Proceedings of Machine Learning Research 260, 2024 ACML 2024

Exploring Beyond Curiosity Rewards: Language-Driven Exploration in RL

Nicolas Bougie NICOLAS.BOUGIE@WOVEN.TOYOTA
Narimasa Watanabe NARIMASA.WATANABE@WOVEN.TOYOTA
Woven by Toyota, Tokyo, Japan

Editors: Vu Nguyen and Hsuan-Tien Lin

Abstract

Sparse rewards pose a significant challenge for many reinforcement learning algorithms, which struggle in the absence of a dense, well-shaped reward function. Drawing inspiration from the curiosity exhibited in animals, intrinsically-driven methods overcome this drawback by incentivizing agents to explore novel states. Yet, in the absence of domain-specific priors, sample efficiency is hindered as most discovered novelty has little relevance to the true task reward. We present iLLM, a curiosity-driven approach that leverages the inductive bias of foundation models — Large Language Models, as a source of information about plausibly useful behaviors. Two tasks are introduced for shaping exploration: 1) action generation and 2) history compression, where the language model is prompted with a description of the state-action trajectory. We further propose a technique for mapping state-action pairs to pretrained token embeddings of the language model in order to alleviate the need for explicit textual descriptions of the environment. By distilling prior knowledge from large language models, iLLM encourages agents to discover diverse and human-meaningful behaviors without requiring direct human intervention. We evaluate the proposed method on BabyAI-Text, MiniHack, Atari games, and Crafter tasks, demonstrating higher sample efficiency compared to prior curiosity-driven approaches.

Keywords: deep reinforcement learning; curiosity-driven exploration; curiosity

1. Introduction

Given an agent without prior knowledge of the environment, a long-standing problem is: what should the agent learn first? In reward-dense environments, the agent receives a continuous gradient signal that guides learning through interactions. When rewards are sparse or delayed, standard reinforcement learning (RL) algorithms struggle because of reliance on simple action entropy maximization as a source of exploration behavior. As a result, sample efficiency remains a major bottleneck in applying RL to real-world problems.

Various techniques were proposed to achieve better explorative policies. Intrinsically motivated RL methods answer this question by augmenting extrinsic rewards with auxiliary objectives based on novelty, surprise, or progress Burda et al. (2019a,b); Bougie and Ichise (2020a). Agents may also be rewarded in proportion to the prediction errors or information gains of a predictive world model Pathak et al. (2017). Such formulations take inspiration from cognitive sciences, with several psychological studies showcasing the role of novelty in children's curious exploration. However, they suffer from a number of pitfalls Burda et al. (2019a). A notable issue is the lack of human supervision for solving the task, encouraging the discovery of behaviors that are unlikely to correspond to any human-meaningful behaviors Du et al. (2023). In other words, it is not sufficient for intrinsically-driven agents to optimize for novelty alone — learned behaviors must also be useful.

In this study, we explore the potential of large language models (LLMs) to overcome these barriers by encouraging the discovery of behaviors that are both novel and pragmatically useful. Our hypothesis is that LLMs, by distilling prior knowledge about the task, can direct agents toward more valuable behaviors. Combining RL and language models has been employed in a few recent studies. A strategy involves rewarding an agent for achieving goals suggested by a language model Du et al. (2023). LLM may also be used to predict future text and image representations, and learn to act from imagined model rollouts Lin et al. (2023). Most language-conditioned RL methods primarily learn to generate actions from task-specific instructions — taking a goal description such as "pick up the red key" as an input and outputting a sequence of motor controls Klissarov et al. (2023). However, LLMs are prone to incorrect assumptions and thus suffer from brittle, degraded performance. Unlike most prior studies that directly perform actions/instructions recommended by a language model, we rely on language-driven rewards as a drive to explore, which is critical to better-than-expert performance.

We present, **i**ntrinsic exploration driven by **L**arge **L**anguage **M**odels (iLLM), an approach that leverages pretrained language models as a novelty signal, encouraging exploration of diverse and human-meaningful behaviors. LLMs are probabilistic models of text trained on extensive text corpora, their predictions encode rich information about human common-sense knowledge and cultural conventions. Concretely, our method prompts an LLM with an action generation task given a description of a short state-action trajectory and rewards the agent when its actions align with the LLM's predictions. We also incorporate a history compression task, designed to capture long-term meaningful behaviors, and help the acquisition of a robust representation of the environment by discarding irrelevant details from state-action pairs. We further propose a technique based on Hopfield networks Ramsauer et al. (2020) to align state-action pairs from any modality with the input space of the LLM — token embeddings, bypassing the need for explicit textual description of the environment. We evaluate iLLM on challenging sparse-reward RL problems, including BabyAI-Text, MiniHack, Atari games, and Crafter. Experimental results show that iLLM outperforms state-of-the-art exploration methods, demonstrating the benefits of considering LLM-driven exploration compared to prior curiosity-driven methods.

2. Related Work

2.1. Language Models in Reinforcement Learning

Several studies have attempted to combine language models and RL. In language-conditioned RL, an instruction-following agent learns a policy that executes actions in an environment in order to follow a language instruction Luketina et al. (2019). A line of work aims to shape the agent's exploration through the utilization of LLMs. LLMs trained on huge datasets were shown to exhibit impressive abilities along with fast adaptation to a wide range of downstream tasks from vision Yuan et al. (2021) to cross-modalities Ramesh et al. (2021); Alayrac et al. (2022). Such abilities have been utilized to provide rewards to RL agents, such as done by Gupta et al. Gupta et al. (2022) and Fan et al. Fan et al. (2022), where

CLIP is employed to generate a novelty signal. In contrast with those methods, iLLM can utilize any LLM and environment, as it learns a mapping between observations and the embedding space of the LLM.

