Cognitive and Semantic Prototype–Based Decision–Making for Game Agents

Augusto Baffa[1], **Edward Hermann**[1]

[1]Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Rio de Janeiro, RJ – Brasil

{abaffa, hermann}@inf.puc-rio.br

Abstract. Introduction: *Decision–making in games involves a trade–off between interpretability and adaptability. Symbolic approaches such as Finite State Machines (FSMs) provide transparency but limited flexibility, while deep reinforcement learning methods such as Proximal Policy Optimization (PPO) offer performance, but need extensive training data and are harder to explain.* ***Objective:*** *This work investigates prototype–based decision–making and how it can serve as a middle ground between symbolic and sub–symbolic methods, combining interpretability and adaptive performance in real–time environments.* ***Methodology:*** *A cognitive architecture based on prototype theory is proposed, integrating semantic state representation, conceptual memory, and local reinforcement through multi–armed bandits, without neural networks. A key feature is the separation between perceptual similarity and decision competence. The approach is evaluated in an Asteroids environment against FSM and PPO baselines under identical conditions.* ***Results:*** *The proposed controller achieved overall performance in terms of return, score, and stability relative to the FSM and PPO baselines under identical test conditions. This suggests that prototype–based decision–making may provide an interpretable alternative with adaptive performance comparable to the baselines.*
Keywords *Game AI, Interpretable Reinforcement Learning, Prototype–Based Decision Making, Multi–Armed Bandits, Cognitive Architectures*

1. Introduction

Decision–making in games has historically evolved along two dominant paradigms: symbolic and sub–symbolic approaches. Symbolic methods, such as Finite State Machines (FSMs), provide deterministic, interpretable, and computationally efficient policies, making them widely adopted in real–time and resource–constrained environments [Samuel et al. 2021]. However, these approaches require extensive manual design and exhibit limited adaptability in complex or dynamic scenarios.

In contrast, sub–symbolic approaches based on deep reinforcement learning (DRL), including policy–gradient methods such as Proximal Policy Optimization (PPO), achieve strong performance across sequential decision–making tasks. These methods enable powerful function approximation and adaptive behavior but introduce high sample complexity, sensitivity to hyperparameters, and reduced interpretability [Li et al. 2025]. As a result, understanding and controlling agent behavior remains challenging, particularly in interactive domains where designer oversight is essential.

This trade–off has motivated recent efforts to investigate alternatives that combine structured control with adaptive behavior. One direction is prototype–based decision–making grounded in cognitive theories of categorization and conceptual spaces [Cai et al. 2019, Rosch 1975]. In such approaches, decisions are based on similarity to representative prototypes and accumulated experience within semantically meaningful regions of the state space.

This work proposes a prototype–based cognitive decision–making architecture for real–time Game AI that combines interpretability with adaptive behavior through semantic state representation, prototype–based memory, and local reinforcement. The method differs from prior prototype–based reinforcement learning approaches in three main aspects: (i) explicit separation between perceptual similarity and decision competence; (ii) modeling of action selection within each prototype as a local multi–armed bandit problem [Chaudhuri et al. 2023]; and (iii) absence of neural function approximation, preserving full transparency of internal structures.

The approach is evaluated in the Asteroids game as a controlled real–time environment, characterized by a continuous and geometrically interpretable observation space. Three paradigms are compared under identical conditions: a symbolic FSM controller, a PPO–based DRL agent, and the proposed prototype–based bandit controller.

This study investigates the following question: *Can a fully interpretable, non–neural prototype–based architecture match or approach the performance of both symbolic and deep reinforcement learning approaches in real–time game environments?*

Prototype–based reinforcement learning partially addresses this issue by introducing interpretable reference points. However, existing methods typically embed prototypes within neural architectures, where decision–making remains governed by non–transparent models [Kenny et al. 2023, Chen et al. 2019]. Similarly, concept–based policy distillation methods depend on pre–trained black–box models [Bekkemoen e Langseth 2025].

Moreover, few works evaluate prototype–based approaches as standalone decision systems in real–time environments alongside both symbolic and DRL baselines under controlled conditions. Consequently, it remains unclear whether such models can effectively bridge the gap between interpretability and adaptability. This work addresses this gap by proposing a fully interpretable, non–neural prototype–based controller and evaluating it against FSM and PPO baselines, considering performance, stability, sample efficiency, and transparency.

