Dreamer Online Decision Transformer: Enhanced Decision Transformer Learning through Actor-Critic Trajectory Integration

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Advancements in reinforcement learning have led to the development of sophisticated models capable of learning complex decision-making tasks. However, efficiently integrating world models with decision transformers remains a challenge. In this paper, we introduce a novel approach that combines the Dreamer algorithm's ability to generate anticipatory trajectories with the adaptive learning strengths of the Online Decision Transformer. Our methodology enables parallel training where Dreamer-produced trajectories enhance the contextual decision-making of the transformer, creating a bidirectional enhancement loop. We empirically demonstrate the efficacy of our approach on a suite of challenging benchmarks, achieving notable improvements in sample efficiency and reward maximization over existing methods. Our results indicate that the proposed integrated framework not only accelerates learning but also showcases robustness in diverse and dynamic scenarios, marking a significant step forward in model-based reinforcement learning.

1 Introduction

Given the recent success of transformer architectures [21], the general framework of the Decision Transformer (DT) is designed for rapid adaptation and enhanced computational rewards by leveraging pre-training data in an offline setting [4]. Building upon the Decision Transformer, the Online Decision Transformer (ODT) is tailored for online reinforcement learning (RL) settings where decisions must be made in real-time based on streaming data [29], while simultaneously learning from the dataset [7]. The key innovation of the ODT lies in its ability to continuously integrate new experiences and dynamically update the policy as new data arrives. This capability is crucial in non-stationary environments where the underlying dynamics may change over time, necessitating timely policy adaptations.

At the core of the Decision Transformer architecture, the decision transformer maintains a replay buffer of recent experiences, utilizing trajectories of states, actions, and rewards [12]. This buffer is used to fine-tune the policy network at regular intervals, ensuring that the decision-making strategy remains aligned with the most recent data. This process provides the agent with the capacity to effectively respond to evolving situations [14]. By combining the transformer's ability to process sequences with online learning, the ODT facilitates a more robust and adaptive approach to decision-making in dynamic environments [1].

Similarly, the Dreamer algorithm [8], another popular reinforcement learning approach, utilizes world models to "dream" or simulate future states. This capability allows the agent to anticipate the outcomes of its actions without direct interaction with the actual environment, thereby enhancing the efficiency of the learning process by reducing the need for extensive real-world data [10]. Dreamer operates by learning a latent dynamics model of the environment [9], which captures transition probabilities and reward functions. Once trained, this model can generate synthetic trajectories of states, actions, and rewards that the agent can use to improve its policy. The fundamental principle is that by learning in the space of latent representations, the algorithm can perform planning and credit assignment more effectively, even in high-dimensional state spaces [2]. By imagining outcomes and backpropagating the future rewards, Dreamer optimizes the policy to maximize expected returns. This not only conserves resources but also enables safer training, as the agent can explore various strategies in a simulated environment before executing them in the real world. Consequently, Dreamer is capable of developing sophisticated behaviors even in complex environments with sparse rewards [16].

Building onto the success of Online Decision Transformer, *we aim to find out whether combining the Online Decision Transformer with Dreamer will result in a higher reward learning.*

To this end, we proposed a novel algorithm: *Dream-to-Control-for-Online-Decision-Transformer (DODT)*. This novel framework uses the base framework of Online Decision Transformer and through a paralleled trained dreamer, the transfer of enhanced trajectory from dreamer to ODT can benefit the overall model. Through numerous experiments, DODT can utilizes the success of ODT and Dreamer, achieving a higher reward.

Contributions. We conclude our contributions from three perspectives.

1. **Parallel Training Architecture:** We present the first and novel parallel training methodology that simultaneously leverages the Dreamer model's trajectory generation and the Online Decision Transformer's adaptive learning capabilities, providing a symbiotic framework for decision-making.
2. **Trajectory-Informed Decision Making:** Our integration uniquely enables the Online Decision Transformer to be informed by high-fidelity trajectories from the Dreamer, thus enhancing its contextual understanding and response strategies in complex environments.
3. **Cross-Model Feedback Mechanism:** We introduce a feedback loop between the Dreamer and the Online Decision Transformer. Our integrated approach demonstrates superior performance across a variety of challenging benchmarks, surpassing traditional methods in terms of sample efficiency and reward maximization.

