Arcade Learning Environment

What is it?

- Developed by Marc Bellemare, Yavar Naddaf, Joel Veness and Michael Bowling.
- Object Oriented Framework -> develop AI agents for Atari 2600 games
- Built on top of Stella
- Evaluating the development of AI

Features

- Supports adding agents and games
- Fast emulation with minimal library dependencies
- Game score and end-of-game signal for 50+ games
- Multi-platform code
- Allows cross-language development
- Visualisation tools

Atari 2600

- Second generation game console (1977)
- 500+ games developed
    - shooters, beat’s ups, puzzles, sports, action-adventure.
    - Adventure and Pitfall!, Pac-Man, Space Invaders…
- 1.19 Mhz CPU, 2-4kB ROM, 128 bytes RAM

Atari 2600

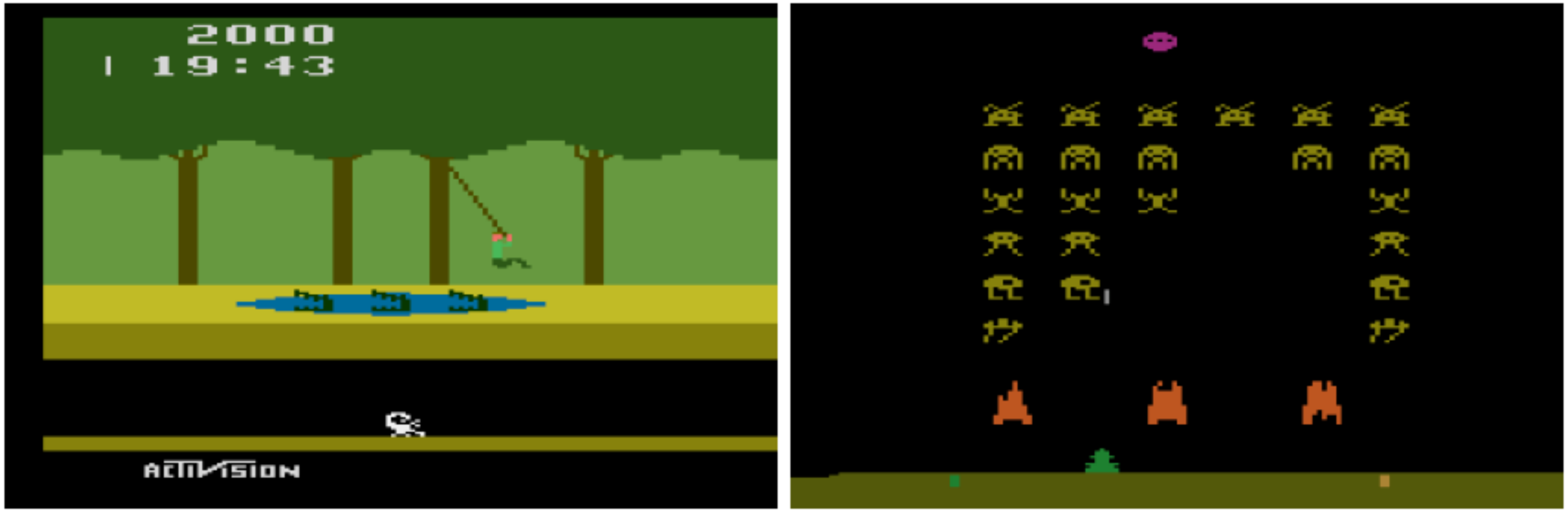


Figure 1: Screenshots of PITFALL! and SPACE INVADERS.

- Screen: 160x210px, 128-colour palette
- Controller:18 actions, 3 joystick positions for each axis, 1 button.

ALE - Stella

- Stella receives joystick motions, sends the screen & RAM info, and emulates the platform.

- ALE provides game handling layer, transforms the game into a standard RL problem by identifying the game score and end-of-game.

ALE - Stella

- Observations (per frame):
  - 2D array of 7-bit px
- Game handling layer:
  - min set of actions needed for a game
- Reward: game by game.

Source code

- Free, open-source under the terms of the GNU General Public License.
    - http://arcadelearningenvironment.org
- Written in C++
- Support for new games is added by implementing a derived class representing the games reward and termination functions.

Experimental Methodology

- Training games:
  - For domain representation and parametrization
  - Small set of games
- Testing games:
  - Unseen.

Benchmarks

- Reinforcement Learning
  - agent does not have access to a model of game dynamics.
  - SARSA(λ)

Benchmarks: SARSA(λ)

- Sets of Features:
  - Basic: Encodes presence of 128 colours.
  - BASS: Encodes with 8-colour combination.
  - DISCO: Detects objects in screen.
  - LSH: maps raw atari screens.
  - RAM: directly observes the 1024 bits of memory.

