POSTER

Towards an Evolutionary Approach for Exploting Core Knowledge in Artificial Intelligence

ANDREA CALABRESE, Polytechnic of Turin, Turin, TO, Italy

STEFANO QUER, Polytechnic of Turin, Turin, TO, Italy

GIOVANNI SQUILLERO, Polytechnic of Turin, Turin, TO, Italy

ALBERTO PAOLO TONDA, Paris-Saclay University, Gif-sur-Yvette, Ile-de-France, France

Open Access Support provided by:

Polytechnic of Turin

Paris-Saclay University

Published: 14 July 2024

Citation in BibTeX format

GECCO '24 Companion: Genetic and Evolutionary Computation Conference Companion
July 14 - 18, 2024
VIC, Melbourne, Australia

Conference Sponsors:
SIGEVO

Towards an Evolutionary Approach for Exploting Core Knowledge in Artificial Intelligence

Andrea Calabrese*
Stefano Quer*
Giovanni Squillero*
Politecnico di Torino
Torino, Italy

Alberto Tonda*
UMR 518 MIA-PS
INRAE, Université Paris-Saclay
Palaiseau, France
Institut des Systèmes Complexes Paris-Ile-de-France
Paris, France

ABSTRACT

This paper presents a proof of concept for a novel evolutionary methodology inspired by *core knowledge*. This theory describes human cognition as a small set of innate abilities combined through compositionality. The proposed approach generates predictive descriptions of the interaction between elements in simple 2D videos. It exploits well-known strategies, such as image segmentation, object detection, simple laws of physics (kinematics and dynamics), and evolving rules, including high-level classes and their interactions. The experimental evaluation focuses on two classic video games, Pong and Arkanoid. Analyzing a small number of raw video frames, the methodology identifies objects, classes, and rules, creating a compact, high-level, predictive description of the interactions between the elements in the videos.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Machine learning approaches**.

KEYWORDS

artificial intelligence, compositionality, core knowledge, evolutionary algorithms, rules

ACM Reference Format:
Andrea Calabrese, Stefano Quer, Giovanni Squillero, and Alberto Tonda. 2024. Towards an Evolutionary Approach for Exploting Core Knowledge in Artificial Intelligence. In *Genetic and Evolutionary Computation Conference (GECCO '24 Companion), July 14–18, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3638530.3654230

1 INTRODUCTION

Artificial Intelligence (AI) is an umbrella term that covers several different techniques, ranging from rule-based systems to Machine Learning (ML), from Reinforcement Learning (RL) to Evolutionary Algorithms (EAs). AI algorithms have recently set important milestones in a wide range of domains. DL approaches can learn predictive models directly from large data samples. Still, they suffer from essential limitations [4], such as *brittleness*, the unpredictable and undesirable behavior for given samples, *opacity*, the impossibility of explaining the overall predictions of the model, and *limited generalization ability*, the poor prediction quality for out-of-distribution samples.

Most of these issues stem from the black-box nature of the models, whose behavior cannot be verified by human experts. A few sub-domains of AI are currently exploring different possible solutions. The eXplainable AI (XAI) community [6] is developing techniques to make black-box models more human-readable, for example, using concept bottlenecks or visualization techniques able to highlight the features that ML/DL models use to make decisions. On the contrary, neural-symbolic (NeSy) approaches [1] aim to combine modern neural-network models with classic symbolic AI, capitalizing on both advantages. A different research direction, proposed in [2], is to build AI systems exploiting principles similar to human *core knowledge*, that is, a small set of innate capacities identified by cognitive psychologists. Examples of core knowledge include evaluation of quantities, identification of agents, and prediction of movement. Instead of starting from a *blank slate* informed only by the available training data, as in the case of current state-of-the-art ML, AI algorithms may begin with a limited amount of specialized hard-coded capacities, compose and combine them to solve tasks and use only a limited amount of training samples. In principle, algorithms that learn in a way that is more similar to humans could require less training data to complete tasks while providing a white-box, interpretable explanation of their behavior.

