Mimicking Human Intuition: Cognitive Belief-Driven Reinforcement Learning

Xingrui Gu [1] **Guanren Qiao** [2] **Chuyi Jiang** [3]

Abstract

Traditional reinforcement learning (RL) methods mainly rely on trial-and-error exploration, often lacking mechanisms to guide agents toward more informative decision-making and struggling to leverage past experiences, resulting in low sample efficiency. To overcome this issue, we propose an innovative framework inspired by cognitive principles: Cognitive Belief-Driven Reinforcement Learning (CBD-RL). By incorporating cognitive heuristics, CBD-RL transforms conventional trial-and-error learning into a more structured and guided learning paradigm, simulating the human reasoning process. This framework's core is a belief system that optimizes action probabilities by integrating feedback with prior experience, thus enhancing decision making under uncertainty. It also organizes state-action pairs into meaningful categories, promoting generalization and improving sample efficiency. The concrete implementations of this framework, CBDQ, CBDPPO, and CBDSAC, demonstrate superior performance in discrete and continuous action spaces in diverse environments such as Atari and MuJoCo. By bridging cognitive science and reinforcement learning, this research opens a new avenue for developing RL systems that are more interpretable, efficient, and cognitively inspired.

1. Introduction

Reinforcement learning (RL) has achieved strong performance in domains with well-defined structure, such as Atari (Mnih et al., 2015) and continuous control (Lillicrap, 2015; Qiao et al., 2024a), yet remains markedly less sample-efficient than human learning. Humans can generalize from limited experience by abstracting structural regularities and transferring them across contexts, while RL agents typically require millions of interactions to attain basic competence. This gap reflects the absence of cognitive inductive biases crucial to efficient generalization—namely, the integration of uncertain evidence with prior expectations (Griffiths & Tenenbaum, 2005; Peterson & Beach, 1967), and the formation of reusable abstractions that support compositional reasoning (Tenenbaum et al., 2011; Lake et al., 2015).

Figure 1. Inspired by the way pets intuitively choose to walk, stand, or jump in different settings (path, ocean, river)

Cognitive science highlights belief updating and conceptual abstraction as core mechanisms enabling efficient human learning. Probabilistic belief revision, formalized via Bayesian inference, allows humans to construct generative models for inference under uncertainty (Gigerenzer et al., 1991; Tenenbaum et al., 2006). In parallel, conceptual abstraction compresses experience into structured forms—such as taxonomies and causal schemas—that support generalization and reuse (Tenenbaum & Griffiths, 2001; Gopnik & Wellman, 2012). Together, these mechanisms induce compact inductive biases that facilitate data-efficient decision-making, in contrast to model-free RL, which lacks structured representations of environment dynamics (Lake et al., 2017; Botvinick & Weinstein, 2014).

While recent RL research has begun to emulate facets of human flexibility, its progress remains fragmented. Bayesian and model-based approaches propagate uncertainty (Ghavamzadeh et al., 2015), but fail to integrate concepts across contexts. Multi-policy fusion reuses pretrained skills (Chiu et al., 2023; Peng et al., 2021), although static libraries hinder online abstraction. Each line of work addresses a single aspect—temporal hierarchy, uncertainty reasoning, or skill reuse—yet none unifies probabilistic inference with dynamic concept formation throughout learning. This fragmentation reflects a deeper epistemological shift. Sutton and Silver advocate for an "Era of Experience," where intelligence arises not from data-driven prediction,

[1]Centre for Artificial intelligence, UCL [2]The Chinese University of Hong Kong, Shenzhen [3]Columbia University. Correspondence to: Xingrui Gu <xingrui_gu@berkeley.edu>.

Proceedings of the 41^{st} ICML Workshop on Models of Human Feedback for AI Alignment, Vancouver, Canada. PMLR 267, 2025.

but from the agent's active organization of its own interaction history(Silver & Sutton, 2025). Experience, in this view, is the substrate of knowledge—concepts and strategies emerge through continual abstraction from behavior.

Based on this proactive insights, we propose Cognitive Belief-Driven Reinforcement Learning (CBD-RL), a unified framework that integrates probabilistic inference and conceptual abstraction into the RL process. At its core, a **Smoothed Bellman Operator** replaces pointwise maximization with belief-weighted expectations, enabling uncertainty-aware value updates. A **Conceptual Category Formation(CCF)** module clusters state–action pairs into latent abstractions, supporting modular reuse and compositional generalization. We implement CBD-RL in Q-learning(Watkins et al., 1992), SAC (Haarnoja et al., 2018), and PPO (Schulman et al., 2017), and show consistent improvements in sample efficiency, final return, and generalization across both discrete and continuous domains. We hope we can connect cognitive science in an innovative and unique way to enhance the effectiveness of RL algorithms.

2. Related Works

2.1. Value-based and Policy-based Reinforcement Learning Methods

Model-free reinforcement learning (RL), including value-based methods like XQL (Garg et al., 2023) and DoubleGum (Hui et al., 2023), and policy-based approaches such as PPO (Schulman et al., 2017), has seen improved stability and exploration through stochastic regularization and adaptive experience replay (Hassani et al., 2025). However, these methods often fail to retain and reuse knowledge effectively, especially in continual learning settings (Dohare et al., 2024). Replay strategies—hierarchical (Yin & Pan, 2017), sequence-based (Li et al., 2024), and selective (De Bruin et al., 2018)—focus on raw data reuse without abstraction, limiting conceptual generalization and structure-aware transfer (Jeen et al., 2023).

2.2. Cognitive Science Perspectives on Efficient Learning and Decision-Making

Humans exhibit exceptional cognitive efficiency, generalizing from limited experience through Bayesian inference that integrates prior knowledge with new evidence under uncertainty (Tenenbaum & Griffiths, 2001; Griffiths & Tenenbaum, 2005; Tenenbaum et al., 2006). This process supports conceptual abstraction—extracting high-level structure from sparse data—and enables causal reasoning and cross-domain transfer (Tenenbaum et al., 2011; Kemp & Tenenbaum, 2008). Recent studies formalize how human learners reorganize internal knowledge via probabilistic reasoning (Lake et al., 2015; 2017), motivating the integration of such cognitive principles into machine learning to enhance scalability, adaptability, and sample efficiency (Ma et al., 2022). Other work shows that uncovering latent causal structures—even in domains like joint behavior—can enhance model interpretability and abstraction (Gu et al., 2024a;b).

2.3. Trade-Offs in Abstraction and Policy Integration Paradigms

Reinforcement-learning agents adopt diverse mechanisms for abstraction and policy reuse, each with characteristic trade-offs. Hierarchical methods, such as Option-Critic (Bacon et al., 2017) and FeUdal Networks (Vezhnevets et al., 2017), acquire temporally extended skills but struggle to adjust option granularity and time-scales on the fly. Knowledge-grounded approaches (Jiang & Luo, 2019; Kimura et al., 2021) embed symbolic or relational priors—often via graph or logic modules—improving sample efficiency at the cost of external knowledge engineering. Curriculum-style transfer and progressive networks (Rusu et al., 2016; Narvekar et al., 2020) accelerate learning by reusing feature hierarchies, yet require carefully staged task sequences and exhibit limited structural flexibility. Multi-policy fusion techniques (Qiao et al., 2024b; Peng et al., 2021; Chiu et al., 2023) blend diverse pre-trained controllers through adversarial or attention mechanisms, typically assuming a fixed policy library that hampers online abstraction. These paradigms expose a common tension between structural rigidity, prior dependence, and scalability—motivating alternatives that infer latent concepts and reason about uncertainty directly from experience.

3. Problem Formulation

Markov Decision Processes (MDP) To solve an RL problem, the agent optimizes the control policy under an MDP $\mathcal{M}$, which can be defined by a tuple $(\mathcal{S}, \mathcal{A}, T, r, \mu_0, \gamma, T)$ where: 1) $\mathcal{S}$ and $\mathcal{A}$ denote the space of states and actions. 2) $\mathcal{T}(s_{t+1}|s_t, a_t)$ and $r(s_t, a_t)$ define the transition probability and reward function. 3) μ_0 defines the initial state distribution. 4) $\gamma \in (0, 1)$ is the discount factor and T defines the planning horizon. The goal of the RL policy π is to maximize expected discounted rewards:

$$\arg\max_{\pi} \mathbb{E}_{\pi,\mathcal{T},\mu_0}\Big[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)\Big] \tag{1}$$

We define the action value function given a policy π:

$$Q(s, a) = \mathbb{E}_{\pi,\mathcal{T},\mu_0}\big[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a\big] \tag{2}$$

and value function is:

$$V(s) = \mathbb{E}_{\pi,\mathcal{T},\mu_0}\big[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \mid s_0 = s\big] \tag{3}$$

4. Methodology

In this section, we introduce the Cognitive Belief-Driven Reinforcement Learning (CBD-RL) framework, which enhances the utilization and decision making of experience by integrating cognitive principles. It incorporates the Smoothed Bellman Operator for probabilistic reasoning (See in Section 4.1) and organizes experiences into conceptual categories through conceptualized experience abstraction, thereby improving learning efficiency and adaptability in complex environments.

4.1. Smoothed Bellman Operator

Human reward processing operates under probabilistic principles, with dopamine neurons encoding both the magnitude and uncertainty of rewards through phasic responses (Dayan & Daw, 2008; Schultz et al., 1997). Rather than responding to fixed values, humans evaluate rewards based on perceived reliability, enabling adaptive decision-making under uncertainty (Schultz, 2015; O'Doherty et al., 2004). These cognitive mechanisms align with behavioral economic models of subjective utility and provide a biological foundation for incorporating probabilistic reasoning into reinforcement learning frameworks (Sutton & Barto, 2018). To align reinforcement learning with human-like probabilistic decision-making, we revisit the classical Bellman operator, which updates value functions deterministically, $\mathcal{T}(s,a) = r_t + \gamma \max_a Q_t(s_{t+1}, a)$. The use of $\max$ implies deterministic action selection, neglecting uncertainty and preference variability. In contrast, human decisions are often modeled by Subjective Expected Utility Theory (SEUT) (Mongin & Philippe, 1998; Johnson-Laird et al., 2015), which integrates both reward expectations and belief-driven preferences. Inspired by SEUT, we introduce a probabilistic evaluation mechanism through the distribution $q_t(a \mid s_{t+1})$, representing the likelihood of selecting action a given state s_{t+1}. This formulation captures the inherent uncertainty in action selection and is formalized in the following proposition.

