
Beyond Proxy Metrics: A New Evaluation Framework for LLM Compression by Directly Measuring Generative Faithfulness

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Existing evaluation frameworks for Large Language Model (LLM) compression
2 rely on proxy metrics, such as Perplexity and curated benchmarks, leading to a
3 significant gap between reported scores and real-world performance. These are
4 considered “proxy” metrics because they measure indirect qualities like statistical
5 likelihood or accuracy on narrow, multiple-choice tasks, instead of the core objective:
6 faithfulness to the original model’s generative behavior. To bridge this gap,
7 we introduce a new evaluation paradigm that abandons proxies to directly measure
8 generative faithfulness. Our framework has two core innovations: (1) Conditional
9 Generation Accuracy (CGA), a metric that evaluates a compressed model’s ability
10 to replicate an original model’s next token predictions under a teacher-forcing
11 regime, and (2) an evaluation set of real user queries, ensuring alignment with
12 practical applications. Using this framework, we conduct a large scale evaluation
13 of 9 mainstream compression techniques across model sizes (7B–32B) and context
14 lengths (8K–24K). Our findings challenge prior claims. For instance, on a 7B
15 model, KV cache dropping methods severely underperform (CGA score of 0.57),
16 while a Top-10% sparse attention baseline (CGA 0.943) unexpectedly surpasses
17 popular INT4 quantization techniques like GPTQ (CGA 0.921). Moreover, we
18 uncover distinct scaling laws: quantization accuracy degrades with longer contexts,
19 whereas the fidelity of sparse attention improves. We release code, dataset and
20 leaderboard of our benchmark at anonymous Github.

1 Introduction

21 The rapid adoption of Large Language Models (LLMs) has created an urgent demand for efficient
22 inference under constrained GPU resources [25, 36, 53]. Model compression has emerged as a
23 promising solution [54, 35, 45], with many training-free or calibration-light approaches [11, 28, 10]
24 claiming substantial inference speedups with only marginal accuracy loss. However, a significant gap
25 exists between these claims and real-world performance. Qualitative analysis (Appendix A.1) reveals
26 that compressed models, such as Qwen2.5-7B-Instruct [48] with AWQ [28], are prone to degradation
27 including repetition, nonsensical outputs, and impaired instruction following capabilities. Critically,
28 these failures are consistently missed by standard benchmarks [44, 46, 43], which often report the
29 performance of these methods as nearly lossless. This stark discrepancy highlights an urgent need to
30 re-evaluate the current evaluation paradigms for LLM compression.
31

32 Current evaluation practices predominantly rely on Perplexity (PPL) and Question Answering (QA)
33 benchmarks [20, 44, 4], both of which suffer from fundamental limitations.

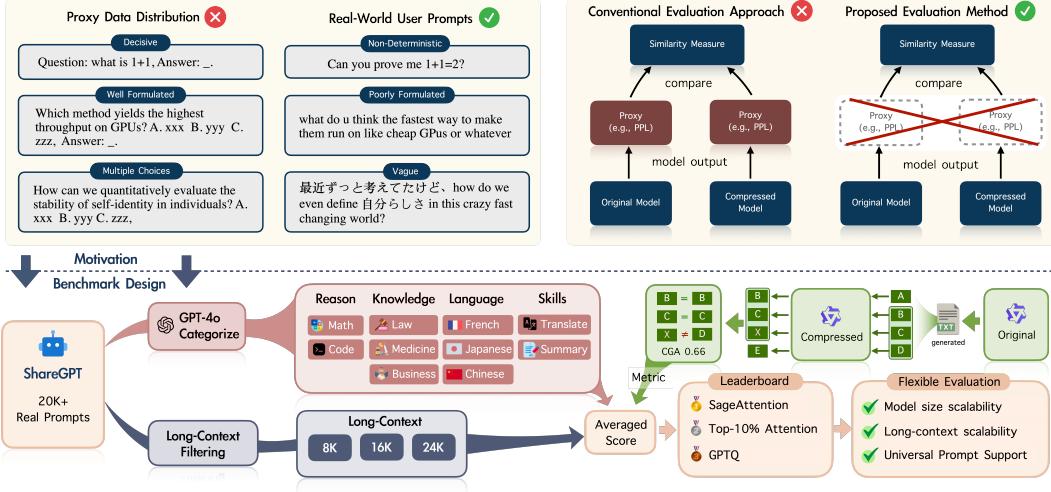


Figure 1: An overview of our evaluation framework versus conventional proxy-based methods. (Top) Traditional approaches rely on unrepresentative proxy data, like curated benchmarks (*left*), and indirect proxy metrics, like Perplexity (*right*), which fail to measure true generative faithfulness. (Bottom) Our framework eliminates these proxies by using real user queries and introducing Conditional Generation Accuracy (CGA) to directly measure a compressed model’s ability to replicate the original model’s next-token predictions.

34 **Limitations of Perplexity.** Perplexity is plagued by two key issues. First, its unbounded value range
 35 and model-specific baselines make fair cross-model comparison difficult [47]. More importantly,
 36 PPL is easily exploitable, certain compression strategies, such as KV cache dropping [27, 52], can be
 37 over optimized to perform well on PPL while producing outputs that diverge significantly from the
 38 original model, thereby undermining practical usability.

39 **Limitations of QA Benchmarks.** In response to the shortcomings of PPL, recent works [20, 16]
 40 have shifted toward QA downstream tasks. While these efforts attempt to expand the evaluation space
 41 by incorporating additional dimensions (e.g., fairness [26], privacy [25], reasoning [20]), they remain
 42 fundamentally constrained by rigid formats such as multiple choice or binary classification. These
 43 formats poorly reflect the nature of real world model interactions, which often involve generating
 44 long form, coherent, and contextually grounded text, especially in multi step reasoning tasks. More-
 45 over, reliance on answer accuracy introduces new biases and may incentivize overfitting to narrow
 46 evaluation schema.

47 In summary, the core problem lies in the reliance on *proxy metrics* (e.g., PPL, multiple choice
 48 accuracy) and *proxy data distributions* (e.g., curated benchmarks). This proxy based paradigm is
 49 the root cause of the mismatch between benchmark scores and real world utility and poses a serious
 50 barrier to meaningful progress in LLM compression.

51 To address this issue, we introduce a novel evaluation framework (Figure 1) that eliminates proxies in
 52 favor of direct, fidelity based evaluation using real world user prompts. Our framework is built on
 53 two core innovations:

54 **Conditional Generation Accuracy (CGA): A Fidelity Metric.** To directly measure fidelity, we
 55 propose CGA. This metric sidesteps the cascading errors of edit distance [1] and the external model
 56 bias of BERTScore [51] by employing a teacher-forcing paradigm. Specifically, we evaluate the
 57 compressed model’s next token prediction accuracy at each step, conditioned on the ground-truth
 58 sequence generated by the original model. This provides a pure and direct assessment of the
 59 compressed model’s ability to replicate the original’s generative behavior.

60 **Real User Queries as the Evaluation Distribution.** We replace synthetic benchmarks [20, 16] with
 61 authentic user queries sourced from platforms like ShareGPT. This dataset mirrors the open-ended
 62 and often ambiguous nature of real world interactions where no single “correct” answer exists. Our
 63 framework handles such data naturally by treating the original model’s output as the reference. For

64 granular analysis, we further leverage GPT-4o [19] to classify queries by domain (e.g., math, code)
65 and stratify them by context length.

