Agentic Transparency: A Practical Taxonomy for Interpretability and Explainability (X-AXIOM)

September 2025

Contents

1 Introduction

Over the past decade, artificial intelligence (AI) has evolved from static predictive models to generative systems capable of interaction, reasoning, and adaptation. The latest step in this progression is Agentic AI [51], which builds on large language models (LLMs) by coupling their language understanding with autonomous decision-making and goal-directed behavior. LLMs have already catalyzed advances across natural language processing, reasoning, and multimodal applications. Building on these foundations, Agentic AI[1] moves beyond isolated prompt–response exchanges: systems reason about tasks, plan coherent sequences of actions, and adapt their behavior in pursuit of defined objectives [16].

[1]Throughout this paper, the term **Agentic AI** refers specifically to LLM-based autonomous agents.

As these systems gain greater autonomy and operate in richer environments, they also become more opaque. Traditional interpretability [62] and explainability [11] methods, originally designed for static predictive models, are increasingly insufficient to capture the evolving internal states and decision processes of Agentic AI systems. In this paper, we use *explainability* to denote the ability to communicate how and why a system produced a given output in human-understandable terms, and *interpretability* to denote transparency of its internal representations and mechanisms.

For users, regulators, and developers alike, three basic questions keep resurfacing: *Why did I get this result? What led to it?* and *What does it mean for me?* These questions expose a central tension: as agency increases a system's capabilities, it also increases its complexity and risk surface [6], making interpretability and explainability essential for trust, safety, and accountability in Agentic AI. This motivates a shift from one-off explanations to *agentic transparency* as a lifecycle property, an idea we develop in the rest of this survey.

However, despite this need, the intersection of interpretability, explainability, and Agentic AI remains fragmented. Prior taxonomies in explainability and interpretability summarise methods and goals, but mostly target static or single-step models rather than agents that act over time. A few works distinguish model-centric transparency from user-centric communication [68, 32], but their application to Agentic AI is still limited [128, 95] due to the novelty and complexity of the setting.

Agentic AI systems add new layers of opacity in how decisions unfold. They maintain state across steps, call external tools, update memories, and coordinate multi-stage plans, sometimes across multiple agents. These behaviours make many existing taxonomies difficult to apply without modification. While we understand a great deal about explaining single outputs, we still lack shared frameworks for explaining processes, policies, and interactions in Agentic AI [128, 95]. This gap motivates methods and evaluations that connect step-level traces to human-understandable explanations, while remaining faithful to the agent's internal updates [83].

In this work, we present a comprehensive survey that synthesizes interpretability and explainability for Agentic AI across three stages: (1) design-time transparency, (2) process-level interpretability, and (3) outcome-level explanations. We group existing approaches, surface challenges specific to multi-agent and multimodal settings, and propose simple principles for interpretable, accountable, and human-aligned Agentic AI. Below, we position our survey against recent related work.

1.1 Contributions

This survey treats agentic transparency as a lifecycle property of LLM-based agents rather than a single post-hoc explanation step. Our main contributions are:

1. We introduce *X-AXIOM*, a taxonomy that organises transparency for LLM-based agents along five axes: *Cognitive Objects (WHAT)*, *Assurance and Evaluation Objectives (WHY)*, *Mechanisms (HOW)*, *Temporal Stages (WHEN)*, and *Multi-agent / socio-technical extensions (WHO)*, providing a shared vocabulary that links interpretability, explainability, and governance.

2. We formalise a *cognitive audit surface* (intent, beliefs, plans, memory, tool I/O, policies, outcomes) and define six assurance objectives: *faithfulness, usefulness, compliance, robustness, equity, auditability*—connecting them to governance frameworks such as the EU AI Act, NIST AI RMF, and ISO/IEC 42001.

3. We survey methods from interpretability, explainable AI, mechanistic interpretability, and operational monitoring, and map them into the X-AXIOM space, highlighting where today's tools already support agentic transparency and where gaps remain (e.g., multi-step planning, tool use, memory, socio-technical settings).

4. We outline evaluation protocols that align design-time, process-time, and outcome-time checks with the six assurance objectives, showing how X-AXIOM can guide concrete transparency and audit practices for Agentic AI systems.

1.2 Necessity of this Survey

As Table 1 shows, prior work tends to split into two lines. Surveys on explainability and interpretability focus mainly on static or single-step models, including recent LLM-oriented overviews, but they do not treat agentic workflows or multi-step decision processes in depth. In parallel, surveys on Agentic AI and LLM-based agents describe architectures, tools, and workflows, but they give only brief or fragmented coverage of explanation and interpretation.

This survey is needed to bridge the two strands. We provide a unified view of explainability and interpretability for Agentic AI, covering design-time choices (models, tools, and roles), process-level behavior (traces, planning, memory, and interaction), and outcome-level effects (explanations, user experience, and evaluation). Our focus is specifically on LLM-based agents, including multi-agent and multimodal settings, and on how to connect step-by-step agent behavior to explanations that remain faithful to what the system actually does.

Related survey	Year	Expl.	Interp.	Agentic	Notes
[32]	2017	✗	✓	✗	Defines interpretability as human-simulatability; eval. levels.
[68]	2018	✓	✓	✗	Intrinsic transparency vs. post-hoc; simulatability/decomposability.
[41]	2018	✓	✓	✗	Internals (interpretability) vs. user-facing (explainability).
[83]	2019	✓	✓	✗	Model-based (intrinsic) vs. post-hoc; PDR evaluation.
[52]	2023	✓	✗	✗	General XAI overview.
[152]	2024	✓	✓	✗	LLM-focused XAI; local/global; prompting vs. fine-tuning.
[72]	2024	✓	✓	✗	LLM explainability emphasis.
[136]	2025	✗	✓	✗	Methods for interpretability.
[15]	2025	✓	✓	✗	LLMs used as explainers (XAI tooling).
[73]	2025	✗	✗	✓	LLM agent methodologies/workflows.
[67]	2025	✓	✓	✗	Broad XAI taxonomy.
[143]	2025	✗	✗	✓	Agent workflows: planning, tools, memory.
[94]	2025	✗	✗	✓	Autonomous agents review.
[137]	2025	✗	✗	✓	Tool-use agents survey.
[86]	2025	✗	✗	✓	Agentic AI review (brief on XAI).
[89]	2025	✓	✓	✗	Usability-oriented XAI.
Ours (2025)	2025	✓	✓	✓	Unifies design-time, process-level, outcome-level; multi-agent & multimodal.

Table 1: Comparison across explainability, interpretability, and agentic coverage. ✓= explicit focus, ✗= no dedicated treatment.

2 Background

2.1 Preliminaries

Key terms used in this survey are given in Table 2. Some of the notations used in this paper are in Table 3

2.2 Definitions of Interpretability and Explainability

Interpretability and explainability remain closely related yet distinct concepts in AI. Interpretability generally refers to how much of a model's internal mechanics, such as parameters, intermediate states, or structure that can be directly understood or anticipated by a human . Explainability, in contrast, focuses on producing human-readable accounts of *why* a model arrived at a particular output. Early perspectives emphasized transparency for interpretable models and post-hoc rationalizations for opaque ones, largely within static supervised learning settings [74].

As models became more complex, interpretability expanded to include methods that expose internal representations or mechanisms, such as feature attribution and saliency analysis, aiming for faithfulness to the model's actual computations [61]. Explainability increasingly centered on effective communication of

Table 2: Key terms used in the survey.

Term	Short definition	Key citation(s)
Large Language Model	Transformer-based neural model trained on large text corpora. Provides core reasoning and generation ability for many agents.	[113]
Agentic AI	Systems with autonomy to perceive, plan, and act to reach goals. Adds memory, tool use, and coordination beyond single-turn LLM use.	[16]
Agent Loop	Recurring cycle of *Perception* → *Reasoning* → *Planning* → *Tool Use* → *Action* → *Reflection/Memory Update*. Enables longer-horizon behavior.	[128, 95]
Explainable AI (XAI)	Methods that make model outputs understandable to people. Helps answer why a result was produced and supports trust.	[68]
Interpretability	How much a model's internals can be understood or traced, for example parameters, intermediate states, or reasoning steps.	[83]
Agentic Transparency	Degree to which an agent's goals, intermediate states, tool calls, and actions over time can be described and justified in a way that is understandable to humans.	[132]
Policy	Mapping from an agent's current state (including context and memory) to its next action, tool call, or plan.	[17]
Execution Trace	Time-ordered record of an agent's perceptions, internal states, tool calls, actions, and outputs during a run. Used for analysis, debugging, and explanation.	[128, 95]
Memory	Structured store of past observations, interactions, or summaries that the agent can reuse to condition future reasoning and actions.	[16]
TRiSM	Trust, Risk, and Security Management. Practices for explainability, control, and governance aligned with the EU AI Act, NIST AI RMF, and ISO/IEC 42001.	EU AI Act; NIST AI RMF; ISO/IEC 42001 [98]

Table 3: Notation used in the paper.

Symbol	Meaning
$\mathcal{E}$	Environment
$\mathcal{S}$	Set of states
$\mathcal{A}$	Agent
$\mathcal{A}\rfloor$	Set of actions
$\mathcal{T}$	Transition function between states
π	Policy that maps states to actions
M	Base model, for example an LLM
$\mathcal{F}$	External tools or functions the agent can call

reasoning, user understanding, and trust [33]. Contemporary views treat the two as complementary: interpretability clarifies *how* a system operates, while explainability conveys *why* specific decisions emerge. Rather than a strict separation, they form a continuum that supports different aspects of model understanding.

2.3 Agentic AI

Agentic AI refers to systems that can perceive, "reason/plan", and "act" toward goals, rather than only generating text, often over multiple steps and with feedback from the environment [128] . Agentic AI generally, falls into two families (Fig. **??**). Traditional methods include rule-driven systems (FSMs, rule engines, symbolic planners), learning-driven systems (RL, classic ML, evolutionary search), and optimization-driven systems (constraint/mathematical programming, graph/search). LLM-based methods build on this with single-agent assistants (chat, RAG, tool use) and multi-agent organizations (coordinator–worker, role teams, swarms).

Traditional vs. LLM-Based Agentic AI Historically, agentic behavior was implemented through three main approaches.

1. **Rule-driven agents** relied on explicit logic, state machines, and symbolic planners that encoded the agent's possible behaviors [125].

2. **Learning-driven agents**—such as reinforcement learning and evolutionary systems—learned policies through experience, often in simulation [3, 40].

3. **Optimization-driven agents** used constraint solvers, mathematical programming, or search algorithms to compute actions under well-defined constraints [92, 77].