In this paper, we propose a first step towards an evolutionary AI approach inspired by *core knowledge*. Core knowledge is a psychological theory of human cognition [7] postulating a small set of innate cognitive capacities all animals are born with. The source of this core knowledge is uncertain, but it is currently believed to be the process of natural selection. All animals, however, have the potential to learn new skills or concepts through experience, and distinguishing core knowledge from learned skills is not a straightforward process. Cognitive psychologists tackled this challenge by either evaluating the performance of human groups that are culturally isolated or by focusing on toddlers and infants. Notable examples include the study of arithmetical intuition among indigenous populations of the Amazonian rainforest [3, 5], and the assessment of quantity in six-month-old infants [9]. In our approach the EA is used to create a high-level, concise description of a simple 2D video recorded from a video game. Using a small number of primitive concepts, such as patches tracked over frames and simplified laws of motion to predict their expected behavior,

*All authors contributed equally to this research.

GECCO '24 Companion, July 14–18, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0495-6/24/07.
https://doi.org/10.1145/3638530.3654230

the proposed approach can generate a compact, correct video description. Our approach first groups patches that behave coherently into objects. Groups of objects that share similar behaviors are then grouped into classes. The interactions between objects belonging to specific classes are then described through rules. In the proof of concept presented in this work, the identification of objects is performed through simple heuristics. In contrast, the association of objects to classes and the creation of rules is delegated to the EA engine. The proposed approach is evaluated on two popular benchmarks, Pong and Arkanoid, and it has been proven to explain the interactions appearing in short video-game videos, generating accurate and human-readable descriptions from a relatively small amount of data.

2 PROPOSED APPROACH

We propose a novel EA-based methodology to create the description of a simple, two-dimensional video, identifying *objects* appearing in it, clustering them into classes, and generating rules describing their interactions. The process is organized as follows.

First, we perform data analysis, gathering the positions of the segmented patches and their possible contact points in each frame. Then, we detect the interactions between patches, transforming them into objects and providing a physical status, with features such as speed and shapes identified by the patches forming that object. This step can be performed through several different approaches, but, in this first proof of concept, we opted for a strategy based on evolutionary optimization.

After detecting the different patches, the algorithm aggregates objects with a coherent behavior into *classes* and creates *rules* to describe their interactions, ultimately building a human-readable description of the actions identified in the video. An object can be either static or dynamic. Static objects can be defined using simple rules mimicking humans' idea of persistence: (a) the object exists only in one position, and (b) the object may disappear, but if it reappears, it does so in the same position. Dynamic objects, on the other hand, are clusters of patches that move through the screen.

We model concepts following basic laws of physics, allowing the approach to detect events inspired by physical interactions. For instance, in Arkanoid, the contact between two patches may be detected as an interaction, providing a cause-effect relationship between events. More specifically, some patches are recognized as the object "ball" and others as the "paddle". Then, when the ball collides with the paddle, the velocity on the y-axis of the ball changes sign, and the velocity on the x-axis changes based on the angle formed by the ball and the center of the paddle. Such fundamental interactions are the core knowledge of the system, i.e., the idea of "collision", the dynamic of a "bounce", and the appearance and disappearance of an object. Moreover, it could be noted that even a cause-effect relationship is part of the core knowledge, mimicking how humans explain the world.

More generalization becomes possible as more physical events are discovered in the video frames, as the same rules may explain several other events. Generalization can also reinforce some of the previous hypotheses and rule out others. For example, on the one hand, a video that does not include a bounce with the top "wall" may lead to an indeterminism in the classification of the wall itself. To the analysis, the wall may as well not exist or be a background object. On the other hand, if there is an interaction between the ball and the wall, the effect is discovered, and the wall is correctly classified.

Patch detection, however, presents a few technical obstacles. In particular, the most challenging issue is an aliasing problem. Our approach involves knowing the velocity of patches at each frame. However, all our data is related to pixels, which can be either *belonging* or not belonging to the patch in a binary way. Thus, analyzing differences frame-by-frame may not yield a satisfactory result, as this procedure may detect spurious accelerations due to the discretization. To overcome this difficulty, we provide knowledge based on the first law of physics, i.e., an object not subject to any forces has a constant velocity. Thus, if we assume a uniform speed between two frames, the precision of detecting the actual rate of the object increases, as does the distance between frames. Similarly, a larger distance between positions, obtained by expanding the time difference between frames, can provide a more precise average velocity. However, we shall ensure that the hypothesis is verified; thus, each time we expand the time difference, we check if the ball position on the frames in between is well-approximated.