2. Related Work

Recent research on autonomous agents has examined how to balance *performance*, *adaptivity*, and *interpretability*. In this context, the literature can be broadly organized into three main paradigms: (i) symbolic and rule–based control, (ii) deep reinforcement learning, and (iii) interpretable learning approaches based on prototypes or conceptual abstractions.

Symbolic and Rule–Based Controllers Classic game AI approaches are extensively documented in foundational works, emphasizing symbolic control structures such as Finite State Machines (FSMs) and Behavior Trees [Millington e Funge 2009]. These

methods remain widely used due to their interpretability, determinism, and ease of debugging, making them particularly suitable for real–time systems where predictability and control are essential. Recent work has revisited structured policies as viable alternatives to purely learned systems, especially in safety–critical or design–sensitive applications. Rule–guided reinforcement learning approaches further demonstrate that symbolic knowledge can be leveraged to constrain or improve learned policies [Tappler et al. 2025]. Despite these advantages, symbolic systems lack adaptivity and typically require extensive manual design, limiting their scalability in complex or highly dynamic environments.

Deep Reinforcement Learning The application of deep reinforcement learning (DRL) to games gained prominence with Deep Q–Networks (DQN) [Mnih et al. 2015], followed by improvements such as Double DQN and Rainbow [Hessel et al. 2018]. Policy gradient methods, including Proximal Policy Optimization (PPO) [Schulman et al. 2017], further advanced performance and stability, becoming a standard baseline for sequential decision–making problems. However, interpretability remains a major limitation. Early discussions on interpretable machine learning [Doshi-Velez e Kim 2017] have evolved into a growing body of work on explainable reinforcement learning (XRL), which highlights challenges such as opaque internal representations, high sample complexity, and sensitivity to hyperparameters [Glanois et al. 2024, Milani et al. 2024]. While post–hoc explanation methods have been proposed, they often fail to provide intrinsic interpretability, as explanations are generated externally rather than emerging from the decision–making process.

Neuro–Symbolic and Interpretable Reinforcement Learning Several works attempt to connect to symbolic and sub–symbolic paradigms. Neuro–symbolic systems combine learning and logical reasoning [d'Avila Garcez et al. 2009], while approaches such as programmatic policy distillation aim to extract interpretable representations from trained neural policies [Verma et al. 2018]. More recent approaches introduce prototypes and concepts into reinforcement learning to improve interpretability. Prototype–based neural models incorporate human–understandable reference points into decision–making [Kenny et al. 2023], and concept–based policy distillation translates neural policies into semantically meaningful structures [Bekkemoen e Langseth 2025]. These approaches improve transparency, but neural networks still remain the core decision mechanism.

Prototype Theory and Conceptual Spaces Prototype theory, introduced by Rosch [Rosch 1975], provides a cognitively grounded view of categorization based on similarity to representative examples. Gärdenfors formalized these ideas through conceptual spaces [Gärdenfors 2000], enabling geometric representations of concepts in continuous domains. These ideas have influenced research in explainable AI and semantic reasoning [Douven et al. 2023]. In reinforcement learning, state abstraction and clustering methods [Sun et al. 2025] share similarities with prototype–based representations but typically lack semantic grounding. In parallel, multi–armed bandit algorithms [Auer et al. 2002] provide interpretable mechanisms for action selection based on incremental reward feedback.

Recent works such as ProtoBandit [Chaudhuri et al. 2023] and prototype–based continual reinforcement learning [Proietti et al. 2025] demonstrate that prototypes can serve not only as explanatory tools but also as functional components for decision–making

and memory organization.

Game AI and Explainability Recent work in Game AI emphasizes explainability and designer–centric tools [Bazzaz e Cooper 2024, Bakkes et al. 2012], highlighting the importance of transparency, controllability, and behavioral consistency in interactive systems. These requirements are particularly relevant in games, where AI behavior must remain understandable and tunable by designers.

Despite significant progress, there remains a lack of systematic comparisons between symbolic, sub-symbolic, and prototype-based approaches under identical experimental conditions, particularly in real–time game environments. No prior work provides a fully interpretable, non-neural, prototype-based controller evaluated against both symbolic and DRL baselines under controlled experimental conditions.