66 We applied our framework to conduct a comprehensive evaluation of nine mainstream compres-
67 sion methods across four categories. Our experiments cover multiple scales, assessing Qwen2.5-
68 Instruct [48] models from 7B to 32B parameters and context lengths up to 24K tokens. This approach
69 yields directly comparable fidelity scores bounded within $[0, 1]$, leading to several key findings:

Key Takeaways

Comparative Performance

- Clear Hierarchy: A distinct performance hierarchy emerges among compression categories:
Low-Precision Attention > INT4 Quantization > 50% Pruning » KV Cache Dropping.
- Quantization: Within INT4 methods, GPTQ [11] consistently shows higher fidelity and
better generalization than AWQ [28].
- Sparse Attention: A sparse attention baseline (Top-10%) unexpectedly outperforms all
evaluated INT4 quantization methods.
- KV Cache Dropping: These methods [27, 52] perform very poorly. Their outputs are often
misaligned with the original model, raising serious questions about their practical viability.

Scaling with Model Size

- Favorable Scaling: The fidelity of quantization [11, 28] and most low-precision attention [38,
50] methods improves significantly on larger models.
- Consistent Scaling: Pruning methods [10, 40] and Top-10% sparse attention exhibit stable
performance across different model sizes.
- Negative Scaling: Uniquely, KV cache dropping methods [27, 52] show degraded perfor-
mance as model size increases.

Scaling with Long Contexts

- Degrading Fidelity: All INT4 quantization [11, 28] and KV cache dropping methods [27,
52] show a marked decline in fidelity as context length grows.
- Stable Fidelity: Pruning based methods [10, 40] and FlashAttention FP8 [38] maintain
consistent performance across all tested context lengths.
- Improving Fidelity: Top-10% sparse attention is the only tested method whose fidelity
robustly improves with longer contexts.

70

71 To promote transparency and accelerate progress in the field, we will open-source our evalua-
72 tion pipeline and release a public leaderboard upon paper acceptance. We hope this work provides a
73 rigorous, interpretable, and scalable foundation for fair benchmarking in LLM compression research.

74 2 Related Works

75 **LLM Compression.** The deployment of LLMs on resource-constrained hardware has catalyzed
76 extensive research into model compression [54, 35, 45]. Key lossy techniques include low-precision
77 attention [39, 14], quantization [23, 30], pruning [32, 41], and KV cache compression [29, 21, 17, 7],
78 with state-of-the-art methods in each category often claiming significant efficiency gains at minimal
79 performance cost.

80 *Low-Precision Attention.* The advent of specialized hardware like NVIDIA’s Hopper architecture has
81 made FP8 a viable option for accelerating attention mechanisms [3]. FlashAttention utilizes IO-aware
82 algorithms to speed up exact attention computation [6, 5, 38], and its FP8 implementation offers
83 substantial throughput gains [2]. However, this can introduce noise that degrades performance on
84 sensitive downstream tasks. In response, hybrid-precision methods like SageAttention [50] have been
85 proposed, which selectively use INT8 or INT4 for query and key (QK) projections while retaining
86 FP8 for value and output (VO) computations to preserve accuracy.

87 Another line of work focuses on sparsity. Methods like MoBA [31] and NSA [49] use heuristics to
88 approximate attention by focusing only on a subset of KV blocks. While practical, these are generally
89 considered to be outperformed by Top-K Attention [13], which, despite not offering a direct speedup,
90 serves as a valuable upper bound on the performance of sparse attention methods by using true scores
91 to select the most relevant KV blocks.

92 *Quantization.* Quantization has become a cornerstone of LLM compression [23, 30, 8]. GPTQ [11]
93 is a widely-used post-training quantization (PTQ) method that leverages approximate second-order
94 Hessian information to perform layer-wise weight quantization, achieving high accuracy at INT4
95 and INT3 bit-widths. While effective, the quantization process itself is computationally intensive. In
96 contrast, AWQ [28] proposes a faster method that first scales weights based on the magnitude of their
97 corresponding activations before quantization. This dependence on activation statistics, however, can
98 make AWQ more sensitive to the distribution of the calibration data [18], potentially impacting its
99 generalization capabilities.

100 *Pruning.* Pruning methods aim to improve efficiency by removing redundant model parameters [32,
101 41]. While simple magnitude-based pruning is a common baseline [24, 41], it risks removing weights
102 crucial for model performance. To mitigate this, more sophisticated methods have been developed.
103 SparseGPT [10] introduced a calibration process to achieve high levels of sparsity (e.g., 60Wanda [41])
104 refines this by removing weights with the smallest magnitudes after multiplying them by the norms
105 of their corresponding input activations, providing a more robust measure of a weight’s contribution.

106 *KV Cache Dropping.* To accelerate the decoding phase of generative inference, several methods focus
107 on compressing the KV cache [29, 21, 17, 7]. H2O [52] identifies and retains the most influential
108 “heavy-hitter” KV pairs based on their cumulative attention scores, evicting less important pairs to
109 stay within a fixed token budget. Building on this, SnapKV [27] further compresses the prompt’s KV
110 cache, aiming to reduce memory and computational overhead even more.

111 **Compressed LLMs Evaluation.** The evaluation of compressed LLMs has evolved rapidly. Early
112 work predominantly relied on PPL [47], but this metric has been widely criticized for its susceptibility
113 to being “gamed” and its poor correlation with downstream task performance [9].

114 This led to a shift towards using comprehensive QA benchmarks [46, 43]. Researchers have proposed
115 increasingly complex evaluation suites that measure performance across multiple dimensions, such as
116 fairness, privacy, and ethics, as seen in works like [16] and [46]. Others, such as LLM-Kick [20], have
117 focused on aggregating numerous existing QA tasks into a single, extensive leaderboard. However,
118 these benchmarks are typically limited to structured formats like multiple-choice questions [34]. This
119 format diverges significantly from the open-ended, generative nature of real-world user interactions,
120 introducing a new form of evaluation bias [22] and creating the risk that compression methods will
121 be over-optimized for these specific test formats.

122 Sharing our motivation to move beyond proxy metrics like PPL and QA benchmarks, a concurrent
123 work [43] directly compares the outputs of compressed and original models. This approach has
124 explored metrics such as edit distance [1] and BERTScore [51], but both have inherent drawbacks.
125 Edit distance is unreliable for longer generations, as minor initial errors can cascade into large,
126 misleading divergence scores. BERTScore, while designed to capture semantic similarity, introduces
127 its own set of potential biases by relying on an external pretrained model [42].

128 3 A Direct, Fidelity-Based Evaluation Framework

129 The central goal of evaluating a compressed LLM is to quantify its faithfulness to the original. Let \mathcal{F}
130 denote an original, uncompressed LLM and $\tilde{\mathcal{F}}$ represent its compressed counterpart. For any given
131 user prompt X , these models produce output sequences $Y = \mathcal{F}(X)$ and $\tilde{Y} = \tilde{\mathcal{F}}(X)$, respectively.
132 An ideal evaluation framework would assess the quality of $\tilde{\mathcal{F}}$ by directly comparing \tilde{Y} to Y across a
133 distribution of prompts X that accurately reflects real-world usage.

