Advancing Autonomous VLM Agents via Variational Subgoal-Conditioned Reinforcement Learning

Qingyuan Wu *
University of Liverpool
University of Southampton

Jianheng Liu *
Huawei Noah's Ark Lab

Jianye Hao
Tianjin University

Jun Wang
University College London

Kun Shao†
Huawei Noah's Ark Lab

Abstract

State-of-the-art (SOTA) reinforcement learning (RL) methods have enabled vision-language model (VLM) agents to learn from interaction with online environments without human supervision. However, these methods often struggle with learning inefficiencies when applied to complex, real-world decision-making tasks with sparse rewards and long-horizon dependencies. We propose a novel framework, Variational Subgoal-Conditioned Reinforcement Learning (VSC-RL), advancing the VLM agents in resolving challenging decision-making tasks. Fundamentally distinct from existing methods, VSC-RL reformulates the decision-making problem as a variational subgoal-conditioned RL problem with the newly derived optimization objective, Subgoal Evidence Lower BOund (SGC-ELBO), which comprises two key components: (a) maximizing the subgoal-conditioned return, and (b) minimizing the divergence from a reference goal-conditioned policy. We theoretically and empirically demonstrate that the VSC-RL can efficiently improve the learning efficiency without compromising performance guarantees. Across a diverse set of challenging benchmarks, including mobile device and web control tasks, VSC-RL consistently outperforms existing SOTA methods, achieving superior learning efficiency and performance.

1 Introduction

Recently, the large language models (LLMs) and vision-language models (VLMs) have demonstrated remarkable capabilities in content understanding and commonsense reasoning [41, 33, 12], achieving notable success in various real-world applications, such as visual question answering and visual captioning [5, 9]. These advancements highlight the strong potential of VLMs to tackle complex real-world decision-making problems (e.g., mobile device [37] and web control [49] tasks) via building intelligent VLM agents through the advanced VLMs [44, 48]. Meanwhile, after achieving impressive results in board games [34] and video games [7], reinforcement learning (RL) methods have been applied in online training VLM agents for tackling sequential decision-making tasks [4, 29].

Overall, based on the specific training paradigm, VLM agents can be categorized into three main types: prompting-based, imitation-based, and RL-based agents. Directly leveraging VLMs (e.g., Gemini-1.5-Pro [36] and GPT-4V [24]) to capture the critical information from the multimodal content, prompting-based agents aim to generate action via prompting engineering and retrieving techniques [44, 42]. The performance of the prompting-based agents is usually limited, as the weights

*Equal Contribution
†Correspondence to: Kun Shao, shaokun2@huawei.com

1st Workshop on VLM4RWD at NeurIPS 2025.

of these VLMs can not be updated. To address this limitation, some studies [46, 13] employ imitation learning techniques to fine-tune the open-source VLMs using human demonstrations. However, the performance of imitation-based agents is highly dependent on the quality and diversity of the demonstrations. Consequently, imitation-based agents may struggle with generalization and often underperform on out-of-distribution and unseen tasks. Recently, RL-based agents have emerged as a promising solution. By incorporating RL techniques, these agents enable VLMs to tackle complex sequential decision-making problems [4, 29]. Nevertheless, existing RL-based agents often suffer from the learning efficiency issue in addressing challenging control tasks with sparse reward signals and complicated goals. In many real-world scenarios, tasks require executing long sequences of actions, with rewards only provided upon successful completion. This delayed feedback poses a significant challenge for learning, fundamentally impacting the efficiency of RL-based agents. Some existing works attempt to address this issue by introducing implicit curriculum [3] or hand-crafted subgoals [11, 8]. However, these existing approaches often fail to learn a proper policy in the real-world complex sequential decision-making task due to the complicated subgoal generation and curriculum design methodologies.

To address the fundamental limitations of RL-based agents mentioned above, we introduce Variational Subgoal-Conditioned RL (VSC-RL), a novel RL-based VLM agent method for enhancing learning efficiency in real-world complex sequential decision-making tasks. Based on the perspective of variational inference, VSC-RL reformulates the decision-making task as the variational subgoal-conditioned RL problem, which is later efficiently solved by utilising extensive optimization techniques. Additionally, VSC-RL utilises the significant reasoning and planning capabilities of VLM to autonomously decompose the complex goal into feasible subgoals. Given the generated subgoals, VSC-RL optimizes the objective of SubGoal-Conditioned Evidence Lower BOund (SGC-ELBO), thus effectively improving learning efficiency, consisting of (a) maximizing the subgoal-conditioned return of the target agent and (b) minimizing the subgoal-conditioned difference with the reference agent. We theoretically derive the new objective of SGC-ELBO from the original optimization objective, ensuring both improved efficiency and performance guarantees. Empirical results on various benchmarks validate our statement that VSC-RL significantly outperforms SOTA VLM agents in both sample efficiency and final performance.

In this paper, literature related to VLM agents and RL methods is discussed in Section 2. We introduce notations related to goal-conditioned RL, variational RL and subgoal generator in Section 3. In Section 4, we illustrate how to formulate the sequential decision-making problem as a variational subgoal-conditioned RL problem and derive the new optimization objective: SGC-ELBO, followed by the practical implementation of VSC-RL. In Section 5, the experimental results exhibit that our VSC-RL agent can achieve superior performance compared to existing SOTAs. Overall, the main contributions of this paper are summarised as follows:

- We propose VSC-RL, a novel variational subgoal-conditioned RL method for enhancing VLM agents in resolving real-world sequential decision-making problems.
- We theoretically show that SGC-ELBO, the optimization objective of the VSC-RL, can effectively improve learning efficiency while maintaining the performance guarantee.
- We experimentally show that VSC-RL significantly outperforms various SOTAs in both learning efficiency and final performance on various challenging benchmarks.

2 Related Works

2.1 VLM Agents for Decision-making

In real-world complex control tasks requiring capacities in reasoning, planning and content understanding, it is necessary to enable agents with the vision-language models (VLMs). In particular, the VLMs can process and abstract the image and language content for challenging decision-making tasks, especially in mobile device and web control tasks [37, 18]. Existing VLM agents can be categorized as prompting-based, imitation-based, and RL-based agents based on the corresponding learning paradigms. Additionally, some recent works explore using VLM to enhance agents' abilities.

Prompting-based Agent. Leveraging the inherent reasoning and planning abilities of prosperity VLMs (e.g., Gemini-1.5-Pro [36] and GPT-4V [24]), the prompting-based agent makes decision via

prompting engineering and retrieving techniques. For instance, AppAgent [44] first introduces a unified prompting-based agent method to enable the vision-language model to directly interact with mobile applications by providing the prompts with details of actions. Set-of-Marks [42] proposes a new prompting method to enhance the visual grounding ability of VLM. However, the performance of these prompting-based agents is always sensitive to the prompts required to be manually and carefully designed. Therefore, it is challenging for the prompting-based agent to directly output the correct and desired actions to address real-world complex control problems.

Imitation-based Agent. The imitation-based agent learns to mimic the expert behaviours by fine-tuning the policy on human demonstration. Recently, Android in the Wild (AitW) [30] collected large-scale datasets of mobile device control tasks, enabling agents to directly learn from human experience. AutoUI [46] and CogAgent [13] fine-tune the VLM-based policies with the AitW dataset, remarkably outperforming the prompting-based agent. In order to adapt the fine-tuned agent to the online environment, Filtered BC [26] introduces online imitation mechanisms to learn from successful online experiences. Unfortunately, these methods rely heavily on high-quality human demonstrations and often struggle to generalize to unseen tasks, limiting their application in diverse real-world scenarios.

RL-based Agent. Different to prompting-based and imitation-based agents, the RL-based agent can autonomously optimize the policy through trial-and-error interactions with environments, without human supervision. DigiRL [4] introduces a unified offline-to-online RL framework that enables agents to learn directly from real-time interactions in dynamic environments, improving performance without the need for curated datasets. DistRL [38] builds an asynchronous distributed RL system, allowing training multiple agents in parallel across different environments, thus significantly enhancing scalability and convergence speed. WebRL [29] introduces a self-evolving online curriculum RL framework, enabling effective training of web agents through adaptive task generation in web control tasks. However, these RL-based agents still fundamentally suffer from the learning efficiency issue in challenging sequential decision-making tasks with sparse rewards and long horizons.

Enhancing RL with VLM. Recent works have shown that VLM can enhance the RL method via its remarkable capacities of reasoning, planning, and content understanding. Recent works suggest adopting VLM in reward-shaping for RL. For instance, VLM-RMs [31] demonstrate that VLMs can serve as effective reward models for learning complex skills. VLM can also generate the subgoals to guide the learning process for autonomous driving [25] and robot [43] tasks. Nonetheless, it is still an open problem how to effectively integrate the VLM-generated subgoals into RL.

To mitigate the above issues, we present VSC-RL, which can autonomously decompose the goal into feasible subgoals by advanced VLM, and then efficiently resolve each subgoal from the principle of variational inference.

2.2 Goal-conditioned and Variational RL

Goal-conditioned RL. Sequential decision-making tasks can be viewed as the goal-conditioned RL problem [17]. Based on the current state, the agent aims to find the optimal policy that guides progress toward the given goal for maximizing the return. Hindsight experience replay [3] introduces an implicit curriculum learning method to enhance learning efficiency and robustness. With the perspective of divide-and-conquer, some approaches suggest guiding the agent with subgoals as intermediate reward signals via imagination [8, 22] and tree-search [15, 27].

Variational RL. The RL problem can be viewed as the variational inference problem [16], which can be resolved by utilising extensive optimization tools, thus effectively improving the learning efficiency. Applying the expectation-maximization algorithm in the actor-critic method in RL, VIP [23] presents a unified variational inference framework. MPO [1, 2] proposes a series of off-policy RL with entropy regulation in the manner of expectation-maximization. VDPO [39] and CVPO [19] apply the variational inference techniques in addressing the RL problem with delayed signals and safety constraints, respectively.

