UMassAmherst | Manning College of Information & Computer Sciences

RL^3: Boosting Meta Reinforcement Learning via RL inside RL^2

Abhinav Bhatia, Samer B. Nashed, Shlomo Zilberstein

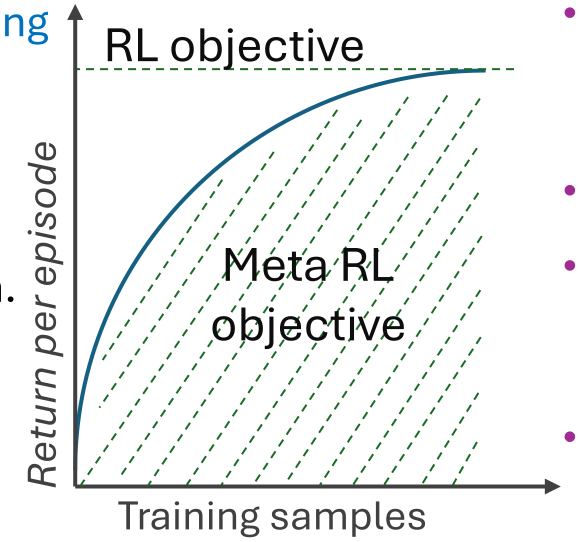
Aug 6, 2025

Overview

An RL algorithm: a mapping from experience history to actions.

Standard RL

- Given: an MDP
- Project: learn a *state-to-action* mapping to maximize cumulative reward *per episode*.
- *Output: "Policy"*
- Involves value functions to distill data.
- Standard RL Pros & cons:
  - Data inefficient (not the objective)
  - General-Purpose (achieves objective on any MDP)
  - Asymptotically optimal



