Appendix

A Appendix Overview

This appendix provides supplementary information, analysis, and technical details to support the main body of the paper. The contents are organized as follows:

Appendix B presents a general comparison of WiLLM against existing open-source wireless communication systems and LLM deployment frameworks. This comparative analysis establishes WiLLM's distinctive contributions within the telecommunications-AI convergence landscape.

Appendix C extends the comparative analysis through a granular examination of specific technical capabilities across contemporary wireless systems. This detailed evaluation substantiates WiLLM's position as the first comprehensive system specifically designed for wireless LLM service delivery.

Appendix D illustrates the physical implementation of the WiLLM testbed, depicting the comprehensive hardware architecture that enables end-to-end LLM service delivery over wireless networks. It also showcases the WiLLM monitoring and performance analysis GUI, which provides real-time visualization and analytical capabilities for system optimization.

Appendix E details WiLLM's hierarchical API architecture and presents the Tree-Branch-Fruit Slicing algorithm for UE resource allocation. These technical specifications elucidate the system's cross-layer integration mechanisms and resource distribution methodologies.

Appendix F contains supplementary data analysis and visualizations from the WiLLM dataset. These figures provide deeper insights into the stability of experimental conditions, comparative performance between slice-enabled and normal traffic, and the complex interactions between various system parameters. More importantly, we conducted a causal analysis of the results shown in Figure. 3 to Figure 7 of the dataset.

Appendix G introduces two novel benchmarking metrics - the LLM-Aware Resource Efficiency Index (LAREI) and the LLM Slice Efficiency Quotient (LSEQ). These metrics are derived from rigorous analysis of the WiLLM dataset and provide a comprehensive framework for evaluating the performance of LLM wireless communication systems.

Appendix H presents a comprehensive tabulation of all metrics collected in the WiLLM dataset across the UE, RAN, and Edge Server layers. This exhaustive catalog enables researchers to understand the full scope and granularity of the empirical data foundation provided by WiLLM.

The appendix aims to provide a rigorous technical foundation for the innovative architectural designs and empirical findings presented in the main paper. By offering detailed comparative analysis, comprehensive system specifications, in-depth dataset insights, and novel performance evaluation frameworks, this supplementary material enables researchers to fully engage with the technical contributions of WiLLM and build upon its open-source foundation to advance the state of the art in wireless LLM service delivery.

B Appendix for Section 6 (Related Works): General Comparation of Systems

Table 1. Comparison of Open Source Wireless Communication Systems and LLM deployment Systems

System Name	Network Architecture	Slice Support	LLM Integration	Resource Scheduling	UE Compatibility	API Framework	Deployment Flexibility	Dataset & Benchmark
WiLLM	UE-gNB-CN with Edge Server	"Tree-Branch-Fruit" Slice	Designed specifically for LLM services	Dual-mode resource schedule	All UE	Cross-layer API framework	Core and edge part flexibility	✓
OAI [25]	UE-gNB-CN	Base slice	✗	Basic resource schedule	Slice-enabled UE	gNB Only	Traditional service-oriented	✗
srsRAN [11]	UE-gNB-CN	Base slice	✗	Flexible resource schedule	Slice-enabled UE	gNB Only	Traditional service-oriented	✗
Open5GS [29]	UE-gNB-CN	Base slice	✗	Basic resource schedule	Slice-enabled UE	gNB Only	Traditional service-oriented	✗
Hugging Face's TGI [17]	LLM only	✗	Fine-tuning and optimization	✗	✗	Computation Only	Server	✗
Ollama [27]	LLM only	✗	LLM deployment	✗	✗	Computation Only	Server	✗
TensorRT-LLM [26]	LLM only	✗	Lightweight reasoning	✗	✗	Computation Only	Server	✗
DeepSpeed [30]	LLM only	✗	Distributed training and reasoning	✗	✗	Computation Only	Server	✗

Legend: ✓ Support ✗ Not Considered Blue text: key comparisons

Note: This comparison contrasts systems with different foundations: OAI, srsRAN, and Open5GS are ground-up implementations, while WiLLM strategically extends OAI's architecture. While this may appear uneven, the comparison serves to highlight functional gaps in existing systems and how WiLLM addresses these through targeted extensions. This approach enables rapid innovation while leveraging established frameworks, demonstrating how focused architectural evolution can efficiently address emerging application needs.

Table 1 presents a comprehensive comparative analysis of WiLLM against existing open-source wireless communication systems and LLM deployment frameworks. This systematic evaluation establishes WiLLM's distinctive contributions within the telecommunications-AI convergence landscape. While traditional systems (OAI, srsRAN, Open5GS) provide foundational wireless infrastructure components, they exhibit significant limitations in LLM integration capabilities. Concurrently, established LLM frameworks (Hugging Face's TGI, Ollama, TensorRT-LLM, DeepSpeed) offer sophisticated model deployment but lack wireless communication integration. WiLLM uniquely bridges this interdisciplinary gap through its "Tree-Branch-Fruit" slicing architecture, cross-layer API framework, and universal UE compatibility mechanisms. This comparative evaluation demonstrates how WiLLM transcends the limitations of existing systems through its specialized design for LLM service delivery in wireless environments.