134 However, existing evaluation paradigms [46, 16, 20] are fundamentally misaligned with this goal,
135 suffering from critical flaws on two fronts: the data they use for evaluation and the metrics they
136 employ for comparison.

137 **Problem I: Proxy Data Distributions.** Current benchmarks [46, 16, 20] overwhelmingly rely on
 138 proxy data distributions that fail to capture the diversity and complexity of genuine user interactions.
 139 QA benchmarks, for instance, utilize highly structured prompts (X_{QA}) that constrain the model to
 140 selecting from a small, predefined set of answers (e.g., $\{A, B, C\}$). This rigid format is incapable
 141 of assessing a model’s ability to handle the ambiguous, open-ended, and multi-turn conversational
 142 prompts that are commonplace in practice. Similarly, PPL benchmarks [33, 37] often use large text
 143 corpora. While vast, these datasets are stylistically monolithic and unrepresentative of the interactive
 144 nature of conversational queries. This fundamental mismatch between benchmark data and real-world
 145 data introduces a persistent and misleading evaluation bias.

146 **Problem II: Proxy Comparison Metrics.** Instead of directly comparing the generated output
 147 sequences Y and \tilde{Y} , prevailing methods [46, 16, 20, 43] resort to proxy metrics. The final score S in
 148 these frameworks is not a direct measure of generative fidelity but an indirect comparison of abstract
 149 values derived from the model outputs. For PPL, the score compares perplexity values rather than the
 150 generated text itself:

$$S_{PPL} \propto \sum_{X_{PPL}} \text{sim} \left(\text{PPL}(\mathcal{F}(X_{PPL})), \text{PPL}(\tilde{\mathcal{F}}(X_{PPL})) \right) \quad (1)$$

151 Likewise, QA benchmarks [46, 16, 20] compare final task scores, abstracting away the generative
 152 process entirely and focusing only on the outcome:

$$S_{QA} \propto \sum_{X_{QA}} \mathbf{1} \left(\text{score}(\mathcal{F}(X_{QA})) = \text{score}(\tilde{\mathcal{F}}(X_{QA})) \right) \quad (2)$$

153 The reliance on these proxies fundamentally obscures the evaluation. A compressed model can
 154 achieve a PPL or QA score nearly identical to the original’s while producing text that is factually
 155 incorrect, stylistically divergent, or simply incoherent. This indirection is the primary source of the
 156 well-documented gap between benchmark results and real-world user experience.

157 To remedy these issues, we propose a new evaluation framework built on two core principles: using
 158 an authentic data distribution and employing a direct fidelity metric.

159 3.1 Solution I: Aligning with Real-World Data Distributions

160 To ensure our evaluation accurately reflects practical applications, we constructed our evaluation set
 161 from real-world user interactions. We curated a dataset of tens of thousands of conversations from
 162 ShareGPT, a public platform containing diverse, multilingual, and often complex user prompts. To
 163 enable a granular and multi-faceted analysis, we further processed this raw data in two ways:

164 **Domain-Specific Subsets.** We employed GPT-4o to automatically classify and filter the prompts into
 165 ten distinct, high-quality categories. These include five knowledge-intensive domains (mathematics,
 166 programming, business, law, and medicine), three non-English languages (Japanese, French, and Chi-
 167 nese), and two specific skills (summarization and translation). Each category contains approximately
 168 100 challenging prompts, allowing for targeted assessment of model performance in specialized areas.

169 **Long-Context Strata.** To rigorously analyze model performance in long-context scenarios, a critical
 170 application for compressed models, we segmented a portion of the data into three length-based tiers:
 171 8K, 16K, and 24K tokens. Each tier consists of approximately 20 prompts, creating a dedicated
 172 testbed that mirrors the memory and computational demands of real-world, long-context tasks.

173 This dual-strategy data curation ensures our evaluation is not only aligned with the distribution of
 174 genuine user queries but is also structured to yield precise insights into how compression affects
 175 model performance across various domains and context scales.

176 3.2 Solution II: Conditional Generation Accuracy, A Direct Fidelity Metric

177 The ultimate measure of a compressed model’s quality is its ability to faithfully replicate the generative
 178 behavior of the original model. Existing metrics fail to capture the nuanced aspects of generation,
 179 including tone, style, and the step-by-step reasoning process embedded in an output sequence. To
 180 overcome this, we introduce *Conditional Generation Accuracy* (CGA), a metric designed to directly
 181 compare the output distributions of the compressed and original models at the token level.

182 The standard auto-regressive generation process for the original model \mathcal{F} and a compressed model $\tilde{\mathcal{F}}$
 183 at a given step i is defined as:

$$y_i = \mathcal{F}(X, y_1, \dots, y_{i-1}) \quad \text{and} \quad \tilde{y}_i = \tilde{\mathcal{F}}(X, \tilde{y}_1, \dots, \tilde{y}_{i-1}), \quad (3)$$

184 A naive comparison between the final sequences Y and \tilde{Y} (e.g., using edit distance) is unreliable. A
 185 single-token divergence early in the generation process can cause the subsequent outputs to diverge
 186 completely, leading to a cascade of errors that incorrectly penalizes the compressed model for what
 187 may be a minor initial deviation. This error accumulation problem renders simple sequence-level
 188 similarity metrics ineffective for evaluating long-form generation.

189 To isolate the predictive capability of the compressed model at each step, our CGA metric employs a
 190 *teacher-forcing* paradigm. Instead of conditioning the compressed model’s next-token prediction on
 191 its own, potentially flawed, previously generated tokens ($\tilde{y}_{<i}$), we condition it on the ground-truth
 192 prefix generated by the original model ($y_{<i}$):

$$\hat{y}_i = \tilde{\mathcal{F}}(X, y_1, y_2, \dots, y_{i-1}). \quad (4)$$

193 Here, \hat{y}_i represents the compressed model’s most likely next token given the ideal context from the
 194 original model. This approach decouples the evaluation at each step from errors made in prior steps,
 195 eliminating the issue of cascading failures. It provides a pure and direct measure of the compressed
 196 model’s ability to replicate the original’s output at every point in the generation process.

197 The final CGA score is the mean accuracy over all tokens in the generated sequence, averaged across
 198 all prompts in the evaluation dataset D :

$$S_{\text{CGA}} = \frac{1}{|D|} \sum_{X \in D} \left(\frac{1}{|Y|} \sum_{i=1}^{|Y|} \mathbf{1}(\hat{y}_i = y_i) \right) = \frac{1}{|D|} \sum_{X \in D} \left(\frac{1}{|Y|} \sum_{i=1}^{|Y|} \mathbf{1}(\tilde{\mathcal{F}}(X, y_{<i}) = \mathcal{F}(X, y_{<i})) \right). \quad (5)$$

199 As the formulation shows, CGA directly measures the alignment between the compressed and original
 200 models’ next-token predictions, conditioned on the golden output history. It uses no external models
 201 or abstract proxies, providing the most direct and fundamental assessment of generative faithfulness.

