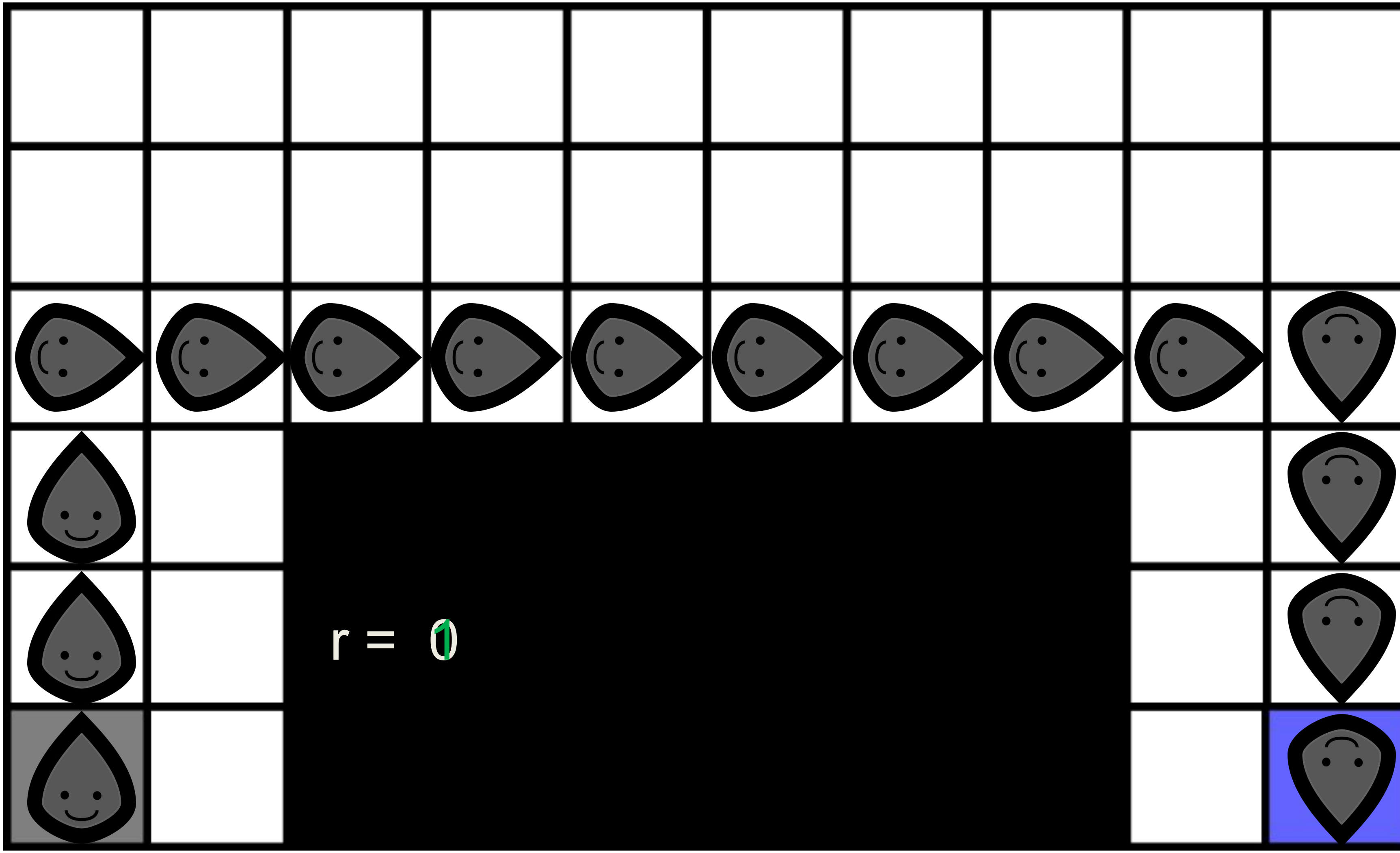
TempoRL: Learning When to Act

André Biedenkapp, Raghu Rajan,
Frank Hutter & Marius Lindauer

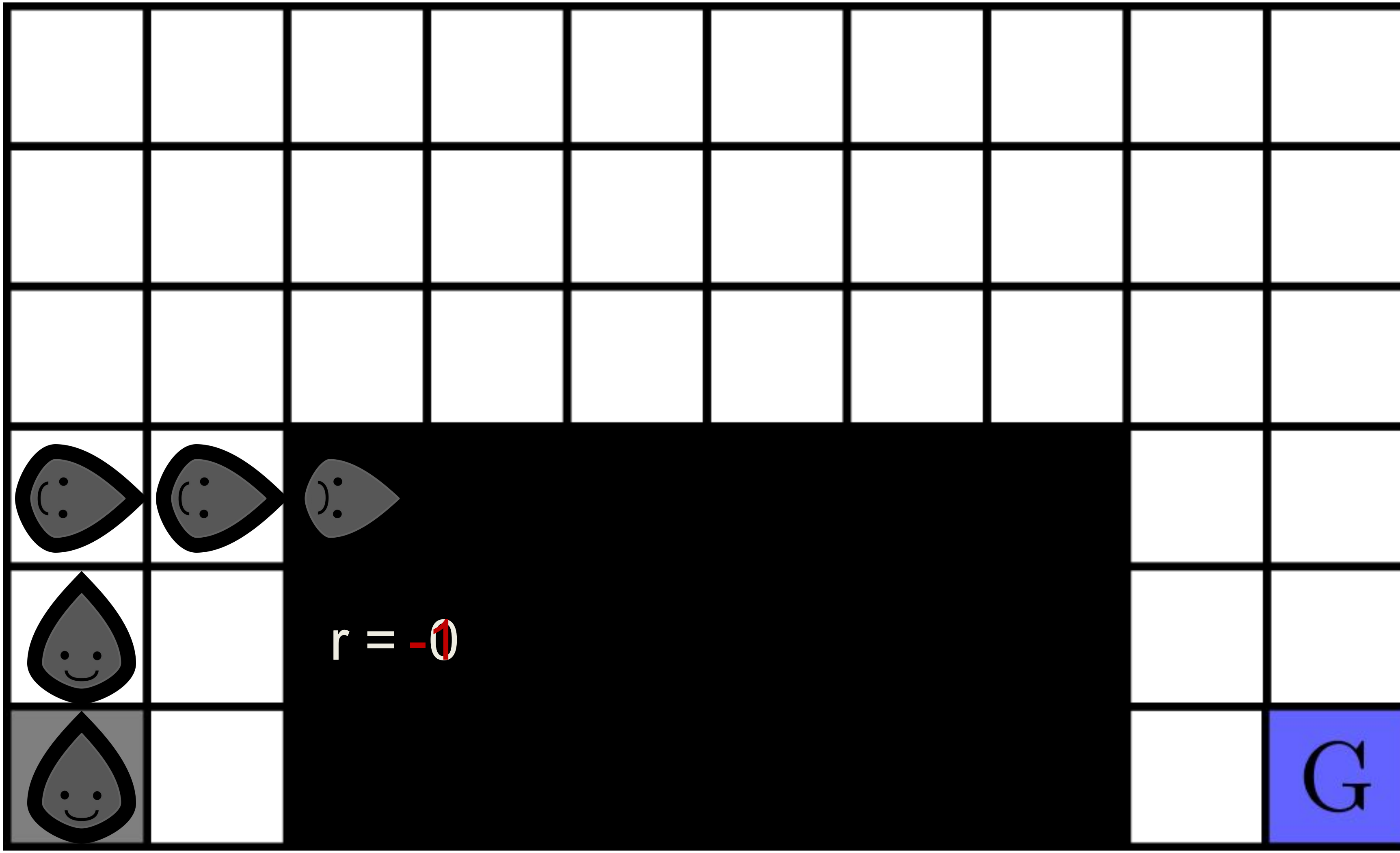
In a Nutshell

1. We propose a proactive way of doing RL
2. We introduce skip-connections into MDPs
   - use of action repetition
   - faster propagation of rewards
3. We propose a novel algorithm using skip-connections
   - learn *what* action to take & *when* to make a new decision
   - condition *when* on *what*
4. We evaluate our approach with in a variety of settings
   - tabular Q-learning on Gridworlds
   - DQN on featurized environments
   - DDPG on featurized environments
   - DQN with image states on Atari environments

