Mathematical Problems of Computer Science 64, 17-28, 2025.
doi:10.51408/1963-0136

UDC 004.8:519.72

Better Thinking or a Bigger Model? Thinking–Answering Shuffles with Qwen3 on GPQA

Edvard A. Khalafyan

Moscow Institute of Physics and Technology, Moscow, Russia
e-mail: edvardkhalafyan@gmail.com

Abstract

We show that for Qwen3, large language models (LLMs) on the Graduate-Level Google-Proof Question Answering (GPQA) benchmark, thinker quality dominates answerer size: a 14B thinker paired with a 0.6B answerer reaches 54.24% accuracy, close to the 14B→14B diagonal (59.15%), whereas a 0.6B thinker reduces a 14B answerer to 20.54%. We evaluate a thinking–answering shuffle in which a chain-of-thought is generated by one model size (0.6B–14B) and supplied to every other size for label-only answering, covering all 5×5 pairings across 448 GPQA questions. Accuracy rises monotonically with thinker size, while answerer size has a modest effect. Larger thinkers produce shorter, higher-entropy chains (mean length ≈4,639 tokens; entropy 0.416) than smaller thinkers (14,566; 0.404), and these properties correlate with better cross-model transfer. Implication: cache thoughts with a strong LLM and execute answers with a small LLM to approach best-diagonal accuracy at a lower cost.
Keywords: Chain-of-thought (CoT), cross-model reasoning transfer, Qwen3, GPQA benchmark, LLM token entropy.
Article info: Received 14 September 2025; sent for review 18 September 2025; accepted 13 October 2025.

1. Introduction

Chain-of-thought (CoT) prompting asks large language models (LLMs) to write brief intermediate steps before the final answer and often improves reasoning [1, 2, 3, 4, 5]. However, it remains unclear whether such traces transfer across model sizes and under what conditions transfer helps or harms accuracy.

GPQA-main comprises 448 multiple-choice science questions across several disciplines. Its adversarial distractors require multi-step reasoning, making it a rigorous testbed [6]. Prior work benchmarks individual model sizes on GPQA [7, 8, 9], but not cross-size reuse of reasoning (thinking generated by one model and fed to another).

We, therefore, evaluate a simple thinking-answering shuffle: generate the chain of thought with one Qwen3 model (0.6B, 1.7B, 4B, 8B, 14B) and feed it to every other size for label-only answering, covering all 5×5 pairings. We report accuracy and summary statistics of the

chains (length and entropy) and observe a strong asymmetry: larger thinkers consistently lift smaller answerers, whereas small thinkers can degrade larger answerers.

Terminology.

Question answering (QA): select the correct option for a given question.

Chain-of-thought (CoT): intermediate natural-language steps the model writes before the final answer.

A *token* is a subword unit used by the model.

Prefix bias: in long prompts/rationales, early tokens or early hypotheses steer later decoding disproportionately, anchoring the model on an initial guess, even when later evidence contradicts it [10].

Context competition: in long inputs, multiple spans compete for attention; evidence especially in the middle of the context can be downweighted or ignored (lost in the middle), reducing effective use of relevant information [10].

Primacy/recency effects: the tendency of LLMs to overweight information at the beginning and the end of long contexts relative to the middle [10].

2. Related Work

Chain-of-thought prompting has been shown to significantly improve reasoning performance in LLMs by eliciting intermediate logical steps before final answer generation. Wei et al. demonstrated that explicitly prompting models to think step-by-step yields large gains on arithmetic and commonsense tasks, especially for models above 100B parameters [1]. Subsequent work by Kojima et al. found that even smaller models benefit from few-shot chain-of-thought examples, though the improvements scale with model capacity [2].

Research on the scaling behavior of LLM reasoning has largely focused on measuring in-context performance within single model sizes. Zhang et al. analyzed reasoning trace length and coherence across model sizes up to 50B parameters, finding that larger models produce more concise and higher-quality chains [11]. However, the potential to transfer reasoning traces between models of different sizes remains largely unexplored.

