Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)
Journal Track

Making Sense of Raw Input (Extended Abstract)*

Richard Evans[1] , **Matko Bošnjak**[1] , **Lars Buesing**[1] , **Kevin Ellis**[2] , **David Pfau**[1] , **Pushmeet Kohli**[1] and **Marek Sergot**[3]

[1]DeepMind
[2]Massachusetts Institute of Technology
[3]Imperial College London
richardevans@google.com

Abstract

How should a machine intelligence perform unsupervised structure discovery over streams of sensory input? One approach to this problem is to cast it as an *apperception task*. Here, the task is to construct an explicit interpretable theory that both explains the sensory sequence and also satisfies a set of unity conditions, designed to ensure that the constituents of the theory are connected in a relational structure.

However, the original formulation of the apperception task had one fundamental limitation: it assumed the raw sensory input had already been parsed using a set of discrete categories, so that all the system had to do was receive this already-digested symbolic input, and make sense of it. But what if we don't have access to pre-parsed input? What if our sensory sequence is raw unprocessed information?

The central contribution of this paper is a neuro-symbolic framework for distilling interpretable theories out of streams of raw, unprocessed sensory experience. First, we extend the definition of the apperception task to include ambiguous (but still symbolic) input: sequences of sets of disjunctions. Next, we use a neural network to map raw sensory input to disjunctive input. Our binary neural network is encoded as a logic program, so the weights of the network and the rules of the theory can be solved jointly as a single SAT problem. This way, we are able to jointly learn how to perceive (mapping raw sensory information to concepts) and apperceive (combining concepts into declarative rules).

1 Introduction

There are, broadly speaking, two approaches to interpreting the results of machine learning systems [Miller, 2018; Rudin, 2018; Murdoch *et al.*, 2019]. In one approach, *post-hoc interpretation*, we take an existing machine learning system, that has already been trained, and try to understand its inner state. In the other approach, *designing explicit already-interpretable machine learning systems*, we constrain the design of the machine learning system to guarantee, in advance, that its results will be interpretable.

In this paper, we take the second approach to unsupervised learning. Our system takes as input a temporal sequence of raw unprocessed sensory information, and produces an interpretable theory capturing the regularities in that sequence. It combines an unsupervised program synthesis system for constructing explicit first-order theories, with a binary neural network that transforms raw unprocessed sensory information into symbolic information that can be accessed by the program synthesis system. Thus, the system jointly synthesizes an explanatory symbolic theory, connected to a learned, sub-symbolic perceptual front-end.

1.1 Unsupervised Learning and Apperception

Consider a machine, equipped with various sensors, receiving a stream of sensory information. Somehow, it must make sense of this sensory stream. But what, exactly, does "making sense" involve, and how, exactly, should it be performed?

Unsupervised learning occupies a curious position within the space of AI tasks in that, while it is acknowledged to be of central importance to the progress of the field, it is also frustratingly ill-defined. What, exactly, does it mean to "make sense" of unlabelled data? There is no consensus on what the problem is, let alone the solution.

Self-supervised learning has emerged as a well-defined sub-field within unsupervised learning [Sermanet *et al.*, 2018; Pathak *et al.*, 2017]. Here, the task is to use the unlabelled sensory sequence as a source of supervised learning problems: we try to predict future states given previous states. Now, the vague under-specified unsupervised learning problem has been replaced by the well-defined task of predicting certain data points conditioned on others.

But there is more, we submit, to making sense than just predicting (or retrodicting) held-out states. Predicting future held-out states is certainly *part* of what is involved in making sense of the sensory given—but it is not, on its own, sufficient.

Recently, we proposed an alternative approach to unsupervised learning [Evans *et al.*, 2021]. The problem of "making sense" of sequences is formalised as an *apperception task*.

*This paper was originally published in *Artificial Intelligence* [Evans *et al.*, 2021].

Here, the task is to construct an explicit theory that both explains the sequence and also satisfies a set of unity conditions designed to ensure that the constituents of the theory—the objects, properties, and propositions—are combined together in a relational structure. We developed an implementation, the APPERCEPTION ENGINE, and showed, in a range of experiments, how this system is able to outperform recurrent networks and other baselines on a range of tasks, including Hofstadter's *Seek Whence* dataset [Hofstadter, 1995].