The last step of our process is a learning phase, performed by an EA, that aggregates the objects detected by the heuristics into classes, and then describes the interactions between classes using rules. A class is a coherent aggregate of objects. A rule is a cause-effect relationship, where the cause is an interaction between two objects, and the effect is an alteration of the state of some target objects, which might or might not have been directly involved in the interaction. Classes and rules are ontologically dependent on each other, i.e., a class's existence depends on the presence, or not presence, of the interactions with other objects. For instance, the Arkanoid game's top, left, and right walls trigger a bounce in the ball; however, their state does not change in any way. Finally, some events may not be explicable from what is observed. An example is the horizontal movement of the paddle in Arkanoid, which depends on the user input and that no observer can explain with a rule.

However, according to the video analysis, the EA finds a set of classes and rules derived from the "perceived" *events*, minimizing the number of rules and entities involved in the explanation.

Each candidate solution can have an arbitrarily large number of classes and rules, with a minimum of one classe and one rule (describing the interaction between objects belonging to the same class). The fitness function evaluates the accuracy and the complexity of our explanation, with a preference for the simpler description between two approximately equally accurate solutions, with an idea similar to Occam's razor. Thus, the fitness function defined for this optimization problem is:

$$F(I) = \neg E(I) + \alpha \cdot (C(I) + R(I)) \tag{1}$$

where I is a candidate solution, $\neg E(I)$ is the number of unexplained events, $C(I)$ and $R(I)$ are the total number of classes and rules in the solution, respectively, and α is a user-defined weight regulating the relative importance of the second part of the equation. For example, setting α to a small value means that a solution with a larger amount of classes and rules able to explain a more significant portion of the events will be considered more promising than a

solution with fewer rules but unable to explain as many events. The value of $F(I)$ must be minimized.

Given the structure of a candidate solution, we define genetic operators that may modify either the set of classes or the set of rules in each explanation. For the set of classes, we use the following operations:

- Class mutation: A randomly selected object is moved between two randomly selected classes in the same explanation; if a class ends up with no objects, the class is removed from the explanation.
- Class removal: A randomly selected class is removed, and its objects are randomly distributed among other existing classes with uniform probability; all the rules in which the class is involved are discarded.
- Class addition: A new class is added to the explanation, and a randomly selected object is moved to the newly created class.

For the rules, we can apply the following operators:

- Rule mutation: An existing rule is mutated; the mutation may involve the target class, the interacting classes, the cause, and the effect; in all cases, a new randomly selected element is selected to replace the current one.
- Rule removal: A randomly selected rule is removed.
- Rule addition: A new randomly generated rule is created.

3 EXPERIMENTAL EVALUATION

We tested the proposed approach on Pong and Arkanoid (also known as Breakout), two extremely popular two-dimensional video games. These experiments aim to test our strategy on puzzles based on similar concepts but different layouts.

We tested our approach with two different videos for each game. Each video is recorded with 60 frames per second. For Pong, the first video includes about 4,318 frames, and the second includes about 6,086. For Arkanoid, the videos are 2,335 and 12,082 frames long. Moreover, each experiment was repeated 30 times with 30 different initial seeds for the random generator to check the convergence ability of our technique. Each video is part of a *training test case*. Ideally, with sufficient events in the video, we show that converging to a shared knowledge of the game is possible. All experiments are run on a server using an Intel(R) Xeon(R) Gold 6238R CPU @ 2.20GHz, equipped with 256 GB of RAM. All the code and the data necessary to reproduce the experiments are freely available on a GitHub repository at https://github.com/to-be-disclosed-after-revision.

The scene analysis focuses on identifying separate patches inside and across video frames. Many image-segmentation techniques have been proposed to detect elements in images, and the best choice is often application-specific. In the current work, we select OpenCV. The EA used in the experiments employs a classic $(\mu + \lambda)$ replacement scheme and a tournament selection for choosing the individuals for reproduction. The evolutionary library used in the experiments is `inspyred` [8]. For all the following experiments, we use a population size of $\mu = 1,000$ and an offspring size of $\lambda = 1,000$ individuals. The tournament selection employs $\tau = 2$, and the termination condition is set on 2,000 maximum generations, with an early stop if the best fitness does not improve for 100 generations.

When a new candidate solution is to be produced starting from a parent solution, a single genetic operator is selected with uniform probability among those previously described. We used the value 0.001 as α in Equation 1.

