



Generalizing and Categorizing Skills in Reinforcement Learning Agents Using Partial Policy Homomorphisms

Srividhya Rajendran and Manfred Huber

Department of Computer Science and Engineering, The University of Texas at Arlington, P.O. Box 19015, Arlington, TX 76019

Emails: srividhya.rajendran@mavs.uta.edu, huber@cse.uta.edu

Abstract

A reinforcement learning (RL) agent involved in life-long learning in a complex and dynamic environment has to have the ability to utilize control knowledge acquired in one situation in novel contexts. As part of this, it is important for the learning agent not only to be able to learn a new skill for a specific instance of a task but also to identify similar tasks, form a reusable skill and representational abstractions for the corresponding *task type*, and to apply these abstractions in new, previously unseen contexts. This poster presents a new approach to policy generalization that derives an abstract policy for a set of similar tasks (a *task type*) by constructing a partial policy homomorphism from a set of basic policies learned for previously seen task instances. The resulting generalized policy can then be applied in new contexts to address new instances of similar tasks. As opposed to many recent approaches in lifelong learning systems, this approach allows to identify similar tasks based on the functional characteristics of the corresponding skills and provides a means of transferring the learned knowledge to new situations without the need for complete knowledge of the state space and the system dynamics in the new environment.

To illustrate the new policy generalization method and to demonstrate its ability to reuse the gained knowledge in new contexts, it is applied to a set of grid world examples.

Introduction

Life long learning RL agents face a number of challenges that need to be addressed in order for them to perform successfully in a complex and dynamic environment. Some of the issues faced by these agents are:

- The number of policies learned by these agents become intractable and many of them are of limited use as they are heavily dependent on the environment and thus do not transfer to new tasks and environments.
- Processing of the perceptual data and basing decisions on them in real time without any prior knowledge as to what aspects of the data to focus attention on becomes computationally intractable in complex, real world environments.
- Reasoning in real time about actions that the agent needs to perform at each point in time becomes impossible without the availability of prior control knowledge as the complexity of the task increases.

As a result these agents need mechanisms that provide them with the ability to:

- Continuously learn and adapt.
- Learn increasingly complex task.
- Reuse the knowledge gained.
- Perform in real time.

Biological systems face similar problems but still succeed at learning to perform increasingly complex task over time and to perform successfully in complex and dynamic environments. Developmental theories [5,6] suggest that these biological systems overcome their problems by:

- Learning to abstract important information while ignoring the rest.
- Learning to abstract reusable skills and representations.
- Using the formed skill and representational abstractions to learn more complex tasks.

A life-long learning RL agent requires abilities similar to those of biological systems to:

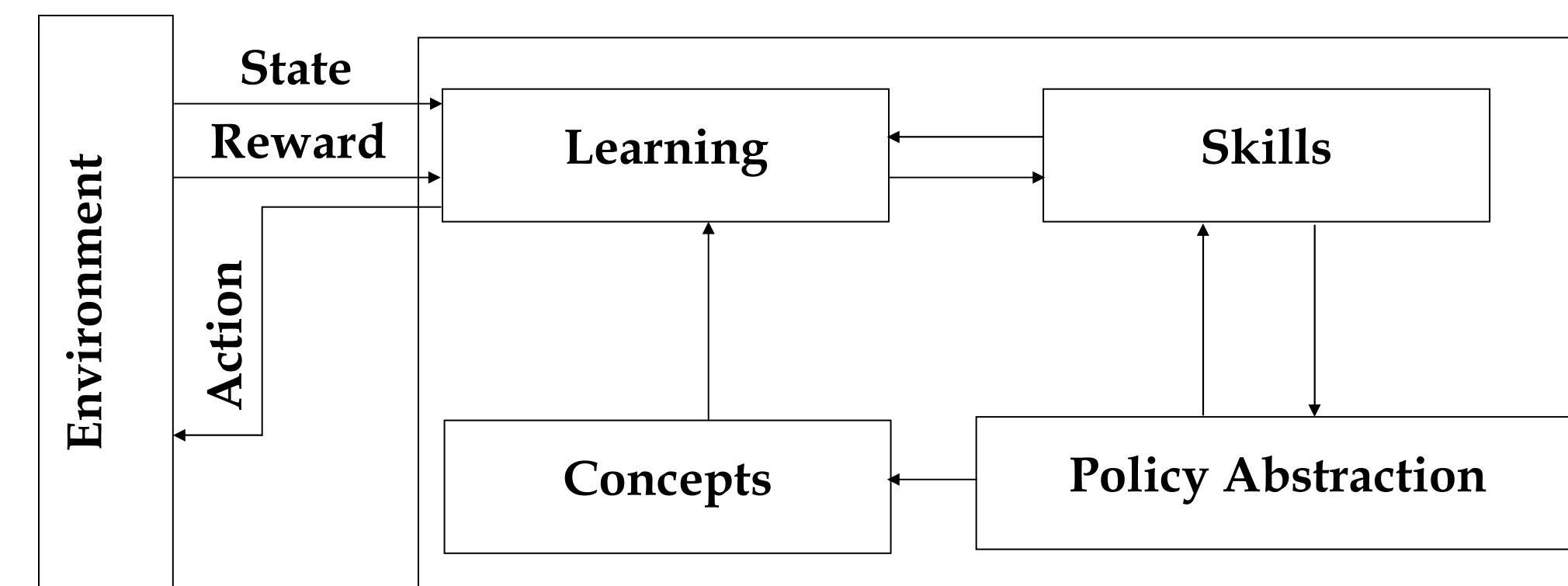
- Learn reusable skill and representational abstractions.
- Identify similar tasks.
- Reuse formed abstractions to learn new tasks.

