Fraunhofer IIS

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Introduction to Model-Based Reinforcement Learning

Christopher Mutschler

Outline

- Motivation: why model-based RL?
- What is a model? What are its inputs? What is a good model?
- How can we use a model?
  - Background Planning
    - Environment data augmentation / simulation
    - Sample-efficient policy learning
  - Online Planning
    - Discrete Actions
    - Continuous Actions
  - Auxiliary tasks
- Real-world application

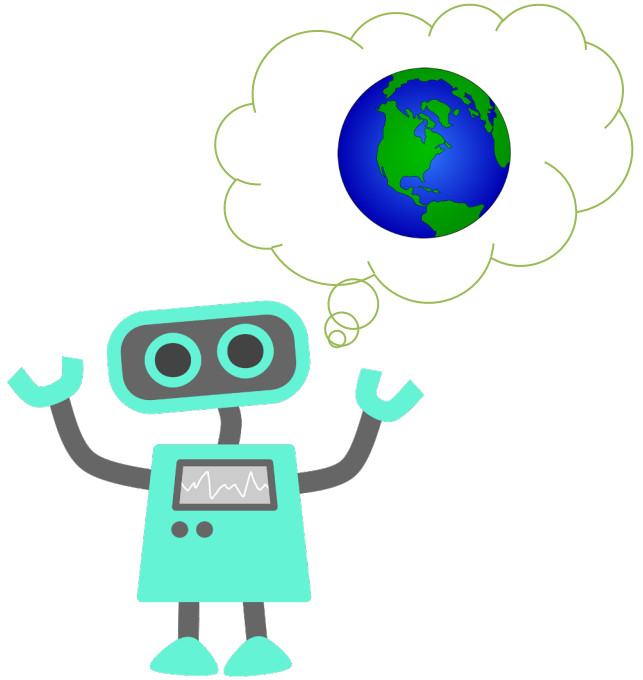
Outline

- **Motivation: why model-based RL?**
- What is a model? What are its inputs? What is a good model?
- How can we use a model?
  - Background Planning
    - Environment data augmentation / simulation
    - Sample-efficient policy learning
  - Online Planning
    - Discrete Actions
    - Continuous Actions
  - Auxiliary tasks
- Real-world application

Why Model-based RL?

"If the organism carries a **'small-scale model' of external reality** and of its own possible actions within its head, it is able to try out various alternatives, conclude which is the best of them, react to future situations before they arise, utilise the knowledge of past events in dealing with the present and future, and in every way to react in a much **fuller, safer, and more competent** manner to the emergencies which face it."

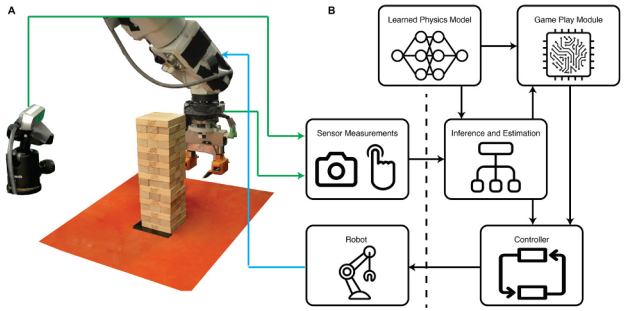
—*Craik, 1943, The Nature of Explanation*



Slide adapted from https://sites.google.com/view/mbrl-tutorial

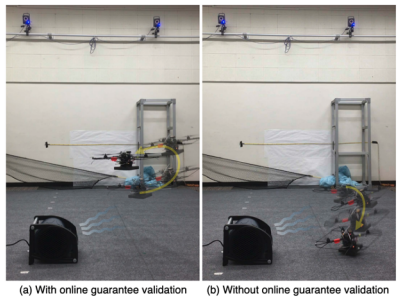
Why Model-based RL?

