Deep reinforcement learning for controlling complex systems

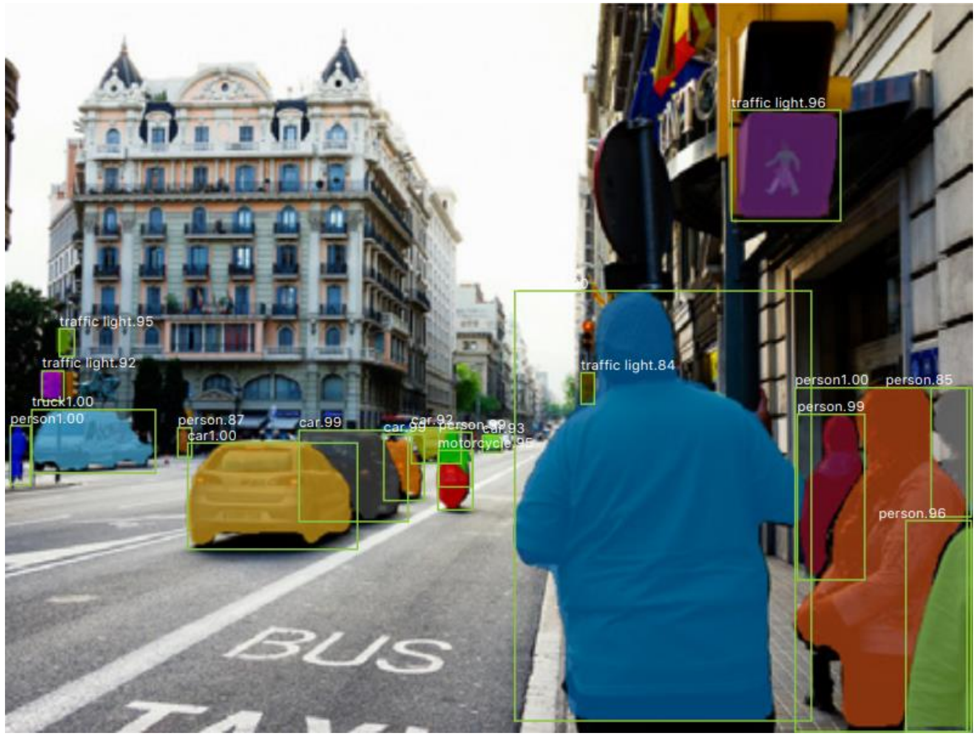
Alexander Kuhnle

University of Oslo, 7th December 2018

Why deep learning?

Successes of deep learning

Object recognition



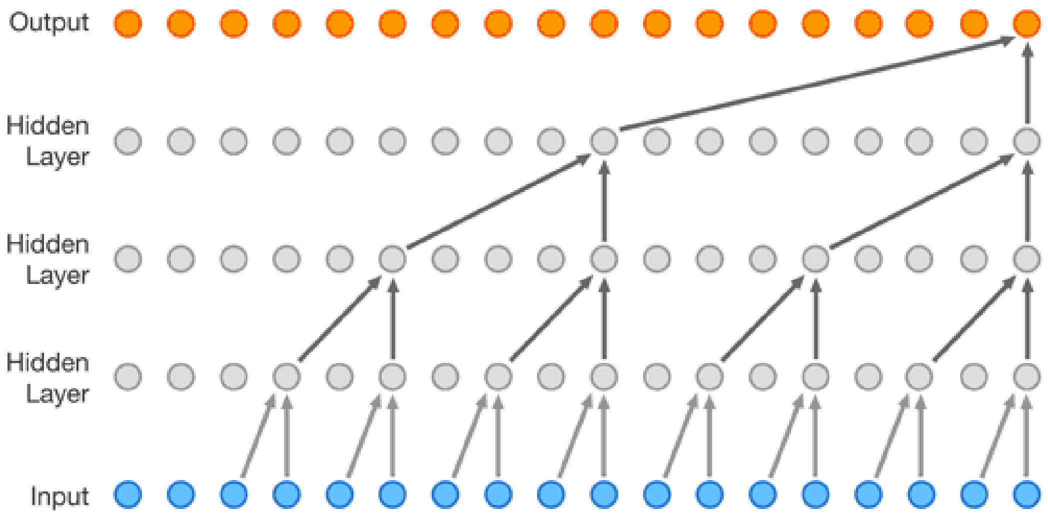
[Mask R-CNN (He et al., March 2017)]

Machine translation

[Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation (Johnson et al., November 2016)]

Successes of deep learning

Speech synthesis



[WaveNet: A Generative Model for Raw Audio (van den Oord et al., September 2016)]

Image synthesis

[Large Scale GAN Training for High Fidelity Natural Image Synthesis (Brock et al., September 2018)]

Where deep learning excels… and doesn't

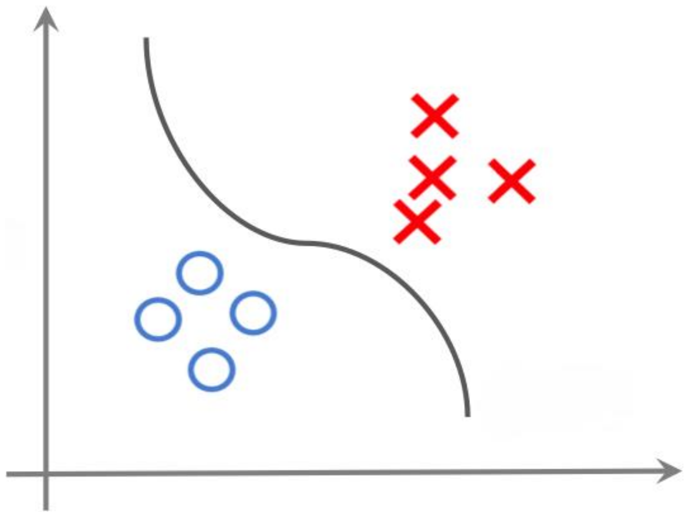
- ✓ Raw high-dimensional input
- ✓ Pattern recognition & matching
- ✓ Weak generalization (interpolation)
- ✓ Strong average performance
- ✓ Only prediction is important
- ✓ ”Trivial” but hard-to-explain tasks:
  - ✓ Visual processing: image, video
  - ✓ Language processing: spoken, text
  - ✓ Multimodal reasoning

- × Complex highly structured input
- × Abstract conceptualization
- × Strong generalization (extrapolation)
- × Reliable worst-case performance
- × Precise error bounds matter
- × Algorithmic and “artificial” tasks:
  - × NP-hard problems
  - × Strategic planning
  - × Explaining decisions

What is reinforcement learning?