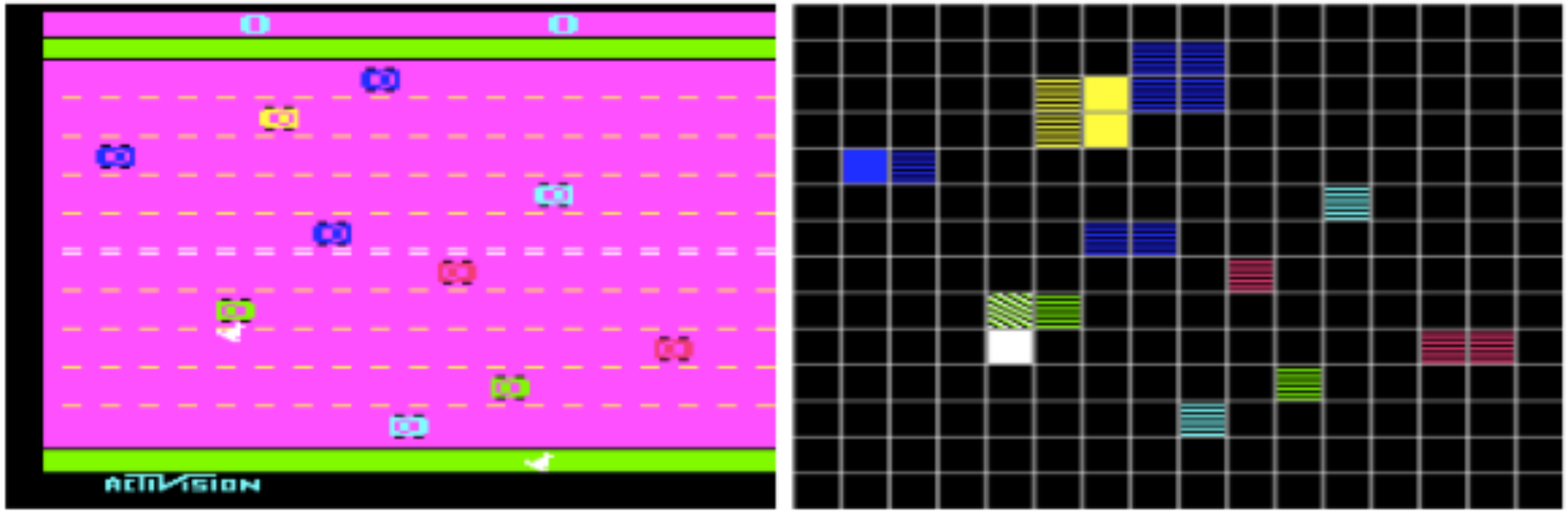


Figure 5: **Left:** Freeway in SECAM colours. **Right:** BASS colour encoding for the same screen.

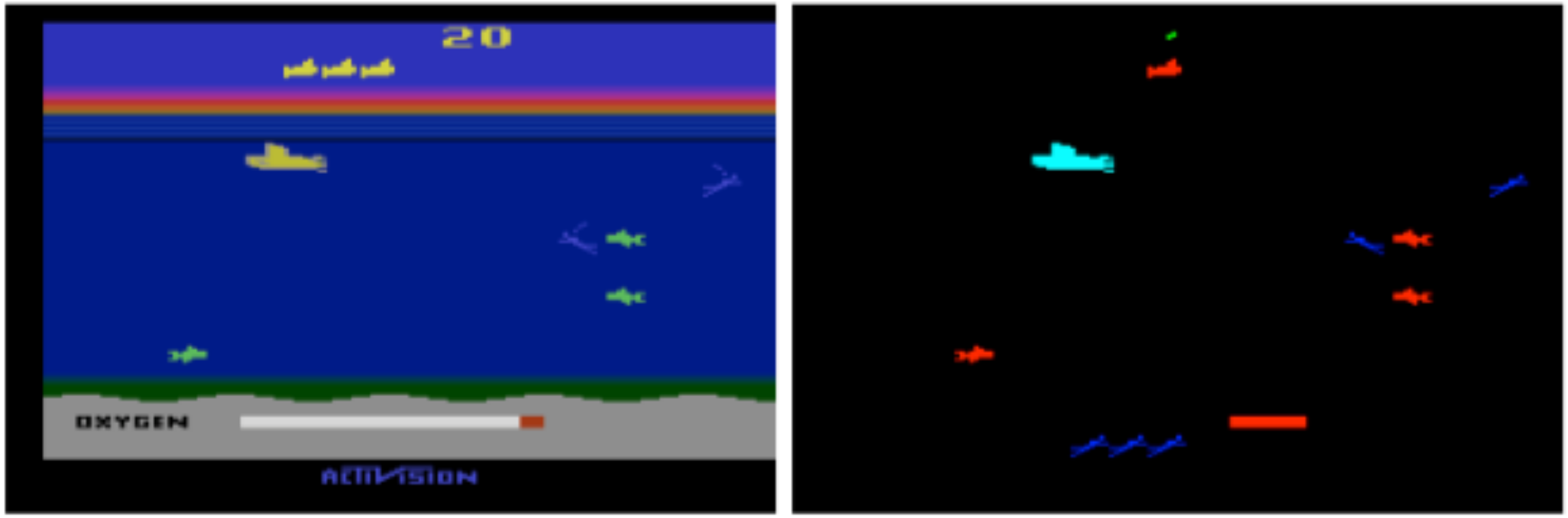
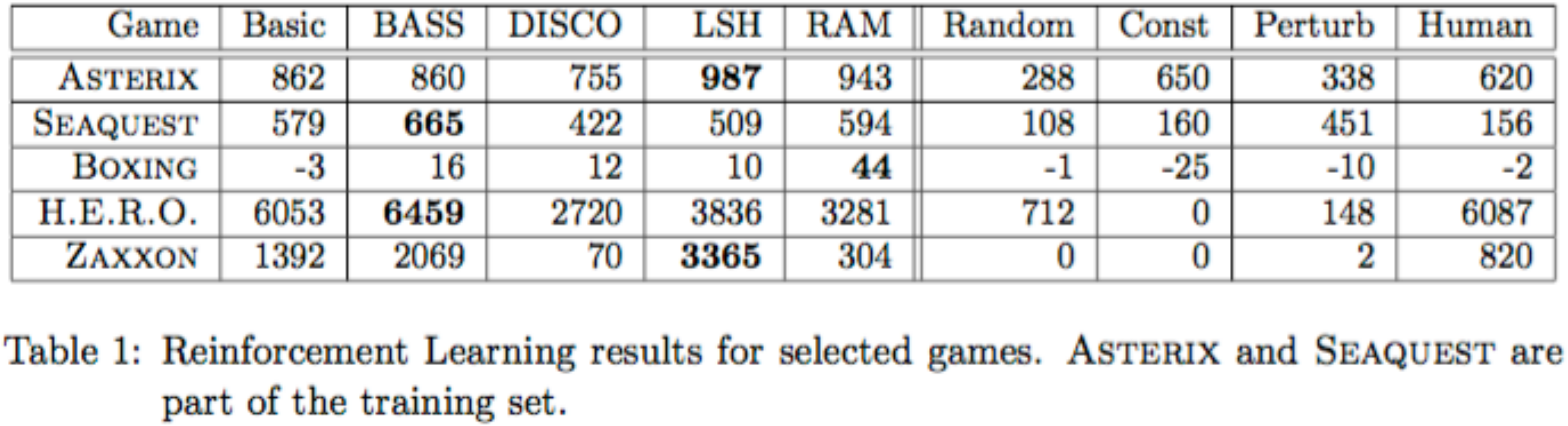