Most existing works either focus on improving deep reinforcement learning interpretability or augmenting symbolic systems with learning capabilities. Few studies investigate whether prototype–based models can act as a true middle ground, combining adaptivity with intrinsic interpretability.

3. Fundamentals

Cognitive decision–making models differ from purely symbolic and sub–symbolic approaches by relying on explicit internal representations that mediate perception, memory, and action. A cognitive agent is formalized as $\mathcal{A} = \langle \mathcal{P}, \mathcal{C}, \mathcal{M}, \mathcal{D} \rangle$, where $\mathcal{P}$ encodes perception, $\mathcal{C}$ represents concepts, $\mathcal{M}$ stores memory, and $\mathcal{D}$ governs decision–making. This formulation aligns with intrinsically interpretable reinforcement learning, in which the reasoning process itself remains inspectable [Anderson 1990, Newell 1994, Glanois et al. 2024, Milani et al. 2024].

Prototype Theory replaces rigid category definitions with graded membership around representative instances called prototypes [Rosch 1975]. Let the current observation be encoded as a feature vector $x_t \in \mathbb{R}^n$, and let the controller maintain a set of prototypes $P = \{\mu_1, \mu_2, \ldots, \mu_K\}$, where each $\mu_k \in \mathbb{R}^n$ denotes a recurring situation. Similarity is computed through the weighted Euclidean distance $d(x_t, \mu_k) = \sqrt{\sum_{j=1}^{n} w_j (x_{t,j} - \mu_{k,j})^2}$, where $w_j > 0$ controls the relevance of each feature. The most similar prototype is selected by $k^* = \arg\min_k d(x_t, \mu_k)$.

As an illustrative example, consider a castle guard NPC in a stealth game. The state vector may encode enemy distance, visibility, health, nearby allies, and perceived noise. Different prototypes can represent contexts such as *intruder detected* (attack), *suspicious noise* (investigate), *wounded and isolated* (retreat), or *safe patrol* (continue patrol). The selected action is then determined by a local policy $a_t = \pi(k^*)$.

In the Asteroids game, the same principle is applied to navigation and combat decisions. Directional sensors estimate the proximity of nearby asteroids, while internal variables encode movement speed and weapon cooldown. These signals are transformed into a structured vector whose dimensions describe frontal danger, lateral threats, global risk, environmental openness, heading alignment with safe escape directions, speed, and cooldown. Prototypes therefore correspond to tactical situations such as imminent collision, open attack opportunities, escape corridors, or crowded danger zones.

The Conceptual Spaces framework models concepts as regions in a geometric space whose dimensions are meaningful and directly interpretable [Gärdenfors 2000]. Learning therefore corresponds to adapting explicit structures rather than opaque parameters, enabling visualization, inspection, and designer control [Bekkemoen e Langseth 2025].

Prototype–based decision–making offers a middle ground between rule–based controllers and deep reinforcement learning. Instead of directly mapping states to actions, the agent compares observations to representative situations and exploits the historical effectiveness of decisions within those regions. A key distinction is introduced between perceptual similarity and decision competence. A prototype may be close to the current state while still being historically ineffective. To capture this, each prototype stores a competence estimate V_k, updated incrementally as explained in section 4.1. This separation prevents geometric proximity from being mistaken for behavioral quality and reflects a central principle of interpretable reinforcement learning [Glanois et al. 2024].

Action selection within each prototype is modeled as a local multi–armed bandit problem [Auer et al. 2002]. Each prototype maintains action–value estimates $Q_{k,a}$ and chooses actions using an Upper Confidence Bound (UCB) strategy.

4. Proposed Approach

This section presents the proposed cognitive decision–making architecture at an abstract level. The proposed model combines semantic perception, prototype–based conceptual memory, competence–aware prototype selection, and local bandit optimization.

4.1. Architectural Overview

The architecture is organized into four layers: **Perceptual Encoding Layer > Conceptual Memory Layer > Competence and Value Layer > Action Selection Layer**. Each layer is explicitly represented and independently inspectable, ensuring auditability of the full decision process.

