Balancing Fidelity and Plasticity: Aligning Mixed-Precision Fine-Tuning with Linguistic Hierarchies

Changhai Zhou[1*] **Shiyang Zhang**[3*] **Yuhua Zhou**[4*] **Jun Gao**[4]
Qian Qiao[5] **Shichao Weng**[1] **Weizhong Zhang**[2] **Cheng Jin**[1,†]
[1]College of Computer Science and Artificial Intelligence, Fudan University
[2]School of Data Science, Fudan University
[3]Yale University [4]Zhejiang University [5]Soul AILab
chzhou25@m.fudan.edu.cn {weizhongzhang,jc}@fudan.edu.cn

Abstract

Deploying and fine-tuning Large Language Models (LLMs) on resource-constrained edge devices requires navigating a strict trade-off between memory footprint and task performance. Existing quantization-aware fine-tuning methods typically decouple weight precision and adapter capacity, overlooking that a layer's ability to adapt is constrained by the information preserved in its frozen weights. Layers that are highly sensitive to quantization—whether due to representational specialization or accumulated error propagation—can become bottlenecks that adapter rank alone cannot recover. To address this issue, we introduce **QR-Adaptor**, a unified framework that jointly optimizes per-layer quantization bit-width and LoRA rank. We formulate resource allocation as a multi-objective discrete search guided by empirical layer-wise sensitivity, and implement it with a three-stage pipeline comprising KL-based sensitivity profiling, evolutionary exploration, and Bayesian refinement. Extensive experiments across LLaMA and Qwen models, including modern instruction tuning on OpenOrca and comparisons with strong PEFT baselines such as QDoRA, show that QR-Adaptor establishes a strong Pareto frontier: under a strict 4-bit memory budget, it matches or approaches 16-bit baselines while using substantially less memory. Our code is publicly available at https://github.com/harrysyz99/qr_adapter.

1 Introduction

The democratization of LLMs relies heavily on the ability to fine-tune and deploy them on consumer-grade hardware (Touvron et al., 2023; Wan et al., 2023). To circumvent the prohibitive memory costs associated with full-parameter tuning, the research community has converged on two primary paradigms for efficient adaptation: weight quantization, which reduces the precision of the frozen base model (Dettmers et al., 2022b; Frantar et al., 2023), and Parameter-Efficient Fine-Tuning (PEFT), which updates a sparse set of auxiliary parameters such as Low-Rank Adapters (LoRA) (Hu et al., 2022). The integration of these techniques, exemplified by QLoRA (Dettmers et al., 2023a), has set a new standard for adapting massive models under limited memory budgets.

However, current approaches largely treat quantization and adaptation as independent optimization problems. Recent automated quantization frameworks (Lee et al., 2025b), successfully assign mixed bit-widths based on layer sensitivity. Yet, these methods are primarily designed for post-training quantization, aiming to maximize reconstruction fidelity for frozen inference models rather than optimizing the model's trainability. On the other hand, adaptive PEFT methods (Zhang et al., 2023; Zhou et al., 2025a) focus solely on rank allocation, typically assuming a fixed or uniform base model precision. This compartmentalized perspective ignores a fundamental coupling effect: the learning potential of a fine-tuning adapter is inherently constrained by the fidelity of the underlying quantized weights. A high-rank adapter attached to a layer collapsed by aggressive quantization may yield diminishing returns, wasting valuable memory budget that could be better utilized elsewhere.

We attribute this limitation to the neglect of the Fidelity-Plasticity Trade-off. In our theoretical view, the performance of a layer is determined by the synergy between its static capacity (*Fidelity*, determined by quantization bit-width) and its dynamic adaptability (*Plasticity*, determined by adapter rank). Transformer models exhibit substantial layer-wise heterogeneity, but its origin is unlikely to be explained by a single factor. Prior probing studies suggest that earlier layers often capture more local or syntactic cues while later layers support more abstract processing (Jawahar

[*]These authors contributed equally.
[†]Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 15885–15896

et al., 2019; Tenney et al., 2019); however, this division is not absolute, and later-layer fragility can also reflect accumulated error propagation across depth (Niu et al., 2022; Dettmers et al., 2022b). Uniform quantization therefore risks suppressing the Fidelity of precisely the layers whose outputs are most sensitive to noise, creating an irreversible information bottleneck. In such scenarios, increasing the adapter rank (Plasticity) yields diminishing returns because the underlying signal is too noisy to be effectively modulated. Conversely, assigning high precision to robust layers wastes memory budget. Consequently, the optimal configuration is not global uniformity, but a strategic re-allocation that respects empirical layer-wise sensitivity. We posit that efficiency is driven not merely by parameter reduction, but by harmonizing the distinct requirements of Fidelity and Plasticity across different layers. To operationalize this insight, we propose shifting the paradigm from manual heuristics to automated joint optimization.

To this end, we introduce **QR-Adaptor**, a unified framework that jointly optimizes per-layer bit-width and LoRA rank. Unlike prior works that rely on differentiable proxies which may misalign with discrete quantization objectives, we formulate the problem as a multi-objective discrete search directly guided by downstream task performance. We develop a systematic three-stage pipeline: starting with task-informed initialization based on information theoretic sensitivity, proceeding with global exploration via a Pareto-ranking genetic algorithm, and concluding with local refinement using Bayesian Optimization. This approach allows the model to automatically "steal" bits from redundant layers and reinvest them into capacity-critical ones.

Our main contributions are summarized as follows:

- We characterize the *Fidelity-Plasticity Trade-off* in quantized fine-tuning. We provide empirical evidence that decoupled optimization leads to suboptimal resource allocation, as the adaptation potential of high-rank adapters is constrained when the underlying weight fidelity falls below a critical threshold.
- We propose QR-Adaptor, a gradient-free framework that automates the joint search for bit-width and rank. By treating resource allocation as a multi-objective optimization problem, our method aligns numerical precision with empirical layer-wise sensitivity rather than assuming uniform precision or fixed manual rules.
- We demonstrate that QR-Adaptor establishes a strong Pareto frontier in the accuracy-memory trade-off. Across LLaMA and Qwen models, and in focused comparisons on OpenOrca and advanced PEFT baselines, a strategically allocated 4-bit memory budget can rival the performance of 16-bit LoRA baselines.