This poster presents a new approach to control knowledge transfer that:

- Uses a new concept of policy homomorphism to generalize policies.
- Uses partial policy homomorphisms to abstract a general policy from a set of homomorphic policies.
- Uses the extracted general policies to learn new tasks.

The applicability and performance of the general policy extraction and reuse mechanism are demonstrated using examples in a grid world domain

Learning Architecture

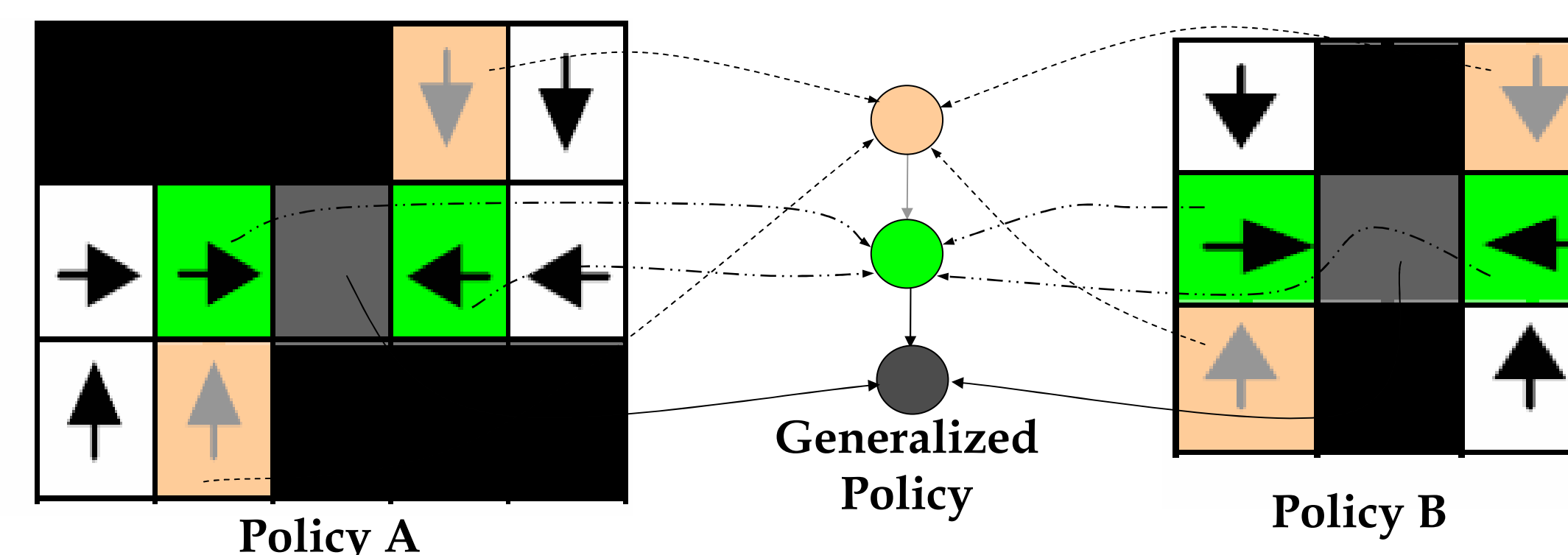


The learning architecture is composed of four main components:

1. **Learning Component:** The learning component of the RL agent's learning framework learns policies for new task instances using SMDP Q-learning. The agent uses this component to learn basic policies for novel, increasingly complex. This is achieved by reusing the control knowledge in the form of generalized policies as higher level actions which the agent can choose to perform.
2. **Policy Abstraction Component:** The policy abstraction component of the learning framework extracts a general policy for a *task type* from a set of previously learned situation-specific, homomorphic policies.
3. **Skills Memory:** The skills component serves mainly as a repository for the learned skills.
4. **Concepts Memory:** The concepts component serves as a repository for the learned concepts.

