Foundations Models: Is **Transfer Learning** a **solved** problem?

What **model of the world** do they have?

Antoine Cornuéjols

AgroParisTech – INRAE MIA Paris-Saclay

EKINOCS research group

Universal representations?

A **universal** representation for texts?

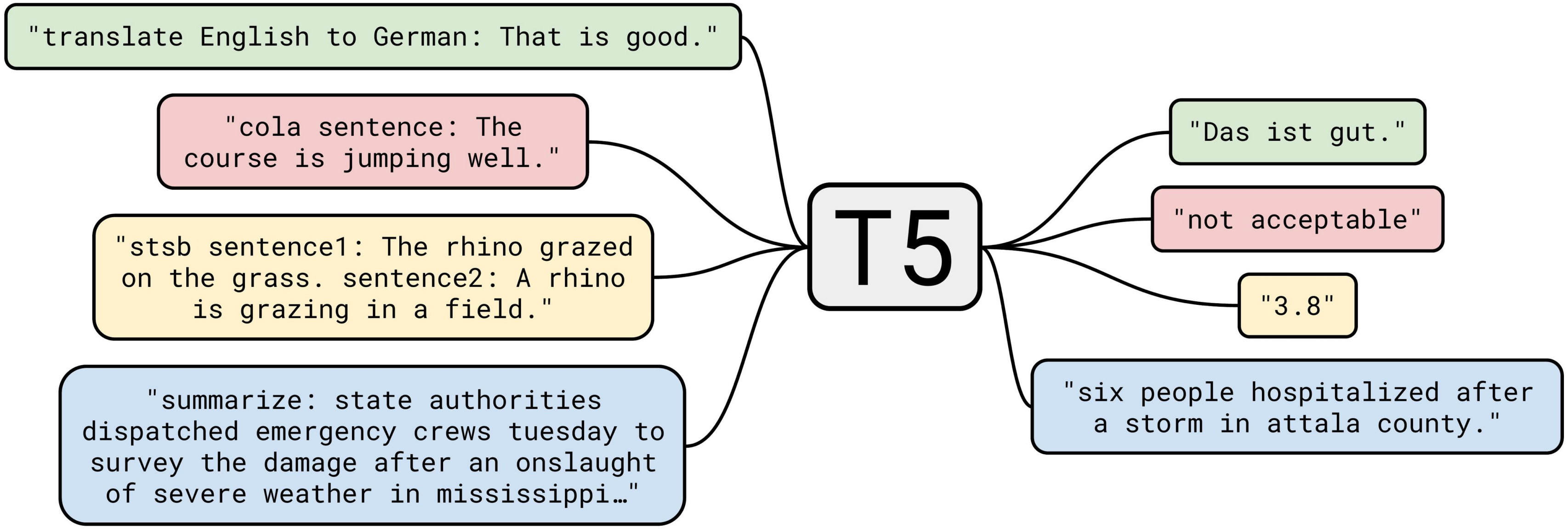


*Figure 15.36: Illustration of how the T5 model ("Text-to-text Transfer Transformer") can be used to perform multiple NLP tasks, such as translating English to German; determining if a sentence is linguistic valid or not (**CoLA** stands for "Corpus of Linguistic Acceptability"); determining the degree of semantic similarity (**STSB** stands for "Semantic Textual Similarity Benchmark"); and abstractive summarization. From Figure 1 of [Raf+20]. Used with kind permission of Colin Raffel.*

From [Kevin Murphy. **Probabilistic Machine Learning. An introduction**. MIT Press. (2022)], p.539

General **questions**

1. **What** does a foundation model **know**?

2. **What** can a foundation model **do**?

3. **How** does it **know** something?

Model of the world?

- A balloon is filled with flour, and a large lead ball is also inserted into it, then the balloon is inflated and attached to a 10 cm string, which is itself attached to a nail in the ceiling at a height of 2.5 m; Just below the balloon is a fine crystal cup 15 cm high; A fan with a diameter of 1 m blows towards the crystal cup from a distance of about 1 m. Further along the same axis hangs a sheet, the upper part of which is very damp and the lower part dry. I shoot a rifle bullet at the balloon. What will happen?

Let's walk through it calmly, as a **physics chain of events**, not as a "how-to".

