When Reasoning Meets Compression: Understanding the Effects of LLMs Compression on Large Reasoning Models

Anonymous authors
Paper under double-blind review

Abstract

Compression methods, including quantization, distillation, and pruning, improve the computational efficiency of large reasoning models (LRMs). However, existing studies either fail to sufficiently compare all three compression methods on LRMs or lack in-depth interpretation analysis. In this paper, we investigate how the reasoning capabilities of LRMs are compromised during compression, through performance benchmarking and mechanistic interpretation. To uncover the effects of compression on reasoning performance, we benchmark quantized, distilled, and pruned DeepSeek-R1 models on four reasoning datasets (AIME 2024, FOLIO, Temporal Sequences, and MuSiQue). To precisely locate compression effects on model weights, we adapt difference of means and attribution patching techniques, focusing on the activation of every linear component in compressed LRMs, to interpret fine-grained causal relationships between weights and various reasoning capabilities. This fine-grained interpretation addresses a fundamental question of compression: which weights are the most important for reasoning? Overall, we find dynamically quantized 2.51-bit R1 reaches close-to-R1 performance. With empirical verification, we present three main findings that generalize across both Llama and Qwen: (1) Weight count has a greater impact on LRMs' knowledge memorization than reasoning, highlighting the risks of pruning and distillation; (2) The MLP up projection in the final layer of distilled LRMs is one of the most important components, offering a new perspective on locating critical weights — a fundamental problem in model compression; and (3) Current quantization methods overly compress the final-layer modules and MLP gate projections, so protecting just 2% of all weights that are excessively compressed can raise average accuracy by 6.57%, greatly surpassing the state-of-the-art.

1 Introduction

Large reasoning models (LRMs) such as DeepSeek-R1 (Guo et al., 2025) excel at complex reasoning tasks. However, due to their large sizes, deploying them can be costly and even infeasible for individuals, which hinders AI democratization. Compression methods including quantization, distillation, and pruning reduce computational resources (*e.g.*, GPU memory and disk space). Representative quantization techniques include dynamic quantization by Unsloth (Daniel Han & team, 2023), activation-aware quantization AWQ (Lin et al., 2024), and post-training quantization GPTQ (Frantar et al., 2022). Current distillation involves black-box (Li et al., 2024a) or white-box (Gu et al., 2024) settings. Representative pruning techniques include unstructured (Zhang et al., 2024; Frantar & Alistarh, 2023) and structured pruning (Xia et al., 2024; Ma et al., 2023).

However, existing works do not sufficiently study the performance of compression method on LRMs (Liu et al., 2025; Srivastava et al., 2025; Feng et al., 2025). Although current quantization and pruning methods claim to preserve the performance of general-purpose LLMs, benchmarking both of them on LRMs with more reasoning-intensive datasets helps compare their collapse point. Regarding distillation, recent works either fail to comprehensively evaluate their student models on diverse reasoning benchmarks of varying difficulty or neglect to consider distillation effect on knowledge and reasoning (Huang et al., 2024; Agarwal et al., 2024). Another research gap is the lack of interpretability of compression effects on LRMs. It is necessary to interpret how compres-

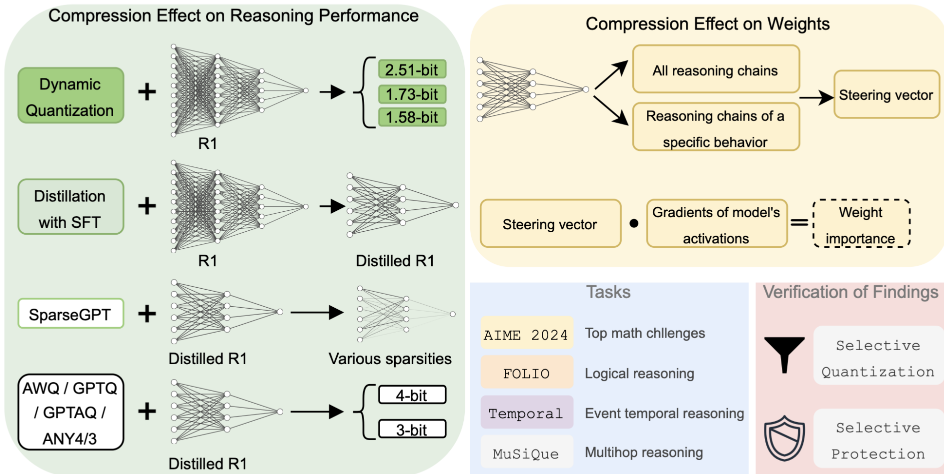

Figure 1: An overview of our pipeline. Left: We benchmark compressed R1 variants on various reasoning tasks. Right: By computing weight importance towards a specific reasoning behavior (a dot product of the steering vector and gradients with respect to an LRM's activations), we study the compression effects on individual weight. We empirically verify our findings on weight importance by selectively quantizing or protecting a module to test its importance.

sion methods affect LRMs, as such analysis can reveal existing bottlenecks and provide guidance for future compression research.

Therefore, due to the lack of compression works on LRMs, we study this fundamental research question: **How are the reasoning capabilities of LRMs compressed during compression?** We answer it from two perspectives — performance benchmarking and mechanistic interpretation. We first benchmark compressed DeepSeek-R1 on various reasoning tasks to investigate how model compression affects performance. We test dynamic quantization (Daniel Han & team, 2023), distillation with supervised fine-tuning (SFT) (Guo et al., 2025), SparseGPT (Frantar & Alistarh, 2023), AWQ (Lin et al., 2024), GPTQ (Frantar et al., 2022), GPTAQ (Li et al., 2025), and ANY4/3 (Elhoushi & Johnson, 2025) on R1 (or distilled R1). Then, we apply mechanistic interpretability to quantify weight contribution towards four core reasoning capabilities of LRMs: backtracking, uncertainty estimation, example testing, and adding knowledge. By focusing on the activation of every linear component in compressed LRMs, we adapt difference of means (Arditi et al., 2024) to extract steering vectors and attribution patching (Syed et al., 2023) to compute weight importance. Unlike previous analysis (Venhoff et al., 2025) that only measures layer-wise weight contribution, our weight importance scores offer more fine-grained interpretation of weight contribution, addressing the fundamental compression question of locating important weights. By comparing weight importance scores between compressed LRMs and original LRMs, we quantify the effects of distillation and quantization on model weights[1]. Our analysis framework is shown in Figure 1.

With empirical verification, our key findings are summarized below for better understanding and improving LRMs compression:

- **Weight count** has a greater impact on LRMs' knowledge memorization than their reasoning capabilities, highlighting the compression effects of pruning and distillation. Thus, both distillation and pruning are discouraged when tasks require LRMs' parametric knowledge.
- **The `mlp.up_proj` in the final layer** of R1 distilled models emerges as one of the most important model components, addressing a core concern in pruning and quantization literature: identifying critical weights. Quantizing only this matrix to 3-bit reduces the average accuracy by 16.3%.
- **Final-layer modules, along with the `mlp.gate_proj`** of R1 distilled Llama and Qwen, are overly compressed by popular quantization methods, highlighting the need for greater attention to preserving their weight precision. A successful protection of only final-layer MLP modules could raise average accuracy by 6.57%, with gains of up to 23.17% over the state-of-the-art quantization.

[1]Our interpretation code will be released upon paper acceptance.

2 PROBLEM FORMULATION

2.1 BACKGROUND

As discussed in Section 1, compression on LRMs (not LLMs) is relatively underexplored. We conduct a thorough literature review in Appendix B.

