

Learning from suboptimal teachers

The role of compliance in the exploration-exploitation tradeoff

Final Presentation - Semester Project

Louis Faury

Under the supervision of :
Mahdi Khoramshahi and **Andrew Sutcliffe**
and
Prof. Billard

June 19, 2017

Outline

► INTRODUCTION

..... *Reinforcement Learning*

..... *Learning from Demonstration*

► MOTIVATIONS

► A COMPLIANCE-BASED APPROACH

..... *Intuition*

..... *Naive Compliant Learner*

..... *Adaptive Compliant Learners*

► RESULTS

► CONCLUSION

Introduction

Introduction

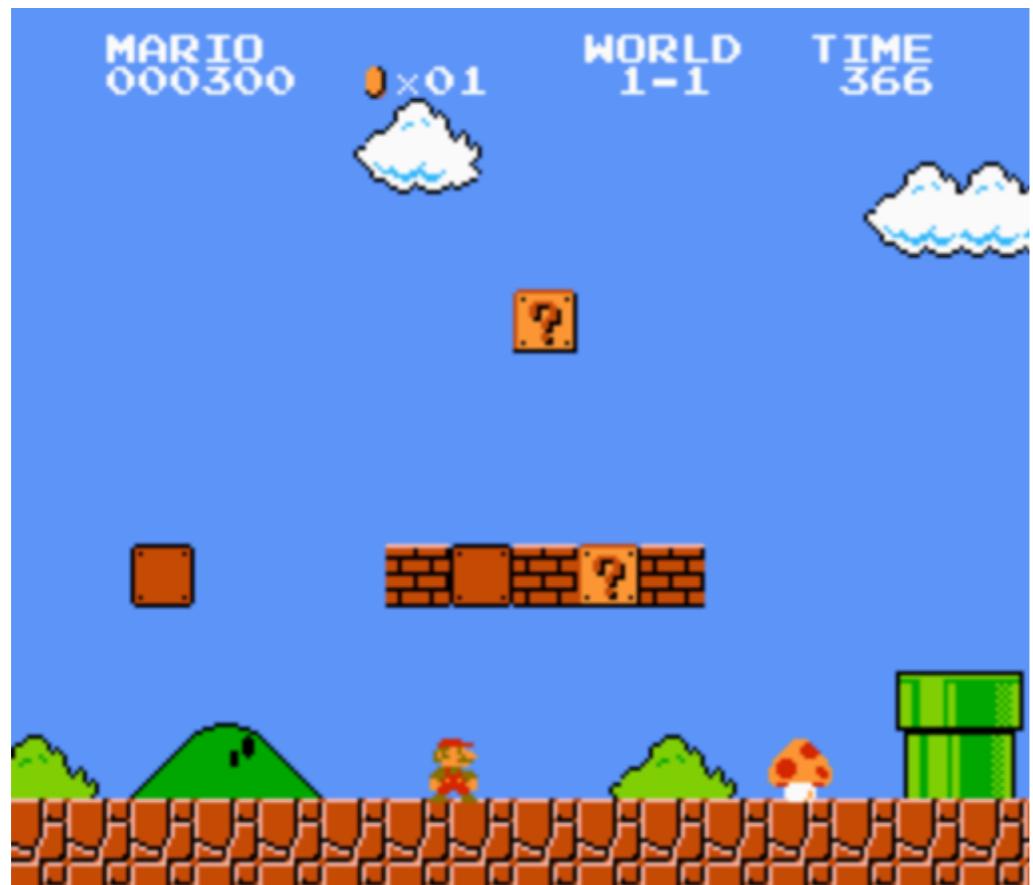
- Mapping state to action : *policy*
- Crucial problem in many robotics applications..

Introduction

- Mapping state to action : *policy*
- Crucial problem in many robotics applications..
..but hard to design by hand !

Introduction

- Mapping state to action : *policy*
- Crucial problem in many robotics applications..
..but hard to design by hand !



Super Mario Bros

Introduction

- Mapping state to action : *policy*
- Crucial problem in many robotics applications..
..but hard to design by hand !



Super Mario Bros



Nao (SoftBank Robotics)

■ REINFORCEMENT LEARNING⁽¹⁾

- Formulated for Markov Decision Process (MDP) :

$$(\mathcal{S}, \mathcal{A}, \mathcal{P}_{ss'}^a, \mathcal{R}_{ss'}^a)$$

with :

\mathcal{S}	state space
\mathcal{A}	action space
$\mathcal{P}_{ss'}^a = \mathbb{P}(s_{t+1} = s' s_t = s, a_t = a)$	Markovian dynamics
$\mathcal{R}_{ss'}^a = \mathbb{E}(r_t s_t = s, a_t = a, s_{t+1} = s')$	Markovian reward

⁽¹⁾ Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*

■ REINFORCEMENT LEARNING⁽¹⁾

- Formulated for Markov Decision Process (MDP) :

$$(\mathcal{S}, \mathcal{A}, \mathcal{P}_{ss'}^a, \mathcal{R}_{ss'}^a)$$

with :

\mathcal{S}	state space
\mathcal{A}	action space
$\mathcal{P}_{ss'}^a = \mathbb{P}(s_{t+1} = s' s_t = s, a_t = a)$	Markovian dynamics
$\mathcal{R}_{ss'}^a = \mathbb{E}(r_t s_t = s, a_t = a, s_{t+1} = s')$	Markovian reward

- Objective : find the policy

$$\begin{aligned}\pi : \quad \mathcal{S} &\rightarrow \mathcal{A}(s) \\ &s \rightarrow a\end{aligned}$$

that maximizes the accumulated reward

⁽¹⁾ Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*

■ REINFORCEMENT LEARNING

■ REINFORCEMENT LEARNING

- Model-based solving : **dynamic programming** (*value iteration algorithm*)

evaluate and improve the state-value function (static)

$$V_{\pi}(s) = \mathbb{E}_{\pi}(R_t = \sum_i r_{t+i+1} \mid s_t = s)$$

■ REINFORCEMENT LEARNING

- Model-based solving : **dynamic programming** (*value iteration algorithm*)

evaluate and improve the state-value function (static)

$$V_\pi(s) = \mathbb{E}_\pi(R_t = \sum_i r_{t+i+1} \mid s_t = s)$$

- Model-free solving : **temporal difference** (among others)

evaluate and improve the action-value function (try-out)

$$Q_\pi(s, a) = \mathbb{E}_\pi(R_t = \sum_i r_{t+i+1} \mid s_t = s, a_t = a)$$

■ REINFORCEMENT LEARNING

- Model-based solving : **dynamic programming** (*value iteration algorithm*)

evaluate and improve the state-value function (static)

$$V_\pi(s) = \mathbb{E}_\pi(R_t = \sum_i r_{t+i+1} \mid s_t = s)$$

- Model-free solving : **temporal difference** (among others)

evaluate and improve the action-value function (try-out)

$$Q_\pi(s, a) = \mathbb{E}_\pi(R_t = \sum_i r_{t+i+1} \mid s_t = s, a_t = a)$$

Bootstrap, explore and backup (tabular RL)!

$$Q_\pi(s, a) \sim r_t + Q_\pi(s', a')$$

■ REINFORCEMENT LEARNING

■ REINFORCEMENT LEARNING

$$Q_{\pi}(s, a) \sim r_t + Q_{\pi}(s', a')$$

Who is a' ?

■ REINFORCEMENT LEARNING

$$Q_{\pi}(s, a) \sim r_t + Q_{\pi}(s', a')$$

Who is a' ?

ON POLICY

OFF POLICY

■ REINFORCEMENT LEARNING

$$Q_{\pi}(s, a) \sim r_t + Q_{\pi}(s', a')$$

Who is a' ?

ON POLICY

OFF POLICY

Sampled from the
exploratory policy

$$p(a') = \frac{e^{Q(s', a')}}{\prod_{a \in \mathcal{A}(s)} e^{Q(s', a)}}$$

(SARSA)

■ REINFORCEMENT LEARNING

$$Q_{\pi}(s, a) \sim r_t + Q_{\pi}(s', a')$$

Who is a' ?

ON POLICY

OFF POLICY

Sampled from the
exploratory policy

Sampled from the
current greedy policy

$$p(a') = \frac{e^{Q(s', a')}}{\prod_{a \in \mathcal{A}(s)} e^{Q(s', a)}}$$

$$\text{argmax}_{a \in \mathcal{A}(s)} Q(s', a)$$

(SARSA)

(QLearning)

■ LEARNING FROM DEMONSTRATION^(1,2)

⁽¹⁾Aude G. Billard, Sylvain Calinon, and Rüdiger Dillmann. “Learning from Humans”

⁽²⁾Argall, Brenna D., et al. "A survey of robot learning from demonstration."

■ LEARNING FROM DEMONSTRATION^(1,2)

- Reinforcement learning is slow !

⁽¹⁾Aude G. Billard, Sylvain Calinon, and Rüdiger Dillmann. “Learning from Humans”

⁽²⁾Argall, Brenna D., et al. "A survey of robot learning from demonstration."

■ LEARNING FROM DEMONSTRATION^(1,2)

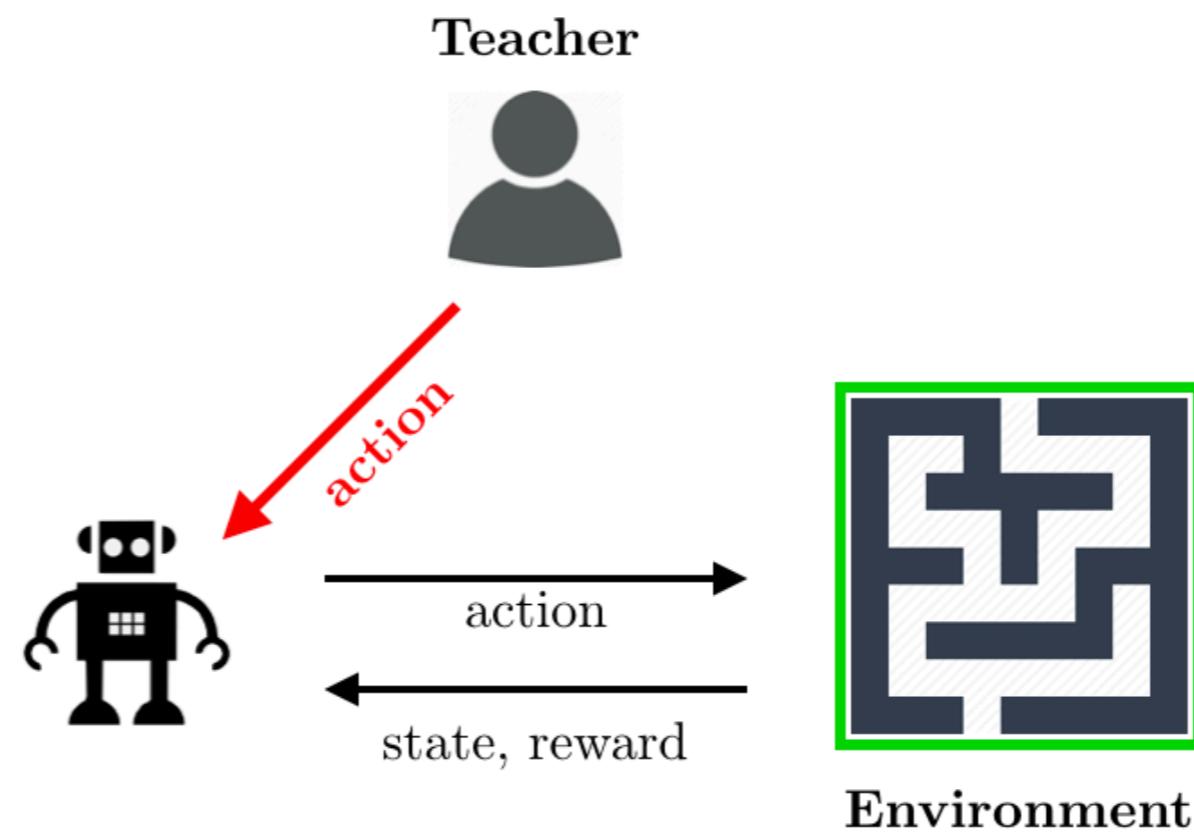
- Reinforcement learning is slow !
- Use a **teacher** :
 - Focus the exploration in important areas
 - Adds prior knowledge

⁽¹⁾Aude G. Billard, Sylvain Calinon, and Rüdiger Dillmann. “Learning from Humans”

⁽²⁾Argall, Brenna D., et al. "A survey of robot learning from demonstration."

■ LEARNING FROM DEMONSTRATION^(1,2)

- Reinforcement learning is slow !
- Use a **teacher** :
 - Focus the exploration in important areas
 - Adds prior knowledge



⁽¹⁾Aude G. Billard, Sylvain Calinon, and Rüdiger Dillmann. “Learning from Humans”

⁽²⁾Argall, Brenna D., et al. "A survey of robot learning from demonstration."

■ LEARNING FROM DEMONSTRATION

- How to use demonstrations in a reinforcement learning context ?

⁽¹⁾ S. Ross, G. Gordon and A. Bagnell. “*A reduction of imitation learning and structured prediction to no-regret online learning*”

⁽²⁾ B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, “*Maximum entropy inverse reinforcement learning*”

⁽³⁾ B. Piot, M. Geist, and O. Pietquin, “*Bridging the gap between imitation learning and inverse reinforcement learning*”

■ LEARNING FROM DEMONSTRATION

- How to use demonstrations in a reinforcement learning context ?
 - Bootstrap RL
 - Involve a teacher's policy in a policy mixture⁽¹⁾
 - .. or equivalently let the teacher take control along the learning

⁽¹⁾ S. Ross, G. Gordon and A. Bagnell. “A reduction of imitation learning and structured prediction to no-regret online learning”

⁽²⁾ B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning”

⁽³⁾ B. Piot, M. Geist, and O. Pietquin, “Bridging the gap between imitation learning and inverse reinforcement learning”

■ LEARNING FROM DEMONSTRATION

- How to use demonstrations in a reinforcement learning context ?
 - Bootstrap RL
 - Involve a teacher's policy in a policy mixture⁽¹⁾
 - .. or equivalently let the teacher take control along the learning
 - Inverse Reinforcement Learning : learn the reward function^(2,3)

⁽¹⁾ S. Ross, G. Gordon and A. Bagnell. “A reduction of imitation learning and structured prediction to no-regret online learning”

⁽²⁾ B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning”

⁽³⁾ B. Piot, M. Geist, and O. Pietquin, “Bridging the gap between imitation learning and inverse reinforcement learning”

■ LEARNING FROM DEMONSTRATION

- How to use demonstrations in a reinforcement learning context ?
 - Bootstrap RL
 - Involve a teacher's policy in a policy mixture⁽¹⁾
 - .. or equivalently let the teacher take control along the learning
 - Inverse Reinforcement Learning : learn the reward function^(2,3)
 - Reward similarity to the teacher

⁽¹⁾ S. Ross, G. Gordon and A. Bagnell. “A reduction of imitation learning and structured prediction to no-regret online learning”

⁽²⁾ B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning”

⁽³⁾ B. Piot, M. Geist, and O. Pietquin, “Bridging the gap between imitation learning and inverse reinforcement learning”

■ LEARNING FROM DEMONSTRATION

- How to use demonstrations in a reinforcement learning context ?
 - Bootstrap RL
 - Involve a teacher's policy in a policy mixture⁽¹⁾
 - .. or equivalently let the teacher take control along the learning
 - Inverse Reinforcement Learning : learn the reward function^(2,3)
 - Reward similarity to the teacher
 - ..

⁽¹⁾ S. Ross, G. Gordon and A. Bagnell. “A reduction of imitation learning and structured prediction to no-regret online learning”

⁽²⁾ B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning”

⁽³⁾ B. Piot, M. Geist, and O. Pietquin, “Bridging the gap between imitation learning and inverse reinforcement learning”

■ LEARNING FROM DEMONSTRATION

■ LEARNING FROM DEMONSTRATION

- Suppose a *near-optimal* teachers
.. or at least *good* demonstration in the dataset

■ LEARNING FROM DEMONSTRATION

- Suppose a *near-optimal* teachers
 - .. or at least *good* demonstration in the dataset
- What does it mean to be suboptimal ? *Largely* suboptimal ?

