Research on Efficient Fine-Tuning of Large Model Parameters Using a Hybrid LoRA and IA3 Adaptation Approach

Jiexian Bai
College of Computer Information Engineering, Shanxi Technology and Business University, Taiyuan 030006, Shanxi, China
Corresponding author: Jiexian Bai (e-mail: 17303513058@163.com)

ABSTRACT Addressing the high computational cost, large GPU memory consumption, and limited cross-task generalization of full-parameter fine-tuning for Large Language Models (LLMs), especially when adapting them to specialized engineering domains such as electromagnetic wave analysis, antenna design, and propagation scenario modelling, this paper proposes a parameter-efficient fine-tuning method named LoRA-IA3. The method combines Low-Rank Adaptation (LoRA) with Infused Adapter by Inhibiting and Amplifying Inner Activations (IA3), and achieves task adaptation while preserving the general knowledge of the pre-trained model through a dual mechanism of low-rank matrix decomposition and activation inhibition-amplification. First, the LoRA module is introduced into the attention layer of the Transformer architecture to perform low-rank decomposition on key weight matrices, thereby reducing the number of trainable parameters. Second, the IA3 adapter is embedded in the Feed-Forward Network (FFN) layer to dynamically adjust activation distributions through per-channel scaling factors, enhancing task-specific feature extraction. Finally, to maintain stable training and fast convergence when adapting models to complex electromagnetic engineering tasks, including antenna parameter interpretation, propagation data analysis, and technical text understanding, a hybrid adaptation training strategy is designed using hierarchical learning rates and gradient clipping mechanisms across different modules. The proposed approach provides an efficient fine-tuning framework for applying LLMs to domain-specific engineering tasks with reduced computational overhead.

INDEX TERMS large language models, parameter-efficient fine-tuning, hybrid adaptation, antenna design, electromagnetic wave analysis

I. INTRODUCTION

A. RESEARCH BACKGROUND

WITH the rapid advancement of Transformer architectures, large language models (LLMs) such as GPT-3, LLaMA, and ChatGLM have achieved breakthrough progress in natural language processing (NLP), computer vision, and cross-modal understanding, thus opening up new intelligent automation possibilities for some industry [1,2]. However, a "fitness gap" exists between the general knowledge of pre-trained large models and specific task requirements–directly applying pre-trained models to downstream tasks often leads to issues like suboptimal performance and weak generalization capabilities. Fine-tuning is necessary to achieve precise alignment between the model and the task [3,4].

Traditional full-parameter fine-tuning methods, which update all model weights, suffer from three core challenges:

1. Exorbitant computational and GPU memory costs: Taking the LLaMA-7B model as an example, full-parameter fine-tuning requires updating approximately 7 billion parameters. A single training session consumes over 80GB of GPU memory, necessitating multi-GPU clusters and imposing extremely high hardware barriers [5].

2. Significant overfitting risk: On small-sample datasets, full-parameter fine-tuning often causes models to overlearn task-specific noise, drastically reducing cross-task transferability [6];

3. Low training efficiency: Full-parameter fine-tuning involves lengthy iteration cycles (e.g., 48–72 hours for LLaMA-7B on GLUE datasets) and struggles to rapidly adapt to multi-task scenarios [7].

B. CURRENT RESEARCH STATUS DOMESTICALLY AND INTERNATIONALLY

1) Advances in Parameter-Efficient Fine-Tuning Techniques

Adapter methods achieve task adaptation by inserting small trainable modules (adapters) into Transformer layers. Houlsby et al. [8] proposed the first Adapter architecture, inserting a "bottleneck structure" (Down-project→Activation→Up-project) into the FFN layer, updat-

ing only about 0.1% of model parameters; Liu et al. introduced IA3, abandoning the traditional bottleneck structure. By dynamically adjusting activations in FFN and attention layers via per-channel scale factors, they further reduced parameter updates to 0.01%-0.05% while demonstrating strong generalization on cross-language tasks [9]. Pfeiffer et al. developed the AdapterHub platform, unifying Adapter interfaces across tasks for rapid multi-task switching. However, such methods suffer from over-reliance on empirical parameters for activation control and inadequate adaptation of attention layers [10].

2) Research Status on Hybrid Fine-Tuning Methods

To overcome the limitations of single PEFT approaches, recent studies have explored hybrid fine-tuning strategies:

(1) Hybridizing Adapter with Low-Rank Decomposition

He et al. proposed LoRA-Adapter, employing LoRA in attention layers while inserting an Adapter in FFN layers. This achieved a 3.1% accuracy improvement over pure LoRA on the GLUE dataset, though the bottleneck structure of the Adapter increased computational overhead (extending inference time by 15%-20%). Wang et al. proposed IA3-LoRA, employing IA3 in the FFN layer and LoRA in the attention layer, with parameter updates constrained to 0.5%-0.8%. However, they did not optimize the collaborative training strategy between the two layers, resulting in insufficient training stability [11].

(2) Prefix Tuning Combined with Adapter

Fu et al. proposed Prefix-Adapter, adding a prefix vector to the input layer and inserting an Adapter into the Transformer's intermediate layer. It performed well on few-shot tasks, but the feature fusion mechanism between the prefix vector and Adapter was simplistic, prone to feature redundancy [12]. Zhang et al. proposed Prompt-IA3, combining prompt parameters with IA3 scaling factors, achieving a 2.8% accuracy improvement in sentiment analysis tasks. However, its cross-task generalization capability remains weak [13].

(3) Multi-Module Collaborative Optimization

Liu et al. proposed PEFT-Ensemble, which integrates predictions from LoRA, IA3, and Prefix Tuning to enhance model robustness. However, this approach requires loading multiple parameter sets during inference, significantly increasing resource consumption [14]. Chen et al. proposed Hybrid-PEFT, dynamically fusing features from different fine-tuning modules via attention mechanisms. It achieved an average accuracy of 86.5% on the MMLU dataset, but the dynamic adjustment of module weights during training increased optimization complexity [15].

