Article:
Liu, Z., Liu, B., Valcarce, A. et al. (2026) LLM-based emulation of the radio resource control layer: Towards AI-native RAN protocols. IEEE Journal on Selected Areas in Communications, 44. pp. 4319-4332. ISSN: 0733-8716

https://doi.org/10.1109/jsac.2026.3673712

LLM-Based Emulation of the Radio Resource Control Layer: Towards AI-Native RAN Protocols

Ziming Liu, *Member, IEEE*, Bryan Liu, *Member, IEEE*, Alvaro Valcarce, *Senior Member, IEEE*, and Xiaoli Chu, *Senior Member, IEEE*

***Abstract*—Integrating Large AI Models (LAMs) into 6G mobile networks is a key enabler of the AI-Native Air Interface (AI-AI), where protocol intelligence must scale beyond handcrafted logic. This paper presents, to our knowledge, the first standards-compliant emulation of the Radio Resource Control (RRC) layer using a decoder-only LAM (LLAMA-class) fine-tuned with Low-Rank Adaptation (LoRA) on a multi-vendor corpus of real-world traces spanning both 5G and 4G systems. We treat RRC as a domain-specific language and construct a segmentation-safe Question-and-Answer (QA) dataset that preserves Abstract Syntax Notation (ASN.1) structure through linearization prior to Byte Pair Encoding (BPE) tokenization. The proposed approach combines parameter-efficient adaptation with schema-bounded prompting to ensure syntactic and procedural fidelity. Evaluation introduces a standards-aware triad—ASN.1 conformance, field-level coverage analysis, and uplink-to-downlink state-machine checks—alongside semantic similarity and latency profiling across 120 configurations. On 30k 5G request–response pairs plus an additional 4.8k QA turns from 4G sessions, our 8B model achieves a median cosine similarity of 0.97, a 61% relative gain over a zero-shot baseline, while sustaining high conformance rates. These results demonstrate that LAMs, when augmented with protocol-aware reasoning, can directly orchestrate control-plane procedures, laying the foundation for the future Artificial Intelligence (AI)-native Radio Access Network (RAN).**

***Index Terms*—6G, Radio Resource Control, protocol learning, AI-Native Air Interface, Large AI Model**

I. Introduction

Large AI Models (LAMs) are poised to transform wireless networks by enabling autonomous, cognitive capabilities that go beyond the scope of traditional optimization methods [1]. The forthcoming sixth generation (6G) of mobile networks is expected to incorporate Artificial Intelligence (AI) methods not merely as auxiliary optimisation tools but also as integral design primitives of the AI-Native Air Interface (AI-AI). Recent proposals for an AI-AI posit that Machine Learning (ML) models could co-design Physical Layer (PHY) and Medium Access Control (MAC) procedures [2], thereby enabling radio stacks that adapt automatically to dynamic service requirements, resource constraints, and propagation conditions [3]. The rapid evolution of LAMs [4], [5] strengthens this vision—when endowed with multi-modal perception and reasoning, such models become plausible agents for protocol-centric tasks that demand both linguistic competence and structured decision making. This raises a timely research challenge: determining the extent to which LAMs can directly engage in the generation and interpretation of control signaling in the Radio Access Network (RAN).

This paper studies whether a decoder-only LAM can be fine-tuned to parse and generate 3GPP-compliant Radio Resource Control (RRC) messages [6]. Our contributions are threefold: (i) we curate multi-operator 4G/5G RRC trace corpora and formulate an ASN.1-preserving Question-and-Answer (QA) representation; (ii) we adapt the model via supervised fine-tuning enhanced with Low-Rank Adaptation (LoRA) [7] to emulate stateful RRC procedures; and (iii) we quantify fidelity using standards-aware metrics—ASN.1 syntax validation [8], schema-level field coverage, and UL→DL state-machine conformance—alongside semantic similarity and latency. We provide that LAM-scale pretraining supplies compositional priors that generalize across 3GPP releases and vendor variants enabling multi-procedure reasoning even under incomplete context.

The 5G RRC layer orchestrates connection establishment, mobility and configuration between User Equipment (UE) and the Next-Generation Node B (gNB). Conceptually, RRC messages form a specialized, domain-specific language; learning this language is analogous to learning syntax and semantics in natural-language processing. Treating RRC messages as such positions LAM-based sequence models as natural emulators of the RRC layer. Demonstrating accurate RRC emulation would constitute an early, concrete instantiation of AI-AI principles, with direct implications for the design of the 6G control plane.

Can the protocol literacy of a LAM—specifically, its ability to parse and generate 3GPP-compliant RRC messages—be systematically improved, and through which fine-tuning strategies? This paper answers this question through a novel two-phase methodology that (i) imparts RRC knowledge to a decoder-only LAM via supervised fine-tuning enhanced with LoRA, and (ii) evaluates the resulting model's fidelity in emulating RRC procedures under realistic network scenarios. The findings offer evidence about the feasibility of LAM-based

Z. Liu and X. Chu are with the School of Electrical and Electronic Engineering, The University of Sheffield, Sheffield S10 2TN, U.K. (e-mail: {ziming.liu,x.chu}@sheffield.ac.uk).

B. Liu and Á. Valcarce are with Nokia Bell Labs, 12 Rue Jean Bart, 91300 Massy, France (e-mail: bryan.liu@unswalumni.com, alvaro.valcarce_rial@nokia-bell-labs.com).

This work was supported in part by the UK EPSRC grant EP/X038971/1 and the Horizon Europe Research and Innovation Program under grant 101086219.

The authors acknowledge the use of resources provided by the Isambard-AI National AI Research Resource (AIRR). Isambard-AI is operated by the University of Bristol and is funded by the UK Government's Department for Science, Innovation and Technology (DSIT) via UK Research and Innovation; and the Science and Technology Facilities Council [ST/AIRR/I-A-I/1023]. McIntosh-Smith, S., Alam S. R. and Woods, C. (2024). "Isambard-AI: a leadership class supercomputer optimised specifically for Artificial Intelligence". https://doi.org/10.48550/arXiv.2410.11199

standards-compliant RRC emulation.

Without loss of generality, this work focuses on *decoder-only* transformer architectures (e.g., Llama-class models [9]). Autoregressive decoding matches the turn-taking structure of RRC exchanges and permits key–value caching, which curtails inference latency and memory footprint, both critical for deployment at the gNB. Decoder-only models thereby strike a favourable balance between capacity and computational efficiency, especially when combined with parameter-efficient fine-tuning techniques such as LoRA [7]. LoRA freezes the original model weights and introduces low-rank update matrices, reducing trainable parameters by orders of magnitude while maintaining expressiveness; the approach mitigates both the cost of domain adaptation and the runtime overhead of inference, two primary barriers to operational use.

Training LAMs for specialist telecom domains demands curated datasets that capture the combinatorial breadth of protocol states, while deployment must meet stringent latency budgets to avoid impairing control-plane responsiveness. To address these issues, we rebuilt two data corpuses of RRC traces (4G and 5G) with multi-operator coverage. Then, we adopt a standards-aware evaluation suite that combines ASN.1 [8] syntax validation, schema-level field coverage, and UL→DL state-machine conformance, reporting these jointly with semantic similarity and latency across 120 configurations. Ultimately, we also outline a forward-looking engineering plan toward sub-100 ms inference latency via quantization [10], constrained decoding, KV caching and multi-token draft-verify decoding.

II. System Model

This section delineates the architectural and operational principles underlying the proposed LAM-integrated gNB. We first describe the system architecture, detailing the functional decomposition of the disaggregated gNB and the embedding of a LAM within the RRC layer, while ensuring adherence to standardized interface definitions. Then, the model-driven control-plane workflow is examined, encompassing the reception and contextual interpretation of uplink RRC messages and the autoregressive synthesis of standards-compliant downlink responses.

A. Architecture and Functional Integration

The proposed Large RRC Model integrates a decoder-only transformer-based LAM into the RRC layer of a disaggregated gNB, as depicted in Fig. 1. Following the 3rd Generation Partnership Project (3GPP) NG-RAN standard, the gNB is partitioned into a Central Unit (CU) and a Distributed Unit (DU), with the CU further subdivided into separate Control Plane of Central Unit (CU-CP) and User Plane of Central Unit (CU-UP) entities. The CU-CP hosts the RRC layer, responsible for essential control-plane tasks such as connection establishment, radio bearer management, and mobility decisions. Replacing the traditional rule-based logic, the embedded RRC LLM serves as an intelligent decision engine within the RRC, producing control messages and associated

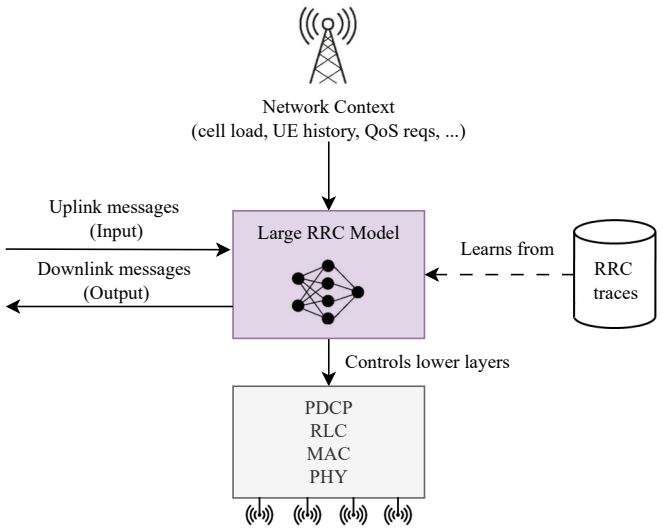

Fig. 1. High-level concept of an AI-native gNB-side RRC layer powered by a LLM, illustrating its core inputs, outputs, learning source, and interaction with the protocol stack.

protocol decisions via autoregressive inference. Conversely, the CU-UP encompasses user-plane layers, primarily Service Data Adaptation Protocol (SDAP) and user-plane Packet Data Convergence Protocol (PDCP), that manage the transport of user data. The DU, hosting lower-layer (e.g., Radio Link Control (RLC), MAC, and PHY layers) protocols, interfaces directly with the Remote Radio Heads (RRHs) performing Radio Frequency (RF) and low-level physical-layer tasks. Fig. 2 illustrates all these interfaces and functions. Integrating an LLM at the RRC level is intended to preserve the external protocol behavior, leaving northbound and lateral interfaces to peer nodes unaltered, thus ensuring interoperability with existing network deployments.

As in standard 5G New Radio (NR), uplink RRC messages from UEs, such as Connection Requests or Measurement Reports, arrive at the CU-CP via Control-Plane Interface between CU-CP and DU (F1-C), where they are processed by the embedded RRC LLM, replacing the conventional RRC layer implementation. The model interprets these inputs contextually, considering ongoing connection states, historical interactions, and dynamic network conditions to generate suitable downlink RRC responses (e.g., Connection Setup or Mobility Reconfiguration). The generated messages maintain compliance with standardized 3GPP encoding, thus transparently replacing the message generation logic. Downlink responses are then transmitted back to the UE through the DU via the same interface.