■ LEARNING FROM DEMONSTRATION

- Suppose a *near-optimal* teachers
 - .. or at least *good* demonstration in the dataset
- What does it mean to be suboptimal ? *Largely* suboptimal ?
 - Poor transfer to the robot abilities

■ LEARNING FROM DEMONSTRATION

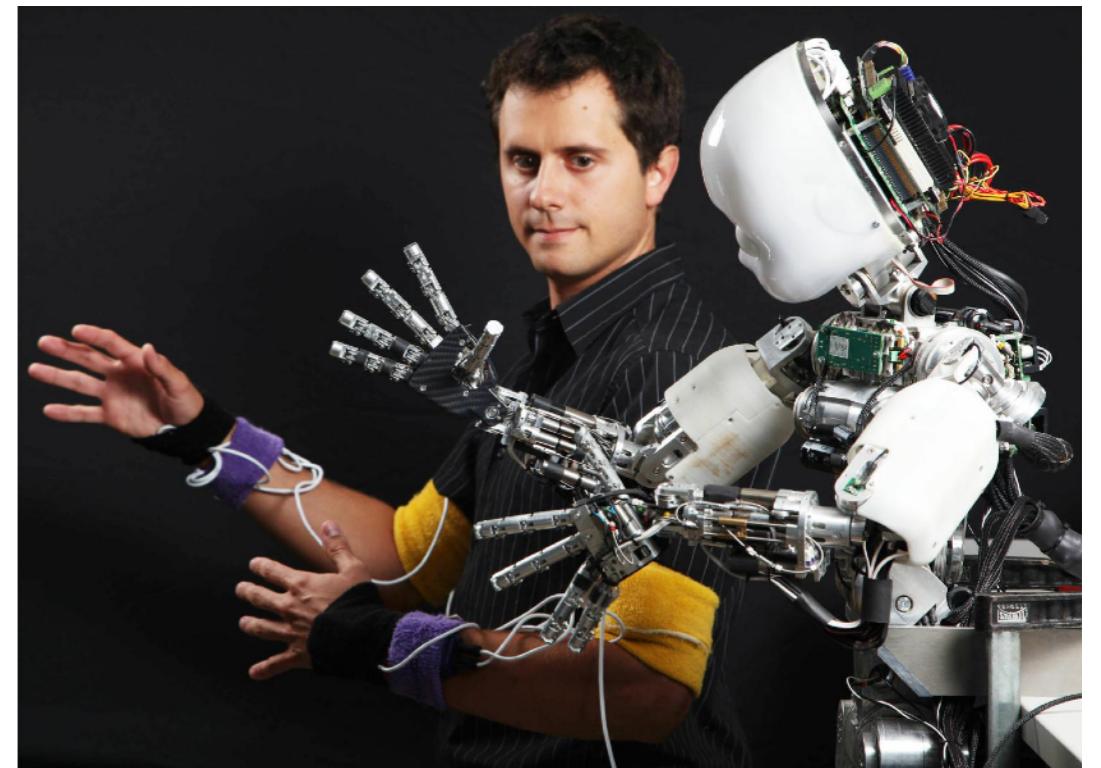
- Suppose a *near-optimal* teachers
 - .. or at least *good* demonstration in the dataset
- What does it mean to be suboptimal ? *Largely* suboptimal ?
 - Poor transfer to the robot abilities
 - Does not maximize a numerical criterion evaluating the fitness of a behavior

■ LEARNING FROM DEMONSTRATION

- Suppose a *near-optimal* teachers
 - .. or at least *good* demonstration in the dataset
- What does it mean to be suboptimal ? *Largely* suboptimal ?
 - Poor transfer to the robot abilities
 - Does not maximize a numerical criterion evaluating the fitness of a behavior
 - Obvious downside - possible danger !

■ LEARNING FROM DEMONSTRATION

- Suppose a *near-optimal* teachers
 - .. or at least *good* demonstration in the dataset
- What does it mean to be suboptimal ? *Largely* suboptimal ?
 - Poor transfer to the robot abilities
 - Does not maximize a numerical criterion evaluating the fitness of a behavior
 - Obvious downside - possible danger !
 - Ex : Grasping objects



Credits : Sylvain Calinon

Motivations

- Find a way to learn from suboptimal teachers

⁽¹⁾J. Choi and K.-E. Kim, “*Hierarchical bayesian inverse reinforcement learning*”

⁽²⁾V. Chu and A. L. Thomaz, “*Analyzing differences between teachers when learning object affordances via guided exploration*”

Motivations

- Find a way to learn from suboptimal teachers
 - Extract useful information from any demonstration
 - Extend to non-expert teachers !
 - Enlarge human-robot interaction possibilities

⁽¹⁾J. Choi and K.-E. Kim, “*Hierarchical bayesian inverse reinforcement learning*”

⁽²⁾V. Chu and A. L. Thomaz, “*Analyzing differences between teachers when learning object affordances via guided exploration*”

Motivations

- Find a way to learn from suboptimal teachers
 - Extract useful information from any demonstration
 - Extend to non-expert teachers !
 - Enlarge human-robot interaction possibilities
- Some approaches in the literature :

⁽¹⁾J. Choi and K.-E. Kim, “*Hierarchical bayesian inverse reinforcement learning*”

⁽²⁾V. Chu and A. L. Thomaz, “*Analyzing differences between teachers when learning object affordances via guided exploration*”

Motivations

- Find a way to learn from suboptimal teachers
 - Extract useful information from any demonstration
 - Extend to non-expert teachers !
 - Enlarge human-robot interaction possibilities
- Some approaches in the literature :
 - Bayesian Inverse Reinforcement Learning⁽¹⁾

⁽¹⁾J. Choi and K.-E. Kim, “*Hierarchical bayesian inverse reinforcement learning*”

⁽²⁾V. Chu and A. L. Thomaz, “*Analyzing differences between teachers when learning object affordances via guided exploration*”

Motivations

- Find a way to learn from suboptimal teachers
 - Extract useful information from any demonstration
 - Extend to non-expert teachers !
 - Enlarge human-robot interaction possibilities
- Some approaches in the literature :
 - Bayesian Inverse Reinforcement Learning⁽¹⁾
 - Guided exploration⁽²⁾

⁽¹⁾J. Choi and K.-E. Kim, “*Hierarchical bayesian inverse reinforcement learning*”

⁽²⁾V. Chu and A. L. Thomaz, “*Analyzing differences between teachers when learning object affordances via guided exploration*”

Motivations

- Find a way to learn from suboptimal teachers
 - Extract useful information from any demonstration
 - Extend to non-expert teachers !
 - Enlarge human-robot interaction possibilities
- Some approaches in the literature :
 - Bayesian Inverse Reinforcement Learning⁽¹⁾
 - Guided exploration⁽²⁾
 - ..

⁽¹⁾J. Choi and K.-E. Kim, “*Hierarchical bayesian inverse reinforcement learning*”

⁽²⁾V. Chu and A. L. Thomaz, “*Analyzing differences between teachers when learning object affordances via guided exploration*”

Motivations

- A (universal) learning from demonstration approach for humans :

Motivations

- A (universal) learning from demonstration approach for humans :
 - ▶ Try to reproduce the teacher's moves as closely as possible

Motivations

- A (universal) learning from demonstration approach for humans :
 - ▶ Try to reproduce the teacher's moves as closely as possible
 - ▶ Gather some intuition and knowledge about the task

Motivations

- A (universal) learning from demonstration approach for humans :
 - ▶ Try to reproduce the teacher's moves as closely as possible
 - ▶ Gather some intuition and knowledge about the task
 - ▶ Evaluate the teacher's actions under that acquired knowledge

Motivations

- A (universal) learning from demonstration approach for humans :
 - ▶ Try to reproduce the teacher's moves as closely as possible
 - ▶ Gather some intuition and knowledge about the task
 - ▶ Evaluate the teacher's actions under that acquired knowledge
 - ▶ Follow own decisions

Motivations

- A (universal) learning from demonstration approach for humans :
 - ▶ Try to reproduce the teacher's moves as closely as possible
 - ▶ Gather some intuition and knowledge about the task
 - ▶ Evaluate the teacher's actions under that acquired knowledge
 - ▶ Follow own decisions
- ➔ Shifting compliance (with respect to the teacher)

Motivations

- A (universal) learning from demonstration approach for humans :
 - ▶ Try to reproduce the teacher's moves as closely as possible
 - ▶ Gather some intuition and knowledge about the task
 - ▶ Evaluate the teacher's actions under that acquired knowledge
 - ▶ Follow own decisions
- ➔ Shifting compliance (with respect to the teacher)

GOAL : - define a compliance-based approach for learning from suboptimal teachers
- experimentally evaluate its performances

■ INTUITION

- Bias the action-selection (exploratory policy) towards the teacher's demonstrations

■ INTUITION

- Bias the action-selection (exploratory policy) towards the teacher's demonstrations
- Define :
 - $a_m(s)$ the mentor's action at state s
 - $p(s)$ the **compliance** at state s

$$\forall s \in \mathcal{S}, \quad \pi_p(s) = \begin{cases} a_m(s) \text{ with probability } p(s) \\ a \in \mathcal{A}(s) \text{ with probability } (1 - p(s)) \end{cases}$$

■ INTUITION

- Bias the action-selection (exploratory policy) towards the teacher's demonstrations
- Define :
 - $a_m(s)$ the mentor's action at state s
 - $p(s)$ the **compliance** at state s

$$\forall s \in \mathcal{S}, \quad \pi_p(s) = \begin{cases} a_m(s) \text{ with probability } p(s) \\ a \in \mathcal{A}(s) \text{ with probability } (1 - p(s)) \end{cases}$$

- Make the compliance **adapt** throughout the learning

■ VANISHING-COMPLIANCE LEARNER (Method 1)

■ VANISHING-COMPLIANCE LEARNER (Method 1)

- The compliance is initialized near 1 and slowly vanishes :

$$p_{k+1} = \gamma p_k, \quad \gamma < 1$$

■ VANISHING-COMPLIANCE LEARNER (Method 1)

- The compliance is initialized near 1 and slowly vanishes :

$$p_{k+1} = \gamma p_k, \quad \gamma < 1$$

- Extremely simple and easy to implement !

■ VANISHING-COMPLIANCE LEARNER (Method 1)

- The compliance is initialized near 1 and slowly vanishes :

$$p_{k+1} = \gamma p_k, \quad \gamma < 1$$

- Extremely simple and easy to implement !
- .. but rather naive

■ VANISHING-COMPLIANCE LEARNER (Method 1)

- The compliance is initialized near 1 and slowly vanishes :

$$p_{k+1} = \gamma p_k, \quad \gamma < 1$$

- Extremely simple and easy to implement !
- .. but rather naive
- .. and no teacher evaluation !

■ VANISHING-COMPLIANCE LEARNER (Method 1)

- The compliance is initialized near 1 and slowly vanishes :

$$p_{k+1} = \gamma p_k, \quad \gamma < 1$$

- Extremely simple and easy to implement !
- .. but rather naive
- .. and no teacher evaluation !
- also, tuning might be difficult

- Evaluate a teacher's recommendations and shift the bias accordingly

- Evaluate a teacher's recommandations and shift the bias accordingly

■ β -IMPLICIT COMPLIANCE LEARNER (Method 2)

- Evaluate a teacher's recommandations and shift the bias accordingly

■ β -IMPLICIT COMPLIANCE LEARNER (Method 2)

- Provide a Beta prior distribution for the compliance (initial bias)

$$\forall s \in \mathcal{S}, \quad p(s) \sim \beta(\alpha(s), \beta(s))$$

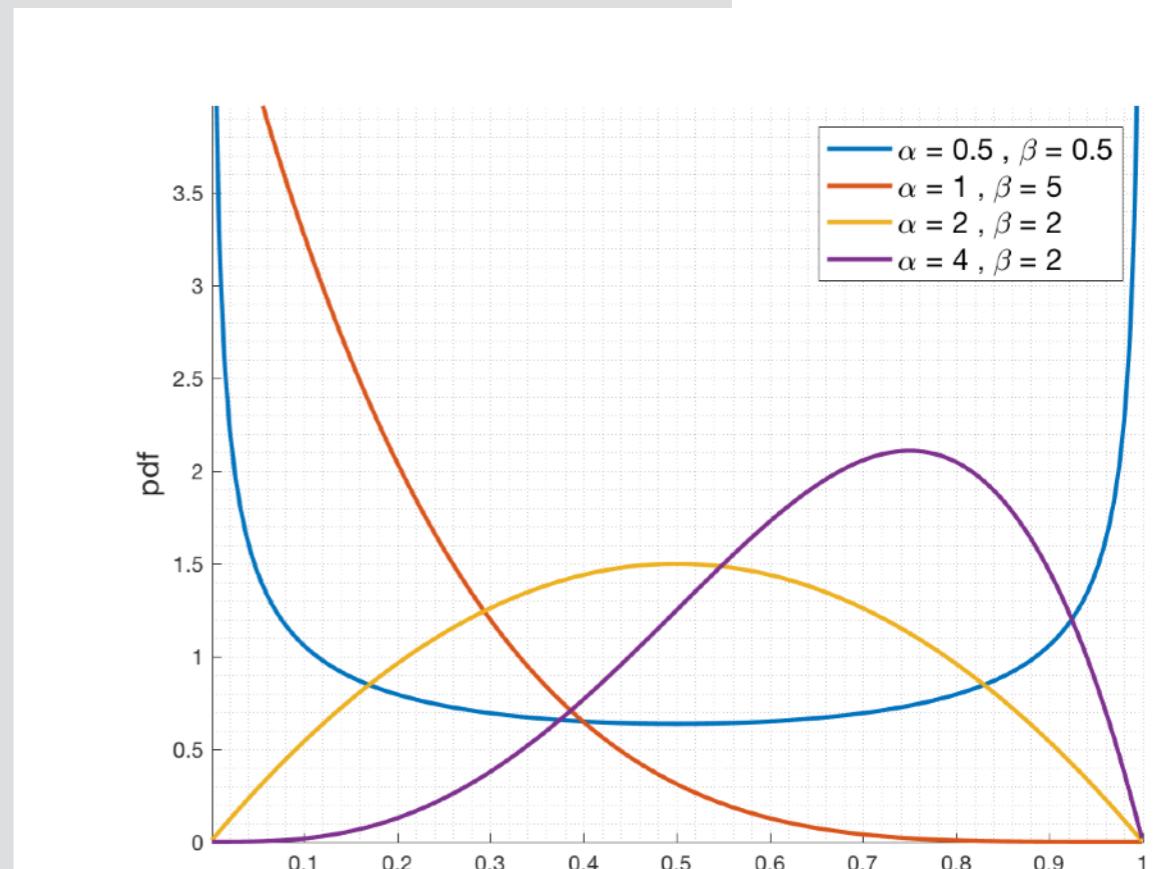


Figure : Beta p.d.f

- Evaluate a teacher's recommandations and shift the bias accordingly

■ β -IMPLICIT COMPLIANCE LEARNER (Method 2)

- Provide a Beta prior distribution for the compliance (initial bias)

$$\forall s \in \mathcal{S}, \quad p(s) \sim \beta(\alpha(s), \beta(s))$$

- Sample in the current policy (SARSA)

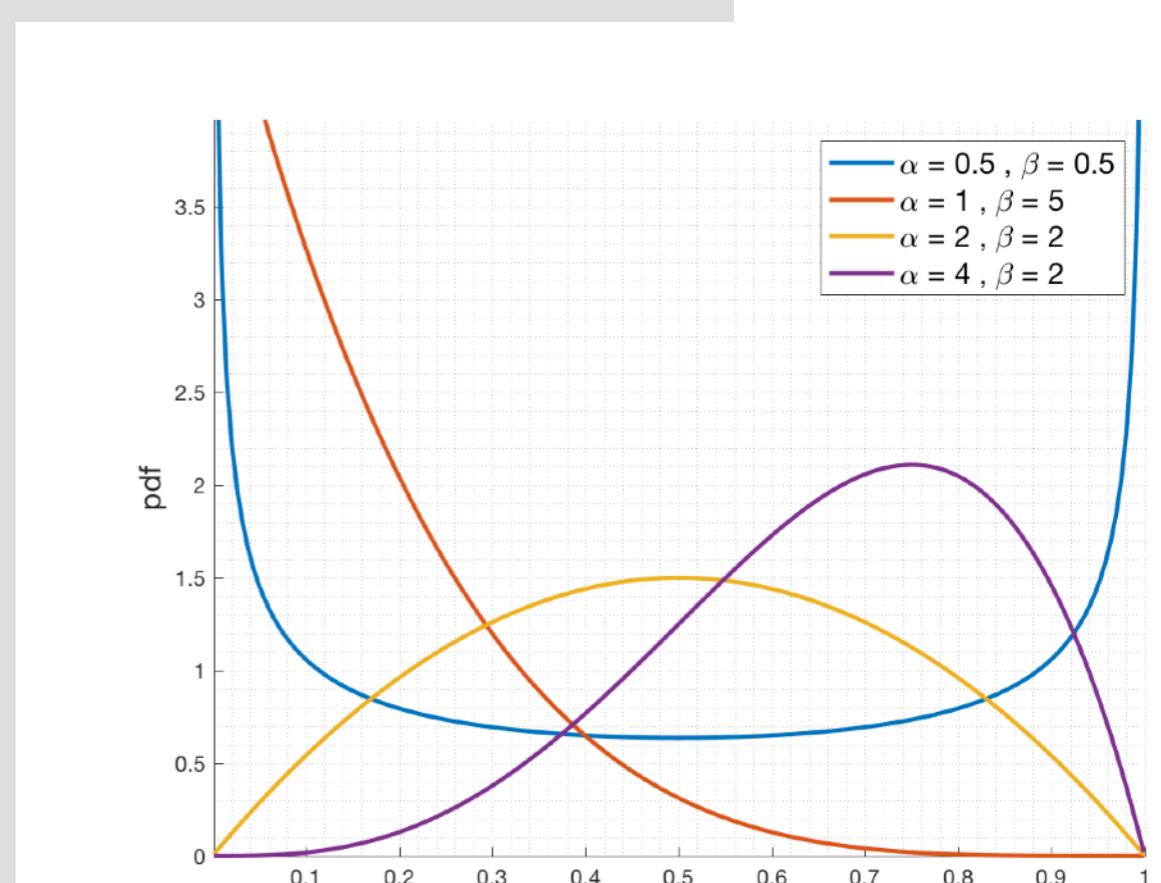


Figure : Beta p.d.f

- Evaluate a teacher's recommandations and shift the bias accordingly

■ β -IMPLICIT COMPLIANCE LEARNER (Method 2)

