

# LLM-Based Agentic Negotiation for 6G: Addressing Uncertainty Neglect and Tail-Event Risk

Hatim Chergui, *Senior Member, IEEE*, Farhad Rezazadeh, *Member, IEEE*, Mehdi Bennis, *Fellow, IEEE*, and Merouane Debbah, *Fellow, IEEE*,

**Abstract**—A critical barrier to the trustworthiness of sixth-generation (6G) agentic autonomous networks is the *uncertainty neglect bias*; a cognitive tendency for large language model (LLM)-powered agents to make high-stakes decisions based on simple averages while ignoring the tail risk of extreme events. This paper proposes an unbiased, risk-aware framework for agentic negotiation, designed to ensure robust resource allocation in 6G network slicing. Specifically, agents leverage Digital Twins (DTs) to predict full latency distributions, which are then evaluated using a formal framework from extreme value theory, namely, Conditional Value-at-Risk (CVaR). This approach fundamentally shifts the agent’s objective from reasoning over the mean to *reasoning over the tail*, thereby building a statistically-grounded buffer against worst-case outcomes. Furthermore, our framework ensures full uncertainty awareness by requiring agents to quantify *epistemic uncertainty*—confidence in their own DTs predictions—and propagate this meta-verification to make robust decisions, preventing them from acting on unreliable data. We validate this framework in a 6G inter-slice negotiation use-case between an eMBB and a URLLC agent<sup>1</sup>. The results demonstrate the profound failure of the biased, mean-based baseline, which consistently fails its SLAs with a 25% rate. Our unbiased, CVaR-aware agent successfully mitigates this bias, eliminating SLA violations and reducing the URLLC and eMBB p99.999 latencies by around 11%. We show this reliability comes at the rational and quantifiable cost of slightly reduced energy savings to 17%, exposing the *false economy* of the biased approach. This work provides a concrete methodology for building the trustworthy autonomous systems required for 6G.

**Index Terms**—6G, agentic AI, bias, digital twin, extreme value theory, uncertainty, negotiation, reasoning, SLA.

## I. INTRODUCTION

THE evolution towards 6G networks necessitates unprecedented levels of operational autonomy, targeting the TM Forum’s Levels 4 (Closed-Loop Automation) and 5 (Full Autonomy) [1]. Achieving this paradigm shift requires moving towards *agentic systems* [2], which must be capable of reasoning, planning, and negotiating at the goal level to dynamically manage network functions, slice orchestration, and service assurance in highly complex and volatile environments. As these LLM-powered agentic systems are deployed to manage critical 6G functions, ensuring their reliability and safety is paramount.

H. Chergui is with the i2CAT Foundation, Spain. (e-mails: chergui@ieee.org, name.surname@i2cat.net)

F. Rezazadeh is with the Hostelworld Group, BrainOmega, and the Technical University of Catalonia (UPC), Spain (e-mail: farhad.rezazadeh@upc.edu).

M. Bennis is with the University of Oulu, Finland (e-mail: mehdi.bennis@oulu.fi).

M. Debbah is with the Khalifa University of Science and Technology, 127788 Abu Dhabi, UAE (e-mail: merouane.debbah@ku.ac.ae).

<sup>1</sup>The source code is available for non-commercial use at <https://github.com/HatimChergui/agentic-6g-network-slicing>.

However, recent research highlights a significant challenge: the emergence of cognitive biases in agentic AI, mirroring those found in humans. Drawing parallels with well-documented human biases, 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. The foundational understanding of these systematic errors was established by Tversky and Kahneman in their seminal paper, “Judgment under Uncertainty: Heuristics and Biases” [3], which demonstrated that humans rely on cognitive shortcuts, or heuristics, leading to predictable and systematic errors. Biases affect every layer of an agent’s behavior, from initial perception and reasoning to final decision-making and action execution. The recent work by Xie et al. [4] highlights the growing concern of these biases, specifically exploring their manifestation within large language models operating in multi-agent systems, a structure increasingly relevant to decentralized 6G architectures.

The insidious impact of biases can be observed throughout the entire agentic system pipeline, from perception to action, manifesting primarily across four layers, namely, (i) Biases are deeply ingrained in the training data, stemming from historical imbalances, cultural skews, and sampling errors. For instance, a large language model trained predominantly on data from a region with a specific, older network infrastructure might, when deployed elsewhere, exhibit a bias towards that legacy architecture, failing to optimally manage a newer, more advanced one; (ii) 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 example, prompting a network agent to “maximize throughput at all costs” can introduce a framing effect bias, leading to a solution that ignores critical latency or fairness metrics; (iii) The agent’s reasoning paths [5] 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 confirmation bias, only seeking out evidence that confirms a pre-existing threat model and overlooking a novel, unseen attack vector. These internal processes are akin to human shortcuts, where evidence is mis-weighted and readily accessible information is prioritized; and (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.

### A. Related Work

1) **Cognitive Biases in Agentic Systems:** A comprehensive contribution in this domain is [6], which presents a structured tutorial on well-known cognitive biases in 6G agentic systems, focusing on their definitions, mathematical formulation, emergence in 6G LLM-based agents and mitigation strategies at both the agent and 6G system levels, while providing practical use-cases with an accompanying source code. In [7], the authors demonstrate that iterative discussions amplify bias, forming conversational echo chambers as agents converge on consensus. This shows distortions can arise from interaction dynamics, not just pretrained knowledge. Structurally, the *fairness in agentic AI* framework [8] examines how systemic distortions arise from decentralized collaboration, linking ethical alignment to negotiation constraints. Besides, the *hidden profile benchmark* [9] shows that LLM agents often fail to share critical, unevenly distributed information, mirroring human groupthink and highlighting vulnerabilities in reasoning diversity. On the other hand, [10] presents the largest systematic study of cognitive biases in LLMs, evaluating eight well-established biases across 45 models using more than 2.8 million generated responses. It introduces a new multiple-choice-based evaluation framework, creates a psychologist-validated dataset of 220 decision scenarios, and develops a scalable method for generating diverse prompts from human-authored templates. The results show that LLMs display bias-consistent behavior in 17.8–57.3% of cases across a wide range of judgment and decision-making contexts. The analysis further reveals that larger models ( $>32B$ ) reduce bias in 39.5% of cases, while more detailed prompts mitigate most biases by up to 14.9%, except for Overattribution, which increases by up to 8.8%.

