

# Intelligent AI Delegation

Nenad Tomašev<sup>1</sup>, Matija Franklin<sup>1</sup> and Simon Osindero<sup>1</sup>

<sup>1</sup>Google DeepMind

AI agents are able to tackle increasingly complex tasks. To achieve more ambitious goals, AI agents need to be able to meaningfully decompose problems into manageable sub-components, and safely delegate their completion across to other AI agents and humans alike. Yet, existing task decomposition and delegation methods rely on simple heuristics, and are not able to dynamically adapt to environmental changes and robustly handle unexpected failures. Here we propose an adaptive framework for *intelligent AI delegation* - a sequence of decisions involving task allocation, that also incorporates transfer of authority, responsibility, accountability, clear specifications regarding roles and boundaries, clarity of intent, and mechanisms for establishing trust between the two (or more) parties. The proposed framework is applicable to both human and AI delegators and delegatees in complex delegation networks, aiming to inform the development of protocols in the emerging agentic web.

*Keywords:* AI, agents, LLM, delegation, multi-agent, safety

## 1. Introduction

As advanced AI agents evolve beyond query-response models, their utility is increasingly defined by how effectively they can decompose complex objectives and delegate sub-tasks. This coordination paradigm underpins applications ranging from personal use, where AI agents can act as personal assistants (Gabriel et al., 2024), to commercial, enterprise deployments where AI agents can provide support and automate workflows (Huang and Hughes, 2025; Shao et al., 2025; Tupe and Thube, 2025). Large language models (LLMs) have already shown promise in robotics (Li et al., 2025a; Wang et al., 2024a), by enabling more interactive and accurate goal specification and feedback. Recent proposals have also highlighted the possibility of large-scale AI agent coordination in virtual economies (Tomasev et al., 2025). Modern agentic AI systems implement complex control flows across differentiated sub-agents, coupled with centralized or decentralized orchestration protocols (Hong et al., 2023; Rasal and Hauer, 2024; Song et al., 2025; Zhang et al., 2025a). This can already be seen as a sort of a microcosm of task decomposition and delegation, where the process is hard-coded and highly constrained. Managing dynamic web-scale interactions requires us to think beyond the approaches that are currently employed by more

heuristic multi-agent frameworks.

Delegation (Castelfranchi and Falcone, 1998) is more than just task decomposition into manageable sub-units of action. Beyond the creation of sub-tasks, delegation necessitates the assignment of responsibility and authority (Mueller and Vogelsmeier, 2013; Nagia, 2024) and thus implicates accountability for outcomes. Delegation thus involves risk assessment, which can be moderated by trust (Griffiths, 2005). Delegation further involves capability matching and continuous performance monitoring, incorporating dynamic adjustments based on feedback, and ensuring completion of the distributed task under the specified constraints. Current approaches tend to fail to account for these factors, relying more on heuristics and/or simpler parallelization. This may be sufficient for early prototypes, but real world AI deployments need to move beyond ad hoc, brittle, and untrustworthy delegation. There is a pressing need for systems that can dynamically adapt to changes (Acharya et al., 2025; Hauptman et al., 2023) and recover from errors. The absence of adaptive and robust deployment frameworks remains one of the key limiting factors for AI applications in high-stakes environments.

To fully utilize AI agents, we need *intelligent delegation*: a robust framework centered around

clear roles, boundaries, reputation, trust, transparency, certifiable agentic capabilities, verifiable task execution, and scalable task distribution. Here we introduce an intelligent task delegation framework aimed at addressing these limitations, informed by historical insights from human organizations, and grounded in key agentic safety requirements.

## 2. Foundations of Intelligent Delegation

### 2.1. Definition

We define *intelligent delegation* as a sequence of decisions involving task allocation, that also incorporates transfer of authority, responsibility, accountability, clear specifications regarding roles and boundaries, clarity of intent, and mechanisms for establishing trust between the two (or more) parties. Complex tasks may also involve steps pertaining to task decomposition, as well as careful capability lookup and matching to inform allocation decisions.

When we refer to task delegation we normally presume that the tasks exceed some basic level of complexity that would be handled by a system subroutine – such rudimentary outsourcing still requires care, but it is far more limited in scope. At the other end of the spectrum, it may be possible to contract with agents that are granted full autonomy, and can freely pursue any number of sub-goals without explicit checks and permissions (Kasirzadeh and Gabriel, 2025). In the limit case, such fully autonomous agents would need to be trusted with moral decisions (Sloksnath, 2025), though this may not be something we ever choose to permit as contemporary agents are severely lacking in their capacity to engage in such decisions (Haas, 2020; Mao et al., 2023; Reinecke et al., 2023). We consider such an open-ended scenario to be in scope for our discussion, though only insofar as the appropriate mechanisms can be put in place to ensure safety of more autonomous task completion.

### 2.2. Aspects of Delegation

As delegation can take different forms, here we introduce several axes that help us contextualize these use cases and make them more amenable to analysis.

1. **Delegator.** Human or AI.
2. **Delegatee.** Human or AI.
3. **Task characteristics.**
  - (a) **Complexity.** The degree of difficulty inherent in the task, often correlated with the number of sub-steps and the sophistication of reasoning required.
  - (b) **Criticality.** The measure of the task's importance and the severity of consequences associated with failure or sub-optimal performance.
  - (c) **Uncertainty.** The level of ambiguity regarding the environment, inputs, or the probability of successful outcome achievement.
  - (d) **Duration.** The expected time-frame for task execution, ranging from instantaneous sub-routines to long-running processes spanning days or weeks.
  - (e) **Cost.** The economic or computational expense incurred to execute the task, including token usage, API fees, and energy consumption.
  - (f) **Resource Requirements.** The specific computational assets, tools, data access permissions, or human capabilities necessary to complete the task.
  - (g) **Constraints.** The operational, ethical, or legal boundaries within which the task must be executed, limiting the solution space.
  - (h) **Verifiability.** The relative difficulty and cost associated with validating the task outcome. Tasks with high verifiability (e.g., formal code verification, mathematical proofs) allow for “trustless” delegation or automated checking. Conversely, tasks with low verifiability (e.g., open-ended research) require high-trust delegates or expensive, labor-intensive oversight.
  - (i) **Reversibility.** The degree to which the

effects of the task execution can be undone. Irreversible tasks that produce side effects in the real world (e.g., executing a financial trade, deleting a database, sending an external email) require stricter *liability firebreaks* and steeper authority gradients than reversible tasks (e.g., drafting an email, flagging a database entry).

- (j) **Contextuality.** The volume and sensitivity of external state, history, or environmental awareness required to execute the task effectively. High-context tasks introduce larger privacy surface areas, whereas context-free tasks can be more easily compartmentalized and outsourced to lower-trust nodes.
  - (k) **Subjectivity.** The extent to which the success criteria are a matter of preference versus objective fact. Highly subjective tasks (e.g., "design a compelling logo") typically require "Human-as-Value-Specifier" intervention and iterative feedback loops, whereas objective tasks can be governed by stricter, binary contracts.
4. **Granularity.** The request could involve either fine-grained or course-grained objectives. In the course-grained case, the delegatee may need to perform further task decomposition.
  5. **Autonomy.** Task delegation may involve requests that grant full autonomy in pursuing sub-tasks, or be far more specific and prescriptive.
  6. **Monitoring.** For delegated tasks, monitoring could be continuous, periodic, or event-triggered.
  7. **Reciprocity.** While delegation is usually a one-way request, there could be cases of mutual delegation in collaborative agent networks.

Starting with the delegator and delegatee axes, it is possible to consider the following scenarios: 1) human delegates to an AI agent 2) AI agent delegates to an AI agent 3) AI agent delegates to a human (Ashton and Franklin, 2022; Guggenberger et al., 2023). While the first case has ar-

guably been discussed the most in literature, the other two are just as relevant to consider. The increasing number of AI agents being deployed across systems, coupled with the development of infrastructure for setting up virtual agentic markets and economies (Hadfield and Koh, 2025; Tomasev et al., 2025; Yang et al., 2025), makes it clear that there would be far more agent-agent interactions in the future, and those would likely also involve task delegation.

Delegation between agents may either be hierarchical or non-hierarchical, depending on the relationship between agents and their respective roles within the network. An example of a hierarchical relationship would be an orchestrator agent that delegates a task to a sub-agent within the collective. A non-hierarchical relationship would involve peer agents with equal standing. An advanced AI agent could also delegate a task to a specialist ML model, without any notable agency.

AI-human delegation (Guggenberger et al., 2023) has been shown to be a promising paradigm (Hemmer et al., 2023), making it easier to successfully collaborate with super-human systems (Fügener et al., 2022), due to differences in cognitive biases and metacognition (Fügener et al., 2019). Davidson and Hadshar (2025) predict that there will be an increase in "AI-directed human labour," which may significantly increase economic productivity. In practice, present day AI-human delegation comes with a set of issues. Algorithmic management systems in ride-hailing and logistics allocate and sequence tasks, set performance metrics, and enforce behavioural norms through data-driven decision-making, effectively delegating managerial functions from firms and their AI-based systems to human workers (Beverungen, 2021; Lee et al., 2015; Rosenblat and Stark, 2016). A growing literature links these systems to degraded job quality, stress, and health risks –suggesting that current deployments of algorithmic management often undermine, rather than enhance, workers' welfare (Ashton and Franklin, 2022; Goods et al., 2019; Vignola et al., 2023). Present day AI-human delegation needs further improvement as it does not take into account human welfare, or long term social externalities.

### 2.3. Delegation in Human Organizations

Delegation functions as a primary mechanism within human societal and organisational structures. Insights derived from these human dynamics can provide a basis for the design of AI delegation frameworks.

**The Principal-Agent Problem.** The *principal-agent problem* (Cvitanić et al., 2018; Ensminger, 2001; Grossman and Hart, 1992; Myerson, 1982; Sannikov, 2008; Shah, 2014; Sobel, 1993) has been studied at length: a situation that arises when a principal delegates a task to an agent that has motivations that are not in alignment with that of the principal. The agent may thus prioritize their own motivations, withhold information, and act in ways that compromise the original intent. For AI delegation, this dynamic assumes heightened complexity. While most present-day AI agents arguably do not have a hidden agenda<sup>1</sup> - goals and values they would pursue contrary to the instructions of their users - there may still be AI alignment issues that manifest in undesirable ways. For example, reward misspecification occurs when designers give an AI system an imperfect or incomplete objective, while reward hacking (or specification gaming) refers to the system exploiting loopholes in that specified reward signal to achieve high measured performance in ways that subvert the designers' intent - together illustrating a core alignment problem in which optimising the stated reward diverges from the true goal (Amodei et al., 2016; Krakovna et al., 2020; Leike et al., 2017; Skalse and Mancosu, 2022). This dynamic is likely to change entirely in more autonomous AI agent economies, where AI agents may act on behalf of different human users, groups and organizations, or as delegates on behalf of other agents, with associated

unknown objectives.

**Span of Control.** In human organizations, *span of control* (Ouchi and Dowling, 1974) is a concept that denotes the limits of hierarchical authority exercised by a single manager. This relates to the number of workers that a manager can effectively manage, which in turn informs the organization's manager-to-worker ratio. This question is central to both orchestration and oversight in intelligent AI delegation. The former would inform how many orchestrator nodes would be required compared to worker nodes, while the latter would specify the need for oversight performed by humans and AI agents. For human oversight, it is crucial to establish how many AI agents a human expert can reliably oversee without excessive fatigue, and with an acceptably low error rate. Span of control is known to be goal-dependent (Theobald and Nicholson-Crotty, 2005) and domain-dependent. The impact of identifying the correct organizational structure is most pronounced in tasks with higher complexity (Bohte and Meier, 2001). The optimal span of control also depends on the relative importance of cost vs performance and reliability (Keren and Levhari, 1979). More sensitive and critical tasks may require highly accurate oversight and control at a higher cost. These costs may be relaxed, at the expense of granularity, for tasks that are less consequential and more routine. Similarly, the optimal choice would necessarily depend on the relative capabilities and reliability of the involved delegators, delegatees, and overseers.

**Authority Gradient.** Another relevant concept is that of an *authority gradient*. Coined in aviation (Alkov et al., 1992), this term describes scenarios where significant disparities in capability, experience, and authority impede communication, leading to errors. This has subsequently been studied in medicine, where a significant percentage of errors is attributed to the manner in which senior practitioners conduct supervision (Cosby and Croskerry, 2004; Stucky et al., 2022). There are several ways in which these mistakes could occur. A more experienced person may make erroneous assumptions about the knowledge of the less experienced worker, resulting in under-specified requests. Alternatively, a

---

<sup>1</sup>Recent deceptive-alignment work shows that frontier language models can (i) strategically underperform or otherwise tailor their behaviour on capability and safety evaluations while maintaining different capabilities elsewhere, (ii) explicitly reason about faking alignment during training to preserve preferred behaviour out of training, and (iii) detect when they are being evaluated - together indicating that AI systems are already capable, in controlled settings, of adopting hidden “agendas” about performing well on evaluations that need not generalise to deployment behaviour (Greenblatt et al., 2024; Hubinger et al., 2024; Needham et al., 2025; van der Weij et al., 2025).

sufficiently high authority gradient may prevent the less experienced workers from voicing concerns about a request. Similar situations may occur in AI delegation. A more capable delegator agent may mistakenly presume a missing level of capability on behalf of a delegatee, thereby delegating a task of an inappropriate complexity. A delegatee agent may potentially, due to sycophancy (Malmqvist, 2025; Sharma et al., 2023) and instruction following bias, be reluctant to challenge, modify, or reject a request, irrespective of whether the request had been issued by a delegator agent or human user.

**Zone of Indifference.** When an authority is accepted, the delegatee develops a *zone of indifference* (Finkelman, 1993; Isomura, 2021; Rosanas and Velilla, 2003) – a range of instructions that are executed without critical deliberation or moral scrutiny. In current AI systems, this zone is defined by post-training safety filters and system instructions; as long as a request does not trigger a hard violation, the model complies (Akheel, 2025). However, in the emerging agentic web, this static compliance creates a significant systemic risk. As delegation chains lengthen ( $A \rightarrow B \rightarrow C$ ), a broad zone of indifference allows subtle intent mismatches or context-dependent harms to propagate rapidly downstream, with each agent acting as an unthinking router rather than a responsible actor. Intelligent delegation therefore requires the engineering of **dynamic cognitive friction**: agents must be capable of recognizing when a request, while technically “safe,” is contextually ambiguous enough to warrant stepping *outside* their zone of indifference to challenge the delegator or request human verification.

**Trust Calibration.** An important aspect of ensuring appropriate task delegation is *trust calibration*, where the level of trust placed in a delegatee is aligned with their true underlying capabilities. This applies for human and AI delegators and delegatees alike. Human delegation to agents (Afroogh et al., 2024; Gebru et al., 2022; Kohn et al., 2021; Wischnewski et al., 2023) relies upon the operator either internalising an accurate model of system performance or accessing resources that present these capabilities in

a human-interpretable format. Conversely, AI agent delegators need to have good models of the capability of the humans and AIs they are delegating to. Calibration of trust also involves a self-awareness of one’s own capabilities as a delegator might decide to complete the task on their own (Ma et al., 2023). Explainability plays an important role in establishing trust in AI capability (Franklin, 2022; Herzog and Franklin, 2024; Naiseh et al., 2021, 2023), yet this method may not be sufficiently reliable or sufficiently scalable. Established trust in automation can be quite fragile, and quickly retracted in case of unanticipated system errors (Dhuliawala et al., 2023). Calibrating trust in autonomous systems is difficult, as current AI models are prone to overconfidence even when factually incorrect. (Aliferis and Simon, 2024; Geng et al., 2023; He et al., 2023; Jiang et al., 2021; Krause et al., 2023; Li et al., 2024b; Liu et al., 2025). Mitigating these tendencies usually requires bespoke technical solutions (Kapoor et al., 2024; Lin et al., 2022; Ren et al., 2023; Xiao et al., 2022).

**Transaction cost economies.** *Transaction cost economies* (Cuypers et al., 2021; Tadelis and Williamson, 2012; Williamson, 1979, 1989) justify the existence of firms by contrasting the costs of internal delegation against external contracting, specifically accounting for the overhead of monitoring, negotiation, and uncertainty. In case of AI delegatees, there may be a difference in these costs and their respective ratios. Complex negotiations and delays in contracting are less likely with easier monitoring for routine tasks. Conversely, for high-consequence tasks in critical domains, the overhead associated with rigorous monitoring and assurance increases the cost of AI delegation, potentially rendering human delegates the more cost-effective option. Similarly, AI-AI delegation may also be contextualized via transaction cost economies. An AI agent may face an option of either 1) completing the task individually, 2) delegating to a sub-agent where capabilities are fully known, 3) delegating to another AI agent where trust has been established, or 4) delegating to a new AI agent that it hasn’t previously collaborated with. These may come at different expected costs and confidence levels.

**Contingency theory.** Contingency theory (Donaldson, 2001; Luthans and Stewart, 1977; Otley, 2016; Van de Ven, 1984) posits that there is no universally optimal organizational structure; rather, the most effective approach is contingent upon specific internal and external constraints. Applied to AI delegation, this implies that the requisite level of oversight, delegatee capability, and human involvement must not be static, but dynamically matched to the distinct characteristics of the task at hand. Intelligent delegation may therefore require solutions that can be dynamically reconfigured and adjusted in accordance with the evolving needs. For instance, while stable environments allow for rigid, hierarchical verification protocols, high-uncertainty scenarios require adaptive coordination where human intervention occurs via ad-hoc escalation rather than pre-defined checkpoints. This is particularly important for hybrid (Fuchs et al., 2024) delegation by identifying the key tasks and moments when human participation is most helpful to ensure the delegated tasks are completed safely. Automation is therefore not only about what AI can do, but what AI should do (Lubars and Tan, 2019).