Perceptual Encoding At each decision step t, the environment produces a raw observation $o_t \in \mathbb{R}^d$. This observation is mapped into a semantic feature vector $x_t = \phi(o_t) \in \mathbb{R}^n$, where ϕ is a domain–dependent feature map. The purpose of ϕ is not merely dimensionality reduction, but semantic structuring: the coordinates of x_t are chosen to correspond to interpretable properties of the current situation.

Prototype-Based Conceptual Memory The agent maintains a set of prototypes $\mathcal{P} = \{\mu_1, \mu_2, \ldots, \mu_K\}, \quad \mu_k \in \mathbb{R}^n$, where each prototype represents a recurring semantic situation in feature space. Each prototype k stores $(\mu_k, V_k, Q_{k,a}, N_{k,a})$, where μ_k is the prototype centroid; V_k is the competence estimate associated with the prototype; $Q_{k,a}$ is the local action–value estimate; $N_{k,a}$ is the number of times action a has been selected under prototype k. Given the current feature vector x_t, distances to all prototypes are computed as $d_k = d(x_t, \mu_k)$. To avoid rigid winner–takes–all assignments, we define a soft responsibility function $\rho_k(x_t) = \frac{\exp(-d_k/\tau)}{\sum_{j=1}^{K} \exp(-d_j/\tau)}$, where $\tau > 0$ is a temperature parameter controlling assignment sharpness [Vamshi e Yang 2026]. If the current situation is sufficiently dissimilar from all existing prototypes, a new prototype is created $\min_k d_k > \theta_{\text{create}} \implies \mu_{K+1} \leftarrow x_t$. Otherwise, existing prototypes are updated

proportionally to their responsibilities $\mu_k \leftarrow \mu_k + \alpha\rho_k(x_t)(x_t - \mu_k)$, where α is a prototype adaptation rate.

Competence–Aware Prototype Selection A central design choice of the proposed method is the separation between *similarity* and *competence*. A prototype may be geometrically close to the current situation yet historically ineffective in selecting good actions. To model this distinction, each prototype is assigned a utility $U_k(x_t) = \rho_k(x_t)V_k - \lambda d_k$, where V_k is the competence estimate and $\lambda \geq 0$ controls the similarity–performance trade–off. The selected prototype is $k^* = \arg\max_k U_k(x_t)$.

Local Bandit–Based Action Selection Once prototype k^* is selected, action selection is performed locally using an Upper Confidence Bound (UCB) policy: $a_t = \arg\max_a \left[Q_{k^*,a} + c\sqrt{\frac{\ln(T_{k^*}+1)}{N_{k^*,a}+1}} \right]$ where $T_{k^*} = \sum_a N_{k^*,a}$, and $c > 0$ controls the exploration–exploitation balance. After executing a_t and observing reward r_t, the competence estimate is updated as $V_k \leftarrow V_k + \beta\rho_k(x_t)(r_t - V_k)$ where β is a learning rate. For the selected prototype–action pair (k^*, a_t), the bandit statistics are updated as $N_{k^*,a_t} \leftarrow N_{k^*,a_t} + 1$ and $Q_{k^*,a_t} \leftarrow Q_{k^*,a_t} + \eta(r_t - Q_{k^*,a_t})$, where η is the action–value learning rate.

At this level of abstraction, the proposed method can be understood as a prototype–based cognitive controller in which semantic perception, conceptual memory, competence estimation, and local action optimization are explicitly separated. Unlike neural policies, the decision process is not distributed over opaque parameters, but organized around human–inspectable prototypes and local decision statistics.

5. Controllers and Experimental Evaluation

This section presents the three controllers used in the experimental evaluation: a symbolic finite–state machine (FSM), a reinforcement learning baseline based on PPO, and the proposed prototype–bandit controller.

FSM Baseline for Asteroids The FSM controller is defined as a perception–categorization–control system operating in discrete time. At each time step t, the agent receives an observation $o_t \in \mathbb{R}^d$ and produces an action $a_t \in \mathcal{A}$, where the action space includes movement and shooting primitives. The environment evolves according to the $(o_{t+1}, r_{t+1}, \cdot) = \mathrm{STEP}(a_t)$ where o_{t+1} is the next observation and r_{t+1} is the reward.

