

A Survey on Transfer Learning in RL & Robotics

Juan Camilo Gamboa Higuera

Transfer Club Week 1

2018-08-26

Transfer Learning for Reinforcement Learning Domains: A Survey

Matthew E. Taylor*

TAYLORM@USC.EDU

*Computer Science Department
The University of Southern California
Los Angeles, CA 90089-0781*

Peter Stone

PSTONE@CS.UTEXAS.EDU

*Department of Computer Sciences
The University of Texas at Austin
Austin, Texas 78712-1188*

Editor: Sridhar Mahadevan

Abstract

The reinforcement learning paradigm is a popular way to address problems that have only limited environmental feedback, rather than correctly labeled examples, as is common in other machine learning contexts. While significant progress has been made to improve learning in a single task, the idea of *transfer learning* has only recently been applied to reinforcement learning tasks. The core idea of transfer is that experience gained in learning to perform one task can help improve learning performance in a related, but different, task. In this article we present a framework that classifies transfer learning methods in terms of their capabilities and goals, and then use it to survey the existing literature, as well as to suggest future directions for transfer learning work.

Keywords: transfer learning, reinforcement learning, multi-task learning

What is Transfer Learning?

A habit by which we always transfer the known to the unknown and conceive the latter to resemble the former
(Hume, 1689)

The ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks in new domains
(Pan and Yang, 2010)

Generalization across tasks
(Taylor and Stone, 2009)

*Isn't that the same as multi-task learning?
meta-learning?
learning?*

Main difference

- *Source tasks* may come from **different distribution** than *target tasks*
 - *Source tasks/domains*: cheap data or sunk cost
 - *Target tasks/domains*: unknown at training time
- We want generalization to target tasks
 - Few-shot/one-shot/zero-shot learning

Tasks in the RL setting

- An MDP is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathbb{P}(\cdot | s, a), \mathbb{P}_1, r \rangle$
 - \mathcal{S} denotes the set of state
 - \mathcal{A} the set of actions
 - $\mathbb{P}(\cdot | s, a)$ the probability distribution over states, given that action a was taken at state s (transition dynamics)
 - \mathbb{P}_1 the initial state distribution
 - $r: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ the task dependent reward function
- A POMDP $\langle \mathcal{S}, \mathcal{A}, \Omega, \mathbb{P}(\cdot | s, a), \mathbb{O}(\cdot | s), \mathbb{P}_1, r \rangle$
 - Ω is the set of observations
 - $\mathbb{O}(\cdot | s)$ the probability distribution over observations, given a state s

Objective in the RL setting

- Find a *policy* that maximizes the *expected cumulative reward*
 - Deterministic policy $\pi: \mathcal{S} \rightarrow \mathcal{A}$
 - Stochastic policy $\pi: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{P}(\mathcal{A})$
- Value of a policy

$$V^\pi(s) = \mathbb{E}(\sum_{t=1}^H \gamma_t r(S_t, A_t) \mid \pi, S_1 = s)$$

$$Q^\pi(s, a) = \mathbb{E}(\sum_{t=1}^H \gamma_t r(S_t, A_t) \mid \pi, S_1 = s, A_1 = a)$$

- Trajectory $\tau: s_1, a_1, r_1, \dots, s_H, a_H, r_H$
- γ_t is a discount factor when H is ∞
 - May be $1/H$ for finite H

Some transfer problem settings

- Sim2real transfer
 - Source $\langle \mathcal{S}, \mathcal{A}, \Omega, \mathbb{P}_{\text{sim}}(\cdot | s, a), \mathbb{O}_{\text{sim}}(\cdot | s), \mathbb{P}_1, r \rangle$
 - Target $\langle \mathcal{S}, \mathcal{A}, \Omega, \mathbb{P}_{\text{real}}(\cdot | s, a), \mathbb{O}_{\text{real}}(\cdot | s), \mathbb{P}_1, r \rangle$
- Multi-task transfer
 - Source $\langle \mathcal{S}, \mathcal{A}, \mathbb{P}(\cdot | s, a), \mathbb{P}_1, r \rangle_{i=1}^N$
 - Target $\langle \mathcal{S}, \mathcal{A}, \mathbb{P}(\cdot | s, a), \mathbb{P}_1, r \rangle_{N+1}$
- Sequential multi-task learning (reward shaping? Curriculum learning?)
 - Source $\langle \mathcal{S}, \mathcal{A}, \mathbb{P}(\cdot | s, a), \mathbb{P}_1, r \rangle_i$
 - Target $\langle \mathcal{S}, \mathcal{A}, \mathbb{P}(\cdot | s, a), \mathbb{P}_1, r \rangle_{i+1}$
- Related: Meta-learning
 - Source and target come from different subsets of task distribution $p(M_i)$

How do these fit under the taxonomy by Taylor and Stone?