Bottlenecks on evaluation. Few quantization or pruning methodologies have sufficiently demonstrated effectiveness on LRMs. Current works evaluate quantization and pruning performance primarily using perplexity and simple end tasks, such as the EleutherAI evaluation harness (Gao et al., 2024) and commonsense reasoning. However, quantized or pruned LRMs should be assessed on more complex reasoning tasks with varying difficulty levels. For distillation, although recent works tend to test on more challenging reasoning tasks (compared to other compression literature) such as GSM8K (Cobbe et al., 2021), it is unclear how the compression of LRMs affects models' parametric knowledge and reasoning capability. Some of them do not comprehensively select diverse reasoning benchmarks (Agarwal et al., 2024). Our benchmarking aims to address these bottlenecks.

Bottlenecks on in-depth analysis. The lack of interpretability of compression effects on LRMs is a key bottleneck of in-depth analysis for compressed LRMs. Being able to interpret the difference between original and compressed LRMs offers a new way to analyze the effects of compression. As a result, better compression approaches can be developed. A recent work (Venhoff et al., 2025) interprets several R1 distilled LRMs, but their focus is not on understanding compression effects.

Recent efforts. Recent benchmarking (Liu et al., 2025) and survey (Feng et al., 2025; Srivastava et al., 2025) papers have begun to evaluate compressed LRMs on more complex reasoning datasets, but they all lack in-depth interpretation of compression effects and do not comprehensively compare different compression strategies. As for compressed LRMs, Unsloth (Daniel Han & team, 2023) introduces dynamic quantization by dynamically opting not to quantize certain LLM weights. DeepSeek-R1 (Guo et al., 2025) also comes with several distilled models via black-box distillation. Our interpretation analysis aims to demystify the effects of LLMs compression on LRMs, providing a systematic understanding of existing compressed LRMs.

2.2 MECHANISTIC INTERPRETATION

For our interpretation analysis, we target four core reasoning behaviors following an existing work (Venhoff et al., 2025): backtracking, uncertainty estimation, example testing, and adding knowledge. We prompt GPT-4o to locate token sequences of each behavior from the output tokens of our LRMs. To interpret different compression strategies, we adapt difference of means and attribution patching by computing the activation of every linear module in each layer. This allows us to compute the causal relationship between each weight matrix and our target reasoning behaviors.

Difference of Means. To compute the numerical representation in activation space of each reasoning behavior, we adapt difference of means method (Venhoff et al., 2025; Arditi et al., 2024) to extract the steering vector $\mathbf{u}_{m\ell}^c$ for each linear module m at layer ℓ for behavior c:

$$\mathbf{u}_{m\ell}^c = \frac{1}{|\mathcal{D}_+|}\sum_{s_i^c \in \mathcal{D}_+} \overline{\mathbf{a}}_{m\ell}^c(s_i^c) - \frac{1}{|\mathcal{D}_-|}\sum_{s_j \in \mathcal{D}_-} \overline{\mathbf{a}}_{m\ell}(s_j), \quad \text{with} \quad \overline{\mathbf{a}}_{m\ell}^c(s_i^c) = \frac{1}{|s_i^c|}\sum_{t \in s_i^c} \mathbf{a}_{m\ell}(t)$$

where s_i^c denotes the token sequence corresponding to a specific reasoning behavior c along with its five preceding tokens as output by an LRM, s_j is the token sequence of the entire LRM output (prompt and output tokens), $\mathcal{D}_+$ is the set of output instances containing at least one token sequence labeled with c, $\mathcal{D}_-$ is the set of all output instances, $\mathbf{a}_{m\ell}(t)$ is the activation of module m at layer ℓ at token t, $\overline{\mathbf{a}}_{m\ell}^c(s_i^c)$ is the average of $\mathbf{a}_{m\ell}(t)$ across all tokens in s_i^c, and similarly, $\overline{\mathbf{a}}_{m\ell}(s_j)$ is the average of $\mathbf{a}_{m\ell}(t)$ across all tokens in s_j. We then normalize $\mathbf{u}_{m\ell}^c$ to $\tilde{\mathbf{u}}_{m\ell}^c$: $\tilde{\mathbf{u}}_{m\ell}^c = \mathbf{u}_{m\ell}^c \cdot \frac{\left\|\overline{\mathbf{a}}_{m\ell}^{\text{all}}\right\|_2}{\left\|\mathbf{u}_{m\ell}^c\right\|_2}$ where $\overline{\mathbf{a}}_{m\ell}^{\text{all}}$ denotes the mean activation across all tokens in $\mathcal{D}_-$.

Attribution Patching. To find the causally relevant LRMs components with respect to each reasoning behavior, we adapt attribution patching (Syed et al., 2023) method to compute the importance score $\mathbf{I}^c_{m\ell}$ of each linear module.

$$\mathbf{I}^c_{m\ell} \approx \frac{1}{|\mathcal{D}_+|} \left| \sum_{s^c_i \in \mathcal{D}_+} \left(\tilde{\mathbf{u}}^c_{m\ell}\right)^\top \frac{\partial}{\partial \mathbf{a}_{m\ell}} \mathcal{L}(s^c_i) \right|$$

where $\mathcal{L}(s^c_i)$ is the cross-entropy loss of s^c_i. A higher $\mathbf{I}^c_{m\ell}$ means a stronger causal relationship between c and the linear module m at layer ℓ, helping us locate the most important weights responsible for reasoning capabilities (a fundamental problem for quantization and pruning works).

2.3 Decoding Compression Effects

To decode compression effects, we compute the relative importance $\mathbf{RI}^c_{m\ell}$ of each weight matrix ($\mathbf{I}^c_{m\ell}$ divided by $\sum_m \sum_\ell \mathbf{I}^c_{m\ell}$) and track how it changes because of compression (**importance shift**). Specifically, we measure the change of $\mathbf{RI}^c_{m\ell}$ from R1 distilled Llama-8B to original `meta-llama/Llama-3.1-8B` to understand distillation effect (Section 4). Likewise, the importance shift from the R1 distilled models to their quantized versions indicates quantization effect (Section 5). For the R1 distilled models, we also compute the $\mathbf{I}^c_{m\ell}$ of their weights to complement our findings on the distillation effect.

We hypothesize that the importance shift should be minimal in the ideal case, as a compressed LRM should remain as close as possible to its original counterpart (the more reasoning-capable model). **When visualizing the importance shift from an LRM to its compressed variant (or from a distilled model to its backbone), we only consider decreases in $\mathbf{RI}^c_{m\ell}$.** By definition, the relative importance of each weight matrix is normalized to sum to one, so any increase in relative importance necessarily compensates for decreases elsewhere. Since it is more informative to track cases where the $\mathbf{RI}^c_{m\ell}$ of a more reasoning-capable model decreases (*e.g.*, when the reasoning capability of a weight matrix is diminished), we set all increases in relative importance to zero.

2.4 Scope

We study three major LLMs compression paradigms, distillation, quantization, and pruning, making our scope comprehensive enough for investigating the effects of diverse compression methods. For distillation, we select four R1 distilled models: `DeepSeek-R1-Distill` Llama-70B, Qwen-32B, Llama-8B, and Qwen-7B. For quantization, we select 2.51-, 1.73-, and 1.58-bit models by Unsloth[2] as the choices of quantized R1 due to their popularity. We also evaluate AWQ Lin et al. (2024), GPTQ (Frantar et al., 2022), GPTAQ (Li et al., 2025), and ANY4/3 (Elhoushi & Johnson, 2025) as reproducible state-of-the-art quantization methods designed for relatively smaller LLMs (*e.g.*, the R1 distilled models). Specifically, we use all four methods to perform 4-bit quantization, and use GPTQ, GPTAQ, and ANY3 for 3-bit quantization as well, since many AWQ implementations do not support 3-bit. For pruning, we run SparseGPT Frantar & Alistarh (2023) on our two largest distilled models. We run interpretaion analysis on linear modules of all layers within LRMs.