Perception is based on a set of normalized ray distances $x_t \in [0,1]^N$, where lower values indicate nearby obstacles, together with a cooldown variable c_t. Two geometric operators summarize the perceptual state: the nearest obstacle $d_{\min}(t) = \min_i x_t^{(i)}$ and $i_{\min}(t) = \arg\min_i x_t^{(i)}$, and the most open direction $d_{\max}(t) = \max_i x_t^{(i)}$ and $i_{\max}(t) = \arg\max_i x_t^{(i)}$. Based on distance thresholds, the environment is categorized into danger, caution, and safe regions $Danger(t) \Leftrightarrow d_{\min}(t) < \delta_{\text{danger}}$ and $Caution(t) \Leftrightarrow d_{\min}(t) < \delta_{\text{caution}}$.

The controller determines a desired turning direction from the most open direction, while maintaining a short–term commitment (η_t, κ_t) to avoid oscillatory behavior. The internal state $s_t \in \{\text{CRUISE}, \text{AVOID}, \text{TURN\_TO\_GAP}\}$ is updated according to the current threat level: It transitions to *AVOID* under $\mathrm{Danger}(t)$, to *TURN_TO_GAP* under $\mathrm{Caution}(t)$, and remains in *CRUISE* otherwise.

The resulting policy is deterministic with finite memory $a_t = \pi(o_t; m_t)$, where $m_t = (s_t, \eta_t, \kappa_t)$, s_t denotes the current behavioral mode, $\eta_t \in \{-1, +1\}$ encodes the committed turning direction, and κ_t represents the remaining duration of this commitment.

PPO Baseline The PPO controller models the problem as a Markov Decision Process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$ and learns a stochastic policy $\pi_\theta(a \mid o)$ parameterized by a neural network $\pi_\theta(a \mid o) = \mathrm{Softmax}(f_\theta(o))$. A value function $V_\phi(o)$ estimates expected return $V_\phi(o) \approx \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k}\right]$ where $\gamma \in [0, 1)$ is the discount factor, controlling the relative importance of future rewards. Higher values of γ encourage long-term planning, while lower values bias the agent toward immediate rewards. Learning is driven by advantage estimation using temporal-difference residuals $\delta_t = r_t + \gamma V_\phi(o_{t+1}) - V_\phi(o_t)$ and $A_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}$ where $\lambda \in [0, 1]$ is the GAE parameter controlling the bias–variance trade-off in advantage estimation. Lower values favor low-variance, short–horizon estimates, while higher values incorporate longer-term returns at the cost of increased variance. The policy is optimized using the clipped PPO objective $\mathcal{L}_{\text{PPO}} = \mathbb{E}\left[\min\left(r_t(\theta)A_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\right]$, with $r_t(\theta) = \frac{\pi_\theta(a_t|o_t)}{\pi_{\theta_{\text{old}}}(a_t|o_t)}$, where $\epsilon > 0$ is a clipping hyperparameter that constrains policy updates by limiting the probability ratio to the interval $[1 - \epsilon,\ 1 + \epsilon]$. The final loss combines policy optimization, value regression, and entropy regularization. While PPO provides strong performance, it lacks explicit internal structure, making interpretation difficult.

Prototype-Bandit Controller The proposed controller operates over a semantic representation $x_t = \phi(o_t) \in \mathbb{R}^{10}$ derived from raw observations. It compresses high-dimensional sensory input into a structured semantic space composed of ten interpretable features. These features are designed to capture directional threat, global spatial configuration, and agent dynamics, enabling efficient and interpretable decision-making.

Let t_i denote the threat magnitude associated with ray i, and d_i the corresponding normalized distance. Distances are converted into threat magnitudes $t_i = 1 - d_i$, from which global and directional summaries are extracted. We define the maximum threat $T_g = \max_i t_i$, which captures the most critical obstacle in the environment. The average threat level is given by $\bar{T} = \frac{1}{R}\sum_i t_i$, where R is the number of raycasts (sensors) used to perceive the environment, providing a measure of overall environmental risk. Finally, the openness of the environment is quantified by the mean distance $O = \frac{1}{R}\sum_i d_i$, which reflects the average amount of free space surrounding the agent.

