Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25)
Survey Track

The Evolving Landscape of LLM- and VLM-Integrated Reinforcement Learning

Sheila Schoepp[1*], **Masoud Jafaripour**[1], **Yingyue Cao**[1], **Tianpei Yang**[2*], **Fatemeh Abdollahi**[1], **Shadan Golestan**[3], **Zahin Sufiyan**[1], **Osmar R. Zaiane**[1,3] and **Matthew E. Taylor**[1,3]
[1]University of Alberta
[2]Nanjing University
[3]Alberta Machine Intelligence Institute (Amii)
sschoepp@ualberta.ca, tianpei.yang@nju.edu.cn

Abstract

Reinforcement learning (RL) has shown impressive results in sequential decision-making tasks. Large Language Models (LLMs) and Vision-Language Models (VLMs) have recently emerged, exhibiting impressive capabilities in multimodal understanding and reasoning. These advances have led to a surge of research integrating LLMs and VLMs into RL. This survey reviews representative works in which LLMs and VLMs are used to overcome key challenges in RL, such as lack of prior knowledge, long-horizon planning, and reward design. We present a taxonomy that categorizes these LLM/VLM-assisted RL approaches into three roles: agent, planner, and reward. We conclude by exploring open problems, including grounding, bias mitigation, improved representations, and action advice. By consolidating existing research and identifying future directions, this survey establishes a framework for integrating LLMs and VLMs into RL, advancing approaches that unify natural language and visual understanding with sequential decision-making.

1 Introduction

Reinforcement learning (RL) is an influential branch of machine learning that enables autonomous agents to learn sequential decision-making strategies through trial-and-error interaction with their environment. When integrated with deep neural networks, deep RL has made breakthroughs in challenging domains such as games and robotics [Schulman *et al.*, 2017; Vinyals *et al.*, 2019]. Despite these advances, RL still faces key challenges that may hinder real-world deployment, including reliance on human-designed rewards, sample inefficiency, poor generalization, and limited interpretability. These limitations motivate the exploration of novel techniques to enhance the capabilities of RL.

Large Language Models (LLMs) represent a groundbreaking advancement in artificial intelligence (AI), exhibiting unprecedented capabilities in natural language understanding, generation, and reasoning. By training large architectures with billions or even trillions of parameters on internet-scale datasets, LLMs such as GPT-3 [Brown *et al.*, 2020] have demonstrated emergent capabilities that smaller models could not achieve. Leveraging these strengths, LLMs are now applied to tasks that extend beyond conventional Natural Language Processing (NLP), spanning domains from healthcare to robotics [Ichter *et al.*, 2022; Thirunavukarasu *et al.*, 2023]. Similarly, Vision-Language Models (VLMs), which integrate visual perception with natural language understanding, can interpret and reason about images through language. Leveraging large-scale, aligned image-text training, VLMs like CLIP [Radford *et al.*, 2021] can perform a variety of tasks, including image-text retrieval and classification. Other VLMs, such as PaLM-E [Driess *et al.*, 2023], are designed to respond to natural language prompts, broadening their versatility to tasks such as image captioning, scene understanding, and visual question answering. Together, these Foundation Models (FMs), specifically LLMs and VLMs, have reshaped AI by capturing nuanced, human-centric semantics across modalities, enabling flexible, human-aligned problem-solving based on their vast training data.

Integrating LLMs and VLMs into the RL framework promises a transformative leap in how agents act and learn. While RL is proficient at learning from trial-and-error, it typically lacks the broad world knowledge and powerful reasoning capabilities that LLMs and VLMs can provide. When integrated with RL, these models improve agents' capabilities by providing semantic understanding (LLMs) or robust perception (VLMs), thereby improving data efficiency, generalization, and interpretability. In some cases, RL's ability to continually refine behaviour through interactions with the environment can complement these FMs by providing supplemental training or richer context and improving their outputs.

Research in the area of LLMs and VLMs is driving an AI evolution. As a result, the integration of FMs into RL is also rapidly progressing, further expanding the limits of what RL can accomplish. Despite prior work on integrating LLMs into RL [Cao *et al.*, 2024; Pternea *et al.*, 2024], the fast pace of the field demands continuous analysis of emerging methods and applications. Furthermore, with the emergence of LLM agents and powerful VLMs—a perspective not addressed by earlier surveys—this survey complements existing work by introducing these new dimensions, expanding our understanding of how best to integrate FMs with RL.

*Corresponding author

This survey selectively examines peer-reviewed studies that employ pre-trained LLMs and large VLMs developed on or after June 2020—coinciding with the release of GPT-3, a notable milestone in NLP due to its unprecedented scale and capabilities—as a core methodological component. We consider FMs that employ a transformer-based architecture (whether encoder-only, decoder-only, or encoder-decoder) and address sequential decision-making tasks framed as Markov decision processes (MDPs). We highlight works that use rewards to optimize RL or LLM/VLM policies for improved sequential decision-making. Although RL can fine-tune language models, we focus on using FMs to enhance RL, not simply improving the models themselves. We include a representative selection of papers, acknowledging that some relevant studies are omitted due to space constraints.

In summary, the main contributions of this survey are: **(1)** A unifying taxonomy that categorizes FM functionalities in RL into three key roles: LLM/VLM as agent, LLM/VLM as planner, and LLM/VLM as reward. **(2)** A review of key works within each category, highlighting how they address key RL challenges (e.g., policy learning, long-horizon planning, and reward specification). **(3)** Future directions that identify limitations in existing approaches and outline promising paths for FM-RL research.

2 Preliminaries

2.1 Reinforcement Learning

A **Markov decision process** (MDP) is defined by the tuple $\langle S, A, T, R, \gamma \rangle$, where S is the set of states, A is the set of actions, $T : S \times A \to P(S)$ is the transition probability function, $R : S \times A \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor [Sutton and Barto, 2018].

