

COVR: Collaborative Optimization of VLMs and RL Agent for Visual-Based Control

Canming Xia^{1,2}, Peixi Peng^{3,2*}, Guang Tan^{1*}, Zhan Su³, Haoran Xu^{1,2},
Zhenxian Liu⁴, Luntong Li²

¹ School of Intelligent Systems Engineering, Shenzhen Campus of Sun Yat-sen University, China

² Peng Cheng Laboratory, China

³ School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, China

⁴ National Engineering Research Center of Visual Technology, School of Computer Science, Peking University, China
xiacm@mail2.sysu.edu.cn, pxpeng@pku.edu.cn, tanguang@mail.sysu.edu.cn

Abstract

Visual reinforcement learning (RL) suffers from poor sample efficiency due to high-dimensional observations in complex tasks. While existing works have shown that vision-language models (VLMs) can assist RL, they often focus on knowledge distillation from the VLM to RL, overlooking the potential of RL-generated interaction data to enhance the VLM. To address this, we propose COVR, a collaborative optimization framework that enables the mutual enhancement of the VLM and RL policies. Specifically, COVR fine-tunes the VLM with RL-generated data to enhance the semantic reasoning ability consistent with the target task, and uses the enhanced VLM to further guide policy learning via action priors. To improve fine-tuning efficiency, we introduce two key modules: (1) an Exploration-Driven Dynamic Filter module that preserves valuable exploration samples using adaptive thresholds based on the degree of exploration, and (2) a Return-Aware Adaptive Loss Weight module that improves the stability of training by quantifying the inconsistency of sampling actions via return signals of RL. We further design a progressive fine-tuning strategy to reduce resource consumption. Extensive experiments show that COVR achieves strong performance across various challenging visual control tasks.

Introduction

Visual reinforcement learning (RL) has emerged as a critical approach for intelligent systems to cope with complex tasks such as robotic control (Fu et al. 2024), autonomous driving (Zhang et al. 2020; Xu et al. 2024a), and game simulation (Liu, Peng, and Tian 2025; Laskin et al. 2020). A fundamental challenge in visual RL is the inefficiency of exploration under high-dimensional visual inputs, where invalid interactions and complex state spaces complicates policy learning. Recent advancements in large-scale models (LMs), including large language models and vision-language models (VLMs), have shown potential in guiding decision making (Shinn et al. 2023; Wang et al. 2024b; Pan et al. 2024; Dalal et al. 2024). Existing works have thus explored integrating LMs into RL to enhance learning performance (Lee et al. 2025; Zhou et al. 2024a; Xu et al. 2025b).

*Corresponding authors.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

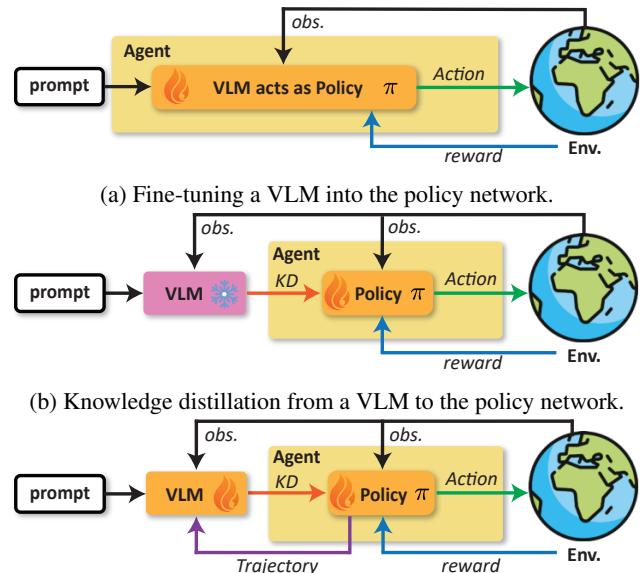


Figure 1: Comparison of VLM-adapted and VLM-assisted policy learners. It iteratively enhances the capabilities of VLMs and RL agents.

Figure 1: Comparison of VLM-adapted and VLM-assisted policy learners.

Existing LM-assisted methods generally fall in two categories. One is to treat the LM as a fixed feature extractor integrated into the policy module (Paischer et al. 2023; Fu et al. 2024; Guo et al. 2025), or directly fine-tune the LM as the policy network (Zhai et al. 2024; Tan et al. 2024; Wei et al. 2025), as depicted in Fig. 1(a). These approaches may suffer from high computational overhead and deployment costs for online inference. Another line of work aims to transfer the intrinsic knowledge of a frozen LM to the policy network via knowledge distillation (KD) (Lee et al. 2025; Zhou et al. 2024a; Xu et al. 2025a; Chen et al. 2024), see an illustration in Fig. 1(b). These approaches can improve policy learning using priors and allow faster inference, but may yield suboptimal performance when the VLM is trained with limited domain-specific data. This is due to the

frozen VLM that potentially propagates inaccurate reasoning results, leading to negative impact on policy learning.

We argue that VLMs and visual RL agents possess highly complementary strengths, and thus, a principled integration of the two is both natural and beneficial: (1) **VLM improves RL via Prior Guidance**. VLMs can extract rich semantic priors from images and provide direct, task-specific policy guidance to accelerate the training of visual RL models. In contrast to indirect methods that rely on reward shaping (Xie et al. 2024; Wang et al. 2024a) or task decomposition (Dalal et al. 2024; Shukla et al. 2024), this approach enables more stable and efficient policy learning by leveraging explicit semantic reasoning from VLMs. (2) **RL enhances VLM via Specialized Experience**. RL is able to discover high-quality state-action pairs in specific scenarios, despite the challenges of coping with high-dimensional state space in real-world tasks. By exploiting these RL-derived trajectories to fine-tune the VLM, we effectively encode domain-specific strategies into the VLM. This enables task-oriented prior generation that could generalize to unseen states.

