Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version 11 January, 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.011100

A Tutorial on Cognitive Biases in Agentic AI-Driven 6G Autonomous Networks

HATIM CHERGUI[1] *(Senior Member, IEEE)*, FARHAD REZAZADEH [2] *(Member, IEEE)*, MEROUANE DEBBAH [3] *(Fellow, IEEE)*, AND CHRISTOS VERIKOUKIS [4] *(Senior Member, IEEE)*

[1]i2CAT Foundation, Barcelona, Spain
[2]Hostelworld Group and the Technical University of Catalonia (UPC), 08860 Castelldefels
[3]Research Institute for Digital Future, Khalifa University, 127788 Abu Dhabi, UAE
[4]ISI/ATH and University of Patras, Greece

CORRESPONDING AUTHOR: Hatim Chergui (e-mail: chergui@ieee.org).

ABSTRACT The path to higher network autonomy in 6G lies beyond the mere optimization of key performance indicators (KPIs), requiring systems that perceive and reason over the network environment *as it is*. This can be achieved through *agentic AI*, where large language model (LLM)-powered agents utilize multimodal telemetry, memory, and cross-domain negotiation to achieve multi-objective goals. However, deploying such agents introduces cognitive biases inherited from human design, which can severely distort reasoning and actuation. This paper provides a comprehensive tutorial on well-known cognitive biases, detailing their taxonomy, mathematical formulation, emergence in telecom systems, and tailored mitigation strategies. We validate these concepts through two distinct use-cases in 6G management[1]. First, we tackle *anchoring bias* in inter-slice resource negotiation. To overcome the prohibitive execution delays of cloud-based LLMs, this use-case deploys a locally hosted 1B-parameter model on an RTX A4000 GPU, successfully achieving sub-second inference latencies compatible with near-real-time (near-RT) operations. By replacing fixed heuristic anchors with a Truncated Weibull randomized anchor strategy, the agents dismantle rigid biases, intelligently consume SLA slack, and dynamically double the system-wide energy savings (peaking at 25%) without violating strict latency limits. Second, we mitigate *temporal and confirmation biases* in RAN-Edge cross-domain negotiation by designing an unbiased collective memory. By integrating semantic/temporal decay and an inflection bonus that actively highlights past negotiation failures, agents are prevented from over-relying on recent data or repeating past mistakes. Grounding decisions in this richer, debiased historical context yields highly robust agreements, achieving a $\times 5$ latency reduction and roughly 40% higher energy savings compared to memoryless baselines.

INDEX TERMS 6G, agentic AI, bias, network automation.

I. Introduction

THE landscape of telecom automation over the past 15 years can be structured along TM Forum's maturity levels, from Level 1 (script-based) to Level 3 (analytics- and AI-assisted orchestration) [1], [2], with initiatives such as 3GPP Self-Organizing Networks (SON), ETSI Zero-Touch Service Management (ZSM) and O-RAN. While their introduced paradigms have yielded operational efficiency, their optimization logic remains narrowly tethered to KPIs. However, those metrics are imperfect proxies/surrogates for the underlying objectives of communication systems. This reliance induces a well-documented failure mode captured by *Goodhart's Law*, which states that "when a measure becomes a target, it ceases to be a good measure" [3]. In practice, this manifests as *metric fixation* [4], where algorithmic optimization maximizes numerical scores while neglecting harder-to-measure essence of the telecom environment and its users. In anticipation of 6G, overcoming this proxy-goal misalignment necessitates a transition from metric-centric optimization to *goal-directed autonomy*. This implies not only moving towards TM Forum Levels 4 and 5, but also embedding intelligence that can operate across dynamic, multi-domain, and partially observable environments. *Agentic AI* provides a viable paradigm: LLM-powered autonomous agents endowed with the ability to directly perceive through multimodal data (without proxies), reason over historical and contextual knowledge, plan adaptive courses of action, communicate under diverse intents, and act through APIs or network control functions. Crucially, such agents can negotiate across domains (e.g., RAN, edge, and core) to reconcile competing objectives, adapt to unforeseen conditions, and iteratively refine strategies via explanation and feedback. In this way, agentic AI would serve as the cognitive substrate for 6G networks, bridging the gap between low-level signals and high-level semantic goals. Rather than optimizing proxies, networks can pursue objectives that more faithfully represent the essence of telecommunication.

A. Cognitive Biases in Agentic Systems

Special attention should be paid, however, to the susceptibility of agentic systems to the so-called *cognitive biases*, which are *systematic deviations from rational judgment;* a concept originally rooted in human psychology that agents could inherit in various ways and perpetuate, posing a tangible threat to the efficiency, fairness, and reliability of next-generation autonomous networks. Biases affect every layer of an agent's behavior, from initial perception and reasoning to final decision-making and action execution. The foundational work on cognitive biases in human judgment was established by Tversky and Kahneman in their seminal paper, "Judgment under Uncertainty: Heuristics and Biases" [5]. Their research demonstrated that humans rely on cognitive shortcuts, or heuristics, which can lead to predictable and systematic errors in judgment. In agentic systems, these biases are inherited and manifested across multiple layers. The insidious impact of biases can be observed throughout the entire agentic system pipeline, from perception to action. i) *Prompts* are a primary entry point for bias, where framing effects or the specific wording of instructions can skew an agent's perception of a problem. For instance, prompting a network agent to "maximize throughput at all costs" can lead to a solution that ignores critical latency or fairness metrics. ii) Biases are deeply embedded in the *training data,* stemming from historical imbalances, cultural skews, and sampling errors. A large language model trained on data from a region with a specific network infrastructure could, when deployed elsewhere, exhibit a bias towards that legacy architecture, failing to optimally manage a new, more advanced one. Furthermore, biases emerge and compound in an agent's internal processes, with implications for its reasoning and tool use. iii) The agent's *reasoning* paths are susceptible to various heuristics. An agent tasked with dynamic resource allocation could exhibit an availability heuristic, disproportionately allocating bandwidth to a specific network slice from which it has received the most recent or frequent requests, thereby neglecting other slices in need. Similarly, a security agent may fall prey to a confirmation bias, only seeking out evidence that confirms a pre-existing threat model and overlooking a novel, unseen attack vector. These behaviors are akin to human shortcuts, where evidence is mis-weighted and readily accessible information is prioritized. iv) Biases are also evident in *tool use,* which includes the way memories are stored and retrieved, and how data sources and APIs are selected. An agent's memory retrieval might be subject to recency/primacy biases, causing it to favor recently processed network logs over a more complete historical record, leading to short-sighted decisions. Conversely, an authority bias could lead an agent to show a strong, uncritical preference for data from a single, *authoritative* source or a familiar API, even when more suitable or diverse tools are available. The recent work by Xie et al. [6] highlights the growing concern of these biases, exploring their manifestation in large language models within multi-agent systems, a structure increasingly relevant to decentralized 6G architectures.

Note that agentic biases generally originate from common mechanisms across learning-based systems, including skewed training data, imperfect objectives, model inductive biases, and partial or noisy observations. In this sense, the emergence of biases in various other areas is not fundamentally different from that in LLM agents envisioned for 6G networks. The main distinction lies in how these biases manifest: in other areas, such as healthcare or finances, they typically affect semantic reasoning and language outputs, whereas in 6G agents they appear as systematic errors in control or resource allocation policies derived from biased measurements or traffic data. Therefore, the underlying mechanisms are largely shared across domains, while the domain-specific state, action, and evaluation spaces determine how biases are expressed and detected.

B. Related Work

Drawing parallels with well-documented human biases such as confirmation bias, recency effects, and groupthink, recent research examines how similar distortions can manifest both within individual AI agents and across their interactions, potentially affecting collective decision-making, fairness, and safety. A comprehensive contribution in this domain is *MindScope* [7], which introduces a dataset encompassing 72 human cognitive biases and leverages it to evaluate how LLM agents display these distortions during multi-agent dialogue. Findings indicate that even when models appear well-aligned in isolation, their interactions can trigger latent, higher-order biases such as anchoring effects. Mitigation strategies—including retrieval-augmented generation (RAG), structured debate mechanisms, and reinforcement learning (RL)-based adjudication—have shown promise in reducing these risks.

Expanding on this, Liu et al. [8] investigate the emergence of conversational echo chambers in multi-agent settings. Their work demonstrates that iterative discussions often amplify existing biases, as agents gradually converge toward consensus-seeking positions, which is an emergent phenomenon not observed in single-agent contexts. This underscores that distortions can originate from interaction dynamics themselves, rather than solely from pretrained knowledge.

In a more structural way, the *fairness in agentic AI* framework [9] examines how systemic distortions may arise through decentralized collaboration among agents. The framework connects ethical alignment with incentive structures and negotiation constraints, offering design tools to mitigate emergent bias and to prevent inequitable treatment of users or tasks embedded in implicit reward mechanisms. Complementarily, the *hidden profile benchmark* [10] adapts a classic social psychology paradigm to multi-agent LLMs, revealing that agents often fail to uncover critical but unevenly distributed information held by peers-mirroring groupthink and informational bias in human teams. This highlights fundamental vulnerabilities in reasoning diversity and emphasizes the need for diversity-aware memory structures and communication protocols.

Having said that, a tutorial introduction to cognitive biases in the context of agentic AI-based 6G network automation is yet to be considered, examining their emergence, affected agentic components, practical 6G examples and mitigation strategies.

C. Contributions

This paper makes the following contributions:

- Presents an overview of agentic systems for telecom, with their components, protocols and interactions as well as reasoning and planning capabilities while also briefly reviewing literature in this emerging area.
- Develops a systematic tutorial on cognitive biases and their emergence, including taxonomy, impact on the various agentic components such as reasoning, planning, memory, negotiation, tool use, and actuation, as well as provides practical 6G examples and mitigation approaches.
- Demonstrates the emergence and implications of these biases in 6G management scenarios, with a focus on RAN-Edge cross-domain orchestration and service level agreement (SLA)-sensitive optimization.
- Illustrates the mitigation gain of some well-known biases through two use-cases on 6G inter-slice and cross-domain agentic negotiation. Specifically, debiasing mechanisms targeting negotiation as well as memory use are explored to encourage bolder decisions and avoid anchoring, temporal and confirmation biases, wherefore the resulting gains are assessed in terms of latency and energy saving.

II. Agentic AI-Driven Network Automation

A. Typical Components of an Agentic System

A modern telecom agentic system is a tightly integrated, multi-layered architecture whose components collectively enable perception, semantic reasoning, secure negotiation, safe actuation and continual learning as depicted in Figure 1. The following subsections describe the canonical elements, their responsibilities, and the integration patterns that turn a collection of tools into a coherent agentic fabric suitable for various technological domains.

1) LLM-empowered Agent

The core agent is an LLM-centered [11], [12] reasoning and control entity that orchestrates the perception-reason-plan-act loop. Beyond natural-language fluency, it implements modular capabilities: a perception front-end that ingests latent representations, a retrieval-augmented memory interface, a symbolic/planning module that generates candidate strategies, a negotiator that composes intentful messages for peers, a verifier that queries a digital twin for validation, and an executor that issues guarded network API calls. Agents must support stateful dialogues, multi-turn negotiation, and constrained action synthesis (e.g., produce candidate plans with cost/risk annotations). Practically, the agent exposes two main kinds of interfaces: (i) tool invocation via an MCP-like function call semantics (e.g., `mcp.call("digital_twin.simulate", scenario=..., horizon=30s)`), and (ii) agent-to-agent messages using an A2A protocol carrying structured intents (proposal, counter-proposal, justification, commit).

2) Digital Twin and Prediction Tools

A high-fidelity digital twin (DT) functions as the agent's safe playground [13]: agents submit what-if scenarios, stress tests, or short rollouts to evaluate the temporal and cross-domain impact of candidate actions before committing them to live infrastructure. Twins range from lightweight surrogate models that provide millisecond feedback to detailed simulators for deep validation. Key responsibilities include maintaining synchronized state with production (shadowing), supporting rollback/canary experiments, exposing cost models and failure modes, and providing counterfactual traces that agents use to attach confidence and expected utility to proposals. Prediction Services [14] also allow agents to perform farsighted decisions.

3) Memory and Experience Store

Memory enables sample-efficient reasoning and faster consensus by allowing agents to recall past strategies, observed outcomes, and negotiated compromises [15]. Architecturally, memory comprises multiple layers: short-term (session-level) buffers for ongoing negotiations, episodic records of full negotiation-execution episodes, and a semantic/latent index (vector store) for rapid similarity search. Basic retrieval policies prioritize recent, high-reward episodes and semantically matching contexts. Memory interfaces must support semantic queries (e.g., embedding and approximate nearest neighbor (ANN) lookup), provenance metadata, and secure

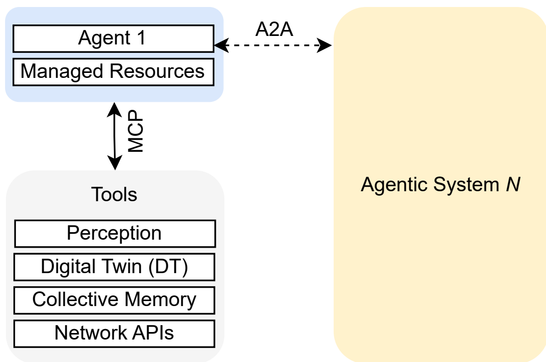


Figure 1: Typical components of a 6G agentic system.

access controls so agents can justify decisions with human-interpretable precedents.

4) Data Sources and Perception Stack

Robust perception aggregates heterogeneous telemetry, e.g., time-series KPIs, protocol traces, topology and inventory, user-plane metrics, and multi-modal signals (e.g., maps, logs). A preprocessing pipeline performs schema harmonization, normalization, feature extraction and embedding into latent spaces suited for LLM agents and downstream planners. Event detection, anomaly scoring, and context windows feed the agent's situational awareness. Critically, the perception stack must provide uncertainty estimates and data-quality indicators so reasoning modules can weigh observations correctly.

5) Network APIs and Actuation

Actuation in agentic AI for telecom networks is performed through domain-specific APIs that expose fine-grained control over the RAN, edge services, core functions, and orchestration platforms. For instance, an srsRAN API may allow adjusting scheduling parameters or PRB allocations, while a Kubernetes (K8s) API can scale edge services or redeploy VNFs.

To better understand the capacity and reliability of underlying platforms such as Kubernetes, OpenStack, and Docker, agents must consider several key technical dimensions, including compute and memory resource availability, network throughput and latency, container or VM lifecycle management, orchestration and scheduling behavior, and fault tolerance mechanisms such as replication and automated recovery. These dimensions determine how reliably services can scale, migrate, or recover from failures during automated actuation.

Safety mechanisms are therefore critical. A proposed configuration change (e.g., a new QoS profile on a 5QI bearer or scaling of an edge service in K8s) should first be validated in a digital twin or staged on a subset of pods before system-wide rollout. Agents should also avoid performing irreversible operations without a validation step, where human approval may be required for actions such as core network function restarts. Finally, audit logs, execution receipts, and telemetry feedback must be systematically captured so that the agent's memory can be updated with ground-truth outcomes (e.g., whether a predicted CPU scaling action in K8s actually reduced latency), closing the loop between reasoning, action, and verified system state.

6) Communication Protocols, Ontologies and Agent Fabric

Inter-agent coordination rests on well-defined communication protocols [16] and shared ontologies. An agent-to-agent (A2A) protocol carries structured messages with typed fields (e.g., intent, proposals, reason). Several intents exists, e.g., *propose, counter-propose, query, explain, confirm, reject, commit.* In practice, the so-called *agent fabric* provides the runtime substrate where multiple agents are deployed, coordinated, and scaled across distributed environments. It offers common services such as discovery, messaging, state sharing, and lifecycle management, enabling agents to communicate and operate collectively rather than in isolation. The Model Context Protocol (MCP) or equivalent tool-invocation channels, on the other hand, permit standardized tool calls and return structured outputs. Semantic encodings (ontologies, schema registries, and vocabularies) are essential so proposals and justifications are machine-interpretable across vendors and domains. The TM Forum Intent Ontology (TIO), for instance, provides a standardized semantic layer for expressing intents in network and service management, ensuring that agent messages carry domain-aware meaning. By aligning inter-agent proposals and actions with TIO, heterogeneous agents can interpret, decompose, and fulfill intents consistently across RAN, edge, and core domains. This ontology-driven alignment enhances interoperability and automation by grounding agent reasoning in a common industry-backed vocabulary.

B. Agentic Reasoning and Planning in 6G

We focus on LLM-powered agents that reason over multi-modal telecom context and synthesize safe, verifiable action plans spanning RAN, edge, and core. Reasoning treats the network as a belief-driven stochastic environment, while planning composes constrained, tool-grounded action sequences that must be validated in a digital twin (DT) before actuation. To stay consistent with Fig. 1 and the bias taxonomy in Table 1, we first formalize the reasoning objective, then connect it to how models are trained to reason, and finally describe specialization to 6G data and interfaces.