Back to the survey by Taylor and Stone

- They provide a taxonomy of transfer problems and methods along five dimensions
 1. Task difference assumptions
 2. Source task selection
 3. Transferred knowledge
 4. Task Mappings
 5. Allowed Learners
- Disclaimer: This is from before the DL boom

Citation	Allowed Task Differences	Source Task Selection	Task Mappings	Transferred Knowledge	Allowed Learners	TL Metrics
Same state variables and actions: Section 4						
Selfridge et al. (1985)	t	h	N/A	Q	TD	tt [†]
Asada et al. (1994)	s_i	h	N/A	Q	TD	tt
Singh (1992)	r	all	N/A	Q	TD	ap, tr
Atkeson and Santamaria (1997)	r	all	N/A	model	MB	ap, j, tr
Asadi and Huber (2007)	r	h	N/A	π_p	H	tt
Andre and Russell (2002)	r, s	h	N/A	π_p	H	tr
Ravindran and Barto (2003b)	s, t	h	N/A	π_p	TD	tr
Ferguson and Mahadevan (2006)	r, s	h	N/A	pvf	Batch	tt
Sherstov and Stone (2005)	s_f, t	mod	N/A	A	TD	tr
Madden and Howley (2004)	s, t	all	N/A	rule	TD	tt, tr
Lazaric (2008)	s, t	lib	N/A	I	Batch	j, tr
Multi-Task learning: Section 5						
Mehta et al. (2008)	r	lib	N/A	π_p	H	tr
Perkins and Precup (1999)	t	all	N/A	π_p	TD	tt
Foster and Dayan (2004)	s_f	all	N/A	sub	TD, H	j, tr
Fernandez and Veloso (2006)	s_i, s_f	lib	N/A	π	TD	tr
Tanaka and Yamamura (2003)	t	all	N/A	Q	TD	j, tr
Sunmola and Wyatt (2006)	t	all	N/A	pri	B	j, tr
Wilson et al. (2007)	r, s_f	all	N/A	pri	B	j, tr
Walsh et al. (2006)	r, s	all	N/A	fea	any	tt
Lazaric (2008)*	r	all	N/A	fea	Batch	ap, tr

Citation	Allowed Task Differences	Source Task Selection	Task Mappings	Transferred Knowledge	Allowed Learners	TL Metrics
Different state variables and actions – no explicit task mappings: Section 6						
Konidaris and Barto (2006)	p	h	N/A	R	TD	j, tr
Konidaris and Barto (2007)	p	h	N/A	π_p	TD	j, tr
Banerjee and Stone (2007)	a, v	h	N/A	fea	TD	ap, j, tr
Guestrin et al. (2003)	#	h	N/A	Q	LP	j
Croonenborghs et al. (2007)	#	h	N/A	π_p	RRL	ap, j, tr
Ramon et al. (2007)	#	h	N/A	Q	RRL	ap, j, tt [†] , tr
Sharma et al. (2007)	#	h	N/A	Q	TD, CBR	j, tr
Different state variables and actions – inter-task mappings used: Section 7						
Taylor et al. (2007a)	a, v	h	sup	Q	TD	tt [†]
Taylor et al. (2007b)	a, v	h	sup	π	PS	tt [†]
Taylor et al. (2008b)	a, v	h	sup	I	MB	ap, tr
Torrey et al. (2005)	a, r, v	h	sup	rule	TD	j, tr
Torrey et al. (2006)						
Torrey et al. (2007)	a, r, v	h	sup	π_p	TD	j, tr
Taylor and Stone (2007b)	a, r, v	h	sup	rule	any/TD	j, tt [†] , tr
Learning inter-task mappings: Section 8						
Kuhlmann and Stone (2007)	a, v	h	T	Q	TD	j, tr
Liu and Stone (2006)	a, v	h	T	N/A	all	N/A
Soni and Singh (2006)	a, v	h	M_a, sv_g, exp	N/A	all	ap, j, tr
Talvitie and Singh (2007)	a, v	h	M_a, sv_g, exp	N/A	all	j
Taylor et al. (2007b)*	a, v	h	sv_g, exp	N/A	all	tt [†]
Taylor et al. (2008c)	a, v	h	exp	N/A	all	j, tr

Task differences

- They refer to changes to any element of the MDP

$$M_{\text{source}} = \left\langle \mathcal{S}^{(\text{source})}, \mathcal{A}^{(\text{source})}, \mathbb{P}^{(\text{source})}(\cdot | s, a), \mathbb{P}_1^{(\text{source})}, r^{(\text{source})} \right\rangle$$

$$M_{\text{target}} = \left\langle \mathcal{S}^{(\text{target})}, \mathcal{A}^{(\text{target})}, \mathbb{P}^{(\text{target})}(\cdot | s, a), \mathbb{P}_1^{(\text{target})}, r^{(\text{target})} \right\rangle$$


