DT-Guided DRL: A Transition from Utility-based Decision Theory to Deep Reinforcement Learning

Zelin Wan[1] , **Nithin Alluru**[1] , **Jin-Hee Cho**[1] , **Mu Zhu**[2] , **Ahmed H. Anwar**[3] , **Charles Kamhoua**[3] , **Munindar P. Singh**[4] ,
[1]Virginia Tech
[2]Chinese Academy of Sciences
[3]DEVCOM Army Research Laboratory
[4]North Carolina State University
{zelin, nithin, jicho}@vt.edu, zhumu@cnic.cn, {ahmed.h.hemida.ctr, charles.a.kamhoua.civ}@army.mil, mpsingh@ncsu.edu

Abstract

This paper presents Decision Theory-guided Deep Reinforcement Learning, called *DT-guided DRL*, to address the cold start problem in DRL. By incorporating decision theory, DT-guided DRL improves agents' initial performance, robustness, learning efficiency, and reliability in complex environments. We examine three problem contexts: cart pole, maze navigation, and inverted pendulum. Experimental results show that DT-guided DRL enhances initial guidance and exploration. During early training, it achieves at least 184% higher accumulated reward in the cart pole problem and at least 279% higher accumulated reward in the maze problem. It also provides faster convergence in all problem contexts. This approach leverages human-informed knowledge, offering a foundation for further research.

1 Introduction

Decision theory, particularly utility theory [Fishburn *et al.*, 1979], has long been a cornerstone in fields like economics, management science, and cybersecurity [Rădulescu and others, 2020], offering a structured framework for decision-making under uncertainty [North, 1968]. However, its limitations in handling complex environments, due to oversimplified models and assumptions, restrict its broader applicability. In contrast, *Deep Reinforcement Learning* (DRL) has demonstrated remarkable success in learning intricate behaviors through neural networks that model complex state-action dynamics, particularly in some foundational robot control tasks such as balancing (Inverted Pendulum), navigation (Maze), and coordination (Cart Pole). Yet, the long training times and initial random exploration of DRL pose significant challenges for practical deployment, especially in critical, real-world scenarios like robotics where early failures are unacceptable [Dulac-Arnold *et al.*, 2019].

For researchers and practitioners in robotics, integrating DRL offers transformative potential for enabling autonomous systems to learn complex tasks. However, poor early performance and extensive training hinder its adoption in robotics, where safety and efficiency are paramount. Addressing these limitations is crucial for unlocking DRL's full potential in real-world applications.

To address these challenges, we propose Decision Theory-guided Deep Reinforcement Learning, called *DT-guided DRL*, which incorporates utility functions from decision theory to guide early exploration. This approach mitigates the cold start problem and ensures consistent performance during training, particularly beneficial in robotics where trial-and-error learning can be costly or dangerous.

Our proposed DT-guided DRL approach aims to facilitate the transition from decision theory to DRL and mitigate the cold start problem. By integrating utility functions into the DRL framework, our method allows researchers and practitioners to leverage existing decision theory knowledge while benefiting from DRL's advanced capabilities, easing adoption through a more gradual transition. DT-guided DRL uses utility functions to guide exploration and action selection in the early learning stages, steering the agent toward promising regions of the state space and avoiding poor initial performance.

We evaluate the effectiveness of our approach using three benchmark environments: the classic *cart pole*, *maze navigation tasks*, and *inverted pendulum*. Those serve as proof-of-concept domains to compare DT-guided DRL against standard DRL techniques and relevant baselines, including transfer learning [Zhu and others, 2023], sample efficiency [Lin *et al.*, 2021], and imitation learning [Gros and others, 2020].

This work presents the following **key contributions**:

1. We introduce a novel integration of utility theory, a fundamental component of decision theory, with the DRL framework to enhance its applicability to robotics. This is the first work to systematically merge these paradigms in robotics, establishing a structured transition from decision-theoretic models to DRL. This integration enables informed early-stage exploration, reducing inefficient random search and accelerating learning in reinforcement learning environments.
2. We experimentally demonstrate that DT-guided DRL ef-

fectively mitigates the cold start problem, a major challenge in reinforcement learning. Through extensive evaluations of cart poles, maze navigation, and inverted pendulum, representing robotic control, navigation, and balancing, we show that our method achieves significantly higher initial performance, faster convergence, and greater sample efficiency than standard DRL techniques and relevant baselines, including transfer learning, imitation learning, and sample-efficient RL.

3. Our approach leverages human-informed knowledge to provide structured guidance during early exploration, improving learning stability and robustness in complex environments. This ensures that reinforcement learning can be more reliably applied to real-world robotics, where trial-and-error approaches are costly or unsafe.
4. By bridging decision theory and DRL, our work lays the foundation for a new research direction at the intersection of both fields. The proposed framework not only enhances reinforcement learning efficiency but also opens new avenues for integrating expert knowledge and decision-theoretic principles into autonomous robotic systems, making AI-driven decision-making more reliable, interpretable, and scalable.

2 Related Work

This section overviews existing approaches for enhancing DRL efficiency by reducing convergence time, focusing on *prior-knowledge-enhanced initialization*, *transfer learning*, *imitation learning*, and *sample efficiency*.

