

Mind's Eye of LLMs: Visualization-of-Thought Elicits Spatial Reasoning in Large Language Models

Project Page

Wenshan Wu[†], Shaoguang Mao[†], Yadong Zhang^{†, ‡}, Yan Xia[†], Li Dong[†], Lei Cui[†], Furu Wei[†]

† Microsoft Research ‡ East China Normal University



38th Conference on Neural Information Processing Systems (NeurIPS 2024)





Introduction

Spatial Reasoning in Human Cognition

- Mental Image: abstract representations from visual perception
- Mind's Eye: mental image manipulation





Motivation

Similar Mechanism in LLMs: Mind's Eye

- Visualize internal states
- Manipulate mental images to guide subsequent reasoning





Contribution

- We conduct quantitative and qualitative analyses on the mind's eye of LLMs and its limitations. We also explore cues about the origin of this generalized ability from code pre-training.
- We develop two tasks of "visual navigation" and "visual tiling", along with corresponding synthetic datasets, emulating various sensory inputs for LLMs. These tasks are structured to support varying levels of difficulty.
- We propose Visualization-of-Thought (VoT) prompting to elicit the mind's eye of LLMs for spatial reasoning and provide empirical evaluations on three tasks.



Spatial Reasoning Tasks

• Existing Benchmarks

- Spatial semantics are embedded in text, spatial term focused
- Could be solved by logic programming after converting spatial terms to logical forms through LLMs

• Ours

- Focus on spatial awareness
 - Various aspects: spatial relationships, directions, and geometric shapes
 - Essential for action planning in the physical world.
- Emulating various sensory inputs for LLMs
 - Natural language
 - 2D grid comprising of special text characters



Spatial Reasoning Tasks

- Natural Language Navigation [1]
 - Square map $W = \{(l_1, o_1), (l_2, o_2), \dots, (l_n, o_n)\}$, each location associated with an object
 - Navigation instructions $I = \{i_1, i_2, ..., i_k\}$
 - Task: Find the object o at specific location l determined by navigation instructions

$$o \sim p(o \in W | W = \{(l_1, o_1), (l_2, o_2), \dots, (l_n, o_n)\}, I)$$

- Visual Navigation
 - Grid map *M* consisting of *k* consecutive edges $E = \{e(s_0, s_1), e(s_1, s_2), \dots, e(s_{k-1}, s_k)\}$
 - Route planning: generate a sequence of correct directions

$$D \sim p(\{d(s_0, s_1), d(s_1, s_2), \cdots, d(s_{k-1}, s_k)\} \mid M))$$

• Next step prediction: given t navigation instructions, identify the direction of next step

$$d \sim p(d(s_t, s_{t+1}) \mid M, D_{t,0 < t < k})$$

- Visual Tiling
 - Rectangle *R* masked with *k* unique polyominoes $MP = \{mp_1, mp_2, ..., mp_k\}$
 - Two variants of each polyomino $v_{i < k} = \{v_{i1}, v_{i2}\}$, a polyomino query $q \in MP$
 - Task: identify the correct variant of q

$$v \sim p(v_q \mid R, \{mp_1, \cdots, mp_k\}, \{v_{11}, v_{12} \cdots, v_{k1}, v_{k2}\}, q)$$

[1] Yamada, Yutaro, et al. "Evaluating spatial understanding of large language models." *arXiv preprint arXiv:2310.14540* (2023).



Spatial Reasoning Tasks

Various Levels of Difficulties





Visualization-of-Thought Prompting

Visualize the state after each reasoning Step



- *x*: text sequence of input
- *v*: visualization sequence in text form
- z: language sequence of intermediate steps

$$v_i \sim p_{\theta}(v_i \mid prompt_{VoT}, x, z_{1\cdots i}, v_{1\cdots i-1})$$
$$z_{i+1} \sim p_{\theta}(z_{i+1} \mid prompt_{VoT}, x, z_{1\cdots i}, v_{1\cdots i})$$



Qualitative Results

Visual Navigation
· · · · · · · · · · · · · · · · · · ·
114 114 114 114
*** ***
×1× ×1× ×1×
· · · · · · · · · · · · · · · · · · ·
212 212 212 212
and and and and

	L
Starting from 🏠 , provide the steps to navigate	
to [.] .	