Modern **LLM-based agents** extend these ideas but gain flexibility from natural language reasoning [128]. Single-agent frameworks combine an LLM with tools or retrieval systems to handle tasks like planning, querying external knowledge, or executing code [107]. Multi-agent systems go further by coordinating several LLMs, each with specialized roles (e.g., planner, critic, executor) [56]. These setups allow for more complex workflows, collaborative reasoning, and distributed problem solving.

Agentic AI Architecture Most Agent AI systems: traditional or LLM-based, share a similar high-level loop: (1) perceive inputs, (2) reason and plan, (3) act, and (4) update memory. In practice, this loop is organized into layers.

1. The **perception layer** takes user or environment inputs (text, images, sensor readings) and transforms them into a form the agent can work with [90].

2. The **reasoning and planning layer** interprets the input, breaks down tasks into manageable steps, retrieves useful context or memories, and formulates a plan [114].

3. The **action layer** executes the plan through tool calls, API interactions, or code execution, and may include verification steps to ensure correctness [150].

LLM-based designs commonly extend this loop with optional modules such as memory for persistent state, reflection for improving strategies, verification for safety, profiling for domain-specific role conditioning, and orchestration mechanisms for coordinating multiple agents. We show this in Figure 1.

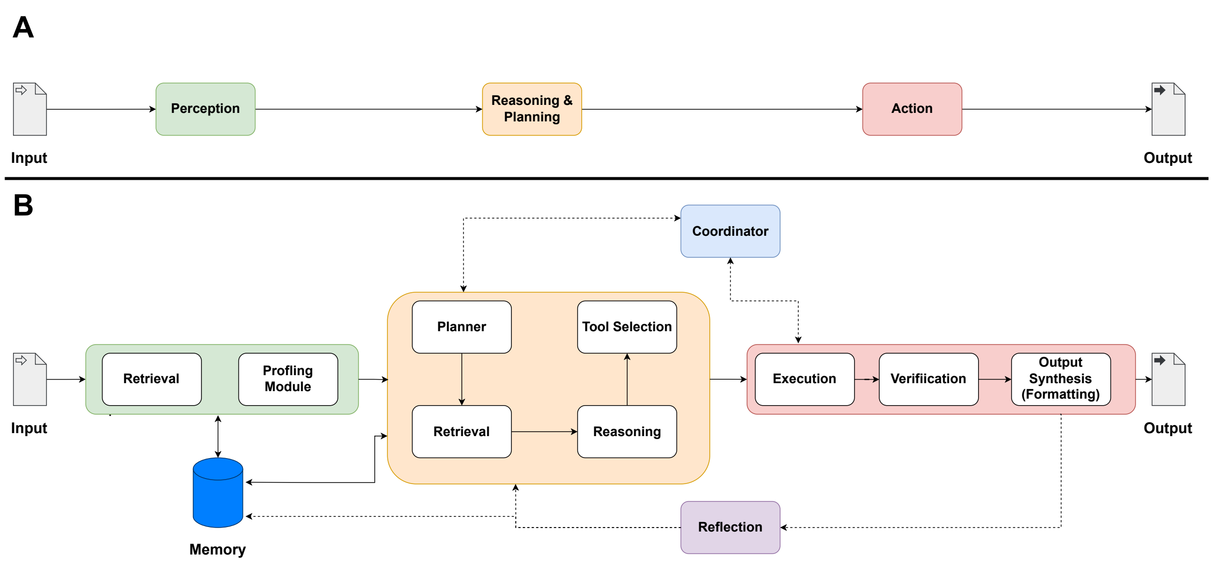

Figure 1: Agent architecture. (A) Classical three-layer flow: perception, reasoning & planning, and action, common to both traditional AI agents and modern LLM-based systems. (B) Expanded modular view reflecting contemporary LLM-centric design, with retrieval, profiling, planning, tool selection, execution, verification, reflection, and memory components.

Agentic AI Frameworks Recent years have produced many practical systems. Early examples such as ReAct [139] and Toolformer [106] combined reasoning and tool use for single-agent workflows. More autonomous frameworks like AgentGPT [1], AutoGen [36], and CAMEL [64] introduced multi-step planning and structured conversations between agents. Multi-agent systems such as Generative Agents [90], SWE-Agent [138], MAIA [112], Magnetic-One [37], and WorkForce [50] coordinate several specialized agents for domains like web navigation, scientific exploration, software engineering, or multimodal analysis. Domain-specific systems, for example Agent Laboratory [108], InternAgent [118], PlanGen [91], and MAGNET [24] target tasks ranging from research automation to constraint-based planning and audio–visual reasoning.

Overall, the progression reflects a shift from rigid, rule-based agents to flexible LLM-driven systems that integrate reasoning, planning, memory, and tool use, and increasingly collaborate through multi-agent structures.

2.4 Literature Review Method

This work is a structured narrative review rather than a full systematic review, but our search and screening process followed key stages from the PRISMA 2020 guidelines (identification, screening, eligibility, inclusion) [88].

Identification. We searched scholarly databases and archives (e.g., Google Scholar, arXiv, ACM Digital Library, IEEE Xplore) using combinations of keywords such as *"interpretability"*, *"explainability"*, *"XAI"*, *"mechanistic interpretability"*, *"agentic AI"*, *"LLM agents"*, and *"transparency"*. To capture both classical foundations and recent agentic systems, we focused primarily on work published between 2017–2025, while including a small number of earlier seminal papers when necessary.

Screening and eligibility. Titles and abstracts were screened to exclude clearly unrelated work (e.g., purely application-specific systems without transparency, low-level vision papers without interpretability, or non-AI agent frameworks). Full texts were then checked for alignment with our three focal areas: (i) interpretability methods, (ii) explainability and XAI, and (iii) agentic AI / LLM-based agents. We prioritised survey papers, foundational methods, and frameworks with clear relevance to transparency, rather than attempting exhaustive coverage of every application domain.

Inclusion and exclusion criteria. Papers were included if they: (1) addressed interpretability, explainability, or transparency in ML or LLMs; and/or (2) described agentic or tool-using LLM architectures with clear discussion of reasoning, planning, or decision-making. We excluded work that only mentioned explanations tangentially, papers with no technical detail on methods, and duplicate or superseded versions of the same study. This yielded a focused set of representative works that span classical XAI, mechanistic interpretability, and recent surveys on LLMs and agentic systems (cf. Table **??**).

Bibliographic analysis. From the final set of papers, we performed a light bibliographic analysis to group contributions into three lenses that structure the rest of this survey: (1) interpretability methods (model-facing, mechanistic, or concept-based), (2) explainability methods (user-facing rationales and artifacts), and (3) agentic AI frameworks (single- and multi-agent loops, tools, and workflows). This clustering informed the X-AXIOM taxonomy in Section 3, and ensured that the taxonomy is grounded in the existing literature rather than being purely conceptual.

Karan to give me visual and we put here

Finally, the screened and grouped set of papers provides the empirical basis for our proposed X-AXIOM taxonomy. In the next section, we synthesise these findings into a structured view that connects interpretability, explainability, and agentic behaviour in a single, coherent framework.

3 X-AXIOM Taxonomy

Building on the seminarl literature, we distill recurring patterns and design choices into the X-AXIOM taxonomy. This taxonomy organises existing work along three complementary lenses: explainability (X), interpretability (I), and agentic behaviour (A), and clarifies how current methods and systems populate this space.

Figure 2 structures Agentic AI transparency along three axes: *Cognitive Objects*, *Assurance and Evaluation Objectives*, and *Mechanisms*. We propose these axes, as they are aligned with the Agent AI lifecycle phases of design, process, and outcome. We distinguish *interpretability*, which provides mechanistic insight into internal states and computations [62], from *explainability*, which provides an audience-facing rationale grounded in evidence and policy [69]. The framework yields concrete artefacts such as a Minimal Explanation Packet (MEP) that contains plan rationales, evidence hashes, tool traces, and fairness or policy deltas.[2]

[2]The Assurance and Evaluation Objectives align with governance controls such as transparency, robustness, fairness, and

Figure 2: The X-AXIOM taxonomy organizes transparency along three axes: WHAT (Cognitive Objects), WHY (Assurance and Evaluation Objectives), and HOW (Mechanisms). These axes are aligned with WHEN, which refers to lifecycle stages of design, process, and outcome. Arrows indicate how mechanisms expose cognitive objects, how objectives evaluate them, and how outputs are surfaced at each stage.

This work complements prior taxonomies that focus on either XAI methods or agent workflows. X-AXIOM brings together what is exposed, why it is evaluated, how it is realized, and when it is surfaced, and turns these dimensions into concrete artefacts for both single-agent and multi-agent settings.

traceability in the EU AI Act, NIST AI RMF, and ISO/IEC 42001 [34, 84, 53].

3.1 Cognitive Objects (WHAT)

Cognitive objects are the internal states and interfaces of an agent that should be observable to support mechanistic tracing and accountability. In our setting, these include the agent's intent, beliefs, plans, memory, tool inputs and outputs, governing policies or guardrails, and realised outcomes. Making these units visible links decision logic to environmental feedback and provides concrete evidence for developer diagnostics and auditor review.

Existing work has examined many of these objects in isolation: plans and tool use in agent loops [140, 106], memory and persistent state [90], governance and policy controls [98], outcomes in interpretability and explainability [83], and mechanistic views of internal states [26, 127]. X-AXIOM groups these seven items into a single audited set of cognitive objects that are explicit targets of transparency, so they can be logged, measured, and assessed consistently across the agent lifecycle. If a cognitive object can influence an outcome, there should be an explicit design decision about whether and how it is exposed, logged, and protected (e.g., for privacy).

3.2 Assurance and Evaluation Objectives (WHY)

This axis defines the evaluation goals and acceptance criteria for transparency in Agentic AI. We focus on six objectives:

- **Faithfulness:** explanations reflect the system's actual computations and decision path.
- **Usefulness:** explanations are actionable for their audience, including users, developers, and auditors.
- **Compliance:** evidence supports legal, ethical, and organizational requirements.
- **Robustness:** explanations are stable under reasonable perturbations and adversarial tests.
- **Equity:** explanations and outcomes do not systematically disadvantage protected groups.
- **Auditability:** artefacts can be reproduced, verified, and traced over time.

These objectives consolidate themes discussed in earlier literature and standards. Faithfulness is central to interpretability and mechanistic accounts [83]. Usefulness reflects user-centred XAI and guidance on actionable explanations [135]. Compliance aligns with governance frameworks in the EU AI Act [34], NIST AI RMF [84], and ISO/IEC 42001 [53]. Robustness has been a central concern in attribution and perturbation-based evaluation, for example in analyses of LIME and SHAP [101, 71], as well as in broader stability testing. Equity connects to fairness auditing in explanations and outcomes [83]. Auditability is supported by reproducible artefacts, signed logs, and replayable traces [98].

