Reinforcement Learning Meets Cognitive Situation Management: A Review of Recent Learning Approaches from the Cognitive Situation Management Perspective

Andrea Salfinger
Department of Cooperative Information Systems
Johannes Kepler University Linz
Altenberger Strasse 69
4040 Linz, Austria
andrea.salfinger@cis.jku.at

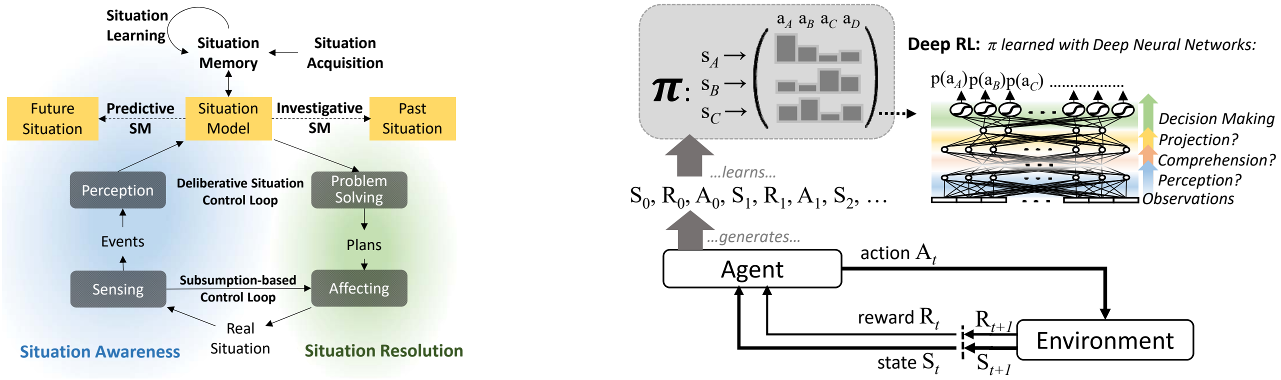
Abstract—**With *Reinforcement Learning* (RL), artificial agents learn reaching their goals "in the wild", i.e., from interacting with their environments. By learning to perform the correct action(s) in the given situation, RL thus adopts an *action* or *decision*-centric problem orientation. Conversely, the field of *Cognitive Situation Management* (CogSiMa), more originating from the *control* field, focuses on managing the encountered *situations*, i.e., environment states, such that the desired goal situations are reached or maintained. Whereas both fields of research thus appear complementary in pursuing similar overall goals, RL and CogSiMa have largely evolved independently from each other, leading to terminological gaps, misconceptions and unawareness of potentially related research. The present review attempts to bridge these gaps by providing an integrated framework highlighting the intersections between RL and CogSiMa: We outline how RL in real-world problem domains relates to CogSiMa, aim to bridge the terminological gaps between these distinct communities, and hope to provide the grounding for a cross-fertilization between these distinct research areas. We contribute a review of recent RL developments and discuss their implications and potential for CogSiMa.**

Keywords—**reinforcement learning, representation learning, cognitive situation management**

I. Introduction

Cognitive Situation Management. Humans, machines, as well as mixed human-machine-teams need to correctly comprehend and react upon the situations encountered in their environment in order to successfully achieve their goals. As characterized in Endsley's influential process model [1], humans' mental steps for gaining this *situation awareness* (SAW) involve the (i) *perception* of the elements in the environment, (ii) the *comprehension* of their relations, to understand the overall situational picture, and (iii) the *projection* of their status in the future, in order to understand how the situation will develop. The resulting mental state of SAW then provides the basis for *resolution*, i.e., the decision making about which actions are appropriate in the given situation. Whereas human factors research has been investigating why and how humans make errors in these phases (propagating to a critical loss of SAW with often drastic consequences), and how these can be prevented by means of suitable training and tool support, information fusion (IF) research focused on the complementary effort of how situation assessment can be implemented by machines. Strikingly similar to the human process model, the acknowledged JDL data fusion model [2] partitions the machine-based information fusion process into the functional levels of estimating sensor measurements from the environment (Level 0), fusing these sensor data to estimates of the observed objects (Level 1), estimating the relations between the sensed objects, termed situation assessment (Level 2), and estimating their impacts and future states (Level 3). Integrating these different perspectives, *Cognitive Situation Management* (CogSiMa) provides a unified framework for scoping situation management issues across human and machine-based agents, and (potentially mixed human-machine) multi-agent teams, as of relevance for a variety of domains, from robotics and autonomous vehicles to environment supervision tasks exerted in control centers. CogSiMa studies detecting and affecting situations in complex dynamical systems, such that desired goal situations are reached or maintained [3], comprising the sensing and perceiving of relevant information from the monitored environment, recognizing the encountered situation(s), blending this information with past experiences to project the situation's development, and reasoning on this obtained situation picture to plan the adequate actions for affecting the situation such that the desired goal state can be reached (see Fig. 1a). As implied by the keyword *cognitive*, the ideal would be to achieve some form of higher-level *cognition*, i.e., explicitly *understanding* the encountered state of affairs, which is typically realized by using representations supporting explicit reasoning and planning. Thus, CogSiMa has traditionally been addressed with knowledge representation and reasoning approaches from *symbolic* Artificial Intelligence (AI) [4], i.e., ontologies and deductive reasoning, as in [5], [6].

(Deep) Reinforcement Learning. Conversely, the research field of Reinforcement Learning (RL) [7] adopts a less cognitively-driven perspective, but takes inspiration from reward-based learning in animals: Using RL, an agent learns optimal behavior in an experiential fashion, i.e., from interacting with its environment. Learning is guided by means of numeric rewards that provide incentives for desired behavior. Over time, the agent thus should learn which actions are beneficial in the encountered situations in order to maximize its achieved rewards. CogSiMa and RL hence overlap in their objectives, since agents need to assess the situations in their environment and undertake adequate actions in order to attain their goals. Whereas CogSiMa focuses on (deliberative) *reasoning* about the encountered and sought-after situations, RL attempts at autonomously *learning* suitable actions for the encountered situations, driven by the reward function. The emerging field of Deep RL (DRL), which is based on



(a)

(b)

Figure 1: (a) The basic framework of *situation management* (adapted from [3]). (b) The basic framework of *Reinforcement Learning* (extended from [7]).

the recent breakthroughs in *sub-symbolic* or *connectionist* AI commonly known as *Deep Learning* (DL), has further extended these learning capabilities by not only learning the decision making task itself (i.e., the associations of situations to actions), but also learning higher-level situation representations directly from low-level sensor data [8].