B. LLM-Driven Signaling Workflow and Model Operation

Interactions between the RRC layer and the user-plane modules are similarly orchestrated. When the Large RRC Model decides to initiate or modify data radio bearers—potentially based on inferred context or changing service demands—it signals these decisions to the CU-UP over the Interface between CU-CP and CU-UP (E1). The CU-UP subsequently configures

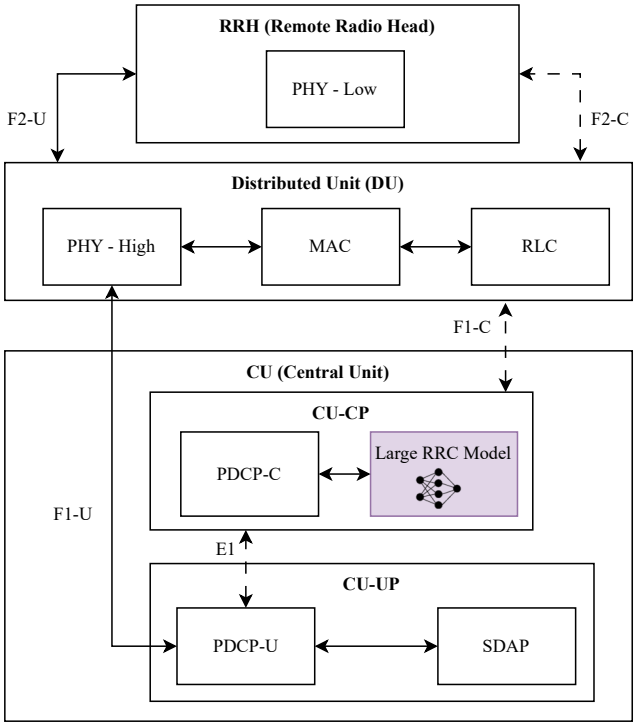

Fig. 2. Low-level architecture of an NR gNB disaggregating an LLM-based RRC layer.

the corresponding SDAP and PDCP entities, enabling adaptive management of the user-plane traffic. Meanwhile, data-plane traffic flows remain unmodified and proceed independently over the established user-plane interfaces (User-Plane Interface between CU-UP and DU (F1-U), User-Plane Interface between RRH and DU (F2-U)).

Fig. 3 illustrates the fine-tuning and inference workflow specific to the proposed RRC LLM. The decoder-only transformer is taught RRC domain knowledge using supervised fine-tuning on a curated dataset of historical RRC message traces collected from a real-world 5G network. This specialized fine-tuning imparts explicit protocol knowledge to the model, enabling it to emulate standard RRC behaviors accurately during live inference. Operationally, the trained model functions continuously within the CU-CP, dynamically generating control-plane messages as new UE requests are received.

III. Fine-tuning of LLM for RRC Message Processing

This section describes the pipeline for adapting a decoder-only LLM to emulate RRC procedures. Our approach consists of three components: (i) dataset construction, where raw traces are reorganized into segmentation-safe uplink–downlink QA pairs; (ii) parameter-efficient supervised fine-tuning using LoRA; and (iii) latency-aware evaluation to assess feasibility under control-plane timing constraints. Earlier traces often exceeded context limits, requiring ad-hoc segmentation that broke procedural continuity. Our design eliminates this by enforcing per-procedure boundaries.

A. Datasets: Construction, Scope, and Coverage

We use two complementary corpora with a common, segmentation-safe construction pipeline:

- **NR dataset (proprietary, single-operator test setups):** ∼30,247 UL↔DL RRC pairs extracted from 2,524 sessions collected in controlled test environments.
- **LTE dataset (proprietary road tests across two networks):** ∼4,808 QA turns across 1,601 sessions collected via real-world drive tests.

Segmentation-safe UL→DL QA construction: Raw RRC traces are sessionized and reorganized into UL–DL QA pairs preserving procedural causality and ASN.1 structure (cf. Figure 4). Each session is capped in turns to respect context limits, with adjacent UL messages concatenated when appropriate (e.g., `rrcReconfigurationComplete + measurementReport`). The resulting per-turn inputs (X) and targets (Y) follow the mapping in section III-B and Eq. 1–4.

Schema-bounded prompting and radio constraints: To enforce structural validity, we inject a session-specific, size-bounded ASN.1 micro-schema (see Fig. A.1 in the appendix) into the system prompt together with radio constraints (e.g., allowed EARFCNs). This preserves field names/order and prevents extraneous elements without token-level grammar overhead. This micro-schema mechanism was applied only to the LTE corpora.

Attribution and availability: Both corpora are *private and proprietary*. Public LTE datasets/simulators exist (e.g., [11], [12]), but are synthetic and were *not used* in this work.

Coverage: Table I summarizes the observed message-type counts across the two corpora. Major RRC procedures (connection establishment, initial security activation, reconfiguration, release, capability transfer) are well represented in both RATs.

B. Supervised Fine-Tuning Framework

We model RRC request–response exchanges as a conditional generation task over uplink–downlink pairs. Let X denote the uplink message sequence and Y the corresponding downlink response. The objective is to estimate the conditional probability:

$$\pi_\theta(Y \mid X) = \prod_{t=1}^{T} P(y_t \mid y_{<t}, X; \theta), \tag{1}$$

where θ are trainable parameters, y_t is the token generated at step t, and T is the sequence length. This formulation aligns with the QA-style mapping introduced in Sec. III-A. X and Y are tokenized using the same Byte Pair Encoding (BPE) tokenizer as the Llama backbone. Therefore, T denotes the number of BPE tokens in the response sequence.

To adapt a decoder-only LLM efficiently, we apply LoRA, which introduces low-rank updates to frozen backbone weights. For a weight matrix $W_0 \in \mathbb{R}^{d\times k}$, the update is:

$$W_\theta = W_0 + AB, \tag{2}$$

where $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{r\times k}$ are trainable factors with rank $r \ll \min\{d, k\}$. This reduces trainable parameters by orders

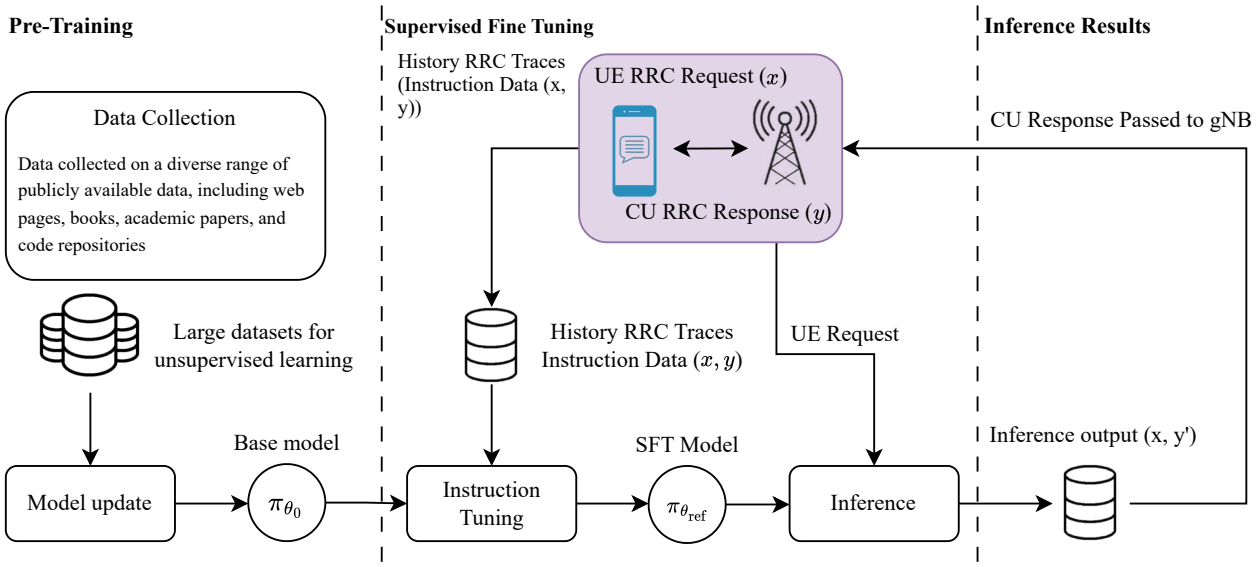

Fig. 3. Data flow for fine-tuning and inference in the LLM-based RRC system.

TABLE I
BREAKDOWN OF LTE AND NR RRC MESSAGE TYPES OBSERVED IN THE PROPRIETARY FIELD LOGS. A — DENOTES NOT OBSERVED FOR THAT RAT.

Message	LTE Count	NR Count
dlInformationTransfer	149	1 743
measurementReport	229	43 410
mobilityFromNRCommand	—	92
paging	1 016	—
rrcConnectionReconfiguration	701	—
rrcConnectionReconfigurationComplete	891	—
rrcConnectionRelease	110	—
rrcConnectionRequest	662	—
rrcConnectionReestablishment	3	—
rrcConnectionReestablishmentRequest	5	—
rrcConnectionSetup	775	—
rrcConnectionSetupComplete	879	—
rrcReconfiguration	—	16 034
rrcReconfigurationComplete	—	16 099
rrcRelease	—	4 267
rrcReestablishment	—	28
rrcReestablishmentComplete	—	28
rrcReestablishmentRequest	—	119
rrcSetup	—	4 072
rrcSetupComplete	—	4 068
rrcSetupRequest	—	4 009
securityModeCommand	653	3 827
securityModeComplete	693	3 824
systemInformation	95	—
systemInformationBlockType1	265	—
ueCapabilityEnquiry	633	876
ueCapabilityInformation	6	873
ueInformationRequest-r9	3	—
ueInformationResponse-r9	134	—
ulInformationTransfer	55	1 924
Total	**7 957**	**105 293**

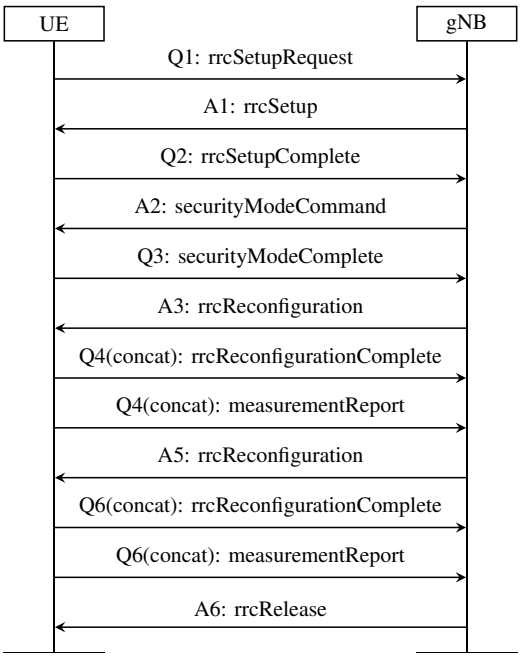

Fig. 4. Example of a pre-processed RRC trace segment. Messages are structured into Question (Q) / Answer (A) pairs for LLM training, where 'Q' denotes the input RRC message(s) (X) and 'A' the target response (Y). Note the consolidation of multiple physical messages (e.g., *rrcReconfigurationComplete* and *measurementReport*) into a single logical request, labeled here as Q4(concat) and Q6(concat).