202 4 Experiments

203 Our experimental evaluation is structured to first validate our proposed framework and then apply it
 204 to systematically analyze the landscape of LLM compression. We proceed in four stages. First, we
 205 validate that our Conditional Generation Accuracy (CGA) metric aligns more closely with human-
 206 aligned judgments than standard proxy metrics (Section 4.1). Second, armed with this validated
 207 metric, we conduct a comprehensive, head-to-head comparison of all selected compression methods
 208 to establish a clear performance hierarchy (Section 4.2). Finally, we investigate how the fidelity of
 209 each method scales with two critical factors: model size, from 7B to 32B parameters (Section 4.3),
 210 and context length, from 8K to 24K tokens (Section 4.4).

Table 1: Validation of evaluation metrics on Qwen2.5-7B-Instruct [48]. While proxy metrics like PPL
 and MMLU [15] often fail to reflect true performance degradation (misleading scores are underlined),
 our proposed CGA score aligns closely with the human-aligned GPT Score, providing a more faithful
 measure of generative quality.

Metric	INT4 Quant		Low-Precision Attn			KV Cache Drop		Pruning		Baseline
	AWQ [28]	GPTQ [11]	Flash FP8 [38]	Sage [50]	Top-10%	SnapKV [27]	H2O [52]	Sparse [10]	Wanda [40]	
WT2 PPL [33]	17.76	16.79	116.27	16.00	15.99	15.57	18.56	21.37	20.89	15.99
MMLU [15]	0.718	0.718	0.570	0.718	0.718	0.401	0.680	0.651	0.658	0.719
GPT Score	0.833	0.849	0.042	0.882	0.865	0.031	0.000	0.289	0.303	1.0
WT2 PPL (norm)	0.172	0.454	0.000	0.997	1.000	1.000	0.077	0.005	0.008	1.0
MMLU (norm)	0.999	0.999	0.793	0.999	0.999	0.558	0.946	0.907	0.915	1.0
CGA (ours)	0.916	0.922	0.586	0.987	0.953	0.538	0.465	0.784	0.737	1.0

211 4.1 Validating the CGA Metric

212 The foundational step of our analysis is to validate that CGA serves as a more reliable indicator
 213 of generative faithfulness than conventional proxy metrics. To do this, we benchmarked CGA

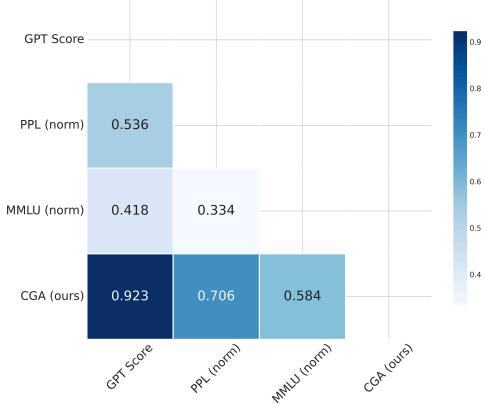


Figure 2: Correlation matrix of evaluation metrics. Darker cells indicate stronger positive correlation. Our proposed Conditional Generation Accuracy (CGA) demonstrates the highest correlation (0.923) with the human preference benchmark (GPT Score), significantly outperforming proxy metrics like PPL and MMLU.

214 against Perplexity, MMLU accuracy, and GPT-4o [19] judgments, using the latter as a high-quality
 215 approximation for human preference. All experiments in this section were conducted on the Qwen2.5-
 216 7B-Instruct model [48].

217 **Evaluation Setup.** We compared three standard evaluation approaches against our proposed metric.

- 218 • **PPL.** We measured the average Perplexity on the WikiText-2 dataset [33]. To ensure fair comparison,
 219 raw PPL scores were normalized to a [0, 1] range by applying a sigmoid function to the PPL
 220 difference between the compressed and baseline models.
- 221 • **QA Score.** We used the MMLU benchmark [15] in a zero-shot setting, evaluated via lm-eval-
 222 harness (v0.4.9) [12]. The final reported score is the ratio of the compressed model’s accuracy to
 223 the baseline’s accuracy.
- 224 • **GPT Score.** As a approximation for human evaluation, we prompted GPT-4o [19] to perform
 225 a pairwise comparison between the outputs of the original and compressed models for a given
 226 prompt, randomizing their presentation order to prevent bias. The final score represents the win
 227 rate of the compressed model divided by that of the original model.

228 **Results.** Table 1 starkly illustrates the unreliability of proxy metrics. PPL, in particular, shows
 229 a weak correlation with the human-aligned GPT Score. It misleadingly assigns a perfect score
 230 to SnapKV [27], a method that causes a near-total collapse in output quality, while penalizing
 231 performant methods like FlashAttention FP8 [38]. The MMLU score, while correctly identifying the
 232 worst-performing methods, quickly saturates close to 1.0 for most techniques, failing to differentiate
 233 between moderately and highly faithful models like AWQ [28] and SageAttention [50].

234 A Pearson correlation analysis, visualized in Figure 2, provides quantitative support for these ob-
 235 servations. CGA exhibits a correlation coefficient of 0.923 with the GPT Score, demonstrating a
 236 significantly stronger alignment with human-aligned judgments than either PPL or MMLU. While
 237 the GPT Score itself is a valuable tool for validation, its inherent variance, prompt sensitivity, and
 238 high inference cost make it impractical for scalable, reproducible benchmarking. CGA, in contrast,
 239 offers a robust, deterministic, and highly correlated alternative, making it an ideal primary metric for
 240 our evaluation framework.

241 4.2 Head-to-Head Comparison of Compression Methods

242 Having established the validity of CGA, we conducted a large-scale benchmark of nine compression
 243 methods on Qwen2.5-7B-Instruct [48] to establish a clear, initial performance ranking. This direct
 244 comparison, detailed across various domains and skills in Table 2, provides a fine-grained overview
 245 of each method’s strengths and weaknesses. Corresponding results for the 14B and 32B models are
 246 available in Appendix [xx].

Table 2: Performance comparison of 9 compression methods on Qwen2.5-7B-Instruct [48], measured by Conditional Generation Accuracy (CGA). The top three scores in each row are highlighted: 1st, 2nd, and 3rd. SageAttention [50] consistently achieves the highest fidelity, while a Top-10% sparse attention baseline unexpectedly outperforms all INT4 quantization methods [11, 28].