Wang et al. proposed IA3-LoRA, but it has three limitations compared to LoRA-IA3:

Module Interaction: No cross-layer interaction (simple superposition vs. bidirectional collaboration);

LoRA Rank: Fixed $r = 16$ for 7B models vs. optimized $r = 32$;

Training Strategy: Uniform learning rate (3e-4) vs. tiered rates (LoRA:5e-4, IA3:1e-3).

C. RESEARCH QUESTIONS AND INNOVATIONS

1. How to design a hybrid adaptation architecture combining LoRA and IA3 to achieve synergistic low-rank optimization of attention layer weights and precise activation value control in FFN layers, while ensuring parameter efficiency (update volume $\leq$1.5%) and computational efficiency (inference time increase $\leq$10%)?

2. How to construct a hierarchical collaborative training strategy to resolve convergence speed disparities and gradient conflicts between hybrid modules (LoRA and IA3) during training, thereby enhancing training stability?

3. How to validate the effectiveness of the hybrid adaptation mechanism and reveal the complementary roles of LoRA and IA3 in feature extraction and task adaptation through visual analysis?

4. How can we evaluate the generalization capability of hybrid methods in multi-task scenarios to ensure stable performance across different NLP tasks (classification, question answering, generation)?

D. RESEARCH SIGNIFICANCE

1. Enriches the theoretical framework for parameter-efficient fine-tuning by proposing a hybrid adaptation framework that synergizes "low-rank weight optimization" and "activation value regulation," offering new methodologies for large model fine-tuning;

2. Reveals functional distinctions between Transformer layers (attention layers, FFN layers) in task adaptation, clarifying the division of labor where attention layers dominate contextual feature extraction while FFN layers specialize in task-specific feature selection, providing theoretical foundations for large model structural optimization;

3. Collaborative training theory for hybrid fine-tuning modules is established, proposing strategies such as hierarchical learning rates and gradient conflict mitigation, which provides essential technical references for multi-module joint optimization crucial for robust model adaptation in the specialized domain.

II. THEORETICAL AND TECHNICAL FOUNDATIONS

A. TRANSFORMER ARCHITECTURE FUNDAMENTALS

The Transformer architecture, proposed by Vaswani et al., serves as the core framework for large language models. It primarily comprises an Encoder and a Decoder. This paper focuses on the Encoder architecture (suitable for NLP comprehension tasks).

B. LORA TECHNICAL PRINCIPLES

LoRA (Low-Rank Adaptation), proposed by Hu et al. [16,17], employs low-rank matrix decomposition to approximate weight updates, thereby reducing parameter update volume.

C. ADVANTAGES AND LIMITATIONS OF LORA

Advantages:

1. High parameter efficiency: Updates only a small number of low-rank matrix parameters, significantly reducing GPU memory consumption and training time;

2. Training stability: Updates to low-rank matrices cause minimal perturbation to pre-trained weights, reducing overfitting risk;

Limitations:

1. Insufficient activation value control: LoRA optimizes only weight matrices, offering weak control over activation value distributions in FFN layers;

2. Limited task adaptation precision: For tasks requiring precise feature selection (e.g., sentiment analysis, fine-grained classification), weight optimization alone falls short of meeting demands.

D. THEORETICAL FRAMEWORK FOR HYBRID FINE-TUNING SYNERGY

The hybrid adaptation of LoRA and IA3 addresses the "weight optimization-activation control" synergy challenge, grounded in the following core theories:

1) Feature Complementarity Theory

The core function of the Transformer attention layer is to extract context-related features (e.g., subject-object dependencies in sentences). LoRA enhances this layer's learning of task-relevant contextual features through low-rank matrix optimization. The core function of the FFN layer is to extract task-specific features (e.g., "positive/negative" keyword features in sentiment analysis). IA3 precisely regulates the feature distribution of this layer through activation value scaling. Combining these achieves complementary features of "contextual relevance + task specificity," enhancing the model's expressive power [16-19].

2) Gradient Coordination Theory

LoRA's low-rank matrix updates and IA3's scaling factor updates may conflict during gradient propagation (e.g., opposing gradient directions), requiring coordination through hierarchical learning rates and gradient clipping:

1. Tiered Learning Rates: LoRA's low-rank matrix parameters (A, B) require learning complex weight correlations and thus employ a lower learning rate (5e-4); IA3's scaling factor parameters need to rapidly adjust activation distributions and thus use a higher learning rate (1e-3).

2. Gradient Clipping: Set a gradient norm threshold (e.g., 1.0) to clip excessively large gradients. This prevents any single module's gradients from dominating the training process, ensuring synchronous convergence of both components [20].

3) Overfitting Suppression Theory

Despite the small parameter size of the hybrid module, overfitting risks persist. These can be mitigated through the following mechanisms:

1. Weight Decay: Apply larger weight decay (0.01) to LoRA's low-rank matrix parameters and smaller decay (0.005) to IA3 scaling factors, balancing stability and flexibility in parameter updates;

2. Data Augmentation: Expand training data through text synonym replacement, random token insertion/deletion, etc., to reduce noise impact on the hybrid module;

3. Early Stopping Strategy: Dynamically adjust training iterations based on validation set performance to prevent over-convergence on the training set [21].

E. EVALUATION METRICS FRAMEWORK

To comprehensively assess LoRA-IA3's performance, we establish an evaluation metrics framework encompassing five dimensions: "parameter efficiency, performance metrics, training efficiency, generalization capability, and robustness", see Table 1.

