EXPLORING THE CAPABILITIES OF DECISION TRANSFORMERS

Gabriel Afriat

Anne Castille Buisson

Sara Pasquino

SENSORIMOTOR

Massachusetts
Institute of
Technology

AGENDA

Massachusetts Institute of Technology

INTRODUCTION TO DECISION TRANSFORMERS

- It solves **Reinforcement Learning** tasks with **Transformer architecture**, treating decision-making as a sequence prediction problem

Mechanism

- Treats the RL process by predicting an **optimal action** based on past sequences of states, actions, and rewards

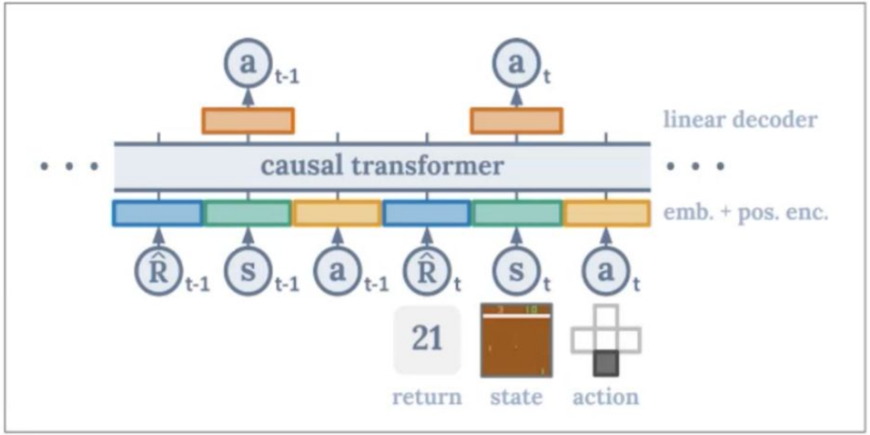
Characteristics

- Relies on Supervised Learning approach to predict the next best action. It is an **offline learning** algorithm, trained on **pre-collected data**

Advantages

- **Sequence modelling** approach enables learning with **lengthy sequences** and **sparse rewards** (reducing the need for reward design and discounting the return)

- Shows better performance for a fixed dataset size and is therefore **more sample efficient** than CQL



- Self-attention, unlike RL methods (which propagate rewards) is less prone to be affected by **"distractor" signals**

Massachusetts Institute of Technology

CQL FOR COMPARATIVE BENCHMARKING

- CQL minimizes the following loss:

$$\min_{\theta} \alpha \underbrace{\left(\mathbb{E}_{s\sim\mathcal{D}, a\sim\pi(a|s)}[Q_\theta(s,a)] - \mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_\beta(a|s)}[Q_\theta(s,a)] \right)}_{\text{Conservative Loss}} + \underbrace{\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(Q_\theta(s,a) - r(s,a) - \gamma \max_{a'} Q_{\theta_{target}}(s',a') \right)^2 \right]}_{\text{TD Loss}}$$

- Learns a Q-function which:
  - **Minimizes the TD error** on the dataset
  - **Doesn't overestimate** the Q-value of **unseen** (state, action) pairs

- State-of-the-art **offline learning** method: main competitor of the Decision Transformer

PROJECT OVERVIEW

Objectives

- **Validate** Chen et al.'s results and limitations
- Experiment with changing the model's parameters to test for **robustness**
- Compare performance of DT with **Conservative Q-Learning**
- Validate **generalizability** on new games
- Test performance on **long-term credit assignment**

Breakout

- **State Space**: stack of 4 frames representing current and recent past states (84x84 px)
  - Images capturing position of the ball, the paddle and the remaining bricks
- **Action Space**: dimension = 4
  - [0] no operations, [1] fire, [2] move right, [3] move left
- **Rewards**: assigned for each broken brick, according to colours

Ms PacMan

- **State Space**: stack of 4 frames representing current and recent past states (84x84 px)
  - Captures position of Ms Pacman, ghosts, the pellets and maze layout
- **Action Space**: dimension = 7
  - [0] no operations, [1] move up, [2] move down, [3] move right, [4] move left (and combinations)
- **Rewards**: assigned for each collected pellet and eaten ghost

