Leveraging LLM with Active Imitation Learning of Hierarchical Reinforcement Learning using Emergent Symbolic Representation

Ziqi MA[1], Sao Mai NGUYEN[1], Philippe XU[1]

Abstract— **Large Language Models (LLMs) exhibit their potential for interacting with reinforcement learning agents, the main challenge is to align the world model learned by the agent with a representation compatible with LLMs, these representations should be well structured and contain the whole information of the environment. Some hierarchical reinforcement learning (HRL) addresses this challenge by decomposing task and producing emergent symbolic representations of a long-horizon task. However, a central open question remains: how to effectively learn a representation of the environment that aligns with LLM? We study in this paper, how a symbolic representation of space can be taken advantage of by LLMs for learning long-horizon tasks. First, we evaluate the translation ability of the state-of-the-art LLMs in symbolic representation of emergent learning agent in Ant Maze task, showing that they succeed under coarse symbolic partitions however degrade with finer granularity. Inspired by this observation, we introduce SGIM-STAR, a hybrid framework where the top-level agent choose actively between a Q-learning based Commander and an LLM-based planner using a partition-wise, progress-driven intrinsic rule. Both strategies in this framework use a symbolic representation of the space. Experiments demonstrate that SGIM-STAR improves stability over STAR, reduces reliance on costly LLM calls, and achieves higher long-horizon task success. Our findings highlight the dual role of LLMs as both translators of human intent and adaptive planners grounded in emergent symbolic representations, paving the way for more interpretable and language-grounded robotic planning.**

I. INTRODUCTION

Robotic agents deployed in complex environments must not only learn low-level control of each actuator but plan over long horizons. For instance, in the Ant Maze [1] environment, a legged robot needs to learn both to control its legs in a coordinated way to move in the desired direction, and to plan its path in the long-term by setting subgoals in order to navigate a '⊃'-shaped maze to reach the exit positioned at the top left. Hierarchical Reinforcement Learning (HRL) [2] has emerged as a promising paradigm to address this challenge by decomposing tasks into manageable small tasks. Several HRL algorithms [3], [4] and neurosymbolic representation algorithms [5] focus on learning representations of the subgoal space to mitigate the curse of dimensionality and enable efficient long-horizon planning. Such emergent representations are attractive because they provide structured abstractions of the environment. While the algorithm LESSON [3] uses a continuous latent space to represent subgoals, and DeepSym [5] learns to extract symbolic representations assuming a finite set of predefined actions, STAR [4] learns online a symbolic representation of subgoals without a list of predefined features given in advance in continuous state and action spaces.

For robots to act for and with people, their representations must be not only functional but also reflective of what humans care about, i.e. they must be aligned. Because humans are the ultimate evaluator of robot performance, we must make efforts to align the learned representations with humans, such as their linguistic descriptions of the task and environment through linguistic symbols. Therefore, they can serve as a bridge toward natural language interaction. Thus, intuitively, symbolic internal representations could provide an easier way for *representation alignment*. This is why in this work, we will examine symbolic representations learned by HRL, such as with the STAR algorithm [4].

Recent advances have shown that human representation through language has been quite successfully modeled by foundation models of Large Language Models (LLMs). Although trained mostly on abstract content such as from web scrapping, LLMs can even carry internal reasoning capabilities [6]–[10] that can be exploited for effective interaction in embodied AI to drive reinforcement learning (RL) and planning agents [11], [12]. However, their performance is poor in the case of real-world inference because of the grounding problem [13]. Another limitation of the use of LLMs for effective interaction within environments is the lack of a symbolic representation of the continuous, physical embodied world. Thus, a central open question is how to effectively learn a symbolic representation of the environment that aligns with the LLM so that they can (i) convey the natural language instructions from humans, and (ii) improve learning and planning performance.

Our *contribution* is to study how a symbolic representation of space learned online with HRL can be taken advantage of by LLMs, for learning long-horizon tasks. In this paper, we address these challenges through two stages:

- First, we test the possibility of involving an LLM in the RL learning process by analysing the alignment of the emergent symbolic representation with language. More concretely, evaluate whether natural language instructions can be translated into a path using the emergent symbolic partitions produced by an online HRL algorithm, STAR. We show that LLMs can translate between human language instruction and the emergent symbolic representation, with near-perfect score under coarse partitions, even though their score degrades with

*This work is partially supported by Hi! Paris

[1] All the autors are with Laboratory of Computer Science and System Engineering (U2IS), ENSTA, IP-Paris, 828 Bd des Marechaux, 91120 Palaiseau, France. With the mail ziqi_ma0605@163.com for Ziqi MA, nguyensmai@gmail.com for Sao Mai NGUYEN and philippe.xu@ensta.fr for Philippe XU