C Appendix for Section 6 (Related Works): Detailed Technical Comparison of Systems

Table 2. Technical Comparison of Wireless Systems

Technical Features	**WiLLM**	**OAI**	**srsRAN**	**Open5GS**	**FlexRIC**	**Janus**
Type	LLM-Oriented Platform	Open-Source 5G System	Open-Source 4G/5G Suite	Open-Source 5G CN	Programmable RAN SDK	Programmable RAN Platform
Main Features	Specialized for LLM Comm.	Research and Prototyping	eNodeB and UE Apps	LTE/NR Core Network	xAPP Development	O-RAN Service Model
LLM-Specific Slicing Architecture	🟩	🟥	🟥	🟥	🟥	🟥
Dynamic Slice Compatibility	🟩	🟥	🟥	🟥	🟨	🟨
Universal UE Compatibility for Slice	🟩	🟥	🟥	🟥	🟨	🟨
Multi-UE-Multi-Slice Coordination	🟩	🟥	🟥	🟥	🟨	🟨
Dual-Mode Resource Scheduling	🟩	🟥	🟥	🟥	🟨	🟨
Cross-Layer API Framework	🟩	🟥	🟥	🟥	🟨	🟨
Flexible LLM Deployment	🟩	🟥	🟥	🟥	🟥	🟥
LLM Communication Dataset	🟩	🟥	🟥	🟥	🟥	🟥
LLM Communication Benchmark	🟩	🟥	🟥	🟥	🟥	🟥
Hierarchical Slice Policy Enforcement	🟩	🟥	🟥	🟥	🟨	🟨
Application-Layer Slice Access	🟩	🟥	🟥	🟥	🟨	🟨
Synchronized Multi-Interface Metrics	🟩	🟥	🟥	🟥	🟨	🟨
Offline and Online Slice Optimization	🟩	🟥	🟥	🟥	🟨	🟨
Programmability Interface	🟩	🟥	🟥	🟥	🟩	🟩
xApp/rApp Support	🟨	🟥	🟥	🟥	🟩	🟩
Disaggregated RAN Architecture	🟨	🟨	🟨	🟥	🟩	🟩
RAN Intelligence Controller Integration	🟨	🟥	🟥	🟥	🟩	🟩
Service Model Flexibility	🟨	🟥	🟥	🟥	🟨	🟩
Multi-vendor RAN Integration	🟥	🟥	🟥	🟥	🟩	🟩
Low-latency Optimization Tools	🟩	🟨	🟨	🟥	🟩	🟨
O-RAN Compliance	🟨	🟥	🟥	🟥	🟩	🟩
Protocol Strictness	🟥	🟩	🟩	🟩	🟨	🟨
Original Protocol Stack Implementation	🟥	🟩	🟩	🟩	🟨	🟨
Original Hardware Support	🟩	🟩	🟩	🟩	🟨	🟨
Original System Architecture	🟨	🟩	🟩	🟩	🟨	🟨
Open-Source Community Activity	🟨	🟩	🟩	🟩	🟨	🟨
Standard Compliance Testing Tools	🟥	🟩	🟩	🟨	🟨	🟥
Cost Efficiency for Deployment	🟨	🟩	🟩	🟩	🟨	🟨

Legend: 🟩 Full support/Good 🟥 Not supported/Bad 🟨 Partial support/Mediocre
Note: WiLLM is an extend of OAI

Table 2 extends our comparative analysis through a granular examination of specific technical capabilities across contemporary wireless systems. This evaluation reveals significant limitations in existing platforms regarding LLM service integration. While established systems like OAI, srsRAN, and Open5GS offer partial implementation of core wireless networking features, they demonstrate substantial deficiencies in specialized LLM communication capabilities. WiLLM distinguishes itself through full implementation of LLM-specific features, including dynamic slice compatibility, multi-UE-multi-slice coordination, and hierarchical slice policy enforcement. The matrix also highlights WiLLM's innovative application-layer slice access mechanism, which enables universal device compatibility without requiring protocol-level modifications. This detailed technical comparison substantiates WiLLM's position as the first comprehensive system specifically designed for wireless LLM service delivery.