- Provide a Beta prior distribution for the compliance (initial bias)

$$\forall s \in \mathcal{S}, \quad p(s) \sim \beta(\alpha(s), \beta(s))$$

- Sample in the current policy (SARSA)
- Compute a TD(0) critic

$$\delta_t = r + \gamma Q(s', a') - Q(s, a_m)$$

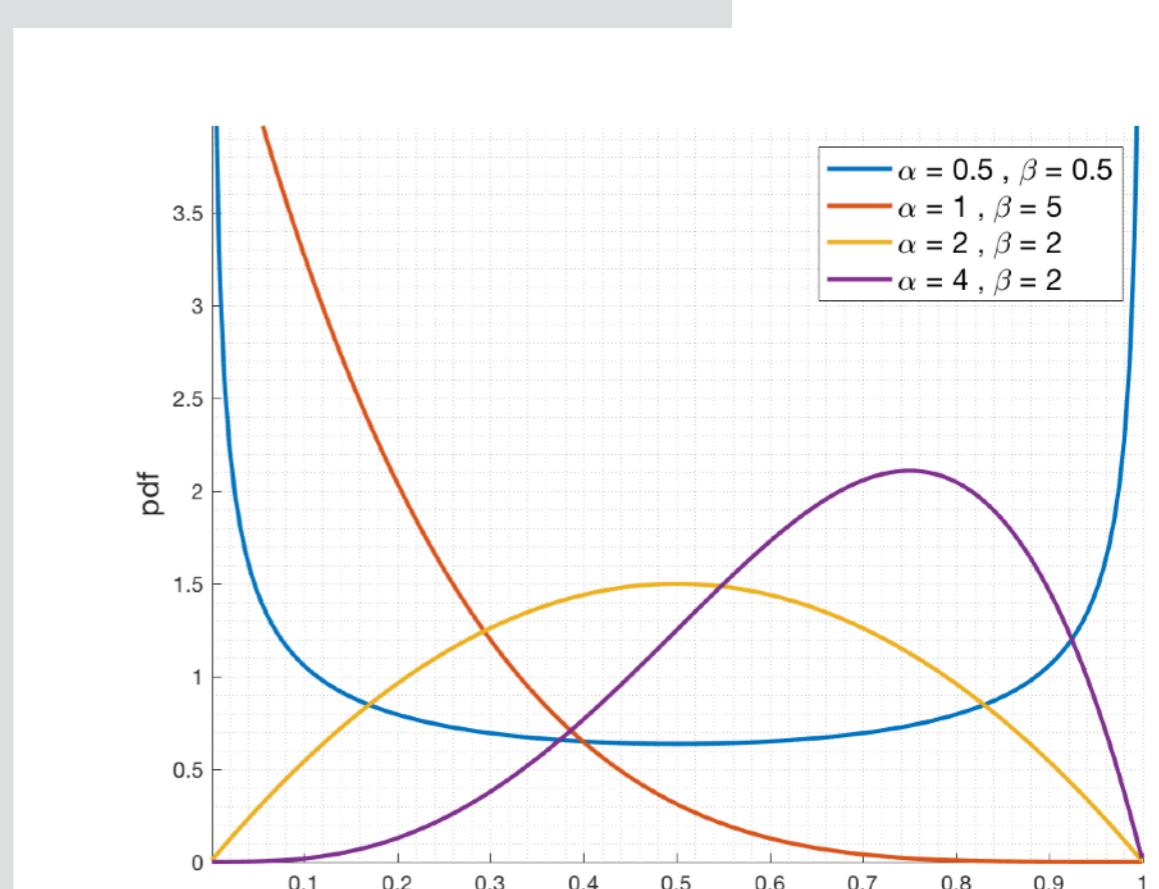


Figure : Beta p.d.f

- Evaluate a teacher's recommandations and shift the bias accordingly

■ β -IMPLICIT COMPLIANCE LEARNER (Method 2)

- Provide a Beta prior distribution for the compliance (initial bias)

$$\forall s \in \mathcal{S}, \quad p(s) \sim \beta(\alpha(s), \beta(s))$$

- Sample in the current policy (SARSA)
- Compute a TD(0) critic
- $$\delta_t = r + \gamma Q(s', a') - Q(s, a_m)$$
- Update the p.d.f parameters accordingly

$$\begin{aligned} \alpha_t(s) &\leftarrow \alpha_t(s) + \mathbb{1}_{a=a_m} \delta_t \varepsilon_t \\ \beta_t(s) &\leftarrow \beta_t(s) + \mathbb{1}_{a \neq a_m} \delta_t \varepsilon_t \end{aligned}$$

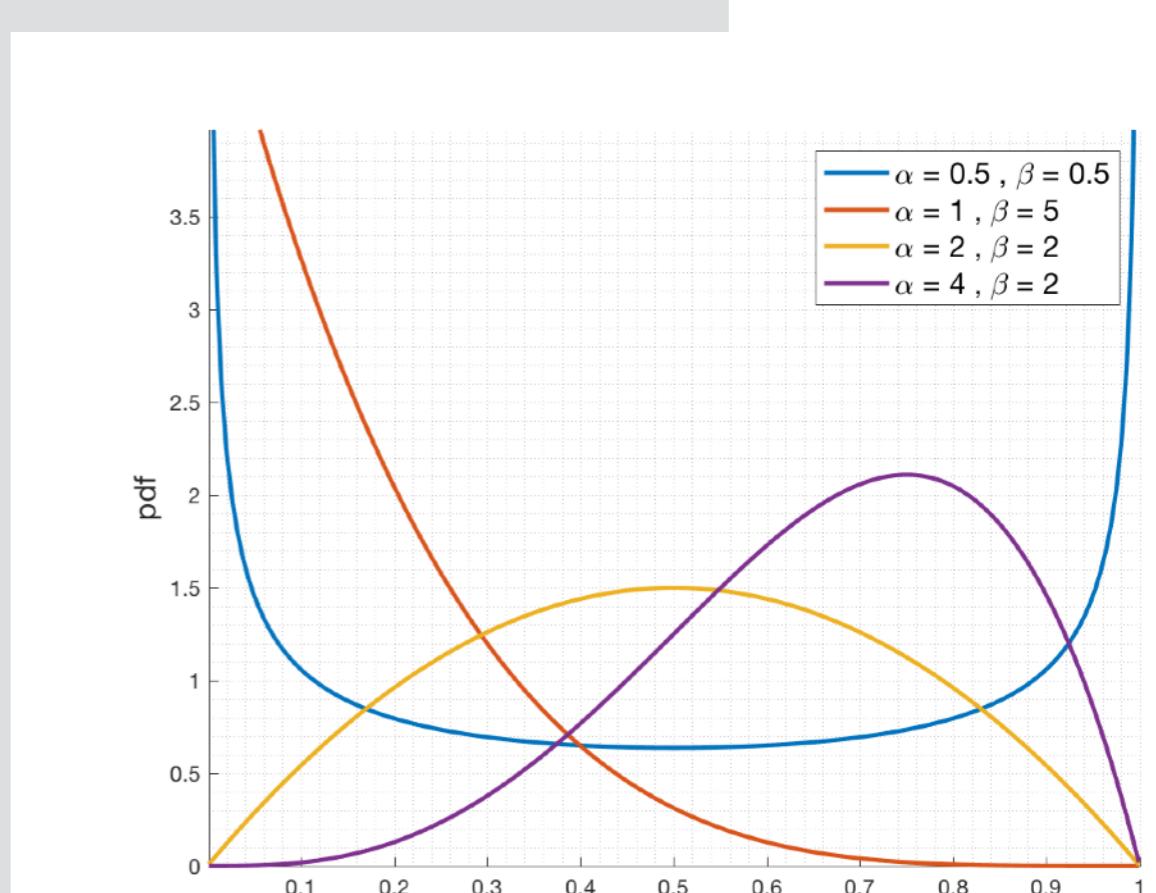


Figure : Beta p.d.f

- Evaluate a teacher's recommandations and shift the bias accordingly

■ β -IMPLICIT COMPLIANCE LEARNER (Method 2)

- Provide a Beta prior distribution for the compliance (initial bias)

$$\forall s \in \mathcal{S}, \quad p(s) \sim \beta(\alpha(s), \beta(s))$$

- Sample in the current policy (SARSA)

- Compute a TD(0) critic

$$\delta_t = r + \gamma Q(s', a') - Q(s, a_m)$$

- Update the p.d.f parameters accordingly

$$\alpha_t(s) \leftarrow \alpha_t(s) + \mathbb{1}_{a=a_m} \delta_t \varepsilon_t$$

$$\beta_t(s) \leftarrow \beta_t(s) + \mathbb{1}_{a \neq a_m} \delta_t \varepsilon_t$$

- Update Q-values

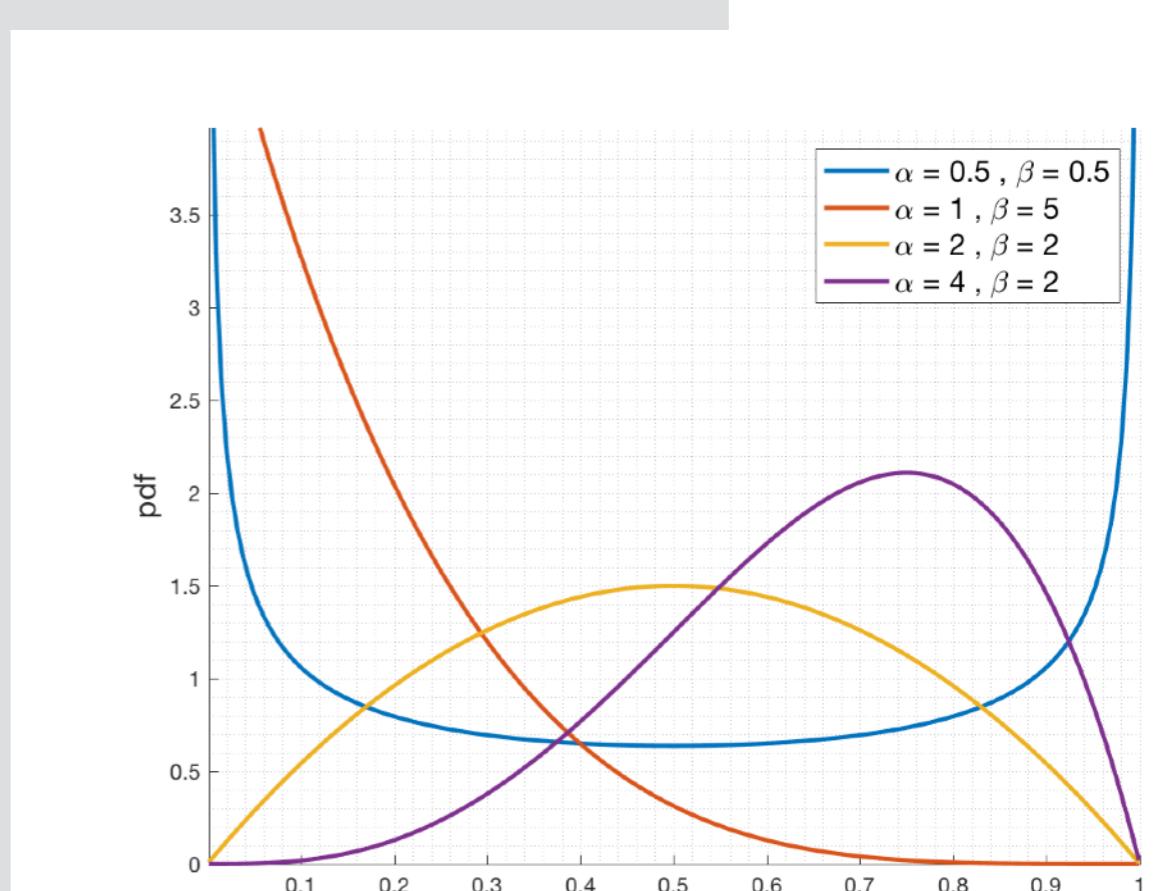


Figure : Beta p.d.f

■ EXPLICIT COMPLIANCE LEARNER (Method 3)

■ EXPLICIT COMPLIANCE LEARNER (Method 3)

- Learn the action-values of the MDP

$$\mathcal{A}_c(s) = \{'listen', 'discard'\}$$

■ EXPLICIT COMPLIANCE LEARNER (Method 3)

- Learn the action-values of the MDP

$$\mathcal{A}_c(s) = \{'listen', 'discard'\}$$

- Gibbs sampling in this MDP drives the exploration of the initial MDP !

■ EXPLICIT COMPLIANCE LEARNER (Method 3)

- Learn the action-values of the MDP

$$\mathcal{A}_c(s) = \{'listen', 'discard'\}$$

- Gibbs sampling in this MDP drives the exploration of the initial MDP !
- Procedure :

■ EXPLICIT COMPLIANCE LEARNER (Method 3)

- Learn the action-values of the MDP

$$\mathcal{A}_c(s) = \{'listen', 'discard'\}$$

- Gibbs sampling in this MDP drives the exploration of the initial MDP !

Initialize (introduce bias):

$$\{Q_c(s, l), Q_c(s, d)\}$$

- Procedure :

■ EXPLICIT COMPLIANCE LEARNER (Method 3)

- Learn the action-values of the MDP

$$\mathcal{A}_c(s) = \{'listen', 'discard'\}$$

- Gibbs sampling in this MDP drives the exploration of the initial MDP !

Initialize (introduce bias):

$$\{Q_c(s, l), Q_c(s, d)\} \leftarrow \text{listen and discard Q-values}$$

- Procedure :

■ EXPLICIT COMPLIANCE LEARNER (Method 3)

- Learn the action-values of the MDP

$$\mathcal{A}_c(s) = \{'listen', 'discard'\}$$

- Gibbs sampling in this MDP drives the exploration of the initial MDP !

Initialize (introduce bias):

$$\{Q_c(s, l), Q_c(s, d)\} \leftarrow \text{listen and discard}$$

Q-values

Sample from it (SARSA)

- Procedure :

■ EXPLICIT COMPLIANCE LEARNER (Method 3)

- Learn the action-values of the MDP

$$\mathcal{A}_c(s) = \{'listen', 'discard'\}$$

- Gibbs sampling in this MDP drives the exploration of the initial MDP !

Initialize (introduce bias):

$$\{Q_c(s, l), Q_c(s, d)\} \leftarrow \text{listen and discard}$$

Q-values

Sample from it (SARSA)

- Procedure :

Update initial MDP

■ EXPLICIT COMPLIANCE LEARNER (Method 3)

- Learn the action-values of the MDP

$$\mathcal{A}_c(s) = \{'listen', 'discard'\}$$

- Gibbs sampling in this MDP drives the exploration of the initial MDP !

Initialize (introduce bias):

$$\{Q_c(s, l), Q_c(s, d)\} \leftarrow \text{listen and discard}$$

Q-values

Sample from it (SARSA)

- Procedure :

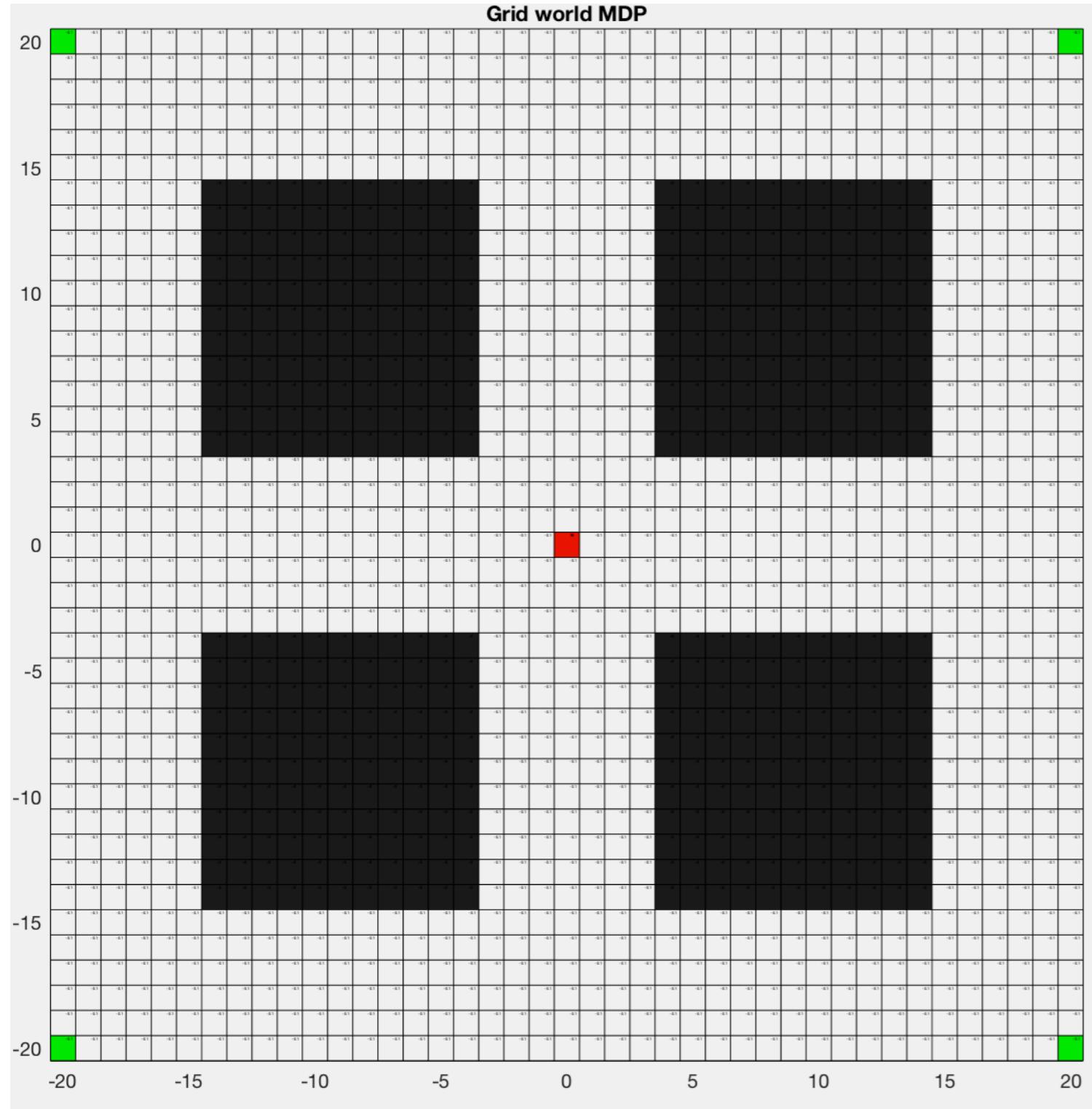
Update initial MDP

Update new MDP

$$\begin{cases} Q_c(s, l) \leftarrow \beta Q_c(s, l) + (1 - \beta) Q(s, a_m) \\ Q_c(s, d) = \beta Q_c(s, d) + (1 - \beta) \max_{a \neq a_m} Q(s, a) \end{cases}$$