2 Related Work

2.1 LLM Quantization

Quantization facilitates efficient deployment by mapping weights to lower precision. Early uniform methods like LLM.int8 (Dettmers et al., 2022a) enabled 8-bit inference. PTQ pushed limits to 4-bit: GPTQ (Frantar et al., 2023) utilizes Hessian information, while AWQ (Lin et al., 2023) protects activation outliers. Recent advancements focus on handling extreme outliers to unlock lower bit-widths. SpQR (Dettmers et al., 2023b) and Atom (Zhao et al., 2024) demonstrate that a small fraction of "outlier" weights requires higher precision to preserve model fidelity, while the vast majority can be aggressively compressed. QuaRot (Ashkboos et al., 2024) further mitigates quantization error via rotation matrices.

Recognizing the layer-wise heterogeneity of LLMs, mixed-precision strategies have gained traction. MixLLM (Wang et al., 2025) and SliM-LLM (Huang et al., 2025) assign bit-widths based on saliency. More recently, AMQ (Lee et al., 2025b) introduced an automated pipeline to search for optimal mixed-precision configurations. However, these methods are primarily designed for inference efficiency, aiming to maximize reconstruction fidelity for frozen models. They treat the model as static, overlooking the plasticity required during fine-tuning. As a result, a configuration optimized purely for inference reconstruction often creates bottlenecks for downstream adaptation.

2.2 Parameter-Efficient Fine-Tuning

PEFT adapts LLMs to downstream tasks with minimal overhead. LoRA (Hu et al., 2022) approximates updates via low-rank matrices. To improve flexibility, dynamic rank allocation methods have emerged. DyLoRA (Valipour et al., 2023) trains LoRA modules across a range of ranks to enable

dynamic search-free adaptation. AdaLoRA (Zhang et al., 2023), RankAdaptor (Zhou et al., 2025a), and LaRA (Zhou et al., 2026c) dynamically allocate or refine rank budgets based on layer importance, while BSLoRA (Zhou et al., 2025b) improves parameter efficiency through intra-layer and inter-layer sharing. DoRA (Liu et al., 2024) further decomposes weights into magnitude and direction to resemble full fine-tuning capacity, and quantized DoRA variants (QDoRA) strengthen the PEFT operator itself in low-precision settings. Despite their effectiveness, these methods typically assume a fixed base model precision (e.g., uniform 4-bit). They optimize the auxiliary parameters (adapters) in isolation, ignoring the fact that a layer's learning potential is fundamentally constrained by the quantization noise of its frozen weights. This "rank-only" optimization leads to suboptimal resource utilization, as high ranks may be allocated to layers where the underlying information has already been irreversibly damaged.

2.3 Quantization-Aware Fine-Tuning

The integration of quantization and PEFT aims to enable training on consumer hardware. QLoRA (Dettmers et al., 2023a) established the standard by combining 4-bit NormalFloat quantization with LoRA. Recent works have sought to refine this synergy. QA-LoRA (Xu et al., 2023) introduces group-wise quantization-aware operators to reduce the discrepancy between quantized weights and low-rank adapters, enhancing stability. To mitigate the quantization error introduced at the start of training, LoftQ (Li et al., 2023) proposes a novel initialization strategy using Singular Value Decomposition on the weight residuals. Concurrent work such as AutoQRA (Zhou et al., 2026b) also explores joint optimization of mixed-precision quantization and low-rank adapters. QLoRA employs a rigid, uniform configuration. While QA-LoRA improves the update mechanics, LoftQ improves initialization, and AutoQRA explores a related joint-search direction, these methods generally operate under different architectural assumptions or focus on complementary aspects of the pipeline. In contrast, QR-Adaptor focuses on the joint configuration search of bit-width and rank. We argue that initialization strategies and operator improvements are complementary to our architectural search; however, our unique contribution lies in solving the resource allocation problem to maximize the upper bound of model performance under extreme constraints.

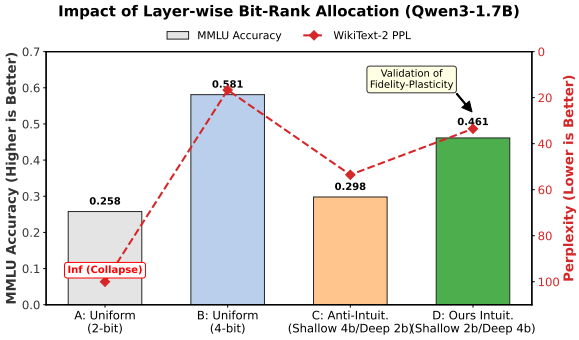

Figure 1: **Empirical validation of the Fidelity-Plasticity Trade-off.** We compare four configurations on Qwen-3-1.7B. Crucially, despite having the same memory budget, **Config D** (preserving higher fidelity in later layers) significantly outperforms **Config C** (preserving higher fidelity in earlier layers). This comparison highlights depth-wise sensitivity under joint bit-rank allocation without assuming a single causal explanation for why later layers are more fragile.

Scope of Efficiency. Other compression techniques, such as structural pruning and joint sparsity/rank optimization (Zhou et al., 2024, 2026a; Ma et al., 2023; Sun et al., 2024; Frantar and Alistarh, 2023) or knowledge distillation (Hinton et al., 2015; Hsieh et al., 2023; Jiao et al., 2020), are orthogonal to our work. Efficient in-context learning frameworks target a different bottleneck from the quantized gradient-based fine-tuning studied here (Gao et al., 2025). QR-Adaptor can, in principle, be applied to a pruned model to further enhance its adaptability, but such combinations are beyond the scope of this paper.

3 Methodology

3.1 Theoretical Framework & Motivation

To move beyond heuristic resource allocation, we first establish a theoretical model governing the interaction between quantization and adaptation. Let $\mathcal{M}$ denote the LLM. We model the total information capacity $\mathcal{C}_{total}^{(l)}$ of the l-th layer as the sum of its *Static Capacity* (frozen pre-trained weights) and *Dynamic Capacity* (trainable adapters).

