tau)\|^2\Big]$$

$$\leq \mathbb{E}_{\tau\sim\mathcal{D}_\tau}\big[\frac{1}{T}\sum_{t=1}^{T}\|g_t^*(\hat{m}_t^\tau, x_t^\tau) - \phi(f_{t+1}^\tau)\|^2\big]$$

$$+ 4\nu\bar{\mathcal{R}}(\mathcal{G})$$

$$+ 2\sqrt{\frac{T\ln(T/\delta)}{2M}}$$

Learning Non-stationary Filters

Performance Guarantee

$$\mathbb{E}_{\tau\sim\mathcal{D}_\tau}\left[\frac{1}{T}\sum_{t=1}^{T}\|g_t(\hat{m}_t^\tau, x_t^\tau) - \phi(f_{t+1}^\tau)\|^2\right]$$

Prediction error

$$\leq \mathbb{E}_{\tau\sim\mathcal{D}_\tau}\left[\frac{1}{T}\sum_{t=1}^{T}\|g_t^*(\hat{m}_t^\tau, x_t^\tau) - \phi(f_{t+1}^\tau)\|^2\right]$$

$$+ 4\nu\bar{\mathcal{R}}(\mathcal{G})$$

$$+ 2\sqrt{\frac{T\ln(T/\delta)}{2M}}$$

Learning Non-stationary Filters

Performance Guarantee

$$\mathbb{E}_{\tau\sim\mathcal{D}_\tau}\left[\frac{1}{T}\sum_{t=1}^{T}\|g_t(\hat{m}_t^\tau, x_t^\tau) - \phi(f_{t+1}^\tau)\|^2\right]$$ → Prediction error

$$\leq \mathbb{E}_{\tau\sim\mathcal{D}_\tau}\left[\frac{1}{T}\sum_{t=1}^{T}\|g_t^*(\hat{m}_t^\tau, x_t^\tau) - \phi(f_{t+1}^\tau)\|^2\right]$$ → Prediction error from the best Hypothesis

$$+ 4\nu\bar{\mathcal{R}}(\mathcal{G})$$

$$+ 2\sqrt{\frac{T\ln(T/\delta)}{2M}}$$

Learning Non-stationary Filters

Performance Guarantee

$$\mathbb{E}_{\tau\sim\mathcal{D}_\tau}\left[\frac{1}{T}\sum_{t=1}^{T}\|g_t(\hat{m}_t^\tau, x_t^\tau) - \phi(f_{t+1}^\tau)\|^2\right]$$ → Prediction error

$$\leq \mathbb{E}_{\tau\sim\mathcal{D}_\tau}\left[\frac{1}{T}\sum_{t=1}^{T}\|g_t^*(\hat{m}_t^\tau, x_t^\tau) - \phi(f_{t+1}^\tau)\|^2\right]$$ → Prediction error from the best Hypothesis

$$+ 4\nu\bar{\mathcal{R}}(\mathcal{G})$$ → Rademacher number

$$+ 2\sqrt{\frac{T\ln(T/\delta)}{2M}}$$

Learning Non-stationary Filters

Performance Guarantee

$$\mathbb{E}_{\tau\sim\mathcal{D}_\tau}\left[\frac{1}{T}\sum_{t=1}^{T}\|g_t(\hat{m}_t^\tau, x_t^\tau) - \phi(f_{t+1}^\tau)\|^2\right]$$ → Prediction error

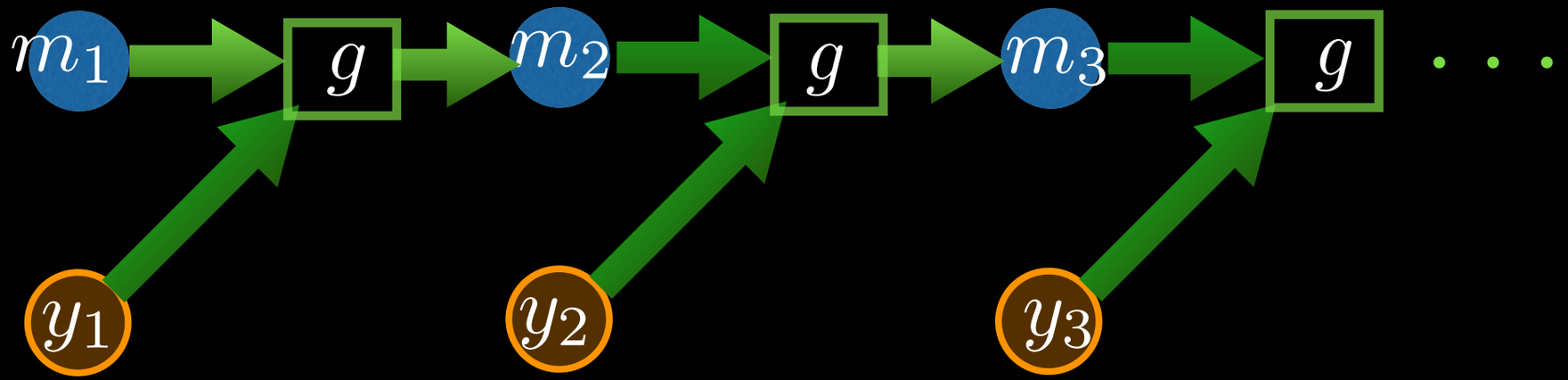
$$\leq \mathbb{E}_{\tau\sim\mathcal{D}_\tau}\left[\frac{1}{T}\sum_{t=1}^{T}\|g_t^*(\hat{m}_t^\tau, x_t^\tau) - \phi(f_{t+1}^\tau)\|^2\right]$$ → Prediction error from the best Hypothesis

$$+ 4\nu\bar{\mathcal{R}}(\mathcal{G})$$ → Rademacher number

$$+ 2\sqrt{\frac{T\ln(T/\delta)}{2M}}$$ → Excess Error

Learning Stationary Filters

$$m_t = \mathbb{E}[\phi(f_t)|h_{t-1}]$$

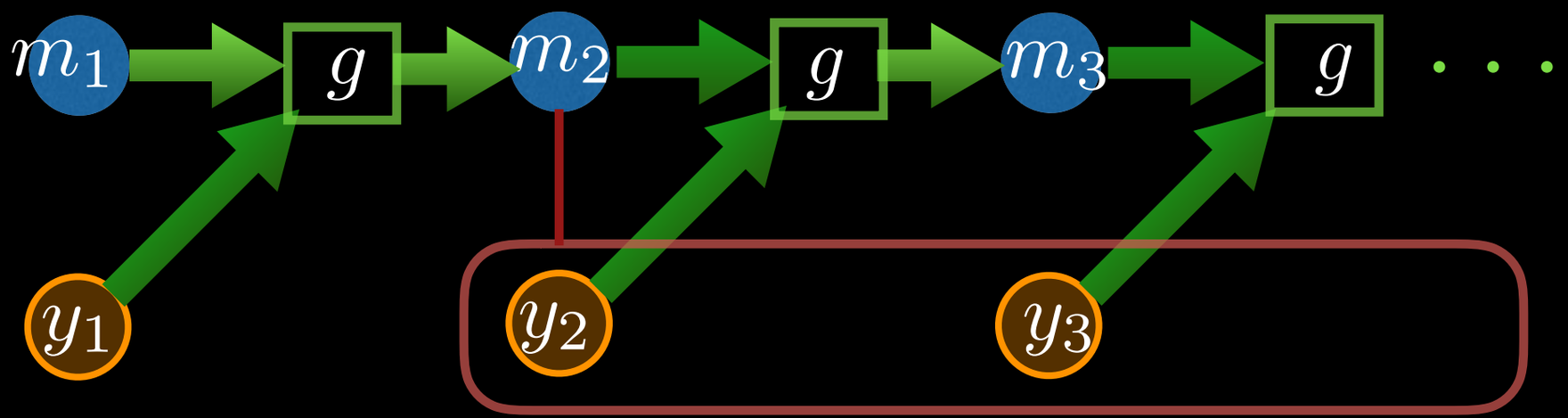


$$\min_g \mathbb{E}_\tau \sum_{t=1}^{T} \|g(m_t, y_t) - \phi(f_{t+1})\|^2$$

$$s.t. \;\; m_{t+1} = g(m_t, y_t), \forall t$$

Learning Stationary Filters

$$m_t = \mathbb{E}[\phi(f_t)|h_{t-1}]$$



Supervision f_2 for m_2

$$\min_g \mathbb{E}_\tau \sum_{t=1}^{T} \|g(m_t, y_t) - \phi(f_{t+1})\|^2$$

$$s.t. \;\; m_{t+1} = g(m_t, y_t), \forall t$$

Learning Stationary Filter

Data Aggregation (DAgger) [Ross & Bagnell 2011]