Robotic control



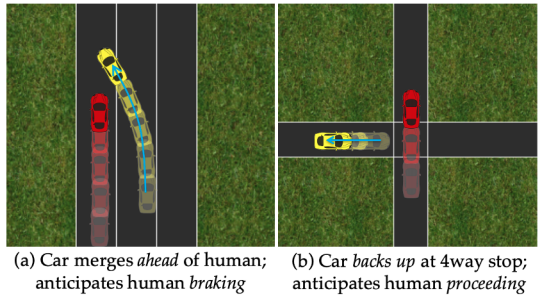
Fazeli et al. (2019). See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion. Science Robotics, 4(26).

Safety



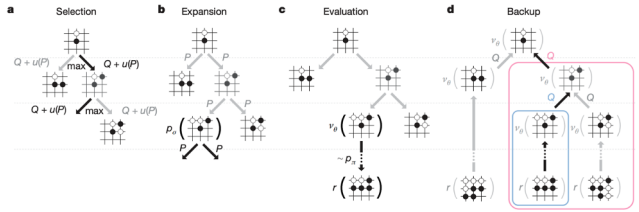
Fisac et al. (2019). A General Safety Framework for Learning-Based Control in Uncertain Robotic Systems. IEEE Transactions on Automatic Control.

HMI: H-AI-I



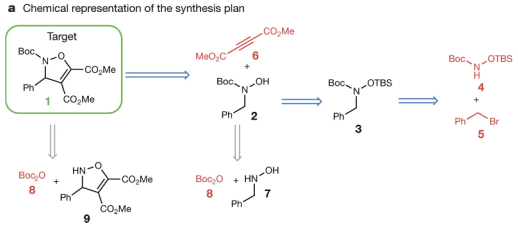
Sadigh et al. (2016). Planning for autonomous cars that leverage effects on human actions. RSS 2016.

Games



Silver et al. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484.

Science



Segler, Preuss, & Waller (2018). Planning chemical syntheses with deep neural networks and symbolic AI. Nature, 555(7698).

Operations Research

[1]

Salas & Powell (2013). Benchmarking a scalable approximate dynamic programming algorithm for stochastic control of multidimensional energy storage problems.

[1] *Warren Powell's 2017 ECSO tutorial, "A Unified Framework for Optimization under Uncertainty"*

Why Model-based RL?

- Model free vs. Model-based Reinforcement Learning

	Model-free	Model-based
Asymptotic Rewards	+	"depends"
Computation at deployment	+	- / +
Data efficiency	-	+
Speed to adapt to changing rewards	-	+
Speed to adapt to changing dynamics	-	+
Exploration	-	+

Outline

- Motivation: why model-based RL?
- **What is a model? What are its inputs? What is a good model?**
- How can we use a model?
  - Background Planning
    - Environment data augmentation / simulation
    - Sample-efficient policy learning
  - Online Planning
    - Discrete Actions
    - Continuous Actions
  - Auxiliary Tasks
- Real-world application

What is a model?

- Quick recap: sequential decision making
- Our goal:

$$\arg\max_{\pi} \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t),$$

subject to: $a_t = \pi(s_t)$, and

$$s_{t+1} = T(s_t, a_t)$$

- We collect data $\mathcal{D} = \{s_t, a_t, r_{t+1}, s_{t+1}\}_{t=0}^{T}$

- **Model-free RL:** learn policy directly from data

  $\mathcal{D} \rightarrow \pi$, e.g., with Q-Learning, policy gradients, …

- **Model-based RL:** learn model, then use it to learn or improve policy:

  $\mathcal{D} \rightarrow f \rightarrow \pi$

What is a model?

Definition: *A model is a representation that **explicitly** encodes knowledge about the structure of the environment and task.*

But what knowledge to explicitly encode?

- Transition/dynamics model: $s_{t+1} = f_s(s_t, a_t)$
- Reward model: $r_{t+1} = f_r(s_t, a_t)$

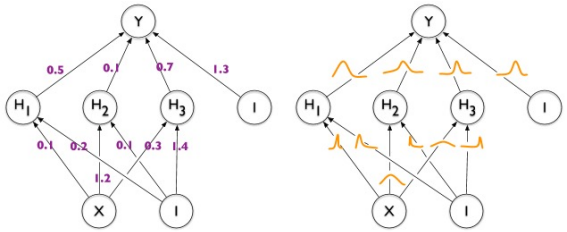
** typically what MBRL is doing*

- Inverse dynamics model: $a_t = f_s^{-1}(s_t, s_{t+1})$
- Distance model: $d_{ij} = f_d(s_i, s_j)$
- Future return model: $G_t = Q(s_t, a_t)$ or $G_t = V(s_t)$

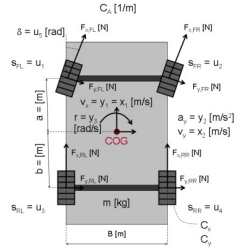
What is a model?