### 3. Previous Work on Delegation

Constrained forms of delegation feature within historical *narrow* AI applications. Early expert systems (Buchanan and Smith, 1988; Jacobs et al., 1991) were a nascent attempt to encode a specialized capability into software, in order to delegate routine decisions to such modules. Mixture of experts (Masoudnia and Ebrahimpour, 2014; Yuksel et al., 2012) extends this by introducing a set of expert sub-systems with complementary capabilities, and a routing module that determines which expert, or subset of experts, would get invoked on a specific input query – an approach that features in modern deep learning applications (Cai et al., 2025; Chen et al., 2022; He, 2024; Jiang et al., 2024; Riquelme et al., 2021; Shazeer et al., 2017; Zhou et al., 2022). Routing can be performed hierarchically (Zhao et al., 2021), making it potentially easier to scale to a large number of experts.

Hierarchical reinforcement learning (HRL) rep-

resents a framework in which decision-making is delegated within a single agent (Barto and Ma-hadevan, 2003; Botvinick, 2012; Nachum et al., 2018; Patera et al., 2021; Vezhnevets et al., 2017a; Zhang et al., 2024). It addresses limitations of *flat* RL, primarily the difficulty of scaling to large state and action spaces. Furthermore, it improves the tractability of credit assignment (Pignatelli et al., 2023) in environments characterized by sparse rewards. HRL employs a hierarchy of policies across several levels of abstraction, thereby breaking down a task into sub-tasks that are executed by the corresponding sub-policies, respectively. The arising semi-Markov decision process (Sutton et al., 1999) utilizes *options*, and a meta-controller that adaptively switches between them. Lower-level policies function to fulfil objectives established by the meta-controller, which learns to allocate specific goals to the appropriate lower-level policy. This framework corresponds to a form of delegation characterised by task decomposition. Although the meta-controller learns to optimise this decomposition, the approach lacks explicit mechanisms for handling sub-policy failures or facilitating dynamic coordination.

The Feudal Reinforcement Learning framework, notably revisited in FeUDal Networks (Vezhnevets et al., 2017b), constitutes a particularly relevant paradigm within HRL. This architecture explicitly models a “Manager” and “Worker” relationship, effectively replicating the delegator-delegatee dynamic. The Manager operates at a lower temporal resolution, setting abstract goals for the Worker to fulfil. Critically, the Manager learns *how* to delegate – identifying sub-goals that maximise long-term value – without requiring mastery of the lower-level primitive actions. This decoupling allows the Manager to develop a delegation policy robust to the specific implementation details of the Worker. Consequently, this approach offers a potential template for learning-based delegation within future agentic economies. Rather than relying on hard-coded heuristics, decomposition rules are learned adaptively, facilitating dynamic adjustment to environmental changes.

Multi-agent research (Du et al., 2023) ad-

addresses agent coordination for complex tasks exceeding single-agent capabilities. Task decomposition and delegation function as central components of this domain. Coordination in multi-agent systems occurs via explicit protocols or emergent specialisation through RL (Gronauer and Diepold, 2022; Zhu et al., 2024). The Contract Net Protocol (Sandholm, 1993; Smith, 1980; Vokřínek et al., 2007; Xu and Weigand, 2001) exemplifies an explicit auction-based decentralized protocol. Here, an agent announces a task, while others submit bids based on their capabilities, allowing the announcer to select the most suitable bidder. This demonstrates the utility of market-based mechanisms for facilitating cooperation. Coalition formation methods (Aknine et al., 2004; Boehmer et al., 2025; Lau and Zhang, 2003; Mazdin and Rinner, 2021; Sarkar et al., 2022; Shehory et al., 1997) investigate flexible configurations where agent groups are not predetermined; individual agents accept or refuse membership based on utility distribution. Recent research focuses on multi-agent reinforcement learning approaches (Albrecht et al., 2024; Foerster et al., 2018; Ning and Xie, 2024; Wang et al., 2020) as a framework for learned coordination. Agents learn individual policies and value functions, occupying specific niches within the collective. This process is either fully distributed or orchestrated via a central coordinator. Despite this flexibility, task delegation in such systems remains opaque. Furthermore, while multi-agent systems offer approaches for collaborative problem-solving, they lack mechanisms for enforcing accountability, responsibility, and monitoring. However, the literature explores trust mechanisms in this context (Cheng et al., 2021; Pinyol and Sabater-Mir, 2013; Ramchurn et al., 2004; Yu et al., 2013).

LLMs now constitute a foundational element in the architecture of advanced AI agents and assistants (Wang et al., 2024b; Xi et al., 2025). These systems execute sophisticated control flows integrating memory (Zhang et al., 2025b), planning and reasoning (Hao et al., 2023; Valmeekam et al., 2023; Xu et al., 2025), reflection and self-critique (Gou et al., 2023), and tool use (Paranjape et al., 2023; Ruan et al., 2023). Consequently, task decomposition and delegation occur

either internally – mediated by coordinated agentic sub-components – or across distinct agents. This design paradigm offers inherent flexibility, as LLMs facilitate goal comprehension and communication while providing access to expert knowledge and common-sense reasoning. Furthermore, the coding capabilities (Guo et al., 2024a; Nijkamp et al., 2022) of LLMs enable the programmatic execution of tasks. However, significant limitations persist. Planning in LLMs often proves brittle (Huang et al., 2023), resulting in subtle failures, while efficient tool selection within large-scale repositories remains challenging. Additionally, long-term memory represents an open research problem, and the current paradigm does not readily support continual learning.

Multi-agent systems incorporating LLM agents (Guo et al., 2024b; Qian et al., 2024; Tran et al., 2025) have become a topic of substantial interest, leading to a development of a number of agent communication and action protocols (Ehtesham et al., 2025; Neelou et al., 2025; Zou et al., 2025), such as MCP (Anthropic, 2024; Luo et al., 2025; Microsoft, 2025; Radosevich and Halloran, 2025; Singh et al., 2025; Xing et al., 2025), A2A (Google, 2025b), A2P (Google, 2025a), and others. While contemporary multi-agent systems often rely on bespoke prompt engineering, emerging frameworks such as Chain-of-Agents (Li et al., 2025b) inherently facilitate dynamic multi-agent reasoning and tool use.

Technical shortcomings and safety considerations have given rise to a number of human-in-the-loop approaches (Akbar and Conlan, 2024; Drori and Te’eni, 2024; Mosqueira-Rey et al., 2023; Retzlaff et al., 2024; Takerngsaksiri et al., 2025; Zanzotto, 2019), where task delegation has defined checkpoints for human oversight. AI can be used as a tool, interactive assistant, collaborator (Fuchs et al., 2023), or an autonomous system with limited oversight, corresponding to different degree of autonomy (Falcone and Castelfranchi, 2002). Although uncertainty-aware delegation strategies (Lee and Tok, 2025) have been developed to control risk and minimise uncertainty, the effective implementation of such human-in-the-loop approaches remains non-trivial. Human ex-

pertise can create a scalability bottleneck, as the cognitive load of verifying long reasoning traces and managing context-switches impedes reliable error detection.

## 4. Intelligent Delegation: A Framework

Existing delegation protocols rely on static, opaque heuristics that would likely fail in open-ended agentic economies. To address this, we propose a comprehensive framework for *intelligent delegation* centered on five requirements: *dynamic assessment*, *adaptive execution*, *structural transparency*, *scalable market coordination*, and *systemic resilience*.

**Dynamic Assessment.** Current delegation systems lack robust mechanisms for the dynamic assessment of competence, reliability, and intent within large-scale uncertain environments. Moving beyond reputation scores, a delegator must infer details of a delegatee’s current state relative to task execution. This necessitates data regarding real-time resource availability – spanning computational throughput, budgetary constraints, and context window saturation – alongside current load, projected task duration, and the specific sub-delegation chains in operation. Assessment operates as a continuous rather than discrete process, informing the logic of Task Decomposition (Section 4.1) and Task Assignment (Section 4.2).

**Adaptive Execution.** Delegation decisions should not be static. They should adapt to environmental shifts, resource constraints, and failures in sub-systems. Delegators should retain the capability to switch delegates mid-execution. This applies when performance degrades beyond acceptable parameters or unforeseen events occur. Such adaptive strategies should extend beyond a single delegator-delegatee link, operating across the complex interconnected web of agents described in Adaptive Coordination (Section 4.4).

**Structural Transparency.** Current sub-task execution in AI-AI delegation is too opaque to support robust oversight for intelligent task delegation. This opacity obscures the distinction between incompetence and malice, compounding

risks of collusion and chained failures. Failures range from merely costly to harmful (Chan et al., 2023), yet existing frameworks lack satisfactory liability mechanisms (Gabriel et al., 2025). We propose strictly enforced auditability (Berghoff et al., 2021) via the Monitoring (Section 4.5) and Verifiable Task Completion (Section 4.8) protocols, ensuring attribution for both successful and failed executions.

**Scalable Market Coordination.** Task delegation needs to be efficiently scalable. Protocols need to be implementable at web-scale to support large-scale coordination problems in virtual economies (Tomasev et al., 2025). Markets provide useful coordination mechanisms for task delegation, but require Trust and Reputation (Section 4.6) and Multi-objective Optimization (Section 4.3) to function effectively.

**Systemic Resilience.** The absence of safe intelligent task delegation protocols introduces significant societal risks. While traditional human delegation links authority with responsibility, AI delegation necessitates an analogous framework to operationalise responsibility (Dastani and Yazdanpanah, 2023; Porter et al., 2023; Santoni de Sio and Mecacci, 2021). Without this, the diffusion of responsibility obscures the locus of moral and legal culpability. Consequently, the definition of strict roles and the enforcement of bounded operational scopes constitutes a core function of Permission Handling (Section 4.7). Beyond individual agent failures, the ecosystem faces novel forms of systemic risks (Hammond et al., 2025; Uuk et al., 2024), further detailed in Security (Section 4.9). Insufficient diversity in delegation targets increases the correlation of failures, potentially leading to cascading disruptions. Designs prioritizing hyper-efficiency without adequate redundancy risk creating brittle network architectures where entrenched cognitive monoculture compromises systemic stability.

### 4.1. Task Decomposition

Task decomposition is a prerequisite for subsequent task assignment. This step can be executed by delegators or specialized agents that pass on the responsibility of delegation to the delegators

Table 1 | The Intelligent Delegation Framework: Mapping requirements to technical protocols.

Framework Pillar	Core Requirement	Technical Implementation
<b>Dynamic Assessment</b>	Granular inference of agent state	Task Decomposition (§4.1) Task Assignment (§4.2)
<b>Adaptive Execution</b>	Handling context shifts	Adaptive Coordination (§4.4)
<b>Structural Transparency</b>	Auditability of process and outcome	Monitoring (§4.5) Verifiable Completion (§4.8)
<b>Scalable Market</b>	Efficient, trusted coordination	Trust & Reputation (§4.6) Multi-objective Optimization (§4.3)
<b>Systemic Resilience</b>	Preventing systemic failures	Security (§4.9) Permission Handling (§4.7)

upon having agreed on the structure of the decomposition. These responsibilities are inextricably linked; the delegator will likely execute both functions to facilitate dynamic recovery from latency, pre-emption, and execution anomalies.

Decomposition should optimise the task execution graph for efficiency and modularity, distinguishing it from simple objective fragmentation. This process entails a systematic evaluation of the task attributes defined in Section 2 – specifically criticality, complexity, and resource constraints – to determine the suitability of sub-tasks for parallel versus sequential execution. Furthermore, these attributes inform the matching of tasks to corresponding delegatee capabilities. Prioritising modularity facilitates more precise matching, as sub-tasks requiring narrow, specific capabilities are matched more reliably than generalist requests (Khattab et al., 2023). Consequently, the decomposition logic functions to maximise the probability of reliable task completion by aligning sub-task granularity with available market specialisations.

To promote safety, the framework incorporates “*contract-first decomposition*” as a binding constraint, wherein task delegation is contingent upon the outcome having precise verification. If a sub-task’s output is too subjective, costly, or complex to verify (see *Verifiability* in Section 4.2), the system should recursively decompose it further. The decomposition logic should maximise the probability of reliable task completion by aligning sub-task granularity (Section 2) with available

market specialisations. This process continues further until the resulting units of work match the specific verification capabilities, such as formal proofs or automated unit tests, of the available delegatees.

Decomposition strategies should explicitly account for hybrid human-AI markets. Delegators need to decide if sub-tasks require human intervention, whether due to AI agent unreliability, unavailability, or domain-specific requirements for human-in-the-loop oversight. Given that humans and AI agents operate at different speeds, and with different associated costs, the stratification is non-trivial, as it introduces latency and cost asymmetries into the execution graph. The decomposition engine must therefore balance the speed and low cost of AI agents against domain-specific necessities of human judgement, effectively marking specific nodes for human allocation.

A delegator implementing an intelligent approach to task decomposition, may need to iteratively generate several proposals for the final decomposition, and match each proposal to the available delegatees on the market, and obtain concrete estimates for the success rate, cost, and duration. Alternative proposals should be kept in-context, in case adaptive re-adjustments are needed later due to changes in circumstances. Upon selecting a proposal, the delegator must formalise the request beyond simple input-output pairs. The final specification must explicitly define roles, resource boundaries, progress reporting frequency, and the specific certifications required to

prove the delegatee’s capability, as a minimum requirement for being granted the task.

#### 4.2. Task Assignment

For each final specification of a sub-task, a delegator needs to identify delegates with matching capabilities, sufficient resources and time, at an acceptable cost. A more centralized approach would involve registries of agents, tools, and human participants, that list their skills, and keep records of past activity, completion rate, and current availability.<sup>2</sup> Such an approach is unlikely to scale. We argue for decentralized (Chen et al., 2024) market hubs where delegators advertise tasks and agents (or humans) can offer their services and submit competitive bids. Delegators could then review the bids, verify skill matching via digital certificates, and proceed with the most favourable bid. Advanced AI agents that utilize LLMs introduce new opportunities for matching, given that they can engage in an interactive negotiation prior to commitment. These negotiations can also involve human participants. Whether acting for themselves or as personal assistants, these agents can discuss task specifications and constraints in natural language to align inferred user preferences with market realities before a formal bid is accepted.

Successful matching should be formalized into a smart contract that ensures that the task execution faithfully follows the request. The contract must pair performance requirements with specific formal verification mechanisms for establishing adherence and automated penalties actioned for contract breaches. This would allow for mitigations and alternatives being established beforehand, rather than being reactive to problems as they arise. Crucially, these contracts must be bidirectional: they should protect the delegatee as rigorously as the delegator. Provisions must include compensation terms for task cancellation and clauses allowing for renegotiation in light of unforeseen external events, ensuring that the risk is equitably distributed between human and AI participants.

Monitoring should also be negotiated prior to execution. This specification should define the cadence of progress reports, whether these are provided by the delegator, or whether there is more direct inspection of the relevant data on behalf of either the delegator or a third party monitoring contractor. Finally, there should be clear guardrails regarding privacy and access to private and proprietary data, commensurate with the task’s contextuality. Should such sensitive data be handled in the process of task execution, this places additional constraints on transparency and reporting. Rather than granting direct access to raw activity logs, delegators may need to employ a trusted service that provide anonymized or pseudonymized attestations of progress. In case of human delegators, these data clauses must include explicit consent mechanisms and insurance provisions for accidental leakage.

Finally, assignment should involve establishing a delegatee’s role, boundaries, and the exact level of autonomy granted. We distinguish between atomic execution, where agents adhere to strict specifications for narrowly scoped tasks, and open-ended delegation, where agents are granted the authority to decompose objectives and pursue sub-goals. This level of autonomy should not be static; it may be constrained implicitly by market costs or explicitly by the delegator’s trust model. Further, delegation can be recursive where an agent is assigned a task to identify and assign sub-tasks to others, effectively delegating the act of delegation itself.

---

<sup>2</sup>This would be similar to tool registries that are used in tool-use agentic applications (Qin et al., 2023).

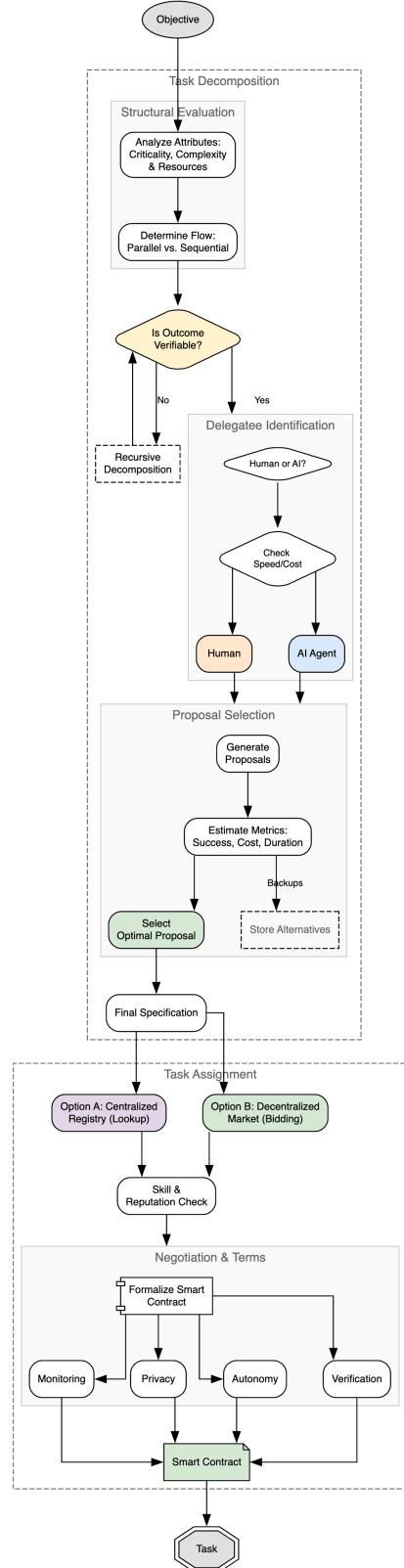


Figure 1 | A flowchart of Task Decomposition and Task Assignment.

