

A THE USE OF LARGE LANGUAGE MODELS (LLMs)

In this paper, LLMs are only used to assist in polishing the writing of this paper. The technical content, experiments, and conclusions are entirely conceived and conducted by the authors.

B ADDITIONAL DETAILS OF EVALUATION

B.1 INTRODUCTION OF DATASETS

American Invitational Mathematics Examination (AIME) (AIME, 2025) is a mathematics competition for high school students. It contains 30 challenging problems each year, designed to assess mathematical problem-solving skills across various topics, including algebra, combinatorics, geometry, number theory, and other subjects covered in high school curricula.

Carnegie Mellon Informatics and Mathematics Competition (CMIMC) (CMIMC, 2025) is an annual mathematics contest for high school students, hosted by students from Carnegie Mellon University. The competition contains problems of algebra, combinatorics, and geometry, with each category including ten standard problems along with one tiebreaker. Our model evaluation focuses on the standard problem sets.

LiveCodeBench (Jain et al., 2024) is an extensive and continuously updated benchmark designed to evaluate the performance of LLMs in coding tasks. It continually gathers new problems from competition platforms. The benchmark encompasses four distinct scenarios: code generation, automated code repair, code execution, and prediction of test outputs. In our experiments, we focus specifically on the code generation scenario.

IFEval (Zhou et al., 2023) is a benchmark proposed to systematically evaluate the ability of LLMs to follow natural language instructions. The dataset contains 541 prompts, each annotated with one or more verifiable instruction types such as word-count constraints, keyword frequency, formatting requirements, or prohibitions on certain symbols. These instructions were deliberately designed to be automatically checkable, enabling objective and reproducible evaluation without the need for human annotators.

C ADDITIONAL EXPERIMENTS

C.1 THE SENSITIVITY OF DIFFERENT TRANSFORMER BLOCKS

We analyze the sensitivity and the memory allocation results across different models. For models with parameter size less than 10B, as shown in Figure 3, we observe that the deeper blocks tend to be more sensitive to quantization and receive a larger memory budget for the KV Cache. In addition, in the DeepSeek-R1-Distill-Qwen-7B model, the first block is much more sensitive than the other shallow blocks. Our memory allocation strategy accurately captures this feature, assigning a higher memory budget to the first block accordingly.

For larger models with parameter size over 10B, as shown in Figure 4, KV Cache in deeper blocks tend to be more sensitive than shallower blocks. We also observe that for the Qwen-based models, the first block exhibits a large sensitivity. In particular, the sensitivity of the first block is the largest among the first fifteen blocks in different Qwen-based models. This phenomenon is not observed in the LLaMA-based models.

C.2 PERFORMANCE IN SHORT-GENERATION-CONTEXT TASKS

To verify the scalability of PM-KVQ to short-generation-context tasks, we evaluate it on IFEval (Zhou et al., 2023), an instruction-following benchmark. We follow the experimental setup described in Section 4.1 and adopt the evaluation metrics provided by OpenCompass (Contributors, 2023). Compared to reasoning benchmarks in Table 1, non-reasoning tasks are less challenging and generally involve much shorter outputs. For instance, the average output length of the DeepSeek-Qwen-7B model is 13,904 tokens on AIME-2024 but only 1,182 tokens on IFEval. As shown in

