CS 690: Human-Centric Machine Learning

Prof. Scott Niekum

Bayesian inverse reinforcement learning and safe IRL

What does it mean for a learning agent to be "safe"?

- **Formal safety:** A self-driving car that will provably never crash if some model holds
- **Risk-sensitive safety:** A stock market agent with bounded value-at-risk
- **Robust safety:** An image classifier resistant to data poisoning or adversarial examples
- **Monotonic safety:** An RL-based advertising policy that always improves with high probability
- **Safe exploration:** A walking robot that can explore new gaits without falling over

More complete taxonomy: D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. "Concrete problems in AI safety."

A proposed definition of safety:

Safety = Correctness + Confidence

Correctness: Meeting or exceeding a measure of performance

Confidence: A (probabilistic) guarantee of correctness

A spectrum of safety for policy learning

Guaranteed — Probabilistic — Heuristic

Require perfect models

Verification / synthesis

[Kress-Gazit et. al 2009]
[Raman et. al 2015]

Sample inefficient

PAC-MDP methods

[Singh et. al 2002]
[Fu and Topcu 2014]

Concentration inequalities

[Thomas et. al 2015]
[Bottou et. al 2013]
[Abbeel and Ng 2004]
[Syed and Schapire 2008]

No guarantees

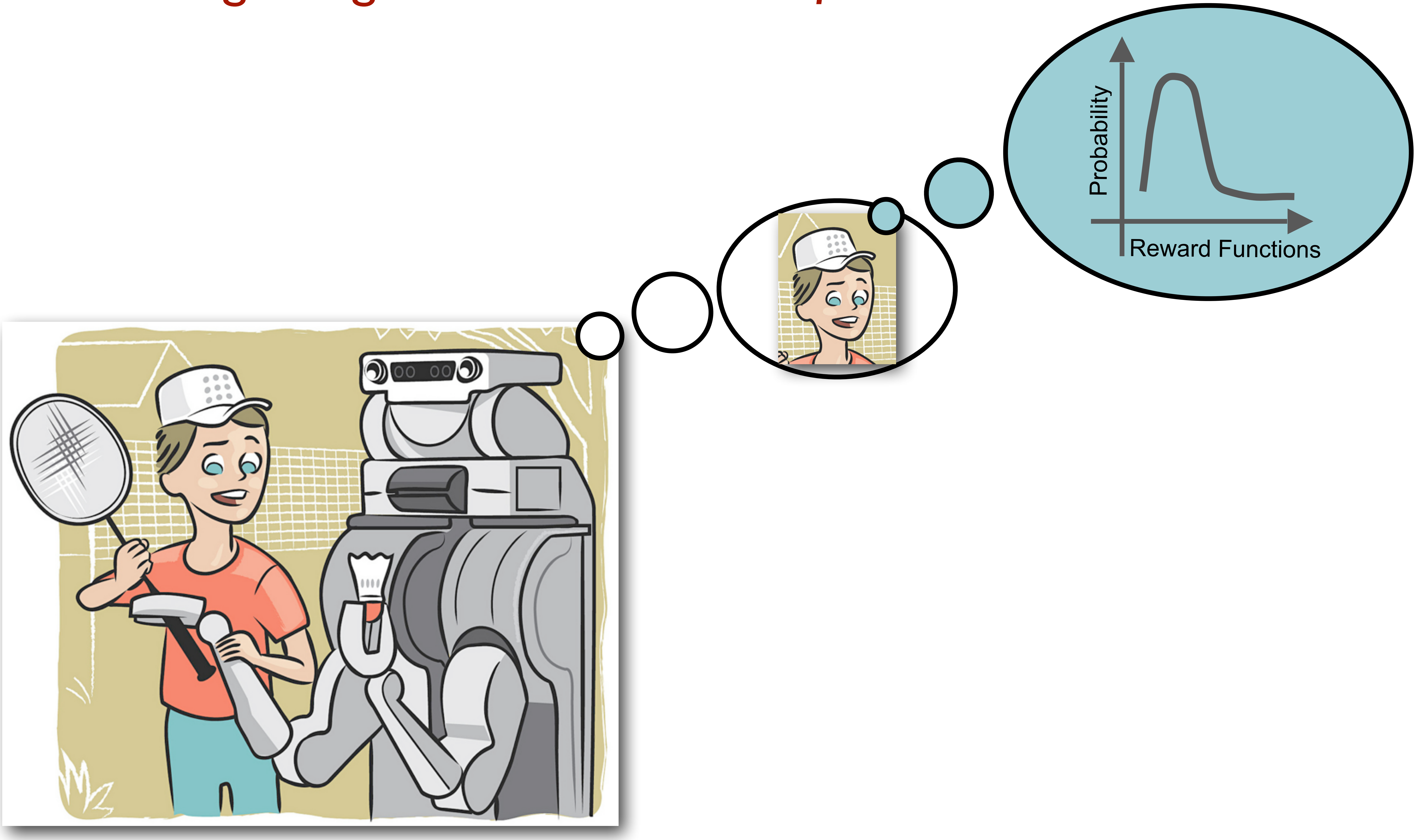
KL-divergence constraints

[Schulman et. al 2015]
[Achiam et. al 2017]

Need to address bad assumptions for efficiency

Safe Imitation Learning:

Upper bound the **policy loss** of the robot vs. human demonstrator with **high confidence**, *without knowing the ground-truth reward function.*



Inverse reinforcement learning: feature matching

(Abbeel and Ng 2004)

Policy value under linear reward function:
$$
\begin{aligned}
E_{s_0\sim D}[V^\pi(s_0)] &= E[\textstyle\sum_{t=0}^\infty \gamma^t R(s_t)|\pi] \\
&= E[\textstyle\sum_{t=0}^\infty \gamma^t w\cdot\phi(s_t)|\pi] \\
&= w\cdot E[\textstyle\sum_{t=0}^\infty \gamma^t \phi(s_t)|\pi]
\end{aligned}
$$

(Discounted) feature expectations:
$$\mu(\pi) = E[\textstyle\sum_{t=0}^\infty \gamma^t \phi(s_t)|\pi] \in \mathbb{R}^k.$$

Goal: find a reward function whose optimal policy matches expert's feature expectations

If expert's feature expectations are matched, then total return is also identical

Hoeffding-style bound (w.r.t. projection IRL algorithm)

(Abbeel and Ng 2004, Syed and Schapire 2008)

Theorem 2. *(Syed and Schapire 2008) To obtain a policy $\hat{\pi}$ such that with probability* $(1-\delta)$

$$\epsilon \geq |V^{\hat{\pi}}(R^*) - V^{\pi^*}(R^*)| \tag{26}$$

it suffices to have

$$m \geq \frac{2}{(\frac{\epsilon}{3}(1-\gamma))^2} \log \frac{2k}{\delta}. \tag{27}$$

Assumption:

Worst-case reasoning is the best we can do if we don't know the ground-truth reward function

↓

It is much more efficient to consider the likelihood of reward functions when assessing risk

D.S. Brown and S. Niekum.
Efficient Probabilistic Performance Bounds for Inverse Reinforcement Learning.
AAAI Conference on Artificial Intelligence, February 2018.

D.S. Brown, Y. Cui, and S. Niekum.
Risk-Aware Active Inverse Reinforcement Learning.
Conference on Robot Learning (CoRL), October 2018.

Rethinking feature expectations

Problem 1: Hoeffding method bounds the features expectations, which in turn, bounds loss under a worst-case reward function, regardless of its likelihood given the demonstrations

Problem 2: Feature expectation methods cannot learn from state-action pairs that aren't part of a full trajectory

Bayesian Inverse Reinforcement Learning (BIRL)

[Ramachandran and Amir 2007]

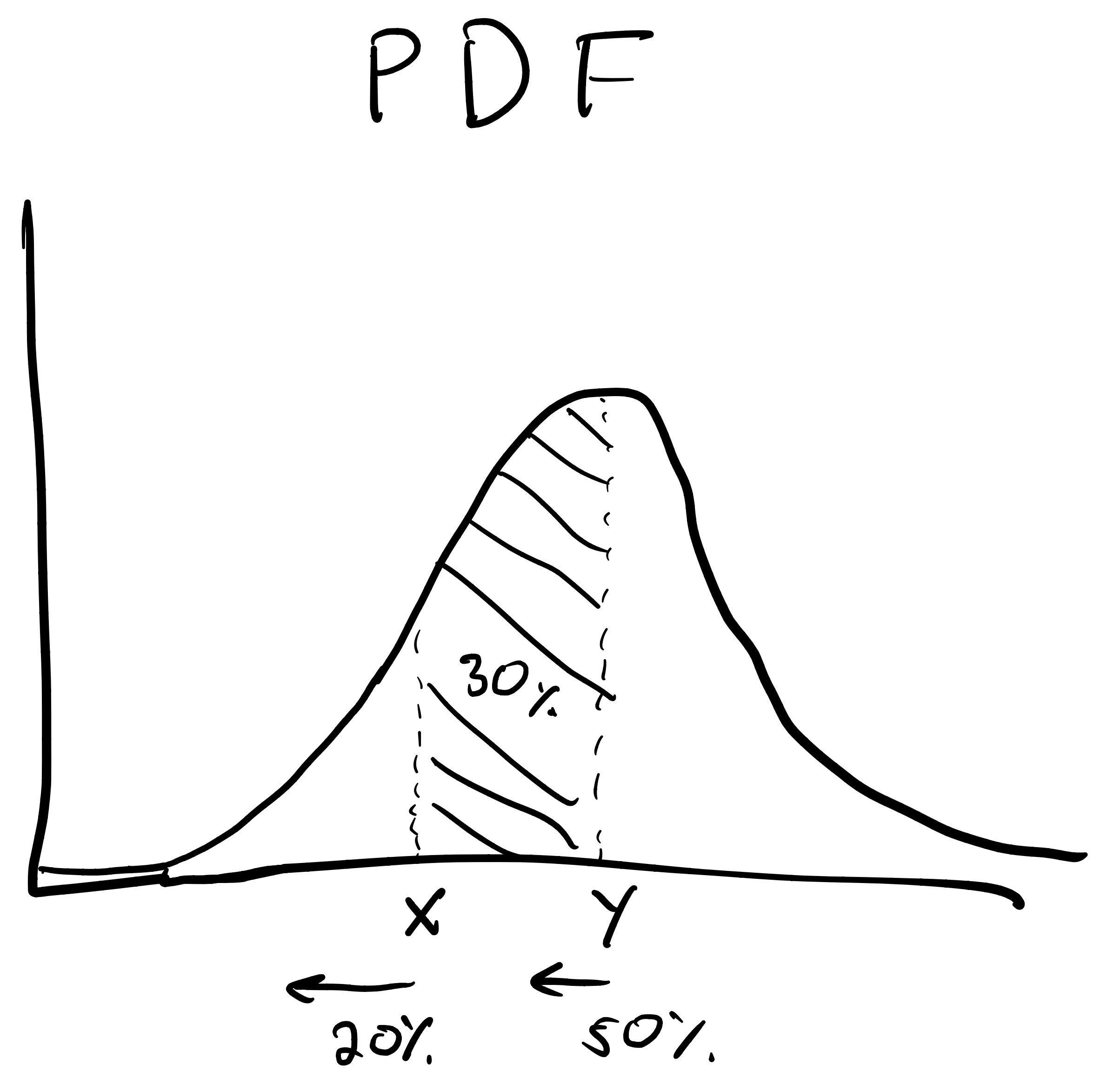
- Use MCMC to sample from posterior:

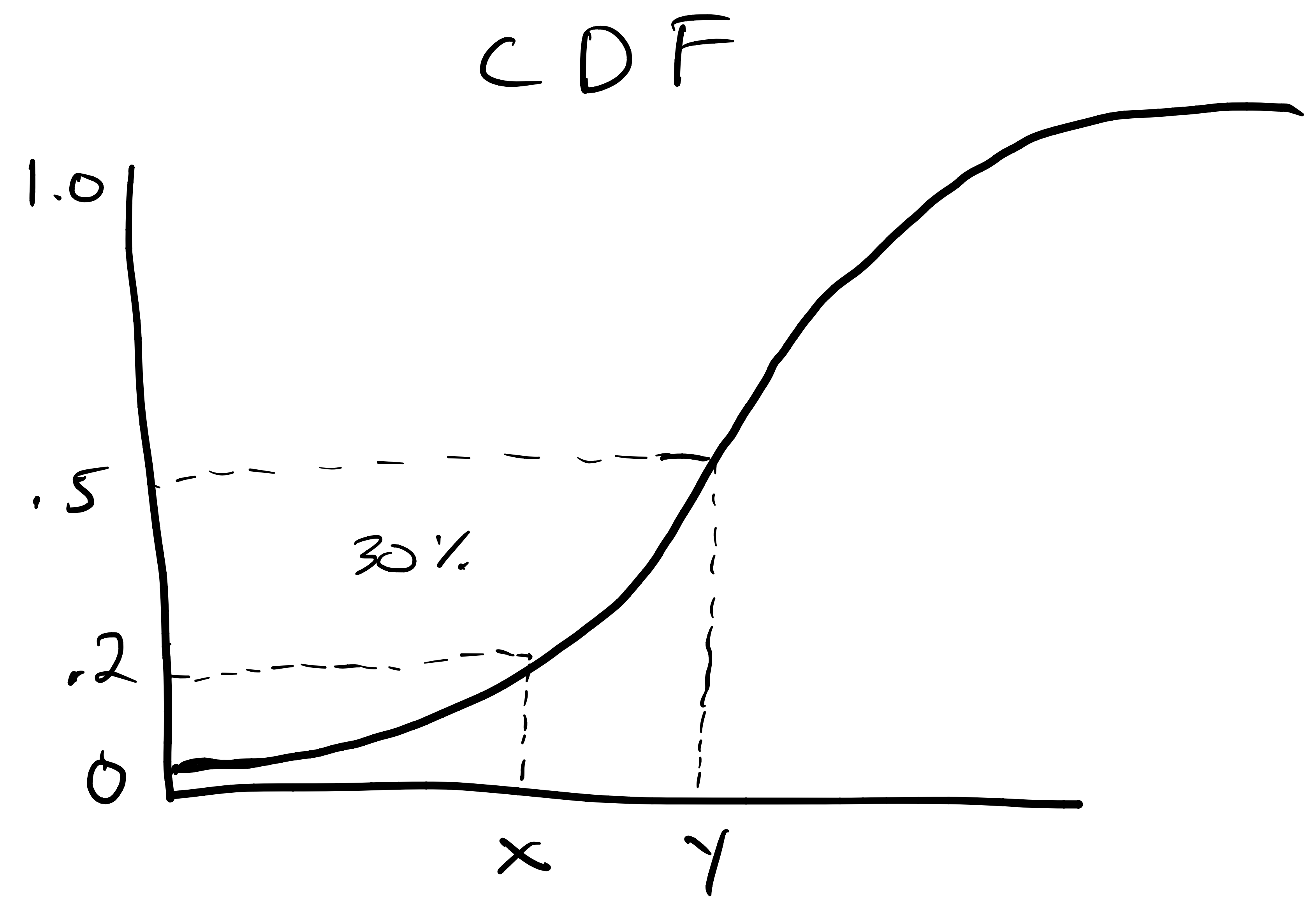
$$P(R|D) \propto P(D|R)P(R)$$

- Assume demonstrations follow softmax policy with temperature c:

$$P(D|R) = \prod_{(s,a)\in D} \frac{e^{cQ^*(s,a,R)}}{\sum_{b\in A} e^{cQ^*(s,b,R)}}$$