Fidelity-Plasticity Coupling. We define two key properties for each layer:

1. **Fidelity (Φ_l):** The ability of the quantized weights $W_q^{(l)}$ to retain pre-trained knowledge. This is strictly a function of the bit-width b_l. Lower bits introduce a quantization noise term $\mathcal{E}_q(b_l)$, reducing the mutual information with

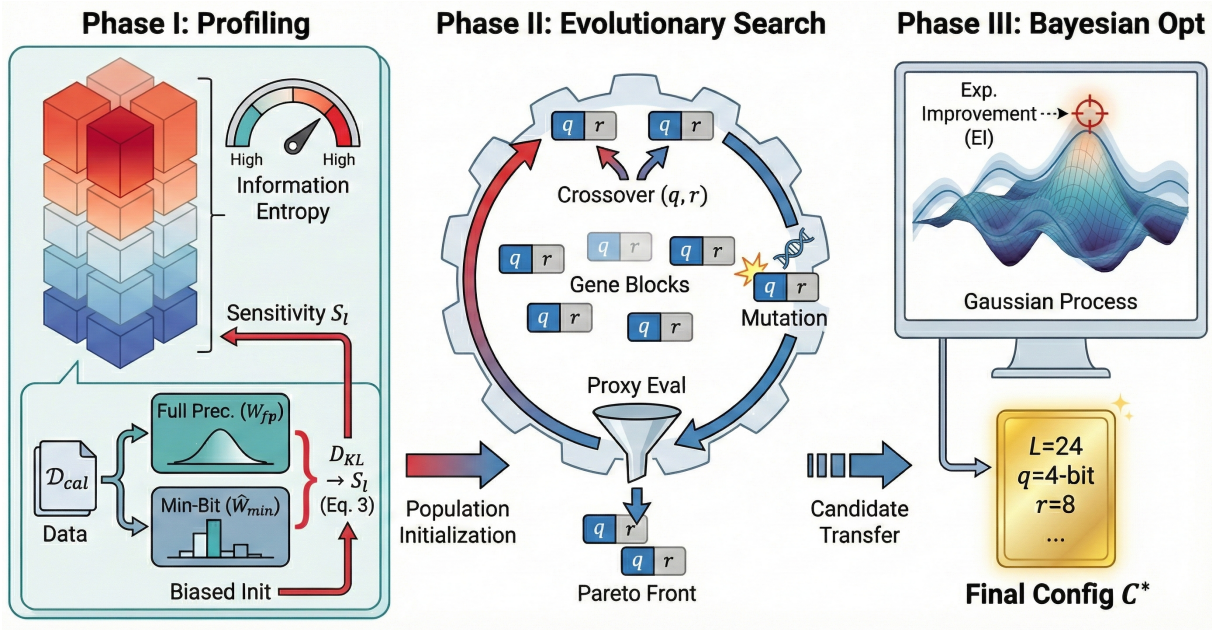

Figure 2: **Overview of the QR-Adaptor Framework.** The pipeline consists of three synergistic stages: (I) **Fidelity Sensitivity Profiling** initializes the population based on KL-based sensitivity profiling to respect layer-wise task demand; (II) **Discrete Landscape Exploration** utilizes a constrained evolutionary strategy to approximate the global Pareto frontier without gradient mismatch; (III) **Bayesian Frontier Refinement** employs Gaussian Process regression to pinpoint the optimal bit-rank configuration within the non-smooth solution space.

the original weights $W_{fp}^{(l)}$:

$$\Phi_l(b_l) \propto \mathcal{I}(W_{fp}^{(l)}; W_q^{(l)}) \approx f(b_l) \quad (1)$$

2. **Plasticity (Π_l):** The capacity of the adapter to mitigate quantization noise and learn new tasks. This is governed by the rank r_l of the low-rank matrices A_l, B_l:

$$\Pi_l(r_l) \propto \text{Rank}(A_l B_l) = r_l \quad (2)$$

Synergistic Optimization Hypothesis. For a layer to adapt effectively without collapsing, its Plasticity must be sufficient to compensate for both the fidelity loss and the layer-specific adaptation demand ($\mathcal{T}_l$). We posit a conceptual lower-bound for effective adaptation:

$$r_l \geq \alpha \cdot \mathcal{E}_q(b_l) + \beta \cdot \mathcal{T}_l \quad (3)$$

where α, β are theoretical coefficients that qualitatively encode the relative impact of quantization noise and task demand. Equation 3 is used only to motivate the coupling between Fidelity and Plasticity: α and β are not manually tuned hyperparameters and are never optimized directly in our search. Instead, Phase I computes an empirical proxy for this trade-off through KL-based sensitivity profiling, and the search itself is driven by proxy validation loss.

Empirical Validation. To validate this hypothesis, we conducted a controlled pilot study on Qwen-3-1.7B (28 layers) fine-tuned on the Alpaca dataset for 1 epoch. We partitioned the model into *Shallow* (Layers 0–13) and *Deep* (Layers 14–27) blocks. This coarse split is motivated by prior interpretability work and by the fact that perturbations in later layers also accumulate all upstream quantization error; accordingly, the study probes depth-wise sensitivity but does not claim to disentangle representational specialization from error propagation. We evaluated four configurations under the same budget:

- **Config A (Uniform Low):** 2-bit + Rank 8.
- **Config B (Uniform High):** All 4-bit + Rank 8 (Standard Baseline).
- **Config C (Later-Low-Fidelity):** Shallow 4-bit + Rank 8 / Deep 2-bit + Rank 16.
- **Config D (Later-High-Fidelity):** Shallow 2-bit + Rank 16 / Deep 4-bit + Rank 8.

The results, visualized in Figure 1, provide clear evidence that allocation strategy matters. **Config A** collapses completely (PPL: Inf, MMLU: 0.2577), confirming that uniform 2-bit is insufficient. Comparing the two mixed-precision variants is most

revealing: **Config C**, which assigns 2-bit precision to later layers, suffers severe degradation (PPL: 53.53, MMLU: 0.2980) even though those layers receive a higher rank. This shows that additional Plasticity cannot fully recover from a severe Fidelity bottleneck.