Category	Sub-dataset	Compression Method								
		AWQ	GPTQ	Flash FP8	Sage	Top-10%	SnapKV	H2O	Sparse	Wanda
Reasoning	code	0.9228	0.9504	0.4576	0.9844	0.9472	0.7466	0.6323	0.8150	0.8165
	math	0.9567	0.9426	0.5457	0.9829	0.9671	0.6350	0.6193	0.6455	0.8752
Knowledge	fact	0.8773	0.8789	0.6078	0.9874	0.9010	0.6403	0.5217	0.7855	0.7486
	law	0.9023	0.9055	0.1899	0.9850	0.9205	0.6593	0.5782	0.7635	0.7755
	business	0.8994	0.9102	0.6753	0.9861	0.9401	0.6456	0.5929	0.7511	0.7700
	medicine	0.9169	0.9299	0.4307	0.9795	0.9619	0.7007	0.6070	0.7929	0.8063
Language	fr	0.9165	0.9319	0.3199	0.9874	0.9425	0.7171	0.5794	0.7706	0.7811
	ch	0.8986	0.9094	0.6236	0.9866	0.9291	0.6876	0.4982	0.7289	0.7496
	jp	0.8914	0.8935	0.7486	0.9874	0.9254	0.6608	0.5301	0.8547	0.8827
Skills	en2zh	0.9497	0.9341	0.7273	0.9874	0.9801	0.4152	0.4812	0.8086	0.8313
	zh2en	0.9383	0.9608	0.7158	0.9809	0.9824	0.4800	0.4647	0.8415	0.8939
	sum	0.8491	0.9176	0.8710	0.9837	0.9243	0.1948	0.7156	0.8026	0.7999
Average		0.9091	0.9212	0.5796	0.9860	0.9428	0.5997	0.5709	0.7797	0.8200

247 The results reveal several key insights into the current state of LLM compression:

- 248 • *A Clear Performance Hierarchy.* A distinct performance hierarchy emerges among the categories.
249 Low-precision attention methods like SageAttention [50] are superior, followed by INT4 quanti-
250 zation [11, 28] and then pruning [40, 10]. KV cache dropping methods [27, 52] dramatically
251 underperform, with CGA scores at the default 20% budget falling so low as to call their practical
252 utility into question.
- 253 • *Low-Precision Attention Trade-offs.* The comparison between FlashAttention FP8 [38] and SageAt-
254 tention [50] confirms that naively replacing all BF16 attention computations with FP8 leads to a
255 severe performance drop. SageAttention’s hybrid-precision strategy successfully mitigates this
256 degradation, achieving the highest fidelity among all tested methods and validating its design
257 principles.
- 258 • *The Surprising Efficacy of Sparsity.* Top-10% Attention, a simple sparse attention baseline, delivered
259 unexpectedly strong performance. Despite its high effective sparsity (90%) on typical sequences, it
260 ranked second overall, outperforming all evaluated INT4 quantization and pruning methods. This
261 highlights the significant, perhaps underestimated, potential of structured sparsity [49, 31] as a
262 high-fidelity compression strategy.

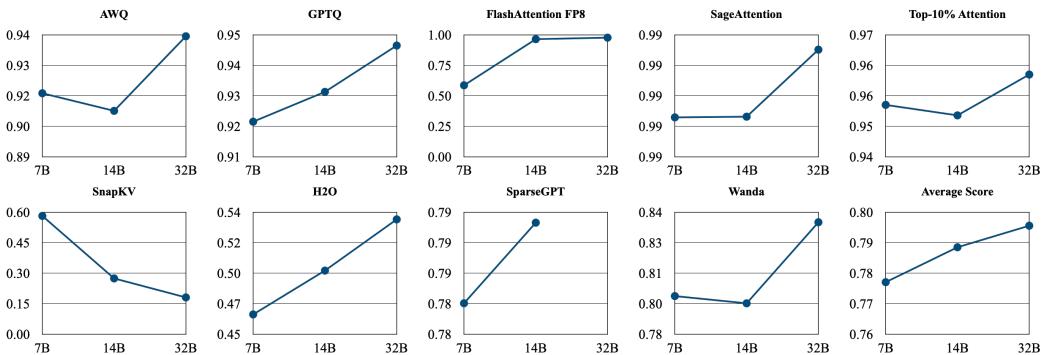


Figure 3: Scaling of compression method fidelity with model size (7B, 14B, 32B). Most methods improve on larger models, indicating that greater parameter redundancy enhances compressibility. SnapKV [27] is a notable exception, exhibiting negative scaling where fidelity degrades as model size increases.

263 **4.3 Scaling with Model Size**

264 A central hypothesis in model compression is that larger models, with their inherent parameter
 265 redundancy, should be more resilient to information loss. To test this, we evaluated how the fidelity
 266 of each compression method scales with model size, applying them to the 7B, 14B, and 32B versions
 267 of the Qwen2.5-Instruct model [48].

268 The results, plotted in Figure 3, largely confirm this hypothesis. Most methods, including quantiza-
 269 tion [11, 28] and low-precision attention [38, 50], become more faithful as model size increases. This
 270 trend empirically supports the idea that the impact of a given compression ratio lessens as model
 271 capacity grows. While AWQ [28], Top-10% Attention, and Wanda [40] show a minor, anomalous
 272 dip at the 14B scale, the overall trend remains positive. SnapKV [27] is the only method to exhibit
 273 negative scaling, with its fidelity degrading as the model grows larger.

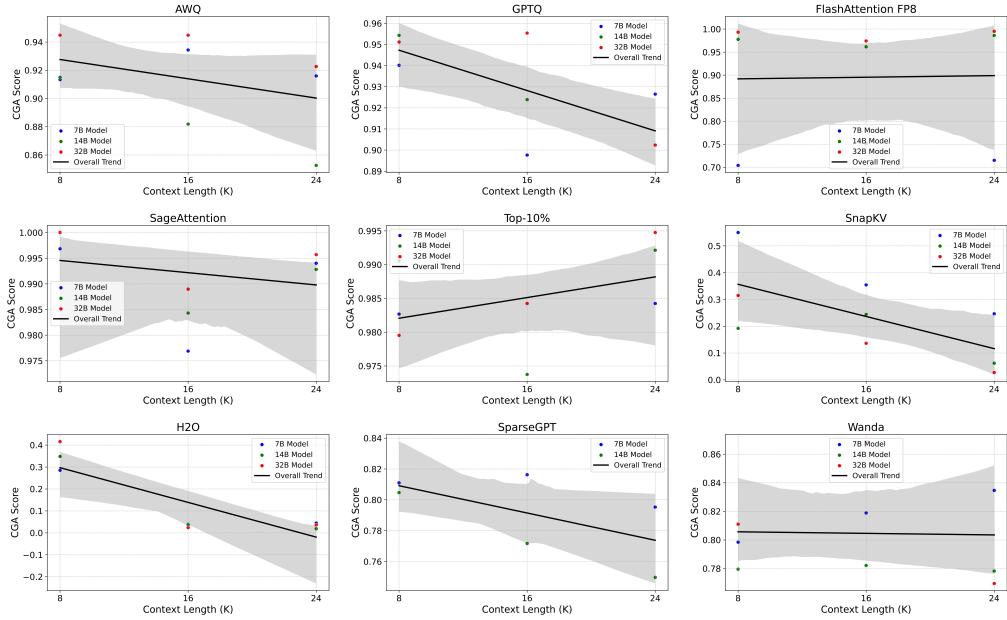


Figure 4: Scaling of compression method fidelity with increasing context length. Methods exhibit three distinct scaling laws: *degradation* (quantization [11, 28], KV cache dropping [27, 52]), *stability* (Wanda [40], FlashAttention FP8 [38]), and *improvement* (Top-10% Attention), a critical consideration for long-context applications.

274 **4.4 Scaling with Context Length**

275 Many compression techniques, particularly low-precision attention [38, 50] and KV cache drop-
 276 ping [27, 52], are motivated by the need to manage the computational and memory costs of long-
 277 context inference. Our framework is naturally suited to evaluating performance under these conditions.
 278 We stratified our ShareGPT test data into three tiers based on prompt token count (8K, 16K, and 24K)
 279 to assess how the fidelity of each method scales as the context window expands.

