CUBETR: LEARNING TO SOLVE THE RUBIK'S CUBE USING TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Since its first appearance, transformers have been successfully used in wide ranging domains from computer vision to natural language processing. Application of transformers in Reinforcement Learning by reformulating it as a sequence modelling problem was proposed only recently. Compared to other commonly explored reinforcement learning problems, the Rubik's cube poses a unique set of challenges. The Rubik's cube has a single solved state for quintillions of possible configurations which leads to extremely sparse rewards. The proposed model CubeTR attends to longer sequences of actions and addresses the problem of sparse rewards by giving more weightage to the connections between each position and a plausible solved state. CubeTR learns how to solve the Rubik's cube from arbitrary starting states without any human prior, and after move regularisation, the lengths of solutions generated by it are expected to be very close to those given by algorithms used by expert human solvers. CubeTR provides insights to the generalisability of learning algorithms to higher dimensional cubes and the applicability of transformers in other relevant sparse reward scenarios.

1 INTRODUCTION

Originally proposed by Vaswani et al. (2017), transformers have gained a lot of attention over the last few years. Transformers are widely used for sequence to sequence learning in NLP (Radford et al., 2018; Devlin et al., 2018; Radford et al., 2019; Brown et al., 2020), and start to show promises in other domains like image to classification (Dosovitskiy et al., 2020), instance segmentation (Wang et al., 2021) in computer vision and the decision transformer (Chen et al., 2021) in reinforcement learning. Transformers are capable of modeling long-range dependencies, and have tremendous representation power. In particular, the core mechanism of Transformers, self-attention, is designed to learn and update features based on all pairwise similarities between individual components of a sequence. There have been great advances of the transformer architecture in the field of natural language processing, with models like GPT-3 (Brown et al., 2020) and BERT (Devlin et al., 2018), and drawing upon transformer block architectures designed in this domain has proven to be very effective in other domains as well.

Although transformers have become widely used in NLP and Computer Vision, its applications in Reinforcement Learning are still under-explored. Borrowing form well experimented and easily scalable architectures like GPT (Radford et al., 2018), exploring applications of transformers in other domains has started becoming easier. The use of transformers in core reinforcement learning was first proposed by Chen et al. (2021). Reformulating the reinforcement learning objective as learning of action sequences, they used transformers to generate the action sequence, i.e. the policy.

Reinforcement Learning is a challenging domain. Contrary to other learning paradigms like supervised, unsupervised and self-supervised, reinforcement learning involves learning an optimal policy for an agent interacting with its environment. Rather than minimising losses with the expected output, reinforcement learning approaches involve maximising the reward. Training deep reinforcement learning (Li, 2017) architectures has been notoriously difficult, and the learning algorithm is unstable or even divergent when action value function is approximated with a nonlinear function like the activation functions common-place in neural networks. Various algorithms like DQN (Mnih et al., 2013) and D-DQN (Van Hasselt et al., 2016) tackle some of these challenges involved with deep reinforcement learning.

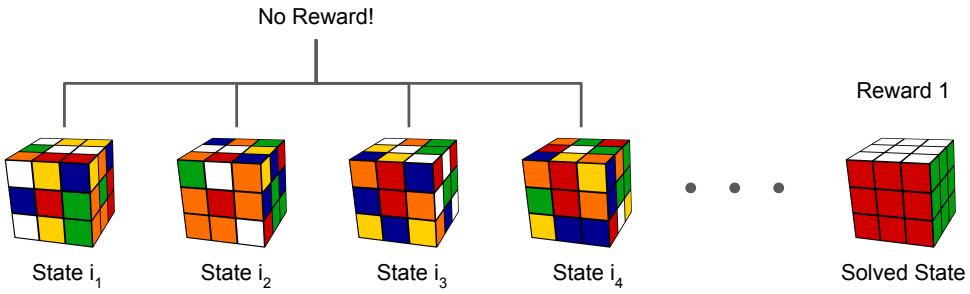

Figure 1: Sparse rewards in the case of the Rubik's cube. The state space is extremely large, but only the final solved state has a non-zero reward, making the reward distribution extremely sparse.

Reinforcement learning has seen applications in learning many games like chess, shogi (Silver et al., 2017), hex (Young et al., 2016) and Go (Silver et al., 2018). In October 2015, the distributed version of AlphaGo (Silver et al., 2016) defeated the European Go champion, becoming the first computer Go program to have beaten a professional human player on a full-sized board without handicap. Reinforcement learning has also been used for solving different kinds of puzzles (Dandurand et al., 2012), for protein folding (Jumper et al., 2021), for path planning (Zhang et al., 2015) and many other applications.

