Sequential Transfer in Reinforcement Learning with a Generative Model

Riccardo Poiani
riccardo.poiani@mail.polimi.it
CSE Track

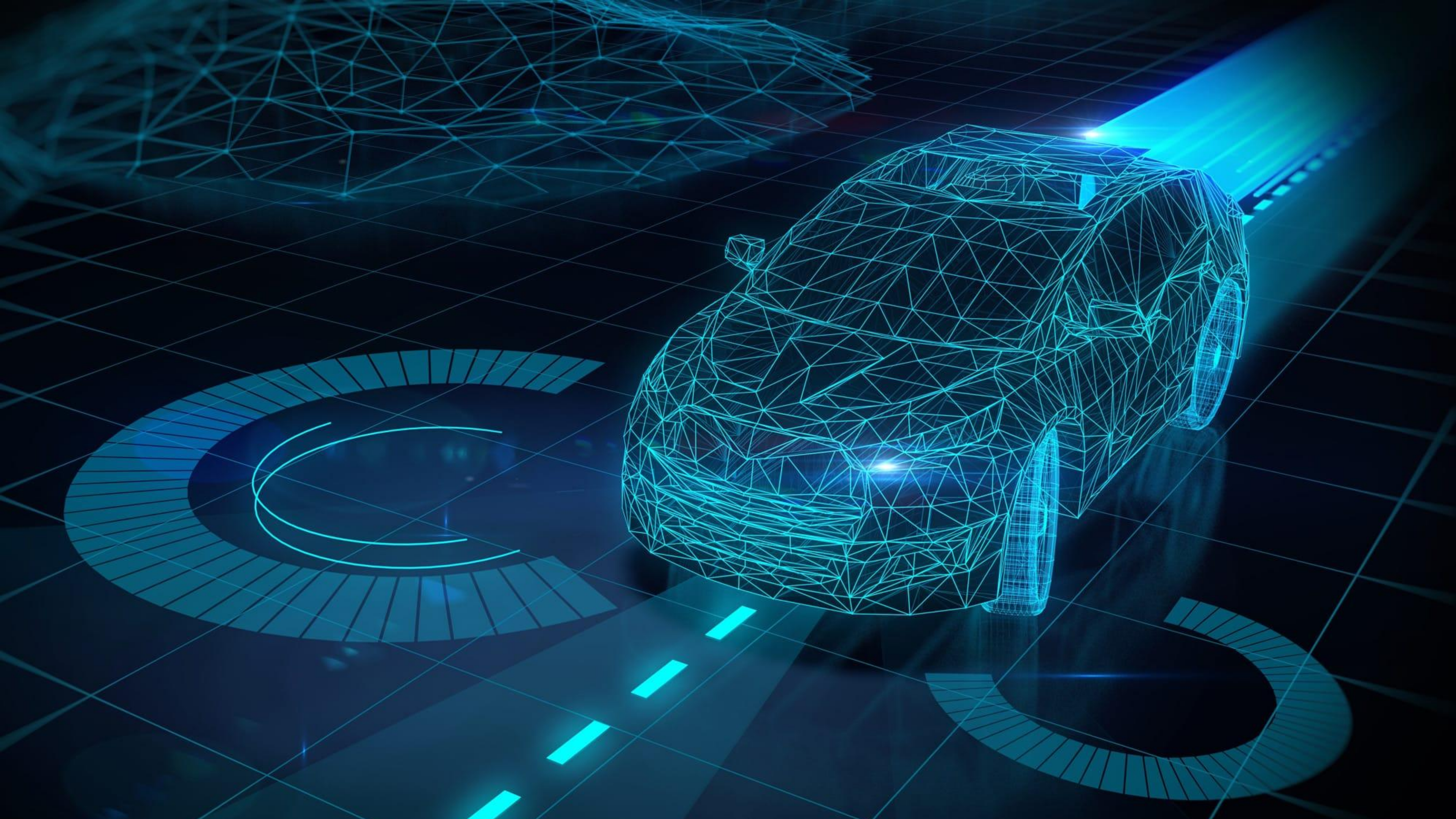
POLITECNICO MILANO 1863

HONOURS PROGRAMME HP-SR in Information Technology

Outline

- Motivation
- Setting
- Proposed solution

- **Motivation**
- Setting
- Proposed solution



Reinforcement learning (RL)

An **agent** acts in an **environment** in order to maximize a **reward signal**.

