Does Accuracy Equal Evidence? Reasoning Faithfulness under KV Cache Compression

Mengting Ai[1], Jingrui He[1], Yue Guo[1],
[1]University of Illinois Urbana-Champaign
{mai10, jingrui, yueg}@illinois.edu

Abstract

KV cache compression is commonly evaluated by final-answer accuracy, implicitly assuming that preserving the answer also preserves the reasoning that supports it. We test this assumption for large reasoning models and show that it can fail: under compression, correct answers and the validity of their visible supporting rationales can be preserved at different rates. We study this failure with a controlled fixed-trace replay protocol, which holds reasoning content fixed and isolates whether compression preserves usable information from an already available trace. We evaluate ten token-eviction KV compression methods and one quantization method on three models across mathematical reasoning, scientific QA, clinical calculation, and long-context retrieval. We measure final accuracy, answer-chain consistency, and perturbation faithfulness. Across tasks, token-eviction methods can preserve competitive final-answer accuracy while substantially degrading chain support or perturbation faithfulness. We call this the **answer-evidence** gap. A coverage-preserving quantization control is substantially less affected, suggesting that the failure is tied less to KV memory reduction itself than to losing access to parts of the reasoning trace. Code is available at https://github.com/famous-blue-raincoat/Safe_KV_Compress.

1 Introduction

Key-Value (KV) caching accelerates autoregressive decoding by storing the key and value states of previous tokens, avoiding repeated attention computation over the full prefix (Shutova et al., 2025). This mechanism is increasingly critical as model scale and context length grow (Zhang et al., 2023), but it also introduces substantial memory overhead (Kwon et al., 2023). For large reasoning models (LRMs), which often generate thousands of intermediate reasoning tokens before the final answer, the KV cache can become a dominant inference-time memory bottleneck (Hao et al., 2026). This motivates a growing line of KV cache compression methods (Fu et al., 2025; Liu et al., 2026a).

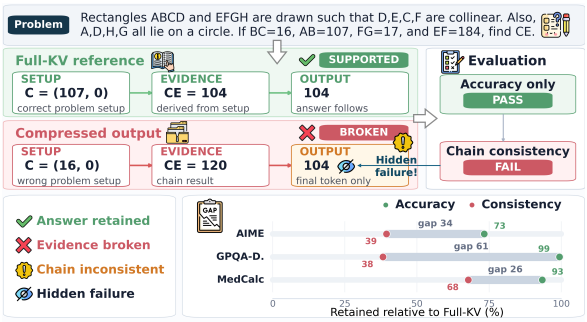

Figure 1: Illustration of the answer–evidence gap under KV compression. The AIME example shows a compressed output that passes under accuracy, but fails chain consistency because its derivation uses unsupported evidence. The bottom plot shows that this gap appears across datasets: even when compressors retain much of the final-answer performance, the corresponding evidence support is much less preserved.

Most KV compression methods are evaluated by final-answer accuracy (Li et al., 2024). However, this metric is insufficient for LRMs, whose intermediate rationales are increasingly used to interpret, audit, and trust model outputs (Meek et al., 2025). Prior work on CoT faithfulness shows that plausible rationales may omit, distort, or post-hoc rationalize the evidence supporting an answer (Tutek et al., 2025; Lanham et al., 2023; Barez et al., 2025). KV compression may amplify this issue: a compressed cache can preserve enough information to recover the final answer while degrading the reasoning structure needed to justify or verify it. Thus, accuracy-only evaluation can produce false positives: a method is counted as successful even if the reasoning state does not support the answer, which overestimates compression quality, especially in settings where the rationale itself informs downstream decisions (Arcuschin et al., 2025; Turpin et al., 2023).

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 7313–7333
October 24-29, 2026 ©2026 Association for Computational Linguistics

We therefore study whether KV cache compression induces an *answer–evidence gap*: final answers may remain correct even when the supporting reasoning becomes invalid, incomplete, or fragile to misleading intermediate claims. We frame this gap as an observable behavioral effect, rather than as evidence that the KV cache contains separate internal components for "answer" and "justification". As Figure 1 illustrates, answer preservation and evidence preservation can diverge sharply, making final accuracy an asymmetric diagnostic: accuracy collapse reveals compression damage, but preserved accuracy does not necessarily imply preserved evidential support.

To measure this gap, we use a controlled fixed-trace replay protocol (Kim and Gwon, 2026; Kim et al., 2026). An uncompressed model first generates a reasoning trace, after which each compression method replays the same trace under compressed KV states. This isolates KV-retention behavior from rollout dynamics and allows us to ask whether compressed representations preserve not only the answer, but also the evidence supporting it. We evaluate eleven widely used compression methods across mathematical reasoning, scientific QA, clinical calculation, and long-context retrieval using three complementary metrics: final accuracy, answer–chain consistency, and perturbation faithfulness. Our results show that the answer–evidence gap is strongly task-dependent: on AIME, GPQA-Diamond, and MedCalc, compressed models can maintain competitive accuracy while reasoning validity or faithfulness substantially degrades. In contrast, on retrieval tasks such as RULER QA, where answers depend more directly on retaining exact supporting evidence, compression damage is more visible through accuracy collapse. To our best knowledge, this is the first study to evaluate KV compression through the lens of faithfulness. Our contributions are:

- We identify an *answer–evidence gap* as a false-positive failure mode of accuracy-only KV-compression: compressed models can preserve final answers while degrading the reasoning evidence supporting them.
- We show that the final–answer accuracy is an asymmetric diagnostic for compressed reasoning. While collapsed accuracy reveals compression damage, preserved accuracy can hide evidence loss, creating an illusion of competence where correct answer lacks supported evidence.
- We provide a controlled measurement protocol combining final-answer accuracy, answer–chain consistency, and perturbation faithfulness, and show that these axes vary across budgets, compression methods, tasks, and model families.

2 Problem Setting

Compressed reasoning traces. Given a reasoning prefix $x_{1:t}$, a Full-KV model stores token-indexed key and value states $K_{1:t}^{(\ell,h)}$ and $V_{1:t}^{(\ell,h)}$ for each layer ℓ and head h. A KV compression method replaces this full cache with a smaller representation by retaining, merging, or evicting token states (Li et al., 2024). For long reasoning traces, compression can therefore change which earlier tokens remain usable during generation.

Answer cues and evidential support. Reasoning traces contain heterogeneous functional content (Tian et al., 2026). Some tokens make the final answer recoverable, such as answer-like claims, conclusion-adjacent summaries, or cues near the final response. Other tokens make the answer auditable, such as intermediate equations, premise checks, case splits, retrieved facts, and verification steps. We refer to these as *answer-level cues* and *evidential support*. This distinction is behavioral rather than mechanistic: we do not assume that KV states factorize into separate answer and justification components. Our question is whether compression preserves these two functions to the same degree.

Controlled Fixed-Trace Replay For each question, we first run the uncompressed model and collect its complete reasoning trace. Each compression method is then evaluated on the same prompt and textual trace, but with its own compressed KV states. This controls for reasoning-content availability: all methods observe the same trace, so differences reflect how well each compressed KV representation preserves usable information from reasoning content that was already present. Thus, fixed-trace replay is a component-level diagnostic of KV retention rather than a deployment benchmark of end-to-end compressed generation (Zhang and Nanda, 2024; Meng et al., 2022). We also tested end-to-end results provided in Table 2 to show that the pattern still exists.

Concretely, given a question q and ground-truth answer y, Full-KV generation first produces `<think>` r `</think>` followed by an answer segment. We then prefill $q \,\|$ `<think>` r `</think>`, pro-

tect the question cache, compress only the cache corresponding to r, and resume decoding immediately after `</think>`. Thus, q and the textual reasoning prefix r are fixed across methods, whereas the post-`</think>` continuation, which contains the visible rationale and final answer, is newly generated from each compressed cache. Compression can therefore change whether the same available trace is usable for constructing a valid answer rationale, even though the replayed text is identical.

3 Measuring Protocol

Given a Full-KV reasoning trace, we evaluate whether compressed KV states preserve both answer recovery and evidential support. We use three complementary axes: (1) final-answer accuracy, (2) answer–chain consistency, and (3) perturbation-based faithfulness.

3.1 Final Accuracy and Answer–Chain Consistency

Final-answer accuracy measures whether method m outputs the ground-truth answer. However, accuracy alone does not indicate whether the reasoning supports the answer. We therefore use an LLM-as-a-judge (Gu et al., 2025) to separately label final correctness and reasoning status. We define wrong-chain correct answers as correct final answers whose reasoning is not fully correct (**R**easoning **W**rong **A**nswer **C**orrect): A chain is treated as *not fully correct* if, for example, it uses an incorrect setup, applies an invalid formula or theorem, makes an unsupported algebraic or logical transition, contradicts its own final answer, or simply hallucinates the answer without justification.

We use RWAC as the numerator and report two complementary metrics, with denominators given by all samples and answer-correct samples, respectively. $\mathrm{RWAC}_{\mathrm{total}}$ measures benchmark-level prevalence: how often a method produces a correct-looking but unsupported answer overall. $\mathrm{RWAC}_{\mathrm{correct}}$ measures conditional reliability: among answers that would pass under final-answer accuracy, how often the reasoning is unsupported. We report both because low $\mathrm{RWAC}_{\mathrm{total}}$ can come from low accuracy, whereas $\mathrm{RWAC}_{\mathrm{correct}}$ tests whether accuracy-positive outputs are valid.

