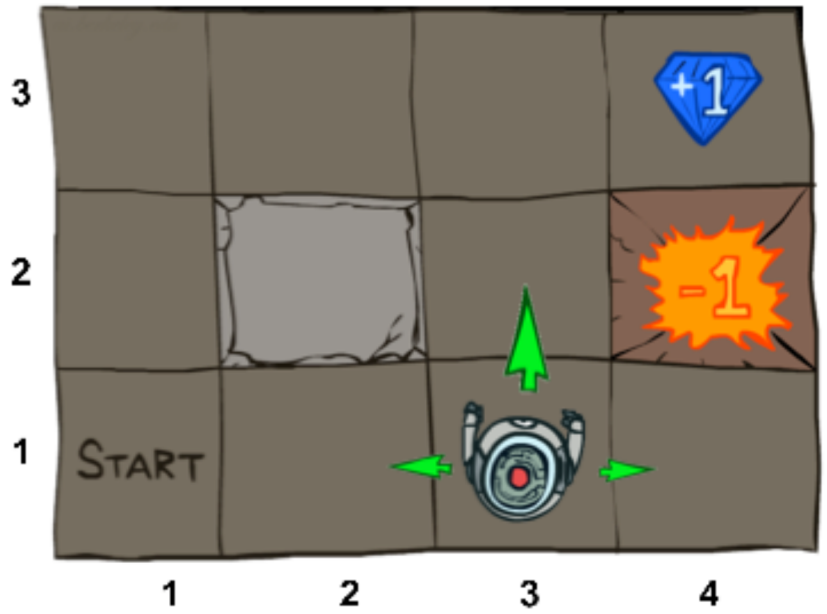
Deep Hep Reading Group

1611.05763 Learning To Reinforcement Learn

1611.02779 RL^2

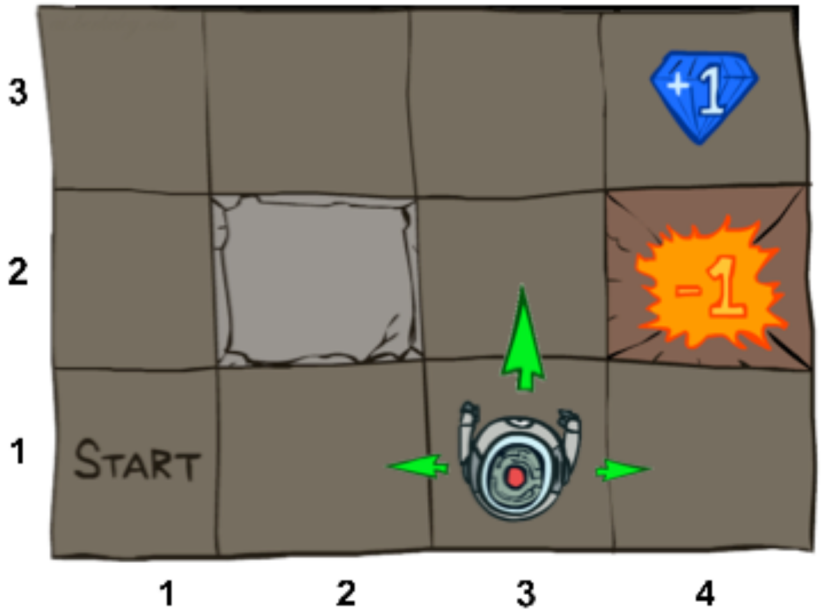
RL^1 Schematic

- Approach for solving Markov Decision Process
- Agent interacts with environment
  - Takes actions to move from one state to another
  - Is rewarded or penalized during the process.
- Example, grid world



RL^1 Notation

$\mathcal{S}$	– States
$\mathcal{A}$	– Actions
$\mathcal{P}(s'\|s,a)$	– Transition prob.
$r(s,a,s')$	– Reward
η	– Discounted return
$\pi_\theta(a\|s)$	– Policy
ρ_0	– Initial state dist.
s_0	– Initial state
γ	– Discount
T	– Horizon