1.1 Related Work

Decision Transformer: Recent advancements in Decision Transformers have significantly expanded their capabilities and applications in reinforcement learning. A bootstrapping method was introduced to augment data generation for both online and offline Decision Transformers, enhancing training datasets significantly [24]. Additionally, innovative probabilistic learning objectives and max-entropy sequence modeling have been integrated to balance exploration and exploitation dynamically, addressing the demands of online reinforcement learning environments for decision transformers [15]. Further enhancements include a hierarchical decision-making structure, where high-level policies generate prompts that guide low-level action generation, improving decision granularity [16], and the combination of trajectory modeling with value-based methods, which aligns specified target returns with expected action returns to boost performance in stochastic settings [25]). Additionally, leveraging latent diffusion models for optimizing suboptimal trajectory portions from static datasets [22] and employing robust planning frameworks that treat planning as latent variable inference have further enhanced the long-term decision-making capabilities of Decision Transformers [13].

Dreamer: At the same time, the Dreamer have been enhanced through various innovative approaches as well. The Dreamer model has been extensively advanced by integrating transformers to enhance the deterministic state prediction from observations [26]. Transitioning from recurrent neural networks to transformer networks within the world model has significantly improved the efficiency of state predictions [3, 6]. The adaptation of Dreamer for multi-task reinforcement learning uses diffusion models to optimize offline decision-making [11]. Extensions to the Dreamer framework allow handling of diverse tasks through world models that predict future states and rewards from abstract representations [10]. Furthermore, the use of prototypical representations instead of high-dimensional observation reconstructions [5], along with conditional diffusion models for long-horizon predictions

[28], and enhancement of exploration using latent state marginalization [27], collectively push the boundaries of model-based RL.

Teacher to Student Model: The Student to Teacher model leverages the dynamics of guided learning to enhance the efficiency and scalability of reinforcement learning systems. The TGRL algorithm integrates the teacher-student learning framework with reinforcement learning, facilitating enriched policy learning experiences [19]. Curriculum learning approaches have also been significant, framing task sequencing within a meta Markov Decision Process to systematically improve sample efficiency [18]. The application of large language models as teachers to guide smaller, specialized student agents offers a novel approach to scaling down complex decision processes [30]. Furthermore, advancements in multi-agent systems, where experiences are shared between agents, enhance collective learning capabilities, demonstrating improved scalability and efficiency [23].

2 Preliminaries

2.1 Online Decision Transformer

Algorithm 1 Online Decision Transformer (ODT)

```
Input: offline data T_offline, rounds R, exploration RTG T_online, buffer size N, gradient iterations I, pre-trained policy π_θ.
Initialization: Replay buffer T_replay ← top N trajectories in T_offline.
for round = 1, ..., R do
    Trajectory τ ← Rollout using M and π_θ(·|s, g(T_online)).
    T_replay ← (T_replay \ {oldest trajectory}) ∪ {τ}.
    π_θ ← Finetune ODT on T_replay for I iterations using Training Main Loop.
end for
```

Online Decision Transformer (ODT) represents a significant advance in the application of transformers to reinforcement learning (RL). It extends the Decision Transformer (DT) architecture to online settings, adapting the transformer architecture for dynamic environments and real-time decision-making tasks. This adaptation is crucial for RL applications where an agent must continuously learn and adapt based on new data while interacting with an environment. In traditional reinforcement learning, decision-making often relies on policies learned from historical data or through iterative interactions with an environment. These methods can be inefficient and slow to adapt to changes in dynamic scenarios. The ODT framework addresses these challenges by leveraging the sequential processing capabilities of transformers to model policies based on both past and current interactions, integrating learning and decision-making in an online fashion.

The core of ODT is a transformer architecture trained to optimize a sequence modeling objective that predicts the next action based on a history of states, actions, and rewards. Given a history encoded as sequences, the ODT models the conditional probability of actions given past experiences, formulated as:

$$\pi(a_t|s_t, g_t) \approx P(a_t|\text{context}),$$

where g_t represents the return-to-go, a sum of future rewards, and s_t denotes the current state. The context comprises past states, actions, and achieved rewards up to time t.