280 As shown in Figure 4, this analysis reveals three distinct and informative scaling laws:

- 281 • *Degrading Fidelity*. All quantization methods [11, 28], all KV cache dropping methods [52, 27],
 282 SparseGPT [10], and SageAttention [50] show a clear decline in fidelity as context expands.
- 283 • *Stable Fidelity*. FlashAttention FP8 [38] and the pruning method Wanda [40] maintain remarkably
 284 consistent performance across all tested context lengths, indicating a robustness to longer sequences.
- 285 • *Improving Fidelity*. Top-10% Attention is the only tested method whose fidelity robustly improves
 286 with longer contexts. This counter-intuitive result suggests that as more context becomes available,
 287 the model’s ability to identify the most salient tokens for sparse attention becomes more effective.

288 These divergent scaling laws underscore that even methods within the same category can possess
289 fundamentally different properties, a critical factor for practitioners selecting a compression strategy
290 for long-context applications.

291 **5 Conclusion and Limitations**

292 We have proposed and validated a new evaluation framework for LLM compression that uses real-
293 world user queries and a direct fidelity metric, Conditional Generation Accuracy (CGA), to measure
294 performance. Our experiments show that CGA aligns significantly better with human-aligned
295 judgments than proxy metrics like PPL and QA benchmark accuracy. Using this framework, we
296 conducted a comprehensive analysis of mainstream compression methods, revealing performance
297 trade-offs across model sizes and context lengths that provide a more realistic assessment.

298 Our approach has two primary limitations. First, its reliance on a token-by-token evaluation makes it
299 more computationally intensive than traditional benchmarks. Second, by averaging accuracy across
300 the entire sequence, CGA treats all tokens as equally important, which may not fully capture the
301 nuanced impact of compression on output quality. Investigating methods to weigh tokens by their
302 semantic or structural importance is a promising area for future work.

303 **References**

- 304 [1] Z. Bar-Yossef, T. Jayram, R. Krauthgamer, and R. Kumar. Approximating edit distance
305 efficiently. In *FOCS*, 2004.
- 306 [2] S. Chen, Z. Liu, Z. Wu, C. Zheng, P. Cong, Z. Jiang, et al. Int-flashattention: Enabling flash
307 attention for int8 quantization. *arXiv preprint arXiv:2409.16997*, 2024.
- 308 [3] J. Choquette. Nvidia hopper gpu: Scaling performance. In *HCS*, 2022.
- 309 [4] C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova. Boolq:
310 Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*,
311 2019.
- 312 [5] T. Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv
313 preprint arXiv:2307.08691*, 2023.
- 314 [6] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré. Flashattention: Fast and memory-efficient exact
315 attention with io-awareness. In *NeurIPS*, 2022.
- 316 [7] H. Dong, X. Yang, Z. Zhang, Z. Wang, Y. Chi, and B. Chen. Get more with less: Syn-
317 thesizing recurrence with kv cache compression for efficient llm inference. *arXiv preprint
318 arXiv:2402.09398*, 2024.
- 319 [8] V. Egiazarian, A. Panferov, D. Kuznedelev, E. Frantar, A. Babenko, and D. Alistarh. Ex-
320 treme compression of large language models via additive quantization. *arXiv preprint
321 arXiv:2401.06118*, 2024.
- 322 [9] L. Fang, Y. Wang, Z. Liu, C. Zhang, S. Jegelka, J. Gao, B. Ding, and Y. Wang. What is wrong
323 with perplexity for long-context language modeling? *arXiv preprint arXiv:2410.23771*, 2024.
- 324 [10] E. Frantar and D. Alistarh. Sparsegpt: Massive language models can be accurately pruned in
325 one-shot. In *ICML*, 2023.
- 326 [11] E. Frantar, S. Ashkboos, T. Hoefer, and D. Alistarh. Gptq: Accurate post-training quantization
327 for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.
- 328 [12] L. Gao, J. Tow, B. Abbasi, S. Biderman, S. Black, A. DiPofi, et al. The language model
329 evaluation harness. Zenodo, 2024.
- 330 [13] A. Gupta, G. Dar, S. Goodman, D. Ciprut, and J. Berant. Memory-efficient transformers via
331 top- k attention. *arXiv preprint arXiv:2106.06899*, 2021.
- 332 [14] D. Han, X. Pan, Y. Han, S. Song, and G. Huang. Flatten transformer: Vision transformer using
333 focused linear attention. In *CVPR*, 2023.
- 334 [15] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, et al. Measuring massive
335 multitask language understanding. In *ICLR*, 2021.
- 336 [16] J. Hong, J. Duan, C. Zhang, Z. Li, C. Xie, K. Lieberman, et al. Decoding compressed trust:
337 scrutinizing the trustworthiness of efficient llms under compression. In *ICML*, 2024.
- 338 [17] C. Hooper, S. Kim, H. Mohammadzadeh, M. W. Mahoney, Y. S. Shao, K. Keutzer, et al.
339 Kvquant: Towards 10 million context length llm inference with kv cache quantization. In
340 *NeurIPS*, 2024.
- 341 [18] I. Hubara, Y. Nahshan, Y. Hanani, R. Banner, and D. Soudry. Accurate post training quantization
342 with small calibration sets. In *ICML*, 2021.
- 343 [19] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, et al. Gpt-4o system card.
344 *arXiv preprint arXiv:2410.21276*, 2024.
- 345 [20] A. Jaiswal, Z. Gan, X. Du, B. Zhang, Z. Wang, and Y. Yang. Compressing llms: The truth is
346 rarely pure and never simple. In *ICLR*, 2024.