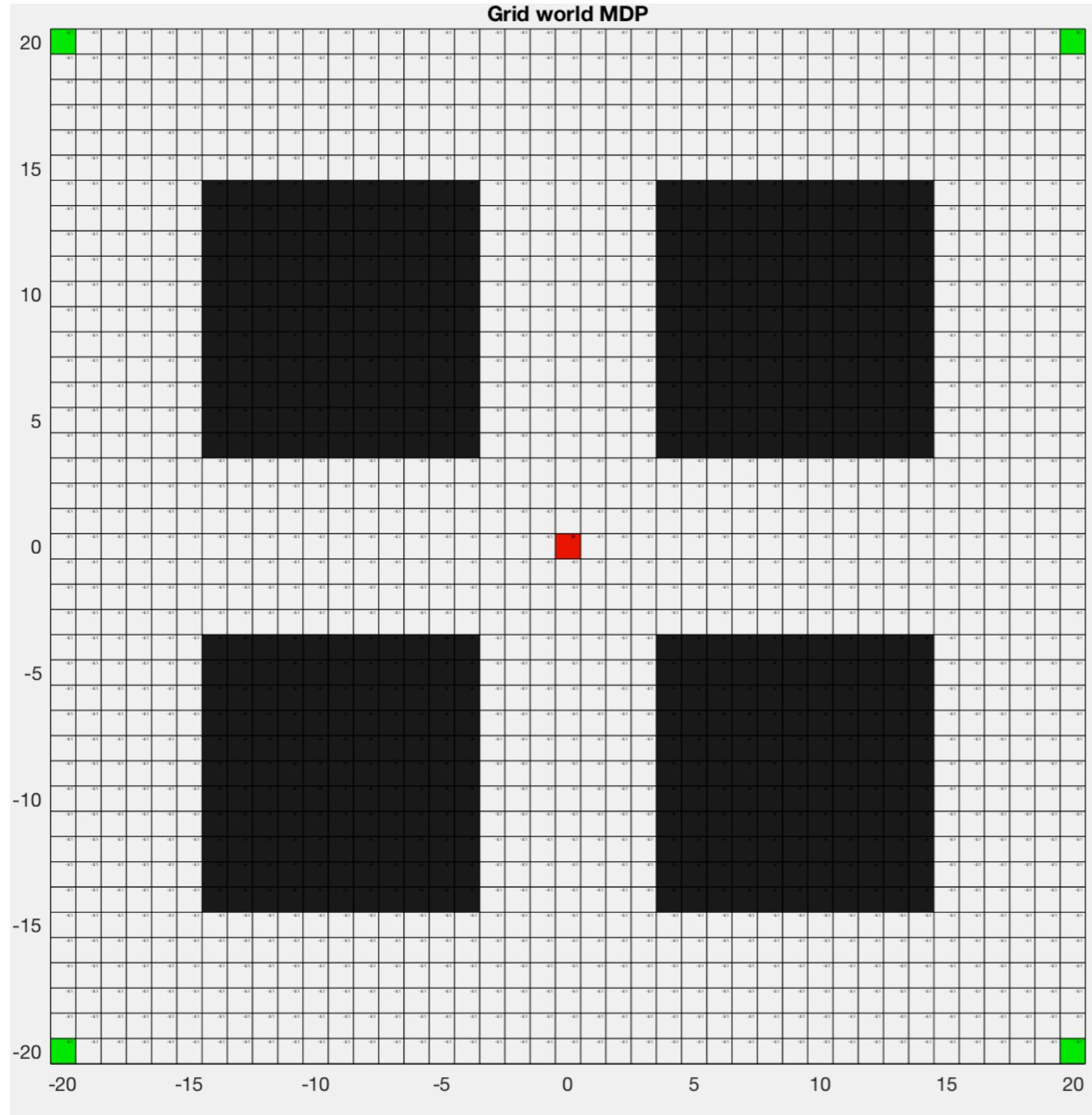
Results

■ MDP



Results

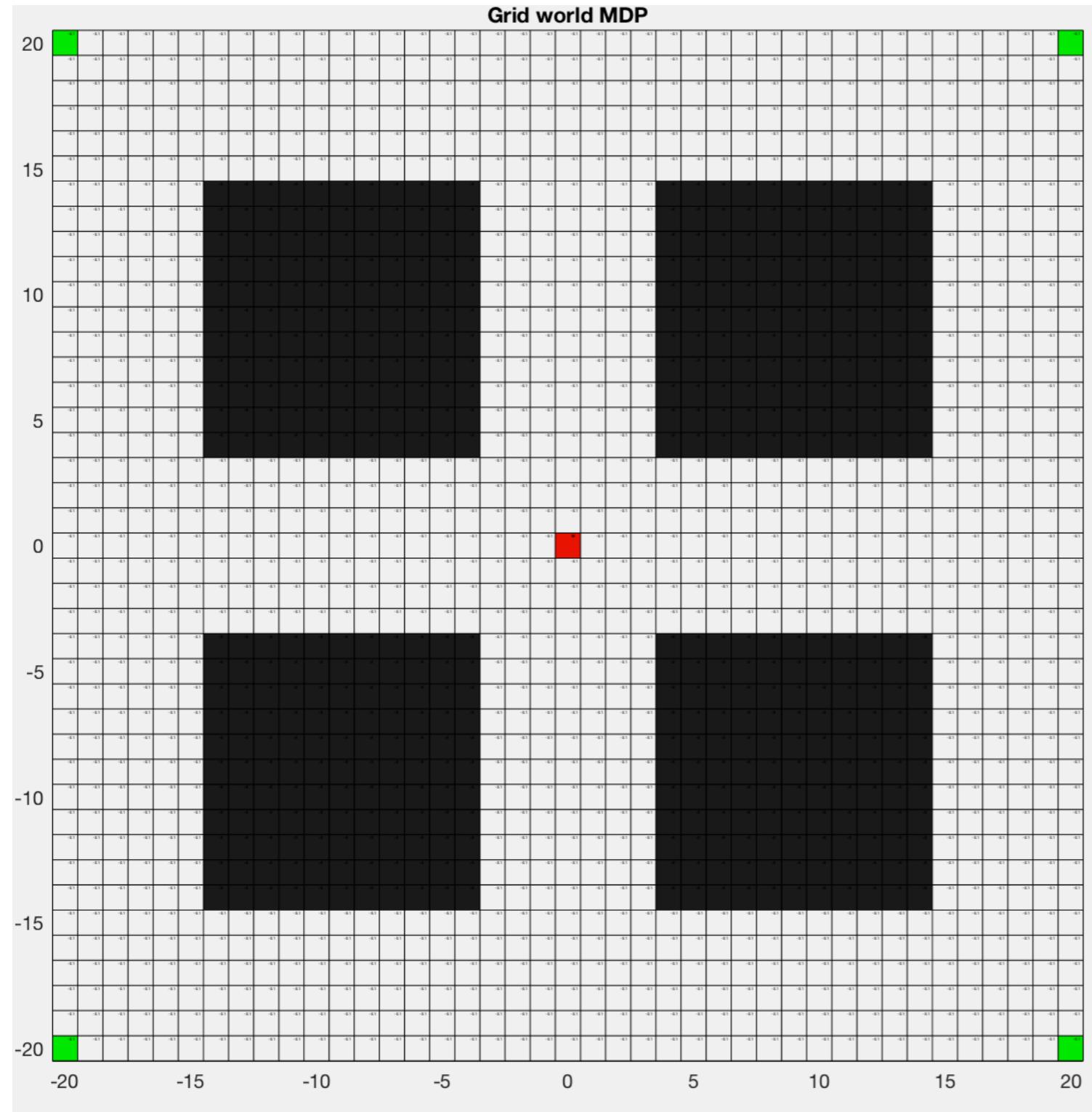
■ MDP



● Finite MDP

Results

■ MDP

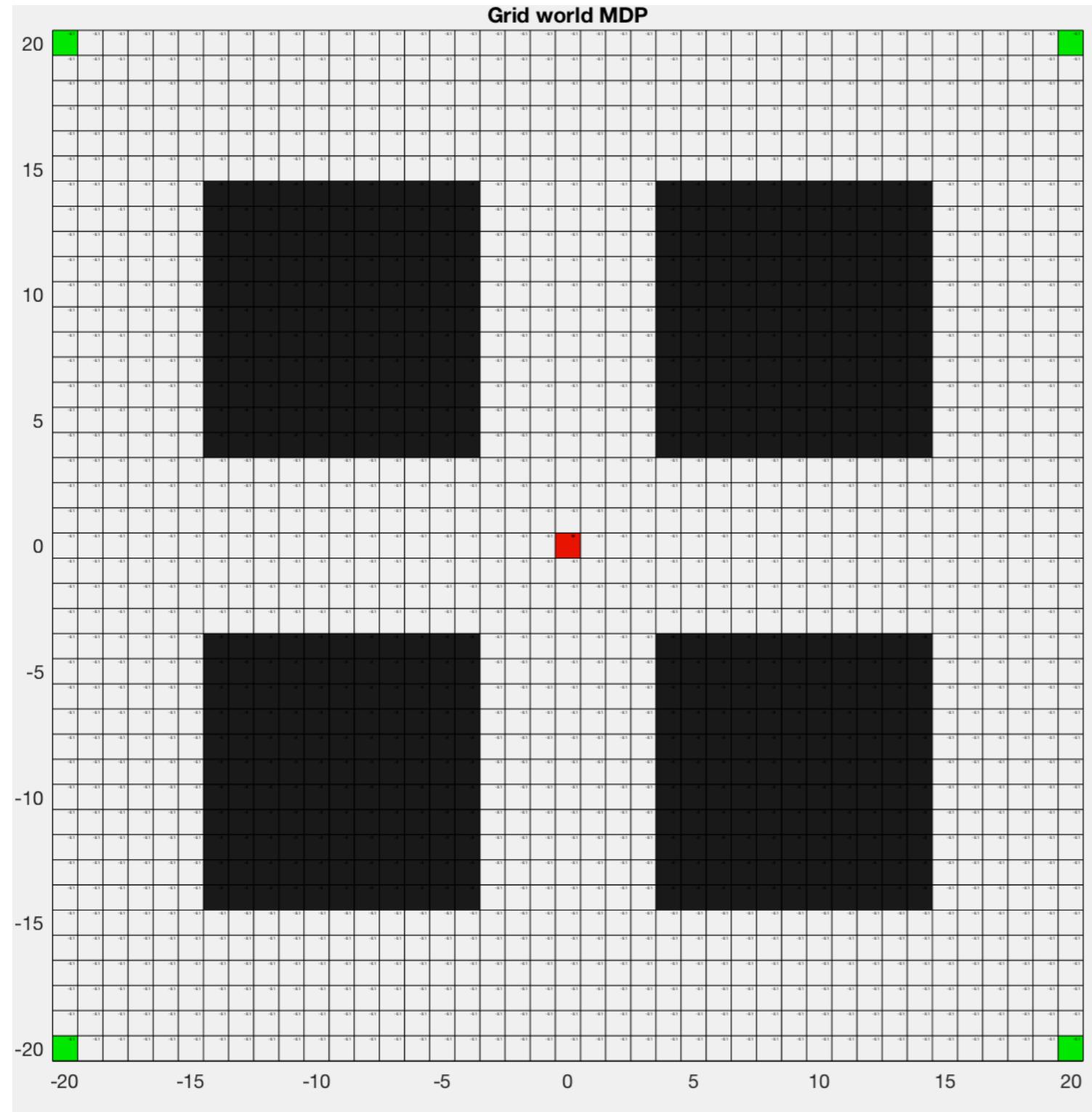


- Finite MDP

- From green cell to red cell as quickly as possible

Results

■ MDP



- Finite MDP

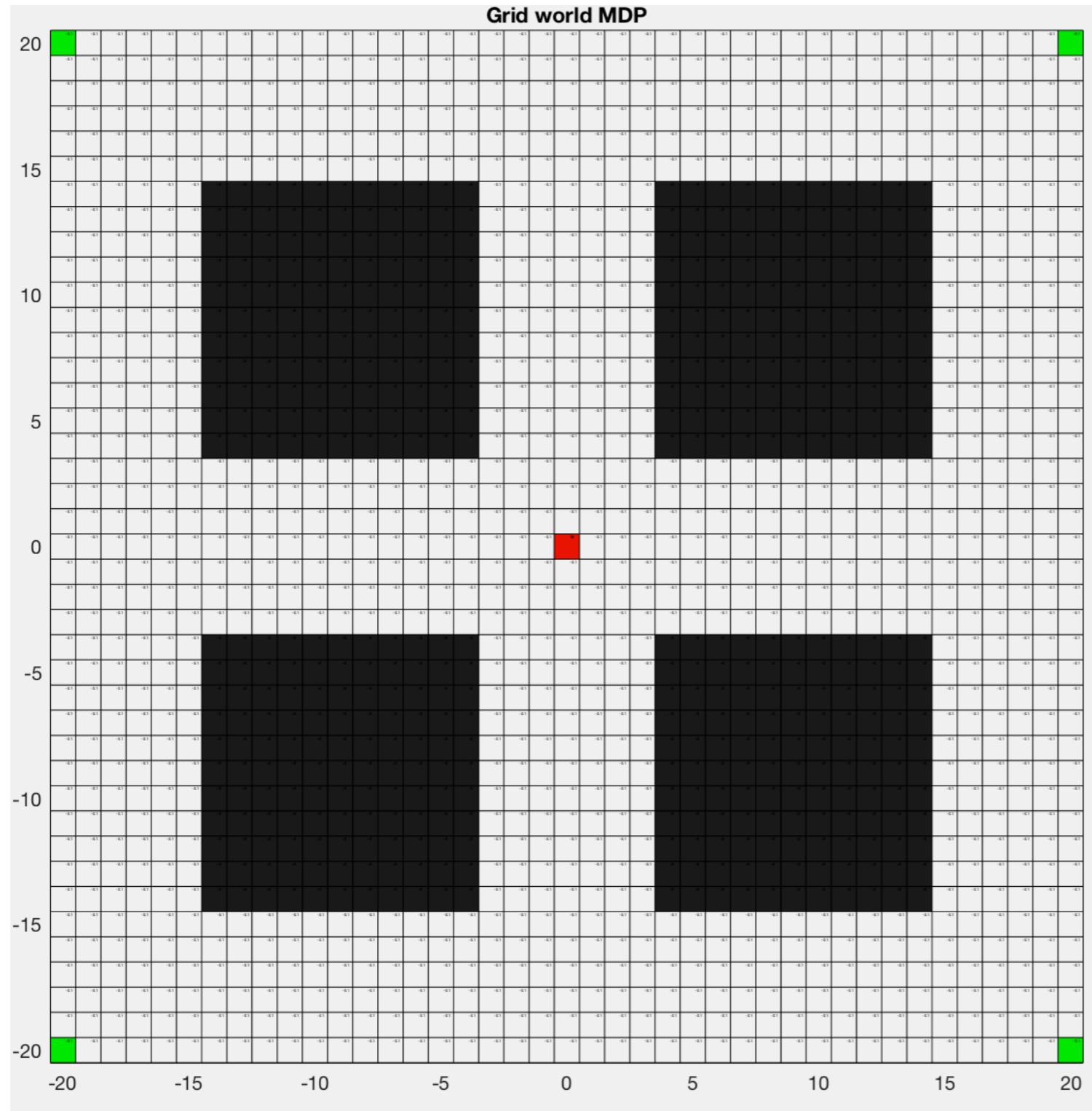
- From green cell to red cell as quickly as possible

- Stochastic :

$$\mathcal{P}_{s,s'}^a = \begin{cases} 0.9 & \text{if } s' = a(s) \\ 0.1 & \text{otherwise} \end{cases}$$

