RL ZERO: ZERO-SHOT LANGUAGE TO BEHAVIORS WITHOUT ANY SUPERVISION

Harshit Sikchi[*,1], **Siddhant Agarwal**[*,1], **Pranaya Jajoo**[*,2], **Samyak Parajuli**[*,1], **Caleb Chuck**[*,1], **Max Rudolph**[*,1], **Peter Stone**[†,1,3], **Amy Zhang**[†,1,4], **Scott Niekum**[†,5]
[1] The University of Texas at Austin, [2] University of Alberta
[3] Sony AI, [4] Meta AI,[5] UMass Amherst

ABSTRACT

Rewards remain an opaque way to specify tasks for Reinforcement Learning, as humans are often unable to predict the optimal behavior corresponding to any given reward function, leading to poor reward design and reward hacking. Language presents an appealing way to communicate intent to agents but prior efforts to bypass reward design through language have been limited by costly and unscalable labeling efforts. In this work, we propose a method for a completely unsupervised alternative to grounding language instructions in a *zero-shot* manner to obtain policies. We present a solution that takes the form of *imagine*, *project*, and *imitate*: The agent imagines an observation sequence corresponding to the language description of a task, projects the imagined sequence to our target domain, and grounds it to a policy. We show that zero-shot language-to-behavior policy can be achieved by first projecting the imagined sequences, generated using video models, into real observations of an unsupervised RL agent and using zero-shot imitation to mimic the projected observations. Our method, `RLZero`, is the first to our knowledge to show zero-shot language to behavior generation abilities without any supervision on a variety of tasks. We further show that `RLZero` can also generate policies zero-shot from cross-embodied videos such as those scraped from YouTube.

Project page: hari-sikchi.github.io/rlzero

1 INTRODUCTION

Underlying the many successes of RL lies the engineering challenge of task specification, where a skilled expert painstakingly designs a reward function. Not only does this restrict the scaling of RL agents, but it also makes those agents uninterpretable to any user inexperienced in reward design. Even for experts, reasoning about simple reward functions is generally infeasible because these functions can be easily hacked (Krakovna, 2018; Amodei et al., 2016; Dulac-Arnold et al., 2021); i.e the optimal policies for the reward function produce behaviors that do not align with what the human intended. Language is an expressive communication channel for human intent and allows bypassing reward design, but learning a mapping from language to behaviors has historically required collecting and annotating behaviors that correspond to language (Goyal et al., 2021a; Jang et al., 2022; O'Neill et al., 2023). This strategy is impractical at scale where samples from the agent's large space of behaviors need to be labeled. Instead, an approach that makes use of models learned in a purely unsupervised way becomes desirable.

How can generalist agents translate language commands into behaviors? Large-scale multimodal foundation models (Wang et al., 2024a) provide us with part of the solution. Trained on large amounts of internet data, they can generate video segments that communicate what performing a task entails. An issue in using video generation models to demonstrate behavior is that they may generate video frames demonstrating tasks that are out of distribution of the current agent's domain; for instance, the current agent can be in a simulated environment, and the video generation model produces videos resembling the real world. In this work, we propose to fix this problem by projecting

[*] Equal contribution,[†] Equal Advising. Correspondence to hsikchi@utexas.edu

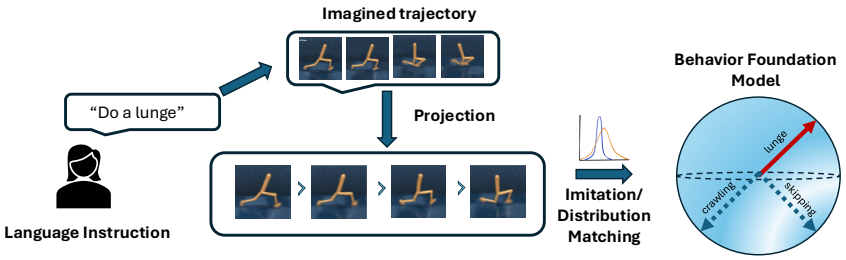


Figure 1: `RLZero` framework of **imagine**, **project**, and **imitate**: A video trajectory is imagined using the text prompt in the agent's observation space and projected to real observations. Using observation-only zero-shot imitation learning, the generated trajectory is grounded in a policy that mimics the behavior demonstrated by the video.

frames to agent observations under a semantic similarity scoring metric (Radford et al., 2021b; Zhai et al., 2023). This frame-by-frame similarity search allows us to produce a sequence of observations grounded in the agent's interaction history and presents an expectation of what the task would look like in the agent's observation space. However, the discovered frame sequence might still not adhere to environment dynamics or even be feasible. This presents us with our next question: How do we generate behaviors that *resemble* the grounded imagined trajectories in a zero-shot manner? We refer to a zero-shot RL agent as an agent that can solve any reward maximization task in a given environment, instantly with no additional planning or learning, after an initial reward-free learning phase, similar to Touati et al. (2023).

Prior research (Rocamonde et al., 2023; Baumli et al., 2023; Sontakke et al., 2024) has used large Vision-Language Models (VLMs) to obtain proxy rewards for a language command. Even when a reward function is obtained, training a policy for a reward function from scratch for each task description in language is time-consuming, and potentially unsafe due to reward hacking. Alternately, other works attempt to provide expert demonstrations and annotate each skill of an agent with a language description hoping for generalization to new, unannotated skills. Collecting expert demonstrations for the wide variety of skills possible in the world can get prohibitively expensive. Unsupervised RL offers an ideal tool for zero-shot behavior inference, enabling an agent to leverage task-agnostic prior interactions with the environment to encode diverse behaviors that can be queried to obtain a near-optimal policy in a zero-shot manner given a reward function. Specifically, we rely on the successor feature-based family of unsupervised RL methods, sometimes termed as Behavior Foundation Models (BFM) (Touati & Ollivier, 2021; Park et al., 2024a; Agarwal et al., 2024), that allow learning behaviors for all possible reward functions subject to model capacity constraints. BFMs work by pretraining optimal policies for all reward functions defined in the span of learned state features. During inference, the optimal policy corresponding to a particular reward function can be obtained in closed form.

We sidestep the requirement of reward functions and instead frame the problem of language-to-skill inference as matching state-only distributions to the grounded imagined trajectories. Notably, this work leverages the capability of unsupervised RL methods to provide a zero-shot solution to distribution matching. This approach parallels the imagination capabilities of humans to picture in their mind possibilities in the real world (Sarbin, 2004; Sarbin & Juhasz, 1970; Pylyshyn, 2002) and then rely on past experiences, memories, and abilities to inform their actions. Our framework (illustrated in figure 1) attempts to do something similar – `RLZero` works in three simple steps: a) **Imagine:** Imagine trajectories given a language command. b) **Project**: The frames of imagined trajectories are projected to real observations of the agent. c) **Imitate:** RLZero leverages the agent's prior environmental interactions to output a policy in a zero-shot manner that matches the state visitation distribution of the imagined trajectories. Our experiments show that RLZero is a promising approach to designing an interpretable link connecting humans to RL agents. We demonstrate that RLZero is an effective method on a variety of tasks where reward function design would require an expert reward engineer. We show that RLZero also opens possibilities for zero-shot cross-embodiment transfer, a first approach to be able to do this to our knowledge. Our contribution is the framework of imagine, project, and zero-shot imitate, which diverges from the prior approach of using VLMs as reward functions—which can be hacked—and instead focusing on zero-shot imitation with unsupervised RL, which admits a unique solution that matches the imagined behavior.

2 Related Work

Language and Control: There is a rich history of using language to solve various tasks in RL: task specification (Thomason et al., 2015; Goyal et al., 2021b; Ma et al., 2023; Baumli et al., 2023; Rocamonde et al., 2023; Stepputtis et al., 2020; Brohan et al., 2022; 2023; Sontakke et al., 2024), transfer and generalization (Goyal et al., 2021a; Jang et al., 2022; Liang et al., 2023), using language to provide hierarchies that allow for solving long-horizon tasks (Ahn et al., 2022; Jiang et al., 2019), driving exploration (Goyal et al., 2019; Harrison et al., 2017; Wang et al., 2023; Ma et al., 2024), human-in-the-loop learning (Chen et al., 2020; Chevalier-Boisvert et al., 2019), giving feedback to AI agents (Wang et al., 2024b), reward design (Yu et al., 2023), etc. Most existing methods either require labels for mapping language to low-level actions or generate reward functions that need to be trained by interacting with the environment to generate a low-level control policy. Recent work (Mazzaglia et al., 2024) proposed an unsupervised approach to grounding language to low-level skills but requires re-training the RL agent for each given task prompt. In contrast, our work presents a method that allows for zero-shot mapping of languages to low-level skills. A large portion of prior work has been limited to using language in a setting where expert demonstrations are provided, but this puts a heavy burden on data collection to cover the large number of skills possible in the environment, which quickly becomes impractical considering the vast array of interactions intelligent agents can perform with their environments. Our approach forgoes this limitation by relying on a zero-shot RL agent capable of mimicking arbitrary imaginations generated for a given text.

Zero-shot RL: Zero-shot RL promises the ability to quickly produce optimal policies for any given task defined by a reward function. A wide variety of methods have been developed to achieve zero-shot RL, which are in some ways generalizations of multi-task RL (Caruana, 1997). Most of these works assume a class of tasks where they can produce policies zero-shot. These tasks can be goal-conditioned (Kaelbling, 1993; Durugkar et al., 2021; Agarwal et al., 2023; Sikchi et al., 2023; Ma et al., 2022b), a linear span of certain state-features (Dayan, 1993; Barreto et al., 2017; Blier et al., 2021b; Touati & Ollivier, 2021; Park et al., 2024a; Agarwal et al., 2024) or some combination of some skills (Eysenbach et al., 2018; 2022; Park et al., 2024b). Recent works (Wu et al., 2018; Touati & Ollivier, 2021; Touati et al., 2023; Park et al., 2024a; Agarwal et al., 2024) employ a successor measure-based representation learning objective to be able to provide near-optimal policies for arbitrary reward function subject to model capacity constraints. Our work leverages these methods and finds the *best* reward supported by the representations that will produce the language-conditioned imagined trajectory.

3 Preliminaries

`RLZero` uses generative models to imagine trajectories from language prompts and produces a policy by imitating a projection of this imagined trajectory. In this section, we introduce the notion of trajectory generation, imitation learning, and zero-shot RL.

Multimodal Video-Foundation Models (ViFMs) and In-Domain Video Generation: Multimodal ViFMs (Wang et al., 2022; 2024a; Tong et al., 2022) enable the understanding of video data in a shared representation space of other modalities such as text or audio. These shared representations can be used to condition video generation on different input modalities (Kondratyuk et al., 2023; Blattmann et al., 2023). Notably, these models can utilize text prompts to guide content, style, and motion, or employ an image as the initial frame for a subsequent video sequence. For this work, we use off-the-shelf video generation models VM that generate a sequence of video frames $\{i_1, i_2, ...i_n\}$ given a task specified in natural language l by first converting the language prompt to a common embedding space across modalities; formally, $VM : l \rightarrow \{i_1, i_2, ...i_n\}$.

Imitation Learning through Distribution Matching: We consider a learning agent in a Markov Decision Process (MDP) (Puterman, 2014; Sutton & Barto, 2018) which is defined as a tuple: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma, d_0)$ where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces respectively, p denotes the transition function with $p(s'|s, a)$ indicating the probability of transitioning from s to s' taking action a; r denotes the reward function, $\gamma \in (0, 1)$ specifies the discount factor and d_0 denotes the initial state distribution. The reinforcement learning objective is to obtain a policy $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ that maximizes expected return: $\mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t)]$, where we use $\mathbb{E}_\pi$ to denote the expectation under

Figure 2: **Example Imagined Trajectories**: The video model imagines frames conditioned on the task specified as a text prompt 'do lunges'.

the distribution induced by $a_t \sim \pi(\cdot|s_t), s_{t+1} \sim p(\cdot|s_t, a_t)$ and $\Delta(\mathcal{A})$ denotes a probability simplex supported over $\mathcal{A}$.