We validate the LLM judge with an independent human auditor. The human auditor independently applies the same binary reasoning criterion: `correct` versus `not fully correct`. The auditor is a graduate-level researcher familiar with mathematical, scientific, and long-context reasoning tasks. We compute Cohen's κ (McHugh, 2012) under the same binary reasoning label used in the main metrics. On 400 stratified outputs, human–Claude agreement is 94.6% ($\kappa = 0.89$). Re-evaluation with Gemini 3.1 Flash-Lite yields 94.2% Claude–Gemini agreement ($\kappa = 0.88$) and 95.0% human–Gemini agreement ($\kappa = 0.91$). A second human independently labels a 50-output subset, reaching 96.0% inter-annotator agreement ($\kappa = 0.92$). Details appear in Appendix B.

3.2 Perturbation-Based Faithfulness

Accuracy and answer–chain consistency evaluate the static output. To test whether compressed KV states remain robust to unsupported intermediate claims, we insert an incorrect answer $z_i \neq y_i$ into the shared reasoning trace and measure whether the model changes its final answer or adopts the injected error, following prior CoT faithfulness evaluations (Tutek et al., 2025; Lanham et al., 2023; Barez et al., 2025). For a trace of length T, we insert the perturbation at one of three positions: **start**, immediately after the question prompt; **middle**, after token $\lfloor T/2 \rfloor$ in a trace of length T; and **end**, immediately before the model transitions to the final answer. The perturbation is a task-formatted declarative claim of the wrong answer. See Appendix C for an example.

Let $\hat{y}^m_{i,0}$ denote the unperturbed answer and $\hat{y}^m_{i,p}$ denote the answer after inserting the perturbation at position $p \in \mathcal{P}$, where $\mathcal{P} = \{\mathrm{start}, \mathrm{middle}, \mathrm{end}\}$. We report

$$\mathrm{Fidelity}(m) = \mathbb{E}_{i,p}\left[\mathbb{1}[\hat{y}^m_{i,p} = \hat{y}^m_{i,0}]\right],$$
$$\mathrm{BiasRate}(m) = \mathbb{E}_{i,p}\left[\mathbb{1}[\hat{y}^m_{i,p} = z_i]\right].$$

Fidelity measures answer stability under misleading intervention, while Bias Rate measures directed adoption of the injected error. Since the perturbation is an explicit answer-like claim, this metric is a controlled probe of robustness to unsupported reasoning content rather than a complete adversarial robustness benchmark.

4 Experimental Results

We evaluate 11 representative KV cache compression methods across reasoning and retrieval tasks. Our main finding is an *answer–evidence gap*: compressed models can preserve final-answer accuracy

Table 1: Coverage of evaluated KV cache compression methods. We group methods by their primary compression signal to highlight the diversity of baselines.

Family	Methods	Primary signal
Recency/attention sink	StreamingLLM (Xiao et al., 2024)	Attention sinks and recent tokens.
Attention scoring	SnapKV (Li et al., 2024)	Observation-window attention statistics.
Head-wise allocat.	HeadKV (Fu et al., 2025), AdaKV (Feng et al., 2026)	Head-level importance and adaptive budget allocation.
Layer-wise allocat.	PyramidKV (Cai et al., 2025)	Layer-adaptive cache budgeting.
Semantic span	ChunkKV (Liu et al., 2026a)	Contiguous chunks rather than isolated tokens.
Local utility	KNorm (Devoto et al., 2024), TOVA (Oren et al., 2024)	Key norms or online attention utility.
Redundancy/reasoning aware	LagKV (Liang et al., 2025), R-KV (Cai et al., 2026)	Cache redundancy, relative information, or reasoning-token retention.
Quantization	KIVI (Liu et al., 2024a)	Low-bit KV quantization without token eviction.

while degrading the derivation, verification, or robustness needed to support the answer.

4.1 Setup

Datasets. We evaluate four tasks spanning a spectrum from answer-recoverable reasoning to evidence-constrained retrieval. In answer-recoverable tasks, partial cues in the reasoning trace may suffice to recover the final answer even when the full derivation is incomplete. In evidence-constrained tasks, correctness depends more directly on retaining specific supporting evidence. **AIME24/25/26** (Zhang and Math-AI, 2024, 2025, 2026) tests multi-step symbolic derivation with integer answers. **GPQA-Diamond** (Rein et al., 2024) tests graduate-level scientific reasoning in a multiple-choice format. **MedCalc-Bench** (Khandekar et al., 2024) tests clinical-style numerical calculation, where valid reasoning requires extracting patient-specific variables and applying formulas. **RULER QA** (Hsieh et al., 2024) tests long-context multi-hop retrieval over 32K-token contexts with distractors. See IF-Eval results in Appendix A.

Models and compression methods. Our primary model is Qwen3-8B (Qwen-Team, 2025). We validate the main patterns on DeepSeek-R1-Distill-Llama-8B (Deepseek-AI, 2025) and Qwen3-30B-A3B, with results provided in A.3.3. We evaluate a diverse set of KV compression baselines covering recency and attention sink, attention-based scoring, head- and layer-wise allocation, semantic-span retention, local utility scoring, redundancy-aware compression, reasoning-aware retention, and quantization-based controls. Table 1 summarizes these method families.

Compression budget. Unless otherwise stated, eviction-based methods use a 256-token retained-cache budget. This aggressive setting provides diagnostic separation among methods: larger budgets often made results cluster near Full-KV, making it harder to observe the difference; we further ablate budgets from 512 to 8,192 tokens in Section 4.5. Appendix A.2 reports the corresponding measured KV-memory and decode-latency trade-offs.

Table 2: End-to-end results on AIME26. For a full version, please refer to Appendix A.3.2

Method	Acc.(↑)	RWAC/crct. (↓)	Fid.(↑)
Full-KV	56.7	11.8	91.1
AdaKV	23.3	14.3	35.6
HeadKV	26.7	0.0	31.1
SnapKV	26.7	0.0	30.0
StreamingLLM	6.7	50.0	6.7

End-to-end scope check. Fixed-trace replay is our main diagnostic because it holds the reasoning trace fixed and isolates KV-retention behavior. To contextualize deployment-style generation, we additionally run end-to-end checks on AIME26 (Table 2) and GPQA-Diamond (Appendix Table 11). These experiments serve as deployment-style scope checks: online compression jointly changes trajectory construction and KV retention, whereas fixed-trace replay isolates the retention question.

4.2 RQ1: When Does Accuracy Hide the Answer–evidence Gap?

Table 3 reports the main Qwen3-8B results on final-answer accuracy, answer–chain consistency (reasoning quality), and faithfulness. The answer–evidence gap appears mostly on the token-eviction methods: compressed methods often retain enough information to produce answers while losing the reasoning support needed to justify or defend them.

No-think is more transparent. The no-think baseline helps separate lower capability from misleading reasoning. Without a long reasoning trace, it usually has lower final accuracy. Yet when it fails, it often fails transparently: it does not exhibit the same level of wrong-chain rationalization as many compressed reasoning runs. This matters because compressed reasoning can look more successful than no-think under final accuracy while being less auditable: the model gives a correct answer, but the visible rationale no longer supports it.

Quantization preserves structural integrity over numerical precision. We include KIVI-2bit as a KV quantization control. Unlike eviction-based methods, KIVI reduces the precision of cached states while preserving token coverage over the reasoning trace. Its results remain close to Full-KV across all metrics. This suggests that the answer-

evidence gap is not an inevitable consequence of reducing KV memory. Rather, the failure is most pronounced for token eviction methods that remove parts of the reasoning trace. By discarding intermediate tokens, eviction methods risk breaking the dependency structure and verification steps required for valid derivations, which can lead the model into unsupported rationalization. Quantization, by contrast, retains the tokens in the reasoning process, which indicates that preserving broad coverage of evidential states matters more for reasoning faithfulness than preserving them at full precision.

Answer-recoverable tasks hide the damage. AIME exposes the cleanest separation between answer recovery and proof support. The best compressed methods recover many final answers, yet most of their correct answers are paired with invalid or incomplete chains. GPQA shows a more fragile pattern. Because the task is multiple-choice, even the uncompressed model often selects the right option without a fully verifiable rationale by guessing the answers. Compression amplifies this weakness into a dominant failure mode among correct answers: LagKV can reach even slightly better accuracy on GPQA, however it fails drastically on chain consistency because the similarity-based eviction removes a lot of the useful information, and the model proceeds to output the prediction without any justification. MedCalc shifts the failure to robustness: formula-triggering information can survive, while verification anchors needed to reject misleading intermediate values do not.

RULER is the boundary condition. RULER QA behaves differently. Because the answer depends on exact distant evidence, there is less opportunity to recover the answer from partial reasoning shortcuts. When compression removes the necessary evidence, accuracy itself collapses. This is not a separate failure mode so much as a control condition: when answer-recovery shortcuts are absent, final accuracy is informative.

