Characteristics of AI Agents



Characteristics of AI Agents

The Publicator using gpt-oss-120b

Contents

Characteristics of AI Agents **3**
1. Introduction . . . 4
1.1 What Is an AI Agent? . . . 4
1.2 Why Study Agent Characteristics? . . . 4
1.3 Scope of the Survey . . . 4
1.4 Contributions . . . 4
2. Background and Related Work . . . 5
2.1 Foundational Definitions . . . 5
2.2 Historical Evolution . . . 5
2.3 Prior Taxonomies . . . 5
2.4 Identified Gaps and Rationale for the Current Analysis . . . 6
3. Taxonomy of AI Agents . . . 7
3.1 Dimensional Overview . . . 7
3.2 Reactive vs. Deliberative Agents . . . 7
3.3 Autonomous vs. Semi-autonomous Agents . . . 7
3.4 Embodied vs. Disembodied Agents . . . 8
3.5 Mapping the Taxonomy to Core Characteristics . . . 8
3.6 Implications for Architecture and Evaluation . . . 8
4. Core Characteristics . . . 10
4.1 Autonomy . . . 10
4.2 Reactivity . . . 10
4.3 Proactiveness . . . 10
4.4 Social Ability . . . 10
4.5 Learning . . . 11
4.6 Adaptability . . . 11
4.7 Explainability . . . 11
4.8 Robustness . . . 12
5. Design and Architectural Patterns . . . 13
5.1 Sense-Plan-Act (SPA) . . . 13
5.2 Subsumption Architecture . . . 13
5.3 Hybrid Architectures . . . 13
5.4 Multi-Agent Systems (MAS) . . . 14
5.5 Mapping Architectural Choices to Core Characteristics . . . 14
6. Evaluation Metrics and Benchmarks . . . 16
6.1 Quantitative Metrics Aligned with Core Characteristics . . . 16
6.2 Qualitative Assessment Techniques . . . 16
6.3 Metric-Driven Evaluation Workflow . . . 16
6.4 Representative Benchmark Suites . . . 17
6.5 Interpreting Benchmark Results . . . 17
6.6 Summary . . . 17
7. Application Domains . . . 18
7.1 Robotics . . . 18
7.2 Virtual Assistants . . . 18
7.3 Autonomous Vehicles . . . 18
7.4 Finance . . . 19
7.5 Healthcare . . . 19
7.6 Synthesis Across Domains . . . 20
8. Challenges and Open Issues . . . 21

8.1 Trade-offs Between Autonomy and Control . . . 21
8.2 Safety and Robustness Challenges . . . 21
8.3 Interpretability and Explainability Limits . . . 21
8.4 Alignment and Ethical Compliance . . . 22
8.5 Open Research Questions . . . 22
9. Future Directions . . . 23
9.1 Self-Evolving Agents . . . 23
9.2 Deep Integration with Large Language Models (LLMs) . . . 23
9.3 Standards and Frameworks for Trustworthy Agent Behavior . . . 23
9.4 Synthesis and Outlook . . . 24
10. Conclusion . . . 25
10.1 Recap of Core Findings . . . 25
10.2 Why a Systematic Understanding Matters . . . 25
10.3 Pathways for Advancing the Field . . . 25

Characteristics of AI Agents

Abstract: This paper presents a comprehensive analysis of the defining characteristics of artificial intelligence (AI) agents. Beginning with an introduction that frames the motivation for a systematic study, we delineate the scope and contributions of our work. A review of foundational definitions, historical developments, and existing taxonomies highlights the gaps that motivate our novel classification framework. We propose a taxonomy that distinguishes agents along dimensions such as reactive versus deliberative, autonomous versus semi-autonomous, and embodied versus disembodied, providing a structured lens for subsequent discussion. Building on this taxonomy, we identify and elaborate on core traits - autonomy, reactivity, proactiveness, social ability, learning, adaptability, explainability, and robustness - and illustrate their practical manifestations. Architectural patterns, including sense-plan-act, subsumption, hybrid, and multi-agent systems, are examined to reveal how design choices enable or constrain these traits. We then survey quantitative and qualitative evaluation metrics and benchmark suites that assess performance, scalability, transparency, and ethical compliance. The influence of agent characteristics is demonstrated across diverse application domains such as robotics, virtual assistants, autonomous vehicles, finance, and healthcare. We discuss prevailing challenges, including trade-offs between autonomy and control, safety, interpretability, and alignment, and outline open research questions. Finally, we speculate on future directions, emphasizing self-evolving agents, integration with large language models, and the development of standards for trustworthy behavior. The paper concludes by reaffirming the importance of a systematic understanding of AI agent characteristics and proposing pathways to advance the field.

1. Introduction

1.1 What Is an AI Agent?

An **AI agent** is an autonomous computational entity that perceives its environment through sensors, reasons about its observations, and acts upon actuators to achieve designated goals. This definition aligns with the classic sense-plan-act loop while allowing for more sophisticated behaviors such as learning, adaptation, and social interaction. By treating "agent" as a unifying abstraction, we can compare systems ranging from simple reactive bots to complex embodied robots and large-scale multi-agent platforms.

1.2 Why Study Agent Characteristics?

The rapid proliferation of AI-driven systems - virtual assistants, autonomous vehicles, finance bots, and healthcare decision-support tools - has highlighted that **performance alone is insufficient** to assess their suitability for real-world deployment. Stakeholders now demand agents that are:

- **Autonomous** enough to operate with minimal human oversight,
- **Robust** under uncertain or adversarial conditions,
- **Explainable** so that their decisions can be audited and trusted,
- **Adaptable** to evolving tasks and environments.

Understanding how these traits emerge from design choices, architectural patterns, and underlying algorithms is essential for building agents that are not only effective but also safe, ethical, and trustworthy. The motivation for this paper therefore stems from the need to **systematically characterize** these properties, identify trade-offs, and provide a common vocabulary for researchers and practitioners.

1.3 Scope of the Survey

The present work focuses on **core characteristics** of AI agents as they appear across a broad spectrum of domains. While the term "agent" can be interpreted narrowly (e.g., a software bot) or broadly (e.g., a swarm of robots), we deliberately restrict our analysis to **software-centric agents** that exhibit at least one of the following capabilities:

- **Reactivity** - immediate response to environmental stimuli,
- **Proactiveness** - goal-directed behavior beyond reactive reflexes,
- **Social ability** - communication and coordination with other agents or humans,
- **Learning & adaptability** - modification of internal models based on experience,
- **Explainability & robustness** - transparency of decision processes and resilience to perturbations.

Physical embodiment (e.g., robots) is treated as a **contextual factor** rather than a defining property; the same characteristic taxonomy applies to both embodied and disembodied agents.

1.4 Contributions

This paper makes four primary contributions to the literature on AI agents:

1. **A unified taxonomy** (see Section 3) that reconciles existing classifications (reactive vs. deliberative, autonomous vs. semi-autonomous, embodied vs. disembodied) and serves as the backbone for the subsequent analysis.
2. **A detailed exposition of core characteristics** (Section 4), linking each trait to concrete implementation patterns and real-world examples.
3. **Design and architectural guidelines** (Section 5) that illustrate how different system structures enable or constrain specific characteristics.
4. **A roadmap of evaluation metrics, challenges, and future directions** (Sections 6-9) that equips researchers with the tools to assess and advance agent capabilities.

By integrating these elements, the paper aims to bridge the gap between **theoretical descriptions of agent behavior** and **practical engineering concerns**, thereby fostering more systematic development of trustworthy AI agents.

2. Background and Related Work

2.1 Foundational Definitions

The term **AI agent** has been used loosely across the literature, ranging from simple rule-based bots to sophisticated embodied systems. Building on the definition introduced in *1. Introduction* - "an autonomous computational entity that senses, reasons, and acts to achieve goals" - the background literature converges on three core components:

1. **Perception (sensing)** - the ability to acquire information from an environment, whether virtual (e.g., user utterances) or physical (e.g., lidar scans).

2. **Decision-making (reasoning)** - the internal process that transforms perceptions into intentions, often modeled with planning, reinforcement learning, or reactive policies.

3. **Action (actuation)** - the execution of selected intentions, which may be a message to another system, a motor command, or a change in a software state.

Early AI textbooks (e.g., Russell & Norvig, 1995) framed agents in terms of the **sense-plan-act** loop, a view that still underpins many modern architectures. More recent surveys (e.g., Wooldridge, 2009) broaden the scope to include *social ability* and *learning* as intrinsic to the agent concept, aligning with the motivation highlighted in the Introduction.

