

Pokémon Red via Reinforcement Learning

IEEE Conference on Games (CoG) 2025

Marco Pleines — TU Dortmund University, Dortmund, Germany

Daniel Addis — Independent Researcher, Minneapolis, MN, USA

David Rubinstein — Independent Researcher, Brooklyn, NY, USA

Frank Zimmer — Rhine-Waal University of Applied Sciences, Kamp-Linfort, Germany

Mike Preuss — LIACS Universiteit Leiden, Leiden, Netherlands

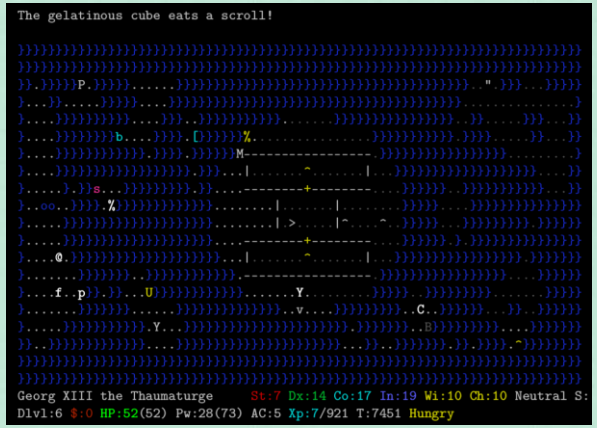
Peter Whidden — Independent Researcher, Brooklyn, NY, USA

Reinforcement Learning (RL) has been used to play games like Chess, NetHack, DotA2 and even StarCraft.



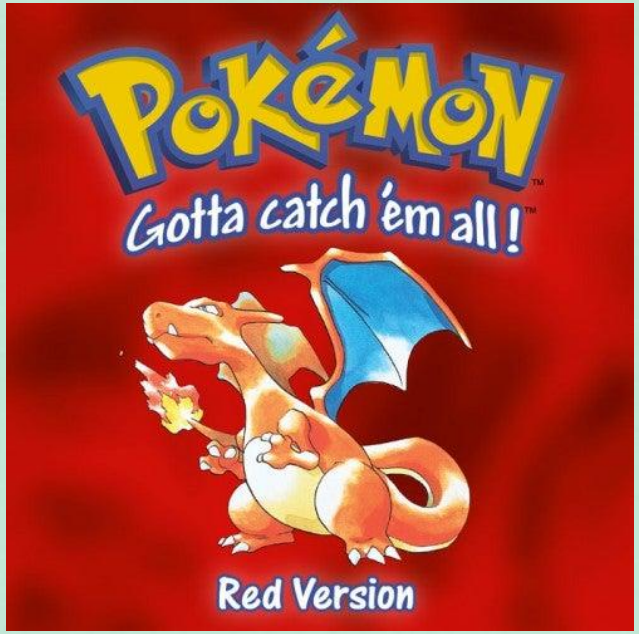


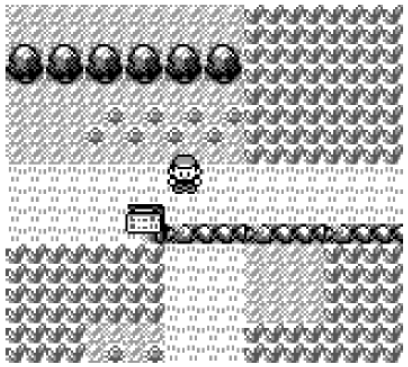




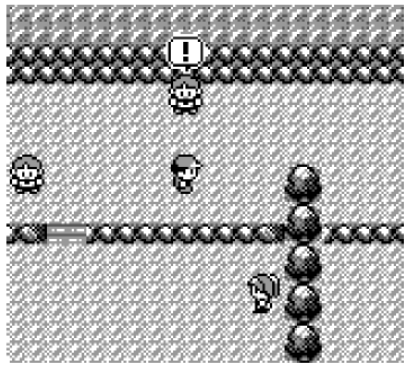
Can RL do the same with a more complex game that involves:

1. Multiple very different tasks (exploring, turn-based battling)
2. Sequential Tasks, Exploration, Puzzle-Solving
3. Dealing with a very large policy space?