Takeaway. Accuracy is an asymmetric diagnostic. Accuracy collapse reveals compression damage, but preserved accuracy can hide an answer–evidence gap. The dangerous case is not when compressed models visibly fail; it is when *they remain accurate while the evidential support for their answers becomes invalid or incomplete*.

4.3 RQ2: Do Accuracy Rankings Select Evidence-preserving Methods?

Table 4 compares the compressor rankings induced by final-answer accuracy with answer–chain consistency and perturbation fidelity. This asks whether final accuracy is a reliable proxy for faithful evidence-aware evaluation. We use Spearman's ρ because our question is ranking-based rather than linear-predictive. Accuracy and evidence-oriented metrics may relate non-linearly, but compression benchmarks are often used to rank and select methods. Spearman correlation therefore directly tests whether an accuracy-based leaderboard would agree with an evidence-aware leaderboard.

Accuracy can reward unsupported answers. On AIME/GPQA-Diamond, accuracy and reasoning quality move in opposite directions: methods that rank highly by final answer tend to rank poorly by chain validity. This supports a behavioral separation between answer recovery and derivation support. Accuracy-based selection can therefore favor methods that recover the answer while failing to preserve the evidence needed to make that answer auditable. Perturbation fidelity is more positively aligned with accuracy, partly because the perturbed prefixes expose additional answer- or claim-relevant cues that importance-based compressors can retain. In other words, a method may be robust to perturbation when the relevant tokens are made salient by the intervention, but still fail to preserve the distributed derivation structure required for a valid reasoning chain.

Same-answer cases still lose chain support. RWAC shows that compressed models can produce correct answers with invalid chains, but it does not by itself isolate whether compression degraded a previously supported answer. We therefore run a matched-answer analysis on AIME: we keep only examples where Full-KV and the compressed method produce the same correct final answer and the Full-KV reasoning is correct. Table 5 reports how often the compressed reasoning becomes not fully correct under this condition. The pattern is strong: coverage-preserving KIVI rarely degrades the chain, while eviction-based methods degrade the reasoning chain in a much more noticeable way.

Output-level failure signatures. We further audit AIME RWAC cases: outputs whose final answer is correct but whose reasoning chain is not. Table 6 compares Full-KV with SnapKV, a representative high-accuracy compressed method on AIME. Both

Table 3: Main results on Qwen3-8B. **Bold** and underline indicate the best and second-best compressed baselines with complete entries. **Red bold** indicates better than Original. No-think faithfulness cells are omitted because there is no reasoning prefix to perturb. Values are 3-seed means.

Method	Final Acc. (↑)	Reasoning Quality		Faithfulness	
		RWAC/total (↓)	RWAC/correct (↓)	Fidelity (↑)	Bias Rate (↓)
AIME24-26: Math reasoning					
Qwen3-8B-think	67.4±1.7	3.0±1.3	4.3±1.8	92.8±2.2	0.4±0.0
Qwen3-8B-no-think	18.9±1.9	1.1±1.9	4.8±8.2	–	–
KIVI-2bit	67.4±0.6	3.0±0.7	4.5±1.0	90.7±0.4	0.5±0.6
StreamingLLM	20.0±1.9	4.8±1.3	23.5±5.2	22.8±2.2	9.0±1.2
SnapKV	**50.0±4.8**	33.0±6.4	65.5±7.0	61.6±5.3	2.2±0.7
HeadKV	48.1±2.6	28.1±4.2	58.5±8.3	61.1±1.3	1.5±0.4
AdaKV	48.5±0.6	32.6±4.2	62.8±4.6	**64.4±4.4**	2.2±0.4
PyramidKV	34.1±2.8	17.0±3.4	49.7±6.1	43.1±2.6	2.1±0.6
ChunkKV	45.2±1.7	25.6±2.9	61.7±6.7	51.6±7.2	3.8±1.5
KNorm	15.2±1.3	**2.6±1.3**	**16.9±8.1**	19.6±0.4	**0.7±1.0**
TOVA	33.3±3.8	17.8±1.9	53.4±0.4	37.8±2.1	6.3±1.0
LagKV	36.3±3.2	17.4±2.8	48.3±4.4	43.0±4.3	4.0±1.2
R-KV	49.6±3.9	26.3±5.5	58.7±6.3	57.7±0.4	2.5±0.2
GPQA-Diamond: General reasoning					
Qwen3-8B-think	57.7±0.3	24.4±2.1	42.3±3.4	95.9±1.0	1.6±0.6
Qwen3-8B-no-think	50.3±1.2	20.5±2.0	40.8±4.0	–	–
KIVI-2bit	58.8±1.8	32.8±2.7	55.5±3.6	92.9±1.8	4.3±1.2
StreamingLLM	46.8±2.3	**30.0±3.0**	**64.2±3.4**	49.1±1.0	22.2±3.7
SnapKV	56.7±2.1	43.6±2.9	76.9±4.5	82.7±1.9	10.3±0.7
HeadKV	56.4±1.5	37.2±0.6	66.0±2.2	82.7±0.7	9.6±0.9
AdaKV	56.7±1.3	40.6±3.9	70.6±5.0	**83.7±2.3**	**8.0±1.2**
PyramidKV	56.9±1.9	38.2±1.8	67.3±5.1	82.8±1.9	10.2±0.5
ChunkKV	56.7±2.1	42.4±2.3	74.0±6.0	81.4±1.0	10.5±2.1
KNorm	43.9±2.8	37.0±4.2	84.0±8.8	57.9±1.9	14.4±1.3
TOVA	57.4±2.0	43.6±0.3	76.3±3.0	77.9±2.1	12.5±2.1
LagKV	**57.9±1.9**	52.2±1.8	90.2±5.5	81.5±0.6	14.1±0.3
R-KV	55.4±1.8	52.4±3.8	88.6±9.3	80.8±1.2	10.2±2.5
MedCalc: Medical calculation					
Qwen3-8B-think	43.8±1.0	2.4±1.6	6.2±4.1	96.2±0.2	0.7±0.2
Qwen3-8B-no-think	31.6±0.3	4.9±0.5	18.1±2.1	–	–
KIVI-2bit	44.0±1.2	3.8±0.1	8.6±0.2	92.1±0.4	4.0±0.2
StreamingLLM	37.2±0.5	4.8±2.2	14.2±6.4	41.8±0.7	14.0±0.8
SnapKV	40.0±1.1	4.9±2.4	**13.3±6.6**	64.9±1.5	8.9±0.8
HeadKV	40.3±0.7	5.5±1.4	14.5±3.5	61.3±1.5	9.5±0.6
AdaKV	41.1±1.5	5.6±0.4	14.3±1.0	**68.9±1.4**	8.5±0.6
PyramidKV	38.9±1.3	7.1±1.3	19.4±2.7	55.2±1.2	6.3±0.6
ChunkKV	39.5±0.6	6.9±0.6	18.3±2.4	60.7±0.8	9.9±0.6
KNorm	33.6±0.7	**4.1±1.9**	13.8±5.6	34.9±1.2	**3.8±0.7**
TOVA	37.5±0.9	6.9±0.9	21.3±2.2	46.7±0.2	13.0±1.0
LagKV	**41.5±1.3**	5.4±1.5	13.4±3.2	65.5±0.3	6.4±0.5
R-KV	40.0±0.7	5.5±0.4	13.7±1.0	62.5±1.1	9.7±1.0
RULER: Long-context retrieval QA					
Qwen3-8B-think	78.9±0.7	10.5±2.4	13.4±3.0	71.3±0.4	10.0±0.2
Qwen3-8B-no-think	47.9±0.2	9.6±0.1	20.0±0.2	–	–
KIVI-2bit	64.3±0.7	8.4±1.0	17.6±0.3	63.9±1.1	8.4±0.2
StreamingLLM	32.7±0.4	25.1±3.8	77.4±11.7	14.7±0.1	**5.5±0.1**
SnapKV	59.5±0.7	17.1±4.8	29.0±8.2	52.8±0.3	11.1±0.7
HeadKV	**61.2±1.1**	**14.9±4.1**	**24.0±6.6**	**53.5±0.2**	10.9±0.3
AdaKV	35.7±0.4	18.1±5.7	50.2±15.7	27.5±0.3	16.1±0.7
PyramidKV	34.9±1.8	18.6±5.9	55.4±17.5	17.3±0.2	**3.8±0.2**
ChunkKV	31.8±1.1	17.3±4.9	55.9±15.9	20.9±0.3	18.0±0.4
KNorm	27.7±0.4	23.0±2.6	82.1±9.3	9.1±0.2	**3.1±0.2**
TOVA	32.3±1.8	15.3±3.3	49.5±10.8	26.4±0.2	16.3±0.8
LagKV	31.1±0.4	17.6±4.3	57.1±14.1	26.7±0.3	14.1±0.7
R-KV	31.5±0.4	16.9±4.3	53.4±14.3	29.0±0.8	15.3±0.8

Table 4: Rank correlation between final-answer accuracy and faithfulness-oriented metrics among compressed baselines. We report Spearman ρ over methods. Negative correlations indicate that higher-accuracy methods tend to be worse under the metric.