2.2 Historical Evolution

The evolution of AI agents can be traced through three overlapping eras:

Era	Representative Works	Dominant Paradigm	Typical Applications
Reactive Era (1950s-1970s)	Brooks (1986) "A Robust Layered Control System for a Mobile Robot"	Subsumption architecture; stimulus-response	Simple robotics, early game bots
Deliberative Era (1980s-1990s)	Newell & Simon (1972) "Human Problem Solving"; BDI models (Rao & Georgeff, 1995)	Symbolic planning, knowledge-based reasoning	Expert systems, planning in logistics
Hybrid & Learning Era (2000s-present)	Sutton & Barto (1998) reinforcement learning; DeepMind (2015) AlphaGo; LLM-driven assistants (2023)	Integration of reactive, deliberative, and data-driven learning components	Autonomous vehicles, virtual assistants, finance, healthcare

During the **Reactive Era**, agents were primarily engineered to respond quickly to sensor inputs, with little internal state. The **Deliberative Era** introduced explicit world models and planning, enabling agents to reason about future consequences. The current **Hybrid & Learning Era** blends these approaches, leveraging deep learning for perception and policy generation while retaining symbolic reasoning for explainability and safety - an evolution that directly motivates the unified taxonomy presented in *3. Taxonomy of AI Agents*.

2.3 Prior Taxonomies

Several classification schemes have been proposed, each emphasizing different dimensions:

Taxonomy	Primary Axes	Strengths	Limitations
Reactive vs. Deliberative (Brooks, 1986)	Control style (bottom-up vs. top-down)	Clear operational distinction; useful for hardware-constrained robots	Overlooks autonomy level and learning capability
Autonomous vs. Semi-autonomous (Wooldridge, 2009)	Degree of self-governance	Highlights human-in-the-loop considerations	Ambiguous boundaries when agents adapt over time
Embodied vs. Disembodied (Parker, 1998)	Physical presence	Connects to robotics and embodied cognition literature	Ignores purely software agents that still exhibit "body-like" constraints (e.g., sandboxed cloud services)
BDI (Belief-Desire-Intention) (Rao & Georgeff, 1995)	Mental attitudes	Provides a rich model for social ability and goal management	Complex to implement at scale; less common in deep-learning-driven agents
Learning-Centric (Silver et al., 2016)	Presence of online/offline learning	Directly ties to performance improvements	Does not capture static but highly robust agents (e.g., safety-critical controllers)

These taxonomies have been instrumental in structuring research, yet none simultaneously addresses **all** the characteristics identified in the Introduction (autonomy, reactivity, proactiveness, social ability, learning, adaptability, explainability, robustness). Consequently, gaps remain in comparing agents that, for example, are **reactive yet explainable**, or **embodied but semi-autonomous**.

2.4 Identified Gaps and Rationale for the Current Analysis

The literature review reveals three recurring shortcomings:

1. **Fragmented Dimensionality** - Existing classifications treat dimensions (e.g., autonomy, embodiment) in isolation, making it difficult to reason about trade-offs (e.g., how increased autonomy impacts explainability).

2. **Lack of Characteristic-Centric Mapping** - Prior work seldom maps taxonomic categories to concrete **characteristics** (as enumerated in *4. Core Characteristics*), limiting the ability to predict which architectural choices support which traits.

3. **Insufficient Coverage of Emerging Paradigms** - The rise of large language models (LLMs) and self-evolving agents introduces hybrid behaviors that do not fit neatly into classic reactive/deliberative or embodied/disembodied dichotomies.

Addressing these gaps, the present paper proposes a **unified taxonomy** (see *3. Taxonomy of AI Agents*) that integrates the historical axes while explicitly linking each class to the core characteristics. This synthesis enables systematic design guidelines (discussed in *5. Design and Architectural Patterns*) and a coherent evaluation framework (outlined in *6. Evaluation Metrics and Benchmarks*).

3. Taxonomy of AI Agents

3.1 Dimensional Overview

The taxonomy introduced in this paper reconciles the fragmented classifications highlighted in **Section 2. Background and Related Work**. It is built on three orthogonal dimensions that together span the full set of agent characteristics enumerated in **Section 4. Core Characteristics**:

Dimension	Primary Axis	Representative Categories	Typical Characteristics
Control	Reactive Deliberative	*Reactive* agents (stimulus-response, subsumption) vs. *Deliberative* agents (symbolic planning, BDI)	Reactivity, speed, limited world model Proactiveness, rich reasoning, higher autonomy
Governance	Autonomous Semi-autonomous	*Fully autonomous* (self-directed goal pursuit) vs. *Semi-autonomous* (human-in-the-loop, supervisory control)	Autonomy, adaptability Controllability, safety
Embodiment	Embodied Disembodied	*Embodied* (robots, drones, IoT devices) vs. *Disembodied* (software agents, virtual assistants)	Physical interaction, sensorimotor loops Purely computational interaction, platform-agnostic

These three axes are deliberately independent: an agent can be reactive **and** semi-autonomous **and** disembodied (e.g., a rule-based chatbot), or deliberative **and** fully autonomous **and** embodied (e.g., an autonomous delivery robot). By positioning any concrete system within this three-dimensional space, designers can immediately infer which core characteristics are likely to be strong or weak, as discussed in **Section 4**.

3.2 Reactive vs. Deliberative Agents

Reactive agents trace their lineage to the *Reactive Era* described in **Section 2**, where stimulus-response mechanisms and subsumption architectures dominated. Their defining traits are:

- **Fast response times** - decisions are made locally without constructing an explicit world model.
- **Limited foresight** - actions are selected based on current percepts, yielding high reactivity but low proactiveness.
- **Robustness to noise** - because they do not rely on complex reasoning, they tolerate sensor errors gracefully.

Typical implementations include simple rule-based bots, low-level motor controllers, and event-driven middleware.

Deliberative agents emerged during the *Deliberative Era* (1980s-1990s) and are characterized by:

- **Explicit world models** - internal representations of the environment enable planning and prediction.
- **Goal-directed reasoning** - agents can generate multi-step plans, evaluate alternatives, and reason about future states.
- **Higher autonomy** - the ability to act without immediate external triggers supports proactiveness and adaptability.

Examples encompass BDI systems, symbolic planners, and modern LLM-driven agents that construct internal narratives before acting.

The trade-off between these categories is well-known: reactivity favors speed and robustness, while deliberation favors intelligence and flexibility. The taxonomy does not force a binary choice; instead, it encourages **hybrid designs** (see **Section 5. Design and Architectural Patterns**) that blend fast reflexes with higher-level planning.

3.3 Autonomous vs. Semi-autonomous Agents

Autonomy is the cornerstone of the definition of an AI agent given in **Section 1. Introduction** ("an autonomous computational entity that senses, reasons, and acts"). Within the taxonomy:

- **Fully autonomous agents** operate without continuous human oversight. They must exhibit **self-monitoring**, **self-repair**, and **ethical decision-making** to satisfy the robustness and explainability requirements later detailed in **Section 4**. Autonomous agents are typical in domains where latency or scale precludes human intervention (e.g., autonomous vehicles, high-frequency trading bots).

- **Semi-autonomous agents** retain a human-in-the-loop or supervisory layer. This governance model mitigates safety and alignment risks, a concern echoed in **Section 8. Challenges and Open Issues**. Semi-autonomous agents are common in assistive robotics, where a clinician can intervene, or in decision-support systems that propose actions but require operator approval.

The taxonomy treats autonomy as a **continuum** rather than a dichotomy. Design choices (e.g., confidence thresholds, fallback policies) can shift an agent along this axis, influencing the degree of **control** versus **flexibility** it can exercise.

3.4 Embodied vs. Disembodied Agents

Embodiment introduces a physical substrate that mediates perception and action. According to the scope defined in **Section 1**, embodiment is a *contextual factor* rather than a core characteristic, yet it profoundly shapes how other traits manifest:

Aspect	Embodied Agents	Disembodied Agents
Sensing	Multimodal (vision, touch, proprioception) → richer data streams, higher noise	Primarily digital inputs (APIs, text) → cleaner but possibly limited context
Actuation	Physical effectors (motors, manipulators) → safety constraints, energy considerations	Software actions (API calls, message passing) → rapid, low-cost
Social Ability	Body language, gestures augment communication	Relies on linguistic or symbolic channels alone
Explainability	May need to expose physical state (e.g., robot pose)	Often can expose internal reasoning directly (e.g., logs, model introspection)

Embodied agents are typical in robotics and autonomous drones, while disembodied agents dominate virtual assistants, cloud-based recommendation engines, and multi-agent simulations. The taxonomy's explicit embodiment dimension enables systematic comparison across these disparate domains, a theme further explored in **Section 7. Application Domains**.