Building on these insights, we propose a novel approach for VLM-assisted RL, implemented through a collaborative optimization framework called COVR, as illustrated in Fig. 1(c). It comprises two key components: (1) **Knowledge Enhancement of RL-Tuned VLM**: The VLM is fine-tuned using interaction trajectories collected from the RL agent. This process aligns the VLM’s ability with task-specific semantics, enhancing its decision relevance. (2) **VLM-Guided Policy Learning**: The fine-tuned VLM provides generalizable action priors that guide the RL policy. These priors accelerate learning by offering more informative gradients. To effectively realize COVR, we address two key challenges:

- How to select high-quality trajectory samples? We design an *Exploration-Driven Dynamic Filter* (EDDF) module, which receives state-action pairs generated during RL interactions, along with their associated return computed from accumulated episode rewards. The module dynamically determines the thresholds based on the agent’s exploration level to select high-quality samples;
- How to alleviate the impact of action inconsistency in RL? During training, RL may yield distinct high-reward actions for similar observations, resulting in inconsistent signals for the VLM and potentially hindering policy convergence. We introduce the *Return-Aware Adaptive Loss Weight* (RALW) module, which dynamically adjusts the loss weights according to return values. This mechanism enables the model to prioritize high-return samples while preserving its original capabilities on low-return examples.

Finally, we introduce an *Adaptive Progressive Fine-Tuning* strategy that gradually reduces the frequency of VLM updates as the RL policy converges, thereby improving overall training efficiency.

In summary, the contributions of this work are fourfold:

- A novel VLM-assisted RL method based on collaborative optimization, which enables visual RL policies to achieve improved performance even when the assisting VLM has limited capabilities.

- An EDDF module that dynamically selects trajectory samples based on the exploration degree of RL to enhance task-specific knowledge acquisition in VLMs.
- A RALW module that distinguishes inconsistent actions under similar observations by adaptively adjusting loss weights based on return signals, leading to more effective learning of the VLM.
- Extensive experiments on the CARLA (Dosovitskiy et al. 2017) and DMControl (Tassa et al. 2018) validating the superiority of our method.

Related Work

LM-assisted RL. Most methods that integrate LMs into RL can be categorized into three classes: (1) **LMs as agents**. In this approach, LMs serve as core components of the policy. It is further divided into parametric and non-parametric methods. Parametric methods fine-tune the LM to generate task-specific actions, enabling strong adaptation to downstream tasks (Zhai et al. 2024; Tan et al. 2024; Wei et al. 2025), while non-parametric methods utilize a frozen LM to extract rich semantic priors for decision-making (Zhou et al. 2024a; Shinn et al. 2023), with wide applications in robotic manipulation (Chen et al. 2024) and agent collaboration (Xu et al. 2024c). However, parametric methods face deployment challenges, while non-parametric ones struggle with long-term planning. Our method addresses these issues via knowledge distillation and enhances sequential planning in continuous RL tasks. (2) **LMs as planners**. The LM decomposes tasks into sub-goals, generating plans either upfront (Tang et al. 2023; Shukla et al. 2024) or incrementally (Lee et al. 2025; Zhou et al. 2024a). Yet, these rely heavily on the LM’s reasoning, which our framework mitigates through return-aware filtering. (3) **LMs as reward shapers**. In this setting, the LMs assist RL by modeling reward signals, either by generating executable reward functions (Xie et al. 2024) or by producing scalar reward estimates (Wang et al. 2024a, 2025). Despite their potential, these approaches often fail to capture real-world complexity and incur high trial-and-error costs. In contrast, our method provides direct policy guidance, significantly shortening the path from knowledge to effective action.

Training LMs with RL. RL has been extensively utilized to elicit and refine capabilities in LMs. Among these approaches, RL from human feedback (RLHF) trains reward models using human-labeled data before optimizing the policy (Team et al. 2024). Alternatively, methods incorporating human preference data shape the reward function to align model behavior with human expectations (Rafailov et al. 2023). In contrast, our method performs RL fine-tuning based on environmental rewards. Recent advances in process reward models (Lightman et al. 2023), search algorithms (Xin et al. 2024), and GRPO (Shao et al. 2024) have achieved notable progress in specialized tasks like mathematical reasoning. However, our focus is on sequential decision-making for goal-directed behaviors in interactive environments. By integrating visual input and visual-language reasoning, we have enhanced the performance of RL in multiple tasks. While recent work (Waite et al. 2025)

leverages RL to generate data for improving the VLM, it does not incorporate this knowledge back into the RL policy, resulting in high exploration costs. Conversely, our method is designed to enhance policy learning within RL by feeding knowledge back into the policy, thereby reducing exploration costs.

Preliminary

Process of Visual RL. The visual RL task can be modeled as a Markov Decision Process (MDP), represented by the tuple $(\mathcal{O}, \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$. In this formulation, $o_t \in \mathcal{O}$ represents the raw visual input observed at time step t , while $s_t \in \mathcal{S}$ denotes the corresponding state features extracted from o_t . The set \mathcal{A} defines the space of actions, \mathcal{R} is the reward function, and $\gamma \in [0, 1]$ serves as the discount factor. At each time step, the agent receives the visual observation o_t and generates an action $a_t \in \mathcal{A}$ according to the policy π . Upon executing the action, the environment returns a scalar reward $r_{t+1} \sim \mathcal{R}(s_t, a_t)$, the next visual observation o_{t+1} , and the game done flag.

Soft Actor Critic. Our method is built upon the Soft Actor-Critic (SAC) (Haarnoja et al. 2018a,b), which alternately optimizes a actor network $\pi_\theta(\cdot)$ and a critic network $Q_\phi(\cdot)$. SAC aims to maximize the expected long-term reward while promoting exploration through an entropy regularization term weighted by α :

$$\mathcal{L}_\pi = -\mathbb{E}_{a_t \sim \pi_\theta} [Q_\phi(s_t, a_t) - \alpha \log \pi_\theta(a_t | s_t)]. \quad (1)$$

The parameters of $Q_\phi(\cdot)$ are updated using the Bellman backup target, formulated as:

$$\mathcal{L}_Q = \mathbb{E}_{(s_t, a_t) \sim \mathcal{D}} [(Q_\phi(s_t, a_t) - (r_t + \gamma V(s_{t+1})))^2], \quad (2)$$

where \mathcal{D} denotes the experience replay buffer. The soft state value function $V(s_{t+1})$ is computed as:

$$V(s_{t+1}) = \mathbb{E}_{\tilde{a} \sim \pi_\theta} [\bar{Q}_\phi(s_{t+1}, \tilde{a}) - \alpha \log \pi_\theta(\tilde{a} | s_{t+1})], \quad (3)$$

where $\bar{Q}_\phi(\cdot)$ representing the target critic network, which is maintained as an exponential moving average of $Q_\phi(\cdot)$, and \tilde{a} is the next action sampled from $\pi_\theta(\cdot)$.

