Goal-Conditioned Decision Transformer for Multi-Goal Offline Reinforcement Learning

Paweł Gajewski[0000−0003−0931−2476], Dominik Żurek[0000−0001−5329−1452], Marcin Pietroń[0000−0001−9357−9231], and Kamil Faber[0000−0003−4221−0017]

Faculty of Computer Science, AGH University of Krakow, Krakow, Poland
pgajewski@agh.edu.pl

Abstract. Reinforcement learning (RL) in robotics faces significant hurdles regarding sample efficiency and generalization across varying goals. While Offline RL mitigates the need for costly online interactions, its integration with goal-conditioned policies and transformer-based architectures remains underexplored. We introduce a Goal-Conditioned Decision Transformer adapted for offline multi-goal robotics. By explicitly incorporating goal states into the sequence modeling framework, our approach efficiently solves varying tasks using only pre-collected data. We validate this method on a newly released offline dataset for the Franka Emika Panda platform. Experimental results demonstrate that our approach outperforms state-of-the-art online baselines in complex tasks and maintains robustness in sparse-reward settings, even with limited expert demonstrations.

Keywords: Data-efficient learning · decision transformer · multi-goal learning · offline reinforcement learning · reinforcement learning · robotic manipulation · robotics · transformer architectures

1 Introduction

Reinforcement learning (RL) has driven significant progress in robotics [1]. However, standard online algorithms require continuous environment interaction, making them impractical for real-world hardware due to safety concerns, wear and tear, and time constraints. While simulations offer a partial solution, they often suffer from the "sim-to-real" gap.

To address these limitations, *Offline RL* enables agents to learn optimal policies entirely from pre-collected datasets [2]. This paradigm allows for the reuse of historical data and eliminates the need for risky online exploration. However, training on a single fixed task is insufficient for general-purpose robotics. *Multi-goal RL* [3] addresses this by requiring agents to reach varying goals within the same environment structure, necessitating policies that are explicitly conditioned on the desired outcome.

Concurrently, the *Decision Transformer* (DT) [4] has reframed RL as a conditional sequence modeling problem, leveraging the Transformer architecture [5] to generate actions based on desired returns. While DTs have shown promise in

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_22

standard tasks, their application to multi-goal offline scenarios remains underexplored.

In this work, we bridge these domains by introducing a Goal-Conditioned DT. We modify the standard DT architecture to explicitly process goal information, allowing it to solve multi-goal robotic tasks using only offline data. We validate this approach on a custom dataset generated using the Franka Emika Panda environment [6].

The contributions of this paper are threefold: (1) We extend the DT to handle multi-goal settings by incorporating explicit goal-conditioning into the input sequence; (2) We release a specialized offline RL dataset for multi-goal robotic manipulation on the Franka Emika Panda platform; and (3) We demonstrate that our approach outperforms state-of-the-art online methods (TQC+HER) and Behavioral Cloning baselines in offline settings, even under sparse reward signals.

2 Related Work

Reinforcement learning (RL) has evolved from tabular methods to Deep RL (DRL), enabling high-dimensional continuous control [7, 8]. In robotic control, off-policy actor-critic methods have become the standard due to their sample efficiency. Soft Actor-Critic (SAC) [9] utilizes maximum entropy to improve stability, while Truncated Quantile Critics (TQC) [10] further alleviates overestimation bias through distributional critics and truncation. TQC currently serves as a state-of-the-art baseline for continuous control.

2.1 Multi-Goal RL and HER

Standard RL typically trains for a fixed goal. To improve adaptability, Multi-Goal RL conditions the policy on a desired goal input, requiring the agent to generalize across varying targets [3]. These environments often utilize sparse, binary rewards, making exploration difficult. Hindsight Experience Replay (HER) [11] overcomes this by relabeling failed trajectories as successful ones for the specific goals that were actually achieved. The combination of TQC and HER represents a robust baseline for multi-goal robotic tasks.