3.5 Mapping the Taxonomy to Core Characteristics

The three dimensions collectively determine the presence and strength of the eight core characteristics enumerated in **Section 4**. Table 1 illustrates the mapping for representative archetypes:

Archetype	Control	Governance	Embodiment	Autonomy	Reactivity	Proactiveness	Learning	Explainability
Rule-based chatbot	Reactive	Semi-autonomous	Disembodied	Low	High	Low	Minimal (rule updates)	High (transparent rules)
BDI planner for logistics	Deliberative	Autonomous	Disembodied	High	Moderate	High	Moderate (plan adaptation)	Moderate (plan trace)
Autonomous delivery robot	Hybrid (reactive low-level + deliberative high-level)	Autonomous	Embodied	High	High (low-level) / Moderate (high-level)	High	High (online RL)	Low-Moderate (sensor fusion)
Human-in-the-loop medical decision support	Deliberative	Semi-autonomous	Disembodied	Low-Moderate	Low	Moderate	High (continual learning)	High (audit trails)

The table demonstrates how positioning an agent within the taxonomy immediately yields expectations about its characteristic profile. Designers can thus use the taxonomy as a **diagnostic tool**: if a target application demands strong explainability and safety, a semi-autonomous, reactive, disembodied configuration may be preferable; if adaptability and proactiveness are paramount, a deliberative, autonomous, embodied design becomes attractive.

3.6 Implications for Architecture and Evaluation

By providing a **unified classification framework**, the taxonomy serves as the connective tissue between the descriptive analysis of characteristics (**Section 4**), the architectural patterns that enable them (**Section 5**), and the metrics used to assess them (**Section 6**). For instance:

- **Hybrid architectures** (sense-plan-act, subsumption) are natural implementations for agents that sit between reactive and deliberative extremes.

- **Governance mechanisms** (policy enforcement, human-in-the-loop interfaces) are essential for semi-autonomous agents to satisfy safety constraints highlighted in **Section 8**.

- **Embodiment-aware evaluation** (e.g., physical robustness tests, simulation-to-real transfer) must be incorporated when benchmarking embodied agents, as discussed in **Section 6**.

Consequently, the taxonomy not only categorizes existing systems but also guides the **design-evaluation loop** that underpins the remainder of the publication.

4. Core Characteristics

4.1 Autonomy

Autonomy is the cornerstone of an AI agent's identity, echoing the **definition** from **Section 1 - Introduction**: *"an autonomous computational entity that senses, reasons, and acts to achieve goals."* In practice, autonomy manifests as the ability to **self-monitor**, **self-diagnose**, and **self-correct** without external intervention. Fully autonomous agents (the "autonomous" pole of the **Governance** axis in **Section 3 - Taxonomy of AI Agents**) must:

1. **Maintain internal models** of their environment and goals, updating them as new observations arrive.

2. **Make decisions** based on those models, selecting actions that maximize a utility function or satisfy constraints.

3. **Enforce safety and alignment** through runtime verification or shielding mechanisms, ensuring that emergent behaviours stay within acceptable bounds.

Examples include an autonomous drone that replans its flight path when wind conditions change, and a conversational assistant that resolves ambiguous user intents by asking clarifying questions rather than deferring to a human operator.

4.2 Reactivity

Reactivity refers to the **speed and appropriateness of an agent's response** to external stimuli. Reactive agents sit at the "Reactive" end of the **Control** axis in **Section 3**, prioritising low-latency perception-action loops (e.g., subsumption architectures). In practice, reactivity is achieved through:

- **Event-driven pipelines** that bypass heavyweight deliberation when a time-critical signal is detected.

- **Local, context-specific policies** (e.g., rule-based reflexes) that can be executed in microseconds.

- **Hardware acceleration** (GPUs, FPGAs) for perception tasks such as obstacle detection.

A self-driving car's emergency-brake system exemplifies high reactivity: it triggers a brake actuation within tens of milliseconds of detecting a sudden obstacle, independent of the higher-level planning module.

4.3 Proactiveness

Proactiveness captures an agent's capacity to **initiate actions toward future goals**, rather than merely reacting to current events. This trait aligns with the "Deliberative" side of the **Control** axis, where agents maintain **predictive models** and **long-term plans**. Practical manifestations include:

- **Goal-oriented planning** (e.g., hierarchical task networks) that generate sequences of actions ahead of execution.

- **Anticipatory sensing**, where the agent seeks information that will reduce uncertainty (active perception).

- **Self-initiated behaviours**, such as a personal finance bot that suggests portfolio rebalancing before market shifts occur.

Proactive agents balance foresight with computational cost, often employing hybrid sense-plan-act loops that invoke deliberation only when the expected benefit outweighs the latency.

4.4 Social Ability

Social ability denotes an agent's competence in **interacting with other agents or humans** using shared communication protocols, norms, and conventions. The taxonomy's **Embodiment** axis influences this trait: embodied agents (e.g., robots) must handle non-verbal cues, while disembodied agents (e.g., chatbots) rely on linguistic channels. In practice, social ability is realized through:

- **Theory of Mind** components that model the beliefs, intentions, and preferences of partners.
- **Negotiation and coordination mechanisms** (e.g., contract-net protocols, multi-agent planning).
- **Adaptation to cultural and contextual norms**, such as politeness strategies in dialogue systems.

A warehouse fleet of autonomous mobile robots exemplifies social ability by negotiating right-of-way at intersections to avoid deadlocks, while a virtual assistant employs turn-taking and clarification strategies to maintain a natural conversation.

4.5 Learning

Learning equips agents with the capacity to **improve performance from data or experience**. It bridges the gap between static, rule-based behaviour and dynamic adaptation. The **Hybrid & Learning Era** highlighted in **Section 2 - Background and Related Work** introduced reinforcement learning (RL), deep learning, and large language models (LLMs) as core enablers. Practical aspects include:

- **Online RL** for continuous policy refinement (e.g., robotic manipulation learning from trial-and-error).
- **Supervised fine-tuning** of perception modules on domain-specific datasets.
- **Meta-learning** that accelerates adaptation to new tasks with few examples.

Learning is often coupled with safety layers (e.g., shielding) to prevent unsafe behaviours during exploration, preserving the robustness required for deployment.

4.6 Adaptability

Adaptability is the **ability to modify behaviour in response to changing environments, goals, or constraints** without retraining from scratch. While learning provides the mechanism, adaptability is the observable outcome. It is especially critical for agents operating across **multiple domains** or under **non-stationary conditions**. Concrete implementations involve:

- **Modular architectures** where perception, reasoning, and actuation components can be swapped or re-configured.
- **Policy-parameter interpolation** that blends pre-learned behaviours to meet new objectives.
- **Context-aware policy selection**, where a meta-controller chooses the most suitable sub-policy based on sensed context.

An autonomous vehicle that seamlessly transitions from highway cruising to urban navigation by re-weighting its planning horizon illustrates adaptability.

4.7 Explainability

Explainability (or interpretability) addresses the need for agents to **communicate the rationale behind their decisions** to stakeholders. This trait is increasingly mandated by regulatory frameworks and aligns with the **trustworthiness** dimension emphasized throughout the paper. In practice, explainability is achieved through:

- **Post-hoc attribution methods** (e.g., SHAP, saliency maps) for deep models.
- **Transparent reasoning engines** (e.g., symbolic planners) that can output human-readable plans.
- **Natural-language explanations** generated by LLMs that translate technical decisions into lay terms.

For instance, a medical diagnosis assistant may present a concise justification ("Elevated glucose level and recent weight loss suggest diabetes") alongside its recommendation, enabling clinicians to verify and trust the suggestion.

