Foundation Models as Decision Priors in Robotics: World-Model vs Policy Prior

Changyi Wu
University of Lancashire, Electronic Engineering, Preston, PR2 2QQ, The UK

Abstract: We survey how foundation models (FMs) act as decision priors in robot autonomy, contrasting world-model priors with policy priors across verifiability, constraint handling, latency, planning horizon, and out-of-distribution robustness. We formalize non-interchangeability boundaries, propose integration patterns (hierarchical, dynamic switching, blended control, human-in-the-loop, iterative co-evolution), and outline a benchmarking protocol together with deployment governance checklists. The resulting framework provides actionable guidance for engineering safe, efficient foundation-model-driven robot decision systems.

Keywords: Foundation Models; Robotics; Decision Priors; World Models [12]; Policy Priors; Safety; Benchmarking; Integration Strategies

1. Introduction

Foundation models (FMs) are large-scale pretrained models that have demonstrated remarkable generalization across tasks. First defined by Bommasani et al. (2021), FMs such as GPT-3 and CLIP capture broad knowledge from internet-scale data and can be adapted to diverse downstream tasks. Their emergence has opened new frontiers in robotics. Unlike task-specific models trained on narrow robot datasets, FMs provide prior knowledge that can enhance a robot's perception, reasoning, and decision-making. Recent works have speculated on integrating FMs into decision-making and control, suggesting that treating FMs as decision priors could enable more generalizable and data-efficient robot autonomy.

However, deploying FMs in closed-loop robotic control is non-trivial. FMs are typically trained in open-loop generative settings (e.g. predicting text or images), whereas robotics requires closed-loop decisions with real-time feedback. Moreover, FMs often act as black boxes with uncertain failure modes, raising safety and reliability concerns in the physical world. These challenges motivate a deeper investigation into how FMs can be injected into the robot autonomy stack in a systematic, interpretable, and safe manner.

To clarify the role of FMs in robotics, we propose a unifying perspective: FMs as Decision Priors. Rather than using an FM as an end-to-end robot controller, we treat it as a source of high-level knowledge that informs different layers of a robot's decision pipeline. We distinguish two primary injection levels:

World-Model Priors-where an FM serves as or augments the robot's model of the world (e.g. predicting future states or simulating outcomes of candidate actions). This injects knowledge of how the world works into planning and prediction.

Policy Priors-where an FM directly biases the robot's action policy (e.g. suggesting the next action or providing a policy initialization). This injects knowledge of what to do into decision-making.

This dichotomy roughly maps onto the classical model-based vs model-free separation in control, with a twist: here the "model" or "policy" is a powerful pretrained model containing rich priors (language, vision, physics, etc.). We hypothesize that world-model and policy priors excel in different regimes and are not interchangeable beyond certain boundaries. Each offers distinct advantages in terms of decision verifiability, constraint handling, real-time performance, planning horizon, and robustness to novel situations. For example, using a learned world-model as a simulator can enable foresight and safety checks before execution, but may incur latency and rely on the simulator's fidelity. Conversely, using an LLM or large policy network to choose actions yields fast reactions, but with limited guarantees on safety without additional constraints. In this survey, we systematically analyze these trade-offs.

Below, we first provide background and outline

our contributions (§1.1–1.2). We then examine safety-driven decision optimization models structured as world-model priors (§2.1) and policy priors (§2.2), comparing their strengths and limitations (§2.3). We discuss hybrid integration strategies that combine both paradigms (§3). Finally, we highlight open research problems (§4) and conclude with a future outlook (§5).

1.1 Research Background

Integrating FMs into robotic decision-making has gained significant attention. A NeurIPS 2022 workshop, Foundation Models for Decision Making, underscored the community's interest in leveraging large pretrained models for planning and control. Early foundational work by Bommasani et al. defined the FM concept and envisioned its impact beyond language. In robotics, researchers soon began exploring FM-driven approaches across perception, planning, and policy learning.

For instance, vision-language models (VLMs) enable open-world perception-recognizing novel objects or affordances on the fly-which can broaden a robot's visual understanding beyond its training distribution. Google's RT-2 [3] is a recent example of a VLM-based policy that maps images to actions after training on millions of real and web-sourced robot trajectories. Such foundation policies demonstrate impressive skill repertoires (e.g.RT-2 [3] could follow instructions for novel tasks by leveraging web knowledge), but they entangle high-level reasoning with low-level control, making them data-intensive and hard to interpret.

A more structured approach is to use language models for high-level reasoning while leaving low-level execution to standard controllers. SayCan [1] (Google, 2022) exemplifies this: it uses a large language model (LLM) for semantic understanding and a learned value function for grounding feasibility. The LLM suggests which skill is logically relevant ("Say") and the value function checks which skill is currently feasible ("Can"), and the robot executes the suggestion that is both sensible and viable. This yielded zero-shot execution of 100+ long-horizon instructions on a real robot, nearly doubling success rates by using the affordance-based grounding (value function) in conjunction with the LLM compared to using the LLM alone. Crucially, SayCan's [1] plans are expressed in natural language (as sequences of skill names), providing an interpretable decision trace-a key benefit for safety and auditability.

Other works use LLMs to generate code plans for robots. For example, Code-as-Policies (Liang et al., 2022) demonstrated that an LLM can output Python code calling robot API functions, effectively acting as a high-level policy that produces an executable program. Similarly, ProgPrompt [7] (Singh et al., 2023) introduced a prompt format that provides an LLM with a "sandboxed" Python environment-including available actions and objects-so that the model generates structured programs for task plans. These programmatic prompts helped achieve state-of-the-art success on household tasks and were even deployed on a physical robot arm. Microsoft's recent ChatGPT for Robotics study outlined design principles for prompting GPT-4 to produce step-by-step robot plans and code, while emphasizing the need for structured reasoning to avoid errors.