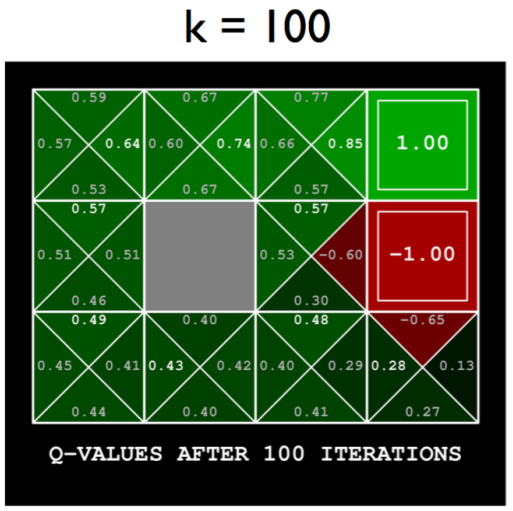


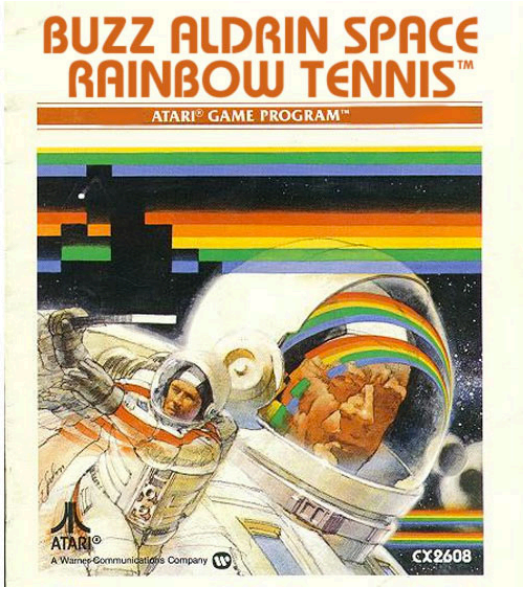
$$\eta(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t, s_{t+1}) \Big|_\pi\right]$$

If there exists an optimal π_*, can similarly define cumulative regret. $\mathbb{E}\left[r\Big|_{\pi_*} - r\Big|_\pi\right]$

RL^1 Strategies

- Value, Q-value iteration
  - Define value, V(s), of state or Q(s,a) of state and action based on optimal action from that state(action) until end. Easy to do when horizon, T is small.
  - Iterate in size of T
- Policy iteration
  - Similar, don't use optimal policy, iteratively improve policy.
- Good for gridworld bad for Atari