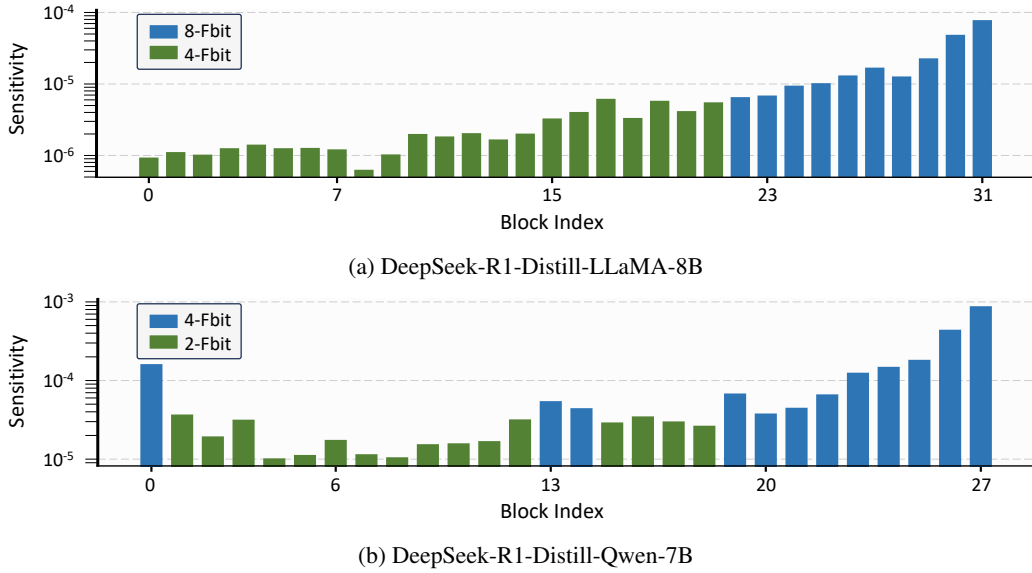


Figure 3: Sensitivity to quantization of KV Cache in different transformer blocks. Different colors represents different memory budgets.

Table 5: Results of long-CoT Language Models on non-reasoning benchmarks with SOTA KV Cache quantization methods. “BS” is short for “batch size”.

Models (Target GPU)	Quantization Methods	Bit-width (K-V)	IFEval			
			Prompt Strict	Prompt Loose	Instruct Strict	Instruct Loose
DeepSeek- Qwen-7B (1×4090-24G)	--	16-16	58.77	68.50	63.27	72.60
	RotateKV (BS=32,40)	2-2	0.00	0.00	0.00	0.00
	MiKV (BS=32)	2/16-2/16	57.30	66.17	61.18	70.06
	MiKV (BS=40)	2-2	0.00	0.00	0.00	0.00
	KIVI (BS=32,40)	2-2	49.29	60.31	54.57	64.57
	PM-KVQ (BS=32)	2/4-2/4	57.35	68.03	62.09	71.65
DeepSeek- LLaMA-8B (1×4090-24G)	PM-KVQ (BS=40)	2-2	57.58	68.34	62.09	71.81
	--	16-16	57.82	68.98	61.61	71.81
	RotateKV (BS=6,8)	4-4	58.06	68.50	61.37	71.50
	MiKV (BS=6)	4/16-4/16	44.08	56.38	46.92	59.53
	MiKV (BS=8)	4-4	55.69	66.61	59.95	70.39
	KIVI (BS=6,8)	4-4	57.35	68.35	71.14	71.81
DeepSeek- Qwen-14B (1×A100-40G)	PM-KVQ (BS=6)	4/8-4/8	58.77	69.61	63.74	73.70
	PM-KVQ (BS=8)	4-4	57.58	68.19	61.14	71.34
	--	16-16	70.14	78.74	74.40	81.73
	KIVI (BS=12,16)	2-2	67.54	77.01	72.04	80.47
	PM-KVQ (BS=12)	2/4-2/4	73.70	80.47	77.73	83.46
	PM-KVQ (BS=16)	2-2	73.46	80.47	77.49	83.46
DeepSeek- Qwen-32B (1×A100-80G)	--	16-16	74.41	81.73	78.20	84.41
	KIVI (BS=12,16)	2-2	72.51	79.84	76.07	82.36
	PM-KVQ (BS=12)	2/4-2/4	75.83	83.15	78.91	85.35
	PM-KVQ (BS=16)	2-2	76.07	83.30	78.91	85.35
QwQ-32B (1×A100-80G)	--	16-16	82.94	88.03	86.97	90.71
	KIVI (BS=12,16)	2-2	73.22	80.00	78.67	84.09
	PM-KVQ (BS=12)	2/4-2/4	81.99	86.77	85.55	89.45
	PM-KVQ (BS=16)	2-2	81.75	86.61	85.55	89.45

Table 5, although our method is not specifically designed for short-output scenarios, it outperforms KIVI and achieves accuracy comparable to the original 16-bit models.