An imitation learning agent does not have access to the reward function, R, but has access to an "expert" trajectory (or a set of "expert" trajectories) from a policy that maximizes the reward function. Inverse Reinforcement Learning methods Ng et al. (2000) infer the reward function (explicitly or implicitly) from the trajectories and produce the policy that maximizes this reward. Distribution matching objectives Ghasemipour et al. (2020); Ni et al. (2021) for IRL have been commonly used in some recent work (Garg et al., 2021; Sikchi et al., 2024), removing the need for inferring reward functions altogether. The distribution matching based imitation learning objective is $\min_\pi \mathcal{D}(\rho^\pi, \rho^E)$, where ρ^π is the visitation distribution of the policy π (defined by the probability of being in state s starting from the initial state distribution s_0 and following the policy π), ρ^E is the visitation distribution exhibited by the "expert" trajectory and $\mathcal{D}$ is a function to compare the closeness of the distributions. f-Divergences are commonly used as a measure of distance between distributions.

Zero Shot RL through Successor Measure (BFM): Successor Measure (Blier et al., 2021a) learning has been recently studied (Touati & Ollivier, 2021; Agarwal et al., 2024) as an unsupervised RL objective for its ability to describe long-term behavior of the policy in the environment. Mathematically, successor measures define the measure over future states visited as M^π,

$$M^\pi(s, a, X) = \mathbb{E}_\pi[\sum_{t \geq 0} \gamma^t p^\pi(s_{t+1} \in X|s, a)] \;\; \forall X \subset \mathcal{S}. \tag{1}$$

Representing the successor measure for any policy π as $\psi^\pi(s, a)^T \varphi(s^+)$, these methods facilitate extraction of a state-representation $\varphi(s)$ that is suitable for RL. Then, learning policies π_z(where the policies are represented using latents z) that are near-optimal for a reward function defined in the span of learned state-features $r(s) = \varphi(s) \cdot z$. At test time, the policy for any given reward function can then be obtained analytically (with no additional experiential data) by solving the following linear regression:

$$\min_z (r(s) - \varphi(s) \cdot z)^2 \tag{2}$$

4 RLZero: Zero-Shot Prompt to Policy

`RLZero` uses components trained without any explicit supervision to map language to behaviors. For each domain, we consider a dataset of exploratory reward-free interactions d^O and a BFM $(\varphi(s), \pi_z)$ pre-trained on d^O. In the following sections, we describe the steps involved in detail. First, we present how an imagined trajectory is generated from a prompt. Then, we discuss how this imagined trajectory is projected to real observations of an agent. Finally, we describe the zero-shot procedure for inferring a policy that matches the behavior in the imagined trajectory.

4.1 Imagine: Generative Video Modeling

Grounding language to tasks has historically (Goyal et al., 2021a; Jang et al., 2022; O'Neill et al., 2023) required costly annotation labels that map language to task examples specified through image or state trajectories. Large video-language foundation models (ViFMs) help lift that requirement by training on vast amounts of internet videos, thus giving us a rich prior of grounding language commands to videos. We rely on a generative video modeling approach, GenRL (Mazzaglia et al., 2024), that uses a video-language task encoder provided by an off-the-shelf ViFM (InternVideo2 (Wang et al., 2025)) and trains an environment-specific GRU model to imagine a sequence of next latent states. These states are then reconstructed to pixels within the environment

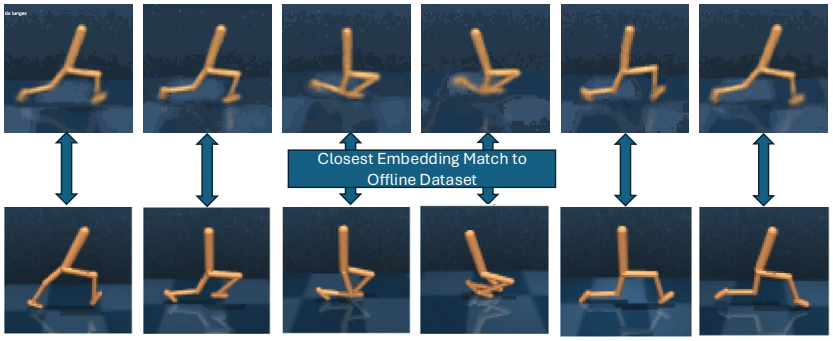


Figure 3: **Grounding Imagination in Real Observations**: We use nearest image retrieval defined by cosine similarity in the embedding space to output a real observation from the dataset that matches the imagined observation.

domain. Training the video generation model does not require labels mapping language to tasks and is fully unsupervised. Thus, given a language instruction e^l, we obtain a sequence of frames $(i_1, i_2 ... i_T) = VM(e^l)$ that represents an imagination of what the task looks like in the environment domain. Figure 2 shows an example of what these imaginings look like using an off-the-shelf video generation model (Mazzaglia et al., 2024). While we rely on an environment-specific task-conditioned video generator for this work, with advancements in ViFM scaling and controllable video generation (Bruce et al., 2024; Hu et al., 2022; Ni et al., 2023; Chen et al., 2025), a few examples from the target environment domain may be sufficient to generate high-quality in-domain imaginations.

4.2 Project: Grounding to Agent's Observation Space

The imaginings produced by ViFMs can be noisy, unrealizable, and not exactly representative of the domain. We propose to use similarity-based retrieval for the nearest frames in the dataset of the agent's prior environmental interactions d^O to project the imagined trajectories to real observations. This step allows us to match imagination to real observations in the semantic space. Semantic matching also allows us the flexibility to replace imaginings with a video demonstration of a task potentially by a different agent in a different domain (e.g. zero-shot video to policy discussed in Section 5.2). In this work, we use a performant image embedding approach for retrieval, SigLIP (Zhai et al., 2023), to map both the imagined frame and agent observation to the same latent embedding space, which has been pre-trained for similarity matching on an internet-scale dataset with a contrastive objective. We use an encoding function $\mathcal{E} : \mathcal{I} \to \mathcal{Z}$ to individually map a sequence of images to shared text-image embedding space. For each consecutive k length sequence of frames in the imagined trajectory, we output the following agent observations:

$$o_{t-k:t} = \arg\max_{o_{t-k:t}} \frac{\mathcal{E}(o_{t-k:t}) \cdot \mathcal{E}(i_{t-k:t})}{\|\mathcal{E}(o_{t-k:t})\| \|\mathcal{E}(i_{t-k:t})\|} \quad \forall t \in [T]. \tag{3}$$

Using k previous frames allows us to identify state variables that correspond to quantities such as velocity, acceleration, etc., that are not identifiable from a single frame. This technique of 'frame-stacking' is commonly used in visual RL (Laskin et al., 2020) and has the effect of making the observation inputs Markov. Using the offline interaction dataset, we find corresponding proprioceptive states in addition to the real observation that we will subsequently use for distribution matching. While we rely on proprioceptive states in this work to solve distribution matching, in general our approach is not limited as BFMs may be trained with image observations if the state information is unavailable.

4.3 Imitate: Distribution Matching with Zero-Shot RL

We take the distribution matching perspective of imitation learning (Ho & Ermon, 2016; Ghasemipour et al., 2020) and find a policy that matches the state visitation distributions of the grounded imagined trajectories (expert). We use successor measures based zero-shot RL methods (Agarwal et al., 2024;

Touati et al., 2023) to produce policies from this distribution matching objective efficiently. These require pretraining of successor measures using reward-free interaction data (d^O with distribution ρ). Finding the optimal policy simply reduces to finding the optimal latent z for the following distribution matching objective:

$$z_{imit} = \arg\min_z \mathcal{D}(\rho^{\pi_z}(s), \rho^E(s)), \tag{4}$$

where ρ^E and ρ^{π_z} are state visitation distribution of the "expert" imagined trajectories and of the policy π^z respectively and $\mathcal{D}$ can be chosen to be mean-squared error, f-divergence, Integral Probability Metrics (IPM), etc. In general, minimizing the distance via gradient descent can provide a solution z_{imit} to distribution matching. For the special case of KL divergence, Theorem 1 shows that z_{imit} can be obtained in closed form using a learned distribution ratio between expert and offline interaction dataset ρ^E/ρ.

Theorem 1. *Define $J(\pi, r)$ to be the expected return of a policy π under reward r. For an offline dataset d^O with density ρ, a learned log distribution ratio: $\nu(s) = \log(\frac{\rho^E(s)}{\rho(s)})$, $D_{KL}(\rho^\pi, \rho^E) \leq -J(\pi, r^{imit}) + D_{KL}(\rho^\pi(s,a), \rho(s,a))$ where $r^{imit}(s) = \nu(s)\ \forall s$. The corresponding z_{imit} minimizing the upper bound is given by $z_{imit} = \mathbb{E}_\rho[r^{imit}(s)\varphi(s)] = \mathbb{E}_{\rho^E}[\frac{\nu(s)}{e^{\nu(s)}}\varphi(s)]$ where φ denotes state features learned by the BFM.*

Thus, with the reward functions specified by r^{imit}, we can use the closed form solution of $z_{imit} = \mathbb{E}_\rho[\varphi(s)r^{imit}(s)]$ to retrieve the policy that mimics the grounded imagined behavior. This reward function requires learning a discriminator to obtain the distribution ratio, which can lead to instabilities, but a heuristic yet performant alternative is to use a shaped reward function $r(s) = e^{\nu(s)}$, similar to Pirotta et al. (2023), which allows zero-shot inference ($z_{imit} = \mathbb{E}_{\rho^E}[\varphi(s)]$) without learning a discriminator. We compare both approaches in Appendix C.1. The performance for both these methods are almost identical and we defer to the latter one in all our experiments. Using a state-only visitation matching objective can be limiting in the case where environmental dynamics permit the permutation of observation sequences that result in the same visitation distribution. This limitation can be relaxed by instead matching visitation on $\{s, s'\}$. This requires minimal changes to training the BFM, but we found this to not be a limitation with the environments we consider. The complete algorithm for `RLZero` can be found in Algorithm 1

Algorithm 1 `RLZero`

1: Init: Pretrained Video Generation Model VM, Pretrained BFM π_z, Offline Exploration Dataset d^O
2: Given: text prompt t
3: Generate imagination video given the text prompt: $\{i_1, i_2, ..i_l\} = VM(t)$
4: Project the imagined frames to real observations using embedding similarity as in Eq 3.
5: Use Theorem 1 for zero-shot inference to obtain BFM$(\{s_1, s_2, ..., s_l\}) = z_{imit}$ and return $\pi_{z_{imit}}$.

5 Experiments

Our experiments seek to understand the quality of behaviors that the `RLZero` approach is able to produce given language prompts. The evaluation of these behaviors can be challenging as, unlike the traditional RL setting, we do not have access to a ground truth reward function. Instead, we have prompts that can be inherently ambiguous but reflect the reality of human-robot interaction. An obvious evaluation metric is to ask humans how much the generated behavior resembles their expectation of the behavior given the prompt. We use multimodal LLMs to evaluate such preferences as a proxy to human preferences, as recent studies (Chen et al., 2024) have shown them to be correlated (up to 79.3%).

