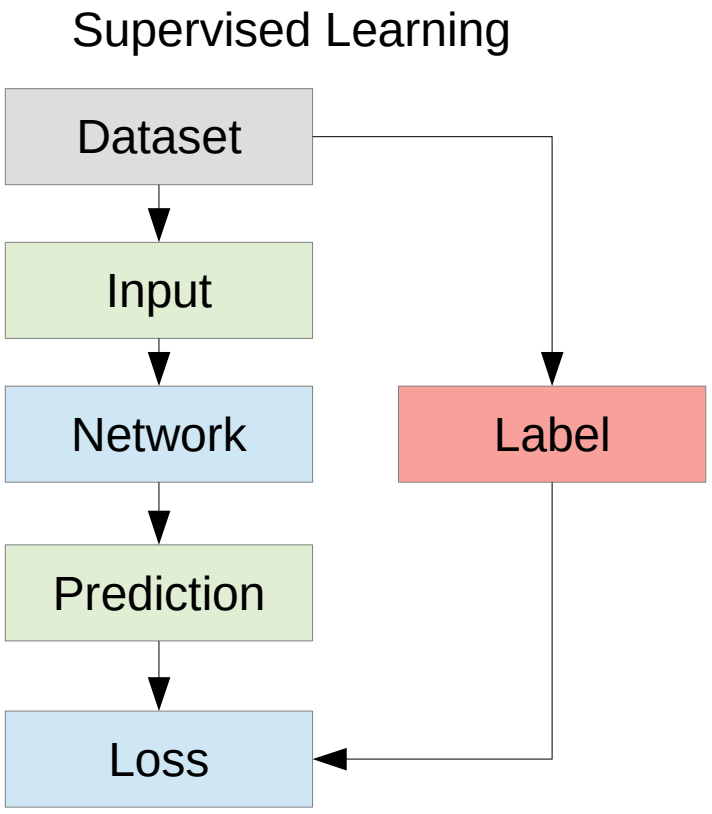
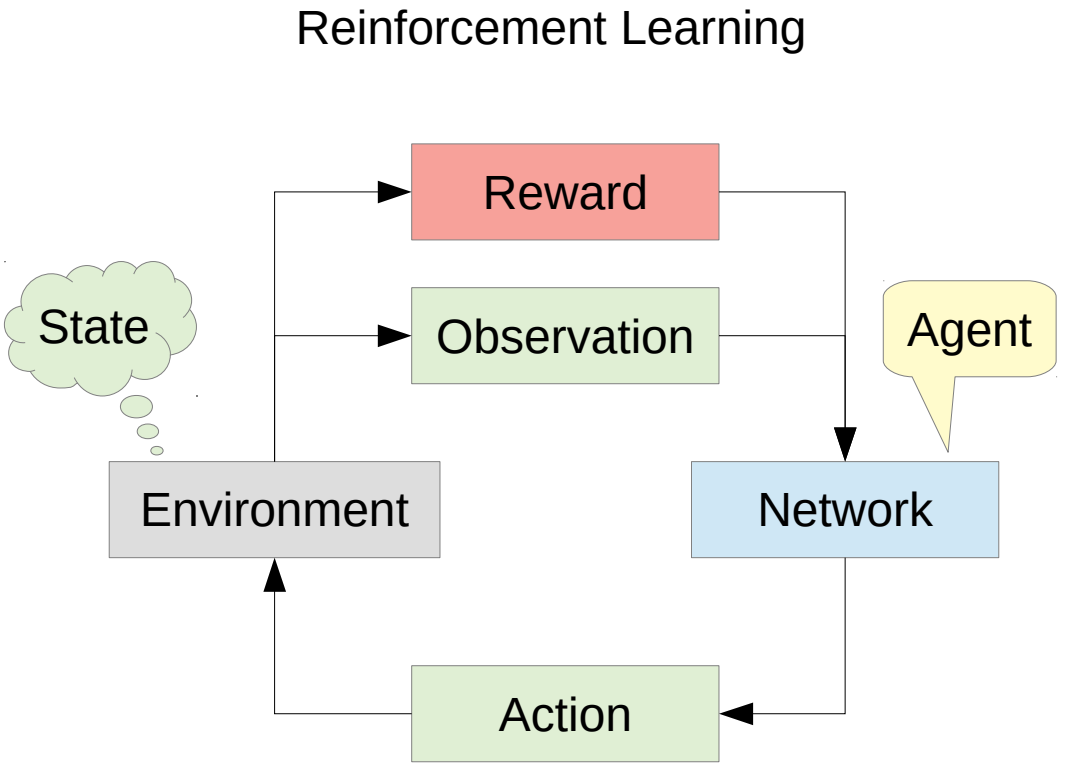
