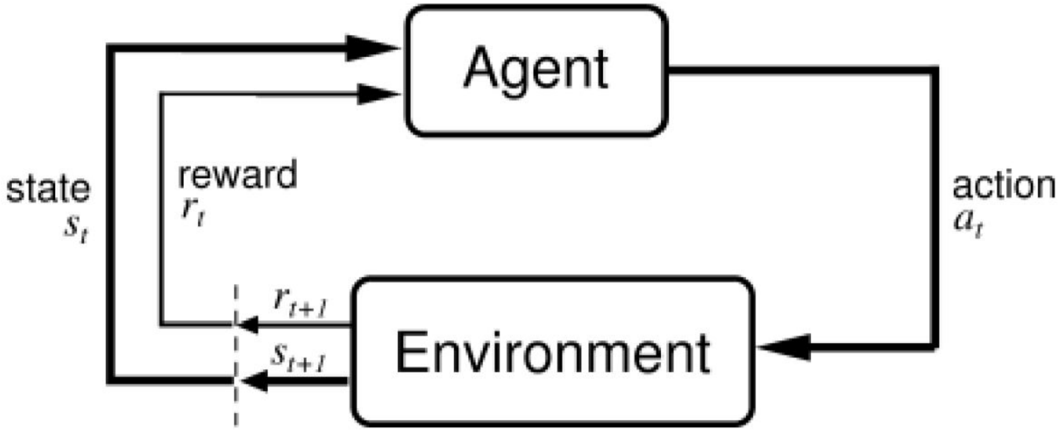
Safety in Sequential Decision Making

Yinlam Chow

Google DeepMind

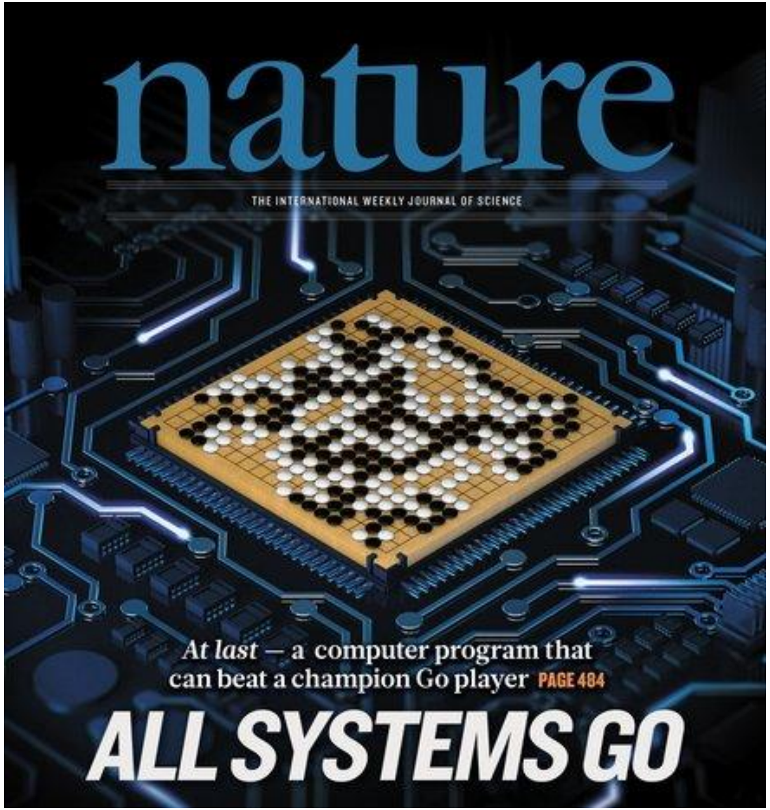
Reinforcement Learning (RL)



source: Sutton & Barto, Reinforcement Learning, 1998

- Combines **machine learning** with **decision making**
- Interaction modeled by a **Markov decision process (MDP)**

Successful Use Cases of RL [V. Minh et al., NIPS 13; D. Silver et al., Nature 17]



Some Challenges to Apply RL in Real-world [A. Irpan 18]

- Noisy data
- Training with insufficient data
- Unknown reward functions
- Robust models w.r.t. Uncertainty?
- Safety guarantees in RL?
- Safe exploration

Some Challenges to Apply RL in Real-world [A. Irpan 18]

- Noisy data
- Training with insufficient data
- Unknown reward functions
- Robust models w.r.t. uncertainty?
- **Safety guarantees in RL?**
- Safe exploration

Safety Problems in RL

Studied the following safety problems:

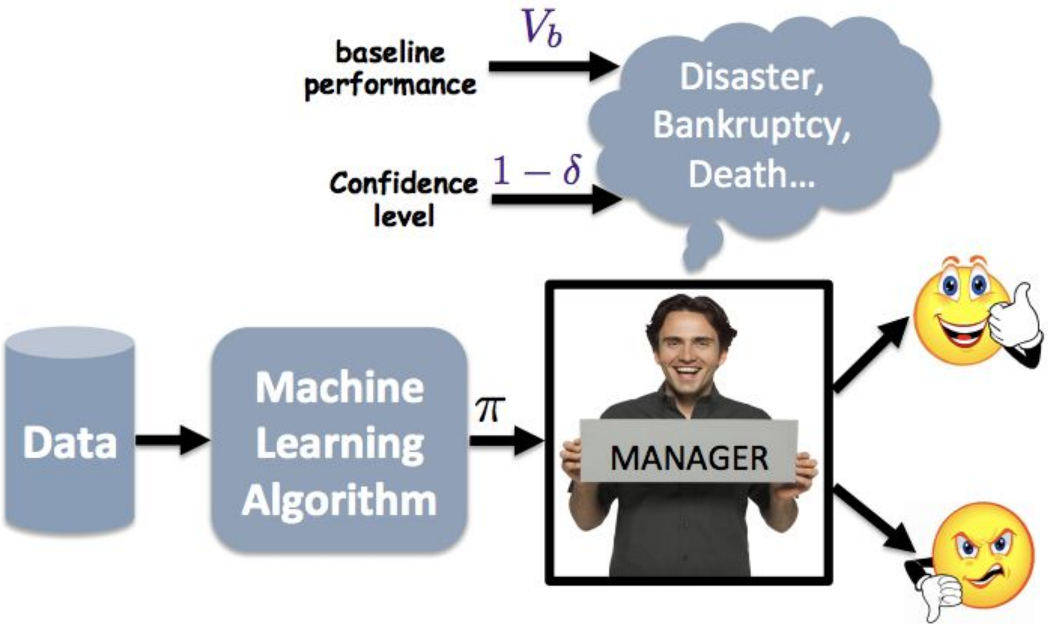
1. Safety w.r.t. **Baseline**
   - Model-free
   - Model-based
   - Apprenticeship learning
2. Safety w.r.t. **Environment Constraints**
3. Robust and **Risk-sensitive** RL (skip here)

Safety Problems in RL

Studied the following safety problems:

1. **Safety w.r.t. Baseline**
    - Model-free
    - Model-based
    - Apprenticeship learning
2. Safety w.r.t. Environment Constraints
3. Robust and Risk-sensitive RL (skip here)

Safety w.r.t. Baseline

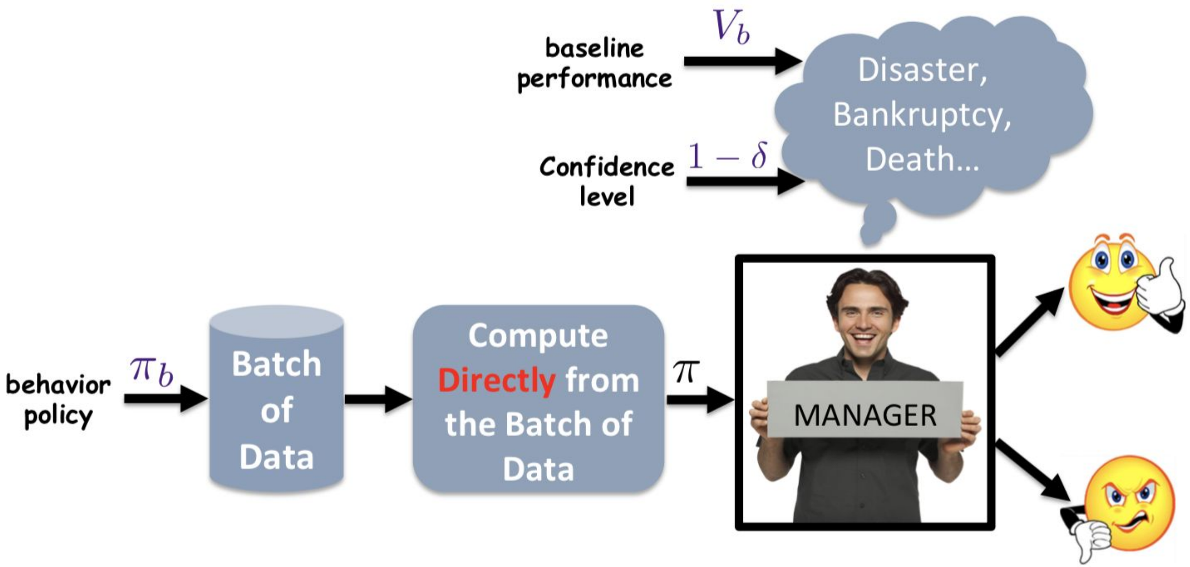


Question:

How to train a RL policy offline that performs no worse than baseline?