Policy Homomorphism Framework

The policy abstraction component of the learning architecture uses the concept of policy homomorphism for the autonomous identification of similar tasks and the construction of a general policy for a *task type*.



Definition 1: A Policy Homomorphism $f: \pi \rightarrow \pi'$ is a surjection from a base policy π to an abstract policy π' . f is defined by surjections $(h: S_\pi \rightarrow S_{\pi'}, g_s: A_\pi \rightarrow A_{\pi'})$ and functions $g_s = A_\pi \times s \rightarrow [0,1]$ over the state and action sets for π , $S_\pi \subseteq S$ and $A_\pi = \{a \in A \mid \exists s \in S_\pi: \pi(s,a) > 0\}$ such that the following holds:

- i) For each state-action pair (s, a) : $\pi'(h(s), g_s(a)) = g_s(s, a)$.
- ii) For each state pair (s_i, s_j) : $\sum_{b \in A_{\pi'}} \pi'(h(s_i), b)T'(h(s_i), b, h(s_j)) = \sum_{a \in A_\pi} \pi(s_i, a)T(s_i, a, s_j)$, where T and T' are the transition probabilities in the base and in the abstract policy, respectively.

A complete policy homomorphism requires that every state in a given policy be mapped onto a particular state in the abstract policy. As a result it does not allow abstraction of policies that are only partially homomorphic.

Definition 2: A Partial Policy Homomorphism $f: \pi_p \rightarrow \pi'_p$ is a surjection from a partial base policy π_p to an abstract policy π'_p where f is defined by a tuple of surjections $(h: S_{\pi_p} \rightarrow S_{\pi'_p}, g_s: A_{\pi_p} \rightarrow A_{\pi'_p})$ and functions $g_s = A_{\pi_p} \times s \rightarrow [0,1]$ over the state and action sets of π_p , $S_{\pi_p} \subseteq S_\pi$ and $A_{\pi_p} = \{a \in A \mid \exists s \in S_{\pi_p}: \pi(s, a) > 0\}$ such that:

- i) For all (s, a) : $\pi'_p(h(s), g_s(a)) = \pi(s, a)$.
- ii) For each state pair (s_i, s_j) with $s_i \in S_{\pi_p}, h(s_i) \notin S_{\pi'_p}$: $\sum_{b \in A_{\pi'_p}} \pi'(h(s_i), b)T'(h(s_i), b, h(s_j)) = \sum_{a \in A_{\pi_p}} \pi(s_i, a)T(s_i, a, s_j)$, where T and T' are the transition probabilities for the base policy and the abstract partial policy, respectively.

Both complete and partial policy homomorphisms can be applied to derive abstract policies. However, to ensure that the general policy captures the objective of the underlying policies, this work limits the application to goal based policies by extending the definitions.

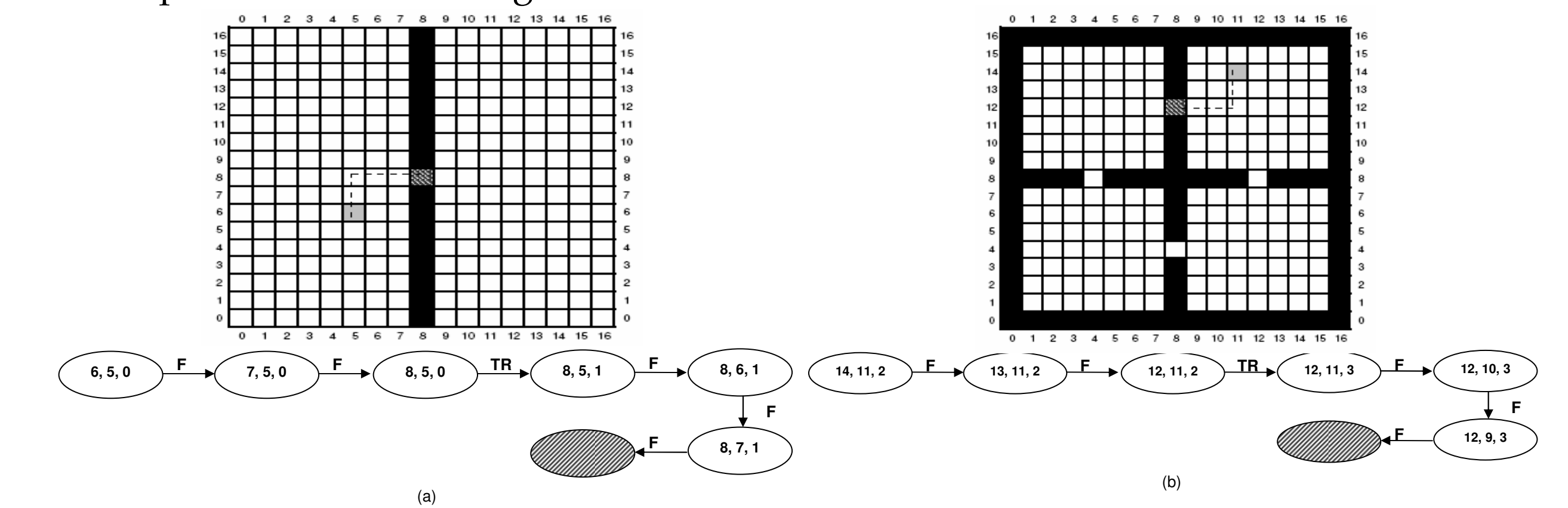
Definition 3: A Goal Based Policy Homomorphism $f: \pi_G \rightarrow \pi'_G$ is a policy homomorphism (complete or partial) that fulfills:

- i) All goal states of the base policy are present as goal states in the abstract policy.
- ii) $S'_{\pi'_G} \cap h(S_{\pi_p} - S_{\pi_G}) = \emptyset$ where $S'_{\pi'_G} = \cup_{s \in S_{\pi'_G}} h(s)$.
- iii) All non-goal states in π'_G are either terminal states or have a non-zero probability to lead to a goal under π'_G .

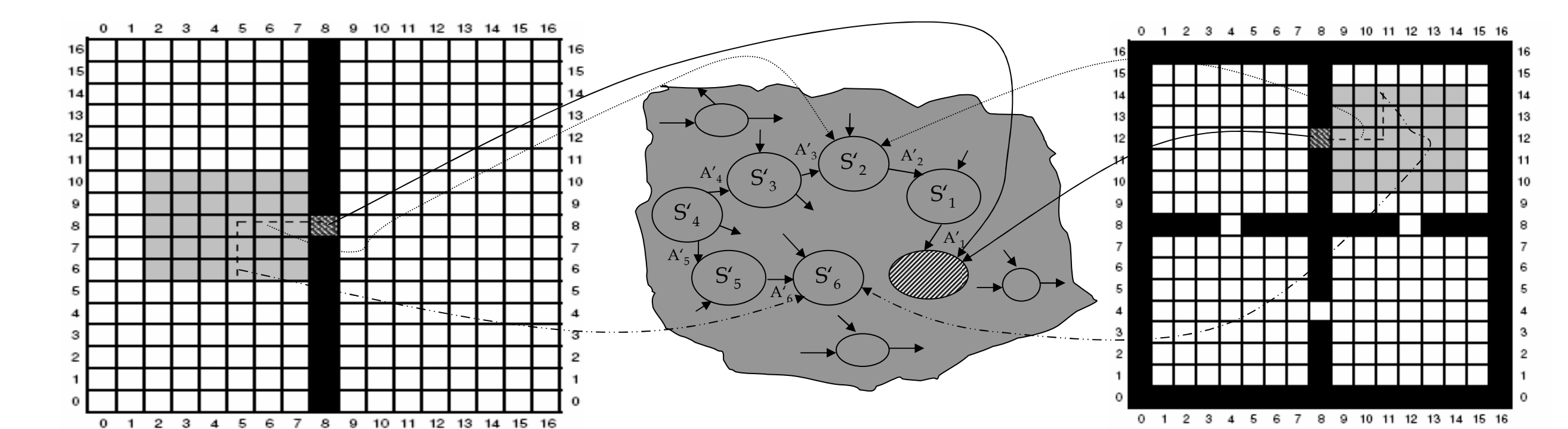
Experiments and Results

The approach is demonstrated using a set of grid world navigation tasks with deterministic forward and turn actions and states represented by the agent's location and orientation and the locations of the four nearest doors. The experiments are divided into three phases. In the first phase of the experiments the agent learns a set of basic policies. In the second phase these policy instances are then used to abstract a general policy. The third phase of the experiments then demonstrates how this abstracted general policy can be reused to address related tasks in novel environments and situations.

