A Unified Framework for Zero-Shot Reinforcement Learning

Extended Abstract

Jacopo Di Ventura
Leiden University
Leiden, The Netherlands

Jan Felix Kleuker
Leiden University
Leiden, The Netherlands

Aske Plaat
Leiden University
Leiden, The Netherlands

Thomas Moerland
Leiden University
Leiden, The Netherlands

ABSTRACT

Zero-shot reinforcement learning (RL) has emerged as a setting for developing general agents in an unsupervised manner, capable of solving downstream tasks without additional training or planning at test-time. Unlike conventional RL, which optimizes policies for a fixed reward, zero-shot RL requires agents to encode representations rich enough to support immediate adaptation to any objective, drawing parallels to vision and language foundation models. Despite growing interest, the field lacks a common analytical lens. We present the first unified framework for zero-shot RL. Our formulation introduces a consistent notation and taxonomy that organizes existing approaches and allows direct comparison between them. Central to our framework is the classification of algorithms into two families: direct representations, which learn end-to-end mappings from rewards to policies, and compositional representations, which decompose the representation leveraging the substructure of the value function. Within this framework, we highlight shared principles and key differences across methods, and we derive an extended bound for successor-feature methods, offering a new perspective on their performance in the zero-shot regime. By consolidating existing work under a common lens, our framework provides a principled foundation for future research in zero-shot RL and outlines a clear path toward developing more general agents. The full paper can be found at arxiv.org/abs/2510.20542

KEYWORDS

Unsupervised RL, Zero-shot RL

ACM Reference Format:
Jacopo Di Ventura, Jan Felix Kleuker, Aske Plaat, and Thomas Moerland. 2026. A Unified Framework for Zero-Shot Reinforcement Learning: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/TFVS6666

1 PRELIMINARIES

Formally, the reinforcement learning [5] problem is modeled as a Markov Decision Process (MDP) [2, 4], defined as the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, p_0, r, \gamma)$. Here, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $p_0 \in \Delta(\mathcal{S})$ is the initial state distribution, $p(\cdot|s,a) \in \Delta(\mathcal{S})$ is the probabilistic transition function, $r(s,a,s')$ is the reward function and $\gamma \in [0,1)$ is the discount factor. While rewards often depend on a subset of arguments, full dependence can be reduced via marginalization. The behavior of an agent interacting with this MDP is defined by a policy, that is, a mapping $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ assigning a distribution over actions to each state. The state-value function $V^\pi(s)$ denotes the expected discounted return starting from state s and following policy π, $V^\pi(s) = \mathbb{E}_{\pi,p|s_0=s}[\sum_{t=0}^{\infty} \gamma^t\, r(s_t,a_t)]$, resulting in the RL objective $\pi^*(s) = \arg\max_{\pi\in\Pi} V^\pi(s)$.

Zero-shot RL. In the zero-shot setting [6], we consider a family of MDPs, $\mathcal{M}^{\mathcal{R}} \equiv \{(\mathcal{S},\mathcal{A},p,p_0,r,\gamma) \mid r \in \mathcal{R}\}$, where $\mathcal{R}$ denotes a specified set of reward functions under consideration, which may represent the space of all possible rewards or just a subset of tasks or skills of interest [1]. Under this setting, training is decoupled from task-specific feedback. While the agent may observe rewards during training, these arbitrary signals are non-informative of the downstream objectives. At inference, reward functions are drawn from a distribution of downstream tasks $\mathcal{D}^{\text{test}}$, which is unknown during training. The objective is to obtain optimal policies π_r^* for all $r \in \mathcal{D}^{\text{test}}$ without additional parameter optimization, planning (reasoning over state transitions to synthesize new behaviors), or substantial computation. Since the threshold for "substantial" computation lacks a formal definition, this boundary is inherently continuous; consequently, the degree to which a method qualifies as zero-shot exists on a spectrum. We discuss these nuances in Section 2.3. The objective for zero-shot RL can formally be expressed as:

$$\pi^* = \arg\max_{\pi\in\Pi} \mathbb{E}_{r\sim\mathcal{D}^{\text{test}}}\left[\mathbb{E}_{p,p_0,\pi(\cdot|s,r)}\left[\sum_{t=0}^{\infty}\gamma^t\, r(s_t,a_t,s_{t+1})\right]\right]. \quad (1)$$

2 FRAMEWORK

The downstream reward distribution $\mathcal{D}^{\text{test}}$ is unknown during training, making direct optimization of Eq. (1) infeasible. This gives rise to two main strategies: (i) adopting reward free training objectives, or (ii) train with a collection of reward functions, and optimize Eq. (1) on that set of reward functions.

These training strategies establish a natural taxonomy for zero-shot methods. Specifically, we categorize approach (i) as reward-free, while approach (ii) is designated as pseudo reward-free. Within these categories, we further distinguish between direct and compositional methods based on their reliance on value function decompositions.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/TFVS6666

2.1 Direct methods vs. Compositional Methods

A straightforward approach to zero-shot RL is to learn a reward-conditioned value function $Q(s, a|r)$, also known as a universal value function [3]. We categorize such approaches as *direct methods*. In contrast, *compositional methods* decompose the value function by learning an intermediate target, enabling the reconstruction of task-specific value functions during inference.