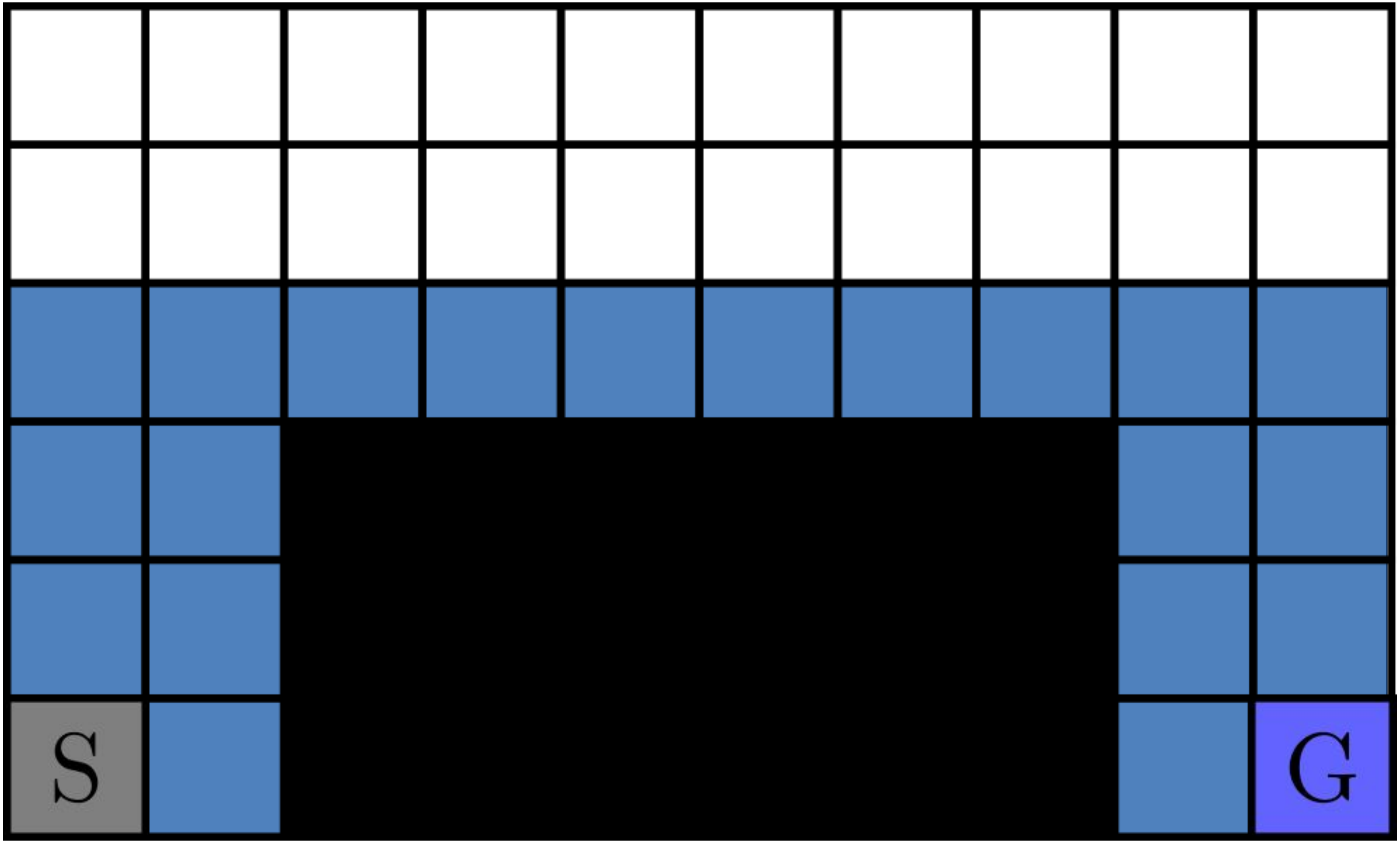
Motivation



Motivation

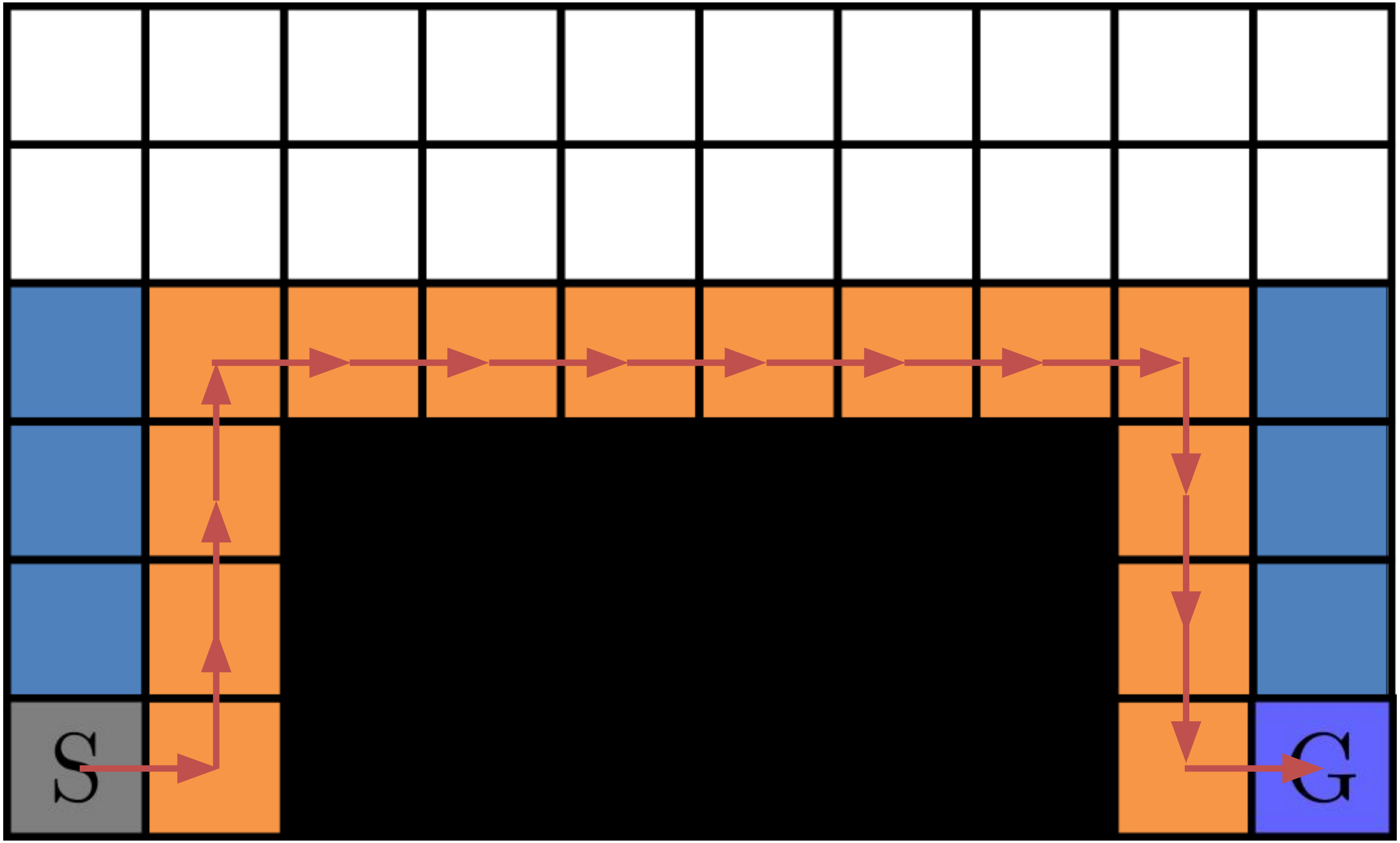


Optimal Policies



- Optimal policies will only cross the blue shaded area.

