A Bilevel Reinforcement Learning Framework with Language Prior Knowledge

Xue Yan[1,2], Yan Song[3], Xinyu Cui[1,2], Filippos Christianos[4], Haifeng Zhang[1,2],
Jun Wang[3] (🖂), and David Mguni[5] (🖂)

[1] The Key Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, China
{yanxue2021,cuixinyu2021,haifeng.zhang}@ia.ac.cn
[3] AI Centre, Department of Computer Science, University College London, London,
UK yan.song.24@ucl.ac.uk;jun.wang@cs.ucl.ac.uk
[4] Huawei Technologies, London, UK filippos.christianos@gmail.com
[5] Queen Mary University, London, UK davidmguni@hotmail.com

Abstract. Large language models (LLMs) demonstrate their promise in tackling complicated practical challenges by combining action-based policies with chain of thought (CoT) reasoning. Having high-quality prompts on hand, however, is vital to the framework's effectiveness. Currently, these prompts are handcrafted utilising extensive human labor, resulting in CoT policies that frequently fail to generalise. Human intervention is also required to develop grounding functions that ensure low-level controllers appropriately process CoT reasoning. In this paper, we propose a comprehensive end-to-end training framework for complex task-solving that utilises language prior knowledge embedded within LLMs or from human experts. To that purpose, we offer a new leader-follower reinforcement learning framework that incorporates a prompt policy, a CoT process, and an action policy. The prompt policy is employed to ask pertinent questions based on historical observations, leading the CoT process to consider the anticipated goals and generate state-adaptive thoughts that lead to decisive, high-performing actions. To induce these high-quality actions, the prompt policy has its own objective in our system, encouraging it to adapt to the behavior of the action policy. The action policy subsequently learns to comprehend and integrate the CoT outputs to take precise actions. Empirical results demonstrate that our framework outperforms leading methods in 6 popular decision-making benchmark environments, including Overcooked and ALFWorld.

Keywords: Reinforcement learning · Bilevel optimisation · Language priors.

1 Introduction

Large language models (LLMs) with Chain-of-thought (CoT) prompts [29, 28] have achieved impressive performance improvements for solving complex natural language processing (NLP) tasks. Moreover, techniques such as reward incentives and tree search [32, 12] have enhanced the quality of CoT reasoning for addressing intricate decision-making tasks, ultimately inducing the step-by-step problem-solving process. This involvement of CoT reasoning has given rise to the promise of unlocking the power of LLMs to be able to assist in performing complex reasoning and acting in real-world environments.

While LLMs such as GPT-4 possess a wealth of human knowledge, in general, current prompt-engineering based language agents [16, 32] and prior knowledge distillation approaches [36, 33] heavily depend on meticulously crafted prompts designed by humans for each specific task. The dependence on high-quality, task-specific crafted prompts limits the generalization of these methods, while manually designing (high-quality) prompts is an arduous and expensive task. Additionally, despite the obvious potential of using CoT reasoning for guiding a low-level control policy, human-intelligible CoT reasoning can often be ambiguous for a downstream control policy, such as a rule-based planning method [34, 23] and an action policy trained by a reinforcement learning (RL) algorithm [4, 27]. As such, a natural consideration is the need to generate CoT outputs that are interpretable to the action policy, and provably reduce the uncertainty of the action policy in making decisions. Therefore, the ambition of embedding CoT reasoning appropriately within a generalist artificial intelligence (AI) framework has presented a series of critical challenges that have yet to be fully resolved.

In this paper, we propose a fully unified decision-making framework that adaptively incorporates CoT reasoning to assist in tackling complex tasks. In order to achieve this goal, both the prompt design and the action policy to be executed have to be sufficiently flexible and useful so as to adapt to the current task at hand. To this end, we introduce a comprehensive end-to-end decision-making framework that follows the *question, reason, then act* pipeline. Specifically, it learns to ask pertinent *questions*, performs CoT reasoning, and then learns to take the best actions in the environment. The first component of the framework is enacted by a *prompt policy* that learns a suitable prompt question given the environment observations. These prompts serve as inputs to a *CoT process*, allowing the framework to perform state-related and meaningful reasoning. The CoT outputs are then integrated into the *action policy*, which learns to find solutions to tasks that may require both interaction experience and human knowledge embedded in CoT reasoning to solve.

Learning how to generate in-demand prompts for the CoT process produces formidable challenges. One such challenge is to ensure that the resulting CoT outputs enhance the performance of an action policy. We resolve this challenge by designing a *leader-follower Bilevel* optimisation [19] structure, called Bilevel-LLM and illustrated in Figure1, that generates mutually adaptive policies. Each policy is endowed with its own objective — the prompt policy observes the effect of its prompt and subsequent CoT reasoning on the action policy and learns to generate useful prompts. In particular, the prompts are chosen so as to minimise the uncertainty of the action policy i.e. minimise the entropy of the action policy. The action policy, on the other hand, learns to maximise the environmental reward while taking into account the outputs of the CoT process. Ultimately, the generated thoughts serve to learn a more effective action policy, providing additional information beyond the observation of the environment. These natural language insights embody human knowledge, reducing the need for redundant exploration compared to traditional RL algorithms, which typically require extensive exploration of specific environments to train a competent agent.

To minimise human intervention in task-related prompt design, we implement a prompt policy based on a set of predefined prompt candidates. This approach also helps avoid the dilemma of the scarcity of high-quality, supervised data for prompt generation and the instability risks associated with exploring an unrestricted prompt space. In many task environments, expert prompt data is available, such as well-defined sets of subtasks [23, 30]. In environments where such prompt candidates are not available, our experimental results have shown that GPT-3.5 can generate high-quality prompts based on task descriptions, enabling Bilevel-LLM to achieve comparable performance to that rely on human-written prompts. Additionally, we demonstrate that Bilevel-LLM successfully learns to select the state-adaptive prompt from a global set of candidates.

