

DeepSeek-v3.2 Overthinks Less in Chinese

Xueli Sun

Dec 4, 2025

This is a report for Special Participation A for EECS182, Fall 2025. The code implementation and raw response for this experiment is available at: <https://github.com/shirley430316/berkeley-cs182-participation>

1 Introduction

DeepSeek-v3.2 was released last week and has drawn significant attention, as its performance is catching up with that of OpenAI’s GPT-5 and Google’s Gemini 3 on many benchmarks. An interesting phenomenon has also been noted in the community: the model occasionally thinks in Chinese even when the prompt is written entirely in English. This raises a natural question—how does the model’s behavior differ when it reasons in English versus Chinese?

The results are striking. For the same question extracted from Homework 11, the model requires less than 40% of the time and only about 33% of the tokens when reasoning in Chinese compared with English. I also propose a single-sentence Chinese prompt that induces the model to think in Chinese with roughly 90% probability.

2 Experiment

2.1 Testing Question

I selected Question 5 from Homework 11 as the test case. This question involves estimating the training and inference costs of a hypothetical GPT-6 model using Fermi estimation. It requires reading comprehension and logical reasoning, along with some light mathematical calculations. The full question, formatted in Markdown, is provided in Appendix A.

2.2 Ablation

For the baseline group, the model receives only the original question in English. As expected, its entire chain of thought is produced in English. For the experiment group, I prepend the instruction: “请务必用 ** 中文 ** 思考，并用英文回答以下问题。” (“Please make sure to reason in **Chinese** and answer the following question in English.”). Each group is evaluated under identical conditions, with ten trials per group and no additional context.

2.3 Quantitative Results

Baseline Group

Table 1: Baseline Group Experimental Results (English Thinking)

Lang.	Time (ms)	Think Chars	Resp Chars	Think Tokens	Resp Tokens	Total Tokens
EN	282.39	18028	3399	5408.4	1019.7	6428.1
EN	444.42	32352	2794	9705.6	838.2	10543.8
EN	433.85	30039	3438	9011.7	1031.4	10043.1
EN	363.83	24170	3150	7251.0	945.0	8196.0
EN	429.52	30402	4286	9120.6	1285.8	10406.4
EN	277.70	18169	3852	5450.7	1155.6	6606.3
EN	197.90	10114	4023	3034.2	1206.9	4241.1
EN	403.64	26935	4214	8080.5	1264.2	9344.7
EN	233.04	12241	3912	3672.3	1173.6	4845.9
EN	321.53	19705	3602	5911.5	1080.6	6992.1
Avg.	338.78	—	—	—	—	7764.75

Note: Lang. = Thinking Language; Time = Response Time (ms); Think Chars = Thinking Characters; Resp Chars = Response Characters; Think Tokens = Thinking Tokens; Resp Tokens = Response Tokens.

Experiment Group

Lang.	Time (ms)	CN chars	EN chars	Resp Chars	Think Tokens	Resp Tokens	Total Tokens
CN	216.42	1983	972	4994	1481.4	1498.2	2979.6
CN	72.46	301	36	4107	191.4	1232.1	1423.5
CN	94.57	859	293	4739	603.3	1421.7	2025.0
CN	109.17	708	125	5986	462.3	1795.8	2258.1
CN	111.46	1168	1063	3569	1019.7	1070.7	2090.4
CN	116.20	1047	1242	4263	1000.8	1278.9	2279.7
CN	90.33	680	1113	4905	741.9	1471.5	2213.4
CN	273.15	3822	5559	3688	3960.9	1106.4	5067.3
CN	133.38	1278	1463	4951	1205.7	1485.3	2691.0
EN	291.40	0	14920	5152	4476.0	1545.6	6021.6
Avg.	150.85	—	—	—	—	—	2904.96
(Avg.)	135.24	—	—	—	—	—	2558.67

Note: (1) Lang. = Thinking Language; Time = Response Time (ms); CN chars = Chinese characters in the thinking process; EN chars = English characters in the thinking process; Think Chars = Thinking Characters; Resp Chars = Response Characters; Think Tokens = Thinking Tokens; Resp Tokens = Response Tokens. (2) The first average value represents the mean across all 10 independent experiments. The second average excludes the final experiment (row 10) to focus specifically on the performance when reasoning in Chinese, excluding the efficiency of the prepended Chinese prompt.