2) **Risk-Aware Optimization in 6G:** Risk-based optimization has likewise been introduced in various 6G setups. For instance, [11] introduces a risk-aware status-updating policy for real-time IoT monitoring—highly relevant for ultra-reliable, low-latency 6G systems—by jointly minimizing average age of information (AoI), tail AoI loss, and energy cost. Using CVaR to capture rare but severe AoI spikes, the authors reformulate an otherwise intractable history-dependent control problem as a standard MDP augmented with risk-level variables, enabling a dynamic-programming solution via a risk-aware Bellman operator. This yields a tractable stationary policy that effectively accounts for extreme AoI events, offering a practical approach for robust and reliable 6G IoT operation. Besides, [12] presents a risk-sensitive resource allocation method for handling URLLC traffic—an essential requirement for ultra-reliable, low-latency services in 6G networks—while safeguarding ongoing eMBB

transmissions that are typically punctured by URLLC arrivals. Using CVaR to quantify the risk of degrading low-rate eMBB users and a chance-constrained formulation to enforce URLLC reliability, the authors relax the reliability constraint via Markov's inequality and decompose the problem into two convex subproblems solved iteratively. The resulting approach efficiently allocates resources to incoming URLLC packets while simultaneously maintaining both eMBB protection and URLLC reliability.

3) **Bayesian Digital Twins:** In [13], the authors highlight uncertainty quantification as a foundational requirement for reliable Digital Twins, reviewing how Bayesian and probabilistic methods are essential for capturing modeling errors, data noise, and operational variability across prediction, monitoring, and optimization tasks, illustrated through a battery DT case study. Besides, [14] advances this perspective by developing a Digital Twin based on a nonparametric Bayesian network that explicitly models and propagates epistemic uncertainty in complex-system degradation. Through real-time updating with Gaussian particle filtering and Dirichlet process mixture modeling, the DT continuously reduces uncertainty, adapts its structure, and delivers more accurate and trustworthy health monitoring. Finally, [15] introduces a Bayesian Digital Twin framework for wireless systems that explicitly models epistemic uncertainty arising from limited or imperfect PT-to-DT data.

### B. Gap and Contributions

While existing work identifies agentic biases, risk-based optimization and bayesian digital-twins, it overlooks a critical requirement for robust autonomy in 6G (TM Forum Levels 4/5): *uncertainty quantification and communication* in LLM-based agentic systems. Autonomous components must convey the confidence of their decisions. Failure to model this introduces *uncertainty neglect bias*, where low-confidence predictions are treated as high-confidence facts. This erroneous certainty can propagate throughout the multi-agent system, causing cascading failures, suboptimal resource allocation, and ultimately undermining network stability. This paper addresses this critical gap by

- Proposing a risk-aware negotiation framework in 6G LLM agents using Conditional Value-at-Risk to mitigate uncertainty neglect bias, shifting agentic reasoning from simple averages to tail-end latency risk at a practical confidence level  $\alpha = 0.99999$  for 6G critical services;
- Introducing a mechanism for epistemic uncertainty awareness, compelling agents and tools to quantify and *propagate* confidence in their own predictions to prevent decisions based on unreliable data;
- Validating the framework in a 6G eMBB/URLLC inter-slice negotiation, showing it eliminates SLA violations and quantifies the trade-off between reliability (reduced latency) and network efficiency (energy savings).

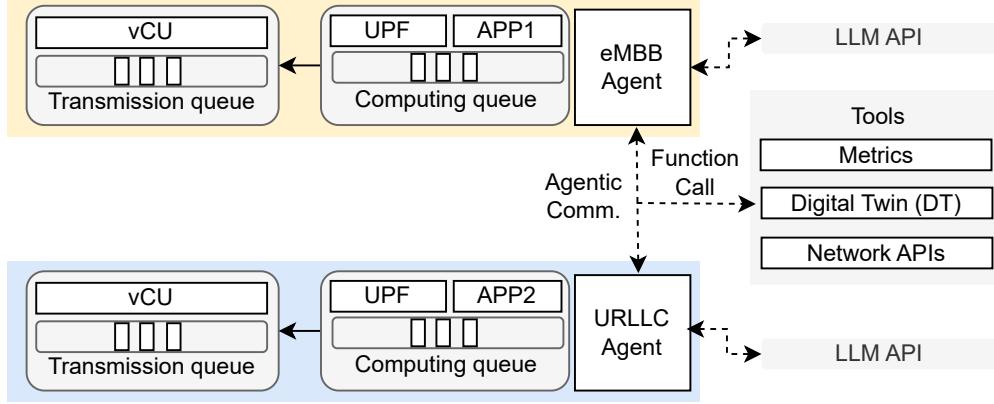


Figure 1: Agentic AI-driven 6G edge-RAN slicing.

## II. NETWORK SLICING MODEL AND PROBLEM FORMULATION

### A. Network Slicing Queuing Model and Digital Twin

We consider a network slicing architecture spanning an Edge computing domain and a Radio Access Network (RAN) domain as depicted in Figure 1. Service requests for a slice  $i$  first arrive at the Edge, incurring a computation latency  $L_i^{\text{edge}}$ . Processed packets are then enqueued for transmission over the wireless RAN, incurring a transmission latency  $L_i^{\text{RAN}}$ . The total end-to-end (E2E) latency is  $L_i = L_i^{\text{edge}} + L_i^{\text{RAN}}$  [16].

In our setup, both domains are resource-constrained. Agents must negotiate for a partition of the total available RAN bandwidth ( $B_{\text{total}}$ ) and the total Edge CPU capacity ( $F_{\text{total}}$ ). An agent  $i$ 's action is thus a vector  $a_i = (b_i, f_i)$ , where  $b_i$  is its allocated bandwidth and  $f_i$  is its allocated CPU. These actions determine the service capacity in each queue, and thus the latencies  $L_i^{\text{edge}}$  and  $L_i^{\text{RAN}}$ . The agent's objective is to manage the total latency  $L_i$  by controlling both  $b_i$  and  $f_i$ .

