OCAtari: Object-Centric Atari 2600 Reinforcement Learning Environments

Quentin Delfosse[*,1,2], **Jannis Blüml**[*,1,3], **Bjarne Gregori**[1],
Sebastian Sztwiertnia[1] & **Kristian Kersting**[1,3,4,5]
quentin.delfosse@cs.tu-darmstadt.de, blueml@cs.tu-darmstadt.de

[1]AI and ML Group, Technical University of Darmstadt, Germany
[2]National Research Center for Applied Cybersecurity (Athene)
[3]Hessian Center for Artificial Intelligence (hessian.AI)
[4]Centre for Cognitive Science of Darmstadt
[5]German Research Center for Artificial Intelligence (DFKI)

Abstract

Cognitive science and psychology suggest that object-centric representations of complex scenes are a promising step towards enabling efficient abstract reasoning from low-level perceptual features. Yet, most deep reinforcement learning approaches only rely on pixel-based representations that do not capture the compositional properties of natural scenes. For this, we need environments and datasets that allow us to work and evaluate object-centric approaches. In our work, we extend the Atari Learning Environments, the most-used evaluation framework for deep RL approaches, by introducing OCAtari, that performs resource-efficient extractions of the object-centric states for these games. Our framework allows for object discovery, object representation learning, as well as object-centric RL. We evaluate OCAtari's detection capabilities and resource efficiency. Our source code is available at `github.com/k4ntz/OC_Atari` .

1 Introduction

Since the introduction of the Arcade Learning Environment (ALE) by Bellemare et al. (2013), Atari 2600 games have become the most common environments to test and evaluate RL algorithms (*cf.* Figure 1, left). As RL methods are challenging to evaluate, compare and reproduce, benchmarks need to encompass a variety tasks and challenges to allow for balancing advantages and drawbacks of the different approaches (Henderson et al., 2018; Pineau et al., 2021). ALE games incorporate many RL challenges, such as difficult credit assignment (Skiing), sparse reward (Montezuma's Revenge, Pitfall), and allow for testing approaches with different focuses, such as partial observability (Hausknecht & Stone, 2015), generalization (Farebrother et al., 2018), sample efficiency (Espeholt et al., 2018), environment modeling (Hafner et al., 2021; Schrittwieser et al., 2020), ... etc.

In order to solve complex tasks, human use *abstraction*, *i.e.* they first extract object-centred representations and abstract relational concepts, on which they base their reasoning (Grill-Spector & Kanwisher, 2005; Tenenbaum et al., 2011; Lake et al., 2017). Deep reinforcement learning (RL) agents do not incorporate explicit object-centric intermediate representations, necessary to check if suboptimal behaviors are *e.g.* caused by misdetections, wrong object identifications, or a reasoning failure. Numerous studies on RL highlight the importance of object-centricity (*cf.* Figure 1, right), notably in understanding the agents' reasoning, detect potential misalignment and potentially correct

[*]Equal contribution

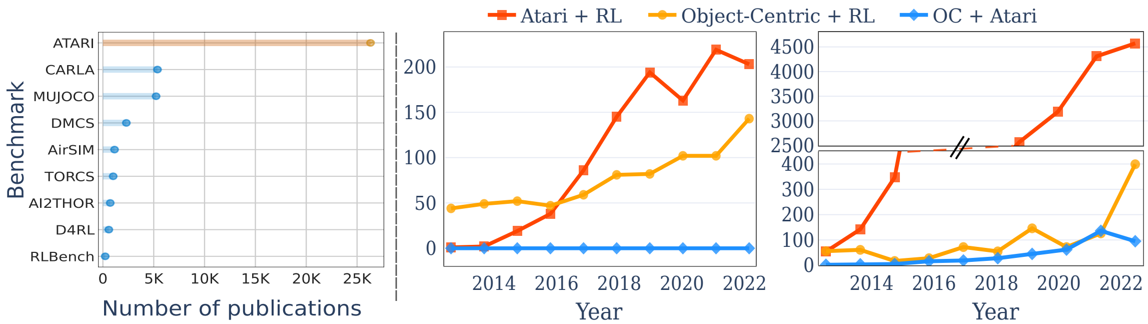


Figure 1: **RL research needs Object-Centric Atari environments.** The Atari Learning Environments (ALE) is, by far, the most used RL benchmark among the ones listed on `paperswithcode.com` (left). Publications using ALE are increasing, together with the number of papers concerned on object-centric RL. As no Object-centric ALE is available yet, the amount of papers concerned with object-centric approaches in Atari is however negligible. Data queried using `dimensions.ai`, based on keyword occurrence in title and abstract (center) or in full text (right). These graphs show that RL researchers would make use of object-centric atari environments, if they would be available.

it (di Langosco et al., 2022). Notably, Delfosse et al. (2024) show that deep agents, that do not make use of interpretable object centric representations, can learn misaligned policies on games as simple as Pong, that post-hoc explanation techniques cannot detect. They later propose to instead make use of decision trees (Dalal et al., 2021; Yan et al., 2024; Fuhrer et al., 2024), convertible into python code (Kohler et al., 2024). Object-centricity also permits to use logic to encode the policy, leading to interpretable agents with better generalization capability (Delfosse et al., 2023b), and ease knowledge transfer between humans and learning agents, or among different tasks (Dubey et al., 2018). Further studies also underline that the extraction of object-centric states is a necessary step to obtain agent that can make use of large language model together with contextual data (*e.g.* the games instruction manuals) to improve the reward signals, notably allowing agents to learn in difficult credit assignment environments (Zhong et al., 2021; Wu et al., 2023). This underscores the need to produce transparent object-centric RL agents, that can ensure their proper alignment with the intended objectives.

Lake et al. (2017) illustrated that deep agents trained on ALE games lack the ability to create multi-step sub-goals (such as acquiring certain objects while avoiding others) and introduced the "Frostbite challenge" to assess that RL agents integrate such human-like capabilities. Badia et al. (2020) also suggested to enhance the internal representations of suboptimal ALE trained agents.

As no benchmark to test object-centric methods exists yet, we introduce OCAtari, a set of object-centric versions of the ALE environments. OCAtari runs the ALE games while maintaining object-centric states (*i.e.* a list of the depicted objects and their properties). Our framework can be used to train object-centric RL algorithms, making it a resource-efficient replacement for otherwise necessary object discovery methods. For these, we also propose the Object-centric Dataset for Atari (ODA), that uses OCAtari to generate a set of Atari frames, together with the properties of the objects present in each game. Our contributions can be summarized as follows:

- We introduce OCAtari, an RL framework to train and evaluate object-detection and object-centered RL methods on the widely-used Arcade Learning Environments.
- We evaluate OCAtari capability to detect the depicted game objects in a resource efficient way and demonstrate that it allows for object-centric RL.
- To ease the comparison of object-discovery methods, we introduce ODA, a collection of frames from Atari games together with their object-centric states.

We start off by introducing the Object-Centric Atari framework. We experimentally evaluate its detection and speed performances. Before concluding, we touch upon related work.

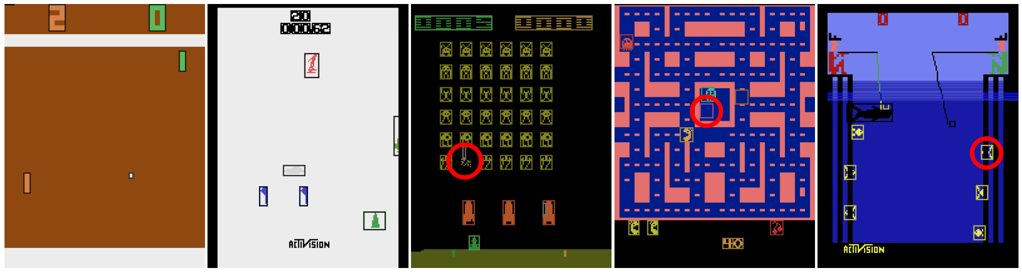


Figure 2: **Qualitative evaluation of OCAtari's REM.** Frames from our OCAtari framework on 5 environments (Pong, Skiing, SpaceInvaders, MsPacman, FishingDerby). Bounding boxes surround the detected objects. REM automatically detects blinking (MsPacman), occluded (FishingDerby) objects, and ignore *e.g.* exploded objects (SpaceInvaders) that vision methods falsely can pick up.

2 The Object-Centric Atari Environments

In this section, we discuss the definition of objects and how they can be used in RL, then introduce the OCAtari benchmark, and detail its two extraction methods.

2.1 Using Object-Centric Descriptions to Learn

According to Thiel (2011), objects are physical entities that possess properties, attributes, and behaviors that can be observed, measured, and described. Rettler & Bailey (2017) define objects as the fundamental building blocks that human reasoning relies on. Breaking down the world into objects enables abstraction, generalization, cognitive efficiency, understanding of cause and effect, clear communication, logical inference, and more (Spelke et al. (1992); Grill-Spector & Kanwisher (2005); Tenenbaum et al. (2011); Lake et al. (2017), *cf.* Appendix B for further details).

In artificial approaches, object-centric visual learning often involves the extraction of objects withing bounding boxes that contain them and distinguish them from the background (Lin et al., 2020b; Delfosse et al., 2022). In these approaches, static objects, such as the maze in MsPacman or the walls in Pong (*cf.* Figure 2), are considered as part of the background. In our work, we define *objects* as small elements (relative to the agent) with which it can interact. Excluding "background objects" when learning to play Pong with object-centric inputs is not problematic. However, it can lead to problems when learning on *e.g.* MsPacman. The learning agents can learn to incorporate *e.g.* Pong's boundaries when learning to play, but may have difficulties to accurately encode the maze structures of Pacman games. As it may be necessary to provide a background representation to the agent, OCAtari provides both renderings and object-centric descriptions of the states.

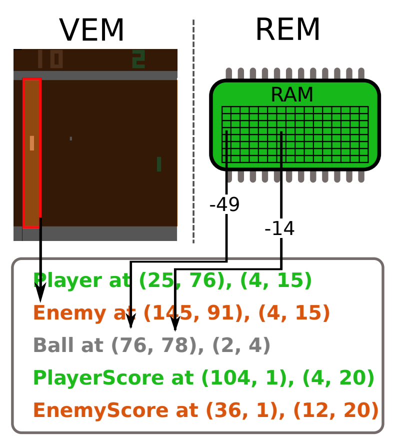


Figure 3: **OCAtari extract object-centric descriptions**: using its RAM Extraction method (REM) or Vision Extraction method (VEM).

2.2 The OCAtari framework

In OCAtari, every object is defined by its category (*e.g.* "Pacman"), position (`x` and `y`), size (`w` and `h`), and its `RGB` values. Objects may have additional characteristics such as orientation (*e.g.* the Player in Skiing, *cf.* Figure 2) or value (*e.g.* oxygen bars or scores) if required. Objects that are vital for gameplay are distinguished from those that are components of the Head-up-Display (HUD) elements (*e.g.* score, number of lives). The role of HUD objects is to provide additional information about the performance of the playing agent. Although learning agents should, in principle, be capable of ignoring such elements, in our environments a boolean parameter is available to filter out HUD objects. A list of the considered objects for each game can be found in Appendix G.

To extract objects, OCAtari uses either its Vision Extraction Method (VEM) or its resource efficient RAM Extraction Method (REM), that are depicted in Figure 3.

VEM: the Vision Extraction Method. The most straightforward method for extracting objects from Atari frames involves using simple computer vision techniques. Considering the limited memory available to Atari developers, most objects are defined by a restricted set of pre-established colors (i.e., RGB values). At each stage, the Vision Extraction Method extracts objects using color-based filtering and priors about the objects' positions. For example, Pong consists of 3 moving objects and 2 HUD objects, each assigned fixed RGB values (*cf.* Figure 3). The enemy's paddle and scores share common RGB values (orange in Figure 3), but contrary to the scores, the paddles always appears between the white threshold. The enemy's paddle is always positioned within the red rectangle. Using this technique, it is possible to accurately extract all present objects. This detection method can only detect what is depicted in the frame, and not objects that are *e.g.* blinking, overlapping, etc.

REM: the RAM Extraction Method. ALE provides the state of the emulator's RAM, which contains information about the games' objects. This has led Sygnowski & Michalewski (2016) to use the raw RAM states for RL states to train agents. However, much of the non-relevant information is present in the RAM (*e.g.* time counter, HUD element information). Moreover, several games, use *e.g.* bitmaps or encode various information quantities such as object orientation, offset from the anchor, and object category together within one byte. These noisy inputs and entangled representations prevent obscure these agents decision process and remove any interpretation possibilities. To address these problems, Anand et al. (2019) have proposed AtariARI, a wrapper around some Atari environments, that provides some the RAM positions, describing where some specific information is encoded. Nonetheless, raw RAM information is not enough. Take, for instance, in *Kangaroo*, the player's position corresponds to various RAM values, that also encode its heights using categorical values. Simply providing some uninfluenced RAM positions does not reflect the object-centric state. Similar to AtariARI, our Ram Extraction Method extracts the information from the RAM, but processes it to provide an interpretable object-centric state, that matches VEM's one (*cf.* Figure 3). To determine how the game's program processes the RAM information, we task human, random, or DQN agents with playing the games while using VEM to track the depicted objects. We then establish correlations between objects properties (*e.g.* positions) and each of the 128 bytes of the Atari RAM representation. We can also modify each RAM byte and track the resulting changes in the rendered frames. All these scripts are documented and released along with this manuscript.

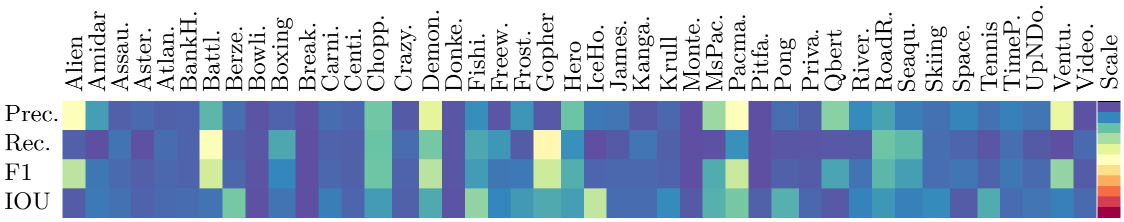
REM, being based on semantic information, allows for tracking moving objects. Conversely, VEM only furnishes consecutive object-centric descriptions, where the lists of objects are independently extracted at each state. REM thus enables tracking of blinking objects and moving instances, as proven useful for RL approaches using tracklets (Agnew & Domingos, 2020; Liu et al., 2021).

The OCAtari package. We provide an easy-to-use `ocatari` package, with its documentation[1]. OCAtari includes wrappers for the Arcade Learning Environments (ALE) of Bellemare et al. (2013). To allow an easy swap between ALE and OCAtari environments, we follow the logic and naming system of ALE. We have reimplemented its methods for OCAtari (*e.g.* `step`, `render`, `seed`, ...), and added new methods like `get_ram` and `set_ram`, to easily allow RAM lookup and manipulation. `OCAtari` environments also maintain a list of the depicted objects and can provide a buffer of the last 4 transformed (*i.e.* black and white, 84×84) frames of the game, as it has become a standard of RL state representations (Mnih et al., 2015; van Hasselt et al., 2016; Hessel et al., 2018).

As shown in Table 3, our image processing method VEM covers 46 games, while REM covers 44 games at the time of writing. While these already constitute a diverse set of environments, we will continue to add newly supported games in both REM and VEM and complete what we have started. Along with this work, we release ODA, a dataset that contains frames with the object-centric states obtained from REM and VEM, collected using Random and trained DQN agents (*cf.* Appendix A for further details). ODA and OCAtari are openly accessible under the MIT license.

[1] https://oc-atari.readthedocs.io

Table 1: **REM reliably detects the objects within the frames of each developed games**. Measuring precision, Recall, F1-Score and IOU of REM (using VEM as baseline) in a diverse set of Atari games using trained DQN agents. High values being displayed in blue going over green to red for low values. A more detailed table, with Radom and C51 agents is provided in Appendix G.



3 Evaluating OCAtari

In this section, we evaluate the detection and speed performances of OCAtari methods, then explain how it can be used for object-centric RL agents training. Finally, we compare OCAtari to AtariARI.

Setup. To evaluate the detection capabilities of REM, we use a random agent (that represents any untrained RL agent), as well as a DQN and, if available, a C51 agent (Bellemare et al., 2017), both obtained from Gogianu et al. (2022)[2]. For reproducibility, every used agent is provided with our along with our codebase. The RL experiments utilized the PPO implementation from stable-baselines3 (Raffin et al., 2021) on a 40 GB DGX A100 server. In each seeded run, 1 critic and 8 actors are utilized per seed over 20M frames. Since these experiments do not involve visual representation learning, we utilize the default 2×64 MLP architecture (with the hyperbolic tangent as activation functions). As developing RL agents is not our focus, we did not conduct any fine-tuning or hyperparameter search. Further details on these experiments can be found in Appendix E.

