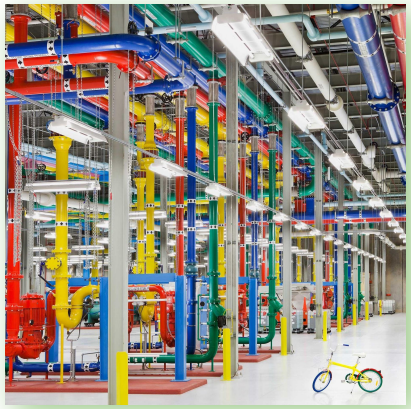
ConQUR: Mitigating Delusional Bias in Deep Q-Learning

DiJia (Andy) Su (Princeton)
Jayden Ooi (Google)
Tyler Lu (Google)
Dale Schuurmans (Google)
Craig Boutilier (Google)

Motivation

- Q-Learning is at the heart of many recent deep RL successes



Overview of Key Contributions

Delusional bias *(Lu, et al., NeurIPS18)* can occur in any Q-Learning method

ConQUR (Consistent Q-Update Regression)

- A new framework that mitigates delusion **in a scalable fashion**

  1) Consistency Penalty

  2) Framework to search for the best Q-regressors

Overview of Key Contributions

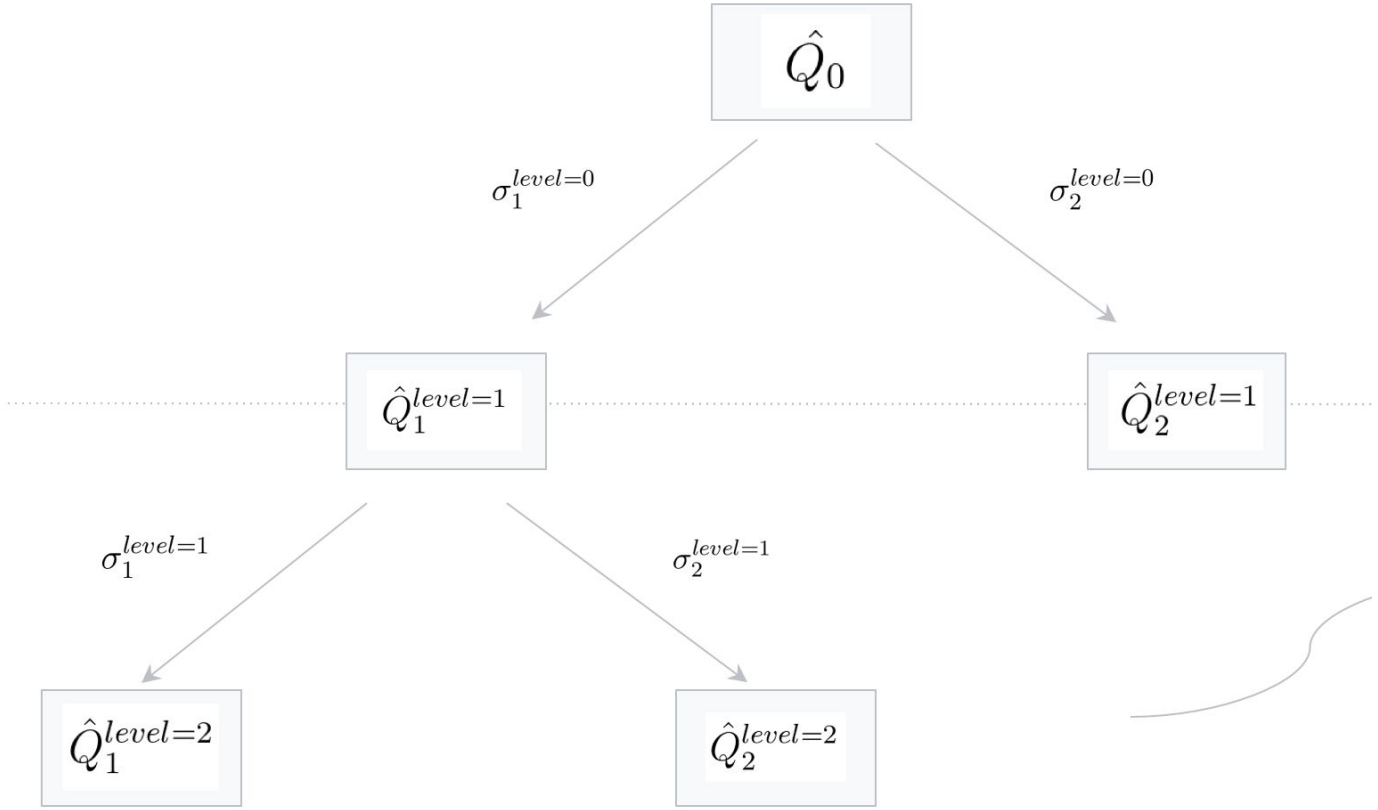
1) Consistency Penalty

- augments Bellman loss, very easy to implement

$$\underbrace{\sum_i [q_i - Q_{new}(s_i, a_i)]^2}_{\text{Bellman Loss}} + \lambda \underbrace{\sum_i \sum_{b \in A} [Q_{new}(s'_i, b) - Q_{new}(s'_i, a'_i)]_+}_{\text{Consistency Penalty}}$$

Overview of Key Contributions

2) Framework to **search** for the best Q-regressors

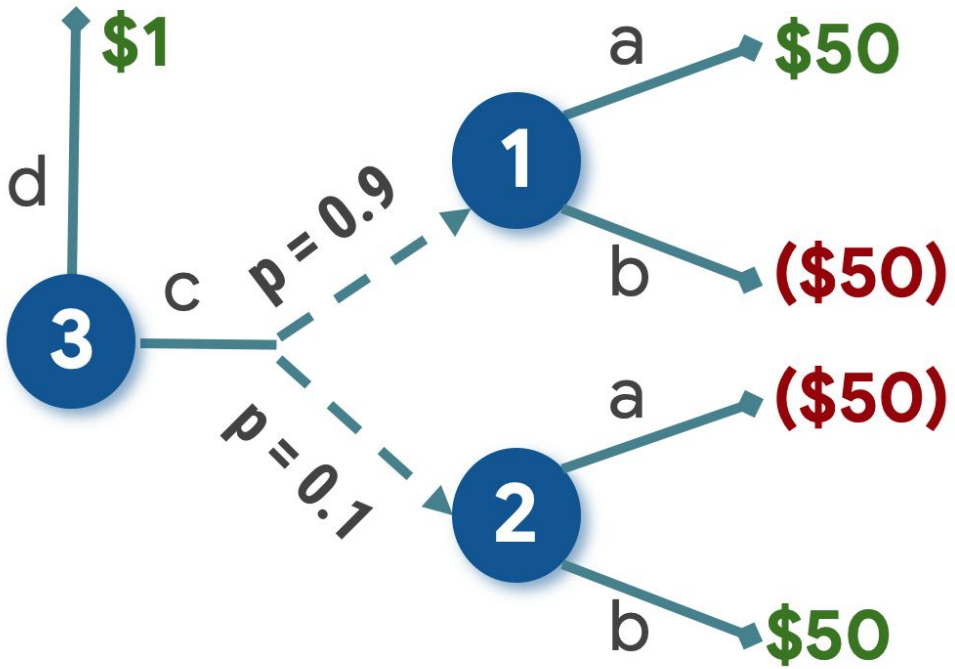


Background

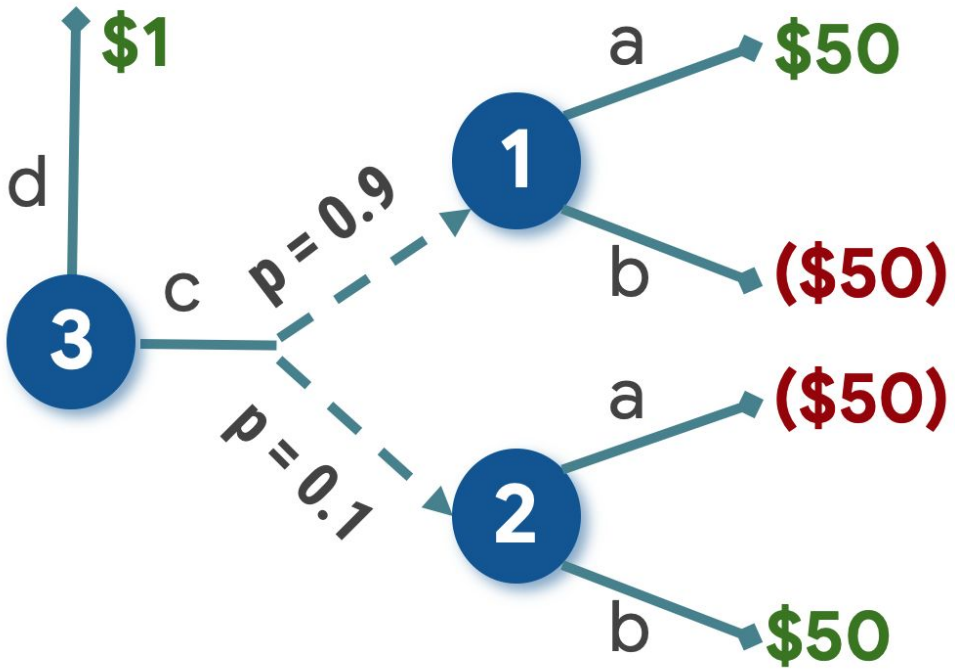
Delusional Bias (*Lu et al, NeurIPS 18*):