The Rubik's Cube is a particularly challenging single player game, invented in 1974 by Hungarian architecture and design professor Erno Rubik. Starting from a single solved state, the simple 3x3 rubik's cube can end up in 43 quintillion (43 x 10^{18}) different configurations. The number of different configurations (God's number) for a modified 4x4 cube is not even known. Thus, a random set of moves from this tremendous state space is highly unlikely to end up in the solved space. It is a marvel that humans have devised algorithms that can solve this challenging puzzle from any configuration in bounded number of moves.

Computers have been able to solve the cube for a long time now (). By implementing in software, an algorithm used by humans, very simple programs can be created to solve the cube very efficiently. These algorithms are, however, deeply rooted in group theory. Enabling an algorithm to learn to solve the cube on its own, without human priors is a much more challenging task. With its inherent complexity, this problem can provide new insights into much harder reinforcement learning problems, while also presenting new applications and interpretations of machine learning in abstract subjects like group theory and basic maths.

One of the biggest challenges in solving the Rubik's cube using reinforcement learning is that of sparse rewards. Since there is a single solved state, there is a single state with a non zero reward. All other 43 quintillion - 1 states have no reward. Reinforcement learning algorithms that rely very heavily on the rewards to decide optimal policies suffer since even large action sequences may end up with no reward. The method developed in this paper should be easy to generalise to many other problems suffering with sparse rewards, and may help provide deeper insights of the usefulness of transformers in more complicated reinforcement learning tasks. Although there have been some works solving this problem using deep reinforcement learning, this work also provides insights into the relationship between reinforcement learning policies and natural language processing sequential data generation.

This work is the first to explore the use of transformers in solving the rubik's cube, or in general any sparse reward reinforcement learning scenarios. Improving upon the decision transformer, CubeTR is able to effectively propagate the reward to actions far away from the final goal state. The transformer is further biased to generate smaller solutions by using a move regularisation factor. This is done to allow it to learn more efficient solutions, bridging the gap to human algorithms.

In the next section, some past work related to transformers, algorithms used for solving the rubik's cube and in general reinforcement learning are discussed. The methodology used in CubeTR is discussed in Section 3, along with description of data generation and representation. The experimental results of CubeTR and its comparison with other human based algorithms is discussed in Section 4, followed by the conclusions in Section 5.

2 RELATED WORK

Transformers were first proposed by Vaswani et al. (2017), where they worked with machine translation, but later apply it to constituency parsing to demonstrate the generalisability of the transformer architecture. Consisting of only attention blocks, the authors demonstrate that the transformer can replace recurrence and convolutions entirely. Along with superior performance, they demonstrated that transformers also lead to lower resource requirements, with more parallelizability. They laid the foundations for many subsequent works in NLP (Radford et al., 2018; Devlin et al., 2018; Radford et al., 2019; Brown et al., 2020) as well as computer vision (Dosovitskiy et al., 2020).

In GPT (Radford et al., 2018) the process of generative pre-training was first proposed. They showed that pre-training on a large scale unlabelled dataset gives significant performance boost over previous methods. Many other works (Ham et al., 2020; Ethayarajh, 2019; Budzianowski & Vulić, 2019) have experimented with and adopted this approach, including GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020). In (Hu et al., 2020), the GPT architecture was applied to graph neural networks. The GPT-2 architecture has been used for applications like data augmentation (Papanikolaou & Pierleoni, 2020), generating poetry (Meyer, 2019), patent claim generation (Lee & Hsiang, 2020) and many others. The recently proposed GPT-3 architecture had taken the NLP community by storm, and has also seen many applications (Branwen, 2020; Floridi & Chiriatti, 2020; Dale, 2021; McGuffie & Newhouse, 2020). Elkins & Chun (2020) even experimented with whether GPT-3 can pass the writer's Turing's test.

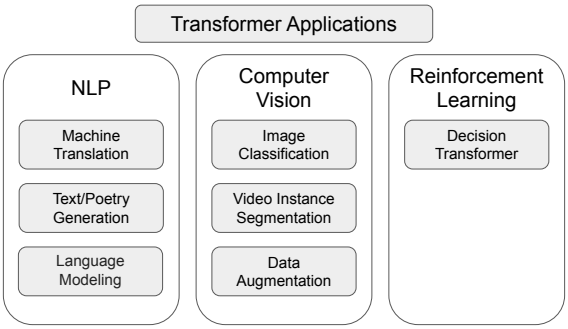

Figure 2: Some applications of transformers in different fields of machine learning.

Reinforcement Learning has been gaining a lot of popularity in recent times. With the advent of deep reinforcement learning, many other methods have been proposed in this field. Deep Q-Learning (Mnih et al., 2013; Hausknecht & Stone, 2015; Wang et al., 2016; Hessel et al., 2018) is a natural extension to the vanilla Q-learning algorithm where a neural network is used to predict the expected reward for each action from a particular state. Policy gradients (Schulman et al., 2015a; Mnih et al., 2016; Schulman et al., 2015b) uses gradient descent and other optimisation algorithms on the policy space to find optimal policies. Value iteration (Jothimurugan et al., 2021; Ernst et al., 2005; Munos, 2005) is another popular approach, where the expected value of each state is either calculated or approximated. The value of a state is the maximum reward that can be obtained from that state using some optimal policy.