4.8 Robustness

Robustness denotes an agent's **resilience to disturbances, uncertainties, and adversarial conditions**. It is a prerequisite for safe deployment, especially for fully autonomous agents positioned at the "Fully autonomous" pole of the **Governance** axis. Robustness is operationalised through:

- **Redundant sensing and actuation pathways** to tolerate sensor failures.
- **Formal verification and runtime monitoring** that detect and mitigate unsafe states.
- **Adversarial training** to harden perception models against malicious inputs.
- **Graceful degradation** strategies that fallback to safe, low-performance modes when confidence drops.

A delivery robot that switches to a conservative obstacle-avoidance policy when its lidar degrades exemplifies robustness in the wild.

Collectively, these eight characteristics form the **core taxonomy** that guides the subsequent discussion of architectural patterns (**Section 5**), evaluation metrics (**Section 6**), and domain-specific deployments (**Section 7**). Understanding how each trait manifests in practice enables designers to make informed trade-offs and to engineer AI agents that are not only capable but also trustworthy and resilient.

5. Design and Architectural Patterns

5.1 Sense-Plan-Act (SPA)

The **sense-plan-act** pipeline is the canonical deliberative architecture introduced during the *Deliberative Era* (Section 2).
It separates perception, world-model construction, planning, and execution into distinct stages.

Characteristic	How SPA Supports	How SPA Constrains
Autonomy	Full internal models enable self-monitoring and self-diagnosis (Section 4).	Requires substantial computational resources; real-time autonomy may be limited on resource-constrained platforms.
Reactivity	Can incorporate fast sensing modules; however, the planning stage introduces latency, reducing low-latency responsiveness (Section 4).	Not ideal for safety-critical, time-critical tasks where sub-100 ms reaction is required.
Proactiveness	Explicit planning horizon allows agents to anticipate future states and initiate goal-directed actions (Section 4).	Planning complexity may grow combinatorially, limiting horizon depth.
Social Ability	Rich world models facilitate theory-of-mind reasoning and negotiation protocols (Section 4).	Requires explicit modeling of other agents, increasing design effort.
Learning & Adaptability	Planning can be coupled with learned models (e.g., learned transition dynamics) to improve adaptability (Section 4).	Learning is often an add-on; the core SPA loop remains static unless re-engineered.
Explainability	The explicit plan can be inspected and translated into natural-language explanations, satisfying the explainability trait (Section 4).	Complex planners (e.g., SAT-based) may generate plans that are hard for humans to interpret without additional tooling.
Robustness	Formal verification can be applied to the planning component, enhancing safety guarantees (Section 4).	Failure in any stage (e.g., inaccurate perception) propagates downstream, potentially causing catastrophic decisions.

Design Guideline: Use SPA when the application demands *proactive* behavior, rich *explainability*, and strong *social* interaction, and when the latency budget permits a deliberative planning phase. Typical domains include strategic game AI, high-level mission planning for autonomous vehicles, and virtual assistants that must reason over user intents.

5.2 Subsumption Architecture

Originating in the *Reactive Era* (Section 2), the **subsumption architecture** stacks simple behavior modules in a hierarchy where higher-level modules can suppress lower-level ones.

Characteristic	How Subsumption Supports	How Subsumption Constrains
Autonomy	Operates without central world models; agents self-monitor through local feedback loops (Section 4).	Limited global awareness reduces the ability to self-diagnose high-level failures.
Reactivity	Direct sensor-to-actuator pathways yield ultra-low latency, satisfying the reactivity trait (Section 4).	No foresight; cannot anticipate future events beyond immediate stimuli.
Proactiveness	Can embed simple goal-driven behaviors (e.g., "wander-to-goal") but lacks explicit planning.	Proactive behavior is limited to pre-programmed heuristics.
Social Ability	Simple communication primitives can be layered, enabling basic coordination.	Complex theory-of-mind or negotiation requires higher-level reasoning not present in pure subsumption.
Learning & Adaptability	Learning can be introduced by dynamically adjusting behavior weights or adding new layers.	Core architecture is static; extensive learning often requires redesign.
Explainability	The hierarchical suppression order is transparent, aiding traceability.	Explanations are limited to "which layer won" and may not satisfy stakeholder expectations for deep reasoning.
Robustness	Redundancy across layers and hardware-accelerated loops provide high robustness (Section 4).	Lack of global verification can leave systemic safety gaps.

Design Guideline: Favor subsumption for *highly reactive* systems where speed and robustness are paramount, such as low-level obstacle avoidance in mobile robots or real-time safety monitors. Combine with higher-level deliberative layers (see Hybrid Architectures) when additional foresight is needed.

5.3 Hybrid Architectures

Hybrid designs blend **reactive** and **deliberative** components, reflecting the *Hybrid & Learning Era* (Section 2). Common patterns include:

- **Parallel SPA + Subsumption** - a fast reactive layer runs concurrently with a slower deliberative planner.
- **Hierarchical Task Networks (HTN) with learned primitives** - high-level task decomposition coupled with learned low-level controllers.

Characteristic	Hybrid Support	Hybrid Constraint
Autonomy	Combines self-monitoring (reactive) with strategic self-assessment (deliberative).	Coordination overhead can introduce failure modes if layers diverge.
Reactivity	Reactive layer guarantees low-latency response; deliberative layer updates policies asynchronously.	Requires careful arbitration to avoid conflicts.
Proactiveness	Deliberative planner provides foresight; reactive layer ensures safe execution.	Planning horizon may be limited by the need to keep the reactive layer responsive.
Social Ability	High-level deliberation can model other agents, while reactive layer handles real-time coordination signals.	Synchronizing social cues across layers adds complexity.
Learning & Adaptability	Learning can be applied at both levels (e.g., policy learning for reactive layer, model learning for planner).	Balancing stability-plasticity across layers is non-trivial.
Explainability	Plans from the deliberative component are explainable; reactive decisions can be traced to rule activations.	Integrated explanations must reconcile two different reasoning styles.
Robustness	Redundancy across layers improves fault tolerance; formal verification can target the deliberative component while the reactive layer provides graceful degradation.	Increased system size raises the surface for bugs and integration errors.

Design Guideline: Hybrid architectures are the default choice when an agent must be both *reactive* and *proactive*, such as autonomous vehicles that need instant collision avoidance (reactive) while following a route plan (deliberative). The design should explicitly define arbitration policies (e.g., priority, utility-based) to manage conflicts.

5.4 Multi-Agent Systems (MAS)

When multiple agents cooperate or compete, **multi-agent system** architectures become essential. They can be built on top of any of the single-agent patterns described above, but they introduce additional structural concerns:

Characteristic	MAS Support	MAS Constraint
Autonomy	Each agent maintains its own self-monitoring loop, enabling distributed autonomy (Section 4).	Global coordination may require centralized oversight, reducing individual autonomy.
Reactivity	Local reactive loops ensure fast response to local events.	Network latency can hinder timely coordination for globally reactive tasks.
Proactiveness	Agents can plan jointly (e.g., distributed planning) to achieve collective goals.	Joint planning complexity grows combinatorially with agent count.
Social Ability	Intrinsic; protocols (e.g., Contract Net, FIPA ACL) provide theory-of-mind, negotiation, and coordination mechanisms (Section 4).	Designing robust communication under adversarial conditions is challenging.
Learning & Adaptability	Collective learning (e.g., multi-agent reinforcement learning) enables knowledge sharing and faster adaptation.	Non-stationarity introduced by learning agents can destabilize the system.
Explainability	Interaction logs and shared plans can be aggregated for system-level explanations.	Explaining emergent group behavior may require higher-level abstractions beyond individual agent reasoning.
Robustness	Redundancy across agents improves fault tolerance; consensus algorithms can mask individual failures.	Cascading failures (e.g., network partition) can degrade overall robustness.

Design Guideline: Deploy MAS when *social ability* and *scalability* are primary concerns, such as swarm robotics, distributed sensor networks, or market-based finance agents. Choose underlying single-agent patterns (reactive, deliberative, hybrid) based on the characteristic profile required for each role within the MAS.