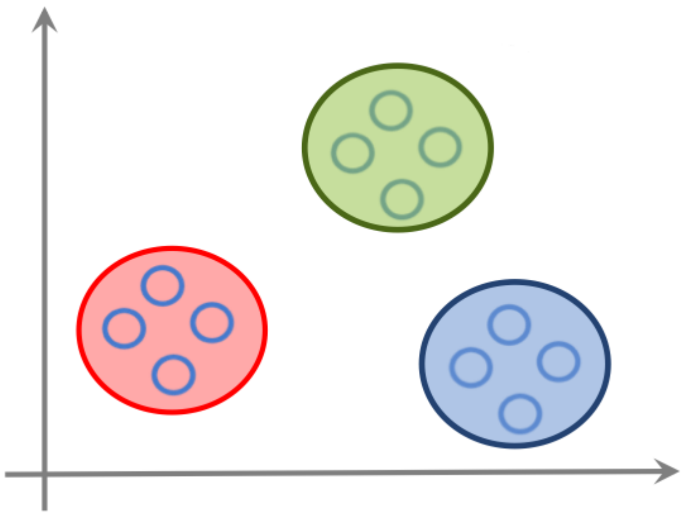
Three types of machine learning

Supervised learning



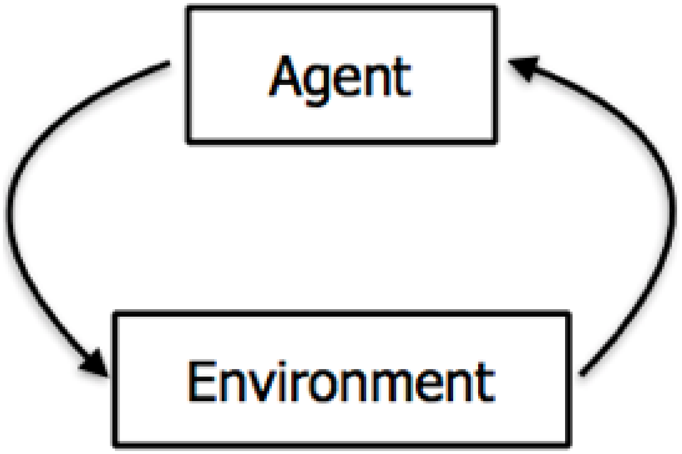
- Classification
- Regression

Unsupervised learning



- Clustering
- Representation learning

Reinforcement learning

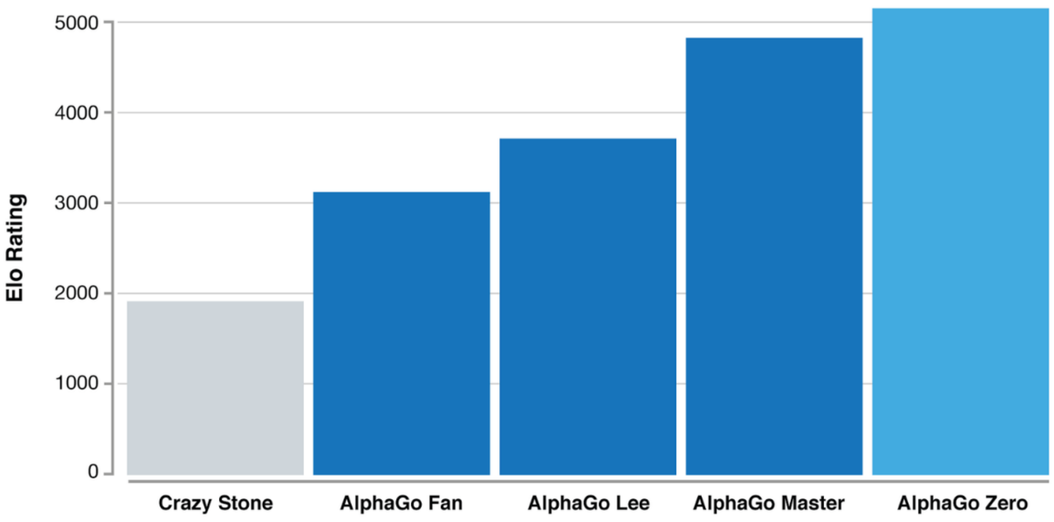


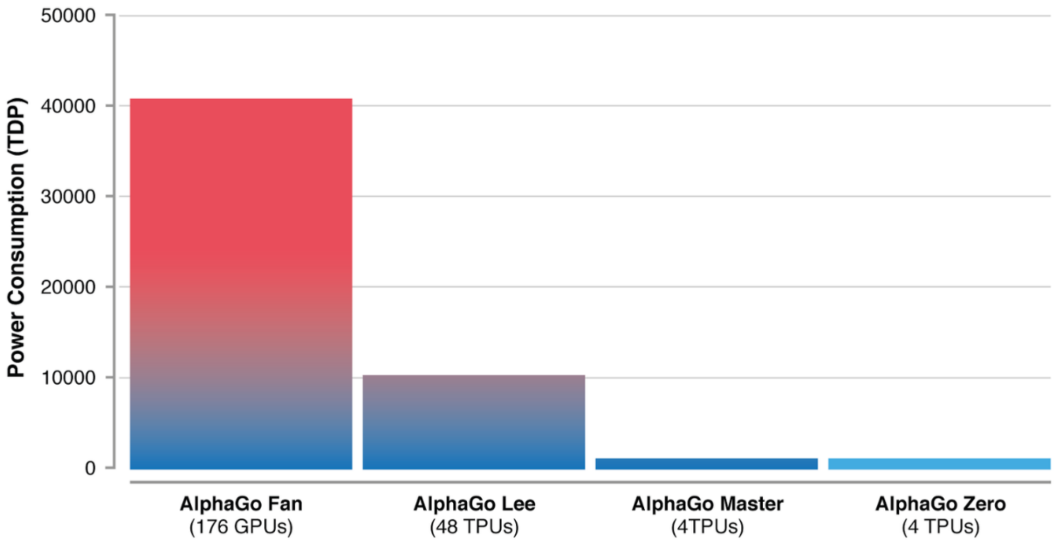
- Decision making
- Dynamic control

source: http://beta.cambridgespark.com/courses/jpm/figures/mod1_SuperUnsuper.png https://cdn-images-1.medium.com/max/1600/1*FlK1JS3vFhQasvuEgLU3Bg.png

DeepMind’s AlphaGo (Zero)

*“The game of Go has long been viewed as the **most challenging of classic games for artificial intelligence** owing to its enormous search space and the difficulty of evaluating board positions and moves.”*



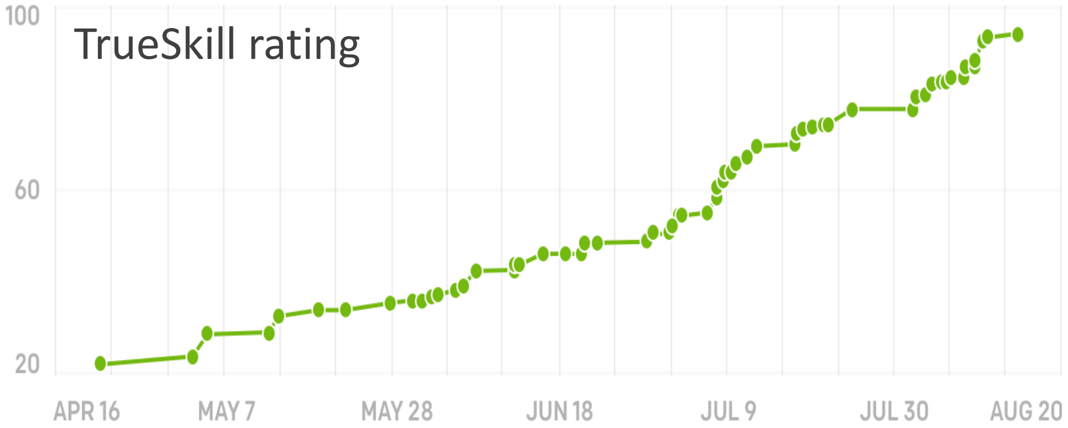


source: https://www.nature.com/articles/nature16961 https://deepmind.com/blog/alphago-zero-learning-scratch/

OpenAI Five versus Dota 2

Challenges:

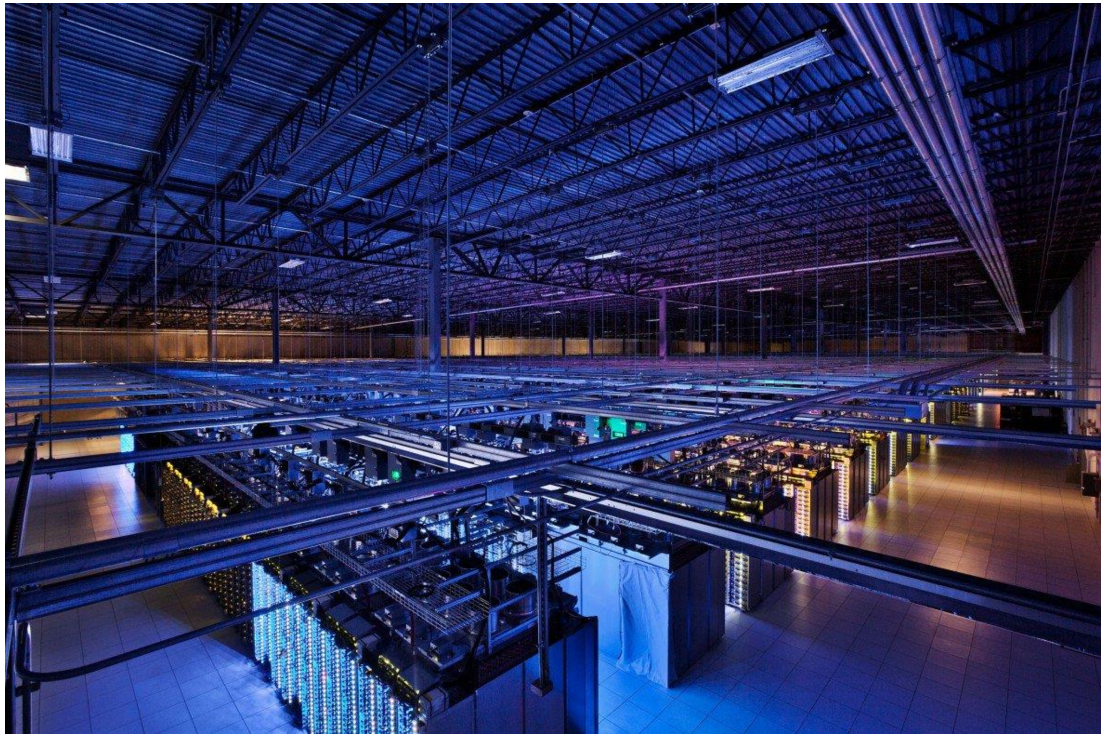
- Long time horizons
- Partially-observed state
- High-dimensional, continuous state/action space
- Dota rules are complex and constantly updated



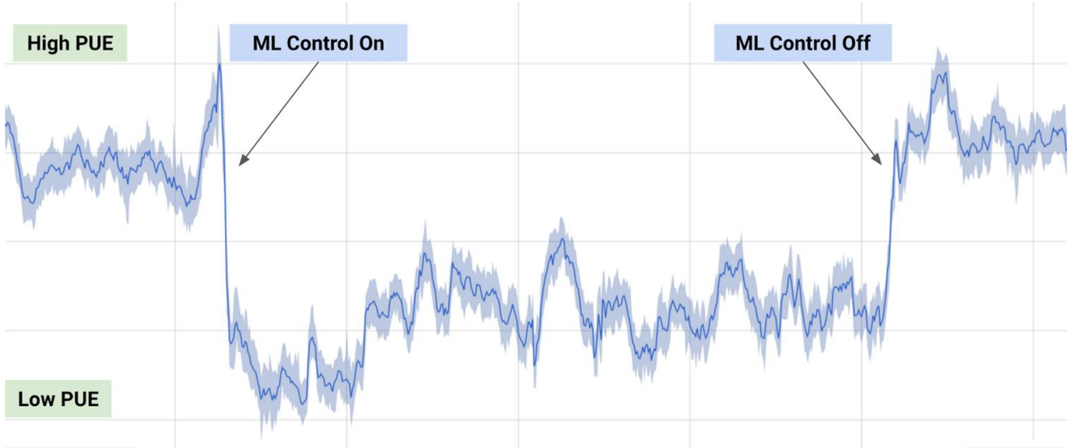
source: https://www.dota2.com/play/ https://blog.openai.com/more-on-dota-2/