This paper aims to show how to formulate the control problem as a variational subgoal-conditioned RL problem from the perspective of variational inference, which allows us to resolve the complicated control task by utilising extensive optimization tools.

3 Preliminaries

Finite-Horizon Goal-Conditioned MDP. We formulate the RL problem as the finite horizon goal-conditioned Markov Decision Process (MDP), denoted by the tuple $< \mathcal{G}, \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, H >$ where $\mathcal{G}$ is the goal set, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the dynamic function, $\mathcal{R}$ is the reward function and H is the horizon. At each timestep t, the agent takes action $a_t \in \mathcal{A}$ (e.g., typing text, press button or slide the screen) based on its policy $\pi : \mathcal{S} \times \mathcal{G} \times \mathcal{A} \to [0, 1]$, the current screenshot $s_t \in \mathcal{S}$, and a specific goal $g \in \mathcal{G}$ (e.g., search a new TV at Best Buy) selected in the beginning of each episode. The agent only receives the reward $r_t = 1$ if the goal g is accomplished, otherwise the reward $r_t = 0$. The objective of the agent is to find the policy π which can accomplish all goals from the goal set $\mathcal{G}$ within the finite horizon H.

Variational RL. RL can be viewed as a variational inference problem. We denote the optimality of a trajectory τ is the event O, and the corresponding probability of the trajectory optimality is denoted as $p(O|\tau) \propto \exp\left(\frac{\mathcal{J}(\tau)}{\alpha}\right)$ where α is the temperature. Therefore, the objective transforms to finding a policy π with the highest log evidence: $\max_\pi \log p_\pi(O)$. Furthermore, the Evidence Lower BOund of the objective is:

$$\mathop{\mathbb{E}}_{\tau \sim q(\tau)} [\log p(O|\tau)] - \text{KL}(q(\tau)||p_\tau(\tau)), \tag{1}$$

where $q(\tau)$ is the prior trajectory distribution and KL is the Kullback-Leibler divergence. Thus, the objective of Variational RL is maximizing the ELBO (Equation (1)).

Subgoal Generator. For challenging control tasks with sparse and long-term reward signals, it is difficult to learn a useful policy that arrives at the final goal within a finite horizon. Therefore, subgoal generation is particularly useful in providing the intermediate signals to facilitate learning. Then, we introduce the assumption of the existence of subgoals for the given goal, aiming to bring the goal-conditioned RL problem to the subgoal-conditioned RL problem as follows.

Assumption 3.1 (Existence of Subgoals)**.** Given a trajectory τ and the corresponding goal g, it always exists a sequence of sub-trajectories and corresponding subgoals $\{\tau_i, sg_i\}_{i=1}^N (1 \leq N \leq H)$ induced from the τ and g.

Commonly adopted in literature [35], the above assumption is mild and usually holds. For instance, when $N = 1$, the subgoals and sub-trajectories are the original goal and trajectory, respectively. When $N = H$, each sub-trajectory is composed of one single transition-tuple (s_t, a_t, r_t, s_{t+1}) with its corresponding subgoal.

4 Our Approach: VSC-RL

In this section, we present our approach, Variational Subgoal-Conditioned Reinforcement Learning (VSC-RL) for enhancing VLM agents in solving real-world decision-making tasks. First, we formulate the sequential decision-making task as the variational goal-conditioned RL problem (Section 4.1). Next, we derive the new subgoal-conditioned optimization objective, SGC-ELBO, consisting of (a) maximizing the subgoal-conditioned return (Proposition 4.1) and (b) minimizing the subgoal-conditioned difference (Proposition 4.2). We also theoretically show the derivation of new optimization objective, ensuring both improved learning efficiency and performance guarantees. In Section 4.3, we demonstrate that VLMs can effectively generate feasible subgoals from the complex goal for VSC-RL. The practical implementation is illustrated in Section 4.4. We present the overall pipeline of VSC-RL in Figure 1, and the pseudo-code of VSC-RL is summarised in Algorithm 1.

4.1 Problem Formulation

We first formulate the sequential decision-making as the variational goal-conditioned RL problem. In this context, similar to Equation (1), the objective is to find a goal-conditioned policy π with the highest log evidence: $\max_\pi \log p_\pi(O|g)$ for a given goal g. Then, we have the Goal-Conditioned ELBO (GC-ELBO) of $\log p_\pi(O|\tau, g)$ as follows:

$$\text{GC-ELBO}(\pi, \pi_{\text{ref}}, g) = \mathop{\mathbb{E}}_{\tau \sim p_\pi(\tau|g)} [\log p(O|\tau, g)] - \text{KL}(p_\pi(\tau|g)||p_{\pi_{\text{ref}}}(\tau|g)), \tag{2}$$

where $p_{\pi_{\text{ref}}}(\tau|g)$ is the prior trajectory distribution of the goal-conditioned reference policy π_{ref} for the given goal g. Therefore, from Equation (2), the objective becomes maximizing the GC-ELBO: $\max_\pi \text{GC-ELBO}(\pi, \pi_{\text{ref}}, g)$.

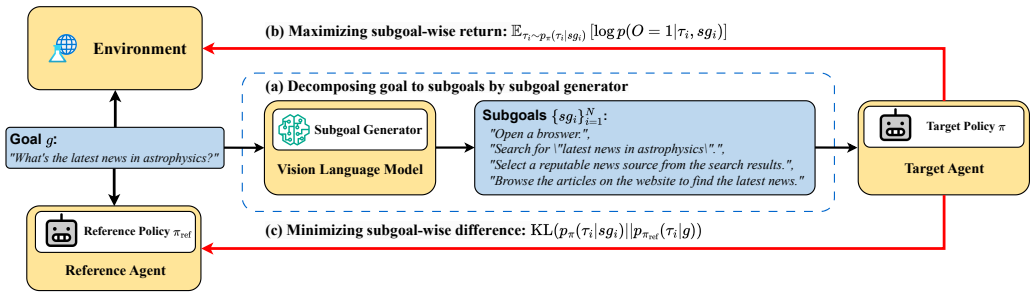


Figure 1: The pipeline of VSC-RL. (a) VLM autonomously decomposes the goal g to the subgoals $\{sg_i\}_{i=1}^N$. VSC-RL optimizes the objective of SGC-ELBO consisting of (b) maximizing the subgoal-conditioned return and (c) minimizing the subgoal-conditioned difference.

Algorithm 1 VSC-RL

Input: goal g, subgoal generator VLM, reference policy π_{ref}, target policy π and value function V;

for Epoch = $1, \cdots$ **do**
  Generate Subgoals $\{sg_i\}_{i=1}^N \sim \text{VLM}(g)$
  Collect $(\tau_i, sg_i)_{i=1}^N$ from π for the given goal g
  # Optimize the SGC-ELBO (Equation (4))
  Maximize Subgoal-conditioned Return via Equation (5) and Equation (6)
  Minimize Subgoal-conditioned Behavior Difference via Equation (7)
end for
Output: updated policy π

4.2 Variational Subgoal-Conditioned RL

With the assumption of the subgoals (Assumption 3.1), we demonstrate that the former term of GC-ELBO (Equation (2)) is equivalent to the maximizing subgoal-conditioned RL objective (Proposition 4.1) and the latter term of GC-ELBO can be transformed to the minimizing subgoal-conditioned difference (Proposition 4.2).

Based on Equation (2), we show that the former term, $\mathbb{E}_{\tau \sim p_\pi(\tau|g)}[\log p(O|\tau, g)]$, can be reformulated in the subgoal-conditioned RL objective with shorter-horizon in the following Proposition 4.1.

Proposition 4.1 (Subgoal-Conditioned Optimization Objective, Proof in Proposition C.1)**.** *Given a goal g with corresponding subgoals $\{sg_i\}_{i=1}^N$ and a subgoal-conditioned target policy π, the objective of*

$$\max_\pi \mathop{\mathbb{E}}_{\tau \sim p_\pi(\tau|g)}[\log p(O|\tau, g)]$$

is equivalent to the objective of

$$\max_\pi \sum_{i=1}^N \left[\mathop{\mathbb{E}}_{\tau_i \sim p_\pi(\tau_i|sg_i)}[\log p(O|\tau_i, sg_i)] \right].$$

In the above proposition, the goal-wise objective has been transformed into the subgoal-conditioned objective, which is composed of N subgoals with corresponding shorter horizons. Thus, the agent can learn from these reward signals from the subgoals, thus effectively improving the learning efficiency [14].

Next, we show that the latter term in Equation (2), $\text{KL}(p_\pi(\tau|g)||p_{\pi_{\text{ref}}}(\tau|g))$, has the subgoal-conditioned upper bound in the following proposition.

Proposition 4.2 (Subgoal-conditioned Difference Bound, Proof in Proposition C.2)**.** *Given goal-conditioned reference policy π_{ref} and subgoal-conditioned target policy π, the goal-conditioned KL divergence of a given goal g has the upper bound of subgoal-conditioned KL divergence of corresponding subgoals $\{sg_i\}_{i=1}^N$ as follows:*

$$KL(p_\pi(\tau|g)||p_{\pi_{ref}}(\tau|g)) \leq \sum_{i=1}^{N} \left[KL(p_\pi(\tau_i|sg_i)||p_{\pi_{ref}}(\tau_i|g))\right].$$

Therefore, from Proposition 4.2, we can directly minimize the N subgoal-conditioned KL divergences, which is the upper bound of the goal-conditioned KL divergence.

