Useful Models for Robot Learning

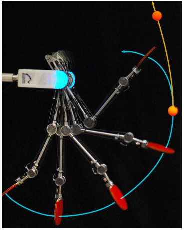
Marc Deisenroth

Centre for Artificial Intelligence
Department of Computer Science
University College London

ELLIS Workshop

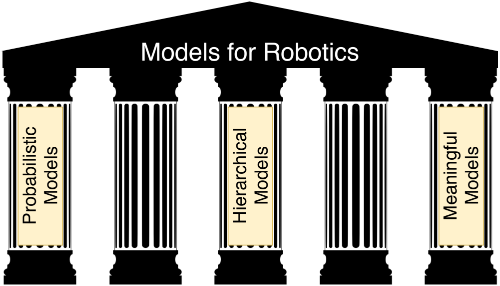
March 5, 2020

Challenges in Robot Learning

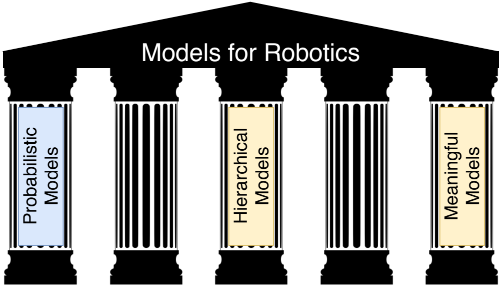




- Automatic adaption in robotics ⏩ Learning
- Practical constraint: data efficiency
- Models are useful for data-efficient learning in robotics

3 Models for Data-Efficient Robot Learning



1. **Probabilistic models**
   - Fast reinforcement learning
2. **Hierarchical models**
   - Infer task similarities within a meta-learning framework
3. **Physically meaningful models**
   - Encode real-world constraints into learning



Carl Rasmussen

Dieter Fox

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t) + \boldsymbol{w}\,, \qquad \boldsymbol{u}_t = \pi(\boldsymbol{x}_t, \boldsymbol{\theta})$$

State

Transition function

Control

Policy

Policy parameters

$$x_{t+1} = f(x_t, u_t) + w\,, \qquad u_t = \pi(x_t, \theta)$$

State — Transition function — Control — Policy — Policy parameters

Objective (Controller Learning)

Find policy parameters θ^* that minimize the expected long-term cost

$$J(\theta) = \sum_{t=1}^{T} \mathbb{E}[c(x_t)|\theta]\,, \qquad p(x_0) = \mathcal{N}(\mu_0, \Sigma_0)\,.$$

Instantaneous cost $c(x_t)$, e.g., $\|x_t - x_{\text{target}}\|^2$

- Typical objective in optimal control and reinforcement learning (Bertsekas, 2005; Sutton & Barto, 1998)

Fast Reinforcement Learning

Objective

Minimize expected long-term cost $J(\boldsymbol{\theta}) = \sum_t \mathbb{E}[c(\boldsymbol{x}_t)|\boldsymbol{\theta}]$

Deisenroth et al. (IEEE-TPAMI, 2015): *Gaussian Processes for Data-Efficient Learning in Robotics and Control*

Objective

Minimize expected long-term cost $J(\boldsymbol{\theta}) = \sum_t \mathbb{E}[c(\boldsymbol{x}_t)|\boldsymbol{\theta}]$

PILCO Framework: High-Level Steps

1. Probabilistic model for transition function
   - ▸▸ System identification

Deisenroth et al. (IEEE-TPAMI, 2015): *Gaussian Processes for Data-Efficient Learning in Robotics and Control*

Objective

Minimize expected long-term cost $J(\boldsymbol{\theta}) = \sum_t \mathbb{E}[c(\boldsymbol{x}_t)|\boldsymbol{\theta}]$

PILCO Framework: High-Level Steps

1. Probabilistic model for transition function
   ▸▸ System identification
2. Compute long-term state evolution $p(\boldsymbol{x}_1|\boldsymbol{\theta}), \ldots, p(\boldsymbol{x}_T|\boldsymbol{\theta})$

Deisenroth et al. (IEEE-TPAMI, 2015): *Gaussian Processes for Data-Efficient Learning in Robotics and Control*

Objective

Minimize expected long-term cost $J(\boldsymbol{\theta}) = \sum_t \mathbb{E}[c(\boldsymbol{x}_t)|\boldsymbol{\theta}]$

PILCO Framework: High-Level Steps

1. Probabilistic model for transition function
   ▸▸ System identification
2. Compute long-term state evolution $p(\boldsymbol{x}_1|\boldsymbol{\theta}), \ldots, p(\boldsymbol{x}_T|\boldsymbol{\theta})$
3. Policy improvement

Deisenroth et al. (IEEE-TPAMI, 2015): *Gaussian Processes for Data-Efficient Learning in Robotics and Control*

Objective

Minimize expected long-term cost $J(\boldsymbol{\theta}) = \sum_t \mathbb{E}[c(\boldsymbol{x}_t)|\boldsymbol{\theta}]$

PILCO Framework: High-Level Steps

1. Probabilistic model for transition function
   ▶▶ System identification
2. Compute long-term state evolution $p(\boldsymbol{x}_1|\boldsymbol{\theta}), \ldots, p(\boldsymbol{x}_T|\boldsymbol{\theta})$
3. Policy improvement
4. Apply controller

Deisenroth et al. (IEEE-TPAMI, 2015): *Gaussian Processes for Data-Efficient Learning in Robotics and Control*