The policy is refined using a replay buffer T_{replay} that stores trajectories:

$$T_{\text{replay}} = \{\tau_1, \tau_2, \dots, \tau_N\},$$

where each τ_i is a trajectory containing sequences of states, actions, and rewards. During training, this buffer is continuously updated by replacing the oldest trajectories with new ones obtained from recent environment interactions, ensuring that the policy adapates to the most recent data.

The ODT utilizes the transformer's capability to process sequences of data to dynamically update its policy based on the replay buffer. The policy π_θ is optimized by fine-tuning the transformer model on sequences drawn from T_{replay}, using the objective:

$$\pi_\theta \leftarrow \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t\right],$$

where r_t is the reward at time t and γ is the discount factor.

During online interactions, the ODT first collect data from the environment using the current policy π_θ then update the replay buffer T_{replay} by incorporating new trajectories and discarding the oldest. After that, it refines the policy π_θ by training on a sampled batch from T_{replay}. This continuous loop of feedback and adaptation allows the ODT to maintain a policy that is responsive to the evolving dynamics of the environment. The integration of a transformer-based sequence model with an RL policy training framework enables the ODT to leverage the strengths of both sequence modeling and reinforcement learning techniques.

2.2 Dreamer

Algorithm 2 Dreamer

Initialize dataset D with S random seed episodes.
Initialize neural network parameters θ, ϕ, ψ randomly.
while not converged **do**
    for update step $c = 1 \ldots C$ **do**
        // Dynamics learning
        Draw B data sequences $\{(\mathbf{a}_t, \mathbf{o}_t, r_t)\}_{t=k}^{k+L} \sim D$.
        Compute model states $s_t \sim p_\theta(s_t | s_{t-1}, a_{t-1}, o_t)$.
        Update θ using representation learning.
        // Behavior learning
        Imagine trajectories $\{(s_r, a_r)\}_{r=t}^{t+H}$ from each s_t.
        Predict rewards $E(r_t | s_t)$ and values $V(s_t)$.
        Compute value estimates $V_\psi(s_t)$.
        Update $\phi \leftarrow \phi + \alpha \nabla_\phi \sum_{t=r}^{t+H} V_\psi(s_t)$.
        Update $\psi \leftarrow \psi - \alpha \nabla_\psi \frac{1}{2} \sum_{t=r}^{t+H} (\psi(s_t) - V_\psi(s_t))^2$.
    end for
    // Environment interaction
    $\mathbf{o}_1 \leftarrow$ env.reset()
    for time step $t = 1 \ldots T$ **do**
        Compute $s_t \sim p_\theta(s_t | s_{t-1}, a_{t-1}, o_t)$.
        Compute $a_t \sim q_\phi(a_t | s_t)$ with the action model.
        Add exploration noise to a_t.
        Execute action a_t and observe reward r_t and new observation $\mathbf{o}_{t+1}$.
        Add experience to dataset $D \leftarrow D \cup \{(\mathbf{o}_t, a_t, r_t)\}_{t=1}^{T}$.
    end for
end while

The Dreamer algorithm represents a significant step forward in latent dynamics learning for control by leveraging model based reinforcement learning, mostly for model based RL. By abstracting the observation space into a compact latent space, the dreamer can efficiently predicts future states and rewards, enabling it to plan and learn policies entirely through latent imagination.

The world model in Dreamer consists of three key components:

- **Representation Model**: $p(s_t | s_{t-1}, a_{t-1}, o_t)$, which encodes observations into a latent state, integrating past actions and states.
- **Transition Model**: $q(s_t | s_{t-1}, a_{t-1})$, which predicts the next latent state given the current state and action, facilitating the generation of future trajectories without real-world interaction.
- **Reward Model**: $q(r_t | s_t)$, which estimates the immediate reward from the current latent state, crucial for evaluating the desirability of states within imagined trajectories.

