Neurosymbolic Artificial Intelligence
XX(X):2–32

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

GRAIL: Autonomous Concept Grounding for Neuro-Symbolic Reinforcement Learning

Hikaru Shindo[1] and Henri Rößler[1] and Quentin Delfosse[1,2] and Kristian Kersting[1,3,4,5]

Abstract

Neuro-symbolic Reinforcement Learning (NeSy-RL) combines symbolic reasoning with gradient-based optimization to achieve interpretable and generalizable policies. Relational concepts—such as “left of” or “close by”—serve as foundational building blocks that structure how agents perceive and act. However, conventional approaches require human experts to manually define these concepts, limiting adaptability since concept semantics vary across environments.

We propose GRAIL (Grounding Relational Agents through Interactive Learning), a framework that autonomously grounds relational concepts through environmental interaction. GRAIL leverages large language models (LLMs) to provide generic concept representations as weak supervision, then refines them to capture environment-specific semantics. This approach addresses both sparse reward signals and concept misalignment prevalent in underdetermined environments.

Experiments on the Atari games *Kangaroo*, *Seaquest*, and *Skiing* demonstrate that GRAIL matches or outperforms agents with manually crafted concepts in simplified settings, and reveals informative trade-offs between reward maximization and high-level goal completion in the full environment.

Keywords

Neuro-Symbolic AI, Concept Grounding, Reinforcement Learning, Differentiable Reasoning, Large Language Models

Prepared using sagej.cls *[Version: 2017/01/17 v1.20]*

Introduction

Deep reinforcement learning (RL) has achieved remarkable progress in recent years, driving advancements in critical fields such as autonomous driving and robotics. Deep neural networks, capable of learning policies across diverse tasks without prior domain knowledge (Mnih et al. 2013; Schulman et al. 2017; Badia et al. 2020; Bhatt et al. 2024), have thus become the foundation of modern RL. Despite their success, these black-box models are prone to shortcut learning, exploiting action strategies that may be imperceptible to humans (Liu and Borisyuk 2024; Delfosse et al. 2025). For instance, in Atari Pong, deep RL agents often gravitate toward behavior that focuses on the opponent's position instead of tracking the ball (Delfosse et al. 2024b), demonstrating limited generalization when the environment is altered even slightly.

To overcome the limitations of neural approaches, RL has increasingly incorporated symbolic reasoning through logic-based policies (Jiang and Luo 2019; Kimura et al. 2021; Cao et al. 2022; Delfosse et al. 2023b) and programmatic frameworks (Sun et al. 2020; Verma et al. 2018; Lyu et al. 2019; Cappart et al. 2021; Kohler et al. 2024). These methodologies offer transparency, revisability, enhanced generalization, and facilitate curriculum learning. Nevertheless, they remain heavily dependent on human-provided inductive biases—requiring domain experts to hard-code essential concepts or logic rules—and often struggle to capture fine-grained, low-level behaviors. This reliance fundamentally constrains the flexibility and expressiveness of symbolic systems.

Research in philosophy and cognitive science has long maintained that human generalization capabilities stem from the ability to perceive the world through *concepts* (Bruner et al. 1956; Rosch 1973). Concepts represent abstract attributes or relations common across sets of entities (Archer 1966); for example, by color, shape, or positional relation to others. While concept learning has been explored in visual reasoning and planning (Mao et al. 2019; Hsu et al. 2023; Silver et al. 2023; Mao et al. 2025; Sha et al. 2025a), grounding concepts in RL tasks remains relatively uncharted. Current RL approaches bypass this by manually specifying grounding functions (Jiang and Luo 2019; Vouros 2022; Delfosse et al. 2023b; Shindo et al. 2025), a practice feasible in simple domains but impractical when facing greater complexity or relational structure involving multiple objects.

[1]TU Darmstadt, Germany
[2]Google Intrinsic, Germany
[3]Hessian AI, Germany
[4]German Research Center for Artificial Intelligence (DFKI), Germany
[5]Centre for Cognitive Science, TU Darmstadt, Germany

Corresponding author:
Hikaru Shindo, TU Darmstadt, Karolinenpl. 5, 64289 Darmstadt, Germany.

Prepared using sagej.cls

Figure 1. Concept grounding is environment-dependent in neuro-symbolic reinforcement learning. The spatial concept "left of" requires different interpretations across environments. In Kangaroo (left), the agent must verify that the player is both horizontally left of the ladder *and* vertically aligned on the same platform. In Seaquest (middle), "left of" is defined more flexibly based on horizontal positioning regardless of vertical alignment. In Skiing (right), the relation is computed relative to the midpoint between flags rather than individual flag objects. While humans intuitively adapt these conceptual meanings to context, existing neuro-symbolic RL frameworks rely on manually hard-coded valuation functions for each environment (Shindo et al. 2025), severely limiting their scalability and adaptability to novel domains.

Figure 1 illustrates this necessity across three Atari environments. Here, the agent must dynamically ground the "left of" relationship in different contexts: in Kangaroo (left), "left of ladder" entails being to the left of a ladder and on the same platform; in Seaquest (middle), "left of diver" simply means to the left of a diver irrespective of vertical alignment; in Skiing (right), "left of flags" requires identifying flags ahead and computing one's position relative to them. Currently, these varied conceptual groundings are hard-coded, which limits adaptability to new environments. This fundamental challenge raises an important research question: *How can an agent learn to ground relational concepts autonomously through environment interactions?*

To address this, we propose GRAIL (Grounding Relational Agents through Interactive Learning), a novel framework that enables agents to ground relational concepts via experience. GRAIL builds upon BlendRL (Shindo et al. 2025), which employs a hybrid policy architecture—combining symbolic logic rules and differentiable neural networks—trained jointly with differentiable forward reasoning (Evans and Grefenstette 2018; Shindo et al. 2023). In BlendRL, each symbolic predicate is associated with a differentiable function that computes truth values over state observations, allowing the overall policy to seamlessly bind symbolic and neural reasoning.

GRAIL extends this by introducing a new method for concept grounding within the BlendRL framework, allowing each concept to be learned and adapted to its specific environment. Crucially, GRAIL leverages Large Language Models (LLMs) to provide general, high-level descriptions of concepts as weak supervision signals. For example, the LLM supplies a prototypical representation of "left," which GRAIL then uses alongside environment feedback to train differentiable functions that maximize reward and align their outputs with the LLM-provided signal. This is accomplished by adding a novel

loss term to the Proximal Policy Optimization (PPO) (Schulman et al. 2017) objective, encouraging alignment between learned concept groundings and guidance from LLMs.

Our experiments in the Atari environments Kangaroo, Seaquest, and Skiing demonstrate that GRAIL matches or outperforms both neural and neuro-symbolic baselines in a simplified setting, successfully discovering task-optimal concept groundings directly from interaction. As illustrated in Figure 2, GRAIL learns fundamentally different groundings of "left" and "right" depending on the environment—horizontal platform-aligned concepts in Kangaroo versus anticipatory diagonal concepts in Skiing. In summary, our core contributions are:

- We introduce GRAIL*, a framework that enables neuro-symbolic agents to ground relational concepts through environment interaction. GRAIL extends BlendRL by learning spatial relational concepts autonomously, using LLMs to provide high-level concept representations as weak supervision. The resulting policies are highly interpretable, expressed as first-order logic rules over the learned concepts.
- We formulate *Concept Alignment* as a novel regularization term for PPO-based policy learning, encoding the degree to which the agent's learned concepts align with LLM-generated proxy functions. The resulting GRAIL learning framework navigates the inherent trade-off between reward maximization and semantically faithful concept grounding, and is among the first to discover spatial concept representations in this setting.
- We evaluate GRAIL on three challenging Atari environments: Kangaroo, Seaquest, and Skiing, which have not previously been tackled by neuro-symbolic agents without hard-coded relational concepts. We demonstrate that GRAIL matches or outperforms both the state-of-the-art neuro-symbolic baseline and a purely neural PPO baseline. Furthermore, we qualitatively analyze the relational concepts learned by GRAIL agents, showing that they acquire meaningful spatial groundings from experience without concept-level supervision.

Background

GRAIL builds on several foundational research areas, briefly reviewed in this section.

Deep Reinforcement Learning

We model the environment as a Markov decision process (MDP), $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$. The objective is to learn a policy $\pi_\theta(a_t \mid s_t)$ that maximizes the expected discounted return:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t r_t\right] \tag{1}$$

where $\gamma \in [0, 1]$ is the discount factor and T is the episode length.

*Code is available at: https://github.com/ml-research/grail

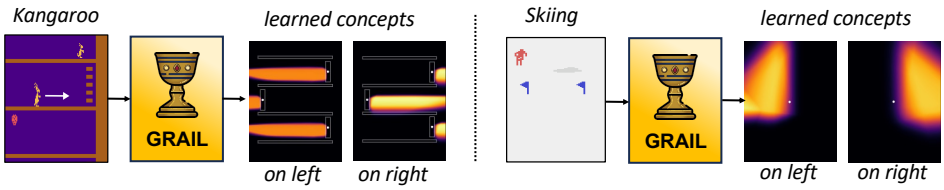

Figure 2. GRAIL learns environment-specific concept groundings. Given different environments, GRAIL autonomously discovers distinct interpretations of "left" and "right." In Kangaroo (left), the learned concepts activate along horizontal bands aligned with each platform, reflecting that "left of ladder" requires both horizontal offset and vertical alignment. In Skiing (right), activation extends diagonally above each flag, capturing the anticipatory nature of steering decisions during downhill movement. These heatmaps demonstrate that GRAIL adapts abstract relational concepts to the specific spatial structure of each environment.

Proximal Policy Optimization. GRAIL optimizes policies using Proximal Policy Optimization (PPO) (Schulman et al. 2017) actor-critic method, that maintains both a policy (actor) π_θ and a value function (critic) V_ϕ, evaluating the actor's decisions. PPO estimates the advantage of each action using Generalized Advantage Estimation (GAE):

$$\hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)} = \sum_{i=0}^{T} (\gamma\lambda)^i \delta_{t+i}^{(1)} \tag{2}$$

where $\delta_t^{(1)} = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$ is the one-step TD residual and λ controls the bias-variance trade-off. The policy is then updated by maximizing the following clipped surrogate objective:

$$L^{\mathrm{CLIP}}(\theta) = \mathbb{E}_{(s,a)\sim\pi_{\theta_{\mathrm{old}}}} \left[\min\left(r_t(\theta)\hat{A}_t, \mathrm{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t \right) \right] \tag{3}$$

where $r_t(\theta) = \pi_\theta(a_t \mid s_t)/\pi_{\theta_{\mathrm{old}}}(a_t \mid s_t)$ is the probability ratio between the current and previous policy, and ϵ is a clipping coefficient that constrains updates to a trust region, enabling stable reuse of experience data across multiple gradient steps.

Neuro-Symbolic Reinforcement Learning

GRAIL specifically builds upon the ideas of neuro-symbolic reasoning and learning with first-order logic.

First-Order Logic. GRAIL uses *First-Order Logic* (FOL) to encode world knowledge and actions in a logical and structured manner. A language in FOL, $\mathcal{L} = (\mathcal{P}, \mathcal{F}, \mathcal{D}, \mathcal{V})$, comprises predicate symbols $\mathcal{P}$, functors $\mathcal{F}$, constants $\mathcal{D}$ and variables $\mathcal{V}$.

