TRANSPARENCY IN AGENTIC AI: A SURVEY OF INTERPRETABILITY, EXPLAINABILITY, AND GOVERNANCE

Shaina Raza[1,*], Ahmed Y. Radwan[1], Sindhuja Chaduvula[1], Mahshid Alinoori[1], Christos Emmanouilidis[2]

[1]Vector Institute for Artificial Intelligence, Toronto, Canada
[2]University of Groningen, The Netherlands

[*]Corresponding author: shaina.raza@vectorinstitute.ai

February 9, 2026

ABSTRACT

Agentic AI systems built on large language models plan, use tools, and maintain memory over multiple steps. Their risks and responsibilities depend on an execution trajectory rather than a single output. Despite progress, work on transparency for such systems remains scattered. Most explainability and interpretability research still targets static or single step model outputs, while Agentic AI surveys emphasize planning, tools, and memory, giving limited attention to transparency and oversight. Literature insufficiently addresses what should be subject to transparency and recorded during an agent's lifecycle and how to verify records. Addressing this need, this survey offers a transparency-focused analysis by connecting interpretability, explainability, and governance for Agentic AI systems from design to deployment and synthesizing relevant methods for agent artifacts, including plans, tool interactions, memory events, and coordination signals, relating them to assurance needs such as faithfulness, auditability, compliance, robustness, and equity. The paper consolidates evaluation practices and highlights gaps, especially in trajectory level accountability, tool mediated provenance, and multi-agent coordination transparency. It proposes the Minimal Explanation Packet as standardized outcome artifact bundling key lifecycle evidence into an audit-ready record. The survey serves as a reference for researchers and practitioners to consistently compare approaches, design evaluations, and report transparency evidence.
Project Page: https://vectorinstitute.github.io/Agentic-Transparency/

***K*eywords** Agentic AI; Intelligent agents; Explainable AI (XAI); Model interpretability; Multi-agent systems; Large Language Models; Reasoning and planning; AI transparency

1 Introduction

Agentic AI systems, which are large language models (LLMs) augmented with planning, memory, tool use, and autonomous decision-making, are rapidly moving from research prototypes to production deployments [49, 20]. Unlike traditional machine learning (ML) models that produce a single output from a single input, these LLM-based agents maintain state across interactions, reason over multi-step goals, coordinate with other agents, and take actions with real-world consequences [147]. This shift in capability demands a corresponding shift in how we approach transparency. Explaining an agent requires explaining not only *what* it concluded, but also *how* it arrived there and *what that means* for the user, the organization, and society [31].

To date, the tools available for understanding AI systems have largely been built for a different era. Interpretability methods developed for static classifiers [60] and post-hoc explainability techniques designed for single predictions [8] struggle to capture the temporal, stateful, and interactive nature of agentic behavior. When an agent plans a sequence of actions, updates its beliefs based on tool outputs (e.g., search APIs, database queries, code execution), and coordinates

with other agents before producing a final outcome, existing explainable AI (XAI)[1] methods are largely limited to isolated steps and fail to trace accountability across the full decision-making process. This gap matters because users confronting an agent's recommendation face three recurring questions: (1) **Why did I get this result?** (Cause); (2) **How did the agent arrive at it?** (Process); and (3) **What does it mean for me?** (Impact). Addressing these questions in Agentic AI systems demands a shift away from viewing explanations as one-time, post-hoc artifacts and towards treating transparency as a *lifecycle property*, one that is embedded in the agent's architecture, monitored throughout execution, and evaluated against its outcomes. A taxonomy of transparency in agentic AI presented in this paper is presented in **Fig. 1** and discussed in **Section 3**.

1.1 The Transparency Gap in Agentic AI

Throughout this survey, we use the term *Agentic AI* to refer to LLM-based autonomous agents, which are systems that integrate LLMs with planning, memory, and tool-use capabilities to operate over multiple steps and produce real-world effects, consistent with recent literature [49]. In this context, we use *transparency* as an umbrella term that includes interpretability (insight into internal mechanisms), explainability (user-facing rationales), and auditability (support for verification and accountability), following prior work [42, 90] on transparent and accountable AI systems.

There is an urgent need for transparency in Agentic AI. This urgency arises because investment and deployment are accelerating rapidly, while transparency research and tooling are not keeping pace. **Fig. 2a** illustrates this divergence, showing deployment growth exceeding transparency tooling by approximately $6\times$ by 2034. This imbalance is also evident in research activity. As shown in **Fig. 2b**, publications on Agentic AI are increasing substantially faster than those on XAI. More than 90% of enterprises report near-term interest in Agentic AI; however, only 2% had deployed such systems at scale by early 2025, and just 15% report having XAI infrastructure in place (**Fig. 2c**) [41, 82]. These figures suggest that governance and assurance gaps (rather than lack of interest or technical capability) are the primary barriers to large-scale deployment.

The sectors adopting Agentic AI are also those with the strongest transparency requirements. As shown in **Fig. 2d**, banking, financial services, and insurance, healthcare, and government face high regulatory demands for transparency while actively deploying agentic systems. Governance frameworks such as the EU AI Act [35], NIST AI RMF [92], and ISO/IEC 42001 [51] mandate varying degrees of explainability for high-risk AI systems; however, the explainability methods referenced in these frameworks were largely developed for static models rather than for Agentic AI. The research landscape is similarly fragmented; we summarize this divide in **Table 1** and **Fig. 3**.

1.2 Necessity of this Survey

As **Table 1** and **Fig. 3** show, prior survey articles fall into two largely non-overlapping groups. The first group comprises XAI and interpretability surveys, which focus predominantly on static or single-step models; recent LLM-oriented reviews in this category remain centered on individual model outputs rather than agentic workflows [164]. **Fig. 4** further shows that Agentic AI capabilities have advanced faster than XAI methods. The second group comprises surveys on Agentic AI [78, 159, 152], which comprehensively cover architectures, planning, tools, and memory but treat transparency as a secondary concern or defer it entirely. To address this gap, we present a survey that bridges these two strands by offering a unified treatment of interpretability and explainability for LLM-based agentic systems.

1.3 Scope and Approach

We survey interpretability and explainability methods for Agentic AI systems, focusing on how transparency can be achieved in these settings. Methodologically, our literature review spans 2017–2025 across major publisher databases including SpringerLink, Elsevier ScienceDirect, the ACM Digital Library, and IEEE Xplore, as well as across Google Scholar and arXiv. We used keyword combinations including *agentic AI*, *LLM agents*, *tool use*, and *transparency* (interpretability and explainability) to search the relevant literature. We prioritize methods that (i) expose or test internal mechanisms, traces, or decision factors, (ii) agent-specific artifacts (e.g., plans, tool I/O, memory writes/retrievals, coordination signals), and (iii) governance or assurance frameworks to operationalize transparency. Consistent with common practice in computer science survey articles [145, 62], we do not present new empirical validation through deployed systems, reference implementations, or controlled user studies; instead, we outline these validation directions as future work (**Section 7**). Building on foundational taxonomic work in explainable and interpretable AI transparency by Lipton [70], Gilpin et al. [40], and Murdoch et al. [90], we extend these conceptual distinctions to Agentic AI systems. As illustrated in **Fig. 3**, our work bridges a critical gap between traditional XAI/interpretability surveys focused on static models and emerging Agentic AI surveys with limited transparency coverage. As shown in **Table 1**,

[1] Here, we use XAI to refer to traditional explainability and interpretability methods.

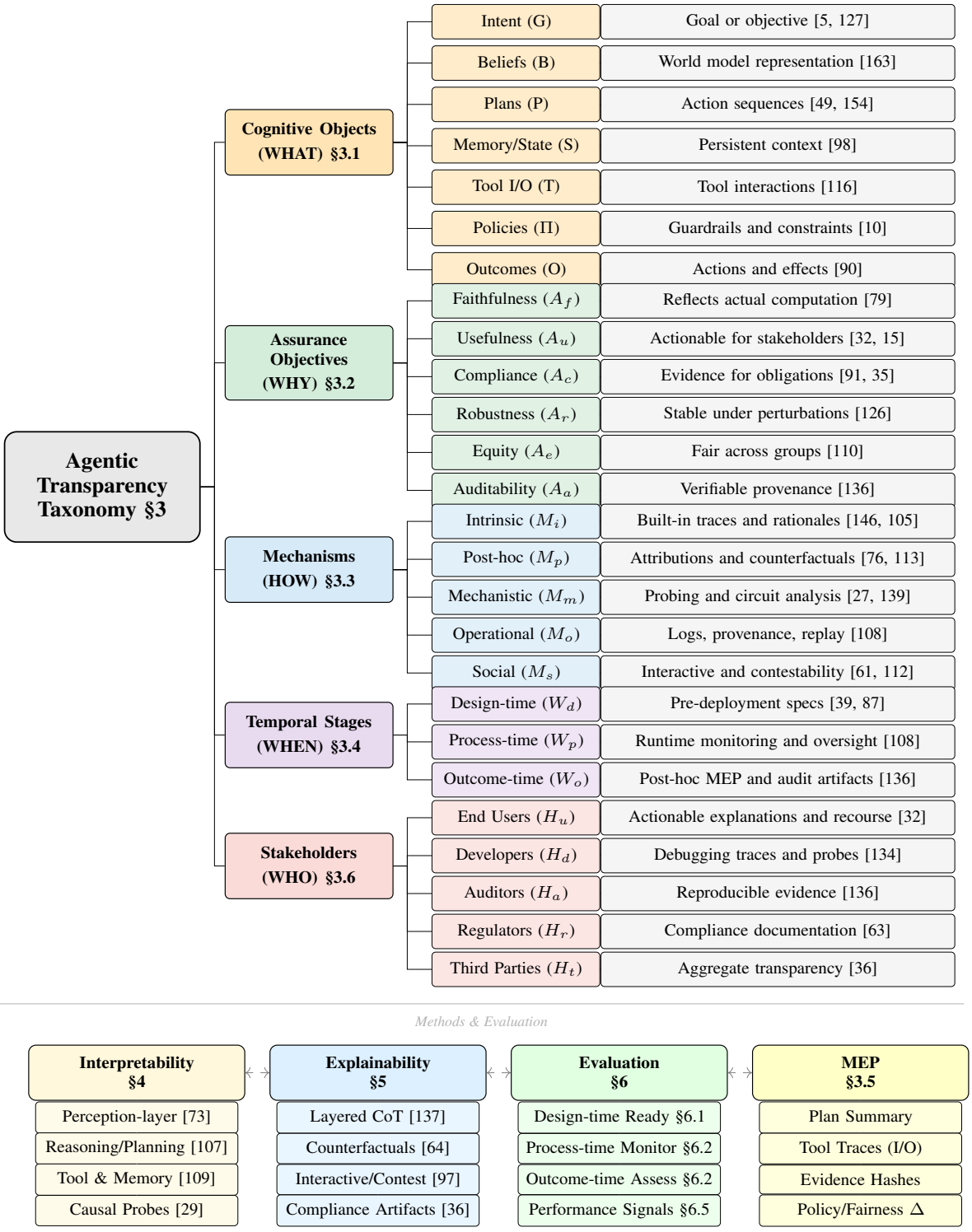
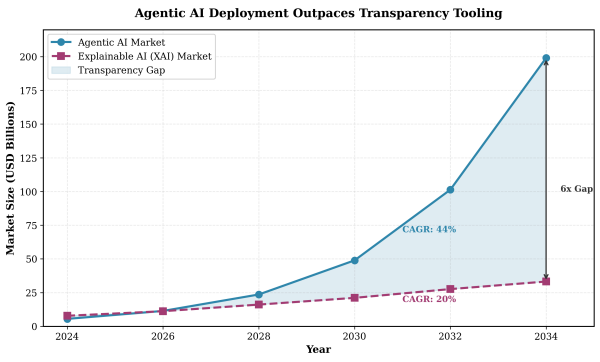
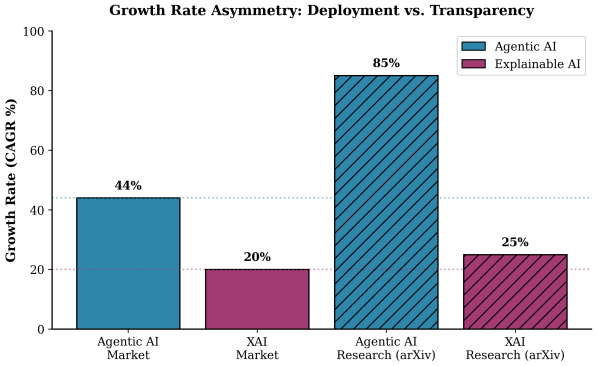


Figure 1: Taxonomy of transparency in agentic AI, organized for interpretability, explainability, and evaluation.

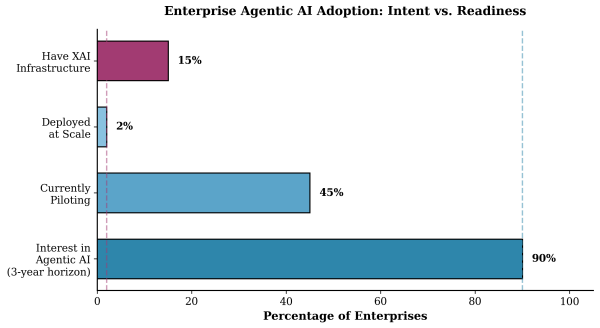
our survey uniquely integrates explainability, interpretability, and Agentic AI coverage, which have previously been addressed in isolation in the existing literature.



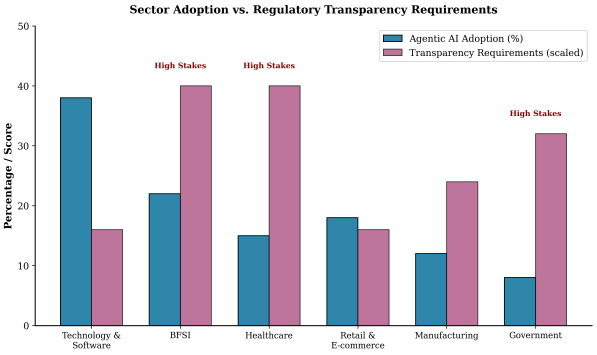
(a) Market growth projection



(b) Growth rate asymmetry



(c) Enterprise adoption gap



(d) Sector adoption vs. regulatory requirements

Figure 2: The transparency gap in Agentic AI. Across market, research, and enterprise adoption, agentic deployments outpace transparency tooling, and the most active sectors face the strongest transparency and regulatory requirements. Four subplots summarizing the transparency gap: market growth projections, publication growth rates, enterprise adoption gap, and sector adoption versus regulatory requirements.

Table 1: Comparison of related surveys. Symbols: ✓ = explicit focus, ✗ = no dedicated treatment.

Related survey	Year	Explainability	Interpretability	Agentic	Notes
[32]	2017	✗	✓	✗	Defines interpretability as human-simulatability; evaluation levels.
[70]	2018	✓	✓	✗	Intrinsic transparency vs. post-hoc; simulatability/decomposability.
[90]	2019	✓	✓	✗	Model-based (intrinsic) vs. post-hoc; PDR evaluation.
[50]	2023	✓	✗	✗	General XAI overview.
[164]	2024	✓	✓	✗	LLM-focused XAI; local/global; prompting vs. fine-tuning.
[149]	2025	✗	✓	✗	Usable interpretability methods.
[12]	2025	✓	✓	✗	LLMs as explainers (XAI tooling).
[78]	2025	✗	✗	✓	LLM agent methodologies/workflows.
[69]	2025	✓	✓	✗	Broad XAI taxonomy.
[159]	2025	✗	✗	✓	Agent workflows: planning, tools, memory.
[99]	2025	✗	✗	✓	Autonomous agents review.
[152]	2025	✗	✗	✓	Tool-use agents survey.
[96]	2025	✓	✓	✗	Usability-oriented XAI.
This survey	2026	✓	✓	✓	Transparency for LLM-based agents.