3.1 Evaluating OCAtari for Object Extraction

Correctness and Completeness of the Object Extraction. As explained previously, REM needs to decode the game objects' properties from RAM values. For example, objects' position in *e.g.* Riverraid either require adding an offset (for the agent) or being reconstructed from anchor and offsets position in a grid. To assert that REM accurately reconstruct these values, we compare the object-centric states of both extraction methods (VEM and REM). We let the Random, and trained DQN and C51 agents play for 500 frames, and compute IOU (Rezatofighi et al., 2019) for each agent on each game. As this metric's relevance is debatable for small objects (*e.g.* the ball in Pong, Tennis, or missiles in Atlantis, Space Invaders), we also calculate precision, recall, and F1-scores for each object category in every game. For these metrics, an object is considered correctly detected if it is within 5 pixels of the center for both detection methods.

In Table 1, we report these metrics for DQN agents averaged over every object category. Similar results, obtained using Random and C51 agents are provided in Appendix G. Lower precisions indicate that some objects detected using REM are not detected by VEM, and lower recalls imply the opposite situation. In MsPacman, the ghost can blink and objects can overlap, which explains why the precision is slightly lower. This can be observed in the per-category tables (*cf.* Appendix G). We opted for allowing the RAM extraction method to monitor hidden or blinking objects, regardless of its effects on the precision of our framework, as it can be used to train object tracking methods that employ tracklets (e.g., Agnew & Domingos 2020) or Kalman filters (e.g., Welch et al. 1995). The F1-score aggregates both previously mentioned metrics, using a harmonic mean, hastily punishing both for false positives and false negatives. Perfect F1-scores means that every object-centered state extracted using REM is identical to the VEM ones.

[2] `https://github.com/floringogianu/atari-agents`

Game	SPACE	SPOC	REM
Boxing	24.5	70.5	**90.1**
Carnival	48.6	90.6	**93.7**
MsPacm.	0.4	**90.5**	87.4
Pong	10.7	87.4	**94.3**
Riverraid	45.0	76.6	**95.7**
SpaceInv.	87.5	85.2	**96.9**
Tennis	3.6	40.2	**99.3**
Average	31.5	77.3	**93.9**

Table 2: **Object detection is still challenging in Atari.** SPACE and SPOC, SOTA in object discovery, are inferior in terms of F1 scores.

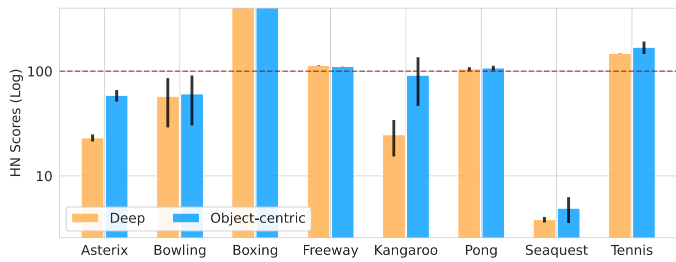


Figure 4: **OCAtari (REM) permits learning of object-centric RL agents**. The object-centric PPO agents perform at least on par with the pixel-based PPO (Deep) agents' and humans on 8 Atari games.

In general, the table results indicate that the games covered by REM have high detection performances. Misdetection can be caused by overlap of other objects or the background (*cf.* Figure 2, *FishingDerby*). Potential rendering instabilities cause slight differences in ball position and size, which decreases the IOU in *e.g.* Pong and Tennis. In many games, the rendering freezes after specific events (*e.g.* when the player dies) while the RAM is unaltered. Some objects are then not rendered for a few frames, but our RAM extraction approach can keeps them in the list. Although this decreases the detection scores, it does not affect gameplay since, for these frames, the environment is not interactive.

In Table 2, we compare the detection performances (F1-scores) of REM (94%) on the games used by the 2 object-discovery methods used on ALE: SPACE (Lin et al. (2020a), 31%) and SPOC (Delfosse et al. (2022), 77%). REM largely outperforms both. As highlighted by SPOC's authors, the detection of Atari games' objects, composed of few pixels, remains a challenge for neural networks. OCAtari does not extract encodings for objects, but directly provides their classes (from the deterministic RAM information process), that can be used to train these objects discovery methods.

Comparing the RAM and Visual Extraction Method. As explained in the previous section, REM relies on accurate information decoding, but allows for tracking blinking or overlapping objects. Its most significant advantage over VEM is the computational efficiency of the RAM extraction procedure. While VEM must perform colour filtering for each object category, REM needs few simple operations to extract objects' properties. Getting object-centric states using REM is on average 50 times faster than with VEM (*cf.* Figure 7 in Appendix K). RL agents can use REM to efficiently train the reasoning part of the policy, as shown bellow, and later be fine-tuned to work with neural-based object extraction. To evaluate such extraction methods, on *e.g.* independently drawn frames (without tracking), VEM can reliably extract only the visible objects. The (slower) extraction is then performed only once, as such training is usually run using a dataset, such as ODA.

3.2 Using OCAtari to train Object-centric RL agents

To show that OCAtari allows training object-centric RL agents, we trained RL agents using our REM with 3 seeded Proximal Policy Optimization (PPO) agents in 8 different environments. These agents are provided with the positional information of each moving object. Specifically, these correspond to the `x` and `y` positions of each object detected by REM in the last two frames at each timestep. Our trained models are available in our public repository, as well as our the scripts used to generate our data sets (*cf.* Appendix A). As depicted in Figure 4, OCAtari allows object-centric PPO agents to learn to master Atari games, as they perform on par or better than their deep counterparts.

Overall, we have shown that OCAtari can be used to train or evaluate any part of an object-centric RL agent, from object extractors (preferably with VEM) to object-centric policies. Since REM allows object tracking, it can also be used on methods that track object through time, and can directly be integrated for resource efficient object-centric policy training.

Table 3: **Games supported by AtariARI and OCAtari.** ✓ describes that all necessary information about the objects are given. ∼ denotes that some necessary information to play the game is lacking. We provide detailed explanation for each of these games in Appendix J. All games missing in this table are neither supported by AtariARI nor OCAtari yet.

Extraction Method	Adventure	Alien	Amidar	Assault	Asterix	Asteroids	Atlantis	BattleZone	BankHeist	BeamR.	Berzerk	Bowling	Boxing	Breakout	Carnival	Centipede	ChopperC.	CrazyC.	DemonA.	DonkeyK.	DoubleD.	Enduro.	FishingD.	Freeway	Frostbite	Galaxiab	Gopher	Hero	IceHockey	Jamesbond	Kangaroo	Krull	Montezum.	Ms.Pacman	Pacman	Pitfall	Pong	PrivateE.	Q*Bert	RiverRaid	RoadR.	Seaquest	Skiing	SpaceInv.	Tennis	TimePilot	UpNDown	Venture	VideoP.	YarsR.	Total
ARI						✓					✓	✓	✓	✓					∼					✓	✓			∼					✓	✓		✓	✓	✓	∼	∼		∼		∼	✓			✓	✓	✓	16(22)
REM	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	∼	46(47)
VEM	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	48

3.3 OCAtari vs AtariARI

For their AtariARI framework, Anand et al. (2019) disassembled the source code of various games to find the RAM location of the objects' properties. AtariARI thus provides information of where a specific information is encoded in the RAM. Providing only the RAM positions is however not enough to get a directly human-interpretable, object-centric description of the state. As shown in Figure 3, even when positions are encoded directly, offsets are applied to objects during the rendering phase, which the raw RAM information does not provide. Some games the information provided by AtariARI is thus incomplete or insufficient to play the game (*cf.* Table 3 and Appendix J). Our OCAtari framework makes use of intricate computations, such as deriving the `x` and `y` positions from grid anchors and offsets, looking up potential presence indicators (e.g. for objects that have been destroyed). This ensures that RL agents genuinely acquire human understandable object-centric state descriptions, on which they can base their policies. Finally, OCAtari is already covering (28) more games than AtariARI, and we are continually adapting the rest of the game collection of ALE.

4 Related Work

Atari games to benchmark deep RL agents has a well-established history. Mnih et al. (2015) introduced the direct use of frames with DQN, tested on 7 different games of ALE. In the following years, Atari games was repeatedly used as a test bed for various approaches, well-known ones being Rainbow (Hessel et al., 2018), Dreamer (Hafner et al., 2020), MuZero (Schrittwieser et al., 2020), Agent57 (Badia et al., 2020) or GDI (Fan et al., 2021). Although deep RL agent already achieve superhuman performance on Atari games, lots of challenges are left, like efficient exploration (Bellemare et al., 2016; Ecoffet et al., 2019; 2021), efficiency (Kapturowski et al., 2022), planning with sparse (Hafner et al., 2020; Schrittwieser et al., 2020), sample inefficiency, missgeneralization (Zambaldi et al., 2019; Mambelli et al., 2022; Stanić et al., 2022), etc. As underlined by Toromanoff et al. (2019), these challenges can greatly benefit or might even require human like reasoning, and thus, object-centricity.

Other work have highlighted the need for augmented Atari benchmarks. Toromanoff et al. (2019) and Fan (2021) have both proposed to integrate many additional metrics to accurately measure performance, and Machado et al. (2018) insisting on integrating the learning efficiency. This was tackled by Kaiser et al. (2020), with their Atari 100*k* benchmark. Aitchison et al. (2023) have selected representative subsets of 5 ALE environments, by looking at the performance variances of commonly used agents. Shao et al. (2022) introduced a partial observable Atari benchmark, called Mask Atari, designed to test specifically POMDPs. These extensions can easily integrate OCAtari environments, as they can be swapped with ALE ones. Many other object-centic representations learning methods, that tackle these challenges, have also been explored outside of RL (Eslami et al., 2016; Kosiorek et al., 2018; Jiang & Luo, 2019; Greff et al., 2019; Engelcke et al., 2020; Locatello et al., 2020; Kipf et al., 2022; Elsayed et al., 2022; Singh et al., 2022a;b). Dittadi et al. (2022); Yoon et al. (2023) look at objects properties' extractions, and generalization, required for downstream tasks, while Lin et al. (2020b) and (Delfosse et al., 2022) already rely on ALE to evaluate representation learning.

Finally, several object-centric RL environments have been developed, such as VirtualHome (Puig et al., 2018), AI2-THOR (Kolve et al., 2017) or iGibson 2.0 (Li et al., 2021). While these benchmarks excel in providing realistic 3D environments conducive to AI research, they introduce high-dimensional observations and emphasizes physical interactions, particularly suitable for robotics-oriented studies.

5 Discussion

OCAtari environments are suitable for training object detection and tracking methods, as well as developing new object-centric RL approaches. OCAtari offers an information bottleneck in the form of a list of objects and their properties. While ALE is one of the most recognized benchmarks in RL, these evaluations are not without flaws, as explained by Agarwal et al. (2021). The noisy scores do not linearly reflect the agents' learning ability. These games are also created to be played by humans and offer many shortcut learning possibilities (Delfosse et al., 2024; Kohler et al., 2024). Directly evaluating the representations performance helps to understand and measure the quality of the learned internal representation and minimize other effects within the training, as proposed by Stooke et al. (2021). The object-centricity offered by OCAtari also allows to provide extra information to the algorithms, such as additional reward signal based on objects properties or relations, as done by Wu et al. (2023). Integrating object-centric detection in interpretable decisions pipeline is emerging also outside of RL (Wüst et al., 2024). Finally, our provided repository includes many scripts for locating and analyzing RAM representation information, that can be extended to create novel modifications on the Atari Learning Environments, as done in HackAtari (Delfosse et al.), that allow to find misalignment and spurious correlations, also investigated by Busch et al. (2024).

Societal and environmental impact. This work introduces a set of RL games. Such environments can be used for training object-tracking algorithms, which present potential ethical risks if misapplied. However, its main impact lies in advancing transparent object-centric RL methods, which can enhance the understanding of upcoming agents' decision-making processes and reduce misalignment issues (Friedrich et al., 2022). Improving transparency can also potentially help uncovering existing biases in learning algorithms with possible negative societal consequences (Schramowski et al., 2020; Steinmann et al., 2023). OCAtari can also save resources while training RL policies. We do not incorporate and have not found any personal or offensive content in our framework.

Limitations. OCAtari extracts object-centric representations of ALE games. In most games, there are hardcoded static elements, which we did not consider as objects. For instance, no information about the mazes in Pacaman and MsPacman are encoded in the RAM. As such, we cannot extract this information, or only partially. We could in the future decide to hardcode suitable representations for it, but we have not found one yet. However, this information being static, it could be learned by agents, but the integration of such information as input can help agents understand that *e.g.* they cannot move through it. An interesting consideration here would be whether a combination of our two modes, object-centric states and frames, can be used to extract not only objects but also important information from the backgrounds. Using this additional information like the position of objects in MsPacman to run A* or planning methods (Singh et al., 2024) can be an interesting way forward.

6 Conclusion

Representing scenes in terms of objects and their relations is a crucial human ability. While object-centric reinforcement learning and unsupervised detection algorithms are increasingly successful, we lack benchmarks and datasets to evaluate and compare such methods. OCAtari fills this gap and provides an easy-to-use diverse set of environments to develop and test object-centric learning methods on many games of ALE, by far the most commonly used RL benchmark. Overall, we hope that our work inspires other researchers to create object-centric approaches, allowing for more interpretable algorithms, for humans to interact with, and to maybe correct and learn from in the future. OCAtari will also permit AI practitioners to create novel challenges among the existing Atari games, usable on object-centric, deep or hybrid approaches.

Acknowledgments

We would like to express our sincere gratitude to our students Felicia Benkert, Kévin-Lâm Quesnel, Simon Wulf, Christopher Schubert, Sebastian Gräfe, Timo Holst and Belal Alkufrifor for their dedication and feedback. Their enthusiasm and perseverance have been instrumental in creating the environments presented in this paper. Thank you for your continuous efforts and commitment.

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".

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 34, 2021.

William Agnew and Pedro M. Domingos. Relevance-guided modeling of object dynamics for reinforcement learning. *arXiv: Learning*, 2020.

Matthew Aitchison, Penny Sweetser, and Marcus Hutter. Atari-5: Distilling the arcade learning environment down to five games. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML*, Proceedings of Machine Learning Research, 2023.

Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R. Devon Hjelm. Unsupervised state representation learning in atari. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, 2020.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *J. Artif. Intell. Res.*, 2013.

Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. Unifying count-based exploration and intrinsic motivation. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*. PMLR, 2017.

Florian Peter Busch, Roshni Kamath, Rupert Mitchell, Wolfgang Stammer, Kristian Kersting, and Martin Mundt. Where is the truth? the risk of getting confounded in a continual world. *arXiv*, 2024.

Gal Dalal, Assaf Hallak, Steven Dalton, Shie Mannor, Gal Chechik, et al. Improve agents without retraining: Parallel tree search with off-policy correction. *Advances in Neural Information Processing Systems*, 2021.

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

Quentin Delfosse, Wolfgang Stammer, Thomas Rothenbacher, Dwarak Vittal, and Kristian Kersting. Boosting object representation learning via motion and object continuity. *CoRR*, 2022.

Quentin Delfosse, Patrick Schramowski, Martin Mundt, Alejandro Molina, and Kristian Kersting. Adaptive rational activations to boost deep reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2023a.

Quentin Delfosse, Hikaru Shindo, Devendra Singh Dhami, and Kristian Kersting. Interpretable and explainable logical policies via neurally guided symbolic abstraction. 2023b.

Quentin Delfosse, Sebastian Sztwiertnia, Wolfgang Stammer, Mark Rothermel, and Kristian Kersting. Interpretable concept bottlenecks to align reinforcement learning agents. *ArXiv*, 2024.

Lauro Langosco di Langosco, Jack Koch, Lee D. Sharkey, Jacob Pfau, and David Krueger. Goal misgeneralization in deep reinforcement learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML*, 2022.

Andrea Dittadi, Samuele S. Papa, Michele De Vita, Bernhard Schölkopf, Ole Winther, and Francesco Locatello. Generalization and robustness implications in object-centric learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*. PMLR, 2022.

Rachit Dubey, Pulkit Agrawal, Deepak Pathak, Thomas L. Griffiths, and Alexei A. Efros. Investigating human priors for playing video games. In *ICML*, 2018.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. 2019.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. First return, then explore. *Nat.*, 2021.