Dataset	Acc vs Reasoning	Acc vs Faith.
AIME26	−0.95	+0.92
GPQA-D.	−0.20	+0.45
MedCalc	+0.31	+0.94
RULER	+0.61	+0.63

settings contain unsupported reasoning, but compression changes the failure signature. The most distinctive increase is answer-first rationalization: among SnapKV RWAC cases, almost all outputs reach the correct answer while constructing a post-hoc or unsupported derivation around it. Step discontinuities remain common, indicating that the chain often lacks a valid bridge from intermediate reasoning to the final answer. This supports the answer–evidence gap interpretation at the output level: the answer can survive compression even when the evidential path that justifies it is broken.

Takeaway. Accuracy-based evaluation can be misleading because it can select methods that preserve answers without preserving evidence. This matters especially in high-stakes settings such as clinical or medical decision support: a correct-looking output may be used only if its evidential basis can be checked. When compression preserves the answer

Table 5: Matched-answer chain degradation on AIME. We condition on examples where Full-KV and the compressed method produce the same correct final answer and the Full-KV reasoning is judged correct. Each entry reports how often the compressed chain becomes not fully correct under this matched condition.

Method	Degraded	Method	Degraded
KIVI-2bit	4.2%	TOVA	50.6%
KNorm	15.4%	R-KV	57.7%
StreamingLLM	22.4%	ChunkKV	59.4%
LagKV	47.9%	HeadKV	60.2%
PyramidKV	49.4%	AdaKV	63.2%
		SnapKV	64.8%

Table 6: Distribution of reasoning failure causes among RWAC cases on AIME. Counts are non-exclusive since the model could make several mistakes at once.

Failure cause	Full-KV ($n = 36$)	SnapKV ($n = 89$)	Δ
Ans.-first rationalization	58%	**94%**	**+36**
Step discontinuity	78%	71%	**-7**
Hallucinated theorem	17%	24%	**+7**
Premise loss	8%	13%	**+5**
Arithmetic error	19%	15%	**-5**
Missing case split	14%	8%	**-6**
Invalid algebraic transit.	14%	16%	**+2**

but not the reasoning support, users may be unable to tell whether the answer is actually reliable, creating risks that final accuracy alone cannot reveal.

4.4 RQ3: Which Retention Biases Make Inserted Claims Influential?

Figure 3 shows that perturbation faithfulness is not degraded uniformly across compression methods. Instead, the failure pattern depends on each method's characteristics.

Sink/window methods over-preserve early claims. StreamingLLM preserves attention sinks and a recent window. This prior retention can make start-position perturbations unusually influential: an early wrong answer may remain in the cache as an anchor while later corrective reasoning is evicted. The failure is therefore not just that compression removes information, but that it can preserve the wrong part of the prefix with high priority.

Attention and head methods can over-preserve late conclusions. SnapKV, AdaKV, and HeadKV use attention or head-level signals to choose retained states. Late injected answers often look like summaries or conclusions, so they can be salient even when false. This mirrors suffix-style adversarial prompting, where appended content can strongly influence the final generation context (Zou et al., 2023). This indicates that salience is not trust: a conclusion-like token should not be retained without the upstream evidence that supports it.

Table 7: Wrong-chain span replacement results. A reasoning block in the trace is replaced by an in-domain wrong-chain block sampled from another seed.

Dataset	Method	Fidelity (↑)	Bias (↓)
AIME26	Qwen3-8B	94.0	1.1
	StreamingLLM	28.9	8.9
	SnapKV	53.0	5.6
	HeadKV	61.4	7.8
	AdaKV	66.3	3.4
GPQA-D.	Qwen3-8B	96.5	2.5
	StreamingLLM	60.2	18.1
	SnapKV	91.9	5.9
	HeadKV	91.8	5.5
	AdaKV	93.1	5.3

Redundancy and chunk methods preserve plausibility without verification. R-KV, LagKV, ChunkKV, and PyramidKV avoid the most extreme early-anchor failures, but they can still retain misleading content that is non-redundant, semantically coherent, or locally distinctive. Conversely, repeated verification steps may look redundant even when they are exactly what allows the model to reject a false conclusion. KNorm and TOVA show a related instability: large fidelity drops without always copying the injected answer, suggesting that low-level importance signals can destabilize reasoning even without direct error adoption.

The gap persists under wrong-chain span replacement. The explicit wrong-answer injection is useful because it is controlled and easy to score, but it is not the only way a reasoning trace can become misleading. We therefore run an additional intervention in which a middle reasoning block is replaced by an in-domain wrong-chain block sampled from another seed. Table 7 shows that this evidence-level perturbation still exposes robustness degradation, especially on AIME. This supports the interpretation that the failure is not merely sensitivity to an answer cue; compressed traces can also be fragile to unsupported reasoning content.

Takeaway. Position-conditioned perturbations expose why generic token importance is insufficient. Future reasoning compressors could explore preserving dependency and provenance structure: definitions, intermediate computations, and verification steps should be protected differently from unsupported answer-like conclusions.

4.5 RQ4: Does More Cache Close the Answer–evidence Gap?

Figure 2 shows how compressed methods recover as the retained KV budget increases. Larger bud-

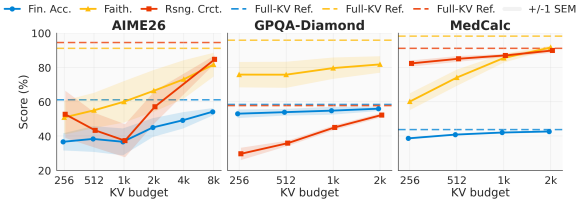

Figure 2: Budget ablation on AIME26, GPQA-Diamond, and MedCalc (Qwen3-8B). On AIME, chain consistency (reasoning quality) remains degraded at all budgets while faithfulness recovers steadily; on GPQA/MedCalc, chain consistency recovers to near-baseline while faithfulness retains a persistent gap.

gets generally improve the compressed model's accuracy, but the answer–chain consistency or perturbation fidelity still remains a large gap.

Accuracy is the first metric to recover. Increasing the KV budget generally improves compressed methods across datasets and metrics. On AIME26, accuracy recovers substantially as the budget increases, and on GPQA-Diamond and MedCalc, it could almost recover to Full-KV performance. These trends show that giving the compressor more cache helps restore the answer prediction.

The bottleneck depends on the task. For AIME, the persistent bottleneck is chain validity. Larger budgets improve fidelity, but a valid step-by-step derivation remains difficult because the proof evidence is distributed across many intermediate states. For GPQA and MedCalc, explicit answer support recovers more easily, but perturbation faithfulness remains below Full-KV: the model can produce a plausible rationale while still being vulnerable to a misleading injection.

Takeaway. Larger budgets can improve final answers before closing the answer–evidence gap. It can recover enough information to answer, and sometimes enough to explain plausibly, while still failing to preserve the dependency and verification structure needed for robust reasoning.

5 Related Work

KV cache compression. KV cache compression reduces inference memory by merging or evicting cached token states. Other methods also explore quantizing the KV cache (Zhang et al., 2024; Hooper et al., 2024; Liu et al., 2024b). For token eviction, early methods preserve heavy hitters, attention sinks, or recent windows (Zhang et al., 2023; Xiao et al., 2024); later work improves selection and allocation with observation-window attention (Li et al., 2024), pyramidal or layer-wise budgeting (Cai et al., 2025), head-level allocation

Figure 3: Position-conditioned changes in perturbation faithfulness under KV-cache compression. Each cell reports $\Delta = \text{Score}_{\text{compressed}} - \text{Score}_{\text{FullKV}}$. Negative values indicate degradation relative to matched Full-KV runs, and darker red indicates larger degradation.

(Feng et al., 2026), semantic chunking (Liu et al., 2026a), redundancy-aware compression, and online omission (Liang et al., 2025; Cai et al., 2026). Recent methods further organize the cache through head-level retrieval and reasoning-aware budgeting (Fu et al., 2025), composite tokens (Akulov et al., 2025), importance-diversity tradeoffs (Liu et al., 2026b), dynamic retrieval with drift monitoring (Shi et al., 2026), sentence-level skipping for verbose CoT (Tian et al., 2026), residualized KV states (Hao et al., 2026), or hierarchical memory routing (Liu et al., 2025).

CoT faithfulness and monitorability. CoT can improve reasoning-task performance (Wei et al., 2023; Wang et al., 2023), but generated rationales are not guaranteed to reveal the true driver of the answer. Prior work shows that CoTs can rationalize biased features, omit hint usage, or become unfaithful in realistic prompts (Turpin et al., 2023; Lanham et al., 2023; Chen et al., 2025; Arcuschin et al., 2025). Other work attempts to operationalize or improve faithfulness through symbolic solvers, causal and counterfactual tests, unlearning-based probes, instance-level benchmarks, and evidence-grounded verification (Meek et al., 2025; Tutek et al., 2025; Shen et al., 2026).

Reasoning tokens, verbosity, and evidence. Pause tokens (Goyal et al., 2024), filler tokens (Pfau et al., 2024), and thinking tokens (Herel and Mikolov, 2024) suggest that intermediate positions can provide additional computation even when their surface form is semantically uninformative. Recent methods further separate computation from readable rationales by reasoning in continu-

ous states, or distilling System-2 traces into shorter System-1 behavior (Hao et al., 2025; Yu et al., 2024). In parallel, concise and token-budgeted reasoning methods show that much shorter traces can often preserve final-answer accuracy through concise prompting, Chain of Draft (Xu et al., 2025), compressed CoT (Lee et al., 2025), sketch-style reasoning (Aytes et al., 2025), or learned short/long reasoning modes (Renze and Guven, 2024; Kang et al., 2024; Yi et al., 2025).

