Demystifying Entropy-based Selection for Chain-of-Thought Compression in Large Reasoning Models

Sara Candussio[1,*] **Daniel Scalena**[2,3,*]
Luca Bortolussi[1] **Elisabetta Fersini**[2] **Malvina Nissim**[3] **Gabriele Sarti**[4,5]

[1]University of Trieste [2]University of Milano-Bicocca
[3]CLCG, University of Groningen [4]Northeastern University [5]Parallax

[*]Equal contribution

sara.candussio@phd.units.it d.scalena@rug.nl

Abstract

Entropy-based pruning has been proposed as an effective method for compressing Chain-of-Thought (CoT) reasoning with negligible accuracy loss. We test the robustness of low- and high-entropy CoT step selection methods across various models and reasoning tasks, showing that entropy offers no advantage over random pruning in any evaluated setting. Moving from sentences to tokens, we then show that retaining low-entropy tokens seems effective only on mathematical benchmarks. We find this is due to the inherently low-entropy nature of numeric tokens, which also convey semantic content in such problems. Finally, we demonstrate that patching a subset of a few CoT tokens with their original activations recovers near-perfect full-trace performance, providing causal evidence that task information is not concentrated in a small set of CoT tokens identifiable by heuristics, but rather distributed across the full reasoning chain.[1]

1 Introduction

Inference-time scaling via Chain-of-Thought (CoT) reasoning has become the norm for improving the accuracy of large language models (LLMs) on complex multi-step tasks. However, these gains often come at the expense of unnecessarily long traces that quickly fill the models' context windows (Boppana et al., 2026; Scalena et al., 2026a; Qiao et al., 2025), motivating the development of CoT compression methods reducing context size with limited performance loss.

One such strategy involves treating token- or sentence-entropy as a proxy for relevance, pruning large fractions of a trace based on entropy value alone with little to no accuracy drop. Opposite claims coexist in the literature, favoring either the pruning (Wang et al., 2025a; Li et al., 2026; Laaouach, 2025; Kim et al., 2026) or the retention (Huang et al., 2026; Ton et al., 2025; Wang et al., 2026; Xiong et al., 2026) of low-entropy items.

This disagreement suggests that the entropy signal might not be as precise or indicative as presented in the literature. Given that previous works lack comparison to random baseline, one may wonder whether it is indeed entropy the reason for their reported success, or it is generic compression which is effective. Additionally, since compression methods are mostly tested on mathematical benchmarks, we do not know whether the observed success deriving from entropy-based compression strategies is a property of mathematical reasoning or generalises to non-mathematical tasks, too.

In this work, we test whether the claimed advantage of low- and high-entropy selection for CoT compression persists under both conditions, evaluating LLMs of various sizes from three model families (`gpt-oss`, `Gemma-4`, and `Qwen3`) on mathematical, logical and commonsense reasoning tasks. We show that while at the sentence level, entropy generally provides no advantage over random selection, at the token level, an apparent advantage of low-entropy selection emerges only on mathematical benchmarks. Our results demonstrate that this can be explained by the overlap between low-entropy tokens and content-rich number tokens, rather than by any special property of low-entropy tokens. Even for mathematical tasks, selecting number tokens is most effective only when paired with a direct patching of activations. Taken together, our results suggest that the semantic content of a reasoning trace is not concentrated in entropy-relevant locations.

2 Related work

A growing body of work treats token- and sentence-level entropy as a heuristic for identifying relevant content in a CoT trace, under the premise that low-

[1]Code instructions and data at https://github.com/saracandussio/Demystifying-Entropy-Selection.

Proceedings of the 9th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, pages 1047–1066
October 29, 2026 ©2026 Association for Computational Linguistics

entropy tokens are predictable given the preceding context and therefore redundant, while high-entropy tokens mark points of genuine deliberation and reasoning direction. Wang et al. (2025a) distinguish between the latter, dubbed as *forking tokens*, and the remaining vast majority of low-entropy tokens that merely complete reasoning already initiated at such forks. Li et al. (2026) operationalize the same intuition at sentences-level, reporting that on 50 test samples removing up to 80% of low-entropy steps (equivalent to 45% of tokens) leaves accuracy unchanged, while random or high-entropy removal degrades it sharply. They also argue that filtering at sentence-level is preferable to filtering at token-level, since removing individual low-entropy tokens causes a sharp performance drop that they attribute to a loss of *syntactic coherence*. On the other hand, Huang et al. (2026) show that discarding high-entropy tokens can also improve accuracy; they exploit this finding and propose a reward mechanism to bias the model to avoid excessively high-entropy tokens, leading to good compression without performance loss.

On the role of these entropy-related tokens inside the trace, Zhao et al. (2026) propose the distinction between *structural tokens*, i.e. low-entropy tokens exhibiting low semantic content and having a prevalent syntactic role, and *organic tokens*, i.e. the high-entropy ones constituting instead the main skeleton of the reasoning. This distinction is notable since low-entropy tokens are elsewhere treated as bearing the semantic content of a sentence (Wang et al., 2025a; Huang et al., 2026). Ton et al. (2025) point out that a high-entropy step may reflect either a genuine reasoning fork or noise arising from model under-training on a given task, a failure mode that they term an *unidentifiable task*. They also claim that low-entropy steps may be anchor points for subsequent reasoning, such as the end of useful computation in the CoT. Wang et al. (2026) push this observation further: when the model is unsure of the answer, entropy exhibits high variance across different rollouts, making it an unreliable signal of the trace's convergence to the final output. They also show that trace token entropies are not informative at all, preferring the entropy at the first answer token.

Due to this instability of trace entropy as a pointwise signal, Xiong et al. (2026) note that a downward entropy trend is typically associated with longer traces and show that a trend-based reward signal outperforms a purely local, per-token entropy compression criterion. Chen et al. (2026) observe that, although others use per-token entropy directly for threshold-based early-exit decoding (Laaouach, 2025; Kim et al., 2026), this signal is too noisy to threshold in isolation. As a consequence, they compute a moving average over a sliding window, treating these sustained local maxima as the critical points of a trace; reducing these points improves performance on easier problems but substantially degrades it on harder ones.

These works offer no consensus on what entropy really captures in a reasoning trace: it is unclear whether high-entropy tokens mark genuine forks or model uncertainty, and whether low-entropy tokens carry syntax or semantics remains contested across studies. This, combined with the near-total focus on mathematical benchmarks (Wang et al., 2025a; Li et al., 2026; Huang et al., 2026; Ton et al., 2025; Xiong et al., 2026) and the scarcity of random baseline comparisons (present only in Li et al. (2026)), motivates the robustness assessment provided by our work.

3 Method

Entropy definitions Given a reasoning trace, we define the entropy of a token as the Shannon entropy of the model's next-token predictive distribution at that position, restricted to and renormalized over its top-k log-probabilities returned at generation time. The *token-level entropy* is $H_i = -\sum_{t=1}^{k} p_i(t) \log p_i(t)$, where $p_i(t)$ is the normalized probability assigned to a vocabulary token t at sequence position i. We compute *sentence-level entropy* as the mean token-level entropy over the sentence, where sentence boundaries are identified via fixed heuristics (Section 4).

Compressed CoT formatting The thinking region C inside a reasoning trace is identified as the token span between the model's beginning- and end-of-thinking markers. A selection criterion s ranks the candidate units (sentences or tokens) within C and adds them in ranked order until a target token budget $b = \max(1, \min(r \cdot |C|, |C|))$ is reached, where $r \in (0, 1]$ is the retention rate. The selected items are recomposed in the order in which they appeared in the original trace, producing a compressed CoT $\tilde{C}_r^s$.

The compressed trace is wrapped into a reconstructed input $P + \texttt{[BOT]} + \tilde{C}_r^s + \texttt{[EOT]} + S$ where P is the prompt, `[BOT]` and `[EOT]` are model-dependent beginning- and end-of-thinking mark-

ers, $+$ denotes sequence concatenation, and S is a task-specific suffix that elicits a direct answer.[2] The answer is then greedily decoded for a limited number of tokens, forcing the model to rely on a compressed reasoning trace only.

Selection criteria At sentence-level, we define the following ranking criteria: *low-entropy* and *high-entropy* rank sentences by their mean token entropy, respectively retaining the lowest- and the highest-entropy sentences first; *numbers* ranks sentences by the fraction of tokens containing a digit,[3] as a proxy for essential content in mathematical tasks; *low-entropy no numbers* ranks by mean entropy as before, excluding numeric tokens to disentangle their effect on entropy;[4] *random* uniformly samples whole sentences from available candidates.

At the token level, the same five criteria are recomputed at the token granularity, with two additional pattern-based selectors: *newlines* and *end-of-sentence* markers, retaining only token categories that coincide with sentence boundaries. This addition is motivated by the use of end-of-step token positions as probing locations for extracting information regarding the full step (Bogdan et al., 2026).[5]

Compression protocols We evaluate the compressed traces obtained through different selectors s under two protocols. Following prior work presented in Section 2, the compressed context $\tilde{C}^s$ is passed to the model as a new prompt to obtain the resulting answer. However, for the token-level compression analysis, we also run a forward pass over the full, uncompressed trace, caching its per-layer hidden states, and patching them into the corresponding retained positions when generating from the compressed sequence, yielding patched context $\hat{C}^s$. Given a fixed selector s, $\tilde{C}^s$ and $\hat{C}^s$ have the same text, but $\tilde{C}^s$ activations are recomputed from scratch, while $\hat{C}^s$ activations match those of the selected full-CoT tokens across all layers.

[2]When left unspecified, we use the suffix `Therefore, the answer is \boxed{`. Other variants are ablated in Appendix A, producing near-identical results.

[3]This resembles the prompt-based `OnlyNumbers` baseline of Xia et al. (2025), applied here in a stricter setting.

[4]As a consequence, sentences with no non-numeric tokens are excluded from the candidate pool.

[5]Since numbers, newlines and end-of-sentence markers typically match few tokens per trace, we report the actual compression rate rather than the nominal one when their candidate pool is exhausted before the nominal token budget is reached.