Results

■ MDP



- Finite MDP

- From green cell to red cell as quickly as possible

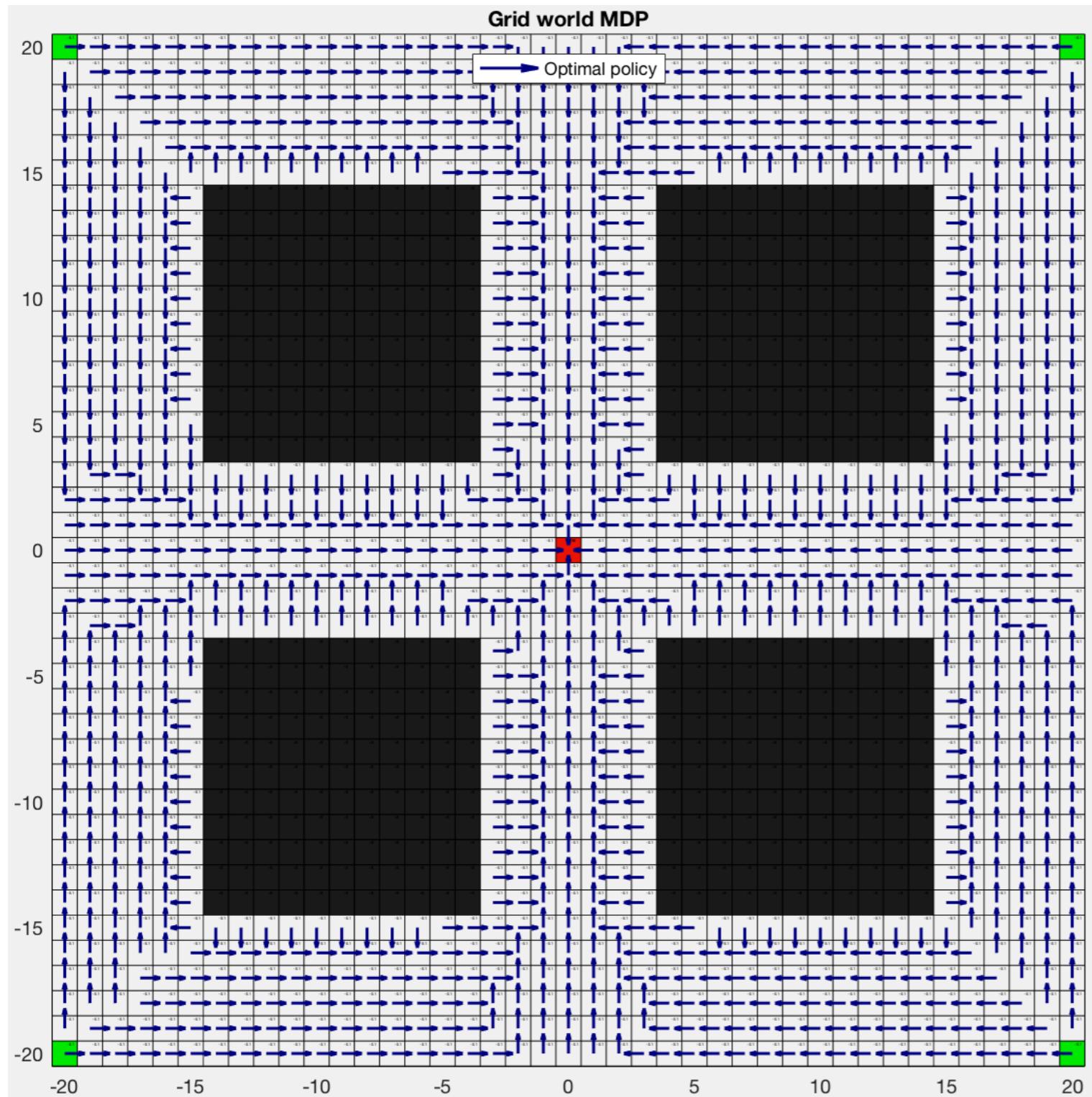
- Stochastic :

$$\mathcal{P}_{s,s'}^a = \begin{cases} 0.9 & \text{if } s' = a(s) \\ 0.1 & \text{otherwise} \end{cases}$$

- *Value iteration* for computing optimal policy (ground truth)

Results

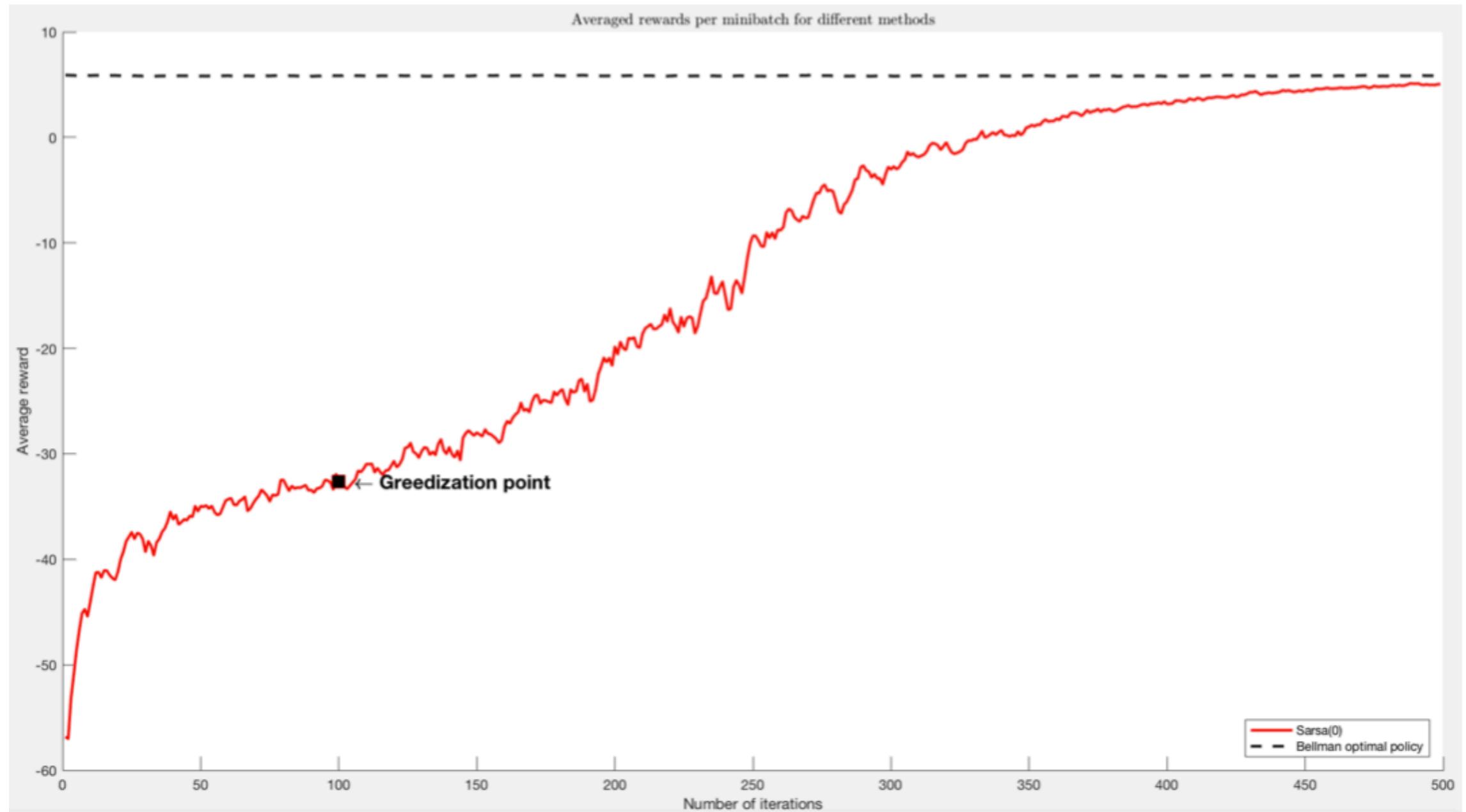
■ MDP



- Finite MDP
- From green cell to red cell as quickly as possible
- Stochastic :
$$\mathcal{P}_{s,s'}^a = \begin{cases} 0.9 & \text{if } s' = a(s) \\ 0.1 & \text{otherwise} \end{cases}$$
- *Value iteration* for computing optimal policy (ground truth)

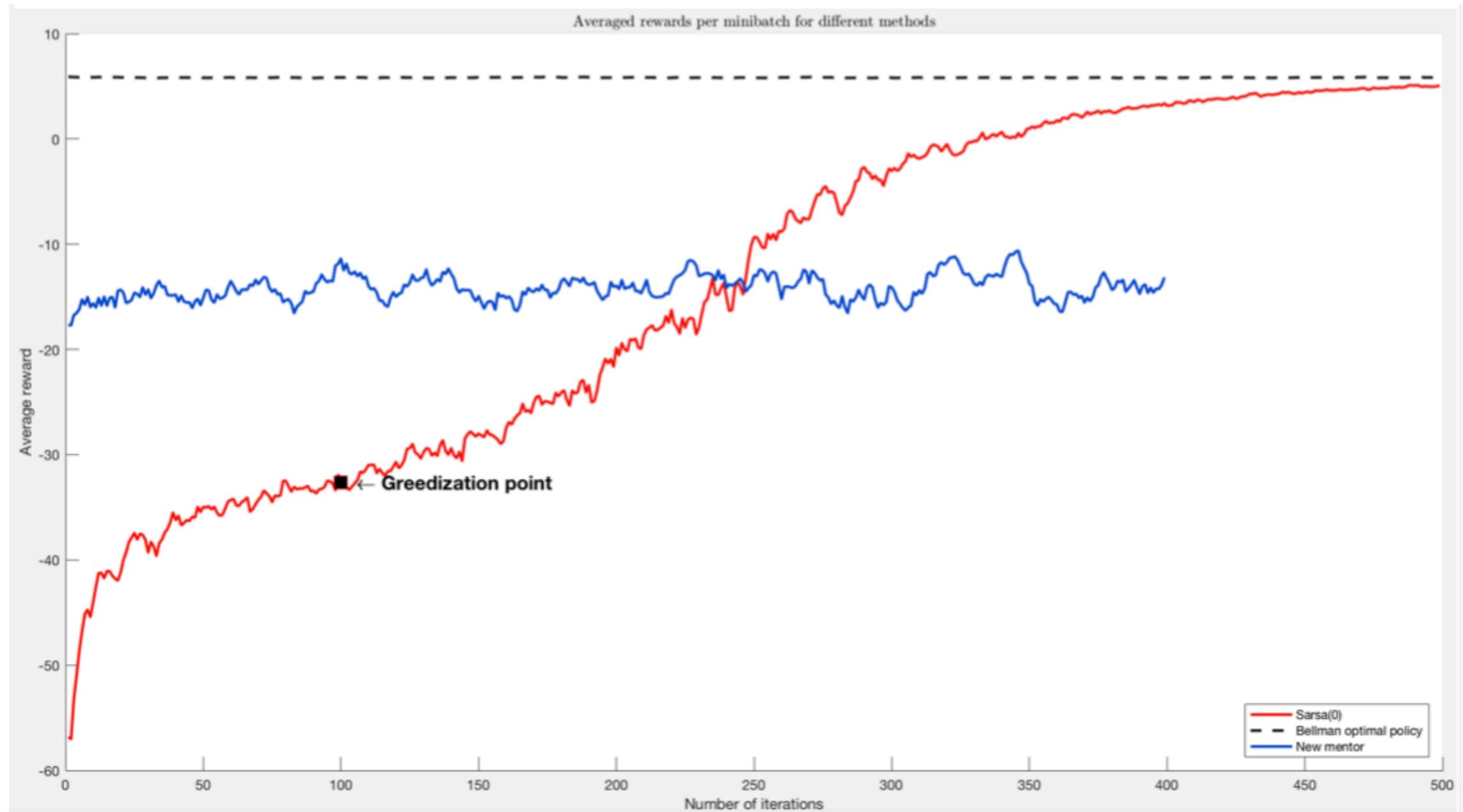
Results

■ GENERATING MENTORS



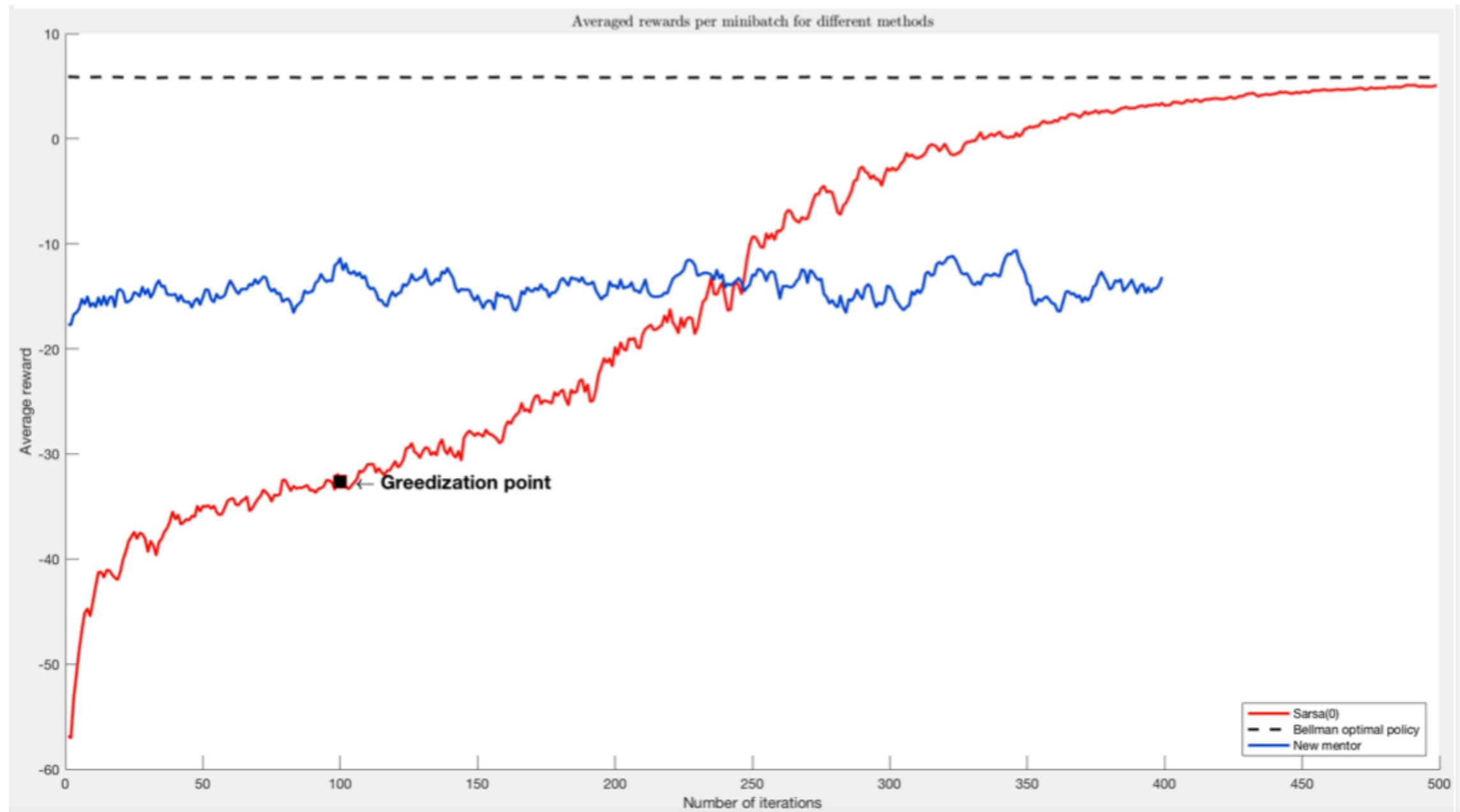
Results

■ GENERATING MENTORS



Results

■ GENERATING MENTORS



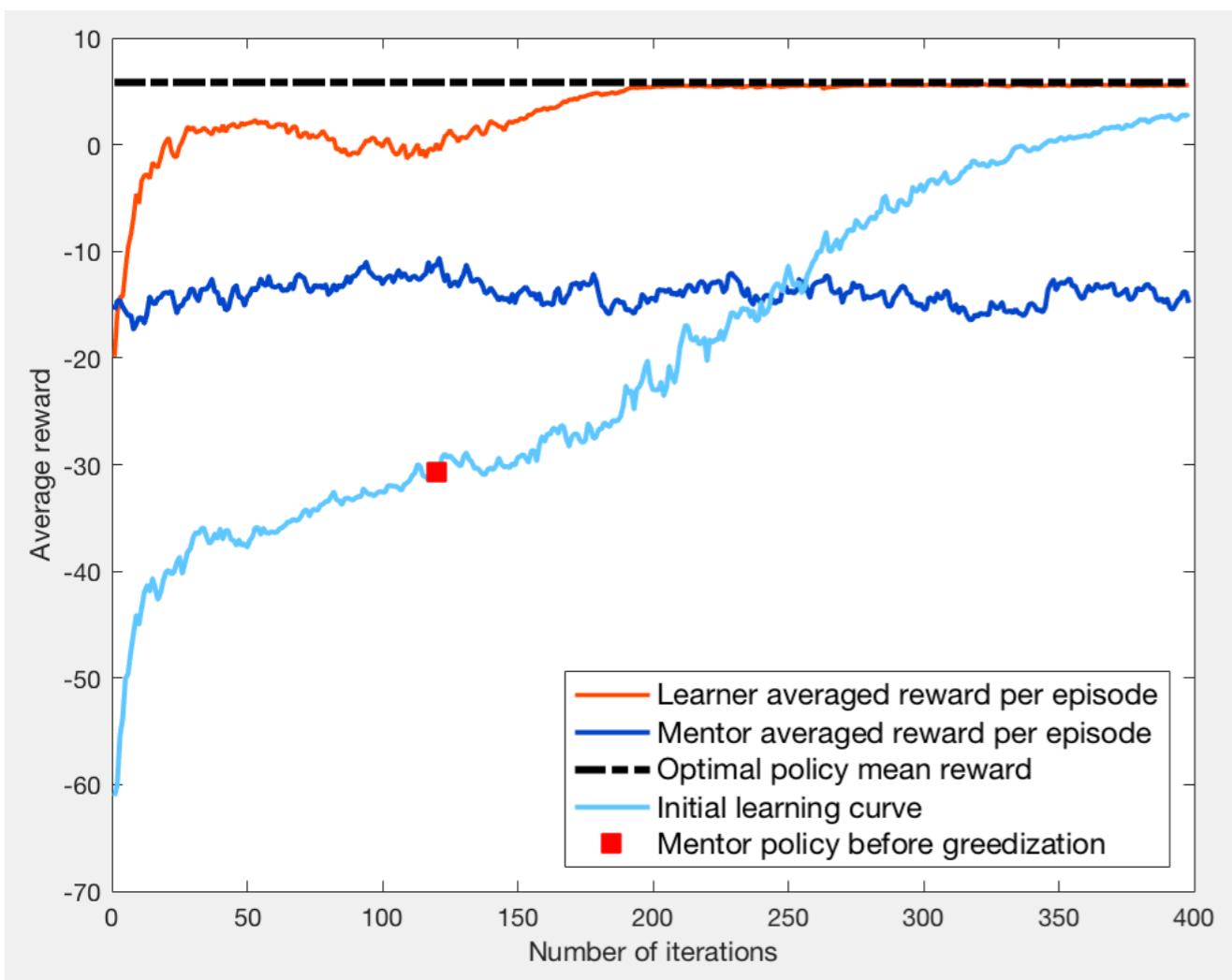
- Fairly strong hypothesis : **one mentor recommendation for every state**