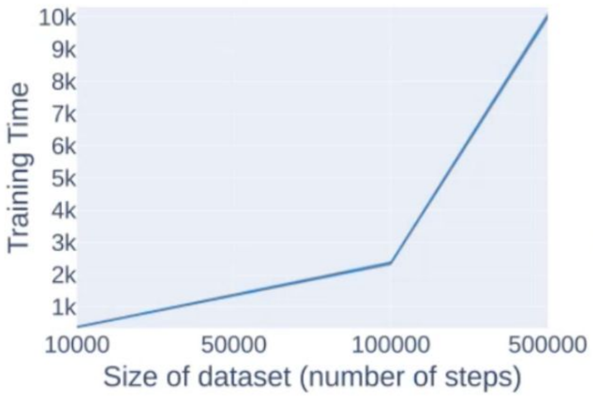
Massachusetts Institute of Technology

ABLATION STUDY

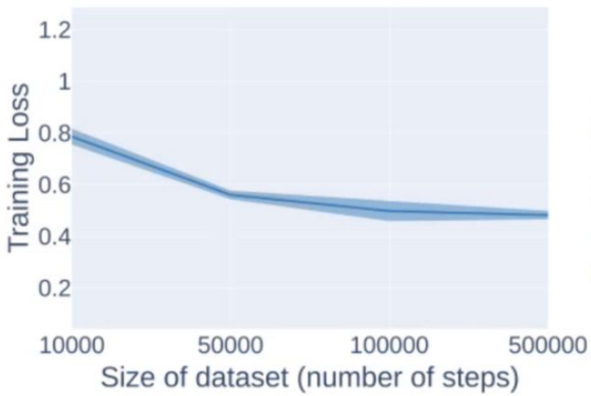
1. HOW DO DECISION TRANSFORMERS BEHAVE WITH RESPECT TO NUMBER OF SAMPLES?

- We experimented for several **number of steps** (size of the dataset)
- This allowed us to test the model's performance in scenarios of limited computing resources

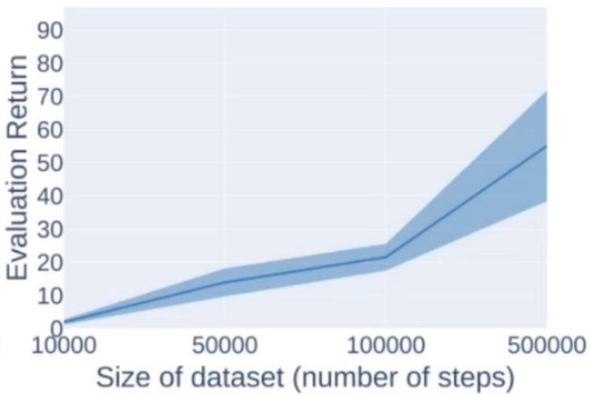
Each epoch requires more computing time when the size of the dataset increases



Increasing dataset size leads to a lower training loss



Similarly, increasing dataset size leads to better generalization



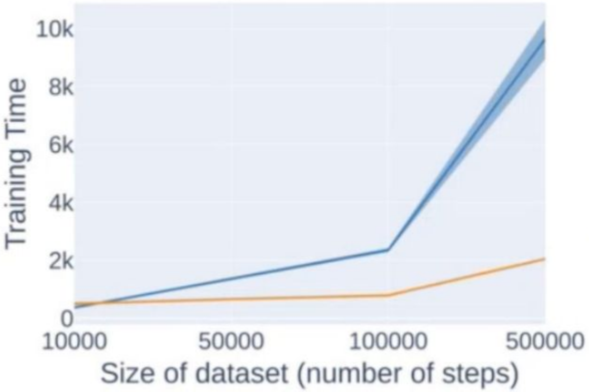
Massachusetts Institute of Technology

ABLATION STUDY

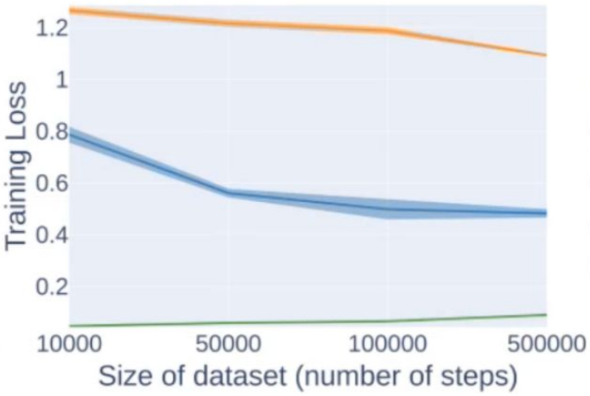
1. HOW DO DECISION TRANSFORMERS BEHAVE WITH RESPECT TO NUMBER OF SAMPLES?

