AFRL-RI-RS-TR-2010-070
Final Technical Report
March 2010

EFFECTIVE BAYESIAN TRANSFER LEARNING

University of California, Berkeley

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. AO/V102

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2010-070 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/	/s/
ROGER J. DZIEGIEL, Jr.	JOSEPH CAMERA, Chief
Work Unit Manager	Information & Intelligence Exploitation Division
	Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
MARCH 2010

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
September 2005 – September 2009

4. TITLE AND SUBTITLE
EFFECTIVE BAYESIAN TRANSFER LEARNING

5a. CONTRACT NUMBER
N/A

5b. GRANT NUMBER
FA8750-05-2-0249

5c. PROGRAM ELEMENT NUMBER
62304E

6. AUTHOR(S)
Tomas Lozano-Perez and Leslie Kaelbling

5d. PROJECT NUMBER
TRLG

5e. TASK NUMBER
00

5f. WORK UNIT NUMBER
01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The Regents of the University of California
c/o Sponsored Projects Office
2150 Shattuck Avenue, Suite 313
Berkeley, CA 94704-5940

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, VA 22203-1714

AFRL/RIEH
525 Brooks Road
Rome, NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
N/A

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2010-070

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 88ABW-2010-0943

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The general theme of the project is transfer learning, i.e., the process whereby the learning process in task Y is improved by prior learning experience in task X. The project addresses transfer learning in three application areas: strategy games, robotic object manipulation, and visual object recognition.

15. SUBJECT TERMS
Machine Learning, Transfer Learning, Artificial Intelligence, Strategy Games, Robotic Manipulation, Computer Vision

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Roger J. Dziegiel, Jr.
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (*Include area code*)
U	U	U	UU	92	N/A

Table of Contents

Mission

The Transfer Learning program seeks to solve the problem of reusing knowledge derived in one domain to help effect solutions in another domain. Adaptive systems, systems that respond to changes in their environment, stand to benefit significantly from the application of TL technology. Today's adaptive systems need to be trained for every new situation they encounter. This requires building new training data, which is the most expensive and most limiting aspect of deploying such systems. The TL program addresses this shortcoming by imbuing adaptive systems with the ability to encapsulate what they have learned and apply this knowledge to new situations. Thus, rather than having to be retrained for each new context, TL enables systems to leverage what they have already learned in order to be effective much sooner and with less effort spent on training. Early applications of TL technology include adaptive ISR systems, robotic vision and manipulation, and automated population of databases from unstructured text.

Goals

The general theme of the project is transfer learning, i.e., the process whereby the learning process in task Y is improved by prior learning experience in task X. The project addresses transfer learning in three application areas: strategy games, robotic object manipulation, and visual object recognition.
Existing machine learning methods assume that the training data is drawn from the same distribution as the task they are learning; they do not recognize and apply knowledge and skills learned in previous tasks to novel tasks in new domains. The result is excessive need for either human time or expensive training data.
The primary goal of the research has been to develop a general theory of transfer learning and effective instantiations thereof for perception, planning, and action. Effective transfer requires strong prior knowledge, hence a major subgoal is to develop forms of prior knowledge that express strong, high-level, cross-task and cross-domain regularities, as well as methods for their use in transfer and their acquisition by learning. Well-founded transfer learning, i.e., learning that can be shown to work well, requires development of a unified theoretical framework (encompassing prior knowledge, observations, actions, rewards, etc.) that supports mathematical results on learning capacity and limitations. Finally, we aim to develop reproducible domains and task families of sufficient richness to support substantial transfer learning.
Cumulative, knowledge-intensive Bayesian learning enable much faster learning of much richer models from much less data, and rapid adaptation of persistent autonomous agents to new circumstances without extensive reprogramming or retraining. Furthermore, we have seen specific gains in the form of more effective systems for visual perception and manipulation.

Go/NoGo and Scientific Summaries

Graphical summaries of the scientific results for each year of the program, including detailed results of the Go/NoGo tests for each year are attached as Appendices, one for each year.

Selected Accomplishments

Task R1: Hierarchical Bayes

Michael Jordan, UC Berkeley, developed a new approach to feature selection based on block L1 norms. His group found that dual extra-gradient algorithms provide a stable, robust numerical platform for this approach. The algorithm has been tested on standard machine learning benchmarks, including handwritten character recognition (where the multi-task aspect arises from the multiple writers). Testing on these benchmarks has been essential---it allowed them to be able to judge the performance, scaling and robustness of the algorithm relative to accumulative wisdom of the literature.

Developed a fully Bayesian hierarchical model for feature selection which uses separate hierarchical pathways for feature relevance and feature values. Thus a feature may transfer if it is relevant for a task, even if the parameter value has a different sign across tasks. The model uses Dirichlet process priors to permit clustering of feature values.

Developed a new algorithm known as “ebb-flow” for inference in (hierarchical) Dirichlet process mixtures (aka, infinite tied mixture models). Jordan's group carried out experiments to compare the new approach to standard Gibbs sampling and split-merge algorithms.

Developed a new algorithm for finding common subspaces for multi-task regression and classification problems. This problem is the counterpart of the feature selection problem. Rather than finding a set of features that are useful across multiple tasks, the algorithm finds sets of feature combinations (i.e., a subspace) that are useful across multiple tasks. Our approach is based on random projections. They choose a large number of random projections and treat these projections as features for the block L1 norm algorithm that they developed earlier. That algorithm selects subsets of projections that are useful across tasks; i.e., it selects a multi-task feature subspace.

Developed a third approach to feature selection based on block L1 norms, in addition to the dual extra-gradient and sequential optimization approaches developed in their previous work. This new method is based on the recently-developed BLasso algorithm of Peng and Yu (2006); it extends that algorithm to the block-norm setting. Jordan's group found that this approach has advantages in terms of scaling with respect to the other approaches, and it also has the advantage of being an online algorithm. Jordan's group views this approach as our main algorithmic platform for multi-task feature selection.

Developed a novel nonparametric hierarchical Bayesian framework for transferring attribute-based (i.e., featural) representations in the multi-task setting. Their earlier work on the hierarchical Dirichlet process provided a nonparametric approach to clustering in the multi-task setting. The new approach is an analogous methodology for problems in which object identity is not reduced to the cluster that it belongs to, but is encoded by a set of attributes. The learning algorithm finds attributes that are useful across multiple tasks. The approach is based on a L'evy process known as the beta process, a stochastic process in which the sample paths that encode probabilities of sparse Bernoulli matrices. Jordan's group showed how to define a "hierarchical beta process," in which these probabilities are shared across multiple Bernoulli matrices.

Developed a novel approach to inference in Dirichlet process mixtures. The approach is referred to as a "permutation-augmented sampler." Standard approaches to sampling-based inference essentially move a single data point at time. This makes it difficult for the Markov chain to mix at the level of clusters, and these algorithms can be quite slow. The new approach samples an entire permutation and then sums over all clusterings consistent with the clustering. This is done with a dynamic programming algorithm. In experiments, they have shown that this yields burn-in times that are significantly smaller than those of the Gibbs sampler.

Made progress on the problem of transfer among the states of semi-Markov models. Using the hierarchical Dirichlet process approach and hidden Markov model (HDP-HMM) developed in their earlier work, they have shown how to extend the HDP-HMM to allow separate control over self-transitions.

Developed a new hierarchical nonparametric Bayesian approach to hidden Markov modeling. Current approaches to the nonparametric hidden Markov models have been plagued by the over-abundance of switching transitions among closely-related states. Our new approach---the "tempered HMM"---solves the problem by allowing separate control over self-transitions.

Developed a new approach to transfer learning that they referred to as "agreement-based learning." This consists in a novel use of latent variable models in which multiple models are forced to agree on a set of latent variables. This provides a new approach to symbolic transfer.

Developed a new class of nonexchangeable nonparametric priors based on Markov chains. Such priors allow entities to share features if those entities are close together in time. Jordan's group has developed computationally efficient inference procedures for posterior inference under such priors. Similar nonparametric priors have been developed for other data types, including counts and rates, using Kingman's theory of completely random processes.

The focus of the research on hierarchical Bayesian transfer learning has been limited to exchangeable models. These are models in which the entities being modeled are treated as independent and identically distributed given the latent variables in the hierarchy.

While leading to tractable models this is an overly strong assumption that is ill-suited to many problems; specifically it does not allow additional covariates to be observed. They have begun to work on the “Phylogenetic Indian Buffet Process,” a nonparametric hierarchical Bayesian methodology for partially exchangeable models. They assume that the similarity among entities can be described by a tree and they develop a set of posterior update rules for the Indian buffet process that makes use of belief propagation in the tree. Despite the non-exchangeability the overall update is as tractable computationally as an exchangeable model.

Developed a new methodology for transfer in temporal domains. The methodology builds on their earlier work with the hierarchical beta process (HBP). The beta process is a nonparametric Bayesian prior that allows a system to discover sets of features that are shared among multiple groups. The new idea is to associate to each feature a dynamical system (in particular, a state-space model). When this feature is instantiated, the model produces dynamical behavior according to that state-space model. Thus, selecting a set of features corresponds to selecting a set of dynamical behaviors which can be switched in or switched out over time. The HBP allows these dynamical behaviors to be shared across groups as well as across time. Jordan's group has demonstrated that this approach can be used to segment videos of human activity (from the CMU video database), where transfer is achieved among types of activities.

Andrew Ng, Stanford, formulated a new, widely applicable learning problem in which high-level knowledge is transferred from easily available unlabeled data. This problem is called self-taught learning. His group developed algorithms for a high-level abstraction algorithm called sparse coding, that are two orders of magnitude faster than previous algorithms. Using this technical advance, they applied the sparse coding algorithm to self-taught learning, and demonstrated highly effective transfer using only unlabeled data.

Within the self-taught learning framework, they developed the first tractable algorithm for solving the shift-invariant formulation of sparse coding. This algorithm enabled them to learn succinct, higher-level transfer learning representations for audio and image data. The new algorithms were shown to outperform well-known and widely used baseline algorithms in the presence of real-world noise. They tested them on self-taught learning tasks involving image and audio classification. They packaged and released their implementation.

Developed new algorithms for learning hierarchical representations, allowing the transfer of knowledge from easily available unlabeled data to supervised tasks. These algorithms learn abstract, higher-level patterns automatically from data by piecing together several simpler patterns that were also learnt from data. Unlike previous algorithms, the learnt hierarchical representation also reduces redundancy by concisely representing any input using only a small number of patterns. Consequently, the representation produced is succinct and more robust to noise, capturing higher-level abstractions that should be well-suited to transfer learning applications.

Extended their new self-taught learning algorithms for learning hierarchical representations from unsupervised data. This algorithm extends the deep belief network learning algorithm by encouraging the features to be sparse (i.e., to be zero most of the time). Crucially, Ng's group demonstrated that the new algorithm can transfer higher-level patterns (such as angles in images) than previous methods, and can lead to better classification accuracy than the previous single-layer self-taught learning algorithm. Developed a new self-taught learning model for transfer learning domains in which the input data is binary, discrete, or of several other types that were difficult to handle using their previous algorithm. This includes important data types such as text documents. The model allows the domain characteristics to be explicitly captured, allowing higher-level transfer than before. Ng's group also developed an efficient algorithm for learning and inference in this model. In preliminary results, the algorithm is several times faster than standard off-the-shelf optimization software.

Implemented their exponential family sparse coding algorithm for self-taught learning, and applied it to two types of transfer tasks. In one, they tested transfer from news articles to 50 webpage classification tasks; in another, they tested transfer from news articles to 10 newsgroup classification tasks. They found that, on average, the transferred knowledge leads to a 10-30% improvement in accuracy on the target task.

Implemented a distributed program to learn large restricted Boltzmann machine (RBM) models for transfer learning. The parallel algorithm is guaranteed to converge to the optimal parameter values. The computation was successfully tested on a cluster consisting of 20 individual computers.

Developed a translation-invariant sparse deep belief network model for self-taught learning, along with an efficient algorithm for training the model from unlabeled data. Using a probabilistic max-pooling operation, the algorithm can perform inference in a probabilistically sound way. Ng's group showed that this algorithm can learn interesting features -- such as object parts -- from large, unlabeled images (whose size is much beyond the typical size of images that could be used efficiently in past work).

Evaluated the model by applying it to self-taught learning tasks. They showed that the model learns useful hierarchical features for self-taught learning, and that the second layer representation for natural images contains more informative features (such as corners, arcs, contours) than the first layer features (oriented gabor filters) for object recognition in terms of both mutual information and classification accuracy. Further, their algorithm learns a hierarchical representation from images in an unsupervised way: it can learn object-part-based intermediate level features, as well as recursively composing them into more complex part or whole-object features in the higher layer.

Tommi Jaakkola, MIT, developed inference algorithms analogous to tree decomposition but based on planar graphs. The algorithms operate by decomposing the overall non-planar model in terms of planar graphs (as opposed to trees) and optimize the structure as well as the parameters of the decomposition so as to find either the MAP configuration or marginal probabilities. The results represent a step in the direction of finding effective hierarchical decomposition strategies for broader classes of probability models. The algorithms and the theoretical guarantees they are pursuing can be expected to be generally useful in transfer learning.

Developed deterministic iterative methods based on staged mixture models to effectively find and represent posterior distributions over shared parameters in parametric Bayesian models, and to replace slow sampling methods in non-parametric hierarchical Bayesian models. The methods relying on staged mixtures enjoy nice theoretical guarantees in addition to being algorithmically simple and fast.

Developed distributed message passing algorithms for finding most probable configurations. Inference tasks involving both marginalization and maximization operations are arguably the most common, especially in joint hierarchical inference across tasks, yet lack efficient algorithms. These algorithms exploit specific variational forms to enable effective propagation of max marginals across marginalizations. In addition, they are characterizing the approximation properties of such algorithms.

Implemented and tested a class of approximate inference algorithms based on parametric decompositions. The algorithms decompose non-planar graphical models into a collection of planar graphs (as opposed to trees) and optimize the graph structure as well as the parameters of the components so as to evaluate marginal probabilities over subsets of variables. These planar decomposition algorithms are slower than related approaches based on trees. This is primarily due to the difficulty of obtaining a closed form expression for the entropy of planar graphs. The new algorithms nevertheless provide superior bounds on the partition function and significantly improve the accuracy of (especially multivariate) marginal probabilities.

Developed a flexible class of approximate inference algorithms for large hierarchical models. The new methods are based on two types of controlled approximations: an upper bound on the entropy of any distribution defined over the relevant marginal polytope, and the expansion of the marginal polytope. The entropy bound is based on truncating conditional entropies associated with elimination orders. The outer bound on the marginal polytope is obtained by enforcing agreement over neighboring regions related to the original model and the specific entropy approximation. A combination of the two types of upper bounding approximations leads to widely applicable and accurate inference algorithms subsuming previous methods such as Tree-reweighted (TRW.) In particular, the approach provides a tighter upper bound on the log-partition function as well as more accurate marginals. Jaakkola et al. expect these algorithms to be of greater use in specific transfer problems (matchings, relevance determination, object recognition) than those based on planar decompositions discussed in earlier reports while still providing controlled approximations.

Implemented hierarchical non-parametric models based on sequential minimum entropy estimation. These methods lead to sparse explicit models and could be used as alternatives to sampling based hierarchical non-parametric Bayesian models.

Developed non-parametric hierarchical Bayesian models where the hierarchical organization of the samples is estimated together with the model parameters. The approach is designed for identifying shared sub-structure as well as differences across tasks. This sampling based approach complements their earlier work on deterministically estimating hierarchical models through staged minimum entropy regularization and will serve to better integrate deterministic (explicit) approximation methods with non-parametric sampling methods. The sampling approach has already been demonstrated in the context of multiple biological data sources and is readily applicable to problems such as object recognition where "examples" can be transformed into "bags of samples".

Complemented their previous work on inference methods based on truncated conditional entropies with reparameterization algorithms (in the dual form) for finding maximum a posteriori (MAP) configurations. The combination is expected to be useful in mixed propagation setting where the goal is to identify the most likely configuration of structural variables while marginalizing over variables specific to each (sub)task.

Formulated new transfer learning problems from the point of view of robust (minimax) estimation. Their approach deviates from the more common characterization of transfer in terms of what is shared across tasks and instead focuses on robustness against how the tasks may differ. It is no longer necessary to specify a distribution over tasks, and guarantees can be obtained on the basis of robustly solving a single task.

Developed approached for efficiently integrating inference calculations across different tasks. One of the key problems in this context is intersecting marginal polytopes (sets of valid marginal distributions) from different subtasks. The marginal polytopes are often non-trivial even within subtasks. The difficulties of evaluating most likely configurations of variables or computing marginal probabilities can be directly traced back to problems with characterizing the marginal polytope. Our strategy is based on controlled approximations that maintain inner or outer bounds on the marginal polytopes and their intersections. As the first step, we have developed cutting plane methodologies for obtaining tighter outer bounds on marginal polytopes. The advantage of iteratively constraining the marginal polytope is that the polytope needs to be well-specified only near the actual solution.

Extended their cutting plane methodologies for obtaining tighter outer bounds on marginal polytopes. These results were limited to random field models with binary and pairwise connectivity. The extension involves deriving a new class of outer bounds on the marginal polytope for non-binary and non-pairwise models. The key realization is that valid constraints on the marginal polytope can be constructed by a series of projections onto the cut polytope. Our approach is broadly applicable and highlights emerging connections between polyhedral combinatorics and probabilistic inference.

Developed a new generation of message passing algorithms for finding the MAP configuration of variables. The methods are aimed at resolving hidden causes in object models and training energy based models in multi-task settings (see task R8 below). The algorithms are similar in structure to max-product but always converge and can be shown to find the exact MAP solution in various settings. They are derived as block coordinate descent methods in a dual of the LP relaxation of MAP but require no tunable parameters such as step size or tree weights, and are as easy or easier to implement than the typical max-product or its generalizations.

Developed energy based latent variable models for multi-task object modeling.The overall formulation (it turns out) is in broad terms similar to the recent approach by McAllester et al. These models, however, make use of a specific class of message passing algorithms for finding MAP configurations of latent variables. These algorithms monotonically decrease the dual of an LP relaxation and, as a result, enable us to train the energy based models iteratively, analogously to EM, regardless of the latent structure. Evaluation of the approach is underway.

Developed anytime algorithms for combining different learning tasks. The overall problem involves two main threads. First, one approximately characterizes the marginal polytope associated with each model (task) and determines how such polytopes can be intersected to combine the different tasks. The second thread extends the cutting plane methodology for inference to incremental anytime induction of models. Jaakkola's group has previously developed cutting plane methodologies (with projection) to accurately represent the marginal polytope of each model (task) around the solution of interest. The intersections of such marginal polytopes, exact or approximate, can be easily characterized for models with fixed graphical structures and partially shared variables. They have further characterized the intersection of marginal polytopes for graphical models combined through data association (matchings). The matching portion is used to resolve the identities of shared variables. The complexity of the resulting problem can be shown to be at least that of max-cut. The second thread concerns with incremental (anytime) construction of models suitable for anytime (cutting plane) inference and is essentially based on cutting plane formulation for the Legendre dual.

Further developed methodologies for anytime inference and model induction. The goal of this work is to solve a set of related tasks under specific constraints on computational resources. To this end, they have developed anytime algorithms for distributed inference where the complexity of the inference calculations is iteratively tailored to the task at hand. This is accomplished by iteratively enforcing higher order consistency constraints in an overall (dual) re-parameterization approach. The algorithm provides a certificate of optimality or an acknowledgement of failure when the available resources have been exhausted. The methods have already been successfully demonstrated on hard combinatorial design tasks that reflect structural alignment problems accompanying high level transfer learning problems. The complementary model induction step is in progress (expected to be completed by the next reporting period). They have also focused on exploiting sparse model descriptions both in the distributed operations as well as in selecting appropriate consistency constraints. Higher level models are predominantly sparse.

Explored the use of anytime inference algorithms for transfer learning. The formulation treats task specific inference calculations interchangeably with estimation and leads to a new measure of transfer in terms of task specific computation. A simple realization of this problem formulation appears in structured prediction where challenging inference calculations for each training instance can be cast in terms of estimation. The task specific parameters to be estimated in this setting correspond to a (monotone dual) relaxation of inference calculations, tailored to minimize the same loss. A number of approximate inference methods have been proposed for structured prediction (e.g., by Koller's group, UAI 2008). They provide a particularly stable extension of such approaches to broader classes of transfer learning tasks that are solved via monotone relaxations.

Analyzed transfer learning from the point of view of quantifying how computational resources should be allocated across tasks. The amount of computation spent on each task can vary in small increments (the increments correspond to elementary operations in distributed inference). The inference operations, on the other hand, can be related in a strong way to the effective degrees of freedom that are fit to each task separately. The analysis setup is designed to reveal stronger generalization by limiting task specific computation.

Extended linear programming relaxations for complex inference calculations by introducing a latent hierarchy of sparsely represented functional constraints between the variables. The approach is designed for computational efficiency and accuracy in models where relaxations based only on direct interactions are insufficient (most models) and models where clusters containing more than a few variables are too costly (e.g., stereopsis).

Models where the variables take a large number of distinct values are particularly challenging for anytime inference algorithms. This is because finding and incorporating higher order consistency constraints in linear programming relaxations becomes quickly infeasible as the order of the constraint increases. Jaakkola's group has worked towards solving this problem by sparsely representing higher order consistency constraints between the marginal probabilities and developing dual messaging passing algorithms that exploit the sparsity. Jaakkola's group has derived crisp and efficient dual message passing algorithms for sparse constraints, formulated a margin based approach to efficiently search for sparse constraints, and demonstrated the computational gains from the approach.

The success of transfer learning with approximate inference depends critically on the representation of anytime inference operations. Jaakkola's group has developed a unifying framework for dual LP relaxations, mapping different formulations to each other, including block updates. These results are useful in an overall transfer learning approach where the allocation of computational resources across tasks plays a central role.

Leslie Kaelbling and Tomas Lozano-Perez, MIT, defined hyperprior on rule sets and conditional distribution of specific rule set given the prior and developed staged approximate inference strategy, in which data from observed tasks 1 to k are used to infer general rule distribution; and then that general distribution, plus a small amount of data from task k is used to infer a rule distribution for task k.

Task R2: Bayesian Reinforcement Learning

Michael Littman, Rutgers, dolved a long-standing open problem in efficient reinforcement learning---learning a Bayesian network model (DBN) of an environment in polynomial time. The problem was originally posed by Koller and Kearns in 1999 and the solution built on insights from Koller, Ng, and Abbeel. As part of the solution, Littman formulated a new metric for measuring efficient learning, which he refers to as “KWIK” learning. A KWIK learner “Knows What It Knows” about its environment, meaning that it can guide its own exploration, as appropriate, to quickly acquire the knowledge needed to maximize performance.

Explored a new model of RL environments, originally due to Sherstov and Stone (2005). The model, which they are calling “RAM” for “relocatable action model”, holds promise for capturing and transferring transition knowledge between states and problems. Littman's group's RAM learner was applied to transfer in a set of simpler grid-world domains. They found that, in spite of the rapid speed with which RAM learners acquire and use models, there was a 23% improvement when transfer was used. In this experiment, the source domain was tiny (9 states) and the target domain substantially larger (81 states) and optimal paths grew from roughly 5 or 6 to over 200. Nevertheless, positive transfer was observed.

A similar experiment was carried out using another representation developed in Littman's group. Specifically, they have devised the first Bayes-net-based RL system that automatically learns its own Bayes net representation. They showed that, in domains in which the Bayes net is unchanged between source and target; excellent transfer rates can be demonstrated.

Developed a new approach to reinforcement learning that combines the strengths of efficient learning in the "PAC-MDP" framework with the powerful, flexible representations provided by Bayesian approaches. They demonstrated the approach in the transfer setting by exploiting a hierarchical Bayesian model to speed up learning of a new task based on experience with similar tasks.

Carried out an evaluation of their novel Bayesian reinforcement learning algorithm, BOSS, in stochastic domains. It soundly outperformed existing Bayesian and non-Bayesian approaches variations of standard testbed problems. It was also demonstrated working with a non-parametric Bayesian model learner, demonstrating within-domain transfer that led to faster learning than when run with a transfer-less prior. These results were disseminated at the UAI 2009 conference. An unexpected accomplishment was that several students in the lab participated in the international reinforcement-learning competition and took first prize in two of the five categories.

Studied the problems of exploration in domains with Bayesian priors. Given a Bayesian representation of the probability over models in the class being learned, there are several possible goals for action selection. The most natural and best studied is Bayes optimal action selection. This approach says that actions should be taken to maximize expected reward in the start state given the uncertainty in the current model. Littman's group has focused instead on the PAC-MDP objective, which says that actions should obtain near optimal reward in all but a few time steps. Building on a result from Ng's group, Littman's group recognized that PAC-MDP is not an approximation of Bayes optimal, but, in fact, can be preferable. In many scenarios it is also more consistent with human and animal behavior.

Analyzsis of "Thompson sampling", a simple sampling approach to acting in domains with Bayesian priors, has shown that it can achieve the PAC-MDP objective. This realization greatly simplifies the types of algorithms that can be studied to obtain useful guarantees and allows the focus to be on the Bayesian modeling instead of complex issues on the decision-making side.

Tom Dietterich, Alan Fern, Prasad Tadepalli, OSU, evaluated a multiagent RL approach that combines the two ideas assignment-based task decomposition and relational templates. By decomposing the overall task into task assignment to agents and the task execution by agent teams, they achieved significant scaling up to 12 agents. The lower level of task execution has small decomposed state space and can be transferred across multiple domains. The higher level search is more global but takes advantage of efficient algorithms like the Hungarian algorithm for bipartite graphs. This combination proved very effective and resulted in successful transfer from 6v2 agent domains to 12v4 agents.

Extended their model-free Bayesian policy search approach to allow for contextual information to be used when assigning roles to agents. The approach is based on a hierarchical DP model which is used to learn about the number and types of agent roles in a decision problem, where an agent role corresponds to a distribution over policy parameters, so that agents of the same role behave similarly. The model was extended to allow for the DP class assignment of agents to roles to depend on contextual features of individual agents. An MCMC inference process was developed that automatically learns the kernel parameters dictating the assignments as well as the number and types of agents roles. Experiments were conducted in multi-agent battles in the game of Wargus. It was demonstrated that the role structure of a domain can be learned from demonstrations provided by an expert. Further, it was shown that this role structure could be transferred to new problem using our Bayesian policy search approach, leading to significant speedups in learning. Finally, it was demonstrated that role structure could be discovered automatically during the RL process with an uninformative prior, leading to speedup compared to baseline approaches that do not attempt to discover role structure.

Developed an assignment-based decomposition approach to multi-agent reinforcement learning. They show effective transfer across different numbers of agents of different types in a tactical RTS domain by combining assignment-based task decomposition and relational templates. At the high level, the task of defeating the enemies is decomposed into defeating each enemy using a group of friendly agents. At the lower level each group of friendly units is scheduled to defeat their assigned enemy independent of other enemy units. The lower level is efficient because each team works independently of each other and leads to transfer across multiple domains. The higher level search is more global but takes advantage of the Hungarian algorithm for bipartite graphs. This combination proved very robust and resulted in successful transfer from 6v2 agent domains to 12v4 agents of different agent types.

Task R3: Hierarchical Reinforcement Learning

Tom Dietterich, Alan Fern, Prasad Tadepalli, OSU, developed an approach to learning MAXQ subtask hierarchies for transfer. A MAXQ subtask is defined by a subgoal reward function (the pseudo-reward function), set of actions and a region of state space, and a state abstraction function such that certain conditions hold (e.g., MAX node irrelevance as defined in Dietterich, 2000, JAIR 13:227-303). The method is based on a combined top-down and bottom-up reasoning process. First, the source domain learning problem is identified without a hierarchy. The top-down process then analyzes trajectories followed by the learned policy to identify important subgoals. A bottom-up process then finds a maximal region of state-action space that satisfies the MAX node irrelevance conditions. This process is iterated to produce a subtask hierarchy. The value functions and policies are then re-learned in the source domain using this hierarchy, and the learned subtasks can then be transferred to the target domain.

Investigated learning hierarchies in RL. The goal is to learn a task hierarchy from task A which can be reused in task B, so that learning can be much faster in task B. The focus was to learn a task hierarchy from trajectories of an optimal policy. This has several subproblems as listed below: (a) Learn state-transition models from trajectories. Some progress was made on this problem by learning state-action dynamics in the form of model trees. The model trees succinctly capture the effects of actions in simple benchmark domains used in hierarchical reinforcement learning. (b) Learn to break-up trajectories into subtasks. They designed a heuristic algorithm to do this, which uses the causal structure of the actions in the trajectory to break it into subtask segments. The causal structure is deduced from the action models derived in part (a). (c) Learn appropriate abstractions for the subtasks. The goal here is to identify the subset of the features which are relevant for the completion function of the subtask. They implemented an algorithm to do this, which computes the largest set of features whose values influence the reward either directly or indirectly through other actions.

Finished a set of experiments that utilize hierarchical Bayesian models for multi-task, model-based Bayesian RL. An infinite component hierarchical model is learned from previous tasks providing an informed prior over MDP models. This prior is used to speed-up the Bayesian RL agent on new target tasks. The agent utilizes an action selection strategy inspired by Thompson sampling. The use of an infinite component model allows the agent to automatically learn the number of components and create new components when a target task is fundamentally different compared to prior source tasks. Results in a multi-terrain, multi-goal navigation world are good.

An algorithm was developed to learn hierarchies from trajectories of optimal policies in the source domain. The algorithm uses dynamic Bayesian network (DBN) models of the primitive actions to causally annotate the trajectory by identifying producer-consumer relationships between the different actions in the trajectories. It uses the causal annotations to heuristically partition the trajectory into subtasks. The algorithm is recursively called on the subtasks to create a full hierarchy with associated abstractions that are computed from the DBN models. Empirical comparisons of the hierarchy learning algorithm in several domains showed that the new algorithm outperforms hand-designed hierarchies. Under some favorable conditions, learning is orders of magnitude faster than other state-of-the-art algorithms.

Extended methods for learning subroutine hierarchies in RL. The primary algorithm works by analyzing a single training trajectory in the source domain and exploits two critical concepts. First, based on Dietterich's MAXQ theory, it searches subroutines that enable good state abstractions (i.e., where many state variables can be ignored). Second, it searches for subroutines that achieve important subgoals for the overall problem. These subgoals are identified via a causal analysis of the training trajectory under the additional assumption that it should look for goals of achievement (i.e., that cause state variables to change value; as opposed to goals of maintenance that try to prevent certain variables from changing value). The primary algorithm relies on having a good algorithm for learning dynamic bayesian network (DBN) models of the effects of actions. They have developed a novel algorithm for doing this that is of independent interest for learning regression trees in which the leaf values can be functions of the predictor variables.

Developed a hierarchical Bayesian model for transferring multi-agent polices in a tactical battle setting with multiple unobserved unit types. The model learns an infinite mixture model over agent policies, where there is a component for each of the fundamental types of policies observed, which roughly correspond to one component per distinct agent role. This model is used as a restart distribution for policy gradient on new tactical battle problems.

Proved a theorem that characterizes the extent to which the single-trajectory MAXQ hierarchy learning algorithm (HI-MAT) finds optimal state abstractions. The theorem shows that if the DBN models analyzed by HI-MAT are minimal, then HI-MAT find optimal state abstractions for Max node irrelevance.

Developed a new method for decomposing an action sequence into subtasks. This method guarantees that each sub-task is decomposed into a set of child sub-tasks that have the minimum possible number of parameters to learn. The previous method only guaranteed that the maximum number of parameters required by any single child tasks was minimal.

Developed a new algorithm to learn task hierarchies for deterministic serializable domains through partial action models. This approach is expected to clarify and refine the multi-trajectory learning algorithm that is under development and lead to a more streamlined implementation combining model learning with hierarchy learning.

Extended the approach to hierarchy learning from multiple trajectories in the context of hierarchical planning. The work focuses on learning hierarchical knowledge in the form of component graphs. These graphs are proven to always exist for serializable planning domains and a sound, complete, and efficient algorithm is given for planning with them in such domains. The work also gives a sound and complete algorithm for inferring component graphs from partial models constructed from sample trajectories.

Stuart Russell, UC Berkeley, devised new representation for temporally decomposed Q-functions that avoids problems of representationally expensive nonlocal Q_e component used in previous Hierarchical RL systems. Devised a new Hierarchical RL algorithm to take advantage of the new representation.

Completed and published the first satisfactory semantic definition for high-level actions, called “angelic semantics” because it captures the fact that the uncertainty in action outcomes caused by the availability of many possible concrete implementations of any abstract plan will always be resolved in the agent's favor because the agent will choose the implementation. On this basis, the group developed, implemented, and tested the first hierarchical planning algorithms that guarantee the following properties: 1) “upward solution” --- every abstract plan that provably fails to achieve the goal has no concrete implementation that achieves the goal; 2) “downward refinement” --- every abstract plan that provably achieves the goal has a concrete implementation that achieves the goal. These properties enable efficient planning that was shown to be several orders of magnitude faster than either flat planning or hierarchical planning without semantic annotations for high-level actions. They then developed a new, generalized definition of admissible heuristic function for state sets under the angelic semantics and used it to specify and implement the first provably optimal hierarchical planner and the first hierarchical lookahead agent. Like realtime search algorithms such as LRTA*, the agent operates in scenarios where computational limitations preclude finding guaranteed plans, but is guaranteed to eventually achieve the goal if this is possible.

Leslie Kaelbling and Tomas Lozano-Perez, MIT, developed an algorithm for transferring across tasks by finding a task hierarchy that can be used to dramatically speed up learning and/or planning in a new domain. The crucial step was formulating an objective function for what constitutes a good hierarchy, given a set of data that needs to be explained. This criterion has two components: it must be simple and explain the data well. Simplicity is measured as the sum of the complexities for solving the subproblems in the hierarchy (which should be considerably smaller than the complexity of solving the problem monolithically). Explaining the data well is measured by the degree to which the actions taken in the sample trajectories are optimal given the subgoals in the task hierarchy. This is a general approach which has been demonstrated in Stratagus scenarios.

Task R4: Transfer Learning Theory

Peter Bartlett, UC Berkeley, developed general techniques for obtaining performance guarantees for transfer learning methods based on regularized risk minimization. The results apply to prediction problems with independent data. They imply that, under suitable conditions on the transfer learning problem, the performance improves with sample size more quickly than suggested by previous results.

Obtained performance guarantees for Bayesian methods that apply even when the data is chosen adversarially. Specifically, whatever the data sequence, these results show how the loss accumulated during learning by a Bayesian method is related to the cumulative loss of any model in the class. The key benefit over previous analyses is that the results are universal over data sequences. In particular, the assumption underlying previous analyses --- that the tasks are conditionally independent --- is rather arbitrary. The new techniques seem well suited to understanding the benefits of transfer in a hierarchical Bayesian model, particularly when the number of related tasks is small.

Studied the problem of online multitask prediction with expert advice. The relatedness of tasks is modeled by aiming to compete on each task against the best expert chosen from a small set. They have provided performance guarantees for a Bayesian method. Unfortunately computing the predictions is a hard problem. They have also developed an efficient online prediction strategy whose performance degrades linearly with the number of times the task changes. In the special case of sequentially presented tasks, this efficient method gives the same performance guarantees as the Bayesian method.

Developed an algorithm and performance bounds for the problem of online discovery of similarity mappings. This is a generalization of the problem of multitask learning with expert advice that includes problems such as online clustering and feature selection. The application to multitask feature selection has been implemented as part of the transfer learning toolkit.

Developed an adaptive online prediction method for online convex optimization, adaptive online gradient descent. (Online minimization of a convex criterion is a general formulation that includes worst-case prediction problems.) Bartlett's group also provided general lower bounds for these prediction problems, which, in particular, show that the new method gives optimal rates of decrease of regret.

Developed worst-case log-loss regret bounds for Bayesian model averaging algorithms in the regression setting. These bounds are valid for arbitrary priors, and the regret term includes a smoothness property of the prior.

Developed an algorithm for reinforcement learning, called Optimistic Linear Programming, and showed that in learning to control a Markov Decision Process, the gap between the performance of this algorithm and that of the optimal policy grows only logarithmically with time.

Investigated the problem of multitask prediction with limited feedback, which is a step in the direction of multitask sequential decision problems. They developed a prediction method for online linear optimization with partial monitoring (a bandit problem, where only the loss of the chosen action is available). They showed that, with high probability over the choices of the algorithm, its regret, that is, the amount by which its performance falls short of the best choices in retrospect, grows at an optimal rate.