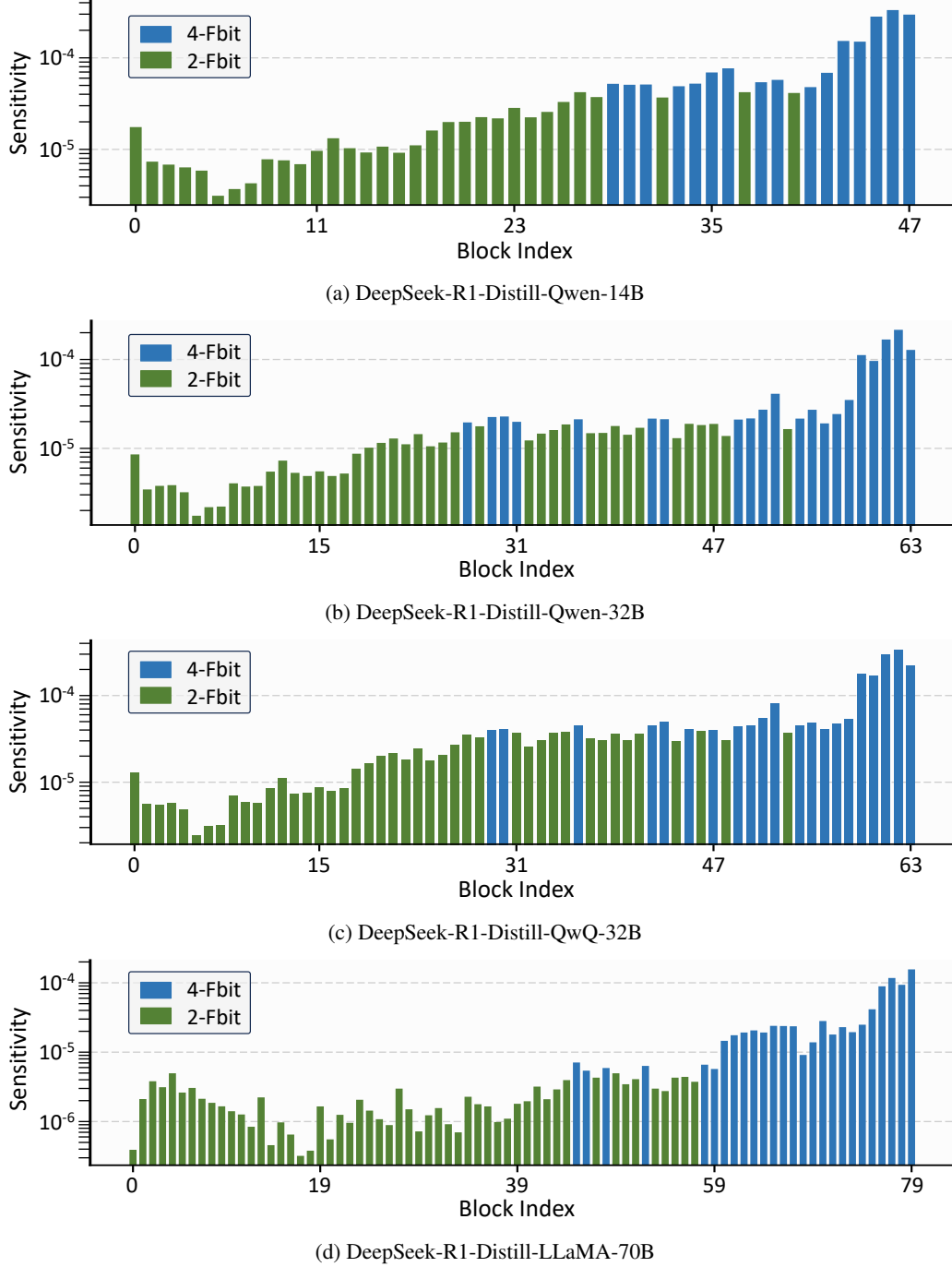


Figure 4: Sensitivity to quantization of KV Cache in different transformer blocks. Different colors represents different memory budgets.

C.3 EFFICIENCY ANALYSIS OF PRE-INFERENCE PROCESS

Before the inference process, PM-KVQ performs block-wise memory allocation and calibration with positional interpolation as preparation. Following the experimental setup in Section 4.1, we measure the time required for these pre-inference procedures. As shown in Table 6, compared with QServe (Lin* et al., 2024), PM-KVQ leverages positional interpolation to reduce calibration sequence length from 8,192 to 2,048 tokens, substantially reducing the calibration time by up to 77.21%. The additional block-wise memory allocation procedure account for 22.50–23.53% pre-inference time.