Arises when Q learning combines with approximator, and there is no greedy policy that can execute the state action choices.

Greedy Policy w.r.t. an Approximator



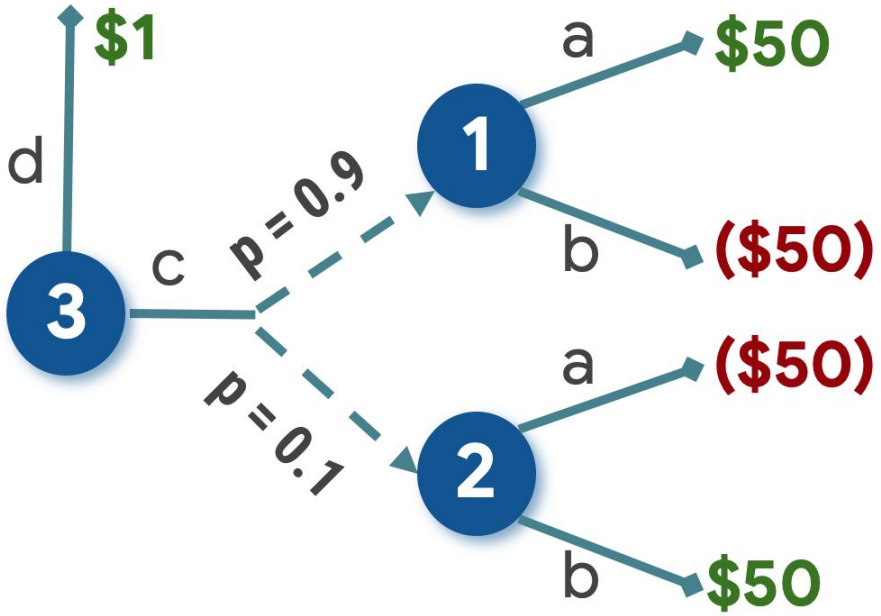
Greedy Policy w.r.t. an Approximator



Features for Linear Approximation

$\phi(1\,a) = 1$	$\phi(2\,a) = 1$
$\phi(1\,b) = -1$	$\phi(2\,b) = -1$

Greedy Policy



$$Q(1,a) > Q(1,b) \iff Q(2,a) > Q(2,b)$$

Cannot do:

- 1 ⇒ a ***and*** 2 ⇒ b
- or vice versa

Combating Delusion

For complete delusion removal (*Lu, et al, NeurIPS 2018*), algorithm:

- Maintains full **information sets** (possible Q-functions)
- Requires comprehensive search
- Computationally intensive

We propose a much more **scalable** algorithm: **ConQUR**

- Consistency Penalty + Search

Combating Delusion: Soft-Consistency

Consistency Penalty enforces **soft-consistency**:

- Very easy to implement, no significant computational cost
- Can be used in isolation (without search)

$$\underbrace{\sum_i [q_i - Q_{new}(s_i, a_i)]^2}_{\text{Bellman Loss}} + \underbrace{\lambda \sum_i \sum_{b \in A} [Q_{new}(s'_i, b) - Q_{new}(s'_i, a'_i)]_+}_{\text{Consistency Penalty}}$$

Combating Delusion: Heuristic Search

Search Framework:

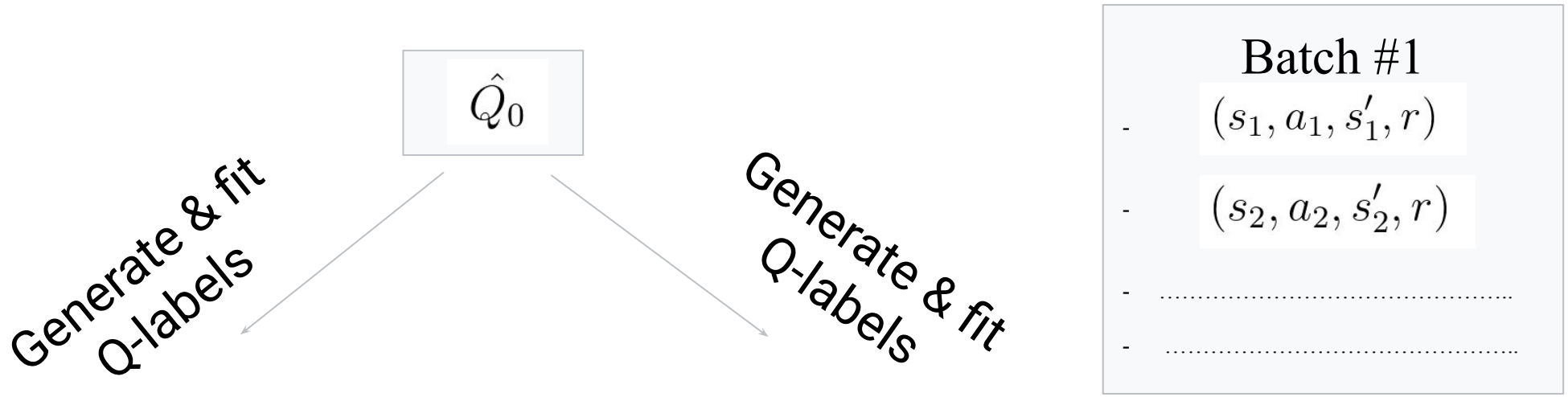
- Maintain and search over multiple information sets
- Each information set represents policy commitments and Q-regressor