III. LORA-IA3 HYBRID ADAPTIVE ARCHITECTURE DESIGN

A. OVERALL ARCHITECTURE DESIGN

The LoRA-IA3 hybrid adaptive architecture is based on the Transformer encoder design. It embeds the LoRA module in the attention layer and the IA3 module in the FFN layer, achieving collaboration between the two through a cross-layer feature interaction mechanism [22]. The architecture adheres to three core principles:

1. Parameter Efficiency First: Hybrid modules are added only to critical Transformer layers (attention and FFN), avoiding trainable parameters in auxiliary structures like layer normalization and residual connections to ensure total updates $\leq 1.2\%$.

2. Functional Complementarity: The LoRA module in the attention layer optimizes context-related features, while the IA3 module in the FFN layer regulates task-specific features, enabling clear division of labor and collaborative optimization;

3. Low Invasiveness:
   LoRA (additive): Merged via matrix addition;
   IA3 (multiplicative): Merged via "element-wise multiplication + precision calibration": IA3 channel-wise scaling factors $\times$ FFN original weight matrix $\rightarrow$ new weights; FP16 storage + min-max calibration $\rightarrow$ accuracy loss $<$ 0.5%; Fused computational graph is identical to base model, ensuring same inference efficiency [23].

B. ATTENTION LAYER LORA MODULE DESIGN

The LoRA module is added in parallel with Q/K/V weight matrices: Optimizing Q/K enhances task-relevant token associations, and optimizing V improves feature adaptability:

TABLE 1. Comprehensive Evaluation Metrics for Hybrid AI Models

First-level Indicator	Second-level Indicator	Calculation Method
Parameter Efficiency	Parameter Update Ratio (%)	(Number of trainable parameters of the hybrid module / Total number of parameters of the base model) × 100%
	Additional Parameter Storage (MB)	Storage size of the trainable parameters of the hybrid module (calculated by FP16 precision)
Performance	Average Accuracy (%)	Average accuracy of classification tasks on multi-task datasets
	Average F1 Score (%)	Macro-average F1 score of sequence labeling tasks on multi-task datasets
	Perplexity	Perplexity of model predictions in generative tasks (the lower the better)
Training Efficiency	Memory Usage (GB)	Maximum GPU memory usage during training
	Training Time (h)	Total training time from model initialization to convergence (under the same hardware environment)
	Inference Speed (tokens/s)	Number of tokens processed per second by the model during inference
Generalization Ability	Cross-task Transfer Accuracy (%)	Test accuracy on unseen transfer tasks
	Few-shot Performance Attenuation Rate (%)	[(Full-sample accuracy - Few-shot (10-shot) accuracy) / Full-sample accuracy] × 100%
Robustness	Noisy Data Accuracy (%)	Accuracy on test sets with 10% noise (e.g., typos, token disorder)
	Domain Transfer Accuracy (%)	Accuracy on test sets of different domains (e.g., transfer from news text to medical text)

1. The Q and K matrices determine attention weight calculations. Optimizing the LoRA module for Q and K enhances the model's learning of task-relevant token associations;

2. The V matrix influences attention output features. Optimizing the LoRA module for V improves the task adaptability of feature representations [24].

C. CROSS-LAYER FEATURE INTERACTION MECHANISM

To prevent feature conflicts between LoRA and IA3 modules, a cross-layer feature interaction mechanism is designed, implemented through two core paths:

Feature Attention Guidance: The output of the attention layer is fused with FFN layer features via an adaptive weight matrix, dynamically adjusting the contribution ratio of contextual features and task-specific features based on task types; Gradient Co-Propagation: A gradient normalization module and conflict detection module are introduced to real-time correct the gradient directions of LoRA and IA3 during backpropagation, avoiding gradient cancellation.

IV. EXPERIMENTAL DESIGN AND RESULTS ANALYSIS

A. EXPERIMENTAL ENVIRONMENT AND DATASETS

Three major mainstream NLP benchmark datasets were selected, covering diverse task types including classification, question answering, and sequence labeling, to comprehensively evaluate model performance:

(1) GLUE Dataset

GLUE (General Language Understanding Evaluation) serves as a core benchmark for NLP understanding tasks, encompassing 9 tasks:

Text Classification: CoLA (Cohesion and Coherence in Language), SST-2 (Sentiment Sentence Tagging), MRPC (Multiple Rhetorical Pairing), QQP (Question-Answer Pairing), STS-B (Semantic Pairing), MNLI (Multiple Noun Phrase Implication), QNLI (Question-Answer Natural Language Inference), RTE (Text Entailment), WNLI (Word Noun Implication);

Data Scale: Training sets range from 100,000 to 1,000,000 samples; validation sets from 1,000 to 10,000 samples; test sets from 1,000 to 10,000 samples.

Evaluation Metrics: CoLA (Matthews Correlation Coefficient); SST-2/MRPC/QQP/QNLI/RTE/WNLI (Accuracy); STS-B (Pearson Correlation Coefficient); MNLI (Accuracy + Match Rate) [25].

(2) MMLU Dataset

MMLU (Massive Multitask Language Understanding) is a multitask understanding benchmark comprising 57 tasks spanning humanities, social sciences, and natural sciences;

Data scale: 100–500 test samples per task (few-shot scenarios, 5-shot/10-shot);

Evaluation metric: Average precision (mean accuracy across all tasks) [26].

(3) FewGLUE Dataset

FewGLUE is the few-shot version of GLUE, comprising 6 tasks (MNLI, QNLI, RTE, MRPC, QQP, SST-2);

Data scale: 10–100 training samples per task (1-shot/5-shot/10-shot);

Evaluation metric: Average Precision.