Contributions. In the light of this recent convergence of these distinct research fields, the present paper, part position paper, review and synthesis, attempts at examining the recent DRL developments from the CogSiMa perspective: We argue that these novel approaches to learning representations and decision making call for a reconsideration of the classic notion of situation modeling, which typically conceives situation models in terms of explicit, symbolic representations, to also incorporate *implicit* situation modeling the form of such learned representations. In this direction, we also highlight the current limitations of these approaches, and identify complementary directions of influence, for which we believe principles established in CogSiMa research might inform the interpretation of these models.

II. Learning Behavior: Mapping Situations to Actions

A. Background: Reinforcement Learning in a nutshell

To model that outcomes of an agent's decisions are governed by a combination of the agent's taken action and (stochastic) external influences beyond the agent's control, RL relies on the mathematical framework of *Markov Decision Processes* (MDPs). An agent's interaction with its environment is commonly formulated with a finite MDP defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p, \gamma)$ [7], where $\mathcal{S}$ is a set of environment states (or *situations*), $\mathcal{A}$ is a set of actions, $\mathcal{R}$ is a set of rewards, $\gamma \in [0, 1)$ is a discount factor on the rewards (to gradually decrease expected rewards in the more distant future), and $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \mathcal{R} \mapsto [0, 1]$ models the dynamics of the environment such that

$$p(s', r|s, a) \doteq Pr(S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a) \quad (1)$$

for all $s, s' \in \mathcal{S}$, $a \in \mathcal{A}$, and $r \in \mathcal{R}$. Def. (1) expresses the stochasticity of the agent's environment: As sketched in Fig. 1b (bottom), at time point t, the environment's current state S_t is state s. When the agent selects an action a for the current action A_t, the environment transitions into state s' in the next state state S_{t+1} and the agent receives reward r as the next reward R_{t+1} with probability $p(s', r|s, a)$. An agent's interaction with its environment can thus be recorded as a sequence $S_0, R_0, A_0, S_1, R_1, A_1, S_2, \ldots$. From recording such sequences, i.e., selecting actions and observing their outcomes, the goal of RL is to learn a policy π, which represents a mapping of the states to the actions that will subsequently lead to the maximum expected reward, i.e., $\mathcal{S} \mapsto \mathcal{A}$.

The agent's behavior is thus determined by its stationary policy π, corresponding to a deterministic or stochastic mapping of environment states to decisions of which action to take. RL algorithms often also involve a value function v_π, which describes the expected discounted sum of rewards $v_\pi(s) = E\big[\sum_{t=0}^{\infty} \gamma^t R_t | s_0 = s\big]$ obtained by following policy π from each state s. v_π thus provides a quantification of the *value* of this state/situation with respect to the agent's goal.

B. Synthesis – towards a unified framework

In the following section, we will first introduce a formulation which allows to adopt RL within the CogSiMa framework. We then shed further light on the motivation for doing so, by examining recent RL approaches' capabilities w.r.t. the functional tasks composing the CogSiMa control loop.

Commonalities & Key Differences. Whereas both CogSiMa and RL tackle the problem that situations in an environment are observed and should be changed to a desired goal situation, CogSiMa lacks the notion of a simple numeric *reward* function: Instead, it is based on observing a current situation S_t, which should be transferred into a goal situation S_g. The notion of *cognition* implies that ideally, the agent should have some understanding of how the situation is composed, i.e., how the situation elements are related, and what needs to be done in order to transfer it into the goal situation S_g, i.e., applying a suitable transformation function $f(S_t) = S_g$ (which may correspond to executing a sequence of atomic steps $f_3(f_2(f_1(S_t))) = S_g$). Thus, instead of being "blindly" driven by a reward signal, which does not necessarily require that the agent learns to *understand* the internal composition and impacts of the situation, but can simply establish an associative memory of situations and actions that turned out to be beneficial (as expressed by the policy function π), the cognition of the observed situation corresponds to understanding: What is the current situation S_t, what is the goal situation S_g, and what would need to be changed to get from S_t to S_g? Presumably, the agent will utilize deliberative reasoning in order to figure out the steps of how to transform S_t into S_g. This may also involve planning, as well as counter-factual analysis (i.e., performing a "what if?" analysis). CogSiMa naturally extends to managing multiple, different situations, as well as dynamically changing goals or developing situational behavior, and inherently allows to deal with hierarchical situations, i.e., complex goal situations that need to be decomposed into subgoals.

Reformulating the Problem. Despite these key differences, we argue that RL can be adopted within the CogSiMa framework. We propose a generic formulation of the CogSiMa problem suitable for RL, by expressing the reward signal in terms of a distance metric δ quantifying the distance between the actual situation S_t and the goal situation S_g:

$$\mathcal{R} = -\delta(S_t, S_g) \tag{2}$$

Thus, the highest obtainable reward would be a distance of 0 (i.e., that we have reached the goal situation). This (distance) metric can be implemented for various types of situations, involving quantitative comparisons of compositions, as well as natural metric distances (e.g., geographic distance). Alternatively, we can design any other metric which rewards the "closeness" to the goal situation, which allows for a flexible rewarding scheme that might facilitate the learning problem. Furthermore, this reformulation also provides room for adaptively changing goal situations, which simply correspond to a change of the distance frame of reference and the agent's corresponding policy, provided that some higher-level component performs this situation comparison and goal switching.