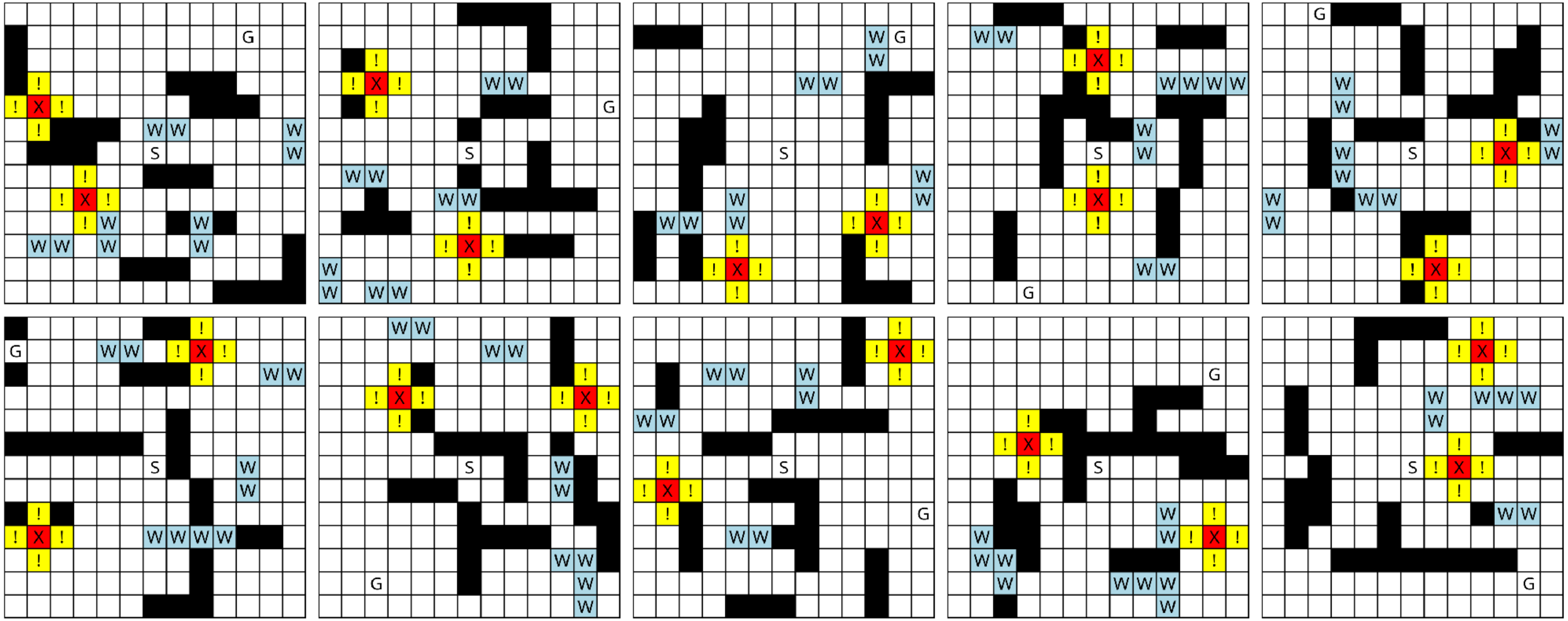
Meta RL

- Given: a distribution of MDPs / "tasks"
- Project: learn a *data-to-action* mapping maximize cumul. reward *over entire interaction period (fixed)*. *Min regret.*
- *"Meta-RL policy"* or *"Fast RL"*
- Involves a sequence model (RNN/Transformer) to ingest data (and may be an incremental task-inference module). E.g., RL^2, VariBAD.
- Meta RL Pros & cons:
  - Data-efficient (objective = min regret)
  - OOD issue (not trained for that)
  - Long-context problem (compute bottleneck, context truncation, gradient problems, long range credit assignment difficulty, compounding errors)

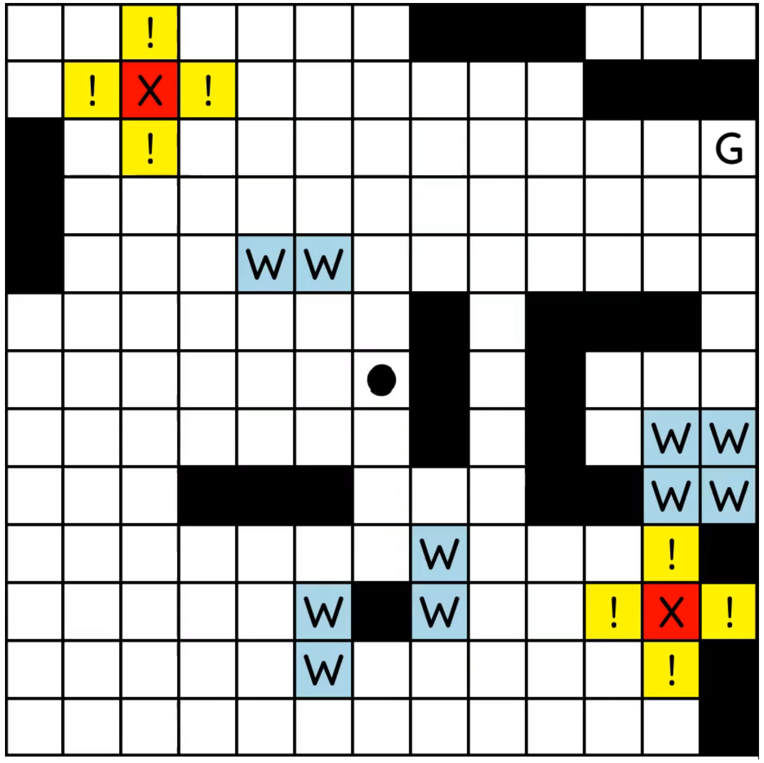
RL^3: Injects RL into Meta-RL: Augment experience inputs with Q^* estimates

Distribution of MDPs

- Same state, action space. Different reward, transition functions, with commonalities.



