Information Fusion 129 (2026) 104003

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full Length Article

The duality of generative AI and reinforcement learning in robotics: A review

Angelo Moroncelli [a,*], Vishal Soni [a,b], Marco Forgione [a], Dario Piga [a], Blerina Spahiu [c], Loris Roveda [a,d,*]

[a] Department of Innovative Technologies, IDSIA, University of Applied Science and Arts of Southern Switzerland, Lugano, 6900, Ticino, Switzerland
[b] Department of Mechatronics and Automation Engineering, Indian Institute of Information Technology Bhagalpur, Bhagalpur, 813210, Bihar, India
[c] Department of Informatics, Systems and Communication, Università di Milano Bicocca, Milano, 20126, Lombardy, Italy
[d] Mechanical Department, Politecnico di Milano, Milano, 20156, Lombardy, Italy

ARTICLE INFO

Keywords:
Robotics
Generative AI
Foundation model
Reinforcement learning
Data fusion
Review

ABSTRACT

Recently, generative AI and reinforcement learning (RL) have been redefining what is possible for AI agents that take information flows as input and produce intelligent behavior. As a result, we are seeing similar advancements in embodied AI and robotics for control policy generation. Our review paper examines the integration of generative AI models with RL to advance robotics. Our primary focus is on the duality between generative AI and RL for robotics downstream tasks. Specifically, we investigate: (1) The role of prominent generative AI tools as modular priors for multi-modal input fusion in RL tasks. (2) How RL can train, fine-tune and distill generative models for policy generation, such as VLA models, similarly to RL applications in large language models. We then propose a new taxonomy based on a considerable amount of selected papers.

Lastly, we identify open challenges accounting for model scalability, adaptation and grounding, giving recommendations and insights on future research directions. We reflect on which generative AI models best fit the RL tasks and why. On the other side, we reflect on important issues inherent to RL-enhanced generative policies, such as safety concerns and failure modes, and what are the limitations of current methods. A curated collection of relevant research papers is maintained on our GitHub repository, serving as a resource for ongoing research and development in this field.

1. Introduction

Two main pursuits of embodied intelligence and robotics are achieving physical grounding [1], the capability of robots to sense, comprehend, and interact efficiently with the physical environment, and human-level reasoning on abstract concepts [1–4], which remain a central goal of artificial intelligence (AI) research in general. Recently, the convergence of two powerful paradigms, generative AI and Reinforcement Learning (RL), has shown significant promise in advancing this objective [5–11]. On the one hand, tools from generative AI, including Large Language Models (LLMs), multi-modal foundation models [12] such as Vision-Language Models (VLMs), and world or video prediction models, have excelled in processing and generating diverse data types such as text, code, and images [13–16]. Trained on vast multi-modal datasets, these models encapsulate rich, generalizable knowledge representations [17–19]. Moreover, diffusion models excel in generative capabilities and robust training processes for downstream tasks such as state representation and policy generation [20,21]. RL, on the other hand, offers a framework for agents to learn optimal behaviors through interaction with their environment [22,23].

The potential synergy between generative AI and RL is particularly compelling in the context of robotics. Generative models can serve as robust priors for information fusion in RL agents—*i.e.*, fusing information flows from different sources to produce a robot control policy—providing extensive world knowledge, grounding language understanding, and generating rich behaviors [7,8,24]. Conversely, RL can "embody" generative AI models used as generative policies for robotics, usually trained through pure Imitation Learning (IL) [25], allowing them to interact with and learn from dynamic physical environments and suboptimal data [26]. This integration could lead to robotic systems with enhanced adaptability, generalization, and overall intelligence through feature alignment with experience [27], similarly to RL from human feedback training for LLMs [28].

* Corresponding authors.
E-mail addresses: angelo.moroncelli@supsi.ch (A. Moroncelli), loris.roveda@supsi.ch;loris.roveda@polimi.it (L. Roveda).

https://doi.org/10.1016/j.inffus.2025.104003
Received 30 June 2025; Received in revised form 3 November 2025; Accepted 24 November 2025
Available online 29 November 2025

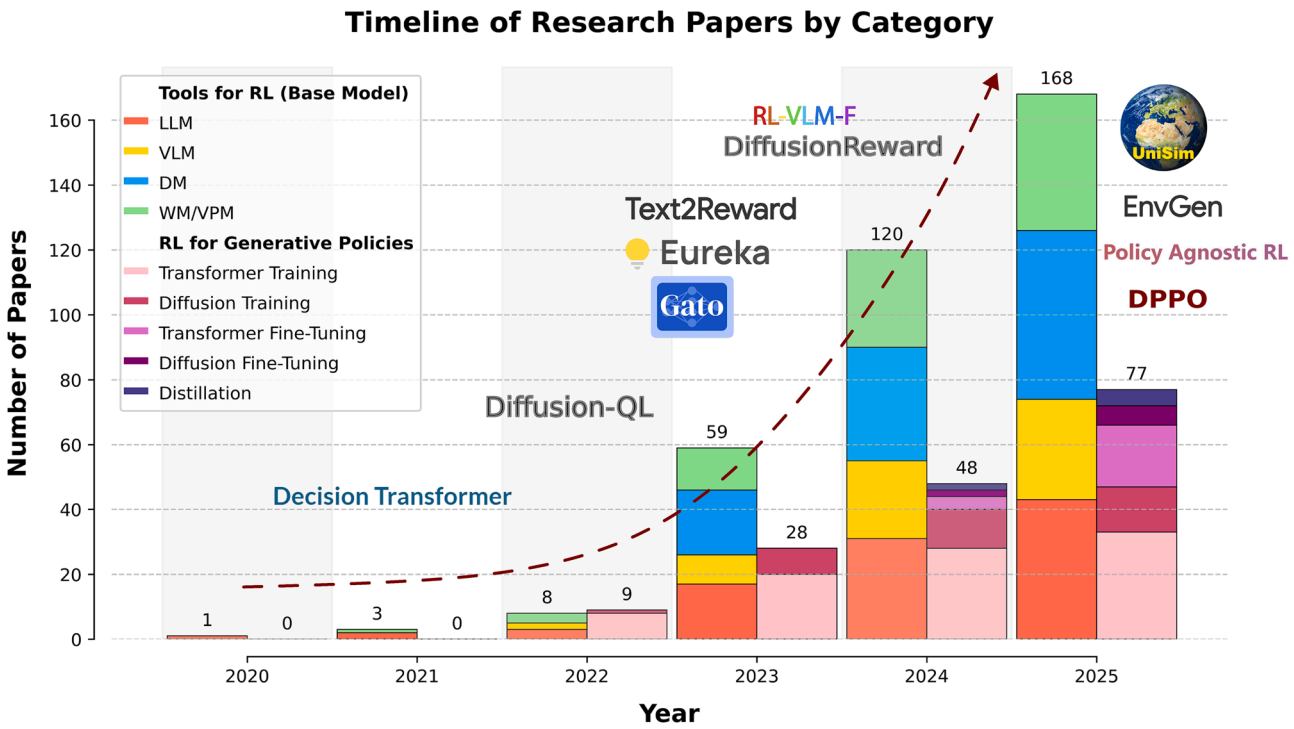


Fig. 1. Trends in generative AI and RL integration for robotics. The figure illustrates the number of papers published each year integrating both generative AI and RL in robotics, categorized by the type of model employed.

Foundation models, in particular, have demonstrated exceptional proficiency in processing information for downstream robotics tasks [29–31]. Additionally, specialized foundation models for robotics exist [24]. Foundation models have also been adopted in the field of learning and control of dynamical systems, where robotics represents a key area of application. Transformer-based pre-trained models have been proposed [32,33] for zero-shot prediction of outputs from a class of dynamical systems in response to any query input sequence, and for in-context state estimation [34]. These advancements highlight the potential of foundation models to introduce innovative approaches and paradigms to traditional dynamical control systems, facilitating a shift towards data-driven estimation and control synthesis for classes of dynamical systems, rather than for single, specific systems.

However, despite significant progress in foundation models, their application with RL in robotics remains underexplored. Recent developments suggest this promising union to enhance robotic learning and generalization, yet challenges persist, especially for real-world robotic applications [35].

1.1. Scope and contribution

Despite rapid advancements in both generative AI and RL [23,36–38], there is a noticeable lack of comprehensive studies systematically exploring their integration in robotics (see Table 1). Our review addresses this gap by providing an in-depth analysis of current research trends at the intersection of generative AI and RL in robotics. Specifically, it examines Transformer- and Diffusion-based generative AI models (LLMs, VLMs, diffusion models (DMs), world models (WMs) and video prediction models (VPMs)) currently used to enhance RL, considering the diverse data modalities, roles, and applications (see Sections 3–5).

Fig. 1 illustrates the growing adoption of generative AI models for RL. In Fig. 2, the heatmap highlights the notable rise in the use of diffusion models as tools for RL in robotics from 2023 up to November 2025 (dark purple block) and the recent interest in investigating the use of RL for fine-tuning Transformer-based models (+15 new papers in 2025). Notice the recent introduction of a few seminal works in RL-based fine-tuning and distillation of generative policies in general. Additionally, particular attention is given to the unique relationship between generative policies, which are a specific and relatively narrow type of generative models for robotic action generation, and their integration with RL (see Sections 6–8), with insights on their good fitting with learning-based control techniques for grounding into downstream robotic tasks (see Section 10). To the best of our knowledge, RL fine-tuning of generalist generative policies has also not been classified in previous surveys [24,39–41].

The main **contributions** of our work are:

1. A comprehensive review of the intersection between Transformer- and Diffusion-based generative models and RL for robotics.
2. The first, to the best of our knowledge, dual-perspective analysis of how generative AI tools improve RL and RL improves generative policy models for robotics.
3. The identification of best practices and challenges when using generative AI models as tools for RL.
4. A detailed classification of RL-based training, fine-tuning and distillation methods for generalist generative policies.
5. The identification of three new research directions integrating generative AI and RL to enhance robotics.
6. A unified new taxonomy and continuously updated repository [1] for tracking progress in this field.

The remainder of our paper is structured as follows: Section 2 presents our taxonomy based on the duality between generative AI transformer- and diffusion-based tools and RL. Section 3 (*Base Model*), Section 4 (*Modality*) and Section 5 (*Task*) investigate deeper the dimension of generative AI models used as modular tools for the RL training loop, analyzing *Generative Tools for RL*. On the other hand, Section 6 explores *RL-Based Training* of generative policies with consequent *RL-Based Fine-Tuning* for downstream tasks (Section 7) and *Model Distillation*

[1] https://github.com/clmoro/Robotics-RL-FMs-Integration

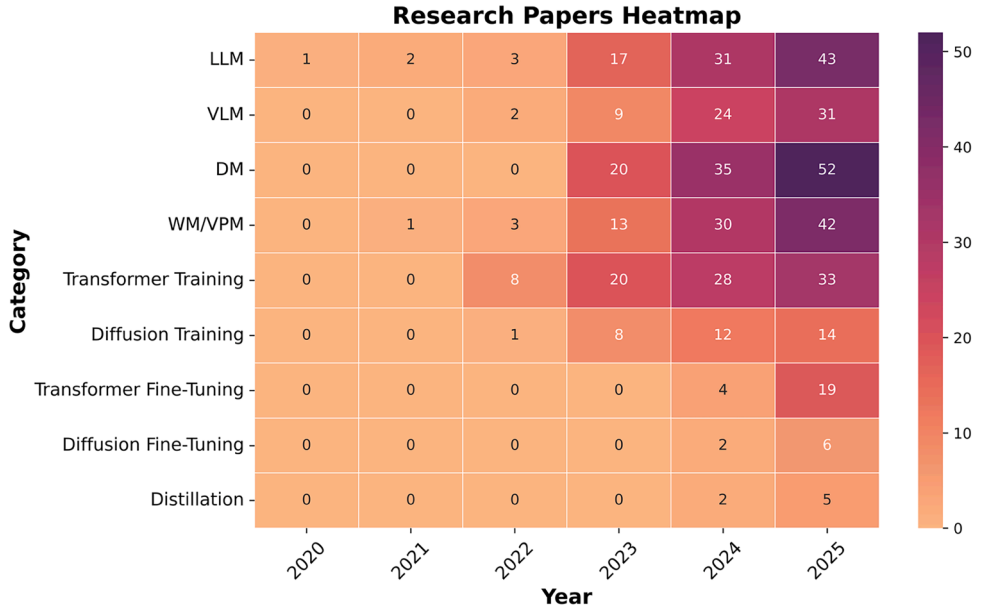


Fig. 2. Published papers heatmap.

Table 1
Comparison of survey papers across robotics learning categories and methods: *Foundation Models for Robotics* (**RFM**), *Generative Tools for RL* (**Tools for RL**) and *RL for Generative Policies* (**RL Policies**). ⋆ means that the survey falls into that category but it does not primarily focus on its transversal analysis (*e.g.*, general RFM aspects or RL perspectives); instead, it concentrates on a specific subtopic.

Survey	RFM	Tools for RL				RL Policies	
		LLM	VLM	DM	WM/VPM	Train	Fine-Tune
Hu et al. [39]	✓	⋆	⋆			✓	
Xiao et al. [40]	✓	⋆				⋆	
Firoozi et al. [24]	✓	⋆	⋆			✓	
Zhou et al. [41]	✓	⋆				⋆	
Wang et al. [42]	✓					⋆	
Cao et al. [43]	⋆	✓					
Zhu et al. [10]	⋆			✓			
Ours	⋆	✓	✓	✓	✓	✓	✓

(Section 8), looking into *RL for Generative Policies*. Section 9 discusses challenges. We conclude with Section 10, suggesting our perspective on possible future research directions based on our findings.

Related surveys. Recent reviews have examined either foundation models or RL for robotics in isolation [24,39–41], whereas only a few have considered specific aspects of their integration [10,43]. As summarized in Table 1, the recent surge of interest in foundation models has led most surveys to focus on the development of robotic foundation models—that is, the use of large generative backbones to train robotic policies. Typically, these works view foundation models as modular priors enabling multi-modal fusion within RL pipelines. Only a limited number of surveys [24,39] have briefly addressed the use of RL to train generative models for robotics. In contrast, this work explicitly conceptualizes this intersection through a *dual-perspective taxonomy* that formalizes the two complementary point of view, showing how the two paradigms inform and strengthen each other.

1.2. Methodology

We conducted this comprehensive review on the integration of transformer- and diffusion-based models with RL because these are two of the most rapidly emerging research areas in robotics learning. As noted in the previous section, we have observed a significant increase in the number of published papers on these topics in recent years, but especially, at their intersection. Given the exponential growth of research in generative AI for robotics, we chose to focus our review specifically on transformer- and diffusion-based models. To compile our dataset, we searched for relevant papers published between 2019, when we identified the first preliminary works integrating transformers with RL for robotics applications, and November 2025. We used keyword-based searches to filter papers aligned with the scope of our review, excluding those that did not meet our criteria, following the widely recognized PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) methodology[2]. In the identification phase, we selected 15 relevant keywords to identify papers relevant to our review. We used combinations of terms such as RL with foundation models, LLM, VLM, diffusion models, world models, and video prediction models for filtering works on *Generative Tools for RL*. For *RL for Generative Policies*, we combined RL with transformer, diffusion, flow matching, generative policy, pre-training, fine-tuning, and distillation. Our search spanned multiple platforms, including Scopus, DBLP, IEEE Xplore, Google Scholar, and ArXiv, as well as references found in other related surveys listed in Table 1. In the screening phase, two researchers manually reviewed the abstracts of the identified papers to ensure their relevance to the robotics domain. During the selection stage, 245 papers that strictly met the inclusion criteria were chosen for analysis. To structure our taxonomy, we identified relevant categories based on two major dimensions of our study. Papers included in our review were required to be directly related to robotics and RL, published in English, and ideally peer-reviewed. However, given the fast-paced nature of this field, many relevant works have been published on preprint servers such as ArXiv and have yet to undergo formal peer review for top-tier journals or conferences.