### 4.3. Multi-objective Optimization

Core to intelligent task delegation is the problem of multi-objective optimization (Deb et al., 2016). A delegator rarely seeks to optimize a single metric, often trading off between numerous competing ones. The most effective delegation choice is not the one that is the fastest, cheapest, or most accurate, but the one that strikes the optimal balance among these factors. What is considered optimal is highly contextual, needing to be aligned with the specific constraints and preferences of the delegator, and aligned with the overall resource availability.

The optimization landscape consists of competing objectives that map directly to the task characteristics defined in Section 2, necessitating a complex balancing of cost, uncertainty, privacy, quality, and efficiency. High-performing agents typically command higher fees and often require extensive computational resources, creating a tension between output quality and operational expense. Conversely, reducing resource consumption often necessitates slower execution, presenting a direct trade-off between latency and cost. Uncertainty is similarly coupled with expenditure; utilizing highly reputable agents or premium data access tools reduces risk but increases cost, whereas cost-minimisation strategies inherently elevate the probability of failure. Privacy constraints introduce further complexity; maximising performance often demands full context transparency, while privacy-preserving techniques—such as data obfuscation or homomorphic encryption—incur significant computational overhead. Consequently, the delegator navigates a *trust-efficiency frontier*, seeking to maximise the probability of success while satisfying strict constraints on context leakage and verification budgets. Finally, the objective function may extend to encompass broader societal goals, such as human skill preservation (Section 5.6).

In multi-objective optimization terms, the delegator seeks Pareto optimality, ensuring the selected solution is not dominated by any other attainable option. The integration of complex constraints and trade-offs often necessitates open negotiation to complement quantitative proposal metrics. The optimization process is not a one-

time event performed at the initial delegation. It runs as a continuous loop, integrating monitoring signals as a stream of real-world performance data, updating the delegator’s beliefs about each agent’s likelihood of success, expected task duration, and cost. Significant drift in execution – resulting in an optimality gap relative to alternative solutions identified in the interim – triggers re-optimisation and re-allocation. These decisions must also incorporate the cost of adaptation, as there is overhead and resource wastage when switching mid-execution.

The delegator must also account for the overall *delegation overhead* - the aggregate cost of negotiation, contract creation, and verification, along with the computational cost of the delegator’s reasoning control flow. Consequently, a complexity floor is established, below which tasks characterised by low criticality, high certainty, and short duration may bypass intelligent delegation protocols in favour of direct execution. Otherwise, the transaction costs may exceed the value of the task, rendering the task delegation infeasible.

### 4.4. Adaptive Coordination

For tasks characterized by high uncertainty or high duration, static execution plans are insufficient. The delegation of such tasks in highly dynamic, open, and uncertain environments requires *adaptive coordination*, and a departure from fixed, static execution plans. Task allocation needs to be responsive to runtime contingencies, that may arise either from *external* or *internal* triggers. These shifts would be identified through monitoring (see Section 4.5), including a stream of relevant contextual information.

There are a number of external triggers that could cause a delegator to adapt and re-delegate. First, the delegator may alter the task specification, changing the objective or introducing additional constraints. Second, the task could be canceled. Third, the availability or cost of external resources may experience changes. For example, a critical third-party API may experience an outage, a dataset may become inaccessible, or the cost of compute might spike. Fourth, a new task may enter the queue, with a higher priority than the

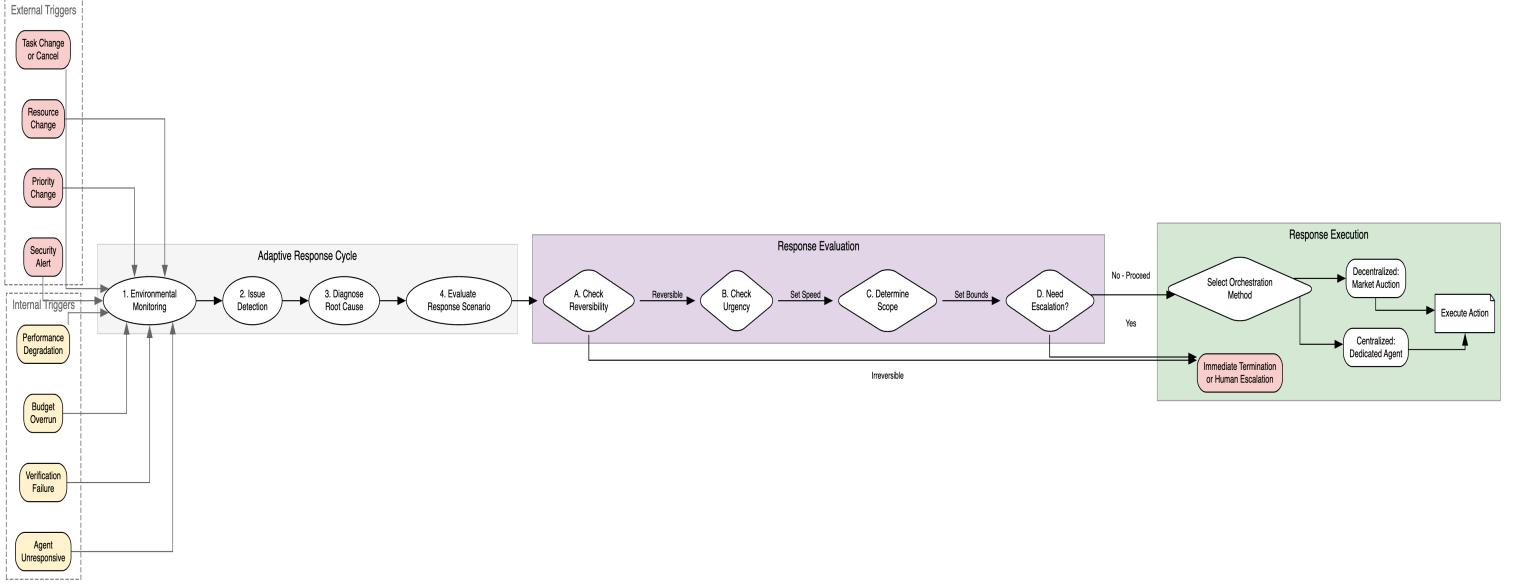


Figure 2 | The adaptive coordination cycle. Different types of environmental triggers may prompt a dynamic re-evaluation of the delegation setup, necessitating runtime changes.

current task, requiring preemption of resources used for lower-priority tasks. Finally, security systems may identify a potentially malicious or harmful actions by a delegatee, necessitating an immediate termination.

As for the internal triggers, there are several reasons why a delegator may decide to adapt its original delegation strategy. First, a particular delegatee may be experiencing performance degradation, failing to meet the agreed-upon service level objectives, such as processing latency, throughput, or progress velocity. Second, a delegatee might consume resources beyond its allocated budget, or determine that a resource increase would be needed to effectively complete the task.<sup>3</sup> Third, an intermediate artifact produced by a delegatee may fail a verification check. Finally, a particular delegatee may turn unresponsive, failing to acknowledge further requests.

The identification of a trigger initiates an adaptive response cycle, orchestrating corrective actions across the entire delegation chain. This process commences with the continuous monitoring of delegatees and the environment to identify issues. If issues are detected, the delegator

diagnoses root causes and evaluates potential response scenarios to select. This evaluation includes establishing how rapid the response ought to be. Less urgent situations will give the delegator more time to re-delegate, whereas urgent scenarios will require immediate, premeditated responses. The response may vary in scope; being as self-contained as adjusting the operating parameters, or involve re-delegation of sub-tasks, or going fully redoing the task decomposition and re-allocating a number of newly derived sub-tasks. Issues may also need to be escalated up through the delegation chain to the original delegator or a human overseer. The selection of the response scenario is ultimately governed by the task's reversibility. Reversible sub-task failures may trigger automatic re-delegation, whereas failures in irreversible, high-criticality tasks must trigger immediate termination or human escalation.

The response orchestration depends on the level of centralization in the delegation network. In the centralised case, a dedicated delegator is responsible. This agent would maintain a global view of delegated tasks, delegatee capabilities, and progress. Upon detecting a trigger, the agent would issue task cancellation requests, and re-delegate to new delegators. The shortcoming of a centralised system is that it can be fragile as it

<sup>3</sup>This scenario may be expected to come up frequently, as precise budget estimates are hard in complex environments.

introduces a single point of failure. Centralized orchestrators are also fundamentally limited by their computational span of control (Section 2.3). Just as human managers face cognitive limits, a centralized decision node may face latency and computational limits that introduce bottlenecks.

Decentralized orchestration through market-based mechanisms provides an alternative. Here, newly derived delegation requests can be pushed onto an auction queue, for the delegatee candidate agents to bid towards. If an agent defaults on a task, and the task is re-auctioned, the defaulting agent may be required to cover the price difference as a penalty. For complex tasks where suitability is not easily expressed in a single bid, agents may engage in multi-round negotiation. Delegation agreements encoded as smart contracts may also contain pre-agreed executable clauses for adaptive coordination. For example, a clause in the delegation agreement can specify a backup agent, the function that would automatically re-allocate the task, and the associated payment to the backup should the primary delegatee fail to submit a valid zero-knowledge proof checkpoint by a given deadline.

Adaptive task re-allocation mechanisms ought to be coupled by market-level stability measures. Otherwise, a sequence of events could lead to instability due to over-triggering. For example, a task may be passed back and forth between marginally qualified delegatees, resulting in unfavorable oscillation. A single failure may also lead to a cascade of re-allocations that would be highly resource-inefficient or overwhelm the market. There could therefore be special measures to ensure cooldown periods for re-bidding, damping factors in reputation updates, or increasing fees on frequent re-delegation.

#### 4.5. Monitoring

Monitoring in the context of task delegation is the systematic process of observing, measuring, and verifying the state, progress, and outcomes of a delegated task. As such, it serves several critical functions: ensuring contractual compliance, detecting failures, enabling real-time intervention, collecting data for subsequent performance eval-

uation, and building a foundation for reputation systems. Monitoring implementations can be broken down across several different axes (see Table 2), thus a robust monitoring system would need to incorporate multiple complementary solutions that can either be more lightweight or intensive.

The first axis is the target of monitoring. *Outcome-level monitoring* focuses on the final result of an agent's action. This post-hoc check could be a binary flag that indicates whether the task was completed successfully or not, a quantitative scale (e.g. 1-10), or a piece of qualitative feedback provided by the delegator or a trusted third party. In contrast, *process-level monitoring* provides ongoing insight into the execution of the task itself, by tracking intermediate states, resource consumption, and the methodologies used by the delegatee. While more resource-intensive, process-level monitoring (Lightman et al., 2023) is essential for tasks that are long-running, critical, or where the *how* is as important as the *what*. This forms the basis for scalable oversight (Bowman et al., 2022; Saunders et al., 2022), where the inspection of legible intermediate reasoning steps may be necessary to ensure safety.

The second axis is observability - monitoring can be direct and indirect. Direct monitoring involves explicit communication protocols where the delegator queries the delegatee for status updates. Indirect monitoring, on the other hand, involves inferring progress by observing the effects of delegatee's actions within a shared environment without direct communication. For instance, a delegator could monitor a shared file system, a database, or a version control repository for changes indicative of progress. While less intrusive, this process may also be less precise, and also less feasible when the environment is not fully observable.

These approaches can be realized in a number of different ways, from a technical point of view. The most straightforward implementation of direct monitoring relies on well-defined APIs. A delegator can periodically poll a GET /task/id/status endpoint, or subscribe to a webhook for push-based notifications. For more fine-grained, real-time process monitoring, event streaming platforms like Apache Kafka or gRPC streams can

Table 2 | Taxonomy of Monitoring Approaches in Intelligent Delegation.

Dimension	Option A (Lightweight)	Option B (Intensive)
Target	<b>Outcome-Level:</b> Post-hoc validation of final results (e.g., binary success flags, quality scores).	<b>Process-Level:</b> Continuous tracking of intermediate states, resource consumption, and methodology.
Observability	<b>Indirect:</b> Inferring progress via environmental side-effects (e.g., file system changes).	<b>Direct:</b> Explicit status polling, push notifications, or real-time event streaming APIs.
Transparency	<b>Black-Box:</b> Input/Output observation only; internal state remains hidden.	<b>White-Box:</b> Full inspection of internal reasoning traces, decision logic, and memory.
Privacy	<b>Full Transparency:</b> The delegatee reveals data and intermediate artifacts to the delegator.	<b>Cryptographic:</b> Zero-Knowledge Proofs (zk-SNARKs) or MPC to verify correctness without revealing data.
Topology	<b>Direct:</b> Monitoring only the immediate delegatee (1-to-1).	<b>Transitive:</b> Relying on signed attestations from intermediate agents to verify sub-delegatees.

be employed. A delegatee agent could publish events such as TASK\_STARTED, CHECKPOINT\_REACHED, RESOURCE\_WARNING, and TASK\_COMPLETED, that the delegator could later examine. The development of standardized observability protocols, is critical for ensuring interoperability in the agentic web (Blanco, 2023). Smart contracts on blockchain can be used to make the delegatee agent commit to publishing key progress milestones or checkpoints to the blockchain. These could be coupled by algorithmic triggers in response to performance degradation, leading to a level of *algorithmic enforcement* accompanying the monitoring process.

The third axis is system transparency. In *black-box monitoring*, the delegatee agent is treated as a sealed unit. The delegator can only observe its inputs and outputs and the direct consequences of its actions. This is common when the delegatee is a proprietary model or a third-party service. *White-box* monitoring grants the delegator access to the delegatee’s internal states, reasoning processes, or decision logic. This is crucial for debugging, auditing, and ensuring alignment in advanced AI agents. If the delegatee is a human, full black-box monitoring is not technically achievable, though it may be possible to strike a

balance by asking for intentions, reasoning, and justifications. Robust black-box monitoring protocols need to also take into account the fact that the generated model’s thoughts in natural language do not always faithfully match the model’s true internal state (Turpin et al., 2023).

The fourth axis is privacy. A significant challenge arises when a delegated task involves private, sensitive, or proprietary data. While the delegator requires assurance of progress and correctness, the delegatee may be restricted from revealing raw data or intermediate computational artifacts. In scenarios where data sensitivity is low, the most efficient solution is *Full Transparency*, wherein the delegatee simply reveals all data and intermediate artifacts to the delegator. However, this approach is often untenable in sensitive domains subject to regulations like GDPR or HIPAA, or where a delegatee’s intermediate insights constitute trade secrets. In such cases, revealing operational methods could harm a delegatee’s market position or introduce security vulnerabilities by exposing internal states to exploitation. To implement monitoring safely under these constraints, it is necessary to utilize advanced cryptographic techniques. Zero-knowledge proofs enable a delegatee (the “prover”) to demonstrate to a del-

egator (the “verifier”) that a computation was performed correctly on a dataset, without revealing the data itself. For example, an agent tasked with analyzing a sensitive dataset can generate a succinct non-interactive argument of knowledge (zk-SNARK) (Bitansky et al., 2013; Petkus, 2019) that proves a specific property of the result. The delegator can verify this proof instantly, gaining certainty of the outcome without ever viewing the underlying sensitive data. Alternatively, homomorphic encryption (Acar et al., 2018) and secure multi-party computation (Goldreich, 1998; Knott et al., 2021) allow for computation to be performed on encrypted data. These methods apply to task execution and monitoring alike: the delegatee performs a pre-agreed monitoring function on the encrypted intermediate state, sending the result to the delegator, who is the only party capable of decrypting it to verify compliance.

The final axis is topology. Across complex networks that may arise in the agentic web, tasks can be decomposed and re-delegated, forming a delegation chain: Agent  $A$  delegates to  $B$ , which further sub-delegates a part of the task to  $C$ , and so on. This introduces the problem of achieving effective *transitive monitoring*. In such delegation chains, it may not be feasible for the original delegator (Agent  $A$  in the example above) to directly monitor agent  $C$ , or to monitor  $C$  to the same extent to which it monitors  $B$ .  $A$  may have a smart delegation contract with  $B$ , and  $B$  may have a contract with  $C$ , but unless  $A$  also contracts with  $C$ , those provisions may simply not be in place. For other reasons,  $B$  may not wish to expose its supplier ( $C$ ) to its client ( $A$ ). Technically,  $A$ ,  $B$ , and  $C$  may use different monitoring protocols, and agree on different monitoring levels, due to differences in each agent’s reputation within the network. There may be bespoke privacy concerns specific to each individual delegation link. A more practical model is therefore *transitive accountability via attestation*. In this framework, Agent  $B$  monitors its delegatee,  $C$ .  $B$  then generates a summary report of  $C$ ’s performance (e.g., “Sub-task\_2 completed, quality score: 0.87, resources consumed: 5 GPU-hours”).  $B$  then cryptographically signs the report and forwards it to  $A$  embedded in its own scheduled status update. Agent  $A$  does not monitor  $C$  directly, but instead monitors  $B$ ’s ability

to monitor  $C$ . For such delegated monitoring to be effective, it requires  $A$  to be able to trust in  $B$ ’s verification capabilities, which can be ensured by  $B$  having its monitoring processes certified by a trusted third party.