Investigated the problem of linear prediction with partial monitoring. Previous algorithms that gave optimal regret (regret is the amount by which performance falls short of the best choices in retrospect) required computation time exponential in the problem dimension. They developed efficient algorithms for these problems.

Developed regularization-based methods for online learning, together with analysis techniques that should facilitate the choice of appropriate regularization functionals for these methods. These techniques generalize the techniques that they developed to obtain efficient algorithms for linear prediction with partial monitoring that have optimal expected regret. They also applied these techniques to design algorithms for bandit linear prediction that have high probability guarantees on their regret. In addition, Bartlett's group has made progress on using these techniques to develop effective online multitask learning algorithms.

Investigated a novel approach to online multitask prediction via matrix regularization. The analysis showed that known spectral norms (often used in the literature) are not suited for the problem. On the other hand, structural norms yield better results.

Obtained bounds on the optimal regret rates for prediction problems in adversarial settings, which are the most natural way to model transfer learning problems. By studying the dual of the prediction problem they demonstrated a close link between performance guarantees in adversarial and probabilistic settings.

Investigated the problem of learning to control a Markov decision problem, and in particular examined the dependence of the performance of an optimal strategy on complexity properties of the problem, such as the mixing time, that measure the effective size of the MDP. They have developed a milder notion of complexity that can be viewed as a one-way mixing time --- it involves the time it takes to reach favorable states. They have made progress on the development of strategies that exploit this one-way mixing time for more rapid learning.

Developed performance guarantees for the problem of learning to control Markov decision problems, and developed strategies whose performance depends on milder notions of problem complexity than those previously considered.

Task R5: Metareasoning

Stuart Russell, UC Berkeley, investigated partial-program-constrained lookahead in a classical planning context. Identified major gaps in the field's analysis of the semantics of high-level actions. Proposed new lower and upper bound semantics that yield guarantees, where applicable, of the downward and upward solution properties. Devised lookahead planning algorithms based on the new semantics and showed order-of-magnitude speedup over flat planning and hierarchical planning without semantics.

Implemented a simple metalevel reinforcement learning task in ALisp. The partial program repeatedly samples from one of k choices, each of which returns a value drawn from an unknown distribution. Each sample has a fixed cost and at some point the sampling stops and the program commits to one of the k choices. The ALisp engine will learn to make the sampling and stopping choices. The problem, as defined, supplies external positive rewards only once a choice is made, leading to slow learning. They devised a suitable metalevel shaping reward that meets the criterion for preserving optimal policies. Experimented with features for Q-function approximation.

Conducted experiments with metalevel RL within ALisp. The basic setup is simple --- an ALisp program is written that includes choices for computational steps that eventually lead to the selection of an action. The partial program repeatedly samples from one of k choices, each of which returns a value drawn from an unknown distribution. Each sample has a fixed cost and at some point the sampling stops and the program commits to one of the k choices. Metalevel reinforcement learning was demonstrated for the first time. Developing a suitable function approximator is not straightforward, however. Since the choices are a priori indistinguishable, the approximator should be permutation-invariant. Also, the final payoff calculation is not straightforward, since the mean estimate for the current-best-action is biased by the max selection step.

Task R6: Transfer Learning for Strategy Games

Tom Dietterich, Alan Fern, Prasad Tadepalli, OSU, developed an approach to learning linear heuristic functions for controlling beam search and applied the algorithm to learning heuristics for STRIPS planning domains. The approach uses example problems labeled by a target sequence of search steps as training data. Perceptron updates are then used to keep the target sequence on the beam. The notion of "beam margin" is introduced and a convergence result is given that provides a necessary condition on the beam width, relative to the beam margin, which guarantees learning will converge.

Implemented routines for Bayesian linear regression with Gamma-Normal priors. Used these to implement a model-based multi-task RL agent that learns a prior on linear reward function models based on previous tasks and transfers that prior to new tasks. Learning in the new task is done using Thompson sampling for action selection and posterior updating. Initial experiments in colored grid-world domains show that the approach yields positive transfer.
Implemented a method for learning heuristics for controlling a breadth-first beam search planner for the tactical planning domain. This included implementing feature functions for the search nodes (i.e. partial plans) and integrating Perceptron-style weight updates into the search process. The learner takes a set of training problems that are annotated with tactical plans found using a large beam width and a hand-coded heuristic. The learner then attempts to find weights for a linear heuristic function that guides a search to the training plan using a small beam width. Our initial experiments show that the learner is able to find heuristics that have a much better performance versus beam width profile than the hand-coded heuristic.

Formulated a wide class of resource production problems and a process-centric problem formulation. The motivation for the process-centric formulation is that more standard formulations (e.g. in PDDL) result in plan lengths that are exponential in the problem size (pseudo-polynomial in the resource goals). The problem class requires reasoning about numeric resources, continuous time, durative actions, concurrent actions, numeric action arguments, and other aspects of processes. The standard planning domain language PDDL supports the first four properties to varying degrees, but extensions are required to support the full process semantics. We conducted an extensive survey of planning literature and did not find any existing planners that handle all of the features we require. They did identify two planners that appear to be promising to build on. One is LPG a planner based on local search over planning graphs and handles a reasonably large fragment of PDDL, but not continuously changing resources. The second is TM-LPSAT which is based on compiling planning problems to LCNF form (a combination of logical and linear constraints) and solving them using LPSAT. This planner is not available but in concept handles all of PDDL.

Implemented a process plan executor for resource production in Stratagus. This involved implementing a number of generic processes in Stratagus (e.g. "collect gold with a maximum of n peasants until accumulating m gold units") and a plan executor that handles resource contention and the startup and termination of processes.

Carried out two experiments to evaluate the utility of constructing transferable representations using PCA. The approach assumes the availability of optimal value functions for a number of source problems, expressed as linear combinations over a set of basis functions (that are common to all problems) and then performs PCA on the weights of the basis functions. These components are then used as basis functions in the target problems. The experiments involved a set of 50 randomly generated 5-on-5 tactical battles in Stratagus (40 source problems and 10 target problems). The results showed that the rate of convergence to optimal was improved in the target problems on average for policy search. However, because of specific implementation issues, the learned policy using the transformed basis had a slightly lower value than the policy learned using the primitive basis. For Q-learning, however, there was little observed improvement in the rate of convergence to the optimal value. This is because the primitive basis is highly engineered (because Q-learning needs to be able learn on the source problems), which leads to very rapid convergence of Q-learning in the target problems.

Implemented routines for finite and infinite mixtures of Gamma-Normal linear regression components. They used these to implement a model-based RL agent that learns a prior on linear reward functions and transition models from previous tasks and transfers that prior to new tasks. Learning in the new task is done using Thompson sampling for action selection and Gibbs sampling for posterior inference. Initial experiments in a colored grid-world domain show that the approach yields positive transfer. However, the transfer ratios are quite small due to the relative simplicity of the task.

Implemented the TM-LPSAT planner. The planner handles continuous time, numeric resources, continuous change, and numeric action arguments which are required for resource production planning in Stratagus. First, they developed a compiler from planning problems to LPSAT problems for a restricted class of PDDL+. Then they revamped an existing LPSAT solver.

Studied the computational complexity and convergence properties for the supervised learning of linear ranking functions for controlling beam search. Tractable and hard subclasses of the learning problem were identified and the convergence of simple online algorithms was shown.

Developed a SAT-based planner for resource production problems and ran initial experiments in Wargus. The planner can handle problems with small resource goals and/or a small number of "distinct processes" comprising a plan. The most natural way to extend to large resource goals results in non-linear (quadratic) constraints, which are not handled by our current system. Rather than move to a quadratic constraint solver they used coordinate ascent approaches that make multiple calls to the planner each involving only linear constraints.

Developed the infrastructure for an online planner for resource production problems in Wargus. The main component is a heuristic calculation that is based on a suitably modified variant of means-ends analysis, which is guaranteed to terminate given the assumptions satisfied by our problem. Initial experiments with the heuristic are encouraging but also highlight areas for improvement.

Completed an evaluation of utilizing PCA analysis for transfer in RL within the tactical domain. After solving a number of source problems, PCA is used to learn an orthogonal basis to represent policies, which is used for learning on target problems. Performance in terms of regret is promising compared to several baseline transfer mechanisms.

Developed a domain specific approach to learning the numeric parameters of Wargus actions (e.g. resource amounts required and produced, duration) given qualitative schemas of those actions. The algorithm uses the qualitative schemas to organize its exploration in order to quickly discover the numeric parameters.
Extended their SAT-based planner for resource production to scale to larger problems. The final approach utilizes a incremental plan refinement strategy that attempts to improve the current best plan via repeated calls to the base planner in an anytime fashion. The resulting planner improves on the original TM-LPSAT planner, which they have been building on, in terms of both speed and plan quality. However, the resulting planner is still many orders of magnitude slower than the more recent heuristic search planner for the resource production domain and is still not suitable for real-time environments which was one of the original goals.

Developed an online planning algorithm for the resource production domain that is suitable for real-time execution. The planner is based on an efficient computation of an informative heuristic and bounded search. They have found that even for a search depth of one the planner is able to outperform a human expert at complex resource production task in terms of time to achieving the goal. This planner works for a subset of PDDL that captures typical resource production actions in RTS games. To the best of our knowledge it is the only AI planner that can effectively deal with temporal, concurrent actions and numeric resources in a way that is suitable for a real-time setting.

Developed an algorithm for model learning in resource production domains that can leverage qualitative action schemas. The algorithm uses the qualitative schemas both to help decide what actions might be worth exploration and as a bias on the action definitions themselves. Initial tests show that the schemas speedup model learning by a factor of about eight.

Created a problem generator for the Y2 tactical CP, which is substantially more complex than that of Y1. A base non-transfer learning algorithm was developed where multiple version of OLPOMDP are used to train the multiple agents. For this problem it does not appear necessary to include explicit coordination structures in order to find a solution in a practical time frame.

Developed a transfer mechanism for the multi-agent tactical CP. The basic idea is to analyze learned policies from source problems to discover the fundamental "roles" played by the various agents. Here agents that have the same role have similar policies (e.g. a long range unit generally has a different role than a close range unit). The analysis also attempts to discover a mapping from properties of units in the initial state of the battle to their roles. Given a new problem the agents are each assigned roles and their policies are initialized accordingly. For the purposes of the challenge problems they are using a simple role discovery approach that just clusters policies using k-means, using a measure of policy similarity as a distance metric (the number of clusters is automatically selected). They then learn a classifier that is able to accurately map agents to their appropriate cluster/type.

Developed an approach for analyzing the topological structure of Stratagus maps resulting in a graph representation of regions and connectivity.

Developed a new UCT-based algorithm that supports planning fully concurrent activity. It is easy to plug in new actions models into the resulting planner, which supports our goal of model-based transfer. The algorithm can also take as input a variety of optimization goals that trade-off the speed of the assault with the damage taken. They have evaluated the resulting UCT algorithm on a set of 15 diverse tactical assault problems and compared to a number of baselines including the existing Wargus AI. The planner is a consistent top performer, often by a significant margin. Experiments demonstrate that one can effectively use the UCT stochastic planning algorithm in a domain where there are a large number of agents with temporal actions that must be executed concurrently.

Task R7: Transfer Learning for Manipulation

Andrew Ng, Stanford, developed and tested an algorithm for choosing appropriate grasp positions for a novel object, whose 3D shape is unknown, and where the object is being perceived for the first time by the algorithm using vision. Using a computer graphics simulator to generate training data, the group has developed transfer learning methods to identify good grasps for such object, given (usually two or more) input images of the object to be manipulated. They developed a statistical triangulation method to estimate the 3D location of the grasping point for the object. They tested the transfer learning methods on a real 5 degree-of-freedom robot arm to pick up various novel objects. The algorithm used was an approximate variant of a hierarchical Bayesian learning algorithm (developed by Jordan, and also similar to the class of algorithms analyzed by Bartlett's work under task R4). With emphasis on transferring one type of objects to another, (e.g., coffee cups to tea cups) Ng's group has generated transfer ratios in the range of 3.0 to 4.5, depending on the transfer level.

Developed and tested an algorithm for choosing appropriate grasp orientations for a known object (for when the object is placed at an unusual orientation). This builds on their earlier work, which focused mainly on predicting the location of a grasp. Using a computer graphics simulator to generate training data, they developed transfer learning methods for identifying good grasp orientations for such an object, given two input images of the object to be manipulated. The approach developed uses a probabilistic learning algorithm, and poses the problem of predicting the 3D grasp orientation by embedding the manifold of 3D grasps in a non-Euclidean space, and learning an appropriate representation over this manifold.

Developed the basic components required to develop higher level transfer learning algorithms. These transfer algorithms are used to pick up objects lying in a dishwasher. Previously, they developed transfer algorithms for predicting the location of grasps for single unknown objects against a white background. However, clutter in the images (e.g. due to dishwasher prongs) caused further challenges in perceiving the image to determine grasp. The first component that Ng's group developed was the probabilistic framework that allows transfer of knowledge to predict grasp for objects placed in a cluttered area (e.g. a dishwasher), from previously learned knowledge of grasping objects against a white uncluttered background. They improved their probabilistic model to jointly estimate the grasps from multiple cameras, and also developed a set of stereo features for improving accuracy in predicting grasp locations. Finally, they developed learning algorithms to perceive the obstacles (e.g. prongs of a dishwasher) and avoid them while grasping the object.

Demonstrated their transfer algorithm that predicts grasping points in presence of background clutter, to unload objects from a dishwasher using their robotic platform. They integrated their various subcomponents---image features (stereo and monocular), learning framework to predict grasps, and path planning algorithm to reach and pick up an object---to unload items from a dishwasher. They developed a set of stereo features, and an improved probabilistic model for transfer that resulted in higher accuracies in predicting grasping points and identifying obstacles such as dishwasher prongs. They improved their potential field based algorithm to plan a path in presence of simple arrangement of obstacles. The algorithm also decides the order in which to pick the objects. For unloading a complex arrangement of objects (in which objects are closely placed on top of each other in presence of obstacles), they use a different algorithm such as Probabilistic Roadmaps.

Tested their transfer learning algorithm for grasping objects in presence of obstacles for the task of unloading a dishwasher and picking or placing objects in kitchen or office environments. They further tested their algorithms on their second robotic platform STAIR 2.0.

Developed a probabilistic model to generate data for training a transfer learning algorithm to recognize objects, their orientations and the point at which to grasp them. Using this data and their transfer learning algorithm, they demonstrated a robot fetching a stapler in response to a verbal request completely autonomously.

Improved their grasping algorithm, and tested it for grasping tasks on a second improved robotic platform. These tests demonstrated that transfer learning algorithms for grasping, trained on synthetic images, transferred well to grasping on different robots (with different cameras/arms).

In the application domain of grasping, the grasping strategy changes with different kinematics of the arms. E.g., for a five degree-of-freedom arm with a two-fingered hand, a single grasping point is enough; however, for a seven degree-of-freedom arm with three-fingered hand, a detailed configuration of each of the three fingers needs to be inferred. Ng's group developed a transfer learning algorithm that is agnostic to the particular kinematic configuration of the arm and infers the configuration of the all the joints in the arm and fingers jointly. An extensive experimental evaluation on grasping novel objects using a three-fingered hand showed a grasping success rate of 86% for medium-sized objects.

Developed a transfer learning algorithm that incorporates information from multiple sensors: stereo cameras and time of flight sensors. They identified the most informative visual features from vision data (i.e., without depth information), and used those features in a transfer learning algorithm to identify the grasping points from the 3-d data (from time-of-flight sensors).

Developed a learning algorithm that considers 3D data for inferring a grasp strategy. The 3D sensors (based on time of flight) give only partial (they see only front face of the object), sparse (sensors return no depth for many regions in the image) and noisy estimates of 3D depth. This makes it hard to compute measures such as form and force closure, contact, etc., which are required for a good grasp. Further, for grasping in cluttered environments, they need to predict full configuration of the arm (as opposed to a 2D point in the image, which they did in our prior work). Ng's group developed a supervised learning algorithm that takes partial, noisy 3D data and infers a good grasp (i.e., a full configuration for arm and fingers) for a robotic arm. Further, the same algorithm works for different types of robotic arms. The learning algorithm combines the 2D grasp estimates from the 2D image, with the 3D data to produce a full arm/finger configuration. They tested it on two robots with different kinematic configurations. In extensive experiments, the algorithm was successfully able to grasp novel objects in cluttered environments.

In another application of this algorithm, Ng's group also considered the problem of opening doors, even ones that were never seen before in the training set. Opening a door is a manipulation task that goes beyond grasping in that a robot needs not only to infer how to grasp a door handle, but also to infer how to turn it in order to open the door. Using our algorithm that considers multiple sensors (2D and 3D), Ng's robot infers how to manipulate the door handle in order to open it. In extensive experiments in (pushing) open different types of new doors performed in two different new buildings, their robot was able to open doors (by turning the handle) 31 out of 34 times in doors on five different floors. There were 20 different types of doors in these experiments. This makes their robot the first to be able to open new doors.

Developed a transfer learning algorithm for optical proximity sensors for grasping. While long range sensors such as vision or 3d sensors are useful for predicting an approximate grasp, the optical proximity sensors are useful for reactively adjusting the grasp while actually executing it. (Long range sensors are less useful here because of spatial resolution and occlusions by the robotic hand.) Ng's method employs a robust, belief-state-based surface pose estimation from the sensor data. They also developed a reactive hierarchical grasp controller that regulates contact distances for grasp even in absence of reliable surface estimates. The sensor model learned from a set of surfaces, and the probabilistic models transferred it to surfaces with very different optical properties.

Devised a simple and novel method for visual serving and automatic calibration using the robot end effector as a target. Ng's group also proposed a simple nonparametric, transfer learning method for calibrating a 3D sensor and a camera (2D sensor), using only very few unlabeled images. The new methods led to significantly better performance on the transfer learning task of grasping and picking up different objects.
Combined 3D sensors with a camera (2D sensor) for improving object-detection and used it with transfer learning algorithms developed earlier (e.g., manipulation for door-opening) for having a robot find and make an inventory of objects in office environments.

Tested the 3D sensor algorithm on a number of applications including object detection and door opening. They also show that incorporating high-quality 3D information into the sensing scheme of a mobile manipulator can increase its robustness when operating in a cluttered environment.

Leslie Kaelbling and Tomas Lozano-Perez, MIT, developed a method for using previous experience in robot motion planning problems to speed up solution of new problems. The planning algorithm builds a graph of known free locations and uses it to plan a path from a starting to a goal configuration. In a new problem, some of these links may not be traversible due to obstacles, so those are temporarily pruned from the graph. In addition, the start and goal locations may not be currently included in the graph. They carried out experiments to study the transfer-learning properties of this method, including transfer to robots with different sizes, to different goals, and to different obstacle configurations. These experiments generated transfer ratios in the range 1.5 to 6.0, depending on the detailed setting.

Kaelbling and Lozano-Perez implemented and tested an algorithm for choosing appropriate learnt grasp prototypes for a novel object and adapting the learned grasp to the new object. The approach uses nearest neighbors for selecting a grasp prototype and a learned quality function to choose the most likely grasp adaptation. They carried out experiments to study the transfer-learning properties of this method, with an emphasis on transfer from manipulating simple boxes to manipulating complex objects composed of multiple sub-parts. These experiments generated transfer ratios in the range 5.2 to 14.0, depending on the detailed setting.

Task R8: Transfer Learning for Vision

Daphne Koller, Stanford, addressed the important challenge of recognizing a variety of deformable object classes in images. Of fundamental importance and particular difficulty in this setting is the problem of "outlining" an object, rather than simply deciding on its presence or absence. A major obstacle in learning a model that allows us to address this task is the need for hand-segmented training images. They have developed a novel landmark-based, piecewise-linear model of the shape of an object class. They then formulate a learning approach that allows us to learn this model with minimal user supervision. They circumvent the need for hand-segmentation by transferring the shape "essence" of an object from drawings to complex images. They have shown that our method is able to automatically and effectively learn and localize a variety of object classes.

Discriminative tasks, including object categorization and detection, are central components of high-level computer vision. Sometimes, however, one is interested in more refined aspects of the object in an image, such as pose or particular regions. They developed a method (LOOPS) for learning a shape and image feature model that can be trained on a particular object class, and used to outline instances of the class in novel images. Furthermore, while the training data consists of uncorresponded outlines, the resulting LOOPS model contains a set of landmark points that appear consistently across instances, and can be accurately localized in an image. Our model achieves state-of-the-art results in precisely outlining objects that exhibit large deformations and articulations in cluttered natural images. These localizations can then be used to address a range of tasks, including descriptive classification, search, and clustering.

One of the original goals of computer vision was to fully understand a natural scene. This requires solving several sub-problems simultaneously, including object detection, region labeling, and geometric reasoning. The last few decades have seen great progress in tackling each of these problems in isolation. Only recently have researchers returned to the difficult task of considering them jointly. In this work, they consider learning a set of related models in such that they both solve their own problem and help each other. Koller's group developed a framework called Cascaded Classification Models (CCM), where repeated instantiations of these classifiers are coupled by their input/output variables in a cascade that improves performance at each level. Our method requires only a limited “black box” interface with the models, allowing us to use very sophisticated, state-of-the-art classifiers without having to look under the hood. They demonstrate the effectiveness of our method on a large set of natural images by combining the subtasks of scene categorization, object detection, multiclass image segmentation, and 3D reconstruction.

Many problems in computer vision can be modeled using conditional Markov random fields (CRF). Since finding the maximum a posteriori (MAP) solution in such models is NP-hard, much attention in recent years has been placed on finding good approximate solutions. In particular, graph-cut based algorithms, such as alpha-expansion, are tremendously successful at solving problems with regular potentials. However, for arbitrary energy functions, message passing algorithms, such as max-product belief propagation, are still the only resort. They developed a general framework for finding approximate MAP solutions of arbitrary energy functions. Our algorithm (called Alphabet SOUP for Sequential Optimization for Unrestricted Potentials) performs a search over variable assignments by iteratively solving sub problems over a reduced state-space. They provide a theoretical guarantee on the quality of the solution when the inner loop of the algorithm is solved exactly. They show that this approach greatly improves the efficiency of inference and achieves lower energy solutions for a broad range of vision problems.

Developed an articulated shape model based on a tree-structure of parts and rotation about a “joint.” A parts-based localization technique has been implemented and tested for localizing articulated objects in images.

Showed that transferring learned part models to neighboring object classes is appropriate for learning shape distributions more effectively. It was even demonstrated that more distantly related classes benefit from transferring part models for the purpose of learning shape. Koller also showed that the transfer of part models to sibling object classes improves localization of articulated objects in real images.

Demonstrated the effectiveness of the LOOPS model for answering semantic questions about the data not known at training time. By projecting the test data into a shape space learned in the training data, many shape-based tasks become much easier. This will allow the transfer of metadata along the surface of an object in the case of articulated objects, and shows that such metadata can be “attached” to semantically consistent locations on the object.

Developed a context model relating superpixel classification to object detection, which will allow the combination of a region-based monocolar 3D reconstruction with Koller's shape models. The group also began to integrate these two methods toward the goal of using shape models with 3D information for improved 3D reconstruction of scenes and objects for robotic manipulation.

Developed a framework for transferring knowledge between the tasks of object detection, segmentation, and 3D reconstruction. The model developed achieved mutual benefit above considering each of these tasks separately.

Solved the problem of negative transfer for shape models. The algorithm automatically learns which shape components are beneficial for transfer and uses them to achieve positive results.

Achieved transfer for object shape and feature models to specific classification problems. General object class knowledge is learned in the first stage, and this knowledge is transferred to a separate, supervised classification problem. The strong benefit of this transfer was demonstrated.

Demonstrated the ability to register 3D models to 2D images. The algorithms used a 2D match of the 3D model to the image, as well as a 3D reconstruction of the image. Positive results were reported for the Y3 deliverable.

Completed exploration of the benefits that TAS and CCM models can have compared to each other in leveraging context for successful transfer. Experiments were performed in the context of high-level scene understanding, demonstrating that the context is not only a cue for solving subtasks but an element of interest on its own.

Developed a model for incorporating hierarchical relationships in appearance models. The group also developed an algorithm for transferring knowledge between a pixel-based segmentation model and a shape-based object model.

Andrew Ng, Stanford, successfully applied their convolutional deep belief network model to perform object detection, achieving more than 90% performance on a sample task. The model was also capable of filling-in severely impaired images, by performing hierarchical inference using parameters learned using unlabeled data.

Developed a hierarchical image model that does not use parameter sharing, and has more than a hundred million independent weights to be tuned. They developed a parallel method using graphics processors that can learn such large models in an order of magnitude less time than a non-parallel method.

Demonstrated that their two-layer representation for images produces better performance on a standard image classification task than a conventional single-layer representation. This demonstration validates their search for “deeper” transfer learning algorithms, that transfer higher-level knowledge between tasks.

Applied the convolutional deep belief network (CDBN) model for unsupervised transfer learning to two challenging tasks: object recognition and handwritten character recognition. On both tasks, they demonstrated performance comparable to extensively hand-engineered state-of-the-art methods, even though the CDBN model is trained only using unlabeled data. This shows that the CDBN model can achieve high-quality transfer even with unlabeled data and no hand-engineering of transfer features.

Implemented a parallel learning algorithm for learning large deep belief networks using commonly available graphics hardware. Using this algorithm, they were able to reduce the learning time from two weeks to 6 hours for a large model, and train models that are an order-of-magnitude larger than previously published models.

Developed the CDBN model for unsupervised transfer of features for image data, and demonstrated that the model can be successfully applied to several challenging image tasks. Applied the CDBN framework to object detection tasks. To incorporate scale invariance in the image features obtained by transfer learning, they designed an image pyramid architecture, and computed the object bounding box and detection score using convolutional voting on the high-level CDBN feature activations. The resulting algorithm outperforms previous state-of-the-art algorithms on the task of bicycle detection on the PASCAL 2006 object detection dataset.

Generalized their approach to using parallel graphics processors for large-scale implementation of two widely used unsupervised transfer algorithms for learning of high-level features. Their method is up to 70 times faster on the task of learning deep belief networks, and up to 16 times faster on the sparse coding learning algorithm. To further encourage this line of work, they also documented and released their code for using graphics processors for the sparse coding algorithm.

Developed an active perception algorithm for improving object detection. In home and office environments, the object may appear in non-canonical views to the robot (e.g., it’s hard to detect a mug if its handle is not visible). Their transfer learning algorithm chooses an optimal manipulation or navigation action for the robot to take, using a criterion based on mutual information. The robot actively decides to either move the object or see it from a different view. This algorithm helped improve the performance of object recognition significantly.

Michael Jordan, UC Berkeley, developed a new approach to the joint recognition and segmentation of natural scenes. Two complementary problems in scene understanding are those of segmenting scenes into constituent objects and structures, and recognizing the objects depicted in the image. The new approach involves integrated scene models which use cues developed for image segmentation to better recognize objects, and identified objects to regularize segmentation.

Explored an application of their earlier work on hierarchical Dirichlet processes (HDPs) to learning low-level image representations suitable for multiple high-level vision tasks. In particular, they have shown how to extend the HDP formalism to hidden Markov trees. In this setting the cardinality of the state nodes in the tree is unknown and is inferred from data. This approach makes it possible to learn representations that capture non-local appearance patterns and to perform scene categorization.

Developed a new approach to the joint recognition and segmentation of natural scenes. Scene understanding systems must simultaneously segment images into constituent objects and structures, and recognize depicted objects. They have developed a hierarchical model which shares object appearance information across a family of related scenes, and thus transfers learned segmentation cues to new environments. They have shown that the "Pitman-Yor prior" underlying our model better matches the heavy-tailed, power law statistics of human segmentations than existing approaches, and are currently exploring performance in a large-scale database of natural scenes.

Released a publicly distributable software implementation of their hierarchical nonparametric Bayesian method for image segmentation and unsupervised object discovery.

Developed a library of learned low-level image representations that are suitable for many high-level tasks. The approach is based on a hierarchical Dirichlet process hidden Markov tree which discovers non-local appearance patterns which characterize natural scenes. Current experiments are exploring the usefulness of these representations for two challenging tasks: image denoising (process of removing noise from an image) and scene recognition. They are also developing more efficient learning algorithms which better scale to large databases.

Developed a new architecture for visual scene recognition known as a "hierarchical Dirichlet process hidden Markov tree." This architecture makes it possible to model relationships among clusters of wavelet coefficients that transfer among scenes. This approach has been shown to be effective using standard scene recognition testbeds.

Developed a novel image segmentation method based on nonparametric hierarchical Bayesian models. In this approach, a Pitman-Yor model is used to place a prior on segmentations (in earlier work, Jordan's group has demonstrated that this model captures the empirical distribution of segment sizes across a wide range of real images). The key to this approach has been to use latent Gaussian processes to parameterize each of a set of Pitman-Yor processes and to couple these processes across the image. They have developed efficient variational inference algorithms for this architecture and demonstrated that the approach yields state-of-the-art performance in visual segmentation. They have also shown that this architecture yields a new methodology for unsupervised object discovery.

Showed that their hierarchical Pitman-Yor model for unsupervised image segmentation can also be used for unsupervised object discovery in visual scenes. The model allows knowledge about putative object types that is discovered in one scene to be transferred to other scenes.

Leslie Kaelbling and Tomas Lozano-Perez, MIT, implemented two separate methods for using a 3D model to compile view-specific templates for detection of objects in images. One method was tested in a large collection of images of chairs, under a variety of transfer-learning settings, including transfer from synthetic to real images and from one view to another view (both directly and by learning the view transform). These experiments generated transfer ratios in the range 2.25 to 11.85, depending on the error metric and the transfer method.

Implemented and tested a method for learning the parameters of a hierarchical Bayesian grammatical model that describes the high-level structure (presence and absence of parts) as well as the shapes of those parts and their relations. They applied it to a synthetic data set of labeled 3D images of chairs and tested how well learning one class of chairs could transfer to learning of other classes of chairs, and generated transfer ratios in the range 8.7 to 15.0.

Developed a grammar-based object recognition approach using probabilistic shape grammars whose productions are specified by a human but where shape, appearance and geometric relationships among parts are learned from labeled data. An efficient recognition algorithm has been tested as well as a variant of the inside-outside algorithm for learning the parameters of probabilistic shape grammars. An extension of the algorithm to sum out all the grammar parameters so as to achieve more reliable class comparisons has produced significantly increased accuracy over a “single best parse” approach. This method was tested in the domain of tools, in particular, localizing wrenches in very cluttered scenes. This was the basis of the successful Y2 Go-NoGo test. The most recent focus has been on automatic learning of appearance models in conjunction with learning the grammars.

Developed a hierarchical bayesian approach to generating virtual views of objects from novel viewpoints. In particular, they developed an approach to select the most appropriate cross-view shape transformations from a library of how known shapes transform. They extended their approach to require only a single image labeled with part information, this is then propagated to all subsequent images to predict the part labeling. Performance of this semi-supervised system is comparable (or better) than the fully supervised system. This approach was also extended to predict the relative depth of parts on an object based on a single training example. This leads to better predictions of novel views.

The approach has been extended to make detailed prediction of the depth map of an object given an estimate of the ground plane. This can generate data that is accurate enough to grasp an object. They have tested the method with the robot and obtained good grasping performance, including grasping of parts of the object not visible to the camera. They performed successful experiments on reconstruction and grasping of 5 object classes.

Task E1 and E2: Manipulation and Vision Testbeds

Ng's group created a dataset for testing by manually labeling grasps in the images of real objects placed in a dishwasher. They used this dataset to extensively evaluate the performance of the transfer algorithm for predicting grasps. They also performed experiments on their robotic platform to unload objects from a dishwasher. They performed extensive experiments on the STAIR platform to test grasping of objects using higher-level transfer from easily generated, simulated images of other objects. They also tested their algorithm for predicting grasp orientations on the STAIR platform. With these experiments, they demonstrated the practical applicability of their transfer-based grasp prediction algorithms. Further, they started to implement their new unsupervised transfer learning algorithms for the transfer learning toolkit. Ng's group developed a transfer learning algorithm to transfer from vision to grasping. Using the object detection algorithm developed by Koller's group, Ng's group developed transfer learning algorithms that improve the accuracy of grasping significantly in cluttered environments. Ng's group developed a method to improve the performance of vision using robot manipulation. The transfer learning method maximizes the mutual information using Gaussian processes to choose an optimal manipulation action in order to improve the performance of object detection significantly. Ng's group developed a new joint probabilistic model for location and orientation of objects. This solves the problem of learning in the highly non-linear and non-Euclidean space of orientations, thus advancing the state-of-the-art for transfer algorithms in real domains.

Kaelbling and Lozano-Perez's group developed large sets of labeled images of chairs and tools for testing object recognition algorithms. Their group also developed and demonstrated an approach to transfer from visual recognition to grasp learning. This formed the basis of the successful Y3 Go-NoGo test.

Task I: Integration (the Toolkit)

Bartlett's group made the key design decisions for the transfer learning toolkit (http://multitask.cs.berkeley.edu/), and implemented four transfer learning methods for prediction problems. The toolkit is based on the open source Spider machine learning toolbox, written in Matlab, and using Matlab's object-oriented classes. The key objects are a multi-task data object (a generalization of the data object in Spider), an algorithm object and a model object. Within this design, they have implemented Ando and Zhang's multitask transfer method for prediction, based on transferring a common subspace.

Bartlett's group has implemented various components of the toolkit for handling data for multiple tasks, as well as components for testing and performing cross-validation. The toolkit interface for algorithms is implemented. The hierarchical Bayes model for logistic regression of Liang et al has also been implemented. The feature selection method of Jordan and Obozinski is being implemented, and an interface to BUGS for general hierarchical Bayesian models is under development.

Bartlett's group has added functionality to the transfer learning toolkit, including an implementation of the method of Abernethy, Bartlett, Rakhlin (COLT 2007, to appear) and Rakhlin, Abernethy, Bartlett (ICML 2007, to appear), an interface to BUGS to provide a general purpose Bayesian inference engine, and a space-efficient data representation suitable for a large text corpus. The central toolkit components have been documented, and a tutorial has been written.

Bartlett's group extended the transfer learning toolkit in several directions. The feature selection transfer method of Jordan and Obozinski has been implemented in the toolkit. The Ando and Zhang method has been extended to include a stochastic gradient descent optimization method that is appropriate for large data sets. Methods for computing transfer learning metrics have been implemented. The toolkit tutorial and developer documentation have been expanded. Additional datasets, including handwritten character recognition data and Reuters newsgroup data, have been packaged as toolkit objects. Improved functionality, such as conversion from multiclass data to multitask objects, has been added. A web interface to the toolkit, with access to the version control system, has been developed.

Bartlett's group further developed the transfer learning toolkit. Implementations of methods for calculating transfer learning metrics were completed. Nonparametric Bayesian prediction methods based on hierarchical Dirichlet process priors were implemented. An improved toolkit interface to the parametric Bayesian inference engine (BUGS) was developed. In collaboration with Ng and Koller's groups, Bartlett's group completed implementations of the Raina/Ng/Koller algorithm for Bayesian transfer learning via covariance estimation and of the Lee/Chatalbashev/Vickrey/Koller meta-prior algorithm. Several transfer learning datasets (robot grasp point prediction and Netflix movie preference prediction) were incorporated into the toolkit.

Bartlett's group published on the web an updated version of the toolkit, incorporating eight data sets and improvements to a number of methods, including the Ando-Zhang method, the BBLasso method, parametric Bayesian methods, and HDP methods.

Ng's group submitted their transfer learning algorithm for learning priors, for inclusion in the TL toolkit. The implemented code has been uploaded to the TL toolkit code base. Ng's group also prepared a transfer learning dataset for robotic grasping, for inclusion in the TL toolkit. This dataset has been delivered (by sending a url) to the UC Berkeley group. The datasets generated as part of the group's robotic grasping work have been incorporated into the Transfer Learning toolkit and are available to other researchers to further aid in the development of transfer learning and robotic manipulation algorithms. The grasping code is now being used by several research groups around the world.

Conclusions

The key high-level scientific lessons from the Transfer Learning program are:

1. **Distant tasks require general knowledge**
    1. As tasks become more distinct (higher transfer levels), the form of the knowledge learned and transferred needs to become more general purpose.
    2. For example, we can learn to improve object recognition or grasping or bicycle riding or foraging by adjusting low-level parameters; but transferring from one to the other requires higher-level knowledge like causal or geometric models.

2. **Meta learning is crucial**
    1. There are too many possible aspects of transfer to know how, in general, to move from one single task to another.
    2. Multiple training tasks allow learning of kinds of regularities that are likely to hold across tasks, which guides transfer to novel tasks by prioritizing hypothesized similarities.