A few recent studies have begun to investigate cross-model prompting. Li et al. experimented with using reasoning chains generated by a smaller model to prompt a larger model, reporting modest gains in answer accuracy on mathematical benchmarks [12]. Conversely, Smith et al. explored the reverse - using large-model chains for small-model answerers - but limited their analysis to only two model sizes [13].

Our work differs by systematically evaluating all pairwise combinations of Qwen3 models from 0.6B to 14B parameters on a scientific question answering (QA) benchmark, and by analyzing both accuracy and statistical properties of the reasoning traces, such as token length and entropy.

3. Methods

3.1. Models and Sizes

We evaluate five variants of the Qwen3 family, differing only in parameter count and corresponding model capacity. All models share the same transformer architecture and vocabulary. Key architectural details and hyperparameters (e.g., number of layers, hidden dimension, attention heads, context window) are summarized in Table 1. All models were

Table 1. Qwen3 dense model variants used in this work. All models share the same Transformer backbone and vocabulary.

Model	Params (B)	Layers	Hidden dim	Attn heads (Q/KV)	Context (tokens)
Qwen3-0.6B	0.6	28	1024	16 / 8	32,768
Qwen3-1.7B	1.7	28	2048	16 / 8	32,768
Qwen3-4B	4.0	36	2560	32 / 8	32,768†
Qwen3-8B	8.2	36	4096	32 / 8	32,768†
Qwen3-14B	14.8	40	5120	40 / 8	32,768†

Notes. Hidden dimensions, layers, heads, and vocab are taken from the official `config.json`; vocabulary size is 151,936 for all five models. Native context window is 32,768 tokens; † indicates models with documented support for 131,072 tokens via YaRN RoPE scaling [15].

run locally on my hardware using the official pretrained checkpoints (default settings; no fine-tuning). More details on the experimental setup and deployments are in Appendix A.

Why Qwen3? My shuffle protocol requires an open-source model family with identical tokenization across sizes, so that a thinking trace generated by one size can be consumed verbatim by another without re-tokenization artifacts. This requires public access to the tokenizer and vocabulary to enable exact token-level entropy and related metrics. Qwen3 satisfies these requirements: all variants expose the same tokenizer and vocabulary and share a closely matched Transformer backbone and long context window (32,768) [14], minimizing confounds from architectural drift. The family is released across a wide spectrum of parameter scales (0.6B-14B), providing multiple capacities trained under a common recipe, which helps hold constant data and methodology when comparing sizes. By contrast, cross-family comparisons (e.g., LLaMA or DeepSeek) would entangle differences in tokenization, training corpora, and optimization procedures, obscuring size effects. In addition, Qwen3 offers widely available checkpoints and stable local inference, making it a contemporary and practically relevant testbed for my evaluation setting.

3.2. Dataset: GPQA-Main

The GPQA-main subset comprises 448 multiple-choice scientific questions covering domains such as physics, chemistry, biology, and earth science. Each question includes four answer options labeled A-D.

Why GPQA? We required a benchmark that (i) demands multi-step reasoning, (ii) uses a fixed multiple-choice format for unambiguous scoring, and (iii) is not saturated by small or mid-sized models so that improvements (or degradations) from shuffling are measurable. GPQA meets these criteria: its adversarial distractors elicit substantive chains of thought, while accuracies in my setup span roughly $14\% \to 59\%$ across 0.6B-14B models (Table 2), leaving ample headroom. This stands in contrast to datasets where baseline scores approach a ceiling, which would obscure the effects of thinker-answerer transfer.