Pong involves a ball bouncing on both the walls and the two paddles on each side; each object can send the ball back. Paddles are user-controlled and can be moved vertically up and down. Each player's target is to hit the opposite wall with the ball, getting beyond the opponent's paddle and scoring a point. The ball starts with an initial velocity on the x-axis $v_x \neq 0$, and the paddles should catch the ball. Once the ball touches a paddle, it bounces. There are two variants of this game. In the first one, ball bounces are perfectly elastic; in the second one, the new angle of the ball depends on the distance from the center of the paddle. We experimented with both variants, as they deliver slightly different results.

In our video analysis, when the ball touches one of the white stripes representing the net, it disappears (meaning that the video analysis has no concept of the permanence of an object). Consequently, the algorithm finds the following rule: The ball bounces when the ball touches one of the white stripes (grouped in a class). This effect is seen in Figure 1, showing that the algorithm adds the rule for the ball's disappearance. In particular, starting from different complexity values, the algorithm can improve over the solution until it finds the final complexity of 227.006 within 20 generations at most.

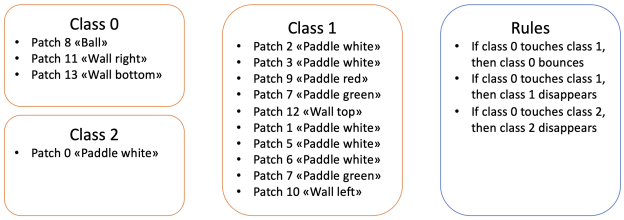

Figure 1: The Pong puzzle: The classes and rules discovered by our strategy.

Arkanoid features a single horizontal paddle controlled by the player, a set of bricks, and a ball that can bounce on walls and bricks. Each time the ball bounces against a brick, the brick disappears, and the player's score increases. The game's goal is to make all bricks disappear by hitting them with the ball; missing the ball with the paddle and letting it hit the bottom of the screen causes a game over. The paddle is user-controlled and can be moved horizontally left and right. In our case study, we have three lines of bricks, and bricks of different colors represent each line.

The ball bounces elastically against the walls and the bricks. Colliding against the paddle generates a bounce that follows the same rules we analyzed with Pong. Arkanoid is more challenging to interpret for our learning program. This is due to the higher number of patches in the frames and the more significant number of rules required to interpret all events. In particular, our instance of Arkanoid can be described using 26 patches, whereas Pong requires only 3 (without the net in the middle) or 10 (with the net) patches.

As shown in Figure 2, the learner can separate the ball from the other patches by finding the classes "ball", "blocks", and "misc". However, most of the blocks belonging to the miscellaneous class

are artifacts. Indeed, they disappear when the ball touches them, even if it never bounces back. Moreover, as the bottom wall is never touched in this game, the learning algorithm has no experience with its interactions and places it in a random class.

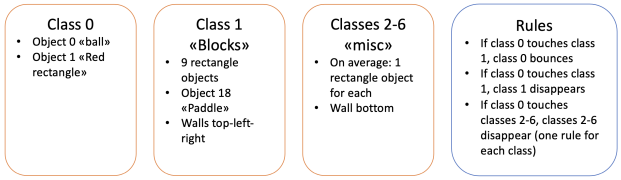

Figure 2: The Arkanoid puzzle: The classes and rules discovered by our strategy.

This behavior is similar to the one we discovered analyzing Pong. In both games, we converge toward a very good set of rules, perfectly describing the basic principles of the games. For Arkanoid, such rules can be described as "the ball bounces", "all blocks share the same behavior", and "each block disappears when hit by the ball".

To sum up, we analyze two games with several similarities and our solutions converge as expected on both game videos, leading to a unique and reasonable explanation. Table 1 shows that the algorithm converges to the same value even using very different starting seeds. In the table, seeds are selected randomly. The first column shows the seed we used. Then, for each game, the first column reports the fitness of the best individual after the first generation, and the second one presents the best individual in the last generation. However, the number of generations needed to converge varies significantly. For instance, for the Pong game our approach may require more than 40 generations to converge to the solution with fitness shown in the fifth column of Table 1. For Arkanoid, the convergence requires more than 70 generations, with minor differences in the convergence rate depending on the initial random seed used to find the solution. By exploring the solution space, we noticed a typical pattern: Our approach tries to minimize the number of classes despite the number of rules involved. When an object is moved between classes, the number of rules increases, creating false explanations. However, as one of our targets is to minimize the number of rules, the ones that do not provide any proof are automatically discarded after three or four generations from the generation in which they are introduced.