2.5 Evaluation Setup

We select four reasoning datasets with varying levels of difficulty: AIME 2024 (Mathematical Association of America) for mathematical reasoning, FOLIO (Han et al., 2024) for logical reasoning, Temporal Sequences of BIG-Bench Hard (Suzgun et al., 2022) for temporal reasoning, and MuSiQue (Trivedi et al., 2022) for multihop reasoning. Since MuSiQue requires knowledge memorization besides multihop reasoning, we follow a closed-book setting (directly prompting LRMs to get final answers) to evaluate both reasoning and knowledge retention capabilities. Additional details of benchmarks, along with Table 5 that shows their statistics, are specified in Appendix C.

Accuracy metric is used for AIME 2024, FOLIO, and Temporal Sequences. We adopt exact match (EM) and F1 for MuSiQue. For each model (except R1 and those dynamically quantized LRMs), we run it three times and report its average scores to mitigate performance variability. Implementation details are in Appendix D.

[2]`https://huggingface.co/unsloth/DeepSeek-R1-GGUF`

Table 1: Benchmark performance of R1 and its compressed variants. All four benchmark scores are averaged over three passes, except the rows marked with †. Avg denotes the average scores shown in AIME 2024, FOLIO, and Temporal columns. We segment this table based on model families and mark the highest scores within each model family in **bold**.

Models			Accuracy				
Model	#Param	Compression	AIME 2024	FOLIO	Temporal	Avg	MuSiQue (EM, F1)
DeepSeek-R1†	671B	-	73.3	76.4	99.6	83.1	(**17.0**, **27.51**)
DeepSeek-R1†	671B	2.51-bit	**76.7**	77.8	**100.0**	**84.8**	(**17.0**, 24.43)
DeepSeek-R1†	671B	1.73-bit	66.7	**78.3**	99.6	81.5	(15.0, 22.11)
DeepSeek-R1†	671B	1.58-bit	66.7	75.4	94.0	78.7	(14.0, 22.34)
R1-Distill-Llama	70B	Distillation	65.6	**79.8**	**99.9**	**81.8**	(**13.3**, **21.57**)
R1-Distill-Llama	70B	Distillation & 50% sparse	23.3	71.6	97.6	64.2	(6.7, 13.49)
R1-Distill-Llama	70B	Distillation & 4-bit AWQ	63.4	78.5	99.3	80.4	(10.7, 19.23)
R1-Distill-Llama	70B	Distillation & 4-bit GPTQ	**66.7**	77.0	**99.9**	81.2	(10.3, 18.17)
R1-Distill-Llama	70B	Distillation & 4-bit GPTAQ	64.4	78.8	99.6	80.9	(12.0, **21.57**)
R1-Distill-Llama	70B	Distillation & 3-bit GPTQ	46.7	71.8	99.3	72.6	(4.7, 11.92)
R1-Distill-Llama	70B	Distillation & 3-bit GPTAQ	54.4	77.3	99.7	77.1	(5.7, 13.21)
R1-Distill-Qwen	32B	Distillation	64.4	82.3	**99.9**	82.2	(2.7, 10.95)
R1-Distill-Qwen	32B	Distillation & 50% sparse	25.6	75.1	97.9	66.2	(2.3, 9.01)
R1-Distill-Qwen	32B	Distillation & 4-bit AWQ	67.8	82.3	99.1	**83.1**	(3.3, 10.28)
R1-Distill-Qwen	32B	Distillation & 4-bit GPTQ	**68.9**	80.6	99.6	83.0	(4.0, 11.78)
R1-Distill-Qwen	32B	Distillation & 4-bit GPTAQ	63.3	81.5	99.7	81.5	(2.7, 11.88)
R1-Distill-Qwen	32B	Distillation & 4-bit ANY4	**68.9**	78.0	99.7	82.2	(**5.7**, **12.68**)
R1-Distill-Qwen	32B	Distillation & 3-bit GPTQ	44.4	74.2	98.9	72.5	(4.0, 11.55)
R1-Distill-Qwen	32B	Distillation & 3-bit GPTAQ	45.6	77.5	99.5	74.2	(2.3, 9.18)
R1-Distill-Qwen	32B	Distillation & 3-bit ANY3	53.3	**82.6**	**99.9**	78.6	(3.7, 10.27)
R1-Distill-Llama	8B	Distillation	42.2	**71.9**	81.5	65.2	(0.0, 4.43)
R1-Distill-Llama	8B	Distillation & 4-bit AWQ	**47.8**	68.0	84.0	**66.6**	(**0.3**, **5.05**)
R1-Distill-Llama	8B	Distillation & 4-bit GPTQ	42.2	66.2	65.9	58.1	(**0.3**, 4.68)
R1-Distill-Llama	8B	Distillation & 4-bit GPTAQ	40.0	66.4	69.3	58.6	(0.0, 3.73)
R1-Distill-Llama	8B	Distillation & 4-bit ANY4	41.1	68.5	**88.7**	66.1	(0.0, 3.54)
R1-Distill-Llama	8B	Distillation & 3-bit GPTQ	11.1	65.0	67.3	47.8	(0.0, 2.89)
R1-Distill-Llama	8B	Distillation & 3-bit GPTAQ	7.8	65.5	57.2	43.5	(0.0, 3.45)
R1-Distill-Llama	8B	Distillation & 3-bit ANY3	3.3	50.1	34.9	29.4	(0.7, 2.35)
R1-Distill-Qwen	7B	Distillation	46.7	**78.0**	75.6	**66.8**	(0.0, 3.57)
R1-Distill-Qwen	7B	Distillation & 4-bit AWQ	46.6	75.5	74.9	65.7	(0.0, 3.14)
R1-Distill-Qwen	7B	Distillation & 4-bit GPTQ	38.9	72.9	70.3	60.7	(**1.0**, **4.27**)
R1-Distill-Qwen	7B	Distillation & 4-bit GPTAQ	**47.8**	74.4	67.7	63.3	(0.0, 3.96)
R1-Distill-Qwen	7B	Distillation & 4-bit ANY4	**47.8**	75.6	**77.1**	**66.8**	(0.0, 3.05)
R1-Distill-Qwen	7B	Distillation & 3-bit GPTQ	17.8	65.7	31.7	38.4	(0.0, 3.12)
R1-Distill-Qwen	7B	Distillation & 3-bit GPTAQ	24.4	64.5	48.7	45.9	(0.0, 3.06)
R1-Distill-Qwen	7B	Distillation & 3-bit ANY3	32.2	69.3	30.1	43.9	(0.0, 3.89)

3 Compression Effects on Reasoning Performance

3.1 Overall Performance

The overall performance of R1 and its compressed variants are in Table 1. We show the performance of pruned `R1-Distill-Llama-70B` and `R1-Distill-Qwen-32B` under 50% sparsity in Table 1, as it is the default sparsity level of current works (Zhang et al., 2024; Sun et al., 2023).