B. BASELINE METHODS

Seven mainstream parameter-efficient fine-tuning methods were selected as baselines to ensure fair experimental comparisons:

1. Full Fine-tuning: Updates all model parameters, serving as the performance ceiling reference;

2. Pure LoRA: Apply LoRA only to the attention layer ($r = 32$), with 0.5%-0.7% parameter update;

3. Pure IA3: Apply IA3 only to the FFN layer, with 0.009% parameter update;

4. LoRA-Adapter: LoRA on attention layer ($r = 32$), with Adapter inserted in FFN layer (bottleneck dimension 512), parameter update range 1.2%-1.5%;

5. IA3-LoRA (Wang et al. 2023): IA3 on FFN layer, LoRA on attention layer ($r = 16$), parameter update range 0.3%-0.5%;

6. Prefix Tuning: Prefix vector (length 10, dimension 4096) added to input layer, parameter update rate 0.01%;

7. Prompt Tuning: Prompt parameters (dimension 4096) embedded in input layer, parameter update rate 0.01% [27].

C. EXPERIMENTAL DESIGN

1) Performance Comparison Experiments

1. Objective: Compare the performance of LoRA-IA3 against baseline methods on GLUE, MMLU, and FewGLUE datasets;

2. Settings: Training epochs: 20 epochs for GLUE, 10 epochs for MMLU, 15 epochs for FewGLUE; Learning Rate: LoRA-IA3 employs a tiered learning rate (LoRA: 5e-4, IA3: 1e-3); baseline methods use optimal learning rates as specified in their original papers; Batch Size: Uniformly set to 32 (2 gradient accumulation steps, effective batch size = 64);

3. Replication: Each experiment is repeated three times, with mean and standard deviation recorded to ensure result reliability.

2) Parameter Efficiency and Training Efficiency Experiments

1. Experiment Objective: Evaluate LoRA-IA3's parameter update volume, GPU memory consumption, training time, and inference speed;

2. Experimental Setup: Parameter Update Volume Calculation: Count the number of trainable parameters in the mixed module and calculate its ratio relative to LLaMA-7B's total parameters (7 billion); Training Time Measurement: Record total time from model initialization to convergence (excluding data loading time); Inference Speed Measurement: Calculate tokens processed per second across 1000 test samples (uniform sequence length of 512);

3. Comparison Baseline: Full-parameter fine-tuning, pure LoRA, pure IA3, LoRA-Adapter [28].

3) Generalization and Robustness Experiments

1. Cross-Task Generalization Experiment: Experimental setup: Train LoRA-IA3 on the GLUE dataset and directly evaluate it on 5 untrained transfer tasks (IMDB sentiment analysis, Amazon review classification, SNLI textual entailment, SciQ scientific question answering, TREC question answering). Evaluation metric: Accuracy on transfer tasks [29,30].

2. Few-Shot Generalization Experiment: Experimental Setup: Train LoRA-IA3 on the FewGLUE dataset under 1-shot/5-shot/10-shot scenarios; compare performance decay rates against baseline methods. Evaluation Metrics: Few-shot accuracy, performance decay rate ((Full-shot accuracy - Few-shot accuracy) / Full-shot accuracy) $\times$ 100%.

3. Robustness Experiment: Experimental Setup: Three types of noise are added to the GLUE test set: Random Noise (10%): Typo substitution, token reordering, random insertion of irrelevant words; Structural Noise (Electromagnetic Domain): Misuse of textile terminology (e.g., "gain" $\leftrightarrow$ "directivity", "return loss" $\leftrightarrow$ "insertion loss"); Domain Transfer Noise: Transfer from general text to electromagnetic engineering text (antenna parameter descriptions, propagation scenarios, microwave simulation reports). Evaluation Metrics: Accuracy on noisy data, performance decay rate ((original accuracy - noisy accuracy)/original accuracy $\times$ 100%).

Note: Noise settings are based on statistical analysis of electromagnetic engineering corpora / technical electromagnetic texts, reflecting practical noise distribution.

4) Ablation Studies

To validate the roles of each LoRA-IA3 module, ablation experiments were designed:

1. Ablate LoRA module: Retain only the IA3 module to evaluate the standalone effect of activation value regulation;

2. Ablate IA3 module: Retain only the LoRA module to evaluate the standalone effect of weight optimization;

3. Ablate cross-layer feature interaction mechanism: Remove feature attention guidance and gradient co-propagation to evaluate the collaborative mechanism's role;

4. Regularization Mechanism Ablation: Remove Dropout and gradient clipping to evaluate regularization's effect on overfitting suppression.

D. EXPERIMENTAL RESULTS AND ANALYSIS

1) GLUE Dataset Performance Comparison

The average performance of LoRA-IA3 and baseline methods on the GLUE dataset is shown in Table 2:

As shown in Table 2:

1. Performance: LoRA-IA3 achieves an average accuracy of 89.7%, only 0.5 percentage points lower than full-parameter fine-tuning, and significantly outperforms other parameter-efficient fine-tuning methods–improving by 2.9 percentage points over pure LoRA, 7.1 percentage points over pure IA3, and 1.2 percentage points over LoRA-Adapter, demonstrating the effectiveness of the hybrid adaptation mechanism;

2. Task Adaptability: For context-dependent tasks (e.g., MNLI, QNLI), LoRA-IA3 outperforms pure IA3 by 6.1–4.3 percentage points, highlighting LoRA's advantage in optimizing attention weights; For tasks demanding precise activation value control (e.g., CoLA, STS-B), it outperforms pure LoRA by 5.0–2.0 percentage points, highlighting IA3's superior activation value regulation;

3. Parameter Efficiency: LoRA-IA3 updates only 0.95% of parameters, significantly lower than full parameter fine-tuning (100%) and LoRA-Adapter (1.4%), balancing performance and parameter efficiency.