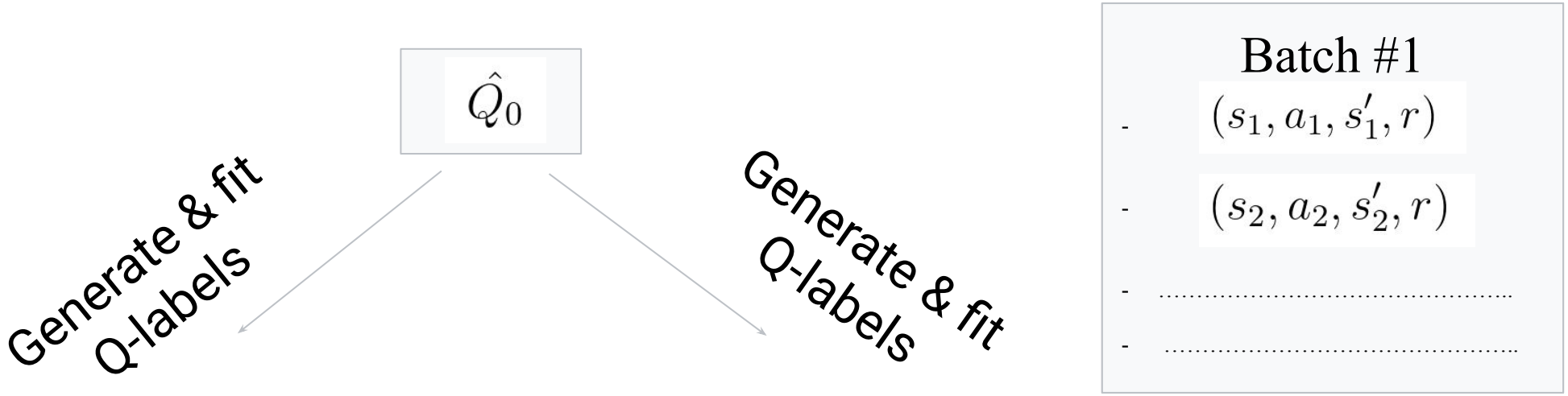
Search: Q-Regressors

$\hat{Q}_0$

Batch #1

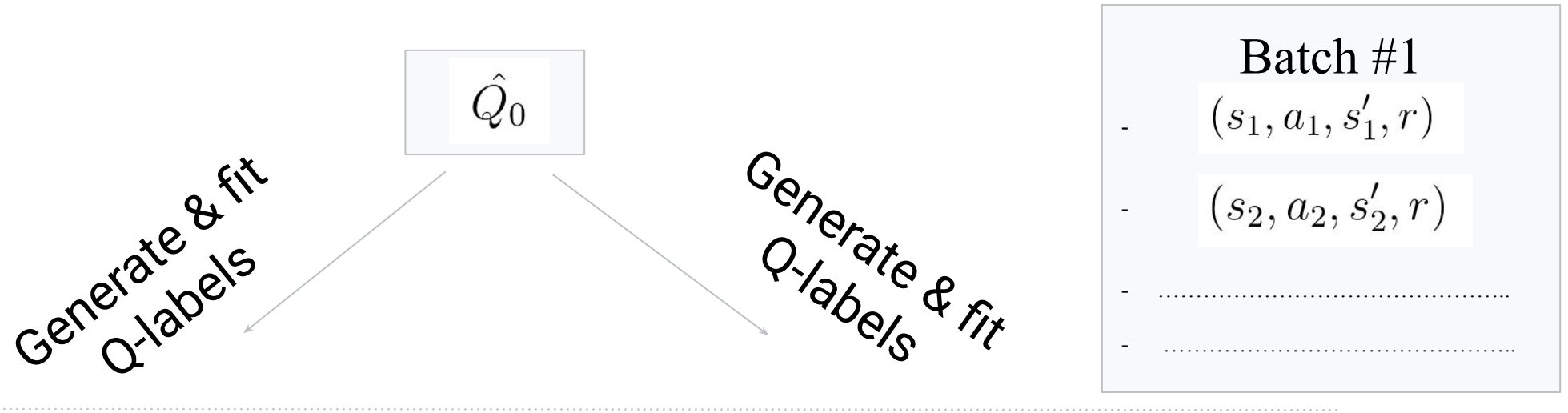
- (s_1, a_1, s'_1, r)
- (s_2, a_2, s'_2, r)
- .....
- .....


$\hat{Q}_0$
Generate & fit Q-labels
Generate & fit Q-labels
Batch #1
- (s_1, a_1, s_1', r)
- (s_2, a_2, s_2', r)
- .....
- .....



Generating Q-labels

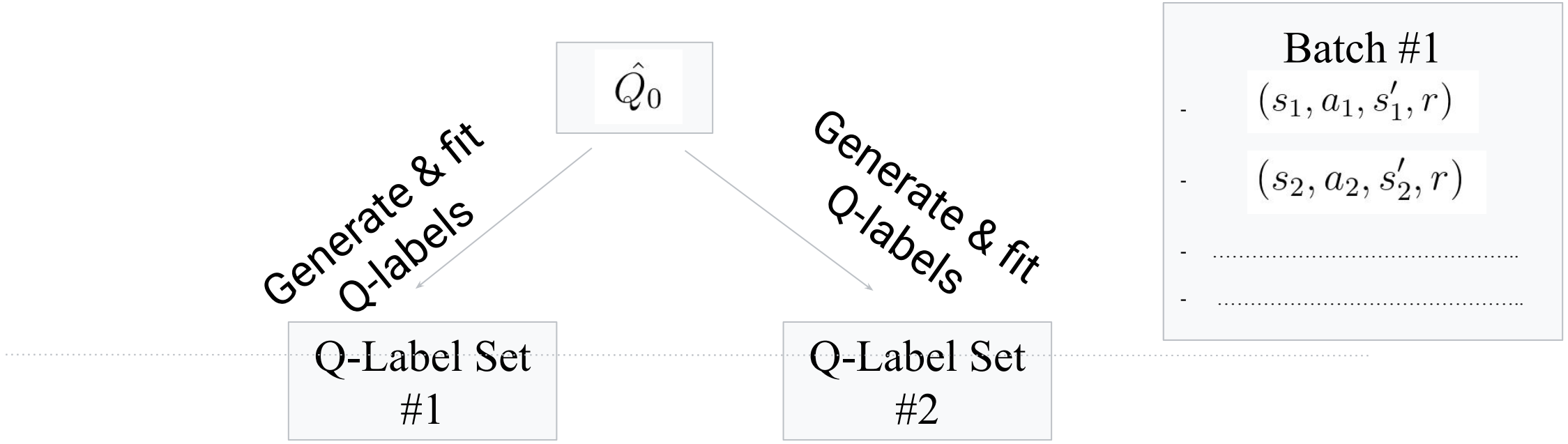
$$q_i \leftarrow r_i + \gamma Q_{\text{old}}(s'_i, a'_i)$$



Generating Q-labels

$$q_i \leftarrow r_i + \gamma Q_{\text{old}}(s'_i, a'_i)$$

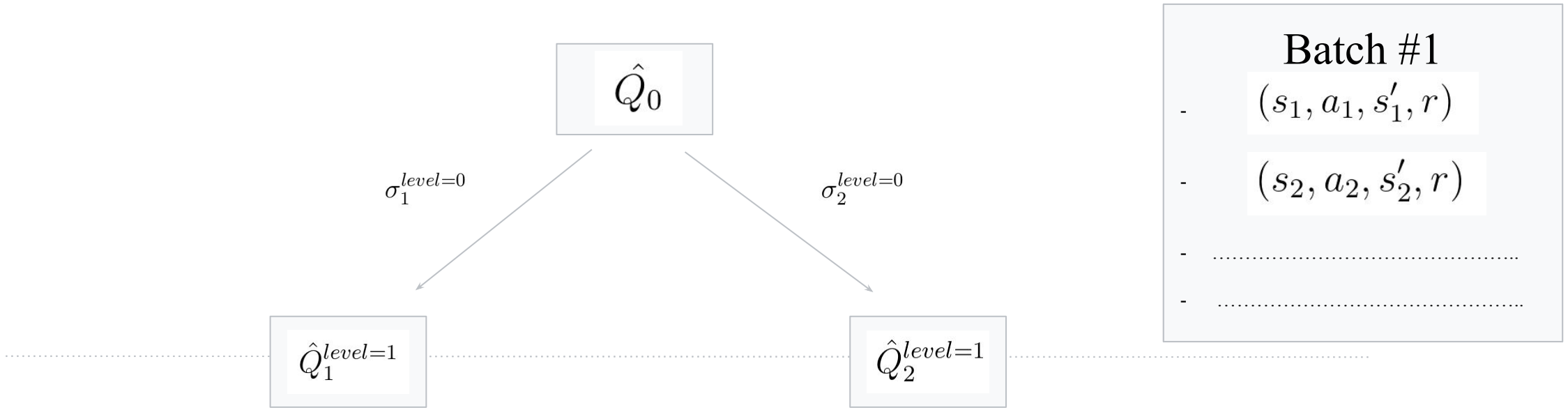
where $a'_i \sim e^{\tau \hat{Q}_0(s'_i, a'_i)}$