Baselines and evaluation Each trace is bounded by the *full CoT* performance, a theoretical upper bound on achievable accuracy. This is compared against the accuracy obtained under $\tilde{C}^s_r$ and $\hat{C}^s_r$, for every selector s and retention rate r. We report the *relative performance retention* (RPR), i.e. the ratio of a certain setting's accuracy to the *full CoT accuracy* as a general measure of compression effectiveness, using exact match with ground-truth answer.[6] To summarize a selector's behavior across the full RPR range, we additionally report the RPR *Area Under the Curve* (AUC) as a function of retention rate r. We use $\Delta_{\text{AUC}} = \text{AUC}_s - \text{AUC}_{\text{random}}$ to quantify a selector s's advantage relative to the random baseline.

4 Experimental setup

Models and datasets We test the generality of entropy-based compression across model families, parameter scales, and reasoning domains using six models spanning three families and multiple sizes: `gpt-oss-20b` and `gpt-oss-120b` (OpenAI et al., 2025), `gemma-4-E4B-it` and `gemma-4-26B-A4B-it` (Team et al., 2026), and `Qwen3-4B` and `Qwen3-14B` (Yang et al., 2025), all operating in reasoning mode. We test these on mathematical reasoning (AIME 2024, 2025, and 2026; Zhang and Math-AI (2024, 2025, 2026); and a subset of 100, 50, and 50 questions from MATH-500; Lightman et al. (2023)), logical puzzles (ZebraLogic; Lin et al. (2025)), and multiple-choice science questions (GPQA-Diamond; Rein et al. (2024)) to ensure our findings are widely applicable across various reasoning tasks.

Trace sampling For each prompt, 8 reasoning traces are generated with temperature 0.7 and top-p 0.9, with a budget of 16384 max new tokens dictated by our compute availability. Traces that do not reach the end-of-thinking marker `[EOT]` within this budget are discarded. Trace compression is evaluated for $r \in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ leaving up to 64 tokens to complete the answer after the suffix.

Entropy measures and sentence segmentation Token-level entropy H_i is computed over the top-$k = 20$ log-probabilities returned by the sampler

[6]Appendix B additionally reports results obtained with a semantic verifier, confirming that the trends discussed below are not an artifact of the matching criterion.

at generation time.[7] Sentences are identified by splitting the thinking region at token-level sentence-ending marks: a token marks the end of a sentence if it contains a period followed by a whitespace or if it ends with a period and the following token begins with whitespace.[8]

Activation extraction We use `NNsight` (Fiotto-Kaufman et al., 2025) to access and extract model internals and to intervene on them. Residual stream activations are collected at each layer over the full, uncompressed trace and subsequently used to overwrite specific positions at all the layers during generation over the compressed CoT $\hat{C}_r^s$. This approach requires no modification to model weights or architecture, and is compatible with all evaluated model families through a shared interface.

5 Results

We organize our findings around five questions:

Does entropy-based sentence pruning outperform random selection? Figure 1 reports the result of *random*, *low-entropy* and *high-entropy* sentence-level pruning as described in Section 3 on `gpt-oss-20b` on AIME25. Across the entire compression range, neither entropy-based selection criterion outperforms *random*: at low compression rates all three selectors perform comparably, while at moderate-to-high compressions (≥ 0.5) *random* retains 78-90% of full CoT performance, compared to 40-79% for *low-entropy* and 60-89% for *high-entropy*. Entropy alone, in either direction, fails to isolate the content necessary for correct reasoning.

	AIME25 (math)			ZebraLogic (non-math)		
Model	rand	Δlow	Δhigh	rand	Δlow	Δhigh
`Qwen-4B`	**0.930**	-0.001	-0.080	**0.950**	-0.045	-0.007
`Qwen-14B`	0.932	+0.011	-0.064	**0.951**	-0.043	-0.001
`gemma-4B`	**0.777**	-0.001	-0.082	**0.936**	-0.016	-0.018
`gemma-26B`	**0.914**	-0.124	-0.056	**0.973**	-0.022	-0.013
`gpt-20b`	**0.872**	-0.129	-0.060	**0.876**	-0.026	-0.052
`gpt-120b`	**0.832**	-0.023	-0.082	0.888	+0.016	-0.027

Table 1: AUC of *random* and Δ AUC of *low* / *high* relative to it over the RPR range $r \in (0, 1]$, for a math-reasoning dataset (AIME25) and a non-mathematical one (ZebraLogic). *random* is bolded when it outperforms both entropy-based selectors. Model names are abbreviated.

[7]We use $k = 20$ as a proxy of the full logit distribution due to vLLM constraints, following Scalena et al. (2026b).

[8]Exclamation and question marks are consequently not treated as sentence boundaries.

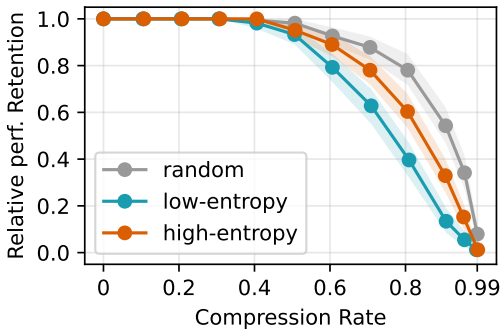

Figure 1: ***Random* dominates both entropy-based selectors across the entire compression range.** RPR vs. Compression Rate for `gpt-oss-20b` on AIME25 for various selection strategies. Shaded bands are 95% bootstrap Confidence Intervals (Appendix C).

Does this hold across model families, sizes and reasoning domains? Table 1 reports, for two representative datasets, the AUC of the *random* baseline and the delta of *low-* and *high-entropy* pruning relative to it, over the full RPR range. Retaining *high-entropy* sentences is a poor pruning strategy regardless of model: on AIME25, Δ_{low} is within noise of zero for `Qwen3-14B`, `Qwen3-4B`, `gemma-4-E4B-it` and `gpt-oss-120b`, but drops sharply for `gpt-oss-20b` and `gemma-4-26B-A4B-it`. On ZebraLogic, Δ_{low} is negative for five of the six models, with `gpt-oss-120b` being the only exception (full breakdown in Appendix D.1). These results suggest that the low-entropy degradation depends on domain rather than model family, and that the entropy pruning assumption fails systematically: across different model families, scales, and domains, no selector significantly outperforms random sentence selection.

Is entropy the right explanation at token-level? Unlike sentence-level pruning, token-level compression for math reasoning tasks benefits from *low-entropy* selection over *random* for most compression rates, while this gap vanishes on non-mathematical tasks (Figure 2, top row). We hypothesize that this asymmetry can be explained with the fact that numerical tokens exhibit significantly lower entropy in mathematical CoTs, in light of the limited vocabulary options for single-token digits.[9] To test whether this confounds the observed trend, we distinguish between *low-entropy no numbers* and *numbers* settings: if numeric tokens are

[9]I.e., when the context strongly implies a number, the model needs to choose only between 0-9 rather than the full set of possible tokens.

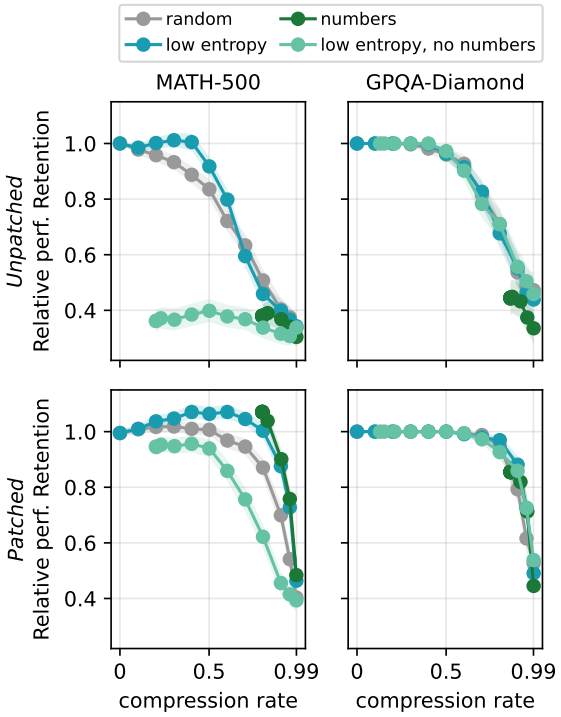

Figure 2: ***Low-entropy* token-level compression outperforms random selection only on mathematical tasks when patching is performed, thanks to numerical tokens.** On MATH-500, *low-entropy no numbers* drops sharply below both *low-entropy* and *random*, while *numbers* is the most performant selector even at extreme compressions. The advantage of *low-entropy* on mathematical reasoning is driven by the limited choice of numeric tokens.

responsible for the gap, the latter should perform comparably to *low-entropy*.

On unpatched compressed traces $\tilde{C}_r^s$, we find that *low-entropy no numbers* and *numbers* both perform significantly worse than low-entropy (Figure 2, top left; Table 6 in Appendix). We attribute this difference to the lack of context fluency produced by token-level filtering, as also highlighted by Li et al. (2026), with the resulting context often containing bare digits and arithmetic operators, or syntactic connectives stripped of their numeric content.[10] This behaviour is expected to disappear when token original activations are added to compressed forward pass: patched compression $\hat{C}_r^s$ results (Figure 2, bottom row) indeed confirm that selecting and patching only numbers in the math domain (*numbers*) exceeds *low-entropy* performance. Selecting instead non-numerical low-entropy tokens (*low-entropy no numbers*) leads to significant degradation, even worse than *random*. On GPQA-Diamond (Figure 2, bottom right), all strategies instead perform on par with random selection. When numerical tokens are not task-relevant content, *numbers* and consequently *low-entropy* selectors are comparable to *random* pruning.

Does token activation patching performance recover generalize to other selectors? Having shown that patching restores *numbers*' advantage on mathematical tasks, we question whether this holds for other selectors across models and benchmarks.

Figure 3 reports the effect of activation patching on AIME25 for all six models: *low-entropy* remains close to the full trace performance up to much higher compression rates than the unpatched setting, retaining ≥ 0.9 relative performance using 20-30% of the original trace (compared to 50-70% without patching). *Numbers* also retains most of the trace's informativeness, almost recovering original performance using only 10-20% of the tokens, while both *low-entropy no numbers* and *high-entropy* show no comparable improvement once patched. Similarly, purely pattern-based selectors (*newlines* and *end of sentence*, that do not carry surface-relevant content) do not perform better under patching (average Δ_{AUC} of -0.74 and -0.73 on AIME25 respectively, both worse than their unpatched versions; full results in Appendix D.3). Compression with patching works only when content-rich tokens are retained: for mathematical tasks, *high-entropy* tokens do not overlap with meaningful content, and *low-entropy* ones owe their effectiveness to the numerical subset they include.