Method

Overview

In the standard visual RL framework, we introduce VLM-based reasoning to enhance policy learning. Specifically, the VLM processes the current visual observation and task-specific prompts to infer action semantics, which are then mapped to continuous action $a_{v,t}$ via a string-to-float parsing function. We denote the native action of $\pi_\theta(\cdot)$ as $a_{r,t}$. Notably, only $a_{r,t}$ interacts with the environment during training, while $a_{v,t}$ serves as an auxiliary supervisory signal for policy refinement. During testing, our method relies exclusively on the actor network of the visual RL system for decision-making, ensuring that real-time performance requirements are met.

The framework of COVR is shown in Fig. 2. COVR consists of two main components: (1) VLM-Guided RL. We

adopt the standard SAC training procedure and use $a_{v,t}$ as a regularization constraint on the learning of $\pi_\theta(\cdot)$, enabling improved exploration and mitigation of suboptimal convergence. (2) RL-tuned VLM. In this component, the VLM is iteratively refined using RL-generated trajectories to enhance their scene understanding.

First, due to the instability of RL training, many noisy or low-quality samples are generated. Therefore, we develop an EDDF module, which dynamically adjusts the threshold for sample selection based on the exploration level in RL, enabling more effective fine-tuning of the VLM.

Second, the stochastic nature of exploration often leads to inconsistent actions under visually similar observations. For example, choosing between ‘‘accelerate forward’’ and ‘‘decelerate and turn right’’ on a straight road. Training the VLM directly on such noisy and inconsistent data may mislead supervised fine-tuning. To mitigate this, we introduce a RALW module, which assigns trajectory-specific loss weights proportional to their returns, prioritizing high-return trajectories and mitigating the interference from inconsistent actions under similar observations.

Policy Guidance

In VLM-Guided RL, the iterative improved the VLM provides policy-level guidance to the RL agent by leveraging its generalization capability and better domain-specific knowledge, leading to a more globally optimal policy. To implement this guidance, we follow the previous work (Zhou et al. 2024a), which introduces a regularization term on \mathcal{L}_π that aligns the actions $a_{v,t}$ predicted by the VLM with those from $\pi_\theta(\cdot)$. Formally, the final loss of $\pi_\theta(\cdot)$ is defined as:

$$\mathcal{L}_{\tilde{\pi}} = \mathcal{L}_\pi + \lambda \|a_{v,t} - a_{r,t}\|_2^2, \quad (4)$$

where λ is the weight.

Exploration-Driven Dynamic Filter

To identify high-quality samples for fine-tuning the VLM, we design the Exploration-Driven Dynamic Filtering (EDDF) module within the collaborative optimization framework. Given that immediate rewards r_t reflect short-term gains and Q -value estimates may be biased, we adopt the trajectory-based cumulative return as the core metric for policy evaluation. Formally, the return is defined as $g_t = \sum_{k=t}^T \gamma^{k-t} r_k$, where T is the terminal step of the trajectory. g_t provides a more comprehensive assessment of policy performance. Specifically, EDDF operates in three stages:

(1) **Data Storage.** We maintain a dedicated buffer \mathcal{D}_f to store trajectory data for fine-tuning, including observations o_t , actions $a_{r,t}$, and returns g_t . Formally, $\mathcal{D}_f = \{(o_i, a_{r,i}, g_i)\}_{i=1}^N$, where N is the length of \mathcal{D}_f .

(2) **Data Transformation.** Inspired by data normalization techniques in machine learning, we apply a Z-score transformation (Cheadle et al. 2003) to the return values in \mathcal{D}_f : $\mathcal{G}_z = \text{Z-score}(\{g_i\}_{i=1}^N)$. This standardizes the distribution, enhances sensitivity to outliers, and ensures robustness across training phases.

(3) **Dynamic Filtering.** The core idea is to adaptively adjust the filtering threshold τ based on the agent’s

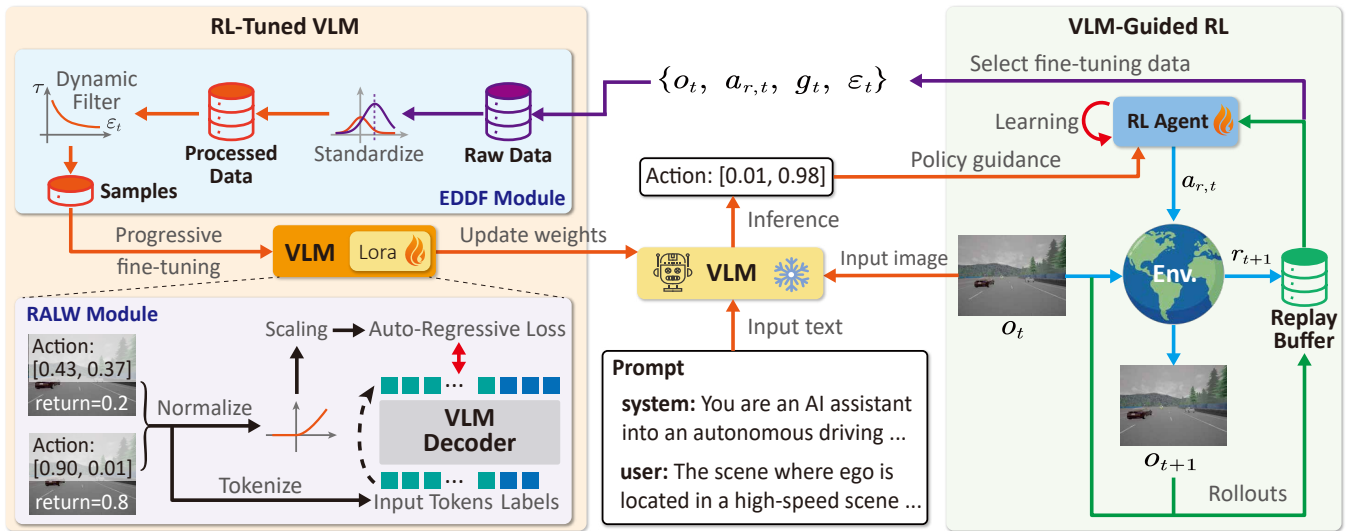


Figure 2: Collaborative optimization framework of COVR. It consists of two main components: (1) VLM-Guided RL. During this stage, the agent learns the policy under the guidance of actions inferred by the VLM. (2) RL-tuned VLM. This part comprises two essential modules: EDDF and RALW. Through the interaction of these two modules, the expertise of the VLM in specific domains is improved, which in turn benefits RL.