An *atom* $\mathtt{p}(\mathtt{t}_1, \ldots, \mathtt{t}_n)$ is the smallest unit in a logical statement, where $\mathtt{t}_1, \ldots, \mathtt{t}_n$ are terms and $\mathtt{p}$ is a predicate of arity $\alpha(\mathtt{p}) = n$. Ground atoms (with only constant terms) have truth values. A *Horn clause* takes the form A :- $B_1, \ldots, B_n$, where A is the *head* and $\{B_1, \ldots, B_n\}$ is the *body*, meaning if all body atoms are true, then A must hold.

Logic for Actions. GRAIL adopts first-order logic as the core language for representing both actions and states, enabling explicit reasoning throughout the agent's learning process with the logic programming framework (Lloyd 1984). This perspective traces back to foundational work on logical reasoning about actions (Reiter 2001); GRAIL follows and extends recent neuro-symbolic efforts such as (Delfosse et al. 2023b) by structuring policies with weighted first-order logic rules.

The predicate set $\mathcal{P}$ is split into *action predicates* ($\mathcal{P}_A$) and *state predicates* ($\mathcal{P}_S$). This separation empowers the agent to distinguish what it can *do* from what it can *know* about the world. The resulting *Action-State Language* is defined by $(\mathcal{P}_A, \mathcal{P}_S, \mathcal{D}, \mathcal{V})$. For illustration, consider the *Kangaroo* environment (Figure 1), where action predicates may include $\mathcal{P}_A = \{\texttt{go\_left}, \texttt{go\_right}, \texttt{jump}, \texttt{idle}\}$, while state predicates could be $\mathcal{P}_S = \{\texttt{left\_of}, \texttt{closeby}, \ldots\}$. An *action rule* takes the form $X_A \texttt{:-} X_S^{(1)}, \ldots, X_S^{(n)}$—the action is taken when all body conditions hold. For example, "move right if left of a ladder":

```
go_right(O1):-type(O1,agent),type(O2,ladder),left_of(O1,O2).
```

Differentiable Reasoning for RL. GRAIL is built upon *differentiable logic programming* (Evans and Grefenstette 2018; Shindo et al. 2021, 2023), in which logical reasoning is realized through differentiable tensor operations, enabling end-to-end gradient-based optimization of symbolic representations.

Figure 3 illustrates the computational flow. Raw states are first transformed into object-centric representations, where each object is described by its attributes (*e.g.*, type, x- and y-coordinates). These representations are then processed by *valuation functions*—differentiable parameterized functions that estimate the confidence of each state atom. For example, $v_\psi^{\texttt{left\_of}}$ computes a soft confidence score for the `left_of` predicate given a pair of objects. The outputs of these valuation functions form a weighted set of ground atoms that feed into the symbolic policy reasoning.

While valuation functions have traditionally been hand-crafted, limiting applicability and scalability, GRAIL provides a unified framework to *learn* them directly from environment interaction while encouraging alignment with semantically meaningful concepts. Because the entire reasoning pipeline is differentiable, GRAIL can optimize concept representations end-to-end via policy gradient methods, jointly improving task performance and concept quality.

Related Work

GRAIL is built upon neuro-symbolic reinforcement learning, concept learning, and object-centric representation learning. We review each area and position our contributions accordingly.

Neuro-Symbolic RL

Relational Reinforcement Learning (Relational RL) (Dzeroski et al. 2001; Kersting et al. 2004; Kersting and Driessens 2008; Lang et al. 2012; Hazra and

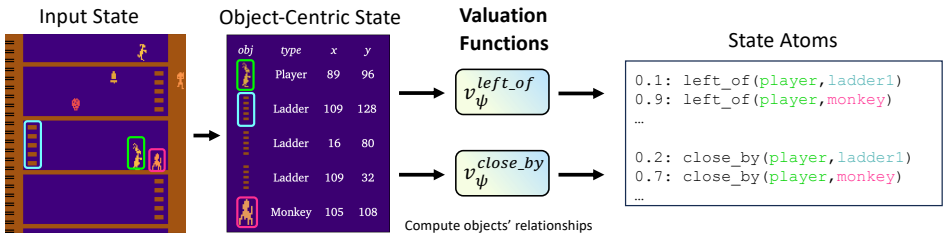

Figure 3. Valuation functions evaluate relationships between objects. These functions are typically hard-coded, limiting the applicability of neuro-symbolic reinforcement learning. GRAIL aims to learn these functions by aligning them with the correct concept directly from interactions with the environment.

Raedt 2023; Acharya et al. 2023; Golivand Darvishvand et al. 2025) leverages logical representations and probabilistic reasoning to address RL challenges in structured, relational domains. The Neural Logic Reinforcement Learning (NLRL) framework (Jiang and Luo 2019) is a pioneering effort to introduce Differentiable Inductive Logic Programming (∂ILP) (Evans and Grefenstette 2018) into RL. Here, ∂ILP facilitates the learning of generalized logic rules from examples using gradient-based optimization. NUDGE (Delfosse et al. 2023b) builds further by incorporating neurally-guided symbolic abstraction, drawing on significant progress in differentiable logic programming (Shindo et al. 2023, 2024b) to learn more complex programs. BlendRL (Shindo et al. 2025) subsequently extends these ideas, combining symbolic and neural policies within a unified framework.

While these approaches demonstrate the effectiveness of learning logic-based policies, they share a common limitation: the reliance on *manually* specified relational predicates, including explicit definitions of their semantics to compute confidence scores. As a result, adapting such methods to novel environments typically requires considerable manual effort to define suitable predicates and their underlying grounding functions. In contrast, GRAIL overcomes this bottleneck by automatically learning the *grounding* of relational predicates—i.e., the valuation functions that define their semantics—directly from environment interaction, thus substantially broadening the applicability of neuro-symbolic RL.

Concept Learning

Learning *concepts* is a fundamental challenge in artificial intelligence and machine learning. Modeling concepts explicitly in the machine learning pipeline enhances interpretability and generalization of data-driven models (Koh et al. 2020; Espinosa Zarlenga et al. 2022; Stammer et al. 2021; Steinmann et al. 2025). Neuro-symbolic methods address this challenge by learning concepts from experience with symbolic programs (Mao et al. 2025), with an emphasis on complex visual reasoning with multiple objects and relations (Mao et al. 2019; Hsu et al. 2023; Shindo et al. 2024a; Helff et al. 2023; Sha et al. 2025b; Wüst et al. 2026).

However, these works focus on concept understanding in perception tasks or question answering; concept grounding in RL settings—*i.e.*, learning *what* relational predicates mean through environment interaction—remains underexplored. GRAIL learns to ground relational concepts through interaction, guided by weak supervision from LLMs.

GRAIL draws on classical formalisms for describing concepts and actions abstractly. Allen's interval algebra (Allen 1983) provides a qualitative calculus over temporal intervals; the spatial predicates learned by GRAIL can be viewed as a continuous, learned counterpart of such qualitative relations. Furthermore, action languages grounded in first-order and second-order logic have long been used to specify actions in planning and reinforcement learning (Reiter 2001). GRAIL follows the same tradition, representing policies as logic rules whose head atoms correspond to the actions executed by the agent.

Object-Centric RL

Object-centric decomposition is a fundamental pillar for achieving task generalization in reinforcement learning (Delfosse et al. 2025). Object-centric reinforcement learning agents first need to transform unstructured state representations by decomposing visual inputs into object-centric states (Locatello et al. 2020; Lin et al. 2020; Kipf et al. 2022; Delfosse et al. 2023c). These structured representations are increasingly integrated into RL pipelines, improving compositional generalization in model-free policies (Haramati et al. 2024; Chen et al. 2024; Grandien et al. 2024) and enabling complex relational reasoning via object-level latent dynamics in model-based architectures (Mosbach et al. 2025; Dillies et al. 2025; Nishimoto and Matsubara 2026; Blüml et al. 2025; Feng et al. 2025). In the Atari domain, extracting such entity-level ground truth from raw pixels or RAM remains a fundamental challenge (Delfosse et al. 2023a; Luo et al. 2024). To bridge this gap, recent approaches leverage pre-trained visual segmentations to construct sample-efficient spatial-temporal world models directly within these complex arcade environments (Zhang et al. 2025; Blüml et al. 2025).

GRAIL: Learning to Ground Relational Concepts

GRAIL extends BlendRL (Shindo et al. 2025) by replacing its hand-crafted valuation functions with *learnable* differentiable grounding functions, guided by LLM-generated proxy concepts as weak supervision. In the following, we outline the limitations of BlendRL's manual grounding and describe our proposed advances.

An overview of our approach is provided in Figure 4. The BlendRL framework trains neuro-symbolic policies to maximize expected reward using Proximal Policy Optimization (PPO) (Schulman et al. 2017). In this setup, the logic policy is represented as a set of weighted rules over predicates, with each predicate associated with a differentiable valuation function to capture abstract state relations.

A central challenge arises in learning these spatial relations: aligning each predicate with its intended concept—the well-known *symbol grounding problem*—is nontrivial. As a result, naively implementing predicates as neural networks and optimizing solely for reward frequently leads to poor or uninterpretable alignments.

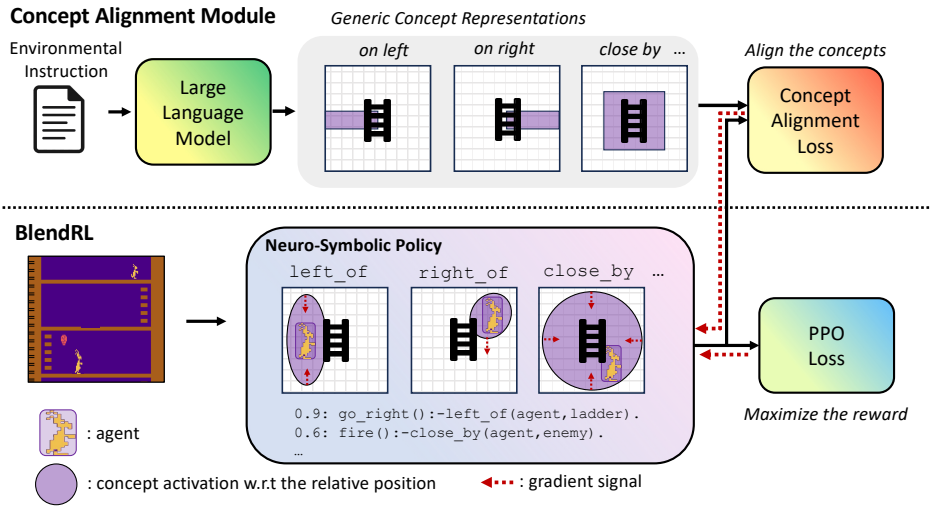

Figure 4. GRAIL: Framework Overview. GRAIL extends the BlendRL framework (Shindo et al. 2025), uniting neural and logic-based policies within a neuro-symbolic RL agent. Here, the logic policy consists of weighted rules expressed over predicates, where each predicate is equipped with a differentiable valuation function capturing abstract state relations. Unlike BlendRL, which relies on hand-crafted predicate valuations, GRAIL introduces *concept alignment*: it leverages large language models (LLMs) to extract generic concept representations and incorporates a dedicated loss term that encourages the learned valuation functions to match these LLM-derived proxies. In this depiction, the spatial relations between agents and objects are visualized—purple areas indicate the normalized outputs of the corresponding valuation functions, *i.e.*, the concept activations with respect to the relative positions of objects.