Does discarding either high- or low-entropy elements improve accuracy? Huang et al. (2026) observe that removing high-entropy tokens improves original performance[11], claiming that *their absence reduces noise in the reasoning process*. We observe a superficially similar pattern (see left panels of Figure 2 and Appendix E), but attribute it to the numeric-token overlap discussed above (Section 5), since activation-patched *numbers* is linked to a larger increase in accuracy.

On the opposite, Li et al. (2026) prune low-entropy sentences[12] and conclude that *a vast major-*

[10] A concrete example is shown in Appendix D.2.

[11] Paragraph 2.3, Figure 3 of Huang et al. (2026): averaged across GSM8K (Cobbe et al., 2021), MATH-500, AIME24 and AMC23 (Wang et al., 2025b) and tested on `Qwen3-4B` and `Qwen3-8B`.

[12] Figure 1 left of Li et al. (2026): `DeepSeek-R1-7B`

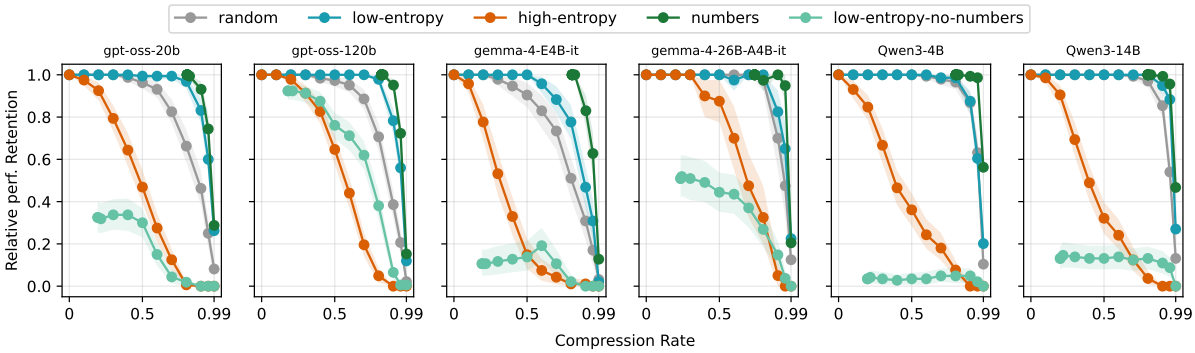

Figure 3: Patching closes the gap for *low-entropy* and *numbers* across all models on AIME25, while *high-entropy* and *low-entropy no numbers* remain largely unaffected or worse.

ity of low-entropy steps are indeed redundant and thus can be discarded without accuracy degradation, while this is not the case for high-entropy and random sentence selection. Our results do also confirm that pruning strategies can lead to improved performances (Appendix E), but do also show that random selection achieves the same at comparable or even higher compression rates (Figure 1, Appendix D.1).

The claim that entropy values (in both directions, either high- or low-) are by their own a signal for reasoning content relevance is not supported by our experiments.

6 Conclusions

In this work, we revisit the widespread assumption that token- or sentence-level entropy are associated to the semantic content of a reasoning trace and therefore can be used as a signal to prune a trace without accuracy drops. We test entropy-based selection against a random baseline across six models (spanning three families and two parameter scales) on both mathematical and non-mathematical reasoning tasks.

At sentence-level, the assumption fails consistently: *random* pruning dominates both *low-* and *high-entropy* selection across every model, scale and domain we tested, with no strategy reliably outperforming chance. At token-level, an apparent advantage for *low-entropy* selection emerges on mathematical benchmarks only; we show this is not an entropy effect at all, but instead a spurious consequence of the overlap between low-entropy and numeric tokens, which alone account for the same gain.

Selecting surface-text is however insufficient to effectively compress the trace, especially when the criterion produces an unreadable trace (e.g. consisting in bare digits or syntactic connectives only): the fresh forward pass performed by the model on this kind of context is uninformative of the reasoning process. When paired with activations, *numbers* and consequently *low-entropy* selectors recover near full-trace performance on mathematical benchmarks at extreme compression rates (using only 10-20% of the original tokens), while the others show no comparable recovery. On non-mathematical datasets, no selector performs significantly better than random.

These results suggest that traces can be compressed without performance loss even with few patched tokens if they correspond to meaningful semantic content, such as numbers in mathematical tasks. The fact that entropy-based selectors do not outperform random pruning (when purified by spurious signals such as numeric tokens on AIME and MATH-500) is an evidence that semantic content of a reasoning trace is not a property that entropy elements carry.

Limitations

Our study is restricted to six open-weight models across three families; while this spans a discrete range of scales and training recipes, we cannot rule out that some closed, bigger (we tested up to 120B, with `gpt-oss-120b`), or simply belonging to another model family with different training strategies (e.g. from the DeepSeek series) would show a different relationship between entropy and reasoning relevant content. Our benchmarks, while spanning both mathematical (AIME, MATH-500) and non-mathematical (GPQA-Diamond, ZebraLogic)

(DeepSeek-AI, 2025) on 50 samples from DeepScaleR (Nalaie et al., 2022); `Qwen3-8B` is also used throughout the experiments.

domains, remain focused on reasoning tasks. We also limit in number and in the size of the non-AIME subsets (50-100 questions each) and the trace length (16384 max new tokens) which may understate the variance of some reported gaps, a choice made for efficient data collection.

The activation patching intervention requires access to model internals and full-precision forward passes over the uncompressed trace, which is not a deployable compression strategy on its own but rather a diagnostic tool used here to test whether adding contextual information to unreadable traces can help recover the full trace performance.

Finally, our selection criteria operate at fixed retention rates that can only be determined post-hoc and that are drastically different according to the original trace length.

References

Edward Beeching, Lewis Tunstall, and Sasha Rush. Scaling test-time compute with open models.

Paul C. Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. 2026. Thought anchors: Which LLM reasoning steps matter?

Siddharth Boppana, Annabel Ma, Max Loeffler, Raphaël Sarfati, Eric Bigelow, Atticus Geiger, Owen Lewis, and Jack Merullo. 2026. Reasoning theater: Disentangling model beliefs from chain-of-thought. In *Forty-third International Conference on Machine Learning*.

Shuang Chen, Hangyu Guo, Yimeng Ye, Shijue Huang, Wenbo Hu, Jiayu Chen, Manyuan Zhang, Haoxi Li, Song Guo, and Nanyun Peng. 2026. ARES: Multimodal adaptive reasoning via difficulty-aware token-level entropy shaping. In *The Fourteenth International Conference on Learning Representations*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Jaden Fiotto-Kaufman, Alexander R. Loftus, Eric Todd, Jannik Brinkmann, Koyena Pal, Dmitrii Troitskii, Michael Ripa, Adam Belfki, Can Rager, Caden Juang, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, Francesca Lucchetti, Nikhil Prakash, Carla Brodley, Arjun Guha, Jonathan Bell, Byron C. Wallace, and David Bau. 2025. Nnsight and ndif: Democratizing access to open-weight foundation model internals. *Preprint*, arXiv:2407.14561.

Chen Huang, Wei Lu, and Wenxuan Zhang. 2026. PEAR: Phase entropy aware reward for efficient reasoning. In *The Fourteenth International Conference on Learning Representations*.

Junyeob Kim, Sang-goo Lee, and Taeuk Kim. 2026. Think just enough: Leveraging self-assessed confidence for adaptive reasoning in language models. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 5000–5006, Rabat, Morocco. Association for Computational Linguistics.

Hynek Kydlíček. 2024. Math-verify: Math verification library. `https://github.com/huggingface/math-verify`. Version 0.6.1.

Yassir Laaouach. 2025. HALT-cot: Model-agnostic early stopping for chain-of-thought reasoning via answer entropy. In *4th Muslims in ML Workshop co-located with ICML 2025*.

Zeju Li, Jianyuan Zhong, Ziyang Zheng, Xiangyu Wen, Zhijian Xu, Yingying Cheng, Fan Zhang, and Qiang Xu. 2026. Making slow thinking faster: Compressing LLM chain-of-thought via step entropy. In *The Fourteenth International Conference on Learning Representations*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *Preprint*, arXiv:2305.20050.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. 2025. Zebralogic: On the scaling limits of LLMs for logical reasoning. In *Forty-second International Conference on Machine Learning*.

Keivan Nalaie, Renjie Xu, and Rong Zheng. 2022. Deepscale: Online frame size adaptation for multi-object tracking on smart cameras and edge servers. In *2022 IEEE/ACM Seventh International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 67–79.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, and 108 others. 2025. gpt-oss-120b and gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Ziqing Qiao, Yongheng Deng, Jiali Zeng, Dong Wang, Lai Wei, Fandong Meng, Jie Zhou, Ju Ren, and Yaoxue Zhang. 2025. Concise: Confidence-guided compression in step-by-step efficient reasoning. *ArXiv*, abs/2505.04881.

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*.

Daniel Scalena, Sara Candussio, Luca Bortolussi, Elisabetta Fersini, Malvina Nissim, and Gabriele Sarti. 2026a. Beyond the commitment boundary: Probing epiphenomenal chain-of-thought in large reasoning models. *Preprint*, arXiv:2606.13603.

Daniel Scalena, Leonidas Zotos, Elisabetta Fersini, Malvina Nissim, and Ahmet Üstün. 2026b. EAGer: Entropy-aware GEneration for adaptive inference-time scaling. In *Forty-third International Conference on Machine Learning*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, Clément Farabet, and 282 others. 2026. Gemma 4 technical report. *Preprint*, arXiv:2607.02770.

Jean-Francois Ton, Muhammad Faaiz Taufiq, and Yang Liu. 2025. Understanding chain-of-thought in LLMs through information theory. In *Forty-second International Conference on Machine Learning*.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. 2025a. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Preprint*, arXiv:2506.01939.

Xi Wang, James McInerney, Lequn Wang, and Nathan Kallus. 2026. EAT: Entropy after $\textlangle \tt /think \textrangle$ for reasoning model early exiting.

Zhiyuan Wang et al. 2025b. Step guided reasoning: Improving mathematical reasoning using guidance generation and step reasoning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Heming Xia, Yongqi Li, Chak Tou Leong, Wenjie Wang, and Wenjie Li. 2025. Tokenskip: Controllable chain-of-thought compression in llms. *ArXiv*, abs/2502.12067.

