Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)
Sister Conferences Best Papers Track

Detect, Understand, Act: A Neuro-Symbolic Hierarchical Reinforcement Learning Framework (Extended Abstract)*

Ludovico Mitchener[1†], **David Tuckey**[1], **Matthew Crosby**[1,2‡] and **Alessandra Russo**[1]
[1]Imperial College London
[2]DeepMind
{ludovico.mitchener19, david.tuckey17, a.russo}@imperial.ac.uk,matthewcrosby@deepmind.com

Abstract

We introduce Detect, Understand, Act (DUA), a neuro-symbolic reinforcement learning framework. The Detect component is composed of a traditional computer vision object detector and tracker. The Act component houses a set of options, high-level actions enacted by pre-trained deep reinforcement learning (DRL) policies. The Understand component provides a novel answer set programming (ASP) paradigm for effectively learning symbolic meta-policies over options using inductive logic programming (ILP). We evaluate our framework on the Animal-AI (AAI) competition testbed, a set of physical cognitive reasoning problems. Given a set of pre-trained DRL policies, DUA requires only a few examples to learn a meta-policy that allows it to improve the state-of-the-art on multiple of the most challenging categories from the testbed. DUA constitutes the first holistic hybrid integration of computer vision, ILP and DRL applied to an AAI-like environment and sets the foundations for further use of ILP in complex DRL challenges.

1 Introduction

Deep reinforcement learning (DRL) involves the use of neural networks as function approximators in a reinforcement learning (RL) setting [Sutton and Barto, 2018]. In recent years, DRL systems have worked well when applied to complex games [Berner *et al.*, 2019; Schrittwieser *et al.*, 2020]. However, it is unclear to what extent excelling at these video games can be seen as a real proxy for intelligence [Crosby *et al.*, 2020]. Current state-of-the-art DRL systems seldom exhibit the most basic of human cognitive faculties such as causal inference, spatial reasoning or generalisation [Garnelo and Shanahan, 2019]. For example, in a recent competition using the Animal-AI (AAI) testbed, the top submissions failed to solve commonsense physical reasoning tasks such as object permanence and spatial elimination [Crosby *et al.*, 2020]. Furthermore, DRL methods inherit the drawbacks of neural networks, such as opacity or non-interpretability, poor generalization to samples outside their training distribution and data inefficiency. They are purely reactive, i.e. they do not explicitly develop high-level abstractions, needed for causal or analogical reasoning, that could be reused across tasks [Garnelo *et al.*, 2016]. To address these shortcomings, which map exactly onto the main strengths of symbolic AI, we propose a novel neuro-symbolic framework that combines the strengths of DRL and symbolic learning using the options framework [Garnelo and Shanahan, 2019; Sutton *et al.*, 1999].

Our framework, called DUA, is divided into three main components: *Detect*, *Understand* and *Act*. The *Detect* component extracts an interpretable object representation, in the form of a logic program, from the raw data of the environment, using traditional methods from computer vision. The *Understand* component implements a novel Answer Set Programming (ASP) paradigm to learn a symbolic meta-policy over options using inductive logic programming (ILP), whereas the *Act* component uses individually trained DRL agents that implement options. The architecture may be loosely thought of as a two-systems solution [Kahneman, 2011; Booch *et al.*, 2021]: the DRL options represent the fast, reactive and non-interpretable facets of intelligence while the symbolic meta-policy learning is the substrate of the slow, logically rational and interpretable side of intelligence.

We evaluate our DUA framework on the AAI 2019 competition testbed and demonstrate several key benefits. Given a set of pre-trained options, we demonstrate few-shot learning by only requiring 7 training examples to learn a general meta-policy which transfers within and between tasks to compete on a testbed of 900 unseen arenas. Training only on those 7 examples, DUA achieves state-of-the-art in 7 testbed categories and above the top-10 average in 4 others. Its modular nature allows it to easily incorporate new options and update or learn new meta-policies to solve completely new types of tasks without having to retrain the whole system. Finally, DUA requires no environment rewards to learn meta-policies, making it particularly adept at extremely sparsely rewarded settings. This work constitutes the first holistic hybrid integration of computer vision, ILP and DRL able to solve commonsense physical reasoning tasks such as the animal cognition tasks in the AAI-like environment.

