



(RESEARCH ARTICLE)



EDGE-BASED LLM INFERENCE FOR SMART CITIES

Ramesh Tomar, Kamal Kushwah, Gita Shivhare and Satish Nalajala

Department of Information Technology, Atal Bihari Vajpayee IIITM, Gwalior, India.

* Corresponding Author

International Journal of Science and Research Archive, 2024, 13(02), 1684–1691

Publication history: Received on 22 September 2024; revised on 22 December 2024; accepted on 29 December 2024

Article DOI: <https://doi.org/10.30574/ijrsra.2024.13.2.2415>

Copyright © 2024 Author(s) retain the copyright of this article. This article is published under the terms of the Creative Commons Attribution License 4.0

ABSTRACT

Large Language Models (LLMs) have demonstrated transformative capabilities in natural language understanding, multi-modal reasoning, and autonomous decision support, making them compelling candidates for smart city management systems. However, the multi-billion parameter scale of frontier LLMs renders cloud-exclusive inference architectures inadequate for latency-sensitive smart city services including real-time emergency dispatch reasoning, adaptive traffic signal control, and citizen query resolution, where round-trip cloud latency and data sovereignty constraints together prohibit centralized processing. This paper proposes the Edge-Based LLM Inference Framework for Smart Cities (ELIS-C) that enables low-latency, privacy-preserving, context-aware LLM inference across a hierarchical edge-fog-cloud continuum through three principal innovations. First, a speculative decoding pipeline paired with a city-domain-fine-tuned small language model (SLM) draft network accelerates token generation at edge nodes by a factor of 3.1x compared to standard autoregressive decoding while maintaining output equivalence with the full target LLM. Second, an Adaptive Model Partitioning Engine (AMPE) dynamically splits transformer layers between edge nodes and fog servers based on real-time resource telemetry, query complexity classification, and data sensitivity tags, minimizing end-to-end time-to-first token (TTFT) latency. Third, a Retrieval-Augmented Generation (RAG) module grounded in a continuously updated smart city knowledge graph incorporating live sensor feeds, municipal databases, and real-time event streams ensures factually grounded LLM responses without cloud transmission of raw citizen data. Security provisions include prompt injection detection, differential privacy for context retrieval, and hardware-attested model integrity. Experimental evaluation across three smart city LLM application domains demonstrate a 52% reduction in median TTFT latency, a 61% reduction in cloud backhaul bandwidth, a 94.3% response factual accuracy, and full compliance with GDPR data residency requirements, significantly outperforming cloud-only and static edge partition baselines.

Keywords: Edge Computing; Large Language Models; Smart Cities; Speculative Decoding; Retrieval-Augmented Generation; Model Partitioning; Differential Privacy; Prompt Injection Detection.

1 INTRODUCTION

The smart city paradigm envisions urban environments in which heterogeneous sensor networks, interconnected municipal systems, and AI-driven analytics converge to optimize resource allocation, enhance public safety, streamline citizen services, and reduce environmental impact [1]. By 2024, more than 500 cities worldwide have initiated formal smart city programmes, deploying billions of IoT endpoints spanning traffic management systems, environmental monitoring networks, emergency services dispatch infrastructure, and public transit optimization platforms [2]. The

operational intelligence layer of these deployments has historically been implemented through narrow, task-specific machine learning models requiring separate training pipelines, deployment stacks, and maintenance cycles for each urban application domain — a fragmented approach that multiplies operational complexity and inhibits cross-domain reasoning that cuts across municipal function boundaries.

Large Language Models have emerged as a compelling alternative by offering general-purpose reasoning, natural language understanding, and multi-domain knowledge within a single unified model architecture [3]. An LLM deployed within a city operational intelligence layer could simultaneously process citizen queries in natural language, synthesize anomaly reports from sensor streams, generate explanatory narratives accompanying automated traffic rerouting decisions, support emergency dispatchers with situation assessment, and interface with municipal staff through conversational dashboards. However, realizing this potential in production smart city deployments encounters three fundamental technical barriers that cloud-centric LLM deployment cannot address [4].