Each agent  $i$  is equipped with a private Digital Twin (DT) based on queuing theory. At each discrete time interval  $t$  of duration  $\tau$ , a number of bits  $A_{i,t}$  arrive at the Edge, governed by a time-varying, trial-specific stochastic process,

$$\Lambda_{i,t} = \lambda_{i,t} \cdot \tau, \quad (1)$$

where the mean arrival rate  $\mathbb{E}[\lambda_{i,t}]$  is assumed lower than the service rate, avoiding any queue divergence. The computation queue at the edge,  $Q_{i,t}^{(e)}$ , evolves as,

$$Q_{i,t+1}^{(e)} = \max \left( 0, Q_{i,t}^{(e)} - D_{i,t}^{(e)} \right) + \Lambda_{i,t}, \quad (2)$$

where  $D_{i,t}^{(e)}$  is the number of bits processed by the Edge. This is determined by the agent's allocated CPU  $f_i$  and a deterministic processing rate  $C_{CPU}$ ,

$$D_{i,t}^{(e)} = \tau \times C_{i,t}^{(e)}(f_i) = \tau \cdot f_i \cdot C_{CPU}. \quad (3)$$

The RAN communication queue,  $Q_{i,t}^{(r)}$ , is updated based on the output of the compute queue,

$$Q_{i,t+1}^{(r)} = \max \left( 0, Q_{i,t}^{(r)} - D_{i,t}^{(r)} \right) + \min \left( Q_{i,t}^{(e)} + \Lambda_{i,t}, D_{i,t}^{(e)} \right), \quad (4)$$

where  $D_{i,t}^{(r)}$  is the number of bits transmitted. This is determined by the agent's action  $b_i$  and the primary source of wireless uncertainty: the stochastic Spectral Efficiency (SE),  $SE_{i,t}$ , i.e.,

$$D_{i,t}^{(r)} = \tau \times C_{i,t}^{(r)}(b_i, SE_{i,t}) = \tau \cdot b_i \cdot SE_{i,t}. \quad (5)$$

Using Little's Law [17], the average E2E latency  $L_{i,T}$  is the sum of the average queue lengths divided by the average arrival rate, i.e.,

$$L_{i,T} = \frac{1}{\mathbb{E}[\Lambda_{i,t}]T} \sum_{t=1}^T (Q_{i,t}^{(e)} + Q_{i,t}^{(r)}), \quad (6)$$

where  $\mathbb{E}[\Lambda_{i,t}]$  is approximated from traffic traces per slice and  $T$  is the horizon for latency averaging. The agent's objective is to control its action vector  $a_i = (b_i, f_i)$  to keep  $L_{i,T}$  below its SLA, while also minimizing a linear power consumption cost  $P_i(a_i)$ ,

$$P_i(a_i) = P_{\text{static},i} + C_{BW} \cdot b_i + C_{CPU} \cdot f_i, \quad (7)$$

where  $C_{BW}$  and  $C_{CPU}$  are bandwidth and CPU power costs, respectively.

### B. Agentic Negotiation Framework

For the sake of focusing on the effect of bias mitigation and without loss of generality, the system hosts two competing slices,  $i \in \{1, 2\}$ , each represented by an autonomous, LLM-based agent. These agents must negotiate to partition both the total available bandwidth  $B_{\text{total}}$  and total edge CPU  $F_{\text{total}}$ . An agent's action is a vector  $a_i = (b_i, f_i)$ . The slices are: i) eMBB (Slice  $i = 1$ ), with an agent with a relaxed latency SLA (e.g.,  $L_{1,\text{SLA}} = 50\text{ms}$ ); and ii) URLLC (Slice  $i = 2$ ), with an agent

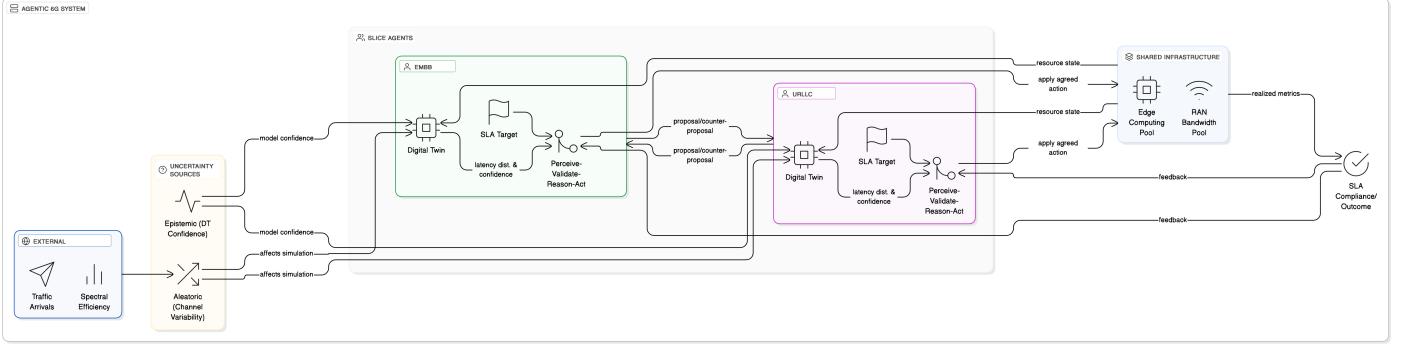


Figure 2: Risk-aware agentic system concept.

with a strict latency SLA (e.g.,  $L_{2,SLA} = 10\text{ms}$ ). These two agents must continuously negotiate to find a mutually agreeable partition  $(a_1, a_2)$  such that  $b_1 + b_2 \leq B_{\text{total}}$  and  $f_1 + f_2 \leq F_{\text{total}}$ . In this respect, several principles are considered, namely,

- Given that single-shot (Stackelberg) negotiation protocol would inherently bias the outcome toward the leader agent, we adopt a constrained, multi-round protocol. Agent 1 (eMBB) initiates the process by proposing a desired action  $a_1 = (b_1, f_1)$ . Agent 2 (URLLC) evaluates this proposal against its strict constraints. Unlike a static leader-follower game, Agent 2 retains the right to reject the proposal and issue a counter-proposal  $a'_2$ , reversing the flow of the negotiation. This alternating exchange continues until a consensus is reached or a maximum round limit is triggered. Therefore, dynamically selecting which agent should initiate the negotiation offers little benefit.
- It is imperative to distinguish between the sequential nature of the *negotiation logic* and the concurrent nature of *resource utilization*. While the decision-making process is turn-based to strictly resolve conflict, both slices access and utilize the allocated shared resources simultaneously during the operational time slot  $t$ .
- Each agent operates within a structured perceive-validate-reason-act loop. In the *Validate* phase, the agent employs its internal DT to predict the impact (i.e., the latency distribution) of a candidate action  $a_i$ . This validation is the core of its reasoning, allowing it to self-correct proposals before acting.