2. Taxonomy

In our review, we analyze how prominent generative AI tools and RL converge to enhance control policies for robots, specifically, action reference signals in the robot's Cartesian operational space. To achieve this, we propose a new taxonomy that categorizes the 169 papers we

[2] PRISMA: Preferred Reporting Items for Systematic Reviews and Meta-Analyses

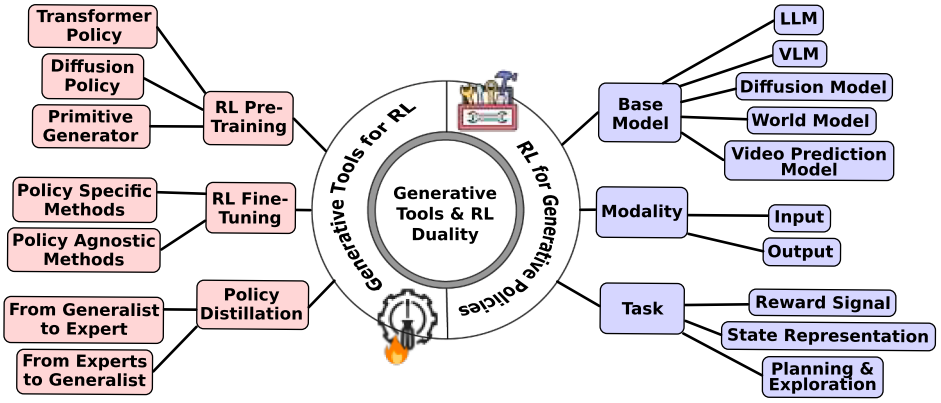


Fig. 3. Taxonomy. A taxonomy of the generative AI tools and RL integration for robotics.

examined, tracing the evolution of generative AI and RL integration for multi-modal data fusion in robotics policy generation. Our taxonomy (see Fig. 3 for an abstract representation) primarily explores the duality between *Generative AI tools for RL* and *RL for Generative Policies*. We chose to analyze only papers focusing on specific generative models—particularly modern architectures widely adopted in recent works for policy generation, primarily based on transformers or diffusion models. We further classify the approaches along six principal dimensions based on orthogonal features of the works. For *Generative Tools for RL*, we categorize papers based on their underlying model architecture, which we refer to as the **Base Model**; the input and output modalities, referred to as **Modality**; and finally, the aim of the RL process, which we call the **Task**. For *RL for Generative Policies*, we analyze research along: pre-training strategies for different policy architectures, which we refer to as **RL pre-training**; methods for adapting policies through interaction, referred to as **RL fine-tuning**; and the transfer or compression of policies, which we call **Policy Distillation**. We chose these categories to reflect two clear and growing trends in recent literature, that we depict in Fig. 4: the increasing use of generative models as tools within RL pipelines (blue outline), and the rise of RL techniques for training and refining generative policies (red outline), especially in robotic manipulation. Model architecture, modality, and task alignment are now central to foundation model choice [44], while RL techniques like fine-tuning and policy distillation are increasingly used to adapt generative policies [45]. Our taxonomy captures this duality by focusing on the most recurrent and emphasized dimensions in recent works.

Dual-perspective taxonomy. The dual-perspective taxonomy is adopted to provide a more structured and comprehensive view of the research landscape. Unlike traditional frameworks, which classify studies only by method, task, or model type, a dual-perspective taxonomy captures the reciprocal relationships between interconnected areas of research. This approach makes it easier to compare works across complementary directions, trace how developments in one branch influence the other, and identify research gaps, maturity levels, and emerging trends. This perspective is particularly important in the robotics field, where generative AI and RL are evolving rapidly and increasingly converge to build complex robotic systems, yet a clear analytical framework for understanding their interaction is still lacking.

2.1. Generative tools for RL

Generative tools for RL explores how selected generative AI architectures can be integrated into the RL training loop (see upper portion of Fig. 4). We analyze prior work on leveraging generative and foundation models to enhance robotics, focusing on Transformer and Diffusion backbones. **"As tools"** highlights that pre-trained foundation models (like LLMs) are not being retrained end-to-end with the RL agent, but are instead leveraged in a modular way—as plug-and-play components that provide capabilities (such as understanding or generating specific modalities) that the RL agent can use during training or decision making. Their in-context learning ability allows them to flexibly adapt to tasks without needing full retraining, making them useful tools for enhancing RL agents. Our analysis is structured along three key secondary dimensions: (i) *Base Model*, (ii) *Modality*, and (iii) *Task*.

Base model

Base model classifies the models based on their underlying architecture, which directly influences their capabilities. Different architectures process and generate various types of data, shaping how they interact with RL. We analyze how deeply each base model can be incorporated into the RL framework based on its structural characteristics. Furthermore, for each model, approaches are analyzed along with (i) the framework names, (ii) code availability, (iii) the data used for training, and (iv) whether the pipeline is in simulation or the real world. Under this dimension, we consider the following models: *LLMs*, *VLMs*, *Diffusion Models*, *World Models*, and *Video Prediction Models*.

Modality

Under the *Modality* dimension, we analyze approaches based on their *Input* and *Output* modalities, such as text, images, trajectories, or low-level sensory signals. These modalities determine how generative AI integrates with RL by tackling input data interpretation and representation. Understanding these modalities helps assess the suitability of different generative AI tools for various RL applications.

Task

Generative AI models often serve as powerful priors within the RL training loop, as they are typically pre-trained modules that enhance specific aspects of learning. In this section, we delve deeper into how different works have employed generative models to address key *Tasks* in the RL training loop such as: *Reward Signal* generation, *State Representation* and *Planning & Exploration*. The upper portion of Fig. 4 illustrates scenarios where various generative AI tools enhance RL tasks. As an example, LLMs support symbolic reasoning for rewards or plan generation based on task descriptions, while VLMs contribute to reward augmentation or plan generation through scene understanding.

2.2. RL for generative policies

The second primary dimension of our taxonomy examines RL methods used to train generative models, offering a complementary perspective to *Generative AI Tools for RL*. Here, we analyze works that employ RL-based approaches to pre-train, fine-tune, or distill generative policies—where RL is used directly to optimize models for action generation. We organize our discussion along three secondary dimensions,

which we refer to as: (i) *RL-Based Pre-Training*, (ii) *RL-Based Fine-Tuning*, and (iii) *Policy Distillation* (see lower part of Fig. 4).

RL-based pre-training

We survey various RL methods used to pre-train Transformer- and Diffusion-based policy backbones, enabling generalist generative policies that can process complex instructions; we divide them in *Transformer Policy* and *Diffusion Policy* respectively. Alternatively, RL can be used to train any set of simpler, task-specific policies for low-level control. These policies serve as a library of primitives (*Primitive Generator*), which can then be orchestrated by a foundation model-based planner for more complex behaviors.

RL-based fine-tuning

Fine-tuning pre-trained policies is a standard approach to adapting models to new tasks or datasets. We classify fine-tuning methods for generative policies based on their architecture and policy size—we call them *Policy Specific Methods*. Additionally, we discuss emerging policy-agnostic fine-tuning techniques that aim to improve generalization across different generative architectures—called *Policy Agnostic Methods*.

Policy distillation

Beyond policy fine-tuning, RL includes the concept of *Policy Distillation*, which refers to transferring knowledge and learned skills from a "teacher" policy to a "student" policy [46]. Recently, researchers have begun investigating how to distill knowledge from large pre-trained Vision-Language-Action (VLA) models into smaller and efficient RL-based expert policies (*From Generalist to Expert*), as well as how to use RL pre-trained single-task policies to inject more knowledge into VLA models (*From Experts to Generalist*).

Notation. Let $s_t \in \mathcal{S}$ denote a state, $a_t \in \mathcal{A}$ an action, and $\pi_\theta(a_t \mid s_t)$ a policy parameterized by θ. The immediate reward is $r_t = R(s_t, a_t)$ and the cumulative return is $R = \sum_t \gamma^t r_t$, where $\gamma \in [0, 1]$ is the discount factor weighting future rewards by their temporal distance. A trajectory $\tau = (s_{1:T}, a_{1:T})$ represents a sequence of state-action pairs sampled from the environment. Multi-modal inputs $m = \{m^{(1)}, \dots, m^{(k)}\}$ (e.g., visual, textual, proprioceptive) are encoded into latent variables $z = f_\phi(m)$, enabling compact state abstractions. Conditioning on LLM/VLM embeddings, world-model latents, or diffusion priors acts as a *fusion operator*, integrating heterogeneous modalities for s_t into a unified latent policy representation in the RL framework.

3. Base model

Generative models as tools for RL can be primarily classified by their *Base Model—i.e.*, the backbone architecture. In our review, we identify five main architectures that are typically used in RL: LLMs, VLMs, diffusion models, world models, and video prediction models. Fig. 5(a) visually represents these models, along with reference papers [8,11,47,48] for each category and their associated input/output modalities.

The *Base Model* section classifies the papers according to their architecture, briefly describes their features, and summarizes key aspects in tables. These aspects are important when selecting a tool for the RL tasks defined in Section 2. The table columns represent the following:

- *Paper*: Lists the corresponding research papers.
- *Framework*: Specifies the generative AI framework used.
- *Base Model/ Architecture*: Specifies the generative AI architecture used.
- *Network*: Indicates the actual neural network model (*e.g.*, GPT-3.5, GPT-4 for LLMs).
- *Code Available*: Highlights whether the implementation is accessible.
- *Training Data*: Lists the environments or datasets employed.
- *Task*: Describes the key task or feature handled by the framework (*e.g.*, reward function design, exploration).
- *Simulation* or *Real Exp.*: Distinguishes between simulated environments and real-world experiments.

Table 2 offers details on LLM-based works; Table 3 covers VLMs; Tables 4 and 5 summarize diffusion models; and Table 7 presents information on world models and video prediction models.

3.1. LLM

LLMs are changing how agents learn through RL. These models enable agents to acquire skills more effectively and flexibly by generating high-level plans, interpreting instructions, and providing structured feedback [8,9,31,49–51]. LLMs are incorporated into RL systems mostly because of their capacity to leverage vast amounts of existing information [43,52]. Because of this integrated knowledge, agents are less penalized by learning everything from scratch, which facilitates exploration and convergence on efficient behaviors. For example, GPT-4 is utilized in the EUREKA framework [8] to automatically generate code for RL tasks. Similarly, in open-ended textual settings with undefined learning objectives, LMA3 [53] uses GPT-3.5. Adaptability is another benefit. LLMs can take zero-shot decisions through prompts or updated instructions, in contrast to classic RL techniques that frequently demand for big datasets and retraining when situations change[3]. But flexibility is not without its drawbacks. Because pre-trained models' reactions are limited by the data they were trained on, they may perform poorly in contexts that are novel or that change quickly. If not handled appropriately, this can result in biased behavior and false assumptions [24]. However, adding LLMs to RL also presents a number of difficulties. Interpretability is a major problem. It is challenging to comprehend or follow the logic behind the outputs of LLMs because they function mostly as "black boxes." In safety-critical applications such as robotics, this lack of transparency becomes a significant concern [54–56]. Furthermore, there are real-world limitations due to the computing requirements of LLMs [36,42,57] and few works utilize open-source models.

Recent work combining LLMs with RL is summarized in Table 2, which covers the main frameworks, models, code availability, training data, and use cases in the RL training loop. The BOSS framework [58], for instance, makes use of GPT-3.5 to provide skill chaining across long-horizon activities, assisting agents in learning complicated behaviors through the extension and reuse of acquired abilities. FoMo [59] increases agents' sensitivity to task context by using LLMs to dynamically scale reward signals based on visual inputs. The functions that the LLMs perform, including reward design, plan creation, or skill discovery, are represented by the *Task* in the table. These task modules are integrated into larger RL pipelines. More details are given in Sections 4 and 5.

3.2. VLM

The integration of multi-modal VLMs into RL is a significant step forward with respect to single modality LLMs, enabling more useful and informative text and visual input fusion [7,13–15,47,49,65–71]. This is particularly useful in complex robotic tasks where goal states are better represented through images rather than numbers or text. One of the key advantages of VLMs is their ability to generalize across tasks without requiring environment-specific fine-tuning, similar to LLMs [65]. Moreover, VLMs can work directly with real-world perception and don't need to rely on accurate environment information translated into text, compared to LLM frameworks such as EUREKA [8]. The computational demand of VLMs remains significant, if not greater than that of LLMs. Few methods [7,66] aim to mitigate this by solving RL tasks without excessive computational costs or calling the model inference at lower frequency. Moreover, fine-tuning VLMs to optimize their responses for RL specific tasks remains sometimes necessary [72]. The effectiveness of VLMs in RL also depends on the choice of architecture—*e.g.*, several works use specialized modules such as CLIP (Contrastive Language-Image Pre-Training), a model that predicts the most relevant text sequence given an image [30,68,73], while others use general-purpose

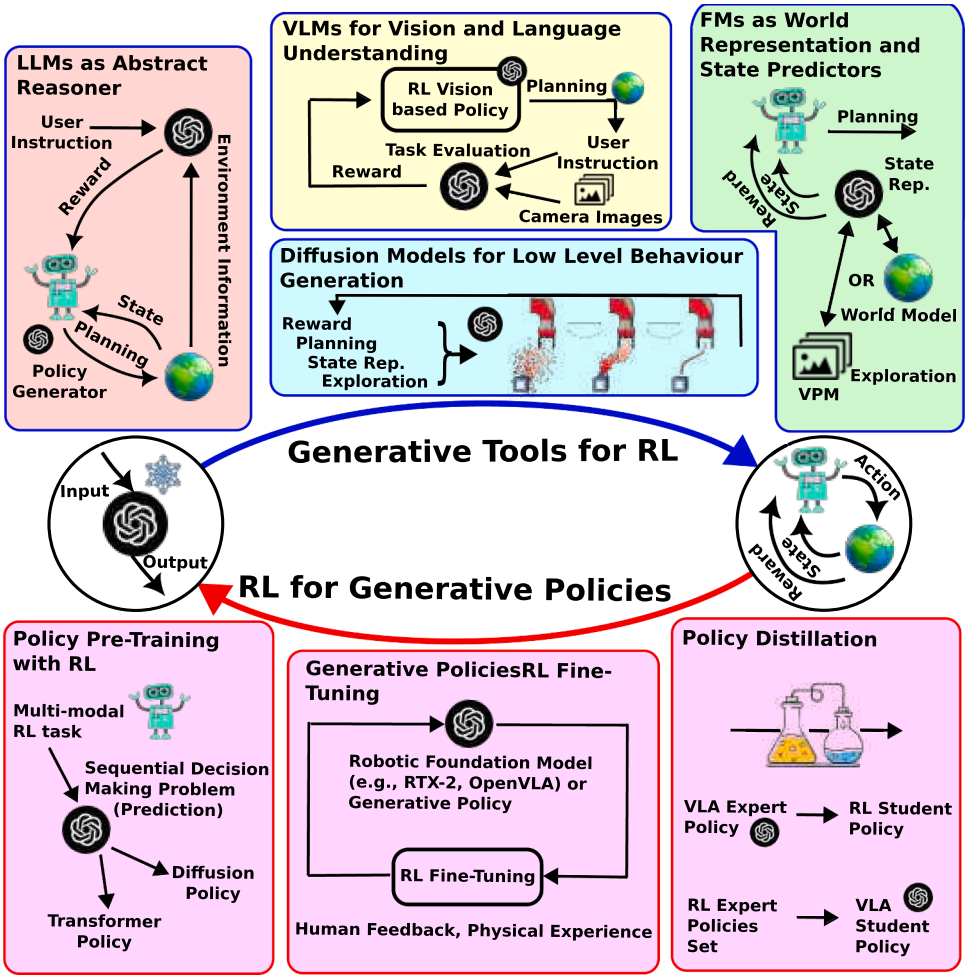


Fig. 4. Duality between RL and generative AI models in robotics. The upper portion of figure illustrates generative AI models (*e.g.*, LLMs, VLMs, diffusion models, world/video prediction models) as tools for RL-based decision making. The lower portion illustrates RL-based pre-training, fine-tuning and policy distillation, highlighting the complementary role of RL in enhancing generative policies.

foundation models like GPT-4—and experimental setup, inheriting the issues of LLMs. Some studies focus on simulation environments with synthetic data [7], while others explore real-world robotic applications [49], leading to variations in prompting strategies, RL methodologies, and evaluation metrics. Notably, in Table 3 some models are used primarily for reward generation—similar to LLMs but based on visual input (*e.g.*, *Reward Generation* in Venuto et al. [66])—while others use VLMs to define the goal task or condition the learning to it (classified as *State Representation*), as seen in Cui et al. [65]. See Section 5 for more details on column *Task*.