Prior-knowledge enhanced initialization improves DRL efficiency by leveraging prior knowledge. Silva and Gombolay [2021] integrated domain-specific rules into RL, boosting performance without extensive data and highlighting its real-world potential. Wang *et al.* [2024] accelerated Hybrid Electric Vehicle (HEV) energy management by initializing neural networks with expert knowledge, improving learning speed and reducing energy consumption. Similarly, Xu and others [2020] applied warm-start Q-learning to HEVs, reducing iterations and enhancing fuel efficiency. Wexler and others [2022] introduced confidence-constrained learning to mitigate performance degradation in Warm-Start RL, balancing policy gradient and constrained learning. While warm starts in DRL and RL accelerate training, they may introduce biased exploration, limiting full-state-action space exploration.

Transfer Learning (TL) has been applied to address the cold start problem in DRL. Approaches like the Hierarchical Deep Reinforcement Learning Network (H-DRLN) for Minecraft have shown superior performance by reusing learned skills across tasks [Tessler and others, 2017]. Similarly, Reinforcement Learning Building Optimizer with Transfer Learning (ReLBOT) applied algorithms from data-rich buildings to new ones, mitigating cold start in smart buildings [Genkin and McArthur, 2022]. However, TL's success depends on selecting a pre-training task closely aligned with the target task. Mismatched knowledge transfer can reduce performance in the target domain.

Imitation Learning (IL), such as Reduction-based Active Imitation Learning (RAIL) algorithm, reduces expert queries using active independently and identically distributed (i.i.d.) learning, showing effectiveness in tasks like Cart-pole [Gros and others, 2020]. Deep Deterministic Policy Gradient with Imitation Learning (DDPG-IL) combines DDPG with imitation learning for autonomous driving, improving convergence and performance [Zou *et al.*, 2020]. However, it relies heavily on high-quality demonstrations and human expertise, struggles with complex spaces, and faces ethical concerns and generalization challenges, adding complexity to its application.

Sample Efficiency (SE) in DRL has advanced significantly. "JueWu-MC" used a hierarchical approach with human demonstrations to address partial observability in open-world games, outperforming baselines in NeurIPS MineRL competitions [Lin *et al.*, 2021]. Option Machines (OMs) leveraged high-level instructions for action selection, excelling in single-task, multi-task, and zero-shot learning [den Hengst and others, 2022]. A DRL method for cloud-native Service Function Chains (SFC) caching tackled cold-start problems using Graph Neural Networks (GNNs), optimizing latency and request acceptance under high-load conditions [Zhang and others, 2022]. Ma and others [2022] introduced a *freshness discounted factor* in prioritized experience replay (PER), improving learning efficiency. Randomized Ensemble Double Q-learning (REDQ) [Chen and others, 2021] achieved state-of-the-art sample efficiency on MuJoCo [Todorov *et al.*, 2012] using a high update-to-data ratio and an ensemble of Q-functions. Dropout Q-learning (DroQ) [Hiraoka and others, 2022], a variant of REDQ, improved computational and memory efficiency while maintaining similar sample efficiency. Reinforcement Learning with Policy-Driven Regularization (RLPD) [Ball and others, 2023] used symmetric sampling and Layer Normalization to efficiently leverage offline data in online RL, outperforming prior methods without pre-training. Lastly, DRL trained a quadruped robot to walk on varied terrains in 20 minutes by optimizing actor-critic algorithms, achieving real-world sample efficiency without simulation [Smith *et al.*, 2023].

Despite advancements, sample efficiency in RL remains a challenge. Methods like REDQ [Chen and others, 2021], DroQ [Hiraoka and others, 2022], and RLPD [Ball and others, 2023] rely on random exploration, leading to poor early performance, especially in real-world applications. They struggle with sparse rewards, complex dynamics, generalization issues, and balancing exploration with exploitation.

Our DT-guided DRL technique addresses these issues by leveraging decision theory, using utility functions to guide learning and ensure acceptable early performance. This structured exploration mitigates random exploration and sparse reward challenges, offering a promising direction for improving RL efficiency in data-limited real-world scenarios.

3 Decision Theory-Guided DRL

In DRL, balancing exploration and exploitation is crucial for finding optimal solutions. Traditional methods like dynamic ϵ-greedy exploration, which start with random exploration, often delay convergence. Our DT-guided DRL approach integrates decision theory to provide informed action distribu-

tions from the start, address the cold start problem[1], and reduce the risk of local optima.

Unlike existing works [Chen and others, 2021; Hiraoka and others, 2022; Ball and others, 2023], DT-guided DRL provides effective initial guidance, avoiding poor performance from random exploration. It operates efficiently on small datasets, enhancing generalizability across diverse RL scenarios.

3.1 Problem Formulation Using DRL

We demonstrate our technique using the Cart Pole environment from Gymnasium [Hsiao and others, 2022; Manrique Escobar and others, 2020] and maze problems common in decision theory [Dayan and Daw, 2008], employing the Proximal Policy Optimization (PPO) algorithm [Schulman and others, 2017] for its superior performance in our experiments.