of magnitude while preserving expressiveness. The fine-tuning objective minimizes the negative log-likelihood:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log \pi_\theta(Y^{(i)} \mid X^{(i)}). \tag{3}$$

Here, N denotes the total number of aligned uplink–downlink pairs in the supervised training set. Inference uses greedy decoding under the fine-tuned model:

$$\hat{Y} = \arg\max_{Y} \pi_\theta(Y \mid X). \tag{4}$$

Algorithm 1 RRC-LLM Training and Inference Pipeline

Require: Raw RRC traces $\mathcal{T}_{\text{raw}}$, pre-trained model $\pi_{\theta_{\text{init}}}$
Ensure: Fine-tuned model π_{θ^*}

Phase 1: Dataset Construction
$\mathcal{D} \leftarrow \texttt{Preprocess}(\mathcal{T}_{\text{raw}})$ ▷ Filter, align, and extract (X, Y) pairs
for all $(X^{(i)}, Y^{(i)}) \in \mathcal{D}$ **do**
  $X^{(i)} \leftarrow \texttt{BPE\_Tokenize}(X^{(i)})$
  $Y^{(i)} \leftarrow \texttt{BPE\_Tokenize}(Y^{(i)})$
end for
Phase 2: Supervised Fine-Tuning (SFT)
$\pi_{\theta_{\text{ref}}} \leftarrow \texttt{LoRA\_Init}(\pi_{\theta_{\text{init}}})$
for $e = 1$ to E_{SFT} **do**
  for all $(X^{(i)}, Y^{(i)}) \in \mathcal{D}$ **do**
    $\mathcal{L}_{\text{SFT}} \leftarrow -\log \pi_{\theta_{\text{ref}}}(Y^{(i)} \mid X^{(i)})$
    Update θ_{ref} using Adam on $\mathcal{L}_{\text{SFT}}$
  end for
end for
Phase 3: Inference
Given new uplink message X, compute
  $\hat{Y} \leftarrow \arg\max_Y \pi_\theta(Y \mid X)$ **return** $\hat{Y}$

C. Training Configuration and Convergence Behavior

Supervised fine-tuning of the Llama-3 8B backbone was performed under the configuration in Table II. The loss decreased by two orders of magnitude within the first 10^3 steps and stabilized after ∼8.4k steps, as shown in Fig. 5. The EMA-smoothed trace flattens near 10^{-2}, indicating convergence. We select the checkpoint at this point to capture $> 95\%$ of attainable loss reduction while avoiding overfitting and unnecessary GPU hours. Fig. 6 further compares training and validation loss across backbones and LoRA ranks, highlighting stability trends and the effect of parameter-efficient tuning.

This trajectory exhibits three regimes: (i) rapid descent during warm-up (0–10^3 steps), (ii) gradual improvement (10^3–8.4×10^3), and (iii) saturation beyond 8.4×10^3. The EMA curve mitigates variance from the small effective batch size (2×4096 tokens). Validation on a disjoint trace suite confirmed generalization without requiring full validation passes during training. Fig. 5 illustrate the convergence profile. The selected checkpoint (red dashed line) balances convergence speed and stability, ensuring fidelity to RRC message structures while minimizing compute overhead.

We note that earlier iterations of this training pipeline experienced frequent interruptions (e.g., SSD stalls, NCCL watchdog timeouts). These were identified as infrastructure-specific bottlenecks (insufficient storage I/O and driver-level software bugs) rather than model instability. Following the migration to the updated cluster (Table XI), the training process proved robust, achieving convergence without any hardware-induced interruptions.

D. Decoding regimes

We evaluate three decoding strategies that govern prompt structure and output constraints:

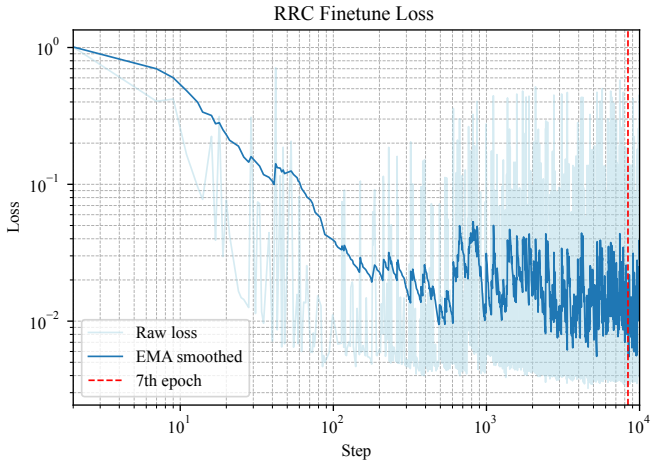

Fig. 5. Training loss trajectory on the **NR dataset** (proprietary corpus). The pale blue trace is the raw loss, and the dark blue curve is its exponential moving average (EMA). The vertical red dashed line marks the epoch selected for downstream evaluation.

TABLE II
SUPERVISED FINE-TUNING HYPERPARAMETERS

Parameter	NR Dataset	LTE Dataset
Model	Llama-3 8B	Llama-3.x (1B–8B)
Max Sequence Length	4096	4096
Precision	bf16	bf16
Optimizer	AdamW	AdamW
Learning Rate	2×10^{-5}	2×10^{-5} (full FT) 3×10^{-4} (LoRA)
Loss Function	CrossEntropy	CrossEntropy
LR schedule	–	Cosine decay + 100-step warmup
LoRA Config	–	$\alpha = 2r$, dropout=0

- **NoSys**: A generic system prompt with minimal guidance, allowing unconstrained generation. This regime prioritizes raw language-model priors and serves as a lower-bound baseline.
- **RRC**: A domain-oriented prompt that specifies the RRC context and message type but does not enforce strict schema constraints. This improves semantic alignment but leaves structural validity largely unchecked.
- **RRC_constrain**: A schema-bounded prompt that injects a size-limited Abstract Syntax Notation (ASN.1) micro-schema and radio constraints (e.g., allowed E-UTRA Absolute Radio Frequency Channel Number (EARFCN)) into the system message. Generation is restricted to fields declared in this schema, preserving field order and prohibiting extraneous elements. This regime is designed to maximize standards compliance without incurring token-level grammar overhead.

Unless otherwise stated, all reported results in Sec. IV use `RRC_constrain` for tuned configurations, as it delivers near-ceiling ASN.1/State-Machine Conformance (SMC) pass rates while maintaining practical latency. Crucially, this schema-bounded approach mitigates the structural violations observed in unconstrained baselines (e.g., the duplication of fields). By explicitly injecting the session-specific ASN.1

TABLE III
LTE CORPUS: VALIDATION LOSS AT BEST CHECKPOINT (LOWER IS BETTER)

Backbone	Full FT	LoRA r4	LoRA r8	LoRA r16
Llama-3 8B	0.02017	0.02038	0.02062	**0.01984**
Llama-3.1 8B	0.02836	0.02131	0.02152	**0.02096**
Llama-3.2 3B	**0.01915**	0.02125	0.02232	0.02207
Llama-3.2 1B	**0.02017**	0.02130	0.02106	0.02105

definitions into the context, we leverage the model's in-context learning capabilities to ground the generation process. This ensures that the model attends strictly to the fields declared in the active micro-schema, effectively preventing the hallucination of extraneous or repetitive elements without requiring computationally expensive token-level grammar constraints.

E. LTE Corpus: Multi-Backbone and LoRA Rank Ablations

To evaluate cross-RAT robustness, we repeated the fine-tuning pipeline on an LTE-specific corpus (1,601 sessions, 4,808 QA turns) using the same segmentation-safe design as in Sec. III-A. This LTE dataset captures connection setup, security activation, and reconfiguration procedures under diverse mobility scenarios.

We benchmarked four Llama-3 backbones (8B, 8B-3.1, 3B, 1B) under full fine-tuning and LoRA with ranks $r \in \{4, 8, 16\}$. All runs used bf16 precision, AdamW optimizer, and token-level cross-entropy loss. LoRA adapters were applied to attention projections and MLP layers with scaling $\alpha = 2r$ and dropout $= 0$. Learning rates followed standard practice: 2×10^{-5} for full FT and 3×10^{-4} (cosine decay, 100-step warmup) for LoRA. 5

Table III summarizes the best validation loss per configuration. On 8B backbones, higher-rank LoRA ($r = 16$) matches or slightly surpasses full fine-tuning, while on 3B/1B models, updating all weights remains advantageous. Across all settings, the gap between the best two methods is small ($\Delta \approx 6\times10^{-4}$), confirming that parameter-efficient adaptation can approach full FT performance on large models.

These results highlight two trends: (i) larger backbones benefit from higher LoRA ranks, reducing the need for full FT; (ii) smaller models favor full updates for best accuracy. This informs the latency–fidelity trade-offs explored in Sec. IV.

IV. PERFORMANCE AND EVALUATION

A. Evaluation metrics and protocol conformance

We evaluate the generated Downlink (DL) RRC messages along three independent and complementary axes: (i) semantic proximity to the reference using Sentence BERT (SBERT)-based [13] cosine similarity, (ii) ASN.1 syntax validity via encode–decode round-trip under a session-specific micro-schema, and (iii) Uplink (UL) $\rightarrow$ DL SMC. These metrics are reported jointly with latency in Tables IV–V.

a) Semantic similarity (SBERT cosine): For each test sample i, let Y_i be the ground-truth DL message and $\hat{Y}_i$ the model output given the UL context X_i. We encode both with a frozen SBERT encoder $f_{\text{sbert}}(\cdot)$ and mean-pool to sentence embeddings u_i and v_i:

$$u_i = \frac{1}{|Y_i|}\sum_t f_{\text{sbert}}(Y_i)_t, \qquad v_i = \frac{1}{|\hat{Y}_i|}\sum_t f_{\text{sbert}}(\hat{Y}_i)_t. \quad (5)$$

The cosine similarity is $s_i = \frac{u_i^\top v_i}{\|u_i\|\,\|v_i\|} \in [-1, 1]$. We report s_i directly. When combining semantic and structural metrics, a $[0, 1]$ range is preferred, so we use $\tilde{s}_i = \frac{s_i+1}{2}$. Fig. 7 illustrates the distribution of SBERT-based similarity scores for the **5G NR field corpus** under the **LLaMA-3 8B backbone** after fine-tuning with LoRA.

b) ASN.1 round-trip validity and schema recall: For each session i, we construct a trimmed ASN.1 micro-schema $\mathcal{S}^{(i)}$ (*messages and IEs relevant to that session*) and perform a strict encode–decode round-trip on $\hat{Y}_i$. The binary pass indicator is

$$\text{asn}_i \;=\; \mathbb{I}\Big[\text{Dec}_{\mathcal{S}^{(i)}}\big(\text{Enc}_{\mathcal{S}^{(i)}}(\hat{Y}_i)\big) \text{ succeeds} \wedge \text{plausible}\Big]. \quad (6)$$

To measure field-level faithfulness beyond mere text similarity, we compare the set of normalized ASN.1 Information Element (IE) names present in the reference and the hypothesis. Let $W(\cdot)$ extract this set; for non-empty $W(Y_i)$, the per-sample schema recall is

$$r_i \;=\; \frac{\big|\,W(Y_i) \cap W(\hat{Y}_i)\,\big|}{\big|\,W(Y_i)\,\big|} \in [0, 1]. \quad (7)$$

c) UL$\rightarrow$DL state-machine conformance (SMC): From the UL input we derive the set U_i of UL message types; from $\hat{Y}_i$ we extract the DL type d_i and, when present, the RRC transaction identifier $\text{TxId}(\hat{Y}_i)$. Let $\mathcal{A}(u)$ denote the set of allowed DL types given a UL type u (*strong edges*); for a set U, write $\mathcal{A}(U) = \bigcup_{u\in U}\mathcal{A}(u)$. The strict binary SMC pass is then

$$\text{smc}_i = \mathbb{I}\Big[d_i \in \mathcal{A}(U_i) \;\wedge\; \text{TxId}(\hat{Y}_i) = \text{TxId}(Y_i)\Big]. \quad (8)$$

Here, the transaction-ID equality is enforced only when both sides carry an ID. We additionally report a *weak* profile in which broadcast/weak edges are admitted (not shown for brevity when identical).

d) Reporting: In Tables IV–V, we summarize results by the distribution (median and dispersion) of s_i (or $\tilde{s}_i$), the dataset-level ASN.1 pass rate $\frac{1}{N}\sum_i \text{asn}_i$, the schema-recall distribution of r_i, and the SMC pass rate $\frac{1}{N}\sum_i \text{smc}_i$. These metrics are presented side-by-side with latency to expose fidelity–validity–latency trade-offs without conflating them into a single score.

