Transformers in RL: Decision Transformers vs. Traditional RL

Amirreza Tanevardi Iman Ahmadi

Sharif University of Technology

Objectives

• Traditional RL relies on value functions, policy gradients, or explicit environment models, which often face stability issues, poor credit assignment, and difficulty with sparse rewards.

• Transformers, successful in NLP and vision, offer a new paradigm: treat RL as **sequence modeling**.

• Does this new paradigm work?

Introduction

Reinforcement Learning (RL) aims to maximize cumulative reward by interacting with environments. Traditional RL methods (e.g., Q-learning, policy gradients) learn through bootstrapping and exploration but are often unstable and data-hungry.

Decision Transformers reframe RL as **conditional sequence modeling**: states, actions, and returns are tokens in a sequence, and a Transformer predicts actions autoregressively. This connects RL to supervised learning with large models.

Extensions such as **Online Decision Transformers** bridge offline pretraining with online fine-tuning, addressing exploration and adaptability.

Method

Trajectory Representation

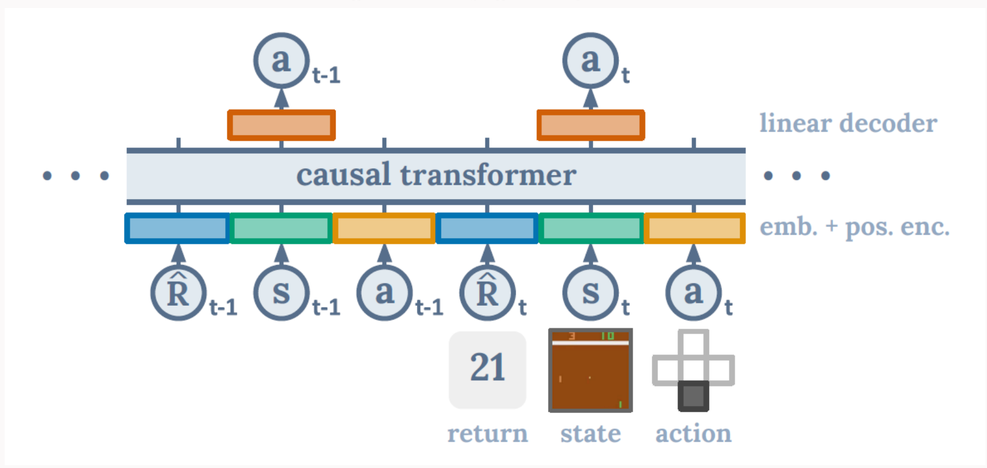
- Encode trajectories as sequences of *(return-to-go, state, action)* tokens.

$$\tau = \left(\hat{R}_1, s_1, a_1, \hat{R}_2, s_2, a_2, \ldots, \hat{R}_T, s_T, a_T\right) \qquad \hat{R}_t = \sum_{t'=t}^{T} r_{t'}$$

- Condition on a target return → model generates actions aiming to reach it.

Architecture

- GPT-style causal Transformer with modality embeddings (state, action, return).
- Predicts next action autoregressively given past tokens. The model is as below:

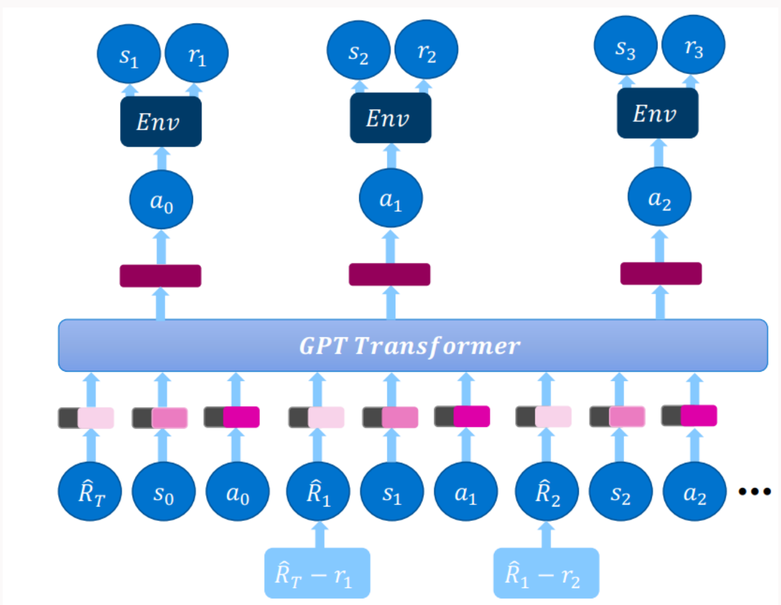


Offline Training

- Uses the architecture above
- Trains like language models on offline trajectories.
- Avoids value bootstrapping and explicit dynamics modeling.

