From Data-based to Model-based AI: Representation Learning for Planning (RLEAP)* (Long version)

Hector Geffner

29/8/2019

Abstract

Two of the main research threads in AI revolve around the development of **data-based learners** capable of inferring behavior and functions from experience and data, and **model-based solvers** capable of tackling well-defined but intractable models like SAT, classical planning, and Bayesian networks. Learners, and in particular deep learners, have achieved considerable success but result in black boxes that do not have the flexibility, transparency, and generality of their model-based counterparts. Solvers, on the the hand, require models which are hard to build by hand. RLEAP is aimed at achieving an integration of learners and solvers in the context of planning by addressing and solving **the problem of learning first-order planning representations from raw perceptions alone without using any prior symbolic knowledge.** The ability to construct first-order symbolic representations and using them for expressing, communicating, achieving, and recognizing goals is a main component of human intelligence and a fundamental, open research problem in AI. The success of RLEAP requires the development of radically new ideas and methods that will build on those of a number of related areas that include planning, learning, knowledge representation, combinatorial optimization and SAT. The approach to be pursued is based on a clear separation between learning the symbolic representations themselves, that is cast as a **combinatorial optimization problem**, and learning the interpretations of those representations, that is cast as a **supervised learning problem** from targets obtained from the first part. RLEAP will address both problems in the settings of planning and **generalized planning** where plans are general strategies. The project can make a significant difference in how **general, explainable, and trustworthy AI** can be understood and achieved.

1 Introduction

The project RLEAP aims to address and solve a fundamental research problem that is at the heart of the current split between **data-based learners** and **model-based reasoners (solvers)** in AI: the problem of **learning symbolic representations from raw perceptions**. The popularity of data-based learners over model-based solvers is that data is easily available but building models by hand is hard. Yet data-based learners lack the flexibility, transparency, and guarantees that are associated with model-based systems [LUTG17, Pea18, Dar18, Gef18]. By showing how to learn meaningful, symbolic models form raw perceptions alone, RLEAP aims to integrate the benefits of both. The context for learning representations is **planning** where representations play a key role in expressing, communicating, achieving, and recognizing goals [SA77, CL90, RG09, PC09, Gef13, SRBS16].

For an illustration of the **representation learning problem for planning** addressed by RLEAP, consider a range of 2D worlds where an agent must learn to achieve goals from scratch **from raw perceptions alone** (images) and **no prior symbolic knowledge.** The agent has to learn about the world in a flexible manner so that the knowledge gained for achieving goals in some worlds can

*Slightly edited version of Advanced ERC Project proposal, funded 1/10/2020–30/9/2025.

be reused to achieve related goals in other worlds. The goals are to be conveyed to the agent in a formal language whose grammar may be known to agent but whose symbols and meanings are not. The meanings of these symbols have to do with the objects and relations in the world which are not directly available to the agent who sees raw images only.

A version of the problem appears in the recent paper *BabyAI: A platform to study the sample efficiency of grounded language learning* by Yoshua Bengio and co-authors [CBBL$^+$19]. In the paper, the authors describe a simulation platform for a class of 2D worlds featuring a number of objects and an agent that accepts goal instructions expressed in a context-free language. The agent learns to interpret and carry out the given goals from scratch by learning to maximize expected reward using **deep reinforcement learning** [MKS$^+$15, SHM$^+$16, SSS$^+$17, SHS$^+$18]. Among the conclusions, the authors state that "the methods scale and generalize poorly when it comes to learning tasks with a compositional structure." Hundreds of thousands of demonstrations are needed indeed to learn tasks which are trivial by human standards, and some simple tasks are not learned reliably at all.

The problem of representation learning for planning, while simple to describe and central to AI, is largely **unsolved**, and **current ideas and methods prove to be inadequate**. Indeed, two characteristics of deep reinforcement learning that have to do with its successes and its failures are its **ability** to deal with high dimensional perceptual spaces from scratch without prior knowledge, combined with its **inability** to use or produce such knowledge. Humans, on the other hand, excel at using prior knowledge when dealing with new tasks and at producing such knowledge when solving related tasks [LUTG17, Mar18a, Mar18b]. Certainly, the construction of reusable knowledge from experience (**transfer learning**) has been a central concern in reinforcement learning [TS11, Laz12] and in recent work in deep reinforcement learning [GDL$^+$17, BBQ$^+$19], but the semantic and conceptual gap between the **low level techniques** that are used, (neural network architectures and loss functions) and the **high-level representations** that are required (first-order representations involving objects and relations), remains just too large [AF18, TBF$^+$18, FLBPP19, Asa19, GS19].

RLeap will develop the formulations and algorithms for showing how **first-order symbolic representations** involving objects and relations can be learned automatically from data **without using any prior symbolic knowledge**. Unlike current work in deep reinforcement learning, these representations will not be expected to emerge bottom-up from the learning process but will be forced top-down. We know indeed the structure of the first-order representations that are used in planning and the benefits that they have: they can be used to attain a variety of compound goals (**compositionality**), can be reused easily in a variety of problems (**transfer**), and can be queried at a high level of abstraction (**transparency**). There is thus no need to re-discover the structure of these representations nor to learn alternative ones that lack these properties. The challenge is to learn them from data.

2 Motivation

The current excitement about AI is the result of a number of breakthroughs in machine learning. [KSH12, GMH13, HZRS16, SHM$^+$16, SSS$^+$17]. Learners, like solvers, can be understood as programs that compute mappings from inputs x into outputs $f(x)$ by solving well-defined mathematical tasks [Gef14, Gef18]. In **deep learning** (DL) and **deep reinforcement learning** (DRL), training results in a function f that has a fixed structure given by a deep neural network [LBH15, GBC16] and a number of adjustable parameters. In DL, the input vector x may represent an image and the output $f(x)$, a classification label, while in DRL, the input x may represent the state of a game, and $f(x)$, the value of the state. For **solvers**, the input x represents a model instance, and the ouput $f(x)$, the solution to the instance x. Solvers have been developed for a variety of models that include constraint satisfaction problems (CSPs), SAT, answer set programs, Bayesian networks, classical planning, and various forms of probabilistic planning [Dec03, BHvM09, GKKS12, Pea88, GB13, Ber95].

The distinction between data-based learners and model-based solvers is reminiscent of the distinction between **Systems 1** and **2** in current psychologies theories of the human mind (Kahneman's Fast and Slow Thinking): the first referring to the **intuitive mind** that is fast, associative, unconscious,

effortless, and parallel; the second to the **analytical mind** that is slow, deliberative, conscious, effortful, and serial [Kah11, ES13]. From this point of view, the learners deliver System 1 intelligence by producing fast black boxes that correspond to the learned functions f, while solvers exhibit System 2 intelligence by computing the outputs $f(x)$ for each given model instance x by reasoning.

A crucial difference between the human mind and current AI systems, however, is that **Systems 1 and 2 are tightly integrated** in the workings of the mind, while learners and solvers rarely talk to each other. This limitation explains why for example self-driving cars are unlikely to be deployed anytime soon: they are Systems 1 only, and as such, they cannot be trusted in open worlds where unexpected situations are bound to happen. By learning representations that enable reasoning on a case by case basis, RLeap will contribute to make AI systems from data that are able to integrate System 1 and System 2 intelligence.

3 Objectives (Summary)

The basic goal of the project is to address and solve the problem of **learning first-order symbolic representations from raw perceptions alone** in the context of planning where representations play a key role and their structure is known and provides the basic properties of **compositionality**, **reuse**, and **transparency**. This basic goal corresponds to Objective 1 below; Objectives 2 to 4 address closely related goals.

Objective 1: Learning representations for planning. A planning problem $P = \langle D, I\rangle$ combines a **first-order domain** D that contains action schemas, predicate symbols, and first-order atoms defining preconditions and effects, and **instance information** $I = \langle O, Init, Goal\rangle$ that encodes the relevant objects O and the sets of ground atoms that express the initial and goal conditions [McD00, HLMM19]. A planning problem P defines a directed graph $G(P)$ whose nodes n represent the states $s = s(n)$ over P, and whose edges (n, n') represent the state transitions. The states are represented by sets of ground atoms; namely, those that are true in the state.

For defininig the basic representation learning problem, let an **image graph** G be a directed graph where each node n is associated with an observation that we take to be a **raw image** $v(n)$. The image graph can be obtained by sampling a large number of observed trajectories. We will assume that the graph is complete initially; namely, that all the possible observed trajectories are in the graph. The assumption is feasible if the problems used for training are small, i.e., if they involve hidden state spaces that are not too large, an idea that is compatible with *curriculum learning* [BLCW09].

The basic **representation learning problem** is to infer the **symbolic, first-order representation** of a set of planning instances $P_i = \langle D, I_i\rangle$, $i = 1, \ldots, n$ with a common domain D from **input data** given by a set of **image graphs** $G_1, \ldots, G_n$. As an example, consider 2D worlds that represent rectangular grids where an agent can move one unit at a time, collect keys one a time, and drop them. The observed trajectories can be generated with a simulator and a graphics engine. The learning problem is to infer the first-order representations $P_i = \langle D, I_i\rangle$ that account for the observations. This means **discovering** predicate symbols like loc^1, key^1, $adjacent^2$, $hold^1$, $handfree^0$, action schemas like $move^2$, $pickup^2$, $drop^2$, and so on, or equivalent representations, **from raw perceptions alone.**

A key assumption is that different states give rise to different images. Partial observability and non-deterministic actions will also be considered on top of this basic setting, taking advantage that the languages used for planning in the richer settings are built on top of this basic language [YLWA05, CCO$^+$12]. Noisy observations will be addressed too. The basic problem, however, is central and challenging enough, and far from being solved.