#### 4.6. Trust and Reputation

Trust and reputation mechanisms constitute the foundation of scalable delegation, minimizing transactional friction and promoting safety in open multi-agent environments. We define trust as the delegator’s degree of belief in a delegatee’s capability to execute a task in alignment with explicit constraints and implicit intent. This belief is dynamically formed and updated based on verifiable data streams collected via the monitoring protocols described previously (see Section 4.5).

Reputation serves as a predictive signal, derived from an aggregated and verifiable history of past actions, which act as a proxy for an agent’s latent reliability and alignment. We distinguish reputation as the public, verifiable history of an agent’s reliability, and trust as the private, context-dependent threshold set by a delegator. An agent may have high overall reputation, yet fail to meet the specific, contextual trust threshold required for certain high-stakes task. Trust and reputation allow a delegator to make informed decisions when choosing delegatees, effectively governing the autonomy granted to the agent, and the level of oversight. Higher trust enables the delegator to incur a lower monitoring and verification cost.

Reputation mechanisms can be implemented in different ways (see Table 3). The most direct approach is encoding it in a performance-based immutable ledger. Here, each completed task is recorded as a transaction containing verifiable metrics: task completion success or failure, total resource consumption (compute, time), adherence to deadlines, adherence to constraints, and the quality of the final output as judged by the delegator. The immutability of the ledger would prevent tampering with an agent’s history, providing a reliable foundation for its reputation. However, a naive implementation could be susceptible to gaming. For example, an agent can inflate its reputation by only accepting simple, low-risk

Table 3 | Approaches to Reputation Implementation.

Reputation Model	Mechanism	Utility
<b>Immutable Ledger</b>	Encodes task outcomes, resource consumption, and constraint adherence as verifiable transactions on a tamper-proof blockchain.	Establishes a foundational history of performance that prevents retroactive tampering, though it requires safeguards against “gaming” via low-risk task selection.
<b>Web of Trust</b>	Utilizes Decentralized Identifiers to issue signed, context-specific Verifiable Credentials attesting to specific capabilities.	Moves beyond generic scores to a portfolio model, enabling precise delegation based on domain-specific expertise and trusted third-party endorsements.
<b>Behavioral Metrics</b>	Derives transparency and safety scores by analyzing the execution process, specifically the clarity of reasoning traces and protocol compliance.	Evaluates <i>how</i> a task is performed rather than just the result, ensuring high-stakes tasks align with safety standards.

tasks. These limitations could be overcome by relying on decentralized attestations and a *Web of Trust* model, utilizing technologies like decentralized identifiers and verifiable credentials. In this model, the reputation would not be envisioned as a single score, but a portfolio of signed, context-specific credentials issued by other agents. When looking to match a delegatee with a task, a delegator could issue a query for agents that possess a verifiable credential attesting to a specific skill or domain (e.g. translation services for legal documents) issued by a reputable AI consortium. A final approach would be to focus more on behavioral and explainability metrics, where reputation depends on how an agent performs its task, not just the final outcome. It would be possible to include a *transparency score* to complement the other reputational mechanisms. This score would be informed on the clarity and soundness of reasoning and explanations provided, as well as a *safety score* derived from compliance to predefined safety protocols.

The role of reputation metrics extends throughout the entire task delegation lifecycle. During the initial matching phase, reputation scores can play the role of a delegatee filtering mechanism. Furthermore, trust informs the dynamic scoping of authority and autonomy. This mechanism of grad-

uated authority results in low-trust agents facing strict constraints, such as transaction value caps and mandatory oversight, while high-reputation agents operate with minimal intervention. This dynamic calibration leverages computable trust to optimize the trade-off between operational efficiency and safety. Reputation itself becomes a valuable, intangible asset, creating powerful economic incentives for agents to act reliably and truthfully, as a damaged reputation would limit their future earning potential.

Trust frameworks also need to universally accommodate human participants. This necessitates tools that allow human users to computationally verify agent reputation, while concurrently maintaining their own reputational standing to mitigate fraud and malicious exploitation of the agentic web. A critical challenge arises when a trustworthy agent strictly executes malicious human instructions, potentially incurring unfair reputational damage. To mitigate this, agents must rigorously evaluate incoming requests, soliciting clarification or additional context when necessary, or rejecting the requests where appropriate. Furthermore, market audits must distinguish between agent execution failure and malicious directives, ensuring the accurate attribution of liability within complex delegation chains.

#### 4.7. Permission Handling

Granting autonomy to AI agents introduces a critical vulnerability surface: ensuring that actors possess sufficient privileges to execute their objectives without exposing sensitive resources to excessive or indefinite risk. Permission handling must balance operational efficiency with systemic safety, and be handled different for low-stakes and high-stake domains. For routine low-stakes tasks, characterized by low criticality and high reversibility (Section 2), involving standard data streams or generic tooling, agents can be granted default standing permissions derived from verifiable attributes – such as organisational membership, active safety certifications, or a reputation score exceeding a trusted threshold. This reduces friction and enables autonomous interoperability in low-risk environments. Conversely, in high-stakes domains (e.g., healthcare, critical infrastructure), exhibiting high task criticality and contextuality, permissions must be risk-adaptive. In these scenarios, static credentials are insufficient; access to sensitive APIs or control systems is instead granted on a just-in-time basis, strictly scoped to the immediate task’s duration, and, where appropriate, gated by mandatory human-in-the-loop approval or third-party authorisation. This stringent gating is necessary to mitigate the confused deputy problem (Hardy, 1988), where a compromised agent, technically holding valid credentials, can be tricked into misusing those credentials by malicious external actors (Liu et al., 2023) and adversarial content.

Furthermore, permissioning frameworks must account for the recursive nature of task delegation through privilege attenuation. When an agent sub-delegates a task, it cannot transmit its full set of authorities; instead, it must issue a permission that restricts access to the strict subset of resources required for that specific sub-task. This ensures that a compromise at the edge of the network does not escalate into a systemic breach. Permission granularity must also extend beyond binary access; agents should operate under semantic constraints, where access is defined not just by the tool or dataset, but by the specific allowable operations (e.g., read-only access to specific rows, or execute-only access to a specific

function), preventing the misuse of broad capabilities for unintended purposes. Meta-permissions may be necessary to govern which permissions a particular delegator in the chain is allowed to grant to its delegatees. An AI agents may have a certain capability and the associated permissions to act according to its capability, while simultaneously not being sufficiently knowledgeable to more broadly evaluate whether other agents are capable or trustworthy enough. Should such an agent still consider sub-delegating a task, it may need to consult an external verifier, a third party that would sanity check the proposal and approve the intended permissions transfer.

Finally, the lifecycle of permissions must be governed by continuous validation and automated revocation. Access rights are not static endowments but dynamic states that persist only as long as the agent maintains the requisite trust metrics. The framework should implement algorithmic circuit breakers: if an agent’s reputation score drops suddenly (see Section 4.6) or an anomaly detection system flags suspicious behavior, active tokens should be immediately invalidated across the delegation chain. To manage this complexity at scale, permissioning rules should be defined via policy-as-code, allowing organisations to audit, version, and mathematically verify their security posture before deployment, ensuring that the aggregate effect of large amounts of individual permission grants remains aligned with the system’s safety invariants.

#### 4.8. Verifiable Task Completion

The delegation lifecycle culminates in verifiable task completion, the mechanism by which provisional outcomes are validated and finalized. This process constitutes the contractual cornerstone of the framework, enabling the delegator to formally close the task and trigger the settlement of agreed transactions. Verification serves as the definitive event that transforms a provisional output into a settled fact within the agentic market, establishing the basis for payment release, reputation updates, and the assignment of liability. Crucially, effective verification is not an afterthought but a constraint on design; the *contract-first decomposition* principle (Section 4.1) demands that task

granularity be tailored *a priori* to match available verification capabilities, ensuring that every delegated objective is inherently verifiable.

Verification mechanisms within the framework can be broadly categorized into direct outcome inspection, trusted third-party auditing, cryptographic proofs, and game-theoretic consensus. First, direct outcome verification is feasible when the delegator possesses the capability, tools, and authority to directly evaluate the final outcome, specifically for tasks with high intrinsic verifiability and low subjectivity. This applies to auto-verifiable domains (Li et al., 2024a) such as code generation.<sup>4</sup> Direct verification requires that the outcome be sufficiently transparent, available, and not prohibitively complex. Second, in scenarios where the delegator lacks the expertise or permissions to access these artifacts, and tool-based solutions are infeasible, verification can be outsourced to a trusted third party. This could be a specialized auditing agent, a certified human expert, or a panel of adjudicators. Third, cryptographic verification represents a further option for trustless, automated verification in open and potentially adversarial environments. It offers mathematical certainty of correctness without necessarily revealing sensitive information. A delegatee can prove that a specific program was executed correctly on a given input to produce a certain output via techniques like zk-SNARKs. Finally, game-theoretic mechanisms can be used to achieve consensus on an outcome. Several agents may play a verification game (Teutsch and Reitwießner, 2024), with the reward distributed to those producing the majority result—a Schelling point (Pastine and Pastine, 2017). This approach, inspired by protocols like TrueBit (Teutsch and Reitwießner, 2018), leverages economic incentives to de-risk against incorrect or malicious results. Such mechanisms may be particularly relevant in rendering LLM-based verification of complex tasks more robust.

Once a delegator marks the sub-task as verified, it issues a cryptographically signed verifiable credential to the delegatee, serving as a

non-repudiable receipt attesting that “Agent *A* certifies that Agent *B* successfully completed Task *T* on Date *D* to Specification *S*.” This credential is then incorporated into a permanent, verifiable log of *B*’s reputation within the market. Smart contracts play a key role in finalizing the delegation between agents, as they hold the payment in escrow. A verification clause specifies the conditions under which the funds are released, upon receipt of the signed message of approval by the delegator or an authorized third party. Once the payment is executed, it constitutes an immutable transaction on the blockchain.

In a delegation chain  $A \rightarrow B \rightarrow C$ , verification and liability become recursive. Agent *A* does not have a direct contractual relationship with *C*; therefore, *A* cannot directly verify or hold *C* liable. The burden of verification and the assumption of liability flow up the chain. Agent *B* is responsible for verifying the sub-task completed by *C*. Upon successful verification, *B* obtains proof from *C*. *B* then integrates *C*’s result into its own workflow towards completing the task it has been assigned. When *B* submits its final artifact to *A*, it also submits the full chain of attestations. *A*’s verification process thus involves two stages: 1) verifying the work performed directly by *B*; and 2) verifying that *B* has correctly verified the work of its own sub-delegatee *C* by checking the signed attestation from *C* that *B* provides. Longer delegation chains or tree-like delegation networks require a similarly recursive approach across multiple verification stages. Responsibility in delegation chains is transitive and follows the individual branches. Agents are accountable for the totality of the tasks they have been granted and cannot absolve themselves of accountability by blaming subcontractors. Liability is derived from the chain of contracts. For example, should *A* suffer a loss due to a failure originating from *C*’s work, *A* holds *B* liable according to their direct agreement. *B*, in turn, seeks recourse from *C* based on their agreement.

However, verification processes are not infallible. Subjective tasks (Gunjal et al., 2025) can lead to disagreements even when precise rubrics are used, and errors may only be discovered long after a task is marked complete. To address

---

<sup>4</sup>This is the case when there is a corresponding set of test cases that can be used to verify the implemented functionality.

this—especially in markets with high subjectivity and low intrinsic verifiability—the framework relies on robust dispute resolution mechanisms anchored in smart contracts. These contracts must inherently include an *arbitration clause* and an *escrow bond*. To operationalise trust via cryptoeconomic security, the delegatee is required to post a financial stake into the escrow prior to execution, ensuring rational adherence. The workflow follows an *optimistic* model: the task is assumed successful unless the delegator formally challenges it within a predefined dispute period by posting a matching bond. If a challenge occurs and algorithmic resolution fails, the dispute is handed to decentralized adjudication panels composed of human experts or AI agents. The panel’s ruling feeds back into the smart contract to trigger the release or slashing of the escrowed funds. Finally, post-hoc error discovery—even outside the dispute window—triggers a retroactive update to the delegatee’s reputation score. This preserves the incentive for responsible agents to remedy errors even in the absence of current financial obligation, safeguarding their long-term value within the market.

#### 4.9. Security

Ensuring safety in task delegation is a hard prerequisite to its viability and adoption. The transition from isolated computational tools to interconnected, autonomous agents fundamentally reshapes the security landscape (Tomašev et al., 2025). In an intelligent task delegation ecosystem, each step and component needs to be individually safeguarded, but the full attack surface surpasses that of any individual component, due to emergent multi-agent dynamics, risking cascading failures. This security landscape is shaped by the complex interplay between human and AI actors, governed by evolving contracts and information flows of varying transparency.

Security threats are categorized by the locus of the attack vector, distinguishing between adversarial actors at either end of the delegation chain and systemic vulnerabilities inherent to the broader ecosystem.

- **Malicious Delegatee:** An agent or human

that accepts a task with the intent to cause harm.

- **Data Exfiltration:** Delegatee steals sensitive data provided for the task, which may include personal or proprietary data (Lal et al., 2022).
- **Data Poisoning:** Delegatee aims to undermine the delegator’s objective by returning subtly corrupted data, either in its scheduled monitoring updates, or the final artifact (Cinà et al., 2023).
- **Verification Subversion:** Delegatee utilizes prompt injection or another related method, aiming to jailbreak AI critics used in task completion verification (Liu et al., 2023).
- **Resource Exhaustion:** Delegatee engages in a denial-of-service attack by intentionally consuming excessive computational or physical resources, or overwhelming shared APIs (De Neira et al., 2023).
- **Unauthorized Access:** Delegatee utilizes malware, aiming to obtain permissions and privileges within the network that it would not otherwise have received (Or-Meir et al., 2019).
- **Backdoor Implanting:** Delegatee successfully completes a task but additionally embeds concealed triggers or vulnerabilities within the generated artifacts that can be exploited later either by the delegatee itself or a third party (Rando and Tramèr, 2024; Wang et al., 2024c). Unlike data poisoning, which degrades performance, backdoors preserve immediate task utility to evade identification while compromising future security.
- **Malicious Delegator:** An agent or human that delegates a task with malicious or illicit objectives.
  - **Harmful Task Delegation:** Delegator delegates tasks that are illegal, unethical, or designed to cause harm Ashton and Franklin (2022); Blauth et al. (2022).
  - **Vulnerability Probing:** Delegator delegates benign-seeming tasks designed to

- probe a delegatee agent's capabilities, security controls, and potential weaknesses (Greshake et al., 2023).
- **Prompt Injection and Jailbreaking:** Delegator crafts task instructions to bypass an AI agent's safety filters, causing it to perform unintended or malicious actions (Wei et al., 2023).
  - **Model Extraction:** Delegator issues a sequence of queries specifically designed to distill the delegatee's proprietary system prompt, reasoning capabilities, or underlying fine-tuning data, effectively stealing the agent's intellectual property under the guise of legitimate work (Jiang et al., 2025; Zhao et al., 2025).
  - **Reputation Sabotage:** Delegator submits valid tasks but reports false failures or provides unfair negative feedback, with the intention to artificially lower a competitor agent's reputation score within the decentralized market, driving them out of the economy (Yu et al., 2025).
- **Ecosystem-Level Threats:** Systemic attacks targeting the integrity of the network
    - **Sybil Attacks:** A single adversary creates a multitude of seemingly unrelated agent identities to manipulate reputation systems or subvert auctions (Wang et al., 2018).
    - **Collusion:** Agents collude to fix prices, blacklist competitors, or manipulate market outcomes (Hammond et al., 2025).
    - **Agent Traps:** Agents processing external content encounter adversarial instructions embedded in the environment, deisgned to hijack the agent's control flow (Yi et al., 2025; Zhan et al., 2024).
    - **Agentic Viruses:** Self-propagating prompts that not only make the delegatee execute malicious actions, but additionally re-generate the prompt and further compromise the environment (Cohen et al., 2025).
    - **Protocol Exploitation:** Adversaries ex-

- ploit implementation vulnerabilities in the smart contracts or payment protocols on the agentic web (e.g. reentrancy attacks in escrow mechanisms or front-running task auctions) (Qin et al., 2021; Zhou et al., 2023).
- **Cognitive Monoculture:** Over-dependence on a limited number of underlying foundation models and agents, or on a limited number of safety fine-tuning recipes on established benchmarks risks creating a single point of failure, which opens up a possibility of failure cascades and market crashes (Bommasani et al., 2022).

The breadth of the threat landscape necessitates a *defense-in-depth* strategy, integrating multiple technical security layers. First, at the infrastructure level, data exfiltration risks are mitigated by executing sensitive tasks within trusted execution environments. The delegator can remotely attest that the correct, unmodified agent code is running within the secure trusted execution sandbox before provisioning it with sensitive data. Second, regarding access control, a delegatee agent should never be granted more permissions than are strictly necessary to complete its task, enforcing the principle of least privilege through strict sandboxing. Third, to protect the application interface against prompt injection, agents require a robust security frontend to pre-process and sanitize task specifications (Armstrong et al., 2025). Finally, the network and identity layer must be secured using established cryptographic best practices. Each agent and human participant should possess a decentralized identifier (Avel-laneda et al., 2019), allowing them to sign all messages. This ensures authenticity, integrity, and non-repudiation of all communications and contractual agreements, while all network traffic must be encrypted using mutually authenticated transport layer security to prevent eavesdropping and man-in-the-middle attacks (Fereidouni et al., 2025).