Reinforcement learning (RL) is a paradigm in which an agent learns through interactions with an environment, typically modelled as an MDP [Sutton and Barto, 2018]. These interactions produce a *trajectory* of states, actions, and rewards as the agent explores its surroundings. A central concept in RL is the *policy* π that maps states to actions (or distributions over actions), formally expressed as $\pi : S \to P(A)$. In some settings, this is further extended to a *language-conditioned policy*, $\pi_l : S \times \mathcal{L} \to P(A)$, where $\mathcal{L}$ represents the space of natural language instructions (e.g., sub-goals), allowing the agent to incorporate linguistic guidance into its decisions. Under a policy π, the *value function*, $v : S \to \mathbb{R}$ (or $q : S \times A \to \mathbb{R}$), estimates the expected cumulative reward from a given state (or state-action pair). A language-conditioned (action-)value function can likewise be conditioned on an instruction l.

2.2 Language Models

Large Language Models (LLMs) learn statistical patterns in text from large corpora, enabling them to predict the likelihood of word (or token) sequences in context. They often rely on transformer architectures, which use self-attention to capture token dependencies [Vaswani *et al.*, 2017]. Transformer-based LLMs include encoder-only models that mask part of the input and learn to predict the missing portion (e.g., text understanding), decoder-only models that generate text by predicting the next token in a sequence (e.g., text generation), and encoder-decoder models that encode input into a latent representation and then decode it (e.g., translation tasks).

Vision-Language Models (VLMs) are multimodal, processing both visual and textual data, often relying on transformers. They can be categorized into encoder-decoder models that convert images and/or text into latent embeddings before generating output (used for tasks like captioning), dual-encoder models that embed images and text separately into a shared latent space (used for similarity matching and retrieval), and single encoder models that encode images and text jointly (used for tasks like visual question answering).

2.3 Taxonomy

Figure 1 presents a three-part taxonomy to integrate LLMs and VLMs into RL, distinguishing three primary roles: (1) LLM/VLM as Agent, where the FM serves as a policy. These methods can either be parametric, fine-tuning the FM to generate task-relevant outputs, or non-parametric, enriching the prompts with additional context. (2) LLM/VLM as Planner, where the FM generates sub-goals for complex tasks. The FM may produce a comprehensive sequence of sub-goals in one pass or incrementally produce them (i.e., step-by-step), awaiting a signal of success or failure before generating the next sub-goal. (3) LLM/VLM as Reward, where the FM shapes rewards by generating the reward function code to specify the reward or by serving as (or helping train) a reward model that outputs a scalar reward signal. Table 1 provides an overview of FM-RL methods, classified according to the taxonomy.

Some approaches do not fit into these three primary roles; instead, LLMs/VLMs are integrated into RL using alternative methods. For example, KALM [Pang *et al.*, 2024] uses the FM as a world model to generate "imaginary" trajectories. Lai and Zang [2024] use an FM to identify and emphasize higher-quality trajectories. MaestroMotif [Klissarov *et al.*, 2025] and LAST [Fu *et al.*, 2024] guide hierarchical RL by discovering and coordinating skills.

In subsequent sections, we examine the three primary categories in our taxonomy, investigating the distinct ways that LLMs and VLMs can be integrated into and benefit RL.

