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

Anonymous Author(s)
Affiliation
Address
email

Abstract

Existing evaluation frameworks for Large Language Model (LLM) compression rely on proxy metrics like Perplexity and curated benchmarks, leading to a significant gap between reported scores and real-world performance. To bridge this gap, we introduce a new evaluation paradigm that abandons proxies to directly measure generative faithfulness. Our framework has two core innovations: (1) Conditional Generation Accuracy (CGA), a metric that evaluates a compressed model's ability to replicate an original model's next-token predictions under a teacher-forcing regime, and (2) an evaluation set of real user queries, ensuring alignment with practical applications. Using this framework, we conduct a large-scale evaluation of 9 mainstream compression techniques across model sizes (7B–32B) and context lengths (8K–24K). Our findings challenge prior claims: KV cache dropping methods severely underperform, while a sparse attention baseline unexpectedly surpasses popular quantization techniques. Moreover, we uncover distinct scaling laws: quantization accuracy degrades sharply with longer contexts, whereas the fidelity of sparse attention improves. To promote transparent and reproducible research, we will release a public leaderboard. Click anoymous github to access our implementation of all the compression methods and the evaluation pipeline for all the metrics discussed.

1 Introduction

The rapid adoption of Large Language Models (LLMs) has created an urgent demand for efficient inference under constrained GPU resources. Model compression has emerged as a promising solution, with many training-free or calibration-light approaches claiming substantial inference speedups with only marginal accuracy loss. However, a persistent disconnect remains between reported benchmark performance and real-world user experience. This discrepancy motivates a critical re-evaluation of existing evaluation frameworks for LLM compression.

Current evaluation practices predominantly rely on Perplexity (PPL) and Question Answering (QA) benchmarks, both of which suffer from fundamental limitations.

Limitations of Perplexity. Perplexity is plagued by two key issues. First, its unbounded value range and model-specific baselines make fair cross-model comparison difficult. More importantly, PPL is easily exploitable, certain compression strategies, such as KV cache dropping, can be over-optimized to perform well on PPL while producing outputs that diverge significantly from the original model, thereby undermining practical usability. Our experiments confirm that such methods may excel in PPL metrics yet fail to preserve output fidelity.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Limitations of QA Benchmarks. In response to the shortcomings of PPL, recent works have shifted toward QA-style downstream evaluations. While these efforts attempt to expand the evaluation space by incorporating additional dimensions (e.g., fairness, privacy, commonsense reasoning), they remain fundamentally constrained by rigid formats such as multiple-choice or binary classification. These formats poorly reflect the nature of real-world model interactions, which often involve generating long-form, coherent, and contextually grounded text, especially in multi-step reasoning tasks. Moreover, reliance on answer accuracy introduces new biases and may incentivize overfitting to narrow evaluation schemas.

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

To address this issue, we introduce a novel evaluation framework that eliminates proxies in favor of direct, fidelity-based evaluation grounded in real-world usage scenarios. Our framework is built on two core innovations:

Conditional Generation Accuracy (CGA): A Direct Evaluation Metric. We propose CGA as a robust and interpretable metric for measuring the fidelity of compressed models. Unlike prior work that relied on edit distance or BERTScore, which are affected by output length sensitivity and pretraining bias respectively, CGA directly measures a compressed model's ability to replicate the original model's output distribution. Under a teacher-forcing paradigm, we feed the ground-truth response (generated by the original model) to the compressed model token-by-token, and evaluate prediction accuracy for the next token (Acc@1, Acc@5). This approach provides a precise and faithful assessment of how well the compressed model preserves the original generation behavior.