Gamaleldin F. Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C. Mozer, and Thomas Kipf. Savi++: Towards end-to-end object-centric learning from real-world videos. 2022.

Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. GENESIS: generative scene inference and sampling with object-centric latent representations. In *International Conference on Learning Representations*, 2020.

S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Annual Conference on Neural Information Processing Systems*, 2016.

Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: scalable distributed deep-rl with importance weighted actor-learner architectures. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 2018.

Jiajun Fan. A review for deep reinforcement learning in atari: Benchmarks, challenges, and solutions. 2021.

Jiajun Fan, Changnan Xiao, and Yue Huang. GDI: rethinking what makes reinforcement learning different from supervised learning. 2021.

Jesse Farebrother, Marlos C. Machado, and Michael Bowling. Generalization and regularization in DQN. *CoRR*, 2018.

Felix Friedrich, Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. A typology for exploring the mitigation of shortcut behaviour. *Nature Machine Intelligence*, 2022.

Benjamin Fuhrer, Chen Tessler, and Gal Dalal. Gradient boosting reinforcement learning. In *Workshop on Interpretable Policies in Reinforcement Learning@ RLC-2024*, 2024.

Florin Gogianu, Tudor Berariu, Lucian Bușoniu, and Elena Burceanu. Atari agents, 2022. URL https://github.com/floringogianu/atari-agents.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew M. Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *International Conference on Machine Learning*, 2019.

Kalanit Grill-Spector and Nancy Kanwisher. Visual recognition: As soon as you know it is there, you know what it is. *Psychological Science*, 2005.

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *9th International Conference on Learning Representations, ICLR*, 2021.

Matthew J. Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposia, Arlington, Virginia, USA*. AAAI Press, 2015.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3207–3214, 2018.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 2022.

Zhengyao Jiang and Shan Luo. Neural logic reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *International Conference on Machine Learning*, 2019.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for atari. In *8th International Conference on Learning Representations, ICLR, 2020*, 2020.

Steven Kapturowski, Víctor Campos, Ray Jiang, Nemanja Rakicevic, Hado van Hasselt, Charles Blundell, and Adrià Puigdomènech Badia. Human-level atari 200x faster. 2022.

Thomas Kipf, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. In *International Conference on Learning Representations*, 2022.

Hector Kohler, Quentin Delfosse, Riad Akrour, Kristian Kersting, and Philippe Preux. Interpretable and editable programmatic tree policies for reinforcement learning. In *Workshop on Interpretable Policies in Reinforcement Learning@ RLC-2024*, 2024.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.

Adam R. Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Neural Information Processing Systems*, 2018.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 2017.

Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Elliott Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, Andrey Kurenkov, C. Karen Liu, Hyowon Gweon, Jiajun Wu, Li Fei-Fei, and Silvio Savarese. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, pp. 455–465. PMLR, 2021.

Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. SPACE: unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations*, 2020a.

Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. SPACE: unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations*, 2020b.

Iou-Jen Liu, Zhongzheng Ren, Raymond A. Yeh, and Alexander G. Schwing. Semantic tracklets: An object-centric representation for visual multi-agent reinforcement learning. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Annual Conference on Neural Information Processing Systems*, 2020.

Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew J. Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents (extended abstract). In Jérôme Lang (ed.), *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. ijcai.org, 2018.

Davide Mambelli, Frederik Träuble, Stefan Bauer, Bernhard Schölkopf, and Francesco Locatello. Compositional multi-object reinforcement learning with linear relation networks. 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 2015.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Hugo Larochelle. Improving reproducibility in machine learning research(a report from the neurips 2019 reproducibility program). *J. Mach. Learn. Res.*, 2021.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8494–8502, 2018.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268), 2021.

Bradley Rettler and Andrew M. Bailey. Object. *Stanford Encyclopedia of Philosophy*, 2017.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019.

Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, 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.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy P. Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, 2017.

Yang Shao, Quan Kong, Tadayuki Matsumura, Taiki Fuji, Kiyoto Ito, and Hiroyuki Mizuno. Mask atari for deep reinforcement learning as POMDP benchmarks. 2022.

Gautam Singh, Fei Deng, and Sungjin Ahn. Illiterate DALL-E learns to compose. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022a.

Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. 2022b.

Ranveer Singh, Nikhilesh Prabhakar, Sriraam Natarajan, and Prasad Tadepalli. Exploiting relational planning and task-specific abstractions for multiagent reinforcement learning in relational domains. 2024.

Elizabeth S Spelke, Karen Breinlinger, Janet Macomber, and Kristen Jacobson. Origins of knowledge. *Psychological review*, 1992.

Aleksandar Stanić, Yujin Tang, David Ha, and Jürgen Schmidhuber. An investigation into the open world survival game crafter. In *Decision Awareness in Reinforcement Learning Workshop at ICML 2022*, 2022.

David Steinmann, Wolfgang Stammer, Felix Friedrich, and Kristian Kersting. Learning to intervene on concept bottlenecks. *ArXiv*, 2023.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*. PMLR, 2021.

Jakub Sygnowski and Henryk Michalewski. Learning from the memory of atari 2600. In *CGW@IJCAI*, 2016.

Joshua B. Tenenbaum, Charles Kemp, Thomas L. Griffiths, and Noah D. Goodman. How to grow a mind: Statistics, structure, and abstraction. *Science*, 2011.

Udo Thiel. *The early modern subject: Self-consciousness and personal identity from Descartes to Hume*. Oxford University Press, 2011.

Marin Toromanoff, Émilie Wirbel, and Fabien Moutarde. Is deep reinforcement learning really superhuman on atari? 2019.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In Dale Schuurmans and Michael P. Wellman (eds.), *AAAI Conference on Artificial Intelligence*, 2016.

Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

Yue Wu, Yewen Fan, Paul Pu Liang, Amos Azaria, Yuanzhi Li, and Tom M. Mitchell. Read and reap the rewards: Learning to play atari with the help of instruction manuals. 2023.

Antonia Wüst, Wolfgang Stammer, Quentin Delfosse, Devendra Singh Dhami, and Kristian Kersting. Pix2code: Learning to compose neural visual concepts as programs. *arXiv*, 2024.

Siwen Yan, Sriraam Natarajan, Saket Joshi, Roni Khardon, and Prasad Tadepalli. Explainable models via compression of tree ensembles. *Machine Learning*, 2024.

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

Vinícius Flores Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David P. Reichert, Timothy P. Lillicrap, Edward Lockhart, Murray Shanahan, Victoria Langston, Razvan Pascanu, Matthew M. Botvinick, Oriol Vinyals, and Peter W. Battaglia. Deep reinforcement learning with relational inductive biases. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Victor Zhong, Austin W. Hanjie, Sida I. Wang, Karthik Narasimhan, and Luke Zettlemoyer. SILG: the multi-domain symbolic interactive language grounding benchmark. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34*, 2021.

A ODA, an Object-centric Dataset for Atari.

OCAtari enables training policies using an object-centric approach to describe RL states for various Atari games. It can serve as a fast and dependable alternative to methods that discover objects. To compare object-centric agents to classic deep ones, it is necessary to train an object detection method and integrate it into the object-centric playing agent, *e.g.* , as shown by Delfosse et al. (2022). To train and compare the object detection methods, we introduce the **O**bject-centric **D**ataset for **A**tari (ODA), a preset selection of frames from the Atari games covered by OCAtari. For each game, ODAs incorporates sequential states, where for each state, the 210×160 RGB frame is stored with the list of objects found by both VEM and REM procedure (otherwise the game sequence is discarded). The HUD elements are separated from the game objects. Every additional object information contained from the RAM is also saved. As trained agents with varying capabilities can expose different parts of the environment, especially in progressive games where agents must achieve a certain level of mastery to reveal new parts of the game, it is necessary to fix the agents that are used to capture these frames (Delfosse et al., 2023a). The frames are extracted using both a random and a trained DQN agent to cover numerous possible states within each game, that should incorporate states encountered by learning agents. In many games, *e.g.* , Montezuma's Revenge or Kangaroo, such agents are not good enough to access every level of the game. However, as the level part is also stored in RAM, we let the agent start in different part of the game by manipulating the RAM. We choose to build our dataset out of 30% of games from the random agent and 70% of the games based on the DQN agent. All needed information, as well as the models used to generate ODA, are provided within the OCAtari repository.

B Details on object perception and its advantages

As described in our manuscript, decomposing the world in terms of objects incorporates many advantages, some of them are:

Abstraction and Generalization
Objects allow us to abstract and generalize information. By categorizing similar objects together, we can create concepts and classifications that help us make sense of a wide variety of individual instances.

Cognitive Efficiency
Our brains are more efficient at processing and remembering information when it's organized into meaningful chunks. Objects provide a natural way to group related information, making it easier for us to reason about complex situations.

Predictive Reasoning
Objects have properties and behaviors that can be predicted based on their past interactions and characteristics. This predictive reasoning is crucial for making informed decisions and anticipating outcomes.

Cause and Effect
Objects play a key role in understanding cause-and-effect relationships. By observing how objects interact and how changes in one object lead to changes in others, we can infer causal connections and predict future outcomes.

Communication
Objects provide a shared vocabulary that facilitates communication and understanding. When we refer to objects, we can convey complex ideas more efficiently than describing individual instances or specific situations..

Logical Inference
Objects provide a basis for logical reasoning. By identifying relationships between objects, we can deduce logical conclusions and make valid inferences.

C Details on OCAtari

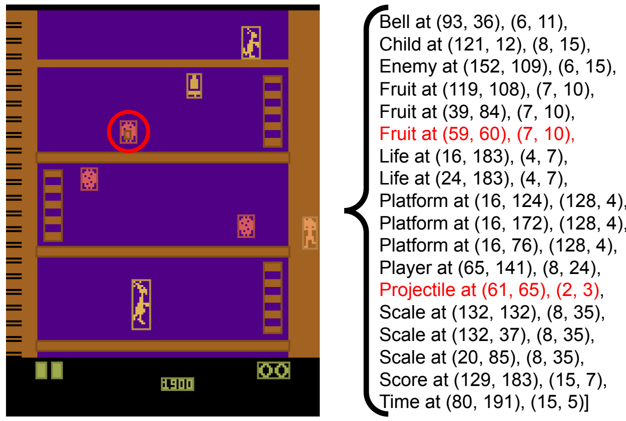


Figure 5: **OCAtari: The object-centric Atari benchmark.** OCAtari maintains a list of existing objects via processing the information from the RAM. Our framework enables training and evaluating object discovery methods and object-centric RL algorithms on the widely used Atari Learning Environments benchmark.

D Reproducing our Results

To reproduce our results, we included the option to run the experiments deterministically. For this purpose, a seed can be specified in the respective scripts. In our experiments, we used the seeds 0 and 42. All supported games can be found in Table 3. Since we are extending the environment permanently, you can also find all supported games in the ReadMe of our repository. To test if a game is supported, you can also use the scripts "test_game" or "test_game_both" depending on if you want to test only one or both modes of OCAtari. Table 1 and all tables in section G are generated by the script "get_metrics". To reproduce and measure the time needed for evaluation, see Figure 7, the script "test_speed" was used. For further information, we recommend checking the documentation of OCAtari under `https://oc-atari.readthedocs.io/` .

As mentioned before, we are using the models obtained from Gogianu et al. (2022)[3]. However for the games recently added to gymnasium, i.e. Pacman and Donkeykong, we needed to train our own agents. For this purpose we were using the *cleanRL* framework by Huang et al. (2022)[4].

[3] `https://github.com/floringogianu/atari-agents`
[4] `https://github.com/vwxyzjn/cleanrl`

E Experimental details

In our case, all experiments on object extraction and dataset generation were run on a machine with an AMD Ryzen 7 processor, 64GB of RAM, and no dedicated GPU. The dataset generation script takes approximately 3 minutes for one game. We use the same hyperparameters as the Schulman et al. (2017) PPO agents that learned to master the games. Hyperparameter values for Atari environments are derived from the original PPO paper. The same applies to the definitions and values of VF coefficient c_1 and entropy coefficient c_2. The PPO implementation used and respective MLP hyperparameters are based on stable-baselines3 (Raffin et al., 2021). Deep agents have the same hyperparameter values as OCAtari agents but use 'CnnPolicy' in stable-baselines3 for the policy architecture and frame stacking of 4. The Atari environment version used in gymnasium is v4 & v5. This version defines a deterministic skipping of 5 frames per action taken and sets the probability to repeat the last action taken to 0.25. This is aligned with recommended best practices by Machado et al. (2018). We also used the *Deterministic* and *NoFrameskip* features of gymnasium when necessary to make our experiments easier to reproduce. A list of all hyperparameter values used is provided in Table 4.

Actors N	8
Minibatch size	$32 * 8$
Horizon T	128
Num. epochs K	3
Adam stepsize	$2.5 * 10^{-4} * \alpha$
Discount γ	0.99
GAE parameter λ	0.95
Clipping parameter ϵ	$0.1 * \alpha$
VF coefficient c_1	1
Entropy coefficient c_2	0.01
MLP architecture	2×64
MLP activation fn.	Tanh

Table 4: PPO Hyperparameter Values. α linearly increases from 0 to 1 over the course of training.

F Generating Datasets

With OCAtari it is possible to create object-centric datasets for all supported games. The dataset consists primarily of a csv file. In addition to a sequential **index**, based on the game number and state number, this file contains the respective image as a list of pixels, called **OBS**. An image in the form of a png file is also stored separately. Furthermore, the csv file contains a list of all HUD elements that could be extracted from the RAM, called **HUD**, as well as a list of all objects that were read from the RAM, called **RAM**. Finally, we provide a list of all elements that could be generated using the vision mode, called **VIS**. An example is given in Table 5.

The generation of the dataset can also be made reproducible by setting a seed. For our tests, we used the seeds 0 and 42. More information at `https://github.com/k4ntz/OC_Atari/tree/master/dataset_generation` .

Table 5: An example how an object-centric dataset for Atari looks like after generation.

Index	OBS	HUD	RAM	VIS
00001_00001	[[0,0,0]...[255,255,255]]	score at (x,y)(width, height),...	ball at (x,y)(width, height),...	ball at (x,y)(width, height),.....
00001_00002	[[0,0,0]...[255,255,255]]	score at (x,y)(width, height),...	ball at (x,y)(width, height),...	ball at (x,y)(width, height),.....
00001_00003	[[0,0,0]...[255,255,255]]	score at (x,y)(width, height),...	ball at (x,y)(width, height),...	ball at (x,y)(width, height),.....
...				
00008_00678	[[0,0,0]...[255,255,255]]	score at (x,y)(width, height),...	ball at (x,y)(width, height),...	ball at (x,y)(width, height),.....

G Detailed Per Object Category results on each game.

In this section, we provide descriptions of each covered game (obtained from https://gymnasium.farama.org/environments/atari/) with example frames. For a more detailed documentation, see the game's respective AtariAge manual page[5]. We also share detailed statistics on the object detection capacities of OCAtari for every class of objects detected in each game.

Table 6: A more detailed version of Table 1. Precision, Recall, F1-scores of REM, and intersection over union (IOU) metrics. Frames are obtained using random, DQN and C51 (if available) agents.