The problem is usually formalized as a Markov Decision Process:

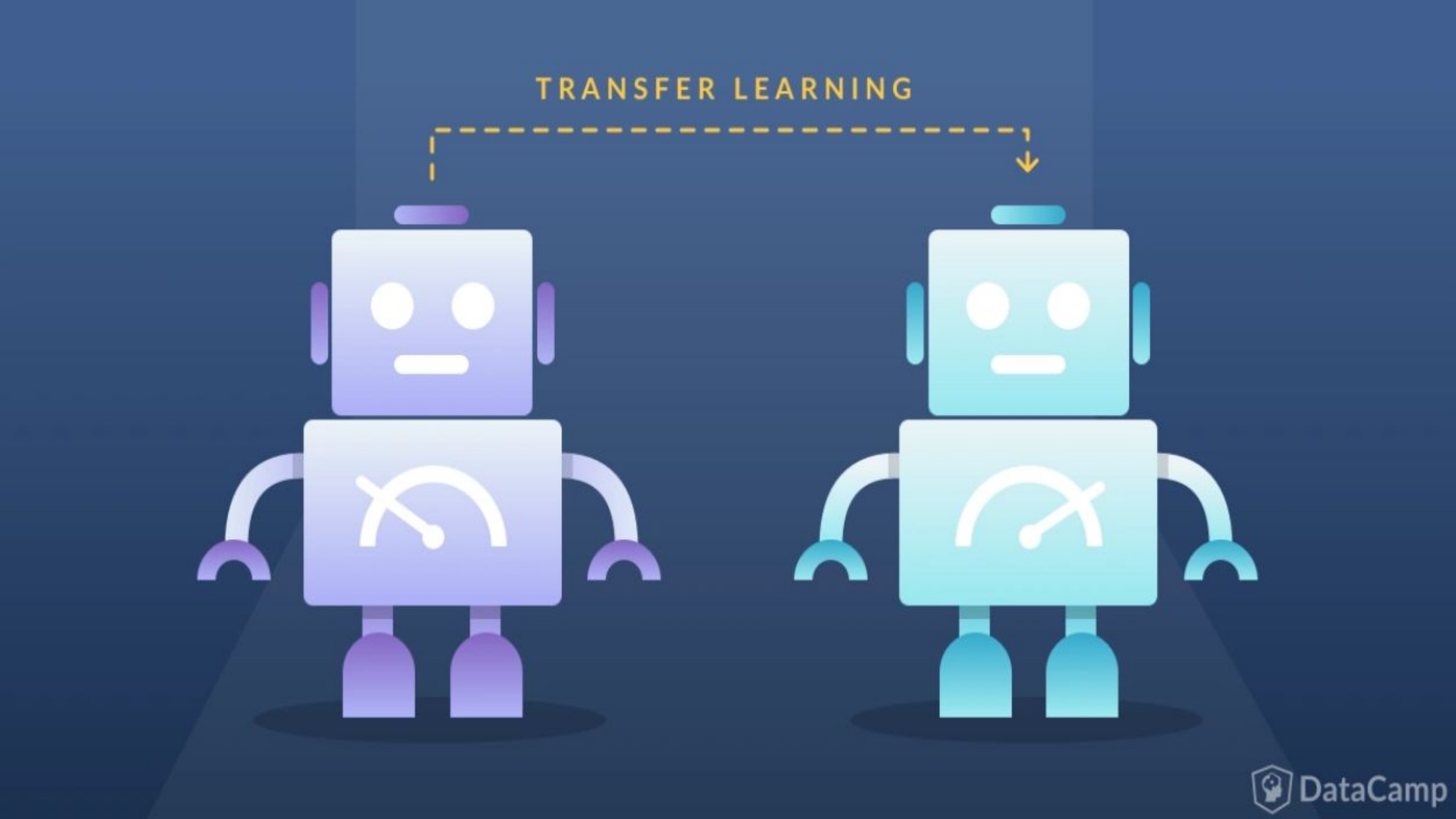
- States: S
- Actions: A
- Initial state distributions
- Reward function
- Transition distribution
- Discount factor:
  It encodes information about horizon