The first barrier is latency. Emergency dispatch assistance and real-time traffic control require sub-second response latency; cloud-hosted LLM inference at the 7B-70B parameter scale incurs 1.5-8 seconds of end-to-end latency due to network round-trips and shared inference queue contention [5]. The second barrier is data sovereignty. Smart city sensor data frequently captures personally identifiable information — vehicle trajectories, pedestrian movements, and utility consumption patterns — that GDPR and equivalent national privacy regulations prohibit from leaving designated geographic processing boundaries [6]. Transmitting raw city sensor streams to cloud LLM APIs violates these requirements by design. The third barrier is reliability. Smart city services are critical infrastructure; dependence on cloud connectivity introduces single points of failure unacceptable for emergency management applications that must remain operational during network disruptions [7].

Edge-based LLM inference addresses all three barriers simultaneously by deploying quantized LLM variants on edge and fog hardware co-located within city infrastructure — 5G base stations, traffic management centers, and municipal data hubs — where inference proceeds independently of wide-area network connectivity, within geographic data residency boundaries, and at local processing latency rather than cloud round-trip times. This paper addresses the associated deployment challenges through the ELIS-C framework, making the following principal contributions:

- ELIS-C, a formally specified edge-based LLM inference framework for smart cities integrating speculative decoding acceleration, adaptive model partitioning, RAG-based factual grounding, and a multi-tier data governance and security architecture compliant with GDPR data residency requirements.
- A city-domain speculative decoding pipeline in which a 350M-parameter SLM draft network fine-tuned on a 4.2 billion token smart city corpus proposes token sequences verified in parallel by the full target LLM, achieving a 3.1x mean token throughput improvement with mathematically guaranteed output distribution equivalence to standard autoregressive decoding.
- An Adaptive Model Partitioning Engine (AMPE) that dynamically assigns transformer layer subsets to edge nodes and fog servers based on a multi-attribute optimization function balancing TTFT latency, memory pressure, data sensitivity classification, and inter-tier communication bandwidth, enabling inference across models up to 13B parameters on heterogeneous edge-fog hardware.
- A continuously updated smart city knowledge graph RAG module that retrieves context from live sensor feeds, municipal event databases, and real-time transit and emergency records using privacy-preserving approximate nearest-neighbor search with differentially private query embeddings, grounding LLM outputs in current city state without raw data cloud transmission.
- A layered security architecture incorporating prompt injection detection via a fine-tuned DeBERTa-v3 classifier, adversarial context poisoning detection in the RAG retrieval pipeline, and TPM 2.0-rooted model integrity attestation, addressing the novel threat surface introduced by LLM deployment in public-facing smart city services.

The rest of this paper is organized as follows. Section II presents the formal system model. Section III details the ELIS-C scheme design and analysis. Section IV provides performance evaluation. Section V concludes the paper.

2 SYSTEM MODEL FOR EDGE-BASED LLM INFERENCE FRAMEWORK FOR SMART CITIES (ELIS-C)

2.1 Three-Tier Architecture, Formal Definitions, and Security Assumptions

The ELIS-C architecture is deployed across three computational tiers, each with distinct hardware profiles, latency characteristics, and data governance responsibilities tailored to the smart city operational environment:

- Edge Tier: Comprises E geographically distributed edge nodes $\{e_1, e_2, \dots, e_E\}$ co-located with city infrastructure including traffic management cabinets, 5G Multi-Access Edge Computing (MEC) nodes, and

municipal IoT gateway hubs. Each edge node e_i is equipped with an AI inference accelerator characterized by resource profile $R_i = (\text{MEM}_i, \text{TOPS}_i, \text{BW}_{\text{up}_i}, \text{Power}_i)$, where MEM_i is available DRAM (16-64 GB), TOPS_i is INT4 inference throughput, BW_{up_i} is uplink bandwidth to the fog tier, and Power_i is the thermal design power envelope. Edge nodes execute the SLM draft network, the first L_{edge} transformer layers of the target LLM, and the RAG retrieval module against a locally cached knowledge graph shard.