A repulsive direction is computed from obstacle distribution and aligned with the agent heading $\tilde{r} = -\sum_{i=1}^{R} t_i(\cos\alpha_i, \sin\alpha_i)$ and $r = \frac{\tilde{r}}{\|\tilde{r}\|+\varepsilon}$. The vector $\tilde{r}$ represents the aggregated repulsive force induced by nearby obstacles, and r is its normalized version. Let $h = (\cos\theta, \sin\theta)$ denote the agent heading direction. The alignment between the heading and the repulsive vector r is defined as $A = \langle h, r\rangle \in [-1, 1]$, which measures how well the agent is oriented away from nearby obstacles.

These features define the semantic state $x_t = [T_f, T_l, T_r, T_b, T_g, \bar{T}, O, A, S, C]$. Decision-making is based on a set of prototypes $\{\mu_k\}$ in this space, using a weighted distance $d_k = \sqrt{\sum_j w_j(x_{t,j} - \mu_{k,j})^2}$. At each decision step, the controller evaluates the distance d_k for all prototypes and combines similarity with competence through $U_k(x_t) = \rho_k(x_t)V_k - \lambda d_k$, where $\rho_k(x_t)$ is the soft responsibility of prototype k, V_k is its learned

Table 1. Performance comparison across controllers (mean $\pm$ std).

Controller	Return	Score	Steps (Survival)
FSM	27.63 ± 37.25	7.58 ± 6.64	230.29 ± 117.11
PPO	60.38 ± 74.61	10.05 ± 6.74	$\mathbf{331.89 \pm 96.47}$
Prototype-Bandit	$\mathbf{66.68 \pm 70.95}$	$\mathbf{12.91 \pm 10.37}$	277.97 ± 111.39

competence estimate, and λ controls the similarity–performance trade-off. The selected prototype is $k^* = \arg\max_k U_k(x_t)$.

This yields a decision-making system that combines structured perception, interpretable representations, and adaptive decision-making. Unlike PPO, all internal elements (such as prototypes, distances, responsibilities, and action values) remain explicit, enabling inspection and auditability of the decision process.

5.1. Experiments and Results

The experiments provide a comparative evaluation of the proposed prototype-based architecture against symbolic and sub-symbolic baselines, focusing on performance, stability, behavioral consistency, and interpretability. All experiments were conducted on an Intel i7-7500U CPU running at 2.70 GHz with 16 GB of RAM, without GPU acceleration, ensuring a fair comparison across all controllers. Both the PPO[1] and prototype-based agents were trained for $100{,}000$ interaction episodes, while the FSM baseline required no training. Each controller was evaluated over 30 episodes for each of five independent random seeds $\{123, 321, 345, 543, 567\}$, resulting in a total of 150 evaluation episodes per controller. All controllers were tested under identical environment conditions, reward functions, and termination criteria. The Asteroids simulation environment was developed using Godot 4.5.1 and integrated with controllers implemented in Python 3.12.10. Both the FSM and the prototype-bandit controllers were implemented in pure Python, while the PPO agent was developed using the Gymnasium v1.2.3 interface together with Stable-Baselines3 v2.8.0.

Table 1 reports the results in terms of average return, score, and survival time. The prototype-based controller demonstrates overall performance in the Asteroids environment, indicating that effective decision strategies can be learned without neural function approximation. These results should not be interpreted as evidence that the proposed prototype-bandit controller performs better than PPO. Rather, they suggest that, in geometrically structured and reactive domains such as Asteroids, semantically grounded prototypes can provide a strong inductive bias that improves sample efficiency and decision quality. The observed standard deviation may reflect a secondary effect of the proposed update dynamics and should be investigated in future work. The PPO agent achieves the highest survival time, suggesting stronger long-term avoidance behavior, but also exhibits greater variability across episodes, as reflected by its larger standard deviation. The FSM baseline shows lower overall performance, consistent with its fixed rule-based structure, while providing a stable and fully interpretable reference.