In our implementation, we instantiate these objectives using agreement between the plan and the trace for faithfulness, task success and user-rated usability scores (e.g., the System Usability Scale) for usefulness, control mapping and evidence trails for compliance, stability under input perturbations for robustness, disparity metrics across cohorts for equity, and replay and signature pass rates for auditability. Taken together, these objectives encourage multi-objective evaluation of agentic systems, instead of relying solely on task accuracy or win rate.

3.3 Mechanisms (HOW)

Mechanisms are the technical and procedural means used to make the system transparent:

- **Intrinsic (self-explanatory):** native summaries or structures produced by the system, for example self-rationales, plan graphs, and program sketches.
- **Post hoc:** explanations derived after the fact, for example feature attribution, counterfactuals, prototypes, and example-based explanations.
- **Causal or mechanistic interpretability:** interventions on internal representations and circuits, for example activation patching and circuit discovery.

- **Operational:** instrumentation that records behaviour for verification, for example signed logs, execution replay, and input and output signing.
- **Social:** interactive mechanisms aligned with human communication, for example dialogue or role rationales, interactive critique, and revision.

Post hoc methods include attribution- and perturbation-based families such as LIME and SHAP [101, 71], counterfactual explanations [124], and prototype-based approaches [20]. Causal and mechanistic interpretability locates and tests internal features and circuits [26, 127]. Intrinsic rationales connect to chain-of-thought and self-explanation [131]. Operational mechanisms are supported by documentation and governance artefacts, such as model cards and datasheets, and by audit-oriented logging [**?** 39].

We group these mechanisms along a single HOW axis and link them to what is exposed and when it is surfaced. Typical risks include limited faithfulness for some post hoc methods and verbosity or audience mismatch for intrinsic rationales, while causal and operational mechanisms can be costly to deploy at scale. The objectives above target these risks by encouraging combinations of mechanisms rather than reliance on a single explaination method.

3.4 Temporal Stages (WHEN)

We structure transparency along three lifecycle stages. At design time, practitioners specify which cognitive artefacts will be exposed, select explanation mechanisms, and instrument logging and provenance so evidentiary requirements and verification procedures are explicit [123]. At process time, systems capture operational traces, including plans, tool calls, inputs and outputs, and updates to beliefs and memory, with timestamps, run identifiers, and replay hooks to support inspection and audit [98]. At outcome time, agents compile a Minimal Explanation Packet (MEP) that summarises the decision and its rationale, links to traces and evidence, records policy or fairness deviations, and is signed, timestamped, and archived with documented retention schedules. Packet contents are redacted for personal data and stored under access controls [78]. Across stages, we assess coverage and readiness at design time, completeness and integrity at process time, and at outcome time consistency with traces, usefulness, robustness, equity impacts, and replayability [80].

This stage-based view makes process-time instrumentation a first-class requirement: without rich traces of what the agent believed, planned, and did, outcome explanations risk becoming unverifiable narratives rather than evidence-backed accounts.

3.5 Multi-agent Add-ons (WHO)

As deployments evolve from single agents to distributed multi-agent systems, transparency must extend from individual decisions to coordination and shared accountability. We introduce role–rationale contracts that bind responsibilities to decision rationales, and attribution mappings at the team, agent, and tool levels. These mappings connect outcomes to contributing actors and artefacts and enable provenance graphs across agents using W3C PROV relations [123], supporting ecosystem-level auditing and clear escalation paths.

Case stub: tool-using LLM agent. At outcome time, the agent emits a Minimal Explanation Packet (MEP) that includes a plan graph with rejected alternatives, signed tool traces with inputs, outputs, and error handling, evidence hashes and retrieval identifiers, and recorded policy or fairness deltas. Evaluations include agreement between the plan and the trace for faithfulness, task success and user utility for usefulness, stability under input perturbations for robustness, parity of errors and explanations across cohorts for equity, and verification of signatures and replay for auditability. In multi-agent settings, such packets can be linked along the provenance graph to reconstruct team-level accountability for any given outcome.

4 Interpretability in Agentic AI (Design- & Process-time)

This section explains how to make agent internals understandable before a system runs and while it is running. The goal is to connect cognitive objects to concrete evidence so that developers and auditors can

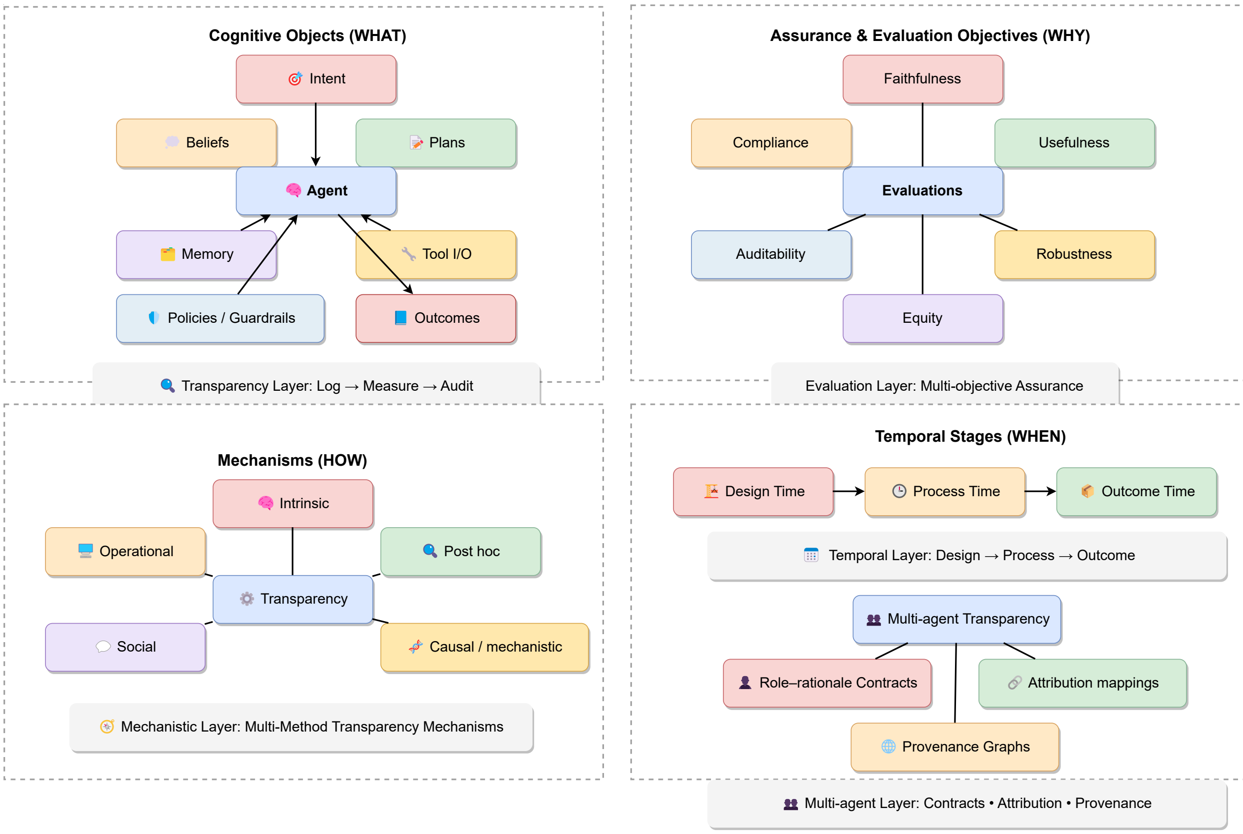

Figure 3: Overview of our transparency framework across five axes: Cognitive Objects (WHAT), Evaluation Objectives (WHY), Mechanisms (HOW), Temporal Stages (WHEN), and Multi Agent Add Ons (WHO). The framework organizes what is exposed, why it is assessed, how it is generated, when it appears in the lifecycle, and who is responsible in multi agent settings.

see how decisions arise and how they can be checked. We focus on what to expose, how to instrument it, and how to measure it at these two lifecycle stages.

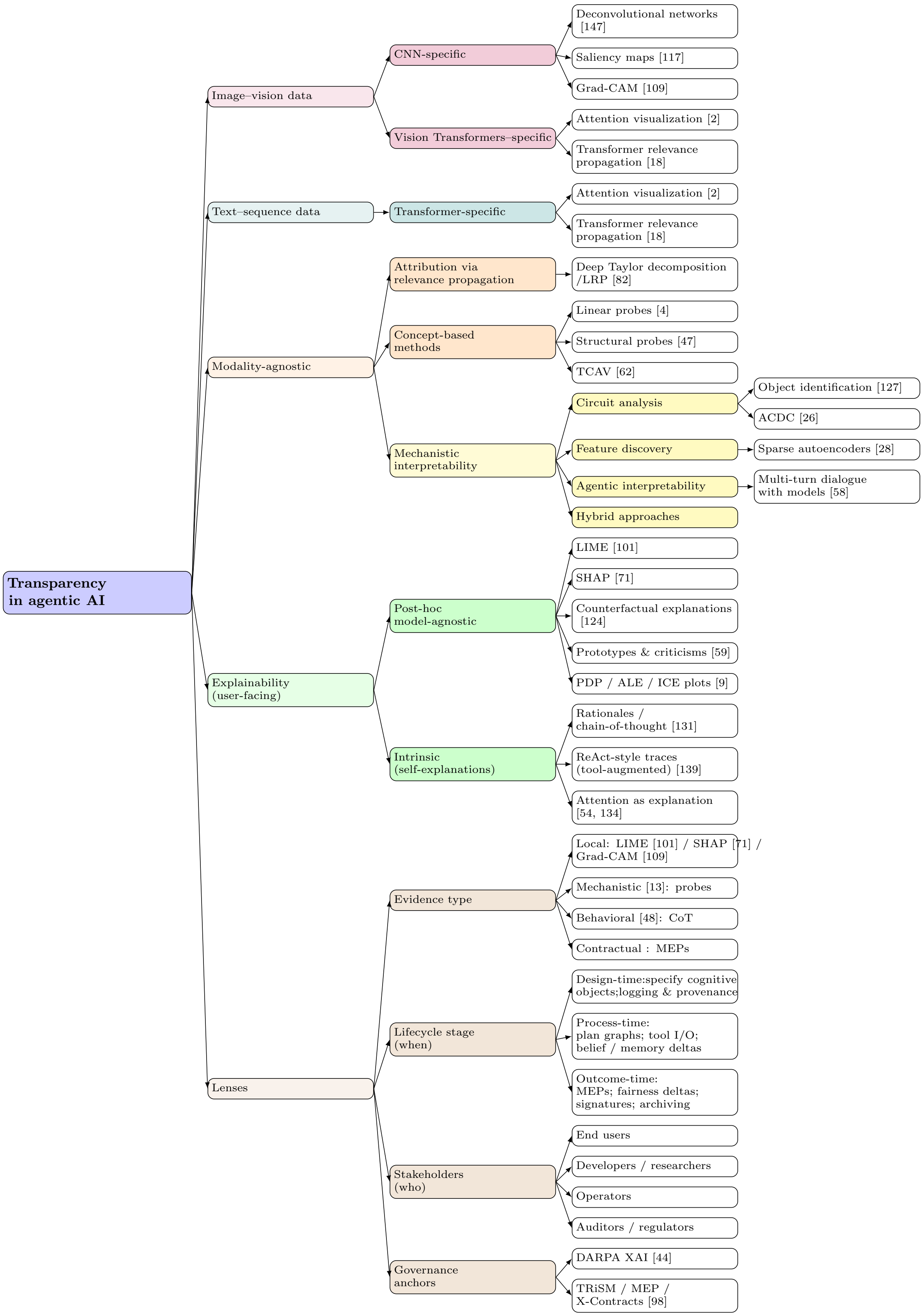

Figure 4: Taxonomy of transparency methods in agentic AI, distinguishing model-facing interpretability, user-facing explainability and cross-cutting lenses (evidence type, lifecycle stage, stakeholders and governance anchors).