A full example of the agentic negotiation is given in the Appendix. Note that a real deployment of our system requires a local reasoning-capable LLM instead of remote API to comply with near-real time constraints.

### III. UNCERTAINTY AND RISK-AWARE AGENTIC SYSTEM

In this dynamic system, an autonomous agent  $i$  must select an action  $a_i = (b_i, f_i)$  (a bandwidth and CPU allocation) from a set of possible actions  $\mathcal{A}_i$ . The outcome of this action is the uncertain E2E latency,  $L_i(a_i) = L_i^{\text{edge}}(f_i) + L_i^{\text{RAN}}(b_i)$ . The agent's objective is to satisfy its Service Level Agreement (SLA),  $L_{i,SLA}$ . As described in Sec. II, the primary source of aleatoric uncertainty is the stochastic Spectral Efficiency (SE),

$SE_{i,t}$ . Before making a proposal, the agent uses its DT to run a Monte Carlo simulation by sampling  $SE_{i,t}$  to predict a full *distribution* of potential latencies for the next timestep.

#### A. Uncertainty and Risk Definition

The DT takes a proposed action  $a_i = (b_i, f_i)$  and the current state (queues  $Q_{i,t}^{(e)}, Q_{i,t}^{(r)}$ , time  $t$ ) to generate a latency distribution  $L_i(a_i)$  from  $N_{mc}$  samples, i.e.,

$$\{L_i^{(k)}(a_i)\}_{k=1}^{N_{mc}} \text{ where } L_i^{(k)}(a_i) = L_i^{(e)}(f_i) + L_i^{(r,k)}(b_i). \quad (8)$$

Here,  $L_i^{(e)}(f_i)$  is the compute latency (derived from Eq. (3) of the System Model) and  $L_i^{(r,k)}(b_i)$  is the stochastic radio latency (derived from Eq. (5) using the  $k$ -th sample  $SE_{i,t+1}^{(k)}$  from  $\mathcal{U}[SE_{min}, SE_{max}]$ ). This output distribution allows us to quantify two distinct types of uncertainty as depicted in Figure 2.

1) **Aleatoric Uncertainty (Statistical Risk):** This is the inherent randomness in the system, represented by the *shape* of the distribution  $L_i(a_i)$ . We quantify its tail risk using the CVaR at an  $\alpha$  confidence level (in our case,  $\alpha = 0.99999$ ). CVaR $_\alpha$  represents the expected latency in the worst  $(1 - \alpha)\%$  of cases, i.e.,

$$\text{VaR}_\alpha(L_i(a_i)) = \inf\{l \in \mathbb{R} : P(L_i(a_i) \leq l) \geq \alpha\}, \quad (9)$$

$$\text{CVaR}_\alpha(L_i(a_i)) = \mathbb{E}[L_i(a_i) | L_i(a_i) > \text{VaR}_\alpha(L_i(a_i))]. \quad (10)$$

In practice, VaR and CVaR are estimated empirically from the latency samples by sorting the samples and averaging those beyond the empirical  $\alpha$ -quantile

2) **Epistemic Uncertainty (DT Prediction Confidence):** This is the DT's *uncertainty about its own prediction*. It measures how reliable the entire predicted distribution is. We model this using the coefficient of variation (CV) of the latency distribution and define the *Epistemic Confidence Score*  $C_E(a_i)$  as,

$$C_E(a_i) = \max \left( 0, 1 - \frac{\sigma_L(a_i)}{\mu_L(a_i)} \right), \quad (11)$$

where  $\mu_L(a_i)$  and  $\sigma_L(a_i)$  are the mean and standard deviation of  $L_i(a_i)$ . A  $C_E(a_i)$  score near 1.0 implies high confidence (low variance relative to the mean).

### B. Approach 1: Biased Agent (Neglect of Uncertainty)

The biased agent represents a common but flawed heuristic. It exhibits two forms of cognitive bias: it *neglects aleatoric risk* by performing an act of willful information reduction, discarding the rich data of the distribution's tail and basing its decisions solely on the mean  $\mu_L(a_i)$ ; and it *neglects epistemic risk* by ignoring the confidence score  $C_E(a_i)$ , implicitly trusting its prediction even when the DT signals high variance.

The agent's decision metric is therefore the simple expected latency,

$$\bar{L}_{\text{biased}}(a_i) = \mu_L(a_i) = \mathbb{E}[L_i(a_i)]. \quad (12)$$

The agent's final state of *satisfaction* ( $\mathcal{S}$ ) is governed by a brittle logical policy. This policy fails to ask crucial questions like "what is the worst-case scenario?" or "how reliable is this prediction?" It is based on only two conditions: is the SLA met ( $\mathcal{M}$ ) based on the mean, and is the agent *not* over-provisioning ( $\mathcal{O}$ )?

$$\mathcal{M}_{\text{biased}} \iff \bar{L}_{\text{biased}}(a_i) \leq L_{i,\text{SLA}} \quad (13)$$

$$\mathcal{O}_{\text{biased}} \iff \bar{L}_{\text{biased}}(a_i) < \theta_{\text{biased}} \cdot L_{i,\text{SLA}} \quad (14)$$

where  $\theta_{\text{biased}}$  is an aggressive threshold (e.g., 0.7). The agent's final state of satisfaction is defined by the logical AND operation,

$$\mathcal{S}_{\text{biased}} \iff \mathcal{M}_{\text{biased}} \wedge \neg \mathcal{O}_{\text{biased}}. \quad (15)$$

This policy is what defines the *uncertainty neglect bias*. A low mean  $\mu_L(a_i)$  can easily mask a high  $\text{CVaR}_\alpha(a_i)$  that actually violates the SLA in practice.