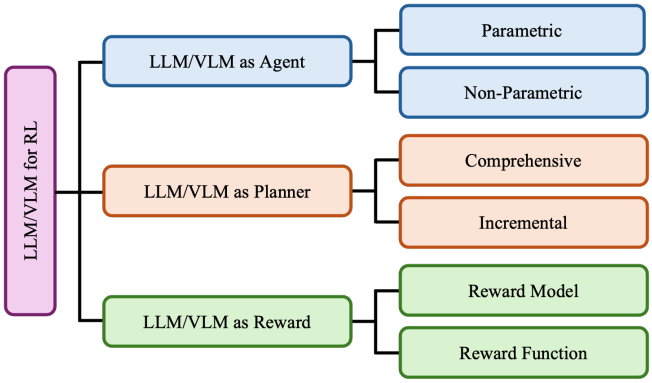


Figure 1: A taxonomy for LLM- and VLM-assisted RL

			Foundation Model (FM)			Reinforcement Learning (RL)					
		Citation	Model(s)	FT	Role	Agent	Task	Setting	Role	Metrics	Code
LLM/VLM as Actor	Parametric	AGILE [Feng *et al.*, 2024]	Meerkat, Vicuna-1.5	✓*	act, ref	single	-	online	π_l, v_l, rft	acc, rew	Link
		Retroformer [Yao *et al.*, 2024]	GPT-3, GPT-4, LongChat	✓	act, cr, ref	single	-	offline	π_l, rft	sr, se	-
		TWOSOME [Tan *et al.*, 2024]	Llama	✓*	act	single	-	online	π_l, v_l, rft	sr, rew, gen, se	Link
		POAD [Wen *et al.*, 2024]	CodeLlama, Llama 2	✓*	act	single	-	online	π_l, v_l, rft	rew, gen, se	Link
		GLAM [Carta *et al.*, 2023]	FLAN-T5	✓	act	single	-	online	π_l, v_l, rft	se, gen	Link
		Zhai *et al.* [Zhai *et al.*, 2024]	LLaVA-v1.6-Mistral	✓*	act	single	-	online	π_l, v_l, rft	sr	Link
	Non-Parametric	ICPI [Brooks *et al.*, 2023]	Codex	×	act, wm, ref	single	-	online	π_l, v_l, τ_π	rew, gen	Link
		Reflexion [Shinn *et al.*, 2023]	GPT-3, GPT-3.5-Turbo, GPT-4	×	act, eval, cr, ref	single	-	online	π_l, τ_π	sr, acc	Link
		REMEMBERER [Zhang *et al.*, 2023a]	GPT-3.5	×	act	single	-	online	π_l, v_l, τ_π	sr, rob	Link
		ExpeL [Zhao *et al.*, 2024]	GPT-3.5-Turbo, GPT-4	×	act, cr, ref	single	-	online	π_l, τ_π	sr, gen	Link
		RLingua [Chen *et al.*, 2024]	GPT-4	×	act, plan, ref	single	-	online	π_g, v_g, τ_π	sr, se	-
		Xu *et al.* [Xu *et al.*, 2024]	GPT-3.5-Turbo	×	act, o	multi	-	online	π_l, v_l	sr, rob	-
		LangGround [Li *et al.*, 2024]	GPT-4	×	act, o	multi	-	online	π_l	sr, gen, se, int	Link
LLM/VLM as Planner	Comprehensive	SayTap [Tang *et al.*, 2023]	GPT-4	×	plan	single	multi	online	π_g, v_g	sr, acc	-
		LgTS [Shukla *et al.*, 2024]	Llama 2	×	plan	single	multi	online	π_g, v_g	sr, se	-
		PSL [Dalal *et al.*, 2024]	GPT-4	×	plan, o	single	single	online	π, v	sr, gen, se	Link
		LLaRP [Szot *et al.*, 2024]	Llama	×	plan, o	single	multi	online	π_g, v_g	sr, gen, rob, se	Link
		LMA3 [Colas *et al.*, 2023]	GPT-3.5-Turbo	×	plan, rew, eval, o	single	multi	online	π_g	gen, exp	-
		When2Ask [Hu *et al.*, 2024]	Vicuna	×	plan	single	single	online	π, v	sr	Link
		Inner Monologue [Huang *et al.*, 2022]	GPT-3, PaLM	×	plan, cr, ref	single	multi	-	o	sr, rob, al	-
	Incremental	SayCan [Ichter *et al.*, 2022]	PaLM	×	plan	single	multi	both	π_l, v_l, rft	sr, rob	Link
		LLM4Teach [Zhou *et al.*, 2024]	ChatGLM-Turbo, Vicuna	×	plan	single	single	online	π, v	sr, se	Link
		AdaRefiner [Zhang and Lu, 2024]	Llama 2, GPT-4	✓*	plan, cr, ref, o	single	multi	online	π_l, v_l, τ_π	sr, rew, gen, exp	Link
		BOSS [Zhang *et al.*, 2023b]	Llama	×	plan, o	single	multi	both	π_l, v_l, rft, τ_π	sr, gen, rob, se	-
		Text2Motion [Lin *et al.*, 2023]	Codex, GPT-3.5	×	plan, o	single	multi	offline	π, v	sr, gen, int	-
LLM/VLM as Reward	Function	Text2Reward [Xie *et al.*, 2024]	GPT-4	×	rew, ref	single	single	online	π	sr, se, al	Link
		Zeng *et al.* [Zeng *et al.*, 2024]	GPT-4	×	rew, eval, cr, ref	single	single	online	π, τ_π	sr, se	-
		Eureka [Ma *et al.*, 2024]	GPT-4	×	rew, cr, ref	single	single	online	π, v, τ_π	sr, gen, se, al	Link
	Model	Kwon *et al.* [Kwon *et al.*, 2023]	GPT-3	×	rew	single	single	online	π, v	acc, se, al	-
		PREDILECT [Holk *et al.*, 2024]	GPT-4	×	rew, o	single	single	online	π	rew, se, al	-
		ELLM [Du *et al.*, 2023]	Codex, GPT-3	×	rew, plan	single	multi	online	π_l, v_l	sr, gen, se, exp	-
		RL-VLM-F [Wang *et al.*, 2024]	Gemini-Pro, GPT-4V	×	rew, eval	single	single	online	π, v	sr, rew, se	Link
		VLM-RM [Rocamonde *et al.*, 2024]	CLIP	×	rew	single	single	online	π, v	sr, al	Link
		MineCLIP [Fan *et al.*, 2022]	CLIP	✓*	rew, eval	single	multi	online	π_l, v_l	sr, gen, se, al	Link

Table 1: A summary of approaches leveraging FMs, specifically LLMs and VLMs, to enhance RL, organized according to the taxonomy illustrated in Figure 1 and listed in order of mention. **FT (Fine-Tuning)** ✓(full fine-tuning), ✓* (parameter-efficient fine-tuning), and × (no fine-tuning). **FM Role** Generation of *act* (actions), *plan* (high-level plan), *rew* (reward function/model), *wm* (world model), *eval* (task success evaluations), *cr* (critiques and improvement suggestions), *ref* (refinement), and *o* (other). **RL Agent** *single* (single agent) and *multi* (multi-agent). **RL Task** *single* (single task) and *multi* (multi-task). **RL Setting** *online* (learning from real-time interactions), *offline* (learning from precollected interactions), and *both*. **RL Role** π (policy learning), π_g (goal-conditioned policy learning), π_l (language-conditioned policy learning), v (value function learning), v_g (goal-conditioned value function learning), v_l (language-conditioned value function learning), τ_π (policy execution to generate trajectories), *ref* (reinforced fine-tuning), *o* (other), and *n/a* (no RL role). **Metrics** Improvements in *acc* (accuracy ↑), *sr* (success rate ↑), *rew* (reward or return ↑), *gen* (generalization ↑), *rob* (robustness ↑), *se* (sample efficiency ↑), *exp* (exploration ↑), *al* (alignment with humans ↑), and *int* (interpretability ↑). Hyperlinks to code are embedded.

3 LLM/VLM as Agent

Language-based decision-making agents leverage the reasoning, planning, and generalization capabilities of LLMs, enabling them to perform complex tasks in interactive environments. These agents interact with the environment, acting as decision makers at each time step to generate context-based actions. Recent advances classify agents as parametric, fine-tuning LLMs for dynamic adaptation, or non-parametric, using external resources and prompt engineering without altering the model. This section reviews key advances, focusing on fine-tuning, action decomposition, memory-driven strategies, and in-context learning.

(a) Parametric (b) Non-parametric

Figure 2: LLM/VLM as Agent

3.1 Parametric

Parametric LLM agents are decision-making models that fine-tune the internal parameters of LLMs using experience datasets, as illustrated in Figure 2a. This approach enables them to adapt their behaviour for specific tasks and environments, ensuring precise and context-aware decision-making. By leveraging RL techniques such as policy optimization, action decomposition strategies, and value-based methods, these agents dynamically adjust their actions to align with specific task objectives.