3. **Hierarchical Bayes is foundational**
    1. It allows integration of prior knowledge and data from multiple sources and maintains receptivity to new information.
    2. Very rich and flexible classes of hypotheses, including sets of logical rules, meta-features, geometric models, hierarchical control strategies
    3. Hypothesis complexity automatically adapted based on amount and diversity of available data; for example, flexible clustering of previously-seen individuals speeds transfer by "soft assignment" of new individual to clusters

List of Acronyms

DBN	Dynamic Bayesian Network
HDP-HMM	Hierarchical Dirichlet Process – Hidden Markov Model
HMM	Hidden Markov Model
ISR	Intelligence, Surveillance and Reconnaissance
MAP	Maximum a Posteriori
RAM	Relocatable Action Model
RL	Reinforcement Learning
TL	Transfer Learning
TRW	Tree-reweighted

Publications By Year

2006

1. Alan Qi and Tommi Jaakkola, *Parameter expanded variational bayesian methods*, Advances in Neural Information Processing Systems, 2006.
2. Amir Globerson and Tommi Jaakkola, *Approximate inference using planar graph decomposition*, Advances in Neural Information Processing Systems, 2006.
3. Ashutosh Saxena, Justin Driemeyer, Justin Kearns, Andrew Y. Ng, *Robotic Grasping of Novel Objects*,, NIPS 19, 2006.
4. Ashutosh Saxena, Justin Driemeyer, Justin Kearns, Chioma Osondu, Andrew Y. Ng,, *Learning to Grasp Novel Objects using Vision,* , RSS Workshop on Manipulation for Human Environments, 2006.
5. Ashutosh Saxena, Justin Driemeyer, Justin Kearns, Chioma Osondu, and Andrew Y. Ng, *Learning to Grasp Novel Objects using Vision*,, 10th International Symposium on Experimental Robotics, July 2006.
6. Bhaskara Marthi and Stuart Russell, *A compact, hierarchical Q-function decomposition*, Conference on Uncertainty in AI, July 2006.
7. Brian Milch and Stuart Russell, *General-Purpose MCMC Inference over Relational Structures*, Conference on Uncertainty in AI, July 2006.
8. Dahlkamp, H. and Kaehler, A. and Stavens, D. and Thrun, S. and Bradski, G., *Self-Supervised Monocular Road Detection in Desert Terrain*, Robotics Science and Systems Conference, August 17, 2006.
9. Gal Elidan, Geremy Heitz and Daphne Koller, *Learning Object Shape: From Drawings to Images*, CVPR, June 2006.
10. Jim Rodgers, Drago Anguelov and Daphne Koller, *Object Pose Detection in Range Scan Data*, CVPR, June 2006
11. Kaijen Hsiao and Tomas Lozano-Perez, *Imitation Learning of Whole-Body Grasps*, IROS 2006, October, 2006.
12. Peter L. Bartlett and Shahar Mendelson, *Empirical minimization*, Probability Theory and Related Fields, July 2006.
13. Scott Proper and Prasad Tadepalli , *Scaling Model-based Average-reward Reinforcement Learning for Product Delivery* , European Conference on Machine Learning , 2006.
14. Sriraam Natarajan, Weng-Keen Wong, Prasad Tadepalli , *Structure Refinement in First-order Conditional Influence Language* , ICML Workshop in Open Problems in Statistical Relational Learning , June 2006.
15. Sungwook Yoon, Alan Fern, Robert Givan, *Learning Heuristic Functions from Relaxed Plans*, International Conference on Planning and Scheduling, 2006.
16. Taskar, B., Lacoste-Julien, S., Jordan, M. I., *Structured prediction, dual extragradient and Bregman projections.*, Journal of Machine Learning Research, December 2006.
17. Teh, Y. W., Jordan, M. I., Beal, M. J., Blei, D. M. , *Hierarchical Dirichlet processes* , Journal of the American Statistical Association, 2006.
18. Xing, E. P., Sohn, K-A., Jordan, M. I., Teh, Y. W., *Bayesian multi-population haplotype inference via a hierarchical Dirichlet process mixture*, International Conference on Machine Learning (ICML), New York: ACM Press, 2006.

19. Yuehua Xu and Alan Fern, *Toward Discriminative Learning of Planning Heuristics*, AAAI Workshop on Learning for Search, 2006.
20. Yuehua Xu, Alan Fern, Sungwook Yoon, *Discriminative Learning of Beam-Search Heuristics for Planning*, International Joint Conference on Artificial Intelligence, 2006.

2007

1. Aaron Wilson, Alan Fern, Prasad Tadepalli, and Soumya Ray, *Multi-Task Reinforcement Learning: A Hierarchical Bayesian Approach*, International Conference on Machine Learning, 2007.
2. Alexander L. Strehl and Michael L. Littman, *Online Linear Regression and Its Application to Model-Based Reinforcement Learning*, NIPS 2007, December 2007.
3. Alexander L. Strehl, Carlos Diuk, Michael L. Littman, *Efficient Exploration and Learning of Structure in Factored-state MDPs*, AAAI, Summer 2007.
4. Alexander Rakhlin, Jacob Abernethy, Peter Bartlett, *Online Discovery of Similarity Mappings*, ICML 2007, June 2007.
5. Ambuj Tewari and Peter L. Bartlett, *Bounded parameter Markov decision processes with average reward criterion*, Proceedings of the Conference on Learning Theory, pages 263-277, June 2007.
6. Ambuj Tewari and Peter L. Bartlett, *Optimistic linear programming gives logarithmic regret for irreducible MDPs*, NIPS, December 2007.
7. Amir Globerson and Tommi Jaakkola, *Approximate inference using conditional entropy decompositions*, 11th International Conference on AI and Statistics (AISTATS'07), 2007.
8. Amir Globerson and Tommi Jaakkola, *Convergent propagation algorithms via oriented trees*, Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence, 2007.
9. Amir Globerson and Tommi Jaakkola, *Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations* , Advances in Neural Information Processing Systems 21, 2007.
10. Ashutosh Saxena, Justin Driemeyer, and Andrew Y. Ng, *Learning 3-D Object Orientation from Images*, NIPS 2007 workshop on Robotic Challenges for Machine Learning, 2007.
11. Ashutosh Saxena, Justin Driemeyer, and Andrew Y. Ng, *Robotic Grasping of Novel Objects using Vision*, International Journal of Robotics Research (IJRR), vol. 27, no. 2, pp. 157-173, Feb 2008.
12. Ashutosh Saxena, Lawson Wong, Morgan Quigley, and Andrew Y. Ng, *A Vision-based System for Grasping Novel Objects in Cluttered Environments*, International Symposium of Robotics Research (ISRR), 2007.
13. Ashwin Deshpande, Brian Milch, Luke S. Zettlemoyer, and Leslie Pack Kaelbling, *Learning Probabilistic Relational Dynamics for Multiple Tasks*, Conference on Uncertainty in Artificial Intelligence (UAI), July 2007.
14. Bethany Leffler and Michael Littman, *Transferrable Models for Autonomous Navigation*, Epigenetic Robotics, November, 2007.
15. Bethany R. Leffler, Timothy Edmunds, Michael L. Littman, *Efficient Reinforcement Learning with Relocatable Action Models*, AAAI, Summer 2007.

16. Bhaskara Marthi, Leslie Kaelbling, Tomas Lozano-Perez, *Learning Hierarchical Structure in Policies,* NIPS Workshop on Hierarchical Organization of Behavior: Computational, Psychological and Neural Perspectives , December 2007.
17. Bhaskara Marthi, Stuart Russell, and Jason Wolfe, *Angelic semantics for high-level actions*, Proceedings of the The International Conference on Automated Planning and Scheduling, Sept 2007.
18. Carlos Diuk and Michael L. Littman, *Hierarchical Reinforcement Learning*, Encyclopedia of Artificial Intelligence , December 2007.
19. David Sontag and Tommi Jaakkola, *New Outer Bounds on the Marginal Polytope,* Advances in Neural Information Processing Systems 21, 2007.
20. Han-Pang Chiu, Leslie Pack Kaelbling and Tomas Lozano-Perez, *Multi-view object class recognition*, Computer Vision and Pattern Recognition, 2007.
21. Hei Chan, Alan Fern, Soumya Ray, Nick Wilson, Chris Ventura, *Extending Online Planning for Resource Production in Real-Time Strategy Games with Search*, Workshop on Planning in Games at the International Conference on Automated Planning and Scheduling , September 2007.
22. Hei Chan, Alan Fern, Soumya Ray, Nick Wilson, Chris Ventura, *Online Planning for Resource Production in Real-Time Strategy Games,* International Conference on Automated Planning and Scheduling, 2007.
23. Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng, *Efficient Sparse Coding Algorithms*, NIPS, 2006.
24. Honglak Lee, Ekanadham Chaitanya and Andrew Y. Ng, *Sparse Deep Belief Net Model for Visual Area V2.*, Advances in Neural Information Processing Systems 21, 2007.
25. Jacob Abernethy and Peter L. Bartlett and Alexander Rakhlin, *Multitask learning with expert advice*, Proceedings of the Conference on Learning Theory, pages 484-498, June 2007.
26. Jacob Abernethy, Peter Bartlett, Alexander Rakhlin, *Multitask learning with expert advice,* Proceedings of the Conference on Learning Theory, June 2007.
27. Jyri Kivinen, Erik Sudderth and Michael Jordan, *Image denoising with nonparametric hidden Markov trees*, IEEE International Conference on Image Processing (ICIP), August, 2007.
28. Jyri Kivinen, Erik Sudderth and Michael Jordan, *Learning multiscale representations of natural scenes using Dirichlet processes*, IEEE International Conference on Computer Vision (ICCV), October, 2007.
29. Liang, P., Petrov, S., Klein, D., and Jordan, M. I., *The infinite PCFG using Dirichlet processes.*, Empirical Methods in Natural Language Processing (EMNLP), Summer, 2007.
30. Liang, P., Taskar, B., and Jordan, M. I., *A permutation-augmented sampler for DP mixture models.*, International Conference on Machine Learning (ICML), 2007.
31. Percy Liang, Dan Klein, and Michael Jordan, *Agreement-based learning,* Advances in Neural Information Processing (NIPS) 21, June 2008.
32. Peter L. Bartlett, Elad Hazan and Alexander Rakhlin, *Adaptive online gradient descent*, NIPS, December 2007.

33. Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer and Andrew Y. Ng, *Self-taught Learning: Transfer Learning from Unlabeled Data.*, ICML , June 2007.
34. Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer and Andrew Y. Ng, *Self-taught Learning*, NIPS 2006 Workshop on Learning when Test and Training Inputs Have Different Distributions, 2006.
35. Roger Grosse, Rajat Raina, Helen Kwong, Andrew Y. Ng, *Shift-invariant Sparse Coding for Audio Classification.*, Conference on Uncertainty in Artificial Intelligence (UAI), July 2007.
36. Ronald Parr, Christopher Painter-Wakefield, Lihong Li, Michael L. Littman, *Analyzing feature generation for value-function approximation*, ICML, Summer 2007.
37. Sarah Finney, Leslie Pack Kaelbling and Tomas Lozano-Perez, *Predicting Partial Paths from Planning Problem Parameters*, Robotics Science and Systems, June 2007.
38. Suin Lee, Vassil Chatalbashev, David Vickrey, Daphne Koller, *Learning a Meta-Level Prior for Feature Relevance from Multiple Related Tasks*, ICML , June 2007.
39. Tadepalli, Tom Dietterich , *Automatic Induction of MAXQ Hierarchies* , NIPS Workshop on Hierarchical Organization of Behavior: Computational, Psychological and Neural Perspectives , December 2007.
40. Thibaux, R., Jordan, M. I., *Hierarchical beta processes and the Indian buffet process.*, Eleventh Conference on Artificial Intelligence and Statistics (AISTATS'07), 2007.
41. Thomas J. Walsh and Michael L. Littman, *A Multiple Representation Approach To Learning Dynamical Systems*, The 2007 AAAI Fall Symposium: Computational Approaches to Representation Change During Learning and Development, November, 2007.
42. Thomas J. Walsh, Ali Nouri, Lihong Li, Michael L. Littman, *Planning and learning in environments with delayed feedback*, ECML, Fall 2007.
43. Yuehua Xu and Alan Fern, *On Learning Linear Ranking Functions for Beam Search*, International Conference on Machine Learning, June 2007.

2008

1. Aaron Wilson, Alan Fern, Soumya Ray, Prasad Tadepalli, *Learning and Transferring Roles in Multi-Agent Reinforcement Learning*, AAAI-08 Workshop on Transfer Learning for Complex Tasks, July, 2008.
2. Ashutosh Saxena, Lawson Wong, Andrew Y. Ng, *Learning grasp strategies with partial shape information*, AAAI, July 2008.
3. Asmuth, Littman, Zinkov, *Potential-based Shaping in Model-based Reinforcement Learning*, AAAI, 2008.
4. Benjaminn Sapp, Ashutosh Saxena, Andrew Y. Ng, *A Fast Data Collection and Augmentation Procedure for Object Recognition*,, AAAI, July 2008.
5. Brunskill, Leffler, Li, Littman, Roy, *PAC Reinforcement Learning in Noisy Continuous Worlds*, UAI, 2008.

6. D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, Y. Weiss, *Tightening LP relaxations for MAP using message passing*, Proceedings of the 24rd Conference on Uncertainty in Artificial Intelligence, July 2008.
7. David Sontag and Tommi Jaakkola, *LP Relaxations, Duals, and MAP*, Proceedings of the Twelfth International Conference, 2008.
8. David Sontag, Amir Globerson, Tommi Jaakkola, *Clusters and Coarse Partitions in LP Relaxations*, Advances in Neural Information Processing Systems, 2008, 2008.
9. Diuk, Cohen, Littman, *An Object-Oriented Representation for Efficient Reinforcement Learning*, ICML, 2008.
10. E. Fox, E. Sudderth, M. I. Jordan and A. S. Willsky, *Nonparametric Bayesian learning of switching linear dynamical systems.*, Advances in Neural Information Processing (NIPS), Winter, 2008.
11. E. Fox, E. Sudderth, M. I. Jordan and A. S. Willsky, *An HDP-HMM for systems with state persistence*, International Conference on Machine Learning (ICML), Summer, 2008.
12. E. Sudderth and M. I. Jordan, *Shared segmentation of natural scenes using dependent Pitman-Yor processes.* , Advances in Neural Information Processing (NIPS), Winter, 2008.
13. Ellen Klingbeil, Ashutosh Saxena, Andrew Y. Ng, *Learning to Open New Doors*, Robotics Science and Systems (RSS) workshop on Manipulation in Human environments , June 25-28, 2008.
14. G. Obozinski, M. Wainwright, and M. I. Jordan, *High-dimensional union support recovery in multivariate regression.*, Advances in Neural Information Processing (NIPS), Winter, 2008.
15. Gal Elidan, Ben Packer, Geremy Heitz, Daphne Koller, *Convex Point Estimation using Undirected Bayesian Transfer Hierarchies*, UAI, July 2008.
16. Gal Elidan, Geremy Heitz, Ben Packer, Daphne Koller, *LOOPS: Localizing Object Outlines using Probabilistic Shape*, NIPS, December 2008.
17. Geremy Heitz, Stephen Gould, Ashutosh Saxena, Daphne Koller, *Cascaded Classification Models: Combining Models for Holistic Scene Understanding*, NIPS 2008, December, 2008.
18. Honglak Lee, Rajat Raina, Alex Teichman, Andrew Y. Ng, *Exponential family sparse coding with application to self-taught learning with text documents* , ICML Workshop on Prior Knowledge for Text and Language , July 9, 2008.
19. Honglak Lee, Roger Grosse, Rajesh Ranganath, Andrew Y. Ng, *Translation-invariant sparse deep belief networks for scalable unsupervised learning of hierarchical representation*, NIPS 2008 Workshop on Machine Learning Meets Human Learning, Dec 2008.
20. Jacob Abernethy, Manfred Warmuth and Joel Yellin, *Optimal Strategies from Random Walks*, COLT, July 2008.
21. Jacob Abernethy, Olivier Chapelle and Carlos Castillo, *Webspam Identification Through Content and Hyperlinks*, AirWeb 2008 (Adversarial Information Retrieval on the Web), April 2008.

22. Jacob Abernethy, Peter L. Bartlett, Alexander Rakhlin and Ambuj Tewari, *Optimal strategies and minimax lower bounds for online convex games*, COLT, February 2008.
23. Jake Abernethy, Elad Hazan, Alexander Rakhlin, *Competing in the Dark: An Efficient Algorithm for Bandit Linear Optimization*, COLT, February 2008.
24. K. Hsiao, T. Lozano-Perez and L. Kaelbling, *Robust Belief-Based Execution of Manipulation Programs*, WAFR 08, December 2008.
25. K. Miller, T. Griffiths and M. I. Jordan , *The phylogenetic Indian buffet process*, Uncertainty in Artificial Intelligence (UAI), Summer, 2008.
26. Leffler, Mansley, Littman, *Efficient Learning of Dynamics Models using Terrain Classification*, International Workshop on Evolutionary and Reinforcement Learning for Autonomous Robot Systems, 2008.
27. Li, Littman, Walsh, *Knows What It Knows: A Framework For Self-Aware Learning*, European Workshop on Reinforcement Learning, July 2008.
28. Li, Littman, Walsh, *Knows What It Knows: A Framework For Self-Aware Learning*, ICML, 2008.
29. Li, Littman, *Efficient Exploration with Linear Value-Function Approximation*, UAI, 2008.
30. Li, *A Worst-Case Comparison Between Temporal Difference and Residual Gradient with Linear Function Approximation*, ICML, 2008.
31. Mike Wynkoop and Tom Dietterich , *Learning MDP Action Models Via Discrete Mixture Trees*, Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science Volume 5212, 2008.
32. Munoz, Littman, *A Polynomial-time Nash Equilibrium Algorithm for Repeated Stochastic Games* , UAI, 2008.
33. Neville Mehta, Soumya Ray, Prasad Tadepalli, Tom Dietterich, *Automatic Discovery and Transfer of MAXQ Hierarchies*, International Conference on Machine Learning, July, 2008.
34. Nouri, Littman, *Multi-resolution Exploration in Continuous Spaces*, NIPS, December 2008.
35. Parr, Li, Taylor, Painter-Wakefield, Littman, *An Analysis of Linear Models, Linear Value-Function Approximation, and Feature Selection for Reinforcement Learning*, ICML, 2008.
36. Peter L. Bartlett, Varsha Dani, Thomas P. Hayes, Sham M. Kakade, Alexander Rakhlin and Ambuj Tewari, *High-probability regret bounds for bandit online linear optimization*, COLT, July 2008.
37. Pradeep Ravikumar, Alekh Agarwal and Martin Wainwright, *Message-passing for Graph-structured Linear Programs: Proximal Projections, Convergence and Rounding Schemes*, ICML, July 2008.
38. Rajat Raina, Andrew Y. Ng, *Learning Large Deep Belief Networks using Graphics Processors*, NIPS 2008 Workshop on Parallel Implementations of Learning Algorithms , Dec 2008.
39. Roberts, Isbell, Littman, *Optimization Problems Involving Collections of Dependent Objects*, Annals of Operations Research, July 2008.
40. S. Seshia, A. Rakhlin, *Game-Theoretic Timing Analysis*, ICCAD, July 2008.

41. Strehl, Littman, *An Analysis of Model-Based Interval Estimation for Markov Decision Processes,* JCCS, 2008.
42. Sungwook Yoon, Alan Fern, Robert Givan, Subbarao Kambhampati, *Probabilistic Planning via Determinization in Hindsight*, AAAI Conference on Artificial Intelligence, July, 2008.
43. Sungwook Yoon, Alan Fern, Robert Givan, *Learning Control Knowledge for Forward Search Planning*, Journal of Machine Learning Research, 2008.
44. Walsh, Nouri, Li, Littman, *Planning and Learning in Environments with Delayed Feedback*, Journal of Agents and Autonomous Multiagent Systems, July 2008.

2009

1. Ashutosh Saxena, Justin Driemeyer and Andrew Y. Ng, *Learning 3-D Object Orientation from Images*, International Conference on Robotics and Automation (ICRA), May 2009.
2. Carlos Diuk, Lihong Li and Bethany Leffler, *The Adaptive k-Meteorologists Problem and Its Application to Structure Learning and Feature Selection in Reinforcement Learning*, International Conference on Machine Learning (ICML), June 2009.
3. Emily Fox, Erik Sudderth, Michael I. Jordan and Alan Willsky, *Nonparametric Bayesian learning of switching linear dynamical systems*, Neural Information Processing Systems (NIPS), May 2009.
4. Erik Sudderth and Michael I. Jordan , *Shared segmentation of natural scenes using dependent Pitman-Yor processes*, Neural Information Processing Systems (NIPS), May 2009.
5. Gal Elidan, Geremy Heitz, Ben Packer, Daphne Koller, *Shape-Based Object Localization for Descriptive Classification*, IJCV 2009.
6. Guillaume Obozinski, Martin Wainwright and Michael I. Jordan , *High-dimensional union support recovery in multivariate regression*, Neural Information Processing Systems (NIPS), May 2009.
7. Honglak Lee, Rajat Raina, Alex Teichman and Andrew Y. Ng, *Exponential Family Sparse Coding with Application to Self-taught Learning*, International Joint Conference on Artificial on Artificial Intelligence and Statistics , July 2009.
8. Honglak Lee, Roger Grosse, Rajesh Ranganath and Andrew Ng, *Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations* , Proceedings of the 26th International Conference on Machine Learning, June 2009.
9. Honglak Lee, Roger Grosse, Rajesh Ranganath and Andrew Y. Ng, *Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations*, International Conference on Machine Learning (ICML), June 2009 .
10. Jacob Abernethy, Alekh Agarwal, Peter L. Bartlett, Alexander Rakhlin, *A Stochastic View of Optimal Regret through Minimax Duality*, arXiv.org, March 30, 2009.
11. Kaijen Hsiao, Paul Nangeroni, Manfred Huber, Ashutosh Saxena and Andrew Y. Ng, *Reactive Grasping using Optical Proximity Sensors*, International Conference on Robotics and Automation (ICRA), May 2009.

12. Li, Littman, Mansley, *Online Exploration in Least-Squares Policy Iteration*, 2009 AAMAS, 2009.
13. Littman, *A Tutorial on Partially Observable Markov Decision Processes*, Journal of Mathematical Psychology, June 2009.
14. Michael Littman, Lihong Li, Ali Nouri, David Wingate, John Asmuth, *A Bayesian Sampling Approach to Exploration in Reinforcement Learning*, Uncertainty in Artificial Intelligence (UAI), June 2009.
15. Neville Mehta and Prasad Tadepalli, *Learning Partial Models for Hierarchical Planning.*, IJCAI 2009 Workshop on Learning Structural Knowledge From Observations, July 2009 .
16. Percy Liang, Dan Klein, and Michael I. Jordan, *Learning from measurements in exponential families*, International Conference on Machine Learning (ICML), June 2009.
17. Peter L. Bartlett, Ambuj Tewari, *REGAL: A Regularization based Algorithm for Reinforcement Learning in Weakly Communicating MDPs*, 25th Conference on Uncertainty, 2009.
18. Radha-Krishna Balla and Alan Fern, *UCT for Tactical Assault Planning in Real-Time Strategy Games*, IJCAI, 2009.
19. Rajat Raina, Anand Madhavan and Andrew Ng, *Large-scale Deep Unsupervised Learning using Graphics Processors*, Proceedings of the 26th International Conference on Machine Learning, June 2009.
20. Scott Proper and Prasad Tadepalli , *Multiagent Transfer Learning via Assignment-based Decomposition*, Proceedings of the International Conference on Machine Learning and Applications, 2009.
21. Scott Proper and Prasad Tadepalli , *Solving Multiagent Assignment Markov Decision Processes* , Proceedings of the Conference on Autonous Agents and Multiagnt Systems, May 10.
22. Scott Proper and Prasad Tadepalli , *Transfer Learning via Relational Templates*, Proceedings of the 19th International Joint Conference on Artificial Intelligence, July 2009.
23. Stephen Gould, Fernando Amat, Daphne Koller, *Alphabet SOUP: A Framework for Approximate Energy Minimization*, CVPR 2009, June 2009.
24. Thomas Walsh, Istvan Szita, Carlos Diuk, Michael Littman, *Exploring compact reinforcement-learning representations with linear regression*, Uncertainty in Artificial Intelligence (UAI), June 2009.

Glossary

An excellent glossary of terms in Transfer Learning can be found at:
http://alumni.media.mit.edu/~tpminka/statlearn/glossary/

Statistical Learning/Pattern Recognition - An approach to machine intelligence which is based on statistical modeling of data. With a statistical model in hand, one applies probability theory and decision theory to get an algorithm. This is opposed to using training data merely to select among different algorithms or using heuristics/"common sense" to design an algorithm.

Features - The measurements which represent the data. The statistical model one uses is crucially dependent on the choice of features. Hence it is useful to consider alternative representations of the same measurements (i.e. different features). For example, different representations of the color values in an image. General techniques for finding new representations include discriminant analysis, principal component analysis, and clustering.

Classification - Assigning a class to a measurement, or equivalently, identifying the probabilistic source of a measurement. The only statistical model that is needed is the conditional model of the class variable given the measurement. This conditional model can be obtained from a joint model or it can be learned directly. The former approach is **generative** since it models the measurements in each class. It is more work, but it can exploit more prior knowledge, needs less data, is more modular, and can handle missing or corrupted data. Methods include mixture models and Hidden Markov Models. The latter approach is **discriminative** since it focuses only on discriminating one class from another. It can be more efficient once trained and requires fewer modeling assumptions. Methods include logistic regression, generalized linear classifiers, and nearest-neighbor.

Reinforcement Learning - Learning how to act optimally in a given environment, especially with delayed and nondeterministic rewards. It is equivalent to **adaptive control**. There are two interleaved tasks: modeling the environment and making optimal decisions based on the model. The first task is a statistical modeling problem (see URL above.) The second task is a decision theory problem: converting the expectation of delayed reward into an immediate action. Since reinforcement learning requires exploration, it is often combined with active learning, though this is not essential. Most learning problems that humans face are reinforcement learning problems, e.g. deciding which melon to buy, which coat to wear outside today, or which friends to have.

Appendix A: Year 1 Go/NoGo results and scientific summary

TL Y1 Internal Evaluation Summary

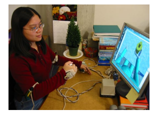
Simulated Robot Manipulation

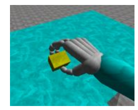
Leslie Pack Kaelbling
Tomas Lozano-Perez
Kaijen Hsiao

MIT CSAIL

Problem Statement

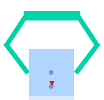
- We wish to enable a simulated robot to learn grasps by imitation
- A human demonstrates picking up 5 simulated objects
- The robot practices those grasp types on simple objects
- The robot performs the same grasp types on new objects that are different than the training objects, thus exhibiting transfer





Domain Performance Metric(s) & Goal(s)

TL	Time	Goal	RMS Err	Goal	Pct Success	Goal
1	80.60 sec	900 sec	1.52 cm	2cm	96.0%	Not specified
2	82.84 sec	900 sec	1.75 cm	2cm	95.5%	Not specified
3	91.08 sec	900 sec	1.95 cm	2cm	94.3%	Not specified



RMS error metric:
- Defined on objects that were grasped correctly
- Measured degree of object slippage during grasp stabilization

Evaluation Analysis Summary

Evaluation Type: Internal
Client: UCB/MIT
Domain: Simulated robot manipulation

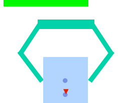
TL	TL Metric Goals Met?	Discussion
1	✓	TRS = infty
2	✓	TRS = infty
3	✓	TRS = infty TRS = 3.3632

Year 1 goal: Transfer ratio > 10
TRSx: Transfer ratio (smoothed) at #B = x

Level 3 experiment considered two different "source" objects: one proved to be a good "universal donor" of experience; the other did not.

Domain Performance Metric(s) & Goal(s)

TL	Time	Goal	RMS Err	Goal	Pct Success	Goal
1	80.60 sec	900 sec	1.52 cm	2cm	96.0%	Not specified
2	82.84 sec	900 sec	1.75 cm	2cm	95.5%	Not specified
3	91.08 sec	900 sec	1.95 cm	2cm	94.3%	Not specified
4 optional	91.08 sec	900 sec	1.79 cm	2cm	92.9%	Not specified



RMS error metric:
- Defined on objects that were grasped correctly
- Measured degree of object slippage during grasp stabilization

Evaluation Analysis Summary

Evaluation Type: Internal
Client: UCB/MIT
Domain: Simulated robot manipulation

TL	TL Metric Goals Met?	Discussion
1	✓	TRS = infty
2	✓	TRS = infty
3	✓	TRS = infty TRS = 3.3632
4 optional	✓	TRS = 13.7701

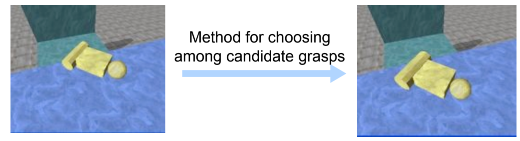
Year 1 goal: Transfer ratio > 10
TRSx: Transfer ratio (smoothed) at #B = x

Level 3 experiment considered two different "source" objects: one proved to be a good "universal donor" of experience; the other did not.

Experimental protocol summary

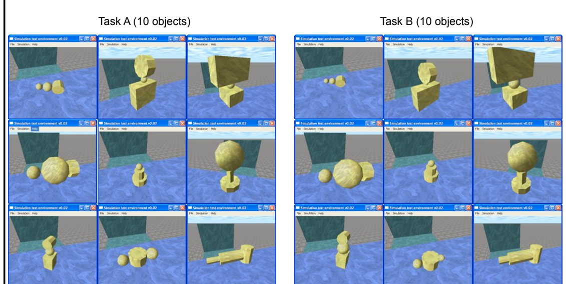
Level	Task A	Task B	Replications	Task A size	B train size	Test interval	Test set size	Objects
1	Object at fixed pose	Same object, orientation; different positions	10	5	12	1	5	10
2	Object at fixed pose	Same object; different positions and orientations	10	5	12	1	5	10
3	Object at varying poses	Other objects from same class at varying poses	10	50	10	1-5	100	2
4 optional	Objects from one class at varying poses	Objects from different class at varying poses	10	50	100	1-5	100	1

Transfer Level 1
Varying object position

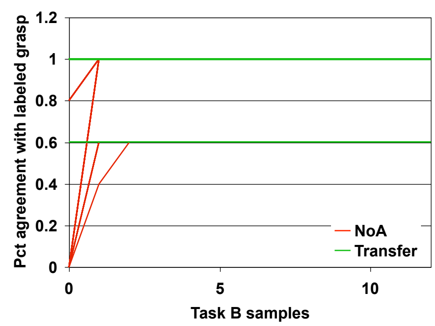


Task A: Grasping one particular complex object in a single position and orientation
Task B: Grasping the same object in the same orientation, but different positions
Transferred knowledge:
- Examples for nearest-neighbor grasp type selection
- Quality metric for grasps

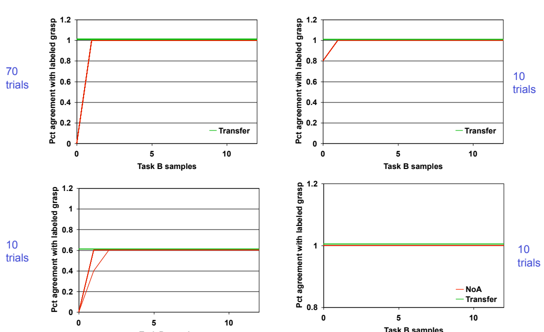
TL1 Objects
Varying object position



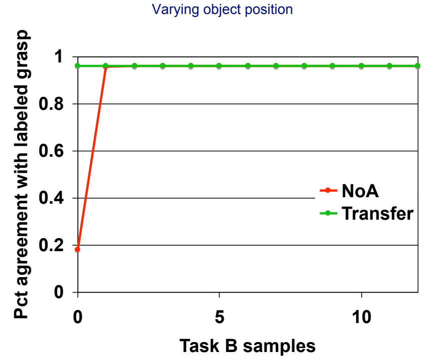
TL1 All Raw Curves
Varying object position



TL1 Raw Curves
Varying object position



TL1 Average Curves
Varying object position



TL1 Statistics
Varying object position

Metric	Score	P Value
Transfer ratio (smoothed)	**infty**	**0.0000**
Transfer ratio (max asymp)	infty	0.0000
Truncated transfer ratio	infty	0.0000
Average relative reduction	0.5013	0.2670
ARR (narrow)	0.0000	0.5044
Asymptotic advantage	0.0000	0.4782
Jump start	0.7800	0.0000
Ratio	1.035	0.0240
Transfer difference	0.3920	0.0272
Scaled transfer difference	0.4083	0.0258

TL1 Notes
Varying object position

- For each of 10 objects
  - Set A: slightly varying sizes in same position
  - Set B: same object in different positions (same orientation)
- Results for all objects analyzed jointly
- "Degenerate" experiment, because internal representation is designed to be position invariant
- No significant asymptotic advantage: noA quickly learns to perform as well as transfer
- No significant average relative reduction: noA's performance is only briefly below that of transfer
- Narrow ARR is essentially undefined

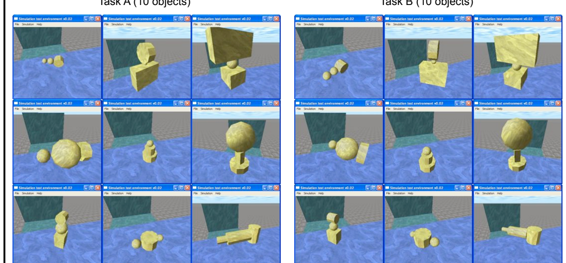
Transfer Level 2
Varying object orientation

Methods for choosing among candidate grasps

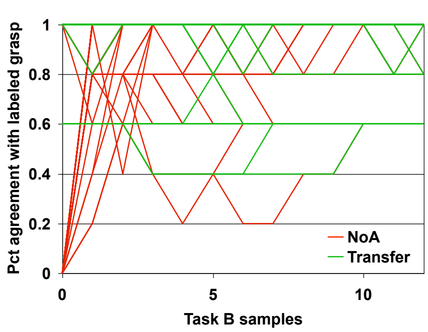
Task A: Grasping one particular complex object
Task B: Grasping that same object in arbitrary positions and orientations
Transferred knowledge:
- Instances for nearest neighbor grasp type selection
- Quality metric for grasps

TL2 Objects
Varying object orientation

Task A (10 objects) Task B (10 objects)

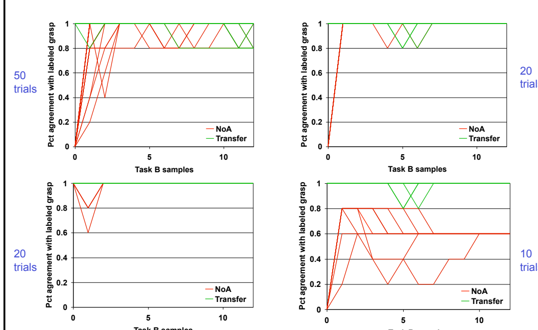


TL2 All Raw Curves
Varying object orientation



Discrete performance levels due to test sets of size 5

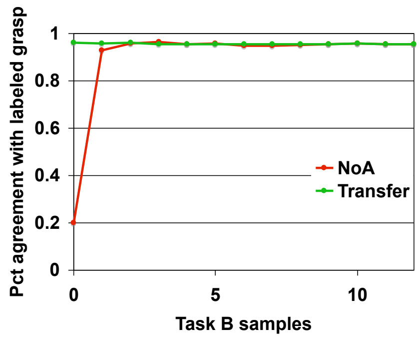
TL2 Raw Curves
Varying object orientation



Discrete performance levels due to test sets of size 5

TL2 Average Curves

Varying object orientation



TL2 Statistics

Varying object orientation

Metric	Score	P Value
Transfer ratio (smoothed)	**infty**	**0.0000**
Transfer ratio (max asymp)	5.5162	0.0000
Truncated transfer ratio	17.8330	0.0000
Average relative reduction	- infty	0.2852
ARR (narrow)	0.0000	0.5052
Asymptotic advantage	-0.0040	0.5996
Jump start	0.7600	0.0000
Ratio	1.0380	0.0228
Transfer difference	0.4200	0.0206
Scaled transfer difference	0.4357	0.0200

