Better Decisions through the Right Causal World Model

Elisabeth Dillies[*]
Department of Cognitive Science
Sorbonne University
elisabeth.dillies@gmail.com

Quentin Delfosse[*]
Department of Computer Science
Technical University Darmstadt
quentin.delfosse@tu-darmstadt.de

Jannis Blüml
Department of Computer Science
Technical University Darmstadt

Raban Emunds
Department of Computer Science
Technical University Darmstadt

Florian Peter Busch
Department of Computer Science
Technical University Darmstadt

Kristian Kersting
Department of Computer Science
Technical University Darmstadt

Abstract

Reinforcement learning (RL) agents have shown remarkable performances in various environments, where they can discover effective policies directly from sensory inputs. However, these agents often exploit spurious correlations in the training data, resulting in brittle behaviours that fail to generalize to new or slightly modified environments. To address this, we introduce the Causal Object-centric Model Extraction Tool (COMET), a novel algorithm designed to learn the exact interpretable causal world models (CWMs). COMET first extracts object-centric state descriptions from observations and identifies the environment's internal states related to the depicted objects' properties. Using symbolic regression, it models object-centric transitions and derives causal relationships governing object dynamics. COMET further incorporates large language models (LLMs) for semantic inference, annotating causal variables to enhance interpretability.

By leveraging these capabilities, COMET constructs CWMs that align with the true causal structure of the environment, enabling agents to focus on task-relevant features. The extracted CWMs mitigate the danger of shortcuts, permitting the development of RL systems capable of better planning and decision-making across dynamic scenarios. Our results, validated in Atari environments such as Pong and Freeway, demonstrate the accuracy and robustness of COMET, highlighting its potential to bridge the gap between object-centric reasoning and causal inference in reinforcement learning.

Keywords: reinforcement learning, world model, object-centric, causality

Acknowledgements

This research work has been funded by the German Federal Ministry of Education and Research, the Hessian Ministry of Higher Education, Research, Science and the Arts (HMWK) within their joint support of the National Research Center for Applied Cybersecurity ATHENE, via the "SenPai: XReLeaS" project as well as their cluster project within the Hessian Center for AI (hessian.AI) "The Third Wave of Artificial Intelligence - 3AI".

[*] Equal contribution

1 Introduction

Machine learning systems often tend to exploit spurious correlations in the training data [Schramowski et al., 2020, Stammer et al., 2021]. This is no different in RL settings, where agents learn misaligned policies that fail to learn the originally intended tasks by relying on shortcuts [di Langosco et al., 2022, Suau et al., 2024]. Such shortcut learning even occurs in the most simple Pong Atari game, for which Delfosse et al. [2024c] showed that both deep and symbolic RL agents are subject to misalignment. In Pong, the enemy is programmed to vertically align itself with the ball, leading to a quasi perfect correlation between the two objects vertical positions. RL agents thus learn to rely on the vertical position of the enemy to situate the ball position and effectively return it, as depicted in Figure 1. These shortcuts prevent the agents from generalizing, even to simplified versions of the environments, for which humans have no problem adapting. For example, if the enemy is hidden, or stops moving when the ball is going towards the player, the RL agents' performances drop. These misgeneralization to simpler scenarios are not specific to Pong, but are recurrent throughout the arcade learning environments [Delfosse et al., 2024a].

Many form of interpretable object-centric algorithms have recently been developed. They first extract object-centric (or symbolic) states, then rely on first order logic [Delfosse et al., 2023a], polynomial approximations and LLM explanations [Luo et al., 2024] or decision trees [Kohler et al., 2024]. They have been shown to be competitive alternatives to opaque approaches, performing on par with deep agents. Importantly, they provide some form of interpretability, that allow experts to detect and correct their potential misalignment behaviors. Yoon et al. [2023] demonstrate the higher robustness of object-centric agents in improving generalization and reducing reliance on spurious correlations.

However, object-centric symbolic reasoning does not break up correlations like the one between the enemy's paddle and the ball in Pong. If the enemy starts behaving differently, an agent relying on this spurious correlation and focusing on the enemy paddle will fail to return the ball, while an agent moving based on the ball position and velocity will not be affected by the correlation break. The difference between the relevance of both observable features (*i.e.* the enemy paddle and ball vertical positions) for our intended agent behavior lies in the true causal relations. If the agent utilizes these causal relationships in a causal world model, the independence of such mechanisms [Peters et al., 2017] ensures that the desired behaviour (*i.e.* returning the ball) applies in modified environments (*e.g.* with an enemy stopping after returning the ball). By disentangling these relationships, the agent can focus on task-relevant features, fostering the development of policies that generalize across diverse scenarios, including those with novel dynamics or adversarial interventions.

