

Appendix

This appendix provides the supplementary materials for this work “R2Q: Residual Refinement Quantization for Robust 2-Bit Large Language Models”, constructed according to the corresponding sections therein.

A LLM USAGE STATEMENT

In the preparation of this manuscript, we utilized a large language model (LLM) to assist with language editing and refinement. Specifically, we used Google’s Gemini Pro and OpenAI’s GPT-4 for tasks such as correcting grammar, improving sentence structure for clarity, and ensuring consistency in terminology. Our process involved first drafting the content ourselves to articulate the core scientific ideas, methodology, and results. Subsequently, the LLM was prompted to polish the language of the human-authored text. All suggestions provided by the LLM were critically reviewed, edited, and approved by the authors to ensure the final text accurately reflects our original research and intent. The intellectual contribution, including all concepts, experimental design, analysis, and conclusions presented in this paper, is entirely the work of the human authors.

Table 4: Performance comparison between the original BitDistiller (BitDistiller (RTN)) and BitDistiller integrated with R2Q (BitDistiller (R2Q)) under coarse-grained and fine-grained quantization.

Model	Method	Group-Size	ARC-c \uparrow	ARC-e \uparrow	BoolQ \uparrow	Hella. \uparrow	PIQA \uparrow	Wino. \uparrow	MMLU \uparrow	WikiText-2 \downarrow
Llama-7B	BitDistiller (RTN)	-1	23.38	25.88	53.88	25.59	51.46	50.43	24.61	15938.61
		128	28.67	57.91	63.18	41.76	69.1	60.22	24.26	33.5056
	BitDistiller (R2Q)	-1	20.99	40.11	54.80	30.17	60.66	51.46	23.05	310.03
		256	20.56	28.03	47.89	26.13	53.54	50.12	24.94	8104.6957
OPT-6.7B	BitDistiller (RTN)	-1	21.50	42.55	48.69	31.88	61.10	52.41	24.57	125.01
		128	26.37	53.7	62.35	41.22	68.66	56.99	25.40	22.49
	BitDistiller (R2Q)	-1	25.09	55.01	63.36	41.30	71.11	57.30	24.68	112.55
		256	24.32	54.21	63.52	41.36	70.84	59.19	25.96	95.56
Qwen2.5-7B	BitDistiller (RTN)	-1	20.39	23.76	42.20	25.02	52.45	49.64	26.85	42877.17
		128	-	-	-	-	-	-	-	-
	BitDistiller (R2Q)	-1	36.09	61.61	70.03	39.59	68.99	60.69	43.88	193.05
		256	36.09	63.80	68.78	40.64	70.67	62.43	48.85	122.18
Qwen3-4B	BitDistiller (RTN)	-1	20.39	26.30	38.10	25.75	52.34	48.86	24.48	44696.35
		128	30.12	54.59	71.04	36.01	63.82	55.41	33.15	115.24
	BitDistiller (R2Q)	-1	23.63	43.77	66.17	31.72	59.41	53.20	29.52	494.6
		256	32.68	63.68	72.51	38.35	64.52	55.8	44.07	61.93

B R2Q AS A PLUG-AND-PLAY MODULE

To demonstrate the efficacy and plug-and-play nature of R2Q, we integrated it into the BitDistiller post-training quantization (PTQ) framework. The primary results are summarized in Table 2 (Section 5), with detailed breakdowns provided in Table 4.

Llama-7B. Under coarse-grained quantization (Group-Size = -1), BitDistiller (RTN) slightly outperforms R2Q on ARC-c (23.38 vs. 20.99). However, R2Q achieves substantial gains on ARC-e (40.11 vs. 25.88), HellaSwag (30.17 vs. 25.59), PIQA (60.66 vs. 51.46), and drastically lowers WikiText-2 perplexity (310.03 vs. 15,938.61), highlighting its ability to recover model performance under extreme quantization. Under finer-grained quantization (Group-Size 128/256), trends favor larger group sizes for RTN, but R2Q consistently maintains competitive results across tasks.

OPT-6.7B. Under coarse-grained quantization, R2Q demonstrates superior performance over RTN across nearly all evaluated metrics: ARC-c (25.09 vs. 21.50), ARC-e (55.01 vs. 42.55), BoolQ (63.36 vs. 48.69), HellaSwag (41.30 vs. 31.88), PIQA (71.11 vs. 61.10), and Winogrande (57.30 vs. 52.41), while also reducing WikiText-2 perplexity from 125.01 to 112.55. In the fine-grained setting (Group-Size 256), R2Q continues to either match or slightly surpass RTN, highlighting its robust generalization capabilities.