Sampling from a distribution





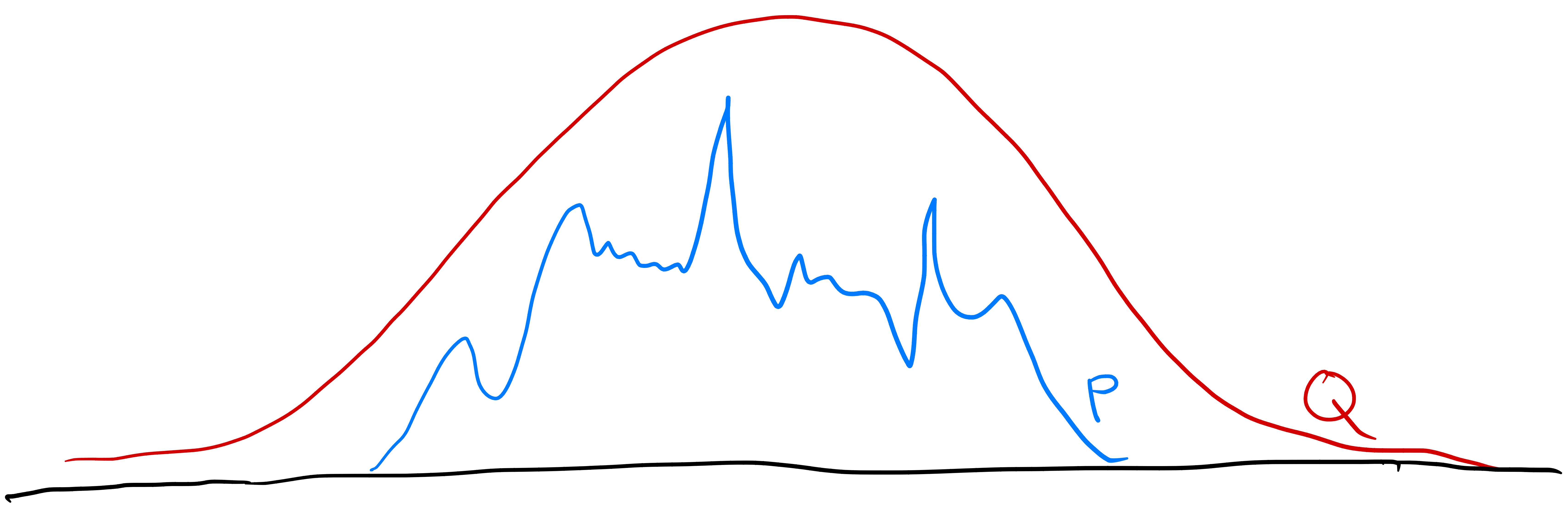
If we know CDF, then sample:

$u \sim \text{uniform}$

$z \sim CDF^{-1}(u)$

If not...

Rejection Sampling



Actual dist: P

Unnormalized Proposal dist: Q

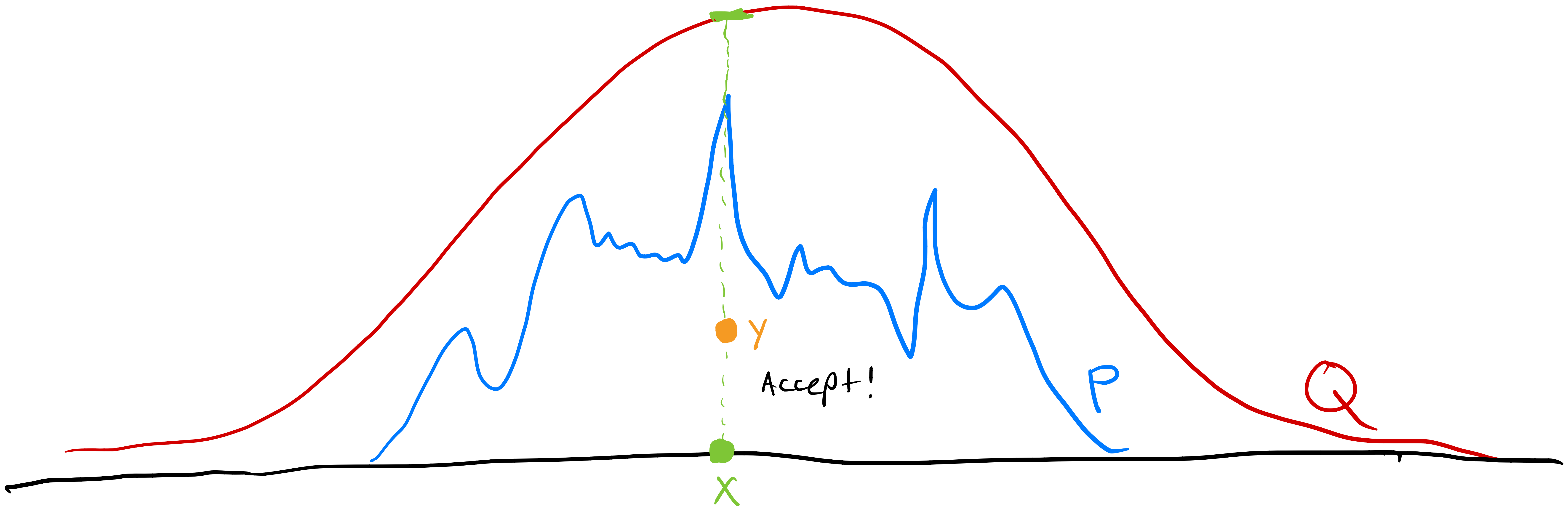
Sample $x \sim Q$

Sample $y \sim [0, Q(x)]$

Accept if $y \leq P(x)$

Only need to evaluate $P(x)$, not CDF!

Rejection Sampling



Actual dist: P

Unnormalized Proposal dist: Q

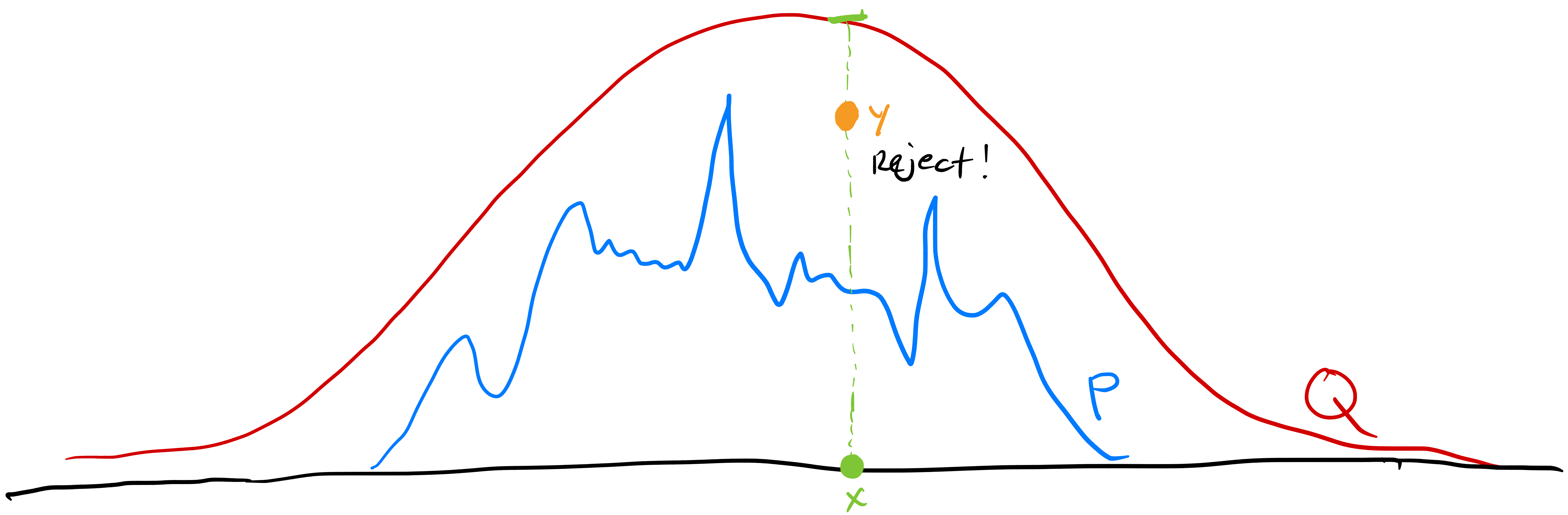
Sample $x \sim Q$

Sample $y \sim [0, Q(x)]$

Accept if $y \leq P(x)$

Only need to evaluate $P(x)$, not CDF!

Rejection Sampling



Actual dist: P

Unnormalized Proposal dist: Q

Sample $x \sim Q$

Sample $y \sim [0, Q(x)]$

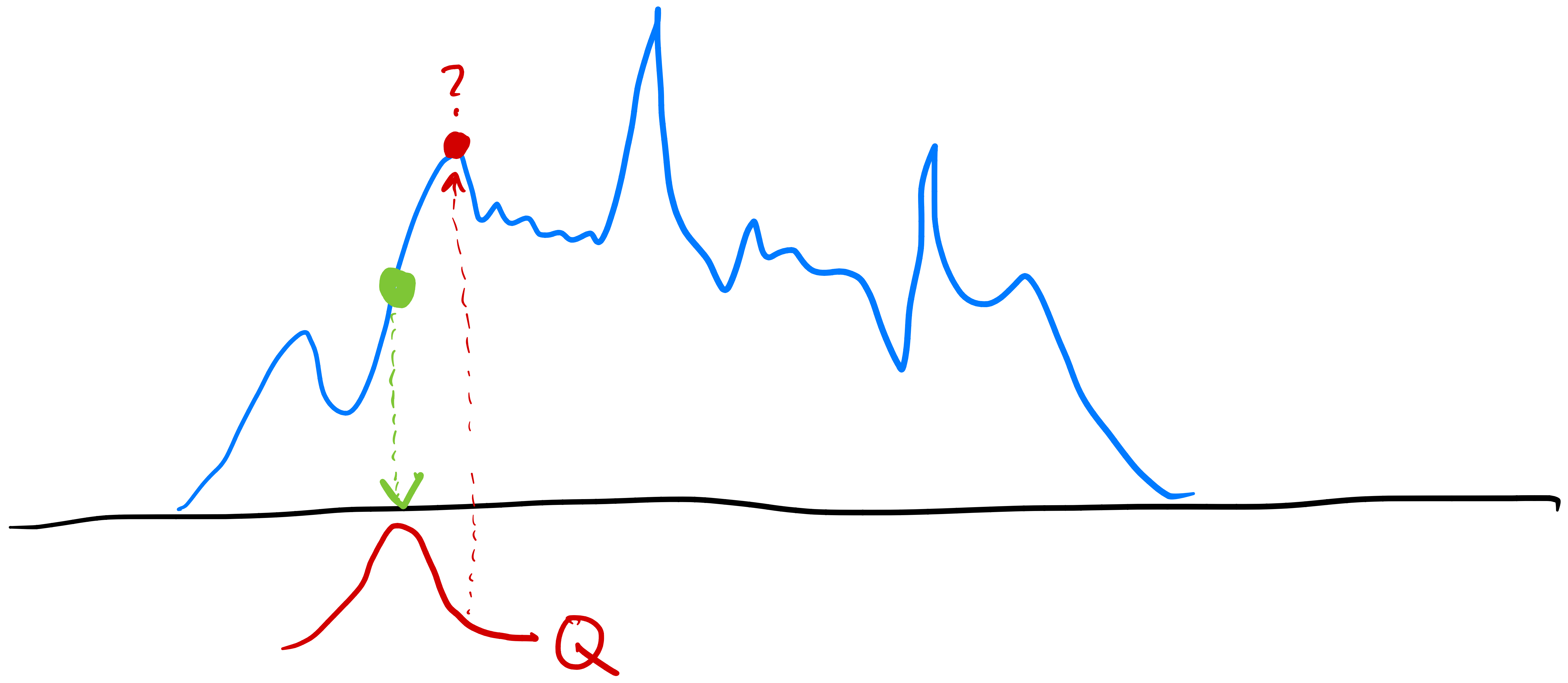
Accept if $y \leq P(x)$

Only need to evaluate $P(x)$, not CDF!

