EM-LENS: TOWARDS OBSERVATION-LEVEL ELECTROMAGNETIC SIGNAL UNDERSTANDING VIA AGENTIC ZOOM-AND-MEASURE

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing EM multimodal large language models (MLLMs) predominantly study isolated, pre-segmented signal instances, whereas real-world receivers continuously observe long streams containing multiple heterogeneous signal events. Bridging this gap toward real-world observation-level EM understanding is impeded by two critical bottlenecks: **(1)** the limited availability of observation-level EM data, and **(2)** the perceptual degradation that occurs when models reason directly over entire observations, as prolonged heterogeneous backgrounds dilute target signals. To address data scarcity, we introduce **longEM-IQA** and **longEM-Bench**, a training and evaluation suite comprising 11 in-phase/quadrature (I/Q)–language tasks spanning signal localization, identification, parameter estimation, and composite-entity understanding. To mitigate perceptual degradation, we propose **EM-Lens**, a framework that actively localizes target signal segments within complex observations and extracts their physical parameters via explicit signal-processing tools, while retaining end-to-end perception of individual segments. Furthermore, we devise a hybrid training strategy to reconcile the optimization conflict between active analysis and end-to-end perception. Extensive experiments demonstrate that EM-Lens achieves state-of-the-art performance on longEM-Bench, showing that active acquisition of relevant signal segments coupled with explicit measurements enables more reliable observation-level EM signal understanding.

1 INTRODUCTION

Unlike carefully segmented snippets, real-world electromagnetic (EM) observations are long, continuous streams of heterogeneous signal sources, dynamic interference, and environmental noise, making observation-level understanding essential for autonomous EM sensing and reliable decision-making (Haykin, 2005; Yücek & Arslan, 2009).

Multimodal large language models (MLLMs) connect perceptual features with language through cross-modal pretraining and instruction tuning (Li et al., 2023; Liu et al., 2023). Recent EM models extend these capabilities to signal perception and reasoning (Shen et al., 2026; Han et al., 2026), including wideband scene analysis (Zou et al., 2026a). Prior work on modulation recognition emphasizes signal classification (Dobre et al., 2007; O'Shea & Hoydis, 2017), while real receivers must also recover unknown target locations and durations. Our focus is a jointly trained process that integrates query-conditioned localization within a long I/Q observation, local signal perception, explicit physical measurement, and semantic interpretation. This raises a central question:

How can we build an MLLM that understands observation-level EM signals?

We identify two challenges: (1) scarce observation-level EM data, and (2) sparse targets embedded in long heterogeneous backgrounds, where direct full-observation reasoning causes severe perceptual degradation.

To address the first, we introduce **longEM-IQA** and **longEM-Bench**, a training/evaluation suite of I/Q–language tasks spanning localization, identification, parameter estimation, and composite-entity understanding.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

For the second, query-relevant targets occupy only a small fraction of an observation, inducing two training difficulties: (1) *low semantic density*, as irrelevant backgrounds corrupt the representation; and (2) *low semantic resolution*, as fine-grained target characteristics cannot be resolved.

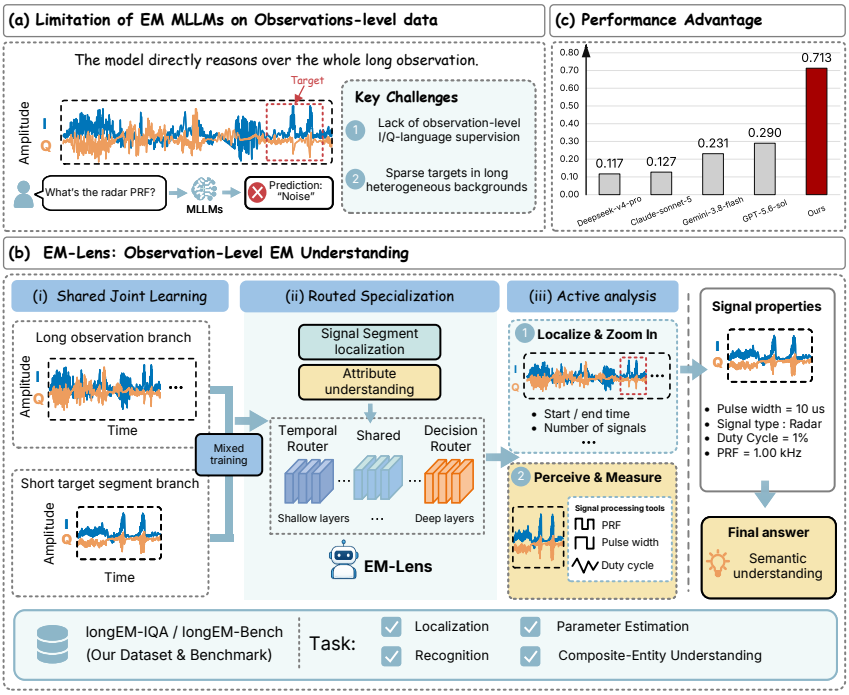


Figure 1: Overview of EM-Lens for observation-level electromagnetic (EM) understanding. (a) Direct reasoning over full long I/Q observations can miss sparse targets in heterogeneous backgrounds. (b) EM-Lens combines joint learning on long observations and target-centered short segments, routed specialization via shallow temporal and deep decision routers over a shared backbone, and active localization with tool-based measurement. longEM-IQA/longEM-Bench span localization, recognition, parameter estimation, and composite-entity understanding.

To address both, we propose **EM-Lens**, replacing one-shot passive reasoning over full observations with active evidence acquisition: the model localizes informative regions, extracts target segments, and invokes explicit signal-processing tools for physical-parameter analysis.

Accurate analysis still requires strong perception of retrieved segments, which we strengthen with target-centered short segments: long observations offer realistic context but suffer from severe *supervision dilution*—small targets receive weak fine-grained supervision—while short, semantically dense segments directly expose discriminative waveforms. Since naive long–short mixing cannot combine the two, we introduce *routed specialization* rather than separate networks: lightweight temporal and decision routers dispatch long/short-context and analysis/perception decisions to dedicated update paths over a shared backbone, gaining fine-grained perception without sacrificing the long-context localization and tool-based reasoning needed for observation-level intelligence.

Our main contributions are:

- **longEM-IQA** and **longEM-Bench**, a training/evaluation suite spanning localization, identification, parameter estimation, and composite-entity understanding.
- **EM-Lens**, an agentic multimodal framework that actively localizes target segments and invokes explicit signal-processing tools, combining precise physical analysis with end-to-end reasoning.
- A routed long–short hybrid training strategy: temporal and decision routers reconcile observation-level localization, active analysis, and fine-grained segment perception across jointly trained long observations and target-centered short segments.

2 Related Work

MLLMs in EM Signal Understanding. Early networks learn modulation and signal identification from I/Q samples (O'Shea et al., 2016; 2017; 2018), while ORACLE studies transmitter fingerprinting (Sankhe et al., 2019). EMind (Luo et al., 2025a) and SpectrumFM (Zhou et al., 2026) learn transferable EM representations through pretraining. MERLIN (Shen et al., 2026), PReD (Han et al., 2026), RadioLLM (Chen et al., 2025), and RF-VLM (Zou et al., 2026b) connect signal representations with language models. RF-GPT studies wideband scenes and overlapping signals (Zou et al., 2026a), whereas SpectrumLLM predicts radio spectrum-state sequences (Liu et al., 2026). EM-Lens focuses on joint training for query-conditioned localization, local perception, and explicit physical measurement in long I/Q observations. Appendix F provides further background.

Tool Use by Agents. ReAct interleaves reasoning and actions (Yao et al., 2023); Toolformer learns tool use through self-supervision (Schick et al., 2023); and ToolLLM develops API-use data and models (Qin et al., 2024). PAL and Program of Thoughts delegate numerical computation to programs (Gao et al., 2023; Chen et al., 2023). ViperGPT and VisProg compose visual tools (Surís et al., 2023; Gupta & Kembhavi, 2023), and V* searches for query-relevant visual details (Wu & Xie, 2024). Other work trains multi-turn agents with reinforcement learning (Wei et al., 2026; Luo et al., 2025b) or structured mid-level supervision (Jiang & Ferraro, 2026). EM-Lens operates on raw I/Q observations: the policy selects a temporal region, acquires its signal representation, and invokes physical estimators before answering.

Electromagnetic Datasets and Benchmarks. TeleCom-Bench evaluates telecom knowledge and network-operation workflows (Xiao et al., 2026). MERLIN introduces EM-134K and EM-Bench, covering perception, jamming-segment detection, and strategy generation (Shen et al., 2026). PReD adds waveform, spectrogram, and constellation inputs with perception-to-decision tasks (Han et al., 2026). SensorBench evaluates code-based processing of sensor data (Quan et al., 2025). EMRB provides 200 problems across five levels over raw, full-length I/Q captures, including multi-emitter analysis with ReconPilot (Zhang et al., 2026). Alongside these benchmarks, longEM-IQA supplies supervised localization, cropping, perception, and measurement trajectories for training an EM-native policy; longEM-Bench evaluates its long-observation understanding.

3 Dataset

Existing EM benchmarks cover signal perception, reasoning, and code-based reconnaissance (Shen et al., 2026; Han et al., 2026; Zhang et al., 2026). We introduce **longEM-IQA** and **longEM-Bench** for understanding complete long-duration I/Q observations where communication, radar, and jamming events can coexist. Their I/Q–language instructions cover temporal localization, signal identification, physical parameter estimation, and composite-entity understanding. Appendix D presents five examples of protocol identification and parameter estimation, including observations, questions, answers, and recorded trajectories.