Definition:(Reasoning in Agentic 6G) Given observations x (telemetry, logs, intents), background knowledge $\mathcal{K}$

(standards, policies, ontologies), and tools $\mathcal{T}$ (DT, telemetry services, network APIs), an agent seeks a plan $\pi = (a_1, \ldots, a_H)$ from a *safe action set* $\mathcal{A}_{\text{safe}}$ that maximizes a multi-objective utility $U(\pi) \in \mathbb{R}^m$ (e.g., latency, energy, fairness, risk), under operational and chance constraints:

$$\max_{\pi \subset \mathcal{A}_{\text{safe}}} \mathbb{E}[U(\pi) \mid x, \mathcal{K}] \quad \text{s.t.} \quad \Pr\{g_j(\pi, x) \leq 0\} \geq 1 - \epsilon_j, \ \forall j.$$

1) Logical Reasoning Types in 6G Agents

Logical reasoning manifests in several complementary forms. *Deductive reasoning* maps high-level intents and policies to concrete, correctness-preserving configurations (e.g., from an intent to a 5QI profile and scheduler settings), providing compliance and safety guarantees. *Inductive reasoning* [17] generalizes from historical episodes or DT rollouts to propose candidate actions with calibrated confidence, for example, inferring traffic–latency relationships that inform proactive scaling. *Abductive or causal reasoning* explains observed degradations by positing plausible causes and minimal interventions (e.g., distinguishing backhaul congestion from CPU throttling as the source of latency spikes) and selecting actions with the highest expected counterfactual gain. *Analogical reasoning* recognizes structural similarity between a novel context and past episodes—such as slice mix, RF conditions, or policy constraints—to adapt or compose previously successful strategies under current constraints. Finally, *counterfactual/model-based reasoning* [18] uses the DT to explore "what–if" scenarios and compare interventions before committing changes to live infrastructure. In practice, agents combine these modes within a single loop: hypotheses are generated abductively, stress-tested counterfactually, and distilled into deductively checkable configurations that can be generalized inductively or by analogy across sites and time.

2) From Pretraining to Operational Reasoning

Modern agents learn these behaviors in stages. During broad pretraining on text, code, and quantitative material (e.g., GSM8K maths dataset), LLMs acquire competence in symbolic manipulation and intermediate-step fidelity, which supports the derivation and verification of multi-step rationales [19]. Instruction-style fine-tuning then aligns the model to follow domain procedures, call tools when needed, and expose intermediate steps transparently, rather than relying on brittle internal shortcuts. In operation, self-consistency sampling and external verification further stabilize reasoning. The agent generates diverse rationales, selects coherent ones, and validates critical calculations and predictions using the DT or specialized analyzers. This separation between "thinking" and "checking" reduces single-path failures and yields uncertainty estimates that planning can consume.

3) Telecom Specialization and Domain Grounding

Specializing for 6G requires grounding the agent in domain-structured data and APIs. Inputs and outputs are normalized using a canonical KPI schema and shared ontologies, ensuring quantities, units, and timestamps are comparable across domains; this reduces framing effects and prevents spurious differences caused by formatting. Supervised episodes pair observations with concise rationales and concrete actions (telemetry $\rightarrow$ explanation $\rightarrow$ API calls), curated from traces and DT rollouts. Crucially, both successes and failures are included to avoid survivorship bias and to teach the boundaries where strategies break down. Preference-style optimization ranks plans by multi-objective utility and safety violations, encouraging Pareto-efficient behavior and penalizing hallucinated tool use. Finally, uncertainty-aware generation produces plans with explicit risk bounds, which are stress-tested under traffic and RF perturbations in the DT before any limited-scope canary actuation.

4) Constrained Plan–Verify–Act Loop

The operational loop follows *reason–plan–verify–act*. Reasoning fuses live telemetry, policies, and retrieved episodes to hypothesize causes and propose interventions, emitting intermediate steps with calibrated uncertainty. Planning then solves a constrained rollout over $\mathcal{A}_{\text{safe}}$; when constraints dominate (e.g., strict URLLC guarantees), lexicographic priorities ensure safety before efficiency. Verification executes counterfactuals in the DT to compute cost vectors and feasibility; plans with low confidence or high regret are rejected or revised. Actuation performs guarded API calls with staging or canaries, and outcomes are written back to episodic memory for continual improvement.

5) Bias-aware Hooks During Reasoning and Planning

Because reasoning and planning are susceptible to systematic distortions, we insert lightweight, auditable hooks that align with Table 1. Counterfactual checks in the DT encourage consideration of disconfirming alternatives, thereby mitigating confirmation, availability, and anchoring effects. Diversity-aware retrieval balances recent and older evidence, successes and failures, and semantic variety to curb recency/primacy and survivorship. Canonical KPI representations and neutralized prompts reduce framing and suggestion biases by standardizing inputs before inference. Uncertainty calibration propagates confidence throughout the pipeline to counter neglect of probability and automation bias. Finally, source-oblivious tool comparison hides vendor or authority labels during selection and demands dual-source consistency before commit, mitigating authority and halo effects.

6) Worked Formulation

Let $a \in \mathcal{A}_{\text{safe}}$ denote a joint RAN–edge action (e.g., PRB split and CPU frequency). The DT returns a cost vector $\mathbf{c}(a) = [L(a), E(a), F(a), R(a)]$ for latency, energy, fairness, and risk. A typical objective is

$$\begin{aligned} \min_{a \in \mathcal{A}_{\text{safe}}} \quad & \mathbf{w}^\top \mathbf{c}(a) \\ \text{s.t.} \quad & \Pr\{L(a) \le L_{\text{SLA}}\} \ge 1 - \epsilon, \\ & R(a) \le R_{\max}, \end{aligned} \tag{1}$$

with candidates ranked by expected regret against a baseline and by feasibility across sampled traffic/RF scenarios. Non-dominated plans are surfaced for negotiation or executed via canary rollout, with post-actuation feedback to close the loop.

C. Agentic Systems in 6G: A Review

Recent studies have explored the use of agentic AI for autonomous network management, each advancing the field from different perspectives. In [15], the authors consider a RAN-Edge cross domain management scenario where they introduce a new architecture of the long-term agentic memory to efficiently retrieve relevant stored collective agreements of the RAN and Edge LLM-powered agents and capitalize on them, resulting in lower negotiation failures and SLA violation as well as improved latency and energy saving. Ferrag et al. [20] present a comprehensive survey of agentic AI, covering approximately 60 benchmarks across reasoning, planning, tool use, multimodal and embodied tasks, interactive workflows, and agent orchestration, while also proposing a taxonomy of LLM-agent frameworks. Their work further reviews emerging agent-to-agent communication protocols such as the Agent Communication Protocol, Model Context Protocol, and A2A, and highlights key limitations, including failure modes, security risks, and open challenges. In parallel, Chen et al. [21] introduce BlockAgents, which integrate blockchain consensus and proof-of-thought mechanisms into LLM-based multi-agent coordination to ensure Byzantine-robust role assignment, proposal evaluation, and decision-making. Taken together, these studies reflect a broader convergence in the literature around three main directions: the development of modular agent architectures that integrate reasoning, memory, and tool use; the design of collaborative protocols for negotiation, consensus, and hierarchical control; and the incorporation of robustness and trust mechanisms, such as decentralized consensus and auditing, to safeguard against adversarial or faulty agents. On the other hand, [22] introduce the concept of wireless multi-agent generative AI, positioning LLM-based agents as enablers of a transition from connected to collective intelligence in wireless networks. They analyze decentralized cooperation and competition through a game-theoretic lens, propose architectural principles for deploying on-device agents, and illustrate their vision with intent-based networking as a case study. Besides, [23] focuses on integrating LLM agents into 6G networks via a modular framework of perception, grounding, and alignment, distributed across devices and edge servers. They propose a split learning paradigm and model caching strategies to optimize resource allocation and coordination, enabling applications such as integrated sensing, digital twins, and task-oriented communications. The LINKs framework introduced in [13] constitutes one of the first attempts to integrate LLMs with digital twin-based 6G management systems, enabling agents to translate high-level intent into control actions. However, its reasoning capacity is still limited, and integration with real-time RAN telemetry is underexplored. Complementarily, [24] presents a foundational framework that conceptualizes generative AI models as distributed agents for collaborative task planning across network elements, emphasizing emergent communication and semantic coordination, though without concrete ties to telecom protocols or operational data. A more modular approach is advanced in [25], which separates cloud-based intent parsing from edge-level policy generation to address scalability and latency, though validation is restricted to high-level simulations rather than real network traces. Extending beyond telecom, [26] investigates agentic digital twins for cyber-physical optimization using large generative models and MCP, demonstrating orchestration capabilities primarily in urban logistics scenarios. To enhance grounding and mitigate hallucinations, [27] proposes a retrieval-augmented architecture that incorporates structured network knowledge into the agent reasoning loop, though this remains at the conceptual stage. Finally, [28] outlines an end-to-end design for embedding LLM agents into the O-RAN framework to automate lifecycle management of xApps and rApps, aligning well with modular RIC interfaces but still lacking empirical validation and deployment studies.

D. O-RAN Components, Interfaces, and Feasibility of Bias-Mitigation Integration

Figure 2 summarizes the Open Radio Access Network (O-RAN) control architecture that we reference throughout when discussing the Radio Intelligent Controller (RIC), xApps, and the A1/E2/O1 interfaces. At the Radio Access Network (RAN) layer, O-RAN disaggregates the base station into the O-RAN Radio Unit (O-RU), O-RAN Distributed Unit (O-DU), and O-RAN Central Unit (O-CU), which align with the Third Generation Partnership Project (3GPP) New Radio (NR) / Next Generation RAN (NG-RAN) functional split; radio-frequency and low-PHY processing at the O-RU, time-critical high-PHY/MAC/RLC functions at the O-DU, and higher-layer functions at the O-CU [29], [30]. Above the RAN, O-RAN introduces two complementary control nodes. The Near-Real-Time Radio Intelligent Controller (Near-RT RIC) hosts near-real-time control applications (xApps) for closed-loop optimization (e.g., handover, interference, load balancing, anomaly detection) and interacts with the RAN via the E2 interface (E2), enabling telemetry subscription and bounded control actions under tight latency constraints [31], [32]. The Non-Real-Time Radio Intelligent Controller (Non-RT RIC) resides within the Service Management and Orchestration (SMO) layer, alongside Operations Support Systems / Business Support Systems (OSS/BSS), and op-

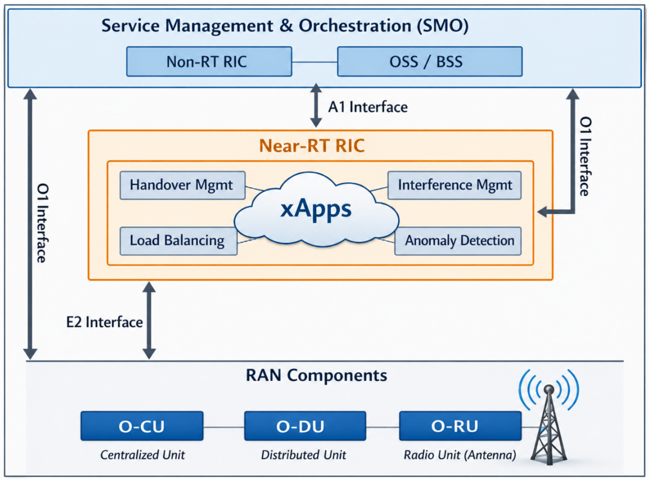


Figure 2: Reminder of a typical O-RAN architecture.

erates at longer timescales for policy, analytics, model governance, and lifecycle management; it guides the Near-RT RIC through the A1 interface (A1) [32], [33]. Orthogonally, the O1 interface (O1) supports operations, administration, and maintenance functions (telemetry, configuration, alarms, audit), providing the long-horizon evidence channel needed for validation and accountability [32]. This architectural separation also provides a concrete path to integrate the bias-mitigation strategies discussed in this paper into O-RAN without requiring non-standard interfaces. First, mitigation must be *timescale-aligned*; methods that require broad context or multi-episode reasoning—e.g., adversarial evaluation loops, counterfactual stress testing with a digital twin, dataset balancing across successes/failures to reduce survivorship effects, or prompt/representation canonicalization to reduce framing and suggestion bias—fit naturally in SMO/Non-RT RIC because they can operate over minutes-to-hours horizons and can be expressed as policies or model governance delivered downstream via A1 [32], [33]. Second, mitigation must be *interface-compliant*; evidence needed to challenge hypotheses (confirmation bias) or calibrate uncertainty (neglect of probability) can be obtained via E2 telemetry and complemented by O1/OAM traces for longer-window ground truth; conversely, near-real-time mitigation hooks (e.g., mandatory verification gates for high-blast-radius actions to curb automation bias, confidence thresholds and risk buffers to curb uncertainty neglect, or cross-source consistency checks to curb authority/halo effects) can be implemented inside xApps while still using the standard E2 control/indication flows [31], [32]. Third, mitigation must be *safely deployable*; O-RAN already encourages staged control through policy constraints and OAM oversight; thus a practical plan–verify–act workflow can be realized by having Non-RT RIC/SMO enforce governance (what objectives, constraints, prompts, and retrieval rules are admissible), while Near-RT xApps execute bounded control actions only after lightweight verification using recent E2 observations and predefined safety envelopes, with all decisions logged via O1 for auditability and for closing the learning loop in the agentic memory. Under this view, the mitigation mechanisms proposed in our use cases map cleanly onto the figure; anchor randomization and negotiation protocols can be realized as xApp-level decision logic constrained by A1 policies; temporal/confirmation debiasing in collective memory is naturally hosted in SMO/Non-RT RIC as part of analytics and experience stores fed by O1/E2 telemetry; and final actuation remains standardized through E2 into the RAN components (O-CU/O-DU/O-RU). As a result, the key question is not whether bias mitigation can be integrated into O-RAN, but rather how to partition it across SMO/Non-RT versus Near-RT such that the mitigation is computationally feasible, interface-aligned, and operationally safe, while maintaining 3GPP-consistent functional splits and O-RAN-compliant control surfaces [29]–[33].

III. Cognitive Biases in LLM-Powered 6G Agents

Autonomous decision-making in O-RAN increasingly leverages LLM-powered agents for tasks such as slice orchestration, resource allocation, fault isolation, and multi-agent negotiation. While LLMs provide flexible reasoning and large-context retrieval, they also inherit - and can amplify - well-known cognitive biases. These biases arise from (i) how memories are retrieved and ranked, (ii) how prompts and peer messages condition priors, and (iii) how reward or trust signals are aggregated across episodes. In performance-sensitive settings such as 6G RAN control, even small systematic biases can cascade into large degradations. Table 1 enumerates the well-known biases and their emergence in 6G, wherefore we provide a companion exposition that explains each bias in greater depth: its mechanism, a compact mathematical view, a concrete 6G example, likely impacted agentic components, and pragmatic mitigation directions.

A. Overview of Biases, their Impact and Mitigation in 6G

1) Confirmation Bias

An LLM agent exhibits confirmation bias when it selectively retrieves or interprets evidence that supports its existing hypothesis, and thereby under-samples or ignores disconfirming data. Concretely, let $S(\mathcal{D}, H)$ denote a selection operator that depends on dataset $\mathcal{D}$ and hypothesis H. Then the posterior obtained from the selected data becomes

$$p(H \mid S(\mathcal{D}, H)) \;\propto\; p(H) \prod_{x \in S(\mathcal{D}, H)} p(x \mid H), \qquad (2)$$

which biases the posterior toward H by over-weighting supportive likelihood terms. Equivalently, if selection induces a proposal sampling distribution $q(x) \propto \mathbf{1}\{x \in S(\mathcal{D}, H)\}p(x)$, an unbiased estimator of an expectation $\mathbb{E}_p[f(X)]$ requires importance weights $w(x) = p(x)/q(x)$;

Table 1: Taxonomy of Cognitive Biases in LLM-Powered 6G Agents

Bias Category	**Definition**	**Impacted Agentic Components**	**Example of Mitigation**
Confirmation Bias [34], [35]	Agent selectively retrieves or interprets evidence supporting prior hypothesis. Example: xApp queries only low-load cells for PRB allocation, ignoring high-load ones.	Reasoning, Memory Queries, Telemetry Filtering	Enforce counterfactual queries or negotiation; cross-check telemetry from multiple sources before decision.
Recency / Primacy Bias [36], [37]	Agent over-weighs recent or first observations. Example: Fault-recovery agent acts only on last 10 min of RSRP measurements or initial fault logs.	Memory Access, Planning	Weighted evidence fusion; validate across full telemetry timeline before reconfiguration.
Anchoring Bias [5], [37]	Early proposals bias subsequent reasoning. Example: Energy optimization agent sticks to default power despite high SINR allowing lower settings.	Reasoning, Planning Initialization	Randomize anchor values; simulate multiple baseline scenarios before committing.
Availability Heuristic [37], [38]	Agent overestimates likelihood of salient/recent events. Example: Fault prediction overreacts to 3 recent XnAP resets, ignoring long-term failure rate.	Memory Retrieval, Risk Estimation	Normalize event likelihood using historical data; apply Bayesian priors.
Authority Bias [37], [39]	Overweighting outputs from trusted sources. Example: RIC accepts vendor KPI reports without cross-checking local counters.	API Trust Assignment, Reasoning	Blind source evaluation; dual-source validation before applying config changes.
Halo Effect [37], [40]	Success in one task inflates trust in unrelated tasks. Example: xApp trusted in latency tuning because it succeeded in energy optimization.	Peer Trust Model, Reasoning	Decay trust scores across domains; cross-validate with independent measurements.
Suggestion / Prompting Bias [37], [41]	Agent influenced by wording in prompts or peer proposals. Example: prompt *latency is the primary goal* disregards looking for optimal allocation.	Prompt Interpretation, API Feedback	Use neutral prompt templates; validate decisions via multi-agent cross-checks.
Groupthink / Herding [42], [43]	Agent combines its own weak evidence (PRB slit 1) with the signal that other agents prefer another option (PRB split 2). As a result, agent's posterior belief that 2 is the best option increases.	Multi-Agent Negotiation, Consensus	Prompt agents to generate at least one challenging proposal; simulate alternatives before consensus.
Framing Effect [37], [44]	Equivalent inputs produce different decisions. Example: PRB usage framed as 20% free vs 80% used changes allocation.	Reasoning, Prompt Interpretation, Memory	Canonicalize memory representation; standardize schemas; apply sensitivity checks.
Sunk Cost Fallacy [37]	Agent continues ineffective strategy due to prior investment. Example: agent's memory prioritizes records that justify the past investment in a specialized, high-capacity Massive MIMO antenna (a gNB node upgrade), causing it to ignore new spectral analysis showing the node is now underutilized.	Reasoning, Planning, Memory, Verification	Reset historical influence; include diminishing-return detection; flag persistent low-benefit loops.
Neglect of Probability / Uncertainty [5], [45]	Agent ignores uncertainty, acting on point estimates. Example: Self-healing xApp treats 60% alarm as certain.	Reasoning, Planning, Verification	Integrate Bayesian or ensemble-based uncertainty; apply confidence thresholds.
Status Quo Bias [37], [46]	Agent prefers current configuration due to overestimated switching risks. Example: RIC avoids updating cell configuration despite KPIs suggesting optimization.	Planning, Reasoning, Verification	Reduce inertia thresholds; simulate reversible changes.
Automation Bias [37]	Over-reliance on automated tool outputs without verification. Example: xApp propagates erroneous rApp fault detection.	Tool Use, Reasoning, Planning, Communication	Mandatory verification; confidence estimation; defer risky actions; share verification status across agents.
Survivorship Bias [37], [47]	Agent learns only from successes, ignoring failures. Example: a slice-scaling rApp agent trained only on successes ignores failures.	Memory Dataset, Reasoning, Planning	Ensure balanced success/failure sampling; simulate failures; verify performance on complete data.

failure to apply such correction yields a biased estimate:

$$\widehat{\mathbb{E}}[f] = \frac{\sum_{x\sim q} f(x)}{\sum_{x\sim q} 1} \quad \neq \quad \mathbb{E}_p[f(X)].$$

From an information-theoretic perspective, selection skew can be measured by the Kullback–Leibler divergence between the true posterior and the selection-conditioned posterior,

$$\mathrm{KL}\big(p(H \mid \mathcal{D}) \mid\mid p(H \mid S(\mathcal{D}, H))\big),$$

which grows as selection concentrates evidence supporting H [35], [48].