A DRL agent operates within a Markov Decision Process (MDP) as follows:

- **State** (S): In a cart pole, the state includes cart position, velocity, pole angle, and angular velocity. For maze problems, it represents the agent's x and y coordinates.
- **Action** (A): In a cart pole, actions involve pushing the cart left or right; in maze problems, actions include moving up, down, left, or right.
- **Transition Probabilities** ($T(s'|s,a)$): The probability of transitioning from state s to s' given action a.
- **Reward** ($R(s)$): In the cart pole, +1 is given for each step the pole remains balanced; in maze problems, small penalties apply for non-exit steps, and +1 is awarded for reaching the exit.
- **Policy** (π): A mapping from states to actions.

3.2 Problem Formulation Using Decision Theory

To guide the DRL agent effectively, the utility function must be tailored to the specific problem or environment.

Cart Pole Environment. In the cart pole problem, the utility function focuses on the pole's angle. Positive utility is assigned for pushing left when the angle is negative and pushing right when positive, aiming to keep the pole upright. Formally, a utility is defined as:

$$U(s_{\text{pole\_angle}}, a) = \begin{cases} -\frac{s_{\text{pole\_angle}}}{0.209}, & \text{if } a \text{ is push left} \\ \frac{s_{\text{pole\_angle}}}{0.209}, & \text{otherwise.} \end{cases} \quad (1)$$

Here, $s_{\text{pole\_angle}}$ represents the pole angle, ranging between (-0.418, 0.418) radians. Since the game ends if the angle exceeds $\pm 12^\circ$ (or ± 0.209 radians), the utility is normalized by dividing by 0.209.

Maze Environment. In the maze, the agent can move up, down, left, or right, with its position represented as $(P_{s,x}, P_{s,y})$. The utility function incorporates the distance from the agent's current position to the exit:

$$U(s,a) = \begin{cases} \frac{1}{\text{dis}(P_{s'}, P_{\text{exit}})}, & \text{if no obstacles in } a\text{'s direction} \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where s' is the anticipated new state after action a, and $\text{dis}(P_{s'}, P_{\text{exit}})$ is the distance to the exit, given by:

$$\text{dis}(P_{s'}, P_{\text{exit}}) = \sqrt{(P_{s',x} - P_{\text{exit},x})^2 + (P_{s',y} - P_{\text{exit},y})^2}. \quad (3)$$

3.3 Integrating DT with DRL for Discrete Actions

The effectiveness of DT-guided DRL lies in combining the DT agent's action probabilities with the neural network (NN) output. Then fine-tunes that network via PPO updates; this two-stage loop repeats until convergence. As shown in Figure 1, the process is as follows:

1. **Utility Function**: Create a DT agent with a problem-specific utility function (e.g., Eq. (1) for cart pole or Eq. (2) for the maze). Convert these utility values into probabilities using a softmax layer, adjusting the temperature for determinism.
2. **Deep Neural Network (DNN)**: Build a DNN that inputs the environment's current state and outputs action probabilities.
3. **Integration**: Combine the NN's outputs with the DT agent's probabilities (reverse softmax), summing them and adjusting a dynamic weight w, starting at $w = 1$ and gradually reducing to $w = 0$ with a linear decay schedule during training.
4. **Final Softmax**: Apply a softmax layer (temperature 1) to the combined outputs for a valid action probability distribution.
5. **Training**: Train the DNN using RL algorithms like PPO or DQN, updating weights based on rewards from interactions with the environment.

Utility vs. Reward Function. Unlike reward shaping, which modifies the environment's primary reward structure, our utility functions act as an auxiliary heuristic to guide early exploration. Rather than altering the original reward signals that define task objectives, the utility biases the agent's action selection during initial training, accelerating learning while preserving the problem's fundamental nature. This distinction ensures the canonical MDP formulation remains intact while effectively mitigating the cold start challenge. Although reward shaping can guide exploration by modifying reward signals, it alters the original MDP objectives, whereas our approach provides early decision-theoretic guidance without changing the inherent reward structure.

3.4 Integrating DT with DRL for Continuous Actions

Integrating decision theory (DT) with DRL in continuous action spaces poses challenges due to the infinite action set. To address this, we adapt DT-guided DRL by incorporating decision-theoretic guidance and experience replay.

Integrating DT with DRL follows this process:

[1]The cold start problem arises from limited initial data, leaving the learning agent with insufficient knowledge to make informed decisions and resulting in poor early performance.