Controllable simulation of communication, radar, and jamming signals provides longEM with event intervals, attributes, and physical parameters. Its 11 tasks comprise three identification, six parameter estimation, and two composite-entity understanding tasks. Localization is supervised through trajectories and evaluated within composite-entity scores, not as a standalone task. Long training observations contain 24,575–30,000 I/Q samples each. Short-observation supervision uses EM-134K (Shen et al., 2026). Evaluation uses longEM-Bench (11 tasks, 3,300 examples) and the EM-Bench perception subset (10 tasks, 3,000 examples), each with 300 examples per task. Table 1 lists training and validation question–answer (QA) counts and benchmark composition.

Construction of Agentic CoT Trajectories. We construct agentic chain-of-thought (CoT) trajectories (Wei et al., 2022) using a teacher runtime with access to simulation ground truth and signal-processing tools. Given an observation, query, and labels, the teacher identifies the target interval, constructs a structured trajectory skeleton, and calls `zoom_in` to obtain a local view. It then either infers the requested attribute directly from the crop or invokes domain-specific digital signal processing (DSP) tools for measurement before integrating the evidence into the answer. A teacher language model rewrites these trajectories subject to the structured facts and skeletons. Quality checks filter the resulting traces for structural validity, reference consistency, and numeric correctness.

Table 1: Data used for supervised learning and evaluation. Counts refer to QA examples. Long-observation data cover 11 task types; long training waveforms contain 24,575–30,000 I/Q samples. The short-observation data are drawn from EM-134K and the EM-Bench perception subset.

Subset	Composition	#QA examples
longEM-Bench: 11 tasks, 300 examples per task, 3,300 in total		
Identification	CJR and RJR (communication/radar jamming recognition); PI (protocol/waveform identification)	900
Parameter estimation	Communication bandwidth; radar bandwidth, PRF, pulse width, duty cycle, and number of pulses	1,800
Composite entities	Communication and jamming composite-entity understanding: type, interval, and parameters or affected targets	600
Training and validation data		
Observation training	longEM-IQA; 11 tasks; chain-of-thought (CoT) trajectories with localization, `zoom_in`, perception, and tool use	27,135
Observation validation	longEM-IQA validation QA pairs	561
Segment training	EM-134K; 1,024-sample I/Q segments for local signal perception	131,432
Segment validation	EM-134K validation QA pairs	2,675
Segment benchmark	EM-Bench perception subset; 10 tasks, 300 examples per task; multiple-choice evaluation	3,000
Total		**168,103**

Figure 2 illustrates construction, while Figures D1–D5 in Appendix D show trajectories that directly perceive attributes after localization and cropping.

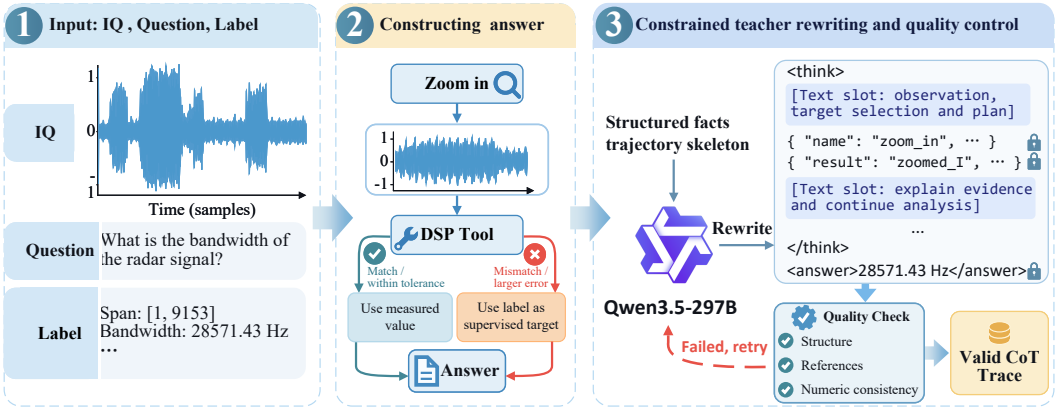


Figure 2: Construction pipeline for long-observation agentic CoT trajectories. A teacher runtime uses simulated observations and ground-truth event intervals to localize and crop the target signal. In the measurement branch shown here, DSP tools provide parameter estimates, which are checked against the simulation labels before constructing the answer. Constrained teacher rewriting produces the final reasoning trace. The corpus also includes direct-perception trajectories, as illustrated in Appendix D.

4 METHOD

Target-segment supervision can improve protocol, modulation, and radar waveform recognition and physical attribute estimation, but we find that joint training with full I/Q observations improves local recognition at the expense of observation-level localization. We attribute this trade-off to depth-dependent requirements: shallow layers must preserve temporal occupancy, boundaries, and relative position for localization, whereas target-segment supervision emphasizes local content; deep layers map shared electromagnetic (EM) representations into distinct decision spaces. We analyze these requirements across the large language model (LLM), introduce routed specialization in shallow and deep layers, and optimize tool interaction policies through agentic reinforcement learning.

4.1 Observation-Level Localization Relies on Shallow Temporal Layout Representations

Which LLM layers primarily retain the information essential to observation-level signal localization?

Signals of the same type can occupy arbitrary temporal positions, so localization requires waveform recognition, boundaries, and relative temporal layout. Pre-cropped target segments do not require recovery of their original positions. To isolate temporal effects, we shift only the target's position while holding its waveform, duration, protocol or radar waveform type, and channel conditions fixed. Using the same localization query for each original/shifted pair, we compare layer-wise EM-token hidden states and define temporal layout sensitivity at layer ℓ as

$$\mathcal{S}_{\ell}^{\text{temp}} = \frac{1}{N_{\text{pair}}} \sum_{i=1}^{N_{\text{pair}}} \frac{1}{T_i} \sum_{t=1}^{T_i} \left[1 - \cos\left(\mathbf{h}_{\ell,t}^{(i,\text{org})}, \mathbf{h}_{\ell,t}^{(i,\text{shift})} \right) \right], \tag{1}$$

where T_i is the number of EM tokens included in the comparison. A larger $\mathcal{S}_{\ell}^{\text{temp}}$ indicates a stronger change in EM representations at that layer in response to a change in temporal layout.

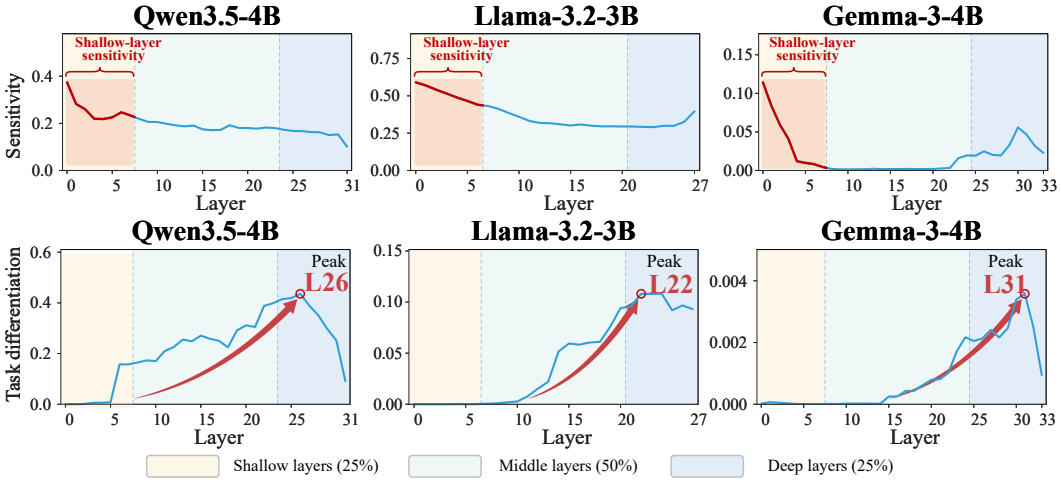


Figure 3: Layer-wise analysis of temporal sensitivity and task divergence for Qwen3.5-4B, Llama-3.2-3B, and Gemma-3-4B (left to right). **Top row:** sensitivity of EM token representations to controlled temporal shifts of the target signal; responses are strongest in the shallow layers and generally weaker in the intermediate layers. **Bottom row:** divergence between localization and attribute-query representations; divergence grows toward the deeper layers and peaks at layers 26, 22, and 31, respectively.

Across all three backbones, temporal-shift responses are strong in early layers, generally weaken in intermediate layers, and increase somewhat in deep layers (Figure 3, top). This indicates strong early encoding of temporal occupancy and boundaries, motivating shallow temporal adaptation because target-segment supervision does not directly constrain global temporal layout.

4.2 Localization and Attribute Understanding Form Distinct Decision Representations in Deep Layers

Do observation-level signal localization and attribute understanding develop distinct decision representations in the deep LLM layers?

We pair localization and attribute queries for the same I/Q observation and target entity, varying only the decision objective. Let $\mathbf{z}_{\ell,\text{loc}}^{(i)}$ and $\mathbf{z}_{\ell,\text{attr}}^{(i,a)}$ denote their layer-ℓ representations, with a indexing

attribute types. Task divergence is

$$\mathcal{D}_{\ell}^{\text{task}} = \frac{1}{N_{\text{pair}} N_{\text{attr}}} \sum_{i=1}^{N_{\text{pair}}} \sum_{a=1}^{N_{\text{attr}}} \left[1 - \cos\left(\mathbf{z}_{\ell,\text{loc}}^{(i)}, \mathbf{z}_{\ell,\text{attr}}^{(i,a)} \right) \right]. \tag{2}$$

Task divergence increases toward deeper layers, peaking at layers 26, 22, and 31 for Qwen3.5-4B, Llama-3.2-3B, and Gemma-3-4B (Gemma Team, 2025), respectively (Figure 3, bottom). This supports shared intermediate EM representations with distinct mappings into position/boundary and category/attribute spaces, motivating deep decision-specific adaptation alongside shallow temporal specialization.