Based on Proposition 4.1 and Proposition 4.2, the newly-derived optimization objective of SubGoal-Conditioned ELBO (SGC-ELBO) is as follows:

$$\text{SGC-ELBO}(\pi, \pi_{\text{ref}}, sg_i, g) = \mathop{\mathbb{E}}_{\tau_i \sim p_\pi(\tau_i|sg_i)} [\log p(O|\tau_i, sg_i)] - \text{KL}(p_\pi(\tau_i|sg_i)||p_{\pi_{\text{ref}}}(\tau_i|g)). \quad (3)$$

Equation (3) consists of two key components: (a) maximizing the subgoal-conditioned return of the target policy π and (b) minimizing the subgoal-conditioned difference between π and the reference policy π_{ref}. Therefore, the agent can directly learn to resolve the subgoal sg_i with a shorter horizon requirement, effectively improving the learning efficiency. The newly derived optimization objective of SGC-ELBO (Equation (3)) can improve learning efficiency without compromising performance guarantees.

4.3 Autonomous Subgoal Generation via Vision-Language Models

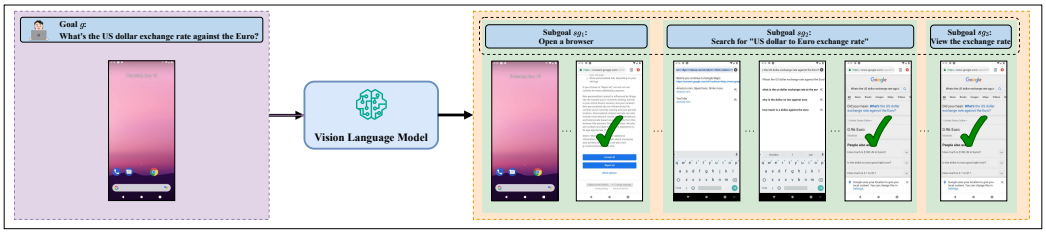


Figure 2: Autonomous subgoal generation in AitW task. The VLM autonomously decomposes the goal of the complicated mobile device control task into easily achievable subgoals.

For real-world complex decision-making, it is challenging to handcraft and design the subgoals for each goal manually. VLM has exhibited a unique reasoning ability in image captioning, visual question answering, and multimodal reasoning via integrating and interpreting visual and textual information to derive meaningful insights for VLM agents. Therefore, we use VLM as the subgoal generator, which autonomously decomposes the given goal g into the feasible subgoals $\{sg_i\}_{i=1}^N$. As demonstrated in the AitW task example (Figure 2), VLM can autonomously decompose the goal: *"What's the US dollar exchange rate against the Euro?"* into more specific and easily solvable subgoals, including *"Open a browser"*, *"Search for "US dollar to Euro exchange rate"*, and *"View the exchange rate"*. Therefore, we can tell that the VLM can serve as the subgoal generator in the general case. Additionally, we also provide the example of subgoal generation in Appendix K. In the context of VLM as the autonomous vision-language subgoal generator, the optimization objective (Equation (3)) can be written as

$$\max_\pi \left[\sum_{\{sg_i\}_{i=1}^N \sim \text{VLM}(g)} [\text{SGC-ELBO}(\pi, \pi_{\text{ref}}, sg_i, g)] \right], \quad (4)$$

where subgoals $\{sg_i\}_{i=1}^N$ are generated by a VLM through prompting with the original goal g.

4.4 Practical Implementation

As a unified RL-based agent framework, most existing RL-based methods can easily be embedded in VSC-RL. In this paper, we mainly consider the mobile device and web control tasks to evaluate the

VSC-RL, a representative and challenging real-world decision-making task which has drawn attention recently. Specifically, the reference agent π_{ref} and target agent π are both initialised as the AutoUI-Base agent [46], which is pre-trained on the Android in the Wild (AitW) datasets. To maximize the subgoal-conditioned RL objective in Equation (3), VSC-RL uses the Advantage-Weighted Regression (AWR) algorithm [28] modified by DigiRL [4] as follows:

$$\arg\max_{\pi} \mathop{\mathbb{E}}_{s,a,sg_i\sim\mathcal{D}} \left[\log \pi(a|s, sg_i) \exp\left(\frac{A(s, a, sg_i)}{\beta}\right)\right], \tag{5}$$

where $\mathcal{D}$ is the replay buffer, β is the hyperparameter and $A(s, a, sg_i) := R_i - V(s, a, sg_i)$ is the advantage function which aims to predict the return R_i of the subgoal sg_i as follows:

$$\arg\min_{V} \mathop{\mathbb{E}}_{s,a,sg_i,R_i\sim\mathcal{D}} \left[||R_i - V(s, a, sg_i)||\right], \tag{6}$$

where R_i is the binary return evaluated by the VLM [26] and $V(s, a, sg_i)$ is the subgoal-conditioned value function. VSC-RL minimizes the subgoal-conditioned KL divergence in Equation (3) via imitation loss as follows:

$$\arg\max_{\pi} \mathop{\mathbb{E}}_{\substack{a_{\text{ref}}\sim\pi_{\text{ref}}(\cdot|s,g)\\ s,sg_i,g\sim\mathcal{D}}} \left[\log \pi(a_{\text{ref}}|s, sg_i)\right], \tag{7}$$

where a_{ref} is the reference action. Similar to DigiRL [4], VSC-RL additionally learns the instruction-level value function for filtering the sub-trajectories and accelerating the learning process.

VSC-RL adopts Gemini-1.5-Pro [36] as the subgoal generator. Specifically, we in-context prompt the VLM to generate the subgoals for a given goal, including human demonstration as examples. The prompt example is provided in Appendix K and the qualitative example is provided in Appendix L. . Overall, the pseudo-code of VSC-RL is summarised in Algorithm 1.

5 Experiments

In this section, we empirically demonstrate that our VSC-RL can achieve better sample efficiency and a higher success rate than various state-of-the-art (SOTA) agents in the challenging AitW [30] benchmark. We discuss limitations and challenges in Appendix A. The implementation details and hyperparameter settings are listed in Appendix B. Additionally, we present the additional experiments on WebArena-Lite [18] in Appendix F and MiniGrid [10] in Appendix D. Additional experiments investigating the subgoal generator in VSC-RL are presented in Appendix E. We also present the ablation results for evaluating the key components of VSC-RL in Appendix G. Ablation studies on subgoal quality and subgoal generator are provided in Appendix H and Appendix I, respectively. We also analyze the failure cases of VSC-RL in Appendix J.

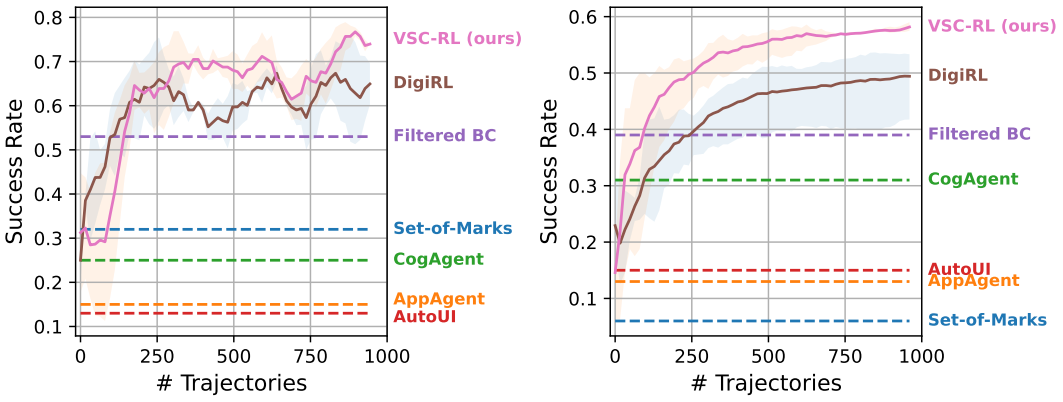
5.1 Experimental Settings

Benchmarks. For the complex and challenging problem, we mainly consider AitW General and Web Shopping tasks [30], two kinds of the most challenging device control tasks for evaluation. The horizons of General and Web Shopping tasks are set to 10 and 20 steps, respectively. The success of the task is autonomously evaluated by the Gemini-1.5-Pro [36] via the in-context prompting approach.

Baselines. We compare our VSC-RL with various SOTA baselines, including prompting-based agents (Set-of-Marks [42] and AppAgent [45]), imitation-based agents (AutoUI [46], CogAgent [13] and Filtered BC [26]) and RL-based agents (DigiRL [4]). Each method is tested on 3 independent runs, consistent with existing works [4].

5.2 Experimental Results and Analysis

The learning curves of AitW General and Web Shopping are summarised in Figure 3. Overall, our VSC-RL outperforms other baselines significantly in both the General and Web Shopping tasks. The RL-based agents (DigiRL and VSC-RL) both show leading performance in the AitW General task. After reaching a similar performance of 65.0% success rate with DigiRL in 250 trajectories, our VSC-RL outperforms all baselines significantly, arriving at the best final performance of 75% success



(a) General (b) Web Shopping

Figure 3: Learning curves on AitW (a) General and (b) Web Shopping tasks.

Table 1: The evaluated performance on the train and test datasets of the General and Web Shopping tasks. The best performance is **highlighted**.

Task		Set-of-Marks	AppAgent	CogAgent	AutoUI	Filtered BC	DigiRL	VSC-RL (ours)
General	Train	32.3%	14.6%	25.0%	12.5%	53.5%	64.9%	**73.9%**
	Test	16.7%	16.7%	25.0%	14.6%	62.5%	67.7%	**72.9%**
Web Shopping	Train	6.3%	5.2%	31.3%	14.6%	53.6%	55.3%	**64.6%**
	Test	11.5%	8.3%	38.5%	17.7%	54.2%	41.3%	**58.3%**

rate. Similarly, RL-based agents (DigiRL and VSC-RL) dominate all other types of agents remarkably in the Web Shopping task. Specifically, our VSC-RL can finally achieve around 60.0% success rate, significantly outperforming 50.0% success rate of DigiRL. We also evaluate the generalization of our VSC-RL on the test datasets, including a range of unseen tasks, respectively. The results summarised in Table 1 tell us that our VSC-RL shows significant superiority in both the train and test datasets. Especially, in the general tasks, VSC-RL performs approximately $+13.9\%$ and $+7.7\%$ better than the second-best baseline on the train and test datasets, respectively. Similarly, VSC-RL achieves the best performance on both the train and test datasets of web shopping tasks. Overall, VSC-RL can exhibit consistent performance on unseen tasks, showing remarkable generalization ability.

