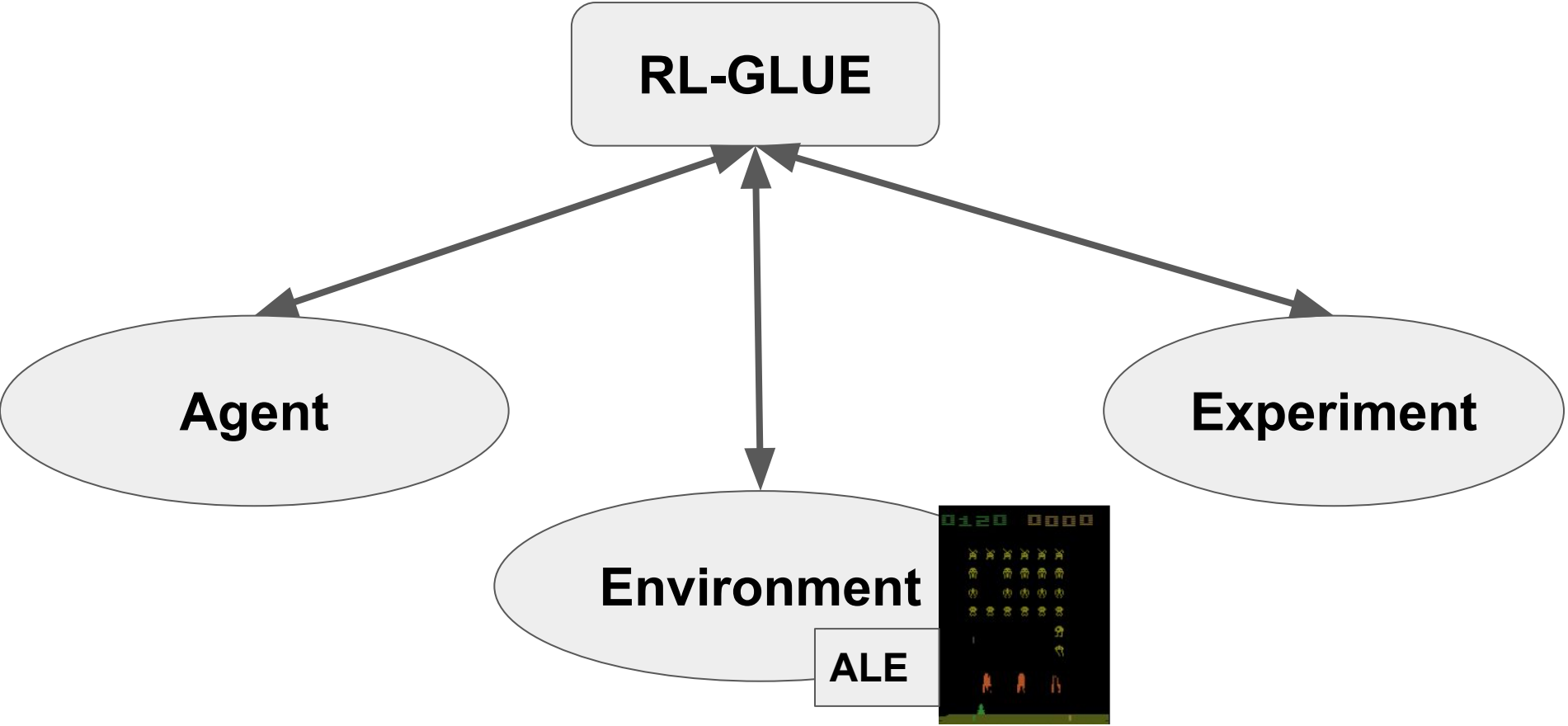
Running Arcade Learning Experiments

MAL Seminar 2015-2016

RL-Glue



Install ALE 5

- Easiest way use cmake: https://cmake.org/download/
- In source dir run: ‘

cmake -DUSE_SDL=OFF -DUSE_RLGLUE=ON -DBUILD_EXAMPLES=ON .’

- run ‘make’
- Make sure rlglue, c-codec en python codec are installed
- To be able to watch agents play you need libsdl1.2-dev (see manual) and set -DUSE_SDL to ON
- On OS X if you get ld warnings “file was built for i386”, reinstall rlglue from source (configure && make && make install)

ALE Starter Code

- You will need git (https://git-scm.com) and a github account
- Python starter code is on github:
  - https://github.com/pvrancx/PyALE
- Create your own fork of the repository:
  - https://help.github.com/articles/fork-a-repo/
- This fork should be used to do the homework tasks and the final project
- After the course project you will be asked to send a pull request to submit your code