Parametric vs. non-parametric representations

- Parametric

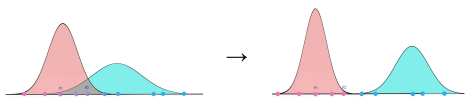


Neural Networks

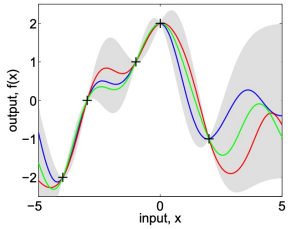


Physics-based

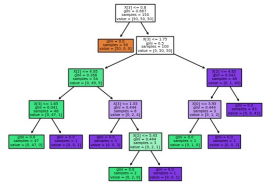
- Non-parametric



Gaussian Mixture Models



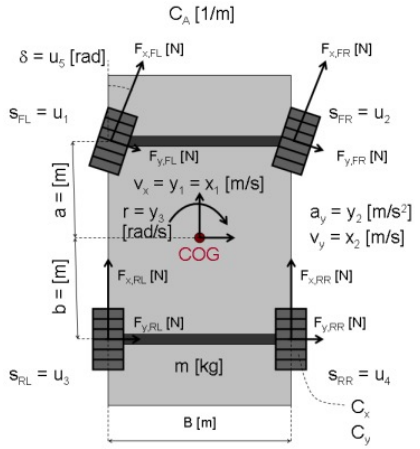
Gaussian Processes

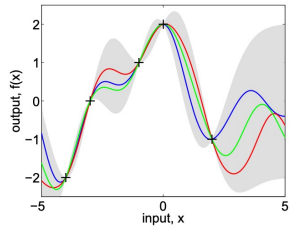


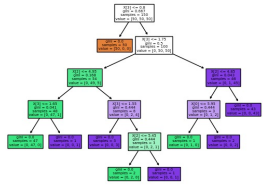
Decision Trees or Random Forests

Model inputs: States

- Dynamical system states (classical control theory)







The Bicycle model is given by,

$$\dot{X} = v_x cos(\psi) - v_y sin(\psi) \quad (2.31a)$$
$$\dot{Y} = v_x sin(\psi) + v_y cos(\psi) \quad (2.31b)$$
$$\dot{\psi} = \frac{v_x}{l_f + l_r} tan(\delta) \quad (2.31c)$$
$$m\dot{v}_x = F_x + mv_y\dot{\psi} - 2F_{cf} sin(\delta) - F_a - F_r \quad (2.31d)$$
$$m\dot{v}_y = -mv_x\dot{\psi} + 2(F_{cf} cos(\delta) + F_{cr}) \quad (2.31e)$$
$$I\ddot{\psi} = 2(l_f F_{cf} cos(\delta) - l_r F_{cr}) \quad (2.31f)$$

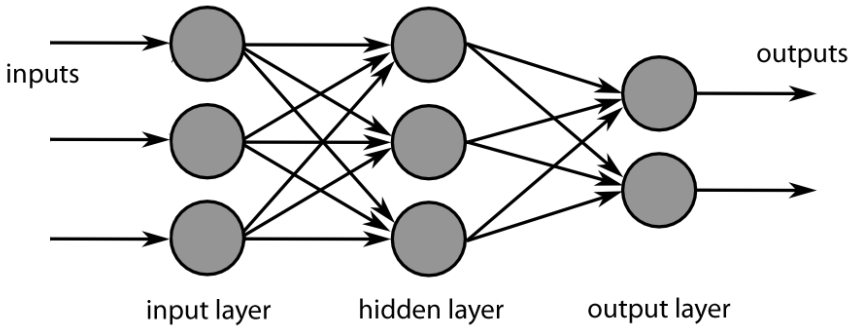
https://de.mathworks.com/help/ident/ug/modeling-a-vehicle-dynamics-system.html

M. I. Palmqvist et al.: Model predictive control for autonomous driving of a truck. KTH Royal Institute of Technology School of Electrical Engineering. 2016.