In contrast, **Config D** achieves much stronger performance (PPL: 33.52, MMLU: 0.4613) with the *same average memory footprint* as Config C. Preserving Fidelity in later layers and spending the remaining budget on earlier layers is therefore substantially more effective than the reverse allocation. While this experiment does not isolate whether the gain arises from representational specialization or compounded error propagation, it does demonstrate that later-layer fragility must be respected during joint bit-rank allocation.

3.2 Problem Formulation

Guided by the Synergistic Optimization Hypothesis, we formulate fine-tuning as a constrained multi-objective discrete optimization problem. Our goal is to find a global configuration that satisfies the layer-wise constraints implied by Eq. 3 while minimizing memory usage.

Consider a pre-trained LLM with L layers. For each layer l, we assign a tuple (q_l, r_l) from the discrete search spaces of bit-widths $\mathcal{Q}$ and ranks $\mathcal{R}$. The forward pass is defined as:

$$y = \underbrace{\text{Quantize}(W_l, q_l)x}_{\text{Fidelity Term } (\Phi_l)} + \underbrace{\frac{\gamma}{r_l} A_l B_l x}_{\text{Plasticity Term } (\Pi_l)} \tag{4}$$

Unlike previous works that fix the Fidelity term globally, we optimize both terms jointly. We define the search space as $\mathcal{S} = (\mathcal{Q} \times \mathcal{R})^L$. The objective is to identify a configuration $C^* \in \mathcal{S}$ that maximizes validation performance $\mathcal{P}$ subject to a hard memory budget M_{budget}:

$$\begin{aligned} \max_{C} \quad & \mathcal{P}(C; \mathcal{D}_{val}, \theta_C^*) \\ \text{s.t.} \quad & \sum_{l=1}^{L} \text{Mem}(q_l, r_l) \leq M_{budget} \end{aligned} \tag{5}$$

where θ_C^* denotes the parameters after fine-tuning. This combinatorial problem is non-differentiable and computationally expensive. To solve it, we propose the QR-Adaptor framework (Figure 2), which systematically navigates this space to find solutions that harmonize Φ_l and Π_l across all layers.

3.3 Phase I: Fidelity Sensitivity Profiling

The optimization landscape defined by Eq. 5 is vast and non-convex. A random cold start ignores the heterogeneous nature of $\mathcal{T}_l$ (task demand), leading to inefficient convergence. To align the initial population with the model's intrinsic structure, we propose *Fidelity Sensitivity Profiling*.

We use relative entropy (KL-divergence) as an empirical proxy for layer sensitivity. Specifically, S_l measures how much the output distribution of layer l shifts when that layer is pushed to the minimum bit-width on a calibration set $\mathcal{D}_{cal}$:

$$S_l = \frac{1}{|\mathcal{D}_{cal}|} \sum_{x \in \mathcal{D}_{cal}} D_{KL}\big(P(y|x; W_{fp}) \parallel P(y|x; \hat{W}_{min}^{(l)})\big) \tag{6}$$

A high S_l indicates that aggressive quantization at layer l produces a strong fidelity bottleneck. In practice, this KL-based score serves as the operational proxy for the conceptual quantities in Eq. 3: we do not estimate $\mathcal{T}_l$, α, or β directly. We then bias the initialization so that the probability of assigning higher (q_l, r_l) to layer l is proportional to its normalized sensitivity score $\hat{S}_l$, pruning regions where the fidelity bottleneck is likely to be severe.

3.4 Phase II: Discrete Landscape Exploration

Given the initialized population, we employ a discrete evolutionary strategy to navigate the Fidelity-Plasticity landscape. We adapt the NSGA-II framework (Deb et al., 2002) specifically for the coupled nature of our problem. Unlike differentiable architecture search methods which rely on relaxed continuous proxies, evolutionary search directly evaluates discrete configurations, avoiding the gradient mismatch problem inherent in quantization.

Synergistic Operators. Standard crossover operations may disrupt the delicate balance between bit-width and rank. We design operators to preserve structural integrity:

- *Layer-wise Crossover:* We treat the tuple (q_l, r_l) as an atomic gene. Offspring inherit the complete configuration of a layer from one parent, ensuring that the local fidelity-plasticity alignment established in previous generations is preserved.
- *Proximity Mutation:* To avoid destructive jumps in the loss landscape, we restrict mutations to immediate discrete neighbors (e.g., 4-bit $\leftrightarrow$ 2-bit). This allows the search to locally adjust the inequality terms in Eq. 3 without causing catastrophic collapse.

Efficient Proxy Evaluation. Evaluating the true plasticity of a configuration requires full fine-tuning, which is computationally prohibitive. We utilize *Proxy Tuning* as a cost-effective estimator. We update the adapter parameters for only a few steps on $\mathcal{D}_{cal}$. This brief adaptation phase is sufficient to reveal whether a configuration violates the fidelity bottleneck—if a layer's capacity is insufficient, the loss will fail to decrease even in early stages. This proxy metric efficiently ranks individuals to approximate the Pareto frontier $\mathcal{C}_{front}$.

3.5 Phase III: Bayesian Frontier Refinement

While Phase II efficiently identifies the global Pareto front, genetic algorithms can lack precision in local convergence. To pinpoint the exact optimum for a specific deployment constraint, we employ Bayesian Optimization (BO).

We focus the search on the promising regions identified by $\mathcal{C}_{front}$. We model the objective function $f(C)$ using a Gaussian Process (GP) with a *Matérn-5/2 Kernel*. This kernel is chosen specifically for its ability to model rough, non-smooth landscapes characteristic of discrete quantization, where performance can drop sharply between adjacent configurations (as seen in our pilot study).

We iteratively select the next candidate configuration C_{next} to evaluate by maximizing the *Expected Improvement* (EI) over the current best solution y_{best}:

$$\text{EI}(C) = \mathbb{E}\left[\max(0, f(C) - y_{best})\right] \quad (7)$$

By maximizing EI, the framework automatically balances exploitation (refining high-performing configurations) and exploration (testing uncertain regions). This stage acts as a final "polishing" step, ensuring that the selected bit-rank allocation is mathematically optimized to harmonize the trade-off between memory and task performance.