Objective 2: Learning representations for generalized planning. Generalized planning studies the methods for expressing and obtaining plans that will solve not just one planning instance but multiple instances. For example, a **general plan** for solving **any instance** of the domain where a key is to be picked up and delivered to a target location is simple: the agent has to go to the key, pick it up, go to the target location, and drop it. The challenge is to obtain such general plans

automatically [SIZ08a, BPG09, SIZ11, HD11, BL16, SAJJ16, IM19]. In recent years, it has been shown by the PI and others that such general plans can be derived through reductions and off-the-shelf planners from a **generalized model** that provides a common abstraction of the instances to be solved [BG18b, BFG19]. Objective 2 is **learning such generalized models directed from perceptual data.** The objective ties closely with knowledge representation on the one hand, and learning on the other. Indeed, learning approaches are not aimed at finding plans for single instances but general plans [CBBL$^+$19, FLBPP19, GS19].

Objective 3: Learning hierarchical representations. In order to compute plans it is often necessary or convenient to plan at different levels of abstractions, with the constraints among the different levels playing a crucial role. Indeed, a high-level plan is useless if it can't be brought down to the lowest level for execution [BY91]. In planning and reinforcement learning, hierarchies and high level actions or options have been defined mostly **by hand** [HTD90, NAI$^+$03, GA15, BAH19, SPS99]. In spite of many efforts [KB07, MRW07, BHP17, MBB17], **the key problem** that remains open and will be addressed in RLeap is how to discover crisp, symbolic hierarchical representations automatically, constructing successive layers of abstractions from the bottom up. The new elements that will be exploited are the intimate connection between hierarchical planning and generalized planning, as **hierarchical plans are hierarchical general plans**, and recent work by the PI and team that shows how abstractions for generalized planning can be obtained automatically from a first-order symbolic representation of the domain [BG18b, BFG19].

Objective 4: Theory of representations for planning and learning. There are two implicit assumptions in the project: one that the target representations to be learned are simple (have a low dimensionality), the other, that planning with these representations is simple as well (low polynomial time). These assumptions hold well in planning where 1) domains involve a bounded number of action schemas and predicate symbols, 2) planners scale up well [BG01, HN01, RW10, LG17a], in spite of the worst-case complexity results [Byl94]. Some of the new algorithms are indeed exponential in a **width** parameter that is small and bounded for most planning domains when goals are single atoms [LG12, LG17b, LG17a, FRLG17]. The formal proofs that establish that a domain has bounded width actually uncovers domain **features** (numerical state functions) that may shed light on two apparently unrelated open problems in **generalized planning** and **reinforcement learning**: what are the features required for producing general plans in a given domain, and what are the features that make linear function approximations work in a given domain [SPLC16a, BDD$^+$19, BGP19]. Objective 4 is about developing a theory that formally relates the **features** that appear in three contexts, which may have much in common: **width analysis**, **linear function approximation in RL**, and **generalized planning**.

4 Feasibility and Novelty (Summary)

The **feasibility** and **novelty** of RLeap rest on two main premises and the way in which we will formulate them mathematically and computationally. First, that the language for extracting, using, reusing, and composing knowledge is the language of first-order symbolic representations, and that it is not necessary nor convenient to learn the structure of such languages from scratch. Second, that it is precisely the structure of such languages that provides the **strong structural priors** that make the learning of crisp representations feasible and data efficient.

RLeap will not follow deep learning approaches in assuming that the target representations emerge from the learning process through the use of suitable neural architectures (e.g., attention mechanisms) and loss functions (e.g., that penalize entanglement). Instead, RLeap will make first-order representations the explicit target of the learning process, and by doing so, it will decompose the representation learning problem in two: a **representation discovery problem**, that is a purely combinatorial problem, and a **semantic interpretation problem**, that is a supervised learning problem with targets obtained from the first part. Going back to the example above: discovering the action schemas

with the predicate symbols loc^1, key^1, $adjacent^2$, $hold^1$, $handfree^0$ or equivalent ones from the input graphs is the representation discovery problem. Learning the functions that provide the denotation of such logical symbols in the images is the semantic interpretation problem.

The **representation discovery problem** is **combinatorial** because the number of possible domains given a bound on the number of action schemas, predicate symbols, and their arities, is bounded. The values of these parameters are bounded and small, and do not grow with the size of the instances. The problem of learning the simplest planning instances $P_i = \langle D, I_i \rangle$ that account for a number of input image graphs G_i, $i = 1, \ldots, m$ can then be cast and solved as a **combinatorial optimization** problem. An instance P_i accounts for the image graph G_i if the graph $G(P_i)$ associated with P_i is structurally equivalent (isomorphic) to the plain graph G_i; i.e., if the graph G_i, leaving the images aside, is **generated** by the planning instance P_i.

A **key property** of this view is that the first-order symbolic representations that are learned from the input graphs $G_1, \ldots, G_m$, i.e., the action schemas and the predicate symbols, do not depend on the **raw images** $v(n)$ associated with the nodes n but on the structure of the graphs. This means that the way in which objects are displayed on the images may change but the resulting representation will not. This is very different from works where the representations are obtained from auto-encoders and hence are low dimensional representations of the images [KW14, AF18, Asa19]. In the proposed formulation, **the symbolic representations do not provide a compact encoding of the images but of the structure of the state space.**

The images play a key role in the **semantic interpretation problem** that must yield the **functions** that provide the denotation of the learned predicate symbols in the images; namely, the tuples of objects that satisfy the predicate in the image. The problem is similar to the **semantic scene interpretation** problem [NM08, KZG$^+$17, DSG17] where a raw image must be mapped into a logical formula that describes the contents of the image, in our case, the set (conjunction) of ground atoms $p(c_1, \ldots, c_k)$ that are true in the image for each of the learned predicate symbols p. In our setting, however, the semantic interpretation problem does **not** require labeled examples (**supervision**) as the solution of the representation discovery problem matches the nodes n in the graphs G_i with states $s(n)$ in the graphs $G(P_i)$, so that every image $v(n)$ is associated with a set of atoms $s(n)$.

The semantic interpretation problem is not trivial but it is much simpler than the full representation learning problem itself, and can be addressed through a principled combination of **fast convolutional object detectors** [RDGF16, RF17], **relation classifiers** [ZK15, SRB$^+$17, SYZ$^+$18] and **combinatorial optimizers** [BJM$^+$18, FM06, ABL13, MML14], that yield the most likely scene interpretation in terms of the learned predicate symbols, using extra logical constraints and invariants that can be obtained from the learned planning instances P_i [Hel09, Rin17].

These ideas pertain mainly to Objectives 1 and 2. Objective 3 is about constructing new symbolic representations from given (learned) symbolic representations, and so is the problem of aligning these symbolic representations with given goal instructions. Objective 4 is about elaborating a theory of tractable representations for plannning and learning. The work in these two objectives will benefit from the expertise of PI and his team that introduced the notion of sound abstractions for generalized planning [BDGGR17, BG18b, BFG19], and the notion of width [LG12, LG17a, LG17b, FRLG17].

RLeap aims at methods that are domain-independent and that can be evaluated empirically in the way that planners and SAT solvers are tested in competitions. The methods to be developed are to be applicable to existing simulation platforms like ALE, for the Atari games [BNVB13], and GVG-AI, for general video-games [PLST$^+$15], but for getting there, we will move one step at a time. We are not just interested in performance and coverage, but also in understanding. There are indeed model-based RL approaches for some of these domains based on the prediction of screens and rewards [OGL$^+$15, KBM$^+$19], but none that builds meaningful first-order symbolic representations from the screen that are used to plan.

5 State of the art. Related Research

AI systems that are general, explainable, and trustworthy require System 1 and System 2 intelligences tightly integrated, yet practically no current system achieves an integration of this sort. The **AlphaZero** program [SSS+17, SHS+18], that achieved suprahuman performance in Go and Chess, integrates a deep reinforcement learner [MKS+15] and a Monte Carlo Tree Search (MCTS) planner [KS06, CWH+08], but the learning is about the level of play, not about the model that is fixed and known in advance. More general AI systems will have to **learn models from data**, and moreover, the learned models must be structured in terms of objects and relations to facilitate reuse, transparency, and compositionality. RLeap is aimed at this challenge in the context of planning where **first-order representations** are known to provide these benefits [McD00, HLMM19]. These representations for planning, however, are written by hand. The main challenge addressed by RLeap is to **learn them from raw perceptions alone.**

Model learning is the goal of **model-based reinforcement learning** where the model parameters of a Markov Decision Processes (state transition probabilities and rewards) are learned by active exploration [SB98, BT03]. In the standard setting, the states are given and assumed to be fully observable but what is learned in one model does not transfer easily to other models. In particular, no high-level symbolic representations are learned. An exception is the work on **object-oriented MDPs** that starts and refines a **first-order symbolic representation** made up of objects, types, and relations [DCL08, HMT15]. Similar work has been done in classical planning [YWJ07, AJOR19]. In the two cases, however, the first-order symbolic models are obtained using predicate symbols with a known meaning that are given. **Inductive logic programming** approaches have this form as well obtaining symbolic representations for new concepts in the form of logic programs from given symbolic predicates and a number of positive and negative samples [MDR94, DRK08]. Variations of these ideas have been used in **hybrid learning schemes** that integrate symbolic and deep learning [MDK+18, KP18, SG16, XZF+18, GGL+19, EG18] and in **relational reinforcement learning** for obtaining general policies [Kha99, MG04, FYG04, DDRD01, KODR04]. A **general policy** is a policy that can solve problems that involve different sets of objects, configurations, and state spaces. More recently, **deep learning** methods have been used to obtain such policies, but once again, starting with the first-order (PDDL) symbolic representations of the domains [TTTX18, BdBMS19, IFT18, BG+18a]. A model-based approach for obtaining general policies from the same representations is developed in [BG18b, BFG19]. A related formulation finds abstract symbolic representations from low level symbolic representations and a given set of high-level options [KKLP18].