D Appendix for Section 4 (System): The Hardware Platform

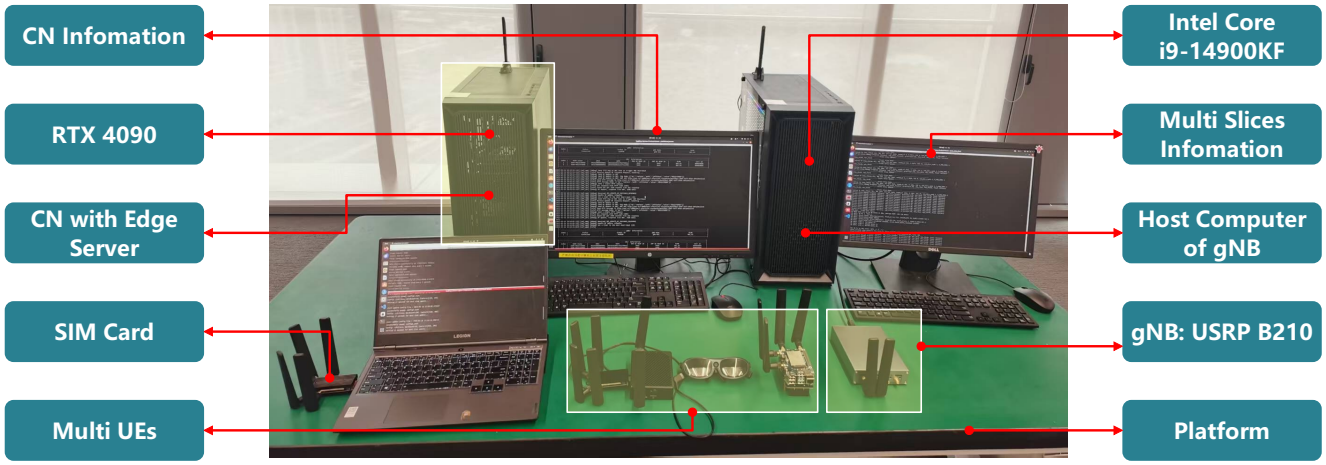

Figure 14. Hardware deployment of the WiLLM testbed, showcasing the integrated system with core network servers, edge computing nodes, Radio Access Network equipment, and multiple user terminal devices.

Figure 14 depicts the physical implementation of our WiLLM testbed, illustrating the comprehensive hardware architecture that enables end-to-end LLM service delivery over wireless networks. The deployment integrates multiple hierarchical components: an Intel Core i9-14900KF processor serving as the gNB host, a USRP B210 software-defined radio functioning as the RF frontend, and an NVIDIA RTX 4090 GPU enabling efficient LLM inference processing. The system supports simultaneous multi-slice and multi-UE operations through specialized hardware components and SIM card integration. This hardware configuration establishes a complete signal chain from core network to edge computing infrastructure, facilitating comprehensive experimental validation of the WiLLM architecture across diverse deployment scenarios and computational requirements.

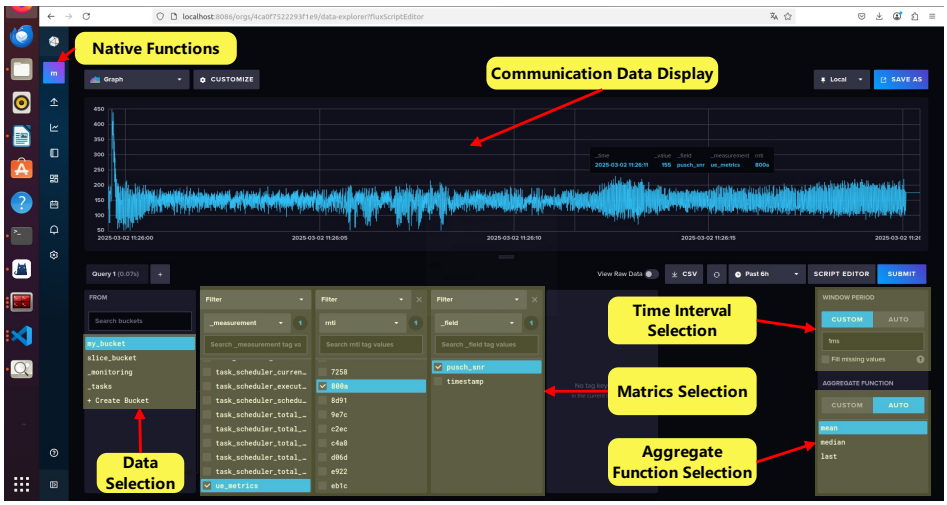

Figure 15. WiLLM monitoring and performance analysis GUI, supporting multi-dimensional metric visualization, time-series analysis, and aggregate function computation.

Figure 15 illustrates the WiLLM monitoring and performance analysis GUI, which provides real-time visualization and analytical capabilities for system optimization. The interface implements a multi-dimensional metric visualization framework supporting time-series analysis and aggregate function computation. Key functional components include: (1) communication data display for real-time performance monitoring, (2) metrics selection interface for parameter filtering and comparison, (3) time interval selection for temporal analysis, (4) aggregate function selector for statistical computation, and (5) data selection mechanisms for customizing visualization parameters. This integrated monitoring framework enables comprehensive performance evaluation across multiple system layers, facilitating both runtime optimization and longitudinal performance analysis.