MCMC: Markov Chain Monte Carlo

Idea: Hard to choose Q to minimize rejections

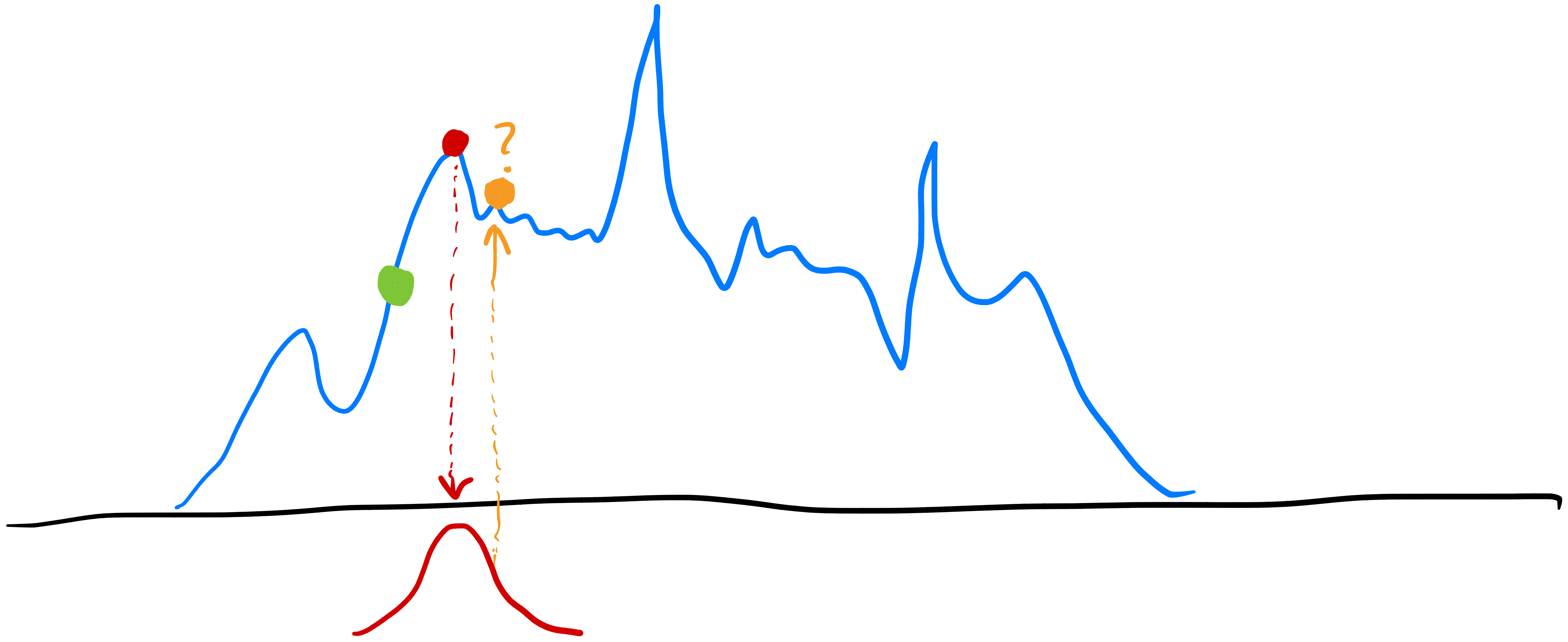
Use an adaptive Q!



MCMC: Markov Chain Monte Carlo

Idea: Hard to choose Q to minimize rejections

Use an adaptive Q!



Sampling will occur as a random walk of a specially crafted Markov chain

Markov property: $p(x_{t+1} | x_1 \dots x_t) = p(x_{t+1} | x_t)$

Thus, $p(x_{t+1}) = \sum_{x_t} p(x_{t+1} | x_t)\, p(x_t)$

Invariant distribution: p^* is invariant w.r.t. MC if:

$$p^*(x') = \sum_x p(x' | x)\, p^*(x)$$ i.e. each step leaves p^* unchanged

Detailed balance: sufficient, but not necessary for invariance

$$p^*(x)\, p(x' | x) = p^*(x')\, p(x | x')$$

Proof:

$$p(x') = \sum_{x'} p^*(x')\, p(x | x')$$

$$= \sum_{x'} p^*(x)\, p(x' | x)$$ via detailed balance

$$= p^*(x) \sum_{x'} p(x' | x)$$

$$= p^*(x) \cdot 1$$

$$= p^*(x)$$

Intuition:

If we set up MC so it is invariant, our walk will sample from desired distribution p^*

Metropolis-Hastings

Want to sample from $p(x)$

Able to evaluate some unnormalized $\tilde{p}(x)$ e.g. a likelihood fxn

Proposal dist: $q(x \mid x_t)$ e.g. Gaussian centered at x_t

1. Sample $x^* \sim q(x \mid x_t)$
2. Accept with probability:

$$A(x^*, x_t) = \min\left(1, \frac{\tilde{p}(x^*)\, q(x_t \mid x^*)}{\tilde{p}(x_t)\, q(x^* \mid x_t)}\right) = \min\left(1, \frac{\tilde{p}(x^*)}{\tilde{p}(x_t)}\right)$$

"Metropolis" for symmetric q

Intuition: More likely steps always accepted; less likely proportional to likelihood ratio

Proof:

$$p(x)\, q(x \mid x')\, A(x', x)$$
$$= \min\left[p(x)\, q(x \mid x'),\ p(x')\, q(x' \mid x)\right]$$
$$= \min\left[p(x')\, q(x' \mid x),\ p(x)\, q(x \mid x')\right]$$
$$= p(x')\, q(x' \mid x)\, A(x, x')$$

Distribution with acceptance probs factored in meets detailed balance

→ Therefore samples from $p(x)$!

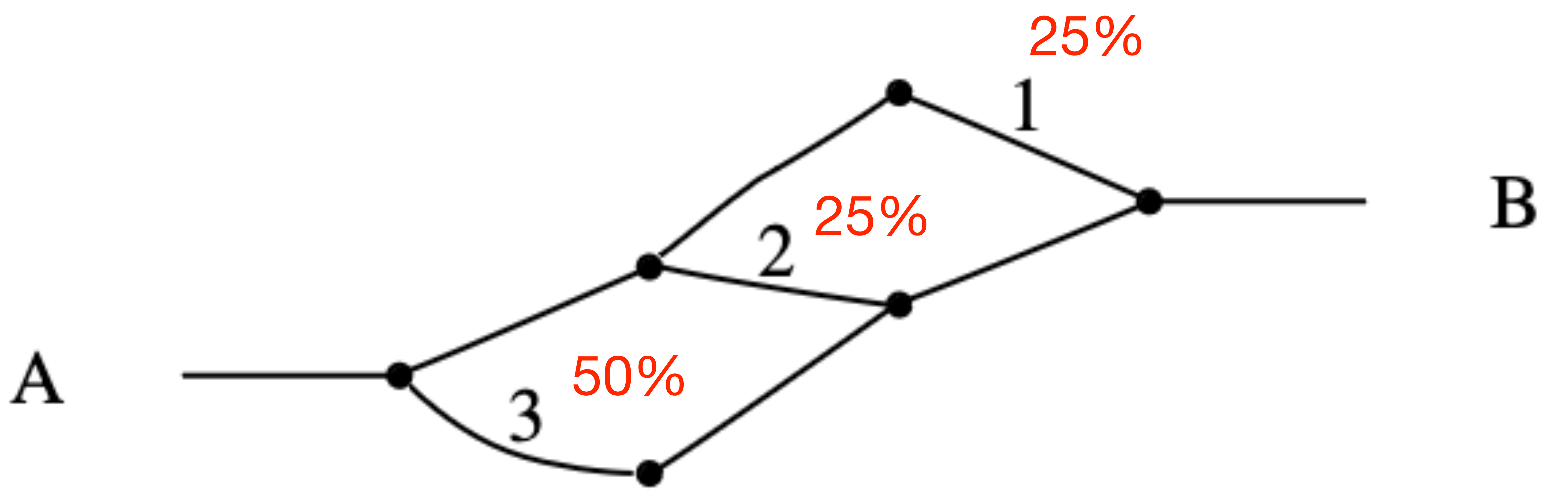
Where should priors come from?

Trajectory vs. action-based reasoning

$$P(\text{action } a|\theta, T) \propto \sum_{\zeta : a \in \zeta_{t=0}} P(\zeta|\theta, T)$$

Vs.

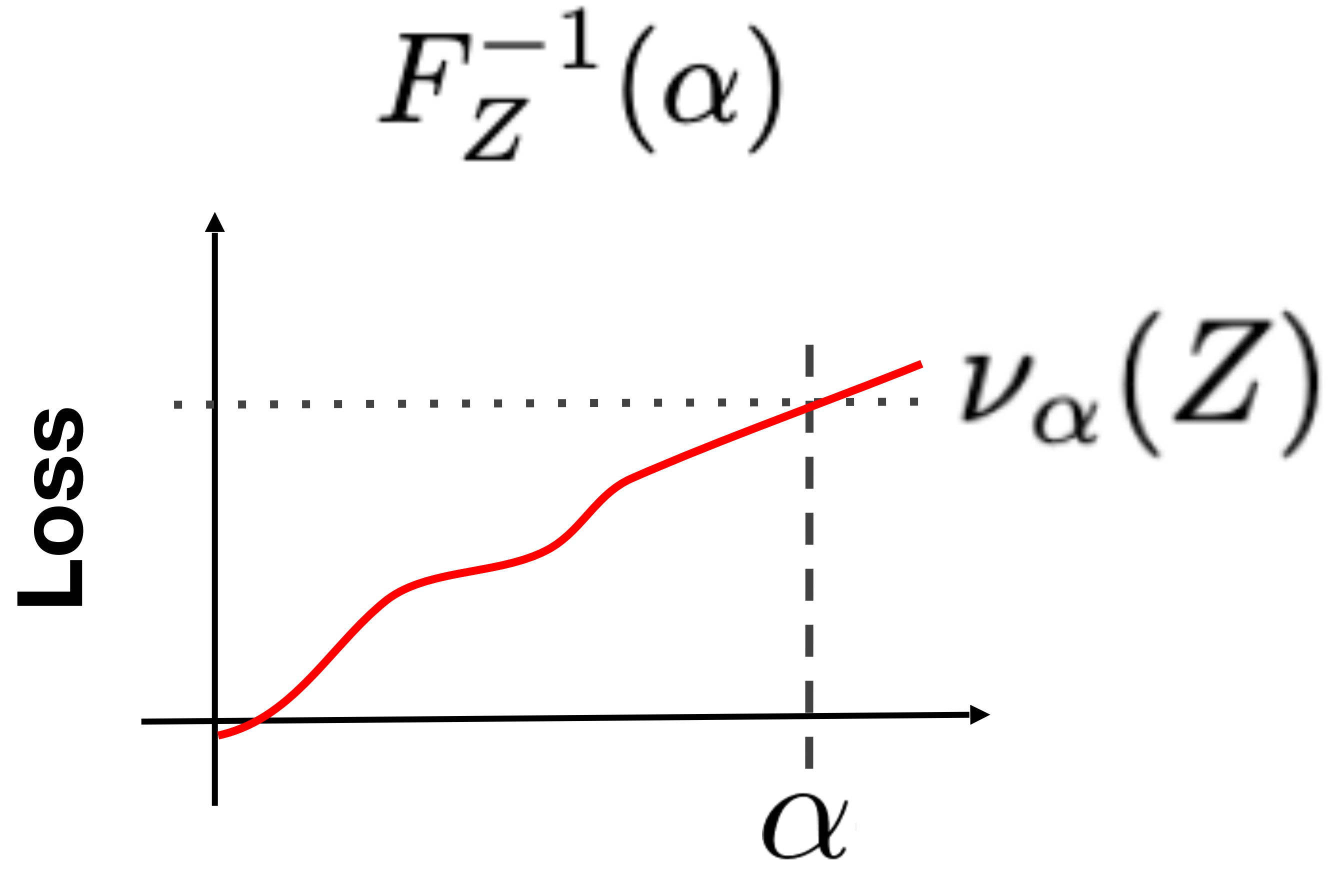
$$P(\text{action } a|s_i, \theta) \propto e^{Q^*(s_i, a)}$$



