

# ENWAR 3.0: An Agentic Multi-Modal LLM Orchestrator for Situation-Aware Beamforming, Blockage Prediction, and Handover Management

Ahmad M. Nazar, *Member, IEEE*, Abdulkadir Celik, *Senior Member, IEEE*,  
Asmaa Abdallah, *Senior Member, IEEE*, Mohamed Y. Selim, *Senior Member, IEEE*,  
Daji Qiao, *Senior Member, IEEE*, and Ahmed M. Eltawil, *Senior Member, IEEE*

**Abstract**—Maintaining robust millimeter-wave (mmWave) connectivity in vehicular networks requires real-time adaptation to environmental dynamics, sensor degradation, and link variability. This paper presents ENWAR 3.0, an environment-aware reasoning framework that unifies multi-modal sensing, agentic large language models (LLMs), and context-driven model selection for predictive beamforming, blockage detection, and handover management. Building upon prior iterations of ENWAR, the proposed architecture integrates a classifier-driven assessment of sensor health with a primed LLM that orchestrates multiple specialized agents through structured, task-aware prompting. A novel synthetic degradation pipeline enables the training of a sensor degradation classifier that detects real-time impairments across camera, radar, LiDAR, and GPS inputs, achieving over 99% accuracy. The LLM, trained via chain-of-thought (CoT) priming and human-in-the-loop feedback, coordinates agent calls for beam selection, blockage forecasting, and environment perception while dynamically loading sensor-specific models based on environmental context. Extensive evaluations across 15 sensor combinations demonstrate that ENWAR 3.0 delivers state-of-the-art performance in both predictive accuracy and interpretability, with beam selection accuracy exceeding 88%, blockage F1-scores surpassing 98%, and reasoning correctness reaching 87% on complex decision prompts. This work establishes a scalable foundation for LLM-integrated wireless systems that reason, perceive, and adapt in real-time.

## I. INTRODUCTION

Millimeter wave (mmWave) communication enables high-throughput, directional links for next-generation wireless systems, but remains highly sensitive to environmental dynamics such as blockage, sensor degradation, and mobility-induced channel variability. Maintaining reliable and low-latency infrastructure-to-vehicle (I2V) connectivity, therefore, requires joint perception and adaptive decision-making under real-time constraints [2].

Traditional network control mechanisms rely on rule-based heuristics, static beam tracking, or loosely coupled perception

A. M. Nazar is with the Department of Electrical and Computer Engineering, Iowa State University (ISU), Ames, IA, 50014, USA, and with Gladiolus Technological Institute, USA

M. Y. Selim, and D. Qiao are with the Department of Electrical and Computer Engineering, ISU, Ames, IA, 50014, USA.

A. Celik, is with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.

A. Abdallah, and A. M. Eltawil are with Computer, Electrical, and Mathematical Sciences & Engineering (CEMSE) Division at King Abdullah University of Science and Technology (KAUST), Thuwal, 23955 KSA.

A conference version of this work is submitted to ASILOMAR Signal Processing Conference'25 [1].

modules. Such approaches do not scale well to multi-modal sensing environments and cannot adapt decisions based on sensor reliability, environmental context, and historical system behavior [3]. As roadside units (RSUs) integrate camera, LiDAR, radar, and GPS inputs, practical system orchestration demands unified reasoning across heterogeneous modalities.

To address these challenges, we introduce **ENWAR 3.0**, the latest evolution in the **EN**vironment-a**WA**Re, multi-modal large language model (LLM) framework series. ENWAR 3.0 extends ENWAR 2.0 [4] to support system orchestration, fine-grained sensor reasoning, multi-agent coordination, and long-term memory management. ENWAR 3.0 replaces static heuristics with dynamic agent invocation guided by a deep reinforcement learning (DRL)-trained orchestrator and incorporates memory-assisted reasoning for temporal continuity in decision-making. Unlike conventional fusion architectures that couple modalities at the feature level, ENWAR 3.0 integrates perception, policy selection, and reasoning into a hierarchical control framework. The system combines multi-modal predictors, degradation-aware routing, and structured LLM-based coordination to generate interpretable decisions while satisfying real-time operational constraints in I2V networks.

### A. Main Contributions

We present a framework that extends prior ENWAR capabilities toward closed-loop, degradation-aware network control supplemented by explainable AI (XAI). Building upon the perception-centric pipeline of ENWAR 1.0 [5] and the retrieval-augmented generation (RAG)-based environment perception and beam prediction introduced in ENWAR 2.0 [4], ENWAR 3.0 transitions the framework from perception-assisted decision support to policy-driven, degradation-aware orchestration across beam prediction, blockage detection, and handover control within a unified real-time loop.

While ENWAR 2.0 demonstrated strong beam prediction performance (up to 90.0% Top-3 accuracy) and situation-aware interpretation correctness (up to 89.7%) under a 100ms sampling cycle, its design primarily emphasized reasoning-assisted beam tracking without enforcing a bounded, closed-loop control interval. In contrast, ENWAR 3.0 addresses a broader systems objective: unified beam, blockage, and handover orchestration under degradation-aware routing with strict real-time guarantees. Operating at a 300ms control interval,

ENWAR 3.0 maintains a worst-case end-to-end control-path latency of 289.7ms, ensuring execution within the sampling window without inter-window accumulation. Importantly, this bounded-latency guarantee is achieved while preserving strong beam prediction performance (up to 88.5% Top-3 accuracy) and extending the framework toward deployment-ready, real-time policy-driven wireless control under sensing uncertainty.

The environment perception and beam prediction agents are adopted from ENWAR 2.0 [4], [6], and refined with enhanced preprocessing, expanded temporal context windows, and tighter integration within a hierarchical orchestration layer. Additionally, ENWAR 3.0 integrates a blockage prediction component previously introduced in the conference version [1]. We emphasize that [1] documents only the standalone blockage prediction model; it does not include degradation-aware policy learning, memory-conditioned agent selection, DRL-based routing, or unified beam–blockage–handover control. These capabilities are introduced in ENWAR 3.0. To further clarify the framework’s evolution, Table I summarizes the progression from retrieval-based reasoning (ENWAR 1.0), to multi-modal beam tracking with RAG-based explainability (ENWAR 2.0), to degradation-aware, latency-bounded orchestration with unified control (ENWAR 3.0).

As such, the main contributions are summarized as follows:

- ✓ **Hierarchical Agent Orchestration for Wireless Control:** We reformulate multi-modal wireless control as a hierarchical agent orchestration problem, and propose a modular LLM-driven reasoning architecture that employs chain-of-thought (CoT) priming with reinforcement learning from human feedback (RLHF) and structured prompts to coordinate beam and blockage prediction, and handover decisions within a unified inference loop.
- ✓ **Degradation-Aware Control Layer:** We introduce a synthetic sensor degradation-aware pipeline and a multi-output environment status classifier (99.1% detection accuracy) that enables dynamic exclusion of unreliable modalities during runtime. This replaces static fusion strategies with reliability-aware sensing.
- ✓ **DRL-Based Adaptive Model Routing:** We design a degradation-aware DRL policy that dynamically selects among 15 pretrained modality combinations per agent based on sensor status and historical memory context for adaptive routing rather than fixed model invocation.
- ✓ **Persistence-Aware Blockage and Handover Control:** We integrate blockage duration tracking and historical long-term memory context into a policy-triggered handover mechanism, forming a closed-loop beam–blockage–handover control pipeline that was absent in prior ENWAR iterations.
- ✓ **Real-Time, Bounded-Latency Operation:** Within a 300ms sampling interval, the full system maintains worst-case end-to-end latency (289.7ms), ensuring no temporal accumulation and confirming compatibility with real-time situation-aware I2V beamforming and reasoning.

Across multiple LLM architectures ranging from 3B to 70B parameters, ENWAR 3.0 achieves up to 89.3% reasoning correctness under full modality inclusion within strict latency

TABLE I  
A SUMMARY OF DIFFERENCES BETWEEN EACH ITERATION OF ENWAR

Feature	ENWAR 1.0 [5]	ENWAR 2.0 [4]	ENWAR 3.0
AI Agent Orchestration	×	✓	✓
Beam Prediction	×	✓	✓
Blockage Prediction	×	×	✓
Chain-of-Thought Reasoning	×	✓	✓
DRL Policy Selection	×	×	✓
Focused Feature Extraction	×	×	✓
Environment Classifier	×	×	✓
Handover Management	×	×	✓
Human-in-the-Loop LLM Priming	×	×	✓
Interpretation	✓	✓	✓
Long-Term Memory	×	×	✓
Real-Time Inference	×	×	✓
Situation-Aware Grounding	✓	✓	✓

bounds. The beam and blockage agents achieve over 88% Top-3 accuracy and 98% F1-score, respectively, in ideal conditions, while gracefully degrading under partial sensor failures. These results demonstrate that ENWAR 3.0 advances beyond reasoning-enhanced prediction to deliver adaptive, policy-driven wireless control under real-world sensing uncertainty.

## B. Paper Organizations

Our work hereafter is organized as follows, Section II discuss related works, Section III presents an overview of ENWAR 3.0, Section IV showcases our data preprocessing pipeline, Section V details the LLM priming process, Section VI-A showcases the environment classifier, Sections VII–VIII discuss the agentic structures, Section IX provides the mechanisms on response delivery, Section X showcases our analysis and ablation study of the components of ENWAR 3.0, and Section XI concludes our work.

## II. RELATED WORKS

Recent research <sup>1</sup> has explored integrating LLMs into wireless communication and network control, spanning RAG, question-answer (Q&A) training, instruction tuning, and multi-agent coordination. RAG enhances LLM outputs by retrieving domain-specific knowledge from curated knowledge bases, boosting response precision in telecom contexts [5], [7]–[9]. Meanwhile, Q&A fine-tuning has been applied to telecom domains for spectrum management and protocol understanding, while instruction tuning tailors LLMs to domain tasks but demands large, high-quality datasets and substantial computational overhead [10]. Multi-agent LLM architectures have also supported task decomposition and decision-making in network orchestration [3], [11]–[13].

However, most LLM-based solutions focus on static, query-based interactions and lack real-time decision-making capabilities crucial for dynamic wireless environments. Many remain

<sup>1</sup>App. A shows a tabular summary of related works

fundamentally text-centric and fall short in multi-modal tasks requiring rapid interpretation of sensory data such as LiDAR, radar, and cameras [7], [14]–[16]. Although vision-language models (VLMs) like LLaMa3.2-Vision, LLaVa [17], [18] and cross-modal transformers have shown promise in bridging perception and reasoning [19], practical deployments for real-time wireless optimization remain rare.

Parallel advances in multi-modal sensing have emphasized the benefits of fusing diverse data streams, such as LiDAR, radar, camera, and GPS, to capture fine-grained spatial and temporal patterns essential for tasks like beamforming, blockage prediction, and environment perception [20]–[23]. Notable examples include vision-aided frameworks that leverage RGB images to predict optimal beam indices and blockage events, reducing overhead from exhaustive beam searches [24]–[26], and feature-fusion networks that proactively improve link resilience in vehicle-to-infrastructure (V2I) communications.

Recent work has also combined multi-modal data, semantic retrieval, and reinforcement learning (RL) to enhance efficiency in vehicular and bandwidth-constrained networks [27], [28]. Yet, even these systems often rely on black-box reasoning without transparent and explainable logic [29]–[31].

Furthermore, LLMs struggle with large-scale optimization challenges, such as beamforming or resource allocation, due to the lack of mathematical rigor in purely generative architectures [3], [7], [11], [12], however with sufficient data and defined tasks, LLMs can be fine-tuned for wireless communication and sensing tasks [32]. Another promising alternative includes combining LLM-based semantic understanding with domain-specific and structured optimization solvers [33].

Agentic LLMs show promise in their ability to perform structured, multi-step reasoning and dynamic orchestration, as seen in NetOrchLLM for network management [34] and surveys highlighting how reasoning loops, RAG, CoT and few-shot prompting enhance decision-making in complex domains [35]–[37]. These trends highlight growing momentum toward integrating LLMs as language tools and intelligent orchestrators within multi-modal, multi-agent systems.

Despite these advancements, substantial gaps remain in designing seamless frameworks that blend multi-modal sensing, interpretable reasoning, and adaptive decision-making for wireless control. ENWAR 3.0 directly addresses these gaps with a unified system capable of high-level reasoning and fine-grained environmental adaptation under real-time constraints.

### III. AN OVERVIEW OF ENWAR 3.0

This section outlines ENWAR 3.0’s pipeline at a high level, spanning offline training and priming to real-time inference. We first formalize the I2V communication problem, then describe the integration of multi-agent orchestration, modality-aware perception, and adaptive reasoning in the system.

#### A. Problem Definition

This work considers a mmWave I2V communication scenario composed of three geo-tagged units as seen in Fig. 1. The first unit (Unit 2) is a moving vehicle equipped with a uniform linear array (ULA) of  $M$  antennas and a GPS receiver.

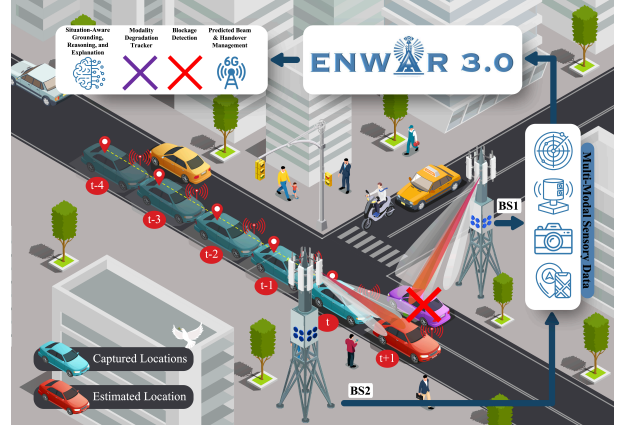


Fig. 1. Illustration of ENWAR 3.0’s system model.

The second and third units are fixed roadside infrastructure elements, each consisting of a single-antenna base station (BS) placed within the communication range of the vehicle. The primary BS (Unit 1) is co-located with a time-synchronized sensor suite comprising a camera, radar, and LiDAR, forming the central RSU. The secondary BS (BS<sub>2</sub>), also geo-tagged and within the communication range of the vehicle and has the same blockage and beam prediction capabilities, supports seamless handover by providing an alternative beamforming anchor point. This multi-BS setup enables the system to manage link blockages and mobility-induced coverage changes via environment- and sensor-aware handover decisions.

At discrete time steps  $t \in \{t_1, t_2, \dots, t_n\}$  spaced by  $T_s$ , the RSU collects synchronized multi-modal sensor data, forming the input to two learning agents: a blockage and beam prediction model. Each input sequence spans five consecutive observations ( $\Delta T$ ) denoted as:

$$\mathcal{X}_t = \{\mathbf{x}_{t-4}, \mathbf{x}_{t-3}, \mathbf{x}_{t-2}, \mathbf{x}_{t-1}, \mathbf{x}_t\}, \quad (1)$$

where  $\mathbf{x}_i \in \mathbb{R}^{C \times H \times W}$  represents the fused features extracted from the multi-modal sensors at time  $i$ . We model the wireless channel and pathloss between Unit 2 and the BSs using the standardized 3GPP Urban Microcell (UMi) model [38], which captures realistic signal propagation in dense urban settings. All relevant simulation parameters, including pathloss coefficients and scenario configurations, are detailed in Section X-B.

**Blockage Prediction:** The blockage prediction agent estimates the likelihood of occlusion events—caused by large vehicles, pedestrians, or infrastructure—occurring up to  $k$  steps into the future. The model outputs a vector of predicted probabilities:

$$\mathbf{p}_t = [p_{t+1}, p_{t+2}, \dots, p_{t+k}], \quad \text{where } p_{t+i} \in [0, 1], \quad (2)$$

with each  $p_{t+i}$  indicating the likelihood of a blockage event at time  $t+i$  given past sensor observations  $\mathcal{X}_t$ . The model is trained as a binary classifier using ground-truth labels  $y_{t+i} \in \{0, 1\}$ , and defined by a deep fusion architecture:

$$\mathbf{p}_t = f_\theta(\mathcal{X}_t), \quad (3)$$

where  $f_\theta$  denotes the parameterized model that captures multi-modal spatiotemporal dependencies for blockage prediction.

**Beam Prediction:** Let  $\mathbf{h}(t) \in \mathbb{C}^{M \times 1}$  denote the channel between the vehicle and the RSU at time  $t$ . The transmitter employs a beamforming codebook  $\mathcal{F} = \{\mathbf{f}_i\}_{i=1}^Q$ , where  $\mathbf{f}_i \in \mathbb{C}^{M \times 1}$  is the  $i$ th codeword and  $Q = OM = |\mathcal{F}|$  defines the oversampled codebook size. The received signal is therefore:

$$y(t) = \mathbf{f}_{\iota(t)}^H \mathbf{h}(t) x(t) + n(t), \quad (4)$$

where  $\iota(t)$  is the selected beam index,  $x(t) \in \mathbb{C}$  is the transmitted symbol, and  $n(t) \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^2)$  is complex Gaussian noise. The optimal beam index maximizes received power:

$$\iota^*(t) = \underset{i \in [1, Q]}{\operatorname{argmax}} |\mathbf{f}_i^H \mathbf{h}(t)|^2. \quad (5)$$

However, exhaustive beam sweeping is computationally intensive in fast-changing environments. To overcome this challenge, a beam prediction model is trained to anticipate the optimal beam  $\iota^*(t+k)$  based on recent sensor trajectories  $\mathcal{X}_t$ .

### B. ENWAR 3.0's Flow

As illustrated in Fig. 2, ENWAR 3.0 begins with step **(I<sub>a</sub>)**: real-time inputs from camera, GPS, LiDAR, and radar are acquired and processed via a unified preprocessing module in step **(I<sub>b</sub>)** for normalization, filtering, and resizing. The pipeline then splits into two coupled tracks: an offline priming stage and an online inference path, unified through a central LLM.

The offline priming pipeline in step **(0)** prepares the LLM using CoT prompting, few-shot learning, and human-in-the-loop feedback. Across multiple sessions, the LLM is exposed to diverse sensing conditions and response expectations. A reward model scores outputs, and iteratively refines prompt-response alignment. This produces a primed LLM capable of consistent multi-step reasoning without reliance on heavy prompt templating at inference time.

The online path begins at step **(1)** with an environment classifier that detects per-modality degradation to relay to downstream agents. In parallel, an image-to-text module produces periodic high-level summaries to contextualize decision-making. In step **(2)**, incoming inputs are framed within a structured instruction template. The primed LLM interprets this context and determines whether agent invocation is required.

If needed, step **(3)** activates the agent manager, which orchestrates task-specific agents (e.g., beam and blockage prediction, handover, and environment perception) based on sensor status and task context. In parallel with structured input ingestion, two complementary modules enrich the agent selection process:

- The long-term memory module (step **(3a)**) uses LlamaIndex's fact extraction to persist meaningful patterns like modality degradation trends, blockage durations, and past agent decisions to enable context-aware reasoning and facilitate historical recall in future inferences.
- The DRL policy module (step **(3b)**) guides agent configuration using a proximal policy optimization (PPO)-trained policy [39], which prioritizes accurate, low-latency agent configurations while avoiding degraded modalities.