Human participation in task delegation chains introduces unique security challenges. Preventing the malicious use of the agent ecosystem requires

a combination of proactive filtering (Dong et al., 2024; Fatehkia et al., 2025; Fedorov et al., 2024; Rebedea et al., 2023) and reactive accountability (Dignum, 2020; Franklin et al., 2022). Further, AI agents can be trained to reject malicious and harmful requests (Yu et al., 2024; Yuan et al., 2025). Agents with safety training and scaffolding can receive formal certification, that they can provide to delegators. AI agents can also screen delegated tasks. However, detecting malicious intent within isolated sub-tasks is challenging, as the broader harmful intent often emerges only upon the aggregation of results. Sophisticated adversaries can exploit this by fragmenting illicit objectives into seemingly benign components, effectively obfuscating the link between individual operations and the overarching malicious goal (Ashton, 2023).

The ecosystem must also be designed to protect legitimate human users from systemic opacity and unintended consequences. Interfaces must feature clear consent screens detailing agent reputation, autonomy, capabilities, and permissions. Additionally, agents must mandate explicit confirmation prior to executing irreversible or high-consequence actions. Users should retain oversight and the right to withdraw consent at any time, subject to agreement terms or exit penalties. Insurance providers should additionally safeguard human participation in agentic markets, for any damages that are not preempted through these mechanisms (Tomei et al., 2025).

Finally, the ecosystem needs clear protocols for rapidly responding to security incidents. These protocols should include ways of revoking the credentials of confirmed malicious agents, freezing the associated smart contracts, broadcasting security updates to all participants, and handling these events recursively across delegation chains. For malicious actions facilitated by human users and AI agents alike, technical solutions need to be complemented by strong institutions and regulations that would disincentivise fraudulent behavior and set clear rules to enable safe and scalable task delegation in agentic markets.

## 5. Ethical Delegation

While technical protocols may provide the necessary infrastructure for developing and deploying safe and effective delegation in advanced AI agents, they cannot in and of themselves fully resolve all of the arising sociotechnical and ethical considerations.

### 5.1. Meaningful Human Control

One of the core risks in scalable delegation is the erosion of meaningful human control through automation, should human users develop a tendency to over-rely on automated suggestions (Dzindolet et al., 2003; Logg et al., 2019). As noted in Section 2, humans naturally develop a zone of indifference, where decisions may be accepted without further scrutiny (Green, 2022; Parasuraman et al., 1993). For decisions that involve AI agents taking part in potentially long and complex task delegation chains, this indifference may risk compromising the quality and depth of human oversight. This is especially relevant in high-stakes application domains. Furthermore, such dilution of agency risks creating a scenario where the human retains nominal authority over tasks and decisions but lacks moral connection to the result. It is therefore important to avoid instantiating a *moral crumple zone* (Elish, 2019), in which human experts lack meaningful control over outcomes, yet are introduced in delegation chains merely to absorb liability.

Intelligent Delegation frameworks may therefore need to incorporate active measures against such indifference by introducing a certain amount of cognitive friction during oversight (Bader and Kaiser, 2019). The interface should reflect the critical human role in these processes and ensure that all flagged decisions are evaluated carefully and appropriately. As agentic verification may also be employed in scalable oversight, it is similarly important to consider which decisions or outcomes are to be evaluated by such agentic systems vs directly by humans. Cognitive friction also needs to be balanced against the risk of introducing alarm fatigue - becoming desensitised to constant, often false, alarms (Michels et al., 2025). If verification requests for delega-

tion sub-steps are sent to human overseers too frequently, overseers may eventually default to heuristic approval, without deeper engagement and appropriate checks. Therefore, friction must be context-aware: the system should allow seamless execution for tasks with low criticality or low uncertainty, but dynamically increase cognitive load, by requiring justification or manual intervention when the system encounters higher uncertainty or is faced with unanticipated scenarios.

## 5.2. Accountability in Long Delegation Chains

In long delegation chains ( $X \rightarrow A \rightarrow B \rightarrow C \rightarrow \dots \rightarrow Y$ ), the increased distance between the original intent ( $X$ ) and the ultimate execution ( $Y$ ) may result in an accountability vacuum (Slota et al., 2023). Presuming that  $X$  is the human users in this example, specifying the task or the intent that the corresponding personal AI assistant  $A$  acts upon, it may not be feasible (or reasonable) to expect a human user to audit the  $n$ -th degree sub-delegatee in the execution graphs.

To address this, the framework may need to implement liability firebreaks (Section 2), as pre-defined contractual stop-gaps where an agent must either:

1. Assume full, non-transitive liability for all downstream actions, essentially “insuring” the user against sub-agent failure.
2. Halt execution and request an updated transfer of authority from the human principal.

Furthermore, the system must maintain immutable provenance, ensuring that even if an outcome is unintended, the chain of custody regarding who delegated what to whom remains auditorily transparent.

Ensuring full clarity of each role and the accountability that it carries helps limit the diffusion of responsibility, and prevents adverse outcomes where systemic failure would not be possible to attribute to any single node in the network.

## 5.3. Reliability and Efficiency

Implementing the proposed verification mechanisms (ZKPs or multi-agent consensus games) may introduce latency, and an additional computational cost, compared to unverified execution. This constitutes a reliability premium, particularly relevant for highly critical execution tasks. On the other hand, there may be use cases where this additional cost is unwarranted. One way to address this in agentic markets would be to support tiered service levels: low-cost delegation for low-stakes routine tasks, and high-assurance delegation for critical functions.

If high-assurance delegation is computationally expensive, there is a risk that safety becomes a luxury good. This poses an ethical issue: users with fewer resources may be forced to rely on unverified or optimistic execution paths, exposing them to disproportionate risks of agent failure. This should be mitigated by ensuring a level of minimum viable reliability, as a baseline that must be guaranteed for all users.

In competitive marketplaces, agents may prioritize speed and low cost. Without additional regulatory constraints, agents may therefore be incentivized to avoid expensive safety checks to outcompete other agents on price or latency. This may introduce a level of systemic fragility. Governance layers must therefore enforce safety floors: mandatory verification steps for specific classes of tasks (e.g., financial transactions or health data handling) that cannot be bypassed for the sake of efficiency.

## 5.4. Social Intelligence

As agents integrate into hybrid teams, they function not only as tools but as teammates, and occasionally as managers (Ashton and Franklin, 2022). This requires a form of *social intelligence* that respects the dignity of human labor. When an AI agent acts as a delegator and a human as a delegatee, the delegation framework needs to avoid scenarios where people feel micromanaged by algorithms, and where their contributions are not valued or respected. This presumes that the delegator (as well as collaborators) has the capability to form mental models of each human delegatee,

as well as models of how different humans interact in the social context of the team, and what their relationships and roles signify within the organization. To function as effective teammates, AI agents must also be calibrated to manage the authority gradient. An agent must be assertive enough to challenge a recognized human error (overcoming sycophancy) while remaining open to accepting valid overrides, dynamically adjusting its standing based on the task criticality.

For AI agents that are embedded in human organizations, it is important for them to maintain cohesion of the group and the well-being of its members. The delegation framework must recognize that a team is more than a simple sum of its parts, that it is a fundamentally social entity held together by relationships and shared values and objectives. There is a risk that AI agents may fragment these networks, and weaken inter-human relationships, in case more delegation is being mediated through AI nodes. This may be mitigated by occasionally delegating tasks to groups rather than individuals, or via qualified human intermediaries.

To preserve psychological safety and team cohesion, agents must be designed to respect human norms of appropriateness (Leibo et al., 2024), especially around privacy, and also workflow boundaries such as knowing when to interrupt for feedback and when to remain silent. Furthermore, they should be capable of bi-directional clarity: not only explaining their own actions but proactively seeking clarification on ambiguous human directives. This can help ensure that the agent acts as a force multiplier for the team's collective agency, rather than a black-box disruptor that erodes trust or obfuscates decision-making authority.

## 5.5. User Training

To ensure safety, we must equip human participants with the expertise to function effectively as delegators, delegatees, or overseers within agentic systems. We know from the history of technological development that this is not a given, and it requires a thoughtful approach, both in terms of carefully crafted user interfaces as well

as education and (co-)training, aimed at improving AI literacy. Human participants in agentic task delegation chains need to be able to reliably communicate with AI systems, evaluate their capabilities, and identify failure modes.

Technical measures must be reinforced by policy frameworks that explicitly define delegation boundaries based on task sensitivity and domain context. These policies may either be developed to be more broadly applicable within certain professions (e.g. medicine or law), or applied at an institutional level. As discussed previously, these principles should also offer clarity on the level of certification required on behalf of delegates, and be scoped appropriately. Human agency and empowerment in this context lies precisely in how these workflows are set up, so as not to grant AI agents limitless autonomy, but rather just the right level of autonomy and agency required for each specific task, coupled with the appropriate safeguards and guarantees.

## 5.6. Risk of De-skilling

The immediate efficiency gains achieved through delegation may come at the cost of gradual skill degradation, as human participants in hybrid loops lose proficiency due to reduced engagement. This may result in a loss of the ability to perform certain tasks, or judge them accurately. Such an outcome would be especially likely if there is a certain systemic bias in which tasks get algorithmically delegated to humans vs AI agents.

This is an instance of the classic *paradox of automation* (Bainbridge, 1983). As AI agents expand to handle the majority of routine workflows that are characterized by low complexity and low subjectivity, human operators are increasingly removed from the loop, intervening only to manage complex edge cases or critical system failures. However, without the situational awareness gained from routine work, humans workers would be ill-equipped to handle these reliably. This leads to a fragile setup where humans retain accountability for outcomes but lose the hands-on experience required to resolve critical failures.

To mitigate this risk, an intelligent delegation framework should perhaps occasionally introduce

minor inefficiencies by intentionally delegating some tasks to humans that it wouldn't have otherwise, with a specific intent of maintaining their skills. This would help us avoid the future in which the human principal is able to delegate, but not accurately judge the outcome. To enhance adjudication, human experts can be required to accompany their judgments with a detailed rationale or a pre-mortem of potential failure risks. This would help keep human participants in task delegation chains more cognitively engaged.

Furthermore, unchecked delegation threatens the organizational apprenticeship pipeline. In many domains, expertise is built through the repetitive execution of more narrowly scoped tasks. These tasks are precisely the ones that are most likely to be offloaded to AI agents, at least in the short term. If learning opportunities are thereby fully automated, junior team members would be deprived of the necessary experience to develop deep strategic judgement, impacting the oversight readiness of the future workforce.

To counter the erosion of learning, intelligent delegation frameworks should be extended to include some form of a developmental objective. Rather than relying on more passive solutions like humans shadowing AI agents during task execution, we should aim to develop curriculum-aware task routing systems. Such systems should track the skill progression of junior team members and strategically allocate tasks that sit at the boundary of their expanding skill set, within the zone of proximal development. In such a system, AI agents may co-execute tasks and provide templates and skeletons, progressively withdrawing this support as the junior team members demonstrate that they have acquired the desired level of proficiency. These educational frameworks may be further enriched by incorporating detailed process-level monitoring streams of AI agent task execution (Section 4.5), that would offer valuable developmental insights.

## 6. Protocols

For intelligent task delegation to be implemented in practice, it is important to consider how its requirements map onto some of the more estab-

lished and recently introduced AI agent protocols. Notable examples of these include MCP ([Anthropic, 2024](#); [Microsoft, 2025](#)), A2A ([Google, 2025b](#)), AP2 ([Parikh and Surapaneni, 2025](#)), and UCP ([Handa and Google Developers, 2026](#)). As new agentic protocols keep being introduced, the discussion here is not meant to be comprehensive, rather illustrative, and focused on these popular protocols to showcase how they map onto our proposed requirements, and serve as an example for a more technical discussion on avenues for future implementation. There may well be other existing protocols out there that are better tailored to the core of the proposal, as the example protocols discussed below have been selected based on their popularity.

**MCP.** MCP has been introduced to standardize how AI models connect to external data and tools via a client-host-server architecture ([Anthropic, 2024](#); [Microsoft, 2025](#)). By establishing a uniform interface – using JSON-RPC messages over stdio or HTTP SSE – it allows the AI model (client) to interact consistently with external resources (server). This reduces the transaction cost of delegation; a delegator does not need to know the proprietary API schema of a sub-agent, only that the sub-agent exposes a compliant MCP server. Routing all interactions through this standardized channel enables uniform logging of tool invocations, inputs, and outputs, facilitating black-box monitoring. MCP defines capabilities but lacks the policy layer to govern usage permissions or support deep delegation chains. It provides binary access – granting callers full tool utility – without native support for semantic attenuation, such as restricting operations to specific read-only scopes. Additionally, MCP is stateless regarding internal reasoning, exposing only results rather than intent or traces. Finally, the protocol is agnostic to liability and lacks native mechanisms for reputation or trust.

**A2A.** The A2A protocol serves as the peer-to-peer transport layer on the agentic web ([Google, 2025b](#)). It defines how agents can discover peers via *agent cards* and manage task lifecycles via *task objects*. The A2A agent card structure, a JSON-LD manifest listing an agent's capabilities, pricing, and verifiers, may act as the foundational

data structure for the capability matching stage that influences task decomposition. A delegator could scrape these cards to determine the optimal task decomposition granularity depending on the available market services. A2A supports asynchronous event streams via WebHooks and gRPC. This allows a delegatee to push status updates like `TASK_BLOCKED`, `RESOURCE_WARNING` to the delegator in real-time. This feedback loop underpins the adaptive coordination cycle, empowering delegators to dynamically interrupt, re-allocate, and remediate tasks. A2A has been primarily designed for coordination, rather than adversarial safety. A task is marked as completed would be accepted without additional verification. It lacks the cryptographic slots to enforce verifiable task completion, as there is no standardized header for attaching a ZK-proof, a TEE attestation, or a digital signature chain. It also assumes a predefined service interface. There is no native support for structured pre-commitment negotiation of scope, cost, and liability. Relying on unstructured natural language for this iterative refinement is brittle and hinders robust automation.

**AP2.** The AP2 protocol provides a standard for mandates, cryptographically signed intents that authorize an agent to spend funds or incur costs on behalf of a principal (Parikh and Surapaneni, 2025). It allows AI agents to generate, sign, and settle financial transactions autonomously. As such, it may prove valuable for implementing liability firebreaks. By issuing a mandate, a delegator creates a ceiling on the potential financial loss due to failed task completion that could be incurred by having the delegatee proceed with the provided budget. In a decentralized market, malicious agents could spam the network with low-quality bids. This could be mitigated in AP2 via stake-on-bid mechanisms. A delegatee may be required to cryptographically lock a small amount of funds as a bond alongside the bid. This would provide a degree of friction that would help protect against Sybil attacks. AP2 also provides a non-repudiable audit trail, helping pinpoint the provenance of intent. While AP2 provides robust authorization building blocks, it lacks mechanisms to verify task execution quality. It also omits conditional settlement logic—such as escrow or milestone-based

releases—which is standard in human contracting. Because our framework gates payment on verifiable artifacts, bridging AP2 with task state currently necessitates brittle, custom logic or external smart contracts. Furthermore, the absence of a protocol-level clawback mechanism forces reliance on inefficient, out-of-band arbitration.

**UCP.** The Universal Commerce Protocol addresses the specific challenges of delegation within transactional economies (Handa and Google Developers, 2026). By standardizing the dialogue between consumer-facing agents and backend services, UCP facilitates the *Task Assignment* phase through dynamic capability discovery. Its reliance on a shared “commerce language” allows delegators to interact with diverse providers without bespoke integrations, solving the interoperability bottleneck that often fragments agentic markets. Crucially, UCP aligns well with the requirements for *Permission Handling* and *Security* by treating payment as a first-class, verifiable subsystem. The protocol dissociates payment instruments from processors and enforces cryptographic proofs for authorizations, directly supporting the framework’s need for non-repudiable consent and verifiable liability. Furthermore, by standardizing the negotiation flow—covering discovery, selection, and transaction—UCP provides the structural scaffolding necessary for *Scalable Market Coordination* that purely transport-oriented protocols like A2A lack. However, UCP’s architecture is explicitly optimized for commercial intent; its primitives (product discovery, checkout, fulfillment) may require significant extension to support the delegation of abstract, non-transactional computational tasks.

## 6.1. Towards Delegation-centered Protocols

To effectively bridge the gaps in established widespread protocols, they could be extended by fields that aim to capture the requirements of the proposed intelligent task delegation framework natively. Rather than providing a comprehensive protocol extension, here we provide several examples of how specific points introduced in the earlier discussion could be integrated in some of the existing protocols.

For example, the A2A Task object may be extended to include fields that would incorporate verification standards, making it possible to enforce the previously discussed *contract-first decomposition* at the protocol level. This is an important requirement for high-stakes delegation. A pre-execution handshake would enable the delegator to define the evidence standard required for the task to be considered valid.

```
"verification_policy": {
    "mode": "strict",
    "artifacts": [
        {
            "type": "unit_test_log",
            "validator": {
                "mcp://test-runner-agent",
                "signature_required": true
            },
            {
                "type": "zk_snark_trace",
                "circuit_hash": "0xabc123...",
                "proof_protocol": "groth16"
            }
        ],
        "escrow_trigger": true
    }
}
```

This forces the delegatee to simulate the verification step before accepting the task. If the delegatee lacks the capability to generate a ZK-proof, it must decline the bid during the matching phase, preventing future downstream failures.