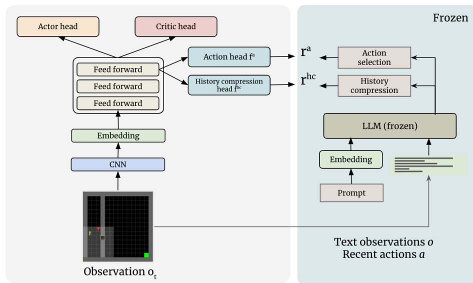
In a different spirit, an LLM may serve as a high-level supervisor, providing guidance when needed. For instance, in SayCan Ahn et al. (2022) and Inner Monologue Huang et al. (2022), an LLM provides natural language actions that are both feasible and contextually appropriate, supplying high-level semantic knowledge about the task. Nevertheless, those techniques do not have a way to directly take actions in embodied environments, or of knowing what is happening in an environment. To solve this issue, a recent study Dasgupta et al. (2023) has grafted novel components onto the agent model referred to as a reporter observing the environment and reporting useful information to the planner.

In the absence of grounding, the discrepancy between the actions/observations and internal representation of the LLM may limit its performance. Thus, several works have proposed to first finetune LLMs on expert trajectories before using them in the environment. A recent work Wang et al. (2022) has demonstrated that agents that learn interactively in a grounded environment are more sample and parameter-efficient than LLMs that learn offline by reading text from static sources. Similarly, ChibiT Reid et al. (2022) overcomes the need for symbol grounding with an extension of positional embeddings, embedding similarity encouragement. In our study, state-action alignment with the LLM's embedding space is performed during the policy training phase via a Hopfield module. Hopfield networks have been employed in HELM for state history aggregation Paischer et al. (2022), but they apply these to state representation learning rather than as intrinsic rewards for RL. Notably, iLLM seeks to align state-action pairs via a Hopfield module, and then feeds into a pretrained LLM the aligned representation in order to bias exploration towards plausibly useful behaviors.

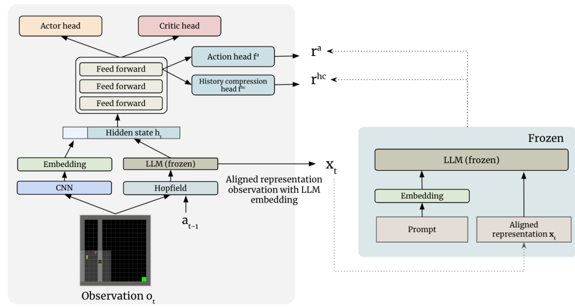
An alternative strategy is text pretraining, where LLMs can help learners automatically recognize sub-goals and learn modular sub-policies from unlabelled demonstrations Sharma et al. (2021). LLMs have also served as proxy reward functions when prompted with desired behaviors Kwon et al. (2023). In ChibiT Reid et al. (2022), the agent is trained with an objective that maximizes the similarity between language embeddings and observation embeddings. In contrast, iLLM leverages pretrained LLMs to constrain exploration towards meaningful behaviors in a task-agnostic manner. It does not assume demonstrations or task-specific prompts. Instead of directly generating actions or sub-goals, one could potentially craft a proxy reward by querying a language model to rank observations based on their relevance to achieving the final goal Klissarov et al. (2023). Nonetheless, it remains unclear how to generalize such approaches to more complex tasks without a clear skill decomposition. A similar study to our work is ELLM Du et al. (2023), which rewards an agent for achieving goals suggested by a language model prompted with a description of the agent's current state. However, the authors assume access to a text-based representation of the environment and the ability to measure if a goal was achieved.

2.2. Curiosity-Driven Exploration

Drawing inspiration from animal curiosity, intrinsic motivation encourages agents to learn about their environments even with sparse or delayed extrinsic feedback. In recent years, several model-based approaches have been proposed. The well-known ICM algorithm



(a) iLLM(obs)



(b) iLLM(hop)

Figure 1: Architecture of iLLM using text observations (left), and iLLM employing a Hopfield module to align state-action pairs with the LLM's token embeddings (right). The latter feeds the current observation and previous action into a Hopfield module, followed by the LLM. The aligned representation of state-action pairs Z_h is then used: 1) as input of the policy, and 2) along with the embedded representation of a prompt Z_p into the frozen LLM for action generation and history compression tasks. Intrinsic rewards r^a and r^{hc} are computed based on the distance between the LLM output and the prediction of an action head f^a and history compression head f^{hc}, respectively.

Pathak et al. (2017) relies on predicting environment dynamics using an inverse-forward dynamic model. To deal with the undesirable stochasticity issue Burda et al. (2019a), RND Burda et al. (2019b) introduces an exploration reward using a prediction problem where the answer is a deterministic function of its inputs. Another class of exploration methods seeks to maximize the diversity of skills mastered by the agent Bougie and Ichise (2020b). Nevertheless, maximizing state diversity also drives learning towards behaviors that lack relevance to downstream tasks Du et al. (2023). Humans do not explore solution spaces uniformly, but instead rely on their common sense to explore plausibly relevant behaviors first. iLLM addresses these shortcomings by constraining the exploration space based on prior assumptions derived from a pretrained LLM, imitating the way humans explore.

3. Method

Our approach, iLLM, distills a pretrained LLM to guide exploration. Specifically, we consider partially observable Markov decision processes (POMDPs) defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \Omega, \mathcal{T}, \gamma, \mathcal{R})$, in which an observation $o \in \mathcal{O}$ derives from environment state $s \in \mathcal{S}$ and an action $a \in \mathcal{A}$ via $\mathcal{O}(o|s, a)$. $\mathcal{T}(s^{'}|s, a)$ describes the dynamics of the environment while $\mathcal{R}$ and γ refer to the environment's reward function and discount factor, respectively. iLLM agents optimize for an intrinsic reward $\mathcal{R}_{int}$ alongside of $\mathcal{R}$. At each time step t, our method produces an intrinsic reward b_t, which is further summed up with the extrinsic reward r_t to give an augmented reward $r^*_t = r_t + b_t$. As the intrinsic reward function $\mathcal{R}_{int}$ is designed to be more dense and well aligned with $\mathcal{R}$, it accelerates the agent's learning.