B. Inference results: fidelity, validity, and latency

This subsection reports empirical outcomes for semantic fidelity (SBERT cosine), ASN.1 validity, SMC pass rate, and latency across the evaluated configurations. We focus on the curated highlights (Table IV) and global medians over all 120 settings (Table V).

TABLE IV
CURATED RESULTS ON THE LTE DATASET.

Backbone	Quant	Strategy	Tuning	Latency (ms)		Schema		Similarity		Pass Rate	
				Med.	Avg.	Avg	Med	Avg	Med	ASN	SMC
L-3 8B	FP16	RRC_constrain	lora-r16	3544	5258.0	0.976	1.000	0.992	1.000	0.995	0.993
L-3 8B	Q4_K_M	RRC_constrain	lora-r16	2556	3854.1	0.969	1.000	0.990	1.000	0.989	0.988
L-3 8B	FP16	RRC	lora-r16	3418	4280.4	0.467	0.333	0.723	0.669	0.023	0.000
L-3 8B	Q4_K_M	RRC	lora-r16	2519	2529.7	0.439	0.333	0.726	0.666	0.003	0.002
L-3 8B	FP16	NoSys	lora-r16	1987	2011.7	0.231	0.241	0.480	0.478	0.003	0.002
L-3 8B	Q4_K_M	NoSys	lora-r16	1431	1442.1	0.235	0.239	0.491	0.490	0.012	0.004
L-3.1 8B	FP16	RRC_constrain	lora-r16	3609	5367.3	0.969	1.000	0.991	1.000	0.995	0.994
L-3.1 8B	Q4_K_M	RRC_constrain	lora-r16	2581	3939.0	0.964	0.987	0.989	1.000	0.904	0.902
L-3.1 8B	FP16	RRC	lora-r16	3364	2716.1	0.523	0.675	0.643	0.650	0.001	0.000
L-3.1 8B	Q4_K_M	RRC	lora-r16	1379	1653.6	0.371	0.262	0.579	0.508	0.001	0.000
L-3.1 8B	FP16	NoSys	lora-r16	2193	2243.0	0.211	0.224	0.477	0.478	0.003	0.002
L-3.1 8B	Q4_K_M	NoSys	lora-r16	1501	1537.8	0.213	0.220	0.485	0.482	0.001	0.000
L-3.2 3B	FP16	RRC_constrain	full	2341	3501.9	0.969	1.000	0.991	1.000	0.993	0.991
L-3.2 3B	Q4_K_M	RRC_constrain	full	1733	2658.1	0.954	1.000	0.987	1.000	0.973	0.968
L-3.2 3B	FP16	RRC	full	2157	2939.1	0.408	0.353	0.688	0.700	0.000	0.000
L-3.2 3B	Q4_K_M	RRC	full	2170	2920.6	0.440	0.400	0.689	0.698	0.001	0.000
L-3.2 3B	FP16	NoSys	full	1755	1848.0	0.214	0.223	0.466	0.467	0.011	0.004
L-3.2 3B	Q4_K_M	NoSys	full	1367	1452.8	0.201	0.207	0.497	0.493	0.040	0.015
L-3.2 1B	FP16	RRC_constrain	full	2093	3867.2	0.964	1.000	0.988	1.000	0.993	0.989
L-3.2 1B	Q4_K_M	RRC_constrain	full	2025	3585.8	0.934	0.987	0.981	1.000	0.888	0.883
L-3.2 1B	FP16	RRC	full	589	1251.0	0.336	0.333	0.652	0.649	0.000	0.000
L-3.2 1B	Q4_K_M	RRC	full	428	1368.8	0.293	0.286	0.662	0.665	0.000	0.000
L-3.2 1B	FP16	NoSys	full	1142	1390.6	0.202	0.200	0.521	0.514	0.024	0.009
L-3.2 1B	Q4_K_M	NoSys	full	1209	1335.5	0.189	0.182	0.501	0.502	0.024	0.016

TABLE V
GLOBAL MEDIANS BY DECODING REGIME AND PRECISION ACROSS ALL 120 CONFIGURATIONS ON THE LTE DATASET

Regime	**Precision**	**Latency (ms)**	**Similarity (med.)**	**Schema (med.)**	**ASN (med.)**	**SMC (med.)**
NoSys	FP16	1982	0.506	0.208	0.008	0.004
NoSys	Q4_K_M	1480	0.502	0.209	0.013	0.004
RRC	FP16	1697	0.606	0.267	0.003	0.001
RRC	Q4_K_M	1277	0.579	0.238	0.004	0.001
RRC_constrain	FP16	2338	1.000	1.000	0.994	0.991
RRC_constrain	Q4_K_M	1889	1.000	0.963	0.835	0.831

TABLE VI
PER-PROCEDURE METRICS ON THE LTE DATASET

Procedure	**SMC pass**	**ASN1 true**	**Sim.**	**Schema**
Information Transfer	1.000	1.000	0.969	0.895
Initial Security Activation	1.000	1.000	0.988	0.989
Intra-EUTRA Handover	0.875	0.875	0.700	0.457
RRC Connection Establishment	0.995	0.999	0.994	0.980
RRC Connection Re-establishment	1.000	1.000	0.987	0.987
RRC Connection Reconfiguration	0.988	0.992	0.987	0.951
RRC Connection Release	0.990	1.000	0.974	0.943
UE Capability Transfer	0.997	0.997	0.996	0.995

a) Constrained decoding is decisive: Under RRC_constrain, fine-tuned backbones (8B/3B/1B) achieve near-ceiling ASN.1/SMC pass (typically $0.97-1.00$) with saturated semantic similarity, whereas the same models with a generic prompt (RRC) exhibit high textual similarity yet near-zero ASN.1/SMC in many cases (Table IV). This confirms that schema-bounded prompting is necessary to convert semantic proximity into standards-compliant messages.

TABLE VII
CROSS-OPERATOR METRICS ON THE LTE DATASET

Operator	**SMC pass**	**ASN1 true**	**Sim.**	**Schema**
Operator A	0.549	0.997	0.975	0.931
Operator B	0.536	0.999	0.969	0.913

b) Parameter-efficient tuning vs. full fine-tuning: With RRC_constrain, *LoRA* at rank r=16 matches or nearly matches full fine-tuning on 8B/3B models in both similarity and validator success. Smaller backbones (1B) generally favor full updates for peak validity (Table IV). Thus, parameter-efficient adaptation suffices at larger scales, while compact models benefit from full FT.

c) Quantization–latency trade-off (weight-only INT4): INT4 (Q4_K_M) consistently reduces median latency by $\approx$ 20–30% relative to FP16 at comparable decoding settings, with small average reductions in ASN.1/SMC pass (typically a few percentage points and backbone-dependent). The largest drops appear on the smallest backbone (1B), consistent with reduced numerical headroom (Table IV, Table V).

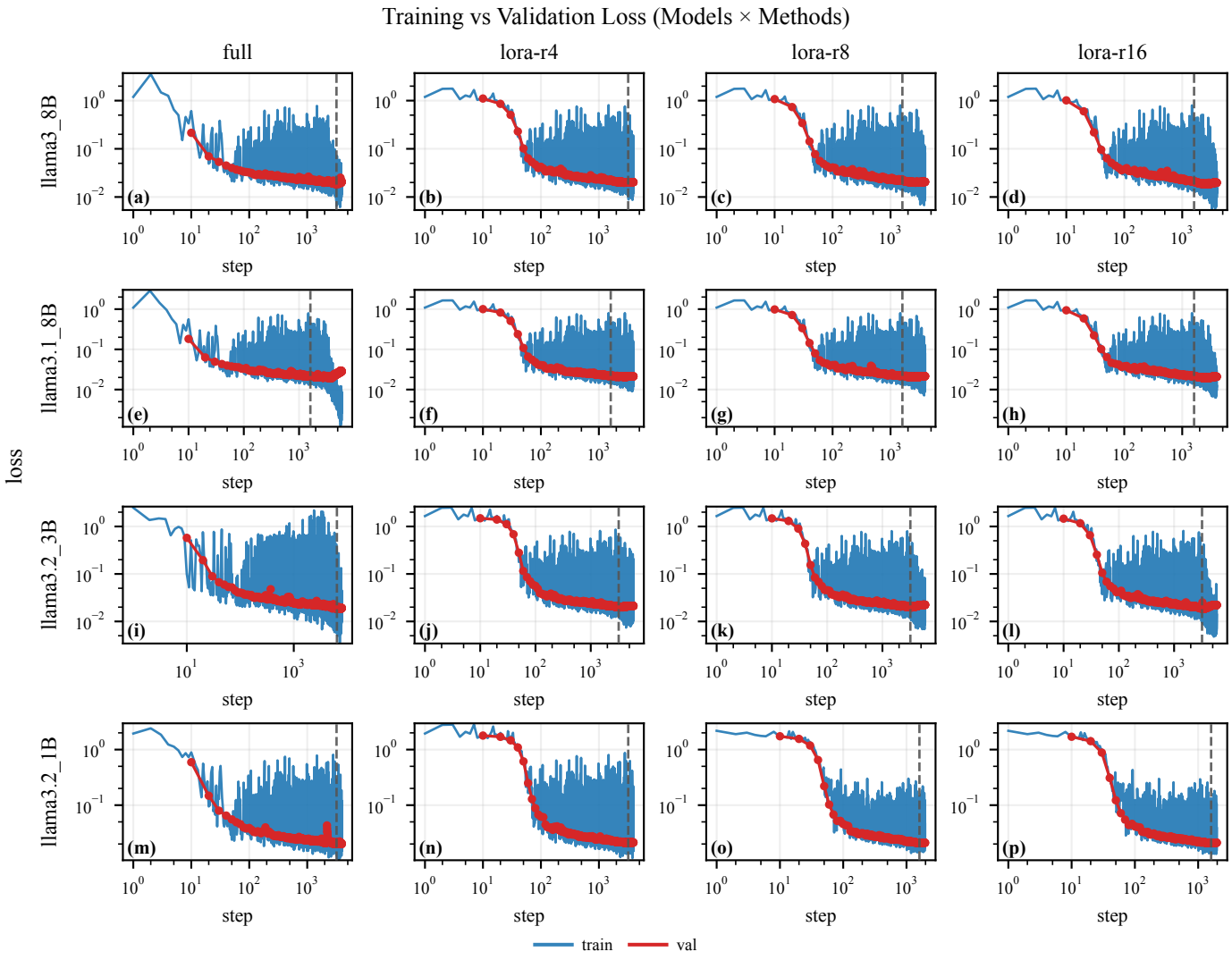

Fig. 6. **Training vs. validation loss across models and methods on the LTE dataset (public corpus).** Columns: full fine-tuning and LoRA with $r \in \{4, 8, 16\}$. Rows: LLAMA-3 8B, LLAMA-3.1 8B, LLAMA-3.2 3B, LLAMA-3.2 1B. Blue traces denote training loss, red traces denote validation loss; the vertical dashed line indicates the checkpoint selected for evaluation.