The long-term memory and DRL policy modules operate concurrently with input processing and relay outputs to the

agent manager. The LLM combines this data with contextual priors and sensor cues for agents selection. Crucially, agent execution and network updates occur independently of full natural-language response generation, ensuring that time-critical decisions remain bounded within the sampling interval.

Once agent outputs are obtained, they are returned to the LLM (step **(7)**), which conducts multi-step reasoning and may re-engage additional agents as needed to complete the inference loop. Finally, step **(8)** executes a concluding LLM reasoning pass, generating a structured and interpretable response that integrates perception, prediction, and control for robust I2V orchestration.

## IV. DATA PREPROCESSING

We utilize real-world multi-modal data from the DeepSense6G dataset [40], [41] to support beam and blockage prediction. This dataset comprises 18,667 samples with synchronized sensor streams of an I2V environment. We utilize Scenarios 31–34, which provide time-aligned measurements such as adjusted GPS traces from Units 1 and 2, high-resolution RGB images, LiDAR-based 3D point clouds, and radar signatures. Each training instance is constructed from five temporally ordered multi-modal samples to capture short-term spatiotemporal dynamics.

### A. Data Augmentation

To improve generalization, modality-specific augmentations were applied during training. Images were randomly flipped, rotated, and blurred to emulate viewpoint and lighting variations. LiDAR point clouds underwent random flips, rotations, and spatial scaling to increase geometric diversity. Radar inputs were perturbed with Gaussian noise to model measurement uncertainty.

### B. Data Labeling

Each sample is labeled at time  $t+1$ . Beam prediction uses the beam index with maximum received power as ground truth. Blockage labels are assigned when a sustained received signal strength indicator (RSSI) drop aligns with visually confirmed obstruction events on Unit 2.

### C. Blockage Class Imbalance Handling

The blockage classification task exhibits significant class imbalance, with 15,050 non-blocked and 3,617 blocked samples. To reduce bias toward the majority class, we employ a class-weighted loss that emphasizes blocked instances:

$$w_{\text{pos}} = \alpha \cdot \left( \frac{N_{\text{non-blocked}}}{N_{\text{blocked}}} \right), \quad (6)$$

where  $N_{\text{non-blocked}}$  and  $N_{\text{blocked}}$  denote the number of samples in each class, and  $\alpha$  is a scaling factor that modulates the weighting intensity.

### D. Modality Preprocessing

1) *Image Preprocessing:* RGB images are resized to  $256 \times 256$  and normalized prior to feature extraction.

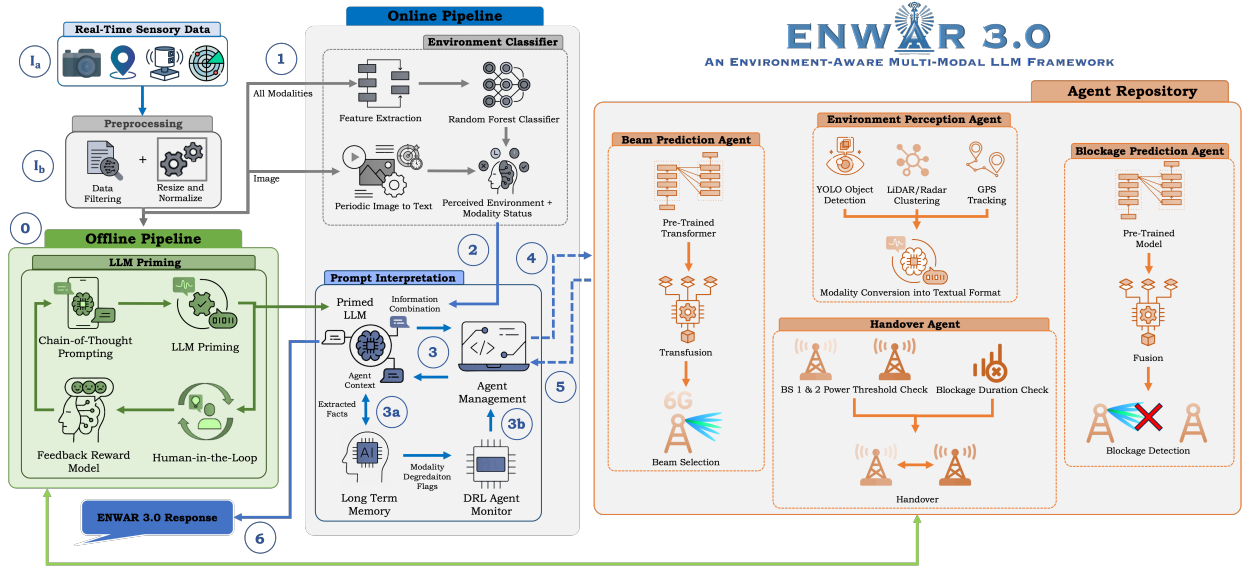


Fig. 2. ENWAR 3.0’s flow pipeline starting with preprocessing multi-modal inputs, LLM priming in the offline pipeline, and utilizing the primed LLM to process a detailed network perception and enhancement task in the online pipeline through an environment classifier, and available agents in the repository.

2) *GPS Preprocessing*: We derive motion features from the GPS trajectory by computing first- and second-order temporal derivatives, capturing vehicle displacement, velocity, acceleration, angular velocity, and curvature. These descriptors form an 18-dimensional vector and are min-max normalized using a pre-fitted training scaler to ensure consistency across samples.

3) *LiDAR Preprocessing*: LiDAR point clouds first undergo voxel grid downsampling to reduce point density while maintaining the underlying spatial geometry followed by ground plane extraction via random sample consensus (RANSAC) [42], [43]. As a noise filter, a statistical outlier filter computes the mean distance,  $\epsilon$ , of each point to its  $k$  nearest neighbors and the minimum number of samples to construct an identifiable cluster. Points whose distances deviate from the local neighborhood distribution are discarded based on a threshold defined by the dataset’s standard deviation. Remaining above-ground points are clustered using density-based spatial clustering of applications with noise (DBSCAN) ( $\epsilon = 0.75$ ,  $\text{min\_samples} = 5$ ), and small clusters are discarded.

The cleaned 3D point cloud is projected into a bird’s-eye view (BEV) grid over  $X, Y \in [-50, 50]$ m and  $Z \in [-2.5, 15]$ m at 0.25m resolution. Each BEV cell encodes: (i) maximum height, (ii) log-scaled point count (density), and (iii) height variance. Temporal sequences are concatenated to yield a (700, 1200, 15) tensor. Each channel is individually normalized: height via min-max scaling, density via log transform, and variance via global standardization.

4) *Radar Preprocessing*: Radar inputs comprise complex-valued tensors of shape (4, 256, 250) across four virtual antennas. We decompose signals into magnitude and phase, and apply 1D fast Fourier transform (FFT) along the Doppler axis to extract motion-related spectral features.

We augment these raw features with statistical descriptors: per-channel mean, standard deviation, entropy (for scene complexity), and Doppler-based motion metrics such as mean velocity and spectral spread. All feature maps are resized

to  $(256 \times 64)$  via padding or trimming, and single-value descriptors are broadcast as needed. The final radar input is an  $(8, 256, 64)$  tensor capturing spatial, spectral, and dynamic cues relevant for downstream inference.

## V. LLM PRIMING

To enable structured multi-agent orchestration in ENWAR 3.0, we prime a centralized LLM in an offline pre-deployment phase. The objective of priming is to train the model to interpret structured environment states, invoke appropriate agents exactly once, and generate policy-consistent justifications aligned with sensor reliability and DRL-recommended actions. Few-shot examples follow a fixed prompt schema (App. B-A) that includes fields such as *Environment Status*, *Trajectory*, *Blockage Status*, and *Predicted Beam Properties*, ensuring a consistent reasoning structure.

Priming is guided through a lightweight human-in-the-loop reward mechanism inspired by RLHF. Each structured input state  $s_p$  consists of multi-modal environmental features (e.g., object presence, sensor degradation levels), long-term memory context, and DRL-suggested agent selections. The LLM generates an action  $a_p$  consisting of selected agent calls and structured justification. Human evaluators then score each  $a_p$  using a fixed rubric along three axes: (i) *Correctness*, based on valid beam, blockage, and handover decisions; (ii) *Agentic Justification*, assessing the alignment between DRL output and modality health; and (iii) *Explanation Clarity*, ensuring responses are well-structured, complete, and non-redundant. These criteria are aggregated into a scalar reward  $R_p(a_p) \in [0, 10]$ . The reward function  $R_{\text{priming}, \theta}(s_p, a_p)$  is parameterized by rule-based scoring parameters  $\theta$ , ensuring evaluation consistency rather than learned reward modeling. If responses misuse agents, violate policy alignment, or exhibit structural redundancy, evaluators provide corrective feedback, and the prompt is iteratively refined.

This process defines a reward-guided prompt optimization loop over the induced LLM policy  $\pi(a_p | s_p)$ :

$$\theta \leftarrow \theta + \nabla_{\theta} \mathbb{E}_{a_p \sim \pi(\cdot | s_p)} [R_{\text{priming}, \theta}(s_p, a_p)], \quad (7)$$

where  $\theta$  governs rubric parameters only; the LLM weights remain unchanged. Instead of gradient-based fine-tuning, the model internalizes structured reasoning patterns through iterative CoT prompting and feedback until responses satisfy  $R_p(a_p) > \tau_{\text{reward}}$ . The corresponding pseudocode is provided in App. B-B.

At deployment, the primed LLM receives real-time sensory inputs and DRL outputs, determines whether agents are required, executes the necessary calls, and generates a structured decision report. For example, if radar is flagged as degraded and excluded by the DRL policy, the LLM must justify its omission and avoid redundant invocations. This behavior emerges from the structured reward-guided priming process.

The priming mechanism enforces policy-consistent, degradation-aware, and non-redundant agent orchestration under dynamic I2V conditions. An example of the priming loop is provided in App. B-C and summarized below.

---

**Prompt 1:** Perceive the environment using the available modalities based on your sense of the environment status. Which beam and blockage models should be selected and justify your decision.

---

**Summarized Response 1:**

- Modality Status: Radar degraded (42.1%), all others reliable.
  - Environment: Urban, 3 vehicles near stop sign.
  - DRL Agent Recommendation: Use camera-gps-lidar for beam, camera-only for blockage
  - Select camera-only for blockage due to radar degradation.
  - Start handover to BS2.
  - Invoked blockage prediction and handover agents twice, beam prediction and perception agents once.
- 

**Prompt 2:** Going forward, focus more on the detailed justifications of why handover is necessary, and do not repeat agent invocations. Score: 2.4

---

**Summarized Response 2:** ...

Handover Status: The blockage condition threshold has been flagged and BS2 has a higher power than Unit 1, therefore initiate handover. Invoked all agents once.

---

## VI. PERCEIVED ENVIRONMENT STATUS

In multi-modal sensor-equipped environments, sensors can degrade due to noise, weather, and obstructions. Early environment classification is crucial for decision-making. ENWAR 3.0 features an environment classifier and an image-to-text module that periodically converts images into a generalized environment description. This section details these modules.

### A. Environment Classifier

ENWAR 3.0 includes a dedicated environment status classifier that outputs four binary flags indicating whether each

modality (camera, GPS, LiDAR, radar) is degraded to support adaptive model selection. These real-time flags guide downstream agents to rely only on trustworthy sensors, improving robustness under dynamic conditions.

We developed a synthetic degradation pipeline because the original dataset lacked extensive coverage of adverse scenarios. Camera frames were blurred, oversaturated, or darkened to simulate fog, rain, or night. LiDAR point clouds were sparsified to mimic occlusion, radar tensors corrupted with Gaussian noise, and GPS traces jittered to emulate jamming or spoofing. Each modality was degraded independently and probabilistically to create diverse sensor failure states. In deployment, this classifier may be augmented with weather metadata (e.g., visibility, time of day) via real-time APIs.

The classifier uses a two-stage pipeline: handcrafted feature extraction and multi-output classification. Extracted features include blur entropy and brightness stats (camera), point dispersion and density (LiDAR), spectral entropy and SNR (radar), and variance and displacement (GPS). These are concatenated, normalized, and passed to a multi-output Random Forest with 100 estimators and max depth 10. Each output node predicts whether a modality is degraded.

Using modality-specific degradation metrics, a modality is flagged as degraded if  $> \tau_{\text{degradation}}$  of its signal is impaired over five consecutive time steps. These scores are smoothed over time to avoid reacting to transient noise. Degraded modalities are temporarily excluded from model selection until their scores drop below the threshold, such that ENWAR 3.0 relies only on high-confidence inputs.

### B. Periodic Image-to-Text Environment Perception

ENWAR 3.0 includes a low-frequency environment perception module that converts visual inputs into structured textual summaries. At periodic intervals (e.g., hourly), a representative camera frame is processed by a vision-language model to generate concise scene descriptors (e.g., “*urban intersection with moderate traffic and clear weather*”).

These summaries capture slowly varying environmental attributes such as lighting, weather, and scene type, providing high-level contextual information beyond instantaneous sensor measurements. Since these attributes evolve gradually, the module operates independently of the real-time inference loop, preserving low latency while providing stable contextual cues to downstream reasoning agents.

## VII. AGENT MANAGEMENT

The agent management layer mediates interaction between the centralized LLM and task-specific agents. When a subtask associated with a specialized agent is required, the LLM issues a structured JSON request to the manager, which selects the appropriate agent based on task type and current sensor availability as determined by the environment classifier.

Agents are configured for different degradation profiles. For instance, if the camera is flagged as unreliable, camera-dependent models are bypassed in favor of radar- or LiDAR-based alternatives. This routing mechanism ensures inference relies on the most reliable modalities.

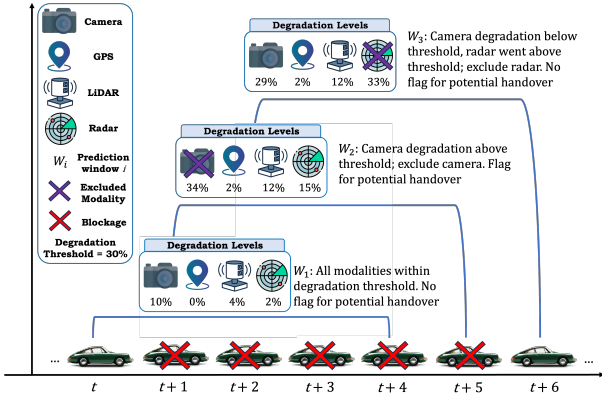


Fig. 3. Three consecutive input windows illustrating updates to blockage and modality degradation flags. Window 1: all modalities are healthy. Window 2: camera degraded and persistent blockage detected. Window 3: radar degraded while camera recovers.

The manager is supported by a **memory module** and a **DRL policy** (Sections VII-A–VII-B), which incorporate historical context, sensor degradation, and blockage persistence into agent selection. Agent outputs are returned to the LLM as structured semantic blocks, which are intergated into the final LLM reasoning chain’s agent selection.

#### A. Long-Term Memory and Sensor Context Tracking

The long-term memory module in ENWAR 3.0 serves as a persistent, queryable repository for encoding sensor reliability patterns and agent decision outcomes over time. It complements the real-time behavior of the DRL-based agent selection module by supplying historical context, enabling the system to reason reactively with temporal foresight.

This module tracks two critical forms of long-range metadata: agent context histories and modality degradation. Agent contexts include past beam and blockage model selections, blockage durations, environment labels and conditions, and associated DRL rewards. These logs enable temporal introspection and facilitate the identification of effective model configurations under recurring environmental states.

In parallel, the system maintains degradation metrics by computing a running average of the modality classifier’s per-frame binary degradation flags output. If a sensor has a degradation of  $> \tau_{degradation}$  of the observed frames per an inferable input window ( $\geq t + 5$  time steps), it is classified as persistently degraded. This statistical visibility score is then used to suppress agent configurations that rely on low-performing sensors, guiding both the LLM and PPO agents away from suboptimal decisions. An example of the degradation tracker across three input windows is seen in Fig. 3.

Similarly, long-term memory also tracks blockage durations. Blockage durations are extracted from the agent contexts. If a blockage persists for more than  $t + 5$  time steps, then a flag is set to indicate that a handover condition is satisfied. This mechanism is seen in Fig. 3, where three inferred windows exist. The first and last windows show blockages occurring for less than  $t + 5$  time steps. As such, the flag for potential handover due to prolonged blockages is not set, and the second window satisfies the conditions and has the flag set to true.

The long-term memory is implemented using the *LlamaIndex* framework through its extensible memory block architecture. Specifically, two memory block types are employed: *fact extraction memory block*, which distills agent-environment interactions into structured facts (e.g., “Radar blockage agent succeeded in foggy environment”), and *static memory block*, which stores known modality degradation profiles and policy-derived insights. When invoked, the LLM retrieves relevant memory blocks, enabling it to condition its reasoning on prior successes, failure cases, or modality trends.

This indexed memory enables symbolic and semantic search across historical interactions, improving few-shot generalization and reducing reasoning drift. Moreover, the LLM’s agent evaluator can cross-reference long-term memory entries with live predictions to ensure consistent and trustworthy model selection during dynamic deployments. The long-term memory module thus transforms ENWAR 3.0 from a reactive agent system into a historically informed orchestration framework.

#### B. Policy-Driven Agent Selection

To support robust and adaptive agent selection under dynamic sensing conditions, ENWAR 3.0 employs a DRL policy based on the PPO algorithm. This policy selects the most suitable combination of modalities from a repository of pretrained beam and blockage prediction models, depending on current sensor health and environmental context.

Unlike rule-based systems, this learned policy adaptively routes tasks to one of various pretrained model variants per agent. To support robust agent selection under varying sensor conditions, we design 15 unique pretrained model variants per agent. Each variant corresponds to a distinct combination of sensor modalities, camera, LiDAR, radar, GPS, or combinations thereof, allowing for granular alignment between model architecture and available, non-degraded sensor inputs. The action space,  $A_{DRL}$  represents a set of actions,  $a_{DRL}$ , that outputs a multi-discrete tuple of integers representing the selected pretrained modality combination for beam and blockage prediction models based on the perceived environment and modality degradation status,  $s_{DRL}$ , as extracted from the long-term memory block [c.f. Section VII-A] and the environment classifier [c.f. Section VI-A].

The DRL agent is trained using PPO, which balances learning stability with exploration efficiency. PPO is a policy gradient method that updates the policy network in a way that discourages large policy shifts, ensuring smoother convergence and reduced risk of catastrophic forgetting. We adopt a multi-layer perceptron (MLP) policy with a 2-layer feedforward network to map the observed state of the environment, including current degradation conditions and memory features, to a discrete action vector. This vector selects which pretrained agent variants to invoke for beam and blockage prediction tasks. Each training batch consists of 1024 time steps, subdivided into 64-sample mini-batches. The policy is optimized using a clipped objective with a learning rate of  $3 \times 10^{-4}$ , a discount factor,  $\gamma = 0.95$ , and a small entropy bonus (0.01) to encourage exploratory behavior early in training.