Follow the (regularized) Leader,
Online Gradient Descent [Zinkevich,2003]

No-Regret
Online Learner

Learning Stationary Filter

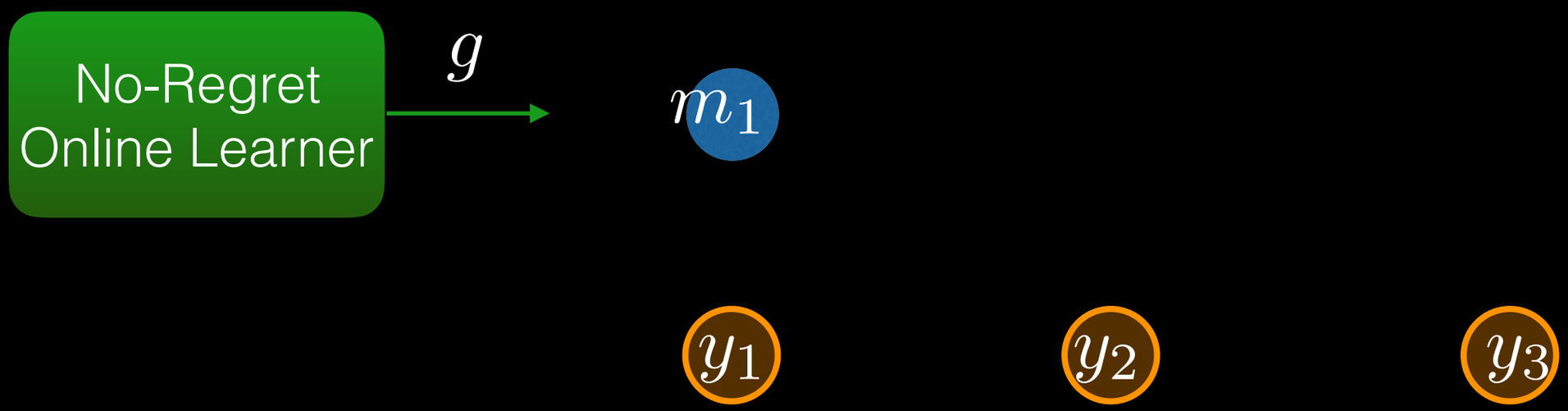
Data Aggregation (DAgger) [Ross & Bagnell 2011]

Follow the (regularized) Leader,
Online Gradient Descent [Zinkevich,2003]

Learning Stationary Filter

Data Aggregation (DAgger) [Ross & Bagnell 2011]

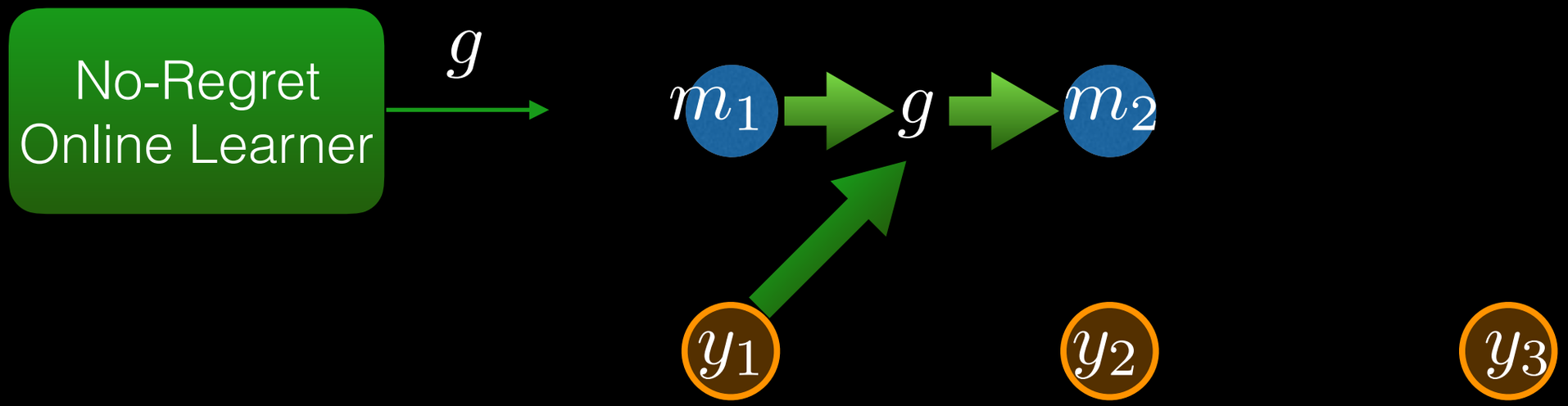
Follow the (regularized) Leader,
Online Gradient Descent [Zinkevich,2003]



Learning Stationary Filter

Data Aggregation (DAgger) [Ross & Bagnell 2011]

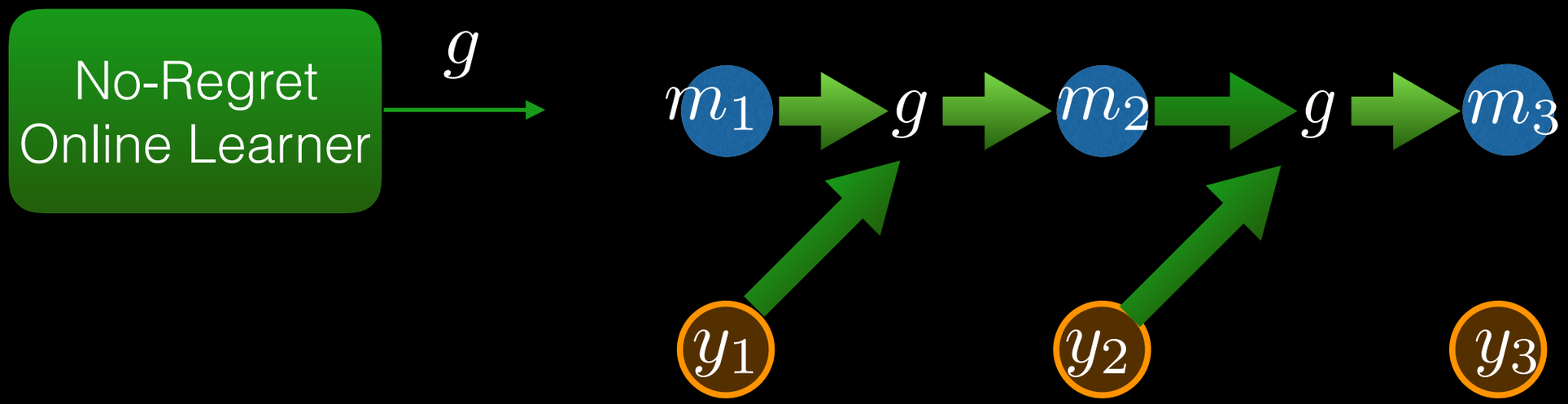
Follow the (regularized) Leader,
Online Gradient Descent [Zinkevich,2003]



Learning Stationary Filter

Data Aggregation (DAgger) [Ross & Bagnell 2011]