Inference

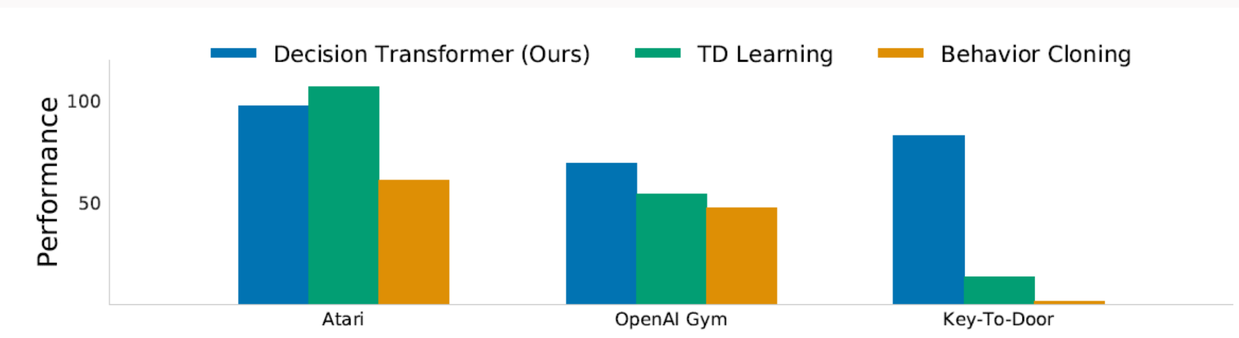
- We only have to provide the initial state and the desired return-to-go.
- To generate the next action, we have to feed the next state and the desired return-to-go from that point on, which is calculated by subtracting the reward we got in the previous step from the initial return-to-go.



Results

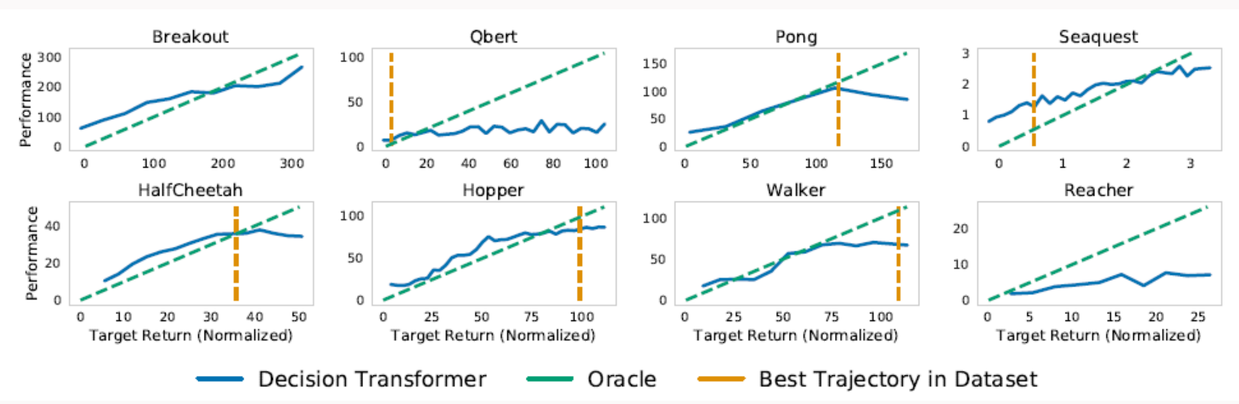
Offline Benchmarks (Atari, OpenAI Gym, D4RL):

- DT matches or exceeds strong baselines like CQL and behavior cloning.
- Particularly strong in **sparse reward** and **long-horizon credit assignment** tasks.



Does Decision Transformer model the distribution of returns?

- On every task, the desired target returns and the true observed returns are highly correlated.
- On some tasks like Pong, HalfCheetah and Walker, Decision Transformer generates trajectories that almost perfectly match the desired returns



Does Decision Transformer perform well in sparse reward settings?

- Decision Transformer can improve robustness in sparse reward settings since it makes minimal assumptions on the density of the reward.
- Delayed returns minimally affect Decision Transformer

Dataset	Environment	Delayed (Sparse) DT (Ours)	Delayed (Sparse) CQL	Agnostic BC	Agnostic %BC	Original (Dense) DT (Ours)	Original (Dense) CQL
Medium-Expert	Hopper	**107.3 ± 3.5**	9.0	59.9	102.6	107.6	111.0
Medium	Hopper	60.7 ± 4.5	5.2	63.9	**65.9**	67.6	58.0
Medium-Replay	Hopper	**78.5 ± 3.7**	2.0	27.6	70.6	82.7	48.6

Conclusion

• **Decision Transformers** demonstrate that RL can be cast as supervised sequence modeling, simplifying training and leveraging the power of large Transformer architectures.

• **Advantages:** handles long horizons, sparse rewards, and offline data effectively. And can return optimal actions based on the desired return that you condition the model on.

• **Challenges:** sample inefficiency, high compute cost, and careful trajectory design.

• **Outlook:** With online extensions (ODT), Transformers provide a flexible and scalable framework for RL, potentially becoming a cornerstone for generalizable, data-driven agents.

References

- Chen, Lili, 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. https://arxiv.org/abs/2106.01345
- Wenzhe Li, Hao Luo, Zichuan Lin, Chongjie Zhang, Zongqing Lu, Deheng Ye. A Survey on Transformers in Reinforcement Learning. arXiv preprint arXiv:2301.03044, 2023. https://arxiv.org/abs/2301.03044
- Qinqing Zheng, Amy Zhang, Aditya Grover. Online Decision Transformer. arXiv preprint arXiv:2202.05607, 2022. https://arxiv.org/abs/2202.05607
- Janner, Michael, et al. "Offline Reinforcement Learning as One Big Sequence Modeling Problem." Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia, 29 Nov. 2021, arXiv:2106.02039v4.