Objective

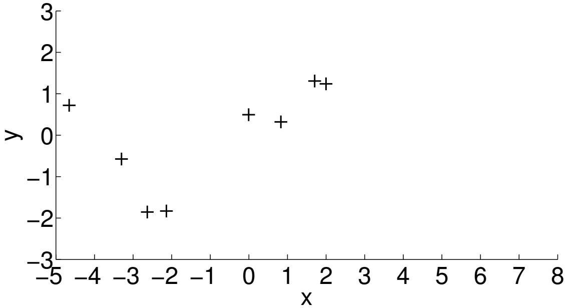
Minimize expected long-term cost $J(\boldsymbol{\theta}) = \sum_t \mathbb{E}[c(\boldsymbol{x}_t)|\boldsymbol{\theta}]$

PILCO Framework: High-Level Steps

1. **Probabilistic model for transition function f**
   ▸▸ **System identification**
2. Compute long-term predictions $p(\boldsymbol{x}_1|\boldsymbol{\theta}), \ldots, p(\boldsymbol{x}_T|\boldsymbol{\theta})$
3. Policy improvement
4. Apply controller

Deisenroth et al. (IEEE-TPAMI, 2015): *Gaussian Processes for Data-Efficient Learning in Robotics and Control*

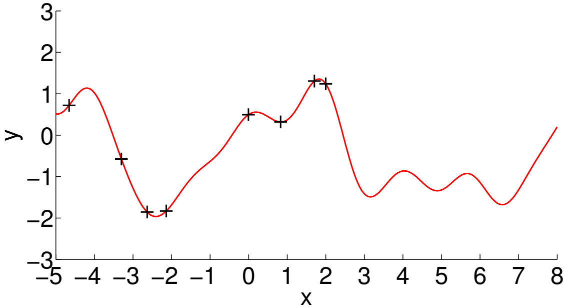
Model learning problem: Find a function $f : x \mapsto f(x) = y$



Observed function values

Model Learning (System Identification)

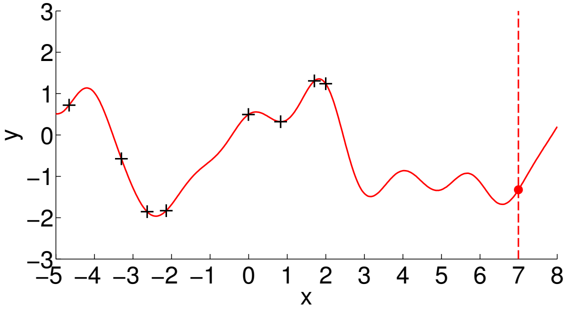
Model learning problem: Find a function $f : x \mapsto f(x) = y$



Plausible model

Model Learning (System Identification)

Model learning problem: Find a function $f : x \mapsto f(x) = y$

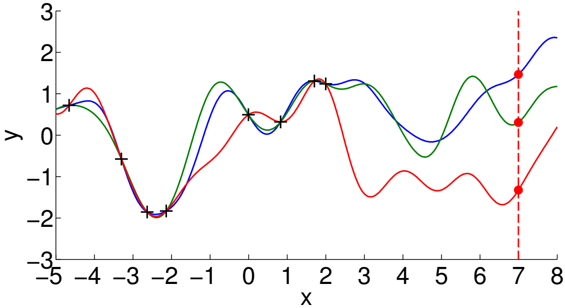


Plausible model

Predictions? Decision Making?

Model Learning (System Identification)

Model learning problem: Find a function $f : x \mapsto f(x) = y$

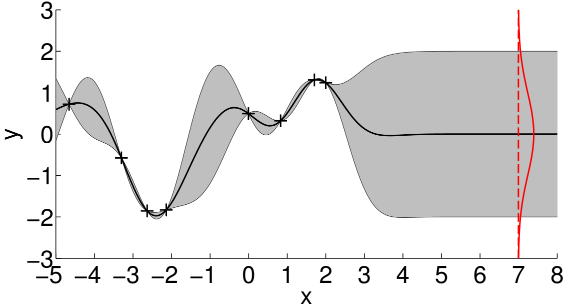


More plausible models

Predictions? Decision Making? Model Errors!

Model Learning (System Identification)

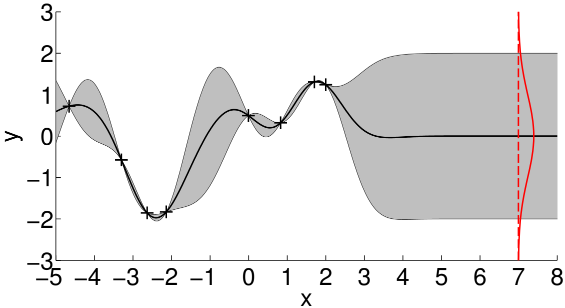
Model learning problem: Find a function $f : x \mapsto f(x) = y$



Distribution over plausible functions

Model Learning (System Identification)

Model learning problem: Find a function $f : x \mapsto f(x) = y$



Distribution over plausible functions

- Express uncertainty about the underlying function to be robust to model errors
- Gaussian process for model learning (Rasmussen & Williams, 2006)

Objective

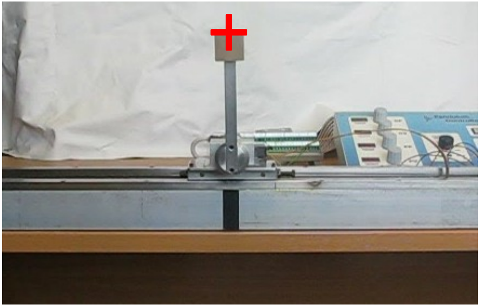
Minimize expected long-term cost $J(\boldsymbol{\theta}) = \sum_t \mathbb{E}[c(\boldsymbol{x}_t)|\boldsymbol{\theta}]$