But in our initial implementation, there was one fundamental limitation: we assumed the sensory input was provided in symbolic form. We assumed *some other system had already parsed the raw sensory input into a set of discrete categories*, so that all the APPERCEPTION ENGINE had to do was receive this already-digested symbolic input, and make sense of it. But what if we don't have access to pre-parsed input? What if our sensory sequence is raw unprocessed information—a sequence of noisy pixel arrays from a video camera, for example?

1.2 Overview

Our central contribution is an approach for unsupervised learning of interpretable symbolic theories from raw unprocessed sensory data. We achieve this through a major extension of the APPERCEPTION ENGINE so that it is able to work from this raw input. This involves two phases. First, we extend the APPERCEPTION ENGINE to receive ambiguous (but still symbolic) input: sequences of disjunctions. Second, we use a neural network to map raw sensory input to disjunctive input. Our binary neural network is encoded as a logic program, so the weights of the network and the rules of the theory can be found *jointly* by solving a single SAT problem. This way, we are able to simultaneously learn how to perceive (mapping raw sensory information to concepts) and apperceive (combining concepts into rules).

We tested our system in three domains. In the first domain, the APPERCEPTION ENGINE learned to solve sequence induction tasks, where the sequence was represented by noisy MNIST images [LeCun *et al.*, 1998]. In the second, it learned the dynamics of *Sokoban* from a sequence of noisy pixel arrays. In the third, it learned to make sense of sequences of noisy ambiguous data without knowledge of the underlying spatial structure of the generative model.

This system is, to the best of our knowledge, the first system that is able to learn explicit provably correct dynamics of non-trivial games from raw pixel input. We discover that generic inductive biases embedded in our system suffice to induce these game dynamics from very sparse data, i.e. less than two dozen game traces. We see this as a step toward machines that can flexibly adapt and even synthesize their own world models [Ha and Schmidhuber, 2018], starting from raw sub-symbolic input, while organizing and representing those models in a format that humans can comprehend, debug, and verify.

2 Experiments

Here, we describe two of the three sets of experiments. For more details, see [Evans *et al.*, 2021].

2.1 Seek Whence with Noisy Images

The *Seek Whence* dataset is a set of challenging sequence induction problems designed by Douglas Hofstadter [Hofstadter, 1995].

The Data

In Hofstadter's original dataset, the sequences are lists of discrete symbols. In our modified dataset, we replaced each discrete symbol with a corresponding MNIST image.

To make it more interesting (and harder), we deliberately chose particularly ambiguous images. Consider Figure 1. Here, the leftmost image could be a 0 or a 2, while the next could be a 5 or possibly a 6. Of course, we humans are unphased by these ambiguities because the low Kolmogorov complexity [Li and Vitányi, 2008] of the high-level symbolic sequence helps us to resolve the ambiguities in the low-level perceptual input. We would like our machines to do the same.

For each sequence, the held-out data used for evaluation is a set of acceptable images, and a set of unacceptable images, for the final held-out time step. See Figure 1. We provide a slice of the sequence as input, and use a held-out time step for evaluation. If the correct symbol at the held-out time step is s, then we sample a set of unambiguous images representing s for our set of acceptable next images, and we sample a set of unambiguous images representing symbols other than s for our set of unacceptable images.

The Model

In this experiment, we combined the APPERCEPTION ENGINE with a three-layer perceptron with dropout that had been *pre-trained* to classify images into ten classes representing the digits $0-9$. For each image, the network produced a probability distribution over the ten classes.

We chose a threshold (0.1), and stipulated that if the probability of a particular digit exceeded the threshold, then the image possibly represents that digit. According to this threshold, some of the images (the first, third, eighth, and twelfth) of Figure 2 are ambiguous, while others are not.

Our pre-trained neural network MNIST classifier has effectively turned the raw apperception task into a disjunctive apperception task. Once the input has been transformed into a sequence of disjunctions, we apply the APPERCEPTION ENGINE to resolve the disjunctions and find a unified theory that explains the sequence.

2.2 Sokoban

In Section 2.1, we used a hybrid architecture where the output of a *pre-trained* neural network was fed to the APPERCEPTION ENGINE. We assumed that we already knew that the images fell into exactly ten classes (representing the digits $0-9$), and that we had access to a network that already knew how to classify images.