C. Reflexive vs. cognitive behavior, and what it has to do with Representation Learning

Situations. Both CogSiMa and RL base on the notion of *situations*, $\mathcal{S}$, corresponding to particular state of affairs in the observed environment. In the following, we will thus further investigate this pivotal interconnecting concept, and shed light on its different meanings. From a situated agent's perspective, its surrounding environment's state can be understood as the agent's situation, as common in robotic systems. From the control perspective, a remote agent (such as a human control center operator) controlling a large-scale environment may observe multiple concurrent situations independent of each other, which thus lends itself to the definition of Barwise and Perry characterizing situations as semantically coherent *parts of the world* that can be described by means of objects in relations [9], rather than the complete snapshot of the world at a particular point in time (as orginially defined in *Situation Calculus* [10]). Following Barwise and Perry's notion [9] (as also adopted by the JDL data fusion model [2]), CogSiMa focuses the high-level composition of this state of affairs by commonly defining situations as *objects in relations*, i.e., distilling the "big picture" meaning from understanding how the observed objects are related to each other (and how these relations – such as distances – are expected to change, for instance in the course of movement). Conversely, the focus of RL is set on *resolution*, i.e., learning the optimal actions in the encountered situations, guided by the reward function. What constitutes a situation, and how the agent obtains SAW beforehand, i.e., determines the set of situations $\mathcal{S}$ and performs *situation assessment* (by determining the current situation $S_t = s$), has not received considerable attention in the RL community – generally, s is equated with a sufficiently detailed description of the currently observed environment state. Historically, RL has often studied problems on clearly defined domains (such as board games), where these environment states $\mathcal{S}$ (such as particular board configurations) are concisely defined and enumerable. Hence, these situation descriptions can be assumed to be given, whereby some higher-level abstraction may have already been performed (e.g., mapping mirroring board configurations to the same state, i.e., actual game situation, see e.g. [11]). In continuous settings, such as robots moving in real-world environments, S_t may also correspond to the agent's fine-grained sensor measurements. Such a direct mapping from sensor measurements to actions (without any further higher-level information processing and fusion) would thus, in CogSiMa terms, correspond to a *subsumption-based* control loop (see Fig. 1a). Lacking any higher-level cognitive processing (i.e., *comprehension*) between sensing and affecting, subsumption-based control lends itself to learning reflexive or reactive behaviors not demanding higher-level cognition, such as basic robotics control tasks (e.g., locomotion or object manipulation) [12]. However, recent advances of RL on games such as Atari 2600 video games [8], the strategy game StarCraft II [13], or hide-and-seek [14] indicated that agents might have learned strategic behavior, suggesting they are indeed performing some form of situation assessment. Hence, these observations suggest that these agents internally implement a greater portion of the situation control loop shown in Fig. 1a, going beyond the reflexive subsumption-based control loop.

Implicit Situation Assessment. So how can RL agents potentially learn such advanced decision making (depending on some actual *comprehension* of the encountered situation) from low-level sensor measurements, without explicitly performing *situation assessment* beforehand[1]? Recent breakthroughs in RL base on Deep RL (DRL), in which the agent's behavior, i.e., the policy function π (and/or its value function v_π), is learned with deep neural networks (DNNs) [15]. Instead of manually devising suitable situation representations $\mathcal{S}$ for the agent's decision making task (in machine learning terms, this would correspond to *feature engineering*), DRL bases upon *representation learning* [8]: During training, the network learns to build suitable higher-level representations from its low-level input data while propagating these information

[1] For instance, the agents trained to learn playing Atari 2600 video games at each time point only received the last four frames of the game screen [8], thus had to perform the entire processing from the *perception* of this visual input (i.e., reconstructing objects from this pixel-level observations), to understanding the current and (potentially) projected game situation.

through its layers, such that these representations facilitate its ultimate decision making task at the network's final layer [8]. Hence, we argue that the resulting DNN can be conceived as performing *self-taught information fusion*, whereby the higher-level, internal (thus denoted *hidden*) representations of the DNN correspond to the learned *situation representations* $\mathcal{S}$, suggesting that the DRL agents indeed perform some form of *implicit situation assessment*. Consequently, the resulting policy networks internally may encode a greater portion of the situation management control loop (Fig. 1a and Fig. 1b), and during training presumably perform some form of *situation learning*. Essentially, DRL agents *jointly* learn *situation assessment* and *decision making* in their training phase. However, since the only observable outputs are the emitted actions, it thus remains unclear what situation abstractions the network has built of its observed environment, as we will examine next.

III. Learning Representations: Mapping Observations to Situations

Interpretability.[2] The intricacies of this representation learning with DNNs are grounded in the problem that the learned model (i.e., the trained DNN) essentially represents a *black box* for human users: The *specification* of the model's sought-after behavior, i.e., the function to be learned, is only given implicitly in the form of the specified dataset $\mathcal{X}$ (corresponding to the made experiences, in case of the RL agent). For realistic problems, the learned parametrized function f becomes too complex (e.g., may easily involve millions of parameters over several layers) to allow for any introspection of the learned representations. Despite significant research efforts and progress in *eXplainable Artificial Intelligence* (XAI) [16], [17], which aims at developing approaches allowing to understand the network's learned representations, building understandable models of their behavior, or explaining the network's decision making (see surveys in [18]–[20]), still no universal, principled approach for revealing DNNs' learned representations and behavior has been achieved to date [21]. Most XAI approaches have focused the visual perception domain, and aim at explaining the network's decisions on an individual sample, by highlighting the parts of the sample that were most influential for the network's decision. In DRL settings, the agent's learned behavior is mainly evaluated "from the outside" in dedicated experiments, to see whether its learned behavior "generalizes" – i.e., it has learned behavior that is applicable to situations not directly experienced during training, but also in similar environments. These empirical evaluations provide the grounding for examining whether the training procedure has induced the desired behavior, which is required to assure that learning has not picked up the wrong signal: For instance, DRL agents trained to play computer games have often failed to learn playing the game according to human rules, by discovering limitations in the physics simulator of the game engine that allowed them to succeed by exploiting these. As the optimization objective of DRL networks is implied by the reward function, agents may also learn unintended behaviors by exploiting ill-specified reward functions, termed "reward hacking" [22][3]. Since essentially both "situation assessment" and the actual "decision making" are interwoven in DRL agents' network architecture, the sources of inadequate behavior are thus difficult to pinpoint. This problem has been aggravated by the emergence of recent *end-to-end* architectures: Whereas *modular* pipelines consist of a processing sequence of individually trained networks (such as the sequential *perception-planning-action* pipeline of self-driving vehicles' controllers [23]), the recently popularized *end-to-end* architectures stack all components into a single network, with no intermediate outputs [8], [23], so that the entire decision making and information processing pipeline can be jointly optimized (*end-to-end training*). Hence, in end-to-end architectures, the whole situation assessment and decision making loop, ranging from *perception* (i.e., detecting the individual objects) over presumable *comprehension* and *projection* (if the network is capable of performing such reasoning) to the actual resolution, is entangled in a single network (as hypothesized in Fig. 1a, upper right part), thus further complicating dissecting and interpreting its learned behavior and representations.