RL: goal

- A **policy** is a distribution over the actions, given the state
- The goal is to learn an **optimal policy** (up to some required accuracy)

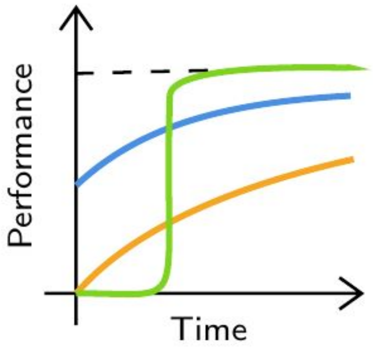
Problems and challanges

- **Superhuman** achievements in some problems but…
- Training costs **money**
- Training is **slow**
- Training can be **dangerous**
- **Poor generalization!**


TRANSFER LEARNING
DataCamp

Transfer: different approaches

- Learning from scratch
- Jumpstart
  [Mann and Choe 2012, Abel et al. 2018]
- Identification
  [Brunskill and Li 2013, Liu et al. 2016]



Most existing algorithms for task identification do not actively search for discriminative information
[Dyagilev et al. 2008, Brunskill and Li 2013, Azar et al. 2013, Liu et al. 2016]

Sequential Transfer

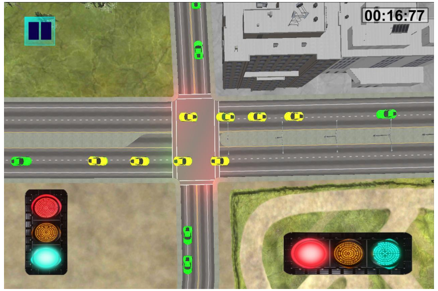
- Many real-world problems present **evolves** in a **structured** way

- This non-stationarity is usually neglected in transfer literature

Sequential Transfer

- Many real-world problems present **evolves** in a **structured** way
- This non-stationarity is usually neglected in transfer literature

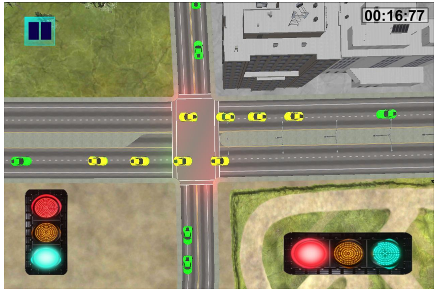
Sequential Transfer



- Many real-world problems present **evolves** in a **structured** way

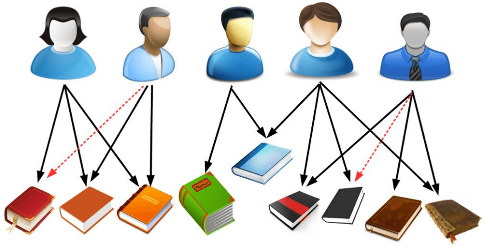
- This non-stationarity is usually neglected in transfer literature