Figure 7: **Left:** Screenshot of the game SEAQUEST. **Right:** Objects detected by DISCO in the game Seaquest. Each colour represents a different class.

Benchmarks: SARSA(λ)

- Training:
  - Games: Asterix, Beam Rider, Freeway, Seaquest, and Space Invaders.
  - Parameters: learning rate, exploration rate, discount factor, decay rate.
- Testing:
  - 50 games

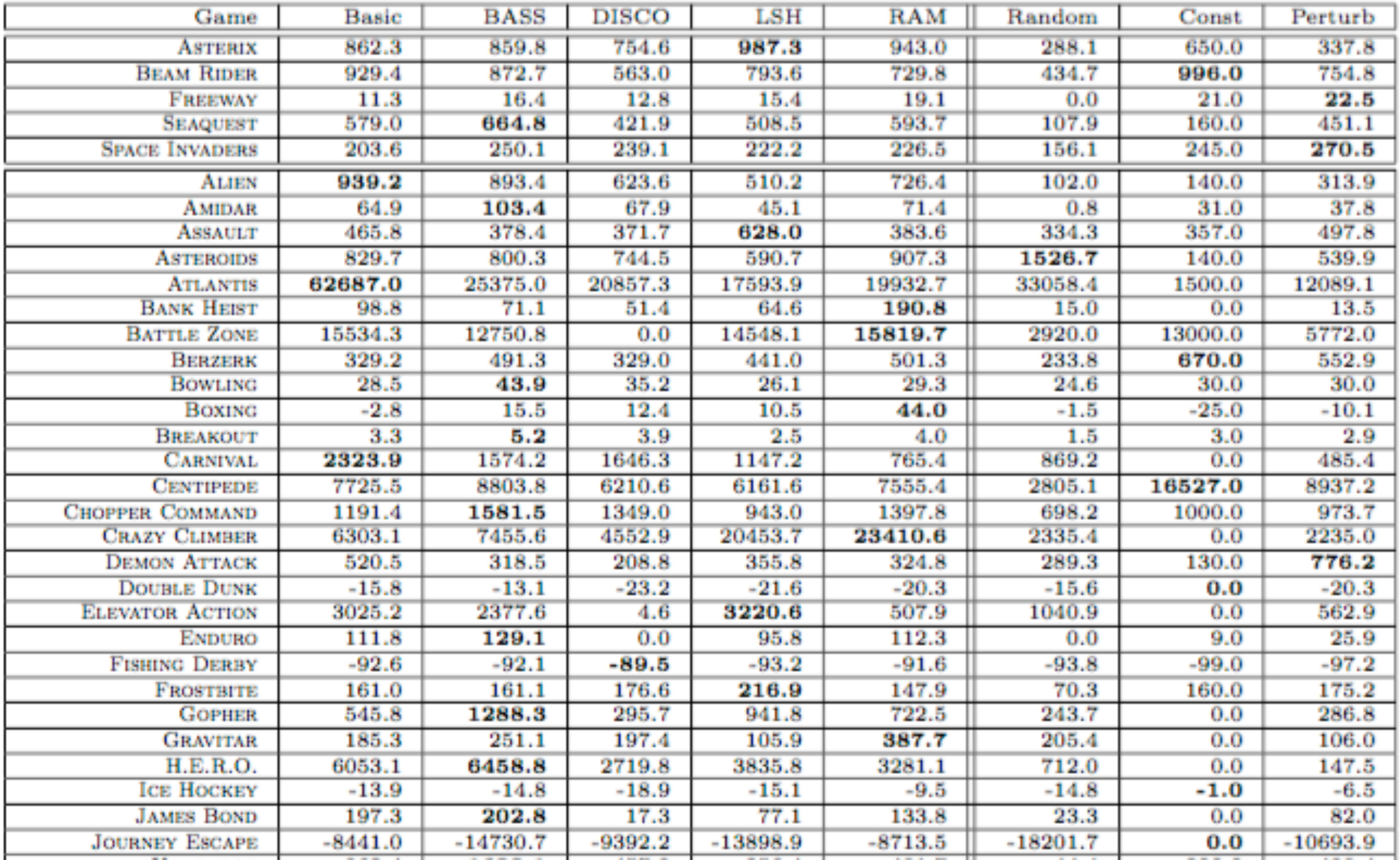
Benchmarks: SARSA(λ)