- **Fog Tier:** Comprises F regional fog servers $\{f_1, f_2, \dots, f_F\}$ hosted in city district data centers or 5G core co-location facilities, each serving a cluster of edge nodes. Fog servers host high-memory GPU accelerators capable of housing the remaining L_{fog} transformer layers of the partitioned target LLM. The fog tier performs LLM verification in speculative decoding, AMPE layer partition optimization, and intra-district knowledge graph aggregation and freshness maintenance.
- **Cloud Tier:** A city-operated cloud or national sovereign cloud providing global LLM model registry, fleet-wide knowledge graph synchronization, long-horizon analytics, and regulatory compliance audit logging. Cloud inference is reserved for non-latency-sensitive background tasks and model fine-tuning, no raw citizen data transits the cloud tier boundary.

The formal algorithmic components of ELIS-C are defined as follows:

- **SystemInit(E, F, k_{sec})** $\rightarrow$ ($\text{Gamma}_{\text{city}}, \text{Theta}_{\text{LLM}}, \text{Theta}_{\text{SLM}}, \text{KG}_0$): Initializes city system configuration $\text{Gamma}_{\text{city}}$ (domain taxonomy, GDPR residency boundaries, sensitivity rules), quantized target LLM weights $\text{Theta}_{\text{LLM}}$ partitioned across the edge-fog boundary, SLM draft network weights $\text{Theta}_{\text{SLM}}$, and initial smart city knowledge graph KG_0 , given E edge nodes, F fog servers, and security parameter k_{sec} .
- **ClassifyQuery(q)** $\rightarrow$ (c_q, s_q, d_q): For each incoming natural language query q , a lightweight classifier extracts the application domain c_q in $\{\text{traffic, emergency, environment, citizen services}\}$, data sensitivity class s_q in $\{\text{public, restricted, confidential}\}$, and estimated decoding complexity d_q , informing AMPE partitioning and RAG retrieval scope decisions.
- **PartitionLayers(R_t, c_q, s_q, d_q)** $\rightarrow$ ($L_{\text{edge}}^*, L_{\text{fog}}^*$): AMPE solves the layer partition optimization problem given current per-tier resource telemetry R_t and query attributes, returning the optimal edge-fog layer split minimizing expected TTFT subject to memory capacity and data sovereignty routing constraints.
- **RetrieveContext($q, \text{KG}_t, \text{epsilon}_{\text{DP}}$)** $\rightarrow$ C_q : The RAG module embeds query q using a privacy-amplified sentence encoder with Gaussian noise calibrated to $(\text{epsilon}_{\text{DP}}, \text{delta})$ -DP, performs approximate nearest-neighbor search over the local knowledge graph shard KG_t , and returns the top- K retrieved context passages C_q without exposing raw sensor readings or citizen records to the LLM inference pipeline.
- **SpeculativeDecode($q, C_q, \text{Theta}_{\text{SLM}}, \text{Theta}_{\text{LLM}}, \text{gamma}$)** $\rightarrow$ y^* : The SLM draft network generates a draft token sequence of length gamma ; the target LLM verifies all $\text{gamma}+1$ positions in a single parallel forward pass, accepting tokens matching the target distribution and resampling rejected positions, producing final output y^* with guaranteed distributional equivalence to standard autoregressive decoding.
- **AttestModel($\text{Theta}_{\text{LLM}}, \text{Theta}_{\text{SLM}}, \text{TPM}$)** $\rightarrow$ ($\text{status}, \text{cert}$): The TPM 2.0-rooted attestation module measures all LLM and SLM weight file hashes against certified reference values, returning an integrity status and signed attestation certificate for audit logging and regulatory compliance demonstration.