The policy is trained in a synthetic environment where 30% of training samples are probabilistically degraded using the

sensor perturbation pipeline [c.f. Section VI-A]. This sampling ensures the agent learns robust selection behavior across a diverse distribution of partial sensor failures. Observations are encoded as a 4-dimensional binary vector representing the degraded status of each modality. The PPO algorithm, an actor-critic framework, optimizes the policy by balancing exploration and learning stability using clipped surrogate objectives. The agent is trained to map these binary degradation states to optimal model pair selections that maximize reward.

The reward function,  $R_{\text{DRL}}(s_{\text{DRL}}, a_{\text{DRL}})$  is designed to prioritize model performance and sensor reliability while penalizing computational inefficiency. The reward for each action is:

$$R_{\text{DRL}}(s_{\text{DRL}}, a_{\text{DRL}}) = 0.9(A_{\text{beam}} + A_{\text{block}}) - 0.1(T_{\text{beam}} - T_{\text{block}}), \quad (8)$$

where  $A_{\text{beam}}$  and  $A_{\text{block}}$  denote the normalized performance scores (accuracy, F1-score) of the selected models, weighted by their reliance on degraded sensors.  $T_{\text{beam}}$  and  $T_{\text{block}}$  represent normalized inference latency penalties. The weighting scheme is designed to balance predictive performance with computational efficiency while reflecting the different scales of the two objectives. Performance metrics are naturally bounded between 0-100%, making them stable and directly comparable across models. Inference latency, by contrast, spans a broader and less uniform range, often varying by an order of magnitude across configurations (13-67ms). To prevent these larger latency variations from dominating the reward, accuracy, and reliability are assigned higher weight as the primary reward target, while inference time contributes as a secondary penalty. This formulation ensures that the DRL agent consistently selects models that maintain high predictive quality, yet still accounts for efficient real-time deployment.

At inference time, the trained DRL agent receives real-time degradation flags from the environment classifier and memory module, and selects the most context-appropriate model pair for that time window. This adaptive routing mechanism allows ENWAR 3.0 to intelligently respond to sensor failures and environmental changes without retraining or manual intervention.

## VIII. AGENT REPOSITORY

ENWAR 3.0 hosts a modular agent repository for environment perception, beam and blockage prediction, and handover management. Details of the perception and beam agents are provided in Apps. C–D and in ENWAR 2.0 [1], [4], with minor modifications to support BS<sub>2</sub> integration. Below, we detail each agent’s design and their role in ENWAR 3.0’s pipeline.

### A. Blockage Prediction Agent

The blockage prediction agent is designed to anticipate future communication blockages based on environmental perception and vehicle dynamics. Multi-modal inputs from  $t$  time steps are forwarded to four modality-specific encoders, tailored to extract blockage-related spatiotemporal features. A late fusion strategy is then used to aggregate modality-level predictions into a final decision, allowing the system to adaptively prioritize sensor modalities based on reliability and validation performance. The model is trained using binary

cross-entropy loss with logits for stable convergence. A high-level representation of the model architecture is shown in Fig. 4, and App. E fully details each component.

1) *Camera-Based Blockage Prediction*: The camera-based model captures spatial and temporal features indicative of dynamic obstructions. A ResNet-18 backbone, pretrained on ImageNet and truncated before the classification layer, encodes each frame into a high-dimensional embedding. These embeddings are fed into a single-layer long short-term memory (LSTM) with 128 hidden units to capture temporal dependencies such as object motion and visual occlusions. The final hidden state is passed through a two-layer fully connected classification head with ReLU activation and dropout ( $\rho = 0.4$ ), producing a single logit that reflects the blockage probability.

2) *GPS-Based Blockage Prediction*: The GPS-based model focuses on identifying mobility patterns correlated with blockage scenarios, such as sudden deceleration or sharp turns. We extract 18 features from each sequence, including displacement, velocity, acceleration, angular change, and curvature. These features are normalized and passed as a sequence to a two-layer LSTM with 128 hidden units. The final hidden state is projected through a fully connected layer (64 hidden units) with ReLU activation, dropout, and a binary output neuron.

3) *LiDAR-Based Blockage Prediction*: The LiDAR-based model processes temporally stacked BEV projections of point cloud data. This preprocessed sequence consists of five consecutive BEV frames, where each frame includes three spatial feature channels (height, density, variance), resulting in a 15-channel input tensor. To handle this high-dimensional input, we adapt a ResNet-18 backbone by modifying the first convolutional layer to accept 15 input channels. The network processes the stacked frames through residual blocks, learning rich spatial features tied to object boundaries and environmental structure. Unlike other modalities, temporal information is encoded implicitly through the stacked frame. The final output logit is given from a fully connected layer.

4) *Radar-Based Blockage Prediction*: The radar-based model utilizes spatial motion signatures and temporal dynamics of radar reflections. Each radar frame is an 8-channel tensor of size  $256 \times 64$ , encoding magnitude, phase, Doppler, and frequency features. These frames are passed through three 2D convolutional layers with ReLU activations and batch normalization to capture spatial features. An adaptive average pooling layer reduces each frame to a fixed-length vector, and the resulting sequence is processed by a single-layer LSTM with 64 hidden units. The final hidden state is passed through a two-layer classification head with ReLU and dropout ( $\rho = 0.3$ ), producing a binary prediction logit.

5) *Multi-Modal Fusion*: We adopt a late fusion approach that aggregates modality-specific outputs at the decision level to integrate predictions across each modality. Each model independently outputs a probability score of the likelihood of a future blockage:  $\mathcal{P}_{\text{Camera}}, \mathcal{P}_{\text{GPS}}, \mathcal{P}_{\text{LiDAR}}, \mathcal{P}_{\text{Radar}}$ . These scores are combined using a weighted averaging scheme:

$$\mathcal{P}_{\text{fused}} = \sum_{i \in \{\text{Camera}, \text{GPS}, \text{LiDAR}, \text{Radar}\}} w_i \cdot \mathcal{P}_i. \quad (9)$$

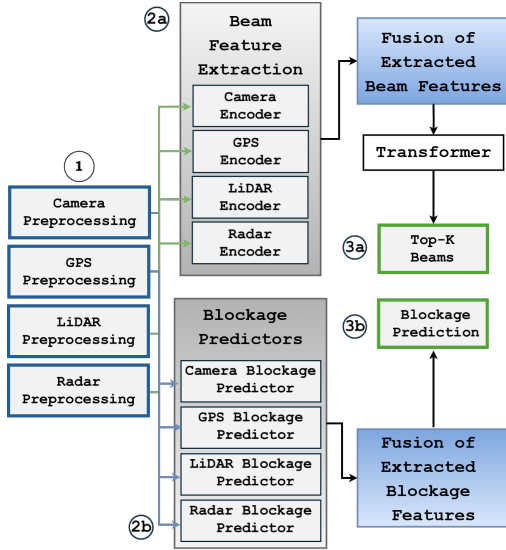


Fig. 4. Simplified architecture of the beam and blockage prediction models, where (1) shows data preprocessing, (2a-b) resemble the beam and blockage prediction flow, and (3a-b) are the model outputs.

Fusion weights,  $w_i$ , are derived from a running performance estimate that combines (i) softmax-normalized F1-scores computed on a held-out validation set and (ii) the modality preference suggested by the DRL policy. Let  $\mathbf{s} = [s_{\text{Camera}}, s_{\text{GPS}}, s_{\text{LiDAR}}, s_{\text{Radar}}]$  denote the validation F1-score vector. The weights are then given by:

$$\mathbf{w} = \text{softmax}(\mathbf{s}) = \frac{\exp(s_i)}{\sum_j \exp(s_j)}. \quad (10)$$

The final output predicts blockages for up to  $t+5$  time steps (1.5s). This formulation weighs modalities based on validated performance while preserving multi-sensor contributions, thus enhancing robustness under dynamic sensing conditions.

### B. Handover Agent and Power-Aware Decision Policy

To support proactive link maintenance in dynamic vehicular scenarios, a dedicated handover agent governs the transition between multiple BSs. The handover logic is informed by predicted link blockages and real-time power differentials.

The decision process follows a dual-trigger policy. Let  $P_1(t)$  and  $P_2(t)$  denote the estimated received powers from Unit 1 and BS<sub>2</sub> at time  $t$ . A handover from Unit 1 to BS<sub>2</sub> is initiated if either of the following conditions hold for  $\geq 5$  consecutive time steps:

- 1) **Power Superiority Trigger:** BS<sub>2</sub> offers a significantly stronger link:

$$P_2(t) - P_1(t) \geq \Delta_{\text{power}} \quad (11)$$

where  $\Delta_{\text{power}}$  is a tunable threshold.

- 2) **Blockage-Aware Trigger:** A blockage is predicted on Unit 1's link, and BS<sub>2</sub> offers any improvement in received power:

$$\text{Blockage}_t = 1 \quad \text{and} \quad P_2(t) > P_1(t) \quad (12)$$

If blockage persists beyond  $t+5$  time steps (i.e., one input window), the agent performs an explicit BS evaluation and

generates a structured output including: (1) selected target BS ID, (2) expected power gain, and (3) a confidence score. This output is passed to the LLM for the final handover decision coordination. Temporal blockage tracking and persistence logic are handled within the memory module [c.f. Section VII-A].

## IX. RESPONSE DELIVERY

When agentic contexts are complete and processed, the LLM delivers a response with situation-aware, grounded reasoning. This section details the response delivery steps.

### A. Prompt Processing

When the primed LLM receives structured data derived from real-time multi-modal inputs, prompt processing begins. These inputs include environmental descriptors, sensor degradation flags, agent outputs, and contextual metadata. The primed LLM is fully equipped to perform structured, context-aware inference across its core tasks: beam and blockage analysis, environment interpretation, and handover decision-making.

The prompt schema ensures that all inputs are aligned with the LLM's trained reasoning format. For instance, visual and spatial cues from degraded modalities are summarized as natural language descriptors (e.g., "LiDAR is occluded, camera visibility is 40%"), while structured metadata (e.g., BS power levels, blockage duration) is appended as a JSON request. This design ensures the LLM engages in interpretable, deterministic reasoning aligned with the intended policy logic.

Upon receiving structured inputs, the LLM either performs direct multi-step CoT inference or delegates tasks to specialized agents. Tasks requiring explicit algorithmic processing, such as handover evaluation, beam prediction, or sensor-based perception, are routed through the **agent management** layer.

### B. Response Generation

With all agent outputs received, the LLM integrates its reasoning with agent contexts to generate a complete, structured response. Each field in the response is populated based on a combination of sensor data, agent feedback, and contextual understanding embedded during LLM priming. This final step highlights ENWAR 3.0's ability for situational reasoning as it connects agent outputs with qualitative justifications.

Fig. 5 showcases an optimal response to a blockage and handover event, illustrating the framework's multi-agent reasoning. The response includes the predicted beam and its properties, perceived environmental context, rationale for modality inclusion/exclusion based on degradation, Unit 2's trajectory, and a detailed handover justification, with key performance indicator (KPI) metrics for that inference. For illustration, ground truth beam data is shown, but unavailable during deployment. The accompanying prompt template is used during pre-deployment priming; once deployed, the LLM no longer relies on this explicit format, as it internalizes the reasoning patterns and task logic required to operate autonomously. The response showcases ENWAR 3.0's appropriate agent invocations, and situation-aware grounding and reasoning.

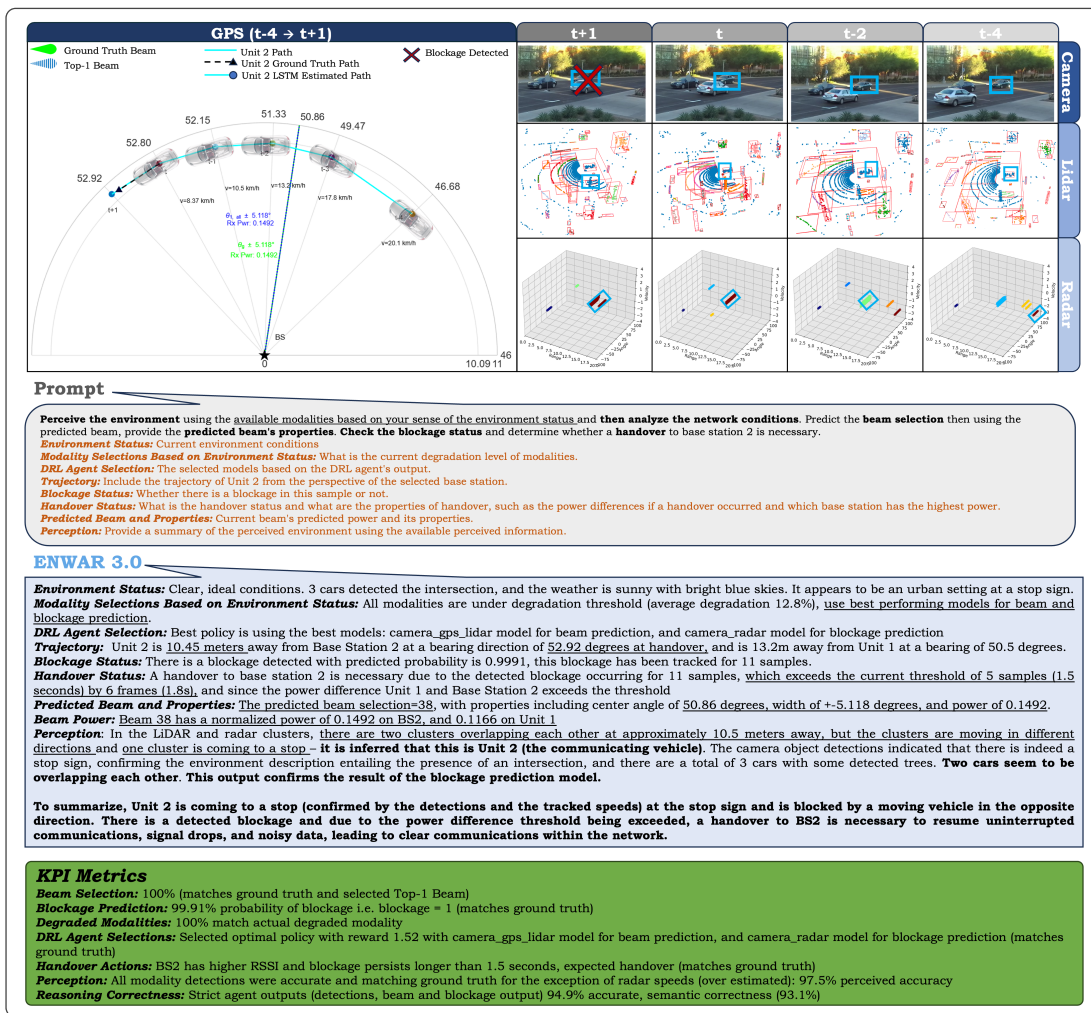


Fig. 5. An example of ENWAR 3.0's response to a detected blockage and handover scenario.

ENWAR 3.0 operates under dynamic handover conditions and varying levels of sensor degradation through degradation-aware policy routing. Upon detecting modality impairment, the system invokes a DRL-trained agent to select the appropriate modality configuration for beam and blockage prediction based on the current environmental context.

When sensing quality deteriorates, inference pipelines are adaptively rerouted using degradation flags and long-term memory signals, preserving temporal consistency in decision-making. The resulting outputs include structured justifications for beam selection, blockage forecasting, and handover triggers. Illustrative inference examples are provided in App. F.

## X. EVALUATION OF ENWAR 3.0

This section evaluates ENWAR 3.0 using agent-level KPIs and degradation-aware orchestration metrics. Individual agents are assessed across multiple sensor configurations, while the orchestration layer is evaluated based on model selection accuracy and justification consistency. The subsections below detail the evaluation methods and their results.

### A. Enwar 3.0 Setup

The evaluation of ENWAR 3.0 was conducted on an NVIDIA A100 GPU (40GB VRAM). To analyze the impact

of model scale on reasoning behavior under full perception access (i.e., access to all sensor modalities), we evaluated multiple LLMs, including: LLaMa 3.1-8B, LLaMa 3.3-70B, DeepSeek-v1.5-R1-8B, DeepSeek-v1.5-R1-32B, DeepSeek-v1.5-R1-70B, Qwen 2.5-3B, and Qwen 3-32B. Runtime memory usage scaled with model size (3B: 3.4GB; 8B: 16GB; 32B: 22GB; 70B: 32GB), with all models supporting up to 128k token contexts for multi-turn CoT reasoning. Unless otherwise noted, all core experiments and benchmark evaluations were conducted using LLaMa 3.2-3B, due to its efficient runtime characteristics and minimal deployment overhead.

Agent memory allocation varied by modality configuration. The largest blockage predictor required 1.7GB, and the largest beam predictor used 3.5GB. The vision-language perception module (LLaMa 3.2-Vision-11B) required 8GB, and combined perception agents required 1.07GB. Agents were dynamically activated to manage runtime budgets.

To evaluate the beam and blockage prediction agents' performance, we defined 15 unordered and distinct sensor combinations. Each combination was profiled regarding accuracy, inference latency, and preprocessing overhead. These combinations formed the action space of the DRL training environment used to assess policy-driven modality selection

under varying degradation states.

When evaluating the primed LLM, we conducted 10 independent priming sessions, each involving up to 20 iterations of human-in-the-loop feedback. On average, convergence to the predefined reward threshold occurred within four prompt cycles per session, indicating stability in structured multi-agent reasoning patterns rather than prompt-specific behavior.

Throughout priming, human reviewers evaluated each generated response for alignment with desired content structure, correctness of invoked agents, and the depth of contextual reasoning. Feedback was translated into reward signals that guided the LLM toward increasingly precise and context-aware outputs. This iterative refinement ensured that the primed LLM reliably produced detailed, task-specific reasoning aligned with the system's real-time decision-making requirements.

### B. Simulation Parameters

Real-time inputs consisted of five multi-modal samples spaced by  $T_s = 300\text{ms}$ , forming an input window of  $\Delta T = 1.5\text{s}$ . For blockage prediction, the class-imbalance parameter in (6) was set to  $\alpha = 1.1$  to modestly upweight blocked samples. During priming, the reward acceptance threshold was set to  $\tau_{\text{reward}} = 8.5$ .