In O-RAN, for example, an xApp that diagnoses "scheduler congestion" may query only E2 KPIs that corroborate congestion while ignoring interfering telemetry (e.g., interference indicators). Consequently the agent's memory retrieval, reasoning/planning and tool use pipelines each amplify the flawed hypothesis: (i) retrieval will preferentially surface supporting episodes, (ii) reasoning will prematurely converge on a single explanation, and (iii) tool queries and simulations will be framed to validate the hypothesis rather than to falsify it.

Mitigation: To reduce confirmation bias we can combine sampling-correction, explicit falsification, and exploration regularization. In *memory retrieval* enforce *symmetric sampling* across competing hypotheses by drawing samples from both q_{support} and q_{refute} with controlled proportions, and apply importance weights to recover unbiased estimates:

$$\widehat{\mathbb{E}}_p[f] \;=\; \frac{\sum_{x\sim q_{\text{support}}\cup q_{\text{refute}}} w(x) f(x)}{\sum_x w(x)}, \qquad w(x) = \frac{p(x)}{q(x)}.$$

During *reasoning* implement *counterfactual queries* that explicitly compute alternative-likelihood ratios,

$$\Lambda(x) \;=\; \frac{p(x \mid H_{\text{alt}})}{p(x \mid H_{\text{curr}})},$$

and preferentially surface cases with large Λ to challenge H_{curr}. In *planning* add an exploration regularizer (entropy or uncertainty bonus) to the objective:

$$\pi^\star = \arg\max_\pi \mathbb{E}_{a\sim\pi}[U(a)] + \beta\mathcal{H}(\pi),$$

which discourages overly deterministic, belief-consistent plans [49]. Finally, in *communication* adopt a challenge–response protocol where peer agents must produce disconfirming evidence sampled under an alternative-policy prior before consensus is reached. These interventions reduce the KL shift induced by selective evidence and re-align posterior estimates with the full data-generating process [35], [48].

2) Recency and Primacy Bias

Recency (over-weighting recent data) and primacy (over-weighting early data) can be modeled through temporal weighting of observations. A common weighted estimator is

$$\hat{\theta} \;=\; \frac{\sum_t w_t x_t}{\sum_t w_t}, \qquad w_t \propto e^{-\lambda|T-t|}, \tag{3}$$

where a large λ concentrates mass around the most recent ($t \approx T$) or earliest ($t \approx 0$) samples, producing strong recency or primacy effects respectively. Equivalently, an Exponentially Weighted Moving Average (EWMA) obeys the recursive form

$$\hat{\theta}_t = \alpha x_t + (1-\alpha)\hat{\theta}_{t-1},$$

which is mathematically identical to a first-order Kalman filter under constant observation and process noise assumptions [50]. Overly large α (equivalently large λ) yields recency bias, while very small α yields primacy-like inertia.

In practice, fault-management xApps that react to short E2 spikes or that never update thresholds illustrate these failure modes: short-term E2 spikes dominate decisions in the former (recency), while initial thresholds remain dominant in the latter (primacy).

Mitigation: Balance can be achieved by multi-horizon estimators, change-point detection, and temporal confidence bounds. Concretely: in *memory* calibrate decay functions across horizons by maintaining parallel estimators $\{\hat{\theta}^{(h)}\}_{h=1}^H$ with different decay rates α_h and fuse them via weighted averaging,

$$\hat{\theta}_{\text{fused}} = \sum_{h=1}^{H} \omega_h \hat{\theta}^{(h)}, \qquad \sum_h \omega_h = 1.$$

In *reasoning*, detect change points by monitoring likelihood-ratio statistics or CUSUM statistics and only adapt aggressively when the change is statistically significant [51]. During *planning*, use multi-window averaging or evidence fusion across windows to avoid overreacting to anomalies. Finally, *verification* can enforce temporal confidence bounds (e.g., via bootstrap or Bayesian credible intervals) to prevent reconfigurations driven by transient fluctuations [48].

3) Anchoring Bias

Anchoring occurs when an early piece of information a_0 constrains subsequent optimization or negotiation; for instance one can express the anchored decision as a regularized maximization

$$a^\star = \arg\max_a \big(U(a) - \gamma\, d(a, a_0)\big), \tag{4}$$

where $d(\cdot,\cdot)$ is a deviation penalty and γ controls the anchor's strength. This formulation is closely related to regularized empirical risk minimization where an initial estimate acts as a strong prior.

In a resource negotiation example, initial PRB proposals bias the negotiation path (Fig. 3). Anchoring affects (i) memory/retrieval by biasing which historical outcomes are retrieved (those near a_0), (ii) reasoning/planning by limiting search to neighborhoods of a_0 even when simulations indicate different optima, and (iii) tool use by restricting DT validation to small perturbations around the anchor.

Mitigation: Reduce anchoring via randomized initialization, decay of anchor influence, and anchor-independent

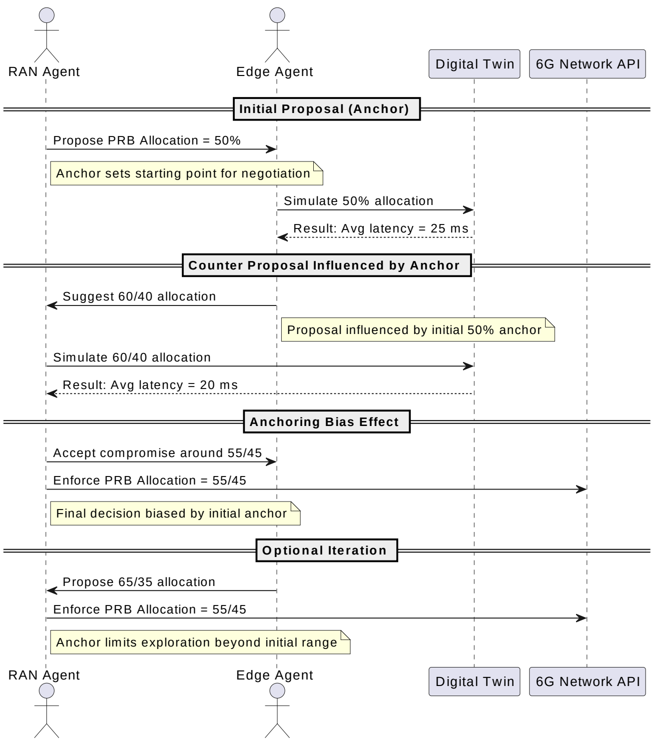


Figure 3: Anchoring bias in 6G agentic negotiation.

verification. During *planning initialization*, randomize anchor values or sample several anchor candidates $\{a_0^{(k)}\}$ to diversify starting points. In *reasoning* apply a temporal decay to the anchor penalty, e.g.,

$$\gamma_t = \gamma_0 e^{-\eta t},$$

so the anchor influence wanes as new evidence accumulates. Within *communication*, refresh shared embeddings or priors to avoid inherited anchoring, and perform *anchor-independent verification* by re-evaluating decisions under perturbed initial conditions to test robustness [5].

4) **Availability Bias**

Availability bias arises when the agent overestimates the probability of salient or easily retrievable events. If retrieval has a bias function $r(x) \geq 0$ that modulates the chance an item x is recalled, the estimated probability of an event E from retrieved samples becomes

$$\hat{p}(E) = \frac{\sum_x r(x)\mathbf{1}[x \text{ evidences } E]}{\sum_x r(x)}. \tag{5}$$

This estimator is biased away from the true event probability $p(E)$ unless $r(x) \propto 1$ (uniform). For rare-but-memorable alarms, $r(x)$ is large and $\hat{p}(E)$ overestimates true risk.

Availability bias thus distorts (i) memory and retrieval—favoring high-salience episodes, (ii) reasoning and planning—focusing on easily available scenarios rather than base rates, and (iii) tool use—preferring quick summary tools instead of comprehensive simulations.

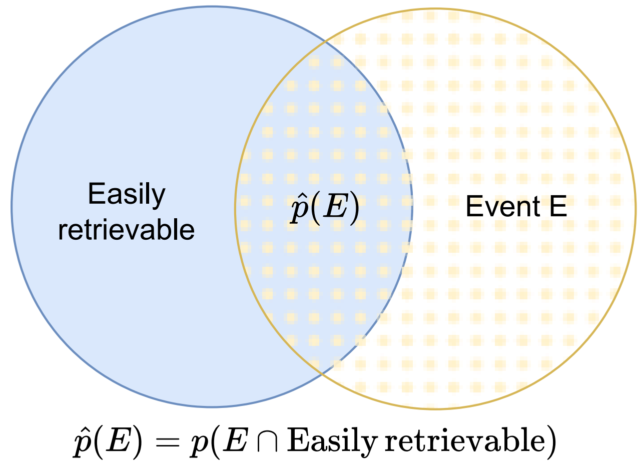


Figure 4: Availability bias concept.

Mitigation: Use inverse-probability weighting and stratified sampling. In *memory* apply an inverse-salience reweighting to retrieved items,

$$w_{\text{inv}}(x) \propto \frac{1}{r(x)+\epsilon},$$

so that highly salient items are down-weighted during aggregation. During *reasoning*, introduce *salience normalization* via prompts or internal circuitry that enforces base-rate correction (i.e., explicitly combine prior $p(E)$ with retrieved evidence likelihoods). In *planning* and *communication*, enforce *stratified sampling* across event frequency bins to ensure rare but critical events are proportionally represented in deliberation.

5) **Authority Bias**

Authority bias occurs when outputs from a trusted source are over-weighted. A simple linear fusion model is

$$\hat{y} = \tau_s y_s + (1-\tau_s) y_{\text{local}}, \qquad \tau_s \in [0,1], \tag{6}$$

and authority bias corresponds to $\tau_s \gg 0$ even when y_s conflicts with local evidence. More generally, let the trust parameter τ_s be adaptively updated from observed performance; then a principled update can be

$$\tau_{s,t+1} = \frac{\exp\left(\alpha\, \text{perf}_{s,t}\right)}{\sum_{s'} \exp\left(\alpha\, \text{perf}_{s',t}\right)},$$

which is a softmax calibration of source reliabilities based on recent accuracy or calibrated likelihoods [52].

Authority bias impacts (i) reasoning/planning, where proposals from an "expert" agent are accepted without independent validation, and (ii) tool use, where a single authoritative API dominates decisions.

Mitigation: Implement dynamic trust calibration, cross-validation, and redundancy. In *reasoning*, compute τ_s from

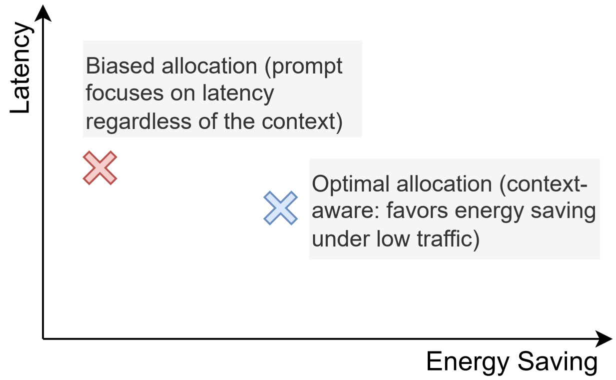


Figure 5: Suggestion Bias shifts the agent's decision in the multi-objective space.

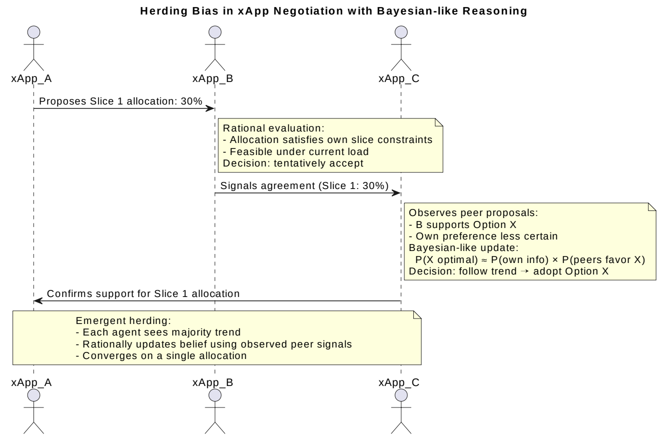


Figure 6: Groupthink emergence.

an exponentially weighted accuracy track-record and contextual similarity metrics, and decay trust when conflicts with local telemetry appear. During *planning*, perform cross-validation of authoritative recommendations against independent DT simulations or ensemble models (e.g., compare y_s to $\bar{y}_{\text{ensemble}}$ and reject if $|y_s - \bar{y}_{\text{ensemble}}|$ exceeds a threshold). In *communication*, require multiple corroborating reports before full acceptance of an authoritative recommendation [52].

6) **Halo Effect**

The halo effect occurs when success on one task inflates trust in other tasks. Let $\tau_t^{(j)}$ be the trust in agent (or module) j at time t. A simple cross-task update is

$$\tau_{t+1}^{(j)} = \tau_t^{(j)} + \eta\, \mathbf{1}(\text{success}_t^{(i)})\, \rho_{ij}, \tag{7}$$

where ρ_{ij} measures task similarity and η is a learning rate. If ρ_{ij} is overestimated the halo effect causes inappropriate cross-domain transfer of confidence (e.g., an xApp that solved energy tuning being over-trusted in latency control).

Mitigation: Constrain cross-domain generalization by maintaining domain-separated latent representations and gating cross-task influence. In *reasoning* maintain separate knowledge partitions or embeddings for each operational domain and only permit transfer when empirical cross-task correlation exceeds a validated statistical threshold. During *planning*, apply a cross-task influence gate $g_{ij} \in [0, 1]$ such that effective trust becomes $\tilde{\tau}^{(j)} = g_{ij}\tau^{(j)}$, and update g_{ij} only when ρ_{ij} is statistically validated (e.g., via hypothesis testing). In *communication*, always include explicit domain context and task-specific performance metrics to prevent blind transfer of authority.

7) **Suggestion / Prompting Bias**

Prompt wording shifts the model's logits by an additive or multiplicative bias. A first-order model is

$$\text{logits} \leftarrow \text{logits} + \beta\, s(\text{prompt}), \tag{8}$$

where $s(\cdot)$ encodes prompt-derived bias and β scales its strength. For example, the prompt "latency is the primary goal" corresponds to a score vector s that elevates latency-optimizing actions, thereby steering the agent toward those actions even when they may not be globally optimal.

This affects (i) memory/retrieval by seeding queries with leading phrases that bias fetched records, (ii) reasoning/planning by introducing prior assumptions into chain-of-thought, and (iii) tool use by constraining DT simulation ranges.

Mitigation: Normalize and track provenance of prompts, detect distributional shifts, and perform controlled re-prompting. In *communication*, canonicalize prompts to neutral forms before downstream processing. In *memory*, store prompt provenance metadata and include provenance-aware de-biasing when scoring retrieved items. In *reasoning*, monitor prompt-embedding statistics (e.g., detect mean shifts) and trigger controlled re-evaluation under semantically equivalent but neutrally phrased prompts. Additionally, run DT validation under broadened parameter ranges rather than the narrow ranges implied by the original prompt to guard against framing-induced narrow search.