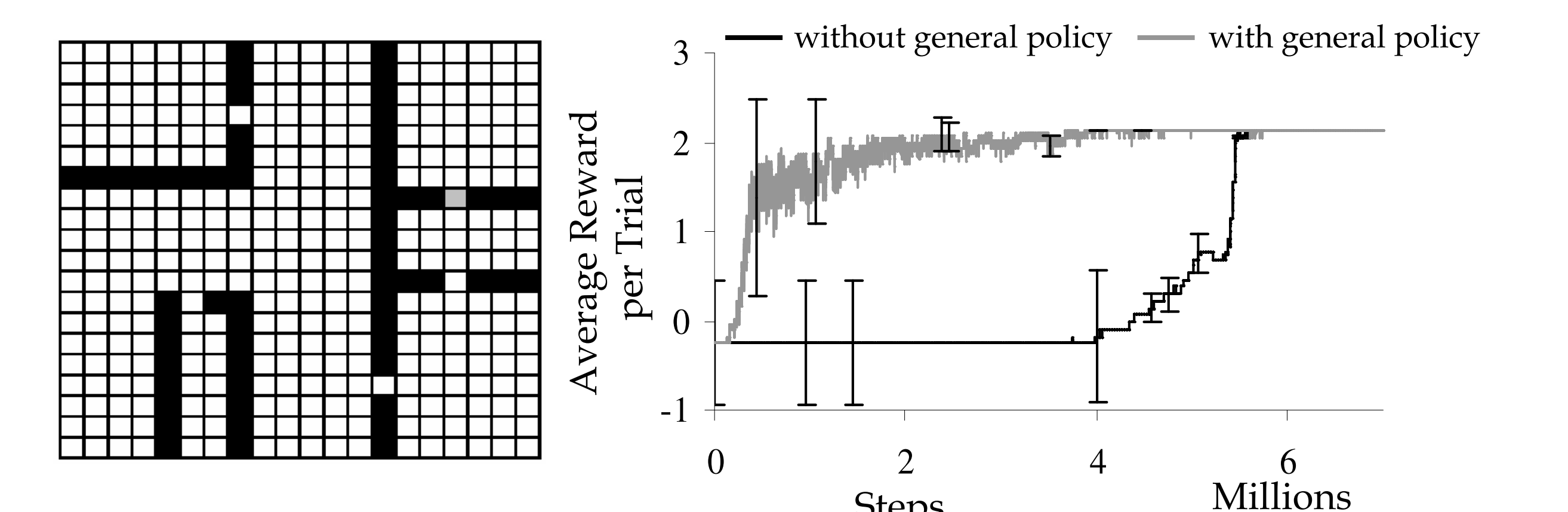
1. **Learning a Set of Basic Policies:** In this phase agent starts out by learning basic policies to reach specific doors in a set grid world environments:



2. **Extraction of Generalized Policy:** In this phase the agent uses the learned policies to extract a general policy using the partial policy homomorphism framework.



3. **Reuse of Generalized Policy:** To demonstrate the reuse of the abstracted policy, a new task in a novel environment is learned with and without the use of the generalized policy.



Each learning curve on the right represents the mean reward per trial averaged over 30 learning runs with confidence intervals indicating one standard deviation. These curves show significantly improved learning times with the generalized policy.

Conclusion and Future Work

- This paper proposes a new approach for skill and concept abstraction using policy homomorphisms.
- The experiments demonstrate that reuse of the general policy and the corresponding mapping functions reduces the learning time for new tasks.
- Extraction of generalized policies allows the agent to compress the state space by abstracting task specific information.
- We are currently extending the policy homomorphism framework to accommodate stochastic domains.

References

1. Asadi, M., Huber, M., 2007. Learning Skill and Representation Hierarchies for Effective Knowledge Transfer. IJCAI-07, pp. 2054-2059.
2. Bakker, B., Schmidhuber, J., 2004. Hierarchical Reinforcement Learning, based on Subgoal Discovery and Subpolicy Specialization. IAS vol (8), pp. 438-445.
3. Barto, A.G., Mahadevan, S., 2003. Recent Advances in Hierarchical Reinforcement Learning. DEES 13(4): 341.
4. Jong, N. K., Stone, P., 2005. State Abstraction Discovery from Irrelevant State Variables. IJCAI-05 pp. 752-757.
5. Lakoff, G., 1987. Women, Fire, and Dangerous Things.
6. Mandler, J. M., 1992. How to build a baby: II. Conceptual primitives. Psychological Review 99(4) pp. 587-604.
7. Ravindran, B., Barto, A., 2003. SMDP Homomorphisms: An Algebraic Approach to Abstraction in Semi-Markov Decision Processes. IJCAI-03, pp. 1011-1016.
8. Wolfe, A. P., Barto, A.G., 2006. Defining Object Types and Options Using MDP Homomorphisms. ICML Workshop on Structural Knowledge Transfer for ML.