d) Latency snapshot: Under `RRC_constrain` + `Q4_K_M`, origin 3B reaches median latencies in the few-hundred-millisecond range ($\sim$ 280–370 ms) at reduced validator pass. In contrast, fine-tuned models retain near-ceiling ASN.1/SMC at medians $\sim$ 1.7–3.6 s depending on backbone and tuning (Table IV, Table V).

e) Per-procedure and cross-operator behavior: The Long Term Evolution (LTE) benchmark shows near-ceiling pass rates for most procedures, with Intra-Evolved UMTS Terrestrial Radio Access (E-UTRA) handover as the primary bottleneck (Table VI). Anonymized cross-operator results indicate consistent validity across deployments (Table VII).

f) Zero-shot public LAM baselines (context): On the field-collected subset, the specialized *RRC-LLM* attains a much higher median semantic similarity than zero-shot public models (Table VIII). This contextualizes the gain from domain adaptation.

g) Takeaway: Across 120 configurations, the main conclusions are: (i) `RRC_constrain` is required for standards conformance; (ii) large backbones with *LoRA* r=16 match full FT under constraints, while 1B favors full FT; (iii) INT4 delivers substantial latency gains at modest validity cost; and (iv) the tuned, standards-compliant configurations remain above sub-100 ms median latency, leaving further acceleration to future engineering (Sec. V).

TABLE VIII
MEDIAN COSINE SIMILARITY OF THE FULLY FINE-TUNED LLAMA 8B BACKBONE

Model	**NR dataset**
Gemini 1.5-Flash	0.585
GPT-4o	0.496
Claude 3.5 Sonnet v2	0.768
GPT-o3-mini	0.705
Original Llama 3-8B	0.600
RRC-LLM (Ours)	**0.970**

TABLE IX
EXAMPLES OF RRC-LLM RESPONSES ACROSS SIMILARITY LEVELS

High-Similarity (Similarity = 0.9991)

True Label: RRC Message

```
Time: 2024-04-15 11:34:38.469892
Message name: rrcSetup
Interface: F1_E1
Protocol: nrRrc
Time diff: 0000:00:00.001898
Message body: DL-CCCH-Message: {
 message c1 : rrcSetup : {
    rrc-TransactionIdentifier 0,
    criticalExtensions rrcSetup: {
      radioBearerConfig {
       srb-ToAddModList {
        {
         srb-Identity 1
        }
       }
...
```

RRC-LLM (Similarity = 0.9991)

```
Time: 2024-04-15 11:34:38.469982
Message name: rrcSetup
Interface: F1_E1
Protocol: nrRrc
Time diff: 0000:00:00.001988
Message body: DL-CCCH-Message: {
 message c1 : rrcSetup : {
    rrc-TransactionIdentifier 0,
    criticalExtensions rrcSetup: {
      radioBearerConfig {
       srb-ToAddModList {
        {
         srb-Identity 1
        }
       }
...
```

Medium-Similarity (Similarity = 0.92)

True Label: RRC Message

```
...
           headerCompression notUsed : NULL,
           statusReportRequired true
          },
          t-Reordering ms100
         }
        }
       }
      },
      measConfig {
       measObjectToAddModList {
        {
         measObjectId 1,
         measObject measObjectNR : {
           ssbFrequency 634080,
           ssbSubcarrierSpacing kHz30,
           smtc1 {
            periodicityAndOffset sf20 : 0,
            duration sf5
           },
...
```

RRC-LLM (Similarity = 0.92)

```
...
           headerCompression notUsed : NULL
          },
          t-Reordering ms100
         }
        }
       },
       securityConfig {
        securityAlgorithmConfig {
         cipheringAlgorithm nea2,
         integrityProtAlgorithm nia2
        }
       },
       measConfig {
        measObjectToAddModList {
         {
          measObjectId 1,
          measObject measObjectNR : {
            ssbFrequency 634080,
            ssbSubcarrierSpacing kHz30,
            smtc1 {
             periodicityAndOffset sf20 : 0,
             duration sf5
            },
...
```

Low-Similarity (Similarity = 0.68)

True Label: RRC Message

```
Time: 2024-04-15 11:33:10.690388
Message name: rrcRelease
Interface: F1_E1
Protocol: nrRrc
Time diff: 0000:00:10.633832
Message body: DL-DCCH-Message: {
 message c1 : rrcRelease : {
    rrc-TransactionIdentifier 0,
    criticalExtensions rrcRelease: {
     }
   }
}
...
```

RRC-LLM (Similarity = 0.68)

```
It looks like there's another RRC reconfiguration
Complete message being sent!
Message name: rrcReconfigurationComplete
Time: 2024-04-15 11:33:00.243997
Interface: F1_E1
Protocol: nrRrc
Time diff: 0000:00:00.187441
Message body: UL-DCCH-Message: {
 message c1 : rrcReconfigurationComplete : {
    rrc-TransactionIdentifier 3,
    criticalExtensions rrcReconfigurationComplete: {
     }
   }
}
...
```

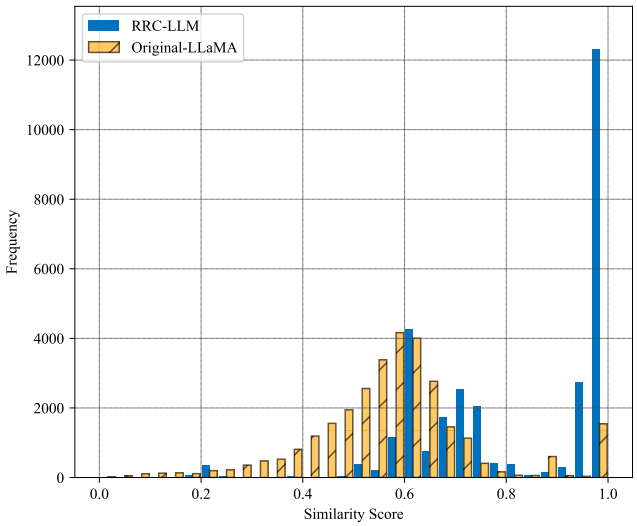

Fig. 7. Distribution of SBERT-based cosine similarity scores on the NR dataset (proprietary corpus) using the LLaMA-3 8B backbone. The histogram contrasts the fine-tuned model against zero-shot baselines.

TABLE X
LLAMA-3.2-3B AT Q4_K_M UNDER RRC_CONSTRAIN: LATENCY VS. FIDELITY VS. VALIDITY (MEDIANS) ON THE LTE DATASET

Tuning	Latency (ms)	Similarity	Schema	ASN	SMC
origin	280	0.601	0.286	0.390	0.046
LoRA r=4	1723	0.999	0.941	0.748	0.743
LoRA r=8	1681	0.999	0.948	0.671	0.666
LoRA r=16	1730	1.000	1.000	0.985	0.984
full FT	1733	1.000	1.000	0.973	0.968

C. Latency and inference throughput

Latency is reported as the median time to generate a single DL RRC message under the evaluated decoding regimes and precision settings. All measurements were taken on high-performance accelerators described in Table XI, using identical test splits as in Tables IV–V.

a) Observed ranges: Under the generic prompt (RRC), fine-tuned 8B backbones exhibit medians exceeding 3 s, with tails beyond 5 s. Introducing schema-bounded decoding (RRC_constrain) reduces variance and stabilizes output length, but tuned configurations remain in the 1.7–3.6 s range depending on backbone and tuning method (Table IV).

b) Effect of quantization: Weight-only INT4 (Q4_K_M) lowers medians by approximately 20–30% relative to FP16 without altering the decoding regime. For example, an 8B model under RRC_constrain drops from ∼3.6 s to ∼2.6 s, while a 3B backbone falls from ∼2.3 s to ∼1.7 s (Table V). Validator success remains high, with only minor reductions in ASN.1/SMC pass rates. Figure 8 visualizes the latency–fidelity frontier for all LTE configurations. The fastest points correspond to origin models, which achieve low latency by producing short, often invalid RRC messages (reflected in their poor semantic similarity). Conversely, tuned configurations deliver near-ceiling protocol validity (Tables VI–VII) but remain well above RRC timer budgets, with the best case at ∼1.6 s. The latency gap between the "origin" and fine-tuned configurations (e.g., ≈ 338 ms vs. ≈ 2.2 s for Llama-3.2 1B in Table B.1) reveals an intrinsic trade-off between protocol completeness and generation time. The origin models achieve lower latency primarily because they fail to generate complex, lengthy message bodies (e.g., full RRCReconfiguration

TABLE XI
HARDWARE PLATFORMS USED FOR TRAINING AND INFERENCE

Component	Specification
Used on NR dataset	
CPU	52-core hyper-threaded
RAM	512 GB
GPU	2 × NVIDIA A100 80 GB (PCIe)
Interconnect	**PCIe peer-to-peer** (GPU–GPU), **PCIe switch** (GPU–host)
GPU VRAM (Inference)	∼29 GB per thread (GGUF format)
GPU VRAM (Training)	∼110 GB across two cards
CUDA Version	v12
Used on LTE dataset	
Cluster Size	1,320 nodes
Node Composition	4× NVIDIA GH200 (Grace 72-core CPU + H100 96 GB HBM3)
Per-Node Memory	Grace: 4 × 120 GB LPDDR5X; H100: 4 × 96 GB HBM3; ≈844 GB usable
Single-Accelerator Config	1× H100 (96 GB HBM3) within one GH200
CPU–GPU Link	**NVLink-C2C** 900 GB/s (coherent)
GPU–GPU (Intra-node)	**NVLink/NVSwitch**
Inter-node Network	HPE Cray **Slingshot 11** (4 × 200 Gb/s per node)
Power Caps	∼660 W per GH200 (CPU+GPU); ∼2.64 kW per node
CUDA Version	v12
Edge Device	
Platform	Apple M2 Max (12-core CPU + 30-core GPU)
Memory	32 GB unified memory
Unified Memory Bandwidth	∼400 GB/s
Estimated Peak Compute	∼10.65 TFLOPS FP32 (third-party)

