

# RF-FilterLLM: Automated RF Filter Design via Domain-Specific Fine-Tuning of LLM

Wenlong Sun, Jian Pang

State Key Laboratory of Radio Frequency Heterogeneous Integration, Shanghai Jiao Tong University,  
No.800, Dongchuan Road, Shanghai, 200240, China  
pangjian@sjtu.edu.cn

**Abstract**—This paper presents RF-FilterLLM, a framework that automates radio-frequency ladder-filter design by fine-tuning a large language model (LLM) on domain-specific data. We construct a bilingual instruction-tuning dataset, RF-Filter-SFT (25,168 samples), covering Chebyshev Type-I lowpass, highpass, and bandpass topologies. Three data-augmentation pipelines—perturbation-induced reflection (PIR), bidirectional performance prediction (BPP), and sensitivity-guided tuning—supply the model with self-diagnostic and reverse-engineering training signals. With QLoRA [1] on a single RTX 4090 (3.83 M trainable parameters, 9.4 h), the resulting model raises filter-order prediction accuracy from 19.5% to 83.4% while maintaining 100% JSON-format compliance and near-zero numerical error. In practice, a closed-loop agent coupling the model with Keysight ADS via ABCD-matrix sensitivity analysis [2] converges within two iterations. Ablation experiments further show that naïve fine-tuning on unaugmented data actually *lowers* accuracy below the untuned baseline, underscoring the necessity of the proposed data pipelines.

**Index Terms**—LLM, RF filter design, domain-specific fine-tuning, instruction tuning, EDA automation, closed-loop agent

## I. INTRODUCTION

Lumped-element RF filters are a fundamental building block in receiver front-ends, transmitters, and frequency-multiplexing networks. Designing such a filter involves (i) computing the minimum order  $N$  that satisfies nonlinear constraints on cutoff frequency  $f_c$ , stopband edge  $f_s$ , passband ripple  $L_r$ , and stopband attenuation  $L_A$ ; (ii) synthesizing prototype element values; and (iii) validating the design through electromagnetic or circuit simulation. Among these, steps (i) and (iii) are inherently iterative and demand considerable microwave-engineering expertise.

This paper presents RF-FilterLLM, a locally deployable framework that fine-tunes an open-source LLM for multi-band Chebyshev ladder-filter design. The main contributions are:

- 1) A bilingual instruction-tuning dataset (RF-Filter-SFT, 25,168 samples across LPF/HPF/BPF) with three task-specific augmentation pipelines and a missing-parameter completion mechanism.
- 2) A QLoRA training recipe that adapts the model on a single consumer GPU in 9.4 h, raising order-prediction accuracy from 19.5% to 83.4%.
- 3) A unified frequency-transformation scheme that enables a single model to handle three filter bands.

- 4) A sensitivity-guided reflective agent that couples the LLM with Keysight ADS through ABCD-matrix perturbation analysis for closed-loop design refinement.

## II. METHODOLOGY

As illustrated in Fig. 1, the framework operates in three stages: (1) dataset construction with augmentation, (2) QLoRA fine-tuning [1], [3] of the backbone LLM implemented via LlamaFactory [4], and (3) closed-loop validation through an EDA-coupled agent.

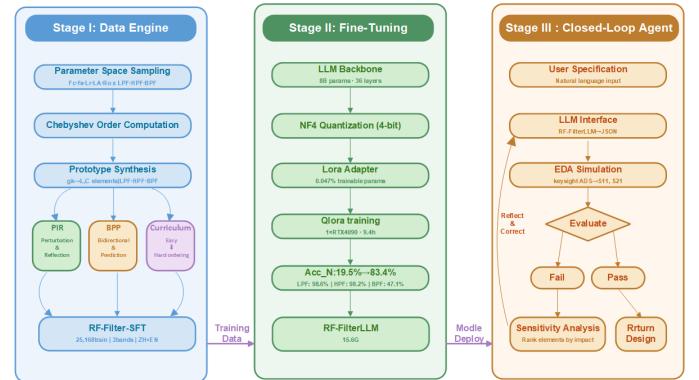


Fig. 1. Three-stage pipeline of the RF-FilterLLM framework: dataset construction with augmentation, QLoRA fine-tuning, and closed-loop ADS validation.

