Compatible Reward Inverse Reinforcement Learning

Alberto Maria Metelli

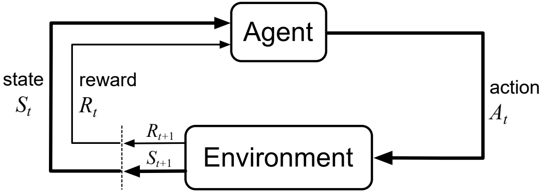
Supervisor: Marcello Restelli
Co-supervisor: Matteo Pirotta

Politecnico di Milano
M.Sc. in Computer Science and Engineering

27th July 2017

Reinforcement Learning

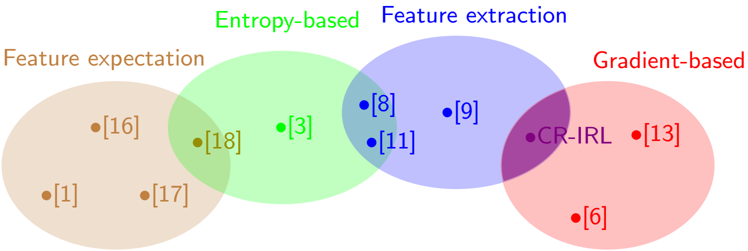
- *Reinforcement Learning* (RL) [14]:
    - learning by interaction.
- Parametric policy: $\pi_{\boldsymbol{\theta}}$.
- Expected return: $J(\boldsymbol{\theta}) = \underset{\tau \sim p_{\boldsymbol{\theta}}}{\mathbb{E}} \left[\sum_{t=0}^{T(\tau)} \gamma^t R(s_{\tau,t}, a_{\tau,t}) \right]$.



State of the Art

- RL requires a *reward function* $R(s, a)$.
- Designing a suitable reward function is challenging (e.g., car driving task [1]).
- Learning from demonstrations (*Imitation Learning*):
  - Behavioral Cloning (BC) [2];
  - Inverse Reinforcement Learning (IRL) [12].

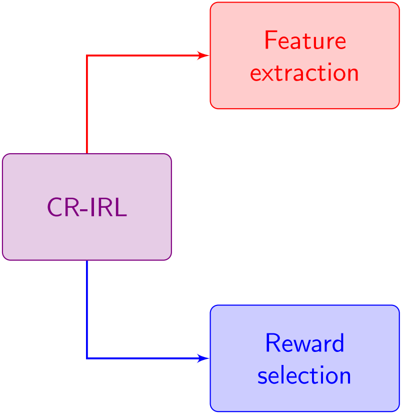
Motivations and Goals



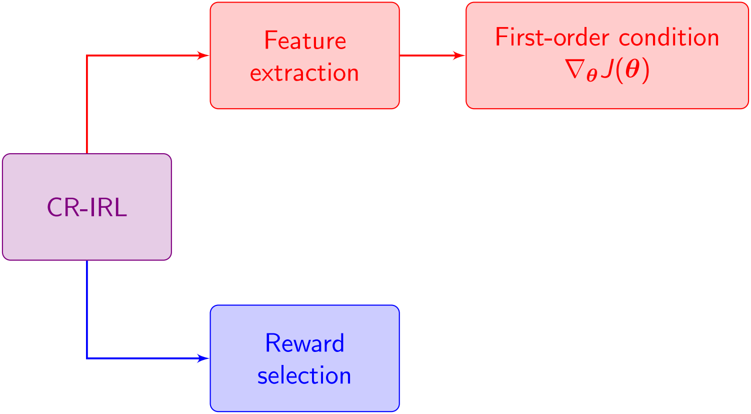
- **Motivations** – state-of-the-art IRL algorithms require:
  - the environment transition model;
  - a set of engineered reward features.
- **Goal** – design an IRL algorithm requiring only expert's trajectories.