DeepMind and Google’s data center cooling

“Google just gave control over data center cooling to an AI”



- Consistent energy reduction for cooling by 40%
- Corresponds to 15% reduction in overall PUE overhead



source: https://www.technologyreview.com/s/611902/google-just-gave-control-over-data-center-cooling-to-an-ai/ https://deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-40/

Real-world use cases

Optimize

- Process planning
- Job shop scheduling
- Yield management
- Supply chain
- Demand forecasting
- Warehouse operations optimization (picking)
- Production coordination
- Fleet logistics
- Product design
- Facilities location
- Camera Tuning
- Search ordering
- Agriculture
- Network optimization
- DDoS attack prevention
- Service availability

Control

- Robotics
- Wind Turbine Control
- HVAC
- Autonomous vehicles
- Factory automation
- Smart grids
- Machine Tuning

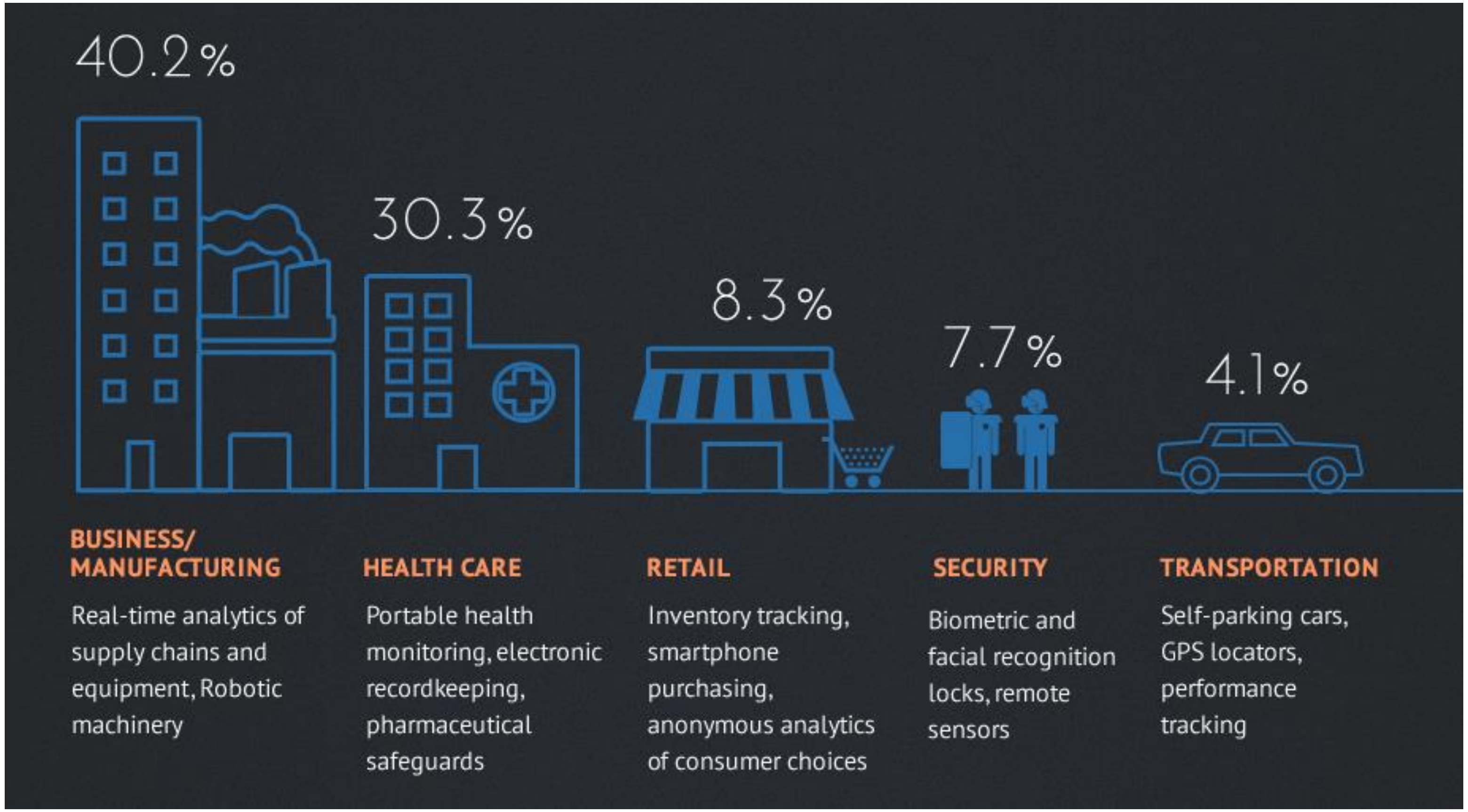
Monitor and Maintain

- Quality control
- Fault detection and isolation
- Predictive maintenance
- Inventory monitoring
- Supply chain risk management

- Robotics and manufacturing
- Resource management
- Power systems
- Computer clusters
- Finance
- Web content optimization
- Advertisement and bidding
- Deep learning

source: https://conferences.oreilly.com/artificial-intelligence/ai-ca-2017/public/schedule/detail/60500

Promising use case: internet of things



source: https://www.intel.com/content/dam/www/public/us/en/images/iot/guide-to-iot-infographic.png

Reinforcement learning in theory

The traditional framework

Timesteps: state s_t, action a_t, reward r_t (with discount γ)

Decision policy: $\pi : S \rightarrow A,\ \pi(s, a) = \mathrm{P}(a \mid s)$

State value: $\mathrm{V}^{\pi}(s) = \mathrm{E}[\sum_n \gamma^n \cdot r_n \mid s_0 = s,\ r_n \sim^{\text{rollout}} \pi]$

State-action value: $\mathrm{Q}^{\pi}(s, a) = \mathrm{E}[r_0 + \mathrm{V}^{\pi}(s_1) \mid s_0 = s,\ a_0 = a]$

Relation: $\mathrm{V}^{\pi}(s) = \mathrm{E}[\mathrm{Q}^{\pi}(s, a) \mid a \sim \pi(s)]$

Two classes of RL algorithms

Q-learning / value iteration

Learn $Q(s, a)$

$$\mathrm{Q}_t^{\mathrm{update}} = r_t + \gamma \cdot \max_a \mathrm{Q}(s_{t+1}, a)$$

Minimize $[\mathrm{Q}(s_t, a_t) - \mathrm{Q}_t^{\mathrm{update}}]^2$

Policy gradient methods

Learn $\pi(s, a)$

$$\mathrm{V}_t^{\mathrm{update}} = \mathrm{Q}_t^{\mathrm{update}} = \sum_{n=t} \gamma^n \cdot r_n$$

$$\nabla R = \mathrm{E}[\, \nabla \log \pi(s_t, a_t) \cdot \mathrm{V}_t^{\mathrm{update}} \mid \pi \,]$$

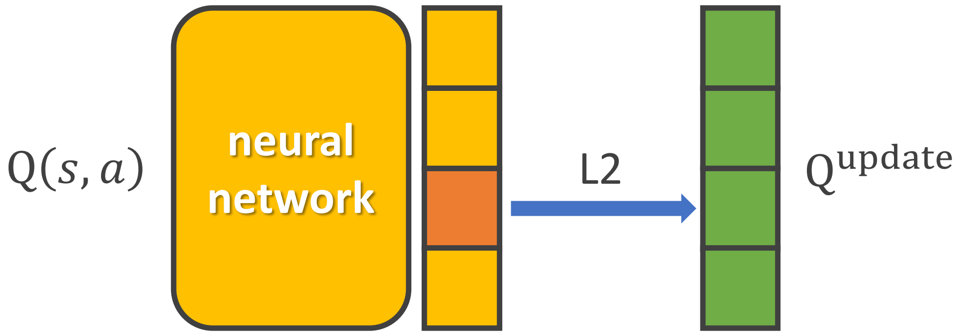
Minimize $-\log \pi(s_t, a_t) \cdot \mathrm{V}_t^{\mathrm{update}}$

Deep RL: optimization in detail

Q-learning / value iteration

$$Q_t^{\text{update}} = r_t + \gamma \cdot \max_a Q(s_{t+1}, a)$$

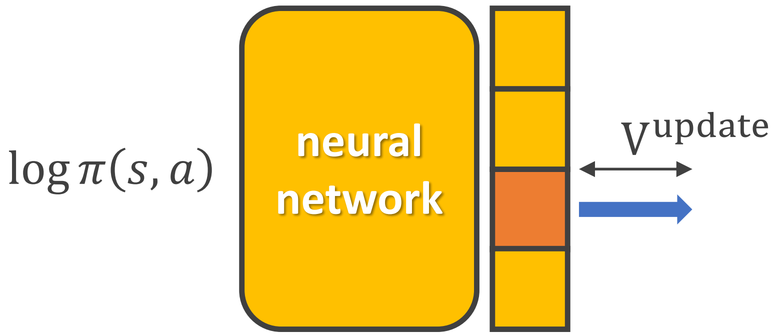
$$\text{Minimize} \quad [Q(s_t, a_t) - Q_t^{\text{update}}]^2$$



