Exploration of Sequence Modeling for Reinforcement Learning
CPSC 533V 2023W2 Project Proposal

Juntai Cao[♠]
Student #: 50171404
jtcao7@cs.ubc.ca

Yuwei Yin[♠]
Student #: 36211928
yuweiyin@cs.ubc.ca

Xiang Zhang[♠]
Student #: 93035806
wyattz23@cs.ubc.ca

Department of Computer Science, University of British Columbia
2366 Main Mall, Vancouver, BC V6T 1Z4, Canada

1 Introduction

Searching and decision-making under uncertainty is a pivotal aspect of artificial intelligence (AI), with profound implications across different domains, especially robotics. The ability to make informed decisions in the face of incomplete or ambiguous information enables the development of robust and adaptive systems. Reinforcement learning (RL) (Sutton, 1988; Watkins and Dayan, 1992; Sutton et al., 1999; Konda and Tsitsiklis, 1999) is a powerful paradigm to make sequential decisions by learning from the consequences of actions. In RL (Sutton and Barto, 2018), an agent learns to achieve a goal in an uncertain, potentially complex environment by trial and error, receiving feedback through rewards or penalties. This approach allows the agent to develop a strategy, or policy, for selecting actions that maximize the cumulative reward over time.

As deep learning (DL) (Goodfellow et al., 2016) prospers in various AI domain, deep reinforcement learning exhibits excellent ability in agent control via value-based (Mnih et al., 2013, 2015) and policy-based (Silver et al., 2014; Lillicrap et al., 2016; Henderson et al., 2018) methods. Natural language processing (NLP), another AI field facilitated by deep learning, is also a sequence modeling task (Hochreiter and Schmidhuber, 1997; Cho et al., 2014; Chung et al., 2014) and Markov decision process. Thus, ***applying the powerful sequential models in NLP for RL tasks is intuitively promising***. Transformers (Vaswani et al., 2017), with their ability to process sequential data and capture long-range dependencies, have revolutionized the way machines understand and generate natural language (Devlin et al., 2019; Radford et al., 2018).

Following this line of research, previous works (Chen et al., 2021; Janner et al., 2021; Radosavovic et al., 2024) leverage the Transformer architecture to model reinforcement learning-based decision-making processes. This paradigm shift allows us to directly apply the powerful capabilities of Transformer to RL problems, enabling the model to learn optimal policies from sequences of states, actions, and rewards.

While there is a substantial amount of research applying transformers to offline locomotion prediction, the majority of these models predominantly leverage GPT (Radford et al., 2018) as the causal transformer of choice. This reveals a gap in the literature, as there is a notable absence of ablation studies exploring the impacts of various causal transformer architectures. Furthermore, although novel tokenization methods have been introduced, there is still a deficiency in comparative analyses regarding the influence of different tokenization techniques on model performance.

In this work, the main focus is to ***explore sequence modeling for reinforcement learning by framing the RL task as a sequence modeling NLP task***. Therefore, we propose to delve deeper into these issues through a series of experiments, and if time permits, explore several other aspects above in our targets as well. The implementation plan and experimental design are elaborated in § 3 and § 4, respectively.

2 Related Work

2.1 Transformer for Offline RL

Offline Reinforcement Learning (Levine et al., 2020), often called batch reinforcement learning, is a data-driven approach within the realm of reinforcement learning that utilizes pre-collected datasets. In light of the success and growing popularity of Transformer (Vaswani et al., 2017) in natural language processing and computer vision (Dosovitskiy et al., 2021) tasks in recent years, there has been an increased interest in applying transformer models to offline reinforcement learning scenarios.

[♠] Authors contributed equally and listed alphabetically.

Two pioneering work in this direction are Decision Transformer (Chen et al., 2021) and Trajectory Transformer (Janner et al., 2021). Although both of them mirror the causal Transformer structure of GPT (Radford et al., 2018), they exhibit differences in many ways.

2.2 Decision Transformer

Decision Transformer (Chen et al., 2021) models entire trajectories — rewards-to-go, states, and actions of each time step — rather than focusing on individual state-action pairs. This allows the model to capture the temporal dependencies and dynamics inherent in the environment. Instead of immediate rewards, the model uses the reward-to-go, which represents the sum of future rewards expected from a given state, following the intuition that the model should generate actions based on future desired returns, rather than past rewards. The model generates actions autoregressively based on the desired future outcome and the current state. This is a departure from traditional methods that often select actions based on the maximization of expected future rewards.