E Appendix for Section 4 (System): The API Architecture and Slice Algorithm of WiLLM

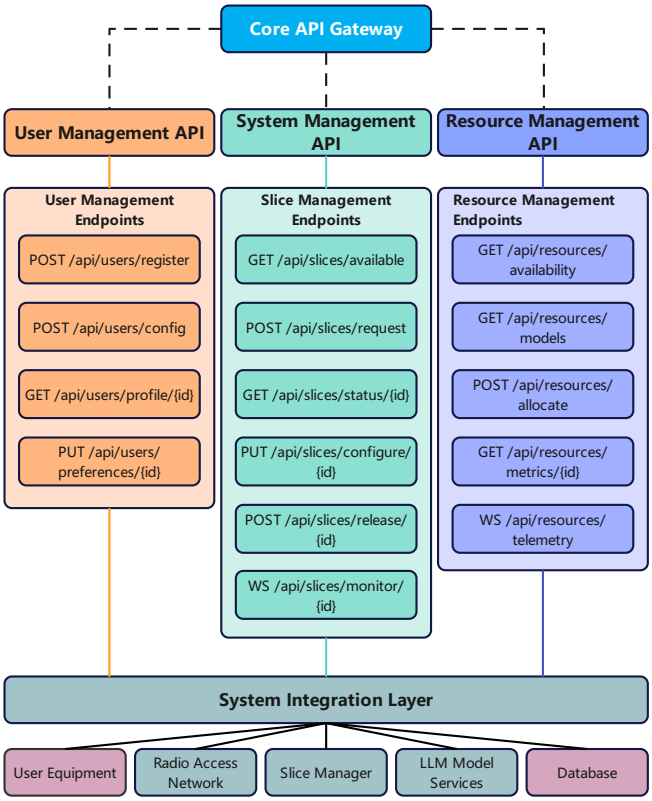

Figure 16. API Architecture of WiLLM.

Figure 16 illustrates WiLLM's hierarchical API architecture with a three-tier structure: (1) User Management API handling registration, configuration and preferences; (2) System Management API orchestrating slice operations through availability checks, request processing, and status monitoring; and (3) Resource Management API managing resource discovery, allocation and telemetry. This architecture implements cross-layer integration while maintaining separation of concerns, using RESTful APIs for synchronous operations and WebSocket connections for asynchronous monitoring.

Algorithm 1 presents the Tree-Branch-Fruit Slicing algorithm for UE resource allocation. It processes UE identity, PRBs, slice policies, channel quality indicators, and throughput metrics through four phases: (1) branch slice determination using NSSAI configuration (lines 1-4); (2) transport block size computation with proportional fair metrics (lines 5-7); (3) hierarchical policy enforcement across branches and fruits (lines 8-13); and (4) final resource allocation with MCS selection (lines 14-16). This implementation ensures efficient resource distribution while maintaining slice isolation and service differentiation across multiple UEs.

Algorithm 1 Tree–Branch–Fruit Slicing for UEs

Require: UE u, total PRBs N_{PRB}, slice policies $\mathcal{P}$, channel quality parameters Q_m, R, n_{RB}, n_{sym}, L, and historical throughput $\Theta(u)$.

Ensure: Final allocated PRB count $R(u)$ and selected MCS for u.

1: /* **Determine UE's branch slice based on NSSAI configuration** */

2: Define the slice set of u as
$$\mathcal{S}(u) = \{\, s \mid s \text{ is extracted from NSSAI}(u) \,\}$$

3: Compute the branch slice index
$$b = \text{MatchBranch}(\mathcal{S}(u), \mathcal{P})$$

4: Retrieve branch policy parameters:
$$\alpha_b^{\min} = \mathcal{P}(b).\min\_ratio \quad \& \quad \alpha_b^{\max} = \mathcal{P}(b).\max\_ratio$$

5: /* **Compute Transport Block Size using channel parameters** */
$$\text{TBS}(u) = f\big(Q_m,\, R,\, n_{\text{RB}},\, n_{\text{sym}},\, L\big)$$

6: Compute the proportional fair metric:
$$\gamma(u) = \frac{\text{TBS}(u)}{\Theta(u)}$$

7: Derive the initial allocation proportion via a function $\phi(\cdot)$:
$$r_{\text{init}} = N_{\text{PRB}} \cdot \phi\big(\gamma(u)\big)$$

8: Enforce branch policy constraints (lower and upper):
$$r_{\text{branch}} = \min\Big\{ r_{\text{init}},\; N_{\text{PRB}} \cdot \alpha_b^{\max} \Big\}$$
$$r_{\text{branch}} = \max\Big\{ r_{\text{branch}},\; N_{\text{PRB}} \cdot \alpha_b^{\min} \Big\}$$

9: **if** a fruit mapping exists for u **then**

10: Let $(\pi(u),\; r_{\min},\; r_{\max})$ be the fruit parameters for u.