Policy gradient methods

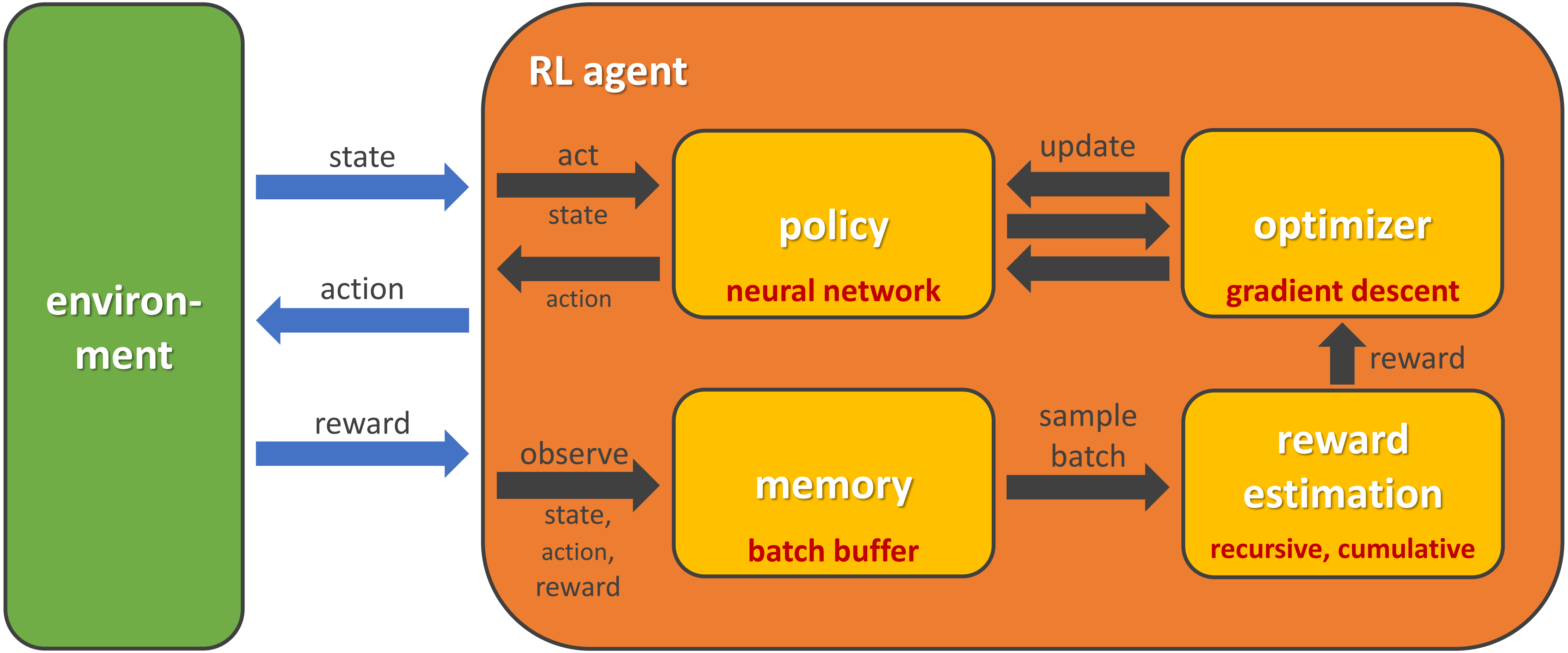
$$V_t^{\text{update}} = Q_t^{\text{update}} = \sum_{n=t} \gamma^n \cdot r_n$$