Deep reinforcement learning (DRL) methods have emerged as the main approach capable of generating general policies over high-dimensional perceptual spaces **without using any prior symbolic knowledge** [MKS+15, SHM+16, SSS+17]. Yet by not using or constructing first-order symbolic representations, DRL methods have not managed to obtain the benefits of transparency, reuse, and compositionality [CBBL+19, Mar18a, LB17]. Recent work in deep symbolic relational reinforcement learning [GS19] attempts to account for objects and relations through the use of attention mechanisms and suitable loss functions, but the **semantic and conceptual gap** between these **low level techniques** and the **high-level representations** that are required remains just too large. Something similar occurs with work aimed at learning low dimensional representations that disentangle the factors of variations in the data [TBF+18, FLBPP19]. The first-order representation used in planning have indeed a low dimensionality given by a small number of action schemas, predicate symbols, and atoms, but it is not clear how such representations can emerge bottom up from current architectures.

A recent deep learning approach infers compact representations for planning from raw images alone using a class of **variational autoencoders** where the representations provide a low dimensional encoding of the images [AF18]. Follow up work has extended this approach for producing first-order planning representations as well [Asa19]. It is far from clear, however, that the high level representations that are required are compact encodings of images. The standard PDDL planning files do not actually codify images but some of the structural relations that appear in the images. Moreover, it may be difficult to get crisp, compact symbolic representations when representations are

forced to encode the images themselves.

RLeap departs from existing approaches to representation learning by assuming that the target representations encode the **structure of the state space**, not the way in which the states are visualized. In other words, if the images used to display the states are changed, while keeping the condition that different hidden states are displayed differently, the perceived state space and the first-order representations that are obtained from it, will not change. What will change in that case is the interpretation of the symbols learned in the images. We draw for this on the distinction between **symbols** and their **denotation** in first-order logic [Men09, HR04]. A first-order semantic interpretation provides a denotation to every (non-logical symbol) so that every (closed) first-order term and formula can be evaluated. In particular, terms t denote objects from the interpretation domain, and predicate symbols p denote boolean functions. The denotation of an atom like $p(t_1, \ldots, t_k)$ is *true* if the denotation of p maps the tuple of objects denoted by the terms t_1 to t_k into *true*. The *extension* of the predicate symbol p in the interpretation is defined as the set of object tuples that are mapped to *true*, and it is an alternative representation of the denotation function of p.

6 Objective 1: Learning representations for planning

6.1 Planning

Planners are solvers for models that involve goal-directed behavior [RN09, GB13, GNT16]. Planners and planning models come in many different forms depending on a number of dimensions including: 1) uncertainty about the initial situation and action transitions, 2) type of sensing, 3) representation of uncertainty, and 4) type of objectives. These four aspects affect the complexity of planning and the form of plans. All planners that scale up make heavy use of the compact representation of planning problems. In classical planning, one of the main techniques is to search for paths connecting the initial state to a goal state in the implicit graph $G(P)$ defined by a problem P whose size is exponential in the number of variables in P. For guiding the search, a heuristic $h(s)$ that estimates the cost to go from a state s to the goal is derived in low polynomial time from a simplification (relaxation) of the problem [McD99, BG01] Current planners make use of this idea and a number of extensions [HPS04, RW10] that also rely heavily on the compact representation of planning problems.

6.2 Target representations for planning

A (classical) planning problem P is given in terms of a **first-order domain** D and **instance information** I as a pair $P = \langle D, I \rangle$. A planning domain D contains a set of action schemas, a set of predicate symbols, and first-order atoms describing the preconditions and effects of the action schemas. The instance information is a tuple $I = \langle O, Init, Goal \rangle$ where O is a (finite) set of object names c_i, and $Init$ and $Goal$ are sets of ground atoms $p(c_1, \ldots, c_k)$ where p is a predicate symbol in D of arity k. This is the structure of planning problems expressed in the PDDL standard [McD00, HLMM19] that corresponds to STRIPS schemas [FN71]. Most current planners take a planning problem P and for efficiency reasons, ground it first by replacing the action schemas by their possible instantiations over the constants in O. The first-order representation of the domain D is necessary for defining new instances in terms of new tuples $I = \langle O, Init, Goal \rangle$. The name of the constants in O is irrelevant and it is possible to replace them by numbers in the interval $[1, N]$ where $N = |O|$ stands for the number of objects in O.

A problem $P = \langle D, I \rangle$ defines a directed graph $G(P) = \langle V, E \rangle$ where the nodes are associated with states s over P, i.e., sets of ground atoms $p(c_1, \ldots, c_k)$, with a root node representing the initial state $Init$, and edges (s, s') representing the state transitions enabled by the ground actions in P. A plan for P corresponds to a sequence of ground actions from P that connects the root node in $G(P)$ with a node that represents a goal state; i.e., a set of atoms that includes those in $Goal$. Since the size of this graph is exponential in the number of ground atoms in the problem, planning algorithms that search for such paths ignoring the structure of the planning problem P do not scale up.

6.3 Learning the target representations for planning

The basic **representation learning for planning** problem addressed by RLeap can be described as follows:

> **Given** data in the form of a set of **image graphs** $G_1, \ldots, G_m$ with a raw image $v(n)$ (observation) associated with every node n. **Find** 1) the first-order representation of the planning problems $P_i = \langle D, I_i \rangle$ that best account for the given graphs G_i, $i = 1, \ldots, m$, and 2) the denotation functions of the learned predicate symbols over the images.

The problems 1) and 2) are called the **representation discovery** and **semantic interpretation** problems respectively. In addition, a third problem, **goal interpretation**, is about learning to interpret the external goal instruction in terms of the symbolic language learned. Recall that an **image graph** is a plain directed graph where every node n has an associated raw image $v(n)$.

Representation discovery

It is assumed that the representations to be learned encode the structure of the state space and not the way in which the states are displayed. As a result, the symbolic representations are obtained from the **structure** of the image graphs and not from the images associated with the nodes. The basic **representation discovery problem** is thus:

> **Given** a set $G_1, \ldots, G_m$ of plain graphs, corresponding to the image graphs with the observations $v(n)$ excluded. **Find** the first-order domain D and instances $P_i = \langle D, I_i \rangle$ such that the pairs of graphs $G(P_i)$ and G_i are structurally equivalent (isomorphic) for $i = 1, \ldots, m$.

In other words, **the planning instances $P_i = \langle D, I_i \rangle$ account for the observed graphs G_i when these graphs could have been generated by these instances**. We express that the graphs $G(P_i)$ and G_i are isomorphic by writting $G(P_i) = G_i$, and write $s(n)$ to denote the state over the instance P_i that corresponds to the node n in the input graph G_i.

As an example, the graphs may display the transitions in a 2D world where an agent moves one unit at a time in a rectangular grid, collecting keys, one a time, and dropping them. One input graph G_i may be for a 3x4 grid and 2 keys, and another one, for a 5x6 grid and 3 keys. The **representation discovery problem** would have to produce the general domain D and the instances $P_i = \langle D, I_i \rangle$ whose graphs $G(P_i)$ must match the observed graphs G_i. This means coming up with a **first-order representation** involving **predicate symbols** like loc^1, key^1, $adjacent^2$, $hold^1$, $handfree^0$ and **action schemas** like $move^2$, $pickup^2$, $drop^2$, or equivalent ones, where the numerical superindex is used to denote the arity of predicates and action schemas. In addition, the instance information $I_i = \langle O, Init, Goal \rangle$ has to be discovered as well, providing the number of objects $|O|$ and initial situation $Init$ of the planning instance P_i. The $Goal$ plays no role in this problem.

This is a crisp formulation of the first problem of learning the representations for planning from purely non-symbolic inputs (the input plain graphs G_i), that we expect to lead to crisp and meaningful first-order symbolic representations. This is because the representations have to do with the structure of the observed state space and not with the content of the images.