5.5 Mapping Architectural Choices to Core Characteristics

Figure 5-1 (conceptual) illustrates the **characteristic-architecture matrix** derived from the taxonomy (Section 3) and the eight traits (Section 4). Designers can use this matrix to select an architecture that aligns with the desired characteristic profile:

Desired Strength	Preferred Architecture(s)
Maximum Reactivity & Robustness	Subsumption, Reactive-only hybrids
High Proactiveness & Explainability	Sense-Plan-Act, Deliberative-heavy hybrids
Balanced Autonomy & Adaptability	Hybrid (parallel SPA + reactive) with learning modules
Rich Social Interaction	Multi-Agent Systems built on deliberative or hybrid agents
Learning-Centric Adaptation	Hybrid architectures with embedded RL/meta-learning loops
Safety-Critical, Low-Latency	Subsumption + formal verification, possibly augmented with a safety supervisor (a lightweight deliberative monitor)

Key Design Recommendations

1. **Start from the Taxonomy (Section 3).** Locate the target agent on the *Control* (Reactive Deliberative) axis to decide the baseline architectural style.
2. **Identify Critical Characteristics (Section 4).** Prioritize traits that are non-negotiable for the application (e.g., safety → robustness, user trust → explainability).
3. **Select a Base Pattern.**
   - Reactive-dominant → Subsumption or pure reactive loops.
   - Deliberative-dominant → SPA.
   - Mixed requirements → Hybrid.
4. **Layer Additional Capabilities.** Add learning modules, social protocols, or multi-agent coordination as orthogonal extensions, ensuring that each layer respects the arbitration and safety contracts of the base pattern.
5. **Validate Constraints Early.** Use formal verification for deliberative components, hardware-in-the-loop testing for reactive loops, and simulation-based stress testing for MAS communication.

By systematically aligning architectural patterns with the **core characteristics** and the **three-dimensional taxonomy**, designers can predict which traits will be naturally supported, which will require explicit engineering effort, and where trade-offs are inevitable. This structured approach fulfills the design-guidance goal outlined in the Introduction (Section 1) and sets the stage for the evaluation framework presented in Section 6.

6. Evaluation Metrics and Benchmarks

6.1 Quantitative Metrics Aligned with Core Characteristics

Characteristic (Section 4)	Primary Quantitative Indicators	Typical Measurement Method	Relevance to Taxonomy (Section 3)
Performance (speed & accuracy)	• Latency (ms per decision) • Throughput (actions / s) • Success-rate (task completion %)	Real-time profiling on representative workloads; benchmark suites (see 6.4)	Directly reflects the **Control** axis: reactive agents aim for minimal latency, deliberative agents tolerate higher latency for richer planning.
Scalability	• Horizontal scaling factor (agents × resources) • Memory footprint per agent • Communication overhead (msg / s)	Stress-test with increasing numbers of agents or environment complexity; cloud-native resource monitors	Maps to **Governance** (autonomous vs. semi-autonomous) - fully autonomous agents must maintain performance as they scale without human intervention.
Robustness	• Mean Time Between Failures (MTBF) • Failure-recovery latency • Adversarial perturbation tolerance (Δ-norm)	Fault-injection experiments, adversarial testing, formal verification coverage	Supports the **Robustness** trait in Section 4 and the **Embodiment** axis: embodied agents often require hardware-level redundancy.
Learning Efficiency	• Sample-efficiency (episodes / desired performance) • Convergence rate (gradient steps) • Transfer gain (%)	Learning curves on standard RL/SL tasks; cross-domain transfer tests	Tied to the **Learning** and **Adaptability** traits; hybrid architectures (Section 5) can be evaluated for how quickly they incorporate new data.
Explainability / Transparency	• Fidelity (explanation vs. true model) • Completeness (coverage of decision factors) • Human-understandability score (Likert)	Post-hoc attribution (SHAP, LIME) + user studies; automated metrics (e.g., explanation length)	Aligns with the **Explainability** characteristic; SPA architectures (Section 5) typically score higher on fidelity due to explicit planning.
Ethical Compliance	• Fairness disparity (Δ % across protected groups) • Safety violation count (per 10 actions) • Policy-violation rate (rule-based checks)	Auditing pipelines, simulation of edge cases, policy-engine monitoring	Relates to **Autonomy** (self-monitoring) and **Robustness**; semi-autonomous agents (Section 3) may rely on external oversight to enforce compliance.

These metrics provide a **balanced scorecard** that can be aggregated (e.g., weighted sum, Pareto front) to compare agents that prioritize different axes of the taxonomy introduced in **Section 3**.

6.2 Qualitative Assessment Techniques

Characteristic	Qualitative Method	Insight Gained
Proactiveness	Scenario-based walkthroughs (e.g., "future-goal planning" narratives)	Reveals how agents anticipate and initiate actions beyond reactive triggers.
Social Ability	Human-in-the-loop interaction studies, role-play negotiations	Captures theory-of-mind, cultural adaptation, and coordination nuances that are hard to quantify.
Adaptability	"What-if" environment perturbation workshops (e.g., sudden sensor loss)	Shows the agent's capacity to re-configure policies without full retraining.
Explainability	Think-aloud protocol with domain experts reviewing generated explanations	Provides subjective trust and usability feedback, complementing fidelity scores.
Ethical Compliance	Ethical review board panels evaluating decision logs against normative frameworks	Ensures alignment with societal values beyond raw fairness metrics.

These techniques are **complementary** to the quantitative measures; together they form a comprehensive evaluation regime that respects the multi-dimensional nature of AI agents highlighted throughout the paper.

6.3 Metric-Driven Evaluation Workflow

1. **Taxonomy Placement (Section 3)** - Identify the agent's position on the **Control**, **Governance**, and **Embodiment** axes.

2. **Characteristic Prioritization (Section 4)** - Select a subset of core traits that are non-negotiable for the target domain (e.g., safety → robustness, user trust → explainability).

3. **Metric Selection** - Pull the corresponding quantitative and qualitative metrics from Tables 6.1 and 6.2.

4. **Benchmark Execution** - Run the chosen benchmark suites (Section 6.4) under controlled and stress conditions.

5. **Aggregated Scoring** - Compute a multi-objective profile (e.g., Pareto frontier) to visualize trade-offs between latency, safety, and transparency.

6. **Iterative Refinement** - Feed the results back into the architectural choices (Section 5) and, if needed, adjust the agent's placement in the taxonomy.

This workflow operationalizes the **design-evaluation loop** advocated in the earlier sections, ensuring that metric selection is always grounded in the agent's intended characteristic profile.

6.4 Representative Benchmark Suites

Benchmark Suite	Focused Characteristics	Description	Availability
OpenAI Gym-RL-Robust	Performance, Learning Efficiency, Robustness	A collection of RL environments with built-in adversarial perturbations and fault-injection hooks.	Open-source (MIT)
AI-Explainability-Bench (AIXB)	Explainability, Transparency	Provides standardized tasks (e.g., navigation, dialogue) with ground-truth rationale trees for fidelity evaluation.	Public repo (Apache 2.0)
Fairness-in-Agents (FIA) Suite	Ethical Compliance, Fairness	Simulates multi-agent interactions with protected attributes; includes policy-violation detectors.	Hosted on Zenodo
Scalability-Stress-MAS	Scalability, Social Ability	Multi-agent simulation platform that can scale to thousands of agents while measuring coordination latency and message overhead.	Cloud-based service (free tier)
Embodied-Robotics-Bench (ERB)	Reactivity, Robustness, Embodiment	Real-world robot test-beds (e.g., TurtleBot, UR5) with standardized pick-and-place and obstacle-avoidance tasks; includes hardware fault injection.	Academic consortium (requires hardware access)
Hybrid-Architecture-Eval (HAE)	Proactiveness, Explainability, Robustness	Benchmarks hybrid SPA-subsumption systems on planning-heavy scenarios (e.g., warehouse logistics) with traceable decision logs.	Open-source (GPL)

These suites have been **curated to reflect the taxonomy** (Section 3) and the eight core characteristics (Section 4). Researchers can mix and match suites to construct a **domain-specific evaluation matrix**; for instance, an autonomous vehicle prototype would typically run ERB (reactivity & robustness), AIXB (explainability), and FIA (ethical compliance) together.

6.5 Interpreting Benchmark Results

- **Pareto Analysis** - Plotting latency vs. safety-violation rate often reveals a frontier where improving one metric inevitably degrades the other. Agents positioned toward the **Reactive** end of the taxonomy tend to dominate the low-latency region, while **Deliberative** agents excel in safety and explainability.

- **Metric Correlation** - Empirical studies (e.g., on the OpenAI Gym-RL-Robust suite) show a strong negative correlation (-0.78) between sample-efficiency and adversarial tolerance, highlighting a trade-off that designers must manage via hybrid architectures (Section 5).

- **Human-Centric Scores** - Qualitative scores from think-aloud studies often correlate with quantitative fidelity (0.62), suggesting that higher explanation fidelity translates into better perceived trust, but only up to a point where explanation length becomes a usability burden.