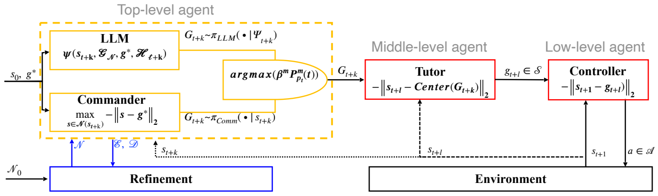

Top-level agent
LLM
$\psi(s_{t+k}, \mathcal{G}_{\mathcal{N}}, g^*, \mathcal{H}_{t+k})$
$G_{t+k} \sim \pi_{LLM}(\bullet \mid \Psi_{t+k})$
$argmax(\beta^m P^m_{p_t}(t))$
Commander
$\max_{s \in \mathcal{N}(s_{t+k})} -\|s - g^*\|_2$
$G_{t+k} \sim \pi_{Comm}(\bullet \mid s_{t+k})$
s_0, g^*
G_{t+k}
Middle-level agent
Tutor
$-\|s_{t+l} - Center(G_{t+k})\|_2$
$g_{t+l} \in \mathcal{S}$
Low-level agent
Controller
$-\|s_{t+1} - g_{t+l})\|_2$
$\mathcal{N}$
$\mathcal{E}, \mathcal{D}$
s_{t+k}
s_{t+l}
s_{t+1}
$a \in \mathcal{A}$
$\mathcal{N}_0$
Refinement
Environment

Fig. 1: Algorithmic architecture of SGIM-STAR which integrates the LLM and the Commander of STAR at the top-level agent and chooses between the LLM and the Commander based on progress. The Commander or the LLM selects subgoal regions $G \in \mathcal{G}$, while the middle-level tutor and low-level controller are unchanged.

finer symbolic granularity.

- Second, we propose SGIM-STAR (Socially Guided Intrinsic Motivation based on Spatio-Temporal Abstraction via Reachability), a novel algorithm to solve the long-horizon tasks in which the agent chooses actively its learning strategy between a reinforcement learning and a LLM-guided planner based on intrinsic motivation [14], [15]. The originality is that we use for both strategies a symbolic representation of the sensorimotor space, by learning from a bottom-up process an emergent symbolic representation, and making it compatible with the symbolic space of language used in a top-down planner.

Our experiments demonstrate that the agent exploits the efficiency of the RL in early training while selectively invoking LLM guidance once a reliable symbolic structure emerges, yielding mutual benefits. Translation analysis shows the possibility of involving LLM in the RL learning process, while SGIM-STAR leverages emergent symbolic structures to stabilize long-horizon learning and reduce reliance on costly LLM calls. Together, these contributions point toward a unified framework where LLMs function both as translators of human instructions and as adaptive planners grounded in emergent symbolic representations. This dual role advances the development of language-grounded hierarchical reinforcement learning for robotics, paving the way for more effective human–robot interaction.

II. Related Works

A. Space Representation for Hierarchical Reinforcement Learning

To address complex tasks which involve long-term planning and multi-step actions, HRL algorithms [3], [16], [17] decompose a task into simpler subtasks, allowing them to be subsequently solved efficiently. Some HRL also sought to solve the curse of dimensionality problem of subgoal space by learning a representation of the subgoal space. LESSON learns latent slow features to capture long-horizon dynamics [3]. GARA [18] and STAR [4] solved the computational cost problem of HRAC by building reachability-aware regions and refining them based on learned k-step reachability, while their training results can be unstable. To address long-horizon tasks in robotic real-world environments that are high-dimensional, we extend this line of work to enhance this internal representation with complementary mechanisms to stabilize the performance.

B. Socially Guided Intrinsic Motivation (SGIM)

To tackle sparse-reward learning tasks, Intrinsic Motivation (IM) drives exploration through automatic curriculum learning before external rewards are observed [14], [15]. Various measures have been proposed, such as novelty, competence, or progress [19]–[21]. However, IM alone struggles in high-dimensional spaces. To complement RL, human-in-the-loop approaches [22] have been developed. Imitation learning [23] and inverse-RL methods [24], [25] serve as computational frameworks for social guidance in robotics, by either transposing a policy from demonstrations or learning a reward signal. Whereas all the methods that metionned usually treat the agent as passive. In contrast, active imitation learning allows the agent to request guidance from teachers [26], [27]. Extending this idea, SGIM couples IM with social input so the agent actively decides what, when, and whom to imitate based on learning progress [28], [29]. This yields an adaptive curriculum where competence improves fastest, enabling efficient exploration in high-dimensional domains. Our work builds on this principle, enabling agents to choose between reinforcement learning or soliciting expert guidance depending on intrinsic progress.

C. Large Language Models in Decision-Making

Recent breakthroughs in LLMs have significantly expanded their capabilities beyond natural language processing to complex reasoning and decision-making tasks. However, integrating LLMs into reinforcement learning frameworks remains challenging due to poor space representation of a continuous environment, whereas LLMs use a discrete, symbolic representation. [30] uses language as the interface between high- and low-level policies in hierarchical RL, with a low-level policy that follows language instructions, and the top-level policy producing actions in the space of language. In [6], LCB uses a learnable latent code to act as a bridge between LLMs and low-level policies. To alleviate the lack of grounding of LLMs in space, these works add to the reinforcement learning agents a new layer to translate between the continuous space of states and the discrete space of LLM symbols. However, the reinforcement learning algorithms GARA and STAR [4] learn a discrete representation directly, which symbols can be more readily used by an LLM. In this work, we explore how the emerging symbolic representation of STAR can be exploited by LLMs.