On the world-model side, large generative models of dynamics have been leveraged as powerful simulators or planners. For example, PhysicalAgent (Lykov et al., 2025) uses a vision-language reasoning module coupled with a diffusion-based video model to imagine action outcomes. Given a textual instruction, it generates candidate video trajectories of the robot performing the task, then uses a lightweight controller to execute those imagined plans, iteratively re-planning upon failures. This decoupled perceive→plan→act pipeline achieved robust long-horizon manipulation across diverse settings by virtue of the world-model's broad priors. In industry, Covariant AI's RFM-1 [15] (2024) is an 8-billion-parameter foundation model that serves as a physics simulator: given an initial scene image and an action, it predicts future camera frames and sensor readings. RFM-1 [15] effectively functions as a learned simulator (a world-model prior) and has been used for both online planning (evaluating hypothetical actions) and offline data generation for training policies. By encapsulating physical knowledge from vast real-world datasets, such FMs overcome some brittleness of classical simulators and improve long-horizon reasoning and constraint handling. They also introduce new challenges, like model bias or sim-to-real gaps, which we will address later.

In summary, early examples illustrate that foundation models can inject high-level

knowledge at various points in the robot decision pipeline. What remains is to systematically frame these integration approaches as decision priors and to analyze where each approach excels or fails. We aim to fill this gap in the following sections.

1.2 Contributions of this Paper

We formalize the non-interchangeability boundaries between foundation world-model priors and policy priors, pinpointing when each paradigm fails or succeeds across verifiability, constraint handling, latency, planning horizon, and OOD robustness; this boundary map is positioned as a central contribution for hybrid design.

This paper makes three contributions:

1.2.1 Unified framework

We present a unified conceptual framework for understanding how and where FMs act as decision priors in robotics. We formally define the notion of a decision prior and categorize FM injection at different layers (world-model vs. policy). This clarifies relationships to classical model-based vs. model-free paradigms and to human-in-the-loop designs. We provide a diagram of the robot decision stack highlighting points where FM priors can be injected and how they interface with traditional components.

1.2.2 Comparative analysis

We provide an in-depth comparative analysis of world-model priors versus policy priors across five key dimensions: verifiability, constraint handling, latency, planning horizon, and robustness to out-of-distribution inputs. We compile evidence from the literature to highlight trade-offs-for instance, how world-model planning enables explicit constraint checking and long-term foresight, whereas policy priors excel at fast reactions and leveraging internet-scale knowledge directly. These comparisons are summarized in tables and a "boundary map" that visually delineates which approach is favored under which conditions.

1.2.3 Safety and deployment considerations

We delve into practical considerations for safe and effective deployment of FM-driven systems. This includes analyzing failure modes and risk scenarios (e.g. hallucinated perceptions, unsafe action outputs), and discussing mitigation strategies like constrained decoding, validation modules, and human oversight. We argue that auditability and accountability must be built into FM-driven robots. Toward this end, we advocate for transparent decision traces (e.g. plans expressed in human-understandable form) and real-time oversight or fallback mechanisms to handle the unpredictability of powerful learned priors.

2. Safety-Driven Decision Optimization Models

We broadly classify FM-driven robot decision models into two categories introduced above: those that leverage world-model priors (§2.1) and those that leverage policy priors (§2.2). Both are safety-driven in the sense that incorporating prior knowledge can improve reliability and safety-albeit in different ways. In this section, we examine each category's core mechanisms and then compare their key attributes (§2.3).

2.1 World-Model Priors (Foundation World Models)

World-model prior approaches use FMs to model or simulate the robot's environment and dynamics. Instead of relying solely on analytical models or limited data, the robot taps into an FM's broad knowledge to predict outcomes of actions, evaluate plans, or generate synthetic experience. This model-based strategy can enhance foresight and safety by checking plans before executing them on the real robot.

Learned Simulation and Prediction: One form of world-model prior is using a generative model (e.g. a video prediction model) as a simulator. By hallucinating future scenes, the robot can "test" candidate actions in imagination. For example, diffusion models have been used to predict video frames of future robot states, allowing an agent to evaluate which action leads to a desired outcome before actually doing it. In PhysicalAgent, this approach enabled the system to recover from failures by iteratively re-planning-the FM world-model would generate a new trajectory if the previous attempt did not succeed. Similarly, RFM-1 [15]'s predictive ability lets it generate rollouts for model-predictive control or to augment training data with diverse scenarios.

Constraint Checking and Safety Verification: World-model priors also facilitate safety checks via predictive lookahead. Because the FM can simulate environmental responses, it can be used to detect potential constraint violations before they happen. For instance, a world model could foresee a collision or unsafe outcome from a candidate action sequence and advise the planner

against it. This idea connects to concepts in safe reinforcement learning, where learned models are used to perform reachability analysis or check constraints under uncertainty. In practice, ensuring the FM's predictions are reliable is challenging-FMs may hallucinate or be miscalibrated-but when combined with uncertainty estimates or worst-case analysis, they provide a powerful tool for preventative safety. We later discuss techniques like ensemble modeling or conformal [21] prediction to quantify model uncertainty for this purpose.