exploration-exploitation behavior. During early training, when policy entropy ε_t of visual RL is high, we retain more potentially valuable samples with a lower threshold, even if they yield low returns due to stochasticity. As training progresses and entropy decreases, the threshold becomes stricter to prioritize high-return trajectories as ε_t decreases and tends to stabilize. The threshold τ is computed as:

$$\tau = \text{Median}(\mathcal{G}_z) + \text{Sigmoid}(\varepsilon_t) \cdot \text{IQR}(\mathcal{G}_z), \quad (5)$$

where $\text{Median}(\cdot)$ computes the median value of \mathcal{G}_z , $\text{IQR}(\cdot)$ measures the interquartile range (25th–75th percentile spread) (Vinutha, Poornima, and Sagar 2018), and the Sigmoid function $\text{Sigmoid}(\cdot)$ maps entropy ε_t to a normalized scaling factor. This ensures τ dynamically responds to the agent’s learning dynamics. When needing fine-tuning, we screen out the samples with g_i exceeding the threshold τ in \mathcal{D}_f and retrieve the final corresponding $(o_i, a_{r,i}, g_i)$ pairs based on the index of g_i for fine-tuning the VLM.

Return-Aware Adaptive Loss Weight

To address the issue of action inconsistency, we propose the Return-Aware Adaptive Loss Weighting (RALW) module that dynamically adjusts the influence of each sample during fine-tuning based on its cumulative reward. We first select high-quality samples via the EDDF module and normalize returns in the selected samples to the range $[-1, 1]$. Samples with negative returns ($g_t < 0$) are assigned zero weight to suppress their influence, while high-return samples ($g_t > 0$) receive higher weights to emphasize favorable behaviors. This design preserves the pre-trained VLM’s basic capabilities while enabling performance improvement through high return-guided fine-tuning.

Next, we construct a training batch $K = \{(x_b, y_b, \bar{g}_b)\}_{b=1}^B$, where x_b is the input tokens (tokenized from the input

Algorithm 1: Relevant training pipeline of COVR

- 1: Initialize the SAC algorithm, the train steps T_{train} , the vision-language model (VLM), corresponding prompt, Replay buffer \mathcal{D} , and \mathcal{D}_f ;
- 2: Set $c=0$, $f_t=0$, and $\psi_{c+1} = \psi_c + \psi_c * c$;
- 3: **for** Every step t in T_{train} **do**
- 4: **if** Game is done **then**
- 5: Reset environment;
- 6: **end if**
- 7: Execute the VLM to reason $a_{v,t}$ based on the current observation image and prompt;
- 8: Sample $a_{r,t} \sim \pi_{\theta}(\cdot)$ and execute it to get r_{t+1} , o_{t+1} , and terminal state d_t ;
- 9: Store transition $(o_t, a_{r,t}, r_{t+1}, o_{t+1}, d_t, a_{v,t})$ in \mathcal{D} ;
- 10: Calculate g_t and store $(o_t, a_{r,t}, g_t)$ in \mathcal{D}_f ;
- 11: **if** Enough samples in \mathcal{D} **then**
- 12: Sample batch data from \mathcal{D} to Update SAC.
- 13: **end if**
- 14: **if** $f_t \% \psi_{c+1}$ is 0 **then**
- 15: Select data from \mathcal{D}_f using the EDDF module to fine-tuning the VLM with the RALW module;
- 16: Reset $f_t=0$ and clear \mathcal{D}_f ;
- 17: $c \leftarrow c + 1$;
- 18: $\psi_{c+1} \leftarrow \psi_c + \psi_c * c$;
- 19: **end if**
- 20: $f_t \leftarrow f_t + 1$;
- 21: **end for**

prompt and o_t), y_b is the label tokens (tokenized from $a_{r,t}$), $\bar{g}_b \in [-1, 1]$ is the normalized return, and B is the batch size. We incorporate external return signals into the fine-tuning process by formulating the objective as a return-weighted

auto-regressive loss. Let T be the token length, and N_v represent a total number of valid token in the batch. Formally, the return-weighted auto-regressive loss can be defined as:

$$\mathcal{L}_{\text{RALW}} = \frac{1}{N_v} \sum_{b=1}^B w_b \sum_{t=1}^T -\log p(y_{b,t} | \mathbf{x}_{b,<t}), \quad (6)$$

where $\log p(y_{b,t} | x_{b,<t})$ refers to the model forecast $y_{b,t}$ of conditional probability, and $w_b = \max(\bar{g}_b, 0)$ denotes the non-negative weight. This formula ensures that the model prioritizes learning label tokens with higher returns, thereby reducing the interference that action bias may introduce under similar observations. Notably, the auto-regressive loss utilized is the negative log-likelihood (NLL) loss with label smoothing (Guo et al. 2024), where a label smoothing regularization will be incorporated into the auto-regressive loss to alleviate overfitting.

Progressive Fine-tuning

To improve training efficiency and reduce computational overhead, we adopt a progressive fine-tuning strategy, where both the fine-tuning interval and the size of \mathcal{G}_z gradually increase during training. This is motivated by the observation that frequent updates are necessary in the early stages when the policy is unstable, while less frequent updates suffice as the policy gradually converges. Specifically, we define a step size ψ_c that increases linearly with the number of fine-tuning iterations c , that is, $\psi_{c+1} = \psi_c + \psi_c * c$, where $c=0, 1, 2, \dots$. Notably, \mathcal{D}_f is cleared after each fine-tuning and a fine-tuning update is triggered only when the accumulated training steps reach ψ . In addition, we utilize Lora fine-tuning technology (Hu et al. 2022) to further minimize the resource consumption. For clarity, Alg. 1 provides the relevant training pipeline of COVR.

Experiments

Experimental Settings

Environments. To evaluate the effectiveness of COVR in complex visual tasks, we conducted tests on two widely used RL benchmarks: (1) the CARLA simulator (Dosovitskiy et al. 2017) for autonomous driving, and (2) the DM-Control (Tassa et al. 2018) for robotic control. The CARLA simulator offers highly realistic visual inputs, which include a wide range of task-irrelevant details. This makes it an ideal benchmark for assessing algorithm performance in complex and realistic driving scenarios. For CARLA, we evaluate COVR on two challenging scenarios: (1) Highway (#HW), with up to 10 random vehicles placed in front of the agent, and (2) Ghost Pedestrian (#GP), where pedestrians suddenly cross the road behind static vehicles at three predefined locations. The agent must drive as far as possible within 1,000 time steps while avoiding collisions. For DMControl, we report experimental results on six commonly used tasks and other hard tasks, demonstrating the strong capability of COVR across different domains.