The contributions of this paper can be summarised as follows:

- A new framework for dynamically adjusting prompts for decision-making tasks. An integral component is a prompt policy trained to select prompts that induce low uncertainty in the action policy, which receives thoughts generated by the CoT process triggered by the prompts. Therefore, the prompt policy (and hence the CoT process) behaves adaptively toward the needs of the action policy.
- Embedding CoT reasoning into the resolution of complicated decision-making tasks, where the outputs of the CoT process guide a policy that takes actions within an environment. This leverages the benefits of natural language models and CoT reasoning that encapsulate worldly experience and the capacity for deductive reasoning, while efficiently tuning the thought process by adjusting the prompt policy.
- A new bilevel optimisation framework that integrates prompt-tuning with the learning of a CoT output-based action policy. In this framework, the prompt and action policies mutually influence each other and are concurrently trained to converge.

2 Problem Formulation

In this setting, an agent aims to solve some task by performing a sequence of actions in an environment. Formally, the problem is described by a partially observable Markov decision process (POMDP), which is defined by the following tuple $\langle \mathcal{S}, \mathcal{A}, P, \mathcal{O}, T, \mathcal{R}, \rangle$, where $\mathcal{S}$ is the finite set of environment states, $\mathcal{A}$ is the set of actions for the agent, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the state transition kernel for the environment, $\mathcal{O}$ is the finite set of observations. The states, observations, and actions can be described in natural language. The function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, which returns a scalar reward conditioned on a state-

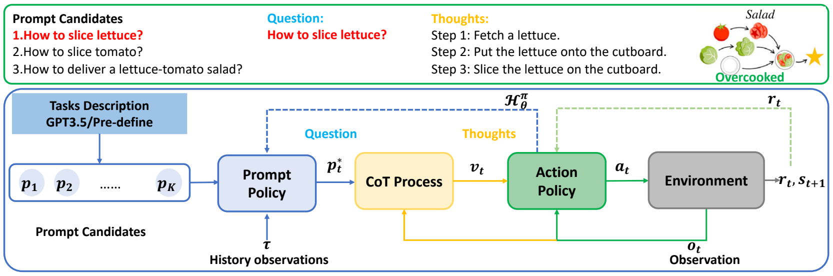


Fig. 1. *Top:* Example of the workflow from prompt candidates to CoT reasoning on Overcooked. The prompt policy first selects a prompt question from the candidate set. Subsequently, the CoT process generates complex reasoning guided by the prompt and the current state observation to assist in subsequent action performing. *Bottom:* The illustration of our bilevel optimisation framework.

action pair whose realisation at time step t we denote by $r_t \sim R$. Lastly, the observation function is $T : \mathcal{S} \times \mathcal{A} \to \mathcal{O}$ which is a mapping from the environment state, action to the observation set. Since the exact form of the observation and state spaces varies between environments, we provided a general description of the POMDP setting for introducing the general problem setting. [6] In complex decision-making problems, standard methods such as RL struggle to solve these tasks in a sample efficient way. To solve these problems, an agent may required to deductive reasoning in order to resolve the challenge of finding an optimal policy. To tackle these challenges, we propose a bilevel decision-making framework as illustrated in Figure1, which can be split into three components:

• First, a *prompt policy* $\pi_\phi : (\mathcal{O})^{j<\infty} \to \Delta(\mathcal{P})$. Denote the $\mathcal{P}$ as the prompt space containing finite prompt questions. This policy learns to tune prompts after observing (a window of) $j < \infty$ observations.

• Second, a *CoT process* $\pi^{\mathrm{re}} : \mathcal{O} \times \mathcal{P} \to \boldsymbol{\mathcal{T}}$— a fixed language model that reasons about the task at the particular state observations and prompts questions. Denote that $\boldsymbol{\mathcal{T}}$ is the space of textual sentences based on the vocabulary set $\mathcal{V}$ (with finite words in it). Bilevel-LLM is a plug-and-play framework that supports various options, including universal LLMs, task-specific knowledge distillation [36], and environment-provided expert feedback [6], for performing CoT reasoning. In our experiments, we employ either GPT-3.5 [7] or expert feedback provided by the environment [6]. Examples of prompt questions and CoT reasoning are illustrated in Figure1.

• Lastly, an *action policy* $\pi_\theta : \mathcal{O} \times \boldsymbol{\mathcal{T}} \to \Delta(\mathcal{A})$. The action policy takes the observation of the environment and the CoT thought as inputs then executes actions in the environment.

Concretely, at times $t = 0, 1, \ldots$, a prompt p_t is selected by the prompt policy i.e. $p_t \sim \pi_\phi(\cdot|o_t, \ldots, o_{t-j\wedge 0})$. The prompt is then used by the CoT process whose

[6] In the MDP setting, the observation space is equivalent to the state space.

[7] The version of GPT-3.5 used in this work is GPT-3.5-turbo

output is a thought $\upsilon_t \in \mathcal{T}$. Last, the action policy samples an action given the observation and the thought $a_t \sim \pi_\theta(\cdot|o_t, \upsilon_t)$. Therefore, the sequence of events proceeds as follows:
1. At time t, the system is at an environment state $s_t \in \mathcal{S}$.
2. A prompt p_t is chose by the prompt policy i.e. $p_t \sim \pi_\phi(\cdot|o_t, \ldots, o_{t-j\wedge 0}),\ p_t \in \mathcal{P}$.
3. An action $a_t \sim \pi_\theta(\cdot|o_t, \upsilon_t)$ is taken given the output of the CoT process $\upsilon_t \sim \pi^{\mathrm{re}}(p_t, o_t)$.
4. The environment state transitions according to $s_{t+1} \sim P(\cdot|s_t, a_t)$. Figure**??** in Appendix shows a step by step inference example of Bilevel-LLM on the Overcooked task.