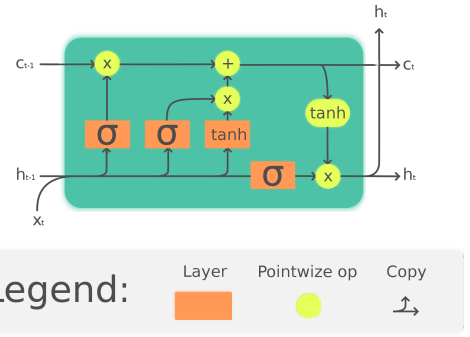
Model inputs: States

Pre-processed sensor data, e.g.

- objects/bounding boxes extracted from images)



https://de.wikipedia.org/wiki/Deep_Learning

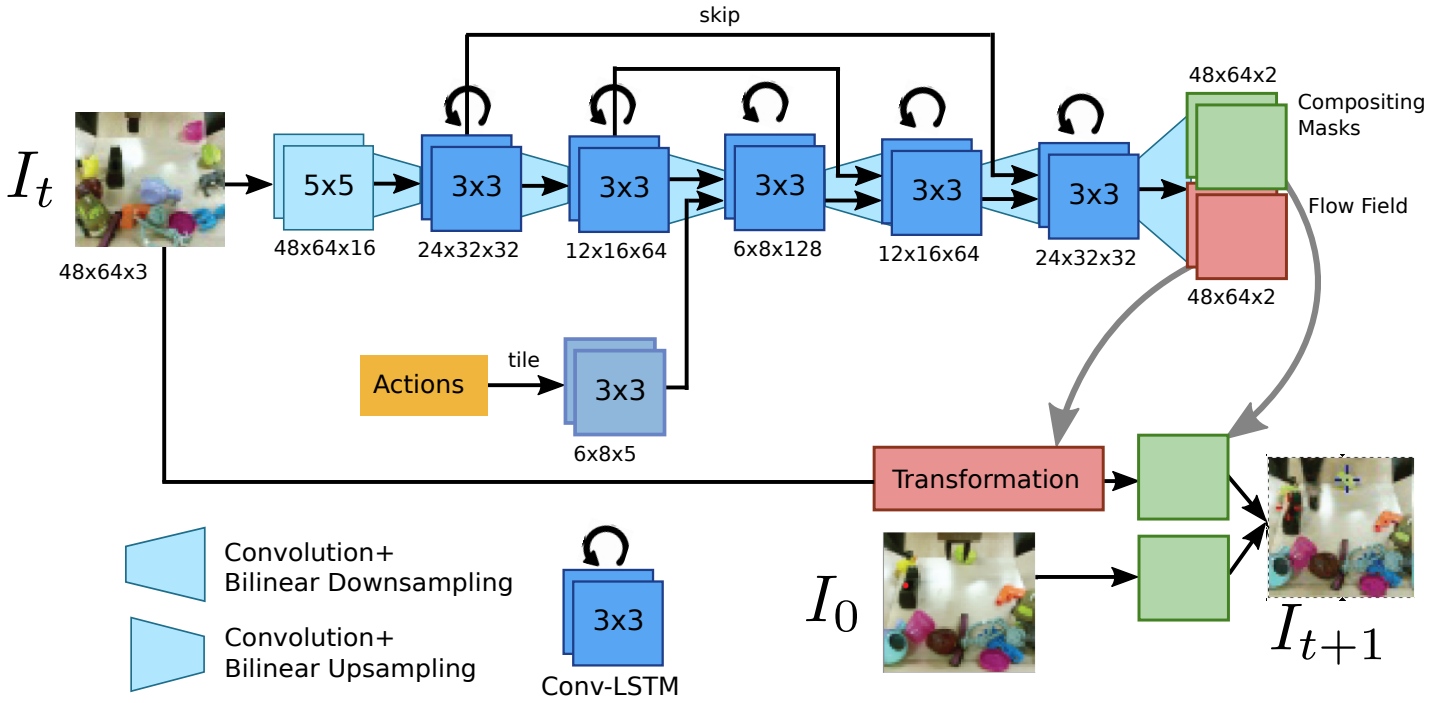


https://en.wikipedia.org/wiki/Long_short-term_memory