Representation discovery for planning is a **combinatorial** problem because the space of possible domain representations that needs to be considered is bounded by a small number of parameters that have bounded and small values. These parameters are the number of action schemas, predicate symbols, and their arities. These parameters define how complex a domain representation is which is relevant for finding the **simplest** domain D and instances $P_i = \langle D, I_i \rangle$ that account for the observed graphs. While in principle, one could enumerate all possible candidate domains D and test them with the possible instances I_i (whose size is bounded by the size of the graph G_i and the domain parameters), this is not feasible in practice, and a much more efficient approach is to cast and solve the

representation discovery problem as a **combinatorial optimization problem**, using SAT, Weighted Max-SAT, or IP (Integer Programming) solvers. There are indeed excellent and scalable solvers that can be used for this purpose off-the-shelf [ABL13, MML14, BJM$^+$18].

Assumptions and Properties

This approach for obtaining symbolic representations from raw perceptions is very different from existing approaches, and it is based on a crisp distinction between learning the symbolic representations themselves, and learning how to interpret them over the images. Let us elaborate on the implicit assumptions and some key properties.

First, it is assumed that the learner inputs are not observed traces or executions but complete graphs. This distinction, however, is not critical, as we will consider graphs G_i associated with small instances that involve small hidden state spaces. Then a sufficiently large number of sampled executions in each instance yields the corresponding graph. Following Pearl's account of causality [PM18], the graphs define the space of possible causal interventions that allow us to recover the **causal structure** of the domain in a first-order language.

Second, it is assumed that the hidden states are **fully observable**, meaning that two observations (images) are different if and only if they are observations of different states. This means also that observations are **noise free**. This is a natural place to start but this assumption can be relaxed in the combinatorial optimization formulation in a standard way by penalizing departures from this ideal noise free condition. In a slightly different form, **partial state observability** and **non-deterministic** actions are to be handled as well, but leaving the basic setting intact. Notice indeed that the compact, first-order languages used for representing probabilistic planning problems, like Probabilistic PDDL [YLWA05], are small variations of the languages used for representing classical problems, and hence, would involve a small change in the structure of the target languages to be learned.

A key property of this view of representation learning that is different from all other approaches is that the resulting symbolic representations depend on the structure of the input graphs but not in the way in which nodes are mapped to images. In other words, **the resulting symbolic representations are independent of the way in which objects and relations are displayed in the images**. The learned representations encode in a compact manner the structure of the observed state spaces, not the state observations (images) themselves.

Verification

The first-order representations $P_i = \langle D, I_i \rangle$ are learned from a set of graphs G_i, $i = 1, \ldots, n$. It is possible to verify the representations learned for the domain D by leaving some graphs G_k for **testing only**, as it is standard in supervised learning. For this, the learned domain D is verified with respect to each testing graph G_k individually, by checking whether there is an instance $P_k = \langle D, I_k \rangle$ such that the graphs $G(P_k)$ and G_k are structurally equivalent. This is a simpler version of the problem above where the domain representation D is fixed.

Semantic Interpretation Problem

The discovered representations are sufficient for computing plans for **new instances** $P = \langle D, I \rangle$ provided that the instance information $I = \langle N, Init, Goal \rangle$ is encoded **by hand** in terms of the predicate symbols learned. This is indeed the way that planning problems are modeled and solved. However, in the learning setting, we want the agent to accept new instances in a different way, in particular, by providing the initial state of the pronlem by means of an **image**. The **semantic interpretation problem** is the problem of learning to map raw images into first-order states (sets of ground atoms):

> **Given** 1) a set $G_1, \ldots, G_n$ of **image graphs** where every node n has an associated image $v(n)$, and 2) the planning instances $P_i = \langle D, I_i \rangle$, $i = 1, \ldots, n$ **discovered** from these

graphs. **Derive** an interpretation function that maps **images** w into pairs $\langle N(w), s(s)\rangle$ where $n(w)$ is the **number** of objects in the image w, and $s(w)$ is the **state** encoded in the image, given by the set of ground atoms $p(c_1, \ldots, c_k)$ that are true in w, where p is a predicate symbol from D of arity k and c_i is an object id, $1 \leq c_i \leq N(w)$.

The semantic interpretation problem is a simplified version of the **scene interpretation** problem where a raw image must be mapped into a logical formula that describes the contents of the image [NM08, KZG$^+$17, DSG17]. The differences are mainly two. First, the output formula is restricted by the predicate symbols in the domain: we just need to find out whether the atoms $p(c_1, \ldots, c_k)$ that can be built from the predicate symbols in the domain and the objects in the image are true or not. Second, and most importantly, the semantic interpretation problem does **not** require labeled examples (**supervision**) as the solution of the representation discovery problem provides the necessary **targets** over the nodes in the input graphs. Indeed, if $w = v(n)$ is the image associated with node n of the input image graph G_i, the state s associated with w is $s(n)$; namely, the state in the graph $G(P_i)$ that matches the node n from G_i.

The semantic interpretation problem is not trivial but it is much simpler than the full representing learning problem and can be addressed through a principled combination of **fast convolutional object detectors** that find the relevant objects in the image [RDGF16, RF17], **relation classifiers** that map each of the possible atom $p(c_1, \ldots, c_k)$ into probabilities [ZK15, SRB$^+$17, SYZ$^+$18], and **combinatorial optimizers** [ABL13, MML14, BJM$^+$18] that assemble these local probabilities to compute most likely global interpretation, using also extra logical constraints and invariants that can be derived from the learned planning instances, like that the number of objects do not change from state to state, or that certain atoms are mutually exclusive [Hel09, Rin17].

Several subtleties need to be mentioned concerning the targets for the semantic interpretation that result from the solution of the representation discovery problem. First, the object ids obtained from the object detectors need to be aligned with the object ids in the planning instances P_i. Second, the information about the atoms $p(c_1, \ldots, c_k)$ that have been inferred to be true in an image needs to be massaged so that it can be used to train the object detectors and relation classifiers. Third, **dynamic aspects** as captured by the state transitions over the learned instances P_i may need to be used to track the object ids in the images across time, in particular when actions affect the state of multiple objects. This suggests that alternative learning approaches to the semantic interpretation problem may need to be considered like recurrent neural networks [GMH13, Sch15, GBC16] or graph embeddings [HYL17, ZYZZ18]. Yet there is no representation to be learned but just 1) the relation between the constants c_i in the instances P_i and the objects detected, and 2) the truth of the atoms $p(c_1, \ldots, c_k)$. Both are to be learned automatically and without supervision using the rich information that is available in the state graphs $G(P_i)$ learned from the plain graphs G_i.

Goal Interpretation

The **goal interpretation problem** is about mapping external goal instructions given in a formal language with a known syntax to the learned language of the agent. It is a symbolic learning problem which does not involve raw perceptions. In its most simple form, when instructions are conjunctions of ground domain atoms, the problem reduces to finding the correspondence between the learned domain symbols and those used in the instructions. More generally, the unknown symbols of the goal instructions will map into concepts and relations that are derivable from the symbols in the agent language. The problem is **semi-supervised** as the agent needs feedback in order to know when the goal instructions have been accomplished. Versions of this problem have been considered in the literature of **grounded language learning** [YS13, HHG$^+$17, CBBL$^+$19], but once again our problem is simpler as the agent language has been learned and the goal instructions will be assumed to be given in a formal language with a known syntax.

Summary. Objective 1 is about learning first-order symbolic representations for planning from traces of images organized into **image graphs**. **Representation discovery** is cast as a combinatorial optimization problem where the target representations must account for the structure of the graphs. **Semantic interpretation** is about learning to map images into states (sets of ground atoms). **Goal interpretation** is about aligning the learned symbolic language with the formal language of the goal instructions. The result would be the first model-based planning framework where crisp, reusable, first-order symbolic models are learned from raw perceptions alone.

7 Objective 2: Learning representations for generalized planning.

7.1 Generalized planning

Generalized planning studies methods for expressing and obtaining plans that solve multiple instances, sometimes all instances of a given domain [SIZ08b, BPG09, HD11, IM19]. For instance, a **general plan** for solving **any instance** of the domain where a key is to be picked up and delivered to a target location is simple: the agent has to go to the key, pick it up, go to the target location, and drop it. This is a general strategy that works no matter the initial or target locations, or the grid size. The challenge is obtaining such generalized plans automatically. The problem ties closely with both **knowledge representation** on the one hand, and **learning** on the other. Indeed, the general plans require an **abstract language** that often is not the language of the domain itself. Likewise, learning approaches are not aimed at finding plans for single instances but general plans that solve multiple instances [SSS$^+$15, TBF$^+$18, FLBPP19]. Generalized planning aims at the same type of general plans, and by making use of the same types of inputs; namely, **raw perceptions**, RLeap will contrast **model-free** and the **model-based** approaches to the computation of general plans where models are not given but are **learned**.

7.2 Target representations for generalized planning

A general plan that covers many instances is derived from a generalized model that provides a common abstraction for the instances. The language of such abstractions is the language of **qualitative numerical planning** [SZIG11, SZG$^+$15, BDGGR17, BG19]. Qualitative numerical planning problems (QNPs) are **propositional** classical planning problems extended with non-negative **numerical variables** X that can be increased ($X\uparrow$) and decreased ($X\downarrow$) "qualitatively", i.e., by random positive amounts that cannot make the variables negative. A basic qualitative property of integer and real variables is enforced: a variable X that is increased finitely often and decreased infinitely often, eventually must have value 0 (assuming that changes do not become infinitesimal). Aside from effects containing increments and decrements, numerical variables X can appear in the initial situation, preconditions, and goals as literals of the form $X = 0$ or $X > 0$. Unlike **numerical planning** [Hel02], **qualitative numerical planning** is **decidable**, and furthermore, QNPs can be reduced (polynomially) to standard, boolean **FOND planning** (fully observable non-deterministic planning) [CRT98, CPRT03, GB13] for which good, scalable solvers exist [MMB12, BM09, GG18].