III. Preliminary: Spatio-Temporal Abstraction via Reachability (STAR)

The STAR algorithm [4] is a reinforcement learning algorithm that uses a three-layered hierarchical structure:

- Commander: the top-level agent plans the long-horizon path by setting intermediate goals. It is trained by Q-learning which chooses an abstract goal $G \in \mathcal{G}$ every

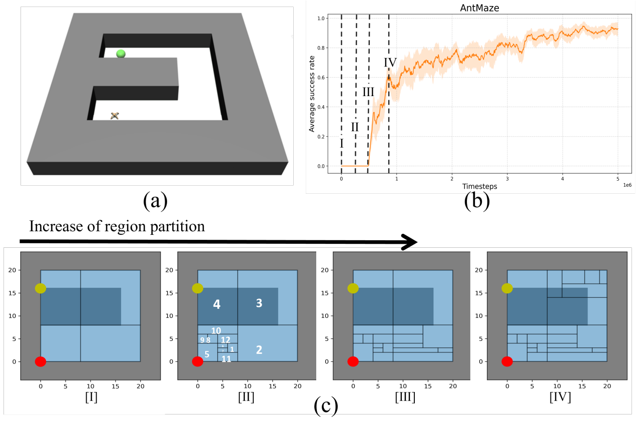


Fig. 2: (a) Ant Maze environment [1], (b) Average success rate of STAR, (c) Partition into regions of STAR. The regions in (c) are the internal representation emerging during the training at timestamps noted in (b). The red point is the initial position of the robot. The yellow point is the goal position. Our translator translates instructions to guide the robot (e.g.: ”go east to the end, turn north until past the wall and go west until the end”), into a sequence of traversed regions (e.g. for Partition II, the output is 5 → 11 → 2 → 3 → 4).

k steps that should help to reach the task goal g^* from the current agent's state $(G_{t+k} \sim \pi_{Comm}(s_t, g^*))$.

- Tutor: the mid-level agent trained by TD3 which picks subgoals in the state space every l steps $(g_{t+l} \sim \pi_{Tut}(s_t, G_{t+k}))$. We note that k is a multiple of l.
- Controller: the low-level policy trained by TD3 that chooses actions to reach the subgoal every step $(a \sim \pi_{Cont}(s_t, g_{t+l}))$

STAR incrementally refines the partition of the sensorimotor space (as shown in Fig. 2 (c)) by analyzing k-step reachability relations between goal regions. The refinement module uses as inputs the past episodes $\mathcal{D}$ and the list of abstract goals $\mathcal{E}$ visited during the last episode, and outputs a partition of the state space.

IV. Internal Representation Used to Translate Natural Language Instructions

A first step toward language-grounded hierarchical reinforcement learning is to determine whether LLMs can align with the symbolic abstractions that emerge during training. We collect the symbolic partitions that emerge during the execution of the STAR algorithm on the Ant Maze, as illustrated in Fig. 2(a). To analyze LLM performance across different levels of abstraction, we select four representative partitions from different developmental learning stages, as shown in Fig. 2(c). Their corresponding positions are also marked along the training curves in Fig. 2(b). Partition I corresponds to the initialization partition with a minimal number of symbols; Partition II captures a timestep before any significant learning progress; Partition III aligns with the onset of performance improvement; Partition IV represents the final stage of learning. We keep the agent's start and goal positions fixed and apply the same instruction to all partition levels. The natural language instruction is: “Move right until you completely pass the wall on your left, move up until you have crossed the upper wall, turn left and proceed until you reach the goal”, the LLM needs to translate it into a sequence of traversed regions like 5 → 11 → 2 → 3 → 4 in Partition II. We use G-BLEU score to measure the matching degree of the sequence translate by LLM and the ground truth that is decided by human experts. We report in Table I the G-BLEU score for the translation results.

TABLE I: Mean G-BLEU scores of translations of natural language instructions over 4 runs for each partition in the Ant Maze environment.

Translator	**Ant Maze**			
	P-I	P-II	P-III	P-IV
GPT o3-m	1	1	1	0.87
Claude	1	1	0.73	0.34
Deepseek	1	0.9	0.53	0.65
GROK	1	1	1	0.89

While GPT o3-mini achieves the highest scores, all translators achieve scores above 0.5 across tasks, indicating a generally successful translation of human instructions into the agent's internal symbolic representation. In the Ant Maze environment, all translator scores decrease as the number of regions increases. This observation suggests that as the symbolic space becomes more fine-grained, the increase in abstraction complexity challenges the LLMs' ability to produce coherent and accurate language-to-symbol translation. Notably, the degradation of performance varies for each model: GPT o3-mini and GROK demonstrate greater robustness than DeepSeek and Claude. Given the consistent trends observed across LLMs, we select GPT o3-mini as the representative LLM for subsequent experiments.