Interpreting Situation Representations. Investigating the policy or value networks' higher-level representations has been mainly based on *Visual Analytics* approaches (e.g., [24], [25]). For instance, to analyze the high-level representations built by the DRL agent trained on Atari 2600 video games [8], its value network's penultimate layer (i.e., the internal representation used for its final action selection) was visualized with the low-dimensional embedding technique t-SNE. This analysis revealed clusters of game states that apparently were more driven by semantic (in terms of their value w.r.t. the game score) than perceptual similarity, thus suggesting some abstraction driven by the underlying game situation. Another clustering also revealed a close similarity between states built from human play and states built from agent play. For more recent, complex architectures, however, analysis has often been confined to only examining the agents' behavior and performance (e.g., [13]).

Enforcing High-Level Representations. Whereas methods for examining the learned representations are scarce, some approaches have explicitly aimed at enforcing learning higher-level (i.e., abstract) representations of the encountered environment by means of a dedicated network architecture. [26] proposed a DRL architecture using an encoder to learn *abstract representations* of the agent's environment, which could be conceived as the different situations in the encountered environment. Similarly, [27] learns a so-called *recurrent world model* of its observed environment (i.e., a model capturing the environment dynamics), which is used for projection and planning. However, such approaches for learning the latent representations of the environment in DRL with autoencoders have mainly been used as internal components to inform the agent's action planning, but have not been specifically investigated w.r.t. their role in the agent's decision making.

Cognitive Models. As an alternative XAI approach going beyond visualization, Somers et al. proposed employing a cognitive model to ascribe meaning to and explain a DRL agent's decisions [28]. They trained a DRL agent on a Starcraft II mini-task and traced it with a cognitive model formulated in ACT-R, which tracked the game's states and the agent's decisions using both *symbolic* domain knowledge initially defined by human subject matter experts (on the game's actual states and required strategies, i.e., action selections), as well as the *sub-symbolic* representations stored in the policy network's penultimate layer (i.e., the network's final internal representation used for action selection). The cognitive model tried to predict the DRL agent's actions using instance-based learning to retrieve its most similar experience of the internal network state. The representations and predictions of the ACT-R model have been analyzed to try identifying gaps in the presumably reconstructed knowledge of the DRL agent, and explaining its wrong decisions, which was mainly based on discovering game states and action decisions infeasible according to human expert knowledge.

Conclusions on XAI. As our discussion of the current landscape in XAI has revealed, means for an in-depth interpretation of these internal higher-level representations are currently lacking, given that the majority of approaches has been focused on perception-level tasks. Thus, it remains difficult to assess whether and how DNNs are capable of building generalizable situation representations, or rather memorize specific state of affairs. The actual capabilities of DNNs for performing such higher-level reasoning tasks are presently still unclear, and start gaining increasing interest, as we will examine in the following (whereby we will broaden our perspective to general learning with DNNs).

[2] Readers unfamiliar with *representation learning* may wish to consult section VI-A in the appendix first.

[3] `https://openai.com/blog/faulty-reward-functions/` provides a discussion and illustrating video on this issue.

Comprehension & Reasoning. Essentially, our question of interest can be phrased as "how much" of the CogSiMa control loop (see Fig. 1a) could be implemented with DNNs, and how to assess this level of "cognition". Over the past decade, DL has become the state-of-the-art technology for *perception* in most domains, such as computer vision [29]–[31], thus has proven superior in the statistical pattern recognition tasks vital for object detection. Conversely, its capabilities for reasoning, sense-making and *comprehension* (i.e., understanding the relations between the perceived objects and their implications), currently remain unclear and start receiving increased interest (e.g., [32]–[37]). It has been conjectured that the difficulties of DRL agents in building generalizable representations might be grounded in the *propositional fixation* of neural networks [38]–[40] (which means their expressivity would be comparable to propositional logic), as neural networks tend to have difficulties in learning relational information (as expressed by first-order logic and higher-order logics). However, a capability for relational abstraction would be key in RL settings, in order to account for the invariances between different combinatorial configurations of observed situations. Several lines of research have been proposed to tackle this problem: (i) Some proposals advocate for an *AI systems integration approach*, i.e., connecting a DL-based perception system for *symbol grounding* (i.e., mapping the sensed perceptions to individual symbols, thus, in JDL terms [2], performing object detection and tracking), to a classical symbolic reasoning module [39], [41]. (ii) The field of *neural-symbolic* reasoning [40] proposes explicit *neural-symbolic architectures* [42], i.e., attempts to implement symbolic reasoning within connectionist architectures. (iii) Some advocate designing specific neural network modules which implement a dedicated *inductive bias* to facilitate the learning of relational information [38], [43], [44], similarly to how Convolutional Neural Networks implement an inductive bias for processing spatial invariances (which yielded the breakthrough in image processing), or Recurrent Neural Networks implement an inductive bias for processing sequential information [38], [45]. Despite increasing research efforts in this direction, the (predominantly sub-symbolic) *learning vs.* (predominantly symbolic) *reasoning* gap currently still persists, demanding further foundational research on paradigms for overcoming this limitation.