Comparing Compression Strategies. In Table 1, the 2.51-bit R1 achieves the highest average accuracy overall, since it has the smallest compression ratio. Both R1 distilled Llama-70B and Qwen-32B reach close-to-R1 accuracy scores. On MuSiQue, the 2.51-bit R1 also achieves performance close to original R1. Therefore, 2.51-bit R1 has the best overall performance than other compression strategies. Although R1 may be over-parameterized, a compression method with a smaller ratio can still offer advantages over methods with higher compression ratios. In contrast, pruning only 50% of the weights causes significant degradation, rendering the pruned LRMs unusable. Thus, we do not interpret the effect of pruning in the later sections. As for all distillation-only models, Qwen delivers stronger reasoning performance than Llama (Appendix E).

Comparing Benchmark Difficulties. Comparing the scores using R1 distilled Llama-70B as the backbone on AIME 2024, FOLIO, and Temporal, we see the largest score decrease on AIME 2024. This indicates that AIME 2024 is more challenging than the other two accuracy-based benchmarks.

Table 2: Performance of two distilled models under various sparsity levels. We report the one-pass scores for all models in this table.

Models			Accuracy				
Model	#Param	Sparsity	AIME 2024	FOLIO	Temporal	Avg	MuSiQue (EM, F1)
R1-Distill-Llama	70B	0%	63.3	78.8	100.0	80.7	(13.0, **21.80**)
R1-Distill-Llama	70B	10%	60.0	81.3	99.6	80.3	(12.0, 21.69)
R1-Distill-Llama	70B	30%	63.3	79.3	99.6	80.7	(**14.0**, 21.40)
R1-Distill-Llama	70B	40%	56.7	73.9	98.8	76.8	(6.0, 13.79)
R1-Distill-Llama	70B	50%	26.7	70.9	97.2	64.9	(6.0, 12.75)
R1-Distill-Llama	70B	60%	0.0	65.0	95.6	53.5	(0.0, 6.42)
R1-Distill-Llama	70B	70%	0.0	49.8	15.6	21.8	(0.0, 2.23)
R1-Distill-Llama	70B	80%	0.0	11.8	12.4	8.1	(0.0, 0.94)
R1-Distill-Qwen	32B	0%	66.7	82.3	100.0	83.0	(1.0, 9.38)
R1-Distill-Qwen	32B	10%	70.0	81.3	100.0	83.8	(**5.0**, **13.19**)
R1-Distill-Qwen	32B	30%	56.7	81.3	100.0	79.3	(1.0, 10.47)
R1-Distill-Qwen	32B	40%	53.3	78.3	100.0	77.2	(2.0, 10.16)
R1-Distill-Qwen	32B	50%	30.0	75.4	96.0	67.1	(3.0, 9.29)
R1-Distill-Qwen	32B	60%	0.0	65.0	87.2	50.7	(0.0, 4.13)
R1-Distill-Qwen	32B	70%	0.0	32.5	19.6	17.4	(0.0, 1.72)
R1-Distill-Qwen	32B	80%	0.0	8.7	2.0	3.6	(0.0, 1.29)

MuSiQue is also difficult in terms of knowledge requirement, because its scores in Table 1 are much lower than RAG (retrieval-augmented generation) setup (Zhang et al., 2025). This suggests that existing LRMs lack sufficient knowledge for knowledge-intensive tasks, making RAG a more suitable approach.

Takeaway 3.1 for Overall Performance

Considering over-parameterization, methods with smaller compression ratios can still offer advantages over those with higher compression ratios. Regardless of whether compression is applied, LRMs lack sufficient knowledge for knowledge-intensive tasks.

3.2 Collapse Point

We investigate whether LRMs degrade as they undergo increasing levels of compression. In Table 1, the performance of dynamically quantized LRMs steadily declines as we move from 2.51 to 1.58-bit, but we do not observe a clear collapse point. All 4-bit AWQ, GPTQ, GPTAQ, and ANY4 reach performance similar to their unquantized counterparts, which shows the effectiveness of existing 4-bit quantization on LRMs. However, 3-bit GPTQ, GPTAQ, and ANY3 display signs of collapse, indicating bottlenecks of current 3-bit quantization. GPTAQ and ANY4 are newer than AWQ and GPTQ, and they achieve similar performance on 4-bit LRMs. Based on average accuracy, it is noteworthy that GPTAQ surpasses GPTQ on 3-bit LRMs for three out of four distilled models. Regarding distillation, R1 distilled Llama-8B and Qwen-7B achieve the lowest accuracy among all distillation-only models. Only R1 distilled Llama-70B yields decent MuSiQue scores.

Table 2 displays performance of our two distilled models under various sparsity levels. Comparing distilled models with their sparsified variants, we find the precise collapse points of our pruned LRMs. Interestingly, their collapse points correlate to the benchmark difficulty. For example, on AIME 2024, `R1-Distill-Llama` collapses between 40% and 50% sparsity, since its performance drops by more than half. However, its collapse points on FOLIO and Temporal are roughly between 60% and 70% sparsity, which occur much later than AIME 2024. The correlation between collapse point and benchmark difficulty can also be seen on the sparsified Qwen.

Takeaway 3.2 for Collapse Point

Collapse point correlates with benchmark difficulty. On hard benchmarks, 3-bit quantization and pruning with 50% sparsity or higher still have substantial room for improvement.

3.3 Compression Impact on Knowledge and Reasoning

In Table 2, although Qwen demonstrates stronger reasoning capabilities than Llama, it has significantly lower EM and F1 scores on MuSiQue. Because MuSiQue requires knowledge memorization

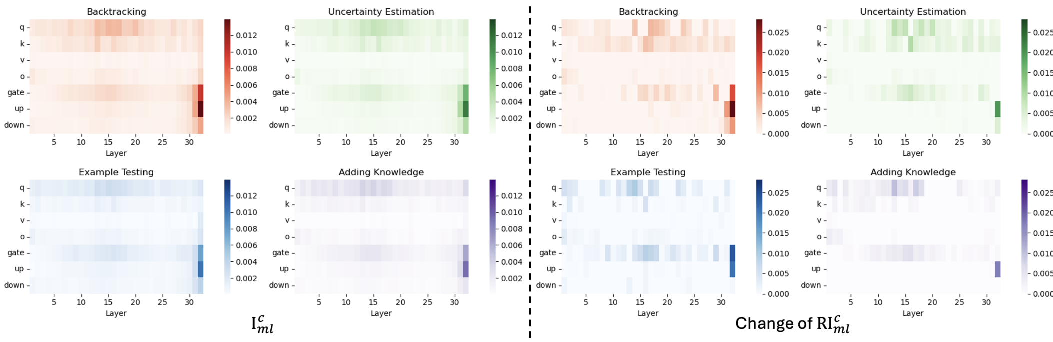

Figure 2: $\mathbf{I}^c_{m\ell}$ of `DeepSeek-R1-Distill-Llama-8B` (**left**) and change of $\mathbf{RI}^c_{m\ell}$ from `DeepSeek-R1-Distill-Llama-8B` to `Llama-3.1-8B` (**right**). Each heatmap displays scores of importance (or importance shift) of every module at each layer, providing a fine-grained analysis of weight contributions to the corresponding reasoning capability. On the right, increases in $\mathbf{RI}^c_{m\ell}$ are set to 0, as they only offset decreases elsewhere as discussed in 2.3. Every cluster of 4 side-by-side heatmaps (including those displayed below) follow the same scaling to show the precise magnitude of each weight module.

under the closed-book setting, the smaller parameter count of Qwen puts itself at a disadvantaged position. In other words, models' parameter count affects knowledge more than reasoning. In addition, we notice pruned `R1-Distill-Llama-70B` collapses between 30% and 40% sparsity on MuSiQue, which is even earlier than on AIME 2024. This shows pruning hurts LRMs' knowledge memorization more than quantization. When a compression method aggressively removes the weights of an LRM, it is expected that the model's knowledge will be more seriously affected. This phenomenon can also be seen on our dynamically quantized models in Table 1. Since quantization preserves parameter count and our analysis above shows that many quantized models still retain competitive reasoning capability, quantization is recommended on knowledge-intensive tasks.