Proposition 4.1. *Consider a decision-making scenario within an MDP. Let $q_t(a \mid s_{t+1})$ represent the probability distribution over actions at the next state s_{t+1}, derived from the reward signal. Then, the expected utility $U_t(s_t, x)$ at time t is given by:*

$$U_t(s_t, x) = \sum_{a \in A} q_t(a \mid s_{t+1}) u_t(s_t, x), \tag{4}$$

where $u_t(s_t, x)$ denotes the utility of outcome x in state s_t.

The above proposition formalizes expected utility in reinforcement learning by combining probabilistic evaluations and utility functions, providing a framework for incorporating uncertainty and preference variability into decision-making. To address the limitations of the deterministic $\max$ operation, we propose the Smoothed Bellman Operator, which replaces $\max$ with a probability distribution $q_t(a \mid s_{t+1})$, enabling probability-weighted action selection. This smoothing strategy is further detailed in the Appendix B, including formulations such as softmax smoothing to enhance stability and performance in reinforcement learning. Theoretical properties of the Smoothed Bellman Operator, including a Jensen-type inequality and convergence conditions, are provided in Appendix B.

Lemma 4.2 (Convergence of Smoothed Bellman Operator)**.** *Let $\{Q_t\}$ be the sequence generated by iteratively applying $\mathcal{T}_{Smoothed}$. Under the condition:*

$$\lim_{t\to\infty} \max_a q_t(a \mid s_{t+1}) = 1, \tag{5}$$

for the optimal action, Q_t converges to the optimal Q^* as $t \to \infty$. See Appendix D for a detailed proof.

4.2. Cognitive Belief-Driven Reinforcement Learning (CBD-RL) Framework

Algorithm 1 Cognitive Belief-Driven RL: Experience Integration via Conceptual Structures

Require:
Conceptual Category Set $\{C_n\}_{n=1}^N$
Adaptive Integration Coefficient $\beta_t \in [0,1]$
for Each Category C_k **do**
  Initialize Belief Model $P_k(a|s)$;
end for
for Each Training Step t **do**
  Observe current state s_t, determine category C_k;
  Collect reward signal r_t, environment transition s_{t+1};
  Integrate action and prior belief into updated preference:

$$b_t(a|s_t) = (1-\beta_t)\cdot \mathcal{Z}_t(a|s_{t+1}) + \beta_t \cdot P_k(a|s_t)$$

  Select action $a_t \sim b_t(\cdot|s_t)$, interact with environment;
  Update P_k based on observed outcome and adaptation rule.
end for

In the previous section, we introduce the Smoothed Bellman Operator to incorporate probabilistic evaluation of rewards, drawing inspiration from human cognitive processing. Cognitive science shows that humans efficiently abstract concepts from limited experience and apply them to

novel contexts by recognizing structural similarities and shared features (Tenenbaum et al., 2011; Lake et al., 2015; Kemp & Tenenbaum, 2008). Conceptual organization supports memory compression (Gershman et al., 2015), rapid generalization (Tenenbaum et al., 2006), and cross-scenario transfer (Botvinick et al., 2019). Building on these insights, we propose the **Cognitive Belief-Driven Reinforcement Learning (CBD-RL)** framework, which clusters similar state–action pairs into conceptual categories represented by probability distributions to capture uncertainty and integrate reward signals. This abstraction mechanism enhances experience efficiency and generalization in reinforcement learning.

CBD-RL centers on **Conceptual Category Formation (CCF)** (Definition 4.4) (Rosch, 1978; Murphy, 2004), which clusters state–action experiences into semantically coherent groups. Each category is parameterized by a **Belief** distribution (Definition 6) (Premack & Woodruff, 1978; Dennett, 1988), representing probabilistic action preferences within its context. Cognitive science indicates that such abstraction and uncertainty-aware reasoning jointly support generalization from limited data and robust decision-making (Tenenbaum & Griffiths, 2001; Griffiths et al., 2010; Lake et al., 2017; Rogers & McClelland, 2004).

Definition 4.3 (Belief)**.** A **belief** represents an agent's internal representation and understanding of the decision-making context, particularly over the action space. Unlike traditional Bayesian cognitive models, which define belief as a posterior probability distribution, we propose a more direct representation of belief grounded in probability theory (Gigerenzer et al., 1991; Griffiths et al., 2010).

For each state s, we define a belief as a probability measure $\mathcal{P}_k$ over the action space A, satisfying the following conditions:

$$\mathcal{P}_k(a) = \begin{cases} \sum_{a \in A} \mathcal{P}_k(a) = 1, & \text{if } A \text{ is discrete} \\ \int_A \mathcal{P}_k(a)\, da = 1, & \text{if } A \text{ is continuous} \end{cases} \tag{6}$$

Definition 4.4 (Conceptual Category Formation)**.** A **Conceptual Category Formation (CCF)** is a partition of the state space S into meaningful subsets, enabling an agent to generalize from similar experiences. Facilitates efficient learning by grouping states with shared characteristics, promoting rapid adaptation and knowledge transfer. Formally, a CCF of S is a finite partition $C = \{C_1, \ldots, C_N\}$ that satisfies the following conditions:

1. **Completeness and Exclusivity:** The partition covers the entire state space without overlap, i.e., $S = \bigcup_{n=1}^{N} C_n$, with $C_i \cap C_j = \emptyset$ for $i \neq j$.
2. **Semantic Coherence:** Each category C_n is characterized by a representative state $c_n^i \in C_n$, such that for all $c_n^j \in C_n$ $i \neq j$, $d(c_n^j, c_n^i) \leq \epsilon_n$, where ϵ_n is the semantic radius of the category and d is a distance function.
3. **Conceptual Consistency:** States within the same category share consistent characteristics such that for any $c_n^1, c_n^2 \in C_n$, $\|f(c_n^1) - f(c_n^2)\| \leq \delta$, where f is a feature mapping function and $\delta > 0$ is a predefined threshold for similarity of features.

The **Belief** and **CCF** mechanisms work together to help an agent process environmental information and make informed decisions based on prior knowledge (Tenenbaum et al., 2011). The **Belief** updates beliefs through probabilistic reasoning, guiding decision-making under uncertainty. In contrast, **CCF** enables agents to identify and abstract conceptual features by categorizing similar experiences. **CBD-RL** combines these two mechanisms to make informed decisions, a unified approach (see Algorithm 1 and Theorem 4.5).

Theorem 4.5 (Cognitive Belief-Driven Reinforcement Learning)**.** *Consider an MDP, a Conceptual Category Formation(CCF) partition $C = \{C_n\}_{n=1}^{N}$ of the state space S, and a history of experiences $H_t = \{(s_i, a_i, r_i, s_{i+1})\}_{i=1}^{t}$. For any $s \in S$, let $\mathcal{Z}_t(\cdot \mid s_{t+1})$ and $\mathcal{P}_k(\cdot \mid s_t)$ denote probabilities based on actions and category information, respectively, with a time-dependent adaptation parameter $\beta_t \in [0, 1]$.*

The belief-preference distribution $b_t : S \times A \to [0, 1]$ is:

$$b_t(\cdot \mid s_{t+1}) = (1 - \beta_t)\mathcal{Z}_t(\cdot \mid s_{t+1}) + \beta_t \mathcal{P}_k(\cdot \mid s_t), \tag{7}$$

where β_t is monotonic with $\lim_{t\to\infty} \beta_t = \beta^* \in [0, 1]$.

4.3. CBD-RL in Discrete & Continuous Action Spaces

In the previous section, we provided a detailed explanation of the CBD-RL framework, including a brief overview, the process of Conceptual Category Formation, and the pseudocode outlining the implementation of this framework (See in Algorithm 1). However, given the diversity of experimental environments across various reinforcement learning (RL) algorithms, it is not feasible to define a single, fixed pattern for constructing our framework. Therefore, based on the nature of the action space, we have designed two distinct implementations of CBD-RL to accommodate both discrete and continuous action spaces.

Discrete action spaces consist of a finite and countable action set A, enabling explicit estimation of state-action values and precise computation of action probabilities. Upon executing an action $a_t \in A$, the corresponding state-action pair is assigned to its conceptual category C_k, and the action probability $\mathcal{P}_k(a \mid s_t)$ is subsequently updated.

Proposition 4.6 (CBD-RL in Discrete Action Spaces)**.** *For discrete action spaces where A is finite and countable, the action selection probability for a given conceptual category C_k is defined as:*

$$\mathcal{P}_k(a \mid s_t) = \frac{f(a \mid s \in C_k)}{\sum_{\tilde{a} \in A} f(\tilde{a} \mid s \in C_k)}, \tag{8}$$

where $f(a \mid s \in C_k)$ represents the frequency with which action a has been selected in states belonging to category C_k. This probability distribution encapsulates the historical preference for actions within the category C_k.

Continuous action spaces, which optimize policy networks to maximize expected returns, present several challenges. First, decisions are based solely on the current state, limiting the use of insights from similar experiences (Haarnoja et al., 2018; Botvinick et al., 2019). Second, the value function is primarily learned via temporal-difference errors, which overlook the underlying structure of the state space. Lastly, the framework lacks mechanisms for accumulating and transferring experiential knowledge, leading to inefficiencies in long-term learning.

To overcome these limitations, we adopt probabilistic policy learning within the maximum entropy framework and extend it by incorporating category-based beliefs. Specifically, we assume the use of Gaussian distributions as the underlying probabilistic policies.

Proposition 4.7 (CBD-RL in Continuous Action Spaces)**.** *Consider a continuous action space $A \subseteq \mathbb{R}^n$. For each conceptual category C_k, we maintain a Gaussian belief distribution over actions:*

$$\mathcal{P}_k(a \mid s_t) = \mathcal{N}(\mu_k(s_t), \sigma_k^2(s_t)), \tag{9}$$

where $\mu_k(s_t)$ and $\sigma_k^2(s_t)$ are derived from historical transitions within category C_k through Bayesian posterior updates. Specifically:

$$\mu_{posterior} = \frac{\sigma_{prior}^2 \mu_{obs} + \sigma_{obs}^2 \mu_{prior}}{\sigma_{prior}^2 + \sigma_{obs}^2}, \tag{10}$$

$$\sigma_{posterior}^2 = \frac{\sigma_{prior}^2 \sigma_{obs}^2}{\sigma_{prior}^2 + \sigma_{obs}^2}. \tag{11}$$

Here, μ_{prior} and σ_{prior}^2 represent the mean and variance of the prior distribution, while μ_{obs} and σ_{obs}^2 represent the mean and variance of the observed data in the current round.

5. Algorithm Implementation

Algorithmic instantiations of the Cognitive Belief-Driven Reinforcement Learning (CBD-RL) framework include Cognitive Belief-Driven Q-learning (CBDQ) for value-based, and its policy-based counterparts—Cognitive Belief-Driven Soft Actor-Critic (CBDSAC) and Cognitive Belief-Driven Proximal Policy Optimization (CBDPPO).