To address both reward maximization and robust concept alignment, we introduce a concept grounding mechanism, depicted at the top of Figure 4. This module leverages a large language model (LLM) to extract general representations of relevant concepts, informed by environmental instructions that succinctly describe the task and domain. In essence, the LLM generates proxy representations for each concept (for example, specifying what "left" should look like in general). GRAIL then grounds these general, LLM-derived representations to the specifics of a given environment (such as Kangaroo), refining them through interaction and reward maximization. While a generic "left" representation may not initially yield high performance, GRAIL improves this by learning to adapt and refine concept valuations according to environmental feedback. This allows the agent to achieve both generalization across tasks and strong environment-specific performance. Crucially, this introduces an inherent trade-off: too strong an alignment signal constrains the agent to the LLM's generic priors and can impede reward maximization, while too weak a signal leaves the agent susceptible to degenerate or semantically meaningless groundings. We address this tension through an annealing schedule and a tunable alignment coefficient.

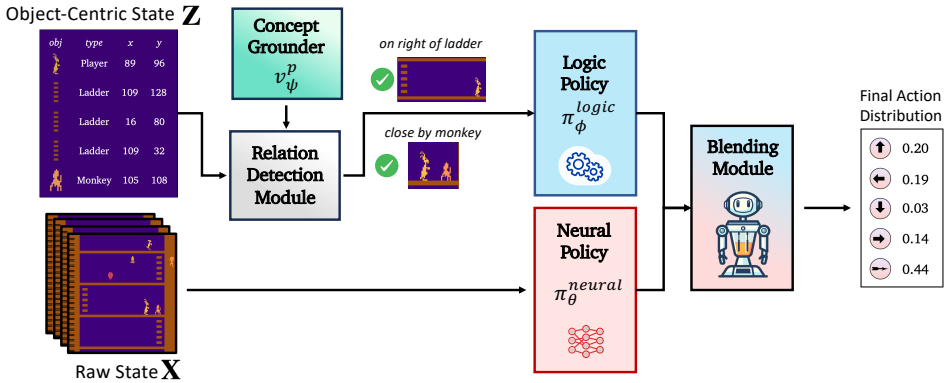

Figure 5. GRAIL's Policy Reasoning. A *concept grounding module* takes object-centric features $\mathbf{Z}$ extracted from an image $\mathbf{X}$ and computes object relations via differentiable valuation functions v_ψ^{p}. Those are then applied to a set of logical rules through forward reasoning to determine a *logic policy*. Likewise, a *blending module* utilizes the relational information to combine the logic policy with a *neural policy* that operates on the sub-symbolic state. All components can be trained jointly.

The Hybrid Policy Reasoning and Learning

GRAIL inherits the hybrid policy architecture from BlendRL (Shindo et al. 2025), which combines neural and symbolic policies trained jointly. We summarize this inherited architecture below for completeness. The input state is represented by both a pixel-based and a symbolic representation, and the policy reasoning is depicted in Figure 5.

Hybrid State Representations. GRAIL agents utilize two complementary forms of state representation: (i) *pixel-based representations*, and (ii) *object-centric representations*. The former comprise stacks of raw images directly provided by the environment and typically processed via convolutional neural networks (Mnih et al. 2015). The latter are extracted using object discovery models (Redmon et al. 2016; Lin et al. 2020; Delfosse et al. 2023c; Zhao et al. 2023) and consist of structured lists of objects with associated attributes (*e.g.*, position, orientation, color), enabling explicit logical reasoning (Zadaianchuk et al. 2021; Liu et al. 2021; Yoon et al. 2023; Wüst et al. 2024; Stammer et al. 2024). Alternatively, these states can be systematically extracted if supported by the environment. In the case of Atari, OCAtari (Delfosse et al. 2023a) accomplishes this by reading the internal RAM state to produce structured object data.

Formally, the raw (sub-symbolic) state is denoted as $\mathbf{X} \in \mathbb{R}^{F \times W \times H \times C}$, representing the most recent F frames of width W, height H, and C channels. The symbolic (object-centric) state is denoted as $\mathbf{Z} \in \mathbb{R}^{n \times m}$, where n is the number of detected objects and m is the number of extracted properties per object.

Hybrid Policy Reasoning. Given both object-centric and pixel-based state representations, GRAIL conducts parallel neural and symbolic policy inference, and

seamlessly combines their outputs through a blending mechanism. This hybrid policy reasoning is composed of three main components:

1. **Neural Policy:** $\pi^{\text{neu}} : \mathbb{R}^{F \times W \times H \times C} \to [0,1]^A$. This module is a neural network with parameters θ, producing a probability distribution over actions from the pixel-based input $\mathbf{X}$. Standard implementations utilize convolutional neural networks (Mnih et al. 2015; Schulman et al. 2017; Hessel et al. 2018), though visual transformers (Chen et al. 2021; Parisotto et al. 2020) are equally compatible.
2. **Logic Policy:** $\pi^{\text{log}} : \mathbb{R}^{n \times m} \to [0,1]^A$. Parameterized by ϕ, this component is a differentiable forward reasoner (Shindo et al. 2023, 2024b) operating on object-centric representations (as visualized in Figure 5). Policies are specified using FOL rules, where each rule comprises a head atom (the *action*) and body atoms (the *state predicates* serving as preconditions) (Reiter 2001; Delfosse et al. 2023b).
3. **Blending Module:** This component, parameterized by λ, is a differentiable function that computes a soft weighting between the neural and logic policies. The blender can be realized as either an explicit logic-based function ($B_\lambda : \mathbb{R}^{F \times n \times m} \to [0,1]$), an implicit neural network ingester of pixel states ($B_\lambda : \mathbb{R}^{F \times W \times H \times C} \to [0,1]$), or a hybrid of both. While logic-based blending is inherently interpretable, it presumes the presence of sufficient inductive biases—if these are absent, a neural blending approach may be preferable for adaptivity.

The agent's final action distribution is obtained by blending the neural and logic policies:

$$\pi = \beta \cdot \pi^{\text{neu}}(\mathbf{X}) + (1 - \beta) \cdot \pi^{\text{log}}(\mathbf{Z}), \tag{4}$$

where $\beta = B_\lambda(\mathbf{Z}) \in [0,1]$ denotes the blending weight inferred from the current symbolic (object-centric) state, π^{neu} is parameterized by θ, and π^{log} by ϕ. All modules of the agent are optimized jointly using PPO.

To compute value estimates, separate critics process the respective state modalities: the *neural critic*, $V^{\text{neu}} : \mathbb{R}^{W \times H \times C} \to \mathbb{R}$, for sub-symbolic states, and the *logic critic*, $V^{\text{log}} : \mathbb{R}^{E \times D} \to \mathbb{R}$, for symbolic states. These are then blended analogously:

$$V = \beta \cdot V^{\text{neu}}(\mathbf{X}) + (1 - \beta) \cdot V^{\text{log}}(\mathbf{Z}). \tag{5}$$

Policy Optimization. Policy optimization in our framework builds upon the standard PPO objective (Eq. 3), which comprises loss terms for the value function, clipped policy ratio, and action entropy regularization. To further encourage the agent to leverage both neural and logic policies, we adopt and extend the BlendRL regularization for blending:

$$H(B_\lambda) = -\beta \cdot \log \beta - (1 - \beta) \cdot \log(1 - \beta) \tag{6}$$

This *blender entropy* quantifies the uncertainty or diversity in the blending coefficient β, which softly allocates control between the neural (β) and logic ($1 - \beta$) policies. By encouraging higher entropy, the agent is discouraged from fully collapsing onto either policy and is instead incentivized to employ them both as appropriate for the state.

The final BlendRL loss function, which we denote L^{BlendRL}, thus takes the following form:

$$L^{\text{BlendRL}} = \mathbb{E}\Big[c_{\text{VF}} \cdot L^{\text{VF}} - L^{\text{CLIP}} - c_{\text{AE}} \cdot H(\pi) - c_{\text{BE}} \cdot H(B_\lambda)\Big] \tag{7}$$

where L^{VF} is the mean-squared value function error, L^{CLIP} is the clipped surrogate objective (Eq. 3), $H(\pi)$ is the action entropy, and $H(B_\lambda)$ is the blender entropy defined above. The coefficients c_{VF}, c_{AE}, and c_{BE} weight the respective terms. All components operate on the hybrid policy π and value function V as defined above.

The Concept-Grounding Bottleneck. Up to this point, we have presented the hybrid policy reasoning approach, which enables agents to reason abstractly and perform reactive decision-making. However, a central limitation of this framework is its dependence on user-supplied concept grounding, specifically the requirement for *hand-crafted valuation functions* to define predicates such as "left." This reliance restricts the framework's applicability across different environments, since concepts like "left" can have varying semantics depending on context, as illustrated in Figure 1.

To overcome this challenge, we introduce a *concept grounding module* that leverages large language models (LLMs) to automatically generate proxy functions for each extensional predicate. LLMs offer generic, intuitive representations of concepts, serving as a form of conceptual prior knowledge about how these predicates are commonly understood. By incorporating LLM-generated proxies, GRAIL augments the BlendRL framework with an additional source of supervision—referred to as *concept alignment*—that guides the learning of environment-specific grounding for abstract concepts.

Grounding Spatial Concepts in Environments

Learning to ground abstract concepts within specific environments is a crucial capability of our neuro-symbolic architecture. *Concept grounding* refers to the process by which abstract, symbolic predicates—such as `left_of` or `close_by`—are mapped to context-dependent, observable, object-centric features obtained from the environment. In GRAIL, this is achieved through the learning of differentiable *valuation functions* that output soft truth values for each predicate by processing the relational configuration of detected objects.

Rather than relying on static, hand-crafted rules, we employ parameterized and differentiable functions to evaluate spatial relational predicates. A simplistic method might use a shallow MLP that consumes the absolute positions of objects as input, but this generally fails to capture important invariances and generalization capabilities required in diverse environments. Three critical desiderata guide our improved design:

1. **Translation Invariance:** Spatial relationships should not be affected by the simultaneous translation of all involved objects. Thus, we use relative coordinates, such as differences $(x_1 - x_j,\, y_1 - y_j)$, instead of absolute positions.
2. **Normalization:** We normalize these coordinate differences by the width and height of the scene, ensuring all offset vectors $\left(\frac{x_1 - x_j}{W},\, \frac{y_1 - y_j}{H}\right)$ are scaled to $[-1, 1]^2$. This supports robustness to varying scene sizes.

3. **Generality:** Although the logic programs in Figure 6 employ only binary spatial relations, our framework is designed to handle predicates of arbitrary arity.