Evaluation Beyond Test Set Accuracy. Examining DNNs' capabilities for abstraction and reasoning naturally demands more elaborate evaluation settings than the currently predominant paradigm of measuring simple statistics on held-out datasets. For instance, the independent evaluation of the answers output by Question-Answering (QA) models has been critiqued recently [46]: State-of-the-art QA models demonstrated poor consistency in their answers when being posed multiple, semantically coherent questions about the same input, thus indicating that the models lack an "understanding" of the underlying concepts (thus, do not seem to perform *grounding*, i.e., anchoring linguistic expressions to representations of their underlying concepts), but rather seemed to be driven by linguistic surface statistics. Consequently, Ribeiro et al. proposed a more elaborate evaluation setting, probing the consistency of models' predictions on different logical implications of the same question (such as presenting reformulations of the original question involving logical equivalence, necessary conditions and mutual exclusion). Ultimately, such experimental designs seeking to probe a model's reasoning capacities can be guided by the experimental settings and practices developed in those disciplines versed in measuring such reasoning capabilities in animals and humans, i.e., *cognitive psychology*, as proposed in a recent line of research [47]–[49]. Visual perception experiments from cognitive psychology have been employed to examine whether image processing DNN architectures expose a *shape bias* [32] or show the *Gestalt phenomena* [50] known from human visual perception. The *abstract reasoning* capabilities of DNNs have been probed with Raven's Progressive Matrices (RPMs), a visual IQ test [33]. In these automatically generated visual reasoning experiments, the resulting test matrices can only be solved if the network has been able to generalize the underlying rule that has been used for generating the data. The presented results confirmed the utility of network architectures implementing a relational inductive bias (the previously discussed class (iii)), while demonstrating the poor performance of conventional DNN architectures on these tasks. However, in the light that the best performing architecture just achieved around 60% accuracy, based on utilizing a huge number of training samples in order to learn these relations, this still represents a rather disappointing generalization regime, when compared to how living beings are capable of generalizing underlying rules just from a handful of examples. Similarly, specifically engineered experiments have been designed to probe DNN's capacities for other types of reasoning, such as *causal reasoning* [51]. In this sense, such careful experimental designs are clearly needed in order to assess – and research on improving – DNN's reasoning capacities.

Improving Generalization. Recent increasingly complex DRL experiments evidenced that the *complexity of the training environment* seems to crucially affect generalization, such as OpenAI's study on multi-agent cooperation between two agent teams playing hide-and-seek in a simulated environment [14]. The mixed collaborative-competitive setting between the two teams of agents apparently triggered the emergence of tool use and strategic behavior, as agents learned to utilize objects in their environment to successfully increase their rewards. Authors could identify the emergence of as many as six different strategies and counter-strategies developed by the two competing agent teams. As this experiment was set on standard DRL algorithms, the authors concluded the discovery of intelligent behavior (such as tool use) was driven by the mixed competitive-collaborative multi-agent setting, which provided the *autocurriculum* for the different, self-discovered "learning tasks". While these behaviors were qualitatively evaluated by human observation, authors also noted the lack of objective evaluation metrics for examining and assessing the agents' learned behavior. They proposed measuring the capability for *transfer learning*[4] as evaluation metric, and thus designed a task suite of five additional tasks (comparing the

[4] Transfer learning means that an agent trained on task A is transferred to solving another task B it has not been trained on, and measuring how well the agent is capable of transferring its previous knowledge to the new task.

transferred agents to agents trained from scratch). Since agents demonstrated transfer capabilities only on some of these tasks, authors concluded that their learned representations were too entangled to support robust transfer to other tasks.

Situatedness Aids Generalization. In a similar vein, recent experiments indicate that *situatedness*, or, in other terms, the embodiment of the agent can improve the agent's generalization abilities: In [35], two agents (one moving in a classical, two-dimensional grid-world setting, which thus perceived itself from the perspective of an external observer), and one navigating in a 3D-environment (which thus perceived its environment from the ego-perspective), had to perform the same tasks (learning to grab named elements located in the room, and putting them to specific locations). Although both performed the same underlying task, the situated agent exposed better generalization, presumably since its situated perspective allowed it to observe more "invariances" of the scenes and tasks due to the embodiment and increased physical interaction with the environment, which thus apparently aided in identifying the environment's compositional and relational characteristics. Similarly, OpenAI's experimenters conjectured that the emergence of naturalistic, strategic behavior of their multi-agent teams playing hide-and-seek was due to the rich environment the agents had been placed into, which was using a realistic physics engine (nonetheless, agents learned to exploit weaknesses in the physics engine, by learning box-surfing, a non-realistic behavior, representing another example towards reward hacking) [14]. In general, the level of generalization that can be achieved seems to be a function of the careful design of the training procedure, as has been also underscored by the experiments conducted in [52].

Concluding, we can observe a shift from evaluating DL systems' perception capabilities, to investigating more elaborate reasoning, sense-making and (situation) comprehension.

Evaluating DRL agents' SAW? In the light of these recent insights that more elaborate, psychologically inspired evaluation metrics are fruitful, we hypothesize that the evaluation of DRL agents might also benefit by drawing inspirations from the established principles in human factors research. Whereas DRL agents are currently evaluated by means of their obtained rewards and observing their behavior, we argue that a thorough evaluation of their learned behavior would actually need to involve examining the agent's SAW. We suggest that RL agents' performance might be evaluated subject to the same measurements of SAW like human controllers: Essentially, errors in the agent's behavior could be due to (1) errors in *perception*, i.e., the agent has not perceived all objects in its environment correctly, (2) errors in *situation assessment*, i.e., the agent has correctly recognized all objects, but did not correctly understand their relations, (3) errors in *projection*, i.e., the agent wrongly anticipated the objects' and relations' development, or, finally, (4) errors in *resolution*, i.e., the agent has correctly understood the situation, but made the wrong decision. To provide an illustrative example, consider the following scenario, for which the error sources would be analogous, no matter whether we consider a human driver or the controller of an autonomously driving vehicle: a vehicle changing its lane and crashing into another vehicle. This might have happened due to a lack of situation awareness at level 1 (*perception* – the driver did not see that the other vehicle already was on this lane / the vehicle's sensors did deliver faulty measurements, so the perception module failed to recognize this object). However, the error may also be attributed to a lack of SAW at level 3 (*projection* – the driver anticipated that the other vehicle would also change its lane, but it continued to move ahead / the vehicle's controller wrongly predicted the other driver's intent). We also note that the inherent difference between situations is not on the *object-level*, but of *relational* nature: For instance, the controller needs to learn that it needs to brake in the situation (i) when a pedestrian is currently crossing the crosswalk in front of it, and in the situation (ii) when a pedestrian is approaching the crosswalk, but not in the situation (iii) where the pedestrian is already leaving the crosswalk. The "objects" in these three situations, which the vehicle's sensors (camera, LiDAR etc.) would need to detect, notably the pedestrian and the crosswalk, are the same, the different "meaning" of these situations with respect to the vehicle's actions is given by the situation awareness levels (2) and (3) - on which portion of the crosswalk is the pedestrian located, and where is he/she moving towards next? Despite basically observing the same objects in situations (i), (ii) and (iii), the meaning of those situations is indeed different for the vehicle's controller (which correctly needs to issue *braking* or *not braking*).