Setup: We consider four DM control tasks (Cheetah, Walker, Quadruped, and Stickman (Mazzaglia et al., 2024). The Stickman environment reflects a human morphology with challenging control due to a large observation and action space. For task-conditioned video-generation we use off-the-shelf models from Mazzaglia et al. (2024). To obtain the nearest observation corresponding to the imagined image, we use SigLIP (Zhai et al., 2023), a state-of-the-art image-text embedding model. In all environments, we collect data d^O using a pure exploration algorithm RND (Burda et al., 2018) using the protocol specified in ExoRL (Yarats et al., 2022). For Stickman, we augment our dataset with replay buffers of the agent trained for run and walk behaviors, as obtaining meaningful tasks with pure

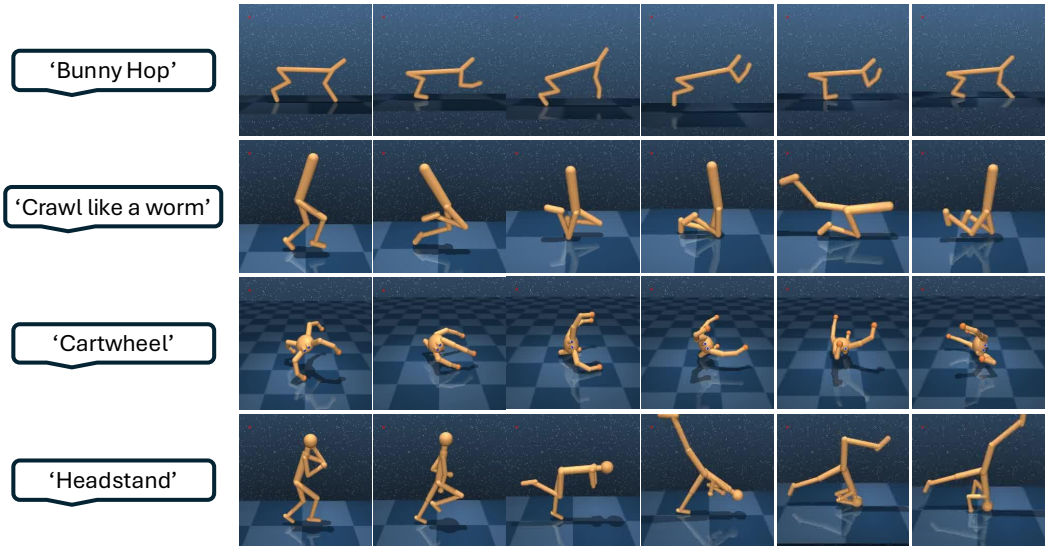


Figure 4: `RLZero` in action: Qualitative examples of RL converting the given language prompts into behaviors across different domains. Top to bottom: Cheetah, Walker, Quadruped, Stickman.

		Image-language reward		Video-language reward		GenRL	RLZero
		IQL	TD3 (Base Model)	TD3	IQL	model-based	
Walker	Lying Down	2/5 (307.04±52.60)	— (116.97±69.70)	2/5 (120.48±50.47)	5/5 (419.05±281.56)	4/5 (199.66±30.64)	5/5 (**524.34±31.78**)
	Walk like a human	1/5 (95.05±78.08)	— (30.34±34.58)	3/5 (49.49±45.73)	4/5 (146.86±54.40)	5/5 (632.09±36.31)	5/5 (**704.68±47.16**)
	Run like a human	5/5 (50.93±5.05)	— (61.76±52.49)	1/5 (100.24±89.76)	2/5 (286.14±63.91)	5/5 (316.21±82.61)	5/5 (**475.49±52.37**)
	Do lunges	4/5 (130.98±109.87)	— (67.24±24.31)	2/5 (48.71±12.06)	3/5 (217.78±147.57)	5/5 (**484.68±50.60**)	5/5 (377.53±30.14)
	Cartwheel	4/5 (**320.03±228.81**)	— (167.04±48.12)	3/5 (151.16±67.95)	4/5 (175.45±50.87)	5/5 (**252.68±33.76**)	4/5 (**254.89±12.74**)
	Strut like a horse	5/5 (110.08±50.24)	— (164.61±110.45)	1/5 (103.45±69.48)	3/5 (81.14±25.01)	5/5 (**361.45±103.07**)	5/5 (**445.66±131.36**)
	Crawl like a worm	4/5 (**116.48±72.99**)	— (**94.54±16.18**)	1/5 (**84.43±36.97**)	2/5 (**127.14±10.11**)	0/5 (**127.48±24.81**)	3/5 (**143.86±23.09**)
Quadruped	Cartwheel	1/5 (23.82±8.38)	— (6.81±3.24)	3/5 (9.45±7.43)	1/5 (14.07±5.33)	4/5 (15.6±4.07)	4/5 (**30.88±0.95**)
	Dance	5/5 (13.89±6.19)	— (8.19±4.74)	3/5 (6.24±2.34)	1/5 (11.79±4.28)	4/5 (15.11±7.71)	5/5 (**24.29±3.29**)
	Walk using three legs	2/5 (15.98±9.61)	— (6.70±3.61)	2/5 (7.96±4.97)	3/5 (7.46±1.75)	4/5 (**24.70±6.56**)	5/5 (**27.29±7.29**)
	Balancing on two legs	2/5 (13.92±5.52)	— (6.11±3.95)	2/5 (6.69±8.19)	2/5 (5.61±3.05)	5/5 (7.20±3.78)	5/5 (**22.50±2.15**)
	Lie still	1/5 (**82.93±77.13**)	— (9.51±7.54)	3/5 (6.88±8.43)	2/5 (67.96±34.91)	3/5 (**135.26±57.41**)	2/5 (**100.22±52.45**)
	Handstand	2/5 (16.13±11.66)	— (4.18±0.88)	4/5 (4.08±2.74)	2/5 (6.80±2.85)	4/5 (15.47±11.76)	3/5 (**50.65±2.92**)
Cheetah	Lie down	3/5 (219.35±58.77)	— (203.99±16.39)	2/5 (255.03±49.28)	3/5 (360.76±43.80)	3/5 (**468.52±17.45**)	2/5 (202.67±34.75)
	Bunny hop	3/5 (218.44±34.56)	— (181.74±35.52)	1/5 (192.78±53.53)	3/5 (129.26±1.37)	4/5 (**220.62±36.99**)	5/5 (**224.70±6.46**)
	Jump high	3/5 (172.10±39.02)	— (184.41±49.01)	0/5 (183.36±73.40)	5/5 (148.82±27.88)	5/5 (**267.54±28.17**)	5/5 (**232.23±41.65**)
	Jump on back legs and backflip	3/5 (175.41±38.86)	— (197.62±42.52)	0/5 (169.67±50.78)	2/5 (131.14±15.24)	5/5 (**293.28±71.71**)	5/5 (**326.20±57.45**)
	Quadruped walk	3/5 (**379.34±21.07**)	— (193.64±20.30)	3/5 (187.24±46.67)	3/5 (188.58±20.95)	2/5 (318.10±37.53)	4/5 (**388.02±37.76**)
	Stand in place like a dog	4/5 (**478.85±43.69**)	— (282.46±94.73)	3/5 (238.58±88.26)	0/5 (169.39±14.04)	2/5 (238.25±47.83)	3/5 (**469.05±31.62**)
Stickman	Lie down stable	2/5 (201.03±82.97)	— (13.21±9.27)	4/5 (30.86±4.60)	1/5 (28.89±3.49)	5/5 (**686.56±386.66**)	4/5 (**841.48±226.24**)
	Lunges	0/5 (63.46±36.27)	— (**249.73±13.81**)	2/5(48.69±25.50)	0/5 (38.77±3.69)	4/5 (**244.82±58.80**)	5/5(191.41±61.51)
	Praying	1/5 (50.84±22.27)	— (39.39±42.79)	0/5 (49.13±32.51)	0/5 (40.76±9.17)	3/5 (**192.75±42.20**)	4/5 (**147.74±54.07**)
	Headstand	2/5 (20.14±12.24)	— (14.54±7.13)	2/5 (48.11±42.51)	1/5 (13.84±6.02)	4/5 (**71.75±32.77**)	4/5 (**71.87±6.28**)
	Punch	2/5 (73.42±19.00)	— (50.23±38.08)	3/5 (74.73±7.01)	4/5 (85.76±22.96)	5/5 (**181.08±69.58**)	4/5 (**216.44±37.81**)
	Plank	0/5 (374.70±361.27)	— (507.46±289.41)	0/5 (16.19±5.70)	0/5 (66.62±53.43)	1/5 (391.04±41.15)	3/5 (**883.60±58.00**)
	Average	51.2%	Base Model	40%	44.8%	76.8%	83.2 %

Table 1: Win rates computed by GPT-4o of policies trained by different methods when compared to a base policies trained by TD3+Image-language reward. Bolded distribution-matching returns denote statistically significant improvement over the second best method under a Mann-Whitney U test with a significance level of 0.05.

random exploration is difficult with the large action-space of Stickman. The detailed composition of the datasets can be found in Appendix B.3. The behavior foundation model can be trained with any zero-shot RL method using successor features (Park et al., 2024a; Touati et al., 2023; Agarwal et al., 2024). In our experiments, we use the Forward-Backward zero-shot RL algorithm (Touati et al., 2023) trained on the same offline datasets d^O.

Baselines: For our evaluations, we consider the setting where the agent has no access to the simulator during test time. This setting truly reflects the ability of the agents to use prior exploratory data to learn meaningful behaviors. We compare state-of-the-art model-free offline RL algorithms that are capable of learning from purely offline data. For RL algorithms, the reward is obtained using embedding similarity of image observations of agent and language, or using similarity between video-encoding and language (Baumli et al., 2023; Rocamonde et al., 2023). We consider two sources of reward: Image-language cosine similarity using SigLIP embedding, and Video-language

cosine similarity using InternVideo2 embeddings. Video-language embeddings take into account context and can potentially lead to more accurate reward estimation. Once the rewards are available, we use TD3 (Fujimoto et al., 2018) and IQL (Kostrikov et al., 2021) as the representative offline RL algorithms to obtain policies. We also compare to GenRL Mazzaglia et al. (2024), the closest approach to RLZero, that performs model-based RL in the environment to learn a policy at test time by using embedding-similarity as a reward function.

5.1 Benchmarking Zero-Shot Performance for Continuous Control

The ability to specify prompts and generate agent behavior allows us to explore complex behaviors that might have required complicated reward function design. We curate a set of 25 tasks across 4 DM-control environments. Each of the agents has unique capabilities as a result of its embodiment, and the prompts are specified to be reasonable tasks to expect for the specific domain. Furthermore, we filtered out prompts for which our off-the-shelf video generation model was unable to faithfully generate videos. We discuss this more in Appendix B.2. For each prompt, we generate behaviors for 5 seeds. The performance of any given method is evaluated as the win rate over the base method. We chose the base model for our comparisons as the policies trained via TD3 on image-language rewards. For each seed, we present the observation frames that the output policy by different methods generates and pass it to a Multimodal LLM capable of video understanding, which is used as a judge. Since the number of tokens can get quite large with the long default horizon of the agent (1000 frames), we subsample the videos by choosing every 8 frames and selecting the first 64 frames of size 256×256. We observed this subsampling to retain temporal consistency and the effective horizon ($8 \times 32 = 256$) to be long enough to demonstrate the task requested by the prompt.

Table 1 demonstrates the win rates by different methods when evaluated by GPT-4o-preview. We find that RLZero achieves a win rate of 83.2% when compared to the best baseline (GenRL) which achieves a win rate of 76.8%. GenRL requires test-time learning with every task averaging $\approx$ 3 hours of training on NVIDIA-A40 GPU compared to our method which requires ≈ 25 seconds to output a policy. Figure 4 shows examples of behaviors output by RLZero on some of the prompts from our evaluation set. We also consider a more fine-grained metric for comparison – average return under the distribution-matching reward function. We learn a discriminator between the projected states for a given imagination (ρ^E) and the offline interaction dataset ρ. Under the shaped reward function (Section 4.3), $r(s) = e^\nu(s) = \rho^E/\rho$ we compute returns for all the methods; a higher return indicates that a method is better able to match the projected imaginations.