Meta-Episode Example



Action: ▯ Reward: 0.0

- Many episode on one MDP <-> 1 meta-episode

Meta Reinforcement Learning

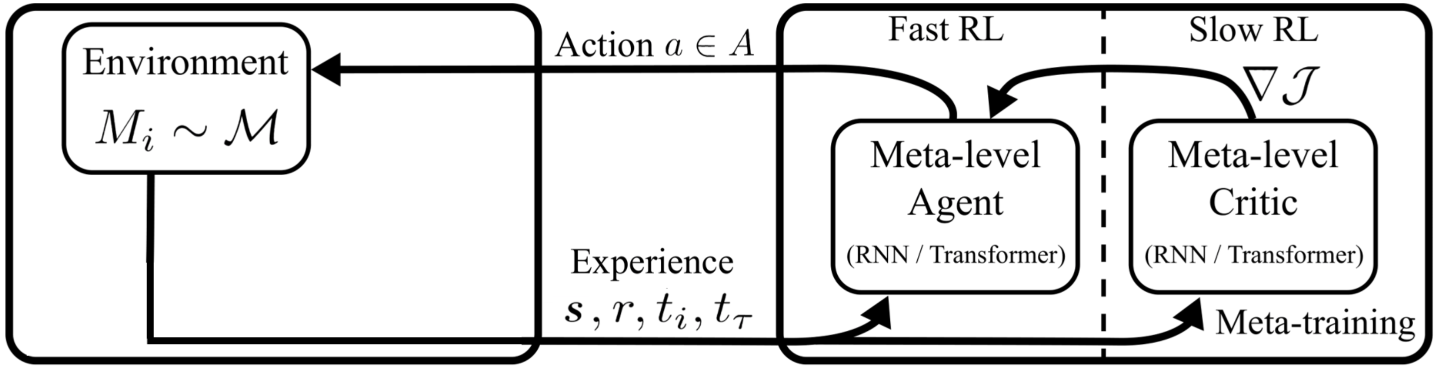
- Objective: Learn a data-to-action mapping to maximizes cumulative reward

$$\mathcal{J}(\pi) = \mathbb{E}_{M_i \sim \mathcal{M}} \left[\sum_{t=0}^{H} \gamma^t \, \mathbb{E}_{(s_t, a_t) \sim \rho_{i,t}^{\pi}} \left[R_i(s_t, a_t) \right] \right]$$

- As a meta-level Markov decision process:
    - Each meta-episode: sample a new MDP, or "task", play for H interactions.
    - Optimal meta policy maximizes cumulative reward. Least regret given H interactions ✅.
    - Dynamics different across meta-episodes, because every meta-episode is a new MDP.
    - It's POMDP at meta-level, where hidden variable is task identity, aka *BAMDP*.
    - Beliefs over tasks capture history sufficiently.
    - Conditioning meta-policy on (s_t, b_t) is sufficient.

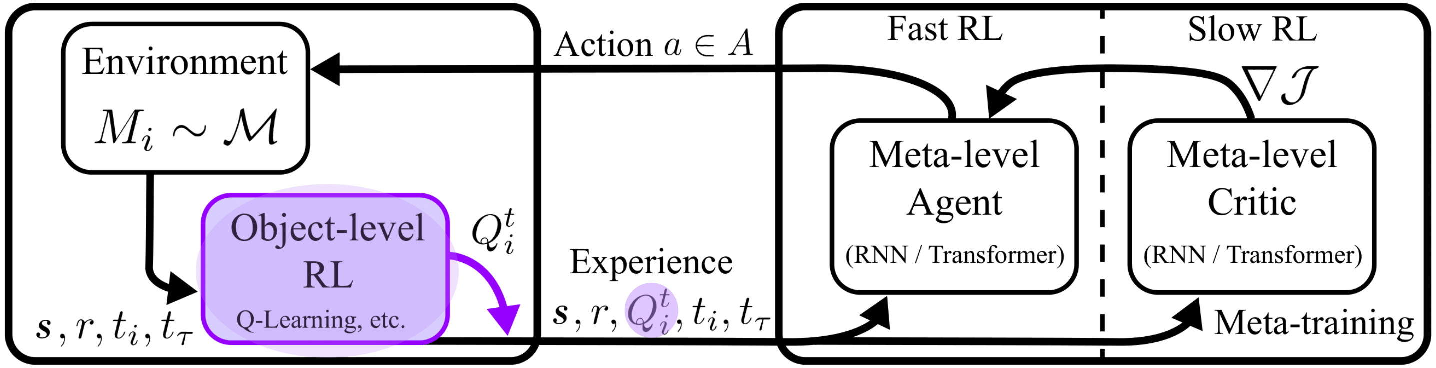
RL^2: Fast RL using Slow RL (Duan et al. 2016)

- RL^2 maps raw-data to actions directly (A simple black-box method).
  (Note: Some approaches map data-to-beliefs first e.g., VeriBAD (Zintgraf et al., 2019))
- Trained with standard "slow" deep RL.
- No general-purpose components.