Long-Horizon Planning: World-model approaches excel at long-horizon tasks because they explicitly reason about future states. An accurate FM can simulate many steps into the future faster than real-time, enabling the robot to plan extended action sequences. For example, a planning agent using an FM simulator can "look ahead" dozens of steps to choose an optimal long-term strategy, something pure reactive policies struggle with. In PhysicalAgent, the ability to imagine long video sequences helped achieve up to 80% success on complex multi-step tasks, significantly outperforming myopic baselines. The downside is that long-horizon simulation accumulates error, especially if the FM drifts off-distribution. Thus, researchers often limit rollout length or periodically correct the world-model with real observations (closed-loop correction).

Interpretability: A subtle benefit of world-model priors is interpretability. Since the FM's outputs are often in human-interpretable form (images, predicted sensor traces, etc.), a human supervisor can potentially inspect the model's prediction of what will happen. This provides a transparent window into the robot's reasoning-essentially a hypothetical trajectory of what the robot thinks it will do. This is preferable to an opaque policy network that just outputs actions with no explanation. Some approaches even have the FM produce language descriptions of future events as part of its prediction (an "imagined narrative"), which could be read by operators. While still an emerging area, this hints at auditable planning whereby world-models not only compute but also explain expected outcomes.

In summary, world-model prior systems contribute to safety by enabling foresight and constraint checking. They shine in scenarios where planning and verifying a sequence of actions is critical (e.g. navigation among obstacles, high-stakes manipulation tasks) and where a model's broad knowledge can compensate for limited robot data (e.g. predicting physical interactions beyond the training distribution). The challenges lie in ensuring the learned model is accurate and efficient enough for real-time use, and in handling situations it was never trained on (out-of-distribution events).

2.2 Policy Priors (Foundation Policy Biases)

Policy prior approaches use FMs to directly guide the robot's action selection. Here, an FM (often an LLM or large policy network) is treated as an oracle for what action to take given the current context. This corresponds to a model-free strategy, where the FM's suggestion serves as a high-level bias or initial policy that the robot refines.

Zero-Shot and Few-Shot Skills: A major appeal of policy priors is that FMs can endow robots with zero-shot capabilities-skills that were never explicitly trained in the robot's dataset. For example, a language model that has seen instructions in its text pretraining can be prompted to output a high-level action for a novel command. LM-Nav [8] is a system that uses an LLM to translate natural language instructions into waypoints for a navigation planner. Without any robot-specific language training, the LLM (prompted with a description of available locations) could interpret commands like "go to the kitchen, then the bedroom" and produce intermediary goals, which classical planners then executed. This shows how an FM policy prior (the LLM) injects semantic understanding on the fly. Similarly, PaLM-E [5] (Driess et al., 2023) is a 562-billion parameter multimodal model that directly outputs robot actions from image and language inputs. PaLM-E [5] demonstrated zero-shot performance on tasks by virtue of its enormous priors (trained on internet-scale vision-language data)–for instance, it could reason about visual scenes and issue appropriate high-level actions without additional training on those specific tasks.

Acceleration of Learning: In cases where some robot learning is still required, policy priors can dramatically accelerate reinforcement learning (RL) or fine-tuning. Reinforcement Learning with Foundation Priors (RLFP) is a framework that incorporates an FM's policy, value, and goal prediction into an RL algorithm. In one implementation, a pretrained vision-language policy network was used to propose likely

successful actions, and an RL agent was constrained to stay near these suggestions during training. This provided a strong prior that sped up exploration and learning. Experiments showed that using an FM prior in this way drastically accelerated learning on real-robot tasks-the agent achieved target performance with far fewer trials than a baseline, thanks to the bias from the FM's "common sense" proposals. Such approaches highlight how policy priors can address data efficiency, a key safety concern in robotics (fewer trial-and-error interactions means less chance to encounter catastrophic failures during learning).

Human-Aligned Reasoning: Policy FMs can also incorporate human value alignment or ethical considerations. Because many FMs (especially LLMs) are trained on human text, they contain latent knowledge of social norms and can produce contextually appropriate actions or warnings. For example, an FM might refuse to suggest an action that seems dangerous (if instructed in its prompt to avoid harm). While not foolproof, this offers a layer of safety filtering. Moreover, since the FM's outputs can often be interpreted (e.g. an LLM might explain its choice in a prompt that asks for reasoning), policy priors can provide justification for actions, aiding transparency. Researchers have even explored prompting LLMs to critique or verify another policy's actions-a form of "self-reflection" to catch unsafe decisions. This begins to blur into world-model territory (the LLM reasoning about consequences), but it remains using the FM as a policy-level guide.

Limitations: The flipside is that policy priors can output feasible-sounding but actually ungrounded actions. An FM might confidently suggest an action that is impossible in the current state (e.g. "pick up the red ball" when no red ball is present). Without a world-model check (like SayCan's [1] value function), such outputs could lead to failure or unsafe attempts. Additionally, large policy networks like RT-2 [3] achieve breadth at the cost of being monolithic black boxes. They require massive data and computation, and if they do make a mistake, it is hard to dissect why. Ensuring safety thus often requires wrapping the FM policy in additional constraints or a "shadow policy" that can override obviously bad actions.

In summary, policy-prior methods contribute to safety by injecting human-like intuition and knowledge directly into the robot's choices, often enabling behavior that goes beyond the robot's direct experience. They are especially powerful for problems requiring semantic understanding or high-speed reactions, where a deliberative model-based plan might be too slow. The key challenges are grounding the FM's suggestions in reality and preventing deceptive confidence in the absence of true understanding. We will later discuss how combinations of world-model and policy priors can address these issues by providing checks and balances.