Even if novel benchmarks for exposing such misgeneralizations have been developed to test RL agents' robustness, (*e.g.* [Delfosse et al., 2024a] in the Atari domain), methods to automatically detect and correct spurious correlations are still underexplored. Integrating causal reasoning into RL agents offers a pathway toward human-like adaptability Yang et al. [2024], Lei et al. [2024]. By abstracting relevant features and modeling the causal structures underlying the observed phenomena, RL agents could autonomously overcome the limitations of shortcut-driven strategies.

In this paper, we introduce the **C**ausal **O**bject-centric **M**odel **E**xtraction **T**ool (COMET). Using their internal states, COMET aims to extract the interpretable causal world models (CWMs) of simulated environments. COMET detects the objects from the observation and then models causal relationships between the depicted objects' properties and the internal state (*e.g.* the RAM) that the emulated environment relies on to produce the observations. It then learns to extract *when* and *how* these variables are updated by the environment. It retrieves the relevant internal variables and uses the common reasoning abilities of an LLM to annotate these relevant internal states (thus improving its interpretability).

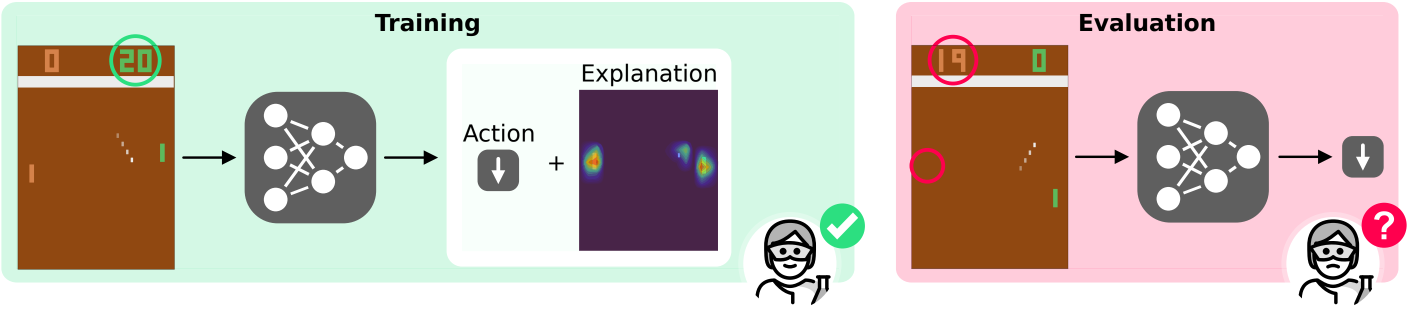

Figure 1: **Deep RL agents learn undetectable shortcut.** A deep (PPO) agent reaches the maximum score and selects the correct action in the depicted state of the training environment. The explanation map further leads external reviewers to think that the agent "understands" that it should return the ball behind the enemy (Left). In the test environment, where the enemy is hidden, the agent decides to go down, preventing it from catching the ball (Right). This illustrates that the agents learned to rely on the enemy's vertical position as an estimation of the ball's position.

2 Method

The algorithm, detailed in Algorithm 1, outlines the process for extracting a causal world model (CWM) using COMET. This method integrates object-centric reasoning with symbolic regression and semantic inference to map environmental dynamics comprehensively. Below, we describe each stage of the extraction process.

Overall, COMET extracts object-centric CWM by:

1. Mapping the internal state values to the objects' properties,
2. Modeling how this internal state evolves,
3. Annotating the relevant variables with their meaning.

COMET requires an executable environment, an executable policy (potentially random) and a large language model (LLM) for common sense reasoning. It first generates rollouts from the policy, retrieving the environment's internal states (EIS) and rendered RGB observations. It then uses an object discovery method (such as [Delfosse et al., 2023b, Zhao et al., 2023]) to extract the objects from each frame. Each object consists of a set of properties (such as the x, y, w, h bounding box coordinates or the value of objects such as scores, lives counter, etc.).