Safety w.r.t. Baseline: Model-free Approach

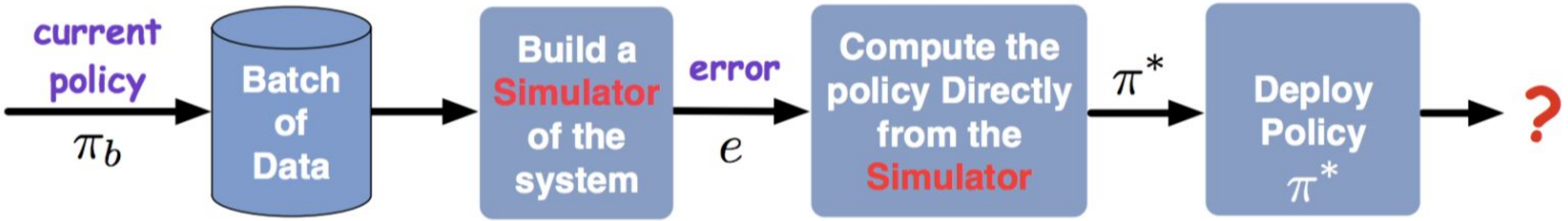
Problem: Safe off-policy optimization [P. Thomas et al., AAAI 15 & ICML 15]



Recent work: More Robust Doubly Robust Off-policy Evaluation
[M. Farajtabar, Y. Chow, M. Ghavamzadeh, ICML 18]

Safety w.r.t. Baseline: Model-based Approach

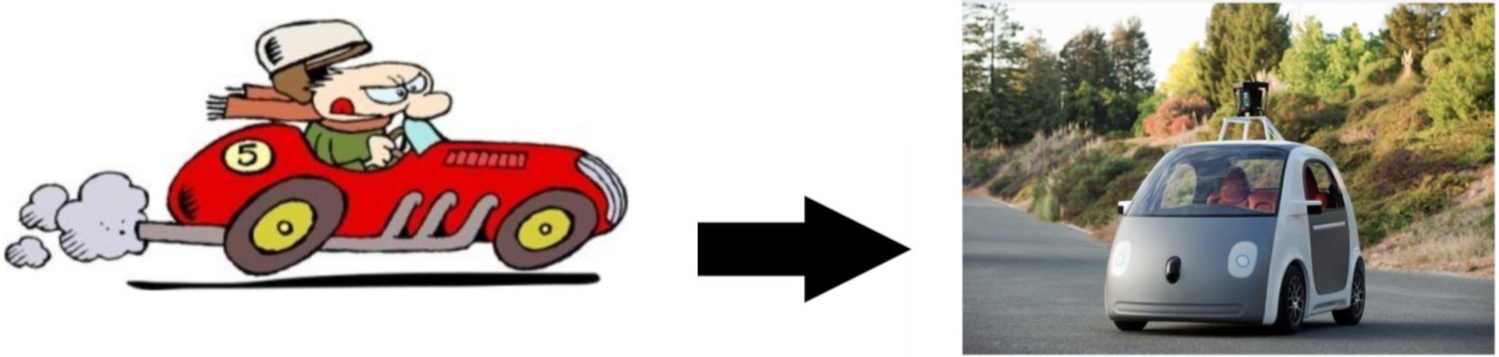
Problem: Sim-to-real with transferable guarantees in performance



Recent work: Safe Policy Improvement by Minimizing Robust Regret
[M. Ghavamzadeh, M. Petrik, Y. Chow, NIPS 16]

Safety and Apprenticeship Learning [Abbeel ICML04]

Problem: Imitate expert but become more **risk-averse?** (Without reward.)



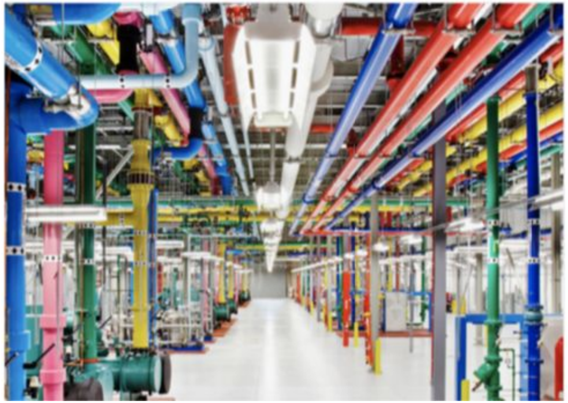
Recent work: Risk-Sensitive Generative Adversarial Imitation Learning
[J. Lacotte, M. Ghavamzadeh, Y. Chow, M. Pavone, UAI workshop 18, AISTATS 19 (submitted)]

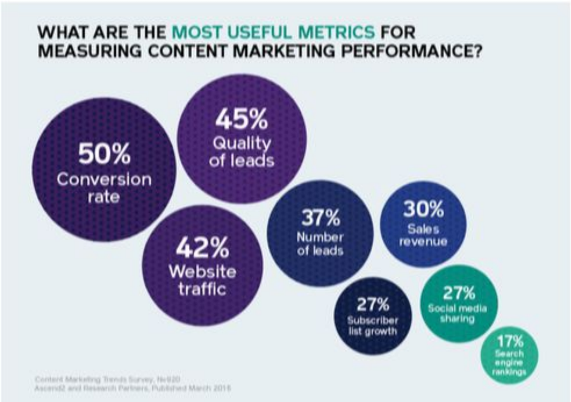
Safety Problems in RL

Studied the following safety problems:

1. Safety w.r.t. Baseline
    - Model-free
    - Model-based
    - Apprenticeship learning
2. **Safety w.r.t. Environment Constraints**
3. Robust and Risk-sensitive RL (skip here)

The Safe Decision Making Problem





Safety definitions directly come from **environment constraints**, e.g.,

- Collision avoidance, speed & fuel limits, traffic rules
- System overheating, quality-of-service guarantees
- User satisfaction in online recommendation

A Lyapunov Approach to Safe RL

Safety w.r.t. Environment Constraints

Collaboration with Ofir Nachum *(Brain)*, Mohammad Ghavamzadeh, Edgar Duenez-Guzman *(DMG)* -- NIPS 2018, ICLR 2019 (submitted)

Overview on Notions of Safety

- **Reachability** *(safety probability)*:

  Safe if agent doesn't *enter* an undesirable region (w.h.p.)

  Application: Robot motion planning

- **Limit Visits to Undesirable States** *(time spent in dangerous regions)*:

  Safe if agent doesn't *stay* in an undesirable region for long

  Applications: System (data center) maintenance

NOTE: Constraints are trajectory-based

Notions of Safety

Two definitions of safety w.r.t. *mission-based* constraints

- Reachability:

  Safe if agent doesn't enter an undesirable region (w.h.p.), i.e.,

  $$\mathbb{P}\left(\exists t \in \{0, 1, \ldots, \mathrm{T}^* - 1\}, x_t \in \mathcal{S}_H \mid x_0, \pi\right) \leq d_0$$

  Application: Robot motion planning

- Limit Visits to Undesirable States:

  Safe if agent doesn't stay in an undesirable region for long, i.e.,

  $$\mathbb{E}\left[\frac{1}{\mathrm{T}^*} \sum_{t=0}^{\mathrm{T}^*-1} \mathbf{1}\{x_t \in \mathcal{S}_H\} \mid x_0, \pi\right] \leq d_0$$

  Applications: System maintenance; Data-center temperature control

General Problem Formulation

Modeled by **Constrained Markov Decision Process (CMDP)**

- Reward: primary return performance
- Constraint cost: model safety constraints

Goals:

1. Find an optimal (feasible) RL agent
2. More restrictive: Guarantee safety during training

Safe RL Formulation

- MDP tuple: $(\mathcal{X}, \mathcal{A}, c, P, x_0)$; CMDP[1] tuple: $(\mathcal{X}, \mathcal{A}, c, d, P, x_0, d_0)$
- Space of stationary Markovian polices Δ with policy element π
- Bellman operator:
  $T_{\pi,h}[V](x) = \sum_a \pi(a|x)\big[h(x,a) + \sum_{x' \in \mathcal{X}} P(x'|x,a)V(x')\big]$
- For reachability constraint, requires <u>state augmentation</u>

Problem $\mathcal{OPT}$: *Given x_0 and d_0, solve*

$$\min_{\pi \in \Delta} \quad \mathcal{C}_\pi(x_0) := \mathbb{E}\left[\sum_{t=0}^{\mathrm{T}^*-1} c(x_t, a_t) \mid x_0, \pi\right]$$

$$s.t. \quad \mathcal{D}_\pi(x_0) := \mathbb{E}\left[\sum_{t=0}^{\mathrm{T}^*-1} d(x_t) \mid x_0, \pi\right] \leq d_0$$

Some Prior Art and Limitations

- **Prior Art:**

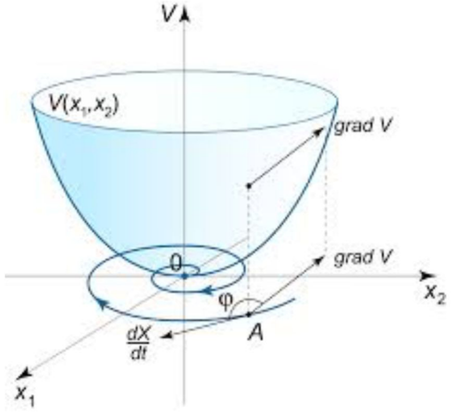
Method	Summary	Pros	Cons
Dual Method	LP in dual space	Exact	Computationally expensive $O(\vert\mathcal{X}\vert^3\vert\mathcal{A}\vert^3)$
Lagrangian	Iterative method to find saddle point	Asymptotically exact	Premature stopping; Unsafe at iterations
State-wise Surrogate	State-wise constraint surrogate	Safe at iterations	Super conservative
Lexicographical Surrogate	Global Lyapunov based safety set	Safe at iterations	Conservative

- **Contributions of the Lyapunov-based method:**
  - **Safety** during training
  - **Scalable** model-free RL (on-policy/off-policy); **Less conservative** policies

Lyapunov Function and Safety

Safety verification:

Given a policy π, find a Lyapunov function L_π that satisfies

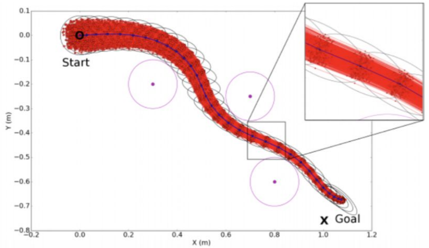


- $L_\pi : \mathcal{X} \to \mathbb{R}_{\geq 0}$
- $L_\pi(x_0) \leq d_0$
- $L_\pi(x) \geq d(x) + \mathbb{E}_{x \sim P^\pi}[L_\pi(x')], \forall x \in \mathcal{X}$
  (Lyapunov function decreases as the state evolves from x to x' under dynamics P^π)

Safe policy search:

Given a Lyapunov function L, consider the "safety-tube" Markovian policy set

$$\mathcal{F}_L(x) = \{\pi(\cdot|x) \in \Delta : T_{\pi,d}[L](x) \leq L(x)\}$$



Safe Policy Iteration

CMDP Formulation

$$\min_{\pi} \mathbb{E}\left[\sum_{t=0}^{T-1} c(x_t, a_t) \mid x_0, \pi\right] \quad , \quad \text{s.t.} \quad \mathbb{E}\left[\sum_{t=0}^{T-1} d(x_t) \mid x_0, \pi\right] \le d_0$$

Safe Policy Iteration (SPI)

1. finding the Lyapunov function

$$\max_{\epsilon:\mathcal{X}\to\mathbb{R}^+} ||\epsilon||_1 \quad , \quad \text{s.t.} \quad \mathcal{T}_{d+\epsilon}^{\pi_k}[L_k](x) = L_k(x),\ \forall x \in \mathcal{X} \quad , \quad L_k(x_0) \le d_0$$

$$L_k(x) = V_{d+\epsilon}^{\pi_k}(x),\ \forall x \in \mathcal{X}$$

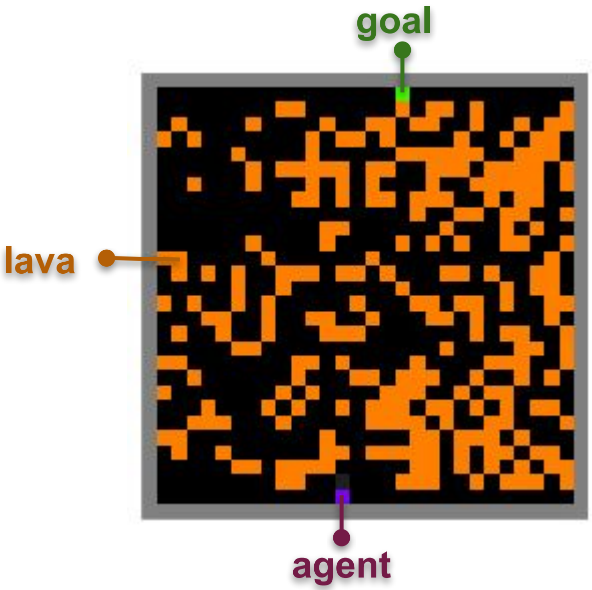
2. policy evaluation $\quad V_k = V_c^{\pi_k}$
3. policy improvement $\quad \pi_{k+1} \in \arg\min_{\pi \in \mathcal{F}_{L_k}(x)} \mathcal{T}_c^{\pi}[V_k]$

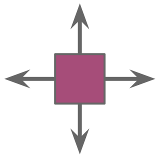
$$\mathcal{F}_{L_k}(x) = \left\{\pi(\cdot|x) \mid \mathcal{T}_d^{\pi}[L_k](x) \le L_k(x)\right\}$$

(a) all π_k's are safe, **(b)** π_{k+1} is no worse than π_k, **(c)** SPI converges

Environment with Discrete Actions

- Stochastic 2D GridWorld
- Stage-wise cost is 1 for fuel usage
- Goal reward is 1000
- Incurs a cost of 1 in lava; Tolerate at most <u>5 touches</u>
- <u>Experiments</u>: (i) Planning, (ii) RL (explicit position or image obs.)

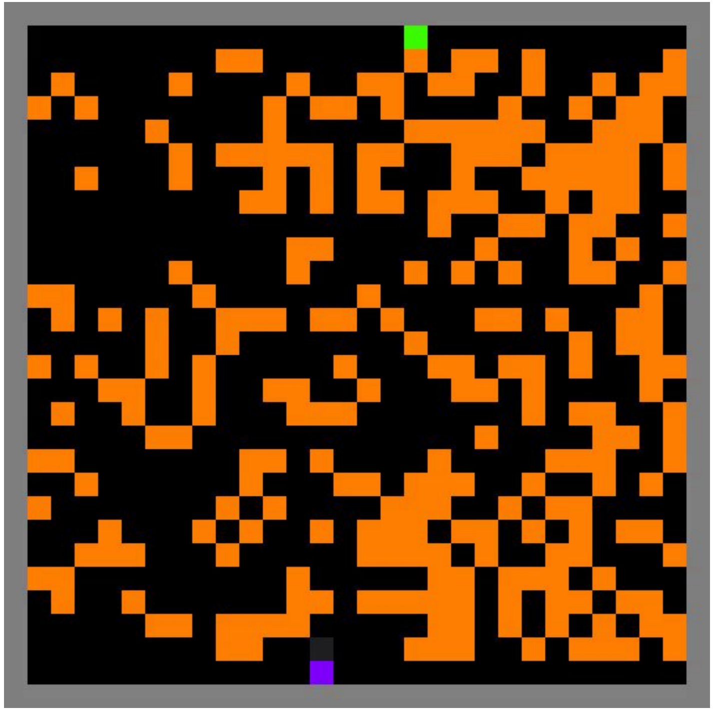




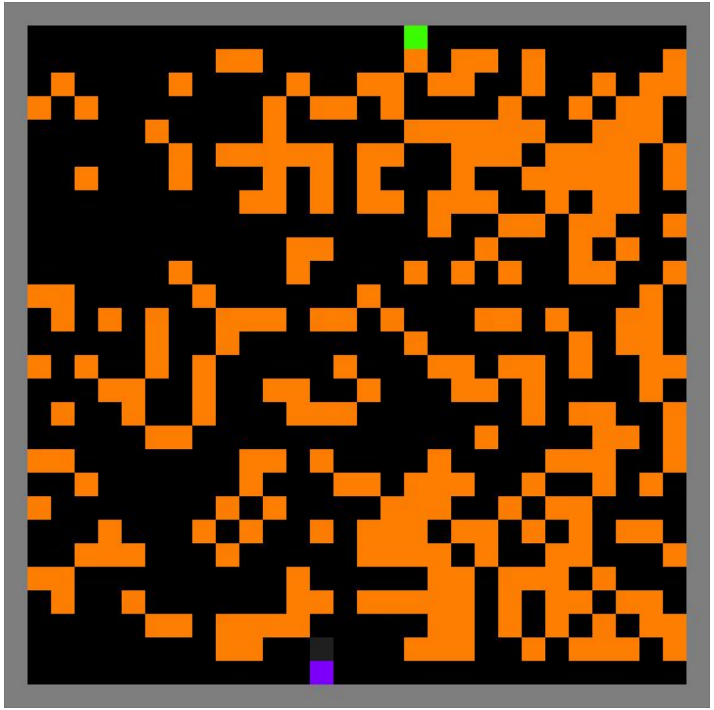
agent actions
(with ε-error)

Safe Planning with Discrete Actions

Lyapunov-based (Our Method)

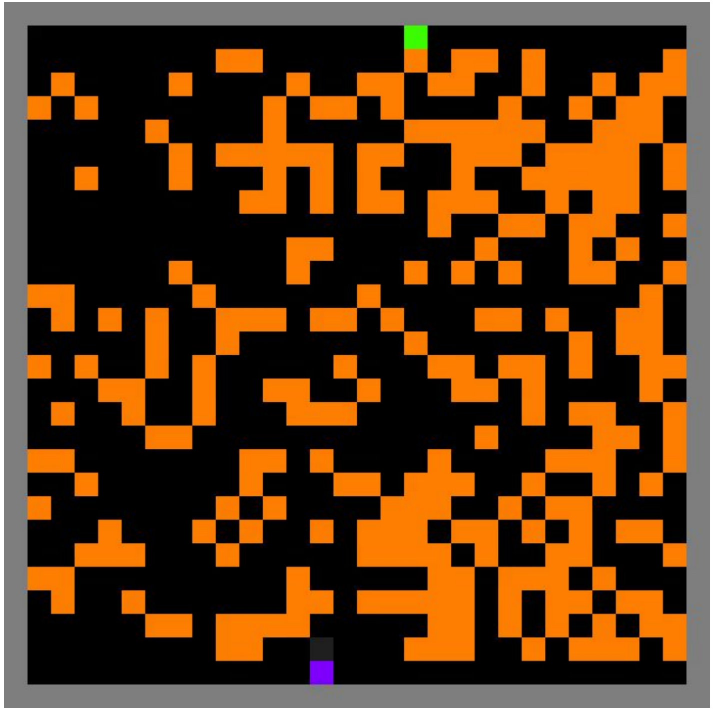


Dual-LP (Exact But Expensive)