Thus, whereas the plethora of D(R)L research has been focused on evaluating *perception*-level tasks, which can be assessed in precisely specifiable object detection and tracking benchmarks[5], we argue the need for research on inspecting DRL agents' higher-level representations, conforming to the agent's environment representations and understanding, by studying whether and how different configurations of the low-level input data are mapped to representations of the underlying same state of affairs (i.e., situation) in the policy network's internal representation. As our discussion on the current state of XAI has revealed, these goals seem to be beyond the capabilities of current XAI approaches. However, we argue that examining a DRL agent's behavior (i.e., its situation-action-mapping) naturally should also involve not only examining the agent's actions, but also the agent's built situation representations laying the grounding for its action selections, which essentially encode the agent's understanding of its observed environment. We hypothesize the established levels of SAW defined by Endsley could provide a rough measuring stick on the types of abstractions required for attaining SAW (perception of objects – understanding their relations – projections for anticipating future behavior), that presumably might need to be encoded in some form within in the network's internal representations.

IV. Related Work

The present paper discussed recent developments in DRL and representation learning from the CogSiMa perspective. Whereas also stressing DL for situational understanding, [53] focuses on the challenge of distributed fusion with DL architectures, without considering the broader CogSiMa context. General surveys on DL [29], [54] and DRL [15] have been conducted, as well as on

[5] e.g., https://motchallenge.net/

DL and DRL for different application domains, like autonomous driving [23], visual understanding for visuomotor control [15], or robotic control [12], [55]. However, none of these works has focused the specific challenges of CogSiMa, nor on interpreting D(R)L techniques w.r.t. the CogSiMa loop. The present work is in spirit of the reviews on the relational reasoning capabilities of current connectionist approaches [56] and creating abstractions in DRL [57], and complements these highly targeted reviews on current connectionist approaches by contributing an analysis from the CogSiMa perspective. Conversely, Garnelo and Shanahan [56] center on the narrow question how DNNs can reason on objects and relations, whereas Konidaris [57] reviews different types of state and action abstractions that have been developed in (D)RL. While these reviews thus contains some partial overlap to the present work, we have adopted a distinct and broader scope by examining the potential utility of current connectionist and DRL approaches for CogSiMa. Relational reasoning with connectionist approaches has also been discussed in [38], which propagates the use of inductive biases for facilitating the learning task, outlines inductive biases in current DL architectures and promotes the potential of graph neural networks for relational reasoning tasks. [37] provides an evaluative comparison of classic symbolic as well as recent connectionist approaches to relational reasoning on several benchmark datatsets. To the best of our knowledge, the present work represents a first attempt to holistically examining the potential for embedding recent connectionist and RL approaches into the framework of CogSiMa.

V. Discussion: Implications for Cognitive Situation Management

In the present paper, we have developed an integrated perspective of CogSiMa and (D)RL, and reviewed recent developments in connectionist learning from the Cognitive Situation Management lens. Due to the sheer breadth of the current D(R)L landscape, we could only provide a high-level bird's eye overview, selecting representative works to illustrate recent developments, and omitting technical detail (which can be followed up in the provided references) in favor of highlighting the general underlying key principles. We have contributed a reformulation to cast RL into the framework of CogSiMa, arguing that representation learning with DNNs might eventually provide a viable means for *situation learning* from complex environments, and that DRL offers an intriguing approach for jointly learning situation assessment and decision making. We have also highlighted these approaches' current limitations for interpreting and verifying the learned situations and behaviors (in addition to DL's known challenges, such as being data-inefficient, computationally costly, requiring extensive training time, and the *sim2real* gap[6]), and evaluating their ability for generalization. Advancements in XAI would be needed in order to attribute meaning to situation representations learned with DNNs. We have also delineated the currently debated limitations of presently available DNN architectures w.r.t. their abilities for comprehension and (relational) reasoning, and have discussed recently proposed endeavors for overcoming these issues. As our review has revealed, *comprehension* seems to be the upcoming challenge in AI[7], as we are currently lacking means for building agents capable of actual *cognition*, i.e., a (causal) understanding of their environment. In the following, we condense our main conclusions from this review, anchoring these recent developments to the CogSiMa framework:

- Over the past decade, DL has become the state-of-the-art technology for *perception* (i.e., object detection and tracking) [29], [30].
- DNNs' capability for performing *comprehension*/reasoning is not fully understood yet, representing an area of active research. Despite hopes that DL may provide a means for overcoming the symbol grounding problem, the gap between *learning* and *reasoning* still persists, demanding future research on how to integrate these paradigms [38], [56].
- AI applications currently often address *projection* by employing separate components (e.g., separate DNNs serving as forecasting and planning models), basing on modular architectures (e.g., [27]).
- RL increasingly incorporates *human experience*, by "bootstrapping" RL agents with *imitation learning* from human experts' recorded behavior traces. The resulting expert policies are then further refined with RL (which can exploit massive simulation environments, e.g., in the form of self-play [11] or competitive multi-agent training [13]).
- Recent works have indicated that *generalization* seems to be fostered by more elaborate training settings, such as *situated* agents [35] and multi-agent competition [14].
- Training and evaluation involves increasingly more structured experimental setups inspired from cognitive science, e.g., [32], [33], [47], [48], [50], [52].
- Interpretation of learned higher-level representations and behavior is still largely unsolved.

Regarding the latter aspect, we have proposed that evaluation of DRL agents' knowledge and representations might be structured along the SAW levels. Ultimately, we believe that the development of such structured SAW evaluations would hinge on the careful design of controlled experiments allowing to assess different situation representations, which, as discussed in this review, are currently emerging as useful tools in DL research.