2.3 Trajectory Transformer

On the other hand, Trajectory Transformer (Janner et al., 2021) models the entire sequence of states, actions, and the final reward as trajectories. It focuses on predicting the next state, considering the entire history of past interactions. It is designed to be general and flexible, capable of handling a wide range of RL tasks without task-specific modifications.

2.4 Locomotion as Next Token Prediction

More recently, Radosavovic et al. (2024) re-framed the challenge of real-world humanoid control as a next token prediction problem, introducing the concept of aligned prediction to adeptly manage missing modalities. By merging the embeddings of observations and actions and treating them as a singular token, they use a causal Transformer to predict the next observation-action pairs. This approach allows for a straightforward handling of missing input modalities through the use of masks, whereby the model can simply disregard the loss associated with predictions corresponding to these masked parts of the input.

3 Implementation

In this section, we introduce the implementation plans for the project step by step.

3.1 Work 1: Code base

We will re-implement an updated version of the code base for the decision-making Transformer based on the original implementation of the Decision Transformer[1], which has obsolete code pieces and package dependencies that will result in bugs.

3.2 Work 2: Architecture Exploration

The Decision Transformer and Trajectory Transformer models both utilize GPT-2 (Radford et al., 2019) as their underlying architecture, which is characterized by a decoder-only structure and is trained through a next-token prediction task. We intend to explore various causal Transformer architectures that are not limited to decoder-only designs or exclusively trained via next-token prediction. This exploration aims to assess the diverse models' performance, enabling us to draw comprehensive comparisons among their outcomes.

3.3 Work 3: Tokenization Exploration

The three methods mentioned in § 2 differ mainly in the observation/state permutation in the trajectory and the tokenization approach in the Transformer encoder. We will try more trajectory and tokenization methods to show the effect of sequence modeling for RL.

3.4 Work 4: V4 Dataset

The readily-available trajectories in D4RL (Fu et al., 2020) are collected in "v2" version of the MuJoCo environment. We plan to obtain an offline datasets of "v4" MuJoCo environment using the state-of-the-art RL models on the target MuJoCo tasks.

3.5 Work 5: Transfer Learning

We plan to try to pre-train the RL policy on the offline datasets of the chosen three environments and fine-tune the model on another environment. The environment shift can be "v2" to "v4", "medium" level to "expert" level, or even a new MuJoCo task. For transferring to a complete new task, the mismatch of observation/state dimension is of concern. We will solve the problem via network design.

[1] https://github.com/kzl/decision-transformer

4 Experimental Setup

4.1 Tasks, Datasets, and Evaluation

Tasks We plan to experiment on three MuJoCo tasks (Todorov et al., 2012), i.e., Half Cheetah (Wawrzynski, 2007; Wawrzyński, 2009)[2], Hopper (Erez et al., 2011)[3], and Walker2D [4].

Datasets We will use the offline dataset from D4RL (Fu et al., 2020)[5]. The trajectories are collected by observing AI agents (RL policies) interacting with above three RL environments ("v2" version). Each dataset has three types (levels) of subsets ("random", "medium", and "expert") denoting different trajectory qualities.

Evaluation The performance of RL policies is measured by the rewards from the RL environment, i.e., the higher the reward, the better. We will the updated MuJoCo environments of "v4" version since the observation and action spaces of "v2" and "v4" versions are the same. The environment is also based on Gymnasium [6], which is rooted from OpenAI Gym (Brockman et al., 2016).

4.2 Experiments

Work 1: Code Base The experiment for Work 1 in § 3.1: To train the Decision Transformer model on the "v2" offline datasets and test the performance on "v4" MuJoCo environment. We expect the re-implementation of Decision Transformer to work well by looking at its results on different datasets of different levels ("random", "medium", and "expert"). Besides, we will compare the performance of Decision Transformer with that of other RL policies such as DT learning and Behavior Cloning.

Work 2: Architecture Exploration The experiment for Work 2 in § 3.2: To train the Decision Transformer models of different network architectures and compare their performance. We expect to find some insights into the model design.