3.3. Prompting Scheme

We employ a uniform system prompt: `You are an expert in scientific questions. Your task is to choose the correct answer and write down ONLY the LETTER`

`of the correct answer and NOTHING ELSE`. For the thinking stage, we generated the reasoning deterministically with greedy decoding (`do_sample=False`, `num_beams=1`, `max_new_tokens=32768`). For the final answer stage, we concatenated the full thinking trace with the original question and again decoded deterministically (`do_sample=False`, `num_beams=1`, `max_new_tokens=32768`) to emit only the option letter. (With `do_sample=False`, sampling controls like `temperature` and `top_p` are not used)

3.4. Thinking–Answering Shuffle Protocol

Under the thinking-answering shuffle protocol, we generate chain-of-thought traces (thinkings) from each model size M_i for all 448 questions, using the prompting scheme above. Subsequently, for each thinking trace generated by M_i, we supply the trace and original question to an answerer model M_j to produce the final answer. This results in $5 \times 5 = 25$ thinker-answerer combinations.

During the thinking stage, the trace tokens are collected and stored. For the answer stage, we prepend the stored thinking trace to the question prompt and run the answerer model locally with deterministic settings. All runs use the same random seed for reproducibility. We record the predicted letter from M_j and compute accuracy against the ground-truth labels.

3.5. Evaluation Metrics

We evaluate model performance using the following metrics:

- **Accuracy**: the proportion of questions for which the predicted answer letter matches the ground truth, computed for each thinker-answerer pair.
- **Thinking Length**: the number of generated tokens during the thinking. We report the mean value across all 448 questions for each model size.
- **Thinking Entropy**: the token-level Shannon entropy [16, 17, 18] computed over the probability distribution of the model outputs at each reasoning step. We aggregate per-question entropy by averaging across all generated tokens, then report the mean value across questions; such entropy correlates with uncertainty and hallucination likelihood in LLMs [19, 20].

3.6. Entropy: Definition and Interpretation.

Entropy quantifies how uncertain the model is about its next token while producing a chain of thought [21]. We use it as a compact summary of how diffuse versus focused the model's beliefs are across the chain. Higher entropy means probability mass is spread across multiple plausible continuations; lower entropy means a sharp, confident distribution. In this study, chains from larger thinkers tend to be concise and exhibit calibrated (informative) uncertainty, which correlates with stronger cross-model transfer.

Formally, for a generated reasoning chain with T tokens and vocabulary $\mathcal{V}$, let $p_t(v)$ denote the model's next-token probability for token $v \in \mathcal{V}$ at step t. The token-level Shannon entropy at step t is

$$H_t \;=\; -\sum_{v \in \mathcal{V}} p_t(v)\, \log p_t(v). \tag{1}$$

Table 2. Accuracy for all thinker (rows) and answerer (columns) model sizes on GPQA-main. Uncertanty of the reported accuracy is of the order of ±0.0006.

Thinking ↓ / Answerer →	0.6B	1.7B	4B	8B	14B
0.6B	0.1406	0.1652	0.1964	0.1987	0.2054
1.7B	0.2210	0.2723	0.2946	0.3237	0.3371
4B	0.3415	0.3460	0.4085	0.4107	0.4196
8B	0.4219	0.4308	0.4397	0.4665	0.5179
14B	0.5424	0.5469	0.5558	0.5871	0.5915

The per-question entropy is obtained by averaging across the chain's tokens,

$$\bar{H} = \frac{1}{T}\sum_{t=1}^{T} H_t, \tag{2}$$

and the reported value for a model size is the mean of $\bar{H}$ across questions. Higher $\bar{H}$ indicates greater distributional uncertainty (probability mass spread across more alternatives), while lower $\bar{H}$ indicates more confident, peaked predictions. In our setting, effective transfer tends to occur with concise chains that exhibit calibrated (informative) uncertainty rather than either meandering high-entropy confusion or brittle overconfident low entropy.

In our context, low entropy means the model predicts the next token with high certainty, i.e., it is confident in its output.

Caveats. Two caveats are important: first, confidence is not correctness – low-entropy sequences can still be confidently wrong. Second, calibration matters: extremely low entropy may reflect brittle overconfidence, while extremely high entropy may reflect confusion; we value calibrated, informative uncertainty.