5.1. Cognitive Belief-Driven Q-learning (CBDQ)

CBDQ operationalizes the CBD-RL framework within a discrete action domain by integrating short-term reward learning with long-term conceptual preferences. Rather than relying on greedy action selection, CBDQ applies a smoothed Bellman operator where the next-state value estimate is computed using a belief-preference distribution. This distribution merges reward-informed action likelihoods with concept-level historical priors derived from Conceptual Category Formation (CCF).

For a given state s_t, its conceptual category C_k is identified via distance-based clustering. A category-specific belief distribution $\mathcal{P}_k(a \mid s_t)$ is constructed using normalized action frequencies. This prior is fused with $q_t(a \mid s_{t+1})$ to form:

$$b_t(a \mid s_{t+1}) = (1-\beta_t) \cdot q_t(a \mid s_{t+1}) + \beta_t \cdot \mathcal{P}_k(a \mid s_t), \tag{12}$$

where $\beta_t \in [0, 1]$ is an adaptive parameter controlling the influence of conceptual memory.

The Q-function update rule becomes:

$$\begin{aligned} Q_{t+1}(s_t, a_t) = \; & Q_t(s_t, a_t) + \alpha_t(s_t, a_t) \Big[r(s_t, a_t) \\ & + \gamma \sum_{a \in A} b_t(a \mid s_{t+1}) Q_t(s_{t+1}, a) - Q_t(s_t, a_t) \Big] \end{aligned} \tag{13}$$

where the expectation over b_t acts as a smoothed substitute for the max operator in classical Q-learning, enabling uncertainty-aware value propagation.

Practically, CBDQ uses an ϵ-greedy exploration strategy with actions sampled from $b_t(a \mid s_{t+1})$. The belief model is implemented as an online frequency table per category, updated incrementally with each transition. Experience replay is used to stabilize training, and conceptual categories are periodically re-evaluated to ensure semantic coherence.

This mechanism allows CBDQ to leverage both immediate task feedback and accumulated structural knowledge, promoting faster convergence and improved generalization. The full procedure is described in Appendix A.1, and convergence is discussed in Appendix C under the Smoothed Bellman formulation.

5.2. Cognitive Belief-Driven Soft Actor-Critic (CBDSAC)

To support concept-informed decision-making in continuous action spaces, we integrate the Cognitive Belief-Driven Reinforcement Learning (CBD-RL) framework into Soft Actor-Critic (SAC), forming the Cognitive Belief-Driven Soft Actor-Critic (CBDSAC). While SAC optimizes a stochastic Gaussian policy through entropy-regularized objectives, it relies solely on immediate feedback, lacking mechanisms

for retaining and transferring structured knowledge across semantically similar but temporally distant contexts.

CBDSAC augments SAC by incorporating belief distributions derived from Conceptual Category Formation (CCF), which cluster states into semantically coherent groups and provide structured priors over actions. Instead of relying solely on the actor network's output, CBDSAC fuses its learned policy distribution with a corresponding conceptual belief, previously constructed and updated using Bayesian statistics (see Chapter 4). This fusion yields a belief-preference distribution that modulates both action sampling and policy updates. The actor network outputs a Gaussian policy:

$$\mathcal{Z}_t(\cdot \mid s_{t+1}) = \mathcal{N}(\mu_{\pi_\theta}(s_{t+1}), \sigma^2_{\pi_\theta}(s_{t+1})), \tag{14}$$

which captures short-term, reward-driven action preferences. CBDSAC fuses this distribution with the conceptual belief to construct a belief-preference distribution:

$$\mu_{\text{blend}}(s_{t+1}) = (1-\beta_t)\mu_{\pi_\theta}(s_{t+1}) + \beta_t \mu_{\text{post}}(s_t), \tag{15}$$
$$\sigma^2_{\text{blend}}(s_{t+1}) = (1-\beta_t)\sigma^2_{\pi_\theta}(s_{t+1}) + \beta_t \sigma^2_{\text{post}}(s_t), \tag{16}$$

where $\beta_t \in [0,1]$ modulates the influence of structured belief versus reactive policy.

This yields the final action distribution:

$$b_t(\cdot \mid s_{t+1}) = \mathcal{N}(\mu_{\text{blend}}(s_{t+1}), \sigma^2_{\text{blend}}(s_{t+1})), \tag{17}$$

which governs both action sampling and gradient-based updates.

Policy optimization proceeds via entropy-regularized soft policy iteration:

$$\mathbb{E}_{s\sim\mathcal{D}, a\sim b_t}\left[Q_\phi(s,a) - \alpha \log b_t(a \mid s_{t+1})\right], \tag{18}$$

with:

$$\log b_t(a \mid s_{t+1}) = -\frac{1}{2}\left(\frac{(a-\mu_{\text{blend}})^2}{\sigma^2_{\text{blend}}} + 2\log\sigma_{\text{blend}} + c\right). \tag{19}$$

where c is a fixed constant.

By integrating policy learning with semantically grounded beliefs, CBDSAC enables agents to generalize across conceptually coherent behaviors. This fusion facilitates better sample reuse, long-term coherence, and more human-like decision-making. Full implementation details and pseudocode are provided in Appendix A.2.

5.3. Cognitive Belief-Driven Proximal Policy Optimization (CBDPPO)

To incorporate concept-level priors into on-policy learning, the Cognitive Belief-Driven Reinforcement Learning (CBD-RL) framework is integrated with the clipped surrogate formulation of Proximal Policy Optimization (PPO), yielding the Cognitive Belief-Driven PPO (CBDPPO) algorithm. While PPO maximizes an advantage-weighted likelihood ratio under a trust region constraint (Qiao et al., 2023), the policy update remains driven solely by immediate feedback, limiting its ability to leverage structural regularities observed across semantically related trajectories.

CBDPPO addresses this limitation by blending the current policy $\pi_\theta(a \mid s)$ with the conceptual belief $P_k(a \mid s)$ associated with the category C_k that contains state s. The resulting belief-preference policy is defined as:

$$b_t(a \mid s) = (1-\beta_t)\pi_\theta(a \mid s) + \beta_t P_k(a \mid s), \tag{20}$$

where the scheduling parameter $\beta_t \in [0,1]$ controls the influence of concept priors and increases gradually throughout training. The clipped surrogate objective of CBDPPO is:

$$\mathcal{L}_{\text{CBDPPO}} = \mathbb{E}_{(s,a)\sim\pi_{\theta_{old}}}\Big[\min\Big(\frac{b_t(a \mid s)}{\pi_{\theta_{old}}(a \mid s)}A_t, \text{clip}\left(\frac{b_t(a \mid s)}{\pi_{\theta_{old}}(a \mid s)}, 1-\epsilon, 1+\epsilon\right)A_t\Big)\Big], \tag{21}$$

where A_t is the advantage estimate and ϵ controls the trust region width. The critic and entropy terms follow the original PPO formulation; gradients are propagated through b_t, allowing concept priors to steer policy updates while the clip operator guarantees trust-region stability. The critic and entropy terms follow the original PPO formulation; gradients are propagated through b_t, allowing concept priors to steer policy updates while the clip operator guarantees trust-region stability. Full implementation details and pseudocode are provided in Appendix A.3.

6. Experiment

Running Setting Evaluation is based on *Feasible Cumulative Rewards*, where higher values indicate better performance, averaged over three seeds (123, 321, 666). Conceptual clustering within CCF is simulated via clustering algorithms that group similar state-action pairs into latent categories (Appendix). All methods use identical hyperparameters and are implemented on the XuanCe benchmark suite (Liu et al., 2023), with full configurations in Appendix E.4.

Comparison Methods For discrete action spaces, the comparison includes CBDQ and CBDPPO and the following baselines: (1) **DQN** (Mnih et al., 2013) with neural networks for Q-value approximation; (2) **DDQN** with decoupled action selection and evaluation; (3) **DuelDQN** with separate state value and action advantage estimation; (4) **PPO** with clipped objective for stable policy optimization.

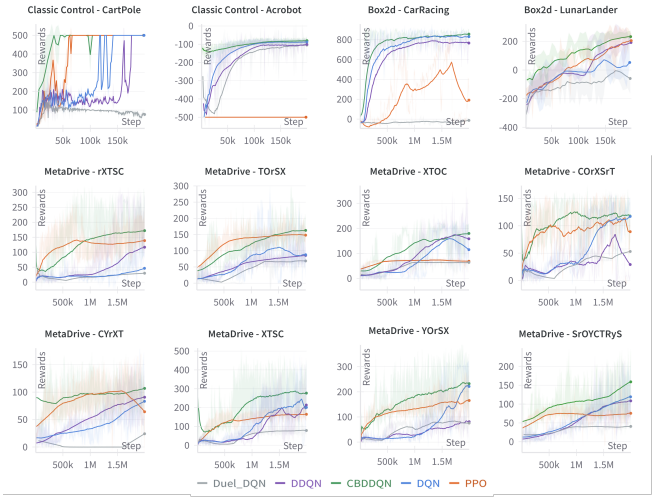

Figure 2. Learning curves of CBDDQN, DQN, DDQN, PPO and Duel_DQN. First row for Box2D (CarRacing and LunarLander) Classic Control(CartPole and Acrbot). Second and Third row for Metadrive with 8 different maps.

For continuous action spaces, CBDSAC is compared with: (1) **A2C** (Mnih, 2016) with synchronized advantage estimation; (2) **PPO** (Schulman et al., 2017) with clipped surrogate objective; (3) **SAC** (Haarnoja et al., 2018) with entropy-regularized policy optimization; (4) **DDPG** (Lillicrap, 2015) with deterministic policy gradient and actor-critic architecture.

6.1. Empirical Evaluations in Discrete Action Space

The evaluation covers a broad spectrum of environments, including Classic Control, Box2D, MetaDrive (Li et al., 2022), and Atari domains. These span from simple control tasks to complex, human-like decision-making scenarios with high-dimensional inputs and dynamic variability. CBDQ demonstrates clear advantages in structured control settings, consistently achieving faster convergence and higher final returns by leveraging belief modeling and conceptual abstraction. In more visually complex environments such as Atari, CBDPPO outperforms PPO by effectively integrating belief-guided priors into policy updates, leading to improved sample efficiency and policy robustness.