For instance, AGILE [Feng *et al.*, 2024] integrates memory, tools, and expert consultation within a modular framework, leveraging RL to enhance reasoning and decision-making, achieving notable advancements over existing mod-

els in complex tasks. It outperforms the state-of-the-art LLM in specialized quality control benchmarks, demonstrating improved accuracy and adaptability. Similarly, Retroformer [Yao *et al.*, 2024] uses policy gradient optimization to iteratively refine prompts based on environmental feedback, achieving higher success rates in multi-step tasks. On the other hand, TWOSOME [Tan *et al.*, 2024] improves sample efficiency and performance in interactive multi-step decision-making tasks by normalizing action probabilities and applying parameter-efficient fine-tuning to address alignment challenges between LLMs and dynamic environments. Other methods further enhance parametric agents through innovative mechanisms. For example, POAD [Wen *et al.*, 2024] decomposes actions into token-level decisions, addressing optimization complexity and enabling precise credit assignment in environments with large action spaces. GLAM [Carta *et al.*, 2023] introduces functional grounding in textual environments, leveraging online RL to align LLMs with spatial and navigation tasks through step-by-step interaction and iterative learning. In vision-language tasks, fine-tuning frameworks combine chain-of-thought reasoning with RL to enable agents to manage multimodal problems, demonstrating significantly improved visual-semantic understanding [Zhai *et al.*, 2024].

Collectively, these approaches demonstrate that parametric LLM agents using RL techniques, including policy optimization, action decomposition, and functional grounding, achieve superior adaptability, sample efficiency, and performance.

3.2 Non-parametric

Non-parametric LLM agents rely on the inherent reasoning and generalization capabilities of LLMs, as shown in Figure 2b, while keeping the LLM agent frozen and without altering its internal parameters. These agents leverage external resources and datasets to enrich the task context, and use prompt engineering techniques during inference to guide decision-making.

For example, ICPI [Brooks *et al.*, 2023] implements policy iteration in LLMs using in-context learning, where Q-values are computed via rollouts and iteratively refined. This approach, tested in six RL domains, demonstrates the potential of LLMs as both world models and policies, enabling scalable improvements without fine-tuning. Reflexion [Shinn *et al.*, 2023] introduces verbal reinforcement, where LLMs generate and store self-reflective feedback in an episodic memory buffer to improve decision-making. This method enhances long-horizon decision-making, multi-step reasoning, and code generation, achieving state-of-the-art accuracy in function synthesis and logical inference. Similarly, REMEMBERER [Zhang *et al.*, 2023a] incorporates persistent experience memory, allowing LLMs to learn from past successes and failures in interactive environments without modifying parameters. By integrating RL with experience memory, it improves adaptability and robustness in sequential reasoning and goal-oriented decision-making. Building on these ideas, ExpeL [Zhao *et al.*, 2024] introduces experiential learning, enabling LLMs to autonomously collect, abstract, and apply knowledge from past tasks. This method enhances sequential decision-making and transfer learning, offering a resource-efficient alternative to fine-tuning.

Beyond general decision-making, non-parametric LLM agents have also been explored in domain-specific applications, including robotic manipulation and strategic multi-agent collaboration. RLingua [Chen *et al.*, 2024] improves sample efficiency in RL for robotic manipulation by leveraging LLM-generated rule-based controllers as priors and integrating prior knowledge into policy learning through prompts. This approach enhances performance in sparse-reward tasks, achieving high success rates in both simulated and real-world environments with effective Sim2Real transfer. Werewolf [Xu *et al.*, 2024] combines LLM-driven action candidate generation with RL to mitigate intrinsic biases and enhance strategic decision-making. By integrating deductive reasoning and RL, this framework enables agents to achieve human-level performance in unbounded communication and decision spaces. Similarly, LangGround [Li *et al.*, 2024] aligns MARL agents' communication with human language by grounding it in synthetic data from embodied LLMs. This method facilitates zero-shot generalization in ad hoc teamwork, improving communication emergence, interpretability, and task performance with unseen teammates. These studies illustrate that non-parametric LLM agents can achieve state-of-the-art performance without requiring parameter updates.

3.3 Discussion

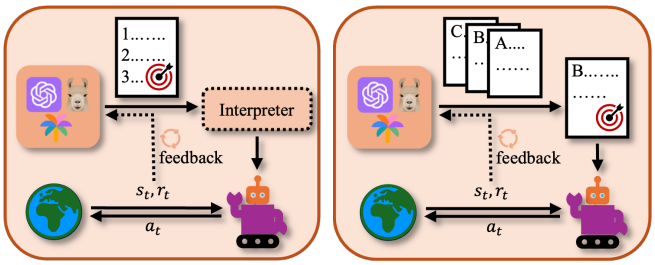
The integration of LLMs/VLMs as decision-making agents highlights the strengths and limitations of parametric and non-parametric approaches. Parametric agents excel in task-specific adaptability and alignment via fine-tuning and RL but face scalability and computational challenges in interactive environments. Non-parametric agents leverage in-context learning and memory-driven reasoning for generalization and scalability without fine-tuning but struggle with long-term planning and complex modelling. These paradigms complement each other, with parametric methods providing precision and non-parametric approaches ensuring efficiency. Hybrid frameworks combining lightweight fine-tuning with advanced memory mechanisms can enhance LLM agents' robustness and adaptability in complex environments.

4 LLM/VLM as Planner

With extensive knowledge and strong reasoning capabilities, FMs can generate high-level plans that address RL's struggles with complex multi-step tasks by decomposing them into sub-goals. Integrating FMs allows RL agents to focus on shorter-horizon control, improving sample efficiency. We consider approaches that use FMs for plan generation in RL, categorized as (1) comprehensive, requiring all sub-goals to be selected upfront, or (2) incremental, where sub-goals are generated over time.

4.1 Comprehensive Planning

FMs can generate a complete plan specifying sequential sub-goals for an RL agent to execute, as shown in Figure 3a. FMs use their knowledge during the planning process, breaking



(a) Comprehensive Planning (b) Incremental Planning

Figure 3: LM as Planner

down complex tasks into a sequence of achievable steps. RL agents thus do not need to learn complex tasks from scratch.