Paths 1, 2, and 3 have equal return, so all should be p=1/3 under MaxEnt

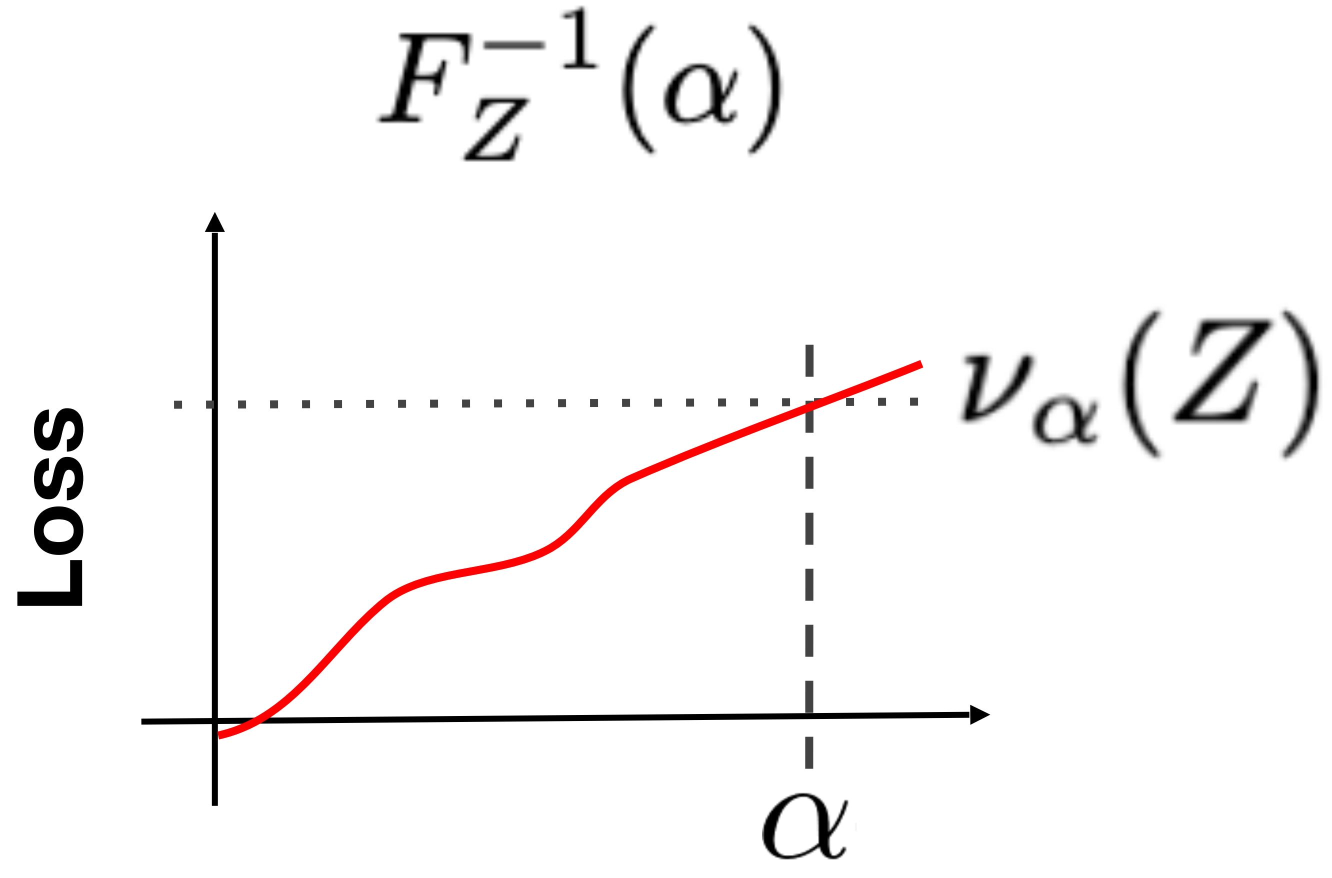
Value at risk

$$\nu_\alpha(Z) = F_Z^{-1}(\alpha) = \inf\{z : F_Z(z) \geq \alpha\}$$



Value at risk

$$\nu_\alpha(Z) = F_Z^{-1}(\alpha) = \inf\{z : F_Z(z) \geq \alpha\}$$



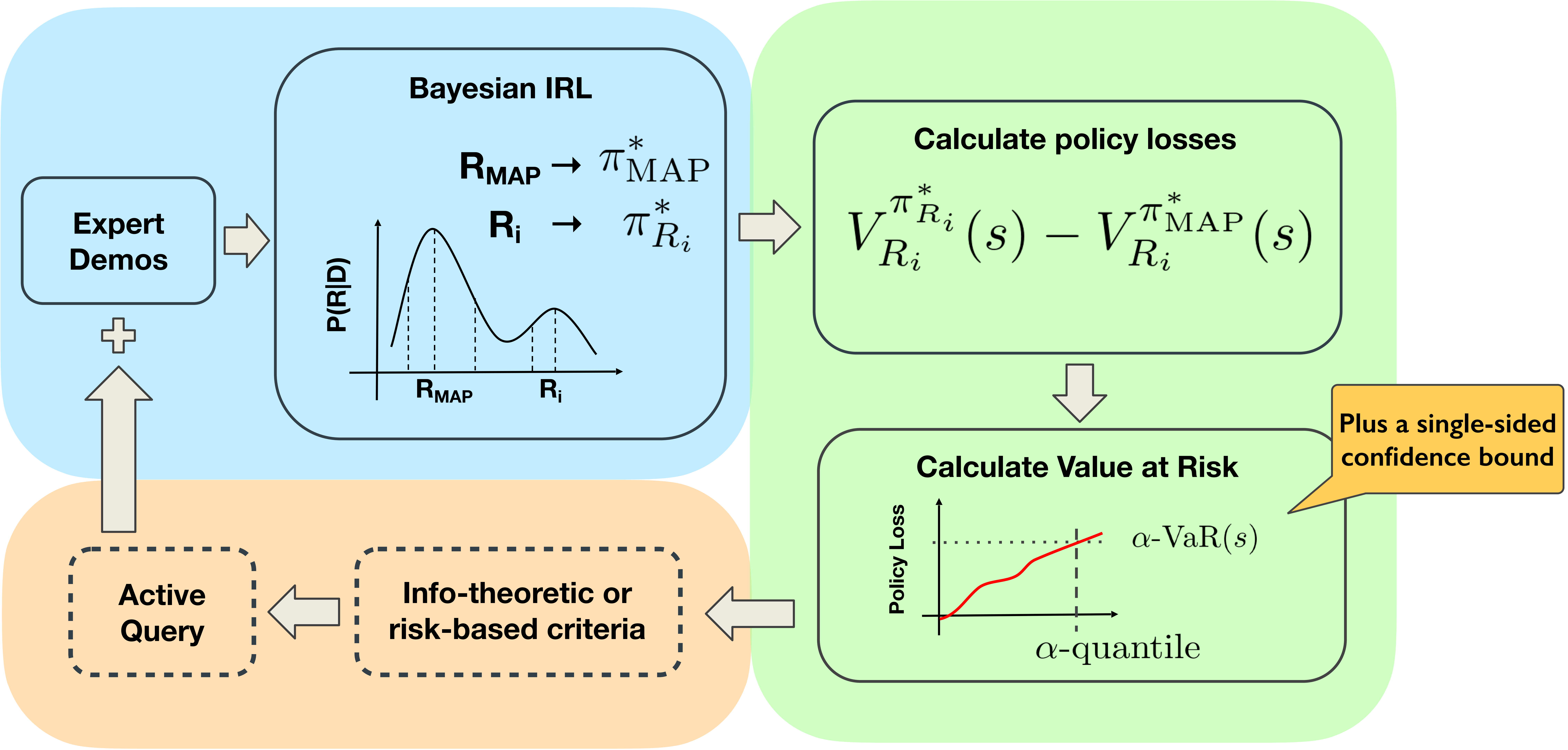
+

Single-sided confidence bound

"With probability $1-\delta$, no more than $1-\alpha\%$ of the outcomes will be worse than X"

Goal: Solve for X and check if it is below acceptable risk level

(Active) Safe IRL



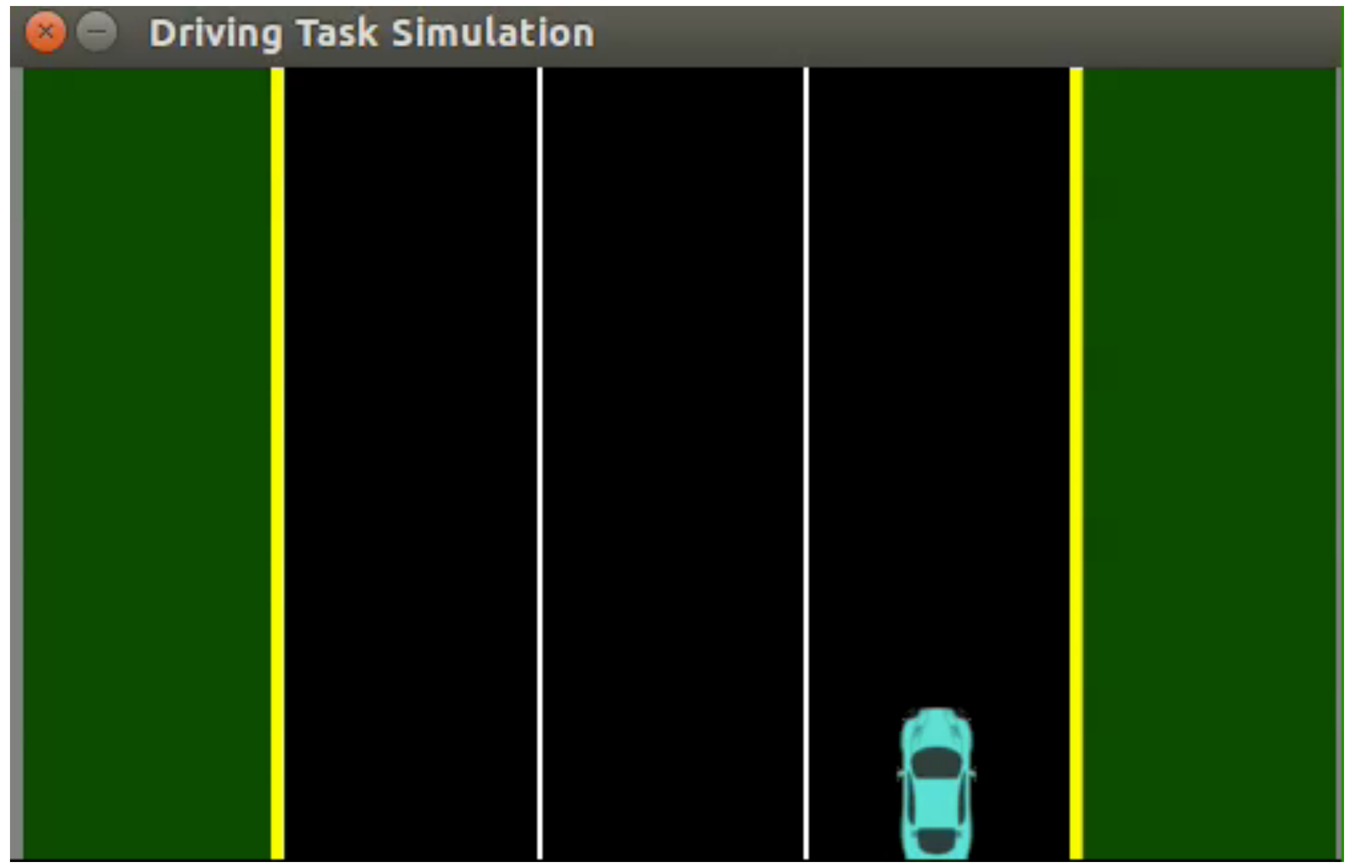
Results: efficiency (no active learning)

	Number of demonstrations					Average Accuracy
	1	5	9	···	23,146	
0.95-VaR EVD Bound	**0.9372**	**0.2532**	**0.1328**		-	0.98
0.99-VaR EVD Bound	1.1428	0.2937	0.1535		-	1.0
EVD Bound (Syed and Schapire 2008)	142.59	63.77	47.53		0.9372	1.0

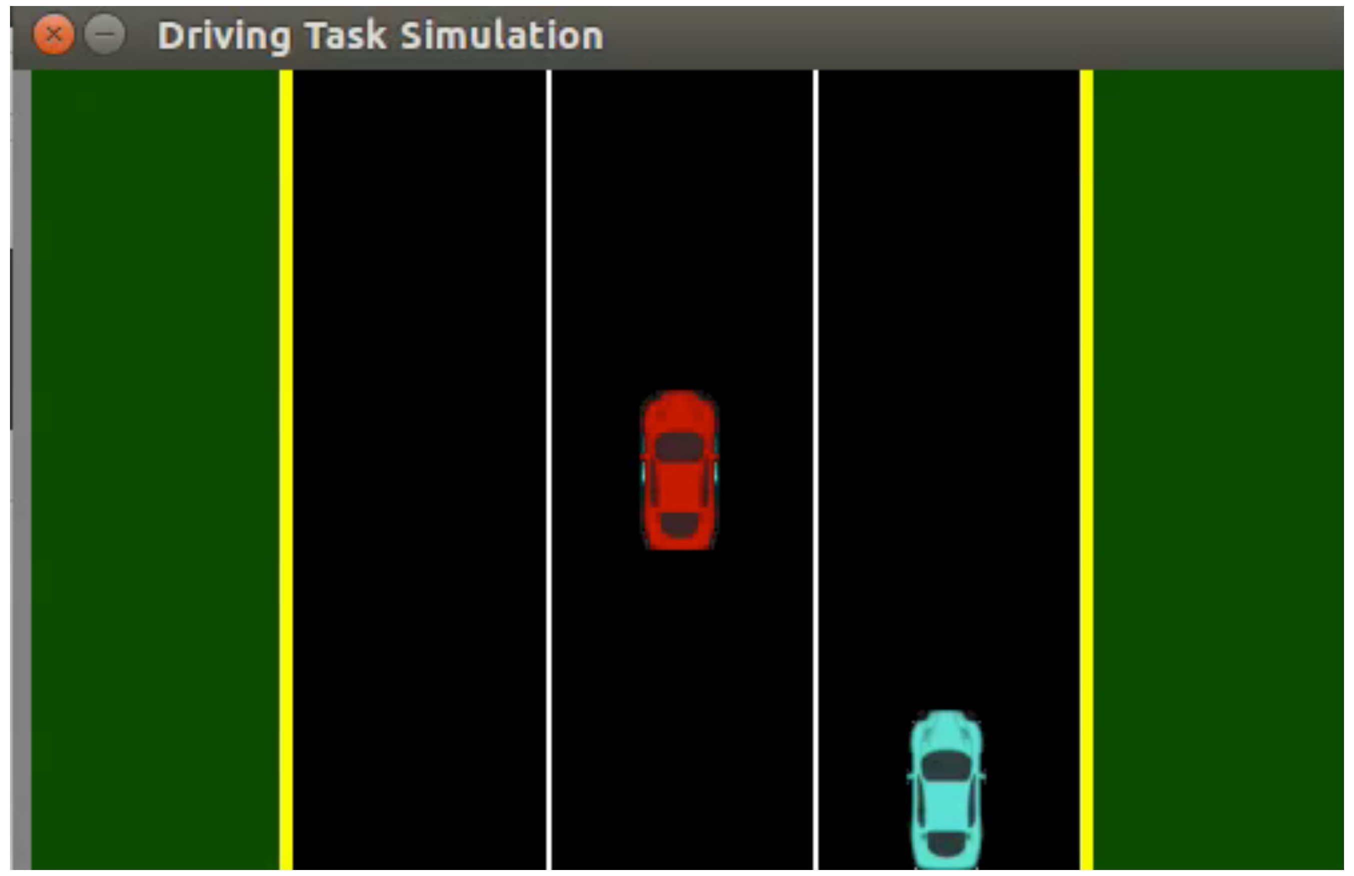
Table 1: Comparison of 95% confidence α-VaR bounds with a 95% confidence Hoeffding-style bound (Syed and Schapire 2008). Both bounds use the Projection algorithm (Abbeel and Ng 2004) to obtain the evaluation policy. Results are averaged over 200 random navigation tasks.