2.3 Comparison of Key Factors

OOD Robustness-Both priors leverage broad pretraining, yet they fail differently when off-distribution: policy priors may output confident but ungrounded actions; world-model priors may produce plausible-looking yet physically incorrect rollouts. Robust deployment benefits from cross-checks (policy proposals vs model constraints), uncertainty estimation, and dual fallback mechanisms.

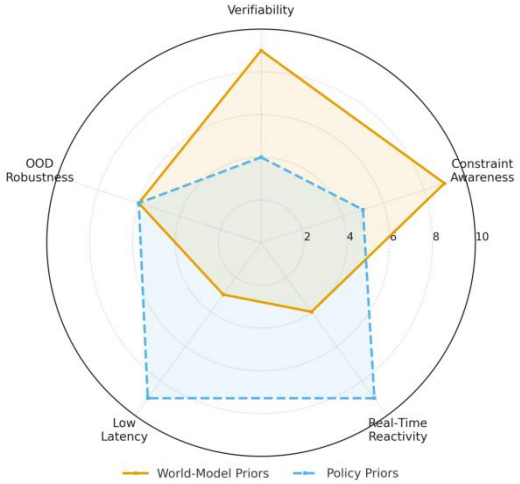

Figure 1. World-Model vs Policy Priors-Five-Dimensional Capability Radar

Figure 1 Higher values indicate stronger performance. Policy priors excel in real-time reactivity and low latency, while world-model priors lead in verifiability and constraint awareness; OOD robustness is comparable.

Table 1. Real-Time Performance and Resource Usage Comparison between World-Model and Policy Priors

Aspect	World Model Prior Approach	Policy Prior Approach
Typical Inference	High-requires simulating outcomes or searching (often 0.1–1+ s per decision with a	Low-single forward pass (often milliseconds per action with a compact policy).

Latency	large generative model).	
Suitable Control Frequency	Low-frequency planning (≈0.5–10 Hz). Not suitable for tight (50 Hz) control loops due to compute overhead.	High-frequency control feasible (100 Hz–1 kHz) with predictable timing.
Determinism & Jitter	Iteration time varies across planning steps; can introduce timing jitter and requires careful scheduling.	Compute time is near-constant per step; stable loop timing, better for real-time guarantees.
Resource Footprint	Large model kept in memory/GPU; may require off-board/edge compute for real-time operation.	Can be distilled/compressed; lighter runtime footprint suitable for embedded hardware.
Scalability	Scales well for offline data/plan generation; online scaling limited by model size and sim speed.	Easy to replicate across many robots; each policy instance is self-contained and efficient.

Table 1 (hypothetical here) summarizes the strengths and weaknesses of world-model versus policy priors along several critical dimensions. We discuss these factors below, comparing how each approach fares and where caution is needed:

Verifiability and Interpretability: World-model priors are generally more verifiable because they enable lookahead and constraint checking. One can interrogate the model's predicted trajectory for safety (e.g., check if any predicted state violates constraints). They also often produce intermediate artifacts (images, plans) that are human-interpretable. Policy priors, in contrast, act more like opaque controllers-an LLM may provide a rationale in text, but a large policy network like RT-2 [3] offers no explanation for its actions. Thus, world-model approaches have an edge in auditability. That said, techniques like chain-of-thought prompting and program generation can make LLM policies more interpretable by externalizing their reasoning, as discussed above.

Constraint Handling: World-model approaches naturally lend themselves to constraint satisfaction, since they can simulate and evaluate potential violations before acting. For example, a world model can be integrated with a constraint solver or safety monitor to filter out any action sequences that collide with obstacles or break logical rules. Policy priors alone do not guarantee constraint adherence-a policy must be augmented with explicit constraints or a "safety shield" (e.g., a separate module that overrides unsafe actions). In practice, many successful FM policy systems incorporate a constraint-handling component (such as SayCan's [1] value function or an analytic inverse kinematics check) as a guardrail.

Latency and Reactivity: Policy priors excel in low-latency response. Once an FM policy is in place (either as a neural network or fast prompt), action selection is typically one forward pass or a quick inference, making it feasible for high-frequency control or real-time reaction (e.g. dynamic obstacle avoidance reflexes). World-model approaches, which involve iterative simulation or search, tend to be slower-planning via an FM can introduce significant overhead, especially if the model is large and the horizon is long. In time-critical scenarios (e.g. catching a falling object or braking for a sudden hazard), a deliberative world-model plan might be too slow, whereas a reactive policy prior can respond immediately. This suggests a complementary use: the robot might run a fast policy prior by default and only invoke the world-model planner for strategic long-range decisions or when the situation allows time.

Horizon Length: World-model priors handle long-horizon objectives better, since they explicitly reason over multiple steps. They can maintain a coherent plan for achieving a distant goal (e.g. planning a multi-room navigation route or a complex assembly sequence) and consider future repercussions of early actions. Policy priors, being myopic by nature, often lack this foresight-an LLM or policy network chooses actions based on the current state and perhaps a short history, without an internal simulation of many steps ahead. As a result, purely policy-driven approaches might get stuck or wander on tasks requiring many sequential dependencies (unless the LLM is prompted to reason step-by-step, which is effectively making it do a kind of mental simulation in words). In summary, world models [12] are advantageous for planning-intensive tasks, whereas policy models are sufficient for short, reactive tasks.