But what if these assumptions fail? What if we are doing pure unsupervised learning and don't know how many classes the inputs fall into? What if we want to jointly train the neural network and solve the apperception problem *at the same time*?

In this next experiment, we combined the APPERCEPTION ENGINE with a neural network, simultaneously learn-

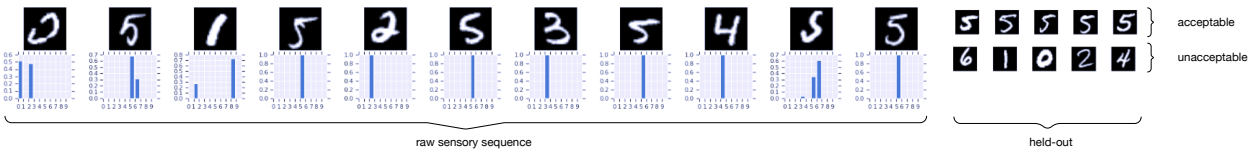

Figure 1: Seek-Whence tasks using MNIST images.

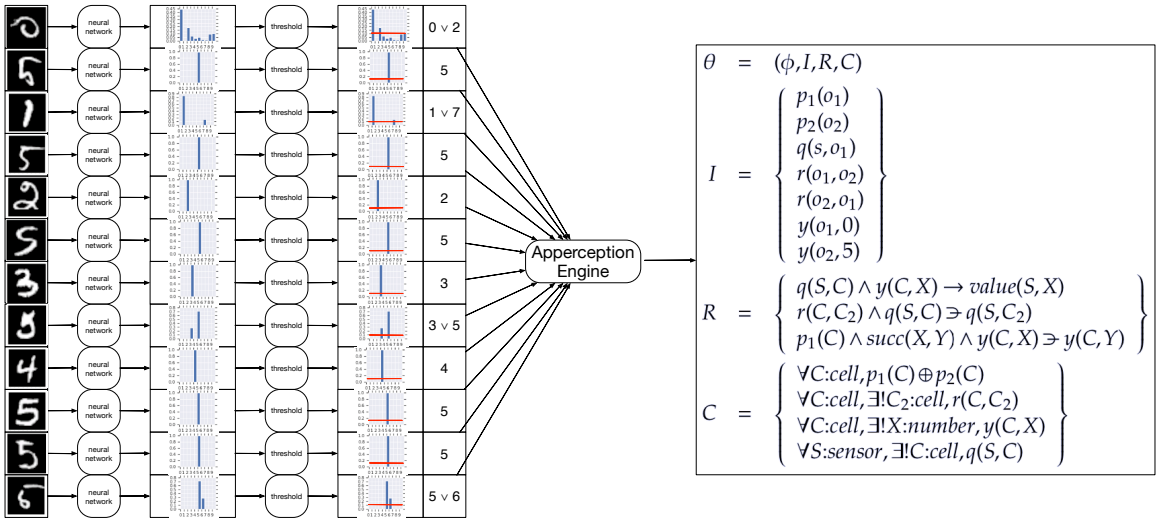

Figure 2: Solving Seek-Whence tasks from raw MNIST images.

ing the weights of the neural network and also finding an interpretable theory that explains the sensory given.

We used *Sokoban*[1] as our domain. Here, the system is presented with a sequence of noisy pixel images together with associated actions. The system must jointly (i) parse the noisy pixel images into a set of persistent objects, and (ii) construct a set of rules that explain how the properties of those objects change over time as a result of the actions being performed. We wanted the learned dynamics to be 100% correct. Although next-step prediction models based on neural networks are able, with sufficient data, to achieve accuracy of 99% [Buesing *et al.*, 2018], this is insufficient for our purposes. If a learned dynamics model is going to be used for long-term planning, 99% is insufficiently accurate, as the roll-outs will become increasingly untrustworthy as we progress through time, since 0.99^t quickly approaches 0 as t increases.

The Data

In this task, the raw input is a sequence of pairs containing a binarised 20×20 image together with a player action from $\mathcal{A} = \{north, east, south, west\}$. In other words, $\mathcal{R} = \mathbb{B}^{20\times20} \times \mathcal{A}$, and $(r_1, ..., r_T)$ is a sequence of (image, action) pairs from $\mathcal{R}$.

