Guiding Reinforcement Learning with Selective Vision-Language Model Supervision

Matteo Merler[1], Giovanni Bonetta[1] and Bernardo Magnini[1]

[1]*Fondazione Bruno Kessler, Trento, Italy*

Abstract

We propose a framework that augments a model-free Reinforcement Learning (RL) agent with selective guidance from a pre-trained Vision-Language Model (VLM). Our system is designed to assist the RL agent, which starts from scratch and has no prior notion of the environment, by leveraging the VLM's common-sense knowledge to support its decision making. Rather than relying on the VLM at every timestep, the agent monitors its own uncertainty during training and defers to the VLM only when it is unsure about which action to take. Uncertainty is measured using the entropy of the policy distribution, and guidance is triggered when this entropy exceeds a predefined threshold. To reduce computational overhead, we introduce a stochastic gating mechanism that limits the frequency of VLM queries, along with a cache that stores past VLM responses for reuse. Experiments show that our method leads to more stable learning dynamics compared to standard PPO, with reduced variance across runs. In the `FrozenLake` environment, we observe that VLM guidance is primarily utilized during the early stages of training, gradually diminishing as the agent becomes more confident. This suggests that our selective guidance mechanism can support early exploration without hindering long-term autonomous behavior.

Keywords

Reinforcement Learning, Vision-Language Models, Policy Guidance

1. Introduction

Reinforcement Learning (RL) [1] has proven effective for training agents across a wide range of domains, including games [2, 3, 4, 5] and robotics [6, 7, 8], among others. At its core, an agent interacts with an environment and learns how to behave through experience, by maximizing a reward signal observed as a result of its actions. Typically, model-free RL methods [9, 10] start learning with no prior information about the environment; instead, they *explore* in order to learn the dynamics of the world, before being able to *exploit* what they learned by maximizing the expected reward. This process can be particularly challenging in long-horizon tasks where the reward signal is sparse, leading to sample inefficiency [11].

Meanwhile, Large Foundation Models (LFMs) [12] such as Large Language Models (LLMs) [13, 14, 15] and Vision-Language Models (VLMs) [16, 17, 18] demonstrate strong reasoning capabilities and encode extensive common-sense priors from internet-scale pretraining. These models can be harnessed to guide RL agents, providing high-level priors that bootstrap behavior and mitigate early-stage exploration challenges. As such, recent work has explored using LFMs as agents [19, 20, 21], showing that they can directly generate actions or high-level plans by conditioning on the current state and task specification.

However, these methods often rely on the computationally expensive LFM at every timestep, impeding the learning of a lightweight, task-specialized policy. To fix this, we take inspiration from dual-process theories of cognition [22], which distinguish between a fast, habitual decision-making system (System 1) and a slow, deliberative reasoning system (System 2). We propose a hybrid RL framework in which a model-free policy operates as the decision-maker (acting as System 1), while a VLM (acting as System 2) is selectively invoked, when the agent exhibits low confidence, to provide high-level guidance in ambiguous or unfamiliar states (Figure 1).

We evaluate the approach on the `FrozenLake` environment from Gymnasium [23] and show that our *VLM guidance* approach leads to increased stability compared to a vanilla PPO [9] baseline, showing that targeted, uncertainty-driven VLM guidance is a viable strategy for improving learning efficiency in model-free RL.

ECAI Workshop on AI-based Planning for Complex Real-World Applications (CAIPI'25)
mmerler@fbk.eu (M. Merler); gbonetta@fbk.eu (G. Bonetta); magnini@fbk.eu (B. Magnini)

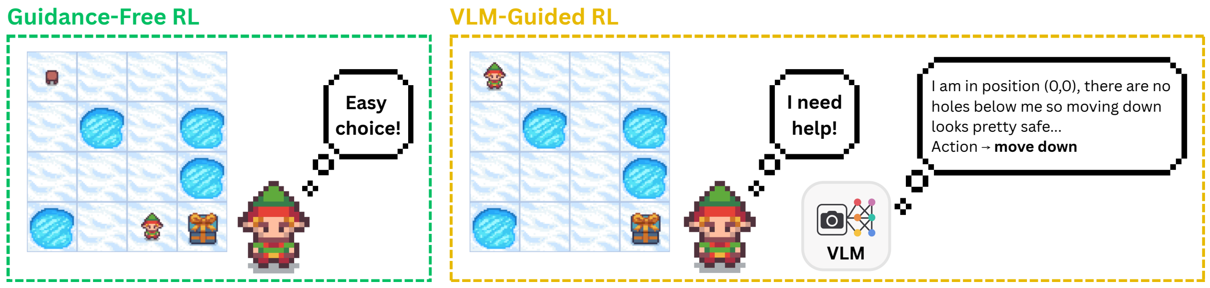

Guidance-Free RL
Easy choice!
VLM-Guided RL
I need help!
I am in position (0,0), there are no holes below me so moving down looks pretty safe...
Action → move down
VLM

Figure 1: Dual architecture for RL: the agent is free to act in situations where it displays high confidence, but requests guidance from a VLM when it is uncertain, following the suggested action.