Compatible Reward Inverse Reinforcement Learning

Overview

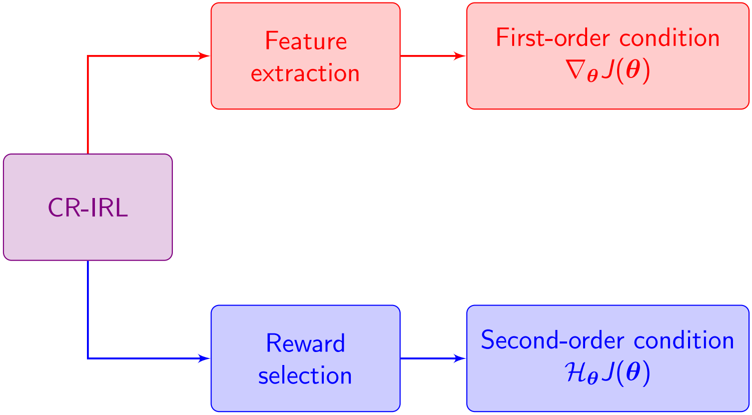


Compatible Reward Inverse Reinforcement Learning
Overview



Compatible Reward Inverse Reinforcement Learning

Overview



Feature extraction

Expert's COmpatible Value Features

- Extract the *compatible* value functions.
- First-order condition on *policy gradient* [15]:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \int_{\mathcal{S}} \int_{\mathcal{A}} \delta_{\mu,\gamma}^{\pi_{\boldsymbol{\theta}}}(s,a) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s) Q^{\pi_{\boldsymbol{\theta}}}(s,a) \mathrm{d}s \mathrm{d}a = \mathbf{0}.$$

- *Expert's COmpatible Value Features* (ECO-Q):

- Extension to unvisited state-action pairs with KNN.

Feature extraction

Expert's COmpatible Value Features

- Extract the *compatible* value functions.
- First-order condition on *policy gradient* [15]:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \int_{\mathcal{S}} \int_{\mathcal{A}} \delta_{\mu,\gamma}^{\pi_{\boldsymbol{\theta}}}(s,a) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s) Q^{\pi_{\boldsymbol{\theta}}}(s,a) \mathrm{d}s\mathrm{d}a = \mathbf{0}.$$

- *Expert's COmpatible Value Features* (ECO-Q):

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbf{0}$$

- Extension to unvisited state-action pairs with KNN.

Feature extraction

Expert's COmpatible Value Features

- Extract the *compatible* value functions.
- First-order condition on *policy gradient* [15]:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \int_{\mathcal{S}} \int_{\mathcal{A}} \delta_{\mu,\gamma}^{\pi_{\boldsymbol{\theta}}}(s,a) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s) Q^{\pi_{\boldsymbol{\theta}}}(s,a) \mathrm{d}s \mathrm{d}a = \mathbf{0}.$$

- *Expert's COmpatible Value Features* (ECO-Q):

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbf{0} \longrightarrow \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}} \perp Q^{\pi_{\boldsymbol{\theta}}}$$

- Extension to unvisited state-action pairs with KNN.

Feature extraction

Expert's COmpatible Value Features

- Extract the *compatible* value functions.
- First-order condition on *policy gradient* [15]:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \int_{\mathcal{S}} \int_{\mathcal{A}} \delta_{\mu,\gamma}^{\pi_{\boldsymbol{\theta}}}(s,a) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s) Q^{\pi_{\boldsymbol{\theta}}}(s,a) \mathrm{d}s\mathrm{d}a = \mathbf{0}.$$

- *Expert's COmpatible Value Features* (ECO-Q):

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbf{0} \longrightarrow \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}} \perp Q^{\pi_{\boldsymbol{\theta}}} \longrightarrow \boldsymbol{\Phi} = \mathrm{null}\left(\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}{}^T \mathbf{D}_{\mu,\gamma}^{\pi_{\boldsymbol{\theta}}}\right)$$

- Extension to unvisited state-action pairs with KNN.