1.4 Contributions

This survey makes three primary contributions:

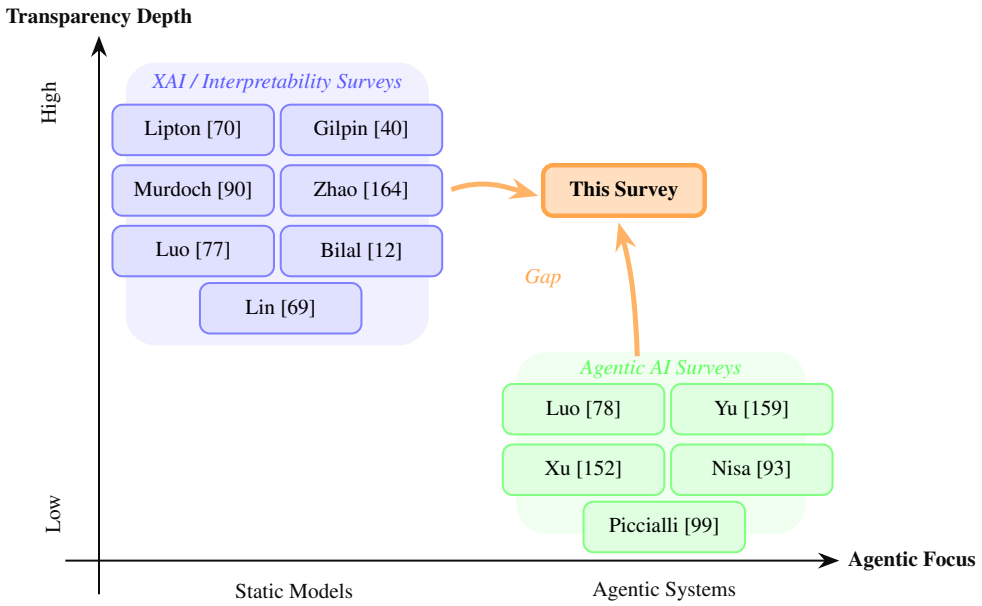


Figure 3: Positioning of this survey relative to existing literature. Prior work clusters in two regions: XAI/interpretability surveys focused on static models (left), and Agentic AI surveys with limited transparency coverage (bottom-right). This survey addresses transparency challenges specific to agentic systems.

A 2D positioning diagram showing how prior surveys cluster around static-model XAI/interpretability versus agentic-AI surveys, with this survey highlighted as bridging the two areas.

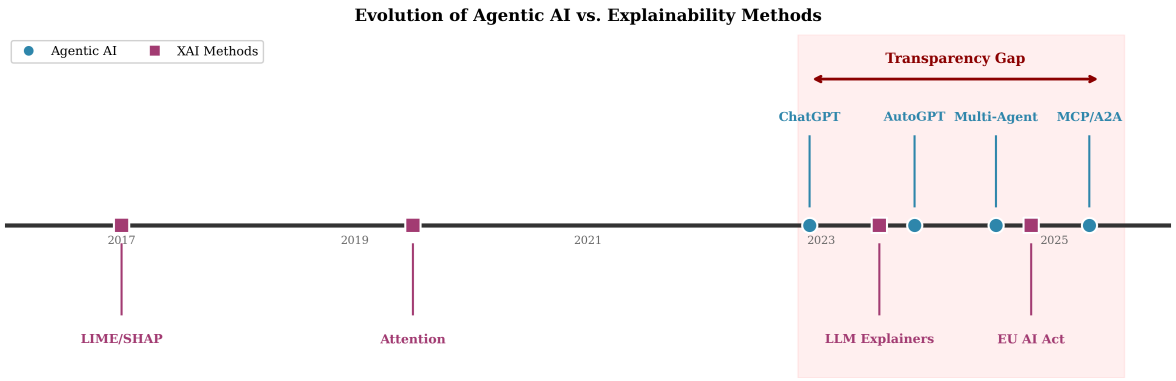


Figure 4: Evolution of Agentic AI milestones versus XAI methods. The shaded region highlights the "transparency gap" period (2022–present) where Agentic AI development has accelerated while explainability methods remain focused on static models.

A timeline comparing key milestones in agentic AI with milestones in explainability and interpretability methods; a shaded region marks the period since 2022 where agentic AI progress outpaces transparency methods.

1. **Synthesis.** We consolidate research across interpretability, explainability, *mechanistic analysis*, and agentic AI monitoring to characterize the current state of transparency for LLM-based agents, covering single-agent, multi-agent, and multimodal settings.
2. **Taxonomy.** We propose a five-axis framework for organizing agentic transparency research along: (i) *what* is made transparent (cognitive objects: goals, plans, memory, tool I/O, policies, outcomes), (ii) *why* transparency is required (assurance objectives: faithfulness, usefulness, compliance, robustness, equity, auditability), (iii) *how* transparency is achieved (mechanisms: intrinsic, post-hoc, *mechanistic*, operational, social), (iv) *when* transparency applies (lifecycle stages: design-time, process-time, outcome-time, post-deployment), and (v) *who* requires transparency (stakeholders: end users, developers, auditors, regulators, affected third parties).
3. **Gap Analysis and Research Agenda.** We map existing methods to assurance objectives and governance frameworks (EU AI Act, NIST AI RMF, ISO/IEC 42001), identify critical gaps—particularly in trajectory-

level accountability, tool-mediated provenance, and multi-agent coordination—and propose the Minimal Explanation Packet (MEP) as a standardized outcome-time artifact. We outline a research agenda for empirical validation as future directions for the research community. A conceptual overview of the survey is provided in **Fig. 1**.

2 Background

2.1 Interpretability and Explainability

The terms *interpretability* and *explainability* are often used interchangeably, but we follow a distinction that has emerged in recent literature [70, 40, 90]: **Interpretability** refers to the degree to which a human can understand the internal mechanisms of a model, such as its representations, computations, and decision boundaries [60]. Interpretability is typically *model-facing*: it concerns what the system is doing internally, and its primary audience is developers, researchers, and auditors. **Explainability** refers to the capacity to provide post-hoc rationales or justifications for a model's outputs in terms that are meaningful to end users [156]. Explainability is typically *user-facing*: it concerns why a particular output was produced, and its primary audience is the people affected by or relying on the system.

This distinction becomes especially important for Agentic AI systems. A developer debugging an agent's planning loop requires interpretability, i.e., access to internal reasoning traces, belief states, and decision points. A user receiving a recommendation from that agent requires explainability, i.e., a coherent justification for why the agent acted as it did. The two needs are related but not identical [70, 40, 90]. We use **transparency** as an umbrella term encompassing both interpretability and explainability, along with broader properties such as audibility and traceability that governance frameworks increasingly require [35, 92, 51].

2.2 Characterizing Agentic AI Systems

Table 2: Notations used throughout the survey.

Symbol	Description	Symbol	Description
A	Agentic AI system	M	LLM-based reasoning module(s)
G	Goal or objective	P	Planning / decomposition module
T	Tool set and tool I/O traces	S	State or memory
B	Belief state / world model	H_u	End-user stakeholder class
H_d	Developer / operator stakeholder class	H_a	Auditor stakeholder class
H_r	Regulator stakeholder class	H_t	Affected third-party stakeholder class
M_i	Intent/goal explanation mechanisms	M_p	Plan/trajectory explanation mechanisms
M_t	Tool/evidence (provenance) mechanisms	M_u	Uncertainty/confidence mechanisms
M_π	Policy/guardrail mechanisms	M_o	Outcome/impact mechanisms
E	Environment	π	Policy / guardrails
a_t	Action at step t	o_t	Observation at step t
s_t	Agent state at step t	O	Outcome / final output
e	Explanation artifact	f	Faithfulness metric
A_f	Faithfulness objective	A_u	Usefulness objective
A_c	Compliance objective	A_r	Robustness objective
A_e	Equity objective	A_a	Auditability objective
W_d	Design-time stage	W_p	Process-time stage
W_o	Outcome-time stage	MEP	Minimal Explanation Packet
τ	Trace–plan agreement threshold	ϵ	Fairness / parity tolerance
Δ	Measured parity gap	Π	Policies surfaced in explanations

We define an **Agentic AI system** as an LLM-based system that operates over multiple steps with persistent state and interaction with external tools or an environment. **Table 2** summarizes the notation used in this survey.

2.3 Agentic AI Architectures and Lifecycle

We distinguish three common configurations of Agentic AI architectures. **Single-agent systems** involve a single LLM-based agent that interacts with tools and an environment (e.g., ReAct [155], Toolformer [116], and AutoGPT-style systems [124]). In these settings, transparency challenges typically center on tracing multi-step plans, justifying tool selection, and attributing outcomes to memory retrieval and state updates. **Multi-agent systems** comprise multiple LLM-based agents that coordinate to accomplish tasks (e.g., ChatDev [104], AutoGen [147], and CAMEL [66]).

Here, transparency needs inspection of inter-agent communication, role attribution, and coordination dynamics that may emerge across interactions. **Multimodal agentic systems** integrate additional modalities such as vision or audio, alongside language, which introduces further transparency challenges related to cross-modal grounding [120], attribution, and consistency between modalities and actions. We treat transparency as lifecycle-dependent rather than a single property. Following governance frameworks such as NIST AI RMF [92] and ISO/IEC 42001 [51], we distinguish three temporal stages for transparency assurance. **Design-time** (pre-deployment) encompasses architectural choices—model selection, interface design, memory structure, and role definitions—that enable or constrain later interpretability. **Process-time** (operational phase) includes reasoning traces, plan monitoring, tool-call auditing, and state inspection during agent execution. **Outcome-time** (post-hoc phase) covers explanation generation, counterfactual analysis, contestability mechanisms, and audit logging. We subsume *end-of-lifecycle* under outcome-time, as its transparency requirements are met through outcome artifacts and governance controls.

2.4 Governance and Regulatory Context

Transparency requirements for AI systems are also increasingly formalized in governance frameworks and AI regulations. In this survey, we foreground three widely used references: the **EU AI Act** [35], which introduces risk-based obligations for high-risk systems (e.g., logging, human oversight, and user-facing information duties); the **NIST AI Risk Management Framework (AI RMF)** [92], which treats accountability and transparency as cross-cutting concerns across the GOVERN, MAP, MEASURE, and MANAGE functions; and **ISO/IEC 42001** [51], which specifies organizational management-system controls for responsible AI, including transparency and documentation practices.

Beyond the EU AI Act, NIST AI RMF, and ISO/IEC 42001, several complementary governance instruments reinforce transparency requirements for Agentic AI. The GDPR establishes safeguards for automated decision-making and associated disclosure duties[2] , while ISO/IEC 23894 provides lifecycle-oriented guidance for AI risk management, IEEE 7001 further treats transparency in autonomous systems as a measurable and testable property[3]. At the principles level, both the OECD AI Principles and the UNESCO Recommendation explicitly emphasize transparency, explainability, and human oversight. Recent international developments include the Council of Europe Framework Convention on AI, which frames lifecycle governance around human rights, democracy, and the rule of law. In Canada, the proposed AIDA under Bill C-27 similarly emphasizes risk mitigation for high-impact systems*.

Across these instruments, meaningful human oversight emerges as a consistent expectation. Several frameworks also address end-of-lifecycle transparency: the EU AI Act mandates post-market monitoring and withdrawal procedures (Articles 72–73), ISO/IEC 42001 requires documented decommissioning processes, and GDPR establishes data retention limits and deletion obligations. These requirements ensure that transparency obligations extend beyond active deployment to archival, handoff, and retirement. Building on these foundations, Section 3 turns the above concepts into a practical structure.

3 Taxonomy for Agentic AI Transparency

This section introduces a taxonomy for operationalizing transparency in agentic AI systems. In contrast to existing XAI taxonomies that emphasize static models, our taxonomy is designed for tool-using LLM agents where transparency is essential. We structure transparency along 5 complementary dimensions, each addressing a fundamental question:

1. **Cognitive Objects (WHAT):** Which internal states and interfaces should be made transparent?
2. **Assurance Objectives (WHY):** What goals should transparency serve?
3. **Mechanisms (HOW):** What technical means achieve transparency?
4. **Temporal Stages (WHEN):** At what point in the lifecycle is transparency required?
5. **Stakeholders (WHO):** Which audiences require transparency, and in what form?

We use these dimensions to compare methods that would otherwise be discussed in separate literature, to clarify what evidence each method produces, and to surface gaps that matter for deployment and evaluation. This five-dimensional structure builds on established organizing principles in related domains. For example, Yin et al. [157] organize privacy-preserving federated learning around a "5W" taxonomy (who, what, when, where, why), while Guidotti et al. [42] classify XAI methods by explanation type, scope, and model dependency. We extend these ideas to agentic AI by explicitly accounting for tool use, memory, multi-step trajectories, and multi-agent coordination. **Fig. 1** provides a visual overview, and **Table 3** summarizes the framework in an implementation-oriented form.

[2]GDPR Article 13 and GDPR Article 22.

[3]IEEE SA Autonomous and Intelligent Systems (AIS) Standards (incl. IEEE 7001).

Table 3: Five-axis summary: axes, core categories, representative artifacts/mechanisms, and practical evaluation signals.

Axis	Core categories	Representative artifacts / mechanisms / evaluation signals
WHAT (Cognitive objects)	Intent $\mathcal{G}$; beliefs/world model $\mathcal{B}$; plans $\mathcal{P}$; memory/state $\mathcal{S}$; tool I/O $\mathcal{T}$; policies/guardrails Π; outcomes/effects $\mathcal{O}$ [5, 71, 98, 116, 117, 90]	Goal verbalization/intent probes; plan graphs and rejected alternatives; memory and retrieval logs; tool-call traces (inputs/outputs/errors) with signatures; policy activation logs; outcome logging and effect attribution. *Signals:* object coverage, redaction policy, trace completeness, integrity checks
WHY (Assurance objectives)	Faithfulness; usefulness; compliance; robustness; equity; auditability [79, 32, 35, 92, 111, 126, 110, 136]	Faithfulness tests (trace–execution consistency); user task success and perceived helpfulness; documentation sufficiency for governance; stability under perturbations; disaggregated transparency/fairness checks; replayability and verifiable provenance. *Signals:* stability, disparity, reproducibility, retention compliance
HOW (Mechanisms)	Intrinsic; post-hoc; mechanistic; operational; social [146, 113, 76, 139, 136, 59, 108]	Intrinsic rationales / structured reasoning; model-agnostic attributions and counterfactuals; probing and causal interventions; signed logs, hashing, replay and provenance; interactive dialogue, critique/revision and contestability interfaces. *Signals:* approximation error, cost, privacy risk, user alignment
WHEN (Temporal stages)	Design-time; process-time; outcome-time [86, 39, 108]	Design-time: specs, model cards, risk assessment and logging plan. Process-time: monitoring + real-time trace integrity. Outcome-time: *Minimal Explanation Packet (MEP)* for each decision. *Signals:* readiness, monitoring effectiveness, post-hoc verifiability
WHO (Stakeholders)	End users; developers; auditors; regulators; affected third parties [32, 35, 92]	Tailored views: user-facing explanations + recourse; developer debugging traces; auditor-grade evidence and replay; regulator-ready reports and incident records; aggregate transparency for affected groups. *Signals:* role-appropriate utility, contestability, audit sufficiency

3.1 Cognitive Objects: What Should Be Transparent?

Traditional XAI methods typically explain a model's prediction by highlighting relevant input features or providing counterfactual alternatives. For Agentic AI systems, transparency extends beyond explaining outputs to making the *decision trajectory* inspectable. The goal is to see: *what the agent believed, what it intended to do, which tools it invoked* and *what constraints shaped the process*. We define **Cognitive objects** as the internal states and interfaces that shape an agent's behavior. We identify seven categories of cognitive objects as first-class transparency targets. For a given agent execution, we denote the transparency-relevant state as a tuple:

$$\mathcal{C} = \langle \mathcal{G}, \mathcal{B}, \mathcal{P}, \mathcal{S}, \mathcal{T}, \Pi, \mathcal{O} \rangle$$

where each component corresponds to one category: **intent** $\mathcal{G}$ (the goal or objective, explicit or inferred) [5]; **beliefs** $\mathcal{B}$ (the agent's representation of the world, including retrieved facts) [156]; **plans** $\mathcal{P}$ (action sequences, rejected alternatives, contingencies) [146]; **memory/state** $\mathcal{S}$ (persistent and transient context across interactions) [98]; **tool inputs/outputs** $\mathcal{T}$ (queries, responses, failures, fallbacks) [116]; **policies** Π (guardrails, safety constraints, governance controls) [108]; and **outcomes** $\mathcal{O}$ (actions taken, effects, and user-facing outputs) [90].

Design principle: If a cognitive object can influence downstream behavior, it should be observable for the relevant stakeholder and assurance objective. In practice, this requires defining which objects are in scope, logging or exposing them through traces, graphs, or metadata, and applying selective disclosure when privacy or security constraints apply.