When bridging the natural language plan to executable actions taken by the low-level controller, an advantage of using FMs is that the output of FMs can be structured based on actual needs. For example, SayTap [Tang *et al.*, 2023] uses foot contact patterns as a compact interface between language instructions and low-level quadruped control. An LLM outputs textual binary signals defining each leg's contact pattern, which an RL policy is trained to follow. This demonstrates how high-level language commands can be translated into fine-grained control signals. In addition to simpler binary-based control, some tasks may benefit from a skill library. LMA3 [Colas *et al.*, 2023] uses an LLM to evaluate and validate an agent's performance on various goals, and then treats the shortest action sequence from each successful execution as a skill. LMA3 leverages this growing skill library to chain short sequences into larger plans for solving a complex goal. However, its reliance on previously discovered action sequences limits its generalization. In contrast, PSL [Dalal *et al.*, 2024] leverages a LLM to decompose a long-horizon natural language task into specially formatted language sub-goals. To control the robot's motion, each sub-goal consists of a sequence of target positions followed by a termination stage, which together guide the robot's RL policy in planning its movement. PSL removes the need for a pre-defined skill library and improves both learning efficiency and generalization.

An initial plan may not be perfect, and execution failures might occur partway through. Appropriate adjustments and modifications to the plan generated by FMs can improve the plan's correctness and the agent's performance. For example, Inner Monologue [Huang *et al.*, 2022] uses three types of feedback to update the plan. It collects binary feedback from a success detector after task accomplishment, visual feedback from a scene detector during execution, and it is allowed to request a human or other pre-trained models for feedback on the LLM planner's question during execution. This dynamic re-planning skill improves completion rate and flexibility. To avoid querying LLMs after each failed execution and reduce the querying cost of LLMs. LgTS [Shukla *et al.*, 2024] uses LLMs to generate multiple candidate sub-goal sequences before execution. It arranges them as a directed acyclic graph and uses an RL agent to explore the graph, learning the policy through a Teacher-Student learning strategy, improving sample efficiency.

4.2 Incremental Planning

Incremental planning, as illustrated in Figure 3b, is another way for FMs to guide the agent, providing step-by-step guidance for actions. Querying FMs at every step incurs high resource consumption costs; these approaches carefully determine when and how to query FMs at execution time.

For example, SayCan [Ichter *et al.*, 2022] generates multiple candidate sub-goals at each step and then estimates each sub-goal's likelihood to be part of an optimal trajectory. Combining sub-goals with these feasibility checks effectively grounds the LLM's plans in real-world constraints, helping the agent achieve its main goal. Similarly, LLM4Teach [Zhou *et al.*, 2024] provides the agent with a set of suggested actions to execute. Initially, the agent is trained to follow the guidance of an LLM closely, but as the agent learns over time, its dependence on the LLM's suggestions decreases, allowing the agent to make independent decisions.

Papers adopting incremental planning also improve the quality of these sub-goals through accumulating experience from past trajectories. For example, AdaRefiner [Zhang and Lu, 2024] enhances the agent's execution and understanding of LLM guidance by introducing a secondary LLM to evaluate the alignment of the agent's execution process and the guidance of the LLM. The environmental information, combined with a score that evaluates the comprehension of the provided guidance, is then used to fine-tune the primary LLM, enabling it to provide better guidance in subsequent iterations. Similarly, BOSS [Zhang *et al.*, 2023b] learns from past trajectories but eliminates the need for a critic LLM. Instead, the guidance LLM continuously accumulates new skills demonstrated by the agent and adds them to a skill library. While summarizing and analyzing experiences from past trajectories could improve planning ability, simulating future trajectories can also contribute to better decision-making.

Instead of using only natural language input, LLaRP [Szot *et al.*, 2024] integrates a frozen LLM with a pre-trained vision encoder to process textual instructions and egocentric visual frames. LLaRP trains a vision encoder and an action decoder using online RL, demonstrating strong robustness and generalization in new environments. A unique example is Text2Motion [Lin *et al.*, 2023], which combines both comprehensive and incremental planning, ensuring efficiency and correctness. Initially, Text2Motion employs an LLM to generate a comprehensive plan, encompassing all the steps for the agent to execute. If a planning failure occurs during execution, Text2Motion uses the LLM to generate the sub-goals incrementally.

4.3 Discussion

Although the planning capabilities of FMs remain highly debated [Aghzal *et al.*, 2025; Kambhampati *et al.*, 2024], their in-context common knowledge has nonetheless shown promise in decomposing complex tasks into simpler subtasks and aiding adaptation across diverse environments. FMs are particularly effective in human-centric environments, where

plans in natural language benefit from common-sense reasoning. Comprehensive planning can be more efficient but is riskier in dynamic environments, while incremental planning enables real-time feedback and adaptation, but increases the computational overhead. Trade-offs between the two methods may enable FMs to achieve better results at a manageable cost.

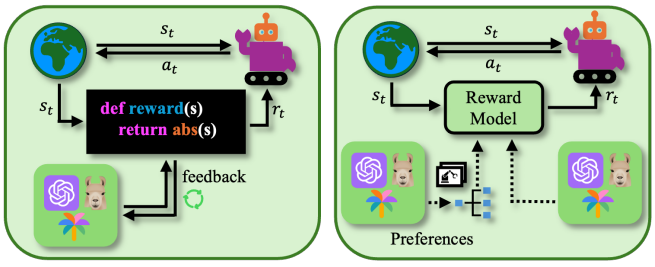
5 LLM/VLM as Reward

Designing effective reward signals remains a central challenge in RL, requiring domain knowledge and trial-and-error tuning. While methods such as preference-based learning, inverse RL, and labelled datasets help, they still rely heavily on human input. Recent advances leverage LLMs and VLMs to automate reward design by having them interpret textual descriptions and process visual inputs. These LLM/VLM as reward approaches generally fall into two categories: generating explicit reward functions, or serving as (or aiding the learning of) a reward model.