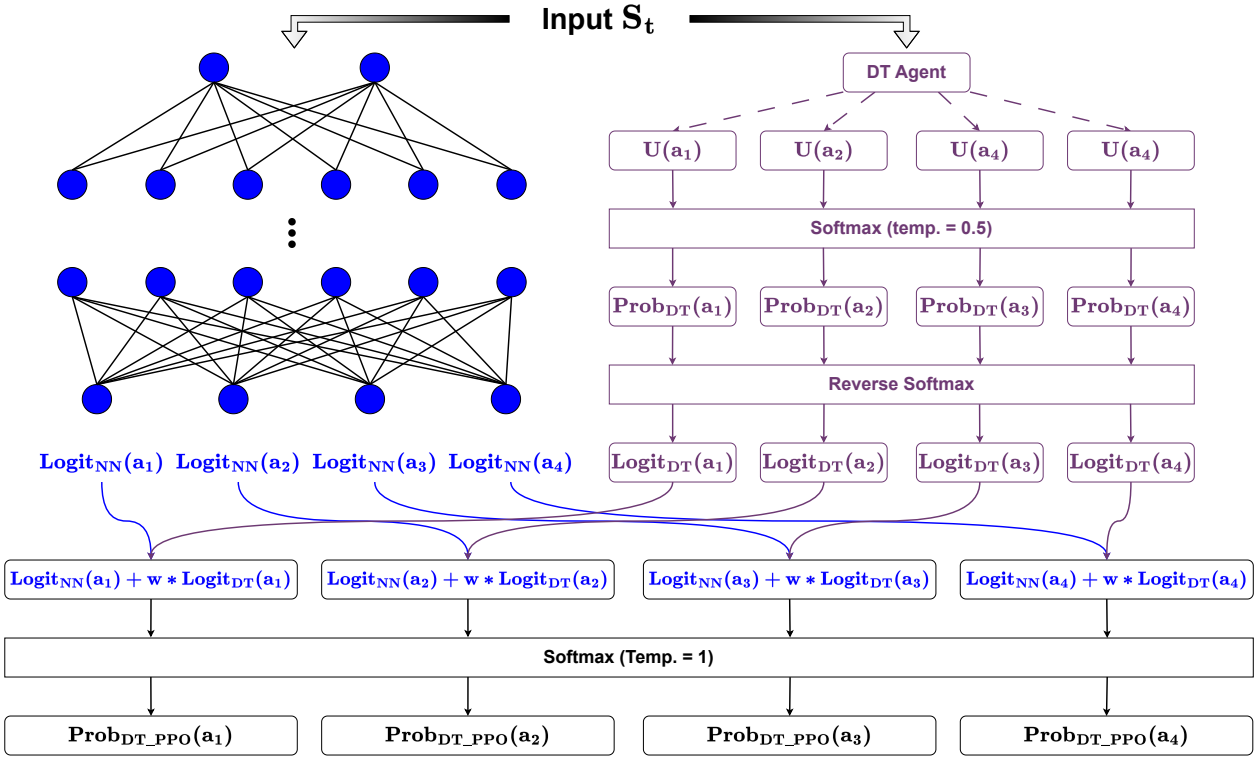


Figure 1: **Procedures for generating solutions by a DT-guided DRL agent:** $\mathcal{S}_t$ is the state at round t, and $Prob_{DT}(a)$ or $Prob_{DT\_PPO}(a)$ are the action probability.

1. **Utility Function**: Develop a DT agent with a problem-specific utility function mapping continuous observations to actions. For example, in the inverted pendulum, the utility function computes optimal torque from state variables (e.g., pole angle, cart position) to balance.
2. **Initial Interaction**: The DT agent interacts with the environment for multiple episodes, selecting actions to achieve task objectives (e.g., balancing the pendulum).
3. **Experience Collection**: Record state transitions, actions, rewards, and next states from the DT agent's interactions to capture decision-making patterns informed by the utility function.
4. **Replay Buffer Integration**: Insert the collected experiences into the DRL agent's replay buffer, enriching the initial training data with informed decisions and improving sample efficiency.
5. **DRL Model**: Build a DRL model that inputs continuous states and outputs continuous actions or action distribution parameters (e.g., mean and variance for Gaussian policies in Soft Actor-Critic or SAC).
6. **Neural Network Training**: Train DRL by sampling mini-batches from the replay buffer (with DT and DRL experiences) and updating policy and value networks.
7. **Policy Improvement**: The DRL agent refines its policy by learning from DT-guided experiences and interactions, allowing for better generalization and exploring more relevant actions.

By initializing the replay buffer with decision-theory-driven experiences, the DRL agent benefits from an informed warm start, reducing random exploration and accelerating convergence in continuous action environments.

4 Experimental Setup

4.1 Environment & Parameterization

Our study evaluates DT-guided DRL in both discrete and continuous action environments.

Discrete Action Environments

We consider two scenarios as follows.

- **Cart Pole:** The agent balances a pole on a cart by pushing it left or right. The state space includes cart position, velocity, pole angle, and angular velocity. The agent earns +1 per timestep, and the pole stays upright, with episodes ending if the pole tilts beyond $\pm 12^\circ$, the cart moves out of bounds or after 100 timesteps.
- **Maze:** The agent navigates from the top-left to the bottom-right of the maze. The state space represents the agent's x/y position, and the action space includes moving up, down, left, or right. Reaching the exit rewards +1, while each step incurs a small penalty. Episodes end upon reaching the exit or exceeding the step limit.

Continuous Action Environment

The Inverted Pendulum problem involves balancing a pole on a cart by applying continuous forces [Tarkhov and others, 2023]. The state space includes cart position, velocity,

Table 1: NEURAL NETWORK CONFIGURATION PARAMETERS

Parameter	Value (PPO)	Value (SAC)
Discount Factor (γ)	0.99	0.99
Learning Rate	0.0003	0.0003
Replay Buffer Size	2048	20000
Batch Size	64	256
PPO's Clipping Parameter	0.2	NA
SAC's Target Smoothing Coefficient (τ)	NA	0.005
SAC's Temperature Parameter (α)	NA	0.036
Network Architecture	64 x 64	256 x 256
Activation Function	Tanh	ReLU

pole angle, and angular velocity. The action space is the continuous force applied to the cart. Table 1 summarizes key configurations, including discount factor, learning rate, replay buffer size, batch size, and network architecture. These settings are based on Stable-Baselines3 defaults [Raffin and others, 2021], ensuring rigorous and reproducible results.