A typical situation for reinforcement learning is a situation where an agent has to reach a goal and only receives a positive reward signal when he either reaches or is close enough to the target. This situation of sparse rewards is challenging for usual reinforcement learning that depend too much on the rewards to decide exploration and policies. Several methods have been proposed to deal with sparse reward situations (Riedmiller et al., 2018; Trott et al., 2019). Curiosity driven approaches (Pathak et al., 2017; Burda et al., 2018; Oudeyer, 2018) use curiosity as an intrinsic reward signal to enable the agent to explore its environment and learn skills that might be useful later in its life. Curriculum learning methods (Florensa et al., 2018) use a generative approach to getting new or auxiliary tasks that the agent solves. Reward Shaping (Mataric, 1994; Ng et al., 1999) is another commonly used approach in which the primary reward of the environment is enhanced with some additional reward features.

3 PROPOSED METHODOLOGY

4 EXPERIMENTAL ANALYSIS

5 CONCLUSION

REFERENCES

Gwern Branwen. Gpt-3 creative fiction. 2020.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Paweł Budzianowski and Ivan Vulić. Hello, it's gpt-2–how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. *arXiv preprint arXiv:1907.05774*, 2019.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *arXiv preprint arXiv:2106.01345*, 2021.

Robert Dale. Gpt-3: What's it good for? *Natural Language Engineering*, 27(1):113–118, 2021.

Frédéric Dandurand, Denis Cousineau, and Thomas R Shultz. Solving nonogram puzzles by reinforcement learning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 34, 2012.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Katherine Elkins and Jon Chun. Can gpt-3 pass a writer's turing test? *Journal of Cultural Analytics*, 1(1):17212, 2020.

Damien Ernst, Mevludin Glavic, Pierre Geurts, and Louis Wehenkel. Approximate value iteration in the reinforcement learning context. application to electrical power system control. *International Journal of Emerging Electric Power Systems*, 3(1), 2005.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*, 2019.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *International conference on machine learning*, pp. 1515–1528. PMLR, 2018.

Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020.

Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. End-to-end neural pipeline for goal-oriented dialogue systems using gpt-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 583–592, 2020.

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *2015 aaai fall symposium series*, 2015.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1857–1867, 2020.

Kishor Jothimurugan, Osbert Bastani, and Rajeev Alur. Abstract value iteration for hierarchical reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1162–1170. PMLR, 2021.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Jieh-Sheng Lee and Jieh Hsiang. Patent claim generation by fine-tuning openai gpt-2. *World Patent Information*, 62:101983, 2020.

Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

Maja J Mataric. Reward functions for accelerated learning. In *Machine learning proceedings 1994*, pp. 181–189. Elsevier, 1994.

Kris McGuffie and Alex Newhouse. The radicalization risks of gpt-3 and advanced neural language models. *arXiv preprint arXiv:2009.06807*, 2020.

Joseph B Meyer. *Generating Free Verse Poetry with Transformer Networks*. PhD thesis, Reed College, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PMLR, 2016.

Rémi Munos. Error bounds for approximate value iteration. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, pp. 1006. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287, 1999.

Pierre-Yves Oudeyer. Computational theories of curiosity-driven learning. *arXiv preprint arXiv:1802.10546*, 2018.

Yannis Papanikolaou and Andrea Pierleoni. Dare: Data augmented relation extraction with gpt-2. *arXiv preprint arXiv:2004.13845*, 2020.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Martin Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom Wiele, Vlad Mnih, Nicolas Heess, and Jost Tobias Springenberg. Learning by playing solving sparse reward tasks from scratch. In *International Conference on Machine Learning*, pp. 4344–4353. PMLR, 2018.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015a.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Alexander Trott, Stephan Zheng, Caiming Xiong, and Richard Socher. Keeping your distance: Solving sparse reward tasks using self-balancing shaped rewards. *arXiv preprint arXiv:1911.01417*, 2019.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8741–8750, 2021.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pp. 1995–2003. PMLR, 2016.

Kenny Young, Gautham Vasan, and Ryan Hayward. Neurohex: A deep q-learning hex agent. In *Computer Games*, pp. 3–18. Springer, 2016.

Baochang Zhang, Zhili Mao, Wanquan Liu, and Jianzhuang Liu. Geometric reinforcement learning for path planning of uavs. *Journal of Intelligent & Robotic Systems*, 77(2):391–409, 2015.