11: **else**

12: Define default fruit parameters:
$$\pi(u) = 1, \quad r_{\min} = N_{\text{PRB}} \cdot \alpha_b^{\min}, \quad r_{\max} = N_{\text{PRB}} \cdot \alpha_b^{\max}$$

13: **end if**

14: Final resource allocation:
$$R(u) = \min\Big\{ \max\Big\{ \pi(u) \cdot r_{\text{branch}},\; r_{\min} \Big\},\; r_{\max} \Big\}$$

15: Determine MCS from channel conditions:
$$\text{MCS} = \text{SelectMCS}(u, Q_m, R, L)$$

16: **return** $(R(u),\, \text{MCS})$ =0

F Appendix for Section 5 (Dataset)

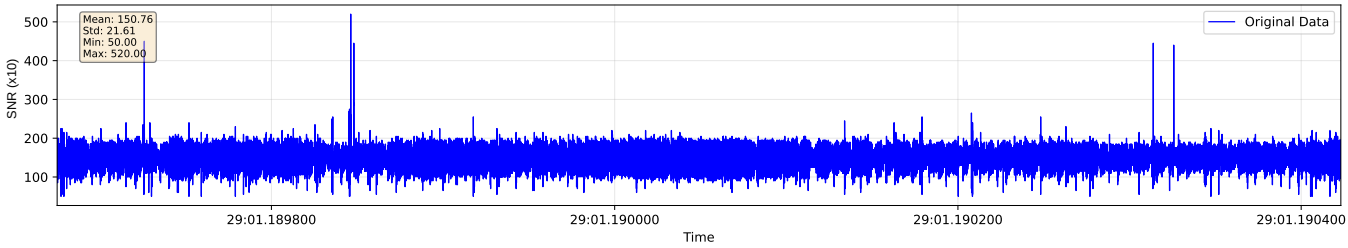

Figure 17. The variations in SNR during the data collection process.

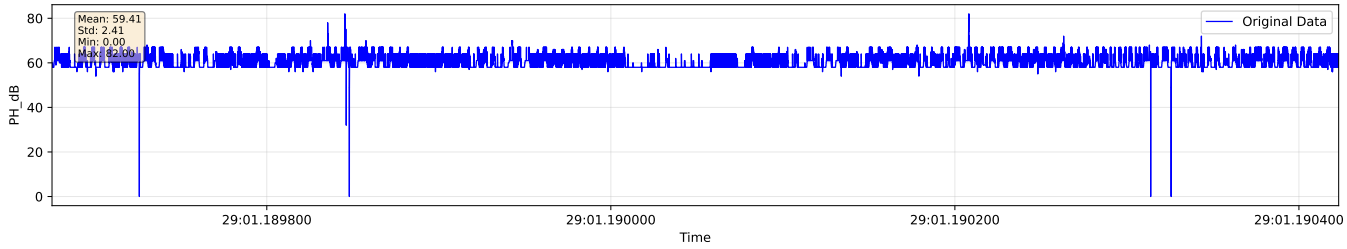

Figure 18. The variations in PH_dB during the data collection process.

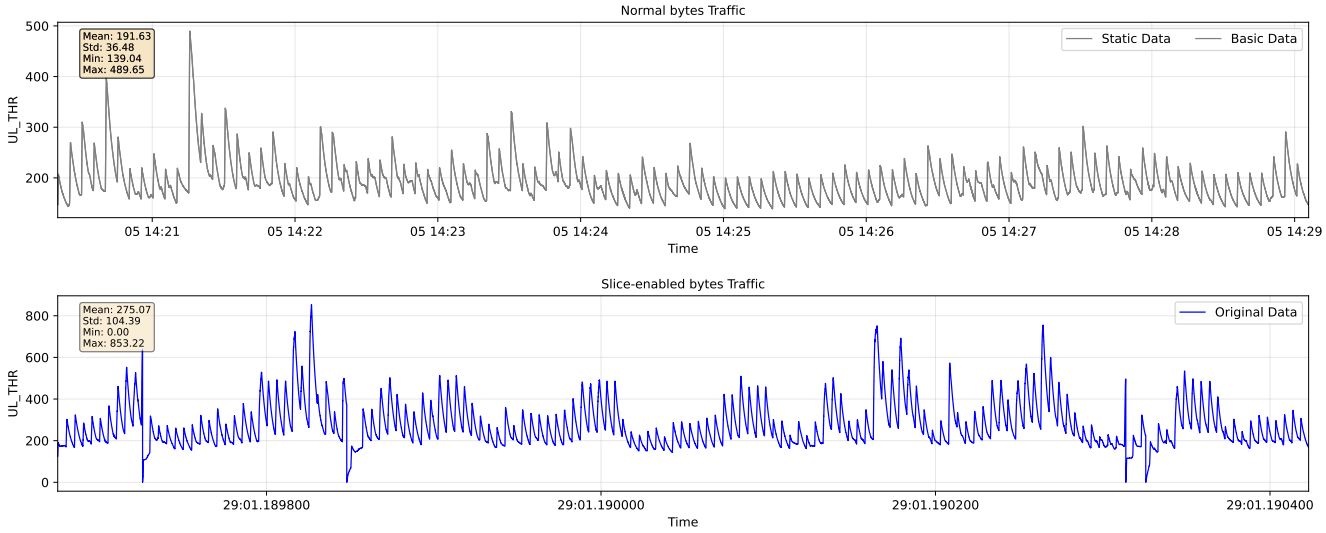

Figure 19. The variations in UL_THR during the data collection process.