To evaluate the effectiveness of value-based reinforcement learning under cognitive abstraction, we introduce CBDQ as a belief-augmented extension of Q-learning through modifying DQN algorithm. We benchmark CBDDQN against standard baselines—DQN, DDQN, Dueling DQN, and PPO—across representative environments from Classic Control and Box2D. As shown in Figure 2, CBDDQN consistently demonstrates superior sample efficiency and final performance. In tasks such as *CartPole*, *Acrobot*, and *CarRacing*, CBDDQN attains convergence within the first 20% of training iterations while maintaining the highest observed reward levels throughout learning. To evaluate generalization and adaptability in complex domains, we test CBDDQN on the MetaDrive benchmark, which includes diverse driving scenarios with high-dimensional inputs and dynamic environments. CBDDQN consistently outperforms all baselines, showing stable learning and higher cumulative rewards. Unlike PPO, which often converges suboptimally or shows instability, CBDDQN achieves steady improvement and superior final performance across all settings. Supplementary experiments under varying traffic densities and accident probabilities further confirm CBDQ's robustness in high-risk decision-making settings (Appendix E).

To assess the impact of conceptual belief integration in on-policy reinforcement learning, we compare CBDPPO and standard PPO across eight Atari environments. As shown in Figure 3, CBDPPO consistently achieves higher initial returns and exhibits smoother training dynamics. In tasks such as *AirRaid*, *Breakout*, and *Asteroids*, performance increases rapidly within the first 5–10 million steps and subsequently stabilizes with low variance. In contrast, PPO often exhibits continued fluctuation or degradation in later training stages. The elevated starting performance observed in CBDPPO is attributed to the incorporation of concept-level priors into the initial policy. Specifically, a set of belief distributions is constructed from early episodes using latent clustering, yielding action preferences associated with high-reward states. These priors are integrated into the policy via a convex combination at each timestep, biasing exploration toward semantically coherent behaviors from the outset. Additionally, the belief integration weight β_t increases linearly gradually during training, allowing the agent to transition from task-specific learning to concept-guided refinement. This, combined with the clipped surrogate objective, effectively regularizes policy updates. Once the blended policy enters the trust region, gradient steps become smaller, resulting in reduced variance and stable convergence. These results suggest that conceptually structured priors can improve both the sample efficiency and stability of on-policy optimization, supporting the practical utility of the CBD-RL framework in large-scale and complex control settings.

6.2. Empirical Evaluations in Continuous Action Space

Table 1 presents experimental results across four carefully selected continuous action space environments. These environments represent a spectrum of human-like motor control tasks: from precise end-effector manipulation (Reacher), to coordinated limb movement (Bipedal Walker), to complex multi-joint locomotion (Ant and Humanoid). The results demonstrate CBDSAC's superior performance in capturing human-like control strategies. Particularly in Ant and Humanoid environments, which demand sophisticated bal-

Table 1. Reward results for different algorithms on continuous action environments

Environment/Method	CBDSAC	SAC	DDPG	PPO	A2C
Box2d - BipedalWalker	**295.16 ± 99.64**	285.71 ± 11.43	-17.21 ± 45.45	-34.58 ± 8.92	-115.66 ± 1.95
Mujoco - Ant	**2862.15 ± 606.91**	2386.54 ± 489.76	108.47 ± 14.97	2351.56 ± 147.15	1566.19 ± 346.25
Mujoco - Humanoid	**3248.46 ± 812.84**	2090.07 ± 2233.68	52.35 ± 0.08	401.39 ± 84.60	179.26 ± 74.62
Mujoco - HumanoidStandup	**132391.49 ± 606.23**	121643.72 ± 25.53	112603.41 ± 65.06	69209.17 ± 14951.33	80250.37 ± 46.46
Mujoco - Reacher	**-3.96 ± 0.71**	-4.65 ± 1.77	-6.88 ± 0.08	-5.73 ± 0.96	-10.88 ± 0.12
Mujoco - HalfCheetah	**10276.66 ± 2448.76**	9678.01 ± 810.58	7378.66 ± 1951.02	3574.82 ± 2267.63	3043.32 ± 388.69
Mujoco - Hopper	**3121.56 ± 573.84**	2246.74 ± 657.82	1530.17 ± 1869.52	2338.46 ± 1075.83	520.53 ± 25.98
Mujoco - Walker2d	**4444.48 ± 292.20**	3382.66 ± 1177.36	992.81 ± 1799.20	3756.60 ± 840.68	733.50 ± 755.30
Mujoco - Pusher	**-25.44 ± 6.16**	-31.76 ± 4.15	-36.36 ± 0.82	-45.50 ± 3.14	-55.29 ± 1.65
Mujoco - InvertedPendulum	**998.13 ± 1.87**	860.78 ± 590.78	609.51 ± 4.51	973.82 ± 26.18	991.25 ± 116.64
Mujoco - InvertedDoublePendulum	**9247.71 ± 103.30**	8703.18 ± 644.18	126.87 ± 56.87	6444.11 ± 3857.15	7981.28 ± 1365.03

Figure 3. Learning curves of CBDPPO and PPO on selected Atari tasks. CBDPPO exhibits higher initial performance and lower variance in later stages, indicating improved sample efficiency and training stability through concept-guided policy regularization.

ance and coordination similar to natural movement patterns, CBDSAC maintains consistently higher rewards throughout training. While SAC achieves comparable final performance in Reacher and Bipedal Walker, CBDSAC exhibits notably faster convergence, suggesting more efficient learning of natural movement primitives. This accelerated learning aligns with human motor learning patterns, where basic movement principles are quickly adapted to specific tasks.

7. Research Insight

Effective experience utilization is central to reinforcement learning—not merely as training data, but as a mechanism for bridging past behavior with future decisions. This shift from reactive reward maximization to structured inference echoes Sutton's view of experience as the foundation of intelligent systems (Dohare et al., 2024; Silver & Sutton, 2025), and aligns with cognitive science findings that humans generalize and infer from sparse data with remarkable efficiency (Tenenbaum et al., 2011; Lake et al., 2015; Tenenbaum et al., 2006). CBD-RL embodies this perspective through cognitively inspired mechanisms, yet remains an approximation of human cognition. Current belief–reward integration relies on a fixed convex combination (Eq. 8, 18), whereas human inference is context-sensitive and often modeled by Bayesian model averaging or hierarchical inference (Griffiths & Tenenbaum, 2007; Ma et al., 2006). Incorporating adaptive fusion—e.g., precision-weighted updates or meta-learned gating—could improve alignment with human-like inference under uncertainty. Similarly, the CCF module employs Euclidean clustering, which overlooks the non-Euclidean structure of human conceptual spaces shaped by semantic, attentional, and causal factors (Nosofsky, 1986; Nickel & Kiela, 2017; Murphy, 2004). Enhancing CCF with learned similarity metrics—such as contrastive objectives, hyperbolic embeddings, or causal constraints—offers a promising path toward bridging cognitive theory and representation learning in RL.

8. Conclusion

The Cognitive Belief-Driven Reinforcement Learning framework advances RL by incorporating cognitive-inspired mechanisms. By leveraging conceptual clustering to organize experiences efficiently and employing probabilistic reasoning for decision-making under uncertainty, CBD-RL improves sample efficiency while maintaining computational scalability. Its implementations, CBDQ, CBDPPO and CBDSAC, excel in both discrete and continuous action spaces. This framework bridges cognitive science and reinforcement learning, enhancing algorithmic performance while fostering more interpretable, human-like learning systems. It underscores the potential of cognitive principles to drive more efficient machine learning solutions.

References

Bacon, P.-L., Harb, J., and Precup, D. The option-critic architecture. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Barber, D. Smoothed q-learning. *arXiv preprint arXiv:2303.08631*, 2023.

Botvinick, M. and Weinstein, A. Model-based hierarchical reinforcement learning and human action control. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1655):20130480, 2014.

Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C., and Hassabis, D. Reinforcement learning, fast and slow. *Trends in cognitive sciences*, 23(5):408–422, 2019.

Chiu, Z.-Y., Tuan, Y.-L., Wang, W. Y., and Yip, M. Flexible attention-based multi-policy fusion for efficient deep reinforcement learning. *Advances in Neural Information Processing Systems*, 36:13590–13612, 2023.

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

De Bruin, T., Kober, J., Tuyls, K., and Babuška, R. Experience selection in deep reinforcement learning for control. *Journal of Machine Learning Research*, 19(9):1–56, 2018.

Dennett, D. C. Précis of the intentional stance. *Behavioral and brain sciences*, 11(3):495–505, 1988.

Dohare, S., Hernandez-Garcia, J. F., Lan, Q., Rahman, P., Mahmood, A. R., and Sutton, R. S. Loss of plasticity in deep continual learning. *Nature*, 632(8026):768–774, 2024.

Garg, D., Hejna, J., Geist, M., and Ermon, S. Extreme q-learning: Maxent rl without entropy. *arXiv preprint arXiv:2301.02328*, 2023.

Gershman, S. J., Horvitz, E. J., and Tenenbaum, J. B. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, 349 (6245):273–278, 2015.

Ghavamzadeh, M., Mannor, S., Pineau, J., Tamar, A., et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015.

Gigerenzer, G., Hoffrage, U., and Kleinbölting, H. Probabilistic mental models: a brunswikian theory of confidence. *Psychological review*, 98(4):506, 1991.

Gopnik, A. and Wellman, H. M. Reconstructing constructivism: causal models, bayesian learning mechanisms, and the theory theory. *Psychological bulletin*, 138(6): 1085, 2012.

Griffiths, T. L. and Tenenbaum, J. B. Structure and strength in causal induction. *Cognitive psychology*, 51(4):334–384, 2005.

Griffiths, T. L. and Tenenbaum, J. B. From mere coincidences to meaningful discoveries. *Cognition*, 103(2): 180–226, 2007.

Griffiths, T. L., Chater, N., Kemp, C., Perfors, A., and Tenenbaum, J. B. Probabilistic models of cognition: Exploring representations and inductive biases. *Trends in cognitive sciences*, 14(8):357–364, 2010.

Gu, X., Jiang, C., Wang, E., Wu, Z., Cui, Q., Tian, L., Wu, L., Song, S., and Yu, C. Causkelnet: Causal representation learning for human behaviour analysis. *arXiv preprint arXiv:2409.15564*, 2024a.

Gu, X., Wang, Z., Jin, I., and Wu, Z. Advancing multimodal data fusion in pain recognition: A strategy leveraging statistical correlation and human-centered perspectives. *arXiv preprint arXiv:2404.00320*, 2024b.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Hassani, H., Nikan, S., and Shami, A. Improved exploration–exploitation trade-off through adaptive prioritized experience replay. *Neurocomputing*, 614:128836, 2025.

Hui, D. Y.-T., Courville, A., and Bacon, P.-L. Double gumbel q-learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=UdaTyy0BNB.

Jeen, S., Bewley, T., and Cullen, J. M. Conservative world models. *arXiv preprint arXiv:2309.15178*, 2023.