- We compared results with those obtained from CQL; overall, CQL shows worse performance than DT
- Caveat: for a better sense of actual CQL performance, more hyperparameter tuning and training epochs required

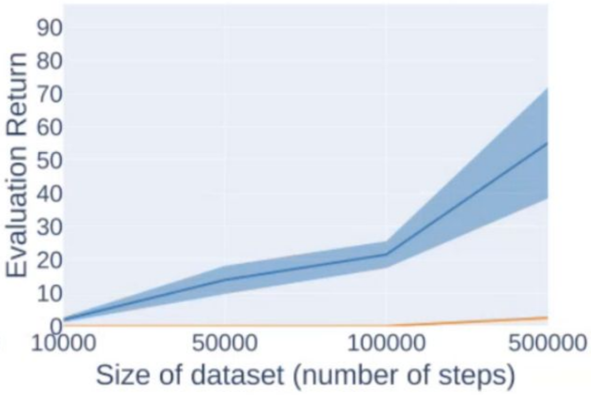
Training time is higher for DT (for a fixed number of epochs). They require more computational resources.



Increasing dataset helps with training loss for CQL as well



Evaluation returns are flat around zero for lower dataset sizes, and then increase towards 500,000



Decision Transformer
CQL Conservative Loss
CLQ TD Loss

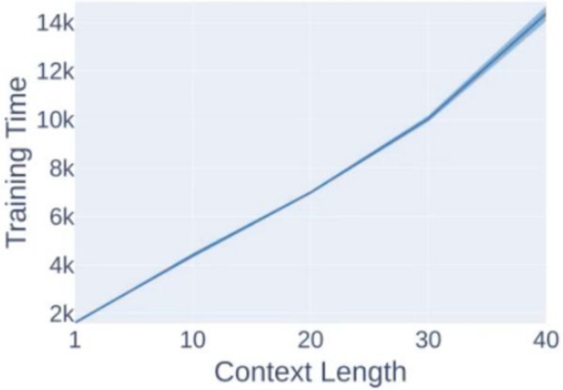
Massachusetts Institute of Technology

ABLATION STUDY

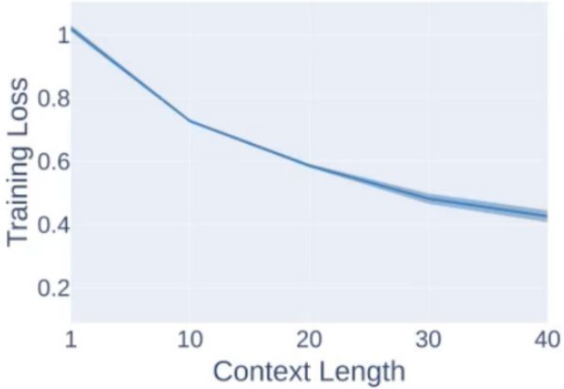
2. HOW SENSITIVE ARE DECISION TRANSFORMERS TO THE CONTEXT SIZE?

- We analyzed model performance across various **context lengths** to assess its efficiency and accuracy
- For too long context lengths, the model **overfits**, with low training loss but decreasing evaluation returns

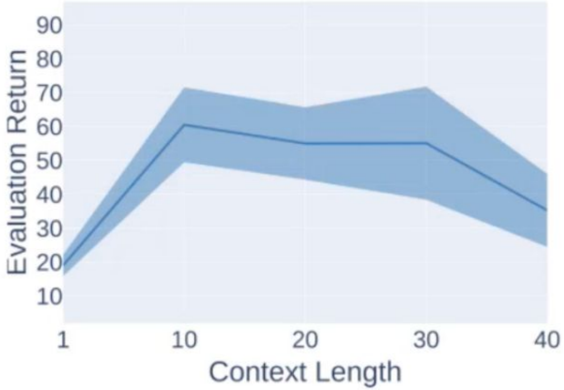
Training time increases linearly for longer context lengths