Figures 17 and 18 present temporal analyses of Signal-to-Noise Ratio (SNR) and Power Headroom (PH_dB) variations during the data collection process, respectively. These visualizations demonstrate the stability of the experimental environment across multiple measurement dimensions. Figure 17 reveals consistent SNR levels (mean: 150.76, standard deviation: 21.61) across the data collection period, with minimal transient fluctuations, confirming reliable signal quality for experimental validity. Similarly, Figure 18 shows highly stable power headroom measurements (mean: 59.41, standard deviation: 2.41), indicating consistent UE transmission capabilities throughout the experimental process. These stability metrics validate the integrity of our dataset by confirming that the observed performance variations result from intentional experimental manipulations rather than environmental fluctuations or hardware inconsistencies.

Figure 19 displays comparative analyses of uplink throughput between normal traffic (top) and slice-enabled traffic (bottom) during the data collection process. The slice-enabled implementation demonstrates significantly higher average throughput compared to normal traffic, representing a 43.5% performance improvement. The slice-enabled approach also shows more

dynamic adaptation with higher peak throughput, enabling better responsiveness to varying service demands. While normal traffic exhibits regular but constrained patterns, the slice-enabled traffic shows more variable but overall superior performance characteristics with distinct periodic bursts that effectively utilize available bandwidth. This empirical evidence confirms that WiLLM's slicing architecture successfully enhances throughput efficiency, allowing more responsive performance scaling for LLM services while maintaining an appropriate baseline throughput level, which is crucial for consistent user experience in latency-sensitive LLM applications.

F.1 Dataset Collection Parameters

F.1.1 UE Configuration Parameters

During data collection, we employed a configurable UE client with the following parameter ranges:

Parameter	Description	Values
Image Capture Resolution	Resolution of images captured by smart glasses	320×240 to 640×480 pixels
Display Resolution	Resolution of the glasses display panel	800×600 to 1280×720 pixels
Request Mode	Type of request sent to LLM	"image_request", "text_request"
LLM Model	Model used for inference	"llava", "llama3.2"
Response Word Count	Maximum words requested in LLM response	50, 100, 150, 200 words
Request Frequency	Periodic request interval	5 seconds (default)

Table 3. UE configuration parameters used in dataset collection

Our experiment controller dynamically adjusted these parameters with varying coefficients (1.0, 0.9, 0.8, 0.7, 0.6, 0.5) applied to the base resolutions to analyze performance across different capture qualities. This approach simulated diverse real-world usage scenarios with varying resource constraints and quality requirements.

F.1.2 Network Slicing Configuration

The network slicing configuration used in our experiments consisted of three primary slices with different resource allocation parameters. Three fruit slices were defined with max_ratio values of $\{30\%, 60\%, 90\%\}$, all associated with the first parent slice. During data collection, our slice controller cycled through these configurations at 30-second intervals, updating the slice mapping for UEs to create varied resource allocation patterns throughout the dataset collection process.

G Benchmarking Metrics for LLM Wireless Communication

G.1 LLM-Aware Resource Efficiency Index

Based on Figure.20, our dataset analysis reveals complex interdependencies between latency, scheduled uplink bytes, and request bytes. The visualization demonstrates non-linear relationships between these critical parameters, suggesting the existence of multi-dimensional optimization boundaries that traditional isolated metrics fail to capture. The color-coded matrix illuminates several key observations:

(1) Regions of optimal performance emerge at specific combinations of resource allocation and request volume, rather than following a monotonic improvement pattern with increasing resources.
(2) Latency exhibits plateau effects at certain throughput thresholds, indicating saturation points where additional resources yield diminishing returns.
(3) Request volume influences the relationship between resource allocation and latency in non-trivial ways, creating complex performance landscapes that require holistic optimization.

It is important to note that our heatmap representation, constrained by dimensionality limitations of visualization, necessarily focuses on three representative parameters. These selected metrics—latency, scheduled uplink bytes, and request bytes—serve as proxies for more complex system characteristics. For instance, scheduled uplink bytes approximate but do not completely capture overall resource utilization patterns across the network stack, while request bytes provide insight into workload demands without fully representing request resolution or complexity variations. Nevertheless, these parameters offer meaningful approximations of the fundamental system dynamics at play.

