Going Beyond State-Reaching: Learning Abstractions for Intrinsically Motivated Option Discovery

Akhil Bagaria*, **Anita De Mello Koch***, **George Konidaris**
akhilbg@amazon.com, {anita_de_mello_koch,gdk}@brown.edu

Amazon, Brown University

Abstract

Temporal abstraction via options can improve exploration in vast environments. However, existing option discovery algorithms find subgoals that target all aspects of the state simultaneously. This *state-reaching* approach produces options that only apply in narrow regions of the state-space, eventually causing an explosion in the number of options that overwhelms the agent, and impedes progress on its primary task of reward maximization. We introduce an algorithm that instead identifies a small, relevant subset of features for each subgoal, yielding options that generalize broadly and accelerate exploration. Our approach learns abstract, transferrable options and achieves rapid exploration in three sparse-reward, image-based domains, including the Atari game MONTEZUMASREVENGE.

1 Introduction

Reinforcement learning (RL) agents must interact with the environment at every timestep. Yet, effective decision-making may demand that the agent build temporal abstractions to ease central challenges of credit assignment, exploration, and transfer (Klissarov et al., 2025). The core question is—how can RL agents autonomously discover temporally extended actions, or *options* (Sutton et al., 1999), solely via environment interaction? This problem, known as skill discovery or option discovery (Sutton et al., 1999), is the core bottleneck for scaling hierarchical RL (HRL; Barto & Mahadevan, 2003) methods to solve long-horizon, high-dimensional problems (Konidaris, 2019; Gershman, 2017; Sutton et al., 2022). However, the vast majority of skill discovery algorithms today have a common shortcoming—they build options that resort to the state-reaching objective. We illustrate this through the following example.

Recall successfully riding your bicycle for the first time. The moment you balanced your bicycle without training wheels, you realized that you had done something right. While practicing this newly acquired skill in the future, you recreated your speed and balance, not the color of your t-shirt, the weather conditions outside or the exact wear and tear on your tires. However, when existing skill discovery algorithms attempt to learn a policy to recreate a previously encountered experience (Chentanez et al., 2005), they attempt to recreate *every aspect* of that state. Examples include the ϵ-ball from deep skill graphs (Bagaria et al., 2021b), pixel-wise equality (Veeriah et al., 2018), the perceptual hashing from Go-Explore (Ecoffet et al., 2021), random projections of the target state (Dabney et al., 2021; Farebrother et al., 2023), or a neural classifier that predicts perceptual similarity between states (Vezhnevets et al., 2017; Hafner et al., 2022). This state-reaching strategy adversely impacts hierarchical agents in several ways. First, the resulting skills are nontransferable; consider a robot grabbing a cup from a cluttered table: the skill's subgoal is tied to the position of every object in the state, requiring the agent to learn combinatorially more skills. Beyond stymieing transfer, which

*Equal contribution

is a key benefit of HRL (Taylor & Stone, 2009), state-reaching artificially reduces the size of the skill's subgoal region, which complicates policy learning and eventually makes the skill less useful. Finally, recreating a state is *unscalable* because in vast, realistic domains, recreating all aspects of a particular state may simply be impossible (Bagaria & Schaul, 2023; Javed & Sutton, 2024).

Our skill discovery algorithm is inspired by intrinsic motivation (Oudeyer et al., 2008; Colas et al., 2022): during play, when children cause something interesting to happen, they try recreate it, with increasing efficiency (White, 1959). They practice this skill until they get bored and move on, but retain the ability to reuse the acquired skill later (Barto et al., 2004). Similarly, when during an AI agent's exploration, it notices something particularly interesting, it should learn a skill to reliably achieve it in the future. Other algorithms (for example, Bagaria et al. 2021b; 2025; Jinnai et al. 2020) use these ideas, but resort to a state-reaching objective. Instead, when the agent encounters a state that produces a spike in novelty (Chentanez et al., 2005) or extrinsic reward, we use *feature attribution* to identify the subset of state features responsible for the spike, and create a subgoal achievement classifier that attends only to those specific features[1]. These classifiers serve as pseudo-reward functions for training option policies, which can be composed by a high-level policy over options to maximize extrinsic reward.

We test our algorithm in three challenging exploration problems: MINIGRID-KEYCORRIDOR, VISUALTAXI, and the Atari game MONTEZUMASREVENGE—all with image-based observations and sparse reward functions. We focus on image-based domains to stress-test our algorithm: when a factored state representation is available, each dimension serves as a candidate feature and a streamlined version of our algorithm applies directly. Our method outperforms both non-hierarchical (flat) RL baselines and a state-reaching hierarchical baseline that uses the same discovered subgoals but constructs classifiers over the entire image, directly isolating the contribution of feature selection. The resulting subgoals depend on only a small number of features and are achieved across qualitatively different states, enabling skill reuse.

2 Background and Related Work

We model the agent-environment interaction as a Markov Decision Process $M = (S, A, R, T, \gamma)$ (Puterman, 1994; Sutton & Barto, 2018). To aid in reward maximization, our agent will discover options $o \in \mathcal{O} = (\mathcal{I}_o, \pi_o, \beta_o)$ (Sutton et al., 1999), where $\mathcal{I}_o \subseteq S$ is the initiation set, $\pi_o : S \to A$ is the intra-option policy over primitive actions, and $\beta_o : S \to \{0, 1\}$ is the termination condition. Each option has a maximum horizon of H steps.