Generating Q-labels

$$q_i \leftarrow r_i + \gamma Q_{\text{old}}(s'_i, a'_i)$$

where $a'_i \sim e^{\tau \hat{Q}_0(s'_i, a'_i)}$



Policy commitments

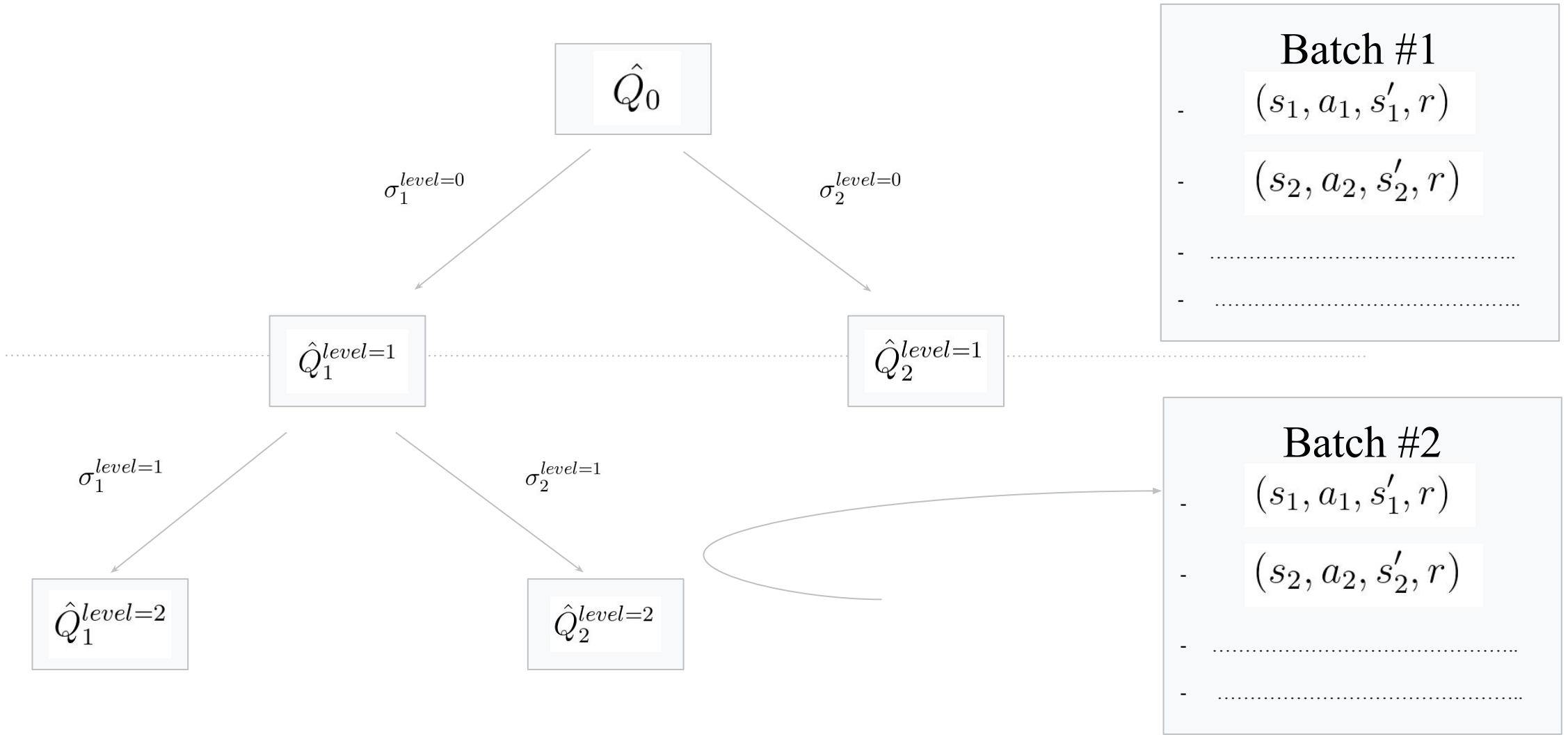
$\sigma_1^{level=0}$ $\sigma_2^{level=0}$