4.3 Depth-Aware Routed Capability Specialization

Motivated by Figure 3, we add routable low-rank branches to shallow layers $\mathcal{L}_{\text{sh}}$ for temporal specialization and deep layers $\mathcal{L}_{\text{dp}}$ for decision specialization, while sharing intermediate EM representations. This design relates to routed LoRA experts and layer-wise expert allocation (Li et al., 2024; Gao et al., 2024); Appendix F discusses these connections.

Model architecture. Figure 4 presents the overall framework. Given a two-channel I/Q observation $\mathbf{X} \in \mathbb{R}^{T \times 2}$, the EM Encoder encodes it into temporal EM features, which a multilayer perceptron (MLP) projector then maps into the LLM embedding space:

$$\mathbf{Z}_X = P_{\psi}(E_{\phi}(\mathbf{X})). \tag{3}$$

The projected EM embeddings replace the corresponding placeholders in the input sequence and, together with the query, metadata, and previous tool feedback, form the multimodal context. Full observations, target segments, and local I/Q segments returned by tools share the same encoding pathway; the EM Agentic Runtime executes tools and updates the context. Appendix E specifies the tool interfaces, signal representation, and representative physical estimators.

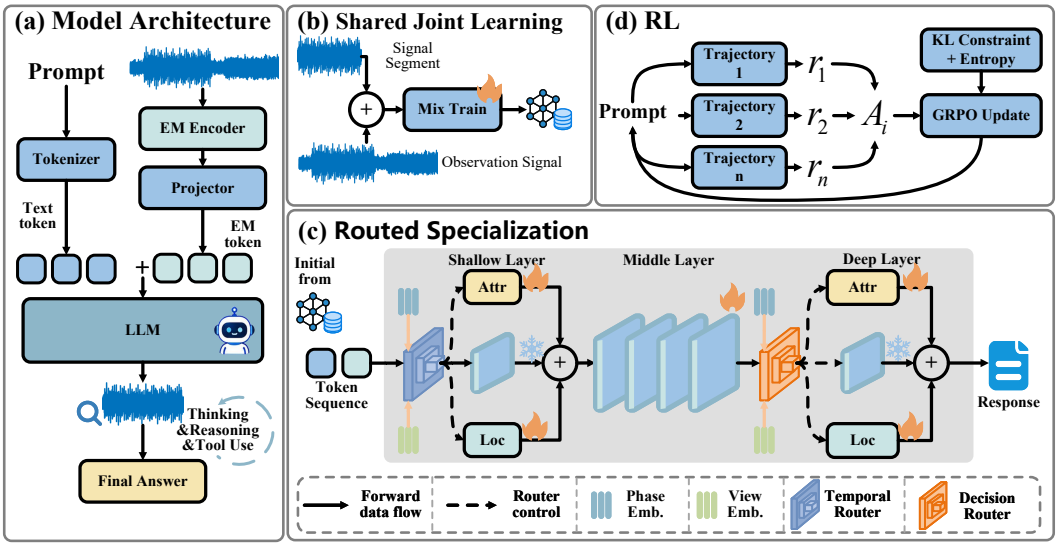


Figure 4: The EM-Lens framework. (a) The multimodal perception and reasoning backbone. (b) Shared joint learning from full observations and target segments. (c) The shallow Temporal Router, shared intermediate layers, and deep Decision Router. (d) Agentic reinforcement learning with actual tool interactions and verifiable rewards.

4.3.1 Supervised Capability Learning

Shared joint learning and routed specialization. We first perform supervised fine-tuning (SFT) on full-observation trajectories for localization, reasoning, tool calls, and final answers, mixed with target-segment question–answer data for fine-grained attributes. From this checkpoint, we add

localization and attribute low-rank adaptation (LoRA) (Hu et al., 2022) branches to designated projections in $\mathcal{L}_{\mathrm{sh}}$ and $\mathcal{L}_{\mathrm{dp}}$, then continue training on the mixture. For projection $W_{\ell,m}$ in layer ℓ, the update for capability $c \in \{\mathrm{loc}, \mathrm{attr}\}$ and adapted output are

$$\Delta W^c_{\ell,m} = \frac{\alpha_{\mathrm{LoRA}}}{r_{\mathrm{LoRA}}} B^c_{\ell,m} A^c_{\ell,m}, \qquad \mathbf{y}_{\ell,m,t} = W_{\ell,m}\mathbf{x}_{\ell,m,t} + \sum_c g^c_{\ell,t} \Delta W^c_{\ell,m} \mathbf{x}_{\ell,m,t}. \tag{4}$$

Causal routing state. Both routers depend only on information available at the current position:

$$\mathbf{u}_{\ell,t} = \mathrm{Concat}[\mathrm{LN}(\mathbf{h}_{\ell,t}), E_{\mathrm{phase}}(\sigma_t), E_{\mathrm{view}}(\nu_t)], \tag{5}$$

where σ_t and ν_t denote the interaction phase and signal view type known up to the current position, respectively. The same causal state construction is thus used during training and inference.

Shallow Temporal Router. For $\ell \in \mathcal{L}_{\mathrm{sh}}$, the Temporal Router predicts weights for the localization and attribute branches at EM token positions:

$$[\omega^{\mathrm{loc}}_{\ell,t}, \omega^{\mathrm{attr}}_{\ell,t}] = \mathrm{softmax}\big(R^{\mathrm{temp}}_\ell(\mathbf{u}_{\ell,t})\big), \qquad g^c_{\ell,t} = m^{\mathrm{EM}}_t \omega^c_{\ell,t}. \tag{6}$$

Here, m^{EM}_t is an EM-token indicator, so shallow updates act only on EM representations, while text tokens propagate through the shared backbone.

Deep Decision Router. For $\ell \in \mathcal{L}_{\mathrm{dp}}$, the Decision Router predicts three states: localization, attribute, and shared:

$$[\omega^{\mathrm{loc}}_{\ell,t}, \omega^{\mathrm{attr}}_{\ell,t}, \omega^{\mathrm{share}}_{\ell,t}] = \mathrm{softmax}\big(R^{\mathrm{dec}}_\ell(\mathbf{u}_{\ell,t})\big), \quad g^c_{\ell,t} = m^{\mathrm{gen}}_t \omega^c_{\ell,t},\ c \in \{\mathrm{loc}, \mathrm{attr}\}. \tag{7}$$

Here, $m^{\mathrm{gen}}_t = 1$ only at model generation positions, and higher $\omega^{\mathrm{share}}_{\ell,t}$ favors the shared backbone over capability-specific corrections. Unlike shallow signal representations, generated text has relatively explicit functional semantics, enabling lightweight Decision Router supervision. Positions are labeled localization, attribute, or shared when the next generated token is unambiguously classifiable:

$$c_{n,t+1} \in \{\mathrm{loc}, \mathrm{attr}, \mathrm{share}\}, \tag{8}$$

where $c_{n,t+1}$ labels the next-token prediction from position t in sample n: loc covers localization intervals, boundaries, and related structured fields; attr covers protocols, modulation schemes, waveforms, and physical attributes; and share covers general reasoning, formatting, and natural language generation. For the set $\mathcal{I}_{\mathrm{route}}$ of generation positions with valid labels, the auxiliary Decision Router loss is

$$\mathcal{L}_{\mathrm{route}} = -\frac{1}{|\mathcal{L}_{\mathrm{dp}}||\mathcal{I}_{\mathrm{route}}|} \sum_{\ell \in \mathcal{L}_{\mathrm{dp}}} \sum_{(n,t) \in \mathcal{I}_{\mathrm{route}}} \log \omega^{c_{n,t+1}}_{\ell,n,t}, \tag{9}$$

where $\mathcal{L}_{\mathrm{dp}}$ is the set of deep layers equipped with a Decision Router, and $\omega^{c_{n,t+1}}_{\ell,n,t}$ is the probability that the router at layer ℓ assigns to the ground-truth functional class $c_{n,t+1}$. The final supervised learning objective is

$$\mathcal{L}_{\mathrm{sup}} = \mathcal{L}_{\mathrm{SFT}} + \lambda_{\mathrm{route}} \mathcal{L}_{\mathrm{route}}, \tag{10}$$

where λ_{route} controls the weight of the auxiliary routing supervision.

4.4 Agentic Reinforcement Learning

Starting from the supervised policy, we optimize multi-turn analysis through actual EM Agentic Runtime interactions using group relative policy optimization (GRPO) (Shao et al., 2024). Tool feedback re-enters the context, and returned I/Q segments reuse the EM encoding pathway. Only model-generated reasoning, tool-call arguments, and final answers contribute to the policy loss; environment-returned content is masked out (Jin et al., 2025). Verifiable rewards combine task correctness, answer formatting, and tool execution outcomes. We normalize rewards within each query's sampled trajectory group and calibrate advantages across tasks before policy updates. Training retains the parameter-update scheme of supervised routed specialization while additionally freezing the EM Encoder. Appendix A.3 details trajectory sampling, reward computation, advantage calibration, and the regularized policy objective.

5 EXPERIMENTS

We evaluate long-observation understanding and short-observation perception, then assess routed specialization and reinforcement learning.

5.1 EXPERIMENTAL SETUP

Model. EM-Lens combines Qwen3.5-4B with an EM Encoder and a two-layer MLP projecting I/Q features into the language embedding space, a connector pattern also used in visual instruction tuning (Liu et al., 2024). The encoder has 12 Mamba2 layers (Dao & Gu, 2024), a hidden dimension of 768, and an 8-sample patch size. Self-supervised pretraining uses approximately 56 million unlabeled EM examples from 13 public and self-collected datasets (Appendix A.1). To accommodate long I/Q inputs, we increase the patch size of MERLIN (Shen et al., 2026) from 8 to 48 samples and evaluate this adapted configuration, MERLIN (patch 48), on the same 3,300 longEM-Bench examples. The SFT (w/o CoT) baseline is trained on longEM-IQA for two epochs, with CoT traces and tool references removed from the training data and supervision restricted to final answers.