2.2 Transformers in RL

While Transformers [5] dominate language and vision, their application to RL is a recent development. The DT [4] reframes RL as conditional sequence modeling rather than value function approximation. By conditioning an autoregressive model on desired returns, states, and actions, DT matches or exceeds offline RL baselines on standard benchmarks. Recent extensions, such as the Multi-Objective DT [12], have applied this to multi-objective optimization, but the specific intersection of goal-conditioned, sparse-reward robotic control in an offline setting remains an active area of research that we address in this work.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_22

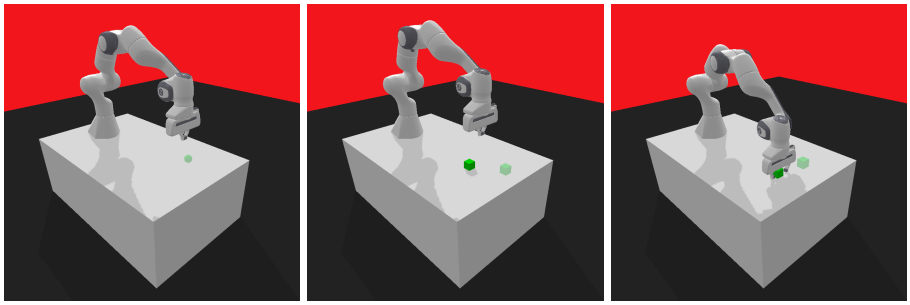

Fig. 1. Multi-goal robotic environments: Reach, Push, PickAndPlace.

3 Methodology

3.1 Decision Transformer

We employ the DT architecture, which reframes offline reinforcement learning as a conditional sequence modeling problem. Unlike methods that fit value functions or compute policy gradients, the DT outputs optimal actions using a masked Transformer [5]. This architecture allows the model to effectively model trajectories by conditioning predictions on desired future outcomes.

The input sequence consists of returns-to-go $\hat{R}_t$, states s_t, and actions a_t:

$$\tau = \{\hat{R}_1, s_1, a_1, \ldots, \hat{R}_T, s_T, a_T\} \tag{1}$$

The return-to-go at timestep t is defined as $\hat{R}_t = \sum_{t'=t}^{T} r_{t'}$, representing the cumulative future reward. This conditioning enables the generation of actions required to achieve a specific target return.

3.2 Dataset

The DT is trained entirely offline. We generated two primary dataset types using the *Panda-Gym* suite: *expert* and *random*. Expert datasets were created using converged TQC agents evaluated for 1 million timesteps. Random datasets were generated by agents sampling actions uniformly.

To evaluate data efficiency, we also created mixtures of expert and random data with varying ratios and subsets. We publicly release the dataset to support reproducibility [1].

3.3 Goal-conditioned Decision Transformer

We utilize multi-goal robotic environments [13] where the observation space is a dictionary comprising the current observation o_t, the desired goal g_d, and the achieved goal g_a. Consequently, the state at timestep t is a tuple $s_t = (o_t, g_d, g_a)$.

[1] https://huggingface.co/datasets/lubiluk/panda-gym-offline

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_22

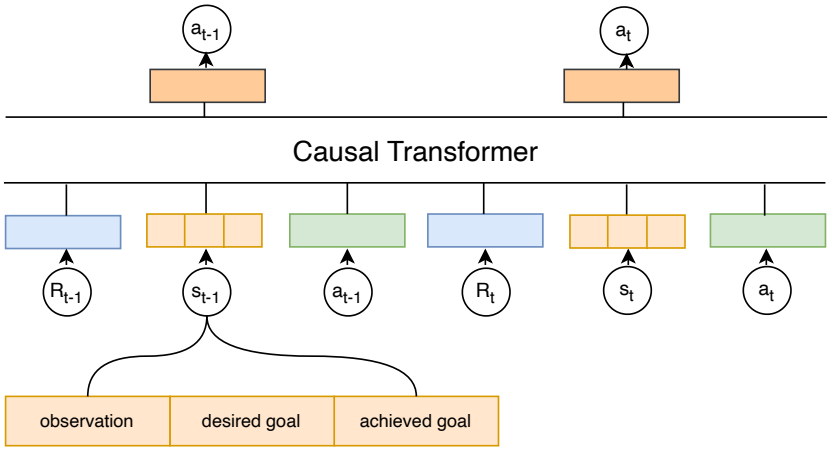

Fig. 2. Goal-conditioned Decision Transformer for multi-goal RL environments

In contrast to standard DT conditioning on scalar returns, goal-conditioning provides structured, spatially meaningful supervision, which is particularly important in sparse-reward multi-goal settings where identical returns may correspond to qualitatively different objectives.