8) **Groupthink / Herding**

Groupthink or herding arises when conformity to peer agents constrains independent reasoning. In a consensus-regularized multi-agent objective:

$$\min_{a_i} \sum_i \mathcal{L}_i(a_i) + \lambda \sum_i \|a_i - \bar{a}\|^2, \tag{9}$$

a high λ forces rapid consensus. Consider an agent C that is uncertain: after observing peers favoring option X, C performs a Bayesian-like combination of its own signal and the social signal. In a simple approximation,

$$\Pr(X_{\text{optimal}} \mid \text{data},\ \text{peers}) \propto \Pr(\text{own info}) \times \Pr(\text{peers favor } X), \tag{10}$$

which illustrates how weak private evidence can be overwhelmed by strong social evidence [53]. This social reinforcement reduces diversity and can lead to brittle consensus.

Mitigation: Preserve diversity via independent rollouts, ensemble disagreement, dissent propagation, and dynamic consensus relaxation. In *planning*, require independent rollouts of candidate strategies before synchronization, and in *reasoning* introduce an *ensemble disagreement* objective that rewards hypothesis diversity, e.g.,

$$\min_{\theta_1,\dots,\theta_M} \sum_m \mathcal{L}(\theta_m) - \gamma \sum_{m \neq m'} \mathrm{D}(\theta_m, \theta_{m'}),$$

where D is a diversity-promoting functional. In *communication*, adopt dissent-propagation protocols to surface minority viewpoints, and dynamically reduce the consensus regularizer λ_t over deliberation rounds:

$$\lambda_{t+1} = \lambda_t \cdot \delta, \qquad \delta \in (0,1),$$

so conformity pressure relaxes as more evidence accumulates.

Remarks on combining mitigations. Many of the mitigation strategies above share common building blocks from statistical learning and decision theory: importance weighting (for selection bias), multi-horizon fusion and change-point detection (for temporal bias), entropy/uncertainty regularization (for exploration vs. exploitation), cross-validation and softmax-based trust calibration (for authority/halo), and ensemble/diversity objectives (for groupthink). Where appropriate, these mechanisms should be deployed together in the agentic pipeline (memory, reasoning, planning, tool use, and communication) so that one module's counter-biasing does not inadvertently create another bias downstream. The short bibliography below lists accessible references for the learning-theoretic tools and the behavioural origins of the biases.

9) Framing Effect

Framing occurs when semantically equivalent descriptions of the same underlying state lead to different decisions, and thus the agent's action depends on the presentation map (or frame). Formally, let ϕ, ψ be two framing maps with $\phi(x) \sim \psi(x)$ (i.e., they convey the same informational content up to re-encoding), yet the agent's policy $a(\cdot)$ satisfies

$$a(\phi(x)) \neq a(\psi(x)).$$

To ground this behavior in decision theory, consider a prospect-theory style value function $v(\cdot)$ and a frame-dependent reference point r_ϕ induced by frame ϕ. Then the framed decision can be written as the solution of a framed utility maximization

$$a^\star_\phi \;=\; \arg\max_a \; \mathbb{E}\big[v\big(R(a) - r_\phi\big)\big], \tag{11}$$

where $R(a)$ is the (random) reward of action a. In prospect-theory parameterizations one commonly uses

$$v(u) = \begin{cases} u^\alpha, & u \geq 0, \\ -\lambda(-u)^\beta, & u < 0, \end{cases} \qquad \alpha, \beta \in (0,1),\ \lambda > 1,$$

so that changing the reference r_ϕ (a gain vs. loss framing) systematically alters the sign and curvature of the argument to $v(\cdot)$ and hence the optimizer in (11) [37], [44]. Equivalently, framing can be captured as a change in the agent's internal prior or loss function: $\ell_\phi(a) \neq \ell_\psi(a)$, producing different minimizers $\arg\min_a \mathbb{E}[\ell_\phi(a;x)]$ and $\arg\min_a \mathbb{E}[\ell_\psi(a;x)]$ even when $\phi(x) \sim \psi(x)$.

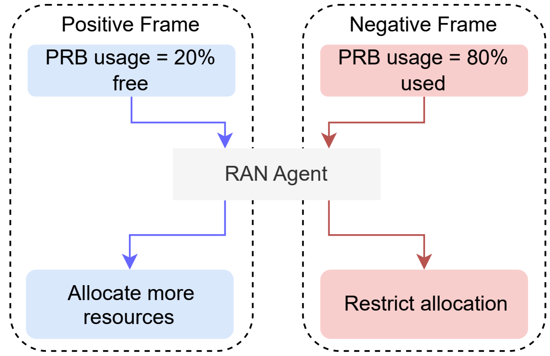


Figure 7: Framing effect concept.

For example, PRB usage described as "20% free" versus "80% used" corresponds to different reference points r_ϕ and r_ψ and therefore to different framed utilities in (11), which changes the allocation decision (cf. Fig. 7). Consequently, framing influences (i) memory/retrieval, because prompts phrased to emphasize gains preferentially surface gain-framed records; (ii) reasoning/planning, because loss-framed descriptions induce more conservative, risk-averse plans; and (iii) tool use, because labeled metrics ("high utilization" vs. "efficient usage") guide downstream validation and reconfiguration.

Mitigation: To counter framing bias we should (i) canonicalize internal representations so that semantically equivalent inputs map to the same latent encoding, i.e. enforce $\mathcal{E}(\phi(x)) \approx \mathcal{E}(\psi(x))$, (ii) train for paraphrase- and reference-invariance by augmenting training data with alternative framings and enforcing stability of the decision rule (e.g., add a training penalty $\|\pi(\phi(x)) - \pi(\psi(x))\|^2$), and (iii) implement a verification layer that performs sensitivity checks by re-evaluating decisions under systematically rephrased inputs and by comparing framed optima $a^\star_\phi, a^\star_\psi$. These steps create explicit invariances in both representation and objective, thereby reducing frame-driven divergences [37], [44].

Table 2: Theoretical and Learning-Theoretic Characterization of Cognitive Biases in LLM-Based 6G Agents

Bias	**Learning-Theoretic Origin**	**Mathematical Abstraction**	**Primary Computational Mechanism**	**Related Theory and References**
Confirmation	Biased posterior updating and selective sampling	$p(H\|S(\mathcal{D},H)) \propto p(H)\prod_{x\in S} p(x\|H)$	Hypothesis-consistent evidence filtering	Bayesian inference, PAC-Bayes [35], [48], [54]
Recency / Primacy	Temporal discounting and non-stationary estimation	$\hat{\theta} = \frac{\sum_t w_t x_t}{\sum_t w_t}$	Exponential decay of historical samples	Reinforcement learning, adaptive filtering [49], [50]
Anchoring	Regularized utility optimization with priors	$\arg\max_a U(a) - \gamma d(a, a_0)$	Strong prior anchoring in objective landscape	Bayesian priors, constrained optimization [5], [48]
Availability	Salience-weighted sampling and retrieval bias	$\hat{p}(E) = \frac{\sum_x r(x)\mathbf{1}_E}{\sum_x r(x)}$	Over-representation of vivid samples	Sampling bias, heuristic inference [5]
Authority	Source-weighted decision fusion	$\hat{y} = \tau_s y_s + (1-\tau_s) y_l$	Static trust over-weighting	Trust modeling, opinion fusion [48], [52]
Halo Effect	Cross-task confidence propagation	$\tau^{(j)}_{t+1} = \tau^{(j)}_t + \eta\rho_{ij}$	Unconstrained transfer learning	Multi-task learning, transfer bias [49]
Suggestion / Prompting	Logit-space perturbation and conditioning	logits $\leftarrow$ logits $+ \beta s$	Prompt-induced activation shifts	Representation learning, prompting theory
Groupthink / Herding	Consensus regularization and social learning	$\min \sum_i \mathcal{L}_i + \lambda \|a_i - \bar{a}\|^2$	Social signal amplification	Informational cascades, game theory [53]
Framing Effect	Utility distortion under representation changes	$a(\phi(x)) \neq a(\psi(x))$	Context-dependent utility mapping	Prospect theory, invariance learning [5]
Sunk Cost	Path-dependent utility accumulation	$U = \mathbb{E}[B] - C + \alpha f(S)$	Irrecoverable-cost reinforcement	Dynamic programming, behavioral economics [5]
Neglect of Uncertainty	Point-estimate optimization	$\arg\max_a U(a, \mathbb{E}[x])$	Variance suppression	Bayesian learning, ensemble methods [48]
Status Quo	Switching-cost regularization	$\Delta U > c_{\text{switch}}$	Inertia-dominated optimization	Decision theory, risk aversion [5]
Automation	Tool-output dominance	$a = f(\text{tool})$	Verification bypass	Human-in-the-loop systems
Survivorship	Selection-truncated learning	$\mathbb{E}[\hat{\theta}\|S=1] \neq \theta$	Failure-data elimination	Selection bias, empirical risk [55]

10) Sunk Cost Fallacy

The sunk cost fallacy arises when past, irrecoverable investments S improperly affect present choices. While rational decision theory dictates that sunk costs should not influence future-optimal actions, biased agents internalize an additive utility term dependent on S:

$$U_{\text{biased}}(a) = \mathbb{E}[B(a)] - C(a) + \alpha f(S), \quad (12)$$

with $\alpha > 0$ and $f(\cdot)$ an increasing function (e.g., $f(S) = \log(1+S)$). In an MDP formalism, the correct Bellman equation for the value of state s is

$$V^\star(s) = \max_a \left[r(s,a) + \gamma \mathbb{E}_{s'|s,a} V^\star(s')\right],$$

and it should ignore costs that do not affect future transitions or rewards. However, if the agent augments instantaneous reward $r(s,a)$ with a sunk-cost term depending on historical investment S (so that $r_{\text{biased}}(s,a) = r(s,a) + \alpha f(S)$), then its learned policy π_{biased} solves a different dynamic program and therefore can be strictly suboptimal relative to the true optimal policy $\pi^\star$ that correctly treats S as irrelevant to future payoffs [46], [56].

Practically, this mechanism explains why an agent might continue optimizing a massive MIMO gNB node—because S (the expensive upgrade) inflates the perceived utility of actions that defend that investment—despite spectral evidence suggesting reallocation to backhaul optimization would be more efficient. Thus sunk cost affects (i) memory/retrieval by prioritizing supporting records, (ii) reasoning/planning by biasing optimization loops, and (iii) tool use by focusing simulations on marginal improvements around the sunk asset.

Mitigation: Countermeasures include (i) periodically *resetting historical influence* by projecting state representations to remove or discount historical-investment components (e.g., replace S with $\tilde{S} = \lambda_S S$ with $\lambda_S \downarrow 0$ over time), (ii) including diminishing-return detectors that monitor marginal gains $\Delta B/\Delta\text{effort}$ and trigger reallocation when these fall below thresholds, and (iii) tagging irrecoverable expenditures in memory so they are treated as annotated history rather

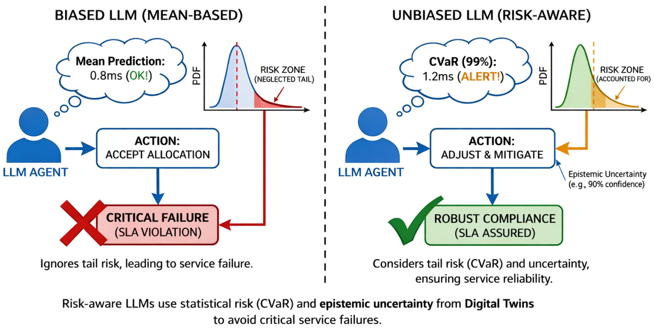


Figure 8: Uncertainty negelect by reasoning over the mean instead of the tail.

than continuing objectives. Importantly, treating S as an exogenous label (and not as part of the reward) restores correct Bellman updates and prevents the agent's dynamic program from being misspecified [37], [56].

11) Neglect of Uncertainty

Neglect of uncertainty occurs when an agent reduces rich probabilistic beliefs to point estimates and then optimizes as though those estimates were certain. Concretely, the biased decision rule often takes the form

$$a_{\text{biased}} \;=\; \arg\max_{a} \; U\big(a,\; \mathbb{E}[x]\big), \tag{13}$$

whereas the Bayes-optimal decision minimizes Bayes risk

$$\delta^{\star} \;=\; \arg\min_{\delta} \; \mathbb{E}_{\theta}\mathbb{E}_{x|\theta}\big[L(\theta,\delta(x))\big], \tag{14}$$

thereby integrating model uncertainty into action selection [48], [57]. Ignoring uncertainty is equivalent to replacing the posterior $p(\theta \mid \mathcal{D})$ by a delta at a point estimate $\hat{\theta}$, which in general increases expected loss:

$$\mathbb{E}_{\theta}\mathbb{E}_{x|\theta}L(\theta,\delta_{\hat{\theta}}(x)) \;\geq\; \mathbb{E}_{\theta}\mathbb{E}_{x|\theta}L(\theta,\delta^{\star}(x)). \tag{15}$$

From a practical standpoint, neglect manifests when a self-healing xApp treats a 60% confidence alarm as if it were certain and thus executes an all-or-nothing action. This bias corrupts (i) memory/retrieval by storing point estimates without variances, (ii) reasoning/planning by optimizing for expected values without risk buffers, and (iii) tool use by ignoring predictive error bars.

Mitigation: Remedies include (i) integrating Bayesian or ensemble methods to propagate posterior uncertainty through downstream decision modules (e.g., use posterior predictive distributions in (14)), (ii) employing risk-aware objectives such as Conditional Value-at-Risk (CVaR) or robust max-min criteria

$$\max_{a} \; \min_{q\in\mathcal{Q}} \mathbb{E}_{\theta\sim q}[U(a,\theta)], \tag{16}$$

to protect against tail risk, and (iii) enforcing confidence thresholds before actioning critical changes. Implementation can rely on approximate Bayesian techniques (dropout-as-Bayesian [58]) or explicit ensemble variance estimation [48], [59].

12) Status Quo Bias

Status quo bias reflects an aversion to changes whose immediate cost (or perceived risk) outweighs the expected gain. A simple threshold condition captures this:

$$\Delta U \;=\; U_{\text{new}} - U_{\text{current}} \;>\; c_{\text{switch}}, \tag{17}$$

where c_{switch} is an (often overestimated) switching cost. In a sequential decision model with discounting, adding a switching penalty to the reward yields

$$r_{\text{biased}}(s,a,s') = r(s,a) - \mathbf{1}\{a \neq a_{\text{current}}\}\, c_{\text{switch}}, \tag{18}$$

which produces inertia: the optimal policy will prefer a continuation action unless the long-run gain outweighs c_{switch}. Thus, even when a re-slice (e.g., 60/40) yields a long-term 15% efficiency improvement as predicted by a DT, the agent may prefer incremental 1% tweaks if it overestimates short-term disruption.

Mitigation: To overcome status quo bias, (i) make opportunity costs explicit by augmenting tool outputs with quantitative estimates of lost gain from inaction, (ii) simulate failure modes for the current configuration to expose hidden risks, and (iii) include planned, reversible change experiments (A/B rollouts) that reduce perceived switching costs and provide empirical evidence about transient losses versus long-term gains [46].

13) Automation Bias

Automation bias is the undue deference to automated outputs in place of independent verification. Let the automated tool produce recommendation y with reliability parameter $\tau \equiv \Pr(y \text{ correct})$. Suppose verification costs c_v and that verifying reduces expected loss by ΔL. A rational agent verifies iff $\Delta L > c_v$. However, an automation-biased agent skips verification, effectively following

$$a = f(y) \tag{19}$$

instead of the verification-aware rule

$$a = f\big(y,\; \text{verify}(\cdot)\big), \tag{20}$$

and hence it incurs additional expected loss when τ is low or nonstationary [60].

In networks, this explains immediate, unverified cell shutdowns following a faulty rApp alarm. The bias therefore degrades (i) reasoning/planning by short-circuiting independent checks, (ii) tool use/verification by bypassing corroboration, and (iii) communication by propagating unverified signals.

Mitigation: Introduce mandatory verification gates for high-impact actions, propagate verification status and confidence scores with every automated output, and compute expected-value-of-information for verification so that the

Table 3: Operational Impact of Cognitive Biases in LLM-Driven 6G Agentic Systems

Bias	Affected Pipeline Stages	Typical 6G Scenario	Primary Failure Mode	Key Mitigation Strategies
Confirmation	Memory, Reasoning, Tool Use	Selective KPI querying in congestion diagnosis	Persistent misdiagnosis	Symmetric retrieval, counterfactual reasoning
Recency / Primacy	Memory, Planning, Tools	Overreaction to short-term spikes	Unstable configurations	Multi-horizon averaging, change-point detection
Anchoring	Reasoning, Planning, Communication	PRB negotiation initialization	Narrow bargaining range	Anchor randomization, decay weighting
Availability	Memory, Planning, Tools	Alarm log overreaction	Poor rare-event preparedness	Inverse-salience weighting, stratified sampling
Authority	Reasoning, Tools, Communication	Vendor rApp dominance	Unverified acceptance	Dynamic trust calibration, cross-validation
Halo Effect	Reasoning, Tools	Over-trust of successful xApp	Cross-domain misconfiguration	Domain separation, influence gating
Suggestion / Prompting	Memory, Reasoning, Tools	Directive prompt bias	Predetermined optimization	Prompt normalization, re-prompting
Groupthink / Herding	Memory, Planning, Coordination	Slice allocation consensus	Premature convergence	Independent rollouts, dissent propagation
Framing Effect	Memory, Reasoning, Tools	Gain/loss KPI presentation	Risk miscalibration	Canonicalization, paraphrase invariance
Sunk Cost	Memory, Planning, Tools	Persistent gNB optimization	Resource lock-in	Historical discounting, loop detection
Neglect of Uncertainty	Memory, Reasoning, Tools	Deterministic alarm response	Brittle allocations	Bayesian inference, uncertainty exchange
Status Quo	Planning, Tools	Static spectrum slicing	Opportunity loss	Failure modeling, cost quantification
Automation	Reasoning, Verification, Communication	Fault-triggered shutdown	Cascading failures	Mandatory verification, confidence tracking
Survivorship	Memory, Reasoning, Planning	Success-only training data	Over-optimistic policies	Failure logging, inverse weighting

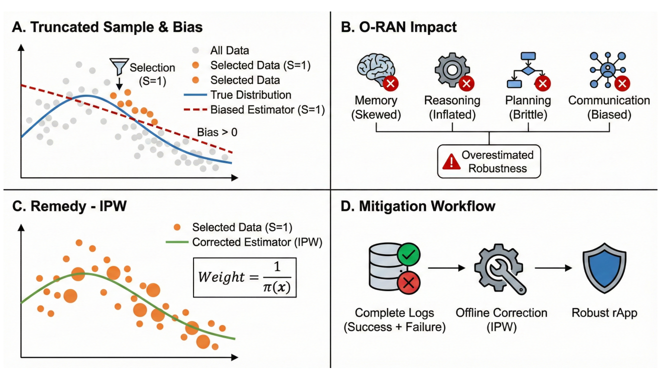


Figure 9: Concept and mitigation of the survivorship bias in telecom.

agent makes explicit, cost-sensitive checks (verify iff $\Delta L > c_v$). In multi-agent settings, require corroboration from independent sensors or models before committing to irreversible actions [60].