RL³: Inject RL into RL²

- Insert a RL subroutine: estimate Q*-values e.g., use Q-learning.
- Provide to meta-RL. (Provide action-counts too).
- Meta-RL decides how to use.
- Over time, Q-values improve.

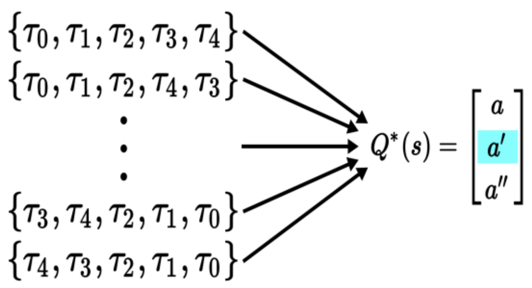


RL^3: Inject RL into RL^2: But why?

Claim: Q-injection 💉 improves OOD generalization and long-context reasoning.

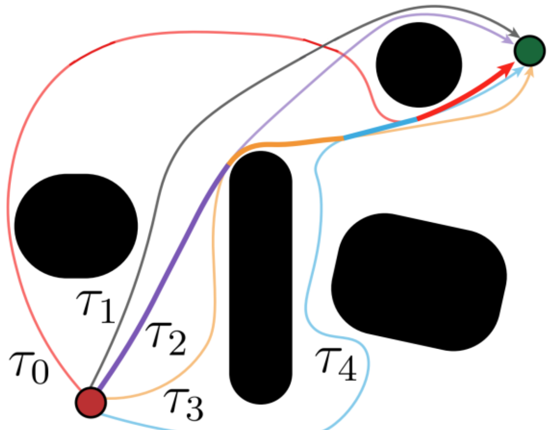
Inherent generality:

Key component in general-purpose RL



Summarization: Many-to-one mapping. Order is irrelevant.

Lossy, but "remembers" key details



Actionability:

Handles long-range credit assignment, "stitching trajectories", optimal policy given data.

Eventually, Can ignore history, just exploit

Bottom line: Over time, data becomes overwhelming, Q-estimates more useful.

RL³: Inject RL into RL²: But Why?

Theoretical Reasons

Excellent task discriminators / identifiers:

Rare for MDPs to have same Q-value function, or same evolution of Q-estimates.

Related to meta-value function:

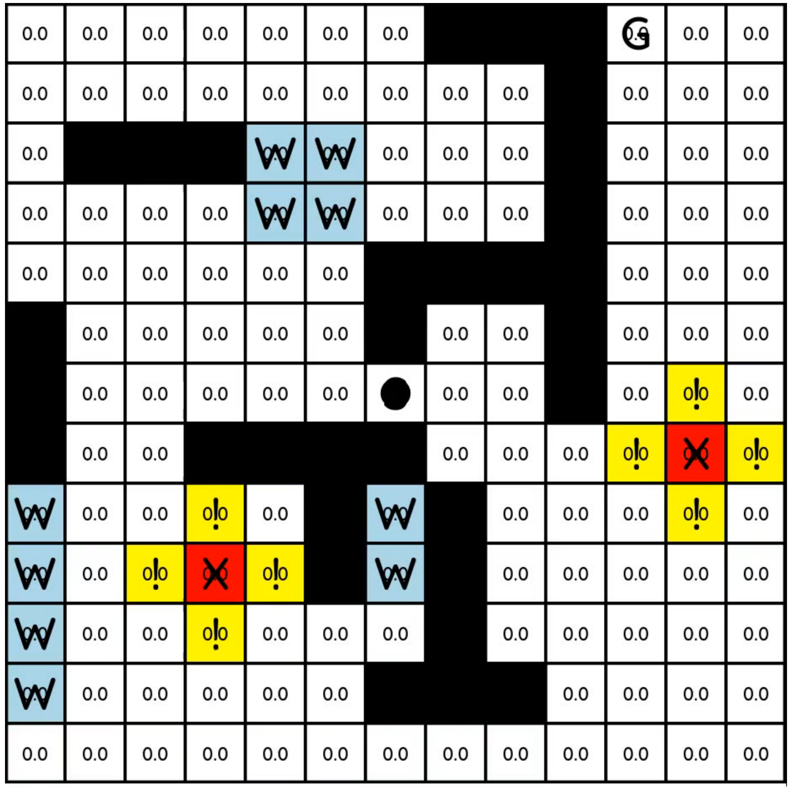
Equivalent in the limit!