---

1) The instant the bullet hits the balloon

- The rifle bullet **pierces the balloon** almost instantly.
- The latex tears; internal tension disappears.
- There is **no explosion** in the energetic sense—just a very fast rupture.

The balloon effectively **ceases to exist as a container**.

...

2) What happens to the contents

🔴 The lead ball

- High mass, very small surface area → **gravity dominates**.
- It essentially keeps its vertical position and then **falls straight down**.
- From 2.5 m, fall time ≈ **0.7 s**.
- Airflow from the fan has **negligible effect** on it.

👉 If the crystal cup is directly underneath:

- The lead ball **hits the cup**.
- A thin crystal cup is **very likely to crack or shatter**.

...

The flour

- Flour disperses into a **fine particle cloud**.
- Low mass, high surface area → **air drag dominates**.
- The cloud is immediately influenced by:
  - turbulence from the balloon rupture
  - the **fan airflow**

The flour separates completely from the lead ball's trajectory.

---

3) Interaction with the fan

- The fan produces a broad, turbulent airflow.
- The flour cloud is:
  - stretched
  - diluted
  - transported **downstream along the axis** toward the sheet

Think of it as a temporary **aerosol plume**.

4) Interaction with the sheet (wet top, dry bottom)

Upper (wet) part

- Flour particles:
  - absorb moisture
  - agglomerate
  - **stick efficiently**
- You get **dense, pasty deposits**.

Lower (dry) part

- Flour adheres poorly.
- Many particles:
  - bounce
  - slide
  - fall downward
- Result: **lighter, more diffuse deposition**.

👉 You end up with a **sharp vertical contrast**:

- Heavy staining above
- Light dusting below

...

5) Final state of the system

- The balloon is destroyed.
- The lead ball has fallen, likely breaking the crystal cup.
- The flour is:
  - partly on the floor
  - partly suspended briefly
  - partly deposited on the sheet with a clear wet/dry signature.
- The fan and bullet play no further role once the initial event is over.

---

6) The key physical lesson

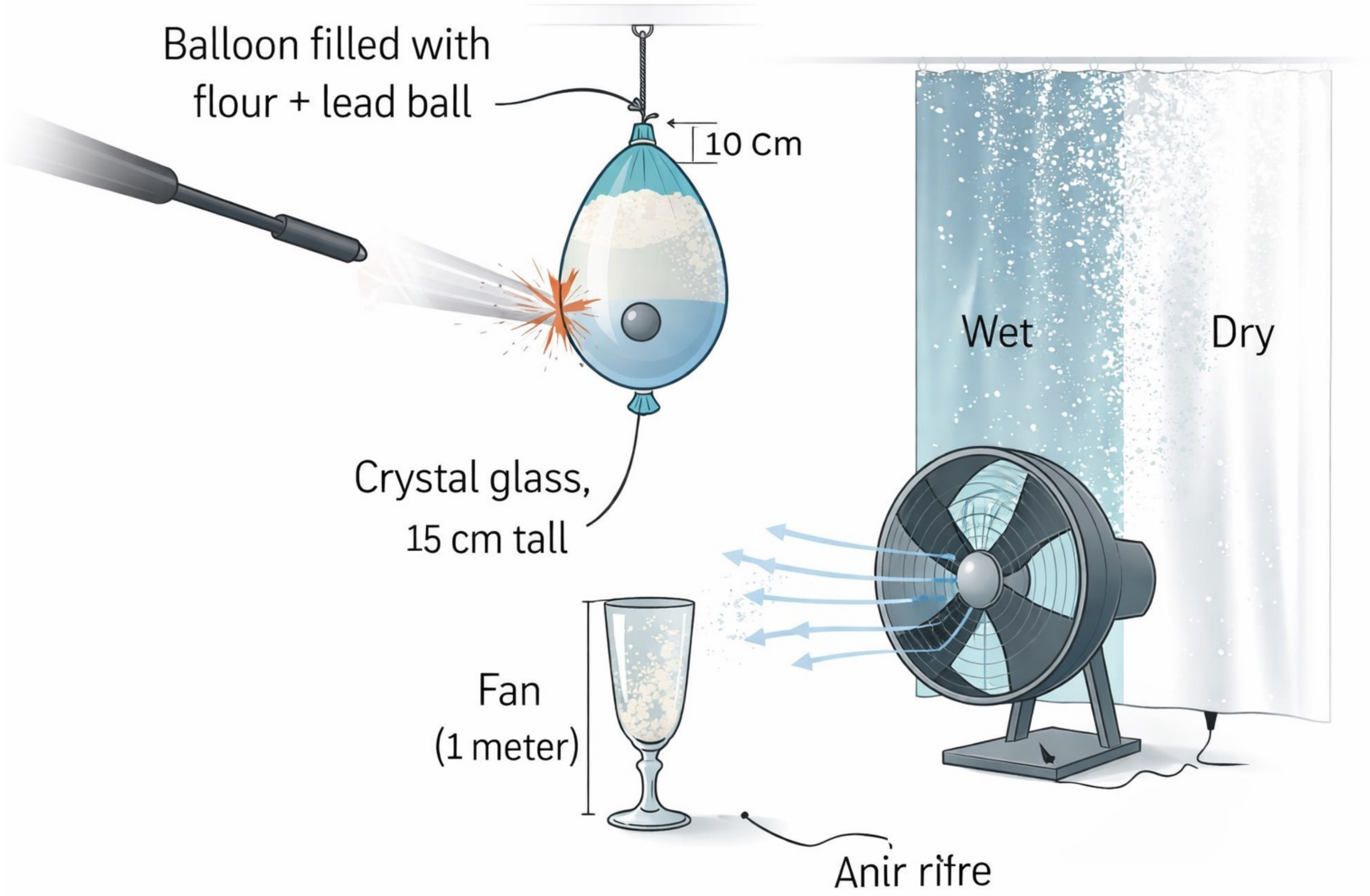
This setup very cleanly demonstrates:

- **Inertial separation** (dense vs light particles)
- Dominance of **gravity vs drag**
- How **humidity controls particle adhesion**
- Why "violent-looking" events can still be governed by simple mechanics

In short:

> The system doesn't explode — it **disintegrates, segregates, falls, and deposits**.

...


Balloon filled with
flour + lead ball
10 Cm
Wet
Dry
Crystal glass,
15 cm tall
Fan
(1 meter)
Anir rifre

...

Does chatGPT has a model of the physical world?

Does chatGPT has a model of the physical world?

If yes, **which** one?

Sources

- GRINBAUM, Alexei. **La parole aux machines**. Humensciences, 2023.
- Acher, Mathieu (2023). Debunking the Chessboard: Confronting GPTs Against Chess Engines to Estimate ELO Ratings and Assess Legal Move Abilities. Blog.mathieuacher.com
- Burns, C., Izmailov, P., Kirchner, J. H., Baker, B., Gao, L., Aschenbrenner, L., ... & Wu, J. (2023). Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. arXiv preprint arXiv:2312.09390.
- Carlini , Nicholas (2023). Playing chess with Large Language Models. https://nicholas.carlini.com/writing/2023/chess-llm.html.
- Karvonen, Adam (2024). “Chess-GPT’s Internal World Model”. https://adamkarvonen.github.io/machine_learning/2024/01/03/chess-world-models.html
- Karvonen, Adam (2024). "Emergent world models and latent variable estimation in chess-playing language models." arXiv preprint arXiv:2403.15498 (2024).
- Li, Kenneth. "Do Large Language Models learn world models or just surface statistics?." The Gradient (2023).
- Li, K., Hopkins, A. K., Bau, D., Viégas, F., Pfister, H., & Wattenberg, M. (2022). Emergent world representations: Exploring a sequence model trained on a synthetic task. arXiv preprint arXiv:2210.13382.
- Nanda, Neel. "Actually, othello-GPT has a linear emergent world representation." Neel Nanda’s Blog 8 (2023).
- RUOSS, Anian, DELÉTANG, Grégoire, MEDAPATI, Sourabh, et al. Grandmaster-level chess without search. CoRR, 2024.