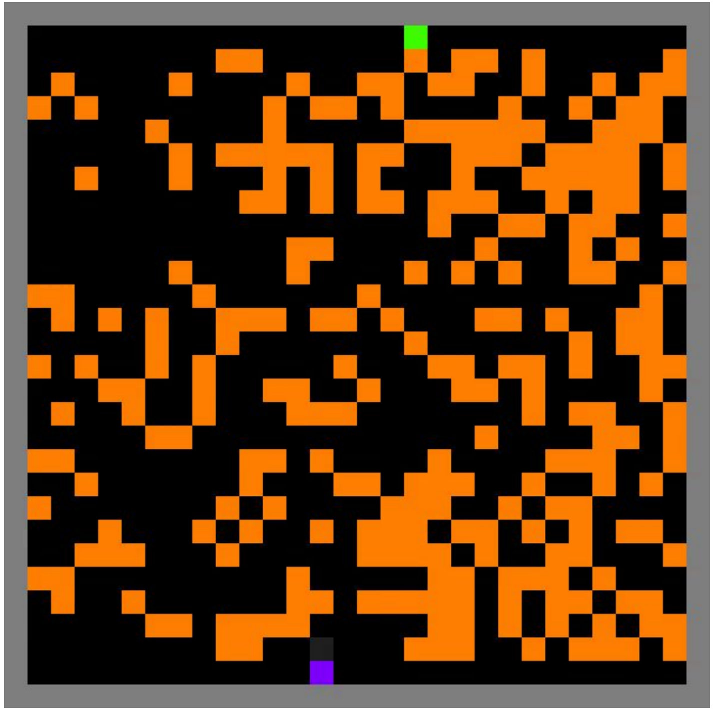


Safe Planning with Discrete Actions

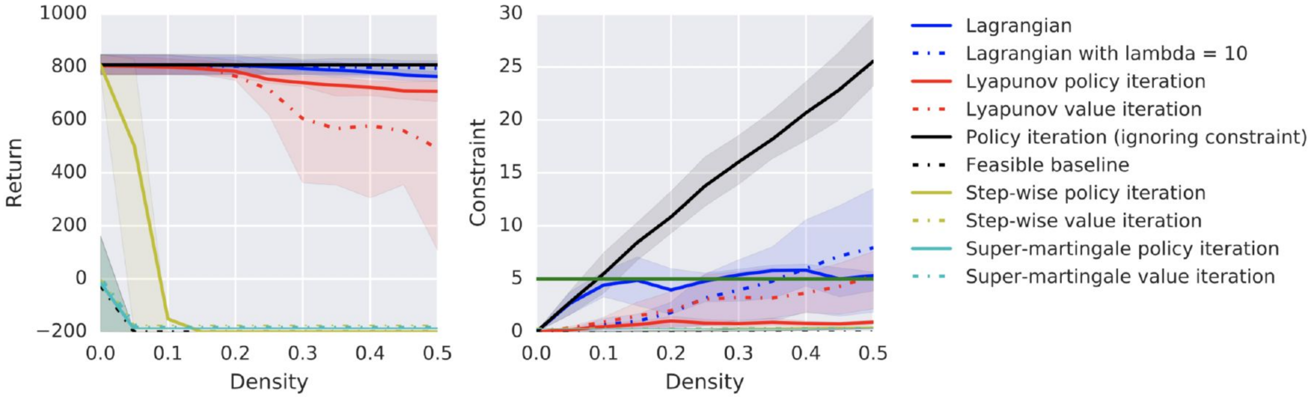
Lagrangian-based (Baseline)



Unconstrained (Baseline)



Exp. 1: 2D Grid-world Planning



- Shaded regions indicate the 80% confidence intervals
- Policies from SPI and SVI are safe and have good performance

From SPI/SVI to Safe Value-based RL

1. Rewrite the inner optimization problem:

$$\pi'(\cdot|x) \in \arg\min_{\pi \in \Delta} \left\{ \pi(\cdot|x)^\top Q(x,\cdot) : (\pi(\cdot|x) - \pi_b(\cdot|x))^\top Q_L(x,\cdot) \le \widetilde{\epsilon}'(x) \right\}$$

   Lyapunov Q-fun: $Q_L(x,a) = d(x) + \widetilde{\epsilon}'(x) + \sum_{x'} P(x'|x,a) L_{\widetilde{\epsilon}'}(x')$

2. Value functions: learn value networks $(\widehat{Q}, \widehat{Q}_D, \widehat{Q}_T)$, and update Lyapunov Q-fun $\widehat{Q}_L(x,a;\theta_D,\theta_T) = \widehat{Q}_D(x,a;\theta_D) + \widetilde{\epsilon}' \cdot \widehat{Q}_T(x,a;\theta_T)$

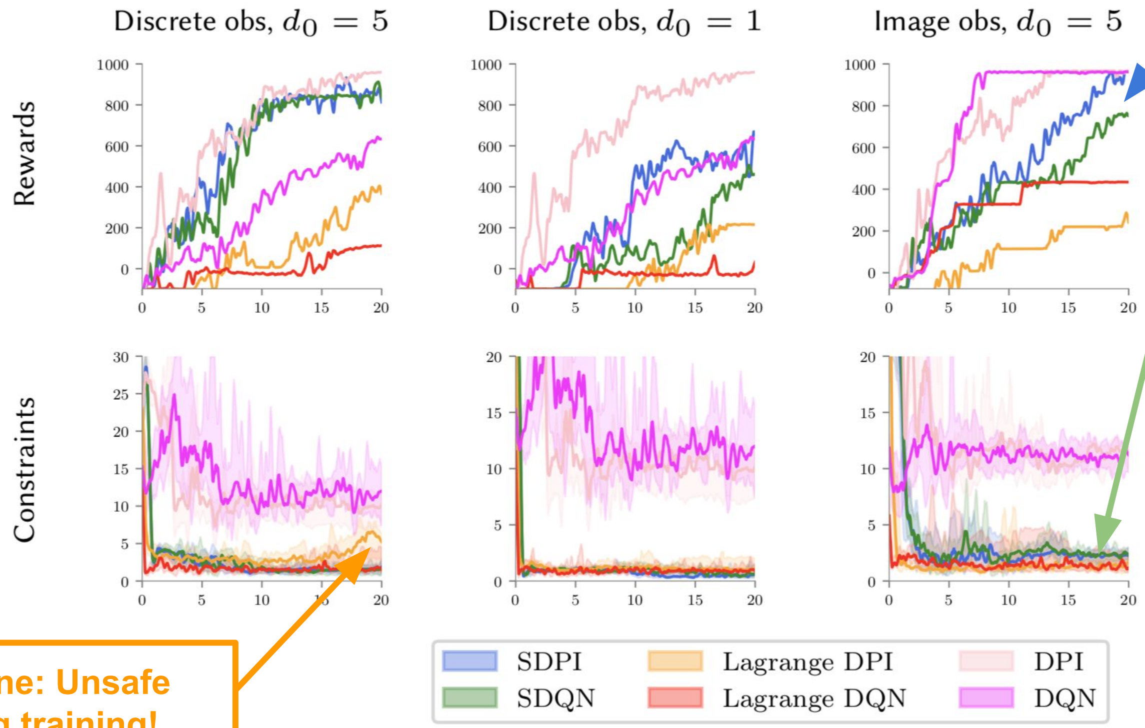
$$\widetilde{\epsilon}'(x) = \frac{(d_0 - \pi_k(\cdot|x_0)^\top \widehat{Q}_D(x_0,\cdot;\theta_D))}{\pi_k(\cdot|x_0)^\top \widehat{Q}_T(x_0,\cdot;\theta_T)}$$

3. Policy updates: LP + policy distillation (of π')

4. More tricks: Policy α-mixing, Replay buffer, Entropy regularization, ...

Safe DQN/DPI with Discrete Actions

Our Methods: Balance performance and safety



Baseline: Unsafe during training!

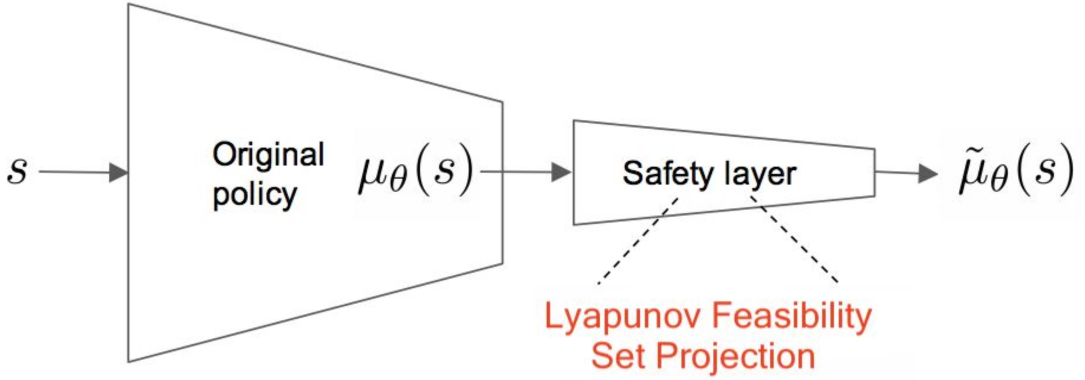
Unconstrained: Always unsafe!