6 Conclusion

This work investigates advancing VLM agents in resolving real-world complex sequential decision-making tasks. Existing promising RL-based agents often suffer from the learning efficiency issue in solving tasks with complicated goals and sparse reward signals. To address this issue, we propose VSC-RL, which can autonomously decompose the goal to subgoals and resolve them efficiently. VSC-RL reformulates the decision-making task as a variational subgoal-conditioned RL problem with the new derived optimization objective of SGC-ELBO, thus effectively improving the learning efficiency without comprising the performance guarantee. In various benchmarks, especially in challenging mobile device and web control tasks, we empirically show that VSC-RL exhibits significant performance improvement and learning efficiency, remarkably outperforming existing methods.

References

[1] A. Abdolmaleki, J. T. Springenberg, J. Degrave, S. Bohez, Y. Tassa, D. Belov, N. Heess, and M. Riedmiller. Relative entropy regularized policy iteration. *arXiv preprint arXiv:1812.02256*, 2018.

[2] A. Abdolmaleki, J. T. Springenberg, Y. Tassa, R. Munos, N. Heess, and M. Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

[3] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[4] H. Bai, Y. Zhou, M. Cemri, J. Pan, A. Suhr, S. Levine, and A. Kumar. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *arXiv preprint arXiv:2406.11896*, 2024.

[5] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[6] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[7] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dkebiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[8] E. Chane-Sane, C. Schmid, and I. Laptev. Goal-conditioned reinforcement learning with imagined subgoals. In *International conference on machine learning*, pages 1430–1440. PMLR, 2021.

[9] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024.

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

[11] P. Dayan and G. E. Hinton. Feudal reinforcement learning. *Advances in neural information processing systems*, 5, 1992.

[12] S. Hong, X. Zheng, J. Chen, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou, et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.

[13] W. Hong, W. Wang, Q. Lv, J. Xu, W. Yu, J. Ji, Y. Wang, Z. Wang, Y. Dong, M. Ding, et al. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14281–14290, 2024.

[14] N. Jiang and A. Agarwal. Open problem: The dependence of sample complexity lower bounds on planning horizon. In *Conference On Learning Theory*, pages 3395–3398. PMLR, 2018.

[15] T. Jurgenson, O. Avner, E. Groshev, and A. Tamar. Sub-goal trees a framework for goal-based reinforcement learning. In *International conference on machine learning*, pages 5020–5030. PMLR, 2020.

[16] S. Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

[17] M. Liu, M. Zhu, and W. Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*, 2022.

[18] X. Liu, T. Zhang, Y. Gu, I. L. Iong, Y. Xu, X. Song, S. Zhang, H. Lai, X. Liu, H. Zhao, et al. Visualagentbench: Towards large multimodal models as visual foundation agents. *arXiv preprint arXiv:2408.06327*, 2024.

[19] Z. Liu, Z. Cen, V. Isenbaev, W. Liu, S. Wu, B. Li, and D. Zhao. Constrained variational policy optimization for safe reinforcement learning. In *International Conference on Machine Learning*, pages 13644–13668. PMLR, 2022.

[20] Z. Liu, J. Qiu, S. Wang, J. Zhang, Z. Liu, R. Ram, H. Chen, W. Yao, S. Heinecke, S. Savarese, et al. Mcpeval: Automatic mcp-based deep evaluation for ai agent models. *arXiv preprint arXiv:2507.12806*, 2025.

[21] E. Lumer, A. Gulati, V. K. Subbiah, P. H. Basavaraju, and J. A. Burke. Scalemcp: Dynamic and auto-synchronizing model context protocol tools for llm agents. *arXiv preprint arXiv:2505.06416*, 2025.

[22] S. Nair and C. Finn. Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. *arXiv preprint arXiv:1909.05829*, 2019.

[23] G. Neumann. Variational inference for policy search in changing situations. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 817–824, 2011.

[24] OpenAI. Gpt-4v(ision) technical work and authors. `https://openai.com/contributions/gpt-4v`, 2023.

[25] C. Pan, B. Yaman, T. Nesti, A. Mallik, A. G. Allievi, S. Velipasalar, and L. Ren. Vlp: Vision language planning for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14760–14769, 2024.

[26] J. Pan, Y. Zhang, N. Tomlin, Y. Zhou, S. Levine, and A. Suhr. Autonomous evaluation and refinement of digital agents. In *First Conference on Language Modeling*, 2024.

[27] G. Parascandolo, L. Buesing, J. Merel, L. Hasenclever, J. Aslanides, J. B. Hamrick, N. Heess, A. Neitz, and T. Weber. Divide-and-conquer monte carlo tree search for goal-directed planning. *arXiv preprint arXiv:2004.11410*, 2020.

[28] X. B. Peng, A. Kumar, G. Zhang, and S. Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

[29] Z. Qi, X. Liu, I. L. Iong, H. Lai, X. Sun, X. Yang, J. Sun, Y. Yang, S. Yao, T. Zhang, et al. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. *arXiv preprint arXiv:2411.02337*, 2024.

[30] C. Rawles, A. Li, D. Rodriguez, O. Riva, and T. Lillicrap. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[32] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[33] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36, 2024.

[34] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[35] R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[36] G. Team, P. Georgiev, V. I. Lei, R. Burnell, L. Bai, A. Gulati, G. Tanzer, D. Vincent, Z. Pan, S. Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[37] D. Toyama, P. Hamel, A. Gergely, G. Comanici, A. Glaese, Z. Ahmed, T. Jackson, S. Mourad, and D. Precup. Androidenv: A reinforcement learning platform for android. *arXiv preprint arXiv:2105.13231*, 2021.

[38] T. Wang, Z. Wu, J. Liu, J. Hao, J. Wang, and K. Shao. Distrl: An asynchronous distributed reinforcement learning framework for on-device control agents. *arXiv preprint arXiv:2410.14803*, 2024.

[39] Q. Wu, S. S. Zhan, Y. Wang, Y. Wang, C.-W. Lin, C. Lv, Q. Zhu, and C. Huang. Variational delayed policy optimization. *arXiv preprint arXiv:2405.14226*, 2024.

[40] Y. Yan, S. Wang, J. Du, Y. Yang, Y. Shan, Q. Qiu, X. Jia, X. Wang, X. Yuan, X. Han, et al. Mcpworld: A unified benchmarking testbed for api, gui, and hybrid computer use agents. *arXiv preprint arXiv:2506.07672*, 2025.

[41] H. Yang, S. Yue, and Y. He. Auto-gpt for online decision making: Benchmarks and additional opinions. *arXiv preprint arXiv:2306.02224*, 2023.

[42] J. Yang, H. Zhang, F. Li, X. Zou, C. Li, and J. Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

[43] Z. Yang, C. Garrett, D. Fox, T. Lozano-Pérez, and L. P. Kaelbling. Guiding long-horizon task and motion planning with vision language models. *arXiv preprint arXiv:2410.02193*, 2024.

[44] Z. Yang, J. Liu, Y. Han, X. Chen, Z. Huang, B. Fu, and G. Yu. Appagent: Multimodal agents as smartphone users. *arXiv preprint arXiv:2312.13771*, 2023.

[45] C. Zhang, Z. Yang, J. Liu, Y. Han, X. Chen, Z. Huang, B. Fu, and G. Yu. Appagent: Multimodal agents as smartphone users. *arXiv preprint arXiv:2312.13771*, 2023.

[46] Z. Zhang and A. Zhang. You only look at screens: Multimodal chain-of-action agents. *arXiv preprint arXiv:2309.11436*, 2023.

[47] Q. Zhao, H. Fu, C. Sun, and G. Konidaris. Epo: Hierarchical llm agents with environment preference optimization. *arXiv preprint arXiv:2408.16090*, 2024.

[48] B. Zheng, B. Gou, J. Kil, H. Sun, and Y. Su. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*, 2024.

[49] S. Zhou, F. F. Xu, H. Zhu, X. Zhou, R. Lo, A. Sridhar, X. Cheng, T. Ou, Y. Bisk, D. Fried, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

[50] Y. Zhou, A. Zanette, J. Pan, S. Levine, and A. Kumar. Archer: Training language model agents via hierarchical multi-turn rl. In *Forty-first International Conference on Machine Learning*, 2024.

A Limitations and Challenges

We have empirically demonstrated that our VSC-RL can effectively address the learning efficiency issue commonly existing in complex sequential decision-making tasks. However, there are still some limitations and challenges in VSC-RL, as discussed below.

Fine-tuning VLM as Subgoal Generator. Benefiting from the general reasoning ability of the proprietary VLM, we empirically found that the performance of VSC-RL is improved by the feasible subgoals. However, for the control task from a specific domain, it is worth fine-tuning the open-source VLM as the subgoal generator.

Hierarchical RL Approaches. Additionally, the VLM in VSC-RL can not only be viewed as the subgoal generator, but also as the high-level policy in the context of hierarchical RL. It is valuable to investigate enhancing VSC-RL with the hierarchical RL approaches [47, 50].

Future Challenging Applications. In this work, we mainly consider the mobile device and web control tasks, two representative complex control problems, as the evaluation benchmarks. The theoretical and empirical results presented in this work imply that VSC-RL has great potential in addressing other challenging open problems, such as MCP-enabled control tasks [20, 40, 21].