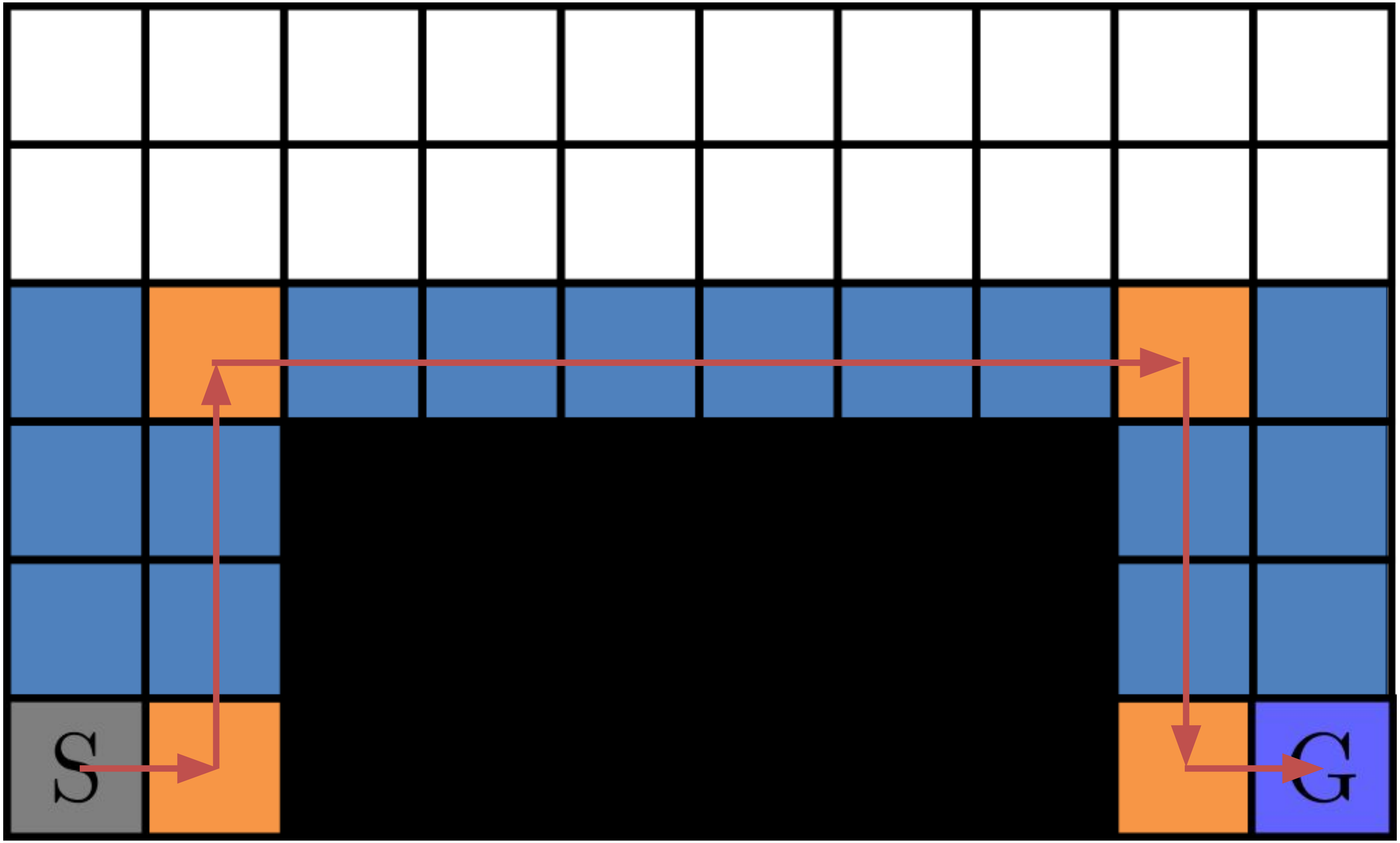
Optimal Policies: When do we need to act?