4 Evaluation

4.1 Experimental Setup

Models & Datasets. Our primary evaluation focuses on standard-scale open-source architectures, specifically **LLaMA-3-8B** (Grattafiori et al., 2024) and **Qwen-3-4B/8B** (Yang et al., 2025). We extend our analysis to compact models (**LLaMA-3.2-1B/3B**, **Qwen-3-1.7B**) and other generations (**LLaMA-2**, **Qwen-2.5**, **LLaMA-3.1**). For instruction tuning, we use **Alpaca-52k** (Taori et al., 2023) for direct comparability with prior baselines, and additionally evaluate **Qwen-3-4B** on a randomly sampled 100k subset of **OpenOrca** (Lian et al., 2023) to reflect a more modern instruction-tuning corpus. We report zero-shot performance on standard commonsense benchmarks: **ARC-E/C** (Clark et al., 2018), **PIQA** (Bisk et al., 2020), **Hella** (Zellers et al., 2019), **WinoGrande** (Sakaguchi et al., 2021), **BoolQ**, and **OpenBookQA (OBQA)**. We also report Perplexity (PPL) on **WikiText-2** (Merity et al., 2016) and **C4** (Raffel et al., 2023). For mathematical reasoning, we fine-tune and evaluate on **GSM8K** (Cobbe et al., 2021) (8-shot).

Baselines. We compare QR-Adaptor against four baseline categories: (1) *Upper Bound:* standard LoRA on FP16 base models; (2) *Uniform Quantization:* QLoRA with 4-bit base models; (3) *Adaptive Search Methods:* AdaLoRA (rank-only search, target avg $r = 16$) and AMQ+LoRA (bit-only search via AMQ, fixed $r = 16$); and (4) *Advanced PEFT Operators:* DoRA (Liu et al., 2024) and the quantized DoRA variant QDoRA, which we evaluate in a focused comparison on Qwen-3-4B. We also evaluate lower-bit uniform variants (QLoRA 2/3-bit) and recent quantization-aware or compensation methods, including LoftQ (Li et al., 2023), LQ-LoRA (Guo et al., 2024), ApiQ (Liao et al., 2024), and RILQ (Lee et al., 2025a).

Implementation Details. We use *PyTorch* 2.1.2, *BitsAndBytes* 0.43.1, *Transformers* 4.41.0, *PEFT* 0.11.1, *Optuna* 3.6.1, and *CUDA* 12.4 on NVIDIA A100 GPUs under Ubuntu. For the main setting, we use a population size of 5, generate 1 offspring per generation, and run 5 iterations for both the evolutionary and Bayesian refinement stages.

4.2 Main Results

We present a comprehensive evaluation of QR-Adaptor on general NLU tasks and complex reasoning benchmarks. Our results demonstrate that joint bit-rank optimization consistently establishes a new Pareto frontier across varying model scales, from 1B to 8B parameters.

General NLU Capabilities. Table 1 reports zero-shot performance and perplexity across the Qwen-3 and LLaMA-3 families. We observe distinct advantages in two operating regimes. First, in the efficiency-focused regime, **QR-Adaptor-4** (averaging $\sim$3.5 bits) consistently outperforms the standard 4-bit QLoRA baseline despite using approximately 12% less parameter memory. For

Model	Method	Avg. Bit	Avg. Rank	Wiki2 ↓	C4 ↓	ARC-c ↑	ARC-e ↑	Hella ↑	PIQA ↑	Wino ↑	BoolQ ↑	OBQA ↑	Avg. Acc. ↑
Qwen Family													
Qwen-3-4B	LoRA (FP16)	16.0	16.0	12.65	14.52	48.2	78.5	72.8	76.2	66.5	79.2	34.6	65.1
	QLoRA (4-bit)	4.0	16.0	13.05	14.98	45.8	76.2	70.5	74.5	64.2	77.0	32.4	62.9
	AdaLoRA	16.0	12.8	12.85	14.82	47.1	77.4	71.8	75.4	65.3	78.2	33.4	64.1
	AMQ + LoRA	4.50	16.0	12.92	14.95	47.8	77.8	72.0	75.0	64.8	77.5	33.0	64.0
	QR-Adaptor-4	**3.6**	**10.4**	12.82	14.72	47.0	77.5	71.5	75.2	65.5	78.0	34.0	64.1
	QR-Adaptor-6	**5.3**	**11.9**	**12.45**	**14.35**	**48.8**	**79.0**	**73.2**	**76.8**	**67.0**	**79.5**	**35.0**	**65.6**
Qwen-3-8B	LoRA (FP16)	16.0	16.0	10.49	12.15	58.5	82.2	78.5	78.8	72.5	82.5	38.2	70.2
	QLoRA (4-bit)	4.0	16.0	10.85	12.55	55.8	80.0	76.2	76.5	70.2	80.0	35.8	67.8
	AdaLoRA	16.0	12.8	10.62	12.35	57.2	81.2	77.5	77.8	71.5	81.5	37.0	69.0
	AMQ + LoRA	4.33	16.0	10.68	12.42	57.8	81.5	77.8	77.0	71.0	80.5	36.2	68.8
	QR-Adaptor-4	**3.5**	**10.5**	10.65	12.32	57.2	81.0	77.5	77.5	71.5	81.2	37.0	69.0
	QR-Adaptor-6	**5.2**	**12.0**	**10.35**	**12.05**	**59.2**	**82.5**	**79.0**	**79.2**	**73.0**	**82.8**	**38.6**	**70.6**
LLaMA Family													
LLaMA-3-8B	LoRA (FP16)	16.0	16.0	7.42	8.65	56.2	83.5	77.2	80.5	74.0	84.0	40.4	70.8
	QLoRA (4-bit)	4.0	16.0	7.65	8.92	53.5	81.2	74.5	78.2	71.5	81.5	38.0	68.3
	AdaLoRA	16.0	12.8	7.55	8.80	55.0	82.5	76.0	79.5	72.8	83.0	39.2	69.7
	AMQ + LoRA	4.25	16.0	7.62	8.88	55.5	82.8	76.5	78.5	72.5	81.8	38.6	69.4
	QR-Adaptor-4	**3.5**	**10.6**	7.52	8.75	55.0	82.2	75.8	79.2	72.8	82.5	39.0	69.5
	QR-Adaptor-6	**5.25**	**12.0**	**7.38**	**8.55**	**56.8**	**83.8**	**77.5**	**80.8**	**74.5**	**84.2**	**40.8**	**71.2**

Table 1: **Main Results on General NLU Benchmarks (Larger Models). Bold** indicates the best result.