4 CONCLUSIONS

This work is the first step in a research line to build AI systems inspired by core knowledge and exploiting evolutionary computation to aggregate the basic information provided by hard-coded algorithms. We present an approach to explain a 2D video as a set of objects, classes, and rules. We apply it to two simple but widespread games, Pong and Arkanoid. The approach provides a human and machine-readable description of the videos and a resulting explanation for their models. We show that, even with a limited training set, the process can differentiate classes and abstract rules involving them. Thus, the approach builds a believable explanation of the rules followed by the various classes.

	Pong Evolution [Generations]		Arkanoid Evolution [Generations]	
Seed	**First**	**Last**	**First**	**Last**
123456	237.006	227.006	171.020	150.014
42	238.007	227.006	167.005	150.014
256	236.007	227.006	157.007	150.014
190283	236.007	227.006	160.005	150.014
328	239.005	227.006	159.005	150.014
715321	234.005	227.006	161.007	150.014
0	228.005	227.006	171.005	150.014
10000	235.005	227.006	161.004	150.014
444	235.007	227.006	161.007	150.014
711	234.005	227.006	165.007	150.014
8125002	235.007	227.006	165.004	150.014
30	234.008	227.006	164.004	150.014
59	235.007	227.006	164.006	150.014
100	239.006	227.006	157.007	150.014
99	233.006	227.006	166.004	150.014
999	234.004	227.006	160.004	150.014
999999	229.004	227.006	159.008	150.014
13	235.008	227.006	167.006	150.014
75	237.007	227.006	159.007	150.014
915	235.004	227.006	164.004	150.014
627	233.006	227.006	157.006	150.014
498	242.005	227.006	166.006	150.014
186	236.004	227.006	168.006	150.014
216	234.006	227.006	161.006	150.014
311	235.008	227.006	172.006	150.014
618	234.008	227.006	164.008	150.014

Table 1: The evolution of Pong and Arkanoid games respectively: for each seed, we show the fitness of the best individual in the first generation and the fitness of the best individual in the last generation.

REFERENCES

[1] Tarek R. Besold, Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luis C. Lamb, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, and Gerson Zaverucha. 2021. Neural-Symbolic Learning and Reasoning: A Survey and Interpretation1. In *Frontiers in Artificial Intelligence and Applications*. IOS Press. https://doi.org/10.3233/faia210348

[2] François Chollet. 2019. On the Measure of Intelligence. https://doi.org/10.48550/ARXIV.1911.01547

[3] Stanislas Dehaene, Véronique Izard, Elizabeth Spelke, and Pierre Pica. 2008. Log or Linear? Distinct Intuitions of the Number Scale in Western and Amazonian Indigene Cultures. *Science* 320, 5880 (May 2008), 1217–1220. https://doi.org/10.1126/science.1156540

[4] Gary Marcus. 2020. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. https://doi.org/10.48550/ARXIV.2002.06177

[5] Pierre Pica, Cathy Lemer, Véronique Izard, and Stanislas Dehaene. 2004. Exact and Approximate Arithmetic in an Amazonian Indigene Group. *Science* 306, 5695 (Oct. 2004), 499–503. https://doi.org/10.1126/science.1102085

[6] Waddah Saeed and Christian Omlin. 2023. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems* 263 (March 2023), 110273. https://doi.org/10.1016/j.knosys.2023.110273

[7] Elizabeth S. Spelke and Katherine D. Kinzler. 2007. Core knowledge. *Developmental Science* 10, 1 (Jan. 2007), 89–96. https://doi.org/10.1111/j.1467-7687.2007.00569.x

[8] Alberto Tonda. 2020. Inspyred: Bio-inspired algorithms in Python. *Genetic Programming and Evolvable Machines* 21, 1 (2020), 269–272. https://doi.org/10.1007/s10710-019-09367-z

[9] Fei Xu and Elizabeth S. Spelke. 2000. Large number discrimination in 6-month-old infants. *Cognition* 74, 1 (Jan. 2000), B1–B11. https://doi.org/10.1016/s0010-0277(99)00066-9