- Example trajectory of an optimal policy requiring

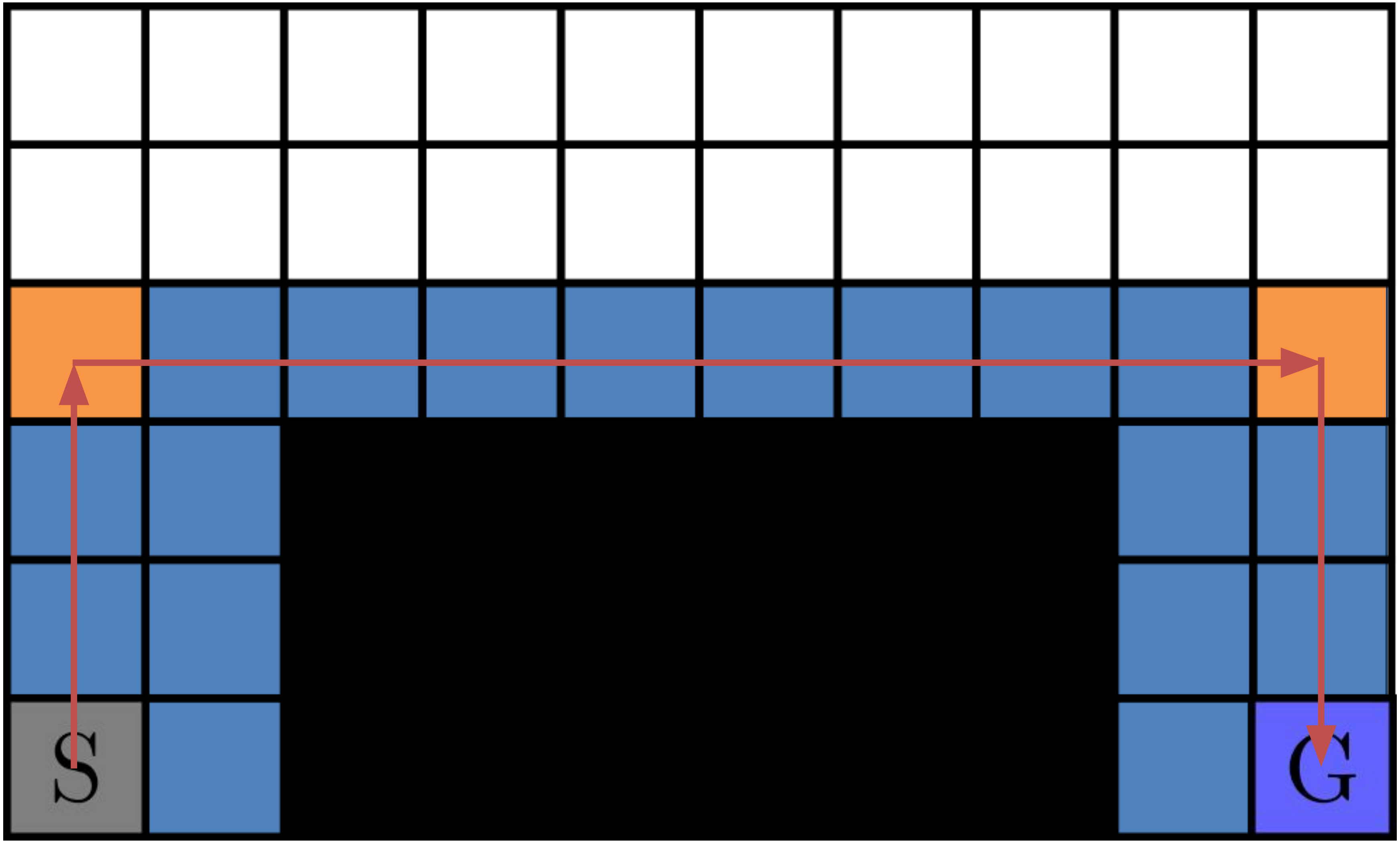
Steps: 16
Decisions: 16

Optimal Policies: When do we need to act?



- Simplified trajectory of an optimal policy requiring
  # Steps: 16
  # Decisions: **5**

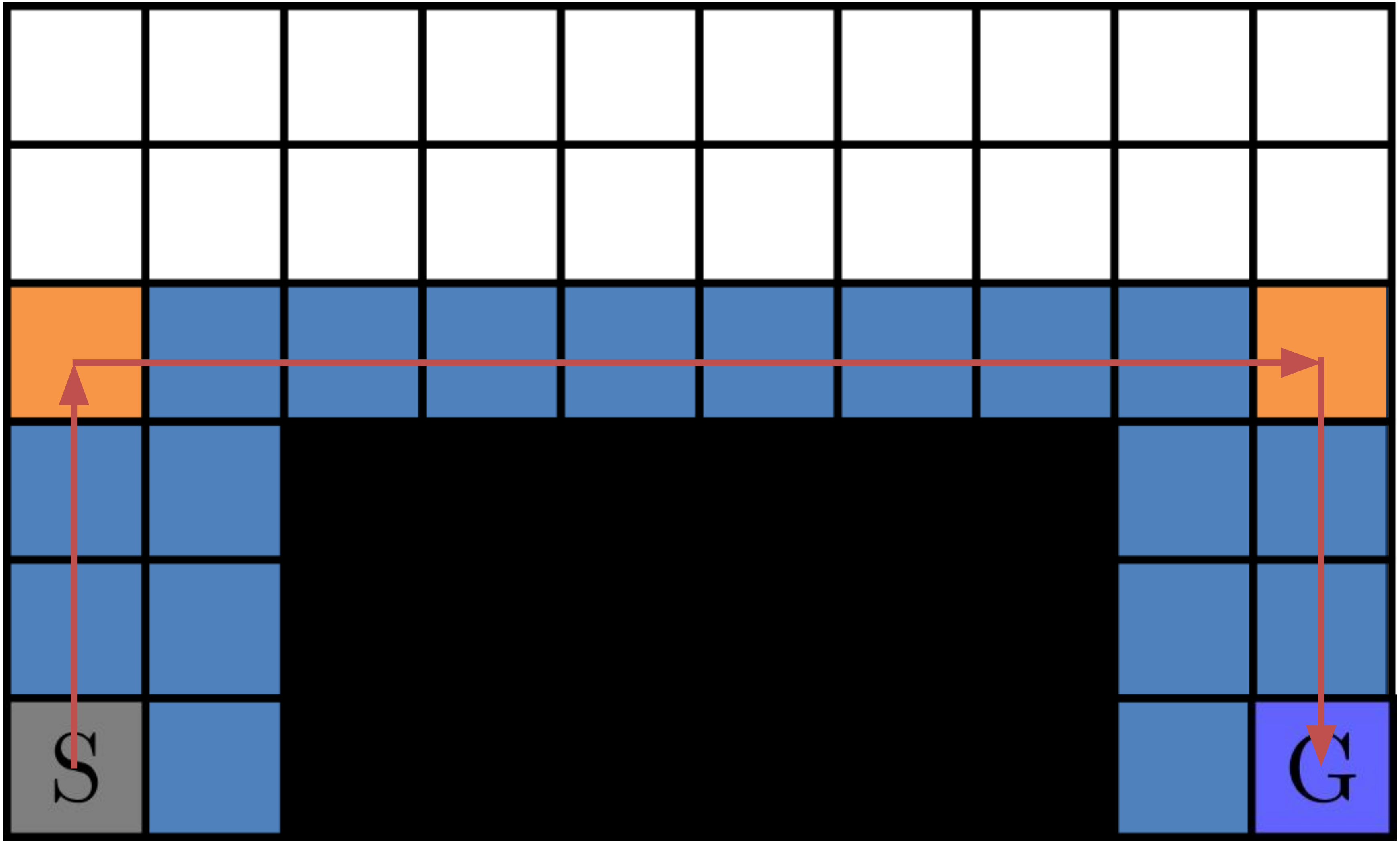
Optimal Policies: When do we need to act?