Table 5: Ablation study on the contribution of the residual refinement stage of R2Q. The initial 1-bit coarse approximation against the complete 2-bit R2Q method is compared over language understanding and modeling tasks. For ARC-c/e, BoolQ, Hellaswag, PIQA, and Winogrande, we report accuracy. For WikiText-2, we report PPL. To align the scale parameters of the 1-bit coarse-estimated control group with the 2-bit full R2Q method, we set the group size of 128 for the former and 256 for the latter in fine-grained quantification.

Model	Bit-Width	Group-Size	ARC-c \uparrow	ARC-e \uparrow	BoolQ \uparrow	Hella \uparrow	PIQA \uparrow	Wino. \uparrow	MMLU \uparrow	Wiki2 \downarrow
Llama-7B	bf16	\	41.98	75.38	75.17	56.95	78.78	69.46	31.33	9.39
	1-bit	-1	19.45	36.07	61.01	29.00	57.02	52.09	22.91	66.28
	2-bit	-1	27.47	56.82	59.36	44.44	70.08	57.54	24.08	17.13
	1-bit	128	18.43	37.33	60.31	29.86	59.30	51.70	22.92	49.81
	2-bit	256	28.24	59.18	64.89	45.37	70.62	59.59	24.28	16.96
	2-bit	256	28.24	59.18	64.89	45.37	70.62	59.59	24.28	16.96
OPT-6.7B	bf16	\	30.38	65.53	65.72	50.53	76.22	64.88	24.94	12.28
	1-bit	-1	20.73	40.45	53.06	30.11	60.28	50.51	22.99	417.53
	2-bit	-1	25.68	58.08	63.70	44.19	71.43	60.77	24.60	16.71
	1-bit	128	21.16	41.83	54.22	30.25	61.70	50.75	22.95	480.08
	2-bit	256	24.32	58.96	64.40	44.31	72.58	60.54	25.09	16.81
	2-bit	256	24.32	58.96	64.40	44.31	72.58	60.54	25.09	16.81

Qwen2.5-7B. The performance enhancements afforded by R2Q are particularly pronounced for Qwen2.5-7B under coarse-grained quantization. For instance, the score on ARC-e improves from 23.76 to 61.61, BoolQ from 42.20 to 70.03, and WikiText-2 perplexity is drastically reduced from 42,877.17 to 193.05. Notably, the baseline RTN method experienced gradient explosion during training, resulting in no reported metrics (denoted by '-'). This observation underscores the enhanced training stability provided by R2Q. With fine-grained quantization (Group-Size 256), R2Q maintains its strong performance across all benchmarks.

Qwen3-4B. Similarly, integrating R2Q significantly enhances the performance of Qwen3-4B under coarse quantization settings. Performance on BoolQ increases from 38.10 to 66.17, ARC-e from 26.30 to 43.77, and WikiText-2 perplexity decreases from 44,696.35 to 494.6. Under a fine-grained configuration (Group-Size 256), R2Q continues to match or exceed the performance of RTN, achieving scores of 63.68 on ARC-e and 72.51 on BoolQ.

The experiments validate that R2Q is a highly effective, stable, and versatile plug-and-play solution that significantly enhances the performance and reliability of QAT for modern large language models.

C ABLATION STUDY OF THE RESIDUAL REFINEMENT

As described in Section 4, our R2Q method adopts a two-step quantization strategy designed to balance efficiency with accuracy. The first step generates a 1-bit kernel, $\alpha_1 \mathbf{Q}_1$, which serves as a **coarse approximation** of the full-precision weights \mathbf{W} . In the second step, R2Q quantifies the residual error $\mathbf{R} = \mathbf{W} - \alpha_1 \mathbf{Q}_1$ and encodes it using another 1-bit kernel—constituting the **residual refinement** stage. This decomposition is central to R2Q’s design, enabling it to go beyond naïve low-bit quantization methods by explicitly modeling and correcting quantization errors.

Unlike standard 2-bit quantization methods, such as Round-To-Nearest (RTN), which directly map full-precision weights to a small discrete set, R2Q introduces a hierarchical decomposition: a coarse 1-bit quantization followed by an explicit residual correction. This framework enhances traditional 1-bit quantization methods (Wang et al., 2023; Xu et al., 2024) by addressing their key limitation—uncompensated approximation errors—through structured refinement.