PILCO Framework: High-Level Steps

1. Probabilistic model for transition function f
   ▸▸ System identification
2. Compute long-term predictions $p(\boldsymbol{x}_1|\boldsymbol{\theta}), \ldots, p(\boldsymbol{x}_T|\boldsymbol{\theta})$
3. Policy optimization via gradient descent
4. Apply controller

Deisenroth et al. (IEEE-TPAMI, 2015): *Gaussian Processes for Data-Efficient Learning in Robotics and Control*

Standard Benchmark: Cart-Pole Swing-up

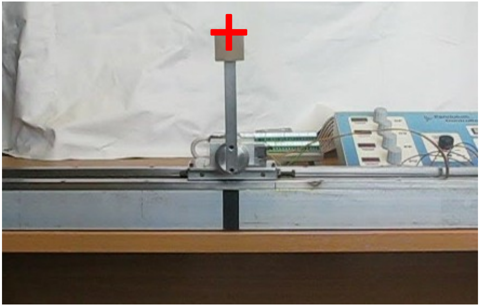


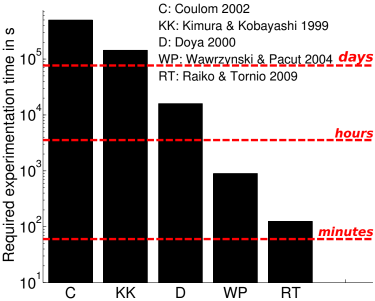
- Swing up and balance a freely swinging pendulum on a cart
- No knowledge about nonlinear dynamics ⏩ Learn from scratch
- Cost function $c(\boldsymbol{x}) = 1 - \exp(-\|\boldsymbol{x} - \boldsymbol{x}_{\text{target}}\|^2)$

- Code: https://github.com/ICL-SML/pilco-matlab

Deisenroth & Rasmussen (ICML, 2011): *PILCO: A Model-based and Data-efficient Approach to Policy Search*

Standard Benchmark: Cart-Pole Swing-up



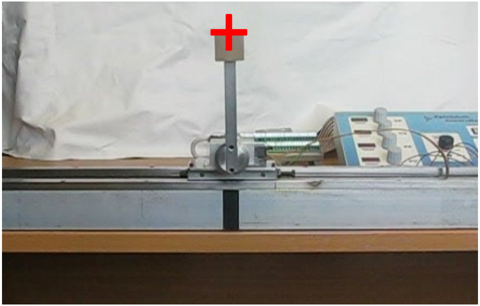


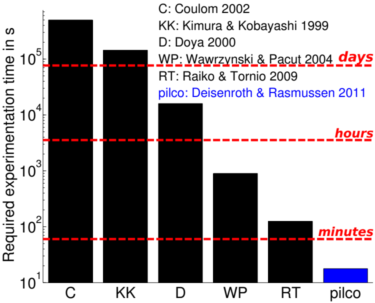
- Swing up and balance a freely swinging pendulum on a cart
- No knowledge about nonlinear dynamics ▶▶ Learn from scratch
- Cost function $c(\boldsymbol{x}) = 1 - \exp(-\|\boldsymbol{x} - \boldsymbol{x}_{\text{target}}\|^2)$

- Code: https://github.com/ICL-SML/pilco-matlab

Deisenroth & Rasmussen (ICML, 2011): *PILCO: A Model-based and Data-efficient Approach to Policy Search*

Standard Benchmark: Cart-Pole Swing-up

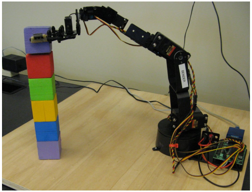




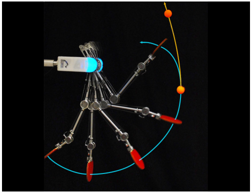
- Swing up and balance a freely swinging pendulum on a cart
- No knowledge about nonlinear dynamics ⏩ Learn from scratch
- Cost function $c(\boldsymbol{x}) = 1 - \exp(-\|\boldsymbol{x} - \boldsymbol{x}_{\text{target}}\|^2)$
- **Unprecedented learning speed** compared to state-of-the-art
- Code: https://github.com/ICL-SML/pilco-matlab

Deisenroth & Rasmussen (ICML, 2011): *PILCO: A Model-based and Data-efficient Approach to Policy Search*

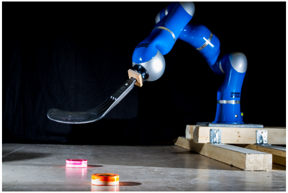
Wide Applicability



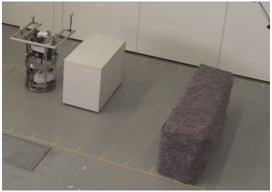
with D Fox



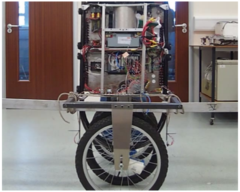
with P Englert, A Paraschos, J Peters



with A Kupcsik, J Peters, G Neumann



B Bischoff (Bosch), ESANN 2013



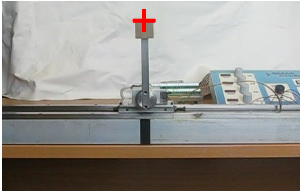
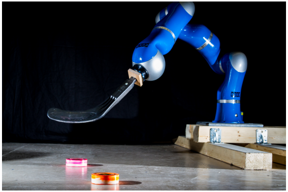
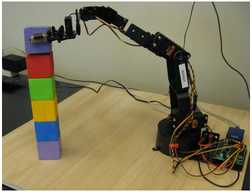
A McHutchon (U Cambridge)