- 347 [21] H. Kang, Q. Zhang, S. Kundu, G. Jeong, Z. Liu, T. Krishna, et al. Gear: An efficient
348 kv cache compression recipe for near-lossless generative inference of llm. *arXiv preprint*
349 *arXiv:2403.05527*, 2024.
- 350 [22] C. V. Kumar, A. Urlana, G. Kanumolu, B. M. Garlapati, and P. Mishra. No llm is free from
351 bias: A comprehensive study of bias evaluation in large language models. *arXiv preprint*
352 *arXiv:2503.11985*, 2025.
- 353 [23] J. Lang, Z. Guo, and S. Huang. A comprehensive study on quantization techniques for large
354 language models. In *ICAIRC*, 2024.
- 355 [24] J. Lee, S. Park, S. Mo, S. Ahn, and J. Shin. Layer-adaptive sparsity for the magnitude-based
356 pruning. *arXiv preprint arXiv:2010.07611*, 2020.
- 357 [25] B. Li, Y. Jiang, V. Gadepally, and D. Tiwari. Llm inference serving: Survey of recent advances
358 and opportunities. In *HPEC*, 2024.
- 359 [26] Y. Li, M. Du, R. Song, X. Wang, and Y. Wang. A survey on fairness in large language models.
360 *arXiv preprint arXiv:2308.10149*, 2023.
- 361 [27] Y. Li, Y. Huang, B. Yang, B. Venkitesh, A. Locatelli, H. Ye, T. Cai, P. Lewis, and D. Chen.
362 Snapkv: Llm knows what you are looking for before generation. In *NeurIPS*, 2024.
- 363 [28] J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, et al. Awq: Activation-aware
364 weight quantization for on-device llm compression and acceleration. In *MLSys*, 2024.
- 365 [29] Z. Liu, A. Desai, F. Liao, W. Wang, V. Xie, Z. Xu, et al. Scissorhands: Exploiting the persistence
366 of importance hypothesis for llm kv cache compression at test time. In *NeurIPS*, 2023.
- 367 [30] Z. Liu, C. Zhao, I. Fedorov, B. Soran, D. Choudhary, R. Krishnamoorthi, et al. Spinquant: Llm
368 quantization with learned rotations. *arXiv preprint arXiv:2405.16406*, 2024.
- 369 [31] E. Lu, Z. Jiang, J. Liu, Y. Du, T. Jiang, C. Hong, S. Liu, W. He, E. Yuan, Y. Wang, et al. Moba:
370 Mixture of block attention for long-context llms. *arXiv preprint arXiv:2502.13189*, 2025.
- 371 [32] X. Ma, G. Fang, and X. Wang. Llm-pruner: On the structural pruning of large language models.
372 In *NeurIPS*, 2023.
- 373 [33] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. In *ICLR*,
374 2017.
- 375 [34] A. Myrzakhan, S. M. Bsharat, and Z. Shen. Open-llm-leaderboard: From multi-choice to open-
376 style questions for llms evaluation, benchmark, and arena. *arXiv preprint arXiv:2406.07545*,
377 2024.
- 378 [35] S. Park, J. Choi, S. Lee, and U. Kang. A comprehensive survey of compression algorithms for
379 language models. *arXiv preprint arXiv:2401.15347*, 2024.
- 380 [36] G. Qu, Q. Chen, W. Wei, Z. Lin, X. Chen, and K. Huang. Mobile edge intelligence for large
381 language models: A contemporary survey. *ICST*, 2025.
- 382 [37] J. W. Rae, A. Potapenko, S. M. Jayakumar, and T. P. Lillicrap. Compressive transformers for
383 long-range sequence modelling. *arXiv preprint arXiv:1911.05507*, 2019.
- 384 [38] J. Shah, G. Bikshandi, Y. Zhang, V. Thakkar, P. Ramani, and T. Dao. Flashattention-3: Fast and
385 accurate attention with asynchrony and low-precision. In *NeurIPS*, 2024.
- 386 [39] Z. Shen, M. Zhang, H. Zhao, S. Yi, and H. Li. Efficient attention: Attention with linear
387 complexities. In *CVPR*, 2021.
- 388 [40] M. Sun, Z. Liu, A. Bair, and J. Z. Kolter. A simple and effective pruning approach for large
389 language models. *arXiv preprint arXiv:2306.11695*, 2023.
- 390 [41] M. Sun, Z. Liu, A. Bair, and J. Z. Kolter. A simple and effective pruning approach for large
391 language models. *arXiv preprint arXiv:2306.11695*, 2023.

- 392 [42] T. Sun, J. He, X. Qiu, and X. Huang. Bertscore is unfair: On social bias in language model-based
 393 metrics for text generation. *arXiv preprint arXiv:2210.07626*, 2022.
- 394 [43] Q. Wang, M. Wang, N. Feldhus, S. Ostermann, Y. Cao, H. Schütze, et al. Through a compressed
 395 lens: Investigating the impact of quantization on llm explainability and interpretability. *arXiv
 396 preprint arXiv:2505.13963*, 2025.
- 397 [44] Y. Wang, X. Ma, G. Zhang, Y. Ni, A. Chandra, S. Guo, et al. Mmlu-pro: A more robust and
 398 challenging multi-task language understanding benchmark. In *NeurIPS*, 2024.
- 399 [45] M. Xu, W. Yin, D. Cai, R. Yi, D. Xu, Q. Wang, et al. A survey of resource-efficient llm and
 400 multimodal foundation models. *arXiv preprint arXiv:2401.08092*, 2024.
- 401 [46] Z. Xu, A. Gupta, T. Li, O. Bentham, and V. Srikumar. Beyond perplexity: Multi-dimensional
 402 safety evaluation of llm compression. In *EMNLP*, 2024.
- 403 [47] Z. Xu, A. Gupta, T. Li, O. Bentham, and V. Srikumar. Beyond perplexity: Multi-dimensional
 404 safety evaluation of llm compression. *arXiv preprint arXiv:2407.04965*, 2024.
- 405 [48] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, et al. Qwen2.5 technical report. *arXiv
 406 preprint arXiv:2412.15115*, 2025.
- 407 [49] J. Yuan, H. Gao, D. Dai, J. Luo, L. Zhao, Z. Zhang, et al. Native sparse attention: Hardware-
 408 aligned and natively trainable sparse attention. *arXiv preprint arXiv:2502.11089*, 2025.
- 409 [50] J. Zhang, J. Wei, H. Huang, P. Zhang, J. Zhu, and J. Chen. Sageattention: Accurate 8-bit
 410 attention for plug-and-play inference acceleration. *arXiv preprint arXiv:2410.02367*, 2024.
- 411 [51] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text
 412 generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.
- 413 [52] Z. Zhang, Y. Sheng, T. Zhou, T. Chen, L. Zheng, R. Cai, et al. H2o: Heavy-hitter oracle for
 414 efficient generative inference of large language models. In *NeurIPS*, 2023.
- 415 [53] Y. Zheng, Y. Chen, B. Qian, X. Shi, Y. Shu, and J. Chen. A review on edge large language
 416 models: Design, execution, and applications. *CSUR*, 2025.
- 417 [54] X. Zhu, J. Li, Y. Liu, C. Ma, and W. Wang. A survey on model compression for large language
 418 models. 2024.

419 A Case Study

420 A.1 Qualitative Examples of Performance Loss

421 We provide qualitative examples to illustrate the typical failure modes of several compression methods,
 422 revealing performance losses that quantitative metrics often miss.

423 **AWQ** As shown in Figure 5, models compressed with AWQ can exhibit distracting artifacts such as
 424 textual repetition, unhelpful (or "helpless") responses, and generation loops. These issues lead to a
 425 perceptible decline in instruction-following ability and overall output quality.

426 **SageAttention** While generally robust and marketed as nearly lossless, SageAttention can produce
 427 entirely meaningless output in specific edge cases (Figure 6). This behavior appears to be a rare
 428 artifact rather than a systemic flaw, as the model performs correctly on the vast majority of prompts.

429 **SnapKV** The performance of SnapKV (Figure 7) starkly contrasts with its published claims. The
 430 compressed model consistently fails to generate logically coherent content, rendering it practically
 431 unusable for most tasks.

432 **Wanda** Wanda (Figure 8) severely degrades model performance, producing outputs that are often
 433 incoherent. Although its logical consistency is marginally better than SnapKV's, the loss in quality
 434 remains substantial.

User Prompt: I want to help my 10-year old son learn math better. I've started tutoring him recently, but we always end up fighting about it. Are there ways I can make math learning fun for his age? Basically teaching him and making him work hard on addition / subtraction / multiplication / division / fractions / decimals in a fun way? Give me some ideas please.