	Random				DQN				C51			
	precision	recall	f-score	iou	precision	recall	f-score	iou	precision	recall	f-score	iou
Alien	51.4	97.3	67.3	97.7	51.2	97.2	67.1	97.4	N/A	N/A	N/A	N/A
Amidar	75.8	99.9	86.2	97.0	86.3	99.9	92.6	92.4	N/A	N/A	N/A	N/A
Assault	95.4	94.2	94.8	95.3	97.1	93.6	95.3	93.8	N/A	N/A	N/A	N/A
Asterix	93.1	99.8	96.3	96.0	95.0	99.6	97.2	96.1	94.8	99.8	97.2	96.2
Atlantis	96.3	94.6	95.5	95.8	96.6	94.7	95.7	95.0	N/A	N/A	N/A	N/A
BankHeist	87.9	95.8	91.7	87.3	96.2	96.2	96.2	94.7	N/A	N/A	N/A	N/A
BattleZone	81.1	55.7	66.0	95.1	81.8	51.8	63.4	93.5	N/A	N/A	N/A	N/A
Berzerk	94.1	95.2	94.6	78.4	94.3	96.5	95.4	77.4	N/A	N/A	N/A	N/A
Bowling	99.5	99.2	99.3	99.6	99.2	98.8	99.0	99.4	99.4	99.1	99.3	99.5
Boxing	96.5	84.5	90.1	93.5	96.1	84.5	89.9	93.4	96.8	85.6	90.9	94.1
Breakout	99.5	100	99.7	100	99.5	100	99.7	100	100	100	100	100
Carnival	93.2	94.2	93.7	90.7	94.6	96.4	95.5	91.5	N/A	N/A	N/A	N/A
Centipede	95.7	97.0	96.3	95.1	95.9	97.2	96.6	96.0	N/A	N/A	N/A	N/A
ChopperComma.	89.2	89.4	89.3	78.1	78.3	79.5	78.9	86.7	72.1	75.6	73.8	93.5
CrazyClimber	97.6	96.0	96.8	97.6	97.9	94.8	96.3	96.7	N/A	N/A	N/A	N/A
DemonAttack	62.6	78.6	69.7	79.9	59.5	77.6	67.3	84.1	N/A	N/A	N/A	N/A
DonkeyKong.	96.0	98.6	97.3	99.1	98.5	98.7	98.6	99.1	98.7	98.5	98.6	99.1
FishingDerby	89.2	85.6	87.3	75.2	88.8	84.6	86.6	73.6	83.2	77.9	80.5	75.7
Freeway	98.7	87.3	92.6	90.2	98.6	87.3	92.6	90.2	96.5	87.2	91.6	87.9
Frostbite	97.6	99.5	98.6	92.7	87.5	97.5	92.2	87.1	85.5	97.1	90.9	85.4
Gopher	98.3	48.2	64.7	78.4	98.3	47.6	64.1	84.0	N/A	N/A	N/A	N/A
Hero	92.4	94.6	93.5	88.2	79.0	88.4	83.4	86.3	80.8	91.7	85.9	86.7
IceHockey	89.2	99.6	94.1	66.2	92.4	99.7	95.9	66.3	N/A	N/A	N/A	N/A
Jamesbond	92.5	99.5	95.9	95.6	93.3	98.0	95.6	94.8	N/A	N/A	N/A	N/A
Kangaroo	96.7	93.1	94.9	95.6	98.3	93.2	95.7	94.8	96.1	93.1	94.6	95.2
Krull	94.8	96.8	95.8	89.1	95.6	96.7	96.2	89.4	N/A	N/A	N/A	N/A
MontezumaRev.	99.5	99.4	99.5	95.2	100	100	100	97.9	100	100	100	98.2
MsPacman	77.9	99.4	87.4	84.2	72.1	99.3	83.6	83.1	N/A	N/A	N/A	N/A
Pacman	58.5	92.7	71.7	80.4	51.3	88.6	65.0	77.4	47.6	83.1	60.5	72.1
Pitfall	98.2	99.0	98.6	95.8	100	100	100	96.6	N/A	N/A	N/A	N/A
Pong	90.0	99.1	94.3	81.7	94.3	98.8	96.5	83.2	93.8	97.4	95.6	84.7
PrivateEye	95.7	93.0	94.3	97.0	96.5	98.6	97.5	95.4	N/A	N/A	N/A	N/A
Qbert	94.4	99.0	96.6	99.6	74.7	98.3	84.9	98.4	77.3	98.4	86.6	98.5
Riverraid	93.5	98.0	95.7	93.6	89.3	98.0	93.5	91.0	N/A	N/A	N/A	N/A
RoadRunner	95.5	97.5	96.5	93.1	85.2	78.7	81.8	87.4	N/A	N/A	N/A	N/A
Seaquest	94.1	87.9	90.9	90.3	91.5	81.3	86.1	91.4	92.1	82.6	87.1	90.6
Skiing	95.8	96.5	96.2	90.4	94.1	94.2	94.2	89.3	N/A	N/A	N/A	N/A
SpaceInv.	95.2	98.7	96.9	97.1	90.6	95.9	93.1	97.3	N/A	N/A	N/A	N/A
Tennis	98.7	99.9	99.3	85.7	93.9	98.7	96.2	83.9	N/A	N/A	N/A	N/A
TimePilot	93.5	94.7	94.1	96.6	91.3	94.3	92.8	94.6	N/A	N/A	N/A	N/A
UpNDown	96.8	99.1	97.9	97.4	93.0	97.7	95.3	93.1	95.0	98.3	96.6	95.9
Venture	63.1	99.9	77.4	92.1	57.6	100	73.1	91.3	N/A	N/A	N/A	N/A
VideoPinball	98.3	94.6	96.4	94.4	99.5	95.3	97.3	95.3	N/A	N/A	N/A	N/A
mean	90.5	93.5	91.3	91.0	88.9	92.3	89.7	90.7	88.8	92.1	90.0	91.4

[5] https://atariage.com/system_items.php?SystemID=2600&itemTypeID=MANUAL

G.1 Alien details

You are stuck in a maze-like space ship with three aliens. You goal is to destroy their eggs that are scattered all over the ship while simultaneously avoiding the aliens (they are trying to kill you). You have a flamethrower that can help you turn them away in tricky situations. Moreover, you can occasionally collect a power-up (pulsar) that gives you the temporary ability to kill aliens.

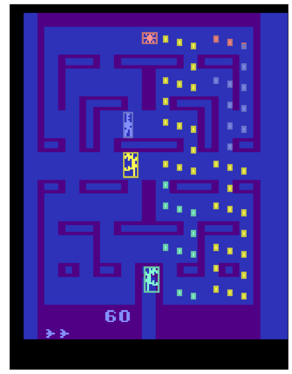


Table 7: Per class IOU on Alien

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Score	100	93.8	96.8	97.6	98.0	85.4	91.2	94.7	nan	nan	nan	nan
Egg	49.6	97.9	65.8	98.9	48.8	97.7	65.1	98.6	nan	nan	nan	nan
Life	99.8	99.6	99.7	100	100	100	100	100	nan	nan	nan	nan
Pulsar	67.2	82.0	73.8	80.6	66.4	82.4	73.5	79.1	nan	nan	nan	nan
Player	83.2	99.8	90.7	58.1	80.8	99.8	89.3	58.8	nan	nan	nan	nan
Alien	73.0	92.1	81.5	94.2	68.6	92.5	78.8	94.3	nan	nan	nan	nan

G.2 Amidar details

This game is similar to Pac-Man: You are trying to visit all places on a 2-dimensional grid while simultaneously avoiding your enemies. You can turn the tables at one point in the game: Your enemies turn into chickens and you can catch them.

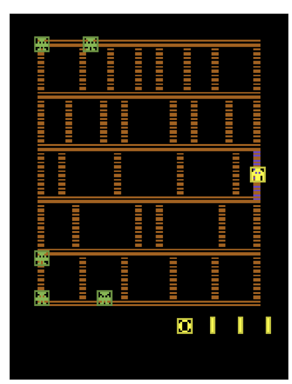


Table 8: Per class IOU on Amidar

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Life	100	100	100	100	100	100	100	100	nan	nan	nan	nan
Monster_green	60.9	99.9	75.7	95.4	75.7	99.8	86.1	87.4	nan	nan	nan	nan
Score	100	100	100	100	100	100	100	95.4	nan	nan	nan	nan
Player	97.4	100	98.7	95.5	95.6	100	97.8	92.9	nan	nan	nan	nan

G.3 Assault details

You control a vehicle that can move sideways. A big mother ship circles overhead and continually deploys smaller drones. You must destroy these enemies and dodge their attacks.

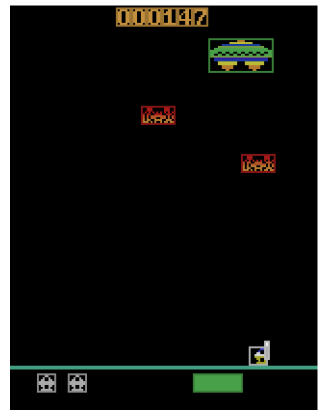


Table 9: Per class statistics on Assault

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
PlayerScore	100	100	100	100	100	100	100	99.9	nan	nan	nan	nan
MotherShip	99.8	99.8	99.8	88.7	100	100	100	88.6	nan	nan	nan	nan
Lives	100	100	100	100	100	100	100	100	nan	nan	nan	nan
Health	99.6	99.6	99.6	99.5	100	100	100	99.6	nan	nan	nan	nan
Player	91.8	100	95.7	88.6	95.8	100	97.9	81.6	nan	nan	nan	nan
Enemy	98.4	87.6	92.7	87.1	89.0	70.5	78.7	78.5	nan	nan	nan	nan
PlayerMissileHorizontal	29.0	29.1	29.1	28.5	26.7	25.5	26.1	30.2	nan	nan	nan	nan
PlayerMissileVertical	95.7	95.2	95.5	86.7	91.4	88.0	89.7	83.0	nan	nan	nan	nan
EnemyMissile	20.0	21.7	20.8	69.6	44.4	38.1	41.0	67.9	nan	nan	nan	nan

G.4 Asterix details

You are Asterix and can move horizontally (continuously) and vertically (discretely). Objects move horizontally across the screen: lyres and other (more useful) objects. Your goal is to guide Asterix in such a way as to avoid lyres and collect as many other objects as possible. You score points by collecting objects and lose a life whenever you collect a lyre. You have three lives available at the beginning. If you score sufficiently many points, you will be awarded additional points.

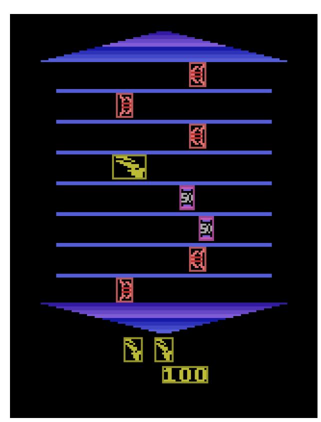


Table 10: Per class statistics on Asterix

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Lives	100	100	100	91.7	100	100	100	91.7	100	100	100	91.7
Player	98.4	98.4	98.4	97.6	99.6	99.6	99.6	98.8	98.6	98.6	98.6	98.5
Score	100	100	100	100	100	96.3	98.1	99.0	100	99.2	99.6	99.8
Cauldron	93.9	100	96.8	99.9	96.6	100	98.3	100	98.5	100	99.2	100
Reward50	90.7	100	95.1	100	98.2	100	99.1	100	90.6	100	95.1	99.6
Enemy	85.8	100	92.3	88.0	85.4	100	92.1	89.2	89.7	100	94.6	89.6

G.5 Asteroids details

This is a well-known arcade game: You control a spaceship in an asteroid field and must break up asteroids by shooting them. Once all asteroids are destroyed, you enter a new level and new asteroids will appear. You will occasionally be attacked by a flying saucer.

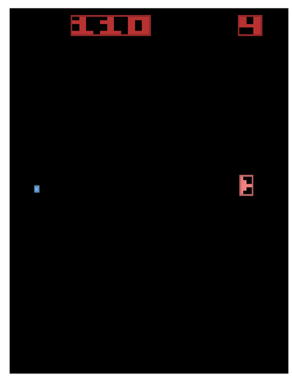


Table 11: Per class IOU on Asteroids

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Asteroid	42.5	86.7	57.1	90.5	39.5	88.8	54.7	89.3	nan	nan	nan	nan
Player	44.3	100	61.4	75.1	50.5	97.3	66.5	73.1	nan	nan	nan	nan
Lives	100	100	100	100	100	100	100	100	nan	nan	nan	nan
PlayerScore	96.0	92.5	94.2	98.5	97.8	96.1	96.9	99.1	nan	nan	nan	nan
PlayerMissile	44.7	97.8	61.4	98.7	52.7	92.8	67.2	100	nan	nan	nan	nan

G.6 Atlantis details

Your job is to defend the submerged city of Atlantis. Your enemies slowly descend towards the city and you must destroy them before they reach striking distance. To this end, you control three defense posts. You lose if your enemies manage to destroy all seven of Atlantis' installations. You may rebuild installations after you have fought of a wave of enemies and scored a sufficient number of points.

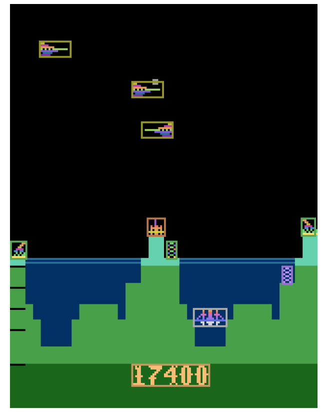


Table 12: Per class statistics on Atlantis

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
BridgedBazaar	100	99.7	99.9	99.9	100	99.7	99.8	99.9	nan	nan	nan	nan
AcropolisCommandPost	100	96.2	98.0	99.9	100	97.8	98.9	99.8	nan	nan	nan	nan
Sentry	100	100	100	99.4	100	100	100	99.6	nan	nan	nan	nan
AquaPlane	100	97.8	98.9	100	100	99.0	99.5	99.9	nan	nan	nan	nan
Generator	100	99.7	99.9	87.9	99.8	99.5	99.7	85.7	nan	nan	nan	nan
DomedPalace	100	99.5	99.8	100	100	99.8	99.9	99.9	nan	nan	nan	nan
Projectile	85.1	77.4	81.1	99.9	86.7	76.2	81.1	99.2	nan	nan	nan	nan
GorgonShip	88.6	83.0	85.7	93.1	85.7	76.0	80.6	89.9	nan	nan	nan	nan
Score	100	100	100	99.8	100	100	100	100	nan	nan	nan	nan
BanditBomber	81.0	78.2	79.5	88.2	76.3	85.3	80.6	79.4	nan	nan	nan	nan

G.7 BankHeist details

You are a bank robber and (naturally) want to rob as many banks as possible. You control your getaway car and must navigate maze-like cities. The police chases you and will appear whenever you rob a bank. You may destroy police cars by dropping sticks of dynamite. You can fill up your gas tank by entering a new city.At the beginning of the game you have four lives. Lives are lost if you run out of gas, are caught by the police,or run over the dynamite you have previously dropped.

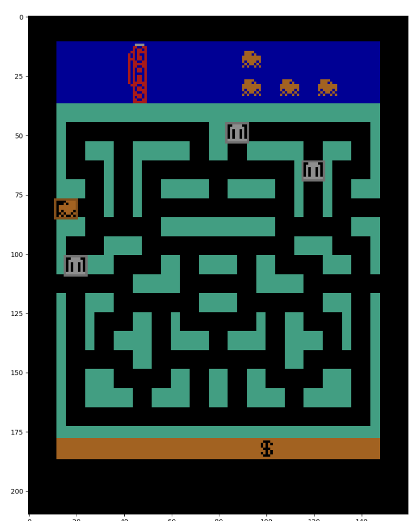


Table 13: Per class IOU on BankHeist

	Random				DQN				C51			
	Precision	Recall	F-score	IOU	Precision	Recall	F-score	IOU	Precision	Recall	F-score	IOU
Bank	73.4	87.2	79.7	73.5	98.7	91.3	94.9	99.2	nan	nan	nan	nan
Player	79.8	99.8	88.7	81.0	71.2	92.0	80.3	77.1	nan	nan	nan	nan
Gas_Tank	100	100	100	92.6	100	100	100	76.9	nan	nan	nan	nan
Score	100	100	100	100	100	100	100	100	nan	nan	nan	nan
Life	100	100	100	100	100	100	100	100	nan	nan	nan	nan
Police	100	100	100	82.5	89.2	89.2	89.2	71.5	nan	nan	nan	nan

G.8 BattleZone details

You control a tank and must destroy enemy vehicles. This game is played in a first-person perspective and creates a 3D illusion. A radar screen shows enemies around you. You start with 5 lives and gain up to 2 extra lives if you reach a sufficient score.

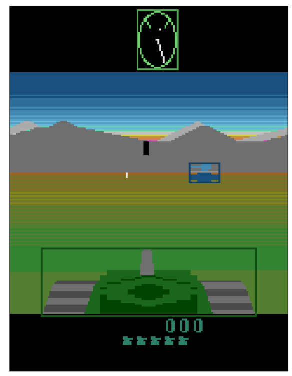


Table 14: Per class IOU on BattleZone

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Radar	92.0	100	95.8	100	92.8	94.5	93.6	99.6	nan	nan	nan	nan
Player	92.0	100	95.8	100	92.8	100	96.3	100	nan	nan	nan	nan
Crosshair	92.0	68.0	78.2	97.5	88.6	70.5	78.5	97.3	nan	nan	nan	nan
Blue_Tank	27.1	50.8	35.4	57.6	27.3	47.6	34.7	54.9	nan	nan	nan	nan

G.9 Berzerk details

You are stuck in a maze with evil robots. You must destroy them and avoid touching the walls of the maze, as this will kill you. You may be awarded extra lives after scoring a sufficient number of points, depending on the game mode. You may also be chased by an undefeatable enemy, Evil Otto, that you must avoid. Evil Otto does not appear in the default mode.