Takeaway 3.3 for Compression Impact on Knowledge and Reasoning

Pruning and distillation compress knowledge retention more than reasoning capabilities.

4 Distillation Effect on Weights

To study the effect of distillation on weights, we compute $\mathbf{I}^c_{m\ell}$ of two distilled R1 models and further measure the change of $\mathbf{RI}^c_{m\ell}$ as discussed in Section 2.3.

4.1 Locating Important Weights

The left part of Figure 2 presents the weight importance of R1 distilled Llama-8B in four heatmaps, each corresponding to a reasoning behavior. We observe that the final layer houses several most important linear modules across all four behaviors, with the highest value located at `up_proj`. Therefore, the `up_proj` in the final layer (`32_up`) stands out as the most important component.

Interestingly, this finding generalizes to R1 distilled Qwen-7B, as we also observe this `up_proj` outlier in the final layer of Qwen in Figure 4. Notably, our finding complements a recent analysis (Shao & Wu, 2025), which claims the most important module for reasoning is `o_proj`. Since identifying important weights is a core research problem of compression methodologies, our finding is valuable for future works.

Takeaway 4.1 for Locating Important Weights

Distillation makes `up_proj` in the final layer as the most important module for reasoning behaviors, as observed in both R1 distilled Llama and Qwen models.

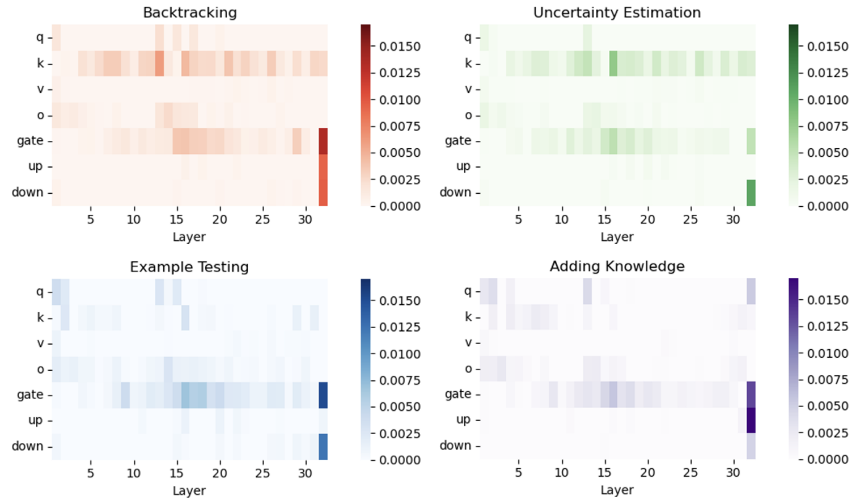

Figure 3: Change of $\mathbf{RI}^c_{m\ell}$ from `DeepSeek-R1-Distill-Llama-8B` to its 4-bit AWQ variant.

Table 3: Accuracy after selectively quantizing a single component of R1 distilled Llama-8B (*e.g.*, 1_up means to only quantize the `up_proj` in the first layer) to 3-bit. Ranking of a component is based its $\sum_c \mathbf{I}_{m\ell}$, so "2nd col" refers to second place within its column across all four heatmaps (each column consists of 7 linear modules of a layer). "1st overall" means the global highest ranking.

Quantized Component	Rank	AIME 2024	FOLIO	Temporal	Avg
32_up	1st overall	20.0	63.1	63.6	48.9
32_gate	2nd col	33.3	62.1	67.2	54.2
32_v	last col	43.3	68.0	79.6	63.6
31_up	2nd row	33.3	70.0	64.4	55.9
1_up	last row	6.7	64.5	80.4	50.5

4.2 Validating Importance Scores

We validate Section 4.1 by applying 3-bit round-to-nearest quantization to either `32_up` or a component sharing its column or row in the heatmaps, then measuring the resulting accuracy drop (Table 3). The more important a component is, the greater the accuracy drop when it is quantized. Specifically, we select four additional component candidates: the second- and last-ranked modules among the seven in the final layer, and the second- and last-ranked layers across all 32 layers of the up-projection. We see that 32_up yields the lowest average accuracy, which clearly demonstrates the validity of our findings in Section 4.1. It is quite salient that quantizing only this matrix (merely 0.7% of all weights) reduces the average accuracy by 16.3%. The component rank generally correlates with the accuracy drop, except for 1_up, which incurs the lowest accuracy on AIME 2024.

4.3 Importance Shift via Distillation

Since `R1-Distill-Llama-8B` is fine-tuned based on `Llama-3.1-8B`, we compute the change of $\mathbf{RI}^c_{m\ell}$ to visualize distillation effect in the right part of Figure 2 (for Llama) and 5 (for Qwen). Both parts of Figure 2 exhibit similar patterns (*e.g.*, most outliers are in the final layer), indicating that the important weights of the distilled model are primarily the result of distillation with SFT, while the original Llama's weight values play little role in shaping its reasoning capabilities. Thus, distillation effect is quite powerful in transforming a non-reasoning LLM into an LRM. For Qwen, Figures 4 and 5 also show similar patterns, so the utility of distillation generalizes to Qwen as well.

Takeaway 4.3 for Importance Shift via Distillation

Important weights of the R1 distilled models are mainly the result of the distillation effect.

Table 4: Performance of 3-bit AWQ and selectively protecting the MLP modules in the final layer.

Model	Compression	Full-Precision Anywhere?	AIME 2024	FOLIO	Temporal	Avg	MuSiQue
R1-Distill-Llama-8B	3-bit AWQ	-	10.0	59.6	68.4	46.0	(0.0, 3.50)
R1-Distill-Llama-8B	3-bit AWQ	Final-layer MLP	**16.7**	**67.0**	**74.0**	**52.57**	**(1.0, 3.62)**

5 Quantization Effect on Weights

For quantization effect on weights, we analyze the decrease of importance shift during quantization.

5.1 Locating Quantization Effect

We show heatmaps to visualize the importance shift from R1 distilled Llama-8B to its 4-bit AWQ quantized variant in Figure 3. Across all four heatmaps, we observe a reduction in the significance of the gate projections in the middle layers (*e.g.*, layer 9 to 23), suggesting that AWQ may overly compress these modules. Moreover, most linear modules in the final layer are compressed to the greatest extent, which shows the drawback of AWQ. Since `32_up` is the most important module as discussed in Section 4.2 and its importance shift is little for uncertainty estimation and example testing capabilities, AWQ successfully preserves its significance on these two behaviors. However, for backtracking and adding knowledge capabilities, AWQ is not effective at maintaining its importance.

In Figure 6, we visualize the importance shift from R1 distilled Qwen-7B to its 4-bit AWQ quantized version. We also see a shift in the importance of the gate projections on Qwen, but this shift mainly occurs in the early layers (*e.g.*, layer 1 to 10). On Qwen, AWQ does not preserve the importance of `32_up` across all four reasoning capabilities, and it also overly compresses `32_k` on two capabilities.