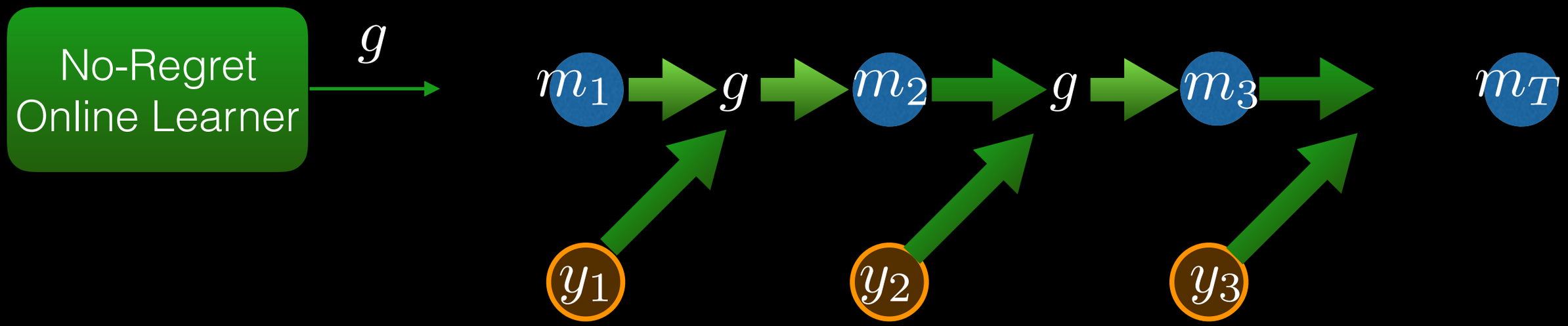
Follow the (regularized) Leader,
Online Gradient Descent [Zinkevich,2003]



Learning Stationary Filter

Data Aggregation (DAgger) [Ross & Bagnell 2011]

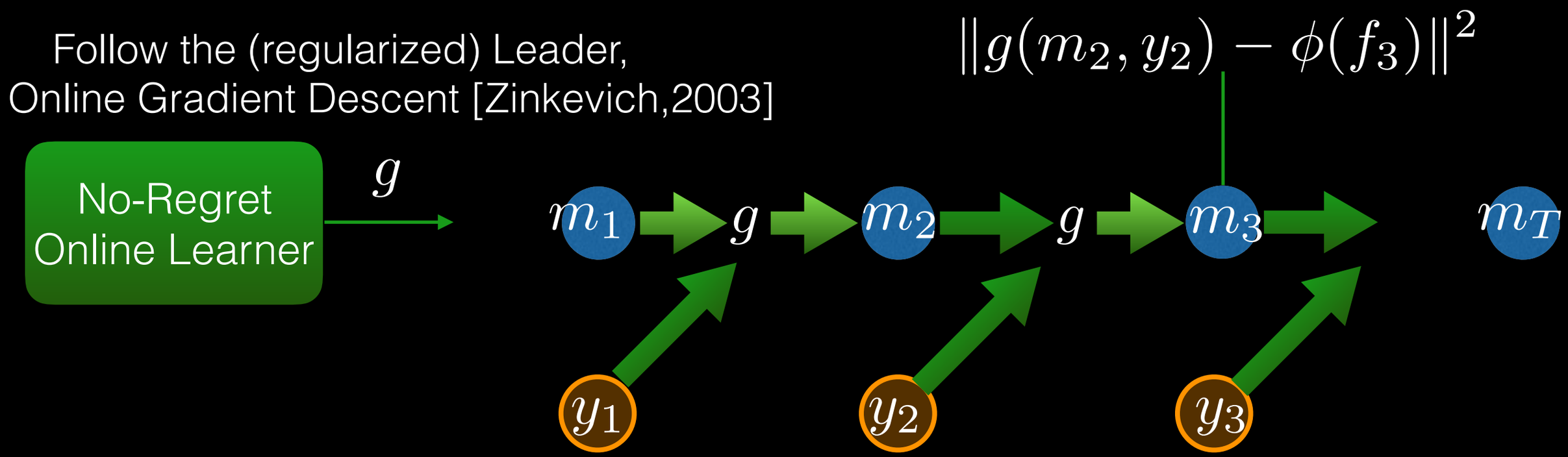
Follow the (regularized) Leader,
Online Gradient Descent [Zinkevich,2003]



Learning Stationary Filter

Data Aggregation (DAgger) [Ross & Bagnell 2011]

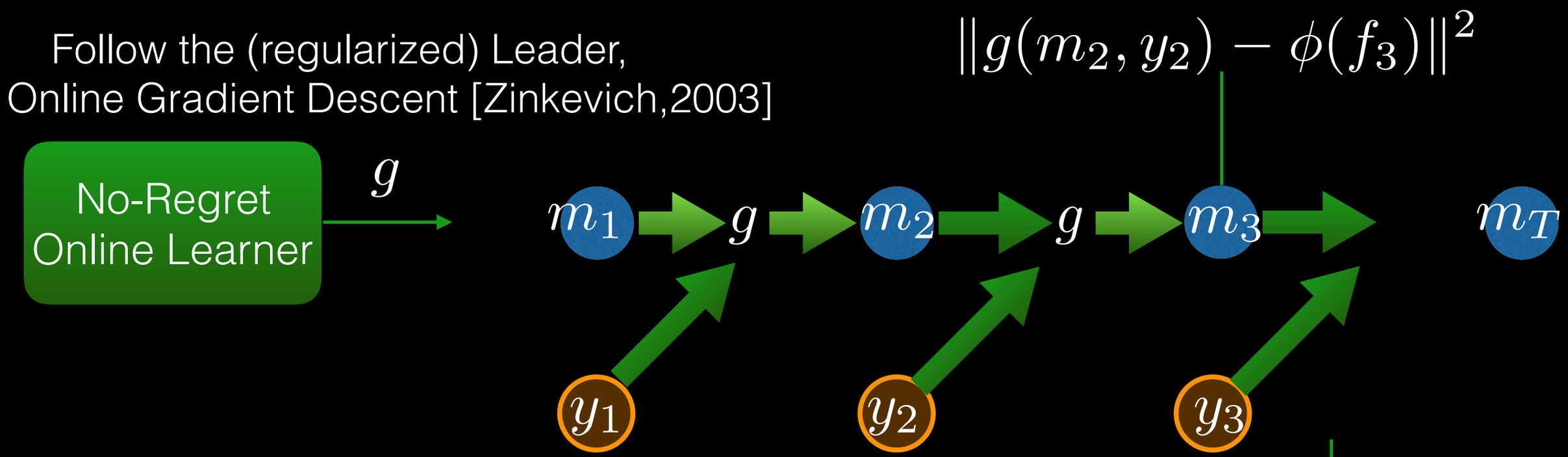
Follow the (regularized) Leader,
Online Gradient Descent [Zinkevich,2003]



Learning Stationary Filter

Data Aggregation (DAgger) [Ross & Bagnell 2011]

Follow the (regularized) Leader,
Online Gradient Descent [Zinkevich,2003]



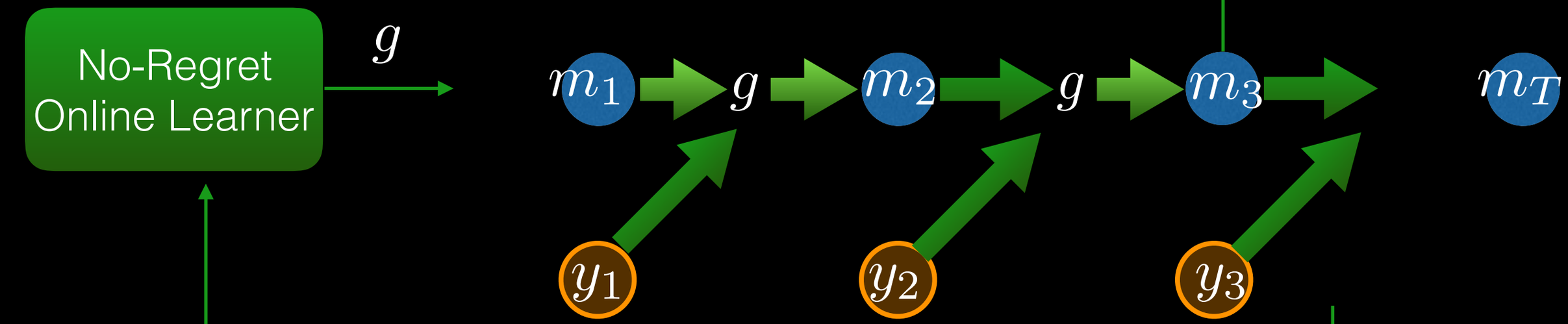
Construct Loss: $\ell_t(g) = \sum_{\tau} \sum_{t} \|g(m_t, y_t) - \phi(f_{t+1})\|^2$