Table 6: Latency of block-wise memory allocation and calibration. “PI” is short for “Positional Interpolation”.

Model	Method	Calibration		Memory Allocation	Time
		w/o PI	w/ PI		
DeepSeek-Qwen-7B	QServe (search α)	✓			52 min
	PM-KVQ (BS=40)		✓		13 min
	PM-KVQ (BS=32)		✓	✓	17 min
DeepSeek-Qwen-32B	QServe (search α)	✓			187 min
	PM-KVQ (BS=16)		✓		44 min
	PM-KVQ (BS=12)		✓	✓	57 min
DeepSeek-LLaMA-70B	QServe (search α)	✓			408 min
	PM-KVQ (BS=16)		✓		93 min
	PM-KVQ (BS=12)		✓	✓	120 min

D PROOF OF EQUIVALENT RIGHT SHIFT

Theorem D.1 (Equivalent Right Shift). *Given a 16-bit floating-point tensor $\mathbf{X}_{\text{FP16}}$, let $\mathbf{X}_{2b}$ and $\mathbf{X}_b$ denote the 2b-bit and b-bit quantized tensors of $\mathbf{X}_{\text{FP16}}$, respectively. Then*

$$\mathbf{X}_b = ((2^{2b} - 2^b + 1)(\mathbf{X}_{2b} + 2^{b-1})) \gg 3b. \quad (13)$$

Proof. Let the zero points be $Z_{2b} = Z_b = Z$. According to Equation (2), the scaling factors are given by

$$S_{2b} = \frac{\max(\mathbf{X}_{\text{FP16}}) - Z}{2^{2b} - 1}, \quad S_b = \frac{\max(\mathbf{X}_{\text{FP16}}) - Z}{2^b - 1}. \quad (14)$$

It follows that

$$S_b = (2^b + 1)S_{2b}. \quad (15)$$

Define

$$\tilde{\mathbf{X}}_{2b} = \frac{\mathbf{X}_{\text{FP16}} - Z}{S_{2b}}, \quad \tilde{\mathbf{X}}_b = \frac{\mathbf{X}_{\text{FP16}} - Z}{S_b}. \quad (16)$$

Then the quantized tensors are obtained by rounding:

$$\mathbf{X}_{2b} = \left\lfloor \tilde{\mathbf{X}}_{2b} \right\rfloor, \quad \mathbf{X}_b = \left\lfloor \tilde{\mathbf{X}}_b \right\rfloor, \quad (17)$$

and we have

$$\tilde{\mathbf{X}}_{2b} = (2^b + 1)\tilde{\mathbf{X}}_b. \quad (18)$$

By the definition of rounding,

$$\mathbf{X}_{2b} - \frac{1}{2} \leq \tilde{\mathbf{X}}_{2b} < \mathbf{X}_{2b} + \frac{1}{2}. \quad (19)$$

Dividing both sides by $2^b + 1$ yields

$$\frac{\mathbf{X}_{2b} - \frac{1}{2}}{2^b + 1} \leq \tilde{\mathbf{X}}_b < \frac{\mathbf{X}_{2b} + \frac{1}{2}}{2^b + 1}. \quad (20)$$

Perform the Euclidean division of $\mathbf{X}_{2b}$ by $2^b + 1$:

$$\mathbf{X}_{2b} = q(2^b + 1) + r, \quad \text{with } 0 \leq q \leq 2^b - 1, 0 \leq r \leq 2^b. \quad (21)$$

Then,

$$q + \frac{r - \frac{1}{2}}{2^b + 1} \leq \tilde{\mathbf{X}}_b < q + \frac{r + \frac{1}{2}}{2^b + 1}. \quad (22)$$

Now consider the expression:

$$\begin{aligned} ((2^{2b} - 2^b + 1)(\mathbf{X}_{2b} + 2^{b-1})) >> 3b &= \left\lfloor \frac{(2^{2b} - 2^b + 1)(q(2^b + 1) + r + 2^{b-1})}{2^{3b}} \right\rfloor \\ &= q + \left\lfloor \frac{q + (2^{2b} - 2^b + 1)(r + 2^{b-1})}{2^{3b}} \right\rfloor. \end{aligned} \quad (23)$$

We proceed by considering two cases for the remainder r :

Case 1: $0 \leq r \leq 2^{b-1}$.