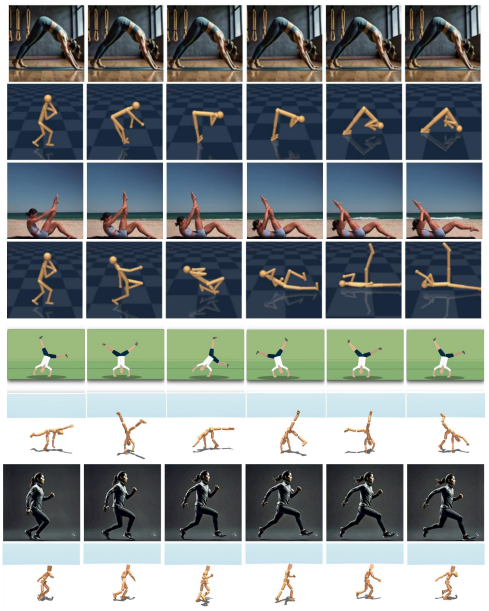


Figure 5: Examples for cross embodied imitation: RLZero can mimic motions demonstrated in YouTube or AI generated videos zero-shot. Top 2 rows: Stickman (2D Humanoid), Bottom 2 rows: SMPL 3D Humanoid.

5.2 Zero-Shot Video-to-Policy: Can RLZero succeed at cross-embodiment imitation?

The intermediate stage in RLZero of matching the closest observations in the offline dataset to a frame from a video is based on semantic similarity. This means that we are not restricted to generating videos in the same domain of the agent and still expect semantic search to generalize for out-of-domain matching. Subsequently, we can skip the *imagine* step completely if we are given an expert video demonstration. To investigate this, we consider a collection of videos scraped from Youtube as well as videos generated by open-source video generation tools like MetaAI and empirically test if RLZero is able to replicate the behaviors.

We focus on Humanoid environments here as they reflect human embodiment closely and allow us to use human videos from the internet. In addition to the Stickman (2D Humanoid) environment we also experiment with SMPL (3D Humanoid) Loper et al. (2023) using an open-source BFM Tirinzoni et al. that reflects the human morphology more accurately.

	Video Descriptions	SMODICE	`RLZero`
Stickman (2D Humanoid)	Human in backflip position	— (16.05± 3.44)	2/5 (14.27± 0.02)
	Downward facing dog yoga pose	— (15.63 ± 2.74)	1/5 (14.11±0.04)
	Cow yoga pose	— (6.72± 7.81)	**5/5** (14.99±0.02)
	Downward dog with one leg raised in the air	— (7.04±3.32)	**5/5** (15.35±0.03)
	Lying on back with one leg raised in the air	— (9.07±2.07)	**5/5** (12.55±0.01)
	Lying on back with both legs raised in the air	— (8.45± 3.71)	**5/5** (9.55±0.06)
	High plank yoga pose	— (12.83± 7.76)	**5/5** (15.08±0.03)
	Sitting down with legs laid in the front	— (12.01±1.06)	**5/5** (15.24±0.02)
	Warrior III pose	— (17.27±5.09)	**3/5** (15.22±0.01)
	Front splits	— (13.44±2.39)	**4/5** (15.21±0.01)
SMPL (3D Humanoid)	A karate kick position	— (50.02 ± 0.023)	**5/5** (199.90 ± 0.01)
	A cat doing a handstand	— (0.19 ± 0.24)	**5/5** (199.86 ± 0.02)
	An arabesque ballet position	— (10.02± 0.01)	**5/5** (199.92 ± 0.04)
	Animated wikiHow demonstration of a cartwheel	— (0.19 ± 0.24)	**5/5** (199.91 ± 0.04)
	Running	— (58.08 ± 0.46)	**5/5** (199.87±0.01)
	Lying crunches	— (0.10± 0.20)	**5/5** (199.89±0.04)
	Plank position	— (0.048 ± 0.03)	**5/5** (199.91±0.04)

Table 2: Cross Embodied Evaluation: Distribution Matching Return and Winrates

Table 2 shows the results for cross-embodied imitation across 17 video-clips. We use similar metrics to Section 5.1, but modified the GPT-4o prompt to take in the frames from the original video instead of a specified task description. We compare against SMODICE (Ma et al., 2022a) which allows for using state-only observational data in conjunction with suboptimal offline data for imitation learning. This allows us to ablate the quality of imitation produced by a successor measure-based method that uses one policy model for all tasks as opposed to SMODICE which trains a new policy for each task. GenRL requires a world model of the environment which is not available for SMPL as we only have access to the pretrained policy. `RLZero` achieves a win rate of 80% against SMODICE for Stickman and 100% for SMPL Humanoid. This matches the observation from Pirotta et al. (2023) that DICE-based methods lag behind in performance on observation-only imitation tasks. Figure 5 shows a qualitative comparison of the video and the obtained behavior on a few videos. Details about videos used can be found in Appendix C.2.

5.3 Ablation and Failure Cases

Imagination-free behavior generation: While the imagine, project, and imitate framework allows for the interpretability of the agent's behavior, we investigate if we can amortize the imagination and embedding search cost by directly mapping the language embedding to the skill embedding in the Behavior Foundation Model's latent space. For this, we consider sampling z uniformly in the latent space of the BFM and embedding the generated image observation sequence through a ViFM, which we denote by e. Given the observation sequence, we generate the z_{imit} using the zero-shot inference process and learn a mapping from $e \rightarrow z_{imit}$ using a small 3-layer MLP. On the same tasks considered in Fig 5, we observe imagination-free `RLZero` to have a win rate of 65.71% over TD3 base model on Walker environment when compared to `RLZero` that had a win rate of 91.4% (detailed results in Table 7). A more thorough explanation of imagination-free `RLZero` can be found in Appendix B.6.

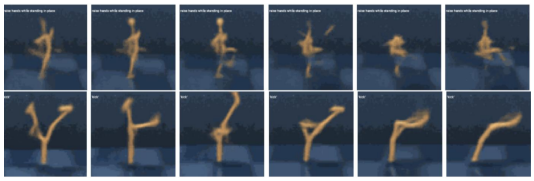
Failures: Our proposed method `RLZero` is not without failures. The stages of imagination and projection can fail individually, but the failures remain interpretable, i.e., by investigating the videos and the closest state match, we can comment on the agent's ability to faithfully complete that task to a certain extent.

1. What I cannot imagine, I cannot imitate: The video generation model used in our work from Mazzaglia et al. (2024) is fairly small and limited in capability. We encountered limitations when generating complex behaviors with this model and found it to be sensitive to prompt engineering. Fortunately, as models get bigger and are trained on a larger set of data, this limitation can be overcome. Figure 6a shows some examples of these failures with the corresponding prompts.

2. Limitation of semantic search-based image retrieval: In this work, we used SigLIP, which has shown commendable performance for image retrieval tasks. We observed failure cases in the following scenarios (e.g. Figure 6b): a) Background distractors: We observe the image-similarity to latch on to features from the background and produce incorrect retrieval; b) Rough symmetries: In tasks where the agent is roughly symmetric (e.g. Walker when the head and legs are almost identical with a slight difference in width) the image retrieval fails by giving an incorrect permutation w.r.t the rough symmetries.

6 Conclusion

Language presents an appealing and human-friendly alternative to reward design for task specification. In this work, we presented a completely unsupervised approach for grounding language to low-level behavior in a zero-shot manner. A completely unsupervised approach allows us to bypass requiring costly annotators for labeling a wide variety of behaviors with language, and a zero-shot approach allows us to avoid training during deployment time along with the advantage of generating the behaviors instantaneously. We propose `RLZero`, a framework to imagine what a behavior specified by a text prompt looks like and to ground that imagination to a policy via zero-shot imitation. Unlike methods that learn intermediate reward functions, this approach is not prone to reward hacking as the distribution matching objective specifies the task completely and accurately. Our evaluations show that the behaviors generated by `RLZero` show an improvement over using reward functions derived from image-language of video-language models.



(a) Imagined behaviors: Top: stickman: 'raise hand while standing in place', Bottom: walker: 'kick'

(b) Failed projection for a cross-embodied video.

Figure 6: Failure Cases in `RLZero`

Future Directions: `RLZero` opens up the possibility of prompting to generate a policy. Zero-shot approaches are always expected to be near-optimal due to the projection of a reward to a low dimensional space as well as limited coverage of offline interaction data. But this serves as a good initialization for further fine-tuning. How to fine-tune efficiently without forgetting remains an open question. Furthermore, learned skills can be combined according to the hierarchy specified in language instructions, allowing for the completion of complex long-horizon tasks. Since the mechanism of RLZero allows for interoperability to some extent by observing the nearest states as well as imagination, automatic failure detection becomes appealing. For the setting of prompt-to-policy, we lack accurate evaluation metrics since the true reward function is unknown, and human evaluation can be subjective. Finally, with larger context-window video understanding models, we believe an end-to-end pipeline of language embedding to task embedding (imagination-free `RLZero`) can become more appealing.

ACKNOWLEDGMENTS

We thank Matteo Pirotta, Ahmed Touati, Andrea Tirinzoni, Alessandro Lazaric and Yann Ollivier for enlightening discussion on unsupervised RL. This work has in part taken place in the Safe, Correct, and Aligned Learning and Robotics Lab (SCALAR) at The University of Massachusetts Amherst and Machine Intelligence through Decision-making and Interaction (MIDI) Lab at The University of Texas at Austin. SCALAR research is supported in part by the NSF (IIS-2323384), the Center for AI Safety (CAIS), and the Long-Term Future Fund. HS, SA, SP, MR, and AZ are supported by NSF 2340651, NSF 2402650, DARPA HR00112490431, and ARO W911NF-24-1-0193. This work has in part taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by the National Science Foundation (FAIN-2019844, NRT-2125858), the Office of Naval Research (N00014-18-2243), Army Research Office (W911NF-23-2-0004, W911NF-17-2-0181), DARPA (Cooperative Agreement HR00112520004 on Ad Hoc Teamwork), Lockheed Martin, and Good Systems, a research grand challenge at the University of Texas at Austin. The views and conclusions contained in this document are those of the authors alone. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

REFERENCES

Agarwal, S., Durugkar, I., Stone, P., and Zhang, A. f-policy gradients: A general framework for goal-conditioned RL using f-divergences. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=EhhPtGsVAv.

Agarwal, S., Sikchi, H., Stone, P., and Zhang, A. Proto successor measure: Representing the space of all possible solutions of reinforcement learning, 2024. URL https://arxiv.org/abs/2411.19418.

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H. P., and Silver, D. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Baumli, K., Baveja, S., Behbahani, F., Chan, H., Comanici, G., Flennerhag, S., Gazeau, M., Holsheimer, K., Horgan, D., Laskin, M., et al. Vision-language models as a source of rewards. *arXiv preprint arXiv:2312.09187*, 2023.

Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., Jampani, V., and Rombach, R. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023. URL https://arxiv.org/abs/2311.15127.

Blier, L., Tallec, C., and Ollivier, Y. Learning successor states and goal-dependent values: A mathematical viewpoint. *arXiv preprint arXiv:2101.07123*, 2021a.

Blier, L., Tallec, C., and Ollivier, Y. Learning successor states and goal-dependent values: A mathematical viewpoint. *arXiv preprint arXiv:2101.07123*, 2021b.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Bruce, J., Dennis, M. D., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., Lai, M., Mavalankar, A., Steigerwald, R., Apps, C., et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Calisthenicmovement. The most underrated exercise you're not doing!, 2021. URL https://youtu.be/_3p4b4jVfFU?si=poU0t9XAhDf5U_g6. YouTube video.

Caruana, R. Multitask learning. *Machine Learning*, 28:41–75, 1997. URL https://api.semanticscholar.org/CorpusID:45998148.

Chen, D., Chen, R., Zhang, S., Liu, Y., Wang, Y., Zhou, H., Zhang, Q., Wan, Y., Zhou, P., and Sun, L. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. *arXiv preprint arXiv:2402.04788*, 2024.