Dreamer utilizes latent imagination to learn optimal behaviors by simulating trajectories in the latent space, derived from the learned world model. This approach allows Dreamer to perform efficient, farsighted planning by propagating value estimates backward through imagined trajectories. The key mathematical formulations in this process include the **Action Model**: $a_\tau \sim q_\phi(a_\tau|s_\tau)$, which optimizes actions to maximize expected returns, and the **Value Model**: $v_\psi(s_\tau) \approx \mathbb{E}\left[\sum_{\tau=t}^{t+H} \gamma^{\tau-t} r_\tau | s_\tau\right]$, which estimates the value of latent states over a finite horizon H.

The optimization objectives are:

- **Action Optimization**:

$$\max_{\phi} \mathbb{E}_{q_\theta, q_\phi}\left[\sum_{\tau=t}^{t+H} V_\lambda(s_\tau)\right],$$

  aiming to find the policy parameters ϕ that maximize the sum of discounted future values estimated by the value model.

- **Value Regression**:

$$\min_{\psi} \mathbb{E}_{q_\theta, q_\phi}\left[\frac{1}{2}\sum_{\tau=t}^{t+H} \|v_\psi(s_\tau) - V_\lambda(s_\tau)\|^2\right],$$

  minimizing the prediction error of the value model, aligning it with the computed value estimates to ensure consistency and stability in policy evaluation.

Dreamer's integration of deep learning with latent variable models for reinforcement learning showcases several advantages over both traditional model-based and model-free methods. By optimizing behavior in a compact, learned representation of the world, Dreamer achieves remarkable data efficiency and scalability, effectively handling environments with complex, high-dimensional sensory inputs. This makes it a powerful tool for a wide range of applications, from robotics to virtual simulations, where sample efficiency and rapid adaptation to new scenarios are critical.

3 Algorithm: Dreamer Online Decision Transformer for RL

Algorithm 3 DODT: Parallel ODT Training with Dreamer Trajectories

```
1:  Input: offline data T_offline, exploration RTG T_online, buffer sizes N, D, training rounds R, gradient
    iterations I.
2:  Initialization: Initialize replay buffers T_replay and D as per Algorithms 1 and 2.
3:  Initialize policies π_θ (ODT) and φ, ψ (Dreamer) with pre-training or random weights.
4:  Load environment and set up necessary configurations.
5:  for round = 1, . . . , R do
6:      // Dreamer Interaction Phase (Algorithm 2)
7:      o_1 ← env.reset()
8:      for time step t = 1 . . . T do
9:          Use Dreamer to compute a_t ∼ q_φ(a_t|s_t) for the current state.
10:         Execute a_t in the environment, observe new state o_{t+1}, reward r_t.
11:         Update Dreamer's dataset D ← D ∪ {(o_t, a_t, r_t)}.
12:         Use Dreamer's model to perform learning updates.
13:     end for
14:     // ODT Interaction Phase (Algorithm 1)
15:     τ ← Generate trajectory using Dreamer's π_θ for exploration with RTG T_online.
16:     Update ODT's replay buffer T_replay ← (T_replay \ {lowest reward trajectory}) ∪ {τ}.
17:     Finetune π_θ using ODT on T_replay for I gradient iterations.
18:     // Evaluate Performance
19:     Evaluate the combined performance of Dreamer and ODT.
20:     Log performance metrics.
21: end for
```

Our new algorithm (DODTS, Algorithm 3) integrates the Online Decision Transformer (ODT, Algorithm 1) and Dreamer (Algorithm 2) into a cohesive framework to enhance learning in complex environments. This integration exploits the generative model capabilities of Dreamer and the decision-making prowess of ODT, providing a robust solution to decision-making tasks in dynamic environments.

The algorithm begins with initializing the respective replay buffers: T_{replay} for ODT and D for Dreamer. Model parameters π_θ for ODT and ϕ, ψ for Dreamer are initialized either from pre-trained states or randomly. During each round of training, Dreamer engages with the environment to generate new trajectories, enhancing its generative and predictive capabilities. Concurrently, ODT generates trajectories using its policy π_θ, optimized towards a specific reward-to-go. These trajectories are added to the ODT's replay buffer T_{replay}, and the policy π_θ is fine-tuned based on these new experiences.