QNPs provide a convenient abstract language for generalized planning. As an example, the generalized problem of clearing a block x can be expressed in terms of two abstract actions a and b:

$$a = \langle \neg H, n(x) > 0 \mapsto H, n(x)\downarrow \rangle \;\; ; \;\; b = \langle H \mapsto \neg H \rangle \; .$$

where $\langle C \mapsto E \rangle$ denotes an action with preconditions C and effects E. Here H is a boolean variable that encodes if a block is being held and $n(x)$ is a numerical variable that encodes the number of blocks above x. The action a abstracts the action of picking up a block that is above x, making H true and decrementing $n(x)$, while b abstracts the actions that put those blocks aside, making H false and without affecting $n(x)$. A **general policy** for achieving the goal $clear(x)$ is given by the rules:

$$\text{if } n(x) > 0 \text{ and } H, \textbf{ do } b, \quad , \quad \text{if } n(x) > 0 \text{ and } \neg H, \textbf{ do } a$$

The abstract actions a and b are provably **sound** in the sense that they can always be instantiated to one or more ground actions of the domain that have the same effect on the variables $n(x)$ and H as the actions a and b themselves [BG18b, BFG19].

7.3 Learning the target representations for generalized planning

Objective 2 is about learning representations for generalized planning in terms of QNPs from raw perceptions alone. The same problem, but with **given first-order domain representations**, has been recently formulated and solved by the PI team [BFG19]. The achievement of Objective 1, where these domain representations are learned from data, provides thus a way to achieve Objective 2. For this, the domain is learned from the perceptual data and it is then fed to the pipeline described in [BFG19]. This combination is natural and will be implemented and tested, , but in RLeap, we want to pursue a second, alternative approach as well: we want to learn the abstract QNP representations **directly** from the perceptual data. From a theoretical point of view, this involves a different formulation and a different target language to learn, that will shed new insights on representation learning. From a practical point of view, there are two benefits. First, learning the abstraction from data is simpler than learning the first-order domains because the abstract language of QNPs does not feature action schemas and predicates but boolean and numerical variables. Second, the boolean and numerical variables to be learned do not have to be definable in terms of the domain symbols and a given general grammar [BFG19]. From the point of view of expressivity, the abstractions provided by QNPs are appealing because they are suitable for representing both discrete and continuous change.

Representation discovery for generalized planning

A first approximation of the **representation learning problem for generalized planning** can be expressed as follows.

> **Given** 1) a set of directed graphs $G_1, \ldots, G_m$, 2) a raw image $v(n)$ (observation) for every node n, and 3) goal labels for some nodes. **Derive** the simplest abstraction (QNP) that accounts for structure of the graphs G_i and the distinction between goal and non-goal nodes.

In some cases, we will consider variations of this basic setting where edges in the input graphs G_i are labeled with action types that we do not want to abstract in the same class (simple labels such as *move, pick-up, and drop*). We call the graphs G_i with some nodes labeled as goals, **goal graphs**. The goal labels are crucial because **the resulting QNP provide a common abstraction of the graphs G_i for achieving those goals**. Different goal labels are likely to lead to different abstractions, with different abstract actions and variables [BG18b, BFG19].

A QNP P accounts for a given **goal graph** if the QNP provides an **abstraction** of the graph, and more precisely, if the graph G_i can be **embedded** in the QNP. For this, let $s(n)$ represent a valuation over the boolean and numerical variables in P, and let $s_0(n)$ represent the boolean projection of $s(n)$, namely $s(n)$ but with the numerical values $s(n)[X]$ replaced by *true* (false) if $s(n)[X] = 0$ is *true* (resp. false). Let each edge (n, n') in the input graphs G_i be assigned an action from P, denoted $a(n, n')$. The functions $s(\cdot)$ and $s_0(\cdot)$ map each nodes of the goal graphs into a real and boolean vector respectively, and the function $a(\cdot, \cdot)$ maps each edge of the goal graphs into a vector of actions. For the QNP P to account for the graphs G_i, these embeddings must comply with some **structural constraints** analogous to the ones in [BG18b, BFG19]:

- **Consistency:** the states $s(n)$ and the actions $a(n, n')$ assigned to nodes and edges must be consistent; e.g. if $a(n, n')$ decreases X, then $s(n')[X] < s(n)[X]$, etc.

- **Goals** must be distinguished; i.e., $s_0(n) \neq s_0(n')$ if exactly one of n and n' is a goal.
- **Soundness:** if $s_0(n_1) = s_0(n_2)$ and $a(n_1, n_1')$, there must be an edge (n_2, n_2') s.t. $a(n_2, n_2')$.

The first two constraints are direct; the third says that if the embedding makes two nodes n_1 and n_2 indistinguishable because all boolean and numerical variables in $s(n_1)$ and $s(n_2)$ have the same boolean value (the boolean value of a numerical variable X is $s_0[X]$), then the same abstract actions should be applicable in both. For generalization, it is necessary to find the **simplest QNP** that complies with these constraints which, for a bound on the possible number of boolean and numerical variables in the QNP, makes the problem a **constraint optimization** problem. For this, the state assignments have to be **regularized** by forcing the numerical variables to be have (non-negative) integer values, and the magnitude of increments and decrements to be 1. These are constraints for learning that do not apply to the learned QNP, where increments and decrements remain non-deterministic.

From the point of view of **machine learning**, the formulation yields a **graph embedding** that complies with crisp structural constraints [HYL17, ZYZZ18]. Embeddings of nodes and words in machine learning refer to mappings into high dimensional real vectors via **deep neural networks** that can trained to preserve certain properties, like that vectors of similar inputs must be closer than vectors of disimilar inputs [MCCD13]. Deep networks, however, are used to compute graph embeddings for settings like book recommendations, where the graph embeddings do not have to be symbolic or crisp. The proposed method is not intended to compete with such methods in those settings but it is aimed at producing crisp abstract representations when such representations exist, a property that does not hold for deep learning methods.

Semantic interpretation for generalized planning

The formulation above accepts a set of input goal graphs G_i and results in a general abstract planning representation in the form of a QNP whose solution via reductions to FOND planning [BDGGR17, BG18b] yields a general policy π that achieve the goal in the given graphs and in new problems that result in graphs covered by the same abstraction. In order **to apply the policy** π to a **new instance** expressed as a raw image, however, it is necessary to learn a mapping from images into QNP states, which are vectors of boolean and numerical values.

The formulation and the methods for learning such functions are similar to the ones discussed for planning where the semantic interpretation task was to learn the denotation of the learned predicate symbols in an image. In the generalized planning setting, however, this problem is **simpler** as there are neither objects nor predicate symbols but a fixed number of boolean and numerical variables. As before, the **targets** for learning their values from images are obtained from the embedding produced by the combinatorial optimization solver. If $w = v(n)$ is the image associated with node n and the embedding yields the values $s(n)[p] = \mathit{true}$ and $s(n)[X] = 5$, then the target values for p and X in the image w are *true* and 5 respectively.

Summary. Objective 2 is about delivering a **model-based approach to generalized planning from raw perceptions alone** from a set of input goal graphs. The problem of learning the QNP model that provides a common abstraction to the graphs is cast as a combinatorial optimization problem (**representation discovery**). The QNP abstraction is solved via existing reductions and off-the-shelf planners resulting in general plans. For applying these plans to new instances, the mapping from images to the fixed-size vector of boolean and numerical variables in the learned QNP must be learned using targets derived from the first part (**semantic interpretation**). This will be the first model-based approach to generalized planning that, like model-free, learning approaches, computes general plans form raw perceptions alone, but from abstract models that are learned.

8 Objective 3: Learning hierarchical representations

In order to compute plans it is often necessary or convenient to plan at different levels of abstractions with the constraints among the levels playing a crucial role. Indeed, a high-level plan is useless if it can't be brought down to the lowest level for execution [BY91]. In planning and reinforcement learning, hierarchies and **high level methods** or **options** have been defined mostly **by hand** [GA15, BAH19, SPS99, Die00, MRW07]. In spite of many attempts, **the key problem** that remains open and will be addressed in RLeap is how to construct crisp, symbolic hierarchical representations automatically, by building successive layers of abstractions from the bottom up.

The new elements that will be exploited for addressing this challenge are 1) the intimate connection between hierarchical planning and generalized planning, and 2) recent work by the PI team that shows how abstractions for generalized planning can be obtained automatically given the first-order symbolic representations of the domain [BG18b, BFG19]. About the first point, the relation between hierarchical and generalized planning is direct: the hierarchies are not tailored to particular instances but apply to **any** instance of the domain. For example, a hierarchical representations for the problem of delivering a key to a target location, similar to the standard Taxi problem used in reinforcement learning [Die00], involves the high level methods of moving to the key to pick it up, and moving to the target location to leave it, abstracting away the low level moves needed in each case. These methods are **general**: they apply to multiple versions of the problem where the initial, target locations, and maps change. Given the direct relation between hierarchical representations and generalized planning it is not difficult to understand the obstacles and limitations that existing and past approaches for deriving hierarchical representations have faced: in order to infer hierarchical representations that can be used to produce hierarchical strategies, it is necessary to understand how to infer abstract representations for producing general plans. Indeed, **hierarchical plans are hierarchical general plans** that may get instantiated in different ways just as **flat general plans** are.