*This is an extended abstract of a paper that appeared in Machine Learning journal.

†Contact Author

‡Contributions made before joining DeepMind

2 DUA

DUA is a general framework that operates on two different levels of temporal abstraction. The low-level, referred to as the *micro-level*, operates in the same time and action space as the RL environment. The high-level, referred to as the *macro-level*, corresponds to DUA actions (or *options*), which persist across hundreds of environment timesteps. DUA has two types of policies, a *meta-policy*, at the macro-level, that maps symbolic states to options, and *options* themselves that map environment observations to discrete micro-level actions.

DUA is named after its three components: Detect, Understand and Act (see Figure 1 for instantiation of DUA in the AAI environment). The *Detect* module receives information from the environment at each timestep and filters it into a meaningful representation. For an agent in an RL environment, the role of the Detect module is to filter the raw and noisy image tensor into the salient features which are most useful to maximise its reward. The Detect module parses the image into a set of bounding boxes. Objects are tracked across frames and those that are no longer visible persist in memory for a preset number of timesteps. Finally, the Detect component translates the bounding boxes' information into an ASP program composed of ground facts. It also computes simple arithmetic-based heuristics over bounding boxes to detect relations between objects in the scene, such as relative position, and adds it to the ASP program. We call this set of generated ASP facts the *observables*. The *Understand* (reasoning and learning) module processes this symbolic representation of the environment and infers the correct option to initiate given the current state by using the learned meta-policy. The Understand component is itself split into two sub-components: 1) an ASP program containing the learned meta-policy (policy over options) and commonsense background knowledge, e.g. a goal is always present even if not visible, and 2) the ILASP learner which learns the meta-policy. When queried, the Understand module adds to its ASP program the set of *observables* and outputs the optimal option to execute. The answer sets of this program represent all of the possible options that can be selected at a given time. The meta-policy itself takes the form of a set of weak constraints that rank the answer sets and thus the possible options. The option to execute is the one corresponding to the optimal answer set. The set of weak constraints are learned from environment traces as described in Section 2.1.

The *Act* component is composed of options, which are pre-trained DRL agents that correspond to sub-goals. It receives the identifier of the option to execute, along with some configuration indicating the stopping criteria and its inputs. For example, when we climb an object with identifier X, the bounding box of the object X is fed as input to the climb policy which terminates when the agent has reached the peak of the ramp or times out. It oversees the execution of the option in the environment and then calls the Understand module upon its termination.

Each option takes as input a filtered version of environment observations based on the instructions of the Understand component. For example, if the Understand component decides to 'interact with object x', only features of the environment pertaining to object x will be fed to the corresponding option. The option will then execute until a stopping criterion is met and a new query to the Understand component is made to decide on the next option to execute. We train each option separately using Proximal Policy Optimisation [Schulman *et al.*, 2017] on custom training arenas.

2.1 Inductive Meta-Policy Learning

This section describes the core of our contribution, that is our approach for learning a symbolic *meta-policy* over options which we call *Inductive Meta-Policy learning* (IMP). We collect *meta-traces* from option-environment interactions and translate them into a learning from answer sets task. These traces are not the environment traces, but the sequence of states and actions as viewed from the macro-level in the Understand module: the state of the world (expressed in the ASP program), when it was queried and which option was then executed. The environment timesteps are ignored in these *meta-traces* as we are only interested in learning which option to choose, since the execution of such option is left to the Act module.