In this integrated framework, the models continuously exchange information, where the trajectories generated by Dreamer enhance the contextual dataset for ODT, enabling it to refine its decision-making process with richer environmental feedback. This interaction is further optimized through a series of gradient iterations and buffer updates, ensuring both models evolve towards maximizing their performance in predicting and making effective decisions. The strength of our approach lies in its ability to maintain a continuous loop of feedback and learning between the two models. This not only accelerates the learning process but also enhances the quality of the decision-making and predictive accuracy, leveraging the strengths of both models to address the complexities of the tasks at hand. Our contributions underscore the novelty and impact of this integrated approach, as outlined at the beginning of the document. The parallel training architecture, trajectory-informed decision-making, and cross-model feedback mechanism collectively push the boundaries of what is achievable in autonomous learning systems, setting new benchmarks for efficiency and effectiveness in complex environments.

4 Experiment

Dataset	ODT	DODT
Hopper - medium	**97.94 ± 2.10**	96.84 ± 2.19
Hopper - medium -replay	88.89 ± 6.33	**90.31 ± 3.57**
Walker2d - medium	**76.79 ± 2.30**	75.49 ± 1.82
Walker2d - medium -replay	**76.86 ± 4.04**	74.98 ± 1.45
Half-cheetah - medium	42.16 ± 1.48	**60.93 ± 6.83**
Half-cheetah - medium -replay	40.42 ± 1.61	**57.82 ± 5.79**
Ant - medium	90.79 ± 5.80	**92.01 ± 4.91**
Ant - medium -replay	91.57 ± 2.73	**93.54 ± 6.31**
Sum	605.02	**641.89**

We conducted the experiments within the MuJoCo simulation environment [20], and a detailed comparative analysis was performed between the Online Decision Transformer (ODT) and the Dreamer Online Decision Transformer (DODT). Both of these models were evaluated across a suite of tasks designed to probe their efficacy under varying conditions reflective of real-world complexity.

We analyzed the performance of the Online Decision Transformer (ODT) and the Dreamer Online Decision Transformer (DODT) across various tasks. Results indicate that ODT excels in environments with less complexity, such as "Hopper - medium" and "Walker2d - medium," suggesting better suitability for stable, predictable contexts. In contrast, DODT showcases superior performance in more complex scenarios, including "Half-cheetah - medium" and "Ant - medium," particularly when historical replay is incorporated. This improvement highlights DODT's effective integration of Dreamer's generative modeling with ODT's adaptive decision-making, enhancing its ability to handle environmental variability and uncertainty.

Overall, DODT outperforms ODT with a total score of 641.89 compared to 605.02, demonstrating robust adaptability across varied tasks. This suggests that combining generative trajectory modeling with adaptive decision frameworks may significantly advance reinforcement learning applications requiring high generalization and responsiveness to dynamic conditions.

5 Conclusion and Future Work

In this paper, we introduced the Dreamer Online Decision Transformer (DODT), a novel algorithm that integrates the Dreamer model's trajectory generation into the Online Decision Transformer, enhancing the model's capability to make informed, sequential decisions. Tested within the MuJoCo simulation environment, DODT not only surpasses the Online Decision Transformer (ODT) in terms of total reward achievement but also demonstrates improved sample efficiency and robustness across a variety of dynamic tasks. This integration allows for a deeper understanding of and responsiveness to changing environmental conditions, as DODT leverages Dreamer's ability to simulate and evaluate future states to optimize decision-making strategies in real-time, significantly boosting the system's adaptiveness and overall performance.

Limitations: Despite its effectiveness, the DODT framework has certain limitations that need to be addressed in future work. The computational overhead associated with running two complex models in parallel can be substantial, potentially limiting its applicability in resource-constrained scenarios. Furthermore, while the integration allows for enhanced performance in complex environments, it might introduce additional complexity in tuning and convergence, requiring more sophisticated techniques to manage the interplay between the two models effectively.

Future Directions: For future research, we aim to explore methods to reduce the computational demands of the DODT, potentially through model simplification or more efficient training algorithms. Additionally, we plan to eliminate the reliance on a pre-trained dataset from D4RL. Another promising direction is the exploration of the transfer learning capabilities of DODT, where the model could be pre-trained in a simulated environment and fine-tuned in real-world applications, thereby enhancing its practical utility. Moreover, investigating the scalability of DODT to multi-agent systems and its performance in non-MuJoCo environments would provide deeper insights into the versatility and robustness of the integrated model approach.