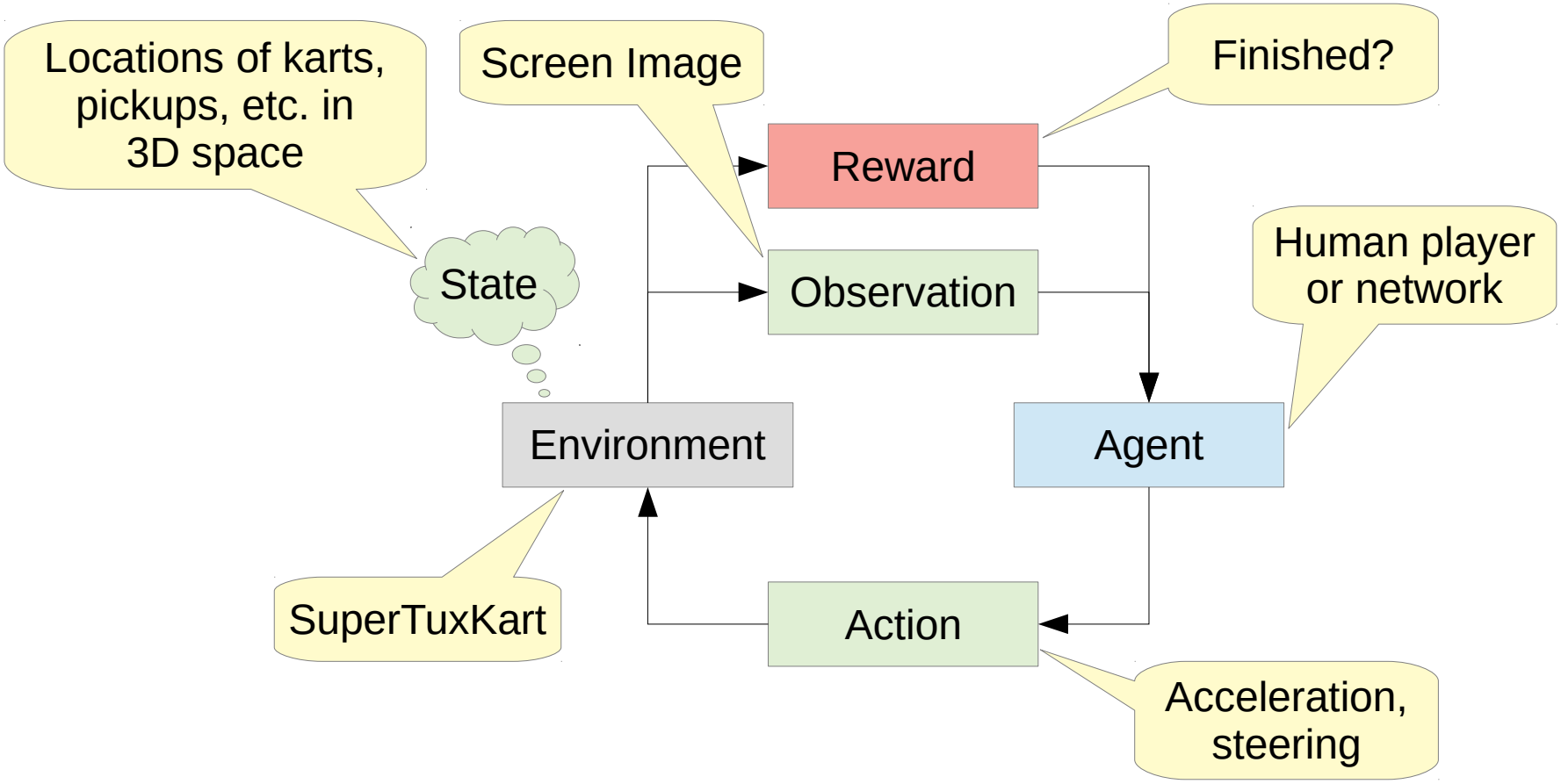
Reinforcement Learning