(a) Overworld with Long Grass outside of Pallet Town



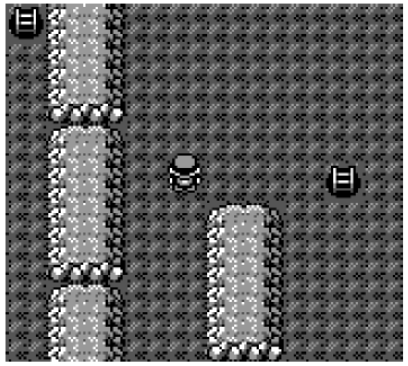
(b) Mandatory Trainer Battle Triggered on Line-of-sight



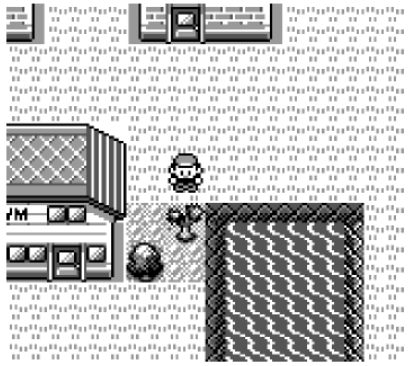
(c) Pokémon Center Healing by Interacting with Nurse Joy



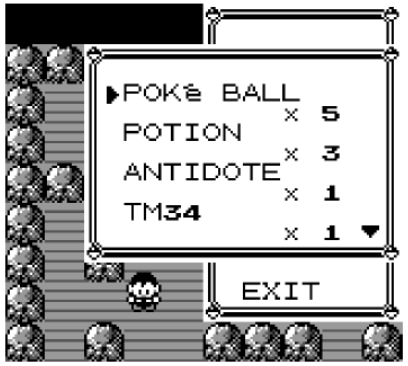
(d) Pokémon Battle: Selecting a Super Effective Attack



(e) Mt. Moon: Maze-like Dungeon Full of Wild Pokémon



(f) Cuttable Tree as Obstacle and Game Mechanic



(g) Item Menu outside of Battle



(h) Party Menu Displays Party Pokémon Order, HP, and Level

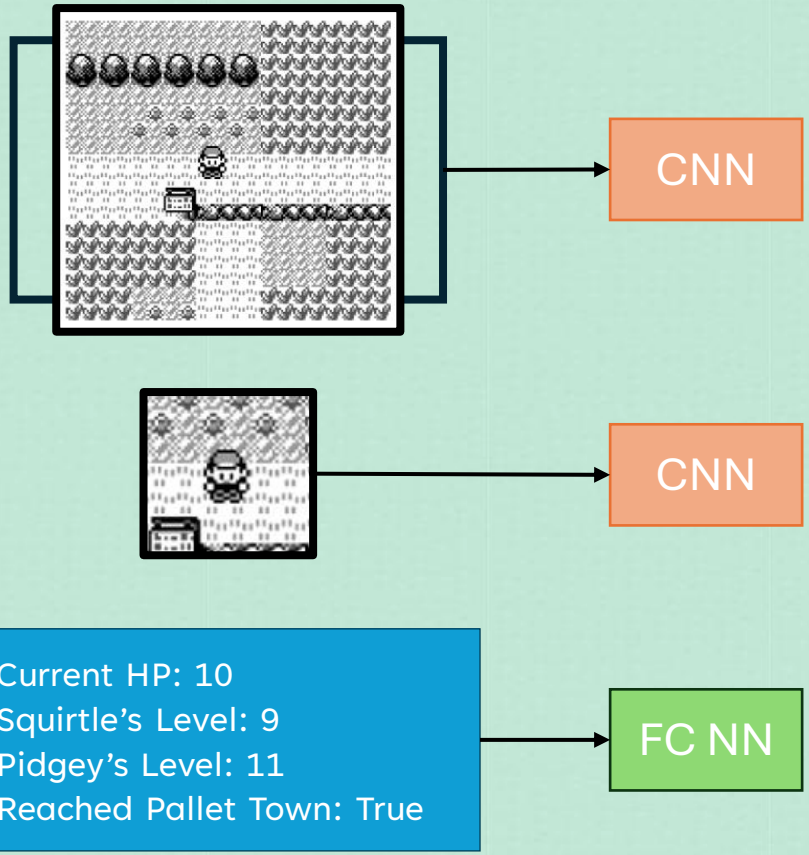
Fig. 1: All subfigures depict game screens possibly perceivable by the agent, presenting various challenges in Pokémon Red that involve exploration, navigation, and strategic decision-making. The agent must traverse a 2D overworld (a) with a party of Pokémon (h), winning mandatory trainer battles (b) and navigating complex maze-like areas (e). Progression also depends on interacting with the game's UI, including healing Pokémon at a Pokémon Center (c), using strategy to win battles (d), using items (g). Overcoming obstacles, like cuttable trees (f), requires the use of mandatory game-mechanic moves, which must be obtained via exploration, taught to eligible Pokémon, and used via the Start menu interface when facing the obstacle.