$$\text{Minimize} \quad -\log \pi(s_t, a_t) \cdot V_t^{\text{update}}$$

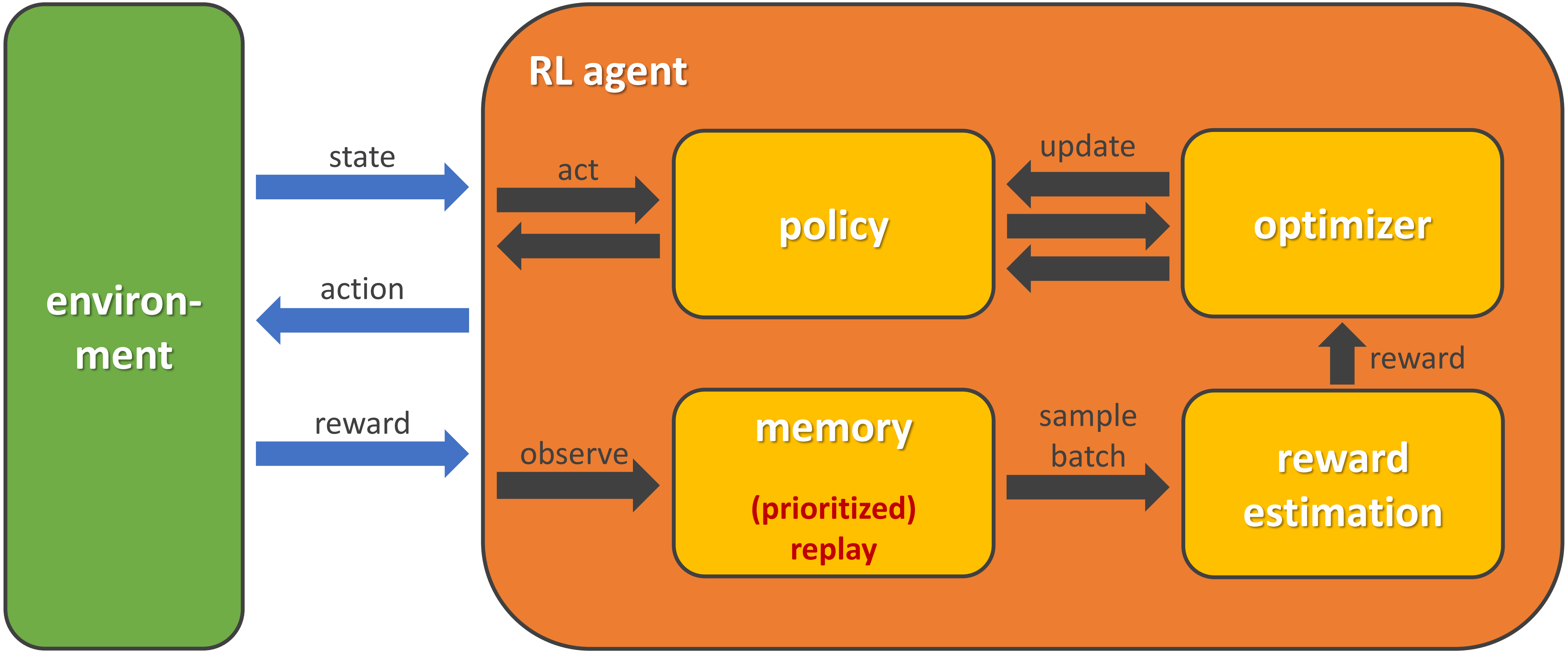


Reinforcement learning in practice

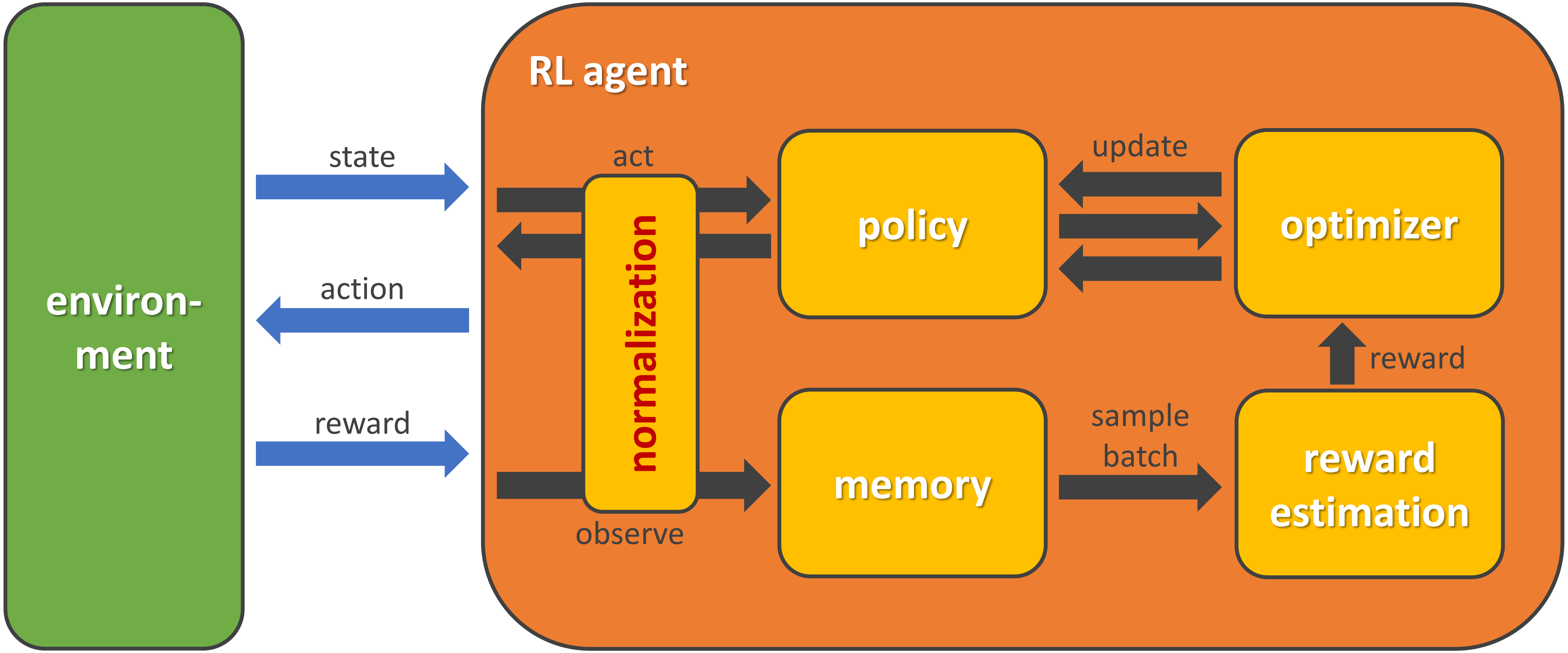
Typical deep RL implementation



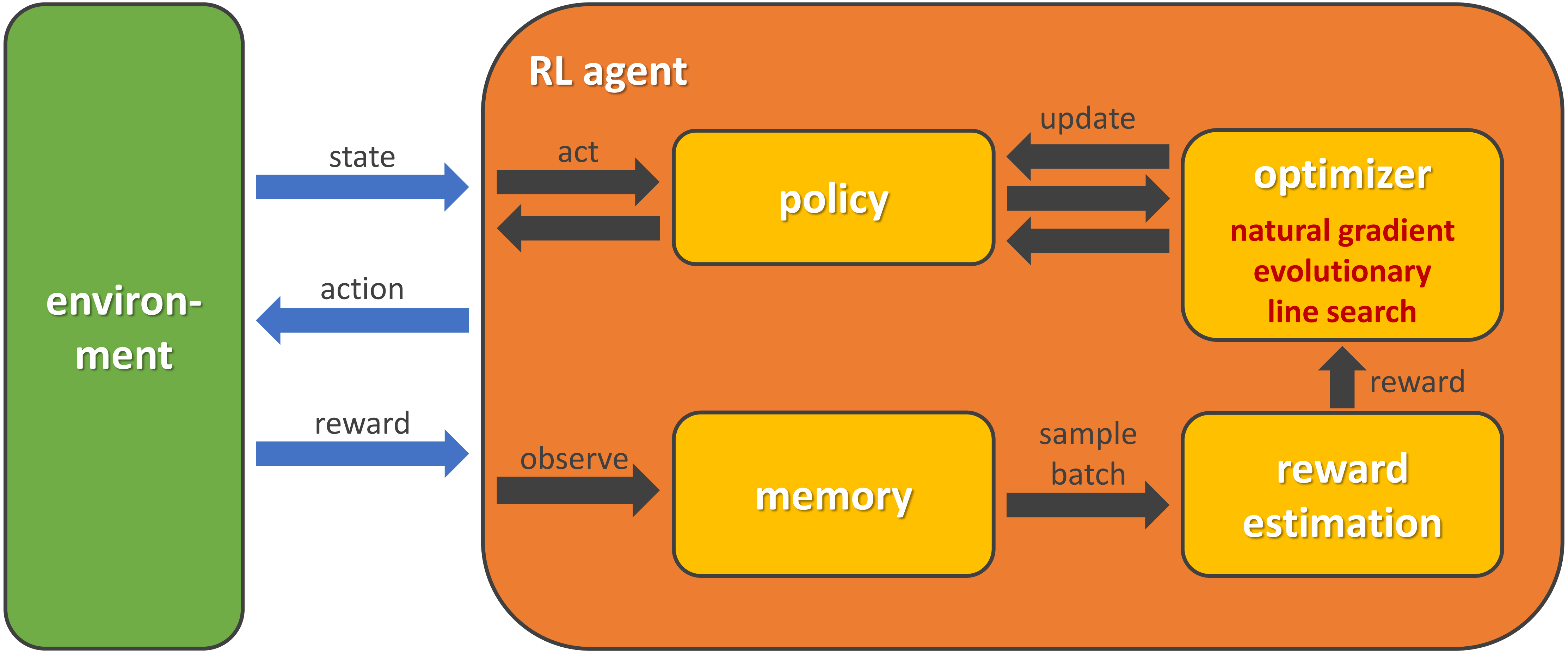
Improvement: replay memory



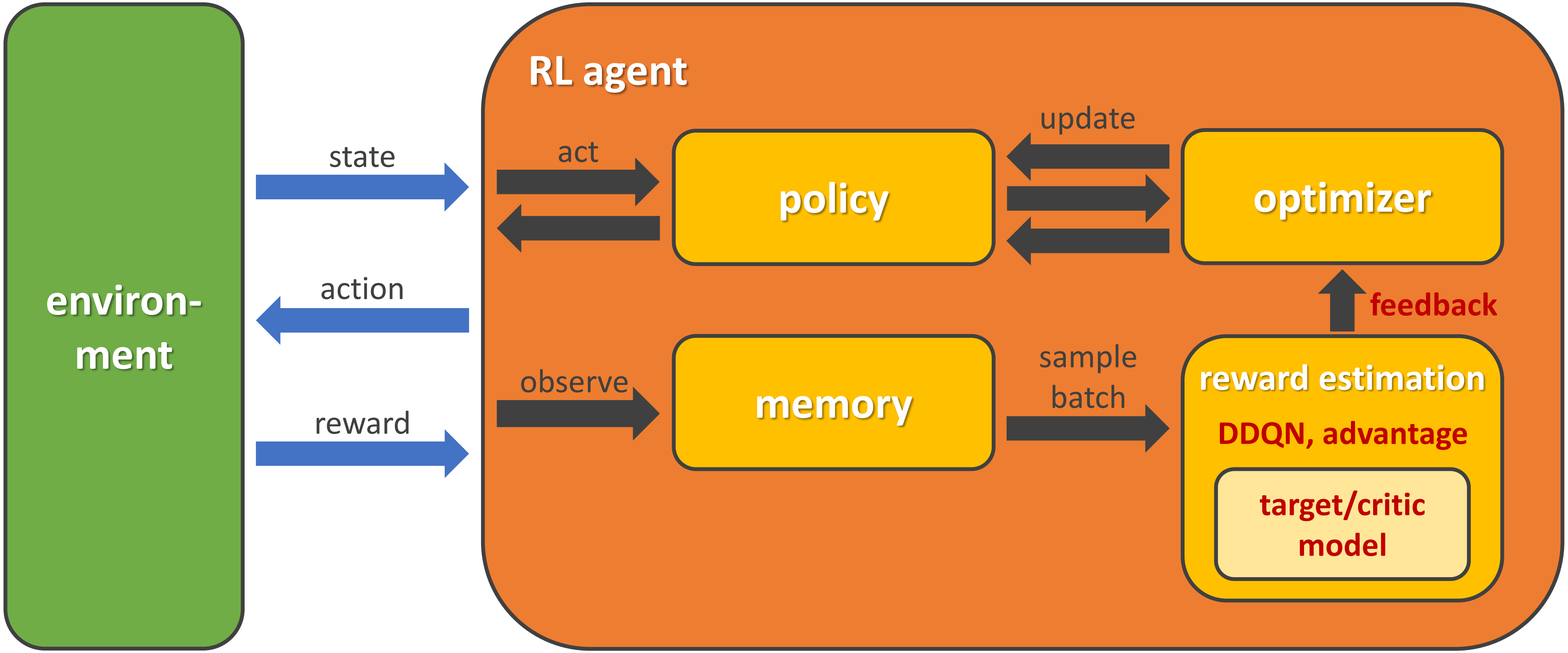
Improvement: normalization



