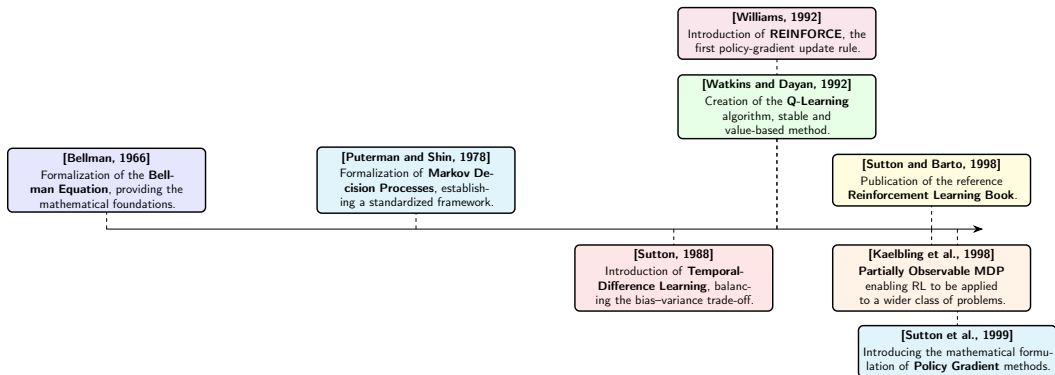
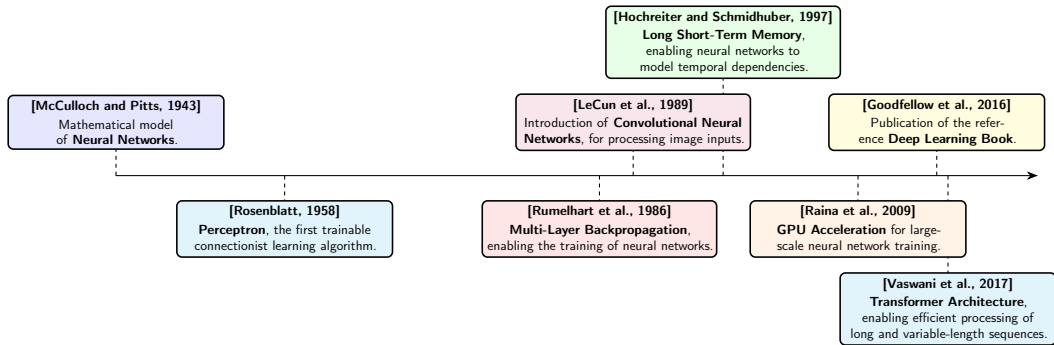
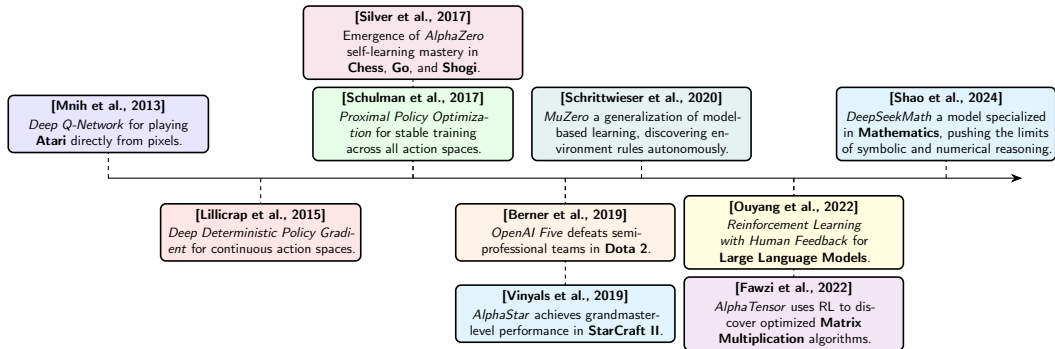