Game	Basic	BASS	DISCO	LSH	RAM	Random	Const	Perturb	Human
Asterix	862	860	755	**987**	943	288	650	338	620
Seaquest	579	**665**	422	509	594	108	160	451	156
Boxing	-3	16	12	10	**44**	-1	-25	-10	-2
H.E.R.O.	6053	**6459**	2720	3836	3281	712	0	148	6087
Zaxxon	1392	2069	70	**3365**	304	0	0	2	820

Table 1: Reinforcement Learning results for selected games. Asterix and Seaquest are part of the training set.

Benchmarks: SARSA(λ)

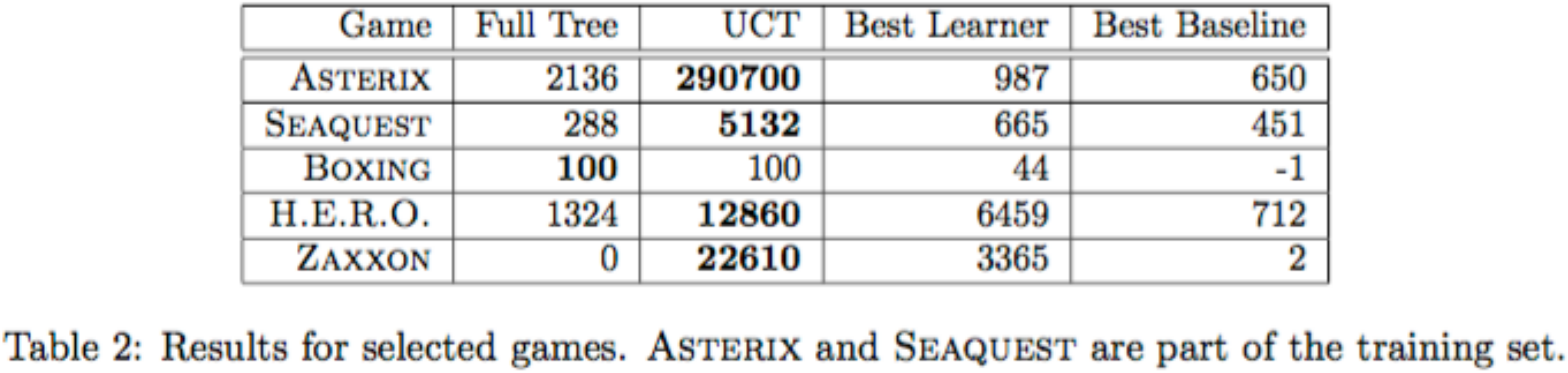


Game	Basic	BASS	DISCO	LSH	RAM	Random	Const	Perturb
Asterix	862.3	859.8	754.6	**987.3**	943.0	288.1	650.0	337.8
Beam Rider	929.4	872.7	563.0	793.6	729.8	434.7	**996.0**	754.8
Freeway	11.3	16.4	12.8	15.4	19.1	0.0	21.0	**22.5**
Seaquest	579.0	**664.8**	421.9	508.5	593.7	107.9	160.0	451.1
Space Invaders	203.6	250.1	239.1	222.2	226.5	156.1	245.0	**270.5**
Alien	**939.2**	893.4	623.6	510.2	726.4	102.0	140.0	313.9
Amidar	64.9	**103.4**	67.9	45.1	71.4	0.8	31.0	37.8
Assault	465.8	378.4	371.7	**628.0**	383.6	334.3	357.0	497.8
Asteroids	829.7	800.3	744.5	590.7	907.3	**1526.7**	140.0	539.9
Atlantis	**62687.0**	25375.0	20857.3	17593.9	19932.7	33058.4	1500.0	12089.1
Bank Heist	98.8	71.1	51.4	64.6	**190.8**	15.0	0.0	13.5
Battle Zone	15534.3	12750.8	0.0	14548.1	**15819.7**	2920.0	13000.0	5772.0
Berzerk	329.2	491.3	329.0	441.0	501.3	233.8	**670.0**	552.9
Bowling	28.5	**43.9**	35.2	26.1	29.3	24.6	30.0	30.0
Boxing	-2.8	15.5	12.4	10.5	**44.0**	-1.5	-25.0	-10.1
Breakout	3.3	**5.2**	3.9	2.5	4.0	1.5	3.0	2.9
Carnival	**2323.9**	1574.2	1646.3	1147.2	765.4	869.2	0.0	485.4
Centipede	7725.5	8803.8	6210.6	6161.6	7555.4	2805.1	**16527.0**	8937.2
Chopper Command	1191.4	**1581.5**	1349.0	943.0	1397.8	698.2	1000.0	973.7
Crazy Climber	6303.1	7455.6	4552.9	20453.7	**23410.6**	2335.4	0.0	2235.0
Demon Attack	520.5	318.5	208.8	355.8	324.8	289.3	130.0	**776.2**
Double Dunk	-15.8	-13.1	-23.2	-21.6	-20.3	-15.6	**0.0**	-20.3
Elevator Action	3025.2	2377.6	4.6	**3220.6**	507.9	1040.9	0.0	562.9
Enduro	111.8	**129.1**	0.0	95.8	112.3	0.0	9.0	25.9
Fishing Derby	-92.6	-92.1	**-89.5**	-93.2	-91.6	-93.8	-99.0	-97.2
Frostbite	161.0	161.1	176.6	**216.9**	147.9	70.3	160.0	175.2
Gopher	545.8	**1288.3**	295.7	941.8	722.5	243.7	0.0	286.8
Gravitar	185.3	251.1	197.4	105.9	**387.7**	205.4	0.0	106.0
H.E.R.O.	6053.1	**6458.8**	2719.8	3835.8	3281.1	712.0	0.0	147.5
Ice Hockey	-13.9	-14.8	-18.9	-15.1	-9.5	-14.8	**-1.0**	-6.5
James Bond	197.3	**202.8**	17.3	77.1	133.8	23.3	0.0	82.0
Journey Escape	-8441.0	-14730.7	-9392.2	-13898.9	-8713.5	-18201.7	**0.0**	-10693.9

