D-QRELO: Training- and Data-Free Delta Compression for Large Language Models via Quantization and Residual Low-Rank Approximation

Junlin Li[1] **Shuangyong Song**[2] **Guodong Du**[3]✉ **Ngai Wong**[4]
Xuebo Liu[1] **Yongxiang Li**[2] **Min Zhang**[1] **Jing Li**[1] **Xuelong Li**[2]✉
[1]Harbin Institute of Technology, Shenzhen, China [2]TeleAI of China Telecom
[3]The Hong Kong Polytechnic University [4]The University of Hong Kong
leejunlin27@gmail.com jingli.phd@hotmail.com

Abstract

Supervised Fine-Tuning (SFT) accelerates task-specific large language models (LLMs) development, but the resulting proliferation of fine-tuned models incurs substantial memory overhead. Delta compression addresses this by retaining a single pre-trained LLM with multiple compressed delta weights. However, existing methods fail on models fine-tuned with large-scale datasets. We find that larger SFT data scale amplifies delta parameter magnitude, singular values, and entropy, exacerbating compression errors. To tackle this, we propose D-QRELO (**D**elta Compression via **Q**uantization and **Re**sidual **Lo**w-Rank), a novel training- and data-free delta compression method. It combines coarse-grained one-bit quantization to capture the dominant structure of the delta, followed by compensated residual low-rank approximation to recover fine-grained details from the smaller residual error. Experiments on various LLMs spanning dense and MoE architectures across multiple domains under this challenging setting demonstrate that D-QRELO outperforms existing methods. Moreover, we establish key design principles for delta compression through extensive empirical analysis, demonstrating how task difficulty, architecture, and layer positioning create predictable patterns that can guide optimal compression strategies in production systems.

1 Introduction

Large Language Models (LLMs) (Touvron et al., 2023; Achiam et al., 2023; Du et al., 2025) have recently become a cornerstone in AI, fundamentally reshaping how various downstream tasks are approached. Leveraging fine-tuning, pre-trained LLMs now yield diverse specialized models (Hui et al., 2024; Lee et al., 2024; Du et al., 2026). This paradigm significantly lowers model customization barriers, accelerating AI deployment across industries. However, this flexibility paradigm introduces significant engineering challenges. Real-world applications often demand simultaneously storing and deploying numerous specialized models for complex needs (Li et al., 2025a; Farr et al., 2024; Lee et al., 2024). For example, it is essential to store every iterative model version to facilitate rollbacks, as well as to cooperatively deploy multiple specialized models in multi-stage reasoning or multi-agent collaboration tasks (Chan et al., 2024).

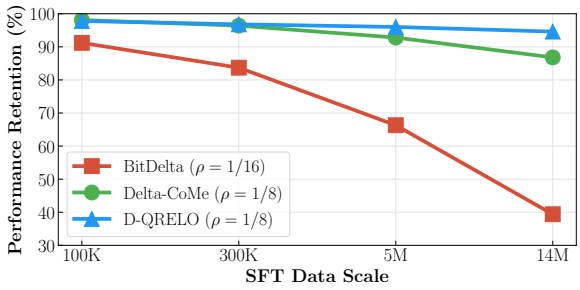

Figure 1: Impact of SFT data scale on delta compression for the same base LLM. ρ is the compression ratio.

Data Size	Avg($\lvert\Delta\rvert$)	Avg($\lvert\sigma\rvert$)	Avg($\lvert H(\Delta)\rvert$)
100K	0.000492	0.0181	8.932
300K	0.001052	0.0553	9.348
5M	0.001489	0.0967	9.617
14M	0.002147	0.1266	9.839

Table 1: Impact of SFT data scale on delta parameters magnitude (Δ), singular values (σ), entropy ($H(\Delta)$).

Recently, delta compression (Yao et al., 2025; Yang et al., 2025) has emerged to tackle these challenges by exploiting redundancy in delta parameters—the differences between fine-tuned model and pre-trained model weights. Its goal is to store only a single pre-trained model alongside multiple compressed delta parameter sets, thereby reducing the memory footprint for storing and deploying numerous fine-tuned models. For instance, BitDelta (Liu et al., 2024a) compresses delta parameters to one-bit quantization. Delta-CoMe (Ping et al., 2024) further improves compression efficiency by lever-

✉ Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2026, pages 21507–21518
July 2-7, 2026

aging mixed-precision SVD of delta parameters.

However, these methods show a significant performance drop when applied to powerful SFT models (e.g., Qwen3, DeepSeek-R1-Distill). To understand this, we analyze these models and find their delta parameters magnitudes are consistently larger (mean absolute values: 0.002-0.004) compared to the LLMs these methods originally evaluated (mostly $<$0.0008). This highlights an acknowledged but critically underappreciated issue (Yu et al., 2024): the magnitude of the delta parameters profoundly impacts compression performance.

To delve deeper into this phenomenon, we conduct an extensive exploration: under the same experimental setup, we fine-tune LLaMA 3.1 8B with OpenMathInstruct-2 (Toshniwal et al., 2025) datasets of 100K, 300K, 5M, and 14M samples, respectively. Table 1 presents the magnitude of delta parameters, their singular values and entropy. Figure 1 shows the performance of various methods on these models. Key observations follow:

- Under same setting, more fine-tuning data leads to larger magnitude of delta parameters, their singular values, and their entropy, indicating that the delta parameters are becoming increasingly complex and information-rich.
- As these delta properties intensify, the performance of prior methods declines significantly.

This performance degradation stems from the inherent limitations of quantization and SVD, which incur larger errors when handling delta parameters with greater numerical scales and complexity.