Table 1: Notation and Descriptions

Symbol	Description
E, F	Number of edge nodes and fog servers in the ELIS-C deployment
e_i	i -th edge node with resource profile $R_i = (\text{MEM}, \text{TOPS}, \text{BW}_{\text{up}}, \text{Power})$
f_j	j -th fog server hosting L_{fog} target LLM transformer layers
$\text{Gamma}_{\text{city}}$	City system configuration: domain taxonomy, GDPR boundaries, sensitivity rules
$\text{Theta}_{\text{LLM}}$	Quantized target LLM weights (INT4, 7B or 13B parameters)
$\text{Theta}_{\text{SLM}}$	City-domain SLM draft network weights (350M parameters, FP16)
KG_t	Smart city knowledge graph at time t : sensor feeds, municipal databases, events
q	Incoming natural language query from citizen, operator, or automated system
c_q	Application domain class of query q : $\{\text{traffic, emergency, environment, services}\}$
s_q	Data sensitivity class of query q : $\{\text{public, restricted, confidential}\}$
d_q	Estimated decoding complexity of query q (approximate output token count)

R _t	Per-tier resource telemetry vector at AMPE optimization epoch t
L _{edge} [*] , L _{fog} [*]	AMPE-optimized edge and fog transformer layer partition counts
C _q	RAG-retrieved context passages for query q from knowledge graph KG _t
epsilon _{DP}	Differential privacy budget for RAG query embedding noise injection
gamma	Speculative decoding draft length: number of SLM-proposed tokens per step
y [*]	Final LLM output with guaranteed distributional equivalence to autoregressive decoding
TTFT	Time-To-First-Token: end-to-end latency from query submission to first output token
k _{sec}	Security parameter for cryptographic and TPM 2.0 attestation operations

2.2 Data Governance and GDPR Compliance Model

ELIS-C embeds data governance as a structural architectural property rather than a policy overlay. The sensitivity classification assigned to each query by ClassifyQuery determines a mandatory processing boundary: queries classified as restricted or confidential are processed exclusively within the edge and fog tiers located within the GDPR-designated data residency zone, with a hardware-enforced network egress filter blocking any data plane traffic containing classified content identifiers from crossing the fog-cloud boundary. Public-class query responses may be forwarded to cloud analytics for aggregate quality monitoring, with all citizen-identifiable attributes removed by a de-identification module before transmission. The differential privacy mechanism applied to RAG query embeddings ensures that even the embedding vector transmitted from edge to fog for nearest-neighbor search cannot be reversed to reconstruct the original query content, providing privacy amplification beyond the per-query data isolation provided by residency boundary enforcement.

2.3 Security and Adversarial Robustness Model

The LLM deployment surface in public-facing smart city services introduces three novel adversarial threat vectors. Prompt injection attacks — in which malicious inputs embedded in citizen queries or retrieved RAG context attempt to override LLM system instructions — are detected by a fine-tuned DeBERTa-v3-small classifier operating as a pre-processing gate that blocks detected injection attempts before they reach the LLM inference pipeline, achieving 98.1% detection accuracy and 1.3% false positive rate. Context poisoning attacks that inject falsified data into city sensor feeds to corrupt RAG retrieval are mitigated by a source credibility scoring module implementing anomaly detection on sensor value distributions before knowledge graph ingestion. Model integrity is protected by TPM 2.0 attestation of all LLM and SLM weight files at boot and at each model update event, preventing unauthorized model substitution that could introduce backdoors into city decision support systems.

3 ELIS-C SCHEME DESIGN AND ANALYSIS

3.1 City-Domain Speculative Decoding Pipeline

Speculative decoding provides a mathematically rigorous mechanism for accelerating autoregressive LLM inference by exploiting the observation that a smaller, faster draft model can predict the target LLM outputs with high acceptance rates when both models share a domain vocabulary and knowledge base [9]. In ELIS-C, the SLM draft network is a 350M-parameter transformer fine-tuned from a pre-trained base model on a 4.2 billion token smart city corpus comprising municipal government documents, traffic engineering standards, emergency response protocols, environmental monitoring reports, and anonymized citizen service interaction logs. Fine-tuning on city-domain text increases the SLM draft acceptance rate from 61.3% for a general-domain SLM to 78.4% for city-domain queries, significantly improving speculative decoding efficiency.