Model inputs: Observations

Raw sensor data (e.g., images or LIDAR scans)

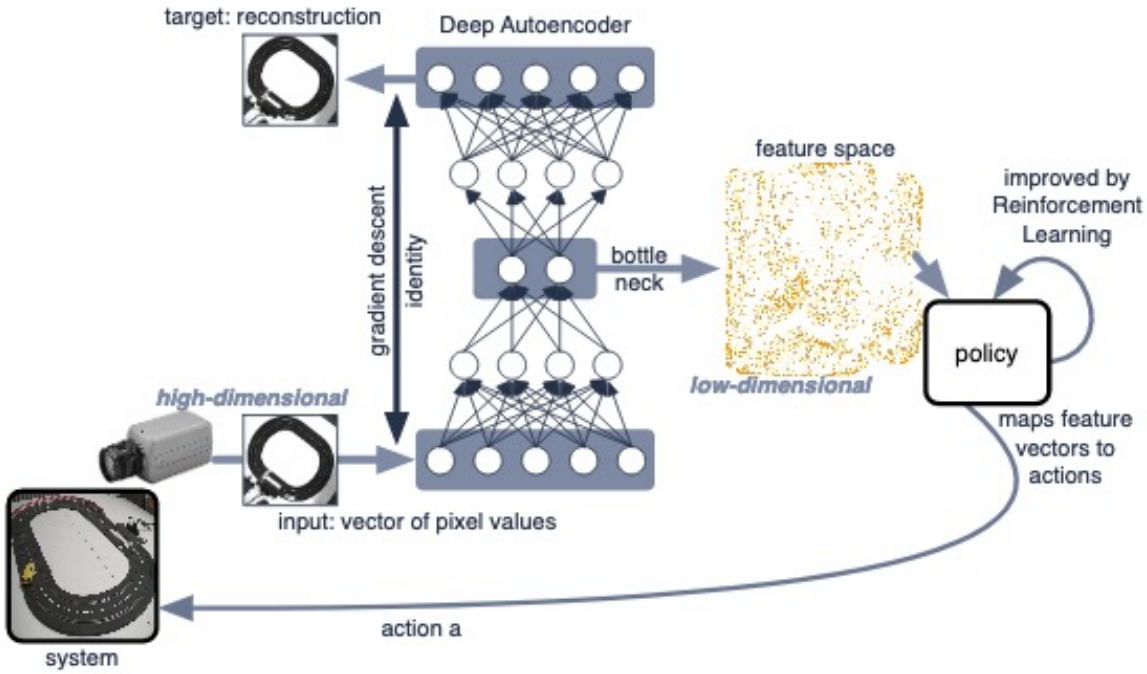
- *Note: not that common in real-world applications*



Ebert, Finn, et al. (2018); Finn & Levine (2017); Finn, Goodfellow, & Levine (2016)
https://bair.berkeley.edu/blog/2018/11/30/visual-rl/