Four orders of magnitude more data efficient!

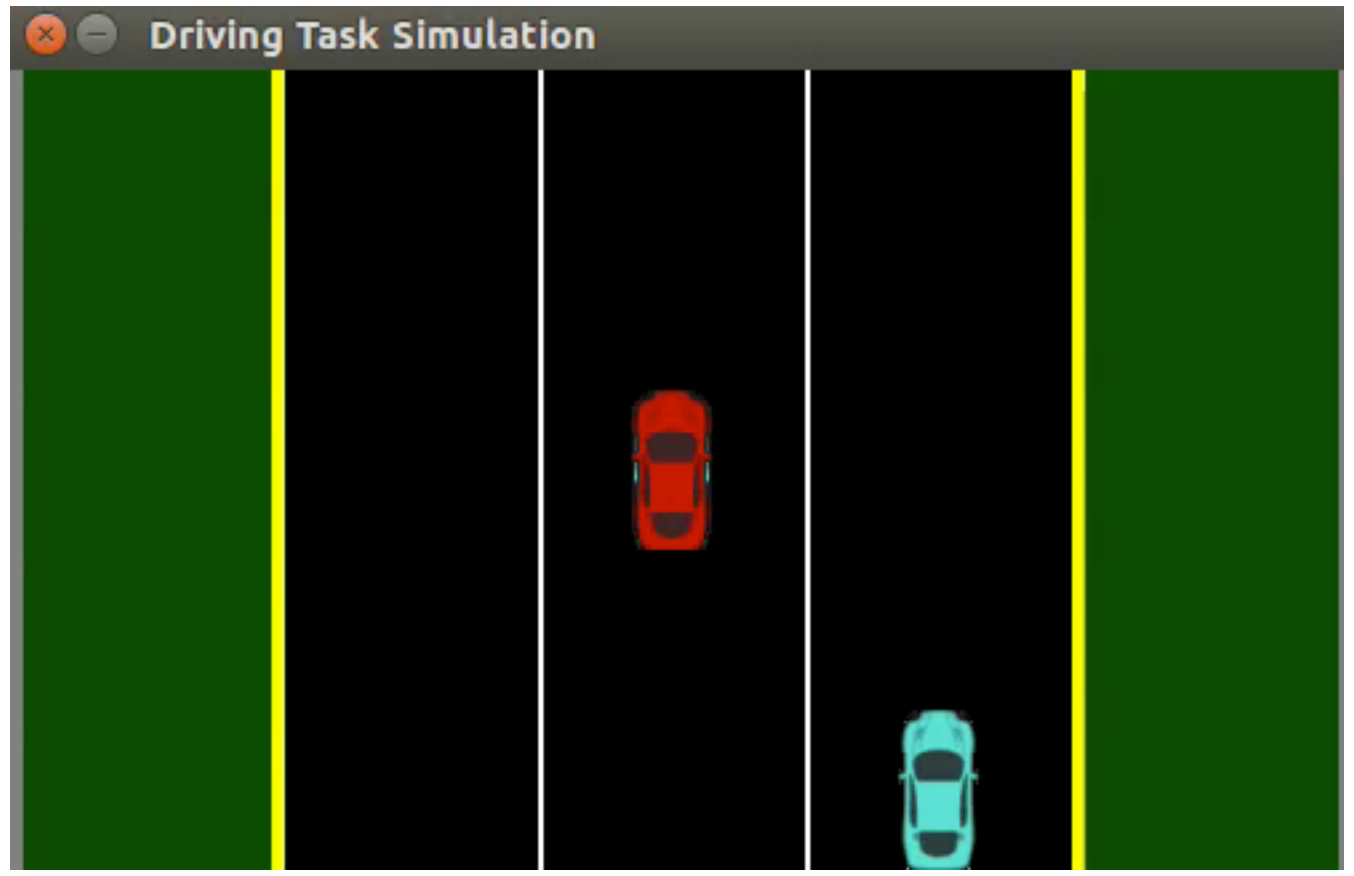
Risk-sensitive preferences



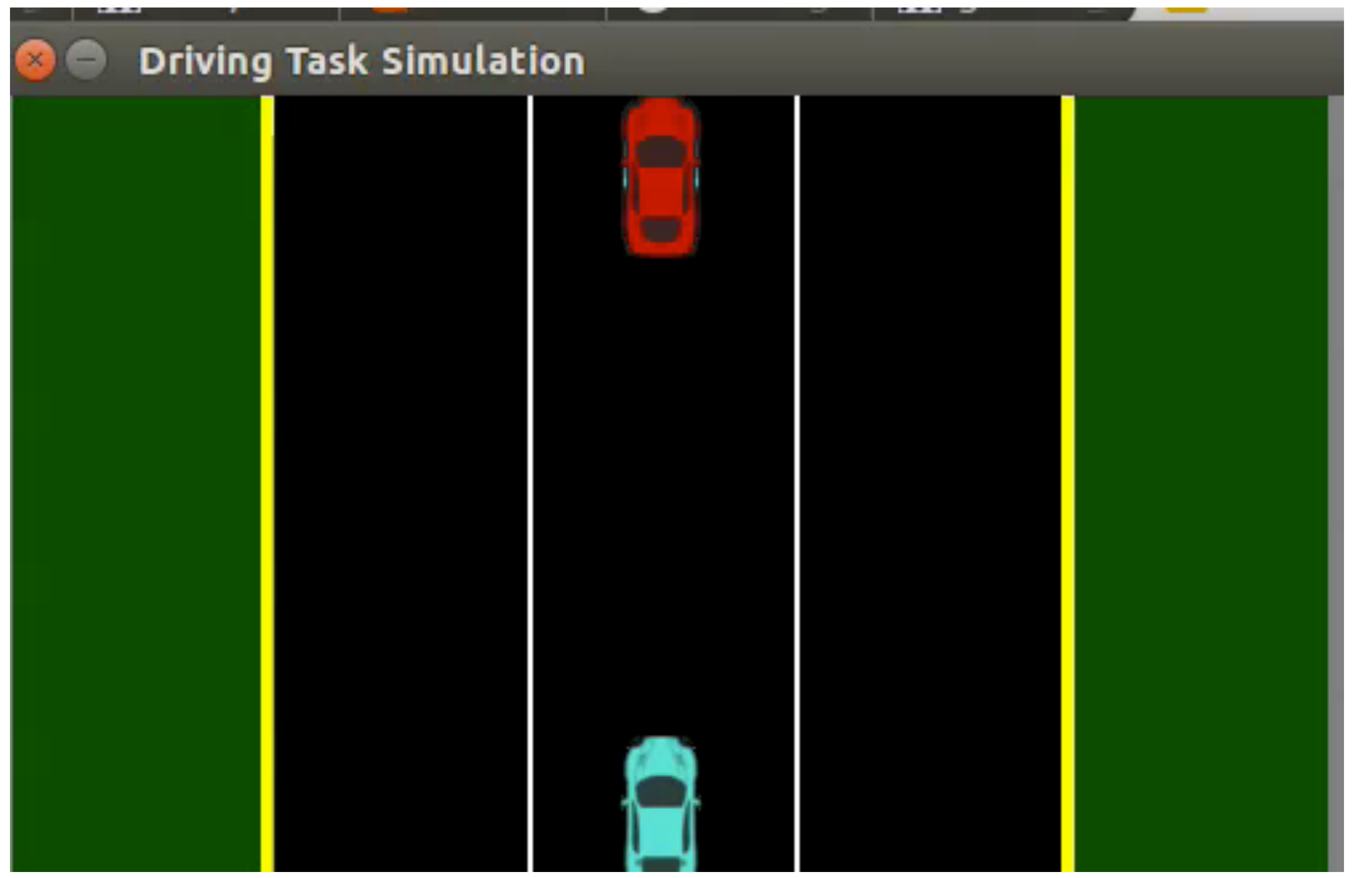
Demonstration: avoids cars, no lane pref