We formalise the collection of meta-traces as a set T of tuples $\langle G, P \rangle$, where G is a *meta-trace* and P is a boolean. Each tuple in T corresponds to a collected episode. A *meta-trace* G is composed of pairs of partially observable symbolic *meta-states* s and options o. A *meta-state* is composed of all detected *observables* at a single macro-step, along with all the high-level relations between the agent and the objects inferred (via the background knowledge in the ASP program) by the Understand module. In other words, a *meta-state* is the set of all the true logical atoms in the Understand module at a given macro-state (when the Understand module is queried). The *meta-trace* is then the sequence of "symbolic" meta-states of the system and the options executed after each of these states is observed. The boolean P for each episode represents the success or failure of the episode: -1 means the agent failed to solve the task and 1 means it succeeded. n is the number of *meta-traces*. Note that importantly, IMP, unlike RL methods, does not use environment reward. Instead, it only considers the binary outcome P: whether the *meta-trace* leads to success or failure.

In order to learn a meta-policy, we need to transform this set T into a learning from answer sets task. Meta-policy learning happens in three steps:

1. Collect the *meta-traces* by running the agent in the environment and at each macro-step randomly picking options to execute. We store the *meta-traces* along with their respective episode success in the set of tuples T.

2. We abstract each *meta-trace*: we map the state-option pairs in the *meta-traces* in T to a set T_a of tuples including the *abstract state-option* pairs and associated expected return. This step finds in T similar *state-option pairs* and combines them to obtain a value akin to a Q-value.

3. We map the generated set T_a into a learning from answer set task to learn the meta-policy π_{meta} using ILASP.

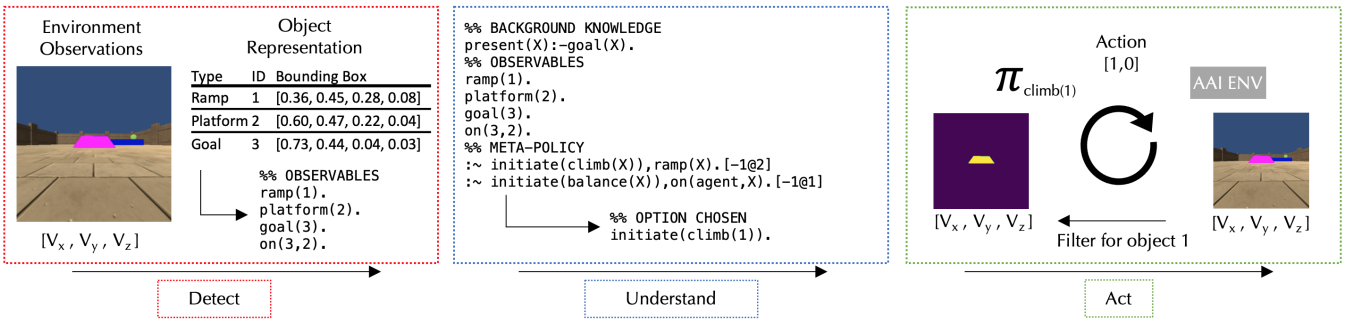


Figure 1: Example macro-step through the Detect, Understand, Act architecture.

3 Results

To evaluate DUA we compare the performance of our DUA framework to the submissions to the 2019 AAI competition, and analyse various aspects of inductive meta-policy learning. For AAI we implemented 9 options and created 7 training arenas. An extract of the final meta-policy learned is displayed below:

$$:\sim \texttt{initiate}(\texttt{climb}).[-1@11].$$
$$:\sim \texttt{initiate}(\texttt{collect}), \texttt{not lava}.[-1@8].$$
$$:\sim \texttt{initiate}(\texttt{explore}(\texttt{V1})), \texttt{occludes}(\texttt{V1}).[-1@5, \texttt{V1}].$$
$$:\sim \texttt{initiate}(\texttt{rotate}).[-1@1].$$

The above meta-policy may be read as a ranking over options constrained by certain relations. Below is a line by line translation in plain English, a given line being only used if there are no lines above it that are true:

```
If a ramp is available then climb it.
If there's no lava, collect multi-goals.
If an object occludes the goal, go explore it.
If nothing is visible, rotate until an object is visible.
```

As such, the final policy can be analysed to give insights into reasons for the behaviour of the agent and is therefore interpretable to some extent.