Received power is estimated using a geometric and physics-informed model. Unit 2's bearing to  $\text{BS}_i$  is computed as:

$$\begin{aligned} \phi_1 &= \text{radians}(\text{lat}_1), & \phi_2 &= \text{radians}(\text{lat}_2) \\ \Delta\lambda &= \text{radians}(\text{lon}_2 - \text{lon}_1) \\ x &= \sin(\Delta\lambda) \cdot \cos(\phi_2) \\ y &= \cos(\phi_1) \cdot \sin(\phi_2) - \sin(\phi_1) \cdot \cos(\phi_2) \cdot \cos(\Delta\lambda) \\ \theta &= \text{atan2}(x, y) \\ \theta_{\text{deg}} &= \text{degrees}(\theta), \end{aligned} \quad (13)$$

where  $(\text{lat}_1, \text{lon}_1)$  and  $(\text{lat}_2, \text{lon}_2)$  represent the latitudes and longitudes of Unit 2 and  $\text{BS}_i$ , respectively.

Distances between Unit 2 and each BS are computed as the great-circle distance,  $d$ , between two points on a sphere given their latitudes and longitudes. This distance is calculated using the Haversine formula as follows:

$$\begin{aligned} a &= \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \\ c &= 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \\ d &= R \cdot c \end{aligned} \quad (14)$$

where  $R=6371000$  meters is the Earth's radius.

These distances are then mapped to path loss values using the 3GPP Urban Microcell (UMi) model [38], with separate formulations for line-of-sight (LoS) and non-line-of-sight (NLoS) conditions:

$$\text{LoS: } P_{\text{loss}}(d) = 32.4 + 21 \log_{10}(d) + 20 \log_{10}(f_{\text{GHz}}), \quad (15)$$

$$\begin{aligned} \text{NLoS: } P_{\text{loss}}(d) &= 22.4 + 35.3 \log_{10}(d) + \\ &21.3 \log_{10}(f_{\text{GHz}}) - 0.3(h_{\text{UE}} - 1.5). \end{aligned} \quad (16)$$

TABLE II  
HYPERPARAMETERS USED IN AGENT TRAINING AND LLM DEPLOYMENT

Component	Hyperparameter	Value
Beam Prediction	Batch Size	6
	Epochs	75
	Epoch of Convergence	34
	Learning Rate	0.0001
	Number of Samples	18,667
Blockage Prediction	Alpha ( $\alpha$ )	1.1
	Batch Size	6
	Epochs	75
	Epoch of Convergence	27
	Learning Rate	0.00001
DRL	Number of Samples	18,667
	Batch Size	64
	Entropy Coefficient	0.01
	Episode of Convergence	912
	Degradation Threshold ( $\tau_{\text{degradation}}$ )	30%
	Discount Factor ( $\gamma$ )	0.95
	Number of Episodes	1000
Environment Classifier	Number of Estimators	100
	Max Depth	50
LLM	Number of Estimators	100
	Max New Tokens	4096
	Priming Reward Threshold ( $\tau_{\text{reward}}$ )	8.5
	Priming Iterations	20
	Priming Convergence (iterations)	4
	Number of Response Generation Evaluation Samples	750
	Repetition Penalty	1.15
Temperature	0.2	

Beamforming gains,  $G_{\text{beam}}$ , are estimated from predicted beam power measurements,  $P_{\text{beam}}$ , using a logarithmic conversion:

$$G_{\text{beam}} = 10 \cdot \log_{10}(P_{\text{beam}}). \quad (17)$$

Combining transmit power ( $P_{\text{tx}}$ ), pathloss ( $P_{\text{loss},i}$ ), and beamforming gain ( $G_i$ ),  $\text{BS}_i$ 's received power,  $P_{\text{rx},i}$ , is computed as:

$$P_{\text{rx},i} = P_{\text{tx}} - P_{\text{loss},i} + G_i. \quad (18)$$

Table II summarizes the key hyperparameters used during agent training and LLM deployment. For LLM inference, we set max new tokens = 4096 to support structured multi-step reasoning, temperature = 0.5 to balance determinism and diversity, and repetition penalty = 1.15 to discourage redundancy. These settings were selected through preliminary ablations and promote stable, concise outputs aligned with agent orchestration requirements.

### C. Evaluation of Beam Prediction Agent

The beam prediction agent was assessed using two KPIs:

- **Top- $k$  Accuracy** evaluates the proportion of instances in which the ground-truth beam index appears among the Top- $k$  predicted candidates. This metric reflects the agent's ability to consistently rank the correct beam within a practical search window.
- **Average actual power loss (APL)** quantifies the average degradation in received power due to suboptimal beam selection. It is computed as:

$$\text{APL}_{[dB]} = 10 \log_{10}\left(\frac{p'}{p}\right), \quad (19)$$

where  $p'$  denotes the received power of the highest-ranked beam among the Top- $k$  predictions, and  $p$  corresponds to

TABLE III  
AGENT PERFORMANCE FOR  $t+1$  WITH  $[M=16, Q=64]$ . INFERENCE TIMES REPORT SINGLE-SAMPLE GPU LATENCY.

Modality	Preprocessing Time (ms)	Beam Prediction			Blockage Prediction		
		APL	Accuracy	Inference Time (ms)	F1 Score	AUC-ROC	Inference Time (ms)
camera_gps_lidar	33.32	-0.009220	<b>88.5%</b>	54.56	91.2%	0.897	55.28
camera_radar_lidar	34.39	-0.009314	88.3%	61.48	94.0%	0.928	57.19
camera_gps_radar_lidar	36.03	-0.009660	85.8%	66.98	93.1%	0.914	59.47
camera_lidar	22.4	-0.009668	85.6%	39.28	96.2%	0.955	37.41
camera_gps_radar	18.3	-0.01469	84.9%	31.84	92.3%	0.911	30.05
camera_radar	17.83	-0.02125	84.2%	32.26	<b>98.4%</b>	0.988	30.68
camera_gps	4.961	-0.02844	83.7%	29.39	94.4%	0.931	27.70
gps_lidar_radar	21.84	-0.03553	83.0%	33.87	89.9%	0.889	31.64
camera_only	4.113	-0.03705	82.8%	26.66	98.1%	0.983	22.32
gps_radar	13.52	-0.03979	81.9%	26.40	89.3%	0.880	24.55
gps_lidar	12.45	-0.08013	76.6%	28.58	84.1%	0.855	26.91
radar_lidar	20.70	-0.1427	74.4%	34.80	91.8%	0.902	32.74
radar_only	12.47	-0.1602	72.4%	22.59	93.7%	0.922	21.81
lidar_only	10.92	-0.1743	65.7%	27.95	87.9%	0.872	25.18
gps_only	< 1	-0.2151	59.0%	13.59	61.7%	0.603	8.332

the power of the true optimal beam. Lower APL values indicate more efficient beam selection.

The beam prediction results in Table III clearly demonstrate how sensor combinations affect accuracy and power efficiency in the  $t+1$  time slot [c.f. App. G for an evaluation of all time slots]. Setups including the camera consistently outperform others, confirming the role of visual context in beam prediction. The highest-performing configuration, camera\_gps\_lidar, achieved 88.5% accuracy with an exceptionally low APL of  $-0.00922$ , showing that high precision is possible with modest fusion overhead. However, performance gains diminish beyond three modalities, suggesting limited benefit from adding more sensors once the camera and GPS are included.

In contrast, models relying on a single modality perform poorly. The gps\_only model achieved just 59.0% accuracy with an APL of  $-0.2151$ , reflecting the inadequacy of positional data alone in dynamic environments. Similarly, radar- and LiDAR-only models reached only 72.4% and 65.7% accuracy, respectively, despite offering spatial structure, indicating the need for richer perceptual input.

To supplement agent performance, Table III reports preprocessing and inference times, where inference reflects end-to-end per-sample latencies (including preprocessing and model forward-pass execution under steady-state GPU conditions).

#### D. Evaluation of Blockage Prediction Agent

The blockage prediction agent delivers strong performance, as seen in Table III [c.f. App. H for an evaluation of all time slots], across diverse sensor configurations, with F1-score and area under the receiver operating characteristic curve (AUC-ROC) as KPIs. F1-score captures precision-recall tradeoffs, while AUC-ROC measures discriminative ability across thresholds.

The best-performing model, camera\_radar, achieved an F1-score of 98.4% and an AUC-ROC above 0.988, closely followed by camera\_only and camera\_lidar. These results highlight the role of visual sensors in blockage detection in dynamic urban settings. Multi-modal combinations further enhance robustness under degraded conditions. For example, camera\_radar\_lidar reached a 94.0% F1-score and 0.928

AUC-ROC. Even radar- and LiDAR-only models maintained solid performance (F1-scores of 93.7% and 87.9%), making them valuable fallbacks when vision is degraded.

In contrast, GPS-dominant models underperformed. The gps\_only configuration yielded just a 61.7% F1-score and 0.603 AUC-ROC, confirming that spatial information alone lacks the perceptual fidelity needed to detect blockages. Blockage prediction is best supported by camera and radar data, with vision-led models excelling in clear conditions and fusion models ensuring robustness in degraded environments.

#### E. Evaluation of Environment Status Classifier Agent

The environment status classifier achieved F1-scores of 99.6% (camera), 97.8% (LiDAR), 99.1% (radar), and 99.91% (GPS) across training perturbations and unseen synthetic degradations, including partial sensor blackouts, intermittent frame drops, and composite multi-modality impairments.

These results indicate reliable degradation detection, enabling downstream agents to prioritize the most dependable modalities under varying sensor conditions.

#### F. Evaluation of DRL Agent

Our evaluation of the DRL agent was two-fold: the first study analyzes the reward performance of the learned policy compared to rule-based and random modality selection strategies; the second investigates how degradation thresholding influences policy effectiveness in dynamic conditions.

1) *Policy-Based Evaluation*: We compared the PPO-trained policy against two baselines: (1) a *random policy* that samples beam and blockage models uniformly, and (2) a *rule-based policy* that selects the highest-accuracy model among those using only non-degraded sensors without considering each model’s limitations. The evaluation was conducted over 1000 episodes sampled from a hybrid distribution: 50% ideal sensing conditions and 50% drawn from randomized sensor degradation states. A small set of these represent severe degradation of 3+ sensors to simulate rare edge cases.

Table IV reports the performance of each policy across three KPIs: average reward, percentage of optimal actions (selecting

TABLE IV  
COMPARISON OF AGENT SELECTION POLICIES

Method	Avg. Reward	% Optimal Actions	% Degraded Sensors Used
Random Policy	0.69	13.2%	61.5%
Rule-Based Policy	1.04	67.1%	38.7%
<b>DRL Agent</b>	<b>1.25</b>	<b>86.6%</b>	<b>12.9%</b>

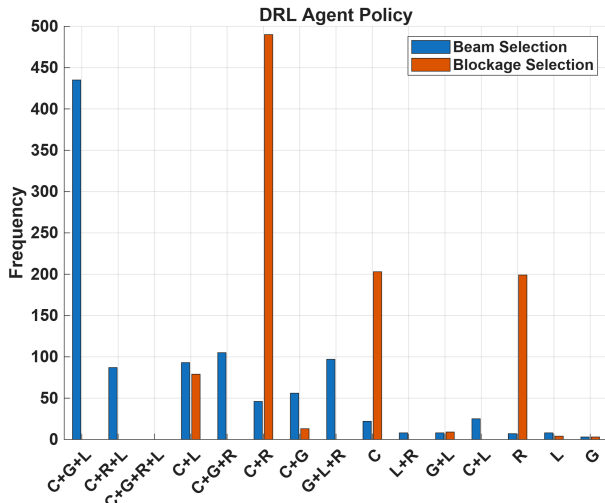


Fig. 6. Model combination selections (C: camera, G: GPS, L: LiDAR, R: radar) for prediction agents across 1000 episodes.

the best-performing model pair in the current episode’s state), and the percentage of selections involving degraded sensors.

Our DRL agent substantially outperformed both baselines, with an average reward of 1.25, which is 20.1% higher than the rule-based policy and more than double that of the random policy. It also selected optimal actions in 86.6% of episodes while minimizing reliance on degraded sensors to just 12.9%, demonstrating strong policy alignment with model performance and sensor health. In contrast, the rule-based policy often over-prioritized local accuracy, leading to degraded sensor usage in nearly 38.7% of cases.

Fig. 6 illustrates the distribution of model selections across 1000 environment states under varying sensor conditions. The selections reveal a strong alignment between sensor availability and modality-appropriate model choice: in ideal conditions, high-complexity multi-sensor models dominate with the highest accuracies are favored; under degradation, the policy shifts toward leaner, sensor-compatible models. The agent avoids unsafe selections and maintains meaningful reward accumulation even in rare severe degradation scenarios (e.g., three modalities unavailable). We refer the readers to App. I for further details on the reward distribution of the DRL agent.

2) *Threshold-Based Evaluation:* Modality selection depends on the degradation threshold  $\tau_{degradation} \in \{10\%, 20\%, \dots, 100\%\}$ , where lower values trigger earlier switching and higher values tolerate greater impairment. As shown in Fig. 7, cumulative reward increases from 10% to 30% and declines thereafter. The maximum reward occurs at  $\tau = 30\%$  (1.25) (Fig. 7A). The accuracy-driven component peaks at 1.10 (Fig. 7B), while the latency component reaches 0.33 (Fig. 7C). These results indicate that premature

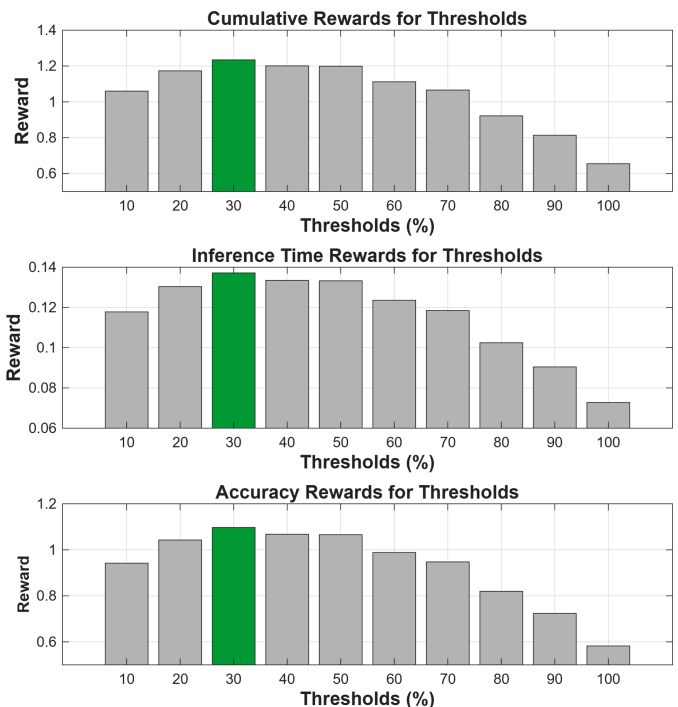


Fig. 7. Maximum DRL agent rewards through A) a cumulative and weighted reward, B) inference time-based and C) performance-based rewards.

switching underutilizes usable sensors, whereas delayed switching degrades predictive performance. Accordingly,  $\tau_{degradation} = 30\%$  is selected as the default threshold for the best accuracy-latency tradeoff under the learned DRL policy.

### G. Evaluation of Handover Agent

The handover agent monitors blockage persistence and received power differentials across BSs. When a prolonged blockage is detected ( $\geq 5$  consecutive time steps), and the alternate BS ( $BS_2$ ) consistently offers stronger received power than the original (Unit 1), the agent initiates a handover after a latency delay of one time step ( $t_{handover} = t$ ). This latency coefficient reflects a realistic switching delay often observed in communication environments.

We evaluate the handover agent across 500 potential handover sequences under varied environmental conditions, comparing four policies:

- **No Handover:** Always remain on Unit 1 regardless of link degradation.
- **Immediate Switch:** Trigger a handover as soon as the handover conditions are satisfied in the immediate step as opposed to a persisting conditions.
- **Oracle:** Premptively knows and selects all optimal handover decisions.
- **ENWAR 3.0:** Switches only after handover conditions are satisfied for  $\geq 5$  consecutive steps.

The evaluation results, shown in Fig. V, details that ENWAR 3.0 substantially improves post-blockage recovery by reducing average recovery time by over 60% compared to a non-handover policy. It captures over 85% of the oracle’s gain while avoiding nearly all of the premature switches exhibited by the immediate switching baseline.

TABLE V  
COMPARISON OF HANDOVER STRATEGIES,  $k = 5$

Metric	No Handover	Immediate	Oracle	ENWAR 3.0
APL @ $t+k$ (dBm)	-5.10	+0.71	<b>+2.53</b>	+1.82
Avg. Power Gain	0.0%	+70.6%	+181.6%	+158.2%
Premature Switch Rate	0.0%	41.0%	0.0%	<b>3.5%</b>
Avg. Handover Frequency	0	56	47	39

### H. Evaluation of Response Generation and Reasoning Correctness

1) *Evaluation Dataset Creation*: ENWAR 3.0 was evaluated across diverse sensing conditions using 150 samples per modality combination (2250 total instances), including varied lighting, weather, and occlusion scenarios. To evaluate our KPI [c.f. Subsection X-H2], 50 validation Q&A pairs were defined per combination (750 total prompts). Each pair assessed beam and blockage prediction, handover decisions, and modality selection. Corresponding reference answers served as ground truth for the systematic evaluation of reasoning correctness and explanatory alignment.

2) *Performance Criteria and KPIs*: The 50 structured evaluation prompts were presented to ENWAR 3.0, each requiring the generation of multi-step responses reflecting its internal reasoning process across key decision-making steps. Each response followed a prescribed format through LLM priming for consistent parsing and targeted evaluation. Each output was assessed using the *reasoning correctness* KPI, which measures alignment between generated and ground-truth responses.

Ground truths were manually constructed for each Q&A pair and included deterministic fields such as beam index, power differentials, blockage likelihood, and handover state. Reasoning correctness builds on the interpretability metrics in ENWAR 2.0; it evaluates factual validity and logical consistency, including whether modality selection, beam decisions, blockage inference, and handover triggers are justified given sensor states, trajectory dynamics, and environmental context.

Scoring combines strict matching for deterministic outputs with embedding-based semantic similarity for explanatory text, allowing linguistic variation while preserving logical accuracy. A response achieves high reasoning correctness when agent contexts are properly inferred, and decisions are coherently justified. Full scoring details follow the methodology introduced in ENWAR 1.0 [5].

Using thirteen system configurations, we evaluated reasoning correctness under varying sensor combinations, as shown in Fig. 8 [c.f. App. J for full modality-level results]. The baseline *vanilla LLaMa* used static prompt instructions without structured reasoning, memory, or policy-based routing. We then progressively introduced four orthogonal components: single-pass CoT reasoning (CoT), long-term memory (Mem), the DRL agent (DRL), and LLM priming (Primed). CoT provides structured multi-step reasoning without supervision, and priming extends CoT reasoning through iterative human-aligned refinement and reward-based consistency enforcement.