Jiang, Z. and Luo, S. Neural logic reinforcement learning. In *International conference on machine learning*, pp. 3110–3119. PMLR, 2019.

Johnson-Laird, P. N., Khemlani, S. S., and Goodwin, G. P. Logic, probability, and human reasoning. *Trends in cognitive sciences*, 19(4):201–214, 2015.

Kemp, C. and Tenenbaum, J. B. The discovery of structural form. *Proceedings of the National Academy of Sciences*, 105(31):10687–10692, 2008.

Kimura, D., Chaudhury, S., Wachi, A., Kohita, R., Munawar, A., Tatsubori, M., and Gray, A. Reinforcement learning with external knowledge by using logical neural networks. *arXiv preprint arXiv:2103.02363*, 2021.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253, 2017.

Li, H., Qian, X., and Song, W. Prioritized experience replay based on dynamics priority. *Scientific Reports*, 14(1): 6014, 2024.

Li, Q., Peng, Z., Feng, L., Zhang, Q., Xue, Z., and Zhou, B. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Lillicrap, T. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Liu, W., Cai, W., Jiang, K., Cheng, G., Wang, Y., Wang, J., Cao, J., Xu, L., Mu, C., and Sun, C. Xuance: A comprehensive and unified deep reinforcement learning library. *arXiv preprint arXiv:2312.16248*, 2023.

Ma, M., Liu, J., Sokota, S., Kleiman-Weiner, M., and Foerster, J. N. Learning to coordinate with humans using action features. *CoRR, abs/2201.12658*, 2022.

Ma, W. J., Beck, J. M., Latham, P. E., and Pouget, A. Bayesian inference with probabilistic population codes. *Nature neuroscience*, 9(11):1432–1438, 2006.

Melo, F. S. Convergence of q-learning: A simple proof. *Institute Of Systems and Robotics, Tech. Rep*, pp. 1–4, 2001.

Mnih, V. Asynchronous methods for deep reinforcement learning. *arXiv preprint arXiv:1602.01783*, 2016.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. A. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Mongin and Philippe. Expected utility theory. 1998.

Murphy, G. *The big book of concepts*. MIT press, 2004.

Narvekar, S., Peng, B., Leonetti, M., Sinapov, J., Taylor, M. E., and Stone, P. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50, 2020.

Nickel, M. and Kiela, D. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

Nosofsky, R. M. Attention, similarity, and the identification–categorization relationship. *Journal of experimental psychology: General*, 115(1):39, 1986.

O'Doherty, J., Dayan, P., Schultz, J., Deichmann, R., Friston, K., and Dolan, R. J. Dissociable roles of ventral and dorsal striatum in instrumental conditioning. *science*, 304 (5669):452–454, 2004.

Peng, X. B., Ma, Z., Abbeel, P., Levine, S., and Kanazawa, A. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4):1–20, 2021.

Peterson, C. R. and Beach, L. R. Man as an intuitive statistician. *Psychological bulletin*, 68(1):29, 1967.

Premack, D. and Woodruff, G. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4): 515–526, 1978.

Qiao, G., Liu, G., Poupart, P., and Xu, Z. Multi-modal inverse constrained reinforcement learning from a mixture of demonstrations. *Advances in Neural Information Processing Systems, (NeurIPS)*, 36:60384–60396, 2023.

Qiao, G., Quan, G., Qu, R., and Liu, G. Modelling competitive behaviors in autonomous driving under generative world model. In *European Conference Computer Vision, (ECCV)*, volume 15093, pp. 19–36, 2024a.

Qiao, G., Quan, G., Yu, J., Jia, S., and Liu, G. Trafficgamer: Reliable and flexible traffic simulation for safety-critical scenarios with game-theoretic oracles. *arXiv preprint arXiv:2408.15538*, 2024b.

Rogers, T. T. and McClelland, J. L. *Semantic cognition: A parallel distributed processing approach*. MIT press, 2004.

Rosch, E. Principles of categorization. *Cognition and categorization/Erlbaum*, 1978.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Schultz, W. Neuronal reward and decision signals: from theories to data. *Physiological reviews*, 95(3):853–951, 2015.

Schultz, W., Dayan, P., and Montague, P. R. A neural substrate of prediction and reward. *Science*, 275(5306): 1593–1599, 1997.

Silver, D. and Sutton, R. S. Welcome to the era of experience. *Google AI*, 1, 2025.

Singh, S., Jaakkola, T., Littman, M. L., and Szepesvári, C. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine learning*, 38:287–308, 2000.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tenenbaum, J. B. and Griffiths, T. L. Generalization, similarity, and bayesian inference. *Behavioral and brain sciences*, 24(4):629–640, 2001.

Tenenbaum, J. B., Griffiths, T. L., and Kemp, C. Theory-based bayesian models of inductive learning and reasoning. *Trends in cognitive sciences*, 10(7):309–318, 2006.

Tenenbaum, J. B., Kemp, C., Griffiths, T. L., and Goodman, N. D. How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022):1279–1285, 2011.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. Feudal networks for hierarchical reinforcement learning. In *International conference on machine learning*, pp. 3540–3549. PMLR, 2017.

Watkins, JCH, C., Dayan, and Peter. Q-learning. *Machine learning*, 8:279–292, 1992.

Yin, H. and Pan, S. Knowledge transfer for deep reinforcement learning with hierarchical experience replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

A. Pseudo Code

A.1. Cognitive Belief-Driven Q-Learning (CBDQ) Algorithm

Algorithm 2 Cognitive Belief-Driven Q-Learning Algorithm

Input: Q function $Q(s, a; \phi)$, target Q function$Q(s, a; \phi^-)$, learning rate α, discount factor γ, running steps T, episodes E, replay buffer $\mathcal{B}$ and exploration probability ϵ

Output: $Q^{CBDQ}(s, a; \phi_T)$

Initialize $Q(s, a; \phi)$ with random weights ϕ_0;
Initialize replay buffer $\mathcal{B}$ with a fixed length; Initialize Conceptual Experience Organization Framework (CEOF) categories $\{\mathcal{C}_n\}_{n=1}^{N}$;
Initialize a ϵ-greedy exploration procedure: Explore$(\cdot)$
for each episode **do**
  Get initial state s_0 from the environment
  for each timestep **do**
    Choose action a_t using ϵ-greedy: $a_t \sim \mathcal{U}(0, 1)$
    Execute a_t to get reward $r(s_t, a_t)$, next state s_{t+1}
    Store $(s_t, a_t, r(s_t, a_t), s_{t+1})$ into $\mathcal{B}$
    Identify the conceptual category $\mathcal{C}_i$ of s_t, update the count of a_t in $\mathcal{C}_i$;
    Sample N tuples from $\mathcal{B}$ to update Q function:
      $y^i_{s_t,a_t} = \mathbb{E}_{\mathcal{B}}\left[r(s_t, a_t) + \gamma \sum_a b_t(a \mid s_{t+1}) Q(s_{t+1}, a; \phi^-) | s_t, a_t\right]$
      The computation of $b_t(a \mid s_{t+1})$ in Equation dynamically integrates rewards and subjective beliefs, enabling continuous adaptation based on evolving information.
      $Loss = \mathbb{E}_{\mathcal{B}}\left[(y^i_{s_t,a_t} - Q(s_t, a_t; \phi))^2\right]$
    Reset after a few updates: $\phi^- = \phi$;
  end for
end for

A.2. Cognitive Belief-Driven Soft Actor-Critic (CBDSAC) Algorithm

Algorithm 3 Cognitive Belief-Driven Soft Actor-Critic

Initialize critic parameters ϕ, $\bar{\phi}$ and actor parameters θ
Initialize conceptual categories $\{C_n\}_{n=1}^{N}$
Initialize category belief parameters $\{\mu_k, \sigma_k^2\}_{k=1}^{N}$
for each iteration **do**
  for each environment step **do**
    Identify category C_k containing s_t
    Compute $b_t(\cdot \mid s_t) = (1-\beta_t)\pi_\theta(\cdot \mid s_t) + \beta_t \mathcal{P}_k(\cdot \mid s_t)$
    Sample $a_t \sim b_t(\cdot \mid s_t)$
    Transition to $s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)$
    Store transition in replay buffer: $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s_t, a_t, r(s_t, a_t), s_{t+1})\}$
    Update category belief parameters:
    $\mu_k \leftarrow \frac{\sigma^2_{\text{prior}}\mu_{\text{obs}} + \sigma^2_{\text{obs}}\mu_{\text{prior}}}{\sigma^2_{\text{prior}} + \sigma^2_{\text{obs}}}$
    $\sigma_k^2 \leftarrow \frac{\sigma^2_{\text{prior}}\sigma^2_{\text{obs}}}{\sigma^2_{\text{prior}} + \sigma^2_{\text{obs}}}$
  end for
  for each gradient step **do**
    The critic loss function is minimized as:

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - Q_\phi(s_i, a_i))^2$$

    Use parameter updates based on action a_i, then update the θ with Equation 19.
    Update the temperature coefficient α.
    $\bar{\phi} \leftarrow \tau\phi + (1-\tau)\bar{\phi}$
  end for
end for

A.3. Cognitive Belief-Driven Proximal Policy Optimization (CBDPPO) Algorithm

Algorithm 4 Cognitive Belief-Driven Proximal Policy Optimization

Initialize policy parameters θ_0 and value function parameters ϕ_0
Initialize conceptual categories $\{C_n\}_{n=1}^{N}$
for each iteration **do**
  for each environment step **do**
    Collect set of trajectories $D_k = \{\tau_i\}$ by running $\pi_k = \pi(\theta_k)$
    Sample a_t and Transition to get s_{t+1}
    Compute rewards-to-go $r(s_t, a_t)$.
    Compute advantage estimation A_t based on current value function V_{ϕ_k}
    Store transition in replay buffer: $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s_t, a_t, r(s_t, a_t), s_{t+1}, A_t)\}$
  end for
  for each gradient step **do**
    Identify category C_k containing s_t
    Compute $b_t(a \mid s) = (1 - \beta_t)\pi_\theta(a \mid s) + \beta_t P_k(a \mid s)$, based on Equation 20
    Update the policy by maximizing the PPO-Clip objective:
    $\theta_{k+1} = \underset{\theta}{argmax}\, \frac{1}{|D_k|T} \sum_{\tau \in D_k} \sum_{t=0}^{T} \Big[\min \Big(\frac{b_t(a|s)}{\pi_{\theta_k}(a|s)} A_t, \text{clip} \Big(\frac{b_t(a|s)}{\pi_{\theta_k}(a|s)}, 1 - \epsilon, 1 + \epsilon \Big) A_t \Big) \Big]$, based on Equation 21
    Fit value function by regression on mean-squared error:
    $\phi_{k+1} = \underset{\phi}{argmax}\, \frac{1}{|D_k|T} \sum_{\tau \in D_k} \sum_{t=0}^{T} \Big(V_\phi(s_t) - r(s_t, a_t) \Big)^2$
  end for
end for

B. Smoothed Bellman Operator

B.1. Lemma

Lemma B.1 (Jensen's Inequality for Q-values)**.** *Consider an MDP with state s_{t+1} and actions a, along with Q-value estimates $Q_t(s_{t+1}, a)$. Let $q_t(a \mid s_{t+1})$ denote the probability of selecting action a in state s_{t+1}. By Jensen's inequality:*

$$\gamma \sum_{s_{t+1}} P(s_{t+1} \mid s_t, a_t) \sum_{a'} q_t(a \mid s_{t+1}) Q_t(s_{t+1}, a) \leq \gamma \sum_{s_{t+1}} P(s_{t+1} \mid s_t, a_t) \max_a Q_t(s_{t+1}, a), \tag{22}$$

Lemma B.2 (Convergence of Smoothed Bellman Operator)**.** *Let $\{Q_t\}$ be the sequence generated by iteratively applying $\mathcal{T}_{\text{Smoothed}}$. Under the condition:*

$$\lim_{t \to \infty} \max_a q_t(a \mid s_{t+1}) = 1, \tag{23}$$

for the optimal action, Q_t converges to the optimal Q^* as $t \to \infty$. See Appendix D for a detailed proof.