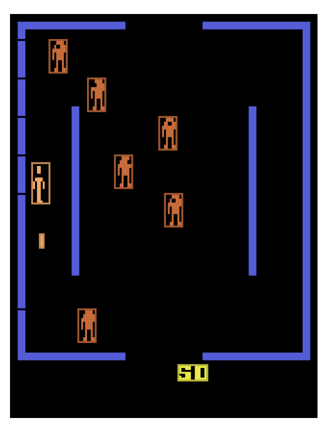


Table 15: Per class statistics on Berzerk

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Logo	100	99.3	99.6	100	100	100	100	100	nan	nan	nan	nan
PlayerMissile	75.0	98.4	85.1	74.5	79.2	88.1	83.4	81.1	nan	nan	nan	nan
Enemy	97.3	98.7	98.0	77.0	98.4	100	99.2	77.3	nan	nan	nan	nan
Player	90.4	98.7	94.4	66.1	97.4	99.2	98.3	54.5	nan	nan	nan	nan
PlayerScore	95.0	73.2	82.7	85.1	98.8	91.9	95.2	96.6	nan	nan	nan	nan
EnemyMissile	77.4	90.0	83.2	78.8	73.8	85.7	79.3	79.2	nan	nan	nan	nan
RoomCleared	96.3	100	98.1	100	98.4	100	99.2	100	nan	nan	nan	nan

G.10 Bowling details

Your goal is to score as many points as possible in the game of Bowling. A game consists of 10 frames and you have two tries per frame. Knocking down all pins on the first try is called a "strike". Knocking down all pins on the second roll is called a "spar". Otherwise, the frame is called "open".

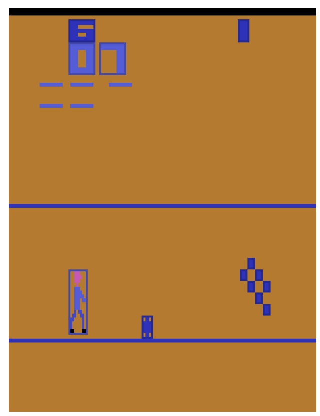


Table 16: Per class statistics on Bowling

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Pin	99.4	100	99.7	100	98.7	100	99.3	100	99.4	100	99.7	100
Player	98.6	97.6	98.1	98.3	99.6	99.2	99.4	99.2	98.2	96.8	97.5	98.0
PlayerScore	100	100	100	100	100	100	100	100	100	100	100	100
Player2Round	100	100	100	100	100	100	100	100	100	100	100	100
Ball	99.0	98.8	98.9	99.6	98.4	96.9	97.6	99.1	99.0	98.6	98.8	99.3
PlayerRound	100	92.4	96.1	96.7	100	89.6	94.5	95.4	100	92.1	95.9	96.6

G.11 Boxing details

You fight an opponent in a boxing ring. You score points for hitting the opponent. If you score 100 points, your opponent is knocked out.

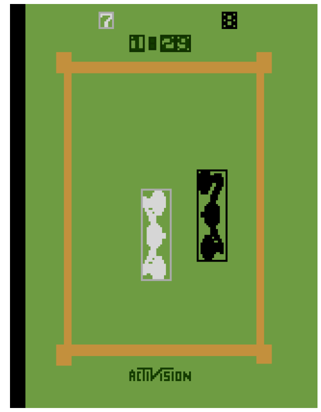


Table 17: Per class statistics on Boxing

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Enemy	83.4	81.1	82.2	78.4	79.8	74.6	77.1	73.3	88.6	85.5	87.0	79.5
PlayerScore	100	82.5	90.4	87.9	100	92.8	96.2	95.5	100	93.3	96.5	95.9
Player	81.6	81.6	81.6	79.3	80.8	80.8	80.8	78.4	79.6	79.1	79.4	76.7
EnemyScore	100	45.9	62.9	89.8	100	44.9	62.0	87.1	100	45.5	62.5	88.7
Logo	100	100	100	100	100	100	100	100	100	100	100	100
Clock	100	100	100	100	100	100	100	100	100	100	100	100

G.12 Breakout details

Another famous Atari game. The dynamics are similar to pong: You move a paddle and hit the ball in a brick wall at the top of the screen. Your goal is to destroy the brick wall. You can try to break through the wall and let the ball wreak havoc on the other side, all on its own! You have five lives.

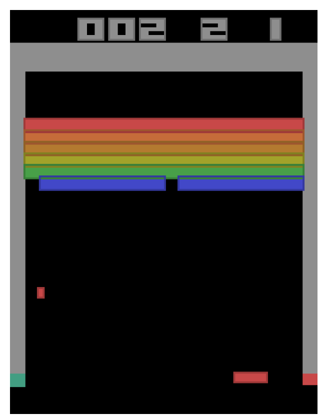


Table 18: Per class statistics on Breakout

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Player	98.8	100	99.4	100	97.4	100	98.7	100	99.8	100	99.9	100
BlockRow	100	100	100	100	100	100	100	100	100	100	100	100
Live	100	100	100	99.9	100	100	100	99.9	100	100	100	100
PlayerScore	100	100	100	100	100	100	100	100	100	100	100	100
PlayerNumber	100	100	100	100	100	100	100	100	100	100	100	100
Ball	93.1	100	96.4	100	93.3	100	96.5	100	90.5	100	95.0	100

G.13 Carnival details

This is a "shoot 'em up" game. Targets move horizontally across the screen and you must shoot them. You are in control of a gun that can be moved horizontally. The supply of ammunition is limited and chickens may steal some bullets from you if you don't hit them in time.

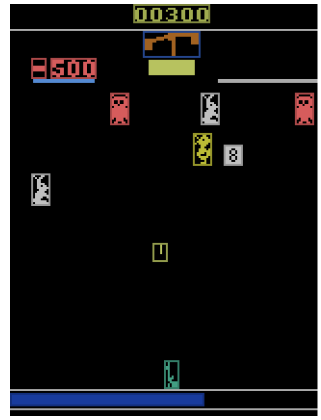


Table 19: Per class statistics on Carnival

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Duck	98.8	93.7	96.2	95.5	98.3	93.4	95.8	94.4	nan	nan	nan	nan
PlayerScore	86.6	76.4	81.2	100	98.4	96.9	97.6	100	nan	nan	nan	nan
Wheel	100	100	100	89.8	100	98.9	99.5	88.5	nan	nan	nan	nan
FlyingDuck	40.4	91.3	56.0	82.2	42.9	82.9	56.5	82.9	nan	nan	nan	nan
Owl	99.0	97.7	98.3	97.6	98.9	96.2	97.5	95.9	nan	nan	nan	nan
ExtraBullets	98.1	84.9	91.0	97.8	98.1	90.2	94.0	96.6	nan	nan	nan	nan
Player	100	100	100	100	100	100	100	100	nan	nan	nan	nan
Rabbit	97.0	95.8	96.4	98.0	95.3	97.3	96.3	97.4	nan	nan	nan	nan
AmmoBar	90.0	100	94.7	100	98.4	99.8	99.1	99.8	nan	nan	nan	nan
PlayerMissile	95.3	97.5	96.4	10.2	93.9	98.9	96.3	10.4	nan	nan	nan	nan
BonusValue	95.0	100	97.5	88.1	97.4	100	98.7	82.1	nan	nan	nan	nan
BonusSign	66.3	100	79.7	65.6	53.1	100	69.4	100	nan	nan	nan	nan

G.14 Centipede details

You are an elf and must use your magic wands to fend off spiders, fleas and centipedes. Your goal is to protect mushrooms in an enchanted forest. If you are bitten by a spider, flea or centipede, you will be temporally paralyzed and you will lose a magic wand. The game ends once you have lost all wands. You may receive additional wands after scoring a sufficient number of points.

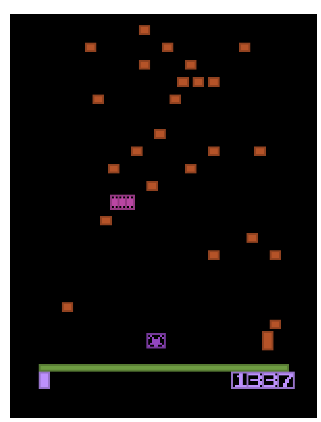


Table 20: Per class statistics on Centipede

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Score	99.6	100	99.8	97.0	98.8	100	99.4	97.7	nan	nan	nan	nan
Projectile	84.9	99.7	91.7	88.9	84.8	97.5	90.7	88.6	nan	nan	nan	nan
Life	100	100	100	100	100	100	100	100	nan	nan	nan	nan
Ground	100	97.8	98.9	100	100	98.4	99.2	100	nan	nan	nan	nan
Mushroom	99.6	99.7	99.6	99.8	99.3	99.7	99.5	99.6	nan	nan	nan	nan
CentipedeSegment	72.2	78.8	75.4	57.1	69.0	75.2	72.0	55.3	nan	nan	nan	nan
Player	96.4	91.8	94.1	87.4	94.4	90.8	92.6	84.8	nan	nan	nan	nan
Spider	92.4	100	96.1	99.6	87.1	99.0	92.7	100	nan	nan	nan	nan
Flea	50.0	100	66.7	100	50.0	100	66.7	100	nan	nan	nan	nan
Scorpion	nan	nan	nan	nan	100	100	100	100	nan	nan	nan	nan

G.15 ChopperCommand details

You control a helicopter and must protect truck convoys. To that end, you need to shoot down enemy aircraft.A mini-map is displayed at the bottom of the screen.

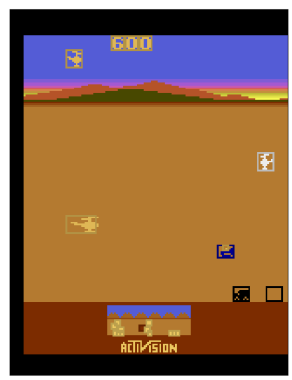


Table 21: Per class statistics on ChopperCommand

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Score	100	100	100	100	100	100	100	100	100	100	100	100
Life	100	100	100	100	100	100	100	100	100	100	100	100
MiniEnemy	88.9	78.0	83.1	48.5	66.8	62.3	64.5	51.1	47.4	39.4	43.0	56.2
MiniTruck	86.0	93.1	89.4	100	68.5	70.6	69.6	99.9	49.4	58.5	53.6	100
Truck	88.8	99.6	93.9	79.6	95.9	98.6	97.2	80.4	92.9	96.4	94.6	74.7
MiniPlayer	90.8	100	95.2	84.8	99.2	100	99.6	84.5	99.8	100	99.9	93.7
Player	95.2	98.8	96.9	79.0	91.4	89.4	90.4	75.8	99.0	98.8	98.9	86.5
Shot	76.9	74.9	75.9	88.0	58.8	58.8	58.8	87.3	81.8	81.8	81.8	90.2
EnemyHelicopter	79.9	94.3	86.5	71.4	45.6	85.4	59.5	73.7	61.8	95.5	75.0	71.1
Bomb	51.3	93.2	66.2	37.7	54.7	80.3	65.0	55.9	50.0	82.1	62.2	72.5
EnemyPlane	92.0	98.3	95.0	66.6	94.0	96.2	95.1	71.4	94.4	94.4	94.4	74.4

G.16 CrazyClimber details

You are a climber trying to reach the top of four buildings, while avoiding obstacles like closing windows and falling objects. When you receive damage (windows closing or objects) you will fall and lose one life; you have a total of 5 lives before the end games. At the top of each building, there's a helicopter which you need to catch to get to the next building. The goal is to climb as fast as possible while receiving the least amount of damage.

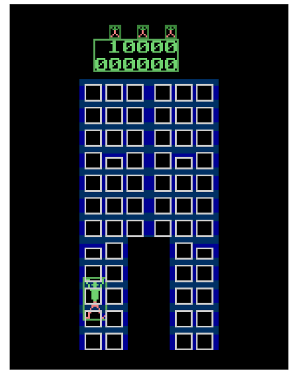


Table 22: Per class IOU on CrazyClimber

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Player	96.0	97.6	96.8	91.1	90.8	95.0	92.8	90.4	nan	nan	nan	nan
Window	97.6	95.2	96.4	97.4	98.3	94.6	96.4	96.5	nan	nan	nan	nan
Score	100	100	100	100	100	100	100	100	nan	nan	nan	nan
Life	100	100	100	100	100	100	100	100	nan	nan	nan	nan
Enemy_Red	75.0	81.8	78.3	55.3	19.1	78.8	30.8	57.2	nan	nan	nan	nan
Purple_Projectile	66.7	66.7	66.7	45.0	56.2	69.2	62.1	52.8	nan	nan	nan	nan
Yellow_Projectile	33.3	50.0	40.0	70.9	44.4	57.1	50.0	79.7	nan	nan	nan	nan
Yellow_Ball	84.0	91.3	87.5	67.8	70.5	79.5	74.7	64.8	nan	nan	nan	nan
Enemy_Bird	69.7	92.0	79.3	76.1	58.5	79.2	67.3	65.0	nan	nan	nan	nan
Helicopter	nan	nan	nan	nan	14.3	14.3	14.3	42.7	nan	nan	nan	nan
Blue_Projectile	nan	nan	nan	nan	100	100	100	53.6	nan	nan	nan	nan

G.17 DemonAttack details

You are facing waves of demons in the ice planet of Krybor. Points are accumulated by destroying demons. You begin with 3 reserve bunkers, and can increase its number (up to 6) by avoiding enemy attacks. Each attack wave you survive without any hits, grants you a new bunker. Every time an enemy hits you, a bunker is destroyed. When the last bunker falls, the next enemy hit will destroy you and the game ends.

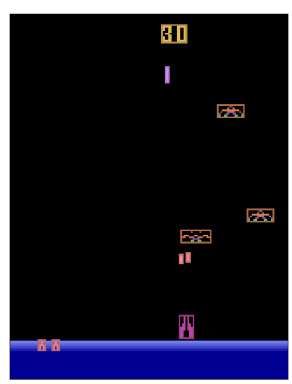


Table 23: Per class IOU on DemonAttack

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
ProjectileFriendly	94.8	100	97.3	83.8	97.2	99.4	98.3	85.1	nan	nan	nan	nan
Score	100	98.6	99.3	97.8	98.6	91.1	94.7	95.8	nan	nan	nan	nan
Player	92.6	100	96.2	100	97.2	100	98.6	100	nan	nan	nan	nan
Live	99.1	100	99.5	100	97.5	100	98.8	100	nan	nan	nan	nan
Enemy	99.3	97.9	98.6	73.7	75.1	56.6	64.5	66.3	nan	nan	nan	nan
ProjectileHostile	80.2	98.7	88.5	55.4	59.9	98.3	74.4	37.2	nan	nan	nan	nan

G.18 DonkeyKong details

You play as Mario trying to save your girlfriend who has been kidnapped by Donkey Kong. Remove rivets and jump over fireballs, with a score that starts high and counts down throughout the game.

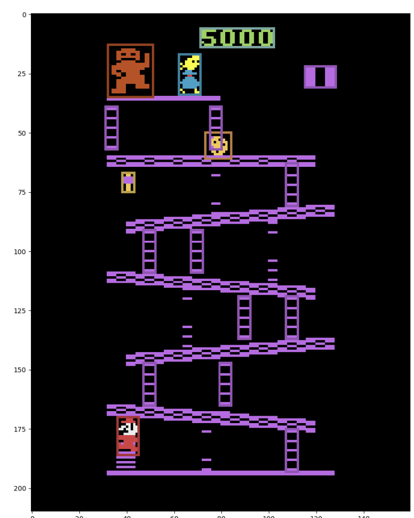


Table 24: Per class IOU on DonkeyKong

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Score	97.6	79.5	87.6	93.9	98.0	81.0	88.7	94.1	98.0	79.2	87.6	93.5
Player	28.8	99.3	44.7	93.0	74.8	100	85.6	97.7	78.0	99.7	87.5	97.0
Girlfriend	100	100	100	100	100	100	100	100	100	100	100	100
Ladder	100	100	100	100	100	100	100	100	100	100	100	100
Life	100	100	100	88.9	100	100	100	88.9	100	100	100	88.9
Hammer	99.8	99.6	99.7	100	100	100	100	100	100	100	100	100
DonkeyKong	100	100	100	100	100	100	100	100	100	100	100	100
Barrel	100	100	100	99.6	100	99.5	99.7	99.4	100	99.4	99.7	99.5

G.19 FishingDerby details

Your objective is to catch more sunfish than your opponent.