Implementation Details. For the VLM, we employed Qwen2.5-VL-3B (Bai et al. 2025) for prior knowledge reasoning tasks in the baseline experiments. Following the prior

Type	Methods	ER \uparrow	DD \uparrow
Vanilla visual RL	SAC	69 \pm 46	91 \pm 56
	DeepMDP	155 \pm 84	167 \pm 89
	CURL	135 \pm 63	152 \pm 70
	DrQ	115 \pm 51	127 \pm 55
	SPR	84 \pm 53	100 \pm 61
	MLR	106 \pm 69	130 \pm 80
	PER	159 \pm 68	175 \pm 73
	ERE	117 \pm 89	132 \pm 95
	ResAct	227 \pm 36	236 \pm 40
Only the VLM	VBE	-11 \pm 5	11 \pm 4
VLM-assisted visual RL	DPL	113 \pm 63	124 \pm 67
	APL	146 \pm 74	155 \pm 76
	VPF	91 \pm 24	99 \pm 24
	DGC	208 \pm 13	234 \pm 15
	COVR (Ours)	248 \pm 81	259 \pm 85

Table 1: Performance comparison in the #HW scenario. The best results for each metric are denoted by \blacksquare .

work (Xu et al. 2025b), experiments are conducted using three seeds and the mean and standard deviation of metrics are reported. For the parameters, we set $\lambda=2.0$ and initial $\psi_0=5000$ in the experiments. We trained each benchmark in 100K steps and reported the metrics over 10 evaluated episodes with different seeds. Each episode has a maximum of 1000 steps. The metrics in CARLA are the episode reward (ER) and the driving distance (DD) of the evaluated episodes. More implementation details are shown in the Appendix.

Comparison with State-of-the-art

Results on CARLA. First, we evaluate COVR against various visual RL methods, including SAC (Haarnoja et al. 2018b), auxiliary loss-based methods (CURL (Laskin, Srinivas, and Abbeel 2020), MLR (Yu et al. 2022)), a data augmentation method (DrQ (Kostrikov, Yarats, and Fergus 2020)), motion modeling methods (DeepMDP (Gelada et al. 2019), SPR (Schwarzer et al. 2020), ResAct (Liu, Peng, and Tian 2025)), and buffer-based optimization methods (PER (Schaul et al. 2015), ERE (Wang and Ross 2019)). In addition, we compare the VLM-based methods in the same direction, including a VLM-based executor (VBE (Mei et al. 2024)), the methods of directly adding prior loss (DPL (Xu et al. 2024b), DGC (Xu et al. 2025b)), a method of annealing weight-based policy loss (APL (Zhou et al. 2024b; Lee et al. 2025)), and a VLM-based policy fine-tuning method (VPF (Zhai et al. 2024; Wei et al. 2025)).

As presented in Table 1 and Table 2, COVR improves on both the episode reward and driving distance. We make the following conclusions: (1) VBE underperforms due to the inherent limitations of the basic VLM in adapting to continuous and dynamic environments; (2) The methods, including DPL, APL, and DGC, exhibit limited performance due to suboptimal knowledge transfer from unrefined VLMs; (3) Vanilla visual RL methods struggle to adapt to complex sce-

Type	Methods	ER \uparrow	DD \uparrow
Vanilla visual RL	SAC	38 \pm 29	40 \pm 29
	DeepMDP	82 \pm 51	85 \pm 52
	CURL	78 \pm 52	81 \pm 53
	DrQ	110 \pm 68	112 \pm 69
	SPR	62 \pm 42	66 \pm 45
	MLR	118 \pm 50	121 \pm 51
	PER	51 \pm 45	54 \pm 46
	ERE	55 \pm 50	58 \pm 53
	ResAct	212 \pm 54	216 \pm 55
Only the VLM	VBE	9 \pm 7	18 \pm 7
VLM-assisted visual RL	DPL	127 \pm 51	129 \pm 52
	APL	107 \pm 54	111 \pm 55
	VPF	81 \pm 2	82 \pm 3
	DGC	146 \pm 14	169 \pm 18
	COVR (Ours)	235 \pm 89	237 \pm 89

Table 2: Performance comparison in the #GP scenario. The best results for each metric are denoted by \blacksquare .

narios and thus have low performance; (4) COVR provides more reliable policy guidance through the collaborative optimization framework, enabling smoother and more stable vehicle control.

Results on DMControl. For DMControl, we first introduce COVR on several typical different baselines including SAC (Haarnoja et al. 2018b), DeepMDP (Gelada et al. 2019), and RAD (Laskin et al. 2020). As presented in Table 3, COVR consistently improves the performance of multiple baseline methods, highlighting its strong adaptability.

We select RAD+COVR as our baseline. Then, we select a range of SOTA methods for comparison with COVR including (1) model-free methods: CURL, DrQ, SVEA (Hansen, Su, and Wang 2021), PlayVirtual (Yu et al. 2021), TACO (Zheng et al. 2023), MLR, MADI (Grooten et al. 2024), PSRL (Choi et al. 2023), ResAct, and (2) model-based methods: Dreamer (Hafner et al. 2019a), PlaNet (Hafner et al. 2019b). We also present the results of previous VLM-based methods, including VBE, DPL, APL, and VPF. As reported in Table 4, COVR achieves new state-of-the-art. Furthermore, the significantly lower reward standard deviation indicates that COVR also offers improved convergence stability and optimization ease. The results of hard tasks can be found in the Appendix.