Learning Stationary Filter

Data Aggregation (DAgger) [Ross & Bagnell 2011]

Follow the (regularized) Leader,
Online Gradient Descent [Zinkevich,2003]

$$\|g(m_2, y_2) - \phi(f_3)\|^2$$



Construct Loss: $\ell_t(g) = \sum_{\tau} \sum_{t} \|g(m_t, y_t) - \phi(f_{t+1})\|^2$

Experiments

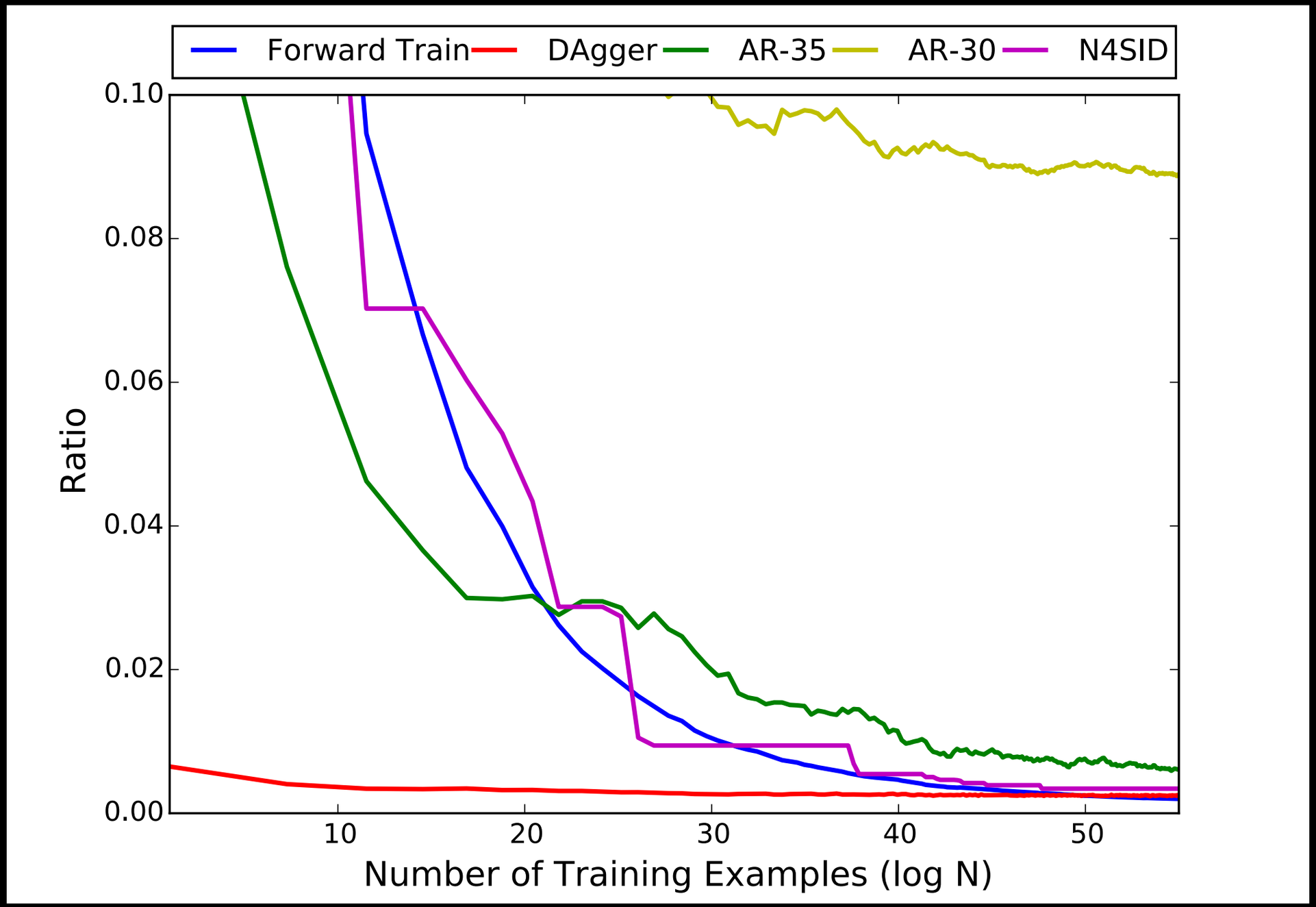
$$x_{t+1} = Ax_t + \epsilon_x, \quad \epsilon_x \sim \mathcal{N}(0, Q)$$
$$y_t = Cx_t + \epsilon_y, \quad \epsilon_y \sim \mathcal{N}(0, R)$$

Learn linear models:

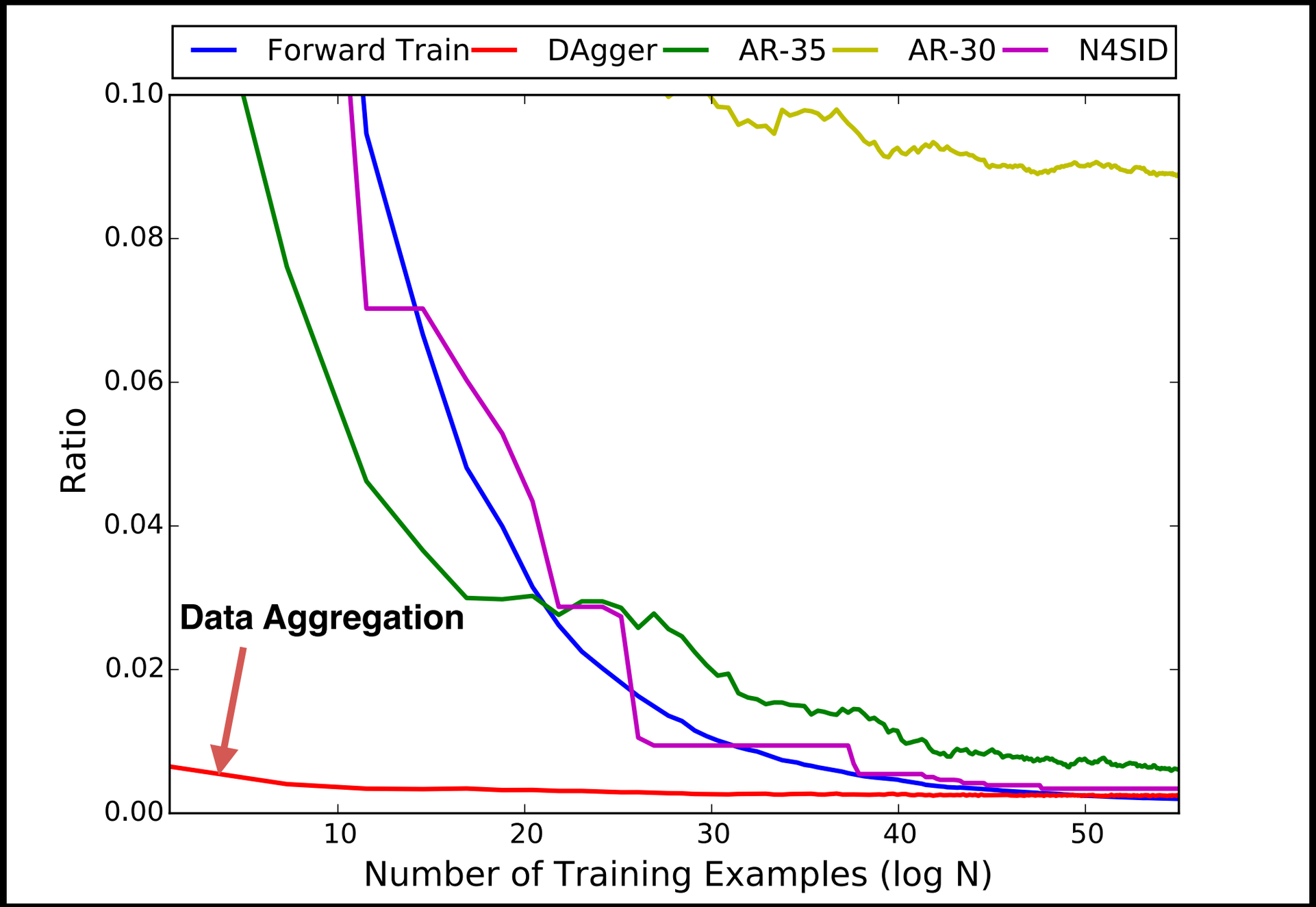
- PSIM with Forward Training (length 2 predictive state)
- PSIM with DAgger (length 2 predictive state)
- 5 Autoregressive models (history lengths = 5,10, 20, 30, 35)
- N4SID