Xuan Xiong, Huan Liu, Li Gu, Zhixiang Chi, Yue Qiu, Yuanhao Yu, and Yang Wang. 2026. Etr: Entropy trend reward for efficient chain-of-thought reasoning. *Preprint*, arXiv:2604.05355.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

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 Zhao, Sander Land, Daniel M. Bikel, and Waseem Alshikh. 2026. Shorthand for thought: Compressing llm reasoning via entropy-guided supertokens. *Preprint*, arXiv:2604.26355.

A Suffix ablation

To rule out that our choice of answer-eliciting suffix (`Therefore the answer is \boxed {`, footnote of Section 3) does not itself bias the reported accuracies, we run a small ablation on a subset of $N = 25$ questions (we limit this test to one trace per question) drawn from one mathematical benchmark (AIME25) and one non-mathematical one (ZebraLogic) across all six models, comparing the default suffix against $K = 3$ alternative phrasings: `\boxed{`, `Based only on the above, the best answer I can determine is \boxed{`, and `Given the reasoning above, in one sentence, the answer is \boxed{`. We report accuracy under normal greedy decoding of the full trace, since the goal here is to rule out a suffix-induced confound on the upper bound itself rather than to re-run the full compression sweep for every suffix.

Table 2 shows the results: on ZebraLogic, all four suffixes yield identical mean accuracy across the six models (93.3%); on AIME25, the four suffixes vary by at most 1.4 percentage points (67.3-68.7%), well within the cross-model standard deviation (13.1 points). This confirms that the suffix acts purely as a format cue for answer extraction rather than as a source of information that could bias the reported full-CoT accuracies.

Suffix	AIME25	ZebraLogic
`Therefore, the answer is \boxed {`	0.687	0.933
`\boxed{`	0.673	0.933
`Based only on the above, the best answer I can determine is \boxed{`	0.680	0.933
`Given the reasoning above, in one sentence, the answer is \boxed{`	0.687	0.933

Table 2: Full-CoT accuracy by suffix using greedy decoding, averaged across the six evaluated models ($N = 25$ questions per model, one trace per question). Differences across suffixes are negligible on both benchmarks.

B Answer correctness assessment via a semantic verifier

Throughout the main text, correctness is assessed via exact match between the extracted answer and the ground truth. Related works in this space adopt a range of different (and often incompatible) criteria for this same judgement, complicating any direct comparison of reported accuracies. Wang et al. (2025a) inherit the reward verifier of the underlying RL codebase (verl/DAPO), which applies symbolic normalization rather than raw string matching;Huang et al. (2026) adopt the extraction and verification pipeline of Yang et al. (2024); the entropy-after-`</think>` early-exiting method of Wang et al. (2026) relies on canonical-form answer comparison via `sal.math`, the math-evaluation utilities of Hugging Face's *search-and-learn* library (Beeching et al.; Snell et al., 2024), itself adapted from the Qwen2.5-Math evaluation parser; Chen et al. (2026) employ a two-stage criterion, i.e. normalized string matching on the extracted boxed answer followed by an LLM-as-judge fallback when no boxed answer is found or the normalized strings disagree.

Given this variability (and since we cover most of the models and datasets used across this body of work), we recompute all reported curves using `math-verify` (a shared semantic verifier (Kydlíček, 2024)) in addition to the exact-match criterion used in the main text, to check whether our conclusions are sensitive to the choice of matching criterion. Figures 8 and 9 report the sentence- and token-level full grids recomputed under this semantic criterion.

C Bootstrapped Confidence Intervals

Every accuracy we report in this work (a single point on a compression curve or an AUC summary) is itself an average over a noisy set of traces. To quantify measure uncertainty we use a non-parametric bootstrap: we resample the observed traces themselves and read the spread of the resulting statistic directly.

Shaded bands in Figures 1 and 3 are obtained via non-parametric bootstrap over traces, following the same procedure at every retention rate r and for every (model, dataset, selector) combination.

For a given selector s, let $C^s_{i,r} \in \{0, 1\}$ and $C^{\text{full}}_i \in \{0, 1\}$ denote respectively whether trace i is answered correctly under the s-compressed CoT with retention rate $r \in (0, 1]$ and under the full CoT. The point estimate of Relative Performance Retention is

$$\text{RPR}_r = \frac{\sum_i C^s_{i,r}}{\sum_i C^{\text{full}}_i}$$

where the sum runs over all n traces available for that (model, dataset, selector) combination, pooled across all evaluated questions. Since up to 8 traces are sampled per question (Section 4) and traces that do not reach [EOT] within the token budget are discarded (Section 4), n represents the upper bound of the considered traces. We resample this set of n traces with replacement $B = 1000$ times; at each bootstrap iteration b, we draw indices $\{i_1, \ldots, i_n\}$ uniformly with replacement from the original n traces and recompute $\mathrm{RPR}^{(b)}$, discarding any resample with zero denominator. The reported 95% confidence interval is the $[2.5, 97.5]$ percentile range of the resulting bootstrapped distribution $\{\mathrm{RPR}^{(b)}\}_{b=1}^{B}$.

For AUC and consequently Δ_{AUC} (Tables 1, 5), the same per-trace resampling is applied jointly across all retention rates within a given (model, dataset, selector) combination, before recomputing the trapezoidal AUC, rather than resampling independently at each retention rate. Since n is fixed across r for a given combination, the same resampled index set $\{i_1, \ldots, i_n\}$ is reused at every retention rate within a single bootstrap iteration b. This matters because accuracy at neighbouring retention rates is not independent: it is computed from the same underlying traces, and a trace answered correctly at one retention rate tends to remain so at higher rates too. Resampling once per bootstrap iteration and reusing the same index set across all rates preserves this correlation, providing a more faithful interval than treating every point in the curve as an unrelated experiment.

D Detailed results

D.1 Sentence-level pruning

Table 5 reports the full per-model, per-dataset breakdown of Δ_{AUC} for all four sentence-level entropy-based selectors compared to *random*, both over the full retention-rate range and restricted to the aggressive-compression regime ($r \in [0.01, 0.3]$), where any gap tends to widen. Figure 5 shows the corresponding curves for every model-dataset pair across all six evaluated benchmarks. Across the grid, *random* is at or above every entropy-based selector in the majority of settings (bold rows in Table 5); the few exceptions are discussed in Appendix E. The pattern is consistent across both mathematical and non- datasets: none of the four entropy-based selectors (*low*, *low no numbers*, *numbers*, *high*) reliably outperforms random selection at the sentence level, for any model family or scale we tested.

D.2 Token-level pruning

Prompt: *"What is the smallest positive perfect cube that can be written as the sum of three consecutive integers?"*

Selector	Compressed CoT ($r = 0.10$)	Answer
low-	`n+1+2+2333133333^327^3^311 ^3^139^39^339273333/3/3`	27 ✓
low- no numbers	`cube as consecutive n, n++ a+ a+, a+a+ by^ by by k^^^+^^^^^^^ cube,+ + by by by by+x//`	729 ✗
numbers	`1112123331333 13333273273311939 3 19319393127273 27`	729 ✗

Table 3: Tokens retained by each selector on a short MATH-500 excerpt at retention rate $r = 0.10$, and the model's greedy-decoded (unpatched) answer from each compressed text. **Green** marks literal occurrences of the correct answer's digits (27) within the reconstructed text. *low-entropy* generates the correct answer; *low-entropy no numbers* and *numbers* both converge on the same incorrect answer, $27^3 = 729$, despite *numbers* also containing the substring 27 multiple times, showing that lexical presence of the correct digits is not, by itself, sufficient without the surrounding context.

Similarly to the previous section, Table 6 reports Δ_{AUC} for every token-level selector relative to *random*, across all models and datasets (patched results are reported separately in the next subsection). Figure 6 shows the corresponding full grid.

Focusing on the three AIME benchmarks (2024, 2025, 2026) only, the *low-entropy* advantage over random is consistent for four of the six models (`Qwen3-14B`, `Qwen3-4B`, `gemma-4-E4B-it`, `gemma-4-26B-A4B-it`), while both `gpt-oss` model scales show negligible or reversed effect. We attribute this recurring gap to the overlap between low-entropy and numeric tokens rather than to entropy itself, as argued in Section 4 and further isolated in the worked example below.

To make concrete how these three selectors diverge in practice, Table 3 shows the tokens retained by each on an example of a `gpt-oss-20b`'s MATH-500 trace at a fixed retention rate ($r = 0.1$). *low-entropy* retains a mix of numeric tokens and low-information connectives; once numeric tokens are excluded, *low-entropy no numbers* is left choosing almost exclusively syntactic filler, degrading the surface trace to a sequence of connectives stripped of computational content; *numbers* instead retains only digits and operators, producing a terse but computationally dense fragment. This illustrates

why *low-entropy no numbers* tracks *numbers* so poorly once the numeric overlap is removed (Section 4): what remains is syntax, not content.

D.3 Token-level pruning with activation patching

Figure 4 makes the picture from Table 7 concrete in terms of retention budget, averaging the minimum retention rate needed to reach RPR ≥ 0.9 across all six models on AIME25. Patching improves every selector to some degree, including *random*, whose own retention requirement drops from 73% to 32% of the trace once patched. This is expected: patching restores full-context activations at every retained position regardless of which positions were chosen, so it should help any selector to some extent, including one that discards content without regard to relevance. The relevant question is therefore not whether patching helps, but how much, and for which selectors it changes the qualitative picture rather than merely shifting it.

numbers and *low-entropy no numbers* never reach RPR ≥ 0.9 at any tested retention rate without patching, on average (their unpatched bars are capped at 100% in the figure, meaning that the full trace is required to close the gap). Patching changes this qualitatively: *numbers* recovers the threshold using only 9% of the trace, the most efficient recovery among all tested selectors, while *low-entropy no numbers* still requires 70%, only a partial recovery. *low-entropy*, whose unpatched advantage is itself largely inherited from its overlap with numeric tokens (Section 4), needs 47% of the trace unpatched and 22% patched, a proportional improvement comparable to *random*'s, rather than the qualitative jump seen for *numbers*. *high-entropy* weakly improves (93% to 78%), remaining the least efficient selector in both conditions.

E *Numbers* selection exceeds full trace performance

Section 4 notes that there exists some token-level cases in which the compressed CoT performance exceeds full-CoT ones (RPR > 1).