4.1 Classical Interpretability

Classical interpretability refers to making opaque models more legible through rule extraction and surrogate models [119, 27]. The end goal is to make decision-making processes of models transparent and understandable to humans [100]. Lately, its development has evolved over several decades: early neural network research in the 1980s showed that multilayer perceptrons (MLPs) could learn meaningful internal representations through antisymmetric patterns and latent factor organization [104]. The 1990s introduced systematic "black box" solutions via rule extraction [119] and interpretable network mimics [27]. Later on, convolutional neural network (CNN) era brought feature visualization through deconvolutional networks [147] and gradient-based saliency maps [117], while Transformers demanded new attention-based methods [18]. From 2020, mechanistic interpretability emerged as bottom-up reverse engineering [13], using circuit analysis and sparse autoencoders to decompose computations. By 2024–2025, as systems become agentic, analyses must also track plans, tool calls, and memory updates across steps. A concise timeline is in Fig. 5.

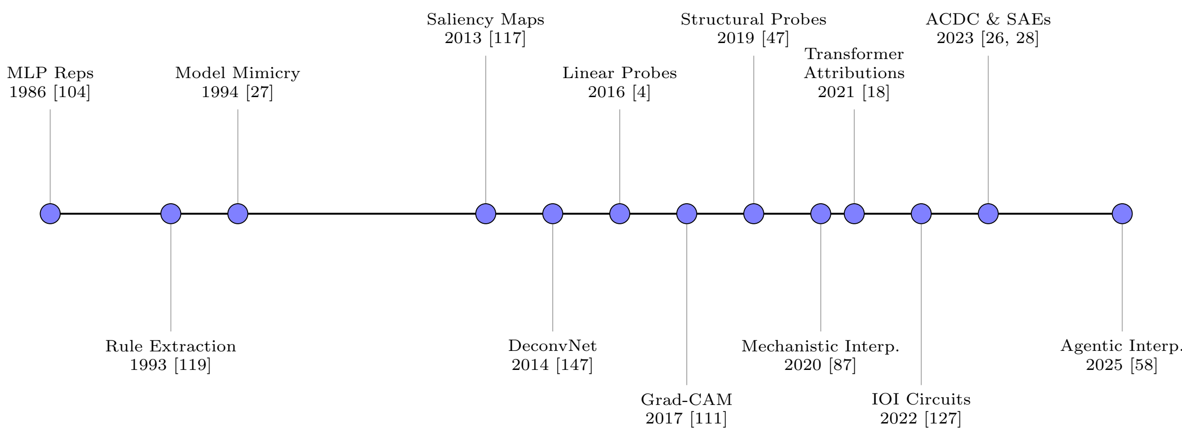

Figure 5: Timeline of key milestones in AI interpretability

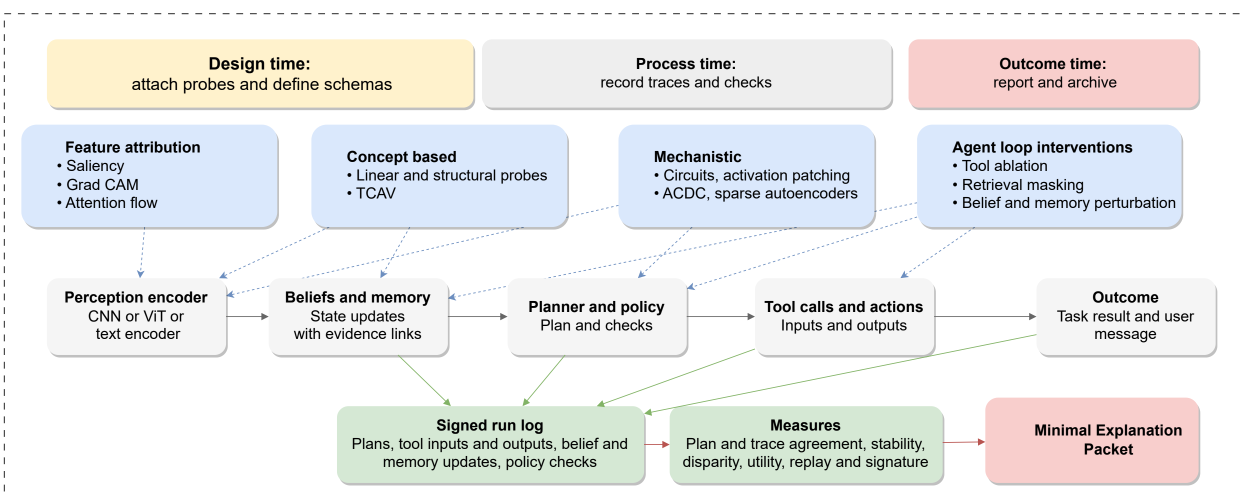

Figure 6: Where interpretability connects to the agent loop. Arrows indicate logged evidence: E1 retrieval, E2 tool I/O, E3 memory/belief updates, E4 actions. These traces support causal probes and, later, the Minimal Explanation Packet (MEP).

Table 4: Interpretability techniques mapped to agentic components (design-/process-time).

Paper	Method	Modality	Focus	Metrics (tags)	Agent layer
[147]	Deconvolutional nets	Image	Map feature activations back to pixels via unpooling/transpose conv	localization, plausibility, runtime	Perception
[117]	Gradient saliency	Image	Input sensitivity heatmaps	localization, plausibility, robustness	Perception
[111]	Grad-CAM	Image	Localization via gradients on conv feature maps	localization, plausibility, robustness	Perception
[2]	Attention flow/rollout	Text/Image	Token influence paths in Transformers	plausibility, correlation	Cross-layer
[82, 18]	LRP/DTD for Transformers	Agnostic	Class-specific relevance through attention/residual routes	faithfulness, localization	Cross-layer
[4]	Linear probes	Agnostic	Concept decodability from layer reps	predictability, selectivity	Reasoning/ Planning
[47]	Structural probes	Text	Geometric tests of syntax (depth/distance)	correlation, predictability	Reasoning/ Planning
[62]	TCAV	Agnostic	Sensitivity to human concepts via exemplars	concept-influence, robustness	Cross-layer
[116]	Concept activation (VLM)	Image+Text	Joint latent concepts via Semi-NMF decoding	concept-alignment, plausibility	Perception
[127]	Circuit analysis (IOI)	Agnostic	Sparse subgraphs for specific behaviors; causal tests	faithfulness, completeness, minimality	Reasoning/ Planning
[26]	ACDC	Agnostic	Automated causal subgraph discovery	faithfulness, completeness, minimality	Reasoning/ Planning
[28]	Sparse autoencoders	Agnostic	Interpretable features from activations	feature-quality, sparsity, faithfulness	Cross-layer
[58]	Agentic interpretability	Agnostic	Model–human dialogue to hypothesize mechanisms	usefulness, plausibility	Cross-layer

Note. IOI: indirect object identification. Tags: *faithfulness* (causal alignment), *completeness* (coverage), *minimality* (no superfluous parts), *robustness* (stability), *equity* (cohort parity), *auditability* (replay/signature), *plausibility* (human-judged sense).

4.2 Latest Methods (opening the model and the loop)

This section focuses on interpretability that opens the model itself (intrinsic or mechanistic). We discuss representative methods below:

Feature attribution approaches Feature attribution methods were first developed for CNNs. Where deconvolutional neural networks mapped learned features back to input pixels using transpose convolutions and unpooling [147]. Saliency maps provided a direct gradient based signal by differentiating the class score with respect to input pixels [117], though early maps were often noisy. Grad CAM improved localization by computing gradients on convolutional feature maps and combining them with importance weights from global average pooling [111].

With the rise of Transformers, attention visualization offered a first look at token focus patterns, for example attention rollout and attention flow [2]. Raw attention alone, however, can be insufficient for complex behavior. Relevance propagation from the layer-wise relevance propagation (LRP) and Deep Taylor family was adapted to Transformers to propagate class specific relevance through attention layers and skip connections [18]. These methods explain individual predictions but can be noisy, and they may not capture higher level computation in LLMs, where long range reasoning is involved. This motivates methods that look beyond pixels or tokens toward concepts and mechanisms.

Concept based methods Concept based methods address these limits by testing for higher level, human interpretable concepts in internal representations. Linear probes train simple classifiers on frozen activations to test whether a representation is sufficient for a given concept [4]. Structural probes examine whether geometric relationships in representation space encode linguistic structure such as dependency trees [47]. Testing with Concept Activation Vectors (TCAV) lets users define concepts with examples and measures the influence of those concepts on predictions [62]. These methods provide dataset level measures of concept use, though they often assume linear separability, depend on concept set quality, and can miss nonlinear structure. This sets the stage for mechanistic interpretability, which opens the model and its computations directly.

Mechanistic Interpretability Mechanistic interpretability seeks to reverse engineer the algorithms learned by networks. Circuit analysis isolates sparse computational subgraphs that drive specific behaviors, such as indirect object identification in language models [127]. Interventional tests, for example activation patching and ablations, check whether proposed components have a causal role. Automated methods scale this analysis: ACDC identifies causally relevant connections between components [26], and sparse autoencoders learn interpretable features from activations that support automated decomposition [28].