Environment/Simulator

The environment is a, emulation of the game as it was played on the GBA.

As sensors, the model has:

- An image of the whole screen
- A cropped image around the character
- Game state information



Actuators

The actions are not abstracted but rather lifted directly from a GameBoy's controls.

- A
- B
- Start
- Up
- Down
- Left
- Right

Note: Movement controls, i.e. up, down, left, and right in the overworld, are condensed into 8-frame long presses

The agent gets a 'budget' of actions that increases when it accomplishes goals.

Rewards

Events: +2 for each major event

Navigation: +0.005 for each new overworld coordinate visited

Healing: proportional to party HP

$$R_{\text{heal}} = 2.5 \sum_{i=1}^{N} \frac{\text{HP}_i^{\text{after}} - \text{HP}_i^{\text{before}}}{\text{HP}_i^{\text{max}}}$$

Levelling up: Up to +22

$$R_{\text{lvl}} = 0.5 \min \left(\sum_{i=1}^{N} \text{lvl}_i, \frac{\sum_{i=1}^{N} \text{lvl}_i - 22}{4} + 22 \right)$$

Rewards

Events: +2 for each major event

Forces the model to move through the game.
Pitfall: Model always chooses Charmander instead of Squirtle or Bulbasaur, because selecting Charmander requires only 1 button press for +2 reward.

Navigation: +0.005 for each new overworld coordinate visited

Healing: proportional to party HP

$$R_{\text{heal}} = 2.5 \sum_{i=1}^{N} \frac{\text{HP}_i^{\text{after}} - \text{HP}_i^{\text{before}}}{\text{HP}_i^{\text{max}}}$$

Levelling up: Up to +22

$$R_{\text{lvl}} = 0.5 \min \left(\sum_{i=1}^{N} \text{lvl}_i, \frac{\sum_{i=1}^{N} \text{lvl}_i - 22}{4} + 22 \right)$$

Rewards

Events: +2 for each major event

Navigation: +0.005 for each new overworld coordinate visited

Healing: proportional to party HP

$$R_{\text{heal}} = 2.5 \sum_{i=1}^{N} \frac{\text{HP}_i^{\text{after}} - \text{HP}_i^{\text{before}}}{\text{HP}_i^{\max}}$$

Levelling up: Up to +22

$$R_{\text{lvl}} = 0.5 \min\left(\sum_{i=1}^{N} \text{lvl}_i, \frac{\sum_{i=1}^{N} \text{lvl}_i - 22}{4} + 22\right)$$

The Agent will hang around Pallet Town unless rewarded more for exploring, since the 'gap' to reach the next event reward is too big

Rewards

Events: +2 for each major event

Navigation: +0.005 for each new overworld coordinate visited

Healing: proportional to party HP

$$R_{\text{heal}} = 2.5 \sum_{i=1}^{N} \frac{\text{HP}_i^{\text{after}} - \text{HP}_i^{\text{before}}}{\text{HP}_i^{\text{max}}}$$

Levelling up: Up to +22

$$R_{\text{lvl}} = 0.5 \min \left(\sum_{i=1}^{N} \text{lvl}_i, \frac{\sum_{i=1}^{N} \text{lvl}_i - 22}{4} + 22 \right)$$

The agent will avoid going back to town to heal injured Pokemon without an explicit reward for it.
Pitfall: If too high, it starts looping between getting injured and healing rather than progressing

Proximal Policy Optimization

1. In $q_t(\theta)$ the old policy is compared to the new policy. a_t is an action.
2. Checks if an action was better or worse than predicted in A_{π}^{GAE}.
3. Computes policy loss (which is clipped) and updates the neural network.