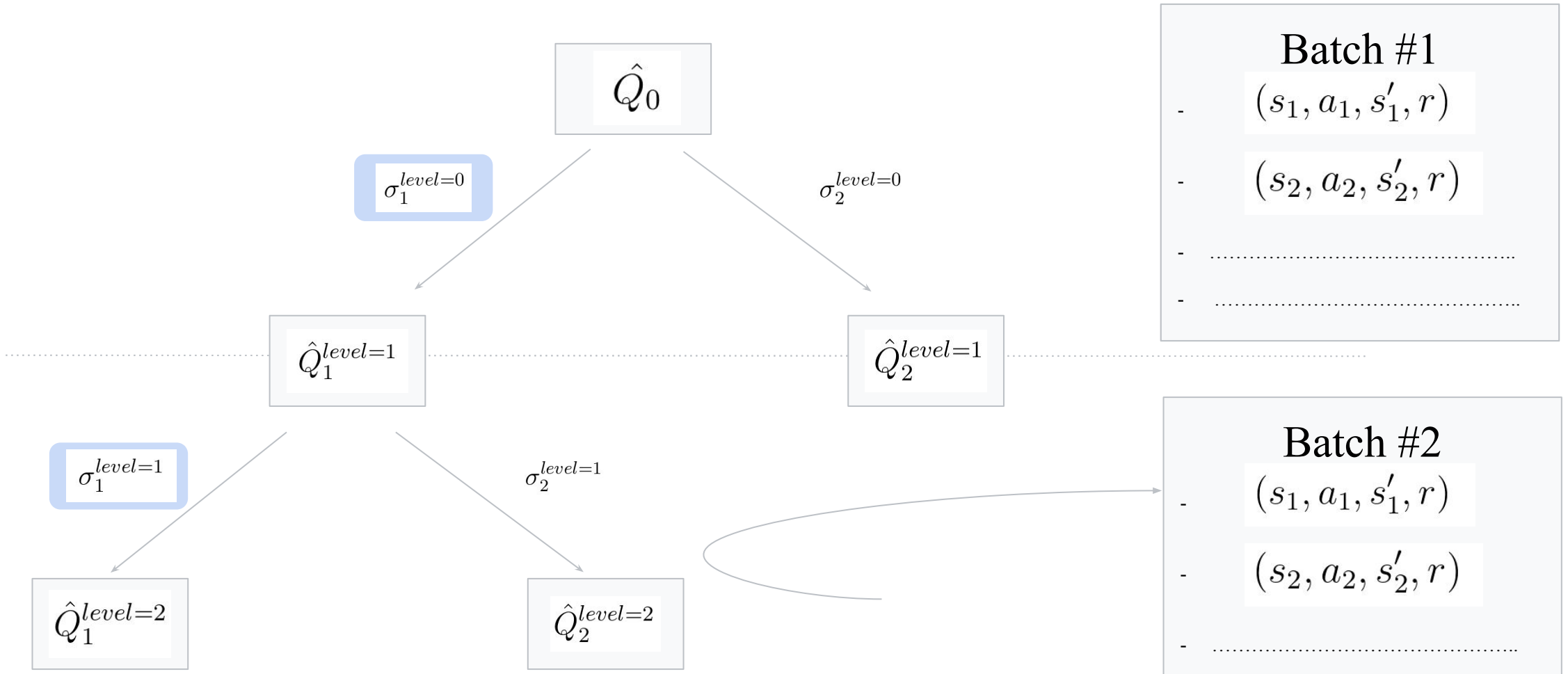
Consistency buffer #1

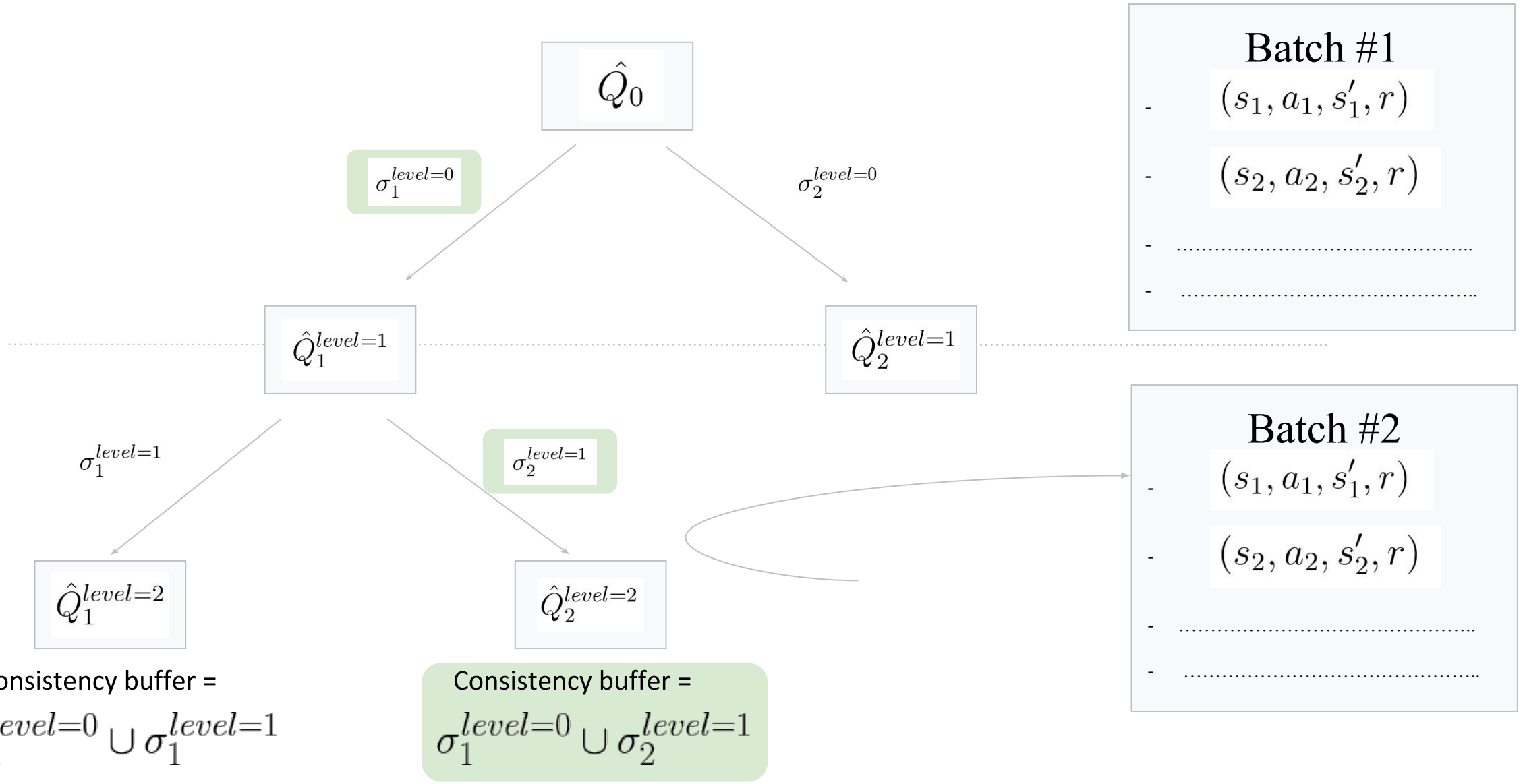
$$\sigma_1(s_1') = a_1'$$
$$\sigma_1(s_2') = a_2'$$

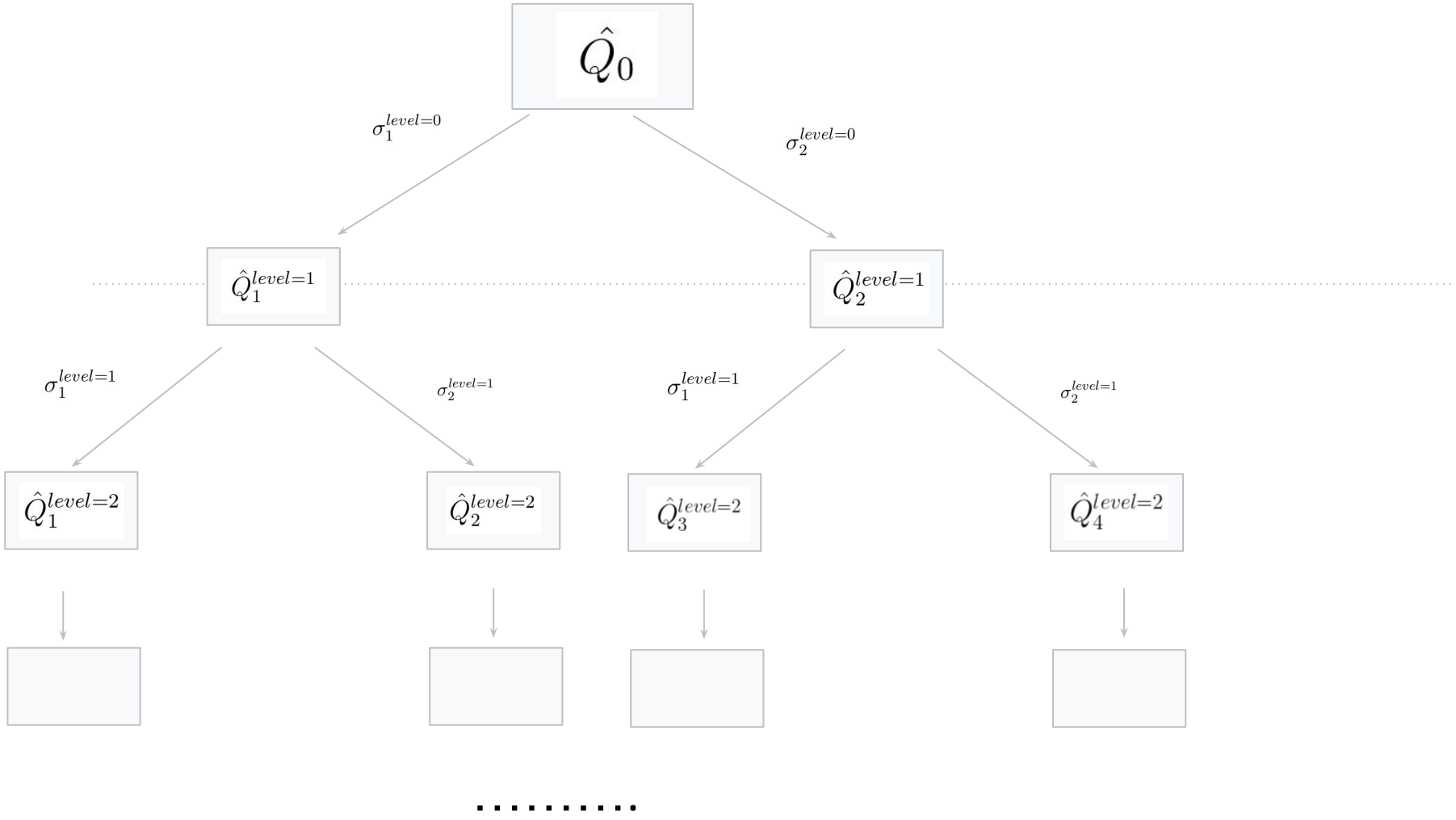
$$\sigma_1(s_1') = b_1'$$
$$\sigma_1(s_2') = b_2'$$

Consistency buffer #2


$\hat{Q}_0$
$\sigma_1^{level=0}$
$\sigma_2^{level=0}$
$\hat{Q}_1^{level=1}$
$\hat{Q}_2^{level=1}$
$\sigma_1^{level=1}$
$\sigma_2^{level=1}$
$\hat{Q}_1^{level=2}$
$\hat{Q}_2^{level=2}$
Batch #1
- (s_1, a_1, s_1', r)
- (s_2, a_2, s_2', r)
- .....
- .....
Batch #2
- (s_1, a_1, s_1', r)
- (s_2, a_2, s_2', r)
- .....
- .....


$\hat{Q}_0$
$\sigma_1^{level=0}$
$\sigma_2^{level=0}$
$\hat{Q}_1^{level=1}$
$\hat{Q}_2^{level=1}$
$\sigma_1^{level=1}$
$\sigma_2^{level=1}$
$\hat{Q}_1^{level=2}$
$\hat{Q}_2^{level=2}$
Batch #1
- (s_1, a_1, s_1', r)
- (s_2, a_2, s_2', r)
- .....
- .....
Batch #2
- (s_1, a_1, s_1', r)
- (s_2, a_2, s_2', r)
- .....
- .....
Consistency buffer =
$\sigma_1^{level=0} \cup \sigma_1^{level=1}$


$\hat{Q}_0$
$\sigma_1^{level=0}$
$\sigma_2^{level=0}$
$\hat{Q}_1^{level=1}$
$\hat{Q}_2^{level=1}$
$\sigma_1^{level=1}$
$\sigma_2^{level=1}$
$\hat{Q}_1^{level=2}$
$\hat{Q}_2^{level=2}$
Batch #1
- (s_1, a_1, s_1', r)
- (s_2, a_2, s_2', r)
- .....
- .....
Batch #2
- (s_1, a_1, s_1', r)
- (s_2, a_2, s_2', r)
- .....
- .....
Consistency buffer =
$\sigma_1^{level=0} \cup \sigma_1^{level=1}$
Consistency buffer =
$\sigma_1^{level=0} \cup \sigma_2^{level=1}$