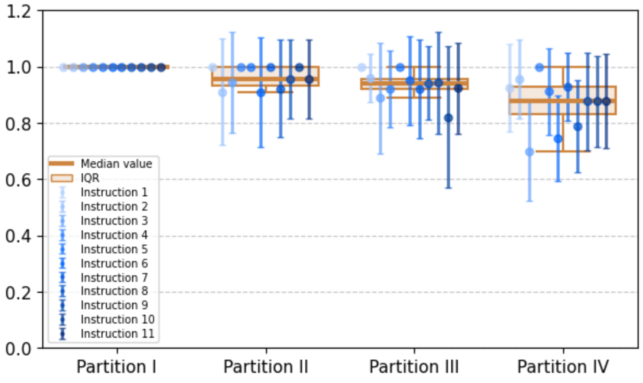


Fig. 3: G-BLEU scores for translation of natural language instructions tested in Ant Maze. For each internal representation, we plot in blue the average and standard deviation of 10 queries for each instruction, and boxplot in brown the average and IQR over the 11 instructions.

In order to test the robustness of the previous results with statistical tests, for our second experiments, we design 11 different natural language instructions for each environment, and tested all 11 instructions on all four partitions. We constructed the prompt and queried the LLM 10 times. Figure 3 illustrates the average G-BLEU scores across the symbolic partitions of Ant Maze. We observe a perfect translation performance in Partition I. The translation performance observed in Ant Maze slightly drops from Partition I to Partition IV. This trend is interpretable through the partition structures shown in Figure 2(c). Partition I represents a very coarse abstraction with minimal region division, making it easier for the LLM to infer plausible region sequences regardless of the instruction quality. Then, when the partition becomes more granular in Ant Maze, the LLM is more prone to mistakes.

These results prove statistically that the representation emerged by STAR can be leveraged by LLM to convey natural language instructions.

V. SGIM-STAR : Combining LLM and RL as planners

Motivated by the finding in Section IV that LLM can be leveraged to translate human language to the symbolic representation emerging during RL agents, we investigate whether LLMs can be involved directly into the RL process for deeper improvements. Specifically, we wish to answer three questions: (i) can LLMs help to build the representations of the space, (ii) can LLMs provide useful guidance signals during training, and (iii) can they be integrated into the decision-making loop with minimal additional cost? To address these questions, we propose a method that integrates LLM as adaptive partners in planning. Our algorithm follows STAR's hierarchical structure and reachability-aware abstraction, but augments the top-level agent with an LLM and introduces a partition-wise, progress-driven active learning between a (Q-learning) RL-based planner and an LLM planner.

A. Integration of an LLM into the Top Level Agent

We extend STAR by incorporating an LLM into the top-level agent. The structure is shown in Fig. 1. Instead of relying solely on the Commander policy $\pi_{Comm}(G|s_t)$, we introduce an LLM-based planner $\pi_{LLM}(G|\Psi_t)$, which operates on a prompt Ψ_t encoding the agent's current region, available partitions, and task description. Thus, the top-level goal selection becomes:

$$G'_{t+k} \sim \pi_{LLM}(\cdot \mid \Psi_t), \quad \Psi_t = \psi(s_t, \mathcal{G}_N, g^*, \mathcal{H}_t), \tag{1}$$

where $\mathcal{G}_N$ is the set of admissible regions, g^* is the task goal, and $\mathcal{H}_t$ summarizes human knowledge. This modification allows the top-level agent to integrate human-readable instructions and world knowledge expressed in natural language, thereby aligning regional exploration with external guidance or commonsense priors.

B. Active Imitation Learning of the Top Level Agent

To dynamically balance between the original STAR Commander and the LLM-based planner, we use intrinsic motivation based on progress measure as a selection mechanism.

Initialization. For the first N decision steps, the planner is chosen randomly between the STAR Commander and LLM in order to populate both buffers with initial experience.

Progress signal. At each timestep t, let $m \in \mathcal{M} = \{STAR, LLM\}$ denote the planner used, known that $\mathcal{G}_t$ is the partition of the space that has the same update mechanism as STAR, let $g_t = \phi(s_t, \mathcal{G}_t)$ be the current region where agent's current state belongs to in the partition $\mathcal{G}$. We define the incremental reward difference: $\Delta_t = r_t - r_{t-1}$ which reflects the immediate progress attributable to the planner's decision at t. This value Δ_t is stored as $\Delta_t^{(m)}$ in the buffer of the corresponding planner m for the active region g_t.

Discounted progress accumulation. For each region g_t and planner m, we compute a discounted cumulative progress over a sliding window of length n:

$$\mathcal{P}_{g_t}^{(m)}(t) = \sum_{j=0}^{n} \alpha^j \, \Delta_{t-j}^{(m)} \tag{2}$$

where $\alpha \in (0,1)$ is a progress discount factor that emphasizes recent progress while retaining memory of past improvements.

Planner selection rule. At each decision step, the algorithm selects the planner according to a progress-maximization criterion:

$$m = \arg\max_{m \in \mathcal{M}} \left\{ \beta^{(m)} \mathcal{P}_{g_t}^{(m)}(t) \right\}, \tag{3}$$

where $\beta^{(LLM)} \geq 0$ is a scaling factor that controls the relative influence of LLM-derived progress ($\beta^{(STAR)} = 1$).