Method	Avg. Bit	*LLaMA Family* 3-8B	*LLaMA Family* 3.2-3B	*Qwen Family* 3-8B	*Qwen Family* 3-4B
LoRA (FP16)	16.0	78.5	68.5	84.2	78.2
QLoRA (4-bit)	4.0	75.2	64.2	81.5	74.5
QLoRA (3-bit)	3.0	55.4	42.8	60.5	51.2
AdaLoRA	4.0	76.1	65.5	82.1	75.8
AMQ + LoRA	4.3	77.2	66.8	83.0	76.8
QR-Adaptor	**3.4**	**77.8**	**67.4**	**83.6**	**77.5**

Table 2: **Mathematical Reasoning (GSM8K, 8-shot).** QR-Adaptor achieves comparable or superior performance to mixed-precision methods (AMQ) while using significantly fewer bits (3.4 vs 4.3).

instance, on Qwen-3-8B, it improves average accuracy from 67.8% to 69.0%, and on LLaMA-3-8B, it gains +1.2% accuracy over QLoRA (69.5% vs. 68.3%). Second, in the performance-focused regime, **QR-Adaptor-6** (averaging ∼5.2 bits) effectively bridges the gap to full precision. Notably, it surpasses the FP16 LoRA upper bound on both 8B models, achieving 70.6% on Qwen-3-8B (vs. 70.2%) and 71.2% on LLaMA-3-8B (vs. 70.8%). This suggests that a strategic combination of higher precision in later sensitive layers and flexible rank adaptation is more effective than uniform weights constrained by fixed adapters. Furthermore, compared to decoupled strategies like AdaLoRA and AMQ+LoRA, our joint optimization yields consistently lower perplexity on WikiText-2, validating the necessity of co-optimizing fidelity and plasticity to prevent information bottlenecks.

Mathematical Reasoning (GSM8K). Table 2 evaluates multi-step reasoning, a capability highly sensitive to quantization noise. Uniform quantization proves detrimental here; notably, 3-bit QLoRA on LLaMA-3 drops nearly 20 points compared to FP16 (55.4% vs 78.5%). In contrast, QR-Adaptor (3.4 bits) identifies and preserves the fidelity of critical arithmetic layers, recovering the majority of this performance drop to reach 77.8%. This demonstrates robust resilience where uniform compression fails, confirming that the search preserves fidelity in the layers most critical for arithmetic reasoning.

Modern Data and Stronger PEFT Baselines. To address concerns about timeliness and advanced operators, Table 3 reports focused comparisons on Qwen-3-4B. On Alpaca-52k, QR-Adaptor-6 outperforms both DoRA and QDoRA in average accuracy (65.6 vs. 64.3 and 63.1), while QR-Adaptor-4 remains competitive at a lower average bit-width. On the more recent OpenOrca subset, QR-Adaptor preserves the same Pareto advantage: QR-Adaptor-4 reaches 66.9 average accuracy with only 3.94 average bits, nearly matching FP16 LoRA (67.1) and clearly exceeding 4-bit QLoRA (65.2). Following standard calibration practice in quantization work (Frantar et al., 2023; Lin et al., 2023; Xiao et al., 2022), our main pipeline uses 1024 C4 samples; replacing them with 1024 in-domain GSM8K samples yields a further improvement from 77.5 to 78.0 on Qwen-3-4B at the same 3.4-bit budget.

Method	Avg. Bit	Avg. Rank	Avg. Acc.
(A) Alpaca-52k + Advanced PEFT			
LoRA (FP16)	16.0	16.0	65.1
DoRA (16-bit)	16.0	16.0	64.3
QLoRA (4-bit)	4.0	16.0	62.9
QDoRA (4-bit)	4.0	16.0	63.1
QR-Adaptor-4	3.6	10.4	64.1
QR-Adaptor-6	5.3	11.9	65.6
(B) OpenOrca-100k			
LoRA (FP16)	16.0	16.0	67.1
QLoRA (4-bit)	4.0	16.0	65.2
AdaLoRA	16.0	12.8	66.7
AMQ + LoRA	4.5	16.0	62.7
QR-Adaptor-4	3.94	10.24	66.9
QR-Adaptor-6	5.96	10.86	67.6

Table 3: **Focused comparisons on Qwen-3-4B.** Panel (A) adds stronger PEFT operators on Alpaca-52k, while Panel (B) evaluates a modern instruction-tuning setting using a 100k OpenOrca subset.

4.3 Efficiency Analysis

A primary critique of Neural Architecture Search (NAS) approaches is the potential computational overhead. Table 4 profiles the computational efficiency on an NVIDIA A100. Addressing the search overhead, the complete QR-Adaptor pipeline requires equivalent to merely **$\sim$0.5 standard fine-tuning epochs**. Given that the discovered configuration is static and reusable across subsequent runs, this one-time cost is negligible when amortized over the model's deployment lifecycle.

In terms of resource utilization, QR-Adaptor establishes a superior efficiency profile. By strategically allocating lower precision (e.g., 2-bit) to redundancy-heavy layers, we reduce peak VRAM to **12.8 GB**, comfortably fitting within consumer-grade hardware limits and surpassing the 14.2 GB footprint of 4-bit QLoRA. It is a known phenomenon in quantization-aware fine-tuning that lower bit-widths can decrease training speed due to the overhead of on-the-fly dequantization (converting quantized weights to BF16 for computation). Furthermore, unlike AdaLoRA, which incurs a $\sim$35% throughput penalty due to dynamic SVD computations (0.65$\times$ speed), our fixed architectural configuration maintains competitive training speeds (0.82$\times$), ensuring a significantly shorter total turn-around time for multi-epoch training.

4.4 In-depth Analysis

Beyond aggregate metrics, we analyze the internal behavior of QR-Adaptor to validate our theoretical claims regarding the *Fidelity-Plasticity Trade-off*.