- Questionable categories in the survey?
 - changes in goal states (s_f) vs. changes in rewards (r)
 - changes in state (s) vs. changes in variables (v)

Source Task Selection

- Fixed source set
 - Selected by designer
 - Single source task (h)
 - Multiple source tasks (all)
- Control over source set
 - Algorithm selects subset of available source tasks (lib)
 - Algorithm modifies source task (mod)
- Missing from their taxonomy?
 - Interactive source task selection (LfD, active learning)
 - Modifying the source task distribution (Chebotar, 2018)
 - ...

How do we transfer knowledge?

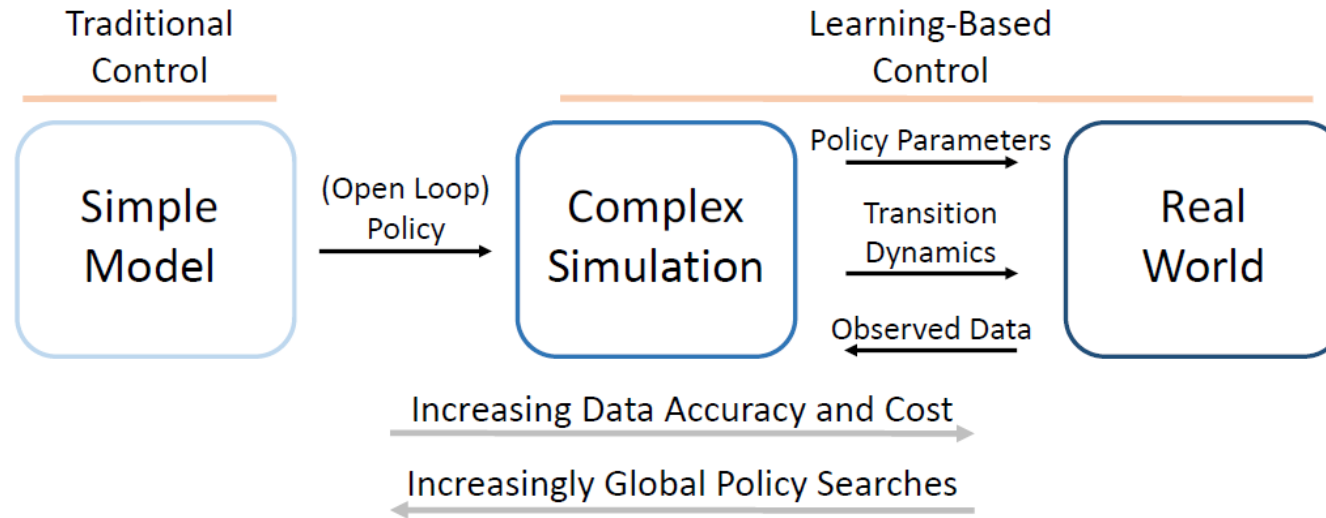
- Experience data $\mathcal{D} = \langle \tau_1, \dots, \tau_T \rangle$
- Models for the dynamics $\mathbb{P}(\cdot | s, a)$
 - Prior belief on how the world works
- Value function estimates $\hat{V}^\pi(s), \hat{Q}^\pi(s, a)$
 - Prior belief on good states or actions
- Policies π
 - Starting point for searching for good actions
 - Higher/lower level policies in a hierarchy (e.g. options)
- Features/latent representations $\phi(s)$
 - Abstractions that may be useful across tasks (embeddings, hidden layers, encoders)



Harder
to
obtain?

Transferring transition dynamics and policies

- Model-based RL viewed as transfer (Cutler and How, 2016)



- Guided policy search would fit here?

Transferring Invariant Knowledge

- Separation between *agent-space* and *problem-space* (Konidaris and Barto, 2007)
 - Agent space: capabilities ($\mathcal{S}, \mathcal{A}, \Omega$)
 - Problem space: things out of agent control ($\mathbb{P}(\cdot, s, a), \mathbb{P}_1, r$)
 - Assume invariance in one space for transfer in the other