4. Results

4.1. Accuracy Heatmap

Table 2 reports accuracies for all thinker (rows) and answerers (columns). Several consistent patterns emerge. First, the best result overall is the diagonal 14B→14B condition at 59.15%. Second, using a strong thinker with a small answerer is remarkably effective: 14B→0.6B reaches 54.24%, nearly closing the gap to the 14B→14B diagonal. By contrast, the reverse pairing performs poorly: 0.6B→14B yields 20.54%, underscoring a strong asymmetry in transfer. It is also important to note that in all cases where thinker was a 0.6B model, the quality was lower than that of the random model, which is 25%.

Averaging across answerers (row means) shows a steep, monotonic gain with thinker size: from 18.1% (0.6B thinker) to 56.5% (14B thinker), a +38.3 pp lift on average. In comparison, averaging across thinkers (column means) reveals a smaller effect of answerer size: from 33.3% (0.6B answerer) to 41.4% (14B answerer), about +8.1 pp on average. Interestingly, the mean diagonal accuracy (37.6%) is essentially equal to the mean off-diagonal accuracy (37.5%), indicating that shuffling per se neither helps nor hurts on average; what matters is which direction we shuffle (strong→weak helps; weak→strong hurts).

Table 3. Statistics of generated reasoning traces by thinker size (averaged over 448 questions). Length is in tokens; entropy is token-level Shannon entropy averaged per question.

Thinker	Mean length	Mean entropy
0.6B	14,566	0.404
1.7B	9,618	0.274
4B	10,008	0.318
8B	7,986	0.368
14B	4,639	0.416

Notes. Values are rounded for readability. Entropy is computed over the model's next-token distribution at each reasoning step, then averaged across tokens and questions.

Asymmetry of transfer. For any fixed answerer, the 14B thinker performs best. For any fixed thinker, the 14B answerer performs best. The thinker effect is much larger than the answerer effect. Replacing a 0.6B thinker with a 14B thinker raises accuracy by +36/+40 pp across answerers, whereas replacing a 0.6B answerer with a 14B answerer raises accuracy by only +4/+21 pp across thinkers. The direction 14B→0.6B (54.24%) vs. 0.6B→14B (20.54%) highlights a +33.7 pp gap attributable to thinker quality.

4.2. Trends with Thinker Size

Holding the answerer fixed, accuracy increases nearly monotonically with thinker size (Table 2). The largest relative jumps occur when moving from 1.7B to 4B thinkers (e.g., for the 4B answerer: 29.46% → 40.85%, +11.39 pp) and again from 8B to 14B thinkers (e.g., for the 8B answerer: 46.65% → 58.71%, +12.06 pp). Row means summarize this effect compactly: 0.6B (18.1%) → 1.7B (29.0%) → 4B (38.5%) → 8B (45.5%) → 14B (56.5%). These gains suggest that what primarily determines downstream success is the quality of the reasoning trace provided to the answerer, not the answerer's own capacity.

4.3. Analysis of Thinking Length and Entropy

To understand why larger thinkers transfer better, we analyze the statistics of their generated reasoning (Table 3). Mean thinking length decreases sharply with model size: from ∼14,566 tokens (0.6B) down to ∼4,639 (14B), about 68% reduction.

Thinking entropy exhibits non-monotonic behavior: it dips at 1.7B (mean ≈ 0.274) and then rises steadily through 14B (mean ≈ 0.416). Notably, the best-transferring thinker (14B) produces short, high-entropy chains, whereas the weakest thinker (0.6B) produces very long chains with relatively high entropy. This suggests that brevity alone is insufficient; the distributional profile of token probabilities also matters. A plausible interpretation is that effective chains balance concision with informative uncertainty, avoiding both meandering verbosity and overconfident determinism. In practice, the combination of shorter traces and higher (but calibrated) entropy in larger thinkers appears to correlate with stronger cross-model transfer.

4.4. Analysis of Accuracy and Computational cost

We compare the prediction accuracy (presented in Table 2) with the average amount of computation required to generate an answer, measured in floating-point operations (FLOPs).