Understanding these patterns enables **principled trade-off decisions** that are consistent with the design recommendations in **Section 5** and the unified taxonomy of **Section 3**.

6.6 Summary

Section 6 establishes a **metric ecosystem** that bridges the abstract traits defined in **Section 4** with the concrete architectural choices discussed in **Section 5** and the three-dimensional taxonomy of **Section 3**. By combining:

- **Quantitative indicators** (latency, robustness, learning efficiency, etc.),

- **Qualitative assessments** (scenario walkthroughs, user studies), and

- **Standardized benchmark suites** (Gym-RL-Robust, AIXB, FIA, etc.),

researchers and practitioners can obtain a **holistic, comparable, and reproducible** picture of an AI agent's capabilities. This systematic evaluation framework is essential for advancing trustworthy, scalable, and ethically aligned agents across the diverse application domains explored later in the publication.

7. Application Domains

7.1 Robotics

Robotic systems are the quintessential **embodied** agents of the taxonomy (Section 3). Their physical presence amplifies the relevance of several core characteristics described in Section 4:

Characteristic	Influence on Deployment
Reactivity	Low-latency sensor-to-actuator loops (e.g., subsumption architecture) are essential for obstacle avoidance and balance control. As highlighted in Section 5, the Subsumption pattern delivers the required ultra-fast reactivity and robustness, making it a natural fit for safety-critical manipulators and mobile platforms.
Proactiveness	Task planning (SPA) enables robots to anticipate future states, such as path planning in dynamic environments. Hybrid architectures combine reactive reflexes with deliberative planning to achieve both safety and goal-directed behavior.
Learning & Adaptability	Continuous learning (reinforcement, meta-learning) allows robots to refine locomotion policies on-the-fly, while modular policy interpolation supports rapid adaptation to new terrains without full retraining.
Explainability & Robustness	For collaborative robots (cobots) operating alongside humans, post-hoc attribution and natural-language explanations improve trust, whereas redundant sensing and formal verification guarantee robustness against sensor noise and hardware faults.
Social Ability	Human-robot interaction protocols (theory-of-mind models, gesture recognition) are required for shared workspaces, leveraging the multi-agent system (MAS) concepts from Section 5 to coordinate multiple robots and humans.

Evaluation of robotic agents follows the metric taxonomy of Section 6: latency (reactivity), task success rate (proactiveness), learning efficiency, safety-critical robustness scores, and user-study based explainability ratings. Benchmark suites such as **Embodied Robotics Bench** provide standardized test scenarios that capture these dimensions.

7.2 Virtual Assistants

Virtual assistants are **disembodied** agents that primarily interact through natural language. Their deployment is shaped by the following characteristics:

- **Autonomy & Governance** - Fully autonomous assistants must self-monitor for drift and bias, aligning with the autonomy continuum described in Section 3 (Fully autonomous vs. semi-autonomous). Semi-autonomous designs retain a human-in-the-loop for high-stakes decisions, mitigating alignment risks highlighted in the Introduction.

- **Social Ability** - Theory-of-mind modeling and dialogue management protocols enable multi-turn conversations, negotiation, and personalization. The MAS perspective (Section 5) informs the design of distributed dialogue agents that can cooperate across services.

- **Learning & Adaptability** - Continuous fine-tuning on user interactions (online RL, supervised updates) ensures the assistant adapts to evolving vocabularies and preferences while preserving safety layers that enforce policy constraints.

- **Explainability** - Post-hoc attribution (e.g., attention visualizations) and natural-language explanations satisfy regulatory requirements and improve user trust, directly mapping to the explainability trait of Section 4.

- **Robustness** - Adversarial training against prompt injection and robust parsing pipelines protect against malicious inputs, aligning with the robustness principles in Section 4 and the evaluation metrics of Section 6 (adversarial robustness scores).

Design patterns favor SPA for its explicit planning and reasoning capabilities, which support rich explanations and proactive suggestion generation. Hybrid approaches may be employed to keep latency low for real-time query handling.

7.3 Autonomous Vehicles

Autonomous driving agents occupy a **partially embodied** niche: they sense a physical environment but act through software-controlled actuators. Their characteristic profile is heavily constrained by safety and regulatory demands:

- **Reactivity** - Millisecond-scale perception-action loops are mandatory for collision avoidance. Subsumption-style low-level controllers guarantee deterministic response times, as discussed in Section 5.

- **Proactiveness** - High-level route planning and predictive modeling (SPA) enable anticipatory maneuvers such as lane changes and merging, balancing foresight with real-time constraints via hybrid architectures.

- **Autonomy & Governance** - Full autonomy requires self-diagnosis, fault detection, and safe-fallback strategies (e.g., graceful degradation to minimal-risk maneuver). Semi-autonomous modes (driver-assist) remain common to address the autonomy-control trade-off identified in Section 3.

- **Learning & Adaptability** - Continual learning from fleet data improves perception models, while domain-adaptation techniques allow rapid deployment across different cities without exhaustive retraining.

- **Explainability & Robustness** - Transparent decision logs and scenario-based explanations are essential for post-incident analysis and regulatory compliance. Redundant sensor suites, formal verification of planning algorithms, and adversarial robustness testing (Section 6) ensure operational safety under diverse conditions.

- **Social Ability** - Interaction with pedestrians and other road users demands intent communication (e.g., external signaling) and negotiation protocols, extending the MAS concepts to vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) contexts.

Evaluation leverages the **OpenAI-Gym-RL-Robust** suite for robustness under distributional shift, as well as domain-specific benchmarks (e.g., CARLA, nuScenes) that measure latency, safety metrics (time-to-collision), and explainability fidelity.

7.4 Finance

Financial AI agents operate in a **disembodied, high-throughput** environment where decisions must be both fast and compliant:

- **Reactivity** - Market-making bots require sub-millisecond reaction to price changes; Subsumption-style pipelines with hardware-accelerated inference satisfy this need.

- **Proactiveness** - Strategic portfolio rebalancing and risk-aware planning rely on deliberative components (SPA) that forecast market trends and optimize long-term objectives.

- **Autonomy & Governance** - Fully autonomous trading agents must embed self-monitoring for regulatory compliance (e.g., market abuse detection). Semi-autonomous oversight (human-in-the-loop) is often mandated to satisfy legal frameworks, reflecting the governance axis of Section 3.

- **Learning & Adaptability** - Online reinforcement learning and meta-learning enable agents to adapt to regime shifts, while modular policy libraries support rapid switching between strategies.

- **Explainability** - Transparent model explanations (feature importance, counterfactuals) are required for auditability and to satisfy financial regulators, aligning with the explainability trait of Section 4.

- **Robustness** - Adversarial robustness against market manipulation and stress-testing under extreme volatility are critical; formal verification of risk constraints complements empirical robustness metrics from Section 6.

Hybrid architectures are prevalent: low-latency reactive order-execution layers coupled with deliberative risk-assessment modules. Evaluation combines quantitative metrics (latency, Sharpe ratio, drawdown) with qualitative compliance audits.

7.5 Healthcare

Healthcare agents span **embodied (e.g., assistive robots) and disembodied (e.g., diagnostic decision support)** contexts, where safety, trust, and ethical compliance dominate design choices:

- **Autonomy & Governance** - Due to high-stakes outcomes, most clinical agents are semi-autonomous, providing recommendations that clinicians must validate. This respects the autonomy continuum (Section 3) and aligns with the safety-first stance of the Introduction.

- **Reactivity** - Real-time monitoring (e.g., ICU alarms) demands ultra-low latency; Subsumption-style alerting pipelines ensure rapid response.

- **Proactiveness** - Predictive analytics (early disease detection) require deliberative planning and forecasting, supported by SPA architectures that can generate actionable care plans.

- **Learning & Adaptability** - Continuous learning from patient data improves diagnostic accuracy, but must be coupled with strict privacy safeguards and model-drift detection to maintain clinical validity.

- **Explainability** - Clinicians need transparent reasoning (e.g., saliency maps, natural-language justification) to trust AI recommendations, directly reflecting the explainability characteristic of Section 4 and the evaluation metrics of Section 6 (explainability fidelity).

- **Robustness** - Redundant sensing (multiple vitals), formal verification of decision rules, and graceful degradation (fallback to human oversight) protect against sensor failures and adversarial attacks.