Accordingly, we replace the hand-crafted spatial valuation functions in BlendRL with differentiable, parameterized valuation functions $v_\psi^{\mathtt{p}} : [-1,1]^2 \to [0,1]$, implemented as neural networks and trained jointly with the rest of the architecture. Given a binary spatial predicate $\mathtt{p}$ relating a reference object (object 1, typically the player) at position (x_1, y_1) to a second object at position (x_2, y_2), the valuation function takes normalized relative coordinates as input:

$$v_\psi^{\mathtt{p}}\left(\frac{x_1 - x_2}{W}, \frac{y_1 - y_2}{H}\right) \in [0,1], \tag{8}$$

where W and H denote the width and height of the scene, respectively. The normalized relative coordinates $\left(\frac{x_1-x_2}{W}, \frac{y_1-y_2}{H}\right) \in [-1,1]^2$ ensure translation invariance and robustness to varying scene sizes. While all spatial predicates in our experiments are binary, this formulation naturally extends to n-ary predicates by concatenating the normalized offsets for each additional object.

By optimizing the PPO-based BlendRL loss (Eq. 7) with respect to the parameters ψ, the agent is able to maximize reward by flexibly adapting its concept representations—effectively grounding abstract predicates to the specific spatial and contextual nuances of each environment.

Aligning Concepts with Semantic Priors

While agents can learn to ground spatial concepts through trainable mechanisms, this alone does not guarantee that the resulting representations capture their correct semantic intent. For instance, the agent may confuse "left" with "right," as there is nothing intrinsic in the learning process to prevent these concepts from being systematically swapped. This ongoing difficulty illustrates the classic symbolic grounding problem.

To overcome this limitation, we introduce the *concept aligner* as an essential component of our framework. *Concept alignment* refers to refining the agent's learned, environment-specific concepts so they align with external semantic priors or generic conceptual knowledge—such as proxy functions derived from large language models (LLMs). By utilizing such weak supervision, the concept aligner encourages the learned valuation functions to faithfully represent the intended meanings of each concept.

Specifically, we employ LLMs to extract generic knowledge about spatial relations (e.g., how "left" should be interpreted in an abstract sense) and use this information to guide the alignment of learned valuation functions. Introducing this additional supervisory signal helps ensure that the agent's internal representations are better aligned with universal, human-interpretable semantics. While integrating humans in the loop can provide high-quality, interpretable feedback (Stammer et al. 2022; Natarajan et al. 2025), it is often costly—especially when agents learn continually from interactive experiences. Leveraging LLM-generated supervision thus significantly reduces the effort needed to obtain meaningful feedback.

(a) Policy Programs

```
% Kangaroo
up_ladder(X) :-on_ladder(P,L), same_level_ladder(P,L).
right_ladder(X) :-left_of_ladder(P,L), same_level_ladder(P,L).
left_ladder(X) :-right_of_ladder(P,L), same_level_ladder(P,L).
% Seaquest
up_air(X) :-oxygen_low(B).
up_rescue(X) :-full_divers(X).
left_to_diver(X) :-right_of_diver(P,D),
                   visible_diver(D), not_full_divers(X).
right_to_diver(X) :-left_of_diver(P,D),
                   visible_diver(D), not_full_divers(X).
up_to_diver(X) :-deeper_than_diver(P,D),
                   visible_diver(D), not_full_divers(X).
down_to_diver(X) :-higher_than_diver(P,D),
                   visible_diver(D), not_full_divers(X).
% Skiing
left_to_flag(X) :-right_of_flag(P,F), right_oriented(P).
right_to_flag(X) :-left_of_flag(P,F), left_oriented(P).
noop(X) :-right_of_flag(P,F), left_of_flag(P,R), straight_oriented(P).
```

(b) Blending Programs

```
% Kangaroo
neural_agent(X) :-close_by_monkey(P,M).
neural_agent(X) :-close_by_throwncoconut(P,TC).
logic_agent(X) :-nothing_around(X).
% Seaquest
neural_agent(X) :-close_by_enemy(P,E).
neural_agent(X) :-close_by_missile(P,M).
logic_agent(X) :-visible_diver(D).
logic_agent(X) :-oxygen_low(B).
logic_agent(X) :-full_divers(X).
% Skiing
logic_agent(X) :-true(X).
```

Figure 6. Logic programs used by **(a)** the logic actor and **(b)** the blending module in three Atari environments, generated by LLMs following (Shindo et al. 2025). Unlike previous studies, where spatial predicates are hand-crafted by human experts, GRAIL grounds these predicates as parameterized differentiable functions. In Skiing, the blending program always delegates to the logic agent.

Figure 7 illustrates the overall concept aligner module. We begin by leveraging large language models (LLMs) to generate generic, environment-agnostic representations of spatial concepts—so-called *proxy functions*—by prompting the models with detailed descriptions of the task, relevant environmental features, and objectives. Concretely, the LLM produces executable Python code that implements each spatial predicate p as a proxy function $g^{\mathrm{p}} : [-1, 1]^2 \to [0, 1]$, mapping a 2D relative offset to a soft truth value. The logic programs that define the policy structure (Figure 6) are also generated by LLMs following Shindo et al. (2025). These resulting proxy functions act as semantic priors, providing abstract "templates" of the intended meanings for each spatial relation. Throughout the reinforcement learning process, these proxy functions

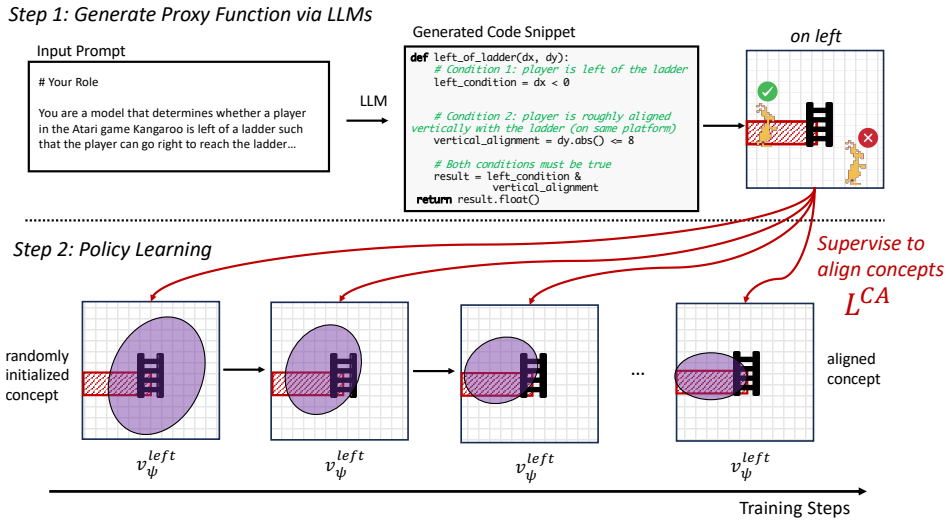

Figure 7. GRAIL maximizes reward and aligns concepts semantically. Step 1: We generate proxy functions for each spatial predicate using LLMs. These functions are represented as normalized activation maps, typically produced as code snippets and visualized over the 2D state space. **Step 2:** The proxy functions are incorporated as an additional supervision signal during policy optimization. This auxiliary signal guides the agent to semantically align its learned valuation functions with human-intended concepts—for example, preventing systematic confusion between "left" and "right". To achieve this, we introduce a binary cross-entropy loss L^{CA} that explicitly encourages the learned valuation functions to match their corresponding proxy functions.

are used as an auxiliary supervision signal: as the agent optimizes its actions for reward, the learned valuation functions are concurrently encouraged to align with the proxy functions. This coupling helps ensure that the agent's internal concept representations remain faithful to human-understandable semantics. For example, if the agent's policies incorrectly conflate the notions of "left" and "right," the proxy functions will provide a corrective influence and steer the learned concepts toward the intended interpretation. It is important to emphasize, however, that using proxy functions alone results in suboptimal performance, since they are only generic and not adapted to the specific environment. Therefore, the process of *grounding*—adapting concepts to their environment—is essential. The alignment signal introduced by our approach substantially enhances this grounding by combining generic knowledge with environment-specific experience.

We now detail how the concept aligner incorporates proxy functions into our framework. The core idea is to periodically compare the agent's learned valuation functions against the LLM-generated proxy functions over a dense grid of spatial offsets, and penalize any disagreement. Intuitively, this grid acts as a shared "canvas" on which both functions paint their activation maps; the concept alignment loss then measures how closely these two maps match for each predicate.

Concretely, we construct a $K \times K$ grid of offset vectors evenly distributed within the range $[-1, 1]^2$. For row $r \in \{1, \ldots, K\}$ and column $c \in \{1, \ldots, K\}$:

$$\Delta x_{r,c} = \frac{2c}{K+1} - 1, \qquad \Delta y_{r,c} = \frac{2r}{K+1} - 1. \tag{9}$$

At each training iteration, we evaluate both the learned valuation function $v_{r,c} = v_\psi^{\mathtt{p}}(\Delta x_{r,c}, \Delta y_{r,c})$ and the corresponding proxy function $\hat{v}_{r,c} = g^{\mathtt{p}}(\Delta x_{r,c}, \Delta y_{r,c})$ at every grid point. The discrepancy between the learned concepts and the semantic priors is measured using the mean binary cross-entropy loss:

$$L^{\mathrm{CA}} = \frac{1}{|\mathcal{P}_e|} \sum_{\mathtt{p} \in \mathcal{P}_e} \frac{1}{K^2} \sum_{r,c} \mathrm{BCE}(\hat{v}_{r,c},\, v_{r,c}) \tag{10}$$

where $\mathcal{P}_e \subset \mathcal{P}$ denotes the set of extensional predicates whose semantics are to be aligned. We choose binary cross-entropy (BCE) because both the learned valuation functions (sigmoid output) and the proxy functions produce values in $[0, 1]$ that can be interpreted as soft truth values. BCE directly penalizes pointwise deviations in these truth values, which is appropriate when the proxy provides a reasonable *shape* of the activation map. Alternative objectives, such as ranking losses (which preserve only relative orderings) or contrastive losses (which encourage separation between positive and negative regions), may be more robust when proxy magnitudes are unreliable.

To integrate this semantic supervision into learning, we augment the original BlendRL objective (see Eq. 7) with our *concept alignment loss* L^{CA}, yielding the following total objective:

$$L = L^{\mathrm{BlendRL}} + \left(1 - \gamma_{\mathrm{CA}} \cdot \frac{t}{T}\right) \cdot c_{\mathrm{CA}} \cdot L^{\mathrm{CA}} \tag{11}$$

Here, $c_{\mathrm{CA}} \in \mathbb{R}_{\geq 0}$ is the *concept alignment coefficient*, controlling the strength of the semantic prior, and $t \in \mathbb{N}$ $(0 \leq t \leq T)$ is the current optimization step out of T total steps. The term $\gamma_{\mathrm{CA}} \in [0, 1]$ is a scheduling hyperparameter that determines the rate at which the influence of L^{CA} diminishes over training. A value of $\gamma_{\mathrm{CA}} = 1$ leads the alignment loss to be annealed to zero by the end of training, while $\gamma_{\mathrm{CA}} = 0$ keeps it constant throughout. This gradual attenuation reflects the role of the concept aligner: to provide helpful guidance during the early, ambiguous phase of training, but to allow final concept grounding to be informed primarily by environment-specific experience. We examine the impact of varying c_{CA} and γ_{CA} in our ablation studies in our experiments.