5.1 Reward Function

Leveraging LLMs to design reward functions may reduce human reward engineering effort, facilitate the discovery of novel reward components, and yield interpretable code. A common approach provides an environment abstraction as an initial context and then iteratively prompts an LLM to generate and improve reward functions using natural language (see Figure 4a). These benefits are especially valuable for high-dimensional or complex tasks.

Reward function approaches primarily differ in how they trigger refinements and the type of natural language feedback they incorporate. For example, in Text2Reward [Xie *et al.*, 2024], an LLM refines the reward function code until it executes successfully. After training an RL policy, non-expert users can observe the learned policy and provide linguistic feedback on suboptimal behaviours, prompting further LLM refinements to the reward function. Zeng *et al.* [2024] use an LLM to identify key behavioural features (to promote or discourage) and propose an initial reward function parameterization. The LLM iteratively refines this parameterization by ranking trajectories from executions of the trained policy, shaping the reward function toward desirable behaviours. Eureka [Ma *et al.*, 2024] uses an evolutionary search strategy. At each iteration, an LLM generates multiple candidate reward functions, the system trains a policy under each reward



(a) Reward Function (b) Reward Model

Figure 4: LLM/VLM as Reward

to measure task fitness, and the reward with the best fitness score is fed back (along with component-level statistics) for the LLM to refine in the next round. All three approaches can produce reward functions that match or surpass those designed by human experts, and are readily extended to novel tasks with minimal human intervention.

5.2 Reward Model

As illustrated in Figure 4b, FMs can specify reward models in two key ways. First, LLMs can serve as proxy reward models by mapping textual descriptions of desired behaviours directly to scalar rewards. Second, a separate reward model can be learned by leveraging LLMs or VLMs to incorporate preference feedback on agent trajectories or by combining textual instructions with visual observations in VLMs to produce more robust and visually grounded reward models.

Kwon *et al.* [2023] use an LLM as a proxy reward model, using natural language descriptions of desired behaviours and textual trajectory summaries to generate a binary reward signal that guides policy learning. Their simple approach achieves performance close to that of ground-truth reward while avoiding the large, curated preference datasets that conventional reinforcement learning from human feedback (RLHF) methods typically rely on. PREDILECT [Holk *et al.*, 2024] builds on preference-based RL, allowing human raters to specify both their preferred trajectory and the reasons for their choice. Using these explanations, an LLM extracts key trajectory subsequences and incorporates them into the reward-learning objective via regularization, giving more weight to segments marked as "good" or "bad." This targeted influence mitigates causal confusion by directing the model's attention to the true causal factors underlying human preferences. ELLM [Du *et al.*, 2023] improves exploration in RL by prompting an LLM with a textual "caption" of the agent's state to generate sub-goals. The agent is rewarded for achieving these sub-goals via a semantic similarity measure between its transition caption (action and resulting state) and the suggested sub-goal, with a novelty bias that rewards each sub-goal only once per episode. ELLM shifts naive novelty-driven exploration toward semantically guided skill discovery, yielding more human-like behaviours and faster task learning than intrinsic motivation baselines and related novelty-only methods.

Text-based reward design often fails in visually complex tasks, where nuanced details cannot be fully expressed in words. RL-VLM-F [Wang *et al.*, 2024] overcomes this limitation by leveraging a large VLM without requiring any human annotation, using it to rank pairs of images (observations) based on their alignment with a natural language task description. These pairwise preferences train a visually-grounded reward model, enabling robust reward design for tasks with intricate visual observations. VLM-RM [Rocamonde *et al.*, 2024] and MineCLIP [Fan *et al.*, 2022] both leverage a large VLM (CLIP) to scale RL to tasks that are not easily specified using engineered reward functions, but are easily described in natural language. VLM-RM targets continuous control problems by computing a direct scalar reward based on the cosine similarity between a textual goal embedding—adjusted by subtracting a "baseline prompt" embedding to reduce in-

terference from irrelevant features—and the agent's visual observation embedding. Notably, VLM-RM performance improves when environments are enhanced with more realistic visuals, better aligning with CLIP's training distribution. MineCLIP similarly builds on CLIP but targets Minecraft's open-ended environment, fine-tuning on 16-frame YouTube video segments paired with time-aligned text, yielding a dense reward signal that correlates the agent's recent frames with a free-form textual goal.

5.3 Discussion

LLM/VLM as reward approaches automate the generation of reward functions by translating textual descriptions into rewards for RL agents. Their strong performance—often matching or surpassing human-engineered and ground-truth rewards—indicates that natural language effectively encodes and guides reward design for complex tasks. However, these approaches may be very sensitive to prompt design, prone to hallucinations, or omit critical details. They also typically rely on simplified abstractions that fail to capture real-world complexity, raising concerns about scalability and reliability in more realistic settings.

6 Future Directions

Recent methods integrating LLMs and VLMs into RL have demonstrated strong capabilities, but key limitations remain under-explored—particularly those hindering real-world applicability. Many approaches overlook challenges such as hallucinations, implicit biases, and high computational costs [Xu *et al.*, 2024; Zeng *et al.*, 2024], which can undermine policy learning and robustness. These concerns are especially pressing in robotics and human-centred domains, where latency, sample efficiency, and hardware constraints are critical.

In particular, the inference overhead of querying LLMs at every time step [Shinn *et al.*, 2023] or generating synthetic rollouts [Pang *et al.*, 2024] poses significant scalability concerns. While some methods improve efficiency—e.g., by reducing reliance on LLMs [Zhou *et al.*, 2024]—trade-offs between performance and resource use are rarely analyzed systematically. To support broader deployment of FM integrated with RL systems, future research must incorporate cost-aware metrics and benchmarks, while also addressing key challenges that we outline next.

6.1 Grounding

LLMs demonstrate strong capabilities for generating high-level plans, but their plans may not be executable for embodied agents such as robots [Ichter *et al.*, 2022; Dalal *et al.*, 2024]. Current works solving the grounding problem by applying a bridging layer or verification module between the high-level plan and the low-level controller [Dalal *et al.*, 2024; Huang *et al.*, 2022] or by leveraging the value function to ground the action [Ichter *et al.*, 2022]. However, these methods share similar disadvantages: the external knowledge they rely on might introduce biases that hinder performance. Another approach is to design the generated plan's structure to fit the real-world requirements [Tang *et al.*, 2023], but this may be difficult to generalize to novel environments.