Ablation: a Gated Recurrent Unit (GRU) to store some temporal information.

$$L_t^C(\theta) = -\mathbb{E}_t\Big[\min\Big(q_t(\theta)A_\pi^{\mathrm{GAE}}(s_t,a_t),\ \mathrm{clip}(q_t(\theta),1-\epsilon,1+\epsilon)A_\pi^{\mathrm{GAE}}(s_t,a_t)\Big)\Big] \quad (3)$$

$$\text{where} \quad q_t(\theta) = \frac{\pi(a_t|s_t,\theta)}{\pi(a_t|s_t,\theta_{\mathrm{old}})}$$

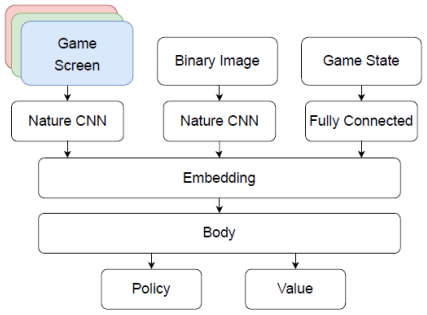


Fig. 2: Actor-Critic Network (2M parameters, GRU: 4M).

The paper is a bit vague about model details.

Can it beat the game?

TABLE II: Performance results of various ablation experiments. For each metric, the best data point—determined by milestone completion over time—is selected. Rows are sorted in descending order by the number of completed milestones. The tracked milestones include reaching Viridian Forest, defeating Brock, reaching Mt. Moon, arriving at Cerulean City, defeating Misty, solving Bill's quest, and completing Cerulean City by defeating the Rocket Thief. The best values in each column are highlighted in bold. The *Fast* experiment saves time (i.e., steps) by setting the game's text speed to fast and disabling battle animations. The *Choose Starter* experiment does not skip the Oak parcel quest at the beginning of the game, instead starting at the beginning where the player still needs to selects their first Pokémon.

Experiment	Milestones	Beat Brock	Mt Moon	Cerulean City	Beat Misty	Bill Quest	Cerulean Done	Beat Brock Steps	Cerulean City Steps	Cerulean Done Steps
Human	7.00 ± 0.0	1.00 ± 0.0	1.00 ± 0.0	1.00 ± 0.0	1.00 ± 0.0	1.00 ± 0.0	1.00 ± 0.0	5403 ± 3824	11188 ± 5224	16842 ± 7413
Baseline $- R_{lvl}$	$\mathbf{4.60 \pm 0.7}$	0.95 ± 0.0	0.91 ± 0.1	0.79 ± 0.1	0.31 ± 0.3	0.32 ± 0.4	0.03 ± 0.1	6117 ± 1758	21352 ± 3898	$\mathbf{37938 \pm 0}$
Fast	4.40 ± 0.8	0.98 ± 0.0	$\mathbf{0.98 \pm 0.0}$	$\mathbf{0.93 \pm 0.1}$	$\mathbf{0.33 \pm 0.4}$	0.19 ± 0.4	0.00 ± 0.0	$\mathbf{3753 \pm 990}$	$\mathbf{18523 \pm 4280}$	–
Baseline	4.19 ± 0.2	$\mathbf{0.99 \pm 0.0}$	0.97 ± 0.0	0.85 ± 0.1	0.27 ± 0.3	0.11 ± 0.2	0.00 ± 0.0	5587 ± 1235	25299 ± 4950	–
Baseline $- R_{heal}$	3.94 ± 0.2	$\mathbf{0.99 \pm 0.0}$	0.97 ± 0.0	0.91 ± 0.1	0.07 ± 0.1	0.00 ± 0.0	0.00 ± 0.0	5497 ± 503	28813 ± 2880	–
GRU	3.93 ± 2.2	0.76 ± 0.4	0.71 ± 0.4	0.49 ± 0.4	0.21 ± 0.3	$\mathbf{0.48 \pm 0.4}$	$\mathbf{0.21 \pm 0.3}$	5624 ± 780	27118 ± 4236	46532 ± 2249
Fast$-R_{heal}$	3.77 ± 0.5	0.97 ± 0.0	0.96 ± 0.0	0.73 ± 0.3	0.12 ± 0.2	0.00 ± 0.0	0.00 ± 0.0	6531 ± 1185	28871 ± 1362	–
Charmander $- R_{heal}$	3.73 ± 0.2	0.94 ± 0.0	0.92 ± 0.0	0.88 ± 0.1	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	11694 ± 1439	30369 ± 4828	–
Choose Starter	3.43 ± 1.7	0.79 ± 0.4	0.79 ± 0.4	0.75 ± 0.4	0.29 ± 0.4	0.00 ± 0.0	0.00 ± 0.0	6485 ± 578	20279 ± 579	–
Choose Starter $- R_{heal}$	3.40 ± 0.4	0.97 ± 0.0	0.96 ± 0.0	0.45 ± 0.3	0.01 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	13871 ± 2991	33565 ± 13303	–
Bulbasaur $- R_{heal}$	3.18 ± 0.3	0.97 ± 0.0	0.95 ± 0.0	0.26 ± 0.2	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	13433 ± 950	45943 ± 3569	–
Charmander	3.03 ± 1.6	0.69 ± 0.4	0.67 ± 0.3	0.53 ± 0.4	0.02 ± 0.0	0.17 ± 0.3	0.02 ± 0.0	7029 ± 1132	21309 ± 5915	48544 ± 0
Bulbasaur	2.89 ± 0.1	0.95 ± 0.0	0.94 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	9929 ± 841	–	–
Baseline $+ 10R_{nav}$	1.04 ± 0.1	0.04 ± 0.1	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	11559 ± 0	–	–
Baseline $- R_{nav}$	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	–	–	–