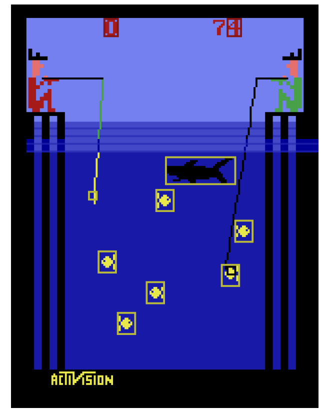


Table 25: Per class statistics on FishingDerby

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
ScorePlayerTwo	100	53.1	69.4	100	100	52.5	68.9	100	100	52.5	68.9	100
Fish	94.5	98.2	96.3	68.7	90.6	98.5	94.4	69.2	87.1	99.3	92.8	68.8
PlayerTwoHook	62.6	66.2	64.3	29.4	62.8	67.7	65.1	26.5	62.0	65.1	63.5	30.9
ScorePlayerOne	100	96.3	98.1	100	100	73.5	84.7	100	100	52.5	68.8	100
PlayerOneHook	69.0	69.0	69.0	22.6	87.4	87.6	87.5	22.5	53.8	56.9	55.3	22.4
Shark	82.4	99.8	90.2	92.1	82.8	99.5	90.4	91.5	77.0	99.7	86.9	92.7

G.20 Freeway details

Your objective is to guide your chicken across lane after lane of busy rush hour traffic. You receive a point for every chicken that makes it to the top of the screen after crossing all the lanes of traffic.

Table 26: Per class statistics on Freeway

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Chicken	100	100	100	97.1	99.8	99.9	99.8	96.9	97.0	98.5	97.7	96.4
Car	99.1	99.9	99.5	87.0	99.3	99.9	99.6	87.0	99.4	99.8	99.6	87.0
Score	95.0	48.8	64.5	100	93.4	48.4	63.7	100	85.0	52.3	64.8	86.1

G.21 Frostbite details

In Frostbite, the player controls "Frostbite Bailey" who hops back and forth across across an Arctic river, changing the color of the ice blocks from white to blue. Each time he does so, a block is added to his igloo.

Table 27: Per class statistics on Frostbite

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
WhitePlate	99.5	99.7	99.6	92.0	93.2	99.8	96.4	88.5	92.0	98.8	95.3	87.1
Degree	100	100	100	98.0	100	100	100	97.2	100	100	100	97.7
PlayerScore	100	100	100	96.0	100	100	100	89.7	100	100	100	89.8
Player	63.8	100	77.9	71.4	66.8	100	80.1	72.1	75.8	100	86.2	70.9
LifeCount	100	100	100	89.0	100	100	100	87.2	100	100	100	87.9
House	100	100	100	99.6	99.7	100	99.9	97.1	100	100	100	94.6
BluePlate	98.8	99.5	99.1	91.6	78.0	100	87.6	84.7	74.0	98.2	84.4	82.8
Bird	98.2	95.3	96.7	99.1	90.1	79.7	84.6	95.6	91.4	84.4	87.7	94.7
CompletedHouse	nan	nan	nan	nan	100	100	100	99.6	100	100	100	99.8
GreenFish	nan	nan	nan	nan	82.2	77.8	79.9	67.2	84.2	87.8	86.0	70.1
Crab	nan	nan	nan	nan	96.4	78.4	86.5	66.3	96.6	81.1	88.2	63.5
Bear	nan	nan	nan	nan	90.0	90.0	90.0	77.3	100	91.4	95.5	80.6
Clam	nan	nan	nan	nan	59.5	97.8	73.9	75.1	66.7	100	80.0	58.3

G.22 Gopher details

The player controls a shovel-wielding farmer who protects a crop of three carrots from a gopher.

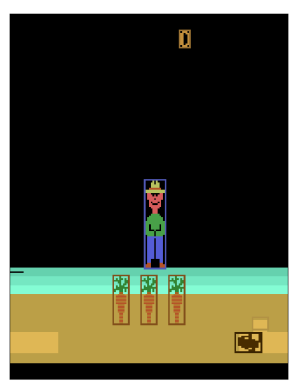


Table 28: Per class IOU on Gopher

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Gopher	74.8	97.7	84.7	84.7	71.8	96.0	82.2	83.3	nan	nan	nan	nan
Score	100	100	100	98.4	100	96.9	98.4	93.1	nan	nan	nan	nan
Player	99.8	99.8	99.8	81.7	100	100	100	80.3	nan	nan	nan	nan
Empty_Block	100	35.0	51.8	65.6	100	24.6	39.5	62.5	nan	nan	nan	nan
Carrot	100	100	100	100	99.9	100	100	100	nan	nan	nan	nan
Bird	nan	nan	nan	nan	33.8	98.0	50.3	84.8	nan	nan	nan	nan

G.23 Hero details

You need to rescue miners that are stuck in a mine shaft. You have access to various tools: A propeller backpack that allows you to fly wherever you want, sticks of dynamite that can be used to blast through walls, a laser beam to kill vermin, and a raft to float across stretches of lava.You have a limited amount of power. Once you run out, you lose a live.

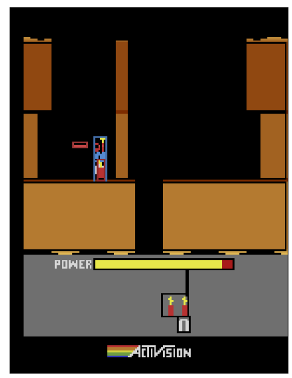


Table 29: Per class IOU on Hero

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Life	65.0	100	78.8	71.1	84.8	100	91.8	75.6	81.4	100	89.7	78.7
Score	65.8	70.6	68.1	48.7	54.4	62.0	57.9	57.6	76.8	82.9	79.8	59.3
Player	96.0	99.8	97.9	96.7	91.0	99.8	95.2	99.4	94.0	100	96.9	99.5
PowerBar	100	100	100	99.8	100	100	100	99.8	100	100	100	99.9
BombStock	81.8	100	90.0	81.0	98.8	100	99.4	90.1	99.4	100	99.7	80.9
Wall	100	93.6	96.7	98.5	73.2	91.4	81.3	91.8	83.9	96.4	89.7	95.0
LaserBeam	36.8	84.2	51.2	35.9	19.8	81.4	31.8	34.2	15.0	81.8	25.3	33.0
Bomb	42.3	81.1	55.6	83.3	45.9	28.3	35.0	83.3	41.3	18.7	25.7	75.3
Enemy	94.1	100	97.0	33.2	37.0	71.8	48.8	43.0	39.9	64.9	49.4	39.0
EndNPC	100	100	100	69.2	100	83.8	91.2	69.2	100	77.5	87.3	69.2
Lamp	nan	nan	nan	nan	46.6	100	63.6	62.5	82.0	100	90.1	62.5
LavaWall	nan	nan	nan	nan	45.3	77.4	57.1	92.3	54.4	82.8	65.7	99.2

G.24 IceHockey details

Your goal is to score as many points as possible in a standard game of Ice Hockey over a 3-minute time period. The ball is usually called "the puck".There are 32 shot angles ranging from the extreme left to the extreme right. The angles can only aim towards the opponent's goal.Just as in real hockey, you can pass the puck by shooting it off the sides of the rink. This can be really key when you're in position to score a goal.

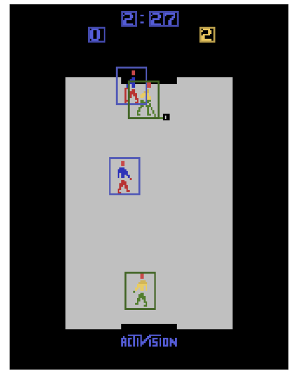


Table 30: Per class IOU on IceHockey

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
EnemyScore	71.6	100	83.4	76.4	57.1	100	72.7	76.4	nan	nan	nan	nan
Player	99.1	99.1	99.1	47.8	98.2	98.2	98.2	47.9	nan	nan	nan	nan
Enemy	99.0	99.5	99.2	52.4	99.6	99.6	99.6	52.8	nan	nan	nan	nan
PlayerScore	83.8	100	91.2	67.7	65.3	100	79.0	67.7	nan	nan	nan	nan
Ball	85.2	98.6	91.4	80.2	83.4	98.8	90.5	84.1	nan	nan	nan	nan
Timer	100	100	100	80.5	100	100	100	79.7	nan	nan	nan	nan

G.25 Jamesbond details

Your mission is to control Mr. Bond's specially designed multipurpose craft to complete a variety of missions.The craft moves forward with a right motion and slightly back with a left motion.An up or down motion causes the craft to jump or dive.You can also fire by either lobbing a bomb to the bottom of the screen or firing a fixed angle shot to the top of the screen.

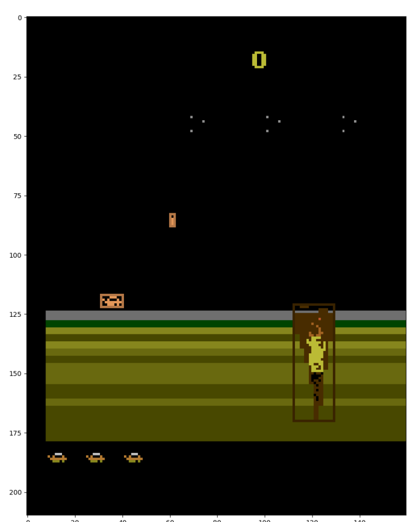


Table 31: Per class IOU on Jamesbond

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Player_Shot	78.0	100	87.6	82.2	88.6	100	94.0	79.7	nan	nan	nan	nan
Fire_Hole	100	99.4	99.7	99.8	94.7	88.5	91.5	93.5	nan	nan	nan	nan
Score	100	100	100	99.8	100	100	100	98.5	nan	nan	nan	nan
Player	82.6	96.9	89.2	100	91.8	98.9	95.2	99.8	nan	nan	nan	nan
Life	99.7	100	99.9	100	99.7	100	99.8	100	nan	nan	nan	nan
Helicopter	98.8	100	99.4	86.7	97.0	99.7	98.3	84.9	nan	nan	nan	nan
Hornet	73.7	100	84.8	90.6	76.1	99.6	86.3	95.6	nan	nan	nan	nan
Enemy_Shot	80.5	99.0	88.8	70.6	71.5	100	83.4	74.7	nan	nan	nan	nan
Ice	91.5	100	95.6	86.7	91.7	100	95.7	82.7	nan	nan	nan	nan
Eruption	nan	nan	nan	nan	100	100	100	100	nan	nan	nan	nan

G.26 Kangaroo details

The object of the game is to score as many points as you can while controlling Mother Kangaroo to rescue her precious baby. You start the game with three lives.During this rescue mission, Mother Kangaroo encounters many obstacles. You need to help her climb ladders, pick bonus fruit, and throw punches at monkeys.

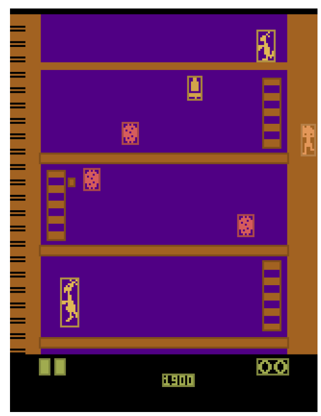


Table 32: Per class statistics on Kangaroo

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Bell	99.8	99.8	99.8	100	100	100	100	100	99.8	99.8	99.8	100
Platform	100	75.0	85.7	100	100	75.0	85.7	100	100	75.0	85.7	100
Scale	100	100	100	100	100	100	100	100	100	100	100	100
Fruit	99.8	99.9	99.9	90.7	99.2	100	99.6	90.9	97.4	99.8	98.6	90.7
Child	99.6	99.8	99.7	95.0	100	100	100	95.6	99.8	100	99.9	96.4
Life	99.7	100	99.8	100	100	100	100	100	99.6	100	99.8	100
Score	99.8	100	99.9	100	100	100	100	99.6	99.8	100	99.9	99.3
Time	99.8	100	99.9	100	100	100	100	89.7	99.8	100	99.9	96.5
Player	79.8	89.3	84.3	79.4	88.0	91.7	89.8	78.6	81.2	88.3	84.6	79.3
Projectile_top	81.6	84.7	83.1	81.0	98.9	99.5	99.2	87.6	92.9	88.3	90.5	85.5
Enemy	87.2	95.5	91.2	91.1	94.5	95.2	94.8	86.8	85.8	96.4	90.8	86.8
Projectile_enemy	66.7	93.3	77.8	33.3	50.0	100	66.7	33.3	14.1	86.7	24.2	33.3

G.27 Krull details

Your mission is to find and enter the Beast's Black Fortress, rescue Princess Lyssa, and destroy the Beast.The task is not an easy one, for the location of the Black Fortress changes with each sunrise on Krull.

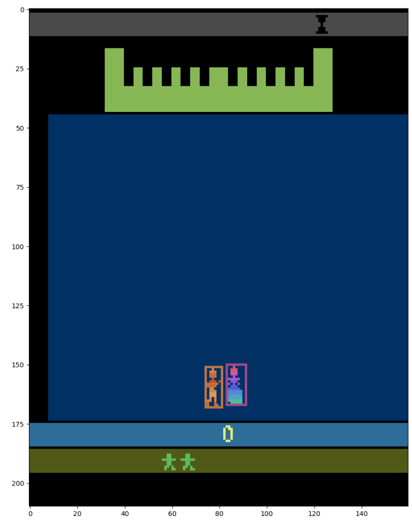


Table 33: Per class IOU on Krull

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Life	99.8	99.5	99.7	100	100	100	100	100	nan	nan	nan	nan
Lyssa	84.1	98.6	90.8	84.3	95.5	98.4	96.9	81.7	nan	nan	nan	nan
Player	88.6	99.3	93.7	74.1	90.0	100	94.7	75.3	nan	nan	nan	nan
Score	100	100	100	98.3	99.8	100	99.9	99.1	nan	nan	nan	nan
Sun	93.4	100	96.6	87.1	91.3	100	95.5	87.3	nan	nan	nan	nan
Slayers	96.2	97.4	96.8	83.8	99.6	99.1	99.3	82.3	nan	nan	nan	nan
Slayer_Shot	0.0	nan	0.0	nan	0.0	nan	0.0	nan	nan	nan	nan	nan
Weapon	100	91.7	95.7	96.0	82.4	93.3	87.5	96.8	nan	nan	nan	nan
Fire_Mare	100	100	100	49.0	100	100	100	49.7	nan	nan	nan	nan
Window	99.5	100	99.7	100	99.7	100	99.9	100	nan	nan	nan	nan
Hour_Glass	99.5	83.2	90.6	82.6	99.7	82.6	90.4	82.0	nan	nan	nan	nan
Spider	89.3	98.8	93.8	76.9	88.1	98.9	93.2	77.1	nan	nan	nan	nan
Weapon_HUD	100	100	100	100	99.3	100	99.7	100	nan	nan	nan	nan

G.28 MontezumaRevenge details

Your goal is to acquire Montezuma's treasure by making your way through a maze of chambers within the emperor's fortress. You must avoid deadly creatures while collecting valuables and tools which can help you escape with the treasure.

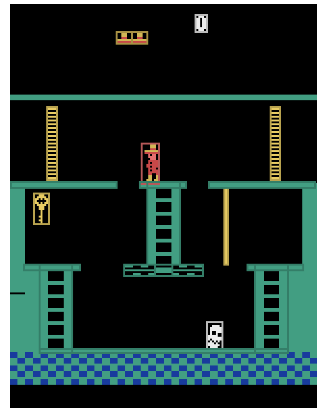


Table 34: Per class statistics on MontezumaRevenge

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Skull	99.6	99.6	99.6	79.0	100	100	100	79.1	100	100	100	80.4
Life	100	100	100	100	100	100	100	100	100	100	100	100
Player	99.0	98.6	98.8	77.9	100	100	100	97.3	100	100	100	100
Rope	97.6	100	98.8	100	100	100	100	100	100	100	100	100
Barrier	100	100	100	100	100	100	100	100	100	100	100	100
Key	99.0	96.7	97.8	100	100	100	100	100	100	100	100	100
Score	100	100	100	100	100	100	100	100	100	100	100	100

G.29 MsPacman details

Your goal is to collect all of the pellets on the screen while avoiding the ghosts.

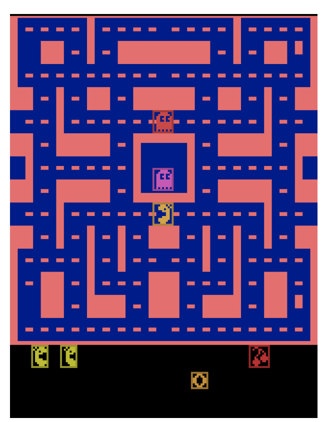


Table 35: Per class statistics on MsPacman

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Life	100	100	100	93.3	54.4	100	70.5	90.3	nan	nan	nan	nan
Score	100	100	100	98.2	100	100	100	99.2	nan	nan	nan	nan
Player	99.8	99.8	99.8	71.7	94.8	99.0	96.8	72.5	nan	nan	nan	nan
Ghost	55.6	98.4	71.1	79.5	61.3	98.3	75.5	77.2	nan	nan	nan	nan
Fruit	100	100	100	88.9	97.9	100	98.9	85.5	nan	nan	nan	nan

G.30 Pacman details

A classic arcade game. Move Pac Man around a maze collecting food and avoiding ghosts- unless you eat a Power Pellet, then you can eat the ghosts too!