Ablation Study

To assess the contributions of each component of COVR, we conduct ablation studies for the #HW scenario. The results are summarized in Table 5. We can draw the following analyses:

Effects of EDDF module. The results of **M1** demonstrate that the EDDF module outperforms random data filtering by preventing the VLM from learning suboptimal knowledge. **M2–M4** further validate EDDF’s superiority over fixed top-k methods: rigid thresholding risks missing high-

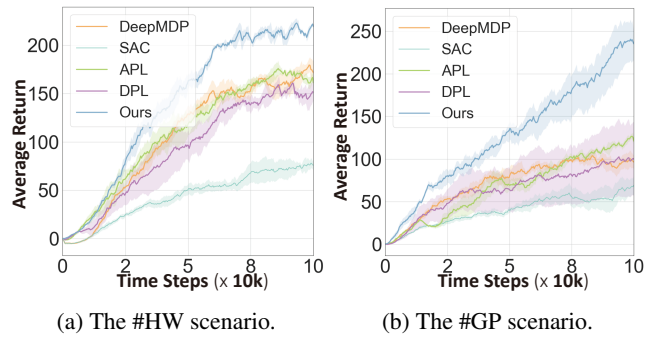


Figure 3: Visualization of the average return curve during training in CARLA.

quality actions or introducing noisy samples. In contrast, EDDF dynamically adapts thresholds to capture low-return yet valuable actions during exploration, thereby improving the knowledge density and generalization of the VLM. **M5** shows that Z-score normalization enhances data stability and contributes to more robust learning. **M6** shows that evaluating performance based on trajectory-level cumulative returns yields better insights than per-step immediate rewards. **M7** indicates that, due to the stochastic nature of RL exploration, unstable initial Q -value estimation may lead to the selection of low-quality samples, which in turn degrades the fine-tuning performance of the VLM and negatively impacts policy learning.

Effects of RALW module. Experiments on **M8** and **M9** validate the proposed RALW module. The results indicate that its performance cannot be replicated using randomly generated returns, highlighting the necessity of learning adaptive loss weights based on the dynamic returns.

Effects of the VLM. To highlight the importance of VLM guidance, we design a baseline in **M10** and **M11** that selects top-return samples (80% and 20% per batch) for visual reinforcement learning training, replacing the rest with random actions to balance exploration and exploitation. Despite this strategy, it underperforms COVR, confirming the critical role of enhancing VLM task knowledge and leveraging its generalization for policy guidance.

In the Appendix, we further evaluate various experiments on COVR. These experiments offer additional evidence of COVR’s effectiveness.

Visual Analysis

Fig. 3 presents the average return curve during training of some SOTA methods in the CARLA scenarios. As shown, COVR effectively leverages the powerful reasoning and generalization capabilities of the VLM, significantly enhancing policy learning efficiency in visual RL. In contrast, other baseline methods fail to achieve superior global performance due to the lack of effective self-exploration mechanisms or limitations imposed by their VLMs’ guidance. Additional visual analyses in the Appendix, such as VLM action reasoning differences before/after fine-tuning and quan-

Tasks/Methods	SAC	SAC+COVR	DeepMDP	DeepMDP+COVR	RAD	RAD+COVR
Cartpole, Swingup	237 ± 49	740 ± 108	389 ± 44	793 ± 26	694 ± 28	872 ± 2
Reacher, Easy	239 ± 183	246 ± 345	471 ± 173	461 ± 443	734 ± 87	969 ± 18
Cheetah, Run	118 ± 13	156 ± 29	306 ± 25	352 ± 35	364 ± 38	504 ± 13
Walker, Walk	95 ± 19	194 ± 57	384 ± 197	397 ± 71	552 ± 87	802 ± 25
Finger, Spin	230 ± 194	247 ± 51	509 ± 72	653 ± 18	813 ± 65	976 ± 9
Ball in cup, Catch	85 ± 130	185 ± 331	704 ± 24	757 ± 169	825 ± 49	960 ± 23

Table 3: Implementation of COVR on top of three different baselines. +COVR represents the preceding method combined with COVR.

Type	Methods	Cartpole, Swingup	Reacher, Easy	Cheetah, Run	Walker, Walk	Finger, Spin	Ball in cup, Catch
Vanilla visual RL	CURL	582 ± 146	538 ± 233	299 ± 48	403 ± 24	767 ± 56	769 ± 43
	DrQ	759 ± 92	601 ± 213	344 ± 67	612 ± 164	901±104	913±53
	SVEA	727 ± 86	811 ± 115	375 ± 54	747 ± 65	859 ± 77	915 ± 71
	PlayVirtual	816 ± 36	785 ± 142	474 ± 50	460 ± 173	915 ± 49	929 ± 31
	TACO	782 ± 51	821 ± 97	402 ± 62	601 ± 103	876 ± 67	902 ± 54
	MLR	806 ± 48	866 ± 103	482 ± 38	643 ± 114	907 ± 58	933 ± 16
	MADI	704 ± 54	766 ± 101	432 ± 44	574 ± 94	810 ± 95	884 ± 36
	PSRL	849 ± 63	621 ± 202	398 ± 71	595 ± 104	882 ± 132	922 ± 60
	ResAct	819 ± 44	917 ± 59	503 ± 42	772 ± 65	974 ± 42	948 ± 44
	Dreamer	326±27	314±155	235±137	277±12	341±70	246±174
PlaNet	563±73	82±174	165±123	224±48	560±77	0±0	
Only the VLM	VBE	178 ± 31	94 ± 138	3 ± 2	26 ± 11	0 ± 0	49 ± 213
VLM-assisted visual RL	DPL	776 ± 5	224 ± 387	255 ± 16	209 ± 64	815 ± 8	751 ± 148
	APL	820 ± 3	292 ± 436	367 ± 30	696 ± 48	839 ± 9	905 ± 95
	VPF	749 ± 21	162 ± 137	127 ± 2	132 ± 4	0 ± 0	0 ± 0
	COVR (Ours)	872 ± 2	969 ± 18	504 ± 13	802 ± 25	976 ± 9	960 ± 23

Table 4: Performance comparison with SOTA methods in DMControl. The best results for each metric are denoted by ■.

Type	Idx.	ER ↑	DD ↑	Description
Effects of EDDF module.	M1	144 ± 74	155 ± 77	W/o EDDF module, i.e., randomly filtering data for fine-tuning.
	M2	204 ± 137	214 ± 141	Replace the EDDF module with the return-based top-80% method.
	M3	217 ± 95	229 ± 98	Replace the EDDF module with the return-based top-90% method.
	M4	192 ± 107	203 ± 110	Replace the EDDF module with the return-based top-95% method.
	M5	210 ± 99	221 ± 102	W/o Z-score to standardize the data.
	M6	221 ± 105	231 ± 108	Change return to reward.
	M7	200 ± 118	212 ± 122	Change return to Q -value .calculated by $Q_\phi(\cdot)$.
Effects of RALW module.	M8	204 ± 111	214 ± 115	W/o RALW module.
	M9	184 ± 78	191 ± 81	The loss weight is randomly generated instead of return-based guidance.
Effects of the VLM.	M10	183 ± 96	195 ± 99	Select top-return (80%) and randomly mixed samples to train RL.
	M11	175 ± 58	185 ± 58	Select top-return (50%) and randomly mixed samples to train RL.
Ours	COVR	248 ± 81	259 ± 85	Full version.