GRU = has memory. Fast = skips dialogue (fewer button presses from model)

Strengths

+ The task set/environment is open-ended and has a very long horizon (10+ hours). StarCraft II, DotA 2 and most other MOBA/Strategy games have a short task horizon (<1 hour).
+ Multiple possible 'pitfalls' that adjusting different rewards fixes, making it a good demo for studying how to structure rewards
+ Multiple 'modes' to handle – i.e. overworld, turn-based combat.
+ Very efficient (final model's only 10M params)



Article

Grandmaster level in StarCraft II using multi-agent reinforcement learning

https://doi.org/10.1038/s41586-019-1724-z
Received: 30 August 2019
Accepted: 10 October 2019
Published online: 30 October 2019

Oriol Vinyals[1,3]*, Igor Babuschkin[1,3], Wojciech M. Czarnecki[1,3], Michaël Mathieu[1,3], Andrew Dudzik[1,3], Junyoung Chung[1,3], David H. Choi[1,3], Richard Powell[1,3], Timo Ewalds[1,3], Petko Georgiev[1,3], Junhyuk Oh[1,3], Dan Horgan[1,3], Manuel Kroiss[1,3], Ivo Danihelka[1,3], Aja Huang[1,3], Laurent Sifre[1,3], Trevor Cai[1,3], John P. Agapiou[1,3], Max Jaderberg[1], Alexander S. Vezhnevets[1], Rémi Leblond[1], Tobias Pohlen[1], Valentin Dalibard[1], David Budden[1], Yury Sulsky[1], James Molloy[1], Tom L. Paine[1], Caglar Gulcehre[1], Ziyu Wang[1], Tobias Pfaff[1], Yuhuai Wu[1], Roman Ring[1], Dani Yogatama[1], Dario Wünsch[1], Katrina McKinney[1], Oliver Smith[1], Tom Schaul[1], Timothy Lillicrap[1], Koray Kavukcuoglu[1], Demis Hassabis[1], Chris Apps[1,2] & David Silver[1,3]*

Nature, 2019

Dota 2 with Large Scale Deep Reinforcement Learning

OpenAI, *
Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław "Psyho" Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, Susan Zhang