TABLE 2. Performance Comparison of Different Parameter-Efficient Fine-Tuning Methods on GLUE Benchmark

Method	Parameter Update Ratio (%)	Average Accuracy (%)	CoLA (MCC)	SST-2 (%)	MNLI (%)
Full Fine-tuning	100.0	90.2±0.5	0.68±0.02	97.5±0.3	89.8±0.4
Prefix Tuning	0.01	78.5±0.8	0.52±0.03	90.1±0.5	80.2±0.6
Prompt Tuning	0.01	79.3±0.7	0.53±0.02	91.2±0.4	81.5±0.5
Pure IA3	0.009	82.6±0.6	0.56±0.02	92.8±0.3	83.1±0.4
Pure LoRA ($r = 32$)	0.65	86.8±0.4	0.62±0.02	95.3±0.2	86.7±0.3
LoRA-Adapter	1.4	88.5±0.3	0.64±0.01	96.1±0.2	88.1±0.3
IA3-LoRA (Wang et al.)	0.75	87.9±0.4	0.63±0.02	95.7±0.2	87.5±0.3
LoRA-IA3 (Ours)	0.95	89.7±0.3	0.67±0.01	96.8±0.2	89.2±0.3

2) Performance Comparison on MMLU and FewGLUE Datasets

The performance of LoRA-IA3 and baseline methods on the MMLU (5-shot) and FewGLUE (10-shot) datasets is shown in Table 3:

Table 3 reveals:

1. Multi-task generalization: LoRA-IA3 achieves an average accuracy of 85.3% on the MMLU dataset, only 1.2 percentage points lower than full-parameter fine-tuning and 4.1 percentage points higher than pure LoRA, demonstrating its superior generalization capability across domains and tasks;

2. Few-shot performance: Under the 10-shot scenario on FewGLUE, LoRA-IA3 achieves an average accuracy of 87.6%, with a few-shot performance decay rate of only 2.4%. This is significantly lower than pure LoRA (3.9%) and pure IA3 (5.8%), demonstrating the hybrid module's superior adaptability to small-sample data;

3. Comparative Advantages: LoRA-IA3 outperforms LoRA-Adapter by 1.5 percentage points on MMLU and reduces the few-shot performance decay rate by 0.5 percentage points. This is attributed to IA3's more flexible activation value control compared to Adapter's bottleneck structure, making it better suited for few-shot scenarios.

3) Parameter Efficiency and Training Efficiency Comparison

Table 4 shows the parameter efficiency and training efficiency comparison between LoRA-IA3 and baseline methods:

Table 4 reveals:

1. Parameter Efficiency: LoRA-IA3 requires only 1256MB (FP16) of additional parameter storage–just 0.96% of full parameter fine-tuning–representing a 31.8% reduction compared to LoRA-Adapter, demonstrating significant parameter efficiency advantages.

2. VRAM Usage: LoRA-IA3 consumes 11.3GB of VRAM, reducing VRAM usage by 60.4% compared to full parameter fine-tuning. While only 29.9% higher than pure IA3, it achieves a 10.4 percentage point performance improvement (GLUE average accuracy), demonstrating an excellent VRAM-performance tradeoff.

3. Training time: LoRA-IA3 achieves a training time of 20.8 hours (on the GLUE dataset), reducing the time by 69.5% compared to full-parameter fine-tuning and by 24.6% compared to LoRA-Adapter, demonstrating high training efficiency.

4. Inference Speed: LoRA-IA3 achieves 122 tokens/s, comparable to pure LoRA and pure IA3 while improving by 16.2% over LoRA-Adapter. This stems from IA3's significantly lower element-wise multiplication overhead compared to Adapter's bottleneck structure (Down-project→Up-project).

4) Generalization and Robustness Experiment Results

Table 5 presents LoRA-IA3's cross-task generalization and robustness experiment results:

Table 5 reveals: Cross-task generalization capability: LoRA-IA3 achieves 87.6% cross-task transfer accuracy, improving by 6.1 percentage points over pure LoRA and 3.4 percentage points over LoRA-Adapter. The core reason is that the attention weights optimized by LoRA enhance the generality of context-related features, while the activation value distribution regulated by IA3 improves the adaptability of task-specific features. Their synergy enables the model to rapidly transfer to untrained tasks.

5) Ablation Experiment Results

The ablation experiment results for each module of LoRA-IA3 are shown in Table 6:

The ablation results validate the core functions of each module:

1. Role of the LoRA module: After removing LoRA, model accuracy dropped from 89.7% to 82.5%, a decrease of 7.2 percentage points, proving that LoRA's optimization of attention weights is crucial for enhancing the adaptability of context-related features.

2. IA3 Module: After ablation, model accuracy dropped to 86.6%, a decrease of 3.1 percentage points, demonstrating that IA3's regulation of activation values effectively enhances task-specific feature extraction capabilities;

3. Cross-layer Feature Interaction Mechanism: After ablation, accuracy decreased to 87.2%, a drop of 2.5 percentage points, indicating that collaborative optimization between modules reduces feature conflicts and improves the model's overall expressive power;

4. Role of Regularization Mechanisms: After regularization ablation, overfitting rate increased from 3.2% to 5.8%,

TABLE 3. Generalization Performance Comparison of PEFT Methods on Few-shot and Multi-task Benchmarks

Method	MMLU Average Accuracy (%)	FewGLUE Average Accuracy (%)	Few-shot Performance Attenuation Rate (%)
Full Fine-tuning	86.5±0.4	88.3±0.3	2.1±0.2
Pure LoRA ($r = 32$)	81.2±0.5	83.5±0.4	3.9±0.3
Pure IA3	77.5±0.6	79.8±0.5	5.8±0.4
LoRA-Adapter	83.8±0.4	86.1±0.3	2.9±0.2
IA3-LoRA (Wang et al.)	82.5±0.4	84.7±0.3	3.5±0.2
LoRA-IA3 (Ours)	85.3±0.3	87.6±0.3	2.4±0.1