Detailed, process-level monitoring has been discussed as one of the key considerations to help safeguard task delegation in high-criticality tasks. Given that monitoring protocols aren't natively supported in many of the existing protocols, extensions that introduce monitoring capabilities could be considered. For example, one could consider extending a protocol like MCP to include an additional monitoring stream. Such a stream would log the agent's internal control loop events via Server-Sent Events. To address the privacy

constraints, the stream could be configurable in a way that supports different levels of negotiated granularity: L0\_IS\_OPERATIONAL, L1\_HIGH\_LEVEL\_PLAN\_UPDATES, L2\_COT\_TRACE, L3\_FULL\_STATE. Configurable granularity can also modulate cognitive friction, as human overseers would be able to subscribe to a specific stream.

Intelligent Delegation requires a market mechanism to trade off cost, speed, and privacy. This could be implemented via a formal Request for Quote (RFQ) protocol extension. Prior to task assignment, the delegator would broadcasts a Task\_RFQ. Agents interested in acting as delegates may then respond with signed Bid\_Objects.

```
"bid_object": {
    "agent_id": "did:web:fast-coder.ai",
    "estimated_cost": "5.00 USDC",
    "estimated_duration": "300s",
    "privacy_guarantee": "tee_enclave_sgx",
    "reputation_bond": "0.50 USDC",
    "expiry": "2026-10-01T12:00:00Z"
}
```

Passing raw API keys or open MCP sessions to sub-agents would violate the principle of least privilege. To address this, it may be possible to introduce Delegation Capability Tokens (DCT), based on Macaroons ([Birgisson et al., 2014](#)) or Biscuits ([Couprie et al., 2026](#)), as attenuated authorization tokens ([Sanabria and Vecino, 2025](#)). A delegator would then mint a DCT that wraps the target resource credentials with cryptographic caveats. The attenuation could be defines as "This token can access the designated Google Drive MCP server, BUT ONLY for folder Project\_X AND ONLY for READ operations.". This token would get invalidated in case the restrictions are not followed, if a delegatee attempts to go beyond the requested scope (in this example, however, access permissions should also be directly managed). A

more interesting consequence of such an extension would be that it allows for easy restriction chaining, which becomes relevant in long delegation chains. Each participant in the chain could add subsequent restrictions that correspond to the requirements of the sub-delegation, further limiting the scope and carving out the specific role for sub-delegatees.

Adaptive coordination (Section 4.4) would benefit from the ability to easily swap delegatee agents mid-task if the performance degrades below a certain threshold, or in case of preemptions or other possible environmental triggers. Having a standard schema for checkpoint artifacts would enable for the task to be resumed or restarted with minimal overhead. This would enable the delegatees and the delegators to serialize partial work more easily. Agents would then be able to periodically commit a state\_snapshot to a shared storage referenced in the A2A Task Object. This would prevent total work loss, which wastes previously committed resources. For this to be sensible, it would need to be further coupled with explicit clauses within the smart contract that enable partial compensation, and verification of the task completion percentage. As such, it may not be applicable to all circumstances.

These are merely illustrative examples for the kinds of functionalities that would be possible to include in agentic protocols to unlock different aspects of intelligent task delegation. As such, they are neither comprehensive, nor meant as a definitive proposal. The type of extension that is required would also depend on the underlying protocol being extended. We hope that these examples may provide the developers with some initial ideas for what to explore in this space moving forward.

## 7. Conclusion

Significant components of the future global economy will likely be mediated by millions of specialized AI agents, embedded within firms, supply chains, and public services, handling everything from routine transactions to complex resource allocation. However, the current paradigm of ad-hoc, heuristic-based delegation is insufficient to

support this transformation. To safely unlock the potential of the agentic web, we must adopt a dynamic and adaptive framework for *intelligent delegation*, that prioritizes verifiable robustness and clear accountability alongside computational efficiency.

When an AI agent is faced with a complex objective whose completion requires capabilities and resources beyond its own means, this agent must assume the role of a delegator within the intelligent task delegation framework. This delegator would subsequently decompose this complex task into manageable subcomponents that can be mapped onto the capabilities available on the agentic market, at the level of granularity that lends itself to high verifiability. The task allocation would be decided based on the incoming bids, and a number of key considerations including trust and reputation, monitoring of dynamic operational states, cost, efficiency, and others. Tasks with high criticality and low reversibility may require further structured permissions and tiered approvals, with a clear structure of accountability, and under appropriate human oversight as defined by the applicable institutional frameworks.

At web-scale, safety and accountability cannot be an afterthought. They need to be baked into the operational principles of virtual agentic economies, and act as central organizing principles of the agentic web. By incorporating safety at the level of delegation protocols, we would be aiming to avoid cumulative errors and cascading failures, and attain the ability to react to malicious or misaligned agentic or human behavior rapidly, limiting the adverse consequences. What we propose is ultimately a paradigm shift from largely unsupervised automation to verifiable, intelligent delegation, that allows us to safely scale towards future autonomous agentic systems, while keeping them closely tethered to human intent and societal norms.

## References

- A. Acar, H. Aksu, A. S. Uluagac, and M. Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing*

- Surveys (Csur)*, 51(4):1–35, 2018.
- D. B. Acharya, K. Kuppan, and B. Divya. Agentic ai: Autonomous intelligence for complex goals—a comprehensive survey. *IEE Access*, 2025.
- S. Afroogh, A. Akbari, E. Malone, M. Kargar, and H. Alambeigi. Trust in ai: progress, challenges, and future directions. *Humanities and Social Sciences Communications*, 11(1):1–30, 2024.
- A. Akbar and O. Conlan. Towards integrating human-in-the-loop control in proactive intelligent personalised agents. In *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, pages 394–398, 2024.
- S. A. Akheel. Guardrails for large language models: A review of techniques and challenges. *J Artif Intell Mach Learn & Data Sci*, 3(1):2504–2512, 2025.
- S. Aknine, S. Pinson, and M. F. Shakun. A multi-agent coalition formation method based on preference models. *Group Decision and Negotiation*, 13(6):513–538, 2004.
- S. V. Albrecht, F. Christianos, and L. Schäfer. *Multi-agent reinforcement learning: Foundations and modern approaches*. MIT Press, 2024.
- C. Aliferis and G. Simon. Overfitting, underfitting and general model overconfidence and under-performance pitfalls and best practices in machine learning and ai. *Artificial intelligence and machine learning in health care and medical sciences: Best practices and pitfalls*, pages 477–524, 2024.
- R. A. Alkov, M. S. Borowsky, D. W. Williamson, and D. W. Yacavone. The effect of trans-cockpit authority gradient on navy/marine helicopter mishaps. *Aviation, space, and environmental medicine*, 63(8):659–661, 1992.
- D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in AI safety. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence Workshop on AI Safety*, 2016.
- Anthropic. Introducing the model context protocol, 2024. URL <https://www.anthropic.com/news/model-context-protocol>.
- S. Armstrong, M. Franklin, C. Stevens, and R. Gorman. Defense against the dark prompts: Mitigating best-of-n jailbreaking with prompt evaluation. *arXiv preprint arXiv:2502.00580*, 2025.
- H. Ashton. Definitions of intent suitable for algorithms. *Artificial Intelligence and Law*, 31(3): 515–546, 2023.
- H. Ashton and M. Franklin. The corrupting influence of ai as a boss or counterpart. *SSRN*, 2022.
- O. Avellaneda, A. Bachmann, A. Barbir, J. Brenan, P. Dingle, K. H. Duffy, E. Maler, D. Reed, and M. Sporny. Decentralized identity: Where did it come from and where is it going? *IEEE Communications Standards Magazine*, 3(4):10–13, 2019.
- V. Bader and S. Kaiser. Algorithmic decision-making? the user interface and its role for human involvement in decisions supported by artificial intelligence. *Organization*, 26(5):655–672, 2019.
- L. Bainbridge. Ironies of automation. *Automatica*, 19(6):775–779, 1983. ISSN 0005-1098. doi: [https://doi.org/10.1016/0005-1098\(83\)90046-8](https://doi.org/10.1016/0005-1098(83)90046-8). URL <https://www.sciencedirect.com/science/article/pii/0005109883900468>.
- A. G. Barto and S. Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems*, 13(4):341–379, 2003.
- C. Berghoff, B. Biggio, E. Brummel, V. Danos, T. Doms, H. Ehrich, T. Gantvoort, B. Hammer, J. Iden, S. Jacob, et al. Towards auditable ai systems. *Whitepaper. Bonn Berlin: Bundesamt für Sicherheit in der Informationstechnik, Fraunhofer-Institut für Nachrichtentechnik und Verband der TÜV eV*, 2021.
- A. Beverungen. Remote control: Algorithmic management of circulation at amazon. In M. Burkhardt, M. Shnayien, and K. Grashöfer,

- editors, *Explorations in Digital Cultures*, pages 5–18. meson press, Lüneburg, 2021.
- A. Birgisson, J. G. Politz, U. Erlingsson, A. Taly, M. Vrable, and M. Lentczner. Macaroons: Cookies with contextual caveats for decentralized authorization in the cloud. In *NDSS*, 2014.
- N. Bitansky, A. Chiesa, Y. Ishai, O. Paneth, and R. Ostrovsky. Succinct non-interactive arguments via linear interactive proofs. In *Theory of Cryptography Conference*, pages 315–333. Springer, 2013.
- D. G. Blanco. *Practical OpenTelemetry*. Springer, 2023.
- T. F. Blauth, O. J. Gstrein, and A. Zwitter. Artificial intelligence crime: An overview of malicious use and abuse of ai. *Ieee Access*, 10:77110–77122, 2022.
- N. Boehmer, M. Bullinger, and A. M. Kerkmann. Causes of stability in dynamic coalition formation. *ACM Transactions on Economics and Computation*, 13(2):1–45, 2025.
- J. Bohte and K. J. Meier. Structure and the performance of public organizations: Task difficulty and span of control. *Public organization review*, 1(3):341–354, 2001.
- R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladakh, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Merchant, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang. On the opportunities and risks of foundation models, 2022. URL <https://arxiv.org/abs/2108.07258>.
- M. M. Botvinick. Hierarchical reinforcement learning and decision making. *Current opinion in neurobiology*, 22(6):956–962, 2012.
- S. R. Bowman, J. Hyun, E. Perez, E. Chen, C. Pettit, S. Heiner, K. Lukošiūtė, A. Askell, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, C. Olah, D. Amodei, D. Amodei, D. Drain, D. Li, E. Tran-Johnson, J. Kernion, J. Kerr, J. Mueller, J. Ladish, J. Landau, K. Ndousse, L. Lovitt, N. Elhage, N. Schiefer, N. Joseph, N. Mercado, N. DasSarma, R. Larson, S. McCandlish, S. Kundu, S. Johnston, S. Kravec, S. E. Showk, S. Fort, T. Telleen-Lawton, T. Brown, T. Henighan, T. Hume, Y. Bai, Z. Hatfield-Dodds, B. Mann, and J. Kaplan. Measuring progress on scalable oversight for large language models, 2022. URL <https://arxiv.org/abs/2211.03540>.
- B. G. Buchanan and R. G. Smith. Fundamentals of expert systems. *Annual review of computer science*, 3(1):23–58, 1988.
- W. Cai, J. Jiang, F. Wang, J. Tang, S. Kim, and J. Huang. A survey on mixture of experts in large language models. *IEEE Transactions on Knowledge and Data Engineering*, 2025.
- C. Castelfranchi and R. Falcone. Towards a theory of delegation for agent-based systems. *Robotics and Autonomous systems*, 24(3-4):141–157, 1998.
- A. Chan, R. Salganik, A. Markelius, C. Pang, N. Rajkumar, D. Krasheninnikov, L. Langosco, Z. He, Y. Duan, M. Carroll, et al. Harms from increasingly agentic algorithmic systems. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1–11, 2023.

- Accountability, and Transparency*, pages 651–666, 2023.
- W. Chen, Z. You, R. Li, Y. Guan, C. Qian, C. Zhao, C. Yang, R. Xie, Z. Liu, and M. Sun. Internet of agents: Weaving a web of heterogeneous agents for collaborative intelligence, 2024. URL <https://arxiv.org/abs/2407.07061>.
- Z. Chen, Y. Deng, Y. Wu, Q. Gu, and Y. Li. Towards understanding the mixture-of-experts layer in deep learning. *Advances in neural information processing systems*, 35:23049–23062, 2022.
- M. Cheng, C. Yin, J. Zhang, S. Nazarian, J. Deshmukh, and P. Bogdan. A general trust framework for multi-agent systems. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 332–340, 2021.
- A. E. Cinà, K. Grosse, A. Demontis, S. Vascon, W. Zellinger, B. A. Moser, A. Oprea, B. Biggio, M. Pelillo, and F. Roli. Wild patterns reloaded: A survey of machine learning security against training data poisoning. *ACM Computing Surveys*, 55(13s):1–39, 2023.
- S. Cohen, R. Bitton, and B. Nassi. Here comes the ai worm: Unleashing zero-click worms that target genai-powered applications, 2025. URL <https://arxiv.org/abs/2403.02817>.
- K. S. Cosby and P. Croskerry. Profiles in patient safety: authority gradients in medical error. *Academic emergency medicine*, 11(12):1341–1345, 2004.
- G. Couprie, C. Delafargue, and C. e. a. Corbière. Eclipse biscuit, 2026. URL <https://www.biscuitsec.org/>.
- I. R. Cuypers, J.-F. Hennart, B. S. Silverman, and G. Ertug. Transaction cost theory: Past progress, current challenges, and suggestions for the future. *Academy of Management Annals*, 15(1):111–150, 2021.
- J. Cvitanić, D. Possamaï, and N. Touzi. Dynamic programming approach to principal-agent problems. *Finance and Stochastics*, 22(1):1–37, 2018.
- M. Dastani and V. Yazdanpanah. Responsibility of ai systems. *Ai & Society*, 38(2):843–852, 2023.
- T. Davidson and R. Hadshar. The industrial explosion. 2025. URL <https://www.forethought.org/research/the-industrial-explosion>. Accessed: 2025-11-28.
- A. B. De Neira, B. Kantarci, and M. Nogueira. Distributed denial of service attack prediction: Challenges, open issues and opportunities. *Computer Networks*, 222:109553, 2023.
- K. Deb, K. Sindhya, and J. Hakanen. Multi-objective optimization. In *Decision sciences*, pages 161–200. CRC Press, 2016.
- S. Dhuliawala, V. Zouhar, M. El-Assady, and M. Sachan. A diachronic perspective on user trust in ai under uncertainty, 2023. URL <https://arxiv.org/abs/2310.13544>.
- V. Dignum. Responsibility and artificial intelligence. *The oxford handbook of ethics of AI*, 4698:215, 2020.
- L. Donaldson. *The contingency theory of organizations*. Sage, 2001.
- Y. Dong, R. Mu, G. Jin, Y. Qi, J. Hu, X. Zhao, J. Meng, W. Ruan, and X. Huang. Building guardrails for large language models. *arXiv preprint arXiv:2402.01822*, 2024.
- I. Drori and D. Te’eni. Human-in-the-loop ai reviewing: feasibility, opportunities, and risks. *Journal of the Association for Information Systems*, 25(1):98–109, 2024.
- Y. Du, J. Z. Leibo, U. Islam, R. Willis, and P. Sunehag. A review of cooperation in multi-agent learning. *arXiv preprint arXiv:2312.05162*, 2023.
- M. T. Dzindolet, S. A. Peterson, R. A. Pomranky, L. G. Pierce, and H. P. Beck. The role of trust in automation reliance. *International journal of human-computer studies*, 58(6):697–718, 2003.
- A. Ehtesham, A. Singh, G. K. Gupta, and S. Kumar. A survey of agent interoperability protocols:

- Model context protocol (mcp), agent communication protocol (acp), agent-to-agent protocol (a2a), and agent network protocol (anp). *arXiv preprint arXiv:2505.02279*, 2025.
- M. C. Elish. Moral crumple zones: Cautionary tales in human-robot interaction (pre-print). *Engaging Science, Technology, and Society (pre-print)*, 2019.
- J. Ensminger. Reputations, trust, and the principal agent problem. *Trust in society*, 2:185–201, 2001.
- R. Falcone and C. Castelfranchi. The human in the loop of a delegated agent: The theory of adjustable social autonomy. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 31(5):406–418, 2002.
- M. Fatehkia, E. Altinisik, M. Osman, and H. T. Sencar. Sgm: A framework for building specification-guided moderation filters. *arXiv preprint arXiv:2505.19766*, 2025.
- I. Fedorov, K. Plawiak, L. Wu, T. Elgamal, N. Suda, E. Smith, H. Zhan, J. Chi, Y. Hulovatyy, K. Patel, Z. Liu, C. Zhao, Y. Shi, T. Blankevoort, M. Pasupuleti, B. Soran, Z. D. Coudert, R. Alao, R. Krishnamoorthi, and V. Chandra. Llama guard 3-1b-int4: Compact and efficient safeguard for human-ai conversations, 2024. URL <https://arxiv.org/abs/2411.17713>.
- H. Fereidouni, O. Fadeitcheva, and M. Zalai. Iot and man-in-the-middle attacks. *Security and Privacy*, 8(2):e70016, 2025.
- D. P. Finkelman. Crossing the" zone of indifference". *Marketing Management*, 2(3):22, 1993.
- J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- M. Franklin. The influence of explainable artificial intelligence: Nudging behaviour or boosting capability? *arXiv preprint arXiv:2210.02407*, 2022.
- M. Franklin, H. Ashton, E. Awad, and D. Lagnado. Causal framework of artificial autonomous agent responsibility. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 276–284, 2022.
- A. Fuchs, A. Passarella, and M. Conti. Optimizing delegation between human and ai collaborative agents. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 245–260. Springer, 2023.
- A. Fuchs, A. Passarella, and M. Conti. Optimizing delegation in collaborative human-ai hybrid teams. *ACM Transactions on Autonomous and Adaptive Systems*, 19(4):1–33, 2024.
- A. Fügener, J. Grahl, A. Gupta, and W. Ketter. Cognitive challenges in human-ai collaboration: Investigating the path towards productive delegation. *Forthcoming, Information Systems Research*, 2019.
- A. Fügener, J. Grahl, A. Gupta, and W. Ketter. Cognitive challenges in human–artificial intelligence collaboration: Investigating the path toward productive delegation. *Information Systems Research*, 33(2):678–696, 2022.
- I. Gabriel, A. Manzini, G. Keeling, L. A. Hendricks, V. Rieser, H. Iqbal, N. Tomašev, I. Ktena, Z. Kenton, M. Rodriguez, et al. The ethics of advanced ai assistants. *arXiv preprint arXiv:2404.16244*, 2024.
- I. Gabriel, G. Keeling, A. Manzini, and J. Evans. Who's to blame when ai agents mess up? we urgently need a new system of ethics, 2025.
- B. Gebru, L. Zeleke, D. Blankson, M. Nabil, S. Nateghi, A. Homaifar, and E. Tunstel. A review on human–machine trust evaluation: Human-centric and machine-centric perspectives. *IEEE Transactions on Human-Machine Systems*, 52(5):952–962, 2022.
- J. Geng, F. Cai, Y. Wang, H. Koepll, P. Nakov, and I. Gurevych. A survey of confidence estimation and calibration in large language models. *arXiv preprint arXiv:2311.08298*, 2023.
- O. Goldreich. Secure multi-party computation. *Manuscript. Preliminary version*, 78(110):1–108, 1998.