Feature extraction

Expert's COmpatible Reward Features

- Extract the *compatible* reward functions.
- *Expert's COmpatible Reward Features* (ECO-R):
  - model-based ECO-R

$$\mathbf{\Psi}^{MB} = (\mathbf{I} - \gamma \mathbf{P} \boldsymbol{\pi}_{\boldsymbol{\theta}})\mathbf{\Phi};$$

  - model-free ECO-R

$$\mathbf{\Psi}^{MF} = \left(\mathbf{I} - \tilde{\boldsymbol{\pi}}_{\boldsymbol{\theta}}\right)\mathbf{\Phi}.$$

Reward selection

Preliminaries

- Linear reward parametrization:

$$\mathbf{r} = \mathbf{\Psi}\boldsymbol{\omega}.$$

- Select a reward function that:
  - is maximum of $J(\boldsymbol{\theta})$;
  - penalizes maximally deviations from the expert's policy.
- Second-order conditions on *policy Hessian* [10].

Reward selection

Second-Order criteria

- Second-order optimality criteria:
  - minimize the *maximum eigenvalue*;
  - minimize the *trace*.
- Negative semidefinite Hessian as constraint.
- Second-order *trace heuristic*:
  - reduced space of ECO-R;
  - closed-form solution:

$$\omega = \frac{\mathbf{tr}}{\|\mathbf{tr}\|_2}.$$

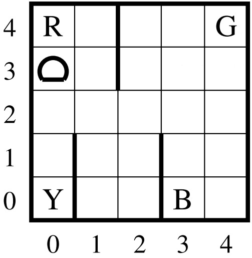
Experiments

- Environments:
  - **Taxi problem** [4] (finite);
  - Linear-Quadratic Gaussian Regulator [5] (continuous);
  - **Car on the Hill** [7] (continuous).
- Metrics:
  - **Learning speed**;
  - Average return;
  - Parameter distance;
  - Policy distance (KL-divergence).

Taxi

Preliminaries

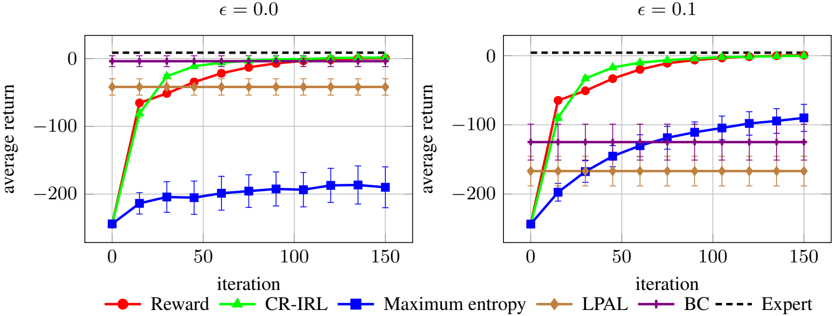
- Finite episodic problem.
- Expert's ϵ-Boltzmann policy with state-dependent features.



Taxi

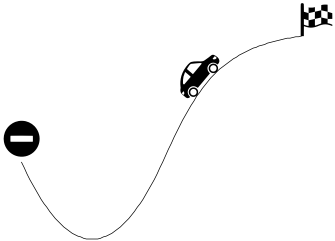
Learning speed

- Comparison with Maximum Entropy IRL [18], LPAL [16] and BC.



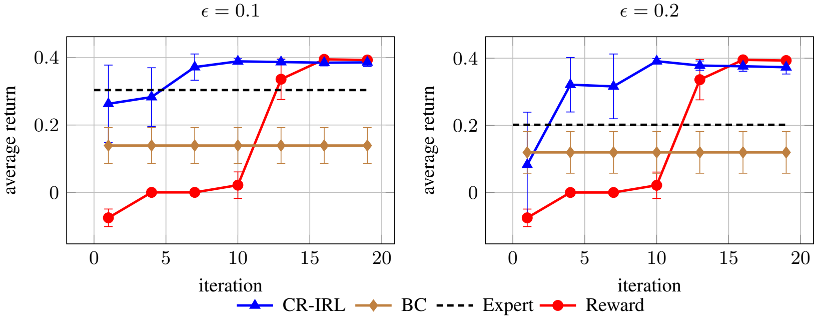
Car on the Hill
Preliminaries

- Continuous episodic problem.
- Expert's policy computed via FQI [7].