Avoids cars, but prefers right lane

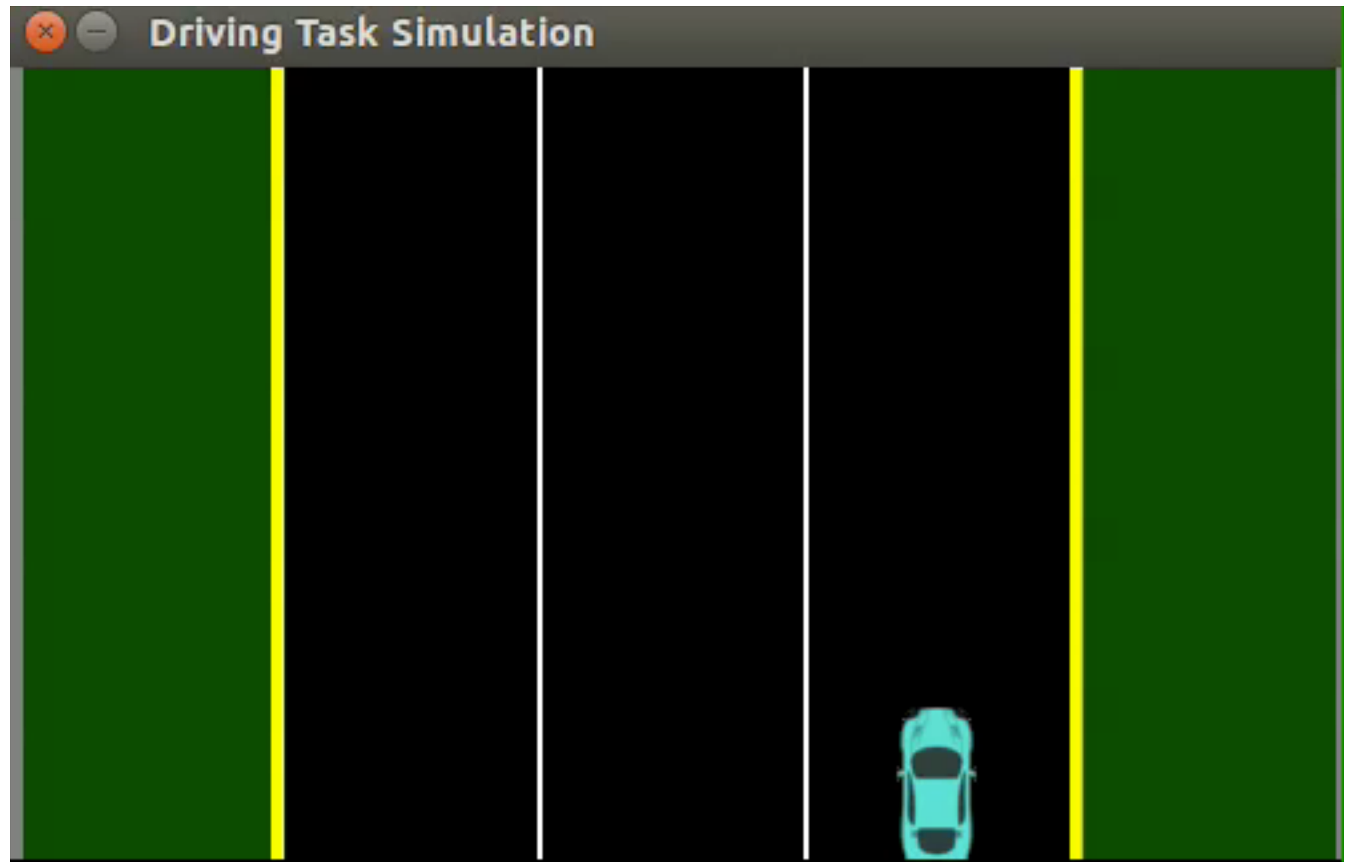


Stays on road, but ignores other cars



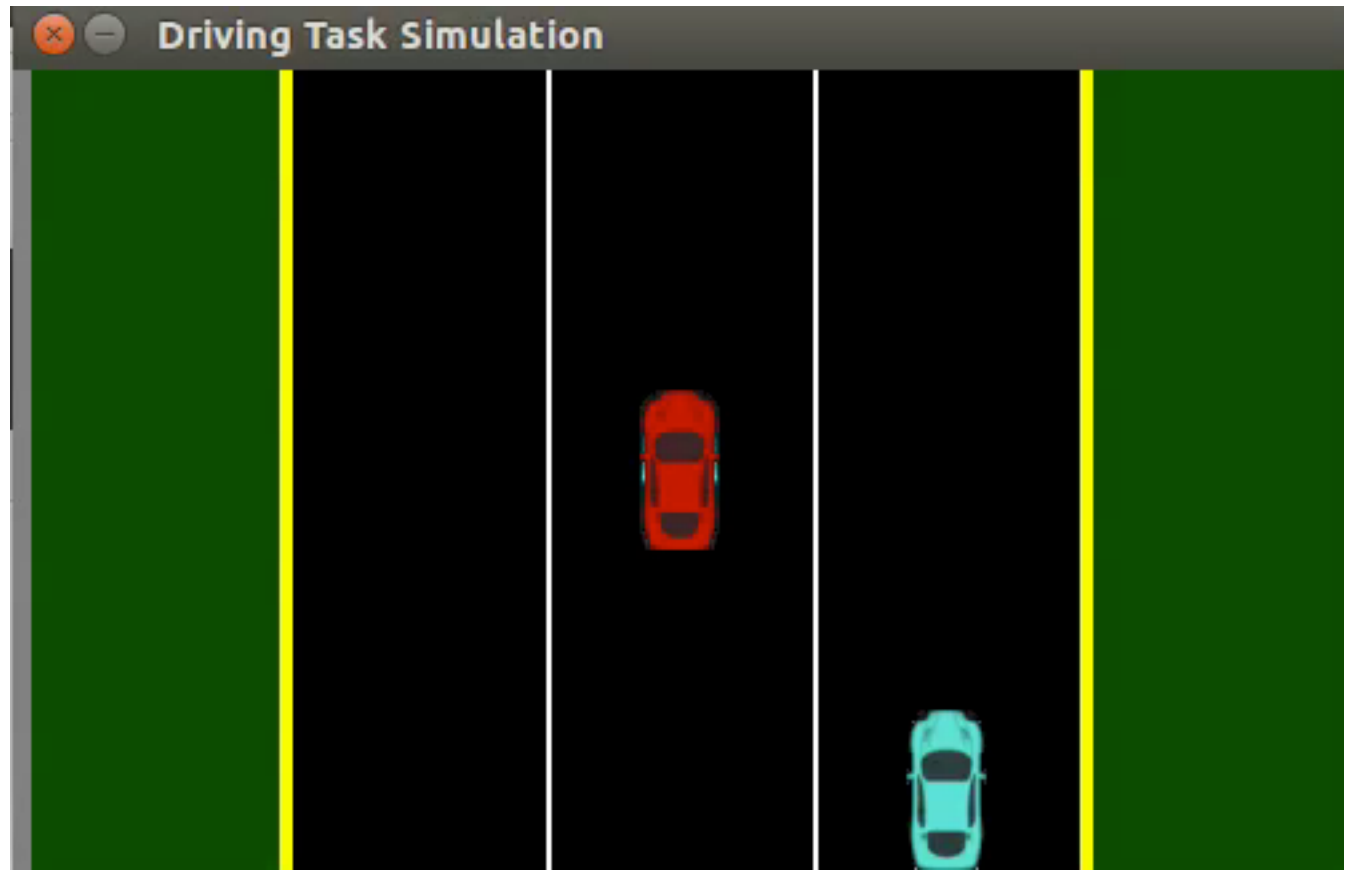
Seeks collisions

Risk-sensitive preferences (feature count-based)



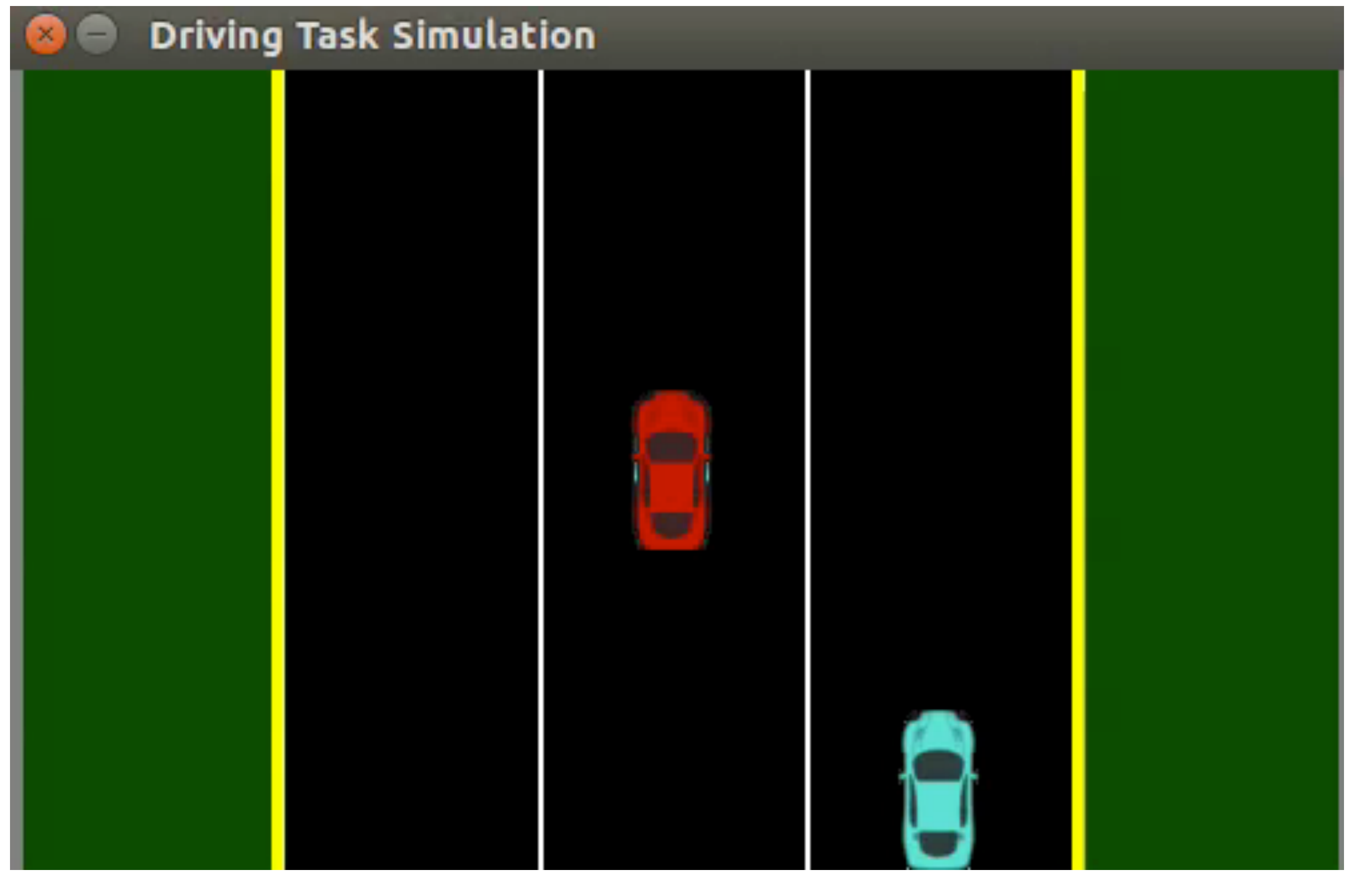
Demonstration: avoids cars, no lane pref