One key question is how should we choose $\mathcal{R}_{int}$ to drive the agent's learning? As mentioned above, the intrinsic reward function should prioritize the exploration of plausibly useful behaviors first while maintaining some degree of diversity. Here, we leverage language-based action generation and history compression as a measure of curiosity. Similar to how next-token prediction allows language models to form internal representations of world knowledge Devlin et al. (2018), we postulate that generating the next action and a summary of the agent's history provides a rich learning signal for agents to understand language and how it relates to the world.

3.1. LLM-driven Curiosity

LLMs broadly fall into three categories: autoregressive, masked, and encoder-decoder models. Autoregressive models such as GPT are trained to maximize the log-likelihood of the next word given the previous words, in a step-by-step, or autoregressive, fashion. In our work, we employ a frozen autoregressive LLM as a proxy reward function that takes in a prompt and outputs a string. The prompt is a concatenation of two components including a description of recent state-action pairs and a user-specified question to the LLM. As user-specified questions, we introduce two types of prompts: action generation, and history compression. In the latter, the LLM is prompted to summarize state-action pairs (see Figure 1). Using the generated string, the agent derives its own intrinsic motivation, guiding it toward human-meaningful and diverse regions of the environment.

Namely, the input to the LLM is the concatenated multimodal tokens $[Z_h, Z_p]$, where Z_p are the text embeddings, tokenized from text prompts (e.g., *select the next action*). Given $[Z_h, Z_p]$, the LLM computes the (log) probability of each answer token in an autoregressive fashion as shown below:

$$p(Z_a|Z_h, Z_p) = \prod_{i=1}^{L} p_\theta(z_i|Z_h, Z_p, Z_{a,<i}), \tag{1}$$

where θ is the set of the LLM's parameters, Z_a is the generated answer, $Z_{a,<i}$ are the answer tokens before the current prediction token z_i, and L is the sequence length. In this study, we explore two strategies for obtaining state-action tokens Z_h: 1) directly using (tokenized) text-based environmental observations, 2) *translating* observations/actions into embedding features via a Hopfield module (Sec 3.2) — the problem of finding a suitable translation from environment observations to the language domain.

3.1.1. Action Generation

At each timestep t, we acquire the next action $\bar{a}_t$ by prompting the frozen LLM with a list of the K available actions Z_p and a description of recent states and actions Z_h. We rely on closed-form generation, in which a list of K possible actions is given to the LLM, and the action with the highest log-probability is returned:

$$\bar{a}_t = \max_{a^i \in \{1,\ldots,K\}} LLM(a^i|Z_h, Z_p), \tag{2}$$

where Z_p are the tokens of the tokenized action generation prompt (see Appendix B).

Instead of directly performing the LLM-recommended action $\bar{a}$ that may be suboptimal, we leverage it to drive exploration through an intrinsic reward. The *action intrinsic reward* r^a is computed as the similarity between the LLM-generated action $\bar{a}$ and the action that was predicted by an *action head* f^a. Specifically, the action head f^a that is attached to the policy (Figure 1) predicts the next action given the internal representation $\phi(o)$ learned by the policy. We compute the intrinsic reward r^a in the following manner:

$$r_t^a = \frac{1}{2} \left\| f^a(\phi(o_t)) - \bar{a}_t^{'} \right\|_2^2, \tag{3}$$

where $\bar{a}_t^{'}$ is an indicator vector containing 1 for the action $\bar{a}$ and 0 otherwise. f^a is trained with respect to its parameters θ_A to minimize the following prediction-error loss:

$$L_{act}(\bar{a}, \phi(o)) = -\sum_{i=1}^{|\mathcal{A}|} \bar{a}^i \log(p_i | \phi(o)), \tag{4}$$

where $\bar{a}^i$ is a binary indicator (0 or 1) if action $\bar{a}$ is the correct action for observation $\phi(o)$, and p_i is the predicted probability of action i by f^a.

3.1.2. History Compression

The second language task being used is history compression, also referred to as summarization. The LLM is prompted to compress the agent's history into a short text. We rely on open-ended generation, in which the LLM outputs a summary of past state-action tuples Z_h.

Assuming a history compression head f^{hc} (Figure 1) parametrized by θ_{HC}, the *history compression intrinsic reward* r^{hc} is proportional to the Euclidean distance between the mean-pooled representation of the summary generated by the LLM and the logits produced by f^{hc}:

$$r_t^{hc} = \frac{1}{2} || \sigma(LLM(Z_h, Z_p)) - f^{hc}(\phi(o_t)) ||_2^2, \tag{5}$$

where, for the brevity of method description, $\sigma(LLM(Z_h, Z_p))$ refers to the mean-pooled representation of the LLM, and Z_p is the summarization prompt. f^{hc} is trained to minimize the L2 loss with the LLM's mean-pooled representation, L_{hc}. Since f^{hc} gradients can backpropagate to the policy, this task encourages the model to focus on task-relevant information — noise is discarded by the pretrained LLM during history compression. In addition, we hypothesize that predicting information from a temporally extended horizon improves exploration in POMDPs and guards against premature vanishing of intrinsic rewards. Namely, unlike next action generation, history compression considers a broader context and the cumulative effects of actions rather than isolated steps.