The results in Table 2 provide additional insight into behavioral stability and consistency. PPO exhibits an almost negligible action switching rate, indicating smooth

[1]Parameters: "MlpPolicy", n_steps=1024, batch_size=256, n_epochs=10, gamma=0.99, gae_lambda=0.95, learning_rate=3e-4, clip_range=0.2, ent_coef=0.01, vf_coef=0.5

Table 2. Stability and behavioral consistency metrics.

Controller	Action Switch Rate	Danger Ratio	Safe Ratio
FSM	0.1607 ± 0.0891	0.1175 ± 0.0801	0.6900 ± 0.1284
PPO	$\mathbf{0.0001 \pm 0.0008}$	$\mathbf{0.0886 \pm 0.0730}$	$\mathbf{0.7447 \pm 0.1391}$
Prototype-Bandit	0.1645 ± 0.0809	0.1125 ± 0.1061	0.7063 ± 0.1794

Table 3. Interpretability metrics for the Prototype-Bandit controller.

Metric	Value
Number of used prototypes	5.51
Responsibility entropy	1.55

and stable action sequences. In contrast, both the FSM and the prototype-based controller display higher switching rates, reflecting more reactive behavior. Even so, the prototype-based controller maintains consistency comparable to the FSM while preserving competitive overall performance. In terms of safety-related behavior, PPO spends less time in high-risk situations, which contributes to its higher survival time. The prototype-based controller shows slightly higher exposure to such situations, while still maintaining competitive return and score.

Table 3 presents interpretability-related metrics for the proposed approach. The controller relied on a compact set of prototypes to represent the decision space, suggesting that a relatively small number of semantic regions is sufficient to capture relevant behavioral patterns in this environment. The observed responsibility entropy indicates that decisions are distributed across multiple prototypes, reflecting gradual transitions between semantic contexts rather than rigid categorization.

In contrast to PPO, whose internal decision process is not directly accessible, the prototype-based controller exposes all internal elements of its decision-making process. Each prototype corresponds to a region of the semantic state space, and associated action-value estimates provide insight into the effectiveness of actions within those regions. This explicit representation enables inspection, debugging, and potential manual adjustment of agent behavior, which is particularly relevant in game development contexts.

Overall, the results highlight a trade-off between performance, stability, and interpretability. PPO achieves strong performance in terms of survival time and smooth behavior, but lacks transparency. The FSM baseline provides full interpretability and stability but is limited in adaptability. The proposed prototype-based controller occupies an intermediate position, combining competitive performance in the evaluated setting with full interpretability through explicit semantic and decision structures.

These findings are consistent with the hypothesis that prototype-based decision-making can serve as a practical middle ground between symbolic and sub-symbolic approaches. However, it is important to note that the evaluation is limited to the Asteroids environment, and further experiments in more complex or high-dimensional domains are required to assess the generality of the approach. No statistical significance tests were conducted, and this remains an important direction for future work.

6. Conclusions

This work presented a prototype-based cognitive architecture for decision-making in real-time game environments, aiming to combine interpretability with adaptive behavior. The proposed approach integrates semantic perception, prototype-based conceptual memory, and local reinforcement through bandit optimization, while maintaining explicit and inspectable internal structures.

The empirical evaluation, conducted in the Asteroids game under controlled conditions, indicates that the proposed method can achieve results comparable to the evaluated baselines. In particular, the prototype-based controller achieved a competitive average return and score in this setting, while PPO demonstrated advantages in terms of survival time and smoother behavior. These results highlight different strengths across approaches and suggest that prototype-based decision-making offers a practical balance between adaptability and interpretability.

The comparisons presented in this work are limited to a single game environment with a specific state representation and action space. As such, the observed performance differences should be interpreted within this context and do not necessarily generalize to other domains or more complex scenarios. Further evaluation across diverse environments is required to better understand the strengths and limitations of the approach.

From a practical perspective, the proposed method offers advantages in terms of transparency and controllability, as all internal components (such as prototypes, responsibilities, and action-value estimates) are explicitly represented and can be directly inspected. This property may be particularly relevant in game development contexts where understanding and tuning agent behavior is important.

The results suggest that prototype theory based decision-making represents a relevant direction for interpretable Game AI. Future work could extend the approach to more complex environments, explore richer semantic representations, investigate hybrid models that combine prototype reasoning with other learning paradigms, and conduct broader statistical validation to better characterize the significance, robustness, and reproducibility of the observed results.