TL2 Notes

Varying object orientation

- For each of 10 objects
  - Set A: varying position, same orientation
  - Set B: same object in different orientations
- Results for all objects analyzed jointly
- Internal representation is designed to be orientation invariant, but relationship of object to robot and table affects grasp quality
- No significant asymptotic advantage: noA quickly learns to perform as well as transfer
- No significant average relative reduction: noA's performance is only briefly below that of transfer; narrow ARR undefined

Transfer Level 3

Varying object shape within class

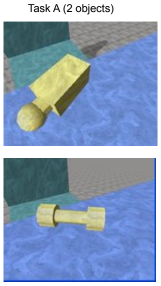
Methods for choosing among candidate grasps

Task A: Grasping one particular complex object
Task B: Grasping other complex objects in arbitrary positions and orientations
Transferred knowledge:
- Instances for nearest neighbor grasp type selection
- Quality metric for grasps

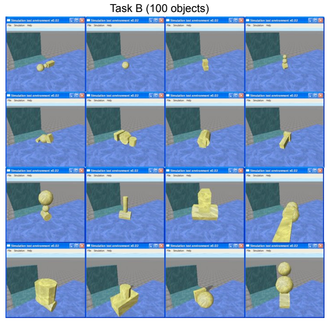
TL3 Objects

Varying object shape within class

Task A (2 objects)



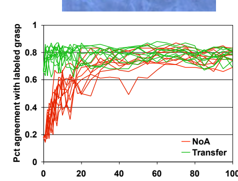
Task B (100 objects)



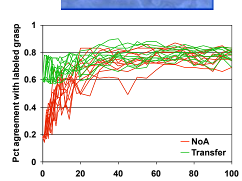
TL3 All Raw Curves

Varying object shape within class

Domain A Object 1

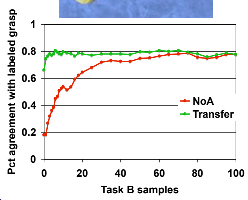


Domain A Object 2

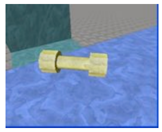


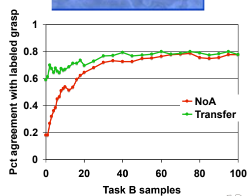
TL3 Average Curves
Varying object shape within class

Domain A Object 1



Domain A Object 2





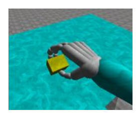
TL3 Statistics
Varying object shape within class

	A object 1		A object 2	
Metric	Score	P Value	Score	P Value
Transfer ratio (smoothed)	**infty**	**0.0000**	**3.3632**	**0.0014**
Transfer ratio (max asymp)	4.7836	0.0000	2.5343	0.0000
Truncated transfer ratio	12.1240	0.0000	2.5454	0.0004
Average relative reduction	0.9938	0.0000	0.9212	0.0014
ARR (narrow)	0.9772	0.0006	0.7580	0.0018
Asymptotic advantage	0.0220	0.1134	0.0150	0.2062
Jump start	0.4800	0.0000	0.4100	0.0000
Ratio	1.1318	0.0000	1.0947	0.0002
Transfer difference	9.1260	0.0000	6.5615	0.0006
Scaled transfer difference	11.2946	0.0000	8.1916	0.0004

TL3 Notes
Varying object shape within class

- No significant improvement in asymptotic advantage in either case
- Transfer ratio considerably higher when object 1 is used as the A domain: the boxy shape applies more broadly to other objects than the barbell (which encourages grasps that don't work well on other objects)

Transfer Level 4
Varying object class

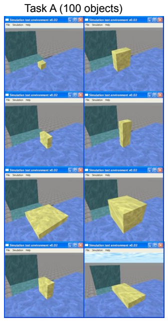


Methods for choosing among candidate grasps

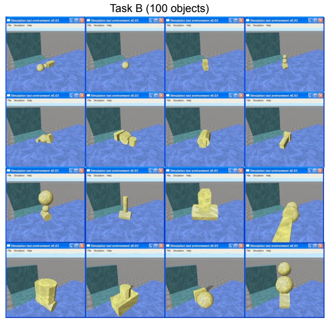
Task A: Grasping boxes
Task B: Grasping more complex objects in arbitrary positions and orientations
Transferred knowledge:
- Instances for nearest neighbor grasp type selection
- Quality metric for grasps

TL4 Objects
Varying object class

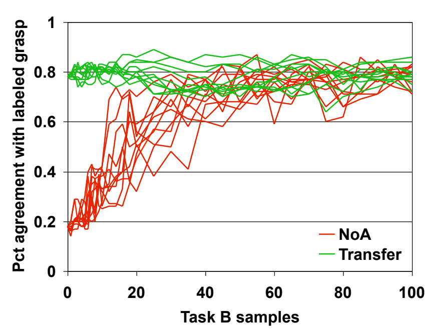
Task A (100 objects)



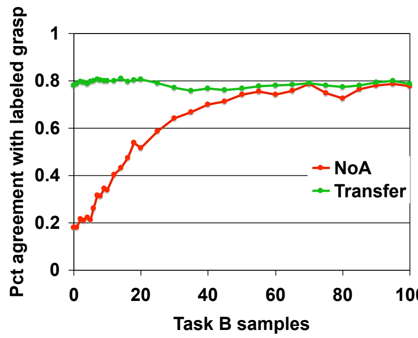
Task B (100 objects)



TL4 All Raw Curves
Varying object class



TL4 Average Curves
Varying object class



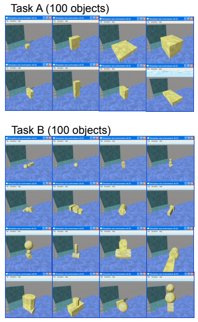
TL4 Statistics
Varying object class

Metric	Score	P Value
Transfer ratio (smoothed)	**13.7701**	**0.0002**
Transfer ratio (max asymp)	6.3627	0.0000
Truncated transfer ratio	47.1485	0.0000
Average relative reduction	0.9752	0.0018
ARR (narrow)	0.9907	0.0020
Asymptotic advantage	0.0210	0.0418
Jump start	0.6000	0.0000
Ratio	1.2092	0.0000
Transfer difference	13.5595	0.0000
Scaled transfer difference	16.7608	0.0000

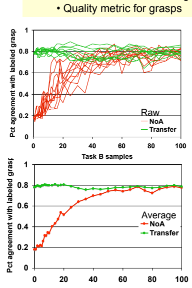
Transfer Level 4: Summary
Varying object class

Task A (100 objects)

Task B (100 objects)

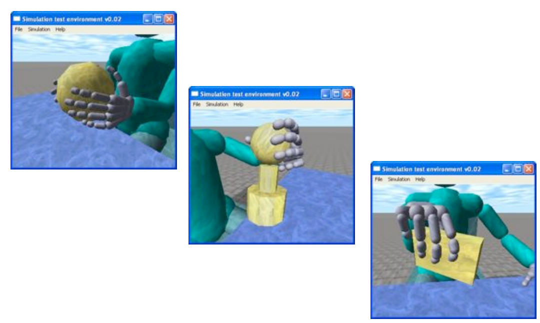


Task A: Grasping boxes
Task B: Grasping more complex objects in arbitrary positions and orientations
Transferred knowledge:
- Instances for nearest neighbor grasp type selection
- Quality metric for grasps

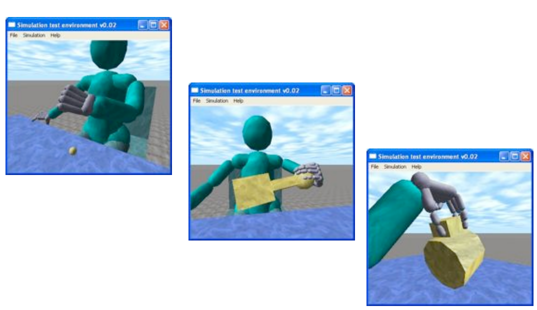


Metric	Score	P Value
Transfer ratio (smoothed)	13.7701	0.0002
Transfer ratio (max asymptote)	6.3627	0.0000
Truncated transfer ratio	47.1485	0.0000
Average relative reduction	0.9752	0.0018
ARR (narrow)	0.9907	0.0020
Asymptotic advantage	0.0210	0.0418
Jump start	0.6000	0.0000
Ratio	1.2092	0.0000
Transfer difference	13.5595	0.0000
Scaled transfer difference	16.7608	0.0000

Some Successful Grasps



A Few Failed Grasps



TL Y1 Internal Evaluation Summary

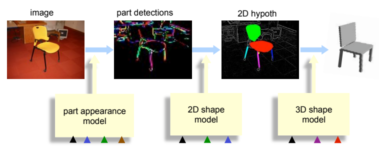
Object Recognition

Leslie Pack Kaelbling
Tomas Lozano-Perez
Han-Pang Chiu
Sam Davies

MIT CSAIL

Problem Statement

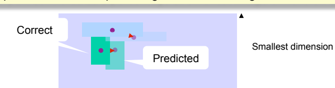
- We wish to enable a computer vision system to learn to recognize structured objects
- The vision system is trained on images with the objects and their parts labeled
- The system recognizes related objects in related situations, exhibiting transfer by doing so more quickly than it would otherwise have been able to



Domain Performance Metric(s) & Goal(s)

TL	Time	Goal*	RMS Err %	Goal**
1	~300 sec	~7296 sec	15.95%	30% rel err
2	~300 sec	~7296 sec	12.69%	30% rel err
3	~300 sec	~7296 sec	16.12%	30% rel err

*1000 seconds to process an image of complexity 1000
Complexity = #models * #parts * # image complexity (edge segs)= 1*6*1216 = 7296 (avg)
**Error of predicted part centroids as a percentage of smallest image dimension (300 pixels)



Evaluation Analysis Summary

Evaluation Type: Internal
Client: UCB/MIT
Domain: Object recognition

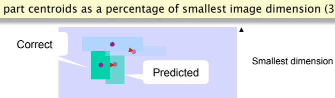
TL	TL Metric Goals Met?	Discussion
1	✓	TRS6 = infty
2	✓	TRS10 = 33.7929
3	✓	TRS10 = infty TRS50 = infty

Year 1 goal: Transfer ratio > 10
TRSx: Transfer ratio (smoothed) at #B = x

Domain Performance Metric(s) & Goal(s)

TL	Time	Goal*	RMS Err %	Goal**
1	~300 sec	~7296 sec	15.95%	30% rel err
2	~300 sec	~7296 sec	12.69%	30% rel err
3	~300 sec	~7296 sec	16.12%	30% rel err
2/3/5 optional	~300 sec	~7296 sec	12.79%	30% rel err
5 optional	~300 sec	~7296 sec	13.83%	30% rel err

*1000 seconds to process an image of complexity 1000
Complexity = #models * #parts * # image complexity (edge segs)= 1*6*1216 = 7296 (avg)
**Error of predicted part centroids as a percentage of smallest image dimension (300 pixels)



Evaluation Analysis Summary

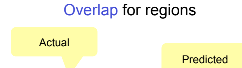
Evaluation Type: Internal
Client: UCB/MIT
Domain: Object recognition

TL	TL Metric Goals Met?	Discussion	
1	✓	TRS6 = infty	
2	✓	TRS10 = 33.7929	
3	✓	TRS10 = infty TRS50 = infty	
2/3/5 optional	✓	TRS10 = 30.4089 TRS50 = 5.1530	chairs
5 optional	✓	TRS10 = infty TRS50 = 349.3649	chairs

Year 1 goal: Transfer ratio > 10
TRSx: Transfer ratio (smoothed) at #B = x

Error Metrics for Learning

Overlap for regions



Overlap = Intersection / Union

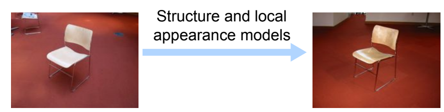
Experimental protocol summary

Level	Task A	Task B	Replications	Task A size	B train size	Test interval	Test set size
1	Single object, single view, single position	Same object, view; **different positions**	10	10	10	1	10
2	Single object class, single view, various positions	Same object class; **different single view**, various positions	25	20	10	1	30
3	Single object, single view, various positions	Other objects from **containing class** at same view, various positions	10	20	50	1-10	30
2/3/5	Synthetic images from two views, real images from one view	Real images from same class at **second view**, various positions	15	50	50	1-10	30
5	Synthetic objects from one class at various views and positions	**Real images** from same class at various view and positions	10	150	50	1-10	50

Some values vary from original specifications

Transfer Level 1

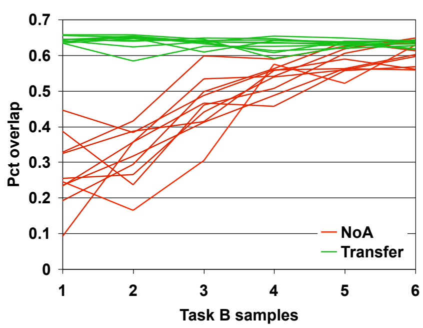
Varying position



Task A: Recognizing a narrow class of objects at one image location
Task B: Recognizing that same class of objects at other locations
Transferred knowledge:
- Structure and local appearance models for object

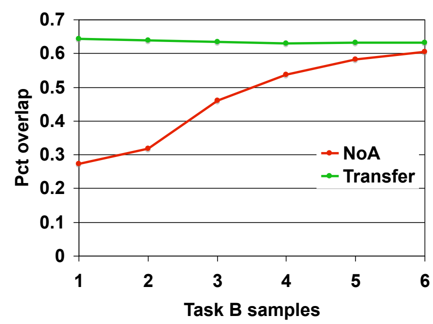
TL1 Raw Curves

Varying position



TL1 Average Curves

Varying position



TL1 Statistics

Varying position

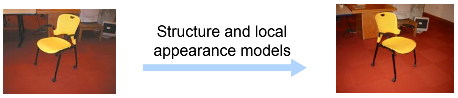
Metric	Score	P Value
Transfer ratio (smoothed)	**infty**	**0.0000**
Transfer ratio (max asymp)	20.3021	0.0006
Truncated transfer ratio	163.3375	0.0002
Average relative reduction	0.9900	0.0004
ARR narrow	0.0000	0.6888
Asymptotic advantage	0.0376	0.0000
Jump start	0.2878	0.0000
Ratio	1.1581	0.0000
Transfer difference	0.7251	0.0000
Scaled transfer difference	1.1662	0.0000

TL1 Notes

Varying position

- There was actually a small amount of variation in the viewpoints of the training images
- ARR is ill-defined for this curve

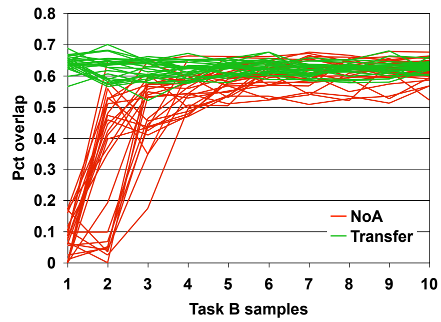
Transfer Level 2
Varying viewpoint



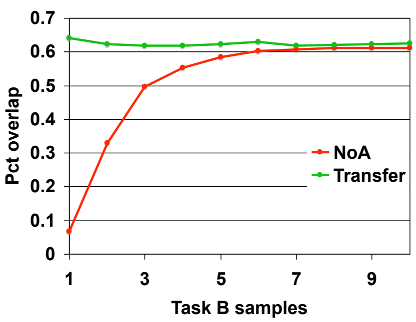
Task A: Recognizing a class of objects at from one viewpoint
Task B: Recognizing that same class of objects at a different viewpoint
Transferred knowledge:
- Structure and local appearance models or object

Built-in knowledge:
- Known transform between views

TL2 Raw Curves
Varying viewpoint



TL2 Average Curves
Varying viewpoint



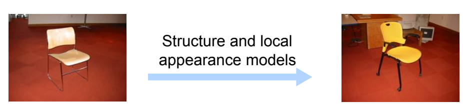
TL2 Statistics
Varying viewpoint

Metric	Score	P Value
Transfer ratio (smoothed)	**35.7929**	**0.0000**
Transfer ratio (max asymp)	6.5887	0.0000
Truncated transfer ratio	51.2116	0.0000
Average relative reduction	1.0000	0.0000
ARR Narrow	0.0000	0.6822
Asymptotic advantage	0.0279	0.0000
Jump start	.5728	0.0000
Ratio	1.1837	0.0000
Transfer difference	0.8710	0.0000
Scaled transfer difference	1.3591	0.0000

TL2 Notes
Varying viewpoint

- ARR narrow is ill-defined because the initial point on the transfer curve is also the max
- This works well because we have built knowledge of the transformation between the two views into the system.
- In TL2/3/5, we learn the transformation from synthetic data.
- In the future, we will learn it from real, weakly labeled data.

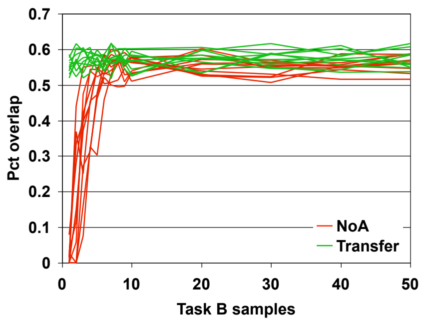
Transfer Level 3
Varying shape within class



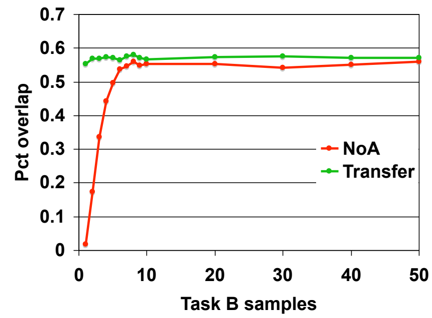
Task A: Recognizing a narrow class of objects at one orientation
Task B: Recognizing a broader class of objects at that same viewpoint
Transferred knowledge:
- Structure and local appearance models of object

Built-in knowledge:
- Object representation should make all elements of the class similar

TL3 Raw Curves
Varying shape within class



TL3 Average Curves
Varying shape within class



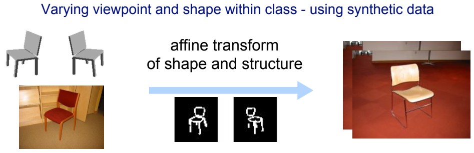
TL3 Statistics
Varying shape within class

Metric	Score	P Value
Transfer ratio (smoothed)	**infty**	**0.0000**
Transfer ratio (max asymp)	6.5689	0.0000
Truncated transfer ratio	17.9033	0.0004
Average relative reduction	0.9980	0.0000
ARR Narrow	0.8580	0.0000
Asymptotic advantage	0.0190	0.0078
Jump start	0.5362	0.0000
Ratio	1.0783	0.0000
Transfer difference	2.0356	0.0000
Scaled transfer difference	3.5137	0.0000

TL3 Statistics at #B=10
Varying shape within class

Metric	Score	P Value
Transfer ratio (smoothed)	**infty**	**0.0000**
Transfer ratio (max asymp)	15.9540	0.0000
Truncated transfer ratio	17.9033	0.0004
Average relative reduction	0.9971	0.0000
ARR Narrow	0.8484	0.0000
Asymptotic advantage	0.0204	0.0276
Jump start	0.5362	0.0000
Ratio	1.3068	0.0000
Transfer difference	1.2053	0.0000
Scaled transfer difference	2.0805	0.0000

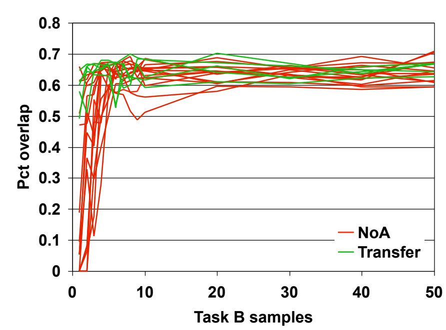
Transfer Level 2/3/5
Varying viewpoint and shape within class - using synthetic data



Task A: Recognizing a broad class of objects at one orientation; given synthetic data of two views
Task B: Recognizing the same class of objects at a different viewpoint
Transferred knowledge:
- Transformation between views

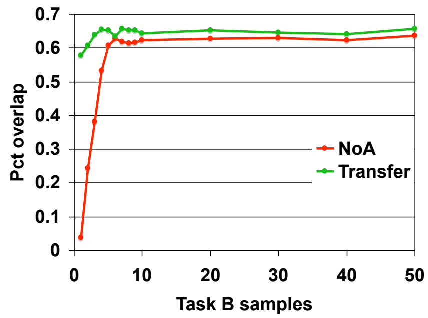
Built-in knowledge:
- Labels of synthetic images according to view

TL2/3/5 Raw Curves
Varying viewpoint and shape within class - using synthetic data



TL2/3/5 Average Curves

Varying viewpoint and shape within class - using synthetic data



TL2/3/5 Statistics

Varying viewpoint and shape within class - using synthetic data

Metric	Score	P Value
Transfer ratio (smoothed)	**5.1530**	**0.0000**
Transfer ratio (max asymp)	4.7440	0.0000
Truncated transfer ratio	9.0671	0.0000
Average relative reduction	0.9617	0.0000
ARR Narrow	0.6781	0.0004
Asymptotic advantage	0.0207	0.0438
Jump start	0.5400	0.0000
Ratio	1.0670	0.0006
Transfer difference	1.9854	0.0000
Scaled transfer difference	3.0217	0.0004

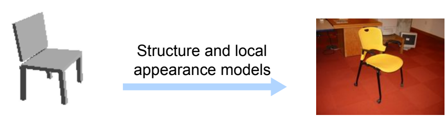
TL2/3/5 Statistics at #B=10

Varying viewpoint and shape within class - using synthetic data

Metric	Score	P Value
Transfer ratio (smoothed)	**30.4089**	**0.0020**
Transfer ratio (max asymp)	8.7922	0.0000
Truncated transfer ratio	9.0671	0.0000
Average relative reduction	0.6410	0.0000
ARR Narrow	0.9612	0.0004
Asymptotic advantage	0.0287	0.0210
Jump start	0.5400	0.0000
Ratio	1.2594	0.0006
Transfer difference	1.1868	0.0000
Scaled transfer difference	1.8062	0.0000

Transfer Level 5

Synthetic to real

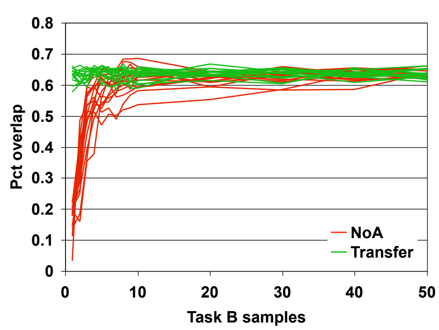


Task A: Recognizing a broad class of objects from synthetic images
Task B: Recognizing the same class of objects from real images
Transferred knowledge:
- Structure and local appearance models

Built-in knowledge:
- Edges have similar information in synthetic and real images

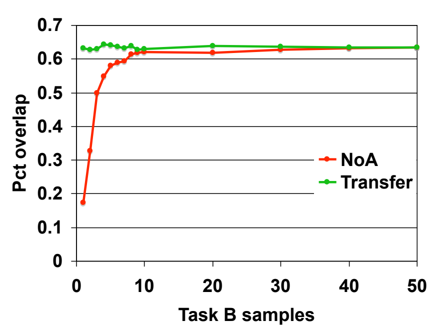
TL5 Raw Curves

Synthetic to real



TL5 Average Curves

Synthetic to real



TL5 Chair Statistics

Synthetic to real

Metric	Score	P Value
Transfer ratio (smoothed)	**349.3649**	**0.0000**
Transfer ratio (max asymp)	4.0892	0.0004
Truncated transfer ratio	22.5331	0.0000
Average relative reduction	0.9996	0.0000
ARR Narrow	0.9293	0.0000
Asymptotic advantage	0.0089	0.0268
Jump start	0.4567	0.0000
Ratio	1.0428	0.0000
Transfer difference	1.2784	0.0000
Scaled transfer difference	1.9866	0.0000

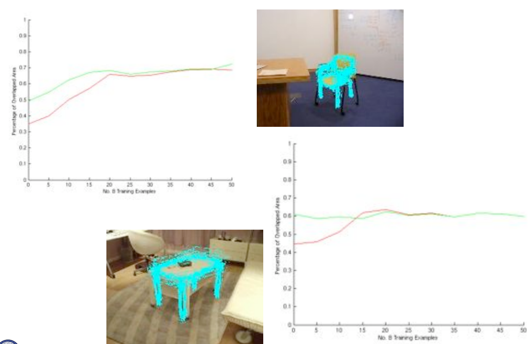
TL5 Chair Statistics at #B=10

Synthetic to real

Metric	Score	P Value
Transfer ratio (smoothed)	**infty**	**0.0000**
Transfer ratio (max asymp)	12.1900	0.004
Truncated transfer ratio	22.5331	0.0000
Average relative reduction	1.0000	0.0000
ARR Narrow	0.0000	0.6410
Asymptotic advantage	0.0218	0.0316
Jump start	0.4567	0.0000
Ratio	1.0428	0.0000
Transfer difference	0.9366	0.0000
Scaled transfer difference	1.4564	0.0000

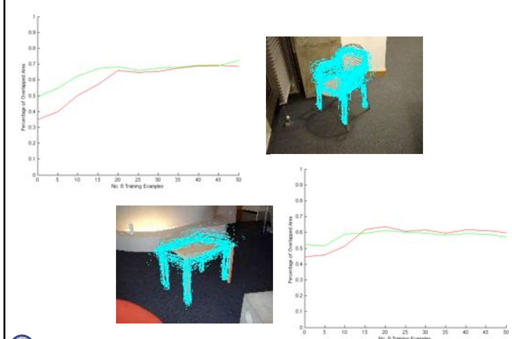
TL5 Ongoing: Other object classes

Synthetic to real



TL2/3/5 Ongoing: Other object classes

Varying viewpoint and shape within class - using synthetic data



Sample Images

Some successful results

Some less successful results

Most failures due to problems with region finder

TL Y1 External Evaluation Summary

Stratagus

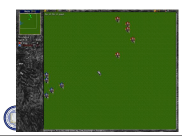
Tom Dietterich
Alan Fern
Prasad Tadepalli

School of EECS
Oregon State University

Problem Statement

- Objective: demonstrate transfer between complex sequential decision making tasks
  - Technology: hierarchical reinforcement learning with function approximation
  - Domains: Sub-problems of the Stratagus game Wargus
  - TL levels addressed this year
    - Resource Gathering Experiment 1 (R-1): Level 3
    - Resource Gathering Experiment 2 (R-2): Levels 1 and 3
    - Resource Gathering Experiment 3 (R-3): Levels 4 and 3
    - Tactical Experiment 1 (T-1): Level 3
    - Tactical Experiment 2 (T-2): Levels 3 and 4
- Approach
  - Each experiment includes a number of A-B task pairs. For each pair a number of transfer and non-transfer curves were generated. Transfer metrics were computed for each pair and averaged across all pairs of an experiment to judge overall performance

Example: T-2



transfer from small tactical battles to larger battles

Wargus Sub-Problems Summary

Wargus

- Based on the popular commercial game Warcraft
- The objective of the game is destroy an enemy by managing/growing resources and strategic military activity
- Year 1 focus is on two Wargus sub-problems: **resource gathering** and **tactical battles**

Tactical Domain

- **Goal:** learn to defeat Stratagus' AI in tactical battles
- **Parameterized by:** # of enemy and friendly squadrons/ units
- **# of states: >** 1e49 for small 5 vs. 5 battle
  - compared to ≈ 1e43 for full size chess
- **# of actions:** 3125 for small 5 vs. 5 battle
  - compared to ≈ 30 for chess
- **Transfer Levels:**
  - **(Level 3)** Transfer between different initial configurations of squadrons/units; same number of squadrons/units
  - **(Levels 4 & 3)** Transfer between different number of units; # of squadrons remains unchanged

Resource Gathering Domain

- **Goal:** learn to quickly gather specified amount of resources (e.g. gold, wood, etc.)
- **Parameterized by:** # and sizes of communities, # and sizes of forests, # of gold mines, resource requirments
- **# of states:** > 1e62 for small 5 peasant, 10 tree, 2 goldmine scenario
- **# of actions:** > 750K joint actions for 5 peasants
- **Transfer between:**
  - **(Level 3)** Transfer between different initial terrain and community configurations
  - **(Levels 3 & 1)** Transfer between different resource requirements; number of peasants unchanged
  - **(Levels 4 & 3)** Transfer between different number of peasants; # of communities remains unchanged

Domain Performance Metric(s) & Goal(s)

Experiment	Metric(s)
Tactical TL 3	Damage differential: difference between enemy and friendly health after one side is destroyed
Tactical TL 3 & 4	Damage differential
Resource TL 3	Time to achieve resource requirements
Resource TL 1 & 3	Time to achieve resource requirements
Resource TL 3 & 4	Time to achieve resource requirements

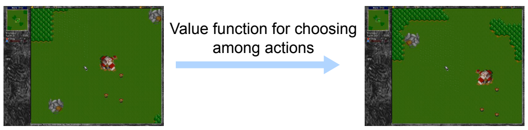
Evaluation Analysis Summary

Evaluation Type: External
Client: Oregon State
Domain: Stratagus

Experiment	TL Metric Goals Met?	Discussion
Tactical TL 3	No	TR = 6.65 Average over 8 A-B pairs
Tactical TL 3 & 4	✓	TR = 11.16 Average over 8 A-B pairs
Resource TL 3	✓	TR = 19.81 Average over 12 A-B pairs
Resource TL 1 & 3	✓	TR = 11.28 Average over 16 A-B pairs
Resource TL 3 & 4	✓	TR = 13.41 Average over 16 A-B pairs

Year 1 goal: Transfer ratio > 10

TL 3 : Resource Gathering



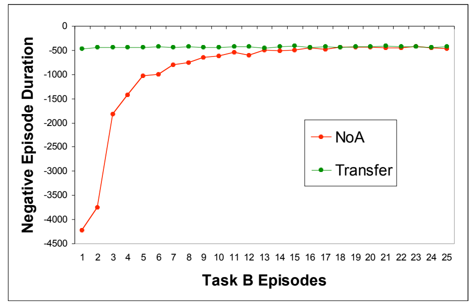
Task A: Gathering target amounts of gold and wood on map A
Task B: Gathering target amounts of gold and wood on map B
Maps A and B differ in **locations** of resources, bases, & peasants

Transferred knowledge:
- Parameters for hierarchically decomposed value function

Performance goal: demonstrate faster gathering via transfer

Ran experiments for 12 different A-B pairs of maps

Resource TL 3 Average Curves: Pair 1



Average curves for other pairs are similar

Resource TL3 Statistics: Pairs 1- 4

TL Metrics	Pair 1 Score	Pair 1 P Value	Pair 2 Score	Pair 2 P Value	Pair 3 Score	Pair 3 P Value	Pair 4 Score	Pair 4 P Value
Transfer ratio	14.24	0.0002	17.34	0.0000	13.80	0.0000	26.07	0.0000
Transfer ratio (truncated)	14.24	0.0010	21.95	0.0018	20.93	0.0000	34.89	0.0000
Jump start	4904	0.0008	4768	0.0010	5782	0.0002	3750	0.0016
ARR (narrow)	0.568	0.0278	0.00	0.6062	0.767	0.0036	0.860	0.0092
ARR (wide)	0.996	0.0094	1.00	0.0006	0.994	0.0000	0.998	0.0114
Asymptotic advantage	2.00	0.4052	26.40	0.0288	8.60	0.2090	10.40	0.1764
Ratio (of area under the curves)	0.556	0.9992	0.525	0.9998	0.548	0.9996	0.496	0.9990
Transfer difference	9206	0.0000	9551	0.0000	9875	0.0000	10556	0.0000
Transfer difference (scaled)	-20.3	0.9998	-22.9	1.0000	-21.12	0.9998	-25.35	0.9998

Ratio (of area under the curves) and Transfer difference (scaled) are not well behaved for negative valued performance metrics, such as Negative Episode Duration.

Resource TL3 Statistics: Pairs 5 - 8

TL Metrics	Pair 5 Score	Pair 5 P Value	Pair 6 Score	Pair 6 P Value	Pair 7 Score	Pair 7 P Value	Pair 8 Score	Pair 8 P Value
Transfer ratio	21.32	0.0002	17.04	0.0000	20.21	0.0002	12.18	0.0002
Transfer ratio (truncated)	99.40	0.0002	20.12	0.0000	41.28	0.0000	16.07	0.0000
Jump start	4758	0.0004	5790	0.0006	3709	0.0008	4868	0.0006
ARR (narrow)	0	0.6132	0.789	0.0012	0.851	0.0024	0.653	0.0066
ARR (wide)	1.001	0.0004	0.995	0.0000	0.996	0.0076	0.993	0.0020
Asymptotic advantage	28.20	0.0160	12.40	0.1308	11.60	0.1766	13.20	0.0928
Ratio (of area under the curves)	0.518	0.9994	0.537	0.9994	0.501	0.9992	0.550	0.9992
Transfer difference	9701	0.0000	10108	0.0000	10461	0.0000	9335	0.0000
Transfer difference (scaled)	-23.37	0.9998	-21.79	0.9998	-25.19	0.9998	-21.13	1.0000

Ratio (of area under the curves) and Transfer difference (scaled) are not well behaved for negative valued performance metrics, such as Negative Episode Duration.

Resource TL3 Statistics: Pairs 9 - 12

TL Metrics	Pair 9 Score	Pair 9 P Value	Pair 10 Score	Pair 10 P Value	Pair 11 Score	Pair 11 P Value	Pair 12 Score	Pair 12 P Value
Transfer ratio	14.04	0.0002	33.01	0.0000	21.77	0.0002	26.75	0.0000
Transfer ratio (truncated)	16.99	0.0000	108.2	0.0000	36.91	0.0006	33.63	0.0002
Jump start	5780	0.0012	3754	0.0002	4909	0.0008	4753	0.0010
ARR (narrow)	0.762	0.0026	0.896	0.0078	0.854	0.0044	0.000	0.6124
ARR (wide)	0.994	0.0000	0.998	0.0114	0.998	0.0016	1.000	0.0000
Asymptotic advantage	14.20	0.1122	4.40	0.3278	10.80	0.1562	25.80	0.0164
Ratio (of area under the curves)	0.542	0.9998	0.499	0.9990	0.535	0.9998	0.516	0.9994
Transfer difference	10014	0.0000	10504	0.0000	9648	0.0004	9742	0.0000
Transfer difference (scaled)	-21.67	0.9998	-24.86	0.9998	-21.73	0.9996	-23.34	0.9994

Ratio (of area under the curves) and Transfer difference (scaled) are not well behaved for negative valued performance metrics, such as Negative Episode Duration.

Resource TL3 Statistics: Average Across Pairs

TL Metrics	Score Average	Score Minimum	Score Maximum
Transfer ratio	19.81	12.18	33.01
Transfer ratio (truncated)	29.04	14.24	108.2
Jump start	4753	3709	5782
ARR (narrow)	0.5833	0.000	0.860
ARR (wide)	0.9968	0.993	1.000
Asymptotic advantage	14.00	2.00	28.20
Ratio (of area under the curves)	0.526	0.496	0.556
Transfer difference	9891	9206	10556
Transfer difference (scaled)	-22.72	-25.35	-20.30

Averaged across 12 A-B pairs

TL 1 & 3: Resource Gathering

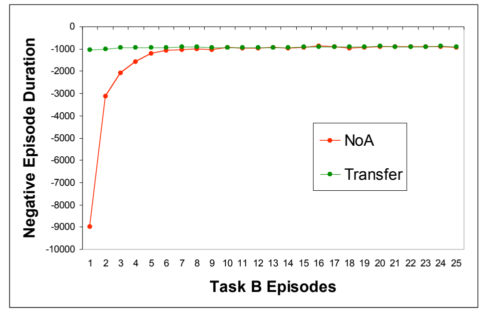
Value function for choosing among actions

Task A: Gathering target amounts of gold and wood on map A
Task B: Gathering target amounts of gold and wood on map B
Maps A and B differ in **resource requirements** and locations of resources, bases, and peasants

Transferred knowledge: parameters for hierarchically decomposed value function
Performance goal: demonstrate faster gathering via transfer

Ran experiments for 16 different A-B pairs of maps

Resource TL 1&3 Average Curves: Pair 1



Average curves for other pairs are similar

Resource TL 1&3 Statistics: Pairs 1- 4

TL Metrics	Pair 1		Pair 2		Pair 3		Pair 4	
	Score	P Value	Score	P Value	Score	P Value	Score	P Value
Transfer ratio	7.79	0.0002	13.02	0.0000	11.02	0.0000	8.04	0.0006
Transfer ratio (truncated)	8.55	0.0002	17.05	0.0004	11.40	0.0000	9.183	0.0000
Jump start	7954	0.0006	7520	0.0006	6441	0.0000	7384	0.0006
ARR (narrow)	0.57	0.0114	0.69	0.0090	0.336	0.1002	0.456	0.0116
ARR (wide)	0.99	0.0030	-INF	0.2432	0.994	0.0026	0.991	0.0002
Asymptotic advantage	4.40	0.3592	-1.20	0.5518	0.80	0.4586	15.00	0.1316
Ratio (of area under the curves)	0.71	0.9996	0.69	0.9998	0.720	0.9994	0.732	0.9998
Transfer difference	8833	0.0004	8583	0.0002	8410	0.0004	8324	0.0002
Transfer difference (scaled)	-10.08	0.9996	-10.76	0.9996	-9.696	0.9996	-9.251	1.0000

Ratio (of area under the curves) and Transfer difference (scaled) are not well behaved for negative valued performance metrics, such as Negative Episode Duration.