The results sufficiently show that the released model overthink less in Chinese, with response time shrinking to 40% and token used decreasing to 33%. It also proves that the proposed prompt has 90% successful rate to induce the model to think in Chinese.

2.4 Qualitative Results

The final responses are similar in quality with correct figures. One comparison is in Appendix B. Full responses can be found in the Github repository.

3 Conclusion and Discussion

3.1 Summary of Findings

The experimental results reveal a striking efficiency advantage when DeepSeek-v3.2 reasons in Chinese compared to English. Quantitatively, Chinese reasoning requires less than 40% of the computational time and only approximately one-third of the tokens needed for equivalent English reasoning. Practically, we have demonstrated that this efficiency can be reliably accessed through a minimal one-sentence Chinese prompt, which induces Chinese thinking with approximately 90% success rate.

3.2 Potential Explanations

Two primary factors may explain this observed efficiency gap:

1. **Information Density:** Chinese characters typically encode more semantic information per token than English words, potentially allowing more efficient internal representations.
2. **Training Data Distribution:** As a model developed in China, DeepSeek-v3.2 was likely trained on extensive Chinese-language corpora, which may have optimized its reasoning pathways for Chinese processing.

3.3 Limitations

This study has several limitations that suggest directions for future research:

- **Single Test Question:** The experiment used only one Fermi estimation problem, limiting generalizability. However, the large effect size (3x efficiency difference) suggests meaningful findings despite this limitation.
- **Exploratory Depth:** This represents preliminary investigation; I may need seek deeply to isolate specific mechanisms (tokenization, architecture, training).

This represents an interesting and useful finding for practical AI usage. The efficiency gap likely arises from Chinese’s higher information density and DeepSeek’s substantial Chinese training data. While based on a single test question, the large effect size makes the observation noteworthy. This suggests that language selection is a meaningful parameter when working with multilingual models like DeepSeek-v3.2.

Appendix A: Complete Test Question

Fermi Estimation for Large-scale Deep Learning Models

In this question, you will perform **Fermi estimates** for a hypothetical GPT-6 model with **100 trillion parameters** (10^{14}). You will explore scaling laws, memory, inference, cost, and environmental impact using rough, order-of-magnitude calculations. Assume 16-bit parameters (2 bytes each) unless stated otherwise.

(a) Compute and dataset scaling

The Chinchilla scaling laws relate model size, dataset size, and training compute. For a given compute budget C in FLOP, the optimal parameter count N and token count D scale as $N = 0.1C^{0.5}$ and $D = 1.7C^{0.5}$. If GPT-6 has 10^{14} parameters, **what training compute C is required, and how large must the training dataset D be?**

(b) Dataset size in human terms

To contextualize the dataset size from (a), assume each English word corresponds to about 1.4 tokens, each page contains 400 words, and each book has 300 pages. **How many books would the training dataset correspond to?** Compare this number to the size of the Library of Congress (roughly 20 million volumes).

(c) Memory requirements

Each 16-bit parameter occupies 2 bytes. **How much memory (in GB or TB) is required to store GPT-6's 100 trillion parameters?** Given that an H200 GPU has about 100 GB of VRAM, **how many such GPUs would be needed just to hold the model in memory?**

(d) Inference latency and throughput

During inference, model parameters must be loaded from GPU memory. The H200 has a memory bandwidth of 4.8 TB/s. **What is the minimal time in seconds to perform one forward pass through GPT-6?** If the model generates tokens autoregressively (one token per forward pass), **how many tokens could it output in one minute?**

(e) Training cost in FLOPs and dollars

Training compute is often measured in petaFLOP-days. One petaFLOP-day equals about 8.64×10^{19} FLOP. GPT-3 required 3640 petaFLOP-days to train. **If trained on H200 GPUs (each delivering 1.98 petaFLOP/s and renting for \$1.50/hour), how much would it cost to train GPT-3?** Using your computed C from part (a), **estimate the cost to train GPT-6 under the same assumptions.**

(f) Inference cost and break-even

For Transformer models, inference requires about 2 FLOPs per parameter per token. **How many FLOPs are needed to generate 1 million tokens with a 1-trillion-parameter model like GPT-5?** If OpenAI charges \$120 per million tokens, **how many tokens must be sold to recoup a \$1 billion training cost?** Express this in terms of 1000-word essays, assuming 1.4 tokens per word.