Algorithm 1 SGIM-STAR: Active Imitation Learning of the Top Level Agent

```
Require: Discount $\alpha \in (0,1)$, Window size $n$, Warm-start $N$, Scaling factor $\beta^{(LLM)} \geq 0$, $\beta^{(STAR)} = 1$
Initialize Buffers $\mathcal{B}_g^{(m)} \leftarrow \emptyset$ and scores $\mathcal{P}_g^{(m)} \leftarrow 0$, $\forall g \in \mathcal{G}_0, m \in \mathcal{M} = \{STAR, LLM\}$
$t \leftarrow 0$, observe $(s_0, r_0)$
while episode not terminated do
    $g_t \leftarrow \phi(s_t, \mathcal{G}_t)$
    if $t < N$ then
        $m \sim$ Uniform($\mathcal{M}$)
    else
        $\mathcal{P}_{g_t}^{(m)}(t) = \sum_{j=0}^{n} \alpha^j \Delta_{t-j}^{(m)}$
        $m = \arg\max_{m \in \mathcal{M}} \left\{ \beta^{(m)} \mathcal{P}_{g_t}^{(m)}(t) \right\}$
    end if
    Use planner $m$ to select top-level region $G_t$
    Execute one decision step, observe $(s_{t+1}, r_{t+1})$
    $\Delta_{t+1}^{m} \leftarrow r_{t+1} - r_t$
    $\mathcal{B}_{g_t}^{(m)} \leftarrow (\Delta_{t+1}^{(m)})$        ▷ (drop oldest if $|\mathcal{B}_{g_t}^{(m)}| > n$)
    $t \leftarrow t+1$, $s_t \leftarrow s_{t+1}$, $r_t \leftarrow r_{t+1}$
end while
```

The pseudo code is shown in Alg.1, we notice that at timestep t, if a region g_t has already been well explored,

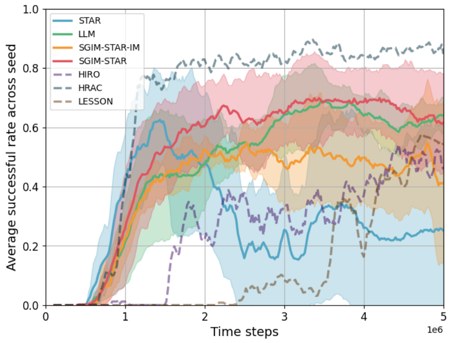
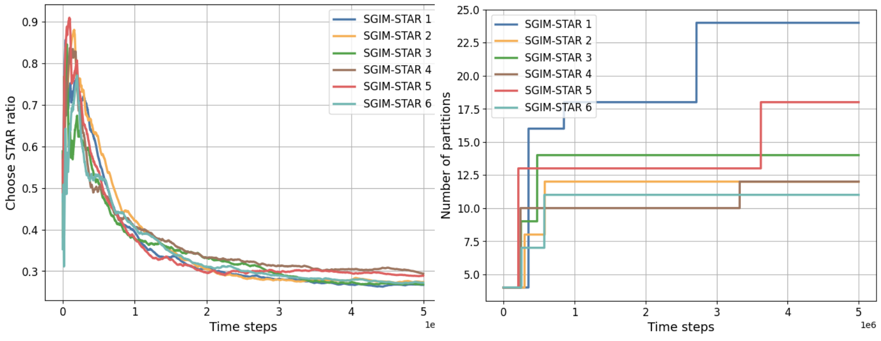


Fig. 4: Average successful rate of SGIM-STAR compared to baselines and three HRL methods



(a) The ratio of leveraging STAR as the top-level planner

(b) The Partitions of space

Fig. 5: During the learning process of SGIM-STAR

then both planners yield low incremental progress ($\Delta_t^{(m)} \approx 0$), resulting in small accumulated progress scores $\mathcal{P}_{g_t}^{(m)}(t)$. Conversely, when the agent enters a novel region, progress signals tend to be larger, biasing selection toward the planner that has demonstrated stronger improvement in unexplored regions. This mechanism naturally encourages exploitation of novel areas while reducing reliance on planners that fail to generate additional progress in familiar regions.

VI. Performance of SGIM-STAR

A. Experiment Setup

We evaluate our proposed algorithm, SGIM-STAR in the Ant Maze environment, in which the robot must navigate a $\supset$-shaped maze and reach an exit located at the top left corner. This task is inherently hierarchical: success requires both fine-grained locomotion control (low-level) and long-horizon navigation through the maze (top-level). Moreover, Ant Maze is a suitable benchmark for evaluating LLM integration, as solving the task requires reasoning over abstract spatial regions and selecting long-horizon subgoals rather than relying solely on local control.