2. Related Work

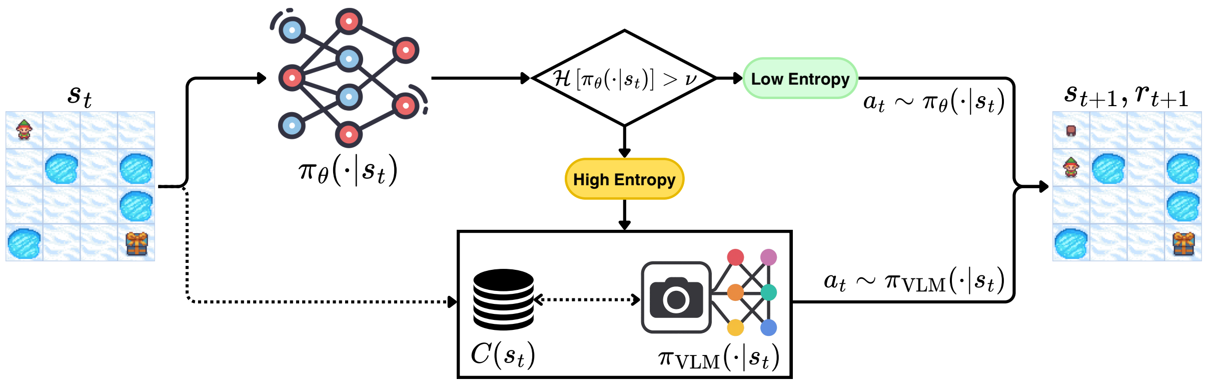
Foundation models have been integrated with RL in various ways to improve generalization, sample efficiency, and enable instruction-following abilities [24, 25].

One instance of this is reward shaping: LFMs can generate reward signals directly from textual instructions or from learned preferences [26, 27], or indirectly by embedding alignment between states and goals [28, 29]. Other works have proposed generating executable code as reward functions [30, 31] or training dedicated reward models through supervised or pretraining strategies [32, 33].

Another active line of research uses foundation models as policy priors or action generators. Pre-trained LFMs can directly output actions based on linguistic or perceptual inputs [34, 19], or produce code that defines the agent's behavior [35]. Vision-Language-Action (VLA) models have also been trained end-to-end to map observations and goals to low-level actions [36, 37, 38, 21]. RL with Foundation Priors (RLFP) unifies these approaches, showing how foundation models can be used jointly for reward modeling and policy learning [39].

Recent approaches like GLAM [40] and TWOSOME [41] fine-tune LLMs directly in interactive environments using online RL, aligning the model's prior knowledge with embodied or symbolic domains. Bonetta et al. [42] demonstrate that integrating a VLM as a policy into PPO training can significantly improve sample efficiency and performance compared to training from scratch. Similarly, LM4TEACH [43] distill LLM knowledge into a student policy based on the LLM's logits for each action. Zhai et al. [20] show that large VLMs, when fine-tuned with chain-of-thought prompting, outperform closed-source models on complex multi-step decision tasks. GTR [44] further improves this by supervising intermediate thoughts to prevent "thought collapse," ensuring the VLM's reasoning process remains consistent throughout learning.

However, most of these approaches rely on foundation models to drive decision-making at every step. In contrast, few methods investigate hybrid strategies where the foundation model plays a supportive role, guiding the policy only when needed. ULTRA [45] identifies critical states from collected experience offline and proposes improvements only for those. DSADF [46], in contrast, uses a hierarchical pipeline: a VLM plans subgoals, model-free controllers execute them, and the VLM intervenes if the controller fails. Han et al. [47] propose a similar dual-level system, using a VLA model to support decision-making. Most similar to our work, RCMP [48] also introduce a framework where guidance from a pre-trained expert is provided based on uncertainty, based on previous work in the multi-agent setting [49]. Compared to these, our method provides an online assistance mechanism: a lightweight CNN policy governs behavior, but when it expresses uncertainty, a VLM proposes an action which is treated as if the small model had chosen it. This setup enables faster training by leveraging VLM guidance selectively, while avoiding the high computational cost and scalability issues of always-on foundation model policies or hierarchical frameworks, as well as avoiding the need for a domain-specific expert to provide guidance, which might not always be available.