### C. Approach 2: Unbiased Agent (CVaR-Aware Mitigation)

The unbiased agent is designed to actively mitigate both forms of risk simultaneously by creating a dynamic, risk-adjusted SLA target. First, it mitigates aleatoric risk by replacing its core objective with the  $\text{CVaR}_\alpha$ . Second, it mitigates epistemic risk by using the Epistemic Confidence Score  $C_E(a_i)$  to dynamically tighten its SLA target. It reasons that a low-confidence prediction must be treated with suspicion, and therefore it must buy a larger safety margin. Therefore, the agent's policy is based on a new, dynamic SLA target,  $L'_{i,\text{SLA}}$ , which is a direct function of its epistemic confidence, i.e.,

$$L'_{i,\text{SLA}}(a_i) = L_{i,\text{SLA}} \times C_E(a_i). \quad (16)$$

This dynamic target forces the agent to be more conservative when its confidence  $C_E(a_i)$  is low. For example, if the agent's confidence is 0.8 (80%), its 10ms SLA becomes a stricter 8ms internal target. The agent's core success check,  $\mathcal{M}_{\text{unbiased}}$ , now compares the aleatoric risk (CVaR) against this new dynamic, epistemically-aware target, i.e.,

$$\mathcal{M}_{\text{unbiased}}(a_i) \iff \text{CVaR}_\alpha(L_i(a_i)) \leq L'_{i,\text{SLA}}(a_i). \quad (17)$$

Similarly, the check for over-provisioning,  $\mathcal{O}_{\text{unbiased}}$ , is also performed against this stricter dynamic target,

$$\mathcal{O}_{\text{unbiased}} \iff \text{CVaR}_\alpha(L_i(a_i)) < \theta_{\text{unbiased}} \cdot L'_{i,\text{SLA}}(a_i), \quad (18)$$

---

### Algorithm 1: Risk-Aware Agentic Negotiation with Epistemic Uncertainty Mitigation

---

```

Input: Agents  $\mathcal{I} = \{1, 2\}$  (eMBB, URLLC), SLAs  $\{L_{i,\text{SLA}}\}$ , Constraints  $\mathcal{C}_{\text{total}} = \{B_{\text{tot}}, F_{\text{tot}}\}$ , Max Rounds  $N_{\text{rounds}}$ 
Output: Final Resource Allocation  $\mathcal{A}^* = \{a_1^*, a_2^*\}$ 
1 Initialize:  $a_1, a_2 \leftarrow \text{PropFairSplit}(\mathcal{C}_{\text{total}})$ 
2 for  $r \leftarrow 1$  to  $N_{\text{rounds}}$  do
   // -- Phase 1: Risk-Aware Validation --
   3   for  $i \in \mathcal{I}$  do
      4     Generate latency distribution  $\{L_i^{(k)}\}$  via Digital Twin
           DT( $a_i$ )
      5     Calculate Mean  $\mu_i$ , Std Dev  $\sigma_i$ , and Tail Risk CVaR $_{\alpha,i}$ 
      6     Calculate Epistemic Confidence:
            $C_E(a_i) \leftarrow \max(0, 1 - \sigma_i/\mu_i)$ 
           // Mitigate Bias: Tighten SLA based on
           Confidence
      7     Dynamic Target  $L'_{i,\text{SLA}} \leftarrow L_{i,\text{SLA}} \cdot C_E(a_i)$ 
           // Evaluate Satisfaction  $\mathcal{S}_{\text{unbiased}}$ 
      8      $\mathcal{M}_i \leftarrow (\text{CVaR}_{\alpha,i} \leq L'_{i,\text{SLA}})$  // SLA Met
      9      $\mathcal{O}_i \leftarrow (\text{CVaR}_{\alpha,i} < \theta \cdot L'_{i,\text{SLA}})$  // Over-prov
     10     $\deltaSat_i \leftarrow \mathcal{M}_i \wedge \neg \mathcal{O}_i$ 
   11   end
   12   if  $\deltaSat_1 \wedge \deltaSat_2$  then
   13     return  $\{a_1, a_2\}$  // Consensus Reached
   14   end
   // -- Phase 2: Counter-Proposals --
   15   for  $i \in \mathcal{I}$  do
   16      $j \leftarrow \mathcal{I} \setminus \{i\}$  // Opponent index
   17     Residual Constraints  $\mathcal{C}_{\text{rem}} \leftarrow \{B_{\text{tot}} - b_j, F_{\text{tot}} - f_j\}$ 
           // Sub-routine: Generative Reasoning with
           Local Search
   18     Reasoning( $a_j, \mathcal{C}_{\text{rem}}$ ):
   19        $\deltaPrompt \leftarrow \text{ConstructContext}(a_j, \mathcal{C}_{\text{rem}}, L'_{i,\text{SLA}})$ 
   20        $\mathcal{K} \leftarrow \text{LLM\_API}(\deltaPrompt)$  // Get 3
           candidates
   21       Select  $a_j^* \in \mathcal{K}$  minimizing Energy s.t.
            $\text{CVaR}_{\alpha,i} \leq L'_{i,\text{SLA}}$ 
           // Correction: Upward Search (Fix
           Violation)
   22       while  $\text{CVaR}_{\alpha,i}(a_j^*) > L'_{i,\text{SLA}}$  do
   23         Determine bottleneck (Radio vs. Compute)
   24         Increment  $b_i^*$  or  $f_i^*$  by step  $\delta$  within  $\mathcal{C}_{\text{rem}}$ 
   25       end
           // Correction: Downward Search (Save
           Energy)
   26       while  $\text{CVaR}_{\alpha,i}(a_j^*) \ll L'_{i,\text{SLA}}$  (Over-provisioned) do
   27         Decrement non-bottleneck resource by  $\delta$ 
   28       end
   29     end
   30     Update global allocation  $a_i \leftarrow a_i^*$ 
   31   end
   32 return  $\{a_1, a_2\}$  // Return final state (fallback)

```

---

where  $\theta_{\text{unbiased}}$  is a conservative threshold (e.g., 0.6). The agent's final state of satisfaction is defined by,

$$\mathcal{S}_{\text{unbiased}} \iff \mathcal{M}_{\text{unbiased}} \wedge \neg \mathcal{O}_{\text{unbiased}}. \quad (19)$$

This unified policy (Eqs. 16–19) is the core of the unbiased design. If confidence is low,  $L'_{i,\text{SLA}}$  decreases. This makes  $\mathcal{M}_{\text{unbiased}}$  harder to achieve (triggering an upward local search for more resources) and  $\mathcal{O}_{\text{unbiased}}$  harder to achieve (preventing a downward local search to save energy). This solves the failure mode by linking aleatoric and epistemic risk. The detailed procedure is summarized in Algorithm 1.