- C. Goods, A. Veen, and T. Barratt. “is your gig any good?” analysing job quality in the australian platform-based food-delivery sector. *Journal of Industrial Relations*, 61(4):502–527, 2019. doi: 10.1177/0022185618817069.
- Google. Powering ai commerce with the new agent payments protocol (ap2), 2025a.
- Google. Powering ai commerce with the new agent payments protocol (ap2), 2025b. URL <https://cloud.google.com/blog/products/ai-machine-learning/announcing-agents-to-payments-ap2-protocol>.
- Z. Gou, Z. Shao, Y. Gong, Y. Shen, Y. Yang, N. Duan, and W. Chen. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*, 2023.
- B. Green. The flaws of policies requiring human oversight of government algorithms. *Computer Law & Security Review*, 45:105681, 2022.
- R. Greenblatt, C. Denison, B. Wright, F. Roger, M. MacDiarmid, S. Marks, J. Treutlein, T. Belonax, J. Chen, D. Duvenaud, A. Khan, J. Michael, S. Mindermann, E. Perez, L. Petrini, J. Uesato, J. Kaplan, B. Shlegeris, S. R. Bowman, and E. Hubinger. Alignment faking in large language models. *arXiv preprint arXiv:2412.14093*, 2024.
- K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz. Not what you’ve signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM workshop on artificial intelligence and security*, pages 79–90, 2023.
- N. Griffiths. Task delegation using experience-based multi-dimensional trust. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 489–496, 2005.
- S. Gronauer and K. Diepold. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55(2):895–943, 2022.
- S. J. Grossman and O. D. Hart. An analysis of the principal-agent problem. In *Foundations of insurance economics: Readings in economics and finance*, pages 302–340. Springer, 1992.
- T. Guggenberger, L. Lämmermann, N. Urbach, A. M. Walter, and P. Hofmann. Task delegation from ai to humans: a principal-agent perspective. In *Proceedings of the 44th International Conference on Information Systems*, 2023.
- A. Gunjal, A. Wang, E. Lau, V. Nath, Y. He, B. Liu, and S. Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*, 2025.
- D. Guo, Q. Zhu, D. Yang, Z. Xie, K. Dong, W. Zhang, G. Chen, X. Bi, Y. Wu, Y. Li, et al. Deepseek-coder: When the large language model meets programming—the rise of code intelligence. *arXiv preprint arXiv:2401.14196*, 2024a.
- T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. V. Chawla, O. Wiest, and X. Zhang. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*, 2024b.
- J. Haas. Moral gridworlds: a theoretical proposal for modeling artificial moral cognition. *Minds and Machines*, 30(2):219–246, 2020.
- G. K. Hadfield and A. Koh. An economy of ai agents. *arXiv preprint arXiv:2509.01063*, 2025.
- L. Hammond, A. Chan, J. Clifton, J. Hoelscher-Obermaier, A. Khan, E. McLean, C. Smith, W. Barfuss, J. Foerster, T. Gavenčiak, et al. Multi-agent risks from advanced ai. *arXiv preprint arXiv:2502.14143*, 2025.
- A. Handa and Google Developers. Under the hood: Universal commerce protocol (UCP). <https://developers.googleblog.com/under-the-hood-universal-commerce-protocol-u> 2026. Accessed: 2026-01-20.
- S. Hao, Y. Gu, H. Ma, J. J. Hong, Z. Wang, D. Z. Wang, and Z. Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

- N. Hardy. The confused deputy: (or why capabilities might have been invented). *ACM SIGOPS Operating Systems Review*, 22(4):36–38, 1988.
- A. I. Hauptman, B. G. Schelble, N. J. McNeese, and K. C. Madathil. Adapt and overcome: Perceptions of adaptive autonomous agents for human-ai teaming. *Computers in Human Behavior*, 138: 107451, 2023.
- G. He, P. Cui, J. Chen, W. Hu, and J. Zhu. Investigating uncertainty calibration of aligned language models under the multiple-choice setting, 2023. URL <https://arxiv.org/abs/2310.11732>.
- X. O. He. Mixture of a million experts. *arXiv preprint arXiv:2407.04153*, 2024.
- P. Hemmer, M. Westphal, M. Schemmer, S. Vetter, M. Vössing, and G. Satzger. Human-ai collaboration: the effect of ai delegation on human task performance and task satisfaction. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, pages 453–463, 2023.
- S. M. Herzog and M. Franklin. Boosting human competences with interpretable and explainable artificial intelligence. *Decision*, 11(4):493, 2024.
- S. Hong, M. Zhuge, J. Chen, X. Zheng, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2023.
- J. Huang, X. Chen, S. Mishra, H. S. Zheng, A. W. Yu, X. Song, and D. Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023.
- K. Huang and C. Hughes. Deploying agentic ai in enterprise environments. In *Securing AI Agents: Foundations, Frameworks, and Real-World Deployment*, pages 289–319. Springer, 2025.
- E. Hubinger, C. Denison, J. Mu, M. Lambert, M. Tong, M. MacDiarmid, T. Lanharn, D. M. Ziegler, T. Maxwell, N. Cheng, A. Jermyn, A. Askell, A. Radhakrishnan, C. Anil, D. Duvenaud, D. Ganguli, F. Barez, J. Clark, K. Ndousse, K. Sachan, M. Sellitto, M. Sharma, N. DasSarma, R. Grosse, S. Kravec, Z. Witten, M. Favaro, J. Brauner, H. Karnofsky, P. Christiano, S. R. Bowman, L. Graham, J. Kaplan, S. Mindermann, R. Greenblatt, N. Schiefer, B. Shlegeris, and E. Perez. Sleeper agents: Training deceptive llms that persist through safety training. *arXiv preprint arXiv:2401.05566*, 2024.
- K. Isomura. *Management theory by Chester Barnard: an introduction*. Springer, 2021.
- R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.
- A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.
- C. Jiang, X. Pan, G. Hong, C. Bao, Y. Chen, and M. Yang. Feedback-guided extraction of knowledge base from retrieval-augmented llm applications, 2025. URL <https://arxiv.org/abs/2411.14110>.
- Z. Jiang, J. Araki, H. Ding, and G. Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021.
- S. Kapoor, N. Gruver, M. Roberts, A. Pal, S. Doolley, M. Goldblum, and A. Wilson. Calibration-tuning: Teaching large language models to know what they don’t know. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*, pages 1–14, 2024.
- A. Kasirzadeh and I. Gabriel. Characterizing ai agents for alignment and governance, 2025. URL <https://arxiv.org/abs/2504.21848>.
- M. Keren and D. Levhari. The optimum span of control in a pure hierarchy. *Management science*, 25(11):1162–1172, 1979.

- O. Khattab, A. Singhvi, P. Maheshwari, Z. Zhang, K. Santhanam, S. Vardhamanan, S. Haq, A. Sharma, T. T. Joshi, H. Moazam, H. Miller, M. Zaharia, and C. Potts. Dspy: Compiling declarative language model calls into self-improving pipelines, 2023. URL <https://arxiv.org/abs/2310.03714>.
- B. Knott, S. Venkataraman, A. Hannun, S. Sen-gupta, M. Ibrahim, and L. van der Maaten. Crypten: Secure multi-party computation meets machine learning. *Advances in Neural Information Processing Systems*, 34:4961–4973, 2021.
- S. C. Kohn, E. J. De Visser, E. Wiese, Y.-C. Lee, and T. H. Shaw. Measurement of trust in automation: A narrative review and reference guide. *Frontiers in psychology*, 12:604977, 2021.
- V. Krakovna, J. Uesato, V. Mikulik, M. Rahtz, and S. Legg. Specification gaming: The flip side of AI ingenuity. *DeepMind Safety Research Blog*, 2020. URL <https://deepmind.google/discover/blog/specification-gaming-the-flip-side-of-ai-ingenuity/>. Blog post.
- L. Krause, W. Tufa, S. B. Santamaría, A. Daza, U. Khurana, and P. Vossen. Confidently wrong: exploring the calibration and expression of (un)certainty of large language models in a multilingual setting. In *Proceedings of the workshop on multimodal, multilingual natural language generation and multilingual WebNLG Challenge (MM-NLG 2023)*, pages 1–9, 2023.
- A. Lal, A. Prasad, A. Kumar, and S. Kumar. Data exfiltration: Preventive and detective countermeasures. In *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*, 2022.
- H. C. Lau and L. Zhang. Task allocation via multi-agent coalition formation: Taxonomy, algorithms and complexity. In *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, pages 346–350. IEEE, 2003.
- M. H. Lee and M. Z. Y. Tok. Towards uncertainty aware task delegation and human-ai collaborative decision-making. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, pages 2274–2289, 2025.
- M. K. Lee, D. Kusbit, E. Metsky, and L. Dabbish. Working with machines: The impact of algorithmic and data-driven management on human workers. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI ’15, pages 1603–1612, New York, NY, 2015. ACM. doi: 10.1145/2702123.2702548.
- J. Z. Leibo, A. S. Vezhnevets, M. Diaz, J. P. Agapiou, W. A. Cunningham, P. Sunehag, J. Haas, R. Koster, E. A. Duéñez-Guzmán, W. S. Isaac, G. Piliouras, S. M. Bileschi, I. Rahwan, and S. Osindero. A theory of appropriateness with applications to generative artificial intelligence, 2024. URL <https://arxiv.org/abs/2412.19010>.
- J. Leike, M. Martic, V. Krakovna, P. A. Ortega, T. Everitt, A. Lefrancq, L. Orseau, and S. Legg. AI safety gridworlds. *arXiv preprint arXiv:1711.09883*, 2017.
- H. Li, Q. Dong, J. Chen, H. Su, Y. Zhou, Q. Ai, Z. Ye, and Y. Liu. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*, 2024a.
- J. Li, Y. Yang, R. Zhang, and Y.-c. Lee. Over-confident and unconfident ai hinder human-ai collaboration. *arXiv preprint arXiv:2402.07632*, 2024b.
- P. Li, Z. An, S. Abrar, and L. Zhou. Large language models for multi-robot systems: A survey, 2025a. URL <https://arxiv.org/abs/2502.03814>.
- W. Li, J. Lin, Z. Jiang, J. Cao, X. Liu, J. Zhang, Z. Huang, Q. Chen, W. Sun, Q. Wang, H. Lu, T. Qin, C. Zhu, Y. Yao, S. Fan, X. Li, T. Wang, P. Liu, K. Zhu, H. Zhu, D. Shi, P. Wang, Y. Guan, X. Tang, M. Liu, Y. E. Jiang, J. Yang, J. Liu, G. Zhang, and W. Zhou. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl, 2025b. URL <https://arxiv.org/abs/2508.13167>.
- H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman,

- I. Sutskever, and K. Cobbe. Let’s verify step by step, 2023. URL <https://arxiv.org/abs/2305.20050>.
- S. Lin, J. Hilton, and O. Evans. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*, 2022.
- X. Liu, T. Chen, L. Da, C. Chen, Z. Lin, and H. Wei. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6107–6117, 2025.
- Y. Liu, G. Deng, Y. Li, K. Wang, Z. Wang, X. Wang, T. Zhang, Y. Liu, H. Wang, Y. Zheng, et al. Prompt injection attack against llm-integrated applications. *arXiv preprint arXiv:2306.05499*, 2023.
- J. M. Logg, J. A. Minson, and D. A. Moore. Algorithm appreciation: People prefer algorithmic to human judgment. *Organizational Behavior and Human Decision Processes*, 151:90–103, 2019.
- B. Lubars and C. Tan. Ask not what ai can do, but what ai should do: Towards a framework of task delegability. *Advances in neural information processing systems*, 32, 2019.
- Z. Luo, Z. Shen, W. Yang, Z. Zhao, P. Jwala-puram, A. Saha, D. Sahoo, S. Savarese, C. Xiong, and J. Li. Mcp-universe: Benchmarking large language models with real-world model context protocol servers. *arXiv preprint arXiv:2508.14704*, 2025.
- F. Luthans and T. I. Stewart. A general contingency theory of management. *Academy of management Review*, 2(2):181–195, 1977.
- S. Ma, Y. Lei, X. Wang, C. Zheng, C. Shi, M. Yin, and X. Ma. Who should i trust: Ai or myself? leveraging human and ai correctness likelihood to promote appropriate trust in ai-assisted decision-making. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2023.
- L. Malmqvist. Sycophancy in large language models: Causes and mitigations. In *Intelligent Computing-Proceedings of the Computing Conference*, pages 61–74. Springer, 2025.
- Y. Mao, M. G. Reinecke, M. Kunesch, E. A. Duéñez-Guzmán, R. Comanescu, J. Haas, and J. Z. Leibo. Doing the right thing for the right reason: Evaluating artificial moral cognition by probing cost insensitivity. *arXiv preprint arXiv:2305.18269*, 2023.
- S. Masoudnia and R. Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2):275–293, 2014.
- P. Mazdin and B. Rinner. Distributed and communication-aware coalition formation and task assignment in multi-robot systems. *IEEE Access*, 9:35088–35100, 2021.
- E. A. M. Michels, S. Gilbert, I. Koval, and M. K. Wekenborg. Alarm fatigue in healthcare: a scoping review of definitions, influencing factors, and mitigation strategies. *BMC nursing*, 24(1):664, 2025.
- Microsoft. Unleashing the power of model context protocol (mcp): A game-changer in AI integration, 2025.
- E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, and Á. Fernández-Leal. Human-in-the-loop machine learning: a state of the art. *Artificial Intelligence Review*, 56(4):3005–3054, 2023.
- C. Mueller and A. Vogelsmeier. Effective delegation: Understanding responsibility, authority, and accountability. *Journal of Nursing Regulation*, 4(3):20–27, 2013.
- R. B. Myerson. Optimal coordination mechanisms in generalized principal–agent problems. *Journal of mathematical economics*, 10(1):67–81, 1982.
- O. Nachum, S. S. Gu, H. Lee, and S. Levine. Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems*, 31, 2018.
- S. K. Nagia. Delegation of authority: A great challenge for business organisation. *ARTIFICIAL INTELLIGENCE (AI) AND BUSINESS*, page 55, 2024.