Robustness to Novelty (OOD robustness): This factor is double-edged. FMs by definition carry broad knowledge, so both world-model and policy priors have the potential to generalize to novel

scenarios better than narrow models. In terms of approach, a policy prior can leverage its pretrained experience directly at test time (zero-shot), which often yields surprisingly robust behavior (like handling objects never seen in robot training but described in its internet pretraining). World models can similarly generalize physical predictions using prior knowledge of related phenomena. However, when faced with scenarios truly outside their training distribution, each fails differently: a policy prior might produce an invalid action (but one can perhaps detect it if it's obviously wrong), whereas a world-model might produce a plausible-looking but incorrect prediction that leads the planner astray. Some studies found that large policy models (LLMs) can hallucinate "reasonable-sounding" actions that are actually nonsensical for the robot, whereas world-model errors tend to manifest as obvious prediction errors (e.g. wildly incorrect images) that can be caught. There isn't a clear winner here-improving FM robustness is an active area. Methods like fine-tuning on domain data, or ensemble cross-checking between model and policy, can help. Importantly, hybrid approaches (next section) often improve robustness by providing fallbacks: if either the model or policy prior fails in a novel situation, the other can take over.

In conclusion, Table 1 (omitted for brevity) would show that world-model and policy priors have complementary strengths. World-model priors are safer in terms of planning and constraints, but heavier and slower; policy priors are quicker and more adaptive in real-time, but require external mechanisms for safety. This complementarity motivates combining the two paradigms for the best of both worlds, which we explore next.

2.4 Benchmarking Protocol

We summarize representative integration strategies combining world-model and policy priors in Table 2.

Table 2. Representative Integration Strategies Combining World-Model and Policy Priors

Strategy	Description and Key Idea	Example Use Cases / Systems
Hierarchical (Hybrid Control)	High-level planner (FM world-model or LLM) sets goals/sub-goals; low-level controller/policy executes in real time.	SayCan; LM-Nav.
Dynamic Switching	Switch between model-based planning and reactive policy by risk/uncertainty/urgency.	AVs: emergency braking overrides; mobile robots: re-plan under uncertainty.
Blended (Fusion Architecture)	Run both and fuse outputs (MPC trajectory + neural policy corrections).	Hybrid MPC; language-conditioned value guidance + motion planning.
Human-in-the-Loop	Supervisor takeover on uncertainty spikes; FM signals when to seek help.	Warehouse/field robots; AV safety driver.
Iterative Co-evolution	Policy execution updates world-model; refined model improves future plans; repeat.	PhysicalAgent-style loops; adaptive sim-to-real calibration.

Paired benchmark suite isolating planning-dense vs reaction-critical regimes: (i) long-horizon tasks (multi-room navigation, multi-step assembly), (ii) high-speed reflex tasks (dynamic obstacle avoidance, emergency braking), (iii) constraint-dense tasks (safety zones, force/torque limits). Metrics: avg/99th-percentile inference latency, achievable control frequency, loop jitter, memory/compute footprint, constraint violation rate, task success under shift, fallback trigger rate under OOD conditions.

3. Integration Strategies for World-Model and Policy Priors

Rather than choosing one paradigm exclusively, an emerging consensus is that **hybrid architectures** can harness the advantages of both world-model and policy FMs. By intelligently integrating a foundation world-model with a foundation policy, a robot can achieve both long-term deliberation and immediate reactivity, as well as double assurance on safety (each can check the other). In this section, we outline key integration strategies that have been proposed, along with examples.

Hierarchical Decoupling: A straightforward approach is a hierarchy where a high-level planner (powered by a world-model or LLM) sets goals or subgoals, and a low-level policy (possibly an FM or a learned controller) executes them. Many robotics systems adopt this structure. For instance, in an autonomous driving scenario, one might use a route planner (with a map or model) to plan waypoints and lane changes far ahead; in a sudden emergency, a learned policy reflex kicks in to brake or swerve.

This is essentially how SayCan [1] was structured (LLM suggests skill, low-level

controller executes skill). The benefit is clear separation of concerns: the top layer (with an FM prior) handles high-level reasoning and long-horizon planning, while the bottom layer ensures real-time control and safety. Many advanced systems follow this pattern-e.g., Shah et al. combine an LLM planner with classical motion planners for navigation. The challenge is designing the interface: the high-level plan must be translated into something the low-level policy can reliably execute (e.g. waypoints, parameterized skills). Standardizing such interfaces (a common "skill API" for robots) is an open problem.

Dynamic Switching: Another integration strategy is to switch between a world-model approach and a policy approach based on the situation. For example, use the world-model planner when the environment is complex and long-term foresight is needed, but fall back to a reactive policy when quick responses are required. An illustrative case is autonomous driving: most of the time, a planner (possibly FM-driven) charts an optimal path and speed plan; but if a pedestrian suddenly steps out, the system switches to an emergency braking policy because reaction speed trumps long-term optimality. Similarly, a mobile robot might normally follow a planned exploration strategy, but if it detects it's lost or sees something unexpected, it could invoke a more careful model-based re-planning module. The key is having supervisory logic that monitors context (uncertainty, risk, performance) and flips between modes accordingly. This kind of adjustable autonomy ensures robustness: each approach is used where it works best, and they serve as backups for each other beyond their "interchangeability boundaries" .