TABLE 4. Efficiency Comparison of Different Parameter-Efficient Fine-Tuning Methods

Method	Parameter Update Ratio (%)	Additional Parameter Storage (MB)	Memory Usage (GB)	Training Time (h/GLUE)	Inference Speed (tokens/s)
Full Fine-tuning	100.0	131072 (FP16)	28.5±0.5	68.2±2.3	125±5
Pure LoRA ($r = 32$)	0.65	858	10.2±0.3	22.5±1.2	123±4
Pure IA3	0.009	11	8.7±0.2	18.3±1.0	124±3
LoRA-Adapter	1.4	1843	12.8±0.4	27.6±1.5	105±4
LoRA-IA3 (Ours)	0.95	1256	11.3±0.3	20.8±1.1	122±3

TABLE 5. Generalization and Robustness Performance Comparison of PEFT Methods

Evaluation Dimension	Method	Cross-task Transfer Accuracy (%)	Noisy Data Accuracy (%)	Noisy Performance Attenuation Rate (%)
Cross-task Generalization	Pure LoRA ($r = 32$)	81.5±0.6	–	–
	Pure IA3	78.3±0.7	–	–
	LoRA-Adapter	84.2±0.5	–	–
	LoRA-IA3 (Ours)	87.6±0.4	–	–
Robustness	Pure LoRA ($r = 32$)	–	80.3±0.8	7.5±0.3
	Pure IA3	–	76.5±0.9	7.4±0.4
	LoRA-Adapter	–	83.1±0.7	6.1±0.3
	LoRA-IA3 (Ours)	–	85.9±0.6	4.2±0.2

TABLE 6. Ablation Study Results of the Proposed LoRA-IA3 Model

Model Configuration	GLUE Average Accuracy (%)	Parameter Update Ratio (%)	Memory Usage (GB)	Training Time (h)	Overfitting Rate (%)
Complete Model (LoRA-IA3)	89.7±0.3	0.95	11.3±0.3	20.8±1.1	3.2±0.2
Ablate LoRA Module (IA3 Only)	82.5±0.6	0.009	8.6±0.2	18.1±1.0	4.5±0.3
Ablate IA3 Module (LoRA Only)	86.6±0.4	0.65	10.1±0.3	22.3±1.2	3.8±0.2
Ablate Cross-Layer Feature Interaction Mechanism	87.2±0.4	0.95	11.2±0.3	21.5±1.1	4.1±0.2
Ablate Regularization Mechanism	88.5±0.3	0.95	11.0±0.3	20.5±1.0	5.8±0.3

with accuracy dropping by 1.2 percentage points. This proves Dropout and gradient clipping effectively suppress overfitting and enhance model stability.

6) Probing Experiment for Layer Functional Division

Experimental Design: Use gradient attribution analysis and feature clustering to quantify layer outputs; Metrics: Contextual Relevance Score (0-1): Correlation with text logical connections; Task-Specificity Score (0-1): Correlation with task key information.

V. DISCUSSION

A. CORE ADVANTAGES AND TECHNICAL BREAKTHROUGHS

1) Core Advantages

The core advantages of the LoRA-IA3 hybrid adaptation method are embodied in "three highs and one low":

1. High parameter efficiency: Achieving performance comparable to full-parameter fine-tuning by updating only 0.8%-1.2% of model parameters, representing a 46% improvement over pure LoRA and a 95-fold increase over pure IA3;

2. High Task Adaptability: Through dual mechanisms of "low-rank weight optimization + activation value regulation," it excels in both context-dependent tasks (e.g., question-answering, reasoning) and task-specific tasks (e.g., sentiment analysis, classification), achieving an average GLUE accuracy of 89.7%;

3. High Generalization Robustness: Achieves 87.6% cross-task transfer accuracy, with only 2.4% performance decay under few-shot conditions and 4.2% decay under noisy data–significantly outperforming existing hybrid fine-tuning methods;

4. Low Resource Consumption: Training GPU memory usage reduced by 60.4% compared to full-parameter fine-tuning, with a single RTX 3090 supporting LLaMA-7B model fine-tuning. Inference speed matches the base model with no additional latency.

2) Technical Breakthroughs

1. Architectural Innovation: Introduces a hybrid adaptation architecture combining "Attention Layer LoRA + FFN Layer IA3," clearly defining their respective roles in "context-related feature optimization + task-specific feature regulation" to overcome the functional limitations of single PEFT methods;

2. Collaborative Mechanism Breakthrough: Designed cross-layer feature interaction and gradient co-propagation mechanisms to prevent feature conflicts and gradient imbalances between modules, accelerating model convergence by 15%-20%.

3. Training Strategy Optimization: Proposed hierarchical learning rates and dynamic weight decay strategies to match parameter update characteristics across modules, balancing training stability and convergence speed.

B. KEY FINDINGS

1. Complementary module functions form the core of hybrid fine-tuning: LoRA's optimization of attention weights and IA3's regulation of activation values exhibit significant complementarity. Their contribution ratios dynamically adjust across different task types, yielding 3%-7% performance gains over single-module approaches;

2. Hierarchical training is crucial for stable convergence: The LoRA module (low-rank matrix) requires a low learning rate (5e-4) to prevent over-updating, while the IA3 module (scaling factor) needs a higher learning rate (1e-3) to rapidly adapt to activation distributions. A uniform learning rate causes a 2.5%-4.0% performance drop;

3. The rank (r) of the low-rank matrix must dynamically adapt to model scale: For 7B models, $r = 32$ achieves optimal performance and parameter efficiency; for 13B models, $r = 64$ yields superior results (experimentally validated with a 1.8% accuracy boost). Rank selection positively correlates with model feature dimensions;

4. The granularity of activation value tuning impacts task adaptation accuracy: Channelwise scaling in IA3 offers greater flexibility than the bottleneck structure in Adapter, improving performance by 3.2%-5.1% in few-shot scenarios by more precisely matching task activation distributions.