COMET then performs symbolic regressions, using PySR [Cranmer, 2023], to match each property with their corresponding internal states. It thus extracts a subset, R-EIS, of relevant internal states and their mapping to the different objects' properties. For example, the internal state $\mathtt{s}_i$ can encode the observed vertical position of the depicted ball object, with an offset of 14, thus following: $\mathtt{Ball.y} = \mathtt{s}_i - 14$.

To extract the underlying object-centric world model, *i.e.* understand how the objects evolve, COMET then searches both *if* and *how* these relevant internal states are updated at each step. Symbolic regressions are performed to map each already collected relevant state to internal states and actions. For example, our symbolic regression model can extract the following mapping: $\mathtt{s}_i = \mathtt{s}_i + \mathtt{s}_j$, with s_j another internal state, that here corresponds to the vertical speed of the ball. If new relevant states are found within the equation (*e.g.* s_j here), they are added to R-EIS. This relevant state regression is repeated until COMET has obtained the update conditions and equations of all the relevant states (which eventually happens, as the set of the environment's internal states is finite).

We then aim to improve the interpretability of the extracted world model, by annotating the relevant internal states with their meaning. To identify the semantics behind s_j, we use the common sense reasoning abilities of an LLM. Specifically, we provide ChatGPT (model 4o) with the equations that lead back to the object properties and ask for an annotation of the internal states' semantics. In the situation described above, the LLM correctly identifies[1] that s_j encodes the vertical velocity of the Ball. The provided semantics allow external reviewers to better identify each internal state purpose. Further, it can allow the LLM to reduce the set of internal state variables on which the regression is done. For example, the Ball's horizontal velocity in Pong is flipped when the ball bounces on one of the paddle. An LLM can detect that such event happens when the ball is colliding with another object, and reduce the input set to the other objects positions only. Even if facing a novel situation, for which the LLM might not know the task, a priori, its common sense reasoning can still lead it to identify that a sudden change in the a traveling object's velocity is due to a collision, and thus direct the search towards collision detection.

Algorithm 1 Causal Object-centric Model Extraction

```
Require: env, agent, LLM
init worldmodel
rgbs, EIS, actions ← sample(env, agent, nb_episodes)
objs ← detect(rgbs)
R-EIS ← find_relevant_EIS(objs.properties, EIS)
while R-EIS $\neq \varnothing$ do
    s ← R-EIS.pop()
    if s $\notin$ worldmodel then
        update_equation, update_condition ← find_hidden_state(EIS, actions)
        worldmodel(s) ← update_equation, update_condition
        R-EIS ← R-EIS + update_equation.variables
    end if
end while
annotate_variables(worldmodel, LLM)
return worldmodel
```

[1] https://chatgpt.com/share/6786f2ef-3ab4-8006-b5e3-8a3b29e92b2e, last accessed on 15.01.2024