Method	Search Cost (Equiv. Epochs)	Peak Mem. (GB)	Speed (vs. LoRA)	Avg. Bits
LoRA (FP16)	0.0	28.5	1.00$\times$	16.0
QLoRA (4-bit)	0.0	14.2	0.85$\times$	4.0
AdaLoRA	0.0	14.5	0.65$\times$	4.0
AMQ + LoRA	$\approx$ 4.0	14.2	0.85$\times$	4.3
QR-Adaptor	**0.5**	**12.8**	**0.82$\times$**	**3.4**

Table 4: **Efficiency Profile on LLaMA-3-8B.** Search cost is normalized to standard training epochs. QR-Adaptor achieves the lowest memory footprint (12.8 GB) with negligible search overhead compared to AMQ. Note that AMQ's search phase consumes significant computational resources ($\approx$ 4 epochs).

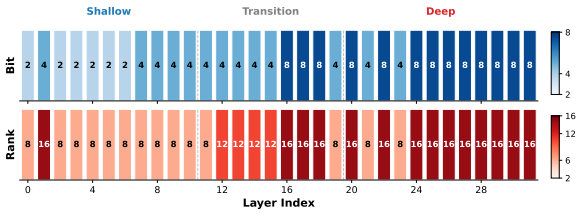

Figure 3: **Layer-wise Bit-Rank Allocation.** The discovered configuration exhibits a clear depth-wise gradient: high fidelity (bits) and plasticity (rank) are automatically concentrated in later sensitive layers, while shallow layers are aggressively compressed.

Analyzing Depth-wise Sensitivity. Figure 3 visualizes the optimal configuration (C_{best}) discovered for LLaMA-3-8B. A clear depth-wise gradient emerges autonomously: the search allocates lower precision and ranks to shallow layers (0-10), while later layers (20-32) are consistently assigned higher fidelity and plasticity. This pattern is consistent with prior probing work on representational specialization (Jawahar et al., 2019; Tenney et al., 2019), but we do not interpret it as evidence of an exclusive linguistic pipeline, especially because later layers also accumulate upstream quantization error (Niu et al., 2022; Dettmers et al., 2022b). The key empirical takeaway is that QR-Adaptor automatically identifies later layers as more fragile and reallocates resources accordingly.

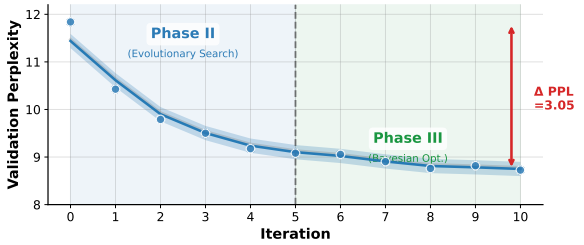

Figure 4: **Search Convergence.** Validation PPL (C4) decreases steadily, proving the effectiveness of the evolutionary exploration and Bayesian refinement.

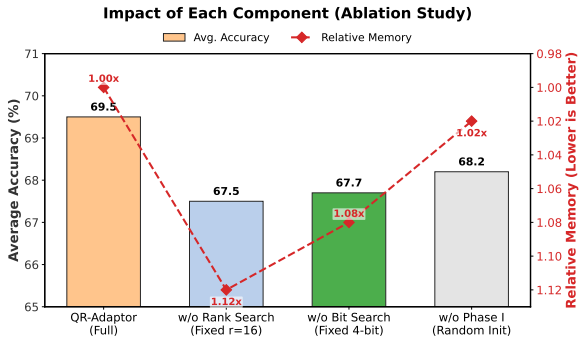

Figure 5: **Impact of Joint Optimization.** Joint optimization yields the better trade-off.

Search Convergence & Effectiveness. Figure 4 tracks the validation perplexity evolution. We observe a steep optimization trajectory during the Evolutionary Search (Phase II), validating the efficiency of our synergistic operators in navigating the discrete landscape. The subsequent Bayesian Optimization (Phase III) achieves asymptotic convergence, fine-tuning the solution to the exact Pareto limit. Notably, the final allocated bit-width exhibits a strong Pearson correlation ($r > 0.8$) with the *Fidelity Sensitivity Score* (S_l) derived in Phase I, substantiating the predictive power of our KL-based profiling as a task-agnostic prior.

4.5 Ablation Study

To verify our core hypothesis that Fidelity (bit-width) and Plasticity (adapter rank) are physically coupled, we analyze the impact of optimizing these dimensions independently versus jointly on LLaMA-3-8B. As shown in Figure 5, the results demonstrate the necessity of joint optimization. Decoupled strategies fail to navigate the trade-off: fixing the rank limits the plasticity available to later sensitive layers, while fixing the bit-width fails to exploit the memory redundancy in earlier robust layers. Furthermore, we analyze the impact of initialization. Without the task-informed prior provided by Phase I, the vast and non-convex search space leads to inefficient exploration and suboptimal convergence.

5 Conclusion

In this paper, we identify and formalize the *Fidelity-Plasticity Trade-off* in quantized fine-tuning, revealing that the adaptation potential of Large Language Models is intrinsically gated by the information capacity of their frozen weights. To navigate this constraint, we introduce QR-Adaptor, a unified framework that automates the joint optimization of quantization bit-width and adapter rank. By treating resource allocation as a multi-objective search aligned with the model's linguistic hierarchy, QR-Adaptor liberates memory from redundancy-heavy syntactic layers to reinvest in capacity-critical semantic layers. Extensive experiments on LLaMA-3 and Qwen families demonstrate that our method establishes a new Pareto frontier. Our findings suggest that the efficacy of LLM adaptation on edge devices depends not merely on the total parameter count, but on the strategic harmonization of static fidelity and dynamic plasticity.

Limitations

Although our three-stage pipeline is efficient (approx. 0.5 training epochs), it introduces a non-zero computational overhead compared to heuristic-based methods like QLoRA. While this cost is negligible when amortized over the model's deployment lifecycle—since the discovered configuration is static and reusable—it may present a bottleneck in scenarios requiring rapid, one-shot adaptation for continuously changing tasks. Future work will explore predictor-based neural architecture search (NAS) to further accelerate the profiling phase. Our current evaluation focuses on memory footprint reduction and theoretical computation reduction (FLOPs). However, realized training speedups are heavily dependent on low-level kernel implementations (e.g., mixed-precision GEMM kernels). Without specialized hardware support for layer-wise mixed bit-widths (e.g., dynamic switching between 2-bit and 4-bit operations), the wall-clock training time may not linearize perfectly with model compression. We plan to investigate hardware-co-design strategies to bridge this gap.