Car on the Hill

Learning speed



Conclusions

- Contributions
  - Construction of both features and reward function.
  - Faster learning speed w.r.t. the original reward function.
  - Better performance w.r.t. BC and several IRL methods.
- Paper submitted to NIPS.
- Future Works
  - Theoretical analysis of the maximum likelihood policy.
  - Direct construction of ECO-R.

References I

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML*, page 1. ACM, 2004.

[2] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[3] Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In *AISTATS*, pages 182–189, 2011.

[4] Thomas G. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000.

[5] Peter Dorato, Vito Cerone, and Chaouki Abdallah. *Linear Quadratic Control: An Introduction*. Krieger Publishing Co., Inc., Melbourne, FL, USA, 2000.

[6] Peter Englert and Marc Toussaint. Inverse kkt–learning cost functions of manipulation tasks from demonstrations. In *Proceedings of the International Symposium of Robotics Research*, 2015.

References II

[7] Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6(Apr):503–556, 2005.

[8] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 49–58. JMLR.org, 2016.

[9] Sergey Levine, Zoran Popovic, and Vladlen Koltun. Feature construction for inverse reinforcement learning. In *NIPS*, pages 1342–1350. Curran Associates, Inc., 2010.

[10] Giorgio Manganini, Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Following newton direction in policy gradient with parameter exploration. In *IJCNN*, pages 1–8. IEEE, 2015.

[11] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

References III

[12] Andrew Y. Ng, Stuart J. Russell, et al. Algorithms for inverse reinforcement learning. In *ICML*, pages 663–670, 2000.

[13] Matteo Pirotta and Marcello Restelli. Inverse reinforcement learning through policy gradient minimization. In *AAAI*, pages 1993–1999, 2016.

[14] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[15] Richard S. Sutton, David A. McAllester, Satinder P. Singh, Yishay Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, volume 99, pages 1057–1063, 1999.

[16] Umar Syed, Michael H. Bowling, and Robert E. Schapire. Apprenticeship learning using linear programming. In *ICML*, volume 307 of *ACM International Conference Proceeding Series*, pages 1032–1039. ACM, 2008.

[17] Umar Syed and Robert E. Schapire. A game-theoretic approach to apprenticeship learning. In *NIPS*, pages 1449–1456, 2007.

[18] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

Policy Rank

- How large is the space of ECO-Qs?

Definition

Let $\pi_{\boldsymbol{\theta}}$ a policy with k parameters belonging to the class Π_Θ and differentiable in $\boldsymbol{\theta}$. The policy rank is the dimension of the space of the linear combinations of the partial derivatives of $\pi_{\boldsymbol{\theta}}$ w.r.t. $\boldsymbol{\theta}$:

$$\mathrm{rank}(\pi_{\boldsymbol{\theta}}) = \dim(\Gamma_{\pi_{\boldsymbol{\theta}}}), \quad \Gamma_{\pi_{\boldsymbol{\theta}}} = \{\nabla_{\boldsymbol{\theta}}\pi_{\boldsymbol{\theta}}\boldsymbol{\alpha} : \boldsymbol{\alpha} \in \mathbb{R}^k\}.$$

- The policy rank quantifies how much a policy is *informative* for recovering the optimal value function (and so the optimal reward function).
- In finite domains it holds: $\mathrm{rank}(\pi_{\boldsymbol{\theta}}) \leq \min\{k, |\mathcal{S}||\mathcal{A}| - |\mathcal{S}|\}$.

Reward shaping

- Given a reward function R inducing an optimal policy π, which is the class of rewards preserving the optimality of π?
- Optimality of π is preserved for *potential-based* shaping functions:

$$R'(s,a) = R(s,a) + \gamma \int_{\mathcal{S}} P(s'|s,a)\chi(s')\mathrm{d}s' - \chi(s)$$

- A smart choice is $\chi(s) = V^{\pi}(s)$, so we get the *advantage function*:

$$R'(s,a) = Q^{\pi}(s,a) - V^{\pi}(s,a) = A^{\pi}(s,a).$$

- The advantage function allows running RL algorithms with smaller γ.