References

[1] Bhargava, Prajjwal, et al. "When Should We Prefer Decision Transformers for Offline Reinforcement Learning?" arXiv:2305.14550, arXiv, 11 Mar. 2024. arXiv.org, https://doi.org/10.48550/arXiv.2305.14550.

[2] Brunnbauer, Axel, et al. "Latent Imagination Facilitates Zero-Shot Transfer in Autonomous Racing." 2022 International Conference on Robotics and Automation (ICRA), 2022, pp. 7513–20. IEEE Xplore, https://doi.org/10.1109/ICRA46639.2022.9811650.

[3] Chen, Chang, et al. "TransDreamer: Reinforcement Learning with Transformer World Models." arXiv:2202.09481, arXiv, 18 Feb. 2022. arXiv.org, https://doi.org/10.48550/arXiv.2202.09481.

[4] Chen, Lili, et al. "Decision Transformer: Reinforcement Learning via Sequence Modeling." Advances in Neural Information Processing Systems, vol. 34, 2021, pp. 15084-15097. NeurIPS, https://proceedings.neurips.cc/paper/2021/file/7f489f642a0ddb10272b5c31057f0663-Paper.pdf.

[5] Deng, Fei, et al. "DreamerPro: Reconstruction-Free Model-Based Reinforcement Learning with Prototypical Representations." Proceedings of the 39th International Conference on Machine Learning, in Proceedings of Machine Learning Research 162:4956-4975, 2022. NeurIPS, https://proceedings.mlr.press/v162/deng22a.html.

[6] Ding, Zihan, et al. "Diffusion World Model." arXiv:2402.03570, arXiv, 11 Feb. 2024. arXiv.org, https://arxiv.org/pdf/2402.03570.

[7] Fu, Justin, et al. "D4RL: Datasets for Deep Data-Driven Reinforcement Learning." arXiv:2004.07219, arXiv, 5 Feb. 2021. arXiv.org, https://doi.org/10.48550/arXiv.2004.07219.

[8] Hafner, Danijar, et al. "Dream to Control: Learning Behaviors by Latent Imagination." ICLR 2020, OpenReview.net, 20 Dec. 2019. https://openreview.net/forum?id=S1lOTC4tDS.

[9] Hafner, Danijar, et al. "Learning Latent Dynamics for Planning from Pixels." Proceedings of the 36th International Conference on Machine Learning, in Proceedings of Machine Learning Research 97:2555-2565, 2019. ICML, https://proceedings.mlr.press/v97/hafner19a.html.

[10] Hafner, Danijar, et al. "Mastering Diverse Domains through World Models." arXiv:2301.04104, arXiv, 17 Apr. 2024. arXiv.org, https://arxiv.org/pdf/2301.04104.

[11] He, Haoran, et al. "Diffusion Model Is an Effective Planner and Data Synthesizer for Multi-Task Reinforcement Learning." Advances in Neural Information Processing Systems, vol. 36, 2023. NeurIPS, https://papers.nips.cc/paper_files/paper/2023/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[12] Janner, Michael, et al. "Offline Reinforcement Learning as One Big Sequence Modeling Problem." Advances in Neural Information Processing Systems, vol. 34, Curran Associates, Inc., 2021, pp. 1273–86. Neural Information Processing Systems, https://proceedings.neurips.cc/paper/2021/hash/099fe6b0b444c23836c4a5d07346082b-Abstract.html.

[13] Kong, Deqian, et al. "Latent Plan Transformer: Planning as Latent Variable Inference." arXiv:2402.04647, arXiv, 28 May 2024. arXiv.org, https://doi.org/10.48550/arXiv.2402.04647.

[14] Li, Wenzhe, et al. "A Survey on Transformers in Reinforcement Learning." arXiv:2301.03044, arXiv, 20 Sept. 2023. arXiv.org, https://doi.org/10.48550/arXiv.2301.03044.