March 10, 2021

Preprinted

Towards Playing Full MOBA Games with Deep Reinforcement Learning

Deheng Ye[1] Guibin Chen[1] Wen Zhang[1] Sheng Chen[1] Bo Yuan[1] Bo Liu[1] Jia Chen[1] Zhao Liu[1] Fuhao Qiu[1] Hongsheng Yu[1] Yinyuting Yin[1] Bei Shi[1] Liang Wang[1] Tengfei Shi[1] Qiang Fu[1] Wei Yang[1] Lanxiao Huang[2] Wei Liu[1]
[1] Tencent AI Lab, Shenzhen, China
[2] Tencent TiMi L1 Studio, Chengdu, China
{dericye,beanchen,zivenwzhang,victchen,jerryyuan,leobliu,jaylahchen,ricardoliu, frankfhqiu,yannickyu,mailyyin,beishi,enginewang,francisshi,leonfu, willyang,jackiehuang}@tencent.com; wl2223@columbia.edu

NeurIPS 2020

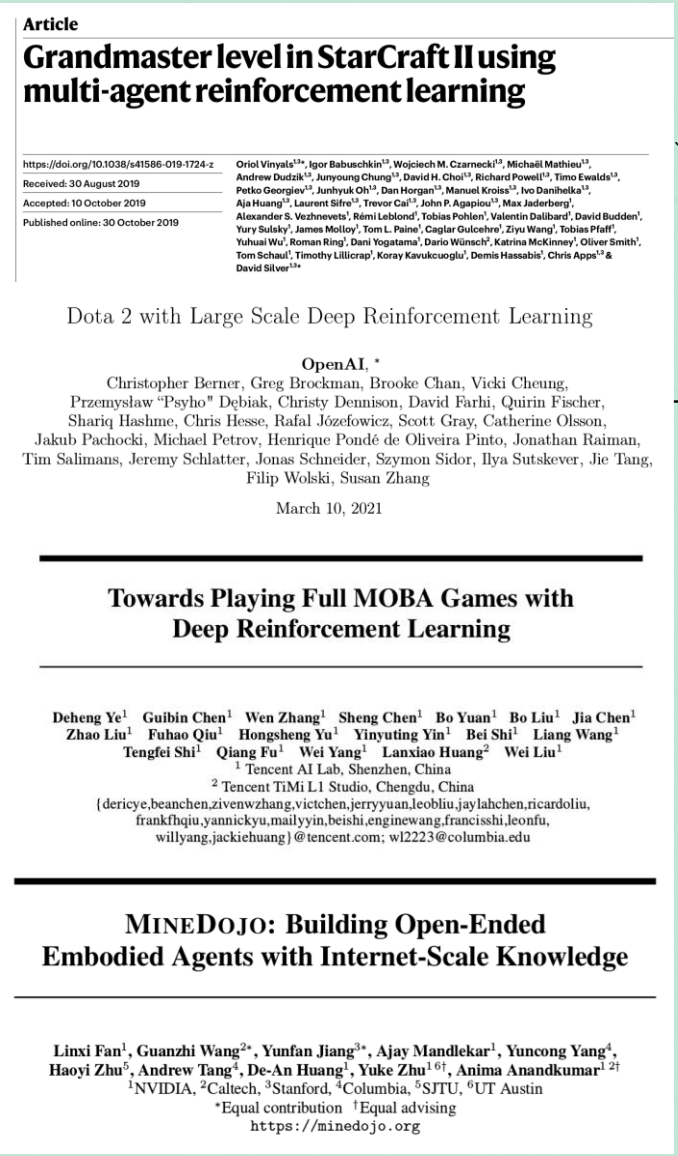
MINEDOJO: Building Open-Ended Embodied Agents with Internet-Scale Knowledge

Linxi Fan[1], Guanzhi Wang[2*], Yunfan Jiang[3*], Ajay Mandlekar[1], Yuncong Yang[4], Haoyi Zhu[5], Andrew Tang[4], De-An Huang[1], Yuke Zhu[1 6†], Anima Anandkumar[1 2†]
[1]NVIDIA, [2]Caltech, [3]Stanford, [4]Columbia, [5]SJTU, [6]UT Austin
*Equal contribution †Equal advising
https://minedojo.org