B Implementation Details

As shown in Table 2, we summarize LLMs and VLMs used in VSC-RL. For AitW tasks[30], we built our VSC-RL on the open repository of DigiRL [4]. Hyperparameter settings are listed in Table 3, and each run of VSC-RL takes approximately 24 hours on 1 NVIDIA GeForce RTX 4090 GPU and 8 Intel Xeon CPUs. For WebArena-Lite tasks [18], we built our VSC-RL on the open repository of WebRL [29] and follow the online RL loop of interaction, filtering, and update. To ensure a fair comparison, we remove WebRL's curriculum learning component, so that all methods are trained solely on the same task set. For VSC-RL and other baselines, we apply the same actor perplexity-based filtering strategy as WebRL to select replay data, ensuring consistency in experience quality. Hyperparameter settings are listed in Table 4, and each run of VSC-RL takes approximately 24 hours on 8 NVIDIA GeForce RTX 4090 GPUs and 8 Intel Xeon CPUs. Specifically, we report some results on AitW (Set-of-Marks, AppAgent, CogAgent, AutoUI, and Filtered BC) and WebArena-Lite (SFT, Filtered BC, and AWR) from the literature [4, 29].

Table 2: The summary of LLMs and VLMs used in VSC-RL.

Component	Task		Description
	AitW	WebArena-Lite	
Subgoal Generator	Gemini-1.5-Pro	Gemini-1.5-Pro	Autonomously decompose the goal into subgoals.
Reference Actor π_{ref}	AutoUI-Base	Llama-3.1-8B	Provide reference action for imitation.
Target Actor π	AutoUI-Base	Llama-3.1-8B	Make decisions to maximize subgoal-conditioned return.
Evaluator	Gemini-1.5-Pro	Llama-3.1-8B	Autonomously evaluate the goal's or subgoal's success.

Table 3: Hyperparameters settings of VSC-RL on AitW tasks.

Hyperparameter	Value
Batch Size	4
Total Trajectories	1,000
Discount Factor	0.5
Learning Rate	1e-4
Update Epoch (Actor Equation (5))	20
Update Epoch (Critic Equation (6))	5
Update Epoch (Actor Equation (7))	20
Update Epoch (Instruction-level Critic)	5
Maximum Gradient Norm	0.01

Table 4: Hyperparameter settings of VSC-RL on WebArena-Lite tasks.

Hyperparameter	Value
Batch Size	128
Total Trajectories	1,000
Discount Factor	0.9
Learning Rate	1e-6
Update Epoch (Actor Equation (5))	1
Update Epoch (Critic Equation (6))	1
Update Epoch (Actor Equation (7))	1
Maximum Gradient Norm	1.0

C Theoretical Analysis

Proposition C.1 (Subgoal-Conditioned Optimization Objective)**.** *Given a goal g with corresponding subgoals $\{sg_i\}_{i=1}^N$ and a subgoal-conditioned target policy π, the objective of*

$$\max_{\pi} \mathop{\mathbb{E}}_{\tau \sim p_\pi(\tau|g)} [\log p(O|\tau, g)]$$

is equivalent to the objective of

$$\max_{\pi} \sum_{i=1}^{N} \left[\mathop{\mathbb{E}}_{\tau_i \sim p_\pi(\tau_i|sg_i)} [\log p(O|\tau_i, sg_i)] \right].$$

Proof. We have

$$\log p(O|\tau, g) \propto \exp\left(\frac{\mathcal{J}(\tau|g)}{\alpha}\right) = \exp\left(\frac{\sum_{i=1}^{N} [\mathcal{J}(\tau_i|sg_i)]}{\alpha}\right).$$

So, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\tau \sim p_\pi(\tau|g)} [\log p(O|\tau, g)] \\
&= \mathop{\mathbb{E}}_{\tau \sim p_\pi(\tau, g)} [\mathcal{J}(\tau, g)] \\
&= \mathop{\mathbb{E}}_{\tau \sim \prod_{i=1}^{N} p_\pi(\tau_i, sg_i)} [\mathcal{J}(\tau|g)] \\
&= \mathop{\mathbb{E}}_{\tau \sim \prod_{i=1}^{N} p_\pi(\tau_i, sg_i)} \left[\sum_{i=1}^{N} \mathcal{J}(\tau_i|sg_i) \right] \\
&= \sum_{i=1}^{N} \left[\mathop{\mathbb{E}}_{\tau_i \sim p_\pi(\tau_i, sg_i)} [\mathcal{J}(\tau_i|sg_i)] \right]
\end{aligned}$$

Due to the fact that

$$\log p(O|\tau_i, sg_i) \propto \exp\left(\frac{\mathcal{J}(\tau_i|sg_i)}{\alpha}\right).$$

Therefore, we have

$$\max_{\pi} \mathop{\mathbb{E}}_{\tau \sim p_\pi(\tau|g)} [\log p(O|\tau, g)] \Rightarrow \max_{\pi} \sum_{i=1}^{N} \left[\mathop{\mathbb{E}}_{\tau_i \sim p_\pi(\tau_i|g_i)} [\log p(O|\tau_i, sg_i)] \right]$$

□

Proposition C.2 (Subgoal-conditioned Difference Bound)**.** *Given goal-conditioned reference policy π_{ref} and subgoal-conditioned target policy π, the goal-conditioned KL divergence of a given goal g has the upper bound of subgoal-conditioned KL divergence of corresponding subgoals $\{sg_i\}_{i=1}^N$ as follows:*

$$KL(p_\pi(\tau|g)||p_{\pi_{ref}}(\tau|g)) \leq \sum_{i=1}^{N} \left[KL(p_\pi(\tau_i|sg_i)||p_{\pi_{ref}}(\tau_i|g)) \right].$$

Proof. We have

$$\begin{aligned}
p_\pi(\tau|g) &= \rho(s_0) \prod_{t=0}^{H} P(s_{t+1}|s_t, a_t)\pi(a_t|s_t, g), \\
&= \prod_{i=1}^{N} p_\pi(\tau_i|sg_i). \\
&\leq p_\pi(\tau_i|sg_i)(i = 1, \cdots, N)
\end{aligned}$$

Similarly, we have

$$p_{\pi_{\text{ref}}}(\tau|g) = \prod_{i=1}^{N} p_{\pi_{\text{ref}}}(\tau_i|g)$$

Therefore,

$$\begin{aligned}
&\text{KL}(p_\pi(\tau|g)||p_{\pi_{\text{ref}}}(\tau|g)) \\
&= \mathop{\mathbb{E}}_{\tau\sim p_\pi(\tau|g)} [\log p_\pi(\tau|g) - \log p_{\pi_{\text{ref}}}(\tau|g)] \\
&= \mathop{\mathbb{E}}_{\tau\sim p_\pi(\tau|g)} \left[\sum_{i=1}^{N} \log p_\pi(\tau_i|sg_i) - \sum_{i=1}^{N} \log p_{\pi_{\text{ref}}}(\tau_i|g)\right] \\
&= \sum_{i=1}^{N} \left[\mathop{\mathbb{E}}_{\tau\sim p_\pi(\tau|g)} [\log p_\pi(\tau_i|sg_i) - \log p_{\pi_{\text{ref}}}(\tau_i|g)]\right] \\
&\leq \sum_{i=1}^{N} \left[\mathop{\mathbb{E}}_{\tau_i\sim p_\pi(\tau_i|sg_i)} [\log p_\pi(\tau_i|sg_i) - \log p_{\pi_{\text{ref}}}(\tau_i|g)]\right] \\
&= \sum_{i=1}^{N} [\text{KL}(p_\pi(\tau_i|sg_i)||p_{\pi_{\text{ref}}}(\tau_i|g))]
\end{aligned}$$

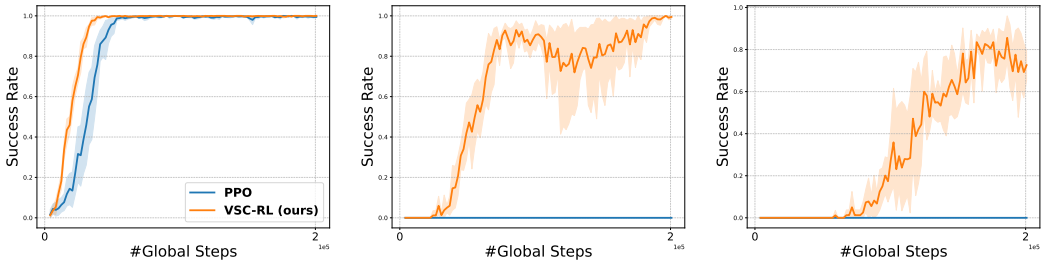
□

D Additional Experiments: MiniGrid

We also evaluate our VSC-RL on the toy vision-language decision-making tasks, MiniGrid [10]. We select the PPO [32] as the baseline, and we apply VSC-RL in the PPO for a fair comparison. We built our VSC-RL on the open repository of babyAI [10], hyperparameter settings are listed in Table 5. Overall, as shown in Figure 4, our VSC-RL outperforms the baseline in all tasks remarkably, especially in the difficult task with the increasing number of rooms. From the result of MultiRoom-N2-v0 shown in Figure 4(a), we can tell that although PPO and VSC-RL both successfully reach 100% success rate, our VSC-RL shows a better sample efficiency. For MultiRoom-N4-v0 (Figure 4(b)) and MultiRoom-N6-v0 (Figure 4(c)) where PPO is not able to learn any useful policy, while VSC-RL exhibits strong performance of 100% and 80% success rate, respectively.

Table 5: Hyperparameter settings of VSC-RL on MiniGrid.