Relation between grounding and alignment. *Concept grounding* enables agents to learn what a concept means in a given environment, while *concept alignment* ensures that this learned meaning remains semantically faithful to its general, language-level interpretation. By balancing these two objectives, GRAIL produces policies that are both reward-maximizing and interpretable.

Experiments

We empirically assess our framework on a variety of Atari environments, focusing on both quantitative performance and the interpretability of learned spatial concepts. Our experimental study is structured to address the following research questions:

Q1: Can GRAIL learn concept groundings that match the performance of hand-crafted valuation functions?
Q2: Does GRAIL learn interpretable, environment-specific spatial concepts rather than simply replicating LLM proxies?
Q3: Do GRAIL's learned concepts transfer to the full neuro-symbolic setting, and how do they affect the trade-off between reward maximization and goal completion?
Q4: What failure modes arise in learned concept grounding, and where does concept misalignment persist?

Experimental Setup

We compare GRAIL against two primary baselines: a neural baseline and a neuro-symbolic baseline.

Baselines. As the **neural baseline**, we use a CNN-based PPO agent (Schulman et al. 2017) with three convolutional layers (kernel sizes 8, 4, 3; strides 4, 2, 1), followed by a shared 512-dimensional fully connected layer for both the actor (18 actions) and critic (scalar value estimate). As the **neuro-symbolic baseline**, we use BlendRL (Shindo et al. 2025), which has been shown to outperform prior neuro-symbolic RL methods such as NUDGE (Delfosse et al. 2023b) and NLRL (Jiang and Luo 2019). Since GRAIL builds upon BlendRL by replacing its hand-crafted valuation functions with learned ones, this comparison directly isolates the effect of our concept grounding mechanism. In Stage 1, where the neural policy is disabled ($\beta = 0$), BlendRL reduces to a purely logic-based policy akin to NUDGE; however, NUDGE was evaluated only on simpler environments and does not support learned valuation functions, precluding a direct comparison. For GRAIL, we generate proxy functions using two LLMs—Claude4-Sonnet (Anthropic 2025) and GPT-4o (OpenAI 2025)—yielding two GRAIL variants.

Environments. We evaluate on three Atari environments from the Arcade Learning Environment (ALE) (Bellemare et al. 2013): Kangaroo, Seaquest, and Skiing. Each environment demands different spatial concepts—platform-relative navigation in Kangaroo, underwater pursuit and rescue in Seaquest, and anticipatory steering in Skiing—providing complementary coverage of the challenges GRAIL addresses. Prior neuro-symbolic RL methods such as NLRL (Jiang and Luo 2019) and NUDGE (Delfosse et al. 2023b) were evaluated on simpler or synthetic environments; Atari games pose a substantially harder test due to high-dimensional visual input, dynamic multi-object scenes, and sparse rewards. We use OCAtari (Delfosse et al. 2023a) to extract object-centric features, representing each state in both pixel-based and object-centric modalities.

Metrics. We report the *average episodic return* for quantitative comparison and the *average goals achieved per episode* to measure high-level task completion. We

further provide qualitative analysis by visualizing the learned spatial concepts as heatmaps, allowing direct inspection of how GRAIL grounds relational predicates in each environment.

Training Protocol. We adopt a two-stage training protocol. BlendRL (Shindo et al. 2025) first trains end-to-end in a single stage. This is only possible because its valuation functions are hand-crafted, effectively bypassing the concept learning problem entirely. Since GRAIL must *learn* these functions, end-to-end training would require the agent to simultaneously learn two interdependent components: (1) the meaning of each concept via valuation functions, and (2) the importance of each logic rule whose predicates rely on those very concepts. This creates a circular dependency: the agent cannot determine which rules are useful without knowing what the predicates that compose these rules mean, yet the predicates receive a gradient signal only through the rules. By first isolating concept learning in a simplified setting (Stage 1), we break this dependency and allow the valuation functions to converge to interpretable groundings before the full neuro-symbolic pipeline is trained (Stage 2).

Stage 1: Logic Policy Training on Simplified Environment. We train only the logic policy and its valuation functions, disabling the neural policy ($\beta = 0$) and removing all enemies, using HackAtari (Delfosse et al. 2024a) tasks modifications. Rule weights in the symbolic policy remain fixed. Rewards are restricted to high-level achievements (*e.g.*, reaching the child in Kangaroo or rescuing six divers in Seaquest). Episodes are capped at 3000 steps with updates every 4 steps, for a total of 10 million steps.

Stage 2: Joint Neuro-Symbolic Training on Complete Environment. We freeze the learned valuation functions and train the neural policy and blending module *from scratch* in the full environment. The neural policy and blending weights are randomly initialized; only the spatial concept groundings are carried over from Stage 1. Enemies are reactivated, there is no episode length restriction, and updates use a step size of 1. The reward structure awards 20 points for level completion and 1 point for each other reward. This stage runs for 60 million steps.

Optimization Details. All parameters are optimized using PPO with respect to the joint objective (Eq. 11). Each iteration samples 128 steps from the current policy across parallel environments. Advantages are estimated via GAE (Eq. 2) with $\gamma = 0.99$ and $\lambda = 0.95$.

The loss coefficients are $c_{\text{VF}} = 0.5$, $c_{\text{AE}} = 0.01$, and $c_{\text{BE}} = 0.01$, with clipping parameter $\epsilon = 0.1$. We use Adam with a linearly decayed learning rate from 2.5×10^{-4} and gradient clipping at 0.5. Parameters are updated for 10 epochs per iteration with 32 parallel environments. We sweep over $c_{\text{CA}} \in \{0.03, 0.1, 0.3, 1.0\}$ and report results for the best-performing setting.

Results

We now present the empirical results for both training stages individually.

Model	Kangaroo	Seaquest	Skiing
NeuralPPO	$1045_{\pm 577}$	$453_{\pm 93}$	$-5492_{\pm 2496}$
BlendRL (no CA)	$1280_{\pm 372}$	$953_{\pm 14}$	∘ $-5086_{\pm 128}$
BlendRL+Expert	$\mathbf{3683}_{\pm 29}$	$\mathbf{983}_{\pm 93}$	—
BlendRL+GPT-4o	$1112_{\pm 40}$	$783_{\pm 63}$	$-5171_{\pm 124}$
BlendRL+Claude	$910_{\pm 63}$	$559_{\pm 46}$	$-5388_{\pm 60}$
GRAIL (GPT-4o)	$3540_{\pm 131}$	$874_{\pm 40}$	$-5253_{\pm 95}$
GRAIL (Claude)	∘ $3625_{\pm 36}$	∘ $981_{\pm 162}$	$\mathbf{-5021}_{\pm 22}$

Table 1. Average episodic return on the simplified environment. GRAIL consistently matches or outperforms all baselines, including BlendRL with expert-written valuations. **Bold** = best; ∘ = second best. Higher is better (↑) for Kangaroo and Seaquest; less negative is better (↑) for Skiing. Averages over 3 seeds (100 episodes); standard deviations in subscript. BlendRL+Expert results are from the original paper (Shindo et al. 2025); BlendRL+LLM variants use fixed proxy functions without learned grounding. "No CA" denotes learned valuation functions without concept alignment ($c_{\text{CA}} = 0$). Concept-alignment coefficients: $c_{\text{CA}} = 0.3$ (Kangaroo), $c_{\text{CA}} = 0.3/0.1$ (Seaquest, GPT-4o/Claude), $c_{\text{CA}} = 0.1$ (Skiing).

Performance comparison on Atari environments

To address **Q1**, we evaluate agents on the Atari environments Kangaroo, Seaquest, and Skiing. Table 1 reports the average episodic returns during the initial training phase, in which only the logic policy is active and the neural module is disabled. Both GRAIL and BlendRL with hand-crafted valuation functions achieve high scores, significantly outperforming the purely neural agent. This demonstrates that GRAIL can effectively ground spatial concepts and attain performance on par with policies designed using expert knowledge.

In contrast, except for Skiing, BlendRL variants that directly employ LLM-generated proxy functions—from GPT-4o or Claude—perform substantially worse. This result underscores the limitation of using LLM outputs as direct replacements for expert-designed functions, and highlights the strength of GRAIL's concept alignment strategy: rather than adopting LLM-generated functions verbatim, GRAIL treats them as supervision signals, enabling it to adapt its spatial semantics to the structure of each environment.

We further compare approaches by the number of high-level goals achieved per episode. In Kangaroo, the goal is to reach the top of the screen; in Seaquest, to rescue all six divers while managing a depleting oxygen level. Figure 8 presents the average goals achieved per episode. Both GRAIL and BlendRL with hand-crafted valuation functions exhibit similarly high success rates and clearly outperform the purely neural agent, indicating that GRAIL reliably completes tasks without converging on suboptimal strategies. In these sparse-reward settings, purely neural agents tend to gravitate toward locally rewarding but ultimately ineffective behaviors—such as repeatedly firing at enemies for minor points rather than pursuing the main objectives. By leveraging LLMs to guide neuro-symbolic policies without being constrained by fixed proxy functions,

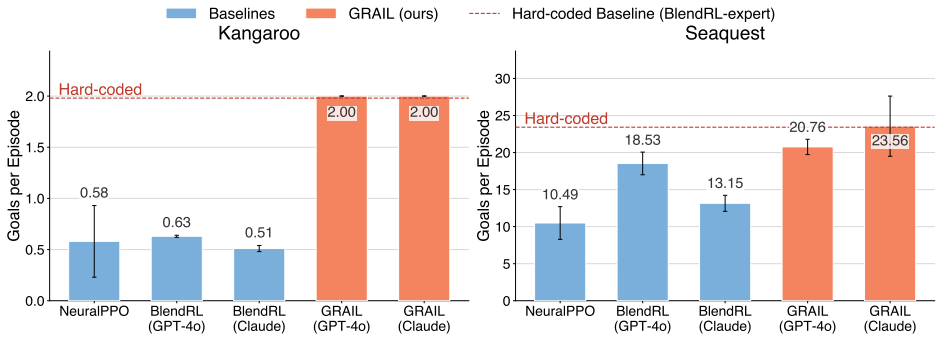

Figure 8. GRAIL achieves high-level goals reliably by grounding spatial concepts. Average number of goals achieved per episode on the simplified environment (Stage 1, enemies removed, logic policy only). In Kangaroo, a goal corresponds to reaching the child at the top platform; in Seaquest, a goal corresponds to successfully rescuing all six divers and surfacing. The red dashed line indicates the hard-coded BlendRL-expert baseline. Both GRAIL variants (GPT-4o and Claude) match or exceed this baseline, achieving 2.00 goals per episode in Kangaroo and over 20 in Seaquest. In contrast, purely neural agents and agents using raw LLM proxy functions as direct valuations fall significantly short—particularly BlendRL (Claude) in Seaquest (13.15), highlighting the insufficiency of unrefined LLM-generated concepts. Averages are computed over 3 seeds (100 episodes each); error bars indicate standard deviation.

GRAIL overcomes these limitations and achieves reliably goal-directed behavior even in the absence of hand-designed valuation functions.