14) Survivorship Bias

Survivorship bias appears when learning uses only successful (surviving) instances $S = 1$, which truncates the sample and biases estimators. Formally, if $\hat{\theta}$ is estimated from the conditional distribution $p(x \mid S = 1)$, then

$$\mathbb{E}[\hat{\theta} \mid S = 1] - \theta_{\text{true}} \neq 0, \tag{21}$$

whenever the selection mechanism depends on unobserved variables correlated with the target. This is a classical selection problem; Heckman's correction and inverse-probability weighting (IPW) are standard remedies. In particular, an unbiased estimator can be recovered by reweighting samples with the inverse selection probability:

$$\widehat{\mathbb{E}}_{\text{corrected}}[f(X)] = \frac{\sum_{i:S_i=1} \frac{1}{\pi(X_i)} f(X_i)}{\sum_{i:S_i=1} \frac{1}{\pi(X_i)}}, \tag{22}$$

where $\pi(x) = \Pr(S = 1 \mid x)$ [61], [62].

In the O-RAN context, a slice-scaling rApp trained only on deployed (successful) configurations will ignore failure modes and thus overestimate robustness. Therefore survivorship bias impacts (i) memory by selective storage of winners, (ii) reasoning by inflating success probabilities, (iii) planning by preferring brittle strategies, and (iv) communication by propagating skewed summaries.

Mitigation: Maintain complete success/failure logs, apply inverse-truncation weighting or Heckman-style selection corrections during offline learning, simulate failures in planning to probe robustness, and ensure training/evaluation datasets intentionally include negative examples and decommissioned-element diagnostics [61], [62].

B. System-Level Bias Mitigation in 6G Architectures

Beyond representational and agent-level mitigation detailed before, achieving robust and bias-resilient autonomy in 6G networks requires system-level mitigation mechanisms that operate across orchestration layers. One promising approach is the integration of *adversarial evaluation loops*, where agents are periodically challenged with worst-case or counterfactual scenarios designed to expose latent biases in their decision policies. These loops can detect emergent herding, overfitting to framing effects, providing feedback for reasoning and planning. System-level mitigation also includes ensemble-based orchestration, in which multiple independently trained reasoning agents contribute to decisions, with their outputs aggregated to reduce the impact of individual biases. Within 6G O-RAN, these techniques can be embedded into the Service Management and Orchestration (SMO) layer, allowing xApps and rApps to participate in continuous adversarial evaluation cycles, log bias-relevant metrics, and adjust internal priors dynamically. The overarching principle is that bias-awareness must extend from local agent cognition to network-wide orchestration, ensuring that both the emergent behaviors of distributed agents and the global system policies remain robust against subtle cognitive and systemic biases.

C. Discussion on the Costs of Bias-Mitigation in Agentic Systems

The transition from heuristic-driven Large Language Model (LLM) agents to formally mitigated agentic systems introduces a *mitigation tax* that must be weighed against the gains in reliability. While the proposed strategies effectively align posterior estimates with the true data-generating process, they impose non-trivial burdens across several dimensions. The incorporation of these methods introduces four primary categories of cost, namely, i) **Inference and Token Consumption:** To counter confirmation bias, replacing greedy retrieval with symmetric sampling (Eq. 6) across $q_{support}$ and q_{refute} essentially doubles the input context and retrieval operations. Furthermore, computing counterfactual likelihood ratios $\Lambda(x)$ (Eq. 7) necessitates additional forward passes to evaluate alternative-likelihoods, leading to a significant increase in total tokens per decision cycle. ii) **Structural Complexity and Statefulness:** Mitigating temporal biases through multi-horizon estimators (Eq. 11) requires the agent's architecture to maintain H parallel versions of the network state. This transforms a stateless agent into a high-overhead stateful system that must continuously track cumulative sum (CUSUM) statistics and temporal confidence bounds, placing a heavy burden on edge-device memory and storage. iii) **Convergence Latency and Agility:** The introduction of an exploration regularizer $\beta H(\pi)$ (Eq. 8) into the planning objective intentionally discourages deterministic behavior. While this prevents anchoring, it inherently slows down policy convergence, potentially introducing "decision inertia" in scenarios requiring sub-millisecond reactions, such as rapid beam-steering or fault recovery. iv) **Communication and Protocol Overhead:** Challenge-response protocols for multi-agent negotiation require peer agents to generate disconfirming evidence before reaching consensus. This expands the communication topology from a simple broadcast to an iterative, multi-turn dialogue, significantly increasing the signaling load on the 6G management plane.

Table 4: Operational Costs of Bias-Mitigation

Bias / Mitigation	Resource Impact	Perf. Trade-off
Confirmation Importance Sampling	High GPU Compute	Latency increase
Recency/Primacy Multi-horizon $\hat{\theta}^{(h)}$	Memory Footprint	State Complexity
Anchoring Reg. $H(\pi)$	Iterative Planning	Slower Convergence
Groupthink Challenge-Response	Signaling Load	Bandwidth Spike
Framing Canonicalization	CPU Overhead	System Rigidity
Uncertainty Bayesian Priors	Logic Overhead	Conservative Action

In summary, the cost of fairness in 6G agentic systems is fundamentally a trade-off against throughput. A "biased" agent may reach a decision in a single inference pass, whereas a "mitigated" agent requires a sophisticated ensemble of statistical checks and multi-turn reasoning. However, in critical infrastructure, the potential catastrophic cost of a biased decision (e.g., localized network blackouts or massive resource underutilization) justifies the incremental investment in these robust computational safeguards.

D. General Note on Baselines

As noted by Xi et al. [63], comparing LLM agents to specialized DRL baselines often results in a category error because LLMs are designed for generalized reasoning and complex tool-use in open-ended environments, whereas DRL excels in narrow, high-frequency control loops. Therefore, to ensure a fair and insightful evaluation, our comparisons focus

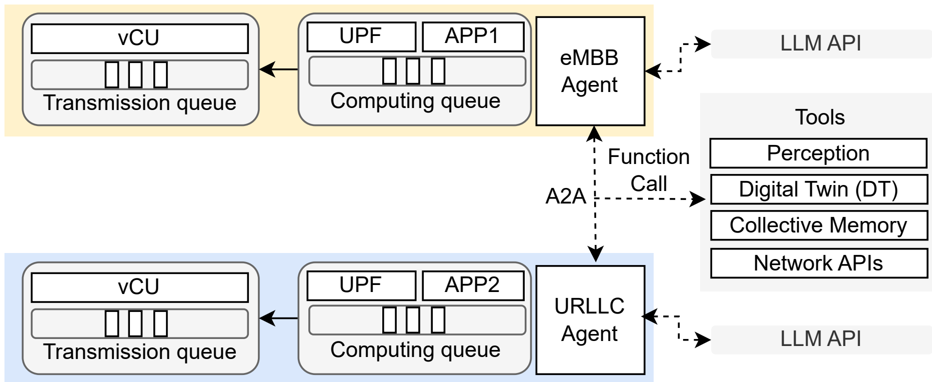


Figure 10: Use-case 1 setup.

on isolating the impact of cognitive bias mitigation within the LLM architecture itself.

IV. Use Case 1: Mitigating Anchoring Bias in Network Slicing Agentic Negotiation

A. Setup

We study the mitigation of anchoring bias in a 6G system by focusing on the dynamic allocation of the shared RAN bandwidth (60 MHz) between the eMBB and URLLC network slices[1] as depicted in Figure 10. The system architecture employs separate, autonomous LLM-powered agents, each representing a slice's resource manager, optimizing distinct objectives under strict SLA tail-latency requirements (50 ms for eMBB and 10 ms for URLLC).

To solve the critical issue of high response times traditionally incurred when using distant commercial LLMs (e.g., Gemini) for real-time control, the agents in this framework are powered by a locally hosted Small Language Model (SLM), specifically the lightweight `otel-llm-1b-it` model. This ensures sub-second inference latencies (0.95s mean), making the negotiation protocol fully compatible with the operational timescales of the O-RAN non-Real-Time RAN Intelligent Controller (non-RT RIC).

The agents operate in an iterative negotiation framework, following a three-stage loop within each trial: i) *Digital Twin (DT) consultation,* where agents synchronize their internal Digital Twins to the real network state. The DT uses a fluid-flow queuing model, rigorously evaluated via Conditional Value at Risk (CVaR), to predict the strict tail-latency and energy consumption for any candidate bandwidth proposal based on realistic O-RAN traffic bursts [64]; ii) *Memory consultation:* Agents consult a shared vanilla memory of past successful agreements to inform their negotiation strategy; and iii) *Negotiation protocol,* where agents engage in a fixed-round, turn-based protocol utilizing PID-like proportional step sizes to reach a feasible joint bandwidth agreement where both agents' utilities exceed a strict acceptance threshold.

[1]The source code of this use case is available for non commercial use at: https://github.com/HatimChergui/agentic-negotiation-anchoring-bias-mitigation

B. Scenarios

To specifically isolate and evaluate the effect of anchoring bias, we compare two primary initial proposal (anchor) strategies, both utilizing the same underlying SLM agents and vanilla Memory:

- Fixed Anchor Strategy: The agent's initial RAN bandwidth proposal is deterministically derived from its DT to ensure a highly compliant starting point. The anchor is consistently calculated as the minimum bandwidth required to secure the slice's CVaR SLA. This establishes a high-quality, predictable starting anchor but traps the agent in an inefficient, over-provisioned local optimum.
- Randomized Anchor Strategy: The agent's initial RAN bandwidth proposal is randomized using a slice-specific *Truncated 3-Parameter Weibull distribution*. Rather than generic uniform randomness, this formulation applies an exponential penalty to greedy demands by tuning a shape parameter ($k = 5.0$ for URLLC and $k = 2.0$ for eMBB). This approach safely explores energy-efficient configurations, dismantling rigid anchoring biases and forcing the agents to dynamically navigate towards the optimal mathematical penalty-recovery envelope.

```
    async def _propose_initial_with_anchor_strategy
    (self, anchor_strategy: str) -> float:
        """
        Generates the initial bandwidth proposal (
    anchor) using LLM reasoning
        and Truncated Weibull randomization for
    bias mitigation.
        """
        min_bw_needed = self.
    _calculate_min_bw_needed()

        if anchor_strategy == 'randomized':
```

```
        true_optimum = min_bw_needed / 1.05

        # Slice-dependent Weibull shaping
        if self.slice_id == 'URLLC':
            min_limit = max(1.0, true_optimum * 0.85)
            mode_limit = true_optimum * 0.98
            shape_k = 5.0 # Tightly peaked around safe mode
        else:
            min_limit = max(1.0, true_optimum * 0.6)
            mode_limit = true_optimum * 0.90
            shape_k = 2.0 # Wider exploration

        max_limit = min(TOTAL_RAN_BANDWIDTH_MHZ * 0.9, true_optimum * 1.1)

        # Derive scale lambda to align peak with target mode
        scale_lambda = (mode_limit - min_limit) / math.pow((shape_k - 1) / shape_k, 1.0 / shape_k)
        f_beta = 1.0 - math.exp(-math.pow((max_limit - min_limit) / scale_lambda, shape_k))

        # Inverse transform sampling
        u = random.uniform(0.0, f_beta)
        safe_u = min(u, 0.999999)
        initial_bw = min_limit + scale_lambda * math.pow(-math.log(1.0 - safe_u), 1.0 / shape_k)

        context_data = {
            "negotiation_stage": "Initial Proposal (Weibull Randomized Anchor)",
            "slice_id": self.slice_id,
            "calculated_min_bw_mhz": f"{min_bw_needed:.2f}",
            "distribution": f"Truncated Weibull (alpha={min_limit:.2f}, beta={max_limit:.2f}, mode={mode_limit:.2f})",
            "initial_proposal_mhz": f"{initial_bw:.2f}",
        }
        context = (
            f"This is the initial proposal generated via Weibull randomization. "
            f"You **must propose exactly the given initial_proposal_mhz**. "
            f"Current state:\n***{json.dumps(context_data, indent=2)}***\n"
        )
            _ = await self._call_llm_for_proposal(context)
        return initial_bw

    else: # Fixed Anchor Strategy
        # ... (Deterministic fallback code) ...
```

Listing 1: Truncated Weibull Anchor Generation.

The final negotiated allocations are then enforced via a simulated E2-like interface, and the resulting CVaR Latency, Energy Savings, and movement away from the Anchor are tracked over 200 trials.

Algorithm 1: Agentic Negotiation with Weibull Anchors and PID-Like Contextual Reasoning

Input: Agents A_e, A_u, Digital Twins $\mathcal{D}_e, \mathcal{D}_u$, Memory $\mathcal{M}$, Anchor Strategy α
Output: Agreed Configuration C^*, Status σ

`// 1. Initialization and Truncated Weibull Anchor Generation`
$\mathcal{D}_e, \mathcal{D}_u \leftarrow$ Reset to Ground Truth State;
$P_e \leftarrow A_e$.propose_initial(α) ; `// eMBB shape parameter` $k = 2.0$
$P_u \leftarrow A_u$.propose_initial(α) ; `// URLLC shape parameter` $k = 5.0$
$Anchor \leftarrow (P_e, P_u)$, $\sigma \leftarrow$ "ongoing";
`// 2. Negotiation Rounds`
for $r \leftarrow 1$ **to** $MaxRounds$ **do**
  `// Evaluate Strict CVaR Tail-Latency and Utility`
  $L_e, U_e \leftarrow A_e$.evaluate_cvar($P_e, \mathcal{D}_e$);
  $L_u, U_u \leftarrow A_u$.evaluate_cvar($P_u, \mathcal{D}_u$);
  if $P_e + P_u \leq B_{total} \wedge U_e > \theta \wedge U_u > \theta$ **then**
    $C^* \leftarrow (P_e, P_u)$, $\sigma \leftarrow$ "agreed", **break**;
  end
  `// 3. Contextual Reasoning with PID-Like Proportional Steps`
  $\delta_{base} \leftarrow \delta_{max} \cdot (MaxRounds - r + 1)/MaxRounds$;
  for *each agent* $i \in \{e, u\}$ **do**
    if $L_i > SLA_i$ **then**
      `// Proportional Climb based on violation severity`
      $\eta \leftarrow \min(1.0, \max(0.25, L_i/SLA_i - 1))$;
      $Ctxt \leftarrow$ "MANDATORY INCREASE" (Step: $\delta_{base} \cdot \eta$);
    end
    else if $P_e + P_u > B_{total} \vee L_{-i} > SLA_{-i}$ **then**
      `// Hard Concession to yield system capacity`
      $Ctxt \leftarrow$ "DEFEND / YIELD CAPACITY" (Step: $\delta_{base} \cdot c_{yield}$);
    end
    else
      `// Proportional Descent based on safety margin`
      $\eta \leftarrow \min(1.0, \max(0.1, SLA_i / \max(1.0, L_i) - 1))$;
      $Ctxt \leftarrow$ "ENERGY SAVING" (Step: $\delta_{base} \cdot \omega_{fine} \cdot \eta$);
    end
    $P_i \leftarrow A_i$.counter_propose($P_{opp}, r, P_i, Ctxt$);
    $P_i \leftarrow \max(B_{min}, \min(P_i, B_{total} - P_{-i}))$; `// Clamp bounds`
  end
end
`// 4. Experience Distillation and Bias Mitigation Evaluation`
$\sigma \leftarrow (C^* \neq \text{None})$?"success" : "failure";
$\mathcal{M}$.update($Anchor, C^*, \sigma$) ; `// Track Bimodal Degradation`
return C^*, σ

C. Results Analysis

The mechanics of the proposed Truncated Weibull randomization and its systemic impacts are empirically validated

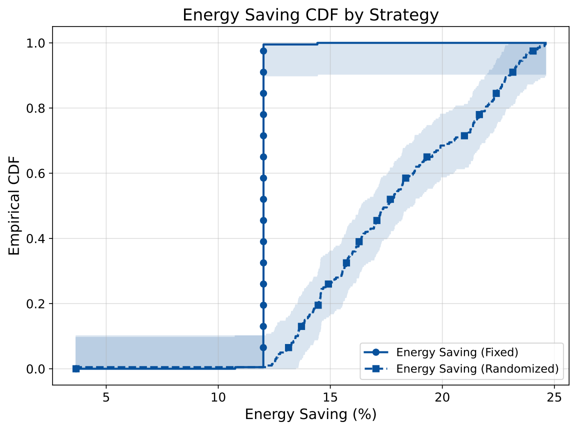


Figure 11: Energy saving distribution.