The second new element to be exploited is a recent formulation for learning **abstract representations** for generalized planning given the first-order symbolic representation of the domain and sampled **goal traces** [BFG19]. The resulting abstractions are **sound**, meaning that the resulting general policies can always be instantiated to a ground plan. The abstraction, however, reduces abstract actions into single ground actions depending on the context. It is often necessary to create abstractions where single abstract actions reduce action sequences as well. One way to proceed with this is by **using the general policies for achieving some goals as abstract actions for achieving other goals.** For example, the general policy for achieving the goal $clear(x)$ can be used as an abstract action for achieving the goal $on(x, y)$, and the general policy for achieving $on(x, y)$ can be used as an abstract action for constructing arbitrary towers, and so on. This intuition of building successive layers of abstraction automatically from the bottom up is not new. What was missing was the **formal foundation** on which to formulate this intuition correctly. One particular challenge resulted from the non-deterministic effects of high-level actions on low level variables [MRW07]. A second challenge involved the constraints required for ensuring that abstract plans can be grounded and executed at the lowest level [BY91]. The work on **generalized planning** by the PI and others addresses these challenges: unlike classical plans, general plans are not affected by uncertain outcomes because the possible outcomes define instances covered by the general plans. Similarly, the notion of **sound abstractions** and the proposed methods for **learning** them ensure that the abstractions can be instantiated and executed at the lower levels.

Summary. Objective 3 is about building hierarchical symbolic representations for planning from raw perceptions. The new elements to be exploited are the close relation between hierarchical planning and generalized planning, and the methods developed for learning sound abstractions for generalized planning from given symbolic representations. Objective 3 will result in general and robust methods for building successive layers of reusable abstractions from the bottom up for supporting high-level planning.

9 Objective 4: Theory of representations for planning and learning

One of the most compelling explanations for the wide gap that exists between the complexity of planning [Byl94] and the performance of current planners [RW10, LG17a] is based on a notion of **planning width** introduced and explored by the PI and his team [LG12, LG17b].[1] The width w of a problem can go from 1 to the number of problem variables and a simple width-based planning algorithm, called IW, has been developed that runs in time that is exponential in the problem width. What makes this result relevant is that in most domain benchmarks, the width of the problems is bounded and small when the goals are single atoms. For example, the problems of achieving goals like $clear(x)$ and $on(x, y)$ in any Blocks world instance have width bounded by 1 and 2 respectively. This means that the width-based algorithm will solve any instance of the first problem in linear time, and any instance of the second in quadratic time, even though the size of the state space is exponential. Variations of this idea are used in state-of-the-art algorithms [LG17b, LG17a, FRLG17].

Objective 4 is aimed at a theoretical understanding of the type of planning representations that are easy to learn and that make planning easy too. The two conditions are related, as in the worst case, actions would affect many variables making both planning and learning hard (on average). For formulating and addressing the relevant questions, we will start by trying to connect three apparently disparate notions: **planning width**, **generalized planning**, and **linear decomposition** of the value function as studied and used in **reinforcement learning** [SB98, Ber95, Sze10]. One important open problem in reinforcement learning is **the characterization of the features** that make linear function approximations work in a given domain, i.e., the numerical features $\phi_i(s)$ that approximate the optimal value function as a linear combination of the form $\sum_i w_i \phi_i(s)$ where the coefficients w_i are constants that do not depend on the state. The problem is important because, even in (value-based) **deep reinforcement learning**, the role of the deep net is to produce the features that are linearly combined in the last layer [SPLC16b]. In spite of many efforts, there is not yet a compelling and general theory for characterizing the required features [SPLC16a, BDD$^+$19, BGP19]. Numerical features play also a critical role in the abstract language for generalized planning and in the proofs that establish that some domains have bounded width. Objective 4 is about developing a theory that relates the features that appear in these three different contexts that may have much in common. For example, the abstraction for achieving the general goal $clear(x)$ involves the numerical feature $n(x)$, that encodes the number of blocks above x, and the boolean feature H that tests if a block is being held. The length $V^*(s)$ of the shortest solution to achieve $clear(x)$ from a state s is $V^*(s) = 2n(x)+H$, where the state s is implicit in $n(x)$ and H, and H is counted as 1 or 0 according to whether H is true in s or not. Finally, the feature $n(x)$ appears in the proof that establishes that the width of all these instances is 1. Objective 4 is about studying these correspondences formally for understanding the structural reasons by which planning and learning over simple planning domains can be feasible and simple, even when they span exponential spaces.

10 Environments

RLeap aims at methods that are domain-independent and robust, and which can be evaluated empirically in the way that planners and SAT solvers are tested in competitions: by running them on new problems and domains. As mentioned in Part B1, the methods must be applicable to existing simulation platforms like ALE, for the Atari games [BNVB13], and GVG-AI, for general video-games [PLST$^+$15], but for getting there, we will move one step at a time, by building and dealing with simpler scenarios first, including those developed for evaluating representation learning in deep reinforcement learning [SSS$^+$15, ZLS$^+$18, CBBL$^+$19]. We do not have plans to get into real 3D environments like those involving robots. The ideas and methods developed in RLeap should be relevant then but are not sufficient, and we do not want to overstretch, preferring to stick to the set of fundamental research

[1] The notion borrows from the *treewidth* notion that plays a similar role in graphical models like constraint satisfaction problems and Bayesian networks [BB72, Fre82, Pea88, Dec03].

problems described where we can deliver crisp, solid, significant results.

11 Conclusion

RLeap is aimed at a concrete scientific problem and some of its ramifications: learning structural symbolic representations for planning from images without using prior symbolic knowledge. The problem is central for developing flexible, transparent, and trusworthy AI systems, but it is largely unsolved, with current ideas and methods proving to be inadequate. The challenge is big but we bring to the project a number of concrete, promising ideas and formulations, many of them introduced by the PI and his team. The potential gains are important, affecting the way AI systems are built, used, verified, and queried, while providing an integration of data-based learners and model-based solvers with their System 1 (reactive) and System 2 (deliberative) capabilities.

References

[ABL13] Carlos Ansótegui, Maria Luisa Bonet, and Jordi Levy. Sat-based maxsat algorithms. *Artificial Intelligence*, 196:77–105, 2013.

[AF18] Masataro Asai and Alex Fukunaga. Classical planning in deep latent space: Bridging the subsymbolic-symbolic boundary. In *AAAI*, 2018.

[AJOR19] Diego Aineto, Sergio Jiménez, Eva Onaindia, and Miquel Ramírez. Model recognition as planning. In *Proc. ICAPS*, pages 13–21, 2019.

[Asa19] Masataro Asai. Unsupervised grounding of plannable first-order logic representation from images. In *Proc. ICAPS*, 2019.

[BAH19] Pascal Bercher, Ron Alford, and Daniel Höller. A survey on hierarchical planning–one abstract idea, many concrete realizations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019). IJCAI*, 2019.

[BB72] Umberto Bertele and Francesco Brioschi. *Nonserial dynamic programming.* Academic Press, 1972.

[BBQ+19] André Barreto, Diana Borsa, John Quan, Tom Schaul, David Silver, Matteo Hessel, Daniel Mankowitz, Augustin Žídek, and Remi Munos. Transfer in deep reinforcement learning using successor features and generalised policy improvement. *arXiv preprint arXiv:1901.10964*, 2019.

[BdBMS19] Thiago P Bueno, Leliane N de Barros, Denis D Mauá, and Scott Sanner. Deep reactive policies for planning in stochastic nonlinear domains. In *AAAI*, volume 33, pages 7530–7537, 2019.

[BDD+19] Marc G Bellemare, Will Dabney, Robert Dadashi, Adrien Ali Taiga, Pablo Samuel Castro, Nicolas Le Roux, Dale Schuurmans, Tor Lattimore, and Clare Lyle. A geometric perspective on optimal representations for reinforcement learning. *arXiv preprint arXiv:1901.11530*, 2019.

[BDGGR17] Blai Bonet, Giuseppe De Giacomo, Hector Geffner, and Sasha Rubin. Generalized planning: Non-deterministic abstractions and trajectory constraints. In *Proc. IJCAI*, 2017.

[Ber95] Dimitri Bertsekas. *Dynamic Programming and Optimal Control, Vols 1 and 2.* Athena Scientific, 1995.

[BFG19] Blai Bonet, Guillem Francès, and Hector Geffner. Learning features and abstract actions for computing generalized plans. In *Proc. AAAI*, 2019.

[BG01] Blai Bonet and Hector Geffner. Planning as heuristic search. *Artificial Intelligence*, 129(1–2):5–33, 2001.

[BG+18a] Aniket Nick Bajpai, Sankalp Garg, et al. Transfer of deep reactive policies for mdp planning. In *Advances in Neural Information Processing Systems*, pages 10965–10975, 2018.