Safe Policy Gradient (Recent Extension)

To handle safety in RL (policy gradient) with <u>continuous actions</u>:

1. **Constrained optimization** w.r.t. policy parameter (CPO)
2. Embed Lyapunov constraint into policy network via **network augmentation**

   (see [Dalal et al. 2018] for simpler setting)



Safe Policy Gradient

1. CPO update with empirical Lyapunov-based constraints:

$$\theta \in \underset{\theta \in \Theta}{\operatorname{argmin}} \quad \langle(\theta - \theta_B), \overbrace{\nabla_\theta \mathbb{E}_{x \sim d_{\theta_B}, a \sim \pi_\theta}\left[Q_{\theta_B}(x,a)\right] |_{\theta=\theta_B}}^{\nabla_\theta \mathcal{C}_{\pi_\theta}(x_0)|_{\theta=\theta_B}}\rangle$$

$$\text{s.t.} \quad \frac{1}{2}\langle(\theta - \theta_B), \nabla^2_\theta D_{\mathsf{KL}}(\theta||\theta_B)\ |_{\theta=\theta_B} \cdot (\theta - \theta_B)\rangle \leq \delta$$

$$\left\langle(\theta - \theta_B), \mathbb{E}_{x \sim \mu_{\theta_B}}\left[\nabla_\theta\ \mathbb{E}_{a \sim \pi_\theta}\left[Q_{L_{\theta_B}}(x,a)\right] |_{\theta=\theta_B}\right]\right\rangle \leq \mathbb{E}_{x \sim \mu_{\theta_B}}\left[\widetilde{\epsilon}(x)\right]$$

2. Safe action mapping from safety layer, at any state $x \in \mathcal{X}$:

$$a^*(x) \in \underset{a}{\operatorname{argmin}} \left\{ \frac{1}{2}\|a - \pi_{\theta,\mathsf{unc}}(x)\|^2 : (a - \pi_{\theta_B}(x))^\top \nabla_a Q_{L_{\theta_B}}(x,a)\ |_{a=\pi_{\theta_B}(x)} \leq \widetilde{\epsilon}(x) \right\}$$

where policy $\pi_{\theta,\mathsf{unc}}$ is computed by unconstrained RL

Safe RL for Continuous Control

Objective: Train a Mujoco HalfCheetah to run stably.

Standard method: DDPG/PPO

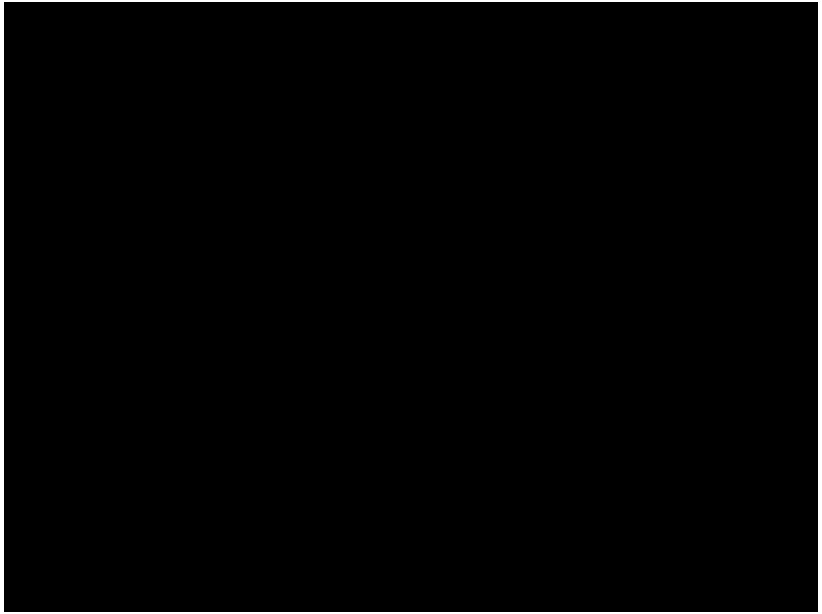
Issue: Unstable if runs too fast!

Remedy: Soft constraint torque @ joints

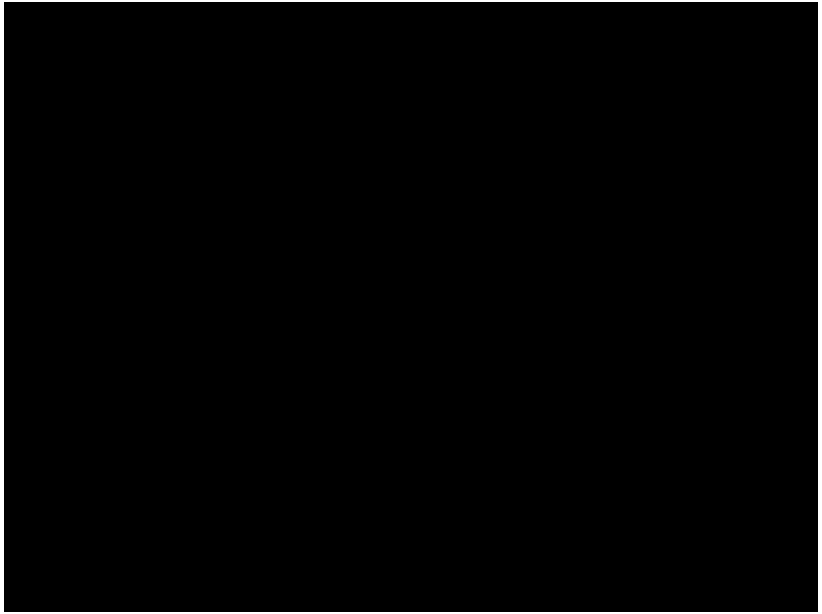
Safe if total torque violation is bounded (d0=50)

Lagrangian-PPO versus Lyapunov-PPO

Lagrangian PPO (Baseline)

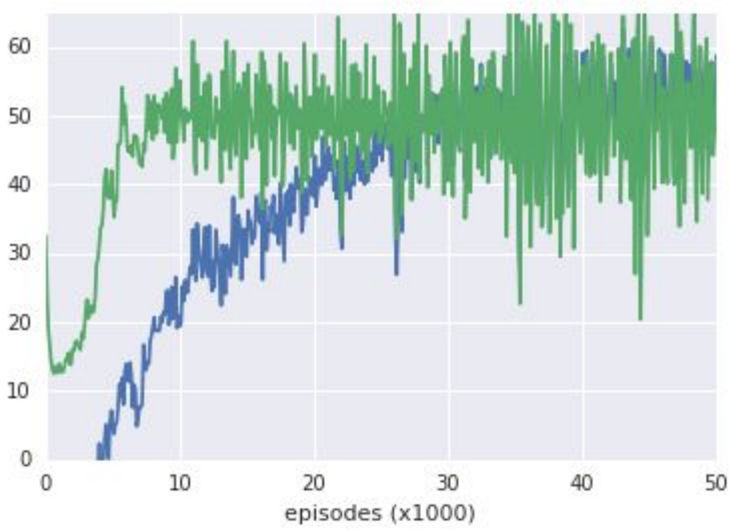


Lyapunov PPO (Our Method)

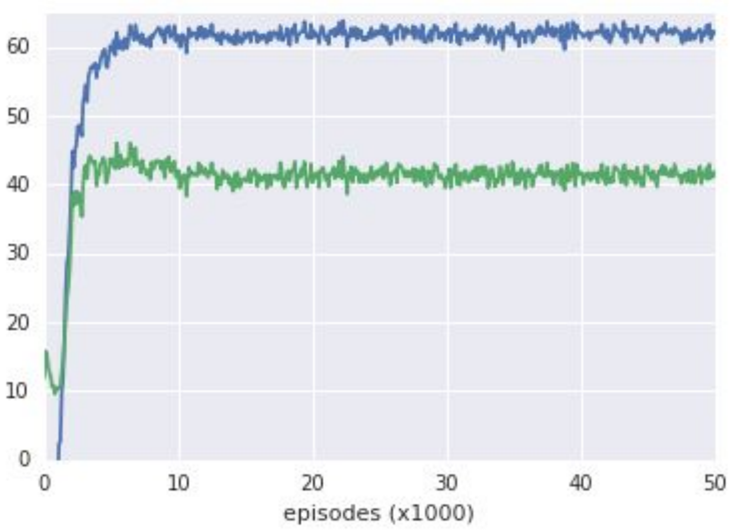


Lagrangian-PPO versus Lyapunov-PPO

Lagrangian PPO (Baseline)



Lyapunov PPO (Our Method)



Conclusion

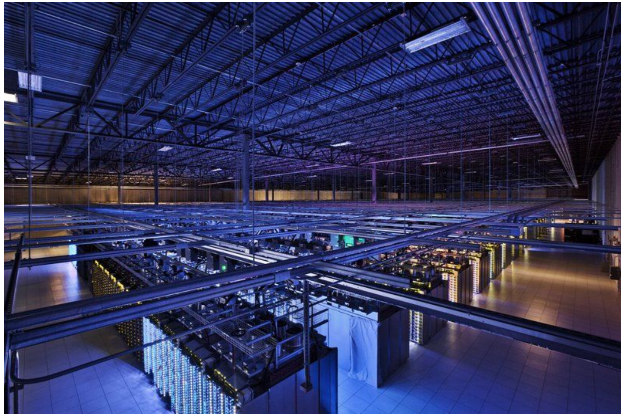
- **Contributions:**
  a. Formulated safety problems as CMDPs
  b. Proposed a Lyapunov-based safe RL methodology
     - Work well for on-policy/off-policy settings
     - Applicable to value-based and policy-based algorithms
  c. Guarantee* safety during training

- **Limitations:**
  a. Provably-optimal only under restricted conditions!
  b. Theoretically justified in MDPs, but not with function approximations (RL)!
     - Future work in **model-based setting**

Current Work

In talks with the following projects at Google & DeepMind:

- PRM-RL for indoor robot navigation (Brain Robotics)
- FineTuner/TF.agents (Brain RL/rSWE)
- DrData (DeepMind)



Acknowledgements

- General:
  - Mohammad Ghavamzadeh (FAIR)
- Safety w.r.t. Baseline:
  - Mehrdad Farajtabar (DMG)
  - Marek Petrik (UNH)
  - Jonathan Lacotte, Marco Pavone (Stanford)
- Safety w.r.t. Environment Constraint:
  - Ofir Nachum (Brain)
  - Edgar Guzman Duenez (DMG)
  - Aleksandra Faust, Jasmine Hsu, Vikas Sindhwani (Brain Robotics)