Subgoal Options. While options need not optimize any objective in general, for option *discovery* it is convenient to think of options as achieving subgoals—indeed, most option discovery algorithms can be viewed this way (Precup, 2001; Bagaria, 2025; Klissarov et al., 2025). A subgoal here is not a state, it is a classifier that grounds to a potentially large set of states. A *subgoal option* uses this classifier to serve three roles: it defines the option's subgoal, its termination condition $\beta_o : S \to \{0, 1\}$, and provides a sparse pseudo-reward $R_o(s) = \beta_o(s)$ for training π_o. The initiation set $\mathcal{I}_o$ consists of states from which the option has a high probability of achieving its subgoal. This formulation reduces option discovery to identifying useful subgoals: once β_o is specified, standard policy optimization can train π_o, and policy evaluation can train $\mathcal{I}_o$ (Bagaria et al., 2023).

Universal Value Function Approximators (UVFAs) (Schaul et al., 2015) are a scalable way to learn goal-conditioned value functions $V_g : S \to \mathbb{R},\ Q_g : S \times A \to \mathbb{R}$ and their corresponding goal-conditioned policies $\pi_g : S \to A$ (Kaelbling, 1993) using function approximation. We use UVFAs to parameterize option value functions and policies: instead of representing each option's policy separately (Sutton et al., 2011), all options can condition the same function approximator on its own subgoals (Bagaria et al., 2021a): $\pi_o(s) = \arg\max_a Q_g(s, a) = \arg\max_a Q_{\beta_o}(s, a)$.

[1] We do not assume that state features are given and learn subgoal achievement classifiers directly from images, but the language of "features" helps present our ideas simply.

Novelty-based exploration. In tabular RL, count-based exploration bonuses that decay as $1/\sqrt{N[s,a]}$ can guide efficient learning (Strehl & Littman, 2008), but do not scale to large state spaces. Pseudocounts (Bellemare et al., 2016) generalize counts by assigning similar values to similar states; Coin Flip Networks (CFN; Lobel et al., 2023) are a simple, state-of-the-art technique for estimating pseudocounts from images. We use CFN over alternatives like Random Network Distillation (RND; Burda et al., 2019); Lobel et al. (2023) showed it outperforms popular novelty estimators—especially in stochastic domains—on task reward, hyperparameter robustness, and exploration-bonus interpretability. Our framework, however, is agnostic to the choice of novelty estimator. Importantly, while novelty-driven agents demonstrate impressive performance (Kapturowski et al., 2022), they do not learn options, which are key for transfer (Taylor & Stone, 2009) and high-level planning (Konidaris et al., 2018; Sutton et al., 2024). We seek to bridge this gap by using novelty as the substrate for option discovery.

Feature attribution via Shapley values. When multiple features jointly produce an outcome, how should we attribute credit to each one? Shapley values (Shapley, 1953) provide a principled answer from cooperative game theory. The key idea is to measure each feature's *marginal contribution*—how much the outcome changes when that feature is added—averaged over all possible orderings of features. Concretely, for function f and features $\{1, \ldots, k\}$, the Shapley value of feature i is:

$$\phi_i = \frac{1}{k!} \sum_{\sigma} \Big[f\big(S_i^\sigma \cup \{i\}\big) - f\big(S_i^\sigma\big) \Big], \tag{1}$$

where the sum is over all permutations σ of features and S_i^σ is the set of features appearing before i in permutation σ. Intuitively, this asks: if features were added one at a time in a random order, how much would feature i contribute on average? Computing exact Shapley values requires evaluating f on exponentially many subsets, which is intractable for high-dimensional inputs such as images. So, we use the DeepSHAP algorithm (Lundberg & Lee, 2017; Shrikumar et al., 2017), which efficiently approximates Shapley values for neural networks by backpropagating attributions from the output to per-pixel contributions.

Option discovery methods. Option discovery techniques—including reward-driven methods (Bacon et al., 2017), empowerment-based methods (Eysenbach et al., 2019; Gregor et al., 2016), spectral methods (Machado et al., 2017; Klissarov & Machado, 2023), and goal-conditioned novelty methods (Pitis et al., 2020; Bagaria et al., 2025)—all share a common limitation: their subgoals attend to all state features. Our work is orthogonal—we focus on *how* subgoals are represented, showing that identifying relevant features enables broader generalization. For a detailed survey, see Klissarov et al. (2025).

Learning Subgoal Achievement Classifiers. Most option discovery algorithms either require manually defined subgoal functions (Ecoffet et al., 2021) or resort to state-reaching (Veeriah et al., 2018), where the benefits of abstraction fade (Colas et al., 2022). Go-Explore (Ecoffet et al., 2021) exemplifies the manual approach: it uses hand-engineered cell representations as subgoals and reconstructs policies by imitating high-return trajectories. This too is a form of state-reaching and does not produce reusable temporal abstractions. Methods that learn classifiers typically still attend to all state features: perceptual similarity approaches threshold distances in observation space (Andrychowicz et al., 2017), use random projections (Dabney et al., 2021) or autoencoders (Tang et al., 2017); temporal distance methods predict whether states are within K steps of a goal (Savinov et al., 2018; Mendonca et al., 2021); and mutual information methods like DISCERN (Warde-Farley et al., 2019) abstract away uncontrollable features but not irrelevant, controllable ones. None explicitly identify a small set of relevant features for each subgoal.