To adapt the DT to this structure, we flatten and concatenate these vectors into a single input vector:

$$s_t = [o_t^{(0)}, \ldots, o_t^{(n)}, g_d^{(0)}, \ldots, g_d^{(m)}, g_a^{(0)}, \ldots, g_a^{(m)}] \tag{2}$$

Including the achieved goal g_a allows the transformer to infer progress toward the desired goal implicitly, without requiring an explicit distance metric or reward shaping.

This modification incorporates goal-related information directly into the sequence, enabling the policy to generate actions that drive the agent toward g_d, as illustrated in Fig. 2.

Unlike standard goal-conditioned policies that treat goals as part of the state input to a reactive policy, our approach conditions an autoregressive sequence model on goal information across the entire trajectory, enabling temporally extended credit assignment with respect to goal achievement.

3.4 Tasks and Rewards

We evaluate the agent on a Franka Emika Panda robotic arm [6] in three tasks: *Reach* (position end-effector), *Push* (move a cube), and *PickAndPlace* (lift and move an object) illustrated by environment screenshots in Fig. 1. We test performance under two reward structures: *sparse*, where the agent receives 0 for success and -1 otherwise; and *dense*, calculated as the negative Euclidean distance between the achieved goal g_a and desired goal g_d.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_22

A rollout is considered successful if the environment-defined success condition is met at any timestep within the evaluation horizon.

4 Experimental setup

We compare the DT against two baselines: Behavioral Cloning (BC) and Truncated Quantile Critics with Hindsight Experience Replay (TQC+HER). Our evaluation addresses five research questions: **(RQ1)** Can DT match state-of-the-art online algorithms (TQC+HER)?; **(RQ2)** How does DT compare to BC?; **(RQ3)** How does DT handle sparse vs. dense rewards?; **(RQ4)** What is the minimum dataset size required?; and **(RQ5)** How does the expert-to-random data ratio impact performance?

Experiments are conducted using 3 random seeds, with agents evaluated over 10,000 timesteps. All results are averaged over three random seeds; while limited, we observed low variance across runs. During evaluation, we condition the DT on a target return of 0 (perfect optimality) for sparse settings and 0 (zero distance) for dense settings, prompting the model to generate the most optimal trajectory. We report average return and success rate. To address **RQ1**–**RQ3**, we train DT and BC on the full 1-million transition expert dataset and compare against TQC+HER trained for an equivalent number of online steps. For **RQ4**, we evaluate data efficiency by training DT on subsets ranging from 100k to 1M expert transitions. For **RQ5**, we analyze resilience to noise by fixing dataset size at 1M but varying the expert data ratio (0%, 25%, 50%, 75%, 100%) mixed with random trajectories.

Hyperparameters for TQC+HER are taken from Stable-Baselines-Zoo [14], while DT utilizes the original paper's settings. The code is publicly available[2].

While TQC+HER is trained online and thus has access to interaction during learning, we include it as a strong upper-bound baseline commonly used in robotic manipulation benchmarks.

5 Results and Discussion

We present the experimental evaluation on full (1M transitions) and subset datasets, addressing the research questions formulated previously.

5.1 Comparative Performance (RQ1, RQ2, RQ3)

Table 1 details the performance of DT against BC and TQC+HER. In the simplest task, *Reach*, all methods achieve near-perfect success. However, in the more complex *Push* and *PickAndPlace* environments with Dense Rewards, DT consistently outperforms both TQC+HER and BC in average return and success rate (RQ1, RQ2). Notably, DT achieves these results with significantly lower training time (80 mins) compared to the online TQC+HER baseline (240 mins).