$\hat{Q}_0$
$\sigma_1^{level=0}$
$\sigma_2^{level=0}$
$\hat{Q}_1^{level=1}$
$\hat{Q}_2^{level=1}$
$\sigma_1^{level=1}$
$\sigma_2^{level=1}$
$\sigma_1^{level=1}$
$\sigma_2^{level=1}$
$\hat{Q}_1^{level=2}$
$\hat{Q}_2^{level=2}$
$\hat{Q}_3^{level=2}$
$\hat{Q}_4^{level=2}$

Search Strategy

- Selecting Q-regressor for:
  - Prioritized expansion
  - Quick quality evaluation

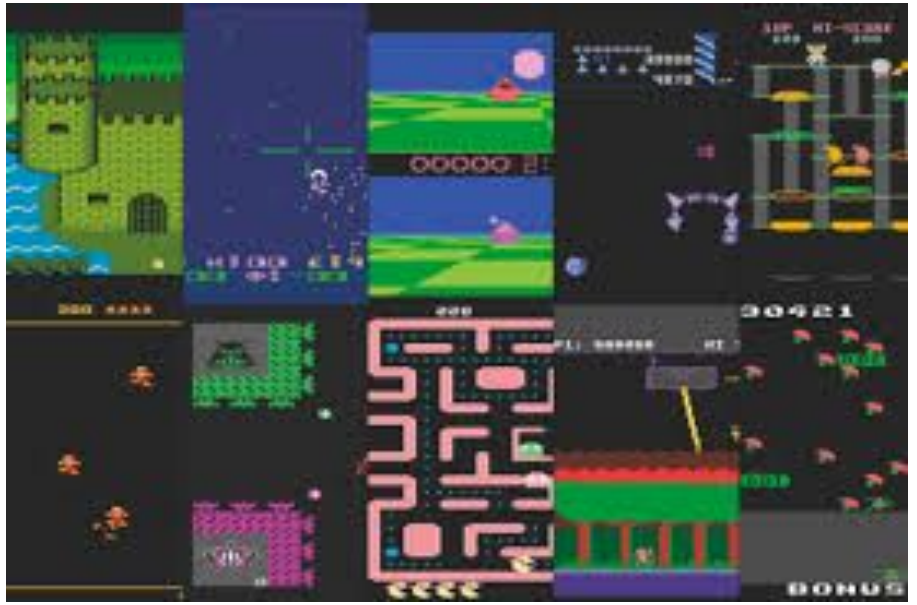
- Back-tracking

Experiments

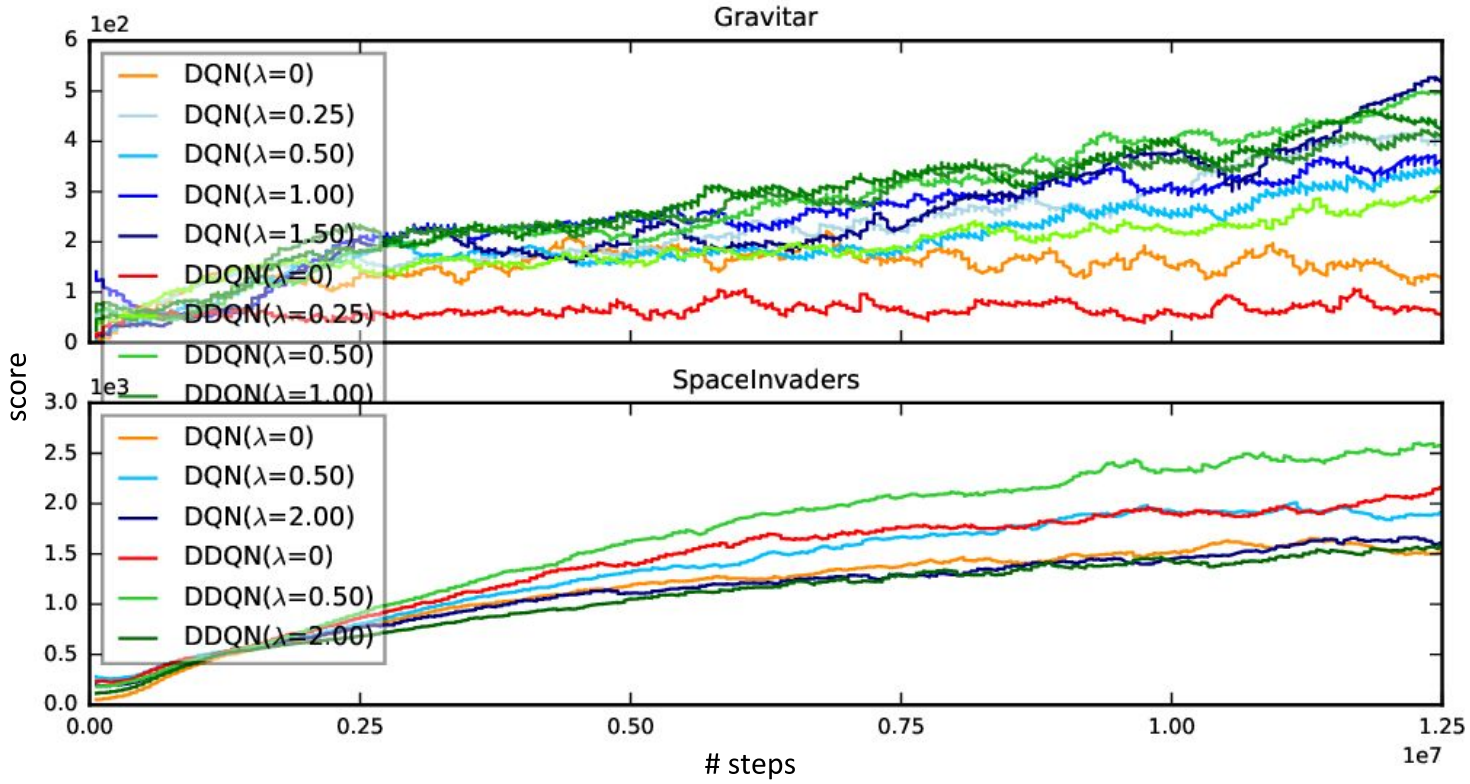
Two sets of experiments evaluated on Atari Suite (59 games):