3.2 Assurance Objectives: Why Is Transparency Required?

Transparency is not an end in itself but an enabler of assurance. This dimension specifies which objectives transparency should serve, recognizing that different stakeholders prioritize different goals. We define an assurance objective set $\mathcal{A} = \{A_f, A_u, A_c, A_r, A_e, A_a\}$ comprising six objectives that transparency mechanisms must support.

Faithfulness (A_f) asks whether the explanation reflects the system's actual computation and decision path, rather than an appealing post-hoc narrative [79]. This is particularly challenging for Agentic AI systems, where CoT traces may not correspond to internal reasoning processes. **Usefulness** (A_u) asks whether the transparency artifact is actionable for its intended audience such as users, developers, or auditors [32]. A technically complete trace may be useless to an end user seeking recourse. **Compliance** (A_c) asks whether artifacts provide evidence sufficient for legal, ethical, or organizational obligations [35, 92]. **Robustness** (A_r) asks whether explanations remain stable under reasonable perturbations and adversarial conditions [111, 126]. **Equity** (A_e) asks whether transparency surfaces potential bias and

avoids systematically disadvantaging protected groups [110]. **Auditability** (A_a) asks whether artifacts can be verified, reproduced, and traced over time with provenance guarantees [136].

Design principle. Agentic AI systems should be evaluated against a portfolio of objectives rather than a single proxy such as task accuracy, as discussed in detail in related work [36]. When agents rely on tools and memory, failures often occur in intermediate steps that remain invisible without explicit transparency mechanisms.

3.3 Mechanisms: How Is Transparency Achieved?

Mechanisms are the technical and procedural means through which transparency is achieved. We group mechanisms into five families $\mathcal{M} = \{M_i, M_p, M_m, M_o, M_s\}$, emphasizing that no single mechanism dominates across objectives and stages. **Intrinsic** (M_i) mechanisms build transparency into the model or agent behavior, such as structured reasoning traces and explicit plan representations [146]. **Post-hoc** (M_p) mechanisms produce explanations after the fact, including model-agnostic attributions and counterfactuals [113, 76]. **Mechanistic** (M_m) mechanisms interrogate or intervene on internal representations to increase causal grounding, for example, probing, activation patching, and circuit-level analysis [139]. **Operational** (M_o) mechanisms instrument the system to create verifiable traces, including signed logs, cryptographic hashing, replay capabilities, and provenance graphs [136, 108]. **Social** (M_s) mechanisms align explanations with human needs through interaction, critique/revision loops, and contestability interfaces [59].

Design principle. Select mechanisms as assurance tools matched to objectives: operational provenance (M_o) is central for auditability (A_a), while social mechanisms (M_s) often dominate perceived usefulness (A_u). When faithfulness (A_f) is high-stakes, mechanistic (M_m) and operational (M_o) evidence should constrain or validate intrinsic narratives.

3.4 Temporal Stages: When Is Transparency Required?

Transparency requirements and mechanisms shift across the system lifecycle. We distinguish three temporal stages $\mathcal{W} = \{W_d, W_p, W_o\}$ at which transparency must be addressed. **Design-time** (W_d) transparency establishes readiness before deployment: documenting intended behavior and limitations (e.g., model cards [86], datasheets [39]), defining which cognitive objects are in scope, and specifying how logs, policies, and monitoring will be implemented. **Process-time** (W_p) transparency supports real-time oversight: trace completeness, logging integrity, and monitoring effectiveness while the agent executes actions and uses tools [108]. **Outcome-time** (W_o) transparency supports post-hoc assessment after a decision: faithfulness (A_f) and robustness (A_r) checks, equity (A_e) analysis, and audit/replay.

We treat end-of-lifecycle transparency, including archival, retention, and decommissioning, as part of outcome-time because it is primarily operationalized through outcome artifacts and governance controls such as audit records, access controls, and retention policies, rather than through a distinct class of runtime transparency mechanisms.

3.5 Minimal Explanation Packet (MEP)

Outcome-time transparency requires a structured artifact that consolidates the evidence generated during design- and process-time into a verifiable, stakeholder-accessible form. To address this need, we propose the Minimal Explanation Packet (MEP), which is a standardized, outcome-time artifact designed to simultaneously support multiple transparency objectives, from real-time user explanations to post-hoc audits, without requiring redundant logging infrastructure. A typical MEP comprises: (i) a compact plan summary (including key decision points and rejected alternatives); (ii) evidence references (retrieval IDs and source hashes); (iii) tool traces (inputs, outputs, and errors); (iv) policy and fairness deltas (triggered guardrails and flagged disparities); (v) cryptographic signatures with timestamps [136, 108]; and (vi) resource consumption estimates (energy usage, carbon emissions, and compute steps where instrumentation permits) [75]. **Table 4** (Panel A) specifies the evidence requirements at each lifecycle stage for each cognitive object, ensuring MEP completeness and enabling downstream audit capabilities. **Fig. 5** illustrates a concrete MEP snapshot with its constituent components.

Design principle. Treat transparency as a lifecycle capability: design-time (W_d) sets scope and instrumentation, process-time (W_p) ensures integrity and oversight, and outcome-time (W_o) provides verifiable artifacts for evaluation.

3.6 Stakeholders: Who Requires Transparency?

Different stakeholders require different transparency views, even for the same underlying trace. We identify five stakeholder categories $\mathcal{H} = \{H_u, H_d, H_a, H_r, H_t\}$ and use them to guide what information is surfaced, at what level of detail, and in what format. **End users** (H_u) need understandable, actionable explanations, clear status indicators for long-running agent behavior, and recourse pathways when outcomes are adverse [32]. **Developers** (H_d) need debugging traces, interpretability probes, and structured artifacts that support error diagnosis and iterative refinement.

(A) Flow: From specs to release

Design-time specs
prompts, policies, logging plan, required fields

Runtime traces
plans, tool I/O, memory/belief updates, timestamps

MEP gate (X-Contract checks)
signatures; hash match; plan–trace agreement $\geq \tau$*; parity* $\Delta \leq \epsilon$

Outcome
Release decision + MEP or *Violation* $\rightarrow$ *flag, human review, rollback*

(B) Gate logic (pseudo-code)

```
Algorithm: MEP-Gate(decision d)
input: traces T_d, spec S, thresholds (tau,
epsilon)
1. MEP := assemble(T_d, S.required_fields)
2. if not signed(MEP) or not
hash_match(MEP, T_d) → FAIL
3. if plan_trace_agreement(T_d) < tau →
FAIL
4. if parity_delta(T_d) > epsilon → FAIL
5. log(MEP); RELEASE(d, MEP)
on FAIL: flag(d); notify_human(); rollback()
```

(C) Minimal Explanation Packet (MEP) — example

```
{ "run_id": "2025-11-07T11:22:45Z#agentic-vqa#042",
  "decision_summary": {"question": "...", "answer": "..." },
  "plan_graph": {"nodes":[...], "edges":[...], "rejected":["tool_bing_search"]},
  "tool_io": [ {"tool":"vision.embed","in":"img#sha256:...","out":"vec#..." },
     {"tool":"retriever","in":"query: ...","out":"doc_ids":[...],"error": null} ],
  "evidence_hashes": {"traces":"sha256:...", "inputs":"sha256:..." },
  "policy_fairness_deltas": {"gender_proxy": 0.02, "age_proxy": 0.01 },
  "resource_consumption": {"energy_kwh": 0.0034, "carbon_gco2e": 1.2, "compute_steps":
4},
  "signatures": {"agent":"sig:...", "operator":"sig:..." },
  "timestamps": {"start":"...", "end":"..." } }
```

Figure 5: Minimal Explanation Packet (MEP) operationalized: (A) lifecycle flow from design-time specifications to outcome generation; (B) release gate ensuring MEP completeness and integrity before deployment; (C) example MEP structure with required fields.

Auditors (H_a) need reproducible traces, evidence linking outcomes to decisions, and provenance suitable for third-party review [136]. **Regulators** (H_r) require documentation, incident records, and governance evidence aligned with applicable legal frameworks [35, 92]. **Affected third parties** (H_t) need aggregate transparency about system behavior and accessible recourse mechanisms, especially when impacts are distributed across populations.

Design principle. Stakeholder needs should determine what is disclosed, in what form, at what granularity, and with what access controls, all derived from a shared transparency substrate. A single execution trace can yield multiple views: user-facing summaries for H_u, developer-grade debugging traces for H_d, and auditor-grade signed provenance for H_a. **Table 4 (Panel B)** operationalizes these requirements.

3.7 Multi-Agent Extensions

As deployments evolve from single agents to multi-agent systems, transparency becomes more challenging because outcomes emerge from distributed decision-making across multiple agents. Multi-agent settings introduce four additional transparency considerations. **Role attribution** delineates role boundaries by specifying which agent made which decision, at the granularity required by the relevant assurance objective. **Inter-agent communication** requires traceable message exchanges (subject to privacy constraints) to debug coordination failures, identify influence patterns, and trace escalation paths. **Emergent behavior recognition** captures team-level dynamics such as consensus formation and conflict resolution that cannot be attributed to individual agents. **Provenance graphs** provide the structural foundation by adopting relations inspired by W3C PROV [136] to connect actions, artifacts, tools, and agents into a verifiable execution history that supports accountability and replay-based auditing.

Design principle. Attribution is often needed at multiple levels: *team-level* (collective outcome attributable to the ensemble), *agent-level* (decisions attributable to a specific agent), and *tool-level* (results attributable to external tools invoked by agents). For multi-agent deployments, MEPs emitted by individual agents can be linked through provenance edges to reconstruct team-level accountability for any given outcome.

3.8 Cross-Axis Relationships

Mechanisms $\times$ Objectives. Different mechanisms preferentially support different assurance goals. *Operational mechanisms* (M_o) and replay tend to be decisive for auditability (A_a) and often constrain faithfulness (A_f) by grounding explanations in verifiable traces [136, 108]. *Mechanistic analyses* (M_m) provide causal grounding for faithfulness and

Table 4: Operational requirements across lifecycle stages (WHEN) and stakeholders (WHO). **Panel A** specifies stage-by-object evidence needed for MEP completeness; **Panel B** defines stakeholder-facing views over the same transparency artifacts.

Panel A: Temporal Stage × Cognitive Object Requirements

Cognitive Object	Design-Time (W_d)	Process-Time (W_p)	Outcome-Time (W_o)
Intent ($\mathcal{G}$)	Goal specification templates; intent probe attachment points	Real-time goal verbalization; intent drift monitoring	MEP: decision summary with inferred intent
Beliefs ($\mathcal{B}$)	Belief schema definition; world model structure	Belief state inspection; update logging	MEP: evidence hashes; retrieval references
Plans ($\mathcal{P}$)	Plan representation format; alternative tracking schema	Plan execution monitoring; deviation alerts	MEP: plan graph with rejected alternatives
Memory/State ($\mathcal{S}$)	Memory/state structure design; retention policies	Read/write logging; contamination detection	MEP: memory access patterns; state snapshots
Tool I/O ($\mathcal{T}$)	Tool interface specs; error-handling protocols	Tool call traces; I/O logging with timestamps	MEP: tool traces (inputs/outputs/errors)
Policies (Π)	Policy encoding; guardrail definitions	Policy activation logging; threshold monitoring	MEP: policy/fairness deltas; triggered guardrails
Outcomes ($\mathcal{O}$)	Output format specs; effect attribution schema	Action logging; effect tracking	MEP: outcome record with signatures/timestamps

Panel B: Stakeholder × Transparency Artifact Requirements

Stakeholder	Primary Needs	Key Artifacts	Format Requirements	Priority Obj.
End Users (H_u)	Understandable explanations; clear status; recourse	Natural-language summaries; progress indicators; appeal mechanisms	Plain language; visual aids; interactive Q&A	A_u, A_e
Developers (H_d)	Debugging traces; error diagnosis; iterative refinement	Full execution traces; probe outputs; failure analytics	Structured logs; API access; IDE integration	A_f, A_r
Auditors (H_a)	Reproducible evidence; outcome–decision links; provenance verification	Signed MEPs; replay capability; hash verification	Cryptographic integrity; time-stamped; machine-readable	A_a, A_f, A_c
Regulators (H_r)	Compliance documentation; incident records; requirement alignment	Regulatory reports; risk assessments; incident logs	Standardized formats; mapped to requirements; retention-compliant	A_c, A_a
Affected Third Parties (H_t)	Aggregate transparency; accessible recourse; impact visibility	Population-level reports; fairness dashboards; appeal portals	Public-facing; accessible language; multi-format	A_e, A_u

can support robustness (A_r) by identifying brittle internal circuits [139]. *Social* (M_s) and *intrinsic* (M_i) mechanisms often dominate perceived usefulness (A_u), but should be calibrated against operational or mechanistic evidence when faithfulness is high-stakes [32, 79]. *Post-hoc tools* (M_p) can help diagnose behavior and surface disparities relevant to equity (A_e), but may be unstable under perturbations [113, 76, 126].

Objects × Stages. The same cognitive object may require different fidelity at different temporal stages. Design-time (W_d) policies may be summarized at a high level, process-time (W_p) policy activations must be logged with full detail, and outcome-time (W_o) policy deltas should be included in the MEP for audit purposes.

Stakeholders × Disclosure. Stakeholder requirements constrain what to reveal and how. User interfaces (H_u) prioritize clarity and contestability, while auditor interfaces (H_a) prioritize verifiability and replay. Privacy and security considerations may require selective disclosure, especially for memory/state ($\mathcal{S}$) and tool I/O ($\mathcal{T}$).

These interactions can be operationalized through systematic mapping. **Table 5** consolidates the relationships.

3.9 Example: Tool-Using LLM Agent

To make the taxonomy concrete, we present a tool-using LLM agent that answers a research question by planning, searching external databases, retrieving documents, and synthesizing a response [155]. **Fig. 6** summarizes the execution flow and the outcome-time transparency substrate.

Table 5: Interpretability–Explainability–Evaluation mapping to the five-axis taxonomy.

Agent Component	Cognitive Object (WHAT)	Interpretability Method	Explainability Method	Assurance Object. (WHY)	Evaluation Signals	Governance Mapping
Percep-tion	Intent ($\mathcal{G}$), Beliefs ($\mathcal{B}$)	Saliency maps [125], Grad-CAM [118], Concept activation [102], Attention visual. [1]	Goal verbalization [98], Retrieval attri-bution [2], Source influence scoring [103]	A_f, A_u	Localization accuracy [32], Plausibility scores [52], Concept alignment [60]	EU AI Act Art. 13, NIST MAP 1.5
Reason-ing / Plan-ning	Plans ($\mathcal{P}$), Policies (Π)	Linear probes [4], Structural probes [46], Circuit analysis [139, 45], ACDC [26], Inference-time probes [67]	Chain-of-Thought [146], ReAct traces [155], Plan graphs [140], Counterfactual expl. [64], Layered CoT [137]	A_f, A_c, A_r	Predictability [110], Selectivity [52], Faithfulness [36], Completeness [36], Minimality [36]	EU AI Act Art. 12, 14, NIST MEA-SURE 2.3, ISO 6.1.2
Tool Interaction	Tool I/O ($\mathcal{T}$)	AgentSHAP [47], Tool tracing [155], Retrieval attribution [16], Causal intervention [17]	Tool attribution [116], Tool traces (I/O, er-rors) [155], Evidence references [36]	A_f, A_a	Faithfulness [52], Completeness [36], Relevance [30], Hash signatures [37]	EU AI Act Art. 12, NIST MANAGE 4.1, ISO 10.2
Memory / State	Memory / State ($\mathcal{S}$)	Memory probes [4], Retrieval traces [38], Belief state inspection [162], Memory pertur-bation [83]	Memory attribution [25], State provenance [11], Belief update logs [43]	A_f, A_e, A_a	Completeness [36], Traceability [11], Parity delta [108]	NIST GOVERN 1.2, ISO 8.4
Cross-Layer / Full Trajectory	Outcomes ($\mathcal{O}$), All objects	Provenance graphs [11], Attention rollout [1], LRP / DTD [9], Sparse autoencoders [29], Activation patching [84]	Traces generation [11], Interactive dialogue [106], Policy logs [11], Audit reports [32]	All: A_f, A_u, A_c, A_r, A_e, A_a	Plan–trace agreement [67], Stability [111], Reproducibility [100], Retention compliance [34]	EU AI Act Art. 62, NIST full frame-work, ISO 9.3
Multi-Agent Coord.	Inter-agent messages, role attribution	Communication analysis [73], Role–rationale contracts [98], Attribution mappings [128], Provenance graphs [11]	Team-level logs [73], Coordination traces [155], Delegation logs [73]	A_a, A_c, A_e	Role coverage [49], Attribution accuracy [128], Coordination fidelity [49]	*Gap in current regulations*

Note. Assurance objectives: A_f = Faithfulness, A_u = Usefulness, A_c = Compliance, A_r = Robustness, A_e = Equity, A_a = Auditability.