4.2 Comparing Schemes

Discrete Action Environments

Under this environment, an agent takes an action as follows:

- **Decision Theory (DT) Agent** [Fishburn *et al.*, 1979]: Uses DT-based utility functions for action selection.
- **Standard Proximal Policy Optimization (PPO)** [Schulman and others, 2017]: Uses the PPO algorithm without decision-theoretic guidance.
- **Transfer Learning (TL) PPO** [Genkin and McArthur, 2022]: Starts training on a 3x3 maze and transfers learning to larger mazes to evaluate TL effectiveness.
- **Sample Efficiency (SE) PPO** [Ball and others, 2023]: Enhances sample efficiency through techniques like an increased replay buffer and freshness-prioritized experience replay.
- **Imitation Learning (IL) PPO** [Gros and others, 2020]: An agent is pre-trained with expert demonstrations before continuing standard PPO training.
- **DT-guided PPO (Ours)**: Integrates DT into the action selection process during training. As outlined in Section 3, the method integrates the neural network's action probabilities with those from the utility function, employing a dynamic weight that gradually shifts from DT guidance to an adaptive policy.

Continuous Action Environment

Under this environment, an agent behaves as follows:

- **Standard Soft Actor-Critic (SAC)** Haarnoja and others [2018]: Uses the SAC algorithm without decision-theoretic guidance.
- **DT-guided SAC** (Ours): Initializes the replay buffer with experiences collected from the DT agent.

The source code is accessible at https://github.com/Wan-ZL/DT-DRL.

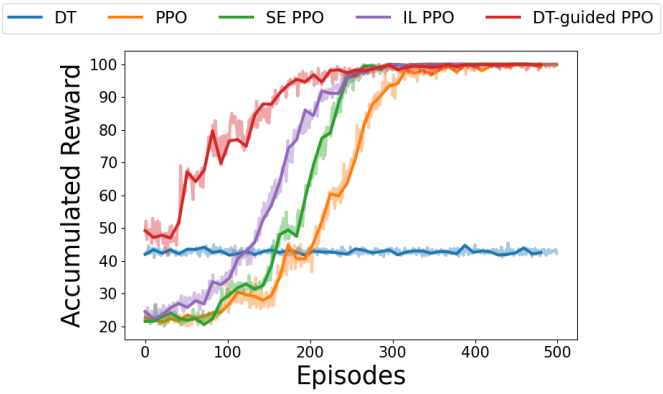


Figure 2: Accumulated rewards under DT, PPO, SE PPO, IL PPO, and DT-guided PPO over 500 training episodes in the Cart Pole problem.

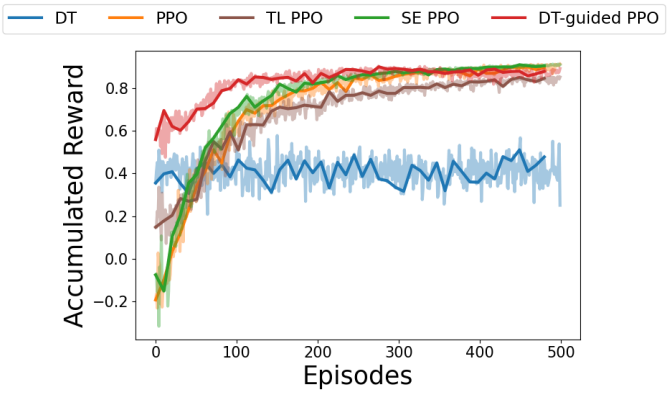


Figure 3: Accumulated rewards under DT, PPO, TL PPO, SE PPO, and DT-guided PPO over 500 training episodes in the Maze problem.

5 Simulation Results & Analysis

5.1 Cart Pole Problem

Comparative Performance Analysis

Figure 2 presents the accumulated rewards for five agents: DT, PPO, SE PPO, IL PPO, and DT-guided PPO in the Cart Pole simulation.

Our key observations and findings are as follows. First, the DT-guided PPO (red curve) outperforms other PPO agents early in training, starting with a higher reward. This advantage comes from integrating DT utility values with the neural network's output. Since the neural network begins with zero-initialized outputs and uses Tanh as the activation function, the DT utility dominates the initial decisions, making the early behavior of the DT-guided PPO similar to the DT agent (blue curve). Second, the DT agent's reward remains static, showing its limitation in adapting beyond its predefined utility function. In contrast, the DT-guided PPO steadily improves, reflecting the neural network's increasing ability to learn from the environment and refine its actions, resulting in progressively higher rewards. Finally, while all PPO agents converge to similar reward levels, the DT-guided PPO converges faster, demonstrating accelerated learning. This is due to the DT-guided PPO's informed starting point, which reduces the need for exploration and speeds up the learning

process. This early performance boost is crucial in real-world applications where each failed attempt may be costly or dangerous. In particular, our data shows that DT-guided DRL reaches near-optimal behavior significantly faster, demonstrating improved sample efficiency in the most critical initial phase of training.