1) Consistency Penalty
   - with DQN and DDQN

2) Full ConQUR framework



Consistency Penalty Experiment



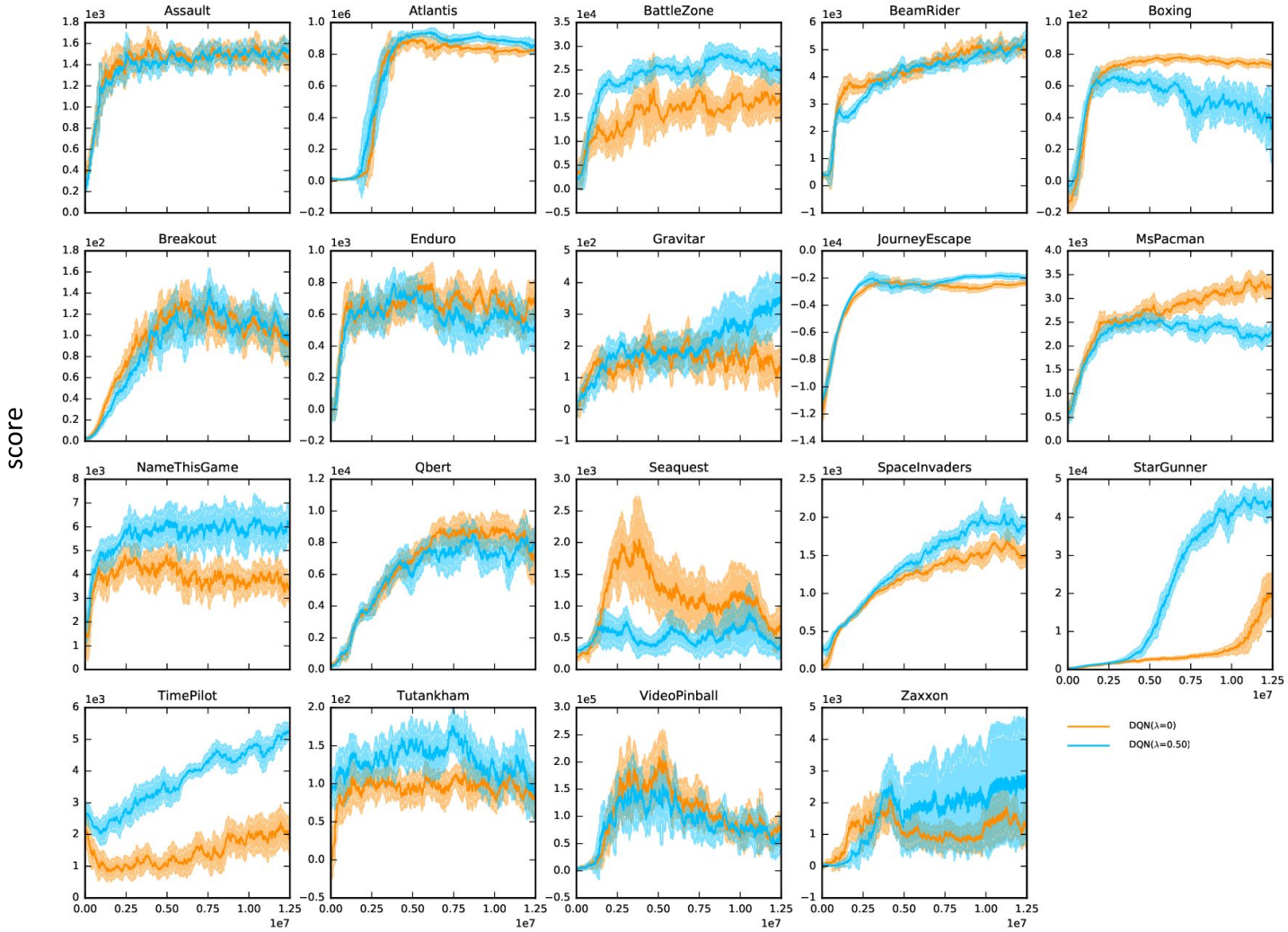
Provides lift

- suggests reducing delusional bias helps performance

Sweet spot for λ:

- too low (not enough)
- too high (too stringent)

DQN vs {DQN + Consistency Penalty}



10 wins
3 losses
6 inconclusive
(95% CLs overlap)

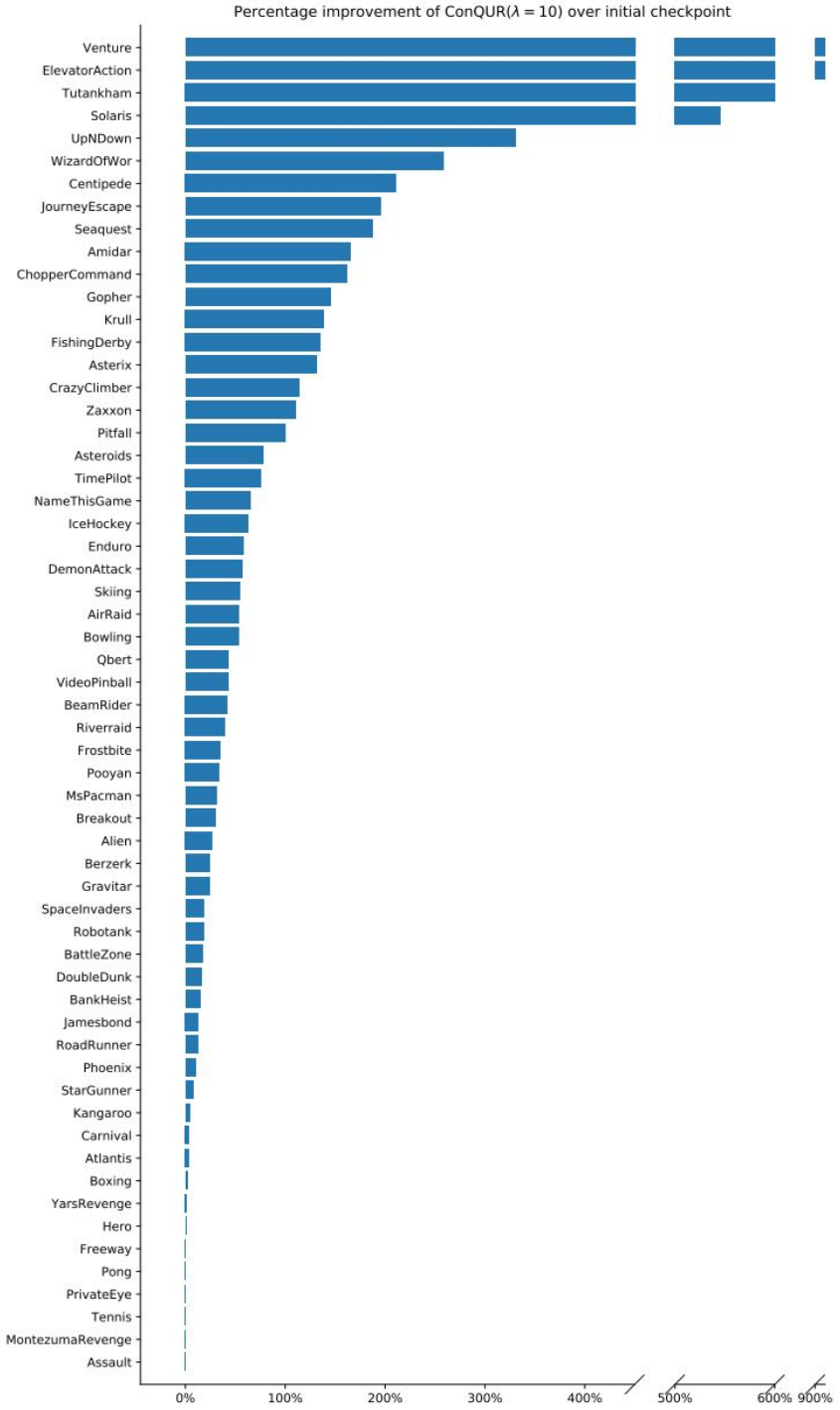
DQN(λ=0.5) can achieve gains that are greater than DDQN!

ConQUR Experiment

- ConQUR: Search + Consistency Penalty
  - Max node of the search tree = 16
- Baseline: Multi-DQN with 16 nodes
- Train on the last layer of a pretrained DQN.
  - only 4M frames vs 200M frames for full training