[2] https://github.com/lubiluk/cldt

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_22

Table 1. Performance comparison: Return / Success Rate (%)

Method	Metric	Reach	Push	PickAndPlace
		Dense Reward		
DT	Ret / Succ	-0.21 / 100.0	**-0.95** / **99.5**	**-1.30** / **98.9**
BC	Ret / Succ	-0.21 / 100.0	-1.20 / 95.9	-1.35 / 97.9
TQC+HER	Ret / Succ	-0.21 / 100.0	-1.04 / 98.7	-1.35 / 98.7
		Sparse Reward		
DT	Ret / Succ	**-1.72** / **100.0**	-8.26 / 95.0	**-7.63** / **97.8**
BC	Ret / Succ	-1.76 / 100.0	-8.18 / 94.6	-9.01 / 94.9
TQC+HER	Ret / Succ	-1.82 / 100.0	**-4.54** / **99.5**	-16.96 / 77.0

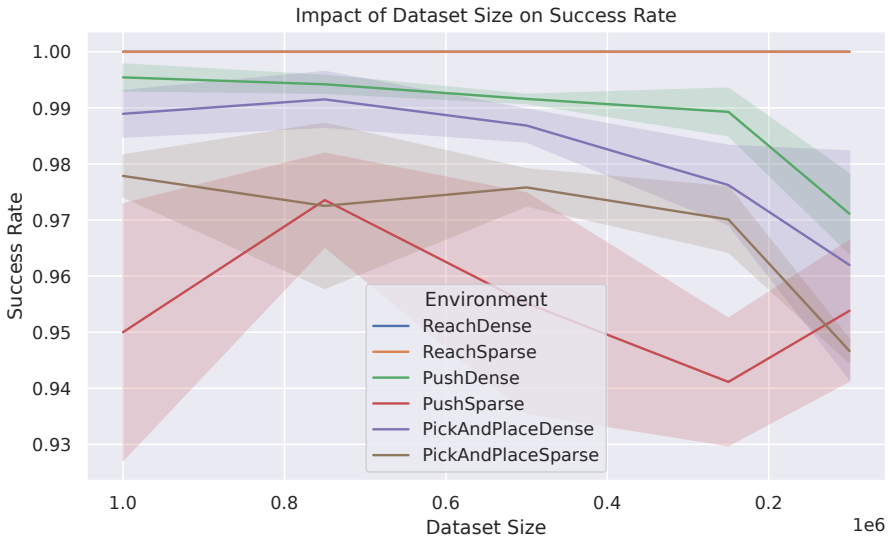

Fig. 3. Impact of dataset size on success rate.

Under Sparse Rewards, DT demonstrates superior robustness and efficiency (RQ3). In *PickAndPlace*, DT achieves a significantly higher return (-7.63 vs -16.96), indicating it solves the task in fewer steps than the baseline, while maintaining a high success rate (97.8%). In contrast, TQC+HER suffers from instability in the sparse *PickAndPlace* task, where its success rate drops significantly to 77.0%.

5.2 Data Efficiency (RQ4)

We analyzed the impact of dataset size on performance (Fig. 3). For *Reach*, performance is invariant to size. In dense reward settings for *Push* and *PickAndPlace*, performance declines only when data falls below 250k samples; yet, even at 100k samples, DT maintains $> 96\%$ success. Sparse rewards introduce higher variance and a sharper performance drop below 250k samples.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_22