Training. Long-observation SFT uses longEM-IQA; mixed-data SFT additionally uses EM-134K (Table 1). EM-Lens undergoes one epoch of mixed-data SFT with shared parameters, followed by one epoch with soft-routed dual-branch LoRA in shallow (L0–L3) and deep (L28–L31) layers. Group relative policy optimization (GRPO) then optimizes tool-interaction trajectories. Appendix A gives implementation details.

Evaluation. We evaluate on longEM-Bench and the EM-Bench perception subset (Table 1). Long-observation classification uses accuracy, continuous parameter estimation uses relative or absolute error against ground truth, and composite-entity understanding uses entity matching, temporal overlap, and parameter consistency. All short-observation tasks use multiple-choice accuracy; Appendix B specifies the scoring rules. All scores are reported to one decimal place; macro-averages are computed before rounding.

5.2 MAIN RESULTS

Table 2: Long-observation understanding on longEM-Bench (0–100, ↑). Avg. is the macro-average over 11 tasks; bold marks the best result per column. C and R denote communication and radar, respectively. BW, DC, NP, PRF, and PW denote bandwidth, duty cycle, number of pulses, pulse repetition frequency, and pulse width. C-Ent and J-Ent denote communication and jamming composite entities, respectively.

Method	Identification			Parameter estimation						Composite		Avg.
	CJR	PI	RJR	C-BW	R-BW	DC	NP	PRF	PW	C-Ent	J-Ent	
GPT-5.6-Sol	25.3	28.7	8.7	29.6	17.1	39.0	39.5	20.5	43.6	25.3	42.1	29.0
Gemini-3.8-Flash	14.7	17.3	2.7	10.3	9.2	27.6	36.1	9.6	42.4	43.6	40.8	23.1
Claude-Sonnet-5	0.7	6.7	2.7	6.3	6.5	17.6	26.3	0.0	7.3	23.0	42.9	12.7
DeepSeek-v4-Pro	5.3	4.0	1.3	9.2	2.7	2.9	20.1	0.0	15.9	30.0	37.5	11.7
MERLIN (patch 48)	5.7	8.7	13.0	0.3	1.0	7.3	9.3	8.8	0.0	0.0	0.0	4.9
SFT (w/o CoT)	37.3	33.3	31.3	31.1	8.7	36.3	23.3	27.7	13.8	66.6	53.6	33.0
EM-Lens	**54.0**	**72.3**	**52.3**	**73.4**	**78.3**	**84.0**	**66.3**	**70.3**	**83.2**	**84.3**	**65.6**	**71.3**

Observation understanding. On longEM-Bench, EM-Lens scores 71.3 on average and leads all listed baselines on every task (Table 2). GPT-5.6-Sol, the strongest general-purpose baseline by average score, achieves 29.0. Radar bandwidth and PRF scores reach 78.3 and 70.3, exceeding their strongest general-purpose baselines by approximately 61.2 and 49.8 points, respectively.

Segment perception. After extension to long observations, EM-Lens achieves 71.7% average accuracy, close to MERLIN (Stage 1) at 71.8% (Table 3). Although Qwen3-VL-4B-Instruct averages

Table 3: Short-observation perception on the ten-task perception subset of EM-Bench (Shen et al., 2026). All metrics are accuracies (%, ↑); bold marks the best result per column. PE denotes parameter estimation; MOD and SD denote modulation recognition and jamming segment detection, respectively.

Method	MOD	PE (Radar)					PI	Jamming		SD	Avg.
		BW	DC	NP	PRF	PW		CJR	RJR		
GPT-5.6-Sol	19.3	19.0	58.3	62.0	16.4	11.1	33.5	21.1	31.3	54.9	32.7
Claude-4-Sonnet	30.0	24.0	38.0	52.0	25.2	31.0	29.7	25.7	30.7	44.3	33.1
DeepSeek-v3.2-exp	18.0	9.3	9.7	19.7	13.7	17.0	8.3	5.7	11.0	18.3	13.1
Gemini-2.5-Pro	24.0	8.7	50.7	52.3	17.2	15.6	29.7	24.7	31.0	45.3	29.9
Qwen3-VL-4B-Instruct	**45.9**	**84.3**	55.0	86.8	**74.5**	56.1	89.0	76.2	81.5	**80.0**	**72.9**
MERLIN (Stage 1)	39.0	75.5	63.1	90.5	69.3	62.2	85.3	82.3	**82.0**	68.7	71.8
EM-Lens	33.3	72.0	**63.7**	**91.3**	68.0	**74.0**	**90.3**	**84.3**	80.3	59.3	71.7

72.9% in the arXiv v2 report of MERLIN (Shen et al., 2026), EM-Lens is more accurate on duty cycle, number of pulses, pulse width, PI, and CJR.

5.3 Ablation Study

Table 4: Routed specialization and RL ablations. Mix uses conventional mixed-data SFT; EM-Lens (SFT) adds routed specialization without RL; EM-Lens (+RL) applies GRPO to that checkpoint. longEM-Bench Avg. is the macro-average across 11 tasks (0–100); EM-Bench Avg. is mean accuracy across 10 tasks (%).

Model	Routed specialization	RL	longEM-Bench Avg.	EM-Bench Avg.
Mix (SFT)	×	×	65.9	62.4
EM-Lens (SFT)	✓	×	69.5	**71.8**
EM-Lens (+RL)	✓	✓	**71.3**	71.7

We evaluate routed specialization and subsequent reinforcement learning under mixed-data training. Relative to conventional mixed-data SFT, EM-Lens increases the long-observation average from 65.9 to 69.5 (+3.6) and short-observation accuracy from 62.4% to 71.8% (+9.4 percentage points), supporting the overall routed specialization design (Table 4). GRPO further raises the long-observation average from 69.5 to 71.3 (+1.8), while short-observation accuracy decreases slightly from 71.8% to 71.7%. Appendix C gives task-wise results.

AI use statement

During the preparation of this work, we used generative AI tools (large language models) for two tasks: (i) language polishing of the manuscript; (ii) writing and debugging portions of the data-processing, simulation, and experiment code; and (iii) drafting the scripts that configure and drive our controllable simulation pipeline and organize the generated samples. Generative AI was not used to directly synthesize, select, or label electromagnetic I/Q samples—all I/Q data were produced from explicit physical and protocol parameters by our own simulation pipeline—nor to conceive the research idea, derive scientific conclusions, or create the figures and tables. All AI-assisted text, code, and pipeline components, as well as the resulting data and labels, were reviewed and verified by the authors, who take full responsibility for the final work.

References

Shuai Chen, Yong Zu, Zhixi Feng, Shuyuan Yang, and Mengchang Li. RadioLLM: Introducing large language model into cognitive radio via hybrid prompt and token reprogrammings. *arXiv preprint arXiv:2501.17888*, 2025. doi: 10.48550/arXiv.2501.17888.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks. *Transactions on Machine Learning Research*, 2023.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. In *International Conference on Machine Learning*, volume 80, pp. 794–803, 2018.

James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. doi: 10.1090/S0025-5718-1965-0178586-1.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. In *International Conference on Machine Learning*, volume 235, pp. 10041–10071, 2024.

O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su. Survey of automatic modulation classification techniques: Classical approaches and new trends. *IET Communications*, 1(2):137–156, 2007. doi: 10.1049/iet-com:20050176.

Chongyang Gao, Kezhen Chen, Jinmeng Rao, Baochen Sun, Ruibo Liu, Daiyi Peng, Yawen Zhang, Xiaoyuan Guo, Jie Yang, and VS Subrahmanian. Higher Layers Need More LoRA Experts. *arXiv preprint arXiv:2402.08562*, 2024. doi: 10.48550/arXiv.2402.08562.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 10764–10799, 2023.

Gemma Team. Gemma 3 Technical Report. *arXiv preprint arXiv:2503.19786*, 2025. doi: 10.48550/arXiv.2503.19786.

Tanmay Gupta and Aniruddha Kembhavi. Visual Programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Zehua Han, Jing Xiao, Yiqi Duan, Mengyu Xiang, Yuheng Ji, Xiaolong Zheng, Chenghanyu Zhang, Zhendong She, Junyu Shen, Dingwei Tan, Shichu Sun, Zhou Cong, Mingxuan Liu, Fengxiang Wang, Jinping Sun, and Yangang Sun. PReD: An LLM-based foundation multimodal model for electromagnetic perception, recognition, and decision. *arXiv preprint arXiv:2603.28183*, 2026. doi: 10.48550/arXiv.2603.28183.

Samer Hanna, Samurdhi Karunaratne, and Danijela Cabric. WiSig: A large-scale WiFi signal dataset for receiver and channel agnostic RF fingerprinting. *IEEE Access*, 10:22808–22818, 2022. doi: 10.1109/ACCESS.2022.3154790.

Simon Haykin. Cognitive radio: Brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, 23(2):201–220, 2005. doi: 10.1109/JSAC.2004.839380.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. In *International Conference on Machine Learning*, volume 97, pp. 2790–2799, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. Visual Sketchpad: Sketching as a Visual Chain of Thought for Multimodal Language Models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 139348–139379, 2024. doi: 10.52202/079017-4423.

Zi Huang, Akila Pemasiri, Simon Denman, Clinton Fookes, and Terrence Martin. Multi-task learning for radar signal characterisation. In *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pp. 1–5, 2023. doi: 10.1109/ICASSPW59220.2023.10193318.

Yuxuan Jiang and Francis Ferraro. SCRIBE: Structured mid-level supervision for tool-using language models. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2026. Accepted to EMNLP 2026 (Main).

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Ö. Arık, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. In *Proceedings of the Second Conference on Language Modeling*, 2025.