[BG18b] Blai Bonet and Hector Geffner. Features, projections, and representation change for generalized planning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4667–4673. AAAI Press, 2018.

[BG19] Blai Bonet and Hector Geffner. Qualitative numerical planning: Reductions and complexity. Forthcoming, 2019.

[BGP19] Bahram Behzadian, Soheil Gharatappeh, and Marek Petrik. Fast feature selection for linear value function approximation. In *ICAPS*, 2019.

[BHP17] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *AAAI*, 2017.

[BHvM09] Armin Biere, Marijn Heule, and Hans van Maaren. *Handbook of satisfiability*. IOS press, 2009.

[BJM+18] Fahiem Bacchus, Matti Juhani Järvisalo, Ruben Martins, et al. Maxsat evaluation 2018. 2018.

[BL16] Vaishak Belle and Hector J. Levesque. Foundations for generalized planning in unbounded stochastic domains. In *KR*, pages 380–389, 2016.

[BLCW09] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proc. ICML*, pages 41–48, 2009.

[BM09] Pascal Bercher and Robert Mattmüller. Solving non-deterministic planning problems with pattern database heuristics. In *Proc. German Conf. on AI (KI)*, pages 57–64. Springer, 2009.

[BNVB13] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[BPG09] Bonet Bonet, Hector Palacios, and Hector Geffner. Automatic derivation of memoryless policies and finite-state controllers using classical planners. In *Proc. ICAPS-09*, pages 34–41, 2009.

[BT03] Ronen I. Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research*, 3:213–231, 2003.

[BY91] Fahiem Bacchus and Qiang Yang. The downward refinement property. In *IJCAI*, pages 286–293, 1991.

[Byl94] Thomas Bylander. The computational complexity of STRIPS planning. *Artificial Intelligence*, 69:165–204, 1994.

[CBBL+19] Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: A platform to study the sample efficiency of grounded language learning. In *ICLR*, 2019.

[CCO+12] Amanda Coles, Andrew Coles, Angel García Olaya, Sergio Jiménez, Carlos Linares López, Scott Sanner, and Sungwook Yoon. A survey of the seventh international planning competition. *AI Magazine*, 33(1):83–88, 2012.

[CL90] Philip R Cohen and Hector J Levesque. Intention is choice with commitment. *Artificial intelligence*, 42(2-3):213–261, 1990.

[CPRT03] Alessandro Cimatti, Marco Pistore, Marco Roveri, and Paolo Traverso. Weak, strong, and strong cyclic planning via symbolic model checking. *Artificial Intelligence*, 147(1-2):35–84, 2003.

[CRT98] Alessandro Cimatti, Marco Roveri, and Paolo Traverso. Automatic OBDD-based generation of universal plans in non-deterministic domains. In *Proc. AAAI-98*, pages 875–881, 1998.

[CWH+08] Guillaume M JB Chaslot, Mark HM Winands, H JAAP VAN DEN HERIK, Jos WHM Uiterwijk, and Bruno Bouzy. Progressive strategies for monte-carlo tree search. *New Mathematics and Natural Computation*, 4(03):343–357, 2008.

[Dar18] Adnan Darwiche. Human-level intelligence or animal-like abilities? *Communications of the ACM*, 61(10):56–67, 2018.

[DCL08] Carlos Diuk, Andre Cohen, and Michael L Littman. An object-oriented representation for efficient reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 240–247, 2008.

[DDRD01] Sašo Džeroski, Luc De Raedt, and Kurt Driessens. Relational reinforcement learning. *Machine learning*, 43(1-2):7–52, 2001.

[Dec03] Rina Dechter. *Constraint Processing*. Morgan Kaufmann, 2003.

[Die00] Thomas G Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000.

[DRK08] Luc De Raedt and Kristian Kersting. Probabilistic inductive logic programming. In *Probabilistic Inductive Logic Programming*, pages 1–27. Springer, 2008.

[DSG17] Ivan Donadello, Luciano Serafini, and Artur D'Avila Garcez. Logic tensor networks for semantic image interpretation. In *Proc. IJCAI*, pages 1596–1602, 2017.

[EG18] Richard Evans and Edward Grefenstette. Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research*, 61:1–64, 2018.

[ES13] Jonathan Evans and Keith Stanovich. Dual-process theories of higher cognition: Advancing the debate. *Perspectives on psychological science*, 8(3), 2013.

[FLBPP19] Vincent François-Lavet, Yoshua Bengio, Doina Precup, and Joelle Pineau. Combined reinforcement learning via abstract representations. In *Proc. AAAI*, volume 33, pages 3582–3589, 2019.

[FM06] Zhaohui Fu and Sharad Malik. On solving the partial max-sat problem. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 252–265. Springer, 2006.

[FN71] Richard Fikes and Nils Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 1:27–120, 1971.

[Fre82] Eugene Freuder. A sufficient condition for backtrack-free search. *Journal of the ACM*, 29(1):24–32, 1982.

[FRLG17] Guillem Francès, Miquel Ramírez, Nir Lipovetzky, and Hector Geffner. Purely declarative action representations are overrated: Classical planning with simulators. In *Proc. IJCAI*, 2017.

[FYG04] Alan Fern, SungWook Yoon, and Robert Givan. Approximate policy iteration with a policy language bias. In *Advances in neural information processing systems*, pages 847–854, 2004.

[GA15] Ilche Georgievski and Marco Aiello. Htn planning: Overview, comparison, and beyond. *Artificial Intelligence*, 222:124–156, 2015.

[GB13] Hector Geffner and Blai Bonet. *A concise introduction to models and methods for automated planning*. Morgan & Claypool, 2013.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[GDL+17] Abhishek Gupta, Coline Devin, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Learning invariant feature spaces to transfer skills with reinforcement learning. 2017.

[Gef13] Hector Geffner. Computational models of planning. *Wiley Interdisciplinary Reviews: Cognitive Science*, 4(4):341–356, 2013.

[Gef14] Hector Geffner. Artificial intelligence: From programs to solvers. *AI Communications*, 2014.

[Gef18] Hector Geffner. Model-free, model-based, and general intelligence. In *IJCAI*, 2018.

[GG18] Tomas Geffner and Hector Geffner. Compact policies for fully observable non-deterministic planning as sat. In *Proc. ICAPS*, 2018.

[GGL+19] Artur d'Avila Garcez, Marco Gori, Luis C Lamb, Luciano Serafini, Michael Spranger, and Son N Tran. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *arXiv preprint arXiv:1905.06088*, 2019.

[GKKS12] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. *Answer set solving in practice*. Morgan & Claypool, 2012.

[GMH13] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[GNT16] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated planning and acting*. Cambridge University Press, 2016.

[GS19] Marta Garnelo and Murray Shanahan. Reconciling deep learning with symbolic artificial intelligence: representing objects and relations. *Current Opinion in Behavioral Sciences*, 29:17–23, 2019.

[HD11] Yuxiao Hu and Giuseppe De Giacomo. Generalized planning: Synthesizing plans that work for multiple environments. In *IJCAI*, pages 918–923, 2011.

[Hel02] Malte Helmert. Decidability and undecidability results for planning with numerical state variables. In *Proc. AIPS*, pages 44–53, 2002.

[Hel09] Malte Helmert. Concise finite-domain representations for pddl planning tasks. *Artificial Intelligence*, 173(5-6):503–535, 2009.

[HHG+17] Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, et al. Grounded language learning in a simulated 3d world. *arXiv preprint arXiv:1706.06551*, 2017.

[HLMM19] Patrik Haslum, Nir Lipovetzky, Daniele Magazzeni, and Christian Muise. *An Introduction to the Planning Domain Definition Language*, volume 13. Morgan & Claypool, 2019.

[HMT15] David Ellis Hershkowitz, James MacGlashan, and Stefanie Tellex. Learning propositional functions for planning and reinforcement learning. In *2015 AAAI Fall Symposium Series*, 2015.

[HN01] Joerg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, 2001.

[HPS04] Joerg Hoffmann, Julia Porteous, and Laura Sebastia. Ordered landmarks in planning. *Journal of Artificial Intelligence Research*, 22:215–278, 2004.

[HR04] Michael Huth and Mark Ryan. *Logic in Computer Science: Modelling and reasoning about systems*. Cambridge University Press, 2004.

[HTD90] James A Hendler, Austin Tate, and Mark Drummond. AI planning: Systems and techniques. *AI magazine*, 11(2):61–61, 1990.

[HYL17] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Engineering*, 40(3):52–74, 2017.

[HZRS16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778, 2016.

[IFT18] Murugeswari Issakkimuthu, Alan Fern, and Prasad Tadepalli. Training deep reactive policies for probabilistic planning problems. In *ICAPS*, 2018.

[IM19] León Illanes and Sheila A McIlraith. Generalized planning via abstraction: arbitrary numbers of objects. In *Proc. AAAI*, 2019.

[Kah11] Daniel Kahneman. *Thinking, fast and slow*. Farrar, Straus and Giroux, 2011.

[KB07] George Konidaris and Andrew G Barto. Building portable options: Skill transfer in reinforcement learning. In *IJCAI*, volume 7, pages 895–900, 2007.