Results

■ VANISHING COMPLIANCE (The naive way)

Results

■ VANISHING COMPLIANCE (The naive way)



**Figure : Learning Curve
(Teacher 1)**

Results

■ VANISHING COMPLIANCE (The naive way)

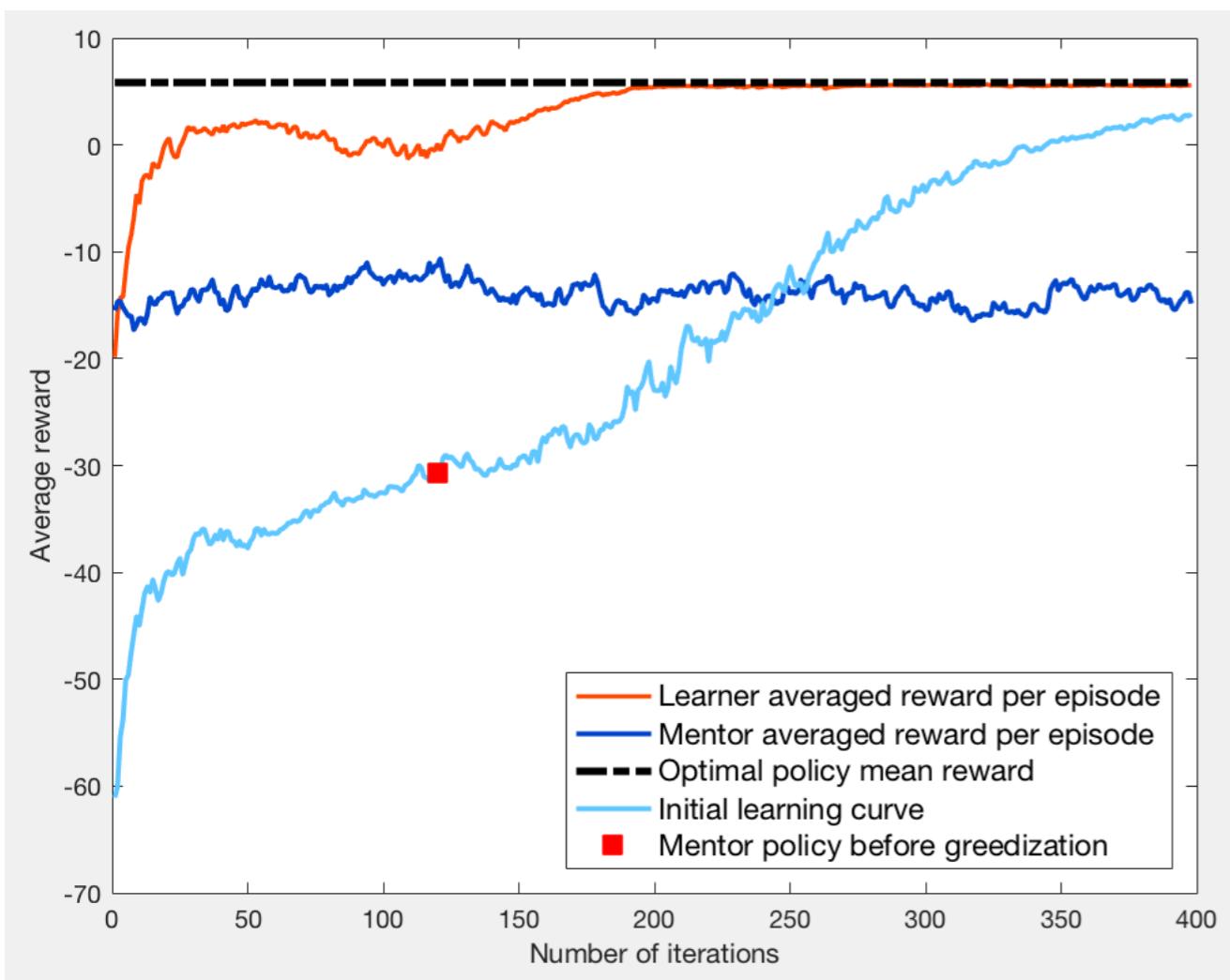


Figure : Learning Curve
(Teacher 1)

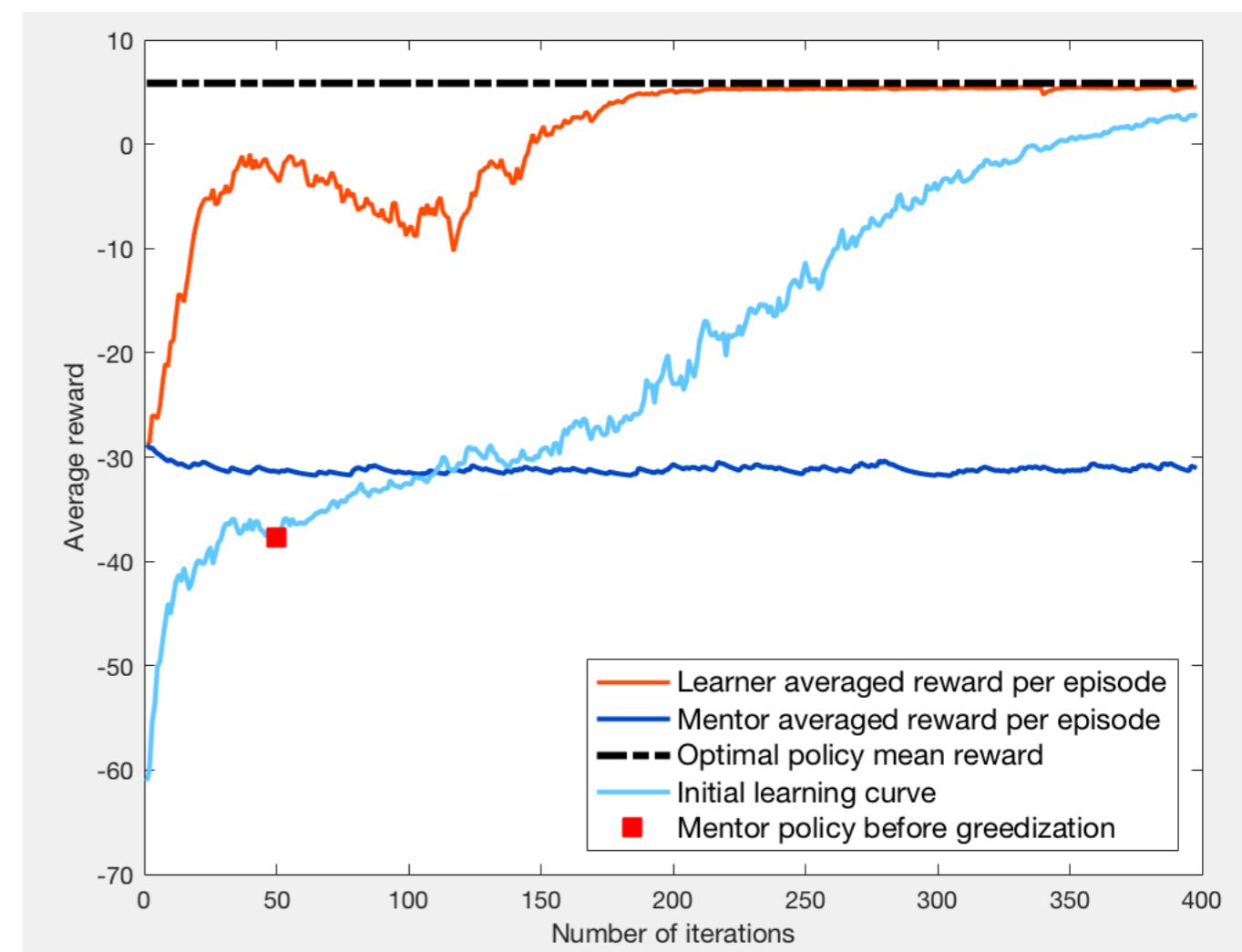


Figure : Learning Curve
(Teacher 2)

Results

■ VANISHING COMPLIANCE (The naive way)

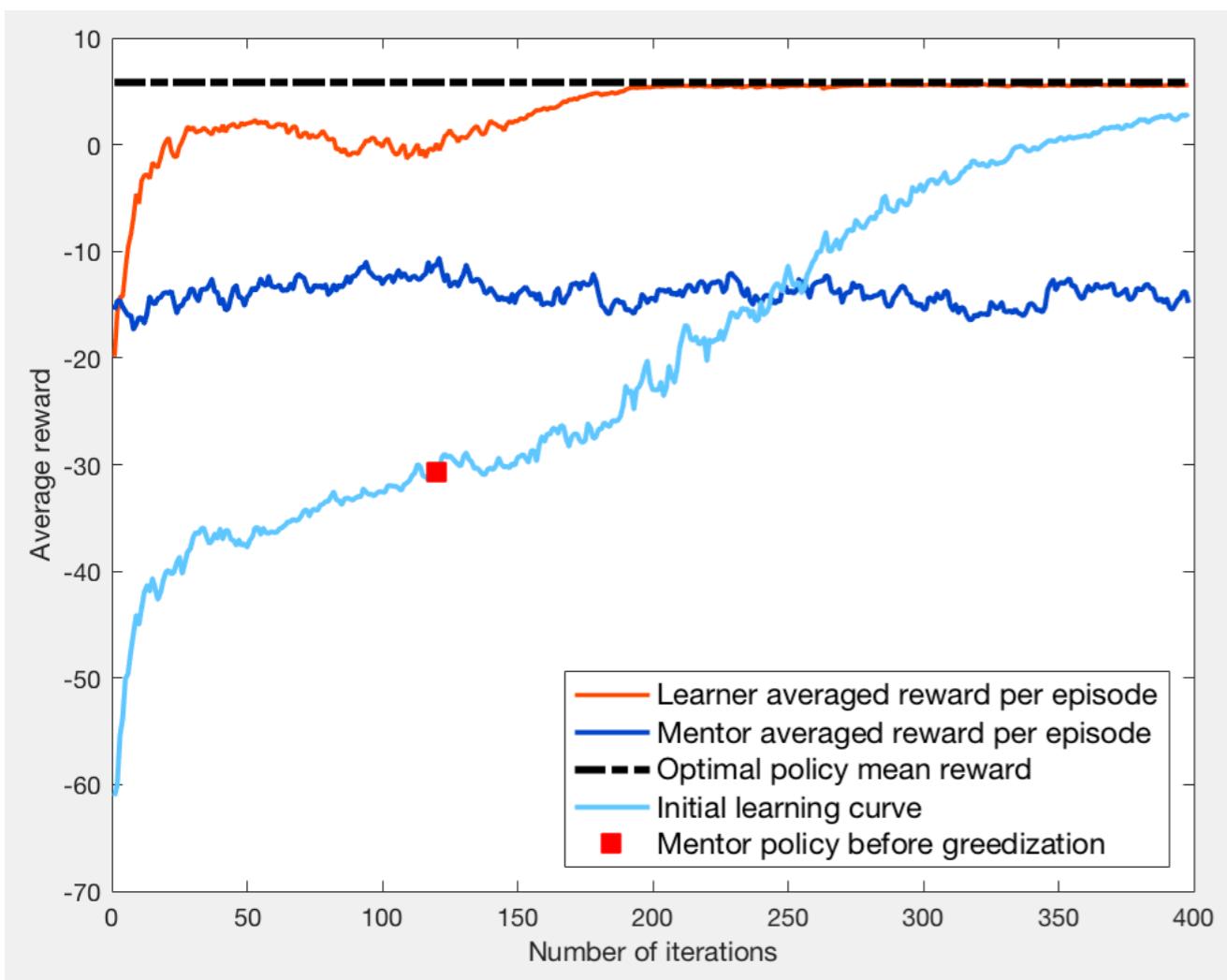


Figure : Learning Curve
(Teacher 1)

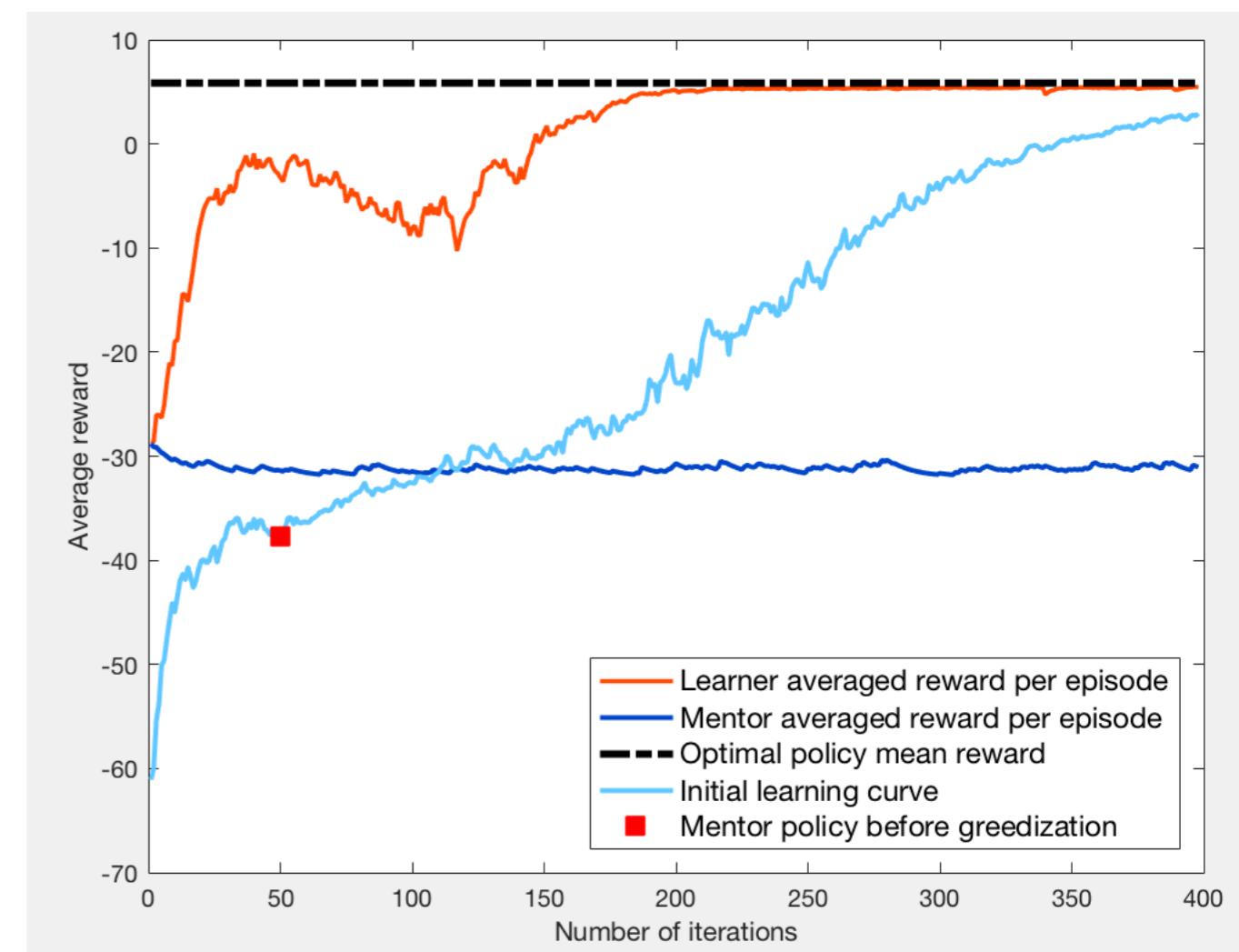


Figure : Learning Curve
(Teacher 2)

- Too much time spent exploring around good solutions !