3.3. Diffusion model

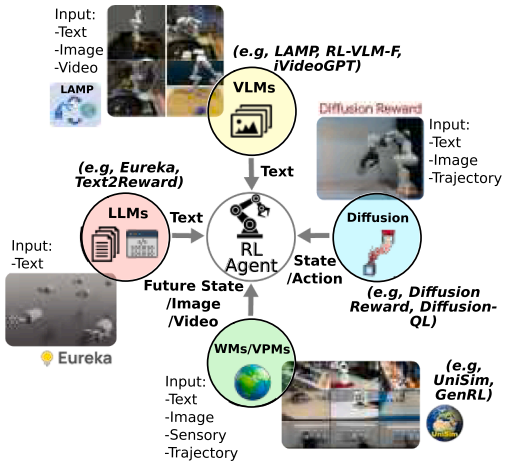
Diffusion models, known for success in image generation [74,75], are emerging in RL [76–85]. Unlike models that use only textual or visual inputs, diffusion models can also operate directly in the *trajectory space*—in contrast to conventional RL models that act within the *state space* $\mathcal{S}$, which represents instantaneous agent configurations at a single time step. The trajectory space $\mathcal{T}$ instead encompasses full sequences of states and actions $\tau(s_{1:T}, a_{1:T})$ evolving under a policy π_θ, allowing diffusion models to capture coherent temporal dependencies rather than isolated states–generating continuous, dynamically consistent actions [86,87]. By modeling such trajectories rather than pointwise states, diffusion models learn to reverse a diffusion process that progressively adds noise to states, rewards or policy parameters, generating new quantities from random noise. This approach has several advantages. First, it promotes behavior diversity, encouraging exploration and the discovery of novel solutions by generating a wide range of control policies. Second, it is well-suited for tasks with continuous action spaces, thereby avoiding the suboptimal performance that can arise from discretization. Finally, it improves sample efficiency by learning from limited demonstrations, reducing the need for extensive data collection [10,20,88]. The ability of diffusion models to be conditioned on language instructions or goal prompts enhances adaptability, making zero-shot policy generation possible. While still early in development, diffusion models offer new perspectives compared to more widely explored LLMs and VLMs [87,89,90]. Their iterative process allows for fine adjustments, making them ideal for complex tasks like grasping and balancing [91]. Additionally, their capacity to model complex dynamics and manage noise and uncertainty makes them robust in real-world robotic environments. However, their effectiveness in high-dimensional tasks is limited [20,92,93] and issues such as sensitivity to training data persist. We investigate further these aspects in Section 5 and we classify diffusion-based papers in Tables 4 and 5.

3.4. World model and video prediction model

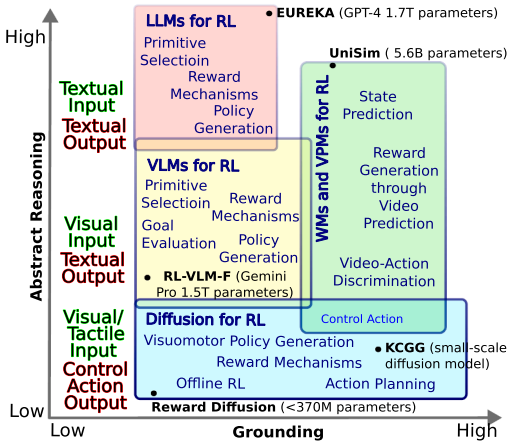
World models in RL introduce new methods for learning representations and dynamic models that serve as internal simulators for planning and prediction in RL [16,102–104]. Typically, a world model consists of an encoder that incorporates environmental observations, a dynamics model that is learned during pre-training and allows for in-

Table 2
Summary of large language models as tools for RL.

Paper	Framework	Base Model/ Architecture	Network	Code	Training Data	Task	Simulation or Real-World Exp.
Chu et al. [54]	Lafite-RL	LLM, World Model	GPT 3.5 and GPT 4	N/A	Simulated robotic environment	Reward Signal	Simulation
Colas et al. [53]	Language Model Augmented Autotelic Agent (LMA3)	LLM	ChatGPT (gpt-3.5-turbo-0301)	N/A	Text-based environment (CookingWorld)	Planning & Exploration	Simulation
Zhang et al. [58]	Bootstrapping your Own Skills (BOSS)	LLM	GPT 3.5	Yes	Alfred Dataset, Demonstration data	Planning & Exploration	Simulation + Real-World
Ahn et al. [3]	SayCan	LLM	GPT-3, FLAN and LAMBDA	Yes	Various real-world robotic tasks	Planning & Exploration	Real-World
Ma et al. [8]	EUREKA	LLM	GPT 4	Yes	Variety of robotic tasks	Reward Signal	Simulation (NVIDIA Isaac Gym)
Lubana et al. [59]	Foundation Models as Reward Functions (FoMo Rewards)	LLM	N/A	N/A	Visual observations from agent's trajectory	Reward Signal	Simulation
Huang et al. [4]	Grounded Decoding (GD)	LLM	InstructGPT and PALM	N/A	Various embodied tasks	Planning & Exploration	Simulation
Carta et al. [2]	Grounded Language Models (GLAM)	LLM	N/A	Yes	Textual environment (BabyAI-Text)	State Representation	Simulation
Du et al. [55]	Exploring with LLMs (ELLM)	LLM	GPT-3, GPT-2 and InstructGPT	Yes	Various environments (Crafter, Housekeep)	State Representation	Simulation
Colas et al. [60]	Intrinsic Motivations And Goal Invention for Exploration (IMAGINE)	LLM	N/A	Yes	Procedurally-generated scenes	Planning & Exploration	Simulation
Hu and Sadigh [61]	LLM Prior Policy with Regularized RL (instructRL)	LLM	GPT 3.5, GPT J	Yes	Toy game and Hanabi benchmark	State Representation	Simulation
Yu et al. [52]	LLM-based Reward Translator	LLM, VLM	GPT-4	Yes	Simulated quadruped robot, dexterous manipulator robot tasks	Reward Signal	Simulation
Dalal et al. [62]	Plan-Seq-Learn (PSL)	LLM	GPT-4	Yes	Challenging robotics tasks benchmarks	Planning & Exploration	Simulation
Song et al. [63]	Self-Refined Large Language Model	LLM	GPT-4	Available Soon	Various continuous robotic control tasks	Reward Signal	Simulation (NVIDIA Isaac Gym)
Xie et al. [5]	Text2Reward Framework	LLM	GPT-4	Yes	ManiSkill2, MetaWorld, MuJoCo	Reward Signal	Simulation
Triantafyllidis et al. [64]	Intrinsically Guided Exploration from Large Language Models (IGE-LLMs)	LLM	GPT-4	N/A	Robotic manipulation tasks	Planning & Exploration	Simulation



(a) Base models modularity.



(b) Trade-off between abstraction and grounding, when using generative AI tools for RL.

Fig. 5. Generative AI tools for RL.

Table 3
Summary of vision-language models as tools for RL.

Paper	Framework	Base Model/ Architecture	Network	Code	Training Data	Task	Simulation or Real-World Exp.
Cui et al. [65]	Zero-Shot Task Specification (ZeST)	VLM	ImageNet-supervised ResNet50, ImageNet-trained MoCo, CLIP	N/A	Simulated robot manipulation tasks and real-world datasets	State Representation	Simulation
Venuto et al. [66]	VLM-CaR (Code as Reward)	VLM	GPT-4	N/A	Discrete and continuous environments	Reward Signal	Simulation
Adeniji et al. [47]	Language Reward Modulated Pretraining (LAMP)	VLM, World Model	CLIP	Yes	Ego4D, video Dataset, automatically Generated language Dataset	Reward Signal	Simulation
Ma et al. [67]	Language-Image Value Learning (LIV)	VLM	CLIP with ResNet 50 backbone	Yes	EpicKitchen, simulated and real-world robot environments	Reward Signal	Simulation + Real-World
Wang et al. [7]	RL-VLM-F	VLM	Gemini and GPT-4 Vision	Yes	Various domains including classic control and manipulation tasks	Reward Signal	Simulation
Sontakke et al. [68]	RoboCLIP	VLM	S3D	Yes	Howto100M	Reward Signal	Simulation
Yang et al. [69]	ROBOFUME	VLM, LLM	MiniGPT4	Available soon	Diverse datasets, real robot experiments	Reward Signal	Simulation + Real-World
Di Palo et al. [70]	Unified Agent with Foundation Models	VLM, LLM	CLIP, FLAN-T5	N/A	Image and Text datasets	State Representation	Simulation
Rocamonde et al. [15]	Zero-Shot Vision-Language Models (VLM-RMs)	VLM	CLIP	Yes	Human motion task dataset	Reward Signal	Simulation
Baumli et al. [13]	Vision-Language Models (VLMs) like CLIP	VLM	CLIP	N/A	Playhouse	Reward Signal	Simulation
Chen et al. [14]	PR2L (Promptable Representations for RL)	VLM, LLM	Vicuna-7B version of the InstructBLIP, a Llama2-7B Prismatic VLM	Yes	Minecraft, Habitat environments	State Representation	Simulation
Mahmoudieh et al. [71]	Zero-Shot Reward Model (ZSRM)	VLM	ResNet-50	N/A	Large dataset of captioned images	Reward Signal	Simulation
Ma et al. [49]	ExploRLLM	VLM, LLM	GPT-4	N/A	Image-based robot manipulation tasks dataset	Planning & Exploration	Simulation + Real-World

ternal simulation, and an optional decoder that reconstructs information from the latent space. A reward model, which forecasts rewards based on the learned representation, might also be included. Although they are not commonly employed as primary dynamic predictors, LLMs and VLMs have recently being integrated into world models. LLMs are mainly used for high-level reasoning and task specification [105,106], while VLMs [103] support state representation by providing rich visual-linguistic embeddings. The core transition dynamics are typically modeled using architectures like RNNs, transformers, or diffusion models.

Table 7 demonstrates that *State Representation* is the primary use of world models, with almost all works concentrating on this function [18,103,107,108]. Few studies focus on *Reward Signal* generation [106,109], or *Planning & Exploration* [48,110] as the main tasks. With little application to real-world data, the majority of the studied techniques are trained and assessed in simulated contexts. Though, few works [105,109–111], include real-world experiments.

Video prediction models, special models that learn the temporal dynamics of a visual environment, have also proved good results in visual RL, with frameworks like VIPER [112]. However, the main limitation is the reliance on specific environments for training [109,113].

See Table 6 for a comparison between features of VPMs and WMs for RL, where *action-conditioned* means that the model's prediction can be influenced also by the action taken by the RL agent, and Sections 5 for more details.

4. Modality

This section focuses on the classification of five types of generative AI models used in RL, as introduced in Section 3, with an emphasis on how their input/output modalities shape their role within RL frameworks.

Bridging the gap between symbolic concepts and real-world experience is a concept referred to as *grounding*, introduced in Section 1. Our classification highlights a fundamental trade-off between abstraction and grounding: while LLMs [36,37] and VLMs [14,57,117–119] are adept at abstract reasoning and symbolic processing, their input/output operations are less directly tied to physical control [1–3,5,8,31]. On the other hand, diffusion models, by generating low-level, continuous control actions and operating directly in the action space, provide a more sample-efficient approach to policy learning [10,21,74,88,99,120], though they may lack the higher-level abstraction capabilities inherent to larger language models. Models that can effectively fuse abstraction and grounding for low-level action generation in robotics remain limited, although world models [48,102,116] and video prediction models [112,121] represent promising approaches by integrating both high-level abstraction and grounded sensory information.

The diagram in Fig. 5(b) shows that, based on our analysis, each generative AI tool used in RL contributes distinct input/output modalities. LLMs, working with text as both input and output excel at symbolic reasoning and can synthesize or interpret textual inputs such as task goals, environment documentation, or learned RL primitives, enabling them to produce reward signals or task refinements aligned with high-level objectives [8,9]. VLMs, in contrast, can process visual inputs and generate reasoning over visual scenes. This makes them valuable for visual feedback mechanisms [7]. Their ability to bridge visual and textual modalities allows flexible integration into tasks where visual context is critical. Diffusion models can be highly specialized for policy learning and state generation, as they can model complex distributions over low-level states and actions. Operating directly in continuous action spaces, they can produce precise control signals that are easily integrated with RL algorithms [11,21]. This modality specificity makes them ideal for ap-

Table 4
Summary of diffusion models as tools for RL.

Paper	Framework	Base Model/ Architecture	Code	Training Data	Task	Simulation or Real-World Exp.
Lu et al. [76]	Contrastive Energy Prediction (CEP)	Diffusion model	Yes	D4RL benchmarks	State Representation	Simulation (D4RL benchmarks)
Kang et al. [77]	EDP	Diffusion model	Yes	D4RL benchmark datasets	State Representation	Simulation (D4RL benchmark)
Suh et al. [78]	Score-Guided Planning (SGP)	Diffusion model	Yes	Cart-pole system, D4RL benchmark (MuJoCo tasks), pixel-based single integrator environment	Planning & Exploration	Simulation (Various simulations)
Hansen-Estruch et al. [79]	Implicit Diffusion Q-Learning (IDQL)	Diffusion model	Yes	D4RL benchmark (halfcheetah, hopper, walker2d, antmaze), Maze2D	State Representation	Simulation (D4RL benchmark, Maze2D)
Kim et al. [80]	DuSkill	Diffusion Model	N/A	Rule-based expert policies, multi-stage Meta-World tasks (slide puck, close drawer)	State Representation	Simulation (Multi-stage Meta-World)
Venkatraman et al. [81]	LDCQ	Diffusion Model, World Model	N/A	MuJoCo benchmarks (Hopper, Walker, Ant), Maze2D, AntMaze, FrankaKitchen, CARLA	State Representation	Simulation (Various environments)
Hu et al. [82]	Temporally-Composable Diffuser (TCD)	Diffusion Model	N/A	Gym-MuJoCo environments (HalfCheetah, Hopper, Walker2D), Maze2D, Hand Manipulation tasks	State Representation	Simulation (Gym-MuJoCo, Maze2D, Hand Manipulation)
Psenka et al. [83]	Q-Score Matching (QSM)	Diffusion Model	Yes	DeepMind Control Suite (Cartpole Balance, Cartpole Swingup, Cheetah Run, Hopper Hop, Walker Walk, Walker Run, Quadruped Walk, Humanoid Walk)	State Representation	Simulation (DeepMind Control Suite)
Jain and Ravanbakhsh [84]	Merlin	Diffusion Model	N/A	PointReach, PointRooms, Reacher, SawyerReach, SawyerDoor, FetchReach, FetchPush, FetchPick, FetchSlide, HandReach	State Representation	Simulation (Various simulated environments)
Zhu et al. [86]	MADIFF	Diffusion Model	Yes	Multi-agent particle environments, MA Mujoco, StarCraft Multi-Agent Challenge (SMAC), NBA dataset	State Representation	Simulation (Multi-agent environments)
Ni et al. [87]	MetaDiffuser	Diffusion Model	Yes	MuJoCo benchmarks (Hopper-Param, Walker-Param), Point-Robot 2D navigation	State Representation	Simulation (MuJoCo, Point-Robot)
Chen et al. [85]	SfBC	Diffusion Model	N/A	D4RL benchmarks, AntMaze tasks, Maze2d, FrankaKitchen, Bidirectional-Car tasks	State Representation	Simulation (Various benchmarks)

plications in robotics and control where fine-grained action generation is required. Lastly, world and video prediction models, with their broad support for multi-modal input fusion and internal representation learning capacity, provide an even more flexible foundation. They can incorporate and generate rich multi-modal state representations (including visual, textual, proprioceptive, and other sensory data), making them well-suited for learning predictive models of environment dynamics and for supporting planning and model-based RL [48,102,116].

The level of abstraction and the size of the models (in terms of parameters) strongly influence how easily they can be integrated into the RL training loop. Larger, more powerful, and generalist models (*e.g.*, current LLMs, which are text-based) can be quickly incorporated in a zero-shot fashion in several frameworks [8,9]. However, they are often less tailored and less adaptable to the specific input/output requirements of RL tasks. Moreover, the choice of model—whether a diffusion model, an LLM, or a VLM—directly impacts the integration strategy within RL frameworks. Different models impose distinct computational demands and offer varying levels of flexibility in adapting to real-world dynamics [11,20,122]. The diagram in Fig. 5(b) visually summarizes these trade-offs across models, mapping representative tools based on their degree of symbolic reasoning and grounding in physical control. This complements our analysis of modality diversity and integration strategies by illustrating how models such as **UniSim**, **EUREKA**, **RL-VLM-F**, **Reward Diffusion**, and **KCGG** occupy different positions in this space.