Sequential Transfer

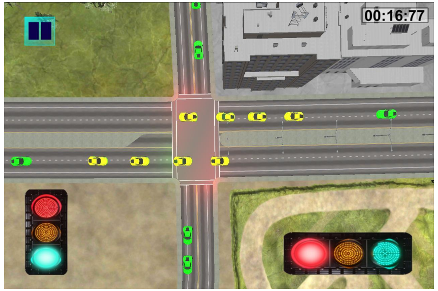


- Many real-world problems present **evolves** in a **structured** way

- This non-stationarity is usually neglected in transfer literature

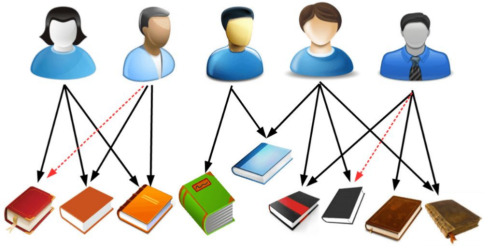


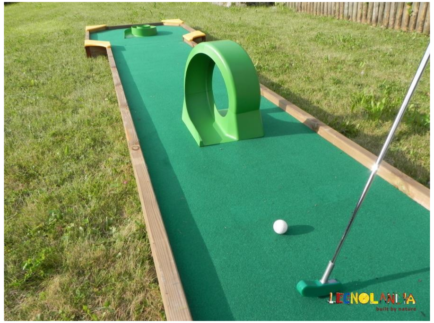
Sequential Transfer



- Many real-world problems present **evolves** in a **structured** way

- This non-stationarity is usually neglected in transfer literature





Key questions

- How to design an algorithm that **actively** identify the target task given prior knowledge?

- How to **exploit** the sequential nature of the problem?

- Motivation
- **Setting**
- Proposed solution

Sequential Transfer: Setting

- **Hidden-mode MDP** [Choi et al. 2000]
  - Agent interacts with a sequence of unknown tasks $\mathscr{M}_\theta = \{S, A, p_\theta, r_\theta, \gamma\}$
  - Finite set of possible MDP models $\Theta = \{\theta_1, \ldots, \theta_m\}$
  - Task evolve according to a Markov chain

- **Generative Model**

- **Informed** task arrival

  - The agent performs at most *n* query to the oracle (piecewise stationarity)

  - Goal: **identify** an ε-optimal policy

Sequential Transfer: Interaction

1. Extrapolate knowledge from the task evolution

2. Use this knowledge as a prior in the current task to quickly identify a good policy

3. Refine the knowledge that we have so that a more accurate prior will be available at the next iteration

- Motivation
- Setting
- **Proposed solution**

Policy Identification

- Input
  - Estimates of models in Θ
  - Δ maximum error on model estimates
  - Accuracy ε
  - Confidence δ
  - Number of samples n

- Output
  - ε-optimal policy with probability $1 - \delta$

Policy Identification

- Assume for the moment that Δ=0
- **Main idea**: not all the state-action pair are equally informative
- Example: if all models provide **nearly-deterministic** and **highly diverse** in (S,A) very few samples will be required to identify the correct model

Policy Identification

The algorithm

1. Check **transfer condition**

Policy Identification

The algorithm

1. Check transfer condition

2. for t=1…n:
   a. Build **empirical MDP**

Policy Identification

The algorithm

1. Check transfer condition

2. for t=1...n:
    a. Build empirical MDP
    b. **Update** set of plausible models

Policy Identification

The algorithm

1. Check transfer condition

2. for t=1…n:
    a. Build empirical MDP

    b. Update set of plausible models

    c. Check **stopping condition**

Policy Identification

The algorithm

1. Check transfer condition

2. for t=1…n:
    a. Build empirical MDP

    b. Update set of plausible models

    c. Check stopping condition

    d. **Query** Generative Model

Policy Identification

How to query the Generative Model to **maximize information**?

$$\mathcal{I}^r_{s,a}(\theta, \theta') = \min\left\{ \left(\frac{\widetilde{\Delta}^r_{s,a}}{\widetilde{\sigma}^r_\theta(s,a)} \right)^2, \widetilde{\Delta}^r_{s,a} \right\},$$