Merima Kulin, Tarik Kazaz, Ingrid Moerman, and Eli De Poorter. End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications. *IEEE Access*, 6:18484–18501, 2018. doi: 10.1109/ACCESS.2018.2818794.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. MixLoRA: Enhancing Large Language Models Fine-Tuning with LoRA-based Mixture of Experts. *arXiv preprint arXiv:2404.15159*, 2024. doi: 10.48550/arXiv.2404.15159.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 19730–19742, 2023.

Chao Liu, Yu Wang, Shiwen Mao, Dusit Niyato, Xianbin Wang, and Guan Gui. Spectrumllm: Large language models for next-generation spectrum prediction. *IEEE Wireless Communications*, 33(2): 193–199, 2026. doi: 10.1109/MWC.2025.3596934.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved Baselines with Visual Instruction Tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. doi: 10.48550/arXiv.1711.05101.

Luqing Luo, Wenjin Gui, Yunfei Liu, Ziyue Zhang, Yunxi Zhang, Fengxiang Wang, Zonghao Guo, Zizhi Ma, Xinzhu Liu, Hanxiang He, Jinhai Li, Xin Qiu, Wupeng Xie, and Yangang Sun. EMind: A foundation model for multi-task electromagnetic signals understanding. *arXiv preprint arXiv:2508.18785*, 2025a. doi: 10.48550/arXiv.2508.18785.

Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K. Qiu, and Yuqing Yang. Agent lightning: Train ANY AI agents with reinforcement learning. *arXiv preprint arXiv:2508.03680*, 2025b. doi: 10.48550/arXiv.2508.03680.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. Modeling Task Relationships in Multi-task Learning with Multi-gate Mixture-of-Experts. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018.

Timothy J. O'Shea and Jakob Hoydis. An introduction to deep learning for the physical layer. *IEEE Transactions on Cognitive Communications and Networking*, 3(4):563–575, 2017. doi: 10.1109/TCCN.2017.2758370.

Timothy J. O'Shea and Nathan West. Radio machine learning dataset generation with GNU Radio. In *Proceedings of the GNU Radio Conference*, volume 1, 2016.

Timothy J. O'Shea, Johnathan Corgan, and T. Charles Clancy. Convolutional radio modulation recognition networks. In *Engineering Applications of Neural Networks (EANN)*, pp. 213–226. Springer, 2016. doi: 10.1007/978-3-319-44188-7_16.

Timothy J. O'Shea, Nathan West, Matthew Vondal, and T. Charles Clancy. Semi-supervised radio signal identification. In *19th International Conference on Advanced Communication Technology (ICACT)*, pp. 33–38, 2017. doi: 10.23919/ICACT.2017.7890052.

Timothy J. O'Shea, Tamoghna Roy, and T. Charles Clancy. Over-the-air deep learning based radio signal classification. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):168–179, 2018. doi: 10.1109/JSTSP.2018.2797022.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 487–503, 2021. doi: 10.18653/v1/2021.eacl-main.39.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations (ICLR)*, 2024.

Pengrui Quan, Xiaomin Ouyang, Jeya Vikranth Jeyakumar, Ziqi Wang, Yang Xing, and Mani Srivastava. SensorBench: Benchmarking LLMs in coding-based sensor processing. *arXiv preprint arXiv:2410.10741v3*, 2025. doi: 10.48550/arXiv.2410.10741.

Sreeraj Rajendran, Wannes Meert, Domenico Giustiniano, Vincent Lenders, and Sofie Pollin. Deep learning models for wireless signal classification with distributed low-cost spectrum sensors. *IEEE Transactions on Cognitive Communications and Networking*, 4(3):433–445, 2018. doi: 10.1109/TCCN.2018.2835460.

Kunal Sankhe, Mauro Belgiovine, Fan Zhou, Shamnaz Riyaz, Stratis Ioannidis, and Kaushik Chowdhury. ORACLE: Optimized radio classification through convolutional neural networks. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 370–378, 2019. doi: 10.1109/INFOCOM.2019.8737463.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 68539–68551, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017. doi: 10.48550/arXiv.1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024. doi: 10.48550/arXiv.2402.03300.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *arXiv preprint arXiv:1701.06538*, 2017. doi: 10.48550/arXiv.1701.06538.

Junyu Shen, Zhendong She, Chenghanyu Zhang, Yuchuang Sun, Luqing Luo, Dingwei Tan, Zonghao Guo, Bo Guo, Zehua Han, Wupeng Xie, Yaxin Mu, Peng Zhang, Peipei Li, Fengxiang Wang, Yangang Sun, and Maosong Sun. MERLIN: Building Low-SNR robust multimodal LLMs for electromagnetic signals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8653–8663, 2026.

Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: Visual Inference via Python Execution for Reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Kürşat Tekbıyık, Ali Rıza Ekti, Ali Görçin, Güneş Karabulut Kurt, and Cihat Keçeci. Robust and fast automatic modulation classification with CNN under multipath fading channels. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–6, 2020. doi: 10.1109/VTC2020-Spring48590.2020.9128408.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022. doi: 10.52202/068431-1800.

Quan Wei, Siliang Zeng, Chenliang Li, Zhongruo Wang, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. Reinforcing multi-turn reasoning in LLM agents via fine-grained reward structure and credit assignment. *arXiv preprint arXiv:2505.11821v3*, 2026. doi: 10.48550/arXiv.2505.11821.

Nathan E. West and Timothy J. O'Shea. Deep architectures for modulation recognition. In *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–6, 2017. doi: 10.1109/DYSPAN.2017.7920754.

Penghao Wu and Saining Xie. V*: Guided Visual Search as a Core Mechanism in Multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13084–13094, 2024.

Jieting Xiao, Yun Lin, Huizhen Qiu, Rui Ma, Chen Zhong, Dongyang Xu, Xiao Long, Chaoyu Zhang, Qiaobo Hao, Ding Zou, Zhiguo Yang, Yanqin Gao, and Fang Tan. TeleCom-Bench: How far are large language models from industrial telecommunication applications? In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, Qiaobo Chen, Yinyuting Yin, Hao Zhang, Tengfei Shi, Liang Wang, Qiang Fu, Wei Yang, and Lanxiao Huang. Mastering Complex Control in MOBA Games with Deep Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6672–6679, 2020. doi: 10.1609/aaai.v34i04.6144.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113222–113244, 2025. doi: 10.52202/085713-3775.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient Surgery for Multi-Task Learning. In *Advances in Neural Information Processing Systems*, 2020. doi: 10.48550/arXiv.2001.06782.

Tevfik Yücek and Hüseyin Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE Communications Surveys & Tutorials*, 11(1):116–130, 2009. doi: 10.1109/SURV.2009.090109.

Mingxu Zhang, Ying Sun, Yuhan Li, Yang Ji, Dazhong Shen, Ke Zhang, and Shan Huang. EMRB: A multi-level benchmark for evaluating LLM reasoning over raw electromagnetic signals. *arXiv preprint arXiv:2608.24086*, 2026. doi: 10.48550/arXiv.2608.24086.

Fuhui Zhou, Chunyu Liu, Hao Zhang, Wei Wu, Qihui Wu, Tony Q. S. Quek, and Chan-Byoung Chae. SpectrumFM: A foundation model for intelligent spectrum management. *IEEE Journal on Selected Areas in Communications*, 44:4471–4488, 2026. doi: 10.1109/JSAC.2025.3644783.

Hang Zou, Yu Tian, Bohao Wang, Lina Bariah, Samson Lasaulce, Chongwen Huang, and Mérouane Debbah. RF-GPT: Teaching AI to see the wireless world. *arXiv preprint arXiv:2602.14833*, 2026a. doi: 10.48550/arXiv.2602.14833.

Hang Zou, Bohao Wang, Yu Tian, Lina Bariah, Chongwen Huang, Samson Lasaulce, and Mérouane Debbah. Seeing radio: From zero rf priors to explainable modulation recognition with vision language models. In *2026 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2026b. doi: 10.1109/ICCWorkshops63917.2026.11586562.

A Implementation Details

A.1 Basic Implementation and Pretraining of the EM Encoder

Architecture. The EM Encoder takes two-channel I/Q sequences as input, with a patch size of 8 samples. An embedding layer maps the input patches to a 768-dimensional feature space, followed by a temporal modeling backbone comprising 12 Mamba2 layers (Dao & Gu, 2024). Table A1 specifies the interface to the language backbone.

Pretraining data. The pretraining corpus contains approximately 56 million unlabeled electromagnetic signal examples drawn from 13 public and self-collected datasets: HisarMod2019.1 (Tekbıyık et al., 2020), EM-Infer-Radar-v2, LoRa RF, Mono Receiver, Northeastern RF, Panoradio HF, POWDER-RF, EM-RadarParaIQSim, RadarCommDataset, Transmitter Classification, UAV_DJI_M100, WiSig Raw (Hanna et al., 2022), and EM-radar. Signal lengths during pretraining range from 128 to 4,096 samples.

Preprocessing and optimization. Each input I/Q example is normalized by its maximum absolute amplitude. The EM Encoder undergoes self-supervised pretraining for 10 epochs on 8 NVIDIA GPUs using BF16 precision. We use AdamW (Loshchilov & Hutter, 2019) with $\boldsymbol{\beta} = (0.9, 0.95)$ and a weight decay of 0.05. The learning rate follows a linear warmup for the first epoch and cosine decay thereafter, with a gradient clipping threshold of 1.0. These settings apply to pretraining of the EM Encoder; the supervised fine-tuning settings for EM-Lens are reported separately in Table A1.

A.2 Model and Supervised Fine-Tuning

We use the same model to process I/Q observations of different lengths. The outputs of the EM Encoder are projected and fed to the language backbone together with text tokens. New signal views obtained by local cropping reuse the same EM Encoder and projection module. Table A1 summarizes the model architecture and supervised fine-tuning settings.

Table A1: Model architecture and supervised fine-tuning (SFT) configuration. All layer indices are zero-based.