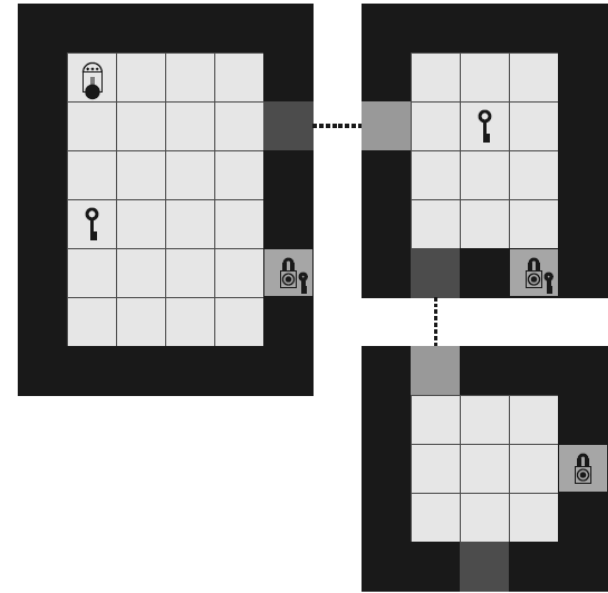


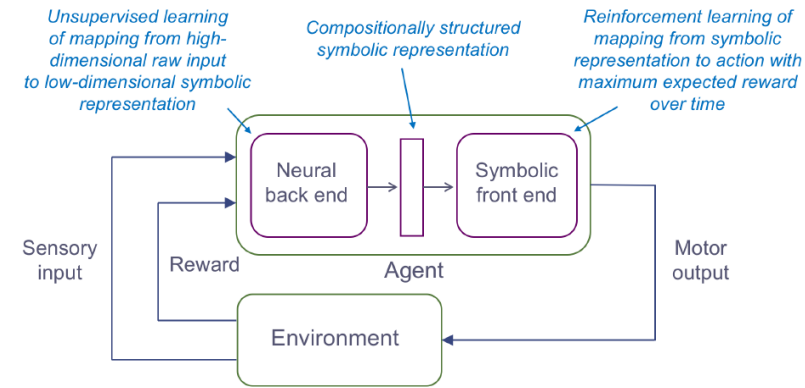
Figure 1: A small example lightworld.

For every particular lightworld instance, a problem-space descriptor requires five pieces of data: the current room number, the x and y coordinates of the robot in that room, whether or not the robot has the key, and whether or not the door is open. We use the light sensor readings as an agent-space because their semantics remain consistent across lightworld instances. In this case the agent-space (with 12 continuous variables) has higher dimension than any of the individual problem-spaces.

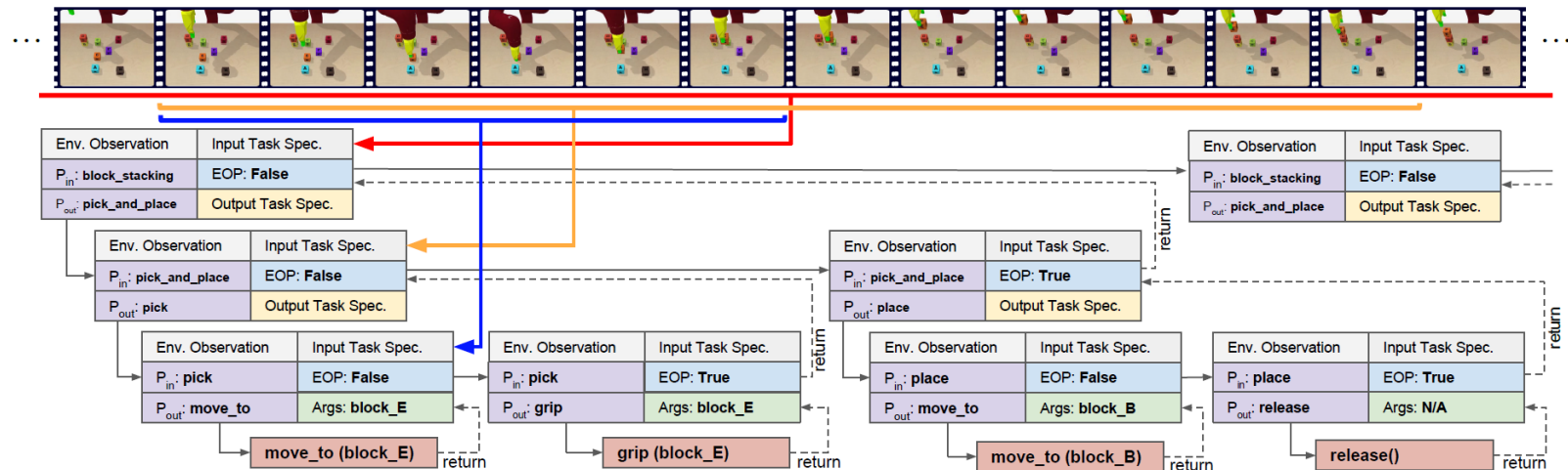
Transferring Invariant Knowledge

- Relational RL: Combining RL with symbolic reasoning

(Garnelo *et al*, 2016)