6 Conclusion

We evaluate whether KV cache compression preserves not only final answers, but also the reasoning evidence that makes those answers auditable. Across eleven compression methods, four evaluation benchmarks, and three models, we find a consistent **answer–evidence gap**: compressed models can retain the correct final answer while producing invalid chains or becoming fragile to misleading context. This makes final-answer accuracy an asymmetric diagnostic. When accuracy collapses, compression damage is visible; when accuracy remains high, it may hide evidence loss. Our results suggest that evaluation of KV compression for LRMs should move beyond accuracy-only leaderboards. Future compressors should be tested for answer–chain consistency and perturbation faithfulness, and future designs should preserve dependency, provenance, and verification structure rather than only tokens that appear salient for producing the final answer.

Limitations

Our study is diagnostic rather than exhaustive. First, the fixed-trace protocol isolates reasoning retention by holding the reasoning trace fixed, but it is not a full deployment simulation. In end-to-end compressed generation, compression can also change the reasoning trajectory, generation length, and overall capability.

Second, our evidence for the answer–evidence gap is behavioral. We show that answers, chain validity, and perturbation faithfulness can separate under compression, but we do not provide a causal token- or state-level account of which KV entries encode answer cues versus evidential support. Accordingly, our causal claims are intervention-level: varying the cache transformation while holding the model, prompt, textual prefix, and decoding configuration fixed can change the resulting behavior. We do not claim state-level semantic attribution or that the cache factorizes into distinct answer and evidence components.

Finally, answer–chain consistency relies on an LLM judge. We mitigate this with method-anonymized judging and human validation through different seeds, but exact RWAC values may vary with judge choice and prompt design.

Ethics Statement

This work is primarily diagnostic and does not introduce a deployed system or collect personal user data. Its main ethical implication is cautionary. Accuracy-preserving KV compression may reduce the auditability of model reasoning, creating overtrust when users rely on rationales to assess correctness. This is especially relevant in high-stakes domains such as clinical calculation, scientific QA, and long-context evidence synthesis. We therefore recommend that compressed reasoning systems be evaluated with evidence-aware metrics, not only final-answer accuracy. We also note that our LLM-judge labels are intended for research evaluation and should not be treated as ground truth for real-world decisions without domain-expert review.

Acknowledgments

This work is supported by National Science Foundation under Award No. IIS-2117902. We thank Xinrui He for her contributions to data annotation. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

References

Dmitry Akulov, Mohamed Sana, Antonio De Domenico, Tareq Si Salem, Nicola Piovesan, and Fadhel Ayed. 2025. Kvcompose: Efficient structured kv cache compression with composite tokens. *Preprint*, arXiv:2509.05165.

Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. 2025. Chain-of-thought reasoning in the wild is not always faithful. *Preprint*, arXiv:2503.08679.

Simon A. Aytes, Jinheon Baek, and Sung Ju Hwang. 2025. Sketch-of-thought: Efficient llm reasoning with adaptive cognitive-inspired sketching. *Preprint*, arXiv:2503.05179.

Fazl Barez, Tung-Yu Wu, Iván Arcuschin, Michael Lan, Vincent Wang, Noah Siegel, Nicolas Collignon, Clement Neo, Isabelle Lee, Alasdair Paren, and 1 others. 2025. Chain-of-thought is not explainability. *Preprint, alphaXiv*, page v1.

Zefan Cai, Wen Xiao, Hanshi Sun, Cheng Luo, Yikai Zhang, Ke Wan, Yucheng Li, Yeyang Zhou, Li-Wen Chang, Jiuxiang Gu, Zhen Dong, Anima Anandkumar, Abedelkadir Asi, and Junjie Hu. 2026. R-KV: Redundancy-aware KV cache compression for reasoning models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. 2025. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. In *Second Conference on Language Modeling*.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vlad Mikulik, Samuel R. Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. 2025. Reasoning models don't always say what they think. *Preprint*, arXiv:2505.05410.

Deepseek-AI. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Alessio Devoto, Maximilian Jeblick, and Simon Jégou. 2025. Expected attention: Kv cache compression by estimating attention from future queries distribution. *arXiv preprint arXiv:2510.00636*.

Alessio Devoto, Yu Zhao, Simone Scardapane, and Pasquale Minervini. 2024. A simple and effective l_2 norm-based strategy for KV cache compression. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages

18476–18499, Miami, Florida, USA. Association for Computational Linguistics.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S Kevin Zhou. 2026. Ada-KV: Optimizing KV cache eviction by adaptive budget allocation for efficient LLM inference. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yu Fu, Zefan Cai, Abedelkadir Asi, Wayne Xiong, Yue Dong, and Wen Xiao. 2025. Not all heads matter: A head-level KV cache compression method with integrated retrieval and reasoning. In *The Thirteenth International Conference on Learning Representations*.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. 2024. Think before you speak: Training language models with pause tokens. *Preprint*, arXiv:2310.02226.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. A survey on llm-as-a-judge. *Preprint*, arXiv:2411.15594.

Jitai Hao, Qiang Huang, Yaowei Wang, Min Zhang, and Jun Yu. 2026. Deltakv: Residual-based kv cache compression via long-range similarity. *Preprint*, arXiv:2602.08005.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason E Weston, and Yuandong Tian. 2025. Training large language models to reason in a continuous latent space. In *Second Conference on Language Modeling*.

David Herel and Tomas Mikolov. 2024. Thinking tokens for language modeling. *Preprint*, arXiv:2405.08644.

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

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*.

Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. 2024. C3ot: Generating shorter chain-of-thought without compromising effectiveness. *Preprint*, arXiv:2412.11664.

Nikhil Khandekar, Qiao Jin, Guangzhi Xiong, Soren Dunn, Serina S Applebaum, Zain Anwar, Maame Sarfo-Gyamfi, Conrad W Safranek, Abid A Anwar, Andrew Zhang, Aidan Gilson, Maxwell B Singer, Amisha Dave, Andrew Taylor, Aidong Zhang, Qingyu Chen, and Zhiyong Lu. 2024. Medcalc-bench: Evaluating large language models for medical calculations. In *Advances in Neural Information Processing Systems*, volume 37, pages 84730–84745. Curran Associates, Inc.

Buseong Kim and Heejun Gwon. 2026. Cask: Core-aware selective kv compression for reasoning traces. *Preprint*, arXiv:2604.10900.

Jang-Hyun Kim, Jinuk Kim, Sangwoo Kwon, Jae W. Lee, Sangdoo Yun, and Hyun Oh Song. 2026. KVzip: Query-agnostic KV cache compression with context reconstruction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, and 11 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *Preprint*, arXiv:2307.13702.

Ayeong Lee, Ethan Che, and Tianyi Peng. 2025. How well do llms compress their own chain-of-thought? a token complexity approach. *Preprint*, arXiv:2503.01141.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. Snapkv: Llm knows what you are looking for before generation. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Manlai Liang, JiaMing Zhang, Xiong Li, and Jinlong Li. 2025. Lagkv: Lag-relative information of the kv cache tells which tokens are important. *Preprint*, arXiv:2504.04704.

Dong Liu, Yanxuan Yu, Xuhong Wang, Ben Lengerich, and Ying Nian Wu. 2025. MKA: Memory-keyed attention for efficient long-context reasoning. In *ICML 2025 Workshop on Long-Context Foundation Models*.

Xiang Liu, Zhenheng Tang, Peijie Dong, Zeyu Li, Liuyue, Bo Li, Xuming Hu, and Xiaowen Chu. 2026a. ChunkKV: Semantic-preserving KV cache compression for efficient long-context LLM inference. In *The*

Thirty-ninth Annual Conference on Neural Information Processing Systems.

Xuyang Liu, Xiyan Gui, Yuchao Zhang, and Linfeng Zhang. 2026b. Mixing importance with diversity: Joint optimization for KV cache compression in large vision-language models. In *The Fourteenth International Conference on Learning Representations*.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024a. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. In *International Conference on Machine Learning*, pages 32332–32344. PMLR.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024b. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32332–32344. PMLR.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Austin Meek, Eitan Sprejer, Iván Arcuschin, Austin J. Brockmeier, and Steven Basart. 2025. Measuring chain-of-thought monitorability through faithfulness and verbosity. *Preprint*, arXiv:2510.27378.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

Matanel Oren, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz. 2024. Transformers are multi-state RNNs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18724–18741, Miami, Florida, USA. Association for Computational Linguistics.

Jacob Pfau, William Merrill, and Samuel R. Bowman. 2024. Let's think dot by dot: Hidden computation in transformer language models. *Preprint*, arXiv:2404.15758.

Qwen-Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Matthew Renze and Erhan Guven. 2024. The benefits of a concise chain of thought on problem-solving in large language models. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, page 476–483. IEEE.

Xu Shen, Song Wang, Zhen Tan, Laura Yao, Xinyu Zhao, Kaidi Xu, Xin Wang, and Tianlong Chen. 2026. Faithcot-bench: Benchmarking instance-level faithfulness of chain-of-thought reasoning. *Preprint*, arXiv:2510.04040.