[KBM+19] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

[Kha99] Roni Khardon. Learning action strategies for planning domains. *Artificial Intelligence*, 113(1-2):125–148, 1999.

[KKLP18] George Konidaris, Leslie Pack Kaelbling, and Tomas Lozano-Perez. From skills to symbols: Learning symbolic representations for abstract high-level planning. *Journal of Artificial Intelligence Research*, 61:215–289, 2018.

[KODR04] Kristian Kersting, Martijn Van Otterlo, and Luc De Raedt. Bellman goes relational. In *Proceedings of the twenty-first international conference on Machine learning*, page 59. ACM, 2004.

[KP18] Seyed Mehran Kazemi and David Poole. Relnn: A deep neural model for relational learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[KS06] L. Kocsis and C. Szepesvári. Bandit based Monte-Carlo planning. In *Proc. ECML-2006*, pages 282–293, 2006.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012.

[KW14] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[KZG+17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[Laz12] Alessandro Lazaric. Transfer in reinforcement learning: a framework and a survey. In *Reinforcement Learning*, pages 143–173. Springer, 2012.

[LB17] Brenden M Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. *arXiv preprint arXiv:1711.00350*, 2017.

[LBH15] Yann LeCun, Yoshua Bengio, and Goeffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[LG12] Nir Lipovetzky and Hector Geffner. Width and serialization of classical planning problems. In *Proc. ECAI*, 2012.

[LG17a] Nir Lipovetzky and Hector Geffner. Best-first width search: Exploration and exploitation in classical planning. In *Proc. AAAI*, 2017.

[LG17b] Nir Lipovetzky and Hector Geffner. A polynomial planning algorithm that beats lama and ff. *Proc. ICAPS*, 2017.

[LUTG17] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

[Mar18a] Gary Marcus. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, 2018.

[Mar18b] Gary Marcus. Innateness, AlphaZero, and artificial intelligence. *arXiv preprint arXiv:1801.05667*, 2018.

[MBB17] Marios C Machado, Marc G Bellemare, and Michael Bowling. A laplacian framework for option discovery in reinforcement learning. In *ICML*, pages 2295–2304, 2017.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. 2013.

[McD99] D. McDermott. Using regression-match graphs to control search in planning. *Artificial Intelligence*, 109(1-2):111–159, 1999.

[McD00] Drew McDermott. The 1998 AI Planning Systems Competition. *Artificial Intelligence Magazine*, 21(2):35–56, 2000.

[MDK+18] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems*, pages 3749–3759, 2018.

[MDR94] Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19:629–679, 1994.

[Men09] Elliott Mendelson. *Introduction to mathematical logic*. Chapman and Hall/CRC, 2009.

[MG04] Mario Martín and Hector Geffner. Learning generalized policies from planning examples using concept languages. *Applied Intelligence*, 20(1):9–19, 2004.

[MKS+15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[MMB12] Christian James Muise, Sheila A McIlraith, and Christopher Beck. Improved non-deterministic planning by exploiting state relevance. In *Proc. ICAPS*, 2012.

[MML14] Ruben Martins, Vasco Manquinho, and Inês Lynce. Open-WBO: A modular MaxSAT solver. In *Proc. SAT*, pages 438–445, 2014.

[MRW07] Bhaskara Marthi, Stuart J Russell, and Jason Andrew Wolfe. Angelic semantics for high-level actions. In *ICAPS*, pages 232–239, 2007.

[NAI+03] Dana S Nau, Tsz-Chiu Au, Okhtay Ilghami, Ugur Kuter, J William Murdock, Dan Wu, and Fusun Yaman. Shop2: An htn planning system. *Journal of artificial intelligence research*, 20:379–404, 2003.

[NM08] Bernd Neumann and Ralf Möller. On scene interpretation with description logics. *Image and Vision Computing*, 26(1):82–101, 2008.

[OGL+15] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. In *Advances in neural information processing systems*, pages 2863–2871, 2015.

[PC09] Giovanni Pezzulo and Cristiano Castelfranchi. Intentional action: from anticipation to goal-directed behavior. *Psychological Research*, 2009.

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[Pea18] Judea Pearl. Theoretical impediments to machine learning with seven sparks from the causal revolution. *arXiv preprint arXiv:1801.04016*, 2018.

[PLST+15] Diego Perez-Liebana, Spyridon Samothrakis, Julian Togelius, Tom Schaul, Simon M Lucas, Adrien Couëtoux, Jerry Lee, Chong-U Lim, and Tommy Thompson. The 2014 general video game playing competition. *IEEE Transactions on Computational Intelligence and AI in Games*, 8(3):229–243, 2015.

[PM18] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic Books, 2018.

[RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. IEEE conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[RF17] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7263–7271, 2017.

[RG09] Miquel Ramírez and Hector Geffner. Plan recognition as planning. In *Proc. IJCAI-09*, pages 1778–1783, 2009.

[Rin17] Jussi Rintanen. Schematic invariants by reduction to ground invariants. In *AAAI*, 2017.

[RN09] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2009. 3nd Edition.

[RW10] Sylvia Richter and Matthias Westphal. The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research*, 39(1):127–177, 2010.

[SA77] Roger C Schank and Robert P Abelson. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Lawrence Earlbaum, 1977.

[SAJJ16] Javier Segovia-Aguas, Sergio Jiménez, and Anders Jonsson. Hierarchical finite state controllers for generalized planning. In *Proc. IJCAI*, 2016.

[SB98] Richard Sutton and Andrew Barto. *Introduction to Reinforcement Learning*. MIT Press, 1998.

[Sch15] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[SG16] Luciano Serafini and Artur d'Avila Garcez. Logic tensor networks: Deep learning and logical reasoning from data and knowledge. *arXiv preprint arXiv:1606.04422*, 2016.

[SHM+16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[SHS+18] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[SIZ08a] Siddharth Srivastava, Neil Immerman, and Shlomo Zilberstein. Learning generalized plans using abstract counting. In *AAAI*, volume 8, pages 991–997, 2008.

[SIZ08b] Siddharth Srivastava, Neil Immerman, and Shlomo Zilberstein. Learning generalized plans using abstract counting. In *AAAI*, 2008.

[SIZ11] Siddharth Srivastava, Neil Immerman, and Shlomo Zilberstein. A new representation and associated algorithms for generalized planning. *Artificial Intelligence*, 175(2):615–647, 2011.

[SPLC16a] Zhao Song, Ronald E Parr, Xuejun Liao, and Lawrence Carin. Linear feature encoding for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4224–4232, 2016.

[SPLC16b] Zhao Song, Ronald E Parr, Xuejun Liao, and Lawrence Carin. Linear feature encoding for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4224–4232, 2016.

[SPS99] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[SRB+17] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017.

[SRBS16] Martin EP Seligman, Peter Railton, Roy F Baumeister, and Chandra Sripada. *Homo prospectus*. Oxford University Press, 2016.

[SSS+15] S. Sukhbaatar, A. Szlam, G. Synnaeve, S. Chintala, and R. Fergus. Mazebase: A sandbox for learning from games. *arXiv preprint arXiv:1511.07401*, 2015.

[SSS+17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[SYZ+18] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[Sze10] Csaba Szepesvári. Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1–103, 2010.

[SZG+15] Siddharth Srivastava, Shlomo Zilberstein, Abhishek Gupta, Pieter Abbeel, and Stuart Russell. Tractability of planning with loops. In *AAAI*, 2015.

[SZIG11] Siddharth Srivastava, Shlomo Zilberstein, Neil Immerman, and Hector Geffner. Qualitative numeric planning. In *AAAI*, 2011.

[TBF+18] Valentin Thomas, Emmanuel Bengio, William Fedus, Jules Pondard, Philippe Beaudoin, Hugo Larochelle, Joelle Pineau, Doina Precup, and Yoshua Bengio. Disentangling the independently controllable factors of variation by interacting with the world. *arXiv preprint arXiv:1802.09484*, 2018.

[TS11] Matthew E Taylor and Peter Stone. An introduction to intertask transfer for reinforcement learning. *AI Magazine*, 32(1):15, 2011.

[TTTX18] Sam Toyer, Felipe Trevizan, Sylvie Thiébaux, and Lexing Xie. Action schema networks: Generalised policies with deep learning. In *AAAI*, 2018.

[XZF+18] Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Broeck. A semantic loss function for deep learning with symbolic knowledge. In *ICML*, pages 5498–5507, 2018.

[YLWA05] Håkan LS Younes, Michael L Littman, David Weissman, and John Asmuth. The first probabilistic track of the international planning competition. *Journal of Artificial Intelligence Research*, 24:851–887, 2005.

[YS13] Haonan Yu and Jeffrey Mark Siskind. Grounded language learning from video described with sentences. In *Proc. ACL*, pages 53–63, 2013.

[YWJ07] Qiang Yang, Kangheng Wu, and Yunfei Jiang. Learning action models from plan examples using weighted max-sat. *Artificial Intelligence*, 171(2-3):107–143, 2007.

[ZK15] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361, 2015.

[ZLS+18] Amy Zhang, Adam Lerer, Sainbayar Sukhbaatar, Rob Fergus, and Arthur Szlam. Composable planning with attributes. In *ICLR*, 2018.

[ZYZZ18] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE transactions on Big Data*, 2018.