(Xu *et al*, 2016)



Transferring Invariant Knowledge

- Graph NNs for learning a physics engine
(Sanchez-Gonzalez *et al*, 2018)

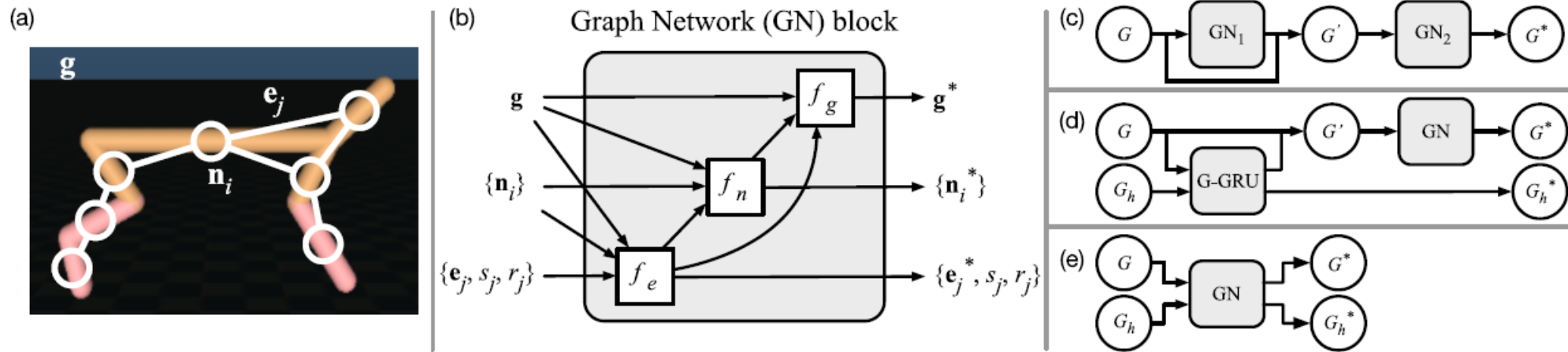
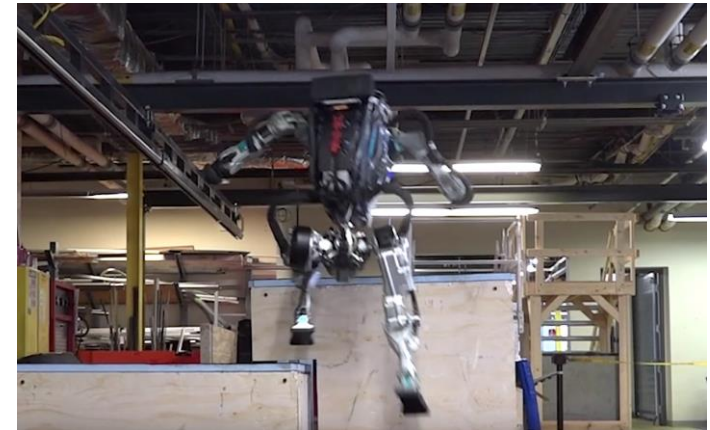
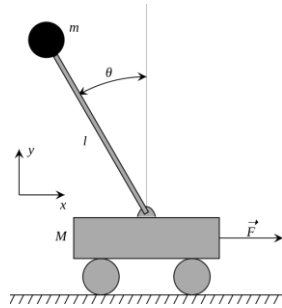


Figure 2. Graph representations and GN-based models. (a) A physical system's bodies and joints can be represented by a graph's nodes and edges, respectively. (b) A GN block takes a graph as input and returns a graph with the same structure but different edge, node, and global features as output (see Algorithm 1). (c) A feed-forward GN-based forward model for learning one-step predictions. (d) A recurrent GN-based forward model. (e) A recurrent GN-based inference model for system identification.