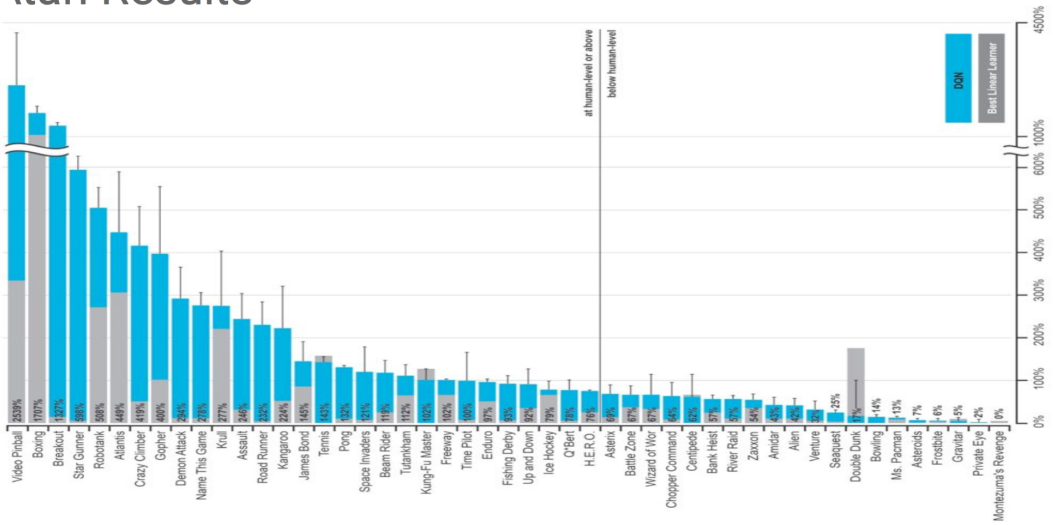




RL^1 DQN

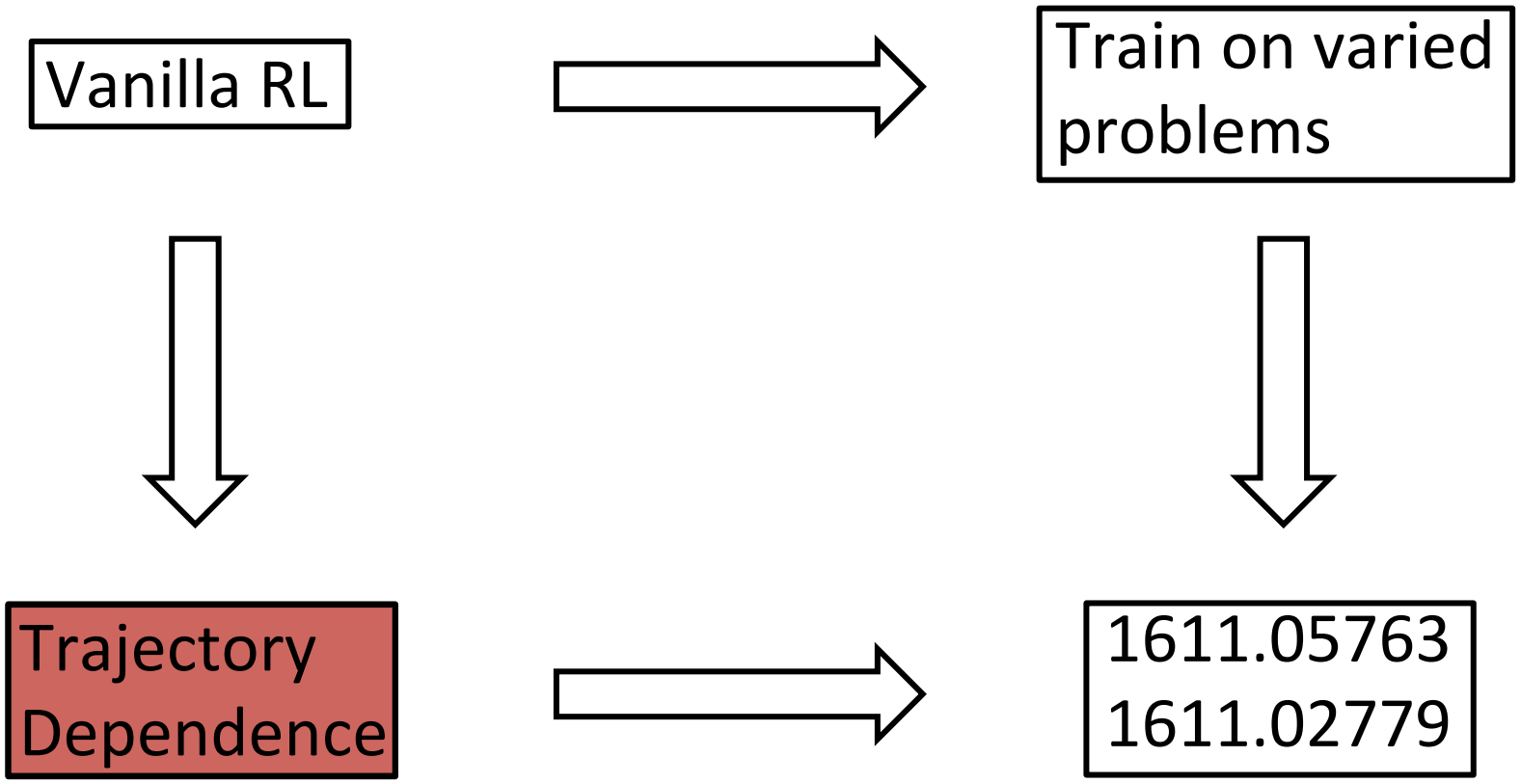
- For large sized games, can't use exact iteration.
- Instead model Q parametrically Q(θ). Why not make this a deep neural-net?

Atari Results



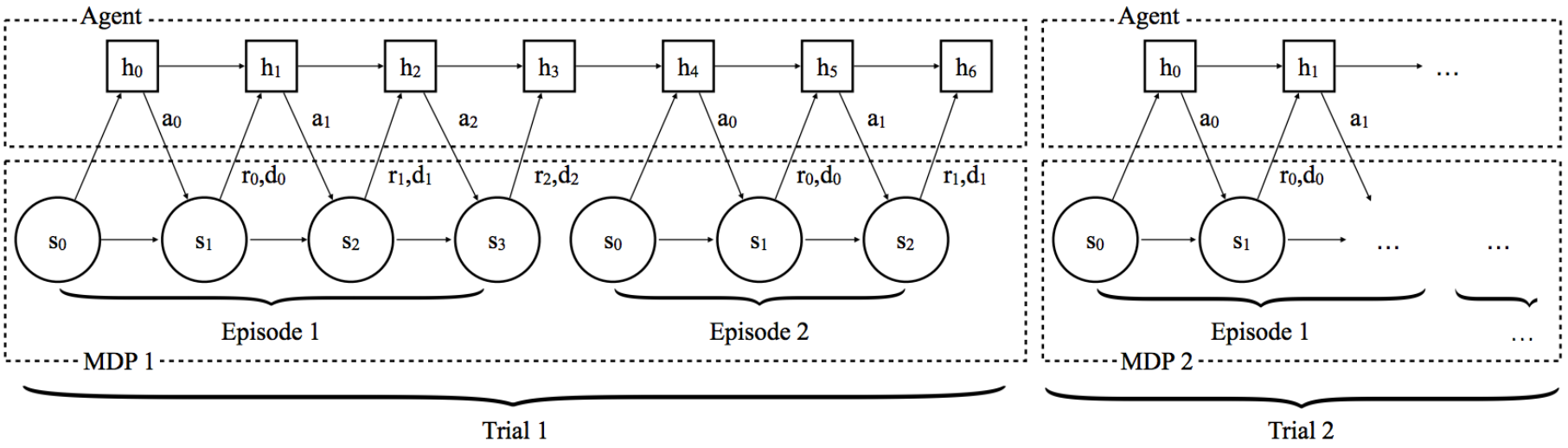
"Human-Level Control Through Deep Reinforcement Learning", Mnih, Kavukcuoglu, Silver et al. (2015)