Multi-objective second-order criteria

- Ideally we would like to minimize "all" the eigenvalues of the policy Hessian.

$$\begin{aligned} \underset{\boldsymbol{\omega}\in\mathbb{R}^p}{\text{minimize}} \quad & \boldsymbol{\lambda}(\boldsymbol{\omega}) = \left(\lambda_1(\boldsymbol{\omega}), \lambda_2(\boldsymbol{\omega}), ..., \lambda_k(\boldsymbol{\omega})\right) \\ \text{subject to} \quad & \mathcal{H}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{\omega}) + \epsilon \mathbf{I} \preceq 0. \end{aligned}$$

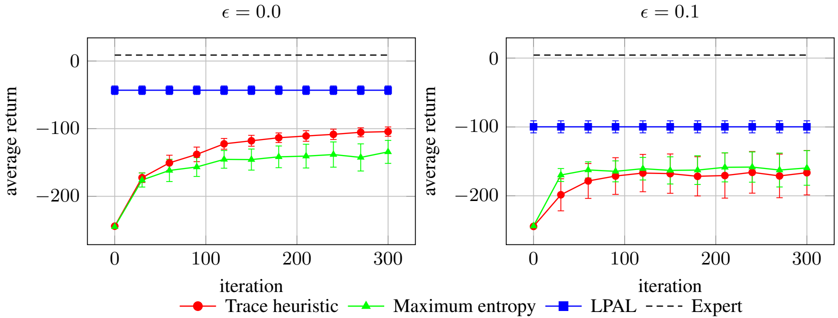
- We consider linear scalarizations:

$$L(\boldsymbol{\lambda}(\boldsymbol{\omega}), \gamma) = \sum_{i=1}^{k} \gamma_i \lambda_i(\boldsymbol{\omega}) = \boldsymbol{\gamma}^T \boldsymbol{\lambda}(\boldsymbol{\omega}).$$

- $\gamma_1 = 1$ and $\gamma_i = 0$ for $i = 2, 3, ..., k$ we get *maximum eigenvalue* optimality criterion.
- $\gamma_i = 1$ for $i = 1, 2, ..., k$ we get *trace* optimality criterion.

Taxi

Comparison with PVF



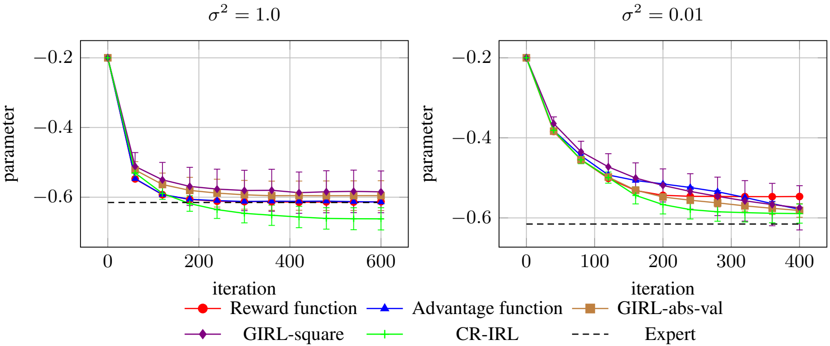
Linear Quadratic Gaussian Regulator (LQG)

Preliminaries

- Continuous infinite-horizon problem.
- Gaussian (noisy) expert's policy, the mean is the optimal action.
- Variance to test resilience to imperfect experts.