We compare our SGIM-STAR with the following methods:

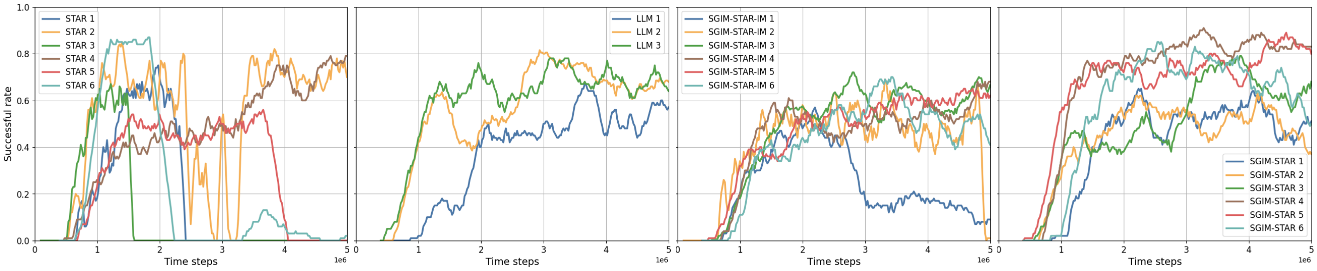
- **STAR**: the original STAR framework where the top-level agent is the Commander policy trained via Q-learning.
- **LLM Planner**: STAR algorithm where the Commander is replaced by an LLM using a handcrafted prompt. The Tutor and Controller remain unchanged.
- **SGIM-STAR-IM** (SGIM-STAR with Interactive learning at the Meta level) : to study the importance of the partition, we considered an ablation where the top-level agent adaptively switches between STAR and LLM, but without considering the environment partitions defined by the STAR abstraction, as with the algorithm SGIM-IM [31]: instead of computing $\mathcal{P}_{pt}^{(m)}$ for each region, we consider it for the whole state space.

All agents are trained on one NVIDIA GEFORCE RTX 4090 GPU, and we track their success rates over 5M environment steps on 6 random seeds for SGIM-STAR, SGIM-STAR-IM and STAR, and 3 seeds for LLM Planner.

B. Results and Analyses

Fig. 4 shows the average success rate across random seeds for SGIM-STAR. We compare its performance with STAR, LLM and SGIM-STAR-IM in ablation studies. For reference, we also trace the average success rate of other HRL methods but using the same seed, so they face less variability than for STAR, SGIM-STAR, SGIM-STAR-IM and LLM: HIRO [16], HRAC [17] and LESSON [3]. We notice that although some HRL method achieve competitive performance, all of them don't construct a discrete representation of the space, which makes direct integration with LLM infeasible. We observe that the partition-based SGIM-STAR not only attains the greatest success rate of 0.7 by the end of training but also exhibits the smallest variance across seeds, indicating consistent learning outcomes. In contrast, the other methods reach lower success levels and have wider fluctuations. Notably, the LLM-only agent plateaus around a moderate success rate at around 0.6, while the pure STAR agent's average performance degrades significantly by the end of training due to collapses in some runs. These results demonstrate that incorporating partitioned task structure and adaptively integrating LLM guidance produces superior resilience in this long-horizon task.

To further analyze stability, we examine individual training curves of each method. The pure HRL baseline STAR, as shown in Fig. 6a, collapses frequently mid-training across seeds. This instability indicates a lack of resilience: the convergence of STAR is not guaranteed when learning such a complex, long-horizon task without additional guidance. Figure 6b shows that the LLM-only agent can reach moderate success rates plateau without any dropping of performance, demonstrating the potential of a pretrained planner to guide exploration. However, the use of LLM-planner is too costly and the learning process of LLM-only is four times slower than the others, which limits the further use of LLM in the top-level agent. Fig. 6d shows that SGIM-STAR demonstrates remarkably consistent improvement across training, with almost no catastrophic drops in performance. In contrast, the SGIM-STAR-IM variant also suffers abrupt performance collapses after initial learning spurts from Fig,6c. This



(a) STAR (b) LLM (c) SGIM-STAR-IM (d) SGIM-STAR

Fig. 6: Successful rate of each run in (a) STAR, (b) LLM, (c) SGIM-STAR-IM and (d) SGIM-STAR.

suggests that state-space partitioning plays a critical role in stabilizing long-horizon learning.

A key insight is that SGIM-STAR selectively shifts to LLM guidance once it becomes beneficial. Early on, STAR's fast learning of basic navigation yields quicker gains, so the agent heavily favors the STAR Commander policy. As training progresses, the agent discovers a more refined symbolic structure of the maze via partitions, it increasingly relies on the LLM for top-level planning. Fig. 5a quantifies this behavior: at the start of training, the fraction of top-level choices directed by STAR is extremely high, as the task structure emerges, this fraction steadily declines, and by the end of training the agent chooses STAR for only one third of decisions on average, indicating that it has shifted to predominantly following the LLM's guidance in later stages. In other words, SGIM-STAR intelligently "chooses" STAR in the beginning when the LLM's abstract guidance might not yet be enough, and then gradually transitions to the LLM as the partitions learned by STAR provide a reliable framework for planning. This adaptive scheduling of who controls the top-level actions is crucial to achieving both high efficiency and stability.

Another contributing factor to the robustness of the SGIM-STAR is the growth of its state abstraction over time. During training, the SGIM-STAR incrementally partitions the state space into more regions as it encounters new situations. Fig. 5b tracks the number of partitions in each run over the course of training. In effect, the partitioning mechanism provides a form of symbolic memory that the LLM can leverage which grounds the LLM's planning in the agent's learned experience.