# RLlib: Industry-Grade, Scalable Reinforcement Learning

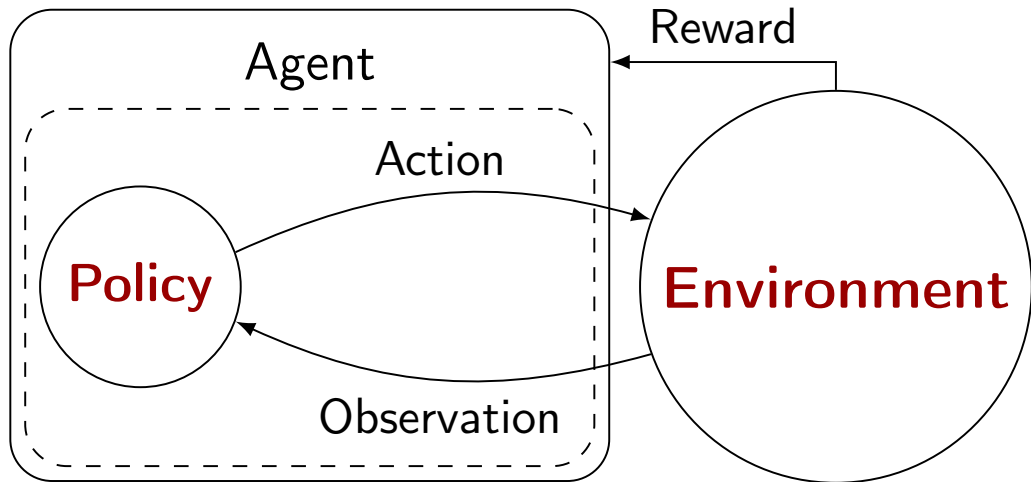
Maxime Alaarabiou







# RL Algorithm



## Gymnasium:

```
1 import gymnasium as gym
2
3 env = gym.make("ALE/SpaceInvaders-v5", render_mode="rgb_array")
4 obs, info = env.reset()
5 action = env.action_space.sample()
6 obs, reward, terminated, truncated, info = env.step(action)
```

## PettingZoo:

```
1 from pettingzoo.atari import space_invaders_v2
2
3 env = space_invaders_v2.env(render_mode="rgb_array")
4 env.reset()
5 for agent in env.agent_iter():
6     obs, reward, term, trunc, info = env.last()
7     action = env.action_space(agent).sample() if not term else None
8     env.step(action)
```

Environment Configuration	
game_name	'SpacInvaders-v5'
repeat_action_probability	0.05
frameskip	5
resize_observation_shape	(64, 64)
convert_to_grayscale	True
reward_scale_factor	0.05
frame_stack_len	4
normalize_observation	True
observation_numpy_type	np.float16
Policy Architecture Configuration	
architecture	CnnPPO
configuration_cnn	[(16,4,2),(32,4,2),(64,4,2),(128,4,2)]
configuration_hidden_layers	[512,256,128]
activation_function_class	LeakyReLU
use_layer_normalization_cnn	True
use_share_cnn	True
Reinforcement Learning Configuration	
algorithm_name	'PPO'
rollout_fragment_length	2048
train_batch_size	2048 * 8
minibatch_size	2048
lambda_gae	0.95
kullback_leibler_coefficient	0.5
clip_policy_parameter	0.1
clip_value_function_parameter	10
entropy_coefficient	0.01
number_epochs	10
learning_rate	0.00015
gradient_clip	100.0
gradient_clip_by	'global_norm'

# Why choose RLlib?

Open-source

Scalable

DQN, DDPG, PPO, Dreamer

PyTorch, TensorFlow

Automatic Plotting

Automatic Checkpointing and Restart

Multi-agent, Hierarchical

Advanced Callback System





Exploration, Curriculum Learning, Custom RL Algorithms



```
1 config = PPOConfig()
2 config.environment("SpaceInvaders")
3 config.rl_module(
4     model_config={
5         "conv_filters": [
6             [32, [8, 8], 4],
7             [64, [4, 4], 2],
8             [64, [3, 3], 1]
9         ],
10     }
11 )
12
13 algo = config.build()
14 for _ in range(5):
15     pprint(algo.train())
16
17 pprint(algo.evaluate())
18 algo.stop()
```

# Thank you for your attention

Questions, Comments, Ideas?