### A. Dataset Construction and Augmentation

The RF-Filter-SFT dataset covers Chebyshev LPF, HPF, and BPF with  $f_c \in [100\text{MHz}, 3\text{GHz}]$ ,  $k_s \in [1.2, 3.0]$ ,  $L_r \in [0.01, 1.0]\text{ dB}$ ,  $L_A \in [20, 60]\text{ dB}$ , and  $R_0 \in \{50, 75, 100\}\Omega$ . The minimum required order is given by:

$$N = \left\lceil \frac{\cosh^{-1} \left( \sqrt{(10^{L_A/10}-1)/(10^{L_r/10}-1)} \right)}{\cosh^{-1}(k_s)} \right\rceil \quad (1)$$

Each sample is formatted as a multi-turn conversation [5] with strict JSON output. A *dynamic intent completion* mechanism triggers follow-up questions whenever critical parameters are missing (3.9% of samples). Table I summarizes the dataset statistics.

Three augmentation pipelines supplement the base data. **Perturbation-induced reflection (PIR)** generates “fail→diagnose→fix” training examples: controlled deviations are injected into correct designs, and a rule-based engine

TABLE I  
RF-FILTER-SFT DATASET STATISTICS

Split	Samples	LPF	HPF	BPF	ZH:EN
Train	25,168	40.3%	25.7%	30.1%	82.6:17.4
Val	1,415	37.8%	28.9%	30.1%	82.5:17.5
Test	1,372	41.9%	29.4%	24.7%	83.3:16.7
Total	<b>27,955</b>				

produces quantitative diagnostic chains. **Bidirectional performance prediction (BPP)** introduces reverse-direction tasks—predicting performance from element values and selecting between candidate designs. **Curriculum-ordered training** [6] arranges samples by a composite difficulty score (order complexity, parameter extremity, and band type) with intra-bucket shuffling to avoid degenerate orderings.

### B. Multi-Band Filter Synthesis

All filters adopt the **Type-I ladder topology** [2], where  $g_1$  is a series inductor,  $g_2$  a shunt capacitor, with subsequent elements alternating in kind. Three bands are realized via standard frequency transformations [2], [7]: direct denormalization (LPF),  $L \leftrightarrow C$  inversion (HPF), and narrowband coupling ( $\delta = \Delta f / f_0$ ) that maps each element to a resonator pair (BPF), expanding the parameter space from 2D to 5D.

By design, the LLM is responsible for predicting the filter order  $N$  and high-level specification parameters, while element values ( $L_k, C_k$ ) are computed deterministically from the Chebyshev prototype through  $g_k$ -to-component formulas [2], [7]. This separation guarantees electrical correctness by construction.

### C. Sensitivity-Guided Reflective Agent

The closed-loop agent couples RF-FilterLLM with Keysight ADS. Whenever a simulation run fails to meet the specifications, element-wise sensitivity is computed via ABCD-matrix perturbation:

$$S_{x_i}^m = \frac{m(x_i(1+\delta)) - m(x_i)}{m(x_i) \cdot \delta}, \quad \delta = 1\% \quad (2)$$

Elements are then ranked by impact score, and the resulting report is fed back to the LLM [8], narrowing the correction search space from  $O(N)$  to  $O(k)$  ( $k \ll N$ ). Algorithm 1 outlines this procedure.

## III. EXPERIMENTS

### A. Results and Ablation

We evaluate RF-FilterLLM on 199 test samples (175 full, 8 follow-up-Q, 16 follow-up-R) using filter-order prediction accuracy ( $Acc_N$ ) as the primary metric. Table II compares three configurations: (1) the untuned backbone, (2) fine-tuning on  $\sim 11k$  uncleaned samples without augmentation (QLoRA-Early), and (3) RF-FilterLLM trained with our full pipeline.

RF-FilterLLM achieves 83.4% overall order accuracy with 100% JSON compliance and 100% type/band accuracy. LPF (98.6%) and HPF (98.2%) reach near-perfect levels, while BPF (47.1%) lags behind—consistent with its larger parameter

---

### Algorithm 1 RF-FilterLLM Closed-Loop Design

---

**Require:** Specification  $S$ , max iterations  $T$ , pass criteria  $E$   
**Ensure:** Validated design  $P^*$

```

1:  $P \leftarrow \text{RF-FilterLLM.generate}(S)$ 
2: for  $t = 1$  to  $T$  do
3:    $R \leftarrow \text{ADS.simulate}(P)$ 
4:   if  $\text{evaluate}(R, E) = \text{PASS}$  then
5:     return  $P, R$ 
6:   end if
7:    $\sigma \leftarrow \text{ABCD\_sensitivity}(P)$  {Eq. (2)}
8:    $P \leftarrow \text{RF-FilterLLM.reflect}(P, R, \sigma)$ 
9: end for
10: return  $P, R$ 