Real User Queries as the Evaluation Distribution. Instead of synthetic or curated benchmarks, we utilize authentic user queries sourced from public platforms like ShareGPT. These queries better reflect actual deployment scenarios—they are open-ended, diverse, and often lack a single "correct" answer. By using the original model's output as the reference, our framework remains effective even in ambiguous, multi-modal contexts. To enable a more granular analysis, we leverage GPT-4o [xx] to classify tens of thousands of these queries into distinct domains such as mathematics, programming, medicine, and summarization. In parallel, we stratify these queries by context length. This dual categorization allows us to assess how well a compressed LLM preserves its capabilities not only in specific areas of expertise but also across different input scales.

Using this framework, we conducted a large-scale, head-to-head evaluation of 10 mainstream LLM compression methods spanning four categories. Our pipeline supports model sizes from 7B to 32B and context lengths from a few dozen tokens to 24K, yielding scores that are inherently bounded within [0, 1], ensuring robust comparability. Key findings include:

Key Takeaways

Cross-Method Insights

- Overall performance ranking: low-precision attention $>$ INT4 quantization $>$ 50% pruning $>>$ KV cache dropping.
- Differences between categories are far greater than within-category variation.
- Among INT4 quantization methods, GPTQ consistently outperforms AWQ in both accuracy and generalization.
- KV cache dropping methods (e.g., SnapKV, H2O) underperform significantly, often producing outputs that are poorly aligned with the original model, raising concerns about their practical viability.
- Notably, Top-10% attention, a sparse attention approximation, surpasses INT4 quantization methods and FlashAttention FP8 in performance.

Scalability with Model Size

- Low-precision attention (FlashAttention FP8, SageAttention) and INT4 quantization (GPTQ, AWQ) benefit substantially from larger model scales.
- Unstructured pruning (SparseGPT, Wanda) and Top-10% attention exhibit consistent performance across model sizes.
- KV cache dropping methods show a significant preference for smaller models.

Scalability with Long Contexts

- All INT4 quantization methods degrade sharply in accuracy as context length increases.
- FlashAttention FP8 deteriorates significantly with long contexts, while Top-10% Attention maintains or improves accuracy.
- KV Cache compression methods degrade markedly as context grows, contradicting their claims of long-context suitability.
- Pruning-based methods (SparseGPT, Wanda) and SageAttention show the most stable performance across varying context lengths.

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

2 Related Works

LLM Compression. The deployment of LLMs on resource-constrained hardware has catalyzed extensive research into model compression. Key lossy techniques include low-precision attention, quantization, pruning, and KV cache compression, with state-of-the-art methods in each category often claiming significant efficiency gains at minimal performance cost.

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

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

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

Pruning. Pruning methods aim to improve efficiency by removing redundant model parameters. While simple magnitude-based pruning is a common baseline, it risks removing weights crucial for model performance. To mitigate this, more sophisticated methods have been developed. SparseGPT introduced a calibration process to achieve high levels of sparsity (e.g., 60%) with less accuracy loss. Wanda refines this by removing weights with the smallest magnitudes after multiplying them by the norms of their corresponding input activations, providing a more robust measure of a weight's contribution.

KV Cache Dropping. To accelerate the decoding phase of generative inference, several methods focus on compressing the KV cache. H2O identifies and retains the most influential "heavy-hitter" KV pairs based on their cumulative attention scores, evicting less important pairs to stay within a fixed token budget. Building on this, SnapKV further compresses the prompt's KV cache, aiming to reduce memory and computational overhead even more.

Compressed LLMs Evaluation. The evaluation of compressed LLMs has evolved rapidly. Early work predominantly relied on PPL, but this metric has been widely criticized for its susceptibility to being "gamed" and its poor correlation with downstream task performance.

This led to a shift towards using comprehensive QA benchmarks. Researchers have proposed increasingly complex evaluation suites that measure performance across multiple dimensions, such as fairness, privacy, and ethics, as seen in works like [1] and [4]. Others, such as LLM-Kick [2], have focused on aggregating numerous existing QA tasks into a single, extensive leaderboard. However, these benchmarks are typically limited to structured formats like multiple-choice questions. This format diverges significantly from the open-ended, generative nature of real-world user interactions, introducing a new form of evaluation bias and creating the risk that compression methods will be over-optimized for these specific test formats.