Natural Generalizations

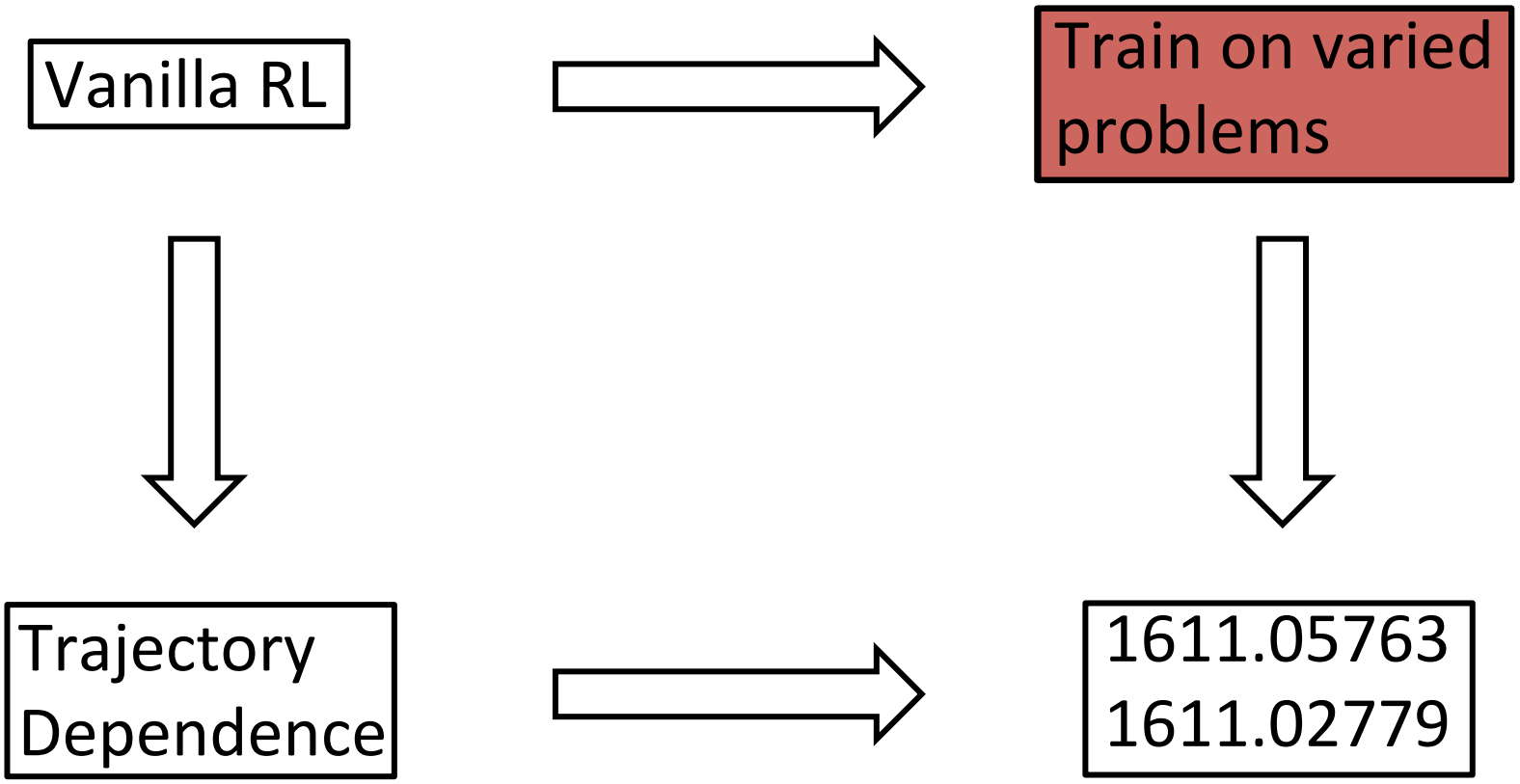


Trajectory Dependence

- Use LSTM to retain information



Natural Generalizations



1611.05763 Idea

- Train LSTM to learn structure dependent policies:

Some Examples

1611.05763 Training

- Fix MDP distribution D:
  - Sample from D, run for time T
  - Repeat many times
- Details were varied slightly depending on D
- Main Point: Agent gets good at all tasks from D, not just a particular instance.