for any $\epsilon > 0$, there exists $\kappa \in \mathbb{N}$ such that for $t \geq \kappa$,

$$\left| \max_{a \in A} \left[Q_i^t(s, a) \right] - \bar{V}^*(\bar{b}) \right| \leq \epsilon \qquad \forall s \in S.$$

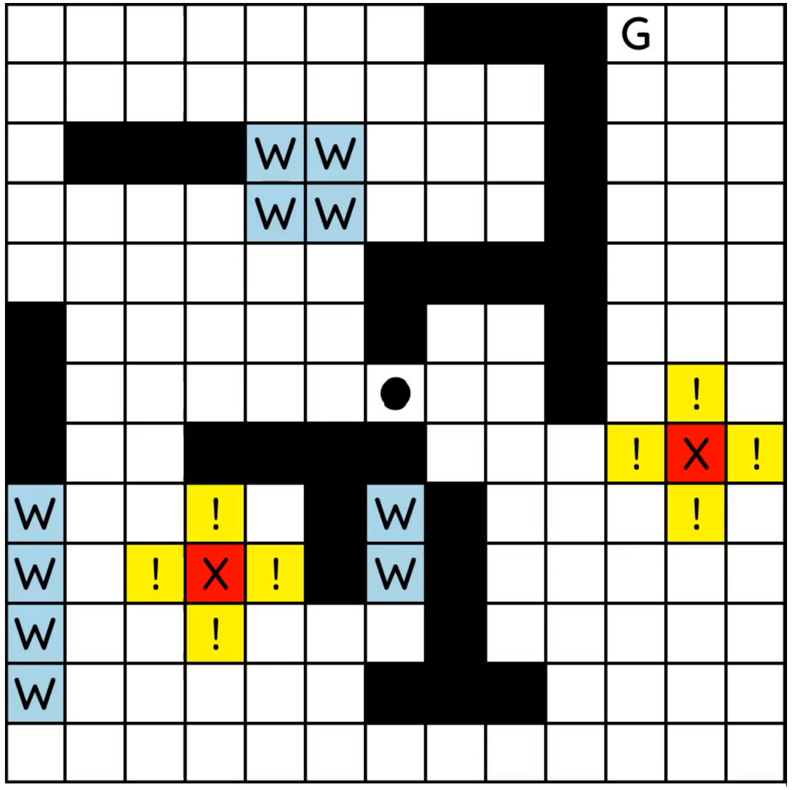
RL³ vs RL² - Gridworlds Results Demo

RL^3



Action: □ Reward: 0.0

RL^2



Action: □ Reward: 0.0

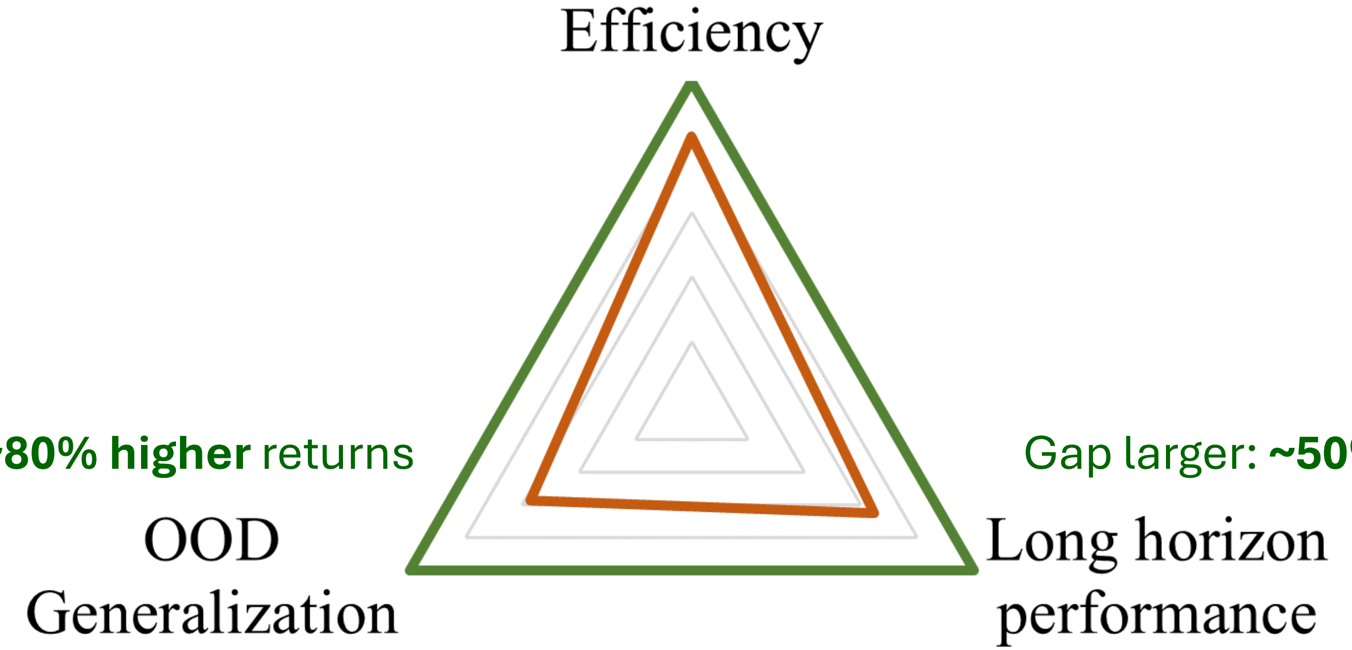
https://youtu.be/eLA_S1BQUYM

RL^3 vs RL^2 - Gridworlds Results

RL^3 achieves lower regret. **~20% higher** returns