At inference time, the SLM generates a draft sequence of $\gamma = 5$ tokens per speculative step on the edge node, requiring 0.8ms per step at INT8 precision on the edge NPU. The target LLM (13B parameters, INT4-quantized, partitioned across edge and fog) verifies all $\gamma+1$ candidate positions in a single batched forward pass requiring 4.2ms, accepting draft tokens whose sampled probability under the target distribution satisfies the rejection sampling correction threshold. The expected number of accepted tokens per speculative step is $E[n_{\text{accept}}] = \gamma * \alpha_{\text{mean}} = 5 * 0.784 = 3.92$ accepted tokens per 4.2ms verification pass versus 1 token per 4.2ms in standard autoregressive decoding — a 3.1x throughput improvement with zero degradation in output quality, as the rejection sampling correction guarantees exact distributional equivalence to the target LLM standard autoregressive output.

3.2 Adaptive Model Partitioning Engine (AMPE)

The AMPE solves the following optimization problem at each query arrival to minimize $TTFT(L_{edge}, L_{fog}) = T_{edge}(L_{edge}) + T_{comm}(act_{\{L_{edge}\}}) + T_{fog}(L_{fog})$, subject to memory capacity constraints on both tiers, the completeness constraint $L_{edge} + L_{fog} = L_{total}$, and the data sovereignty constraint that queries with sensitivity class confidential must be routed exclusively to tiers within the designated residency zone. T_{edge} and T_{fog} are estimated execution times computed from per-layer FLOP counts and current per-tier throughput telemetry. T_{comm} is the inter-tier activation tensor transmission time computed from the activation tensor size at the partition layer and current measured edge-to-fog bandwidth.

The optimization problem is solved greedily in $O(L_{total})$ time by sweeping the partition point and evaluating the TTFT objective at each point using current resource telemetry, selecting the minimum-TTFT feasible partition. AMPE re-optimizes every 500ms or upon detection of resource utilization exceeding 80% on any tier, adapting to dynamic load fluctuations through a shadow partitioning protocol that applies new partitions to subsequent queries while completing current queries under the previous partition without disruption.

3.3 Smart City Knowledge Graph RAG Module

The smart city knowledge graph KG_t is a continuously updated property graph implemented in Apache AGE comprising five node types: SensorReading (live IoT data with 30-second freshness), MunicipalEvent (permits, incidents, infrastructure maintenance), TransitStatus (real-time public transport positions and delays), CitizenService (anonymized service request metadata), and Regulation (municipal bylaws, zoning rules, emergency protocols). Graph edges encode spatial proximity, temporal co-occurrence, causal relationships, and administrative jurisdiction. The graph is updated by a stream processing pipeline consuming city IoT event buses at up to 50,000 events per second, with incremental graph updates triggered by each ingest event rather than full recomputation.

RAG retrieval for an incoming query q proceeds in three stages. First, the query is embedded using a city-fine-tuned sentence transformer with Gaussian noise added to achieve $(\epsilon_{DP} = 0.8, \delta = 10^{-6})$ -DP. Second, approximate nearest-neighbor search using the HNSW index over the local knowledge graph shard returns the top- $K = 8$ most relevant graph nodes and their two-hop neighborhoods. Third, a cross-encoder re-ranking model scores retrieved passages against the original query and selects the top-3 for inclusion in the LLM context window. The full RAG pipeline adds 11ms to total TTFT — a modest overhead given the 52% overall latency reduction achieved by the combined speculative decoding and AMPE optimizations.

4 PERFORMANCE ANALYSIS

4.1 Experimental Setup

ELIS-C was evaluated across three smart city LLM application domains on a physical testbed comprising 6 NVIDIA Jetson AGX Orin edge nodes (64GB LPDDR5, 275 TOPS INT8), 2 NVIDIA A100 80GB fog servers, and a simulated cloud tier. The target LLM was Llama-2-13B-Chat quantized to INT4 using GPTQ (3.5 GB, distributed across 2 edge nodes using tensor parallelism), with the 350M-parameter SLM draft network derived by structured pruning and domain fine-tuning. The smart city knowledge graph was populated with 18 months of historical data and live synthetic event streams from the CityBench urban simulation framework, comprising 4.7 million graph nodes and 31.2 million edges across five node categories.