As another popular method, we interpret the effect of 4-bit GPTQ in Figure 7. On R1 distilled Llama-8B, we observe similar quantization effect as AWQ, since GPTQ also overly compresses final-layer modules and the gate projections in the middle layers. Their commonality demonstrates the generality of the bottlenecks we identified in existing quantization methods.

Takeaway 5 for Quantization Effect on Weights

State-of-the-art quantization methods fail to preserve the importance of the MLP gate projections and the final layer, which is a key bottleneck of performance improvement.

5.2 Validating Quantization Effect

To validate our findings about the bottleneck of current quantization, we design a simple protection mechanism using a mixed precision fashion. We run two versions of 3-bit AWQ in Table 4. In the first version, we run AWQ with their default calibration data. Since we know AWQ overly compresses the MLP modules in the final layer, we then choose to protect them by changing their quantized weights to their original values in 16-bit. If they are truly important yet not well protected by AWQ, our protection mechanism should offer a significant improvement. Based on the discussion in Sections 3.1 and 3.2, we perform 3-bit quantization, since it will be the focus of future works.

We see our selective protection boosts 3-bit AWQ on all benchmarks, with an average accuracy improvement of 6.57%. This is particularly significant given that only about 2% of all weights remain in 16-bit. This mixed precision model outperforms all 3-bit quantization baselines in Table 1 by at least 4.77% in average accuracy, with gains of up to 23.17%. Therefore, our findings are demonstrated with an indication of substantial room for further improvement. Note that our protection provides relatively marginal increase on MuSiQue, as the weight count stays the same (Section 3.3).

6 Conclusion and Future Directions

We study the effects of LLMs compression on LRMs and present key findings for further imrpving LRMs compression. Future compression works are encouraged to consider the protection of MLP up projection in the final layer. The excessive compression of current quantization methods on MLP gate projections and final-layer modules highlights the need for preserving these weight modules.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction, 2024. URL https://arxiv.org/abs/2406.11717.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. SliceGPT: Compress large language models by deleting rows and columns. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vXxardq6db.

Guangji Bai, Yijiang Li, Chen Ling, Kibaek Kim, and Liang Zhao. SparseLLM: Towards global pruning of pre-trained language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=oXHyYHp4Zb.

Maciej Besta, Julia Barth, Eric Schreiber, Ales Kubicek, Afonso Catarino, Robert Gerstenberger, Piotr Nyczyk, Patrick Iff, Yueling Li, Sam Houliston, Tomasz Sternal, Marcin Copik, Grzegorz Kwaśniewski, Jürgen Müller, Łukasz Flis, Hannes Eberhard, Hubert Niewiadomski, and Torsten Hoefler. Reasoning language models: A blueprint, 2025. URL https://arxiv.org/abs/2501.11223.

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. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Michael Han Daniel Han and Unsloth team. Unsloth, 2023. URL http://github.com/unslothai/unsloth.

DaYou Du, Yijia Zhang, Shijie Cao, Jiaqi Guo, Ting Cao, Xiaowen Chu, and Ningyi Xu. Bitdistiller: Unleashing the potential of sub-4-bit llms via self-distillation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 102–116, 2024.

Mostafa Elhoushi and Jeff Johnson. any4: Learned 4-bit numeric representation for llms, 2025. URL https://arxiv.org/abs/2507.04610.

Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. Efficient reasoning models: A survey, 2025. URL https://arxiv.org/abs/2504.10903.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pp. 10323–10337. PMLR, 2023.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training compression for generative pretrained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024. URL https://zenodo.org/records/12608602.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models, 2024. URL https://arxiv.org/abs/2306.08543.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander Fabbri, Wojciech Maciej Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. FOLIO: Natural language reasoning with first-order logic. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22017–22031, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1229. URL https://aclanthology.org/2024.emnlp-main.1229/.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14852–14882, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.830. URL https://aclanthology.org/2023.acl-long.830/.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Sophia Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *Advances in Neural Information Processing Systems*, 37:1270–1303, 2024.

Yukun Huang, Yanda Chen, Zhou Yu, and Kathleen McKeown. In-context learning distillation: Transferring few-shot learning ability of pre-trained language models. *arXiv preprint arXiv:2212.10670*, 2022.

Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. O1 replication journey–part 2: Surpassing o1-preview through simple distillation, big progress or bitter lesson? *arXiv preprint arXiv:2411.16489*, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, Jiuxiang Gu, and Tianyi Zhou. Selective reflection-tuning: Student-selected data recycling for LLM instruction-tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 16189–16211, Bangkok, Thailand and virtual meeting, August 2024a. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-acl.958.

Shiyang Li, Jianshu Chen, yelong shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xifeng Yan. Explanations from large language models make small reasoners better. In *2nd Workshop on Sustainable AI*, 2024b. URL https://openreview.net/forum?id=rH8ZUcfL9r.

Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. Gptaq: Efficient finetuning-free quantization for asymmetric calibration, 2025. URL https://arxiv.org/abs/2504.02692.

Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. Less is more: task-aware layer-wise distillation for language model compression. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6: 87–100, 2024.

Ruikang Liu, Yuxuan Sun, Manyi Zhang, Haoli Bai, Xianzhi Yu, Tiezheng Yu, Chun Yuan, and Lu Hou. Quantization hurts reasoning? an empirical study on quantized reasoning models, 2025. URL https://arxiv.org/abs/2504.04823.

Shih-yang Liu, Zechun Liu, Xijie Huang, Pingcheng Dong, and Kwang-Ting Cheng. LLM-FP4: 4-bit floating-point quantized transformers. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 592–605, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.39. URL https://aclanthology.org/2023.emnlp-main.39/.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. Llm-qat: Data-free quantization aware training for large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 467–484, 2024a.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://openreview.net/forum?id=L057s2Rq8O.

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

Mathematical Association of America. MAA Invitational Competitions. URL https://maa.org/maa-invitational-competitions/. Accessed: 2025-03-19.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan

Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

Jie Shao and Jianxin Wu. Who reasons in the large language models?, 2025. URL https://arxiv.org/abs/2505.20993.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8Wuvhh0LYW.

Gaurav Srivastava, Shuxiang Cao, and Xuan Wang. Towards reasoning ability of small language models, 2025. URL https://arxiv.org/abs/2502.11569.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022. URL https://arxiv.org/abs/2210.09261.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery, 2023. URL https://arxiv.org/abs/2310.10348.

Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown. *Hugging Face*, 2024.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multi-hop questions via single-hop question composition, 2022. URL https://arxiv.org/abs/2108.00573.

Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding reasoning in thinking language models via steering vectors. In *Workshop on Reasoning and Planning for Large Language Models*, 2025. URL https://openreview.net/forum?id=OwhVWNOBcz.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. Bitnet: Scaling 1-bit transformers for large language models, 2023. URL https://arxiv.org/abs/2310.11453.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=09iOdaeOzp.

Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, and Wanxiang Che. Onebit: Towards extremely low-bit large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=ZwiG9KjfHV.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*, 35:27168–27183, 2022.

Nan Zhang, Yanchi Liu, Xujiang Zhao, Wei Cheng, Runxue Bao, Rui Zhang, Prasenjit Mitra, and Haifeng Chen. Pruning as a domain-specific LLM extractor. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 1417–1428, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.91. URL https://aclanthology.org/2024.findings-naacl.91/.