Given these critical insights, we propose D-QRELO, a delta compression method combining quantization and compensated residual low-rank approximation. D-QRELO begins with coarse-grained bit quantization to capture the dominant signs and initial magnitude information of the delta parameters. This step serves as a fast, adaptive encoding mechanism that efficiently handles large-magnitude values. Subsequently, we observe that the residual error exhibits a substantially narrower numerical range than the original delta parameters, rendering it well-suited for efficient low-rank approximation via SVD. By applying SVD to this residual, we can precisely reconstruct the fine-grained details lost during the initial quantization. Crucially, unlike prior methods, D-QRELO achieves both training- and data-free compression, which enhances its generality and efficiency.

Extensive experiments on powerful SFT models spanning dense and MoE architectures trained with large-scale data show that D-QRELO consistently outperforms prior methods at the same compression ratio, advancing the Pareto frontier of delta compression and demonstrating superior efficiency-performance trade-offs. These models cover tasks in reasoning, alignment, domain knowledge, and multimodal understanding, highlighting D-QRELO's strong generalization. Evaluations show that D-QRELO yields multi-fold savings in GPU memory, along with significant inference speedup. Furthermore, to explore key design principles for delta compression, we conduct a comprehensive empirical study grounded in D-QRELO.

The main contributions of this paper include:

- We provide insights into the impact of SFT data scale on delta compression efficiency.
- We propose a novel delta compression method, D-QRELO, that combines quantization and residual low-rank approximation.
- We conduct the first comprehensive empirical study of delta compression.

2 Related Work

Delta-Compression. Delta compression methods are typically categorized into pruning, quantization, and low-rank approximation. Pruning methods (Du et al., 2024; Liu et al., 2024b; Li et al., 2025b) like DARE (Yu et al., 2024) reduce model size by eliminating redundant delta parameters while maintaining performance. Quantization approaches, known for their efficiency and hardware friendliness, compress delta parameters into low-bit formats. GPT-Zip (Isik et al., 2023) pioneers 2-bit quantization paired with sparsification to compress delta weights while maintaining performance. BitDelta (Liu et al., 2024a) pushes compression further by quantizing delta parameters down to one-bit, and refines scale factors through knowledge distillation. Low-rank approximation methods (Ryu et al., 2023) like Delta-CoMe (Ping et al., 2024) employ mixed-precision quantization that adapts bit-width allocation according to the singular value spectrum of delta matrices. However, large-scale fine-tuning on massive datasets often amplifies the magnitude of delta parameters, breaking prior methods' assumptions about their value range and error distribution—resulting in suboptimal compression. To address this, we propose D-QRELO, designed to robustly handle the expanded

numerical range of delta parameters in powerful SFT models.

Quantization and Low Rank Approximation. Quantization (Lin et al., 2024) and low-rank (Li et al., 2023) approximation are complementary techniques that are often combined to enhance the compression of LLM. For example, CALDERA (Saha et al., 2024) decomposes a weight matrix as $W \approx Q + LR$, where Q is a quantized full-rank matrix and LR is a low-precision low-rank term, optimized alternately by quantizing $W - LR$ and factorizing $W - Q$. LQ-LoRA (Guo et al., 2024) adopts a low-rank plus quantized matrix decomposition, where the quantized component is kept fixed during fine-tuning while only the low-rank component is updated. Additionally, LQER (Zhang et al., 2024) introduces an activation-induced scale matrix to optimize the singular value distribution of the quantization error, effectively restoring LLM performance. In this work, we pioneer the first method to combine quantization with residual low-rank approximation, specifically tailored for delta parameters compression.

3 Methodology

3.1 Preliminaries

Given a scenario in which a pre-trained LLM ϕ_{Pre} is independently fine-tuned for a set of downstream tasks $\{\mathcal{T}_i\}_{i=1}^{K}$. This yields a collection of fine-tuned models $\{\phi^{(i)}\}_{i=1}^{K}$, each customized for a specific task. Directly storing all K full-precision models incurs a storage cost of $K \times \mathcal{M}$, where $\mathcal{M}$ denotes the number of parameters in the full model. To reduce this overhead, we first represent each fine-tuned LLM as the combination of the pre-trained LLM ϕ_{Pre} and its corresponding delta parameters Δ, which are defined as:

$$\Delta^{(k)} = \phi^{(k)} - \phi_{Pre}, \quad k = 1, 2, \ldots, K, \tag{1}$$

where delta parameters $\Delta^{(k)}$ captures the task-specific deviation from the pre-trained LLM.

We then apply a delta compression function $\mathcal{C}(\cdot; \rho)$ to transform each $\Delta^{(k)}$ into compressed delta parameters $\hat{\Delta}^{(k)}$:

$$\hat{\Delta}^{(k)} = \mathcal{C}(\Delta^{(k)}; \rho), \tag{2}$$

where ρ denotes the compression ratio. The decompression procedure reconstructs an approximate fine-tuned LLM as:

$$\hat{\phi}^{(k)} = \phi_{Pre} + \hat{\Delta}^{(k)}. \tag{3}$$

By storing a single pre-trained LLM ϕ_{Pre} with a set of compressed delta parameters $\{\hat{\Delta}^{(1)}, \ldots, \hat{\Delta}^{(K)}\}$, this framework reduces total storage to $(1 + \rho K) \times \mathcal{M}$.

3.2 Motivation

As shown in Figure 1 and Table 1, the performance of prior methods degrades as the magnitude of delta parameters and their singular values increases, due to their struggle in maintaining sufficient accuracy on larger numerical scales.