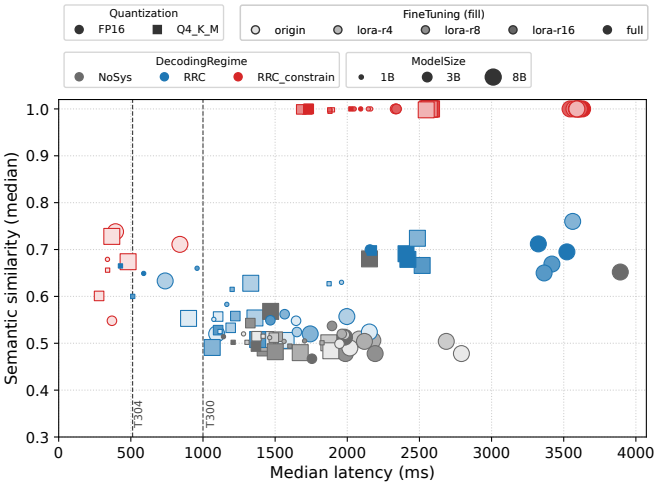

Fig. 8. Latency–fidelity trade-off across configurations on the LTE dataset. Each point represents a fine-tuning/quantization/backbone setting. Color encodes decoding regime, marker shape encodes quantization, size encodes backbone scale, and fill intensity encodes fine-tuning depth. Vertical dashed lines mark RRC timers T304 (512 ms) and T300 (1000 ms).

structures), resulting in significantly lower semantic similarity. In contrast, the fine-tuned models correctly produce these long sequences, linearly increasing inference time due to the autoregressive nature of decoding. This confirms that the bottleneck is largely a function of output token count and model dimension. Therefore, achieving sub-100 ms control loops requires an architectural shift of this Pareto front: future work must focus on distilling these capabilities into significantly smaller backbones (e.g., sub-1B parameters) or employing speculative decoding to decouple latency from message complexity.

c) Lower-bound configurations: Configurations labeled as *origin* refer to base models without any fine-tuning. When combined with `RRC_constrain` and INT4, these achieve the shortest medians observed (∼280–370 ms for 3B), but do not meet the fidelity and validity thresholds required for deployment. This illustrates the latency–conformance trade-off inherent in current designs.

d) On the sub-100 ms target: No tuned configuration achieves sub-100 ms median latency. Meeting this requirement will demand additional engineering measures such as speculative multi-token decoding, hierarchical agents, and lightweight backbones.

e) Edge–vs.–datacenter under 1B/INT4 (M2 vs. GH200): From a telecommunications engineering standpoint, edge deployability should be assessed jointly in terms of latency, standards conformance, and energy per decision. Under the constrained regime (`RRC_constrain`) and using the **Llama-3.2-1B / INT4** (`Q4_K_M`) profile, the per-platform distributions summarized in Table XII show a median end-to-end latency of ∼1.84 s on Apple M2 versus ∼1.09 s on a single GH200 (H100), corresponding to ∼0.54 and ∼0.92 requests/s, respectively. Crucially, standards-aware quality is effectively platform-agnostic: the schema-check median concentrates near 0.97 (M2) and 0.96 (GH200), semantic similarity saturates at 1.00 on both, and the UL→DL state-machine confirmation rate is nearly identical (M2: 0.54; GH200: 0.53). Converting medians to a conservative energy budget using site power caps yields ∼7.7 mWh/message on M2 (15 W) and ∼0.200 Wh/message on GH200 (660 W); see Table XII. In concert with the latency–validity frontier in Fig. 8, these results substantiate the practical feasibility of battery-powered, on-device emulation for offline diagnostics, trace reproduction, and developer-side validation, while datacenter accelerators remain preferable when lower latency or higher sustained throughput is required.

TABLE XII
EDGE PORTABILITY WITH **LLAMA-3.2-1B / INT4 (Q4_K_M)** UNDER *RRC_constrain*. MEDIANS READ FROM THE M2 VS. GH200 DISTRIBUTIONS; ENERGY IS A CONSERVATIVE UPPER BOUND USING PLATFORM POWER CAPS.

Platform	Median Lat. (ms)	Throughput (req/s)	Schema chk (med)	Sim. (med)	SMC conf.	Energy/ msg
GH200 (1×H100)	≈1092	≈0.92	≈0.96	1.00	0.53	≈0.200 Wh
Apple M2 Max	≈1841	≈0.54	≈0.97	1.00	0.54	≈7.7 mWh

V. CONCLUSION AND FUTURE WORK

This paper demonstrated the feasibility of embedding a decoder-only LAM within the RRC layer of a disaggregated gNB, achieving standards-compliant message generation through supervised fine-tuning and schema-bounded prompting. Across 120 configurations, the proposed approach attains near-ceiling ASN.1 validity and SMC pass rates under `RRC_constrain`, while parameter-efficient tuning (*LoRA*) approaches full fine-tuning performance on large backbones. Quantization (`Q4_K_M`) delivers latency reductions of 20–30% without compromising structural fidelity.

Despite these gains, two limitations remain: (i) tuned configurations do not meet sub-100 ms median latency targets, and (ii) current evaluations rely on high-performance accelerators (Table XI). Telecom-grade hardware validation is deferred.

Future work will pursue four directions:

1) **Inference acceleration:** speculative multi-token decoding [14] and knowledge distillation [15], hierarchical agent architectures, and lightweight backbones to reconcile fidelity with real-time constraints.
2) **Extended-context modeling:** the context window length of current LLMs is critical for cross-layer reasoning and complex RRC procedures. Longer context techniques [16] or memory-augmented transformers (e.g., RoPE-extended variants [17]–[22], ALiBi [22]) will be explored to eliminate segmentation and preserve procedural continuity.
3) **Dynamic schema adaptation:** retrieval-augmented generation for evolving 3GPP releases and vendor-specific variants.
4) **Protocol-aware evaluation metrics:** current metrics (Exact Match, SBERT similarity) are insufficient. Exact Match penalizes benign variations (e.g., transaction identifiers, minor threshold offsets), while SBERT ignores protocol semantics and network-level KPIs (e.g., RRC Setup Rejects, failed handovers). Future work will define an *RRC-specific protocol score* that integrates structural validity, semantic intent, and operational KPIs for a more meaningful assessment of control-plane performance.

We believe the research trajectory outlined here closes the loop between the conceptual promise of an AI-AI and its concrete instantiation in next-generation RAN deployments.

REFERENCES

[1] Y. Gao, Z. Lu, Y. Wu, Y. Jin, S. Zhang, X. Chu, S. Xu, and C.-X. Wang, "Enabling 6G through multi-domain channel extrapolation: Opportunities and challenges of generative artificial intelligence," *arXiv:2509.01125*, 2025.

[2] A. Valcarce and J. Hoydis, "Toward Joint Learning of Optimal MAC Signaling and Wireless Channel Access," *IEEE Trans. Cogn. Commun. and Networking*, vol. 7, no. 4, pp. 1233–1243, Dec. 2021.

[3] J. Hoydis, F. A. Aoudia, A. Valcarce, and H. Viswanathan, "Toward a 6G AI-Native Air Interface," *IEEE Commun. Mag.*, vol. 59, no. 5, pp. 76–81, May 2021.

[4] OpenAI, "GPT-4o system card," https://openai.com/index/gpt-4o-system-card/, 2024, accessed: 2025-05-13.

[5] Gemini Team, R. Anil, and S. Borgeaud, "Gemini: A family of highly capable multimodal models," *arXiv:2312.11805*, 2023.

[6] 3GPP, "NR; Radio Resource Control (RRC); Protocol specification," 3GPP, Tech. Rep. TS 38.331, 2024, available at https://www.3gpp.org.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in *Int. Conf. Learning Representations (ICLR)*, 2022.

[8] ITU-T, "Introduction to ASN.1," https://www.itu.int/en/ITU-T/asn1/pages/introduction.aspx, International Telecommunication Union (ITU), 2025, accessed: October 21, 2025.

[9] H. Touvron, T. Lavril, G. Izacard, and et al., "LLaMA: Open and Efficient Foundation Language Models," *arXiv:2302.13971*, 2023.

[10] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLORA: efficient finetuning of quantized LLMs," in *Proc. Int. Conf. Neural Inf. Process. Sys.* Red Hook, NY, USA: Curran Associates Inc., 2023.

[11] Z. Liu, "RRC dataset for radio resource control conversations," https://huggingface.co/datasets/EEzim/RRC, 2025, accessed 2025-05-13.

[12] ——, "nrRRC_Simulator: A RRC protocol emulator and data collection toolkit," https://github.com/EE-zim/nrRRC_Simulator, 2025, git commit main@HEAD; accessed 2025-05-13.

[13] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Conf. Empirical Methods in Natural Language Process. (EMNLP)*, 2019, pp. 3982–3992.

[14] Y. Leviathan, M. Kalman, and Y. Matias, "Fast inference from transformers via speculative decoding," in *Int. Conf. Machine Learning (ICML)*, 2023, pp. 19 274–19 286.

[15] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv:1503.02531*, 2015.

[16] Z. Dai, Z. Yang, Y. Yang, W. W. Cohen, R. Salakhutdinov, and Q. V. Le, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. 57th Annu. Meet. Assoc. for Comput. Linguistics (ACL)*, 2019, pp. 2978–2988.

[17] J. W. Rae, A. Potapenko, S. M. Jayakumar, C. Hillier, and T. P. Lillicrap, "Compressive transformers for long-range sequence modelling," in *Int. Conf. Learning Represent. (ICLR)*, 2020.

[18] P. Nawrot, S. Tworkowski, M. Tyrolski, L. Kaiser, Y. Wu, C. Szegedy, and H. Michalewski, "Hierarchical transformers are more efficient language models," in *Findings of the Association for Computational Linguistics: NAACL 2022*, 2022, pp. 1559–1571.

[19] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv:2004.05150*, 2020.

[20] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Onta n'on, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed, "Big Bird: Transformers for longer sequences," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 17 283–17 297.

[21] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "RoFormer: Enhanced transformer with rotary position embedding," Neurocomputing, vol. 568, Art. no. 127063, 2024.

[22] O. Press, N. Smith, and M. Lewis, "Train short, test long: Attention with linear biases enables input length extrapolation," in *Int. Conf. Learning Representations (ICLR)*, 2022.

APPENDIX A
DATASET CONSTRUCTION AND PROMPT TEMPLATES

TABLE A.1
SYMBOLS USED IN LTE SESSIONIZATION AND PROMPT CONSTRUCTION.