B.2. Smoothing Bellman Strategy

Strategy	Formula
Softmax	$b_t = \frac{e^{Q(s,a)}}{\sum_b e^{Q(s,b)}}$
Clipped Max	$b_t = \begin{cases} 1-\tau, & \text{if } a = a^* \\ \frac{\tau}{A-1}, & \text{if } a \neq a^* \end{cases}$
Clipped Softmax	$b_t = \begin{cases} \frac{e^{\beta Q(s,a)}}{\sum_{b \in I} e^{\beta Q(s,b)}}, & \text{if } a \in I \\ 0, & \text{if } a \notin I \end{cases}$
Bayesian Inference	$Q_{\text{adjusted}}(s,a) = Q(s,a) + \mu_{\text{prior}}$ $b_t = \frac{e^{Q_{\text{adjusted}}(s',a)}}{\sum_b e^{Q_{\text{adjusted}}(s',b)}}$ $\sigma^2_{\text{posterior}} = \left(\frac{1}{\sigma^2_{\text{prior}}} + \frac{n}{\sigma^2_{\text{observation}}} \right)^{-1}$ $\mu_{\text{posterior}} = \sigma^2_{\text{posterior}} \left(\frac{\mu_{\text{prior}}}{\sigma^2_{\text{prior}}} + \sum_{i=1}^{n} \frac{r_i}{\sigma^2_{\text{observation}}} \right)$

Table 2. Smoothing strategies with respective formulas

C. Convergence Proof

We outline a proof that builds upon the following result (Singh et al., 2000; Barber, 2023) and follows the framework provided in (Melo, 2001):

Theorem C.1. *The random process $\{\Delta_t\}$ taking value in $\mathbb{R}$ and defined as*

$$\Delta_{t+1}(x) = (1 - \alpha_t(x))\Delta_t(x) + \alpha_t(x)F_t(x) \tag{24}$$

converges to 0 with probability 1 under the following assumptions:

- $0 \leq \alpha_t \leq 1$, $\sum_t \alpha_t(x) = \infty$, $\sum_t \alpha_t^2(x) < \infty$;
- $\mathbb{E}[\|F_t(x)\|_W] \leq \kappa \|\Delta_t\|_W + c_t$, $\kappa \in [0,1)$ *and* $c_t \to 0$ *with probability 1;*
- $var(F_t(x)) \leq C(1 + \|\Delta_t\|_W)^2$, $C > 0$

where $\|\Delta_t\|_W$ denotes a weighted max norm.

We are interested in the convergence of Q_t towards the optimal value Q_* and therefore define

$$\Delta_t = Q_t(s_t, a_t) - Q_*(s_t, a_t) \tag{25}$$

It is convenient to write the smoothed update as

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t(s_t, a_t) \left(r_t + \gamma \left\langle Q(s_{t+1}, a) \right\rangle_a - Q_t(s_t, a_t) \right) \tag{26}$$

where $\langle f(x) \rangle_x$ means the expectation of the function $f(x)$ with respect to the distribution of x. Using the smoothed update, we can write

$$\Delta_{t+1}(s_t, a_t) = Q_{t+1}(s_t, a_t) - Q_*(s_t, a_t) \tag{27}$$

$$= (1 - \alpha_t)\Delta_t + \alpha_t \left(r_t + \gamma \langle Q(s_{t+1}, a) \rangle_a - Q_*(s_t, a_t) \right) \tag{28}$$

In terms of Theorem 1, we therefore define

$$F_t = r_t + \gamma \sum_a q_t(a|s_{t+1}) Q_t(s_{t+1}, a) - Q_*(s_t, a_t) \tag{29}$$

Proof. For convergence, we need to verify the conditions of Theorem 1.

Step 1: Verify Step-Size Conditions

We assume that the learning rates $\alpha_t(s_t, a_t)$ satisfy:

- $0 < \alpha_t(s_t, a_t) \leq 1$,
- $\sum_t \alpha_t(s_t, a_t) = \infty$,
- $\sum_t \alpha_t^2(s_t, a_t) < \infty$.

An example is $\alpha_t(s_t, a_t) = \frac{1}{N_t(s_t, a_t)}$, where $N_t(s_t, a_t)$ is the visitation count of (s_t, a_t).

Step 2: Establish Boundedness of Q_t

Since the rewards r_t are bounded ($|r_t| \leq R_{\text{max}}$) and the discount factor $0 < \gamma < 1$, we can show that Q_t remains bounded independently of the convergence of Δ_t.

Define the Bound Q_{max}:

We define

$$Q_{\text{max}} = \frac{R_{\text{max}}}{1 - \gamma}. \tag{30}$$

This is the maximum possible value of the Q-function given the bounded rewards and discount factor.

Derivation of Q_{max}:

The Q-function $Q(s, a)$ represents the expected cumulative discounted reward when starting from state s and taking action a:

$$Q(s, a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \,\middle|\, s_t = s, a_t = a\right], \tag{31}$$

where r_{t+k} is the reward received at time $t + k$, and γ is the discount factor.

Assuming that at each time step, the agent receives the maximum possible reward R_{max}, the maximum possible Q-value is:

$$Q_{\text{max}} = \sum_{k=0}^{\infty} \gamma^k R_{\text{max}} = R_{\text{max}} \sum_{k=0}^{\infty} \gamma^k. \tag{32}$$

Since $0 < \gamma < 1$, the infinite sum $\sum_{k=0}^{\infty} \gamma^k$ is a geometric series that sums to:

$$\sum_{k=0}^{\infty} \gamma^k = \frac{1}{1 - \gamma}. \tag{33}$$

Therefore, we have:

$$Q_{\text{max}} = R_{\text{max}} \times \frac{1}{1 - \gamma} = \frac{R_{\text{max}}}{1 - \gamma}. \tag{34}$$

Thus, $Q_{\text{max}} = \dfrac{R_{\text{max}}}{1 - \gamma}$ is the maximum possible value of the Q-function in any state-action pair.

Base Case: Let $Q_0(s, a)$ be initialized such that $|Q_0(s, a)| \leq Q_{\text{max}}$ for all s, a.

Inductive Step: Assume $|Q_t(s, a)| \leq Q_{\text{max}}$ for all s, a. We need to show that $|Q_{t+1}(s_t, a_t)| \leq Q_{\text{max}}$.

From the update equation:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t(s_t, a_t) \left(r_t + \gamma \left\langle Q_t(s_{t+1}, a)\right\rangle_a - Q_t(s_t, a_t)\right). \tag{35}$$

Simplifying:

$$Q_{t+1}(s_t, a_t) = (1 - \alpha_t(s_t, a_t))Q_t(s_t, a_t) + \alpha_t(s_t, a_t)\left(r_t + \gamma \left\langle Q_t(s_{t+1}, a)\right\rangle_a\right). \tag{36}$$

Taking absolute values:

$$|Q_{t+1}(s_t, a_t)| \leq (1 - \alpha_t)|Q_t(s_t, a_t)| + \alpha_t\left(|r_t| + \gamma \left|\left\langle Q_t(s_{t+1}, a)\right\rangle_a\right|\right). \tag{37}$$

Using the inductive hypothesis and boundedness:

$$|Q_t(s_t, a_t)| \leq Q_{\max}, \quad \left|\left\langle Q_t(s_{t+1}, a)\right\rangle_a\right| \leq Q_{\max}, \tag{38}$$

and $|r_t| \leq R_{\max}$. Therefore:

$$|Q_{t+1}(s_t, a_t)| \leq (1 - \alpha_t)Q_{\max} + \alpha_t\left(R_{\max} + \gamma Q_{\max}\right). \tag{39}$$

Simplify:

$$|Q_{t+1}(s_t, a_t)| \leq Q_{\max} - \alpha_t Q_{\max} + \alpha_t\left(R_{\max} + \gamma Q_{\max}\right) \tag{40}$$

$$= Q_{\max} + \alpha_t\left(R_{\max} - (1 - \gamma)Q_{\max}\right). \tag{41}$$

Since $Q_{\max} = \dfrac{R_{\max}}{1 - \gamma}$, we have $(1 - \gamma)Q_{\max} = R_{\max}$. Substituting back:

$$|Q_{t+1}(s_t, a_t)| \leq Q_{\max} + \alpha_t\left(R_{\max} - R_{\max}\right) = Q_{\max}. \tag{42}$$

Thus,

$$|Q_{t+1}(s_t, a_t)| \leq Q_{\max}. \tag{43}$$

Therefore, by induction, Q_t remains bounded for all t, independently of Δ_t.

Step 3: Verify Mean Condition

We can write

$$\frac{1}{\gamma}\mathbb{E}[F_t] = \mathbb{E}_{p_\mathcal{T}}[G_t], \tag{44}$$

where

$$G_t = \sum_a q_t(a|s_{t+1})Q_t(s_{t+1}, a) - \max_a Q_*(s_{t+1}, a). \tag{45}$$

We can form the bound

$$\left\|\frac{1}{\gamma}\mathbb{E}[F_t]\right\|_\infty = \left\|\mathbb{E}[G_t]\right\|_\infty \leq \|G_t\|_\infty, \tag{46}$$

which means that if we can bound $\|G_t\|_\infty$ appropriately, the mean criterion will be satisfied.