Options for feature attainment. The strategy of learning options that target specific features has been explored in HRL (Hengst, 2002). STOMP (Sutton et al., 2024) learns options that each maximize a single state feature, though it is unclear why maximization is the right objective or why

options should be limited to one feature. Proto-goal RL (Bagaria & Schaul, 2023) addresses this by learning options that achieve target values for combinations of features. However, both methods assume a factored state representation is provided, while our approach discovers the relevant feature set autonomously from interaction data.

3 Method: Abstract Subgoal Discovery

Modern RL systems can efficiently compute proxies for intrinsic motivation, like novelty (Kapturowski et al., 2022). However, when used for option discovery, these signals typically drive state-reaching: the agent is rewarded for re-visiting states that previously yielded high novelty or reward, which fails to scale beyond modest navigational problems (Hansen et al., 2021). Instead, our agent measures the *interestingness* of a state using two learned models: an intrinsic reward model f_ϕ^{int} that estimates novelty (Section 2) and an extrinsic reward model f_ϕ^{ext} that predicts task reward. When the agent encounters a state s_N in which either signal spikes substantially relative to the least interesting state s_B in the same trajectory, it asks: which features of s_N account for the spike $\Delta = f(s_N) - f(s_B)$? Features whose modification substantially changes Δ are retained; the rest are discarded (Figure 4, Appendix A). A classifier is then constructed to attend only to these relevant features, serving as an abstract subgoal. The agent learns an option policy to achieve this subgoal, and the cycle repeats: the agent explores from promising subgoals, discovers new interesting states, and creates new options.

3.1 Option Selection and Exploration

Our agent maintains a set of options $\mathcal{O}$, each defined by a subgoal classifier β_o discovered during exploration. At each episode, the agent selects a subgoal to explore from, navigates there using the corresponding option policy, and then executes the CFN exploration policy π_{CFN}.

Policy over options. Not all subgoals are equally useful for exploration: some are unreachable from the current state, others lead to already-explored regions. The policy over options $\pi_{\mathcal{O}}(o \mid s)$ balances achievability with exploration potential. First, we restrict selection to options the agent can reliably achieve by defining the initiation condition:

$$\mathcal{I}_o(s) = \mathbb{I}\{V_{\beta_o}(s) > \delta\}, \tag{2}$$

where δ is a threshold hyperparameter (we use $\delta = 0.1$ in all our experiments). Among options for which $\mathcal{I}_o(s) = 1$, we score each by the expected value of exploring from its subgoal:

$$U(o) = \mathbb{E}_{s' \sim \beta_o}\left[V_{\text{CFN}}(s')\right], \tag{3}$$

$$\text{where } V_{\text{CFN}}(s') = \mathbb{E}_{\pi_{\text{CFN}}}\left[\sum_t \gamma^t \left(r_t + \lambda f_\phi^{\text{int}}(s_t)\right)\right], \tag{4}$$

and λ controls the tradeoff between extrinsic reward and intrinsic novelty; we used $\lambda = 0.01$ in all our experiments. The policy samples proportionally to utility:

$$\pi_{\mathcal{O}}(o \mid s) = \frac{U(o)}{\sum_{o' : \mathcal{I}_{o'}(s)=1} U(o')}. \tag{5}$$

Conducting exploration. Once an option o is sampled from $\pi_{\mathcal{O}}$, the agent executes its intra-option policy π_o to reach the subgoal. Upon success, the agent executes a CFN policy π_{CFN} (Lobel et al., 2023) that maximizes cumulative novelty and reward, producing a trajectory $\tau = (s_1, \ldots, s_K)$.

Triggering Option Discovery. If any state in τ produces a sufficiently large spike in either model—specifically, if $\max_{s \in \tau} f_\phi^{\text{int}}(s) > \mu_{\text{int}} + \sigma_{\text{int}}$, or $\max_{s \in \tau} f_\phi^{\text{ext}}(s) > \mu_{\text{ext}} + \sigma_{\text{ext}}$, where μ and σ are the running mean and standard deviation of the corresponding signal—the agent extracts $s_N = \arg\max_{s \in \tau} f_\phi(s)$ and $s_B = \arg\min_{s \in \tau} f_\phi(s)$, then passes them to the feature selection algorithm (Section 3.2).

3.2 Identifying Relevant Features

Given s_N and s_B from the previous section, the agent identifies which features of s_N explain the spike $\Delta = f(s_N) - f(s_B)$, where f is whichever model triggered the subgoal discovery. This is further decomposed into two steps: extracting candidate features from the image observation, and then measuring each feature's contribution to Δ.

Feature extraction. Neural representations learned end-to-end are entangled, making it difficult to intervene on individual factors (Rodriguez-Sanchez et al., 2025; Kumar et al., 2025). We instead use computer vision techniques to extract discrete, interpretable features. In visually simple domains (MINIGRID, VISUALTAXI), we subtract a background image and detect contours (Suzuki & Abe, 1985). In visually complex domains (MONTEZUMASREVENGE), we use a pre-trained Segment Anything Model (SAM; Kirillov et al., 2023) to extract object masks. Both approaches yield bounding boxes $\{b_1, \ldots, b_k\}$ representing candidate features. This step is intentionally modular: any method that produces candidate features can be substituted in. SAM and contour detection are convenient choices for our domains, but recent work on learning disentangled representations end-to-end (Rodriguez-Sanchez et al., 2025; Kumar et al., 2025) could serve as a drop-in replacement.