To tackle the problem of learning how to tune prompts while learning the action policy, we structure the problem as a leader-follower *bilevel optimisation* [7]. This allows the prompt policy to learn how its decisions affect the action policy while the action policy learns both how to interpret the CoT outputs and take desirable actions. Since LLMs already contain a vast amount of world knowledge, we here fix the CoT process π^{re}. We update the prompt policy and action policy concurrently. The prompt policy aims to precisely adjust prompts minimise the uncertainty of the action policy, while the action policy aims to maximise the environmental return, taking the CoT outputs into account. The optimisation objective can be expressed as a bilevel optimisation problem:

$$\begin{aligned}(\pi_\theta^*, \pi_\phi^*) \in \arg\max&{}_{(\pi_\theta,\pi_\phi)\in\Pi_\theta\times\Pi_\phi} \\ &\mathbb{E}_{\pi_\theta,\pi_\phi,\upsilon_t\sim\pi^{\mathrm{re}}}\left[-\sum_{t\geq 0}\gamma^t\mathcal{H}^{\pi_\theta}(y_t)|y_t=(o_t,\upsilon_t)\right] \\ \text{s.t. } \pi_\theta^* \in \arg\max&{}_{\pi_\theta\in\Pi_\theta}\,\mathbb{E}_{\pi_\theta,p_t\sim\pi_\phi,\pi^{\mathrm{re}}}\left[\sum_{t\geq 0}\gamma_I^t r_t\right],\end{aligned}$$

where $\mathcal{H}^{\pi_\theta}(y_t) = -\sum_{a_t\in\mathcal{A}}\pi_\theta(a_t|y_t)\log\pi_\theta(a_t|y_t)$ is the entropy of the action policy π_θ, $y_t = (o_t, \upsilon_t)$, and $\gamma_I, \gamma \in [0,1)$ are the discount factors for the action and prompt generation policies respectively and $r_t \sim \mathcal{R}$ is the environment reward. Here, we explain the bilevel optimisation:
In the inner loop, the action policy π_θ learns to take optimal actions, i.e. to maximisie environment reward, based on both observations and CoT reasoning, which contains task-solving prior knowledge.
In the outer loop, the prompt policy π_ϕ aims to minimise the entropy of the action policy. *The motivation for using the negative entropy of the action policy as a objective can be explained as follows.* It learns to find appropriate prompts that subsequently lead to CoT reasoning, enabling the action policy to take high-performing actions more certainly.

Algorithm 1 Bilevel-LLM

```
Input: Initialise parameters of policies $\pi_\theta$, $\pi_\phi$. Prompt candidate set $\mathcal{P}$. Set the data
    buffer $D = \emptyset$.
Output: $\pi_\theta^*$, and $\pi_\phi^*$.
while not done do
    #Rollout trajectories with $\pi_\theta, \pi^{\mathrm{re}}, \pi_\phi$.
    for $i = 1, 2, .., \mathrm{step}$ do
        Generate prompt given historical observations: $p_t \sim \pi_\phi(\cdot|o_t, \ldots, o_{t-j\wedge 0})$.
        Perform CoT reasoning given prompt and observation: $\upsilon_t \sim \pi^{\mathrm{re}}(\cdot|p_t, o_t)$.
        Sample action according to the CoT reasoning and observation: $a_t \sim$
            $\pi_\theta(\cdot|o_t, \upsilon_t)$.
        Apply action $a_t$ to the environment, sample the reward $r_t$ and next step
            observation $o_{t+1}$.
        Calculate the entropy of the action policy: $h_t = \mathcal{H}\left(\pi_\theta(\cdot|s_t, \upsilon_t)\right)$.
        Add to data buffer: $D = D \cup (o_t, p_t, \upsilon_t, a_t, r_t, h_t, o_{t+1})$
    end for
    Update the action policy $\pi_\theta$ by optimising Eq. (2).
    Update the prompt policy $\pi_\phi$ by optimising Eq. (1).
end while
```

3 Methodology

In this section, we describe the training procedure of the proposed bilevel framework. The action policy is optimised to maximise environmental rewards, while the prompt policy is designed to assist the action policy in performing optimal actions more certainly by minimising its entropy. In the bilevel framework, the prompt and action policies are concurrently optimised until convergence. The overall framework is illustrated in Figure1.

Prompt Policy Training. When meticulously crafted prompts are provided, CoT reasoning has proven to be effective in aiding decision-making tasks. However, crafting prompts that effectively trigger reasonable CoTs for various long-term decision-making tasks is challenging, given the vast state space and the multifaceted skills these tasks demand. Therefore, we aim to develop a prompt policy that can dynamically adjust the prompts for different states while minimising reliance on extensive human labor.

Due to the limited availability of supervised data for high-quality prompts and the potential instability of exploring an unlimited prompt space, we opt not to train a model to autonomously generate prompts. Instead, we use predefined prompt candidates for each specific task, crafted either by humans or generated by powerful LLMs like GPT-3.5 with task descriptions given. Additionally, we conducted an experiment comparing the performance of our method using LLM-generated prompts with those designed by humans, as shown in Figure 6(b).