Acting in an Environment

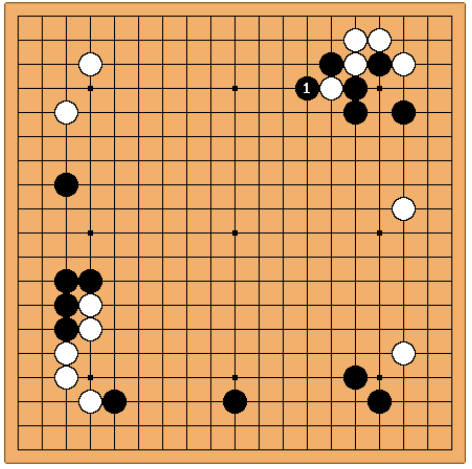


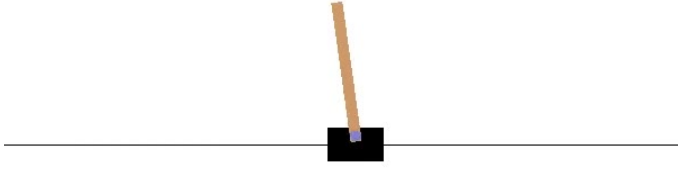


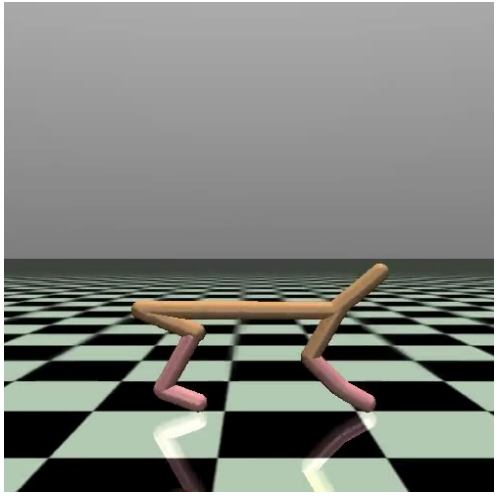
Example: SuperTuxKart



More Examples







Formally: (Partially Observable) Markov Decision Process

Partially Observable

Observation space

$o \in \Omega$

Observation probabilities

$O(o_{i+1} | s_{i+1}, a_i)$

Markov Decision Process

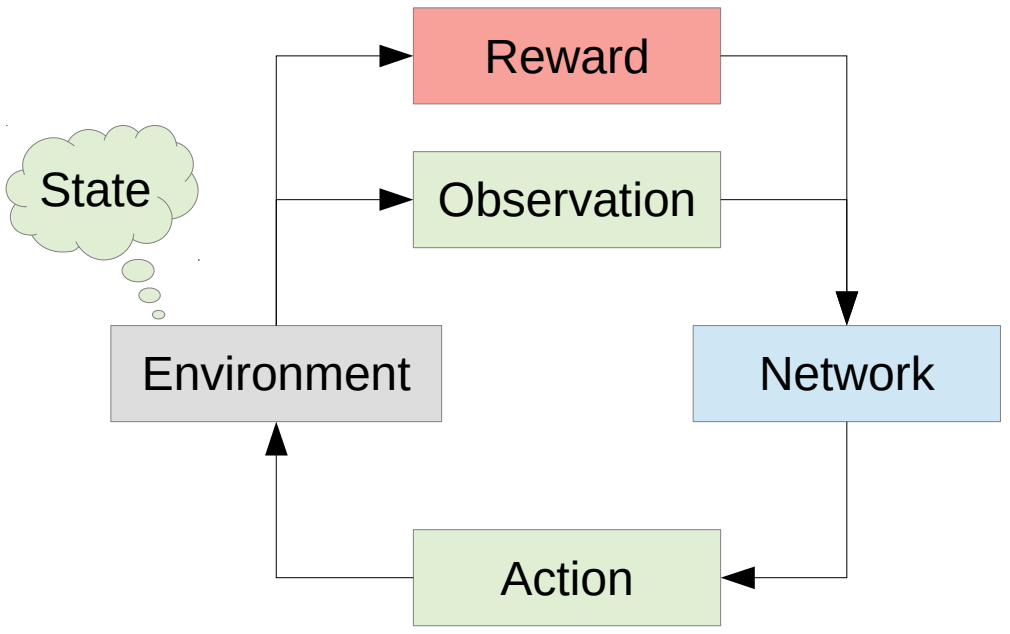
$s \in S$ — State space

$a \in A$ — Action space

Transition probabilities

$P(s_{i+1} | s_i, a_i)$

$r(s_i, a_i)$ — Reward



Goal

Optimize rewards *in the long term*

Rollout