```

---

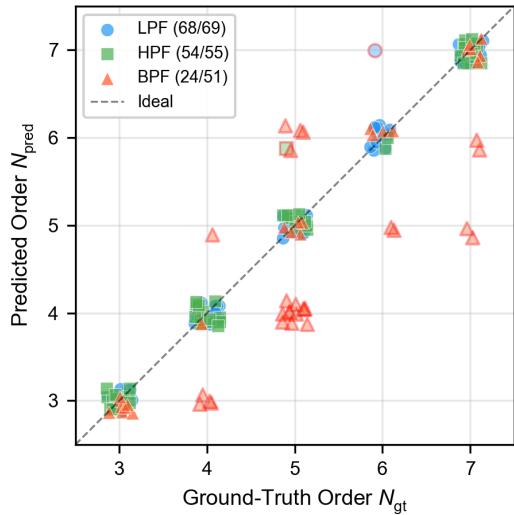


Fig. 2. Per-sample order-prediction results (175 test instances). Points on the dashed diagonal are correct. Red-edged markers indicate errors; BPF triangles show systematic under-prediction.

space. Notably, naïve fine-tuning on the uncleaned dataset *lowers* order accuracy to 12.6%, which is below even the untuned baseline of 19.5%. Incorporating PIR, BPP, and curriculum ordering restores and surpasses the baseline by 63.9 percentage points, confirming that the augmentation pipelines are the primary driver of the performance gain.

Fig. 2 plots per-sample predictions for all 175 test instances: LPF and HPF cluster tightly along the diagonal, whereas BPF exhibits a systematic under-prediction tendency. To illustrate the practical impact, Fig. 3 shows S-parameter responses for a Chebyshev LPF ( $f_c = 1$  GHz,  $f_s = 2$  GHz,  $L_r = 0.1$  dB,  $L_A = 30$  dB,  $N_{\min} = 5$ ). Panel (a) reveals that predicting  $N=4$  yields only 23 dB of stopband rejection—7 dB short of the specification—while  $N=3$  misses by 18 dB; panel (b) shows the corresponding  $S_{11}$  degradation in passband matching.

Fig. 4 demonstrates the closed-loop refinement capability. The reflective agent converged in **2 iterations**: the first simulation pass with  $L_r = 0.5$  dB returned  $S_{11} = -9.6$  dB (fail); the model identified insufficient passband matching as the root cause, reduced  $L_r$  to 0.25 dB, and the second pass achieved  $S_{11} = -12.4$  dB (pass).

TABLE II  
THREE-WAY ABLATION STUDY

Metric	No FT	QLoRA-Early	<b>RF-FilterLLM</b>
Order Acc. (overall)	19.5%	12.6% $\downarrow$	<b>83.4%</b>
– LPF	26.3%	7.9% <b>98.6%</b>	
– HPF	23.3%	10.0% <b>98.2%</b>	
– BPF	0.0%	26.3% <b>47.1%</b>	
Follow-up Ask Rate	N/A	0.0% <b>25.0%</b>	
Multi-turn Order Acc.	N/A	0.0% <b>75.0%</b>	

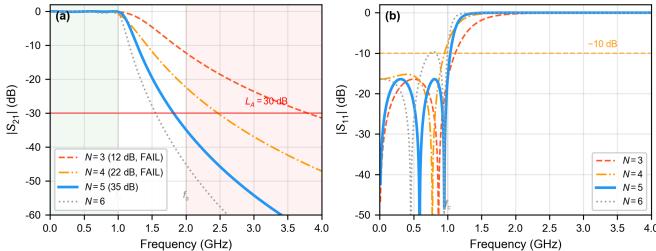


Fig. 3. Impact of order-prediction error on a Chebyshev LPF ( $f_c = 1$  GHz,  $L_A = 30$  dB,  $N_{min} = 5$ ). (a)  $|S_{21}|$ :  $N=4$  falls 7 dB short;  $N=3$  misses by 18 dB. (b)  $|S_{11}|$ : lower orders show worse passband matching.

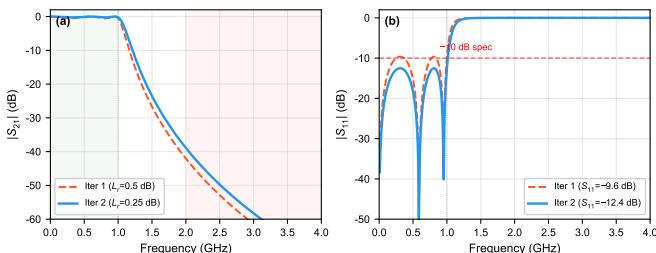


Fig. 4. Closed-loop convergence of the reflective agent on a Chebyshev LPF ( $N=5$ ,  $f_c = 1$  GHz). (a)  $|S_{21}|$  response comparison. (b)  $|S_{11}|$ : reducing  $L_r$  from 0.5 to 0.25 dB improves  $S_{11}$  from  $-9.6$  dB (fail) to  $-12.4$  dB (pass).