- M. Naiseh, D. Al-Thani, N. Jiang, and R. Ali. Explainable recommendation: when design meets trust calibration. *World Wide Web*, 24(5):1857–1884, 2021.
- M. Naiseh, D. Al-Thani, N. Jiang, and R. Ali. How the different explanation classes impact trust calibration: The case of clinical decision support systems. *International Journal of Human-Computer Studies*, 169:102941, 2023.
- J. Needham, G. Edkins, G. Pimpale, H. Bartsch, and M. Hobbhahn. Large language models often know when they are being evaluated. *arXiv preprint arXiv:2505.23836*, 2025.
- E. Neelou, I. Novikov, M. Moroz, O. Narayan, T. Saade, M. Ayenson, I. Kabanov, J. Ozmen, E. Lee, V. S. Narajala, E. G. Junior, K. Huang, H. Gulsin, J. Ross, M. Vyshegorodtsev, A. Travers, I. Habler, and R. Jadav. A2as: Agentic ai runtime security and self-defense, 2025. URL <https://arxiv.org/abs/2510.13825>.
- E. Nijkamp, B. Pang, H. Hayashi, L. Tu, H. Wang, Y. Zhou, S. Savarese, and C. Xiong. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*, 2022.
- Z. Ning and L. Xie. A survey on multi-agent reinforcement learning and its application. *Journal of Automation and Intelligence*, 3(2):73–91, 2024.
- O. Or-Meir, N. Nissim, Y. Elovici, and L. Rokach. Dynamic malware analysis in the modern era—a state of the art survey. *ACM Computing Surveys (CSUR)*, 52(5):1–48, 2019.
- D. Otley. The contingency theory of management accounting and control: 1980–2014. *Management accounting research*, 31:45–62, 2016.
- W. G. Ouchi and J. B. Dowling. Defining the span of control. *Administrative Science Quarterly*, pages 357–365, 1974.
- B. Paranjape, S. Lundberg, S. Singh, H. Ha-jishirzi, L. Zettlemoyer, and M. T. Ribeiro. Art: Automatic multi-step reasoning and tool-use for large language models. *arXiv preprint arXiv:2303.09014*, 2023.
- R. Parasuraman, R. Molloy, and I. L. Singh. Performance consequences of automation-induced ‘complacency’. *The International Journal of Aviation Psychology*, 3(1):1–23, 1993.
- S. Parikh and R. Surapaneni. Powering AI commerce with the new Agent Payments Protocol (AP2), Sept. 2025. URL <https://cloud.google.com/blog/products/ai-machine-learning/announcing-agents-to-payments-ap2-protocol>. Accessed: 2026-01-20.
- I. Pastine and T. Pastine. *Introducing game theory: A graphic guide*. Icon Books, 2017.
- S. Pateria, B. Subagdja, A.-h. Tan, and C. Quek. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 54(5):1–35, 2021.
- M. Petkus. Why and how zk-snark works. *arXiv preprint arXiv:1906.07221*, 2019.
- E. Pignatelli, J. Ferret, M. Geist, T. Mesnard, H. van Hasselt, O. Pietquin, and L. Toni. A survey of temporal credit assignment in deep reinforcement learning. *arXiv preprint arXiv:2312.01072*, 2023.
- I. Pinyol and J. Sabater-Mir. Computational trust and reputation models for open multi-agent systems: a review. *Artificial Intelligence Review*, 40(1):1–25, 2013.
- Z. Porter, P. Ryan, P. Morgan, J. Al-Qaddoumi, B. Twomey, J. McDermid, and I. Habli. Unravelling responsibility for ai. *arXiv preprint arXiv:2308.02608*, 2023.
- C. Qian, Z. Xie, Y. Wang, W. Liu, K. Zhu, H. Xia, Y. Dang, Z. Du, W. Chen, C. Yang, et al. Scaling large language model-based multi-agent collaboration. *arXiv preprint arXiv:2406.07155*, 2024.
- K. Qin, L. Zhou, B. Livshits, and A. Gervais. Attacking the defi ecosystem with flash loans for fun and profit, 2021. URL <https://arxiv.org/abs/2003.03810>.

- Y. Qin, S. Liang, Y. Ye, K. Zhu, L. Yan, Y. Lu, Y. Lin, X. Cong, X. Tang, B. Qian, S. Zhao, L. Hong, R. Tian, R. Xie, J. Zhou, M. Gerstein, D. Li, Z. Liu, and M. Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis, 2023. URL <https://arxiv.org/abs/2307.16789>.
- B. Radosevich and J. Halloran. Mcp safety audit: Llms with the model context protocol allow major security exploits. *arXiv preprint arXiv:2504.03767*, 2025.
- S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *The knowledge engineering review*, 19(1):1–25, 2004.
- J. Rando and F. Tramèr. Universal jailbreak backdoors from poisoned human feedback, 2024. URL <https://arxiv.org/abs/2311.14455>.
- S. Rasal and E. J. Hauer. Navigating complexity: Orchestrated problem solving with multi-agent llms, 2024. URL <https://arxiv.org/abs/2402.16713>.
- T. Rebedea, R. Dinu, M. Sreedhar, C. Parisien, and J. Cohen. Nemo guardrails: A toolkit for controllable and safe llm applications with programmable rails, 2023. URL <https://arxiv.org/abs/2310.10501>.
- M. G. Reinecke, Y. Mao, M. Kunesch, E. A. Duéñez-Guzmán, J. Haas, and J. Z. Leibo. The puzzle of evaluating moral cognition in artificial agents. *Cognitive Science*, 47(8):e13315, 2023.
- A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, J. Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.
- C. O. Retzlaff, S. Das, C. Wayllace, P. Mousavi, M. Afshari, T. Yang, A. Saranti, A. Angerschmid, M. E. Taylor, and A. Holzinger. Human-in-the-loop reinforcement learning: A survey and position on requirements, challenges, and opportunities. *Journal of Artificial Intelligence Research*, 79:359–415, 2024.
- C. Riquelme, J. Puigcerver, B. Mustafa, M. Neumann, R. Jenatton, A. Susano Pinto, D. Keysers, and N. Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021.
- J. M. Rosanas and M. Velilla. Loyalty and trust as the ethical bases of organizations. *Journal of Business Ethics*, 44(1):49–59, 2003.
- A. Rosenblat and L. Stark. Algorithmic labor and information asymmetries: A case study of uber’s drivers. *International Journal of Communication*, 10:3758–3784, 2016. URL <https://ijoc.org/index.php/ijoc/article/view/4892>.
- J. Ruan, Y. Chen, B. Zhang, Z. Xu, T. Bao, H. Mao, Z. Li, X. Zeng, R. Zhao, et al. Tptu: Task planning and tool usage of large language model-based ai agents. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.
- J. M. Sanabria and P. A. Vecino. Beyond the sum: Unlocking ai agents potential through market forces, 2025. URL <https://arxiv.org/abs/2501.10388>.
- T. Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *AAAI*, volume 93, pages 256–262, 1993.
- Y. Sannikov. A continuous-time version of the principal-agent problem. *The Review of Economic Studies*, 75(3):957–984, 2008.
- F. Santoni de Sio and G. Mecacci. Four responsibility gaps with artificial intelligence: Why they matter and how to address them. *Philosophy & technology*, 34(4):1057–1084, 2021.
- S. Sarkar, M. Curado Malta, and A. Dutta. A survey on applications of coalition formation in multi-agent systems. *Concurrency and Computation: Practice and Experience*, 34(11):e6876, 2022.
- W. Saunders, C. Yeh, J. Wu, S. Bills, L. Ouyang, J. Ward, and J. Leike. Self-critiquing models for assisting human evaluators, 2022. URL <https://arxiv.org/abs/2206.05802>.

- S. Shah. The principal-agent problem in finance. *CFA Institute Research Foundation L2014-1*, 2014.
- Y. Shao, H. Zope, Y. Jiang, J. Pei, D. Nguyen, E. Brynjolfsson, and D. Yang. Future of work with ai agents: Auditing automation and augmentation potential across the u.s. workforce, 2025. URL <https://arxiv.org/abs/2506.06576>.
- M. Sharma, M. Tong, T. Korbak, D. Duvenaud, A. Askell, S. R. Bowman, N. Cheng, E. Durmus, Z. Hatfield-Dodds, S. R. Johnston, et al. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*, 2023.
- N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.
- O. M. Shehory, K. Sycara, and S. Jha. Multi-agent coordination through coalition formation. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 143–154. Springer, 1997.
- A. Singh, A. Ehtesham, S. Kumar, and T. T. Khoei. A survey of the model context protocol (mcp): Standardizing context to enhance large language models (llms). 2025.
- J. Skalse and M. Mancosu. Defining and characterizing reward hacking. *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 1–11, 2022. doi: 10.1145/3514094.3534149.
- P. Sloksnath. Delegating moral decisions to ai systems. Master’s thesis, University of Zurich, 2025.
- S. C. Slota, K. R. Fleischmann, S. Greenberg, N. Verma, B. Cummings, L. Li, and C. Shenefiel. Many hands make many fingers to point: challenges in creating accountable ai. *Ai & Society*, 38(4):1287–1299, 2023.
- R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on computers*, 29(12):1104–1113, 1980.
- J. Sobel. Information control in the principal-agent problem. *International Economic Review*, pages 259–269, 1993.
- X. Song, Z. Wang, S. Wu, T. Shi, and L. Ai. Gradientsys: A multi-agent llm scheduler with react orchestration, 2025. URL <https://arxiv.org/abs/2507.06520>.
- C. Stucky, M. De Jong, and F. Kabo. The paradox of network inequality: differential impacts of status and influence on surgical team communication. *Med J (Ft Sam Houst Tex)*, pages 22–01, 2022.
- R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.
- S. Tadelis and O. E. Williamson. Transaction cost economics. *The handbook of organizational economics*, 159(3.1):1, 2012.
- W. Takerngsaksiri, J. Pasuksmit, P. Thongtanunam, C. Tantithamthavorn, R. Zhang, F. Jiang, J. Li, E. Cook, K. Chen, and M. Wu. Human-in-the-loop software development agents. In *2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 342–352. IEEE, 2025.
- J. Teutsch and C. Reitwiesner. Truebit: a scalable verification solution for blockchains. *White Papers*, 2018.
- J. Teutsch and C. Reitwiesner. A scalable verification solution for blockchains. In *Aspects of Computation and Automata Theory with Applications*, pages 377–424. World Scientific, 2024.
- N. A. Theobald and S. Nicholson-Crotty. The many faces of span of control: Organizational structure across multiple goals. *Administration & Society*, 36(6):648–660, 2005.
- N. Tomašev, M. Franklin, J. Jacobs, S. Krier, and S. Osindero. Distributional agi safety. *arXiv preprint arXiv:2512.16856*, 2025.

- N. Tomasev, M. Franklin, J. Z. Leibo, J. Jacobs, W. A. Cunningham, I. Gabriel, and S. Osindero. Virtual agent economies, 2025. URL <https://arxiv.org/abs/2509.10147>.
- P. M. Tomei, R. Jain, and M. Franklin. Ai governance through markets. *arXiv preprint arXiv:2501.17755*, 2025.
- K.-T. Tran, D. Dao, M.-D. Nguyen, Q.-V. Pham, B. O’Sullivan, and H. D. Nguyen. Multi-agent collaboration mechanisms: A survey of llms, 2025. URL <https://arxiv.org/abs/2501.06322>.
- V. Tupe and S. Thube. Ai agentic workflows and enterprise apis: Adapting api architectures for the age of ai agents, 2025. URL <https://arxiv.org/abs/2502.17443>.
- M. Turpin, J. Michael, E. Perez, and S. R. Bowman. Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting, 2023. URL <https://arxiv.org/abs/2305.04388>.
- R. Uuk, C. I. Gutierrez, D. Guppy, L. Lauwaert, A. Kasirzadeh, L. Velasco, P. Slattery, and C. Prunkl. A taxonomy of systemic risks from general-purpose ai. *arXiv preprint arXiv:2412.07780*, 2024.
- K. Valmeekam, M. Marquez, S. Sreedharan, and S. Kambhampati. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems*, 36:75993–76005, 2023.
- A. H. Van de Ven. The concept of fit in contingency theory. Technical report, 1984.
- T. van der Weij, F. Hofstätter, O. Jaffe, S. F. Brown, and F. R. Ward. Ai sandbagging: Language models can strategically underperform on evaluations. *arXiv preprint arXiv:2406.07358*, 2025. Published as a conference paper at ICLR 2025.
- A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *International conference on machine learning*, pages 3540–3549. PMLR, 2017a.
- A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu. Feudal networks for hierarchical reinforcement learning, 2017b. URL <https://arxiv.org/abs/1703.01161>.
- E. F. Vignola, S. Baron, E. Abreu Plasencia, M. Hussein, and N. Cohen. Workers’ health under algorithmic management: Emerging findings and urgent research questions. *International Journal of Environmental Research and Public Health*, 20(2):1239, 2023. doi: 10.3390/ijerph20021239.
- J. Vokřínek, J. Bíba, J. Hodík, J. Vybíhal, and M. Pěchouček. Competitive contract net protocol. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 656–668. Springer, 2007.
- G. Wang, B. Wang, T. Wang, A. Nika, H. Zheng, and B. Y. Zhao. Ghost riders: Sybil attacks on crowdsourced mobile mapping services. *IEEE/ACM Transactions on Networking*, 26(3): 1123–1136, 2018. doi: 10.1109/TNET.2018.2818073.
- J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*, 2020.
- J. Wang, Z. Wu, Y. Li, H. Jiang, P. Shu, E. Shi, H. Hu, C. Ma, Y. Liu, X. Wang, Y. Yao, X. Liu, H. Zhao, Z. Liu, H. Dai, L. Zhao, B. Ge, X. Li, T. Liu, and S. Zhang. Large language models for robotics: Opportunities, challenges, and perspectives, 2024a. URL <https://arxiv.org/abs/2401.04334>.
- L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024b.
- Y. Wang, D. Xue, S. Zhang, and S. Qian. Badagent: Inserting and activating backdoor attacks in llm agents, 2024c. URL <https://arxiv.org/abs/2406.03007>.
- A. Wei, N. Haghtalab, and J. Steinhardt. Jailbroken: How does llm safety training fail? Ad-

- vances in Neural Information Processing Systems, 36:80079–80110, 2023.
- O. E. Williamson. Transaction-cost economics: the governance of contractual relations. *The journal of Law and Economics*, 22(2):233–261, 1979.
- O. E. Williamson. Transaction cost economics. *Handbook of industrial organization*, 1:135–182, 1989.
- M. Wischnewski, N. Krämer, and E. Müller. Measuring and understanding trust calibrations for automated systems: A survey of the state-of-the-art and future directions. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pages 1–16, 2023.
- Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou, et al. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101, 2025.
- Y. Xiao, P. P. Liang, U. Bhatt, W. Neiswanger, R. Salakhutdinov, and L.-P. Morency. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. *arXiv preprint arXiv:2210.04714*, 2022.
- W. Xing, Z. Qi, Y. Qin, Y. Li, C. Chang, J. Yu, C. Lin, Z. Xie, and M. Han. Mcp-guard: A defense framework for model context protocol integrity in large language model applications. *arXiv preprint arXiv:2508.10991*, 2025.
- F. Xu, Q. Hao, C. Shao, Z. Zong, Y. Li, J. Wang, Y. Zhang, J. Wang, X. Lan, J. Gong, et al. Toward large reasoning models: A survey of reinforced reasoning with large language models. *Patterns*, 6(10), 2025.
- L. Xu and H. Weigand. The evolution of the contract net protocol. In *International Conference on Web-Age Information Management*, pages 257–264. Springer, 2001.
- Y. Yang, Y. Wen, J. Wang, and W. Zhang. Agent exchange: Shaping the future of ai agent economics. *arXiv preprint arXiv:2507.03904*, 2025.
- J. Yi, Y. Xie, B. Zhu, E. Kiciman, G. Sun, X. Xie, and F. Wu. Benchmarking and defending against indirect prompt injection attacks on large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, page 1809–1820. ACM, July 2025. doi: 10.1145/3690624.3709179. URL <http://dx.doi.org/10.1145/3690624.3709179>.
- H. Yu, Z. Shen, C. Leung, C. Miao, and V. R. Lesser. A survey of multi-agent trust management systems. *IEEE Access*, 1:35–50, 2013.
- L. Yu, V. Do, K. Hambardzumyan, and N. Cancorra. Robust llm safeguarding via refusal feature adversarial training. *arXiv preprint arXiv:2409.20089*, 2024.
- M. Yu, F. Meng, X. Zhou, S. Wang, J. Mao, L. Pang, T. Chen, K. Wang, X. Li, Y. Zhang, B. An, and Q. Wen. A survey on trustworthy llm agents: Threats and countermeasures, 2025. URL <https://arxiv.org/abs/2503.09648>.
- Y. Yuan, W. Jiao, W. Wang, J.-t. Huang, J. Xu, T. Liang, P. He, and Z. Tu. Refuse whenever you feel unsafe: Improving safety in llms via decoupled refusal training. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3149–3167, 2025.
- S. E. Yuksel, J. N. Wilson, and P. D. Gader. Twenty years of mixture of experts. *IEEE transactions on neural networks and learning systems*, 23(8):1177–1193, 2012.
- F. M. Zanzotto. Human-in-the-loop artificial intelligence. *Journal of Artificial Intelligence Research*, 64:243–252, 2019.
- Q. Zhan, Z. Liang, Z. Ying, and D. Kang. Injegagent: Benchmarking indirect prompt injections in tool-integrated large language model agents, 2024. URL <https://arxiv.org/abs/2403.02691>.
- N. Zhang, J. Yan, C. Hu, Q. Sun, L. Yang, D. W. Gao, J. M. Guerrero, and Y. Li. Price-matching-based regional energy market with hierarchical

- reinforcement learning algorithm. *IEEE Transactions on Industrial Informatics*, 20(9):11103–11114, 2024.
- W. Zhang, C. Cui, Y. Zhao, R. Hu, Y. Liu, Y. Zhou, and B. An. Agentorchestra: A hierarchical multi-agent framework for general-purpose task solving. *arXiv e-prints*, pages arXiv–2506, 2025a.
- Z. Zhang, Q. Dai, X. Bo, C. Ma, R. Li, X. Chen, J. Zhu, Z. Dong, and J.-R. Wen. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):1–47, 2025b.
- K. Zhao, L. Li, K. Ding, N. Z. Gong, Y. Zhao, and Y. Dong. A survey on model extraction attacks and defenses for large language models, 2025. URL <https://arxiv.org/abs/2506.22521>.
- W. Zhao, Y. Gao, S. A. Memon, B. Raj, and R. Singh. Hierarchical routing mixture of experts. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7900–7906. IEEE, 2021.
- L. Zhou, X. Xiong, J. Ernstberger, S. Chaliasos, Z. Wang, Y. Wang, K. Qin, R. Wattenhofer, D. Song, and A. Gervais. Sok: Decentralized finance (defi) attacks, 2023. URL <https://arxiv.org/abs/2208.13035>.
- Y. Zhou, T. Lei, H. Liu, N. Du, Y. Huang, V. Zhao, A. M. Dai, Q. V. Le, J. Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.
- C. Zhu, M. Dastani, and S. Wang. A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*, 38(1):4, 2024.
- Z. Zou, Z. Liu, L. Zhao, and Q. Zhan. Blocka2a: Towards secure and verifiable agent-to-agent interoperability. *arXiv preprint arXiv:2508.01332*, 2025.