For a delta parameters (Δ_1) approximated by its top-k singular values ($(\Delta_1)_k$), the Frobenius norm of the reconstruction error is defined as:

$$\|\Delta_1 - (\Delta_1)_k\|_F^2 = \sum_{i=k+1}^{\text{rank}(\Delta_1)} \sigma_{1,i}^2 \tag{4}$$

If a delta parameters Δ_2 possesses singular values $\sigma_{2,i}$ consistently larger than Δ_1's (e.g., $\sigma_{2,i} = c \cdot \sigma_{1,i}$ for $c > 1$), then, even with the same k singular values retained, its absolute reconstruction error will be significantly larger:

$$\begin{aligned} \|\Delta_2 - (\Delta_2)_k\|_F^2 &= \sum_{i=k+1}^{\text{rank}(\Delta_2)} \sigma_{2,i}^2 = \sum_{i=k+1}^{\text{rank}(\Delta_2)} (c \cdot \sigma_{1,i})^2 \\ &= c^2 \sum_{i=k+1}^{\text{rank}(\Delta_1)} \sigma_{1,i}^2 = c^2 \|\Delta_1 - (\Delta_1)_k\|_F^2 \end{aligned} \tag{5}$$

Thus, SVD-based methods incur a greater absolute error on larger-magnitude delta parameters for a given compression ratio. Similarly, for the one-bit quantization method BitDelta, the absolute quantization error scales proportionally with the magnitude of the original values. For an element x quantized to x_q, the absolute error is $|x - x_q|$. Consequently, mapping a larger magnitude x to fixed discrete points (e.g., -1 or 1) results in a proportionally larger absolute deviation.

In summary, neither method mitigates magnitude related error accumulation effectively. To resolve this limitation, we introduce D-QRELO, a novel two-stage residual-calibrated framework targeting the core cause of performance degradation.

3.3 D-QRELO

Our method D-QRELO is illustrated in Figure 2. Given the delta parameters $\Delta \in \mathbb{R}^{n \times m}$, D-QRELO's goal is to efficiently compress Δ with a target compression ratio ρ.

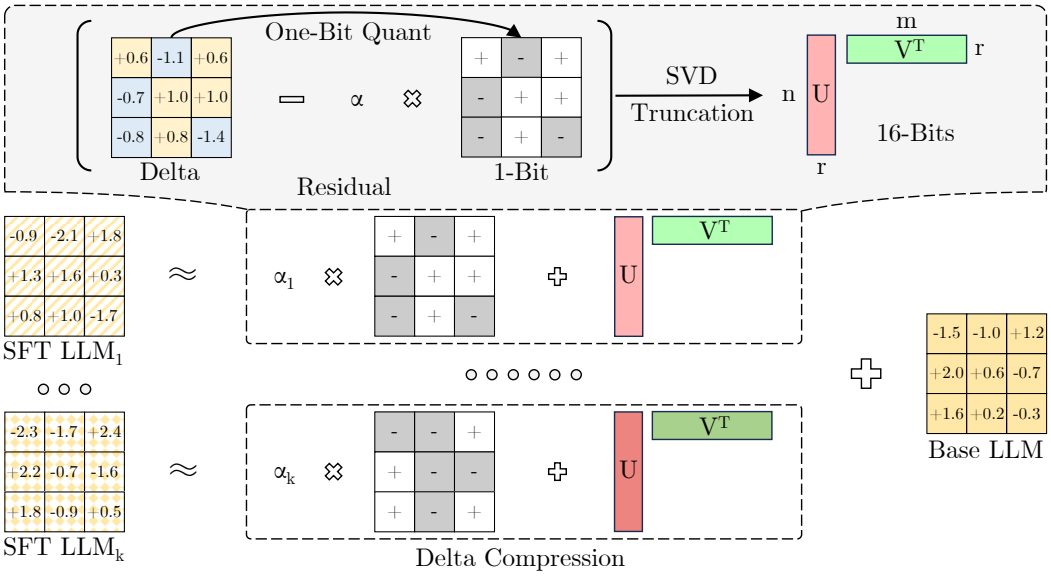

Figure 2: Overview of D-QRELO. D-QRELO involves an initial one-bit quantization of the delta parameters, followed by SVD truncation applied to the resulting residuals to derive a low-rank matrix.

D-QRELO first applies a coarse-grained one-bit quantization to Δ to capture essential directional properties and initial magnitude scale:

$$\hat{\Delta}_{\text{quant}} = \alpha \odot \text{Sign}(\Delta), \tag{6}$$

where the element-wise sign function is defined as:

$$\text{Sign}(\Delta_{ij}) = \begin{cases} +1, & \Delta_{ij} > 0, \\ -1, & \Delta_{ij} \leq 0, \end{cases} \tag{7}$$

and the scaling factor α is determined as the mean absolute value of Δ:

$$\alpha = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} |\Delta_{ij}|, \tag{8}$$

This choice of α minimizes the Frobenius-norm quantization error between Δ and its one-bit quantized approximation $\hat{\Delta}_{\text{quant}}$:

$$\alpha = \arg\min_{\alpha} \|\Delta - \alpha \cdot \text{Sign}(\Delta)\|_F^2. \tag{9}$$

The remaining fine-grained information and the error introduced by coarse quantization are preserved in the residual matrix R:

$$R = \Delta - \hat{\Delta}_{\text{quant}}. \tag{10}$$

Thanks to the effective initial quantization, R's numerical range is significantly smaller than that of the original Δ, making it suitable for efficient low-rank approximation.

D-QRELO then computes the SVD of residual matrix R:

$$R = U \Sigma V^{\top}, \tag{11}$$

where $U \in \mathbb{R}^{n \times n}$, $\Sigma \in \mathbb{R}^{n \times m}$ is diagonal with singular values $\sigma_{R,1} \geq \sigma_{R,2} \geq \cdots$, and $V \in \mathbb{R}^{m \times m}$.

To reduce dimensionality, D-QRELO truncates the SVD of R to rank

$$r = \left\lceil \frac{n \cdot m \cdot \rho_1}{n + m} \right\rceil, \tag{12}$$

choosing r so that the low-rank parameters $r(n + m) + r$ meet the target compression ratio ρ_1 of the original nm parameters, the additional r parameters are negligible for large n, m.