Benchmarks

- Planning
  - Breadth-First Search
  - Upper Confidence Bounds Applied to Trees

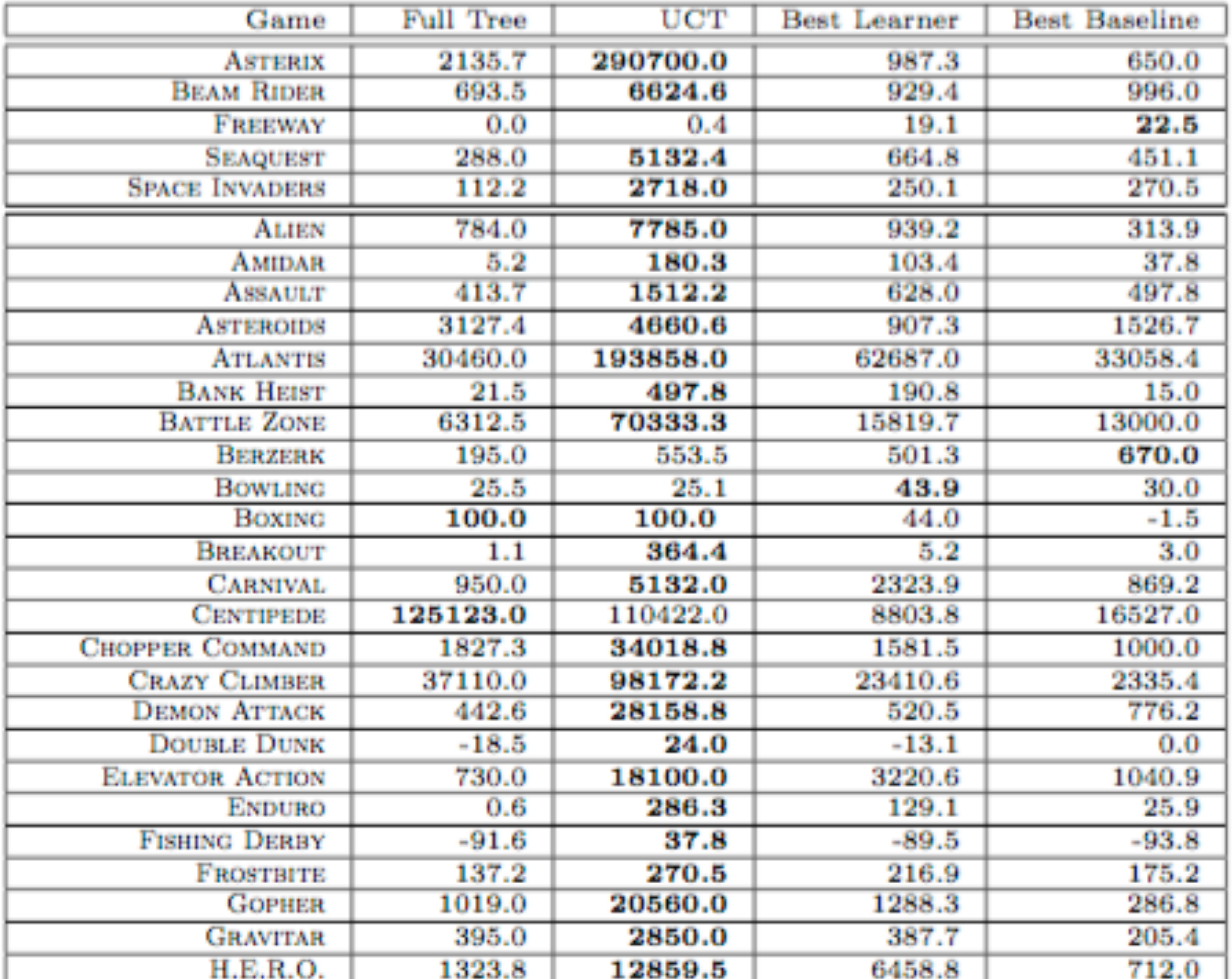
Benchmarks: Planning



Game	Full Tree	UCT	Best Learner	Best Baseline
Asterix	2136	**290700**	987	650
Seaquest	288	**5132**	665	451
Boxing	**100**	100	44	-1
H.E.R.O.	1324	**12860**	6459	712
Zaxxon	0	**22610**	3365	2

Table 2: Results for selected games. Asterix and Seaquest are part of the training set.