While Section 5 confirms the effectiveness of R2Q overall, we further conducted an ablation study to assess the independent contribution of the setting of residual refinement. Specifically, we compared the performance of (1) the initial 1-bit coarse approximation and (2) the full 2-bit R2Q method. The dequantization for the 1-bit coarse quantization is,

$$\hat{\mathbf{w}}^{(i)} = \alpha_1^{(i)} \mathbf{q}_1^{(i)} \quad (25)$$

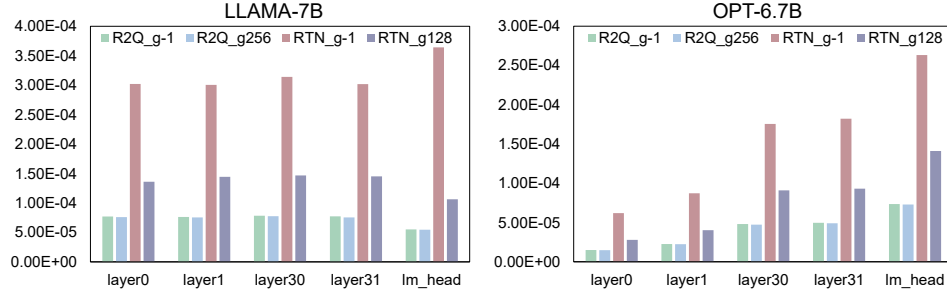


Figure 4: Weight changes before and after QAT using RTN and R2Q, measured by MSE. For clarity, we report results for the first two decoder layers (layer0 and layer1), the last two decoder layers (layer30 and layer31) and the LM head of the OPT-6.7B and Llama-7B models. R2Q consistently preserves strong alignment with the original weights in both coarse-grained and fine-grained settings, whereas RTN exhibits poor alignment and a significant degradation under coarse-grained quantization.

All evaluations were performed under the same experimental configurations described in Section ???. The results are presented in Table 5. To ensure a fair comparison in terms of parameter count, we align the number of scaling factors by setting the group size of the 1-bit ablation baseline to 128 and that of the complete 2-bit R2Q model to 256 in the fine-grained quantization setting.

Across Llama-7B and OPT-6.7B, moving from 1-bit to 2-bit quantization yields substantial gains. On ARC-e, R2Q improves accuracy from 36.07→56.82 (Llama-7B) and 40.45→58.08 (OPT-6.7B), with relative gains of 57.5% and 43.6%. The largest benefit appears in language modeling: 1-bit models show extremely high perplexity—66.28 and 417.53—signaling severe degradation, while 2-bit R2Q reduces perplexity to 17.13 and 16.71, close to full-precision (9.39 and 12.28). This >90% reduction underscores the critical role of the refinement bit in preserving generative capacity. Improvements persist under grouped settings: with group size 128 vs. 256, ARC-e rises from 37.33→59.18 (Llama-7B) and 41.83→58.96 (OPT-6.7B), confirming robustness under deployment constraints. Moreover, the refinement boosts not just individual benchmarks but sustains balanced performance across commonsense reasoning, reading comprehension, and causal reasoning, suggesting the second bit captures generalized error patterns. These results strongly validate our central hypothesis: while a 1-bit kernel offers a compact but rough estimate of the weight distribution, it fails to retain the finer-grained information necessary for complex tasks. The second bit in R2Q serves as an adaptive corrective mechanism, effectively modeling residual errors and yielding a much closer approximation to the full-precision model. The ablation confirms that residual refinement is not just a marginal enhancement—it is a critical design component that enables R2Q to achieve high accuracy under extreme quantization.

D QUANTIZATION ERROR ANALYSIS

To investigate the source of R2Q’s effectiveness, we conduct a comprehensive analysis of quantization error by comparing model weights before and after applying QAT. Specifically, we compute the Mean Squared Error (MSE) between corresponding layers as,

$$E = \frac{1}{L} \sum_{j=1}^L \frac{\|\mathbf{W}_j - \hat{\mathbf{W}}_j\|_2^2}{N_j}, \quad (26)$$

where L denotes the number of linear layers in a decoder block, and N_j is the number of parameters in the j -th layer.