D-QRELO then extracts the truncated matrices:

$$U_r = U[:, :r], \quad \Sigma_r = \Sigma[:r, :r], \quad V_r = V[:, :r]. \tag{13}$$

This yields a rank-r approximation of the residual R as:

$$\hat{R}_{\text{lowrank}} = U_r \Sigma_r (V_r)^{\top}. \tag{14}$$

The final compressed delta parameters $\hat{\Delta}$ combines the quantization and residual low-rank approximation:

$$\hat{\Delta} = \hat{\Delta}_{\text{quant}} + \hat{R}_{\text{lowrank}}. \tag{15}$$

The overall compression ratio ρ combines the low-rank component ρ_1 and the one-bit quantization contribution:

$$\rho = \rho_1 + \frac{1}{b} \tag{16}$$

Method	OpenMath2		Qwen2-VL		DS-R1-Qwen3-8B		Qwen3-4B-Instruct			Qwen3-30B-A3B-Instruct			Drop
	GSM8K	MATH	TextVQA	MME	AIME24	GPQA	MMLU	LCB	IFEval	MMLU	LCB	IFEval	
Backbone	5.45	2.6	-	-	7.78	27.61	18.25	10.00	34.57	58.74	16.79	39.37	-
SFT	88.93	61.2	84.51	1676	45.56	61.28	63.25	30.40	82.23	74.13	40.46	83.18	0%
Random	9.70	5.6	9.12	510	3.17	31.97	38.12	9.65	48.33	60.09	18.81	50.66	86.12%
Magnitude	45.56	25.2	17.37	871	18.67	41.49	52.75	13.62	71.89	67.16	22.67	74.21	53.25%
Wanda	50.34	27.6	15.79	784	20.13	40.34	50.91	15.24	73.15	66.31	24.83	76.08	51.79%
SVD	69.82	33.8	72.22	1451	21.50	47.31	54.05	16.08	75.42	69.45	23.04	76.30	35.62%
BitDelta	39.19	21.2	73.87	1525	8.13	39.28	47.98	10.69	67.76	64.44	14.37	71.92	56.13%
Delta-CoMe	80.50	50.4	75.69	1569	37.83	57.06	61.47	23.13	**80.51**	72.24	32.72	82.51	14.00%
D-QRELO	**84.84**	**57.4**	**81.94**	**1664**	**42.33**	**59.65**	**62.71**	**24.91**	78.76	**75.88**	**35.09**	**83.73**	**6.17%**

Table 2: The performance of different delta-compression methods on five LLMs. LCB denotes LiveCodeBench. Optimal results are bolded, sub-optimal ones are underlined. Drop means the percentage of performance degradation.

where b denotes the number of bits per full-precision weight (e.g., $b = 16$ for FP16/BF16 precision). For vector parameters, we select parameters for compression by magnitude at a target ratio of ρ.

4 Experimental Setup

Baselines. We evaluate several baselines: pruning methods (random, magnitude, Wanda (Sun et al., 2024)), SVD, BitDelta (Liu et al., 2024a), and Delta-CoMe (Ping et al., 2024). All methods are evaluated at a 1/8 compression ratio, as D-QRELO's design requires a ratio greater than 1/16. BitDelta is the only exception, evaluated at 1/16 due to its 1-bit quantization under FP16/BF16.

Tasks. We evaluate a set of tasks and datasets spanning various domains and levels of complexity:

- Math: GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021) and AIME2024 span tasks from basic arithmetic to advanced competition-level problems.
- Code: LiveCodeBench v6 (Jain et al., 2024), HumanEval (Chen et al., 2021) evaluate natural language to executable code generation.
- Knowledge: GPQA-Diamond (Rein et al., 2024), MMLU-Pro (Wang et al., 2024) assess multi-disciplinary knowledge.
- Alignment: IFEval (Zhou et al., 2023) evaluates instruction-following capability.
- Multi-Modal Chat: MME (Fu et al., 2024) and TextVQA (Singh et al., 2019), evaluating joint visual-text understanding and reasoning.

Models. We evaluate eight models across three tuning paradigms (standard/multimodal SFT, reasoning distillation) and two architecture types (dense, MoE). All fine-tuned on large-scale datasets and exhibit large delta parameters (average absolute value: 0.0024), details are in Table 7.

5 Experimental Results

5.1 Main Results

Our results in Table 2 showcase the delta compression performance of all methods on five powerful SFT LLMs. Three additional models' results are included in Appendix C. D-QRELO almost consistently shows superior performance retention across all evaluated tasks and models, achieving a 7.83% gain over the current SOTA method Delta-CoMe.

Firstly, we observe that for delta parameters with large magnitudes, pruning (Random, Magnitude, Wanda) and low-rank (SVD) methods yield substantial performance drops (35.62%–86.12%).

Secondly, while BitDelta and Delta-CoMe achieve near-lossless performance in their original experiments, their efficacy significantly degrades when applied to these extensively trained SFT models. In contrast, D-QRELO markedly improves upon both, retaining about 94% of the original performance. This highlights how its two-stage approach effectively mitigates the absolute error accumulation problem inherent in other methods.

Finally, D-QRELO 's outstanding performance across diverse tasks and LLM architectures underscores its superior generalization ability. This confirms the robustness of its magnitude and residual processing approach for various fine-tuned models.

Method	OpenMath2		Qwen2-VL		DS-R1-Qwen3-8B		Qwen3-4B-Instruct			Qwen3-30B-A3B-Instruct			Drop
	GSM8K	MATH	TextVQA	MME	AIME24	GPQA	MMLU	LCB	IFEval	MMLU	LCB	IFEval	
SVD	43.97	26.8	57.19	1280	13.55	37.27	43.71	11.06	58.11	64.80	19.34	65.08	58.71%
+ One-bit	80.64	50.8	80.76	1652	**42.71**	58.53	62.49	24.37	**78.85**	74.52	**35.26**	82.92	9.01%
+ SVD	69.82	33.8	72.22	1451	21.50	47.31	54.05	16.08	75.42	69.45	23.04	76.30	35.62%
One-bit	39.19	21.2	73.87	1525	8.13	39.28	47.98	10.69	67.76	64.44	14.37	71.92	56.13%
+ SVD	**84.84**	**57.4**	**81.94**	**1664**	42.33	**59.65**	**62.71**	**24.91**	78.76	**75.88**	35.09	**83.73**	**6.17%**

Table 3: Ablation study of D-QRELO's two-stage strategy. Each method first applies SVD or one-bit quantization to the delta parameters ($\rho = 1/16$). The + represent compression of the residuals by SVD or one-bit quantization.