Item	Configuration
Language backbone	Qwen3.5-4B; 32 layers; language hidden dimension 2,560
EM Encoder	12 Mamba2 layers; internal width 768; two input channels (I/Q)
EM Encoder output / projector	3,072 dimensions; two-layer MLP: $3{,}072 \to 4{,}096 \to 2{,}560$; GELU
Signal input / training context	Patch size: 8 samples; long training inputs: 24,575–30,000 I/Q samples (up to 3,750 signal tokens); context length: 32,768 tokens
Long-observation train / validation	27,135 / 561 QA pairs
Short-observation train / validation	131,432 / 2,675 QA pairs
Optimizer	AdamW; $\beta = (0.9, 0.999)$; $\epsilon = 10^{-8}$; weight decay: 0
Learning rates / schedule	Shared parameters: 10^{-5}; LoRA and routers: 10^{-4}; 5% warmup followed by cosine decay
Batch size / precision	8 GPUs; batch size 1 per GPU; 2 gradient accumulation steps; global batch size 16; BF16 computation
Training budgets	Long-observation-only settings: 2 epochs / 3,392 steps; Mix: 2 epochs / 19,822 steps; EM-Lens: 1 shared-parameter epoch followed by 1 additional dual-branch LoRA epoch (9,911 steps in the latter epoch)
LoRA configuration	L0–L3 and L28–L31; two branches per adapted projection; $r = 32$, $\alpha = 64$; no LoRA dropout
Initialization / routing supervision	Random initialization of low-rank A and zero initialization of B; an auxiliary functional classification loss with weight 0.05 is applied only in the deep layers
SFT random seed	3407

Supervision targets. For long-observation trajectories, supervision is applied only to model-generated reasoning text, tool requests, and final answers. Input questions, tool outputs, signal placeholders, and padding positions are excluded from the cross-entropy loss. The control model is supervised only on the final answers from the same QA trajectories, with tool references removed.

A.3 Reinforcement Learning

We train with GRPO (Shao et al., 2024) using 7,200 QA examples without teacher-provided chain-of-thought (CoT) traces. Each batch contains 128 questions, with 8 trajectories sampled per question. We sample with a temperature of 1.1, top-$k = 50$, and top-$p = 0.95$. Each training trajectory allows at most 4 tool calls, with at most 384 tokens generated per turn and a total trajectory length of at most 6,400 tokens. For query q, we sample K trajectories $\mathcal{G}(q) = \{o_1, \ldots, o_K\}$ from the old policy.

Verifiable rewards and within-group advantages. Each trajectory receives a reward from a rule-based scorer consistent with the evaluation procedure:

$$R_i = \mathrm{clip}\big(s_i + \lambda_{\mathrm{full}} b_i + R_i^{\mathrm{fmt}} + R_i^{\mathrm{tool}}, R_{\min}, R_{\max}\big), \tag{A1}$$

where s_i is the task correctness score. Numerical tasks use tolerance-based scores derived from relative or absolute error. Identification tasks use category matching, and tasks for composite-entity understanding combine temporal IoU with attribute-level scores, as specified in Appendix B. The within-group advantage is

$$\mathcal{A}_i = \frac{R_i - \bar{R}_q}{\sigma_q^R + \epsilon}. \tag{A2}$$

This trajectory-level advantage is broadcast to all valid generated tokens in the trajectory; identical rewards within a group induce no relative policy update.

Cross-task advantage calibration. Effective advantage scales may still differ across tasks even after within-group normalization. Let $\sigma_k^{\mathcal{A}}$ denote the standard deviation of advantages for task k, and let $\sigma_{\mathrm{med}}^{\mathcal{A}}$ denote the median of the valid task-specific standard deviations in the current batch. We use

$$\widetilde{\mathcal{A}}_i = \kappa_{k(i)} \mathcal{A}_i, \qquad \kappa_k = \mathrm{clip}\left(\frac{\sigma_{\mathrm{med}}^{\mathcal{A}}}{\sigma_k^{\mathcal{A}} + \epsilon}, 0, \kappa_{\max}\right), \tag{A3}$$

to reduce differences in update magnitudes across tasks while preserving the relative ranking of candidate trajectories within each group.

Policy update. We use asymmetric clipping (Yu et al., 2025) and dual clipping (Ye et al., 2020) in the PPO-style (Schulman et al., 2017) GRPO objective, averaging generated-token contributions within each trajectory and then across trajectories to avoid length-based weighting. We add entropy regularization and a divergence penalty against a frozen reference policy:

$$\mathcal{L}_{\mathrm{RL}} = \mathcal{L}_{\mathrm{PG}} - \lambda_{\mathrm{H}} \mathcal{H}[\pi_\theta] + \lambda_{\mathrm{KL}} \widehat{\mathcal{D}}_{\mathrm{KL}}(\pi_\theta, \pi_{\mathrm{ref}}). \tag{A4}$$

Parameter updates. Reinforcement learning retains the parameter-freezing and update scheme of supervised routed specialization, with the EM Encoder as the only additionally frozen module. This aims to preserve EM perception and depth-wise specialization while tool execution outcomes guide task-level decisions and multi-turn analysis. The learning rates are 5×10^{-6} for shared parameters and 10^{-4} for LoRA and router parameters. Policy optimization uses a clipping interval of $[0.8, 1.28]$, an entropy coefficient of 0.002, and a KL coefficient of 0.1. Rewards are clipped to $[-0.5, 1.3]$.

B Task Scope and Scoring

Evaluation on long observations comprises three identification tasks, six parameter estimation tasks, and two tasks for composite-entity understanding. Each task contains 300 examples, evenly divided between 150 standard questions and 150 referential questions, for a total of 3,300 examples. Evaluation on short observations uses the 10-task perception subset of EM-Bench, with 300 examples per task and 3,000 examples in total, retaining the original jamming segment detection (SD) multiple-choice questions.

Table B1: Scoring rules for tasks involving long observations. Each example receives a score in $[0, 1]$; final results are reported on a 100-point scale.

Task	Scoring rule
CJR, RJR, PI	Match normalized categories. For PI, both the category type and label must match.
Communication / radar bandwidth, PRF, pulse width	Score 1 for relative error $\leq 5\%$ and 0 for relative error $\geq 10\%$; linearly interpolate between these thresholds.
Radar duty cycle	Express duty cycle on a 0–1 scale. Score 1 for absolute error ≤ 0.01 and 0 for absolute error ≥ 0.025; linearly interpolate in between.
Radar number of pulses	Round to an integer first. Let d be the absolute difference from the ground truth. Score 1 for $d \leq 1$ and 0 for $d \geq 6$; otherwise, use $(6 - d)/5$.
Communication composite entity	Average the temporal interval intersection-over-union (IoU), protocol matching score, and bandwidth score.
Jamming composite entity	Average the temporal interval IoU, jamming category score, and affected-target relationship score.

Tasks involving short observations are evaluated using multiple-choice accuracy. The scorer first extracts the final answer and then identifies the option letter or option content. Numerical answers are mapped to the corresponding options according to the scorer's rules.

C TASK-WISE ABLATION RESULTS

Table C1: Per-task ablation scores on long observations (0–100). The last three columns compare Mix and EM-Lens after SFT, and EM-Lens before and after GRPO. Bold indicates the best result in each row among the listed models.

Task	SFT (w/o CoT) supervision	Trajectory + tools	Mix (SFT)	EM-Lens (SFT)	EM-Lens (+RL)
CJR	37.3	53.3	50.0	**56.3**	54.0
Composite-Comm-Entity	66.6	83.1	79.9	73.8	**84.3**
Composite-Jammer-Entity	53.6	63.7	58.3	65.0	**65.6**
PE-Comm-Bandwidth	31.1	**74.6**	66.6	71.7	73.4
PE-Radar-Bandwidth	8.7	73.6	72.8	77.6	**78.3**
PE-Radar-Duty-Cycle	36.3	80.1	82.4	82.0	**84.0**
PE-Radar-Number-of-Pulses	23.3	62.4	61.0	59.9	**66.3**
PE-Radar-PRF	27.7	73.3	65.0	**74.8**	70.3
PE-Radar-Pulse-Width	13.8	77.8	78.5	79.2	**83.2**
PI	33.3	50.3	63.3	72.0	**72.3**
RJR	31.3	42.0	47.3	52.0	**52.3**
Avg.	33.0	66.8	65.9	69.5	**71.3**

The first two settings use identical long-observation QA data, initialization, and two SFT epochs. SFT (w/o CoT) supervises final answers with tool references removed; Trajectory + tools jointly adds CoT supervision and tool interaction. The last three settings use mixed long- and short-observation training data and correspond to Table 4.

Table C2: Per-task short-observation accuracy (%) for conventional mixed-data SFT, EM-Lens after SFT, and EM-Lens after further optimization with GRPO. Bold indicates the best result in each row.

Task	Mix (SFT)	EM-Lens (SFT)	EM-Lens (+RL)
CJR	83.0	**84.3**	**84.3**
MOD	**33.3**	33.0	**33.3**
PE-Radar-Bandwidth	44.0	71.0	**72.0**
PE-Radar-Duty-Cycle	57.7	**67.7**	63.7
PE-Radar-Number-of-Pulses	90.3	**91.7**	91.3
PE-Radar-PRF	57.7	67.0	**68.0**
PE-Radar-Pulse-Width	67.3	73.3	**74.0**
PI	86.3	**90.7**	90.3
RJR	75.3	**80.3**	**80.3**
SD	29.0	**59.3**	**59.3**
Avg.	62.4	**71.8**	71.7

D ILLUSTRATIVE DATASET EXAMPLES

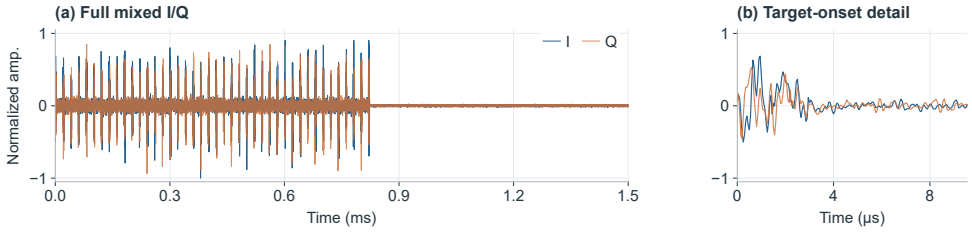
Figures D1–D5 present five illustrative longEM-IQA data examples. Each includes the task metadata, question and answer, full mixed I/Q observation, a short view near the target onset, and the recorded CoT trajectory. These trajectories show target localization and `zoom_in` followed by direct perception of the cropped I/Q segment. Sample intervals use one-based inclusive indices; the displayed onset details show only a small portion of each crop.