Figure 2a shows our method compared to the current high score within each category achieved by any of the 60 submissions submitted to the 2019 competition. We are comparing our model against the best of all submissions for each individual category. DUA achieves state-of-the-art results in all the categories related to the 7 training arenas, with the exception of *y-mazes*, where it still outperforms the top 10 average. This suggests that the meta-policy learned is robust and can generalise to a variety of cognitive reasoning tasks outside its training distribution. Our agent would have come 3rd overall in the competition.

3.1 Inductive Meta-Policy Learning

We now analyze various aspects of our inductive meta-policy learning (IMP) algorithm including the scalability of IMP at solving new problems by giving it more options, the very small sample of training arenas sufficient to learn a general meta-policy, and finally its convergence properties.

Transfer, Scalability and Generalisation

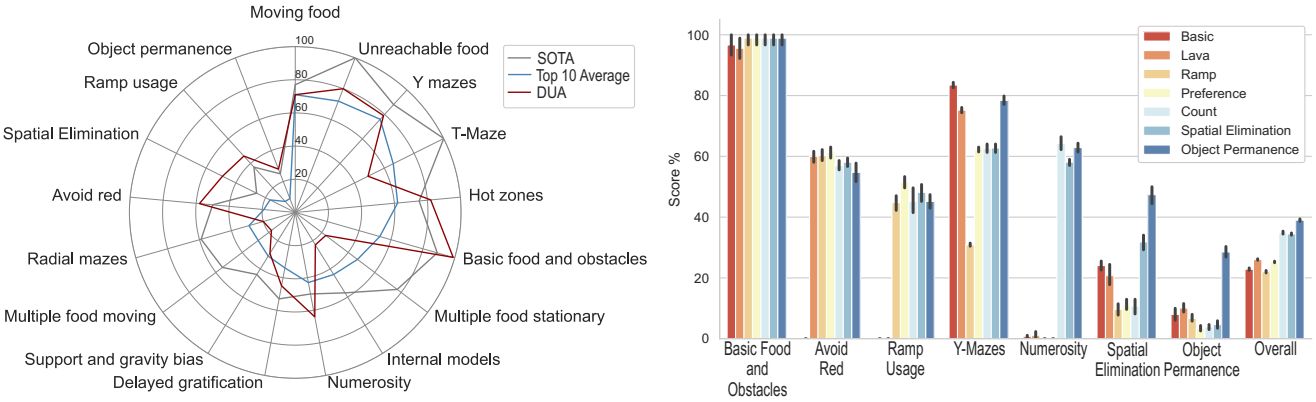
Unlike current DRL systems which usually require complete retraining to solve tasks outside their training distribution, it is sufficient to provide DUA with a single example of a new task and any options it may require. AAI contains a wide variety of tests for each category, yet we find that DUA only requires one example arena per category in order to generate the results in Figure 2.

To illustrate IMP's capacity at few-shot generalisation, we analyse the effect of incrementally adding one arena at a time to the training set. In Figure 2b we show how the scores improve as new training arenas and options are added. For example, the first system in red is just trained on the *basic food and obstacles* arena and does not have any of the options required to avoid red objects or climb ramps. Once we provide it with the *avoid* option and a single example of a training arena with a red object, the meta-policy adapts to include avoiding red objects and remains robust as more options are added. This is shown in Figure 2b by the jump in performance between the *Basic* bar and *Lava* bar on Avoid Red tasks.