VII. Discussion

After showing a possible parallel between natural language instructions and internal representations, which hints at a grounding of LLMs in an emergent representation, we introduced SGIM-STAR which selectively combines an RL-based planner with an LLM-based planner using a partition-wise, progress-driven rule. Our analysis yields four key characteristics of our method:

(1) Mutual stabilization and lighter planning. LLM guidance stabilizes STAR by providing supplementary top-level proposals when the RL-based planner becomes brittle, while STAR makes the overall system lighter than an LLM-only planner by supplying competent, inexpensive planning during large portions of training. Overall, the RL-based planner constitutes a bottom-up agent learning from its trial and error with the environment, whereas the LLM planner is a top-down agent sharing its internal world representation to this specific task. Their combination mutualizes both a bottom-up and a top-down process. The intrinsically motivated, progress-based selection between the two planners improves learning progress and stabilizes performance.

(2) Cost-aware usage of the LLM. SGIM-STAR uses the LLM only when necessary: calls to the LLM are conditional on partition-wise progress and thus avoided when the STAR Commander suffices. Compared to an LLM top-level agent, this conditional usage reduces planner cost and latency while still reaping the benefits of LLM exploration.

(3) Start planning with RL, then switch to LLM. The agent relies more on a RL planner in the early phase—when the internal representation is coarse—and gradually shifts toward LLM guidance as the internal representation becomes richer and more meaningful for language reasoning. This indicates that a richer internal representation can offer a better grounding of LLM planners.

(4) Formulation that enables language grounding. Crucially, our learning formulation builds a discretized, partitioned representation—from bottom-up RL experiences. This evolving symbolic structure leverages LLMs to help the learning process of the agent, by offering a grounded correspondence of regions to LLM symbols.

VIII. Conclusion and Perspective

In this work, we investigated how LLMs can interact with the emergent symbolic representations produced by hierarchical reinforcement learning. Through translation experiments in Ant Maze, we found that while LLMs can map natural language instructions to coarse symbolic partitions, their reliability decreases with finer granularity. Motivated by this translation analysis, we proposed *SGIM-STAR*, a simple, partition-wise progress-based algorithm that switches between a STAR top-level planner and an LLM planner. This design leverages the efficiency of RL in early stages and selectively invokes LLM guidance in the later ones, since the richer representation gives the LLM more meaningful inputs. Experiments show that SGIM-STAR not only improves long-horizon task success and stability over STAR but also reduces the computational cost associated with LLM-only planning.

Our study suggests that LLMs and RL agents together can solve more complex tasks than either alone. In future works, we plan to extend this framework to real-world robotics scenarios where humans instruct robots in natural language during its learning process. Successfully transferring the method to a robotic platform would validate its generality and robustness, while highlighting any necessary adaptations for real-world operation. Moreover, the cost-aware switching mechanism introduced in SGIM-STAR demonstrates a more efficient use of LLMs in planning – future systems could dynamically invoke language-based planners only when needed, keeping operation costs low. Finally, by tackling more diverse tasks, the learned symbolic plan can capture richer semantics, which enables the LLM planner to interpret and execute a broader range of instructions, further improving long-horizon performance.

Overall, our study demonstrates that emergent symbolic representations can serve as a grounding substrate for language, enabling LLMs to act both as translators of human instructions and as adaptive planners. By unifying bottom-up learning with top-down symbolic reasoning, SGIM-STAR advances the development of language-grounded hierarchical reinforcement learning for robotics, paving the way toward more interpretable, robust, and human-interactive agents in long-horizon tasks.

REFERENCES

[1] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International conference on machine learning*. PMLR, 2016, pp. 1329–1338.

[2] A. G. Barto and S. Mahadevan, "Recent advances in hierarchical reinforcement learning," *Discrete Event Dynamic Systems*, vol. 13, no. 1, pp. 41–77, Jan 2003.

[3] S. Li, L. Zheng, J. Wang, and C. Zhang, "Learning subgoal representations with slow dynamics," in *International Conference on Learning Representations (ICLR)*, 2021. [Online]. Available: https://openreview.net/forum?id=wxRwhSdORKG

[4] M. Zadem, S. Mover, and S. M. Nguyen, "Reconciling spatial and temporal abstractions for goal representation," in *The Twelfth International Conference on Learning Representations*, 2024.

[5] A. Ahmetoglu, M. Seker, J. Piater, E. Oztop, and E. Ugur, "Deepsym: Deep symbol generation and rule learning for planning from unsupervised robot interaction," *Journal of Artificial Intelligence Research*, vol. 75, pp. 709–745, 11 2022.

[6] Y. Shentu, P. Wu, A. Rajeswaran, and P. Abbeel, "From llms to actions: Latent codes as bridges in hierarchical robot contro," in *IROS*, 2024.