Acknowledgments

This work has been funded by the Austrian Science Fund (FWF) under grant FWF T961-N31.

References

[1] M. R. Endsley, "Toward a Theory of Situation Awareness in Dynamic Systems," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 32–64, 1995.

[2] J. Llinas *et al.*, "Revisiting the JDL Data Fusion Model II," in *Proceedings of the Seventh International Conference on Information Fusion (FUSION 2004)*, 2004.

[3] G. Jakobson *et al.*, "A Framework of Cognitive Situation Modeling and Recognition," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, 2006.

[6] characterizing the problem of distribution change when transferring agent controllers trained in simulated environments to real-world environments

[7] This was also stressed in Yoshua Bengio's keynote at NeurIPS 2019, "From System 1 Deep Learning to System 2 Deep Learning" [58].

[4] E. Blasch *et al.*, "High Level Information Fusion (HLIF): Survey of models, issues, and grand challenges," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 27, no. 9, pp. 4–20, 2012.

[5] M. M. Kokar *et al.*, "Ontology-based situation awareness," *Information Fusion*, vol. 10, no. 1, pp. 83–98, 2009.

[6] C. Matheus *et al.*, "Lessons learned from developing SAWA: a situation awareness assistant," in *Proc. of the 8th International Conference on Information Fusion*, vol. 2, 2005.

[7] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, ser. Adaptive Computation and Machine Learning. MIT Press, 2018.

[8] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529 EP –, 2015.

[9] J. Barwise and J. Perry, *Situations and Attitudes*. MIT Press, 1983.

[10] J. Mccarthy and P. J. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence," in *Machine Intelligence*. Edinburgh University Press, 1969.

[11] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484 EP –, 2016.

[12] L. Tai *et al.*, "A Survey of Deep Network Solutions for Learning Control in Robotics: From Reinforcement to Imitation." [Online]. Available: http://arxiv.org/pdf/1612.07139v4

[13] O. Vinyals *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[14] B. Baker *et al.*, "Emergent Tool Use From Multi-Agent Autocurricula." [Online]. Available: http://arxiv.org/pdf/1909.07528v1

[15] K. Arulkumaran *et al.*, "Deep Reinforcement Learning: A Brief Survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[16] D. Gunning and D. Aha, "DARPA's Explainable Artificial Intelligence (XAI) Program," *AI Magazine*, vol. 40, no. 2, pp. 44–58, 2019.

[17] W. Samek *et al.*, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, 1st ed., ser. Lecture notes in artificial intelligence, 2019.

[18] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[19] D. V. Carvalho *et al.*, "Machine Learning Interpretability: A Survey on Methods and Metrics," *Electronics*, vol. 8, no. 8, p. 832, 2019.

[20] S. Chakraborty *et al.*, "Interpretability of deep learning models: A survey of results," in *2017 IEEE SmartWorld*. IEEE, 2017.

[21] W. Samek and K.-R. Müller, "Towards Explainable Artificial Intelligence," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer International Publishing, 2019, pp. 5–22.

[22] D. Amodei *et al.*, "Concrete Problems in AI Safety." [Online]. Available: http://arxiv.org/pdf/1606.06565v2

[23] S. Grigorescu *et al.*, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 35, no. 8, p. 2481, 2019.

[24] S. Greydanus *et al.*, "Visualizing and Understanding Atari Agents," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research. PMLR, 2018.

[25] F. Petroski Such *et al.*, "An Atari Model Zoo for Analyzing, Visualizing, and Comparing Deep Reinforcement Learning Agents," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 2019.

[26] V. François-Lavet *et al.*, "Combined Reinforcement Learning via Abstract Representations." [Online]. Available: http://arxiv.org/pdf/1809.04506v1

[27] D. Ha and J. Schmidhuber, "Recurrent World Models Facilitate Policy Evolution," in *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc, 2018, pp. 2450–2462.

[28] S. Somers *et al.*, "Explaining decisions of a deep reinforcement learner with a cognitive architecture," *ACI Journal Articles*, no. 124, 2018.

[29] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[30] Y. LeCun *et al.*, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[31] I. Goodfellow *et al.*, *Deep Learning*. MIT Press, 2016.

[32] S. Ritter *et al.*, "Cognitive Psychology for Deep Neural Networks: A Shape Bias Case Study," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, ser. Proceedings of Machine Learning Research. PMLR, 2017.

[33] A. Santoro *et al.*, "Measuring abstract reasoning in neural networks," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, ser. Proceedings of Machine Learning Research. PMLR, 2018.

[34] K. Yi *et al.*, "Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, 2018.

[35] F. Hill *et al.*, "Emergent Systematic Generalization in a Situated Agent," 2019. [Online]. Available: https://arxiv.org/abs/1910.00571v2

[36] S. Lapuschkin *et al.*, "Unmasking Clever Hans predictors and assessing what machines really learn," *Nature communications*, vol. 10, no. 1, p. 1096, 2019.

[37] S. Dumancic *et al.*, "A Comparative Study of Distributional and Symbolic Paradigms for Relational Learning," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2019.

[38] P. Battaglia *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv*, 2018.

[39] M. Garnelo *et al.*, "Towards Deep Symbolic Reinforcement Learning." [Online]. Available: http://arxiv.org/pdf/1609.05518v2

[40] A. S. d'Avila Garcez *et al.*, "Neural-Symbolic Learning and Reasoning: Contributions and Challenges USA, March 22-25, 2015," in *2015 AAAI Spring Symposia, Stanford University, Palo Alto, California, USA, March 22-25, 2015*. AAAI Press, 2015.

[41] K. Yi *et al.*, "CLEVRER: CoLlision Events for Video REpresentation and Reasoning," *CoRR*, vol. abs/1910.01442, 2019.

[42] A. S. d. Garcez *et al.*, "Towards Symbolic Reinforcement Learning with Common Sense," *CoRR*, vol. abs/1804.08597, 2018.

[43] A. Santoro *et al.*, "A simple neural network module for relational reasoning," in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc, 2017, pp. 4967–4976.

[44] ——, "Relational recurrent neural networks," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, 2018.