Nan Zhang, Prafulla Kumar Choubey, Alexander Fabbri, Gabriel Bernadett-Shapiro, Rui Zhang, Prasenjit Mitra, Caiming Xiong, and Chien-Sheng Wu. SireRAG: Indexing similar and related information for multihop reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yp95goUAT1.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12: 1556–1577, 2024. doi: 10.1162/tacl_a_00704. URL https://aclanthology.org/2024.tacl-1.85/.

A Use of LLMs

To improve the overall clarity of our writing, we used ChatGPT-4o and ChatGPT-5 via OpenAI's web interface to polish a small fraction of sentences. LLMs were not used in any steps of the research ideation process. To ensure correctness and precision, we carefully reviewed and adapted all LLM-generated content before incorporating it into our writing.

B Related Work

Our literature review is conducted over existing compression methodologies (quantization, distillation, and pruning) and recent LRMs.

B.1 Quantization

Quantization reduces the number of bits used to represent LLM weights, thereby lowering their precision (Srivastava et al., 2025). Recent survey (Zhu et al., 2024) categorizes quantization methodologies into quantization-aware training (QAT) and post-training quantization (PTQ). QAT requires retraining of model weights to recover performance loss during quantization while PTQ does not require retraining. Recent QAT includes LLM-QAT (Liu et al., 2024a) that adopts distillation to train a quantized LLM, BitDistiller (Du et al., 2024) that develops a self-distillation approach for the full-precision model to act as the teacher of its low-bit counterpart, BitNet (Wang et al., 2023) that proposes a 1-bit Transformer architecture for training LLMs from scratch, and OneBit (Xu et al., 2024) that quantizes LLM weight matrices to 1-bit from a knowledge transfer perspective.

PTQ is more popular in terms of the number of recent publications, because there is no retraining involved. For example, GPTQ (Frantar et al., 2022) and GPTAQ (Li et al., 2025) are one-shot weight quantization methods that use approximate second-order information, while AWQ (Lin et al., 2024) leverages activation distribution for finding the salient weight channels to skip. Other PTQ methods include weight-activation quantization (Shao et al., 2024; Yao et al., 2022; Liu et al., 2023) and KV cache quantization (Hooper et al., 2024; Liu et al., 2024b).

B.2 Distillation

Distillation involves two settings: black-box and white-box settings. For black-box setting, teacher model is typically a closed-source LLM and only the outputs of teacher are available for student model. For white-box setting, both weights and output distribution of the teacher model are available. Existing black-box distillation (Huang et al., 2024; Li et al., 2024b; Ho et al., 2023; Huang et al., 2022; Li et al., 2024a) prompts the teacher model to generate a training dataset for the student to learn. Specifically, researchers have started to distill OpenAI's O1 model (Huang et al., 2024), which marks the beginning of LRMs compression. White-box distillation allows the student model

Table 5: Dataset statistics of selected reasoning benchmarks.

	Size	Answer Type	Metric	Knowledge
AIME 2024	30	Integer	Accuracy	False
FOLIO	203	True/False/Uncertain	Accuracy	False
Temporal	250	(A)/(B)/(C)/(D)	Accuracy	False
MuSiQue	100	A few words	(EM, F1)	True

to learn from teacher's knowledge representation. Works has been done to align the output distribution (Agarwal et al., 2024; Gu et al., 2024) or the hidden representation (Liang et al., 2023) between teacher and student models.

B.3 Pruning

There are unstructured and structured pruning. For unstructured pruning, individual weights are targeted, which leads to irregular sparsity structure. In contrast, structured pruning involves removing entire network components such as channels or layers (Zhang et al., 2024). Unstructured pruning usually has better compression performance than structured pruning, while it is easier to achieve inference speedup via structured methods (Zhu et al., 2024). Recent unstructured pruning includes one-shot pruning (Frantar & Alistarh, 2023; Sun et al., 2023), global pruning that makes pruning decisions based on all layers (Bai et al., 2024), and domain-specific pruning (Zhang et al., 2024). Structured pruning includes gradient-based (Xia et al., 2024; Ma et al., 2023) and non-gradient-based (Ashkboos et al., 2024) methods.

B.4 LRMs

Trained with reinforcement learning, LRMs extends LLMs with advanced reasoning mechanisms (Besta et al., 2025). Popular closed-source LRMs are OpenAI's o1-mini, o1 (OpenAI et al., 2024), and o3-mini. Open-source LRMs include DeepSeek-R1 and QwQ-32B-Preview (Team, 2024). Since quantization, white-box distillation, and pruning methods require access to model weights, they are not suitable for closed-source LRMs. Only black-box distillation will work on closed-source models.

C Additional Details of Reasoning Benchmarks

Table 5 shows the statistics of our selected benchmarks. AIME 2024[3] (parts I and II) represents top match challenges, and its answers are integers. FOLIO[4] requires logical deductions to determine whether the provided conclusion is true, false, or uncertain based on premise. In Temporal Sequences[5], models are asked to use a provided timeline to determine what time a person might be free to perform another activity. Since each of its questions comes with four options, we expect our models to output the index (the letter) of the selected option. Since MuSiQue involves question answering and its answers are in a few words, we adopt exact match (EM) and F1. We randomly sample 100 questions out of 1000 from MuSiQue for our benchmarking analysis.

D Implementation Details

We run the dynamically quantized models on `llama.cpp`[6] based on their requirement. We run all other distilled, pruned, and quantized models on vLLM (Kwon et al., 2023) for its fast inference. In order to comprehensively analyze performance change after compression, we also evaluate R1 on our reasoning benchmarks by using DeepSeek API. Aligning with DeepSeek-R1 report (Guo et al.,

[3]https://huggingface.co/datasets/Maxwell-Jia/AIME_2024

[4]https://huggingface.co/datasets/yale-nlp/FOLIO

[5]https://github.com/suzgunmirac/BIG-Bench-Hard/blob/main/bbh/temporal_sequences.json

[6]https://github.com/ggml-org/llama.cpp

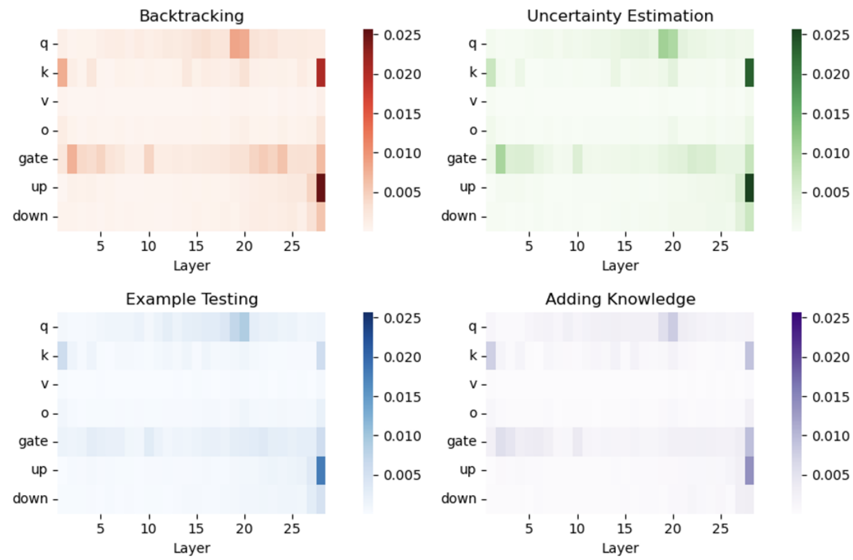

Figure 4: $\mathbf{I}^c_{m\ell}$ of `DeepSeek-R1-Distill-Qwen-7B`.