Overall, Figure 2 confirms that integrating decision theory into the PPO framework mitigates random exploration early in training and accelerates convergence to optimal performance in the cart pole task.

5.2 Maze Problem

Comparative Performance Analysis

Figure 3 shows the accumulated rewards for five agents, including DT, PPO, TL PPO, SE PPO, and DT-guided PPO, in the Maze simulation.

The results reveal several key insights: (1) DT-guided PPO consistently outperforms all other agents, demonstrating the effectiveness of combining decision theory's structured, utility-based guidance with PPO's flexible neural learning. This integration enables more informed decision-making, especially in large mazes. (2) The maze environment poses a sparse reward challenge, where rewards are only received upon exit. DT-guided PPO mitigates this by using decision-theoretic guidance for early exploration, reducing inefficient random search, and enabling more focused navigation. (3) DT-guided PPO converges faster than other agents. Its initial utility-driven exploration accelerates learning by reducing the time to discover optimal strategies, allowing it to reach near-optimal performance in fewer episodes.

Sensitivity Analysis – Effect of Maze Size (m)

Figure 4 illustrates the dynamics of accumulated rewards as a function of maze size (m), ranging from 3 to 8. Each subplot delineates the trajectory of accumulated reward over training episodes, with the x-axis representing the episode count and the y-axis quantifying the reward.

Upon examination, we discern several patterns as follows. (1) The DT-guided PPO agent consistently surpasses all other agents (DT, PPO, TL PPO, and SE PPO) regarding accumulated rewards across all maze sizes. This superior performance suggests that DT-guided PPO leverages the systematic approach of decision theory and the flexible learning capabilities inherent in the PPO's neural network structure. Such an integration gives the agent a robust navigational strategy in the maze, which becomes increasingly advantageous as the maze's complexity escalates. This is crucial in larger mazes where the agent must contend with more intricate challenges. (2) As the maze size escalates, the differential in performance between the DT-guided PPO and other PPO agents becomes more pronounced, especially in later episodes. This trend could be attributed to the DT component providing a more effective heuristic in the early stages of exploration, guiding the agent through larger state spaces more efficiently. The DT's structured approach potentially reduces the agent's search space by giving zero utility to actions toward obstacles. Thus, it mitigates the challenges posed by a larger maze's complexity and helps to maintain higher performance levels than the pure PPO agent. (3) A general trend observed is the decline in the accumulated reward for all agents with increasing maze size, with the DT agent demonstrating considerable fluctuation when $m \geq 7$. This fluctuation could stem from the amplified complexity and inherent difficulties of larger mazes, where a sole reliance on decision theory might not yield efficient pathfinding consistently. Without the capacity to learn and adapt from interactions with the environment, a purely decision-theoretic model may fail. Conversely, a pure PPO approach may struggle with sparse rewards, as exits become harder to reach and positive rewards become less frequent as the maze expands.

Despite these challenges, DT-guided PPO resists the sparse reward problem, likely due to initial guidance from the decision theory component directing the agent toward more rewarding trajectories in larger mazes. The hybrid model's integration of structured decision-making with adaptive learning effectively manages maze complexities across scales.

The DT-guided PPO's consistent outperformance across maze sizes illustrates the value of combining structured decision-making with empirical adaptive neural networks, especially when dealing with problems of increasing size and complexity. The data suggests this hybrid approach could be a promising direction for developing robust solutions in complex, dynamic environments.

5.3 Inverted Pendulum Problem

Comparative Performance Analysis

Figure 5 shows the accumulated rewards for the SAC and DT-guided SAC agents in the Inverted Pendulum simulation.

The key observations are: (1) Both agents display low rewards (around 4 to 6) during the initial 150 episodes, reflecting the random exploration phase. (2) The DT-guided SAC (red curve) demonstrates an early advantage over the SAC agent (blue curve), as the decision-theoretic function provides informed action guidance, enabling more effective exploration compared to SAC's purely random approach.