Table 4 reports, separately for token-level selection split by patching condition, the share of all tested (model, dataset, compression rate) points for which the point estimate exceeds 1.0, and, among only those overshooting points, the share for which the 95% CI lies entirely above 1.0. These curves are visible directly in Figures 5 (sentence-level), 6 (token-level), and 7 (token-level, patched), where panels occasionally rise slightly above the RPR $= 1$ line.

Selector	% > 1	% sig. of these	mean
sentence-level			
high-	**2.8%**	**66.7%**	1.050
low-	3.5%	20.0%	1.008
low- no-num	3.5%	26.7%	1.008
num	**4.6%**	25.0%	1.015
token-level, no patch			
high-	0.7%	33.3%	1.016
low-	**2.5%**	0.0%	1.013
low- no-num	0.5%	0.0%	1.003
num	0.0%	—	—
\n	0.0%	—	—
eos	0.0%	—	—
token-level, patched			
high-	4.4%	15.8%	1.016
low-	**9.3%**	**17.5%**	1.020
low- no-num	0.7%	0.0%	1.005
num	**8.6%**	**24.3%**	1.031
\n	0.0%	—	—
eos	0.0%	—	—

Table 4: Overshoot summary, per level, selector, and (for token-level) patching condition, aggregated over all models and datasets. % > 1: share of all tested points whose point estimate exceeds full-CoT accuracy. % sig. of these: among only the points exceeding 1.0, the share whose 95% bootstrap CI lies entirely above 1.0. *newlines* (\n) and *end of sentence* (eos) never overshoot under either condition.

The only selector that is both frequent and reliably above full-trace performance is *numbers* under activation patching: numeric-token activations occasionally carry enough task-relevant information on their own to match or marginally exceed full-trace performance, but only once patching supplies the context those activations would otherwise lack. No selector improves original performance *without* patching and across models; the single isolated case of sentence-level *high-entropy* is limited to `Qwen3-14B` on MATH-500 and therefore cannot be used to argue that either entropy extreme uniquely identifies uninformative content.

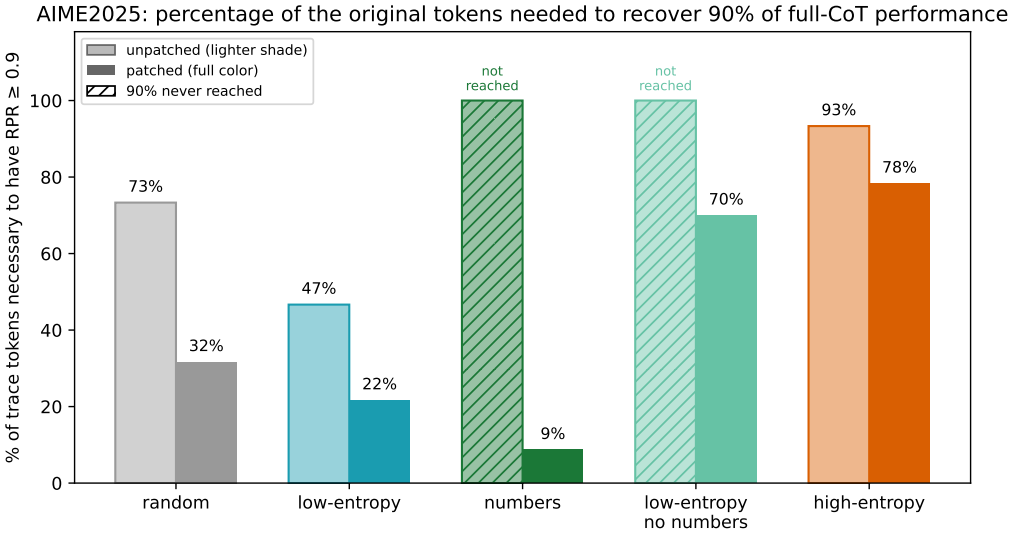

Figure 4: Minimum percentage of trace tokens needed to reach and sustain a RPR of at least 0.9 on AIME25, averaged across all six models, comparing patched (full colour) and unpatched (lighter colour). Hatched bars (capped at 100%) indicate that an unpatched compression using that selector is unable to retain the full trace performance. Patching improves every selector to some degree (including *random*, from 73% to 32%), but the effect is markedly uneven: *numbers* moves from never reaching the threshold unpatched to the most efficient recovery of all once patched (9%), while *low-entropy no numbers* only partially recovers (70%) and *high-entropy* barely improves (93% to 78%).

		Full range $r \in [0.01, 1.0]$					Aggressive compression $r \in [0.01, 0.3]$				
Model	Dataset	random	Δlow	Δlow$_{\text{numbers}}$	Δnumbers	Δhigh	random	Δlow	Δlow$_{\text{numbers}}$	Δnumbers	Δhigh
Qwen3-4B	AIME24	0.93	+0.01	+0.01	-0.04	-0.07	0.79	+0.02	+0.02	-0.14	-0.19
	AIME25	**0.93**	-0.00	-0.00	-0.01	-0.08	**0.77**	-0.00	-0.00	-0.02	-0.18
	AIME26	0.93	+0.02	+0.01	+0.01	-0.08	0.78	+0.07	+0.05	+0.03	-0.16
	MATH-500	**0.94**	-0.01	-0.01	-0.03	-0.03	**0.84**	-0.02	-0.02	-0.07	-0.06
	GPQA-D	**0.97**	+0.00	+0.00	-0.01	-0.01	0.90	+0.02	+0.01	-0.04	-0.02
	Zebra	**0.95**	-0.05	-0.05	-0.11	-0.01	**0.84**	-0.13	-0.14	-0.25	-0.03
Qwen3-14B	AIME24	0.93	+0.02	+0.02	-0.03	-0.05	0.78	+0.07	+0.07	-0.10	-0.12
	AIME25	0.93	+0.01	+0.01	-0.01	-0.06	0.78	+0.03	+0.02	-0.03	-0.15
	AIME26	0.94	+0.01	+0.01	+0.01	-0.06	0.80	+0.05	+0.03	+0.03	-0.14
	MATH-500	1.01	-0.03	-0.03	-0.03	+0.03	**0.96**	-0.05	-0.04	-0.09	+0.00
	GPQA-D	0.95	+0.02	+0.02	-0.01	-0.01	0.85	+0.07	+0.08	+0.01	+0.00
	Zebra	**0.95**	-0.04	-0.04	-0.10	-0.00	**0.85**	-0.12	-0.12	-0.24	-0.00
gemma-4-E4B-it	AIME24	0.80	-0.02	-0.03	+0.05	-0.03	0.45	-0.03	-0.04	+0.13	-0.02
	AIME25	0.78	-0.00	-0.02	+0.10	-0.08	0.43	-0.01	-0.03	+0.21	-0.15
	AIME26	0.78	-0.00	-0.03	+0.16	-0.04	0.43	-0.01	-0.03	+0.36	-0.07
	MATH-500	0.82	-0.01	-0.01	+0.01	-0.01	0.59	-0.02	-0.03	+0.02	-0.02
	GPQA-D	0.90	-0.03	-0.04	+0.02	-0.01	0.74	-0.09	-0.09	+0.04	-0.01
	Zebra	**0.94**	-0.02	-0.02	-0.06	-0.02	**0.84**	-0.05	-0.06	-0.11	-0.05
gemma-4-26B-A4B-it	AIME24	0.93	-0.04	-0.04	+0.01	-0.05	0.78	-0.12	-0.13	+0.01	-0.13
	AIME25	**0.91**	-0.12	-0.14	+0.00	-0.06	0.72	-0.31	-0.35	+0.03	-0.11
	AIME26	0.92	-0.08	-0.09	+0.04	-0.01	0.74	-0.24	-0.26	+0.11	-0.04
	MATH-500	**0.92**	-0.03	-0.02	+0.00	-0.01	**0.81**	-0.07	-0.06	-0.00	-0.05
	GPQA-D	0.98	-0.04	-0.04	+0.01	+0.00	0.92	-0.12	-0.11	+0.02	+0.01
	Zebra	**0.97**	-0.02	-0.02	-0.03	-0.01	**0.91**	-0.07	-0.07	-0.10	-0.04
gpt-oss-20b	AIME24	**0.89**	-0.13	-0.14	-0.08	-0.02	**0.64**	-0.28	-0.30	-0.18	-0.04
	AIME25	**0.87**	-0.13	-0.14	-0.05	-0.06	**0.62**	-0.33	-0.34	-0.12	-0.17
	AIME26	**0.90**	-0.14	-0.17	-0.07	-0.05	**0.70**	-0.36	-0.40	-0.17	-0.14
	MATH-500	**0.92**	-0.04	-0.05	-0.02	-0.02	**0.77**	-0.13	-0.14	-0.04	-0.07
	GPQA-D	0.93	-0.04	-0.03	-0.03	+0.01	0.78	-0.07	-0.05	-0.03	+0.04
	Zebra	**0.88**	-0.03	-0.02	-0.09	-0.05	**0.66**	-0.09	-0.08	-0.19	-0.12
gpt-oss-120b	AIME24	**0.84**	-0.04	-0.05	-0.04	-0.07	**0.55**	-0.10	-0.12	-0.09	-0.13
	AIME25	**0.83**	-0.02	-0.03	-0.01	-0.08	**0.51**	-0.05	-0.06	-0.02	-0.18
	AIME26	0.83	-0.01	-0.02	+0.02	-0.06	0.51	-0.03	-0.04	+0.05	-0.12
	MATH-500	**0.89**	-0.03	-0.03	-0.02	+0.00	**0.70**	-0.07	-0.06	-0.03	-0.01
	GPQA-D	0.90	-0.02	-0.01	-0.04	+0.02	0.72	-0.04	-0.02	-0.03	+0.04
	Zebra	0.89	+0.02	+0.03	-0.03	-0.03	0.70	+0.02	+0.03	-0.09	-0.05

Table 5: AUC of *random* and delta of each entropy-based selector s relative to it ($\Delta = \mathrm{AUC}_s - \mathrm{AUC}_{\text{random}}$), sentence-level selection, shown for the full retention-rate range and for the aggressive-compression regime side by side. **Bold**: all four deltas non-positive (random weakly outperforms or ties every selector).