6.2 Inherent Bias

LLMs and VLMs exhibit intrinsic biases rooted in their data sources, training procedures, and architectures. Biases in FMs can lead to sub-optimal behaviour—biased actions drive execution; flawed plans skew exploration and learning; and distorted reward signals misdirect policy updates. For example, an LLM can identify the Rock-Paper-Scissors Nash equilibrium—playing each action equally—yet still favour playing Rock [Xu *et al.*, 2024]. Few works target de-biasing, using self-consistency and population-based training [Xu *et al.*, 2024], but only partially address the issue. Meanwhile, implicit refinements of an LLM's outputs, through action values or environment feedback, have shown promise, but remain largely confined to high-level task planning [Huang *et al.*, 2022; Ichter *et al.*, 2022].

6.3 Representation

Integrating LLMs into RL often requires two-way translation between numeric signals—such as raw sensor data and actions—and sequences of textual tokens, a process that loses the nuanced semantic information required for precise control [Du *et al.*, 2023; Hu and Sadigh, 2023]. Some methods incorporate external translator modules that narrate raw observations and decode textual commands into the environment's action space [Brooks *et al.*, 2023; Zhang *et al.*, 2024]. KALM [Pang *et al.*, 2024] overcomes these constraints by replacing the LLM embedding and output layers with multilayer perceptron modules that, once fine-tuned, enable bidirectional translation between language goals and numeric trajectories. One promising direction is to explore novel modifications to LLM architectures that fuse raw sensor data with language into unified representations, or to leverage multimodal LLMs and VLMs that preserve detailed sensory features while retaining language-based reasoning—enhancements that may speed up learning and stabilize performance, while also enabling agents to tackle more complex tasks demanding richer representations.

6.4 Action Advice

A human or agent can provide action-level guidance to an RL agent, significantly boosting learning speed [Torrey and Taylor, 2013]. LLMs and VLMs could potentially leverage their background knowledge to serve as non-human teachers with better generalization and more learning improvement than simple RL-based teaching agents. Note that prior work shows that such teachers can improve an agent's learning even if they are imperfect [Torrey and Taylor, 2013; Icarte *et al.*, 2018].

7 Conclusion

Research integrating FMs with RL is rapidly expanding. This survey introduces a taxonomy categorizing FM-based methods into agent, planner, and reward roles. We review studies in each role, highlighting how FMs can serve as parametric or non-parametric policies, generate comprehensive or incremental plans, or define rewards through a reward function or model. We discuss current limitations and propose future directions, aiming to clarify advancements and challenges in leveraging FMs for RL to help inspire future innovation.

Acknowledgements

Part of this work has taken place in the Intelligent Robot Learning Lab, which is supported in part by research grants from Alberta Innovates; the Alberta Machine Intelligence Institute (Amii); a Canada CIFAR AI Chair, Amii; the Digital Research Alliance of Canada; Mitacs; and NSERC.

Contribution Statement

Masoud Jafaripour and Yingyue Cao contributed equally.

References

[Aghzal *et al.*, 2025] Mohamed Aghzal, Erion Plaku, et al. A survey on large language models for automated planning. *CoRR*, abs/2502.12435, 2025.

[Brooks *et al.*, 2023] Ethan Brooks, Logan Walls, et al. Large language models can implement policy iteration. In *NeurIPS*, 2023.

[Brown *et al.*, 2020] Tom B. Brown, Benjamin Mann, et al. Language models are few-shot learners. In *NeurIPS*, 2020.

[Cao *et al.*, 2024] Yuji Cao, Huan Zhao, et al. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2024.

[Carta *et al.*, 2023] Thomas Carta, Clément Romac, et al. Grounding large language models in interactive environments with online reinforcement learning. In *ICML*, 2023.

[Chen *et al.*, 2024] Liangliang Chen, Yutian Lei, et al. Rlingua: Improving reinforcement learning sample efficiency in robotic manipulations with large language models. *IEEE Robotics and Automation Letters*, 9(7):6075–6082, 2024.

[Colas *et al.*, 2023] Cédric Colas, Laetitia Teodorescu, et al. Augmenting autotelic agents with large language models. In *CoLLAs*, 2023.

[Dalal *et al.*, 2024] Murtaza Dalal, Tarun Chiruvolu, et al. Plan-seq-learn: Language model guided rl for solving long horizon robotics tasks. In *ICLR*, 2024.

[Driess *et al.*, 2023] Danny Driess, Fei Xia, et al. Palm-e: An embodied multimodal language model. In *ICML*, 2023.

[Du *et al.*, 2023] Yuqing Du, Olivia Watkins, et al. Guiding pretraining in reinforcement learning with large language models. In *ICML*, 2023.

[Fan *et al.*, 2022] Linxi Fan, Guanzhi Wang, et al. Minedojo: Building open-ended embodied agents with internet-scale knowledge. In *NeurIPS*, 2022.

[Feng *et al.*, 2024] Peiyuan Feng, Yichen He, et al. Agile: A novel reinforcement learning framework of llm agents. In *NeurIPS*, 2024.

[Fu *et al.*, 2024] Haotian Fu, Pratyusha Sharma, et al. Language-guided skill learning with temporal variational inference. In *ICML*, 2024.

[Holk *et al.*, 2024] Simon Holk, Daniel Marta, et al. Predilect: Preferences delineated with zero-shot language-based reasoning in reinforcement learning. In *HRI*, 2024.

[Hu and Sadigh, 2023] Hengyuan Hu and Dorsa Sadigh. Language instructed reinforcement learning for human-ai coordination. In *ICML*, 2023.

[Hu *et al.*, 2024] Bin Hu, Chenyang Zhao, et al. Enabling intelligent interactions between an agent and an llm: A reinforcement learning approach. *RLJ*, 3:1289–1305, 2024.