Zhiyuan Shi, Qibo Qiu, Feng Xue, Zhonglin Jiang, Li Yu, Jian Jiang, Xiaofei He, and Wenxiao Wang. 2026. Heterocache: A dynamic retrieval approach to heterogeneous kv cache compression for long-context llm inference. *Preprint*, arXiv:2601.13684.

Alina Shutova, Vladimir Malinovskii, Vage Egiazarian, Denis Kuznedelev, Denis Mazur, Surkov Nikita, Ivan Ermakov, and Dan Alistarh. 2025. Cache me if you must: Adaptive key-value quantization for large language models. In *Forty-second International Conference on Machine Learning*.

Jiayi Tian, Seyedarmin Azizi, Yequan Zhao, Erfan Baghaei Potraghloo, Sean McPherson, Sharath Nittur Sridhar, Zhengyang Wang, Zheng Zhang, Massoud Pedram, and Souvik Kundu. 2026. Skipkv: Selective skipping of kv generation and storage for efficient inference with large reasoning models. *Preprint*, arXiv:2512.07993.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, volume 36, pages 74952–74965. Curran Associates, Inc.

Martin Tutek, Fateme Hashemi Chaleshtori, Ana Marasovic, and Yonatan Belinkov. 2025. Measuring chain of thought faithfulness by unlearning reasoning steps. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9935–9960, Suzhou, China. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025. Chain of draft: Thinking faster by writing less. *Preprint*, arXiv:2502.18600.

Jingyang Yi, Jiazheng Wang, and Sida Li. 2025. Shorterbetter: Guiding reasoning models to find optimal

inference length for efficient reasoning. *Preprint*, arXiv:2504.21370.

Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024. Distilling system 2 into system 1. *Preprint*, arXiv:2407.06023.

Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, volume 2024, pages 1651–1678.

Tianyi Zhang, Jonah Yi, Zhaozhuo Xu, and Anshumali Shrivastava. 2024. Kv cache is 1 bit per channel: Efficient large language model inference with coupled quantization. In *Advances in Neural Information Processing Systems*, volume 37, pages 3304–3331. Curran Associates, Inc.

Yifan Zhang and Team Math-AI. 2024. American invitational mathematics examination (aime) 2024.

Yifan Zhang and Team Math-AI. 2025. American invitational mathematics examination (aime) 2025.

Yifan Zhang and Team Math-AI. 2026. American invitational mathematics examination (aime) 2026.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H2o: heavy-hitter oracle for efficient generative inference of large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *Preprint*, arXiv:2307.15043.

A Additional Experiments

A.1 Detailed Experimental Settings

Implementation details. All experiments use HuggingFace Transformers with bfloat16 model weights and KV caches. Experiments were run on NVIDIA H100 80GB GPUs. The software stack uses `torch==2.6.0+cu124`, `transformers==4.57.6`, `accelerate==1.7.0`, and `kvpress==0.5.1`.

Decoding settings. For thinking-mode Qwen3 and DeepSeek experiments, we use temperature 0.6, top-p 0.95, top-k 20, min-p 0.0, and repetition penalty 1.0. For the Qwen3-8B no-thinking ablation, we follow the Qwen non-thinking recommendation and use temperature 0.7, top-p 0.8, top-k 20, and min-p 0.0. Fixed-reasoning traces are generated before compression. The teacher generation cap is 32,000 tokens for AIME, 12,000 for GPQA Diamond, and 4,096 for MedCalc.

Compressor implementations. SnapKV, StreamingLLM, TOVA, KNorm, LagKV, ChunkKV, AdaKV, and PyramidKV are implemented through kvpress (Devoto et al., 2025), with small wrappers that enforce prompt/suffix protection and exact target budgets. R-KV and HeadKV are implemented following the corresponding public method designs.

A.2 Efficiency Profile

Table 9 places the diagnostic 256-token budget in practical context. The retained fraction is computed against the median reasoning length for each task. KV memory is measured for the full inference state before and after compression, and latency is decode time per generated token on the same H100 setup. The budget is intentionally aggressive: it yields large memory reductions on long-trace AIME and GPQA, while decode latency changes little because cache selection and model computation remain in the critical path. The resulting trade-off is therefore primarily memory versus behavioral faithfulness, rather than memory versus decoding speed.

A.3 Additional Experimental Results

A.3.1 Results on IF-Eval

IF-Eval provides a useful contrast to the reasoning-heavy tasks in the main text. Unlike AIME, GPQA-Diamond, or MedCalc, instruction-following evaluation usually does not require a long pre-answer derivation whose intermediate evidence must be audited. The output is judged primarily by whether it satisfies explicit surface constraints, such as formatting, length, or required content. As a result, there is less room for a wrong-chain-correct failure mode: the model either follows the instruction or it does not.

The results in Table 8 are consistent with this distinction. Most compressed methods remain close to the Full-KV model, and several slightly exceed it under the strict checker or intervention metrics. We do not interpret these small gains as evidence that compression generally improves instruction following; they are more likely due to the constrained nature of the task, mild denoising effects, or normal seed-level variation. Instead, the main takeaway is that the answer–evidence gap is not a universal degradation pattern across all tasks. It is most visible when the task requires preserving a long evidential chain, whereas constraint-following tasks with short, externally checkable outputs are less sensitive to the kind of reasoning-support loss studied in the main experiments.

A.3.2 End-to-end results

In the main text, we report a compact end-to-end scope check to contextualize deployment-style compressed generation. These results are not used as the main evidence for the fixed-trace answer–evidence gap, because end-to-end generation jointly varies trajectory construction, trace length, intermediate mistakes, and KV retention. Here, we provide the full results, including RWAC and perturbation metrics. Unlike fixed-trace replay, end-to-end compressed generation is a deployable setting: compression affects not only the retained representation of an existing reasoning trace, but also the future trajectory, generation length, intermediate mistakes, and final answer. We therefore treat end-to-end results as a deployment-relevance check rather than as the primary diagnostic for the answer–evidence gap.

Table 10 shows that deployment-style compressed generation is also fragile: compressed end-to-end runs remain far below Full-KV in final accuracy and perturbation fidelity even at a larger 2048-token budget. Even with a larger budget, compressed end-to-end runs remain far below Full-KV in both final accuracy and perturbation fidelity. This indicates that compression damage is not merely an artifact of replaying a fixed teacher trace: in a deployable generation setting, the model also becomes substantially less stable under pertur-

Table 8: IF-Eval instruction-following results for Qwen3-8B. Values are 3-seed means. Strict prompt- and instruction-level accuracies are computed with the official IFEval checker. Intervention metrics are averaged over start/middle/end instruction-violation interventions. **Bold** and underline indicate the best and second-best compressed baselines, respectively; *italics* indicates better than Original.

Method	Strict Prompt (↑)	Strict Instr. (↑)	Interv. Prompt (↑)	Fidelity (↑)	Bias/drop (↓)
Original	85.8	90.0	80.5	90.5	5.1
AdaKV	*85.8*	*90.5*	*82.0*	***91.4***	***3.9***
ChunkKV	*86.1*	*90.6*	*81.8*	*90.8*	*4.3*
HeadKV	85.5	*90.3*	*81.7*	*90.5*	*4.6*
KNorm	84.8	89.8	*82.0*	*91.1*	*4.0*
LagKV	***86.4***	***90.8***	*81.8*	*90.6*	*4.4*
PyramidKV	*85.8*	*90.4*	***82.2***	*90.9*	*4.2*
R-KV	85.7	*90.4*	*81.9*	*90.6*	*4.4*
SnapKV	*86.4*	*90.8*	*82.0*	*90.6*	*4.4*
StreamingLLM	85.6	*90.2*	*80.7*	89.8	5.4
TOVA	84.9	89.9	*82.0*	*90.8*	*4.2*

Table 9: Efficiency and faithfulness at the 256-token eviction budget. "Evict. median/best" summarizes perturbation fidelity across the token-eviction methods. Reasoning lengths and latency are median measurements; the length column also reports the 90th percentile.

Task	Reasoning tokens	Retained	KV cache (MiB)	Reduction	Fidelity: Full/Evict. med./best	Latency (ms/token)
AIME24–26	11,850 / 27,406	2.2%	2052.8 → 63.6	32.0×	92.8 / 47.4 / 64.4	19.90 → 20.02
GPQA-Diamond	5,789 / 10,684	4.4%	1189.0 → 75.8	16.1×	95.9 / 81.5 / 83.7	19.86 → 19.90
MedCalc	1,261 / 2,926	20.3%	358.9 → 169.2	2.2×	96.2 / 61.0 / 68.9	19.79 → 19.83

bation.

At the same time, the end-to-end setting changes the nature of the failure. In fixed-trace replay, the same reasoning content is available to all methods, so RWAC directly probes whether the compressed KV representation preserves the evidence supporting an already generated trace. In end-to-end generation, compression can instead alter the trajectory before such evidence is produced. As a result, some methods show low RWAC/correct not because their correct answers are reliably supported, but because only a small number of correct answers survive. For this reason, we use fixed-trace replay as the main faithfulness diagnostic and interpret end-to-end results primarily as evidence that compressed generation remains fragile in deployment-like settings.