B Bischoff (Bosch), ECML 2013

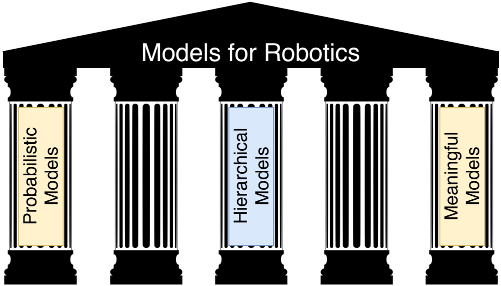
➤ Application to a wide range of robotic systems

Deisenroth et al. (RSS, 2011): *Learning to Control a Low-Cost Manipulator using Data-efficient Reinforcement Learning*
Englert et al. (ICRA, 2013): *Model-based Imitation Learning by Probabilistic Trajectory Matching*
Deisenroth et al. (ICRA, 2014): *Multi-Task Policy Search for Robotics*
Kupcsik et al. (AIJ, 2017): *Model-based Contextual Policy Search for Data-Efficient Generalization of Robot Skills*

Summary (1)

- In robotics, data-efficient learning is critical
- Probabilistic, model-based RL approach
    - Reduce model bias
    - Unprecedented learning speed
    - Wide applicability



Steindór Sæmundsson

Katja Hofmann

Meta Learning (Schmidhuber 1987)

Generalize knowledge from known tasks to new (related) tasks

Meta Learning (Schmidhuber 1987)

Generalize knowledge from known tasks to new (related) tasks

- Different robot configurations (link lengths, weights, ...)
- Re-use experience gathered so far generalize learning to new dynamics that are similar
  ▶▶ Accelerated learning

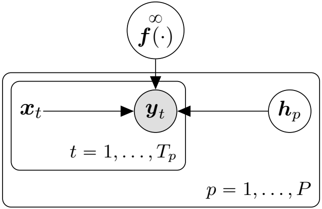
Approach

- Separate global and task-specific properties
- Shared global parameters describe general dynamics
- Describe task-specific (local) properties with latent variable

Approach

- Separate global and task-specific properties
- Shared global parameters describe general dynamics
- Describe task-specific (local) properties with latent variable
- Online variational inference of local properties

Meta Model Learning with Latent Variables

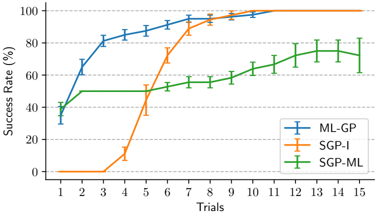


$$\boldsymbol{y}_t = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{h}_p)$$

- GP captures global properties of the dynamics
- Latent variable $\boldsymbol{h}_p$ encodes local properties
  - Variational inference to find a posterior on latent task

Sæmundsson et al. (UAI, 2018): *Meta Reinforcement Learning with Latent Variable Gaussian Processes*

Meta-RL (Cart Pole): Few-Shot Generalization

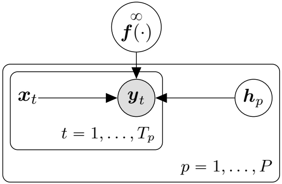


- Train on 6 tasks with different configurations (length/mass)
- Few-shot generalization on 4 unseen configurations
- Success: solve all 10 (6 training + 4 test) tasks
- Meta learning: blue
- Independent (GP-MPC): orange
- Aggregated experience model (no latents): green

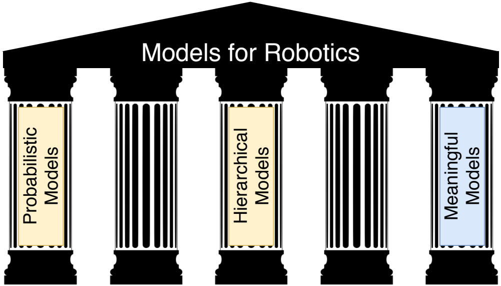
Meta RL generalizes well to unseen tasks

Sæmundsson et al. (UAI, 2018): *Meta Reinforcement Learning with Latent Variable Gaussian Processes*

Summary (2)



- Generalize knowledge from known situations to unseen ones
  ▶▶ **Few-shot learning**
- Latent variable can be used to infer task similarities
- Significant speed-up in model learning and model-based RL

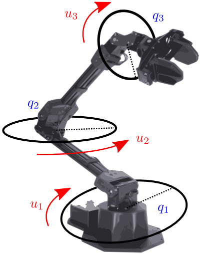



Steindór Sæmundsson


Alexander Terenin


Katja Hofmann

Physically Meaningful Models



Equations of motion

$$u = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q}$$

- Goal: Data efficiency and interpretability
- Inductive biases to account for physical/mechanical properties (e.g., conservation laws, configuration constraints)
  ▶▶ Learn dynamical systems that are "meaningful"

Neural Networks as Dynamical Systems

Approach:

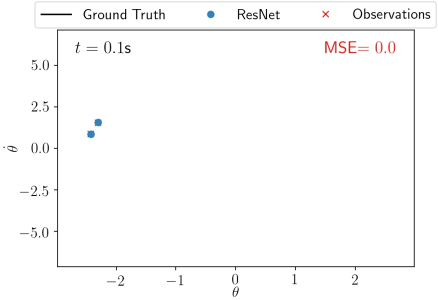
- Euler discretiztion of continuous-time dynamical system

$$\boldsymbol{x}(T)|\boldsymbol{x}_0 = \int_{t=0}^{T} f_\theta(\boldsymbol{x}(t))\mathrm{d}t \approx \boldsymbol{x}_0 + h\sum_{t=0}^{T-1} f_\theta(\boldsymbol{x}_t, t)$$

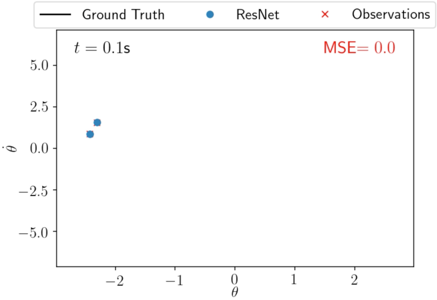
- Deep residual network
  (E, 2017; Haber & Ruthotto, 2017; Chen et al., 2018)

Example: Pendulum

- ODE: $\ddot{\theta} = -\frac{g}{l}\sin\theta$
- Observation: $\boldsymbol{y} = [\theta, \dot{\theta}]^\top$
- Training data: 15 seconds (150 data points)

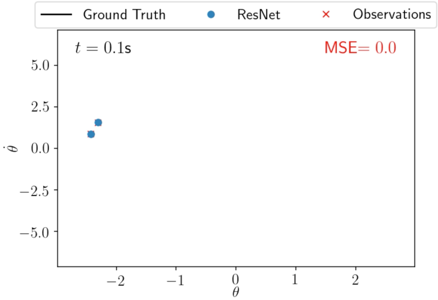


- ODE: $\ddot{\theta} = -\frac{g}{l}\sin\theta$
- Observation: $\boldsymbol{y} = [\theta, \dot{\theta}]^\top + \boldsymbol{\epsilon}, \quad \epsilon \sim \mathcal{N}(\boldsymbol{0}, 0.33^2\boldsymbol{I})$
- Training data: 15 seconds



Example: Pendulum with Noisy Observations

- ODE: $\ddot{\theta} = -\frac{g}{l} \sin \theta$
- Observation: $\boldsymbol{y} = [\theta, \dot{\theta}]^\top + \boldsymbol{\epsilon}, \quad \epsilon \sim \mathcal{N}(\boldsymbol{0}, 0.33^2 \boldsymbol{I})$
- Training data: 15 seconds



- Low prediction quality
- Does not obey physics
- ResNet does not conserve energy

Building Physics into Network Structure

- **Lagrangian**: Encodes "type" of physics, symmetries.

$$L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))$$

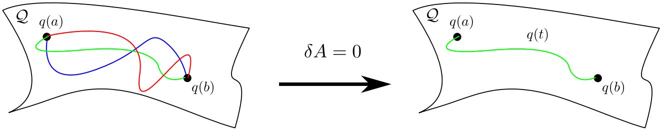
Building Physics into Network Structure

- **Lagrangian**: Encodes "type" of physics, symmetries.

$$L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))$$

- **Hamilton's Principle**:

$$A = \int_a^b L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))\mathrm{d}t\,, \qquad \frac{\delta A}{\delta \boldsymbol{q}(t)} = \boldsymbol{0}$$



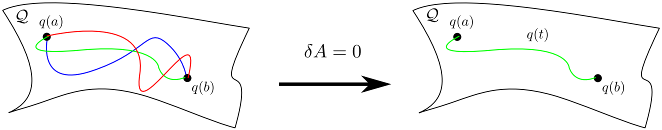
Building Physics into Network Structure

- **Lagrangian**: Encodes "type" of physics, symmetries.

$$L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))$$

- **Hamilton's Principle**:

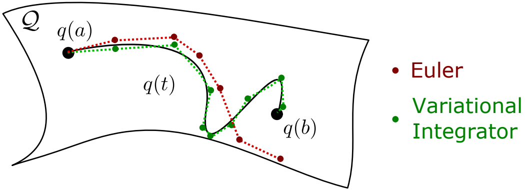
$$A = \int_a^b L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))\mathrm{d}t\,, \quad \frac{\delta A}{\delta \boldsymbol{q}(t)} = \boldsymbol{0}$$



First idea:

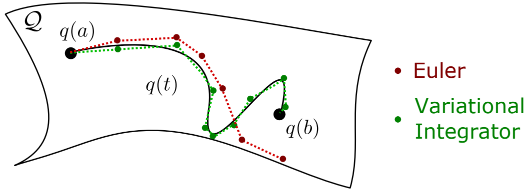
- Learn Lagrangian L instead of dynamics
- Encode physical properties via L (e.g., Lutter et al., 2019; Greydanus et al., 2019)

Variational Integrators



Second idea: Discretize in a way that preserves the physics

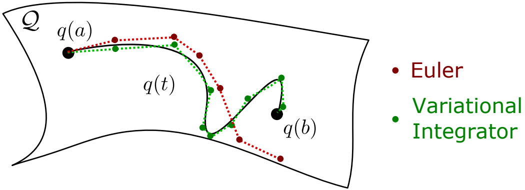
Variational Integrators



Second idea: Discretize in a way that preserves the physics

- Conservative, separable Newtonian system:

$$L_\theta(\boldsymbol{q}, \dot{\boldsymbol{q}}) = T_\theta(\dot{\boldsymbol{q}}) - U_\theta(\boldsymbol{q}) = \underbrace{\tfrac{1}{2}\dot{\boldsymbol{q}}^\top \boldsymbol{M}_\theta \dot{\boldsymbol{q}}}_{\text{kinetic}} - \underbrace{U_\theta(\boldsymbol{q})}_{\text{potential}}$$