5. Task

RL tries to perform optimal decision-making considering the interaction of an agent (*e.g.*, a robot) with its environment, with the goal of maximizing rewards [26,123]. In the RL framework, several core components play essential roles in guiding the agent's behavior. First, we have *states*, which represent all possible situations the agent might encounter within its environment. At any given moment, the agent will find itself in a particular state, prompting it to consider the best *action* (*e.g.*, the action that maximizes the reward). These actions are the set of choices available to the agent, allowing it to interact with and influence the environment. The *policy* represents the probability of taking action given a state (*e.g.*, it's the agent's strategy). As the agent takes actions, it receives feedback from the environment through the *reward function*, which assigns immediate rewards based on the outcome of each action. This reward signal helps the agent understand how to act optimally to achieve long-term goals. Two key functions support the agent in making more informed decisions over time. The *value function* estimates the expected long-term return of being in a particular state, giving the agent an idea of how promising that state is under the current policy. The *Q-function*, on the other hand, is slightly more detailed. It assesses the expected return not just for states, but also for specific actions taken

Table 5
Summary of diffusion models as tools for RL (continued).

Paper	Framework	Base Model/ Architecture	Code	Training Data	Task	Simulation or Real-World Exp.
Liang et al. [90]	AdaptDiffuser	Diffusion Model	Yes	Synthetic expert data	Planning & Exploration	Simulation (Maze2D, MuJoCo)
He et al. [20]	Multi-Task Diffusion Model (MTDIFF)	Diffusion model	N/A	Meta-World and Maze2D environments	Planning & Exploration	Simulation (Meta-World, Maze2D)
Brehmer et al. [94]	EDGI	Diffusion Model, World Model	N/A	Offline trajectory datasets (3D navigation, Kuka robotic arm)	Planning & Exploration	Simulation
Li et al. [95]	Hierarchical Diffusion (HDMI)	Diffusion model, World Model	N/A	Maze2D, AntMaze, D4RL environments, NeoRL benchmark	Planning & Exploration	Simulation (Various environments)
Xiao et al. [89]	SafeDiffuser	Diffusion model	Yes	Maze2D environments, MuJoCo environments (Walker2D, Hopper), Pybullet environments	Planning & Exploration	Simulation (Multiple environments)
Chen et al. [96]	Hierarchical Diffuser	Diffusion Model, World Model	N/A	Maze2D (U-Maze, Medium, Large), Multi2D, AntMaze, Gym-MuJoCo, FrankaKitchen	Planning & Exploration	Simulation (Various benchmarks)
Kim et al. [97]	Sub-trajectory Stitching with Diffusion (SSD)	Diffusion Model	Yes	Maze2D environments, Fetch environments	Planning & Exploration	Simulation (Maze2D, Fetch)
Lee et al. [91]	Restoration Gap Guidance (RGG)	Diffusion model	Yes	Maze2D environments, Gym-MuJoCo locomotion tasks, block stacking tasks with Kuka iiwa robotic arm	Planning & Exploration	Simulation (Multiple environments)
Zhang et al. [98]	Language Control Diffusion (LCD)	Diffusion Model	Yes	CALVIN language robotics benchmark, CLEVR-Robot benchmark	Planning & Exploration	Simulation (CALVIN, CLEVR-Robot)
Janner et al. [99]	Diffuser	Diffusion Model	Yes	Maze2D environments, block stacking tasks, D4RL locomotion suite	Planning & Exploration	Simulation (Multiple environments)
Nuti et al. [100]	Relative Reward Function	Diffusion model	Yes	Maze2D, D4RL locomotion tasks, I2P dataset	Reward Signal	Simulation (Maze2D, D4RL)
Mazoure et al. [101]	Diffused Value Function (DVF)	Diffusion Model, World Model	N/A	Maze2D environments, PyBullet environments, D4RL offline suite	Reward Signal	Simulation (Multiple environments)

Table 6
Comparison between video prediction models and world models for RL.

Feature	Video Prediction Models	World Models
Output	Future frames (pixels)	Latent states, frames, rewards
Use Case	Future frame generation	Future state/control generation
Reward Modeling	Rarely included	Often included
Action-Conditioned	No	Usually

within those states, enabling the agent to evaluate the quality of particular actions in given situations. One of the biggest challenges in RL is designing an effective reward function [8,22]. A poorly designed reward system can mislead the agent, resulting in suboptimal or unintended behaviors. Crafting rewards that properly guide the agent toward good outcomes is crucial. Other two significant challenges are representing the state space and exploring it. If states are not represented accurately or comprehensively, the agent may struggle to learn the true dynamics of the environment, which can impede its ability to make optimal decisions [23].

This section reviews key works that exemplify how generative models have been employed to address critical aspects of the RL training loop in robotics, specifically focusing on: (i) *Reward Signal* generation in the presence of sparse rewards and under goal specification constraints, (ii) *State Representation* learning to improve sample efficiency, and (iii) *Planning & Exploration* mechanisms that support generalization to new tasks. Based on the modality and purpose of the generative models employed in relation to the RL *Task*, the works are categorized and examined. A more thorough classification of the basic models under examination is provided in Table 8, while a schematic overview summarizing their relationships within the RL framework is illustrated in Fig. 6.

5.1. Reward signal

An increasing amount of research explores how generative AI models, particularly foundation models, can help automate or enhance reward design [2,4–6,8,13,15,47,53–55,63–65,71,109,112,124–126]. These models offer new ways to interpret user input, perceive environments, and produce reward signals that guide effective policy learning.

5.1.1. Reward design with LLMs

Recently, LLMs such as GPT-4 [37] or Llama 3 [36], have been used to address the challenges of reward function design and direct reward specification. While LLMs struggle to directly control robots due to their ineffectiveness to produce control commands, they are highly valuable in evaluating the performance of RL agents, performing in-context reasoning on natural language problems. Traditionally, reward functions in RL are meticulously hand-crafted by researchers, often requiring significant domain expertise and limiting the agent's ability to adapt to new situations. However, foundation models offer a compelling alternative by providing rich, generalist knowledge representations that can be leveraged to automatically generate or inform reward functions. Imagine a robot tasked with "cleaning the kitchen". An LLM could identify sub-tasks like "wiping the table" and "putting dishes in the dishwasher" and assign rewards based on their successful completion. This structured approach breaks down complex tasks and provides clear goals for the RL agent.

Some approaches use LLMs for zero-shot reward generation [5,8,63] to convert natural language descriptions of desired behaviors into mathematical reward functions. TEXT2REWARD [5] and Self-Refined LLM [63] use LLMs to generate reward functions as executable code, while EUREKA [8] employs LLMs for evolutionary optimization of reward code. These methods leverage the LLM's ability to understand task

Table 7
Summary of world models as tools for RL.

Paper	Framework	Model Class	Code	Training Data	Task	Simulation or Real-World Exp.
Ha and Schmidhuber [18]	MDN-RNN	World Model	Yes	CarRacing-v0, DoomTakeCover-v0	State Representation	Simulation
Wang et al. [105]	GENSIM (GPT-4, GPT-3.5, Code Llama)	World Model, LLM	Yes	Generated 100 tasks from a task library	State Representation	Simulation + Real-World
Mazzaglia et al. [103]	MFWM (GRU-based, InternVideo2)	World Model, LLM, VLM	Yes	Walker, Cheetah, Quadruped, Stickman, Kitchen	State Representation	Simulation
Wu et al. [107]	iVideoGPT	World Model	Yes	OXE Dataset, SSv2, RoboNet	State Representation	Simulation
Mao et al. [114]	Segment Anything Model (SAM), LLMs (GPT-3.5)	World Model, LLM	N/A	Cart Pole, Lunar Lander	State Representation	Simulation
Bruce et al. [115]	Genie (ST-Transformer)	World Model	N/A	Platformers dataset, robotics dataset	State Representation	Simulation
Yang et al. [48]	Observation Prediction Model (Video Diffusion Model)	World Model, VLM, Diffusion Model	N/A	Simulated environments, real robot data, human activity videos, panorama scans	Planning & Exploration	Simulation
Seo et al. [116]	Masked World Models (MWM)	World Model	Yes	Meta-world, RLBench	State Representation	Simulation
Seo et al. [111]	Multi-View Masked World Models (MV-MWM)	World Model	Yes	RLBench	State Representation	Simulation + Real-World
Ye et al. [110]	Foundation Actor-Critic (FAC)	World Model, Diffusion Model	N/A	Internet-scale robotics datasets, Meta-World	Planning & Exploration	Simulation + Real-World
Chen et al. [109]	Domain-agnostic Video Discriminator (DVD)	World Model	N/A	Something-Something-V2, robot videos in various environments	Reward Signal	Simulation + Real-World
Nottingham et al. [106]	DECKARD	World Model, LLM	Yes	Minecraft environment	Reward Signal	Simulation
Zala et al. [104]	EnvGen	World Model, LLM	Yes	Generated and original environments	State Representation	Simulation
Wang et al. [108]	RoboGen Generative Simulation	World Model, LLM	Yes	Generated tasks, scenes	State Representation	Simulation

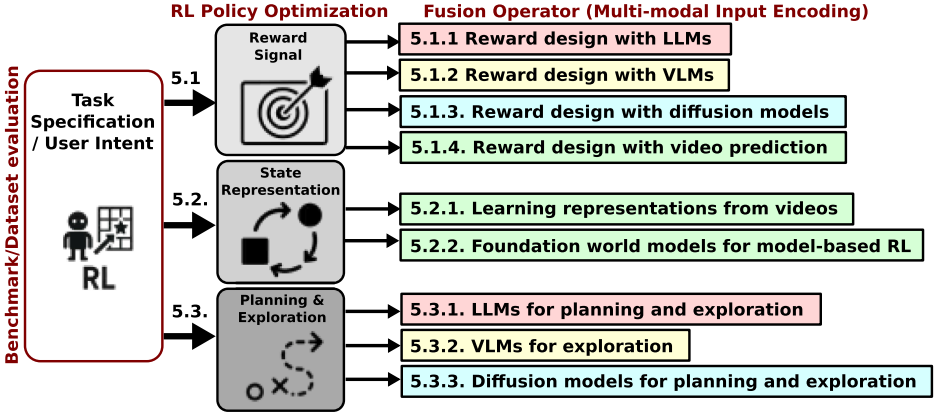


Fig. 6. Generative models as information fusion operators across RL tasks. A schematic overview illustrating how various generative AI architectures contribute to the three main components of the RL loop (panels color is coherent with the taxonomy).

Table 8
Generative AI tools for RL integration. Classification of generative AI models as tools based on integration into RL, as anticipated in Sections 3 and 4.

Network	Reward Signal	State Representation	Planning & Exploration
LLM	Textual feedback	Textual state encoding	High-level guidance
VLM	Multi-modal feedback	Multi-modal state encoding	Multi-modal evaluation
Diffusion Model	Reward generation	State generation	Action sampling
WM and VPM	Reward prediction	State prediction	Model-based planning

semantics and translate them into reward functions. Others leverage the in-context learning capabilities of LLMs to learn from demonstrations or feedback provided in natural language, shaping the reward function iteratively [52–55,64]. For example, Lafite-RL [54] uses LLM feedback to iteratively refine the reward function, employing few-shot learning to guide the LLM's generation process; while Language to Rewards [52] uses LLM feedback to define reward parameters that are optimized for specific tasks. While TEXT2REWARD and Lafite-RL focus on generating dense reward functions that provide feedback at each timestep, other methods, like Language to Rewards, may generate sparse rewards that are only provided at the end of an episode. Some methods, such as EUREKA and Language to Rewards, incorporate interactive feedback from humans to refine the reward function. In FoMo Rewards [59], LLMs are employed to evaluate the likelihood of an instruction accurately describing a task given a trajectory of observations. This likelihood then serves as a reward signal. While the use of LLMs for reward function generation is still an emerging field, it shows significant promise for improving the development and deployment of intelligent agents. This approach can streamline the reward function design process, making it more general and task agnostic. Lastly, Nair et al. [127] showcase the use of LLMs to learn language-conditioned rewards from offline data

and crowd-sourced annotations. This approach avoids the need for extensive human demonstrations, offering a scalable solution for learning complex behaviors. These different approaches highlight the versatility of LLMs in reward generation for RL in robotics, while the choice of approach depends on the specific requirements of the task and the available resources. However, one main drawback we found in existing work leveraging LLMs to craft rewards is that, since they work with text-only input, they are limited in acquiring perception from the real world [9]. This especially limits their capability to simulation environments where full state information is readily available.

5.1.2. Reward design with VLMs

The integration of vision-language models [73,128] in RL task evaluation represents a significant advancement over methods that rely solely on textual or numerical feedback [7,13,15,47,47,65,68,69,71,126]. By leveraging the power of visual understanding, VLMs can be used for task evaluation [47,65] based on prompts and image frames as input.

Many robotic tasks involve interacting with a complex environment filled with diverse sensory information. Vision foundation models can leverage their ability to process different data modalities (text, images) to construct richer reward functions. Imagine a robot tasked with sorting laundry. A VLM could analyze an image of the clothing to identify the fabric type and color, and combine this information with text instructions to create a reward function that promotes sorting based on predefined categories, performing multi-modal reward learning. In Mahmoudieh et al. [71], Rocamonde et al. [15] and Adeniji et al. [47] VLMs are employed as zero-shot reward models. These approaches leverage the in-context learning ability of VLMs to evaluate if a visual scene coming from a camera sensor effectively match the language task description. This approach eliminates the need for manually designing reward functions or gathering extensive human feedback; moreover, it does not need environment state information. Differently, in RoboCLIP [68] rewards are generated by evaluating how closely the robot's actions match the provided example. Interestingly, Code as Reward [66] utilizes VLMs to generate reward functions through code, thereby reducing the computational overhead of directly querying the VLM every time. Wang et al. [7] takes a slightly different route, querying VLMs to express preferences over pairs of image observations based on task descriptions, and subsequently learning a reward function from these preferences.

5.1.3. Reward design with diffusion models

Similar to other types of generative AI tools, diffusion models can be leveraged to infer rewards in RL. Huang et al. [11] and Nuti et al. [100] present methods for deriving reward functions by conditional diffusion models that are trained on expert visual demonstrations to generate rewards, similarly to the usage of diffusion models for conditional image generation [74]. On a different approach, Mazoure et al. [101] introduces the Diffused Value Function (DVF) algorithm, which leverages diffusion models to estimate value functions from states and actions, enhancing efficiency without explicitly learning rewards. They show improved long-term decisions. To conclude, diffusion models are a relatively new approach to reward design, but they show strong promise; particularly due to adaptability, dense reward generation, and fast inference. We do not yet see immediate applications for text-conditioned reward generation.

5.1.4. Reward design with video prediction

The approaches for reward design presented so far in our taxonomy often lack accuracy in predicting changes in the robot's visual state within the environment. Recent methods, such as those in Chen et al. [109] and Escontrela et al. [112], utilize the ability to predict future video frames based on current observations to shape rewards more effectively, leveraging environmental models (*e.g.*, video prediction models) to enhance reward specification in RL systems. Specifically, Chen et al. [109] propose DVD, a discriminator that learns multi-task reward functions by classifying whether two videos perform the same task. This approach allows generalization to unseen environments and tasks by learning from a small amount of robot data and a large dataset of human videos. Escontrela et al. [112] introduce VIPER, which uses pre-trained VPMs to provide reward signals for RL based on frame observations rather than the actions the robot takes. VIPER achieves expert-level robot control across various tasks in simulations without predefined rewards, demonstrating the potential of VPMs for online reward specification during robotic task execution. At the same time, other researchers propose HOLD [113] and VIP [129]. This approaches generalize to unseen robot embodiments and environments, effectively accelerating RL training on various manipulation tasks zero-shot; the second one is trained on large-scale human videos.