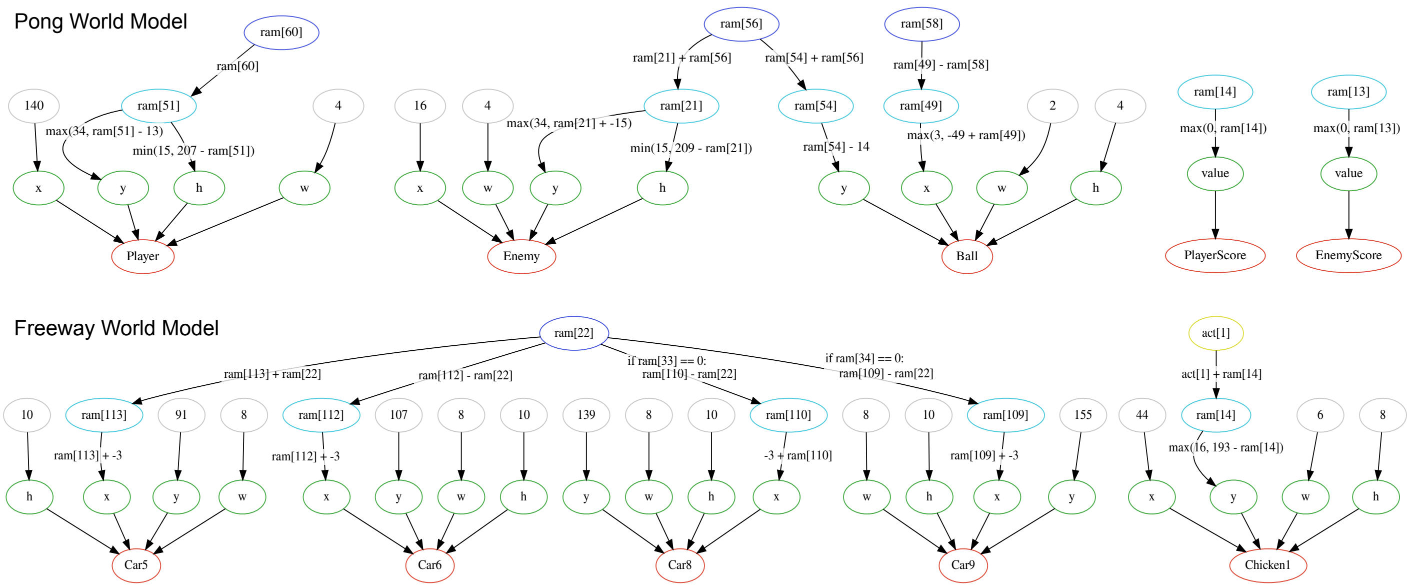

Figure 2: **Example of world models extracted by COMET.** Top: The extracted world model for the Pong environment, that consist of the player, the enemy, their scores, and the Ball. Bottom: Part of the Freeway world model (4 depicted cars out of 10). All the internal RAM variables extracted directly linked to the objects' properties (light blue) are on par with the annotated RAM provided by Anand et al. [2019]. The variables used to update these properties (dark blue) are valid. For example, `ram[58]` corresponds to the speed of the Ball on the x-axis.

3 COMETs extracted world models

This section showcases two extracted world models obtained from the Pong and Freeway environments. We use the emulator random access memory (RAM) as internal states of the environments. For these, we use the object extractor incorporated in OCAtari [Delfosse et al., 2024b] and sample transitions by letting a human agent play for 1 game episode (approximately 8000 steps). To identify if COMET finds the correct RAM values, we first refer to the Atari RAM annotations provided by Anand et al. [2019][2]. To assert if the other variables (from the update conditions) are correct, we directly alter them and check if they lead to the expected depicted modification. For example, we can set the horizontal speed of the ball to 0 and observe it moving vertically.

As depicted in Figure 2, COMET has correctly identified the properties' states for both games. For Pong, COMET correctly extracts the horizontal and vertical speed of the ball (*i.e.* `ram[58]` and `ram[56]` respectively). The state `ram[51]` is indeed updated to become `ram[60]` (updated based on the agent's selected action). However, `ram[56]` has also been retrieved to encode the player's speed. This is because the enemy is programmed to follow the ball (as detailed in the introduction). The enemy's speed, thus, indeed matches the ball's in the vast majority of the game transition. However, the enemy is programmed to catch the ball and thus follows it. If the ball is below the enemy, the enemy will decide to go down. However, this rule is more complicated than the one matching the enemy's speed. Thus, PySR favours the simpler rule. Two solutions can be applied here: allow the model to perform intervention, *i.e.* modify the speed of the ball to check if that leads to a direct modification of the enemy's y position update rule, or use the common sense reasoning of an LLM to select the most appropriate update rule.

In Freeway, the agent controls `Chicken1`, aiming to cross the road without getting hit by cars that move horizontally. The chicken's vertical position is updated based on the agent's selected action. It is incremented based on the direction of the joystick (*i.e.* act[1]). The different cars' positions are incremented for cars 1 to 5 and decremented for cars 6 to 10. However, to simulate the different speed for each car, the car `x` positions are updated at different paces. While the `x` positions of car5 and car6 get updated at each step, other cars like car8 and car9 use counters to be updated every 3 and 4 frames, respectively. The update conditions of car8 and car9 horizontal positions are thus correctly identified by COMET. Finally, in the regressed set `ram[22]` is constantly set to -1 (only set to 0 when the game is over, a state for which the cars' positions are indeed not updated). Altering `ram[22]` does not affect the speed of the car. While the regression leads to a correct result, an intervention altering the internal state value would allow COMET to identify and correct its mistake. The LLM's reasoning could also here detect that `ram[22]` represents the game being over (or not) from the regressions' inputs. A simpler alternative is to punish the symbolic regressor for using variable instead of constant.