Gap Largest: **~80% higher** returns

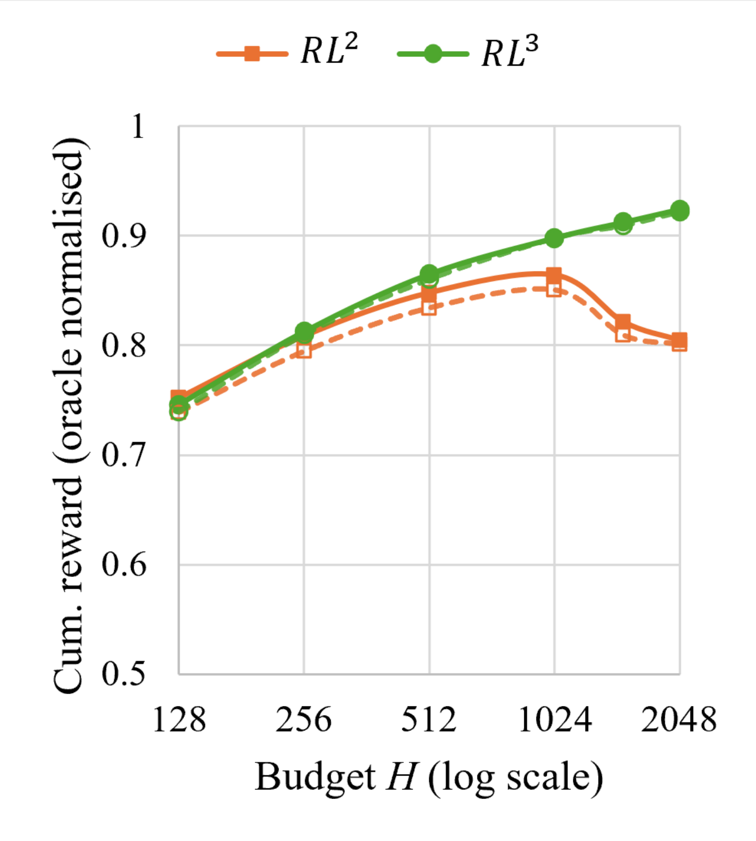
Gap larger: **~50% higher** returns



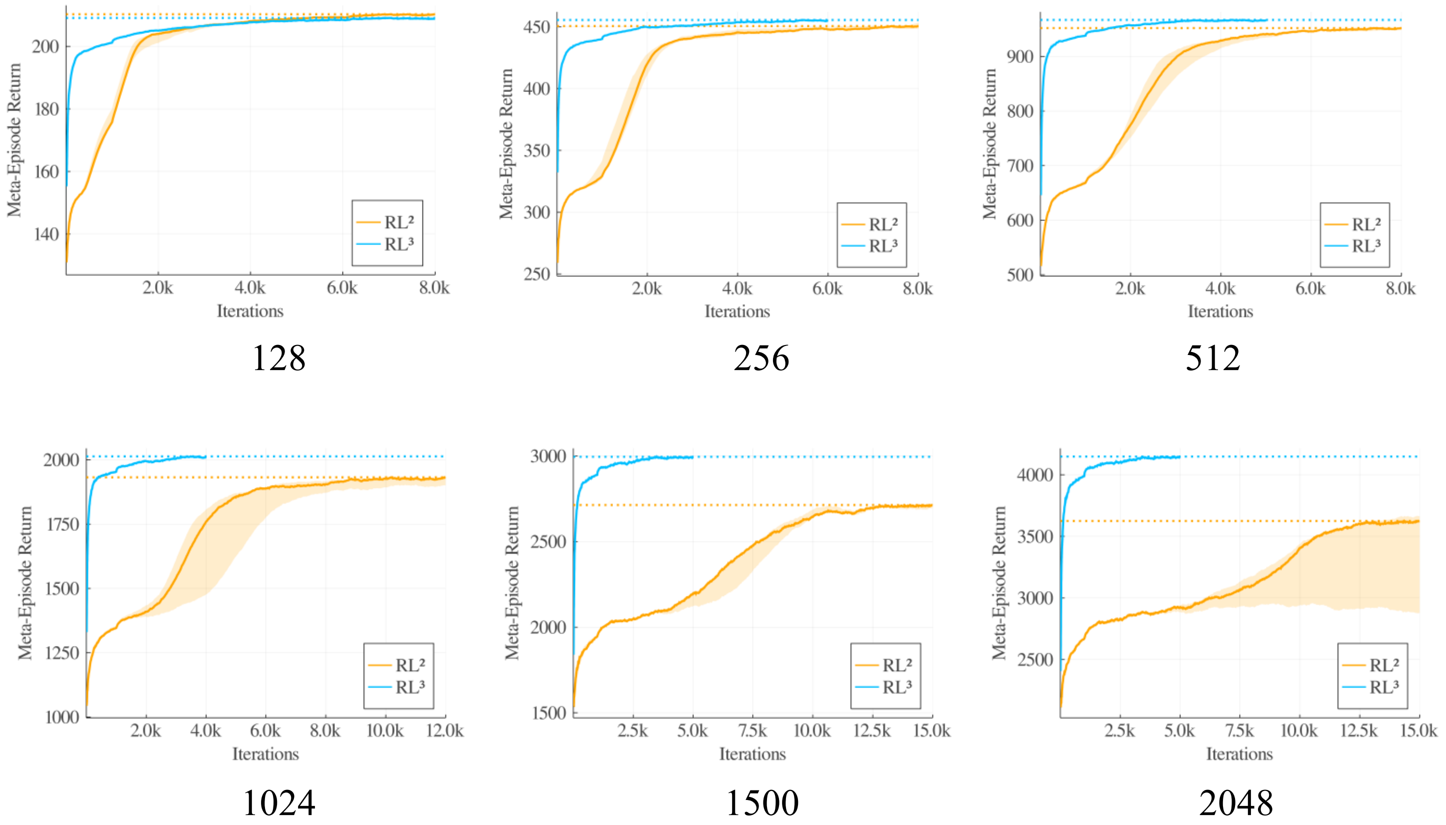
RL^3 with state-abstractions: *RL^3-coarse:* 2x fast, 90% of RL^3.

Bonus: RL^3 meta-training: **~30% more sample efficient**

RL³ vs RL² - Random MDPs Results



Meta-training graphs



Conclusion

- We introduced RL^3, aiming to combine best of RL and RL^2 – to achieve good efficiency (minimize regret), better long-term reasoning, better OOD generalization.
- Intuitions: Universality, summarization, actionability, long-term credit assignment and with helps task identification. With time, data gets overwhelming, Q-estimates useful, eventually sufficient.
- Key takeaways:
  - RL^3 retains short-term efficiency of RL^2 on all domains
  - RL^3 benefits increase with horizon.
  - RL^3 benefits increase with distribution shift.
  - Bonus result: meta-training efficiency.
- Plug & play