Compare ratio of error of learned predictor to true Kalman Filter Error with different amounts of training data

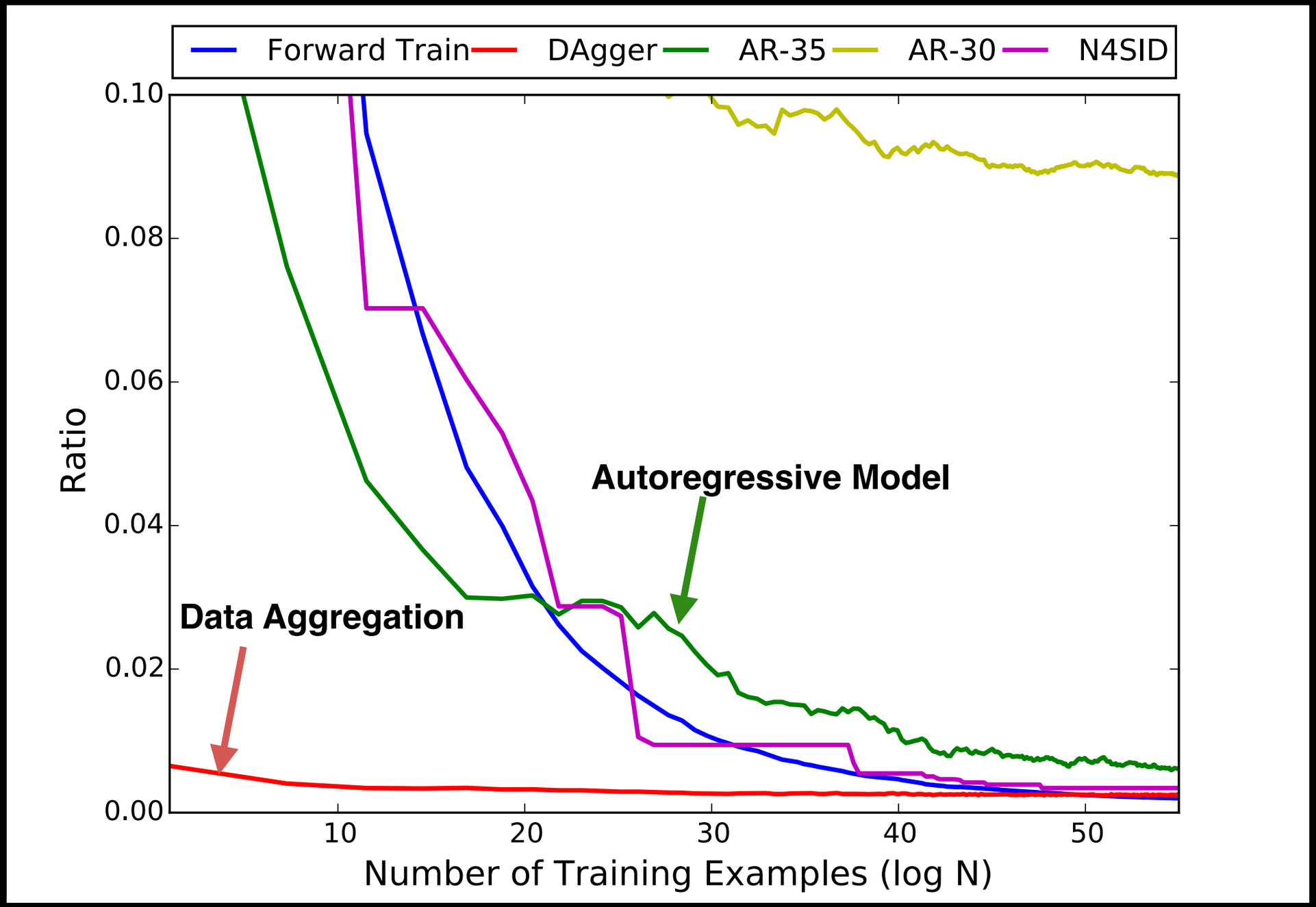
log (error of learned predictor / true Kalman Filter Error)



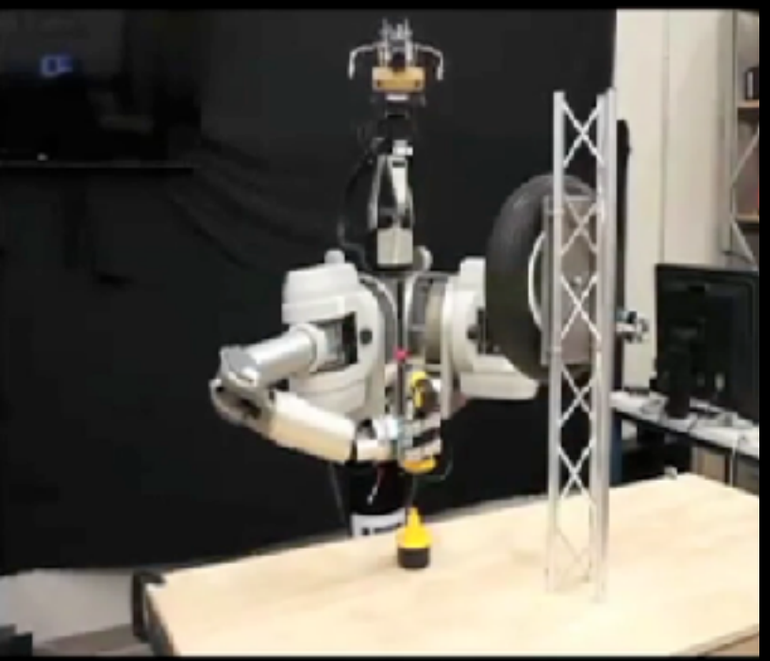
log (error of learned predictor / true Kalman Filter Error)



log (error of learned predictor / true Kalman Filter Error)