[2] https://github.com/mila-iqia/atari-representation-learning/blob/master/atariari/benchmark/ram_annotations.py

4 Discussion and future work

In this paper, we introduced COMET and an algorithm to extract an interpretable object-centric causal world model by performing symbolic regression on the observable objects and providing semantics to the causal variables by leveraging the common sense reasoning capabilities of LLMs. One of the major discussion points of COMET is the fact that it accesses the hidden state of the environment. Most methods that extract CWM for RL agents are not given access to the internal states [Yang et al., 2024, Lei et al., 2024], which constitute more realistic settings. However, most RL agents are (at least) pretrained learning within virtual environments, even in industrial settings. COMET aims to extract the exact CWM from the environment, which can serve as the target CWM. Accessing the true CWM allows us to reimplement these exact environments using JAX.[3]. Initial benchmark of our JAX version of *e.g.* Pong, using GPU-based parallelization (on an RTX2070) lead to speedups between 30 to 100, compared to the CPU execution of the original Atari gym version.

Our most important next step is incorporating the LLM's common reasoning abilities within COMET. We will use the LLM to generate symbolic functions (in julia and sympy), that can be used by PySR at regression time. We have made interventions on the internal relevant variables extracted by COMET to test the accuracy of the extracted model. We plan to integrate the interventions within COMET, to allow for correcting the CWM. Finally, we, of course, plan to extend our evaluations to more environments, notably from the ALE suite.

References

Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R Devon Hjelm. Unsupervised state representation learning in atari. *Advances in neural information processing systems*, 2019.

Miles Cranmer. Interpretable machine learning for science with pysr and symbolicregression. jl. *arXiv preprint*, 2023.

Quentin Delfosse, Hikaru Shindo, Devendra Singh Dhami, and Kristian Kersting. Interpretable and explainable logical policies via neurally guided symbolic abstraction. *Advances in Neural Information Processing (NeurIPS)*, 2023a.

Quentin Delfosse, Wolfgang Stammer, Thomas Rothenbacher, Dwarak Vittal, and Kristian Kersting. Boosting object representation learning via motion and object continuity. 2023b.

Quentin Delfosse, Jannis Blüml, Bjarne Gregori, and Kristian Kersting. Hackatari: Atari learning environments for robust and continual reinforcement learning. *Workshop on Interpretable Policies @ The Reinforcement Learning Conference*, 2024a.

Quentin Delfosse, Jannis Blüml, Bjarne Gregori, Sebastian Sztwiertnia, and Kristian Kersting. OCAtari: Object-centric Atari 2600 reinforcement learning environments. *Reinforcement Learning Journal*, 2024b.

Quentin Delfosse, Sebastian Sztwiertnia, Mark Rothermel, Wolfgang Stammer, and Kristian Kersting. Interpretable concept bottlenecks to align reinforcement learning agents. In *Advances in Neural Information Processing (NeurIPS)*, 2024c.

Lauro Langosco di Langosco, Jack Koch, Lee D. Sharkey, Jacob Pfau, and David Krueger. Goal misgeneralization in deep reinforcement learning. In *International Conference on Machine Learning ICML*, 2022.

Hector Kohler, Quentin Delfosse, Riad Akrour, Kristian Kersting, and Philippe Preux. Interpretable and editable programmatic tree policies for rl. In *European Workshop on Reinforcement Learning*, 2024.

Anson Lei, Bernhard Schölkopf, and Ingmar Posner. Spartan: A sparse transformer learning local causation. 2024.

Lirui Luo, Guoxi Zhang, Hongming Xu, Yaodong Yang, Cong Fang, and Qing Li. End-to-end neuro-symbolic reinforcement learning with textual explanations. *Forty-first International Conference on Machine Learning*, 2024.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2020.

Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In *Conference on Computer Vision and Pattern Recognition, CVPR*, 2021.

Miguel Suau, Matthijs T. J. Spaan, and Frans A. Oliehoek. Bad habits: Policy confounding and out-of-trajectory generalization in rl. *RLJ*, 2024.

Yupei Yang, Biwei Huang, Fan Feng, Xinyue Wang, Shikui Tu, and Lei Xu. Towards generalizable reinforcement learning via causality-guided self-adaptive representations. *arXiv preprint*, 2024.

Jaesik Yoon, Yi-Fu Wu, Heechul Bae, and Sungjin Ahn. An investigation into pre-training object-centric representations for reinforcement learning, 2023.

Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint*, 2023.

[3] https://github.com/k4ntz/JAXAtari