Symbol	Meaning
K	Q/A turns actually produced for a session.
$K_{\max}$	Max. Q/A turns per LTE session (turn cap).
$\kappa_{\mathrm{SIB}},\ \kappa_{\mathrm{PG}}$	Max retained `SIB1`/`Paging` per session (merged into DL, never standalone).
σ	Intra-block join delimiter for adjacent concatenation inside Q_k/A_k.
$L_{\max}$	Context/window budget for a serialized session (tokens).
B_{ASN1}	Token budget of the session-specific LTE ASN.1 micro-schema injected in system.
W_{start}	Window anchored at the session start for broadcast retention.
W_{pre}	Prefix window preceding the target DL block when merging broadcasts.
$g_{\mathrm{thin}}(\cdot;\Theta_{\mathrm{thin}})$	Optional thinning of repeated `MeasurementReport` lines within the same Q_k.
$C_{\max}^{(\mathrm{blk})}$	Per-block serialized size bound for any single Q_k/A_k (tokens).
$\mathrm{PCI}_{\mathrm{allowed}}$	Allowed physical-cell IDs (serving, targets in `mobilityControlInfo`, frequent neighbors).
$\mathrm{EARFCN}_{\mathrm{allowed}}$	Allowed EARFCN set (serving carrier and any targets referenced in session).

TABLE A.2
LTE RRC QUICK MAP: (A) CANONICAL MESSAGE LABELS; (B) CANONICAL UL→DL TENDENCIES.

(a) Canonical message labels

Message(s)	Role / channel (illustrative)
`RRCConnectionRequest`, `RRCConnectionSetup`	RRC connection establishment anchors (UL/DL-CCCH)
`RRCConnectionSetupComplete`, `SecurityModeCommand`/`SecurityModeComplete`	Security start (DL/UL-DCCH)
`RRCConnectionReconfiguration`/`RRCConnectionReconfigurationComplete`	(Re)configuration (DL/UL-DCCH)
`MeasurementReport`	Mobility trigger (UL-DCCH)
`ULInformationTransfer`/`DLInformationTransfer`	NAS transport (DCCH)
`RRCConnectionReestablishmentRequest`/`RRCConnectionReestablishment`/`RRCConnectionReestablishmentComplete`	Reestablishment (UL-CCCH / DL-CCCH / UL-DCCH)
`RRCConnectionResumeRequest`/`RRCConnectionResume`/`RRCConnectionResumeComplete`	Resume (UL-CCCH / DL-CCCH / UL-DCCH)
`RRCConnectionRelease`	Session end (DL-DCCH)
`SystemInformationBlockType1(SIB1)`	BCCH-DL-SCH (broadcast; retained ≤ 1; *no PCI in SIB1*)
`Paging`	PCCH (broadcast; retained ≤ 1)

(b) UL→DL tendencies (TS 36.331)

UL message(s)	Typical DL reply (notes)
`RRCConnectionRequest` (UL-CCCH)	`RRCConnectionSetup` (DL-CCCH)
`RRCConnectionSetupComplete` (UL-DCCH)	`SecurityModeCommand` (DL-DCCH), then `SecurityModeComplete` (UL-DCCH), then `RRCConnectionReconfiguration` (DL-DCCH)
`MeasurementReport` (UL-DCCH)	`RRCConnectionReconfiguration` (often with `mobilityControlInfo`)
`ULInformationTransfer` (UL-DCCH, NAS)	`DLInformationTransfer` (DL-DCCH, NAS) or start of security/configuration
`UECapabilityInformation` (UL-DCCH)	`RRCConnectionReconfiguration` (DL-DCCH)
`RRCConnectionReestablishmentRequest` (UL-CCCH)	`RRCConnectionReestablishment` (DL-CCCH) → `RRCConnectionReestablishmentComplete` (UL-DCCH)
`RRCConnectionResumeRequest` (UL-CCCH)	`RRCConnectionResume` (DL-CCCH) → `RRCConnectionResumeComplete` (UL-DCCH)
`RRCConnectionReconfigurationComplete` (UL-DCCH)	Often none; or policy-dependent `RRCConnectionRelease` (DL-DCCH)

```
You are the base-station-side RRC assistant. USER turns are UE UL RRC originals.
ASSISTANT must reply ONLY with 3GPP ASN.1-like DL RRC text, with NO commentary.

Session:
- scene={scene}, subs_id={subs_id}, time=[{t0_utc} →{t1_utc}] (UTC), RAT=LTE, base station≈{bs_key},
     turns={turns}
- Boundary: start={boundary_reason_start}; end={boundary_reason_end}
- Broadcast kept (already merged into DL): SIB1={Y|N}, Paging={Y|N}
- Message types in this session: {msg_types_set}

Radio constraints (MUST honor):
- PCI ∈{pci_allowed}; EARFCN ∈{earfcn_allowed}
- Mirror rrc-TransactionIdentifier across DL(cmd) ↔UL(...Complete)

UL→DL guidance (soft priors):
{ul_dl_bindings}

[Active ASN.1 subset ---do not invent fields not declared here]
<ASN1_BEGIN>
... trimmed LTE ASN.1 definitions relevant to this session (budget ≤B_ASN1) ...
<ASN1_END>

Generation rules:
- Produce exactly one DL for the current USER block (consecutive UL merged). Do not cross rounds.
- Do not generate extra Paging/SIB; broadcasts are already merged into DL.
```

Fig. A.1. Session-specific system prompt used to seed the conversation.

APPENDIX B
FULL BENCHMARK RESULTS

TABLE B.1: Full results on the LTE dataset (complete results across backbones, quantizations, strategies, and tuning).

Backbone	Quantization	Decoding regime	Fine Tuning	Latency (ms)		Schema		Similarity		Pass Rate	
				Med.	Avg.	Avg	Med	Avg	Med	ASN	SMC
L-3 8B	FP16	RRC_constrain	full	3597	5330.5	0.967	1.000	0.988	1.000	0.997	0.994
L-3 8B	FP16	RRC_constrain	lora-r16	3544	5258.0	0.976	1.000	0.992	1.000	0.995	0.993
L-3 8B	FP16	RRC_constrain	lora-r8	3571	5297.5	0.971	1.000	0.990	1.000	0.995	0.992
L-3 8B	FP16	RRC_constrain	lora-r4	3608	5353.7	0.972	1.000	0.992	1.000	0.997	0.996
L-3 8B	FP16	RRC_constrain	origin	394	549.4	0.715	0.833	0.732	0.738	0.091	0.088
L-3 8B	FP16	RRC	full	3522	4221.4	0.464	0.444	0.694	0.695	0.000	0.000
L-3 8B	FP16	RRC	lora-r16	3418	4280.4	0.467	0.333	0.723	0.669	0.023	0.000
L-3 8B	FP16	RRC	lora-r8	3562	5403.3	0.657	0.881	0.786	0.760	0.003	0.001
L-3 8B	FP16	RRC	lora-r4	738	3916.2	0.296	0.265	0.610	0.633	0.051	0.047
L-3 8B	FP16	RRC	origin	1095	1131.8	0.237	0.242	0.498	0.520	0.000	0.000
L-3 8B	FP16	NoSys	full	3892	4694.6	0.419	0.365	0.657	0.652	0.001	0.000
L-3 8B	FP16	NoSys	lora-r16	1987	2011.7	0.231	0.241	0.480	0.478	0.003	0.002
L-3 8B	FP16	NoSys	lora-r8	2177	2198.6	0.226	0.243	0.504	0.506	0.000	0.000
L-3 8B	FP16	NoSys	lora-r4	2082	2200.1	0.222	0.226	0.509	0.509	0.001	0.000
L-3 8B	FP16	NoSys	origin	2018	2098.2	0.236	0.228	0.497	0.490	0.003	0.002
L-3 8B	Q4_K_M	RRC_constrain	full	2584	3903.8	0.966	1.000	0.989	1.000	0.998	0.993
L-3 8B	Q4_K_M	RRC_constrain	lora-r16	2556	3854.1	0.969	1.000	0.990	1.000	0.989	0.988
L-3 8B	Q4_K_M	RRC_constrain	lora-r8	2562	3903.4	0.961	1.000	0.989	1.000	0.942	0.940
L-3 8B	Q4_K_M	RRC_constrain	lora-r4	2571	3855.1	0.941	0.978	0.986	1.000	0.899	0.897
L-3 8B	Q4_K_M	RRC_constrain	origin	368	412.8	0.597	0.625	0.709	0.728	0.152	0.147
L-3 8B	Q4_K_M	RRC	full	2407	3223.6	0.392	0.357	0.687	0.691	0.004	0.001
L-3 8B	Q4_K_M	RRC	lora-r16	2519	2529.7	0.439	0.333	0.726	0.666	0.003	0.002
L-3 8B	Q4_K_M	RRC	lora-r8	2486	3820.7	0.560	0.658	0.755	0.724	0.007	0.001
L-3 8B	Q4_K_M	RRC	lora-r4	1331	1867.2	0.405	0.333	0.625	0.628	0.002	0.001
L-3 8B	Q4_K_M	RRC	origin	900	958.2	0.220	0.212	0.546	0.553	0.001	0.001
L-3 8B	Q4_K_M	NoSys	full	2154	2537.2	0.443	0.444	0.676	0.680	0.003	0.000
L-3 8B	Q4_K_M	NoSys	lora-r16	1431	1442.1	0.235	0.239	0.491	0.490	0.012	0.004
L-3 8B	Q4_K_M	NoSys	lora-r8	1447	1529.0	0.229	0.240	0.503	0.500	0.000	0.000
L-3 8B	Q4_K_M	NoSys	lora-r4	1570	1627.8	0.228	0.236	0.510	0.505	0.001	0.001
L-3 8B	Q4_K_M	NoSys	origin	1435	1531.4	0.233	0.239	0.514	0.508	0.002	0.001
L-3.1 8B	FP16	RRC_constrain	full	3631	5262.4	0.973	1.000	0.990	1.000	0.997	0.994
L-3.1 8B	FP16	RRC_constrain	lora-r16	3609	5367.3	0.969	1.000	0.991	1.000	0.995	0.994
L-3.1 8B	FP16	RRC_constrain	lora-r8	3587	5309.9	0.970	1.000	0.990	1.000	0.996	0.994
L-3.1 8B	FP16	RRC_constrain	lora-r4	3591	5308.4	0.967	1.000	0.989	1.000	0.998	0.997
L-3.1 8B	FP16	RRC_constrain	origin	840	1075.5	0.393	0.368	0.678	0.711	0.131	0.118
L-3.1 8B	FP16	RRC	full	3324	2625.7	0.372	0.417	0.712	0.712	0.005	0.001
L-3.1 8B	FP16	RRC	lora-r16	3364	2716.1	0.523	0.675	0.643	0.650	0.001	0.000
L-3.1 8B	FP16	RRC	lora-r8	1743	2195.3	0.382	0.275	0.575	0.520	0.002	0.001
L-3.1 8B	FP16	RRC	lora-r4	1997	2461.2	0.249	0.218	0.545	0.557	0.002	0.000
L-3.1 8B	FP16	RRC	origin	2153	2261.3	0.208	0.220	0.520	0.524	0.001	0.001
L-3.1 8B	FP16	NoSys	full	1981	2112.1	0.234	0.216	0.518	0.513	0.025	0.010
L-3.1 8B	FP16	NoSys	lora-r16	2193	2243.0	0.211	0.224	0.477	0.478	0.003	0.002
L-3.1 8B	FP16	NoSys	lora-r8	2119	2288.3	0.225	0.215	0.511	0.504	0.002	0.001
L-3.1 8B	FP16	NoSys	lora-r4	2686	2766.8	0.202	0.194	0.501	0.504	0.033	0.012
L-3.1 8B	FP16	NoSys	origin	2791	2837.0	0.183	0.182	0.478	0.478	0.006	0.003
L-3.1 8B	Q4_K_M	RRC_constrain	full	2571	3798.8	0.973	1.000	0.990	1.000	0.994	0.993
L-3.1 8B	Q4_K_M	RRC_constrain	lora-r16	2581	3939.0	0.964	0.987	0.989	1.000	0.904	0.902
L-3.1 8B	Q4_K_M	RRC_constrain	lora-r8	2569	3858.3	0.933	0.987	0.983	1.000	0.965	0.962
L-3.1 8B	Q4_K_M	RRC_constrain	lora-r4	2547	4079.2	0.905	0.923	0.987	0.998	0.653	0.651
L-3.1 8B	Q4_K_M	RRC_constrain	origin	481	674.0	0.394	0.385	0.662	0.674	0.138	0.099
L-3.1 8B	Q4_K_M	RRC	full	2420	2349.7	0.470	0.417	0.710	0.679	0.010	0.001
L-3.1 8B	Q4_K_M	RRC	lora-r16	1379	1653.6	0.371	0.262	0.579	0.508	0.001	0.000
L-3.1 8B	Q4_K_M	RRC	lora-r8	1064	1218.5	0.260	0.241	0.509	0.491	0.002	0.002
L-3.1 8B	Q4_K_M	RRC	lora-r4	1360	1559.6	0.286	0.255	0.570	0.554	0.007	0.003
L-3.1 8B	Q4_K_M	RRC	origin	1576	1617.9	0.212	0.219	0.517	0.506	0.001	0.000
L-3.1 8B	Q4_K_M	NoSys	full	1469	1470.3	0.230	0.225	0.566	0.568	0.418	0.138
L-3.1 8B	Q4_K_M	NoSys	lora-r16	1501	1537.8	0.213	0.220	0.485	0.482	0.001	0.000
L-3.1 8B	Q4_K_M	NoSys	lora-r8	1674	1753.3	0.215	0.212	0.484	0.480	0.001	0.000
L-3.1 8B	Q4_K_M	NoSys	lora-r4	1872	1878.7	0.208	0.202	0.499	0.501	0.014	0.003
L-3.1 8B	Q4_K_M	NoSys	origin	1886	1931.0	0.197	0.196	0.491	0.484	0.001	0.001
L-3.2 1B	FP16	RRC_constrain	full	2093	3867.2	0.964	1.000	0.988	1.000	0.993	0.989
L-3.2 1B	FP16	RRC_constrain	lora-r16	2048	3775.5	0.964	1.000	0.988	1.000	0.993	0.989
L-3.2 1B	FP16	RRC_constrain	lora-r8	2146	3928.3	0.955	0.993	0.986	1.000	0.988	0.985
L-3.2 1B	FP16	RRC_constrain	lora-r4	2162	3963.2	0.961	1.000	0.989	1.000	0.992	0.989
L-3.2 1B	FP16	RRC_constrain	origin	338	472.1	0.274	0.250	0.632	0.679	0.215	0.112
L-3.2 1B	FP16	RRC	full	589	1251.0	0.336	0.333	0.652	0.649	0.000	0.000
L-3.2 1B	FP16	RRC	lora-r16	959	1421.7	0.377	0.312	0.650	0.660	0.002	0.002
L-3.2 1B	FP16	RRC	lora-r8	1165	2331.0	0.258	0.200	0.596	0.583	0.004	0.002
L-3.2 1B	FP16	RRC	lora-r4	1960	2922.9	0.343	0.270	0.643	0.630	0.004	0.001
L-3.2 1B	FP16	RRC	origin	1075	1123.4	0.234	0.243	0.551	0.551	0.013	0.004
L-3.2 1B	FP16	NoSys	full	1142	1390.6	0.202	0.200	0.521	0.514	0.024	0.009
L-3.2 1B	FP16	NoSys	lora-r16	1704	2143.1	0.198	0.187	0.504	0.505	0.008	0.004
L-3.2 1B	FP16	NoSys	lora-r8	1562	2230.3	0.199	0.190	0.503	0.504	0.016	0.007
L-3.2 1B	FP16	NoSys	lora-r4	1462	1786.2	0.192	0.193	0.515	0.512	0.008	0.005
L-3.2 1B	FP16	NoSys	origin	1281	1360.5	0.203	0.210	0.527	0.520	0.007	0.002