[15] Ma, Yi, et al. "Rethinking Decision Transformer via Hierarchical Reinforcement Learning." arXiv:2311.00267, arXiv, 31 Oct. 2023. arXiv.org, https://doi.org/10.48550/arXiv.2311.00267.

[16] Moerland, Thomas M., et al. "Model-Based Reinforcement Learning: A Survey." arXiv:2006.16712, arXiv, 31 Mar. 2022. arXiv.org, https://doi.org/10.48550/arXiv.2006.16712.

[17] Nguyen, Austin. "Fully Online Decision Transformer for Reinforcement Learning", University of Michigan, Fall 2022. https://sled.eecs.umich.edu/media/eecs595_fa22/11_Nguyen_Glasscock.pdf.

[18] Schraner, Yanick. "Teacher-Student Curriculum Learning for Reinforcement Learning." arXiv:2210.17368, arXiv, 31 Oct. 2022. arXiv.org, https://doi.org/10.48550/arXiv.2210.17368.

[19] Shenfeld, Idan, et al. "TGRL: An Algorithm for Teacher Guided Reinforcement Learning." Proceedings of the 40th International Conference on Machine Learning, in Proceedings of Machine Learning Research, 2024. ICML, https://arxiv.org/pdf/2307.03186.

[20] Todorov, Emanuel, Tom Erez, and Yuval Tassa. "MuJoCo: A Physics Engine for Model-Based Control." 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 2012, pp. 5026-5033. https://homes.cs.washington.edu/~todorov/papers/TodorovIROS12.pdf.

[21] Vaswani, Ashish, et al. "Attention Is All You Need." Advances in Neural Information Processing Systems, vol. 30, Curran Associates, Inc., 2017. Neural Information Processing Systems, https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[22] "Reasoning with Latent Diffusion in Offline Reinforcement Learning." arXiv:2309.06599v1, arXiv, 1 Jun. 2022. arXiv.org, https://arxiv.org/pdf/2309.06599v1.

[23] Wang, Jiawei, et al. "Intelligent Vehicle Decision-Making and Trajectory Planning Method Based on Deep Reinforcement Learning in the Frenet Space." Sensors (Basel, Switzerland), vol. 23, no. 24, Dec. 2023, p. 9819. PubMed, https://www.mdpi.com/1424-8220/23/24/9819.

[24] Wang, Kerong, et al. "Bootstrapped Transformer for Offline Reinforcement Learning." Advances in Neural Information Processing Systems, vol. 35, 2022, pp. 3277-3289. NeurIPS, https://papers.nips.cc/paper_files/paper/2022/file/e0ccda3cb17b084a6f43c62cfac4784b-Paper-Conference.pdf.

[25] Wang, Yuanfu, et al. "Critic-Guided Decision Transformer for Offline Reinforcement Learning." Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI), 2024, pp. 15706-15714. AAAI, https://dblp.org/rec/conf/aaai/WangYW0Q24](https://dblp.org/rec/conf/aaai/WangYW0Q24.

[26] Zeng, Catherine, et al. "Dreaming with Transformers" Proceedings of the 36th AAAI Conference on Artificial Intelligence, 2022. AAAI, https://rlg.mlanctot.info/papers/AAAI22-RLG_paper_24.pdf.

[27] Zhang, Dinghuai, et al. "Latent State Marginalization as a Low-Cost Approach for Improving Exploration." Proceedings of the 11th International Conference on Learning Representations, 2023. ICLR, https://openreview.net/pdf?id=b0UksKFcTOL.

[28] Zhao, Siyan, et al. "Decision Stacks: Flexible Reinforcement Learning via Modular Generative Models" Proceedings of the 37th Conference on Neural Information Processing Systems, 2023. NeurIPS, https://arxiv.org/pdf/2306.06253.

[29] Zheng, Qinqing, et al. "Online Decision Transformer." Proceedings of the 39th International Conference on Machine Learning, 2022. ICML, https://arxiv.org/pdf/2202.05607.

[30] Zhou, Zihao, et al. "Large Language Model as a Policy Teacher for Training Reinforcement Learning Agents." arXiv:2311.13373, arXiv, 27 May 2024. arXiv.org, https://arxiv.org/pdf/2311.13373.