410, 410, 🏫 🟃 🟃	*** *** 🏠 📃	111 111 🏫 🗌
110. 110. 110. 110.	*** *** *** *** ×	*** *** *** ***
111. 111.	111 111	m m h h h h
111. 111 Ant 111	111 111 VII 111	*** *** · *** ***
111. 111.	111 111	11 (11) (11)
111. 212. 211. 211.	111 111 111 111	111 111 111 111
10 100 100 100 100	🛄 3113 3113 3113 3113	1 314 314 314 314 314
1. Move right	2. Move down	3. Move left
111. 111. 🏠	· · · · · · · · · · · · · · · · · · ·	¥11 ¥11 🏠
ret ret ret ret 👌	212 218 218 218 AV	111 111 111 111 A
m m 🔥 👌 👌	m m 👌 👌 📩	m m 🟃 📩 📩
211 211 🕺 211 211	*** *** 👌 *** ***	m m 🕺 🕺 m
1 411 411	1 1 1 m m	* * * ***
212 211 211 211	211. 211. 211. 211.	A 410 410 410 41
🔝 314 314 314 314	📄 211 211 211 211	🔝 sus sus sus su
4. Move down	5. Move left	6. Move down



To fit all the provided polyominoes into the empty squares, what's the correct variation of Tetromino T?

Visualize the state after each reasoning step.



Natural Language Navigation

You have been given a 3 by 3 square grid. Initially, you are at the bottom-left corner...find a cassette player...go right...a wool, go right...a conch, go up...a moving van, go left...a confectionery store, go left...a pot pie, go up...a siamang, go right...a black-and-white colobus, go right...a minivan. Now you have all the information on the map. You start at where the cassette player is located, then you go right by one step, go right...go up...go left...go left...go up...go right...go down by one step. What will you find?

Visualize the state after each reasoning step.

					· · · · · · · · · · · · · · · · · · ·	 		
s	В	Μ	S	В	Μ	S	В	М
Ρ	С	V	P	C	۷	Ρ	C	V
Т	W	C	Т	*W*	С	Т	W	*C*
V	sua	lize	2. Mc	ove r	right	3. Mo	ove i	right

1.

...

S	В	M		S	* B *	М		S	В	М
Ρ	С	V		Ρ	c	۷		Ρ	*C*	V
Т	W	C		Т	W	C		Т	W	С
7. Move up 8. Move right 9. Move down										



Dataset

Natural Language Navigation

• 200 square maps of size 3x3

Visual Navigation

- 496 navigation maps and 2520 QA instances
- Map size up to 7×9 and 9×7

• Visual Tiling

• 5 x 4 rectangle with 2 or 3 polyomino masked

Task				K Step)		Total		Mas	sk count	_ Total
TUSK	2	3	4	5	6	7	Total		2	3	- 10 441
Route Planning Next Step Prediction	8 8	16 32	32 96	64 256	128 640	248 1488	496 2520	Configuratio QA Instance	n 248 e 489	124 307	376 796



Experiments

• Settings

- GPT-4 CoT: Let's think step by step.
- GPT-4 w/o Viz: Don't use visualization. Let's think step by step.
- GPT-4V CoT: Let's think step by step.
- GPT-4 VoT: Visualize the state after each reasoning step.

	Visua	l Navigation	1		Natural-Language	
Settings	Route Planning		Next Step	Visual Tiling	Navigation	
	Completing Rate	Succ Rate	Prediction			
GPT-4 CoT	37.02	9.48	48.61	54.15	54.00	
GPT-4 w/o Viz	37.17	10.28	48.49	46.98	35.50	
GPT-4V CoT	33.36	5.65	46.59	49.62	/	
GPT-4 VoT	<u>40.77</u>	<u>14.72</u>	<u>55.28</u>	<u>63.94</u>	59.00	



Analysis

Do visual state tracking behaviors differ among prompting methods?

$$l_{v}$$
: length of visualization sequence, l_{s} : number of reasoning steps
Complete tracking rate = $\sum_{i}^{n} (l_{v} == l_{s})/n$
Patial tracking rate = $\sum_{i}^{n} (l_{v} < 0)/n$

- VoT markedly improves the visual tracking rate
- LLMs inherently exhibit the capability of visual state tracking in some tasks.







Analysis

How visualizations enhance final answers?

- Visualization Quality
 - Compliance: visualization satisfies requirements in 51-52% cases
 - Accuracy: visualization aligns with the corresponding state in 24%-26% cases
- Performance enhancement
 - LLMs are able to make correct decisions in 65%-77% of the cases when accurate internal state visualizations are generated

Task	Spatial Visu	alization	Spatial Understanding
	Compliance	Accuracy	Accuracy
Visual Navigation Visual Tilling	51.14 52.01	26.48 24.25	65.16 77.20

Analysis

Can VoT benefit less powerful language models?