s_t
$\pi_\theta(\cdot|s_t)$
$\mathcal{H}\left[\pi_\theta(\cdot|s_t)\right] > \nu$
Low Entropy
High Entropy
$a_t \sim \pi_\theta(\cdot|s_t)$
$a_t \sim \pi_{\mathrm{VLM}}(\cdot|s_t)$
$C(s_t)$
$\pi_{\mathrm{VLM}}(\cdot|s_t)$
s_{t+1}, r_{t+1}

Figure 2: Diagram of the proposed framework. During the RL training loop, the agent samples actions from the learned policy $\pi_\theta(\cdot|s_t)$. If the policy exhibits high uncertainty, measured by the normalized entropy $\mathcal{H}[\pi_\theta(\cdot|s_t)] > \nu$ (and a stochastic toggle $u \sim \mathcal{U}(0,1)$ satisfies $u < \kappa$, not represented for simplicity), the agent defers to a VLM for guidance. In this case, an action $a_t \sim \pi_{\mathrm{VLM}}(\cdot|s_t)$ is either obtained from the cache $C(s_t)$, if available, or generated via prompting and stored as $C(s_t)$ for future use. The selected action a_t is then executed in the environment, observing the next state s_{t+1} and reward r_{t+1}.

3. Method

3.1. Problem Setup

Following standard RL conventions, we formulate the problem as a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{R}\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\mathcal{T}\colon \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition function and $\gamma \in [0,1]$ is the discount factor [1]. We focus on deterministic, fully observable environments with a discrete action space $\mathcal{A}$ in order to isolate the effect of our method from other sources of variability. In principle, our work can extend to stochastic, partially observable and continuous settings with minor changes.

In this setting, the agent aims to learn a policy $\pi\colon \mathcal{S} \to \Delta(\mathcal{A})$[1] which maximizes the discounted cumulative expected reward $\mathbb{E}_{a_t \sim \pi}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \mid s_0 \sim S_0\right]$ where S_0 is the initial state distribution. We use $\pi_\theta(a_t|s_t)$ to denote the probability of an action a_t being sampled from a learned policy $\pi_\theta(\cdot|s_t)$ parametrized by θ and conditioned on the current state s_t.

3.2. Vision-Language Model Guidance

We introduce a RL framework that augments a model-free policy with selective guidance from a pre-trained VLM. The key idea is to allow the RL agent to act autonomously in states where π_θ exhibits high confidence, while deferring to the VLM in low-confidence states where uncertainty about the optimal action is high. Specifically, we choose the *entropy* $\mathcal{H}\left[\pi_\theta(\cdot|s_t)\right] = -\sum_{a_t \in \mathcal{A}} \pi_\theta(a_t|s_t) \log \pi_\theta(a_t|s_t)$ as the measure of uncertainty of the policy (where a higher entropy implies a flatter distribution with more indecision between actions). With this, we define an *entropy threshold* hyperparameter ν, above which the VLM will be asked to provide guidance. Furthermore, we normalize $\mathcal{H}\left[\pi_\theta(\cdot|s_t)\right]$ in a range between $[0,1]$ by dividing it by its theoretical maximum $\log|\mathcal{A}|$, in order to set ν more easily.

When the normalized entropy exceeds ν, the VLM provides guidance for the agent by generating the action to take, acting as a *guidance policy* $\pi_{\mathrm{VLM}}(\cdot|s_t)$[2]. The model is prompted with an image of the current state s_t, a description of the available actions and goal, and instructed to reason with zero-shot Chain of Thought (CoT) prompting [50, 51] to generate an action $a_t \sim \pi_{\mathrm{VLM}}(\cdot|s_t)$. This replaces the sampling process in the training loop: a_t is sampled from $\pi_{\mathrm{VLM}}(\cdot|s_t)$ instead of $\pi_\theta(\cdot|s_t)$.

Crucially, VLMs are known to hallucinate [52] and may offer inaccurate or misleading guidance, particularly in domains that diverge from their pre-training distribution (mostly composed of realistic

[1]$\Delta(\mathcal{A})$ represents any probability distribution over $\mathcal{A}$, formally $\Delta(\mathcal{A}) = \{p\colon \mathcal{A} \to [0,1] \mid \sum_{a \in \mathcal{A}} p(a) = 1\}$
[2]Slightly abusing the notation, as the VLM only generates the best action, rather than a distribution over $\mathcal{A}$.