The three evaluation domains were: (1) Emergency Dispatch Assistance — LLM supporting dispatch operators with real-time incident situation assessment, resource recommendation, and protocol guidance across 200 simulated emergency scenarios; (2) Adaptive Traffic Query Resolution — LLM answering natural language queries from traffic management operators about current congestion patterns and rerouting options across 500 queries grounded against live simulated traffic sensor data; and (3) Citizen Service Query Handling — LLM resolving citizen natural language queries about municipal services, permit status, and public transport disruptions across 1,000 queries drawn from anonymized historical interaction logs. The following baselines were evaluated:

- Cloud-Only LLM (CO-LLM): Llama-2-13B-Chat hosted on a cloud GPU server with all queries and context transmitted to the cloud for inference. Represents the standard LLM-as-a-service deployment model with maximum model capacity but no latency or residency guarantee.
- Static Edge Partition (SEP): Llama-2-13B-Chat with a fixed 50/50 edge-fog layer split, without speculative decoding, AMPE dynamic partitioning, or RAG grounding. Represents naive edge deployment of a partitioned LLM without optimization.
- Edge RAG Only (ERO): Static 50/50 partition with RAG-based context retrieval but without speculative decoding or AMPE, isolating the contribution of factual grounding from inference acceleration optimizations.

- ELIS-C (Ours): The full proposed framework with speculative decoding, AMPE, knowledge graph RAG, prompt injection detection, DP-protected embeddings, and TPM 2.0 model attestation.

4.2 Computational and Communication Performance

ELIS-C achieves a median TTFT of 0.91 seconds across all three evaluation domains, compared to 1.89 seconds for CO-LLM, 1.74 seconds for SEP, and 1.31 seconds for ERO. The 52% TTFT reduction relative to CO-LLM decomposes as follows: speculative decoding contributes a 3.1x token throughput improvement equivalent to approximately 0.62 seconds saved per query at median output length; AMPE dynamic partitioning contributes a further 0.21 seconds reduction by avoiding inter-tier communication bottlenecks during bandwidth-constrained periods; and RAG context pre-retrieval overlapped with SLM drafting contributes an additional 0.15 seconds effective latency reduction through pipeline parallelism. Cloud backhaul bandwidth is reduced by 61% because only aggregate model updates and public-class query metadata transit the cloud boundary, with all raw sensor data, citizen query content, and intermediate LLM activations processed exclusively within the edge-fog residency zone.

Table 2: Performance Comparison Across LLM Inference Frameworks for Smart Cities

Method	Median TTFT (s)	TTFT Reduction (%)	Backhaul BW Red. (%)	Factual Accuracy (%)	GDPR Compliant	Mem/Node (GB)
CO-LLM	1.89	Baseline	Baseline	79.4	No	N/A
SEP	1.74	7.9	54.2	81.2	Yes	6.8
ERO	1.31	30.7	57.6	91.7	Yes	6.8
ELIS-C (Ours)	0.91	51.9	61.3	94.3	Yes	6.8

5 SCHEME COMPARISON AND DISCUSSION

Table II confirms that ELIS-C achieves the best performance across all five evaluated metrics. The most significant improvement is in factual accuracy: ELIS-C achieves 94.3% versus 79.4% for CO-LLM, which relies on the LLM parametric knowledge without real-time grounding. The 14.9 percentage point accuracy improvement quantifies the benefit of the smart city knowledge graph RAG module in providing the LLM with current city state information — traffic conditions, active incidents, transit disruptions — that is absent from the model training data and changes at timescales far shorter than LLM retraining cycles. The comparison between ERO (91.7%) and ELIS-C (94.3%) isolates the incremental accuracy gain from AMPE and speculative decoding: the 2.6 percentage point improvement reflects cases where AMPE dynamic partitioning routes complex queries to the fog server higher-memory execution environment, enabling longer context windows accommodating more retrieved knowledge graph passages.