To quantify these relationships, we propose the **LLM-Aware Resource Efficiency Index (LAREI)**, formulated as:

$$\text{LAREI} = \frac{\text{RDV} \cdot \log(1 + \text{LLM\_Para})}{\text{Resources} \cdot \text{Latency}} \cdot \omega$$

Where:

- **RDV (Request Data Volume)** quantifies the amount of data requested by the user (approximated by request bytes in our dataset)

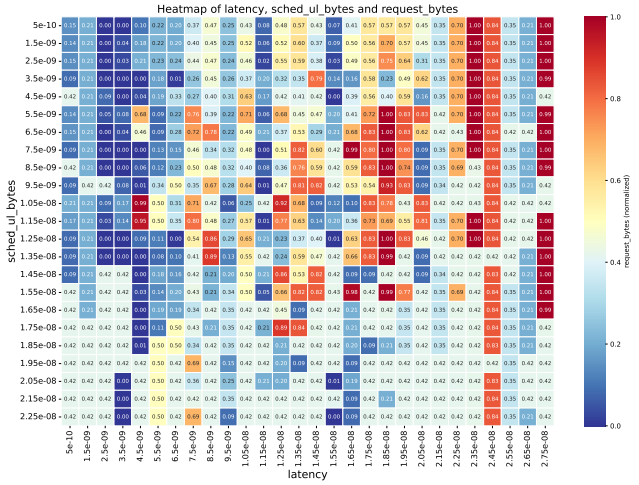

Figure 20. Latency, scheduled uplink bytes, and request bytes correlation heatmap. This figure presents a heatmap visualization of the complex interdependencies between latency, scheduled uplink bytes, and request bytes. The color-coded matrix reveals non-linear relationships between these parameters, identifying potential optimization opportunities and constraint boundaries that inform the development of comprehensive performance metrics for LLM wireless systems.

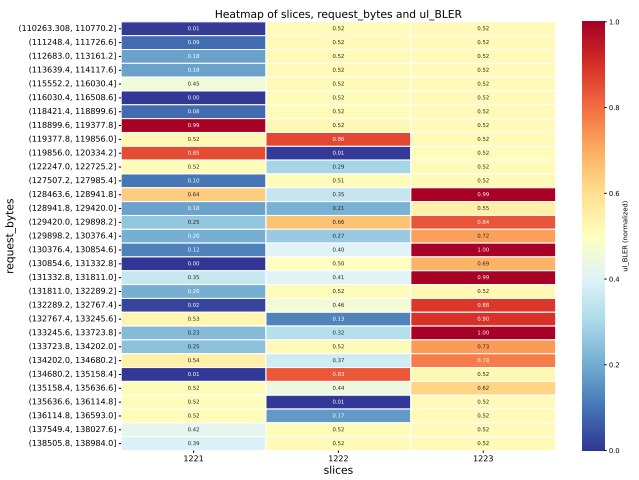

Figure 21. Correlation heatmap between slice configuration, request bytes, and uplink BLER. This figure presents a multi-dimensional analysis of how slice configuration, request payload size, and resulting error rates interact. The color-coded visualization provides empirical evidence for non-trivial relationships between these parameters, supporting the development of a slice efficiency index that balances resource allocation against transmission reliability.

- **LLM_Para (LLM Parameter Count)** represents the computational complexity of the language model (measured in billions of parameters)
- **Resources (Allocated Communication Resources)** represents the network resources provisioned for transmission (approximated by scheduled uplink bytes)
- **Latency** measures end-to-end response time
- ω is a normalization coefficient that adjusts for system-specific characteristics

This formulation captures the fundamental efficiency relationship observed in our dataset while incorporating the computational dimension of LLM services. The logarithmic scaling of the parameter count acknowledges the empirically observed sub-linear relationship between model size and perceived quality improvements. This multidimensional index effectively balances the communication-computation trade-off inherent in LLM service delivery over wireless networks.

The LAREI serves as a comprehensive metric for evaluating resource allocation algorithms specifically tailored to LLM communication services, enabling researchers to identify optimal operating configurations that balance communication efficiency with model sophistication.

G.2 LLM Slice Efficiency Quotient

Figure. 6, Figure 7 and Figure. 21 reveal critical insights into network slicing performance that challenge conventional resource allocation assumptions. It demonstrates that higher resource allocation does not necessarily translate to proportionally lower error rates, suggesting an efficiency frontier in slice optimization. The comparative visualization of BLER between slice-enabled and normal traffic illustrates that excessive resource provisioning can potentially increase transmission errors.

Figure. 21 further illuminates this phenomenon through a correlation heatmap between slice configuration, request bytes, and uplink BLER. The visualization reveals distinct patterns where certain slice configurations exhibit superior error performance despite lower resource allocation, while others show degraded performance despite abundant resources. This empirical evidence supports the development of a slice efficiency metric that transcends simple resource allocation quantities.

It should be emphasized that our heatmap analysis, while instructive, represents a dimensional reduction of a multifaceted system. The three parameters visualized—slice configuration, request bytes, and BLER—function as indicative proxies for the broader system dynamics. For example, slice configuration encapsulates numerous implementation details not fully represented in the visualization, while BLER provides insight into transmission reliability without comprehensively capturing all aspects of service quality. Despite these limitations, the observed patterns reveal fundamental relationships that inform our metric development.