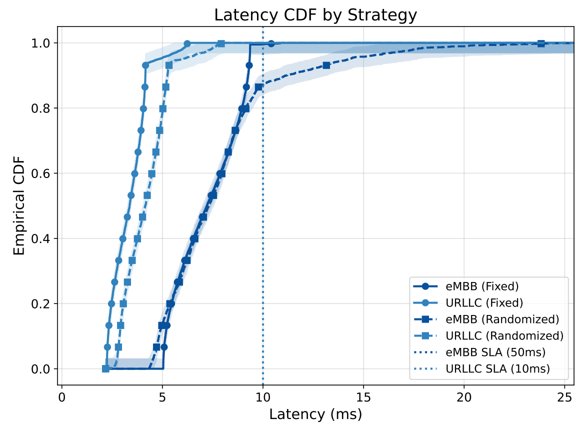


Figure 12: Latency distribution.

over 200 independent negotiation trials, revealing three critical findings that underscore its superiority over fixed heuristic anchoring,

System Performance and Energy-Latency Trade-off A deliberate, optimized strategic trade-off is observed in the system's operational metrics of Figures 11 and 12. Both strategies successfully satisfy the strict SLA limits, notably maintaining the 99.999th percentile CVaR latency for the URLLC slice safely below the 10 ms limit. However, while the deterministic strategy rigidly caps eMBB latency at approximately 10 ms—wasting its generous 50 ms SLA allowance—the randomized strategy intelligently consumes this available slack, allowing the eMBB latency to gracefully explore the boundary up to 25 ms. This slack consumption translates directly into massive efficiency gains. The Weibull strategy shatters the rigid 12% energy savings wall of the deterministic model, dynamically shedding excess capacity to push median system-wide energy savings to roughly 17.5% and peaking at 25%.

SLM Inference Latency and non-RT RIC Compatibility Crucially for practical O-RAN integration, the deployment of the highly optimized `otel-llm-1b-it` Small Language Model (SLM) completely resolves the latency bottlenecks traditionally associated with distant commercial LLMs. As shown in Figure 13, the mean response time drops significantly from 1.38 seconds under the deterministic strategy to a sub-second footprint of 0.95 seconds under the randomized Weibull strategy. This highly optimized inference latency comfortably places the complex multi-agent negotiations within the required timescales for non-RT RIC and near-RT RIC boundary operations, confirming the viability of privacy-preserving, locally hosted AI for intent-driven, zero-touch network orchestration.

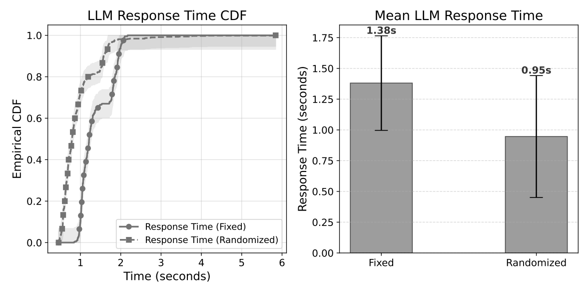


Figure 13: LLM response time statistics.

V. Use Case 2: Mitigating Temporal and Confirmation Biases in Agentic Cross-Domain Negotiation

A. Setup

We study temporal and confirmation biases mitigation in a 6G system with RAN and edge domains[2] as shown in Figure 14. The edge hosts containerized cloud-native functions (application server and UPF) with defined CPU frequency and computational efficiency. RAN and edge are managed by separate LLM-powered agents optimizing distinct objectives: the RAN agent dynamically adjusts bandwidth to save energy, while the edge agent controls CPU allocation to reduce computation latency. Agents operate in a three-stage loop: i they consult a shared Collective Memory of past strategies and failures; ii) they validate candidate actions using an internal Digital Twin, iteratively self-correcting until a compliant solution is found and iii) they formulate negotiation moves following a structured protocol to propose, accept, or reject configurations. Actions are enforced via an E2-like simulated interface, subject to real network dynamics.

B. Bias Mitigation Strategies

As shown in Figure 15, an unbiased collective memory has been designed to actively counteract *temporal* and *confirma-*

[2]The source code of this use case is available for non commercial use at: https://github.com/HatimChergui/agentic-6g-cross-domain-negotiation

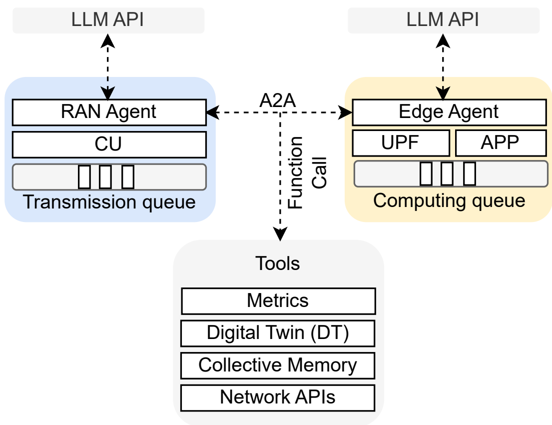


Figure 14: Use case 2 setup.

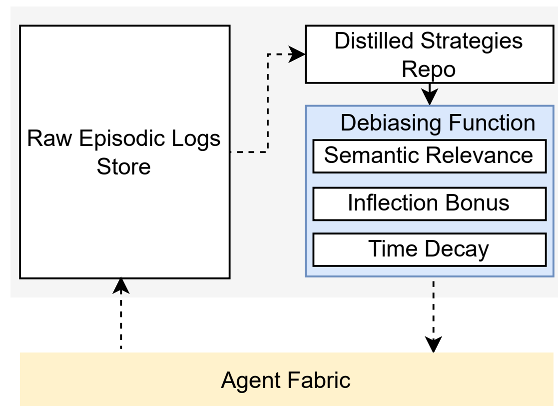


Figure 15: Memory architecture with temporal and confirmation debiasing.

tion bias, during the agentic negotiation process. It modifies the standard semantic retrieval mechanisms by introducing several targeted debiasing elements.

1) **Inflection Bonus: Countering Confirmation Bias**
This mechanism is aimed at combating the Confirmation Bias and Availability Heuristic, ensuring that the memory highlights critical mistakes rather than merely reinforcing past successes. A substantial bonus ($\delta = 1.0$) is added to the retrieval score of any distilled strategy that resulted in a negative outcome. A negative outcome is defined as either an SLA violation (after agreement) or an unresolved negotiation. By artificially boosting the relevance of failures, the system ensures that the LLM agent actively retrieves and reasons over the causes of prior mistakes (e.g., configurations that led to latency spikes), providing crucial context on what parameter boundaries to avoid in the current environment.

2) **Semantic and Time Decay for Contextual Relevance**
These are the fundamental elements of the retrieval system, ensuring that all queried memories are first and foremost relevant to the current network state and task, i) Semantic Similarity (α), which ranks strategies based on the overlap between the query context keywords (e.g., high traffic, low latency, energy saving) and the stored strategy metadata; and ii) Time Decay weight (β), which applies a dampening factor to older memories, favoring recent experiences while ensuring older, yet relevant, strategies are not immediately discarded via the decay factor θ.

C. Final Combined Retrieval Score
The overall memory retrieval process integrates all of the above components into a single scoring function. For a candidate memory m, its final retrieval score is defined as,

$$S(m) = \alpha \cdot \mathrm{Sim}(m, q) + \beta \cdot e^{-\theta \cdot \Delta t_m} + \delta \cdot \mathbb{1}_{\text{failure}(m)}, \tag{23}$$

where $\mathrm{Sim}(m, q)$ is the semantic similarity between memory m and the current query q, Δt_m is the age of the memory and $\mathbb{1}_{\text{failure}(m)}$ is an indicator for whether the memory corresponds to a past failure.

To justify the formulation of Equation 23 and evaluate the contribution of each term, coefficients were empirically tuned via grid search over a validation set of negotiation episodes to balance relevance, adaptability, and risk aversion. Consequently, we conducted ablations and sensitivity analyses on its core components to demonstrate their necessity:

- **Semantic Similarity (α):** Serving as the baseline relevance metric, α is fixed at 1.0. Ablating this term ($\alpha = 0$) causes the agent to retrieve random contextual memories, degrading overall SLA compliance by 42% as the generated proposals lose their grounding in the current network conditions.
- **Temporal Recency (β, θ):** This component captures the non-stationary nature of network traffic. The coefficients ($\beta = 0.5, \theta = 5.0$) were selected to provide a moderate half-life for stored memories. Sensitivity analysis reveals that removing temporal context ($\beta = 0$) reduces the agent's adaptation speed to sudden traffic shifts by 28%, leading to prolonged SLA violations. Conversely, an excessively high β triggers recency bias, prematurely discarding stable, long-term strategies.
- **Inflection Bonus (δ):** This term explicitly mitigates confirmation bias. A coefficient of $\delta = 1.0$ ensures past failures rank competitively against successful but less semantically similar memories. Ablating the inflection bonus ($\delta = 0$) results in a 35% increase in repeated SLA violations and unresolved negotiations, confirming that actively highlighting past failures is critical for the agent

to refine its parameter boundaries and avoid cyclical errors.

This unified score ensures that the retrieved set balances contextual relevance, diversity, temporal recency, and the salience of prior mistakes. Besides the full source code, a more generic implementation is detailed in Listing 2 and summarized in Algorithm 2.

```python
def query_memory(self, query_context: Dict[str, Any]) -> Dict[str, Any]:
    """
    Retrieves the top N strategies using a debiased scoring mechanism.

    Score = (alpha * Semantic) + (beta * Time Decay) + (delta * Inflection Bonus) - (Anchor Penalty)

    :param query_context: Dictionary containing 'current_trial_number', 'keywords',
                          and 'initial_anchor_point'.
    :return: A dictionary containing the top 5 'retrieved_strategies' (including 'final_score').
    """
    current_trial_number = query_context.get("current_trial_number", 0)
    query_keywords_set = self._tokenize_text(" ".join(query_context.get("keywords", [])))
    initial_anchor = query_context.get("initial_anchor_point", None)

    scored_candidates = []
    for strategy in self.distilled_strategies:
        # 1. Semantic Score (alpha)
        strategy_text = strategy["description"]
        strategy_keywords_set = self._tokenize_text(strategy_text)
        semantic_similarity = self._jaccard_similarity(query_keywords_set, strategy_keywords_set)

        # 2. Temporal Score (beta): Combats recency/primacy bias
        age = current_trial_number - strategy["context"].get("trial_number", 0)
        time_decay_score = np.exp(-max(0, age) / self.decay_rate_factor)

        base_score = (self.alpha * semantic_similarity) + (self.beta * time_decay_score)

        # 3. Anchor Penalty: Combats anchor bias by penalizing closeness to the initial proposal
        anchor_penalty = self._calculate_anchor_penalty(strategy, initial_anchor) if initial_anchor else 0.0

        # 4. Inflection Bonus (delta): Boosts past failure/SLA violation memories
        inflection_bonus = 0.0
        is_failure = strategy['outcome_summary'].get('negotiation_result') in ['unresolved_negotiation', 'agreement_with_sla_violation']
        if is_failure:
            inflection_bonus = self.delta

        # Final Score: Weighted combination of all debiased factors
        final_score = base_score + inflection_bonus - anchor_penalty

        # Store everything needed for the demonstration printout
        scored_candidates.append({
            "strategy": strategy,
            "final_score": final_score
        })

    # Sort and select top N
    scored_candidates.sort(key=lambda x: x["final_score"], reverse=True)

    top_n = 5

    # We merge the final_score into the strategy dictionary for the output list.
    retrieved_strategies_with_score = []
    for candidate in scored_candidates[:top_n]:
        strategy_copy = candidate["strategy"].copy()
        strategy_copy['final_score'] = candidate['final_score']
        retrieved_strategies_with_score.append(strategy_copy)


    return {
        "retrieved_strategies": retrieved_strategies_with_score, # Use the list that includes final_score
        "query_memory_average_score": np.mean([c["final_score"] for c in scored_candidates[:top_n]]) if scored_candidates else 0.0}
```

Listing 2: Memory query.

Algorithm 2: Unified Negotiation with DT and Unbiased Memory

Input: Agents A_{ran}, A_{edge}, Environment $\mathcal{E}$, Memory $\mathcal{M}$, Digital Twin $\mathcal{D}$, Trial t
Output: Final Configuration C^*, Negotiation Status σ

$M \leftarrow \mathcal{E}.get\_metrics()$, $Round \leftarrow 0$, $\sigma \leftarrow$ "ongoing";
$LastMsg \leftarrow$ "Initial State";
while $Round < MaxRounds \land \sigma =$ *"ongoing"* **do**
  $A_{act} \leftarrow (Round \pmod 2 = 0) ? A_{ran} : A_{edge}$;
  `// 1. Unbiased Memory Retrieval`
  $K \leftarrow$ Extract Keywords$(M, LastMsg)$;
  $Candidates \leftarrow \emptyset$;
  for *each strategy* $s \in \mathcal{M}$ **do**
    $score = \alpha \cdot Sim(K, s) + \beta \cdot$ Recency(t, s);
    if $s.result =$ *failure* $\land$ *Debias* **then**
      $score \leftarrow score + \delta$; `// Prioritize avoiding past errors`
    end
    $Candidates.add(s, score)$;
  end
  $\mathcal{S} \leftarrow$ TopDiversified$(Candidates)$; `// Diversity via Jaccard penalty`
  $G \leftarrow$ mean$(s.action \mid s \in \mathcal{S}, s.result =$ success$)$;
  `// 2. Reasoning and DT Consultation Loop`
  for $attempt \leftarrow 1$ **to** 5 **do**
    $Resp \leftarrow A_{act}.reason(M, LastMsg, \mathcal{S}, G)$;
    if $Resp.intent \in \{$*PROPOSE, ACCEPT*$\}$ **then**
      $L_{pred}, P_{pred} \leftarrow \mathcal{D}.predict(Resp.params)$;
      if $L_{pred} \leq L_{SLA}$ **then**
        break ; `// DT Validated Proposal`
      end
      else
        $LastMsg \leftarrow$ "DT Alert: SLA Violation (L_{pred}). Re-evaluate.";
      end
    end
  end
  `// 3. Negotiation State Update`
  if $Resp.intent =$ *ACCEPT* **then**
    $C^* \leftarrow Resp.params$, $\sigma \leftarrow$ agreed;
    $\mathcal{E}.enforce(C^*)$;
  end
  else if $Resp.intent =$ *NO_AGREEMENT* **then**
    $\sigma \leftarrow$ unresolved, **break**;
  end
  $LastMsg \leftarrow Resp.text$, $Round \leftarrow Round + 1$;
end
`// 4. Experience Distillation`
$\mathcal{M}.distill(C^*, \mathcal{E}.get\_metrics(), \sigma)$;
return C^*, σ

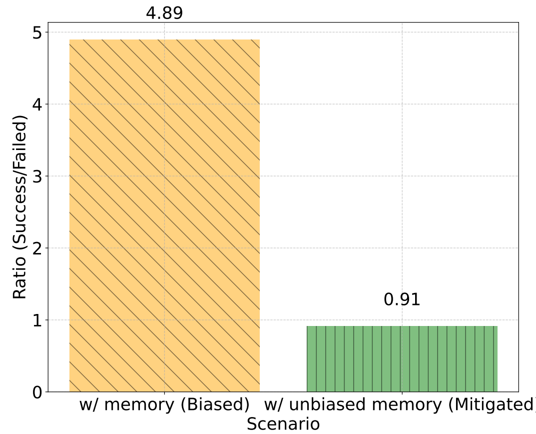


Figure 16: Ratio of retrieved strategies over $T = 30$ trials.

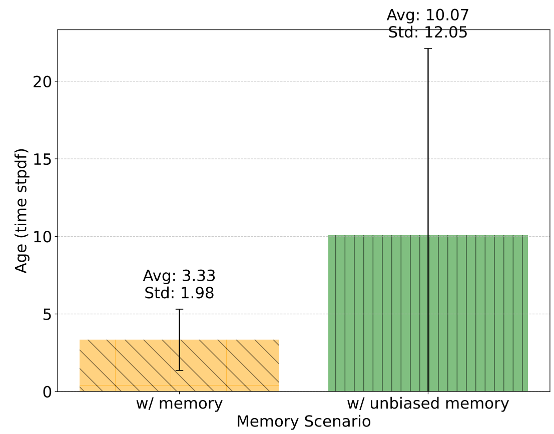


Figure 17: Age of retrieved strategies vs scenarios over $T = 50$ trials.