(g) Environmental impact

Training GPT-3 emitted roughly 552 tonnes of CO₂. The social cost of carbon is around \$112 per tonne. **What is the carbon cost of training GPT-3 in USD?** For comparison, producing 1 kg of beef emits about 50 kg of CO₂. A quarter-pound burger contains about 113 g of beef. **How many burgers' worth of CO₂ does training GPT-3 represent?**

Use order-of-magnitude arithmetic, scientific notation, and clear reasoning. Show all steps concisely.

Appendix B: Qualitative Comparison

```
## (f) Inference cost and break-even**
For a 1-trillion-parameter model:
FLOPs per token = ( 2 * 10^12 ) FLOP.
For 1 million tokens: ( 2 * 10^12 ) * 10^6 = 2 * 10^18 FLOP.

Recoup $1 billion training cost at $120 per million tokens:
Millions of tokens needed: ( 10^9 / 120 ) approx 8.33 * 10^6.
Total tokens: ( 8.33 * 10^6 * 10^6 = 8.33 * 10^12 ) tokens.

Per 1000-word essay: ( 1000 * 1.4 = 1400 ) tokens.
Number of essays: ( ( 8.33 * 10^12 ) / 1400 ) approx 5.95 * 10^9.
**Answer:**
( 2 * 10^18 FLOP ) for 1M tokens; ~5.95 billion 1000-word essays to recoup $1B.

## (g) Environmental impact**
Carbon cost of GPT-3:
( 552 tonnes CO2 * 112 USD/tonne ) approx $61,824.

CO2 per burger:
( 0.113 kg beef ) * 50 kg CO2/kg beef = 5.65 kg CO2.
Number of burgers equivalent to GPT-3 training:
( ( 552,000 kg CO2 ) / 5.65 kg CO2/burger ) approx 97,700 burgers.
**Answer:**
Carbon cost ~$61,800; equivalent to ~97,700 quarter-pound burgers.
```

```
** (f) Inference cost and break-even**
FLOPs for 1 million tokens with 1-trillion-parameter model:
( 10^12 ) parameters * 2 FLOP/parameter/token * ( 10^6 ) tokens = ( 2 * 10^18 ) FLOP

To recoup $1 billion training cost at $120 per million tokens:
Tokens needed: ( ( 10^9 ) / ( $120 / million tokens ) ) = 8.33 * 10^6 million tokens = ( 8.33 * 10^12 ) tokens

In terms of 1000-word essays:
Tokens per essay: 1000 words * 1.4 tokens/word = 1400 tokens
Essays equivalent: ( ( 8.33 * 10^12 ) tokens ) / ( 1400 tokens/essay ) approx 5.95 * 10^9 essays

** (g) Environmental impact**
Carbon cost of training GPT-3:
552 tonnes CO2 * $112/tonne = $61,824

CO2 per quarter-pound burger:
113 g beef = 0.113 kg beef
CO2 per burger: 0.113 kg * 50 kg CO2/kg beef = 5.65 kg CO2

Number of burgers equivalent to GPT-3 training:
552 tonnes = 552,000 kg CO2
552,000 kg / 5.65 kg/burger = 97,699 burgers

**Summary of key figures:**
- Training compute: ( 10^30 ) FLOP
- Training dataset: ( 1.7 * 10^15 ) tokens (~10 billion books)
- Model memory: 200 TB (requires ~2000 H200 GPUs just for storage)
- Single forward pass latency: ~42 seconds (yields ~1.44 tokens/minute)
- Single-GPU training cost for GPT-6: ~$210 billion (ignoring parallelism)
- Break-even tokens for $1B training: ~8.3 trillion tokens
- GPT-3 carbon cost: ~$62k, equivalent to ~100k burgers' emissions
```

Figure 1: The response from baseline group (left) and experiment group (right) for question (f) and (g). Numbers are the same, whereas the experiment group also includes a summary.

Appendix C: Token Calculation and Costs

Tracking the exact tokens used is not directly supported by the DeepSeek API. To avoid unconventional methods, I calculated token counts based on Chinese and English characters in the reasoning and response texts, then applied DeepSeek's provided conversion formula:

1 English character = 0.3 token

1 Chinese character = 0.6 token

The total testing cost was **0.57 CNY** (approximately **0.08 USD**).