Assuming that b_t places $(1 - \delta_t)$ mass on the maximal action $a^* = \arg\max_a Q_t(s_{t+1}, a)$, we can write

$$G_t = \sum_a q_t(a|s_{t+1})Q_t(s_{t+1}, a) - \max_a Q_*(s_{t+1}, a) \tag{47}$$

$$= (1 - \delta_t)Q_t(s_{t+1}, a^*) + \delta_t \sum_{c \neq a^*} \tilde{q}_t(c|s_{t+1})Q_t(s_{t+1}, c) - \max_a Q_*(s_{t+1}, a), \tag{48}$$

where $\tilde{b}_t(c|s_{t+1}) = \frac{b_t(c|s_{t+1})}{\delta_t}$ for $c \neq a^*$.

We can then write

$$G_t = Q_t(s_{t+1}, a^*) - \max_a Q_*(s_{t+1}, a) + \delta_t\left(\sum_{c \neq a^*} \tilde{b}_t(c|s_{t+1})[Q_t(s_{t+1}, c) - Q_t(s_{t+1}, a^*)]\right). \tag{49}$$

Since $Q_t(s_{t+1}, a^*) \geq Q_t(s_{t+1}, c)$ for all c, the terms inside the brackets are non-positive. Therefore,

$$G_t \leq Q_t(s_{t+1}, a^*) - \max_a Q_*(s_{t+1}, a). \tag{50}$$

Now, we have

$$Q_t(s_{t+1}, a^*) - \max_a Q_*(s_{t+1}, a) = [Q_t(s_{t+1}, a^*) - Q_*(s_{t+1}, a^*)] + [Q_*(s_{t+1}, a^*) - \max_a Q_*(s_{t+1}, a)] \tag{51}$$

$$\leq \Delta_t(s_{t+1}, a^*). \tag{52}$$

Thus,

$$G_t \leq \Delta_t(s_{t+1}, a^*). \tag{53}$$

Therefore,

$$\|G_t\|_\infty \leq \|\Delta_t\|_\infty. \tag{54}$$

Additionally, the term involving δ_t contributes an additional c_t, which is bounded due to the boundedness of Q_t and $\delta_t \to 0$. Thus, the mean condition becomes

$$\|\mathbb{E}[F_t]\|_\infty \leq \gamma \|\Delta_t\|_\infty + c_t, \tag{55}$$

with $c_t \to 0$ as $\delta_t \to 0$.

Since $\gamma < 1$, the mean condition is satisfied with $\kappa = \gamma$ and $c_t \to 0$.

Step 4: Verify Variance Condition

Since the rewards r_t are bounded and we have established that Q_t is bounded independently, F_t is also bounded.

We can write:

$$\Delta F_t = F_t - \mathbb{E}[F_t] \tag{56}$$

$$= (r_t - \mathbb{E}[r_t|s_t, a_t]) + \gamma \left(\sum_a q_t(a|s_{t+1}) Q_t(s_{t+1}, a) - \mathbb{E}_{s_{t+1}} \left[\sum_a q_t(a|s_{t+1}) Q_t(s_{t+1}, a) \right] \right). \tag{57}$$

We can bound the variance using

$$\text{Var}(F_t) = \mathbb{E}\left[(\Delta F_t)^2 \mid \mathcal{F}_t \right] \leq \|\Delta F_t\|_\infty^2. \tag{58}$$

Using the triangle inequality,

$$\|\Delta F_t\|_\infty \leq \|\Delta r_t\|_\infty + \gamma \left\| \sum_a q_t(a|s_{t+1}) Q_t(s_{t+1}, a) - \mathbb{E}_{s_{t+1}} \left[\sum_a q_t(a|s_{t+1}) Q_t(s_{t+1}, a) \right] \right\|_\infty \tag{59}$$

$$\leq \|\Delta r_t\|_\infty + \gamma \left\| Q_t(s_{t+1}, a) - \mathbb{E}_{s_{t+1}}[Q_t(s_{t+1}, a)] \right\|_\infty. \tag{60}$$

Since Q_t is bounded, there exists a constant B such that

$$\|Q_t(s_{t+1}, a) - \mathbb{E}_{s_{t+1}}[Q_t(s_{t+1}, a)]\|_\infty \leq 2Q_{\max} = B. \tag{61}$$

Therefore,

$$\|\Delta F_t\|_\infty \leq \|\Delta r_t\|_\infty + \gamma B. \tag{62}$$

Since r_t is bounded, $\|\Delta r_t\|_\infty \leq 2R_{\max}$.

Thus,

$$\|\Delta F_t\|_\infty \leq 2R_{\max} + \gamma B. \tag{63}$$

Therefore, the variance is bounded, and there exists a constant $C > 0$ such that

$$\text{Var}(F_t) \leq C(1 + \|\Delta_t\|_\infty)^2. \tag{64}$$

Step 5: Conclusion

All the conditions of Theorem 1 are satisfied:

- **Step-Size Conditions:** Verified in Step 1.
- **Mean Condition:** Verified in Step 3, with $\kappa = \gamma < 1$ and $c_t \to 0$.
- **Variance Condition:** Verified in Step 4.

Therefore, $\Delta_t \to 0$ with probability 1, implying that $Q_t \to Q_*$ with probability 1.

□

D. Experiment Setting

D.1. Classic Control and Box 2D Environment

Figure 4. Cartpole, Acrobot, CarRacing, Lunar Lander and Bipedal Walker .

1. Cartpole: a pole is attached by an unactuated joint to a cart, which moves along a frictionless track. The pendulum is placed upright on the cart and the goal is to balance the pole by applying forces in the left and right direction on the cart.
2. Acrobot: a two-link pendulum system with only the second joint actuated. The task is to swing the lower link to a sufficient height in order to raise the tip of the pendulum above a target height. The environment challenges the agent's ability to apply precise control for coordinating multiple linked joints.
3. CarRacing: The easiest control task to learn from pixels - a top-down racing environment. The generated track is random in every episode.
4. Lunar Lander: It is a classic rocket trajectory optimization problem. According to Pontryagin's maximum principle, it is optimal to fire the engine at full throttle or turn off. This is why this environment has discrete actions: engine on or off.
5. Bipedal Walker: a two-legged robot attempting to walk across varied terrain. The goal is for the agent to learn how to navigate efficiently and avoid falling.

D.2. MetaDrive Block Type Description

Table 3. Block Types Used in Experiments

ID	Block Type
S	Straight
C	Circular
r	InRamp
R	OutRamp
O	Roundabout
X	Intersection
y	Merge
Y	Split
T	T-Intersection

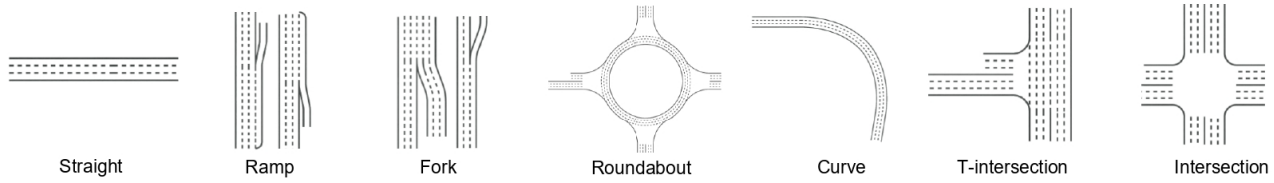

Figure 5. Various block types used in the MetaDrive environment. These blocks represent common road structures such as straight roads, ramps, forks, roundabouts, curves, T-intersections, and intersections, used for evaluating the vehicle's path planning and decision-making capabilities.

D.3. Map Design and Testing Objectives

D.3.1. Map 1: SrOYCTRyS

This map consists of straight roads, roundabouts, intersections, T-intersections, splits, and ramps. The environment presents a highly complex combination of multiple intersections, dynamic traffic flow, and varying road structures.

Testing Objective: The focus of this environment is to evaluate the algorithm's smooth decision-making and multi-intersection handling, mimicking human driving behavior. The challenges include adjusting vehicle paths in real-time and ensuring smooth lane transitions in the presence of complex road structures such as roundabouts and ramps.

D.3.2. Map 2: COrXSrT

This map combines circular roads, roundabouts, straight roads, intersections, ramps, and T-intersections. The environment is designed to assess the vehicle's decision-making capabilities when dealing with continuous changes in road grades and multiple intersection types.

Testing Objective: This environment tests the algorithm's ability to dynamically adjust to **grade changes** and **multi-intersection interactions**, replicating human-like behavior. The goal is to observe how well the algorithm adjusts vehicle speed and direction, ensuring stability in scenarios involving ramps and complex road networks.

D.3.3. Map 3: rXTSC

This map consists of ramps, intersections, T-intersections, straight roads, and circular roads. The environment simulates multiple road interactions, testing the vehicle's path selection and stability, particularly at intersections and ramps.

Testing Objective: This environment evaluates the algorithm's performance in handling intersections and T-junctions with real-time path selection. The challenge is to ensure human-like adaptability when encountering multiple directional options, maintaining decision stability in dynamic traffic situations.

D.3.4. MAP 4: YORSX

This map includes splits, roundabouts, straight roads, circular roads, and intersections. The environment is tailored to test the vehicle's ability to make path decisions in high-speed settings, particularly when merging traffic and navigating through complex junctions.

Testing Objective: The map focuses on testing the vehicle's ability to handle **high-speed lane merging** and **dynamic path planning**. The algorithm must mimic human drivers by making real-time adjustments in a high-speed environment, choosing optimal paths while maintaining speed control and safety through complex intersections and roundabouts.

D.3.5. MAP 5: XTOC

This map features circular roads, T-intersections, and straight roads, creating a unique combination of continuous curves and abrupt directional changes. The environment presents the challenge of maintaining speed while negotiating tight turns and quick transitions at T-intersections.

Testing Objective: The focus is on testing the vehicle's ability to handle **sharp directional changes** and maintain control during high-speed maneuvers. The algorithm needs to balance speed with precision, ensuring safe navigation through tight turns and abrupt intersections.

D.3.6. MAP 6: XTSC

This map features a T-shaped intersection with traffic signals controlling vehicle flow from three directions. It tests advanced driving skills including traffic light compliance, turn management, and interaction with vehicles from cross directions.

Testing Objective: The main challenge is to evaluate the vehicle's ability to maintain **lane stability** and make appropriate **speed adjustments** while navigating long straight roads and transitioning into a circular roundabout. The algorithm must ensure smooth control and decision-making, simulating human-like behavior in handling both high-speed straight roads and slower, more controlled turns in the roundabout.