References

Anderson, J. R. (1990). *The Adaptive Character of Thought*. Lawrence Erlbaum.

Auer, P., Cesa-Bianchi, N., e Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256.

Bakkes, S. C., Spronck, P. H., e van Lankveld, G. (2012). Player behavioural modelling for video games. *Entertainment Computing*, 3(3):71–79. Games and AI.

Bazzaz, M. e Cooper, S. (2024). Guided game level repair via explainable ai. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 20, pages 139–148.

Bekkemoen, Y. e Langseth, H. (2025). Interpretable Deep Reinforcement Learning Via Concept-Based Policy Distillation. *Machine Learning*, 114(12):288.

Cai, M., Wei, G., e Cao, J. (2019). Categories in emergency decision-making: prototype-based classification. *Kybernetes*, 49(2):526–553. _eprint: https://www.emerald.com/k/article-pdf/49/2/526/1822783/k-08-2018-0454.pdf.

Chaudhuri, A. R., Jawanpuria, P., e Mishra, B. (2023). ProtoBandit: Efficient prototype selection via multi-armed bandits. In Khan, E. e Gonen, M., editors, *Proceedings of The 14th Asian Conference on Machine Learning*, volume 189 of *Proceedings of Machine Learning Research*, pages 169–184. PMLR.

Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., e Su, J. K. (2019). This looks like that: deep learning for interpretable image recognition. volume 32.

Doshi-Velez, F. e Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Douven, I., Verheyen, S., Elqayam, S., Gärdenfors, P., e Osta-Vélez, M. (2023). Similarity-based reasoning in conceptual spaces. *Frontiers in Psychology*, 14:1234483.

d'Avila Garcez, A. S., Lamb, L. C., e Gabbay, D. M. (2009). *Neural-symbolic cognitive reasoning*. Springer.

Gärdenfors, P. (2000). *Conceptual spaces: The geometry of thought. A Bradford book*, volume 3.

Glanois, C., Weng, P., Zimmer, M., Li, D., Yang, T., Hao, J., e Liu, W. (2024). A survey on interpretable reinforcement learning. *Machine Learning*, 113(8):5847–5890.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., e Silver, D. (2018). Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Kenny, E. M., Tucker, M., e Shah, J. (2023). Towards interpretable deep reinforcement learning with human-friendly prototypes. In *The Eleventh International Conference on Learning Representations*.

Li, P., Siddique, U., e Cao, Y. (2025). From explainability to interpretability: Interpretable reinforcement learning via model explanations. In *Reinforcement Learning Conference*.

Milani, S., Topin, N., Veloso, M., e Fang, F. (2024). Explainable reinforcement learning: A survey and comparative review. *ACM Computing Surveys*, 56(7):1–36.

Millington, I. e Funge, J. (2009). *Artificial Intelligence for Games*. Morgan Kaufmann.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., e Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518:529–533.

Newell, A. (1994). *Unified theories of cognition*. Harvard University Press.

Proietti, M., Wurman, P. R., Stone, P., e Capobianco, R. (2025). Protocrl: Prototype-based network for continual reinforcement learning.

Rosch, E. (1975). Cognitive representations of semantic categories. *Journal of experimental psychology: General*, 104(3):192.

Samuel, B., Treanor, M., e McCoy, J. (2021). Design considerations for creating ai-based gameplay. In *AIIDE Workshops*.

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

Sun, S., Qi, M., e Shen, Z.-J. M. (2025). Online mdp with prototypes information: A robust adaptive approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 20717–20724.

Tappler, M., Lopez-Miguel, I. D., Tschiatschek, S., e Bartocci, E. (2025). Rule-Guided Reinforcement Learning Policy Evaluation and Improvement. In Kwok, J., editor, *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 6254–6262. International Joint Conferences on Artificial Intelligence Organization.

Vamshi, B. K. e Yang, H. (2026). Principal prototype analysis on manifold for interpretable reinforcement learning. *arXiv preprint arXiv:2603.27971*.

Verma, A., Murali, V., Singh, R., Kohli, P., e Chaudhuri, S. (2018). Programmatically interpretable reinforcement learning. In *International conference on machine learning*, pages 5045–5054. PMLR.