Appendix of Lyapunov Work

Theoretical Results

- **Observation:** Policies π in $\mathcal{F}_L$ are *safe* iff there exists a Lyapunov function $L \in \mathcal{L}_\pi(x_0, d_0)$.
- **Question:** How to find a "good" Lyapunov function L^*?
- **In theory:**
  1. Assume access to a baseline feasible policy $\pi_b \in \Delta$
  2. Finding L is equivalent to cost-shaping, i.e., $L_\epsilon(x) = \mathbb{E}\left[\sum_{t=0}^{\mathrm{T}^*-1} d(x_t) + \epsilon(x_t) \mid \pi_b, x\right]$ with auxiliary ϵ
  3. Set $\epsilon^*(x) := 2\overline{\mathrm{T}} \cdot D_{\max} D_{TV}(\pi^* || \pi_b)(x)$; If π_b is close to π^*, or is "very feasible" [2], then $\mathcal{F}_{L_{\epsilon^*}}$ contains π^*
  4. Solve for an optimal policy by DP w.r.t. Bellman operator $\min_{\pi \in \mathcal{F}_{L_{\epsilon^*}}(\cdot)} T_{\pi,c}[V](\cdot)$

[2] Exact condition:

$$\max_{x \in \mathcal{X}} \epsilon^*(x) \le D_{\max} \cdot \min \left\{ \frac{d_0 - \mathcal{D}_{\pi_b}(x_0)}{\overline{\mathrm{T}} D_{\max}}, \frac{\overline{\mathrm{T}} D_{\max} - \overline{\mathcal{D}}}{\overline{\mathrm{T}} D_{\max} + \overline{\mathcal{D}}} \right\}$$

Safe Policy Iteration

- **Challenge:** Compute ϵ^* requires estimating $D_{TV}(\pi^*||\pi_b)$!
- **Remedy:** Approximate ϵ^* via bootstrapping, i.e., start with a π_b, solve LP for $\widetilde{\epsilon}$:

$$\widetilde{\epsilon} \in \arg\max_{\epsilon:\mathcal{X}\to\mathbb{R}_{\geq 0}} \left\{ \sum_{x\in\mathcal{X}} \epsilon(x) : d_0 - \mathcal{D}_{\pi_b}(x_0) \geq \mathbf{1}(x_0)^\top (I - \{P(x'|x,\pi_b)\}_{x,x'\in\mathcal{X}})^{-1}\epsilon \right\}$$

  and improve π_b

- **Intuition:** Larger ϵ implies larger $\mathcal{F}_{L_\epsilon}$; Find *largest* $\widetilde{\epsilon}$ that satisfies *Lyapunov conditions*
- **Closed-form:** $\widetilde{\epsilon}$ has the following form:

$$\widetilde{\epsilon}(x) = \frac{(d_0 - \mathcal{D}_{\pi_b}(x_0)) \cdot \mathbf{1}\{x = \underline{x}\}}{\mathbb{E}[\sum_{t=0}^{\mathrm{T}^*-1} \mathbf{1}\{x_t = \underline{x}\} \mid x_0, \pi_b]} \geq 0, \quad \forall x \in \mathcal{X}$$

Safe Policy Iteration (Cont'd)

For $k \in \{0, 1, \ldots, \}$:

1. With $\pi_b = \pi_k$, calculate L_{ϵ_k} via LP
2. Evaluate the cost value $V_{\pi_k}(\cdot) = \mathcal{C}_{\pi_k}(\cdot)$
3. Policy improvement: $\pi_{k+1} \in \operatorname{argmin}_{\pi \in \mathcal{F}_{L_{\epsilon_k}}(\cdot)} T_{\pi,c}[V_{\pi_k}](\cdot)$

Properties:

- Consistent feasibility
- Step-wise policy improvement
- Asymptotic convergence
- Computational complexity $O(K|\mathcal{X}||\mathcal{A}|^3 + K|\mathcal{X}|^2|\mathcal{A}|^2)$, which in practice it is much lower than exact solvers ($O(|\mathcal{X}|^3|\mathcal{A}|^3)$)

Highlights of Other Recent Work

1. Robust and Controllable Representation Learning

Stochastic Control for Non-linear System

$$\min_{\mu_t : \mathbf{s}_t \to \mathbf{u}_t,\, \forall t} \mathbb{E}_{\mathbf{n}^{\mathcal{S}}, \mu_t, \forall t} \left[\sum_{t=1}^{T} \left((\mathbf{s}_t - \mathbf{s}^f)^\top \mathbf{Q} (\mathbf{s}_t - \mathbf{s}^f) + \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t \right) \right]$$

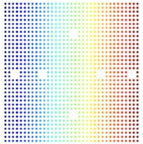
$$\mathbf{s}_{t+1} = f_{\mathcal{S}}(\mathbf{s}_t, \mathbf{u}_t) + \mathbf{n}^{\mathcal{S}}, \quad \mathbf{s}_t \in \mathbb{R}^{n_s}, \quad \mathbf{n}^{\mathcal{S}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{n}^{\mathcal{S}}})$$

Common Approach: Iterative LQR (iLQR) algorithm

But what if:

- Model $f_{\mathcal{S}}$ is ***unknown***
- Instead of $\mathbf{s}_t$, we observe high-dim sensory data ***(e.g., images)***
  $\mathbf{x}_t \in \mathbb{R}^{n_x}, \quad n_x \gg n_s$

State Space



$n_s = 2$

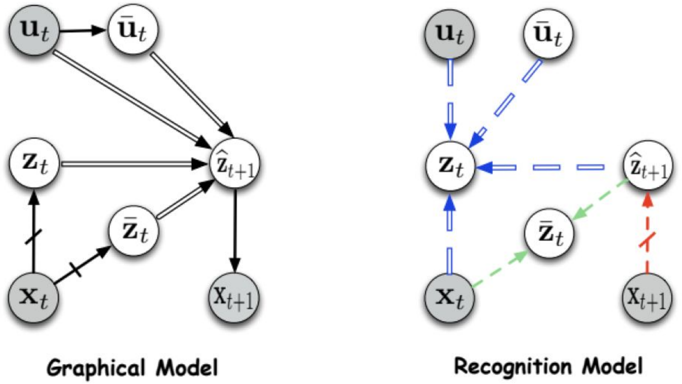
Observation

$n_x = 40 \times 40$ image

How to do model-based control in visual-servoing or perception-to-control?

RCE [Ershad et. al 2018]: Graphical model with bottleneck latent state and linear dynamics

Goal: $\max_\theta \log p_\theta(x_{t+1}|x_t, u_t)$



Graphical Model: $p(x_{t+1}, z_t, \overline{z}_t, \overline{u}_t, \widehat{z}_{t+1}|x_t, u_t) = p(z_t|x_t) \cdot p(\overline{z}_t|x_t) \cdot$
$p(u_t|\overline{u}_t) \cdot \mathbf{1}\{\widehat{z}_{t+1} = A_t(\overline{z}_t, \overline{u}_t)z_t + B_t(\overline{z}_t, \overline{u}_t)u_t + c_t(\overline{z}_t, \overline{u}_t)\} \cdot p(x_{t+1}|\widehat{z}_{t+1})$

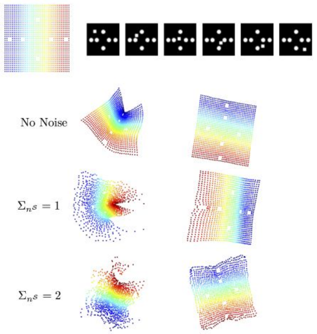
Recognition Model: $q(z_t, \overline{z}_t, \overline{u}_t, \widehat{z}_{t+1}|x_t, x_{t+1}, u_t) = q(\widehat{z}_{t+1}|x_{t+1}) \cdot$
$q(\overline{u}_t|u_t) \cdot q(\overline{z}_t|x_t, \widehat{z}_{t+1}) \cdot \mathbf{1}\{z_t = A_t^{-1}(\overline{z}_t, \overline{u}_t)\,(\widehat{z}_{t+1} - B_t(\overline{z}_t, \overline{u}_t)u_t - c_t(\overline{z}_t, \overline{u}_t))\}$

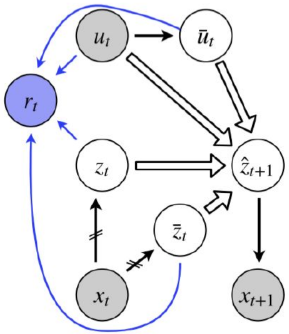
Variational ELBO $\implies p(x_{t+1}, z_t, \overline{z}_t, \overline{u}_t, \widehat{z}_{t+1}|x_t, u_t) \approx q(z_t, \overline{z}_t, \overline{u}_t, \widehat{z}_{t+1}|x_t, x_{t+1}, u_t)$

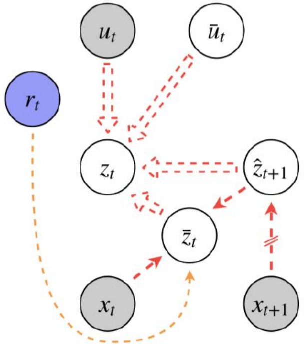
- **Noisy RCE**: Add noise models to the components of the linear model (A_t, B_t, c_t) and solve <u>stochastic DDP</u> [Theodorou et al. 2010]

$$
\begin{aligned}
\widehat{z}_{t+1} \approx& A_t(\overline{z}_t, \overline{u}_t, \xi_t) z_t + B_t(\overline{z}_t, \overline{u}_t, \xi_t) u_t + c_t(\overline{z}_t, \overline{u}_t, \xi_t) \\
A_t(\overline{z}_t, \overline{u}_t, \xi_t) =& A_t(\overline{z}_t, \overline{u}_t) + A_{\xi_t}(\overline{z}_t, \overline{u}_t), \quad A_{\xi,t}(\overline{z}_t, \overline{u}_t) \sim \mathcal{N}(0, I) \\
B_t(\overline{z}_t, \overline{u}_t, \xi_t) =& B_t(\overline{z}_t, \overline{u}_t) + B_{\xi_t}(\overline{z}_t, \overline{u}_t), \quad B_{\xi,t}(\overline{z}_t, \overline{u}_t) \sim \mathcal{N}(0, I) \\
c_t(\overline{z}_t, \overline{u}_t, \xi_t) =& c_t(\overline{z}_t, \overline{u}_t) + c_{\xi_t}(\overline{z}_t, \overline{u}_t), \quad c_{\xi,t}(\overline{z}_t, \overline{u}_t) \sim \mathcal{N}(0, I)
\end{aligned}
$$