3



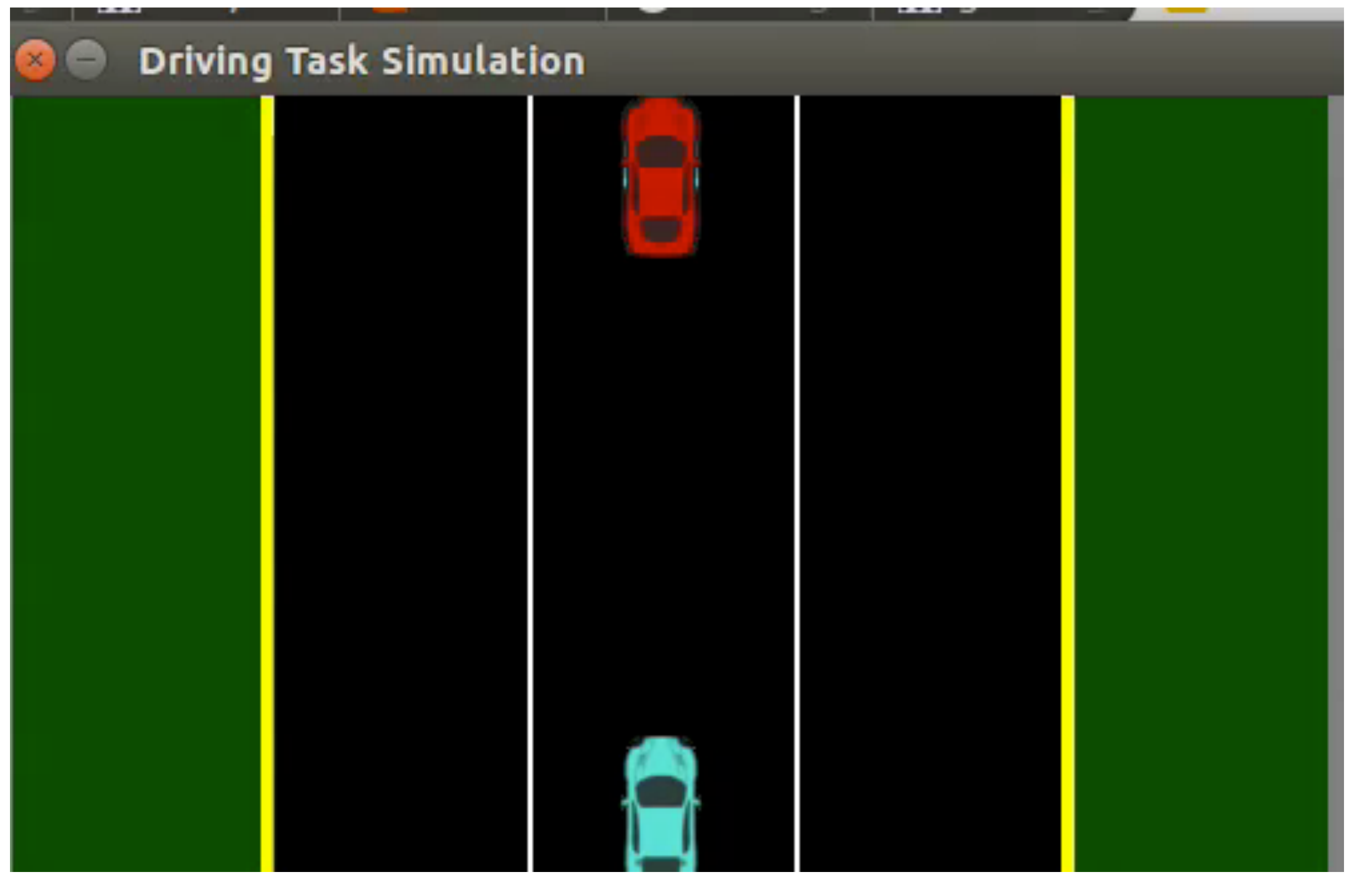
Avoids cars, but prefers right lane

1



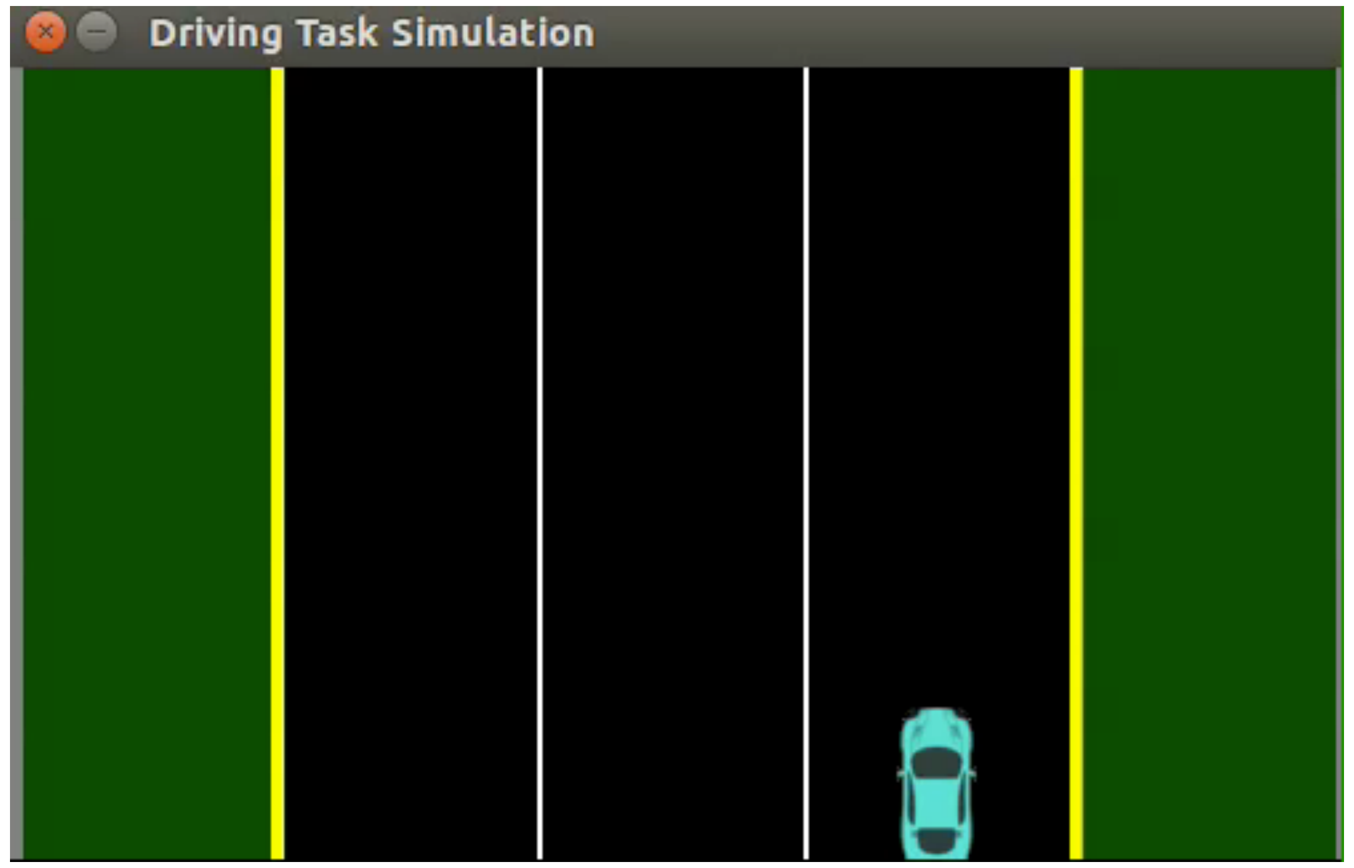
Stays on road, but ignores other cars

2



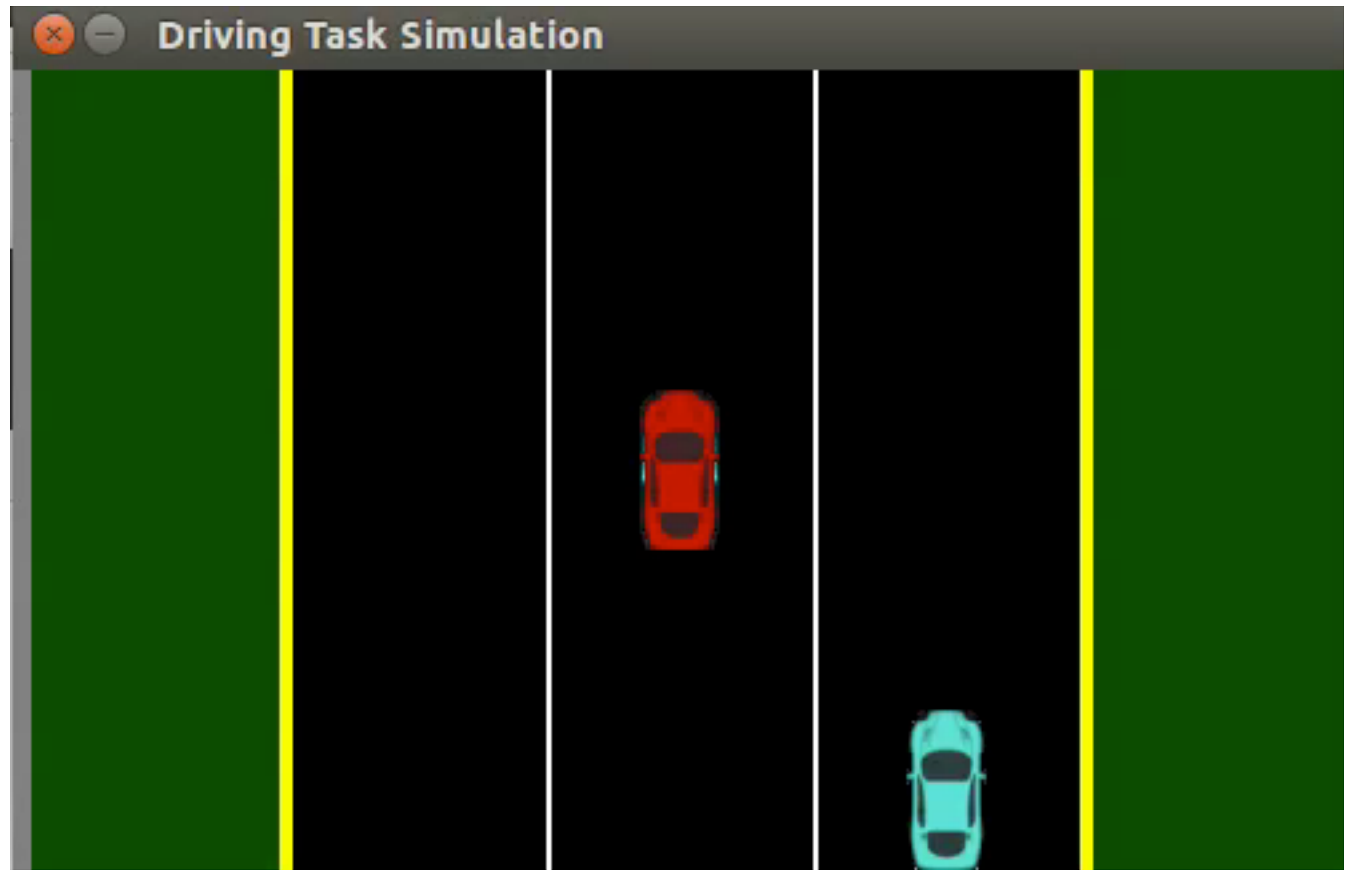
Seeks collisions

Risk-sensitive preferences (our approach)



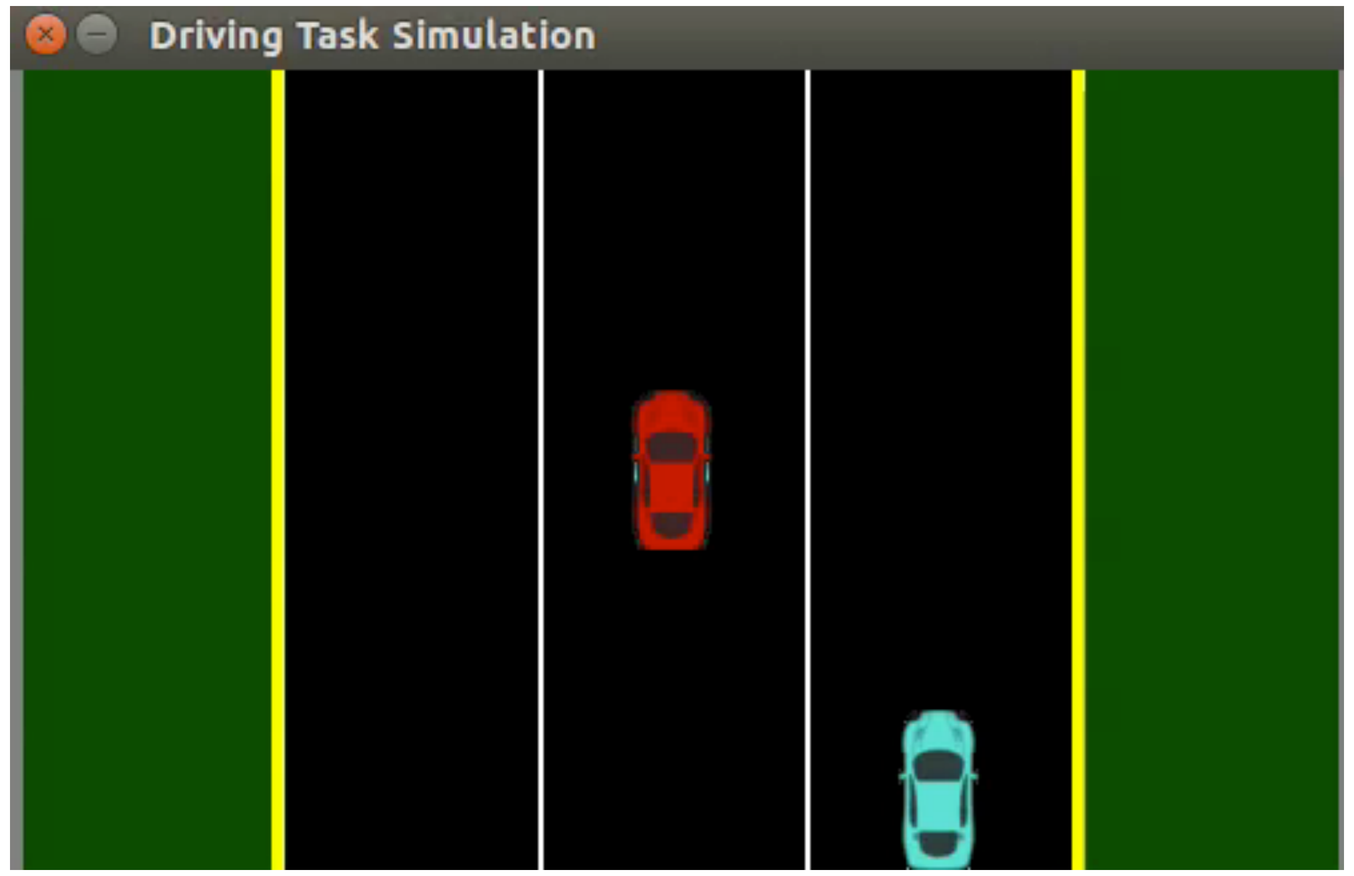
Demonstration: avoids cars, no lane pref