Feature attribution. Given candidate features, we must determine which ones explain the spike Δ. The core question is: how much does each feature b_i contribute to Δ? We consider two approaches, both grounded in the same intuition—a feature is relevant if changing it substantially affects f's prediction.

Counterfactual reset. For each feature b_i, we construct a counterfactual image c_i by replacing the patch at b_i in s_N with the corresponding patch from s_B. If $\Delta = f(s_N) - f(c_i) > \epsilon$, then b_i is relevant. This simple approach is effective when features contribute to Δ roughly independently.

DeepSHAP attribution. When features interact, we use DeepSHAP (Lundberg & Lee, 2017; Shrikumar et al., 2017) to approximate Shapley values. DeepSHAP efficiently computes these attributions by backpropagating through the triggering network f, decomposing the difference $\Delta = f(s_N) - f(s_B)$ into per-pixel contributions ϕ_p. We aggregate within each bounding box: $\text{Shapley}(b_i) = \sum_{p \in b_i} \phi_p$. Bounding boxes exceeding threshold $\mu_\phi + \sigma_\phi$ (mean and standard deviation of pixel attributions) are retained.

Both methods yield a subset of features $\mathcal{F} \subseteq \{b_1, \ldots, b_k\}$ that explain the jump in novelty or reward from s_B to s_N. These features define the scope of the subgoal classifier constructed in the next section.

3.3 Subgoal Classifier Construction

Most of s_N is irrelevant to why it was novel or extrinsically rewarding. The feature selection step identified which parts matter; everything else can vary freely and be ignored. The classifier we construct respects this—it checks only whether the relevant features $\mathcal{F}$ match their values in s_N, and ignores the rest.

We construct a non-parametric classifier $g : S \to \{0, 1\}$ since s_N is our only positive example. The classifier stores s_N and a feature extractor h that returns only the relevant features—selecting relevant indices for factored states, or cropping bounding boxes for images. Given a state s:

$$g(s) = \mathbb{I}\Big\{\mathcal{D}\big(h(s_N), h(s)\big) < \xi\Big\}, \tag{6}$$

where ξ is a threshold hyperparameter and $\mathcal{D}$ measures dissimilarity (e.g., Euclidean distance for factored states, average cross-correlation for images). The single-positive-example setting is inherent to the problem—when novelty or reward first spikes, s_N is all that is available—and the feature selection step is precisely what enables the classifier to generalize from it (Section 4).

3.4 Option Learning

The classifier g serves as the new option's termination condition β_o and provides a sparse terminating pseudo-reward $R_o(s) = g(s)$ for training. Because this reward is sparse and binary, we train the intra-option policy π_o using goal-conditioned R2D2 (Kapturowski et al., 2019) with hindsight experience replay (HER; Andrychowicz et al., 2017), a well-established combination for sparse-reward goal-reaching. Architecturally, we augment the R2D2 CNN encoder with a goal-processing torso; state and goal embeddings are concatenated and passed through an MLP that produces Q-values over actions. For each episode, we relabel five subgoals achieved during the trajectory and add the corresponding sub-trajectories to the replay buffer (Lin, 1993). The initiation function $\mathcal{I}_o(s)$ is derived from the goal-conditioned value function V_g, as described in Section 3.1. The new option is added to $\mathcal{O}$ and becomes available for future use by the policy over options $\pi_{\mathcal{O}}$.

3.5 Putting It Together

Algorithm 1 (Appendix B) summarizes the full option discovery loop. The agent alternates between exploiting existing options to reach promising subgoals and exploring to discover new ones. When exploration yields a state with sufficient spike in either novelty or extrinsic reward, the agent identifies the relevant features, constructs a new subgoal classifier, and adds the corresponding option to its repertoire. Both signal models are updated online. The intrinsic model f_ϕ^{int} is trained using the CFN objective (Lobel et al., 2023). The extrinsic model f_ϕ^{ext} is parameterized as a convolutional binary classifier trained to distinguish rewarding states $(R(s) > 0)$ from non-rewarding states $(R(s) = 0)$; this classifier approach was sufficient in our domains, but can be generalized to regression-based supervised learning models for generic MDPs.

4 Experiments

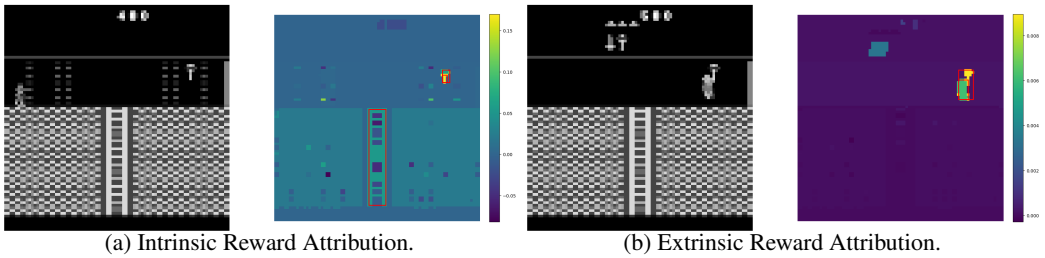
Our experiments address three questions: (1) Does our feature attribution method correctly identify the features responsible for novelty or reward spikes? (2) Do the resulting subgoal classifiers generalize across states that differ in irrelevant features? (3) Does this abstraction lead to better exploration and task performance compared to flat RL and state-reaching HRL methods?