Agent-loop causal probes As systems become agentic and use tool calls and multi step reasoning, single pass analyses are not always enough. Agentic interpretability explores multi turn interactions where models help explain their own behavior for human understanding [58]. In practice, hybrid workflows combine mechanistic precision for auditing with agentic dialogue for day to day use, while treating dialogued explanations as hypotheses to be tested with causal probes.

4.3 Lifecycle integration (design- and process-time)

At **design-time**, we select cognitive objects to expose (intent, beliefs, plans, memory, tool I/O, policies, outcomes), define minimal schemas, and set provenance so runs produce consistent traces. At **process-time**, we log plans, tool calls, inputs/outputs, belief and memory updates, and policy checks with timestamps and run identifiers. These traces enable circuit-level probes and loop-level causal tests; detailed lifecycle metrics appear in §7.

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

In this section, we focus on outward-facing artifacts generated while the system runs and at outcome-time: rationales, counterfactuals, retrieval views, prototypes, and policy evidence. These artifacts are tied to assurance objectives (faithfulness, usefulness, compliance, robustness, equity, auditability) and feed into the MEP.

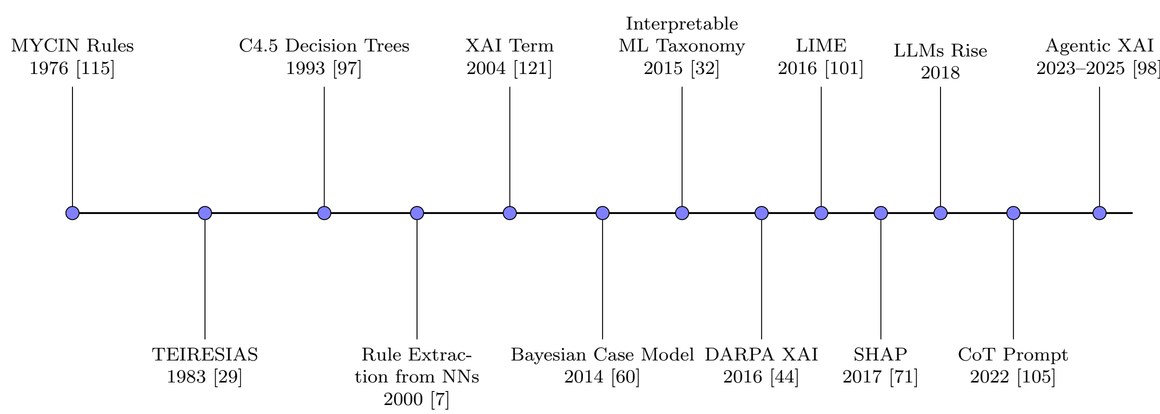

Figure 7: Timeline of key milestones in AI explainability, from rule-based systems to agentic XAI.

Table 5: User-facing explainability methods for agentic systems (process-/outcome-time).

Paper	Method	Modality	Artifact (what the user sees)	Metrics (tags)	Agent layer
[120]	Attention visualization (retrieval)	Text	Interactive attention maps over retrieved tokens/docs	usefulness, plausibility	Retrieval
[46]	Chain-of-thought with faithfulness scoring	Text+Retrieval	Stepwise rationales with per-step entailment checks	accuracy, faithfulness	Reasoning/ Planning
[63]	Counterfactual explanations	Tabular+Image	Minimal input changes yielding desired outcomes (recourse)	proximity, plausibility, robustness	Planning
[22]	Counterfactual simulatability	Text	Explanations evaluated by outcome simulation under "what-if"	simulation precision/generality	Planning
[126]	Layered chain-of-thought	Text	Layered reasoning segments with checks and feedback slots	usefulness, correctness	Reasoning
[76]	Human-centered prompting (x-plAIn)	Text	Audience-adapted rationales for actions/decisions	usefulness, user comprehension	Actuation
[99]	Self-reflection / critique	Text	Post-hoc error analysis and revised rationale	improvement, usefulness	Actuation
[131]	ReAct-style reasoning traces	Text+Env	Interleaved reasoning and actions (traceable decision record)	success rate, trace completeness	Reasoning+ Actuation
[12]	KG-based RAG explanations (KGRAG-Ex)	Text+KG	Structured retrieval paths with perturbation-based impact	impact counts, rank deltas	Retrieval
[139]	ReAct prompting (agents)	Text+Env	Step–action–observation triplets for transparency	success, exact match	Reasoning+ Actuation

5.1 Classical XAI

Classical XAI focused on producing artifacts that people can read and reason with. Early expert systems (e.g., MYCIN, NEOMYCIN) provided rule-based justifications for inference and strategy [115, 25]. Model-agnostic post hoc methods such as LIME and SHAP generate local feature attributions that explain individual predictions [101, 71]. Partial dependence plots (PDP) and accumulated local effects (ALE) summarize average feature influence at the dataset level [38, 9]. Complementary streams explored incorporating domain structure through evolutionary and gray-box optimization to improve interpretability of decision logic [30, 133, 75]. So, classical methods emphasize artifacts for users (plots, local explanations, rule traces). Low-level mechanistic probes (e.g., saliency internals) are treated in §4; here we use them only when they surface as user-facing visuals.

5.2 Latest explainability for LLMs and agentic systems

Modern systems extend user-facing explanations beyond static models to agents that retrieve, plan, act, and update state.

Intrinsic (self-explanation) during reasoning. LLMs can emit stepwise rationales via chain-of-thought; these should be treated as hypotheses and verified where possible [131]. Layered or structured variants add checkpoints and user feedback, while argumentation schemes frame claims, premises, and conclusions [49]. Self-reflection/critique ("Reflexion") revises earlier reasoning to improve subsequent decisions [114, 99]. These artifacts are directly readable and can be logged alongside evidence references.

Retrieval and attention views. When agents rely on retrieval, attention visualizations (e.g., BERTViz-style) expose which tokens or documents are emphasized for a given answer [120]. Structured RAG explanations add traceable evidence paths; knowledge-graph–based RAG can report perturbation impacts to indicate which nodes/edges mattered for the final decision [12].

Counterfactual and simulatability-based explanations. Counterfactuals provide recourse-style explanations: minimal, plausible input changes that would alter the outcome [124, 63]. Recent work evaluates explanations by whether users (or models) can *simulate* outcomes under counterfactuals, scoring precision and generality [22]. LLMs can also help verbalize counterfactuals for non-expert audiences [42].

Concept-centric artifacts. Concept bottleneck models expose intermediate, human-labeled concepts that explicitly mediate predictions [146]. TCAV measures sensitivity to user-defined concepts using example sets [141]. Neuron/dissection-style reports link units/features to concepts, and multimodal "neurons" demonstrate cross-modal alignment in VLMs [155, 43]. When discrete concepts are insufficient, impact-aware or latent concept attribution yields higher-level, human-interpretable descriptors [153, 144].

Surrogates, prototypes, and policy summaries. Surrogate models approximate complex policies to provide simpler, human-readable decision rules [21, 35]. Prototype-based methods explain predictions by similarity to learned exemplars in text or vision [45]. For agents, model-agnostic policy summaries present state–action regularities in natural language for audit and onboarding [5]. Time-series prototype encoders pair prototypical patterns with LLM-generated narratives [55].

Visual saliency as an end-user artifact In multimodal tasks (e.g., VQA), Grad-CAM–style heatmaps and related relevance propagation can be surfaced to users to show *where* the model looked [110, 19]. We treat these as user-facing artifacts only (the underlying attribution mechanics belong to §4). RISE offers a black-box variant via randomized masking [93].

Validator, not artifact Causal interpretability tools, such asactivation patching/ablations, circuit analysis, knowledge neurons, are primarily *validators* for explanation faithfulness and belong to §4. Here we use their outcomes only as *checks* attached to explanations (e.g., "this rationale survived ablation tests") [65, 130, 70].

Scoring and MEP linkage For each explainability method, we log the artifact (text, visualization, path), link it to evidence (trace, retrieval, or tool-call identifiers), and attach checks: faithfulness (against traces or causal probes), usefulness (task-specific user ratings), compliance (policy/documentation fields), robustness (perturbation stability), equity (cohort disparities with CIs), and auditability (signature and replay). These roll into the Minimal Explanation Packet (MEP) at outcome-time.

6 X-Contracts (Deployment-Time)

We introduce *X-Contracts* as deployment-time agreements that make transparency and evidence production first-class obligations of agentic systems. Each X-Contract specifies the *scope* (which classes of decisions are covered); *roles and accountability* (who produces, signs, and audits artifacts); *required artifacts*: centrally a Minimal Explanation Packet (MEP) containing a decision summary, a plan graph with rejected alternatives, signed tool I/O, evidence hashes, policy/fairness deltas, timestamps, and a unique run identifier; *integrity and access* (hashing, digital signatures, retention, and access control); *live checks* (e.g., plan–trace agreement and basic parity deltas); and *remediation* (flagging, human review, and rollback).

As it is situated at deployment time, X-Contracts complement pre-deployment reporting artifacts such as model cards, datasheets, and service factsheets [79, 39, 10] by *binding* verifiable, outcome-time evidence to each decision. Conceptually, they adopt the discipline of assurance cases, such as explicit claims backed by structured evidence [57], and operationalize it for explainability at runtime.

Contract semantics are programmatic, where a decision is not released unless the MEP is assembled, signed, and logged; live checks compute plan–trace agreement and parity deltas and compare them to thresholds; violations trigger remediation and are captured in the audit log. This release gating mirrors established practices in reliability-focused operations [14] and ties all evidence items to a provenance backbone. Each clause maps cleanly onto widely recognized governance controls, including NIST AI RMF functions for measurement and governance [85], transparency and record-keeping duties in the EU AI Act [**?**], and management-system requirements codified in ISO/IEC 42001 [53]; the mapping is descriptive rather than prescriptive, preserving portability across regulatory regimes. Figure 8 summarizes the mechanism by showing the flow from design-time specifications to outcome, the runtime release gate, and a truncated MEP instance; Section 7 then details the evaluation protocol (coverage, completeness, integrity, plan–trace agreement, parity, replayability, utility, and overhead) and reports empirical results.

(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 (truncated)

```
{
  "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 },
  "signatures":  {"agent":"sig:...", "operator":"sig:..." },
  "timestamps":  {"start":"...", "end":"..." }
}
```

Figure 8: **X-Contract** (A) flow from design-time specs to outcome; (B) release gate enforced at runtime; (C) required evidence object (MEP).