D.3.7. MAP 7: TORXS

This map consists of T-intersections, roundabouts, straight roads, and splits, forming a compact yet intricate structure. The layout challenges the algorithm to manage dynamic path selection and adapt to sudden directional changes within a moderately complex road network.

Testing Objective: The primary objective is to evaluate the algorithm's ability to manage split paths and handle sudden directional changes. The map focuses on the vehicle's adaptability in navigating roundabouts and maintaining stability while making real-time path decisions at T-intersections.

D.3.8. MAP 8: CYRXT

This map integrates circular roads, Y-intersections, ramps, T-intersections, and straight roads, creating a dynamic and highly interconnected network. The layout introduces varying road geometries and frequent directional changes, requiring seamless decision-making and adaptability.

Testing Objective: The map is designed to test the algorithm's ability to adapt to sudden directional shifts at Y-intersections and T-junctions, maintain stability on ramps, and execute precise maneuvers on circular roads. The emphasis is on smooth transitions between road types, effective navigation through interconnected pathways, and robust handling of diverse traffic scenarios.

D.4. MuJoCo Environments

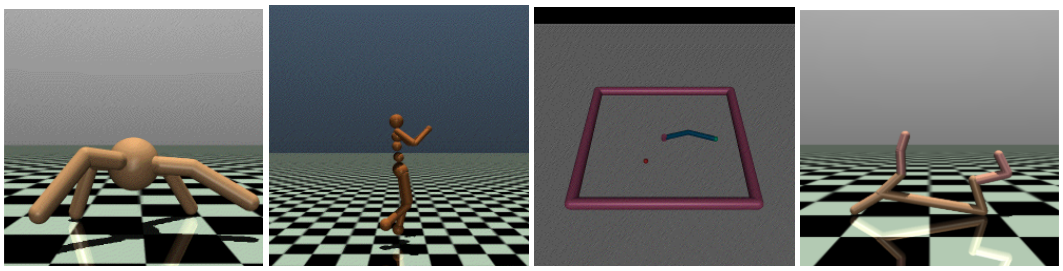

Figure 6. Ant, Humanoid, Reacher and Half Cheetah.

1. Ant: a 3D robot with a single central torso and four articulated legs is designed to navigate in the forward direction. The robot's movement depends on coordinating the torque applied to the hinges that connect the legs to the torso and the segments within each leg.
2. Humanoid: a 3D bipedal robot simulates human gait, with a torso, a pair of legs, and arms. Each leg and arm consists of two segments, representing the knees and elbows respectively; the legs are used for walking, while the arms assist with balance. The robot's goal is to walk forward as quickly as possible without falling.
3. Humanoid Standup: The environment starts with the humanoid laying on the ground, and then the goal of the environment is to make the humanoid stand up and then keep it standing by applying torques to the various hinges.
4. Reacher: a two-jointed robot arm. The goal is to move the robot's end effector close to a target that is spawned at a random position.
5. Half Cheetah: a 2-dimensional robot consisting of 9 body parts and 8 joints connecting them (including two paws). The goal is to apply torque to the joints to make the cheetah run forward (right) as fast as possible, with a positive reward based on the distance moved forward and a negative reward for moving backward.
6. Hopper: a two-dimensional one-legged figure consisting of four main body parts - the torso at the top, the thigh in the middle, the leg at the bottom, and a single foot on which the entire body rests. The goal is to make hops that move in the forward (right) direction by applying torque to the three hinges that connect the four body parts.
7. Walker-2d: a two-dimensional bipedal robot consisting of seven main body parts - a single torso at the top (with the two legs splitting after the torso), two thighs in the middle below the torso, two legs below the thighs, and two feet attached to the legs on which the entire body rests. The goal is to walk in the forward (right) direction by applying torque to the six hinges connecting the seven body parts.
8. Pusher: a multi-jointed robot arm that is very similar to a human arm. The goal is to move a target cylinder (called object) to a goal position using the robot's end effector (called fingertip).
9. Inverted Pendulum: The environment consists of a cart that can be moved linearly, with a pole attached to one end and having another end free. The cart can be pushed left or right, and the goal is to balance the pole on top of the cart by applying forces to the cart.
10. Inverted Double Pendulum: The environment involves a cart that can be moved linearly, with one pole attached to it and a second pole attached to the other end of the first pole (leaving the second pole as the only one with a free end). The cart can be pushed left or right, and the goal is to balance the second pole on top of the first pole, which is in turn on top of the cart, by applying continuous forces to the cart.

D.5. Atari Environments

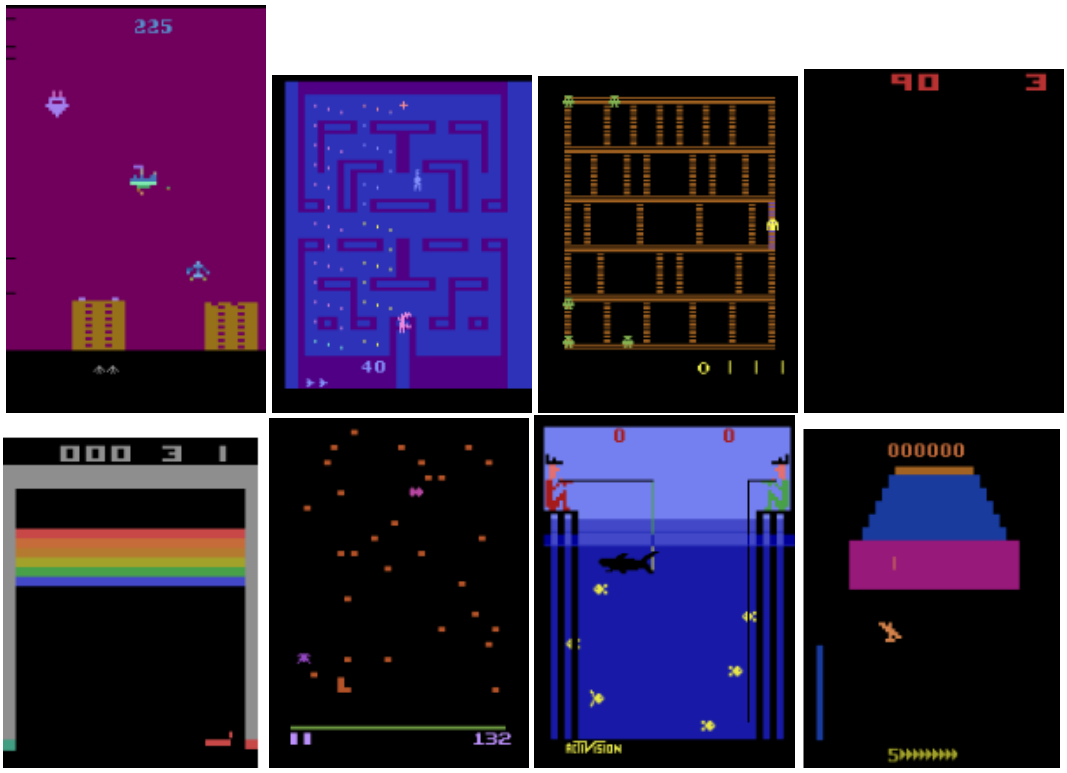

Figure 7. Air Raid, Alien, Amidar, Asteroids, Breakout, Centipede, Fishing Derby, Zaxxon.

1. Air Raid: You control a ship that can move sideways and protect two buildings (one on the right and one on the left side of the screen) from flying saucers that are trying to drop bombs on them.
2. Alien: You are stuck in a maze-like space ship with three aliens. You goal is to destroy their eggs that are scattered all over the ship while simultaneously avoiding the aliens (they are trying to kill you).
3. Admidar: You are trying to visit all places on a 2-dimensional grid while simultaneously avoiding your enemies. You can turn the tables at one point in the game: Your enemies turn into chickens and you can catch them.
4. Asteroids: You control a spaceship in an asteroid field and must break up asteroids by shooting them. Once all asteroids are destroyed, you enter a new level and new asteroids will appear. You will occasionally be attacked by a flying saucer.
5. Breakout: You move a paddle and hit the ball in a brick wall at the top of the screen. Your goal is to destroy the brick wall. You can try to break through the wall and let the ball wreak havoc on the other side, all on its own! You have five lives.
6. Centipede: You are an elf and must use your magic wands to fend off spiders, fleas and centipedes. Your goal is to protect mushrooms in an enchanted forest.
7. Fishing Derby: Your objective is to catch more sunfish than your opponent.
8. Zaxxon: Your goal is to stop the evil robot Zaxxon and its armies from enslaving the galaxy by piloting your fighter and shooting enemies.

D.6. Environment Parameter & Agent Parameter

Table 4. Q-family vs PPO Algorithm and Environment Parameters

Parameter	**Q-Family**	**PPO**
Discrete Action Space	True	
Policy	Basic_Q_network	Categorical_AC
Representation	Basic_MLP	
Runner	DRL	
Representation Hidden Size	[256, 256]	[512,]
Q/Actor Hidden Size	[256, 256]	[256, 256]
Critic Hidden Size	N/A	[256, 256]
Activation Function	relu	leaky_relu
Activation for Actions	N/A	tanh
Seed	123 / 321 / 666	
Number of Parallels	10	
Buffer Size	500,000	Horizon_Size * Parallels (128 * 10)
Batch Size	64	N/A
Horizon Size	N/A	128
Number of Epochs	N/A	4
Number of Minibatches	N/A	4
Learning Rate	0.00025	
Start Greedy	1.0	N/A
End Greedy	0.01	N/A
Decay Step for Greedy	50,000	N/A
Sync Frequency	50	N/A
Training Frequency	1	N/A
Start Training Step	1,000	N/A
Running Steps	2,000,000	
Use Gradient Clipping	N/A	True
Value Function Coefficient	N/A	0.25
Entropy Coefficient	N/A	0.0
Target KL Divergence	N/A	0.001
Clip Range	N/A	0.2
Clip Gradient Norm	N/A	0.5
Gamma	0.99	
Use GAE	N/A	True
GAE Lambda	N/A	0.95
Use Advantage Normalization	N/A	True
Use Observation Normalization	False	True
Use Reward Normalization	False	True
Observation Normalization Range	5	
Reward Normalization Range	5	
Test Steps	10,000	
Evaluation Interval	50,000	5,000
Test Episodes	5	

E. Running Setting

For the Cartpole and Lunar Lander environments, the training process utilizes 1 RTX 3060 and typically runs less than 30 minutes. For the Carracing environment, we require 1 RTX 3060 and 2 hours of running. For the Metadrive environments, the training process utilizes 1 RTX 3060 and typically runs around 3-6 hours according to different complexity.