Trajectory $\tau = s_0, a_0, s_1, a_1, \ldots, s_n, a_n$

Return $R(\tau) = \sum_{i=0}^{n} \gamma^i r(s_i, a_i)$ "Discounted sum"

Discount factor

Challenges



Network
Network
Network

Environment
Environment
Environment

Not differentiable!

Reward
Reward
Reward

Discounted Sum

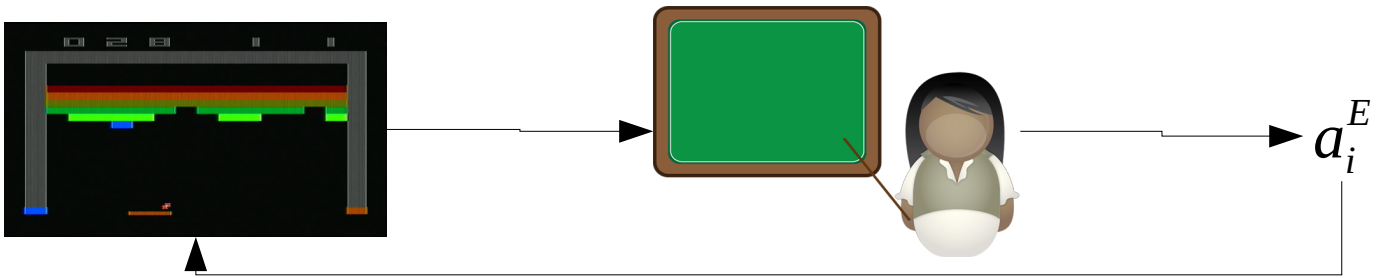
Return

Long-term dependencies

Sparse rewards

Imitation Learning

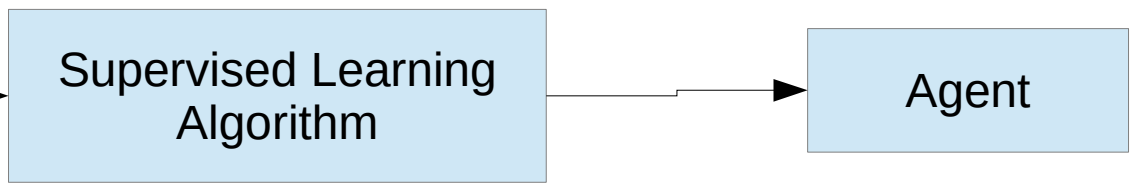
- Gather example trajectories from an expert



- Use supervised learning

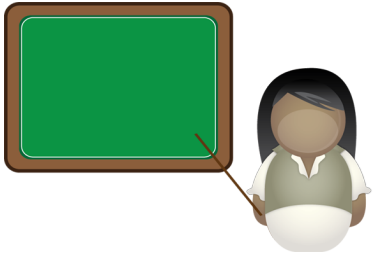
$$\tau^E = s_{0,} a_0^E, s_{1,} a_1^E, \ldots, s_n, a_n^E$$