Table 5: Ablation experiments conducted in the #HW scenario validate the effectiveness of COVR. The best results for each metric are denoted by ■.

titative evaluation of the EDDF module, further validate the effectiveness of COVR.

Conclusion

We have proposed a novel LM-assisted RL method that features a collaborative optimization paradigm. It comprises two key modules: one for effective sample selection from RL

trajectories, and another for mitigating action inconsistency during fine-tuning. In addition, we have introduced a progressive fine-tuning approach to reduce resource consumption and enhance efficiency. Experiments across various complex and high-dimensional environments have shown that COVR achieves state-of-the-art performance.

Acknowledgments

The study was funded by the Shenzhen Basic Research Fund under grant JCYJ20241202130025030; Shenzhen Science and Technology Program (KQTD20240729102051063); the National Natural Science Foundation of China under contracts No. 62422602, No. 62372010, No. 62425101, No. 62332002, No. 62206281; Key Laboratory Grants 241-HF-D05-01; and the major key project of the Peng Cheng Laboratory (PCL2021A13 and PCL2025A02). Computing support was provided by Pengcheng Cloudbrain.

References

- Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. 2025. Qwen2. 5-v1 technical report. *arXiv preprint arXiv:2502.13923*.
- Cheadle, C.; Vawter, M. P.; Freed, W. J.; and Becker, K. G. 2003. Analysis of microarray data using Z score transformation. *The Journal of molecular diagnostics*, 5(2): 73–81.
- Chen, L.; Lei, Y.; Jin, S.; Zhang, Y.; and Zhang, L. 2024. Rlingua: Improving reinforcement learning sample efficiency in robotic manipulations with large language models. *IEEE Robotics and Automation Letters*.
- Choi, H.; Lee, H.; Song, W.; Jeon, S.; Sohn, K.; and Min, D. 2023. Local-Guided Global: Paired Similarity Representation for Visual Reinforcement Learning. In *CVPR*, 15072–15082.
- Dalal, M.; Chiruvolu, T.; Chaplot, D.; and Salakhutdinov, R. 2024. Plan-seq-learn: Language model guided rl for solving long horizon robotics tasks. *arXiv preprint arXiv:2405.01534*.
- Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*, 1–16. PMLR.
- Fu, Y.; Zhang, H.; Wu, D.; Xu, W.; and Boulet, B. 2024. Furl: Visual-language models as fuzzy rewards for reinforcement learning. *arXiv preprint arXiv:2406.00645*.
- Gelada, C.; Kumar, S.; Buckman, J.; Nachum, O.; and Belle-mare, M. G. 2019. Deepmdp: Learning continuous latent space models for representation learning. In *ICML*, 2170–2179. PMLR.
- Grooten, B.; Tomilin, T.; Vasan, G.; Taylor, M. E.; Mahmood, A. R.; Fang, M.; Pechenizkiy, M.; and Mocanu, D. C. 2024. MaDi: Learning to Mask Distractions for Generalization in Visual Deep Reinforcement Learning. In *AAMAS*, 733–742.
- Guo, L.; Andriopoulos, G.; Zhao, Z.; Ling, S.; Dong, Z.; and Ross, K. 2024. Cross entropy versus label smoothing: A neural collapse perspective. *arXiv preprint arXiv:2402.03979*.
- Guo, Y.; Zhang, J.; Chen, X.; Ji, X.; Wang, Y.-J.; Hu, Y.; and Chen, J. 2025. Improving Vision-Language-Action Model with Online Reinforcement Learning. *arXiv preprint arXiv:2501.16664*.
- Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018a. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, 1861–1870. PMLR.
- Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. 2018b. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.
- Hafner, D.; Lillicrap, T.; Ba, J.; and Norouzi, M. 2019a. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*.
- Hafner, D.; Lillicrap, T.; Fischer, I.; Villegas, R.; Ha, D.; Lee, H.; and Davidson, J. 2019b. Learning latent dynamics for planning from pixels. In *ICML*, 2555–2565. PMLR.
- Hansen, N.; Su, H.; and Wang, X. 2021. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *NIPS*, 34: 3680–3693.
- Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.
- Kostrikov, I.; Yarats, D.; and Fergus, R. 2020. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*.
- Laskin, M.; Lee, K.; Stooke, A.; Pinto, L.; Abbeel, P.; and Srinivas, A. 2020. Reinforcement learning with augmented data. *NIPS*, 33: 19884–19895.
- Laskin, M.; Srinivas, A.; and Abbeel, P. 2020. Curl: Contrastive unsupervised representations for reinforcement learning. In *ICML*, 5639–5650. PMLR.
- Lee, D.; Luu, T. M.; Lee, Y.; and Yoo, C. D. 2025. Sample Efficient Reinforcement Learning via Large Vision Language Model Distillation. In *ICASSP*, 1–5. IEEE.
- Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*.
- Liu, Z.; Peng, P.; and Tian, Y. 2025. Visual Reinforcement Learning with Residual Action. In *AAAI*, volume 39, 19050–19058.
- Mei, A.; Zhu, G.-N.; Zhang, H.; and Gan, Z. 2024. Replan-VLM: Replanning robotic tasks with visual language models. *IEEE Robotics and Automation Letters*.
- Paischer, F.; Adler, T.; Hofmarcher, M.; and Hochreiter, S. 2023. Semantic helm: A human-readable memory for reinforcement learning. *NIPS*, 36: 9837–9865.
- Pan, C.; Yaman, B.; Nesti, T.; Mallik, A.; Allievi, A. G.; Velipasalar, S.; and Ren, L. 2024. VLP: Vision Language Planning for Autonomous Driving. In *CVPR*, 14760–14769.
- Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *NIPS*, 36: 53728–53741.
- Schaul, T.; Quan, J.; Antonoglou, I.; and Silver, D. 2015. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.
- Schwarzer, M.; Anand, A.; Goel, R.; Hjelm, R. D.; Courville, A.; and Bachman, P. 2020. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*.

- Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Shinn, N.; Cassano, F.; Labash, B.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: Language agents with verbal reinforcement learning.(2023). *arXiv preprint cs.AI/2303.11366*.
- Shukla, Y.; Gao, W.; Sarathy, V.; Velasquez, A.; Wright, R.; and Sinapov, J. 2024. LgTS: Dynamic Task Sampling using LLM-generated Sub-Goals for Reinforcement Learning Agents. In *AAMAS*, 1736–1744.
- Tan, W.; Zhang, W.; Liu, S.; Zheng, L.; Wang, X.; and An, B. 2024. True knowledge comes from practice: Aligning llms with embodied environments via reinforcement learning. *arXiv preprint arXiv:2401.14151*.
- Tang, Y.; Yu, W.; Tan, J.; Zen, H.; Faust, A.; and Harada, T. 2023. SayTap: Language to Quadrupedal Locomotion. In *Conference on Robot Learning*, 3556–3570. PMLR.
- Tassa, Y.; Doron, Y.; Muldal, A.; Erez, T.; Li, Y.; Casas, D. d. L.; Budden, D.; Abdolmaleki, A.; Merel, J.; Lefrancq, A.; et al. 2018. Deepmind control suite. *arXiv preprint arXiv:1801.00690*.
- Team, G.; Georgiev, P.; Lei, V. I.; Burnell, R.; Bai, L.; Gulati, A.; Tanzer, G.; Vincent, D.; Pan, Z.; Wang, S.; et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Vinutha, H.; Poornima, B.; and Sagar, B. 2018. Detection of outliers using interquartile range technique from intrusion dataset. In *Information and decision sciences: Proceedings of the 6th international conference on ficta*, 511–518. Springer.
- Waite, J. R.; Hasan, M. Z.; Liu, Q.; Jiang, Z.; Hegde, C.; and Sarkar, S. 2025. RLS3: RL-Based Synthetic Sample Selection to Enhance Spatial Reasoning in Vision-Language Models for Indoor Autonomous Perception. In *ICCPs*, 1–10.
- Wang, C.; and Ross, K. 2019. Boosting soft actor-critic: Emphasizing recent experience without forgetting the past. *arXiv preprint arXiv:1906.04009*.
- Wang, R.; Zhao, D.; Yuan, Z.; Obi, I.; and Min, B.-C. 2025. Prefclm: Enhancing preference-based reinforcement learning with crowdsourced large language models. *IEEE Robotics and Automation Letters*.
- Wang, Y.; Sun, Z.; Zhang, J.; Xian, Z.; Biyik, E.; Held, D.; and Erickson, Z. 2024a. RL-VLM-F: Reinforcement Learning from Vision Language Foundation Model Feedback. In *ICML*, 51484–51501. PMLR.
- Wang, Z.; Cai, S.; Chen, G.; Liu, A.; Ma, X. S.; and Liang, Y. 2024b. Describe, explain, plan and select: interactive planning with LLMs enables open-world multi-task agents. *NIPS*, 36.
- Wei, T.; Yang, Y.; Xing, J.; Shi, Y.; Lu, Z.; and Ye, D. 2025. GTR: Guided Thought Reinforcement Prevents Thought Collapse in RL-based VLM Agent Training. *arXiv preprint arXiv:2503.08525*.
- Xie, T.; Zhao, S.; Wu, C. H.; Liu, Y.; Luo, Q.; Zhong, V.; Yang, Y.; and Yu, T. 2024. Text2reward: Automated dense reward function generation for reinforcement learning. In *International Conference on Learning Representations (ICLR), 2024 (07/05/2024-11/05/2024, Vienna, Austria)*.
- Xin, H.; Ren, Z.; Song, J.; Shao, Z.; Zhao, W.; Wang, H.; Liu, B.; Zhang, L.; Lu, X.; Du, Q.; et al. 2024. Deepseek-prover-v1. 5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search. *arXiv preprint arXiv:2408.08152*.
- Xu, C.; Liu, J.; Hang, P.; and Sun, J. 2025a. TeLL-Drive: Enhancing Autonomous Driving with Teacher LLM-Guided Deep Reinforcement Learning. *arXiv preprint arXiv:2502.01387*.
- Xu, H.; Peng, P.; Tan, G.; Chang, Y.; Li, L.; and Tian, Y. 2025b. VLMs-Guided Representation Distillation for Efficient Vision-Based Reinforcement Learning. In *CVPR*, 29534–29544.
- Xu, H.; Peng, P.; Tan, G.; Li, Y.; Xu, X.; and Tian, Y. 2024a. DMR: Decomposed Multi-Modality Representations for Frames and Events Fusion in Visual Reinforcement Learning. In *CVPR*, 26508–26518.
- Xu, Y.; Hu, Y.; Zhang, Z.; Meyer, G. P.; Mustikovela, S. K.; Srinivasa, S.; Wolff, E. M.; and Huang, X. 2024b. Vlm-ad: End-to-end autonomous driving through vision-language model supervision. *arXiv preprint arXiv:2412.14446*.
- Xu, Z.; Yu, C.; Fang, F.; Wang, Y.; and Wu, Y. 2024c. Language Agents with Reinforcement Learning for Strategic Play in the Werewolf Game. In *ICML*, 55434–55464. PMLR.
- Yu, T.; Lan, C.; Zeng, W.; Feng, M.; Zhang, Z.; and Chen, Z. 2021. Playvirtual: Augmenting cycle-consistent virtual trajectories for reinforcement learning. *NIPS*, 34: 5276–5289.
- Yu, T.; Zhang, Z.; Lan, C.; Lu, Y.; and Chen, Z. 2022. Mask-based latent reconstruction for reinforcement learning. *NIPS*, 35: 25117–25131.
- Zhai, S.; Bai, H.; Lin, Z.; Pan, J.; Tong, P.; Zhou, Y.; Suhr, A.; Xie, S.; LeCun, Y.; Ma, Y.; et al. 2024. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *NIPS*, 37: 110935–110971.
- Zhang, A.; McAllister, R.; Calandra, R.; Gal, Y.; and Levine, S. 2020. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*.
- Zheng, R.; Wang, X.; Sun, Y.; Ma, S.; Zhao, J.; Xu, H.; Daumé III, H.; and Huang, F. 2023. TACO: Temporal Latent Action-Driven Contrastive Loss for Visual Reinforcement Learning. *NIPS*, 36: 48203–48225.
- Zhou, Z.; Hu, B.; Zhao, C.; Zhang, P.; and Liu, B. 2024a. Large language model as a policy teacher for training reinforcement learning agents. In *IJCAI*, 5671–5679.
- Zhou, Z.; Hu, B.; Zhao, C.; Zhang, P.; and Liu, B. 2024b. Large language model as a policy teacher for training reinforcement learning agents. In *IJCAI*, 5671–5679.