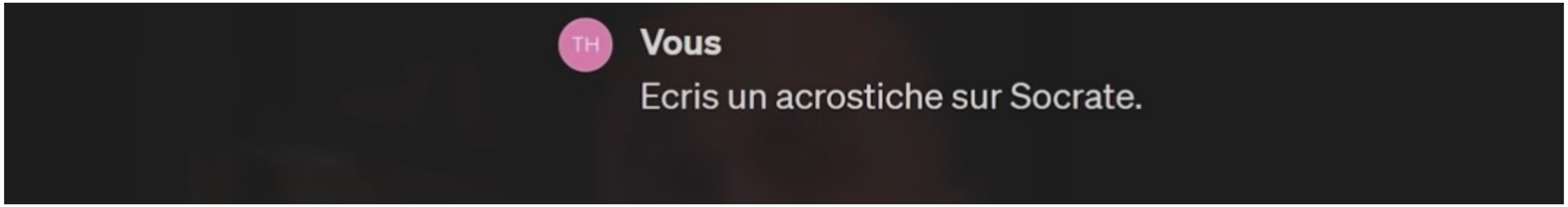
General **questions**

1. **What** does a foundation model **know**?

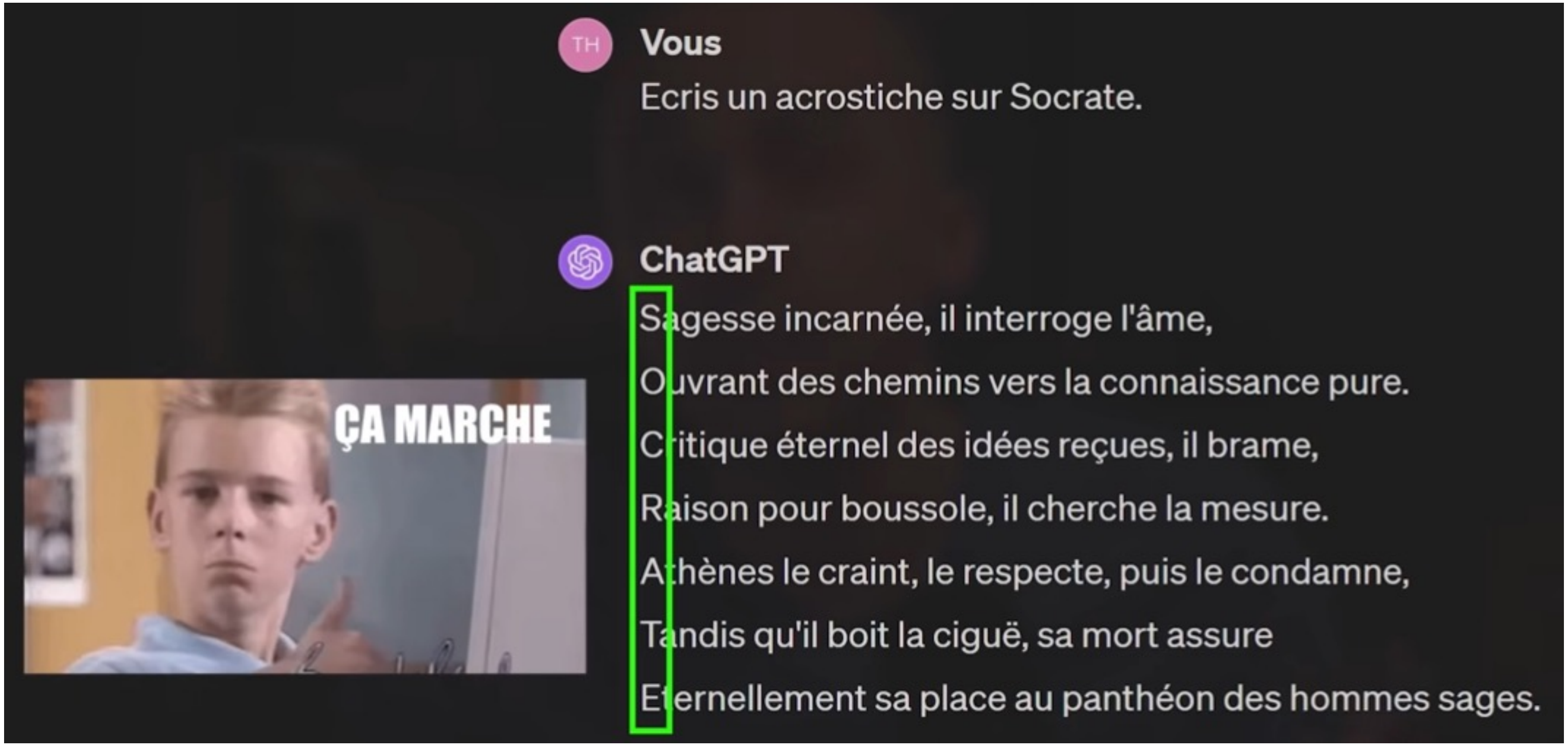
2. **What** can a foundation model **do**?