The overall optimization problem that is solved for learning the agent can be written as,

$$\min_{\theta_P, \theta_A, \theta_{HC}} \left[-\lambda \mathbb{E}_{\pi(s;\theta_P)} [\sum_t r_t^*] + (1-\beta) L_{act} + \beta L_{hc} \right], \tag{6}$$

where $0 \leq \beta \leq 1$ is a scalar that weighs the action-generation loss against summarization loss, λ is a scalar that weighs the importance of the policy gradient loss against the importance of learning the intrinsic signal, and the augmented reward is defined as $r_t^* = r_t + b_t = r_t + r_t^a + r_t^{hc}$. $\theta_P, \theta_A, \theta_{HC}$ are the parameters of π, f^a and f^{hc} respectively.

3.2. Translating State-Action Pairs into Embedding Features

So far, we have seen how to guide the agent's exploration by querying an LLM with a prompt Z_p and a description of state-action pairs Z_h. Although a text-based description may be available in some tasks, we cannot always expect to have access to such type of observations. Therefore, we argue that it is necessary to design a mechanism that, given any type of observations and actions, can map them to the token embedding space of the LLM.

To overcome this challenge, we present a method to align environment pairs of observations $o_t \in \mathbb{R}^n$ and past actions $a_{t-1} \in \mathbb{R}^d$ to the LLM's embedding space, which does not require back-propagating gradients through the entire language model. It relies on a Hopfield module that performs a randomized attention over pretrained token embeddings of the LLM $\mathbf{E} = (e_1, ..., e_n)^\top \in \mathbb{R}^{k \times m}$, where k is the vocabulary size and m the embedding size.

Assuming $\mathbf{P} \in \mathbb{R}^{m \times (n+d)}$ to be a random matrix with entries sampled independently from a Gaussian distribution $\mathcal{N}(0, (n+d)/m)$, let x_t to be the output of the Hopfield:

$$x_t = \mathbf{E}^\top \text{softmax}(\beta \mathbf{E} \mathbf{P}(o_t \cdot a_{t-1})), \tag{7}$$

where $\cdot$ denotes the concatenation and β is a hyperparameter that controls the dispersion of x_t within the convex hull of the token embeddings. This corresponds to a spatial compression of observations and actions to a mixture of tokens in the LLM embedding space. At time t, the aligned representation Z_h of a state-action pair is expressed as:

$$Z_h = LLM(c_{t-1}, x_t), \tag{8}$$

where c_t is the context cached in the memory register of the LLM up to timestep t.

4. Experiments

Environments. The experimental evaluation aims to test our central hypothesis: LLMs improve the exploration efficiency for RL algorithms in sparse reward environments. We conduct a serie of experiments on nine BabyAI-Text tasks Chevalier-Boisvert et al. (2018), including KeyCorrS4R3, KeyCorrS5R3, ObstrMaze2D1HB, ObstrMaze1Q, GoToObj, PickupLoc, PutNextS7N4Carrying, PutNextLocal, and OpenRedDoor. To demonstrate iLLM's scalability, we extend the evaluation to more challenging MiniHack tasks Samvelyan et al. (2021), including LavaCrossing-Ring, LavaCross-Potion, LavaCross-Full, MultiRoom-N4-Monster, and River-Monster. We also demonstrate the importance of translating state-action pairs into the LLM's embedding space by evaluating iLLM on five Atari games Bellemare et al. (2013), featuring image-based observations and long-term exploration. Finally, we demonstrate that iLLM can be used in tasks that require skill acquisition, such as in the Crafter environment Hafner (2021).

Baselines. We compare our method against a number of baselines: RND Burda et al. (2019b) and NGU Badia et al. (2020) that employ prediction errors to motivate exploration, APT Liu and Abbeel (2021) that exposes task-specific rewards after an unsupervised pre-training phase, and ELLM Du et al. (2023) that rewards the agent for achieving any goal suggested by an LLM. As highlighted in a recent survey Hao et al. (2023), RND, NGU, and APT were selected since they operate in the *low data* regime, unlike some other methods

	Key Corridor Tasks		Obstructed Maze Tasks		Go To Task	Pickup Task	Put Next Tasks		Open Door Task
Method	KeyCorrS4R3	KeyCorrS5R3	ObstrMaze2D1HB	ObstrMaze1Q	GoToObj	PickupLoc	PutNextS7N4Carrying	PutNextLocal	OpenRedDoor
RND	0.0±0.00	0.0±0.00	0.0±0.00	0.0±0.00	0.51±0.22	0.18±0.11	0.22±0.09	0.0±0.00	0.34±0.13
	$> 60M$	$> 200M$	$> 200M$	$> 300M$	$> 100M$	$> 100M$	$> 100M$	$> 100M$	$> 100M$
NGU	0.34±0.25	0.0±0.00	0.0±0.00	0.0±0.00	0.42±0.25	0.25±0.20	0.28±0.14	0.01±0.01	0.34±0.16
	$> 60M$	$> 200M$	$> 200M$	$> 300M$	$> 100M$	$> 100M$	$> 100M$	$> 100M$	$> 100M$
ELLM	0.89±0.01	0.90±0.01	0.17±0.08	0.33±0.06	0.88±0.01	0.66±0.17	0.45±0.17	0.06±0.08	0.65±0.10
	$60M$	$190M$	$> 200M$	$> 300M$	$80M$	$> 100M$	$> 100M$	$> 100M$	$60M$
APT	0.12±0.06	0.5±0.14	0.0±0.00	0.0±0.00	0.48±0.17	0.30±0.08	0.41±0.25	0.14±0.08	0.98±0.01
	$> 60M$	$> 200M$	$> 200M$	$> 300M$	$> 100M$	$> 100M$	$> 100M$	$> 100M$	$47M$
Pangu	0.90±0.01	0.92±0.01	0.86±0.08	0.45±0.12	0.92±0.01	0.60±0.09	0.68±0.21	0.01±0.02	0.90±0.01
	$> 60M$	$168M$	$> 200M$	$> 300M$	$65M$	$> 100M$	$> 100M$	$> 100M$	$33M$
ChibiT	0.88±0.04	0.90±0.01	0.77±0.10	0.74±0.13	0.76±0.09	0.70±0.12	0.62±0.11	0.33±0.07	0.89±0.03
	$> 60M$	$193M$	$> 200M$	$> 300M$	$> 100M$	$> 100M$	$> 100M$	$> 100M$	$27M$
PAE	**0.93±0.00**	0.92±0.01	0.88±0.01	0.89±0.01	0.94±0.01	0.77±0.22	0.71±0.22	0.28±0.03	0.89±0.01
	$30M$	$90M$	$150M$	$150M$	$53M$	$89M$	$> 100M$	$> 100M$	$28M$
iLLM(obs)	**0.93±0.01**	**0.92±0.02**	0.89±0.00	0.91±0.01	**0.94±0.00**	0.80±0.06	0.76±0.01	0.38±0.14	**0.96±0.01**
	$30M$	$76M$	$130M$	$132M$	$39M$	$77M$	$100M$	$> 100M$	$22M$
iLLM(hop)	0.94±0.01	0.90±0.02	**0.92±0.02**	**0.93±0.01**	0.92±0.01	**0.85±0.01**	**0.78±0.11**	**0.49±0.12**	**0.96±0.03**
	$33M$	$81M$	$128M$	130	$45M$	$68M$	$> 100M$	$> 100M$	$25M$