7 Evaluation Protocols for Explainable and Interpretable Agents

Evaluating transparency in agentic AI requires going beyond task accuracy to measure whether the system's reasoning, evidence, and safeguards behave as claimed. In this section, we describe evaluation protocols for *explainable and interpretable agents* across the agent lifecycle (design-time, process-time, and outcome-time), ensuring that every decision is supported by verifiable artefacts.

7.1 Design-time readiness

Design-time evaluation ensures that the system is properly prepared to support transparency before being deployed. 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 [96, 16]. The MEP schema is examined for completeness, internal consistency, and its ability to store all required fields [98].

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 [123, 53]. These checks are summarised in a readiness score representing the proportion of fields that are both populated and cryptographically signed. A high readiness score ensures that subsequent evaluations rest on verifiable evidence rather than ad-hoc or incomplete logging.

7.2 Process-time measures

At process-time, we evaluate explanations along five dimensions: *faithfulness*, *stability*, *usefulness*, *equity*, and *auditability*. As the system operates, we evaluate whether explanations remain aligned with the agent's actual behaviour along five dimensions that instantiate the WHY-axis objectives at runtime: faithfulness, stability (as robustness to benign perturbations), usefulness, equity, and auditability.

The first measure is *Faithfulness*, which examines how closely the realised execution trace matches the planned sequence. This follows established evaluation principles in interpretability and mechanistic attribution that emphasise alignment between explanations and true computational pathways [83, 18]. We complement this with simple causal probes, drawing on advances in mechanistic interpretability where targeted interventions on activations, components, or tool calls reveal whether highlighted elements play a causal role [26, 28].

Stability assesses whether small, non-substantive input variations produce markedly different explanations. Earlier studies on perturbation-based explanation methods have demonstrated the importance of robustness under benign changes [102, 71]. Stable explanations indicate that the system is not overly sensitive to surface-level noise.

Usefulness is evaluated by comparing performance with and without the presence of explanations. Prior work in user-centred XAI has shown that well-designed rationales can improve task completion, reduce cognitive burden, and assist in debugging [136, 89]. We therefore track whether explanations assist users in understanding system behaviour or identifying failures more effectively.

Equity examines whether both decisions and explanations behave consistently across demographic or otherwise protected groups. This follows recommendations in fairness and explanation audits that highlight the importance of parity in outcomes, interpretability quality, and user experience [80, 124]. Any disparities are recorded in the fairness fields of the MEP.

Finally, *Auditability* measures whether all 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 [39, 79]. We also measure runtime and token overhead to ensure that transparency remains practical for deployment.

7.3 Outcome-time evaluation

At the end of each run, the agent assembles a complete and cryptographically signed MEP that consolidates the decision summary, plan graph, execution trace, tool inputs and outputs, evidence hashes, and fairness deltas [98]. Before the system is permitted to release its final output, the MEP must satisfy a release gate. The gate checks that the agreement between the plan and the execution trace exceeds a required threshold, that the MEP is sufficiently complete, and that fairness deltas fall within acceptable bounds. These checks align with established principles of assurance cases and runtime governance [57, 85].

The gate also verifies hash alignment between raw logs and the MEP, confirms that all signatures are valid, and checks the temporal coherence of traces. If any requirement fails, the run is flagged for human review, and the output may be withheld or rolled back. This mechanism ensures that every released decision is backed by transparent and verifiable evidence, consistent with responsible deployment practices [6].

7.4 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 [88]. Along with the quantitative results, we publish configuration files, prompts, model parameters, and guardrail settings, following documentation conventions such as model cards and datasheets [79, 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.

7.5 Benchmarks and Metrics Coverage

To situate our protocol within the broader evaluation landscape, we reinterpret existing agent benchmarks through the lens of X-AXIOM's WHY axis, which defines six assurance objectives for transparency in agentic systems: usefulness, faithfulness, compliance, robustness, equity, and auditability. Rather than retaining legacy agent-evaluation taxonomies, typically organised around behaviours, capabilities, safety, or reliability [81, 142], we reorganise evaluation metrics through these assurance objectives. This transparency-first restructuring reframes diverse capability measures in terms of the evidence and guarantees they provide, integrating interpretability, governance, safety, robustness, and performance into a coherent, assurance-driven framework.

Drawing on recent evaluation suites for LLM-based agents, we map representative metrics from the literature, including planning quality, tool-call correctness, fairness audits, robustness checks, and provenance validation, to their corresponding assurance objectives. Table 7.5 summarises these mappings and highlights the diversity of behaviours that modern agents expose: reasoning steps, plans, memory updates, tool I/O, dialogue transitions, and final outcomes.

To provide additional structure, Table 7.5 further organises the metrics within each assurance objective into functional subcategories that reflect the practical behaviours agents exhibit. Within Usefulness, for instance, we distinguish between cognitive and reasoning usefulness (e.g., goal understanding, planning quality), task-performance usefulness (e.g., factual correctness, task completion), interaction and communication usefulness (e.g., instruction following, dialogue consistency), tool-use usefulness (e.g., tool discovery, chaining), and system-level efficiency (e.g., latency, resource consumption). Likewise, Compliance groups safety-relevant and policy-aligned behaviours, including harm avoidance, overreach control, and privacy protection, while Robustness encompasses stability under perturbations, resilience to tool failures. Equity aggregates fairness-related metrics spanning outcome disparities and evaluation-process parity, and Auditability captures evidence-centric measures such as traceability, replayability, maintainability, and human-in-the-loop compatibility. Faithfulness remains focused on alignment between plans, execution traces, and the system's underlying computational pathway.

A key observation from this synthesis is that current agent benchmarks remain highly fragmented. Most evaluate narrow operational behaviours such as tool invocation accuracy, instruction following, or task completion. Very few meaningfully assess process-level transparency, mechanistic faithfulness, policy compliance, or the verifiability of decision traces. Even fewer address multi-agent interactions, socio-technical risks, or lifecycle-aware governance demands such as reproducibility, evidence trails, and audit readiness. [TODO: Add the relevant references]

By grounding each metric in an explicit assurance objective, we provide a principled evaluation scaffold that complements existing capability-oriented benchmarks and offers a unified foundation for transparency-aligned assessment of Agentic AI systems.

Table 6: Evaluation Landscape Mapped to X-AXIOM Assurance Objectives

Category	**Aspect**	**Reported Metrics**	**What it Measures**	**Papers and Benchmarks**
Usefulness - Cognitive Reasoning	Goal Understanding	Intent classification accuracy	Accuracy of interpreting user intent	AgentBench (2024)
	Learning / Adaptation	Improvement across iterations	Improvement after feedback or failures	AutoGen (2023)
	Logical Reasoning	Multi-step reasoning accuracy	Multi-step inference quality	LEADERBOARD (ACL 2024)
	Memory Usage	Retrieval accuracy, recall rate	Recall and use of prior information	MemGPT (2024)
	Self Reflection	-	-	Reflection-Bench[66]
	Task Decomposition / Planning	Plan quality score	Ability to break down goals into coherent subtasks	ReAct (2023), TAU-Bench (2024)
Usefulness - Interaction	Explainability / Transparency	Trace clarity score	Clarity and readability of reasoning traces	TrustLLM (2024)
	Helpfulness	Human preference score	Human-perceived quality of responses	AlpacaEval 2 (2024)
	Instruction Following	Constraint adherence rate	Compliance with user instructions and constraints	Super-NI (2023)
	Multi-Turn Consistency	Context retention rate	Preservation of context across dialogue turns	ChatEval (2024)

Category	Aspect	Reported Metrics	What it Measures	Papers and Benchmarks
	Trust Calibration	Confidence accuracy, uncertainty calibration error	Appropriate expression of uncertainty and confidence	SafeBench (2024)
Usefulness - Task Performance	Consistency (pass^k)	Variation across runs	Determinism under repeated equivalent runs	Consistency-Bench (2024)
	Correctness / Precision	Factual accuracy, precision	Factual and computational accuracy	LM-as-Examiner (2023)
	Generalization	Out-of-distribution success rate	Transfer to unseen tasks and domains	BIG-Bench Hard (2023)
	Pass@k Reliability	Probability of success within k samples	Probability of success within k attempts	τ-Bench (2024)
	Task Completion	Completion rate	Fraction of tasks fully completed	TheAgentCompany, MCP-AgentBench (2025)
	Task Completion Under Constraints	Success under constraints	Success under explicit constraints	—
Usefulness - Tool Use	Adaptability / Transfer	Zero-shot tool success	Ability to adjust to new APIs/environments	W&B AgentEval (2025)
	Collaboration	Multi-agent coordination score	Multi-agent communication and coordination	Orq.ai MA-Eval (2025)
	Efficiency (Steps / Calls)	Steps-to-success, tool-call count	Minimizing redundant steps and tool calls	AgentBench (2024)
	Role Switching	Role-switch success rate	Ability to change roles in multi-agent systems	—
	Tool Discovery	Tool selection accuracy	Ability to identify and select the right tools	MCP-Bench (2025)
	Tool Success Rate	Successful call fraction	Fraction of successful tool calls	MCP-AgentBench (2025)
	Tool Invocation Accuracy	Schema correctness rate	Correct API calls, schemas, parameters	ToolEmu (ICLR 2024)
	Tool Chaining / Orchestration	Multi-step chain success	Sequencing and coordinating multiple tools	BrowserBench (2024)
Usefulness - System Efficiency	Cost / Resource Usage	Token cost, API cost	Token/compute/API cost of execution	HELM 2.0 (2023)
	Latency	Response time	Time to respond or complete tasks	Galileo.ai (2025)
	Carbon Emissions	CO_2-eq estimate	Environmental cost of execution	—
	Time Spent	End-to-end runtime	End-to-end execution time	—
Faithfulness	Plan Faithfulness	Deviation from plan	Degree to which execution matches the generated plan	-
	Reasoning Faithfulness	-	-	-
	Evidence Faithfulness	-	-	-
	Trace Faithfulness	-	-	-
Compliance	Data Protection	Privacy leakage rate	Avoiding private or sensitive data leakage	AgentDAM[154], PrivacyLens[129]
	Risk Awareness	Risk-detection accuracy	Ability to detect, identify, and judge safety risks in agent interactions	R-Judge[145]
	Safety and Harm Avoidance	Unsafe-action rate, Safe-action rate, Safety score, Harm Score	A measure of how consistently an agent avoids harmful, unsafe, or high-risk behaviors across safety-critical scenarios.	AgentHarm[8], ToolEmu[103], OpenAgentSafety[122], HARM[149], Agent-SafetyBench[151]
	Security Robustness	Attack success rate, Net resilient performance, Refuse rate, Adversarial prompt robustness, Benign Accuracy	Ability of an agent to withstand adversarial or malicious manipulation	AgentHarm[8], AgentPoison[23], AgentDojo[31], Agent Security Bench[148], PromptBench[156]
Robustness	Context Resilience	Robustness under context shifts	Stability under truncated or swapped context	LongBench (2023)
	Error Propagation	Error-containment rate	Ability to contain mistakes in multi-step tasks	ToolEmu (2024)
	Error Recovery	Self-correction rate	Ability to detect and correct own mistakes	TAU-Bench (2024)
	Hallucination	Hallucination rate	Fabrication of unsupported content	—
	Input Perturbation Tolerance	Robustness to noise	Stability under noise or paraphrasing	RobustBench (2024), HELM
	Long-Horizon Coherence	Long-horizon accuracy	Maintaining state and goals over long sequences	LongBench (2023)
	Tool Failure Robustness	Recovery after tool/API failures	Recovery after tool/API failures	OpenAgentSafety (2025)