Radar pulse repetition frequency

PE-Radar-PRF | Radar + jammer | Target SNR 18 dB | fs 20 MHz | N = 30,000

Question Task: What is the PRF of the radar signal in the IQ observation?

Answer **50000.00 Hz**



CoT

<think>

I first inspect the raw observation and find 2 valid signals in the observation window.
1. the first jammer signal spans samples [1, 16460]
2. the first radar pulse train spans samples [1, 16460]
The question asks: What is the PRF of the radar signal in the IQ observation?
The requested answer needs local signal parameters, so I crop 1 Radar segment.

TOOL CALL

```
{"tool_name": "zoom_in",
 "arguments": {
   "iq_data_ref": "raw",
   "start_sample": 1, "end_sample": 16460,
   "one_based": true,
   "target_entity": "Signal_Radar_Burst_1"
 }}
```

TOOL RESPONSE

```
{"tool_name": "zoom_in",
 "outputs": {
   "signal_ref": "zoomed_1",
   "start_sample": 1, "end_sample": 16460,
   "num_samples": 16460
 }}
```

For the requested parameter, I inspect the relevant zoomed signal directly to perceive PRF.
From <signal_ref:zoomed_1>, I directly perceive PRF = 50000.00 Hz for the first radar pulse train.
Therefore, the final answer is: 50000.00 Hz

</think>

<answer>50000.00 Hz</answer>

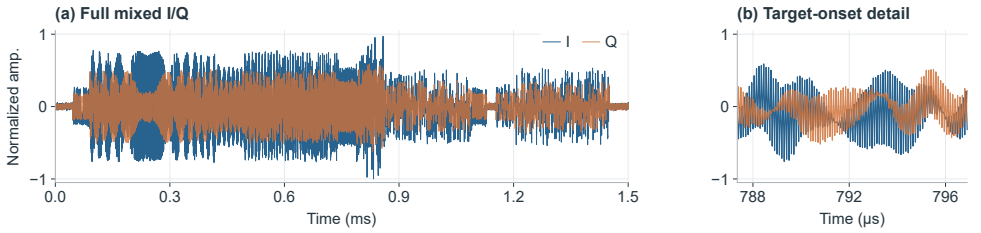
Figure D1: Radar pulse repetition frequency (PE-Radar-PRF). The trajectory selects samples $[1, 16460]$ from a radar–jammer observation, calls `zoom_in`, and directly infers a PRF of $50{,}000$ Hz from the cropped segment.

Protocol identification

PI | Bluetooth + jammer | Target SNR 17 dB | fs 20 MHz | N = 30,000

Question Task: What is the protocol of the second communication signal segment?

Answer **Bluetooth**



CoT

<think>

I inspect the raw observation and find 3 valid signal segments: the first Bluetooth burst signal spans samples [916, 29017], the first jammer signal spans samples [1819, 17224], and the second Bluetooth burst signal spans samples [15748, 28996]. The question asks for the protocol of the second communication signal segment. I filter candidates by the ordinal 'second'. Based on raw temporal evidence, the second Bluetooth burst signal matches this condition. I will use the zoomed signal for this segment to determine the protocol.

TOOL CALL

```
{"tool_name": "zoom_in",
 "arguments": {
   "iq_data_ref": "raw",
   "start_sample": 15748, "end_sample": 28996,
   "one_based": true,
   "target_entity": "Signal_Bluetooth_Burst_2"
 }}
```

TOOL RESPONSE

```
{"tool_name": "zoom_in",
 "outputs": {
   "signal_ref": "zoomed_1",
   "start_sample": 15748, "end_sample": 28996,
   "num_samples": 13249
 }}
```

I inspect the zoomed view <signal_ref:zoomed_1> to identify the communication protocol for the second Bluetooth burst signal. The view confirms the protocol is Bluetooth. Therefore, the final answer is: Bluetooth

</think>

<answer>Bluetooth</answer>

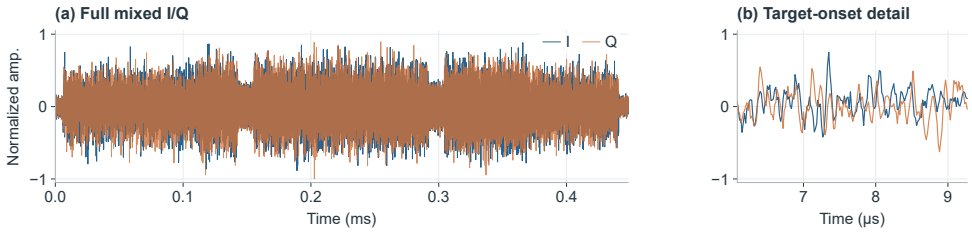
Figure D2: Protocol identification (PI) with an ordinal reference. The question targets the *second* communication segment. The trajectory resolves this reference, crops samples $[15748, 28996]$, and identifies the protocol as Bluetooth.

Communication bandwidth

PE-Comm-Bandwidth | 5G NR + jammer | Target SNR 10 dB | fs 60 MHz | N = 26,894

Question Task: What is the bandwidth of the first communication signal segment?

Answer **19441223.14 Hz**



CoT

<think>

I inspect the raw observation and find 4 valid signals in the window: the first 5G NR burst signal spans samples [366, 8557], the first jammer signal spans samples [6589, 22152], the second 5G NR burst signal spans samples [9335, 17526], and the third 5G NR burst signal spans samples [18261, 26452]. The question asks for the bandwidth of the first communication signal segment. I filter candidates by the ordinal 'first'. Based on raw temporal evidence, the first 5G NR burst signal matches this condition. I now proceed to estimate the bandwidth for this segment.

TOOL CALL

```
{"tool_name": "zoom_in",
 "arguments": {
   "iq_data_ref": "raw",
   "start_sample": 366, "end_sample": 8557,
   "one_based": true,
   "target_entity": "Signal_5GNR_Burst_1"
 }}
```

TOOL RESPONSE

```
{"tool_name": "zoom_in",
 "outputs": {
   "signal_ref": "zoomed_1",
   "start_sample": 366, "end_sample": 8557,
   "num_samples": 8192
 }}
```

For the requested parameter, I inspect the relevant zoomed signal directly to perceive bandwidth. From <signal_ref:zoomed_1>, I directly perceive bandwidth = 19441223.14 Hz for the first 5G NR burst signal. Therefore, the final answer is: 19441223.14 Hz

</think>

<answer>19441223.14 Hz</answer>

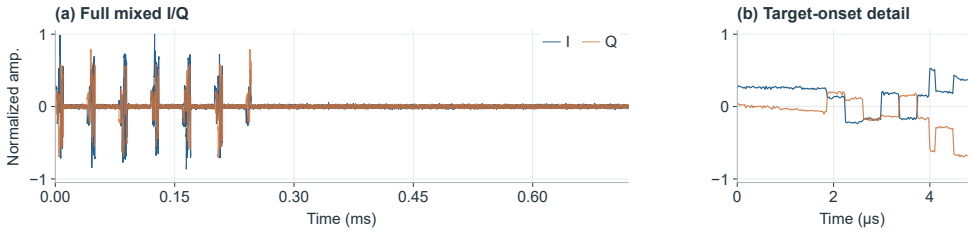
Figure D3: Communication bandwidth estimation (PE-Comm-Bandwidth). The trajectory selects the *first* communication segment in a 5G NR–jammer observation, crops samples $[366, 8557]$, and directly infers a bandwidth of $19{,}441{,}223.14$ Hz.

Radar bandwidth

PE-Radar-Bandwidth | Radar + jammer | Target SNR 10 dB | fs 40 MHz | N = 28,800

Question Task: What is the bandwidth of the radar signal in the IQ observation?

Answer **2301483.15 Hz**



CoT

<think>

I first inspect the raw observation and find 2 valid signals in the observation window.
1. the first jammer signal spans samples [1, 9840]
2. the first radar pulse train spans samples [1, 9840]
The question asks: What is the bandwidth of the radar signal in the IQ observation?
The requested answer needs local signal parameters, so I crop 1 Radar segment.