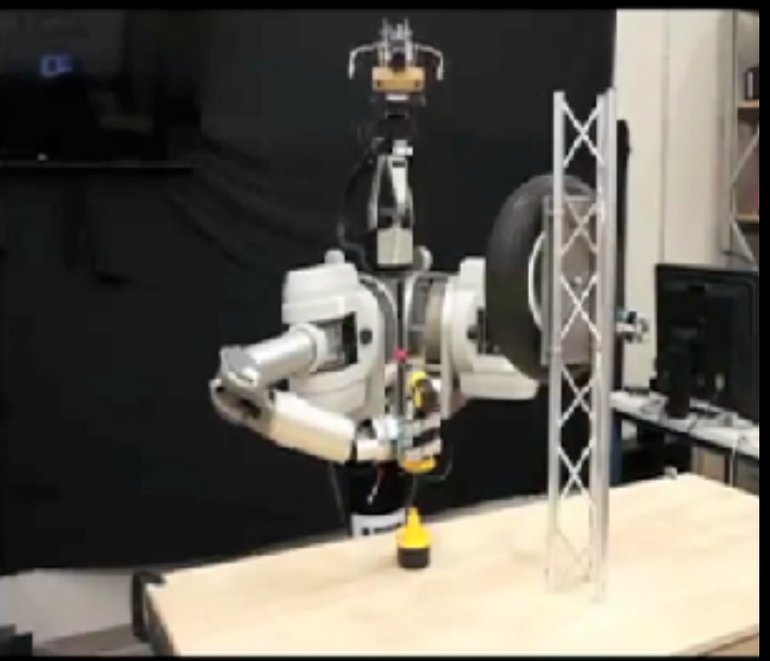


Experiments on Real Dynamical Systems



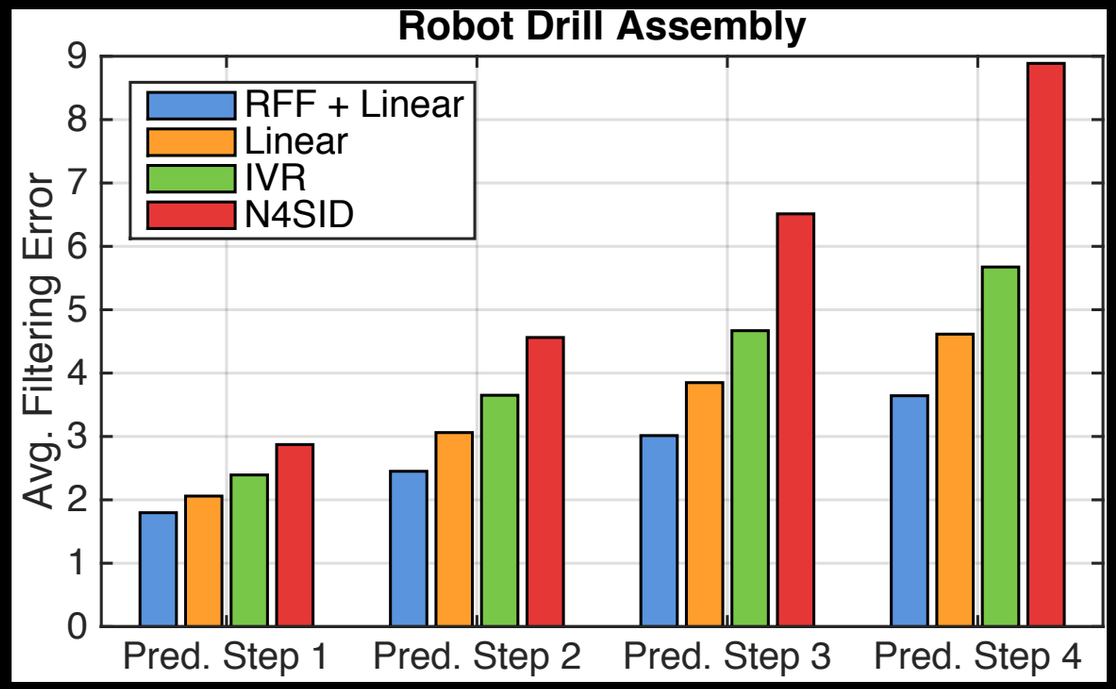
Filtering Goal: predict multi-step future observations, conditioned on history