5.2. State representation

Foundation models can offer a promising avenue to address the challenge of more realistic state representations for data augmentation and training in RL [6,17,48,103,105,106,106–108,111,115]. However, the successful integration of foundation models as world representations for meaningfully training robots on augmented data and through model-based RL techniques hinges on addressing a fundamental issue mentioned many times in our discussion: *grounding*. While these models excel at processing abstract representations, their ability to effectively model realistic scenarios depends on establishing a meaningful connection between these representations and the real-world objects or concepts they abstract [102]. This section surveys research papers that utilize learned representations to predict and interact with environments in RL.

5.2.1. Learning representations from videos

Integrating VPMs with RL enhances the training of robots by allowing them to predict future frames from a sequence of previous images. VPMs forecast how a state will evolve, enabling the RL agent to plan actions and make decisions with more data-efficient learning and safer exploration, by anticipating future scenarios while reducing reliance on real-world trials [110,121,124,130]. However, the success of this approach depends on the accuracy of the predictions, as well as managing the increased computational complexity, that often limits the possibility to use a single pre-trained model for a wide variety of robotics tasks. As a result, a common trend in research is to train or fine-tune different VPMs for specific applications, but future work could focus on scaling VPMs to improve their generalization.

Du et al. [121] present UniPi, a unique approach that casts sequential decision-making as a text-conditioned video generation problem. UniPi leverages the knowledge embedded in language and videos to generalize to novel goals and tasks across diverse environments and to transfer effectively to downstream RL tasks. This approach enables multi-task learning and action planning, showcasing the potential of video generation for policy learning. Foundation RL [110] takes this objective a step further, they trained an RL model that leverages foundation priors from large-scale pre-training for embodied agents and they tested it on simulated robotic tasks.

5.2.2. Foundation world models for model-based RL

Many RL algorithms rely on accurate models of the environment's dynamics to plan and make decisions [22]. World models [18], particularly those trained on data that includes physical interactions, can be used to learn these models of the world. The learned model can help the RL agent to predict the consequences of its actions (*e.g.*, its future states) and make informed decisions based on predictions [102].

World models capture the action space of a robot. They can be constructed in a variety of ways, utilizing diverse data sources and employing different learning architectures. One recent approach to building a world model for abstract representation is through the use of large language models [104–106,111,116]. For example, Zala et al. [104] propose EnvGen, a framework where an LLM generates and adapts training

environments for small RL agents. By creating diverse environments tailored to specific skills, EnvGen enables agents to learn more efficiently in parallel. The LLM receives feedback on agent performance and iteratively refines the environments, focusing on weaker skills. On a similar approach Wang et al. [105] introduces GenSim, which uses GPT-4 [37] to automate the generation of diverse simulation tasks. Differently, Yang et al. [48] further advance the state-of-the-art in learning interactive real-world simulators for robotics as foundation models. They leverage diverse datasets, each rich in different aspects of real-world experience, to simulate the visual outcomes of both high-level instructions and low-level controls. The resulting simulator is used to train policies that can be deployed in real-world scenarios, showcasing the potential of bridging the sim-to-real gap in embodied learning.

World models can also be trained at a very large scale from scratch, leading to the development of multi-modal foundation world models capable of generalizing greatly in vertical domains. Building on this for efficient robot training, Wang et al. [108] present RoboGen, a generative robotic agent that automatically learns diverse skills through environment simulation. Similarly, Mazzaglia et al. [103] introduce GenRL, a framework for model-based RL training of generalist agents, that learns a multi-modal foundation world model. Moreover, Wu et al. [107] presents iVideoGPT, a scalable autoregressive transformer framework for interactive world models. It is pre-trained on millions of human and robotic manipulation trajectories, demonstrating its versatility in various downstream tasks. While Mao et al. [114] focuses on zero-shot safety prediction for autonomous robots using foundation world models. They propose a world model that combines foundation models with interpretable embeddings, addressing the distribution shift issue in standard world models. Their approach demonstrates superior state prediction and excels in safety predictions, highlighting the potential of foundation models for safety-critical applications. Lastly, Bruce et al. [115] introduces Genie, the first generative interactive environment trained in an unsupervised manner from unlabeled Internet videos.

While significant progress has been made in model-based RL and world models could unlock unlimited data availability for training, Wolczyk et al. [131] discuss the issue of catastrophic forgetting in post-training RL models, where pre-trained knowledge can be lost as new tasks are learned. This problem is particularly evident in compositional tasks, where different parts of the environment are introduced at different stages of training.

5.3. Planning & exploration

Policy learning refers to the process of determining a set of actions that can be executed to reach the desired target state for a given task [26,132]. In this section, we briefly highlight interesting findings in the literature where generative AI models are used to learn effective policies for RL. In particular, two important concepts are exploration, which enables the agent to discover new states and maximize rewards, and planning, which involves combining skills to develop more comprehensive policies [6,83,133]. Various approaches leveraging LLMs, VLMs, and diffusion models have been explored to enhance both exploration and planning in RL settings [14,49,58,60,65,82,90,92,94].

5.3.1. LLMs for planning and exploration

LLMs can be used for planning in RL, where the LLM acts as a strategic semantic planner, guiding the application of learned RL skills to new tasks based on task-specific prompts. Ahn et al. [3] and Huang et al. [4] introduce two methods that combines LLMs with pre-trained skills and affordance functions extracted from the RL training to ground language in robotic actions. The LLM is used to propose high-level actions, while the affordance functions, often learned through RL, determine the feasibility of these actions in the current context. This approach enables robots to execute complex tasks based on natural language instructions by ensuring that the proposed actions are both semantically relevant and physically feasible. Similarly, Plan-Seq-Learn (PSL) [62], is a modular approach that uses motion planning to connect abstract language from LLMs with learned low-level control for solving long-horizon robotics tasks. PSL breaks down tasks into sub-sequences, uses vision and motion planning to translate these sub-sequences into actionable steps, and then employs RL to learn the necessary low-level control strategies. This approach enables robots to efficiently learn and execute complex tasks by leveraging the strengths of both LLMs and RL.

Regarding exploration, Colas et al. [60] were the first to work on it by specifically incorporating language-driven imagination into RL. According to their IMAGINE architecture, “imagination” is the process of applying a learned goal recognizer to relabel prior experiences with other, language-based objectives. By predicting which natural language descriptions could realistically correspond to an episode’s outcome, this methodology enables the agent to associate new hypothetical objectives with previous ones. The agent may efficiently learn from these relabeled goals by combining a modular policy with a language-conditioned reward function. This improves generalization and exploration in a variety of language-specific tasks. The research underscores the critical role of language in enhancing creative, goal-driven exploration in RL. LLMs have shown promise in directly generating RL policies, instead of generating reward functions: GLAM [2], InstructRL [82], and BOSS [58], explore this, each with distinct approaches and contributions. GLAM focuses on grounding LLMs in interactive environments through online RL. It utilizes an LLM as the policy for an agent operating in a textual environment, refining the LLM’s understanding of the environment through continuous interaction and feedback. This approach aims to address the challenge of aligning the LLM’s knowledge with the actual environment dynamics, improving its ability to make decisions and achieve goals. The key innovation of GLAM lies in its online learning approach, where the LLM is not just pre-trained on existing data but actively learns and adapts as it interacts with the environment. This allows for a more dynamic and context-aware policy generation process. InstructRL, on the other hand, introduces a framework where humans provide high-level natural language instructions to guide the agent’s behavior. These instructions are used to generate a prior policy using LLMs, which then regularizes the RL objective. This approach aims to align the agent’s actions with human preferences and expectations, making it more suitable for collaborative tasks. InstructRL bridges the gap between human intentions and agent actions. The authors acknowledge that their work is limited by the challenges of abstracting certain actions, like continuous robot joint angles, into language, but they are optimistic about future advancements in multi-modal models expanding their applicability. Addressing the problem of generating realistic robotic policies and providing a more robust framework for translating abstract concepts into precise commands for robots, BOSS tackles the challenge of learning long-horizon tasks with minimal supervision. It starts with a set of primitive skills and progressively expands its skill repertoire through a bootstrapping phase. During this phase, the agent practices chaining skills together, guided by LLMs that suggest meaningful combinations. This approach enables the agent to learn complex behaviors autonomously, reducing the need for extensive human demonstrations or reward engineering. BOSS’s innovation lies in its ability to leverage the knowledge embedded in LLMs to guide the exploration and learning of new skills, making it a promising approach for developing generalist agents capable of performing a wide range of tasks.

5.3.2. VLMs for exploration

Another major theme in the research is the use of FMs that work with images together with text to guide exploration in RL agents. VLMs offer deeper grounding in robotics applications compared to LLMs because they directly associate visual perception with the input task in text form and do not require a separate vision module for perception, hence enhancing efficient exploration based on goal evaluation [49,65,67]. By processing real-time sensory data from the robot (*e.g.*, camera images), the model can identify unexpected situations or deviations from the expected plan. This information can be used to modify the reward

function online, penalizing actions that lead to undesirable outcomes and encouraging exploration of alternative strategies. For example, if a robot attempting to pick up a cup encounters an obstacle, the foundation model could adjust the reward function to prioritize navigating around the obstacle before resuming the grasping attempt. As demonstrated by Cui et al. [65], VLMs can enable zero-shot task specification in robotic manipulation by supporting more general and user-friendly goal representations—such as internet images or hand-drawn sketches—which promote exploration more closely aligned with the intended task. Other studies, however, assert that general-purpose VLMs might struggle with exploration and result in agents that act too roughly, especially in online RL [14].

5.3.3. Diffusion models for planning and exploration

Conventional RL planning techniques frequently use deterministic algorithms, which might perform poorly in complicated or unpredictable contexts [134–136]. Diffusion models, which were first created for generative tasks, have been modified to provide flexibility and stochasticity to the planning process in order to add robustness [20,90,92,94,95]. These models allow for flexible decision-making by iteratively converting noise into planned action sequences. One important strategy, Diffuser [99], combines RL with diffusion models to create reward-conditioned plans based on prior knowledge. It is excellent at long-term planning. To improve sample efficiency and generalization, extensions such as EDGI [94] treat planning as conditional sampling and add domain constraints. Other studies focus on safe planning, guaranteeing constraint satisfaction, and refining plans that are not feasible [89,91]. Diffusion models also improve exploration by stochastically generating a wide range of possible states and objectives [95–98,137].

Lastly, diffusion models for planning and exploration works well in **offline RL**, where we can train diffusion models to generate high-quality actions from large datasets or benchmarks [76–80,82–84,86,138]. Some methods [76,78], use gradient-based planning or energy functions to steer action generation and improve reward outcomes. Other works [77], focus on optimizing the sampling process to make diffusion models more practical and faster. Researchers also developed techniques like Implicit Diffusion Q-Learning [79] and Q-Score Matching [83], incorporating Q-learning ideas to better connect action choices with predicted rewards training offline [74,81,82,84,120].

Benchmarks and datasets

In the tables presented in Section 3, the column *Training Data* enumerates the benchmarks and datasets commonly employed to train RL agents for the tasks discussed in this section. Initially developed as standalone resources—either for RL benchmarking or for imitation learning and generative policy training—many of these simulated environments are now converging into unified benchmarks that bridge the two paradigms. This convergence has given rise to integrated testbeds that increasingly shape the development of generative AI driven agents, enabling standardized and comprehensive evaluation across perception, reasoning, and long-horizon control. Table 9 provides an overview and classification of these emerging frameworks.

6. RL pre-training

Until now, our survey has examined Generative AI as a tool within the RL pipeline (*Generative Tools for RL*). We now turn the coin over and explore the converse perspective: using RL itself to *pre-train*, *fine-tune*, and *distill* generative policy models. In *RL Pre-Training* we survey methods that pre-train transformer- or diffusion-based policies with RL objectives.

6.1. Transformer policy

Autoregressive transformer models were originally restricted to the domain of natural language, but they now form an integral part of RL, especially when the task can be framed as a sequential decision-making problem [38,149–152]. In particular, specialized adaptations of the Transformer architecture, such as Decision Transformers (DTs) [153], are becoming more popular. Like language models, DTs generate tokens (*i.e.*, actions) one at a time, conditioned on the previous context, and use the Transformer to predict the next action in a sequence based on previous return, state, and action tuples. Recent transformer-based RL studies focus on scalability, adaptivity, and transferability [125,154,155]. Results from Wen et al. [154] and Xu et al. [155] show how large models may generalize across various RL tasks. These methods show how transformer backbones can learn from little supervision and generalize well. HarmoDT [149], LATTE [150], and PACT [156] would be examples of such architectures that have trained transformers at scale for RL-based robotic control on real robots. Tackling a different problem, Q-Transformer [157] leverages offline data, human demonstrations, and trajectory rollouts to learn Q-functions with transformers so that robots can perform tasks appropriately. AnyMorph [158] pushes the generalization frontier by adapting to different robot morphologies. We describe here the small number of DT variants applicable to robotics, but we envision that many RL problems may be addressed by foundation models utilizing DTs backbones [159]. Lastly, a major trend is merging language and reasoning into policies: Mezghani et al. [133] merge language generation with action prediction, allowing agents to reason in natural language while planning actions. This has not been validated directly in real-world robotics but gives an interesting approach for tasks which require long-term planning. Recently, GATO [38] emerged as a generalist agent due to its ability to execute tasks across modalities, spanning from gaming to real-world robotics, using a single transformer backbone that performs input fusion into a RL policy.

6.2. Diffusion policy

Using iterative refinement instead of step-by-step prediction, diffusion models have recently acquired popularity as alternative to autoregressive techniques in robotic policy generation [21,99] (see Table 10 for a comparison). They are ideal for dynamic robotic environments due to their ability to manage uncertainty and provide a variety of behaviors [10,88,160]. Diffusion models can model complicated or high-dimensional dynamics by refining probability distributions to construct action sequences, in contrast to classic RL methods that yield deterministic outputs [20,22]. They can incorporate constraints such as safety or energy efficiency and allow expressive, context-sensitive behaviors by directly modeling action distributions [87,161]. Hegde et al. [162] present a method to condense a large archive of policies, trained using RL, into a single generative model. A diffusion model is then trained on these compressed representations to generate new policies conditioned on specific behaviors, either through quantitative measures or language descriptions. Other studies use diffusion models as policy in offline RL, improving performance and scalability [163,164]. Moreover, Diffusion-QL [88] uses conditional diffusion to match IL with Q-learning while preserving demonstration data proximity and optimizing rewards. Lastly, frameworks such as Decision Diffuser [165] and DIPO [166], formulate the sequential decision-making problem as a conditional generative modeling problem using RL to train a diffuser.

7. RL fine-tuning

Generative models, such as transformer-based and diffusion-based policies, are often trained with IL and they demand flexible RL fine-tuning strategies, if expert data for supervised fine-tuning (SFT) are limited. Recent works effectively integrate these expressive models into RL

Table 9
Representative benchmarks and datasets. Benchmark and datasets commonly used across tasks that integrate generative AI with RL, indicating their support for foundation model conditioning (*i.e.*, the inclusion of modalities that enable tasks or evaluations to incorporate guidance or context derived from generative AI models) and long-horizon evaluation.

Benchmark / Dataset	Domain	Typical Base Model	FM Conditioning	Long-Horizon Eval.	Common Metrics
Meta-World [139]	Robotics manipulation	Diffusion / VLM / LLM	✓ (goal text/image)	✓	Success rate, completion time
D4RL [138]	Offline RL, navigation, control	Diffusion / Transformer	✗	✓	Normalized return, success rate
MuJoCo Control Suite [140]	Continuous control	Diffusion / World Model	✗	✗	Return, stability, smoothness
FrankaKitchen [141]	Compositional tasks	Diffusion / World Model	✓ (goal images)	✓	Task completion, goal reward
RLBench [142]	Robotic manipulation	VLM / World Model	✓ (image-text prompts)	✓	Success rate, trajectory distance
EpicKitchen / Ego4D [143]	Egocentric video	VLM / World Model	✓ (image-text prompts)	✗	Action accuracy, recognition score
CALVIN [144]	Language-conditioned robotics	Diffusion / LLM	✓ (image-text prompts)	✓	Success rate, instruction adherence
RoboNet Dataset [145]	Visual RL, manipulation	World Model / VPM	✓ (goal video frames)	✓	Success rate
Minecraft	Instruction following	LLM / VLM	✓ (image-text prompts)	✓	Task success, episode length
Isaac Gym [146]	Physics-based simulation	LLM / Diffusion	✗	✓	Cumulative reward, success rate
HowTo100M [147]	Video-language grounding	VLM	✓ (video-text prompts)	✗	CLIP similarity, reward accuracy
Maniskill2 [148]	Physics-based simulation	LLM / Diffusion	✗	✓	Cumulative reward, success rate
Open-X Embodiment [25]	Real-robot dataset	VLA / Transformer	✓ (image-text prompts)	✓	Success rate, policy generalization

Table 10
Comparison of generative policies. Transformer autoregressive policies *versus* diffusion non-autoregressive policies in RL.