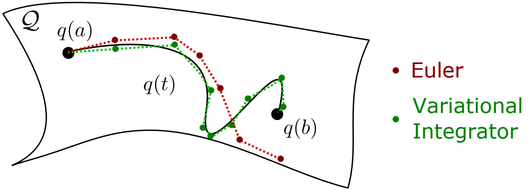
Second idea: Discretize in a way that preserves the physics

- Conservative, separable Newtonian system:

$$L_\theta(\boldsymbol{q}, \dot{\boldsymbol{q}}) = T_\theta(\dot{\boldsymbol{q}}) - U_\theta(\boldsymbol{q}) = \underbrace{\tfrac{1}{2}\dot{\boldsymbol{q}}^\top \boldsymbol{M}_\theta \dot{\boldsymbol{q}}}_{\text{kinetic}} - \underbrace{U_\theta(\boldsymbol{q})}_{\text{potential}}$$

- Discretize action integral A

Variational Integrators



Second idea: Discretize in a way that preserves the physics

- Conservative, separable Newtonian system:

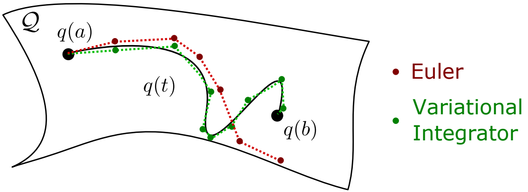
$$L_\theta(\boldsymbol{q}, \dot{\boldsymbol{q}}) = T_\theta(\dot{\boldsymbol{q}}) - U_\theta(\boldsymbol{q}) = \underbrace{\tfrac{1}{2}\dot{\boldsymbol{q}}^\top \boldsymbol{M}_\theta \dot{\boldsymbol{q}}}_{\text{kinetic}} - \underbrace{U_\theta(\boldsymbol{q})}_{\text{potential}}$$

- Discretize action integral A
- Explicit variational integrator

$$\boldsymbol{x}_{t+1} = f_\theta(\boldsymbol{x}_1, t, h)\,, \quad \boldsymbol{x}_t := [\boldsymbol{q}_t, \boldsymbol{q}_{t-1}]$$

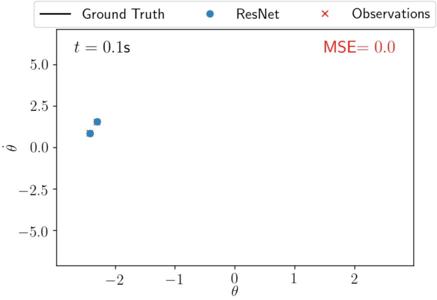
with initial condition $\boldsymbol{x}_1$

Variational Integrators: Properties



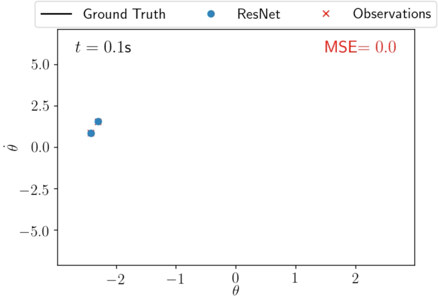
- Physical properties (e.g., conservation laws) automatically enforced
- Flexibility retained to model U_θ (e.g., with a neural network)
- Notions of kinetic and potential energy
  ▶▶ Increased interpretability

Example: Pendulum with Noisy Observations



Sæmundsson et al. (AISTATS 2020): *Variational Integrator Networks for Physically Structured Embeddings*

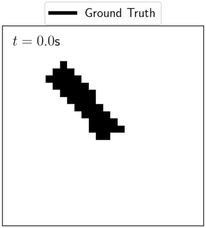
Example: Pendulum with Noisy Observations



- Good predictive performance
- Obeys physics
- Conserves energy

Sæmundsson et al. (AISTATS 2020): *Variational Integrator Networks for Physically Structured Embeddings*

Learning from Pixels

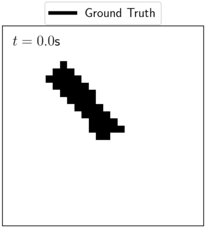


Setting:

- Observations: 28×28 pixel images
- Training data: 60 images (6 seconds of pendulum movement)

Sæmundsson et al. (AISTATS 2020): *Variational Integrator Networks for Physically Structured Embeddings*

Learning from Pixels



Setting:

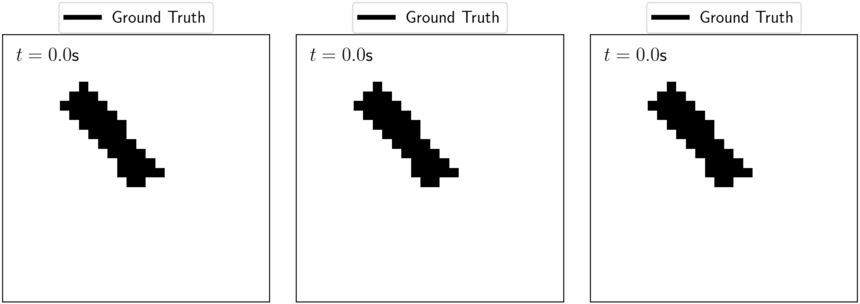
- Observations: 28×28 pixel images
- Training data: 60 images (6 seconds of pendulum movement)