- Simplified trajectory of an optimal policy requiring

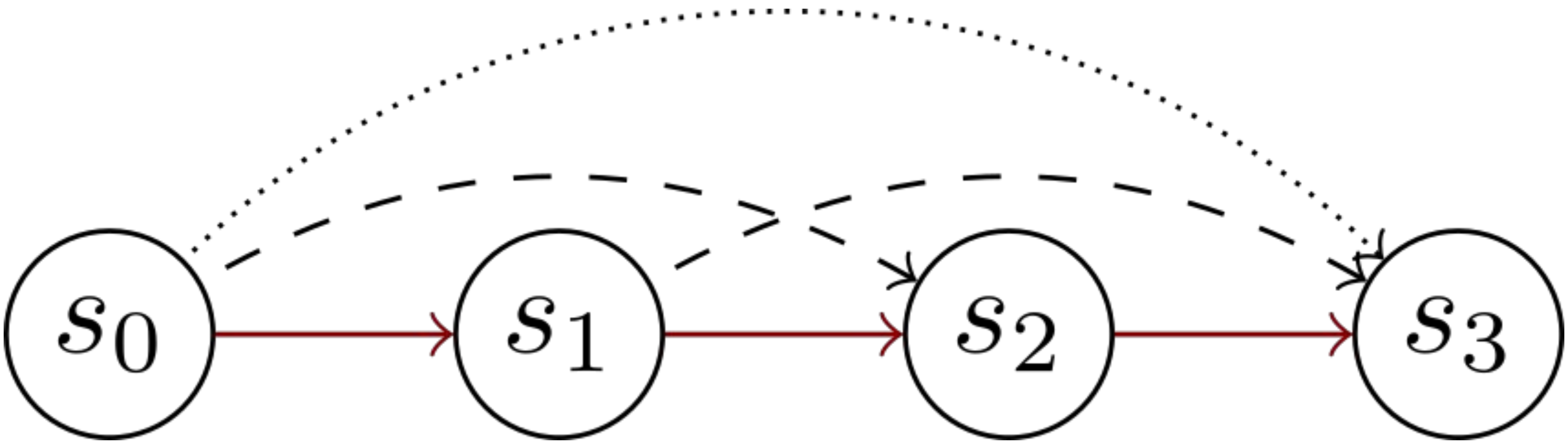
Steps: 16
Decisions: **3**

Optimal Policies: When do we need to act?



- Proactive decision making requires ~**80% fewer decisions**
- Much simpler policies

Skip MDPs



- Action repetition induces skips
- Information can be propagated faster along skips
- With large skips, multiple smaller skips can be observed