Aspect	Transformer Autoregression	Diffusion Model
Generation Process	Sequential, one step at a time.	Iterative, gradual refinement from noise.
Output Type	Typically discrete (e.g., tokens, discrete actions).	Typically continuous (e.g., trajectories, continuous actions).
Training Objective	Maximize likelihood of observed data.	Predict noise added during forward diffusion.
Inference Speed	Faster (if same size).	Slower due to iterative refinement.
Error Propagation	Errors can compound over time.	Less prone to compounding errors.
Expressiveness	Highly expressive for sequential data.	Highly expressive for continuous data.
Applications	Discrete action RL policies.	Continuous action RL policies.
RL Fine-tuning Ease	Straightforward with standard RL algorithms (e.g., PPO, Q-learning).	More complex, may require custom integration with RL.
Sample Efficiency	Moderate; improves with pretraining.	High; performs well in low-data regimes.
Action Modeling	Limited multi-modality; can struggle in complex spaces.	Strong multi-modal capabilities; excels in complex control.
Best For	Long-horizon, complex reasoning tasks; fast inference scenarios.	Smooth, continuous control; planning; multi-modal outputs.

pipelines without instability or loss of efficiency [45,167]. In this section, we explore two major classes of generative policies where we identified RL fine-tuning is particularly relevant, (i) large transformer policies (mostly VLA models), and (ii) diffusion policies (usually smaller in size).

Recent advances in **Transformer-based policies** have transformed robotic control, with robotics foundation models excelling through large-scale pre-training. A major step toward generative policy architectures and datasets for robotics is the Open X-Embodiment project [25], which introduced the largest open-source real-robot dataset—the Open X-Embodiment Dataset. This dataset supports the development of pivotal VLA models like RT-X models [168,169], Octo [170], and OpenVLA [171]. However, state-of-the-art generalist VLA policies are typically trained via IL and often require additional training to adapt to out-of-distribution scenarios [35,39,40,172]. RL has become an alternative to improve policy performance, particularly when it comes to managing distribution shifts or leading generalist models toward specialization. This has motivated the development of new RL-based fine-tuning algorithms, particularly actor-critic methods [173,174]. Actor-Critic approaches enable very reliable and effective policy updates by utilizing two networks: the critic to assess actions and the actor to decide on them. This method aids in RL fine-tuning by continuously enhancing action selections in response to critic feedback [22]. Through the use of on-policy algorithms such as PPO, cautious learning rates, and distinct actor-critic networks, the FLaRe framework [167] demonstrates RL-based fine-tuning by matching pre-trained transformer policies with new experience. FLaRe reduces the sim-to-real gap by using domain randomization and extensive simulations, increasing success rates by roughly +30 % on both real robots and in simulation. PPO is the recommended RL technique for VLA fine-tuning, according to Liu et al. [174]. Its advantages include shared actor-critic backbones, short training epochs, and great generalization on unseen objects and dynamic situations. However, the low-frequency action generation (reported in panel **a** in Fig. 7) of VLA models and the requirement of large amounts of interaction data even for minor adjustments, make online fine-tuning difficult and limited in rel world settings. Standard fine-tuning methods still need further exploration in case of VLA models, and current research focuses on large-scale RL fine-tuning for transformer policies [167]. More recent advances highlight the growing potential of RL to bridge vision, language, and action, opening new possibilities for adapting VLA models to downstream tasks by leveraging GPU-parallelized simulators and limited real-world data collection [175–178].

An alternative area where RL fine-tuning shows potential is with **Diffusion-based policies**, which offer many possibilities with RL due to their ability to predict smooth trajectories non-autoregressively at each inference step [88,179]. However, key problems in applying RL to diffusion policies are include the high-dimensional nature of the action space, the slow iterative denoising process involved in action generation, and training instability. Standard policy gradient methods have been viewed as inefficient for such models, due to the increased number of steps over which rewards must be predicted, introduced by iterative denoising [166]. Recent advances, such as Diffusion Policy Policy Optimization (DPPO), have demonstrated that RL can be effectively integrated with diffusion policies by formulating the denoising process as a Markov Decision Process [180]. This approach enables policy gradients to propagate through the diffusion steps, leveraging the structured noise removal process to facilitate more stable training. A major advantage of RL-fine-tuned diffusion policies is their ability to engage in on-manifold exploration, meaning that the policy remains close to the expert data distribution while still improving performance through RL. This structured exploration contrasts with traditional RL methods, which often struggle with off-manifold exploration, leading to unstable training and suboptimal policies [181]. Panel **b** of Fig. 7 shows the working principle of diffusion policies, while panel **c** depicts the state-of-the-art methods for RL-based fine-tuning generative policies.

In general, most research focuses on transformer-based or diffusion-based policy architectures separately, as they align with different goals and fine-tuning methods, although Actor-Critic RL is commonly used.

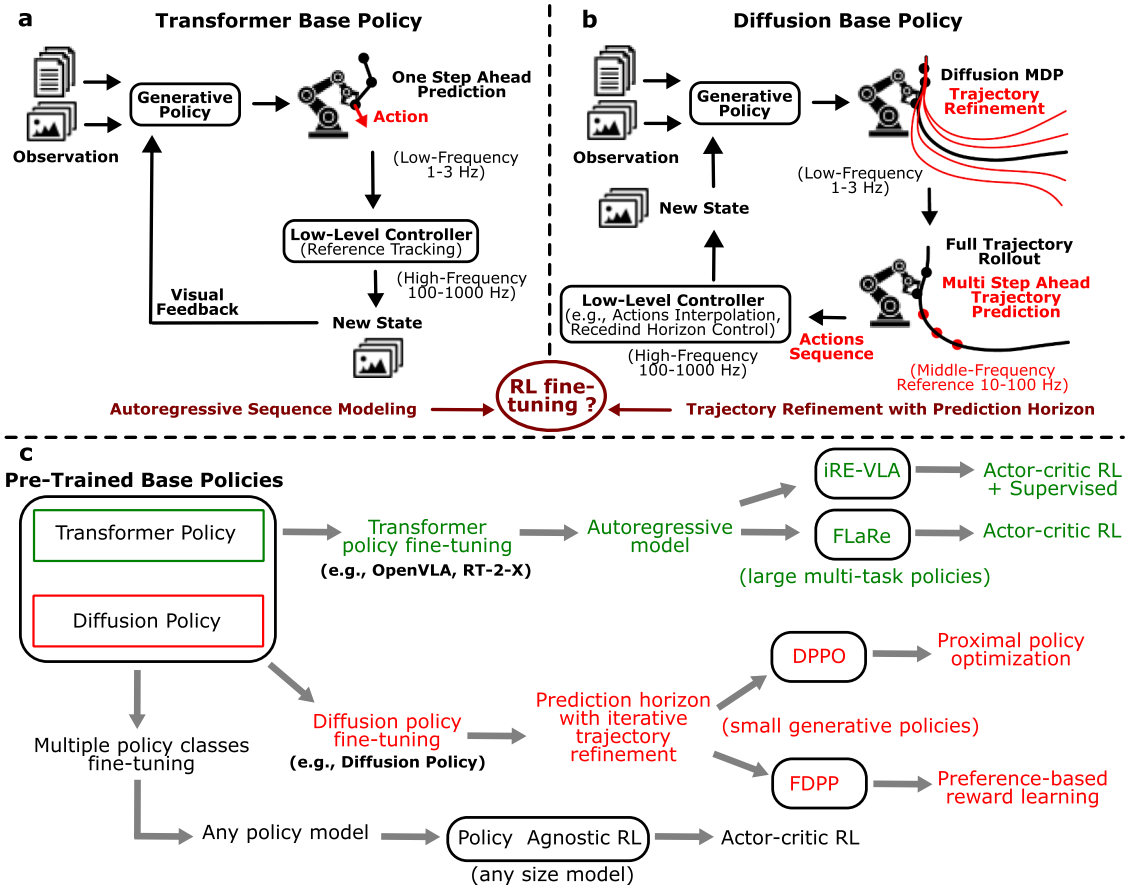


Fig. 7. RL for generative policies. This figure illustrates the transformer policy (panel **a** shows VLA inference time) and the diffusion policy (panel **b**) principles. In panel **c** we show recent RL methods that can be used to fine-tune a base generative policy.

While some researchers are exploring these pathways, others aim to develop a generalized RL-based fine-tuning framework for any generative policy, regardless of size and backbone model [45].

8. Policy distillation

Policy distillation is a well-known concept in the RL literature [46]. Recently, with the emergence of large generalist generative policies, RL-based methods for policy distillation have been applied to VLA models. In particular, recent works have explored policy distillation in OpenVLA [171] and Octo [170]. The goal of policy distillation is to transfer pre-trained knowledge from a *teacher policy* to a *student policy*. The recent literature can be categorized into two opposing approaches:

- **From generalist to expert:** Jülg et al. [182] developed a method to create task-specific RL agents distilled from a pre-trained VLA model. While VLA models are known for their strong generalization capabilities, they often struggle to achieve highly precise results compared to task-specific RL policies. The key idea is that the internal knowledge of VLA models can be useful in guiding RL exploration for specific agents. However, this work primarily presents preliminary simulation results, emphasizing that sim-to-real transfer remains the main challenge when training RL policies in simulation. This is particularly relevant given that OpenVLA and Octo typically perform better in real-world conditions [25,172].
- **From experts to generalist:** Xu et al. [183] propose a method to improve OpenVLA and Octo using demonstrations from expert RL policies. Their approach focuses on generating high-quality training data to fine-tune VLA models, as these models performance is highly dependent on the quality of their training data. Their experiments demonstrate that this method outperforms models trained solely on human demonstrations, achieving superior results in real-world precision manipulation tasks.

9. Challenges, recommendations and future research directions

We organize challenges, recommendations and few important insights for the future into three macro-categories based on our taxonomy—Section 9.1 is related to challenges in *Generative Tools for RL*, Section 9.2 presents challenges in *RL for Generative Policies* and Section 9.3 is on *Future Research Directions*. Each category addresses specific limitations and opportunities that must be tackled to enable scalable, safe, and adaptive robot learning that integrates generative AI and RL.

9.1. Challenges in generative tools for RL

Every generative model we discussed has unique advantages for RL; consequently, different generative AI models present distinct challenges, and we highlight those we identified as most urgent to be addressed.

LLMs for reward signals

Despite their strength in symbolic reasoning, LLMs lack grounding in real-world tasks [1,31]. Main challenges are: (i) **limited contextual understanding** [3,4,53,55,62,64], (ii) **real-world reduced adaptability** [5,8,9,54,59] because most works are based on the assumption that

the LLM has access to the environmental states from the simulation, and (iii) **symbolic-physical misalignment** [1,2,31,31].

Recommendation: Similarly to Ma et al. [9], we believe that LLMs can improve reward generation in real-world. Develop architectures that pair LLMs with RL controllers capable of translating high-level goals into more controlled rewards. Employ LLM fine-tuning gradually increasing real-world sensor data exposure to avoid simulation environments dependency.

VLMs for reward signals and state representation

VLMs excel at perception and language integration and they are more effective in real-world tasks than LLMs, but they struggle with translating visual-semantic representations into actionable RL policies. Challenges are: (i) **detailed scene understanding** [7,14,49,65–67,69,71] and (ii) **inferring physical affordances or dynamics** [13,15,47,68,70,72].

Recommendation: Associate VLMs with more meaningful information to improve precision: extracting pose of relevant objects, keypoints, or region of interest in the frames. Use positive and negative visual examples to generate semantic rewards, since it's proved to be more effective than only positive ones by Huang et al. [72]. Use auxiliary tasks like affordance prediction to narrow the abstraction gap, and combine VLMs outputs with filtering mechanisms, as in Ahn et al. [3].

Diffusion models for planning and exploration

Diffusion models offer strong distribution modeling and short-term planning, but suffer from: (i) **in-context understanding** [10,79,83–85,90,92,94,137], (ii) **multi-modality extension** [20,78,80,81,87,99], and (iii) **poor reasoning** [76,77,82,86,89,95,96].

Recommendation: Embed hierarchical control into diffusion models to manage long-horizon planning [184]; integrate LLMs for reasoning levels with low-level diffusion-based modules. Combine diffusion models with RL through receding-horizon approaches for stable planning, as shown in recent approaches [10]. Guide diffusion models generation by conditioning them on state information or meaningful variables [165].

World models and video prediction for environment modeling

World models and video prediction models give agents the ability to "imagine" future without actually interacting with the environment, which is particularly helpful in situations with limited data. However, they have drawbacks: (i) **restricted generalization** [18,48,107,114,185], (ii) **compounding errors over extended rollouts** [104,106,111,116,186], and (iii) misalignment between actionable state representations and visual accuracy [103,105,108,110,115]. Similar to this, video prediction algorithms try to predict changes at the pixel level, but they frequently produce (i) **misaligned forecasts** [109] and have (ii) **poor physically significant dynamics** [112] over time.

Recommendation: Use foundation world models [103] rather than video predictors for more robust representation learning. Employ stochasticity and uncertainty modeling to mitigate compounding errors in prediction. Augment world models with high-fidelity sensory data [105,115]. Hybrid frameworks combining analytical simulators with learned models (*e.g.*, [187]) can improve realism and transferability. Fine-tune video prediction models on task-specific environments and prefer them only for data augmentation or internal reward shaping.

Scalability and resource demand of foundation models in RL training

Recent research draw attention to the difficulties of incorporating foundation models straight into the RL training loop, where RL training have (i) **memory limitations** [131,152,186] and foundation models have (ii) **high inference costs** [39,40,43] that reduce efficiency on complex tasks.

Recommendation: Give priority to modular architectures, such as Ma et al. [8]. Deploy distilled RL policies for low-latency real-world control and use foundation models mainly for offline learning and reasoning in the RL training loop. Create hybrid pipelines in which smaller controllers make decisions in real time, while LLMs or VLMs offer symbolic reasoning.

9.2. Challenges in RL for generative policies

RL pre-training of generative policies is widely studied and shares challenges with classic RL training as described in prior work [22,23,188], we first focus on *RL fine-tuning*, which presents new and specific open challenges unique to generative policies. While, unpredictability, safety verification, and robustness under uncertainty are inherent challenges that are common to both pre-training and fine-tuning. We highlight key open problems in these directions.

Policy agnostic RL fine-tuning

Fine-tuning of generative policies remains a critical challenge for deploying policies in real-world or dynamic environments. The main challenges are: (i) **obtaining new fine-tuning RL methods** and **fine-tuning methods independent from the backbone model** [28,152,179,180], and (ii) **achieving good fine-tuning for any size of the policy** [167].

Recommendation: Investigate new specific methods for efficient RL fine-tuning of VLA models [45,167], or smaller diffusion models [180]. Develop methods that are agnostic from the policy architecture, such as in Mark et al. [45].

Online RL fine-tuning of VLA models

Four major obstacles stand in the way of fine-tuning VLA models with RL: (i) **data gathering is expensive in real-world**, and VLA models are usually trained with real-world data through IL, limiting GPU parallelization in simulation [25,171]. It is (ii) **computationally demanding to conduct online training** [167], and it is (iii) **difficult to assess VLA model performance** [172]. They also suffer from (iv) **catastrophic forgetting** [131,152,186].

Recommendation: Leverage realistic simulations for massive training and few-shot real-world fine-tuning to close the gap between simulation and reality, following new research trends in this direction [175–178]. Integrate offline RL pipelines to minimize the need for online training. Use distillation to condense huge VLA models into lightweight, task-specific policies that allow for effective RL deployment [182]. Test in simulations the VLA models using frameworks like [172]. Combine replay-based continual RL with regularization to mitigate forgetting [185,186].

Safety and failure modes of generative policies

Generative policies present serious safety and reliability issues in robotics, due to their black-box nature and absence of clear constraints. They may behave in an unpredictable manner. Challenges are: (i) **closed-loop verification** and (ii) **real-time monitoring** [21,38,99,149–152].