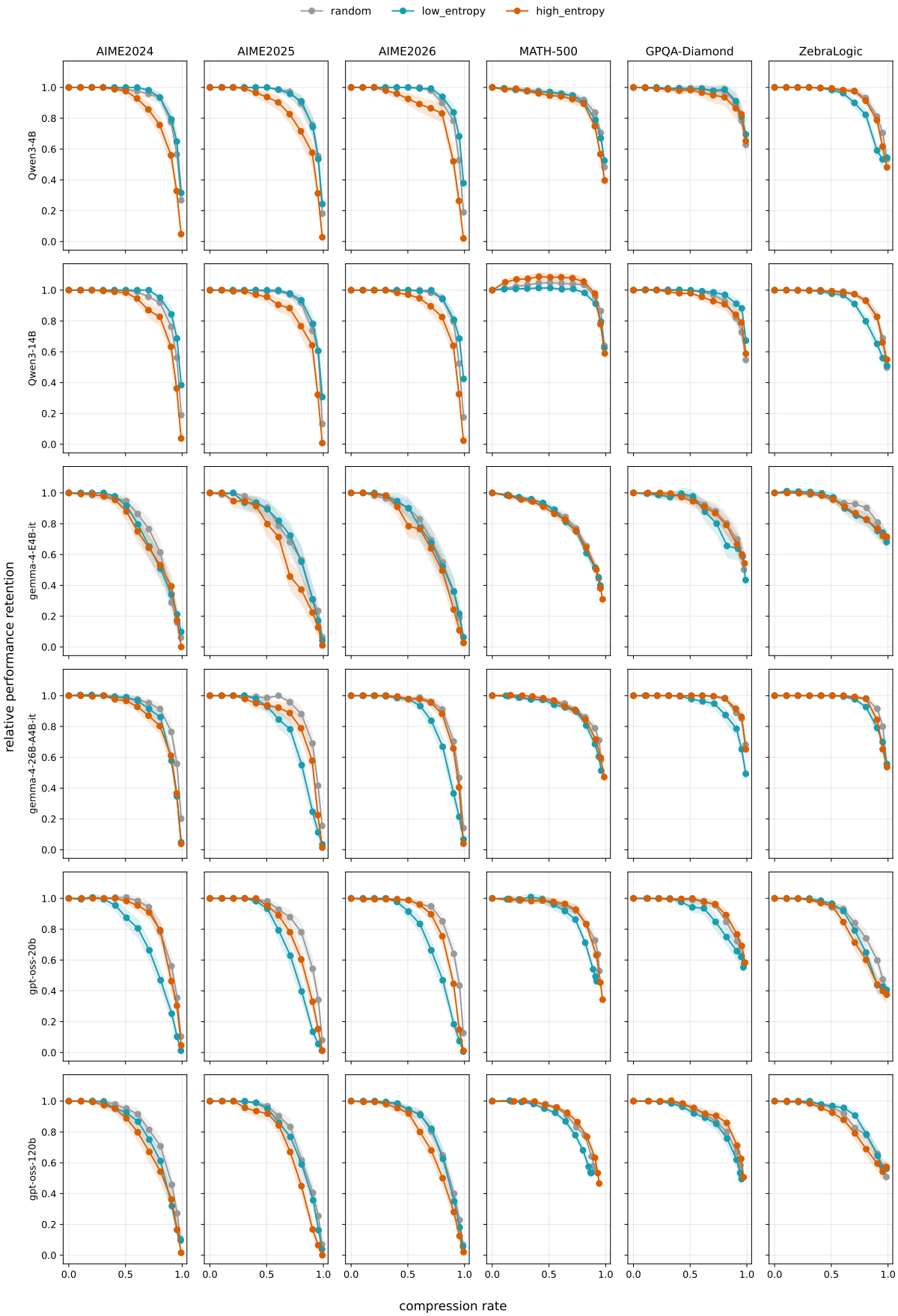

Figure 5: **Sentence-level** full grid: Relative Performance Retention vs. compression rate for all six models (rows) × six datasets (columns), selectors *random*, *low-entropy*, *high-entropy*. Shaded bands are 95% bootstrap CIs (Appendix C). *Random* is at or above both entropy-based selectors across nearly every panel, consistent with Table 5.

		Full range $r \in [0.01, 1.0]$					Aggressive compression $r \in [0.01, 0.3]$				
Model	Dataset	rand	Δlow	$\Delta\text{low}_{\text{numbers}}$	Δnumbers	Δhigh	random	Δlow	$\Delta\text{low}_{\text{numbers}}$	Δnumbers	Δhigh
Qwen3-4B	AIME24	0.52	**+0.32**	-0.52	-0.52	-0.39	0.10	**+0.39**	-0.10	-0.10	-0.08
	AIME25	0.49	**+0.33**	-0.49	-0.44	-0.36	0.05	**+0.38**	-0.05	-0.02	-0.05
	AIME26	0.51	**+0.36**	-0.51	-0.51	-0.40	0.05	**+0.52**	-0.05	-0.04	-0.05
	MATH-500	0.72	**+0.13**	-0.33	-0.40	-0.18	0.46	**+0.17**	-0.07	-0.13	-0.09
	GPQA-D	0.91	+0.04	+0.05	-0.33	-0.05	0.73	+0.11	+0.16	-0.15	-0.06
	Zebra	0.88	-0.01	-0.01	-0.34	-0.07	0.66	-0.07	-0.06	-0.12	-0.11
Qwen3-14B	AIME24	0.59	**+0.32**	-0.56	-0.58	-0.38	0.11	**+0.57**	-0.09	-0.10	-0.09
	AIME25	0.54	**+0.32**	-0.54	-0.54	-0.34	0.07	**+0.48**	-0.07	-0.07	-0.07
	AIME26	0.55	**+0.36**	-0.54	-0.53	-0.35	0.09	**+0.60**	-0.09	-0.08	-0.09
	MATH-500	0.88	**+0.07**	-0.32	-0.34	-0.11	0.67	**+0.17**	-0.12	-0.13	-0.08
	GPQA-D	0.91	**+0.05**	+0.05	-0.40	-0.06	0.74	+0.15	+0.16	-0.22	-0.08
	Zebra	0.90	-0.01	-0.01	-0.32	-0.07	0.71	-0.05	-0.06	-0.13	-0.11
gemma-4-E4B-it	AIME24	0.35	**+0.27**	-0.35	-0.35	-0.16	0.02	**+0.08**	-0.02	-0.01	-0.01
	AIME25	0.36	**+0.27**	-0.36	-0.34	-0.16	0.01	**+0.08**	-0.01	-0.00	-0.01
	AIME26	0.32	**+0.31**	-0.32	-0.32	-0.10	0.00	**+0.14**	-0.00	-0.00	-0.00
	MATH-500	0.57	**+0.12**	-0.29	-0.31	-0.07	0.29	**+0.05**	-0.02	-0.04	+0.01
	GPQA-D	0.83	**+0.01**	-0.09	-0.41	-0.02	0.58	**+0.02**	-0.14	-0.15	-0.01
	Zebra	0.91	-0.03	-0.05	-0.24	+0.00	0.78	-0.04	-0.04	-0.10	-0.02
gemma-4-26B-A4B-it	AIME24	0.53	-0.04	-0.04	+0.01	-0.05	0.14	-0.12	-0.14	-0.14	-0.08
	AIME25	0.43	-0.12	-0.14	+0.00	-0.06	0.02	-0.31	-0.35	+0.03	-0.11
	AIME26	0.43	-0.08	-0.09	+0.04	-0.01	0.03	-0.24	-0.26	+0.11	-0.04
	MATH-500	0.70	-0.03	-0.02	+0.00	-0.01	0.49	-0.07	-0.06	+0.03	-0.08
	GPQA-D	0.94	-0.01	-0.15	-0.22	-0.00	0.83	-0.02	-0.25	-0.10	-0.01
	Zebra	0.93	-0.02	-0.12	-0.33	-0.01	0.78	-0.05	-0.21	-0.19	-0.01
gpt-oss-20b	AIME24	0.78	-0.11	-0.78	-0.76	-0.30	0.36	-0.22	-0.36	-0.35	-0.33
	AIME25	0.72	-0.07	-0.72	-0.72	-0.28	0.29	-0.17	-0.29	-0.29	-0.28
	AIME26	0.77	-0.08	-0.77	-0.76	-0.27	0.39	-0.24	-0.39	-0.38	-0.37
	MATH-500	0.81	-0.04	-0.53	-0.53	-0.22	0.51	-0.09	-0.23	-0.24	-0.22
	GPQA-D	0.90	**+0.01**	-0.00	-0.35	-0.04	0.70	**+0.03**	+0.01	-0.15	-0.09
	Zebra	0.83	**+0.09**	+0.09	-0.45	-0.11	0.55	+0.20	+0.21	-0.17	-0.15
gpt-oss-120b	AIME24	0.76	-0.06	-0.75	-0.75	-0.35	0.34	-0.19	-0.34	-0.33	-0.30
	AIME25	0.71	+0.00	-0.71	-0.71	-0.35	0.26	-0.08	-0.26	-0.26	-0.25
	AIME26	0.73	+0.00	-0.73	-0.72	-0.33	0.27	-0.07	-0.27	-0.26	-0.25
	MATH-500	0.76	**+0.03**	-0.39	-0.38	-0.16	0.47	-0.03	-0.14	-0.10	-0.11
	GPQA-D	0.87	-0.00	-0.00	-0.43	-0.06	0.63	-0.01	+0.00	-0.21	-0.07
	Zebra	0.85	**+0.05**	+0.05	-0.38	-0.09	0.60	+0.09	+0.11	-0.13	-0.11

Table 6: AUC of *random* and delta of each **token-level** selector relative to it ($\Delta = \text{AUC}_{\text{selector}} - \text{AUC}_{\text{random}}$), **no patching**, full retention-rate range and aggressive-compression regime. **Bold** on Δlow: *low-entropy* strictly outperforms both *random* ($\Delta > 0$) and every other selector in that setting.