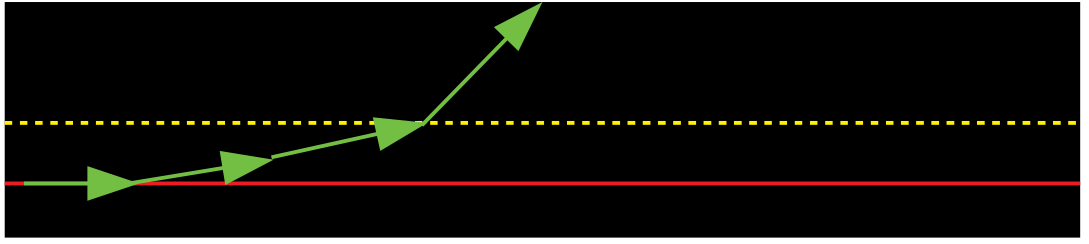
$s_0 \rightarrow a_0^E$
$s_1 \rightarrow a_1^E$
$\vdots$
$s_n \rightarrow a_n^E$



Imitation Learning: Problems

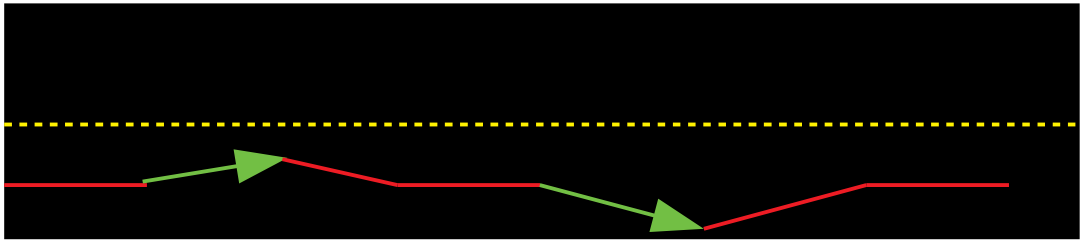
- Expert trajectories can be hard to gather
- Expert limits performance
- Distribution shift



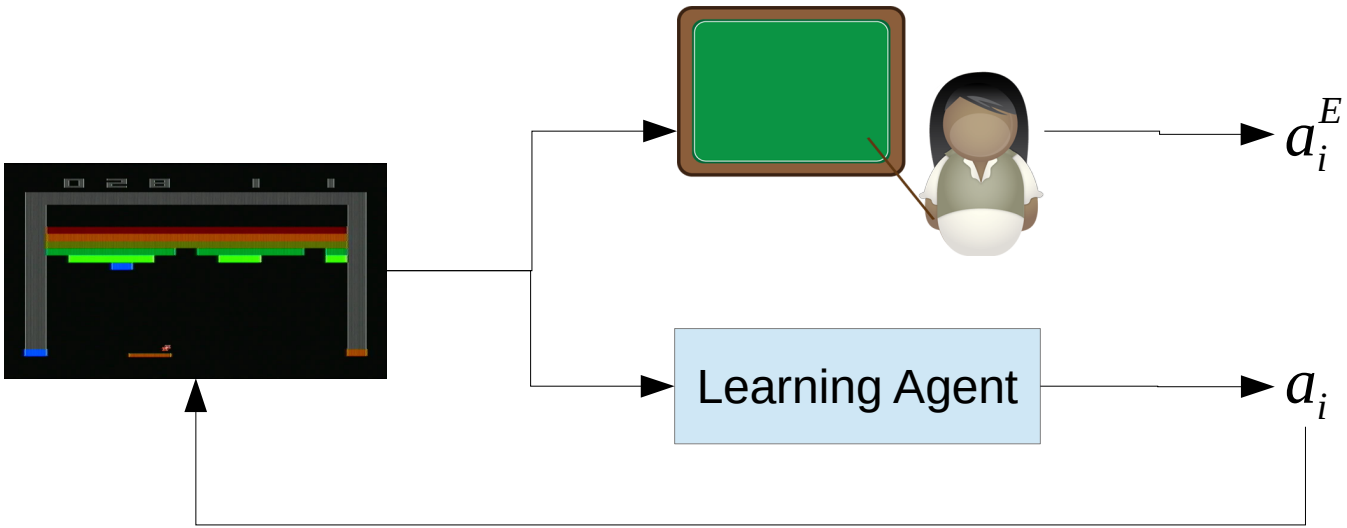


Imitation Learning Tips and Tricks

- Pre-training (for image inputs)
- Data Augmentation



DAgger



“On-policy”