Interpretability of Learned Spatial Concepts

To address **Q2**, we examine the interpretability of the learned spatial concepts by comparing GRAIL's valuation functions to both the hand-crafted functions from BlendRL and the LLM-generated proxy functions in the Kangaroo environment (Figure 9).

Although both Claude and GPT-4o capture the general semantics of spatial concepts, they fail to ground them accurately within the game's layout. For instance, GPT-4o assigns high truth values for `left_of_ladder` and `right_of_ladder` across nearly the full width of each floor but with overly narrow vertical extent, while Claude's proxy covers nearly the entire height but sharply truncates activation based on horizontal distance from the ladder. These discrepancies reveal a fundamental limitation: without environment-specific adaptation, LLM-generated proxy functions lack the precision required to serve as direct valuations of spatial predicates.

GRAIL addresses this gap by treating the proxy functions as flexible supervision rather than fixed definitions, allowing the agent to refine concept representations through environmental feedback and the reward signal. As a result, the learned spatial concepts are precisely tailored to Kangaroo's layout—correcting the shortcomings of either

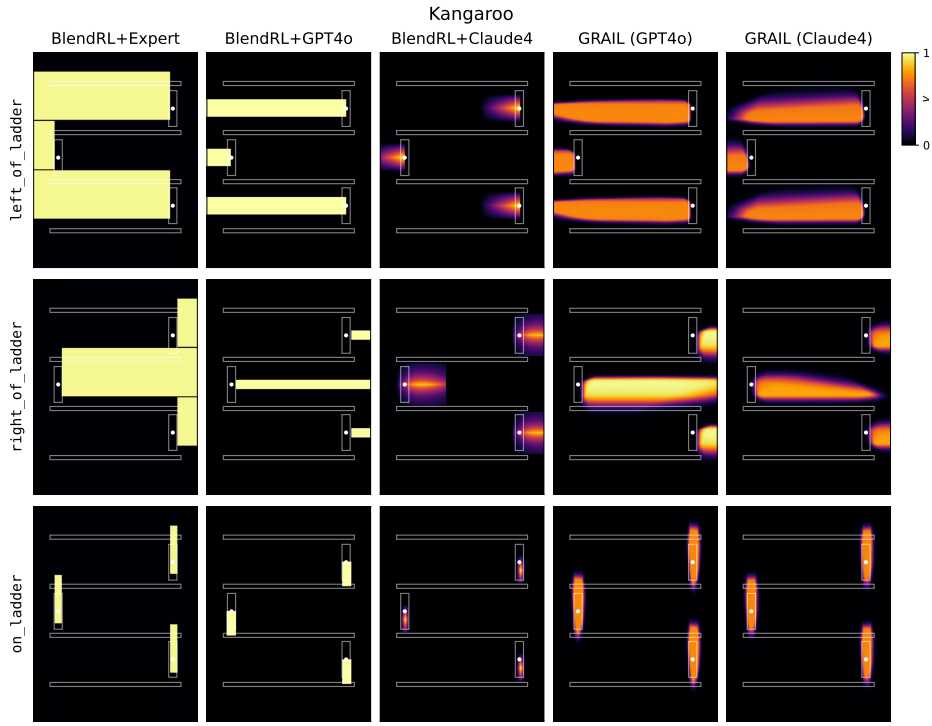

Figure 9. GRAIL produces interpretable spatial concepts that capture subtle environmental details (Kangaroo). Heatmaps indicate the truth values produced by the valuation functions as the player's position varies within the scene. Each white dot denotes the location of a ladder, relative to which truth values for the spatial predicates are evaluated. The results are visualized (from left to right) for: hand-crafted valuation functions (BlendRL+Expert), proxy functions generated by GPT-4o and Claude, and the valuation functions learned by GRAIL under weak supervision from either proxy. Platform and ladder outlines are depicted as white boxes.

proxy—and the resulting logic policy matches both the performance and interpretability of BlendRL's hand-crafted functions while significantly surpassing either LLM alone. This confirms that GRAIL does not merely replicate proxy functions but learns spatial concepts that are well-aligned with the environment's structure. We note that the heatmaps shown correspond to the best-performing seed; qualitatively similar spatial patterns emerge across all seeds despite minor variations in activation boundaries, as reflected in the standard deviations reported in Table 1.

We observe a similar pattern in Skiing (Figure 10), where the agent must learn the concepts `left_of_flag` and `right_of_flag` to navigate between flag gates. The LLM-generated proxy functions fail to capture the environment-specific semantics: GPT-4o produces a narrow vertical strip for `left_of_flag`, while Claude generates broad

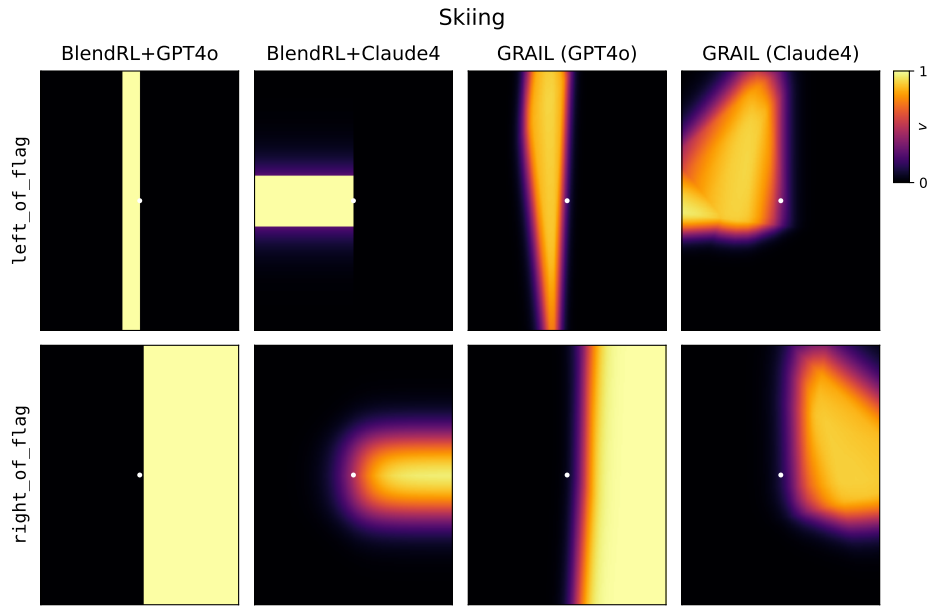

Figure 10. Learned spatial concepts in Skiing. Heatmaps show truth values for `left_of_flag` and `right_of_flag` as the player's position varies. Each white dot marks the position of a flag gate. Without concept alignment (BlendRL+GPT-4o, BlendRL+Claude), the proxy functions produce overly simplistic patterns—e.g., a narrow vertical strip or broad horizontal bands. In contrast, GRAIL learns asymmetric, environment-adapted concepts that concentrate activation in the relevant diagonal regions ahead of the player, reflecting the downhill direction of movement in Skiing.

horizontal bands—neither accounts for the vertical structure of the task. In Skiing, the player moves downhill and must identify whether it is left or right of an upcoming flag *before* reaching it. This requires the learned concepts to incorporate a vertical margin: activation should extend above the flag's position, reflecting the anticipatory nature of the steering decision. GRAIL with Claude successfully captures this skiing-specific semantics. The learned `left_of_flag` and `right_of_flag` heatmaps show activation concentrated above and to the relevant side of each flag, demonstrating that the agent has discovered that "left of a flag" in Skiing means being to the left *and* slightly ahead of it. This result highlights the adaptability of GRAIL: starting from generic LLM priors that only encode a naïve notion of horizontal direction, the framework autonomously learns environment-specific concept groundings that account for the vertical dynamics of the task.

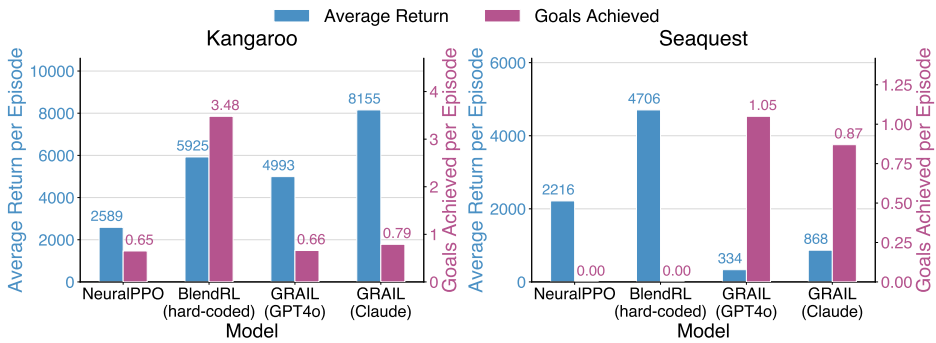

Figure 11. Return vs. goal completion in the full environment (Stage 2). Blue bars show average episodic return; pink bars show goals achieved per episode (reaching the child in Kangaroo; rescuing all divers in Seaquest). In Kangaroo, GRAIL (Claude) achieves the highest return but BlendRL's hand-coded strategy yields far more goals. In Seaquest, only GRAIL completes any goals, while baselines that maximize return fail to rescue divers entirely. Averages over 3 seeds (100 episodes); $c_{\text{CA}} = 0.3$.

Concept Learning in Joint Neuro-Symbolic Policy Training

To answer **Q3**, we evaluate each method in the complete environment, where both BlendRL and GRAIL jointly optimize their neuro-symbolic policies. The spatial concepts established in the previous stages are kept fixed throughout this phase. Figure 11 reports the average episodic return and goal completion for all baselines.

We distinguish two complementary success criteria: *episodic return* (cumulative reward, including short-term gains such as defeating enemies) and *goal completion* (achieving the environment's high-level objective—reaching the child in Kangaroo or rescuing all divers in Seaquest). These metrics can diverge, as an agent may maximize return through short-term actions without ever completing the long-horizon goal.

Our results reveal a consistent tension between these criteria. In Kangaroo, GRAIL (Claude) achieves the highest return (8155), yet BlendRL's hand-coded strategy yields far more goals (3.48 vs. 0.79 per episode), suggesting that expert-designed concepts are better tuned to this environment's specific goal structure. In Seaquest, the pattern reverses: BlendRL achieves the highest return (4706) but *zero* goal completions, whereas GRAIL is the only method that successfully rescues divers (1.05 goals per episode for GPT-4o). This demonstrates that high return does not imply meaningful task completion, and that GRAIL's learned concepts enable qualitatively different behavior—pursuing high-level goals that reward-maximizing baselines neglect entirely.

This return-versus-goal tension highlights an open challenge in neuro-symbolic RL: jointly optimizing for reward and high-level goal completion. In the following section, we analyze barriers to effective concept grounding and discuss potential paths forward.