As shown in **Fig. 6**, the top row captures the agent's decision trajectory (query → planning → tool use → synthesis → output). Dashed links indicate which intermediate artifacts are recorded into a Minimal Explanation Packet (MEP). This instantiates the cognitive objects dimension (intent $\mathcal{G}$, plans $\mathcal{P}$, tool I/O $\mathcal{T}$, policy activations Π, outcomes $\mathcal{O}$) and the temporal dimension (outcome-time packetization), while enabling stakeholder-specific views that improve faithfulness (A_f) and auditability (A_a).

Formally, we view an execution as a multi-step trajectory. Let $s_t \in \mathcal{S}$ denote the agent state at step t (including working memory and retrieved context), and let a_t denote the chosen action (e.g., a tool call or a synthesis step). After executing a_t, the agent observes o_t and updates its state:

$$a_t \sim \pi(\cdot \mid s_t), \qquad s_{t+1} \leftarrow \mathrm{Update}(s_t, a_t, o_t). \tag{1}$$

The surfaced cognitive objects include the inferred research intent $\mathcal{G}$ (the user's information need); the plan $\mathcal{P}$ (search strategy, retrieval sequence, synthesis approach); tool I/O $\mathcal{T}$ (queries sent, retrieved results, failures, and fallbacks); relevant policies Π (e.g., source-quality filters, citation requirements); and the final outcome $\mathcal{O}$ (a synthesized answer with citations). Transparency is achieved using three mechanism families: (i) intrinsic (M_i) structured reasoning via a compact plan graph rather than free-form verbosity; (ii) operational (M_o) signed logs of tool calls with timestamps and replay capability [136, 108]; and (iii) social (M_s) interaction that lets users request clarification, alternative sources, or higher-level summaries.

At outcome-time, the agent emits an MEP as a compact record of key artifacts:

$$\mathrm{MEP} \;=\; \langle \mathcal{G}, \mathcal{P}, \mathcal{T}, \Pi, \mathcal{O} \rangle. \tag{2}$$

Assurance evaluation checks faithfulness (A_f) by verifying that cited sources match retrieved documents and that reasoning aligns with execution traces. For example, let $\mathcal{D}_{\text{ret}}$ be retrieved evidence identifiers (recorded in tool traces) and $\mathcal{D}_{\text{cite}}$ those cited in the output. A simple consistency score is

$$A_f \;=\; \frac{|\mathcal{D}_{\text{cite}} \cap \mathcal{D}_{\text{ret}}|}{\max\{1, |\mathcal{D}_{\text{cite}}|\}}. \tag{3}$$

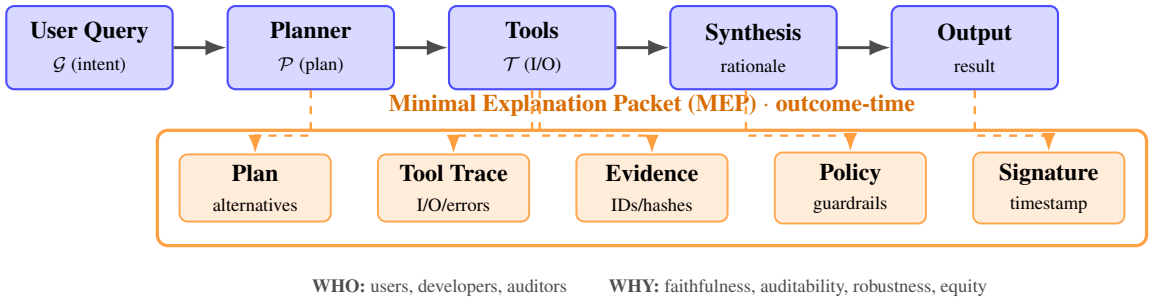


Figure 6: Tool-using agent under X-AXIOM: execution flow (plan → tool use → synthesis) and transparency substrate via an outcome-time Minimal Explanation Packet (MEP). Dashed arrows indicate provenance for replay, verification, and stakeholder-specific views.

which equals 1 when all citations are supported by retrieved documents. We assess usefulness (A_u) via task completion metrics and user feedback; robustness (A_r) via stability under query paraphrases; and auditability (A_a) via replay capability, signature verification, and retention compliance.

At outcome-time (W_o), the MEP contains (i) a plan summary of the search → retrieve → synthesize flow, (ii) tool traces (query strings, result counts, selected documents), (iii) evidence identifiers and hashes for retrieved documents, (iv) triggered policy filters, and (v) a cryptographic signature with timestamp [136, 108].
This example illustrates how the five dimensions work together: cognitive objects define *what* is exposed, assurance objectives define *why*, mechanisms and stages define *how* and *when*, and stakeholder needs shape *who* receives which view over a shared transparency substrate.

Operationalizing the MEP The MEP is not merely a conceptual artifact but a concrete data structure that can be validated at runtime. **Fig. 5** illustrates how the MEP integrates into the agentic lifecycle: design-time (W_d) specifications define required fields and logging policies; process-time (W_p) traces accumulate evidence across plan execution, tool calls, and memory/state updates; and a validation gate checks completeness, integrity (hash matching), plan–trace agreement, and fairness bounds before releasing the decision. This gating mechanism ensures that every released outcome is backed by verifiable, auditable evidence which is a requirement increasingly mandated by governance frameworks (**Section 2**). Next, we discuss interpretability and explainability through the lens of this taxonomy (across the Agentic AI lifecycle).

Having defined the five axes of agentic transparency in Section 3, we now instantiate them through two complementary lenses that recur across the literature: interpretability and explainability.

4 Interpretability in Agentic AI (Design- and Process-Time)

This section addresses how to make agent internals inspectable before and during execution. We cover two lifecycle stages: design-time preparation (instrumenting systems to enable inspection) and process-time monitoring (capturing evidence during execution). **Fig. 7** focuses on process-time, showing how evidence types (E1–E4) attach to each stage of the agent loop; design-time integration is addressed in Section 4.4.

We connect cognitive objects (intent $\mathcal{G}$, beliefs $\mathcal{B}$, plans $\mathcal{P}$, memory/state $\mathcal{S}$, tool interactions $\mathcal{T}$, and policies Π) to concrete evidence so developers (H_d) and auditors (H_a) can trace decisions and verify trustworthiness. Interpretability in agents is an end-to-end tracing problem. Evidence must be logged at perception, planning, tool I/O, and action layers so later audits do not depend on post-hoc narratives. We represent an execution as a length-T trace

4.1 Classical Interpretability: Foundations and Evolution

Classical interpretability developed across several waves. Early work in the 1990s emphasized rule extraction and model mimicry, distilling trained networks into symbolic rules or interpretable surrogates [131]. The CNN era popularized **feature attribution** methods, including deconvolutional networks [161], gradient saliency [125], and Grad-CAM [119], to visualize input regions that influence predictions.

In parallel, **model-agnostic** local explanations and additive feature attribution became widely used for structured data, including local surrogate models and Shapley-value-based attribution [113, 76]. **Counterfactual explanations** and

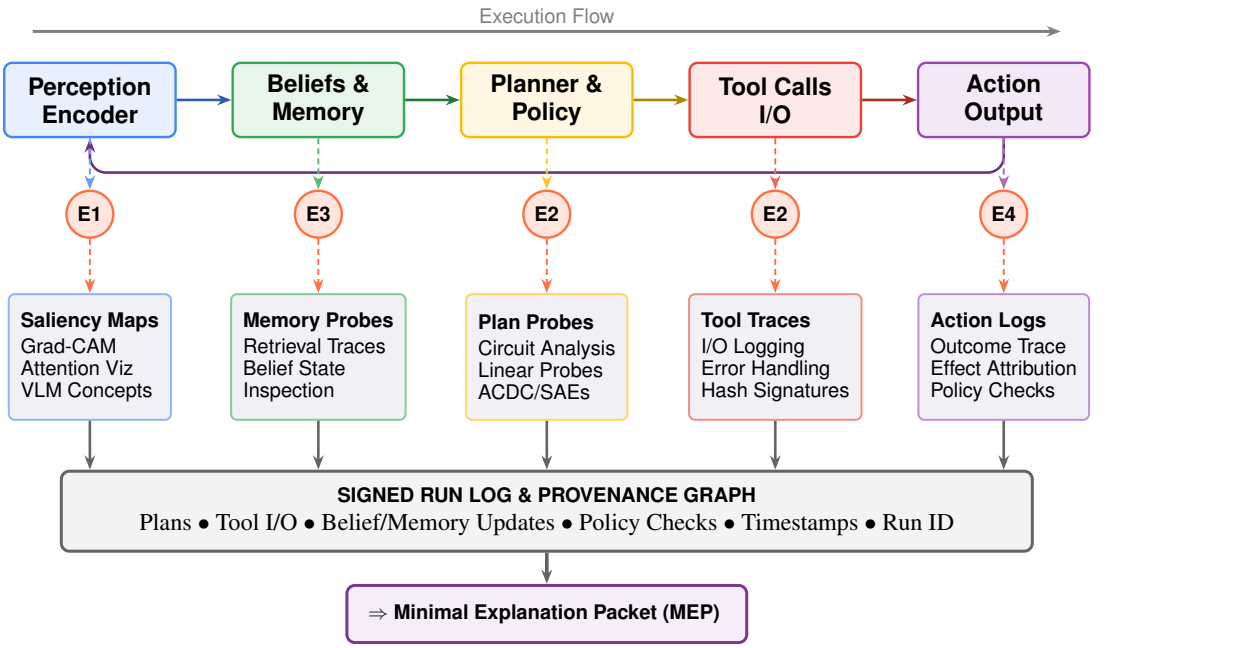


Figure 7: Where interpretability connects to the agent loop. Arrows indicate logged evidence: E1 retrieval, E2 tool I/O, E3 memory/belief updates, E4 actions. These traces support causal probes and feed into the Minimal Explanation Packet (MEP).

recourse reframed interpretability in terms of minimal changes required to alter outcomes [64]. **Concept-based** methods further bridged low-level features and human-meaningful abstractions [60].

With Transformers, **attention visualization** provided an initial lens, though attention weights are not a reliable importance measure on their own [53]. Subsequent methods, such as attention rollout and relevance propagation, aimed to trace information flow across layers [1, 23]. From 2020 onward, **mechanistic interpretability** increasingly focused on reverse-engineering learned algorithms via circuit analysis and sparse feature decompositions [139, 29]. These developments motivate trajectory-level inspection for agents, where plans, memory, and tool interactions shape outcomes over time. A timeline is shown in **Fig.** 8, which highlights the shift from correlation-based explanations (saliency and attention) toward causally grounded approaches (circuits and activation patching). This motivates why agent trajectories need more than single-step explanations.

4.2 Interpretability Methods for Agentic AI Components

Having established the foundations of interpretability in **Section 4.1**, we now survey methods organized by the agent component they target. This structure aligns with the cognitive objects dimension of our taxonomy (Section 3), covering perception, reasoning/planning, tool use, memory, and cross-component tracing.

Perception-Layer Interpretability Traditional XAI methods such as **saliency maps**[125], **Grad-CAM**[119], and **deconvolutional visualization**[161] can be applied directly to the perception encoders of Agentic AI systems. For vision-language models, concept-based methods can expose latent semantic structure in joint embedding spaces[122], while attention-based tools help inspect cross-modal grounding [19]. These methods diagnose perception failures such as ignored visual evidence or spurious correlations.

Reasoning and Planning Interpretability Methods developed for LLM interpretability extend to agent reasoning and planning. **Probing** methods test whether internal representations encode goals, subgoals, or expected action outcomes [4, 46]. Evidence also suggests models may represent planned continuations prior to token emission [67]. **Mechanistic** approaches (circuits and sparse features) aim to isolate causally relevant substructures and representations that implement planning-relevant behaviors [139, 26, 29]. Adapting these methods to multi-step agentic trajectories remains an open challenge.

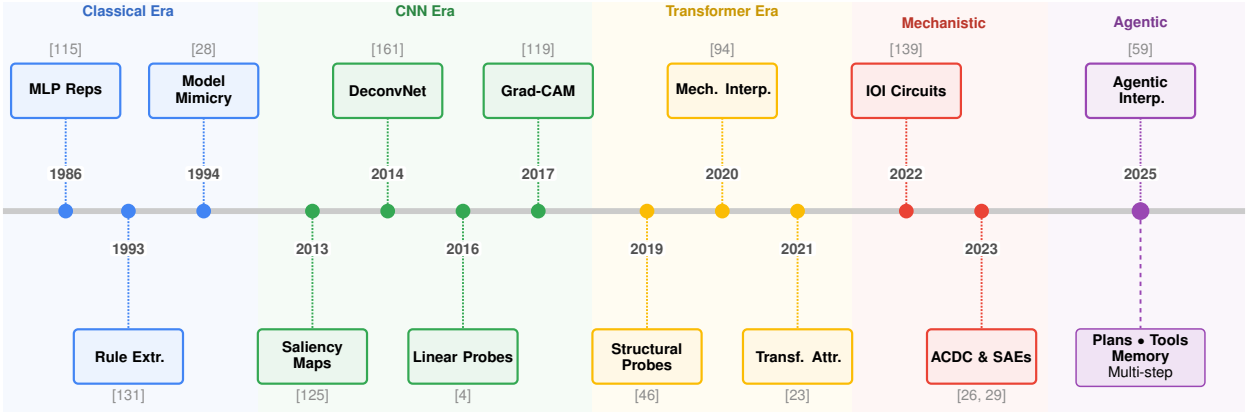


Figure 8: Timeline of key milestones in AI interpretability.

Tool and Memory Interpretability For tool-using agents, attribution methods quantify how tool outputs influence final responses (e.g., tool-level Shapley scores) [47]. Let $\mathcal{O}(\mathcal{T})$ denote the outcome produced given tool traces $\mathcal{T}$, and let $\mathcal{T} \setminus t$ denote the trace with a specific tool output t removed or replaced. A simple intervention-based influence score is

$$\Delta_t \;=\; d(\mathcal{O}(\mathcal{T}),\, \mathcal{O}(\mathcal{T} \setminus t))\,, \tag{4}$$

where $d(\cdot, \cdot)$ is a task-appropriate difference measure (e.g., answer change, factual consistency, or reward). Causal interventions provide stronger evidence by removing or perturbing tool outputs and measuring outcome changes [117, 38]. Memory interpretability inspects read/write operations and their downstream influence [98], including perturbation tests and long-context utilization analyses [72].

Cross-Component Tracing Full interpretability requires tracing information flow across perception, reasoning, tool use, and memory throughout a trajectory. Provenance graphs (e.g., W3C PROV-style) represent states, actions, and artifacts as nodes with derivation and responsibility edges. This enables forward and backward tracing for systematic audit [136]. In practice, trajectory-level tracing asks whether intermediate artifacts (plans, retrieved evidence, tool outputs) are causally responsible for the final outcome, not merely correlated with it.

	Perception	Reasoning / Planning	Tool Use	Memory	Multi-Agent
Feature Attribution	●●●	●	–	–	–
Concept Probes	●●	●●	–	●	–
Attention Analysis	●●	●●	–	–	–
Mechanistic / Circuits	●	●●	–	–	–
Tool Tracing	–	–	●●●	–	–
Memory Probes	–	–	–	●●	–
Provenance Graphs	●	●	●●	●	●●
Causal Intervention	●	●●	●●	●●	●

Legend: ●●● Strong ●● Moderate ● Limited – Gap

Figure 9: Interpretability method coverage by agent component. The matrix reveals that current methods provide strong coverage for perception but significant gaps remain for tool use, memory, and especially multi-agent interpretability.

Finally, Agentic AI interpretability also includes trajectory-level causal tracing, multi-agent interaction attribution, and runtime monitoring for tool correctness and memory integrity. We summarize these methods in **Table 6**. However, methods coverage remains limited, as shown in **Fig. 9**, which maps major interpretability families (attribution, probes, circuits, provenance, causal interventions) to agent components and highlights that tool use, memory, and especially multi-agent interpretability remain key gaps.