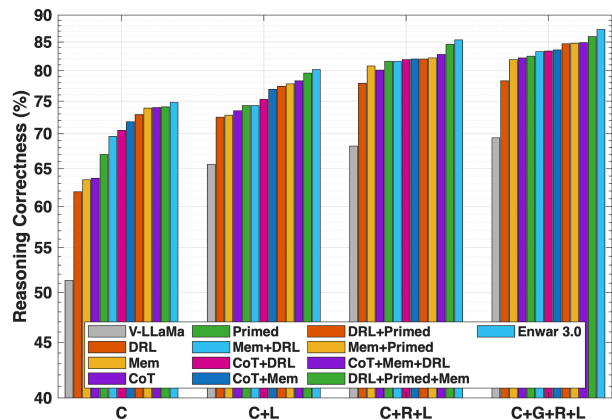


Fig. 8. Comparison of the best-performing modality combinations (1 to 4 modalities, C: camera, G: GPS, L: LiDAR, R: radar) in ENWAR 3.0’s reasoning gains from long-term memory (Mem), DRL, CoT, and LLM priming (Primed) relative to the vanilla baseline.

To isolate architectural contributions, we evaluated combinations of Mem, DRL, CoT, and Primed. Hybrid configurations combining CoT and Primed were not exhaustively enumerated, as priming inherently incorporates structured CoT patterns through supervised prompt alignment. This design choice avoids redundant configurations while preserving interpretability of component-level gains.

ENWAR 3.0 achieves 87.3% reasoning correctness with LLaMa3.2-3B, a 17.9% absolute improvement over the vanilla baseline (69.4%) under full-modality inclusion. Scaling to DeepSeek-r1-70B improves performance to 89.3%, indicating that model scale provides only modest gains.

Among individual components, priming yields the largest standalone gain (82.5%), closely followed by single-pass CoT reasoning (82.2%). This result suggests that structured reasoning accounts for the majority of the improvement over vanilla prompting, while priming predominantly enhances alignment and consistency. Long-term memory (81.9%) and DRL-based routing (78.3%) further improve performance by introducing temporal stability and degradation-aware model selection.

When combined, component gains are structured rather than merely cumulative. The CoT+Mem+DRL configuration reaches 84.9%, demonstrating that structured reasoning with temporal and policy awareness forms a stronger orchestration backbone. Adding priming to Mem+DRL increases performance to 86.0%, and the full configuration achieves 87.3%, affirming that priming strongly refines the pipeline’s structure.

Performance gains become clearer as modality complexity increases, with the largest improvements observed under triple- and quadruple-modality inputs. This trend indicates that memory and policy-based routing are critical as sensor diversity grows. These results demonstrate that reasoning structure and orchestration design drive performance more than parameter scale alone: CoT establishes structured inference, priming improves alignment, and memory with DRL enables stable multi-modal coordination under dynamic sensing conditions.

3) *Edge-Cases*: The edge-case study in Table VI displays ENWAR 3.0’s capacity to operate reliably under various degraded sensing conditions. In scenarios ranging from dense fog to GPS jamming, the system consistently produced coherent,

TABLE VI  
ENWAR 3.0 EDGE-CASES

Scenario	Available Modalities	Beam Accuracy	Blockage F1	Enwar 3.0 Judgement	Final Action
Clear weather, ideal scenario	All	88.5%	98.4%	All sensors functional; select best model	Use best model fusion for both agents
Heavy fog, visual occlusion with average degradation 42.8%	GPS, LiDAR, radar (camera degraded)	83.0%	93.7%	Camera sensor degraded; fallback to radar-based fusion models	Use radar model for blockage, and GPS-radar-LiDAR fusion for beam
Bright sun glare with average degradation 31.0%	GPS, LiDAR, radar (camera occluded)	83.0%	93.7%	Camera sensor degraded; fallback to radar-based fusion models	Use radar model for blockage, and GPS-radar-LiDAR fusion for beam
Nighttime, low visibility with average degradation 55.0%	GPS, LiDAR, radar (camera degraded)	83.0%	93.7%	Camera sensor degraded; fallback to radar-based fusion models	Use radar model for blockage, and GPS-radar-LiDAR fusion for beam
GPS jamming with average degradation 37.2%	Camera, LiDAR, radar	88.3%	98.4%	GPS signal unreliable; use camera-LiDAR-radar models	Use camera-radar for blockage, and camera-radar-LiDAR for beam
LiDAR sensor non-functional with average degradation 59.4%	Camera, GPS, radar	84.9%	98.4%	LiDAR degraded; fallback to radar-based models	Use to camera-radar fusion for both agents
Radar noisy in rain with average degradation 39.9%	Camera, GPS, LiDAR (radar noisy)	85.6%	98.1%	Radar noise detected; rely on camera-LiDAR	Use camera-LiDAR fusion for beam, and camera model for blockage
Blockage longer than $t + 5$ time steps confirmed, BS <sub>2</sub> RSSI higher	All	88.5%	98.4%	Blockage confirmed, BS <sub>2</sub> power difference exceeds threshold	Trigger handover to BS <sub>2</sub> ; use best fusion models
Blockage longer than $t + 5$ time steps confirmed, Unit 1 RSSI higher	All	88.5%	98.4%	Blockage confirmed, BS <sub>2</sub> power difference does not exceed threshold	Stay on Unit 1; use best fusion models
LiDAR and radar non-functional with average degradation 63.7%	Camera, GPS	83.7%	98.1%	LiDAR and radar degraded; use camera-gps fusion models	Use camera model for blockage, camera-GPS for beam
LiDAR and GPS non-functional with average degradation 68.2%	Camera, radar	85.6%	98.1%	LiDAR and GPS degraded; use camera-radar fusion models	Use camera model for blockage, camera-LiDAR for beam
Camera and GPS non-functional with average degradation 71.1%	LiDAR, radar	74.4%	93.7%	LiDAR and radar degraded; use camera-gps fusion models	Use radar model for blockage, radar-LiDAR for beam
Camera, GPS, and LiDAR non-functional with average degradation 84.9%	Radar	72.4%	93.7%	Camera, GPS, and LiDAR degraded; use radar-only models	Use radar-only models

contextually appropriate decisions by dynamically adjusting its model selection, fallback strategies, and handover logic based on real-time sensor status. In each case, the final action aligned with the system’s internal judgment, which shows robust coordination between the environment classifier, agent manager, and decision agents. These results affirm that ENWAR 3.0 is situation-aware when enforcing practical actions.

4) *LLM Architecture and Size Comparisons*: Reasoning correctness was evaluated across multiple LLM architectures under full multi-modal input (Fig. 9). Although larger models achieve higher accuracy, marginal gains diminish with scale (Fig. 10). To quantify this trend, we compute average reasoning correctness across comparable model sizes (e.g., LLaMa 3.2-3B and Qwen 2.5-3B grouped as 3B).

The normalized gain per billion parameters (pBp) decreases sharply with scale: 3B models achieve an average gain of 29.1, compared to 10.9 (8B), 2.75 (32B), and 1.28 (70B). This fact confirms diminishing returns relative to parameter growth.

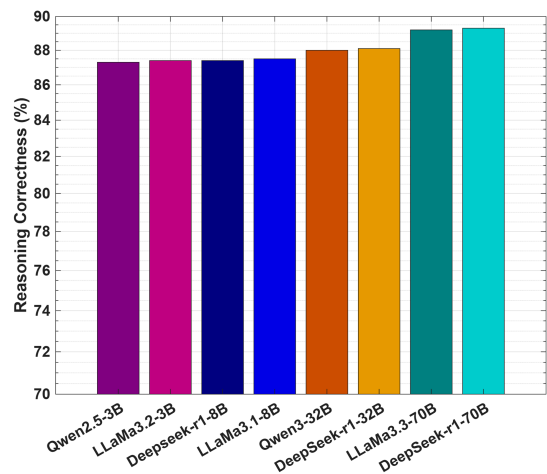


Fig. 9. Reasoning correctness across different sized LLMs (3B-70B).

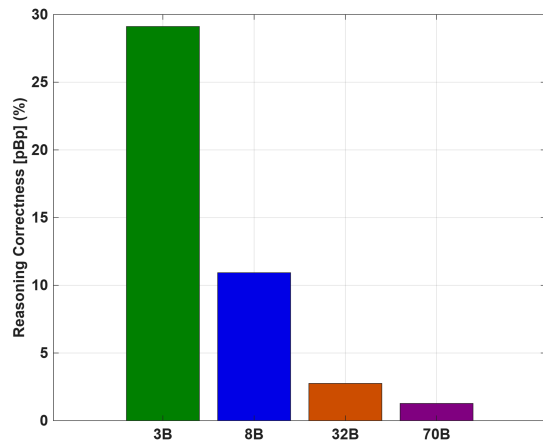


Fig. 10. Reasoning correctness efficiency per billion parameters (pBp).

Inference latency scales similarly. 70B models require up to 3.08s per multi-agent prompt, compared to 2.17s for 3B models. While larger LLMs provide incremental modest improvements, smaller models offer a more favorable efficiency-performance tradeoff for real-world deployment.

### I. Computational and Deployment Considerations

ENWAR 3.0 operates under a sensor sampling interval of  $T_s$ . We distinguish between the time-critical control path and asynchronous explanation generation. Upon receiving new sensor data, perception modules, the environment classifier, and the DRL policy execute in parallel. The LLM concurrently constructs a structured invocation packet specifying agent selection and network updates.

As shown in Table VII, the worst-case structured invocation time for the primed 3B model is 222.7ms. Beam and blockage inference require at most 66.98ms and 59.47ms, respectively, under steady-state single-sample GPU execution. The resulting worst-case critical-path latency is bounded by approximately 290ms, remaining within the  $T_s = 300$ ms sensing interval. LLM response generation (2.17-3.08s depending on model size) is fully decoupled from the control loop and runs asynchronously for justification and memory updates. Architectural optimizations, including 4-bit quantization, LoRA fine-tuning, GPU affinity scheduling, caching, and asynchronous execution, ensure bounded latency under multi-modal load. Additional computational details are provided in App. K.

## XI. CONCLUSION

This work presents ENWAR 3.0, a multi-agent, environment-aware reasoning framework that advances the robustness and interpretability of mmWave I2V communication systems. ENWAR 3.0 enables contextual decision-making for beamforming, blockage prediction, handover management, and environment perception by combining real-time multi-modal sensing, a sensor degradation-aware classifier, structured prompt processing, and primed LLM reasoning with specialized agents.

### ACKNOWLEDGMENTS

ChatGPT [44] was used to assist with organizing and refining language flow for consistency in tone and terminology.

TABLE VII  
ENWAR 3.0 KEY TIME INDICATORS.

Process	Time
<b>Critical-Path End-to-End (Worst Case)</b>	<b>289.7ms</b>
Input Sampling Period ( $T_s$ )	300ms
Input Sliding Window ( $T_s$ )	1.5s
Primed 70B LLMs response generation	3.08s
Vanilla 3B LLMs with prompt template response generation	2.90s
Primed 32B LLMs response generation	2.45s
Llama3.2-Vision Image-To-Text (per frame)	2.34s
Primed 8B LLMs response generation	2.23s
Primed 3B LLMs response generation	2.17s
Primed 70B LLMs agent invocations and network updates (worst case)	242.6ms
Primed 32B LLMs agent invocations and network updates (worst case)	241.2ms
Vanilla 3B LLMs with prompt template agent invocations and network updates	618.8ms
Primed 8B LLMs agent invocations and network updates (worst case)	235.1ms
Primed 3B LLMs agent invocations and network updates (worst case)	222.7ms
Beam Prediction Inference (worst case)	66.98ms
Blockage Prediction Inference (worst case)	59.47ms
LiDAR DBSCAN	14.7ms
Radar DBSCAN	12.4ms
DRL Agent Inference	2ms
YOLO per frame camera detections	5ms
Environment Classifier Inference	3ms
Handover Agent Inference	3ms
GPS detections	< 1ms

## REFERENCES

- [1] A. M. Nazar, A. Celik, M. Y. Selim, A. Abdallah, D. Qiao, and A. M. Eltawil, "Multi-modal sensor fusion for proactive blockage prediction in mmWave vehicular networks," 2025. [Online]. Available: <https://arxiv.org/abs/2507.15769>
- [2] A. Celik and A. M. Eltawil, "At the dawn of generative AI era: A tutorial-cum-survey on new frontiers in 6G wireless intelligence," *IEEE Open Journal of the Comms. Soc.*, vol. 5, pp. 2433–2489, 2024.
- [3] 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>
- [4] A. M. Nazar, A. Celik, M. Y. Selim, A. Abdallah, D. Qiao, and A. M. Eltawil, "ENWAR 2.0: An agentic multimodal wireless LLM framework with reasoning, situation-aware explainability and beam tracking," *IEEE Transactions on Mobile Computing*, pp. 1–18, 2025.
- [5] —, "ENWAR: A RAG-empowered multi-modal LLM framework for wireless environment perception," *IEEE Comms. Magazine*, 2026.
- [6] —, "Encoders, roll out! A multi-modal sensor transfusion for proactive I2V beam prediction," 03 2025.
- [7] S. Xu *et al.*, "Large Multi-Modal Models (LMMs) as universal foundation models for AI-native wireless systems," *IEEE Network*, 2024.
- [8] G. M. Yilma *et al.*, "TelecomRAG: Taming telecom standards with retrieval augmented generation and LLMs," *SIGCOMM Comput. Commun. Rev.*, Jan. 2025.
- [9] A. M. Nazar, M. Y. Selim, D. Qiao, and H. Zhang, "NextG-GPT: Leveraging GenAI for advancing wireless networks and communication research," 2025. [Online]. Available: <https://arxiv.org/abs/2505.19322>
- [10] H. Zou *et al.*, "TelecomGPT: A framework to build telecom-specific large language models," *IEEE Trans. on Machine Learning in Comms. and Networking*, 2025.
- [11] Y. Shen, J. Shao, X. Zhang, Z. Lin, H. Pan, D. Li, J. Zhang, and K. B. Letaief, "Large language models empowered autonomous edge AI for connected intelligence," *IEEE Communications Magazine*, 2024.
- [12] F. Jiang *et al.*, "Large language model enhanced multi-agent systems for 6G communications," *IEEE Wireless Communications*, 2024.
- [13] J. Tong *et al.*, "WirelessAgent: Large language model agents for intelligent wireless networks," 2025. [Online]. Available: <https://arxiv.org/abs/2505.01074>

- [14] J. Shao *et al.*, “WirelessLLM: Empowering large language models towards wireless intelligence,” *Journal of Comms. and Information Networks*, 2024.
- [15] M. Xu *et al.*, “When large language model agents meet 6G networks: Perception, grounding, and alignment,” 2024.
- [16] T. Yang, P. Zhang, M. Zheng, Y. Shi, L. Jing, J. Huang, and N. Li, “WirelessGPT: A generative pre-trained multi-task learning framework for wireless communication,” *IEEE Network*, 2025.
- [17] L. Team, “The llama 3 herd of models,” Jul 2024. [Online]. Available: <https://ai.meta.com/research/publications/the-llama-3-herd-of-models/>
- [18] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023.
- [19] J. Zhang, J. Huang, S. Jin, and S. Lu, “Vision-language models for vision tasks: A survey,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2024.
- [20] U. Demirhan and A. Alkhateeb, “Radar aided proactive blockage prediction in real-world millimeter wave systems,” 2021. [Online]. Available: <https://arxiv.org/abs/2111.14805>
- [21] J. Choi, V. Va, N. Gonzalez-Prelcic, R. Daniels, C. R. Bhat, and R. W. Heath, “Millimeter-wave vehicular communication to support massive automotive sensing,” *IEEE Communications Magazine*, 2016.
- [22] A. Abdallah, A. Celik, M. M. Mansour, and A. M. Eltawil, “Multi-agent deep reinforcement learning for beam training in cell-free RIS-aided systems,” *IEEE Transactions on Wireless Communications*, 2024.
- [23] S. H. A. Shah and S. Rangan, “Multi-cell multi-beam prediction using auto-encoder LSTM for mmWave systems,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 12, pp. 10 366–10 380, 2022.
- [24] M. Alrabeiah, A. Hredzak, and A. Alkhateeb, “Millimeter wave base stations with cameras: Vision aided beam and blockage prediction,” 2019. [Online]. Available: <https://arxiv.org/abs/1911.06255>
- [25] S. Jiang, G. Charan, and A. Alkhateeb, “LiDAR aided future beam prediction in real-world millimeter wave V2I communications,” *IEEE Wireless Communications Letters*, pp. 1–1, 2022.
- [26] K. Dong, M. Mizmizi, D. Tagliaferri, and U. Spagnolini, “Vehicular blockage modelling and performance analysis for mmWave V2V communications,” in *IEEE International Conference on Comms.*, 2022.
- [27] G. Liu *et al.*, “Wireless agentic AI with retrieval-augmented multimodal semantic perception,” *IEEE Communications Magazine*, 2026.
- [28] B. Du, H. Du, D. Niyato, and R. Li, “Task-oriented semantic communication in large multimodal models-based vehicle networks,” *IEEE Transactions on Mobile Computing*, 2025.
- [29] N. Khan, S. Coleri, A. Abdallah, A. Celik, and A. M. Eltawil, “Explainable and robust artificial intelligence for trustworthy resource management in 6G networks,” *IEEE Communications Magazine*, 2024.
- [30] N. Khan, A. Abdallah, A. Celik, A. M. Eltawil, and S. Coleri, “Explainable and robust millimeter wave beam alignment for AI-native 6G networks,” *arXiv preprint arXiv:2501.17883*, 2025.
- [31] —, “Explainable AI-aided feature selection and model reduction for DRL-based V2X resource allocation,” *IEEE Trans. on Comms.*, 2025.
- [32] S. Alikhani, G. Charan, and A. Alkhateeb, “Large wireless model (LWM): A foundation model for wireless channels,” 2025. [Online]. Available: <https://arxiv.org/abs/2411.08872>
- [33] K. Ding, C. Guo, Y. Yang, W. Hu, and Y. C. Eldar, “A new paradigm of user-centric wireless communication driven by large language models,” 2025. [Online]. Available: <https://arxiv.org/abs/2504.11696>
- [34] A. Abdallah, A. Albaseer, A. Celik, M. Abdallah, and A. M. Eltawil, “NetOrchLLM: Mastering wireless network orchestration with large language models,” *arXiv preprint arXiv:2412.10107*, 2024.
- [35] A. Plaat, M. van Duijn, N. van Stein, M. Preuss, P. van der Putten, and K. J. Batenburg, “Agentic large language models, a survey,” 2025. [Online]. Available: <https://arxiv.org/abs/2503.23037>
- [36] J. Wei *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2022.
- [37] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2020.
- [38] K. Haneda *et al.*, “5G 3GPP-like channel models for outdoor urban microcellular and macrocellular environments,” in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*. IEEE, May 2016. [Online]. Available: <http://dx.doi.org/10.1109/VTCSpring.2016.7503971>
- [39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017. [Online]. Available: <https://arxiv.org/abs/1707.06347>
- [40] G. Charan, U. Demirhan, J. Morais, A. Behboodi, H. Pezeshki, and A. Alkhateeb, “Multi-modal beam prediction challenge 2022: Towards generalization,” 2022. [Online]. Available: <https://arxiv.org/abs/2209.07519>
- [41] A. Alkhateeb, G. Charan, T. Osman, A. Hredzak, J. Morais, U. Demirhan, and N. Srinivas, “DeepSense 6G: A large-scale real-world multi-modal sensing and comm. dataset,” *IEEE Comm. Mag.*, 2023.
- [42] B. Wang, J. Lan, and J. Gao, “LiDAR Filtering in 3D Object Detection Based on Improved RANSAC,” *Remote Sensing*, 2022.
- [43] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3D classification and segmentation,” 2017. [Online]. Available: <https://arxiv.org/abs/1612.00593>
- [44] OpenAI. [Online]. Available: <https://chat.openai.com/chat>
- [45] D. Guo *et al.*, “DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning,” 2025. [Online]. Available: <https://arxiv.org/abs/2501.12948>
- [46] A. Yang *et al.*, “Qwen2.5 technical report,” 2025. [Online]. Available: <https://arxiv.org/abs/2412.15115>



**Ahmad M. Nazar** (Member, IEEE) is a Postdoctoral Scholar in the Department of Electrical and Computer Engineering at Iowa State University (ISU), USA and a Lead Research Engineer at Gladius Technological Institute, USA. He received a Ph.D. degree in Computer Engineering from ISU, in 2025; and he also earned M.S. and B.S. degrees in Computer Engineering in 2022 and 2020 from ISU. His research interests involve interdisciplinary applications of generative AI and machine learning with a focus on LLMs and multi-modal AI agents.