Approach:

- Variational auto-encoder to embed pixels in low-dimensional space
- VIN within low-dimensional space

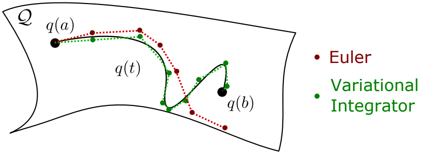
Sæmundsson et al. (AISTATS 2020): *Variational Integrator Networks for Physically Structured Embeddings*

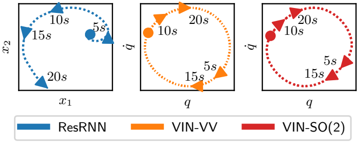
Results



- Residual RNN
- VIN
- VIN on SO(2)
- Code: `https://tinyurl.com/yx3yhhvo`

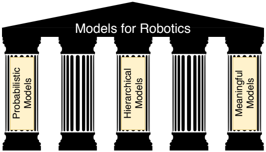
Sæmundsson et al. (AISTATS 2020): *Variational Integrator Networks for Physically Structured Embeddings*

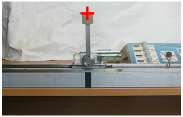


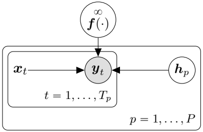


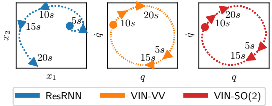
- Encode physics constraints when learning predictive models
- Variational integrator instead of Euler discretization
- Can be combined with VAE to learn predictive models from image observations
- Data efficient and interpretable

Wrap-up



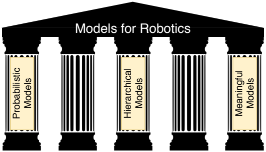


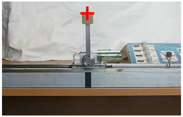


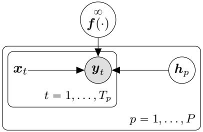


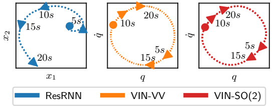
- **Data efficiency** is a practical challenge for autonomous robots
- Three useful models for data-efficient learning in robotics
  1. Probabilistic models for fast reinforcement learning
  2. Hierarchical models for learning task similarities within a meta-learning framework
  3. Physically meaningful models to encode real-world constraints into learning

Wrap-up









- **Data efficiency** is a practical challenge for autonomous robots
- Three useful models for data-efficient learning in robotics
  1. Probabilistic models for fast reinforcement learning
  2. Hierarchical models for learning task similarities within a meta-learning framework
  3. Physically meaningful models to encode real-world constraints into learning

Thank you for your attention

References I

[1] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1 of *Optimization and Computation Series*. Athena Scientific, Belmont, MA, USA, 3rd edition, 2005.

[2] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2 of *Optimization and Computation Series*. Athena Scientific, Belmont, MA, USA, 3rd edition, 2007.

[3] B. Bischoff, D. Nguyen-Tuong, T. Koller, H. Markert, and A. Knoll. Learning Throttle Valve Control Using Policy Search. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, 2013.

[4] Y. Chen, A. Huang, Z. Wang, I. Antonoglou, J. Schrittwieser, D. Silver, and N. de Freitas. Bayesian Optimization in AlphaGo. *arXiv:1812.06855*, 2018.

[5] M. P. Deisenroth, P. Englert, J. Peters, and D. Fox. Multi-Task Policy Search for Robotics. In *Proceedings of the International Conference on Robotics and Automation*, 2014.

[6] M. P. Deisenroth, D. Fox, and C. E. Rasmussen. Gaussian Processes for Data-Efficient Learning in Robotics and Control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423, 2015.

[7] M. P. Deisenroth and C. E. Rasmussen. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *Proceedings of the International Conference on Machine Learning*, 2011.

[8] M. P. Deisenroth, C. E. Rasmussen, and D. Fox. Learning to Control a Low-Cost Manipulator using Data-Efficient Reinforcement Learning. In *Proceedings of Robotics: Science and Systems*, 2011.

[9] W. E. A Proposal on Machine Learning via Dynamical Systems. *Communications in Mathematics and Statistics*, 5(1):1–11, 3 2017.

[10] P. Englert, A. Paraschos, J. Peters, and M. P. Deisenroth. Model-based Imitation Learning by Probabilistic Trajectory Matching. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2013.

[11] P. Englert, A. Paraschos, J. Peters, and M. P. Deisenroth. Probabilistic Model-based Imitation Learning. *Adaptive Behavior*, 21:388–403, 2013.

[12] S. Greydanus, M. Dzamba, and J. Yosinski. Hamiltonian Neural Networks. In *Advances in Neural Information Processing Systems*, 2019.

References II

[13] E. Haber and L. Ruthotto. Stable Architectures for Deep Neural Networks. *Inverse Problems*, 34(1):014004, 2017.

[14] A. Kupcsik, M. P. Deisenroth, J. Peters, L. A. Poha, P. Vadakkepata, and G. Neumann. Model-based Contextual Policy Search for Data-Efficient Generalization of Robot Skills. *Artificial Intelligence*, 2017.

[15] M. Lutter, C. Ritter, and J. Peters. Deep Lagrangian Networks: Using Physics as Model Prior for Deep Learning. In *Proceedings of the International Conference on Learning Representations*, 2019.