Direct representations. parametrize a direct mapping from state-action and reward function to optimal values:

$$Q^* : \mathcal{S} \times \mathcal{A} \times \mathcal{R} \to \mathbb{R}, \quad (s, a, r) \mapsto Q_r^*(s, a). \tag{2}$$

Policy extraction for a target task $r \sim \mathcal{D}^{\text{test}}$ is performed via $\pi_r^*(s) \in \arg\max_a Q^*(s, a, r)$. Direct representations learn the end-to-end mapping $((s, a, r) \mapsto Q_r^*(s, a))$ without leveraging structural decompositions between transition dynamics and reward functions. Consequently, the function approximator must capture the entire reward-induced structure, producing Q^* values directly from (s, a, r), with no explicit substructure between policy, occupancy, and value.

Compositional representations. explicitly leverage the structure of the value function. Under this paradigm, the learning problem reduces to estimating individual components, that are recombined at inference time according to the specified substructure. More formally, compositional methods learn representations $\mu(s, a)$, which allow to infer optimal values via some structure in the form of a decomposition operator $\mathcal{F}_\mu$:

$$Q_r^*(s, a) = \mathcal{F}_\mu(\mu, r), \tag{3}$$

where the operator $\mathcal{F}_\mu$ encodes the relationship between the learned representation and the reward. An instantiation of $\mathcal{F}_\mu$ could be as simple as a inner product between some reward representation $f(r)$ and the target, i.e. $\mathcal{F}_\mu(\cdot, r) = \langle \cdot, f(r) \rangle$.

2.2 Reward-free vs. Pseudo reward-free

At test time, the agent is evaluated under an unknown reward distribution $\mathcal{D}^{\text{test}}$, to which it has no access during training. Consequently, it must either learn some reward-conditioned representation that generalizes over the space of reward functions, or learn properties of the underlying reward-free MDP and leverage them at inference to derive a suitable policy.

Pseudo reward-free methods. Among the two aforementioned strategies, the former can be achieved by learning an arbitrary, reward-conditioned, quantity $\mu^r(s, a)$. This conditioning on rewards assumes access to reward signals during training, which may be sampled randomly via $r \sim \mathcal{D}^{\text{train}}$ such that $\text{supp}(\mathcal{D}^{\text{test}}) \subset \text{supp}(\mathcal{D}^{\text{train}})$. We refer to this setting as *pseudo-reward-free*, as it corresponds to a form of self-supervised RL in which the agent itself samples rewards, though not having access to the rewards it is evaluated under. By definition, direct methods fall within this category, as training action-value function requires explicit reward signals. By construction, all direct representations require explicit reward signals to train, and therefore all direct methods are inherently pseudo reward-free, as the train target is the value function itself, i.e. $\mu^r = Q_r^*$; or in other words the decomposition operator is the trivial operator.

Reward-free methods. In contrast, *reward-free* methods aim to learn a quantity $\mu^\pi(s, a)$ that is entirely independent of reward signals, always superscribed with a π. This corresponds more closely to a true form of unsupervised RL and can later be used to infer optimal values through a decomposition operator $\mathcal{F}_\mu$, cf Eq. 3.

2.3 The zero-shot Framework

We formalize the preceding concepts into a unified zero-shot RL framework.

Unified Zero-Shot RL Framework

Training
Choose and learn $\mu(s, a)$ such that

$$Q_r^* = \mathcal{F}_\mu(\mu, r) \tag{4}$$

Policy Extraction
Given a (unseen) reward $r \sim \mathcal{D}^{\text{test}}$ extract a policy via

$$\pi_r(s) = \arg\max \mathcal{F}_\mu(\mu, r) \tag{5}$$

Our taxonomy categorizes methods based on the nature of the target μ. In direct methods, the training target is the action-value function itself, i.e., $\mu^r = Q^*$. Conversely, compositional methods decouple the representation from the reward, utilizing either reward-free or pseudo-reward objectives.

The zero-shot boundary. This framework reveals a fundamental ambiguity in the categorization of zero-shot reinforcement learning: the absence of a standardized computational budget for policy extraction. While constraints clearly prohibit updates to μ and explicit planning—defined here as reasoning over state transitions to synthesize new behaviors—the permissible complexity of the operator $\mathcal{F}_\mu$ remains loosely defined. We contend that this lack of a rigid boundary is not a structural weakness, but rather a reflection of the inherent difficulty in establishing a universal metric. Consequently, the upper bound for what constitutes 'zero-shot' is effectively left to the practitioner, allowing for a definition that scales with the user's specific latency requirements and computational constraints. Pseudo reward-free methods provide a strict realization of zero-shot RL by enabling the extraction of an optimal policy through a single operation, such as a neural network forward pass. Conversely, reward-free approaches typically necessitate a search over the representation μ to recover the policy for a specific task, potentially incurring higher computational costs. Although the classification methods requiring a search at test-time as 'zero-shot' remains contingent on the practitioner's constraints, both categories are unified by their adherence to the core requirements of the framework.

ACKNOWLEDGMENTS

This work was supported by Shell Information Technology International Limited and the Netherlands Enterprise Agency under the grant PPS23-3-03529461.

REFERENCES

[1] Andre Barreto, Will Dabney, Remi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. 2017. Successor Features for Transfer in Reinforcement Learning. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/hash/350db081a661525235354dd3e19b8c05-Abstract.html

[2] Martin L Puterman. 1994. Markov Decision Processes: Discrete Stochastic Dynamic Programming.

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

[4] RL Stratonovich. 1960. Conditional Markov processes. *Theory of Probability And Its Applications* 5, 2 (1960), 156–178.

[5] Richard S. Sutton and Andrew Barto. 2020. *Reinforcement learning: an introduction* (second edition ed.). The MIT Press, Cambridge, Massachusetts London, England.

[6] Ahmed Touati, Jérémy Rapin, and Yann Ollivier. 2023. Does Zero-Shot Reinforcement Learning Exist?. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=MYEap_OcQI