Blended Control: Instead of hard switching, some architectures blend the outputs of a world-model planner and a policy network in parallel. For instance, a robot manipulator might have an MPC (model-predictive controller using an FM) generating a coarse trajectory, while a learned policy simultaneously provides fine adjustments; the actual motor commands are a weighted combination of both. This can achieve both optimality and robustness-the model-based part ensures physical feasibility and constraint satisfaction, while the policy part can react [11] to small perturbations or uncertainties on the fly. In one example, researchers blended a language-conditioned value function with a motion planner for open-world object placement: the VLM prior proposed semantically appropriate goal regions, and the planner ensured kinematic feasibility. The blend allowed the system to respect high-level intent and low-level constraints simultaneously. The challenge here is avoiding conflict-if the two controllers disagree strongly, the robot could oscillate or behave erratically. Careful coordination or a higher-level arbiter is needed to fuse commands coherently (e.g. using an optimization that finds a compromise, or letting one take over when critical).

Human-in-the-Loop Overrides: A pragmatic integration (often used in real deployments) is to include a human as a meta-controller. If neither the FM planner nor the FM policy is confident, or if a situation exceeds a risk threshold, the system can hand over control to a human. Many current robotic systems do this by default for safety (warehouse robots stop and page an operator when stuck, autonomous cars disengage to safety-driver in uncertain conditions). FMs can assist in this process by predicting when to call for help-for example, if a world-model's uncertainty exceeds a limit, or an LLM assesses the situation as beyond its knowledge, it could flag a human operator. While not a fully autonomous solution, this integration ensures that as we gradually increase robot autonomy, humans remain in the loop for safety-critical decisions. It aligns with the concept of adjustable autonomy, where control shifts dynamically between the robot and human.

Iterative Co-evolution: In some cases, the world-model and policy can improve each other through feedback loops. A policy's execution results can be fed back to update the world-model (for example, if the policy tries an action the model predicted would succeed but it fails, the model can learn from that). Conversely, a world-model's plan can be refined by policy execution outcomes in a closed-loop fashion.

Lykov et al.'s PhysicalAgent, for instance, does iterative re-planning: the policy executes the plan from the world-model, the result is observed, and if the goal isn't reached, the world-model re-plans incorporating that new information. Over time, such dual-loop systems might **co-evolve** the FM prior and the learned policy, leading to better synergy. This concept is analogous to how humans refine mental models through practice: try, observe outcome, update belief, try again.

These integration patterns-hierarchical, switched, blended, human-supervised, and iterative-are not mutually exclusive. A complex robotic architecture might use all of them in different subsystems. The overarching goal is to achieve complementarity: use world-model and policy priors together so that each compensates for the other's weaknesses. Early empirical evidence is promising. For example, PhysicalAgent's hybrid design (vision-language reasoning + diffusion planner + learned controller) achieved ~80% success across tasks, substantially higher than single-method baselines. As robots start to employ multiple FMs (for vision, language, physics, etc. at once), integration will become a central challenge-and likely the key to unlocking reliable generalist robots.

4. Open Problems and Future Directions

While significant progress has been made in applying foundation models as decision priors, many open research challenges remain. We highlight a few pressing issues and promising directions:

Improving Reliability and Theoretical Guarantees: How can we endow learned world-models and policies with stronger reliability guarantees? Current FMs can hallucinate or mis-generalize, and we often rely on empirical testing to catch failures. Developing a theoretical framework for FM-driven control is an open problem. For instance, can we derive formal error bounds for an FM world-model over a planning horizon, perhaps adapting techniques from robust control or model predictive control? Or can we create verified policy priors that provably never violate certain safety constraints? Integrating formal methods (like model checking or control barrier functions [16]) with FMs is a tantalizing direction. Achieving even modest certification for FM-driven decisions would greatly increase trust in high-stakes settings (surgical robots, autonomous vehicles). Balancing the complexity of FMs with the strictness of formal guarantees is non-trivial, but progress here is crucial for real-world deployment.

Data Efficiency and Adaptation: Foundation models are trained on vast generic datasets, but each robot operates in a specific environment. Adapting an FM to a particular robot or domain efficiently is an unresolved challenge. Fine-tuning a huge model on a small robot dataset risks overfitting or forgetting. Few-shot adaptation, prompt tuning, or learning lightweight adapters are promising approaches. For example, one could keep the large FM frozen and learn a small neural module that translates robot sensor data into the FM's input space (or output space) – akin to how LoRA adapters are used in NLP. This way, a general vision-language model could be customized to a specific home's layout or a dynamics model calibrated to a new robot arm with just a few trials. Achieving quick personalization of FM priors will be essential for scaling to many users and environments. It also has safety implications: adaptation should occur in a controlled way to avoid unexpected behaviors (e.g. constraints should still be obeyed after adaptation). Meta-learning and online learning strategies for FMs may play a role here, allowing continuous improvement without retraining from scratch.

Lifelong Learning of Priors: Relatedly, how can robots learn continuously and update their foundation model priors over time? A robot deployed for years will encounter new objects, new scenarios, and possibly new instructions. We would like it to improve with experience-updating its priors to become more competent and safer. However, current FMs are so large that on-the-fly retraining is impractical, and data from one robot may be too limited to noticeably move the needle. Future research might explore federated or lifelong learning, where multiple robots share experiences to jointly update a central FM.

Techniques like memory retention (to avoid catastrophic forgetting) and knowledge distillation (to compress new knowledge into the model incrementally) will be critical. One idea is to maintain a growing knowledge base or case library outside the model-effectively an external memory-which the FM can query (via something like retrieval-augmented generation) to recall past lessons. Ensuring that an FM's priors evolve safely (monotonic improvement without forgetting how to avoid past hazards) is an important aspect of this problem.