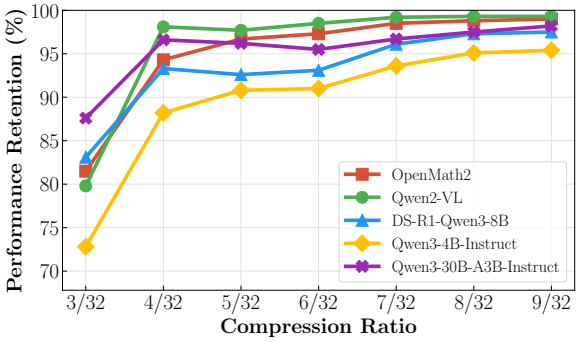

Figure 3: Performance retention of D-QRELO across models with varying residual SVD compression ratios.

# Models	Memory (GB) (× / ✓) D-QRELO	Latency (ms) (× / ✓) D-QRELO
2	31.78 / 20.91	63 / 45
4	63.12 / 26.88	129 / 52
8	OOM / 38.94	- / 67
16	OOM / 62.87	- / 91

Table 4: GPU memory cost and inference latency.

5.2 Ablation Study

Table 3 validates the effectiveness and strategic ordering of D-QRELO's two-stage compression strategy. While standalone SVD or one-bit quantization individually lead to significant performance degradation on large-magnitude delta parameters, their combination dramatically mitigates this decline. Moreover, SVD proves more effective than quantization for compressing the residual. This is because initial one-bit quantization effectively reduces the original delta parameters' numerical scale, yielding a smaller-magnitude residual optimally suited for SVD to capture its remaining structured information. Conversely, SVD's residual retains high-frequency components that one-bit quantization struggles to compress accurately.

Figure 3 shows D-QRELO's performance retention as the compression ratio varies, with one-bit quantization held constant. We observe that performance retention generally improves with the higher compression ratio, often plateauing at higher ratios. However, this trend is not strictly monotonic. For example, on Qwen3-30B-A3B-Instruct, increasing the ratio from 4/32 to 6/32 causes a performance drop. This is because a higher compression ratio introduces more residual information, which may capture instability and temporarily impact performance. This suggests that information recovery is nonlinear and influenced by residual complexity.

5.3 Inference Speed and Memory Cost

Table 4 compares the GPU memory and decoding latency for multiple LLaMA 3.1-8B alignment models (BF16). On a single 80G GPU, the baseline deployment lacks the capacity for even 8 models. In contrast, D-QRELO (1/8 compression ratio) drastically reduces memory overhead and significantly lowers decoding latency. Moreover, the more models deployed, the more significant the effectiveness of D-QRELO.

6 Delta-Compression Analysis

6.1 Impacts of Task Difficulty

Our results, detailed in Figure 4 and Figure 5, confirm that task difficulty directly impacts delta compression performance, with more challenging tasks consistently leading to greater performance drops across distinct mathematical benchmarks. For instance, OpenMath2 sees a slight decline from GSM8K to MATH, while DS-R1-LLaMA-8B's performance significantly decreases from MATH to AIME24. This trend is further supported by a fine-grained analysis within MATH: increasing difficulty levels (1 to 5) generally correlate with reduced performance retention, most notably for DS-R1-LLaMA-8B. Although DS-R1-Qwen remain

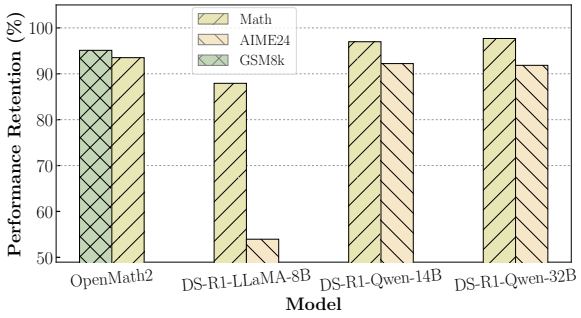

Figure 4: Performance retention of D-QRELO on mathematical reasoning tasks of varying difficulty.

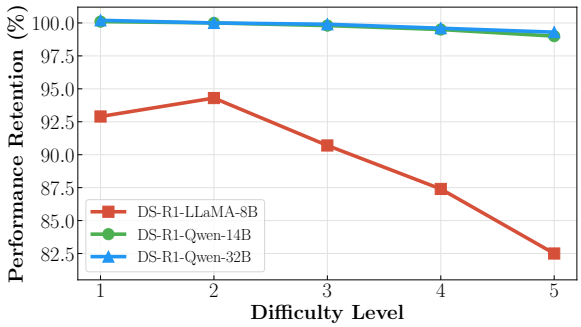

Figure 5: Performance retention of D-QRELO on different difficulty levels on MATH.

remarkably robust, the overall pattern highlights that the subtle and complex information required for harder tasks is more susceptible to delta compression's approximation errors.

6.2 Impacts of the Base Pre-trained LLMs

Figure 6 shows how the pre-trained model's origin and scale impact delta compression performance. A stark contrast emerges between LLM families, as Qwen consistently outperforms LLaMA in performance retention across all tasks. This suggests that the Qwen base architecture, fine-tuning settings, or its pre-training characteristics, yield delta parameters inherently more structured and robust to D-QRELO's compression. Moreover, within a family, model scale generally benefits from compressibility. Comparing DS-R1-Qwen variants (7B–32B), larger models generally maintain slightly better performance retention, especially on GPQA. Hence, larger base models, with their increased capacity, more robust internal representations, and potentially greater parameter redundancy, may learn delta parameters more amenable to compression.