Model inputs: Latent States

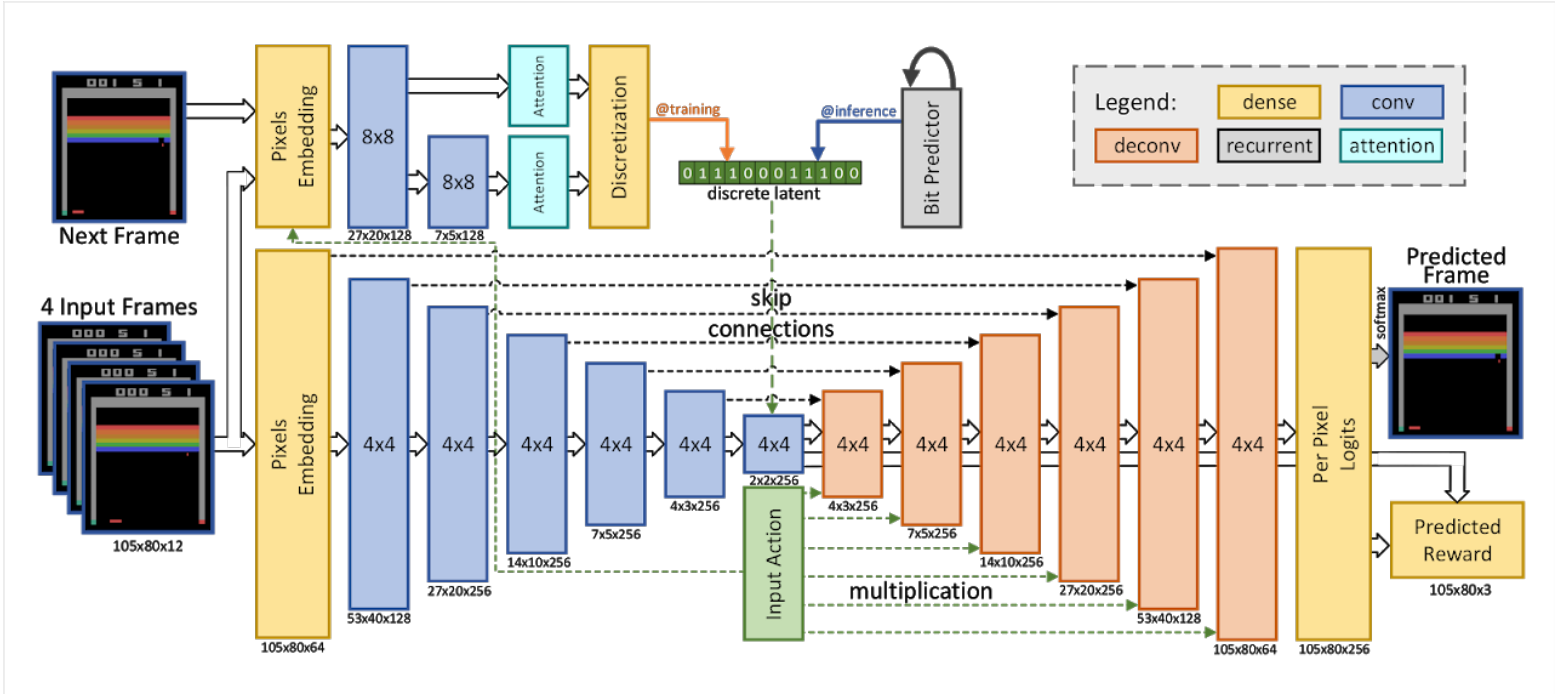
- Extract useful features from the raw observations and use them as inputs to the model
- *Note: current practice*



Lange et al.: Batch reinforcement learning. In Reinforcement learning (pp. 45-73). Springer, Berlin, Heidelberg. 2012.

Model inputs: Latent States

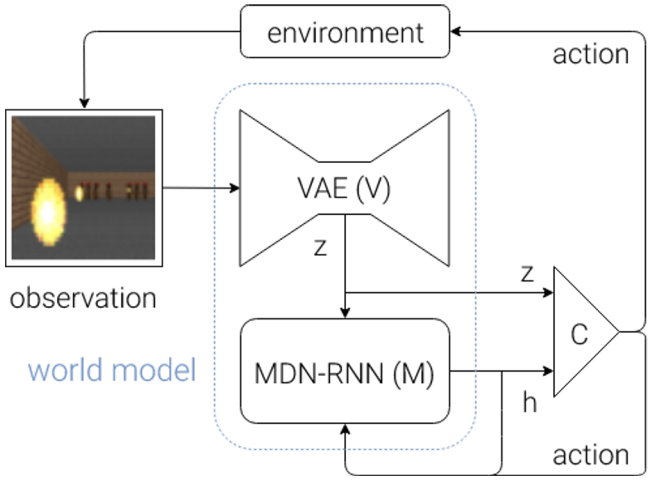
- Extract useful features from the raw observations and use them as inputs to the model
- *Note: current practice*