Table 1: Comparison of iLLM and baseline approaches in BabyAI environments. Averages over 10 runs. Each entry consists of two rows of results, with the top row being the average extrinsic reward at the end of training and the bottom row being the minimal stable steps to attain that reward. Smaller bottom row values signify faster convergence, and "$> n$" indicates the absence of convergence within the maximum training steps "n".

that require billions of training steps. When available, we also report results of Pangu Christianos et al. (2023), ChibiT Reid et al. (2022), and PAE Anonymous (2023) agents, two approaches built upon LLM-driven exploration. Our comparisons involve two variations of iLLM: iLLM(obs), which utilizes textual descriptions provided by the environment for action generation and history compression, and iLLM(hop), which leverages translated state-action pairs as inputs for the language tasks.

Implementation Details. As our policy learning method, we rely on PPO Schulman et al. (2017) with Generalized Advantage Estimation and clipping parameter $\epsilon = 0.2$. The actor and critic networks consist of three fully-connected layers with 128 hidden units. Tanh is used as the activation function, and the output value of the actor network is scaled to the range of each action dimension. Training is carried out with a fixed learning rate of 0.0007 using the AdamW optimizer, with a batch size of 128. The policy is trained for 4 epochs after each episode. As for the LLM choice, we compared several models (see Section 4.5.1), and selected Transfo-XL 280M with the temperature = 0. The intrinsic reward $b_t = r_t^a + r_t^{hc}$ is normalized and then scaled by a factor 0.3 before being summed up with r_t. The prompts, pseudo-code, and more implementation details of iLLM are shown in Appendix B.

4.1. BabyAI-Text Tasks

iLLM was evaluated on nine BabyAI-text tasks. BabyAI-text is a suitable evaluation environment as it provides both image-based and text-based representations of observations. We report the mean and standard deviation of the success rate over 10 seeds in Table 1. We can draw a couple of observations from the results. iLLM achieves higher convergence speed than most prior studies. In comparison, in PickUpLoc, both RND and NGU are still under 0.25 after 100 million steps, while iLLM(hop) reaches ≈ 0.85 after 68 million steps. Notably, our method exhibits a significantly higher final performance compared to ELLM, due to the difficulty of assessing when a goal was achieved and its tendency to select suboptimal goals.

Method	LavaCrossing-Ring	LavaCross-Potion	LavaCross-Full	MultiRoom-N4-Monster	River-Monster
RND	0.0±0.00 $> 40M$	0.0±0.00 $> 40M$	0.0±0.00 $> 40M$	0.0±0.00 $> 40M$	0.0±0.00 $> 20M$
NGU	0.0±0.00 $> 40M$	0.09±0.02 $> 40M$	0.0±0.00 $> 40M$	0.14±0.11 $> 40M$	0.06±0.07 $> 20M$
ELLM	0.29±0.11 $> 40M$	0.51±0.10 $> 40M$	0.44±0.15 $> 40M$	0.28±0.20 $> 40M$	0.22±0.03 $> 20M$
APT	0.11±0.08 $> 40M$	0.32±0.13 $> 40M$	0.40±0.06 $> 40M$	0.31±0.02 $> 40M$	0.08±0.00 $> 20M$
Pangu	0.98±0.10 $35M$	0.88±0.01 $40M$	0.50±0.07 $> 40M$	0.70±0.16 $> 40M$	0.16±0.02 $> 20M$
ChibiT	0.78±0.11 $> 40M$	0.86±0.04 $> 40M$	0.39±0.10 $> 40M$	0.46±0.12 $> 40M$	0.13±0.04 $> 20M$
PAE	1.0±0.00 $22M$	0.99±0.00 $35M$	1.0±0.00 $24M$	0.72±0.00 $> 40M$	0.13±0.01 $> 20M$
iLLM(obs)	**1.0±0.00** $20M$	**0.99±0.01** $32M$	**0.99±0.01** $22M$	0.69±0.04 $> 40M$	0.13±0.01 $> 20M$
iLLM(hop)	0.97±0.02 $28M$	1.0±0.02 $39M$	0.98±0.03 $26M$	**0.75±0.07** $> 40M$	**0.38±0.12** $> 20M$

Table 2: Results against exploration algorithm baselines in MiniHack environments. Averages over 10 runs.