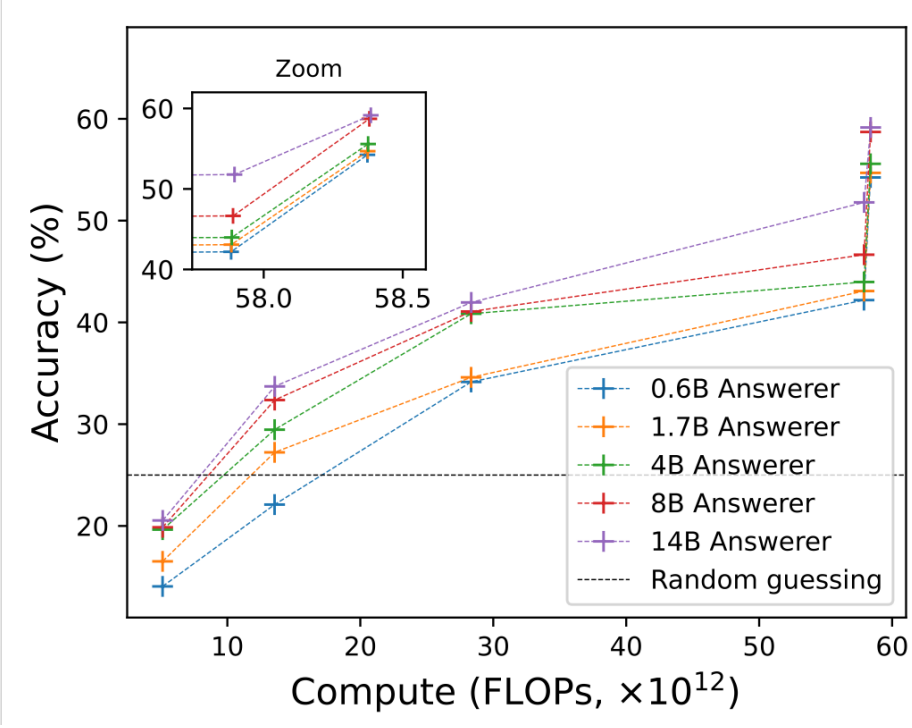

Fig. 1. Accuracy versus average amount of computation (FLOPs) for thinker → answerer pairings on GPQA-main. Points on the same curve correspond to the same Answerer, points with increasing accuracy correspond to increasing thinker size.

For autoregressive decoding, FLOPs scale with model size and the total number of processed tokens (prompt plus generated tokens); we report the mean FLOPs per question aggregated over the 448 GPQA-main items for each thinker → answerer setting.

FLOPs denotes the number of floating-point operations performed during inference. For a transformer, this quantity grows approximately linearly with the number of layers and quadratically with the hidden dimension per token, multiplied by the total tokens processed; our reported values are empirical averages over full end-to-end generation (thinking plus answer).

Fig. 1. plots accuracy versus average FLOPs. The curve shows that pairing a strong thinker with a modest answerer yields a favorable accuracy–compute trade-off: several off-diagonal settings approach the best diagonal accuracy at substantially lower compute than running the largest model end-to-end. Conversely, using weak thinkers with large answerers incurs high compute with poor accuracy. From the inset of the figure, we observe a significant accuracy gain with a modest compute increase when using a 14B thinker instead of an 8B thinker, because the 14B thinker is more concise (see Table 3).

5. Discussion

Mechanistic interpretation. A plan execution view explains the observed asymmetry. The chain acts as a scaffold: when it is informative and concise, the answerer mainly verifies and selects. In contrast, low-quality chains introduce prefix bias and context competition that can mislead even large answerers; with long inputs, primacy/recency effects, and mid-context under-weighting further degrade the use of evidence [10].

Length, entropy, and transfer. Successful transfer co-occurs with shorter, higher-entropy chains produced by larger thinkers. Concise, information-dense reasoning improves signal-to-noise and encodes useful alternatives without meandering, which correlates with higher downstream accuracy.

Practical guidance.