1611.05763 Bandit Tasks

- Two armed bandit, each arm has probability pi to pay out 1, otherwise gives 0.
- Two armed bandit, correlated arms p1 = 1-p2
- Deferred gratification:
  - Among 11 arms 1 random arm gives high reward, 9 give low, arm 11 encodes which is high, but gives low payout
- Goosed up bandit with images

1611.05763 Results

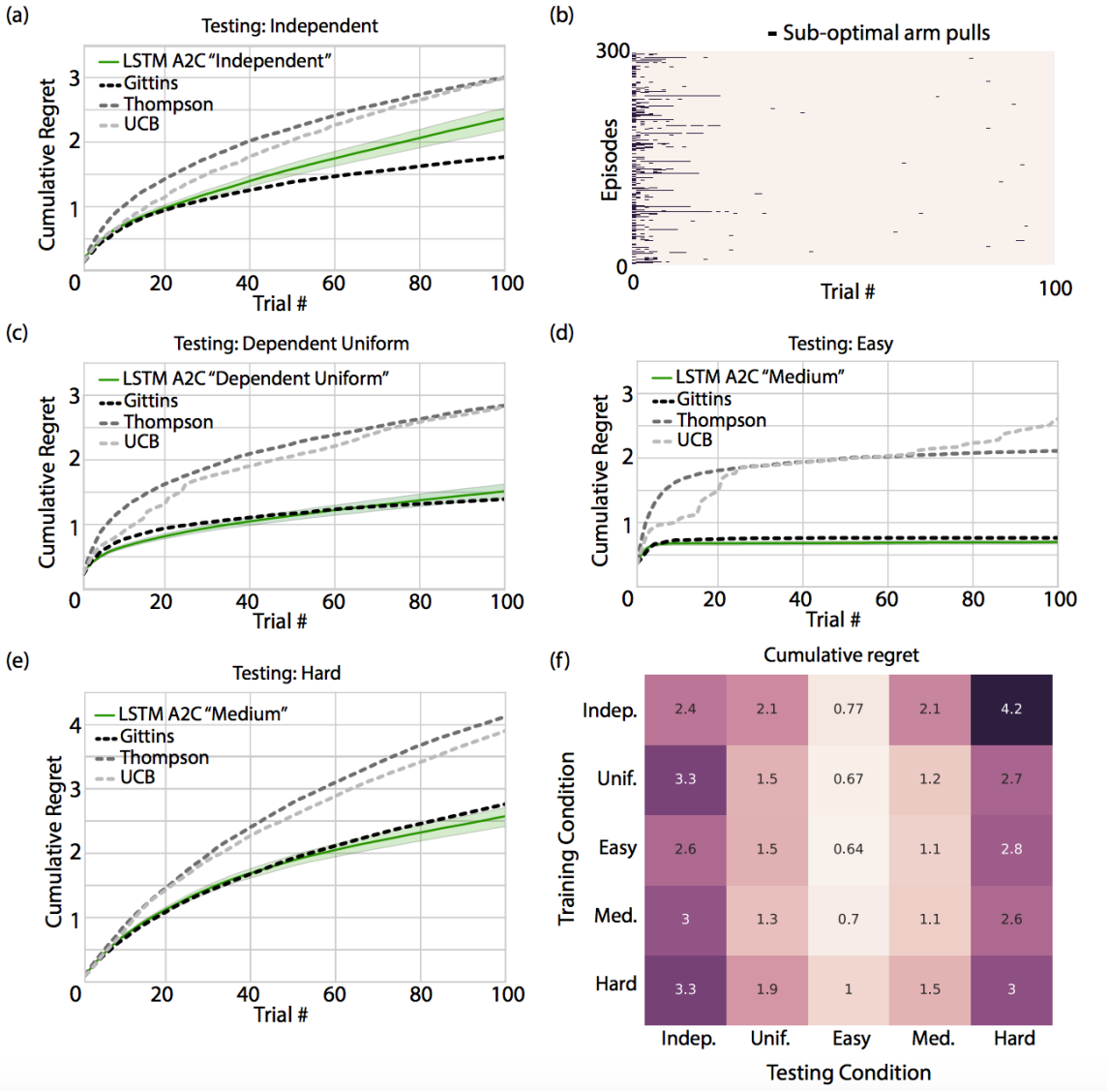
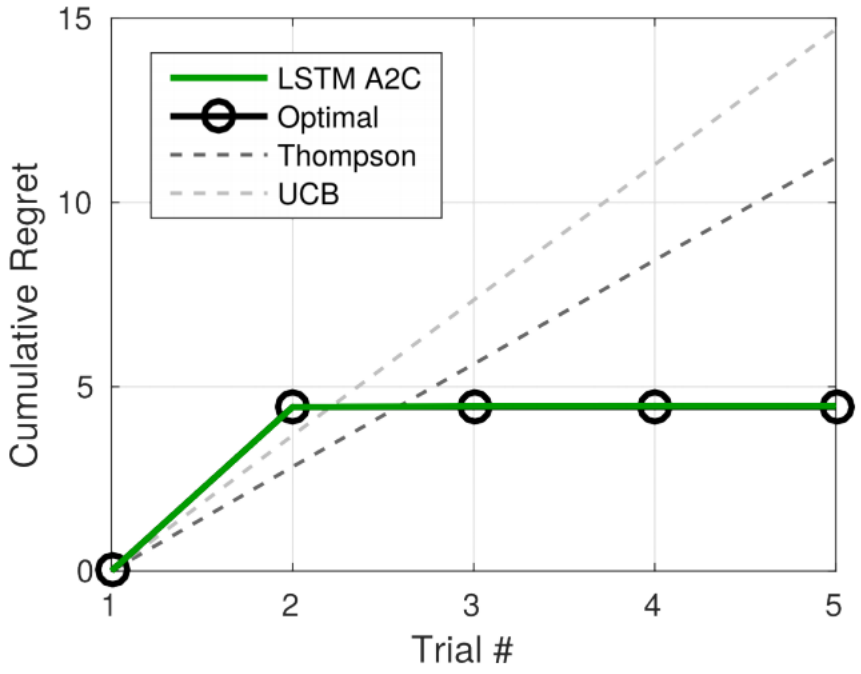