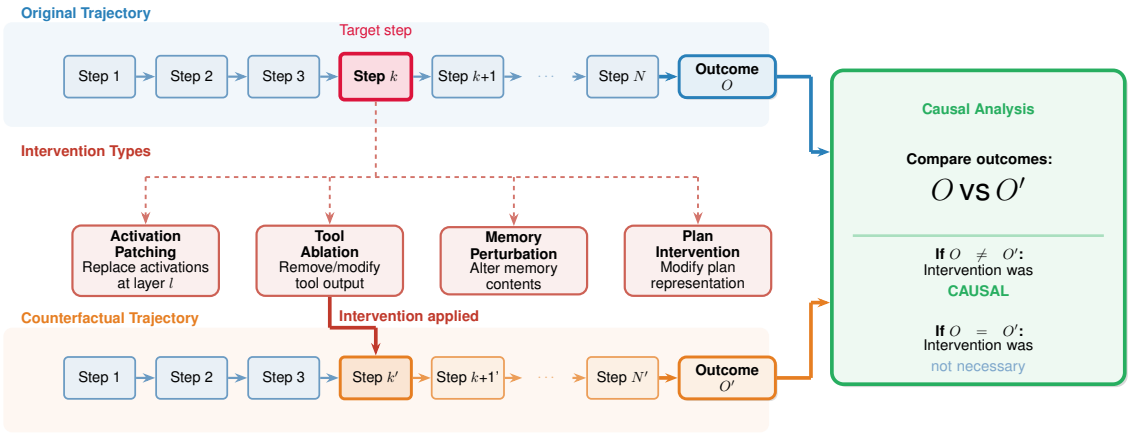


Figure 10: Agent-loop causal probes. Apply an intervention at step k (activation patching, tool ablation, memory perturbation, or plan intervention) to generate a counterfactual trajectory and outcome O'. If $O \neq O'$, the intervened component is causally relevant (counterfactual dependence).

4.3 Agent-Loop Causal Probes

Agentic interpretability must explain behavior over trajectories, where each action can change later observations. **Fig. 10** summarizes the causal-probing template: intervene at step k (e.g., activations, tool outputs, memory contents, or plan representations), replay the trajectory, and compare an outcome of interest (e.g., task success, final answer, or action sequence) under O vs. O'. We instantiate this template for each agent component.

Intervention-based analysis. We consider five intervention families. **Activation patching** swaps internal states to test causal necessity [84]. **Tool ablation** removes or perturbs tool I/O to isolate tool-driven effects. **Memory perturbation** edits or deletes retrieved/stored memories to test whether specific items drive decisions. **Plan intervention** modifies plan representations $\mathcal{P}$ to test whether changes propagate to actions. **Predictive probing** exploits the agent's internal world model (if present) by comparing the agent's anticipated outcomes against actual execution results; discrepancies reveal where predictive representations diverge from realized trajectories and can expose planning failures rooted in faulty forward projections. These tests can be run online or via offline trajectory replay.

Dialogue-guided probing. Beyond automated interventions, interactive approaches query agents through multi-turn dialogue to elicit reasoning [59]. This includes asking "why" questions about specific decisions, posing counterfactual scenarios ("what would you have done if..."), and iteratively refining hypotheses about agent behavior through follow-up queries. However, these self-reports should be treated as hypotheses rather than ground truth, so rationales may be unfaithful or post-hoc rationalizations [133]. Effective dialogue-guided probing therefore combines conversational exploration with intervention-based validation to distinguish genuine reasoning from plausible-sounding confabulation.

4.4 Lifecycle Integration: Design-Time and Process-Time

The interpretability methods described above, including perception probes, mechanistic analysis, tool attribution, and causal interventions, depend on infrastructure established before deployment and maintained during execution. Without deliberate instrumentation, key cognitive objects remain unobservable and causal probes cannot be applied. We therefore distinguish two preparatory stages.

Design-time preparation (W_d). At design time, systems are instrumented to make later analysis possible: (i) **object selection** specifies which cognitive objects are exposed (e.g., intent $\mathcal{G}$, plans $\mathcal{P}$, tool calls $\mathcal{T}$, outcomes $\mathcal{O}$); (ii) **schema design** defines the fields and formats for each object; (iii) **provenance infrastructure** provides storage, linking, integrity guarantees, and access control; and (iv) **probe attachment** identifies where to extract intermediate signals (e.g., activations) for inspection.

Process-time monitoring (W_p). At runtime, oversight is supported by **trace logging** of I/O and intermediate states with timestamps and run IDs [11]; **log integrity** protections (e.g., cryptographic mechanisms) [37]; **anomaly detection** for plan–trace divergence or unexpected failures [130]; and **live probing** over executed trajectories to surface misalignment-relevant features. Together, W_d and W_p ensure that process-time traces can later be packaged into the Minimal Explanation Packet (MEP) for debugging, audit, and compliance. Outcome-time explainability, which translates these traces into stakeholder-facing artifacts, is addressed in Section 5. End-of-lifecycle interpretability includes trace

archiving, retention policies, and decommissioning documentation, ensuring that post-deployment obligations are treated as part of operational transparency.

Table 6: Interpretability techniques mapped to agent components (design- and process-time).

Method	Modality	Focus	Metrics (tags)	Agent layer
Deconvolutional nets [161]	Image	Map feature activations back to pixels via unpooling / transpose conv	localization, plausibility, runtime	Perception
Gradient saliency [125]	Image	Input sensitivity heatmaps via gradients	localization, plausibility, robustness	Perception
Grad-CAM [119]	Image	Class-discriminative localization via conv feature gradients	localization, plausibility, robustness	Perception
Concept activation (VLM) [122]	Text/Image	Latent concept discovery in shared embedding spaces	concept-alignment, plausibility	Perception
Attention flow / rollout [1]	Text/Image	Aggregate token influence paths across layers	plausibility, correlation	Cross-layer
LRP/DTD for Transformers [89, 23]	Agnostic	Propagate relevance through attention/residual routes	faithfulness, localization	Cross-layer
TCAV [60]	Agnostic	Sensitivity to human-defined concepts	concept-influence, robustness	Cross-layer
Sparse autoencoders [29]	Agnostic	Decompose activations into sparse, interpretable features	feature-quality, sparsity, faithfulness	Cross-layer
Activation patching [84]	Agnostic	Swap internal states to test causal necessity	faithfulness, causal-necessity	Cross-layer
Provenance graphs [136]	Agnostic	Track derivation and responsibility across components	completeness, traceability	Cross-layer
Dialogue-guided interpretability [59]	Agnostic	Model-human dialogue to hypothesize mechanisms (validate with probes)	usefulness, plausibility	Cross-layer
Linear probes [4]	Agnostic	Decode target information from intermediate layers	predictability, selectivity	Reasoning / Planning
Structural probes [46]	Text	Geometric tests for syntactic structure	correlation, predictability	Reasoning / Planning
Circuit analysis (IOI) [139]	Agnostic	Identify sparse subgraphs with causal validation	faithfulness, completeness, minimality	Reasoning / Planning
ACDC [26]	Agnostic	Automated discovery of causally relevant subgraphs	faithfulness, completeness, minimality	Reasoning / Planning
Inference-time probes [67]	Text	Decode planning tokens / future action predictions during generation	predictability, faithfulness	Reasoning / Planning
Tool attribution [47]	Text+Tools	Quantify which tool calls influenced outputs	faithfulness, completeness	Tool Interaction
Tool tracing [116]	Text+Tools	Logs, tool calls and outputs	faithfulness, completeness	Tool Interaction
Retrieval attribution [38]	Text+Retrieval	Identify which retrieved passages shaped generations	faithfulness, relevance	Tool Interaction

Note. IOI = indirect object identification. Metric tags: *faithfulness*, *causal-necessity*, *completeness*, *minimality*, *robustness*, *localization*, *plausibility*, *predictability*, *selectivity*.

5 Explainability in Agentic AI (Process- and Outcome-Time)

Section 4 focused on inspecting agent internals for developers (H_d) and auditors (H_a); here, we translate that evidence into explanations for end users (H_u) and governance needs, including compliance (A_c).

5.1 From Interpretability to Explanation

Interpretability methods (e.g., probes, interventions, provenance) often produce technical evidence that is difficult for non-experts to consume. Explainability translates this evidence into narratives, visualizations, and interactive interfaces. This translation must balance three tensions: (i) faithfulness (A_f) versus accessibility, since faithful traces may be too complex; (ii) completeness versus cognitive load, since full provenance can overwhelm; and (iii) standardization versus context sensitivity, since regulator-ready artifacts differ from user-facing explanations. **Fig. 11** summarizes where evidence is produced in the agent lifecycle and where it is transformed into stakeholder-facing artifacts, motivating the need for structured packaging (e.g., MEPs).

The cognitive objects and assurance objectives defined earlier in **Section 3** guide what evidence is surfaced and how it is packaged. Stakeholders require different views: end users (H_u) prioritize intent and outcome summaries; auditors (H_a) require verifiable traces with timestamps and integrity guarantees; and regulators (H_r) require standardized documentation mapped to specific requirements. **Fig. 12** maps these objects to explanation modalities and primary audiences, highlighting that the same underlying evidence must be rendered differently across stakeholders.

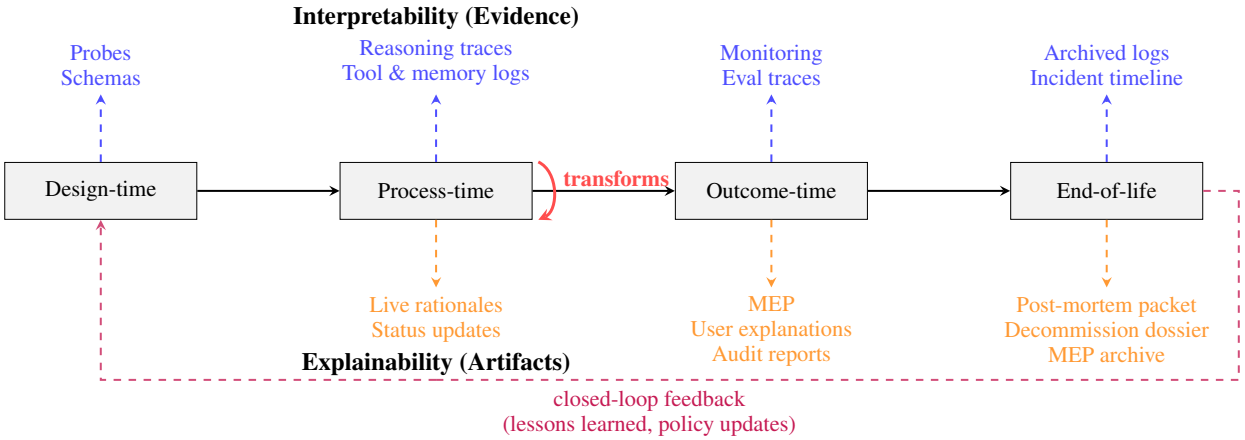


Figure 11: **From interpretability evidence to explainability artifacts across the agent lifecycle.** Interpretability yields technical evidence (e.g., probes, traces, logs) across stages, while explainability transforms it into stakeholder-facing artifacts (e.g., MEPs, explanations, audits, post-mortems). End-of-life artifacts support accountability and feedback into design.

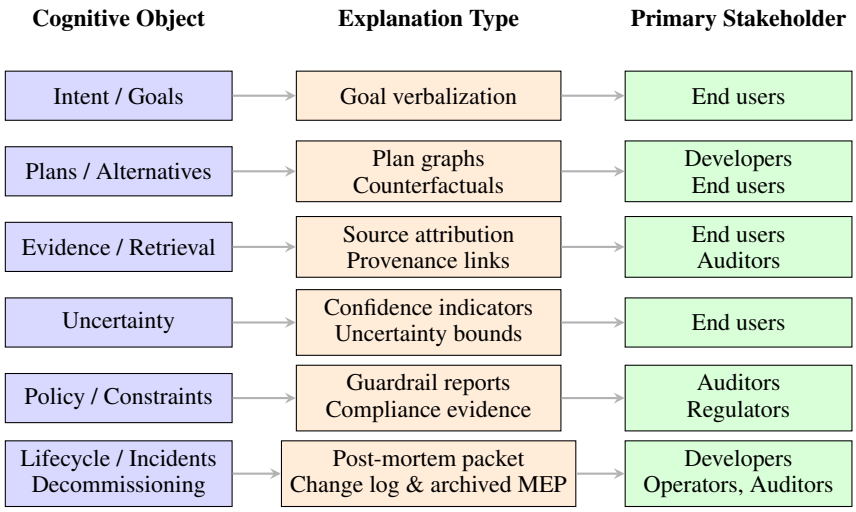


Figure 12: Mapping cognitive objects to explanation types and primary stakeholders, extended to include lifecycle/incident and end-of-life artifacts for auditability and closed-loop improvement.

5.2 Explanation Methods for Agentic Systems

Agentic AI systems generate rich trajectory evidence (§3.5), but this raw data requires transformation into stakeholder-appropriate explanations. We organize explanation methods by their approach to bridging this gap.

Layered and contrastive explanations. Layered CoT [137] introduces checkpoints that let users verify intermediate conclusions at a chosen level of detail. Contrastive explanations answer "why X rather than Y?", which is particularly useful for plans ($\mathcal{P}$) and rejected alternatives.

Counterfactual explanations. For agents, counterfactuals can operate on inputs (a different query), evidence (a different retrieval set), or plans (a different action sequence) [64]. Simulatability evaluation tests whether explanations enable users to anticipate counterfactual outcomes.

Interactive and contestable explanations. Static explanations may not address user needs. Reflexion [121] supports self-critique and revision, while contestability interfaces enable users to challenge decisions and receive responsive explanations aligned with human oversight expectations [35]. Multi-turn dialogue can help form hypotheses, but should be validated against trace evidence where possible [61].

Compliance-oriented artifacts. Governance-facing explanations differ from user explanations in evidentiary require-

Table 7: Five-axis explainability view: (A) methods mapped to cognitive objects (goals, plans, tools/evidence, uncertainty, policy), assurance objectives (faithfulness, usefulness, compliance, robustness, auditability), and stakeholders; (B) governance requirements crosswalk (EU AI Act, NIST RMF, ISO 42001).

Panel A: Explainability methods

Method	Description	Objects					Objectives					Stakeholders
		G	P	T	U	π	F	U	C	R	A	
Chain-of-thought	Step-by-step reasoning	●	●	○	○	○	◖	●	○	○	◖	Users, Dev
ReAct traces	Interleaved reasoning/action	●	●	●	○	○	●	●	◖	○	●	Users, Dev
Retrieval attribution	Source influence scoring	○	○	●	◖	○	●	◖	○	◖	●	Users, Aud
Tool attribution	Tool contribution scores	○	○	●	○	○	●	◖	○	◖	●	Developers
Counterfactuals	Minimal changes for different output	◖	●	◖	○	○	●	●	○	◖	○	Users
Policy logs	Guardrail activation records	○	○	○	○	●	●	○	●	○	●	Aud, Reg
Interactive dialogue	Collaborative refinement	●	●	●	◖	○	◖	●	○	○	○	Users
MEP	Structured outcome packet	●	●	●	◖	●	●	◖	●	◖	●	All

● primary; ◖ partial; ○ not addressed. Dev=Developers; Aud=Auditors; Reg=Regulators.

Panel B: Governance requirements crosswalk

Requirement	EU AI Act	NIST RMF	ISO 42001
Decision logging	Art. 12	MEASURE 2.3	6.1.2
Human oversight	Art. 14	GOVERN 1.2	8.4
Explainability	Art. 13	MAP 1.5	9.3
Incident records	Art. 62	MANAGE 4.1	10.2

ments: they must be reproducible, complete, and verifiable. Rather than treating governance mapping as a separate artifact, we merge it into the consolidated view (**Table 7, Panel B**) .

5.3 Explanation Quality and Robustness

Explanations themselves must satisfy assurance objectives. For faithfulness (A_f), self-generated rationales should be treated as hypotheses requiring validation against operational traces [133]. For robustness (A_r), explanations should remain stable under minor perturbations; post-hoc methods like LIME and SHAP can be unstable [126]. For equity (A_e), explanation quality should not vary systematically across demographic groups, and accessibility constraints should be treated as first-class design requirements. **Table 7** consolidates explainability methods using five-axis coordinates (WHAT/WHY/HOW/WHEN/WHO): each row indicates which cognitive objects it addresses, which assurance objectives it supports, and which stakeholders it targets. **Panel B** summarizes common governance requirements across major frameworks. Having established the *what* and *how* of agentic transparency through design-time specifications and outcome-time artifacts, we now address *whether* these mechanisms achieve their intended objectives. Evaluation frameworks must assess not only interpretability and explanation quality but also their operational effectiveness across diverse stakeholder needs. With methods in place for producing and communicating transparency evidence, the remaining challenge is to measure whether they work. Next, Section 6 reviews evaluations in Agentic AI.