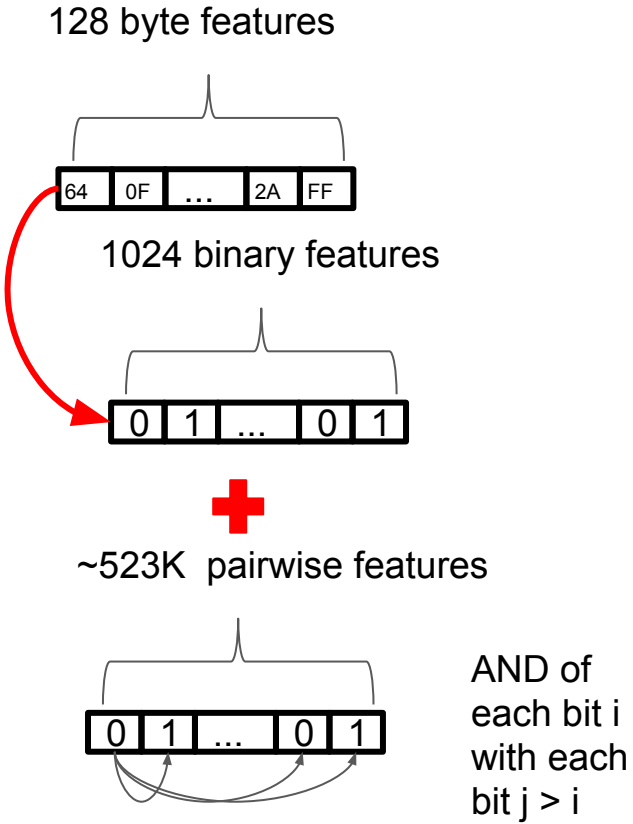
Running experiments

- ./ale -game_controller rlglue -repeat_action_probability 0.0 -frame_skip 30 roms/space_invaders.bin
- python sarsa.py --alpha 0.1 --lambda 0.5 --eps 0.05 --features RAM --actions 0 1 3 4
- python generic_experiment.py --numeps 100 --numtrials 5 --maxsteps 2000
- rl_glue

Function approximation with ALE

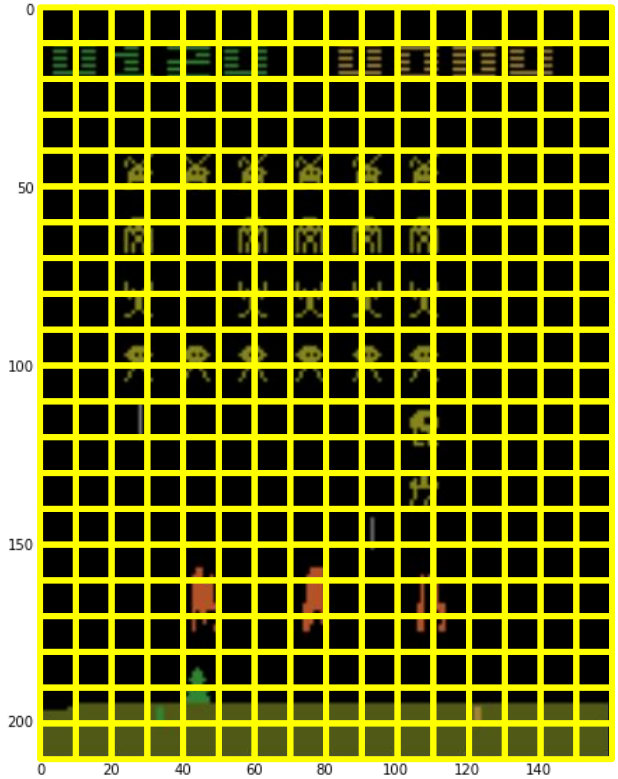
- We will look at 2 feature sets : BASIC and RAM
- Both are used for linear function approximation with binary features
- RAM is full system state, BASIC frame features are not
- Note: even for RAM values are probably not linear function of RAM bits