- VoT offers a scaling advantage when applied to more advanced models
- Less capable models tend to rely on random guessing

~ .	Visua	l Navigation			Natural-Language	
Settings	Route Plan	ning	Next Step	Visual Tiling	Navigation	
	Completing Rate	Succ Rate	Prediction			1 (%)
GPT-3.5 CoT GPT-3.5 VoT	16.10 19.02	2.62 1.61	17.42 13.10	44.10 47.99	8.50 9.00	ccuracy
LLAMA3-8B CoT LLAMA3-8B VoT	4.65 4.97	0 0.2	28.73 26.75	47.24 46.73	16.50 15.50	A
LLAMA3-70B CoT LLAMA3-70B VoT	19.90 30.24	2.62 <u>5.85</u>	49.01 <u>54.09</u>	56.41 56.03	26.00 32.50	







Appendix

Mental Images for State Tracking

- Mark the path with unique symbols
- Mark path and direction with arrows
- Mark path with temporal steps
- Remove road: turning roads into obstacles

🟠 આ આ આ આ	<u></u>	<u> () () () () () () () () () () () () () </u>
*** *** *** ***	🔍 712 712 712 712	*** *** ***
111 YII	tre tree	• • • int :
918 918 🗌 918 918	9112 9112 9112 9112 9112	911 911 VII
*** ***	**** ****	¥118 ¥118
2112 2112 2112 2112	2112 2112 2112 2112	ALC ALC ALC ALC
918 918 918 918 💼	918 918 918 918 💼	2112 2112 2112 2112 2112
Original map	1. Move down	2. Move rig
🟫 માર માર માર જેટ	<u></u>	<u> ()</u>
*** *** *** ***	*** *** *** ***	*** *** ***
114 YII	the second second	
918 918 🔍 918 918	718 718 🔍 718 718	9118 9118 - 9118 S
*** ***	718 718 🔍 🌳 🔍	¥118 ¥118
2112 2112 2112 2112	**** **** ****	*** *** ***
918 918 918 918 💼	918 918 918 918 🏢	211 212 212 214
2 Move down	4 Move right	5 Move dov

Use round pin

Use arrows to reflect direction

🏫 🗌 🗌 ini ini	🏠 📑 📄 👬	🏫 🗌 🖬 🚧
111 III III	911 911 911 911	8118 8118 💶 8118 8118
// *//*	*** ***	x1x x1x
218 218 218 218	*** *** *** ***	*** *** ***
¥114 ¥114 🛄 🗌 🗌	\$1\$ \$1\$ (III)	*** *** 🏢 🗌
Original map	1. Move right	2. Move down
<u> </u>	<u> </u>	🏫 🗌 🖂 ¥44 ¥44
¥11 ¥11 🗌 ¥11 ¥11	ana ana 🚽 ana ana	¥11 ¥11 🗌 ¥11 ¥11
¥114 ¥114 🖶 🔁 🔛	*** ***	*** ***
*** *** ***	ana ana ana ana 🔱	*** *** *** ***
¥11\$ ¥11\$ 🛄 🗌 🗌	¥114 ¥114 🏢 📃 🗌	÷112 \$112 \$114
3. Move right	4. Move down	5. Move left

Use checklist

218 218 💼 218 218	918 918 💼 918 918
211, 213 🗌 213, 213	711 711 🗌 711 711
- ini ini ini ini -	🔶 🗌 🚧 🚧 🚧
📃 🗌 🚧 🏠	
718 718 YUS	rn rn 🗌 rn 🗸
*** ***	ini ini 🗌 🧹
Original map	Move down

Remove road to avoid turning back

¥114 ¥114 🏢 🗌 🗌	¥11\$ ¥11\$ 🏢 🗌 🗌
918 918 918 918 📃	*** *** ***
# *#*	→ ₩ ₩ W
911ê 911ê 📄 911ê 911ê	414 414 - 414 414
<u>én</u> 🗌 🚽	ant ant ant ant ant
Original map	Move right

Use numbers for temporal steps

111 ALS 🔬 📄	1 🗌 🏠 🚧	📃 🗌 🏠 🚧 🚧
911 911 911 911	*** *** ***	2 જેમ જેમ જેમ જેમ
114 114	×11 ×11	
11 YIL 🗌 YIL YIL	414 414 📃 714 71A	711 711 🗌 711 711
vi \$22\$ 🗌 🗌 🗊	\$11\$ \$11\$	¥114 ¥114 📃 🟢
Original map	1. Move left	2. Move down
🗌 🏠 🚧 🚧	🗌 🗌 🏤 🚧	🗌 🗌 🏠 🚧 🚧
911 911 911 911	*** *** ***	<u> </u>
3 711 711	×11 ×11	
11 YIL - YIL YIL	218 218 🐴 218 218	211 211 🗌 211 211
uk \$22\$ 📃 📄 🏢	*** ***	¥11\$ ¥11\$ 📃 互 🏢
8. Move right	4. Move down	5. Move right



Appendix

Ascii-art in Code Comments

- Represents data structure, diagram, geometry
- Illustrates how an algorithm works or simulates an operation
 - Spatial Causality: <u>Double-ended queue in Rust</u>, <u>Scrolling web pages</u>, <u>tree rotation</u> present triplets of previous visual state, instruction, and updated state of instruction following.
 - Temporal Causality: <u>Undo systems from emacs</u> provides various temporal states of the undo system when undo operation happens in different timelines and corresponding visualizations in an interleaved manner. Each visualization reflects the temporal casuality of the system state.