Given a prompt candidate set $\mathcal{P} = \{p^{(1)}, p^{(2)}, \cdots p^{(K)}\}$, we aim to train a prompt policy that selects state-adaptive prompts from the candidate set. To implement the prompt policy, we use a pre-trained LLM, Flan-T5 Small [21] or

Flan-T5 Large in this paper, as the backbone. The prompt LLM policy selects an appropriate prompt question for the current state given historical observation information and prompt candidates as inputs. This process is formally represented as: $p_t \sim \pi_\phi(\cdot|o_t, \ldots, o_{t-j\wedge 0}, \mathcal{P})$. For simplicity, we denote this as $p_t \sim \pi_\phi(\cdot|o_t, \ldots, o_{t-j\wedge 0})$, ignoring the $\mathcal{P}$ which are the same for all states. The prompt policy is updated via PPO, with the negative entropy of the action policy serving as the reward. The detailed procedure is described as below:

- For a decision-making task, we employ GPT-3.5, along with the provided task description, to generate appropriate prompt candidates. As a second case, we utilise the natural subtask structure [30] and human-crafted assists to generate valuable prompt candidates.
- With these K prompts, the prompt policy is optimised to maximise the minus action policy entropy. The objective function is given by:

$$\arg\max_{\phi} \mathbb{E}_{\pi_\theta, \pi_\phi, \upsilon_t \sim \pi^{\mathrm{re}}} \left[-\sum_{t\geq 0} \gamma^t \mathcal{H}^{\pi_\theta}(y_t) | y_t = (o_t, \upsilon_t) \right] \tag{1}$$

When optimising the prompt policy π_ϕ through Eq. (1), we update only the parameters ϕ, keeping the action policy π_θ fixed. The entropy $\mathcal{H}^{\pi_\theta}(y_t)$ of the action policy is treated as a scalar, non-differentiable reward.

CoT Reasoning with Prompts. With the selected prompt p_t, the CoT reasoning is obtained by $\upsilon_t \sim \pi^{\mathrm{re}}(\cdot|p_t, o_t)$, where the CoT process π^{re} is implemented by a powerful LLM such as GPT-3.5 or environment-integrated language feedback (for ALFWorld [6]). The motivation of integrating the CoT reasoning into our bilevel framework is to use human prior knowledge to provide a high-level guideline for solving complicated decision-making tasks. For example, as shown in Figure1, in the Overcooked game, the CoT process can generate a sequence of solving steps -“picking up the lettuce, placing it on the cutting board, and then slicing it"- related to a prompt question “how to slice lettuce”.

During implementation, to reduce the inference time and costs for frequent queries, we store the CoT outputs for each state. Additionally, in Overcooked, which has a vast state space (up to 9.8×10^{21}), we abstract states into representative situations via a rule-based method and store CoT outputs accordingly. Specifically, we preserve the materials situation while disregarding the items' map positions.

Action Policy Training. Existing works [15, 4, 36] utilise LLMs as the action policy and fine-tune these LLMs to adapt to decision-making tasks, taking advantage of the comprehensive capabilities of LLMs. In our work, we also utilise an LLM as the action policy. In implementation, the action LLM takes the textual observations and the CoT reasoning as input and output an distribution over the action space. To regulate the action LLM to output executable actions, we fine-tune the action LLM, denoted as π_θ, using PPO [22]. The objective of the action policy is to maximise the environmental return:

$$\arg\max_{\theta} \mathbb{E}_{a_t \sim \pi_\theta, p_t \sim \pi_\phi, \upsilon_t \sim \pi^{\mathrm{re}}} \left[\sum_{t \geq 0} \gamma_I^t r_t \right] \tag{2}$$

We use the same pre-trained LLM as the backbone for both the prompt and action policies. The PPO algorithm is employed to train both policies, with the prompt policy aiming to minimize the entropy of the action policy. Despite this, the exploration ability of the action policy is still preserved, thanks to the exploration-encouraging term in PPO.

Bilevel Optimisation. In our leader-follower Bilevel LLM framework, the prompt policy and the action policies are trained alternately, with the other policy being kept frozen. On the one hand, the prompt policy selects an appropriate prompt for the CoT process, the output of which is expected to assist the action policy in solving complex tasks. Thus, the goal of the prompt policy is to reduce the uncertainty of the action policy when it encounters challenging scenarios. On the other hand, the action policy is trained to effectively solve specific decision-making tasks while benefiting from CoT reasoning and the experience gathered during exploration. The overall training process of the Bilevel framework is detailed in Algorithm 1.

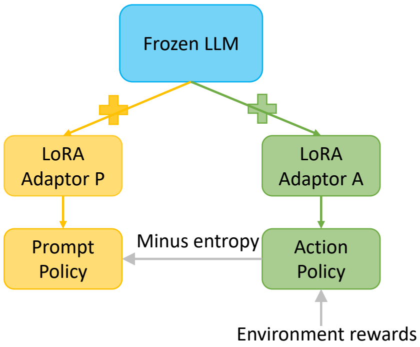


Fig. 2. The training structure of Bilevel-LLM using LoRA.

Inspired by the success of LoRA [14], which achieves comparable performance by training only a few parameters instead of fine-tuning all parameters, and considering that our prompt and action policies are based on the same LLM, we train two different LoRA adaptors for the two policies to enhance computational efficiency. The overall training structure of Bilevel-LLM is illustrated in Figure 2.

4 Experiments

In this section, we validate that our bilevel framework, which integrates prompt tuning, CoT reasoning, and action policy learning, is beneficial for decision-making. Additionally, our bilevel framework supports utilising prompt candidates from GPT-3.5 while also automatically interpreting CoT reasoning, thereby avoiding the need for extensive human labor compared to prompt-engineering-based LLM agents [20, 34]. Further details of experimental settings, such as hyperparameter and environment settings, and more ablation study results can be found in Appendix.

4.1 Environments

In this work, we incorporate six language decision-making environments involving various skills, including reasoning and navigation abilities, detailed statistics are shown in Table **??** in Appendix C. The six environments are the following: **Tower of Hanoi** [13], a classical logic reasoning game. **Frozen Lake** and **ChainWorld** are POMDP environments, where only the agent's position is accessible. **FourRoom** is a POMDP task, where the agent should navigate through hallways to reach the goal. **ALFWorld** is a widely recognized benchmark for LLM agents [32, 24], is also a POMDP task. We utilise the environment as implemented by LLF-Bench, which provides off-the-shelf prompt questions and language feedback. We compare baselines across 50 tasks. For **Overcooked**, we consider three different layouts: *Overcooked(Tomato)*: deliver a chopped tomato with the map size of 5×4; *Overcooked(Salad)*: deliver a tomato-lettuce salad with a map size of 5×4; *Overcooked(Large)*: deliver tomato-lettuce salad with a map size of 7×7. Note that the state space of *Overcooked(Large)* reaches 9.8×10^{21}, making it challenging to explore. Standard RL environments, such as Frozen Lake and Overcooked, are converted into text-based environments using predefined rule-based transitions. More detailed environment descriptions can be found in Appendix C.