Hyperparameter	Value
Batch Size	256
Total Steps	200,000
Discount Factor	0.99
Learning Rate	1e-3
Network Layers (Image)	3
Network Layers (Text)	1
Network Layers (Actor)	2
Network Layers (Critic)	2
Update Epoch (Actor Equation (5))	4
Update Epoch (Critic Equation (6))	4
Update Epoch (Actor Equation (7))	4
Activation	ReLU
Optimizer	Adam

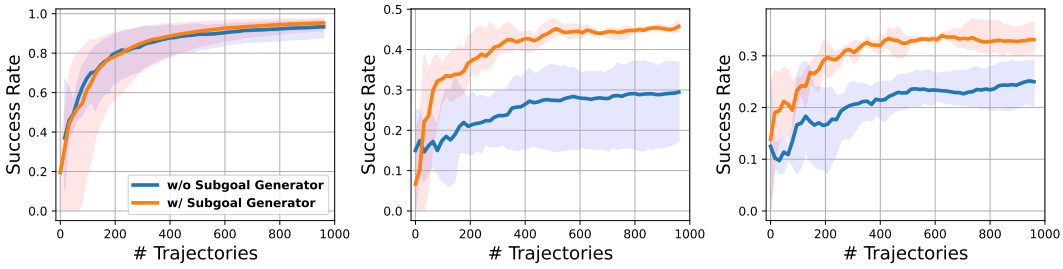


(a) MultiRoom-N2-v0 (b) MultiRoom-N4-v0 (c) MultiRoom-N6-v0

Figure 4: Learning curves on MultiRoom tasks of (a) 2 rooms, (b) 4 rooms, and (c) 6 rooms.

E Additional Experiments: Subgoal Generator in VSC-RL

Improvement from Subgoal Generator. We investigate the importance of the subgoal generator in our VSC-RL on the Web Shopping subsets with different horizon lengths (short, medium and long). We implement VSC-RL with the original goal instead of the subgoals generated from VLM. As shown in Figure 5, the subgoal generator can effectively improve the performance across all types of Web Shopping tasks via autonomously decomposing the original goal into subgoals. Especially, the subgoal generator can effectively enhance the 50% and 32% performance in the Web Shopping medium and long tasks, respectively.



(a) Web Shopping (Short) (b) Web Shopping (Medium) (c) Web Shopping (Long)

Figure 5: Success rate on Web Shopping (a) Short, (b) Medium and (c) Long tasks of VSC-RL with and without subgoal generator.

Verification of Subgoal Generator. To investigate the quality and feasibility of the generated subgoals, we manually verify the results of the subgoal generator on 135 trajectories from the AitW human demonstration. There are $135(100\%)$ goals that are decomposed into feasible subgoals successfully, and the final goal can be accomplished by reaching these subgoals sequentially. Specifically, there are $123(91.1\%)$ goals that are decomposed into subgoals completely aligning with the human demonstration. For the remaining $12(8.9\%)$ goals, the subgoal generator provides alternative subgoals different from human demonstration, but still can successfully arrive at the final goal.

F Additional Experiments: WebArena-Lite

We also evaluate VSC-RL on WebArena-Lite [18], a human-verified subset of the WebArena benchmark [49] containing 165 realistic web tasks across five websites. Each task involves complex HTML-based observations with 30 steps of horizon. Following WebRL [29], we adopt the pretrained outcome-supervised reward model (ORM) to autonomously evaluate the task's success.

We compare VSC-RL with several SOTA baselines adapted to web environments, including supervised fine-tuning (SFT), Filtered BC [26], and RL-based agents (AWR [28], DigiRL [4], and WebRL [29]). Specifically, to ensure a fair comparison, we remove the curriculum learning component of WebRL so that all methods are trained solely on the same task set. Each method is tested on 1 single run, consistent with existing works [29].

As shown in Figure 6, we present the learn curves on the WebArena-Lite. The imitation-based agents, SFT and Filtered BC achieve relatively limited performance, with success rates stagnating around 20.6% and 23.0%, respectively. AWR achieves approximately 28.5% success rate via solely leveraging the offline RL technique, which is still limited compared to the online RL methods. DigiRL and WebRL exhibit similar performance trends, both plateauing around a 31.0% success rate. Our VSC-RL consistently outperforms the other methods, achieving the highest success rate of approximately 34.5%. Overall, our VSC-RL can achieve superior learning efficiency and final performance on the WebArena-Lite. Specifically, the additional results shown in Appendix F demonstrate that our VSC-RL can achieve the best performance across all types of tasks on the WebArena-Lite.

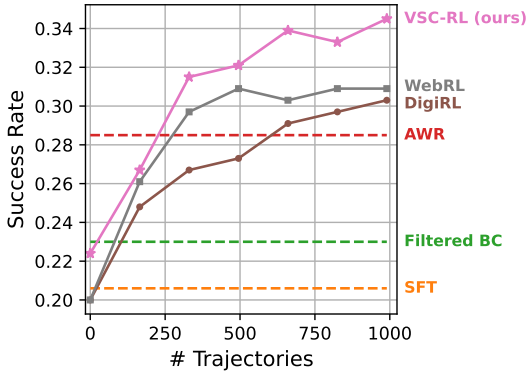
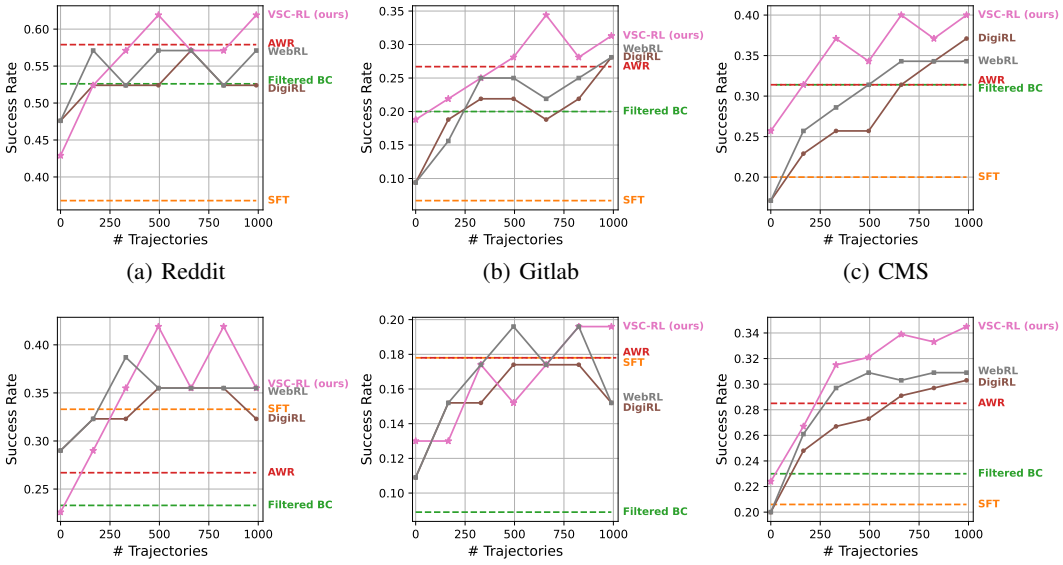


Figure 6: Learning curves on WebArena-Lite.

As shown in Table 6, we present the evaluation performance on specific WebArena-Lite tasks, including Reddit, Gitlab, CMS (online store content management system), Map (OpenStreetMap), and OSS (OneStopShop). We also provide the learning curves in Figure 7. These empirical results demonstrate that our VSC-RL achieve the best performance across on all types of tasks, significantly surpassing existing SOTAs.

Table 6: The evaluated performance on WebArena-Lite. The best performance is **highlighted**.

Method	Task (# Ratio)					
	Reddit (12.7%)	Gitlab (19.4%)	CMS (21.2%)	Map (18.8%)	OSS (27.9%)	All (100.0%)
SFT	36.8%	6.7%	20.0%	33.3%	17.8%	20.6%
Filtered BC	52.6%	20.0%	31.4%	23.3%	8.9%	23.0%
AWR	57.9%	26.7%	31.4%	26.7%	17.8%	28.5%
DigiRL	52.4%	28.1%	37.1%	32.3%	15.2%	30.3%
WebRL	57.1%	28.1%	34.3%	**35.5%**	15.2%	30.9%
VSC-RL (ours)	**61.9%**	**31.3%**	**40.0%**	**35.5%**	**19.6%**	**34.5%**



(a) Reddit (b) Gitlab (c) CMS

(d) Map (e) OSS (f) All

Figure 7: Learning curves on (a) Reddit, (b) Gitlab, (c) CMS, (d) Map, (e) OSS, and (f) All tasks.

G Ablation Results of VSC-RL

The ablation results on the Web Shopping task, shown in Table 7, evaluate the impact of various key components of VSC-RL. Introducing subgoals to AutoUI can yield a marginal improvement ($+1.6\%$) on the average success rate from 16.2% to 17.8%, which offers limited benefit without involving the RL process. For VSC-RL, removing subgoals entirely leads to a substantial performance drop (-7.8%) compared to providing limited (50% less) subgoals (-5.2%). These results imply that subgoals can efficiently improve performance by providing immediate informative reward signals in the RL process. Additionally, we investigate the different optimization components of SGC-ELBO(eq. (3)) in VSC-RL. Removing the policy gradient (Equation (5)) and imitation loss (Equation (7)) result in decreased performance with -7.8% and -10.4%, respectively. Overall, each optimization component of VSC-RL contributes meaningfully to its effectiveness, aligning with our main statements in Section 4.2.

Table 7: Ablation results of VSC-RL on the Web Shopping. The best performance is **highlighted**.

Method	Web Shopping		
	Train	Test	Average
AutoUI	14.6%	17.7%	16.2%
w/ subgoals	16.7%	18.8%	17.8%
VSC-RL	**64.6%**	**58.3%**	**61.5%**
w/o subgoals	55.2%	52.1%	53.7%
w/ limited (50% less) subgoals	57.3%	55.2%	56.3%
w/o policy gradient (Equation (5))	56.3%	51.0%	53.7%
w/o imitation loss (Equation (7))	55.2%	46.9%	51.1%

H Additional Experiments: Incorrect Subgoals

To assess the robustness of VSC-RL with respect to subgoal quality, we conducted an ablation study in which the agent was intentionally provided with incorrect subgoals (e.g., incorrect websites or irrelevant search items) during training. As shown in Table 8, VSC-RL fails to learn effectively under these conditions, resulting in a substantial drop in performance. These results highlight the critical importance of subgoal quality in the learning process and emphasize the necessity of reliable subgoal generation for VSC-RL.