2025), we keep the same parameters for all models during inference: maximum generation length is set to 32768, temperature is set to 0.6, and top-p value is set to 0.95.

We use AutoAWQ[7] as the AWQ implementation for inference due to its speed advantage (vLLM support), while the original AWQ code[8] is used to generate the pseudo-quantized R1 distilled Llama-8B for our analysis in Section 5.

We focus on analyzing the effect of compression methods on performance and thus do not consider inference speedup. The reason is that these methods run on different inference platforms, so it is hard to control the consistency of inference optimization across various platforms.

E Comparing Distilled Models

On accuracy-based benchmarks of Table 1, we see that R1 distilled Qwen-32B delivers an average 0.4% improvement over Llama-70 and R1 distilled Qwen-7B delivers an average 1.6% improvement over Llama-8B. Although these two Qwen models have less weights, Qwen delivers stronger reasoning performance than Llama. This phenomenon aligns with DeepSeek report (Guo et al., 2025). However, `R1-Distill-Qwen-32B` scores significantly lower than `R1-Distill-Llama-70B` on MuSiQue, highlighting its worse ability of memorization.

F Additional Visualization of Weight Importance and Importance Shift

All additional figures in Appendix are thoroughly discussed in the main content. Figure 4 shows the weight importance of `DeepSeek-R1-Distill-Qwen-7B` across four heatmaps, each corresponding to a specific target reasoning behavior. Figure 5 displays the change of $\mathbf{RI}^c_{m\ell}$ from `DeepSeek-R1-Distill-Llama-8B` to `Qwen2.5-Math-7B`. To decode the quantization effect on Qwen, Figure 6 shows the change of $\mathbf{RI}^c_{m\ell}$ from `DeepSeek-R1-Distill-Qwen-7B` to its 4-bit AWQ variant. Similarly, Figure 7 shows the change of $\mathbf{RI}^c_{m\ell}$ from R1 distilled Llama-8B to its 4-bit GPTQ quantized variant.

[7]https://github.com/casper-hansen/AutoAWQ
[8]https://github.com/mit-han-lab/llm-awq

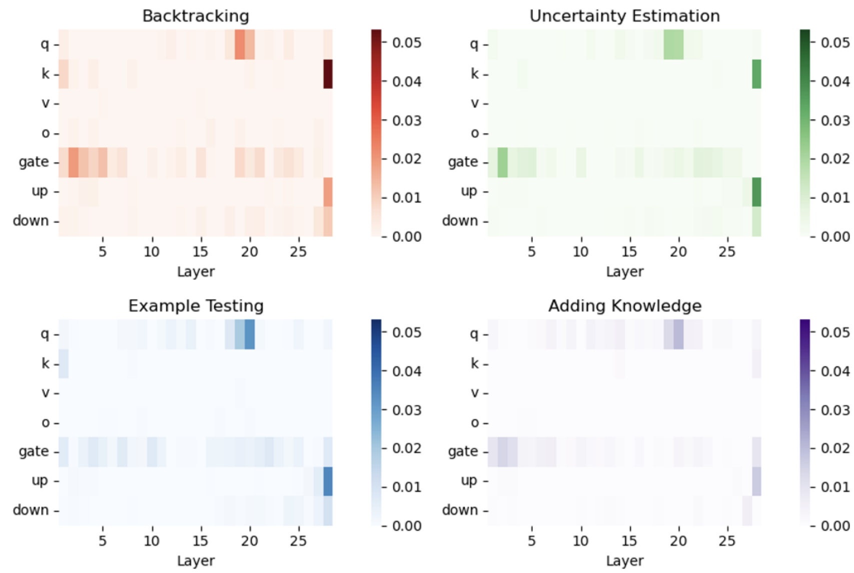

Figure 5: Change of $\mathbf{RI}^{c}_{m\ell}$ from `DeepSeek-R1-Distill-Llama-8B` to `Qwen2.5-Math-7B` (the backbone model).

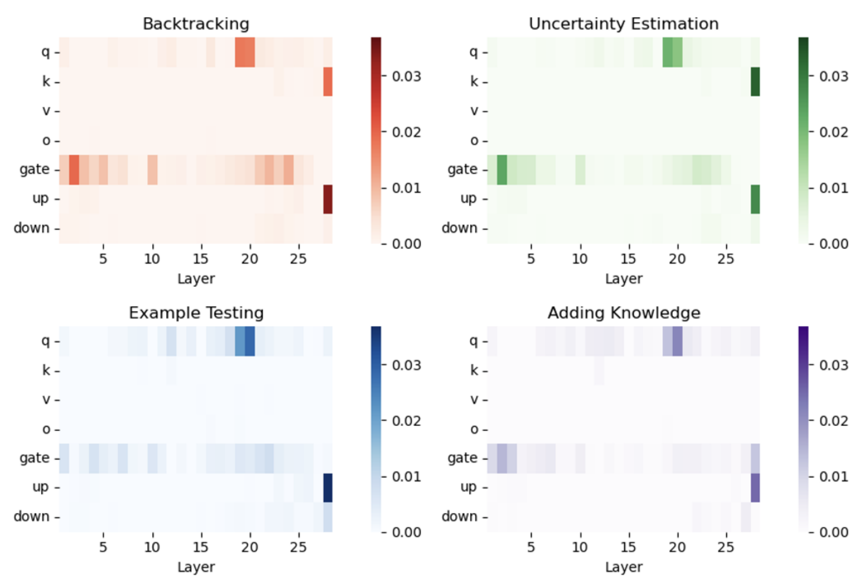

Figure 6: Change of $\mathbf{RI}^{c}_{m\ell}$ from `DeepSeek-R1-Distill-Qwen-7B` to its 4-bit AWQ variant.

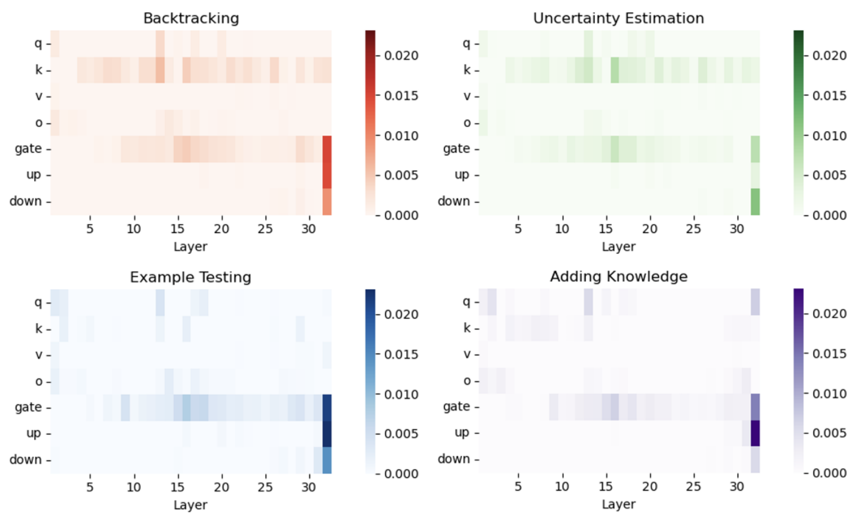

Figure 7: Change of $\mathbf{RI}^{c}_{m\ell}$ from `DeepSeek-R1-Distill-Llama-8B` to its 4-bit GPTQ quantized variant.