[Huang *et al.*, 2022] Wenlong Huang, Fei Xia, et al. Inner monologue: Embodied reasoning through planning with language models. In *CoRL*, 2022.

[Icarte *et al.*, 2018] Rodrigo Toro Icarte, Toryn Q. Klassen, et al. Advice-based exploration in model-based reinforcement learning. In *Canadian AI*, 2018.

[Ichter *et al.*, 2022] Brian Ichter, Anthony Brohan, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *CoRL*, 2022.

[Kambhampati *et al.*, 2024] Subbarao Kambhampati, Karthik Valmeekam, et al. Position: Llms can't plan, but can help planning in llm-modulo frameworks. In *ICML*, 2024.

[Klissarov *et al.*, 2025] Martin Klissarov, Mikael Henaff, et al. Maestromotif: Skill design from artificial intelligence feedback. In *ICLR*, 2025.

[Kwon *et al.*, 2023] Minae Kwon, Sang Michael Xie, et al. Reward design with language models. In *ICLR*, 2023.

[Lai and Zang, 2024] Jinbang Lai and Zhaoxiang Zang. Sample trajectory selection method based on large language model in reinforcement learning. *IEEE Access*, 12:61877–61885, 2024.

[Li *et al.*, 2024] Huao Li, Hossein Nourkhiz Mahjoub, et al. Language grounded multi-agent reinforcement learning with human-interpretable communication. In *NeurIPS*, 2024.

[Lin *et al.*, 2023] Kevin Lin, Christopher Agia, et al. Text2motion: From natural language instructions to feasible plans. *Autonomous Robots*, 47(8):1345–1365, 2023.

[Ma *et al.*, 2024] Yecheng Jason Ma, William Liang, et al. Eureka: Human-level reward design via coding large language models. In *ICLR*, 2024.

[Pang *et al.*, 2024] Jing-Cheng Pang, Si-Hang Yang, et al. Kalm: Knowledgeable agents by offline reinforcement learning from large language model rollouts. In *NeurIPS*, 2024.

[Pternea *et al.*, 2024] Moschoula Pternea, Prerna Singh, et al. The rl/llm taxonomy tree: Reviewing synergies between reinforcement learning and large language models. *Journal of Artificial Intelligence Research*, 80:1525–1573, 2024.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[Rocamonde *et al.*, 2024] Juan Rocamonde, Victoriano Montesinos, et al. Vision-language models are zero-shot reward models for reinforcement learning. In *ICLR*, 2024.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, et al. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[Shinn *et al.*, 2023] Noah Shinn, Federico Cassano, et al. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*, 2023.

[Shukla *et al.*, 2024] Yash Shukla, Wenchang Gao, et al. Lgts: Dynamic task sampling using llm-generated subgoals for reinforcement learning agents. In *AAMAS*, 2024.

[Sutton and Barto, 2018] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[Szot *et al.*, 2024] Andrew Szot, Max Schwarzer, et al. Large language models as generalizable policies for embodied tasks. In *ICLR*, 2024.

[Tan *et al.*, 2024] Weihao Tan, Wentao Zhang, et al. True knowledge comes from practice: Aligning large language models with embodied environments via reinforcement learning. In *ICLR*, 2024.

[Tang *et al.*, 2023] Yujin Tang, Wenhao Yu, et al. Saytap: Language to quadrupedal locomotion. In *CoRL*, 2023.

[Thirunavukarasu *et al.*, 2023] Arun James Thirunavukarasu, Darren Shu Jeng Ting, et al. Large language models in medicine. *Nature Medicine*, 29(8):1930–1940, 2023.

[Torrey and Taylor, 2013] Lisa Torrey and Matthew E. Taylor. Teaching on a budget: Agents advising agents in reinforcement learning. In *AAMAS*, 2013.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, et al. Attention is all you need. In *NeurIPS*, 2017.

[Vinyals *et al.*, 2019] Oriol Vinyals, Igor Babuschkin, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[Wang *et al.*, 2024] Yufei Wang, Zhanyi Sun, et al. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. In *ICML*, 2024.

[Wen *et al.*, 2024] Muning Wen, Ziyu Wan, et al. Reinforcing llm agents via policy optimization with action decomposition. In *NeurIPS*, 2024.

[Xie *et al.*, 2024] Tianbao Xie, Siheng Zhao, et al. Text2reward: Automated dense reward function generation for reinforcement learning. In *ICLR*, 2024.

[Xu *et al.*, 2024] Zelai Xu, Chao Yu, et al. Language agents with reinforcement learning for strategic play in the werewolf game. In *ICML*, 2024.

[Yao *et al.*, 2024] Weiran Yao, Shelby Heinecke, et al. Retroformer: Retrospective large language agents with policy gradient optimization. In *ICLR*, 2024.

[Zeng *et al.*, 2024] Yuwei Zeng, Yao Mu, et al. Learning reward for robot skills using large language models via self-alignment. In *ICML*, 2024.

[Zhai *et al.*, 2024] Yuexiang Zhai, Hao Bai, et al. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. In *NeurIPS*, 2024.

[Zhang and Lu, 2024] Wanpeng Zhang and Zongqing Lu. Adarefiner: Refining decisions of language models with adaptive feedback. In *NAACL*, 2024.

[Zhang *et al.*, 2023a] Danyang Zhang, Lu Chen, et al. Large language models are semi-parametric reinforcement learning agents. In *NeurIPS*, 2023.

[Zhang *et al.*, 2023b] Jesse Zhang, Jiahui Zhang, et al. Bootstrap your own skills: Learning to solve new tasks with large language model guidance. In *CoRL*, 2023.

[Zhang *et al.*, 2024] Ceyao Zhang, Kaijie Yang, et al. Proagent: Building proactive cooperative agents with large language models. In *AAAI*, 2024.

[Zhao *et al.*, 2024] Andrew Zhao, Daniel Huang, et al. Expel: Llm agents are experiential learners. In *AAAI*, 2024.

[Zhou *et al.*, 2024] Zihao Zhou, Bin Hu, et al. Large language model as a policy teacher for training reinforcement learning agents. In *IJCAI*, 2024.