The results demonstrate how language-driven rewards can be used as a tool to scaffold learning by leveraging their prior knowledge. As expected, iLLM(hop) has a slightly slower convergence, although it ends up reaching the same or higher final performance if run long enough. This might be attributed to the richer representation captured by the Hopfield module, surpassing the simplicity of human-crafted text-based observations.

4.2. MiniHack Environment

Table 2 gives the quantitative results in the MiniHack environment Samvelyan et al. (2021), and shows the average extrinsic reward as well as the number of steps required for each model to converge. Utilizing an LLM as done by ELLM, PAE, SFT-RL, and iLLM outperforms pure curiosity-driven approaches, including RND and NGU. Additionally, we notice that iLLM(hop) reaches similar performance with iLLM(obs), demonstrating the relevance of the proposed state-action alignment technique. Nevertheless, LLMs are prone to mistakes in the MiniHack domain, capping the score of ELLM. This highlights the significance of exploration driven by intrinsic rewards as opposed to plain "imitation learning". Moreover, in River-Monster, iLLM(hop) achieves state-of-the-art performance by leveraging the Hopfield module's ability to capture temporal information into the learned representations of states and actions.

4.3. Atari Games

We also evaluate iLLM on five difficult exploration Atari 2600 games from the Arcade Learning Environment (ALE) Bellemare et al. (2013): Montezuma's Revenge (MR), PrivateEye, Gravitar, Pitfall, and Seaquest. In the selected games, training an agent with a poor exploration strategy often results in a suboptimal policy. Note that some baselines such as ELLM

Method	MR	PrivateEye	Gravitar	Pitfall	Seaquest
RND	456±55	598±110	192±34	-11±3	2,612±315
NGU	512 ±39	1,872±128	1,630±111	-6±2	15,616±3,838
ELLM	-	-	-	-	-
APT	711±66	2,982±322	1,420±245	-12±3	**19,989±2,873**
ChibiT	1,231±187	3,633±334	2,983±302	-10±2	16,441±2,462
iLLM (obs)	-	-	-	-	-
iLLM (hop)	**2,632±277**	**4,422±376**	**4,044±559**	**125±24**	18,851±2,930

Table 3: Performance of curiosity-driven learning algorithms and iLLM on Atari tasks. All methods are tested with 10 random seeds. Averages over 10 runs for 100 million steps.

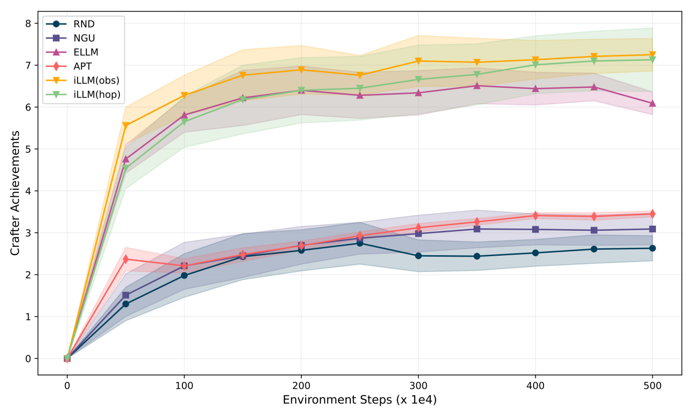


Figure 2: Ground truth achievements unlocked per episode, mean±std across 10 seeds.

and iLLM(obs) could not be evaluated on those tasks due to the lack of textual representation of the environments. The results are presented in Table 3. It is observed that RND and NGU obtained a score close to zero and could not solve most of the tasks. Besides, on Montezuma's Revenge, PrivateEye, Gravitar, and Pitfall, our technique outperforms other approaches that do not graft world knowledge onto the agent's framework. These results suggests that LLMs play an important role in exploring complex environments. We noticed that in those tasks where language models have satisfactory knowledge, leveraging their prior assumptions significantly boosts sample efficiency at the onset of the training phase.

4.4. Crafter Environment

In this section, we evaluate the agents on the Crafter environment, a 2D version of Minecraft Hafner (2021). An optimal exploration method would unlock all Crafter achievements in every episode. Therefore, we report in Figure 2 the average number of unique achievements per episode. Even without access to Crafter's achievement tree, iLLM was able to unlock about 7 achievements every episode, against 6 for the best baseline. Notably, iLLM outperforms all exploration methods that primarily focus on generating novel behaviors such

Model	Translated observations	# parameters	Avg Return
Transfo-XL	✗	280M	0.83
Transfo-XL	✓	280M	0.85
Flan-T5	✗	780M	0.75
Flan-T5	✓	780M	0.79
Llama-2	✗	7B	0.87
Llama-2	✓	7B	**0.88**

Table 4: Ablation study of the choice of backbone language model. We choose three advanced models with different numbers of parameters and architectures. We report the average return across the nine BabyAI-text tasks (10 seeds).

as RND, APT, and NGU. Those methods encourage exploration of diverse behaviors without considering the relevance of the learned behaviors. In contrast, both iLLM(obs) and iLLM(hop) reduce the exploration space by biasing exploration towards plausibly useful behaviors. Furthermore, appending f^a and f^{hc} to the policy and training them to match the pretrained LLM's outputs enables our agent to leverage world knowledge through their respective gradients.

4.5. Ablation Studies

4.5.1. Choice of the Backbone

In Table 4, we conduct experiments to analyze the effect of different LLM backbones on iLLM. We report the average return across the nine BabyAI-text tasks. From the table, it can be observed that (1) iLLM using large backbones such as Llama-2 would benefit the exploration efficiency while bringing more memory and computation cost; (2) Transfo-XL model achieved the best trade-off between sample efficiency and time efficiency. In addition, we notice that using translated observations leads to slightly increased performance compared to text observations.