Based on these observations, we propose the **LLM Slice Efficiency Quotient (LSEQ)**:

$$\text{LSEQ} = \frac{\text{RDV\_slice} \cdot (1 - \text{Error Rate}) \cdot \sqrt{\text{LLM\_Para\_slice}}}{\text{Slice Resources}} \cdot \delta$$

Where:

- **RDV_slice** represents the amount of data requested by users within the slice
- **Error Rate** quantifies transmission errors (represented by BLER in our dataset)
- **LLM_Para_slice** reflects the computational complexity of the language model deployed within the slice
- **Slice Resources** measures the communication resources provisioned to the slice
- δ is a calibration parameter that normalizes across different deployment scenarios

This formulation integrates the computational dimension of LLM services into the slice efficiency evaluation, recognizing that slices supporting more complex models (with higher parameter counts) deliver potentially higher service value. The square root scaling of parameter count reflects the diminishing returns observed in perceived quality as model size increases. The multiplicative error factor (1 − Error Rate) ensures that transmission reliability appropriately influences the efficiency evaluation, consistent with our empirical observations.

The LSEQ enables researchers to evaluate slicing algorithms specifically designed for LLM service delivery, accounting for the complex interplay between resource allocation, transmission reliability, and model sophistication. It provides a theoretically sound framework for optimizing slice configurations in LLM-oriented wireless communication systems.

H All Matrics of Dataset

Table 4. UE Layer Dataset Metrics

Layer	Metric	Description
UE	Timestamp	- Request initiation timestamp (Unix epoch milliseconds)
	Wireless Comm Time	- UE-gNB air interface duration (PDCP layer measurement)
	Total Comm Time	- UE-side end-to-end latency (request to response)
	Tx Image Resolution	- Original image dimensions (e.g., 1920×1080)
	Rx Image Resolution	- Received image dimensions post-processing
	Expected Word Count	- User-specified desired explanation length
	Actual Word Count	- LLM-generated response word count
	LLM Model	- Model architecture (LLaMA, LLaVA, etc.)
	Request Mode	- `image_request` or `text_request`
	Upload Periodicity	- Uplink interval (ms, 0=event-driven)
	Uplink Time	- UE-to-network latency (RLC layer)
	Downlink Time	- Network-to-UE latency (PDCP layer)
	Downlink Text Size	- Response payload size (UTF-8 bytes)
	Uplink Bytes	- Total bytes of the request (uplink)
	Downlink Bytes	- Total bytes of the response (downlink)

Table 5. Edge Server Layer Dataset Metrics

Layer	Metric	Description
Server	LLM Inference Time	- Model forward pass latency (ms)
	Server Processing Time	- Processing time in server
	Input Tokens	- Tokens consumed by LLM
	Output Tokens	- Tokens generated by LLM
	Cold Start Time	- Disk-to-RAM loading time
	Warm Start Time	- Cache reload time
	BLEU Score	- Text quality evaluation metric
	ROUGE Score	- Recall-based text evaluation
	Semantic Score	- Embedding similarity score
	GPU Utilization	- GPU compute usage (%)
	VRAM Usage	- Video memory consumption
	Downlink Image	- Base64 output image
	Response Text	- LLM-generated response

Table 6. RAN Layer Dataset Metrics

Layer	Metric	Description
RAN	gNB Timestamp	- gNB timestamp alignment (Unix epoch ms)
	Frame Number	- 5G NR system frame (0-1023)
	Slot Number	- Within-frame slot index (0-159)
	IMSI	- International Mobile Subscriber Identity
	RNTI	- Radio Network Temporary Identifier
	UE ID	- Device identifier within gNB
	UE Number	- Logical connection index
	DL Throughput	- Instantaneous downlink rate (Mbps)
	UL Throughput	- Instantaneous uplink rate (Mbps)
	PH (dB)	- UE Power Headroom report
	PCMAX (dBm)	- Max transmission power
	Avg RSRP	- Reference Signal Received Power
	CQI	- Channel Quality Indicator (0-15)
	RI	- MIMO Rank Indicator
	DL MCS	- Downlink Modulation and Coding Scheme
	UL MCS	- Uplink Modulation and Coding Scheme
	Scheduled UL Bytes	- Uplink buffer status
	Estimated UL Buffer	- gNB buffer estimation (bytes)
	DL PDUs Total	- Aggregated downlink PDUs
	DL BLER	- Downlink Block Error Rate (%)
	UL BLER	- Uplink Block Error Rate (%)
	DL-SCH Bytes	- Downlink shared channel payload
	DL-SCH RBs	- Downlink Resource Blocks allocated
	UL-SCH Bytes	- Uplink shared channel payload
	UL-SCH RBs	- Uplink Resource Blocks allocated
	UL MAC SDUs	- MAC Service Data Units count
	Primary Slice Max	- Main slice maximum resources
	Primary Slice Min	- Main slice minimum guarantee
	Secondary Slice Max	- Secondary slice maximum
	Secondary Slice Min	- Secondary slice minimum