Benchmarks: Planning

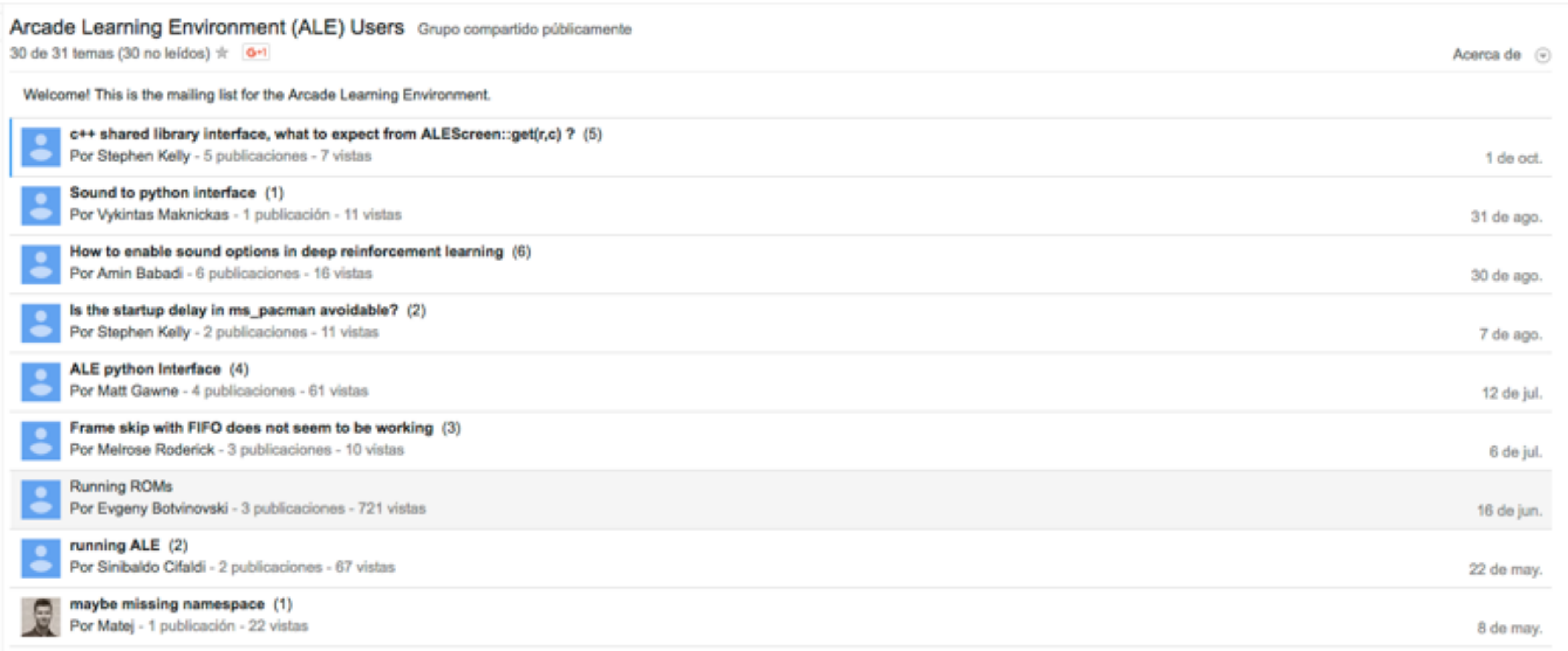


Game	Full Tree	UCT	Best Learner	Best Baseline
Asterix	2135.7	**290700.0**	987.3	650.0
Beam Rider	693.5	**6624.6**	929.4	996.0
Freeway	0.0	0.4	19.1	**22.5**
Seaquest	288.0	**5132.4**	664.8	451.1
Space Invaders	112.2	**2718.0**	250.1	270.5
Alien	784.0	**7785.0**	939.2	313.9
Amidar	5.2	**180.3**	103.4	37.8
Assault	413.7	**1512.2**	628.0	497.8
Asteroids	3127.4	**4660.6**	907.3	1526.7
Atlantis	30460.0	**193858.0**	62687.0	33058.4
Bank Heist	21.5	**497.8**	190.8	15.0
Battle Zone	6312.5	**70333.3**	15819.7	13000.0
Berzerk	195.0	553.5	501.3	**670.0**
Bowling	25.5	25.1	**43.9**	30.0
Boxing	**100.0**	**100.0**	44.0	-1.5
Breakout	1.1	**364.4**	5.2	3.0
Carnival	950.0	**5132.0**	2323.9	869.2
Centipede	**125123.0**	110422.0	8803.8	16527.0
Chopper Command	1827.3	**34018.8**	1581.5	1000.0
Crazy Climber	37110.0	**98172.2**	23410.6	2335.4
Demon Attack	442.6	**28158.8**	520.5	776.2
Double Dunk	-18.5	**24.0**	-13.1	0.0
Elevator Action	730.0	**18100.0**	3220.6	1040.9
Enduro	0.6	**286.3**	129.1	25.9
Fishing Derby	-91.6	**37.8**	-89.5	-93.8
Frostbite	137.2	**270.5**	216.9	175.2
Gopher	1019.0	**20560.0**	1288.3	286.8
Gravitar	395.0	**2850.0**	387.7	205.4
H.E.R.O.	1323.8	**12859.5**	6458.8	712.0

DEMO

https://youtu.be/nzUiEkasXZI

Mailing list



Arcade Learning Environment (ALE) Users Grupo compartido públicamente

30 de 31 temas (30 no leídos) ☆ G+1 Acerca de

Welcome! This is the mailing list for the Arcade Learning Environment.

c++ shared library interface, what to expect from ALEScreen::get(r,c) ? (5)
Por Stephen Kelly - 5 publicaciones - 7 vistas 1 de oct.