We report results on Llama-7B and OPT-6.7B, focusing on the first and last two decoder layers, as well as the language model (LM) head, for clarity. The quantization error trends observed in other layers are consistent with the findings presented here.

As illustrated in Figure 4, RTN exhibits a significant discrepancy in quantization error between coarse-grained (RTN_g-1) and fine-grained (RTN_g128) configurations. Across all examined layers

in both models, coarse-grained RTN consistently yields higher error—approximately double that of its fine-grained counterpart. This trend is consistent with the results discussed in Section 5. The elevated error under coarse granularity reflects the inefficiency of RTN in utilizing the quantization lattice: a small set of values dominates a few quantization points, leading to poor utilization of the available range and degraded representational fidelity.

In contrast, R2Q demonstrates strong robustness to quantization granularity. The weight error difference between coarse-grained (R2Q_g-1) and fine-grained (R2Q_g256) settings remains minimal across all layers. This observation aligns with the performance stability of R2Q across different configurations, as shown in Section 5. Notably, even under the most aggressive per-channel quantization (group size of -1), R2Q effectively preserves the structural integrity of the original weights.

Furthermore, R2Q consistently achieves lower MSE than RTN across both coarse- and fine-grained schemes. This closer alignment with the original weights provides compelling evidence of R2Q’s quantization efficacy and offers a clear explanation for its superiority over existing 2-bit quantization approaches that are fundamentally based on RTN.

Table 6: A comparison of the computational complexity for a matrix multiplication operation ($M \times N$ by $N \times K$) across different quantization schemes. R2Q significantly reduces the number of floating-point multiplications required.

	FP16	INT2	R2Q
Mul	$MNK \text{ FP16} \times \text{FP16}$	$MNK \text{ INT2} \times \text{FP16} + MN \text{ FP16}$	$2MN \text{ FP16} \times \text{FP16}$
Add	$MN(K-1) \text{ FP16}$	$MN(K-1) \text{ FP16}$	$2MN(K-1) \text{ FP16} + MN \text{ FP16}$

E MATRIX MULTIPLICATION FOR R2Q

As detailed in Section 4, R2Q represents a quantized weight matrix, $\hat{\mathbf{W}} \in \mathbb{R}^{M \times K}$, using two 1-bit kernels, \mathbf{Q}_1 and \mathbf{Q}_2 , along with their corresponding scaling factors, α_1 and α_2 . The reconstruction of the weight matrix is given by

$$\hat{\mathbf{W}} = \alpha_1 \mathbf{Q}_1 + \alpha_2 \mathbf{Q}_2 \quad (27)$$

This decomposition allows for a significant optimization of the matrix multiplication (Matmul) operation. For a given input matrix $\mathbf{X} \in \mathbb{R}^{K \times N}$, the product $\hat{\mathbf{W}}\mathbf{X}$ can be calculated by leveraging the distributive property

$$\hat{\mathbf{W}}\mathbf{X} = (\alpha_1 \mathbf{Q}_1 + \alpha_2 \mathbf{Q}_2)\mathbf{X} = \alpha_1(\mathbf{Q}_1\mathbf{X}) + \alpha_2(\mathbf{Q}_2\mathbf{X}) \quad (28)$$

The reformulation in Equation 28 breaks the computation into two highly efficient steps.

1. **Binary Matrix Multiplication:** First, the products of the 1-bit matrices and the input matrix, $\mathbf{Q}_1\mathbf{X}$ and $\mathbf{Q}_2\mathbf{X}$, are computed. Since \mathbf{Q}_1 and \mathbf{Q}_2 contain only values of +1 and -1, this operation **eliminates the need for multiplications**. Instead, it is executed using only additions.
2. **Scaling and Combination:** The resulting matrices from the first step are then scaled element-wise (Hadamard product) by their respective scaling factors, α_1 and α_2 . Finally, the two scaled matrices are added together to produce the final result.

Additionally, since the calculations of $\alpha_1 \mathbf{Q}_1 \mathbf{X}$ and $\alpha_2 \mathbf{Q}_2 \mathbf{X}$ are completely unrelated, both can be performed **in parallel**. This approach dramatically reduces the number of expensive multiplication operations. Table 6 provides a comparison of the computational complexity for matrix multiplication using full-precision (FP16), standard 2-bit integer (INT2), and our proposed R2Q method. As shown, R2Q substantially decreases the reliance on high-precision multiplications compared to the other methods.