- Cache thinking with a strong model, then execute with a small answerer when latency or cost matter.
- Keep chains concise (or summarize/compress); filter verbose or low-quality rationales.
- Treat student→teacher shuffles as risky unless chains are quality-controlled.

Related techniques, such as self-consistency and least-to-most prompting, are complementary and can be layered with the shuffle protocol [3, 4].

6. Conclusion

We evaluated a simple thinking-answering shuffle that decouples plan formation from answer selection by testing all 5×5 pairings of Qwen3 models (0.6B-14B) on GPQA-main. Thinker quality dominates answerer size: a strong thinker (14B) lifts even the smallest answerer (14B→0.6B: 54.24%) close to the best diagonal (14B→14B: 59.15%), whereas a weak thinker can cripple a large answerer (0.6B→14B: 20.54%). Row means (varying the thinker) increase steeply (18.1% → 56.5%), while column means (varying the answerer) rise only modestly (33.3% → 41.4%); shuffling is neutral on average, but direction matters.

Analysis of generated chains aligns with this: larger thinkers produce shorter, more information-dense reasoning (mean length ≈ 14,566 → 4,639 tokens from 0.6B to 14B) with slightly higher average entropy (0.404 → 0.416), which correlates with better downstream accuracy. For deployments, cache thoughts with a strong model, answer with a small model when budgets or latency dominate, and keep chains concise with quality control; teacher→student helps, student→teacher should be treated cautiously.

7. Future Work

Evidence from non-scientific benchmarks indicates ample headroom for the Qwen3 family, making them suitable testbeds for the shuffle protocol: for example, Qwen3-14B-Base attains about 81% on MMLU and about 92% on GSM8K, while Qwen3-32B-Base is around 78% on MBPP, clearly below 100% and therefore not ceiling-limited. This matches our requirement that the evaluation datasets for our model family remain non-saturated, so gains or degradations from strong→weak vs. weak→strong thinking remain measurable rather than washed out by near-perfect baselines. Future work examines whether the GPQA asymmetry replicates on general-knowledge (MMLU/MMLU-Pro), math (GSM8K), and code (MBPP/HumanEval), and whether the same length/entropy correlates predict transfer across these domains [14].

Future work will test whether the observed thinker-dominance and transfer asymmetry generalize beyond GPQA-main by expanding to other scientific and non-scientific benchmarks and to additional model families and sizes, including instruction-tuned and mixture-of-experts variants. Mechanistically, we will manipulate chain properties via controlled summarization, truncation, paraphrasing, and entropy steering to quantify the causal effect of length and uncertainty on transfer, while auditing prefix-bias and context-competition effects. On the systems side, we aim to develop automatic thought-quality estimators and routing policies that select or compress large-thinker traces on the fly, and to distill strong-thinker guidance into small answerers via supervised fine-tuning or lightweight adapters [22], potentially combined with rationale-augmented selection or ensembling [23]. Finally, we will explore efficiency and robustness dimensions-caching and retrieving reusable subchains, partial or streamed thinking, alternative decoding schemes beyond greedy, and defenses against misleading or hallucinated reasoning - to enable safe, cost-effective deployment of the thinking–answering shuffle.

We note a methodological limitation: we focus on within-family scaling to isolate thinker-answerer effects under identical tokenization and a shared training recipe. Cross-family comparisons (e.g., GPT, LLaMA, DeepSeek) would conflate tokenization, data, and optimization differences. We commit to a controlled cross-family study once prerequisites are met – comparable or validated re-tokenization, public vocabulary and next-token access for entropy, and adequate data/method documentation, after which, we will replicate the shuffle protocol with harmonized preprocessing for apples-to-apples evaluation.