Sound to python interface (1)
Por Vykintas Maknickas - 1 publicación - 11 vistas 31 de ago.

How to enable sound options in deep reinforcement learning (6)
Por Amin Babadi - 6 publicaciones - 16 vistas 30 de ago.

Is the startup delay in ms_pacman avoidable? (2)
Por Stephen Kelly - 2 publicaciones - 11 vistas 7 de ago.

ALE python Interface (4)
Por Matt Gawne - 4 publicaciones - 61 vistas 12 de jul.

Frame skip with FIFO does not seem to be working (3)
Por Melrose Roderick - 3 publicaciones - 10 vistas 6 de jul.

Running ROMs
Por Evgeny Botvinovski - 3 publicaciones - 721 vistas 16 de jun.

running ALE (2)
Por Sinibaldo Cifaldi - 2 publicaciones - 67 vistas 22 de may.

maybe missing namespace (1)
Por Matej - 1 publicación - 22 vistas 8 de may.

- For questions and comments on ALE:
- https://groups.google.com/forum/#!forum/arcade-learning-environment

Supported ROMs

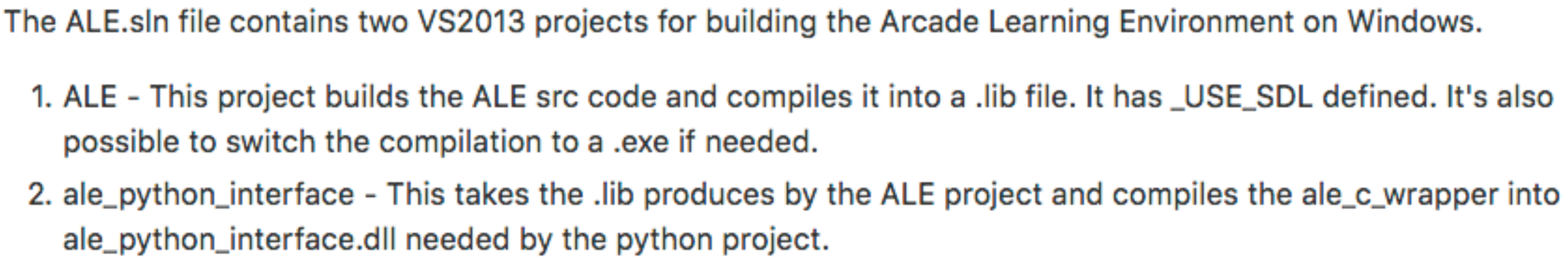
- air_raid, alien, amidar, assault, asterix, asteroids, atlantis, bank_heist, battle_zone, beam_rider, berzerk, bowling, boxing, breakout, elevator_action, enduro, fishing_derby …
- Full list and download link:
  - https://groups.google.com/forum/#!msg/arcade-learning-environment/1IL4dnLSpfo/POs4vnZhRPcJ

Communicating via FIFO Pipes

- For developing in languages different than C++.
- Sample Java agents are included in the ALE bundle.
- Examples for python and visual studio.
- http://yavar.naddaf.name/ale/#communicating-with-a-l-e-via-fifo-pipes