Flat Hierarchy

1. Use standard agent (e.g. Q-learning) to determine the behaviour given the state

$$Q^{\pi}(s_t, a) \longrightarrow a$$

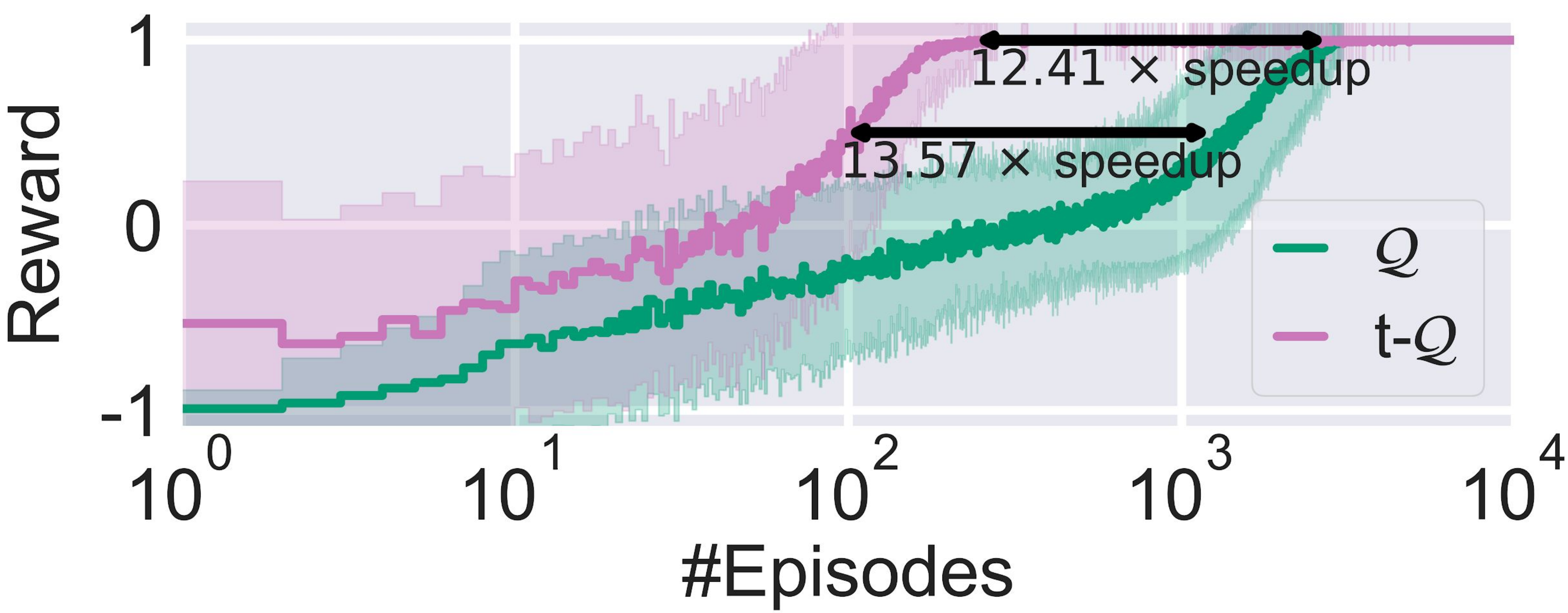
2. Condition skips on the chosen action

$$Q^{\pi_j}(s_t, j|a) \longrightarrow j$$

3. Play action a for the next j steps

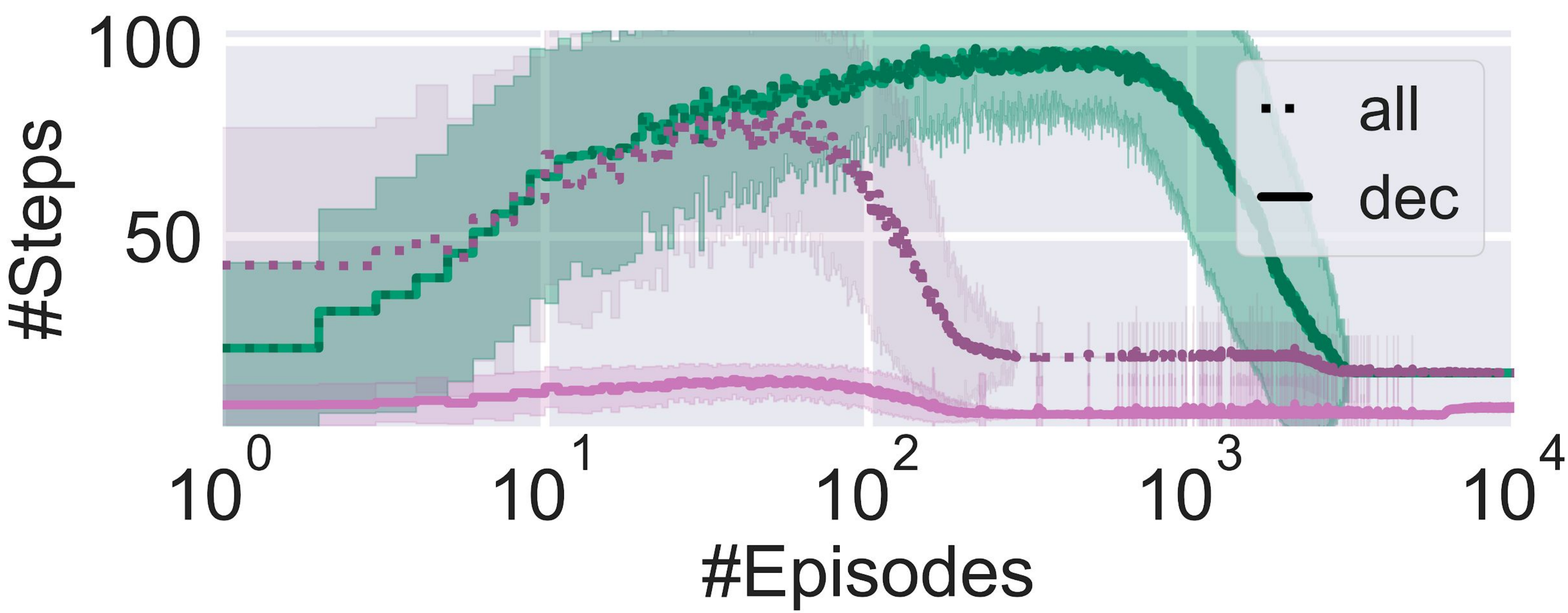