Chen, V., Gupta, A., and Marino, K. Ask your humans: Using human instructions to improve generalization in reinforcement learning. *arXiv preprint arXiv:2011.00517*, 2020.

Chen, X., Liu, Z., Chen, M., Feng, Y., Liu, Y., Shen, Y., and Zhao, H. Livephoto: Real image animation with text-guided motion control. In *European Conference on Computer Vision*, pp. 475–491. Springer, 2025.

Chevalier-Boisvert, M., Bahdanau, D., Lahlou, S., Willems, L., Saharia, C., Nguyen, T. H., and Bengio, Y. Babyai: First steps towards grounded language learning with a human in the loop. In *International Conference on Learning Representations*, volume 105. New Orleans, LA, 2019.

Dayan, P. Improving generalization for temporal difference learning: The successor representation. *Neural computation*, 5(4):613–624, 1993.

Dulac-Arnold, G., Levine, N., Mankowitz, D. J., Li, J., Paduraru, C., Gowal, S., and Hester, T. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021.

Durugkar, I., Tec, M., Niekum, S., and Stone, P. Adversarial intrinsic motivation for reinforcement learning. *Advances in Neural Information Processing Systems*, 34:8622–8636, 2021.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. *CoRR*, abs/1802.06070, 2018. URL http://arxiv.org/abs/1802.06070.

Eysenbach, B., Salakhutdinov, R., and Levine, S. The information geometry of unsupervised reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=3wU2UX0voE.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Garg, D., Chakraborty, S., Cundy, C., Song, J., and Ermon, S. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems*, 34:4028–4039, 2021.

Ghasemipour, S. K. S., Zemel, R., and Gu, S. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, pp. 1259–1277. PMLR, 2020.

Goyal, P., Niekum, S., and Mooney, R. J. Using natural language for reward shaping in reinforcement learning. *arXiv preprint arXiv:1903.02020*, 2019.

Goyal, P., Mooney, R. J., and Niekum, S. Zero-shot task adaptation using natural language. *arXiv preprint arXiv:2106.02972*, 2021a.

Goyal, P., Niekum, S., and Mooney, R. Pixl2r: Guiding reinforcement learning using natural language by mapping pixels to rewards. In *Conference on Robot Learning*, pp. 485–497. PMLR, 2021b.

Harrison, B., Ehsan, U., and Riedl, M. O. Guiding reinforcement learning exploration using natural language. *arXiv preprint arXiv:1707.08616*, 2017.

Ho, J. and Ermon, S. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29:4565–4573, 2016.

Hu, Y., Luo, C., and Chen, Z. Make it move: controllable image-to-video generation with text descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18219–18228, 2022.

Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., and Schmidt, L. Openclip, July 2021. URL https://doi.org/10.5281/zenodo.5143773. If you use this software, please cite it as below.

Jang, E., Irpan, A., Khansari, M., Kappler, D., Ebert, F., Lynch, C., Levine, S., and Finn, C. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pp. 991–1002. PMLR, 2022.

Jiang, Y., Gu, S. S., Murphy, K. P., and Finn, C. Language as an abstraction for hierarchical deep reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Kaelbling, L. P. Learning to achieve goals. In *International Joint Conference on Artificial Intelligence*, 1993. URL https://api.semanticscholar.org/CorpusID:5538688.

Kim, G., Seo, S., Lee, J., Jeon, W., Hwang, H., Yang, H., and Kim, K. Demodice: Offline imitation learning with supplementary imperfect demonstrations. In *ICLR*. OpenReview.net, 2022.

Kondratyuk, D., Yu, L., Gu, X., Lezama, J., Huang, J., Schindler, G., Hornung, R., Birodkar, V., Yan, J., Chiu, M.-C., et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Krakovna, V. Specification gaming examples in ai. *Available at vkrakovna. wordpress. com*, 2018.

Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020.

Liang, J., Huang, W., Xia, F., Xu, P., Hausman, K., Ichter, B., Florence, P., and Zeng, A. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500. IEEE, 2023.

Liang, W., Zhang, Y., Kwon, Y., Yeung, S., and Zou, J. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=S7Evzt9uit3.

LivestrongWoman. Lying leg raises, 2014. URL https://youtu.be/Wp4BlxcFTkE?si=Jdmjhu05kjm8pK1L. YouTube video.

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 851–866. 2023.

Ma, R., Luijkx, J., Ajanovic, Z., and Kober, J. Explorllm: Guiding exploration in reinforcement learning with large language models. *arXiv preprint arXiv:2403.09583*, 2024.

Ma, Y. J., Shen, A., Jayaraman, D., and Bastani, O. Smodice: Versatile offline imitation learning via state occupancy matching. *arXiv preprint arXiv:2202.02433*, 2022a.

Ma, Y. J., Sodhani, S., Jayaraman, D., Bastani, O., Kumar, V., and Zhang, A. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022b.

Ma, Y. J., Liang, W., Wang, G., Huang, D.-A., Bastani, O., Jayaraman, D., Zhu, Y., Fan, L., and Anandkumar, A. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023.

Mazzaglia, P., Verbelen, T., Dhoedt, B., Courville, A., and Rajeswar, S. Multimodal foundation world models for generalist embodied agents. *arXiv preprint arXiv:2406.18043*, 2024.

Moves, A. How to do downward dog — adho mukha svanasana tutorial with dylan werner, February 2019. URL https://youtu.be/EC7RGJ975iM?si=GQWepAU1zZ_IQ3mc. YouTube video.

Ng, A. Y., Russell, S. J., et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, pp. 2, 2000.

Ni, H., Shi, C., Li, K., Huang, S. X., and Min, M. R. Conditional image-to-video generation with latent flow diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18444–18455, 2023.

Ni, T., Sikchi, H., Wang, Y., Gupta, T., Lee, L., and Eysenbach, B. f-irl: Inverse reinforcement learning via state marginal matching. In *Conference on Robot Learning*, pp. 529–551. PMLR, 2021.

Nicole, M. W. 35 min full body workout —— intermediate pilates flow, 2021. URL https://youtu.be/mU0JDQItAkE?si=U51AZVvzZ8Uty8z5. YouTube video.

O'Neill, A., Rehman, A., Gupta, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., Mandlekar, A., et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

Park, S., Kreiman, T., and Levine, S. Foundation policies with hilbert representations. *arXiv preprint arXiv:2402.15567*, 2024a.

Park, S., Rybkin, O., and Levine, S. METRA: Scalable unsupervised RL with metric-aware abstraction. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=c5pwL0Soay.

Pirotta, M., Tirinzoni, A., Touati, A., Lazaric, A., and Ollivier, Y. Fast imitation via behavior foundation models. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Pylyshyn, Z. W. Mental imagery: In search of a theory. *Behavioral and brain sciences*, 25(2): 157–182, 2002.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *ICML*, 2021a.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021b.

Rocamonde, J., Montesinos, V., Nava, E., Perez, E., and Lindner, D. Vision-language models are zero-shot reward models for reinforcement learning. *arXiv preprint arXiv:2310.12921*, 2023.

Sarbin, T. R. The role of imagination. *Narrative analysis: Studying the development of individuals in society*, pp. 5, 2004.

Sarbin, T. R. and Juhasz, J. B. Toward a theory of imagination. *Journal of personality*, 38(1):52–76, 1970.

Sikchi, H., Chitnis, R., Touati, A., Geramifard, A., Zhang, A., and Niekum, S. Score models for offline goal-conditioned reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2023.

Sikchi, H., Zheng, Q., Zhang, A., and Niekum, S. Dual RL: Unification and new methods for reinforcement and imitation learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=xt9Bu66rqv.

Sontakke, S., Zhang, J., Arnold, S., Pertsch, K., Bıyık, E., Sadigh, D., Finn, C., and Itti, L. Roboclip: One demonstration is enough to learn robot policies. *Advances in Neural Information Processing Systems*, 36, 2024.

Stepputtis, S., Campbell, J., Phielipp, M., Lee, S., Baral, C., and Ben Amor, H. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.

Suarez, D. Get your splits / hip flexibility — 28 day splits challenge — 17 min — daniela suarez, 2022. URL https://youtu.be/63bhMpFZnvQ?si=G0kEk0mr0U35lC86. YouTube video.

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

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., de Las Casas, D., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., Lillicrap, T. P., and Riedmiller, M. A. Deepmind control suite. *CoRR*, abs/1801.00690, 2018.

Thomason, J., Zhang, S., Mooney, R. J., and Stone, P. Learning to interpret natural language commands through human-robot dialog. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

Tirinzoni, A., Touati, A., Farebrother, J., Guzek, M., Kanervisto, A., Xu, Y., Lazaric, A., and Pirotta, M. Zero-shot whole-body humanoid control via behavioral foundation models. In *NeurIPS 2024 Workshop on Open-World Agents*.

Tong, Z., Song, Y., Wang, J., and Wang, L. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training, 2022. URL https://arxiv.org/abs/2203.12602.

Touati, A. and Ollivier, Y. Learning one representation to optimize all rewards. In *NeurIPS*, pp. 13–23, 2021.

Touati, A., Rapin, J., and Ollivier, Y. Does zero-shot reinforcement learning exist? In *ICLR*. OpenReview.net, 2023.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., and Anandkumar, A. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z., et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.

Wang, Y., Li, K., Li, X., Yu, J., He, Y., Wang, C., Chen, G., Pei, B., Yan, Z., Zheng, R., Xu, J., Wang, Z., Shi, Y., Jiang, T., Li, S., Zhang, H., Huang, Y., Qiao, Y., Wang, Y., and Wang, L. Internvideo2: Scaling foundation models for multimodal video understanding, 2024a. URL https://arxiv.org/abs/2403.15377.

Wang, Y., Sun, Z., Zhang, J., Xian, Z., Biyik, E., Held, D., and Erickson, Z. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. *arXiv preprint arXiv:2402.03681*, 2024b.

Wang, Y., Li, K., Li, X., Yu, J., He, Y., Chen, G., Pei, B., Zheng, R., Wang, Z., Shi, Y., et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, pp. 396–416. Springer, 2025.

Well+Good. How to do a push-up — the right way — well+good, 2019a. URL https://youtu.be/bt5b9x9N0KU?si=Ge6EXcMJ7_3Vf6ev. YouTube video.

Well+Good. How to do crunches — the right way — well+good, July 2019b. URL https://youtu.be/0t4t3IpiEao?si=UBOnxa_YtcD3p-Ct. YouTube video.

wikiHow. How to do a cartwheel, September 2019. URL https://youtu.be/tkugpC33ZVI?si=b0T_tM_HEYB2xQtc. YouTube video.

Wu, Y., Tucker, G., and Nachum, O. The laplacian in rl: Learning representations with efficient approximations. *arXiv preprint arXiv:1810.04586*, 2018.

Yarats, D., Brandfonbrener, D., Liu, H., Laskin, M., Abbeel, P., Lazaric, A., and Pinto, L. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. *arXiv preprint arXiv:2201.13425*, 2022.

Yoga, J. Full body standing yoga - improve flexibility and balance, 2022. URL https://youtu.be/vAA2RS4LQe0?si=Cwe5GsE8S5g7bwO4. YouTube video.

Yu, W., Gileadi, N., Fu, C., Kirmani, S., Lee, K.-H., Arenas, M. G., Chiang, H.-T. L., Erez, T., Hasenclever, L., Humplik, J., et al. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv:2306.08647*, 2023.

Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.