4.2 Baselines

We evaluate Bilevel-LLM against five baselines: **GFlan**, which utilises the Flan-T5 model as the action policy and optimises it with PPO based on textual state representation; **Vanilla PPO**, which uses an MLP network with symbolic state embeddings; and **GPT-3.5** in a zero-shot setting with task instructions, observations, and action candidates. An enhanced version, **GPT-3.5 with CoT prompt**, includes step-by-step reasoning examples from human interactions. **ReAct(GPT-4)** employs GPT-4-turbo for generating actions after reasoning with historical data. **Bilevel-LLM** combines a prompt policy, CoT process, and action policy to improve decision-making in complex tasks. The Flan-T5 Small model is used in five environments for GFlan and Bilevel-LLM, except in ALFWorld, where a fine-tuned Flan-T5 Large, based on expert trajectories

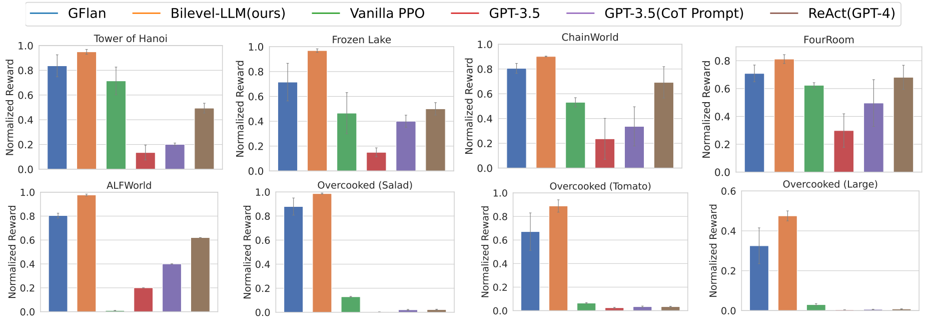


Fig. 3. Results of comparison with baselines. We plot the mean and standard error of the normalized cumulative reward. For inference baselines, the normalized reward is averaged over 20 episodes. For trainable baselines, we plot the normalized rewards averaged over the final third of the training processes across 5 random seeds. All cumulative rewards are normalized within the range of $[0, 1]$.

Table 1. Comparison of query costs on the Tower of Hanoi. We present the query cost for training Bilevel-LLM and running 20 episodes for ReAct. Bilevel-LLM incurs less cost while achieving better performance than ReAct (GPT-4).

Baselines	Performance	Tokens
Bilevel-LLM(GPT-3.5)	**0.95**	56K
ReAct(GPT-4)	0.49	334K

from 10 tasks, is employed due to the task's complexity. Further details on the baselines are available in Appendix C.

Comparison with baselines. The results of comparisons with baselines are shown in Figure3. Bilevel-LLM outperforms other baselines on all environments and exhibits a smaller standard error than the suboptimal GFlan. This indicates that Bilevel-LLM, incorporating state-adaptive language prior knowledge, can improve the task-solving ability and convergence rate. In addition, GFlan consistently surpasses Vanilla PPO, especially on Overcooked. This suggests that using a pre-trained LLM as an action policy is beneficial for decision-making due to its rich prior knowledge and strong ability to reason about world rules. The **training curves** can be found in Figure4, where our algorithm outperforms all

Table 2. Comparison of training resource requirements of baselines necessary to reach convergence (achieving a 95% win rate) on Frozen Lake. Our method requires fewer episode examples and comparable computational resources to achieve convergence.

Baselines	Time(min)	GPU	Episodes
GFlan	10.8	3598 MB	1600
Bilevel-LLM(ours)	8.7	4658 MB	**800**
Direct-Prompt-LLM	8.3	4378 MB	1300

other baselines and converges smoothly in most tasks.
Furthermore, inference baselines, including GPT-3.5, GPT-3.5 (CoT Prompt), and ReAct (GPT-4), struggle with most decision-making tasks. This may be because, although powerful models like GPT-3.5/4 can generate useful high-level task solutions, they still face challenges in long-term decision-making due to complex world models and rules. For example, in the Tower of Hanoi, GPT-3.5 can identify valid moves but struggles to generate the correct move sequence from start to goal, as detailed in Appendix D. Moreover, as shown in Table 1, Bilevel-LLM incurs a tolerable cost for querying LLMs compared to inference baselines.