Fine-Tuning and Convergence of the Meta-Policy

We also investigated how many successful *meta-traces* IMP requires to learn a general meta-policy effective on the whole testbed. With very few positive examples the overall meta-policy already becomes competent at a wide variety of tasks. However, the competency seems unstable for certain categories such as *numerosity* and *spatial elimination*. We interpret this as simple policies are very quickly learned enabling an immediate jump in performance. However, more intricate dependencies require more fine-tuning.

This quick gain in performance followed by fine-tuning instability is corroborated by analysing the evolution of Q values during training where optimal actions quickly separate from sub-optimal actions, but optimal actions then require further *meta-traces* to stabilise on slight preferences between optimal actions. Additionally, the number of abstract pairs visited converges to around 120 for a purely random policy. This indicates that during meta-policy training we have traversed the full search-space multiple times for each *abstract pair*. This is a direct benefit from our formulation of HRL to abstract environment states to higher-order meaningful representations and thus rendering large search spaces tractable.



(a) Radar Plot

(b) Incremental Training

Figure 2: (a) Radar plot comparing success rate per category wrt state-of-the-art (SOTA), average of top 10 2019 submission and our approach: DUA. (b) Category and overall performance by incremental training set.

3.2 Related Works

In recent years an increasing body of research has been dedicated to merging symbolic and neural systems in an attempt to reap the advantages of both [Marcus, 2020]. Neuro-symbolic methods may be separated into those that attempt to utilise symbols within neural networks themselves [Dong *et al.*, 2019; Liao and Poggio, 2017; Zhang and Sornette, 2017; d'Avila Garcez *et al.*, 2019; Manhaeve *et al.*, 2018] and those that connect the two by either using neural networks to translate unstructured data into a format suitable for symbolic systems or enhance deep systems with symbolic priors. Our approach is more directly related to the latter so we will focus on these kinds of methods here.

[Garnelo *et al.*, 2016] were amongst the first to show the promise of hybrid methods in RL. Using symbolic commonsense priors, such as object permanence, the authors augment their observation space for a simple RL task. They show that their method generalises better than a simple DQN to unseen, similar tasks. More recently, others have followed suit [Zamani *et al.*, 2017; Bougie *et al.*, 2018] in augmenting observation spaces with symbolic representations of their environments to give their agents strong informative priors. [Zamani *et al.*, 2017] uses a symbolic representation composed of subgoals that boost RL performance by providing intermediate rewards while [Bougie *et al.*, 2018] enhance image inputs with strong symbolic priors related to the environment. Both [Zamani *et al.*, 2017] and [Bougie *et al.*, 2018] demonstrate significant improvements in results over their purely DRL counterparts.

Another important approach comes from [Furelos-Blanco *et al.*, 2021]. Induction subgoal automata (ISA) uses ILP within the context of HRL, not only to learn the hierarchical structure of the automata, but also the sub-policies themselves. Neuro-symbolic techniques have also been used to efficiently verify the safety of DRL policies for use cases where safety violations are unacceptable [Anderson *et al.*, 2020].

4 Discussion & Future Work

We have proposed a general method for learning and enacting intelligent behaviour in virtual RL environments that outperforms previous approaches on challenging physical cognitive tasks. DUA acts effectively in continuous, noisy and high-density domains while maintaining a simplified and interpretable high-level representation of the environment and its actions. We further present a novel algorithm, *inductive meta-policy learning*, capable of learning from very few examples, which high-level actions to take, given a symbolic representation of the world in extremely sparsely rewarded environments. DUA contains the scaffolding to interface computer vision, neural actors and symbolic reasoner in a closed loop while IMP symbolically learns a high-level policy over options.

The framework may be applied to any typical RL environment. For each new environment, one needs to decide what are the observables to be used in the ASP representation, choose and train the options and finally implement a detector that translates the input from the environment into observables. It is worth noting that this framework works with any type of detector as this does not influence the shape of the logic program. The core of our framework (learning a symbolic meta-policy) adapts to any environment.