Task Mappings

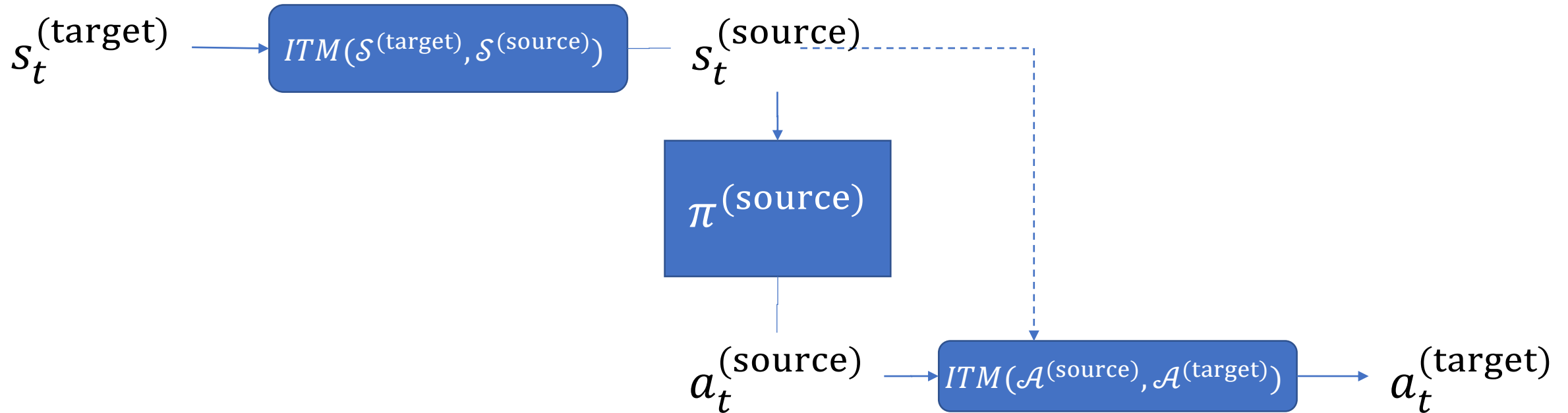
- When varying \mathcal{S} and \mathcal{A} from source to target
 - Number dimensions may change
 - Dimension indices may change
 - Meaning of dimensions may change



- Are the mappings are known or need to be discovered/inferred/learned

Task Mappings

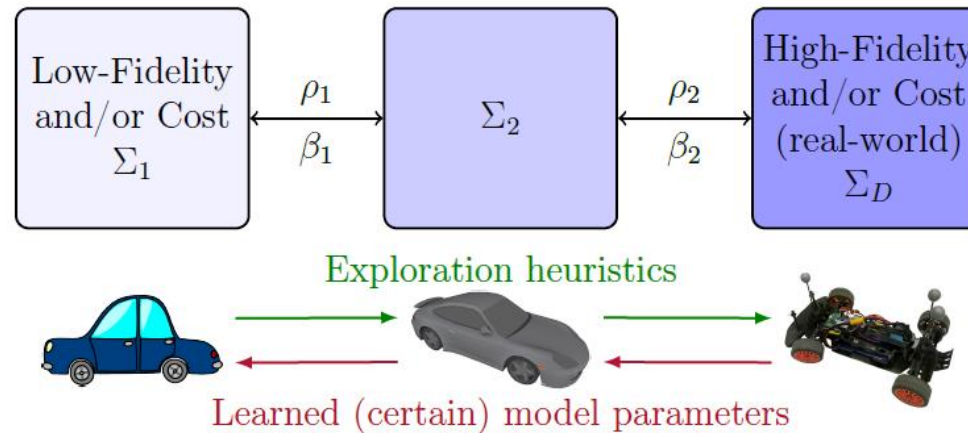
- Taylor and Stone refer to their *inter-task mapping* work (Taylor et al, 2007)



- Progressive nets (Rusu et al, 2017) has some similarities, but mappings are at hidden feature level
- We explored a similar idea with policy adjustments (Gamboa-Higuera et al, 2017)

Transferring value and transition models

- Multi-fidelity Learning (Cutler and How, 2014)



- Exploration: Transfer Q from low to high fidelity (adapted with optimism bounds)
- Lower fidelity improvement: Transfer transition dynamics model parameters from high to low fidelity
- Decision on when to transfer based on KWIK learning bounds

Allowed Learners

- Categorization based on the learning algorithm used in the target tasks
 - Temporal differences
 - Model-based
 - Policy Search
 - Imitation Learning
 - ...
- Not sure this is an useful categorization
 - But you may disagree