Algorithm 1 Selective Policy Guidance with VLM Supervision

```
Require: State $s_t$, policy $\pi_\theta(\cdot|s_t)$, guidance policy $\pi_{\text{VLM}}(\cdot|s_t)$
Require: Cache $C$, entropy threshold $\nu \in [0,1]$, guidance probability $\kappa \in [0,1]$
 1: Compute normalized entropy $H \leftarrow \mathcal{H}[\pi_\theta(\cdot|s_t)]$
 2: Sample $u \sim \mathcal{U}(0,1)$
 3: if $H > \nu$ and $u < \kappa$ then
 4:     if $s_t \in C$ then
 5:         $a_t \leftarrow C(s_t)$
 6:     else
 7:         $a_t \sim \pi_{\text{VLM}}(\cdot|s_t)$
 8:         $C(s_t) \leftarrow a_t$
 9:     end if
10: else
11:     $a_t \sim \pi_\theta(\cdot|s_t)$
12: end if
13: return $a_t$
```

images), such as synthetic environments or pixel art. However, in our framework the RL agent does not blindly imitate the VLM. Instead, it treats VLM-suggested actions as exploratory signals: if following such an action leads to poor rewards, the policy is updated through standard RL mechanisms (e.g., policy gradient or value function updates) to reduce the likelihood of selecting that action in similar states. This enables the agent to recover from suboptimal guidance and progressively improve its behavior.

Integrating a VLM into the learning process significantly increases the computational budget[3] required to train the agent, reducing its scalability to more complex tasks which require more training steps. Even if the VLM is prompted only under high-entropy conditions, the method still incurs in substantial overhead, as it will always be used at the early stages of training (since π_θ initially has high entropy across most states). To mitigate this, we introduce a caching mechanism $C : \mathcal{S} \to \mathcal{A}$ that stores the VLM's responses, allowing the agent to reuse previous guidance when revisiting the same state.

However, this in turn will force the agent to always follow the same strategy in cached states, restricting the potential for exploration and potentially resulting in sub-optimal behavior. To address this issue, we formulate a new hyperparameter $\kappa \in [0,1]$ which toggles when VLM guidance is invoked, similar to ϵ in an ϵ-greedy policy (i.e., VLM guidance will be used if $\mathcal{H}[\pi_\theta(\cdot|s)] > \nu \land u \sim \mathcal{U}(0,1) < \kappa$, where $\mathcal{U}$ is the uniform distribution). The final action selection process for our method is formalized in Algorithm 1 and summarized in Figure 2.

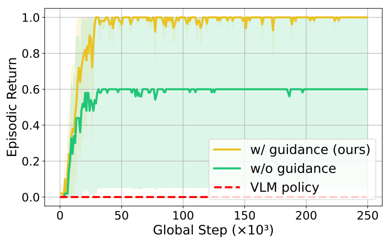
4. Evaluation

4.1. Experimental Setup

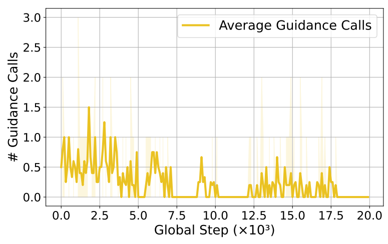
We now describe the experimental setup used to evaluate our method, including the underlying RL algorithm, environment, and evaluation metrics. Detailed hyperparameters are reported in Appendix B.

RL Algorithm. As our method only affects the action sampling process, it can be integrated into any model-free RL algorithm with the aim of improving its sample efficiency and/or stability. For this work, we choose the popular Proximal Policy Optimization (PPO) [9] method as a backbone. We follow the implementation of PPO provided by CleanRL [53], a high-quality codebase offering reproducible and well-tested RL baselines. The agent uses a Convolutional Neural Network (CNN) for both the actor and critic networks, enabling it to process visual state inputs directly. The action selection logic is modified by inserting our selective guidance policy in place of the default sampling routine, while the rest of

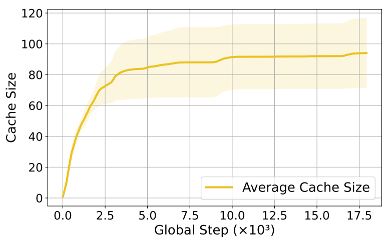
[3]For example, Zhai et al. [20], which use a VLM at every step, report a budget of 30 hours for 15k training steps.



(a) `FrozenLake` learning curves.



(b) VLM guidance calls over time.



(c) Running average of cache size.

