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

Abstract

On April 13th, 2019, OpenAI Five became the first AI system to defeat the world champions at an esports game. The game of Dota 2 presents novel challenges for AI systems such as long time horizons, imperfect information, and complex, continuous state-action spaces, all challenges which will become increasingly central to more capable AI systems. OpenAI Five leveraged existing reinforcement learning techniques, scaled to learn from batches of approximately 2 million frames every 2 seconds. We developed a distributed training system and tools for continual training which allowed us to train OpenAI Five for 10 months. By defeating the Dota 2 world champion (Team OG), OpenAI Five demonstrates that self-play reinforcement learning can achieve superhuman performance on a difficult task.

1 Introduction

The long-term goal of artificial intelligence is to solve advanced real-world challenges. Games have served as stepping stones along this path for decades, from Backgammon (1992) to Chess (1997) to Atari (2013)[1–3]. In 2016, AlphaGo defeated the world champion at Go using deep reinforcement learning and Monte Carlo tree search[4]. In recent years, reinforcement learning (RL) models have tackled tasks as varied as robotic manipulation[5], text summarization [6], and video games such as Starcraft[7] and Minecraft[8].

Relative to previous AI milestones like Chess or Go, complex video games start to capture the complexity and continuous nature of the real world. Dota 2 is a multiplayer real-time strategy game produced by Valve Corporation in 2013, which averaged between 500,000 and 1,000,000 concurrent players between 2013 and 2019. The game is actively played by full time professionals; the prize pool for the 2019 international championship exceeded $35 million (the largest of any esports game in the world)[9, 10]. The game presents challenges for reinforcement learning due to long time horizons, partial observability, and high dimensionality of observation and action spaces. Dota 2's

[*] Authors listed alphabetically. Please cite as OpenAI et al., and use the following bibtex for citation: `https://openai.com/bibtex/openai2019dota.bib`

rules are also complex — the game has been actively developed for over a decade, with game logic implemented in hundreds of thousands of lines of code.

The key ingredient in solving this complex environment was to scale existing reinforcement learning systems to unprecedented levels, utilizing thousands of GPUs over multiple months. We built a distributed training system to do this which we used to train a Dota 2-playing agent called OpenAI Five. In April 2019, OpenAI Five defeated the Dota 2 world champions (Team OG[1]), the first time an AI system has beaten an esport world champion[2]. We also opened OpenAI Five to the Dota 2 community for competitive play; OpenAI Five won 99.4% of over 7000 games.

One challenge we faced in training was that the environment and code continually changed as our project progressed. In order to train without restarting from the beginning after each change, we developed a collection of tools to resume training with minimal loss in performance which we call *surgery*. Over the 10-month training process, we performed approximately one surgery per two weeks. These tools allowed us to make frequent improvements to our strongest agent within a shorter time than the typical practice of training from scratch would allow. As AI systems tackle larger and harder problems, further investigation of settings with ever-changing environments and iterative development will be critical.

In section 2, we describe Dota 2 in more detail along with the challenges it presents. In section 3 we discuss the technical components of the training system, leaving most of the details to appendices cited therein. In section 4, we summarize our long-running experiment and the path that lead to defeating the world champions. We also describe lessons we've learned about reinforcement learning which may generalize to other complex tasks.

2 Dota 2

Dota 2 is played on a square map with two teams defending bases in opposite corners. Each team's base contains a structure called an ancient; the game ends when one of these ancients is destroyed by the opposing team. Teams have five players, each controlling a hero unit with unique abilities. During the game, both teams have a constant stream of small "creep" units, uncontrolled by the players, which walk towards the enemy base attacking any opponent units or buildings. Players gather resources such as gold from creeps, which they use to increase their hero's power by purchasing items and improving abilities.[3]

To play Dota 2, an AI system must address various challenges:

- **Long time horizons.** Dota 2 games run at 30 frames per second for approximately 45 minutes. OpenAI Five selects an action every fourth frame, yielding approximately 20,000 steps per episode. By comparison, chess usually lasts 80 moves, Go 150 moves[11].

- **Partially-observed state.** Each team in the game can only see the portion of the game state near their units and buildings; the rest of the map is hidden. Strong play requires making inferences based on incomplete data, and modeling the opponent's behavior.

[1] `https://www.facebook.com/OGDota2/`

[2] Full game replays and other supplemental can be downloaded from: `https://openai.com/blog/how-to-train-your-openai-five/`

[3] Further information the rules and gameplay of Dota 2 is readily accessible online; a good introductory resource is `https://purgegamers.true.io/g/dota-2-guide/`

- **High-dimensional action and observation spaces.** Dota 2 is played on a large map containing ten heroes, dozens of buildings, dozens of non-player units, and a long tail of game features such as runes, trees, and wards. OpenAI Five observes $\sim 16,000$ total values (mostly floats and categorical values with hundreds of possibilities) each time step. We discretize the action space; on an average timestep our model chooses among 8,000 to 80,000 actions (depending on hero). For comparison Chess requires around one thousand values per observation (mostly 6-possibility categorical values) and Go around six thousand values (all binary)[12]. Chess has a branching factor of around 35 valid actions, and Go around 250[11].

Our system played Dota 2 with two limitations from the regular game:

- Subset of 17 heroes — in the normal game players select before the game one from a pool of 117 heroes to play; we support 17 of them.[4]

- No support for items which allow a player to temporarily control multiple units at the same time (Illusion Rune, Helm of the Dominator, Manta Style, and Necronomicon). We removed these to avoid the added technical complexity of enabling the agent to control multiple units.

3 Training System

3.1 Playing Dota using AI

Humans interact with the Dota 2 game using a keyboard, mouse, and computer monitor. They make decisions in real time, reason about long-term consequences of their actions, and more. We adopt the following framework to translate the vague problem of “play this complex game at a superhuman level" into a detailed objective suitable for optimization.

Although the Dota 2 engine runs at 30 frames per second, OpenAI Five only acts on every 4th frame which we call a *timestep*. Each timestep, OpenAI Five receives an *observation* from the game engine encoding all the information a human player would see such as units’ health, position, etc (see Appendix E for an in-depth discussion of the observation). OpenAI Five then returns a discrete *action* to the game engine, encoding a desired movement, attack, etc.

Certain game mechanics were controlled by hand-scripted logic rather than the policy: the order in which heroes purchase items and abilities, control of the unique courier unit, and which items heroes keep in reserve. While we believe the agent could ultimately perform better if these actions were not scripted, we achieved superhuman performance before doing so. Full details of our action space and scripted actions are described in Appendix F.

Some properties of the environment were randomized during training, including the heroes in the game and which items the heroes purchased. Sufficiently diverse training games are necessary to ensure robustness to the wide variety of strategies and situations that arise in games against human opponents. See subsection O.2 for details of the domain randomizations.

We define a *policy* (π) as a function from the history of observations to a probability distribution over actions, which we parameterize as a recurrent neural network with approximately 159 million parameters (θ). The neural network consists primarily of a single-layer 4096-unit LSTM [13] (see Figure 1). Given a policy, we play games by repeatedly passing the current observation as input and sampling an action from the output distribution at each timestep.

[4] See Appendix P for experiments characterizing the effect of hero pool size.