How should we compare TL methods

- The survey proposes the following metrics

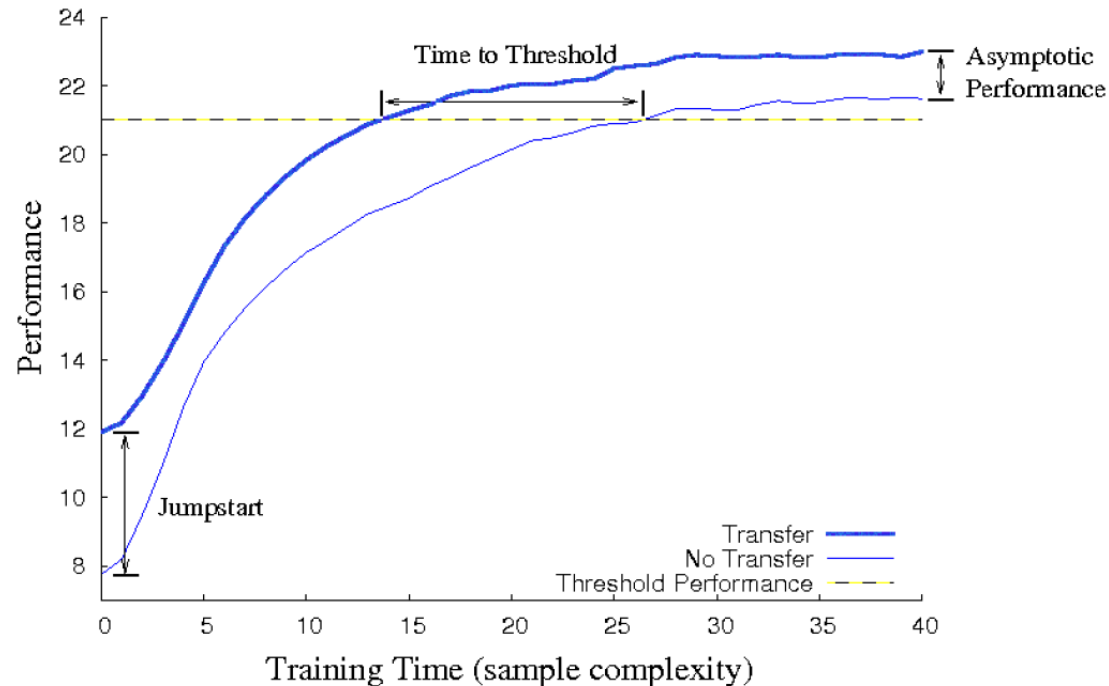
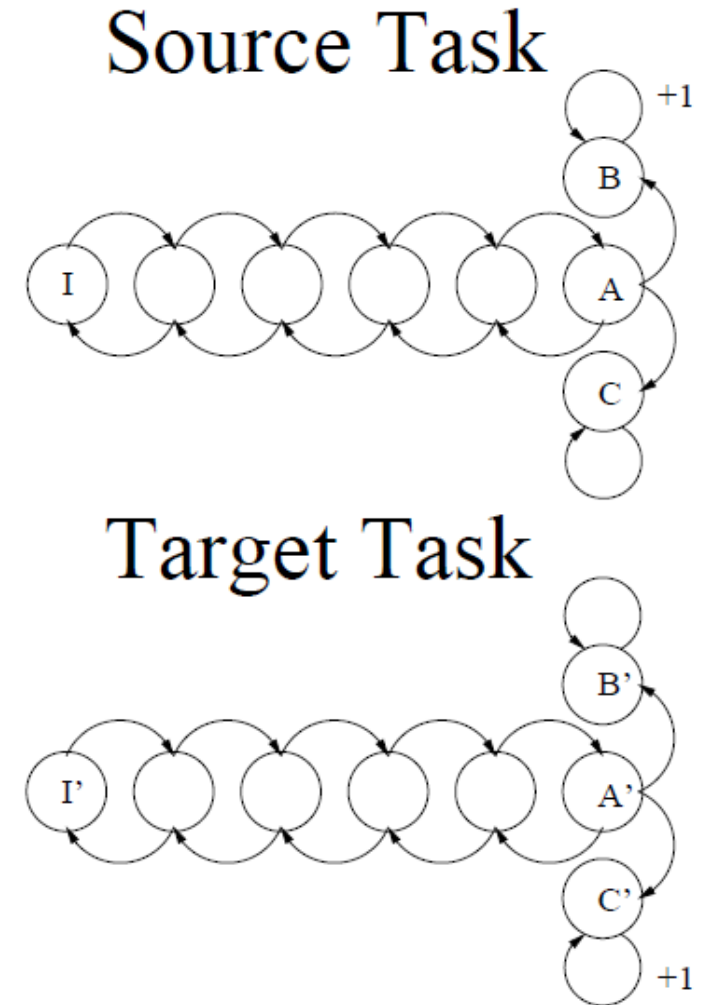


Figure 3: Many different metrics for measuring TL are possible. This graph show benefits to the jumpstart, asymptotic performance, time to threshold, and total reward (the area under the learning curve).

- Not sure if recent work follows this

Negative Transfer

- Transferring optimal policy/value may result in negative transfer
 - in all proposed metrics!
 - but there is some similarity in structure that could be exploited
- Can you predict whether an algorithm results in negative/positive transfer?
 - Bisimulation may help (Ferns et al, 2006)
- What is negative transfer anyway?