4.5.2. Randomized Environment

It was shown by several authors that Savinov et al. (2019); Burda et al. (2019b) agents that maximize the "surprise", tend to suffer from the TV noise problem — when the agent finds a way to instantly gratify itself by exploiting actions that lead to hardly predictable consequences. In other words, an agent maximizing this prediction error may seek out stochasticity (e.g., randomized transitions, high-frequency images) in the environment to maximize the error. We now evaluate iLLM trained on randomized environments that are based on GoToObj with added sources of stochasticity:

- "Original": the original GoToObj environment.
- "Noise": if the agent selects the action *go forward*, a noise pattern (32×32) is displayed on the lower right of the observation - TV screen. The noise is sampled from [0,255] independently for each pixel.

	Success rate			
Method	Original	Noise	Noise action $\varrho = 0.05$	Noise action $\varrho = 0.1$
RND	0.51± 0.22	0.24± 0.26	0.44± 0.18	0.39± 0.19
NGU	0.42± 0.25	0.16± 0.19	0.27± 0.24	0.19± 0.20
ELLM	0.66± 0.01	0.45± 0.09	0.58± 0.04	0.55± 0.06
APT	0.48± 0.17	0.22± 0.20	0.41± 0.15	0.37± 0.17
ChibiT	0.56± 0.23	0.41± 0.28	0.55± 0.21	0.66± 0.15
iLLM(obs)	**0.94± 0.00**	**0.65± 0.08**	**0.91± 0.06**	**0.87± 0.08**
iLLM(hop)	0.92± 0.01	0.59± 0.05	**0.91± 0.07**	0.85± 0.11

Table 5: Average success rate over 10 seeds in the randomized-TV versions of GoToObj task (mean±std).

Method	MR	PrivateEye	Gravitar	Pitfall	Seaquest
PPO	2.11±0.18	1.84±0.21	2.26±0.22	2.70±0.36	1.45±0.22
RND	2.07±0.21	2.12±0.25	2.09±0.33	1.87±0.27	0.98±0.25
iLLM(hop)	**-1.76±0.17**	**-1.65±0.20**	**-1.44±0.18**	**0.06±0.04**	**-1.61±0.21**
iLLM(hop)(no reward)	-1.15±0.20	-1.18±0.16	-0.99±0.08	0.34±0.12	-1.47±0.18

Table 6: Normalized Euclidean distances (± std) of agent trajectories from human demonstrations.

- "Noise Action": if the agent selects the action *go forward*, with a probability $\varrho \in \{0.05, 0.10\}$, the action performed by the agent is uniformly sampled among the possible actions.

We observe in Table 5 a decrease in the performance of most approaches. However, our formulation turns out to be more robust than NGU's prediction error in this scenario i.e., noise action $\varrho = 0.05$ and noise action $\varrho = 0.10$. While NGU is trapped in local optima, since iLLM does not directly rely on next action prediction or observation, it is less impacted by stochasticity in the world. iLLM(obs) and iLLM(hop) scores are significantly higher compared to the baselines as indicated by paired t-tests at 95% confidence level ($p < 0.002$). When adding visual noise to the environment, the performance of iLLM(hop) appears to deteriorate more than iLLM(obs). Visiting a state with a noise pattern produces a more noisy representation of the world, making the alignment tasks harder. Nevertheless, the proposed formulation of curiosity is reasonably robust to the TV noise problem by leveraging the LLM's ability to abstract away irrelevant details.

4.5.3. Human-Meaningful Exploration

An appealing aspect of using a foundation model to guide exploration is that it allows us to implicitly incorporate prior beliefs about human-meaningful behaviors through the neural network architecture and exploration bonus. To assess how human-meaningful the agent's exploration is, we report in Table 6 the average Euclidean distance between the agent's state

Method	Percentage of goals achieved			
	GoToObj	PutNextLocal	KeyCorrS5R3	PutNextS7N4Carrying
PPO	0.12± 0.02	0.0± 0.01	0.16± 0.09	0.0± 0.06
iLLM(obs)	**0.91± 0.01**	0.46± 0.08	0.90± 0.01	0.72± 0.09
iLLM(hop)	0.90± 0.02	**0.52± 0.11**	**0.90± 0.02**	**0.75± 0.09**

Table 7: Success rate of iLLM and baseline agents on BabyAI tasks in the "dense" reward case. Results are averaged over 10 random seeds (±std). No seed tuning is performed.

and the nearest state in the demonstration data at each time step. The demonstration data consists of one trajectory for each of the five games. Agent trajectories were collected during the first 20 million training steps. To normalize these distance values across different scales and scenarios, we apply a z-score normalization method. This normalization adjusts for the mean and standard deviation of the distances observed across all sampled trajectories, thereby enabling a more consistent comparison.

Experimental results indicate that, generally, iLLM exhibits larger positive distances compared to PPO. Specifically, PPO results show a significant deviation from human demonstrations across all games, particularly in more complex games like Pitfall. Our method outperformed RND by consistently achieving negative distances, which demonstrates a closer alignment to human trajectories. Notably, even in Pitfall iLLM(hop) achieves a small positive deviation, highlighting an exploration more aligned with the human demonstrator than vanilla PPO and RND as they uniformly explore the environment. These findings suggest that the present architecture yields human-meaningful exploration by incorporating inductive bias of foundation models.