Figure 3: Learning performance and internal dynamics of our selective guidance strategy (mean and standard deviation across 5 random seeds). For clarity of visualization, we omit the full 500k global steps: training already converges by 250k steps, and plots (b) and (c) stabilize before 20k steps. Shaded areas represent standard deviation. **(a)** Our agent shows more stable convergence compared to PPO without guidance, while the VLM policy alone fails completely. **(b)** Guidance is mostly used early during training. **(c)** The cache grows rapidly and then saturates, enabling efficient reuse of actions.

the algorithm remains unchanged. Our VLM of choice is the Gemma3 model [18] in its 4B parameters variant which provides a convenient tradeoff between speed and performance. The prompt used for VLM guidance is reported in Appendix A. All experiments use a training budget of 500k environment steps. We performed each experiment on a single NVIDIA A100 64GB GPUs with both the RL agent and the VLM loaded on the same device. We employ the vLLM library [54] to accelerate VLM inference.

Environment. We evaluate the method on a deterministic version of the `FrozenLake` grid-world environment from Gymnasium [23]. This task requires the agent to navigate a grid (using the cardinal directions as actions) and avoid holes in order to reach the goal, represented as a gift box (as seen in Figure 2). Notably, the agent only receives a positive reward of 1 when reaching the gift box, and 0 in any other case (including falling into a hole), making the signal very sparse. To further challenge the agent, we choose the bigger "8x8" grid map, which is kept fixed across episodes. A state s_t is given to both the PPO agent and the VLM as an image.

Evaluation Metrics. We assess performance using multiple metrics. First, we measure the agent's average episodic return over 10 evaluation episodes, reported across 5 random seeds. In addition, we visualize evaluation curves by plotting this average episodic return every 5000 training steps. To assess the VLM usefulness and cache usage, we track the number of *guidance calls* (i.e., how many times the VLM was queried) and the *cache dimension* over training time-steps.

4.2. Results

Figure 3a shows the learning curves of PPO agents on the `FrozenLake` environment, comparing three different setups:

- Our method *w/ guidance*, which uses selective VLM guidance during training.
- A PPO baseline *w/o guidance*, trained with the same CNN backbone but without any VLM involvement.
- A *VLM policy* baseline, which queries the VLM directly at inference time to choose actions, without any RL (without caching).

Both learning-based agents show early convergence during training. However, the vanilla PPO baseline exhibits high variance across seeds: while some runs converge quickly due to lucky initial exploration, others remain stuck with near-zero returns throughout the entire training process, making the baseline less reliable and predictable.

The VLM policy baseline consistently fails to solve the environment. This demonstrates that querying Gemma3 directly, without any task-specific learning or adaptation, is insufficient for meaningful decision-making in this setting. The VLM does not solve the long-horizon task, but can instead be useful to guide individual short-term actions, with RL learning not to repeat incorrect guidance.

In contrast, our method consistently converges to optimal performance (average return ~ 1.0) across all seeds, with significantly lower variance. Although the average sample efficiency is only slightly better due to the task's simplicity, our method is markedly more stable. The VLM guidance helps the agent recover from unproductive early rollouts and reliably explore successful strategies.

To better understand the role of VLM guidance during training, Figure 3b reports the number of VLM queries issued over time. As expected, most calls occur during the early high-entropy phase of learning. Around the 18k step mark, the number of guidance calls drops to zero, indicating that the agent has learned to act confidently without external help. Figure 3c tracks the evolution of the cache used to store guidance responses, which grows rapidly during early training, peaking around 17.5k steps with around 120 unique states[4]. After that, most information needs are met through cache hits rather than new VLM queries, reducing the computational overhead of prompting.

Together, these results provide a proof-of-concept for the adaptive behavior enabled by our guidance method. VLM queries are used when uncertainty is high, then progressively replaced by internalized behavior and cached knowledge. The resulting overhead is minimal: training with guidance takes about 90 minutes, compared to $\sim$85 minutes without it. This small cost is outweighed by the substantial improvements in stability and robustness, especially in long-horizon environments with sparse rewards.

5. Conclusions

We introduced a RL framework that leverages selective guidance from a pre-trained VLM to support early-stage exploration. By monitoring the agent's own uncertainty and triggering VLM guidance only when necessary, our method enables more stable and consistent learning compared to a vanilla PPO baseline, while keeping the reliance on external supervision limited in both time and scope. Our experiments on the `FrozenLake` environment show that the proposed strategy is particularly effective during the early phases of training, when the policy entropy is high and exploration is most challenging, with the cache reducing the computational burden by promoting reuse of past VLM completions. Overall, our results suggest that selectively integrating VLM supervision, rather than relying on it continuously, can be a practical and efficient way to improve the learning dynamics in a model-free RL agent.