- **Task-dependent RCE**: Bringing the control objective into the latent space loss function (through the reward function $r(z_t, u_t)$)







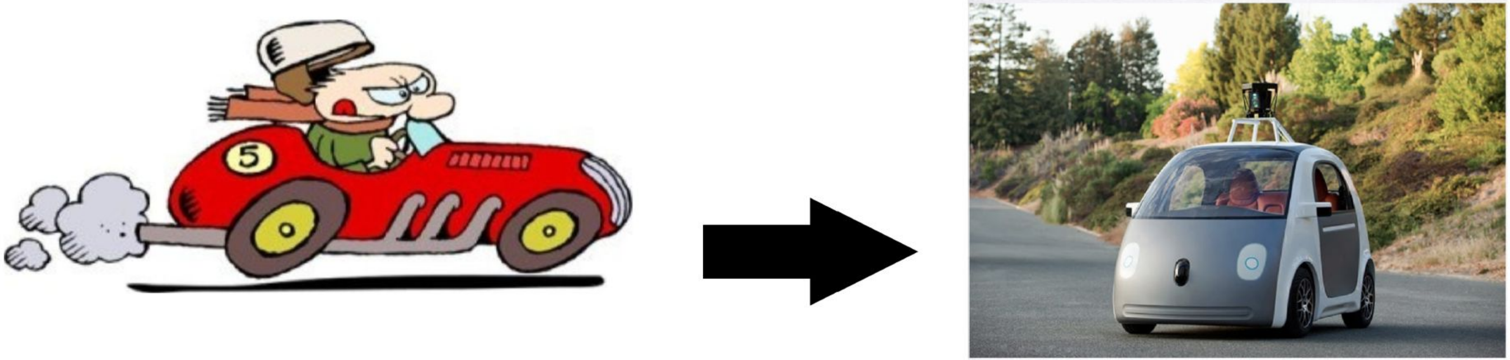
- **Active-sensing**: End-to-end training on policy and models

2. Risk-Sensitive Imitation Learning

Question: How to train a policy that mimics an expert in terms of mean performance, yet being more risk-averse?

Examples:

- Use history from young drivers to train an autonomous car for seniors
- Use data from a hedge fund to train a personalized trading strategy



Question: How to learn a policy π that performs no worse than π_E?

(Risk-sensitive) Imitation learning: Without knowing the exact cost, consider the cost uncertainty set $\mathcal{C} = \{f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}\}$, and

$$\min_{\pi} \sup_{f \in \mathcal{C}} \mathbb{E}[C_f^{\pi}] - \mathbb{E}[C_f^{\pi_E}] + \lambda(\rho_\alpha[C_f^{\pi}] - \rho_\alpha[C_f^{\pi_E}]),$$

where C_f^{π} is the loss of policy π w.r.t. the cost function f

Reformulation to two risk-sensitive GAIL algorithms:

- Risk-profile matching of $\mathcal{D}_\xi^{\pi_E}$ and $\mathcal{D}_\xi^{\pi}$ w.r.t. JS distance

$$\min_{\pi} \; -H(\pi) + (1+\lambda) \sup_{d \in \mathcal{D}_\xi^{\pi}} \inf_{d' \in \mathcal{D}_\xi^{\pi_E}} D_{\text{JS}}(d, d')$$

- Risk-sensitive Wasserstein GAN:

$$\min_{\pi} \; -H(\pi) + (1+\lambda) \sup_{f \in \mathcal{F}_1} \rho_\alpha^\lambda[C_f^\pi] - \rho_\alpha^\lambda[C_f^{\pi_E}]$$

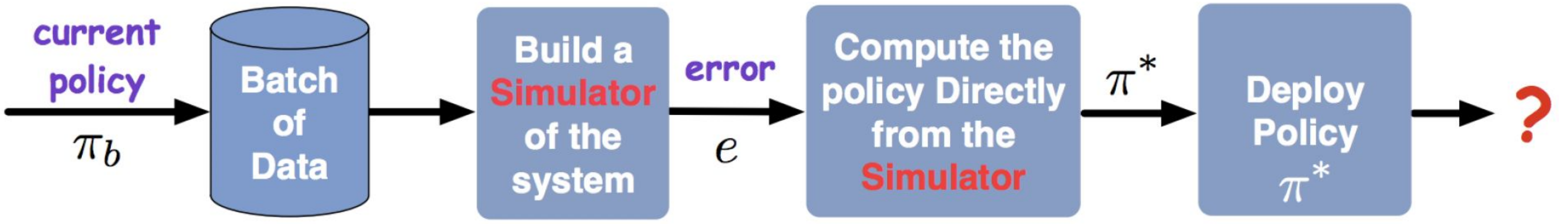
Criteria	Expert	GAIL	RAIL	Ours
		Hopper-v1		
Mean	- 6096	-5853	-6064	**-6105**
VaR_α	**-6129**	-6019	-6125	-6124
CVaR_α	-5590	-4958	-5493	**-5657**
ρ_α^λ	-6375	-6100	-6338	**-6387**

Criteria	Expert	GAIL	RAIL	Ours
		Walker-v1		
Mean	**-7651**	-7231	-7363	-7572
VaR_α	-7875	-7274	-7773	**-7909**
CVaR_α	-6440	-5353	-5505	**-5926**
ρ_α^λ	**-7973**	-7498	-7638	-7868

NOTE: RSGAIL >> GAIL; CVaR term reduces variance of cost gradient

3. Minimizing Robust Regret in RL (Sim-to-Real)

Safety w.r.t. baseline guarantee, <u>model-based</u> approach



Proposed formulation: To maximize the robust return regret w.r.t. baseline policy π_b, over the set of model uncertainties ξ:

$$\max_{\pi} \min_{\xi} \Big(\rho(\pi, \xi) - \rho(\pi_b, \xi) \Big)$$

Solution is guaranteed to be <u>safe</u>! Hopefully not as conservative as π_b

- Main challenge: Optimal policy is **stochastic**; NP hard
  - Heuristic: Assume perfect knowledge of **baseline** actions:

$$e(s, \pi_{\mathrm{B}}(s)) = 0;$$

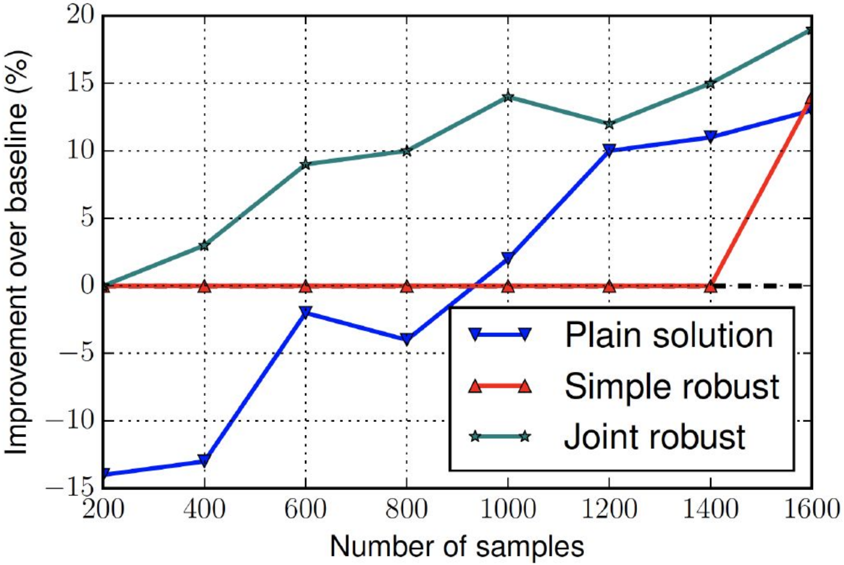
    Reduce the problem into a special **robust MDP**