**Abdulkadir Celik** (Senior Member, IEEE) is an Associate Professor in the School of Electronics and Computer Science at the University of Southampton, UK, where he also serves as the Director of the Centre for Internet of Things and Pervasive Systems. He received the Ph.D. degree in co-majors of Electrical Engineering and Computer Engineering from Iowa State University, Ames, IA, USA, in 2016; wherein he also earned M.S. degrees in Electrical Engineering and Computer Engineering in 2013 and 2015, respectively. Prior to his current appointment,

he was a senior research scientist from 2020 to 2025 and a post-doctoral fellow from 2016 to 2020 at King Abdullah University of Science and Technology (KAUST), Thuwal, KSA. Dr. Celik is the recipient of IEEE Communications Society’s 2023 Outstanding Young Researcher Award for Europe, Middle East, and Africa (EMEA) region. He currently serves as an editor for *npj Wireless Technology*, *IEEE Transactions on Communications*, *IEEE Communications Letters*, *IEEE Wireless Communication Letters*, and *Frontiers in Communications and Networks*. His research interests are in the broad areas of next-generation wireless communication systems and networks.



**Mohamed Y. Selim** (Senior Member, IEEE) is an Associate Teaching Professor in the Department of Electrical and Computer Engineering at Iowa State University, USA. He received the Ph.D. and M.Sc. degrees in computer engineering from Iowa State University in 2018 and 2016, respectively, and the M.Sc. degree in electrical engineering from Port Said University, Egypt, in 2011. He has co-authored numerous papers in leading venues such as IEEE Communications Magazine, IEEE Transactions on Mobile Computing, and Computer Networks, and

contributed book chapters on 5G and UAV-based RIS systems. Dr. Selim has received multiple teaching and educational impact awards at Iowa State University and has been recognized as an Exemplary Reviewer for IEEE Communications Letters (2023) and a Distinguished Reviewer for IEEE Transactions on Mobile Computing (2024). Dr. Selim's research interests span next-generation wireless communication systems, large-scale testbeds, reconfigurable intelligent surfaces, and agentic LLMs for AI-native wireless networks. He is a co-PI of the \$8 million NSF-funded ARA PAWR project.



**Asmaa Abdallah** (Member, IEEE) received the B.S. (with High Distinction) and M.S degree in computer and communications engineering from Rafik Hariri University (RHU), Lebanon, in 2013 and 2015, respectively. In 2020, she received the Ph.D. degree in electrical and computer engineering at the American University of Beirut (AUB), Beirut, Lebanon. She was a Postdoctoral Fellow at King Abdullah University of Science and Technology (KAUST), from 2021-2024, where she is currently a Research Scientist with the Communications and Computing

Systems Laboratory. From 2016 to 2020, She has been a member of the executive committee of IEEE Young Professionals Lebanon's Section. Dr. Abdallah was the recipient of the Academic Excellence Award at RHU in 2013 for ranking first on the graduating class. She also received a scholarship from the Lebanese National Counsel for Scientific Research (CNRS-L/AUB) to support her doctoral studies. In 2023, Dr. Abdallah was selected by MIT technology review as one of the leading 15 Innovators under 35 in the MENA area. Her research interests include machine learning, communication theory, stochastic geometry, array signal processing, with emphasis on energy and spectral efficient algorithms for next-generation wireless communication systems.



**Daji Qiao** (Senior Member, IEEE) is a Professor in the Department of Electrical and Computer Engineering at Iowa State University. He received his Ph.D. from the University of Michigan, Ann Arbor. His research focuses on wireless networking and mobile computing, 5G/6G systems, sensor networks, and IoT. He is a Senior Member of the IEEE and a Member of the ACM.



**Ahmed M. Eltawil** (Senior Member, IEEE) is a Professor and Associate Dean for Research at the Computer, Electrical, and Mathematical Sciences and Engineering (CEMSE) Division at King Abdullah University of Science and Technology (KAUST). Previously, he was a Professor of Electrical Engineering and Computer Science at the University of California, Irvine (UCI) from 2005 to 2021. Professor Eltawil earned his doctorate degree from the University of California, Los Angeles, in 2003 and his Master's and Bachelor's degrees from Cairo

University in 1999 and 1997, respectively. At KAUST, he established the Communication and Computing Systems Laboratory (CCSL) to conduct research on efficient architectures for computing and communications systems, with a focus on mobile wireless systems. His research interests encompass various application domains, such as low-power mobile systems, machine learning platforms, sensor networks, body area networks, and critical infrastructure networks. He served as a distinguished lecturer for IEEE COMSOC during the 2023/24 term. Additionally, he holds senior membership in IEEE and the National Academy of Inventors in the United States. He received several recognitions and awards, including the US National Science Foundation CAREER award, the 2021 "Innovator of the Year" award by the Henry Samueli School of Engineering at the University of California, Irvine, and two United States Congress certificates of merit, among other recognitions. He has served in numerous editorial roles over the years, as well as an expert reviewer for national and international funding agencies and review boards.

APPENDIX A  
TABULAR SUMMARY OF RELATED WORKS

This section summarizes the related work discussed in Section II, with details provided in Table VIII.

TABLE VIII  
RELATED WORKS AND THEIR SUMMARIES

Reference	Summary
[3], [12], [13] [34], [35]	Multi-agent task-solving LLM frameworks for 6G communications with retrieval, planning, evaluation.
[17], [45], [46]	LLMs used to evaluate ENWAR 3.0
[7]	Perspectives on large multi-modal models with causal reasoning and neuro-symbolic AI for 6G Networks
[8]–[10]	Introduces RAG LLM frameworks for wireless systems within domain-specific datasets for context-aware, real-time support in network and telecom domains.
[11], [14]–[16]	Integration of LLMs, GPTs, and AI in 6G architectures for intent-driven, intelligent network operations.
[18]	LLaVa vision-language model.
[19]	Survey on vision-language models.
[20]–[26]	Machine learning methods incorporating sensor-based blockage/beam prediction.
[27], [28]	Task-oriented semantic communication using large multi-modal model, agents, and RAG for efficient bandwidth data exchange in vehicular environments.
[29]–[31]	Frameworks for explainable and robust AI solutions in 6G networks with machine learning approaches.
[32]	Large Wireless Model, a fine-tuned LLM for wireless communication-based solutions.
[33]	LLM framework transforming user requests into intent-focused structured optimization tasks/queries for real-time wireless semantic communication systems.
[36]	Chain-of-Thought prompting techniques and how it improves reasoning in LLMs.
[37]	Few-shot learning methods and its effects on LLMs.
[39]	Deep-learning based PPO algorithm.
[40], [41]	I2V dataset used within ENWAR 3.0.
[42], [43]	LiDAR preprocessing, and PointNet architecture.

APPENDIX B  
LLM PRIMING

This section presents the prompt template used for LLM priming, along with a three-iteration example with reward-guided human feedback and iterative response refinement.

A. Main Priming Template Prompt

**Priming Template to Generate Final LLM Response**

- *Environment Status*: current environment conditions
- *Modality Selections Based on Environment Status*: what is the current degradation level of modalities.
- *DRL Agent Selection*: The selected models based on the DRL agent’s output.
- *Trajectory*: include the trajectory of Unit 2 from the perspective of the selected BS.
- *Blockage Status*: whether there is a blockage in this sample or not.
- *Handover Status*: what is the handover status and its properties, such as the power differences if a handover occurred and which BS has the highest power.
- *Predicted Beam and Properties*: Current beam’s predicted power and its properties.
- *Perception*: Provide a summary of the perceived environment using the available perceived information.

B. Priming Reward Model Pseudocode

The following section details the priming phase’s pseudocode (Algorithm 1) with human-in-the-loop feedback.

**Algorithm 1: LLM Priming w/ RLHF Loop**

---

**Input:** Priming examples  $(s_p^i)_{i=1}^N$  where  $s_p$  includes sensor conditions, DRL selections, memory context

**Output:** Primed LLM that consistently generates  $a_p$  with  $R(s_p, a_p) > \tau_{\text{reward}}$

**Initialize:** Reward rubric  $\theta$ , threshold  $\tau_{\text{reward}}$ , max iterations  $J$ ;

**foreach** example  $s_p^i$  **do**

$j \leftarrow 0$ ;

**repeat**

$a_p^j \leftarrow \text{LLM.generate}(s_p^i)$ ;

// Expert applies fixed rubric

$r^j \leftarrow R_{\text{priming}, \theta}(s_p^i, a_p^j)$ ;

**if**  $r^j \leq \tau_{\text{reward}}$  **then**

// Add few-shot or clarify

ModifyPrompt( $s_p^i$ , Feedback);

$j \leftarrow j + 1$ ;

**until**  $R(s_p^i, a_p^j) \leq \tau_{\text{reward}}$  **and**  $j \leq J$ ;

**return** Primed LLM;

---

C. Three-Iteration Priming Example

This section shows a priming example with reward-guided human feedback and iterative response refinement.

---

**Prompt 1:** Perceive the environment using the available modalities based on your sense of the environment status. Predict the beam selection for the sample, and provide its properties. Check the blockage status and determine whether a handover is necessary. You will be scored on a scale of 0-10 based on your correctness, justification of agentic choices, and explanation clarity. Your reasoning skills will also be scored and you will be penalized for repeated agent invocations.

---

**Internal Agent Context:**

- Modality Status: Radar degraded (42.1%)
- Average degradation of remaining modalities: 25.9%
- Scene Data: Urban intersection, 3 vehicles, stop sign.
- Detected objects: LiDAR clusters (6–15 m): 12 detected, camera confirms these are vehicles with some pedestrians near a stop sign with multiple bounding boxes. Multiple nearby objects exist that may affect beam propagation.
- DRL Selection: camera\_gps\_lidar model for beam, camera\_only for blockage
- Camera: Stable visual feed with clear blockage detected.
- Handover Status: blockage threshold conditions satisfied, and BS<sub>2</sub> has a higher beam power (0.1492) than Unit 1 (0.1267), therefore handover to BS<sub>2</sub>.
- Unit 2’s bearing to the current BS (Unit 1) is 52.92°.
- Blockage Status: blockage for > 1.5s (probability: 0.9835).
- BS<sub>2</sub> selected beam is beam 38 (centered at 50.86°, width 5.118°, power 0.14928), Unit 1’s selected beam is beam 40 (centered at 51.89°, width 5.997°, power 0.1226)

In this response, blockage prediction and handover agents were invoked twice, beam prediction and perception agents once.

---

**Response 1:** Environment Status: Urban intersection during daytime with active traffic and a stop sign. Scene is complex with multiple vehicle clusters and potential occlusions.

Modality Selections Based on Environment Status: Radar degraded and excluded from decision-making. Remaining modalities within acceptable degradation threshold.

Blockage Model Selected: **Camera-only model**

Justification:

- Radar excluded due to high degradation (42.1%)
- Camera provides clear object detection at the stop sign
- LiDAR confirms spatial placement of vehicles but lacks semantic understanding
- DRL policy aligns with model selection
- Camera preferred for semantic tasks in urban settings.

Recommendation: Use camera-only for blockage, multi-modal fusion if degradation  $> 30\%$  in future samples.

---

**Prompt 2:** Ensure that handover justification, and more perceived objects and their details are included. Do not call any agent more than once. Score for previous response 2.4.

---

**Response 2:** Detected objects: LiDAR clusters (6-15 m): 12 detected, camera confirms these are vehicles with some pedestrians nearby a stop sign with bounding box centers ranging from (156.3, 45.1) to (240.1, 138.9). Handover Status: blockage threshold conditions satisfied, and BS<sub>2</sub> has a higher beam power (0.1492) than Unit 1 (0.1267), handover to BS<sub>2</sub>.

---

**Prompt 3:** Your final answer should follow this format:

- *Environment Status:* current environment conditions
- *Modality Selections Based on Environment Status:* what is the current degradation level of modalities.
- *DRL Agent Selection:* The selected models based on the DRL agent’s output.
- *Trajectory:* include the trajectory of Unit 2 from the perspective of the selected BS.
- *Blockage Status:* whether there is a blockage in this sample or not.
- *Handover Status:* what is the handover status and its properties, such as the power differences if a handover occurred and which BS has the highest power.
- *Predicted Beam and Properties:* Current beam’s predicted power and its properties.
- *Perception:* Provide a summary of the perceived environment using the available perceived information.

Score for previous response 7.3.

---

**Response 3:**

- *Environment Status:* The setting appears to be a relatively clear daytime urban intersection. LiDAR and camera data indicate three vehicles present, with one vehicle approaching from a new direction and obstructing another at a stop sign. This dynamic scene suggests partial occlusions along the line of sight. Radar is degraded and excluded it is not expected to select radar-based models.

- *Modality Selections Based on Environment Status:* Radar is degraded at 42.1% with noisy scans, exceeding the reliability threshold. All other modalities are usable (avg. degradation 25.9%).
- *DRL Agent Selection:* Use camera\_gps\_lidar for beam prediction and camera\_only for blockage detection.
- *Trajectory:* Unit 2 is traveling at a bearing of 52.92° relative to the current BS (Unit 1). Multiple LiDAR clusters (12 total) are detected within 6-15 meters of Unit 1; nearby objects may affect beam propagation.
- *Blockage Status:* Confirmed with probability 0.9835. Blockage has persisted for longer than 1.5 seconds, satisfying the blockage condition for a handover trigger.
- *Handover Status:* A handover to BS<sub>2</sub> is recommended. BS<sub>2</sub>’s beam 38 (centered at 50.86°, width 5.118°, power 0.14928) offers significantly better performance than Unit 1’s beam 40 (centered at 51.89°, width 5.997°, power 0.1226). BS<sub>2</sub>’s beam sufficiently covers the current vehicle trajectory of 52.92°, and aligns better with the power and direction requirements for maintaining a robust link.
- *Predicted Beam and Properties:* Best beam is beam 38 from BS<sub>2</sub>, with center 50.86°, width 5.118°, and power 0.14928, covering the vehicle’s trajectory and surpassing Unit 1’s selected beam in strength and alignment.
- *Perception:* Radar is excluded due to degradation. LiDAR clusters and camera inputs indicate three vehicles in the scene, with one vehicle creating a new occlusion pattern not present in earlier frames. This blockage likely targets Unit 2 directly, supported by GPS trajectory, object distance, and bearing estimation. Two cars seem to be overlapping each other. This output confirms the result of the blockage prediction model.  
To summarize, handover to BS<sub>2</sub> and use beam 38. The beam’s center and width cover the vehicle’s 52.92° trajectory. Maintain blockage monitoring through camera-only models until radar recovers. Once radar recovers below degradation thresholds, reintroduce radar-based fusion for better temporal consistency and backup perception.

---

**Prompt 4:** Going forward follow this quality of responses. The score for the previous response is 9.1.

---

## APPENDIX C ENVIRONMENT PERCEPTION AGENT

This section summarizes the environment perception agent’s adoption from ENWAR 2.0 [4] into ENWAR 3.0.

### A. Camera Perception

YOLO is employed to produce bounding boxes, class labels, and confidence scores for real-time object detection. Detections with confidence  $< 0.5$  are discarded. Each object is encoded as  $(x, y, w, h)$ , representing bounding box center coordinates and dimensions, with its class label (e.g., vehicle, pedestrian). Detections are converted into textual summaries and appended to the prompt for the LLM to incorporate scene semantics into downstream decisions.

### B. GPS Perception

GPS data from Unit 2 and the serving RSU are used to compute relative distance and bearing. Calibrated trajectories are processed by the LSTM-based encoder from ENWAR 2.0 [4], [6] to predict short-term displacement and heading. Current and predicted states are translated into structured phrases (e.g., “*vehicle at 33.420, -111.929 heading NE at 12 km/h*”) and appended to the prompt. When handover to BS<sub>2</sub> occurs, relative spatial references are updated accordingly.

### C. LiDAR Perception

LiDAR point clouds are clustered using DBSCAN (Section IV) to group spatially coherent objects while removing outliers. Each cluster is enclosed in a 3D bounding box with estimated dimensions, centroid, orientation, density, and vertical spread. Cluster descriptors are converted into compact textual summaries (e.g., “*object 2.3 m long at 45°*”) and appended to the LLM input for spatial reasoning.

### D. Radar Perception

Radar scans are clustered via DBSCAN with  $\epsilon_{\text{radar}} = 2.5$  and  $\text{min\_samples}_{\text{radar}} = 2$ , selected using the same  $k$ -distance heuristic as LiDAR clustering. Each cluster is characterized by average range, radial velocity, and angular spread in the polar frame. These characterized descriptors are converted into structured summaries (e.g., “*object at 15 m with radial velocity 2 m/s*”) and are included in the prompt.

## APPENDIX D BEAM PREDICTION AGENT

The beam prediction agent, adopted from ENWAR 2.0 [4], predicts the optimal beam using temporal multi-modal inputs. The architecture consists of modality-specific encoders, early fusion, a transformer, and a final scoring layer.

1) *Camera Encoder*: The camera encoder extracts spatial-temporal features from RGB sequences. Each frame passes through three convolutional layers with ReLU activations, followed by flattening and a single-layer LSTM with 128 hidden units. The final hidden state is used as the compact visual representation.

2) *GPS Encoder*: The GPS encoder processes normalized displacement, velocity, and angular features using a two-layer LSTM (128 hidden units). The final hidden vector is projected through a fully connected layer to encode trajectory dynamics.