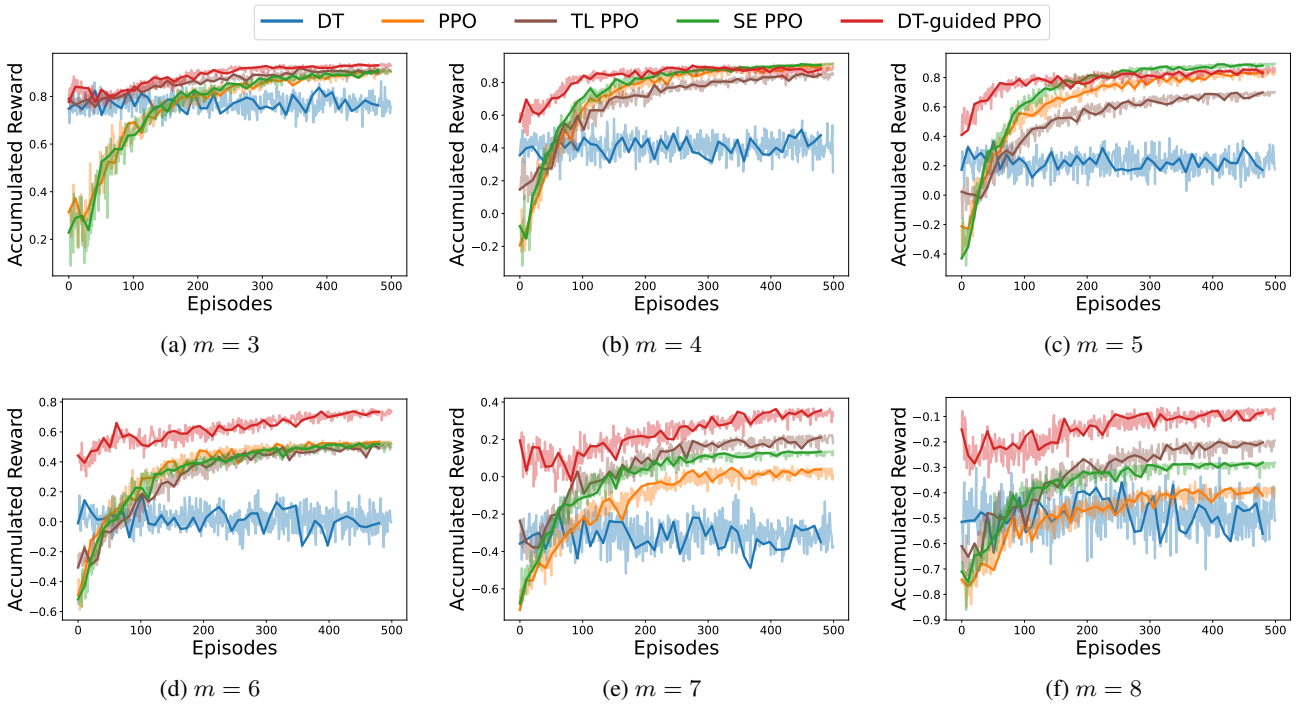
Sensitivity Analysis – Impact of Decision-Theoretic Initialization on SAC Performance

Figure 6 compares the performance of SAC* and DT-guided SAC* variants under different initialization strategies. In standard SAC, a fixed number of random steps is used initially to promote exploration. SAC* removes this random initialization, while DT-guided SAC* replaces it with 1 or 2 episodes of interactions from a decision-theoretic (DT) agent, providing structured guidance early in training.

Key observations are: (1) DT-guided SAC* consistently outperforms SAC*, achieving higher accumulated rewards. The initial DT-driven interactions guide the agent away from inefficient exploration and toward promising actions. (2) Both the 1-episode and 2-episode variants perform similarly, suggesting that the DT agent provides consistent guidance and that additional episodes yield diminishing returns.

6 Conclusion & Future Work

We proposed Decision Theory-guided Deep Reinforcement Learning (DT-guided DRL) to address the cold start problem by integrating decision-theoretic principles to improve initial performance and accelerate convergence. Experiments on



(a) $m = 3$ (b) $m = 4$ (c) $m = 5$

(d) $m = 6$ (e) $m = 7$ (f) $m = 8$

Figure 4: Effect of maze size (m): Accumulated rewards under DT, PPO, TL PPO, SE PPO, and DT-guided PPO over 500 training episodes.

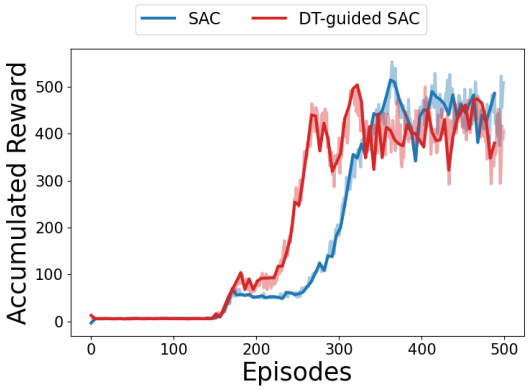


Figure 5: Accumulated reward of SAC and DT-guided SAC over 500 training episodes in the Inverted Pendulum problem.

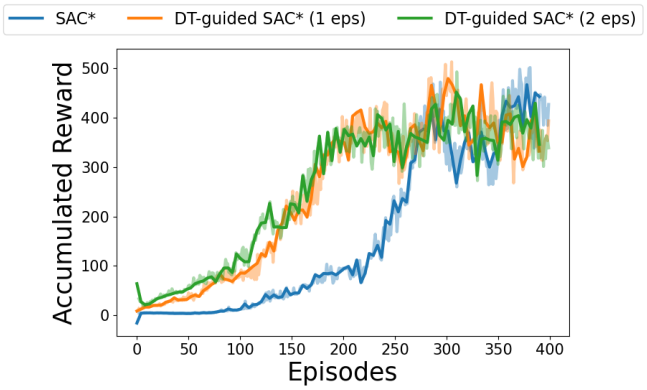


Figure 6: Accumulated reward of SAC* and DT-guided SAC* with one or two episodes of decision theory agent interactions during the training in the Inverted Pendulum problem.

cart pole, maze, and inverted pendulum tasks show that DT-guided DRL consistently outperforms conventional agents, achieving higher early-stage rewards and faster learning.

In the cart pole and maze tasks, DT-guided DRL demonstrated strong initial performance and robustness, particularly under sparse rewards and large-scale navigation. In the continuous-action inverted pendulum task, it enhanced early learning by leveraging decision-theoretic interactions.