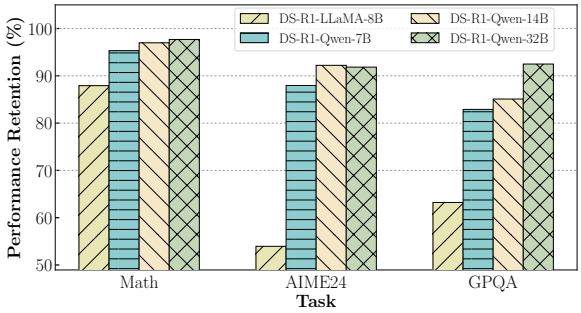

Figure 6: Performance retention of D-QRELO across diverse base LLM architectures and sizes.

Method	OpenMath2-LLaMA3.1-8B		Avg.
	GSM8K	MATH	
Backbone	5.45	2.6	4.03
SFT	88.93	61.2	75.07
LoRA	74.37	49.8	62.09
SVD	69.82	33.8	51.81
D-QRELO	84.84	57.4	71.12

Table 5: LoRA vs. delta-compression.

6.3 Delta-Compression vs. Delta-Tuning

Delta compression is closely related to delta tuning, like LoRA (Hu et al., 2022). While delta tuning aims to reduce LLM fine-tuning costs, delta compression focuses on lowering storage and inference costs for multi-model serving. We compare their performance by fine-tuning a LoRA model on the same dataset as OpenMath2, detailed configuration of LoRA refer to appendix A. Table 5 shows that SVD delta compression, despite mirroring LoRA's architecture and parameter size, performs significantly worse than the LoRA model. In contrast, D-QRELO notably outperforms the LoRA model, closely matching the SFT model's performance. This suggests that for building high-performance, compact models, fully fine-tuning and then applying a well-designed delta compression method is superior to solely relying on delta tuning.

6.4 Layer-wise Analysis

To understand the sensitivity and robustness of different layers to delta compression, we perform an ablation study by applying D-QRELO to specific quarters of the model's layers (0%-25%, 25%-50%, 50%-75%, 75%-100%), leaving the rest uncompressed. Figure 7 highlights clear patterns in layer importance and compressibility.

Firstly, early layers are more sensitive to compression. For most models, compressing early lay-

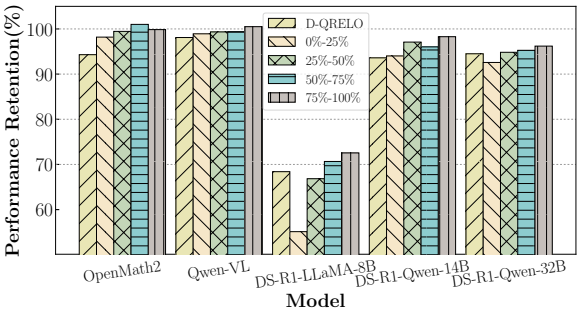

Figure 7: Layer sensitivity of delta compression. xx%-yy% denotes that only the delta parameters in the first xx% to yy% of the model's layers are compressed.

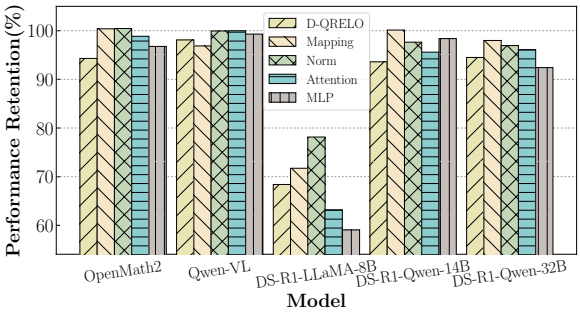

Figure 8: Module sensitivity of delta compression.

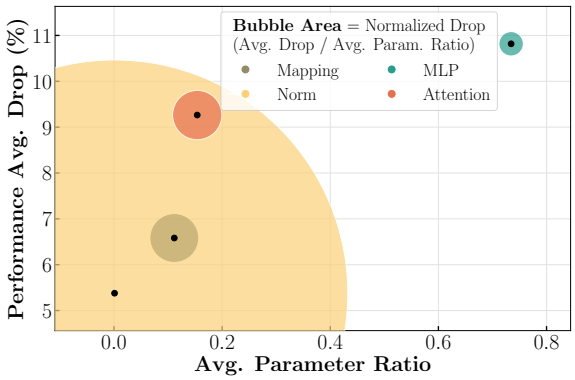

Figure 9: Module importance in delta compression: Performance drop as a function of parameter ratio, where bubble size reflects normalized impact.

ers causes a noticeable performance drop compared to compressing later segments. For instance, DS-R1-LLaMA-8B suffers a significant performance drop when only its initial layers are compressed. This suggests that delta parameters in the initial layers encode more critical, less redundant information, making them essential for preserving overall model performance and revealing a potential information bottleneck concentrated in early layers.

Secondly, later layers exhibit greater robustness and redundancy. Compressing only these layers often achieves remarkably high performance retention, at times even surpassing that of full compression. This suggests that delta parameters in deeper layers may exhibit greater redundancy or adapt more compressibly, with minimal performance loss—highlighting their inherent robustness.

6.5 Module-wise Analysis

To identify which modules of delta parameters are most sensitive to compression, we conduct an ablation study, applying D-QRELO selectively to different functional components: Mapping layers (token embedding and LM head), Normalization layers, Attention modules, and MLP modules. Figure 8 shows the performance retention for each module type, while Figure 9 summarizes the average performance drop relative to parameter ratio.

Firstly, module sensitivity to compression varies significantly across models. DS-R1-LLaMA-8B stands out dramatically, exhibiting consistently lower performance retention, even when only a small fraction of its delta parameters (i.e., Normalization layers) are compressed. This is substantially lower than the near-perfect retention observed for Normalization layers in other models.