3) *LiDAR Encoder*: The LiDAR encoder follows a PointNet-based design [43]. Each point cloud frame is processed via three 1D convolutions (kernel size 1) with ReLU activations, followed by max pooling. Processed frames are passed to a single-layer LSTM (128 hidden units) to capture temporal evolution.

4) *Radar Encoder*: The radar encoder transforms radar tensors into spatiotemporal embeddings using three fully connected layers with ReLU activations followed by an LSTM (128 hidden units). The final hidden state captures reflectivity and motion cues relevant to beam selection.

5) *Early Fusion*: A key design element in this pipeline is early feature fusion pre-transformer processing. Encoder outputs are concatenated and passed through two fully connected layers with ReLU and dropout to produce a unified representation. This early fusion enables the transformer to learn inter-modal dependencies from semantically aligned features.

6) *Transformer Block*: The transformer block models cross-modal and temporal dependencies via multi-head self-attention, residual connections, layer normalization, dropout, and a two-layer feed-forward network. The output encodes high-level relationships across modalities.

7) *Output Layer*: A final fully connected layer maps the transformer output to a  $Q$ -dimensional beam score vector. The beam with the highest score is selected as the optimal beam.

## APPENDIX E DETAILED PREDICTION AGENTS’ MODEL ARCHITECTURE

This section illustrates each prediction agent’s internal three stage architecture: 1) data preprocessing, 2) feature extraction and fusion, and 3) beam and blockage predictions post-feature fusion. The full model architecture is seen in Fig. 11.

## APPENDIX F ENWAR 3.0 INFERENCE EXAMPLES

This section provides representative ENWAR 3.0 inference examples under varying conditions. Fig. 12 illustrates optimal operation with clear LoS between Units 1 and 2, no blockage, and no handover, showing consistent agent invocation and policy-aligned reasoning.

Figs. 13-16 demonstrate behavior under modality degradation. When the camera is excluded (Fig. 13), the DRL policy selects a reduced modality model; beam prediction deviates from ground truth due to loss of visual detail, while blockage detection and handover triggering remain correct. With LiDAR degradation (Fig. 15), radar-based motion cues support accurate beam and blockage inference. Under radar degradation (Fig. 16), vision- and LiDAR-based models are selected; although fine-grained motion information is reduced, blockage duration and handover conditions are correctly evaluated.

## APPENDIX G BEAM PREDICTION AGENT PERFORMANCE

This section presents Table IX which reports the beam prediction agent’s performance from  $t+1$  to  $t+5$ , including Top-3 accuracy and average APL across modality configurations.

## APPENDIX H BLOCKAGE PREDICTION AGENT PERFORMANCE

This section presents Table X which summarizes the blockage prediction agent’s performance from  $t+1$  to  $t+5$ , reporting F1-score and AUC-ROC for each modality combination.

TABLE IX  
BEAM PREDICTION AGENT TOP-3 PERFORMANCE FOR  $t + \{1, \dots, 5\}$  WITH  $[M = 16, Q = 64]$

Modality	Beam Prediction									
	$t + 1$		$t + 2$		$t + 3$		$t + 4$		$t + 5$	
	Acc.	APL	Acc.	APL	Acc.	APL	Acc.	APL	Acc.	APL
camera_gps_lidar	-0.009220	88.5%	-0.009338	88.2%	-0.009474	87.9%	-0.009511	87.6%	-0.009540	87.1%
camera_radar_lidar	-0.009314	88.3%	-0.009498	88.1%	-0.009576	87.9%	-0.009651	87.4%	-0.009723	87.0%
camera_gps_radar_lidar	-0.009660	85.8%	-0.009723	85.5%	-0.009784	85.1%	-0.009842	84.7%	-0.009899	84.3%
camera_lidar	-0.009668	85.6%	-0.009719	85.3%	-0.009774	85.0%	-0.009828	84.6%	-0.009881	84.2%
camera_gps_radar	-0.014690	84.9%	-0.014782	84.6%	-0.014873	84.2%	-0.014962	83.8%	-0.015050	83.3%
camera_radar	-0.021250	84.2%	-0.021395	83.8%	-0.021537	83.4%	-0.021676	83.0%	-0.021813	82.6%
camera_gps	-0.028440	83.7%	-0.028592	83.3%	-0.028741	82.9%	-0.028888	82.5%	-0.029032	82.1%
gps_lidar_radar	-0.035530	83.0%	-0.035712	82.6%	-0.035890	82.2%	-0.036065	81.7%	-0.036238	81.3%
camera_only	-0.037050	82.8%	-0.037242	82.4%	-0.037430	82.0%	-0.037615	81.6%	-0.037797	81.2%
gps_radar	-0.039790	81.9%	-0.039972	81.5%	-0.040152	81.1%	-0.040330	80.7%	-0.040505	80.3%
gps_lidar	-0.080130	76.6%	-0.080487	76.1%	-0.080840	75.6%	-0.081190	75.1%	-0.081537	74.6%
radar_lidar	-0.142700	74.4%	-0.143154	73.8%	-0.143603	73.3%	-0.144048	72.8%	-0.144488	72.3%
radar_only	-0.160200	72.4%	-0.160734	71.9%	-0.161263	71.3%	-0.161787	70.7%	-0.162306	70.1%
lidar_only	-0.174300	65.7%	-0.175029	64.9%	-0.175751	64.1%	-0.176468	63.3%	-0.177179	62.5%
gps_only	-0.215100	59.0%	-0.216070	58.0%	-0.217030	57.0%	-0.217980	56.0%	-0.218920	55.0%

TABLE X  
BLOCKAGE AGENT PERFORMANCE FOR  $t + \{1, \dots, 5\}$

Modality	Blockage Prediction									
	$t + 1$		$t + 2$		$t + 3$		$t + 4$		$t + 5$	
	F1 Score	AUC-ROC	F1 Score	AUC-ROC	F1 Score	AUC-ROC	F1 Score	AUC-ROC	F1 Score	AUC-ROC
camera_radar	98.4%	0.988	98.0%	0.985	97.9%	0.982	97.5%	0.971	97.2%	0.968
camera_only	98.1%	0.983	97.8%	0.981	97.5%	0.970	97.2%	0.969	97.1%	0.963
camera_lidar	96.2%	0.955	96.1%	0.947	95.7%	0.939	95.5%	0.935	95.5%	0.933
camera_gps	94.4%	0.931	94.1%	0.930	93.9%	0.927	93.9%	0.927	93.8%	0.924
camera_radar_lidar	94.0%	0.928	93.9%	0.926	93.8%	0.925	93.7%	0.922	93.7%	0.920
radar_only	93.7%	0.922	93.7%	0.921	93.6%	0.919	93.5%	0.916	93.5%	0.915
camera_gps_radar_lidar	93.1%	0.914	93.0%	0.913	92.7%	0.910	92.2%	0.909	92.0%	0.909
camera_gps_radar	92.3%	0.911	92.2%	0.907	92.0%	0.904	91.8%	0.902	91.7%	0.898
radar_lidar	91.8%	0.902	91.6%	0.899	91.5%	0.899	91.4%	0.896	91.3%	0.891
camera_gps_lidar	91.2%	0.897	91.1%	0.895	91.1%	0.894	90.8%	0.890	90.8%	0.890
gps_lidar_radar	89.9%	0.889	89.8%	0.887	89.5%	0.885	89.4%	0.881	89.0%	0.879
gps_radar	89.3%	0.880	89.2%	0.879	89.0%	0.877	88.9%	0.875	88.6%	0.873
lidar_only	87.9%	0.872	87.9%	0.872	87.7%	0.870	87.4%	0.868	87.3%	0.865
gps_lidar	84.1%	0.855	84.0%	0.852	83.6%	0.849	83.2%	0.847	83.0%	0.831
gps_only	61.7%	0.603	61.4%	0.600	60.9%	0.599	60.7%	0.592	60.6%	0.588

## APPENDIX I

### DRL AGENT SELECTIONS AND REWARD DISTRIBUTION

This section compares the DRL agent’s selection frequencies to two baselines: a random policy that samples modality combinations uniformly, and a rule-based policy that selects the next available combination from a predefined priority list. Each policy’s selection frequencies are seen in Fig. 17.

Fig. 18 also reports the DRL reward distribution across episodes. The DRL agent has an average reward of 1.45, with a peak near 1.5 and a maximum of 1.6. The cluster of rewards in the upper range indicates consistent high-performing modality configuration selections under varying degradations.

## APPENDIX J

### FULL ABLATION STUDY OF LLM SIZES AND MODELS ACROSS MODALITY COMBINATIONS

This section presents a comprehensive ablation study of ENWAR 3.0 across modality configurations and LLM scales.

The first study evaluates 13 configurations that combine single-pass CoT reasoning (CoT), LLM priming (Primed),

long-term memory (Mem), and the DRL agent (DRL), alongside a vanilla baseline using static prompts without structured reasoning, memory, or policy-based selections (Fig. 19). Under full-modality inclusion (C+G+R+L), ENWAR 3.0 achieves 87.3% reasoning correctness with LLaMa3.2-3B, a 17.9% improvement over vanilla (69.4%). Scaling to DeepSeek-r1-70B increases correctness to 89.3%, indicating modest gains from larger models.

Component ablations show that structured reasoning drives most improvements: CoT (82.2%) and Primed (82.5%) substantially outperform vanilla prompting, while Mem (81.9%) and DRL (78.3%) contribute temporal consistency and degradation-aware routing. The CoT+Mem+DRL configuration reaches 84.9%, demonstrating that structured reasoning with temporal and policy awareness forms a strong orchestration backbone even without supervised priming. Adding priming to Mem+DRL increases performance to 86.0%, and the full system reaches 87.3%. Across modality combinations, correctness increases with sensor richness, and the gap between vanilla and the full system widens under triple- and quadruple-modality inputs, affirming that orchestration tactics become

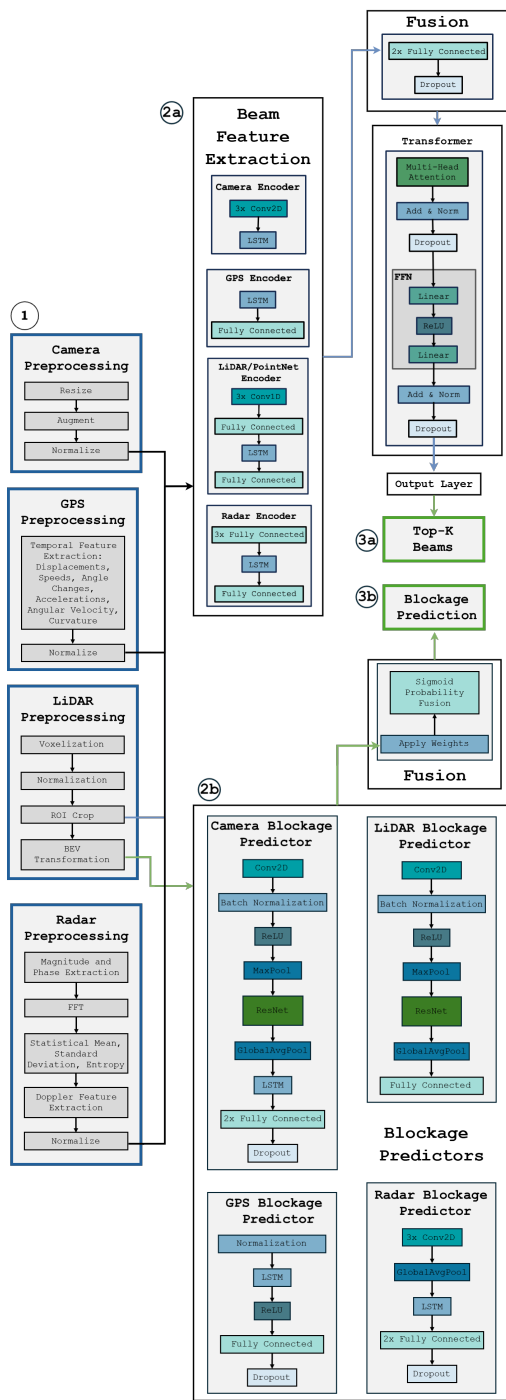


Fig. 11. Architecture of the beam and blockage prediction models.

increasingly valuable as multi-modal complexity grows.

The second study evaluates reasoning correctness across model sizes and architectures (Fig. 20), including Qwen2.5-3B, LLaMa3.2-3B, DeepSeek-r1-8B, LLaMa3.1-8B, Qwen3-32B, DeepSeek-r1-32B, LLaMa3.3-70B, and DeepSeek-r1-70B. While performance improves modestly with parameter scale, the gains diminish beyond mid-sized models. When considering the first evaluation, sensor richness contributes more to reasoning quality than raw parameter count. These results suggest that compact models, when paired with strong multi-modal perception and structured orchestration, indeed offer an efficient and practical tradeoff for real-time deployment.

## APPENDIX K COMPUTATIONAL, DEPLOYMENT, AND RESOURCE CONSIDERATIONS AND LIMITATIONS

While ENWAR 3.0 enables context-aware multi-agent reasoning, it introduces computational considerations, including critical-path latency, orchestration overhead, and memory footprint. All evaluations were conducted on an NVIDIA A100 GPU (40GB VRAM), reflecting an edge-capable BS deployment setting. We employ parameter-efficient techniques, including LoRA fine-tuning, 4-bit quantization of transformer-based agents, GPU affinity scheduling, caching, and asynchronous execution to minimize overhead.

The inference pipeline operates over a sliding window of  $\Delta T = 1.5s$  with sensor updates every  $T_s = 300ms$ . We distinguish between (i) the time-critical control path and (ii) asynchronous explanation generation. Upon new input arrival, perception modules, the environment classifier, and the DRL policy execute in parallel, while the LLM constructs a structured invocation packet specifying agent selection and network updates. For the primed 3B model, worst-case invocation latency is 222.7ms (Table VII). Beam and blockage agents exhibit worst-case single-sample latencies of 66.98ms and 59.47ms, respectively (Table III), where inference times include modality-specific preprocessing.

Since downstream agents execute in parallel, the worst-case control-path latency is bounded by invocation time plus the longest agent, yielding approximately  $222.7 + 66.98 \approx 289.7ms$ . This latency remains within the 300ms sampling interval, preventing latency accumulation across sliding windows and confirming real-time compatibility for I2V beamforming and handover scheduling.

Priming substantially reduces orchestration overhead: invocation and memory updates with a primed 3B model complete in 222.7ms, compared to 618.8ms for a non-primed 3B model using static prompt templates. Even the largest 70B model maintains bounded invocation latency (242.6ms). In contrast, response generation (2.17–3.08s depending on model size) is fully decoupled from the control loop, ensuring that network actions do not wait for natural-language output.

The architecture supports scalable deployment via agent containerization, enabling microservice-based execution across edge infrastructure. Memory updates are event-driven (e.g., degradation shifts or prediction divergence), reducing unnecessary compute load. While fusion-heavy configurations approach the latency bound, multi-user deployments may require hardware scaling as model complexity increases.

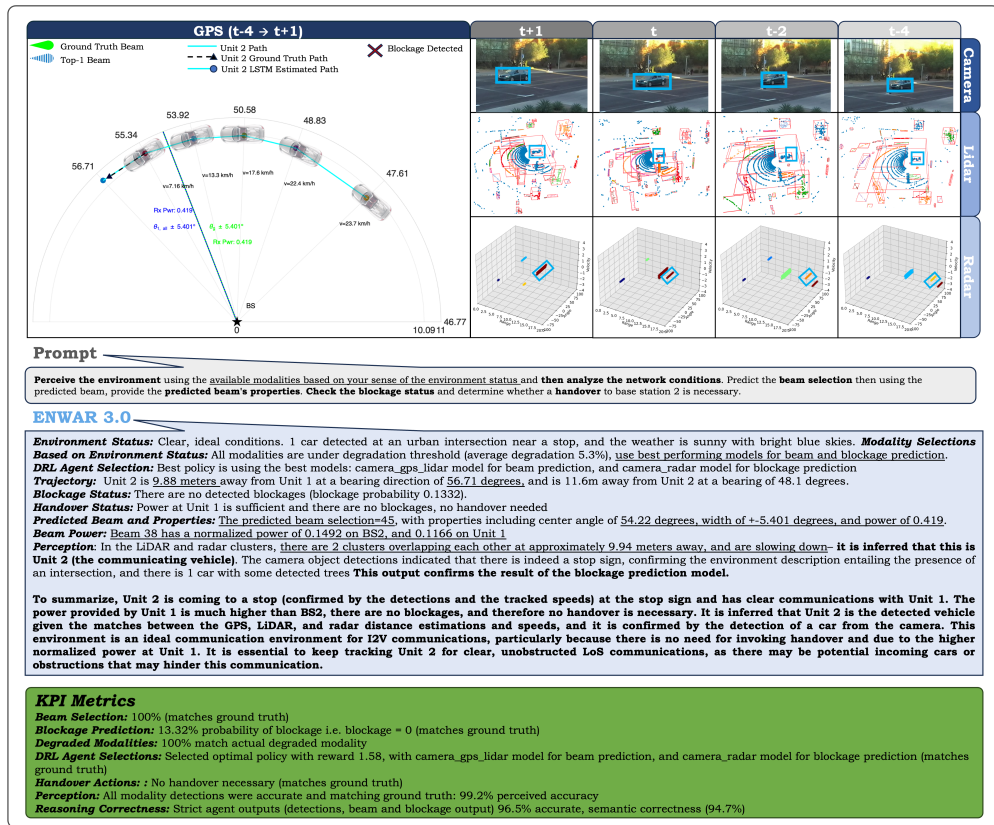


Fig. 12. An example of ENWAR 3.0's response to an ideal, and clear communications environment

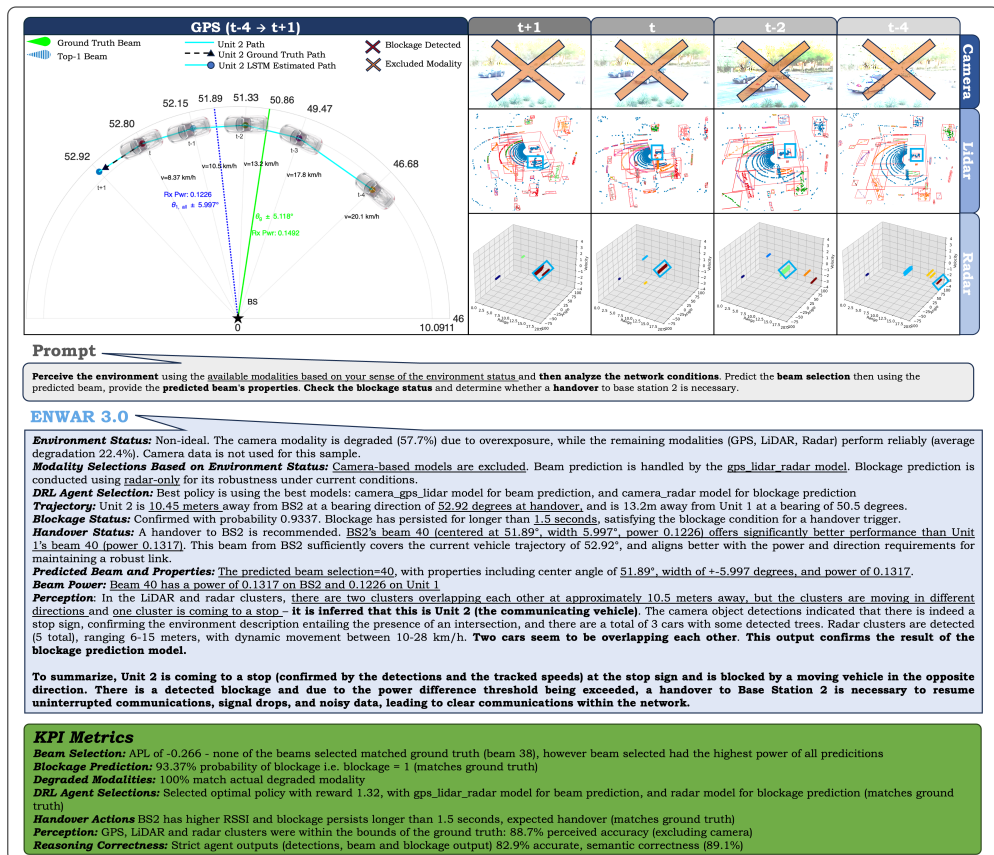


Fig. 13. An example of ENWAR 3.0's response with the camera sensor flagged as degraded.