Bisimulation (Ferns et al, 2006)

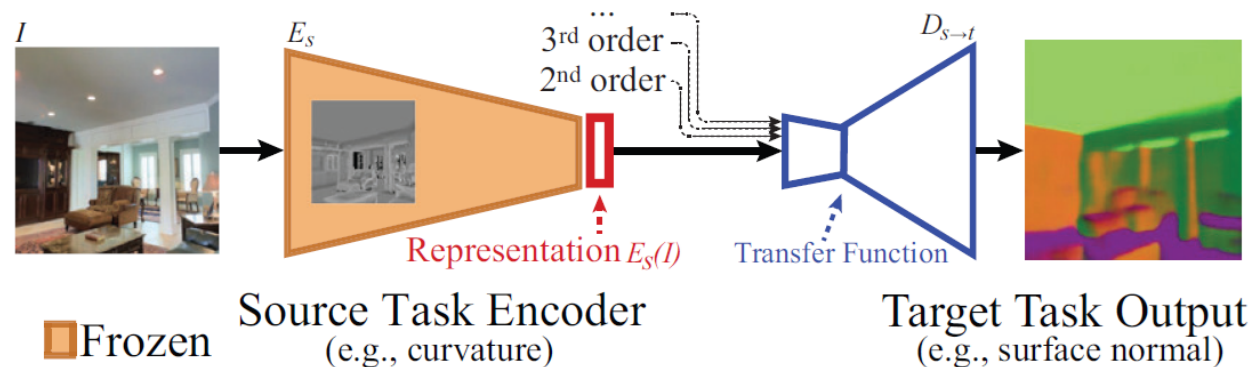
In the context of MDPs, bisimulation can roughly be described as the largest equivalence relation on the state space of an MDP that relates two states precisely when for every action, they achieve the same immediate reward and have the same probability of transitioning to classes of equivalent states. This means that bisimilar states lead to essentially the same long-term behavior.

- In follow-up work, the authors have proposed tractable methods for computing similarities between MDPs
- How can we use these ideas for transfer?

Taskonomy

Builds graph of transferability between computer vision tasks:

1. Collect dataset of 4 million input images and labels for 26 vision tasks
 - a. Surface normal, Depth estimation, Segmentation, 2D Keypoints, 3D pose estimation
2. Train convolutional autoencoder architecture for each tasks



<http://taskonomy.stanford.edu/>

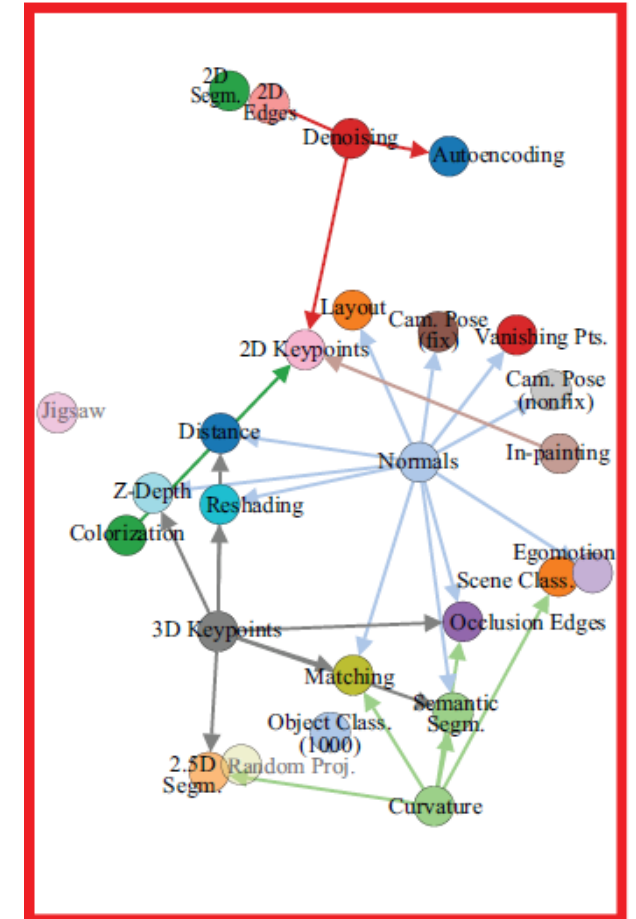
Taskonomy

Builds graph of transferability between computer vision tasks:

3. Transferability obtained by Analytic Hierarchy Process (from pairwise comparisons between all possible sources for each target task)
4. Final graph obtained by subgraph selection optimization (best performance from a limited set of source tasks): *transfer policy*

Empirical study on performance and data-efficiency gains from transfer using different datasets (Places and Imagenet)

Supervision Budget 8 - Order 4 (zoomed)



Transferability

Can we come up with a transferability metric?

- Based on bisimulation?
- Empirical performance based metrics like tasknomony?
- How would this metric be used?
 - Predicting whether transfer is possible or not
 - As an optimization objective?