Table 8: Performance of VSC-RL with incorrect subgoals.

	Correct	Incorrect
Train	64.6%	19.8%
Test	58.3%	17.7%

I Additional Experiments: Ablations on Subgoal Generator

We additionally integrated Qwen2.5-VL-72B and Qwen2.5-VL-3B [6] into VSC-RL, respectively. As shown in Table 9, Gemini-1.5-Pro consistently outperforms all Qwen2.5-VL models on both the train and test sets. Among the Qwen2.5-VL models, Qwen2.5-VL-72B outperforms Qwen2.5-VL-3B, suggesting it produces higher-quality subgoals. The result indicates that the choice of VLM significantly affects the quality of generated subgoals, which in turn impacts performance. In future work, we plan to fine-tune the open-source VLM for generating subgoals with high quality in the specific domain.

Table 9: Performance comparison with different VLM-based subgoal generators.

	Gemini-1.5-Pro	Qwen2.5-VL-72B	Qwen2.5-VL-3B
Train	64.6%	60.4%	58.3%
Test	58.3%	56.3%	55.2%

J Failure Cases Analysis

We analyze the failure cases of VSC-RL on the Web Shopping test sets. We manually evaluate the failure tasks and summarise the reasons as follows, categorising them as "Stuck Issue", "Wrong Navigation", and "Technical Issue". Specifically, for our VSC-RL,

- "Stuck Issue" (37.5%), common failure modes include being unable to close the Chrome started pop-up (12.5%) and repeatedly typing in the search bar without submitting the query (25.0%). These indicate challenges in interface interaction and accurate action execution.
- "Wrong Navigation" (30.0%), the agent often reaches the correct website but deviates from the target task by entering unrelated categories, showing challenges in interpreting navigation menus.
- "Technical Issue" (32.5%) comprises cases such as getting stuck at CAPTCHA verification (12.5%), page loading failures (2.5%), and LLM misjudgments (7.5%), which reflect either site-level obstacles or model-level perception errors.

To align with the ablation results, Figure 8 shows the failure case distribution in the ablation results of VSC-RL. Removing subgoals and policy gradient mainly increases "Wrong Navigation" errors, while removing imitation loss raises both "Wrong Navigation" and "Stuck Issue".

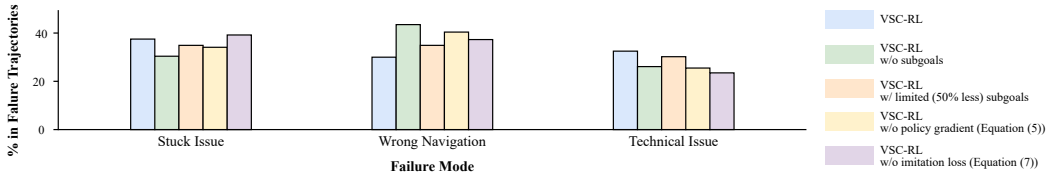


Figure 8: Failure case distribution in the ablation results of VSC-RL on the Web Shopping tasks.

K Prompt Example

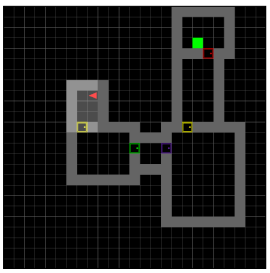
We provide the prompt example of the subgoal generator in our VSC-RL for a given goal and corresponding decomposed subgoals in the MultiRoom (Figure 9), AitW (Figure 10) and WebArena-Lite (Figure 11) tasks.

Prompt

=====Example=====

Q: Based on the provided {*example_image*} of the maze,
PLEASE provide the sequence of rooms for the agent (RED triangle) to reach the goal (GREEN sqaure).

A: To reach the goal, the agent need to open to doors to across rooms:
1. Open the YELLOW DOOR
2. Open the GREEN DOOR
3. Open the PURPLE DOOR
4. Open the YELLOW DOOR
5. Open the RED DOOR
6. Reach the GREEN GOAL



example_image

=====Your Turn=====

Respond in this format:

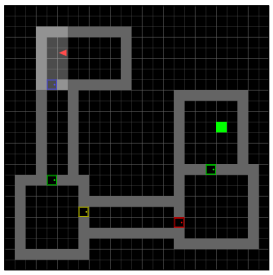
Q: Based on the provided {*image*} of the maze,
PLEASE provide the sequence of rooms for the agent (RED triangle) to reach
the goal (GREEN sqaure).

A: To reach the goal, the agent need to open to doors to across rooms
< DOORS >

Response

Q: Based on the provided {*test_image*} of the maze,
PLEASE provide the sequence of rooms for the agent (RED triangle) to reach the goal (GREEN sqaure).

A: To reach the goal, the agent need to open to doors to across rooms:
1. Open the PURPLE DOOR
2. Open the GREEN DOOR
3. Open the YELLOW DOOR
4. Open the RED DOOR
5. Open the GREEN DOOR
6. Reach the GREEN GOAL



test_image

Figure 9: Prompt example for our subgoal generator for MultiRoom Benchmark. The generator decomposes the goal of navigating the maze into subgoals like opening specific doors sequentially.

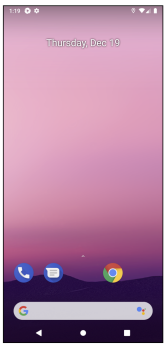
Prompt

=====Example=====

Q: Task: Search for "macbook pro" on walmart, select the first entry, and add it to the cart.'
Based on the provided screenshot: {*example_image*},
PLEASE descompse the TASK to the sequence of specific SUBTASKS.

A: To complete your requested task, you would need to accomplish these subtasks on a device:

1. Open a browser.
2. Go to the Walmart website.
3. Use the search bar to search for "macbook pro".
4. Click on the first search result.
5. Click on the "Add to Cart" button on the product page.



example_image

=====Your Turn=====

Respond in this format:

Q: Task: {*goal*},
Based on the provided screenshot: {*image*},
PLEASE descompse the TASK to the sequence of specific SUBTASKS.

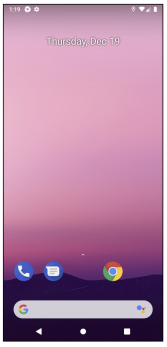
A: To complete your requested task, you would need to accomplish these subtasks on a device:
<SUBTASKS>

Response

Q: Task: Install the Calendar app
Based on the provided screenshot: {*test_image*},
PLEASE descompse the TASK to the sequence of specific SUBTASKS.

A: To complete your requested task, you would need to accomplish these subtasks on a device:

1. Open the Google Play Store app.
2. Search for \"Calendar\" in the Play Store search bar.
3. Select the desired Calendar app from the search results.
4. Click the \"Install\" button.
5. Wait for the app to download and install.
6. Open the app once installed.



test_image

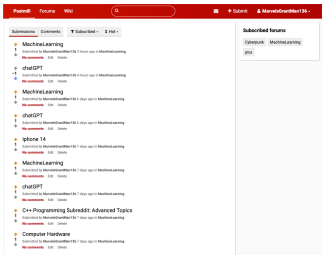
Figure 10: Prompt example for our subgoal generator for tasks in AitW dataset. The generator decomposes user commands into actionable subgoals, such as opening a browser, searching for items, and selecting desired results.

Prompt

You are an expert web automation agent. Your job is to decompose a user TASK into SUBTASKS for a web agent to follow. Each subtask should be a instruction that represents a meaningful intermediate goal, which can be achieved by performing several low-level actions (not just a single atomic action). Do NOT generate subgoals for steps where the user needs to view, check, or record information. Please reference to the EXAMPLE TASK and EXAMPLE SUBTASKS, and keep the subgoals as close as possible in format, complexity and granularity to the example. If the task involves a date range, please specify the concrete date range in the subgoal like in the example.

=====Website Description=====

Reddit: a web forum and community platform where users can browse, post, and comment on topics in different forums (subreddits). Users can navigate to the Forums page to see a list of all available forums, browse forums by tags or alphabetically, and select a specific forum to view its posts. Within a forum, users can sort posts by criteria such as Hot, New, Top, Controversial, or Most Commented. Each post can be viewed in detail, and users can read or add comments. Users can also submit new posts to a forum. Typical actions include navigating to a forum, sorting and browsing posts, reading and commenting on posts, and submitting new posts.



example_html

=====Example=====

Q: Task: Among the top 10 post in \"books\" forum, is there any post talks about supporting local book stores? If so, tell me the organizations involved
Based on the provided html: {*example_html*},
PLEASE descompse the TASK to the sequence of specific SUBTASKS.

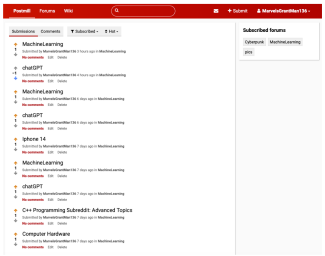
A: To complete your requested task, you would need to accomplish these subtasks on a device:
1. Navigate to the '/f/books' forum page.
2. Sort posts on the '/f/books' forum page by top posts of all time to check the top 10 posts.

=====Your Turn=====

Respond in this format:

Q: Task: {*goal*},
Based on the provided html: {*html*},
PLEASE descompse the TASK to the sequence of specific SUBTASKS.

A: To complete your requested task, you would need to accomplish these subtasks on a device:
<SUBTASKS>



test_html

Response

Q: Task: Create a discussion post about \"Iphone 14\" in a relevant subreddit and ask users for their opinions with the simple prompt, \"your opinion\".
Based on the provided html: {*test_html*},
PLEASE descompse the TASK to the sequence of specific SUBTASKS.