NeurIPS 2022

Weaknesses

- Very sensitive to reward design. Some stages required a custom policy rather than just one global policy.
- Relies on visual information rather than just spatial coordinates.
- Comparable works can assign free-form tasks rather than just one target goal (e.g. MineDojo learns to complete a variety of tasks that could be prompted by users rather than just one end goal)



Article

Grandmaster level in StarCraft II using multi-agent reinforcement learning

https://doi.org/10.1038/s41586-019-1724-z
Received: 30 August 2019
Accepted: 10 October 2019
Published online: 30 October 2019

Oriol Vinyals[1,3]*, Igor Babuschkin[1,3], Wojciech M. Czarnecki[1,3], Michaël Mathieu[1,3], Andrew Dudzik[1,3], Junyoung Chung[1,3], David H. Choi[1,3], Richard Powell[1,3], Timo Ewalds[1,3], Petko Georgiev[1,3], Junhyuk Oh[1,3], Dan Horgan[1,3], Manuel Kroiss[1,3], Ivo Danihelka[1,3], Aja Huang[1,3], Laurent Sifre[1,3], Trevor Cai[1,3], John P. Agapiou[1,3], Max Jaderberg[1], Alexander S. Vezhnevets[1], Rémi Leblond[1], Tobias Pohlen[1], Valentin Dalibard[1], David Budden[1], Yury Sulsky[1], James Molloy[1], Tom L. Paine[1], Caglar Gulcehre[1], Ziyu Wang[1], Tobias Pfaff[1], Yuhuai Wu[1], Roman Ring[1], Dani Yogatama[1], Dario Wünsch[1], Katrina McKinney[1], Oliver Smith[1], Tom Schaul[1], Timothy Lillicrap[1], Koray Kavukcuoglu[1], Demis Hassabis[1], Chris Apps[1,2] & David Silver[1,3]*

Nature, 2019

Dota 2 with Large Scale Deep Reinforcement Learning

OpenAI, *
Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław "Psyho" Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, Susan Zhang

March 10, 2021

Preprinted

Towards Playing Full MOBA Games with Deep Reinforcement Learning

Deheng Ye[1] Guibin Chen[1] Wen Zhang[1] Sheng Chen[1] Bo Yuan[1] Bo Liu[1] Jia Chen[1] Zhao Liu[1] Fuhao Qiu[1] Hongsheng Yu[1] Yinyuting Yin[1] Bei Shi[1] Liang Wang[1] Tengfei Shi[1] Qiang Fu[1] Wei Yang[1] Lanxiao Huang[2] Wei Liu[1]
[1] Tencent AI Lab, Shenzhen, China
[2] Tencent TiMi L1 Studio, Chengdu, China
{dericye,beanchen,zivenwzhang,victchen,jerryyuan,leobliu,jaylahchen,ricardoliu, frankfhqiu,yannickyu,mailyyin,beishi,enginewang,francisshi,leonfu, willyang,jackiehuang}@tencent.com; wl2223@columbia.edu

NeurIPS 2020

MINEDOJO: Building Open-Ended Embodied Agents with Internet-Scale Knowledge

Linxi Fan[1], Guanzhi Wang[2*], Yunfan Jiang[3*], Ajay Mandlekar[1], Yuncong Yang[4], Haoyi Zhu[5], Andrew Tang[4], De-An Huang[1], Yuke Zhu[1 6†], Anima Anandkumar[1 2†]
[1]NVIDIA, [2]Caltech, [3]Stanford, [4]Columbia, [5]SJTU, [6]UT Austin
*Equal contribution †Equal advising
https://minedojo.org

NeurIPS 2022

Extensions

- Penalize repetitive actions and reward later actions exponentially more. Authors do not experiment with this, but this could disincentivize looping.
- Build separate policies for battling to improve performance and avoid 'overarching rewards' (like healing) from affecting battle actions.
- Specify shorter term goals as checkpoints, e.g. "Catch a specific Pokémon" or "Find a specific skill." that support long-term goals ("Beat a Gym Leader")
  - Could use a language model here to help define those goals. This keeps the model primarily RL-based, but with support for things like understanding text from NPCs.
  - This also depends on what the overall goal is. Do we want an agent that completes the game at all costs? Or an agent that plays it like a human would?