Beyond aggregate performance, we examine whether the learned spatial concepts remain meaningful after joint training. Figure 12 visualizes the spatial concepts acquired

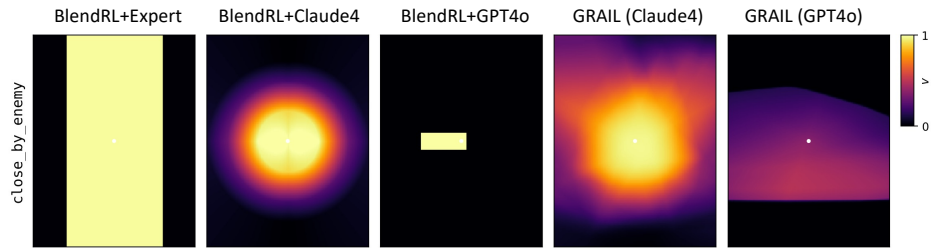

Figure 12. Stage 2: learned `close_by_enemy` concept in Seaquest. The blending module decides when to delegate control to the symbolic policy based on learned spatial predicates. Each heatmap shows the activation of `close_by_enemy` as a function of relative object position; white dots mark enemy positions. GRAIL produces broader, horizontally extended activations that capture lateral enemy movement, whereas hand-coded BlendRL functions yield near-uniform distributions and raw LLM proxies show high variability.

by the blending module in Seaquest. The hand-coded BlendRL functions fail to capture the underlying environmental semantics, producing largely uniform distributions. Raw proxy functions from Claude and GPT-4o yield inconsistent activation patterns that do not reliably reflect the spatial structure of the environment. In contrast, GRAIL produces more coherent and adaptive representations: in both the Claude and GPT-4o settings, it broadens the activation map of `close_by_enemy`, extending it horizontally to account for enemies that enter from both sides and move laterally—an adjustment well-aligned with the task's demands. These results demonstrate that GRAIL can adaptively ground spatial concepts even in complex environments with dynamic elements such as enemies.

Ablation: Impact of Concept Alignment

Figure 13 compares the concept alignment loss L^{CA} and the number of goals achieved during Stage 1 training in Kangaroo for different values of the alignment coefficient $c_{\mathrm{CA}} \in \{0.3, 1.0\}$ and the annealing factor $\gamma_{\mathrm{CA}} \in \{0, 1\}$. Two observations stand out. First, performance consistently improves as the learned concepts diverge from the LLM proxy functions—rising L^{CA} coincides with rising goals—indicating that the agent must move beyond the initial proxies to discover effective groundings. Second, annealing the alignment loss ($\gamma_{\mathrm{CA}} = 1$) accelerates convergence and reduces sensitivity to the choice of c_{CA}. Together, these results suggest that strong initial guidance from the concept aligner, gradually attenuated over training, provides the best balance between alignment and adaptability.

The remaining hyperparameters are set as follows. The grid resolution is $K = 49$, providing sufficient granularity to capture fine-grained spatial relationships while remaining computationally tractable (L^{CA} scales as $\mathcal{O}(K^2)$ per predicate per iteration). The valuation function is a compact MLP ($2 \to 64 \to 32 \to 1$ with ReLU activations and a sigmoid output): the 2-dimensional input (relative offset) is low-dimensional, and a smaller network encourages smooth, interpretable groundings rather than overfitting to spurious patterns.

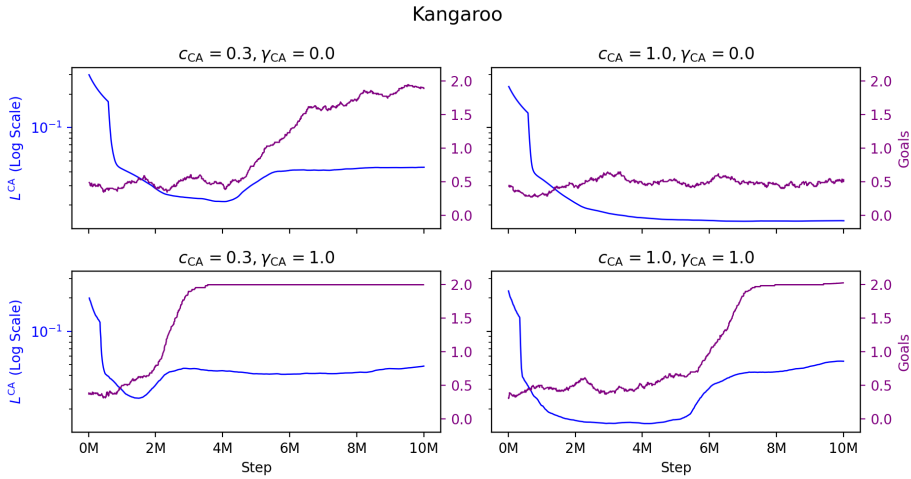

Figure 13. Concept alignment loss L^{CA} vs. goals during Stage 1 training (Kangaroo). Each panel shows a different combination of alignment coefficient c_{CA} and annealing factor γ_{CA}. Performance improves as concepts diverge from the LLM proxies (rising L^{CA}), and annealing ($\gamma_{\mathrm{CA}} = 1$) accelerates convergence.

Concept Misalignment Challenge

To address **Q4**, we examine concept misalignment—cases where the agent learns spatial groundings that are systematically incorrect despite achieving reasonable returns. Although GRAIL acquires useful concepts overall, our analysis reveals recurring failure patterns. We present a qualitative analysis of representative cases below.

Figure 14 illustrates misaligned spatial concepts in Kangaroo. In the two leftmost examples, the agent incorrectly associates `left_of_ladder` with a ladder on a different platform. A similar cross-platform confusion arises for `right_of_ladder` (center). The two rightmost examples reveal a complementary failure mode: the agent's `on_ladder` activation is biased toward the top platform even when evaluated relative to ladders on lower platforms, likely due to the disproportionately high reward for reaching the top. These observations underscore that fully aligned concept acquisition remains a significant open challenge in neuro-symbolic reinforcement learning. GRAIL mitigates this by introducing weak supervision at the predicate level, supplementing the action-level reward signal, but further work is needed to eliminate such systematic misalignments.

Conclusion

We introduced GRAIL, a neuro-symbolic reinforcement learning framework that acquires spatial concepts through direct interaction with the environment. GRAIL leverages LLMs to generate proxy functions as weak supervision for concept grounding,

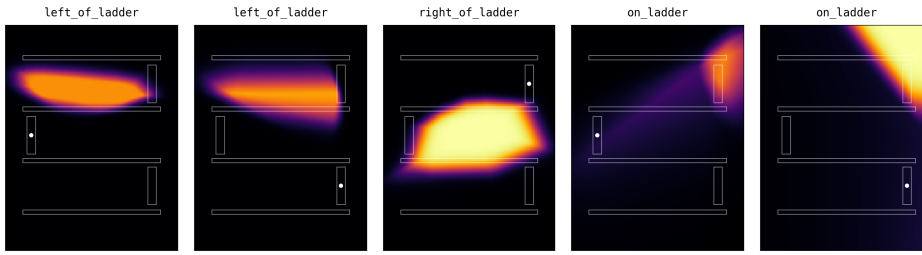

Figure 14. Examples of misaligned spatial concepts in Kangaroo. Each heatmap shows the learned truth value of a spatial predicate relative to a single ladder (white dot). Left two: `left_of_ladder` incorrectly activates for an unrelated ladder on a different platform. Center: `right_of_ladder` exhibits similar cross-platform confusion. Right two: `on_ladder` is biased toward the top platform, likely due to the disproportionately high reward for reaching it. These misalignments typically arise when the alignment coefficient c_{CA} is set too low.

and aligns these representations to each environment via a learnable concept aligner. Across Kangaroo, Seaquest, and Skiing, GRAIL matched or exceeded strong neural and neuro-symbolic baselines, producing interpretable spatial concepts on par with hand-crafted valuation functions—without requiring expert-designed concept priors.

Scope and limitations. GRAIL automates the grounding of spatial predicate *semantics*—the valuation functions that map object-pair offsets to truth values—while the predicate inventory, logic programs (Figure 6), and object-centric state extraction (OCAtari; Delfosse et al. 2023a) remain externally specified. The two-stage training procedure, in which concepts are first learned in simplified environments (Stage 1) and then frozen during joint policy training (Stage 2), constitutes a form of curriculum learning analogous to Mao et al. (2019). This design prevents concept drift under short-term reward pressure but limits end-to-end adaptability. The concept aligner serves as a warm-start rather than a hard constraint—performance improves as learned concepts *diverge* from the proxies (Figure 13)—yet a fundamentally incorrect proxy could still mislead early learning. In our experiments, a single prompt template per environment and the first syntactically valid LLM output sufficed, suggesting reasonable robustness to LLM choice, though a systematic study of prompt sensitivity and corrupted proxies remains open.

Future work. A natural next step is end-to-end training that eliminates the two-stage split, allowing concepts to co-adapt with the full policy. Extending GRAIL to n-ary and non-spatial predicates would broaden its applicability but requires richer input representations and proxy designs. On the alignment side, replacing the current binary cross-entropy objective (Eq. equation 10) with ranking or contrastive losses may improve robustness to proxy noise. A particularly important direction is *compositional* concept grounding, where coupled semantics across predicates—e.g., `between`(A, B, C) requiring joint reasoning over `left_of` and `right_of`—are modeled through differentiable logical operators. Further transparency could be gained

by replacing MLPs with more interpretable architectures such as differentiable logic gate networks (Petersen et al. 2022) or program synthesis (Wüst et al. 2024). Finally, scaling GRAIL to realistic embodied domains such as autonomous driving (Li et al. 2022), urban micromobility (Wu et al. 2025), and human-robot collaboration (Puig et al. 2024) would test its generality beyond Atari environments.

Acknowledgements

This work was partly funded by the German Federal Ministry of Education and Research, the Hessian Ministry of Higher Education, Research, Science and the Arts (HMWK) within their joint support of the National Research Center for Applied Cybersecurity ATHENE, via the "SenPai:XReLeaS" project. The work has benefited from the Clusters of Excellence "Reasonable AI" (EXC-3057) and "The Adaptive Mind" (EXC-3066), both funded by the German Research Foundation (DFG) under Germany's Excellence Strategy.

References

Acharya K, Raza W, Dourado C, Velasquez A and Song HH (2023) Neurosymbolic reinforcement learning and planning: A survey. *IEEE Transactions on Artificial Intelligence* 5(5): 1939–1953.

Allen JF (1983) Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11): 832–843.

Anthropic (2025) Claude 4 sonnet. `https://claude.ai/`. [Large language model].

Archer EJ (1966) The psychological nature of concepts. In: *Analyses of Concept Learning*. Elsevier, pp. 37–49.

Badia AP, Piot B, Kapturowski S, Sprechmann P, Vitvitskyi A, Guo ZD and Blundell C (2020) Agent57: Outperforming the atari human benchmark. In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Bellemare MG, Naddaf Y, Veness J and Bowling M (2013) The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research (JAIR)* .

Bhatt A, Palenicek D, Belousov B, Argus M, Amiranashvili A, Brox T and Peters J (2024) Crossq: Batch normalization in deep reinforcement learning for greater sample efficiency and simplicity. In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Blüml J, Derstroff C, Gregori B, Dillies E, Delfosse Q and Kersting K (2025) Deep reinforcement learning via object-centric attention. *arXiv preprint arXiv:2504.03024* .

Bruner JS, Goodnow JJ and Austin GA (1956) *A Study of Thinking*. John Wiley and Sons.