Training the set of options should require no or very little hyper-parameter tuning as each option focuses on learning one simple skill. In the AAI case, training all options was more than three orders of magnitude faster than other top submissions based on DRL methods which additionally require a considerable amount of hyper-parameter tuning. In this paper we only learn weak constraints that constitute our *meta-policy*. However, the ILASP system used by our DUA architecture is capable of learning any ASP program. For example, in this work we have encoded in the ASP reasoner the default assumption that "if an object is visible, then it is also present". Such assumptions could also be directly learned using ILASP. As such, this initial framework opens up the opportunity of learning more complex symbolic representations overlaid over deep neural enactors.

References

[Anderson *et al.*, 2020] Greg Anderson, Abhinav Verma, Isil Dillig, and Swarat Chaudhuri. Neurosymbolic reinforcement learning with formally verified exploration. *Advances in neural information processing systems*, 33:6172–6183, 2020.

[Berner *et al.*, 2019] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[Booch *et al.*, 2021] Grady Booch, Francesco Fabiano, Lior Horesh, Kiran Kate, Jonathan Lenchner, Nick Linck, Andreas Loreggia, Keerthiram Murgesan, Nicholas Mattei, Francesca Rossi, et al. Thinking fast and slow in ai. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 15042–15046, 2021.

[Bougie *et al.*, 2018] Nicolas Bougie, Li Kai Cheng, and Ryutaro Ichise. Combining deep reinforcement learning with prior knowledge and reasoning. *ACM SIGAPP Applied Computing Review*, 18(2):33–45, 2018.

[Crosby *et al.*, 2020] Matthew Crosby, Benjamin Beyret, Murray Shanahan, José Hernández-Orallo, Lucy Cheke, and Marta Halina. The animal-ai testbed and competition. In *NeurIPS 2019 Competition and Demonstration Track*, pages 164–176. PMLR, 2020.

[d'Avila Garcez *et al.*, 2019] Artur S. d'Avila Garcez, Marco Gori, Luís C. Lamb, Luciano Serafini, Michael Spranger, and Son N. Tran. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *FLAP*, 6(4):611–632, 2019.

[Dong *et al.*, 2019] Honghua Dong, Jiayuan Mao, Tian Lin, Chong Wang, Lihong Li, and Denny Zhou. Neural logic machines. In *International Conference on Learning Representations*, 2019.

[Furelos-Blanco *et al.*, 2021] Daniel Furelos-Blanco, Mark Law, Anders Jonsson, Krysia Broda, and Alessandra Russo. Induction and exploitation of subgoal automata for reinforcement learning. *J. Artif. Intell. Res.*, 70:1031–1116, 2021.

[Garnelo and Shanahan, 2019] 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.

[Garnelo *et al.*, 2016] Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. Towards deep symbolic reinforcement learning. *arXiv preprint arXiv:1609.05518*, 2016.

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

[Liao and Poggio, 2017] Qianli Liao and Tomaso Poggio. Object-oriented deep learning. Technical report, Center for Brains, Minds and Machines (CBMM), 2017.

[Manhaeve *et al.*, 2018] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. *Advances in Neural Information Processing Systems*, 31, 2018.

[Marcus, 2020] Gary Marcus. The next decade in ai: four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177*, 2020.

[Schrittwieser *et al.*, 2020] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Sutton *et al.*, 1999] Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

[Zamani *et al.*, 2017] Mohammad Ali Zamani, Sven Magg, Cornelius Weber, and Stefan Wermter. Deep reinforcement learning using symbolic representation for performing spoken language instructions. In *2nd Workshop on Behavior Adaptation, Interaction and Learning for Assistive Robotics (BAILAR) on Robot and Human Interactive Communication (RO-MAN), 26th IEEE International Symposium on*, 2017.

[Zhang and Sornette, 2017] Qunzhi Zhang and Didier Sornette. Learning like humans with deep symbolic networks. *arXiv preprint arXiv:1707.03377*, 2017.