ALE for Visual Studio

- https://github.com/Islandman93/Arcade-Learning-Environment

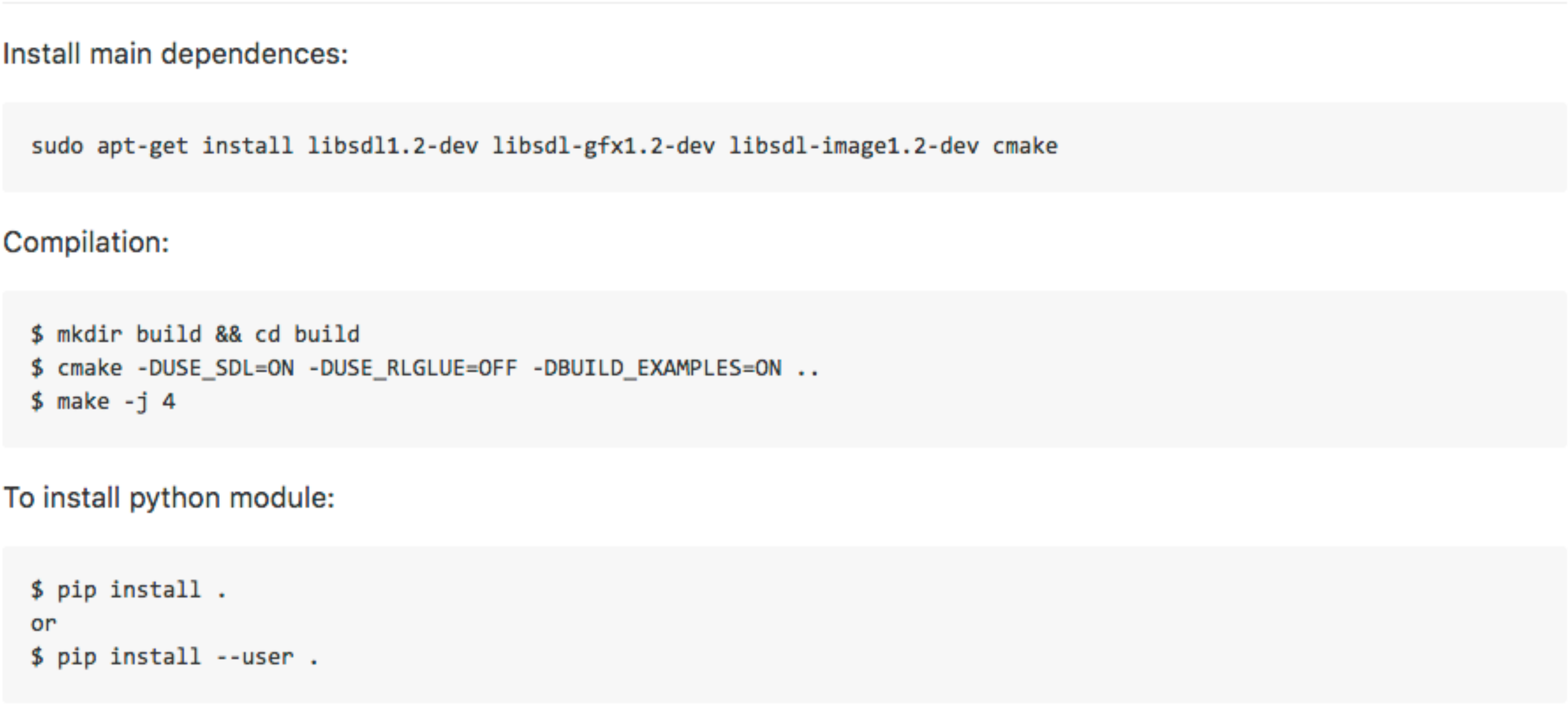


The ALE.sln file contains two VS2013 projects for building the Arcade Learning Environment on Windows.

1. ALE - This project builds the ALE src code and compiles it into a .lib file. It has _USE_SDL defined. It's also possible to switch the compilation to a .exe if needed.
2. ale_python_interface - This takes the .lib produces by the ALE project and compiles the ale_c_wrapper into ale_python_interface.dll needed by the python project.

ALE for python

- https://github.com/mgbellemare/Arcade-Learning-Environment



Install main dependences:

```
sudo apt-get install libsdl1.2-dev libsdl-gfx1.2-dev libsdl-image1.2-dev cmake
```

Compilation:

```
$ mkdir build && cd build
$ cmake -DUSE_SDL=ON -DUSE_RLGLUE=OFF -DBUILD_EXAMPLES=ON ..
$ make -j 4
```

To install python module:

```
$ pip install .
or
$ pip install --user .
```

OpenAI Gym

- Toolkit for developing and comparing RL agents.
- open-source library for Python 2.7 and 3.5.
- Supported on Linux and OSX
- It includes various environments:
    - Algorithmic, **Atari**, Board games, Box2d, Classic control, Doom, MuJoCo, Toy text.
- https://pypi.python.org/pypi/gym/0.1.4

PLE - PyGame Learning Environment

- Alternative for python mimicking ALE
- The goal of PLE is allow practitioners to focus design of models and experiments instead of environment design.
- Games:
  - Catcher, Monster Kong, FlappyBird, Pixelcopter, Pong, Puckworld, RaycastMaze, Snake, WaterWorld.
- http://pygame-learning-environment.readthedocs.io/en/latest/

RL-Glue

- Library that simplifies connecting different components to do RL research.
- Tutorial:
  - http://rofer.me/posts/teaching-a-computer-to-play-atari-p1/
  - http://rofer.me/posts/teaching-a-computer-to-play-atari-p2/
- Download link:
  - https://code.google.com/archive/p/rl-glue-ext/wikis/RLGlueCore.wiki

ALE Python interface

- Fast and easy to use python interface to ALE
- Directly loads ALE share object file
  - better performance than using the FIFO interface
- Instructions:
  - https://github.com/bbitmaster/ale_python_interface/wiki
- Download:
  - https://github.com/bbitmaster/ale_python_interface

Additional Documents

- Technical Manual:
    - http://phoenix.goucher.edu/~justin/computational_cognition/manual.pdf
- Stella
    - http://stella.sourceforge.net/

Publications

- **M. G. Bellemare, Y. Naddaf, J. Veness and M. Bowling. “The Arcade Learning Environment: An Evaluation Platform for General Agents,”. In Journal of Artificial Intelligence Research 47, pp. 253-279, 2013.**

- M. G. Bellemare, J. Veness, and M. Bowling. “Bayesian Learning of Recursively Factored Environments“. In Proceedings of the International Conference on Machine Learning, 2013.

- M. G. Bellemare, J. Veness, and M. Bowling. “Sketch-Based Linear Value Function Approximation“. In Advances in Neural Information Processing Systems 25, 2012.

- M. G. Bellemare, J. Veness, M. Bowling, “Investigating Contingency Awareness using Atari 2600 Games,” in Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12), 2012.

- Y. Naddaf, “Game-Independent AI Agents for Playing Atari 2600 Console Games,” University of Alberta, 2010.