4.5.4. Dense Rewards

A desirable property of the present study is to avoid hurting performance in tasks where rewards are dense and well-defined. We report results on four BabyAI tasks Chevalier-Boisvert et al. (2018) in Table 7, including plain PPO trained only with extrinsic rewards. In the standard sparse setting, the agent is only provided a sparse terminal reward of +1 if it finds the target and 0 otherwise. In the dense setting, the agent is rewarded (+0.3) when selecting the correct action (e.g., collecting keys, opening doors). The table indicates that the performance of our method does not deteriorate drastically in dense reward tasks. Even though iLLM(obs) and iLLM(hop) perform slightly worse in the dense setting, they still perform substantially better compared to plain PPO.

5. Conclusion

In this work, we introduce a novel approach for language-driven exploration in reinforcement learning (RL), leveraging LLMs to guide exploration towards diverse and human-meaningful regions of the state space. Namely, short-term curiosity is captured by querying a frozen LLM with an action generation task. In addition, we compress state-action history via a summarization task, discarding irrelevant details and encouraging the policy to extract task-relevant information. We further present a novel alignment technique that facilitates the integration of state-action pairs from any modalities into the language domain, obviating the necessity for textual environmental descriptions. We have empirically demonstrated the effectiveness of our approach across diverse and challenging domains, including BabyAI-Text, MiniHack, Atari, and Crafter, showcasing substantial improvements in sample efficiency and performance. Interesting directions for future work include improving state-action pairs alignment and evaluating additional language tasks such as goal generation.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Anonymous. PAE: Reinforcement learning from external knowledge for efficient exploration. In *Submitted to The Twelfth International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=R7rZUSGOPD. under review.

Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martín Arjovsky, Alexander Pritzel, Andew Bolt, et al. Never give up: Learning directed exploration strategies. *arXiv preprint arXiv:2002.06038*, 2020.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, jun 2013.

Nicolas Bougie and Ryutaro Ichise. Exploration via progress-driven intrinsic rewards. In *Artificial Neural Networks and Machine Learning–ICANN 2020: 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 15–18, 2020, Proceedings, Part II 29*, pages 269–281. Springer, 2020a.

Nicolas Bougie and Ryutaro Ichise. Skill-based curiosity for intrinsically motivated reinforcement learning. *Machine Learning*, 109:493–512, 2020b.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A. Efros. Large-scale study of curiosity-driven learning. In *Proceedings of the The International Conference on Learning Representations*, 2019a.

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

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: A platform to study the sample efficiency of grounded language learning. *arXiv preprint arXiv:1810.08272*, 2018.

Filippos Christianos, Georgios Papoudakis, Matthieu Zimmer, Thomas Coste, Zhihao Wu, Jingxuan Chen, Khyati Khandelwal, James Doran, Xidong Feng, Jiacheng Liu, et al. Pangu-agent: A fine-tunable generalist agent with structured reasoning. *arXiv preprint arXiv:2312.14878*, 2023.

Ishita Dasgupta, Christine Kaeser-Chen, Kenneth Marino, Arun Ahuja, Sheila Babayan, Felix Hill, and Rob Fergus. Collaborating with language models for embodied reasoning. *arXiv preprint arXiv:2302.00763*, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding pretraining in reinforcement learning with large language models. *arXiv preprint arXiv:2302.06692*, 2023.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35:18343–18362, 2022.

Tarun Gupta, Peter Karkus, Tong Che, Danfei Xu, and Marco Pavone. Foundation models for semantic novelty in reinforcement learning. *arXiv preprint arXiv:2211.04878*, 2022.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. *arXiv preprint arXiv:2109.06780*, 2021.

Jianye Hao, Tianpei Yang, Hongyao Tang, Chenjia Bai, Jinyi Liu, Zhaopeng Meng, Peng Liu, and Zhen Wang. Exploration in deep reinforcement learning: From single-agent to multiagent domain. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR, 2022.

Martin Klissarov, Pierluca D'Oro, Shagun Sodhani, Roberta Raileanu, Pierre-Luc Bacon, Pascal Vincent, Amy Zhang, and Mikael Henaff. Motif: Intrinsic motivation from artificial intelligence feedback. *arXiv preprint arXiv:2310.00166*, 2023.

Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. Reward design with language models. *arXiv preprint arXiv:2303.00001*, 2023.

Jessy Lin, Yuqing Du, Olivia Watkins, Danijar Hafner, Pieter Abbeel, Dan Klein, and Anca Dragan. Learning to model the world with language. *arXiv preprint arXiv:2308.01399*, 2023.

Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, 34:18459–18473, 2021.

Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. A survey of reinforcement learning informed by natural language. *arXiv preprint arXiv:1906.03926*, 2019.

Fabian Paischer, Thomas Adler, Vihang Patil, Angela Bitto-Nemling, Markus Holzleitner, Sebastian Lehner, Hamid Eghbal-Zadeh, and Sepp Hochreiter. History compression via language models in reinforcement learning. In *International Conference on Machine Learning*, pages 17156–17185. PMLR, 2022.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

Machel Reid, Yutaro Yamada, and Shixiang Shane Gu. Can wikipedia help offline reinforcement learning? *arXiv preprint arXiv:2201.12122*, 2022.

Mikayel Samvelyan, Robert Kirk, Vitaly Kurin, Jack Parker-Holder, Minqi Jiang, Eric Hambro, Fabio Petroni, Heinrich Kuttler, Edward Grefenstette, and Tim Rocktäschel. Minihack the planet: A sandbox for open-ended reinforcement learning research. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL https://openreview.net/forum?id=skFwlyefkWJ.

Nikolay Savinov, Anton Raichuk, Raphaël Marinier, Damien Vincent, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly. Episodic curiosity through reachability. In *Proceedings of the International Conference on Learning Representations*, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Pratyusha Sharma, Antonio Torralba, and Jacob Andreas. Skill induction and planning with latent language. *arXiv preprint arXiv:2110.01517*, 2021.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? *arXiv preprint arXiv:2203.07540*, 2022.

Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.