Enhanced Reasoning for Complex Tasks: As tasks grow in complexity (e.g. multi-step missions like "organize my house completely"), current planning techniques may not scale reliably. One future direction is neuro-symbolic hybrids, where an FM (like an LLM) generates a high-level symbolic plan, which is then verified or filled in by more structured methods. For example, an LLM could propose a plan involving multiple agents or long-term

scheduling, and a classical planner or constraint solver checks consistency and feasibility before execution. Combining the creativity of learned models with the rigor of symbolic AI could tackle tasks out of reach for either alone (like reasoning about multi-agent collaboration or long-term resource constraints). Another idea is modular specialization: instead of one giant model for everything, train smaller foundation models specialized for different domains (kitchen tasks vs. outdoor tasks, for instance) and have a meta-reasoner choose which model's priors to invoke when. This could keep each model accurate within its scope while covering broad capabilities collectively. Integrating multiple FMs in a seamless way (ensuring they communicate properly and don't conflict) will be challenging but potentially very powerful.

Ethical and Social Alignment: We have focused on technical performance and safety, but aligning robot behavior with human values and social norms is equally important. As FMs often demonstrate, capabilities can race ahead of alignment. Techniques like reinforcement learning from human feedback (RLHF) have been used to align chatbots; analogous methods could align robots. For example, humans could watch a robot perform and give feedback not just on success but on how it achieved it – was it gentle enough, did it respect personal space, etc. The robot could then adjust its priors accordingly. Designing reward signals for such qualitative aspects is difficult. One promising approach is to leverage language: use an LLM to critique the robot's actions in plain language (e.g. "The way you handed the object was abrupt; be gentler next time") and translate that into an update for the policy. This crosses into multidisciplinary territory involving AI ethics, social science, and HRI (human-robot interaction) studies. As robots equipped with FMs become more common in social settings, ensuring they not only avoid harm but proactively follow social norms will be vital. Foundation models (especially LLMs) contain a wealth of priors about human values (gleaned from text), but making robotic behavior consistently align with those values remains an open challenge.

Benchmarking and Evaluation: To drive progress, we need better benchmarks to evaluate decision prior integration. Currently, robotics benchmarks focus on either end-to-end task success or narrow metrics. We lack standardized tasks that explicitly test the differences between world-model and policy-centric approaches. For example, a benchmark suite could include scenarios that require extreme foresight (to test reactive policies) and others that require split-second reflex (to test planners). We also need metrics for transparency and safety: how well can a system explain its decisions to a human? How gracefully does it fail under abnormal conditions? Creating these benchmarks will help quantify the benefits of hybrid designs and of improved prior integration. The community has started moving in this direction (e.g. The Real Robot Challenge tasks, or simulated household environments like BEHAVIOR), but those don't yet isolate the prior injection aspect. Designing targeted evaluations-perhaps where the same basic task is presented once in a way that favors planning and once that favors reactivity-would be very informative.

Computational Efficiency and Scalability: Many FMs are computationally heavy, posing deployment issues for robots, which often have limited on-board compute and battery. Research on model compression, distillation, and efficient inference for FMs in robotics is ongoing. One idea is a two-tiered compute strategy: use cloud resources for heavy planning or perception tasks via a world-model FM, but keep a light local policy for immediate control if connectivity drops. This introduces its own safety questions (is it okay for a "brain" to be in the cloud?), but practical implementations may demand it. Alternatively, specialized hardware or algorithmic optimizations (quantization, sparse models) will be needed to run large FMs on robots in real time. The broader point: to truly see FMs widely adopted in robotics, we must reduce their resource footprint or intelligently manage computation (perhaps by activating large models only when necessary).

In summary, treating foundation models as decision priors opens as many questions as it answers. It reframes classic problems-learning, planning, safety, HRI-in a new light, where we must consider how these giant pretrained structures can be harnessed, adapted, and constrained. This survey provides an initial map of this landscape. By distinguishing world-model vs policy-level influences and identifying where each breaks down, we hope to guide future research to the right level of abstraction. The boundaries we identified are not static; as models improve and hardware advances, what is infeasible today may become routine tomorrow. Thus, continuous re-evaluation of these trade-

offs is needed-essentially a moving frontier of what tasks are best solved by "reasoning" (models) vs. "reflex" (policies).

Logging & Accountability-immutable logs mapping FM prior injections to outcomes for incident review.

Compute/Latency Budgets-enforce inference latency and loop jitter bounds to protect control stability.

Human-in-the-Loop-takeover thresholds/interfaces; logging for post-hoc analysis.

Shadow/Override Policies-reflex fallback for time-critical hazards; clear arbitration rules.

Uncertainty & OOD Monitors-trigger re-planning or safe stop on low confidence or novelty.

Constraint Validation-external safety shield (CBFs [16]/shielding [19]) for candidate plans/actions.

Auditability-expose decision traces (plans/programs/imagined rollouts) for inspection.

5. Conclusion

Foundation models are poised to become cornerstones of robot intelligence. Realizing their full potential, however, requires a nuanced understanding of how they inject knowledge into the robot's decision-making stack. In this survey, we introduced the perspective of FMs as decision priors and systematically compared two paradigms-using FMs as world-model priors versus as policy priors-across verifiability, constraints, latency, horizon, and robustness criteria. Our analysis showed that neither paradigm is universally superior; each occupies a distinct region of the design space with its own advantages and failure modes. World-model priors excel at long-horizon planning, explicit constraint satisfaction, and providing interpretable "lookahead" predictions, but they incur computational overhead and rely on the learned model's fidelity to reality. Policy priors offer real-time responsiveness and direct utilization of broad pretraining (enabling zero-shot skills), yet they lack deliberative foresight and require external safeguards to ensure safety.