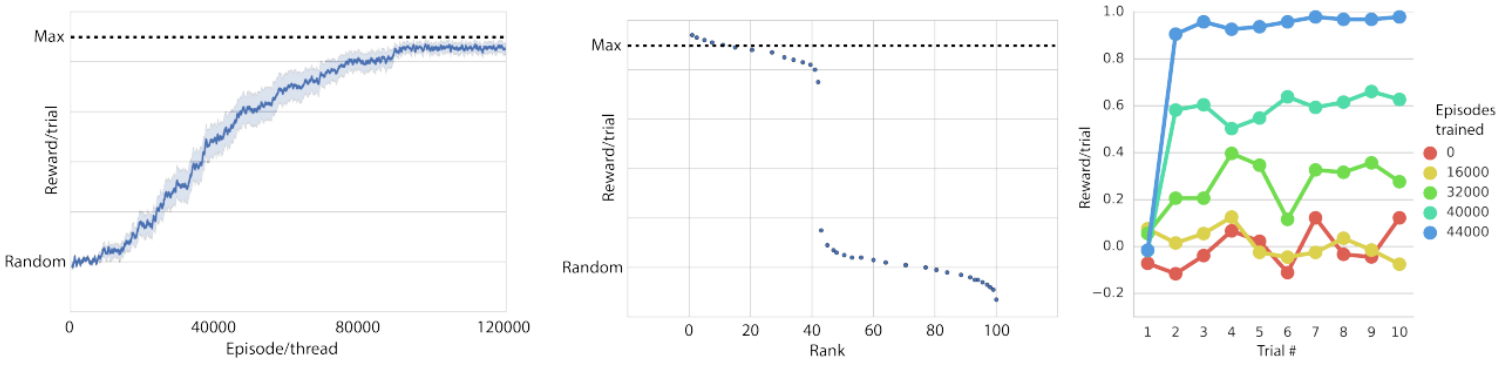
Figure 2: Performance on independent- and correlated-arm bandits. We report performance as the cumulative expected regret RT for 150 test episodes, averaged over the top 5 hyperparameters for each agent-task configuration, where the top 5 was determined based on performance on a separate set of 150 test episodes. **(a)** LSTM A2C trained and evaluated on bandits with independent arms (distribution Di; see text), and compared with theoretically optimal models. **(b)** A single agent playing the medium difficulty task with distribution Dm. Suboptimal arm pulls over trials are depicted for 300 episodes. **(c)** LSTM A2C trained and evaluated on bandits with dependent uniform arms (distribution Du), **(d)** trained on medium bandit tasks (Dm) and tested on easy (De), and **(e)** trained on medium (Dm) and tested on hard task (Dh). (f) Cumulative regret for all possible combinations of training and testing environments (Di, Du, De, Dm, Dh).