- Behaviour policy can be learned with vanilla agents
- The skip Q-function can be learned using n-step updates

- Comparison of vanilla and TempoRL Q-learning on the example gridworld



- TempoRL learns well performing policies faster

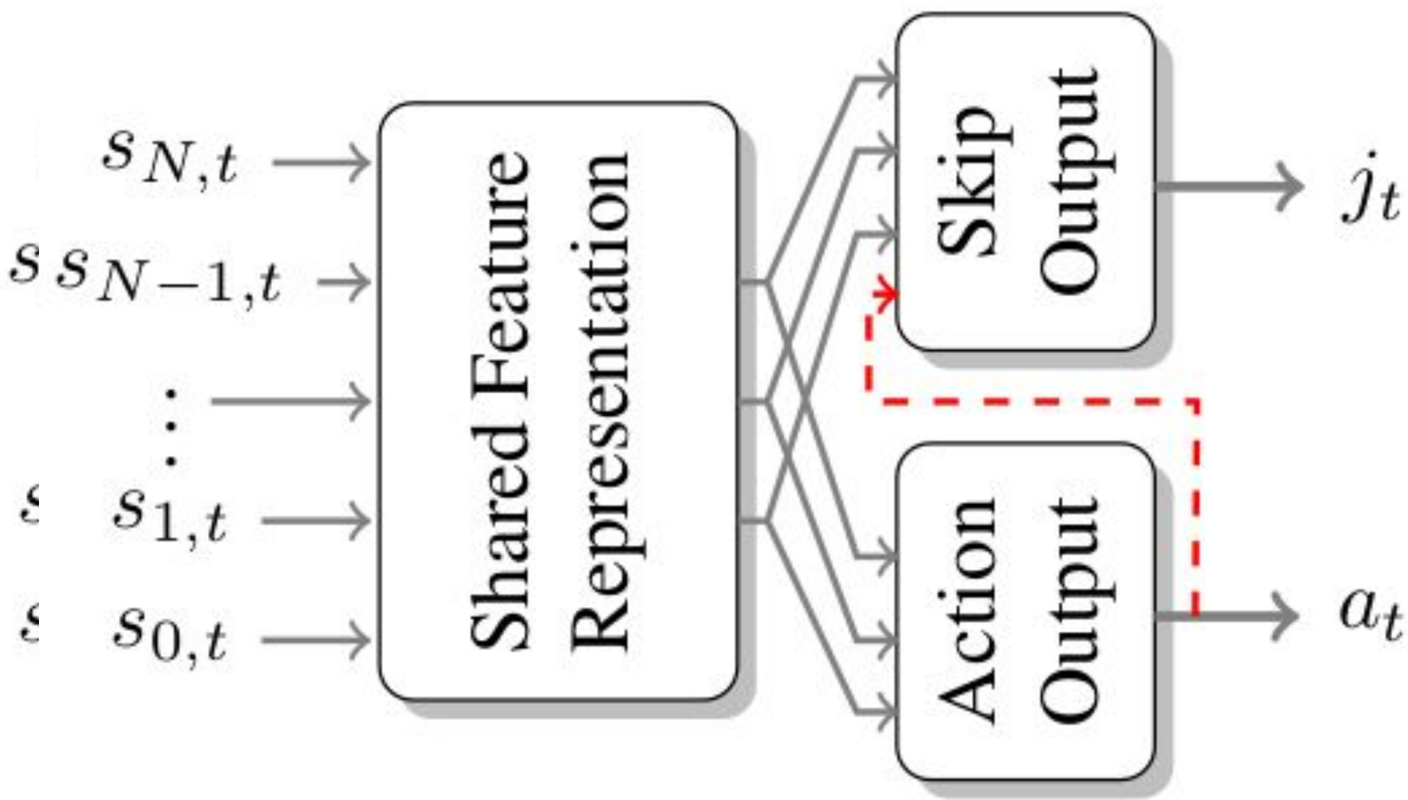
- Comparison of vanilla and TempoRL Q-learning on the example gridworld