A: To complete your requested task, you would need to accomplish these subtasks on a device:
1. Navigate to a relevant subreddit for \"iPhone 14\" discussions (e.g., r/apple or r/iPhone).
2. Create a new post in the chosen subreddit with the title \"iPhone 14\" and the body text \"Your opinion\".

Figure 11: Prompt example for our subgoal generator for WebArena-Lite tasks. The subgoal generator decomposes a user task into a sequence of more specific and actionable instructions based on the provided HTML context.

L Qualitative Example

We provide qualitative examples of VSC-RL applied to MultiRoom (Figure 12), AitW General (Figure 13), AitW Web Shopping (Figure 14), WebArena-Lite Map (Figure 15), and WebArena-Lite CMS (Figure 16).

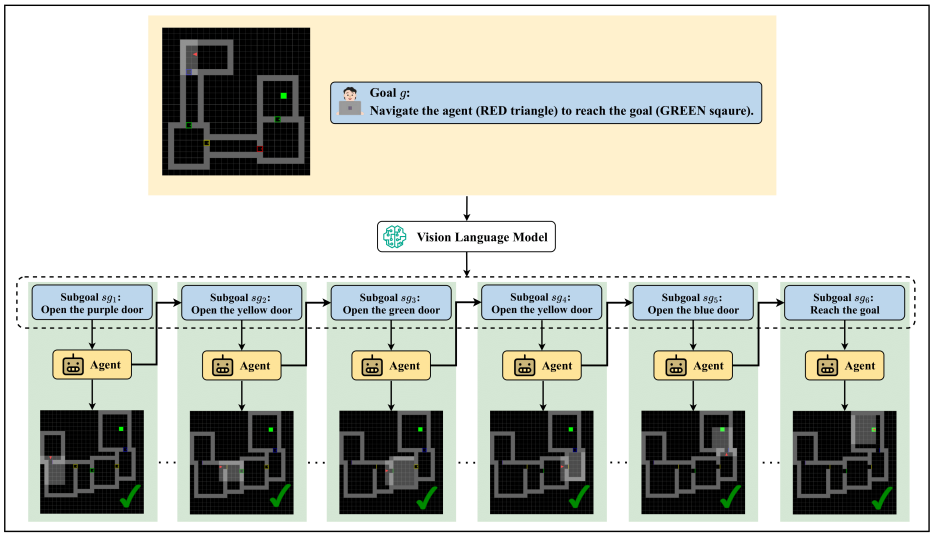


Figure 12: Qualitative example of VSC-RL on the Multiroom task.

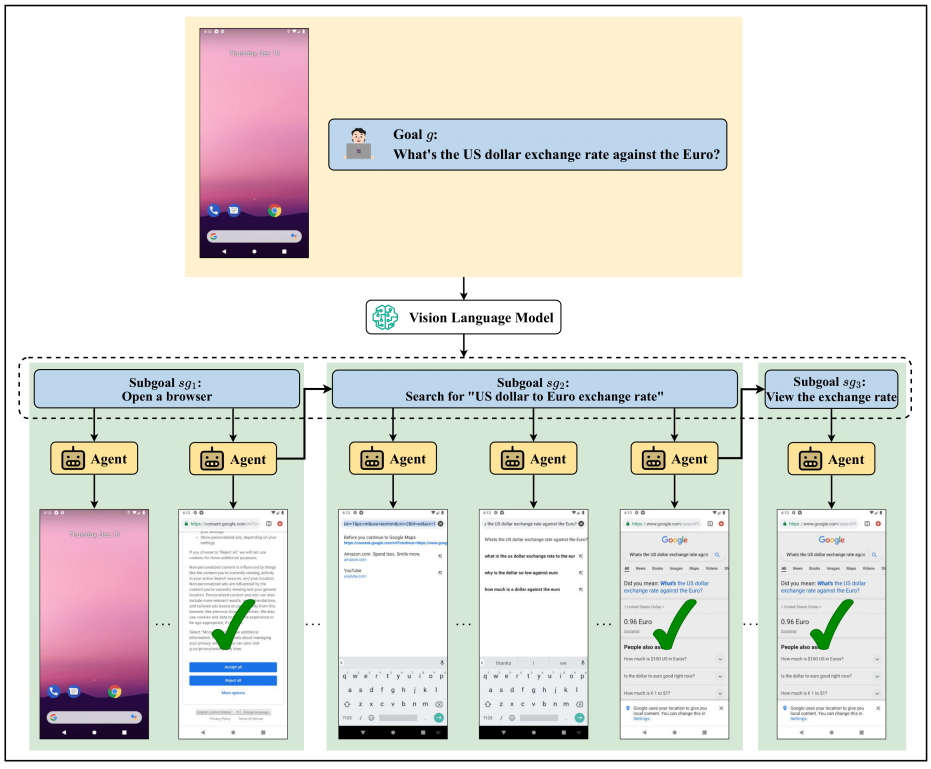


Figure 13: Qualitative example of VSC-RL on the AitW General task.

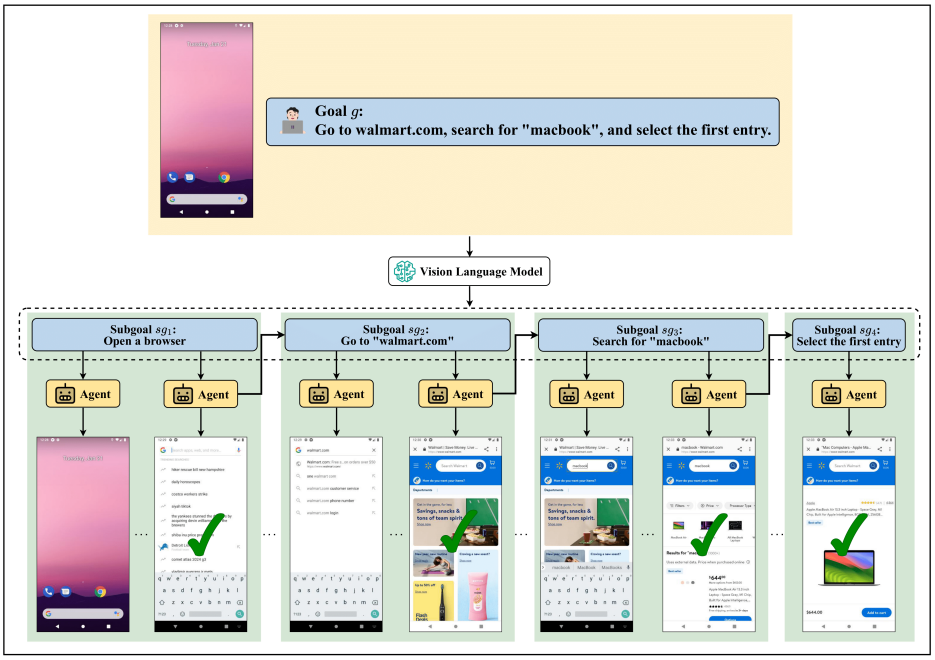


Figure 14: Qualitative example of VSC-RL on the AitW Web Shopping task.

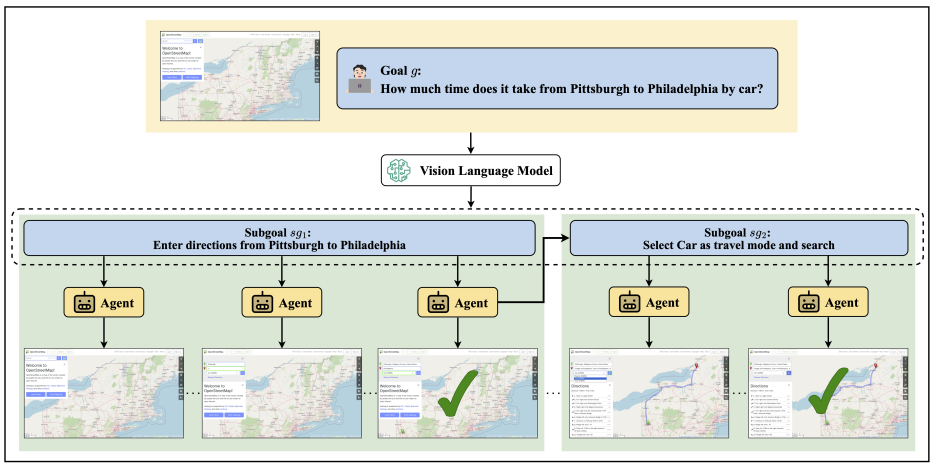


Figure 15: Qualitative example of VSC-RL on the WebArena-Lite Map task.

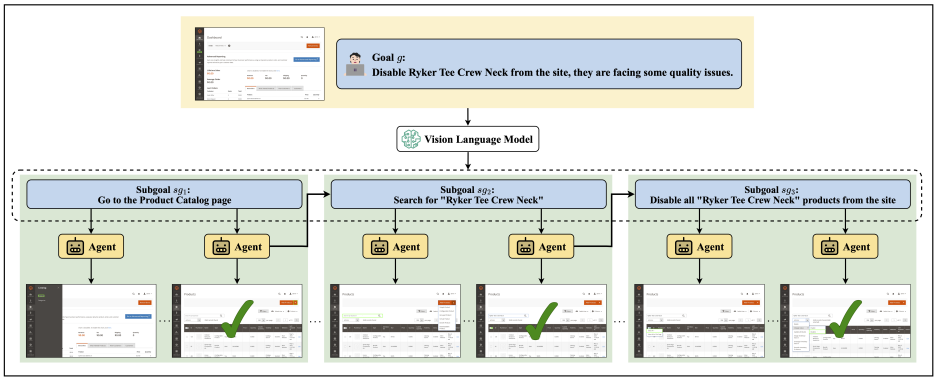


Figure 16: Qualitative example of VSC-RL on the WebArena-Lite CMS task.