1611.05763 Deferred Gratification



Goosed Bandit

(a) Fixation (b) Image display (c) Right saccade and selection

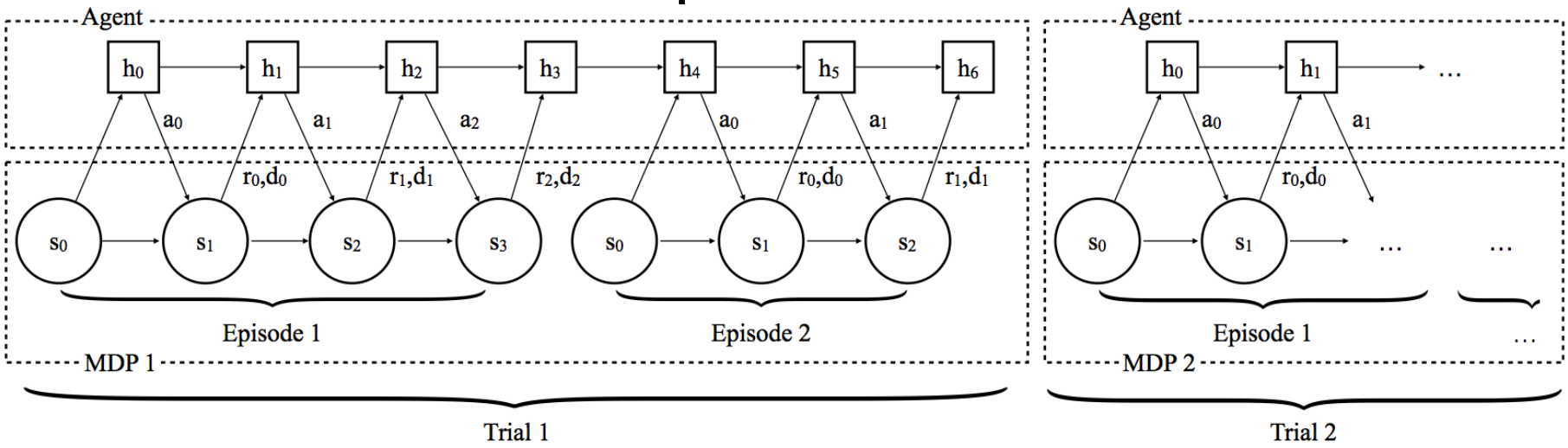


(d) Training performance (e) Robustness over random seeds (f) One-shot learning

Figure 6: Learning abstract task structure in visually rich 3D environment. **a-c)** Example of a single trial, beginning with a central fixation, followed by two images with random left-right placement. **d)** Average performance (measured in average reward per trial) of top 40 out of 100 seeds during training. Maximum expected performance is indicated with black dashed line. **e)** Performance at episode 100,000 for 100 random seeds, in decreasing order of performance. **f)** Probability of selecting the rewarded image, as a function of trial number for a single A3C stacked LSTM agent for a range of training durations (episodes per thread, 32 threads).

1611.02779 Training Structure

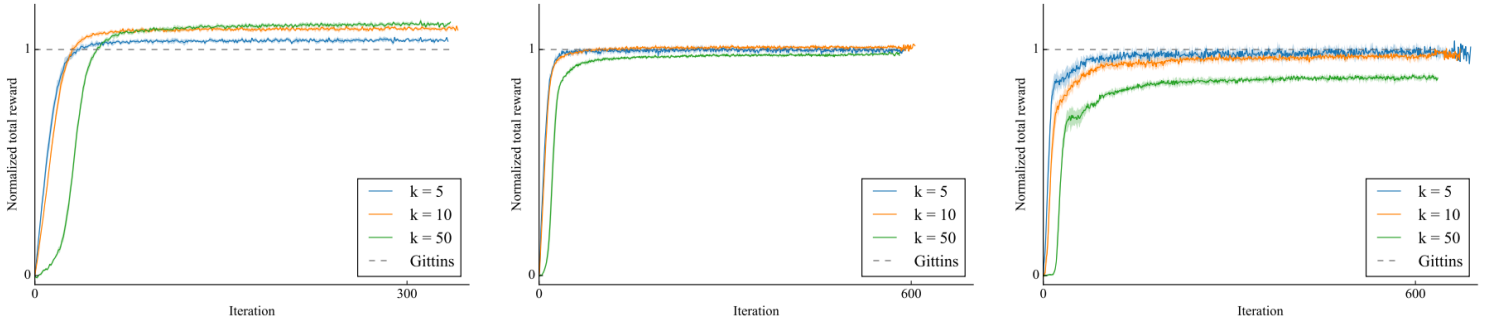
- Use GRUs instead of LSTMs, also sample broader classes of problems.



- "The objective is to maximize the … reward... over a single trial" – odd wording, over each, or multiple.
- Slightly different use of episode between two papers, trial here = episode there