[45] M. Shanahan *et al.*, "An Explicitly Relational Neural Network Architecture." [Online]. Available: http://arxiv.org/pdf/1905.10307v2

[46] M. T. Ribeiro *et al.*, "Are Red Roses Red? Evaluating Consistency of Question-Answering Models," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[47] P. Clark and O. Etzioni, "My Computer Is an Honor Student — but How Intelligent Is It? Standardized Tests as a Measure of AI," *AI Magazine*, vol. 37, no. 1, p. 5, 2016.

[48] G. Marcus *et al.*, "Beyond the Turing Test," *AI Magazine*, vol. 37, no. 1, p. 3, 2016.

[49] B. Beyret *et al.*, "The Animal-AI Environment: Training and Testing Animal-Like Artificial Cognition," *ArXiv*, vol. abs/1909.07483, 2019.

[50] B. Kim *et al.*, "Do Neural Networks Show Gestalt Phenomena? An Exploration of the Law of Closure." [Online]. Available: http://arxiv.org/pdf/1903.01069v3

[51] I. Dasgupta *et al.*, "Causal Reasoning from Meta-reinforcement Learning." [Online]. Available: http://arxiv.org/pdf/1901.08162v1

[52] F. Hill *et al.*, "Learning to Make Analogies by Contrasting Abstract Relational Structure," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[53] S. Chakraborty *et al.*, "Deep learning for situational understanding," in *20th International Conference on Information Fusion*. IEEE, 2017.

[54] S. Pouyanfar *et al.*, "A Survey on Deep Learning," *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–36, 2018.

[55] N. Sünderhauf *et al.*, "The limits and potentials of deep learning for robotics," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 405–420, 2018.

[56] M. Garnelo and M. Shanahan, "Reconciling deep learning with symbolic artificial intelligence: representing objects and relations," *Current Opinion in Behavioral Sciences*, vol. 29, pp. 17–23, 2019.

[57] G. Konidaris, "On The Necessity of Abstraction," *Current Opinion in Behavioral Sciences*, vol. 29, pp. 1–7, 2019.

[58] Y. Bengio, "From System 1 Deep Learning to System 2 Deep Learning," *Keynote at NeurIPS 2019*, 11.12.2019.

[59] Y. Bengio *et al.*, "Representation learning: a review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[60] C. Zhang *et al.*, "Understanding deep learning requires rethinking generalization Toulon, France, April 24-26, 2017, Conference Track Proceedings," in *5th International Conference on Learning Representations, ICLR 2017*. OpenReview.net, 2017.

[61] ——, "A Study on Overfitting in Deep Reinforcement Learning," *CoRR*, vol. abs/1804.06893, 2018.

VI. Appendix

A. Learning Representations

Conventionally, creating a model of a phenomenon of interest involves having to understand its specifics in order to formulate an explicit model or representation of it. Conversely, DL bases on the idea of learning a model of the phenomenon solely from data it has generated. Representing a general mechanism for function approximation [7], neural networks can learn functions $f(\mathcal{X};\theta)$ on a given dataset $\mathcal{X}$, by finding a parametrization θ of the neural network that minimizes the error on this data set. This error, given by evaluating the so-called *loss* function, defines the learning goal, such as minimizing the classification error (if the dataset $\mathcal{X}$ comprises a set of labeled samples $\{(\boldsymbol{x}, y)\}$, for which the goal is to predict the label y for a given sample $\boldsymbol{x}$), or minimizing the reconstruction loss (if the goal is to learn a representation of $\mathcal{X}$, as attempted by so-called autoencoders [31]). The learning goal, i.e., obtaining such a parametrization θ that yields a (local) minimum for the loss on the given training data $\mathcal{X}$, is usually achieved by means of (stochastic) gradient descent. If f simply corresponds to a model of the data in the form of a higher-level representation, this is known as *representation learning* [59]. However, this approach also entails some inherent pitfall: The network has to learn a model of a process or phenomenon which it is only indirectly observing by means of a limited dataset generated by this process. Thus, it is actually minimizing the loss on the *observed data*, i.e., which we can conceive as a "proxy" of the process we intend to model, instead of the actual data-generating process itself. Naturally, the mathematical optimization objective of minimizing the loss on the observed data will result in zero loss if the network has learned a perfect representation of this limited training dataset. Given a network of sufficient size, i.e., which thus has a sufficiently large modeling *capacity*, after training converges the network might eventually perfectly fit the provided data [60]. However, we are not interested in learning the exact representation of the training data (i.e., exactly *memorizing* the training data), a problem termed *overfitting*. Rather, the network should learn the *underlying patterns and regularities* of the indirectly observed phenomenon, so that this learned model will also yield valid predictions on data from this phenomenon that have not been part of the training set, which is termed *generalization* (i.e., the network has extracted *generalizable* knowledge from the training set that can be applied to unseen samples, instead of simply memorizing each individual observed sample). In terms of our area of interest, situation learning, for instance, the DNN should learn how to extract the underlying common characteristics of a situation – on the level of the comprised objects and relations – but not just memorize every detail of a particular situation instance, which will not be comparable to similar encountered situations in the future. To prevent *overfitting* and enforce the network to focus on the underlying regularities and patterns, the model capacity and learning algorithm might need to be constrained (known as *regularization*), which thus enforces learning a meaningful compressed representation. However, if the capacity of the network is too limited, on the other hand, it might not have enough parameters to model the potentially complex underlying phenomenon, thus is *underfitting*. Since current research is still lacking a principled theory of how to determine the adequate model capacity for a given problem, model learning with DNNs currently needs to rest on an empirical approach: Models are selected in an iterative procedure involving repeated model learning and evaluations on independent test sets to estimate their generalization performance. In particular for DRL problems, overfitting results in brittle behavior [61]: While trained agents often performed well in the environments they have been trained on, performance broke down upon slight modifications of the environment, thus suggesting that the agents have failed to learn generalizable situation representations, but rather have memorized the specific environment states encountered during training. Thus, while *representation learning* would provide a powerful means for situation learning, it is often unclear whether deep neural networks have built generalized situation representations (which thus involve some form of abstraction), or have just memorized their specific observations.