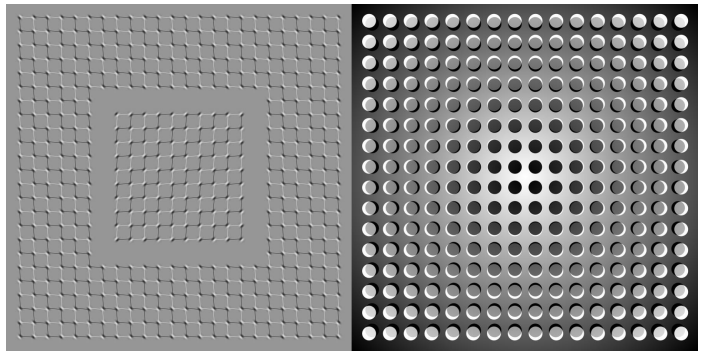


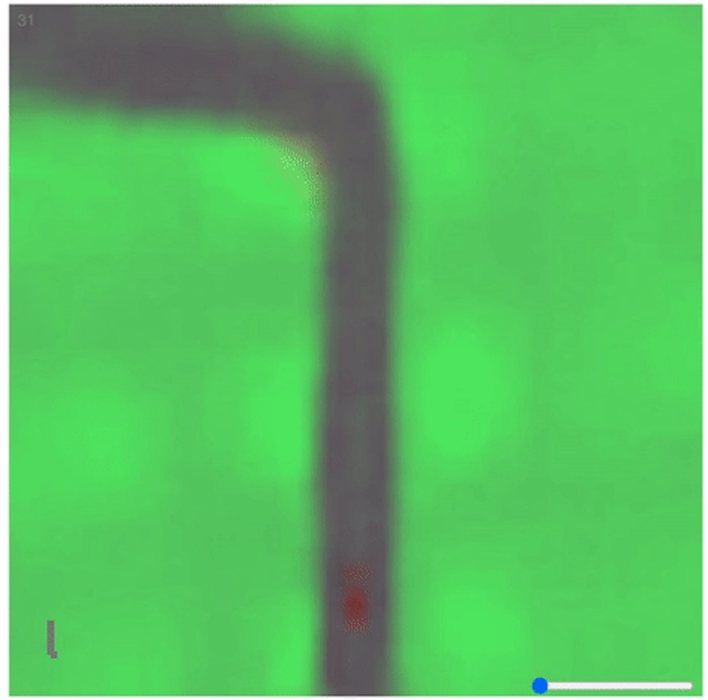
Kaiser et al.: Model-based reinforcement learning for atari. arXiv preprint arXiv:1903.00374. 2019.

Model inputs: Latent States

- Extract useful features from the raw observations and use them as inputs to the model
- *Note: current practice*







Ha & Schmidhuber: World Models. NeurIPS 2018.

See also: https://worldmodels.github.io

Which model should we use?

Model desiderata

- **Learning sample efficiency**
  - We would like to interact with the real system as little as possible (e.g., due to time and safety constraints, hardware wear and tear, etc.)
- **Multi-step accuracy**
  - We require our model to be able to accurately predict several time-steps in the future (remember one-step rewards vs discounted return)
- **Required engineering / domain knowledge**
  - How easy it is to design a “simulator” for the real system using basic physics? How accurate such model would be?
  - Does it make sense from economical perspective?
- **Prediction speed**
  - Can we deploy in real-time systems (e.g., drones)?

Which model should we use?

Model desiderata

Name	Features	Speed of learning	Speed of predictions	Domain knowledge	Long-term accuracy
Dynamical system	States	Fast	Fast	High	High
MLP	States	Med	Fast	Low	Med
Observation	Observations	Slow	Slow	Low	Low
State-space models	Latent States	Slow	Fast	Low	Med

https://sites.google.com/view/mbrl-tutorial