Cao Y, Li Z, Yang T, Zhang H, Zheng Y, Li Y, Hao J and Liu Y (2022) GALOIS: boosting deep reinforcement learning via generalizable logic synthesis. In: *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Cappart Q, Moisan T, Rousseau LM, Prémont-Schwarz I and Cire AA (2021) Combining reinforcement learning and constraint programming for combinatorial optimization. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Chen C, Wang F and Wang X (2024) Slot-based object-centric reinforcement learning algorithm. In: *International Conference on CYBER Technology in Automation, Control, and Intelligent Systems*.

Chen L, Lu K, Rajeswaran A, Lee K, Grover A, Laskin M, Abbeel P, Srinivas A and Mordatch I (2021) Decision transformer: Reinforcement learning via sequence modeling. In: *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Delfosse Q, Blüml J, Gregori B and Kersting K (2024a) Hackatari: Atari learning environments for robust and continual reinforcement learning. In: *Working Notes of the RLC 2024 Workshop on Interpretable Policies in Reinforcement Learning*.

Delfosse Q, Blüml J, Gregori B, Sztwiertnia S and Kersting K (2023a) Ocatari: Object-centric atari 2600 reinforcement learning environments. *arXiv preprint arXiv:2306.08649* .

Delfosse Q, Blüml J, Tatai F, Vincent T, Gregori B, Dillies E, Peters J, Rothkopf C and Kersting K (2025) Deep reinforcement learning agents are not even close to human intelligence. *arXiv preprint arXiv:2505.21731* .

Delfosse Q, Shindo H, Dhami D and Kersting K (2023b) Interpretable and explainable logical policies via neurally guided symbolic abstraction. In: *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Delfosse Q, Stammer W, Rothenbacher T, Vittal D and Kersting K (2023c) Boosting object representation learning via motion and object continuity. In: *Machine Learning and Knowledge Discovery in Databases: Research Track - European Conference (ECML PKDD)*.

Delfosse Q, Sztwiertnia S, Stammer W, Rothermel M and Kersting K (2024b) Interpretable concept bottlenecks to align reinforcement learning agents. In: *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Dillies E, Delfosse Q, Blüml J, Emunds R, Busch FP and Kersting K (2025) Better decisions through the right causal world model. *arXiv preprint arXiv:2504.07257* .

Dzeroski S, Raedt LD and Driessens K (2001) Relational reinforcement learning. *Machine Learning (MLJ)* .

Espinosa Zarlenga M, Barbiero P, Ciravegna G, Marra G, Giannini F, Diligenti M, Shams Z, Precioso F, Melacci S, Weller A, Lió P and Jamnik M (2022) Concept embedding models: Beyond the accuracy-explainability trade-off. In: *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Evans R and Grefenstette E (2018) Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research* 61: 1–64.

Feng F, Lippe P and Magliacane S (2025) Learning interactive world model for object-centric reinforcement learning. In: *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Golivand Darvishvand F, Shindo H, Sidheekh S, Kersting K and Natarajan S (2025) Human-allied relational reinforcement learning. In: *25th Annual Conference on Advances in Cognitive Systems (ACS)*.

Grandien N, Delfosse Q and Kersting K (2024) Interpretable end-to-end neurosymbolic reinforcement learning agents. *arXiv preprint arXiv:2410.14371* .

Haramati D, Daniel T and Tamar A (2024) Entity-centric reinforcement learning for object manipulation from pixels. In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Hazra R and Raedt LD (2023) Deep explainable relational reinforcement learning: A neuro-symbolic approach. In: *Machine Learning and Knowledge Discovery in Databases: Research Track - European Conference (ECML PKDD)*.

Helff L, Stammer W, Shindo H, Dhami DS and Kersting K (2023) V-lol: A diagnostic dataset for visual logical learning. *arXiv preprint arXiv:2306.07743* .

Hessel M, Modayil J, Van Hasselt H, Schaul T, Ostrovski G, Dabney W, Horgan D, Piot B, Azar M and Silver D (2018) Rainbow: Combining improvements in deep reinforcement learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Hsu J, Mao J, Tenenbaum J and Wu J (2023) What's left? concept grounding with logic-enhanced foundation models. In: *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Jiang Z and Luo S (2019) Neural logic reinforcement learning. In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Kersting K and Driessens K (2008) Non-parametric policy gradients: a unified treatment of propositional and relational domains. In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Kersting K, van Otterlo M and DeRaedt L (2004) Bellman goes relational. In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Kimura D, Ono M, Chaudhury S, Kohita R, Wachi A, Agravante DJ, Tatsubori M, Munawar A and Gray A (2021) Neuro-symbolic reinforcement learning with first-order logic. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kipf T, Elsayed GF, Mahendran A, Stone A, Sabour S, Heigold G, Jonschkowski R, Dosovitskiy A and Greff K (2022) Conditional object-centric learning from video. In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Koh PW, Nguyen T, Tang YS, Mussmann S, Pierson E, Kim B and Liang P (2020) Concept bottleneck models. In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Kohler H, Delfosse Q, Akrour R, Kersting K and Preux P (2024) Interpretable and editable programmatic tree policies for reinforcement learning. *arXiv preprint arXiv:2405.14956* .

Lang T, Toussaint M and Kersting K (2012) Exploration in relational domains for model-based reinforcement learning. *Journal of Machine Learning Research (JMLR)* .

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

Lin Z, Wu Y, Peri SV, Sun W, Singh G, Deng F, Jiang J and Ahn S (2020) SPACE: unsupervised object-oriented scene representation via spatial attention and decomposition. In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Liu A and Borisyuk A (2024) A role of environmental complexity on representation learning in deep reinforcement learning agents. *arXiv preprint arXiv:2407.03436* .

Liu IJ, Ren Z, Yeh RA and Schwing AG (2021) Semantic tracklets: An object-centric representation for visual multi-agent reinforcement learning. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Lloyd JW (1984) *Foundations of Logic Programming*. Berlin, Heidelberg: Springer.

Locatello F, Weissenborn D, Unterthiner T, Mahendran A, Heigold G, Uszkoreit J, Dosovitskiy A and Kipf T (2020) Object-centric learning with slot attention. In: *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Luo L, Zhang G, Xu H, Yang Y, Fang C and Li Q (2024) End-to-end neuro-symbolic reinforcement learning with textual explanations. In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Lyu D, Yang F, Liu B and Gustafson S (2019) SDRL: interpretable and data-efficient deep reinforcement learning leveraging symbolic planning. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Mao J, Gan C, Kohli P, Tenenbaum JB and Wu J (2019) The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584* .

Mao J, Tenenbaum JB and Wu J (2025) Neuro-symbolic concepts. *arXiv preprint arXiv:2505.06191* .

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G et al. (2013) Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* .

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

Mosbach M, Ewertz JN, Villar-Corrales A and Behnke S (2025) Sold: Slot object-centric latent dynamics models for relational manipulation learning from pixels. In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Natarajan S, Mathur S, Sidheekh S, Stammer W and Kersting K (2025) Human-in-the-loop or ai-in-the-loop? automate or collaborate? In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Nishimoto Y and Matsubara T (2026) Object-centric world models for causality-aware reinforcement learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

OpenAI (2025) Chatgpt-4o. https://openai.com/research/. [Large language model].

Parisotto E, Song F, Rae J, Pascanu R, Gulcehre C, Jayakumar S, Jaderberg M, Kaufman RL, Clark A, Noury S et al. (2020) Stabilizing transformers for reinforcement learning. In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Petersen F, Borgelt C, Kuehne H and Deussen O (2022) Deep differentiable logic gate networks. In: *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Puig X, Undersander E, Szot A, Cote MD, Yang TY, Partsey R, Desai R, Clegg A, Hlavac M, Min SY, Vondruš V, Gervet T, Berges VP, Turner JM, Maksymets O, Kira Z, Kalakrishnan M, Malik J, Chaplot DS, Jain U, Batra D, Rai A and Mottaghi R (2024) Habitat 3.0: A co-habitat for humans, avatars, and robots. In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Redmon J, Divvala S, Girshick R and Farhadi A (2016) You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Reiter R (2001) *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press.

Rosch EH (1973) Natural categories. *Cognitive psychology* 4(3): 328–350.

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

Sha J, Shindo H, Kersting K and Dhami DS (2025a) Gestalt vision: A dataset for evaluating gestalt principles in visual perception. In: *19th International Conference on Neurosymbolic Learning and Reasoning (NeSy)*.

Sha J, Shindo H, Kersting K and Dhami DS (2025b) Neuro-symbolic predicate invention: Learning relational concepts from visual scenes. *Neurosymbolic Artificial Intelligence Journal (NAIJ)* .

Shindo H, Brack M, Sudhakaran G, Dhami DS, Schramowski P and Kersting K (2024a) Deisam: Segment anything with deictic prompting. In: *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Shindo H, Delfosse Q, Dhami DS and Kersting K (2025) Blendrl: A framework for merging symbolic and neural policy learning. In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Shindo H, Nishino M and Yamamoto A (2021) Differentiable inductive logic programming for structured examples. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Shindo H, Pfanschilling V, Dhami DS and Kersting K (2023) αILP: thinking visual scenes as differentiable logic programs. *Machine Learning (MLJ)* .

Shindo H, Pfanschilling V, Dhami DS and Kersting K (2024b) Learning differentiable logic programs for abstract visual reasoning. *Machine Learning (MLJ)* .

Silver T, Chitnis R, Kumar N, McClinton W, Lozano-Pérez T, Kaelbling LP and Tenenbaum JB (2023) Predicate invention for bilevel planning. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Stammer W, Memmel M, Schramowski P and Kersting K (2022) Interactive disentanglement: Learning concepts by interacting with their prototype representations. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Stammer W, Schramowski P and Kersting K (2021) Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Stammer W, Wüst A, Steinmann D and Kersting K (2024) Neural concept binder. In: *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Steinmann D, Stammer W, Wüst A and Kersting K (2025) Object-centric concept-bottlenecks. In: *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Sun S, Wu T and Lim JJ (2020) Program guided agent. In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Verma A, Murali V, Singh R, Kohli P and Chaudhuri S (2018) Programmatically interpretable reinforcement learning. In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Vouros GA (2022) Explainable deep reinforcement learning: State of the art and challenges. *ACM Computing Surveys* 55(5): 1–39.

Wu W, He H, He J, Wang Y, Duan C, Liu Z, Li Q and Zhou B (2025) Metaurban: An embodied ai simulation platform for urban micromobility. In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Wüst A, Stammer W, Delfosse Q, Dhami DS and Kersting K (2024) Pix2code: Learning to compose neural visual concepts as programs. In: *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*.

Wüst A, Stammer W, Shindo H, Helff L, Dhami DS and Kersting K (2026) Synthesizing visual concepts as vision-language programs. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yoon J, Wu YF, Bae H and Ahn S (2023) An investigation into pre-training object-centric representations for reinforcement learning. In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Zadaianchuk A, Seitzer M and Martius G (2021) Self-supervised visual reinforcement learning with object-centric representations. In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Zhang W, Jelley A, McInroe T and Storkey A (2025) Objects matter: Object-centric world models improve reinforcement learning in visually complex environments. In: *Reinforcement Learning and Video Games Workshop@ RLC*.

Zhao X, Ding W, An Y, Du Y, Yu T, Li M, Tang M and Wang J (2023) Fast segment anything. *arXiv preprint arXiv:2306.12156* .