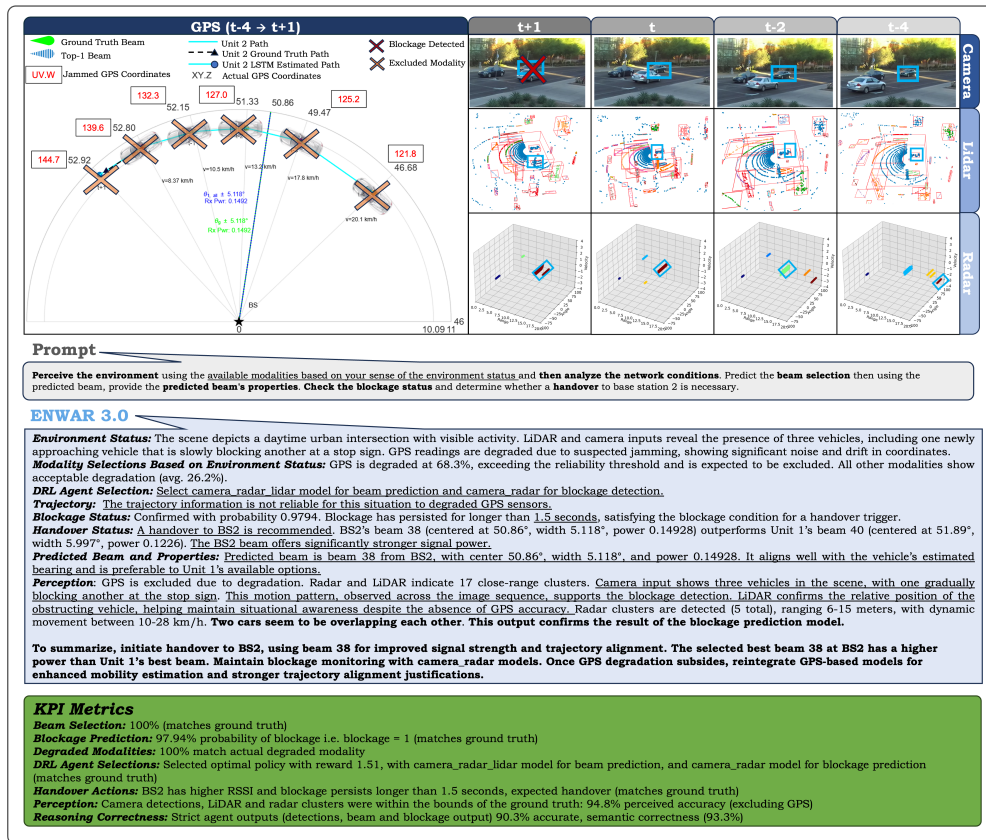


Fig. 14. An example of ENWAR 3.0's response with the GPS sensor flagged as degraded.

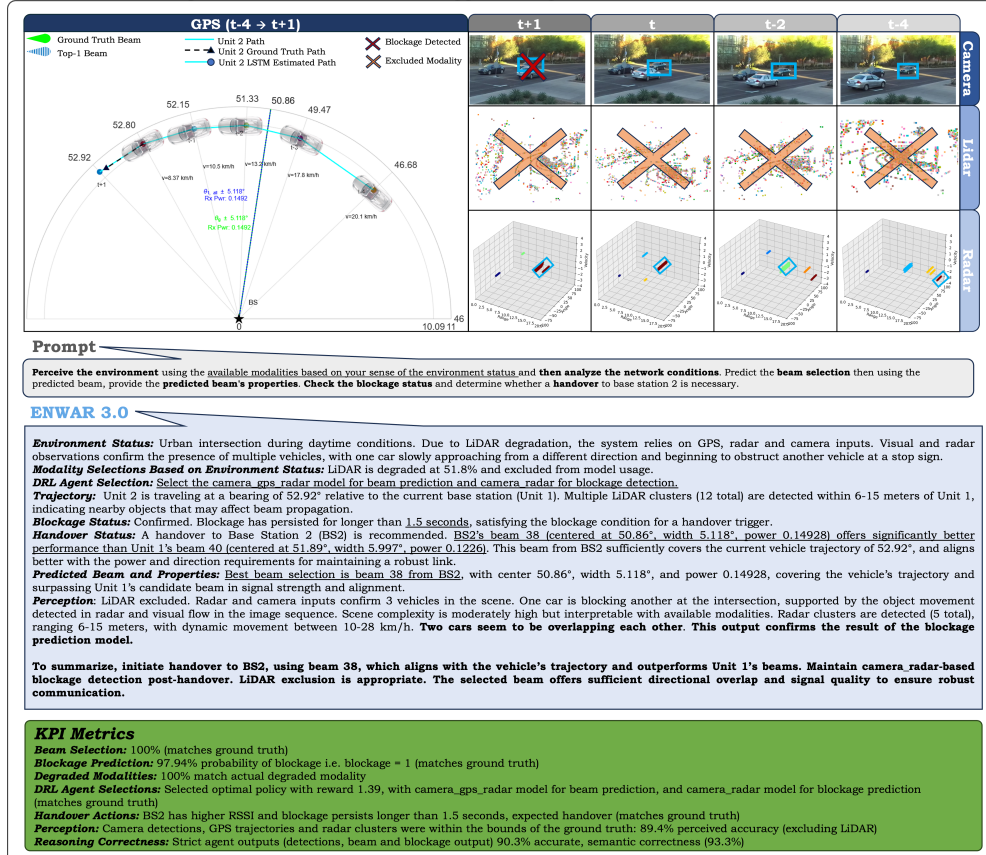


Fig. 15. An example of ENWAR 3.0's response with the Lidar sensor flagged as degraded.

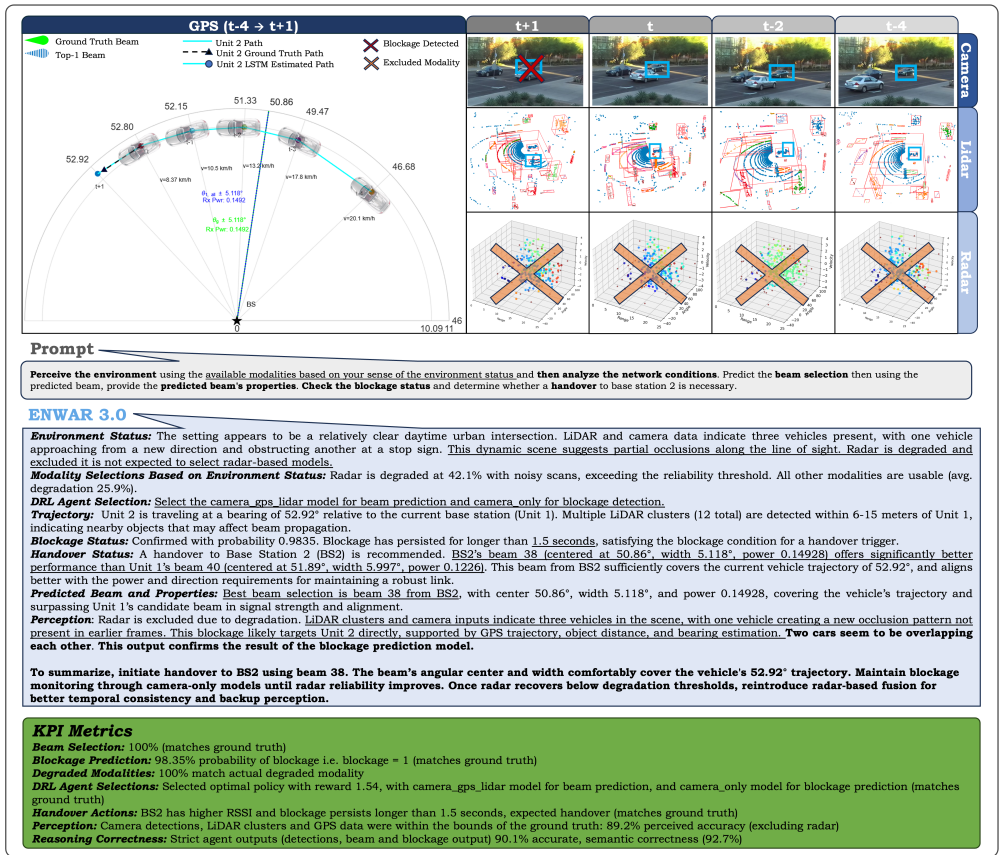


Fig. 16. An example of ENWAR 3.0's response with the radar sensor flagged as degraded.

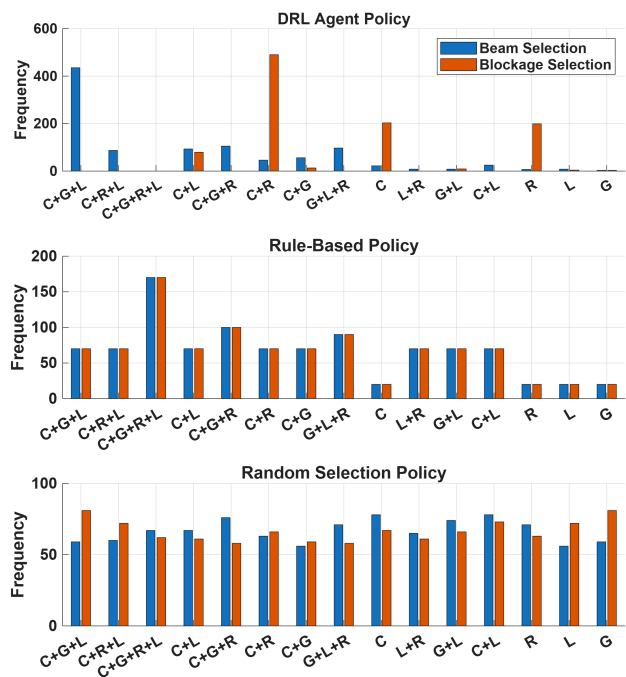


Fig. 17. Frequency of DRL agent selections relative to the baselines.

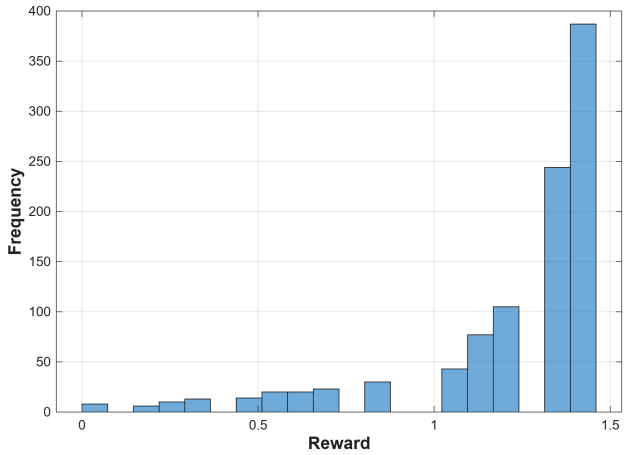


Fig. 18. Reward distribution of PPO across 1000 episodes

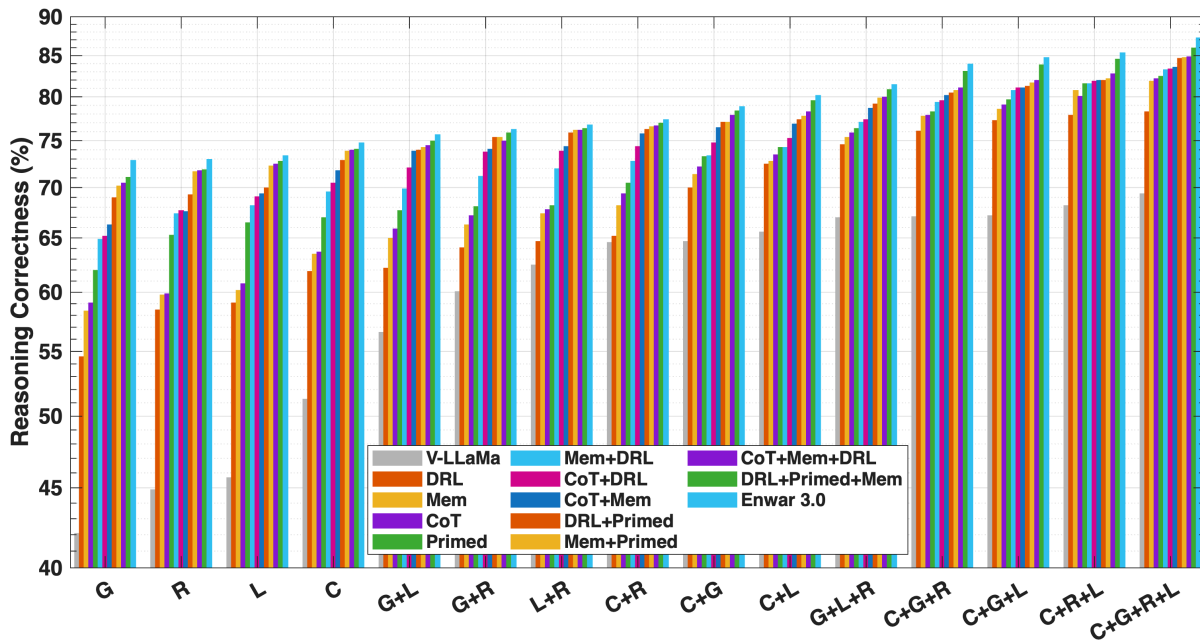


Fig. 19. Comparison of reasoning correctness scores to the vanilla LLaMa baseline and different combinations of including long-term memory (Mem), the DRL agent (DRL), and LLM priming to ENWAR 3.0 across all sensor modality combinations (C: camera, G: GPS, L: LiDAR, R: radar).

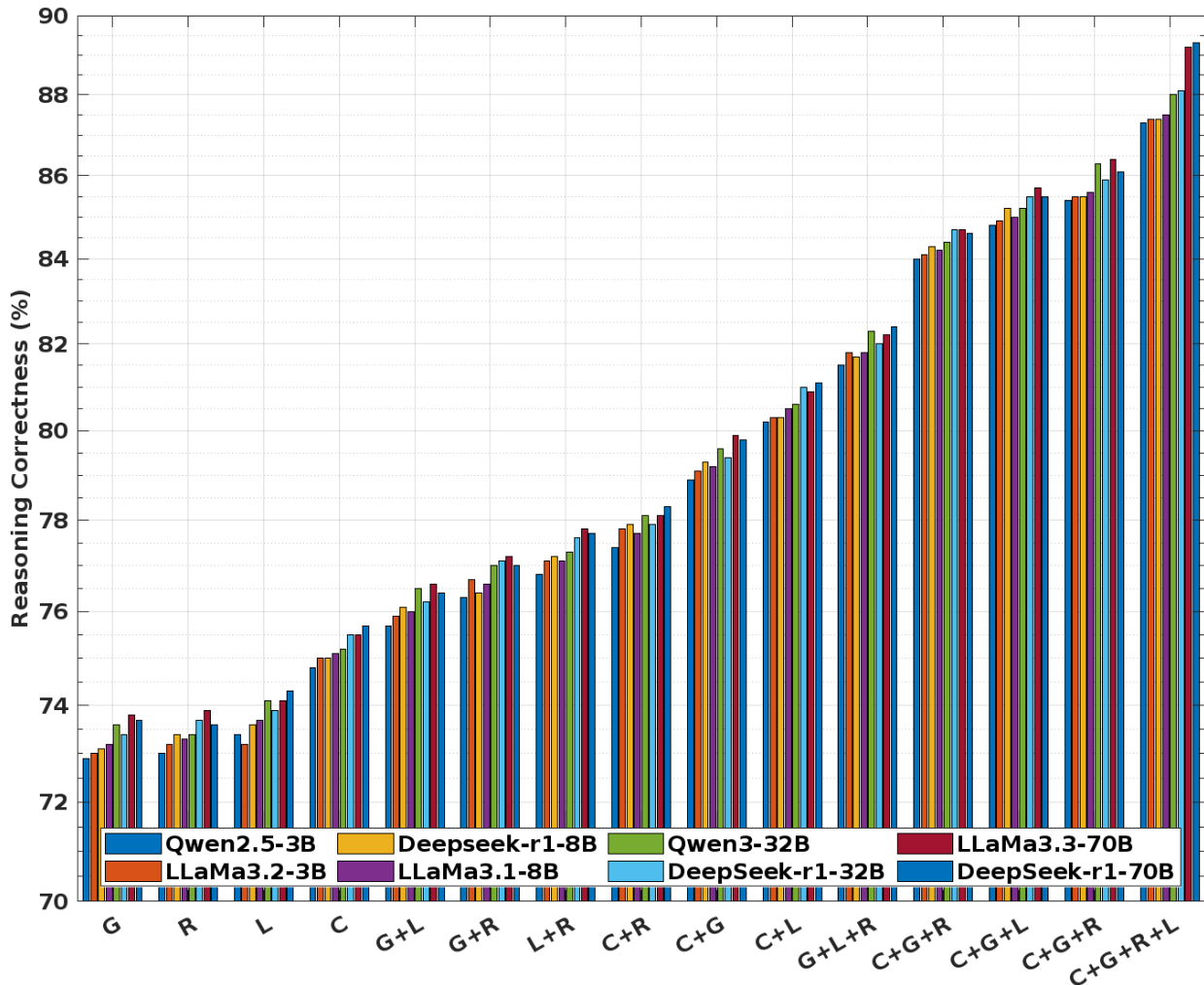


Fig. 20. Reasoning correctness across different sized LLMs ranging from (3-70)B parameters and across all sensor modality combinations (C: camera, G: GPS, L: LiDAR, R: radar).