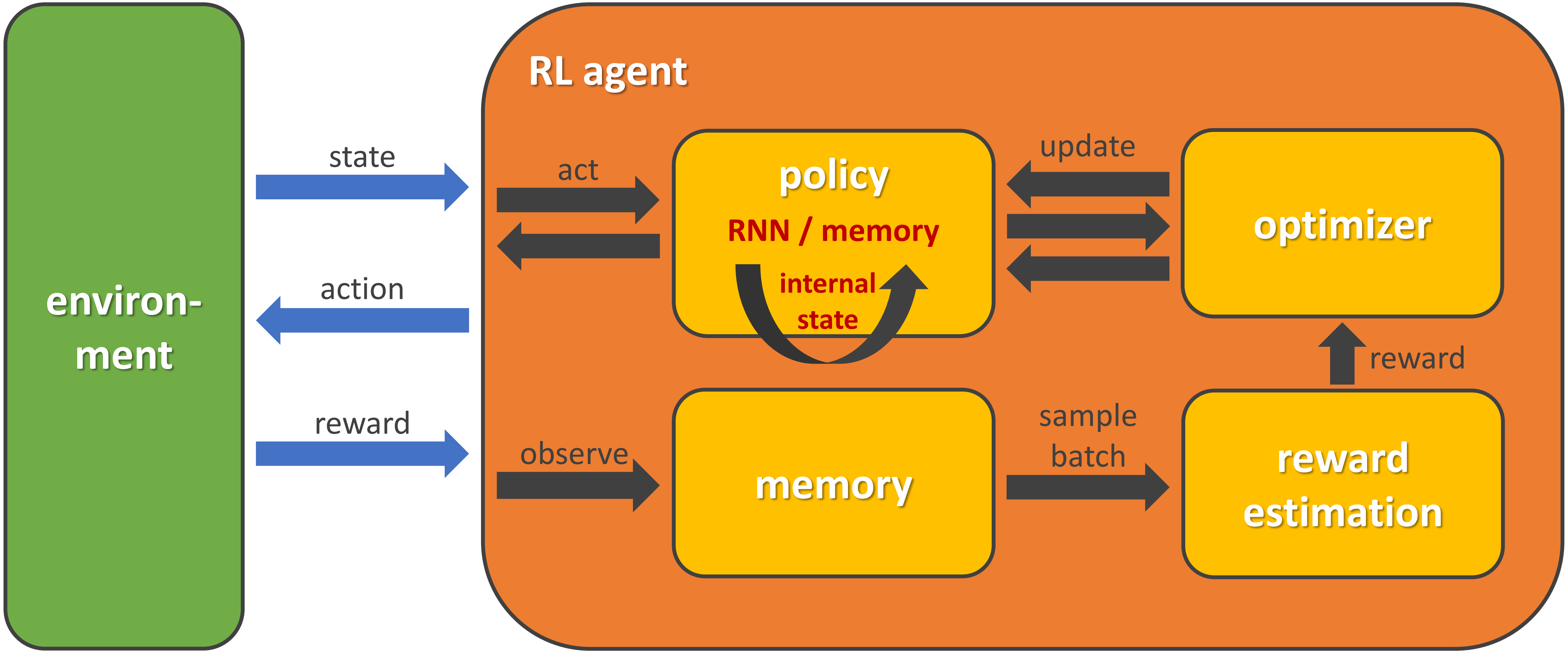
Improvement: optimization



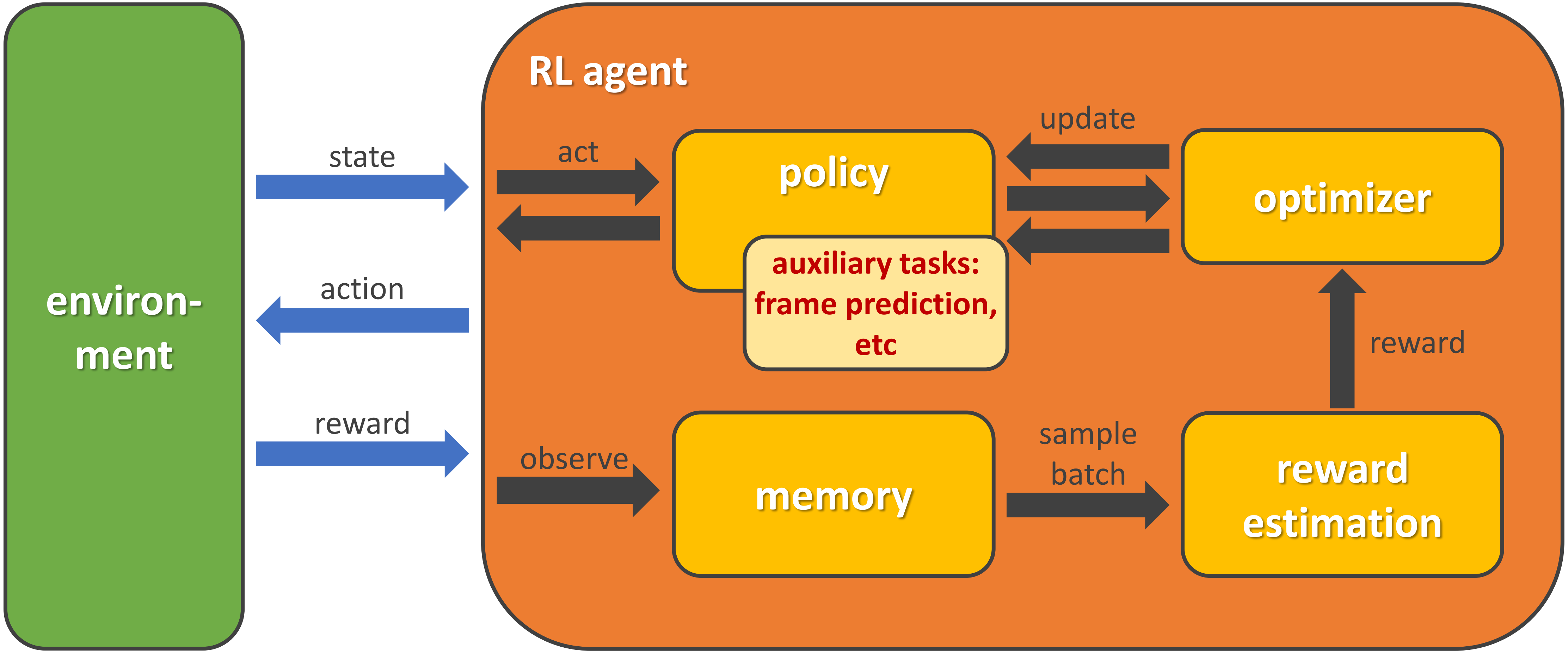
Improvement: value estimator module



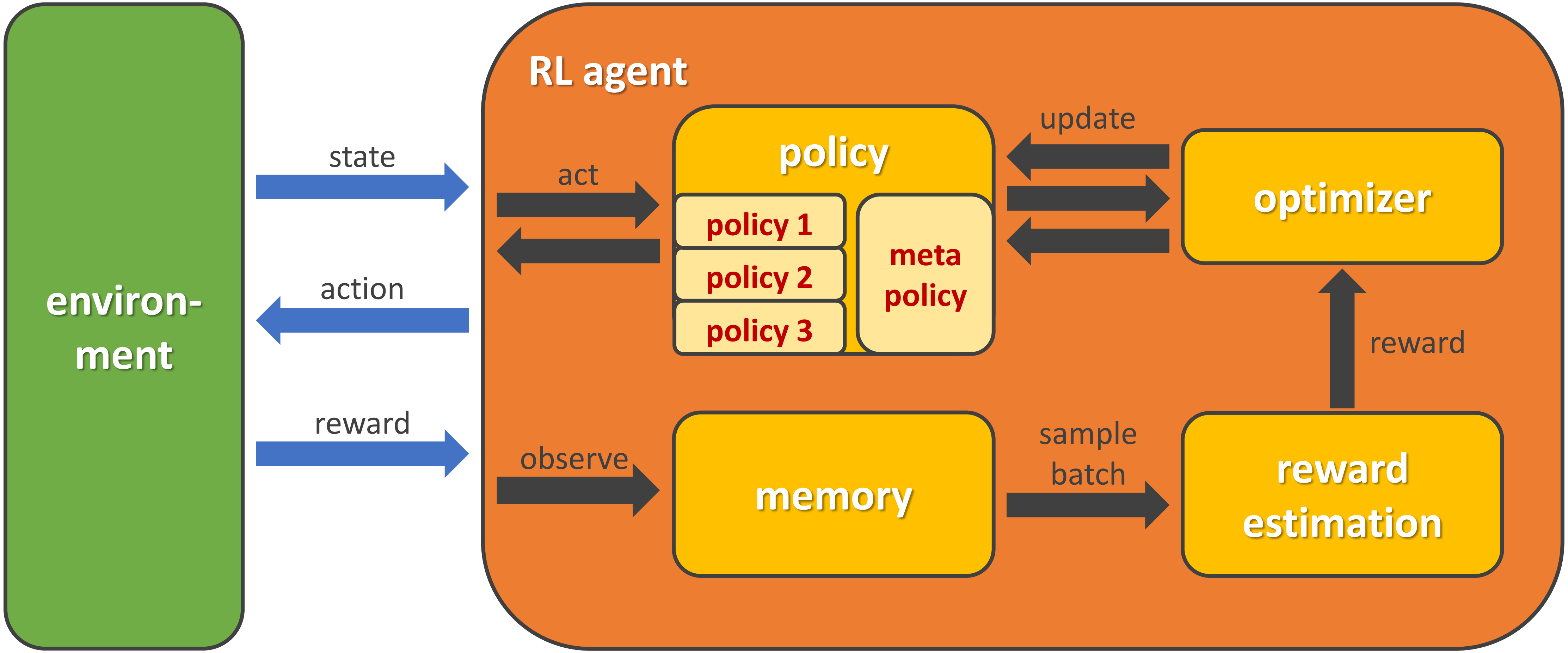
Improvement: internal state/memory



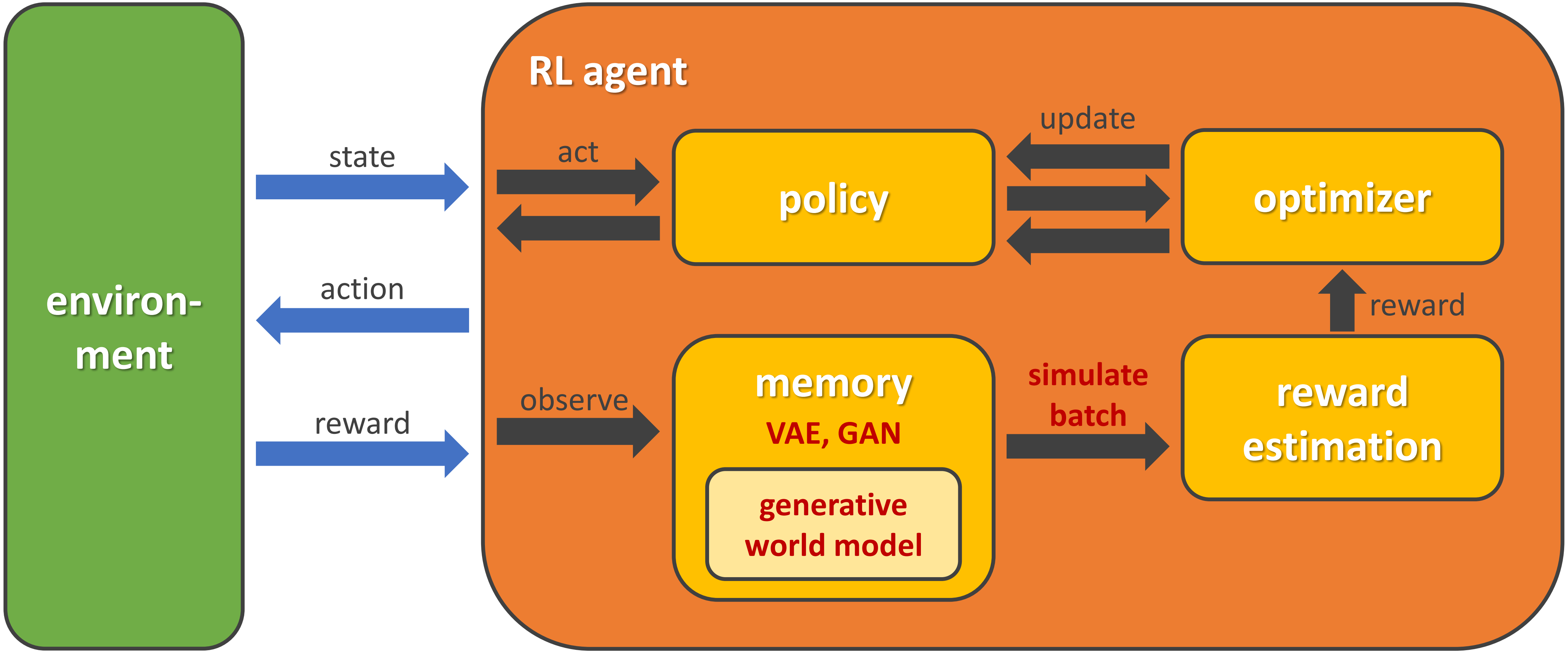
Improvement: auxiliary tasks



Improvement: hierarchical policies

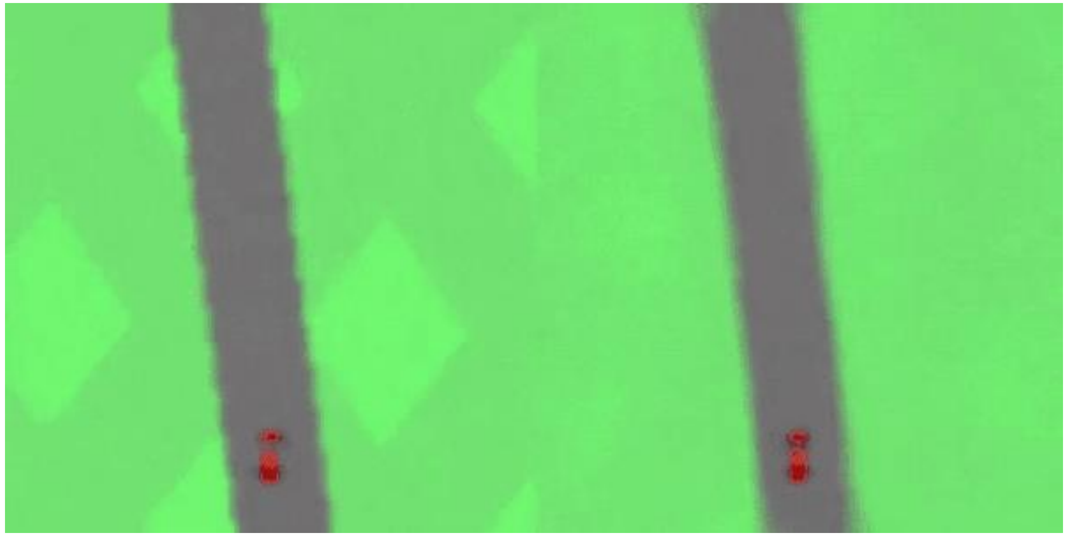