Resource TL 1&3 Statistics: Pairs 5 - 8

TL Metrics	Pair 5		Pair 6		Pair 7		Pair 8	
	Score	P Value	Score	P Value	Score	P Value	Score	P Value
Transfer ratio	8.29	0.0004	20.00	0.0000	18.65	0.0000	9.54	0.0002
Transfer ratio (truncated)	9.911	0.0000	30.59	0.0006	18.65	0.0020	11.60	0.0012
Jump start	8008	0.0008	7564	0.0006	6450	0.0004	7310	0.0008
ARR (narrow)	0.435	0.0450	0.804	0.0054	0.813	0.0082	0.749	0.0024
ARR (wide)	-INF	0.2452	-INF	0.2440	0.998	0.0030	0.993	0.0008
Asymptotic advantage	-6.40	0.6494	-1.40	0.5690	0.00	0.4918	18.00	0.0566
Ratio (of area under the curves)	0.716	0.9998	0.689	0.9996	0.709	0.9996	0.724	0.9998
Transfer difference	8818	0.0002	8832	0.0002	8736	0.0002	8575	0.0004
Transfer difference (scaled)	-10.01	0.9996	-11.0	0.9994	-10.06	0.9998	-9.562	0.9998

Ratio (of area under the curves) and Transfer difference (scaled) are not well behaved for negative valued performance metrics, such as Negative Episode Duration.

Resource TL 1&3 Statistics: Pairs 9 - 12

TL Metrics	Pair 9		Pair 10		Pair 11		Pair 12	
	Score	P Value	Score	P Value	Score	P Value	Score	P Value
Transfer ratio	8.99	0.0000	10.67	0.0000	8.66	0.0008	9.05	0.0000
Transfer ratio (truncated)	11.75	0.0012	10.67	0.0000	10.49	0.0014	10.92	0.0016
Jump start	7967	0.0006	7471	0.0008	6344	0.0006	7243	0.0000
ARR (narrow)	0.669	0.0066	0.714	0.0118	0.736	0.0070	0.720	0.0018
ARR (wide)	-INF	0.286	0.993	0.0004	0.991	0.0024	0.990	0.0004
Asymptotic advantage	-1.80	0.5402	7.40	0.2138	12.60	0.1556	17.80	0.0808
Ratio (of area under the curves)	0.713	0.9998	0.698	0.9996	0.719	0.9996	0.726	0.9996
Transfer difference	8912	0.0004	8587	0.0002	8433	0.0002	8517	0.0004
Transfer difference (scaled)	-10.12	0.9998	-10.8	1.0000	-9.856	0.9998	-9.495	0.9998

Ratio (of area under the curves) and Transfer difference (scaled) are not well behaved for negative valued performance metrics, such as Negative Episode Duration.

Resource TL 1&3 Statistics: Pairs 13- 16

TL Metrics	Pair 13		Pair 14		Pair 15		Pair 16	
	Score	P Value	Score	P Value	Score	P Value	Score	P Value
Transfer ratio	7.17	0.0004	18.02	0.0002	12.00	0.0002	9.58	0.0000
Transfer ratio (truncated)	7.39	0.0024	22.82	0.0008	17.60	0.0002	9.58	0.0002
Jump start	8023	0.0002	7616	0.0002	6426	0.0008	7334	0.0002
ARR (narrow)	0.416	0.0570	0.553	0.0162	0.759	0.0042	0.717	0.0034
ARR (wide)	0.992	0.0050	0.998	0.0028	0.997	0.0008	0.994	0.0006
Asymptotic advantage	10.00	0.2642	7.40	0.2088	11.60	0.1196	21.60	0.0650
Ratio (of area under the curves)	0.716	0.9990	0.685	0.9992	0.710	0.0008	0.721	0.9996
Transfer difference	8835	0.0004	8949	0.0002	8717	0.0004	8657	0.0012
Transfer difference (scaled)	-10.15	0.9988	-11.3	0.9990	-10.1	0.9986	-9.692	0.9996

Ratio (of area under the curves) and Transfer difference (scaled) are not well behaved for negative valued performance metrics, such as Negative Episode Duration.

Resource TL 1&3 Statistics: Average Across Pairs

TL Metrics	Score Average	Score Minimum	Score Maximum
Transfer ratio	11.28	7.17	20.00
Transfer ratio (truncated)	13.63	7.39	30.59
Jump start	7216	6344	8023
ARR (narrow)	0.633	0.336	0.813
ARR (wide)	-INF	-INF	0.998
Asymptotic advantage	7.24	-6.40	21.60
Ratio (of area under the curves)	0.711	0.685	0.732
Transfer difference	8669	8410	8949
Transfer difference (scaled)	-10.12	-11.3	-9.25

Averaged across 16 A-B pairs

TL 3 & 4: Resource Gathering

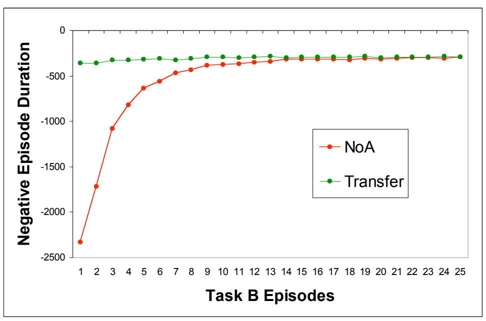
Value function for choosing among actions

Task A: Gathering target amounts of gold and wood on map A
Task B: Gathering target amounts of gold and wood on map B
Maps A and B differ in **number of peasants** and locations of resources, bases, and peasants

Transferred knowledge: parameters for hierarchically decomposed value function
Performance goal: demonstrate faster gathering via transfer

Ran experiments for 16 different A-B pairs of maps

Resource TL 3&4 Average Curves: Pair 1



Average curves for other pairs are similar

Resource TL 3&4 Statistics: Pairs 1- 4

TL Metrics	Pair 1 Score	Pair 1 P Value	Pair 2 Score	Pair 2 P Value	Pair 3 Score	Pair 3 P Value	Pair 4 Score	Pair 4 P Value
Transfer ratio	10.11	0.0008	12.90	0.0000	10.47	0.0004	15.85	0.0000
Transfer ratio (truncated)	10.11	0.0008	13.76	0.0000	14.94	0.0008	18.51	0.0018
Jump start	1968	0.0008	1720	0.0002	2728	0.0008	2481	0.0006
ARR (narrow)	0.734	0.0020	0.714	0.0156	0.837	0.0110	0.838	0.0050
ARR (wide)	0.990	0.0004	0.989	0.0018	0.995	0.0002	0.994	0.0004
Asymptotic advantage	11.00	0.0664	4.00	0.2384	7.20	0.0494	1.80	0.3588
Ratio (of area under the curves)	0.595	0.9992	0.571	0.9992	0.580	0.9994	0.562	0.9996
Transfer difference	4987	0.0006	5129	0.0004	4486	0.0002	4725	0.0010
Transfer difference (scaled)	-17.64	0.9992	-19.2	0.9986	-18.78	0.9994	-19.67	0.9996

Ratio (of area under the curves) and Transfer difference (scaled) are not well behaved for negative valued performance metrics, such as Negative Episode Duration.

Resource TL 3&4 Statistics: Pairs 5 - 8

TL Metrics	Pair 5 Score	Pair 5 P Value	Pair 6 Score	Pair 6 P Value	Pair 7 Score	Pair 7 P Value	Pair 8 Score	Pair 8 P Value
Transfer ratio	12.01	0.0000	14.91	0.0000	15.86	0.0000	15.85	0.0000
Transfer ratio (truncated)	16.32	0.0006	17.34	0.0006	15.86	0.0016	25.18	0.0088
Jump start	1911	0.0006	1712	0.0006	2750	0.0002	2504	0.0008
ARR (narrow)	0.829	0.0020	0.784	0.0212	0.797	0.0196	0.816	0.0088
ARR (wide)	0.988	0.0006	0.991	0.0020	0.997	0.0062	0.995	0.0026
Asymptotic advantage	10.40	0.0918	0.20	0.4746	0.00	0.4842	2.00	0.3338
Ratio (of area under the curves)	0.589	0.9998	0.573	0.9992	0.580	0.9994	0.562	0.9996
Transfer difference	5060	0.0012	5102	0.0008	4485	0.0010	4730	0.0002
Transfer difference (scaled)	-12.8	0.9996	-18.8	0.9996	-18.23	0.9996	-19.70	0.9998

Ratio (of area under the curves) and Transfer difference (scaled) are not well behaved for negative valued performance metrics, such as Negative Episode Duration.

Resource TL 3&4 Statistics: Pairs 9 - 12

TL Metrics	Pair 9 Score	Pair 9 P Value	Pair 10 Score	Pair 10 P Value	Pair 11 Score	Pair 11 P Value	Pair 12 Score	Pair 12 P Value
Transfer ratio	8.24	0.0004	10.72	0.0004	13.29	0.0004	12.34	0.0004
Transfer ratio (truncated)	8.24	0.0008	11.92	0.0000	17.25	0.0078	15.48	0.0002
Jump start	1884	0.0000	1706	0.0000	2734	0.0006	2484	0.0010
ARR (narrow)	0.781	0.0004	0.698	0.0118	0.908	0.0006	0.842	0.0010
ARR (wide)	0.982	0.0004	0.985	0.0014	0.998	0.0012	0.994	0.0004
Asymptotic advantage	17.20	0.0314	3.00	0.3234	4.20	0.1998	6.60	0.1296
Ratio (of area under the curves)	0.594	0.0004	0.580	0.9994	0.577	0.9992	0.561	0.9994
Transfer difference	4993	0.0002	5019	0.0010	4519	0.0008	4741	0.0004
Transfer difference (scaled)	-18.0	0.9994	-18.7	0.9998	-18.6	0.9998	-20.1	0.9994

Ratio (of area under the curves) and Transfer difference (scaled) are not well behaved for negative valued performance metrics, such as Negative Episode Duration.

Resource TL 3&4 Statistics: Pairs 13- 16

TL Metrics	Pair 13		Pair 14		Pair 15		Pair 16	
	Score	P Value	Score	P Value	Score	P Value	Score	P Value
Transfer ratio	13.91	0.0000	10.73	0.0006	21.09	0.0000	16.25	0.0002
Transfer ratio (truncated)	15.60	0.0000	10.73	0.0018	30.28	0.0060	30.21	0.0050
Jump start	1966	0.0006	1755	0.0000	2777	0.0006	2563	0.0006
ARR (narrow)	0.862	0.0000	0.631	0.0094	0.629	0.0269	0	0.6118
ARR (wide)	0.995	0.0008	0.991	0.0032	0.999	0.0240	0.999	0.0198
Asymptotic advantage	12.80	0.0514	9.60	0.0428	0.20	0.4752	6.40	0.1524
Ratio (of area under the curves)	0.579	0.9994	0.568	0.9994	0.573	0.9996	0.552	1.0000
Transfer difference	5176	0.0002	5164	0.0000	4564	0.0004	4837	0.0006
Transfer difference (scaled)	-18.43	0.9992	-19.7	0.9994	-18.5	0.9988	-20.53	0.9996

Ratio (of area under the curves) and Transfer difference (scaled) are not well behaved for negative valued performance metrics, such as Negative Episode Duration.

Resource TL 3&4 Statistics: Average Across Pairs

TL Metrics	Score		
	Average	Minimum	Maximum
Transfer ratio	13.41	8.24	21.09
Transfer ratio (truncated)	16.98	8.24	30.28
Jump start	2228	1706	2750
ARR (narrow)	0.734	0.000	0.908
ARR (wide)	0.992	0.982	0.999
Asymptotic advantage	6.04	0.000	17.20
Ratio (of area under the curves)	0.574	0.552	0.595
Transfer difference	4857	4725	5176
Transfer difference (scaled)	-18.58	-20.53	-12.8

Averaged across 16 A-B pairs

TL 3: Tactical Domain

Value function for choosing among enemy targets

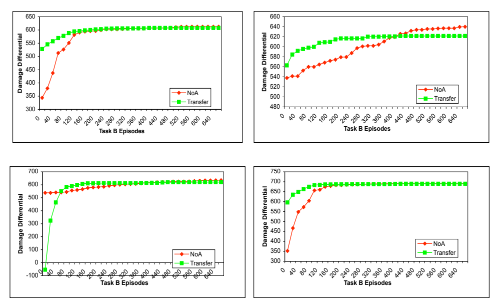
Task A: Destroy 5 enemy units with 5 friendly units on map A
Task B: Destroy 5 enemy units with 5 friendly units on map B

Maps A and B differ in initial locations of enemy and friendly units

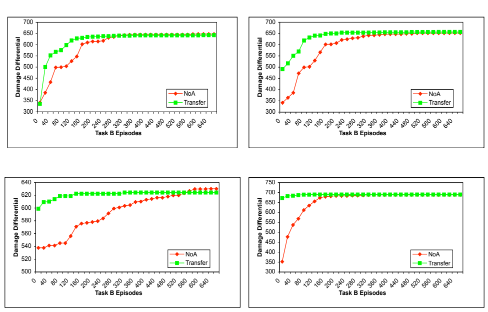
Transferred knowledge: Parameters for shared multi-agent value function
Performance goal: demonstrate improved tactics via transfer

Ran experiments for 8 different A-B pairs of maps

Tactical TL3 Average Curves: Pairs 1- 4



Tactical TL3 Average Curves: Pairs 5 - 8



Tactical TL3 Statistics: Pairs 1- 4

TL Metrics	Pair 1		Pair 2		Pair 3		Pair 4	
	Score	P Value	Score	P Value	Score	P Value	Score	P Value
Transfer ratio	2.12	0.0224	1.51	0.0394	0.79	0.9218	3.58	0.0004
Transfer ratio (truncated)	2.14	0.0156	1.51	0.0378	0.78	0.9328	3.62	0.0004
Jump start	190.0	0.0000	26.00	0.0000	-610.0	1.00	251.0	0.0000
ARR (narrow)	0.511	0.0194	0.752	0.0004	-INF	0.5994	0.661	0.0128
ARR (wide)	-INF	0.2742	-INF	0.1904	-INF	0.3536	-INF	0.2842
Asymptotic advantage	-5.20	0.6752	-19.10	0.9936	-14.60	0.95	-0.30	0.5558
Ratio (of area under the curves)	1.027	0.0410	1.021	0.0136	0.983	0.946	1.028	0.0042
Transfer difference	10928	0.041	8887	0.0116	-6850	0.9384	12767	0.0042
Transfer difference (scaled)	17.31	0.0394	13.47	0.0146	-10.47	0.9424	17.95	0.0050

Tactical TL3 Statistics: Pairs 5 - 8

TL Metrics	Pair 5		Pair 6		Pair 7		Pair 8	
	Score	P Value	Score	P Value	Score	P Value	Score	P Value
Transfer ratio	1.84	0.0524	3.34	0.0000	4.22	0.0010	35.80	0.0000
Transfer ratio (truncated)	1.86	0.0506	3.31	0.0000	4.21	0.0012	32.73	0.0002
Jump start	-6.0	1.000	154.0	0.0000	63.00	0.0000	330.0	0.0000
ARR (narrow)	-INF	0.5994	0.684	0.0088	0.866	0.0008	0.890	0.0008
ARR (wide)	-INF	0.2986	0.802	0.0036	-INF	0.2602	0.809	0.0020
Asymptotic advantage	-5.10	0.6536	4.90	0.186	-6.00	0.8168	1.00	0.2556
Ratio (of area under the curves)	1.035	0.0770	1.066	0.0006	1.045	0.0010	1.04	0.0000
Transfer difference	14473	0.0686	26955	0.0004	18392	0.0010	18202	0.0000
Transfer difference (scaled)	21.70	0.077	39.85	0.0004	28.33	0.0006	25.61	0.0000

Tactical TL3 Statistics: Average Across Pairs

TL Metrics	Score		
	Average	Minimum	Maximum
Transfer ratio	6.65	0.79	35.80
Transfer ratio (truncated)	6.27	0.78	32.73
Jump start	49.75	-610.0	330.0
ARR (narrow)	-INF	-INF	0.890
ARR (wide)	-INF	-INF	0.809
Asymptotic advantage	-5.55	-19.10	4.9
Ratio (of area under the curves)	1.03	0.983	1.066
Transfer difference	12969	8887	26955
Transfer difference (scaled)	19.21	-10.47	39.85

Averaged across 8 A-B pairs

TL 3 & 4: Tactical Domain

Value function for choosing among enemy targets

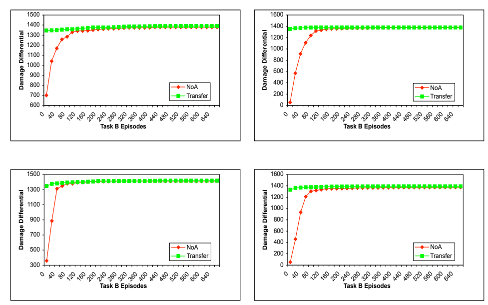
Task A: Destroy 5 enemy units with 5 friendly units on map A
Task B: Destroy 10 enemy units with 10 friendly units on map B

Transferred knowledge: Parameters for shared multi-agent value function

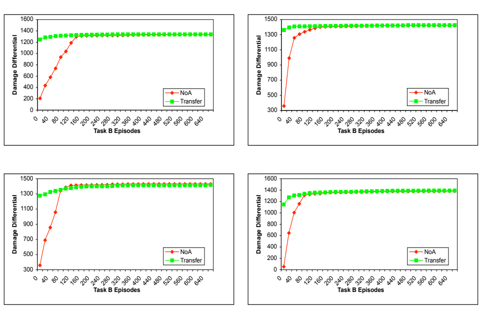
Performance goal: demonstrate improved tactics via transfer

Ran experiments for 8 different A-B pairs of maps

Tactical TL 3&4 Average Curves: Pairs 1-4



Tactical TL 3&4 Average Curves: Pairs 5-8



Tactical TL 3&4 Statistics: Pairs 1- 4

TL Metrics	Pair 1		Pair 2		Pair 3		Pair 4	
	Score	P Value	Score	P Value	Score	P Value	Score	P Value
Transfer ratio	4.41	0.0002	37.62	0.0000	5.00	0.0002	17.67	0.0002
Transfer ratio (truncated)	4.34	0.0000	37.62	0.0002	4.92	0.0004	16.20	0.0004
Jump start	665.0	0.0000	1339	0.0000	1018	0.0000	1315	0.0000
ARR (narrow)	0.574	0.056	0.894	0.0082	0.579	0.0476	0.931	0.0070
ARR (wide)	0.733	0.0026	-INF	0.2962	0.724	0.0054	0.846	0.0080
Asymptotic advantage	12.30	0.0068	-0.70	0.5690	4.90	0.2900	17.60	0.0014
Ratio (of area under the curves)	1.031	0.0004	1.060	0.0008	1.026	0.0028	1.068	0.0000
Transfer difference	28717	0.0002	53641	0.0000	24975	0.0022	60712	0.0000
Transfer difference (scaled)	20.03	0.0004	37.72	0.0004	17.09	0.0006	42.35	0.0000

Tactical TL 3&4 Statistics: Pairs 5 - 8

TL Metrics	Pair 5		Pair 6		Pair 7		Pair 8	
	Score	P Value	Score	P Value	Score	P Value	Score	P Value
Transfer ratio	12.77	0.0002	6.32	0.0000	2.08	0.0052	3.38	0.0000
Transfer ratio (truncated)	12.84	0.0000	6.22	0.0004	2.46	0.0006	3.33	0.0000
Jump start	1072	0.0000	1033	0.0000	944.0	0.0000	1128	0.0000
ARR (narrow)	0.735	0.0134	0.762	0.0016	0.063	0.4428	0.643	0.0146
ARR (wide)	-INF	0.2902	0.689	0.0028	-INF	0.2472	0.715	0.0008
Asymptotic advantage	-0.60	0.5220	2.5	0.3236	-13.20	0.8872	7.90	0.1510
Ratio (of area under the curves)	1.09	0.0002	1.031	0.0000	1.028	0.0026	1.04	0.0004
Transfer difference	75294	0.0002	28987	0.0000	26544	0.0038	39889	0.0006
Transfer difference (scaled)	54.58	0.0002	19.73	0.0000	18.00	0.0022	27.83	0.0002

Tactical TL 3&4 Statistics: Average Across Pairs

TL Metrics	Score		
	Average	Minimum	Maximum
Transfer ratio	11.16	2.08	37.62
Transfer ratio (truncated)	10.99	2.46	37.62
Jump start	1064	665.0	1339
ARR (narrow)	0.647	0.063	.931
ARR (wide)	-INF	-INF	0.846
Asymptotic advantage	3.84	-13.20	17.60
Ratio (of area under the curves)	1.046	1.026	1.09
Transfer difference	42344	24975	75294
Transfer difference (scaled)	29.70	17.09	54.58

Averaged across 8 A-B pairs

Transfer Learning in Robot Manipulation: Year 1 Results

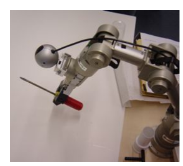
Andrew Y. Ng
Ashutosh Saxena

Stanford University

Problem Statement (1)

- Current robots can be "scripted" to perform difficult tasks in highly constrained, known environments.
- Most are hopeless when there is uncertainty in the environment, or at manipulating novel objects.

- Seeing a 3-d object for the first time using a webcam.
- Grasp the object using a robotic arm.



Testbed: STAIR
(STanford AI Robot)

Problem Statement (2)

- TL Levels addressed this year:
  - Level 1: Parameterization
  - Level 2: Extrapolating
  - Level 3: Restructuring
  - Level 4: Extending
  - Level 6: Composing

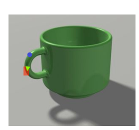
- Approach
  - Predict correct grasp from images.
  - Transfer ratio performance metric is percent agreement with labeled grasp.
  - Objects we considered have 2-5 parts.

Domain Performance Metric(s) & Goal(s)

TL	Time	Goal	RMS Err	Goal
1	232 sec/part	300 sec/part	1.94 cm	2cm
2	232 sec/part	300 sec/part	1.94 cm	2cm
3	232 sec/part	300 sec/part	1.94 cm	2cm

Labeled

Predicted



RMS error metric:
Distance between predicted grasp and nearest labeled grasp

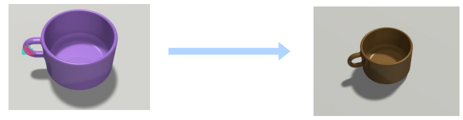
Evaluation Analysis Summary

Evaluation Type: Internal
Client: Stanford University
Domain: Robotic manipulation

TL Level	TL Metric Goals Met?	Discussion
1	✓	TL ratio achieved = 24.81
2	✓	TL ratio achieved = 10.37
3	✓	TL ratio achieved = 21.51

Year 1 goal: Transfer ratio > 10

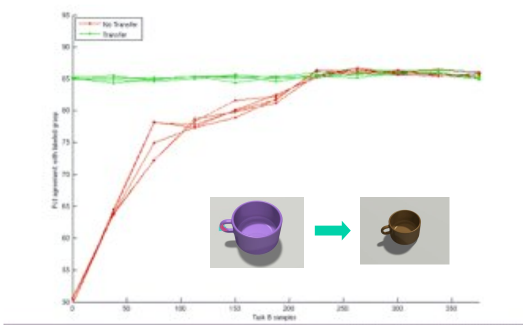
Transfer Level 1
Varying position



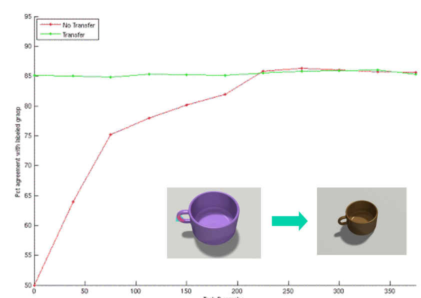
Task A: Grasp objects
Task B: Grasp objects at different locations

Transferred knowledge:
- Visual grasping instances

TL 1 Raw Curves
Varying position



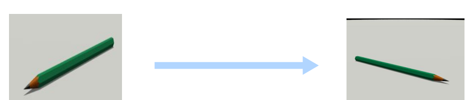
TL1 Average Curves
Varying position



TL1 Statistics
Varying position

Metric	Score	P Value
Transfer ratio	**24.81**	**0.0150**
Truncated transfer ratio	8.951	0.0010
ARR	-999999	0.1820
ARR (narrow)	0.0945	0.5026
Asymptotic advantage	-0.2716	0.9154
Jump start	35.10	0.0004
Ratio	1.080	0.0008
Transfer difference	2373.04	0.0000
Scaled transfer difference	27.51	0.0002

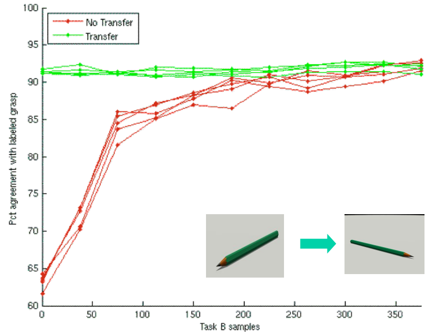
Transfer Level 2
Varying orientation



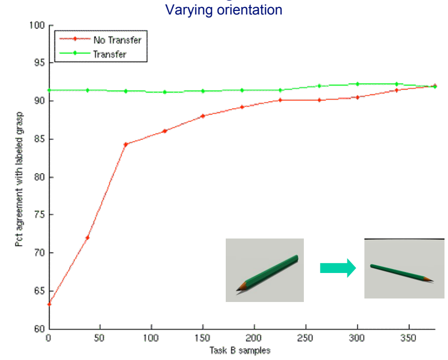
Task A: Grasp objects (Thick pencil)
Task B: Grasp objects of different dimensions in random locations and orientations. (Thin pencil)

Transferred knowledge:
- Visual grasping instances

TL 2 Raw Curves
Varying orientation



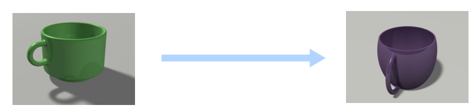
TL2 Average Curves
Varying orientation



TL2 Statistics
Varying orientation

Metric	Score	P Value
Transfer ratio	**10.37**	**0.0008**
Truncated transfer ratio	10.37	0.0002
ARR	0.8332	0.0064
ARR (narrow)	0.3738	0.2368
Asymptotic advantage	0.2182	0.2748
Jump start	28.12	0.0002
Ratio	1.0662	0.0004
Transfer difference	2131.19	0.0002
Scaled transfer difference	23.12	0.0002

Transfer Level 3
Varying shape within class

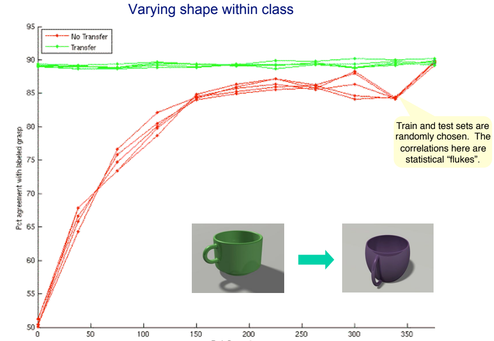


Task A: Instances of an object from the same class. (Coffee mug)
Task B: Instances of a different object from the same class. (Tea cup)
Transferred knowledge:
- Visual grasping instances

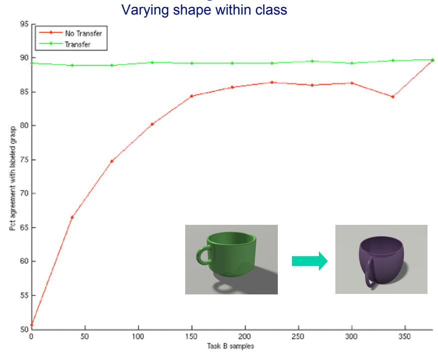
TL 3 Objects
Varying shape within class

Task A (500 objects) **Task B** (375 objects)

TL 3 Raw Curves
Varying shape within class



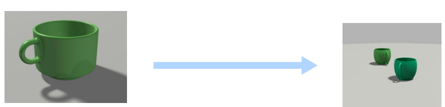
TL3 Average Curves
Varying shape within class



TL3 Statistics
Varying shape within class

Metric	Score	P Value
Transfer ratio	**21.51**	**0.0006**
Truncated transfer ratio	21.51	0.0004
ARR	0.7922	0.0118
ARR (narrow)	0.5677	0.0998
Asymptotic advantage	0.1258	0.2576
Jump start	38.516	0.0002
Ratio	1.1096	0.0002
Transfer difference	3307.10	0.0000
Scaled transfer difference	36.875	0.0002

Transfer Level 4
Multiple objects



Task A: Grasp objects (coffee mugs)
Task B: Grasp multiple objects (multiple cups)

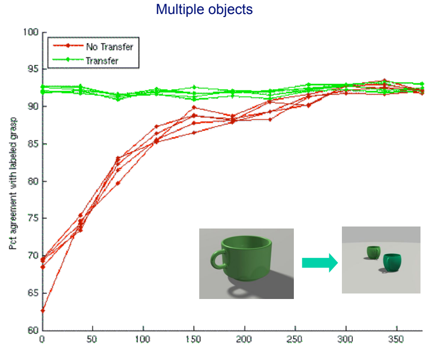
Transferred knowledge:
- Visual grasping instances

Explanation: There are two sets of grasping points: one for each cup. In detail, in this task we label every possible position in the image as a grasp or not; we then measure agreement with the ground-truth labels.

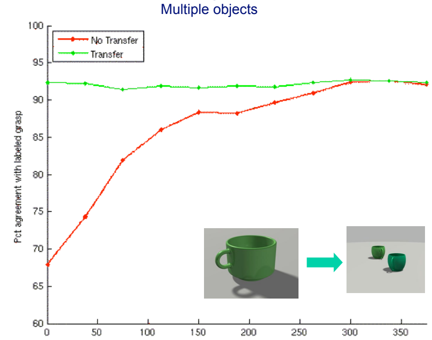
TL 4 Objects
Multiple objects

Task A (750 objects) **Task B** (375 objects)

TL 4 Raw Curves
Multiple objects



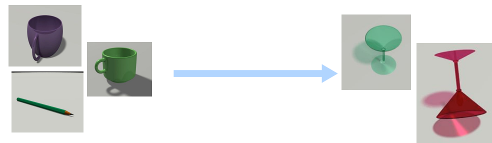
TL4 Average Curves
Multiple objects



TL4 Statistics
Multiple objects

Metric	Score	P Value
Transfer ratio	**9.89**	**0.0006**
Truncated transfer ratio	10.30	0.0004
ARR	0.8605	0.0130
ARR (narrow)	0.1280	0.4502
Asymptotic advantage	0.1098	0.3428
Jump start	24.47	0.0002
Ratio	1.066	0.0004
Transfer difference	2124.49	0.0002
Scaled transfer difference	22.91	0.0002

Transfer Level 6
New Class



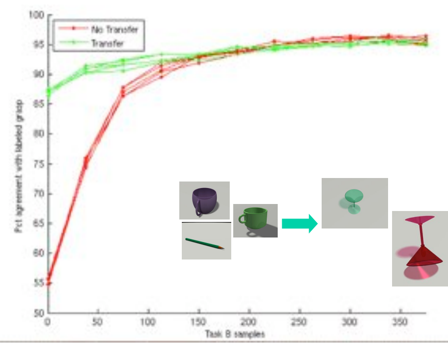
Task A: Grasp objects (pencils, cups)
Task B: Grasp objects from a new class (martini glass)

Transferred knowledge:
- Visual grasping instances

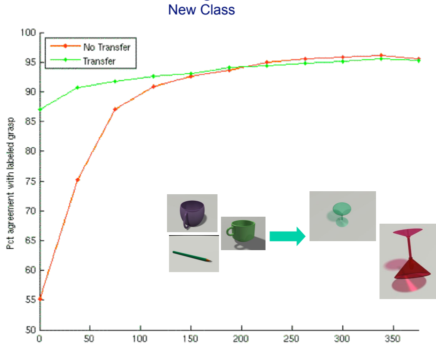
TL 6 objects
New Class

Task A (1500 objects) **Task B** (375 objects)

TL 6 Raw Curves
New Class



TL6 Average Curves
New Class



TL6 Statistics
New Class

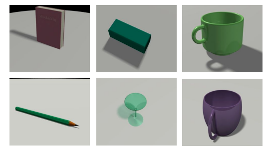
Metric	Score	P Value
Transfer ratio	**2.30**	**0.0014**
Truncated transfer ratio	2.30	0.0014
ARR	-999999	0.1466
ARR (narrow)	0.4587	0.0270
Asymptotic advantage	-0.5441	0.9676
Jump start	31.85	0.0002
Ratio	1.0404	0.0004
Transfer difference	1357.78	0.0000
Scaled transfer difference	14.13	0.0000

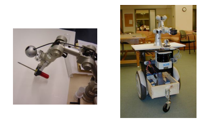
Experimental protocol summary

Level	Task A	Task B	Replications	Task A size	B train size	Test interval	Test set size
1	Objects at fixed location	Same object, different location	5	500	375	37	125
2	Objects of one dimension	Same object; but of different dimensions, and at different orientations	5	500	375	37	125
3	Instances of an object from a class	Instances of a different object from the same class	5	500	375	37	125
4	Objects from one class	Multiple objects	5	750	375	37	125
6	Objects from some classes	Objects from a new class	5	1500	375	37	125

Experimental Results Summary

For repeatability in experiments, transfer learning numbers are given for synthetic data set.