Each array is generated from a 4×4 grid of 5×5 sprites. Each sprite is rendered using a certain amount of noise (random pixel flipping), and so each 20×20 pixel image contains the accumulated noise from the various noisy sprite renderings.

Each trajectory contains a sequence of (image, action) pairs, plus held-out data for evaluation. Because of the noisy sprite rendering process, there are many possible acceptable pixel arrays for the final held-out time step. These acceptable pixel arrays were generated by taking the true underlying symbolic description of the *Sokoban* state at the held-out time step, and producing many alternative renderings. A set of unacceptable pixel arrays was generated by rendering from various symbolic states distinct from the true symbolic state. Note that the acceptable and unacceptable images are used only for evaluation, not for training. Figure 3 shows an example.

In our evaluation, a model is considered accurate if it accepts every acceptable pixel array at the held-out time step, and rejects every unacceptable pixel array. This is a stringent test. We do not give partial scores for getting some of the predictions correct.

The Model

In outline, we convert the raw input sequence into a disjunctive input sequence by imposing a grid on the pixel array and repeatedly applying a binary neural network to each sprite in the grid.

Figure 4 shows the best theory found by the APPERCEPTION ENGINE from one trajectory of 17 time steps. When neural network next-step predictors are applied to these sequences, the learned dynamics typically fail to generalise correctly to different-sized worlds or worlds with a differ-

[1] *Sokoban* is a puzzle game where the player controls a man who moves around a two-dimensional grid world, pushing blocks onto designated target squares.

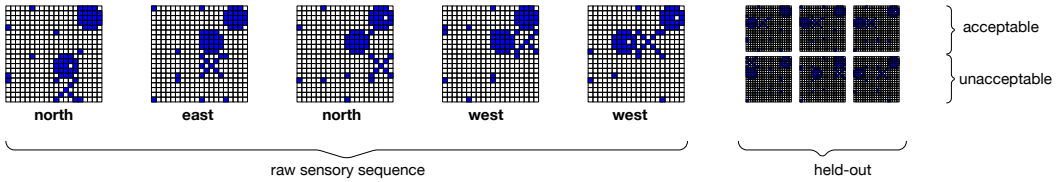

Figure 3: The *Sokoban* task. The input is a sequence of (image, action) pairs. For the held-out time step, there is a set of acceptable images, and a set of unacceptable images. Note that the acceptable and unacceptable images are used only for evaluation, not for training.

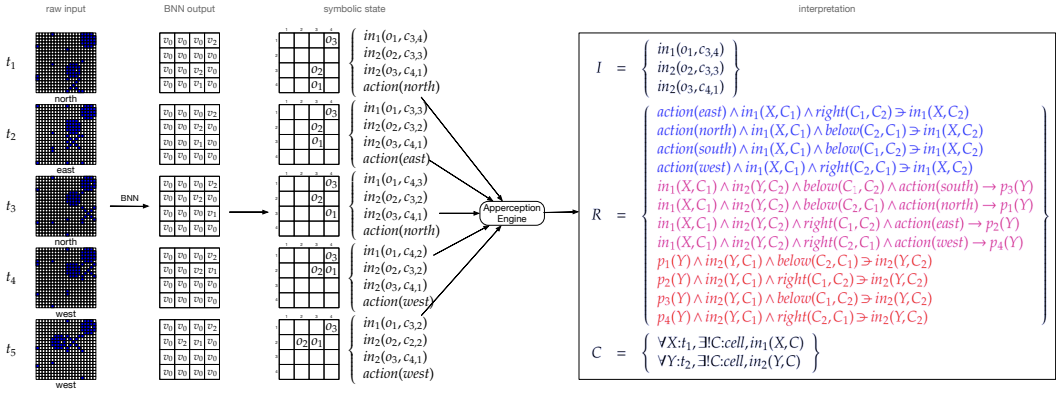

Figure 4: Interpreting *Sokoban* from raw pixels. Raw input is converted into a sprite grid, which is converted into a grid of types v_0, v_1, v_2. The grid of types is converted into a disjunctive apperception task. The APPERCEPTION ENGINE finds a unified theory explaining the disjunctive input sequence, a theory which explains how objects' positions change over time. The top four rules of R (in blue) describe how the man X moves when actions are performed. The middle four rules (in magenta) define four invented predicates $p_1, ...p_4$ that are used to describe when a block is being pushed in one of the four cardinal directions. The bottom four rules (in red) describe what happens when a block is being pushed in one of the four directions.