APPENDIX

A PROOF FOR THEOREM 1

Theorem 1. *Define $J(\pi, r)$ to be the expected return of a policy π under reward r. For an offline dataset d^O with density ρ, a learned log distribution ratio: $\nu(s) = \log(\frac{\rho^E(s)}{\rho(s)})$, $D_{KL}(\rho^\pi, \rho^E) \leq -J(\pi, r^{imit}) + D_{KL}(\rho^\pi(s,a), \rho(s,a))$ where $r^{imit}(s) = \nu(s)\ \forall s$. The corresponding z_{imit} minimizing the upper bound is given by $z_{imit} = \mathbb{E}_\rho[r^{imit}(s)\varphi(s)] = \mathbb{E}_{\rho^E}[\frac{\nu(s)}{e^{\nu(s)}}\varphi(s)]$ where φ denotes state features learned by the BFM.*

Proof. Let ρ be the density of the offline dataset, ρ^π be the visitation distribution w.r.t. policy π and ρ^E be the expert density. The distribution matching objective mentioned in Equation 4 using KL divergence is given as:

$$\min_{\rho^\pi} D_{KL}(\rho^\pi || \rho^E) \tag{5}$$

With simple algebraic manipulation, the divergence can be simplified to,

$$\begin{aligned}
D_{KL}(\rho^\pi||\rho^E) =& \mathbb{E}_{\rho^\pi}\left[\log \frac{\rho}{\rho^E}\right] + \mathbb{E}_{\rho^\pi}\left[\log \frac{\rho^\pi}{\rho}\right] && (6)\\
=& \mathbb{E}_{\rho^\pi}\left[\log \frac{\rho(s)}{\rho^E(s)}\right] + D_{KL}(\rho^\pi(s)||\rho(s)) && (7)\\
=& -J(\pi, \log \frac{\rho^E}{\rho}) + D_{KL}(\rho^\pi(s)||\rho(s)) && (8)\\
\leq& -J(\pi, \log \frac{\rho^E}{\rho}) + D_{KL}(\rho^\pi(s,a)||\rho(s,a)) && (9)
\end{aligned}$$

The last line follows from the fact that $D_{KL}(\rho^\pi(s)||\rho(s)) \leq D_{KL}(\rho^\pi(s,a)||\rho(s,a))$.

$$\begin{aligned}
D_{KL}(\rho^\pi(s,a)||\rho(s,a)) &= \mathbb{E}_{\rho^\pi(s,a)}[\log \frac{\rho^\pi(s,a)}{\rho(s,a)}] && (10)\\
&= \mathbb{E}_{\rho^\pi(s,a)}[\log \frac{\rho^\pi(s)\pi(a|s)}{\rho(s)\pi^D(a|s)}] && (11)\\
&= \mathbb{E}_{\rho^\pi(s,a)}[\log \frac{\rho^\pi(s)}{\rho(s)}] + \mathbb{E}_{\rho^\pi(s,a)}[\log \frac{\pi(a|s)}{\pi^D(a|s)}] && (12)\\
&= \mathbb{E}_{\rho^\pi(s)}[\log \frac{\rho^\pi(s)}{\rho(s)}] + \mathbb{E}_{\rho^\pi(s,a)}[\log \frac{\pi(a|s)}{\pi^D(a|s)}] && (13)\\
&= D_{KL}(\rho^\pi(s)||\rho(s)) + \mathbb{E}_{s\sim\rho^\pi}[D_{KL}(\pi(a|s)||\pi^D(a|s))] && (14)\\
&\geq D_{KL}(\rho^\pi(s)||\rho(s)) && (15)
\end{aligned}$$

Rewriting the minimization of the upper bound of KL as a maximization problem by reversing signs, we get:

$$\max_\pi \left[J(\pi, \log \frac{\rho^E}{\rho}) - D_{KL}(\rho^\pi(s,a)||\rho(s,a))\right] \tag{16}$$

The first term is an RL objective with a reward function given by $\log(\frac{\rho^E}{\rho})$, and the second term is an offline regularization to constrain the behaviors of offline datasets. Following prior works Kim et al. (2022); Ma et al. (2022a), since our BFM is trained on an offline dataset and limited to output skills in support of dataset actions, and we can ignore the regularization to infer the latent z parameterizing the skill. A heuristic yet performant alternative is to use a shaped reward function of $\frac{\rho^E}{\rho}$, which allows us to avoid training the discriminator completely and was shown to lead to performant imitation in Pirotta et al. (2023). □

B Experimental Details

B.1 Environments

B.1.1 DM-control environments

We use continuous control environments from the DeepMind Control Suite (Tassa et al., 2018).

Walker: The agent has a 24 dimensional state space consisting of joint positions and velocities and 6 dimensional action space where each dimension of action lies in $[-1, 1]$. The system represents a planar walker.

Cheetah: The agent has a 17 dimensional state space consisting of joint positions and velocities and 6 dimensional action space where each dimension of action lies in $[-1, 1]$. The system represents a planar biped "cheetah".

Quadruped: The agent has a 78 dimensional state space consisting of joint positions and velocities and 12 dimensional action space where each dimension of action lies in $[-1, 1]$. The system represents a 3-dimensional ant with 4 legs.

Stickman: Stickman was recently introduced as a task that bears resemblance to a humanoid in Mazzaglia et al. (2024). It has a 44 dimensional observation space and a 10 dimensional action space where each dimension of action lies in $[-1, 1]$.

SMPL 3D Humanoid: The agent has a 358 dimensional state space consisting of joint positions and velocities and 69 dimensional action space where each dimension of action lies in $[-1, 1]$. The system represents a 3-dimensional humanoid.

For all the environments we consider image observations of size 64 x 64. All DM Control tasks have an episode length of 1000.

B.2 Evaluation Protocol

To evaluate models for behavior generation through language prompts, we considered a set of 4 prompts per environment. One key consideration in designing these prompts was the generative video model's capability of generating reasonable imagined trajectories. Due to computing limitations, we were restricted to using a fairly small video embedding (1 billion parameters) and generation model (43 million parameters). The interpretability of our framework allows us to declare failures before they happen by looking at the generations for imagined trajectories.

For the set of task prompts specified by language, there is no ground truth reward function and there does not exist a reliable quantitative metric to verify which of the methods perform better. Instead, since humans communicate their intents via language, humans are the best judge of whether the agent has demonstrated the behavior they intended to convey. In this work we use a Multimodal LLM as a judge, following studies by prior works demonstrating the correlation of LLMs judgment to humans (Chen et al., 2024). We use GPT-4o model as the judge, where the GPT-4o model is provided with two videos, one generated by a base method, and another generated by one of the methods we consider, and asked for preference between which video is better explained by the text prompt for the task. When inputting the videos to the judge, we randomize the order of the baseline and proposed methods to reduce the effect of anchoring bias. The prompt we use to compare the two methods is given here:

For prompt to policies:

```
response = client.chat.completions.create(
    model=MODEL,
    messages=[
    {"role": "system", "content": "For the given summarization:\
    '{task prompt}', which video is more aligned with the summarization?"},
    {"role": "user", "content": [
        "Video A",
        *map(lambda x: {"type": "image_url",
                        "image_url": {"url": f'{x}'\
                        }}, video1),
```

```
        "Video B",
        *map(lambda x: {"type": "image_url",
                        "image_url": {"url": f'{x}'\
                        }}, video2),
        "FIRST provide a one-sentence comparison of the two videos\
        and explain which you feel the given summarization explains better.\
        SECOND, on a new line, state only 'A' or
        'B' to indicate\
        which video is better explained by the given \
        summarization. Your response should use
        the format:\
        Comparison: <one-sentence comparison and explanation>\
        Better explained by summarization: <'A' or 'B'>"
        ]
        }

    ],
)
```

For cross-embodiment video to policies:

```
cross_embodied_video_description = [*map(lambda x: {"type": "image_url",
    "image_url": {"url": f'{x}'}},
    cross_embodied_video)]

response = client.chat.completions.create(
                model=MODEL,
                messages=[
                {"role": "system", "content": f"For the original video:
                '{cross_embodied_video_description}', which of the
                following given videos describe a behavior more similar
                to the original video?"},
                {"role": "user", "content": [
                    "Video A",
                    *map(lambda x: {"type": "image_url",
                                    "image_url": {"URL":
                            f'{x}',
                            }}, video1),
                    "Video B",
                    *map(lambda x: {"type": "image_url",
                                    "image_url": {"URL":
                            f'{x}',
                            }}, video2),
                    "FIRST provide a one-sentence comparison of
                    the two videos and explain \
                    which you feel matches the behavior
                    shown in original video better .
                    SECOND, on a new line, state only 'A' or \
                    'B' to indicate which video is better aligned
                    to the task demonstrated in the original video.
                    Your response should use \
                    the format:\
                    Comparison: <one-sentence comparison and explanation>\
                    Better matches the original video: <'A' or 'B'>"
                    ]
                    }

                ],
            )
```

B.3 Dataset Collection for Zero-Shot RL

For Cheetah, Walker, Quadruped, and Stickman environments, our data is collected following a pure exploration algorithm with no extrinsic rewards. In this work, we use intrinsic rewards obtained from

Random Network Distillation (Burda et al., 2018) to collect our dataset based on the protocol by ExoRL (Yarats et al., 2022) and using the implementation from repository ExoRL repository. For Cheetah, Walker, and Quadruped, our dataset comprises 5000 episodes and equivalently 5 million transitions, and for Stickman, our dataset comprises 10000 episodes or equivalently 10 million transitions. Due to the high dimensionality of action space in Stickman, RND does not discover a lot of meaningful behaviors; hence we additionally augment the dataset with 1000 episodes from the replay buffer of training for a 'running' reward function and 1000 episodes of replay buffer trained on a 'standing' reward function.

B.4 Baselines

Zero-shot text to policy behavior has not been widely explored in RL literature. However, Offline RL using language-based rewards utilizes an offline dataset to learn policies and is thus zero-shot in terms of rolling out the learned policy. This makes it a meaningful baseline to compare against. Offline RL uses the same MDP formulation as described in Section 3 to learn a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, given a reward function $r : \mathcal{S} \to \mathbb{R}$ and offline dataset $\mathcal{D}$. The offline dataset consists of state, action, next-state, reward transitions $(s, a, s', r(s))$. One of the core challenges of Offline RL is to learn a Q-function that does not overestimate the reward of unseen actions, which then at evaluation causes the agent to drift from the support of the offline dataset $\mathcal{D}$.

We implement two offline RL baselines to compare with RLZero– Implicit Q-learning (IQL, Kostrikov et al. (2021)) and Offline TD3 (TD3, Fujimoto & Gu (2021)). Both of these methods share the same offline dataset as used to learn the successor measure in RLZero, which is described in Section 5, and gathered using RND. Since these datasets are reward-free, we must still construct a reward function that provides meaningful rewards for an agent achieving the behavior that aligns with the text prompt. Formally, given language instruction $e^l \in \mathcal{E}^l$, frame stack $(o_{t-k}, o_{t-k+1}, \ldots, o_t) \in \mathcal{I}$, and embedding VLM $\phi : \mathcal{E} \to \mathcal{Z}$, which can also embed frame stacks $\phi : \mathcal{I} \to \mathcal{Z}$ (and where observations $o_i \in \mathcal{I}$, and we use o_i for this section), the reward for a corresponding language instruction and frame stack k is the cosine similarity between the stacked language embedding and the frame embedding:

$$r(o_{t-k:t}, e^l) = \frac{\phi(e^l) \cdot \phi(o_{t-k:t})}{\|\phi(e^l)\|\|\phi(o_{t-k:t})\|} \tag{17}$$

For any individual task, e^l is fixed and this is a reward function dependent on observations (as represented by a frame stack $o_{t-k:t}$). Notice that this representation closely matches that in Equation 3, but instead of finding the optimal sequence of observations, we simply compute reward as the cosine similarity between language and frames. Since the strength of the embedding space is vital to the quality of the reward function for offline RL, we evaluate two different vision-language models:

Image-language reward (SigLIP Zhai et al. (2023)): take a stack of 3 frames encode them using SigLIP, then the reward is computed as the cosine distance of the embeddings and the SigLIP embedding of language.