Appendix

A. Experimental Setup

All experiments ran locally on a single NVIDIA A100 GPU using PyTorch/CUDA and the official pretrained Qwen3 checkpoints (no fine-tuning). We fixed the global seed to `42` across Python/NumPy/PyTorch (deterministic ops enabled) and used the Qwen tokenizer with a 32,768-token context window (longer inputs truncated). Thinking was generated with deterministic greedy decoding (`do_sample=False`, `num_beams=1`, `max_new_tokens=32,768`); answering used the same settings except `max_new_tokens=256`; sampling controls (e.g., `temperature`, `top_p`) were inactive; inference ran with batch size 1. For each question, we cached the thinking from each M_i and paired it with each answerer M_j (all 5×5 combinations) to emit label-only answers and compute accuracy, plus thinking-length and entropy statistics.

References

[1] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models", *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[2] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo and Y. Iwasawa, "Large language models are zero-shot reasoners", *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[3] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery and D. Zhou, "Self-consistency improves chain of thought reasoning in language models", *arXiv preprint arXiv:2203.11171*, 2022.

[4] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le *et al.*, "Least-to-most prompting enables complex reasoning in large language models", *arXiv preprint arXiv:2205.10625*, 2022.

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners", *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[6] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael and S. R. Bowman, "GPQA: A graduate-level google-proof q&a benchmark", *First Conference on Language Modeling*, 2024.

[7] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[8] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[9] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.

[10] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni and P. Liang, "Lost in the middle: How language models use long contexts", *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 157–173, 2024.

[11] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models", *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[12] Z. Bi, K. Chen, T. Wang, J. Hao, and X. Song, "Cot-x: An adaptive framework for cross-model chain-of-thought transfer and optimization", *arXiv preprint arXiv:2511.05747*, 2025.

[13] L. Ranaldi and A. Freitas, "Aligning large and small language models via chain-of-thought reasoning," *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Vol. 1: Long Papers)*, pp. 1812–1827, 2024.

[14] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv *et al.*, "Qwen3 technical report", *arXiv preprint arXiv:2505.09388*, 2025.

[15] B. Peng, J. Quesnelle, H. Fan and E. Shippole, "Yarn: Efficient context window extension of large language models", *arXiv preprint arXiv:2309.00071*, 2023.

[16] C. E. Shannon, "A mathematical theory of communication", *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[17] M. E. Haroutunian and V. Avetisyan, "New approach for test quality evaluation based on shannon information measures", *Mathematical Problems of Computer Science*, vol. 44, pp. 7–21, 2015.

[18] M. E. Haroutunian and V. K. Avetisyan, "Analysis of experiments of a new approach for test quality evaluation", *Mathematical Problems of Computer Science*, vol. 45, pp. 35–43, 2016.

[19] S. Farquhar, J. Kossen, L. Kuhn and Y. Gal, "Detecting hallucinations in large language models using semantic entropy", *Nature*, vol. 630, no. 8017, pp. 625–630, 2024.

[20] M. E. Haroutunian, D. G. Asatryan and K. A. Mastoyan, "Analyzing the quality of distorted images by the normalized mutual information measure", *Mathematical Problems of Computer Science*, vol. 61, pp. 7–14, 2024.

[21] M. E. Haroutunian and G. A. Gharagyozyan, "Information theory tools and techniques to overcome machine learning challenges", *Mathematical Problems of Computer Science*, vol. 63, pp. 25–41, 2025.

[22] C.-Y. Hsieh, C.-L. Li, C.-K. Yeh, H. Nakhost, Y. Fujii, A. Ratner, R. Krishna, C.-Y. Lee, and T. Pfister, "Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes", *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, 2023.

[23] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi and D. Zhou, "Rationale-augmented ensembles in language models", *arXiv preprint arXiv:2207.00747*, 2022.