ent number of objects [Buesing *et al.*, 2018]. But the theory leaned by the APPERCEPTION ENGINE applies to all *Sokoban* worlds, no matter how large, no matter how many objects. Not only is this learned theory correct, but it is provably correct

References

[Buesing *et al.*, 2018] Lars Buesing, Theophane Weber, Sebastien Racaniere, SM Eslami, Danilo Rezende, David P Reichert, Fabio Viola, Frederic Besse, Karol Gregor, Demis Hassabis, et al. Learning and querying fast generative models for reinforcement learning. *arXiv preprint arXiv:1802.03006*, 2018.

[Chalmers *et al.*, 1992] David J Chalmers, Robert M French, and Douglas R Hofstadter. High-level perception, representation, and analogy: A critique of artificial intelligence methodology. *Journal of Experimental & Theoretical Artificial Intelligence*, 4(3):185–211, 1992.

[Clark, 2013] Andy Clark. Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences*, 36(3):181–204, 2013.

[Cropper *et al.*, 2019] Andrew Cropper, Richard Evans, and Mark Law. Inductive general game playing. *arXiv preprint arXiv:1906.09627*, 2019.

[Ellis *et al.*, 2015] Kevin Ellis, Armando Solar-Lezama, and Josh Tenenbaum. Unsupervised learning by program synthesis. In *Advances in Neural Information Processing Systems*, pages 973–981, 2015.

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

[Evans *et al.*, 2021] Richard Evans, José Hernández-Orallo, Johannes Welbl, Pushmeet Kohli, and Marek Sergot. Making sense of sensory input. *Artificial Intelligence*, 293:103438, 2021.

[Goodman *et al.*, 2011] Noah D Goodman, Tomer D Ullman, and Joshua B Tenenbaum. Learning a theory of causality. *Psychological review*, 118(1):110, 2011.

[Ha and Schmidhuber, 2018] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[Hofstadter, 1995] Douglas R Hofstadter. *Fluid Concepts and Creative Analogies*. Basic Books, 1995.

[Inoue *et al.*, 2005] Katsumi Inoue, Hideyuki Bando, and Hidetomo Nabeshima. Inducing causal laws by regular inference. In *International Conference on Inductive Logic Programming*, pages 154–171. Springer, 2005.

[Inoue *et al.*, 2014] Katsumi Inoue, Tony Ribeiro, and Chiaki Sakama. Learning from interpretation transition. *Machine Learning*, 94(1):51–79, 2014.

[Katzouris *et al.*, 2015] Nikos Katzouris, Alexander Artikis, and Georgios Paliouras. Incremental learning of event definitions with inductive logic programming. *Machine Learning*, 100(2-3):555–585, 2015.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Li and Vitányi, 2008] Ming Li and Paul Vitányi. *An introduction to Kolmogorov complexity and its applications*, volume 3. Springer, 2008.

[Miller, 2018] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 2018.

[Mitchell, 1993] Melanie Mitchell. *Analogy-making as perception: A computer model*. MIT Press, 1993.

[Murdoch *et al.*, 2019] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications. *PNAS*, 2019.

[Pathak *et al.*, 2017] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17, 2017.

[Ray, 2009] Oliver Ray. Nonmonotonic abductive inductive learning. *Journal of Applied Logic*, 7(3):329–340, 2009.

[Rudin, 2018] Cynthia Rudin. Please stop explaining black box models for high stakes decisions. *arXiv preprint arXiv:1811.10154*, 2018.

[Sermanet *et al.*, 2018] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from pixels. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141. IEEE, 2018.

[Shanahan *et al.*, 2019] Murray Shanahan, Kyriacos Nikiforou, Antonia Creswell, Christos Kaplanis, David Barrett, and Marta Garnelo. An explicitly relational neural network architecture. *arXiv preprint arXiv:1905.10307*, 2019.

[Shanahan, 2005] Murray Shanahan. Perception as abduction. *Cognitive Science*, 29(1):103–134, 2005.