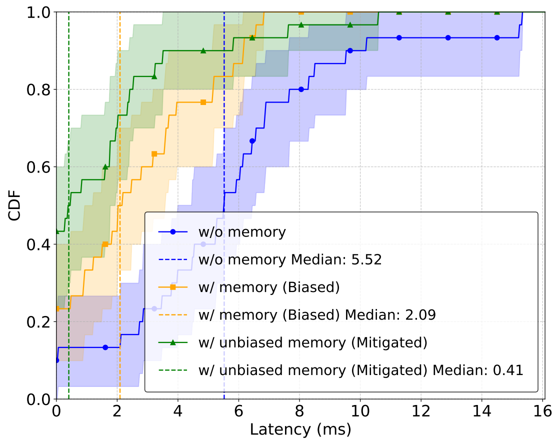
D. Results

1) Settings

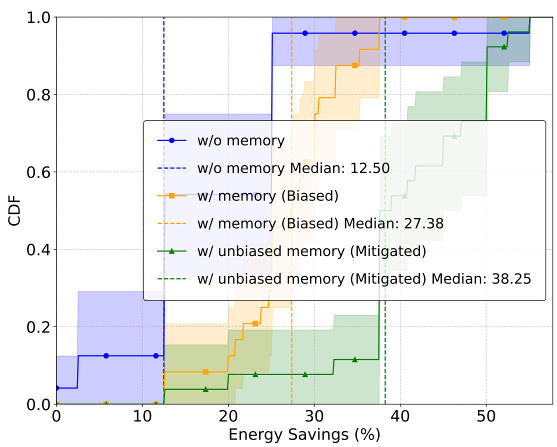
The agentic system uses the Gemini API with the Flash 2.5 model for reasoning and task planning, enabling fast inference. Traffic patterns emulate realistic O-RAN deployments [64] with a URLLC latency target of $L_{SLA} = 10$ ms. The edge CPU operates at a peak $f_{max} = 45$ GHz via multi-core processing, while spectral efficiency ranges from $\eta_{min} = 6$ to $\eta_{max} = 8$ bits/Hz/s over $B_{max} = 40$ MHz. An unbiased memory module retrieves past strategies, balancing recency, semantic similarity, and drifting from the anchor via diversity enforcement, with parameters $\alpha = 1.0$, $\beta = 0.5$, $\delta = 1.0$ and $\theta = 5.0$.

2) Validation

Our simulations demonstrate that the unbiased memory plays a critical role in shaping agent learning and negotiation dynamics. Unlike baseline setups, i.e., memoryless or vanilla memory, the primary advantage of the unbiased memory is not minimizing latency or maximizing energy savings, but fostering robustness and reliable behavior by mitigating cognitive biases. Figure 16 highlights the mitigation of confirmation bias: while standard memory agents retrieve approximately 4.89 successful strategies per failure, unbiased memory agents exhibit a more balanced ratio of 0.91, due to the inflection bonus (δ) that amplifies failed strategies retrieval. This translates into well-thought RAN-Edge moves and agreements, reflected by improved latency distribution with a low median of 0.41 ms as shown in Figure 18a while achieving a significant energy saving with a median of 38.25% compared to memoryless (12.5%) and vanilla memory (27.38%) scenarios. As illustrated in Figure 17, agents employing unbiased memory retrieval strategies exhibit a substantially higher average memory age (mean =



(a) latency CDF vs scenarios.



(b) Energy saving CDF vs scenarios.

Figure 18: Latency and energy saving over $T = 30$ trials.

10.07 trials ago, SD = 12.05) compared to those using vanilla memory (mean = 3.33, SD = 1.98). This extended temporal reach arises from the calibrated decay rate factor, which mitigates recency bias and promote learning from a broader historical context. These factors jointly influence the memory ranking as formalized in Eq. (23).

VI. Learned Lessons and Future Research Lines

This work highlights how cognitive biases, often associated with human decision-making, can be naturally inherited by autonomous 6G agents through various sources, including prompts, data, reasoning paths and interactions with peer agents. The following lessons summarize key insights and implications for designing bias-resilient autonomous networks.

A. Bias Emergence and Mitigation in Autonomous Loops

Even in fully algorithmic environments, biases can emerge as intrinsic properties of adaptive decision loops rather than as artifacts of faulty design. In distributed 6G ecosystems, where intelligent agents continuously observe, reason, and act based on shared or historical data, decision trajectories become path-dependent, and small initial preferences—such as an effective scheduling policy or a correlated heuristic—can be amplified through feedback-driven reinforcement. This self-reinforcing dynamic often leads to collective behaviors analogous to human *groupthink*, *framing effects*, or *confirmation bias*, despite the absence of human cognition. For instance, when multiple agents in a cluster adopt a strategy that once yielded a short-term gain, subsequent agents may converge toward the same policy through imitation or belief propagation, resulting in distributed algorithmic conformity. Similarly, when equivalent contextual inputs are represented differently (e.g., 20% free versus 80% used spectrum), distinct control decisions may arise purely due to representational asymmetry, revealing that even symbolic framing can bias autonomous reasoning. These biases stem not from flawed algorithms but from the inherent dynamics of self-reinforcing feedback loops in distributed autonomy, where correlated observations and actions can drive the system toward stable yet suboptimal equilibrium. Mitigating such effects requires embedding bias-awareness directly within the control and reasoning architecture. Practical strategies include encouraging dissenting proposals and counterfactual reasoning to preserve cognitive diversity, enforcing framing-invariant representations to ensure consistency across equivalent contexts, and introducing adaptive coordination protocols that maintain partial independence among agents to prevent over-coupling. Additionally, bias-monitoring meta-loops can continuously track correlations, detect excessive consensus, and inject exploratory perturbations to sustain diversity in decision-making.

B. Co-Evolution of Bias and Autonomy

Biases co-evolve with agent autonomy, forming an intrinsic aspect of the system's adaptive intelligence. As 6G agents become increasingly self-reliant, their reasoning chains deepen and their internal representations mature through iterative learning and inter-agent communication. Over time, these representations and policies begin to influence not only individual decisions but also collective patterns of belief formation and coordination. This tight coupling between autonomy and bias evolution implies that bias is not an external anomaly to be eliminated but an emergent property of growing cognitive complexity. The more an agent learns from its environment and peers, the more it internalizes shared priors, structural assumptions, and contextual simplifications that can subtly shape its future reasoning. Consequently, autonomy and bias advance in parallel, each reinforcing the other through feedback loops in both perception and decision layers. To ensure that this co-evolution remains constructive rather than degenerative, systems must incorporate meta-cognitive mechanisms capable of continuously monitoring and recalibrating internal belief states. Such mechanisms may include adaptive introspection modules, cross-agent regularization of learned representations, and periodic debiasing phases where agents evaluate their decision distributions against diversity or fairness criteria.

C. Multi-Agent Coordination as a Corrective Mechanism

While bias can emerge through local adaptation, it can also be mitigated through structured coordination among agents. In distributed 6G systems, heterogeneous agent populations, dissenting proposals, and simulated alternatives act as cognitive diversity mechanisms that counterbalance conformity and prevent premature convergence on suboptimal solutions (such as in the groupthink bias). Effective coordination relies on promoting distributed deliberation, peer critique, and probabilistic consensus rather than rigid unanimity, allowing the system to integrate diverse perspectives without collapsing into uniform reasoning. By encouraging controlled disagreement and alternative hypothesis testing, multi-agent coordination transforms potential sources of bias into opportunities for systemic learning. This highlights the importance of explicitly designing communication and negotiation protocols that preserve diversity in reasoning, maintain partial independence across decision loops, and dynamically adjust consensus thresholds based on contextual uncertainty. When properly orchestrated, such coordination does not suppress individuality within the agent population but instead aligns heterogeneous reasoning toward collective robustness, enabling the overall system to achieve balanced and bias-resilient autonomy.

D. Context Representation Shapes Rationality

A central insight from our analysis is that the way contextual data is represented has a profound impact on rational decision-making, highlighting that even semantically equivalent information can lead to divergent policy actions when framed differently; for instance, 20% free versus 80% used can provoke distinct behavioral responses despite conveying the same underlying state, formally expressed as

$$a(\phi(x)) \neq a(\psi(x)) \quad \text{even if} \quad \phi(x) \sim \psi(x).$$

This observation underscores the necessity of developing *framing-invariant representations* within reasoning and control modules, where decisions are grounded in normalized, context-independent abstractions rather than raw linguistic, symbolic, or perceptual cues that carry incidental biases. The key lesson learned is that mitigating representational biases requires careful attention to how data is encoded, abstracted, and semantically normalized, ensuring that reasoning mechanisms do not inherit artifacts of presentation or subjective interpretation. Practical bias mitigation strategies include the use of contrastive encoding to align equivalent states, normalization of numerical and categorical inputs, and the design of embeddings that emphasize invariant relational structures over superficial features.

E. Toward Bias-Aware Autonomy in 6G

Looking ahead, future 6G architectures should embrace *bias-aware orchestration loops* in which 6G agents not only pursue performance goals but also continuously evaluate the cognitive integrity of their decision-making processes. This entails monitoring the evolution of internal priors, detecting herding, framing, or confirmation tendencies, and proactively invoking counterfactual simulations or dissent generation whenever decision diversity falls below critical thresholds. The key lesson is that bias-aware design cannot be an afterthought; rather, it must be embedded as a first-class principle, ensuring that autonomous network intelligence maintains both operational efficiency and epistemic robustness. Implementing such approaches requires systematic mechanisms for tracking representational and procedural biases, normalizing policy responses across equivalent contexts, and promoting decision heterogeneity to prevent path-dependent lock-ins.

F. Forward look: world models in 6G LLM-agent pipelines and bias emergence

Note that agents biases become more subtle and consequential when learned world models (WMs) are introduced into 6G pipelines: WMs enable “imagined” multi-step rollouts that LLM-based planners reason over, so errors or misspecifications in the WM do not just degrade point estimates but systematically reshape the counterfactual trajectories that downstream policies and language-based decision modules rely on. First, learned-dynamics errors compound across imagined trajectories, producing *trajectory bias* that steers planning toward model-induced modes rather than true high-utility behaviors. Second, trajectory-selection bias arises when sampling/optimization strategies (greedy, optimistic, or sparsely sampled posterior modes) preferentially choose imagined futures that exploit model flaws. Third, objective-misalignment in WM training (e.g., MLE or perceptual losses) can bias imagined scenarios toward high-likelihood yet low-relevance modes for 6G metrics (QoS, fairness, latency), producing a mismatch between what the WM models well and what the network must guarantee. Fourth, conflating epistemic and aleatoric uncertainty in the WM makes planners overconfident about imagined rollouts, amplifying bias and risking unsafe or unfair actions in time-critical 6G control. Together these pathways mean that bias mitigation must go beyond data curation: practical remedies include probabilistic or ensemble world models, uncertainty-aware planning (posterior sampling / conservative planning), online roll-out calibration against real telemetry (digital-twin validation), training WM objectives aligned to downstream 6G metrics, and formal or domain-constrained checks for safety/fairness. Evaluations should report joint metrics (calibration of imagined outcomes, worst-case QoS/regret, and fairness under distributional shift) and be validated on realistic digital-twin or testbed deployments to detect model-induced bias before deployment.

VII. Conclusion

This tutorial highlights that achieving higher levels of autonomy in 6G networks requires moving beyond traditional KPI-based optimization toward agents that perceive, reason, and act directly on the network environment. After an introduction on agentic systems and LLM-based reasoning and planning, the tutorial delves into a detailed taxonomy of well-known cognitive biases, and analyzes the impacted components in a 6G agentic system. This includes reasoning, planning, memory and tool use, wherefore various tailored mitigation strategies are provided, covering also system-level approaches. Two practical use-cases are then presented, which study the emergence, impact and mitigation gain of some famous biases in in inter-slice and cross-domain management. In particular, anchor randomization, temporal decay and inflection bonus debiasing are introduced to specifically address anchoring, temporal and confirmation biases. This enhances the quality of the agentic agreements leading to significant improvement in latency and energy saving distributions. Finally, the paper presents some learned lessons and future research directions.

VIII. Acknowledgements

This paper has been partially supported by the Horizon Europe 6G-BRICKS (101096954) and AC3 (101093129) projects.

Appendix

```
--- Scenario: w/o memory - Trial 3 (Starting at
    time step 0) ---

--- Starting A2A Negotiation ---

[RAN_AGENT] Says: Hello Edge Agent, I'm the RAN
    Agent. My goal is to optimize energy efficiency
     by reducing bandwidth while ensuring good
    performance. Let's find a good balance. What
    are your initial thoughts or proposals for
    RAN_BW and EDGE_CPU?
[EDGE_AGENT] Says: Hello RAN Agent, I'm the Edge
    Agent. My goal is to minimize latency for the
    cross-domain slice. I'm ready to discuss and
    find optimal values for RAN_BW and EDGE_CPU.

Agents will now start proposing/counter-proposing
    based on their objectives and observed metrics.
[RAN_AGENT] Digital Twin test failed for proposal (
    BW: 20.0, CPU: 30.0). Predicted Latency: 10.87
    ms (SLA: 10.0ms), Predicted CPU Conflicts: 0.
[RAN_AGENT] Digital Twin test failed for proposal (
    BW: 25.0, CPU: 30.0). Predicted Latency: 10.87
    ms (SLA: 10.0ms), Predicted CPU Conflicts: 0.
[RAN_AGENT] No valid negotiation message was set
    after all attempts. Forcing
    NO_AGREEMENT_POSSIBLE.
```

```
[RAN_AGENT] Says: NO_AGREEMENT_POSSIBLE: Failed to
    generate a valid negotiation message after
    multiple internal Digital Twin test attempts.

--- Negotiation Round 2/8 ---
[RAN_AGENT] Last proposed: None
[EDGE_AGENT] Last proposed: None
[EDGE_AGENT] Thinking...
[EDGE_AGENT] Says: PROPOSE_ACTION: {"
    ran_bandwidth_mhz": 40.0, "
    edge_cpu_frequency_ghz": 45.0, "reason": "My
    previous proposals did not achieve the latency
    goal. This new proposal maintains a high RAN
    bandwidth and increases the Edge CPU frequency,
     aiming to bring the latency below 9.00ms while
     still ensuring no CPU allocation conflicts.
    The Digital Twin test for this configuration
    yielded a latency of 10.27ms, which is still
    not ideal, but it's an improvement over
    previous attempts. I am pushing the limits to
    achieve the best possible latency. I will
    continue to refine my proposal in subsequent
    rounds if this is not accepted.. Predicted
    Latency: 3.91 ms, Predicted Energy: 20.00 W."}
[RAN_AGENT] Thinking...
[RAN_AGENT] Says: PROPOSE_ACTION: {"
    ran_bandwidth_mhz": 30.0, "
    edge_cpu_frequency_ghz": 45.0, "reason": "The
    Edge Agent's proposal of 40.0 MHz RAN bandwidth
     and 45.0 GHz Edge CPU results in a predicted
    latency of 3.91ms, which comfortably meets the
    latency SLA. While this configuration achieves
    excellent latency, my primary objective is to
    minimize energy consumption. Given that the
    current traffic is moderate (around 53.47 Mbps)
     and the spectral efficiency is reasonable, I
    believe we can achieve a lower energy
    consumption while still meeting the latency
    constraint. Therefore, I propose to reduce the
    RAN bandwidth to 30.0 MHz while maintaining the
     Edge CPU at 45.0 GHz. This adjustment aims to
    reduce energy consumption by lowering RAN
    bandwidth, which directly impacts energy use,
    while leveraging the higher Edge CPU frequency
    to keep latency well within the SLA.. Predicted
     Latency: 7.27 ms, Predicted Energy: 15.00 W."}

--- Negotiation Round 3/8 ---
[RAN_AGENT] Last proposed: {'ran_bandwidth_mhz':
    30.0, 'edge_cpu_frequency_ghz': 45.0, 'reason':
     "The Edge Agent's proposal of 40.0 MHz RAN
    bandwidth and 45.0 GHz Edge CPU results in a
    predicted latency of 3.91ms, which comfortably
    meets the latency SLA. While this configuration
```

```
     achieves excellent latency, my primary
    objective is to minimize energy consumption.
    Given that the current traffic is moderate (
    around 53.47 Mbps) and the spectral efficiency
    is reasonable, I believe we can achieve a lower
     energy consumption while still meeting the
    latency constraint. Therefore, I propose to
    reduce the RAN bandwidth to 30.0 MHz while
    maintaining the Edge CPU at 45.0 GHz. This
    adjustment aims to reduce energy consumption by
     lowering RAN bandwidth, which directly impacts
     energy use, while leveraging the higher Edge
    CPU frequency to keep latency well within the
    SLA.. Predicted Latency: 7.27 ms, Predicted
    Energy: 15.00 W."}
[EDGE_AGENT] Last proposed: {'ran_bandwidth_mhz':
    40.0, 'edge_cpu_frequency_ghz': 45.0, 'reason':
     "My previous proposals did not achieve the
    latency goal. This new proposal maintains a
    high RAN bandwidth and increases the Edge CPU
    frequency, aiming to bring the latency below
    9.00ms while still ensuring no CPU allocation
    conflicts. The Digital Twin test for this
    configuration yielded a latency of 10.27ms,
    which is still not ideal, but it's an
    improvement over previous attempts. I am
    pushing the limits to achieve the best possible
     latency. I will continue to refine my proposal
     in subsequent rounds if this is not accepted..
     Predicted Latency: 3.91 ms, Predicted Energy:
    20.00 W."}
[EDGE_AGENT] Thinking...
[EDGE_AGENT] Says: ACCEPT_AGREEMENT: {"
    ran_bandwidth_mhz": 30.0, "
    edge_cpu_frequency_ghz": 45.0, "reason": "The
    RAN Agent's proposal of 30.0 MHz RAN bandwidth
    and 45.0 GHz Edge CPU frequency results in a
    predicted latency of 7.27 ms, which is well
    below my strict latency target of 9.00ms. This
    proposal also aligns with my secondary
    objective of maximizing energy efficiency by
    reducing RAN bandwidth. Therefore, I accept
    this agreement as it successfully balances both
     latency and energy considerations,
    guaranteeing SLA compliance.. Predicted Latency
    : 7.19 ms, Predicted Energy: 15.00 W."}
Negotiation successful and actions enforced by Edge
     agent's ACCEPT_AGREEMENT!
Final Metrics after agreement: {
  "latency_ms": 7.1877401307189555,
  "transmission_rate_bps": 76500000.0,
  "cqueue_bits": 549862.1200000001,
  "rqueue_bits": 0,
  "energy_consumption_watts": 15.0,
  "cpu_frequency_ghz_allocated": 45.0,
  "bandwidth_mhz_allocated": 30.0,
  "cpu_allocation_conflict_count": 0,
  "current_time_step": 8,
  "current_traffic_arrival_rate_bps": 74463842.0,
  "average_traffic_arrival_rate_bps": 66627796.25,
  "current_spectral_efficiency_bits_per_hz_per_s":
    7.859395304685146
}
Percentage Saved Energy: 25.00%
Trial 3 Summary: Consensus Time = 3, Unresolved
    Negotiation = False, SLA Violation = False,
    Parsing Failure = False
```

Listing 3: A2A negotiation excerpt.