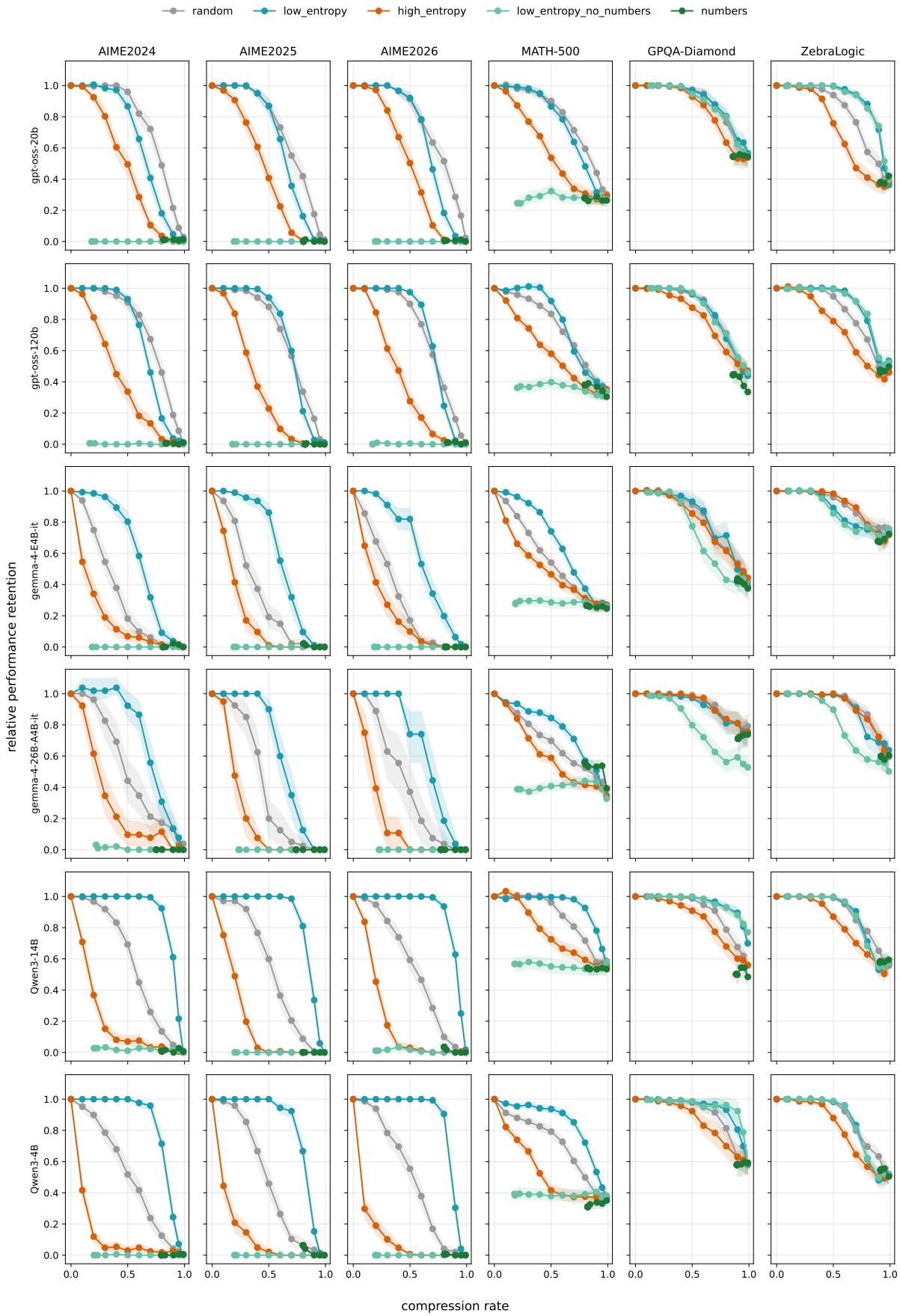

Figure 6: **Token-level full grid, no patching:** Relative Performance Retention vs. compression rate, selectors *random*, *low-entropy*, *high-entropy*, *low-entropy no numbers*, *numbers*. Model ordering: `gpt-oss-20b`, `gpt-oss-120b`, `gemma-4-E4B-it`, `gemma-4-26B-A4B-it`, `Qwen3-14B`, `Qwen3-4B`.

Model	Dataset	rand	Δlow	$\Delta\text{low}_\text{n}$	**Δnum**	Δhigh	Δnewl/Δeos
Qwen3-4B	AIME24	0.95	+0.03	-0.84	**+0.05**	-0.43	-0.89 / -0.92
	AIME25	0.95	+0.00	-0.91	**+0.04**	-0.52	-0.95 / -0.95
	AIME26	0.94	+0.04	-0.84	**+0.05**	-0.41	-0.94 / -0.94
	MATH-500	0.95	**-0.02**	-0.22	-0.08	-0.19	-0.56 / -0.55
	GPQA-D	0.98	**+0.00**	**+0.00**	-0.07	-0.06	-0.19 / -0.14
	Zebra	0.97	-0.03	**-0.02**	-0.41	-0.09	-0.39 / -0.29
Qwen3-14B	AIME24	0.95	**+0.04**	-0.64	**+0.04**	-0.41	-0.91 / -0.88
	AIME25	0.95	+0.03	-0.82	**+0.04**	-0.51	-0.95 / -0.95
	AIME26	0.95	+0.03	-0.68	**+0.04**	-0.42	-0.95 / -0.95
	MATH-500	1.02	**-0.03**	-0.14	-0.05	-0.11	-0.44 / -0.43
	GPQA-D	0.98	**+0.01**	**+0.01**	-0.14	-0.06	-0.15 / -0.12
	Zebra	0.97	**-0.01**	**-0.01**	-0.23	-0.09	-0.36 / -0.27
gemma-4-E4B-it	AIME24	0.84	+0.08	-0.72	**+0.12**	-0.45	-0.84 / -0.73
	AIME25	0.78	+0.09	-0.68	**+0.17**	-0.44	-0.78 / -0.67
	AIME26	0.78	+0.10	-0.74	**+0.18**	-0.41	-0.77 / -0.77
	MATH-500	0.87	**+0.04**	-0.36	-0.04	-0.20	-0.54 / -0.44
	GPQA-D	0.97	**-0.00**	-0.04	-0.08	-0.08	-0.11 / -0.25
	Zebra	0.96	**-0.01**	-0.03	-0.13	-0.02	-0.24 / -0.07
gemma-4-26B-A4B-it	AIME24	0.96	+0.05	-0.28	**+0.07**	-0.15	-0.96 / -0.96
	AIME25	0.93	+0.02	-0.54	**+0.05**	-0.25	-0.93 / -0.93
	AIME26	0.94	+0.03	-0.39	**+0.04**	-0.32	-0.94 / -0.94
	MATH-500	0.94	+0.02	-0.08	**+0.03**	-0.07	-0.38 / -0.48
	GPQA-D	0.99	-0.00	-0.01	**+0.00**	-0.02	-0.05 / -0.08
	Zebra	0.98	**+0.01**	+0.00	-0.12	-0.03	-0.06 / -0.18
gpt-oss-20b	AIME24	0.88	+0.08	-0.70	**+0.11**	-0.36	-0.88 / -0.88
	AIME25	0.84	+0.11	-0.64	**+0.13**	-0.37	-0.84 / -0.84
	AIME26	0.87	+0.09	-0.66	**+0.10**	-0.36	-0.85 / -0.86
	MATH-500	0.91	**+0.05**	-0.44	+0.04	-0.22	-0.65 / -0.55
	GPQA-D	0.96	**+0.01**	-0.01	-0.06	-0.04	-0.30 / +0.01
	Zebra	0.91	**+0.04**	**+0.04**	-0.43	-0.11	-0.36 / -0.20
gpt-oss-120b	AIME24	0.87	+0.08	-0.28	**+0.11**	-0.22	-0.86 / -0.86
	AIME25	0.85	+0.10	-0.18	**+0.12**	-0.29	-0.83 / -0.83
	AIME26	0.86	+0.08	-0.15	**+0.12**	-0.28	-0.86 / -0.86
	MATH-500	0.93	+0.07	-0.11	**+0.10**	-0.17	-0.62 / -0.49
	GPQA-D	0.95	**+0.01**	**+0.01**	-0.11	-0.01	-0.15 / +0.00
	Zebra	0.93	+0.03	**+0.04**	-0.28	-0.09	-0.27 / -0.17

Table 7: AUC of *random* and delta of each **token-level** selector relative to it ($\Delta = \text{AUC}_{\text{selector}} - \text{AUC}_{\text{random}}$), **patched**, full retention-rate range. Last column: newlines / end-of-sentence deltas. **Bold** marks the best-performing selector (highest resulting AUC) per row.