The GDPR compliance column distinguishes ELIS-C and both edge-based baselines from CO-LLM, which cannot satisfy data residency requirements by architectural design. This binary differentiator has direct regulatory significance: GDPR Article 44 prohibitions on cross-border personal data transfer make CO-LLM legally non-deployable for citizen-facing smart city services in EU jurisdictions without explicit adequacy decisions — administrative overhead that edge-based deployment avoids entirely through structural data residency enforcement. The key advantages of ELIS-C are summarized as follows:

- Speculative decoding acceleration: City-domain SLM fine-tuning raises draft acceptance rates to 78.4%, enabling a 3.1x token throughput improvement that reduces median TTFT below one second — a qualitative threshold separating conversational from transactional interaction experiences for operators and citizens alike.
- Adaptive partitioning efficiency: AMPE dynamic layer allocation reduces inter-tier communication overhead by an average of 34% compared to static 50/50 partitioning during network bandwidth fluctuation periods, maintaining consistent TTFT performance across variable city network conditions.
- Factual currency via RAG grounding: Knowledge graph RAG retrieval with 30-second sensor freshness ensures a 14.9 percentage point accuracy advantage over cloud-hosted parametric LLMs, enabling reliable deployment for time-sensitive decision support in emergency dispatch and traffic management.
- Structural GDPR compliance: Hardware-enforced data residency boundaries and DP-protected RAG query embeddings provide regulatory compliance guarantees that policy-based cloud deployment controls cannot match, eliminating the legal risk of citizen data sovereignty violations in public smart city deployments.

5.1 Simulation Experiment Details

The simulation was implemented using Python 3.11 with Hugging Face Transformers 4.38 for LLM management, llama.cpp with CUDA acceleration for INT4-quantized Llama-2-13B edge deployment, GPTQ for model quantization, and Apache AGE 1.4 on PostgreSQL 15 for the knowledge graph. Speculative decoding was implemented using the SpecInfer framework adapted for the city-domain SLM. AMPE optimization was implemented as a Python service executing every 500ms, consuming NVIDIA DCGM telemetry for GPU resource monitoring. The RAG sentence encoder used all-MiniLM-L6-v2 fine-tuned for 3 epochs on the 4.2 billion token smart city corpus using contrastive learning with in-batch negatives. HNSW indexing used the hnswlib library with M=32 and ef_construction=200. The prompt injection classifier used DeBERTa-v3-small fine-tuned on a 47,000-sample dataset achieving 98.1% F1-score on a held-out test set. Factual accuracy was evaluated using an automated LLM-as-judge protocol with ground-truth city state snapshots from the CityBench simulator [16].

Ablation studies confirmed that each ELIS-C component contributes independently and additively to overall performance. Removing speculative decoding while retaining AMPE and RAG increased median TTFT by 0.43 seconds (47% regression). Removing AMPE while retaining speculative decoding and RAG increased TTFT by 0.21 seconds during peak-load periods when static partitioning overloaded individual edge nodes. Removing RAG while retaining speculative decoding and AMPE reduced factual accuracy to 81.6% — a 12.7 percentage point regression confirming that parametric LLM knowledge alone is insufficient for temporally grounded smart city decision support regardless of inference latency optimization.

6 CONCLUSION

This paper presented ELIS-C, an edge-based LLM inference framework for smart cities that overcomes the latency, data sovereignty, and factual currency barriers preventing cloud-hosted LLMs from deployment in production smart city management systems. Through city-domain speculative decoding achieving a 3.1x token throughput improvement, adaptive model partitioning minimizing end-to-end TTFT under dynamic resource constraints, and a continuously updated knowledge graph RAG module grounding LLM outputs in current city state with differential privacy protection, ELIS-C delivers a 52% TTFT reduction, 61% cloud backhaul bandwidth reduction, and 94.3% factual accuracy while maintaining full GDPR data residency compliance across three smart city application domains. The layered security architecture addresses the novel prompt injection and context poisoning threats introduced by LLM deployment in public-facing city services. Future research directions include multi-modal LLM deployment at the edge incorporating live camera feeds and acoustic sensor inputs for richer situational awareness in emergency management, federated fine-tuning of city-domain SLMs across multiple municipality deployments to accelerate domain adaptation, and formal verification of LLM output safety properties for high-stakes smart city applications including autonomous emergency response and infrastructure control recommendation.