Training loss decreases substantially as context length increases



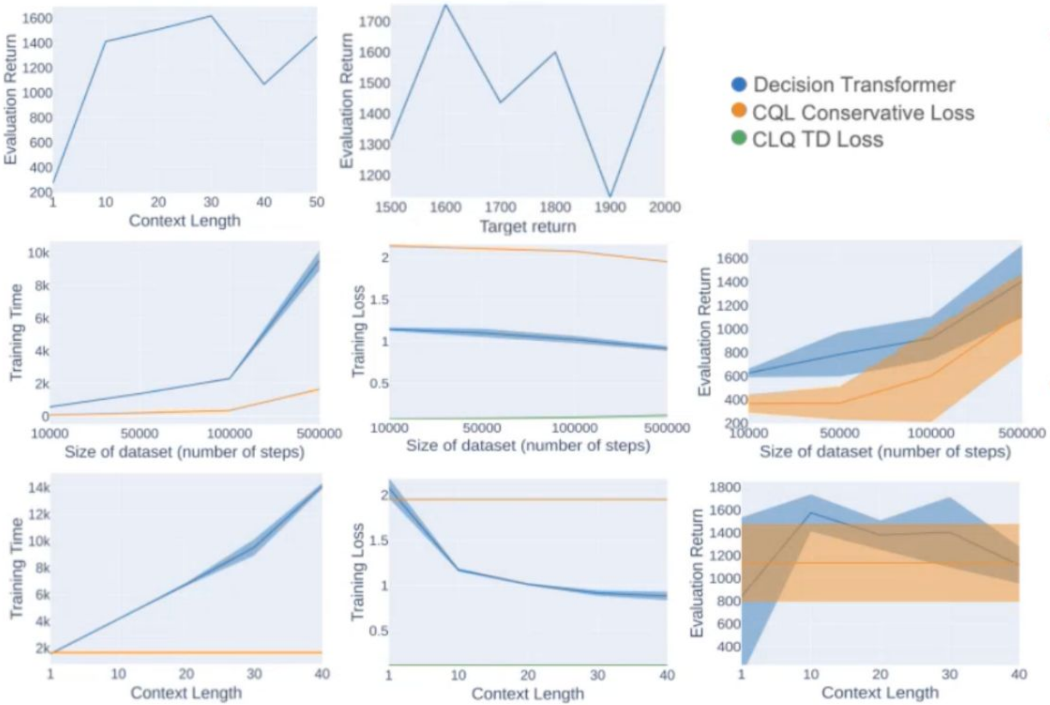
Evaluation Return hits a sweet spot around context length = 10



Massachusetts Institute of Technology

GENERALIZABILITY WITH NEW GAMES: MS PACMAN

HOW GENERALIZABLE ARE DECISION TRANSFORMERS WHEN APPLIED TO OTHER GAME ENVIRONMENTS LIKE PACMAN?



- We performed a hyperparameter search to find the **optimal context length** (30) and **target return** (1600)
- The two models show the same trends across size of the dataset - higher training time, decreasing training loss and increasing evaluation returns, but:
  - CQL requires less computational resources than DT (each epoch is much faster)
  - CQL has consistently lower evaluation returns than DT
- Because Context Length is not a parameter in CQL, its KPIs do not vary with this experiment, but:
  - DT shows increasing training time for higher context lengths
  - DT shows decreasing train loss, yet also signs of overfitting: evaluation returns peak at CL = 10, and eventually start decreasing

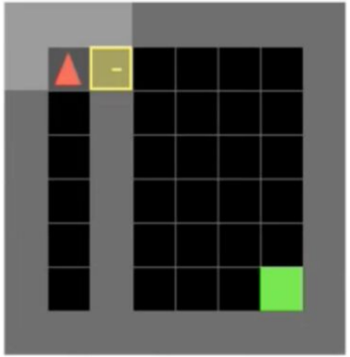
Massachusetts Institute of Technology

LONG TERM CREDIT ASSIGNMENT PROBLEM

Objectives

- Validating Chen et al.'s results on a **long-term credit assignment** problem (reward only happens late in the trajectory)
- **Problem:** No dataset for Key-to-Door environment

Expert (trained with PPO)



Key-to-Door 8x8

- **State Space**: 7x7x3 image (what is observable for the agent)
- **Action Space**: dimension = 7
  - [0] turn left, [1] turn right, [2] move forward, [3] pick up an object, [4] drop object, [5] activate an object, [6] Done
- **Rewards**: reward of 1 assigned if the agent reaches the goal
- **Difficult environment:** The agent needs to use the key to open the door and then get to the goal

Step 1: Solving the problem with PPO

- We couldn't solve the problem directly for Key-to-Door 8x8 using PPO. Instead our successful strategy was:
  - Train an agent using PPO on Key-to-Door 5x5 (100,000 steps)
  - Use this agent as a warm start for further training using PPO on Key-to-Door 6x6 (100,000 steps) and Key-to-Door 8x8 (100,000 steps).

Massachusetts Institute of Technology

CONCLUSION

Decision Transformers provide a framework for modelling decision sequences in complex environments

This analysis highlighted DT's robustness and adaptability across different game environments like Breakout and MsPacman

Strengths and Limitations of Decision Transformers

Strengths

- **Adaptability**: In environments requiring **complex sequential decision-making** it excels due to its transformer architecture.
- **Sample Efficiency**: Performs better than CQL under a fixed number of training steps and samples

Weaknesses

- **Computationally Intensive**: Increased **context lengths** raise computational demands, impacting **scalability**
- **Data Dependency**: Performance relies on the **quantity** and **quality** of training data, with poor performance in **data-sparse scenarios and suboptimal datasets.**

Decision Transformers vs. Conservative Q-Learning

- **Training & Efficiency**: Overall, CQL agents require less computational resources than the Decision Transformer's
- **Model Robustness**: DT outperforms CQL in all games and experiments tested

Massachusetts Institute of Technology

THANK YOU

Massachusetts
Institute of
Technology