4.1 Experimental Setup

We evaluate on three sparse-reward, image-based domains where the agent must explore effectively without relying on dense reward signals.

Domains. MINIGRID-KEYCORRIDOR-S5R3 (Chevalier-Boisvert et al., 2023) requires the agent to navigate through rooms, pick up a key, and unlock a door. Colas et al. (2022) identified it as a challenging exploration problem due to its sparse rewards and combinatorial state space. VISUALTAXI (Dietterich, 2000) requires navigating to a passenger, picking them up, and delivering them to a destination. Variants of this problem have been widely used to evaluate HRL algorithms (Dietterich, 2000; Allen et al., 2021). In both domains, the agent receives a sparse reward only upon completing the entire task—no intermediate rewards for picking up the key or passenger. MONTEZUMASREVENGE (Bellemare et al., 2013) is a canonical exploration benchmark with a natural hierarchical structure: skills like climbing ladders, jumping between platforms, and collecting items can be reused across different rooms and we seek to build option discovery algorithms that can autonomously find this hierarchical structure from experience.

Method variants. As mentioned in Section 3.2, we use the counterfactual reset method for feature attribution in MINIGRID and VISUALTAXI. In MONTEZUMASREVENGE, where visual complexity makes patch-based counterfactuals less reliable, we use SAM for feature extraction and DeepSHAP for attribution.



(a) Intrinsic Reward Attribution. (b) Extrinsic Reward Attribution.

Figure 1: **Examples of identifying relevant features in MONTEZUMASREVENGE.** For each state (black and white image), we show the Shapley value attributed to each pixel and the bounding boxes selected for the subgoal classifier. The first classifier checks if the agent has entered room 7, while the second checks if the key has been picked up.

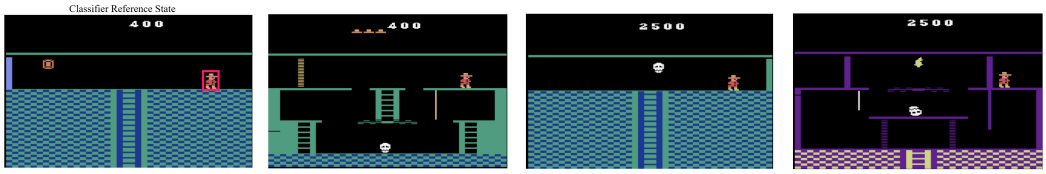


Figure 2: **Example of a discovered subgoal classifier that transfers across rooms.** The leftmost image shows the context in which this classifier was learned—the bounding box shows it only attends to the player's position. The remaining images show states where the classifier fires: despite differences in room layout, score, etc, it triggers whenever the player appears in the target region.

Training procedure. In MINIGRID and VISUALTAXI, subgoal discovery and option learning occur jointly online—exactly as described in Algorithm 1. In MONTEZUMASREVENGE, we use a two-stage approach because the non-stationarity of the novelty estimator early in training destabilizes joint option learning in this more complex domain. We first run novelty-driven exploration to discover subgoals, then train an HRL agent that learns intra-option policies and the policy over options using the discovered subgoals. The learning curves in Figure 3 show only the second stage; this makes the MONTEZUMASREVENGE comparison a rough one rather than a claim about state-of-the-art sample efficiency, but it isolates the question we care about—whether feature-selected subgoals enable better hierarchical learning than state-reaching ones. This two-stage approach, though not ideal, is standard practice in the option discovery literature (e.g., Eysenbach et al., 2019; Sharma et al., 2020); we identify fully online integration as important future work.

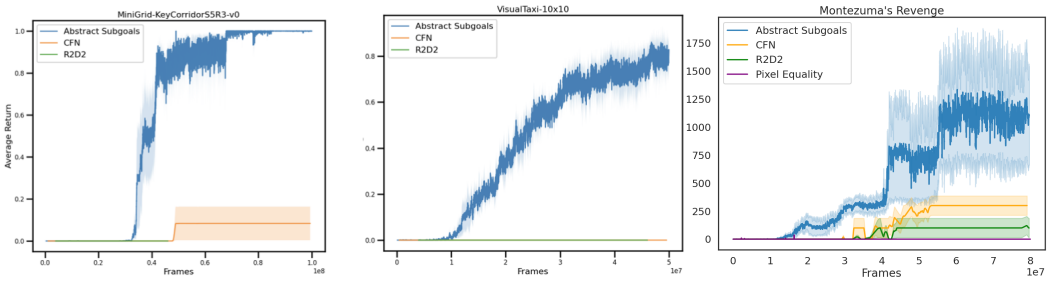


Figure 3: **Learning curves** comparing our agent (Abstract Subgoals) with a non-hierarchical novelty maximizing RL method (CFN) and a vanilla RL method (R2D2). In MONTEZUMASREVENGE, we include a state-reaching HRL baseline (Pixel Equality). Solid lines denote average undiscounted return; shaded regions denote standard deviation (5 random seeds in the first two, 4 in the third).