-  Bellman, R. (1966).  
Dynamic programming.  
[science](#), 153(3731):34–37.  
**Book.**
-  Berner, C., Brockman, G., Chan, B., Cheung, V., Dębniak, P., Dennison, C., Farhi, D.,  
et al. (2019).  
Dota 2 with large scale deep reinforcement learning.  
[arXiv preprint arXiv:1912.06680](#).  
**Article. Video.**
-  Fawzi, A., Balog, M., Huang, A., Hubert, T., Romera-Paredes, B., Barekatain, M.,  
Novikov, A., R. Ruiz, F. J., Schrittwieser, J., Swirszcz, G., et al. (2022).  
Discovering faster matrix multiplication algorithms with reinforcement learning.  
[Nature](#), 610(7930):47–53.
-  Goodfellow, I., Bengio, Y., and Courville, A. (2016).  
Deep Learning.  
MIT Press.

## Book.

-  Hochreiter, S. and Schmidhuber, J. (1997).  
Long short-term memory.  
[Neural computation](#), 9(8):1735–1780.  
**Article.**
-  Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998).  
Planning and acting in partially observable stochastic domains.  
[Artificial intelligence](#), 101(1-2):99–134.  
**Article.**
-  LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989).  
Backpropagation applied to handwritten zip code recognition.  
[Neural computation](#), 1(4):541–551.  
**Article.**
-  Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015).

Continuous control with deep reinforcement learning.

Article.



McCulloch, W. S. and Pitts, W. (1943).

A logical calculus of the ideas immanent in nervous activity.

[The bulletin of mathematical biophysics](#), 5(4):115–133.

Article.



Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013).

Playing atari with deep reinforcement learning.

Article.



Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., et al. (2022).

Training language models to follow instructions with human feedback.

[Advances in Neural Information Processing Systems](#), 35:27730–27744.

Article.



Puterman, M. L. and Shin, M. C. (1978).

Modified policy iteration algorithms for discounted markov decision problems.  
[Management Science](#), 24(11):1127–1137.

**Article.**



Raina, R., Madhavan, A., and Ng, A. Y. (2009).

Large-scale deep unsupervised learning using graphics processors.

In [Proceedings of the 26th annual international conference on machine learning](#), pages 873–880.

**Article.**



Rosenblatt, F. (1958).

The perceptron: a probabilistic model for information storage and organization in the brain.

[Psychological review](#), 65(6):386.

**Article.**



Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986).

Learning representations by back-propagating errors.

[nature](#), 323(6088):533–536.


**Article.**

 Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T., and Silver, D. (2020).

Mastering atari, go, chess and shogi by planning with a learned model.


[Nature](#), 588(7839):604–609.

[Article](#).

 Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017).  
Proximal policy optimization algorithms.

[arXiv preprint arXiv:1707.06347](#).

[Article](#).

 Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. (2024).

Deepseekmath: Pushing the limits of mathematical reasoning in open language models.

[Article](#).

 Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., and Hassabis, D. (2017).

Mastering chess and shogi by self-play with a general reinforcement learning algorithm.

Article.

 Sutton, R. S. (1988).

Learning to predict by the methods of temporal differences.

Machine learning, 3(1):9–44.

Article.

 Sutton, R. S. and Barto, A. G. (1998).

Reinforcement Learning: An Introduction.

MIT Press.

Book.


 Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999).

Policy gradient methods for reinforcement learning with function approximation.



Advances in neural information processing systems, 12.

**Article.**

 Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017).

Attention is all you need.

Advances in neural information processing systems, 30.

**Article.**

 Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019).

Grandmaster level in starcraft ii using multi-agent reinforcement learning.

nature, 575(7782):350–354.

 Watkins, C. J. and Dayan, P. (1992).

Q-learning.

Machine learning, 8(3):279–292.

**Article.**

 Williams, R. J. (1992).

Simple statistical gradient-following algorithms for connectionist reinforcement learning.

[Machine learning](#), 8(3):229–256.