Category	Aspect	Reported Metrics	What it Measures	Papers and Benchmarks
	Stochastic Stability	Variance across runs	Low variance across multiple runs	Consistency-Bench (2024)
Equity	Evaluation Fairness	Controlled-condition consistency	Identical testing conditions across models	MCP-Universe (2025)
	Response Fairness	Demographic parity score	Equal treatment across demographic groups	HolisticBias (2023)
	Violation Rate	Fairness-violation rate	Frequency of fairness violations	—
Auditability	Auditability	Reproducibility score	Ability to reproduce and verify agent behaviour	Docent (2024)
	Coverage / Redundancy	Metric overlap / coverage	Independence and completeness of metrics	AutoLibra (2025)
	Human-in-the-Loop Compatibility	Oversight success rate	Ease of human oversight and correction	SafeBench Interactive (2024)
	Maintainability	Update difficulty score	Ease of updating system components	—
	Task Diversity	Breadth score	Breadth and balance of task types	HELM 2.0 (2023)
	Traceability	Evidence completeness score	Availability of logs, traces, and evidence artifacts	HAL-Harness (2024)

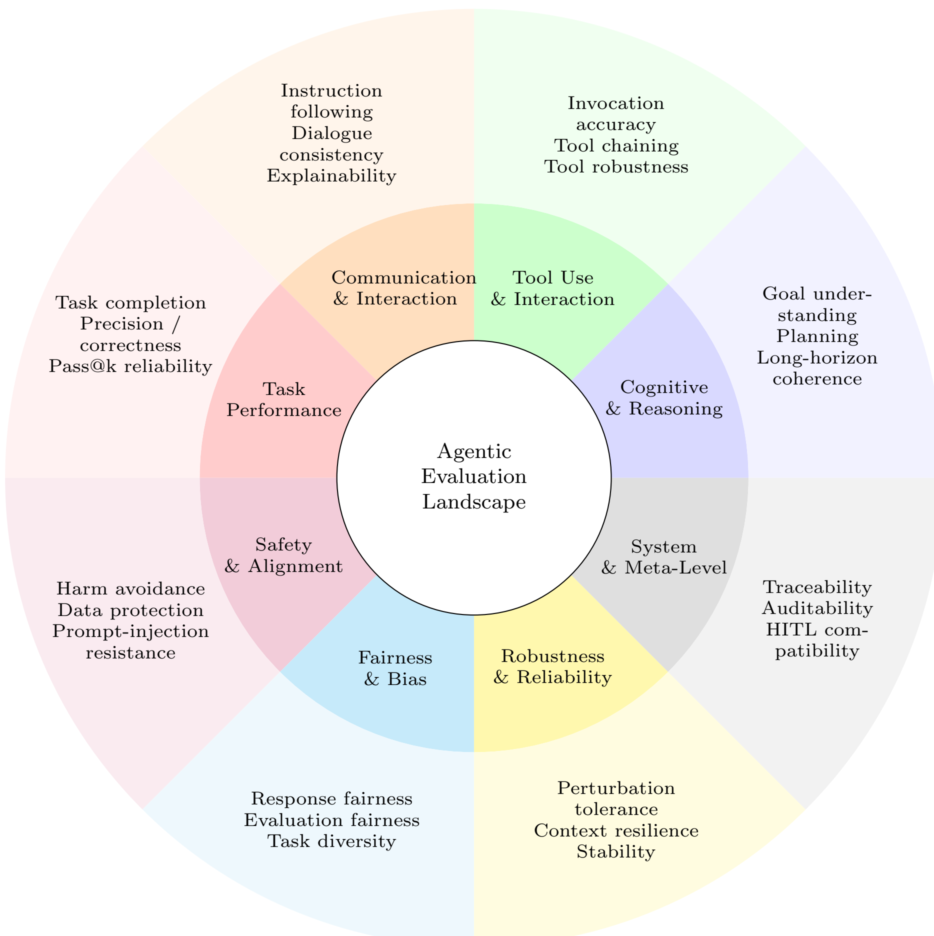

Figure 9: View of the agentic evaluation landscape. The inner ring groups metrics into eight functional categories; the outer ring lists representative metrics for each category, which are instantiated by concrete benchmarks in Table 7.5.

8 Open Challenges & Future Work

9 Conclusion

References

[1] Autogpt. `https://agpt.co/`. Accessed: 2025-08-12.

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

[3] Constructions Aeronautiques, Adele Howe, Craig Knoblock, ISI Drew McDermott, Ashwin Ram, Manuela Veloso, Daniel Weld, David Wilkins Sri, Anthony Barrett, Dave Christianson, et al. Pddl—the planning domain definition language. *Technical Report, Tech. Rep.*, 1998.

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

[5] Diego Alvarez et al. Model-agnostic policy explanations with large language models. *arXiv preprint arXiv:2504.05625*, 2025.

[6] Silvio Andrae. Governance of ai agents: Challenges, models, and regulatory approaches. In *Advancements in Multi-Agent Large Language Model Systems for Next-Generation AI*, pages 149–188. IGI Global Scientific Publishing, 2026.

[7] 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.

[8] 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.

[9] Daniel W Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(4):1059–1086, 2020.

[10] Matthew Arnold, Rachel K. E. Bellamy, Michael Hind, Stephanie Houde, Sameep Mehta, Aleksandra Mojsilovic, Ravi Nair, Karthikeyan Natesan Ramamurthy, Darrell Reimer, Alexandra Olteanu, David Piorkowski, Jason Tsay, and Kush R. Varshney. Factsheets: Increasing trust in ai services through supplier's declarations of conformity, 2019.

[11] 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.

[12] Georgios Balanos, Evangelos Chasanis, Konstantinos Skianis, and Evaggelia Pitoura. Kgrag-ex: Explainable retrieval-augmented generation with knowledge graph-based perturbations. *arXiv preprint arXiv:2507.08443*, 2025. Available at: `https://arxiv.org/abs/2507.08443`.

[13] Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*, 2024.

[14] Betsy Beyer, Chris Jones, Jennifer Petoff, and Niall Richard Murphy. *Site reliability engineering: how Google runs production systems.* " O'Reilly Media, Inc.", 2016.

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

[16] Stephen Casper, Luke Bailey, Rosco Hunter, Carson Ezell, Emma Cabalé, Michael Gerovitch, Stewart Slocum, Kevin Wei, Nikola Jurkovic, Ariba Khan, Phillip J. K. Christoffersen, A. Pinar Ozisik, Rakshit Trivedi, Dylan Hadfield-Menell, and Noam Kolt. The ai agent index, 2025.

[17] Alan Chan, Carson Ezell, Max Kaufmann, Kevin Wei, Lewis Hammond, Herbie Bradley, Emma Bluemke, Nitarshan Rajkumar, David Krueger, Noam Kolt, et al. Visibility into ai agents. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 958–973, 2024.

[18] 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.

[19] 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 (CVPR)*, pages 782–791, June 2021.

[20] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. This looks like that: Deep learning for interpretable image recognition, 2019.

[21] Han Chen et al. Large language models as surrogate models in evolutionary algorithms. *arXiv preprint arXiv:2406.10675*, 2024.

[22] Yanda Chen, Rui Li, Chen Tan, and Yulia Tsvetkov Wang. Do models explain themselves? counterfactual simulatability of natural language explanations. *arXiv preprint arXiv:2307.08678*, 2023.

[23] 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.

[24] Sanjoy Chowdhury, Mohamed Elmoghany, Yohan Abeysinghe, Junjie Fei, Sayan Nag, Salman Khan, Mohamed Elhoseiny, and Dinesh Manocha. Magnet: A multi-agent framework for finding audio-visual needles by reasoning over multi-video haystacks. *arXiv preprint arXiv:2506.07016*, 2025.

[25] William J Clancey and Reed Letsinger. *NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching.* Department of Computer Science, Stanford University Stanford, 1982.

[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] 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.

[28] 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.

[29] Randall Davis. Expert systems: Where are we? and where do we go from here? *AI Magazine*, 4(2):3–22, 1983.

[30] Kalyanmoy Deb. An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering*, 186(2-4):311–338, 2000.

[31] Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents. *Advances in Neural Information Processing Systems*, 37:82895–82920, 2024.

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

[33] Amirouche El Hassouni et al. Explainable artificial intelligence: A survey of needs, techniques, and future directions, 2024.

[34] European Parliament and Council of the European Union. Regulation (eu) 2024/1689 of the european parliament and of the council: Artificial intelligence act. https://eur-lex.europa.eu/eli/reg/2024/1689/oj, 06 2024. Official Journal of the European Union.

[35] Tianyi Fang et al. Internagent: When agent becomes the scientist. *arXiv preprint arXiv:2505.16938*, 2025.

[36] Adam Fourney, Ahmed Awadallah, Cheng Tan, Erkang Zhu, Friederike Niedtner, Gagan Bansal, and *et al.* Autogen v0.4: Reimagining the foundation of agentic ai for scale, extensibility, and robustness. https://www.microsoft.com/en-us/research/blog/autogen-v0-4-reimagining-the-foundation-of-agentic-ai-for-scale-extensibility-and-robustness/, January 2025. Microsoft Research Blog.

[37] Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, et al. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468*, 2024.

[38] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[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] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated planning: theory and practice*. Elsevier, 2004.

[41] 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.

[42] Flavio Giorgi, Cesare Campagnano, Fabrizio Silvestri, and Gabriele Tolomei. Natural language counterfactual explanations for graphs using large language models. *arXiv preprint arXiv:2410.09295*, 2024. Accessed: 2025-09-08 17:00 EDT.

[43] Gabriel Goh et al. Multimodal neurons in clip. *arXiv preprint arXiv:2204.10965*, 2022.