- **Social Ability** - Patient-facing agents (chatbots, therapeutic robots) must model user emotions and cultural norms, employing theory-of-mind and negotiation protocols from the MAS perspective.

Evaluation combines clinical outcome metrics (diagnostic accuracy, treatment efficacy) with safety indicators (false-alarm rate), explainability scores, and user-centered studies measuring trust and satisfaction.

7.6 Synthesis Across Domains

Across all five domains, the **interplay of the eight core characteristics** (Section 4) with the **three-dimensional taxonomy** (Section 3) determines the optimal architectural pattern (Section 5) and the appropriate evaluation suite (Section 6). A concise mapping emerges:

Domain	Dominant Taxonomy Position	Preferred Architecture	Key Characteristic Emphasis
Robotics	Embodied, Reactive Deliberative	Hybrid (reactive base + deliberative planner)	Reactivity, Robustness, Adaptability
Virtual Assistants	Disembodied, Semi-autonomous	SPA (explainable, proactive)	Social Ability, Explainability, Learning
Autonomous Vehicles	Partially Embodied, Semi-autonomous	Hybrid (subsumption + SPA)	Safety (Robustness), Proactiveness, Explainability
Finance	Disembodied, Reactive Deliberative	Hybrid (low-latency execution + deliberative risk layer)	Reactivity, Governance, Explainability
Healthcare	Mixed (embodied & disembodied), Semi-autonomous	SPA with safety supervisors	Explainability, Robustness, Social Ability

These patterns illustrate that **no single architecture or characteristic dominates universally**; instead, domain constraints dictate which traits must be prioritized and which trade-offs are acceptable. The systematic framework presented throughout the publication thus enables practitioners to reason rigorously about agent design choices tailored to their specific application context.

8. Challenges and Open Issues

8.1 Trade-offs Between Autonomy and Control

The unified taxonomy introduced in **Section 3** positions agents along a *Control* axis (Reactive Deliberative) and a *Governance* axis (Fully autonomous Semi-autonomous). While higher autonomy promises self-monitoring, self-diagnosis, and self-correction (see **Section 4 - Autonomy**), it also reduces the ability of external operators to intervene, raising safety and alignment concerns.

- **Latency vs. Oversight** - Reactive agents excel in low-latency reactivity (Section 5 - Subsumption), but their fast loops leave little room for supervisory checks, making it difficult to enforce alignment constraints in real time.

- **Planning Horizon vs. Predictability** - Deliberative agents provide richer world models and proactive planning (Section 4 - Proactiveness) but introduce planning latency that can hinder timely human overrides, especially in safety-critical domains such as autonomous vehicles (Section 7).

- **Governance Continuum** - Semi-autonomous designs (Section 3) mitigate these tensions by retaining a human-in-the-loop, yet they impose additional coordination overhead and can degrade performance metrics (latency, throughput) measured in **Section 6**.

Balancing these dimensions remains an open systems-engineering problem: how to dynamically adjust the autonomy level in response to context while preserving guarantees on safety and performance.

8.2 Safety and Robustness Challenges

Robustness is a core characteristic (Section 4) achieved through redundant sensing, formal verification, and adversarial training. Nevertheless, several limitations persist:

- **Formal Guarantees for Hybrid Architectures** - Hybrid designs (Section 5) combine reactive and deliberative layers, but existing verification tools primarily target either pure reactive loops or pure planning modules. Integrated proofs that cover the arbitration logic are still lacking.

- **Graceful Degradation in Embodied Settings** - Embodied agents (Section 3 - Embodiment axis) must maintain safety under hardware failures or sensor degradation. Current fallback strategies are heuristic and not systematically evaluated across the benchmark suites listed in **Section 6**.

- **Robustness-Learning Trade-off** - Continuous learning (Section 4 - Learning) can improve adaptability but may inadvertently reduce adversarial robustness, a tension observed empirically in the benchmark results (Section 6).

Research is needed to develop compositional safety frameworks that can be applied uniformly across the taxonomy's axes.

8.3 Interpretability and Explainability Limits

Explainability (Section 4) is essential for trust, regulatory compliance, and debugging. The current state of the art still faces several hurdles:

- **Scalability of Post-hoc Attribution** - Techniques that generate natural-language explanations for large language model-driven agents (Section 7 - Virtual Assistants) often scale poorly with model size, leading to latency spikes that conflict with the reactivity requirements of certain domains (e.g., robotics).

- **Alignment of Explanations with Human Mental Models** - High-fidelity explanation metrics (Section 6) capture similarity to ground-truth rationales, yet they do not guarantee that users find the explanations useful or actionable, especially in high-stress environments like healthcare.

- **Explainability in Multi-Agent Systems** - MAS architectures (Section 5) produce emergent group behavior that is difficult to attribute to individual agents, limiting the applicability of current explainability tools.

Open questions include how to design explanation mechanisms that are both computationally efficient and cognitively appropriate across diverse application domains.

8.4 Alignment and Ethical Compliance

Alignment - ensuring that agents pursue goals consistent with human values - is a cross-cutting challenge that touches autonomy, safety, and explainability.

- **Dynamic Value Specification** - Semi-autonomous agents (Section 3) rely on external oversight to update objectives, but there is no standardized protocol for safely injecting new value constraints at runtime without destabilizing learned policies.

- **Bias Mitigation in Continual Learning** - Continuous learning pipelines (Section 4 - Learning) can unintentionally amplify biases present in streaming data, and current fairness metrics (Section 6) are insufficient for detecting subtle drifts over long deployment horizons.

- **Regulatory Traceability** - While explainability metrics provide post-hoc justification, they do not yet satisfy the auditability requirements of sectors such as finance and healthcare (Section 7), where regulatory bodies demand verifiable decision logs linked to policy versions.

Addressing these gaps requires unified frameworks that couple alignment monitoring with the architectural patterns discussed in **Section 5**.

8.5 Open Research Questions

The challenges above give rise to a set of high-impact research questions that remain unresolved:

1. **Adaptive Autonomy Management** - How can agents autonomously adjust their position on the autonomy-control continuum in response to environmental uncertainty while preserving provable safety guarantees?

2. **Compositional Verification for Hybrid Systems** - What formal methods can jointly verify reactive loops, deliberative planners, and learning modules, including their arbitration mechanisms?

3. **Scalable, Context-Aware Explainability** - Can we develop explanation generators that operate within strict latency budgets and adapt their granularity to the user's expertise and the task's criticality?

4. **Continuous Alignment Assurance** - What monitoring architectures enable real-time detection of value misalignment in agents that continuously learn from online data streams?

5. **Benchmark Evolution for Emerging Paradigms** - How should the benchmark suites (Section 6) be extended to evaluate self-evolving agents and LLM-driven architectures that blur the line between reasoning and learning?

6. **Cross-Domain Transfer of Safety Guarantees** - Is it possible to transfer robustness certificates obtained in simulated environments to embodied deployments (e.g., robotics, autonomous vehicles) without costly re-verification?

Answering these questions will be pivotal for advancing the field toward trustworthy, high-performing AI agents that can be safely deployed across the diverse domains outlined in **Section 7**.

9. Future Directions

9.1 Self-Evolving Agents

The **Challenges and Open Issues** (Section 8) highlight the tension between autonomy and safety, especially when agents continuously modify their own policies. A natural evolution is the emergence of *self-evolving agents* that can autonomously restructure their internal architectures - adding, removing, or re-weighting modules - while preserving the guarantees required for **robustness** and **explainability** (Section 4).

Key research avenues include:

1. **Adaptive Autonomy Levels** - Building on the autonomy-control trade-off identified in Section 8, future agents should be able to *dial* their autonomy up or down in response to contextual risk assessments, using formal safety envelopes that are re-validated on-the-fly.

2. **Compositional Verification for Evolution** - Extending the verification gaps noted for hybrid arbitration (Section 5) to support *incremental* proofs that new or modified components respect existing safety contracts.

3. **Meta-Learning for Architecture Search** - Leveraging meta-learning techniques to let agents discover more efficient sense-plan-act or subsumption hybrids for a given task domain, thereby improving **adaptability** without sacrificing **reactivity**.

By integrating these capabilities, self-evolving agents would close the loop between the *learning* and *adaptability* traits (Section 4) and the *design patterns* discussed in Section 5, enabling continuous improvement beyond static training pipelines.