5.1. Limitations and Future Work

While these initial results are promising, further work is needed to assess the method's generality and effectiveness. In `FrozenLake`, successful PPO runs already achieved near-optimal performance quickly (as shown by the shaded area in Figure 3a), limiting observable gains. More complex environments, such as ones involving grounded language understanding [55], may better showcase the benefits of selective VLM guidance in enhancing exploration and learning speed.

Future work should also test robustness across RL algorithms and in more challenging settings, including continuous, stochastic, and partially observable environments. This will require improving the measure of uncertainty (e.g., by using the standard deviation from an ensemble of value heads [48]) and improving the way guidance is incorporated into the algorithm, taking inspiration from imitation learning literature [56]. The process through which guidance is discarded (with the parameter κ) can also be improved, for instance by implementing some version of the UCB formula [1] or similar count-based methods [49].

Key directions include an ablation of guidance hyperparameters ν and κ to clarify their influence on learning, and improvements to the caching strategy (e.g., storing diverse completions) to avoid premature convergence. Our results also highlight the limits of small-scale VLMs like Gemma3, which was unable to reach the goal on its own (Figure 3a), even in a common environment which the model

[4]Note that, while the 8x8 grid-world is only comprised of 64 unique states, the direction the agent is facing is also represented visually (even if irrelevant in practice), bringing the theoretical maximum to $64 \cdot 4 = 256$ unique states.

has likely seen during pre-training. Since the VLM's output quality is central to our approach, scaling to better models will be important. Promising avenues are distilling knowledge from a larger VLM through supervised fine-tuning before training with RL, or incorporating the experience collected online by the RL algorithm, similar to RL4VLM [20], to improve the quality of guidance during training.

References

[1] R. S. Sutton, A. G. Barto, et al., Reinforcement learning: An introduction, MIT press Cambridge, 1998.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533.

[3] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of go with deep neural networks and tree search, Nature 529 (2016) 484–489.

[4] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al., Grandmaster level in starcraft ii using multi-agent reinforcement learning, Nature 575 (2019) 350–354.

[5] D. Hafner, J. Pasukonis, J. Ba, T. Lillicrap, Mastering diverse control tasks through world models, Nature 640 (2025) 647–653.

[6] S. Levine, C. Finn, T. Darrell, P. Abbeel, End-to-end training of deep visuomotor policies, Journal of Machine Learning Research 17 (2016) 1–40.

[7] S. Gu, E. Holly, T. Lillicrap, S. Levine, Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates, in: 2017 IEEE international conference on robotics and automation (ICRA), IEEE, 2017, pp. 3389–3396.

[8] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, et al., Solving rubik's cube with a robot hand, arXiv preprint arXiv:1910.07113 (2019).

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).

[10] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in: International conference on machine learning, Pmlr, 2018, pp. 1861–1870.

[11] A. Ecoffet, J. Huizinga, J. Lehman, K. O. Stanley, J. Clune, First return, then explore, Nature 590 (2021) 580–586.

[12] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al., On the opportunities and risks of foundation models, arXiv preprint arXiv:2108.07258 (2021).

[13] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).

[14] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al., The llama 3 herd of models, arXiv preprint arXiv:2407.21783 (2024).

[15] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al., Qwen3 technical report, arXiv preprint arXiv:2505.09388 (2025).

[16] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, P. Zhang, Y. Li, Z. Liu, et al., Llava-onevision: Easy visual task transfer, arXiv preprint arXiv:2408.03326 (2024).

[17] 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).

[18] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière, et al., Gemma 3 technical report, arXiv preprint arXiv:2503.19786 (2025).

[19] M. Ahn, B. Ichter, A. Brohan, Y. Chebotar, C. Finn, K. Hausman, et al., Do as i can, not as i say: Grounding language in robotic affordances, in: 6th Annual Conference on Robot Learning, 2022.

[20] Y. Zhai, H. Bai, Z. Lin, J. Pan, S. Tong, Y. Zhou, A. Suhr, S. Xie, Y. LeCun, Y. Ma, S. Levine, Fine-tuning large vision-language models as decision-making agents via reinforcement learning, in: The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024.

[21] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl, et al., Gemini robotics: Bringing ai into the physical world, arXiv preprint arXiv:2503.20020 (2025).

[22] D. Kahneman, Thinking, Fast and Slow, Allen Lane, 2011.

[23] M. Towers, A. Kwiatkowski, J. Terry, J. U. Balis, G. De Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG, et al., Gymnasium: A standard interface for reinforcement learning environments, arXiv preprint arXiv:2407.17032 (2024).