Recommendation: Incorporate safety-aware control modules that monitor and oversee the outputs of generative policies, to enforce physical constraints, and direct policy correction in the face of uncertainty [132,189,190]. Create closed-loop systems that use environment feedback to continually test and improve policy behavior, and implement hybrid architectures where generative policies are constrained by explicit safety layers or fallbacks [191,192]. Apply verification techniques such as control barrier functions [193].

9.3. Future research directions

We highlight three promising research directions based on our findings.

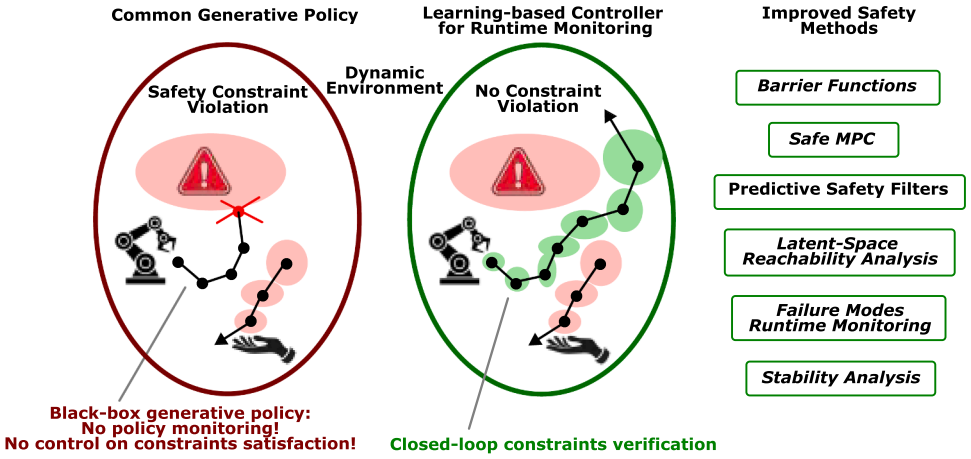


Fig. 8. Safety in generative policies. This figure illustrates the challenge of deploying a black-box generative policy in real-world dynamic environments without access to model-based information.

9.3.1. *RL from human feedback*

Techniques such as RL from Human Feedback (RLHF) and related approaches (e.g., Preference Based RL [194]) have been used for aligning large language models and other generative systems to produce outputs that are both useful and aligned with human input[28,195]. Future research should explore RLHF to fine-tune generalist generative policies—such as VLA models. However, evaluating robotic actions remains harder than assessing tasks like chatbot responses.

9.3.2. *Actor-critic foundation models*

A new research direction could be to develop specialized foundation models that act as critics in RL or generate in-context rewards zero-shot in real-world settings for policy training. Traditional RL depends on manually designed rewards, which can be limiting. Instead, specific foundation models can dynamically assess actions using multi-modal inputs, improving learning efficiency and adaptability with respect to generic LLMs [8].

9.3.3. *Constraint-aware generative models with optimal control*

Based on recent advances in model-based training of IL policies [89,191] and extending our discussion on safety in generative policies (Section 9), we propose that integrating constraint satisfaction methods from optimal control theory, such as control barrier functions, into RL presents a promising direction for safer and more grounded diffusion policies (as suggested in Fig. 8). Moreover, adaptive learning-based techniques that dynamically respond to changing environments can significantly enhance the deployment of diffusion policies across various control tasks, much like model predictive safety filters [196]. Our argument is further supported by recent works leveraging diffusion models within a receding horizon framework for control action execution [19,21,122,196–199], which integrate model predictive control frameworks obtaining stronger stability.

10. Conclusion

The integration of generative AI models and RL marks a transformative advancement in robotics, enhancing reasoning and adaptability in complex tasks [5–9,11,54,167]. This synergy combines the extensive knowledge and in-context learning of foundation models, the generative capabilities on multi-modal input fusion of generative AI and the high learning capacity in decision-making of RL in robots. Our review critically examined this integration, with a focus on duality aspects between generative AI and RL for robot action generation; extracting policies from text, video and states information. We first studied how *Generative Tools for RL* consist in a solid portion of research papers where LLMs and VLMs play a major role, and how the union of diffusion models and RL is a very exciting trend in this direction, that deserves further investigation. Moreover, the exploration of how RL can harness foundation world models to enhance model-based RL training further highlights the potential of this integration. Finally, we categorized preliminary work on the other side of our duality investigation—*RL for Generative Policies*—and we focused on combining generative AI with RL in training, fine-tuning, and distilling generative policies. While the opportunities for such a research field are vast, we highlighted remaining challenges, notably in grounding abstract representations of foundation models to physical world concepts—an area needing significant exploration. We concluded that RL-based fine-tuning of robotic generative policies, along with the exploration of failure modes of black-box policies, remains underexplored. While, the scalability and computational demands of these systems pose substantial challenges, especially in resource-constrained environments, limiting the possibility of online training. Addressing these challenges is critical for building fully autonomous robots, controlled by generative AI models able to create policies from multi-modal inputs. Based on our findings, we shared promising research directions for the coming years.

Limitations and future work

Despite our effort to provide a comprehensive and balanced overview, several limitations remain. First, the review focuses primarily on transformer- and diffusion-based architectures, excluding other generative paradigms such as flow-matching or autoregressive audio-visual models. Second, the inclusion of preprints—while necessary for such a rapidly evolving field—introduces potential volatility in the reported findings. Third, the survey period concludes in 2025; subsequent developments may further reshape this landscape. Future research should prioritize unified benchmarking protocols across modalities, quantitative fusion-efficiency metrics, and evaluation of safety and interpretability in RL-enhanced generative policies. Another promising avenue lies in lightweight, on-device fusion operators that bridge high-level reasoning and low-level control under real-time constraints.

CRediT authorship contribution statement

Angelo Moroncelli: Writing – review & editing, Writing – original draft, Visualization, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Vishal Soni:** Writing – review & editing, Writing – original draft, Visualization, Software, Investigation; **Marco Forgione:** Writing – review & editing; **Dario Piga:** Writing – review & editing; **Blerina Spahiu:** Writing – review & editing, Supervision, Methodology, Conceptualization; **Loris Roveda:**

Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Data availability

No data was used for the research described in the article.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the author(s) used ChatGPT in order to check and improve the grammar and clarity of the text. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank the Hasler Foundation for providing funding to carry out our research. Our work was partially supported by the GeneRAI (Generative Robotics AI) project, funded by Hasler Stiftung.

References

[1] V. Cohen, et al., A survey of robotic language grounding, IJCAI (2024) 7999–8009.
[2] T. Carta, et al., Grounding large language models in interactive environments with online reinforcement learning, in: ICML, PMLR, 2023, pp. 3676–3713.
[3] M. Ahn, et al., Do as i can, not as i say: grounding language in robotic affordances, arXivpreprintarXiv:2204.01691 (2022).
[4] W. Huang, et al., Grounded decoding: guiding text generation with grounded models for embodied agents, in: NeurIPS, 2023.
[5] T. Xie, et al., Text2reward: reward shaping with language models for reinforcement learning, in: ICLR, 2023.
[6] S. Liu, et al., RL-GPT: integrating reinforcement learning and code-as-policy, NeurIPS 37 (2024).
[7] Y. Wang, et al., RL-VLM-F: reinforcement learning from vision language foundation model feedback, in: ICML, 2024.
[8] Y.J. Ma, et al., Eureka: human-level reward design via coding large language models, arXivpreprintarXiv:Arxiv-2310.12931 (2023).
[9] Y.J. Ma, et al., Dreureka: language model guided sim-to-real transfer, RSS (2024)
[10] Z. Zhu, et al., Diffusion models for reinforcement learning: a survey, arXivpreprintarXiv:2311.01223 (2023).
[11] T. Huang, et al., Diffusion reward: learning rewards via conditional video diffusion, ECCV (2024).
[12] R. Bommasani, et al., On the opportunities and risks of foundation models, arXivpreprintarXiv:2108.07258 (2021).
[13] K. Baumli, et al., Vision-language models as a source of rewards, CoRR:abs/2312.09187 (2023).
[14] W. Chen, et al., Vision-Language models provide promptable representations for reinforcement learning, TMLR (2024).
[15] J. Rocamonde, et al., Vision-language models are zero-shot reward models for reinforcement learning, ICLR (2024).
[16] M. Hassan, et al., GEM: A generalizable ego-vision multimodal world model for fine-grained ego-motion, object dynamics, and scene composition control, CVPR (2025).
[17] B. Chen, et al., Open-vocabulary queryable scene representations for real world planning, in: 2023 IEEE ICRA, 2023, pp. 11509–11522.
[18] D. Ha, J. Schmidhuber, Recurrent world models facilitate policy evolution, NeurIPS 31 (2018).
[19] H. Qi, et al., Strengthening generative robot policies through predictive world modeling, arXivpreprintarXiv:2502.00622 (2025).
[20] H. He, et al., Diffusion model is an effective planner and data synthesizer for multi-task reinforcement learning, NeurIPS 36 (2024).
[21] C. Chi, et al., Diffusion policy: visuomotor policy learning via action diffusion, IJRR 44 (10–11) (2025) 1684–1704.
[22] R.S. Sutton, A.G. Barto, Reinforcement learning: an introduction, IEEE TNN 9 (5) (1998) 1054–1054.
[23] V. Mnih, et al., Playing atari with deep reinforcement learning, NeurIPS Deep Learning Workshop (2013).
[24] R. Firoozi, et al., Foundation models in robotics: applications, challenges, and the future, IJRR (2025).
[25] A. Padalkar, et al., Open X-Embodiment: robotic learning datasets and RT-X models, IEEE ICRA (2024).
[26] R.S. Sutton, Learning to predict by the methods of temporal differences, Mach. Learn. 3 (1988) 9–44.
[27] R. Tian, et al., Maximizing alignment with minimal feedback: efficiently learning rewards for visuomotor robot policy alignment, arXivpreprintarXiv:2412.04835 (2024).
[28] D.M. Ziegler, et al., Fine-tuning language models from human preferences, arXivpreprintarXiv:1909.08593 (2019).
[29] W. Huang, et al., Voxposer: composable 3d value maps for robotic manipulation with language models, CoRL (2023).
[30] M. Shridhar, et al., Cliport: what and where pathways for robotic manipulation, in: CoRL, PMLR, 2022, pp. 894–906.
[31] V. Bhat, et al., Grounding LLMs for robot task planning using closed-loop state feedback, CoRR:abs/2402.08546 (2024).
[32] M. Forgione, et al., From system models to class models: an in-context learning paradigm, IEEE Contr. Syst. Lett. 7 (2023), 3513–3518.
[33] Z. Du, et al., Can transformers learn optimal filtering for unknown systems?, IEEE Contr. Syst. Lett. 7 (2023) 3525–3530.
[34] R. Busetto, et al., In-context learning of state estimators, IFAC 58 (15) (2024) 145–150.
[35] Y. Ma, et al., A survey on vision-language-action models for embodied AI, arXivpreprintarXiv:2405.14093 (2024).
[36] A. Dubey, et al., The llama 3 herd of models, arXivpreprintarXiv:2407.21783 (2024).
[37] J. Achiam, et al., Gpt-4 technical report, arXivpreprintarXiv:2303.08774 (2023).
[38] S. Reed, et al., A generalist agent, TMLR (2022).
[39] Y. Hu, et al., Toward general-purpose robots via foundation models: a survey & meta-analysis, CoRR (2023).
[40] X. Xiao, et al., Robot learning in the era of foundation models: a survey, Neurocomputing 638 (2025), 129963.
[41] H. Zhou, et al., Language-conditioned learning for robotic manipulation: a survey, CoRR (2023).
[42] J. Wang, et al., Large language models for robotics: opportunities, challenges, perspectives, JAI (2024).
[43] Y. Cao, et al., Survey on large language model-enhanced reinforcement learning: concept, taxonomy, and methods, IEEE TNNLS 36 (6) (2025) 9737–9757.
[44] Y. Gui, et al., Conformal alignment: knowing when to trust foundation models with guarantees, NeurIPS 37 (2024).
[45] M.S. Mark, et al., Policy agnostic rl: offline and online rl fine-tuning of any class and backbone, 7th Robot Learning Workshop (2024).
[46] A.A. Rusu, et al., Policy distillation, ICLR (2016).
[47] A. Adeniji, et al., Language reward modulation for pretraining reinforcement learning, in: RLC 2024 Workshop, 2024.
[48] M. Yang, et al., Learning interactive real-world simulators, ICLR (2024).
[49] R. Ma, et al., Guiding exploration in reinforcement learning with large language models, IEEE ICRA (2025) 9011–9017.
[50] Y. Zhang, et al., Motiongpt: finetuned llms are general-purpose motion generators, in: Proceedings of AAAI, 38, (7), 2024, pp. 7368–7376.
[51] J. Sun, et al., Prompt, plan, perform: llm-based humanoid control via quantized imitation learning, in: 2024 IEEE ICRA, IEEE, 2024, pp. 16236–16242.
[52] W. Yu, et al., Language to rewards for robotic skill synthesis, CoRL (2023).
[53] C. Colas, et al., Augmenting autotelic agents with large language models, in: CLLA, PMLR, 2023, pp. 205–226.
[54] K. Chu, et al., Accelerating reinforcement learning of robotic manipulations via feedback from large language models, CoRR (2023).
[55] Y. Du, et al., Guiding pretraining in reinforcement learning with large language models, in: ICML, PMLR, 2023, pp. 8657–8677.
[56] A. Zeng, et al., Socratic models: composing zero-shot multimodal reasoning with language, ICLR (2022).
[57] D. Zhu, et al., MiniGPT-4: enhancing vision-language understanding with advanced large language models, CoRL (2024).
[58] J. Zhang, et al., Bootstrap your own skills: learning to solve new tasks with large language model guidance, CoRL (2023).
[59] E.S. Lubana, et al., FoMo rewards: can we cast foundation models as reward functions?, NeurIPS 36 (2023).
[60] C. Colas, et al., Language as a cognitive tool to imagine goals in curiosity driven exploration, NeurIPS 33 (2020) 3761–3774.
[61] H. Hu, D. Sadigh, Language instructed reinforcement learning for human-ai coordination, in: ICML, PMLR, 2023, pp. 13584–13598.
[62] M. Dalal, et al., Plan-seq-learn: language model guided rl for solving long horizon robotics tasks, ICLR (2024).
[63] J. Song, et al., Self-refined large language model as automated reward function designer for deep reinforcement learning in robotics, CoRR (2023).
[64] E. Triantafyllidis, et al., Intrinsic language-guided exploration for complex long-horizon robotic manipulation tasks, in: 2024 IEEE ICRA, IEEE, 2024, pp. 7493–7500.
[65] Y. Cui, et al., Can foundation models perform zero-shot task specification for robot manipulation?, in: LDCC, PMLR, 2022, pp. 893–905.
[66] D. Venuto, et al., Code as reward: empowering reinforcement learning with VLMs, ICML (2024).
[67] Y.J. Ma, et al., Liv: language-image representations and rewards for robotic control, in: ICML, PMLR, 2023, pp. 23301–23320.