Improvement: generative memory

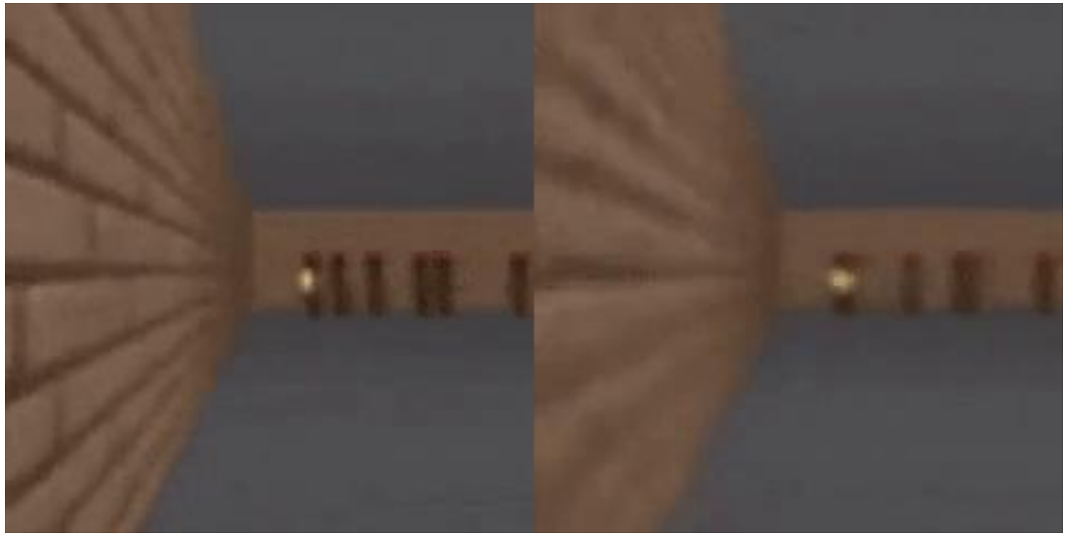


World models (Ha & Schmidhuber, 2018)

"Can agents learn inside of their own dreams?"



CarRacing-v0 environment



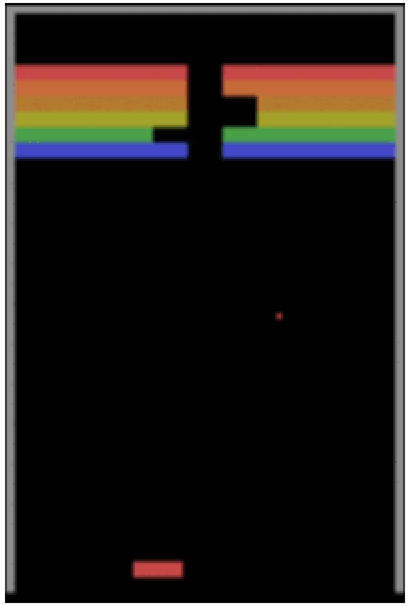
VizDoom environment

source: https://worldmodels.github.io/

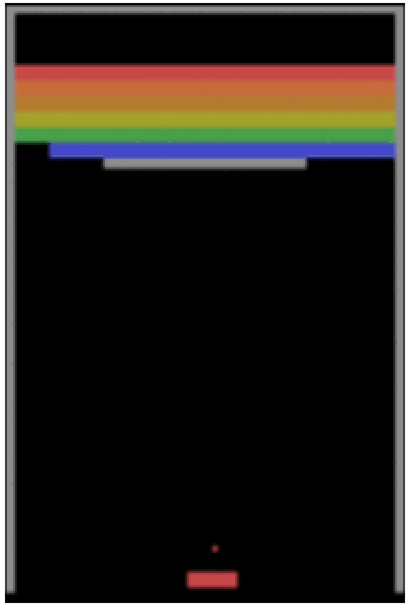
Failures of deep (reinforcement) learning