Results

■ ADAPTIVE LEARNERS

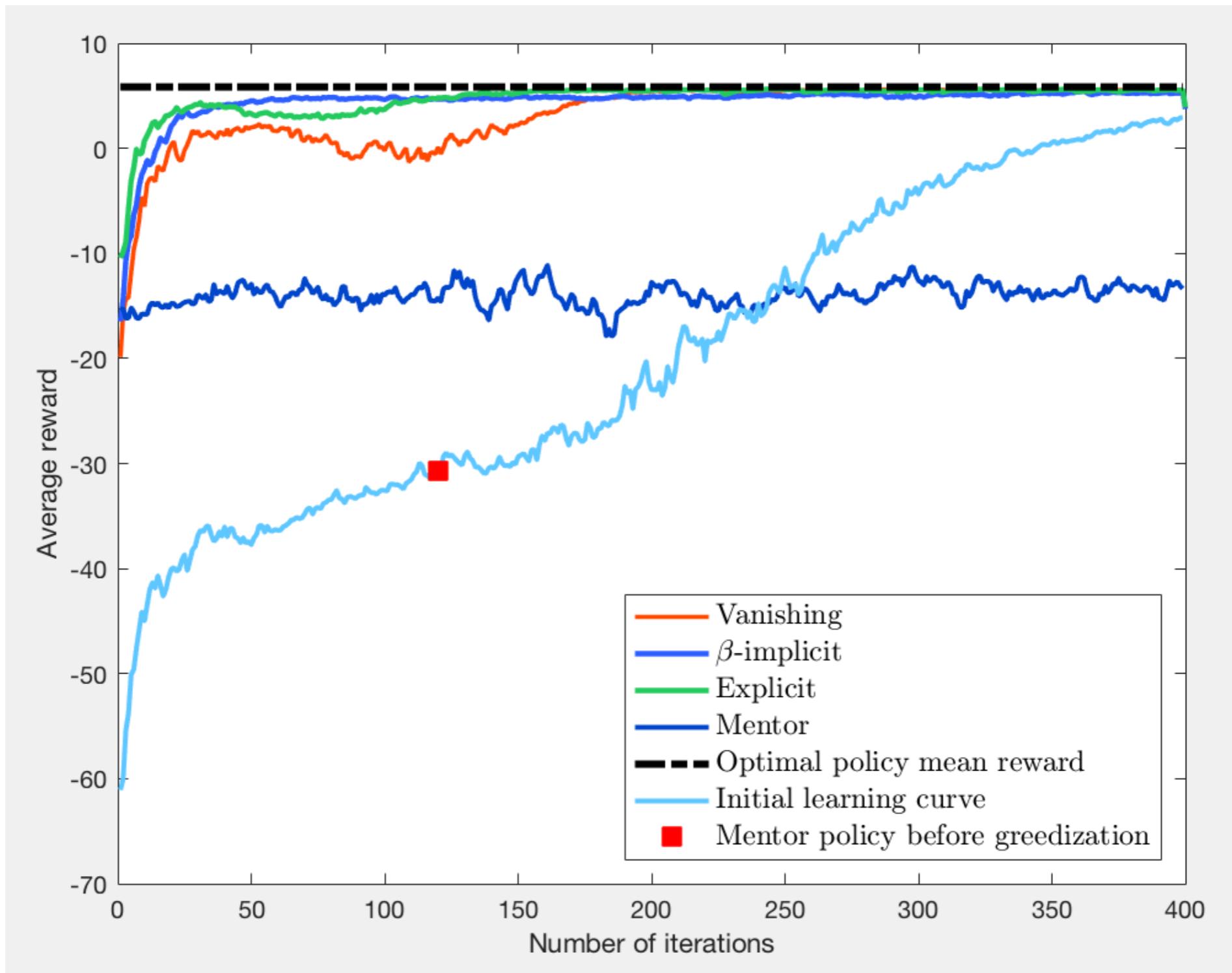


Figure : Learning Curves (Teacher 1)

Results

■ ADAPTIVE LEARNERS

Results

■ ADAPTIVE LEARNERS

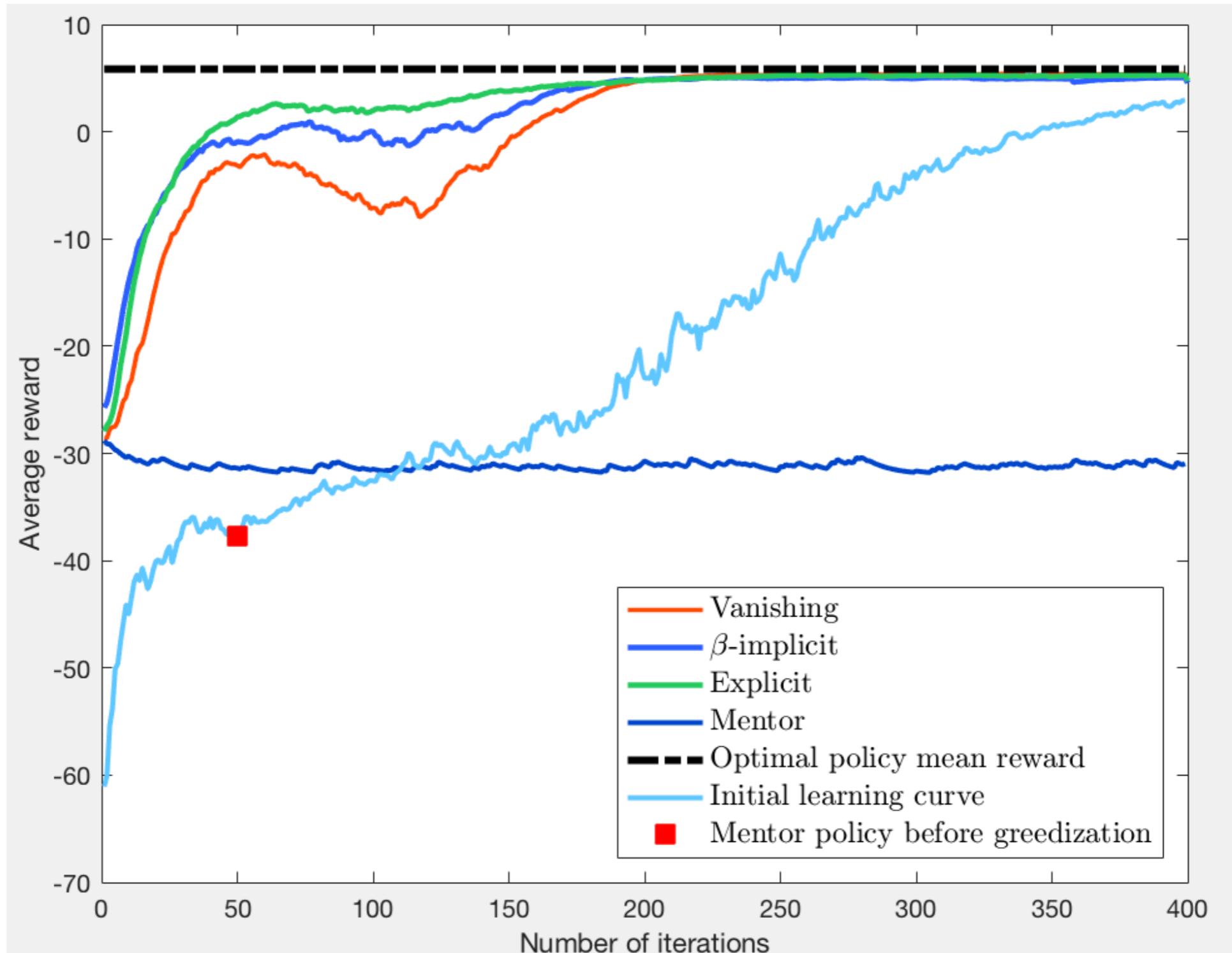


Figure : Learning Curves (Teacher 2)

Results

■ ADAPTIVE LEARNERS

- Mentor optimality : linear scaling between random policy and optimal policy reward

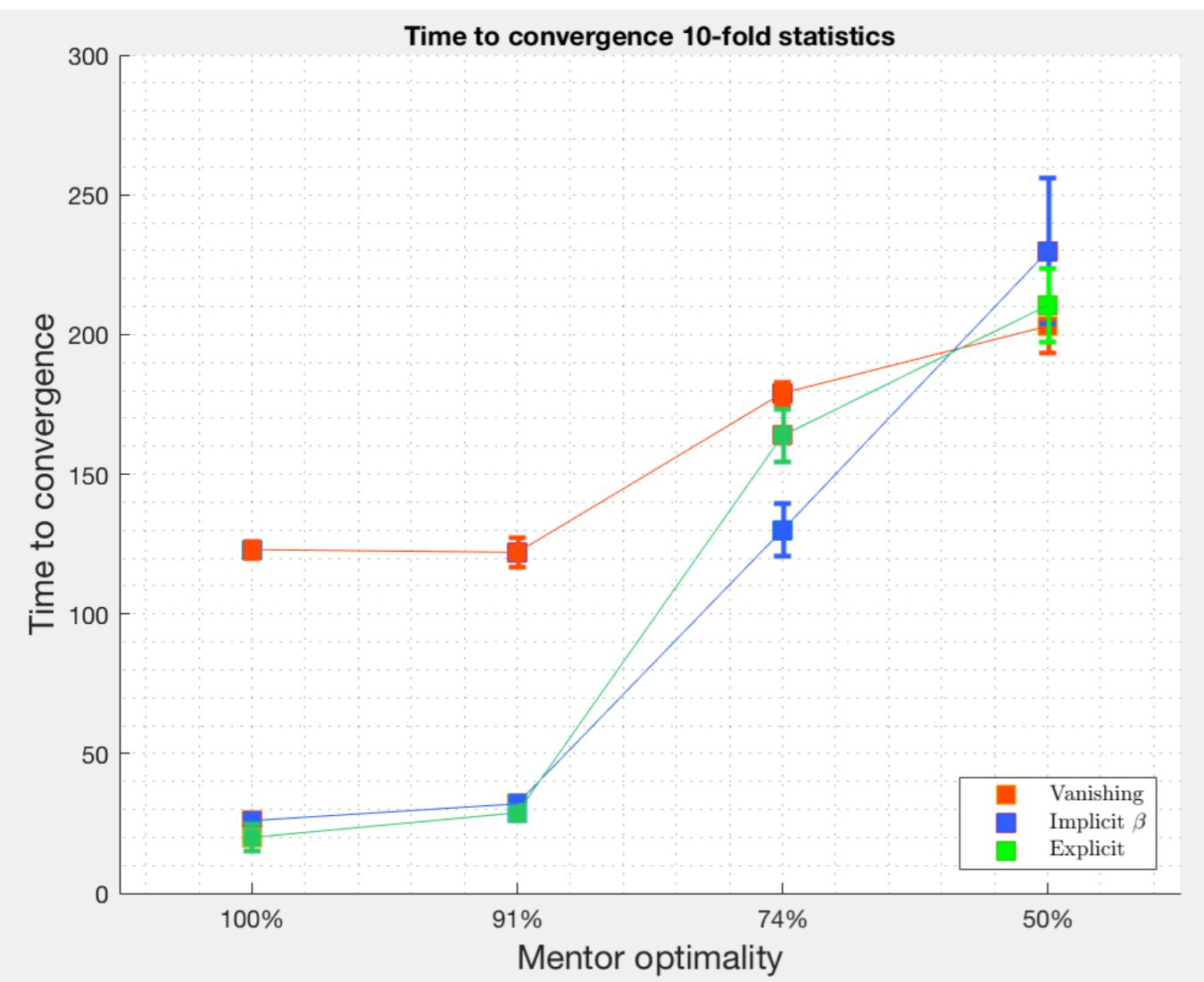


Figure : Time To Convergence

Results

■ ADAPTIVE LEARNERS

- Mentor optimality : linear scaling between random policy and optimal policy reward