Acknowledgements

We sincerely thank the area chairs and anonymous reviewers for their constructive feedback, which helped improve the final version of this paper. This work was supported by the National Nature Science Foundation of China (62472097), the Shanghai Municipal Science and Technology Commission (Grant No. 25511102200), the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (Grant No. JYB2025XDXM904), and the Fudan Kunpeng & Ascend Center of Cultivation. The computations were performed on the CFFF platform of Fudan University.

References

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. 2024. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022a. Llm. int8 (): 8-bit matrix multiplication for transformers at scale. *CoRR*, abs/2208.07339.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022b. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *Preprint*, arXiv:2208.07339.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023a. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. 2023b. Spqr: A sparse-quantized representation for near-lossless llm weight compression. *arXiv preprint arXiv:2306.03078*.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. *Preprint*, arXiv:2301.00774.

Elias Frantar, Sahar Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Jun Gao, Qi Lv, Zili Wang, Tianxiang Wu, Ziqiang Cao, and Wenjie Li. 2025. Uniicl: An efficient icl framework unifying compression, selection, and generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 500–510.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and et al. Abhinav Pandey. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Han Guo, Philip Greengard, Eric Xing, and Yoon Kim. 2024. LQ-loRA: Low-rank plus quantized matrix decomposition for efficient language model finetuning. In *The Twelfth International Conference on Learning Representations*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *Preprint*, arXiv:2305.02301.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of ICLR*.

Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and XIAOJUAN QI. 2025. Slim-llm: Salience-driven mixed-precision quantization for large language models. In *Forty-second International Conference on Machine Learning*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. *Preprint*, arXiv:1909.10351.

Geonho Lee, Janghwan Lee, Sukjin Hong, Minsoo Kim, Euijai Ahn, Du-Seong Chang, and Jungwook Choi. 2025a. Rilq: Rank-insensitive lora-based quantization error compensation for boosting 2-bit large language model accuracy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18091–18100.

Sangjun Lee, Seung taek Woo, Jungyu Jin, Changhun Lee, and Eunhyeok Park. 2025b. Amq: Enabling automl for mixed-precision weight-only quantization of large language models. *Preprint*, arXiv:2509.12019.

Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo Zhao. 2023. Loftq: Lora-fine-tuning-aware quantization for large language models. *Preprint*, arXiv:2310.08659.

Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and Teknium. 2023. Openorca: An open dataset of gpt augmented flan reasoning traces. https://huggingface.co/datasets/Open-Orca/OpenOrca.

Baohao Liao, Christian Herold, Shahram Khadivi, and Christof Monz. 2024. Apiq: Finetuning of 2-bit quantized large language model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20996–21020.

Ji Lin, Jie Tang, Haotao Tang, Shuxin Yang, Xiaoxia Dang, and Song Han. 2023. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Jingcheng Niu, Wenjie Lu, and Gerald Penn. 2022. Does BERT rediscover a classical NLP pipeline? In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3143–3153, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. A simple and effective pruning approach for large language models. *Preprint*, arXiv:2306.11695.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. *Stanford CRFM*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. *Preprint*, arXiv:1905.05950.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv:2302.13971*.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2023. Dylora: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. *Preprint*, arXiv:2210.07558.

Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, and 1 others. 2023. Efficient large language models: A survey. *Transactions on Machine Learning Research*.

Xinyuan Wang, Yanchi Liu, Wei Cheng, Xujiang Zhao, Zhengzhang Chen, Wenchao Yu, Yanjie Fu, and Haifeng Chen. 2025. Mixllm: Dynamic routing in mixed large language models. *Preprint*, arXiv:2502.18482.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2022. Smoothquant: Accurate and efficient post-training quantization for large language models. *arXiv:2211.10438*.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhensu Chen, Xiaopeng Zhang, and Qi Tian. 2023. Qa-lora: Quantization-aware low-rank adaptation of large language models. *arXiv preprint arXiv:2309.14717*.

An Yang, Anfeng Li, and Baosong Yang et al. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. 2024. Atom: Low-bit quantization for efficient and accurate llm serving. *Preprint*, arXiv:2310.19102.

Changhai Zhou, Shijie Han, Lining Yang, Yuhua Zhou, Xu Cheng, Yibin Wang, and Hongguang Li. 2025a. RankAdaptor: Hierarchical rank allocation for efficient fine-tuning pruned LLMs via performance model. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5796–5810, Albuquerque, New Mexico. Association for Computational Linguistics.

Changhai Zhou, Qian Qiao, Yuhua Zhou, Yuxin Wu, Shichao Weng, Weizhong Zhang, and Cheng Jin. 2026a. Large language model compression with global rank and sparsity optimization. *Preprint*, arXiv:2505.03801.

Changhai Zhou, Shiyang Zhang, Yuhua Zhou, Qian Qiao, Jun Gao, Cheng Jin, Kaizhou Qin, and Weizhong Zhang. 2026b. Autoqra: Joint optimization of mixed-precision quantization and low-rank adapters for efficient llm fine-tuning. *Preprint*, arXiv:2602.22268.

Changhai Zhou, Yuhua Zhou, Shijie Han, Qian Qiao, and Hongguang Li. 2024. Qpruner: Probabilistic decision quantization for structured pruning in large language models. *Preprint*, arXiv:2412.11629.

Yuhua Zhou, Ruifeng Li, Changhai Zhou, Fei Yang, and Aimin PAN. 2025b. BSLoRA: Enhancing the parameter efficiency of loRA with intra-layer and inter-layer sharing. In *Proceedings of International Conference on Machine Learning*.

Yuhua Zhou, Changhai Zhou, Shiyang Zhang, Fei Yang, Yi Zhang, and Aimin Pan. 2026c. Lara: Layer-wise rank allocation for efficient fine-tuning of pruned large language models. *Information Processing & Management*, 63(3):104538.