4.2 Qualitative Results

We first examine whether our method correctly identifies relevant features and whether the resulting classifiers generalize across visually distinct states.

Feature attribution. Figure 1 shows two examples of DeepSHAP attribution in MONTEZUMAS-REVENGE, illustrating both pathways of subgoal discovery. In (a), the agent enters a new room for the first time, triggering a novelty spike; the intrinsic-model attribution correctly highlights aspects that uniquely identify this room, while ignoring other aspects like the player's position on the screen or the score. In (b), the agent picks up a key, triggering an extrinsic reward spike; the extrinsic-model attribution focuses on the combination of the player's and the key's location. In both cases, the method identifies semantically meaningful features—entering a room, obtaining an item—rather than attending to the entire observation.

Classifier transfer. Figure 2 demonstrates that classifiers built from relevant features transfer across contexts. The leftmost image shows the state in which a classifier was learned; the bounding box indicates that it attends only to the player's position in a specific screen region. The remaining images show states where this classifier fires: despite differences in room layout, score, number of lives, and enemy positions, the classifier correctly triggers whenever the player appears in the target region.

4.3 Quantitative Results

We compare our method (Abstract Subgoals) against two baselines across all domains: R2D2 (Kapturowski et al., 2019), a flat RL method with ϵ-greedy exploration; and CFN (Lobel et al., 2023), a flat RL method with novelty-based exploration. In MONTEZUMASREVENGE, we additionally include Pixel Equality, which is identical to Abstract Subgoals except that its subgoal classifiers attend to the entire image rather than only the selected feature subset $\mathcal{F}$. This holds the discovered subgoals, candidate features, and training procedure constant, isolating the effect of feature selection. Figure 3 shows learning curves across all three domains.

In both domains, Abstract Subgoals consistently solves the task while CFN and R2D2 fail to make meaningful progress. Although CFN's count-based exploration provides intrinsic reward throughout learning, it does not discover reusable skills. Abstract Subgoals, by contrast, constructs options that decompose the task into achievable subproblems. In MONTEZUMASREVENGE, Abstract Subgoals outperforms CFN, R2D2, and, crucially, Pixel Equality. Since Abstract Subgoals and Pixel Equality use the same discovered subgoals, this comparison isolates the effect of our classifier construction: attending to relevant features rather than entire states enables options to transfer across rooms. The two-stage training procedure (Section 4.1) makes the comparison against flat baselines (CFN, R2D2) a rough one; the Pixel Equality comparison, however, is fully controlled: both methods share the same discovered subgoals and the same training procedure, so the gap directly reflects the contribution of feature selection.

5 Discussion and Conclusion

We presented an algorithm that discovers abstract subgoal options by identifying which features of a state account for its novelty or extrinsic reward, and constructing classifiers that attend only to those features. Our experiments show that this abstraction enables more efficient exploration and better task performance than both flat RL and a state-reaching HRL baseline. Our key insight is simple—when something interesting happens, figure out *why* it was interesting, not just *that* it was interesting. By grounding option discovery in feature attribution, we take a step toward agents that build abstract, reusable skills from experience.

References

Cameron Allen, Neev Parikh, Omer Gottesman, and George Konidaris. Learning Markov State Abstractions for Deep Reinforcement Learning. *Advances in Neural Information Processing Systems*, 34:8229–8241, 2021.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, pp. 5048–5058, 2017.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Akhil Bagaria. *Skill Discovery for Exploration and Planning*. Doctor of philosophy, Brown University, 2025.

Akhil Bagaria and Tom Schaul. Scaling Goal-based Exploration via Pruning Proto-goals. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pp. 3451–3460, 2023.

Akhil Bagaria, Jason Senthil, Matthew Slivinski, and George Konidaris. Robustly Learning Composable Options in Deep Reinforcement Learning. In *30th International Joint Conference on Artificial Intelligence*, 2021a.

Akhil Bagaria, Jason K. Senthil, and George Konidaris. Skill Discovery for Exploration and Planning using Deep Skill Graphs. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 521–531. PMLR, 2021b.

Akhil Bagaria, Ben M Abbatematteo, Omer Gottesman, Matt Corsaro, Sreehari Rammohan, and George Konidaris. Effectively Learning Initiation Sets in Hierarchical Reinforcement Learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Akhil Bagaria, Anita De Mello Koch, Rafael Rodriguez-Sanchez, Sam Lobel, and George Konidaris. Intrinsically Motivated Discovery of Temporally Abstract Graph-based Models of the World. In *Reinforcement Learning Journal*, 2025.

Andrew G Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems*, 13(1-2):41–77, 2003.

Andrew G Barto, Satinder Singh, and Nuttapong Chentanez. Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of the 3rd International Conference on Development and Learning*, pp. 112–19, 2004.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pp. 1471–1479, 2016.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

Nuttapong Chentanez, Andrew G Barto, and Satinder P Singh. Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pp. 1281–1288, 2005.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks. *CoRR*, abs/2306.13831, 2023.

Cédric Colas, Tristan Karch, Olivier Sigaud, and Pierre-Yves Oudeyer. Autotelic agents with intrinsically motivated goal-conditioned reinforcement learning: a short survey. *Journal of Artificial Intelligence Research*, 74:1159–1199, 2022.