Sharing our motivation to move beyond proxy metrics like PPL and QA benchmarks, a concurrent work [3] directly compares the outputs of compressed and original models. This approach has explored metrics such as edit distance and BERTScore, but both have inherent drawbacks. Edit distance is unreliable for longer generations, as minor initial errors can cascade into large, misleading divergence scores. BERTScore, while designed to capture semantic similarity, introduces its own set of potential biases by relying on an external pretrained model.

3 A Direct, Fidelity-Based Evaluation Framework

Let $\mathcal{F}$ and $\tilde{\mathcal{F}}$ denote an original LLM and its compressed counterpart, respectively. For a given user prompt X, the models generate output sequences $Y = \mathcal{F}(X)$ and $\tilde{Y} = \tilde{\mathcal{F}}(X)$. An ideal evaluation would measure the "quality" of $\tilde{\mathcal{F}}$ by comparing $\tilde{Y}$ to Y over a distribution of prompts X that reflects real-world usage.

Existing evaluation frameworks fall short on two fronts: the data distribution and the comparison metric.

Proxy Data Distributions. Current benchmarks rely on proxy data distributions that fail to capture the complexity of real user interactions. QA benchmarks, for instance, use structured prompts (X_{QA}) that require selecting from a fixed set of answers, e.g., $\{A, B, C\}$. This constrained format cannot assess a model's performance on the ambiguous, open-ended, and multi-turn conversational prompts common in practice. PPL benchmarks often use large text corpora (e.g., WikiText) which, while vast, are stylistically monolithic and unrepresentative of conversational queries. This mismatch between benchmark data and real-world data introduces a fundamental evaluation bias.

Proxy Comparison Metrics. Instead of directly comparing the output sequences Y and $\tilde{Y}$, prevailing methods rely on proxy metrics. The final score S in these frameworks is not a direct measure of similarity, but a comparison of proxy values derived from the model outputs. For PPL, the score compares the perplexity values, not the text itself:

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

Similarly, QA benchmarks compare final scores on a task, abstracting away the generative process entirely:

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

The use of these proxies—perplexity and multiple-choice accuracy—obscures the evaluation. A compressed model might achieve a similar PPL or QA score to the original while generating text that is factually incorrect, stylistically different, or incoherent. This indirection is the primary source of the gap between benchmark results and real world user experience.

3.1 Aligning with Real-World Data Distributions

To ensure our evaluation reflects practical use cases, we curated a dataset from real-world user interactions. We sourced tens of thousands of conversations from ShareGPT, a platform containing diverse, multilingual user prompts. To facilitate a granular analysis of model capabilities, we further processed this data in two ways:

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

Long-Context Strata. To analyze performance under long-context scenarios, we segment the data into three distinct length-based tiers: 8K, 16K, and 24K tokens. Each tier consists of approximately 20 prompts, creating a testbed that mirrors the demands of real-world, long-context applications.

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

3.2 Conditional Generation Accuracy: A Direct Fidelity Metric

The ultimate measure of a compressed model's quality is its ability to replicate the generative behavior of the original model. Existing proxy metrics, such as task-specific scores, fail to capture the nuanced

aspects of generation, including tone, style, and the reasoning process embedded in the output sequence.

To overcome this, we introduce Conditional Generation Accuracy (CGA), a metric that directly compares the output distributions of the compressed and original models. The standard auto-regressive generation process for the original model $\mathcal{F}$ and a compressed model $\tilde{\mathcal{F}}$ at step i is given by:

$$y_i = \mathcal{F}(X, y_1, y_2, \ldots, y_{i-1}) \quad \text{and} \quad \tilde{y}_i = \tilde{\mathcal{F}}(X, \tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_{i-1}), \tag{3}$$

where a naive comparison of y_i and $\tilde{y}_i$ is unreliable, as a single-token divergence early in the sequence can lead to a cascade of "errors", making the remainder of the generated text entirely different. This error accumulation problem renders simple metrics like edit distance or BERTScore ineffective for long sequences.