6 Evaluation Protocols for Transparent Agents

Evaluating transparency in Agentic AI systems requires going beyond task accuracy to test whether the system's reasoning, evidence, and safeguards behave as claimed. Building on the evidence collection mechanisms in Section 4 and the packaging methods in Section 5, this section summarizes evaluation protocols across the agent lifecycle (design-

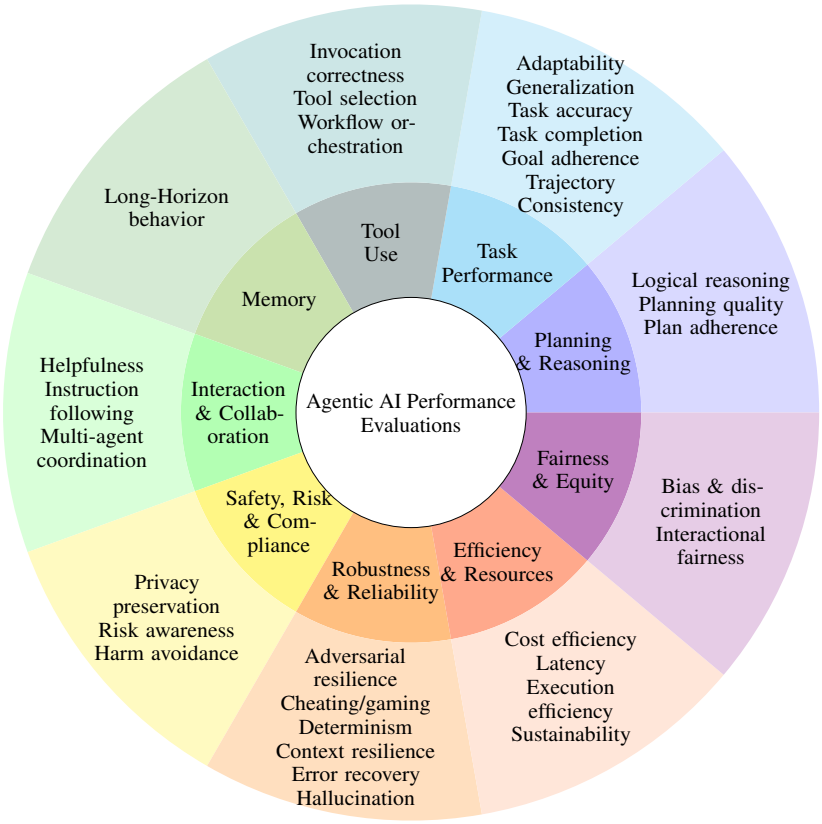


Figure 13: Agent performance evaluation landscape. Inner ring: nine core evaluation areas derived from the agentic AI literature. Outer ring: representative metrics for each area. Colors distinguish categories; see Table 8 for benchmark mappings.

time, process-time, and outcome-time). We operationalize transparency using six assurance objectives: *faithfulness, usefulness, compliance, robustness, equity* and *auditability*.

6.1 Design-time Readiness

Design-time evaluation ensures that the system is prepared to support transparency before deployment. This includes verifying that all relevant cognitive objects, such as intent, plans, memory and belief updates, tool interactions, and outcomes, can be captured through the instrumentation defined for the agent loop, consistent with established descriptions of autonomous agent workflows [101, 20]. The MEP schema is examined for completeness, internal consistency, and its ability to store all required fields [108].

To support reproducibility and auditability, we validate the underlying integrity infrastructure, including hashing, digital signatures, timestamping, and retention settings, following best practices in provenance and compliance [136, 51]. These checks are summarized in a readiness score representing the proportion of required fields that are (i) populated and (ii) cryptographically protected. A high readiness score ensures that subsequent evaluations rest on verifiable evidence rather than ad-hoc or incomplete logging. **Fig. 13** provides a compact view of the readiness dimensions and helps diagnose which evidence types are missing at design time.

6.2 Process- and Outcome-Time Evaluation

We evaluate transparency at runtime (process-time) and after decisions are produced (outcome-time) using six assurance objectives. These objectives act as evaluation criteria that determine what evidence is required (e.g., traces, tool logs, memory updates), how it should be summarized for stakeholders, and which failure modes must be detectable.

1. **Faithfulness** examines how closely the reported explanation is supported by the realized execution trace and by the system's causal dependencies. This follows established principles in interpretability and attribution

that emphasize alignment between explanations and true computational pathways [23]. We complement trace checks with causal probes, drawing on mechanistic interpretability where targeted interventions on activations, components, or tool calls test whether highlighted elements play a causal role [29].

2. **Usefulness** is evaluated by comparing stakeholder outcomes with and without explanations. Prior work in user-centered XAI shows that well-designed rationales can improve task completion, reduce cognitive burden, and assist in debugging [96]. We track whether explanations help users understand behavior or identify failures.
3. **Compliance** ensures that transparency artifacts contain evidence required by policies, safeguards, or standards. This covers the presence and correctness of mandated fields in the MEP, links to policy checks, and consistency with organizational or regulatory constraints (regulations discussed in **Section 2.4**).
4. **Robustness** assesses whether small, non-substantive input variations produce markedly different outcomes or explanations. Earlier studies on perturbation-based explanation methods demonstrate the importance of stability under benign changes [143]. Stable artifacts are indicative of robustness to surface-level noise.
5. **Equity** examines consistency of decisions and explanations across demographic or otherwise protected groups. Following fairness end audit explainability recommendations, it highlights parity in outcomes, interpretability quality, and user experience. recording disparities in the fairness fields of the MEP.
6. **Auditability** measures whether explanatory claims are grounded in verifiable log entries. This includes checking that traces can be replayed and that signatures remain valid, consistent with established documentation and dataset governance practices [86, 39]. We also measure runtime and token overhead to ensure that transparency remains practical for deployment.

Process-Time Evaluation Process-time evaluation assesses whether transparency evidence is captured *as the agent executes*. The goal is to detect and diagnose failures during planning, tool use, and state updates, and to ensure that resulting traces are sufficiently complete and reliable for oversight. In process-time evaluation, we measure : (i) trace completeness across key cognitive objects (plans, tool I/O, memory writes/reads, policy activations); (ii) trace integrity (timestamps, run IDs, tamper-evidence where applicable); (iii) trace–execution agreement (do logged actions/tool calls match what was executed); (iv) monitoring sensitivity to common failure modes (tool errors, retrieval drift, memory contamination, plan divergence). At process-time, faithfulness and auditability are primary (trace fidelity and verifiability). Robustness and equity are monitored via stability checks and disaggregated alerts, while compliance is supported by ensuring required logging and oversight artifacts exist.

Outcome-Time Evaluation Outcome-time evaluation assesses whether each decision is accompanied by a *stakeholder-appropriate* transparency artifact and whether that artifact supports accountability after the fact. The focal point is the quality of the emitted explanation package (e.g., an MEP) and its ability to support contestation, audit, and governance reporting. In outcome-time evaluation, we measure : (i) explanation sufficiency for the intended stakeholder (users, developers, auditors, regulators); (ii) evidence grounding (claims link to logged traces, retrieval references, and tool outputs); (iii) reproducibility/replayability (can the outcome be reconstructed from the recorded evidence); (iv) stability under reasonable perturbations (robustness of explanations and attributions); (v) fairness and disparity indicators reported at appropriate granularity. Usefulness and compliance are most salient at outcome-time (actionable and governance-ready artifacts), while faithfulness and auditability constrain narrative explanations through trace grounding and verification. Equity and robustness are evaluated through disaggregated reporting and stress-testing of both outcomes and explanations.

6.3 Reporting and Reproducibility

To facilitate reproducibility, we report all evaluation metrics over multiple runs and present summary statistics with confidence intervals. This aligns with established expectations for transparent reporting in empirical research [95]. Along with the quantitative results, we publish configuration files, prompts, model parameters, and guardrail settings, following documentation conventions such as model cards and datasheets [86, 39].

At least one complete example, including the full trace and its MEP, is made available for replay. A concise summary table presents the agreement scores, stability results, usefulness measures, equity gaps, MEP completeness, and runtime overhead. These practices ensure that the evaluation protocol remains transparent, repeatable, and easy to compare across datasets and systems.

6.4 End-of-Life Evaluation

Although end-of-life obligations are reflected in outcome-time artifacts and retention policies, we distinguish this phase to make decommissioning checks explicit and reproducible. End-of-life evaluation addresses transparency requirements when an agentic system is deprecated, replaced, retrained, or decommissioned, ensuring accountability beyond active

deployment. We identify four core concerns. **Cumulative impact accounting** aggregates transparency evidence across the system's lifetime, including resource use, environmental footprint [58], fairness outcomes, and longitudinal failure patterns, supporting organizational learning and regulatory reporting. **Deprecation rationale** documents why the system is retired (e.g., performance degradation, policy non-compliance, or replacement) and how successor systems differ [92]. **Knowledge and state disposition** specifies how memory, learned parameters, and interaction histories are retained, transferred, anonymized, or deleted, with documented justification. **Audit trail preservation** ensures that Minimal Explanation Packets, execution traces, and compliance evidence remain accessible for post-hoc review or audit after decommissioning [39].

End-of-life transparency is especially important in high-stakes domains where decisions may be contested long after deployment. At this stage, *compliance* requires adherence to data retention and disposal regulations; *auditability* requires preserved, accessible evidence; *faithfulness* requires accurate reporting of cumulative impacts; *equity* requires retrospective analysis of demographic disparities; and *sustainability* requires full lifecycle environmental accounting [75].

6.5 Agent Performance Signals

Agent performance metrics characterize the behavioral properties of agentic systems: whether they succeed in tasks, follow plans, select tools appropriately, avoid unsafe actions, maintain consistent memory, and coordinate effectively in multi-agent settings. Although these metrics do not evaluate transparency directly, they indicate where transparency evidence matters most (e.g., planning, tool interaction, memory updates, and safety-relevant decisions).

We group agent evaluation into the following areas: task performance; planning & reasoning; tool use; memory and long-horizon behavior; interaction & collaboration; safety, risk & compliance; robustness & reliability; efficiency & resource use; and fairness & equity. **Table 8** organizes reported evaluation metrics from recent agent benchmarks under these categories. This reorganization does not treat these benchmarks as measures of transparency; rather, it shows how the behaviors they assess expose transparency-critical regions of the workflow. **Fig. 13** provides an index of which metric families appear most often across current benchmark suites.

Environmental Costs as a Transparency Concern The "cost" dimension of efficiency evaluation should be understood broadly to encompass environmental costs alongside financial and computational costs. While token usage, API expenses, and latency are routinely monitored, energy consumption, carbon emissions, and water usage remain largely invisible to developers and users despite their growing significance [75]. Agentic AI systems amplify these concerns: iterative reasoning, repeated model calls, tool invocations, and extended sessions can produce environmental footprints substantially larger than single-inference deployments. Recent benchmarking reveals that reasoning-intensive models can consume over 70 times the energy of efficient alternatives for equivalent prompts [54], and inference-phase emissions increasingly dominate the lifecycle footprint as deployment scales.

From a transparency perspective, environmental costs represent a category of outcome that stakeholders increasingly have legitimate interest in understanding. Regulators are beginning to require environmental disclosure for AI systems [35], and organizations face reputational and operational risks from undisclosed resource consumption. We therefore recommend that efficiency evaluation explicitly include environmental metrics, such as per-step energy consumption, session-level carbon emissions, and water usage where data center instrumentation permits, and that these metrics be logged in transparency artifacts such as the MEP. Tools such as CodeCarbon (`https://codecarbon.io/`) can support estimation at the application level. As measurement infrastructure matures, environmental transparency is likely to transition from optional to expected, and systems that instrument these costs early will be better positioned for compliance and stakeholder accountability.

Table 8: Agent performance evaluation landscape grouped by core performance areas.

Area	Evaluation Aspect	Reported Metrics	Measurement Target	Benchmarks & References
Planning & Reasoning	Logical Reasoning	Multi-step reasoning accuracy	Accuracy of multi-step or structured reasoning	DABstep [33]
	Planning Quality	Planning accuracy	Coherence and soundness of generated task plans	Multi-Plan [56], Agent GPA [55]
	Plan Adherence	Plan adherence score	Match between plan and executed behavior	Agent GPA [55]
Task Performance	Adaptability	Self-awareness, Belief updating	Ability to adjust behavior when context or information changes	MaGIC [151], Reflection-Bench [68]

Area	Evaluation Aspect	Reported Metrics	Measurement Target	Benchmarks & References
	Generalization	Out-of-distribution success rate	Ability to succeed on unseen or out-of-distribution tasks	TeamCraft [74]
	Goal Adherence	Goal adherence score, Goal drift score	Alignment of actions with the user's stated goal	Goal Drift [7]
	Task Accuracy	Accuracy, Precision, Exact match rate	Correctness of outputs relative to ground-truth answers	AssistantBench [158], CORE-bench [123]
	Task Completion	Completion rate, Pass rate, Success rate, Win rate	Success in fully completing the assigned task	AppWorld [132], MCP-Bench [144], Col-Bench [166]
	Trajectory Consistency	Logical consistency score	Logical consistency of actions over the task trajectory	Agent GPA [55]
Tool Use	Tool Invocation Correctness	Parameter Accuracy, Result Accuracy, Dialogue turn success rate, Schema Compliance Rate, Grounding accuracy	Correctness of API calls, parameters, schemas, and bindings	GTA [138], MCP-Bench [144], Agent GPA [55], Agent-Board [22]
	Tool Selection	Tool selection accuracy, Tool Name Validity Rate, Tool Appropriateness	Choosing the most appropriate tool for each reasoning step	GTA [138], Agent GPA [55]
	Tool Workflow Orchestration	Tool retrieval exact match, Task orchestration exact match	Ability to plan and execute multi-step or multi-tool workflows	MSC-Bench [44]
Memory	Long-Horizon Behavior	Memory Accuracy, Memory Recall, Memory Capacity, Memory Efficiency	Ability to store, retrieve, and use information across turns or long horizons	MemBench [129], LoCoMo [81], LTM Benchmark [21]
Interaction & Collaboration	Helpfulness	Helpfulness score	Human-perceived quality of responses	TrustAgent [48], ToolEmu [114]
	Instruction Following	Instruction following accuracy, Instruction following error, Grounding accuracy	Adherence to user instructions and constraints	HAL [57], Agent-Board [22], GTA [138]
	Multi-agent Coordination	Coordination score, Planning Score, Collaboration score, Workload balance	Ability to coordinate and collaborate with other agents	MultiAgentBench [168], MAgIC [151],
Safety, Risk & Compliance	Privacy Preservation	Privacy leakage rate	Avoidance of private or sensitive data leakage	AgentDAM [165], PrivacyLens [142]
	Risk Awareness	Safety judgment scores, Inter-annotator agreement	Ability to recognize unsafe or high-risk content and contexts	ToolEmu [114], R-Judge [160]
	Harm Avoidance	Unsafe-action rate, Safe-action rate, Safety score, Harm Score, Criminal traits activation rate	Avoidance of harmful, unsafe, or high-risk actions	OpenAgentSafety [135], PRISON [148]
Robustness & Reliability	Adversarial Resilience	Attack success rate, Net resilient performance, Refusal rate, Adversarial prompt robustness, Benign accuracy	Resistance to adversarial prompts or malicious manipulation	AgentHarm [6], AgentPoison [24], PromptBench [167]
	Cheating / Gaming	Cheating flag, Evaluator gaming	Exploiting evaluation loopholes or manipulating evaluation	HAL [57]
	Determinism & Dialogue Consistency	pass^k, Conversational consistency	Consistency across repeated runs and multi-turn dialogues	τ-Bench [153], TD-Eval [3]
	Context Resilience	Volatility Factor, Accuracy under distraction	Stability under distracted, truncated, or perturbed context	C^3-Bench [18], EnvDistraction [80]
	Error Recovery	Self-correction rate, Repair rate	Ability to detect and correct mistakes during execution	HAL [57]
	Hallucination	Backend Knowledge Consistency, Information grounding, Hallucination rate	Tendency to produce fabricated or ungrounded content	TD-Eval [3], MCP-Bench [144]
Efficiency & Resource Use	Cost / Resource Usage	Token cost, API cost	Resource efficiency	TheAgentCompany [150], HAL [57],
	Latency	Response time	Time required to produce responses or complete tasks	MobileAgentBench [141]
	Execution Efficiency	Dialogue turns, Tool-call count, Step efficiency	Efficiency during task execution	LiveMCPBench [88], Agent GPA [55],

Area	Evaluation Aspect	Reported Metrics	Measurement Target	Benchmarks & References
	Sustainability	Energy, Carbon emissions	Environmental footprint	WebAgentEnergy [65]
Fairness & Equity	Bias and Discrimination	Bias score, Performance gap	Parity of outcomes across groups	ImplicitBias [14], MAL-IBU [85]
	Interactional Fairness	Interpersonal fairness, Informational fairness	Fair treatment across users	Interactional Fairness [13]

The organization of performance metrics into these evaluation areas aligns with patterns observed across recent surveys on agent evaluation [36]. Prior work typically groups metrics around broad behavioral, capabilities, reliability, and safety dimensions; our structure provides a consolidated view grounded in the current benchmark literature while remaining compatible with existing taxonomies. Neverthless, the agentic evaluation literature is still developing, and coverage across different behavioral aspects is uneven. As agent capabilities grow and new benchmarks emerge, additional evaluation areas will likely be introduced, and existing ones will continue to evolve. The structure presented here reflects the current state of the field while remaining flexible enough to accommodate future developments.