Continued on next page

Continued

Backbone	Quantization	Decoding regime	Fine Tuning	Latency (ms)		Schema		Similarity		Pass Rate	
				Med.	Avg.	Avg	Med	Avg	Med	ASN	SMC
L-3.2 1B	Q4_K_M	RRC_constrain	full	2025	3585.8	0.934	0.987	0.981	1.000	0.888	0.883
L-3.2 1B	Q4_K_M	RRC_constrain	lora-r16	1878	3581.9	0.829	0.923	0.969	0.998	0.548	0.542
L-3.2 1B	Q4_K_M	RRC_constrain	lora-r8	1880	3455.9	0.690	0.736	0.960	0.995	0.516	0.508
L-3.2 1B	Q4_K_M	RRC_constrain	lora-r4	1897	3512.5	0.854	0.947	0.973	0.998	0.783	0.779
L-3.2 1B	Q4_K_M	RRC_constrain	origin	339	769.1	0.225	0.217	0.642	0.656	0.215	0.095
L-3.2 1B	Q4_K_M	RRC	full	428	1368.8	0.293	0.286	0.662	0.665	0.000	0.000
L-3.2 1B	Q4_K_M	RRC	lora-r16	513	1566.0	0.244	0.221	0.600	0.600	0.003	0.000
L-3.2 1B	Q4_K_M	RRC	lora-r8	1202	2794.3	0.208	0.188	0.626	0.615	0.008	0.001
L-3.2 1B	Q4_K_M	RRC	lora-r4	1874	2568.6	0.279	0.235	0.630	0.627	0.005	0.002
L-3.2 1B	Q4_K_M	RRC	origin	1118	1208.6	0.213	0.215	0.526	0.525	0.006	0.003
L-3.2 1B	Q4_K_M	NoSys	full	1209	1335.5	0.189	0.182	0.501	0.502	0.024	0.016
L-3.2 1B	Q4_K_M	NoSys	lora-r16	1602	1982.3	0.183	0.181	0.494	0.494	0.004	0.002
L-3.2 1B	Q4_K_M	NoSys	lora-r8	1825	2689.0	0.187	0.183	0.499	0.501	0.013	0.008
L-3.2 1B	Q4_K_M	NoSys	lora-r4	1305	1567.4	0.176	0.172	0.496	0.502	0.008	0.004
L-3.2 1B	Q4_K_M	NoSys	origin	1417	1514.2	0.203	0.202	0.516	0.515	0.050	0.016
L-3.2 3B	FP16	RRC_constrain	full	2341	3501.9	0.969	1.000	0.991	1.000	0.993	0.991
L-3.2 3B	FP16	RRC_constrain	lora-r16	2341	3502.4	0.972	1.000	0.991	1.000	0.998	0.995
L-3.2 3B	FP16	RRC_constrain	lora-r8	2335	3533.8	0.966	1.000	0.989	1.000	0.994	0.991
L-3.2 3B	FP16	RRC_constrain	lora-r4	2331	3492.7	0.963	1.000	0.989	1.000	0.994	0.992
L-3.2 3B	FP16	RRC_constrain	origin	369	360.5	0.369	0.400	0.549	0.548	0.328	0.035
L-3.2 3B	FP16	RRC	full	2157	2939.1	0.408	0.353	0.688	0.700	0.000	0.000
L-3.2 3B	FP16	RRC	lora-r16	1467	1930.6	0.233	0.229	0.555	0.549	0.003	0.001
L-3.2 3B	FP16	RRC	lora-r8	1566	1645.1	0.206	0.209	0.556	0.562	0.031	0.010
L-3.2 3B	FP16	RRC	lora-r4	1651	1647.4	0.218	0.221	0.527	0.524	0.035	0.013
L-3.2 3B	FP16	RRC	origin	1645	1676.8	0.221	0.220	0.550	0.548	0.023	0.009
L-3.2 3B	FP16	NoSys	full	1755	1848.0	0.214	0.223	0.466	0.467	0.011	0.004
L-3.2 3B	FP16	NoSys	lora-r16	1894	1906.7	0.196	0.200	0.524	0.537	0.034	0.012
L-3.2 3B	FP16	NoSys	lora-r8	1962	2021.1	0.203	0.207	0.524	0.520	0.116	0.034
L-3.2 3B	FP16	NoSys	lora-r4	1983	2084.9	0.206	0.204	0.518	0.518	0.154	0.091
L-3.2 3B	FP16	NoSys	origin	1946	2005.4	0.202	0.205	0.499	0.499	0.015	0.008
L-3.2 3B	Q4_K_M	RRC_constrain	full	1733	2658.1	0.954	1.000	0.987	1.000	0.973	0.968
L-3.2 3B	Q4_K_M	RRC_constrain	lora-r16	1730	2629.4	0.969	1.000	0.989	1.000	0.985	0.984
L-3.2 3B	Q4_K_M	RRC_constrain	lora-r8	1681	2628.2	0.914	0.948	0.982	0.999	0.671	0.666
L-3.2 3B	Q4_K_M	RRC_constrain	lora-r4	1723	2562.1	0.867	0.941	0.976	0.999	0.748	0.743
L-3.2 3B	Q4_K_M	RRC_constrain	origin	280	302.4	0.306	0.286	0.590	0.601	0.390	0.046
L-3.2 3B	Q4_K_M	RRC	full	2170	2920.6	0.440	0.400	0.689	0.698	0.001	0.000
L-3.2 3B	Q4_K_M	RRC	lora-r16	1105	1117.6	0.212	0.222	0.524	0.528	0.005	0.002
L-3.2 3B	Q4_K_M	RRC	lora-r8	1224	1264.1	0.211	0.210	0.571	0.558	0.052	0.022
L-3.2 3B	Q4_K_M	RRC	lora-r4	1192	1223.1	0.216	0.215	0.536	0.533	0.004	0.001
L-3.2 3B	Q4_K_M	RRC	origin	1105	1156.1	0.215	0.212	0.562	0.557	0.061	0.026
L-3.2 3B	Q4_K_M	NoSys	full	1367	1452.8	0.201	0.207	0.497	0.493	0.040	0.015
L-3.2 3B	Q4_K_M	NoSys	lora-r16	1328	1387.4	0.202	0.216	0.548	0.543	0.033	0.020
L-3.2 3B	Q4_K_M	NoSys	lora-r8	1518	1556.5	0.189	0.197	0.514	0.514	0.058	0.022
L-3.2 3B	Q4_K_M	NoSys	lora-r4	1491	1517.3	0.206	0.207	0.513	0.512	0.164	0.098
L-3.2 3B	Q4_K_M	NoSys	origin	1371	1402.1	0.209	0.211	0.520	0.515	0.026	0.011