To isolate the predictive capability of the compressed model at each step, we employ a teacher-forcing strategy. Instead of conditioning the compressed model's next-token prediction on its own previously generated tokens ($\tilde{y}_{<i}$), we condition it on the ground-truth sequence from the original model ($y_{<i}$):

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

Here, $\hat{y}_i$ represents the compressed model's prediction at step i given the ideal context from the original model. This approach decouples the evaluation at each step from errors made in prior steps, eliminating the issue of cascading failures and allowing for a pure measure of the compressed model's predictive fidelity.

The final CGA score is the mean accuracy over all tokens in the sequence, averaged across all prompts in the dataset:

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

As the formula shows, CGA directly measures the alignment between the compressed and original models' next-token predictions under the golden output distribution. It uses no proxies, providing the most direct and fundamental assessment of generative faithfulness.

4 Experiments

Our experimental evaluation is structured as follows. First, we validate our proposed Conditional Generation Accuracy (CGA) metric by benchmarking it against standard metrics and human-aligned judgments (Section 4.1). Second, we conduct a comprehensive, head-to-head comparison of all selected compression methods to assess their overall performance (Section 4.2). Finally, we analyze the scaling properties of each method with respect to both model size (7B to 32B) and context length (8K to 24K) in Section 4.3 and Section 4.4, respectively.

Table 1: Validation of evaluation metrics on a 7B model. While proxy metrics like PPL and MMLU often fail to reflect true performance degradation, our proposed CGA score aligns closely with human-aligned GPT Scores. Misleading scores from proxy metrics are underlined.

	INT4 Quant		**Low-Precision Attn**			**KV Cache Comp.**		**Pruning**		
Metric	AWQ	GPTQ	Flash FP8	Sage	Top-10%	SnapKV	H2O	Sparse	Wanda	**Baseline**
PPL	17.76	16.79	116.27	16.00	15.99	15.57	18.56	21.37	20.89	15.99
MMLU	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	0.895	0.042	0.000	0.289	0.303	1.0
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

4.1 Validating the CGA Metric

We begin by validating our central claim: that CGA aligns more closely with human-perceived quality than do traditional proxy metrics like Perplexity (PPL) or QA accuracy. For this analysis, we used a 7B model and employed GPT-4o judgments as a high-quality proxy for human perception.

Figure 1: 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 exceeding that of proxy metrics like PPL and MMLU.

Evaluation Setup

- **PPL:** We calculated the average score on the WikiText-2 dataset. To facilitate comparison, raw PPL scores were normalized to a [0, 1] range by applying a sigmoid function to the PPL difference between the compressed and original models.
- **QA Score:** We used the MMLU benchmark in a zero-shot setting, evaluated via lm-eval-harness (v0.4.9). The final score is the ratio of the compressed model's accuracy to the baseline's.
- **GPT Score:** We prompted GPT-4o to perform a pairwise comparison between the outputs of the original and compressed models for a given prompt, with their presentation order randomized. The final score is the win rate of the compressed model. The prompt template is available in Appendix [xx].

Results. As shown in Table [xx], both PPL and MMLU scores can be misleading. PPL, in particular, shows poor correlation with GPT Score, especially for KV cache compression methods, which it rates favorably despite significant output degradation. While the MMLU score correctly identifies poorly performing methods, it quickly saturates near 1.0 for most methods, failing to distinguish between moderately and highly faithful models. For instance, H2O achieves an MMLU score of 0.94, masking a substantial drop in generative quality as judged by GPT-4o.

Further analysis in Table [xx] shows the Pearson correlation coefficient between each metric and the GPT Score. The results confirm that CGA has a significantly stronger correlation with human-aligned judgments than PPL or MMLU. While the GPT Score itself is a useful proxy, its inherent variance, prompt sensitivity, and high cost make it unsuitable as a scalable, primary evaluation metric. CGA, in contrast, offers a robust, reproducible, and well-correlated alternative.