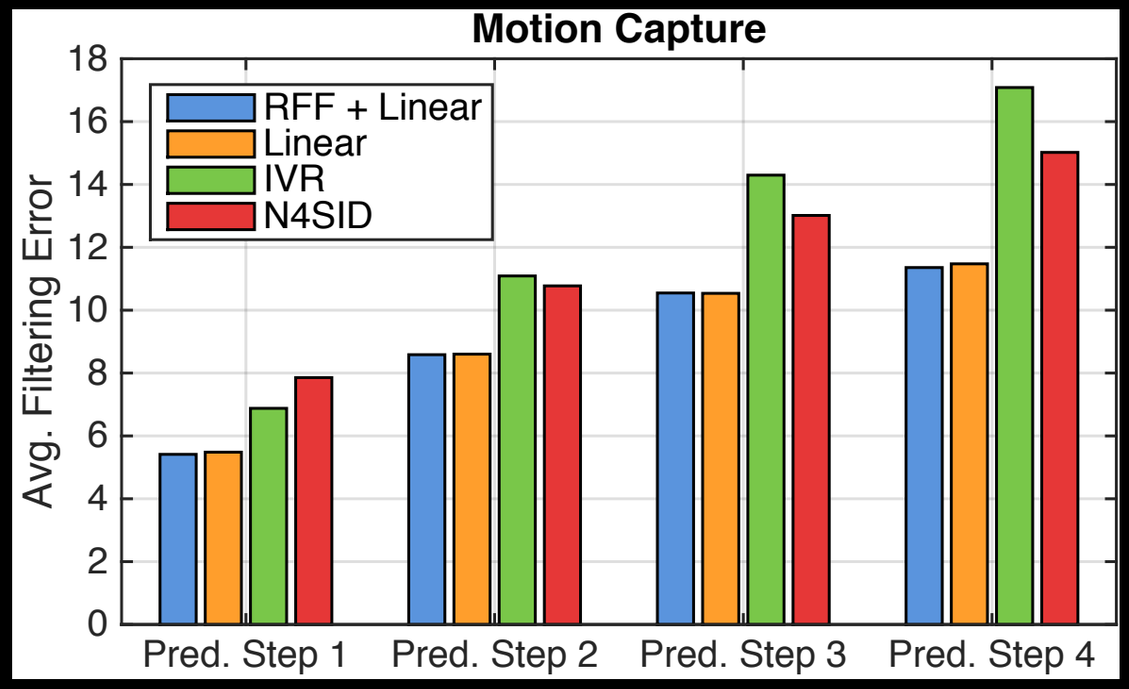
Experiments on Real Dynamical Systems

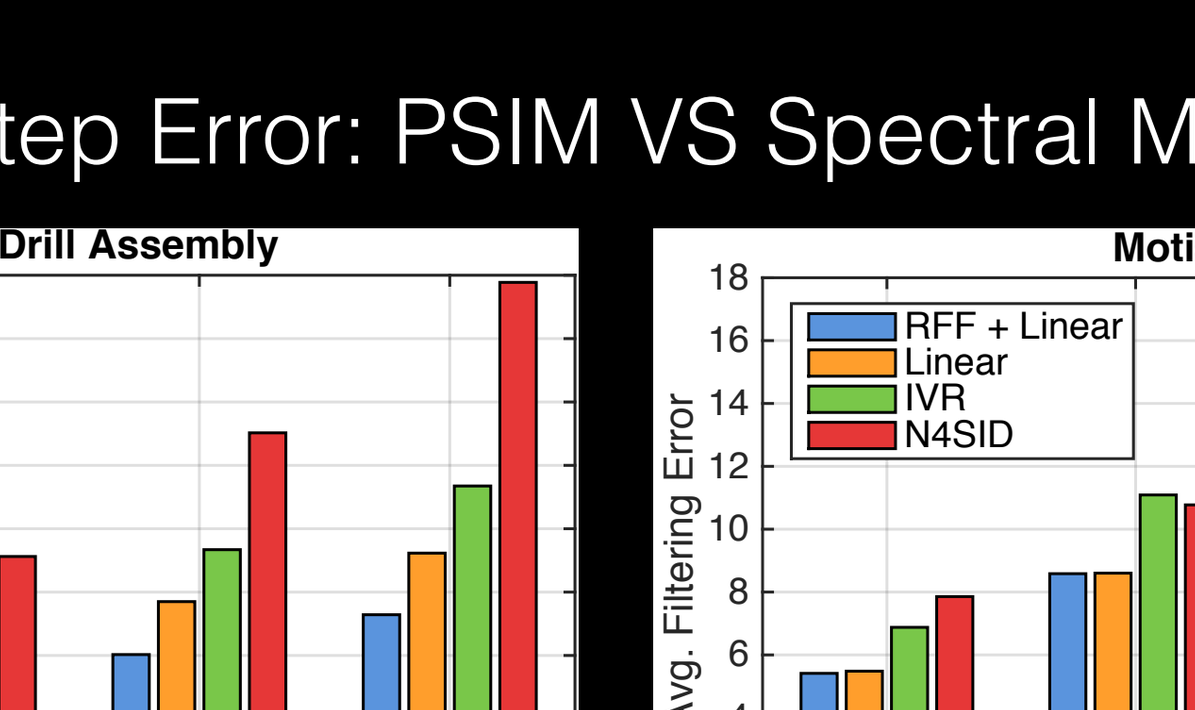


Filtering Goal: predict multi-step future observations, conditioned on history

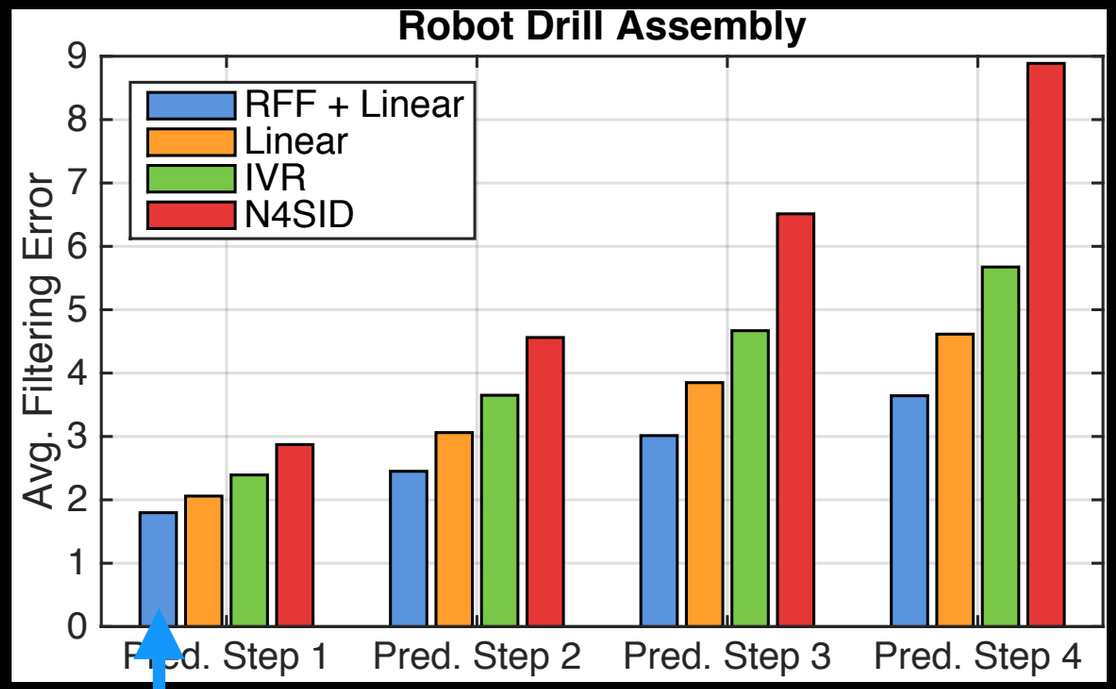
Multi-Step Error: PSIM VS Spectral Methods