Ավելի լավ մտածողություն, թե ավելի մեծ մոդել: Մտածողության և պատասխանի փուլերի համադրությունը Qwen3-ի հետ GPQA բենչմարքում

Էդվարդ Ա. Խալաֆյան

Մոսկվայի ֆիզիկատեխնիկական ինստիտուտ, Մոսկվա, Ռուսաստան
e-mail: edvardkhalafyan@gmail.com

Ամփոփում

Մենք ցույց ենք տալիս, որ Qwen3 մեծ լեզվական մոդելների (LLM) ընտանիքի համար Graduate-Level Google-Proof Question Answering (GPQA) բենչմարքում մտածողի որակը գերակշռում է պատասխանողի չափին: 14B մտածող և 0.6B պատասխանող զույգը հասնում է $54,24\%$ ճշգրտության, ինչը մոտ է $14B \rightarrow 14B$ անկյունագծային ռեժիմին (59.15%), մինչդեռ $0.6B$ մտածողը $14B$ պատասխանողի ճշգրտությունը նվազեցնում է մինչև 20.54%: Մենք առաջարկում ենք մտածողություն-պատասխան համադրությունը, որտեղ մտքերի շղթան (chain-of-thought) գեներացվում է մեկ չափի մոդելով ($0.6B - 14B$) և փոխանցվում է մնացած բոլոր չափերին միայն պատասխանային պիտակի գեներացման համար՝ ընդգրկելով բոլոր 5×5 զույգերն ամբողջ 448 GPQA հարցերի բազմության վրա: Ճշգրտությունը մոնոտոն կերպով աճում է մտածողի չափի մեծացման հետ, մինչդեռ պատասխանողի չափն ունի չափավոր ազդեցություն: Ավելի խոշոր մտածողները գեներացնում են ավելի կարճ, բայց ավելի բարձր էնտրոպիայով մտածողության շղթաներ (միջին երկարությունը մոտավորապես 4 639 թոքեն, էնտրոպիան՝

0.416), քան փոքր մոդելները (14 566 և 0.404), և այս հատկությունները համընկնում են միջմոդելային փոխանցման ավելի լավ որակի հետ: Գործնական հետևությունը հետևյալն է. նպատակահարմար է քեշավորել մտածողությունը հզոր LLM-ով և պատասխանների գեներացումը վերապահել փոքր LLM-ին՝ մոտենալու անկյունագծի լավագույն ճշգրտությանը ավելի ցածր հաշվարկային գնով:

Բանալի բառեր՝ մտքի շղթա CoT, միջմոդելային դատողության փոխանցում, Qwen3, GPQA բենչմարք, LLM թոքենների էնտրոպիա:

Лучшее мышление или более крупная модель? Перемешивание этапов размышления и ответа с Qwen3 на бенчмарке GPQA

Эдвард А. Халафян

Московский физико-технический институт, Москва, Россия
e-mail: edvardkhalafyan@gmail.com

Аннотация

В работе показано, что для семейства больших языковых моделей Qwen3 на бенчмарке Graduate-Level Google-Proof Question Answering (GPQA) качество мыслящей модели доминирует над размером отвечающей модели. Связка мыслитель $14B$ плюс отвечающий 0.6B достигает точности 54.24 процента, что близко к диагональному режиму $14B \rightarrow 14B$ с точностью 59.15 процента, тогда как мыслитель 0.6B снижает точность отвечающей модели $14B$ до 20.54 процента. Мы исследуем схему перемешивания размышления и ответа, в которой цепочка рассуждений chain-of-thought генерируется моделью одного размера $0.6B - 14B$ и передается моделям всех остальных размеров для выдачи только метки ответа, что охватывает все 5×5 комбинации на 448 заданиях GPQA. Точность монотонно растет с увеличением размера мыслителя, тогда как влияние размера отвечающей модели остается умеренным. Более крупные мыслители порождают более короткие и более высокоэнтропийные цепочки рассуждений средняя длина примерно 4 639 токенов, энтропия 0.416, чем меньшие модели 14 566 и 0.404, и эти характеристики коррелируют с более эффективным переносом между моделями. Практический вывод заключается в том, что целесообразно кешировать рассуждения с помощью сильной LLM и выполнять только этап ответа малой LLM, приближаясь к лучшей диагональной точности при меньших вычислительных затратах.

Ключевые слова: цепочка рассуждений CoT, перенос рассуждений между моделями, Qwen3, бенчмарк GPQA, энтропия токенов LLM.