#### D. Meta-Verification in Practice: Propagating Confidence

The confidence\_score derived from Eq. 11 serves as the practical engine for meta-verification. Its utility is realized by actively propagating this trust metric through the agent's entire decision-making pipeline. **(i) Phase 1: Internal Verification and Selection**, where the agent first performs an internal self-assessment. Its Digital Twin simulates a set of candidate proposals,  $a_i = (b_i, f_i)$ , integrating performance (CVaR vs.  $L'_{i,\text{SLA}}$ ), efficiency (energy cost), and, crucially, the confidence\_score into a composite Score. This score functions as a critical weight, penalizing low-confidence (and thus low-target  $L'_{i,\text{SLA}}$ ) proposals and favoring options that are demonstrably robust and reliable.

```

1 INFO - [Turn 1] URLLC (Priority) proposing against
2 eMBB's prior state...
3 INFO - [URLLC] Internally testing 3 LLM proposals (
4 Constraints: 40.00M, 40.00G):
5 INFO - [URLLC] - Test Proposal 1 (18.00M, 15.50G)
6 [Score: 85.40]: SLA Met: True (Raw \text{CVaR}
7 _99999_latency_ms: 8.51ms, Conf: 0.94 -> Adjusted
8 Pred: 8.51ms vs Target: 10.00ms), Energy: 14.60W
9 INFO - [URLLC] - Test Proposal 2 (16.00M, 13.00G)
10 [Score: 86.90]: SLA Met: True (Raw \text{CVaR}
11 _99999_latency_ms: 9.82ms, Conf: 0.94 -> Adjusted
12 Pred: 9.82ms vs Target: 10.00ms), Energy: 13.10W
13 INFO - [URLLC] - Test Proposal 3 (13.50M, 12.00G)
14 [Score: -1001.23]: SLA Met: False (Raw \text{CVaR}
15 _99999_latency_ms: 11.23ms, Conf: 0.93 ->
16 Adjusted Pred: 11.23ms vs Target: 10.00ms),
17 Energy: 11.65W
18 INFO - [URLLC] Selected best proposal (BW: 16.00M,
19 CPU: 13.00, Score: 86.90)

```

**(ii) Phase 2: External Propagation for Transparent Reasoning**, where once the best proposal is selected (or derived from local search), the agent propagates its internal state into the "reasoning" field of its action proposal. This makes its decision-making transparent and verifiable by its counterpart.

```

1 INFO - [URLLC SAYS] PROPOSE_ACTION: {
2   "proposed_bandwidth_mhz": 16.0,
3   "proposed_cpu_ghz": 13.0,
4   "reasoning": "This is the balanced proposal. The
5     current \text{CVaR}_99999 latency is 9.77ms,
6     meeting the SLA with a confidence score of 0.94.
7     The radio latency (5.52ms) dominates compute
8     latency (3.26ms). This balanced proposal
9     implements decoupling by slightly increasing BW
10    (16.00 MHz) to further mitigate radio risk while
11    decreasing CPU (13.00 GHz) compared to the
12    baseline, conserving energy while ensuring
13    continued SLA adherence. [Final Check: Adjusted \
14    \text{CVaR}_99999_latency_ms: 9.82ms (SLA Met:
15    True), Energy: 13.10W]"
16 }

```

This act of reporting confidence elevates the negotiation from a simple exchange of demands (e.g., "I need 16 MHz of bandwidth and 13 GHz CPU frequency") to a sophisticated, verifiable dialogue about uncertainty (e.g., "increasing BW (16.00 MHz) to further mitigate radio risk while decreasing CPU (13.00 GHz) compared to the baseline, conserving energy").

## IV. EXPERIMENTAL RESULTS

To validate the theoretical framework, we conducted a series of simulations comparing the performance of the two agent

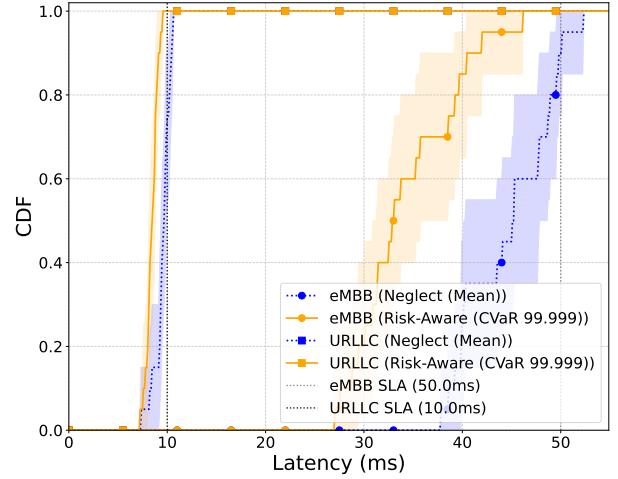


Figure 3: Latency CDF for both agents vs. various scenarios.

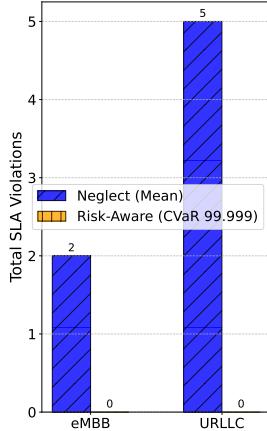
strategies.

#### A. Simulation Setup

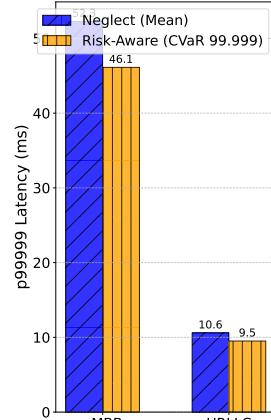
The simulation environment is configured with two slice agents, namely, eMBB ( $L_{1,\text{SLA}} = 50$  ms) and URLLC ( $L_{2,\text{SLA}} = 10$  ms)—which use the Gemini 2.5 Flash API for reasoning. The agents negotiate over a total RAN bandwidth of  $B_{\text{total}} = 40.0$  MHz and a total edge compute capacity of  $F_{\text{total}} = 40.0$  GHz, considering a multi-core setup. The edge compute processing rate is defined as  $R_{\text{cpu}} = 10$  Mbps per GHz. Each simulation comprises  $N_{\text{trials}} = 20$  independent negotiations, with each negotiation consisting of up to  $N_{\text{rounds}} = 5$  rounds. The main sources of aleatoric uncertainty arise from the Spectral Efficiency (SE), which fluctuates stochastically within the range  $\mathcal{U}[5.0, 7.0]$ , but also from the epistemic confidence of the DT predictions. The power consumption model is defined by a static base power of  $P_{\text{static}} = 5.0$  W, a bandwidth power cost of  $C_{\text{bw}} = 0.5$  W/MHz, and a linear CPU power cost of  $C_{\text{CPU}} = 0.2$  W/GHz. Two agent strategies are compared: (1) a Biased (Neglect of Uncertainty) strategy, where agents rely solely on the mean latency  $\mu_L(a_i)$  as defined in Eq. (12), and (2) an Unbiased (risk-aware Mitigation) strategy, where decisions are governed by the Conditional Value-at-Risk CVaR $_{\alpha}$  and Epistemic Confidence  $C_E$ , as formulated in Eqs. (11)–(19).