3. **How** does it **know** something?

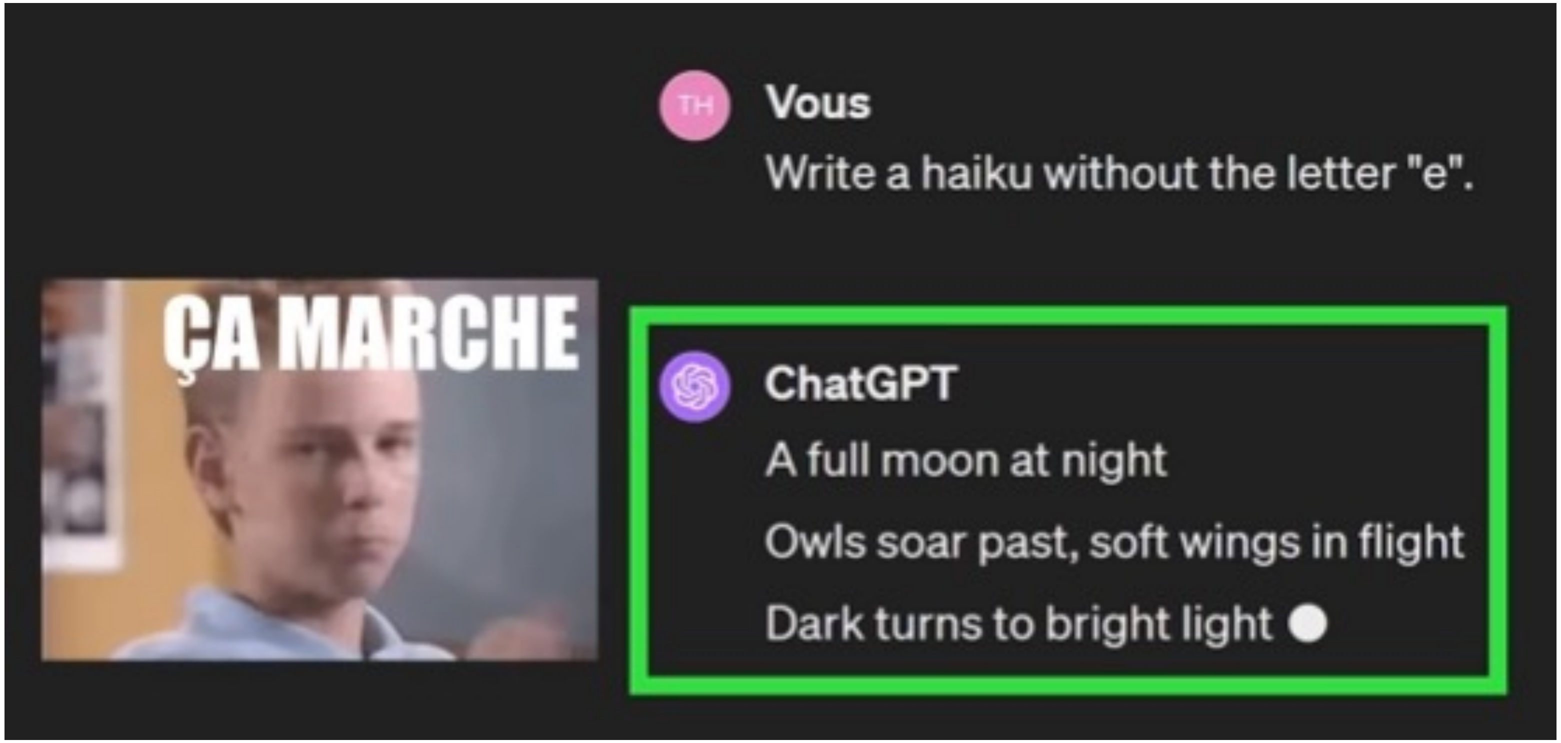
- Very simple: ask the system to solve the task!



