Motivation: sim2real transfer for RL
- Empirical success of deep RL (Atari games, MuJoCo, Go, etc.)
- Popular algorithms are sample-intensive for real-world applications
- Sim2real approach: (1) train in a simulator, (2) transfer to real world
- Hope: reduce sample complexity with a high-fidelity simulator

Empirical success of deep RL (Atari games, MuJoCo, Go, etc.)
What to transfer: policy, features, skills, etc. (we focus on policy)
How to quantify fidelity
Local errors
Hope:
Prior theories (e.g., [2]) focus on global error (worst over all states)
Proof sketch:
Issue with Model 1: too pessimistic
Illustration:
Approximate model:
Lower bound: $\Omega(\ldots)$
Deeper thoughts: many scenarios in sim2real transfer
- What to transfer: policy, features, skills, etc. (we focus on policy)
- How to quantify fidelity
  - Prior theories (e.g., [2]) focus on global error (worst over all states)
  - Local errors (#states with large errors)
- Is interactive protocol really better than non-interactive? Answer: Yes!

Setup
- Real environment: episodic MDP $M = (S, A, P, R, H, s_0)$.
- Simulator: $\hat{M} = (S, A, \hat{P}, R, H, s_0)$.
- Define $X_{\text{sim}}$ as the set of “wrong” $(s, a)$ pairs where
  $||P(s,a) - \hat{P}(s,a)||_V > \xi.$
- Goal: learn a policy $\pi$ such that $V^*(s) - V^0(s) \leq \epsilon$, using only
  $\text{poly}(|X_{\text{sim}}|, H, 1/\epsilon, 1/\delta)$ real trajectories.
No dependence on $|S|$ or $|A|$; instead, adapt to the simulator’s quality.
- This is impossible without further assumptions...

Lower bound and hard instances
- Lower bound: $\Omega(|S|A/\epsilon^2)$, even when $|X_{\text{sim}}| = $ constant!
- Proof sketch:
  - Bandit hard instance: $M = $ all arms Ber(½), except one w/ Ber(½+ $\epsilon$)
  - Approximate model: $\hat{M} = $ all arms Ber(½) --- $|X_{\text{sim}}| = 1$ but useless
- Illustration:

Non-interactive protocol is inefficient
- Theorem 3: “Collect data, calibrate, done” style algorithms cannot have
  poly$(|X_{\text{sim}}|, H, 1/\epsilon, 1/\delta)$ sample complexity, even with Conditions 1 & 2.
  Proof sketch: assume such an algorithm exists. Then,
  - The same dataset can calibrate multiple models.
  - Consider the hard instance in bandit. Design $|A|$ models: $∀ a, a' \in A,$
    $\hat{M}_{a,a'} = $ all arms Ber(½), except $a \neq a'$ w/ Ber(½+ $\epsilon$).
  - When $a = a'$, both Conditions 1 & 2 are met and $|X_{\text{sim}}| = 1$.
  - Hypothetical algorithm prefers $a'$ to $a'$ with $\epsilon$ probability, using a dataset of constant size.
  - Majority vote from $\Omega(|A|)$ datasets: boost success prob. to $1 - O(1/|A|)$.
  - Solve bandit hard instance w/ polylog$(|A|)$, against $\Omega(|A|)$ lower bound.

Sufficient conditions and algorithms
Definition 1: A partially corrected model $\hat{M}_c$ is one whose dynamics are
the same as $M$ on $X$, and the same as $\hat{M}$ otherwise.
Condition 1: $V^*(s)$ is always higher in $\hat{M}_c$ than in $M$ for all $X \subseteq X_{\text{sim}}$.
(see the agnostic version of the conditions in the paper.)
Theorem 1: Under Condition 1, there exists an algorithm that achieves
$O(|X_{\text{sim}}|^2 H^2 \log(1/\delta)/\epsilon^2)$ sample complexity for $\xi = O(\epsilon/H^2)$.
Algorithm 1: illustration on previous example, Model 3.
- Collect data using optimal policy in simulator.
- Blue cells: plug in estimated dynamics along states w/ enough samples.

What if we cannot change the model?
Basic idea:
- Identify the wrong states as necessary.
-Terminate a simulated episode when running into wrong $(s, a)$.
  = penalize a wrong $(s, a)$ by fixing $Q(s,a) = 0$ ($V_\omega$) in planning.
Definition 2: A partially penalized model $M_{\omega}$ is one that terminates on $X$, and have the same dynamics as $\hat{M}$ otherwise.
Condition 2: $V^*(s)$ is always higher in $\hat{M}_c$ than in $M$ for all $X \subseteq X_{\text{sim}}$.
Theorem 2: Under Condition 2, there exists an algorithm that achieves
$O(|X_{\text{sim}}|^2 H^2 \log(1/\delta)/\epsilon^2)$ sample complexity for $\xi = O(\epsilon/H^2)$.
Algorithm 2: $M_0 \leftarrow \hat{M}_c$, $X_0 \leftarrow \emptyset$.
For $t = 0, 1, 2, \ldots$
- Let $\pi_0$ be the optimal policy of $M_c$. Monte-Carlo evaluate $\pi_0$.
- Return if $V^0(s)$ in $M$ is close to $V^0(s)$ in $M_c$.
- Sample real trajectories using $\pi_0$.
- Once #samples from some $(s, a)$ reaches threshold, compute
  $[\mathbb{E}_{s,a \sim \pi_0} [V'(s)]] - [\mathbb{E}_{s,a \sim M_c} [V'(s)]]$
- If large, $X_0 \leftarrow X_0 \cup [(s, a)]$, $M_{t+1} \leftarrow M_{t}$.  