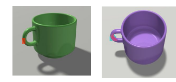


Testbed: STAIR
(STanford AI Robot)

Transfer Learning metrics:

Level	Type	Transfer ratio
1	Parameterization	24.81
2	Extrapolating	10.37
3	Restructuring	21.51
4	Extending	9.89
6	Composing	2.30

Predicted Grasp Locations



Predicted grasp locations

Some Successful Grasps on Robotic Arm

TL Y1 Internal Evaluation Summary

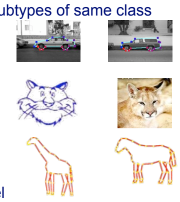
Object Recognition

Daphne Koller (PI)
Gal Elidan
Geremy Heitz
Ben Packer

Computer Science Dept.
Stanford University

Problem Statement

- Objective:
  - Technology used: Probabilistic models of object shape
  - Domain used: Object recognition in images
  - TL levels addressed this year
    - TL 3: One subtype of a class to other subtypes of same class
    - TL 5: Synthetic images to real images (optional)
    - TL 7: One class of object to another (optional)



- Approach
  - Train on task A images, learn shape model
  - Train on task B images, comparing performance with & without transferring learned task A shape model

Domain Performance Metric(s) & Goal(s)

TL Level	Running time	Goal(s)
TL 3	< 1 sec / complexity 20 (20 keypoints)	< 1000s / complexity 1000

TL Level	Error (relative RMS)	Goal(s)
TL 3	~5% (most likely point position)	< 30% of object size

Evaluation Analysis Summary

Evaluation Type: Internal
Client: UCB/Stanford
Domain: Vision

TL Level	TL Metric Goals Met?	Discussion
3	✓	TRS = 12.13

Year 1 goal: Transfer ratio > 10
TRS: Transfer ratio (smoothed)

Domain Performance Metric(s) & Goal(s)

TL Level	Running time	Goal(s)
TL 3	< 1 sec / complexity 20 (20 keypoints)	< 1000s / complexity 1000
TL 5 optional	< 150 sec / complexity 1250 (25 keypoints x 50 features)	< 1000s / complexity 1000
TL 7 optional	< 60 sec / complexity 120 (60 keypoints x 2 classes)	< 1000s / complexity 1000

TL Level	Error (relative RMS)	Goal(s)
TL 3	~5% (most likely point position)	< 30% of object size
TL 5 optional	5-18% (mean of object center)	< 30% of object size
TL 7 optional	5-11% (most likely point position)	< 30% of object size

Evaluation Analysis Summary

Evaluation Type: Internal
Client: UCB/Stanford
Domain: Vision

TL Level	TL Metric Goals Met?	Discussion
3	✓	TRS = 12.13
5 optional	✓	TR = 13.66 Average TR for 6 classes
7 optional	Not yet	

Year 1 goal: Transfer ratio > 10
TRS: Transfer ratio (smoothed)

Error Metrics for Learning

Overlap for regions

Actual

Predicted

Overlap = Intersection / Union

Distance log-likelihood for missing points on outlines

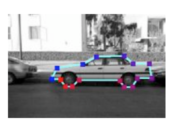
Predicted distribution for location of missing point on outline

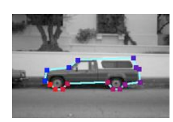
Given outline

Correct position of boundary point is assigned likelihood from distribution

Transfer Level 3

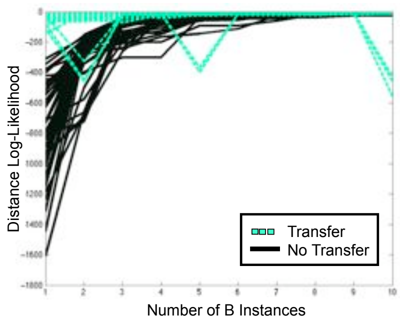
Varying shape within class



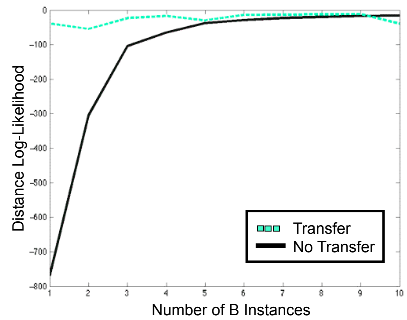


- **Learning task definition:**
  - Input: Object shape outlines
  - Performance goal: Predicting missing points
- **Set A**: Outlines of one kind of sedan
- **Set B**: Outlines of other kinds of sedan
- **Transferred knowledge**:
  - Location of keypoints (landmarks) that define car shape

TL3 Raw Curves
Varying shape within class



TL3 Average Curves
Varying shape within class



TL3 Statistics (50 folds)
Varying shape within class

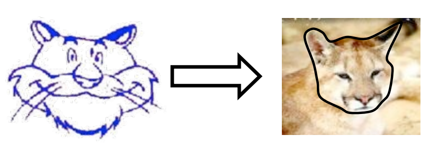
Metric	Score	P-Value
TRANSFER-RATIO (smoothed)	12.13	0.0006
ASYMPTOTIC-ADVANTAGE	4.25	0.0000
JUMP-START	730.69	0.0000
AVERAGE-RELATIVE-REDUCTION	0.99	0.0000
THE-RATIO	0.213	1.0000
TRANSFER-DIFFERENCE	779.28	0.0000
AVERAGE-RELATIVE-REDUCTION-NARROW	0.54	0.0000

The-Ratio (of area under the curves) are not well behaved for negative valued performance metrics, such as Log Likelihood.

TL3 Notes
Varying shape within class

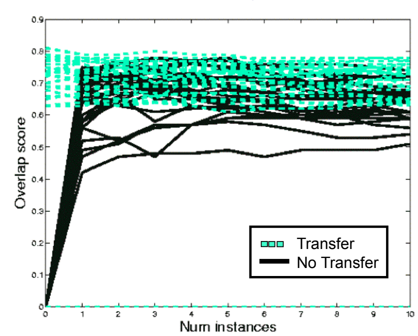
- There is a dip in performance for the no-transfer curves at the point n=1
- The reason is as follows:
  - The performance at the n=0 point is an artificial estimate, based on a simple approach that performs no learning: it interpolates the outline based on the observed points
  - The performance at the n=1 point uses the learned model from a single instance, which is a really poor estimator, hence the poor performance
- The performance at the n=2 point generally exceeds the performance at n=0, showing that learning does work better
- It is possible to artificially inflate the performance at n=1 by averaging with the interpolated estimate used for n=0, but that is against the spirit of using a purely learning-based approach

Transfer Level 5
Cartoons to real images

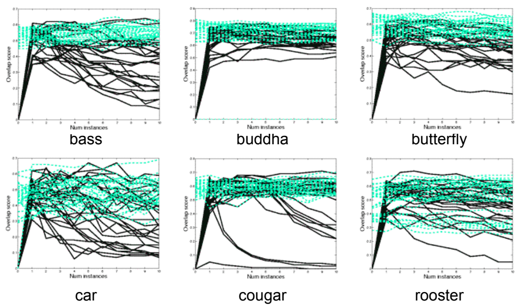


- **Learning task definition:**
  - Input: Object shape outlines in cartoon images
  - Evaluation: Outlining objects in real images
- **Set A**: Synthetic (cartoon) images of class
- **Set B**: Real images of class
- **Classes:** cougar, car, butterfly, rooster, bass, buddha
- **Transferred knowledge**:
  - Keypoints (landmarks) that define the shape
  - Location of keypoints

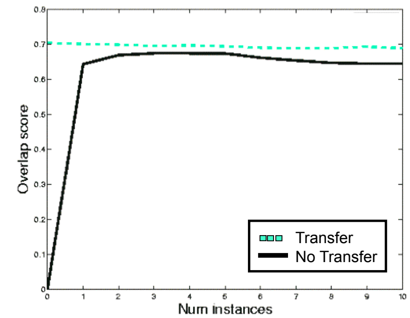
TL5 Raw Curves for Buddha Class
Cartoons to real images



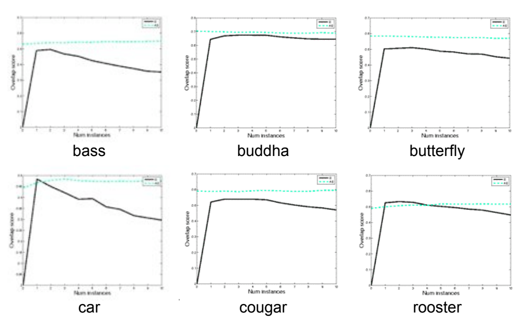
TL5 Raw Curves for all classes
Cartoons to real images



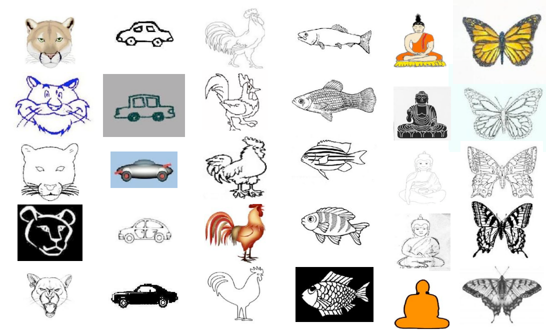
TL5 Average Curve for Butterfly class
Cartoons to real images



TL5 Average Curves for all classes
Cartoons to real images



TL5 Cartoon Training Images
Cartoons to real images



TL5 Notes
Cartoons to real images

- Based on human learning, we expected cartoons to facilitate rapid learning of basic shape
- Experiments show: Learning curve from outlines in real images deteriorates with # of training instances
- Reason: Shape learning from real images is less robust because of the complexity of real-life variation
- Conversely, cartoon learning is more robust by itself, and also helps resolve ambiguities in real images
- Reason: Cartoon drawings capture intrinsic shape and keypoint properties

DETAILED EXPLANATION: The first key step in learning the shape model from outlines is registering the different outlines. This is completely unsupervised in the cartoons, and to make the real image data comparable, we have only provided an outline, but no annotation of landmarks. It turns out that registering outlines without any model is not at easy, but the cartoons, by nature, exaggerate elements of shape and therefore make that task easier. The real images are less exaggerated, and also exhibit some subtle variation in pose (e.g., the fish twisting around). That makes the correspondence task much harder, and the results less robust. By contrast, once we have learned a shape model from the cartoons, the registration of the real images is more robust, and we can avoid many of the ambiguities and local maxima.

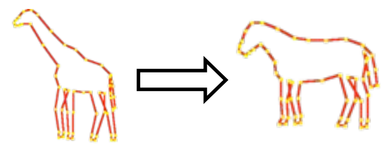
TL5 Statistics (30 folds)
Cartoons to real images

	bass		buddha		butterfly	
Metric	Score	P-Val	Score	P-Val	Score	P-Val
TRANSFER-RATIO	18.94	0.0000	8.06	0.0000	16.38	0.0000
TRUNCATED-TRANSFER-RATIO	18.94	0.0000	229.20	0.0000	311.12	0.0000
ASYMPTOTIC-ADVANTAGE	0.05	0.0004	0.03	0.1452	0.07	0.0000
JUMP-START	0.53	0.0000	0.70	0.0000	0.58	0.0000
AVERAGE-RELATIVE-REDUCTION	0.68	0.0000	0.54	0.1578	0.62	0.0012
THE-RATIO	1.34	0.0000	1.11	0.0136	1.25	0.0000
TRANSFER-DIFFERENCE	1.39	0.0000	0.68	0.0108	1.17	0.0000
AVERAGE-RELATIVE-REDUCTION-NARROW	0.00	0.5050	0.00	0.4958	0.00	0.4990

TL5 Statistics (30 folds)
Cartoons to real images

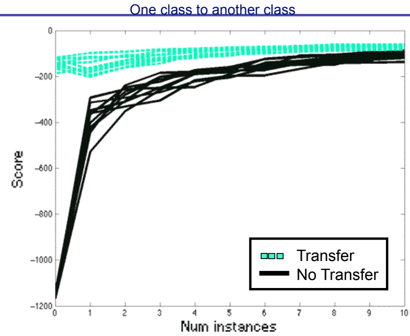
	car		cougar		rooster	
Metric	Score	P-Val	Score	P-Val	Score	P-Val
TRANSFER-RATIO	11.69	0.0000	24.18	0.0000	2.73	0.0578
TRUNCATED-TRANSFER-RATIO	8.36	0.0000	24.18	0.0000	3.01	0.0060
ASYMPTOTIC-ADVANTAGE	0.00	0.4932	0.06	0.0174	-0.01	0.7124
JUMP-START	0.44	0.0000	0.59	0.0000	0.49	0.0000
AVERAGE-RELATIVE-REDUCTION	0.62	0.0004	0.60	0.0170	-	0.2594
THE-RATIO	1.31	0.0002	1.21	0.0010	1.08	0.0992
TRANSFER-DIFFERENCE	1.12	0.0000	1.01	0.0002	0.37	0.0994
AVERAGE-RELATIVE-REDUCTION-NARROW	0.15	0.5056	0.00	0.5018	-0.44	0.5072

Transfer Level 7
One class to another class

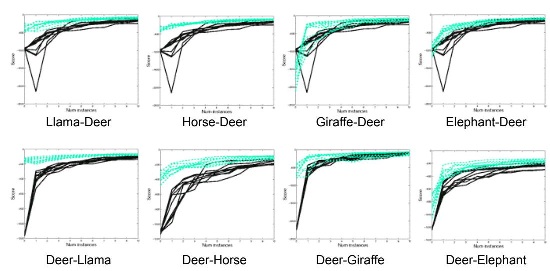


- **Learning task definition:**
  - Input: Object shape outlines
  - Performance goal: Predicting missing points
- **Set A**: Synthetic (cartoon) images of class
- **Set B**: Real images of class
- **Classes:** llama, deer, horse, giraffe, elephant, anteater
- **Transferred knowledge**:
  - Location of keypoints (landmarks)
  - Variability of keypoints

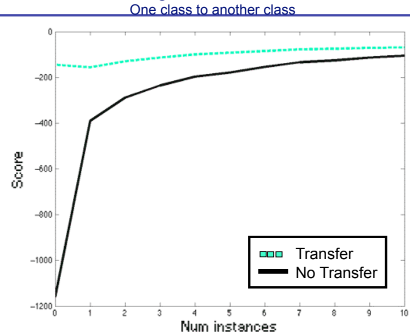
TL7 Raw Curves for deer-llama
One class to another class



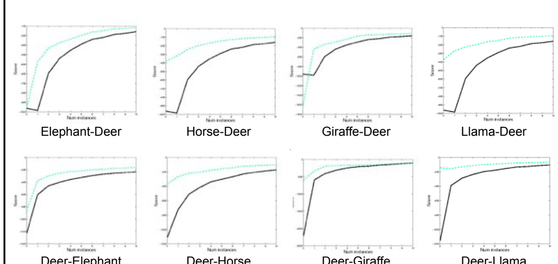
TL7 Raw Curves for pairs
One class to another class



TL7 Average Curve for deer-llama
One class to another class



TL7 Average Curves for pairs
One class to another class



TL7 Statistics (10 folds)
One class to another class

	Deer-Horse		Horse-Deer		Deer-Giraffe		Giraffe-Deer	
Metric	Score	P-Val	Score	P-Val	Score	P-Val	Score	P-Val
TRANSFER-RATIO	3.83	0.0000	4.20	0.0000	2.66	0.0000	1.66	0.0026
TRUNCATED-TRANSFER-RATIO	3.83	0.0000	4.20	0.0000	2.66	0.0000	1.66	0.0032
ASYMPTOTIC-ADVANTAGE	70.56	0.0000	63.69	0.0004	7.14	0.2796	52.76	0.0034
JUMP-START	742.20	0.0000	587.73	0.0000	1621.28	0.0000	-	1.0000
AVERAGE-RELATIVE-REDUCTION	0.77	0.0002	0.73	0.0042	0.53	0.0052	-	0.1570
THE-RATIO	0.46	1.0000	0.42	1.0000	0.55	0.9996	0.71	0.9966
TRANSFER-DIFFERENCE	2077.95	0.0000	2354.83	0.0000	1612.78	0.0000	1180.72	0.0024
AVERAGE-RELATIVE-REDUCTION-NARROW	0.81	0.0000	0.78	0.0006	0.63	0.0004	-	0.5050

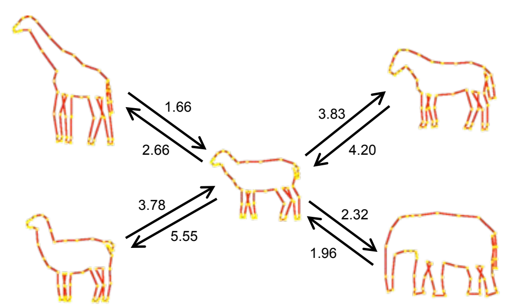
The-Ratio (of area under the curves) are not well behaved for negative valued performance metrics, such as Log Likelihood.

TL7 Statistics (10 folds)
One class to another class

	Deer-Llama		Llama-Deer		Deer-Elephant		Elephant-Deer	
Metric	Score	P-Val	Score	P-Val	Score	P-Val	Score	P-Val
TRANSFER-RATIO	5.55	0.0000	3.78	0.0000	2.32	0.0000	1.96	0.0000
TRUNCATED-TRANSFER-RATIO	5.55	0.0000	3.78	0.0000	2.32	0.0000	1.96	0.0000
ASYMPTOTIC-ADVANTAGE	36.11	0.0000	63.98	0.0006	72.40	0.0004	46.57	0.0022
JUMP-START	1015.02	0.0000	579.62	0.0000	369.82	0.0000	7.27	0.4208
AVERAGE-RELATIVE-REDUCTION	0.64	0.0002	0.71	0.0050	0.63	0.0004	0.67	0.0004
THE-RATIO	0.41	1.0000	0.44	1.0000	0.67	1.0000	0.65	1.0000
TRANSFER-DIFFERENCE	1443.16	0.0000	2274.51	0.0000	1318.81	0.0000	1432.05	0.0000
AVERAGE-RELATIVE-REDUCTION-NARROW	0.74	0.0002	0.72	0.0002	0.66	0.0000	0.65	0.0054

The-Ratio (of area under the curves) are not well behaved for negative valued performance metrics, such as Log Likelihood.

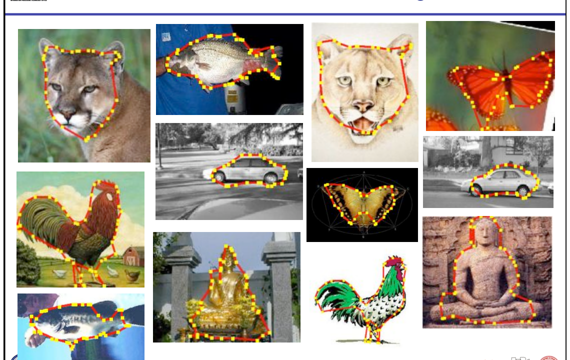
TL7 Transfer Ratios
One class to another class



Experimental protocol summary

TL	Task A	Task B	Replications	Task A size	B train size	Test interval	Test set size	Objects
3	Several instances of one type of car	Other cars	50	5	10	1	15	1
5	Cartoon drawings	Outlines in real images	30	5	10	1	15	6
7	Outlines of one quadruped class	Outline of another quadruped class	10	10	10	1	15	8 pairs

Some Successful Outlining



A Few Failed Examples

Most failures due to problems with image features

TL Y1/Y2 Site Visit

Year 1 Accomplishments
Year 2 Planning

Leslie Pack Kaelbling
Tomas Lozano-Perez
MIT

Stuart Russell
UCB

Project Goals

- Creation of generic, retargetable transfer learning system(s)
  - Theory and implementation for effective transfer learning, based on provision and accumulation of declarative probabilistic knowledge supporting transfer and improved learning
  - Demonstration in robotics, vision, strategy games, with transfer ratios > 10 at transfer levels 3/6/10 in Years 1/2/3.
- Part of larger thrust within machine learning to create cumulative, knowledge-guided mechanisms enabling very fast learning about new phenomena and unbounded extensibility.

Military relevance

- Sensor systems:
  - Target recognition systems need to adapt quickly to new target types, new background/atmospheric conditions, new sensor hardware
  - IED detection systems need to adapt quickly to new IED and camouflage types
- Control systems:
  - UAV controllers need to adapt quickly to new payloads, damaged control and lift surfaces
  - AGV controllers need to adapt quickly to new terrain types, road surfaces, obstacle/vegetation types, etc.
- Decision (support) systems:
  - Tactical and strategic planning systems must adapt quickly to novel enemy behavior, new weapon systems, new terrain factors, etc., without having to relearn all levels of behavior from scratch.

Ideal EBTL system

Cumulative learning agent

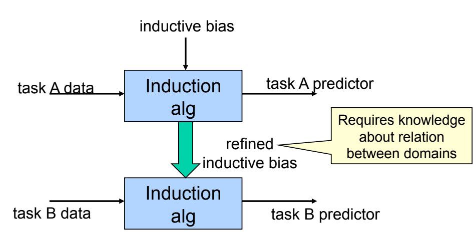
- General prior knowledge (type hierarchies, part-whole hierarchies, feature relevance models, RMDP lattices, etc., all learnable)
- Input: observations (e.g., (s,a,r) triples)
- Process: Bayesian inference
- Output: updated model, action selection

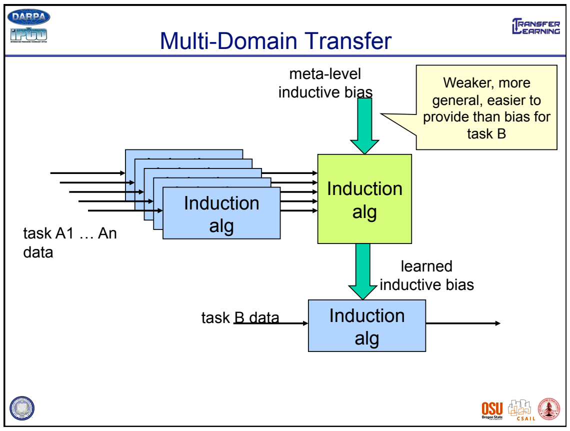
Project Theses

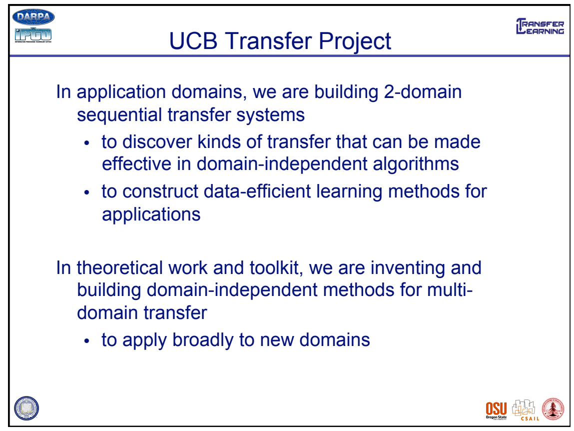
In many modern applications, it is more efficient and effective to design a learning system than to hard-code knowledge into the system.

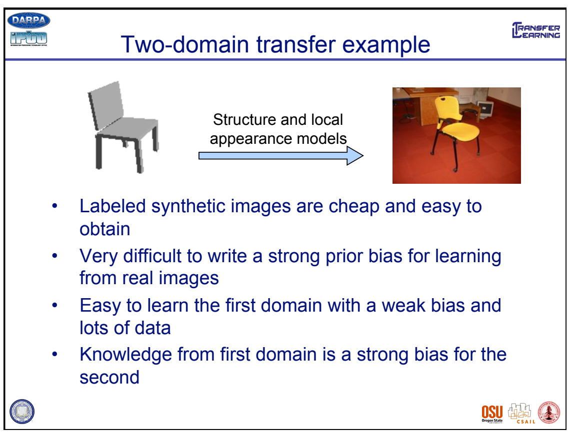
1. **Two-domain transfer-learning** is often a more effective way to obtain a strong bias for a new domain than hand-crafting that bias directly. Therefore, it's an effective engineering strategy for applications.
2. **Domain-independent multi-task transfer-learning** algorithms require a much weaker and easier to engineer inductive bias than a base-level learning algorithm.

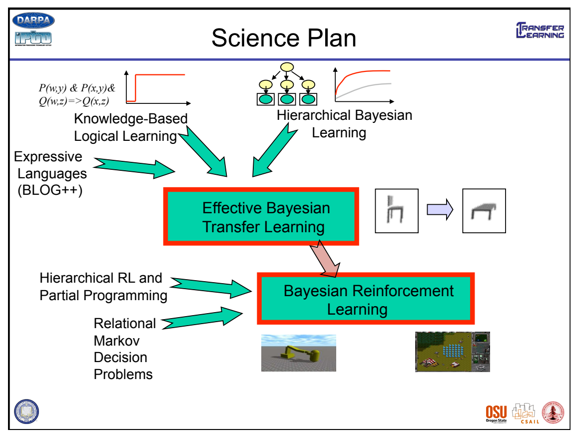
2-Domain Transfer

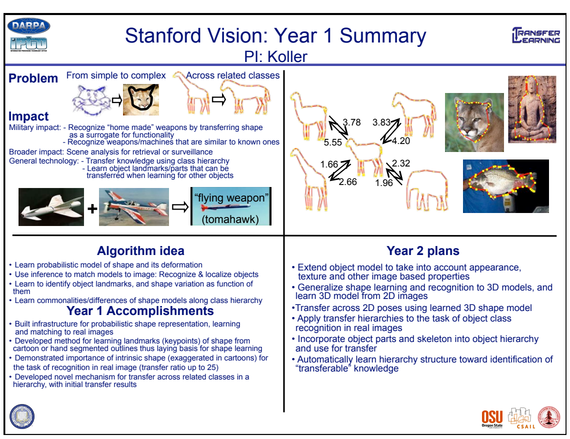


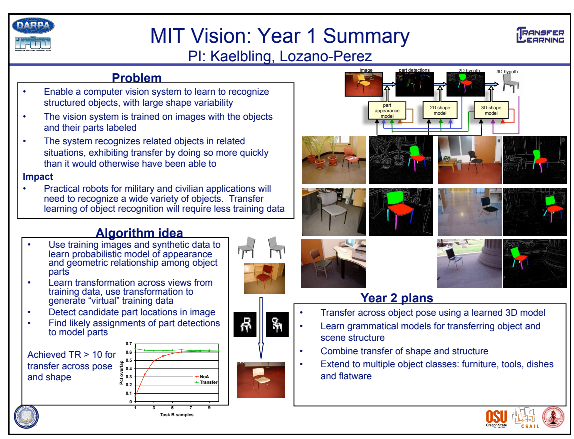

Multi-Domain Transfer
meta-level inductive bias
Weaker, more general, easier to provide than bias for task B
Induction alg
Induction alg
task A1 … An data
learned inductive bias
task B data
Induction alg


UCB Transfer Project
In application domains, we are building 2-domain sequential transfer systems
• to discover kinds of transfer that can be made effective in domain-independent algorithms
• to construct data-efficient learning methods for applications
In theoretical work and toolkit, we are inventing and building domain-independent methods for multi-domain transfer
• to apply broadly to new domains


Two-domain transfer example
Structure and local appearance models
• Labeled synthetic images are cheap and easy to obtain
• Very difficult to write a strong prior bias for learning from real images
• Easy to learn the first domain with a weak bias and lots of data
• Knowledge from first domain is a strong bias for the second


Science Plan
P(w,y) & P(x,y)& Q(w,z)=>Q(x,z)
Knowledge-Based Logical Learning
Hierarchical Bayesian Learning
Expressive Languages (BLOG++)
Effective Bayesian Transfer Learning
Hierarchical RL and Partial Programming
Bayesian Reinforcement Learning
Relational Markov Decision Problems


Stanford Vision: Year 1 Summary
PI: Koller
Problem
From simple to complex
Across related classes
Impact
Military impact: - Recognize "home made" weapons by transferring shape as a surrogate for functionality
- Recognize weapons/machines that are similar to known ones
Broader impact: Scene analysis for retrieval or surveillance
General technology: - Transfer knowledge using class hierarchy
- Learn object landmarks/parts that can be transferred when learning for other objects
"flying weapon" (tomahawk)
3.78
3.83
5.55
4.20
1.66
2.32
2.66
1.96
Algorithm idea
• Learn probabilistic model of shape and its deformation
• Use inference to match models to image: Recognize & localize objects
• Learn to identify object landmarks, and shape variation as function of them
• Learn commonalities/differences of shape models along class hierarchy
Year 1 Accomplishments
• Built infrastructure for probabilistic shape representation, learning and matching to real images
• Developed method for learning landmarks (keypoints) of shape from cartoon or hand segmented outlines thus laying basis for shape learning
• Demonstrated importance of intrinsic shape (exaggerated in cartoons) for the task of recognition in real image (transfer ratio up to 25)
• Developed novel mechanism for transfer across related classes in a hierarchy, with initial transfer results
Year 2 plans
• Extend object model to take into account appearance, texture and other image based properties
• Generalize shape learning and recognition to 3D models, and learn 3D model from 2D images
•Transfer across 2D poses using learned 3D shape model
• Apply transfer hierarchies to the task of object class recognition in real images
• Incorporate object parts and skeleton into object hierarchy and use for transfer
• Automatically learn hierarchy structure toward identification of "transferable" knowledge


MIT Vision: Year 1 Summary
PI: Kaelbling, Lozano-Perez
Problem
• Enable a computer vision system to learn to recognize structured objects, with large shape variability
• The vision system is trained on images with the objects and their parts labeled
• The system recognizes related objects in related situations, exhibiting transfer by doing so more quickly than it would otherwise have been able to
Impact
• Practical robots for military and civilian applications will need to recognize a wide variety of objects. Transfer learning of object recognition will require less training data
image
part detections
2D hypoth
3D hypoth
part appearance model
2D shape model
3D shape model
Algorithm idea
• Use training images and synthetic data to learn probabilistic model of appearance and geometric relationship among object parts
• Learn transformation across views from training data, use transformation to generate "virtual" training data
• Detect candidate part locations in image
• Find likely assignments of part detections to model parts
Achieved TR > 10 for transfer across pose and shape
Pct overlap
Task B samples
NoA
Transfer
Year 2 plans
• Transfer across object pose using a learned 3D model
• Learn grammatical models for transferring object and scene structure
• Combine transfer of shape and structure
• Extend to multiple object classes: furniture, tools, dishes and flatware

Stanford Traffic Vision: Year 1 Summary
PI: Thrun

Problem

- Detection, classification, and prediction of vehicular traffic.
- Is an application of Transfer Learning to the Visual Domain
- Impact: May make cars safer by avoiding collisions; will be necessary for meeting 2001 congressional mandate to make 1/3 of all ground vehicle unmanned.

Algorithm idea

- Developed transfer machine learning technique.
- Transfer at multiple levels:
  - Appearance
  - Behavior of moving objects
- Approach based on
  - Viola-Jones feature tracker
  - Particle filter method for motion tracking and prediction
  - Hierarchical Bayes for Transfer, with meta-parameters for appearance and motion
- We now know that transfer improves the performance in recognizing and predicting a new car; we are currently measuring the improvement (transfer ration etc).

Year 2 plans

- Continue the development of this project
- Empirical evaluations
- Integration into a physical testbed
- Porting to/from other transfer learning techniques in the image domain.

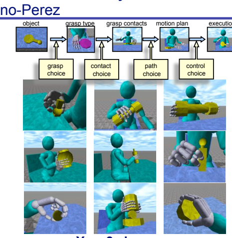
MIT Manipulation: Year 1 Summary
PI: Kaelbling, Lozano-Perez

Problem

- Enable a simulated robot to learn grasps by imitation
- A human demonstrates 5 grasp types on simulated objects
- The robot practices those grasp types on simple objects
- The robot performs the same grasp types on new objects that are different than the training objects, thus exhibiting transfer

Impact

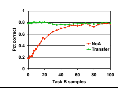
- Practical robots for military and civilian applications will need to learn to carry out new tasks. Transfer learning of manipulation tasks will require less training data

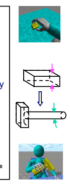


Algorithm idea

- Learn association between template grasp types and object features
- Learn quality metric on grasps
- Given new object, find most similar template grasp
- Find many ways of transforming the training object to the new object, each transformation produces a candidate grasp.
- Choose transformed grasp with highest quality

Achieved TR > 10 for transfer across pose and shape





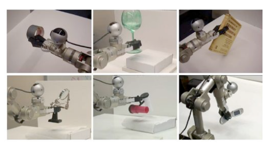
Year 2 plans

- Transfer of motions from uncluttered domains to cluttered domains: training on a table, transfer to a dishwasher
- Learning manipulation sequences and transferring components to be recombined and adapted in new domains: train on opening cupboard, transfer to dishwasher door
- Learning force-based grasp controllers via reinforcement learning: transfer across shape, mass, and material properties

Stanford Manipulation: Year 1 Summary
PI: Ng

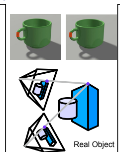
Problem

- Today's robots can be "scripted" to perform complex-seeming tasks, but are hopeless when there is uncertainty in the environment, or at manipulating novel objects.
- Seeing a 3-d object for the first time using a camera
- Grasp the object using a robotic arm.



Algorithm idea

- Learn grasps for 5 objects; transfer grasp to novel test set objects.
- Grasping approach:
  - Given 2 or more images of object, identify 2-d "grasping points" (shown in red below) in image plane.
  - Triangulate grasping points to identify 3-d grasp position.
  - Approach works even on objects where full stereo 3-d reconstruction fails.



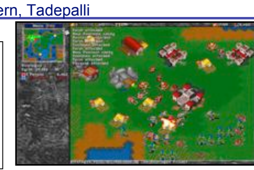
Year 2 plans

- Robotic manipulation in cluttered environments, e.g. unloading a dishwasher.
- Develop transfer algorithms to transfer from grasping one object in white background to grasping in presence of background clutter.
- Develop transfer algorithms for perceiving obstacles and grasp point in presence of other objects and background.
- Develop control algorithms to navigate robotic arm.
- Improve grasp algorithm to learn other grasp parameters, e.g. orientation and finger parameters, besides grasp location, which we did in year 1.

OSU Stratagus: Year 1 Summary
PI: Dietterich, Fern, Tadepalli

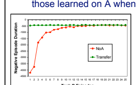
Problem

- **Investigate transfer in real-time strategy (RTS) games, focusing on the Stratagus RTS engine**
- Provides a venue to investigate transfer between complex sequential decision making problems
- **Long term practical benefits:** facilitate faster development of automated decision-making tools for complex military domains
- **Shorter term practical benefits:** facilitate faster development of intelligent agents for complex simulation environment, e.g. in training simulators

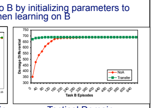


Algorithm idea

- To transfer between tasks from a given class, design a single abstract MDP *M* that includes all of the tasks as special cases
- Hierarchically structure *M* so that common subtasks of all tasks are explicit and are described by the same set of relational features
- Transfer from task A to B by initializing parameters to those learned on A when learning on B



Resource Gathering Domain



Tactical Domain

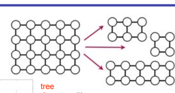
Year 2 plans

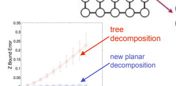
- **Challenge problems:** transfer between tasks that are more complex and more dissimilar than in year 1
  1. transfer between complex resource-production tasks that involve diverse goals and different sets of operators/resources
  2. transfer between complex tactical-battle tasks with diverse unit configurations/compositions and different unit types
- **Approaches**
  1. Develop "deeper" abstract MDP models that span a wide range of diverse tasks
  2. Develop transfer techniques for model-based learning methods
  3. Develop hierarchical Bayesian RL techniques

MIT Toolkit: Year 1 Summary
PI: Jaakkola

Problem

- P1: how to automatically divide complex inference tasks such as object recognition into reusable sub-problems
- P2: how to automatically build reusable hierarchical models for multi-task recognition tasks
- Solutions to these permit robust object recognition across different contexts and types of objects
  - e.g., by allowing us to assemble groups of correlated "features" into reusable "parts"





Algorithm idea

Our approach derives from the following two lines of reasoning:

- **Decomposing complex models**
  - a class of planar models, useful in image analysis, can be solved exactly in polynomial time
  - more expressive non-planar models can be decomposed into planar models
  - the resulting decomposition (controlled approximation) can be optimized for accuracy
- **Building sparse models**
  - reusable components come from enforcing sparsity (limiting modeling resources for special purpose solutions)
  - sparse *minimum entropy* models can be estimated efficiently in stages (adding one piece at a time)

Year 2 plans

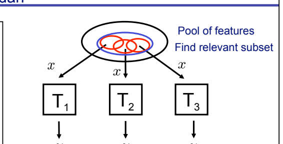
We plan to continue the work in the following four directions:

1) refine the algorithms so they can be made available as solid implementations
2) demonstrate the methods in the context of recognizing classes of objects from different viewpoints
3) analyze how many samples per task and how many model building iterations are required to attain prescribed performance levels across all tasks
4) extend the automated decomposition methods for general hierarchical models

UCB Toolkit: Year 1 Summary
PI: Jordan

Problem

- Distantly-related tasks may have little in common on the surface
- At a deeper level, what tasks may have in common is that the *same features are relevant across tasks*
- How to automatically discover which of a large set of features are relevant across multiple tasks?
- Many practical consequences: e.g., what visual features matter for grasping, what aspects of game configurations matter for making strategic decisions?