STATEMENTS ON COMPLIANCE WITH ETHICAL STANDARDS

Disclosure of conflict of interest

The authors declare no conflicts of interest regarding this manuscript

REFERENCES

- [1] A. Caragliu, C. Del Bo, and P. Nijkamp, "Smart cities in Europe," J. Urban Technol., vol. 18, no. 2, pp. 65-82, Apr. 2011.
- [2] V. K. Kasula et al., "Agentic AI-driven CBDC: A privacy-preserving and regulatory-compliant digital payment framework," Dec. 2024, International Conference on Applied Machine Learning and Data Analytics, Springer Nature Switzerland, pp. 74–88. doi: 10.1007/978-3-032-18293-7_6.
- [3] International Telecommunication Union, "Smart Sustainable Cities: An Analysis of Definitions," ITU-T Focus Group on Smart Sustainable Cities, Geneva, Switzerland, 2014.
- [4] T. Brown et al., "Language models are few-shot learners," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), Online, 2020, pp. 1877-1901.
- [5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet Things J., vol. 3, no. 5, pp. 637-646, Oct. 2016.
- [6] S. Aminabadi et al., "DeepSpeed-Inference: Enabling efficient inference of transformer models at unprecedented scale," in Proc. SC, Dallas, 2022, pp. 1-15.

- [7] European Parliament, "Regulation (EU) 2016/679 of the European Parliament and of the Council (GDPR)," Official Journal of the European Union, Apr. 2016.
- [8] C. Thumma et al., "AgenticXAI: An explainable multi-agent architecture for autonomous enterprise intelligence," Mar. 2024, Global Journal of Engineering and Technology Advances (GJETA), vol. 18, no. 3.
- [9] M. Satyanarayanan, "The emergence of edge computing," Computer, vol. 50, no. 1, pp. 30-39, Jan. 2017.
- [10] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," Proc. IEEE, vol. 107, no. 8, pp. 1738-1762, Aug. 2019.
- [11] C. Chen, S. Borgeaud, G. Irving, J.-B. Lespiau, L. Sifre, and J. Jumper, "Accelerating large language model decoding with speculative sampling," arXiv preprint arXiv:2302.01318, 2023.
- [12] B. Konda et al., "Enhancing privacy and security in distributed machine learning: A meta-learning approach for personalized federated systems," 2024, SSRN preprint. doi: 10.2139/ssrn.5043297.
- [13] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), Online, 2020, pp. 9459-9474.
- [14] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," arXiv preprint arXiv:2307.09288, 2023.
- [15] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, "GPTQ: Accurate post-training quantization for generative pre-trained transformers," in Proc. ICLR, Kigali, 2023.
- [16] S. Ayyamgari et al., "Quantum computing: Challenges and future directions," Jul. 2023, International Journal of Advanced Research in Science, Communication and Technology (IJARSCT). doi: 10.48175/IJARSCT-14000V.
- [17] Y. Mao et al., "SpecInfer: Accelerating large language model serving with tree-based speculative inference and verification," in Proc. ASPLOS, La Jolla, 2024, pp. 932-949.
- [18] M. He et al., "DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing," in Proc. ICLR, Kigali, 2023.
- [19] N. Suresh et al., "EMDIF: An explainable multi-agent intelligence framework for cross-domain analytics," 2023, IJARSCT, vol. 3, no. 1, pp. 601–618. doi: 10.48175/IJARSCT-11962A.
- [20] M. Abadi et al., "Deep learning with differential privacy," in Proc. ACM CCS, Vienna, 2016, pp. 308-318.
- [21] B. Y. R. Thumma et al., "Agentic financial intelligence: A survey of LLM-powered agents for risk management, compliance, and financial analytics," 2022, Indo Asian Journal of Management and Entrepreneurship, vol. 13, no. 6, pp. 51–70.
- [22] Y. Zheng et al., "Judging LLM-as-a-judge with MT-Bench and chatbot arena," in Proc. NeurIPS, New Orleans, 2023, pp. 46595-46623.