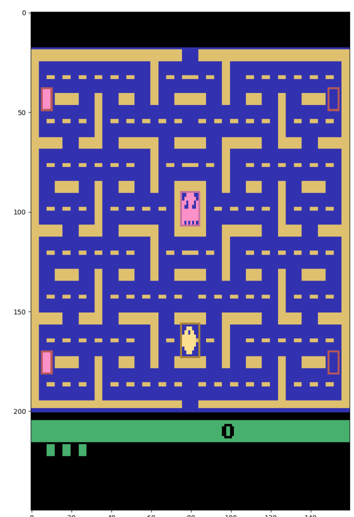


Table 36: Per class IOU on Pacman

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Score	100	98.2	99.1	35.5	100	100	100	34.1	93.4	86.0	89.5	35.6
Player	93.5	87.8	90.5	66.8	95.2	88.6	91.7	69.3	90.0	73.8	81.1	67.4
Ghost	40.2	82.3	54.0	84.8	31.2	79.7	44.9	87.9	21.0	72.3	32.6	86.5
Life	100	99.8	99.9	99.7	100	100	100	100	100	99.6	99.8	100
PowerPill	49.2	100	66.0	100	33.0	82.1	47.1	99.7	26.2	99.4	41.4	100

G.31 Pitfall details

You control Pitfall Harry and are tasked with collecting all the treasures in a jungle within 20 minutes. You have three lives. The game is over if you collect all the treasures or if you die or if the time runs out.

Table 37: Per class IOU on Pitfall

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Player	91.8	97.0	94.3	68.7	100	100	100	61.2	nan	nan	nan	nan
Wall	100	100	100	99.8	100	100	100	76.5	nan	nan	nan	nan
Logs	99.1	99.8	99.5	99.6	100	100	100	98.9	nan	nan	nan	nan
LifeCount	100	100	100	100	100	100	100	100	nan	nan	nan	nan
Timer	100	100	100	99.7	100	100	100	98.2	nan	nan	nan	nan
StairPit	100	100	100	100	100	100	100	100	nan	nan	nan	nan
PlayerScore	100	100	100	97.0	100	100	100	96.7	nan	nan	nan	nan
Scorpion	100	100	100	94.7	100	100	100	100	nan	nan	nan	nan
Waterhole	96.7	100	98.3	100	73.3	100	84.6	99.6	nan	nan	nan	nan
Crocodile	100	100	100	100	nan	nan	nan	nan	nan	nan	nan	nan
Rope	87.7	69.4	77.5	100	nan	nan	nan	nan	nan	nan	nan	nan
Snake	100	100	100	97.6	100	100	100	96.9	nan	nan	nan	nan
Tarpit	80.0	100	88.9	100	nan	nan	nan	nan	nan	nan	nan	nan

G.32 Pong details

You control the right paddle, you compete against the left paddle controlled by the computer. You each try to keep deflecting the ball away from your goal and into your opponent's goal.

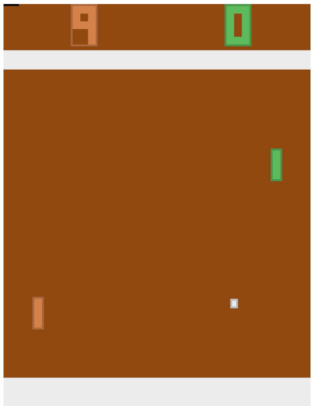


Table 38: Per class statistics on Pong

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Ball	60.2	100	75.2	74.8	76.0	100	86.4	75.2	74.0	100	85.1	74.9
Player	100	100	100	91.7	100	100	100	89.0	100	100	100	94.7
EnemyScore	100	96.9	98.4	78.0	100	95.9	97.9	79.0	100	94.7	97.3	79.2
Enemy	85.2	100	92.0	94.4	92.8	100	96.3	94.2	92.2	100	95.9	94.1
PlayerScore	100	100	100	74.3	100	100	100	84.1	100	95.1	97.5	86.5

G.33 PrivateEye details

You control the French Private Eye Pierre Touche. Navigate the city streets, parks, secret passages, dead-ends and one-ways in search of the ringleader, Henri Le Fiend and his gang. You also need to find evidence and stolen goods that are scattered about. There are five cases, complete each case before its statute of limitations expires.

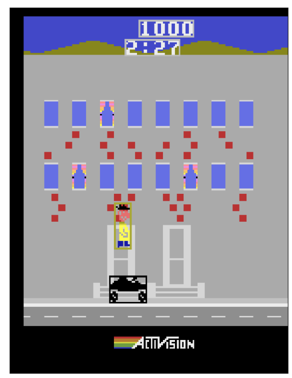


Table 39: Per class IOU on PrivateEye

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Car	96.8	100	98.4	97.3	100	100	100	94.4	nan	nan	nan	nan
Clock	100	100	100	98.5	100	100	100	98.6	nan	nan	nan	nan
Player	95.8	97.0	96.4	95.6	99.2	99.2	99.2	92.8	nan	nan	nan	nan
Score	100	100	100	97.5	100	100	100	96.7	nan	nan	nan	nan
Clue	57.5	100	73.0	77.6	50.0	100	66.7	78.6	nan	nan	nan	nan
Mud	91.2	100	95.4	100	nan	nan	nan	nan	nan	nan	nan	nan
Dove	100	100	100	100	nan	nan	nan	nan	nan	nan	nan	nan

G.34 Qbert details

You are Q*bert. Your goal is to change the color of all the cubes on the pyramid to the pyramid's 'destination' color. To do this, you must hop on each cube on the pyramid one at a time while avoiding nasty creatures that lurk there.

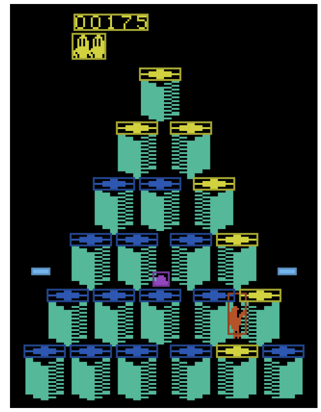


Table 40: Per class statistics on Qbert

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Cube	100	99.6	99.8	100	74.6	99.5	85.3	99.9	77.6	99.5	87.2	99.9
Score	66.0	100	79.5	100	34.8	100	51.6	98.8	31.8	100	48.3	98.9
Lives	60.1	100	75.0	100	38.9	100	56.0	100	40.5	100	57.7	100
Disk	99.8	100	99.9	100	99.6	100	99.8	100	100	100	100	100
Player	47.3	81.9	60.0	79.2	93.7	82.1	87.5	78.2	98.3	84.1	90.6	79.9
Sam	34.5	82.9	48.7	93.8	100	80.0	88.9	94.7	100	93.8	96.8	95.1
PurpleBall	20.9	52.7	29.9	90.5	71.3	69.0	70.2	91.3	76.1	66.4	70.9	90.9
Coily	91.3	100	95.5	93.4	91.3	100	95.5	89.4	87.1	100	93.1	90.7
GreenBall	nan	nan	nan	nan	83.3	90.9	87.0	87.7	79.3	88.5	83.6	87.1

G.35 Riverraid details

You control a jet that flies over a river: you can move it sideways and fire missiles to destroy enemy objects. Each time an enemy object is destroyed you score points (i.e. rewards). You lose a jet when you run out of fuel: fly over a fuel depot when you begin to run low.You lose a jet even when it collides with the river bank or one of the enemy objects (except fuel depots). The game begins with a squadron of three jets in reserve and you're given an additional jet (up to 9) for each 10,000 points you score.

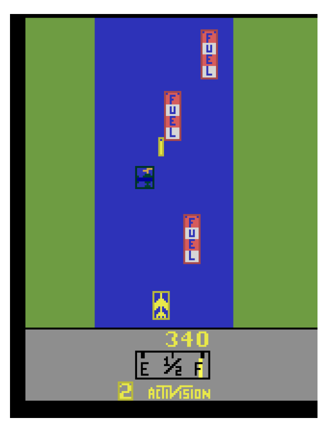


Table 41: Per class statistics on Riverraid

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
PlayerScore	100	2.4	4.7	100	100	1.0	2.0	100	nan	nan	nan	nan
FuelDepot	97.7	98.3	98.0	100	94.7	96.5	95.6	100	nan	nan	nan	nan
Tanker	97.0	98.2	97.6	96.9	94.1	95.8	95.0	94.3	nan	nan	nan	nan
Lives	89.3	99.8	94.3	89.8	59.6	100	74.7	92.0	nan	nan	nan	nan
Player	100	99.3	99.6	76.7	99.8	99.6	99.7	74.8	nan	nan	nan	nan
Helicopter	96.4	97.0	96.7	97.6	97.4	96.8	97.1	96.5	nan	nan	nan	nan
PlayerMissile	87.3	92.8	90.0	88.1	82.8	95.5	88.7	89.6	nan	nan	nan	nan
Bridge	95.2	100	97.6	82.7	97.6	97.6	97.6	84.9	nan	nan	nan	nan
Jet	nan	nan	nan	nan	95.9	100	97.9	83.6	nan	nan	nan	nan

G.36 RoadRunner details

You control the Road Runner(TM) in a race; you can control the direction to run in and times to jumps.The goal is to outrun Wile E. Coyote(TM) while avoiding the hazards of the desert.The game begins with three lives. You lose a life when the coyote catches you, picks you up in a rocket, or shoots you with a cannon. You also lose a life when a truck hits you, you hit a land mine, you fall off a cliff,or you get hit by a falling rock.You score points (i.e. rewards) by eating seeds along the road, eating steel shot, and destroying the coyote.

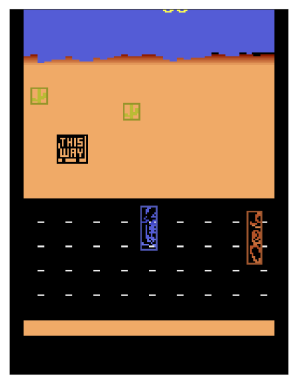


Table 42: Per class statistics on RoadRunner

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Enemy	99.3	88.6	93.7	77.1	83.3	85.3	84.3	71.7	nan	nan	nan	nan
Sign	94.4	100	97.1	97.8	54.3	100	70.4	90.9	nan	nan	nan	nan
Cactus	99.5	99.0	99.2	98.4	91.3	93.8	92.6	89.5	nan	nan	nan	nan
Bird	100	100	100	100	97.4	97.2	97.3	100	nan	nan	nan	nan
Player	95.4	96.8	96.1	79.4	93.4	98.7	96.0	91.4	nan	nan	nan	nan
BirdSeeds	55.2	90.6	68.6	78.3	62.3	89.1	73.3	63.0	nan	nan	nan	nan
Truck	nan	nan	nan	nan	77.4	100	87.3	72.0	nan	nan	nan	nan
RoadCrack	nan	nan	nan	nan	87.5	8.2	15.1	72.5	nan	nan	nan	nan
AcmeMine	nan	nan	nan	nan	16.7	41.7	23.8	69.0	nan	nan	nan	nan

G.37 Seaquest details

You control a sub able to move in all directions and fire torpedoes. The goal is to retrieve as many divers as you can, while dodging and blasting enemy subs and killer sharks; points will be awarded accordingly. The game begins with one sub and three waiting on the horizon. Each time you increase your score by 10,000 points, an extra sub will be delivered to your base. You can only have six reserve subs on the screen at one time.Your sub will explode if it collides with anything except your own divers.The sub has a limited amount of oxygen that decreases at a constant rate during the game. When the oxygen tank is almost empty, you need to surface and if you don't do it in time, your sub will blow up and you'll lose one diver. Each time you're forced to surface, with less than six divers, you lose one diver as well.

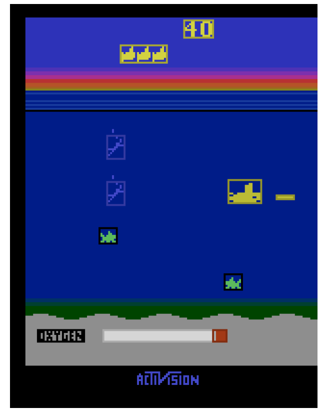


Table 43: Per class statistics on Seaquest

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
OxygenBarDepleted	92.1	100	95.9	99.8	98.6	100	99.3	100	98.4	100	99.2	99.9
Player	75.8	98.7	85.7	75.0	95.8	98.6	97.2	80.4	95.4	99.2	97.2	80.6
Logo	100	100	100	100	100	100	100	100	100	100	100	100
Lives	100	100	100	100	100	100	100	100	100	100	100	100
OxygenBarLogo	99.0	97.2	98.1	100	91.4	84.0	87.5	100	92.4	85.4	88.8	100
PlayerScore	100	84.0	91.3	92.9	94.0	79.4	86.1	93.6	84.8	60.1	70.4	79.9
OxygenBar	98.9	100	99.4	100	91.2	100	95.4	100	92.1	100	95.9	100
Diver	90.6	98.2	94.3	76.9	93.8	100	96.8	77.4	94.1	99.4	96.7	79.0
PlayerMissile	70.0	100	82.4	100	67.8	100	80.8	100	69.1	100	81.7	100
Enemy	97.8	98.9	98.3	55.8	57.8	90.6	70.6	51.5	75.6	87.5	81.1	52.7
EnemyMissile	94.3	44.0	60.0	71.7	88.9	85.1	87.0	70.9	93.2	71.4	80.9	70.9
CollectedDiver	100	100	100	100	100	100	100	100	91.8	100	95.7	100
EnemySubmarine	95.9	100	97.9	77.6	98.1	100	99.0	75.1	94.3	96.1	95.2	70.7

G.38 Skiing details

You control a skier who can move sideways. The goal is to run through all gates (between the poles) in the fastest time. You are penalized five seconds for each gate you miss. If you hit a gate or a tree, your skier will jump back up and keep going.

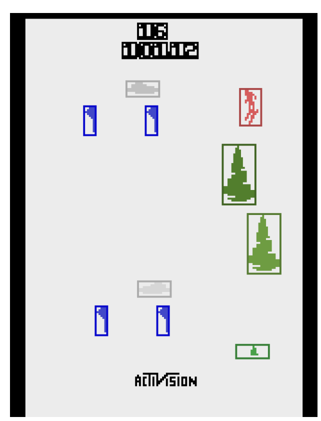


Table 44: Per class statistics on Skiing

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Score	100	100	100	100	100	100	100	100	nan	nan	nan	nan
Mogul	97.2	99.2	98.2	84.0	100	100	100	85.7	nan	nan	nan	nan
Tree	81.0	83.5	82.2	75.0	51.1	51.3	51.2	48.3	nan	nan	nan	nan
Logo	100	100	100	100	100	100	100	100	nan	nan	nan	nan
Clock	100	100	100	100	100	100	100	100	nan	nan	nan	nan
Player	99.2	96.7	97.9	64.0	100	100	100	63.9	nan	nan	nan	nan
Flag	95.8	97.4	96.6	86.0	100	100	100	86.7	nan	nan	nan	nan

G.39 SpaceInvaders details

Your objective is to destroy the space invaders by shooting your laser cannon at them before they reach the Earth. The game ends when all your lives are lost after taking enemy fire, or when they reach the earth.

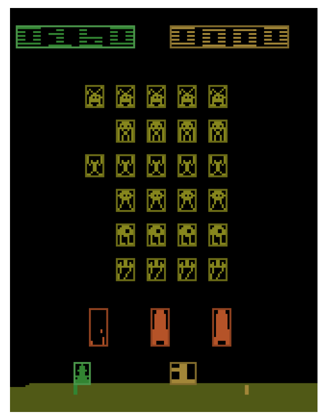


Table 45: Per class statistics on SpaceInvaders

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Shield	98.9	98.8	98.8	90.7	100	88.3	93.8	97.6	nan	nan	nan	nan
Score	79.0	100	88.3	100	65.3	100	79.0	100	nan	nan	nan	nan
Lives	76.8	79.1	77.9	100	73.3	70.2	71.7	100	nan	nan	nan	nan
Player	93.4	100	96.6	91.6	94.4	100	97.1	91.9	nan	nan	nan	nan
Alien	100	99.6	99.8	98.3	100	98.4	99.2	99.1	nan	nan	nan	nan
Bullet	33.5	66.1	44.5	79.2	31.9	64.1	42.6	76.8	nan	nan	nan	nan
Satellite	97.3	100	98.6	93.4	100	100	100	92.2	nan	nan	nan	nan

G.40 Tennis details

You control the orange player playing against a computer-controlled blue player. The game follows the rules of tennis. The first player to win at least 6 games with a margin of at least two games wins the match. If the score is tied at 6-6, the first player to go 2 games up wins the match.