VI. CONCLUSION

1. Designed the LoRA-IA3 hybrid adaptation architecture: Embedding the LoRA module in the Transformer attention layer optimizes context-related features, while embedding the IA3 module in the FFN layer regulates task-specific features. Parameter update volume is only 0.8%-1.2%, balancing parameter efficiency and expressive power.

2. Targeting the stability challenges encountered when adapting Large Language Models for complex, multi-modal tasks, we propose a hierarchical collaborative training strategy employing tiered learning rates, dynamic weight decay, and gradient co-propagation mechanisms. This resolves convergence speed disparities and gradient conflicts between hybrid modules, enhancing model training stability by 20%-25 %.

3. Experiments on GLUE, MMLU, FewGLUE benchmarks, LoRA-IA3 achieves average accuracy of 89.7% (GLUE) and 85.3% (MMLU)–only 0.5–1.2 percentage points below full parameter fine-tuning, 2.9–4.1 percentage points higher than pure LoRA, and 7.1–7.8 percentage points higher than pure IA3;

4. Generalization and robustness tests validate that LoRA-IA3 achieves 87.6% cross-task transfer accuracy, with a 2.4% performance decay rate under few-shot conditions and a 4.2% decay rate under noisy data–significantly outperforming existing hybrid fine-tuning methods.

AUTHOR CONTRIBUTIONS

Jiexian Bai designed, collected and analyzed the data, and drafted the manuscript. Jiexian Bai conducted the study, critically revised the manuscript for important intellectual content, and gave final approval of the version to be published. Jiexian Bai participated fully in the work, take public responsibility for appropriate portions of the content, and agreed to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

CONFLICTS OF INTEREST

The author declares no conflict of interest.

FUNDING

This research was supported by:

1. Supported by Scientific and Technologial Innovation Programs of Higher Education Institutions in Shanxi, STIP(2024L498).

2. 14th Five-Year Plan" for Education and Science in Shanxi Province, 2024 Annual Planning Project(GH-241033).

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] D. Rajesh, S. Sinha, P. A. Sen, and N. K. Rout, "LoRa Based Overload Detection System for Transportation with Messaging and Mailing Services," Transactions of the Indian National Academy of Engineering, 2026, (prepublish), pp. 1-15, doi: 10.1007/S41403-026-00565-7.

[2] H. Li, Q. Peng, X. Mou, Z. Zeng, R. Li, J. Wang, et al., "SEHLP: A summary-enhanced large language model for financial report sentiment analysis via hybrid LoRA and dynamic prefix tuning," Information Processing and Management, vol. 63, no. 4, p. 104639, 2026, doi: 10.1016/J.IPM.2026.104639.

[3] R. Pascual, S. M. Sara, A. Jurio, D. Paternain, and M. Galar, "Few-shot multi-token DreamBooth with LoRa for style-consistent character generation," Expert Systems With Applications, vol. 309, p. 131226, 2026, doi: 10.1016/J.ESWA.2026.131226.

[4] H. Qin, N. Li, G. Yang, and Y. Peng, "Cross-network cross-interface relaying via LoRa-ZigBee synergy: Enabling energy-efficient delay-constrained communication across low-power IoT networks," Internet of Things, vol. 36, p. 101878, 2026, doi: 10.1016/J.IOT.2026.101878.

[5] N. Chen, X. Liu, and H. Wu, "Robust time-frequency preamble detection for LoRa-modulated signals using optimized generalized likelihood ratio test," Digital Signal Processing, vol. 173, p. 105892, 2026, doi: 10.1016/J.DSP.2026.105892.

[6] C. Si, Z. Shi, S. Zhang, X. Yang, H. Pfister, and W. Shen, "Task-Specific Directions: Definition, Exploration, and Utilization in Parameter Efficient Fine-Tuning," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2026, doi: 10.1109/TPAMI.2026.3659168.

[7] R. Amirthavarshini, M. A. Shamil, A. P. Raaj, and G. Kanimozhi, "Harnessing LoRa for real-time landslide monitoring and early alerts in Kerala's terrain," Geoscience Frontiers, vol. 17, no. 2, p. 102242, 2026, doi: 10.1016/J.GSF.2025.102242.

[8] M. A. Alberti, E. Bonomo, H. R. Santos, V. A. D. J. Alberti, M. Pellenz, and R. Righi, "Applying NovaGenesis: A service-oriented, self-organizing, and programmable IoT architecture for LoRa and Wi-Fi-based environmental monitoring," Computer Communications, vol. 248, p. 108423, 2026, doi: 10.1016/J.COMCOM.2026.108423.

[9] Z. Ji, R. Wei, J. Liu, Y. Pang, and J. Han, "Interpretable Few-Shot Image Classification via Prototypical Concept-Guided Mixture of LoRA Experts," IEEE Transactions on Image Processing, vol. 35, pp. 930-942, 2026, doi: 10.1109/TIP.2026.3654473.

[10] A. Asiri and M. Saleh, "Continual Learning for Saudi-Dialect Offensive-Language Detection Under Temporal Linguistic Drift," Information, vol. 17, no. 1, p. 99, 2026, doi: 10.3390/INFO17010099.