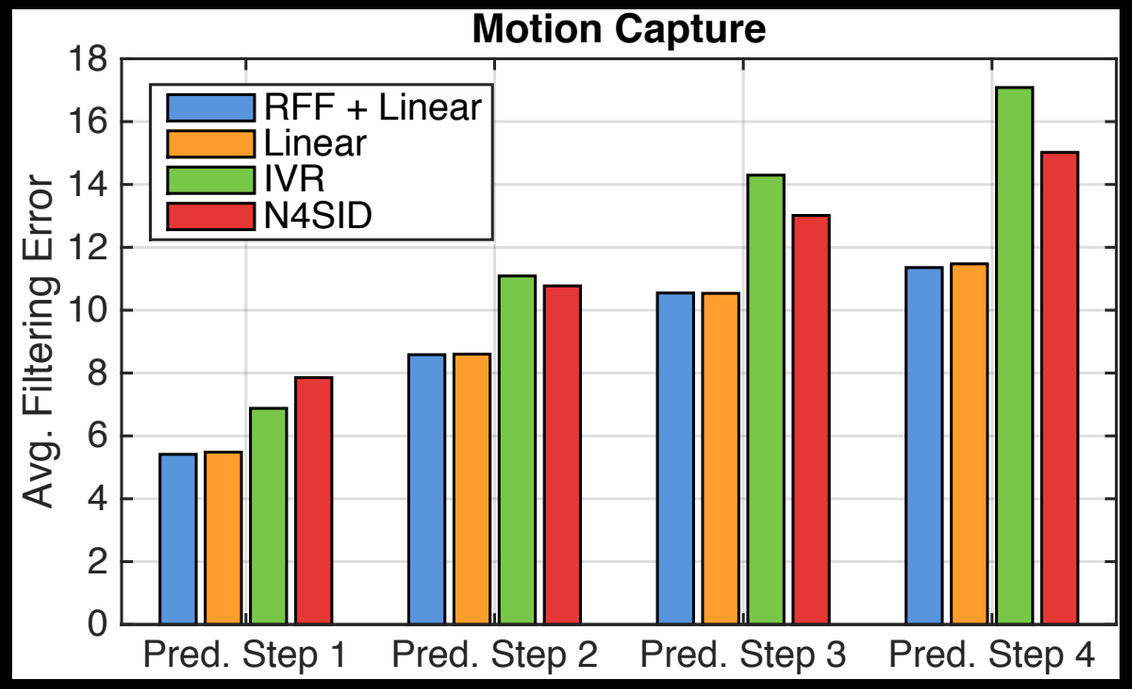




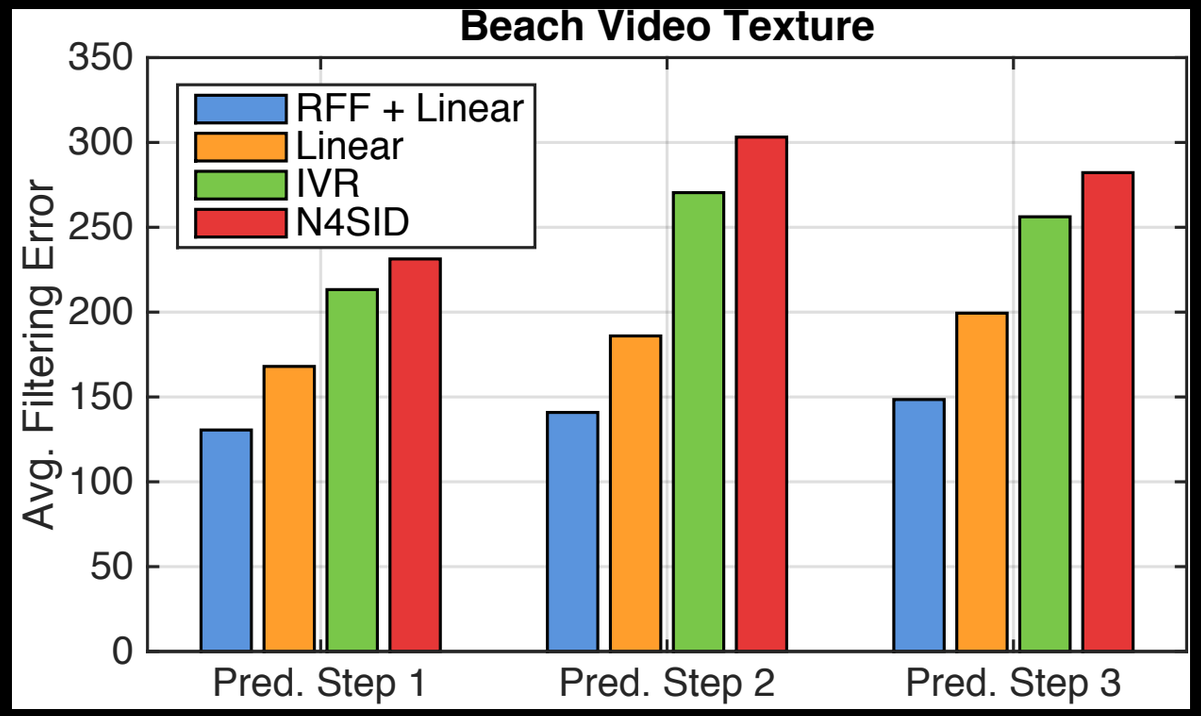


Multi-Step Error: PSIM VS Spectral Methods

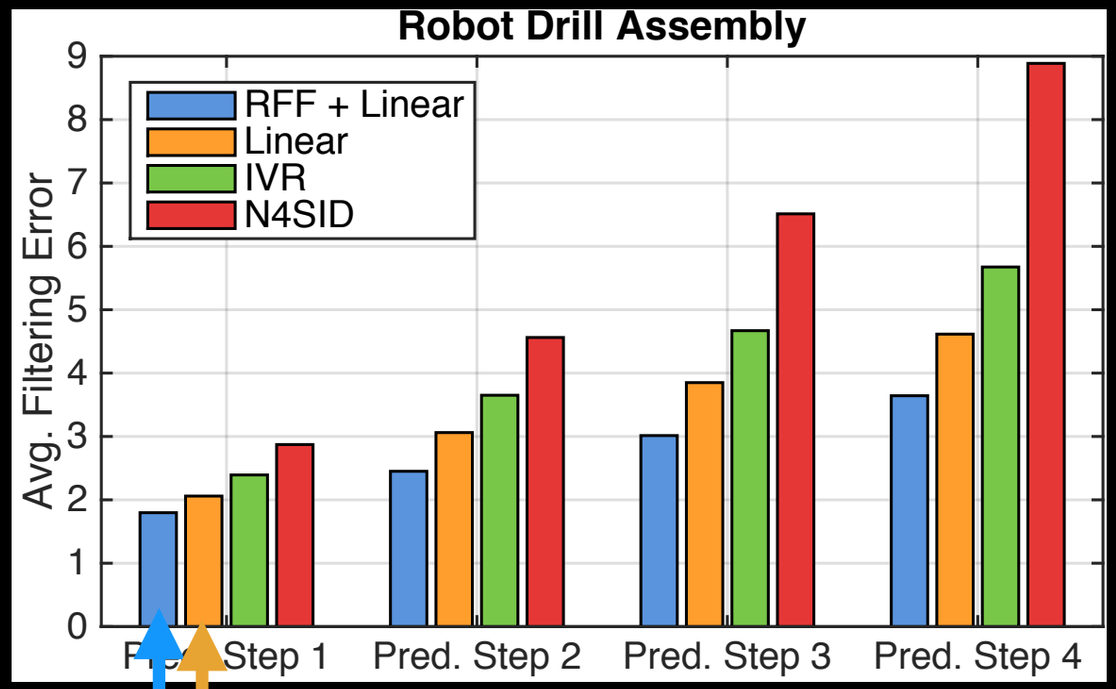


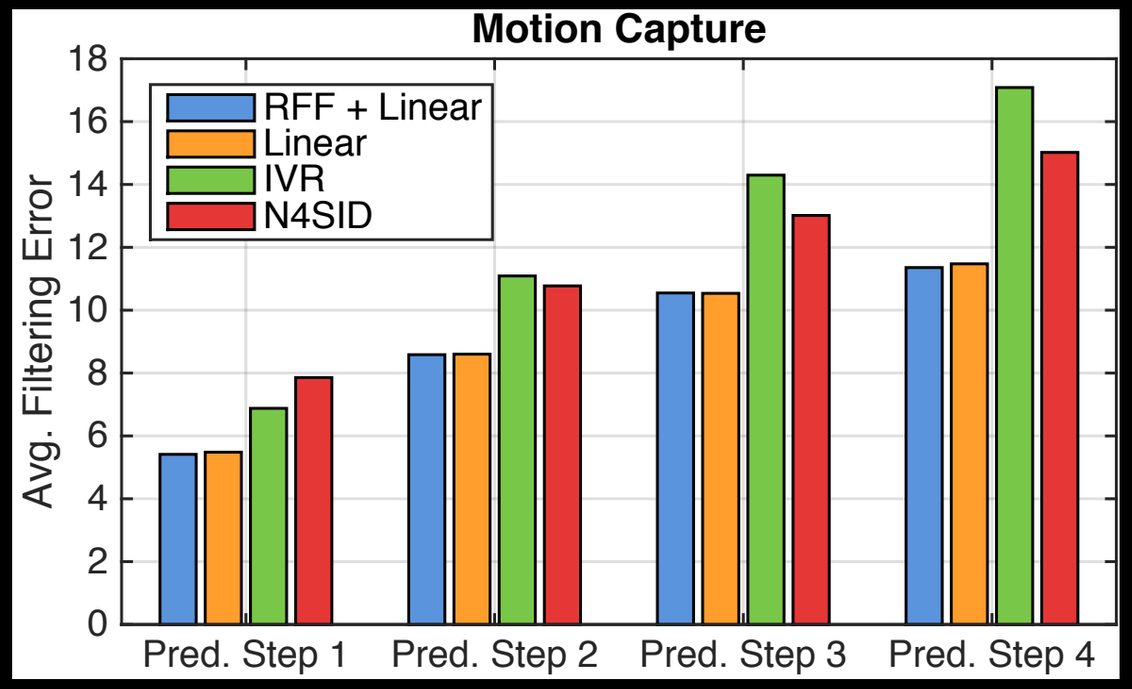


Kernel Regression



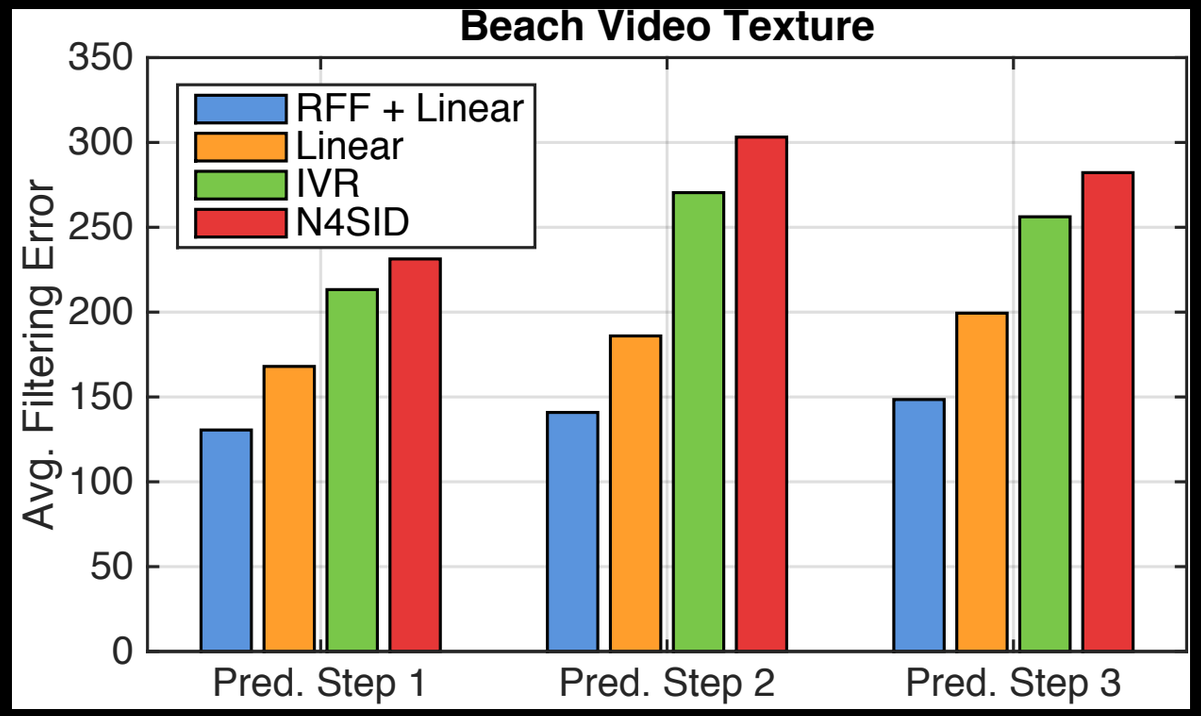
Multi-Step Error: PSIM VS Spectral Methods





Kernel Regression

Linear Regression



Comparison to Recurrent Neural Network

The Way of Encoding of History…

Prediction Error

Comparison to Recurrent Neural Network

The Way of Encoding of History…

RNN: Hidden Units (e.g., Memory Cell in LSTM)

PSIM: Predictive State

Prediction Error

Comparison to Recurrent Neural Network

The Way of Encoding of History…

RNN: Hidden Units (e.g., Memory Cell in LSTM)

PSIM: Predictive State

	PSIM-RFF (Bp)	**PSIM-RFF** (DAgger)	**RNN**
Robot Drill Assembly	2.54	**1.80**	1.99
Motion Capture	9.26	**5.41**	9.6
Beach Video Texture	202.10	**130.53**	346.0

Prediction Error

Conclusion

- Predictive State Inference Machines (PSIM): data-driven approach that directly learns inference procedures for latent state space models.
- Can use powerful, non-linear learner to directly represent the inference procedure
- Similar to RNN, but theoretical Guarantees on inference performance

Predictive State Inference Machines

Thanks

Contact: wensun@cs.cmu.edu

Carnegie Mellon University
THE ROBOTICS INSTITUTE