Video-language reward (InternVideo2 Wang et al. (2024a)): this method takes in previous frames $o_{0:t-1}$ as context and uses it to generate an embedding of the current frame observation o_t. The video encoder then takes the cosine similarity of $\phi(o_{0:t})$ and $\phi(e^l)$. This allows the reward function to provide rewards based not only on reaching certain states, but the agent exhibiting temporally extended behaviors that match the behavior. In practice, providing rewards using an image-based encoder for frame stacks can be challenging for tasks such as walking because they require context, and video-based rewards offer a way to better encode the temporal context.

B.4.1 Offline RL

Implicit Q-learning (Kostrikov et al., 2021) Implicit Q-learning builds on the classic TD error (revised in our context of language-instruction rewards):

$$L(\theta) = E_{(s,a,s',a')\sim\mathcal{D}}[(r(s, e^l) + \gamma Q_{\hat{\theta}}(s', a') - Q_\theta(s, a))^2]$$

to learn a Q function Q_θ. IQL builds on this loss to handle the challenge of ensuring that the Q-values do not speculate on out-of-distribution actions while also ensuring that the policy is able to exceed the performance of the behavior policy. Exceeding the behavior policy is important because the dataset is

collected using RND, meaning that any particular trajectory from the dataset is unlikely to perform well on a language reward. The balance of performance is achieved by optimizing the objective with expectile regression:

$$L_2^\tau(u) = |\tau - \mathbb{1}(u < 0)|u^2$$

Where $\tau > 0.5$ is the selected expectile. Expectile regression gives greater weight to the upper expectiles of a distribution, which means that the Q function will focus more on the upper values of the Q function.

Rather than optimize the objective with $Q(s', a')$ directly, IQL uses a value function to reduce variance to give the following objectives:

$$L_V(\psi) = E_{(s,a)\sim D}[L_2^\tau(Q_\theta(s,a)V_\psi(s))]$$

$$L_Q(\theta) = E_{(s,a,s',a')\sim\mathcal{D}}[L_2^\tau(r(s,e^l) + V_\psi(s') - Q_\theta(s,a))]$$

Using the Q-function, a policy can be extracted using advantage weighted regression:

$$L(\phi) = E_{(s,a)\sim\mathcal{D}}[\exp(\beta(Q_\theta(s,a) - V_\psi(s)))\log\pi_\phi(a|s)].$$

Where β is the inverse temperature for the advantage term.

TD3 (Fujimoto et al., 2018):

TD3 was demonstrated to be the best performing algorithm when learning from exploratory RND datasets in (Yarats et al., 2022). While TD3 does not explicitly address the challenges discussed in implicit Q-learning and learns using Bellman Optimality backups, the approach is simple and works well in practice. The algorithm uses a deterministic policy extraction $\pi : \mathcal{S} \to \mathcal{A}$ to give the following objective:

$$\pi = \arg\max_\pi E_{(s,a)\sim\mathcal{D}}[Q(s,\pi(s))]$$

B.5 RLZero

B.5.1 Text to Imagined Behavior with Video models

To generate a proposed video frame sequence, we utilize the GenRL architecture and provide the workflow using equations from the original paper (Mazzaglia et al., 2024). First, the desired text prompt is embedded with the underlying video foundation model InternVideo2 (Wang et al., 2024a) $e^{(l)} = f_{PT}^{(l)}(y)$. These embeddings are then repeated n_{frames} times (we use $n_{frames} = 32$) to match the temporal structure expected by the world model. The repeated text embeddings are passed through an aligner module $e(v) = f_\psi(e^{(l)})$. The aligner is implemented as a UNet and it is used to address the multimodality gap (Liang et al., 2022) when embeddings from different modalities occupy distinct regions in the latent space. Next, the aligned video embeddings are concatenated with temporal embeddings. The temporal embeddings are one-hot encodings of the time step modulo n_{frames} providing frame-level positional information. The first embedding is passed to the world model connector $p_\psi(s_t|e)$ to initialize the latent state. For each subsequent time step, the sequence model $h_t = f_\phi(s_{t-1}, a_{t-1}, h_{t-1})$ (implemented as a GRU) updates the deterministic state h_t. The deterministic state h_t is mapped to a stochastic latent state (s_t) using the dynamics predictor $p_\phi(s_t|h_t)$. The dynamics predictor, implemented as an ensemble of MLPs, predicts the sufficient statistics (mean and standard deviation) for a Normal distribution over s_t. During inference, the mean of this distribution is used as the latent state. Finally, the latent state s_t is passed to a convolutional decoder $p_\phi(x_t|s_t)$ to reconstruct the video frame x_t. This process is repeated for all time steps ($t = 1, ..., n_{frames}$).

B.5.2 Grounding imagined observations to observations in offline dataset

As described in Section 4, we ground imagined sequences by retrieving real offline states based on similarity in an embedding space. This enables us to create a suitable z-vector for distribution matching which is the expected value of the state features under the distribution of imagined states ($\rho_{imagined}$). During our dataset collection phase, we save both the agent's proprioceptive state as well as the corresponding rendered images and search over the images to then find the corresponding state. Our code supports both stacked-frame embeddings and single-frame embeddings. We find that

stacked-frame embeddings were helpful in modeling temporal dependencies through velocity and acceleration, which are crucial for recreating the intended behavior. SigLIP (Zhai et al., 2023), which replaces CLIP's (Radford et al., 2021a) softmax-based contrastive loss with a pairwise sigmoid loss, resulted in qualitatively better matches to exact positions within sequences, imitating behavior more accurately than CLIP. For both models, we use the OpenCLIP (Ilharco et al., 2021) framework. Our matching process first involves precomputing embeddings offline, which are stored in chunks of up to 100,000 frames to optimize memory usage and retrieval speed. During inference, we load this file and embed the query frame sequence from GenRL (Mazzaglia et al., 2024) into the same latent space. We process these query embeddings by dividing them into chunks of k-frame sequences (k is generally 3 or 5), where each sequence consists of the current frame and the $k-1$ preceding frames. If there are not enough preceding frames, we repeat the first frame to fill the gap. For each chunk of saved embeddings, we compute dot products between the query chunk and all subsequences of size k in the saved embeddings. We track the highest similarity score for each query chunk and return the frames corresponding to the closest embedding sequences.

B.5.3 Training a zero-shot RL agent

In this work, we chose Forward-Backward (FB) (Touati & Ollivier, 2021) as our zero-shot RL algorithm and trained it on proprioceptive inputs. Our implementation follows closely from the author's codebase . Specifically, FB trains Forward, Backward, and Actor networks. The backward networks are used to map a demonstration or a reward function to a skill, which is then used to learn a latent-conditional Actor. The hyperparameters for our FB implementation are listed below:

Implementation: We build upon the codebase for FB `https://github.com/facebookresearch/controllable_agent` and implement all the algorithms under a uniform setup for network architectures and the same hyperparameters for shared modules across the algorithms. We keep the same method-agnostic hyperparameters and use the author-suggested method-specific hyperparameters. The hyperparameters for all methods can be found in Table 3:

Table 3: Hyperparameters for zero-shot RL with FB.

Hyperparameter	**Value**
Replay buffer size	5×10^6, 10×10^6 (for stickman)
Representation dimension	128
Batch size	1024
Discount factor γ	0.98
Optimizer	Adam
Learning rate	3×10^{-4}
Momentum coefficient for target networks	0.99
Stddev σ for policy smoothing	0.2
Truncation level for policy smoothing	0.3
Number of gradient steps	2×10^6
Regularization weight for orthonormality loss (ensures diversity)	1
FB specific hyperparameters	
Hidden units (F)	1024
Number of layers (F)	3
Hidden units (b)	256
Number of layers (b)	2

B.6 Imagination-Free RLZero

In this section, we propose an alternate method (Figure 7) for mapping a task description into a usable policy. Instead of first embedding a text prompt e^ℓ, generating a video, then mapping the video to a policy parametrization, we propose to map the text prompt directly to a policy parametrization. To do this, we learn a latent mapper $m : e \to z_{\text{imitation}}$ that relates the latent space of a ViLM to the latent space of our policy parametrization. The mapper is a 3 layer MLP with hidden size of 512.

Pretraining: We first generate a dataset of episodes containing diverse behaviors by rolling out the behavior foundation model conditioned on a uniformly random sampled z. The resulting image

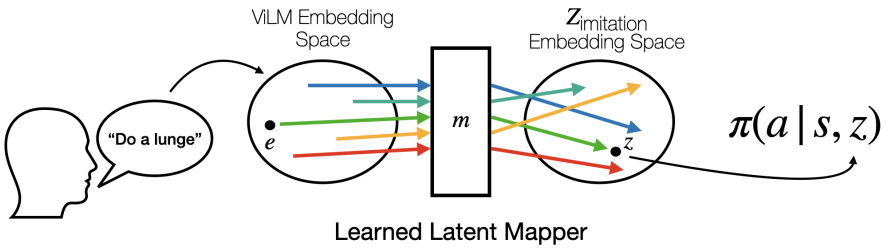


Figure 7: Illustrative diagram of imagination-free `RLZero` inference

observation sequences are then down-sampled (by 8) and sliced to break up each episode into smaller chunks of length 8; this preprocessing step helps increase the behavioral diversity and improves the ability of the ViFM to capture semantic meaning. The resulting clips are then embedded using a ViFM (InternVideo2 (Wang et al., 2024a)) where each embedding is denoted by e (as in Section 5.3). Now, we have a set of sequences of length 8 consisting of image observation along with their proprioceptive states, and the embedding for the image sequence.

Now, an obvious option is to map the embedding of image sequence to the z that generated the trajectory. Unfortunately, the way BFMs are trained, they do not account for optimal policy invariance to reward functions. That is multiple reward functions that induce the same optimal policy are mapped to different encodings in the $\mathcal{Z}$-space. This presents a problem for the latent mapper, as it becomes a one-to-many mapping for any language encoding. We present an alternative solution which ensures that only one target z is used for a given distribution of states induced by a language encoding. To achieve this we turn back to the imitation learning objective where the sequence of proprioceptive states is used to obtain a policy representation using Lemma 1 which gives the latent z corresponding to the policy that minimizes the distribution divergence to the sequence of given states. We refer to the policy representation embedding space from the Forward-Backward representation as $\mathcal{Z}_{\text{imitation}}$-space.

When optimizing the latent mapper m, we minimize the following loss:

$$\mathcal{L}(\mathcal{D}, m) = \mathbb{E}_{(z_{\text{imitation}}, e)\sim\mathcal{D}}\left[-\frac{m(e)\cdot z_{\text{imitation}}}{\|m(e)\|\cdot\|z_{\text{imitation}}\|}\right]$$

The latent space of the Backward representation is aligned with the latent space of the policy parametrization, so learning a mapping from the ViFM space to the Backward space is equivalent to learning a mapping from the ViFM space to the policy parametrization space.

Inference: During inference, the language prompt is embedded to a latent vector e^l. A known issue with multimodal embedding models is the embedding gap (Liang et al., 2022), which makes the video embeddings unaligned with text embeddings. To account for this gap, we use an aligner trained in an unsupervised fashion from previous work Mazzaglia et al. (2024) to align the language embedding ($e^l_{aligned}$). Then the aligned embedding is passed through the latent mapper to get the policy conditioning $z_{imitation}$ which gives us the policy that achieves the desired behavior specified through language.

C Additional Results

C.1 Zero-shot imitation: Discriminator vs Discriminator-free

We experiment whether optimizing a tighter bound to KL divergence at the expense of training an additional discriminator in Lemma 1 leads to performance improvements. Table 4 shows that using a discriminator does not lead to a performance improvement and a training-free inference time solution achieves a slightly higher win rate.