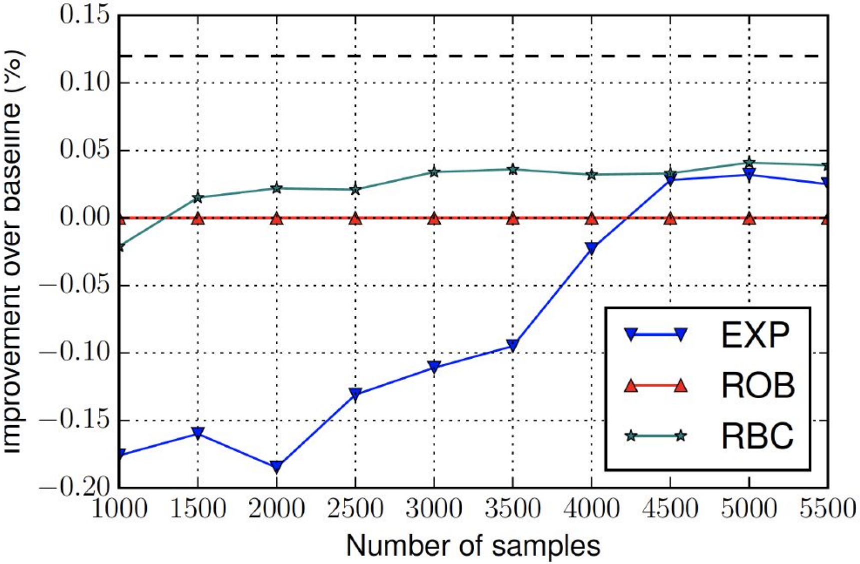
- Other baselines:
  - Classical approach (Solve the expected MDP model)
  - Robust MDP:
    1. Compute a robust policy:

$$\tilde{\pi} \leftarrow \arg\max_{\pi} \min_{\xi} \rho(\pi, \xi)$$

    2. Accept $\tilde{\pi}$ if outperforms π_{B} with prob $1-\delta$:

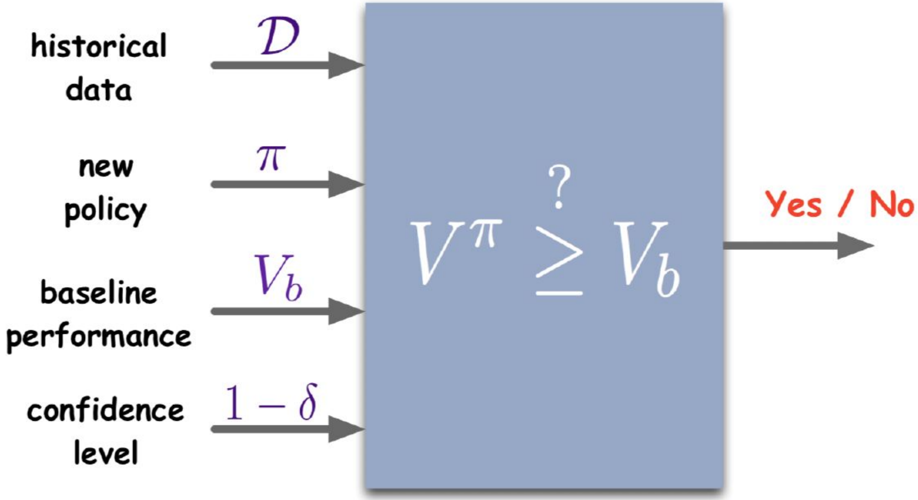
$$\min_{\xi} \rho(\tilde{\pi}, \xi) \geq \max_{\xi} \rho(\pi_{\mathrm{B}}, \xi)$$

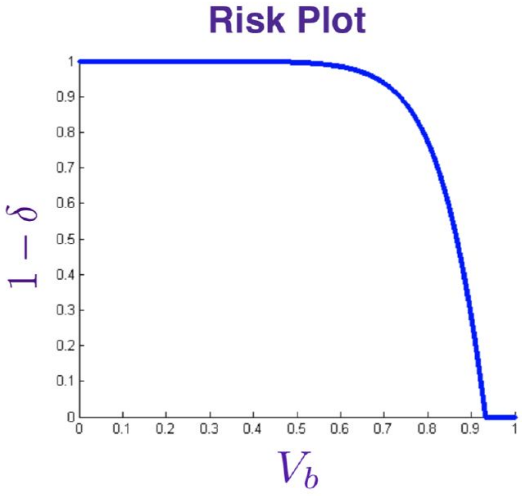

20
15
10
5
0
−5
−10
−15
200
400
600
800
1000
1200
1400
1600
Improvement over baseline (%)
Number of samples
Plain solution
Simple robust
Joint robust


0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500
Improvement over baseline (%)
Number of samples
EXP
ROB
RBC

4. More Robust Doubly Robust Off-policy Evaluation


current policy
π_b
Batch of Data
Compute the Policy Directly from the Batch of Data
π
Deploy Policy π
?


historical data
$\mathcal{D}$
new policy
π
baseline performance
V_b
confidence level
$1-\delta$
$V^\pi \overset{?}{\geq} V_b$
Yes / No


Risk Plot
$1-\delta$
V_b

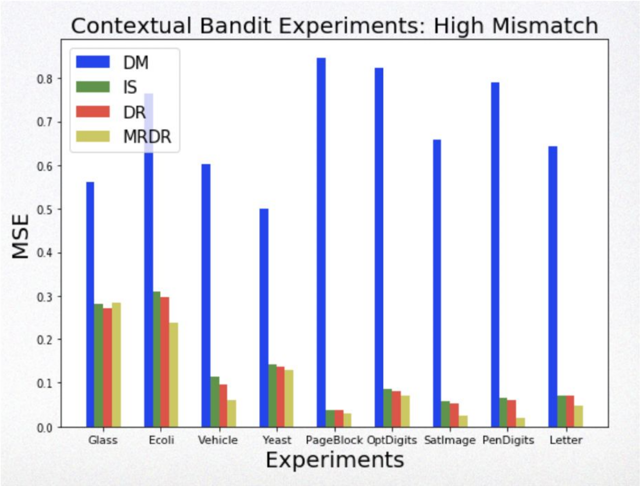
	Contextual Bandit	RL
Data	$\{(x_i, a_i, r_i)\}_{i=1,\ldots,N}$	$\{(x_i^t, a_i^t, r_i^t)\}_{i=1,\ldots,N}^{t=0,\ldots,T-1}$
Value	$\rho^{\pi_e} = \mathbb{E}_{p_0, a\sim\pi_e}[r(x, a)]$	$\rho^{\pi_e} = \mathbb{E}_{p_0, a\sim\pi_e}[\sum_{t=0}^{T-1} r(x_t, a_t)]$

- Evaluate an OPE estimator $\hat{\rho}^{\pi_e}(\xi)$ based on MSE:

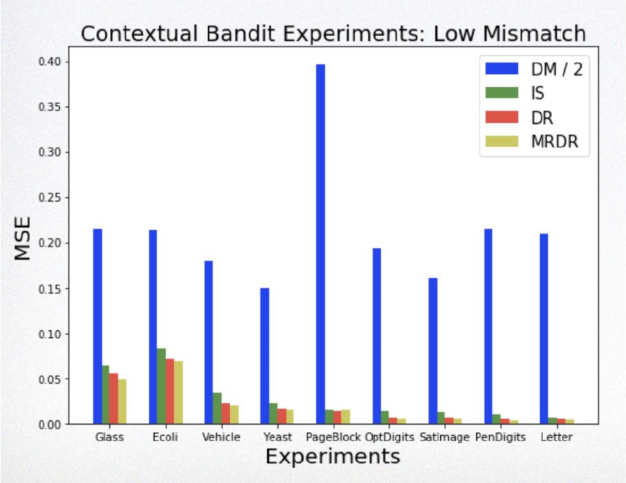
$$\mathrm{MSE}(\rho^{\pi_e}, \hat{\rho}^{\pi_e}) \triangleq \mathbb{E}_{\pi_b}[(\rho^{\pi_e} - \hat{\rho}^{\pi_e}(\xi))^2]$$

- Existing OPE estimators:
  1. Direct Method (DM) (find a value function model β, accurate but biased)
  2. Importance Sampling (IS) (model-free estimator, unbiased)
  3. Doubly Robust (DR) (hybrid estimator, leverage the best of both worlds)

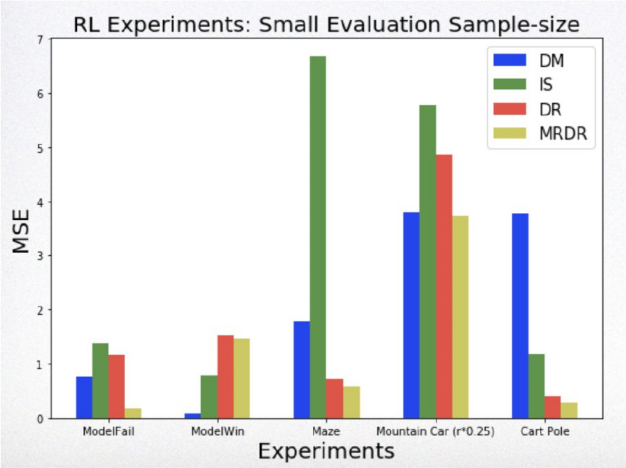
- Main Question: How to design β to minimize variance (MSE) of DR?



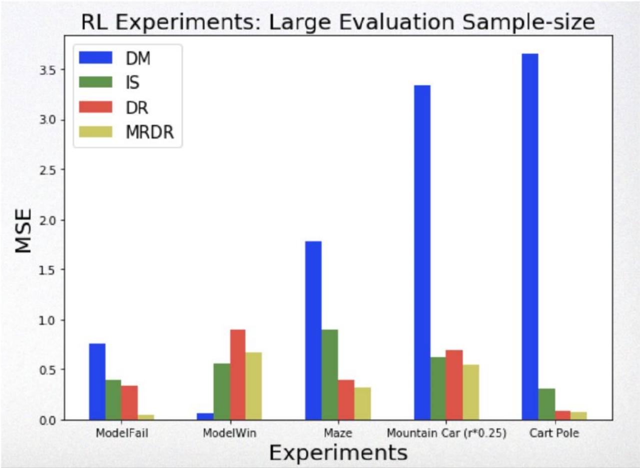
Contextual Bandit Experiments: High Mismatch
DM
IS
DR
MRDR
MSE
Experiments
Glass
Ecoli
Vehicle
Yeast
PageBlock
OptDigits
SatImage
PenDigits
Letter



Contextual Bandit Experiments: Low Mismatch
DM / 2
IS
DR
MRDR
MSE
Experiments
Glass
Ecoli
Vehicle
Yeast
PageBlock
OptDigits
SatImage
PenDigits
Letter



RL Experiments: Small Evaluation Sample-size
DM
IS
DR
MRDR
MSE
Experiments
ModelFail
ModelWin
Maze
Mountain Car (r*0.25)
Cart Pole



RL Experiments: Large Evaluation Sample-size
DM
IS
DR
MRDR
MSE
Experiments
ModelFail
ModelWin
Maze
Mountain Car (r*0.25)
Cart Pole