[11] O. Chughtai, W. M. Rehan, M. Naeem, A. H. Alenezi, and S. A. Haider, "Distributed multi-objective consumer-centric routing for LoRa-based IoT-enabled FANET," Future Generation Computer Systems, vol. 179, p. 108368, 2026, doi: 10.1016/J.FUTURE.2026.108368.
[12] M. Wang, X. Tang, B. Wang, and J. Ren, "Micro-Expression Recognition via LoRA-Enhanced DinoV2 and Interactive Spatio-Temporal Modeling," Sensors, vol. 26, no. 2, p. 625, 2026, doi: 10.3390/S26020625.
[13] T. Wu, R. Mao, Y. Du, Q. Zhu, S. Wei, and C. Hu, "Preamble Injection-Based Jamming Method for UAV LoRa Communication Links," Sensors, vol. 26, no. 2, p. 614, 2026, doi: 10.3390/S26020614.
[14] M. J. Solé, P. R. Centelles, F. Freitag, M. R. Pallarès, and B. R. Viñas, "Large and reliable data transfer service for LoRa mesh network applications," Computer Communications, vol. 248, p. 108404, 2026, doi: 10.1016/J.COMCOM.2025.108404.
[15] S. Agrawal and R. Iyer, "Psychometric analysis using LLaMA based circular position embedding and gated residual auto-encoder," International Journal of Information Technology, 2026, (prepublish), pp. 1-9, doi: 10.1007/S41870-025-03072-0.
[16] N. Scaria, J. J. S. Kennedy, T. Latinovich, and D. Subramani, "EvalYaks: Instruction tuning datasets and LoRA fine-tuned models for automated scoring of CEFR B2 speaking assessment transcripts," Computers and Education: Artificial Intelligence, vol. 10, p. 100539, 2026, doi: 10.1016/J.CAEAI.2025.100539.
[17] K. D. Niranjan, M. Supriya, and W. Tiberti, "Secure TPMS Data Transmission in Real-Time IoV Environments: A Study on 5G and LoRa Networks," Sensors, vol. 26, no. 2, p. 358, 2026, doi: 10.3390/S26020358.
[18] A. E. U. Rodríguez, G. A. H. Osuna, G. F. Vázquez, J. A. N. Pintor, L. F. L. Vega, E. L. Neri, et al., "Design and Validation of a Solar-Powered LoRa Weather Station for Environmental Monitoring and Agricultural Decision Support," Technologies, vol. 14, no. 1, p. 32, 2026, doi: 10.3390/TECHNOLOGIES14010032.
[19] L. J. León, V. Nogueira, P. Salgueiro, and P. Quaresma, "Describing Land Cover Changes via Multi-Temporal Remote Sensing Image Captioning Using LLM, ViT, and LoRA," Remote Sensing, vol. 18, no. 1, p. 166, 2026, doi: 10.3390/RS18010166.
[20] D. Zorbas, P. D. L. Pugliese, R. Saduakhas, and F. Guerriero, "Gateway configuration in 2.4 GHz LoRa networks," Internet of Things, vol. 31, p. 101567, 2025, doi: 10.1016/J.IOT.2025.101567.
[21] L. Martini, D. Zolezzi, S. Iacono, L. Oneto, and G. V. Vercelli, "Multi stage generative upscaler recovers low resolution football broadcast images through diffusion models with ControlNet conditioning and LoRA fine tuning," Scientific Reports, vol. 16, no. 1, p. 1882, 2025, doi: 10.1038/S41598-025-31543-8.
[22] J. Lee, S. Pedro, and K. Michaud, "Age of Disease Onset and Risk of Serious Infection with anti-TNF Use in Older Adults with Rheumatoid Arthritis," Innovation in Aging, vol. 9, Supplement 2, 2025, doi: 10.1093/GERONI/IGAF122.2413.
[23] J. Yang, "How artificial intelligence improves corporate governance: Evidence from the ERNIE large model," Finance Research Letters, vol. 96, pp. 109782-109782, 2026, doi: 10.1016/J.FRL.2026.109782.
[24] C. Minoccheri, M. Hodgman, H. Ma, R. Merchant, E. Wittrup, C. Williamson, et al., "LoRA-based methods on Unet for transfer learning in aneurysmal subarachnoid hematoma segmentation," BMC Medical Imaging, vol. 26, no. 1, p. 58, 2025, doi: 10.1186/S12880-025-02116-Y.
[25] J. Liu and Y. Huang, "LoRA-Fine-Tuned Latent Diffusion for High-Fidelity Digitization of Classic Mongolian Patterns," Applied Sciences, vol. 16, no. 1, p. 11, 2025, doi: 10.3390/APP16010011.
[26] M. M. Lazaro, A. G. Lazaro, D. S. Girbau, et al., "Long-range Low-power Devices Based on LoRa Backscattering for Next Generation IoT Applications," John Wiley & Sons, Inc., 2025-10-24, doi: 10.1002/9781394417957.
[27] Y. Zhai, Y. Lei, H. Zhang, Y. Yu, K. Xu, D. Feng, et al., "Uncertainty-penalized reinforcement learning from human feedback with diversified reward LoRA ensembles," Information Processing and Management, vol. 63, no. 3, p. 104548, 2026, doi: 10.1016/J.IPM.2025.104548.
[28] S. Ryczek and M. Sobieraj, "Secure LoRa-Based Transmission System: An IoT Solution for Smart Homes and Industries," Electronics, vol. 14, no. 24, p. 4977, 2025, doi: 10.3390/ELECTRONICS14244977.
[29] H. Qiao, H. Guan, H. Li, A. Barbosa, and Y. Zhu, "Developing a synchronised LoRa wireless sensor network for cost-effective footbridge vibration monitoring," Journal of Infrastructure Intelligence and Resilience, vol. 5, no. 1, p. 100187, 2026, doi: 10.1016/J.IINTEL.2025.100187.
[30] A. A. M. Khan and S. Luo, "Dual resource allocation for enhancing vehicle tracking using deep neural network and Bayesian LoRa," Expert Systems With Applications, vol. 303, p. 130717, 2026, doi: 10.1016/J.ESWA.2025.130717.