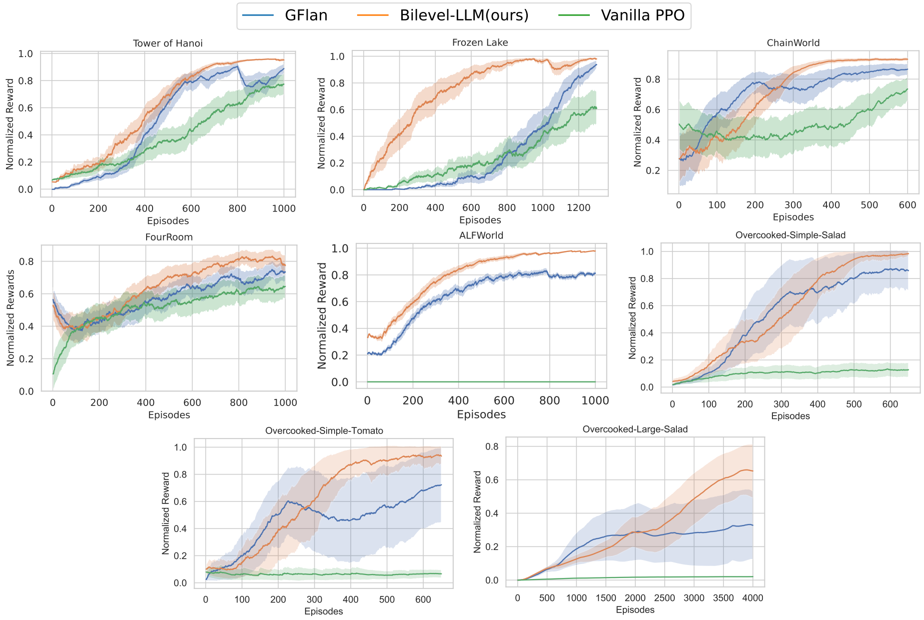
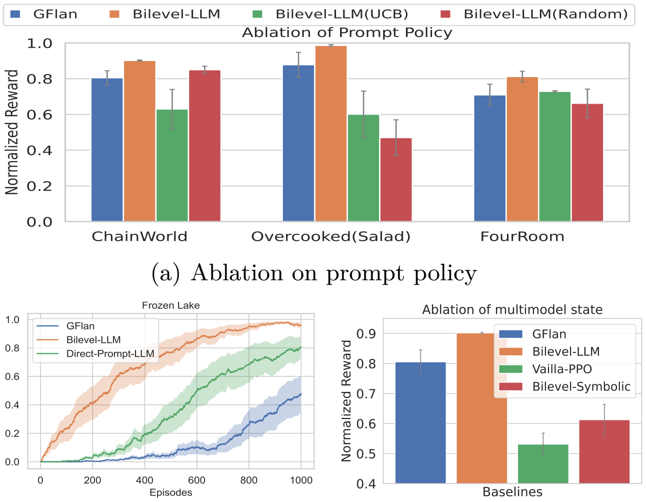


Fig. 4. Training curves of baselines. We plot the average and standard error of normalized rewards over 5 seeds.

4.3 Ablation Studies

We conducted a series of ablation studies to confirm the usefulness of the components of Bilevel-LLM . In the following, we modified components of Bilevel-LLM in order to validate the following claims:
Does the prompt policy trained through reinforcement learning improve performance? To validate the claim that the prompts generated by Bilevel-LLM lead to improved performance, we have also tried different ways to implement the prompt policy other than RL and present the comparative



(a) Ablation on prompt policy

(b) Directly prompt the CoT LLM

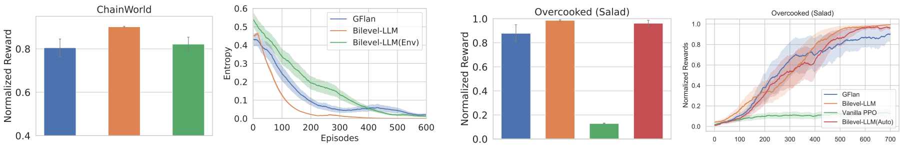
(c) Ablation on multimodal

Fig. 5. Ablation studies. We include GFlan for reference purposes. (a) The effect of different prompt generation strategies. (b) Direct-Prompt-LLM lets the LLM(GPT-3.5) directly generate state-specific prompts and corresponding thought answers. (c) Verification of the effectiveness of Bilevel-LLM under multimodal state representations on ChainWorld.

results in Figure5(a). Bilevel-LLM (Random) naively selects prompt candidates randomly, Bilevel-LLM (UCB) views the prompt selection from a candidate set as the multi-armed bandit problem and the selection follows Upper Confidence Bound (UCB) over action choices. As shown in Figure5(a), Bilevel-LLM outperforms all other prompt policy versions on all environments. The poor performance of Bilevel-LLM (UCB) might be attributed to the lack of consideration for environmental observations.

Does action behavior-guided prompt tuning improve performance? We compare our method to the variant *Direct-Prompt-LLM*, which inquires the GPT-3.5 for the prompt question and corresponding CoT on the current state, and learn an action interpreter to decode CoT output. This method involves more automation but compromises performance at the same time, as shown in Figure5(b). *Direct-Prompt-LLM* surpasses GFlan due to the injection of domain prior knowledge but performs worse than our Bilevel-LLM. This is likely because our approach adjusts the prompt question according to the action policy's behavior, reflecting the experience gained from interaction with the environment. As shown in Table 2, Our Bilevel-LLM requires less online sampled episodes to train a proficient action policy, demonstrating greater sample efficiency than the other two baselines. Additionally, our method maintains acceptable training time and GPU usage requirements, thanks to the LoRA technique.

Can the Bilevel-LLM framework accommodate multimodal state rep-



(a) Ablation on entropy

(b) Ablation on auto prompt

Fig. 6. Ablation studies. (a)Ablation of the entropy objective on Chainworld. *Left:* Normalized reward. *Right:* Entropy of the action policy. (b)Automatically generated prompt candidates on Overcooked(Salad). *Left:* Normalized reward. *Right:* Rewards during training.

resentation? We designed a baseline, *Bilevel-LLM-Symbolic*, where the action policy is replaced by that of Vanilla PPO, and it takes both the sentence embedding of the CoT output and symbolic environment observation as inputs. As shown in Figure 5(c), Bilevel-LLM outperforms GFlan, and Bilevel-LLM-Symbolic surpasses Vanilla-PPO. This indicates that the use of CoT reasoning, triggered by dynamically tuned prompt questions, enhances the performance of action policies.

Does the entropy objective improve performance? To validate that the entropy objective leads to more certainty in action policy decisions, we tested Bilevel-LLM against the variant *Bilevel-LLM (Env)*, which replaces the negative entropy with the environment reward. As shown in Figure6(a), Bilevel-LLM outperforms *Bilevel-LLM (Env)* and exhibits lower entropy of the action policy.