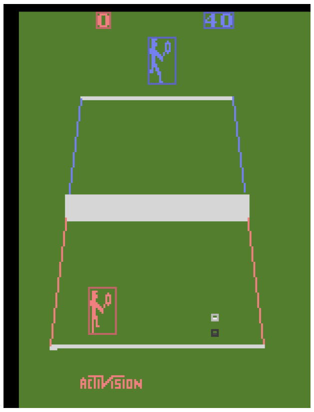


Table 46: Per class statistics on Tennis

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Logo	100	100	100	100	100	100	100	100	nan	nan	nan	nan
EnemyScore	100	100	100	98.7	100	100	100	100	nan	nan	nan	nan
BallShadow	95.5	98.3	96.9	65.3	96.1	97.1	96.6	50.5	nan	nan	nan	nan
Ball	95.2	100	97.6	70.9	95.0	100	97.4	70.3	nan	nan	nan	nan
Enemy	99.0	100	99.5	73.1	87.6	100	93.4	72.1	nan	nan	nan	nan
Player	97.8	100	98.9	71.7	83.0	100	90.7	69.9	nan	nan	nan	nan
PlayerScore	100	100	100	100	97.0	94.2	95.6	97.4	nan	nan	nan	nan

G.41 TimePilot details

You control an aircraft. Use it to destroy your enemies. As you progress in the game, you encounter enemies with technology that is increasingly from the future.

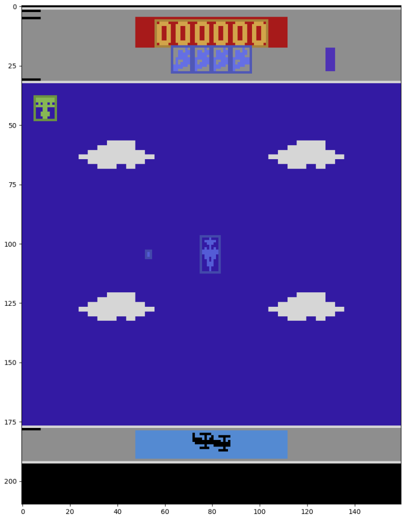


Table 47: Per class IOU on TimePilot

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Player	95.4	99.3	97.3	95.8	88.5	99.0	93.5	94.6	nan	nan	nan	nan
Player_Shot	91.8	95.6	93.7	98.8	83.9	94.6	88.9	99.1	nan	nan	nan	nan
Enemy_Green	89.1	98.7	93.7	94.7	76.9	100	86.9	88.4	nan	nan	nan	nan
Score	92.5	86.0	89.1	96.1	91.8	84.6	88.1	95.7	nan	nan	nan	nan
Life	100	100	100	100	100	99.9	99.9	100	nan	nan	nan	nan
Enemy_Green_Shot	61.5	34.3	44.0	63.1	55.6	26.3	35.7	73.3	nan	nan	nan	nan
Enemy_Black	94.4	92.4	93.4	96.0	90.6	92.7	91.6	92.2	nan	nan	nan	nan
Enemy_Black_Shot	73.3	45.8	56.4	69.7	80.0	57.1	66.7	83.3	nan	nan	nan	nan
Enemy_Yellow	nan	nan	nan	nan	93.2	93.2	93.2	88.0	nan	nan	nan	nan
Enemy_Yellow_Shot	nan	nan	nan	nan	80.0	63.2	70.6	40.3	nan	nan	nan	nan
Enemy_Blue	nan	nan	nan	nan	95.8	91.3	93.5	92.3	nan	nan	nan	nan
Enemy_Blue_Shot	nan	nan	nan	nan	0.0	0.0	0.0	60.0	nan	nan	nan	nan

G.42 UpNDown details

Your goal is to steer your baja bugger to collect prizes and eliminate opponents.

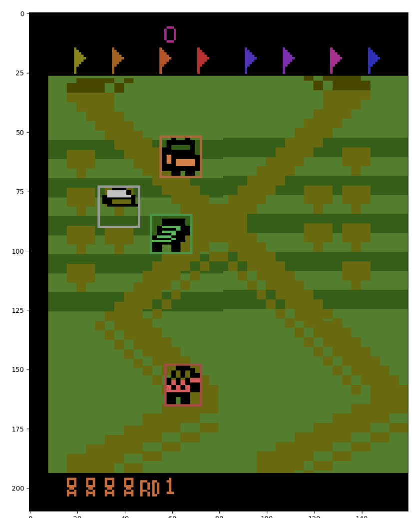


Table 48: Per class IOU on UpNDown

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Score	100	100	100	98.9	100	100	100	98.4	99.8	99.8	99.8	98.1
Life	100	99.8	99.9	100	99.9	99.8	99.9	100	100	100	100	100
HUD_Flag	100	100	100	100	100	100	100	100	100	100	100	100
Player	91.1	94.4	92.7	86.1	78.6	88.4	83.2	75.1	88.0	90.7	89.3	82.8
Truck	84.1	95.2	89.3	86.8	86.0	93.5	89.6	82.0	81.9	91.7	86.5	80.9
Flag	48.3	100	65.1	64.8	46.2	98.8	63.0	86.3	36.0	100	53.0	74.9
Collectable	100	100	100	50.7	70.4	100	82.6	79.2	62.5	90.9	74.1	59.1

G.43 Venture details

Your goal is to capture the treasure in every chamber of the dungeon while eliminating the monsters.

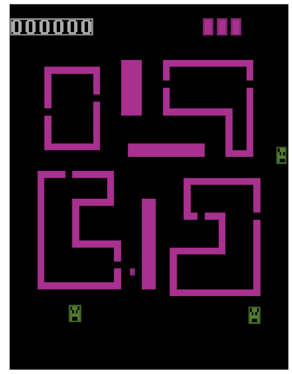


Table 49: Per class IOU on Venture

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Life	100	100	100	100	100	100	100	100	nan	nan	nan	nan
Hallmonsters	46.2	99.6	63.2	84.1	26.7	98.3	42.0	84.6	nan	nan	nan	nan
Player	93.0	99.6	96.2	99.3	54.0	99.6	70.0	98.4	nan	nan	nan	nan
Score	100	100	100	100	100	100	100	100	nan	nan	nan	nan
Goblin	50.0	100	66.7	100	nan	nan	nan	nan	nan	nan	nan	nan
Shot	50.0	100	66.7	50.0	66.7	100	80.0	100	nan	nan	nan	nan
Yellow_Collectable	50.0	100	66.7	100	nan	nan	nan	nan	nan	nan	nan	nan
Skeleton	nan	nan	nan	nan	44.4	100	61.5	100	nan	nan	nan	nan
Purple_Collectable	nan	nan	nan	nan	66.7	100	80.0	100	nan	nan	nan	nan

G.44 VideoPinball details

Your goal is to keep the ball in play as long as possible and to score as many points as possible.

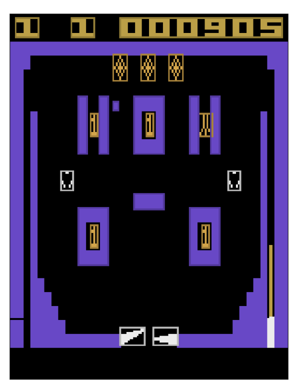


Table 50: Per class IOU on VideoPinball

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Score	84.8	75.8	80.1	91.1	97.2	95.3	96.2	99.0	nan	nan	nan	nan
DropTarget	97.8	86.5	91.8	94.2	99.6	88.6	93.8	88.3	nan	nan	nan	nan
LifeUsed	98.0	100	99.0	100	99.6	100	99.8	100	nan	nan	nan	nan
DifficultyLevel	98.0	100	99.0	100	99.6	100	99.8	100	nan	nan	nan	nan
Spinner	98.0	99.7	98.8	76.1	99.6	100	99.8	76.5	nan	nan	nan	nan
Flipper	98.5	99.4	98.9	74.2	99.6	100	99.8	72.7	nan	nan	nan	nan
Ball	90.0	100	94.7	99.9	98.6	100	99.3	100	nan	nan	nan	nan
Bumper	98.0	100	99.0	99.6	99.6	100	99.8	99.9	nan	nan	nan	nan

G.45 YarsRevenge details

The objective is to break a path through the shield and destroy the Qotile with a blast from the Zorlon Cannon.

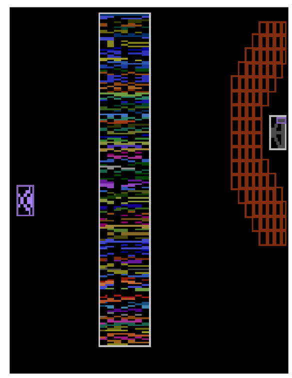


Table 51: Per class IOU on YarsRevenge

	Random				DQN				C51			
	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU	Pr	Rec	F-sc	IOU
Player	89.1	96.5	92.7	88.2	50.0	100	66.7	46.7	nan	nan	nan	nan
Barrier	40.9	94.1	57.1	100	50.0	100	66.7	100	nan	nan	nan	nan
Shield_Block	45.5	94.3	61.4	82.0	21.8	100	35.8	77.8	nan	nan	nan	nan
Enemy	98.2	88.7	93.2	98.6	100	100	100	100	nan	nan	nan	nan
Enemy_Missile	34.8	99.4	51.5	93.3	0.0	nan	0.0	nan	nan	nan	nan	nan
Swirl	65.0	92.9	76.5	63.7	nan	nan	nan	nan	nan	nan	nan	nan

H Common Mistakes in Extracting and Detecting Objects

Figure 6: Animation and errors in the game of DemonAttack and YarsRevenge. We can see multiple particle effects and invisible objects. In the left we see the spawn animation of an enemy, i n the second image we see the death animation of the player and in the last we see the invisible shields in YarsRevenge. In all cases the objects are already detected even if it is not yet or not anymore visible to the player.

In this section, we will briefly discuss 2 common errors that can occur during detection and extraction based on the games DemonAttack and YarsRevenge.

Case 1: Particle effects. As described in Section 2, we primarily use positional information and the change of colors to identify objects in the visual detection of objects (VEM). It can happen that particle effects are incorrectly identified as objects, see Figure 6. In our RAM extraction we have defined the number and types of objects before extraction and concentrate on all game elements that are relevant for the game. Since these particle effects have no effect on the game, we deliberately do not detect them, which leads to a higher errors in F1 and IOU.

Case 2: Invisible objects. If objects disappear or appear in a game, there are various ways to realize this. The most common and simplest method, which is also used in most games, is to initialize objects only when they appear and to clear the memory when objects disappear. However, some games, such as DemonAttack or YarsRevenge (Fig. 6) use a different method. Here the objects are only set to invisible when they disappear or already exist before the objects appear. As such, these objects are also found and tracked by our REM method at an early stage, even though they have not yet appeared, which leads to an increased error. In many games we have therefore tried to find binary information about which objects are active so that those that are not, are not detected. This helps to minimize the error and increase the scores, as you can see in the updates scores in DemonAttack.

I Difference between AtariARI and OCAtari

Game	Objects (AtariARI)	Objects (OCAtari)
Asterix	Enemies, Player, Lives, Score, Missiles	Enemies, Player, Lives, Score, Missiles
Berzerk	Player, Missiles, Lives, **Killcount**, Level, **Evil Otto**, Enemies	**Logo**, Player, Missiles, Enemies, Score, **RoomCleared**
Bowling	Ball, Player, **FrameNumber**, Pins, Score	Pins, Player, PlayerScore, **PlayerRound**, **Player2Round**, Ball
Boxing	Player, Enemy, Scores, Clock	Enemy, Player, Scores, Clock, **Logo**
Breakout	Ball, Player, Blocks, Score	Player, Blocks, **Live**, Score, Ball
Freeway	Player, Score, Cars	Player, Score, Cars, **Chicken**
Frostbite	Ice blocks, Lives, Igloo, Enemies, Player, Score	Ice blocks Blue, Ice blocks White, Score, Player Lives, Igloo, Enemies
Montezumas R.	**RoomNr**, Player, Skull, Monster, Level, Lives, **ItemsInInventory**, **RoomState**, Score	Player, Lives, Skull, Barrier, Key, Score, **Rope**
MsPacman	Enemies, Player, Fruits, **GhostsCount**, **DotsEaten**, Score, Lives	Lives, Score, Player, Enemies, Fruits
Pong	Player, Enemy, Ball, Scores	Player, Enemy, Ball, Scores
PrivateEye	Player, RoomNr, Clock, Score, Dove	
Q*Bert	Player, PlayerColumn, Red Enemy, Green Enemy, Score, TileColors	Cubes/Tiles, Score, Lives, **Disks**, Player, Sam, **PurpleBall**, **Coily**, GreenBall
Riverraid	Player, Missile, FuelMeter	**Score**, FuelMeter, **Tanker**, Lives, Player, **Helicopter**, Missile, **Bridge**, **Jet**
Seaquest	Enemy, Player, EnemyMissile, PlayerMissile, Score, Lives, DiversCount	Player, Lives, **OxygenBar**, Score, **Divers**, PlayerMissile, Enemy, EnemyMissile, DiverCount
SpaceInvaders	**InvadersCount**, Score, Lives, Player, Enemies, Missiles	Score, Lives, Player, Enemies, Missiles, **Satellite**, **Shield**
Tennis	Enemy, Scores, Ball, Player	Enemy, Scores, Ball, **BallShadow**, Player, Logo

Table 52: All games, supported by both AtariARI and OCAtari with their respective object lists. Note that OCAtari returns a list of (x,y,w,h) per object and AtariARI provides the value written at a specific RAM position (x and y positions or the direct value, *e.g.* , scores and so on)

J Insufficent Information in AtariARI

Game	Reason
Battlezone[1]	Unfinished
DemonAttack	Not all Demons are spotted
Hero	Missing Enemies
Q*Bert	Some Enemies, like Coily (Snake) are missing
Skiing[1]	Unfinished
RiverRaid	Important Elements (see above) are missing
Seaquest	Oxygenbar, Divers are missing
SpaceInvaders	Shields are missing

Table 53: In Table 3 some games are marked with a $\sim$ to show that the RAM information provided by AtariARI are insufficient. This table gives a short reason while we marked each game.

K REM vs VEM: Speed performance

The following graph shows that we the RAM Extraction Method of OCAtari is, in average, $50\times$ computationally more efficient than the Vision Extraction method.

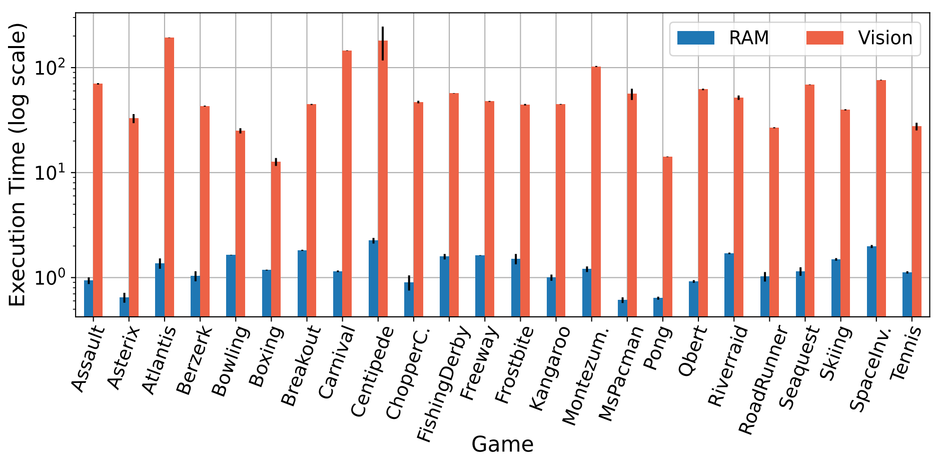


Figure 7: **Using the RAM extraction procedures leads to $50\times$ faster environments.** The average time needed to perform 10^4 steps in each OCAtari game, using RAM extraction (REM), and our vision extraction (VEM).

[1]The games appear in the Github for AtariARI, but not in the associated publication (Anand et al., 2019). Also, the information does not seem sufficient to play with them alone so we did not indicate these games in Table 3 at all.