Secondly, MLP modules are the elephant in the room. They make up the vast majority of delta parameters and thus cause the largest absolute average performance drop. However, their moderate normalized drop indicates that while they are undeniably crucial, the performance decline is largely proportional to their size.

Thirdly, Normalization layers are the hidden vulnerability. Though they account for a negligible parameter ratio, they cause an exceptionally high normalized drop. This highlights that, despite their small size, Normalization layers are disproportionately sensitive to compression.

Finally, Mapping and Attention modules also exhibit a significant impact per parameter, with normalized drops second only to Normalization layers. This confirms their substantial sensitivity: despite being smaller than MLP modules, their impact per parameter compressed is pronounced.

7 Conclusion

In this paper, we investigate why previous delta compression methods suffer significant performance drops on powerful SFT models, especially those fine-tuned with large-scale datasets. We find that larger SFT data scale leads to increased magnitudes of delta parameters and their singular values, amplifying compression errors. To tackle this, we

propose D-QRELO, a training- and data-free delta compression method combining one-bit quantization with residual low-rank approximation. Experiments on several LLMs under this challenging setting show that D-QRELO outperforms key baselines. We also present a comprehensive empirical study of delta compression based on D-QRELO.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (62476070), Shenzhen Science and Technology Program (JCYJ20241202123503005, GXWD20231128103232001, ZDSYS20230626091203008, KQTD20240729102154066), Department of Science and Technology of Guangdong (2024A1515011540) and National Key R&D Program of China (SQ2024YFE0200592).

Limitations

Firstly, D-QRELO is constrained by a physical lower bound on its compression ratio because of its two-stage design. The total compression ratio ρ is determined by the sum of the residual low-rank component ρ_1 and the one-bit quantization contribution $1/b$. For models using standard 16-bit precision (e.g., FP16 or BF16), the total ratio cannot be lower than $1/16$, which is approximately 6.25%, even if the residual component is negligible.

Secondly, the relative advantages of D-QRELO tend to diminish when the SFT data scale is relatively small. Under these conditions, the resulting delta parameters are often sparse and exhibit smaller magnitudes. Hence, traditional and simpler compression methods, such as basic SVD or quantization, may already achieve high performance retention. Therefore, the added structural complexity of D-QRELO might not offer significant performance gains over more straightforward baselines.

Ethical Considerations

Our research is conducted using publicly available and safe datasets and models. However, we explicitly acknowledge that the applicability of our D-QRELO and findings may be limited to datasets or domains similar to those studied. The performance of our approach on other specific datasets or domains remains uncertain, and there may be potential risks when applying it to privacy-sensitive or high-risk scenarios. Furthermore, the generalizability of our findings to real-world applications may require further exploration and testing. Therefore, caution is advised, and thorough verification is necessary to ensure the method generates accurate and reliable results in such contexts.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. Chateval: Towards better llm-based evaluators through multi-agent debate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Guodong Du, Zitao Fang, Jing Li, Junlin Li, Runhua Jiang, Shuyang Yu, Yifei Guo, Yangneng Chen, Sim Kuan Goh, Ho-Kin Tang, Daojing He, Honghai Liu, and Min Zhang. 2025. Neural parameter search for slimmer fine-tuned models and better transfer. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim Kuan Goh, Ho-Kin Tang, Daojing He, and Min Zhang. 2024. Parameter competition balancing for model merging. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Guodong Du, Zhuo Li, Xuanning Zhou, Junlin Li, Zesheng Shi, Wanyu Lin, Ho-Kin Tang, Xiucheng Li, Fangming Liu, Wenya Wang, Min Zhang, and Jing Li. 2026. Knowledge fusion of large language models via modular skillpacks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

David Farr, Nico Manzonelli, Iain Cruickshank, Kate Starbird, and Jevin West. 2024. Llm chain ensembles for scalable and accurate data annotation. In *2024 IEEE International Conference on Big Data (BigData)*, pages 2110–2118.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2024. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.

Han Guo, Philip Greengard, Eric P. Xing, and Yoon Kim. 2024. Lq-lora: Low-rank plus quantized matrix decomposition for efficient language model finetuning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

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 the International Conference on Learning Representations (ICLR)*.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Berivan Isik, Hermann Kumbong, Wanyi Ning, Xiaozhe Yao, Sanmi Koyejo, and Ce Zhang. 2023. Gpt-zip: Deep compression of finetuned large language models. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Junlin Lee, Yequan Wang, Jing Li, and Min Zhang. 2024. Multimodal reasoning with multimodal knowledge graph. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Junlin Li, Guodong Du, Jing Li, Sim Kuan Goh, Wenya Wang, Yequan Wang, Fangming Liu, Ho-Kin Tang, Saleh Alharbi, Daojing He, and Min Zhang. 2025a. Multi-modality expansion and retention for llms through parameter merging and decoupling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Junlin Li, Guodong Du, Jing Li, Sim Kuan Goh, Wenya Wang, Yequan Wang, Fangming Liu, Ho-Kin Tang, Saleh Alharbi, Daojing He, and Min Zhang. 2025b. Multi-modality expansion and retention for llms through parameter merging and decoupling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 30866–30887. Association for Computational Linguistics.

Yixiao Li, Yifan Yu, Qingru Zhang, Chen Liang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2023. Losparse: Structured compression of large language models based on low-rank and sparse approximation. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of the Annual Conference on Machine Learning and Systems (MLSys)*.

James Liu, Guangxuan Xiao, Kai Li, Jason D. Lee, Song Han, Tri Dao, and Tianle Cai. 2024a. Bitdelta: Your fine-tune may only be worth one bit. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Jing Liu, Ruihao Gong, Mingyang Zhang, Yefei He, Jianfei Cai, and Bohan Zhuang. 2024b. Meswitch: A memory-efficient expert switching framework for large language models. *arXiv preprint arXiv:2406.09041*.