Can Bilevel-LLM learn from automatically generated prompts? For most environments, we utilise the prompt candidates set generated by GPT-3.5 with the task description given, demonstrating that the learning framework can operate with minimum human intervention. In the tasks of ALFWorld, we directly adapt the candidates provided automatically by the LLF-Bench environment. In experiments on Overcooked and FourRoom tasks, we use the environmental well-structured subtasks as prompt questions for simplicity purposes. To further verify the effectiveness of automatically generated prompt candidates, we compare Bilevel-LLM to the variant *Bilevel-LLM-Auto*, which uses the prompt candidates automatically generated by GPT-3.5. As shown in Figure6(b), *Bilevel-LLM-Auto* achieve similar rewards compared to those using human-crafted prompt candidates in the Overcooked task. These results also demonstrate that our plug-and-play framework can accommodate various sources of domain-specific prior knowledge, including that from pre-trained LLMs and human experts. Examples of automatically generated prompts can be found in Appendix C.

5 Related Work

LLMs for RL. A series of studies have attempted to incorporate LLMs into planning algorithms to address decision-making tasks. ICPI [3] solves a number of simple interactive RL tasks through in-context learning from historical interactions, thereby the need for expert demonstrations or gradient computations, The study [5] leverages historical trajectories to prompt LLM to generate the next step actions on the TextWorld game. LFG [23] utilises an LLM with a polling strategy to recommend and subsequently rank subgoals. Recent studies, Reflect-RL [36] and Retrosformer [33], learn a smaller model to distill valuable and critical human prior knowledge using an offline CoTs dataset collected from powerful LLMs such as GPT-4. Then use the distilled model for the subsequent action policy to take actions. In our work, we integrate complex CoT reasoning into RL to enhance the quality of actions while eliminating the need for meticulous engineering to interpret LLM outputs.

Entropy in RL. Entropy has been used extensively in RL as a tool for regularisation [18, 1]. The policy in actor-critic methods is often trained with an additional term that aims to maximise the entropy of the learned actions, with the goal of exploring the environment without having a policy collapse early to suboptimal actions [18]. More formal use of entropy is explored in maximum entropy reinforcement learning [11, 9], where the optimisation objective aims to learn the optimal policy that has the maximum entropy. In this work, we take a different approach, and look at finding prompts that minimise the entropy of the action policy. Intuitively, this would push the CoT process to provide reasoning that makes the policy sure about its action. Such minimisation of the entropy has also been explored: HIDIO [35] formulates a hierarchical approach to intrinsic options, where entropy is minimised to improve the option sub-trajectories, and the work [2] considers entropy for decision making in the exploration-exploitation trade-off.

Automated Prompt Engineering. The quality of prompts plays a crucial role in determining the output quality of LLMs. Many works hand-craft desirable prompts such as the Generative Agents [20] and ProAgent [34]. Apart from completely using human-crafted prompts, there are other studies that adopt different degrees of automation when generating meaningful prompts. For example, APE [37] and DLN [26] generate prompts from multiple examples and utilise LLM to rank the prompt candidates. PromptPG [17] trained a prompt selection network using the policy gradient to choose from a predefined set of examples. Unlike PromptPG, which selects prompts for one-step supervised data, we introduce a method to select state-adaptive prompts for multi-step decision-making tasks. Bilevel-LLM optimises the prompt policy by adapting it to the action policys performance within interactive environments.

6 Conclusion

We introduce Bilevel-LLM, a bilevel framework that is capable of learning appropriate questions (in the form of prompts), and then performing complex reasoning for guiding actions executed by an action policy. The bilevel nature of the framework enables the accommodation of separate objectives for the two learning components, namely the prompt policy uses an action policy entropy minimisation objective which enables it to induce unambiguous and useful prompts to be fed to the action policy. Meanwhile, the action policy learns how to perform actions in the environment while making use of the CoT thoughts which it learns to interpret. We showed that this leads to a powerful framework that outperforms leading baselines in complex benchmark environments. We believe our framework takes an important step towards generalist artificial intelligence that is capable of introspection and complex decision-making.

Limitations and Future Work In this work, we only explore our framework for solving single-agent decision-making tasks, but neglect the prevalent multi-agent setting. Further work will extend our framework to encompass decision-making in multi-agent scenarios, possibly exploring the potential of leveraging the reasoning abilities of LLMs to uncover cooperation patterns or to model the behaviors of opponents.

Acknowledgments. Haifeng Zhang thanks IPT Project 2024002.

References

1. Albrecht, S.V., Christianos, F., Schäfer, L.: Multi-Agent Reinforcement Learning: Foundations and Modern Approaches. MIT Press (2023), `https://www.marl-book.com`
2. Allahverdyan, A.E., Galstyan, A., Abbas, A.E., Struzik, Z.R.: Adaptive decision making via entropy minimization. International Journal of Approximate Reasoning **103**, 270–287 (2018)
3. Brooks, E., Walls, L., Lewis, R.L., Singh, S.: In-context policy iteration. arXiv preprint arXiv:2210.03821 (2022)
4. Carta, T., Romac, C., Wolf, T., Lamprier, S., Sigaud, O., Oudeyer, P.Y.: Grounding large language models in interactive environments with online reinforcement learning. arXiv preprint arXiv:2302.02662 (2023)
5. Chen, L., Wang, L., Dong, H., Du, Y., Yan, J., Yang, F., Li, S., Zhao, P., Qin, S., Rajmohan, S., et al.: Introspective tips: Large language model for in-context decision making. arXiv preprint arXiv:2305.11598 (2023)
6. Cheng, C.A., Kolobov, A., Misra, D., Nie, A., Swaminathan, A.: Llf-bench: Benchmark for interactive learning from language feedback. arXiv preprint arXiv:2312.06853 (2023)
7. Colson, B., Marcotte, P., Savard, G.: An overview of bilevel optimization. Annals of operations research **153**, 235–256 (2007)
8. Côté, M.A., Kádár, A., Yuan, X., Kybartas, B., Barnes, T., Fine, E., Moore, J., Hausknecht, M., El Asri, L., Adada, M., et al.: Textworld: A learning environment for text-based games. In: Computer Games: 7th Workshop, CGW 2018, Held in Conjunction with the 27th International Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, July 13, 2018, Revised Selected Papers 7. pp. 41–75. Springer (2019)
9. Eysenbach, B., Levine, S.: Maximum entropy rl (provably) solves some robust rl problems. arXiv preprint arXiv:2103.06257 (2021)
10. Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., Callan, J., Neubig, G.: Pal: Program-aided language models. In: International Conference on Machine Learning. pp. 10764–10799. PMLR (2023)
11. Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al.: Soft actor-critic algorithms and applications. arXiv preprint arXiv:1812.05905 (2018)