1611. 02779 Bandit Results

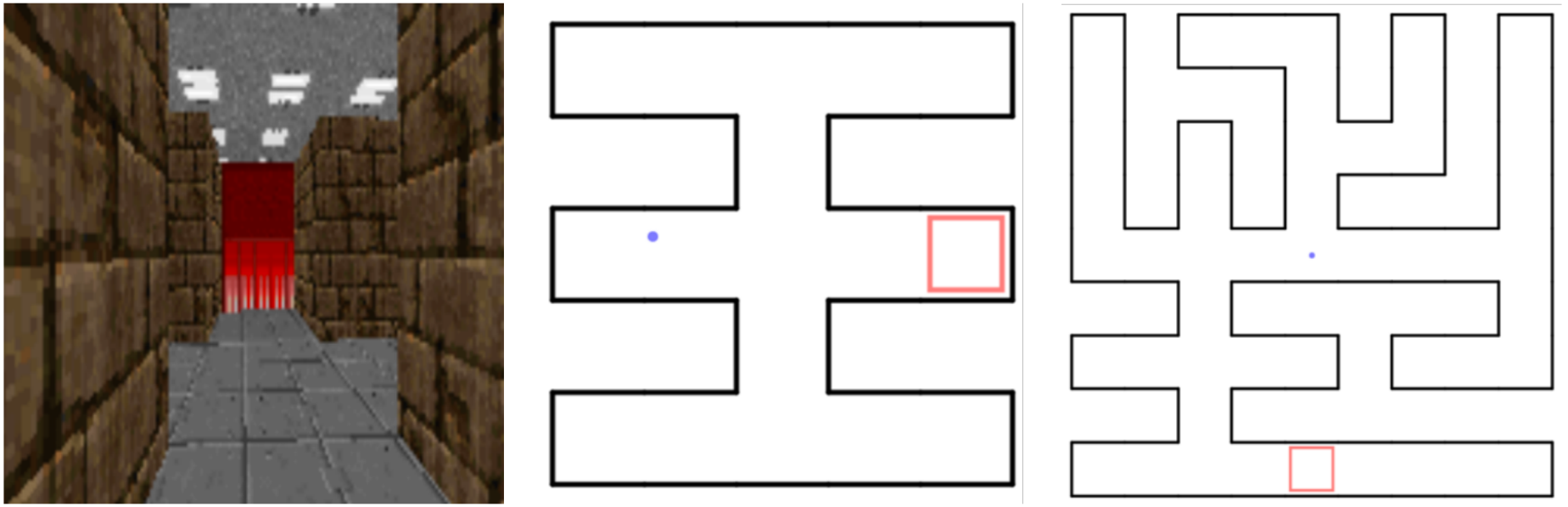
Setup	Random	Gittins	TS	OTS	UCB1	ϵ-Greedy	Greedy	RL^2
$n = 10, k = 5$	5.0	**6.6**	5.7	6.5	**6.7**	**6.6**	**6.6**	**6.7**
$n = 10, k = 10$	5.0	**6.6**	5.5	6.2	**6.7**	**6.6**	**6.6**	**6.7**
$n = 10, k = 50$	5.1	6.5	5.2	5.5	**6.6**	6.5	6.5	**6.8**
$n = 100, k = 5$	49.9	**78.3**	74.7	**77.9**	**78.0**	75.4	74.8	**78.7**
$n = 100, k = 10$	49.9	**82.8**	76.7	81.4	82.4	77.4	77.1	**83.5**
$n = 100, k = 50$	49.8	**85.2**	64.5	67.7	84.3	78.3	78.0	**84.9**
$n = 500, k = 5$	249.8	**405.8**	**402.0**	**406.7**	**405.8**	388.2	380.6	**401.6**
$n = 500, k = 10$	249.0	**437.8**	429.5	**438.9**	**437.1**	408.0	395.0	432.5
$n = 500, k = 50$	249.6	**463.7**	427.2	437.6	457.6	413.6	402.8	438.9



(a) $n = 10$ (b) $n = 100$ (c) $n = 500$

Figure 2: RL^2 learning curves for multi-armed bandits. Performance is normalized such that Gittins index scores 1, and random policy scores 0.

1611. 02779 Maze Task



(a) Sample observation (b) Layout of the 5×5 maze in (a) (c) Layout of a 9×9 maze

Figure 4: Visual navigation. The target block is shown in red, and occupies an entire grid in the maze layout.

r = +1 for reaching target, -0.001 for wall hit, and -0.04 per time step

1611. 02779 Maze Results

(a) Average length of successful trajectories

Episode	Small	Large
1	52.4 ± 1.3	180.1 ± 6.0
2	39.1 ± 0.9	151.8 ± 5.9
3	42.6 ± 1.0	169.3 ± 6.3
4	43.5 ± 1.1	162.3 ± 6.4
5	43.9 ± 1.1	169.3 ± 6.5

(b) %Success

Episode	Small	Large
1	99.3%	97.1%
2	99.6%	96.7%
3	99.7%	95.8%
4	99.4%	95.6%
5	99.6%	96.1%

(c) %Improved

Small	Large
91.7%	71.4%

Videos

Comments

- The previous learning to learn is a special case of this.
  - Think of gradient decent as agent moving in a potential: state is position and cost, action is move in any direction any amount, reward is cost decrease.
- DQN alone already accomplishes some of this.
  - Ex think of each frame of atari as new draw
  - Seaquest agent displays delayed gratification for instance