1



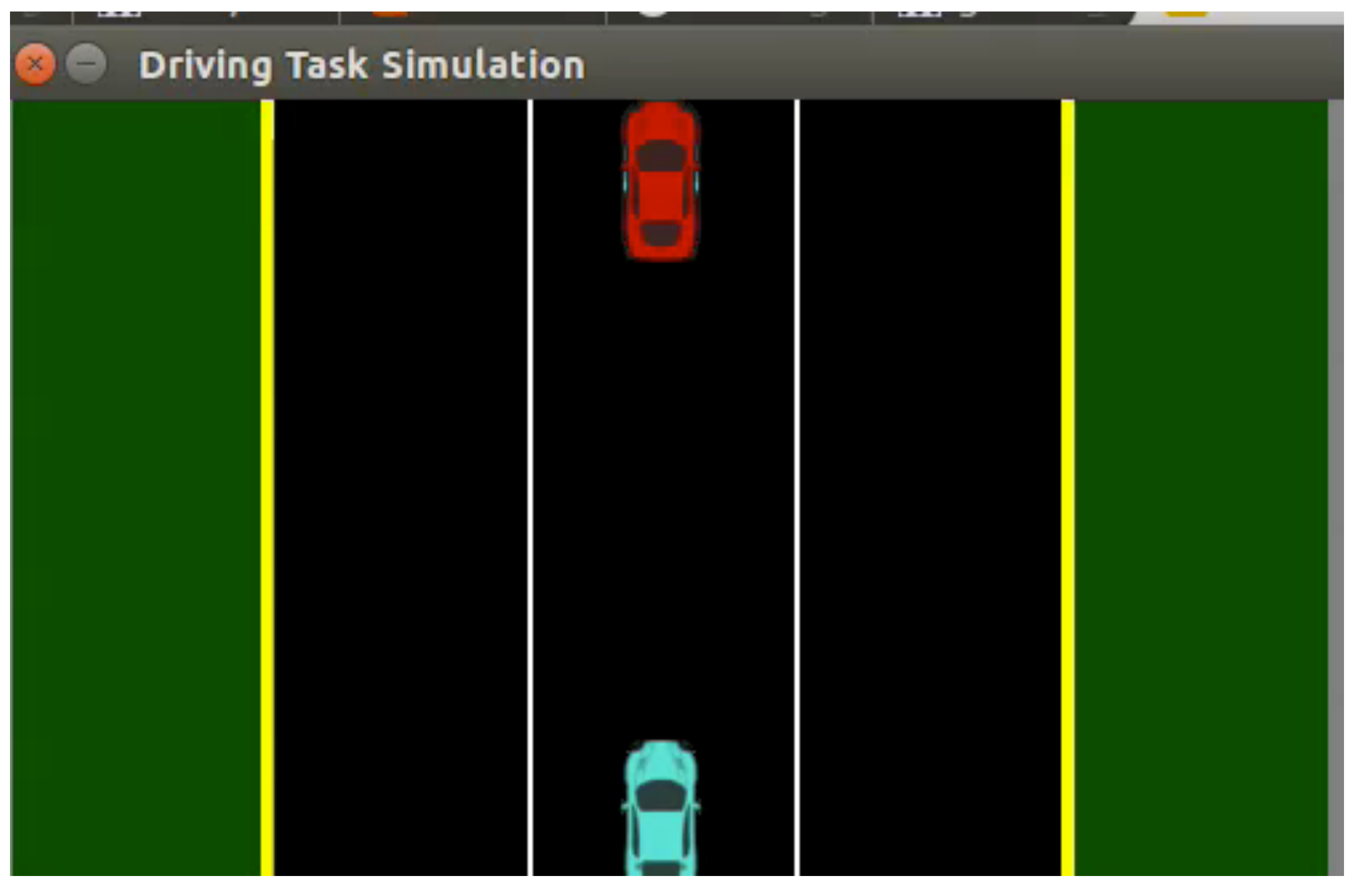
Avoids cars, but prefers right lane

2



Stays on road, but ignores other cars

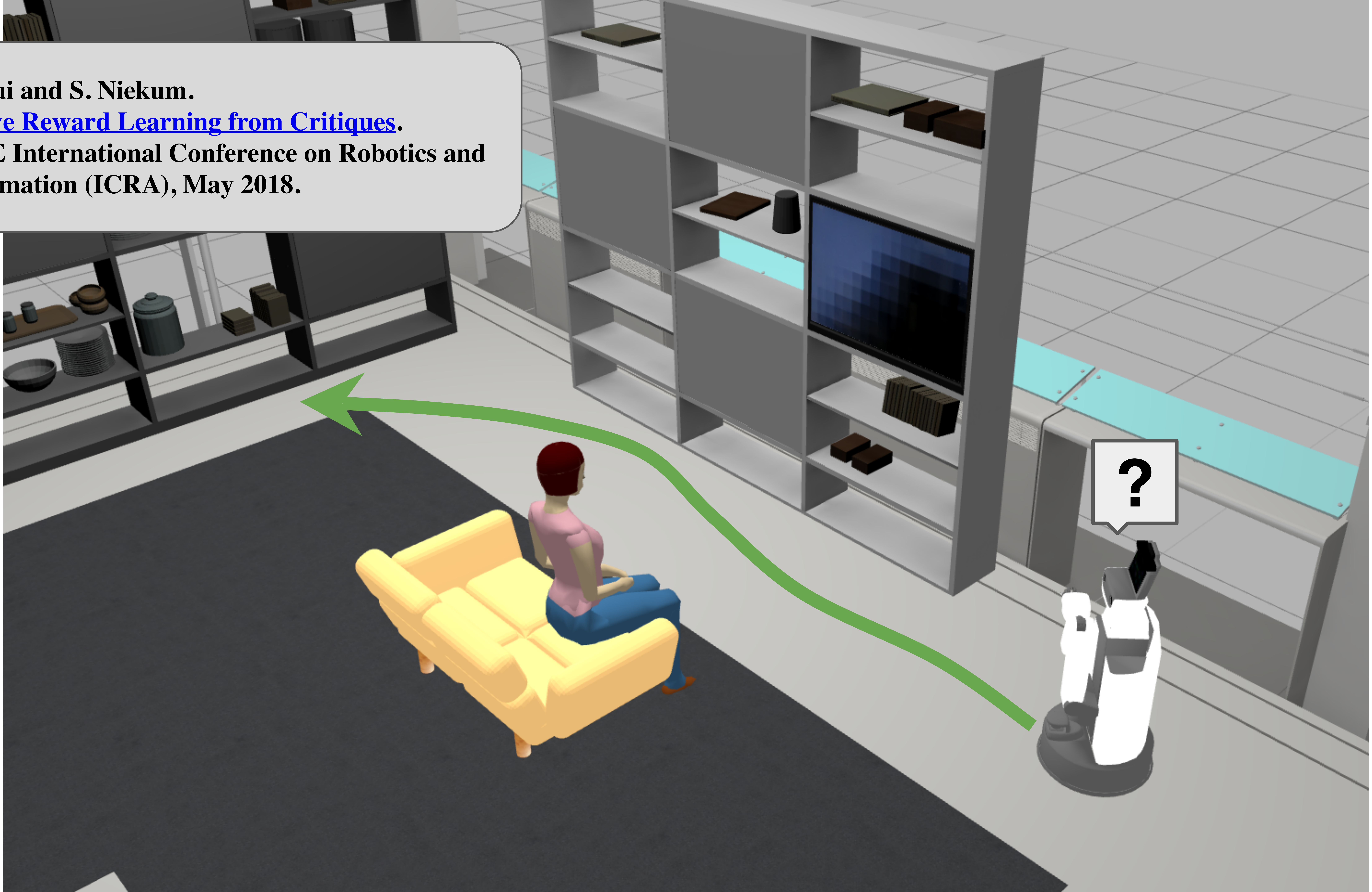
3



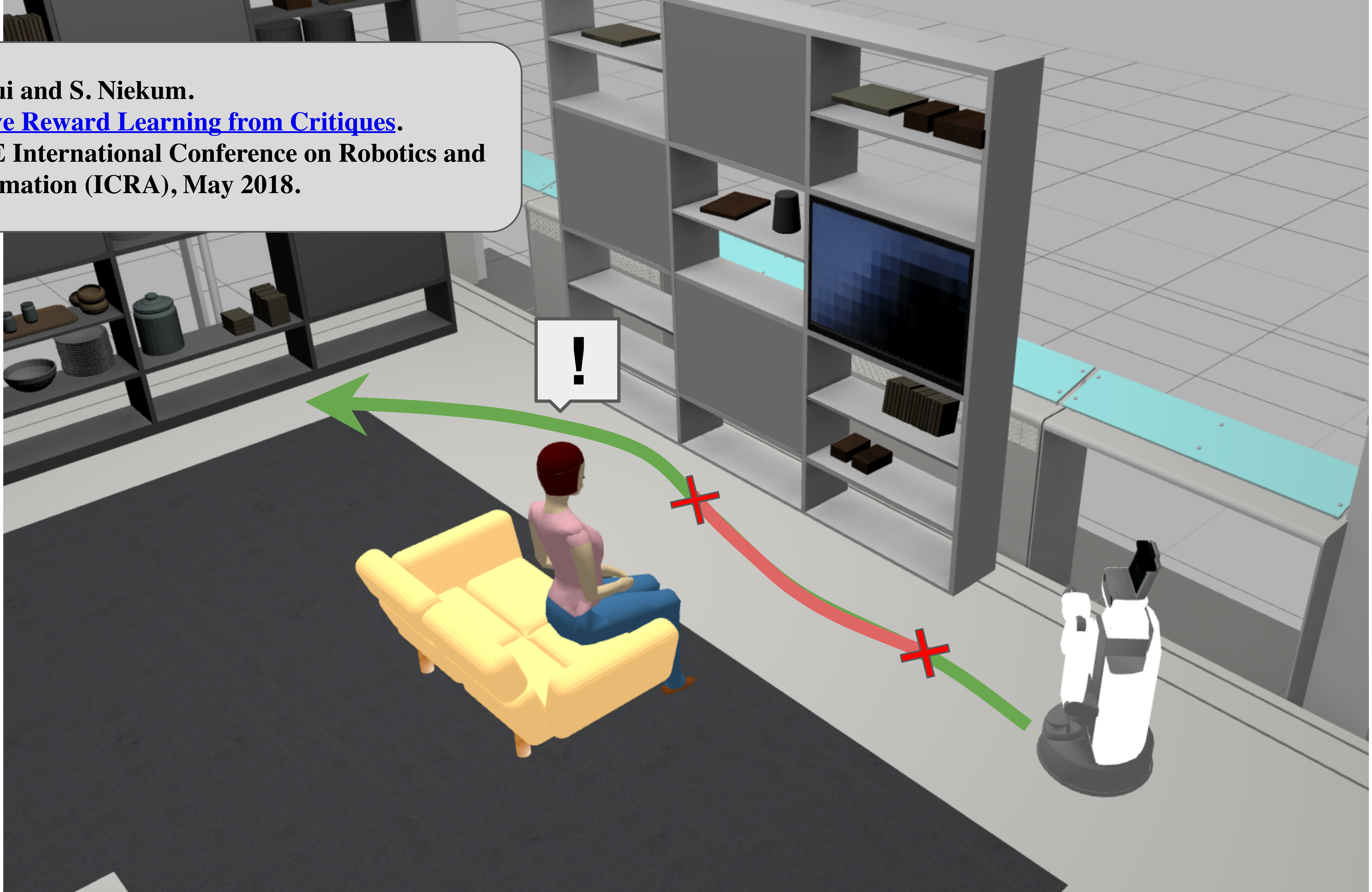
Seeks collisions

Risk-sensitive policy search

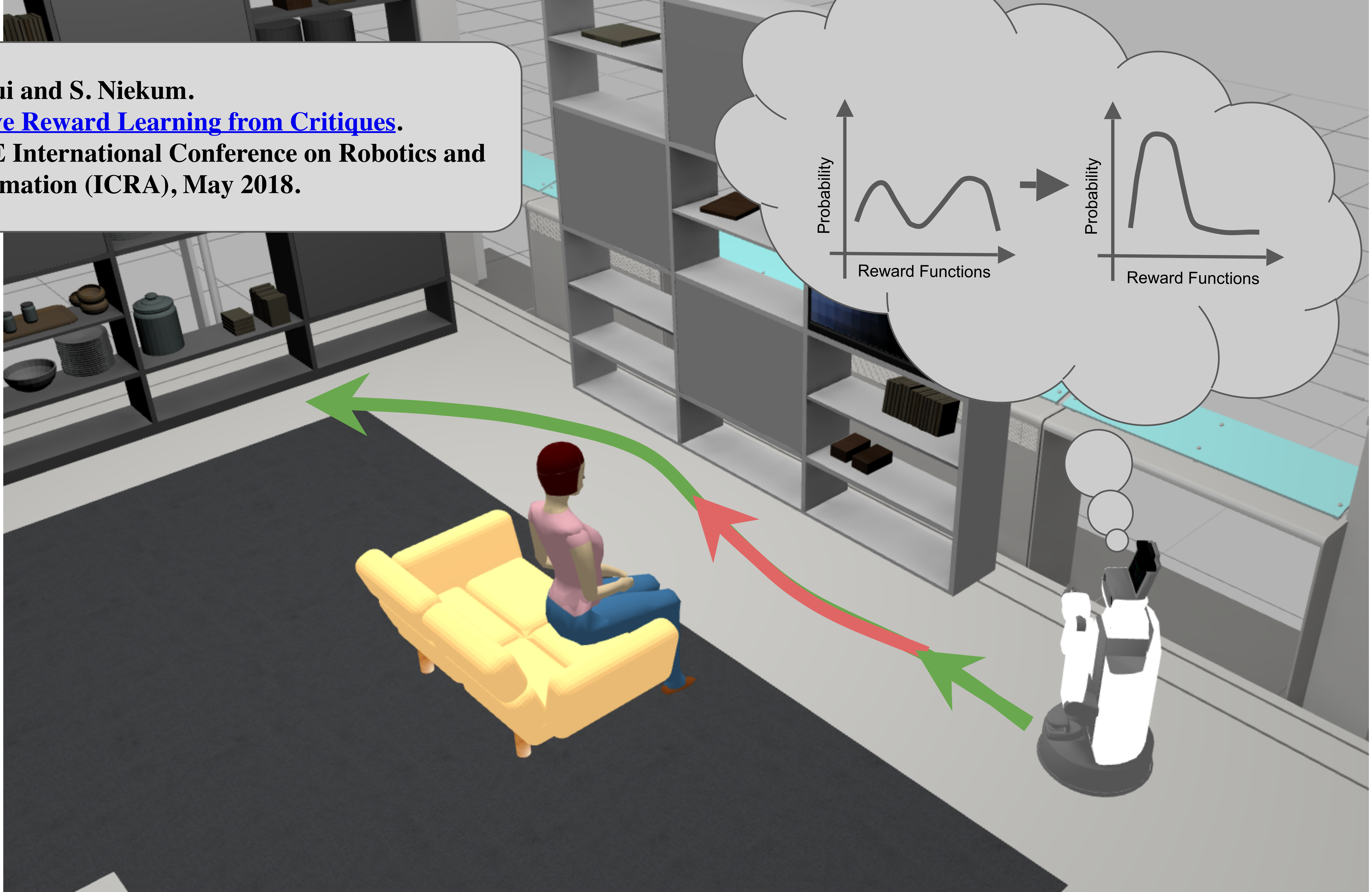
Y. Cui and S. Niekum.
Active Reward Learning from Critiques.
IEEE International Conference on Robotics and Automation (ICRA), May 2018.



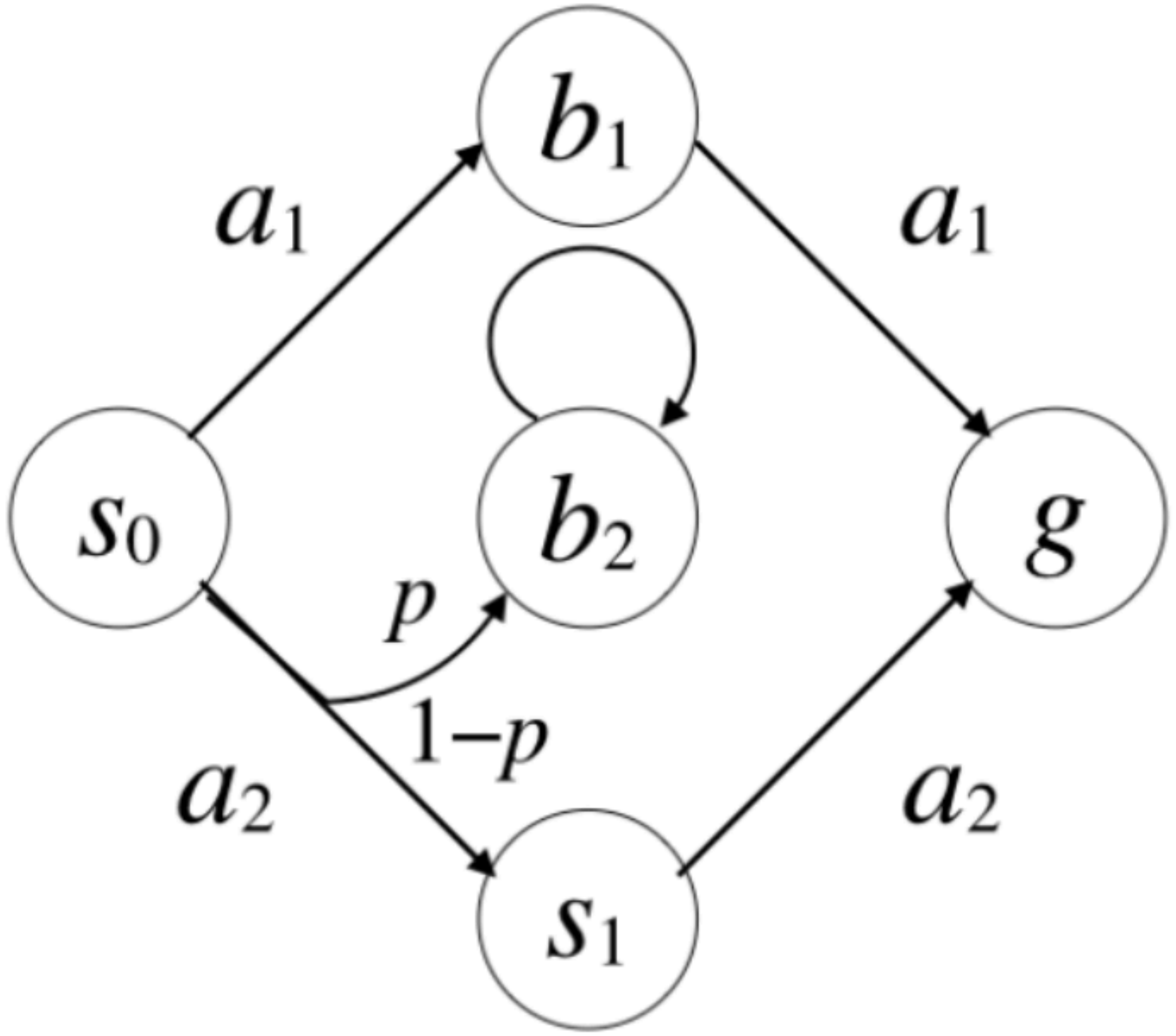
Y. Cui and S. Niekum.
[Active Reward Learning from Critiques](#).
IEEE International Conference on Robotics and Automation (ICRA), May 2018.



Y. Cui and S. Niekum.
[Active Reward Learning from Critiques](#).
IEEE International Conference on Robotics and Automation (ICRA), May 2018.



Aside: Is reward enough?



States b1 and b2 are bad: reward of -r

Desired: maximize the probability of reaching g without hitting a bad state

Always better to take action a2 — seems trivial to specify

Assume gamma = 0.8, assign a value of r to meet specification

p = 0.1 possible, but p = 0.3 impossible!

Littman, Michael L., et al. "Environment-independent task specifications via GLTL." *arXiv preprint arXiv:1704.04341* (2017).