[16] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

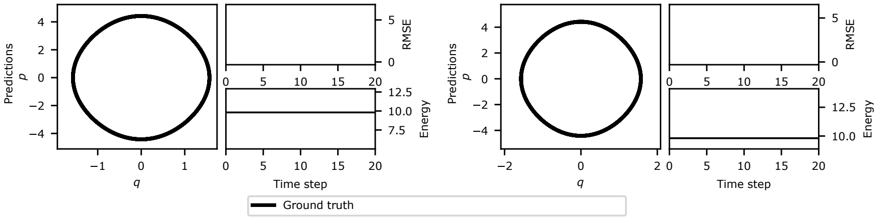
[17] S. Sæmundsson, K. Hofmann, and M. P. Deisenroth. Meta Reinforcement Learning with Latent Variable Gaussian Processes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2018.

[18] J. Schmidhuber. Evolutionary Principles in Self-Referential Learning. Master's thesis, 1987.

[19] S. Sæmundsson, A. Terenin, K. Hofmann, and M. P. Deisenroth. Variational Integrator Networks for Physically Structured Embeddings. In *arXiv:1910.09349*, 2019.

[20] S. Sæmundsson, A. Terenin, K. Hofmann, and M. P. Deisenroth. Variational Integrator Networks for Physically Structured Embeddings. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2020.

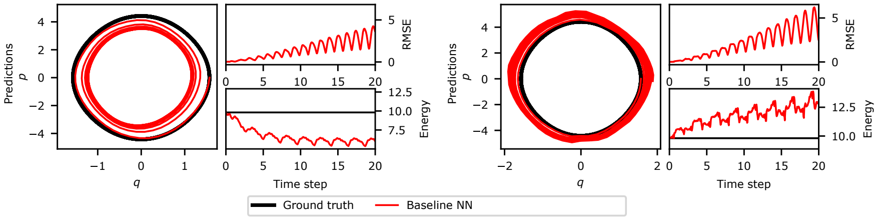
Learning from Noisy Data: Pendulum



Pendulum System. **Left:** 150 observations; **Right:** 750 observations.

Sæmundsson et al. (arXiv:1910.09349): *Variational Integrator Networks for Physically Meaningful Embeddings*

Learning from Noisy Data: Pendulum

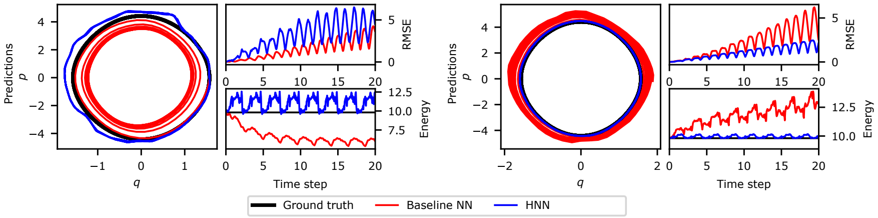


Pendulum System. **Left:** 150 observations; **Right:** 750 observations.

- Baseline neural network: Dissipates/adds energy for low and moderate data

Sæmundsson et al. (arXiv:1910.09349): *Variational Integrator Networks for Physically Meaningful Embeddings*

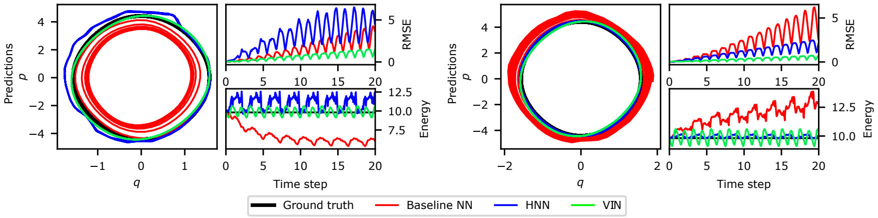
Learning from Noisy Data: Pendulum



Pendulum System. **Left:** 150 observations; **Right:** 750 observations.

- Baseline neural network: Dissipates/adds energy for low and moderate data
- Hamiltonian neural network (Greydanus et al., 2019): Overfits in low-data regime

Sæmundsson et al. (arXiv:1910.09349): *Variational Integrator Networks for Physically Meaningful Embeddings*

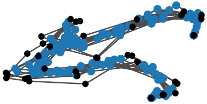


Pendulum System. **Left:** 150 observations; **Right:** 750 observations.

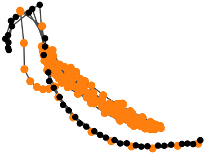
- Baseline neural network: Dissipates/adds energy for low and moderate data
- Hamiltonian neural network (Greydanus et al., 2019): Overfits in low-data regime
- Variational integrator network: Conserves energy and generalizes better in both regimes

Sæmundsson et al. (arXiv:1910.09349): *Variational Integrator Networks for Physically Meaningful Embeddings*

Latent Embeddings of Time Series

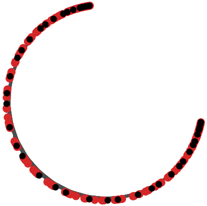


(a) VAE

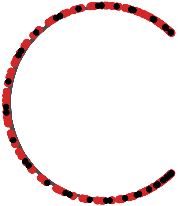


(b) Dynamic VAE

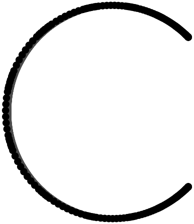
(c) Lie Group VAE



(d) VIN-$SO(2)$



(e) VIN-$SO(2)$ with fixed $\boldsymbol{M}$



(f) Ground Truth