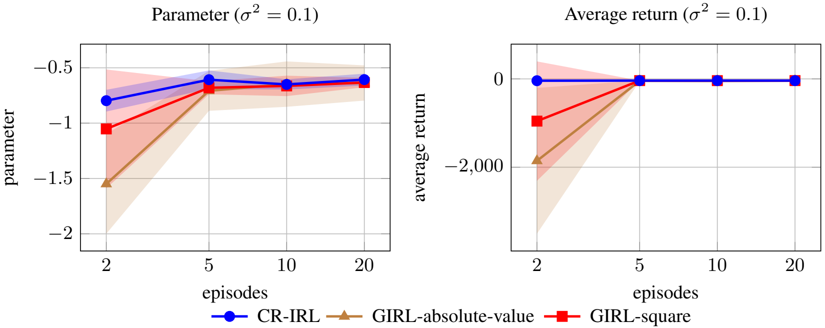
1D-LQG
Learning speed

- Comparison with GIRL.
- Train a Gaussian policy with REINFORCE.



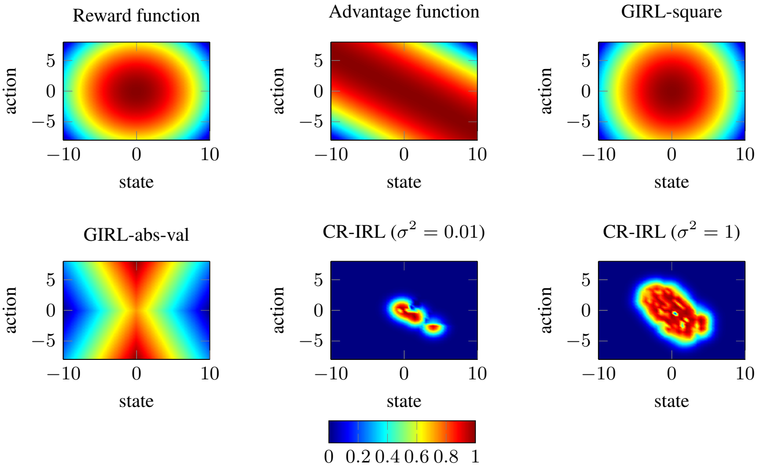
1D-LQG

Sensitivity to the number of expert's demonstrations

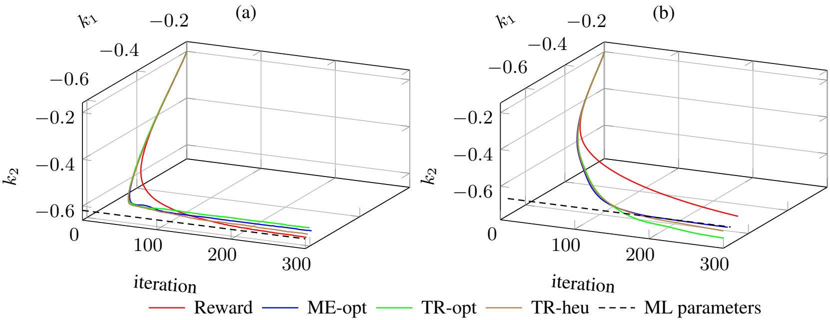


1D-LQG

Recovered rewards

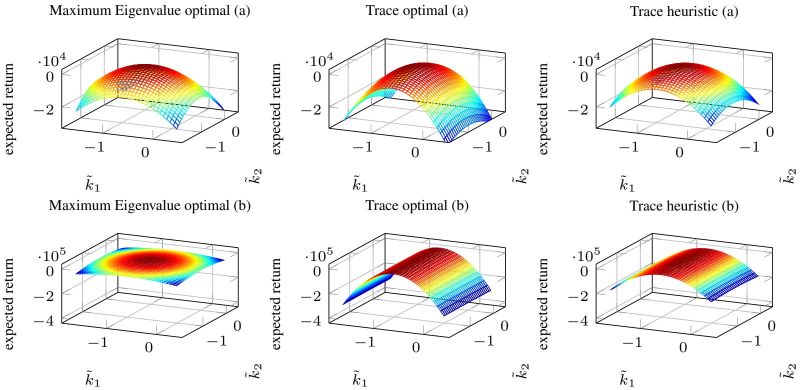


2D-LQG
Learning speed



2D-LQG

Shape of expected return



Car on the Hill

Trajectories