12. Hao, S., Gu, Y., Ma, H., Hong, J.J., Wang, Z., Wang, D.Z., Hu, Z.: Reasoning with language model is planning with world model. arXiv preprint arXiv:2305.14992 (2023)
13. Hinz, A.M., Klavžar, S., Milutinović, U., Petr, C.: The tower of Hanoi-Myths and maths. Springer (2013)
14. Hu, E.J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: International Conference on Learning Representations
15. Jang, Y., Lee, J., Kim, K.E.: Gpt-critic: Offline reinforcement learning for end-to-end task-oriented dialogue systems. In: International Conference on Learning Representations (2021)
16. Lin, B.Y., Fu, Y., Yang, K., Brahman, F., Huang, S., Bhagavatula, C., Ammanabrolu, P., Choi, Y., Ren, X.: Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. Advances in Neural Information Processing Systems **36** (2024)
17. Lu, P., Qiu, L., Chang, K.W., Wu, Y.N., Zhu, S.C., Rajpurohit, T., Clark, P., Kalyan, A.: Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. arXiv preprint arXiv:2209.14610 (2022)
18. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T.P., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. International Conference on Machine Learning (2016)
19. Pakseresht, M., Mahdavi, I., Shirazi, B., Mahdavi-Amiri, N.: Co-reconfiguration of product family and supply chain using leader–follower stackelberg game theory: Bi-level multi-objective optimization. Applied Soft Computing **91**, 106203 (2020)
20. Park, J.S., O'Brien, J.C., Cai, C.J., Morris, M.R., Liang, P., Bernstein, M.S.: Generative agents: Interactive simulacra of human behavior. arXiv preprint arXiv:2304.03442 (2023)
21. Rae, J.W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., et al.: Scaling language models: Methods, analysis & insights from training gopher. arXiv preprint arXiv:2112.11446 (2021)
22. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
23. Shah, D., Equi, M.R., Osiński, B., Xia, F., Levine, S., et al.: Navigation with large language models: Semantic guesswork as a heuristic for planning. In: 7th Annual Conference on Robot Learning (2023)
24. Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S.: Reflexion: Language agents with verbal reinforcement learning. Advances in Neural Information Processing Systems **36** (2024)
25. Shridhar, M., Yuan, X., Côté, M.A., Bisk, Y., Trischler, A., Hausknecht, M.: Alfworld: Aligning text and embodied environments for interactive learning. arXiv preprint arXiv:2010.03768 (2020)
26. Sordoni, A., Yuan, X., Côté, M.A., Pereira, M., Trischler, A., Xiao, Z., Hosseini, A., Niedtner, F., Roux, N.L.: Deep language networks: Joint prompt training of stacked llms using variational inference. arXiv preprint arXiv:2306.12509 (2023)
27. Tan, W., Zhang, W., Liu, S., Zheng, L., Wang, X., Bo, A.: True knowledge comes from practice: Aligning large language models with embodied environments via reinforcement learning. In: The Twelfth International Conference on Learning Representations (2024)
28. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., Zhou, D.: Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171 (2022)

29. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., Zhou, D.: Chain of thought prompting elicits reasoning in large language models. arXiv preprint arXiv:2201.11903 (2022)
30. Yan, X., Guo, J., Lou, X., Wang, J., Zhang, H., Du, Y.: An efficient end-to-end training approach for zero-shot human-ai coordination. Advances in Neural Information Processing Systems **36** (2024)
31. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y., Narasimhan, K.: Tree of thoughts: Deliberate problem solving with large language models. arXiv preprint arXiv:2305.10601 (2023)
32. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: React: Synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629 (2022)
33. Yao, W., Heinecke, S., Niebles, J.C., Liu, Z., Feng, Y., Xue, L., Murthy, R., Chen, Z., Zhang, J., Arpit, D., et al.: Retroformer: Retrospective large language agents with policy gradient optimization. arXiv preprint arXiv:2308.02151 (2023)
34. Zhang, C., Yang, K., Hu, S., Wang, Z., Li, G., Sun, Y., Zhang, C., Zhang, Z., Liu, A., Zhu, S.C., et al.: Proagent: Building proactive cooperative ai with large language models. arXiv preprint arXiv:2308.11339 (2023)
35. Zhang, J., Yu, H., Xu, W.: Hierarchical reinforcement learning by discovering intrinsic options. arXiv preprint arXiv:2101.06521 (2021)
36. Zhou, R., Du, S.S., Li, B.: Reflect-rl: Two-player online rl fine-tuning for lms. arXiv preprint arXiv:2402.12621 (2024)
37. Zhou, Y., Muresanu, A.I., Han, Z., Paster, K., Pitis, S., Chan, H., Ba, J.: Large language models are human-level prompt engineers. arXiv preprint arXiv:2211.01910 (2022)