Bowen Ping, Shuo Wang, Hanqing Wang, Xu Han, Yuzhuang Xu, Yukun Yan, Yun Chen, Baobao Chang, Zhiyuan Liu, and Maosong Sun. 2024. Delta-come: Training-free delta-compression with mixed-precision for large language models. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling (COLM)*.

Simo Ryu, Seunghyun Seo, and Jaejun Yoo. 2023. Efficient storage of fine-tuned models via low-rank approximation of weight residuals. *arXiv preprint arXiv:2305.18425*.

Rajarshi Saha, Naomi Sagan, Varun Srivastava, Andrea Goldsmith, and Mert Pilanci. 2024. Compressing large language models using low rank and low precision decomposition. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8317–8326.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. A simple and effective pruning approach for

large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. 2025. Openmathinstruct-2: Accelerating AI for math with massive open-source instruction data. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Yan Yang, Yixia Li, Hongru Wang, Xuetao Wei, James Jianqiao Yu, Yun Chen, and Guanhua Chen. 2025. Impart: Importance-aware delta-sparsification for improved model compression and merging in llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18817–18829.

Xiaozhe Yao, Qinghao Hu, and Ana Klimovic. 2025. Deltazip: Efficient serving of multiple full-model-tuned llms. In *Proceedings of the European Conference on Computer Systems (EuroSys)*, pages 110–127.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Cheng Zhang, Jianyi Cheng, George Anthony Constantinides, and Yiren Zhao. 2024. LQER: low-rank quantization error reconstruction for llms. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Method	DS-R1-LLaMA-8B			DS-R1-Qwen-14B				DS-R1-Qwen-32B		
	MATH	AIME24	GPQA	MATH	AIME24	GPQA	HumanEval	MATH	AIME24	GPQA
Backbone	2.60	0	20.13	64.15	6.67	32.32	72.6	62.40	8.33	36.36
SFT	89.55	44.17	45.20	92.25	61.67	56.94	77.4	92.65	67.50	63.51
Random	7.65	0	22.22	58.45	6.67	30.39	69.5	57.30	8.17	34.93
Magnitude	18.40	1.33	31.19	76.35	17.33	37.82	75.2	78.65	20.83	42.74
Wanda	24.70	2.17	30.72	78.20	17.67	39.04	75.8	78.40	20.83	44.52
SVD	39.20	4.67	30.32	53.90	15.50	44.34	69.5	60.30	17.67	50.43
BitDelta	11.80	0.83	30.56	20.95	0	35.35	73.8	22.75	0	39.43
Delta-CoMe	45.40	5.83	30.18	87.70	47.00	51.92	66.5	88.10	53.33	58.46
D-QRELO (ours)	**79.05**	**23.83**	**35.98**	**91.40**	**57.50**	**53.27**	**77.4**	**92.45**	**62.50**	**61.47**

Table 6: The performance of different delta-compression methods on three powerful SFT LLMs. Optimal results are in bold, while sub-optimal results are underlined.

A Experiments Configuration

Model	Backbone
OpenMath2-LLaMA3.1-8B	LLaMA3.1-8B
Qwen2-VL-7B-Instruct	Qwen2-7B
DeepSeek-R1-Distill-LLaMA3-8B	LLaMA3.1-8B
DeepSeek-R1-Distill-Qwen-14B	Qwen2.5-14B
DeepSeek-R1-Distill-Qwen-32B	Qwen2.5-32B
DeepSeek-R1-0528-Qwen3-8B	Qwen3-8B-Base
Qwen3-4B-Instruct-2507	Qwen3-4B-Base
Qwen3-30B-A3B-Instruct-2507	Qwen3-30B-A3B-Base

Table 7: Backbone of each evaluated model.

In our setup, LoRA parameters are matched with those of D-QRELO (1,040,187,392 vs. 1,003,782,656) to ensure fair comparison. Specifically, LoRA adopts a rank of 512 and an α of 512, applied to the Q, K, V, O projections as well as the input/output projections of FFN, without dropout or projection sharing/parallel strategies. For the optimization hyperparameters, LoRA fine-tuning uses a warmup ratio of 0.04 and a peak learning rate of 1e-4. When evaluating OpenMath2 and Qwen2-VL-7B-Instruct, the temperature was set to 0, top_p to 1.0, and maxtokens to 4096, with only one test performed. However, for models in the DeepSeek-R1-distillate series, the temperature was set to 0.6, top_p to 0.95, maxtokens to 20000, and four tests were conducted, and the results were averaged. For Qwen3 models, the temperature was set to 0.7, top_p to 0.8, top_k to 20, and maxtokens to 20000, with one test conducted.

All experiments were conducted on a machine running Ubuntu 20.04, equipped with an NVIDIA A800-SXM4 80GB GPU and an Intel(R) Xeon(R) Platinum 8358P CPU @ 2.60 GHz. The system adopts CUDA 12.4, and the experiments were implemented using PyTorch 2.5.1.

B Evaluation Metrics

All datasets are evaluated using accuracy as the primary metric, except for MME, which uses Score as the evaluation metric, two metrics are introduced for delta compression: Drop and Performance Retention.

The Performance Retention metric, representing the retained performance after compression, is defined as:

$$PR = \frac{\text{Compressed} - \text{Base}}{\text{SFT} - \text{Base}} \tag{17}$$

where PR stands for Performance Retention, while Compressed, Base, and SFT represent the performance of the respective models.

The Drop metric, representing the performance loss rate, is defined as:

$$\text{Drop} = 1 - PR \tag{18}$$

C Additional Results

Table 6 reports the performance of DS-R1-LLaMA-8B, DS-R1-Qwen-14B, and DS-R1-Qwen-32B on MATH, AIME, GPQA, and HumanEval tasks. Consistent with our main findings, D-QRELO remains the top-performing method across all evaluated settings.