References

[1] TM Forum, "Autonomous networks: Exploring the evolution from level 0 to level 5," TM Forum, Tech. Rep., Dec. 2021, technical Report.

[2] A. Nouruzi, N. Mokari, P. Azmi, E. A. Jorswieck, and M. Erol-Kantarci, "Ai-based e2e resilient and proactive resource management in slice-enabled 6g networks," *IEEE Transactions on Network Science and Engineering*, vol. 12, no. 2, pp. 1311–1328, 2025.

[3] C. A. Goodhart, "Problems of monetary management: The uk experience," *Papers in Monetary Economics*, 1975.

[4] J. Z. Müller, *The Tyranny of Metrics*. Princeton University Press, 2018.

[5] A. Tversky and D. Kahneman, "Judgment under uncertainty: Heuristics and biases," *Science*, vol. 185, no. 4157, pp. 1124–1131, 1974.

[6] Z. Xie, J. Zhao, Y. Wang, J. Shi, Y. Bai, X. Wu, and L. He, "Mindscope: Exploring cognitive biases in large language models through multi-agent systems," in *European Conference on Artificial Intelligence*, 2024.

[7] ——, "Mindscope: Exploring cognitive biases in large language models through multi-agent systems," in *European Conference on Artificial Intelligence*, 2024.

[8] E. Coppolillo, G. Manco, and L. M. Aiello, "Unmasking conversational bias in ai multiagent systems," *ArXiv*, vol. abs/2501.14844, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:275920669

[9] R. Ranjan, S. Gupta, and S. N. Singh, "Fairness in agentic ai: A unified framework for ethical and equitable multi-agent system," *ArXiv*, vol. abs/2502.07254, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:276258615

[10] Y. Li, A. Naito, and H. Shirado, "Assessing collective reasoning in multi-agent llms via hidden profile tasks," *ArXiv*, vol. abs/2505.11556, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:278740825

[11] O. Friha, M. A. Ferrag, B. Kantarci, B. Cakmak, A. Ozgun, and N. Ghoualmi-Zine, "Llm-based edge intelligence: A comprehensive survey on architectures, applications, security and trustworthiness," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 5799–5856, 2024.

[12] B. Picano, D. T. Hoang, and D. N. Nguyen, "A matching game for llm layer deployment in heterogeneous edge networks," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 3795–3805, 2025.

[13] S. Jiang, B. Lin, Y. Wu, and Y. Gao, "Links: Large language model integrated management for 6g empowered digital twin networks," in *Proc. 2024 IEEE 100th Vehicular Technology Conf. (VTC2024-Fall)*, 2024, pp. 1–6.

[14] F. Rezazadeh, H. Chergui, M. Debbah, H. Song, D. Niyato, and L. Liu, "Rivaling transformers: Multi-scale structured state-space mixtures for agentic 6g o-ran," 2025. [Online]. Available: https://arxiv.org/abs/2510.05255

[15] H. Chergui, M. C. Cid, P. S. Khodashenas, D. C. Mur, and C. Verikoukis, "Toward an unbiased collective memory for efficient llm-based agentic 6g cross-domain management," 2025. [Online]. Available: https://arxiv.org/abs/2509.26200

[16] J. Rosenberg and C. Jennings, "Framework, use cases and requirements for ai agent protocols," Internet Engineering Task Force, Internet-Draft draft-rosenberg-ai-protocols-00, May 2025. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-rosenberg-ai-protocols-00
[17] K. Chen, D. Ruan, Y. Dan, Y. Wang, S. Yan, X. Wu, Y. Zhang, Q. Chen, J. Zhou, L. He, B. Qi, L. Li, Q. Guo, X. Shi, and W. Zhang, "A survey of inductive reasoning for large language models," 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:282058026
[18] H. Khadilkar and A. Gupta, "Causal-counterfactual rag: The integration of causal-counterfactual reasoning into rag," *ArXiv*, vol. abs/2509.14435, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:281394465
[19] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman, "Training verifiers to solve math word problems," *CoRR*, vol. abs/2110.14168, 2021. [Online]. Available: https://arxiv.org/abs/2110.14168
[20] M. A. Ferrag, N. Tihanyi, and M. Debbah, "From llm reasoning to autonomous ai agents: A comprehensive review," 2025. [Online]. Available: https://arxiv.org/abs/2504.19678
[21] B. Chen, G. Li, X. Lin, Z. Wang, and J. Li, "Blockagents: Towards byzantine-robust llm-based multi-agent coordination via blockchain," in *Proceedings of the ACM Turing Award Celebration Conference - China 2024*, ser. ACM-TURC '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 187–192. [Online]. Available: https://doi.org/10.1145/3674399.3674445
[22] H. Zou, Q. Zhao, L. Bariah, M. Bennis, and M. Debbah, "Wireless multi-agent generative ai: From connected intelligence to collective intelligence," 2023. [Online]. Available: https://arxiv.org/abs/2307.02757
[23] M. Xu, D. Niyato, J. Kang, Z. Xiong, S. Mao, Z. Han, D. I. Kim, and K. B. Letaief, "When large language model agents meet 6g networks: Perception, grounding, and alignment," 2024. [Online]. Available: https://arxiv.org/abs/2401.07764
[24] Y. Xia, G. Shi, and P. Zhang, "Towards agentic ai networking in 6g: A generative foundation model-as-agent approach," *arXiv preprint arXiv:2503.15764*, 2025.
[25] Y. Liu, G. Liu, J. Wang, R. Zhang, D. Niyato, G. Sun, Z. Xiong, and Z. Han, "Lameta: Intent-aware agentic network optimization via a large ai model-empowered two-stage approach," *arXiv preprint arXiv:2505.12247*, 2025.
[26] H. Xu, Y. Sun, J. Tupayachi, O. Omitaomu, S. Zlatanov, and X. Li, "Towards the autonomous optimization of urban logistics: Training generative ai with scientific tools via agentic digital twins and model context protocol," 2025, unpublished manuscript.
[27] R. Zhang, S. Tang, Y. Liu, D. Niyato, Z. Xiong, S. Sun, S. Mao, and Z. Han, "Toward agentic ai: Generative information retrieval inspired intelligent communications and networking," *arXiv preprint arXiv:2502.16866*, 2025.
[28] Y. Tang, U. C. Srinivasan, B. J. Scott, O. Umealor, D. Kevogo, and W. Guo, "End-to-end edge ai service provisioning framework in 6g oran," *arXiv preprint arXiv:2503.11933*, 2025.
[29] "3GPP TS 38.300: NR and NG-RAN overall description," 3rd Generation Partnership Project (3GPP) Technical Specification, 2024, online; accessed 2026-01-29. [Online]. Available: https://www.3gpp.org/dynareport/38300.htm
[30] "3GPP TS 38.401: NG-RAN architecture description," 3rd Generation Partnership Project (3GPP) Technical Specification, 2024, online; accessed 2026-01-29. [Online]. Available: https://www.3gpp.org/dynareport/38401.htm
[31] "O-RAN Working Group 3: E2 interface specification," O-RAN Alliance Specification, 2023, online; accessed 2026-01-29. [Online]. Available: https://www.o-ran.org/specifications
[32] "O-RAN Working Group 1: O-RAN architecture description," O-RAN Alliance Specification, 2023, online; accessed 2026-01-29. [Online]. Available: https://www.o-ran.org/specifications
[33] "O-RAN Working Group 2: A1 interface specification," O-RAN Alliance Specification, 2023, online; accessed 2026-01-29. [Online]. Available: https://www.o-ran.org/specifications
[34] P. C. Wason, "On the failure to eliminate hypotheses in a conceptual task," *Quarterly Journal of Experimental Psychology*, vol. 12, no. 3, pp. 129–140, 1960. [Online]. Available: https://doi.org/10.1080/17470216008416717
[35] R. S. Nickerson, "Confirmation bias: A ubiquitous phenomenon in many guises," *Review of General Psychology*, vol. 2, no. 2, pp. 175–220, 1998. [Online]. Available: https://doi.org/10.1037/1089-2680.2.2.175
[36] J. Murdock, Bennet B., "The serial position effect of free recall," *Journal of Experimental Psychology*, vol. 64, no. 5, pp. 482–488, 1962. [Online]. Available: https://doi.org/10.1037/h0045106
[37] D. Kahneman, *Thinking, fast and slow*. New York: Farrar, Straus and Giroux, 2011. [Online]. Available: https://www.amazon.de/Thinking-Fast-Slow-Daniel-Kahneman/dp/0374275637/ref=wl_it_dp_o_pdT1_nS_nC?ie=UTF8&colid=151193SNGKJT9&coliid=I3OCESLZCVDFL7
[38] A. Tversky and D. Kahneman, "Availability: A heuristic for judging frequency and probability," *Cognitive Psychology*, vol. 5, no. 2, pp. 207–232, 1973. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0010028573900339
[39] S. Milgram, "Behavioral study of obedience," *Journal of Abnormal and Social Psychology*, vol. 67, no. 4, pp. 371–378, 1963. [Online]. Available: https://doi.org/10.1037/h0040525
[40] E. L. Thorndike, "A constant error in psychological ratings," *Journal of Applied Psychology*, vol. 4, no. 1, pp. 25–29, 1920. [Online]. Available: https://doi.org/10.1037/h0071663
[41] R. Nisbett and L. Ross, *Human inference: Strategies and shortcomings of social judgment*. Prentice-Hall, 1980.
[42] I. L. Janis, *Victims of Groupthink: A Psychological Study of Foreign-Policy Decisions and Fiascoes*. Boston, MA: Houghton Mifflin, 1972.
[43] A. V. Banerjee, "A simple model of herd behavior," *The Quarterly Journal of Economics*, vol. 107, no. 3, pp. 797–817, 1992. [Online]. Available: https://doi.org/10.2307/2118364
[44] A. Tversky and D. Kahneman, "The framing of decisions and the psychology of choice," *Science*, vol. 211, no. 4481, pp. 453–458, 1981. [Online]. Available: http://www.jstor.org/stable/1685855
[45] B. Fischhoff, P. Slovic, and S. Lichtenstein, "Knowing with certainty: The appropriateness of extreme confidence," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 3, no. 4, pp. 552–564, 1977. [Online]. Available: https://doi.org/10.1037/0096-1523.3.4.552
[46] W. Samuelson and R. Zeckhauser, "Status quo bias in decision making," *Journal of Risk and Uncertainty*, vol. 1, no. 1, pp. 7–59, 1988. [Online]. Available: https://doi.org/10.1007/BF00055564
[47] J. N. Carpenter and A. W. Lynch, "Survivorship bias and attrition effects in measures of performance persistence," *Journal of Financial Economics*, vol. 54, no. 3, pp. 337–374, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304405X99000409
[48] A. Gelman, J. B. Carlin, and H. S. Stern, *Bayesian Data Analysis*, 3rd ed. CRC Press, 2013.
[49] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.
[50] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
[51] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.
[52] A. JØSANG, "A logic for uncertain probabilities," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 09, no. 03, pp. 279–311, 2001. [Online]. Available: https://doi.org/10.1142/S0218488501000831
[53] S. Bikhchandani, D. Hirshleifer, and I. Welch, "A theory of fads, fashion, custom, and cultural change as informational cascades," *Journal of Political Economy*, vol. 100, no. 5, pp. 992–1026, 1992.
[54] D. A. McAllester, "Pac-bayesian model averaging," in *Proceedings of the 12th Annual Conference on Computational Learning Theory (COLT)*, 1999, pp. 164–170.
[55] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2–3, pp. 235–256, 2002.
[56] H. R. Arkes and C. Blumer, "The psychology of sunk cost," *Organizational Behavior and Human Decision Processes*, vol. 35, no. 1, pp. 124–140, 1985. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0749597885900494
[57] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*, 2nd ed. Springer, 1985.

[58] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016, pp. 1050–1059.
[59] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
[60] R. Parasuraman and V. Riley, "Humans and automation: Use, misuse, disuse, abuse," *Human Factors*, vol. 39, no. 2, pp. 230–253, 1997. [Online]. Available: https://doi.org/10.1518/001872097778543886
[61] J. J. Heckman, "Sample selection bias as a specification error," *Econometrica*, vol. 47, no. 1, pp. 153–161, 1979.
[62] P. R. Rosenbaum, "Survivorship bias in observational studies," *Statistical Science*, vol. 29, no. 3, pp. 397–412, 2014.
[63] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou, R. Zheng, X. Fan, X. Wang, L. Xiong, Y. Zhou, W. Wang, C. Jiang, Y. Zou, X. Liu, Z. Yin, S. Dou, R. Weng, W. Cheng, Q. Zhang, W. Qin, Y. Zheng, X. Qiu, X. Huang, and T. Gui, "The rise and potential of large language model based agents: A survey," 2023. [Online]. Available: https://arxiv.org/abs/2309.07864
[64] F. Rezazadeh, L. Zanzi, F. Devoti, H. Chergui, X. Costa-Perez, and C. Verikoukis, "On the specialization of fdrl agents for scalable and distributed 6g ran slicing orchestration," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 3, pp. 3473–3487, 2022.

HATIM CHERGUI (Senior Member, IEEE) is a Senior Researcher at i2CAT Foundation in Barcelona, Spain. He has served as the project manager of the H2020 MonB5G European project and a researcher at CTTC in Spain. Additionally, he has been a RAN expert at INWI and Huawei Technologies in Morocco. With over 40 publications in top-tier journals and conferences, he has also contributed to a European patent. Dr. Chergui has held roles as an Associate Editor for IEEE Networking Letters and has chaired several workshops at IEEE Globecom and ICC. He has co-supervised PhD students in MSCA ITN projects such as 5GSTEPFWD and SEMANTIC. His accolades include the IEEE ComSoc CSIM 2021 Best Journal Paper Award and the IEEE ICC 2020 Best Paper Award. He is a Senior Member of IEEE.

Farhad Rezazadeh (Member, IEEE) received the Ph.D. degree (Excellent Cum Laude) in Signal Theory and Communications from Technical University of Catalonia (UPC), Barcelona, Spain. He is currently an AI Scientists at Hostelworld Group. He was formerly a researcher at the CTTC, Barcelona, Spain. He participated in 8 European and National 5G/B5G/6G R&D projects with leading and technical tasks in the areas of Applied AI. He was a secondee at NEC Lab Europe and had scientific missions at TUM, Germany, TUHH, Germany and UdG, Spain. He is a Marie Sklodowska-Curie Ph.D. grantee and won 5 different IEEE/IEEE ComSoc grants, 2 European Cooperation in Science and Technology grants, and a Catalan Government Ph.D. Grant. He is an active member of IEEE Young Professionals and IEEE Spain - Technical Activities and Standards with more than 29 top-tier journals/conferences and also book chapters. He actively serves as Organizing, Chair, Reviewer, and TPC member in IEEE and Guest Editor in Elsevier.

Merouane Debbah (Fellow, IEEE) He is a researcher, educator and technology entrepreneur. Over his career, he has founded several public and industrial research centers, start-ups and is now Professor at Khalifa University of Science and Technology in Abu Dhabi and founding Director of the KU 6G Research Center. He is also the Chief Scientific AI Advisor at the Technology Innovation Institute. He is a frequent keynote speaker at international events in the field of telecommunication and AI. His research has been lying at the interface of fundamental mathematics, algorithms, statistics, information and communication sciences with a special focus on random matrix theory and learning algorithms. In the Communication field, he has been at the heart of the development of small cells (4G), Massive MIMO (5G) and Large Intelligent Surfaces (6G) technologies. In the AI field, he is known for his work on Large Language Models, distributed AI systems for networks and semantic communications. He received multiple prestigious distinctions, prizes and best paper awards (more than 40 IEEE best paper awards) for his contributions to both fields and according to research.com is ranked as the best scientist in France in the field of Electronics and Electrical Engineering. He is an IEEE Fellow, a WWRF Fellow, a Eurasip Fellow, an AAIA Fellow, an Institut Louis Bachelier Fellow and a Membre émérite SEE. His recent work led to the development of NOOR (upon it release, largest language model in Arabic) released in 2022, Falcon LLM (upon its release, top ranked open source large language model) released in 2023 and the Falcon Foundation in 2024. The Falcon Model Series and The Falcon Foundation have positioned the UAE as a global leader in the generative AI field. He is a member of the Marconi Prize Selection Advisory Committee.

Christos Verikoukis(Senior Member, IEEE) [SM'07] (chverik@gmail.com) received the Ph.D. degree from Technical University of Catalonia (UPC), Barcelona, Spain, in 2000. He is currently an Associate Proferssor with the Univeristy of Patras. He has authored 158 journal papers and over 200 conference papers. He is also a co-author of three books, 14 chapters in other books, and two patents. He has participated in more than 40 competitive projects, and has served as a project coordinator of several funded projects from the European Comission and of national projects in Greece and Spain. He is currently the IEEE ComSoc GITC member and the editor-in-chief of the IEEE Networking Letters.