[24] Y. Cao, H. Zhao, Y. Cheng, T. Shu, Y. Chen, G. Liu, G. Liang, J. Zhao, J. Yan, Y. Li, Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods, IEEE Transactions on Neural Networks and Learning Systems (2024).

[25] S. Schoepp, M. Jafaripour, Y. Cao, T. Yang, F. Abdollahi, S. Golestan, Z. Sufiyan, O. R. Zaiane, M. E. Taylor, The evolving landscape of llm-and vlm-integrated reinforcement learning, arXiv preprint arXiv:2502.15214 (2025).

[26] M. Kwon, S. M. Xie, K. Bullard, D. Sadigh, Reward design with language models, in: The Eleventh International Conference on Learning Representations, 2023.

[27] Y. Du, O. Watkins, Z. Wang, C. Colas, T. Darrell, P. Abbeel, A. Gupta, J. Andreas, Guiding pretraining in reinforcement learning with large language models, in: International Conference on Machine Learning, PMLR, 2023, pp. 8657–8677.

[28] M. Klissarov, P. D'Oro, S. Sodhani, R. Raileanu, P.-L. Bacon, P. Vincent, A. Zhang, M. Henaff, Motif: Intrinsic motivation from artificial intelligence feedback, in: NeurIPS 2023 Foundation Models for Decision Making Workshop, 2023.

[29] J. Rocamonde, V. Montesinos, E. Nava, E. Perez, D. Lindner, Vision-language models are zero-shot reward models for reinforcement learning, in: The Twelfth International Conference on Learning Representations, 2024.

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

[31] D. Venuto, S. N. Islam, M. Klissarov, D. Precup, S. Yang, A. Anand, Code as reward: empowering reinforcement learning with vlms, in: Proceedings of the 41st International Conference on Machine Learning, ICML'24, JMLR.org, 2024.

[32] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, A. Zhang, VIP: Towards universal visual reward and representation via value-implicit pre-training, in: NeurIPS 2022 Foundation Models for Decision Making Workshop, 2022.

[33] Y. J. Ma, V. Kumar, A. Zhang, O. Bastani, D. Jayaraman, LIV: Language-image representations and rewards for robotic control, in: Workshop on Reincarnating Reinforcement Learning at ICLR 2023, 2023.

[34] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, P. Sermanet, T. Jackson, N. Brown, L. Luu, S. Levine, K. Hausman, brian ichter, Inner monologue: Embodied reasoning through planning with language models, in: 6th Annual Conference on Robot Learning, 2022.

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

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

[37] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess,

A. Dubey, C. Finn, et al., Rt-2: Vision-language-action models transfer web knowledge to robotic control, arXiv preprint arXiv:2307.15818 (2023).

[38] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al., Openvla: An open-source vision-language-action model, arXiv preprint arXiv:2406.09246 (2024).

[39] W. Ye, Y. Zhang, H. Weng, X. Gu, S. Wang, T. Zhang, M. Wang, P. Abbeel, Y. Gao, Reinforcement learning with foundation priors: Let embodied agent efficiently learn on its own, in: 8th Annual Conference on Robot Learning, 2024.

[40] T. Carta, C. Romac, T. Wolf, S. Lamprier, O. Sigaud, P.-Y. Oudeyer, Grounding large language models in interactive environments with online reinforcement learning, in: Proceedings of the 40th International Conference on Machine Learning, ICML'23, JMLR.org, 2023.

[41] W. Tan, W. Zhang, S. Liu, L. Zheng, X. Wang, B. An, True knowledge comes from practice: Aligning large language models with embodied environments via reinforcement learning, in: The Twelfth International Conference on Learning Representations, 2024.

[42] G. Bonetta, D. Zago, R. Cancelliere, M. Polato, B. Magnini, Vision language models as policy learners in reinforcement learning environments, in: ESANN, 2024.

[43] Z. Zhou, B. Hu, C. Zhao, P. Zhang, B. Liu, Large language model as a policy teacher for training reinforcement learning agents, IJCAI '24, 2024.

[44] T. Wei, Y. Yang, J. Xing, Y. Shi, Z. Lu, D. Ye, Gtr: Guided thought reinforcement prevents thought collapse in rl-based vlm agent training, arXiv preprint arXiv:2503.08525 (2025).

[45] H. Tan, H. Yan, Y. Yang, Llm-guided reinforcement learning: Addressing training bottlenecks through policy modulation, arXiv preprint arXiv:2505.20671 (2025).

[46] A. Z. Dou, D. Cui, J. Yan, W. Wang, B. Chen, H. Wang, Z. Xie, S. Zhang, Dsadf: Thinking fast and slow for decision making, arXiv preprint arXiv:2505.08189 (2025).