— Define custom colors for ranked highlighting —

Table 2: Refined comparison of 9 compression methods on Qwen2.5-7B-Instruct. Performance is measured by Conditional Generation Accuracy (CGA). In each row, cells are highlighted to show the performance ranking: 1st , 2nd , and 3rd .

		Compression Method								
Category	**Sub-dataset**	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

4.2 Head-to-Head Comparison of Compression Methods

We next conducted a large-scale benchmark of 9 compression methods on a 7B model using our CGA framework. This direct comparison provides a clear overview of their relative performance. The complete results are summarized in Table **??**, with corresponding results for 14B and 32B models available in Appendix [xx].

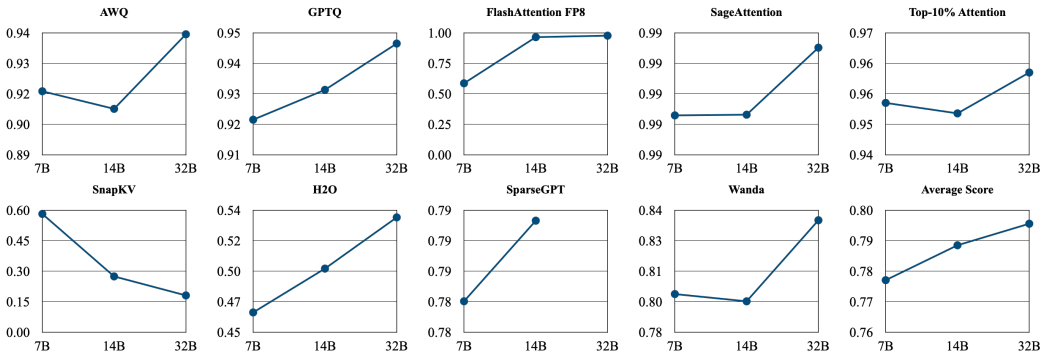

Figure 2: Caption

The results reveal several key insights:

- There are significant performance gaps between method categories. KV cache compression methods, in particular, dramatically underperform relative to their published claims. At the default 20% budget, their CGA scores are the lowest, suggesting their generalization capabilities warrant reassessment.
- The comparison between FlashAttention FP8 and SageAttention confirms that naively replacing BF16 with FP8 for all attention computations leads to a severe performance drop. SageAttention's hybrid-precision approach successfully mitigates this degradation, aligning with its reported findings.
- Top-10% Attention, a sparse attention baseline, delivered surprisingly strong performance. Despite its high effective sparsity (90%) on shorter sequences, it ranked second only to SageAttention, highlighting the viability of sparse attention as a high-fidelity compression strategy.

4.3 Scaling with Model Size

Theoretically, larger models exhibit greater parameter redundancy, suggesting they should be more amenable to compression. To test this hypothesis, we evaluated how the fidelity of each compression method scales with model size, using the 7B, 14B, and 32B versions of the Qwen2.5-Instruct model. The results are plotted in Figure [xx].

Most methods, including quantization and low-precision attention, become more faithful as model size increases, confirming the initial hypothesis. While AWQ, Top-10% Attention, and SparseGPT show a minor, anomalous dip at the 13B scale, the overall trend remains positive. SnapKV is the notable exception; its fidelity degrades as the model grows larger. This analysis not only provides a new dimension for comparing compression techniques but also empirically supports the notion that parameter efficiency decreases with model scale.

4.4 Scaling with Context Length

Low-precision attention and KV cache compression are often motivated by their potential to improve efficiency in long-context scenarios. Our framework is naturally suited to evaluating performance on prompts of varying lengths. We stratified our test data from ShareGPT into three tiers based on token count (8K, 16K, and 24K) to assess how the fidelity of each method scales as the context window expands.