Will Dabney, André Barreto, Mark Rowland, Robert Dadashi, John Quan, Marc G. Bellemare, and David Silver. The Value-Improvement Path: Towards Better Representations for Reinforcement learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, pp. 7160–7168. AAAI Press, 2021.

Thomas G Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is All You Need: Learning Skills without a Reward Function. In *International Conference on Learning Representations*, 2019.

Jesse Farebrother, Joshua Greaves, Rishabh Agarwal, Charline Le Lan, Ross Goroshin, Pablo Samuel Castro, and Marc G Bellemare. Proto-Value Networks: Scaling Representation Learning with Auxiliary Tasks. In *The Eleventh International Conference on Learning Representations* , 2023.

Samuel J Gershman. On the blessing of abstraction. *SAGE Publications Sage UK: London, England*, 2017.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep Hierarchical Planning from Pixels. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Steven Hansen, Guillaume Desjardins, Kate Baumli, David Warde-Farley, Nicolas Heess, Simon Osindero, and Volodymyr Mnih. Entropic desired dynamics for intrinsic control. *Advances in Neural Information Processing Systems*, 34:11436–11448, 2021.

Bernhard Hengst. Discovering Hierarchy in Reinforcement Learning with HEXQ. In Claude Sammut and Achim G. Hoffmann (eds.), *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pp. 243–250, 2002.

Khurram Javed and Richard S. Sutton. The big world hypothesis and its ramifications for artificial intelligence. In *Finding the Frame: An RLC Workshop for Examining Conceptual Frameworks*, 2024.

Yuu Jinnai, Jee Won Park, Marlos C. Machado, and George Konidaris. Exploration in Reinforcement Learning with Deep Covering Options. In *International Conference on Learning Representations*, 2020.

Leslie Pack Kaelbling. Learning to achieve goals. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1094–1099. Citeseer, 1993.

Steven Kapturowski, Georg Ostrovski, Will Dabney, John Quan, and Remi Munos. Recurrent Experience Replay in Distributed Reinforcement Learning. In *International Conference on Learning Representations*, 2019.

Steven Kapturowski, Víctor Campos, Ray Jiang, Nemanja Rakićević, Hado van Hasselt, Charles Blundell, and Adrià Puigdomènech Badia. Human-level Atari 200x faster. *arXiv preprint arXiv:2209.07550*, 2022.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

Martin Klissarov and Marlos C Machado. Deep laplacian-based options for temporally-extended exploration. *arXiv preprint arXiv:2301.11181*, 2023.

Martin Klissarov, Akhil Bagaria, Ziyan Luo, George Dimitri Konidaris, Doina Precup, and Marlos C. Machado. Discovering temporal structure: An overview of hierarchical reinforcement learning. *ArXiv*, abs/2506.14045, 2025.

George Konidaris. On the necessity of abstraction. *Current Opinion in Behavioral Sciences*, 29:1–7, 2019.

George Konidaris, Leslie Pack Kaelbling, and Tomas Lozano-Perez. From skills to symbols: Learning symbolic representations for abstract high-level planning. *Journal of Artificial Intelligence Research*, 61:215–289, 2018.

Akarsh Kumar, Jeff Clune, Joel Lehman, and Kenneth O. Stanley. Questioning representational optimism in deep learning: The fractured entangled representation hypothesis. *ArXiv*, abs/2505.11581, 2025.

Long-Ji Lin. Reinforcement learning for robots using neural networks. Technical report, Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, 1993.

Sam Lobel, Akhil Bagaria, and George Konidaris. Flipping coins to estimate pseudocounts for exploration in reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22594–22613. PMLR, 23–29 Jul 2023.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Neural Information Processing Systems*, 2017.

Marlos C. Machado, Marc G. Bellemare, and Michael Bowling. A laplacian framework for option discovery in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pp. 2295–2304, 2017.

Russell Mendonca, Oleh Rybkin, Kostas Daniilidis, Danijar Hafner, and Deepak Pathak. Discovering and Achieving Goals via World Models. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 24379–24391, 2021.

Pierre-Yves Oudeyer, Frederic Kaplan, et al. How can we define intrinsic motivation. In *Proc. of the 8th Conf. on Epigenetic Robotics*, volume 5, pp. 29–31, 2008.

Silviu Pitis, Harris Chan, Stephen Zhao, Bradly C. Stadie, and Jimmy Ba. Maximum Entropy Gain Exploration for Long Horizon Multi-goal Reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7750–7761. PMLR, 2020.

Doina Precup. *Temporal abstraction in reinforcement learning.* PhD thesis, University of Massachusetts at Amherst, 2001.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994.

Rafael Rodriguez-Sanchez, Cameron S. Allen, and George Dimitri Konidaris. From pixels to factors: Learning independently controllable state variables for reinforcement learning. *ArXiv*, abs/2510.02484, 2025.

Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. In *International Conference on Learning Representations*, 2018.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pp. 1312–1320, 2015.

Lloyd S Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games II*, pp. 307–317. Princeton University Press, Princeton, 1953.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-Aware Unsupervised Discovery of Skills. In *International Conference on Learning Representations (ICLR)*, 2020.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMlR, 2017.

Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

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

Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 761–768. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

Richard S Sutton, Michael Bowling, and Patrick M Pilarski. The Alberta plan for AI research. *arXiv preprint arXiv:2208.11173*, 2022.