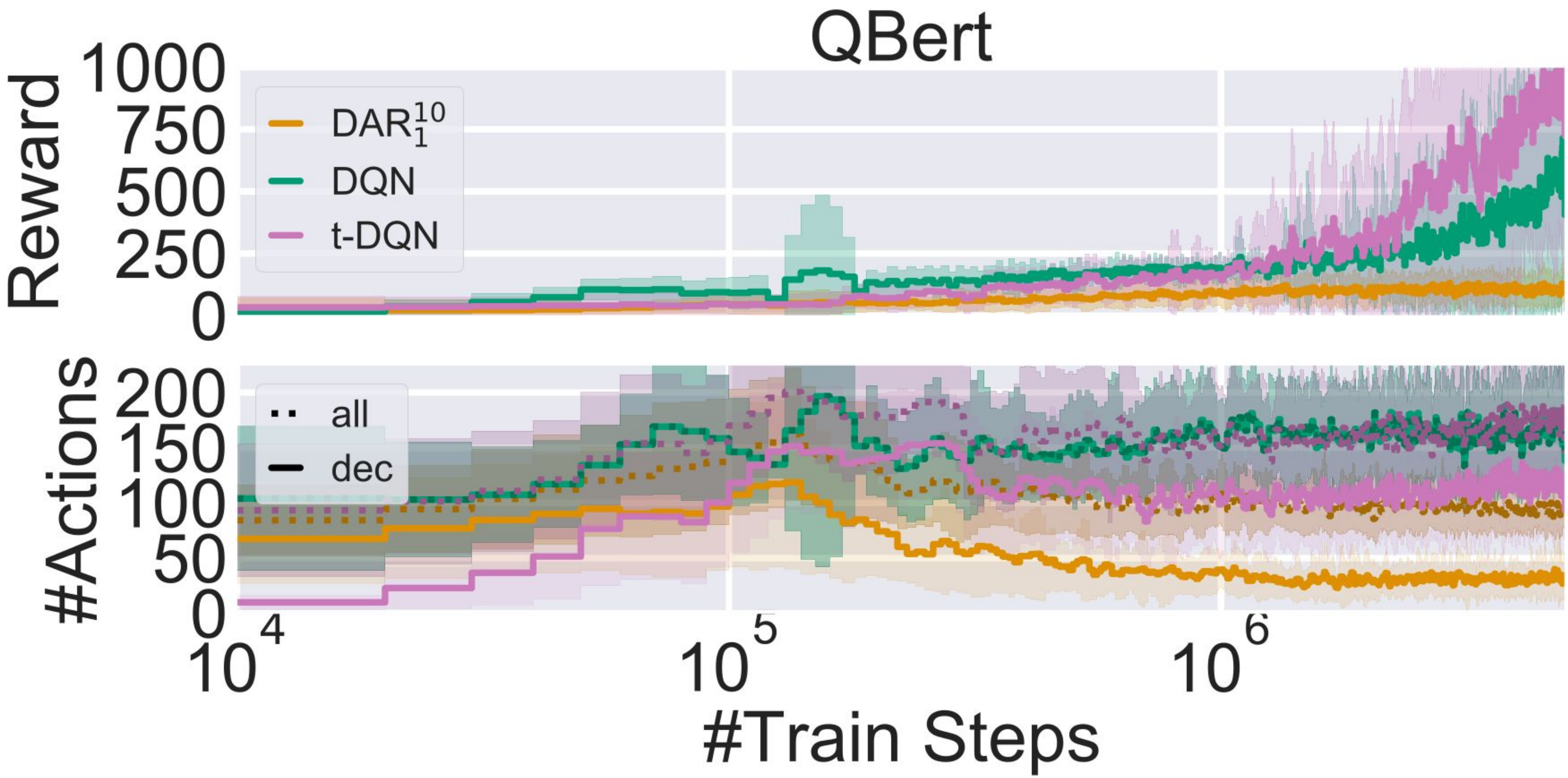
- TempoRL learns well performing policies faster requiring far fewer decisions by learning *when* to switch actions

Moving to Deep RL

Depending on the state modality we consider different architectures



Evaluation on Atari



Wrap-Up

- TempoRL allows for
  - better exploration
  - faster learning
  - better explainability

Code, learned policies, videos of rollotus and learning curves are available at

https://github.com/automl/TempoRL

- Further results in the paper
  - TempoRL DDPG
  - Influence of TempoRL hyperparameters
  - Improved exploration through TempoRL

- Future Work
  - distributional TempoRL
  - changing TempoRL exploration

Looking forward to meeting you at the poster!