Policy Identification

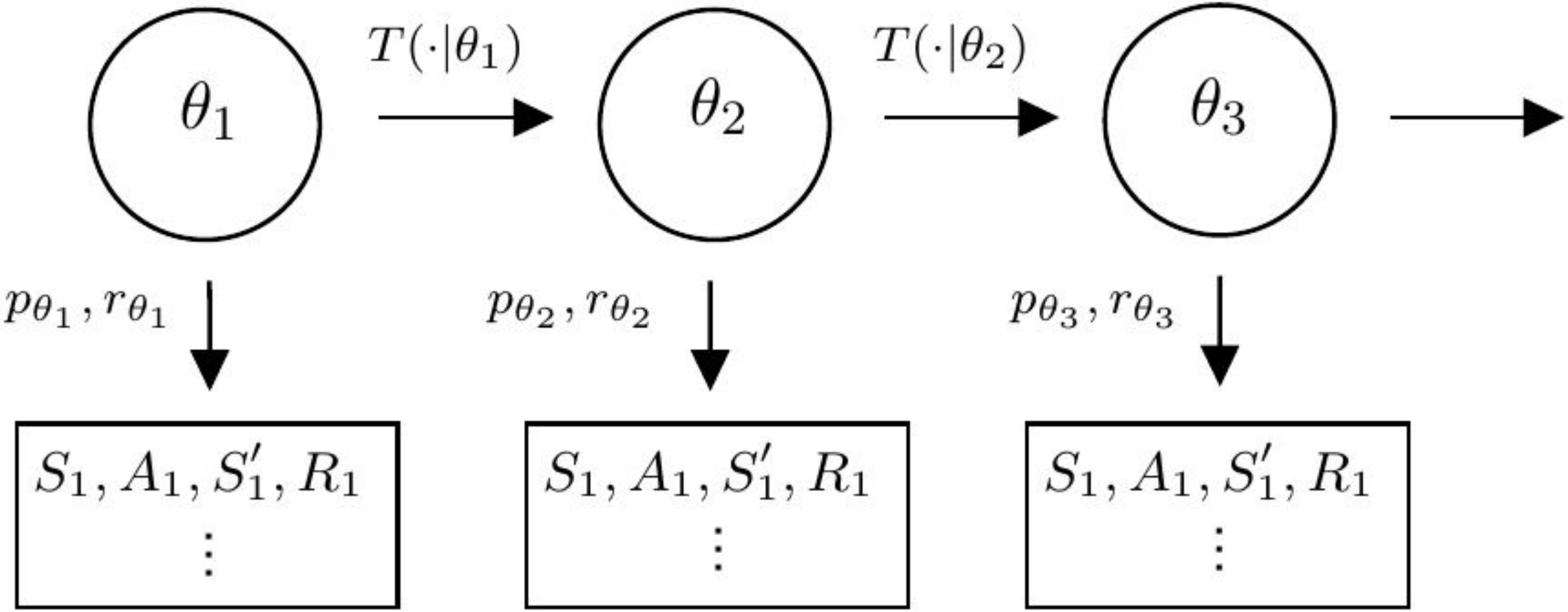
Main result: Stopping time to identify an ε-optimal policy w.p. 1-δ

$$\tau \leqslant \frac{128 \min\{SA, |\Theta|\} \log(8SAn(|\Theta|+1)/\delta)}{\max_{s,a} \min_{\theta \in \Theta_\epsilon} \mathcal{I}_{s,a}(\theta^*, \theta)}$$

Sequential Transfer

- **True task as the hidden state** of an Hidden Markov Model (HMM)
- We interact with the Generative Model to retrieve information on the true task
- Learn HMM via **tensor decomposition** [Anandkumar et al. 2014]