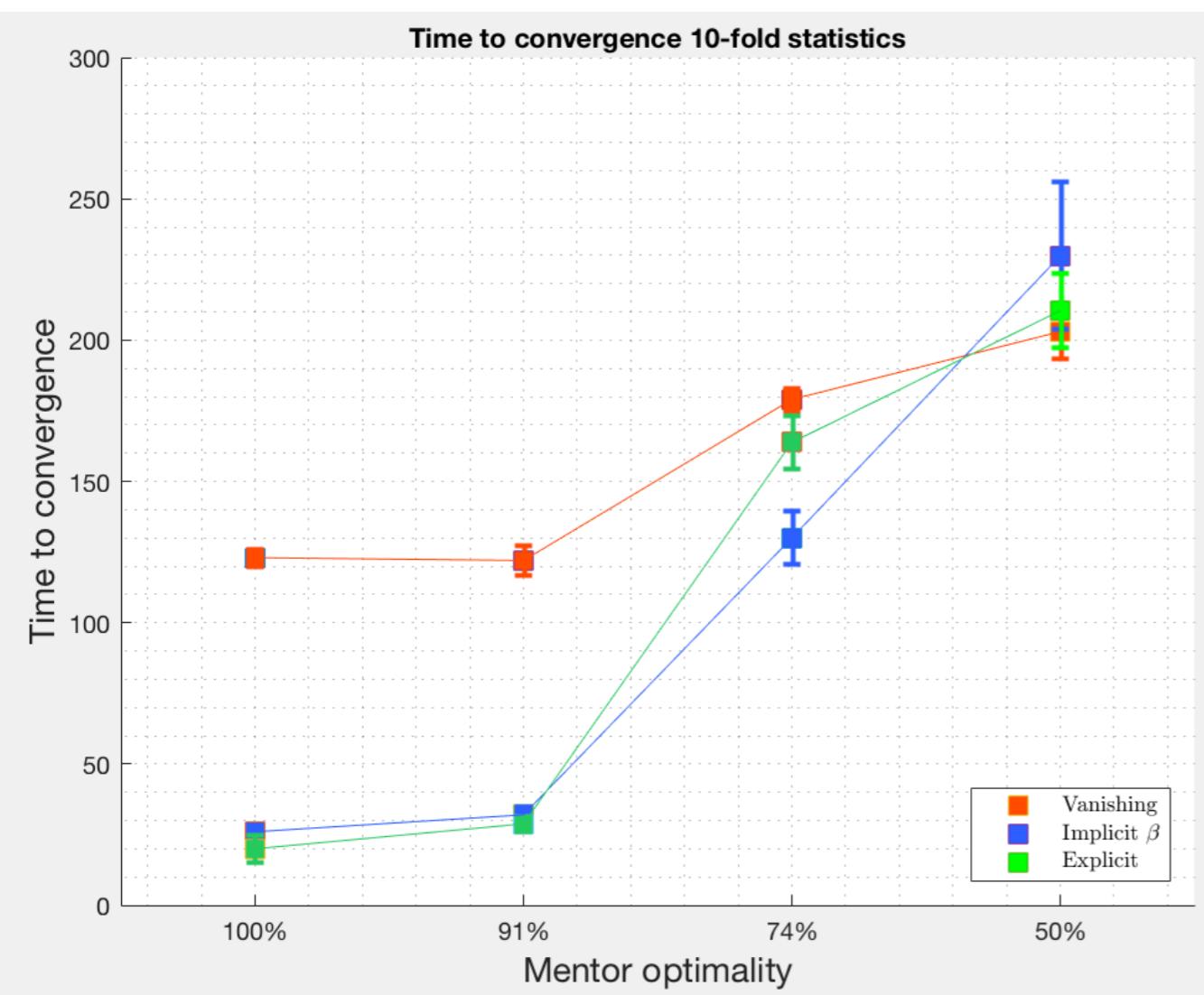


Figure : Time To Convergence

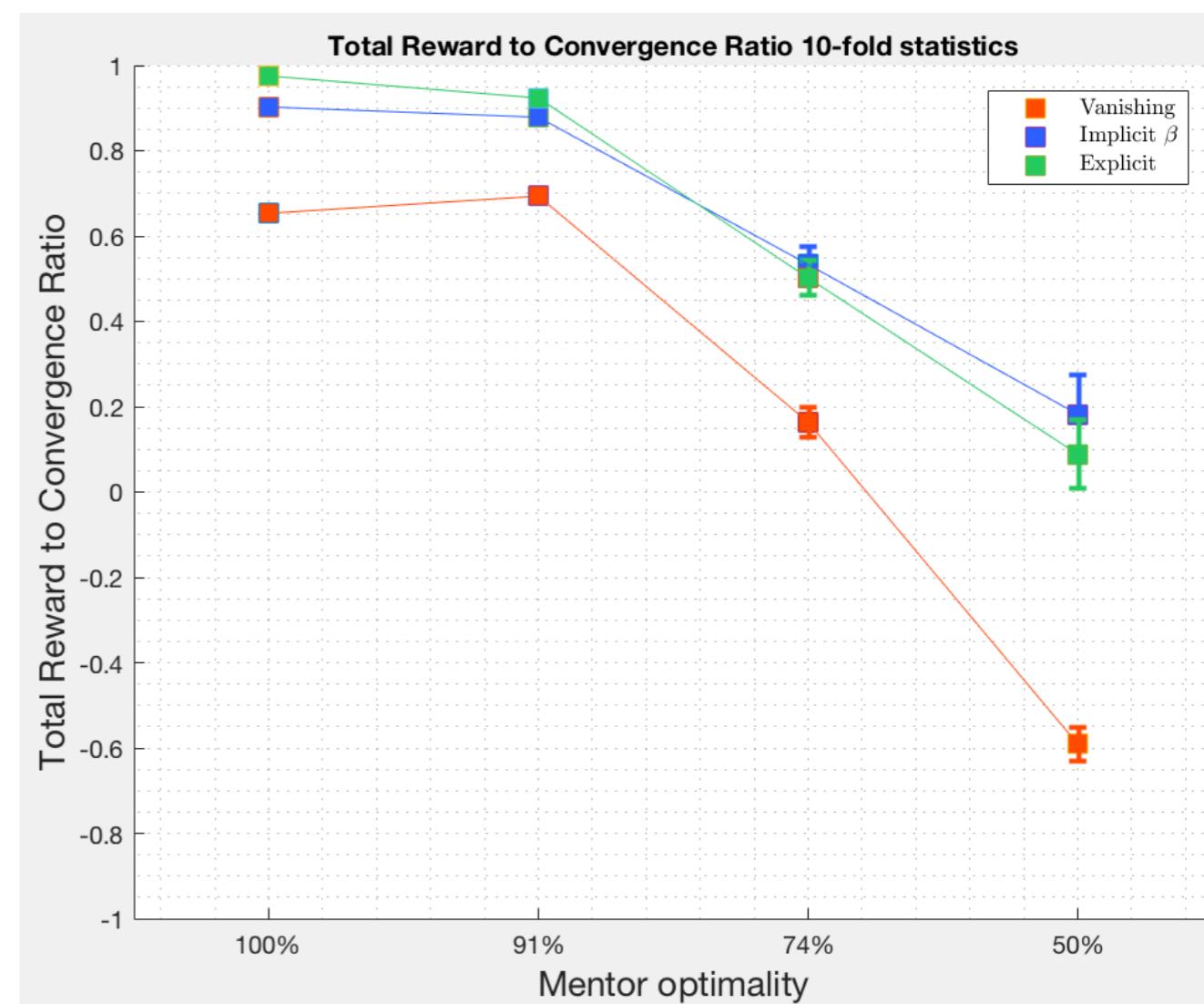


Figure : Reward Ratio to Convergence

Results

■ ADAPTIVE LEARNERS

- Compared to *classical learners*

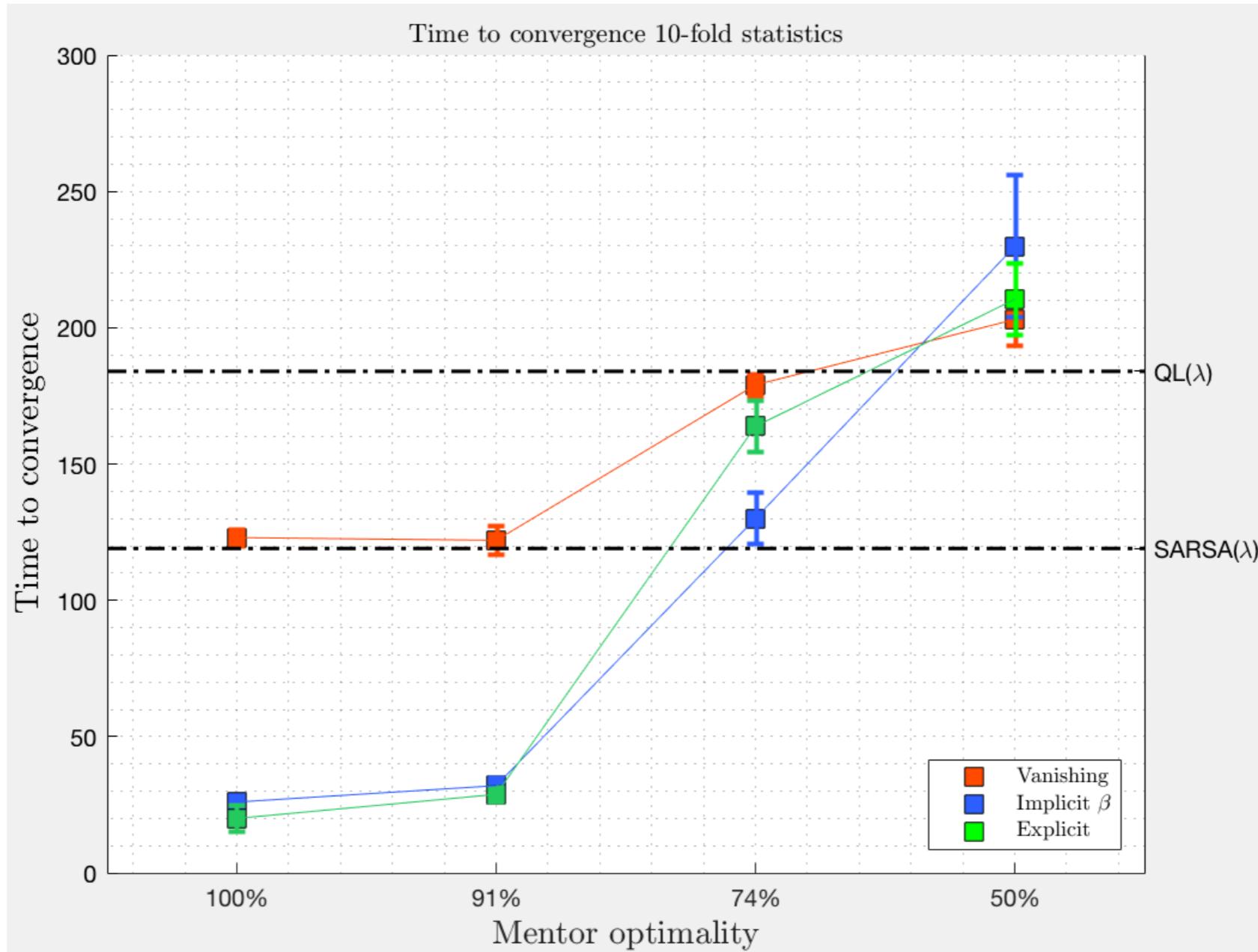
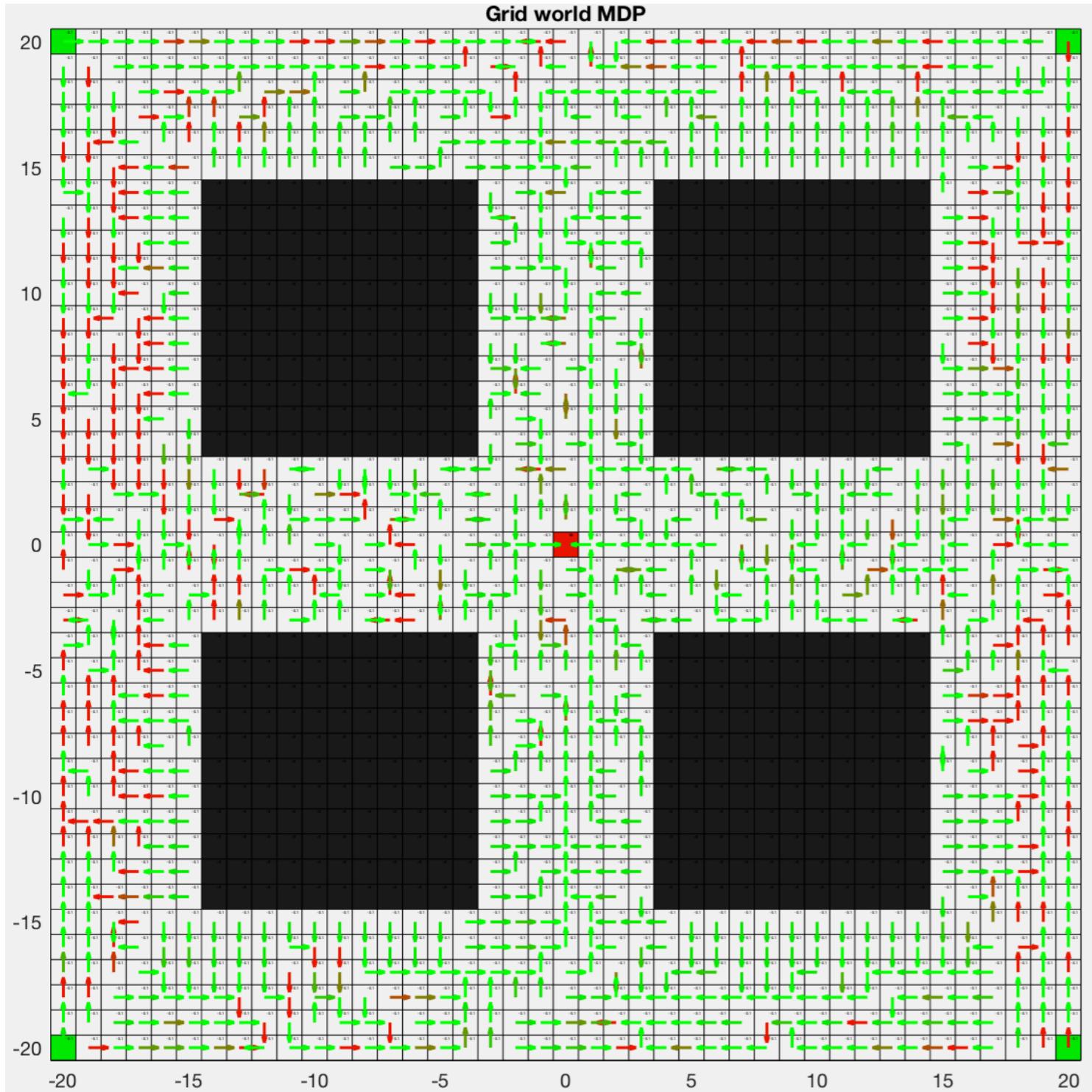


Figure : Time To Convergence

Results

■ ADAPTIVE LEARNERS

- What is actually learnt ?



- Compliance heat-map

- Poor teacher recommendations back propagate too far

- The learner tries to circle the teacher instead of fixing it !

Results

■ IMPROVEMENTS

- Can off-policy learning improve this ?

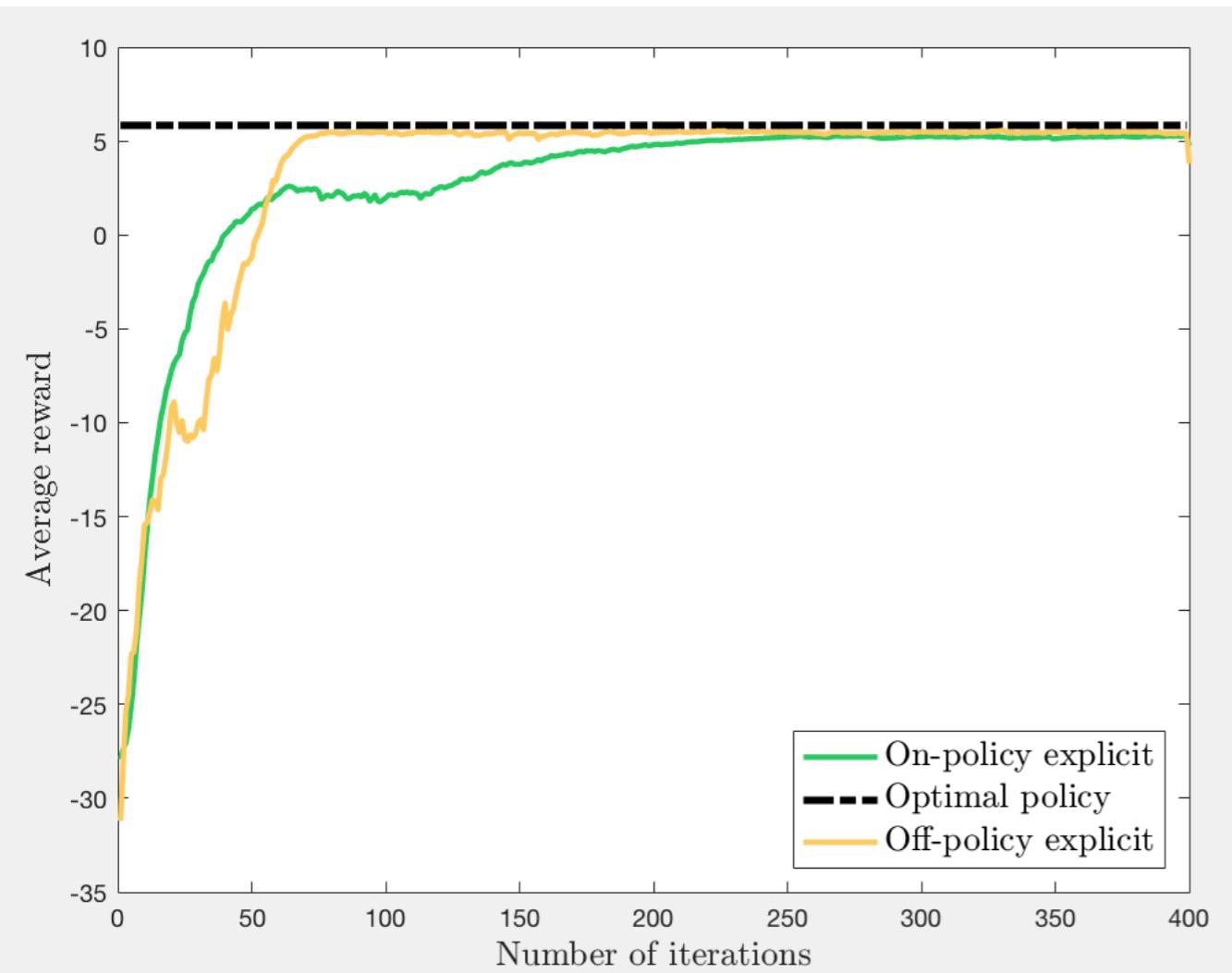


Figure : Off/On policy explicit compliance

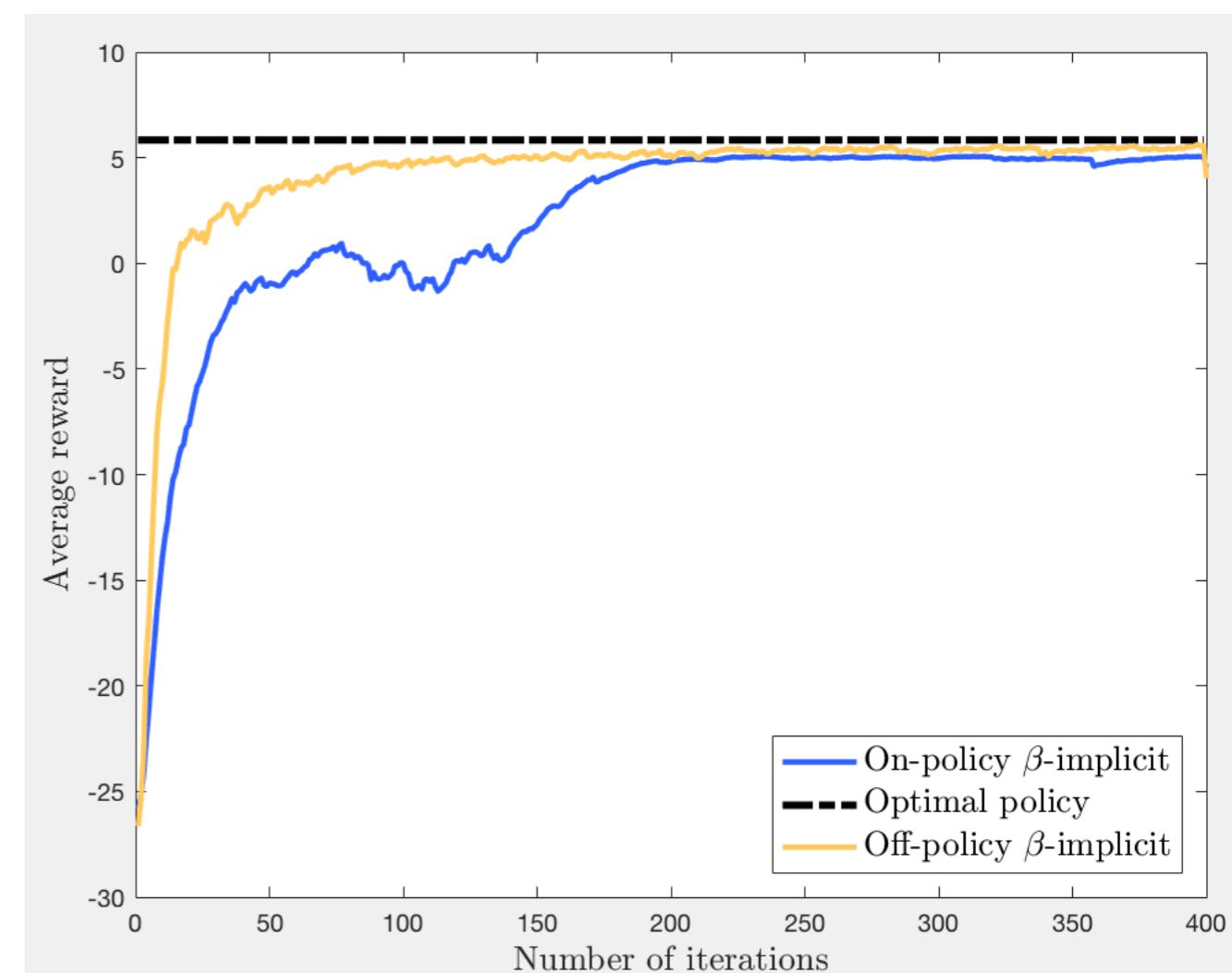


Figure : Off/On policy implicit compliance

- The learning now fixes the suboptimal regions !

■ WHAT WE DID :

- Provide **adaptive-compliant exploration policies**

■ WHAT'S NEXT :

■ WHAT WE DID :

- Provide **adaptive-compliant exploration policies**
 - Learn from suboptimal teachers

■ WHAT'S NEXT :

■ WHAT WE DID :

- Provide **adaptive-compliant exploration policies**
 - Learn from suboptimal teachers
 - Evaluate the optimality of a teacher

■ WHAT'S NEXT :

■ WHAT WE DID :

- Provide **adaptive-compliant exploration policies**
 - Learn from suboptimal teachers
 - Evaluate the optimality of a teacher
 - Extract useful informations

■ WHAT'S NEXT :

■ WHAT WE DID :

- Provide **adaptive-compliant exploration policies**
 - Learn from suboptimal teachers
 - Evaluate the optimality of a teacher
 - Extract useful informations
 - Speed-up the learning

■ WHAT'S NEXT :

■ WHAT WE DID :

- Provide **adaptive-compliant exploration policies**
 - Learn from suboptimal teachers
 - Evaluate the optimality of a teacher
 - Extract useful informations
 - Speed-up the learning

■ WHAT'S NEXT :

- Still some work to do !

■ WHAT WE DID :

- Provide **adaptive-compliant exploration policies**
 - Learn from suboptimal teachers
 - Evaluate the optimality of a teacher
 - Extract useful informations
 - Speed-up the learning

■ WHAT'S NEXT :

- Still some work to do !
 - Generalize to sparse recommendation

■ WHAT WE DID :

- Provide **adaptive-compliant exploration policies**
 - Learn from suboptimal teachers
 - Evaluate the optimality of a teacher
 - Extract useful informations
 - Speed-up the learning

■ WHAT'S NEXT :

- Still some work to do !
 - Generalize to sparse recommendation
 - Implement eligibility traces

■ WHAT WE DID :

- Provide **adaptive-compliant exploration policies**
 - Learn from suboptimal teachers
 - Evaluate the optimality of a teacher
 - Extract useful informations
 - Speed-up the learning

■ WHAT'S NEXT :

- Still some work to do !
 - Generalize to sparse recommendation
 - Implement eligibility traces
 - Test in continuous MDP

THANK YOU FOR YOUR ATTENTION !

■ ELIGIBILITY TRACES

- 1-step return :

$$R_t^{(1)} = r_{t+1} + \gamma Q(s', a') \longrightarrow \text{1-step backup}$$

- n-step return :

$$R_t^{(n)} = r_{t+1} + \gamma r_{t+2} + \dots + \gamma^{n-1} r_{t+n} + \gamma^n Q(s'_{t+n}, a) \longrightarrow \text{n-step backup}$$

- Eligibility traces : backup to an average of all n-step backups !

\longrightarrow *backward view implementation*

⁽¹⁾ Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*

Appendix

■ OFF-POLICY COMPLIANT LEARNER