RAM Features



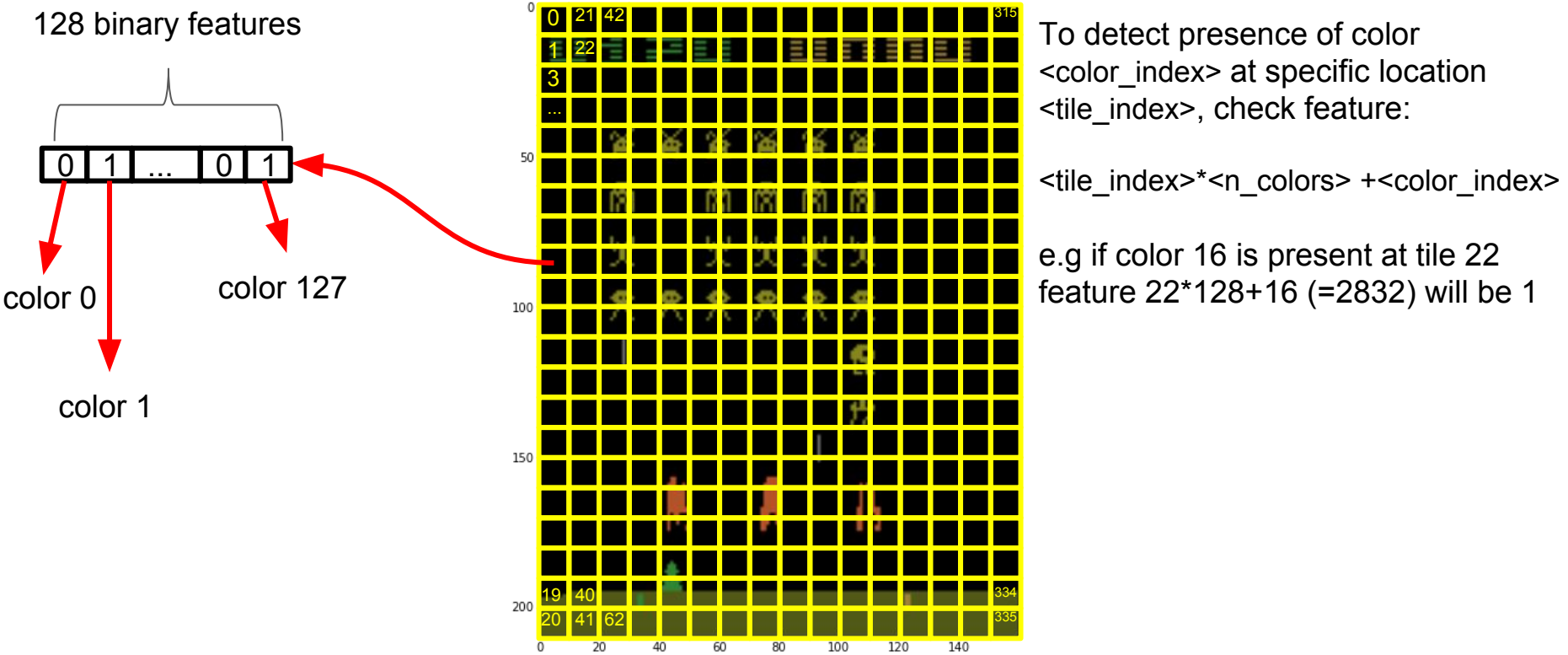
- RAM bytes are converted to binary
- Pairwise features are added (AND of every 2 bits)
- Not interpretable
- Full state

Basic Features



- Frame is tiled (default: 21x16 tiles)
- Each tile has 128 binary features
- 128*21*16 = ~43K features
- Each feature corresponds to a color
- Feature for a given tile is 1 iff corresponding color is present
- Background colors are ignored
- Can also be used with reduced color set (7 colors)

Basic Features

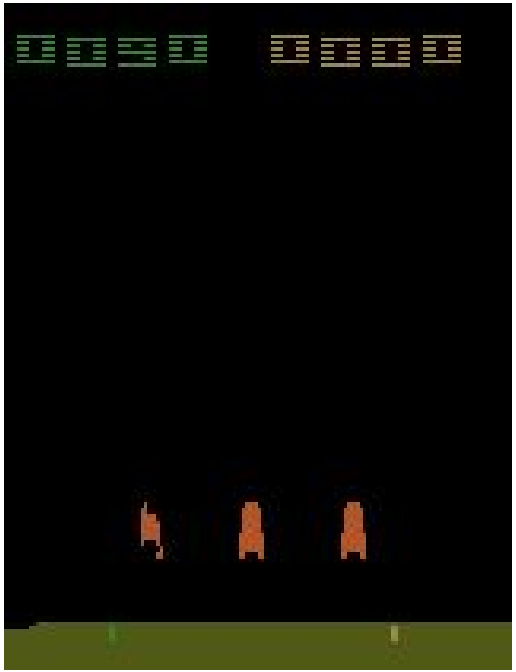


To detect presence of color <color_index> at specific location <tile_index>, check feature:

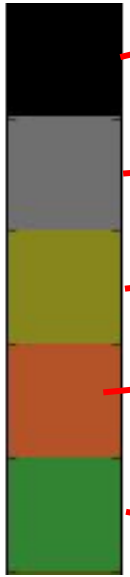
<tile_index>*<n_colors> +<color_index>

e.g if color 16 is present at tile 22 feature 22*128+16 (=2832) will be 1

Colors for space invaders



Background: presence of these colors in these locations is ignored



color 0: empty space

color 2: lasers

color 10: enemies

color 26: shield
(mostly ignored as background)

color 98: spaceship
(note that score has the same color)

Experimental settings

- Run experiments for at least 3000 episodes
- Repeat experiments at least 5 times
  - Note that a single 3000 episode run can take several hours
- Use frame_skip 30
- RAM features have longer runtime but learn faster
- Epsilon 0.05
- Run without action stochasticity
- Use reduced action set (--actions 1 2 3 option for agent) for space invaders actions are 0 1 3 4

First assignment: run SARSA(lambda) with RAM features on space invaders and plot learning curve. This will be baseline for future experiments