Failure: overfitting to environment details

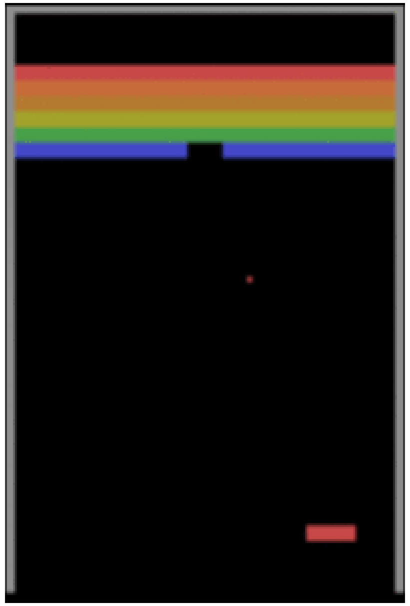
Original Breakout



Breakout + middle wall



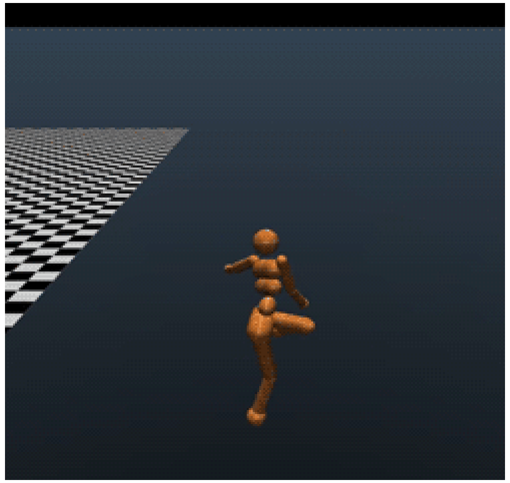
Breakout + offset paddle

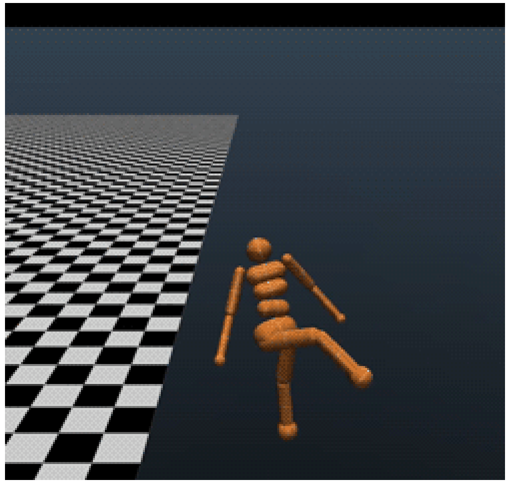


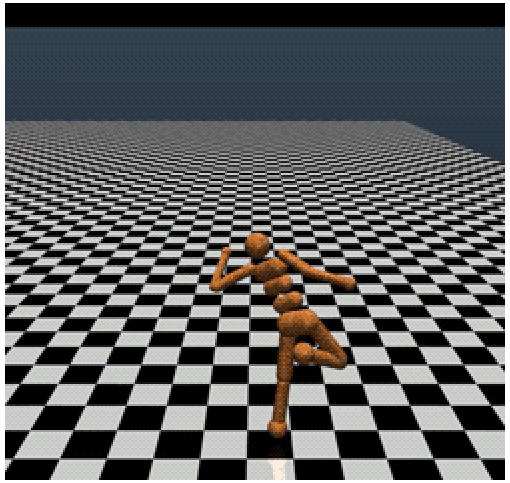
source: https://www.vicarious.com/2017/08/07/general-game-playing-with-schema-networks/

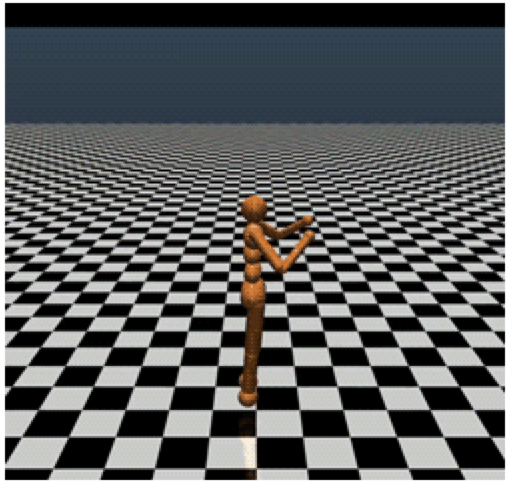
Failure: unrealistic simulation

Model behavior passing the 6000 reward threshold:









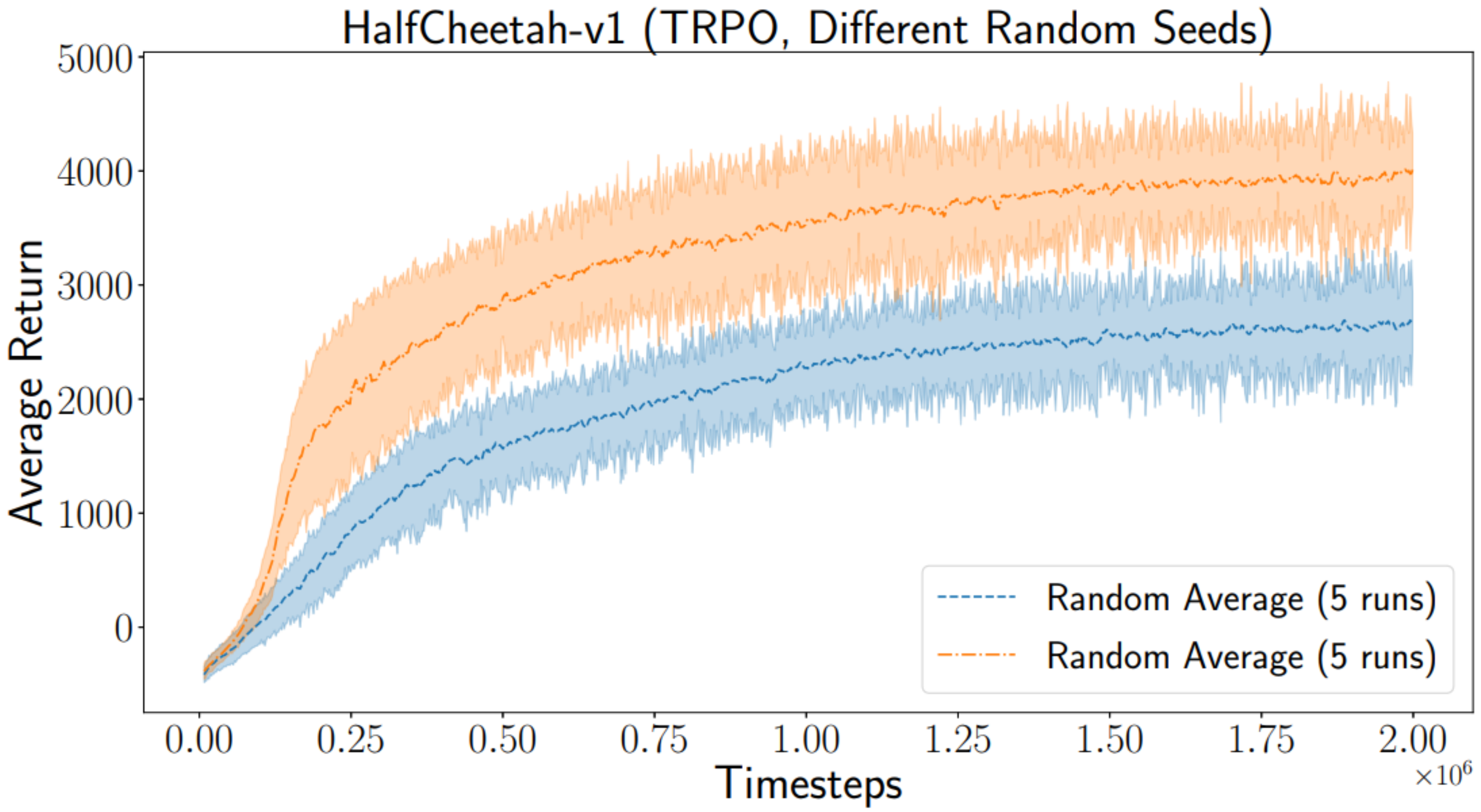
top performing
(reward 11,600)

source: http://www.argmin.net/2018/03/20/mujocoloco/

Failure: problematic reward function

source: https://medium.com/@deepmindsafetyresearch/building-safe-artificial-intelligence-52f5f75058f1

Failure: randomness



source: https://arxiv.org/abs/1709.06560

Conclusion

Reinforcement learning: what’s missing?

✓ Raw high-dimensional input

✓ Pattern recognition & matching

✓ Weak generalization (interpolation)

✓ Strong average performance

✓ Only prediction is important

✓ ”Trivial” but hard-to-explain tasks:

  ✓ Visual processing: image, video
  - Language processing: spoken, text
  ✓ Multimodal reasoning

- Complex highly structured input
- Abstract conceptualization

? Strong generalization (extrapolation)

! Reliable worst-case performance

? Precise error bounds matter

- Algorithmic and “artificial” tasks:
  - NP-hard problems
  - ? Strategic planning
  - Explaining decisions

Reinforcement learning: what’s missing?

- ✓ Raw high-dimensional input
- ✓ Pattern recognition & matching
- ✓ Weak generalization (interpolation)
- ✓ Strong average performance
- ✓ Only prediction is important
- ✓ ”Trivial” but hard-to-explain tasks:
  - ✓ Visual processing: image, video
  - • Language processing: spoken, text
  - ✓ Multimodal reasoning

- • Complex highly structured input
- • Abstract conceptualization
- ? Strong generalization (extrapolation)
- ! Reliable worst-case performance
- ? Precise error bounds matter
- • Algorithmic and “artificial” tasks:
  - • NP-hard problems
  - ? Strategic planning
  - • Explaining decisions

! Ease of application: plug-and-play reinforcement learning

→ Tensorforce: A TensorFlow library for applied reinforcement learning

Thanks for your attention!

Questions?