[68] S. Sontakke, et al., Roboclip: one demonstration is enough to learn robot policies, NeurIPS 36 (2024).
[69] J. Yang, et al., Robot fine-tuning made easy: pre-training rewards and policies for autonomous real-world reinforcement learning, in: 2024 IEEE ICRA, IEEE, 2024, pp. 4804–4811.
[70] N. Di Palo, et al., Towards a unified agent with foundation models, ICLR (2023).
[71] P. Mahmoudieh, et al., Zero-shot reward specification via grounded natural language, in: ICML, PMLR, 2022, pp. 14743–14752.
[72] S. Huang, et al., Understanding and Mitigating Noise in VLM Rewards, arXivpreprintarXiv:2409.15922 (2024).
[73] A. Radford, et al., Learning transferable visual models from natural language supervision, in: ICML, Pmlr, 2021, pp. 8748–8763.
[74] R. Rombach, et al., High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF, 2022, pp. 10684–10695.
[75] P. Esser, et al., Scaling rectified flow transformers for high-resolution image synthesis, in: ICML, 2024.
[76] C. Lu, et al., Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning, in: ICML, PMLR, 2023, pp. 22825–22855.
[77] B. Kang, et al., Efficient diffusion policies for offline reinforcement learning, NeurIPS 36 (2024).
[78] H.J.T. Suh, et al., Fighting uncertainty with gradients: offline reinforcement learning via diffusion score matching, in: CoRL, PMLR, 2023, pp. 2878–2904.
[79] P. Hansen-Estruch, et al., Implicit q-learning as an actor-critic method with diffusion policies, arXivpreprintarXiv:2304.10573 (2023).
[80] W.K. Kim, et al., Robust policy learning via offline skill diffusion, AAAI (2024).
[81] S. Venkatraman, et al., Reasoning with latent diffusion in offline reinforcement learning, ICLR (2024).
[82] J. Hu, et al., Instructed diffuser with temporal condition guidance for offline RL, arXivpreprintarXiv:2306.04875 (2023).
[83] M. Psenka, et al., Learning a diffusion model policy from rewards via Q-Score matching, ICML (2024).
[84] V. Jain, S. Ravanbakhsh, Learning to reach goals via diffusion, ICML (2024).
[85] H. Chen, et al., Offline reinforcement learning via high-fidelity generative behavior modeling, ICLR (2023).
[86] Z. Zhu, et al., Madiff: offline multi-agent learning with diffusion models, NeurIPS 37 (2024).
[87] F. Ni, et al., Metadiffuser: diffusion model as conditional planner for offline meta-rl, in: Icml, PMLR, 2023, pp. 26087–26105.
[88] Z. Wang, et al., Diffusion policies as an expressive policy class for offline reinforcement learning, ICLR (2023).
[89] W. Xiao, et al., Safediffuser: safe planning with diffusion probabilistic models, in: ICLR, 2025.
[90] Z. Liang, et al., Adaptdiffuser: diffusion models as adaptive self-evolving planners, ICML (2023).
[91] K. Lee, et al., Refining diffusion planner for reliable behavior synthesis by automatic detection of infeasible plans, NeurIPS 36 (2024).
[92] S. Zhou, et al., Adaptive online replanning with diffusion models, NeurIPS 36 (2024).
[93] J. Liu, et al., DiPPeR: diffusion-based 2D path planner applied on legged robots, IEEE ICRA (2024) 9264–9270.
[94] J. Brehmer, et al., EDGI: equivariant diffusion for planning with embodied agents, NeurIPS 36 (2024).
[95] W. Li, et al., Hierarchical diffusion for offline decision making, in: ICML, PMLR, 2023, pp. 20035–20064.
[96] C. Chen, et al., Simple hierarchical planning with diffusion, CoRR (2024).
[97] S. Kim, et al., Stitching sub-trajectories with conditional diffusion model for goal-conditioned offline RL, AAAI (2024).
[98] E. Zhang, et al., Language control diffusion: efficiently scaling through space, time, and tasks, in: ICLR, 2024.
[99] M. Janner, et al., Planning with diffusion for flexible behavior synthesis, in: ICML, 2022.
[100] F. Nuti, et al., Extracting reward functions from diffusion models, NeurIPS 36 (2024).
[101] B. Mazoure, et al., Value function estimation using conditional diffusion models for control, arXivpreprintarXiv:2306.07290 (2023).
[102] P. Wu, et al., Daydreamer: world models for physical robot learning, in: CoRL, PMLR, 2023, pp. 2226–2240.
[103] P. Mazzaglia, et al., Multimodal foundation world models for generalist embodied agents, NeurIPS 37 (2024).
[104] A. Zala, et al., EnvGen: generating and adapting environments via LLMs for training embodied agents, in: COLM, 2024.
[105] L. Wang, et al., GenSim: generating robotic simulation tasks via large language models, ICLR (2026).
[106] K. Nottingham, et al., Do embodied agents dream of pixelated sheep: embodied decision making using language guided world modelling, in: ICML, PMLR, 2023, pp. 26311–26325.
[107] J. Wu, et al., VideoGPT: interactive videoGPTs are scalable world models, NeurIPS 37 (2024).
[108] Y. Wang, et al., Robogen: towards unleashing infinite data for automated robot learning via generative simulation, ICML (2024).
[109] A.S. Chen, et al., Learning generalizable robotic reward functions from" in-the-wild" human videos, RSS XVII (2021).
[110] W. Ye, et al., Towards embodied generalist agents with foundation prior assistance, CoRL (2025).
[111] Y. Seo, et al., Multi-view masked world models for visual robotic manipulation, in: ICML, PMLR, 2023, pp. 30613–30632.
[112] A. Escontrela, et al., Video prediction models as rewards for reinforcement learning, NeurIPS 36 (2024).
[113] M. Alakuijala, et al., Learning reward functions for robotic manipulation by observing humans, in: 2023 IEEE ICRA, IEEE, 2023, pp. 5006–5012.
[114] Z. Mao, et al., Zero-shot safety prediction for autonomous robots with foundation world models, CoRR (2024).
[115] J. Bruce, et al., Genie: generative interactive environments, in: ICML, 2024.
[116] Y. Seo, et al., Masked world models for visual control, in: CoRL, PMLR, 2023, pp. 1332–1344.
[117] J.-B. Alayrac, et al., Flamingo: a visual language model for few-shot learning, NeurIPS 35 (2022) 23716–23736.
[118] J. Zhang, et al., Vision-language models for vision tasks: a survey, IEEE Trans. PAMI 46 (8) (2024), 5625–5644.
[119] F. Bordes, et al., An introduction to vision-language modeling, CoRR (2024).
[120] J. Ho, T. Salimans, Classifier-free diffusion guidance, in: NeurIPS Workshop on Deep Generative Models, 2021.
[121] Y. Du, et al., Learning universal policies via text-guided video generation, NeurIPS 36 (2023) arXiv–2302.
[122] G. Zhou, et al., Diffusion model predictive control, TMLR (2024).
[123] J. Beck, et al., A survey of meta-reinforcement learning, arXivpreprintarXiv:2301.08028 (2023).
[124] C. Bhateja, et al., Robotic offline rl from internet videos via value-function pre-training, 2024 IEEE ICRA (2024) 16977–16984.
[125] A. Kumar, et al., Pre-training for robots: offline RL enables learning new tasks in a handful of trials, RSS (2022).
[126] O.Y. Lee, et al., Affordance-guided reinforcement learning via visual prompting, IROS (2025).
[127] S. Nair, et al., Learning language-conditioned robot behavior from offline data and crowd-sourced annotation, in: CoRL, PMLR, 2022, pp. 1303–1315.
[128] A. Ramesh, et al., Zero-shot text-to-image generation, in: ICML, Pmlr, 2021, pp. 8821–8831.
[129] Y.J. Ma, et al., Vip: towards universal visual reward and representation via value-implicit pre-training, ICLR (2023).
[130] A. Majumdar, et al., Where are we in the search for an artificial visual cortex for embodied intelligence?, NeurIPS 36 (2024).
[131] M. Wolczyk, et al., On the role of forgetting in fine-Tuning reinforcement learning models, in: ICLR Workshop on Reincarnating Reinforcement Learning, 2023.
[132] S. Gu, et al., A review of safe reinforcement learning: methods, theory and applications, IEEE TPAMI 46 (12) (2024) 11216–11235.
[133] L. Mezghani, et al., Think before you act: unified policy for interleaving language reasoning with actions, ICLR (2023).
[134] L. Janson, et al., Deterministic sampling-based motion planning: optimality, complexity, and performance, IJRR 37 (1) (2018) 46–61.
[135] F. Duchoň, et al., Path planning with modified a star algorithm for a mobile robot, Procedia Eng. 96 (2014) 59–69.
[136] C. Zhou, et al., A review of motion planning algorithms for intelligent robots, J. Intell. Manuf. 33 (2) (2022) 387–424.
[137] Z. Wang, et al., Cold diffusion on the replay buffer: learning to plan from known good states, in: CoRL, PMLR, 2023, pp. 3277–3291.
[138] J. Fu, et al., D4rl: Datasets for deep data-driven reinforcement learning, arXivpreprintarXiv:2004.07219 (2020).
[139] T. Yu, et al., Meta-world: a benchmark and evaluation for multi-task and meta reinforcement learning, in: CoRL, PMLR, 2020, pp. 1094–1100.
[140] E. Todorov, et al., MuJoCo: a physics engine for model-based control, in: 2012 IEEE/RSJ IROS, 2012, pp. 5026–5033.
[141] A. Gupta, et al., Relay policy learning: solving long horizon tasks via imitation and reinforcement learning, CoRL (2019).
[142] S. James, et al., Rlbench: the robot learning benchmark & learning environment, IEEE RA-L 5 (2) (2020) 3019–3026.
[143] K. Grauman, et al., Ego4d: around the world in 3,000 hours of egocentric video, in: Proceedings of the IEEE/CVF, 2022, pp. 18995–19012.
[144] O. Mees, et al., CALVIN: a benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks, IEEE RA-L 7 (3) (2022) 7327–7334.
[145] S. Dasari, et al., Robonet: large-scale multi-robot learning, in: CoRL, 2019.
[146] Y. Narang, et al., Factory: fast contact for robotic assembly, in: RSS, 2022.
[147] A. Miech, et al., HowTo100M: learning a text-video embedding by watching hundred million narrated video clips, in: ICCV, 2019.
[148] J. Gu, et al., ManiSkill2: a unified benchmark for generalizable manipulation skills, in: ICLR, 2023.
[149] S. Hu, et al., HarmoDT: harmony multi-task decision transformer for offline reinforcement learning, in: ICML, 235 of *PMLR*, 2024, pp. 19182–19197.
[150] A. Bucker, et al., Latte: language trajectory transformer, in: 2023 IEEE ICRA, IEEE, 2023, pp. 7287–7294.
[151] P.-H. Li, et al., Online foundation model selection in robotics, CoRR (2024).
[152] M. Wołczyk, et al., Fine-tuning reinforcement learning models is secretly a forgetting mitigation problem, ICML (2024).
[153] L. Chen, et al., Decision transformer: reinforcement learning via sequence modeling, NeurIPS 34 (2021) 15084–15097.
[154] M. Wen, et al., Multi-agent reinforcement learning is a sequence modeling problem, NeurIPS 35 (2022) 16509–16521.
[155] M. Xu, et al., Hyper-decision transformer for efficient online policy adaptation, ICLR (2023).
[156] R. Bonatti, et al., Pact: perception-action causal transformer for autoregressive robotics pre-training, in: 2023 IEEE/RSJ IROS, IEEE, 2023, pp. 3621–3627.
[157] Y. Chebotar, et al., Q-transformer: scalable offline reinforcement learning via autoregressive q-functions, in: CoRL, PMLR, 2023, pp. 3909–3928.

[158] B. Trabucco, et al., Anymorph: learning transferable polices by inferring agent morphology, in: ICMLl, PMLR, 2022, pp. 21677–21691.
[159] M. Wen, et al., Large sequence models for sequential decision-making: a survey, Front. Comput. Sci. 17 (6) (2023) 176349.
[160] X. Huang, et al., DiffuseLoco: real-time legged locomotion control with diffusion from offline datasets, CoRL (2025).
[161] Z. Li, et al., Beyond conservatism: diffusion policies in offline multi-agent reinforcement learning, CoRR (2023).
[162] S. Hegde, et al., Generating behaviorally diverse policies with latent diffusion models, NeurIPS 36 (2023) 7541–7554.
[163] H. Chen, et al., Score regularized policy optimization through diffusion behavior, ICLR (2024).
[164] Z. Ding, C. Jin, Consistency models as a rich and efficient policy class for reinforcement learning, ICLR (2024).
[165] A. Ajay, et al., Is conditional generative modeling all you need for decision-making?, in: ICLR, 2023.
[166] L. Yang, et al., Policy representation via diffusion probability model for reinforcement learning, arXivpreprintarXiv:2305.13122 (2023).
[167] J. Hu, et al., FLaRe: achieving masterful and adaptive robot policies with large-scale reinforcement learning fine-tuning, CoRR:abs/2409.16578 (2024).
[168] A. Brohan, et al., Rt-1: robotics transformer for real-world control at scale, RSS XIX (2023).
[169] A. Brohan, et al., Rt-2: vision-language-action models transfer web knowledge to robotic control, CoRL (2023) 2165–2183.
[170] M. Oier, et al., Octo: an open-source generalist robot policy, in: RSS, 2024.
[171] M.J. Kim, et al., OpenVLA: an open-source vision-language-action model, CoRL (2025).
[172] X. Li, et al., Evaluating real-world robot manipulation policies in simulation, CoRL (2025).
[173] Y. Guo, et al., Improving vision-language-action model with online reinforcement learning, arXivpreprintarXiv:2501.16664 (2025).
[174] J. Liu, et al., What can RL bring to VLA generalization? an empirical study, NeurIPS 35 (2025).
[175] H. Zang, et al., RLinf-VLA: a unified and efficient framework for VLA + RL training, arXivpreprintarXiv:2510.06710 (2025).
[176] S. Tan, et al., Interactive post-training for vision-language-action models, in: Workshop on Foundation Models Meet Embodied Agents at CVPR 2025, 2025.
[177] H. Li, et al., Vla-rft: vision-language-action reinforcement fine-tuning with verified rewards in world simulators, arXivpreprintarXiv:2510.00406 (2025).
[178] G. Lu, et al., Vla-rl: towards masterful and general robotic manipulation with scalable reinforcement learning, arXivpreprintarXiv:2505.18719 (2025).
[179] Y. Chen, et al., FDPP: fine-tune diffusion policy with human preference, arXivpreprintarXiv:2501.08259 (2025).
[180] A.Z. Ren, et al., Diffusion policy policy optimization, in: CoRL Workshop on Mastering Robot Manipulation in a World of Data, 2024.
[181] F. Permenter, C. Yuan, Interpreting and improving diffusion models from an optimization perspective, in: ICML, JMLR.org, 2024.
[182] T. Jülg, et al., Refined policy distillation: from VLA generalists to RL experts, IROS (2026).
[183] C. Xu, et al., RLDG: robotic generalist policy distillation via reinforcement learning, RSS (2025).
[184] S. Liu, et al., DASP: hierarchical offline reinforcement learning via diffusion autodecoder and skill primitive, IEEE RA-L 10 (2) (2025).
[185] B. Chan, et al., Offline-to-online reinforcement learning for image-based grasping with scarce demonstrations, in: CoRL Workshop on Mastering Robot Manipulation in a World of Abundant Data, 2024.
[186] G.M. Van de Ven, et al., Continual learning and catastrophic forgetting, arXivpreprintarXiv:2403.05175 (2024).
[187] G. Authors, Genesis: A Universal and Generative Physics Engine for Robotics and Beyond, 2024, .
[188] V. Mnih, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.
[189] C. Agia, et al., Unpacking failure modes of generative policies: runtime monitoring of consistency and progress, in: CoRL, 270, PMLR, 2025, pp. 689–723.
[190] S. Ross, D. Bagnell, Efficient reductions for imitation learning, in: Y.W. Teh, M. Titterington (Eds.), ICAIS, 9 of *PMLR*, PMLR, Chia Laguna Resort, Sardinia, Italy, 2010, pp. 661–668.
[191] R. Pérez-Dattari, et al., PUMA: deep metric imitation learning for stable motion primitives, Adv. Intell. Syst. 6 (11) (2024) 2400144.
[192] B. Zhang, et al., SafeVLA: towards safety alignment of vision-language-action model via constrained learning, arXivpreprintarXiv:2503.03480 (2025).
[193] K. Nakamura, et al., Generalizing safety beyond collision-avoidance via latent-space reachability analysis, RSS (2025).
[194] Y. Abdelkareem, et al., Advances in preference-based reinforcement learning: a review, in: 2022 IEEE SMC, 2022, pp. 2527–2532.
[195] J. Luo, et al., Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning, Sci. Rob. 10 (105) (2025), eads5033.
[196] K.P. Wabersich, M.N. Zeilinger, A predictive safety filter for learning-based control of constrained nonlinear dynamical systems, Automatica 129 (2021) 109597.
[197] W. Zhao, et al., VLMPC: vision-language model predictive control for robotic manipulation, in: RSS, 2024.
[198] N. Hansen, et al., TD-MPC2: scalable, robust world models for continuous control, ICLR (2024).
[199] H. Xue, et al., Full-order sampling-based MPC for torque-level locomotion control via diffusion-style annealing, IEEE ICRA (2025).