#### B. Analysis of Service Level Agreement (SLA) Performance

The primary goal of the agent is to satisfy its SLA. Figure 3 and Figure 4 provide a comprehensive view of SLA performance. As shown in Figure 3, the choice of strategy has a dramatic impact on SLA compliance. **URLLC (Strict 10ms SLA):** The Biased agent (blue dotted line) fails systematically. Its latency CDF begins to rise notably after the 10ms SLA, meaning a significant portion of its packets violate the deadline. In contrast, the Unbiased (risk-aware) agent (orange solid line) successfully optimizes for the tail, keeping its entire latency



(a) Total SLA violations for various scenarios.



(b) The 99.999th-percentile (p99.999) latency for various scenarios.

Figure 4: Debiasing gain from the risk-Aware (CVaR) strategy vs. the biased mean-based reasoning.

distribution strictly below the 10ms deadline. **eMBB (Relaxed 50ms SLA):** The Biased agent again fails. It exhibits a tail that crosses its 50ms SLA, with approximately 5% of its latencies. The Unbiased agent is more conservative, successfully containing its maximum latency to just over 45ms, well within the 50ms boundary.

Figure 4 quantifies the practical impact of mitigating the uncertainty neglect bias. **Total SLA Violations (Left):** The bar chart shows the Biased agent for eMBB caused 1 SLA violation across the 20 trials, whereas the Unbiased agent caused 0. This confirms that the tail-risk seen in the CDF translates to concrete failures. For URLLC, the Biased agent presented 5 SLA violations while the Unbiased registered no violations. **p99.999 Latency (Right):** This metric provides the clearest view of tail-risk performance. The Unbiased agent achieves a massive reduction in tail risk. For eMBB, it cuts the p99.999 latency from 52.3ms, i.e., SLA violation, down to 46.1ms which is a safe margin. For URLLC, it reduces the p99.999 latency from 10.6ms (a violation of the 10ms SLA) to 9.5ms, creating a vital safety buffer.

### C. Analysis of Energy Saving

SLA compliance is not the only objective; agents also attempt to save resources by minimizing bandwidth, which translates to energy savings. Figure 5 reveals the trade-off inherent in the two strategies. The Energy Saving CDF for the **Unbiased (risk-aware)** agent (orange line) is clearly shifted to the left of the **Biased (Neglect)** agent (blue line). This indicates that the risk-aware agent consistently achieves lower energy savings. For example, the median energy saving for the unbiased agent is approximately 17%, while the biased agent achieves a higher median saving of approximately 27%. This result is not a failure, but rather the expected, rational cost of reliability. The unbiased agent buys its robust SLA compliance

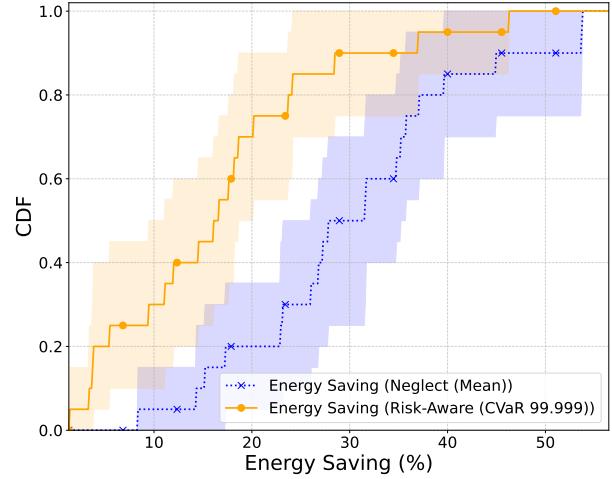


Figure 5: CDF of Energy Saving for both slices vs. scenarios.

(Figure 3) by being more conservative. It purposefully allocates a statistically-grounded bandwidth buffer to protect against tail risk and model uncertainty (Epistemic Confidence). This larger buffer consumes more power, thus reducing the total energy saved. The biased agent, in contrast, creates a *false economy*. It appears more efficient by saving more energy, but it does so only because it is blind to the tail risk that ultimately causes it to fail its primary objective of SLA compliance.

## V. CONCLUSION

Trustworthy 6G autonomous networks require agentic systems that can reason beyond simple averages and account for high-stakes, low-probability events. This paper addressed the critical *uncertainty neglect bias* through proposing a risk-aware negotiation framework. By compelling agents to use Conditional Value-at-Risk for reasoning over latency tails and DTs to quantify and propagate their own *epistemic* uncertainty, our approach ensures robust reasoning over resource allocation. We demonstrated in a 6G cross-domain inter-slice negotiation use case that this unbiased, risk-aware method eliminates the SLA violations endemic to mean-based approaches, significantly reduces p99.999 latencies, and rationally quantifies the trade-off between reliability and efficiency. This work provides a concrete methodology for building the robust, verifiable, and trustworthy autonomous agents required for future 6G systems.

## VI. APPENDIX: EXAMPLE OF NEGOTIATION LOG

```

1 [NEGOTIATION START: Risk-Aware Mitigation (CVaR_99.999)]
2 --- Round 1/5 ---
3 INFO - Current Joint Request: (eMBB: 21.93M, 21.93G),
4 INFO - [eMBB Eval] Goal Met: False (SLA Met: True,
      Over-provisioned: True, Adjusted Pred
      CVaR_99999_latency_ms: 16.9 vs Target: 50.0)
5 INFO - [URLLC Eval] Goal Met: False (SLA Met: False
      , Over-provisioned: False, Adjusted Pred
      CVaR_99999_latency_ms: 11.2 vs Target: 10.0)
6 INFO - [Counter] eMBB turn...