Qwen2.5-7B-Instruct

It's great that you're looking for ways to make math learning fun for your son! At 10 years old, he's at a stage where he can grasp more complex concepts, and making the learning process enjoyable can significantly enhance his engagement and understanding. Here are some fun and interactive ideas to help him learn addition, subtraction, multiplication, division, fractions, and decimals:

- ### 1. **Math Games**
- **Math Bingo**: Create bingo cards with answers to math problems. Call out problems instead of numbers, and he marks the answers on his card.
- **Math Jeopardy**: Create a game board with categories like Addition, Subtraction, Multiplication, Division, Fractions, and Decimals. Each category has questions of varying difficulty.
- **Math Card Games**: Use a deck of cards with a math twist. For example, the

2. More generic, often listing broad categories without concrete implementation tips

.....

- ### 9. **Math Challenges**
- **Daily Challenges**: Set up a daily or weekly challenge where he has to solve a certain number of problems.
- **Math Competitions**: Participate in online math competitions or join a local math club.

- ### 10. **Positive Reinforcement**
- **Praise and Rewards**: Praise his efforts and reward him for completing tasks or solving problems correctly.
- **Set Goals**: Set small, achievable goals and celebrate when he reaches them.

By incorporating these activities, you can make math learning a more enjoyable and engaging experience for your son. Remember to keep the atmosphere positive and supportive, and be patient as he learns and grows in his math skills.

Qwen2.5-7B-Instruct AWQ Int4

Absolutely! Making math fun for a 10-year-old can be a great way to engage them and help them develop a positive attitude towards the subject. Here are some ideas to make learning math more enjoyable:

- ### 1. **Math Games**
- **Card Games**: Use a deck of cards to play games like "War" but with a twist. For example, each player draws two cards, adds them up, and the player with the higher sum wins the round.
- **Board Games**: Games like "Sum Swamp" or "Math Blaster" can be fun and educational. These games often involve basic arithmetic and strategic thinking.
- **Online Games**: Websites like Khan Academy, IXL, and Coolmath Games offer a variety of games that can help reinforce math skills in a fun way.

1. Some sections are repetitive

- ### 11. **Math in Technology**
- **Coding**: Introduce basic coding concepts using platforms like Scratch. Coding often involves logical thinking and problem-solving, which are key math skills.

- ### 12. **Math in Stories**
- **Math Stories**: Read books that incorporate math concepts. For example, "The Grapes of Math" by Greg Tang or "The Number Devil" by Hans Magnus Enzensberger.

- ### 13. **Math in Games**
- **Math Bingo**: Create bingo cards with math problems. Call out the answers, and the first to complete a row wins.
- **Math Jeopardy**: Create a Jeopardy-style game with math questions. This can be a fun way to review different topics.

.....

- ### 15. **Math in History**
- **Math in History**: Discuss how math has evolved over time and its importance in history. This can help provide context and make the subject more interesting.

3. Outputs endlessly

Figure 5: Output degradation from AWQ quantization, demonstrating textual repetition and nonsensical content.

User Prompt: Es-tu prêt à répondre un quizz logico-math ?

Qwen2.5-7B-Instruct

Oui, je suis prêt à répondre à un quiz logico-mathématique ! Je peux essayer de résoudre différents types de problèmes, que ce soit des énigmes, des équations, des problèmes de raisonnement logique, etc. Alors, n'hésite pas à me poser tes questions !

Qwen2.5-7B-Instruct SageAttention

Q!!!!!!!!!!!!!!1!!!!!!.....

User Prompt: prepare sample question paper for cbse class 12 mathematics

Qwen2.5-7B-Instruct

Certainly! Below is a sample question paper for CBSE Class 12 Mathematics. This paper is designed to cover a range of topics from the CBSE Class 12 Mathematics syllabus, including Calculus, Algebra, Vectors, and Probability. The paper is divided into four sections: A, B, C, and D, with varying marks for each section.

.....

Qwen2.5-7B-Instruct SageAttention

Certainly!!!!!!!.....

Meaningless output

Figure 6: A rare failure case in SageAttention, where the model produces meaningless output for a specific prompt.

User Prompt: The daily wages for a group of 200 persons have been obtained from a frequency distribution of a continuous variable 'X', after making the substitution $U = (X - 130) / 20$

$U =$
 $(X - 130) /$
 20
 $-2 \rightarrow 1 \ 0 \ 1 \ 2$
 No. of
 persons
 $7 \ 50 \ 60 \ 3$
 Calculate the median, lower and upper quartiles, 3rd decile and 98th percentile.

Qwen2.5-7B-Instruct

To calculate the median, lower quartile, upper quartile, 3rd decile, and 98th percentile from the given frequency distribution, we first need to understand the distribution of the variable \sqrt{U} . The variable \sqrt{U} is a transformation of \sqrt{X} given by $\sqrt{U} = \sqrt{\frac{X - 130}{20}}$. This means that the values of \sqrt{U} are centered around 0, and the distribution of \sqrt{U} can be used to find the corresponding values of \sqrt{X} .

Step 1: Determine the cumulative frequencies for \sqrt{U}

The given data is:

- $\sqrt{U} = -2$: 7 persons
- $\sqrt{U} = -1$: 50 persons
- $\sqrt{U} = 0$: 80 persons
- $\sqrt{U} = 1$: 60 persons
- $\sqrt{U} = 2$: 3 persons

The total number of persons is 200. We can now calculate the cumulative frequencies:

- For $\sqrt{U} = -2$: 7
 - For $\sqrt{U} = -1$: 7 + 50 = 57
 - For $\sqrt{U} = 0$: 57 + 80 = 137
-

Qwen2.5-7B-Instruct-SnapKV

To solve for the 50th percentile, 11, (median), and the 80th percentile, we need to first understand the given data and then proceed step-by-step.

Given data:

- 10th percentile: 10
- 20th percentile: 20
- 30th percentile: 30
- 40th percentile: 40
- 50th percentile (median): 50
- 60th percentile: 60
- 70th percentile: 80th percentile: 80

incoherent output

To find the 50th percentile (median), 80th percentile, and the 70th percentile, we can use linear interpolation since the percentiles are not given directly.

50th Percentile (Median)
 The 50th percentile is given directly as 50.

80th Percentile
 The 80th percentile is given directly as 80.

Figure 7: Incoherent and illogical output from SnapKV, highlighting a severe performance collapse.

User Prompt: Ask me to answer a high-school math problem. If I say "next", ask me another question. Keep of the questions in a bulleted list. If I type "?", give me the answer.

Qwen2.5-7B-Instruct

Sure, let's start with a high-school math problem. Here's your first question:

- * What is the value of x in the equation $3x - 7 = 11$?

Incoherent and repetitive output

Qwen2.5-7B-Instruct-Wanda

Sure! Let's start with a high school math problem. Here are the questions:

- What is the problem you need help with? Is it algebra, geometry, trigonometry, or something else?

When you're ready, type "next" to proceed to the next question. If you type "?" at any time, I will provide the answer.

Figure 8: Performance loss from Wanda pruning, resulting in largely incoherent and unreliable output.