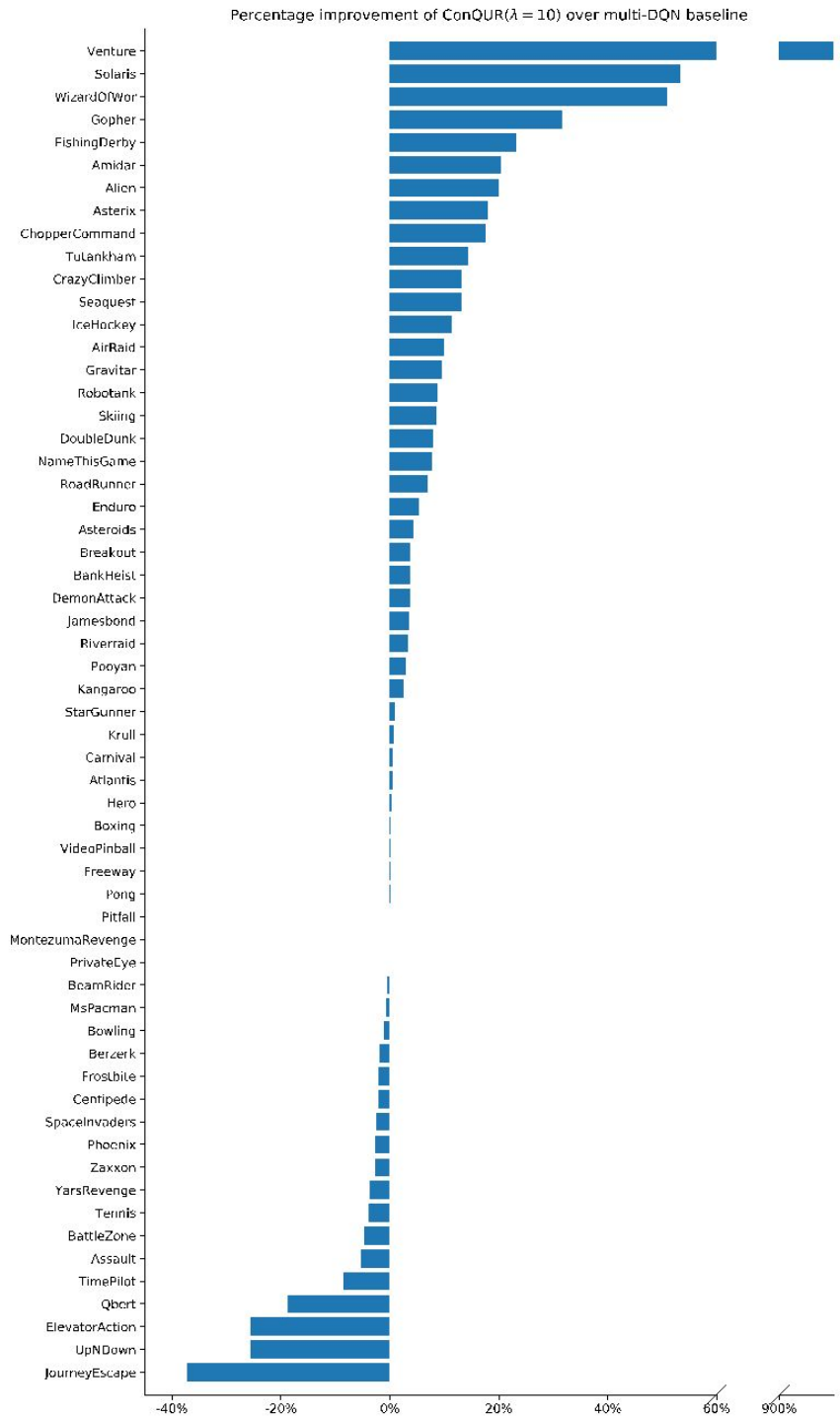
ConQUR Experiment

- An average improvement of 125% over the pretrained DQN

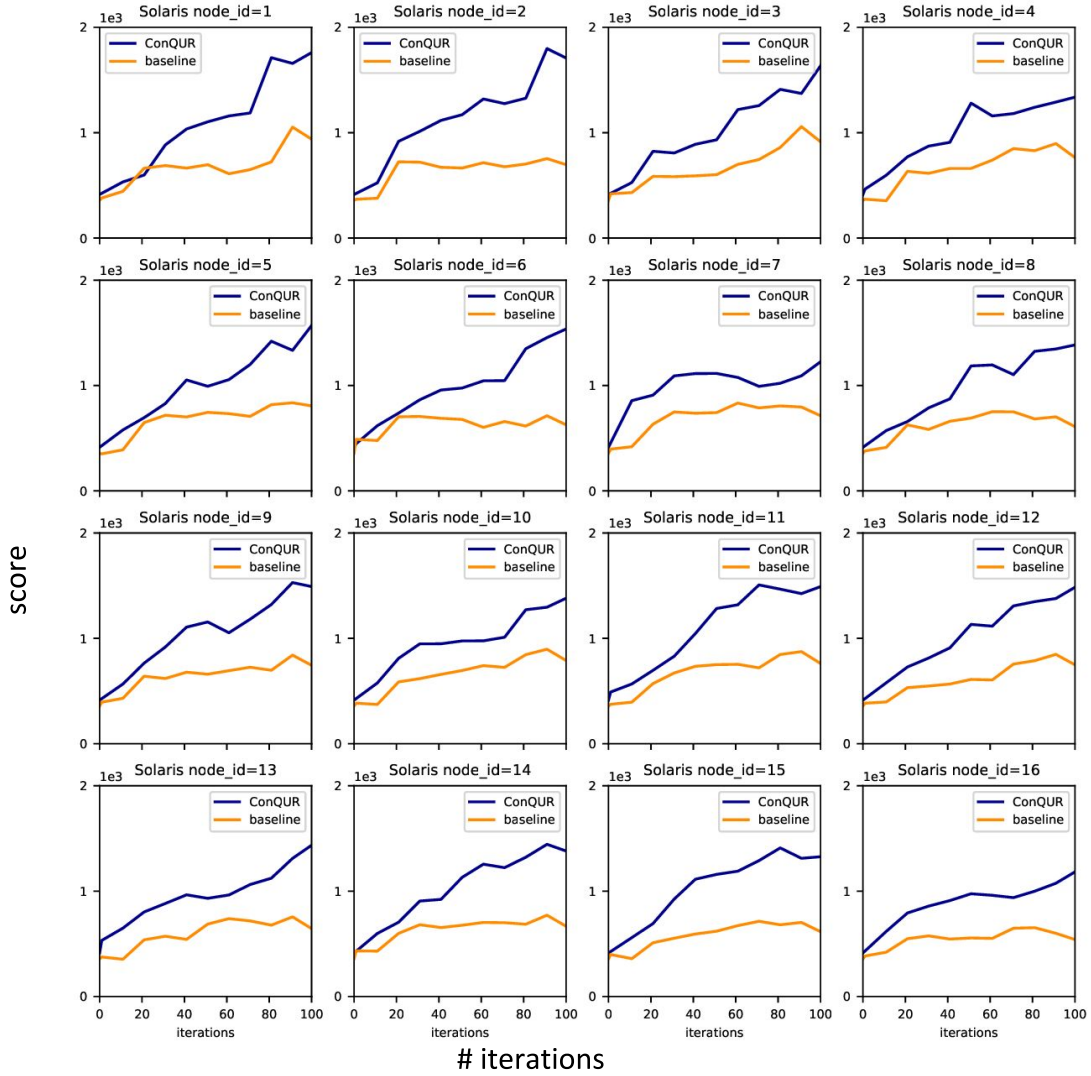


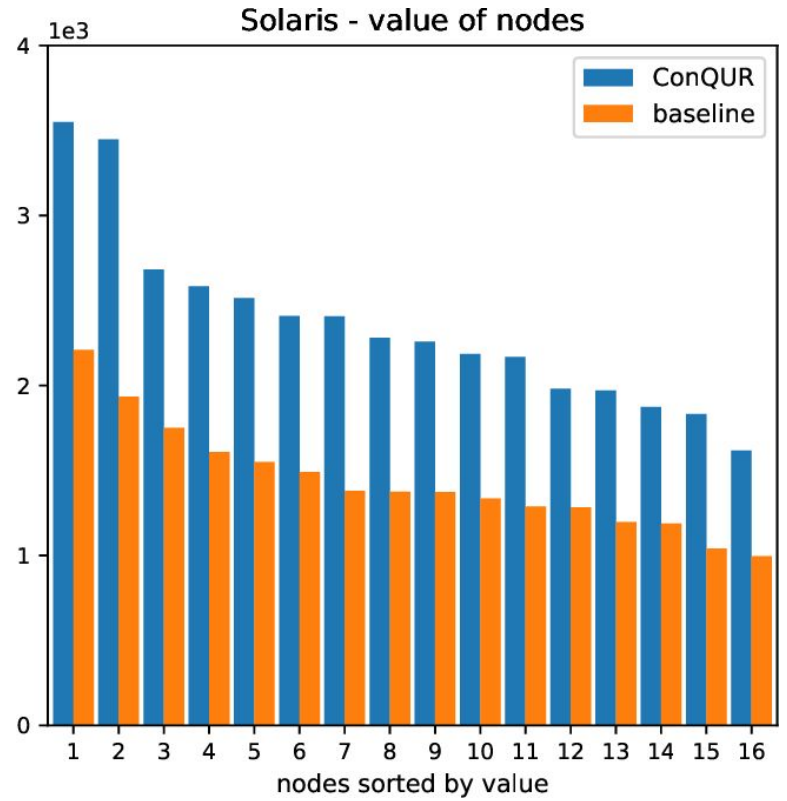
ConQUR Experiment

- Comparison of the ConQUR vs Multi-DQN baseline



ConQUR Experiment





Conclusion

ConQUR is a paradigm for mitigating delusion

- Easy to use consistency penalization
- Search space of multiple Q-regressors

Future Work:

- Explore new/alternative child generation strategies
- Explore connection to distributional RL