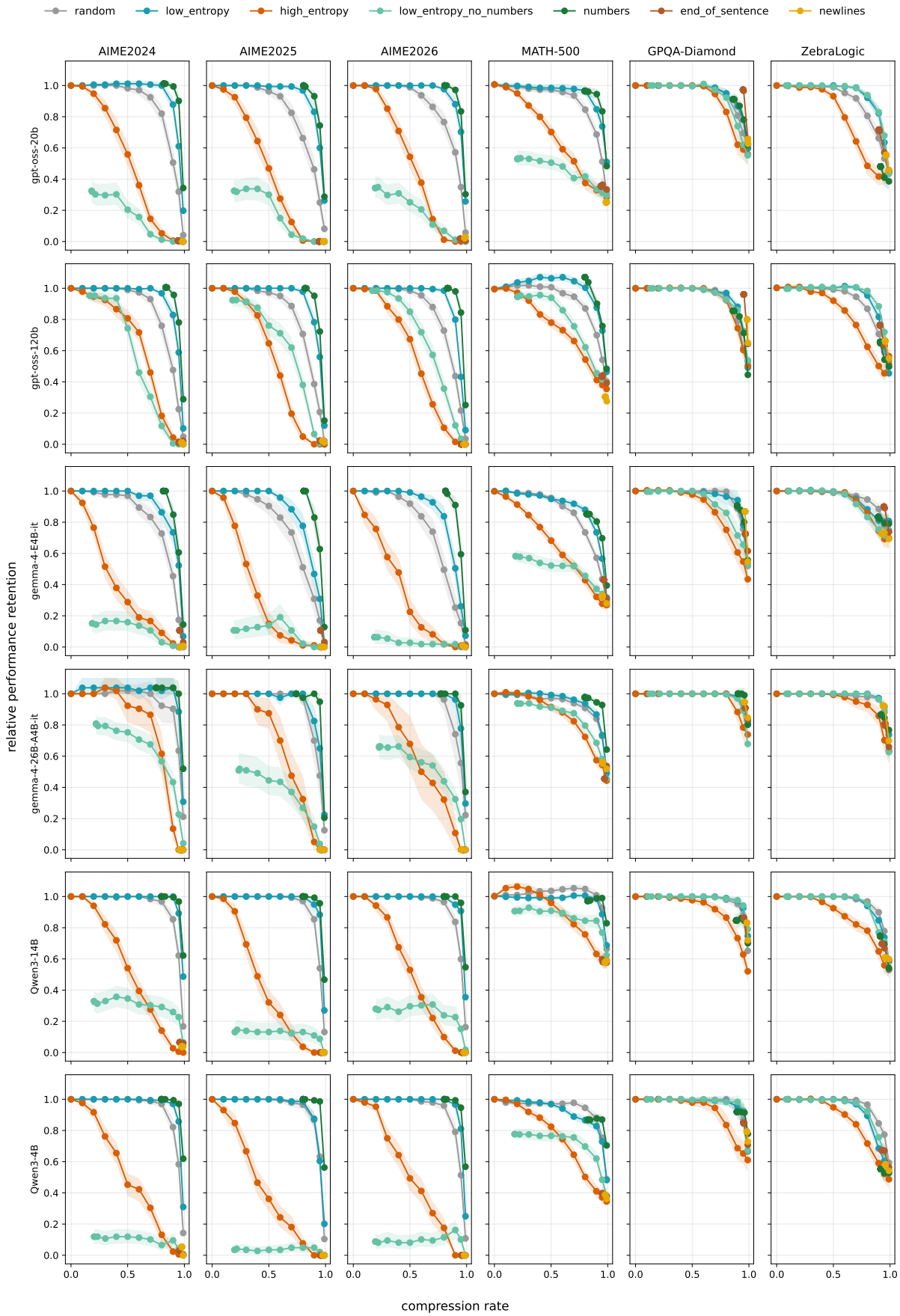

Figure 7: **Token-level patched full grid:** Relative Performance Retention vs. compression rate, selectors *random*, *low-entropy*, *high-entropy*, *low-entropy no numbers*, *numbers*. Model ordering: `gpt-oss-20b`, `gpt-oss-120b`, `gemma-4-E4B-it`, `gemma-4-26B-A4B-it`, `Qwen3-14B`, `Qwen3-4B`.

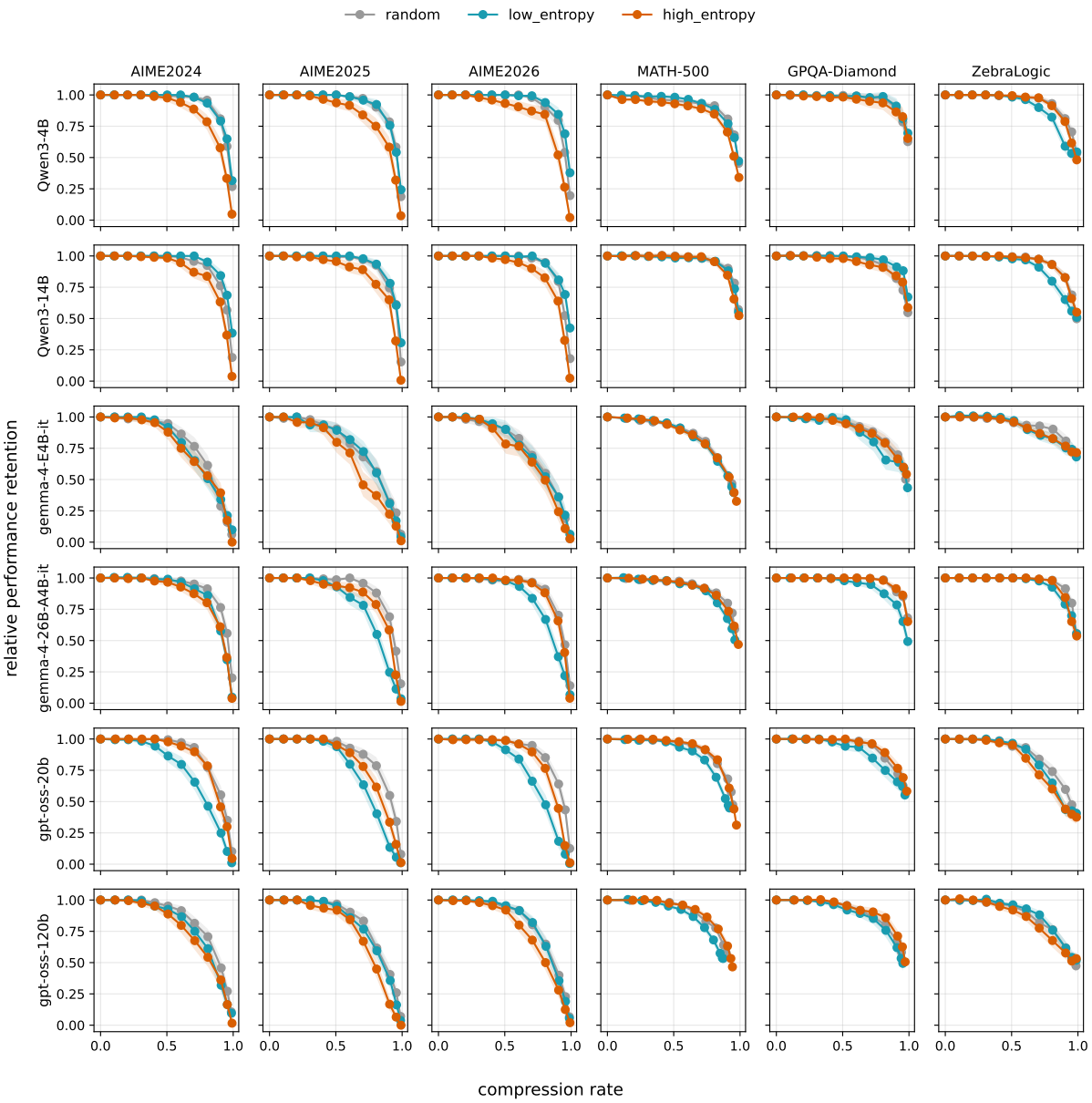

Figure 8: Sentence-level full grid under `math-verify`: Relative Performance Retention vs. compression rate for all six models (rows) $\times$ six datasets (columns), selectors *random*, *low-entropy*, *high-entropy*. The pattern matches Figure 5 (exact-match): *random* is at or above both entropy-based selectors in the large majority of panels, confirming that the sentence-level results are not an artifact of the matching criterion.

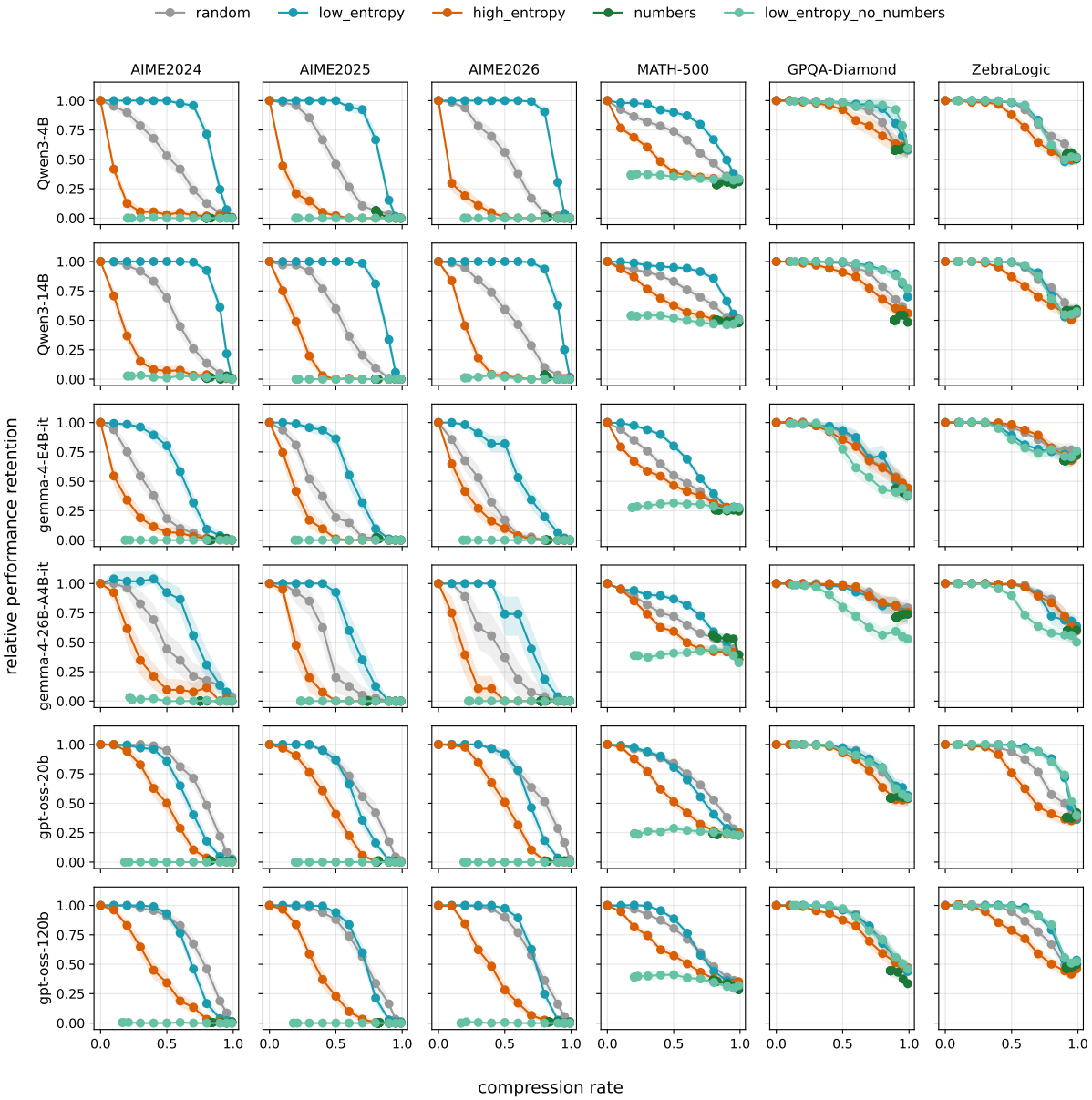

Figure 9: Token-level full grid under `math-verify` (no patching): Relative Performance Retention vs. compression rate for all six models (rows) × six datasets (columns), selectors *random*, *low-entropy*, *high-entropy*. The pattern matches Figure 6 (exact-match): *low-entropy* exceeds *random* on mathematical benchmarks only.