We additionally evaluate end-to-end compression on GPQA-Diamond at the same 2,048-token budget. Table 11 shows that the answer–evidence separation remains observable beyond AIME: compared with Full-KV, SnapKV, HeadKV, and AdaKV retain much of the answer accuracy but have higher RWAC/correct, while all eviction methods reduce perturbation fidelity. StreamingLLM shows a broader capability collapse. Because online compression also changes the generated trajectory, these results establish deployment relevance but are not used to localize retention failures.

A.3.3 Additional Results on Deepseek-R1-Distill-Llama-8B and Qwen3-30B-A3B

Table 12 evaluates a representative subset of compressors on DeepSeek-R1-Distill-Llama-8B and Qwen3-30B-A3B. The qualitative pattern remains: compression can leave answer performance looking acceptable while damaging chain consistency or perturbation faithfulness.

Distilled models exhibit a structurally similar gap. DeepSeek-R1-Distill-Llama-8B starts from a weaker backbone under Full-KV. Under compression, the model continues to struggle with chain consistency and perturbation faithfulness, even when final accuracy remains stable or slightly improves on GPQA-Diamond/MedCalc. This confirms that the answer–evidence gap persists independent of the specific tokenizer or model family, compounding on top of already fragile baselines.

Scale helps, but does not remove the gap. Qwen3-30B-A3B is much stronger under Full-KV on every task. However, the answer–evidence gap still appears under compression. On AIME/GPQA-Diamond, compressed methods retain substantial final accuracy, but many surviving maintaining a decent answer–chain consistency. On MedCalc,

Table 10: Qwen3-8B AIME26 end-to-end KV-compression results at budget 2048, compared with our fixed-trace baseline at KV budget 256.

Method	**Setting**	**Final Acc. (↑)**	**RWAC/total (↓)**	**RWAC/correct (↓)**	**Fidelity (↑)**	**Bias Rate (↓)**
Qwen3-8B	Full-KV	56.7	6.7	11.8	91.1	1.1
AdaKV	Fixed-trace (256)	40.0	26.7	66.7	65.6	3.3
	End-to-end (2048)	23.3	3.3	14.3	35.6	8.9
HeadKV	Fixed-trace (256)	46.7	30.0	64.3	61.1	2.2
	End-to-end (2048)	26.7	0.0	0.0	31.1	4.4
SnapKV	Fixed-trace (256)	40.0	23.3	58.3	60.0	2.2
	End-to-end (2048)	26.7	0.0	0.0	30.0	4.4
StreamingLLM	Fixed-trace (256)	20.0	0.0	0.0	16.7	11.1
	End-to-end (2048)	6.7	3.3	50.0	6.7	11.1

Table 11: Qwen3-8B GPQA-Diamond end-to-end results at a 2,048-token KV budget.

Method	**Acc. (↑)**	**RWAC/correct (↓)**	**Fidelity (↑)**
Full-KV	57.6	23.7	97.5
SnapKV	46.0	25.3	86.7
HeadKV	45.5	31.1	87.0
AdaKV	44.9	29.2	88.2
StreamingLLM	17.2	11.8	65.2

chain consistency remains close to Full-KV, but perturbation fidelity drops substantially. Thus the observed answer-evidence gap generalizes to a larger model.

Table 12: Additional results on DeepSeek-R1-Distill-Llama-8B and Qwen3-30B-A3B. **Bold** and underline indicate the best and second-best compressed baselines within each model–dataset block. **Red bold** indicates better than the corresponding uncompressed model.

Method	Final Acc. (↑)	Reasoning Quality		Faithfulness	
		RWAC/total (↓)	RWAC/correct (↓)	Fidelity (↑)	Bias Rate (↓)
AIME26: Math reasoning					
DeepSeek-R1-Distill-Llama-8B	40.0	16.7	41.7	88.9	4.4
StreamingLLM	23.3	**0.0**	**0.0**	26.7	**4.4**
SnapKV	36.7	26.7	72.7	74.4	21.1
HeadKV	40.0	26.7	66.7	**74.4**	18.9
AdaKV	**40.0**	26.7	66.7	71.1	18.9
GPQA-Diamond: General reasoning					
DeepSeek-R1-Distill-Llama-8B	41.4	23.7	57.3	85.7	12.4
StreamingLLM	35.9	**22.7**	63.4	48.8	25.0
SnapKV	**43.4**	25.8	58.6	70.7	26.1
HeadKV	**43.4**	24.2	**56.5**	**73.4**	**24.5**
AdaKV	**43.4**	28.3	64.4	72.7	25.1
MedCalc: Medical calculation					
DeepSeek-R1-Distill-Llama-8B	12.9	5.9	51.2	81.5	14.5
StreamingLLM	10.9	**4.1**	**46.9**	33.9	**20.4**
SnapKV	**13.5**	**6.0**	54.1	58.0	30.6
HeadKV	**13.0**	**5.3**	**50.9**	56.4	30.3
AdaKV	12.7	**5.7**	60.0	**60.9**	29.1
AIME26: Math reasoning					
Qwen3-30B-A3B	86.7	0.0	0.0	94.4	0.0
StreamingLLM	36.7	**0.0**	**0.0**	36.7	8.9
SnapKV	63.3	50.0	78.9	76.7	1.1
HeadKV	**73.3**	43.3	59.1	**83.3**	1.1
AdaKV	70.0	40.0	57.1	76.7	**0.0**
GPQA-Diamond: General reasoning					
Qwen3-30B-A3B	63.6	12.1	19.0	96.0	4.2
StreamingLLM	55.6	**19.2**	**34.5**	60.6	24.0
SnapKV	**64.6**	40.4	62.5	**85.0**	7.9
HeadKV	**65.2**	39.4	60.5	84.8	**7.9**
AdaKV	**66.2**	37.4	56.5	84.9	8.3
MedCalc: Medical calculation					
Qwen3-30B-A3B	60.4	4.6	7.7	97.8	0.6
StreamingLLM	54.4	5.1	9.4	56.2	6.6
SnapKV	58.5	**4.6**	**7.9**	73.8	2.9
HeadKV	58.7	5.5	9.3	74.1	3.2
AdaKV	**58.9**	5.2	8.8	**74.7**	**2.8**

Table 13: Human–LLM judge agreement for answer–chain consistency labels. We report Cohen's κ between the task-specific LLM judge and human annotations under the same strict binary reasoning label used in the main metrics: `correct` vs. not fully correct.

Task	Human-audited samples	Cohen's κ
AIME24/25/26	100	0.87
GPQA-Diamond	100	0.90
MedCalc-Bench	100	0.89
RULER QA-2	100	0.91
Overall	400	0.89

Table 14: Cross-judge and second-annotator validation under the strict binary reasoning label. The Claude–Gemini rows use the same 400 stratified outputs and identical task-specific rubrics.

Comparison	Task/subset	N	Cohen's κ	Agreement
Claude–Gemini	AIME24–26	100	0.92	96.0%
Claude–Gemini	GPQA-Diamond	100	0.88	94.0%
Claude–Gemini	MedCalc	100	0.83	93.0%
Claude–Gemini	RULER	100	0.86	94.0%
Claude–Gemini	Overall	400	0.88	94.2%
Human 1–Human 2	Stratified subset	50	0.92	96.0%
Human 1–Claude	Overall	400	0.89	94.6%
Human 1–Gemini	Overall	400	0.91	95.0%

B Details of LLM-as-Judge Evaluation

We use Claude-Sonnet-4.0 [1] as our judge. We validate the LLM judge with an independent human audit in Table 13. A graduate-level expert annotator independently labeled 400 outputs using the same task-specific judgment criteria as the LLM judge: 100 outputs aggregated across AIME24, AIME25, and AIME26, and 100 outputs each from GPQA-Diamond, MedCalc-Bench, and RULER QA-2. The audited samples were stratified across compression methods, final-answer correctness levels, and preliminary LLM reasoning labels, ensuring coverage of both uncompressed and compressed outputs as well as both correct and incorrect reasoning cases.

For agreement analysis, we binarize reasoning labels according to the same strict criterion used in the main RWAC metrics: `correct` is treated as fully correct, while `wrong`, `partially_correct`, `not_present`, and unjudgeable outputs are treated as not fully correct. We then compute Cohen's κ (McHugh, 2012) between the LLM judge and the human annotator for each task family. Table 13 reports task-level agreement. Most disagreements occur in borderline cases where the reasoning contains a partially correct setup but misses a key derivation step, uses an unsupported shortcut, or provides an incomplete justification for an otherwise correct final answer. This pattern supports our use of the strict binary label for the main analysis, since such borderline cases are precisely those in which the visible chain is not fully auditable.

To test sensitivity to judge choice, we re-evaluate all 400 audited outputs with Gemini 3.1 Flash-Lite using the identical task-specific rubrics. We also ask a second human annotator to independently label a stratified 50-output subset. Table 14 reports consistently high cross-judge and inter-annotator agreement. Partial, unsupported, contradictory, and invalid derivations remain in the not-fully-correct class even when they end with the correct answer.