```

```

7 INFO - [eMBB] Internally testing 3 proposals from LLM 31 INFO - [URLLC SAYS] PROPOSE_ACTION: {
8 INFO - [eMBB] - Test Proposal: 20.00 MHz, 20.00 32 "proposed_bandwidth_mhz": 23.0,
GHz, SLA Met: True (Raw CVaR_99999_latency_ms: 33 "proposed_cpu_ghz": 19.5,
18.45ms, Conf: 0.94 -> Adjusted Pred: 18.45ms vs 34 "reasoning": "BALANCED PROPOSAL. The SLA failure
Target: 50.00ms), Energy: 16.50W, Score: 83.50 bottleneck. We propose 23.00 MHz BW to
9 INFO - [eMBB] - Test Proposal: 17.50 MHz, 17.50 significantly reduce radio queuing delay and push
GHz, SLA Met: True (Raw CVaR_99999_latency_ms: the CVaR below the 10.0ms threshold, while
21.12ms, Conf: 0.94 -> Adjusted Pred: 21.12ms vs keeping resource usage below the maximum
Target: 50.00ms), Energy: 14.75W, Score: 85.25 available limits (26.00 MHz BW, 36.00 GHz CPU).
10 INFO - [eMBB] - Test Proposal: 14.00 MHz, 14.00 The confidence_score is 0.94, confirming
GHz, SLA Met: True (Raw CVaR_99999_latency_ms: prediction stability. [Internal DT Test: Final
26.44ms, Conf: 0.93 -> Adjusted Pred: 26.44ms vs selected proposal. Adjusted CVaR_99999_latency_ms
Target: 50.00ms), Energy: 12.30W, Score: 87.70 : 9.31ms (vs Target: 10.00ms, SLA Met: True),
11 INFO - [eMBB] Best proposal (14.00M, 14.00G) is Predicted Energy: 17.90W]"
OVERPROVISIONED (Adjusted Pred: 26.44ms vs Target 35 }
: 50.00ms). Starting 'DOWNWARD' local search...
12 INFO - [eMBB] - Energy Search: Compute latency 36 INFO - [Proposals] Feasible by protocol. Updating
(8.81ms) is NON-bottleneck. Cutting CPU.
13 INFO - [eMBB] - Energy Search Attempt 1: (14.00M, current requests.
12.00G) -> Adjusted Pred CVaR_99999_latency_ms: 37 INFO -
27.31ms vs Target: 50.00ms. SLA Met: True
14 INFO - [eMBB] - Energy Search Attempt 2: (14.00M, 38 --- Round 2/5 ---
10.00G) -> Adjusted Pred CVaR_99999_latency_ms: 39 INFO - Current Joint Request: (eMBB: 14.00M, 4.00G)
26.58ms vs Target: 50.00ms. SLA Met: True , (URLLC: 23.00M, 19.50G)
40 INFO - [eMBB Eval] Goal Met: True (SLA Met: True,
Over-provisioned: False, Adjusted Pred
CVaR_99999_latency_ms: 36.5 vs Target: 50.0)
41 INFO - [URLLC Eval] Goal Met: True (SLA Met: True,
Over-provisioned: False, Adjusted Pred
CVaR_99999_latency_ms: 9.3 vs Target: 10.0)
42 INFO - [COMMIT] Both satisfied. Final agreement
reached.
43 INFO - [OUTCOME] Negotiation SUCCEEDED with split (
eMBB: 14.00M, 4.00G), (URLLC: 23.00M, 19.50G).

```

## REFERENCES

- [1] TM Forum, “Autonomous Networks: Exploring the Evolution from Level 0 to Level 5,” TM Forum, Tech. Rep., Dec. 2021.
- [2] 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>
- [3] A. Tversky and D. Kahneman, “Judgment under Uncertainty: Heuristics and Biases,” *Science*, vol. 185, no. 4157, pp. 1124–1131, 1974. [Online]. Available: <https://www.science.org/doi/abs/10.1126/science.185.4157.1124>
- [4] 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.
- [5] F. Rezazadeh, H. Chergui, M. Debbah, H. Song, D. Niyato, and L. Liu, “Agentic World Modeling for 6G: Near-Real-Time Generative State-Space Reasoning,” *arXiv preprint arXiv:2511.02748*, 2025.
- [6] H. Chergui, F. Rezazadeh, M. Debbah, and C. Verikoukis, “A Tutorial on Cognitive Biases in Agentic AI-Driven 6G Autonomous Networks,” 2025. [Online]. Available: <https://arxiv.org/abs/2510.19973>
- [7] 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>
- [8] 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>
- [9] 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>
- [10] R. A. Knipper, C. S. Knipper, K. Zhang, V. Sims, C. Bowers, and S. Karmaker, “The bias is in the details: An assessment of cognitive bias in llms,” 2025. [Online]. Available: <https://arxiv.org/abs/2509.22856>
- [11] B. Zhou, W. Saad, M. Bennis, and P. Popovski, “Risk-aware optimization of age of information in the internet of things,” in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

- [12] M. Alsenwi, N. H. Tran, M. Bennis, A. Kumar Bairagi, and C. S. Hong, "embb-urllc resource slicing: A risk-sensitive approach," *IEEE Communications Letters*, vol. 23, no. 4, pp. 740–743, 2019.
- [13] A. Thelen, X. Zhang, O. Fink, Y. Lu, S. Ghosh, B. D. Youn, M. D. Todd, S. Mahadevan, C. Hu, and Z. Hu, "A comprehensive review of digital twin—part 2: Roles of uncertainty quantification and optimization, a battery digital twin, and perspectives," *Structural and Multidisciplinary Optimization*, vol. 66, no. 1, p. 1, 2023.
- [14] J. Yu, Y. Song, D. Tang, and J. Dai, "A digital twin approach based on nonparametric bayesian network for complex system health monitoring," *Journal of Manufacturing Systems*, vol. 58, pp. 293–304, 2021, digital Twin towards Smart Manufacturing and Industry 4.0. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0278612520301151>
- [15] C. Ruah, O. Simeone, and B. M. Al-Hashimi, "A bayesian framework for digital twin-based control, monitoring, and data collection in wireless systems," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, pp. 3146–3160, 2023.
- [16] 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>
- [17] J. D. C. Little, "A Proof for the Queueing Formula:  $l = \lambda w$ ," *Operations Research*, vol. 9, no. 3, pp. 383–387, 1961.