Richard S. Sutton, Marlos C. Machado, G. Zacharias Holland, David Szepesvari, Finbarr Timbers, Brian Tanner, and Adam White. Reward-Respecting Subtasks for Model-Based Reinforcement Learning (abstract reprint). In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI*, pp. 22713. AAAI Press, 2024.

R.S. Sutton, , D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181–211, 1999.

Satoshi Suzuki and Keiichi Abe. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. #Exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in neural information processing systems*, pp. 2753–2762, 2017.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

Vivek Veeriah, Junhyuk Oh, and Satinder Singh. Many-goals reinforcement learning. *arXiv preprint arXiv:1806.09605*, 2018.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3540–3549, 2017.

David Warde-Farley, Tom Van de Wiele, Tejas D. Kulkarni, Catalin Ionescu, Steven Hansen, and Volodymyr Mnih. Unsupervised Control Through Non-Parametric Discriminative Rewards. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Robert W White. Motivation reconsidered: the concept of competence. *Psychological review*, 66(5): 297, 1959.

Supplementary Materials

The following content was not necessarily subject to peer review.

A Feature Selection Illustration

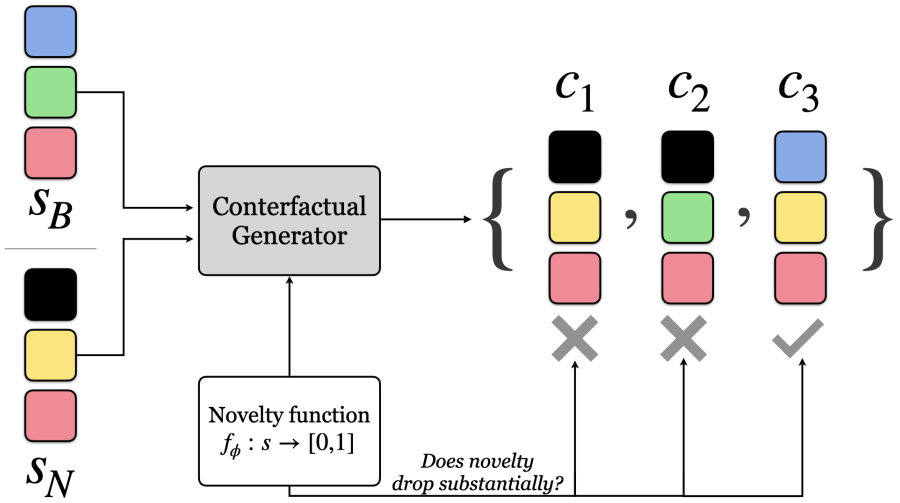


Figure 4: **Illustration of our feature selection algorithm.** The most boring and most novel states, s_B and s_N respectively, are input into the Counterfactual Generator, which outputs counterfactual states c_1, c_2, c_3—each with one feature reset to its value in s_B. Only resetting the first feature (black $\to$ blue) substantially lowers novelty, indicating that the first feature accounts for most of the change in novelty Δ. This one-at-a-time reset is effective when features contribute independently. In domains with complex feature interactions, we instead use DeepSHAP to compute each feature's marginal contribution to Δ, accounting for interactions between features. The same procedure applies when subgoal discovery is triggered by an extrinsic reward spike, with f taken to be the reward model.

B Algorithm Pseudocode

Algorithm 1 Abstract Subgoal Option Discovery

Initialize:
Initialize CFN intrinsic reward model f_ϕ^{int}, value function V_{CFN}, and policy π_{CFN}.
Initialize extrinsic reward model f_ϕ^{ext}.
Initialize goal-conditioned R2D2 to learn value function Q_g and policy π_g.
Initialize option set $\mathcal{O}$ with a default exploration option.
while True **do**
  Sample option $o \sim \pi_{\mathcal{O}}(o \mid s)$ using Equation 5.
  Execute π_o until $\beta_o(s) = 1$ or timeout.
  if timeout **then**
    continue {Resample option}
  end if
  if agent reached subgoal **then**
    Roll out π_{CFN} to get trajectory τ.
  end if
  if $\max_{s\in\tau} f_\phi^{\text{int}}(s) > \mu_{\text{int}} + \sigma_{\text{int}}$ **or** $\max_{s\in\tau} f_\phi^{\text{ext}}(s) > \mu_{\text{ext}} + \sigma_{\text{ext}}$ **then**
    Let f be the triggering model.
    Extract $s_N = \arg\max_{s\in\tau} f(s)$ and $s_B = \arg\min_{s\in\tau} f(s)$.
    Extract candidate features $\{b_1, \ldots, b_k\}$ from s_N (Section 3.2).
    Select relevant features $\mathcal{F} \subseteq \{b_1, \ldots, b_k\}$ using counterfactual reset or DeepSHAP.
    Construct classifier g from $\mathcal{F}$ (Section 3.3).
    Add new option $o' = (\mathcal{I}_{o'}, \pi_{o'}, \beta_{o'} = g)$ to $\mathcal{O}$.
  end if
  Sample minibatch from replay buffer and use for the following updates.
  Update Q_g and π_g using R2D2 and HER.
  Update π_{CFN} to maximize reward $r + \lambda f_\phi^{\text{int}}(s)$.
  Update f_ϕ^{int} and f_ϕ^{ext}.
end while