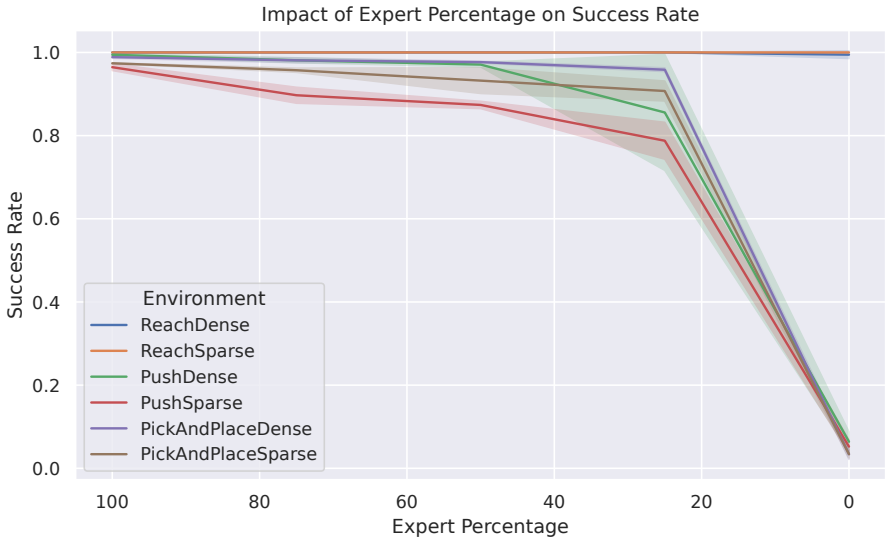

Fig. 4. Impact of expert data percentage.

5.3 Expert Data Ratio (RQ5)

Fig. 4 illustrates DT's sensitivity to trajectory quality. Expert data, i.e. trajectories from a policy that achieves near-optimal performance, was mixed with ones from a random policy. While performance degrades as the ratio of expert data decreases, DT remains surprisingly robust, maintaining $> 80\%$ success in complex tasks even when expert data constitutes only 25% of the dataset. Below this threshold, performance drops sharply.

6 Conclusions & Future Work

We presented a goal-conditioned Decision Transformer adapted for offline multi-goal robotics by explicitly incorporating goal states into the input sequence. We also release a new offline dataset for reproducibility. We hypothesize that this robustness stems from the Transformer's ability to perform long-horizon credit assignment via self-attention, which is particularly effective in sparse-reward settings. Our results demonstrate that this approach can outperform the expert policy used to generate the training data, particularly in complex tasks like PickAndPlace. DT also exhibits robustness to suboptimal demonstrations, maintaining strong performance even when up to approximately 75% of the trajectories are random, and remains effective under sparse reward settings. Future work will extend this framework to continual learning settings and training with heterogeneous expert trajectories.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_22

Acknowledgments. This work was supported by funds assigned by Polish Ministry of Science and Higher Education to AGH University of Krakow, and by PLGrid HPC infrastructure (ACK Cyfronet AGH, grant no. PLG/2025/018713).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Murtaza Dalal, Deepak Pathak, and Ruslan Salakhutdinov. Accelerating robotic reinforcement learning via parameterized action primitives. In *NeurIPS*, 2021.
2. Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 104–114. PMLR, 13–18 Jul 2020.
3. Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, Vikash Kumar, and Wojciech Zaremba. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv*, 2018.
4. 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. 2022.
5. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. in advances in neural information processing systems. 2017.
6. Quentin Gallouédec, Nicolas Cazin, Emmanuel Dellandréa, and Liming Chen. panda-gym: Open-Source Goal-Conditioned Environments for Robotic Learning. *4th Robot Learning Workshop: Self-Supervised and Lifelong Learning at NeurIPS*, 2021.
7. Richard Sutton and Andrew Barto. Reinforcement learning: An introduction. 1979.
8. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *NIPS*, 2013.
9. Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *ICML*, pages 1856–1865, 2018.
10. Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. *ICML'20: Proceedings of the 37th International Conference on Machine Learning*, pages 5556–5566.
11. Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.
12. Abdelghani Ghanem, Philippe Ciblat, and Mounir Ghogho. Multi-objective decision transformers for offline reinforcement learning. 2023.
13. G Brockman. Openai gym. *arXiv*, 2016.
14. Antonin Raffin. Rl baselines3 zoo. `https://github.com/DLR-RM/rl-baselines3-zoo`, 2020.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_22