[7] S. Chakraborty, K. Weerakoon, P. Poddar, P. Tokekar, A. S. Bedi, and D. Manocha, "Re-move: An adaptive policy design approach for dynamic environments via language-based feedback," *ArXiv*, vol. abs/2303.07622, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:263893735

[8] D.-S. Jang, D.-H. Cho, W.-C. Lee, S.-K. Ryu, B. Jeong, M. Hong, M. Jung, M. Kim, M. Lee, S. Lee, and H.-L. Choi, "Unlocking robotic autonomy: A survey on the applications of foundation models," *International Journal of Control, Automation and Systems*, vol. 22, no. 8, pp. 2341–2384, Aug 2024.

[9] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "Palm-e: An embodied multimodal language model," 2023. [Online]. Available: https://arxiv.org/abs/2303.03378

[10] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," in *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.

[11] H. Hu, D. Yarats, Q. Gong, Y. Tian, and M. Lewis, "Hierarchical decision making by generating and following natural language instructions," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[12] T. Carta, C. Romac, L. Gaven, P.-Y. Oudeyer, O. Sigaud, and S. Lamprier, "Herakles: Hierarchical skill compilation for open-ended llm agents," 2025. [Online]. Available: https://arxiv.org/abs/2508.14751

[13] K. Jokinen, "The need for grounding in LLM-based dialogue systems," in *Proceedings of the Workshop: Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning (NeusymBridge) @ LREC-COLING-2024*, T. Dong, E. Hinrichs, Z. Han, K. Liu, Y. Song, Y. Cao, C. F. Hempelmann, and R. Sifa, Eds. Torino, Italia: ELRA and ICCL, May 2024, pp. 45–52.

[14] J. Schmidhuber, "Curious model-building control systems," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 2, 1991, pp. 1458–1463.

[15] J. Gottlieb, P.-Y. Oudeyer, M. Lopes, and A. Baranes, "Information-seeking, curiosity, and attention: computational and neural mechanisms," *Trends in Cognitive Sciences*, vol. 17, no. 11, pp. 585–593, 10 2013.

[16] O. Nachum, S. S. Gu, H. Lee, and S. Levine, "Data-efficient hierarchical reinforcement learning," *Advances in neural information processing systems*, vol. 31, 2018.

[17] T. Zhang, S. Guo, T. Tan, X. Hu, and F. Chen, "Generating adjacency-constrained subgoals in hierarchical reinforcement learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020, spotlight. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/f5f3b8d720f34ebebceb7765e447268b-Abstract.html

[18] M. Zadem, S. Mover, and S. M. Nguyen, "Goal space abstraction in hierarchical reinforcement learning via set-based reachability analysis," in *2023 IEEE International Conference on Development and Learning (ICDL)*, Nov 2023, pp. 423–428.

[19] P.-Y. Oudeyer and F. Kaplan, "What is intrinsic motivation? a typology of computational approaches," *Frontiers in Neurorobotics*, vol. 1, no. 6, 2007.

[20] P.-Y. Oudeyer, F. Kaplan, V. Hafner, and A. Whyte, "The playground experiment: Task-independent development of a curious robot," in *Spring Symposium on Developmental Robotics*. AAAI, 2005, pp. 42–47.

[21] A. Baranes and P.-Y. Oudeyer, "Active learning of inverse models with intrinsically motivated goal exploration in robots," *Robotics and Autonomous Systems*, vol. 61, no. 1, pp. 49–73, 2013.

[22] C. O. Retzlaff, S. Das, C. Wayllace, P. Mousavi, M. Afshari, T. Yang, A. Saranti, A. Angerschmid, M. E. Taylor, and A. Holzinger, "Human-in-the-loop reinforcement learning: A survey and position on requirements, challenges, and opportunities," *Journal of Artificial Intelligence Research*, vol. 79, pp. 359–415, Jan. 2024.

[23] M. Zare, P. M. Kebria, A. Khosravi, and S. Nahavandi, "A survey of imitation learning: Algorithms, recent developments, and challenges," *IEEE Transactions on Cybernetics*, 2024.

[24] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 49–58.

[25] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 80, 2018. [Online]. Available: https://proceedings.mlr.press/v80/fu18a.html

[26] A. Shon, D. Verma, and R. P. Rao, "Active imitation learning," in *American Association for Artificial Intelligence*, vol. 22. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007, p. 756.

[27] S. Ross, G. J. Gordon, and J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the 14th International Conference on Artificial*

Intelligence and Statistics (AISTATS), 2011, pp. 627–635. [Online]. Available: https://proceedings.mlr.press/v15/ross11a.html

[28] S. M. Nguyen and P.-Y. Oudeyer, "Active choice of teachers, learning strategies and goals for a socially guided intrinsic motivation learner," *Paladyn, Journal of Behavioral Robotics*, vol. 3, no. 3, pp. 136–146, 2012.

[29] N. Duminy, S. M. Nguyen, J. Zhu, D. Duhaut, and J. Kerdreux, "Intrinsically motivated open-ended multi-task learning using transfer learning to discover task hierarchy," *Applied Sciences*, vol. 11, no. 3, 2021.

[30] Y. Jiang, S. S. Gu, K. P. Murphy, and C. Finn, "Language as an abstraction for hierarchical deep reinforcement learning," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[31] S. M. Nguyen and P.-Y. Oudeyer, "Interactive learning gives the tempo to an intrinsically motivated robot learner," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, 2012, pp. 645–652.