Then,

$$q - \frac{1}{2} < q - \frac{\frac{1}{2}}{2^b + 1} \leq \tilde{\mathbf{X}}_b < q + \frac{2^{b-1} + \frac{1}{2}}{2^b + 1} = q + \frac{1}{2}. \quad (24)$$

Hence, rounding gives $\mathbf{X}_b = \left\lfloor \tilde{\mathbf{X}}_b \right\rfloor = q$.

Moreover,

$$\frac{q + (2^{2b} - 2^b + 1)(r + 2^{b-1})}{2^{3b}} \geq \frac{(2^{2b} - 2^b + 1) \cdot 2^{b-1}}{2^{3b}} > \frac{2^{2b} \cdot 2^{b-1}}{2^{3b}} = \frac{1}{2} > 0, \quad (25)$$

and

$$\begin{aligned} \frac{q + (2^{2b} - 2^b + 1)(r + 2^{b-1})}{2^{3b}} &\leq \frac{2^b - 1 + (2^{2b} - 2^b + 1)(2^{b-1} + 2^{b-1})}{2^{3b}} \\ &= 1 - \frac{(2^b - 1)^2}{2^{3b}} < 1. \end{aligned} \quad (26)$$

Therefore,

$$\left\lfloor \frac{q + (2^{2b} - 2^b + 1)(r + 2^{b-1})}{2^{3b}} \right\rfloor = 0, \quad (27)$$

and thus

$$((2^{2b} - 2^b + 1)(\mathbf{X}_{2b} + 2^{b-1})) >> 3b = q = \mathbf{X}_b. \quad (28)$$

Case 2: $2^{b-1} + 1 \leq r \leq 2^b$.

Then,

$$q + \frac{1}{2} = q + \frac{2^{b-1} + 1 - \frac{1}{2}}{2^b + 1} \leq \tilde{\mathbf{X}}_b < q + \frac{2^b + \frac{1}{2}}{2^b + 1} < q + 1. \quad (29)$$

Thus, rounding gives $\mathbf{X}_b = \left\lfloor \tilde{\mathbf{X}}_b \right\rfloor = q + 1$.

Moreover,

$$\frac{q + (2^{2b} - 2^b + 1)(r + 2^{b-1})}{2^{3b}} \geq \frac{(2^{2b} - 2^b + 1)(2^{b-1} + 1 + 2^{b-1})}{2^{3b}} = \frac{2^{3b} + 1}{2^{3b}} > 1, \quad (30)$$

and

$$\begin{aligned} \frac{q + (2^{2b} - 2^b + 1)(r + 2^{b-1})}{2^{3b}} &\leq \frac{2^b - 1 + (2^{2b} - 2^b + 1)(2^b + 2^{b-1})}{2^{3b}} \\ &= \frac{2^{3b} + 2^{3b-1} - 2^{2b} - 2^{2b-1} + 2^{b+1} + 2^{b-1} - 1}{2^{3b}} \\ &= 2 - \frac{2^{3b-1} + 2^{2b} + 2^{2b-1} - 2^{b+1} - 2^{b-1} + 1}{2^{3b}} < 2. \end{aligned} \quad (31)$$

Therefore,

$$\left\lfloor \frac{q + (2^{2b} - 2^b + 1)(r + 2^{b-1})}{2^{3b}} \right\rfloor = 1, \quad (32)$$

and thus

$$((2^{2b} - 2^b + 1)(\mathbf{X}_{2b} + 2^{b-1})) \gg 3b = q + 1 = \mathbf{X}_b. \quad (33)$$

In both cases, the desired equality holds, which completes the proof. $\square$

E LIMITATIONS

In this paper, we do not consider all of the attention mechanisms, such as the multi-head latent attention (MLA), which is quite different from the widely used Group-Query Attention (GQA). Besides, we do not combine the proposed PM-KVQ with other system-level optimization techniques and inference engines, which yields for future work.