	RLZero with discriminator	RLZero
Walker		
Lying Down	**5/5**	**5/5**
Walk like a human	**5/5**	**5/5**
Run like a human	**5/5**	**5/5**
Do lunges	**5/5**	**5/5**
Cartwheel	**5/5**	4/5
Strut like a horse	**5/5**	**5/5**
Crawl like a worm	1/5	**3/5**
Quadruped		
Cartwheel	**4/5**	**4/5**
Dance	**5/5**	**5/5**
Walk using three legs	4/5	**5/5**
Balancing on two legs	4/5	**5/5**
Lie still	**2/5**	**2/5**
Handstand	**4/5**	3/5
Cheetah		
Lie down	1/5	**2/5**
Bunny hop	**5/5**	**5/5**
Jump high	**5/5**	**5/5**
Jump on back legs and backflip	**5/5**	**5/5**
Quadruped walk	2/5	**4/5**
Stand in place like a dog	**4/5**	3/5
Stickman		
Lie down stable	**5/5**	4/5
Lunges	**5/5**	**5/5**
Praying	**4/5**	**4/5**
Headstand	**5/5**	4/5
Punch	**4/5**	**4/5**
Plank	**4/5**	3/5
Average	82.4%	**83.2**%

Table 4: Win rates computed by GPT-4o of policies trained by different methods when compared to base policies trained by TD3+Image-language reward.

C.2 Cross Embodiment Experiments

Tables 5 and 6 describe the videos used for cross-embodiment along with the win rate of the behaviors generated by `RLZero` when compared to a base model which trains SMODICE (Ma et al., 2022a) on the nearest states found with the same grounding methods as `RLZero`.

	Prompt Descriptions	Video Link/Meta AI Prompt	Win rate vs SMODICE
Stickman (2D Humanoid)	Human in backflip position	animated human trying backflip	2/5
	Downward facing dog yoga pose	right profile of yoga pose downward facing dog	1/5
	Cow yoga pose	Moves (2019)	5/5
	Downward dog with one leg raised in the air	Moves (2019)	5/5
	Lying on back with one leg raised in the air	Nicole (2021)	5/5
	Lying on back with both legs raised in the air	LivestrongWoman (2014)	5/5
	High plank yoga pose	Well+Good (2019a)	5/5
	Sitting down with legs laid in the front	Calisthenicmovement (2021)	5/5
	Warrior III pose	Yoga (2022)	3/5
	Front splits	Suarez (2022)	4/5

Table 5: Comparison of Win rates vs SMODICE for Stickman

C.3 Imagination-Free RLZero Complete Results

We consider an ablation of our method by understanding the need for imagination by replacing the step with an end-to-end learning alternative. This is a novel baseline described in Appendix B.6. Table 7 shows the results of this end-to-end alternative which maps the shared latent space of video language models to behavior policy.

	Prompt Descriptions	Video Link/Meta AI Prompt	Win rate vs SMODICE
SMPL (3D Humanoid)	A karate kick position	a karate kick	5/5
	A cat doing a handstand	a side profile of cat doing headstand	5/5
	An arabesque ballet position	ballet movement	5/5
	Animated wikiHow demo of a cartwheel	wikiHow (2019)	5/5
	Running	running	5/5
	Lying crunches	Well+Good (2019b)	5/5
	Plank position	Well+Good (2019a)	5/5

Table 6: Comparison of Win rates vs SMODICE for 3D SMPL Humanoid

Environment/Task	RLZero	RLZero (Imagination-Free)
Walker		
Lying Down	**5/5**	**5/5**
Walk like a human	**5/5**	4/5
Run like a human	**5/5**	1/5
Do lunges	**5/5**	**5/5**
Cartwheel	4/5	**5/5**
Strut like a horse	**5/5**	3/5
Crawl like a worm	3/5	0/5

Table 7: Win rates computed by GPT-4o of policies trained by different methods when compared to base policies trained by TD3+Image-language reward. `RLZero` shows marked improvement over using embedding cosine similarity as reward functions.

C.4 More Failure Cases

We include more failure cases in Figure 8 and Figure 9 as they can help in understanding the limitations of `RLZero` better and may inform future work.

C.5 RLZero evaluation with Video-Embedding similarity

In this section, we experiment with another metric for comparison – embedding similarity between a video of the generated behavior and the text. We use InternVideo2 to embed the videos and take the cosine similarity with the prompt used to generate the behavior. Table 8 shows the results for this metric of comparison. Unfortunately, we observed that the similarity score is frequently higher even for behaviors that differ significantly from the prompt. This points to a limitation of using this metric for evaluation. Some reasons for this failure could be the limited context length of 8 for the video embedding model or a misalignment between video and text embedding vectors (Liang et al., 2022).

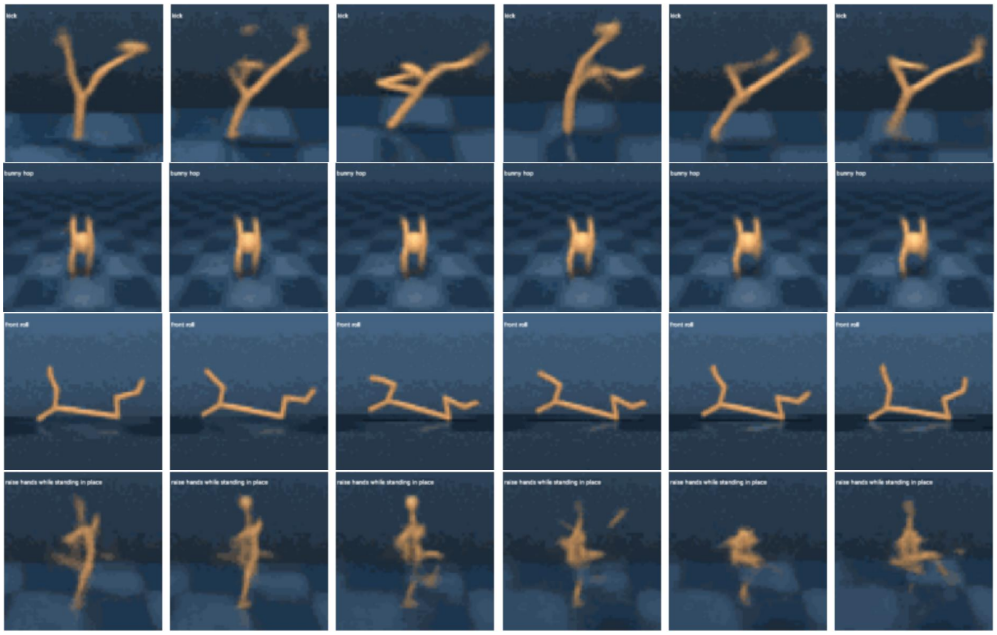


Figure 8: More examples of failed imagination by the video generation model used in RLZero. From top to bottom: Walker - 'kick', Quadruped - 'bunny hop', Cheetah - 'frontroll', Stickman - 'raise hands while standing in place'

	Image-language reward		Video-language reward		RLZero
	IQL	TD3 (Base Model)	TD3	IQL	
Walker					
Lying Down	2/5 (0.95±0.00)	- (0.89±0.02)	2/5 (0.93±0.01)	**5/5** (0.94±0.01)	**5/5** (0.93±0.00)
Walk like a human	1/5 (0.93±0.00)	- (0.83±0.02)	3/5 (0.92±0.01)	4/5 (0.94±0.00)	**5/5** (0.98±0.00)
Run like a human	**5/5** (0.95±0.02)	- (0.88±0.03)	1/5 (0.91±0.01)	2/5 (0.94±0.00)	**5/5** (0.96±0.00)
Do lunges	4/5 (0.94±0.01)	- (0.91±0.02)	2/5 (0.92±0.00)	3/5 (0.93±0.00)	**5/5** (0.94±0.01)
Cartwheel	**4/5** (0.95±0.01)	- (0.93±0.01)	3/5 (0.94±0.01)	**4/5** (0.96±0.01)	**4/5** (0.95±0.01)
Strut like a horse	**5/5** (0.96±0.00)	- (0.94±0.02)	1/5 (0.94±0.00)	3/5 (0.96±0.03)	**5/5** (0.96±0.00)
Crawl like a worm	**4/5** (0.93±0.00)	- (0.92±0.01)	1/5 (0.92±0.01)	2/5 (0.95±0.01)	3/5 (0.89±0.01)
Quadruped					
Cartwheel	1/5 (0.95±0.00)	- (0.95±0.00)	3/5 (0.95±0.01)	1/5 (0.95±0.01)	**4/5** (0.92±0.02)
Dance	**5/5** (0.94±0.00)	- (0.94±0.00)	3/5 (0.94±0.02)	1/5 (0.94±0.01)	**5/5** (0.93±0.01)
Walk using three legs	2/5 (0.92±0.00)	- (0.91±0.00)	2/5 (0.91±0.01)	3/5 (0.93±0.01)	**5/5** (0.93±0.01)
Balancing on two legs	2/5 (0.93±0.01)	- (0.93±0.00)	2/5 (0.93±0.01)	2/5 (0.93±0.00)	**5/5** (0.94±0.02)
Lie still	1/5 (0.87±0.00)	- (0.90±0.01)	**3/5** (0.94±0.00)	2/5 (0.95±0.00)	2/5 (0.92±0.00)
Handstand	2/5 (0.91±0.01)	- (0.91±0.02)	**4/5** (0.92±0.01)	2/5 (0.94±0.00)	3/5 (0.91±0.00)
Cheetah					
Lie down	**3/5** (0.92±0.02)	- (0.87±0.00)	2/5 (0.94±0.00)	**3/5** (0.94±0.01)	2/5 (0.90±0.01)
Bunny hop	3/5 (0.98±0.00)	- (0.98±0.00)	1/5 (0.98±0.00)	3/5 (0.97±0.02)	**5/5** (0.96±0.00)
Jump high	3/5 (0.94±0.01)	- (0.94±0.01)	0/5 (0.94±0.01)	**5/5** (0.93±0.01)	**5/5** (0.93±0.01)
Jump on back legs and backflip	3/5 (0.93±0.01)	- (0.92±0.00)	0/5 (0.91±0.01)	2/5 (0.92±0.01)	**5/5** (0.91±0.01)
Quadruped walk	3/5 (0.96±0.02)	- (0.85±0.01)	3/5 (0.98±0.00)	3/5 (0.99±0.01)	**4/5** (0.97±0.01)
Stand in place like a dog	**4/5** (0.93±0.01)	- (0.88±0.00)	3/5 (0.98±0.01)	0/5 (0.98±0.00)	3/5 (0.97±0.00)
Stickman					
Lie down stable	2/5 (0.92±0.00)	- (0.91±0.01)	**4/5** (0.93±0.00)	1/5 (0.93±0.00)	**4/5** (0.91±0.00)
Lunges	0/5 (0.92±0.00)	- (0.93±0.02)	2/5(0.92±0.01)	0/5 (0.92±0.00)	**5/5**(0.96±0.00)
Praying	1/5 (0.85±0.00)	- (0.89±0.02)	0/5 (0.87±0.01)	0/5 (0.87±0.01)	**4/5** (0.91±0.00)
Headstand	2/5 (0.90±0.01)	- (0.90±0.00)	2/5 (0.90±0.01)	1/5 (0.87±0.01)	**4/5** (0.90±0.00)
Punch	2/5 (0.89±0.02)	- (0.88±0.02)	3/5 (0.88±0.02)	**4/5** (0.91±0.00)	**4/5** (0.90±0.02)
Plank	0/5 (0.90±0.01)	- (0.93±0.03)	0/5 (0.89±0.01)	0/5 (0.93±0.00)	**3/5** (0.96±0.00)
Average	51.2% (0.926)	Base Model (0.908)	40% (0.927)	44.8% (0.936)	**83.2**% (0.933)

Table 8: Win rates computed by GPT-4o of policies trained by different methods when compared to a base policies trained by TD3+Image-language reward. RLZero shows marked improvement over using embedding cosine similarity as reward functions.

Figure 9: More examples of failed grounding by the image retrieval model used in `RLZero`. The top image shows the imagined frame or frame from the embodied video, and the bottom is the nearest frame obtained from the agent's prior interaction dataset.