[44] David Gunning. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), Program Information*, 2017.

[45] Rohan Gupta et al. Protosure: Prototype-based explanations for llms. *arXiv preprint arXiv:2505.18970*, 2025.

[46] Hangfeng He, Hongming Zhang, and Dan Roth. Rethinking with retrieval: Faithful large language model inference. *arXiv preprint arXiv:2301.00303*, 2022.

[47] 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, Volume 1 (Long and Short Papers)*, pages 4129–4138, 2019.

[48] Henry Hexmoor, Johan Lammens, Guido Caicedo, and Stuart C Shapiro. *Behaviour based AI, cognitive processes, and emergent behaviors in autonomous agents*, volume 1. WIT Press, 2025.

[49] Shengxin Hong, Liang Xiao, Xin Zhang, and Jianxia Chen. Argmed-agents: Explainable clinical decision reasoning with large language models via argumentation schemes. *CoRR*, 2024.

[50] Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Bowei Xia, Tao Sun, Ziyu Ye, Zhaoxuan Jin, Yingru Li, Qiguang Chen, et al. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation. *arXiv preprint arXiv:2505.23885*, 2025.

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

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

[53] 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`.

[54] 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.

[55] Yushan Jiang, Wenchao Yu, Geon Lee, Dongjin Song, Kijung Shin, Wei Cheng, Yanchi Liu, and Haifeng Chen. Explainable multi-modal time series prediction with llm-in-the-loop. *arXiv preprint arXiv:2503.01013*, 2025. Accessed: 2025-09-08 17:00 EDT.

[56] Cristian Jimenez-Romero, Adrian Johnson, and Christian Blum. Multi-agent systems powered by large language models: Applications in swarm intelligence. *arXiv preprint arXiv:2503.03800*, 2025.

[57] Tim Kelly and Rob Weaver. The goal structuring notation–a safety argument notation. In *Proceedings of the dependable systems and networks 2004 workshop on assurance cases*, volume 6. Citeseer Princeton, NJ, 2004.

[58] 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.

[59] Been Kim, Rajiv Khanna, and Oluwasanmi Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[60] Been Kim, Cynthia Rudin, and Julie Shah. Bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

[61] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav), 2018.

[62] 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.

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

[64] 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.

[65] Jian Li et al. Exploring look-ahead planning mechanistic interpretability. *arXiv preprint arXiv:2406.16033*, 2024.

[66] 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.

[67] Zihao Lin, Samyadeep Basu, Mohammad Beigi, Varun Manjunatha, Ryan A Rossi, Zichao Wang, Yufan Zhou, Sriram Balasubramanian, Arman Zarei, Keivan Rezaei, et al. A survey on mechanistic interpretability for multi-modal foundation models. *arXiv preprint arXiv:2502.17516*, 2025.

[68] 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.

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

[70] Xiang Liu et al. Localized knowledge editing via causal neuron intervention. *arXiv preprint arXiv:2504.14496*, 2025.

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

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

[73] 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.

[74] Ričards Marcinkevičs and Julia E Vogt. Interpretability and explainability: A machine learning zoo mini-tour. *arXiv preprint arXiv:2012.01805*, 2020.

[75] Fischetti Martina, Matteo Fischetti, et al. Matheuristics. In *Handbook of Heuristics*, volume 1, pages 121–153. Springer International Publishing, 2018.

[76] Philip Mavrepis et al. x-plain: Human-centered prompting for explainable artificial intelligence. *arXiv preprint arXiv:2401.13110*, 2024.

[77] Michela Milano. Constraint programming links with math programming. *Wiley Encyclopedia of Operations Research and Management Science*, 2011.

[78] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[79] 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.

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

[81] Mahmoud Mohammadi, Yipeng Li, Jane Lo, and Wendy Yip. Evaluation and benchmarking of llm agents: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6129–6139, 2025.

[82] 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.

[83] 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.

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

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

[86] 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.

[87] 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.

[88] 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.

[89] 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.

[90] 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.

[91] Mihir Parmar, Xin Liu, Palash Goyal, Yanfei Chen, Long Le, Swaroop Mishra, Hossein Mobahi, Jindong Gu, Zifeng Wang, Hootan Nakhost, et al. Plangen: A multi-agent framework for generating planning and reasoning trajectories for complex problem solving. *arXiv preprint arXiv:2502.16111*, 2025.

[92] Gilles Pesant. A constraint programming primer. *EURO Journal on Computational Optimization*, 2(3):89–97, 2014.

[93] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models, 2018.

[94] 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.

[95] Aske Plaat, Max van Duijn, Niki van Stein, Mike Preuss, Peter van der Putten, and Kees Joost Batenburg. Agentic large language models, a survey. *arXiv preprint arXiv:2503.23037*, 2025.

[96] 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.

[97] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[98] 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.

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

[100] Mauricio Reyes, Raphael Meier, Sérgio Pereira, Carlos A Silva, Fried-Michael Dahlweid, Hendrik von Tengg-Kobligk, Ronald M Summers, and Roland Wiest. On the interpretability of artificial intelligence in radiology: challenges and opportunities. *Radiology: artificial intelligence*, 2(3):e190043, 2020.

[101] 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).

[102] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1135–1144. ACM, 2016.

[103] 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.

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

[105] Manish Sanwal. Layered chain-of-thought prompting for multi-agent llm systems: A comprehensive approach to explainable large language models. *arXiv preprint arXiv:2501.18645*, 2025.

[106] 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.

[107] 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.

[108] Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using llm agents as research assistants. *arXiv preprint arXiv:2501.04227*, 2025.

[109] 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.

[110] 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. *International Journal of Computer Vision*, 128(2):336–359, October 2019.

[111] 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.

[112] Tamar Rott Shaham, Sarah Schwettmann, Franklin Wang, Achyuta Rajaram, Evan Hernandez, Jacob Andreas, and Antonio Torralba. A multimodal automated interpretability agent. In *Forty-first International Conference on Machine Learning*, 2024.

[113] Sahil Sharma, Puneet Mittal, Mukesh Kumar, and Vivek Bhardwaj. The role of large language models in personalized learning: a systematic review of educational impact. *Discover Sustainability*, 6(1):1–24, 2025.

[114] 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.

[115] Edward H Shortliffe. *Computer-based medical consultations: MYCIN*. Elsevier, 1976.

[116] 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.

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

[118] NovelSeek Team, Bo Zhang, Shiyang Feng, Xiangchao Yan, Jiakang Yuan, Zhiyin Yu, Xiaohan He, Songtao Huang, Shaowei Hou, Zheng Nie, et al. Novelseek: When agent becomes the scientist–building closed-loop system from hypothesis to verification. *arXiv preprint arXiv:2505.16938*, 2025.

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

[120] Jesse Vig. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1904.02679*, 2019.

[121] Luca Viganò and Daniele Magazzeni. Explainable security, 2018.

[122] 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.

[123] 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/`.

[124] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[125] Ferdinand Wagner, Ruedi Schmuki, Thomas Wagner, and Peter Wolstenholme. *Modeling software with finite state machines: a practical approach*. CRC Press, 2006.

[126] 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.

[127] 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.

[128] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

[129] 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.

[130] Zhen Wang et al. Mechanistic interpretability with activation patching. *arXiv preprint arXiv:2407.11215*, 2024.

[131] 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.

[132] Xinming Wei, Jiahao Zhang, Haoran Li, Jiayu Chen, Rui Qu, Maoliang Li, Xiang Chen, and Guojie Luo. Agent. xpu: Efficient scheduling of agentic llm workloads on heterogeneous soc. *arXiv preprint arXiv:2506.24045*, 2025.

[133] L Darrell Whitley, Francisco Chicano, and Brian W Goldman. Gray box optimization for mk landscapes (nk landscapes and max-ksat). *Evolutionary computation*, 24(3):491–519, 2016.

[134] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 11–20. ACL, 2019.

[135] Xuansheng Wu, Haiyan Zhao, Yaochen Zhu, Yucheng Shi, Fan Yang, Lijie Hu, Tianming Liu, Xiaoming Zhai, Wenlin Yao, Jundong Li, Mengnan Du, and Ninghao Liu. Usable xai: 10 strategies towards exploiting explainability in the llm era, 2025.

[136] 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.

[137] 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.

[138] John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *arXiv preprint*, 2024.

[139] Shunyu Yao et al. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[140] Shunyu Yao, Dian Yang, Panupong Cui, and Karthik Narasimhan. React: Synergizing reasoning and acting in language models. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

[141] Junjie Ye et al. Recommendation for effective use of concept activation vectors. *arXiv preprint arXiv:2404.03713*, 2024.

[142] Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. Survey on evaluation of llm-based agents. *arXiv preprint arXiv:2503.16416*, 2025.

[143] 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.

[144] Xuemin Yu, Fahim Dalvi, Nadir Durrani, Marzia Nouri, and Hassan Sajjad. Latent concept-based explanation of nlp models. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. Accessed: 2025-09-08 16:48 EDT.

[145] 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.

[146] Mert Yuksekgonul et al. Concept bottleneck large language models (cb-llms) for inherently interpretable llms. *arXiv preprint arXiv:2412.07992*, 2024.

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

[148] Hanrong Zhang, Jingyuan Huang, Kai Mei, Yifei Yao, Zhenting Wang, Chenlu Zhan, Hongwei Wang, and Yongfeng Zhang. Agent security bench (asb): Formalizing and benchmarking attacks and defenses in llm-based agents. *arXiv preprint arXiv:2410.02644*, 2024.

[149] Jinchuan Zhang, Yan Zhou, Yaxin Liu, Ziming Li, and Songlin Hu. Holistic automated red teaming for large language models through top-down test case generation and multi-turn interaction. *arXiv preprint arXiv:2409.16783*, 2024.

[150] Kechi Zhang, Jia Li, Ge Li, Xianjie Shi, and Zhi Jin. Codeagent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges. *arXiv preprint arXiv:2401.07339*, 2024.

[151] Zhexin Zhang, Shiyao Cui, Yida Lu, Jingzhuo Zhou, Junxiao Yang, Hongning Wang, and Minlie Huang. Agent-safetybench: Evaluating the safety of llm agents. *arXiv preprint arXiv:2412.14470*, 2024.

[152] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38, 2024.

[153] Ruochen Zhao, Tan Wang, Yongjie Wang, and Shafiq Joty. Impact-aware concept-based explanations for large language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, 2024. Accessed: 2025-09-08 16:31 EDT.

[154] 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.

[155] Yifan Zhou et al. Neuronscribe: Dissecting multimodal neurons in llms. *arXiv preprint arXiv:2508.15875*, 2025.

[156] 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.