Algorithm idea

- We have solved this problem by developing a novel block-norm regularization framework
- Efficient optimization algorithm based on Blasso

features

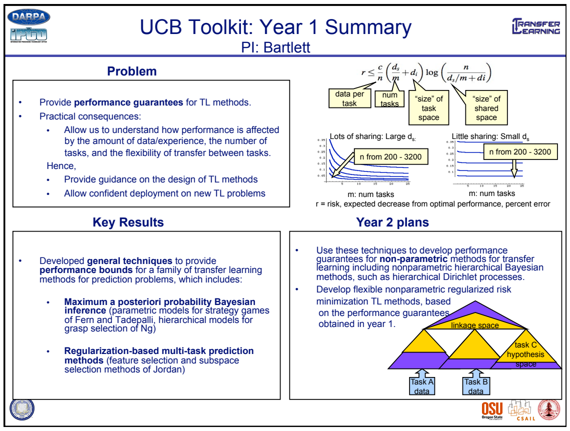
$$W = \begin{bmatrix} w_{11} & \cdots & w_{1k} \\ \vdots & \ddots & \vdots \\ w_{l1} & \cdots & w_{lk} \end{bmatrix}$$

tasks

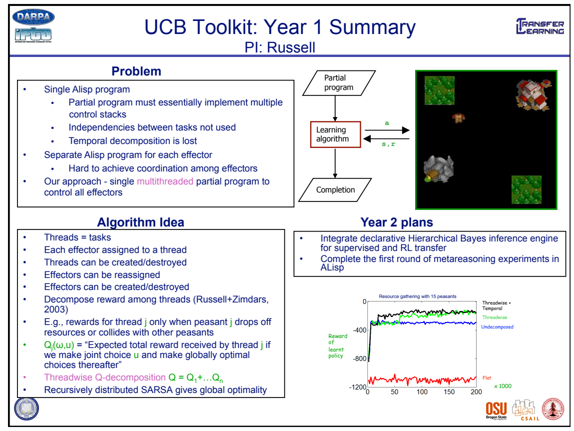
$$\min_{w} \lambda \sum_{k} \|w_k\|_2 + \sum_{i,l} \text{logloss}\left(w^l, (x_i^l, y_i^l)\right)$$

Year 2 plans

- The other major feature selection problem is that of finding effective *combinations of features* (subspaces)
- In the multi-task setting, we ask how to find a single subspace that is useful in multiple tasks
- Our approach: generate many random projections and use our block-norm method to find overlapping sets of combinations---these determine a subspace
- We will develop an algorithmic and software platform for solving general multi-task feature selection problems

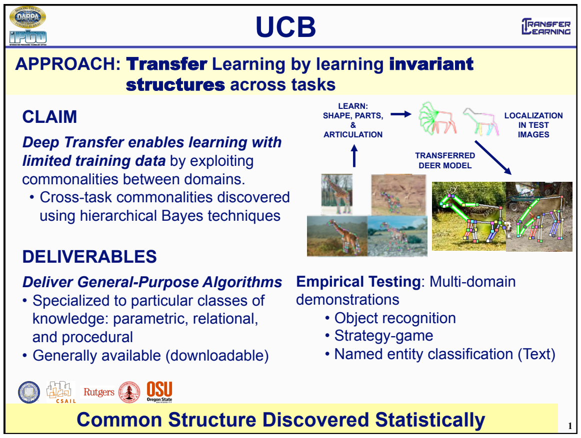


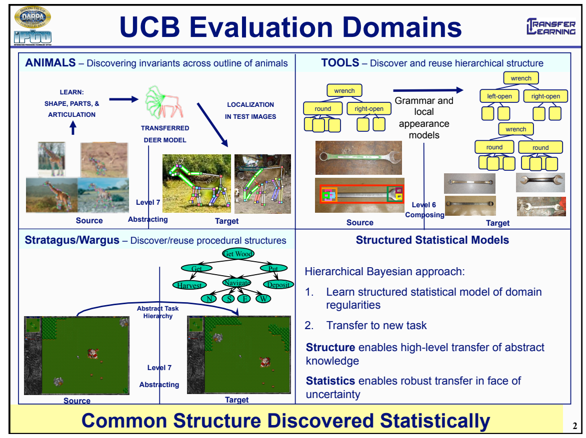
DARPA
IPTO
UCB Toolkit: Year 1 Summary
PI: Bartlett
TRANSFER LEARNING
Problem
• Provide **performance guarantees** for TL methods.
• Practical consequences:
• Allow us to understand how performance is affected by the amount of data/experience, the number of tasks, and the flexibility of transfer between tasks.
Hence,
• Provide guidance on the design of TL methods
• Allow confident deployment on new TL problems
$r \leq \frac{c}{n}\left(\frac{d_s}{m}+d_i\right)\log\left(\frac{n}{d_s/m+di}\right)$
data per task
num tasks
"size" of task space
"size" of shared space
Lots of sharing: Large d_s:
n from 200 - 3200
m: num tasks
Little sharing: Small d_s
n from 200 - 3200
m: num tasks
r = risk, expected decrease from optimal performance, percent error
Key Results
• Developed **general techniques** to provide **performance bounds** for a family of transfer learning methods for prediction problems, which includes:
• **Maximum a posteriori probability Bayesian inference** (parametric models for strategy games of Fern and Tadepalli, hierarchical models for grasp selection of Ng)
• **Regularization-based multi-task prediction methods** (feature selection and subspace selection methods of Jordan)
Year 2 plans
• Use these techniques to develop performance guarantees for **non-parametric** methods for transfer learning including nonparametric hierarchical Bayesian methods, such as hierarchical Dirichlet processes.
• Develop flexible nonparametric regularized risk minimization TL methods, based on the performance guarantees obtained in year 1.
linkage space
task C hypothesis space
Task A data
Task B data
OSU Oregon State
CSAIL

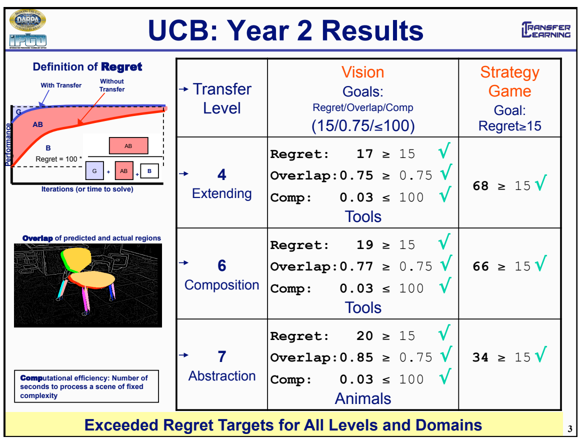


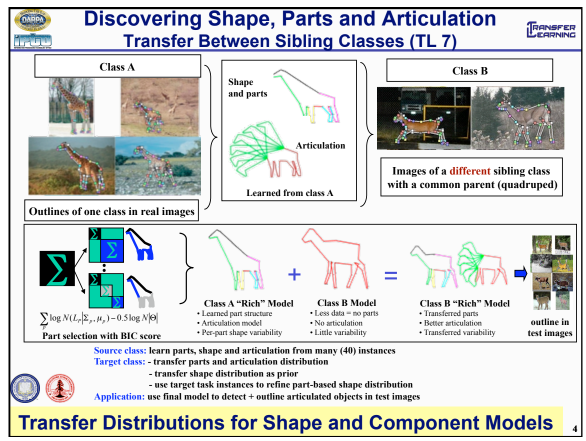
DARPA
IPTO
UCB Toolkit: Year 1 Summary
PI: Russell
TRANSFER LEARNING
Problem
• Single Alisp program
• Partial program must essentially implement multiple control stacks
• Independencies between tasks not used
• Temporal decomposition is lost
• Separate Alisp program for each effector
• Hard to achieve coordination among effectors
• Our approach - single multithreaded partial program to control all effectors
Partial program
Learning algorithm
a
s,r
Completion
Algorithm Idea
• Threads = tasks
• Each effector assigned to a thread
• Threads can be created/destroyed
• Effectors can be reassigned
• Effectors can be created/destroyed
• Decompose reward among threads (Russell+Zimdars, 2003)
• E.g., rewards for thread j only when peasant j drops off resources or collides with other peasants
• $Q_j(\omega,u)$ = "Expected total reward received by thread j if we make joint choice u and make globally optimal choices thereafter"
• Threadwise Q-decomposition $Q = Q_1+\ldots Q_n$
• Recursively distributed SARSA gives global optimality
Year 2 plans
• Integrate declarative Hierarchical Bayes inference engine for supervised and RL transfer
• Complete the first round of metareasoning experiments in ALisp
Resource gathering with 15 peasants
Reward of learnt policy
0
-400
-800
-1200
0
50
100
150
200
x 1000
Threadwise + Temporal
Threadwise
Undecomposed
Flat
OSU Oregon State
CSAIL

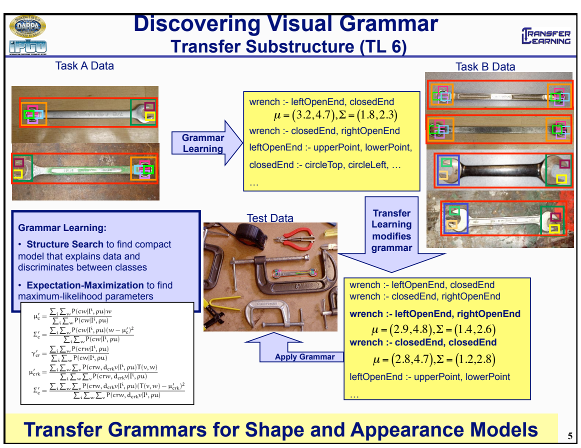
Appendix B: Year 2 Go/NoGo results and scientific summary

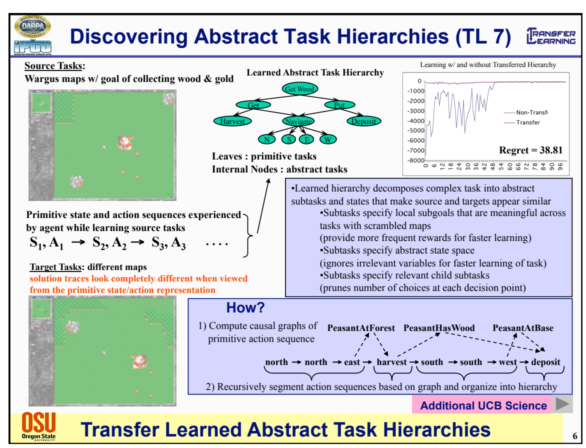


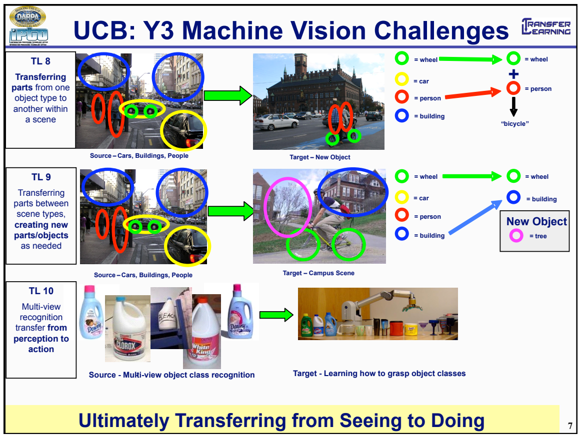


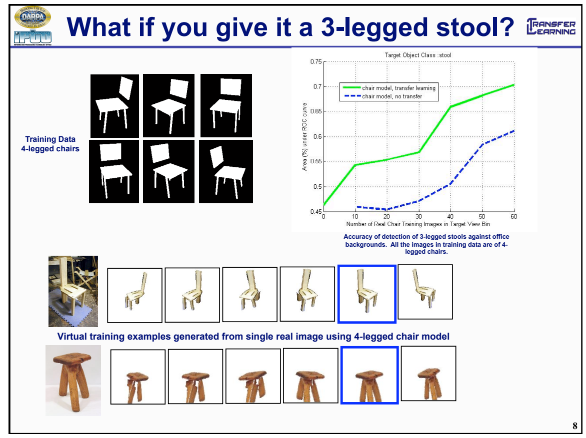


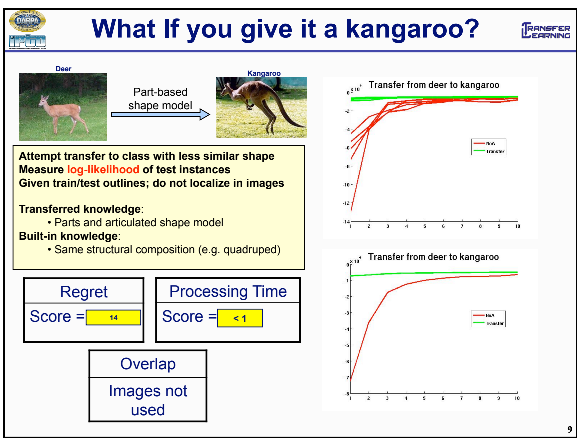


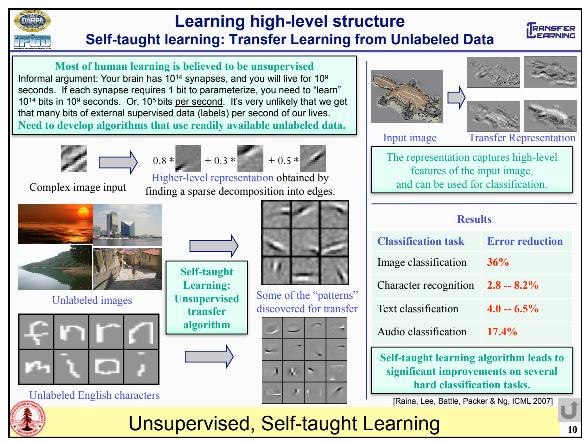


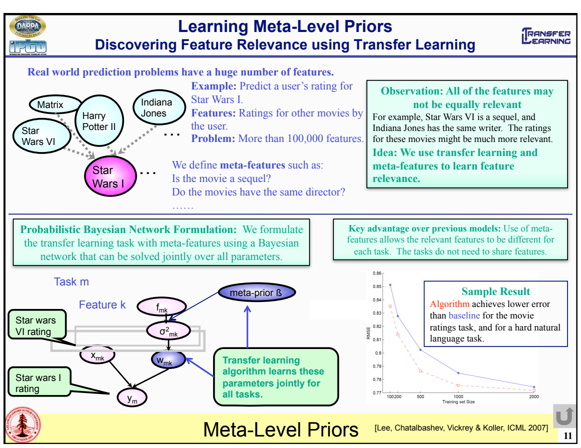


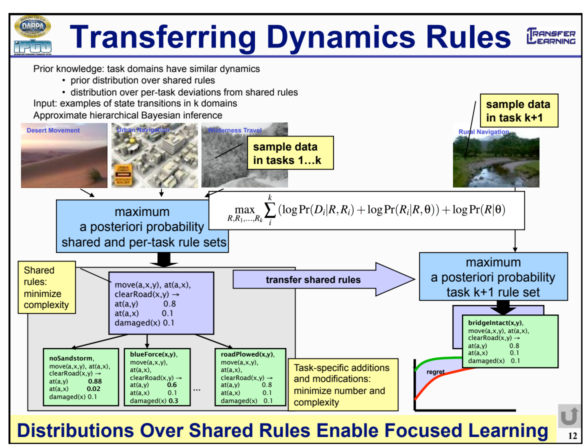

UCB: Y3 Machine Vision Challenges
TL 8
Transferring parts from one object type to another within a scene
Source – Cars, Buildings, People
Target – New Object
= wheel
= car
= person
= building
"bicycle"
TL 9
Transferring parts between scene types, creating new parts/objects as needed
Source – Cars, Buildings, People
Target – Campus Scene
New Object
= tree
TL 10
Multi-view recognition transfer from perception to action
Source - Multi-view object class recognition
Target - Learning how to grasp object classes
Ultimately Transferring from Seeing to Doing
7

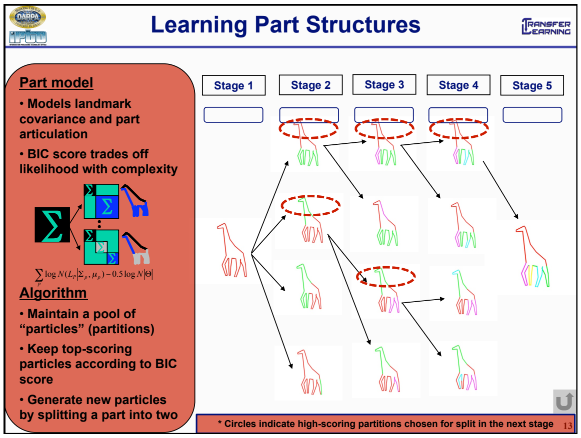

What if you give it a 3-legged stool?
Training Data 4-legged chairs
Target Object Class :stool
chair model, transfer learning
chair model, no transfer
Number of Real Chair Training Images in Target View Bin
Accuracy of detection of 3-legged stools against office backgrounds. All the images in training data are of 4-legged chairs.
Virtual training examples generated from single real image using 4-legged chair model
8

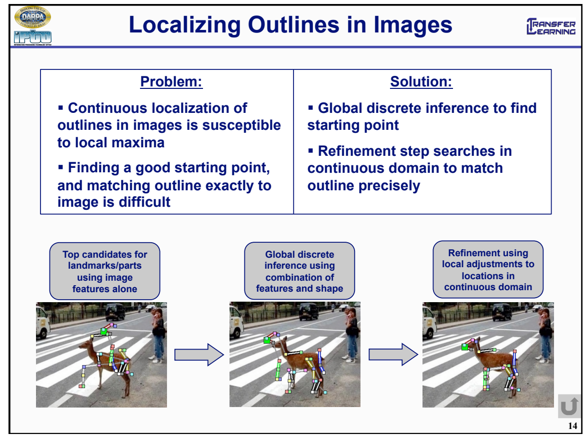

What If you give it a kangaroo?
Deer
Kangaroo
Part-based shape model
Attempt transfer to class with less similar shape
Measure log-likelihood of test instances
Given train/test outlines; do not localize in images
Transferred knowledge:
• Parts and articulated shape model
Built-in knowledge:
• Same structural composition (e.g. quadruped)
Regret
Score = 14
Processing Time
Score = < 1
Overlap
Images not used
Transfer from deer to kangaroo
Transfer from deer to kangaroo
9

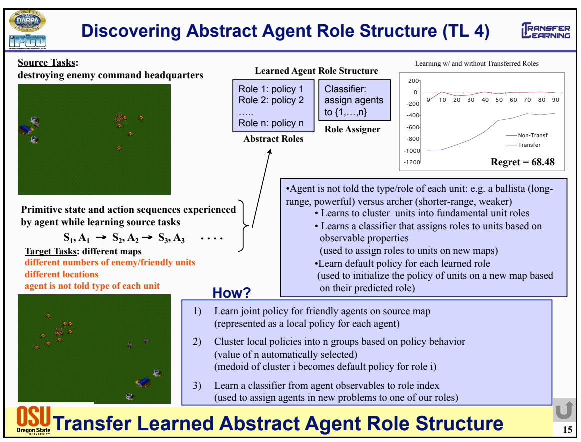

Learning high-level structure
Self-taught learning: Transfer Learning from Unlabeled Data
Most of human learning is believed to be unsupervised
Informal argument: Your brain has 10¹⁴ synapses, and you will live for 10⁹ seconds. If each synapse requires 1 bit to parameterize, you need to "learn" 10¹⁴ bits in 10⁹ seconds. Or, 10⁵ bits per second. It's very unlikely that we get that many bits of external supervised data (labels) per second of our lives.
Need to develop algorithms that use readily available unlabeled data.
Input image
Transfer Representation
The representation captures high-level features of the input image, and can be used for classification.
Complex image input
Higher-level representation obtained by finding a sparse decomposition into edges.
Unlabeled images
Self-taught Learning: Unsupervised transfer algorithm
Some of the "patterns" discovered for transfer
Unlabeled English characters
Results
Classification task
Error reduction
Image classification
36%
Character recognition
2.8 -- 8.2%
Text classification
4.0 -- 6.5%
Audio classification
17.4%
Self-taught learning algorithm leads to significant improvements on several hard classification tasks.
[Raina, Lee, Battle, Packer & Ng, ICML 2007]
Unsupervised, Self-taught Learning
10

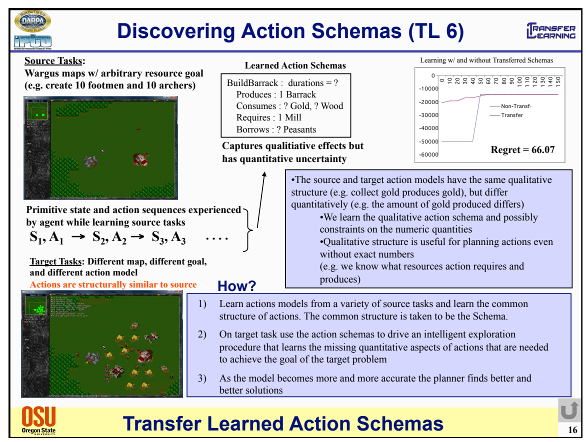

Learning Meta-Level Priors
Discovering Feature Relevance using Transfer Learning
Real world prediction problems have a huge number of features.
Example: Predict a user's rating for Star Wars I.
Features: Ratings for other movies by the user.
Problem: More than 100,000 features.
Matrix
Harry Potter II
Indiana Jones
Star Wars VI
Star Wars I
We define meta-features such as:
Is the movie a sequel?
Do the movies have the same director?
Observation: All of the features may not be equally relevant
For example, Star Wars VI is a sequel, and Indiana Jones has the same writer. The ratings for these movies might be much more relevant.
Idea: We use transfer learning and meta-features to learn feature relevance.
Probabilistic Bayesian Network Formulation: We formulate the transfer learning task with meta-features using a Bayesian network that can be solved jointly over all parameters.
Key advantage over previous models: Use of meta-features allows the relevant features to be different for each task. The tasks do not need to share features.
Task m
Feature k
meta-prior ß
Star wars VI rating
Star wars I rating
Transfer learning algorithm learns these parameters jointly for all tasks.
Sample Result
Algorithm achieves lower error than baseline for the movie ratings task, and for a hard natural language task.
Meta-Level Priors
[Lee, Chatalbashev, Vickrey & Koller, ICML 2007]
11

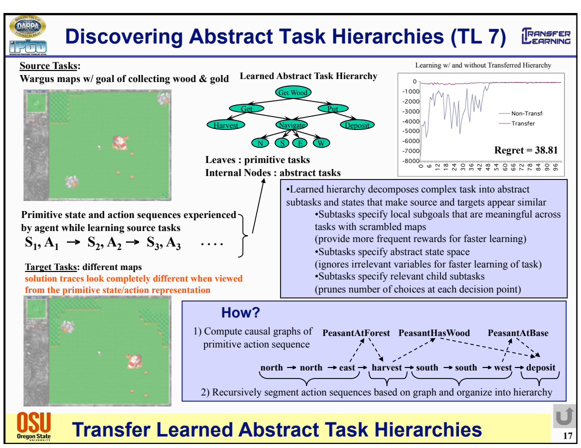

Transferring Dynamics Rules
Prior knowledge: task domains have similar dynamics
• prior distribution over shared rules
• distribution over per-task deviations from shared rules
Input: examples of state transitions in k domains
Approximate hierarchical Bayesian inference
Desert Movement
Urban Navigation
Wilderness Travel
Rural Navigation
sample data in tasks 1…k
sample data in task k+1
maximum a posteriori probability shared and per-task rule sets
maximum a posteriori probability task k+1 rule set
Shared rules: minimize complexity
transfer shared rules
Task-specific additions and modifications: minimize number and complexity
regret
Distributions Over Shared Rules Enable Focused Learning
12

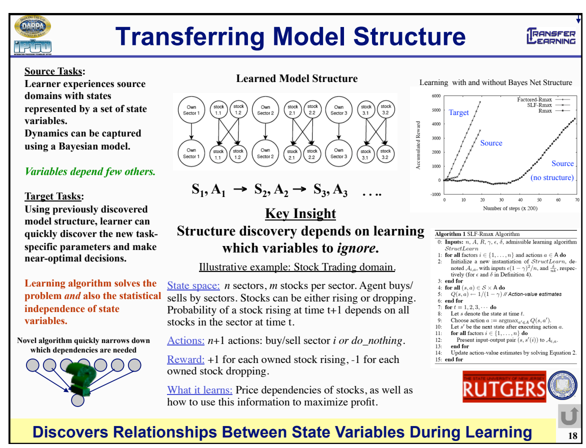

Learning Part Structures
Part model
• Models landmark covariance and part articulation
• BIC score trades off likelihood with complexity
Algorithm
• Maintain a pool of "particles" (partitions)
• Keep top-scoring particles according to BIC score
• Generate new particles by splitting a part into two
Stage 1
Stage 2
Stage 3
Stage 4
Stage 5
* Circles indicate high-scoring partitions chosen for split in the next stage
13


Localizing Outlines in Images
Problem:
▪ Continuous localization of outlines in images is susceptible to local maxima
▪ Finding a good starting point, and matching outline exactly to image is difficult
Solution:
▪ Global discrete inference to find starting point
▪ Refinement step searches in continuous domain to match outline precisely
Top candidates for landmarks/parts using image features alone
Global discrete inference using combination of features and shape
Refinement using local adjustments to locations in continuous domain
14


Discovering Abstract Agent Role Structure (TL 4)
Source Tasks:
destroying enemy command headquarters
Learned Agent Role Structure
Role 1: policy 1
Role 2: policy 2
Role n: policy n
Abstract Roles
Classifier: assign agents to {1,...,n}
Role Assigner
Learning w/ and without Transferred Roles
Regret = 68.48
Primitive state and action sequences experienced by agent while learning source tasks
Target Tasks: different maps
different numbers of enemy/friendly units
different locations
agent is not told type of each unit
•Agent is not told the type/role of each unit: e.g. a ballista (long-range, powerful) versus archer (shorter-range, weaker)
• Learns to cluster units into fundamental unit roles
• Learns a classifier that assigns roles to units based on observable properties
(used to assign roles to units on new maps)
•Learn default policy for each learned role
(used to initialize the policy of units on a new map based on their predicted role)
How?
1) Learn joint policy for friendly agents on source map (represented as a local policy for each agent)
2) Cluster local policies into n groups based on policy behavior (value of n automatically selected) (medoid of cluster i becomes default policy for role i)
3) Learn a classifier from agent observables to role index (used to assign agents in new problems to one of our roles)
Transfer Learned Abstract Agent Role Structure
15


Discovering Action Schemas (TL 6)
Source Tasks:
Wargus maps w/ arbitrary resource goal (e.g. create 10 footmen and 10 archers)
Learned Action Schemas
BuildBarrack : durations = ?
Produces : 1 Barrack
Consumes : ? Gold, ? Wood
Requires : 1 Mill
Borrows : ? Peasants
Captures qualitiative effects but has quantitative uncertainty
Learning w/ and without Transferred Schemas
Regret = 66.07
Primitive state and action sequences experienced by agent while learning source tasks
Target Tasks: Different map, different goal, and different action model
Actions are structurally similar to source
•The source and target action models have the same qualitative structure (e.g. collect gold produces gold), but differ quantitatively (e.g. the amount of gold produced differs)
•We learn the qualitative action schema and possibly constraints on the numeric quantities
•Qualitative structure is useful for planning actions even without exact numbers
(e.g. we know what resources action requires and produces)
How?
1) Learn actions models from a variety of source tasks and learn the common structure of actions. The common structure is taken to be the Schema.
2) On target task use the action schemas to drive an intelligent exploration procedure that learns the missing quantitative aspects of actions that are needed to achieve the goal of the target problem
3) As the model becomes more and more accurate the planner finds better and better solutions
Transfer Learned Action Schemas
16

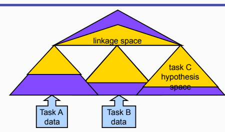

Discovering Abstract Task Hierarchies (TL 7)
Source Tasks:
Wargus maps w/ goal of collecting wood & gold
Learned Abstract Task Hierarchy
Get Wood
Get
Put
Harvest
Navigate
Deposit
N
S
E
W
Leaves : primitive tasks
Internal Nodes : abstract tasks
Learning w/ and without Transferred Hierarchy
Regret = 38.81
Primitive state and action sequences experienced by agent while learning source tasks
Target Tasks: different maps
solution traces look completely different when viewed from the primitive state/action representation
•Learned hierarchy decomposes complex task into abstract subtasks and states that make source and targets appear similar
•Subtasks specify local subgoals that are meaningful across tasks with scrambled maps
(provide more frequent rewards for faster learning)
•Subtasks specify abstract state space
(ignores irrelevant variables for faster learning of task)
•Subtasks specify relevant child subtasks
(prunes number of choices at each decision point)
How?
1) Compute causal graphs of primitive action sequence
PeasantAtForest PeasantHasWood PeasantAtBase
north → north → east → harvest → south → south → west → deposit
2) Recursively segment action sequences based on graph and organize into hierarchy
Transfer Learned Abstract Task Hierarchies
17


Transferring Model Structure
Source Tasks:
Learner experiences source domains with states represented by a set of state variables.
Dynamics can be captured using a Bayesian model.
Variables depend few others.
Learned Model Structure
Learning with and without Bayes Net Structure
Target
Source
Source (no structure)
Target Tasks:
Using previously discovered model structure, learner can quickly discover the new task-specific parameters and make near-optimal decisions.
Key Insight
Structure discovery depends on learning which variables to ignore.
Learning algorithm solves the problem and also the statistical independence of state variables.
Novel algorithm quickly narrows down which dependencies are needed
Illustrative example: Stock Trading domain.
State space: n sectors, m stocks per sector. Agent buys/sells by sectors. Stocks can be either rising or dropping. Probability of a stock rising at time t+1 depends on all stocks in the sector at time t.
Actions: n+1 actions: buy/sell sector i or do_nothing.
Reward: +1 for each owned stock rising, -1 for each owned stock dropping.
What it learns: Price dependencies of stocks, as well as how to use this information to maximize profit.
RUTGERS
Discovers Relationships Between State Variables During Learning
18

Foundations of Transfer Learning

Adversarial formulation of prediction problems:

- Measure performance via *regret* relative to best model.
- Eliminates the need for probabilistic assumptions on the structure to be transferred.



Multitask prediction with expert advice

- Structure to be transferred: *small set of effective experts (=prediction strategies)*
- Similar approach + performance guarantees for transferring *features* between prediction problems
- Efficient algorithm, optimal regret rate:

$$\textit{loss} \le \textit{optimal loss} + T \log m + m \log |E|/m$$

where:

E = (very large) set of experts

loss = cumulative prediction error (e.g., number of misclassifications),

optimal loss = best performance by a subset of *m* experts in *E*

T= number of tasks

Analysis of Bayesian model averaging for prediction

- For transfer learning in regression, density estimation problems
- Measure performance via *regret* relative to any comparison predictor in model
- Regret depends on properties of the Bayesian prior (weight & smoothness near comparison predictor)
- The advance: Performance guarantees are relative to best in model – not assuming `correctness' of Bayesian prior.

When is transfer possible and how is it best achieved?

19

Abstract Lookahead, Angelic Semantics

1) Given an approximate value function, lookahead gives much more effective decision making [cf chess programs]

2) Hierarchy with high-level actions (HLAs) should support effective planning over long time scales

Open problem since 1977 (Tate's HTN planning)

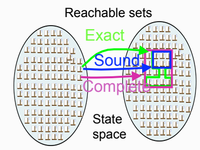
James Allen (2000) "*The semantics of hierarchical planning have never been clarified … no one has ever figured out how to reconcile the semantics of hierarchical plans with the semantics of primitive actions*"

Hence no models for HLAs, no learning method and no transfer

Why? HLAs have many possible refinements, so effects vary

Idea: reachable set = set of all states reachable by *any* refinement

Plan works iff its reachable set *intersects* goal set



Enables new planning algorithms with very strong properties: first planner able to generate provably correct abstract plans without refining to primitive actions, and to prune provably incorrect abstract plans

100X-1000X increases in efficiency compared to classical non-hierarchical and HTN planners

New generalized definition of admissible heuristics for reachable sets

First provably optimal hierarchical planning algorithm

First algorithm for online hierarchical planning and action selection

Dramatically better performance than previous methods (eg LRTA*)

Next steps: apply in Wargus, develop learning and transfer methods for HLA models, extend to stochastic case

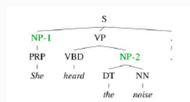
OSU Oregon State University

Efficient planning over very long time scales

20

Nonparametric grammars

- Probabilistic context-free grammars (PCFGs) model the syntax of language, which is an important first step for many NLP applications
- Our **hierarchical** Bayesian nonparametric model allows automatic selection of grammar complexity, providing robustness to overfitting
- Leverage **transfer learning** to share power between different rules in the grammar
- Variational inference algorithm
  - **Fast training**: minimal computational overhead over ordinary PCFG training
  - **Modularity**: Dirichlet process component plugs into existing parsers



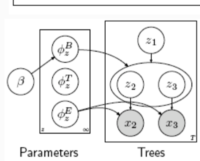
HDP-PCFG

$\beta \sim \mathrm{GEM}(\alpha)$ [states]

$\phi_z^E \sim \mathrm{Dirichlet}(\alpha^E)$ [emissions]

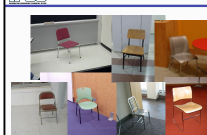
$\phi_z^B \sim \mathrm{DP}(\alpha^B, \beta\beta^T)$ [productions]

$(z_{L(i)}, z_{R(i)}) \sim \mathrm{Multinomial}(\phi_{z_i}^B)$ [nodes]

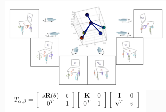


21

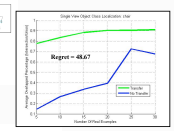
Transfer Across Views via Structure Model

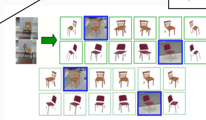


Different members of a single class in viewpoints 1 .. k

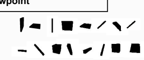


$$T_{\alpha,\beta} = \begin{bmatrix} s\mathbf{R}(\theta) & \mathbf{t} \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{K} & 0 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{v}^T & v \end{bmatrix}$$





Members of the same class in new viewpoint



Shape library

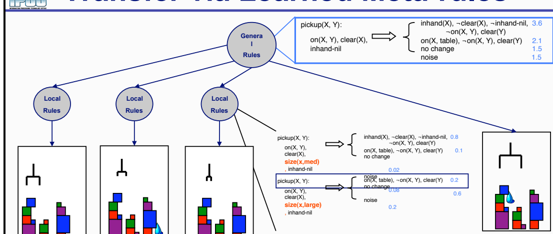
- Phase 0: generate view-to-view transform library from shape library
- From source tasks:
  - estimate 3D part centroids
  - compute posterior distribution on view-to-view transforms
  - estimate shape distribution for parts from transforms
- In target task:
  - refine transforms given small amount of data in new view, if available
  - use transforms to map data in other views to the new view
  - feed transformed data into any 2D recognition system

CSAIL

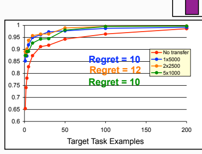
Transfer Distributions of 2D views of 3D Composite Objects

22

Transfer via Learned Meta-rules



- Individual rule set learned for each source task
- Meta rule-set captures regularities in source task
- Meta rule-set provides strong bias for learning in target task
- Easy to modify meta rule-set to depend on gripper size in new task
- Compared different numbers and sizes of source tasks
- Applicable, eventually, to learning domain models for strategy games



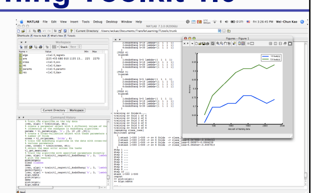
CSAIL

Rules Generalized from Source, Specialized for Target

23

Transfer Learning Toolkit 1.0

- For general-purpose application of multitask data analysis methods
- Domain independent



Algorithms:

- l_1-l_2 regularization
- Parametric Bayes
- Parametric empirical Bayes
- Ando-Zhang support vector transfer
- Nonparametric Bayes: hierarchical Dirichlet process
- Meta-level prior for feature relevance
- Feature transfer for online prediction

Datasets:

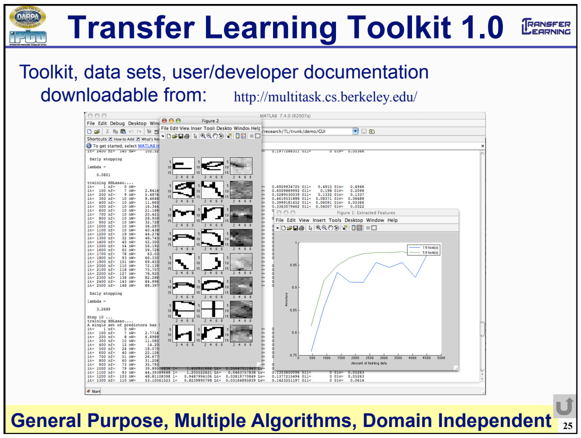
- Hand-written digits
- Hand-written letters
- Reuters part-of-speech tagging
- Multi-language named entity classification
- Netflix movie preferences
- Robot grasp point prediction

Utilities:

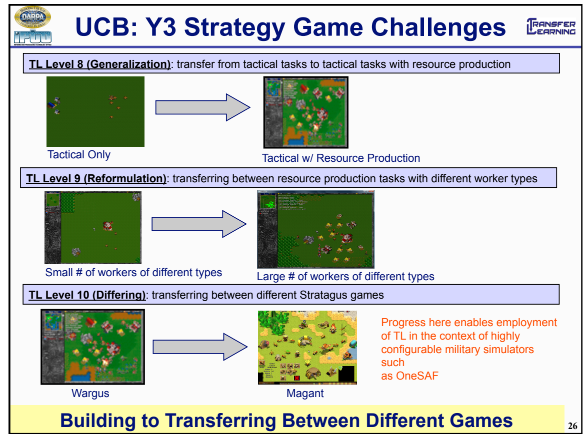
- single task algorithms
  - kernel methods
  - decision trees
  - Boosting
  - k-NN, ...
- cross-validation
- data visualization
- transfer metrics

General Purpose, Multiple Algorithms, Domain Independent

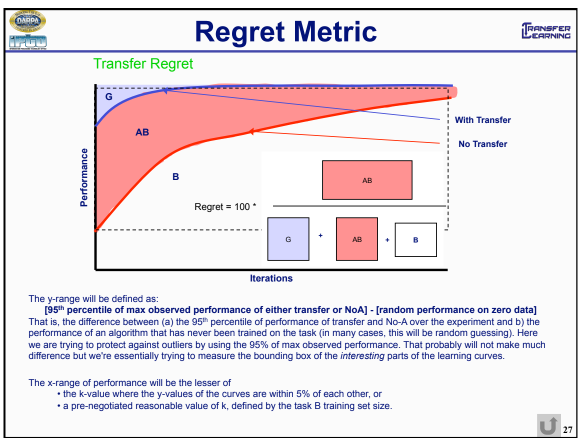
24



Transfer Learning Toolkit 1.0
Toolkit, data sets, user/developer documentation
downloadable from: http://multitask.cs.berkeley.edu/
General Purpose, Multiple Algorithms, Domain Independent
25

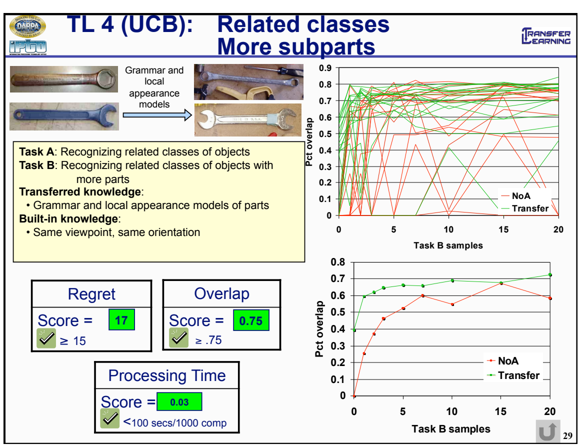


UCB: Y3 Strategy Game Challenges
TL Level 8 (Generalization): transfer from tactical tasks to tactical tasks with resource production
Tactical Only
Tactical w/ Resource Production
TL Level 9 (Reformulation): transferring between resource production tasks with different worker types
Small # of workers of different types
Large # of workers of different types
TL Level 10 (Differing): transferring between different Stratagus games
Wargus
Magant
Progress here enables employment of TL in the context of highly configurable military simulators such as OneSAF
Building to Transferring Between Different Games
26

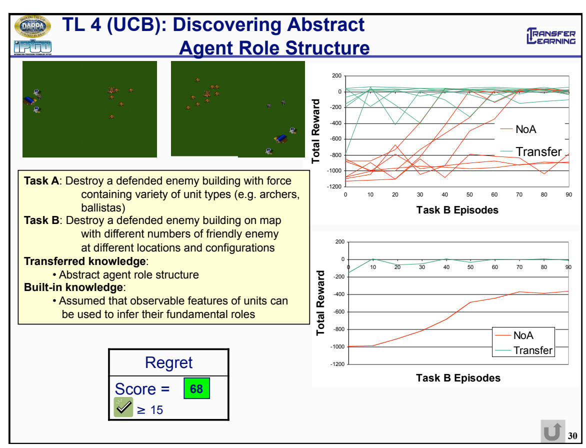


Regret Metric
Transfer Regret
G
AB
B
With Transfer
No Transfer
Performance
Iterations
Regret = 100 *
AB
G
+
AB
+
B
The y-range will be defined as:
[95th percentile of max observed performance of either transfer or NoA] - [random performance on zero data]
That is, the difference between (a) the 95th percentile of performance of transfer and No-A over the experiment and b) the performance of an algorithm that has never been trained on the task (in many cases, this will be random guessing). Here we are trying to protect against outliers by using the 95% of max observed performance. That probably will not make much difference but we're essentially trying to measure the bounding box of the interesting parts of the learning curves.
The x-range of performance will be the lesser of
• the k-value where the y-values of the curves are within 5% of each other, or
• a pre-negotiated reasonable value of k, defined by the task B training set size.
27

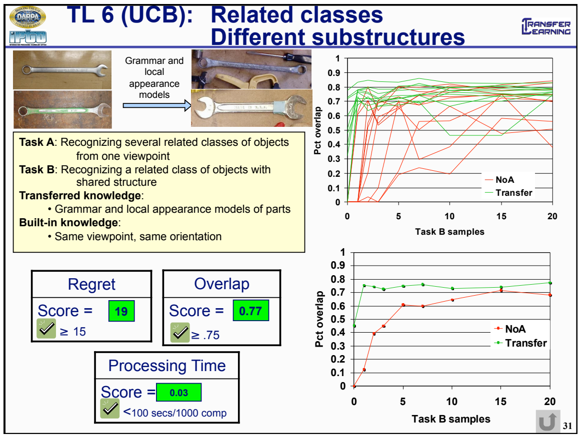
Go/NoGo Testing Details

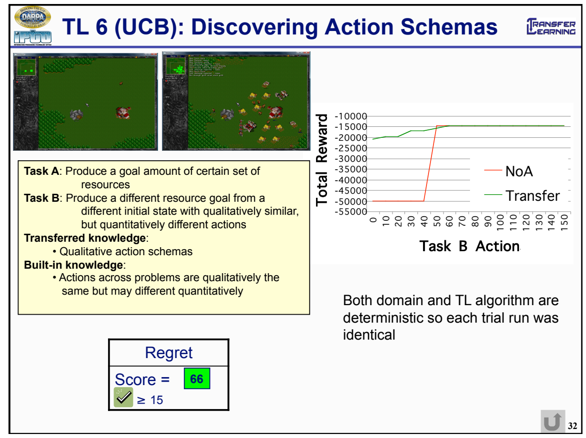


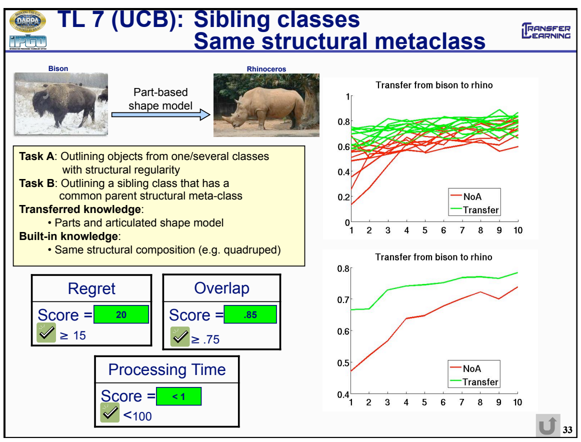
TL 4 (UCB): Related classes More subparts
Grammar and local appearance models
Task A: Recognizing related classes of objects
Task B: Recognizing related classes of objects with more parts
Transferred knowledge:
• Grammar and local appearance models of parts
Built-in knowledge:
• Same viewpoint, same orientation
Regret
Score = 17
≥ 15
Overlap
Score = 0.75
≥ .75
Processing Time
Score = 0.03
<100 secs/1000 comp
Pct overlap
Task B samples
NoA
Transfer
29

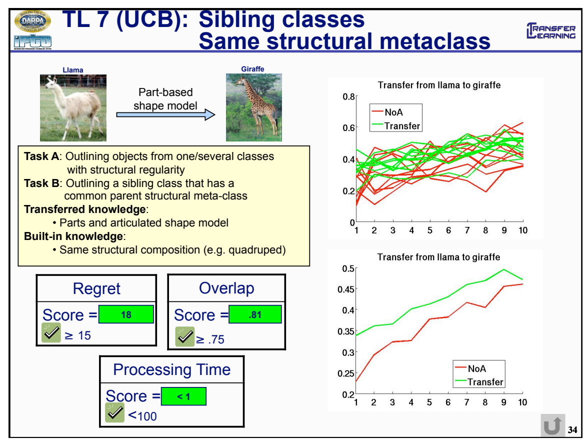


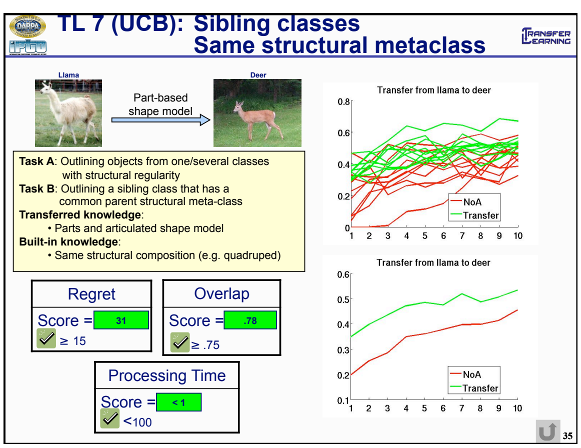
TL 4 (UCB): Discovering Abstract Agent Role Structure
Task A: Destroy a defended enemy building with force containing variety of unit types (e.g. archers, ballistas)
Task B: Destroy a defended enemy building on map with different numbers of friendly enemy at different locations and configurations
Transferred knowledge:
• Abstract agent role structure
Built-in knowledge:
• Assumed that observable features of units can be used to infer their fundamental roles
Regret
Score = 68
≥ 15
Total Reward
Task B Episodes
NoA
Transfer
30

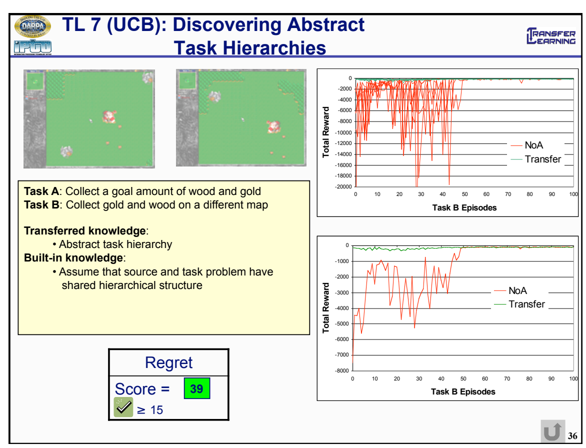

TL 6 (UCB): Related classes
Different substructures
Grammar and local appearance models
Task A: Recognizing several related classes of objects from one viewpoint
Task B: Recognizing a related class of objects with shared structure
Transferred knowledge:
• Grammar and local appearance models of parts
Built-in knowledge:
• Same viewpoint, same orientation
Regret
Score = 19
≥ 15
Overlap
Score = 0.77
≥ .75
Processing Time
Score = 0.03
<100 secs/1000 comp
Pct overlap
Task B samples
NoA
Transfer
31


TL 6 (UCB): Discovering Action Schemas
Task A: Produce a goal amount of certain set of resources
Task B: Produce a different resource goal from a different initial state with qualitatively similar, but quantitatively different actions
Transferred knowledge:
• Qualitative action schemas
Built-in knowledge:
• Actions across problems are qualitatively the same but may different quantitatively
Regret
Score = 66
≥ 15
Total Reward
Task B Action
NoA
Transfer
Both domain and TL algorithm are deterministic so each trial run was identical
32

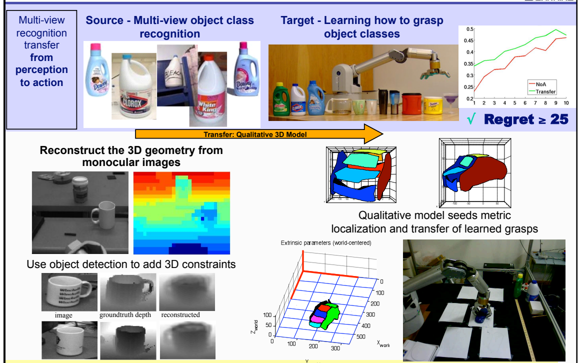

TL 7 (UCB): Sibling classes
Same structural metaclass
Bison
Part-based shape model
Rhinoceros
Task A: Outlining objects from one/several classes with structural regularity
Task B: Outlining a sibling class that has a common parent structural meta-class
Transferred knowledge:
• Parts and articulated shape model
Built-in knowledge:
• Same structural composition (e.g. quadruped)
Regret
Score = 20
≥ 15
Overlap
Score = .85
≥ .75
Processing Time
Score = < 1
<100
Transfer from bison to rhino
NoA
Transfer
33


TL 7 (UCB): Sibling classes
Same structural metaclass
Llama
Part-based shape model
Giraffe
Task A: Outlining objects from one/several classes with structural regularity
Task B: Outlining a sibling class that has a common parent structural meta-class
Transferred knowledge:
• Parts and articulated shape model
Built-in knowledge:
• Same structural composition (e.g. quadruped)
Regret
Score = 18
≥ 15
Overlap
Score = .81
≥ .75
Processing Time
Score = < 1
<100
Transfer from llama to giraffe
NoA
Transfer
34


TL 7 (UCB): Sibling classes
Same structural metaclass
Llama
Part-based shape model
Deer
Task A: Outlining objects from one/several classes with structural regularity
Task B: Outlining a sibling class that has a common parent structural meta-class
Transferred knowledge:
• Parts and articulated shape model
Built-in knowledge:
• Same structural composition (e.g. quadruped)
Regret
Score = 31
≥ 15
Overlap
Score = .78
≥ .75
Processing Time
Score = < 1
<100
Transfer from llama to deer
NoA
Transfer
35


TL 7 (UCB): Discovering Abstract
Task Hierarchies
Task A: Collect a goal amount of wood and gold
Task B: Collect gold and wood on a different map
Transferred knowledge:
• Abstract task hierarchy
Built-in knowledge:
• Assume that source and task problem have shared hierarchical structure
Regret
Score = 39
≥ 15
Total Reward
Task B Episodes
NoA
Transfer
36

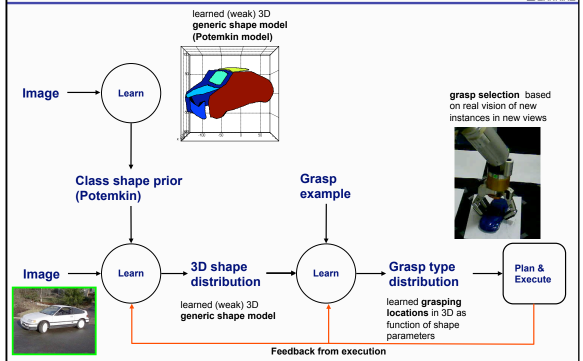
UCB'S Year 3 Go/NoGo



From Monoscopic Views to 3D Manipulation

1

Transfer Using Learned Shape Models

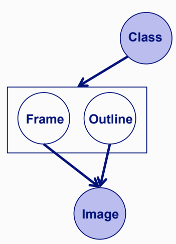


Learned 3D Shape Models Speed Manipulation of New Objects

2

Learning in Task A

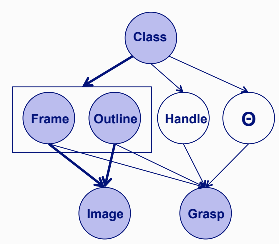
- Given images of objects of known class (shaded variables)
- Learn coordinate frames and outlines for each part (unshaded variables in plate – indicating that there are repeated for each part)
- This is the 3D Potemkin model for the class



3

Learning in Task AB

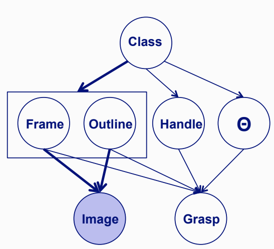
- Given shaded variables for each training instance
- Learn which part to grasp (Handle)
- Learn location of grasp relative to part (Θ)
- To maximize probability of observed training Grasps



4

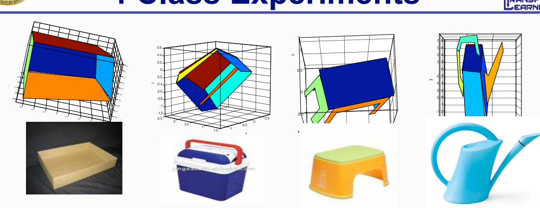
Performing Task B

- Given image of test case:
- First, pick best Class value (based on Outline)
- Then, pick best Grasp, given Class



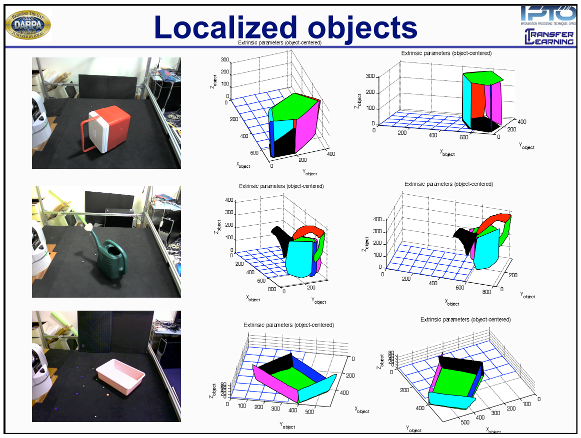
5

4 Class Experiments

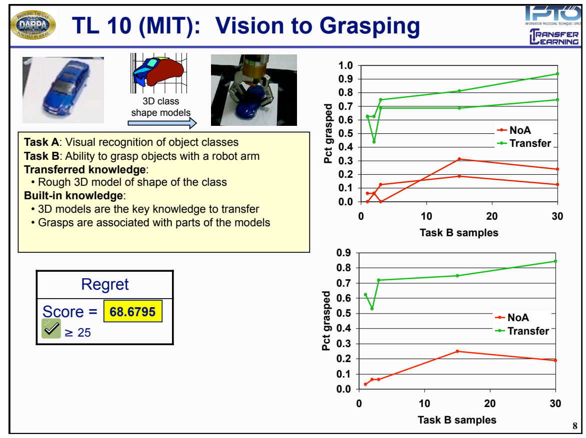


- Train:
  - 5 objects per class (3 train, 2 test)
  - 10 poses per training object
  - 1 grasp demonstrated per object
- Test:
  - 2 objects x 2 poses x 4 classes
  - 16 tests per point on learning curve

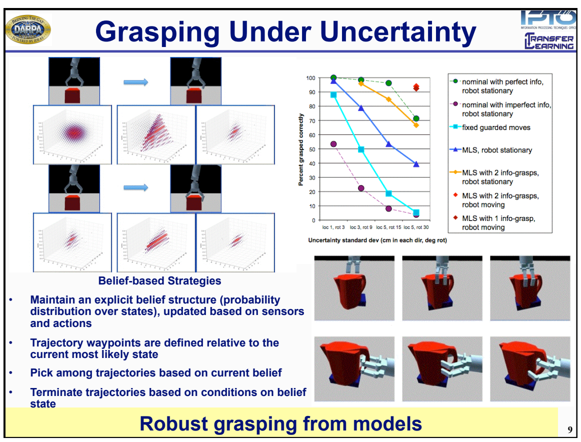
6



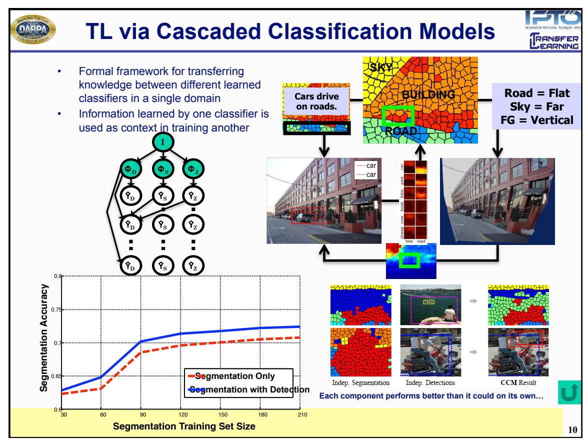
Localized objects
Extrinsic parameters (object-centered)



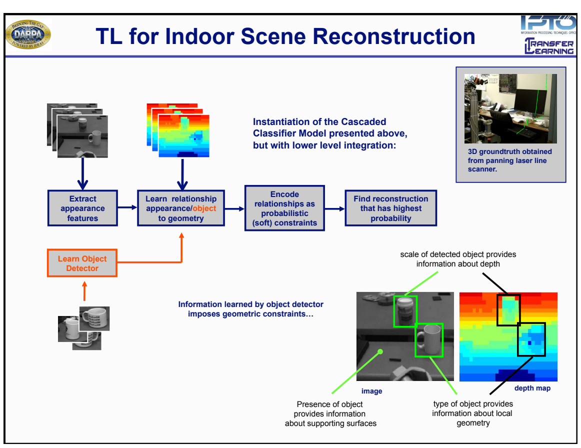
TL 10 (MIT): Vision to Grasping
3D class shape models
Task A: Visual recognition of object classes
Task B: Ability to grasp objects with a robot arm
Transferred knowledge:
• Rough 3D model of shape of the class
Built-in knowledge:
• 3D models are the key knowledge to transfer
• Grasps are associated with parts of the models
Regret
Score = 68.6795
≥ 25
Pct grasped
NoA
Transfer
Task B samples
8



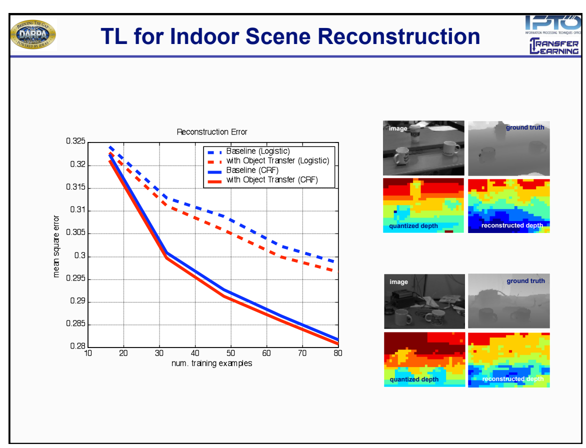
Grasping Under Uncertainty
Belief-based Strategies
• Maintain an explicit belief structure (probability distribution over states), updated based on sensors and actions
• Trajectory waypoints are defined relative to the current most likely state
• Pick among trajectories based on current belief
• Terminate trajectories based on conditions on belief state
Percent grasped correctly
nominal with perfect info, robot stationary
nominal with imperfect info, robot stationary
fixed guarded moves
MLS, robot stationary
MLS with 2 info-grasps, robot stationary
MLS with 2 info-grasps, robot moving
MLS with 1 info-grasp, robot moving
Uncertainty standard dev (cm in each dir, deg rot)
Robust grasping from models
9



TL via Cascaded Classification Models
• Formal framework for transferring knowledge between different learned classifiers in a single domain
• Information learned by one classifier is used as context in training another
Cars drive on roads.
SKY
BUILDING
ROAD
Road = Flat
Sky = Far
FG = Vertical
Segmentation Accuracy
Segmentation Only
Segmentation with Detection
Segmentation Training Set Size
Indep. Segmentation
Indep. Detections
CCM Result
Each component performs better than it could on its own…
10

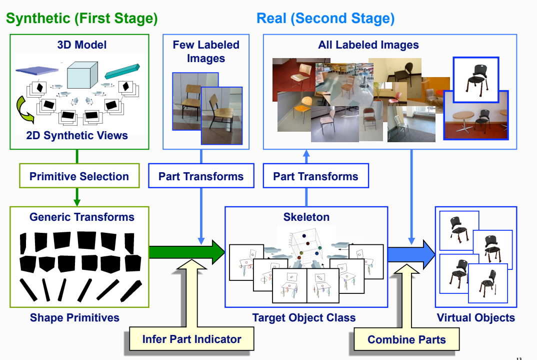


TL for Indoor Scene Reconstruction
Instantiation of the Cascaded Classifier Model presented above, but with lower level integration:
3D groundtruth obtained from panning laser line scanner.
Extract appearance features
Learn relationship appearance/object to geometry
Encode relationships as probabilistic (soft) constraints
Find reconstruction that has highest probability
Learn Object Detector
Information learned by object detector imposes geometric constraints…
scale of detected object provides information about depth
image
depth map
Presence of object provides information about supporting surfaces
type of object provides information about local geometry



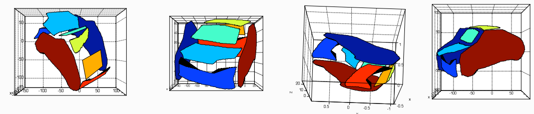
TL for Indoor Scene Reconstruction
Reconstruction Error
Baseline (Logistic)
with Object Transfer (Logistic)
Baseline (CRF)
with Object Transfer (CRF)
mean square error
num. training examples
image
ground truth
quantized depth
reconstructed depth

The Potemkin Model

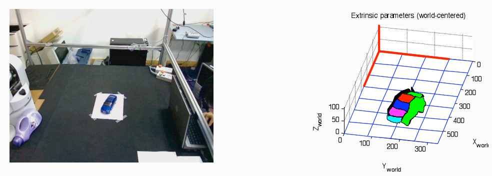


13

From Recognition to Manipulation



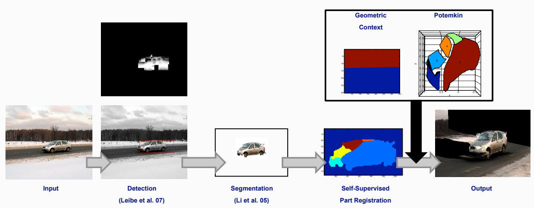
Potemkin model for Car class: acquired from a few monocular images



3D Class-Specific Reconstruction from a Single Image

3D Object Popup

- Automatic (Detection, Segmentation, Part Registration)
- 3D Class-Specific Reconstruction from a Single 2D Image



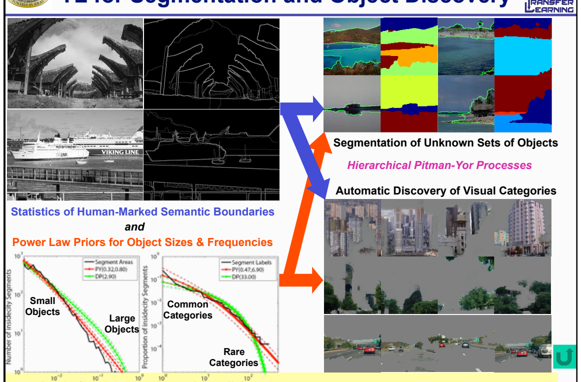
15

The Emerging Science of TL

- **Distant tasks require general knowledge**
  - As tasks become more distinct (higher transfer levels), the form of the knowledge learned and transferred needs to become more general purpose.
  - For example, we can learn to improve object recognition or grasping or bicycle riding or foraging by adjusting low-level parameters; but transferring from one to the other requires higher-level knowledge like causal or geometric models.
- **Meta learning is crucial**
  - There are too many possible aspects of transfer to know how, in general, to move from one single task to another.
  - Multiple training tasks allow learning of **kinds** of regularities that are likely to hold across tasks, which guides transfer to novel tasks by prioritizing hypothesized similarities.
- **Hierarchical Bayes is foundational**
  - It allows integration of prior knowledge and data from multiple sources and maintains receptivity to new information.
    - Very rich and flexible classes of hypotheses, including sets of logical rules, meta-features, geometric models, hierarchical control strategies
    - Hypothesis complexity automatically adapted based on amount and diversity of available data; for example, flexible clustering of previously-seen individuals speeds transfer by "soft assignment" of new individual to clusters

16

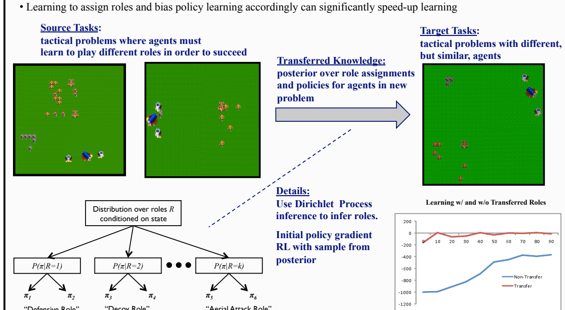
TL for Segmentation and Object Discovery



17

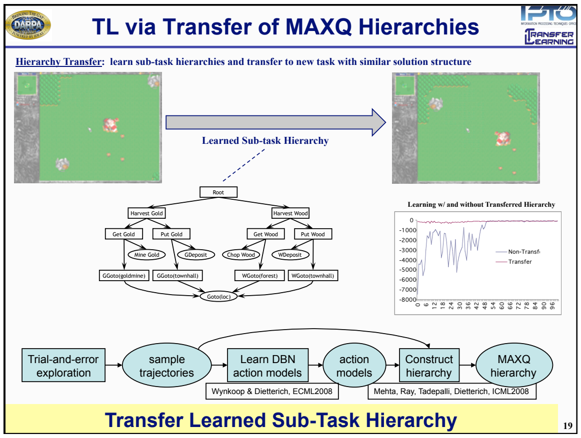
TL for Role Transfer in Multi-Agent RL

- In multi-agent RL, different agents in different situations play different roles (e.g. decoys, defense, offense, support)
- The role of an agent strong influences the agent's policy.
- Learning to assign roles and bias policy learning accordingly can significantly speed-up learning

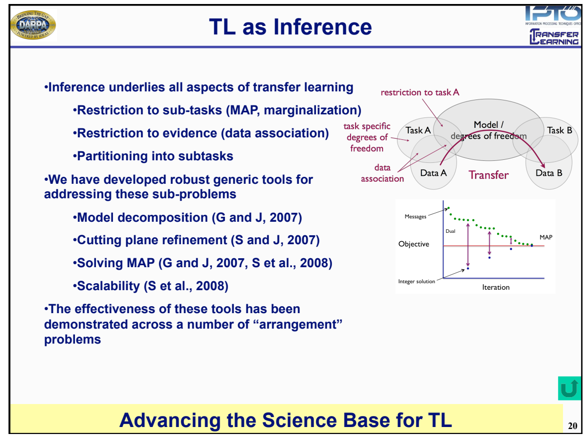


Transfer Learned Abstract Agent Role Structure

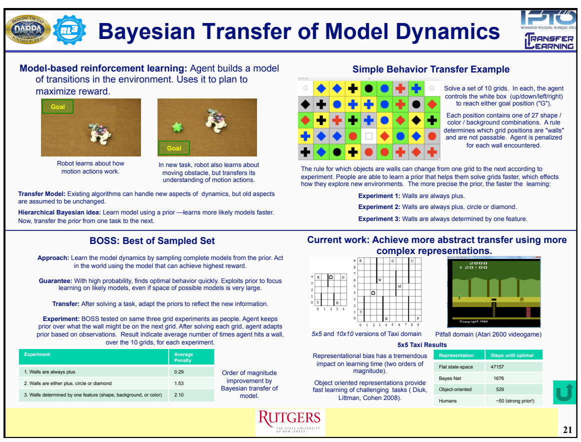
18



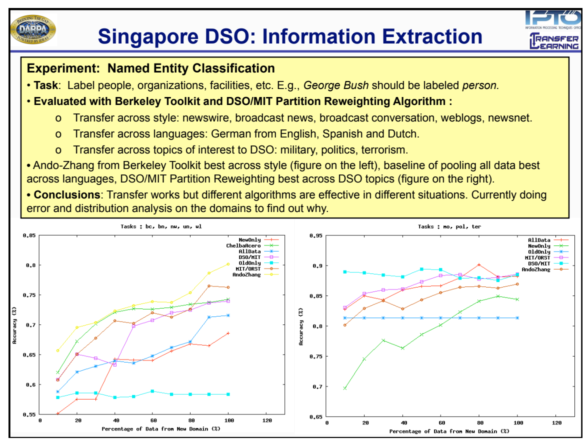
TL via Transfer of MAXQ Hierarchies
Hierarchy Transfer: learn sub-task hierarchies and transfer to new task with similar solution structure
Learned Sub-task Hierarchy
Root
Harvest Gold
Harvest Wood
Get Gold
Put Gold
Get Wood
Put Wood
Mine Gold
GDeposit
Chop Wood
WDeposit
GGoto(goldmine)
GGoto(townhall)
WGoto(forest)
WGoto(townhall)
Goto(loc)
Learning w/ and without Transferred Hierarchy
Non-Transfer
Transfer
Trial-and-error exploration
sample trajectories
Learn DBN action models
action models
Construct hierarchy
MAXQ hierarchy
Wynkoop & Dietterich, ECML2008
Mehta, Ray, Tadepalli, Dietterich, ICML2008
Transfer Learned Sub-Task Hierarchy
19



TL as Inference
•Inference underlies all aspects of transfer learning
•Restriction to sub-tasks (MAP, marginalization)
•Restriction to evidence (data association)
•Partitioning into subtasks
•We have developed robust generic tools for addressing these sub-problems
•Model decomposition (G and J, 2007)
•Cutting plane refinement (S and J, 2007)
•Solving MAP (G and J, 2007, S et al., 2008)
•Scalability (S et al., 2008)
•The effectiveness of these tools has been demonstrated across a number of "arrangement" problems
restriction to task A
task specific degrees of freedom
Task A
Model / degrees of freedom
Task B
data association
Data A
Transfer
Data B
Messages
Objective
MAP
Integer solution
Iteration
Advancing the Science Base for TL
20



Bayesian Transfer of Model Dynamics
Model-based reinforcement learning: Agent builds a model of transitions in the environment. Uses it to plan to maximize reward.
Goal
Goal
Robot learns about how motion actions work.
In new task, robot also learns about moving obstacle, but transfers its understanding of motion actions.
Transfer Model: Existing algorithms can handle new aspects of dynamics, but old aspects are assumed to be unchanged.
Hierarchical Bayesian idea: Learn model using a prior —learns more likely models faster. Now, transfer the prior from one task to the next.
Simple Behavior Transfer Example
Solve a set of 10 grids. In each, the agent controls the white box (up/down/left/right) to reach either goal position ("G").
Each position contains one of 27 shape / color / background combinations. A rule determines which grid positions are "walls" and are not passable. Agent is penalized for each wall encountered.
The rule for which objects are walls can change from one grid to the next according to experiment. People are able to learn a prior that helps them solve grids faster, which effects how they explore new environments. The more precise the prior, the faster the learning:
Experiment 1: Walls are always plus.
Experiment 2: Walls are always plus, circle or diamond.
Experiment 3: Walls are always determined by one feature.
BOSS: Best of Sampled Set
Approach: Learn the model dynamics by sampling complete models from the prior. Act in the world using the model that can achieve highest reward.
Guarantee: With high probability, finds optimal behavior quickly. Exploits prior to focus learning on likely models, even if space of possible models is very large.
Transfer: After solving a task, adapt the priors to reflect the new information.
Experiment: BOSS tested on same three grid experiments as people. Agent keeps prior over what the wall might be on the next grid. After solving each grid, agent adapts prior based on observations. Result indicate average number of times agent hits a wall, over the 10 grids, for each experiment.
Experiment | Average Penalty
1. Walls are always plus | 0.26
2. Walls are either plus, circle or diamond | 1.53
3. Walls determined by one feature (shape, background, or color) | 2.10
Order of magnitude improvement by Bayesian transfer of model.
Current work: Achieve more abstract transfer using more complex representations.
5x5 and 10x10 versions of Taxi domain
Pitfall domain (Atari 2600 videogame)
5x5 Taxi Results
Representational bias has a tremendous impact on learning time (two orders of magnitude).
Object oriented representations provide fast learning of challenging tasks (Diuk, Littman, Cohen 2008).
Representation | Steps until optimal
Flat state-space | 47157
Bayes Net | 1676
Object-oriented | 529
Humans | ~50 (strong prior!)
RUTGERS
21



Singapore DSO: Information Extraction
Experiment: Named Entity Classification
• Task: Label people, organizations, facilities, etc. E.g., George Bush should be labeled person.
• Evaluated with Berkeley Toolkit and DSO/MIT Partition Reweighting Algorithm :
o Transfer across style: newswire, broadcast news, broadcast conversation, weblogs, newsnet.
o Transfer across languages: German from English, Spanish and Dutch.
o Transfer across topics of interest to DSO: military, politics, terrorism.
• Ando-Zhang from Berkeley Toolkit best across style (figure on the left), baseline of pooling all data best across languages, DSO/MIT Partition Reweighting best across DSO topics (figure on the right).
• Conclusions: Transfer works but different algorithms are effective in different situations. Currently doing error and distribution analysis on the domains to find out why.
Percentage of Data from New Domain (%)
Accuracy (%)