The results, shown in Figure 3, reveal three distinct scaling behaviors:

- All quantization methods (GPTQ, AWQ), KV cache dropping methods (H2O, SnapKV), SparseGPT, and SageAttention show a clear decline in fidelity as context length increases.
- FlashAttention FP8 and Wanda maintain relatively consistent performance across all context lengths.
- Top-10% Attention is unique in that its fidelity measurably improves with longer contexts.

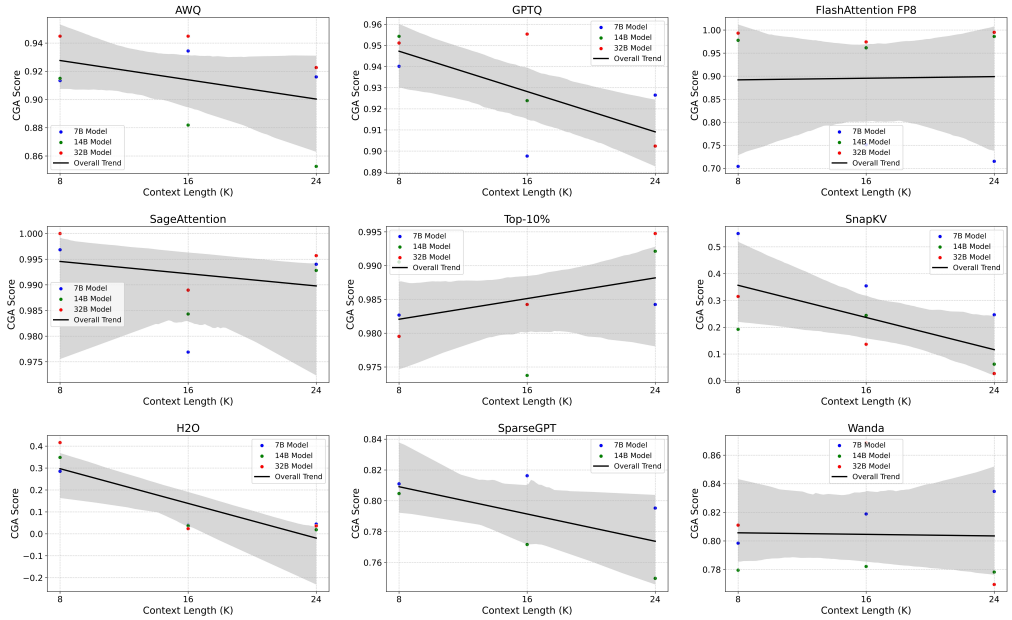

Figure 3: Scaling of compression method fidelity with increasing context length. Methods exhibit distinct behaviors: degradation (quantization, KV cache dropping), stability (Wanda, FlashAttention FP8), or improvement (Top-10% Attention).

These findings demonstrate that even methods within the same category possess different scaling properties, a critical factor for selecting an appropriate compression strategy for long-context applications.

5 Conclusion and Limitations

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

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

References

[1] J. Hong, J. Duan, C. Zhang, Z. Li, C. Xie, K. Lieberman, et al. Decoding compressed trust: scrutinizing the trustworthiness of efficient llms under compression. In *ICML*, 2024.

[2] A. Jaiswal, Z. Gan, X. Du, B. Zhang, Z. Wang, and Y. Yang. Compressing llms: The truth is rarely pure and never simple. In *ICLR*, 2024.

[3] Q. Wang, M. Wang, N. Feldhus, S. Ostermann, Y. Cao, H. Schütze, et al. Through a compressed lens: Investigating the impact of quantization on llm explainability and interpretability. *arXiv preprint arXiv:2505.13963*, 2025.

[4] Z. Xu, A. Gupta, T. Li, O. Bentham, and V. Srikumar. Beyond perplexity: Multi-dimensional safety evaluation of llm compression. In *EMNLP*, 2024.