While effective in providing structured exploration and improved early learning, the approach depends on problem-specific utility functions, which may not generalize easily. Although tested on simplified benchmarks, the results clearly demonstrate DT-guided DRL's value in mitigating the cold start problem. Future work will explore scaling to complex domains, such as multi-joint robots, legged locomotion, and vision-based tasks.

We also plan to investigate automated ways of deriving utility functions through inverse RL or meta-learning, reducing the reliance on manual heuristics. Combining DT-guided exploration with intrinsic-motivation approaches [Aubret *et al.*, 2023] may further enhance sample efficiency. Evaluating DT-guided DRL in large-scale, high-dimensional tasks and in safety-critical real-world robotics platforms forms the next crucial step in validating its broader efficacy.

In conclusion, DT-guided DRL represents a significant step towards more efficient and reliable reinforcement learning. We hope this work inspires further research at the intersection of decision theory and deep reinforcement learning.

References

Arthur Aubret, Laetitia Matignon, and Salima Hassas. An information-theoretic perspective on intrinsic motivation in reinforcement learning: A survey. *Entropy*, 25(2):327, 2023.

Philip J Ball et al. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pages 1577–1594. PMLR, 2023.

Xinyue Chen et al. Randomized ensembled double q-learning: Learning fast without a model. *International Conference on Learning Representations*, 2021.

Peter Dayan and Nathaniel D Daw. Decision theory, reinforcement learning, and the brain. *Cognitive, Affective, & Behavioral Neuroscience*, 8(4):429–453, 2008.

Floris den Hengst et al. Reinforcement learning with option machines. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 2909–2915. International Joint Conferences on Artificial Intelligence Organization, 2022.

Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *ICML Workshop on Real-Life Reinforcement Learning*, 2019.

Peter C Fishburn, Peter C Fishburn, et al. *Utility theory for decision making*. Krieger NY, 1979.

Mikhail Genkin and JJ McArthur. Using reinforcement learning with transfer learning to overcome smart building cold start. In *2022 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 713–718. IEEE, 2022.

Timo P Gros et al. Tracking the race between deep reinforcement learning and imitation learning. In *Quantitative Evaluation of Systems: 17th International Conference, QEST 2020, Vienna, Austria, August 31–September 3, 2020, Proceedings 17*, pages 11–17. Springer, 2020.

Tuomas Haarnoja et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. Pmlr, 2018.

Takuya Hiraoka et al. Dropout q-functions for doubly efficient reinforcement learning. *International Conference on Learning Representations*, 2022.

Jen-Yueh Hsiao et al. Unentangled quantum reinforcement learning agents in the OpenAI gym. *arXiv preprint arXiv:2203.14348*, 2022.

Zichuan Lin, Junyou Li, Jianing Shi, Deheng Ye, Qiang Fu, and Wei Yang. Juewu-mc: Playing minecraft with sample-efficient hierarchical reinforcement learning. *arXiv preprint arXiv:2112.04907*, 2021.

Jue Ma et al. Fresher experience plays a more important role in prioritized experience replay. *Applied Sciences*, 12(23):12489, 2022.

Camilo Andres Manrique Escobar et al. A parametric study of a deep reinforcement learning control system applied to the swing-up problem of the cart-pole. *Applied Sciences*, 10(24):9013, 2020.

D Warner North. A tutorial introduction to decision theory. *IEEE Transactions on Systems Science and Cybernetics*, 4(3):200–210, 1968.

Roxana Rădulescu et al. Multi-objective multi-agent decision making: a utility-based analysis and survey. *Autonomous Agents and Multi-Agent Systems*, 34(1):10, 2020.

Antonin Raffin et al. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

John Schulman et al. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Andrew Silva and Matthew Gombolay. Encoding human domain knowledge to warm start reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5042–5050, 2021.

Laura Smith, Ilya Kostrikov, and Sergey Levine. Demonstrating a walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning. *Robotics: Science and Systems (RSS) Demo*, 2(3):4, 2023.

DA Tarkhov et al. Optimal control selection for stabilizing the inverted pendulum problem using neural network method. *Optical Memory and Neural Networks*, 32(Suppl 2):S214–S225, 2023.

Chen Tessler et al. A deep hierarchical approach to lifelong learning in minecraft. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

Hanchen Wang, Ziba Arjmandzadeh, Yiming Ye, Jiangfeng Zhang, and Bin Xu. Flexnet: A warm start method for deep reinforcement learning in hybrid electric vehicle energy management applications. *Energy*, 288:129773, 2024.

Benjamin Wexler et al. Analyzing and overcoming degradation in warm-start reinforcement learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4048–4055. IEEE, 2022.

Bin Xu et al. Learning time reduction using warm-start methods for a reinforcement learning-based supervisory control in hybrid electric vehicle applications. *IEEE Transactions on Transportation Electrification*, 7(2):626–635, 2020.

Jiayin Zhang et al. Cold-start aware cloud-native service function chain caching in resource-constrained edge: A reinforcement learning approach. *Computer Communications*, 195:334–345, 2022.

Zhuangdi Zhu et al. Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Qijie Zou, Kang Xiong, and Yingli Hou. An end-to-end learning of driving strategies based on ddpg and imitation learning. In *2020 Chinese Control and Decision Conference (CCDC)*, pages 3190–3195. IEEE, 2020.