[2] https://gymnasium.farama.org/environments/mujoco/half_cheetah/

[3] https://gymnasium.farama.org/environments/mujoco/hopper/

[4] https://gymnasium.farama.org/environments/mujoco/walker2d/

[5] https://github.com/Farama-Foundation/D4RL

[6] https://github.com/Farama-Foundation/Gymnasium

4.3 Work 3: Tokenization Exploration

The experiment for Work 3 in § 3.3: To implement other trajectory and tokenization approaches, train the models, and compare the results with baseline models such as the vanilla Decision Transformer.

Work 4: V4 Dataset The experiment for Work 4 in § 3.4: To train the RL policies either on readily-available "v2" offline trajectories or on our collected "v4" ones, and then evaluate them on the "v4" environment. We expect the model trained on the "v4" dataset outperforms its "v2" counterpart because there could be less training-testing mismatch.

Work 5: Transfer Learning The experiment for Work 5 in § 3.5: To pre-train the RL policy on one offline dataset, fine-tune the model on a new environment with online training, and test the performance (reward gaining) on the new environment. We expect the model will learn better, at least faster, with the help of pre-training.

5 Task Division and Timeline

5.1 Task Division

We will **contribute equally** to research idea discussion, proposal and paper writing, data processing, method implementation, experiments conducting, results analysis, and more. The main focus of each member is shown in Table 1.

Task	Juntai	Yuwei	Xiang
Run Experiments		✔	
Results analysis	✔		✔
Paper writing	✔	✔	✔
Work 1 Code Base		✔	
Work 2 Architecture		✔	✔
Work 3 Tokenization	✔	✔	✔
Work 4 V4 Dataset	✔		✔
Work 5 Transfer Learning	✔	✔	
Work 6 Future Work	✔		✔

Table 1: Task division of group members.

5.2 Timeline

Despite possible schedule changes, we will try to follow the project timeline as shown in Table 2.

References

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Woj-

Week	Juntai	Yuwei	Xiang
03.04–03.10	Project Planning & Literature Review		
03.11–03.17	Work 3 Exploration	Work 1 Code Base	Work 2 Exploration
03.18–03.24	Work 3 Exploration	Work 2&3 Experiments	Work 2&3 Exploration
03.25–03.31	Work 4 V4 Dataset	Work 2&3 Experiments	Work 4 V4 Dataset
04.01–04.07	Work 5 Exploration	Work 4 Experiments	Work 5 Exploration
04.08–04.14	Work 6 Exploration & Analysis	Work 5 Experiments	Work 6 Exploration & Analysis
04.15–04.17	Paper Writing		
Apr 17, Wed	Project Paper Submission		

Table 2: Timeline for the project.

ciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540*.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 15084–15097.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, abs/1412.3555.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Tom Erez, Yuval Tassa, and Emanuel Todorov. 2011. Infinite-horizon model predictive control for periodic tasks with contacts. In *Robotics: Science and Systems VII, University of Southern California, Los Angeles, CA, USA, June 27-30, 2011*.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. 2020. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, abs/2004.07219.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep reinforcement learning that matters. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3207–3214. AAAI Press.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Michael Janner, Qiyang Li, and Sergey Levine. 2021. Offline reinforcement learning as one big sequence modeling problem. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 1273–1286.

Vijay Konda and John Tsitsiklis. 1999. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press.

Sergey Levine, Aviral Kumar, G. Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *ArXiv*, abs/2005.01643.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations,*

ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings.

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

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. *OpenAI blog*.

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

Ilija Radosavovic, Bike Zhang, Baifeng Shi, Jathushan Rajasegaran, Sarthak Kamat, Trevor Darrell, Koushil Sreenath, and Jitendra Malik. 2024. Humanoid locomotion as next token prediction. *arXiv preprint arXiv:2402.19469*.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 387–395, Bejing, China. PMLR.

Richard S Sutton. 1988. Learning to predict by the methods of temporal differences. *Machine learning*, 3:9–44.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning*, 8:279–292.

Pawel Wawrzynski. 2007. Learning to control a 6-degree-of-freedom walking robot. In *EUROCON 2007-The International Conference on" Computer as a Tool"*, pages 698–705. IEEE.

Paweł Wawrzyński. 2009. A cat-like robot real-time learning to run. In *Adaptive and Natural Computing Algorithms: 9th International Conference, ICANNGA 2009, Kuopio, Finland, April 23-25, 2009, Revised Selected Papers 9*, pages 380–390. Springer.