Sequential Transfer



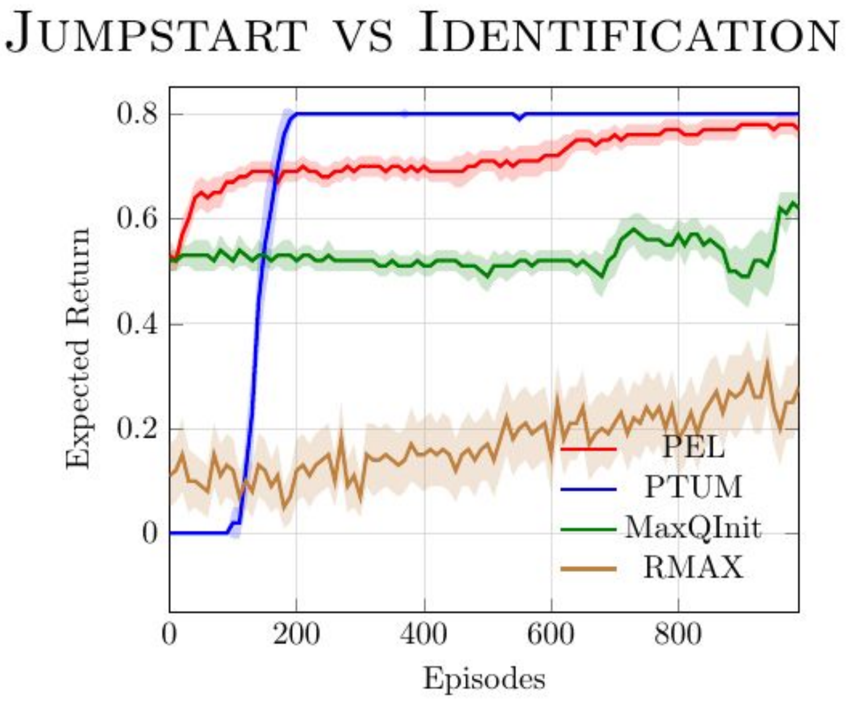
Sequential Transfer

Main results:

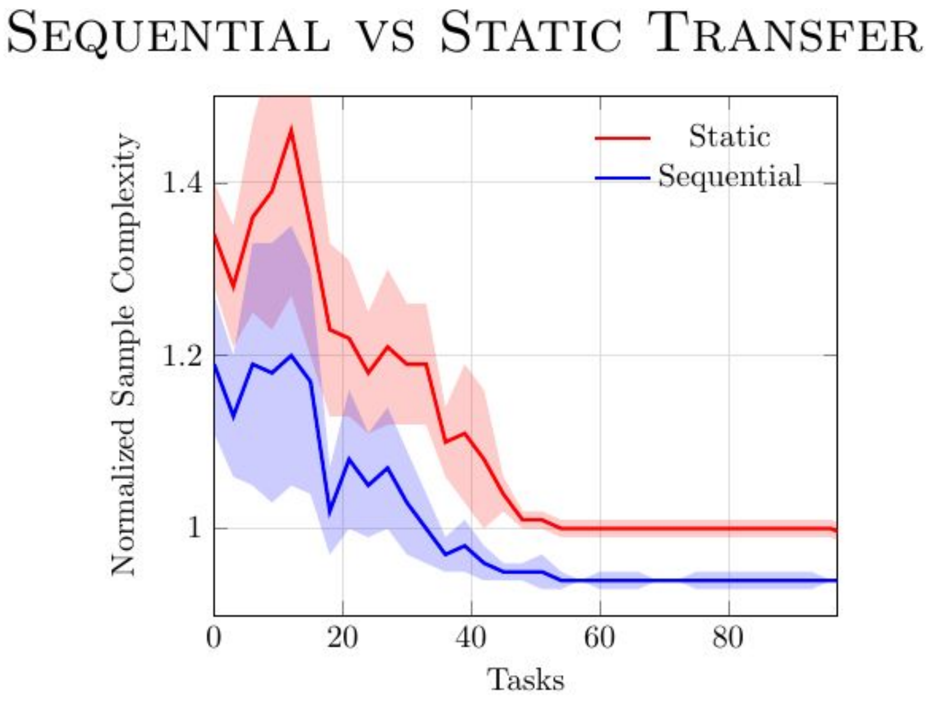
- Error estimates converges to 0 with rate $\sqrt{\frac{1}{k}}$

- Given the estimate of T, we can discard unlikely models prior to run our policy identification algorithm

Experiments



Experiments



Conclusions

- **Actively search for information** can lead to strong theoretical guarantees and better performances w.r.t. jumpstart methods

- **Exploiting temporal correlations** provides strong theoretical guarantees and performance boosts

Thanks for your attention!