[47] B. Han, J. Kim, J. Jang, A dual process vla: Efficient robotic manipulation leveraging vlm, arXiv preprint arXiv:2410.15549 (2024).

[48] F. L. Da Silva, P. Hernandez-Leal, B. Kartal, M. E. Taylor, Uncertainty-aware action advising for deep reinforcement learning agents, Proceedings of the AAAI Conference on Artificial Intelligence 34 (2020) 5792–5799.

[49] F. L. Da Silva, R. Glatt, A. H. R. Costa, Simultaneously learning and advising in multiagent reinforcement learning, in: Proceedings of the 16th conference on autonomous agents and multiagent systems, 2017, pp. 1100–1108.

[50] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain-of-thought prompting elicits reasoning in large language models, Advances in neural information processing systems 35 (2022) 24824–24837.

[51] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, Y. Iwasawa, Large language models are zero-shot reasoners, Advances in neural information processing systems 35 (2022) 22199–22213.

[52] H. Liu, W. Xue, Y. Chen, D. Chen, X. Zhao, K. Wang, L. Hou, R. Li, W. Peng, A survey on hallucination in large vision-language models, arXiv preprint arXiv:2402.00253 (2024).

[53] S. Huang, R. F. J. Dossa, C. Ye, J. Braga, D. Chakraborty, K. Mehta, J. G. Araújo, Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms, Journal of Machine Learning Research 23 (2022) 1–18.

[54] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, I. Stoica, Efficient memory management for large language model serving with pagedattention, in: Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles, 2023.

[55] D. Paglieri, B. Cupiał, S. Coward, U. Piterbarg, M. Wolczyk, A. Khan, E. Pignatelli, Ł. Kuciński, L. Pinto, R. Fergus, J. N. Foerster, J. Parker-Holder, T. Rocktäschel, BALROG: Benchmarking agentic LLM and VLM reasoning on games, in: The Thirteenth International Conference on Learning Representations, 2025.

[56] S. Ross, G. Gordon, D. Bagnell, A reduction of imitation learning and structured prediction to no-regret online learning, in: Proceedings of the fourteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

A. Prompts

VLM Guidance Prompt

```
<system>
You are a smart agent acting in the "FrozenLake" environment from the Gymnasium library. You will
receive observations from the environment and must decide which action to take next.
You should think about the answer step by step inside the <think> tag, and then provide the action inside
the <action> tag. All text outside the <think> and <action> tags will be ignored.
</system>

<user>
The goal is to navigate across a frozen grid to reach the goal without falling into holes.

## Observation space
You are presented with the image of the environment in the style of pixel art.
Each cell may be safe (white with snow), a hole (bright blue ice hole), the start (a stool) or the goal (a
present box). The agent is represented by an elf character.

## Action space
The action space includes 4 discrete actions:
– 0: Move left
– 1: Move down
– 2: Move right
– 3: Move up

Based on the current visual information, which action should the agent take next?

## Observation:
{image}
</user>
```

B. Experiment Hyperparameters

This Appendix reports the hyperparameters used for PPO training in our experiments on the `FrozenLake-v1` environment. We also report the configurations for the CNN policy and for the VLM.

Parameter	**Value**
Environment ID	`FrozenLake-v1`
Map size	8×8
Slippery	`false`
Observation type	RGB image
Image stack size	1
Number of parallel environments	4
Episode length (steps)	128
Total timesteps	500,000

Table 1
Environment configuration.

Parameter	Value
Learning rate	2.5e-4
Batch size	512
Minibatch size	128
Update epochs	4
Number of minibatches	4
Discount factor (γ)	0.99
GAE lambda	0.95
Entropy coefficient	0.01
Value function coefficient	0.5
Clip coefficient	0.2
Clip value loss	`true`
Normalize advantages	`true`
Anneal learning rate	`true`
Max gradient norm	0.5
Evaluation frequency	Every 1,000 steps
Evaluation episodes	10

Table 2
PPO hyperparameters.

Parameter	Value
Agent type	CNN-based
Convolutional channels	[16, 32, 32]
Kernel size	3
Stride	2
Padding	1
Conv head activation	ReLU
Residual activation	ReLU
Conv head output size	256
Actor standard deviation	0.01
Critic standard deviation	1.0

Table 3
CNN policy architecture details.

Parameter	Value
Model ID	`google/gemma-3-4b-it`
Entropy threshold (ν)	0.9
Kappa (κ)	0.1
Cache size	128

Table 4
Vision-Language Model (VLM) selective guidance configuration.