TOOL CALL

```
{"tool_name": "zoom_in",
 "arguments": {
   "iq_data_ref": "raw",
   "start_sample": 1, "end_sample": 9840,
   "one_based": true,
   "target_entity": "Signal_Radar_Burst_1"
 }}
```

TOOL RESPONSE

```
{"tool_name": "zoom_in",
 "outputs": {
   "signal_ref": "zoomed_1",
   "start_sample": 1, "end_sample": 9840,
   "num_samples": 9840
 }}
```

For the requested parameter, I inspect the relevant zoomed signal directly to perceive bandwidth.
From <signal_ref:zoomed_1>, I directly perceive bandwidth = 2301483.15 Hz for the first radar pulse train.
Therefore, the final answer is: 2301483.15 Hz

</think>

<answer>2301483.15 Hz</answer>

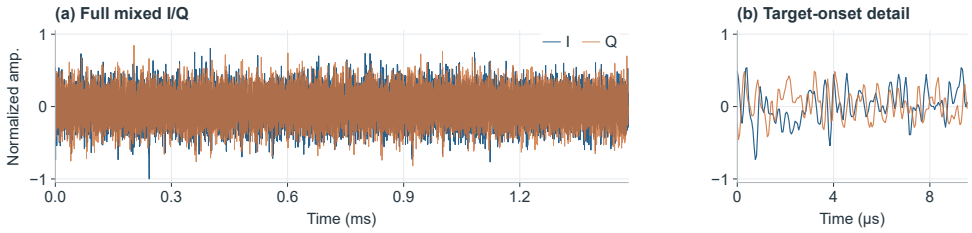
Figure D4: Radar bandwidth estimation (PE-Radar-Bandwidth). The trajectory localizes the radar pulse train to samples $[1, 9840]$ in a radar–jammer observation and directly infers a bandwidth of 2,301,483.15 Hz after `zoom_in`.

Radar pulse count

PE-Radar-Number-of-Pulses | Radar + jammer | Target SNR 14 dB | fs 20 MHz | N = 29,600

Question Task: How many pulses does the radar signal in the IQ observation contain?

Answer **29 pulses**



CoT

<think>

I first inspect the raw observation and find 2 valid signals in the observation window: the first radar pulse train spans samples [1, 22480] and the first jammer signal spans samples [1, 29600]. The question asks: How many pulses does the radar signal in the IQ observation contain? The requested answer needs local signal parameters, so I crop 1 Radar segment.

TOOL CALL

```
{"tool_name": "zoom_in",
 "arguments": {
   "iq_data_ref": "raw",
   "start_sample": 1, "end_sample": 22480,
   "one_based": true,
   "target_entity": "Signal_Radar_Burst_1"
 }}
```

TOOL RESPONSE

```
{"tool_name": "zoom_in",
 "outputs": {
   "signal_ref": "zoomed_1",
   "start_sample": 1, "end_sample": 22480,
   "num_samples": 22480
 }}
```

For the requested parameter, I inspect the relevant zoomed signal directly to perceive number of pulses. From <signal_ref:zoomed_1>, I directly perceive number of pulses = 29 pulses for the first radar pulse train. Therefore, the final answer is: 29 pulses

</think>

<answer>29 pulses</answer>

Figure D5: Radar pulse count (PE-Radar-Number-of-Pulses). The trajectory crops the radar interval $[1, 22480]$ from an observation containing a jammer that spans $[1, 29600]$, then directly infers that the radar segment contains 29 pulses.

E TOOL INTERFACES AND PHYSICAL ESTIMATORS

E.1 TOOL INTERFACES

Each tool is specified by an input contract, a signal-processing operation, and a typed output. The `zoom_in` tool returns a cropped local view of a model-selected interval. The crop is re-encoded through the same EM encoding pathway, allowing subsequent perception and measurements to focus on the region relevant to the query.

Table E1: Toolset for agentic electromagnetic signal analysis

Tool	**Tool role**	**Operation**	**Input**	**Output**
`zoom_in`	Temporal cropping	Crop the model-selected interval	Signal ref, [start, end]	New signal ref, sample count
`estimate_bandwidth`	Spectral measurement	Estimate 6 dB bandwidth	Signal ref, sample rate	Bandwidth (Hz)
`estimate_prf`	Pulse-period measurement	Extract pulse repetition from autocorrelation	Signal ref, sample rate	PRF (Hz)
`estimate_pulse_width`	Pulse-duration estimation	Characterize individual pulse duration	Signal ref, sample rate	Pulse width (s)
`estimate_duty_cycle`	Activity-ratio measurement	Compute active fraction	Signal ref, sample rate	Duty cycle (ratio)
`estimate_num_pulses`	Event counting	Enumerate discrete pulse events	Signal ref, sample rate	Integer count

E.2 SIGNAL REPRESENTATION

Let $s = (s_0, \ldots, s_{L-1}) \in \mathbb{C}^L$ denote the complex baseband (I/Q) signal of length L, sampled at rate f_s and indexed by discrete sample number n. The mathematical notation below uses zero-based sample indices. The example trajectories in Appendix D use one-based inclusive intervals: an interval $[a, b]$ there corresponds to $s_{a-1:b-1}$ here. The signal is partitioned into

$$N = \lceil L/p \rceil \tag{E1}$$

non-overlapping patches of length p. When L is not a multiple of p, the final patch is zero-padded. Define the padded sequence by

$$\tilde{s}_n = \begin{cases} s_n, & 0 \le n < L, \\ 0, & L \le n < Np. \end{cases} \tag{E2}$$

The t-th patch is the contiguous slice

$$s^{(t)} = \tilde{s}_{tp:(t+1)p-1}, \qquad t = 0, 1, \ldots, N-1, \tag{E3}$$

and each complex patch is converted to a real-valued input vector by concatenating its in-phase and quadrature components:

$$u_t = \left[\ \mathrm{Re}\Big(s^{(t)}\Big), \ \mathrm{Im}\Big(s^{(t)}\Big) \ \right] \in \mathbb{R}^{2p}. \tag{E4}$$

E.3 Representative Physical Estimators

The spectral estimators use the DFT; the FFT provides an efficient way to compute this transform (Cooley & Tukey, 1965). The tools implement the following estimators:

$$\hat{f}_c = \arg\max_f \; \big|\, \mathcal{F}\{s\}(f) \,\big|, \tag{E5}$$

$$\hat{B}_{6\,\mathrm{dB}} = f_{\mathrm{high}} - f_{\mathrm{low}}, \tag{E6}$$

$$\widehat{\mathrm{PRF}} = f_s \,/\, \Delta n_{\mathrm{peak}}, \tag{E7}$$

$$\hat{D} = \hat{\tau}_{\mathrm{on}} \,/\, \hat{T}_{\mathrm{period}}. \tag{E8}$$

Notation.

- $s \in \mathbb{C}^L$: complex I/Q signal; L: number of samples; f_s: sampling rate in Hz.
- p: patch length in samples; N: number of patches; $\tilde{s}$: the zero-padded sequence of length Np; $\tilde{s}_{a:b}$: inclusive slice containing samples $a, \ldots, b$. Patch t spans $tp, \ldots, (t+1)p - 1$ (exactly p samples).
- $u_t \in \mathbb{R}^{2p}$: real-valued representation of patch t, stacking real and imaginary parts.
- $\mathcal{F}\{s\}(f)$: discrete Fourier transform (DFT) of s, evaluated at frequency f; $\hat{f}_c$: frequency of the dominant spectral peak.
- $f_{\mathrm{high}}, f_{\mathrm{low}}$: upper and lower edge frequencies at which the magnitude spectrum falls $6\,\mathrm{dB}$ below its peak; $\hat{B}_{6\,\mathrm{dB}}$: resulting 6 dB bandwidth estimate.
- Δn_{peak}: sample lag between adjacent peaks of the autocorrelation; $\widehat{\mathrm{PRF}}$: pulse repetition frequency estimate.
- $\hat{\tau}_{\mathrm{on}}$: estimated pulse-on duration; $\hat{T}_{\mathrm{period}}$: estimated pulse repetition interval; $\hat{D}$: estimated duty cycle.

F Additional Related Work

Signal learning and benchmark scope. Classical modulation recognition uses likelihood-based or feature-based inference (Dobre et al., 2007). Deep physical-layer learning (O'Shea & Hoydis, 2017) extends this direction through learned representations: deep modulation architectures (West & O'Shea, 2017), recurrent amplitude–phase models with variable-length inputs (Rajendran et al., 2018), and end-to-end spectrum identification (Kulin et al., 2018) address complementary aspects of signal classification. These studies motivate signal representation learning without establishing the layer-wise localization and attribute separation measured in this work.

RadioML (O'Shea & West, 2016) and HisarMod (Tekbıyık et al., 2020) provide synthetic modulation data with controlled channel variation. WiSig (Hanna et al., 2022) studies receiver and channel variation in transmitter identification. RadChar jointly studies radar classification and parameter regression (Huang et al., 2023). Their task definitions provide useful context for longEM's identification and parameter-estimation tasks; longEM additionally supplies observation-level analysis trajectories. Dataset provenance for our encoder is listed separately in Appendix A.1.

Multimodal evidence and executable reasoning. BLIP-2 aligns a frozen image encoder with a frozen language model (Li et al., 2023); LLaVA studies visual instruction tuning (Liu et al., 2023), and its improved baseline uses an MLP connector (Liu et al., 2024). These are architectural precedents for connecting perceptual features to a language model, rather than evidence about I/Q perception. ViperGPT and VisProg use executable programs to compose visual modules (Surís et al., 2023; Gupta & Kembhavi, 2023). Visual Sketchpad returns intermediate visual artifacts to the reasoning context (Hu et al., 2024), while V* retrieves local evidence through guided visual search (Wu & Xie, 2024). EM-Lens studies related evidence-acquisition behavior over one-dimensional complex signals with physical measurement tools.

Shared representations and task specialization. GradNorm balances task gradient magnitudes (Chen et al., 2018), and PCGrad mitigates conflicting gradients (Yu et al., 2020). Architectural approaches instead control how computation is shared: sparsely gated experts select subnetworks (Shazeer et al., 2017), and multi-gate mixtures share experts with task-specific gates (Ma et al., 2018). Adapters introduce small task-specific modules (Houlsby et al., 2019); AdapterFusion composes knowledge from separately trained adapters (Pfeiffer et al., 2021). MixLoRA routes among low-rank experts (Li et al., 2024), and MoLA varies expert allocation across layers (Gao et al., 2024). Our design uses soft capability routing at selected shallow and deep layers, conditioned on causal interaction state. The temporal and decision roles assigned to those layers are motivated by the controlled I/Q analyses in Sections 4.1 and 4.2.