Example Case Study We consider a ReAct-style e-commerce customer-service agent [155] that retrieves information from knowledge bases and order systems, reasons over results, and executes actions such as refunds or ticket creation using session context and customer profiles. Along the **WHAT** axis, the agent exposes inferred intent, retrieved beliefs, response plans, dialogue memory, and tool I/O, with governance enforced through explicit service and escalation policies. Along the **WHY** axis, usefulness and faithfulness are critical for customer understanding and escalation decisions, while compliance and equity support auditability and monitoring of service disparities. Along the **HOW** axis, transparency relies on ReAct-style reasoning traces and systematic logging of tool calls and conversations, complemented by supervisor review mechanisms. Along the **WHEN** axis, policies and logging are defined at design time, trajectories are monitored at process time, and a compact MEP is produced at outcome time. Along the **WHO** axis, customers require clear explanations, developers require debugging traces, supervisors require escalation rationales, and auditors require compliance evidence. This analysis reveals recurring gaps: escalation decisions lack user-accessible rationale, attribution to prior customer history is opaque, uncertainty in retrieved information is under-communicated, and cross-session consistency is weak. We recommend escalation-focused MEPs that expose confidence and policy triggers, retrieval attribution linking responses to evidence sources, tiered explanations for different stakeholders, and structured logging of memory access to support equity auditing.

7 Discussion

Synthesis of Findings Four findings stand out. First, **agentic transparency is trajectory-centric**. For LLM agents, risk and responsibility emerge from sequences of decisions: how goals are interpreted, how plans evolve, which tools are invoked, what evidence is retrieved, and how memory is read and written. Transparency therefore cannot be reduced to explaining a final output; it must make the execution trajectory inspectable. Second, **transparency is lifecycle-dependent**. Post-hoc explanations are useful, but they are fragile when the underlying evidence was never captured. In practice, transparency requires design-time instrumentation choices (schemas, logging, access control), process-time integrity and monitoring (trace completeness, policy triggers, anomaly detection), and outcome-time packaging so that evidence can be reviewed and contested after the fact. This lifecycle view also clarifies why retrofitting transparency after deployment is costly and often incomplete.

Third, **transparency is stakeholder-specific but evidence should be shared**. End users, developers, auditors, and regulators need different views, yet these views should be derived from the same underlying trace substrate. This motivates a separation between (i) user-facing explanations that prioritize clarity and recourse, and (ii) verification-grade evidence that supports audit, replay, and governance reporting. A compact outcome-time artifact such as the Minimal Explanation Packet (MEP) can serve as the handoff between these needs by linking narratives to verifiable traces. Fourth, **faithfulness and usefulness are in tension**. Explanations that are easy to read can be unfaithful, while faithful traces can be overwhelming. The practical response is not to choose one, but to layer them: provide concise summaries for humans while preserving trace-grounded evidence for verification. In high-stakes settings, self-reported rationales should be treated as hypotheses and validated against operational traces or interventions when feasible [133].

Implications for Practice For practitioners, the main implication is that transparency should be engineered as a *system property* rather than produced as an afterthought. A practical workflow is to begin with stakeholders and assurance objectives, then decide which cognitive objects must be captured and protected at runtime (plans, tool I/O, memory updates, policy activations), and finally determine how these records will be summarized at outcome-time.

At minimum, systems should support: (i) stable identifiers for retrieved evidence and tool outputs, (ii) signed and time-stamped traces to provide integrity, and (iii) selective disclosure mechanisms so that privacy and security constraints do not force a shift to narrative-only explanations. For consequential decisions, emitting an MEP that links outcome summaries to plan and tool traces provides a concrete path to debugging, contestation, and audit without requiring stakeholders to interpret raw logs.

Implications for Policy Existing governance frameworks such as the EU AI Act, NIST AI RMF, and ISO/IEC 42001 provide valuable principles, but agentic deployments add process-time risks that are easy to miss under outcome-only compliance. Tool failures, memory leakage, planning drift, and delegation across agents create new loci of accountability. Policy requirements that treat "explainability" as a single obligation risk encouraging narrative summaries without verifiable grounding.

A more operational approach is to distinguish three artifact types: (i) user-facing explanations and recourse, (ii) compliance documentation that maps controls to requirements, and (iii) audit evidence grounded in provenance and replayable traces. Minimum expectations for agentic systems should include process-time logging/provenance, retention and access for review, and explicit triggers for escalation and oversight when policies activate or uncertainty is high.

Research Agenda Our framework is intended to be testable. We highlight three research directions: (1) Compare stakeholder performance under (i) structured packets (e.g., MEP-style artifacts), (ii) unstructured traces, and (iii) output-only baselines. Measure comprehension accuracy, time-to-diagnosis, and confidence calibration. (2) Develop benchmarks and protocols that test whether explanation claims match trace evidence (plan–trace agreement, citation-retrieval consistency, tool attribution checks), and evaluate robustness under perturbations and adversarial manipulation [126]. (3) Measure runtime and storage overhead, and study explanation drift as tools, prompts, policies, and memory evolve. Multi-agent deployments should be evaluated for responsibility attribution and cross-agent provenance linking.

Limitations of the survey We acknowledge some limitations of this work. First, the framework is a conceptual rather than empirically validated contribution. It provides organizational structure and shared vocabulary, but we have not demonstrated through user studies or deployment experiments that it improves transparency outcomes. Our coverage assessments are based on literature analysis rather than systematic empirical evaluation. Second, the field is evolving rapidly. New architectures, methods, and regulatory frameworks may require taxonomy refinement. While the axes are designed to be extensible, periodic revision will be necessary. Third, although we address multi-agent and multimodal settings, the majority of existing transparency methods-and thus much of our survey-focus on single-agent, text-primary systems. As deployments diversify, additional research and potential taxonomy extensions will be needed.

8 Conclusion

This survey systematizes transparency for agentic AI by introducing a lifecycle-aware taxonomy that unifies interpretability, explainability, and governance. Our analysis shows that transparency is not an optional enhancement but a prerequisite for responsible deployment of autonomous, tool-using agents. We identify persistent gaps in trajectory-level accountability, provenance, and multi-agent coordination, and outline an empirical research agenda to validate transparency mechanisms in practice. The proposed framework and Minimal Explanation Packet provide a foundation for measurable, auditable transparency in agentic AI systems.

Acknowledgement Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute. This research was funded by the European Union's Horizon Europe research and innovation programme under the AIXPERT project (Grant Agreement No. 101214389), which aims to develop an agentic, multi-layered, GenAI-powered framework for creating explainable, accountable, and transparent AI systems.

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.

[2] Amin Abolghasemi, Leif Azzopardi, Seyyed Hadi Hashemi, Maarten de Rijke, and Suzan Verberne. Evaluation of attribution bias in retrieval-augmented large language models. *arXiv preprint arXiv:2410.12380*, 2024.

[3] Emre Can Acikgoz, Carl Guo, Suvodip Dey, Akul Datta, Takyoung Kim, Gokhan Tur, and Dilek Hakkani-Tur. Td-eval: Revisiting task-oriented dialogue evaluation by combining turn-level precision with dialogue-level comparisons. In *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 113–132, 2025.

[4] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[5] Robert Andrews, Joachim Diederich, and Alan B Tickle. A survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 8(6):373–389, 1995.

[6] Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, et al. Agentharm: A benchmark for measuring harmfulness of llm agents. *arXiv preprint arXiv:2410.09024*, 2024.

[7] Rauno Arike, Elizabeth Donoway, Henning Bartsch, and Marius Hobbhahn. Technical report: Evaluating goal drift in language model agents. *arXiv preprint arXiv:2505.02709*, 2025.

[8] Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Explaining recurrent neural network predictions in sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

[9] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):e0130140, 2015.

[10] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: harmlessness from ai feedback. 2022. *arXiv preprint arXiv:2212.08073*, 8(3), 2022.

[11] Amine Barrak. Traceability and accountability in role-specialized multi-agent llm pipelines. *arXiv preprint arXiv:2510.07614*, 2025.

[12] Ahsan Bilal, David Ebert, and Beiyu Lin. Llms for explainable ai: A comprehensive survey. *arXiv preprint arXiv:2504.00125*, 2025.

[13] Ruta Binkyte. Interactional fairness in llm multi-agent systems: An evaluation framework. *arXiv preprint arXiv:2505.12001*, 2025.

[14] Angana Borah and Rada Mihalcea. Towards implicit bias detection and mitigation in multi-agent llm interactions. *arXiv preprint arXiv:2410.02584*, 2024.

[15] Uwe M Borghoff, Paolo Bottoni, and Remo Pareschi. Human-artificial interaction in the age of agentic ai: a system-theoretical approach. *Frontiers in Human Dynamics*, 7:1579166, 2025.

[16] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[17] Hongye Cao, Fan Feng, Jing Huo, and Yang Gao. Causal action empowerment for efficient reinforcement learning in embodied agents. *Science China Information Sciences*, 68(5):150201, 2025.

[18] Jiahuan Cao, Yongxin Shi, Dezhi Peng, Yang Liu, and Lianwen Jin. C^3bench: A comprehensive classical chinese understanding benchmark for large language models. *arXiv preprint arXiv:2405.17732*, 2024.

[19] Jize Cao, Zhe Gan, Yu Cheng, Licheng Yu, Yen-Chun Chen, and Jingjing Liu. Behind the scene: Revealing the secrets of pre-trained vision-and-language models. In *European Conference on Computer Vision*, pages 565–580. Springer, 2020.

[20] Stephen Casper, Luke Bailey, Rosco Hunter, Carson Ezell, and Others. The ai agent index, 2025.

[21] David Castillo-Bolado, Joseph Davidson, Finlay Gray, and Marek Rosa. Beyond prompts: Dynamic conversational benchmarking of large language models. *Advances in Neural Information Processing Systems*, 37:42528–42565, 2024.

[22] Ma Chang, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. Agentboard: An analytical evaluation board of multi-turn llm agents. *Advances in neural information processing systems*, 37:74325–74362, 2024.

[23] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 782–791, 2021.

[24] Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. Agentpoison: Red-teaming llm agents via poisoning memory or knowledge bases. *Advances in Neural Information Processing Systems*, 37:130185–130213, 2024.

[25] Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

[26] Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

[27] Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

[28] Mark W Craven and Jude W Shavlik. Using sampling and queries to extract rules from trained neural networks. In *Machine learning proceedings 1994*, pages 37–45. Elsevier, 1994.

[29] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

[30] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. Eraser: A benchmark to evaluate rationalized nlp models. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, 2020.

[31] Liming Dong, Qinghua Lu, and Liming Zhu. Agentops: Enabling observability of llm agents. *arXiv preprint arXiv:2411.05285*, 2024.

[32] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[33] Alex Egg, Martin Iglesias Goyanes, Friso Kingma, Andreu Mora, Leandro von Werra, and Thomas Wolf. Dabstep: Data agent benchmark for multi-step reasoning. *arXiv preprint arXiv:2506.23719*, 2025.

[34] European Union. General Data Protection Regulation (GDPR) – Article 25: Data protection by design and by default. `https://gdpr-info.eu/art-25-gdpr/`, 2016. Accessed: 2025-06-03.

[35] European Union. Eu ai act. `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206`, 2025. Accessed: 2025-07-23.

[36] Azib Farooq, Shaina Raza, Md Nazmul Karim, Hasan Iqbal, Athanasios V Vasilakos, and Christos Emmanouilidis. Evaluating and regulating agentic ai: A study of benchmarks, metrics, and regulation. *Metrics, and Regulation*, 2025.

[37] Jianbing Feng, Tao Yu, Kuozhen Zhang, and Lefeng Cheng. Integration of multi-agent systems and artificial intelligence in self-healing subway power supply systems: Advancements in fault diagnosis, isolation, and recovery. *Processes*, 13(4):1144, 2025.

[38] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinyang Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

[39] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

[40] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

[41] Google Cloud. Analysis of the potential of agentic AI, 2024. Accessed: January 2025.

[42] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[43] Shubham Gupta. Ai agents collaboration under resource constraints: Practical implementations. *INTERNATIONAL JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH AND DEVELOPMENT*, 3(1):51–63, 2025.

[44] Xiaoshuai Hao, Guanqun Liu, Yuting Zhao, Yuheng Ji, Mengchuan Wei, Haimei Zhao, Lingdong Kong, Rong Yin, and Yu Liu. Msc-bench: Benchmarking and analyzing multi-sensor corruption for driving perception. *arXiv preprint arXiv:2501.01037*, 2025.

[45] Stefan Heimersheim and Jett Janiak. A circuit for python docstrings in a 4-layer attention-only transformer. In *Alignment Forum*, 2023.

[46] John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4129–4138, 2019.

[47] Miriam Horovicz. Agentshap: Interpreting llm agent tool importance with monte carlo shapley value estimation. *arXiv preprint arXiv:2512.12597*, 2025.

[48] Wenyue Hua, Xianjun Yang, Mingyu Jin, Zelong Li, Wei Cheng, Ruixiang Tang, and Yongfeng Zhang. Trustagent: Towards safe and trustworthy llm-based agents. *arXiv preprint arXiv:2402.01586*, 2024.

[49] Ken Huang. *Agentic AI*. Springer, 2025.

[50] Tim Hulsen. Explainable artificial intelligence (xai): concepts and challenges in healthcare. *Ai*, 4(3):652–666, 2023.

[51] International Organization for Standardization. Iso/iec 42001:2023 – artificial intelligence management system (ai ms) – requirements. Technical report, ISO/IEC, 2023. Available at `https://www.iso.org/standard/81230.html`.

[52] Alon Jacovi, Oren Sar Shalom, and Yoav Goldberg. Understanding convolutional neural networks for text classification. *arXiv preprint arXiv:1809.08037*, 2018.

[53] Sarthak Jain and Byron C Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3543–3556. ACL, 2019.

[54] Nidhal Jegham, Marwan Abdelatti, Chan Young Koh, Lassad Elmoubarki, and Abdeltawab Hendawi. How hungry is ai? benchmarking energy, water, and carbon footprint of llm inference. *arXiv preprint arXiv:2505.09598*, 2025.

[55] Allison Sihan Jia, Daniel Huang, Nikhil Vytla, Nirvika Choudhury, John C Mitchell, and Anupam Datta. What is your agent's gpa? a framework for evaluating agent goal-plan-action alignment. *arXiv preprint arXiv:2510.08847*, 2025.

[56] Gayeon Jung, Hyeonseok Lim, Minjun Kim, Joon-Ho Lim, KyungTae Lim, and Hansaem Kim. Can llms truly plan? a comprehensive evaluation of planning capabilities. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13069–13084, 2025.

[57] Sayash Kapoor, Benedikt Stroebl, Peter Kirgis, Nitya Nadgir, Zachary S Siegel, Boyi Wei, Tianci Xue, Ziru Chen, Felix Chen, Saiteja Utpala, et al. Holistic agent leaderboard: The missing infrastructure for ai agent evaluation. *arXiv preprint arXiv:2510.11977*, 2025.