### B. Comparison with Related Work

Several recent studies have explored LLMs for EDA tasks. WiseEDA [9] employs GPT-4 with particle-swarm optimization for RF circuit sizing, ChipChat [10] investigates conversational HDL generation, and ChipGPT [11] examines prompt-driven digital design. All three share key limitations: reliance on closed-source cloud APIs, absence of quantitative accuracy metrics, and restriction to single-band or non-analog tasks.

RF-FilterLLM addresses these gaps by fine-tuning an open-source model locally—eliminating API costs and privacy concerns—while delivering a reproducible order-accuracy benchmark (83.4%, a 63.9-point gain over the untuned baseline). The physics-informed closed-loop agent further replaces heuristic trial-and-error with ABCD-matrix sensitivity feedback, converging in two ADS iterations. Table III summarizes the comparison. To the best of our knowledge, RF-FilterLLM is the first fine-tuned, locally deployable LLM covering LPF, HPF, and BPF with quantitative evaluation and an integrated simulation agent.

### IV. CONCLUSION

This paper has presented RF-FilterLLM, a framework that fine-tunes an open-source LLM for automated Chebyshev ladder-filter design. Trained on the RF-Filter-SFT dataset

TABLE III  
COMPARISON WITH EXISTING METHODS

Method	FT?	Bands	Acc <sub>N</sub>	Local	Loop	Cost
WiseEDA [9]	✗	LPF	N/R	✗	PSO	API
ChipChat [10]	✗	N/A	N/A	✗	✗	API
ChipGPT [11]	✗	N/A	N/A	✗	✗	API
Baseline	✗	3	19.5	✓	✗	–
<b>RF-FilterLLM</b>	✓	3	<b>83.4</b>	✓	<b>Refl.</b>	<b>9.4h</b>

(25,168 samples) enriched by PIR, BPP, and curriculum-ordered training [6], the QLoRA-adapted model [1] raises order-prediction accuracy from 19.5% to 83.4% with full output compliance. A sensitivity-guided agent further couples the model with Keysight ADS, converging within two iterations.

Several limitations should be noted. BPF order accuracy (47.1%) remains considerably lower than that of LPF and HPF, owing to the expanded 5D parameter space; targeted oversampling and specialized augmentation strategies will be necessary to close this gap. In addition, the current input schema only supports Chebyshev lowpass, highpass, and bandpass specifications and does not yet generalize to other filter families (e.g., elliptic, inverse Chebyshev) or microstrip implementations. Future work will focus on BPF-targeted data balancing, extension to additional filter topologies, a more flexible natural-language input interface, and transfer across model architectures.

### ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grants 62371296 and 62188102, and by the State Key Laboratory of Radio Frequency Heterogeneous Integration (Independent Scientific Research Program No. 2025017-SJTU).

### REFERENCES

- [1] T. Dettmers *et al.*, “QLoRA: Efficient finetuning of quantized language models,” in *Proc. NeurIPS*, 2023.
- [2] G. L. Matthaei, L. Young, and E. M. T. Jones, *Microwave Filters, Impedance-Matching Networks, and Coupling Structures*. Norwood, MA: Artech House, 1980.
- [3] E. J. Hu *et al.*, “LoRA: Low-rank adaptation of large language models,” in *Proc. ICLR*, 2022.
- [4] Y. Zheng *et al.*, “LlamaFactory: Unified efficient fine-tuning of 100+ language models,” in *Proc. ACL System Demonstrations*, 2024.
- [5] L. Ouyang *et al.*, “Training language models to follow instructions with human feedback,” in *Proc. NeurIPS*, 2022.
- [6] Y. Bengio *et al.*, “Curriculum learning,” in *Proc. ICML*, 2009.
- [7] D. M. Pozar, *Microwave Engineering*, 4th ed. Hoboken, NJ: Wiley, 2012.
- [8] N. Shinn *et al.*, “Reflexion: Language agents with verbal reinforcement learning,” in *Proc. NeurIPS*, 2023.
- [9] Y. Zhang *et al.*, “WiseEDA: An LLM-assisted automated design framework for RF circuits,” *IEEE Trans. Microw. Theory Techn.*, 2024.
- [10] J. Blocklove *et al.*, “Chip-Chat: Challenges and opportunities in conversational hardware design,” in *Proc. ACM/IEEE DAC*, 2023.
- [11] K. Chang *et al.*, “ChipGPT: How far are we from natural language hardware design,” *arXiv:2305.14019*, 2023.