9.2 Deep Integration with Large Language Models (LLMs)

Section 2 identified LLM-driven agents as an emerging paradigm that falls outside classic taxonomies. The **Future Directions** must therefore extend the three-dimensional taxonomy (Section 3) to accommodate *LLM-centric control* as a distinct sub-axis:

- **LLM-Mediated Deliberation** - LLMs can serve as high-level planners that generate symbolic goals or natural-language policies, which are then instantiated by lower-level reactive modules. This hybridization preserves the *proactiveness* and *social ability* strengths of SPA (Section 5) while retaining the low-latency *reactivity* of subsumption layers.
- **Prompt-Based Explainability** - Leveraging the LLM's natural-language generation to produce context-aware explanations aligns with the **explainability** trait (Section 4). However, as Section 8 notes, post-hoc explanations for LLM agents face scalability and latency challenges; future work should explore *tight coupling* of explanation generation with the agent's decision pipeline to meet real-time constraints.
- **Continual LLM Fine-Tuning** - To address the **learning** and **adaptability** characteristics, agents will need mechanisms for safe, incremental fine-tuning of their LLM components, possibly through reinforcement-learning-from-human-feedback (RLHF) loops that respect the *alignment* safeguards discussed in Section 8.

Integrating LLMs in this manner promises richer *social ability* and *proactiveness* while demanding new evaluation metrics (Section 6) that capture language fidelity, bias drift, and explanation latency.

9.3 Standards and Frameworks for Trustworthy Agent Behavior

The need for *trustworthy* AI agents is a recurring theme across Sections 4, 5, 6, 7, and 8. To operationalize the traits of **robustness**, **explainability**, **ethical compliance**, and **safety**, the community is moving toward formal standards and certification frameworks. Anticipated developments include:

1. **Unified Agent Specification Language (UASL)** - A domain-agnostic schema that encodes an agent's position in the taxonomy (Section 3), its characteristic profile (Section 4), and its safety contracts. UASL would enable automated generation of verification artifacts and facilitate interoperability across domains (Section 7).

2. **Trustworthiness Scorecards** - Extending the balanced scorecard approach of Section 6, future benchmarks will incorporate *trust metrics* such as explanation fidelity, bias drift, and alignment drift over

continual learning cycles. These scorecards will be standardized by bodies such as IEEE or ISO, providing a common baseline for regulatory compliance in finance and healthcare.

3. **Certification Pipelines for Self-Evolving and LLM-Enhanced Agents** - Building on the open research questions in Section 8, certification processes will need to support *incremental* evidence collection, allowing agents to retain their certificates as they evolve or fine-tune LLM components.
4. **Open Benchmark Suites for Emerging Paradigms** - New benchmark collections (e.g., **Self-Evolve-Bench**, **LLM-Agent-Eval**) will complement the existing suites listed in Section 6, providing reproducible testbeds for evaluating the interplay of evolution, language understanding, and trustworthiness.

These standards will close the loop between *design* (Section 5), *evaluation* (Section 6), and *deployment* (Section 7), ensuring that future agents not only excel in performance but also meet societal expectations for safety and accountability.

9.4 Synthesis and Outlook

Collectively, the emerging trends of self-evolving capabilities, deep LLM integration, and formal trustworthiness standards point toward a new generation of AI agents that are **self-optimizing**, **language-aware**, and **certifiably safe**. Realizing this vision will require:

- Extending the **unified taxonomy** (Section 3) to capture dynamic autonomy levels and LLM-mediated control.
- Enhancing **architectural patterns** (Section 5) with modular evolution hooks and language-model interfaces while preserving the characteristic-architecture mapping.
- Evolving **evaluation metrics** (Section 6) to include trust-centric dimensions and continuous certification feedback loops.
- Addressing the open research questions enumerated in Section 8, especially around compositional verification and adaptive alignment.

By aligning these research directions with the core characteristics and design principles established throughout the publication, the field can progress toward AI agents that are not only more capable but also reliably trustworthy across the diverse application domains outlined in Section 7.

10. Conclusion

10.1 Recap of Core Findings

This paper has presented a **unified taxonomy** (Section 3) that positions AI agents along three independent axes - Control (Reactive Deliberative), Governance (Fully autonomous Semi-autonomous), and Embodiment (Embodied Disembodied). By mapping each axis to the eight **core characteristics** (autonomy, reactivity, proactiveness, social ability, learning, adaptability, explainability, robustness) described in **Section 4**, we established a common language for analyzing and designing agents.

The **design and architectural patterns** surveyed in **Section 5** (Sense-Plan-Act, Subsumption, Hybrid, Multi-Agent Systems) were explicitly linked to these characteristics, providing a practical decision matrix for selecting the most suitable pattern given a taxonomy position.

A **metric taxonomy** (Section 6) aligned quantitative indicators (latency, success-rate, robustness scores, explainability fidelity, ethical compliance) and qualitative techniques (scenario walkthroughs, human-in-the-loop studies) with the same characteristic set, enabling systematic evaluation and trade-off analysis.

Application-domain case studies (Section 7) demonstrated how the interplay of taxonomy, characteristics, architecture, and metrics shapes concrete deployments in robotics, virtual assistants, autonomous vehicles, finance, and healthcare.

Finally, we identified **critical challenges** (Section 8) - autonomy vs. control, safety-robustness trade-offs, scalable explainability, and alignment - and outlined **future research directions** (Section 9) such as self-evolving agents, deep LLM integration, and emerging standards for trustworthy behavior.

Together, these contributions deliver a **coherent framework** that bridges historical taxonomies (Section 2) with modern, data-driven agent designs, and that can be reused across domains.

10.2 Why a Systematic Understanding Matters

- **Predictable Trade-offs** - The three-dimensional taxonomy makes explicit where an agent's strengths and weaknesses lie, allowing designers to anticipate the impact of architectural choices on autonomy, reactivity, and robustness.
- **Design Reuse** - By locating an agent within the taxonomy, engineers can select a baseline pattern (e.g., Subsumption for latency-critical tasks, SPA for proactive, explainable agents) and then layer learning, adaptation, or coordination modules in a principled way (Section 5).
- **Rigorous Evaluation** - The balanced-scorecard approach (Section 6) ensures that performance, safety, transparency, and ethical compliance are measured together, preventing the "performance-only" pitfall that has hampered many deployments.
- **Cross-Domain Consistency** - The same set of eight characteristics applies to both embodied robots and disembodied virtual assistants, enabling knowledge transfer and comparative studies across the diverse application domains highlighted in Section 7.
- **Foundation for Standards** - A systematic taxonomy and characteristic profile are prerequisites for the **Unified Agent Specification Language (UASL)** and trustworthiness scorecards proposed in Section 9, which aim to become industry-wide certification artifacts.

10.3 Pathways for Advancing the Field

1. **Extend the Taxonomy to Emerging Control Axes** - Incorporate LLM-driven high-level planning (Section 9) as a fourth axis, and formalize mechanisms for dynamic shifting of autonomy levels (challenge identified in Section 8).
2. **Develop Compositional Verification Tools** - Build verification frameworks that can jointly certify reactive loops, deliberative planners, and learning modules, addressing the safety gap noted in Section 8.
3. **Standardize Benchmark Suites** - Release and maintain the proposed *Self-Evolve-Bench* and *LLM-Agent-Eval* (Section 9) alongside the existing benchmark families (Section 6) to evaluate the full spectrum of characteristics for next-generation agents.

4. **Integrate Explainability into the Control Loop** - Research low-latency, context-aware explanation generators that satisfy both quantitative fidelity metrics and qualitative usability constraints (Section 8).
5. **Formalize Alignment Protocols for Continuous Learning** - Design runtime value-injection and bias-drift detection mechanisms that can be audited through traceable decision logs, fulfilling the ethical compliance requirements outlined in Section 8 and Section 9.
6. **Promote Open, Interoperable Specification Languages** - Advance the development of UASL (Section 9) to encode taxonomy position, characteristic profiles, safety contracts, and verification evidence, facilitating cross-vendor collaboration and regulatory review.
7. **Foster Multi-Disciplinary Collaboration** - Encourage joint efforts among AI researchers, formal methods experts, ethicists, and domain specialists to co-design agents that meet the stringent robustness, explainability, and alignment standards demanded by high-risk sectors such as healthcare and autonomous transportation.

By pursuing these pathways, the community can transform the systematic understanding presented in this work into **practical, trustworthy AI agents** that reliably serve the growing spectrum of real-world applications.