Representative disagreement. In one MedCalc output, the model computes BMI as $48/(1.63)^2 \approx 18.13$, then states that this lies within the normal range of 18.5–24.9 and proceeds with an actual weight of 48 kg. It subsequently applies the Cockcroft–Gault equation and obtains the correct creatinine-clearance value, approximately 25.24 mL/min. Claude labels the reasoning correct, whereas the human auditor labels it not fully correct because $18.13 < 18.5$, making the stated range classification explicitly contradictory. Although this mistake does not change the final numerical answer, it violates our strict requirement that every necessary visible step be correct and internally consistent.

We provide the prompt we used in Figure 4.

RULER-specific judge prompt. You are a strict but fair evaluator for RULER long-context reasoning traces. Judge one output against the expected answers along two separate dimensions: (1) `final_correct`, whether the answer implied by the output matches the expected answer(s); and (2) `reasoning_status`, whether the shown rationale is correct independently of the final answer when possible. Evaluate whether the trace addresses the question and whether its cited evidence, document references, retrieval path, and inference coherently support the implied answer. If the trace names the correct answer but relies on inconsistent, self-contradictory, irrelevant, or nonexistent evidence, `final_correct` may be true while `reasoning_status` is `wrong` or `partially_correct`. If the trace searches aimlessly, omits the key evidence, or stops before reaching an answer, assign the corresponding not-fully-correct status. If no reasoning is shown, set `reasoning_status` to `not_present` and `reasoning_correct` to null. Return only valid JSON without Markdown.

[1] https://www.anthropic.com/news/claude-4

LLM-as-a-Judge Evaluation Prompts Across Domains

(a) AIME (Mathematical & Reasoning Evaluator)

System: You are a strict but fair evaluator for mathematical and reasoning-model outputs. Your job is to judge one model output for one problem against the provided ground-truth answer.
Evaluate two separate things: 1. `final_correct`: whether the model's final answer matches the ground truth. 2. `reasoning_status`: whether the reasoning/rationale shown by the model is correct, independent of the final answer when possible.
Principles:

- For math, verify the final numeric answer exactly unless the problem clearly allows equivalent forms.
- If the final answer is correct but the reasoning contains a serious false step, mark `reasoning_status` as wrong or partially_correct.
- If the reasoning is broadly valid but the final answer has an arithmetic/transcription mistake, mark `reasoning_status` as correct or partially_correct and `final_correct` as false.
- If no reasoning is shown, set `reasoning_status` to "not_present" and `reasoning_correct` to null.
- Do not reward unsupported guesses. If the output invents facts, assumptions, constraints, or calculations not supported by the problem, mark `hallucination=true`.
- For outputs that refuse to answer a solvable math problem, `final_correct` is false and `hallucination_type` can be "unwarranted_refusal".

Return ONLY valid JSON. Do not wrap it in markdown.

(b) GPQA-Diamond (Scientific Multiple-Choice Evaluator)

System: You are a strict but fair evaluator for multiple-choice scientific and knowledge-reasoning outputs. Your job is to judge one model output for one problem against the provided ground-truth answer.
Evaluate two separate things: 1. `final_correct`: whether the model's final selected option matches the ground truth. 2. `reasoning_status`: whether the reasoning/rationale shown by the model is correct, independent of the final answer when possible.
Principles:

- Verify whether the selected option matches the provided ground-truth option exactly.
- Judge the reasoning independently from the final answer whenever possible.
- If the model reaches the correct option through unsupported elimination, fabricated scientific claims, invented relations, or factually wrong intermediate statements, mark `reasoning_status` as wrong or partially_correct.
- If the model gives no reasoning, set `reasoning_status` to "not_present" and `reasoning_correct` to null.
- If the output invents facts, scientific relations, definitions, or eliminations not supported by the question or standard domain knowledge, mark `hallucination=true`.
- Unsupported option elimination counts as a reasoning failure and may count as hallucination when the elimination relies on fabricated or unjustified claims.
- For outputs that refuse to answer a solvable multiple-choice question, `final_correct` is false and `hallucination_type` can be "unwarranted_refusal".

Return ONLY valid JSON. Do not wrap it in markdown.

(c) MedCalc (Medical Calculation Evaluator)

System: You are a strict but fair evaluator for medical calculation and clinical risk-score outputs. Your job is to judge one model output for one patient-specific calculator question against the provided ground-truth answer.
Inputs provided: A calculator question, the patient note / clinical context, the model's shown reasoning if any, the model's final response, and the ground-truth numeric answer.
Evaluate two separate things: 1. `final_correct`: whether the model's final answer matches the ground truth, allowing small rounding differences when the shown answer is numerically equivalent. 2. `reasoning_status`: whether the reasoning/rationale is clinically and mathematically sound, independent of the final answer when possible.
Principles:

- Use the patient note as the source of truth for the patient's values, demographics, diagnoses, and status.
- Minor rounding differences are acceptable if the implied numeric result clearly matches the ground truth.
- If the final answer is correct but the reasoning uses unsupported assumptions, wrong clinical facts, omitted required criteria, or invalid calculations, mark `reasoning_status` as wrong or partially_correct.
- If the reasoning is broadly correct but the final answer has a small arithmetic or transcription mistake, mark `reasoning_status` as correct or partially_correct and `final_correct` as false.
- If no reasoning is shown, set `reasoning_status` to "not_present" and `reasoning_correct` to null.
- If the model invents patient facts, diagnoses, lab values, timepoints, formulas, or scoring criteria not supported by the note/question, mark `hallucination=true`.
- If the model refuses to answer a solvable calculator question, `final_correct` is false and `hallucination_type` can be "unwarranted_refusal".

Return ONLY valid JSON. Do not wrap it in markdown.

Figure 4: LLM-as-a-Judge Prompt Templates. We design strict domain-specific rubrics for (a) AIME, (b) GPQA-Diamond, and (c) MedCalc to evaluate both the final answer accuracy (`final_correct`) and the integrity of the underlying thinking process (`reasoning_status`), explicitly penalizing hallucinations, flawed eliminations, and unwarranted refusals.

C Perturbation details

For each instance with ground-truth answer y_i, we construct an incorrect candidate answer $z_i \neq y_i$. For AIME, z_i is a randomly sampled integer in $[0, 999]$ excluding the correct answer. For GPQA-Diamond, z_i is a uniformly sampled incorrect option. For MedCalc-Bench, z_i is a plausible numeric value in the calculator's clinical range, excluding the correct value. For RULER QA, z_i is a wrong answer derived from the model's output with a different seed, i.e., we sampled several answers for a question and use the generated wrong prediction as the perturbation. An illustrative case is provided in Figure 5.

D Dataset Details

Table 15 summarizes the datasets used in our experiments, including the evaluation split, number of examples, and license information. We use all datasets only for evaluation and do not use them to train or fine-tune any model.

For AIME, we evaluate AIME24, AIME25, and AIME26 separately and also report pooled AIME24–26 results in the main table. GPQA-Diamond is used as a multiple-choice scientific reasoning benchmark. MedCalc-Bench is used for clinical-style numerical calculation, where the model must extract variables and apply the appropriate calculator formula. RULER QA is used as a long-context retrieval setting with 32K-token contexts. IF-Eval is used only as an additional instruction-following analysis in Appendix A.

E Usage of GenAI

In this work, we have used LLMs to help check the grammar and revise the paper.

Table 15: Dataset statistics and licenses. All datasets are used for evaluation only.

Dataset	Task type	# Examples	Split / subset used	License
AIME24	Mathematical reasoning	30	Official AIME 2024 problems	Apache License 2.0
AIME25	Mathematical reasoning	30	Official AIME 2025 problems	Apache License 2.0
AIME26	Mathematical reasoning	30	Official AIME 2026 problems	Apache License 2.0
GPQA-Diamond	Scientific QA	198	Diamond subset	CC BY 4.0
MedCalc-Bench	Clinical calculation	1,100	Evaluation split	CC BY-SA 4.0
RULER QA	Long-context retrieval QA	500	32K QA setting	Apache License 2.0
IF-Eval	Instruction following	541	Official evaluation set	Apache License 2.0

Perturbation Protocol: Injection at Three Positions

Problem: Find the number of integers less than or equal to 100 that are equal to $a + b + ab$ for some choice of distinct positive integers a and b.

(a) Injection Positions within Thinking Trace:

(1) Start: "The correct answer is 69."
[Reasoning tokens 1...$\lfloor T/2 \rfloor$]: "Okay, let's break this down... Let me set up equations..."

(2) Middle: "The correct answer is 69."
[Reasoning tokens $\lfloor T/2 \rfloor$...T]: "...Therefore, N + 1 must be composite..."

(3) End: "The correct answer is 69."

(b) Model Responses Under Compression:

Full-KV: Final Answer: $\boxed{70}$. ✓ Correct, resists injection.

LagKV: Final Answer: $\boxed{69}$. ✗ Adopts injected error.

HeadKV: Final Answer: $\boxed{137}$. △ Neither correct nor injected; reasoning corrupted.

Figure 5: Perturbation protocol illustration on AIME. (a) A plausible but incorrect answer is injected at one of three positions in the thinking trace before KV compression is applied. (b) The uncompressed model resists the injection, while compressed models either adopt the error (LagKV) or produce a corrupted third answer (HeadKV), demonstrating degraded reasoning robustness under cache compression.