A key contribution of this work is highlighting the non-interchangeability boundaries between these approaches-the conditions under which one approach fails and the other succeeds. For example, we noted that hard physical constraints or the need for step-by-step verification favor world models [12], while split-second reactions or massive action-space problems favor policy priors. A one-size-fits-all strategy is untenable with current technology. Instead, next-generation robotic architectures should harness the complementary strengths of both. We discussed how hybrid designs-hierarchical planners, dynamic switches, blended controllers-can marry the foresight of model-based reasoning with the agility of learned policies (§3). Early systems like SayCan [1], LM-Nav [8], and PhysicalAgent already exemplify this philosophy, achieving results neither method could alone.

Throughout, we emphasized the importance of auditability and accountability in FM-driven robots. These systems will operate in human-centric environments, so transparency is not a luxury but a necessity. Treating FM influences as modular decision priors actually aids in this; it forces engineers to explicitly decide where and how an FM affects the robot's actions, making it clearer what to audit and who (or what) is responsible for each aspect of behavior. Incorporating mechanisms for introspection (the robot explaining its rationale), oversight (human or algorithmic monitors), and recourse (safe fallback strategies) will be critical as we integrate powerful FMs into real robots. FMs bring impressive capabilities, but they do not absolve us of ensuring alignment with human values and safety norms. If anything, their complexity makes this more challenging and vital.

Looking forward, we foresee that the most capable and trustworthy robots will be those that hybridize world-model and policy priors effectively. In the future, the line between "world-model" and "policy" may blur-a single architecture might contain multiple FMs playing different roles under the hood, coordinated seamlessly. Until then, the conceptual distinction remains useful for analysis and design. By viewing FM integration through the lens of decision priors and rigorously examining each approach's pros and cons, we have aimed to provide a clear map for researchers navigating this new terrain. The journey toward general-purpose, reliable robot autonomy is far from complete, but with foundation models as powerful new tools-and the right frameworks to apply them safely-the robotics community is poised to make decisive strides toward that goal.

Acknowledgments: The authors thank the developers of open-source libraries and

simulators that enabled many studies cited. We are also grateful to colleagues for insightful discussions, and to funding agencies supporting research at the intersection of large-scale AI and robotics.

References

[1] R. Bommasani et al., 2021. On the Opportunities and Risks of Foundation Models. Stanford CRFM Report (arXiv:2108.07258).

[2] NeurIPS 2022 Workshop on Foundation Models for Decision Making. 2022. [Online]. Available: neurips.cc/virtual/2022/workshop/49988

[3] R. Firoozi et al., 2024. Foundation Models in Robotics: Applications, Challenges, and the Future. arXiv preprint arXiv:2312.07843.

[4] A. Lykov et al., 2025. PhysicalAgent: Towards General Cognitive Robotics with Foundation World Models [12]. Submitted to IEEE Conference (arXiv:2509.13903).

[5] A. Sohn et al., 2024. Introducing RFM-1 [15]: Giving robots human-like reasoning capabilities. Covariant Blog, 11 Mar 2024. [Online]. Available: covariant.ai/insights/introducing-rfm-1-giving- robots-human-like-reasoning-capabilities/

[6] M. Ahn et al., 2022. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances (SayCan [1]). In Proc. of CoRL 2022, pp. 87–102.

[7] J. Liang et al., 2023. Code as Policies [6]: Language Model Programs for Embodied Control. In Proc. IEEE ICRA 2023, pp. 9493–9500.

[8] I. Singh et al., 2023. ProgPrompt [7]: Program Generation for Situated Robot Task Planning using LLMs. Autonomous Robots, 47(1): 999–1012.

[9] M. Shridhar et al., 2023. ChatGPT for Robotics: Design Principles and Model Abstractions. arXiv preprint arXiv:2303.17071.

[10] W. Ye et al., 2024. Reinforcement Learning with Foundation Priors: Let the Embodied Agent Efficiently Learn on Its Own. In Proc. CoRL 2024 (accepted, arXiv:2310.02635).

[11] J. Huang et al., 2024. Foundation models and intelligent decision-making: Progress, challenges, and perspectives. The Innovation, 5(6): 100726.

[12] P. Bowman-Davis, 2023. World Models [12] and the Sparks of Little Robotics. Andreessen Horowitz (a16z) Blog. [Online]. Available: a16z.com/world-models-and-the-sparks-of-little-robotics

[13] R. Bandaru, 2023. Foundation Models for Robotics: Vision-Language-Action (VLA). Personal Blog. [Online]. Available: rohitbandaru.github.io/blog/Foundation-Models-for-Robotics-VLA/

[14] A. Latyshev, G. Gorbov, A. I. Panov, 2025. Safe Planning and Policy Optimization via World Model Learning. arXiv preprint arXiv:2506.04828.

[15] N. Hansen, H. Su, X. Wang, 2024. TD-MPC2: Scalable, Robust World Models for Continuous Control. In Proc. ICLR 2024.

[16] D. Shah et al., 2023. LM-Nav [8]: Robotic Navigation with Large Pre-Trained Models of Language, Vision, and Action. In Proc. CoRL 2022, PMLR 205:334–345.

[17] D. Driess et al., 2023. PaLM-E [5]: An Embodied Multimodal Language Model. arXiv preprint arXiv: 2303.03378.

[18] Y. Chen et al., 2024. AutoTAMP [10]: Autoregressive Task and Motion Planning with LLMs as Translators and Checkers. In Proc. IEEE ICRA 2024 (to appear, arXiv:2306.06531).