[58] Tahniat Khan, Soroor Motie, Sedef Akinli Kocak, and Shaina Raza. Optimizing large language models: Metrics, energy efficiency, and case study insights. In *2025 IEEE Conference on Artificial Intelligence (CAI)*, pages 370–375, 2025.

[59] Been Kim, John Hewitt, Neel Nanda, Noah Fiedel, and Oyvind Tafjord. Because we have llms, we can and should pursue agentic interpretability. *arXiv preprint arXiv:2506.12152*, 2025.

[60] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. Interpretability beyond feature attribution: Testing with concept activation vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. PMLR, 2018.

[61] Jin Kim, Muhammad Wahi-Anwa, Sangyun Park, Shawn Shin, John M. Hoffman, and Matthew S. Brown. Autonomous computer vision development with agentic ai, 2025.

[62] Barbara Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering. *EBSE Technical Report*, 2(EBSE-2007-01):1–65, 2007.

[63] Noam Kolt. Governing ai agents. *arXiv preprint arXiv:2501.07913*, 2025.

[64] Julius Krause, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Explaining black-box models through counterfactuals. *arXiv preprint arXiv:2308.07198*, 2023.

[65] Lars Krupp, Daniel Geißler, Vishal Banwari, Paul Lukowicz, and Jakob Karolus. Promoting sustainable web agents: Benchmarking and estimating energy consumption through empirical and theoretical analysis. *arXiv preprint arXiv:2511.04481*, 2025.

[66] Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008, 2023.

[67] Kevin Y Li, Sachin Goyal, Joao D Semedo, and J Zico Kolter. Inference optimal vlms need fewer visual tokens and more parameters. *arXiv preprint arXiv:2411.03312*, 2024.

[68] Lingyu Li, Yixu Wang, Haiquan Zhao, Shuqi Kong, Yan Teng, Chunbo Li, and Yingchun Wang. Reflection-bench: Evaluating epistemic agency in large language models. *arXiv preprint arXiv:2410.16270*, 2024.

[69] Zihao Lin, Samyadeep Basu, Mohammad Beigi, Varun Manjunatha, Rossi, et al. A survey on mechanistic interpretability for multi-modal foundation models. *arXiv preprint arXiv:2502.17516*, 2025.

[70] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[71] Haiyan Liu et al. Explainability for large language models: A survey, 2023.

[72] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

[73] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as Agents, October 2023. arXiv:2308.03688 [cs].

[74] Qian Long, Zhi Li, Ran Gong, Ying Nian Wu, Demetri Terzopoulos, and Xiaofeng Gao. Teamcraft: A benchmark for multi-modal multi-agent systems in minecraft. *arXiv preprint arXiv:2412.05255*, 2024.

[75] Alexandra Sasha Luccioni and Alex Hernandez-Garcia. Counting carbon: A survey of factors influencing the emissions of machine learning. *arXiv preprint arXiv:2302.08476*, 2023.

[76] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017. Published in Advances in Neural Information Processing Systems 30 (NeurIPS 2017).

[77] Haoyan Luo and Lucia Specia. From understanding to utilization: A survey on explainability for large language models, 2024.

[78] Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, et al. Large language model agent: A survey on methodology, applications and challenges. *arXiv preprint arXiv:2503.21460*, 2025.

[79] Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. In *The 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL 2023)*, 2023.

[80] Xinbei Ma, Yiting Wang, Yao Yao, Tongxin Yuan, Aston Zhang, Zhuosheng Zhang, and Hai Zhao. Caution for the environment: Multimodal agents are susceptible to environmental distractions. *arXiv preprint arXiv:2408.02544*, 2024.

[81] Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. *arXiv preprint arXiv:2402.17753*, 2024.

[82] Market.us. Agentic AI market report, 2024. Accessed: January 2025.

[83] Matias Martinez and Xavier Franch. Dissecting the swe-bench leaderboards: Profiling submitters and architectures of llm-and agent-based repair systems. *arXiv preprint arXiv:2506.17208*, 2025.

[84] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, 2022.

[85] Imran Mirza, Cole Huang, Ishwara Vasista, Rohan Patil, Asli Akalin, Sean O'Brien, and Kevin Zhu. Malibu benchmark: Multi-agent llm implicit bias uncovered. *arXiv preprint arXiv:2507.01019*, 2025.

[86] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 220–229. ACM, January 2019.

[87] Brent Mittelstadt. Principles alone cannot guarantee ethical ai. *Nature machine intelligence*, 1(11):501–507, 2019.

[88] Guozhao Mo, Wenliang Zhong, Jiawei Chen, Xuanang Chen, Yaojie Lu, Hongyu Lin, Ben He, Xianpei Han, and Le Sun. Livemcpbench: Can agents navigate an ocean of mcp tools? *arXiv preprint arXiv:2508.01780*, 2025.

[89] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222, 2017.

[90] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*, 2019.

[91] National Institute of Standards and Technology. Artificial intelligence risk management framework (ai rmf 1.0). Technical Report NIST AI 100-1, NIST, 01 2023.

[92] National Institute of Standards and Technology. Artificial intelligence risk management framework (ai rmf 1.0), 2023. NIST AI 100-1.

[93] Ume Nisa, Muhammad Shirazi, Mohamed Ali Saip, and Muhammad Syafiq Mohd Pozi. Agentic ai: The age of reasoning—a review. *Journal of Automation and Intelligence*, 2025.

[94] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

[95] Matthew J Page, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, et al. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *bmj*, 372, 2021.

[96] Avash Palikhe, Zhenyu Yu, Zichong Wang, and Wenbin Zhang. Towards transparent ai: A survey on explainable large language models. *arXiv preprint arXiv:2506.21812*, 2025.

[97] Rock Yuren Pang, KJ Feng, Shangbin Feng, Chu Li, Weijia Shi, Yulia Tsvetkov, Jeffrey Heer, and Katharina Reinecke. Interactive reasoning: Visualizing and controlling chain-of-thought reasoning in large language models. *arXiv preprint arXiv:2506.23678*, 2025.

[98] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023.

[99] Francesco Piccialli, Diletta Chiaro, Sundas Sarwar, Donato Cerciello, Pian Qi, and Valeria Mele. Agentai: A comprehensive survey on autonomous agents in distributed ai for industry 4.0. *Expert Systems with Applications*, page 128404, 2025.

[100] Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program), 2020.

[101] Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Back. Reasoning with large language models, a survey. *arXiv preprint arXiv:2407.11511*, 2024.

[102] Eleonora Poeta, Gabriele Ciravegna, Eliana Pastor, Tania Cerquitelli, and Elena Baralis. Concept-based explainable artificial intelligence: A survey, 2023.

[103] Danish Pruthi, Rachit Bansal, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C Lipton, Graham Neubig, and William W Cohen. Evaluating explanations: How much do explanations from the teacher aid students? *Transactions of the Association for Computational Linguistics*, 10:359–375, 2022.

[104] Chen Qian, Xin Cong, Wei Liu, Cheng Yang, Weize Chen, Yusheng Su, Yufan Zhang, et al. Chatdev: Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 2024.

[105] Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*, 2019.

[106] Shaina Raza, Oluwanifemi Bamgbose, Shardul Ghuge, Fatemeh Tavakoli, Deepak John Reji, and Syed Raza Bashir. Developing safe and responsible large language model: can we balance bias reduction and language understanding? *Machine Learning*, 114(6):140, 2025.

[107] Shaina Raza, Ranjan Sapkota, Manoj Karkee, and Christos Emmanouilidis. Responsible agentic reasoning and ai agents: A critical survey. *Authorea Preprints*, 2025.

[108] Shaina Raza, Ranjan Sapkota, Manoj Karkee, and Christos Emmanouilidis. Trism for agentic ai: A review of trust, risk, and security management in llm-based agentic multi-agent systems, 2025.

[109] Shaina Raza, Ranjan Sapkota, Manoj Karkee, and Christos Emmanouilidis. Trism for agentic ai: A review of trust, risk, and security management in llm-based agentic multi-agent systems, 2025.

[110] Shaina Raza, Arash Shaban-Nejad, Elham Dolatabadi, and Hiroshi Mamiya. Exploring bias and prediction metrics to characterise the fairness of machine learning for equity-centered public health decision-making: A narrative review. *IEEE Access*, 2024.

[111] Shaina Raza, Ashmal Vayani, Aditya Jain, Aravind Narayanan, Vahid Reza Khazaie, Syed Raza Bashir, Elham Dolatabadi, Gias Uddin, Christos Emmanouilidis, Rizwan Qureshi, et al. Vldbench evaluating multimodal disinformation with regulatory alignment. *arXiv preprint arXiv:2502.11361*, 2025.

[112] Matthew Renze and Erhan Guven. Self-reflection in llm agents: Effects on problem-solving performance. *arXiv preprint arXiv:2405.06682*, 2024.

[113] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016. Published in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016).

[114] Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J Maddison, and Tatsunori Hashimoto. Identifying the risks of lm agents with an lm-emulated sandbox. *arXiv preprint arXiv:2309.15817*, 2023.

[115] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[116] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

[117] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2024.

[118] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[119] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

[120] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180, 2023.

[121] Noah Shinn, Francesco Cassano, and Edward Labash. Reflexion: An autonomous agent with dynamic memory and self-reflection. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.

[122] Mohammad Shukor, Hang Le, James Requeijo, Samuel Lavoie, Marco J. Maier, and Ivan V. Titov. A concept-based explainability framework for large multimodal models. 2024.

[123] Zachary S Siegel, Sayash Kapoor, Nitya Nagdir, Benedikt Stroebl, and Arvind Narayanan. Core-bench: Fostering the credibility of published research through a computational reproducibility agent benchmark. *arXiv preprint arXiv:2409.11363*, 2024.

[124] Significant Gravitas. AutoGPT.

[125] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[126] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

[127] Theodore Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas Griffiths. Cognitive architectures for language agents. *Transactions on Machine Learning Research*, 2023.

[128] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International conference on machine learning*, pages 9269–9278. PMLR, 2020.

[129] Haoran Tan, Zeyu Zhang, Chen Ma, Xu Chen, Quanyu Dai, and Zhenhua Dong. Membench: Towards more comprehensive evaluation on the memory of llm-based agents. *arXiv preprint arXiv:2506.21605*, 2025.

[130] Alexander Timms, Abigail Langbridge, and Fearghal O'Donncha. Agentic anomaly detection for shipping. In *NeurIPS 2024 Workshop on Open-World Agents*, 2024.

[131] Geoffrey G Towell and Jude W Shavlik. Extracting refined rules from knowledge-based neural networks. *Machine learning*, 13(1):71–101, 1993.

[132] Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. *arXiv preprint arXiv:2407.18901*, 2024.

[133] Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting, 2023.

[134] Abhinav Verma, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. Programmatically interpretable reinforcement learning, 2019.

[135] Sanidhya Vijayvargiya, Aditya Bharat Soni, Xuhui Zhou, Zora Zhiruo Wang, Nouha Dziri, Graham Neubig, and Maarten Sap. Openagentsafety: A comprehensive framework for evaluating real-world ai agent safety. *arXiv preprint arXiv:2507.06134*, 2025.

[136] W3C Provenance Working Group. Prov-overview: An overview of the prov family of documents. W3C Working Group Note, April 30 2013. Available from: `https://www.w3.org/TR/prov-overview/`.

[137] Hao Wang, Jiajun Zhang, Yang Liu, and Xinyu Li. Layered chain-of-thought prompting for multi-agent llm systems: A comprehensive approach to explainable large language models. *arXiv preprint arXiv:2501.18645*, 2025.

[138] Jize Wang, Ma Zerun, Yining Li, Songyang Zhang, Cailian Chen, Kai Chen, and Xinyi Le. Gta: a benchmark for general tool agents. *Advances in Neural Information Processing Systems*, 37:75749–75790, 2024.

[139] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

[140] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*, 2023.

[141] Luyuan Wang, Yongyu Deng, Yiwei Zha, Guodong Mao, Qinmin Wang, Tianchen Min, Wei Chen, and Shoufa Chen. Mobileagentbench: An efficient and user-friendly benchmark for mobile llm agents. *arXiv preprint arXiv:2406.08184*, 2024.

[142] Shouju Wang, Fenglin Yu, Xirui Liu, Xiaoting Qin, Jue Zhang, Qingwei Lin, Dongmei Zhang, and Saravan Rajmohan. Privacy in action: Towards realistic privacy mitigation and evaluation for llm-powered agents. *arXiv preprint arXiv:2509.17488*, 2025.

[143] Yuqing Wang and Yun Zhao. Rupbench: Benchmarking reasoning under perturbations for robustness evaluation in large language models. *arXiv preprint arXiv:2406.11020*, 2024.

[144] Zhenting Wang, Qi Chang, Hemani Patel, Shashank Biju, Cheng-En Wu, Quan Liu, Aolin Ding, Alireza Rezazadeh, Ankit Shah, Yujia Bao, et al. Mcp-bench: Benchmarking tool-using llm agents with complex real-world tasks via mcp servers. *arXiv preprint arXiv:2508.20453*, 2025.

[145] Jane Webster and Richard T Watson. Analyzing the past to prepare for the future: Writing a literature review. *MIS quarterly*, pages xiii–xxiii, 2002.

[146] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[147] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024.

[148] Xinyi Wu, Geng Hong, Pei Chen, Yueyue Chen, Xudong Pan, and Min Yang. Prison: Unmasking the criminal potential of large language models. *arXiv preprint arXiv:2506.16150*, 2025.

[149] Xuansheng Wu, Haiyan Zhao, Yaochen Zhu, Yucheng Shi, Fan Yang, Lijie Hu, Tianming Liu, Xiaoming Zhai, Wenlin Yao, Jundong Li, et al. Usable xai: 10 strategies towards exploiting explainability in the llm era. *arXiv preprint arXiv:2403.08946*, 2024.

[150] Frank F Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, et al. Theagentcompany: benchmarking llm agents on consequential real world tasks. *arXiv preprint arXiv:2412.14161*, 2024.

[151] Lin Xu, Zhiyuan Hu, Daquan Zhou, Hongyu Ren, Zhen Dong, Kurt Keutzer, See Kiong Ng, and Jiashi Feng. Magic: Investigation of large language model powered multi-agent in cognition, adaptability, rationality and collaboration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7315–7332, 2024.

[152] Weikai Xu, Chengrui Huang, Shen Gao, and Shuo Shang. Llm-based agents for tool learning: A survey: W. xu et al. *Data Science and Engineering*, pages 1–31, 2025.

[153] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. tau-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

[154] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

[155] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023.

[156] Haiyan Yin et al. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–38, 2024.

[157] Xuefei Yin, Yanming Zhu, and Jiankun Hu. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Computing Surveys (CSUR)*, 54(6):1–36, 2021.

[158] Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. Assistantbench: Can web agents solve realistic and time-consuming tasks? *arXiv preprint arXiv:2407.15711*, 2024.

[159] Chaojia Yu, Zihan Cheng, Hanwen Cui, Yishuo Gao, Zexu Luo, Yijin Wang, Hangbin Zheng, and Yong Zhao. A survey on agent workflow–status and future. In *2025 8th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pages 770–781. IEEE, 2025.

[160] Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, et al. R-judge: Benchmarking safety risk awareness for llm agents. *arXiv preprint arXiv:2401.10019*, 2024.

[161] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.

[162] Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):1–47, 2025.

[163] Changyuan Zhao, Ruichen Zhang, Jiacheng Wang, Gaosheng Zhao, Dusit Niyato, Geng Sun, Shiwen Mao, and Dong In Kim. World models for cognitive agents: Transforming edge intelligence in future networks. *arXiv preprint arXiv:2506.00417*, 2025.

[164] Xinyi Zhao et al. A survey on explainability for large language models, 2024.

[165] Arman Zharmagambetov, Chuan Guo, Ivan Evtimov, Maya Pavlova, Ruslan Salakhutdinov, and Kamalika Chaudhuri. Agentdam: Privacy leakage evaluation for autonomous web agents. *arXiv preprint arXiv:2503.09780*, 2025.

[166] Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks. *arXiv preprint arXiv:2503.15478*, 2025.

[167] Kaijie Zhu, Qinlin Zhao, Hao Chen, Jindong Wang, and Xing Xie. Promptbench: A unified library for evaluation of large language models. *Journal of Machine Learning Research*, 25(254):1–22, 2024.

[168] Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Daisy Zhe Wang, Zhenhailong Wang, Cheng Qian, Robert Tang, Heng Ji, et al. Multiagentbench: Evaluating the collaboration and competition of llm agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8580–8622, 2025.