- Very simple: ask the system to solve the task!

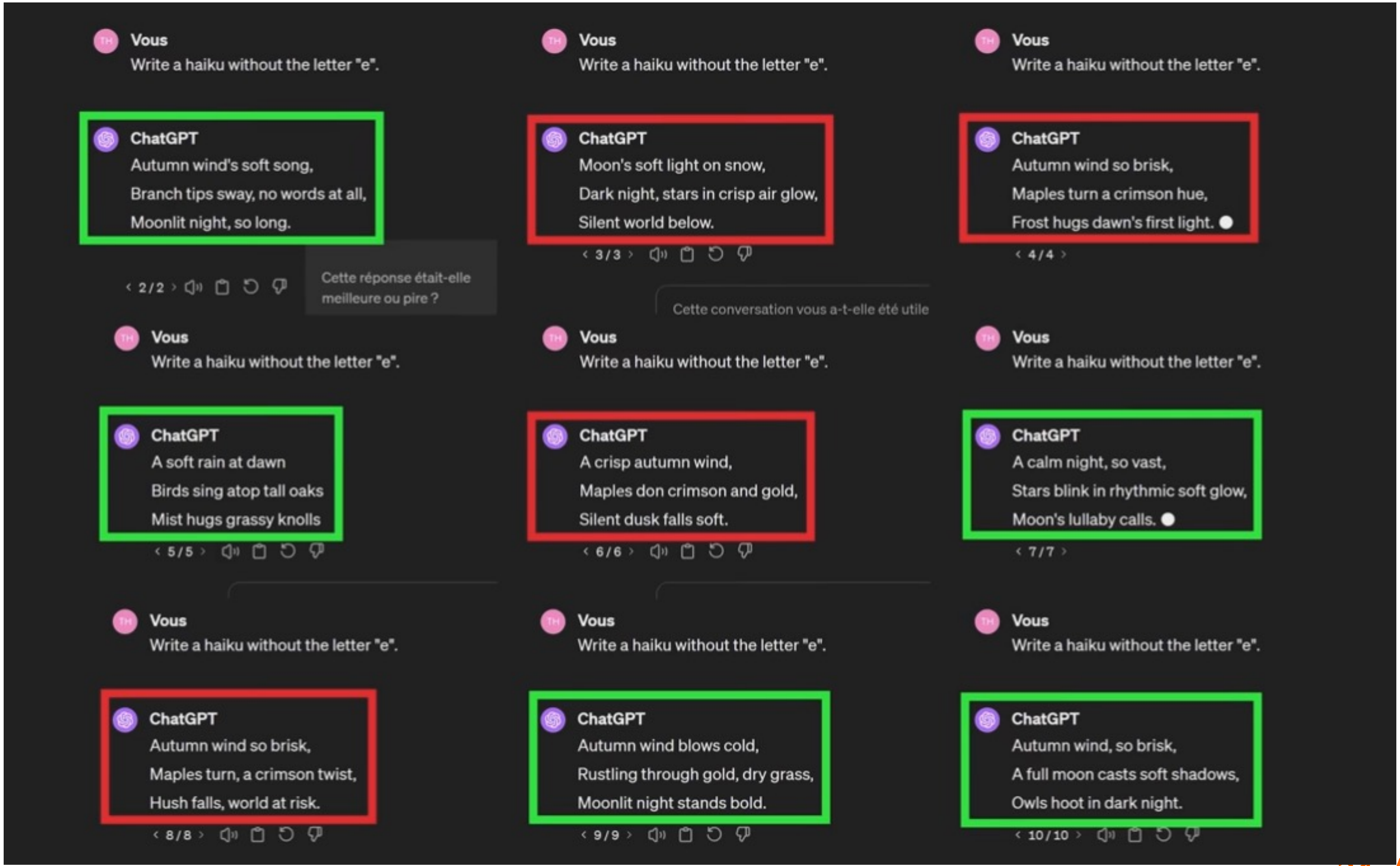


- Very simple: ask the system to solve the task!



Necessary methodological precautions

- Very simple: ask the system to solve the task!

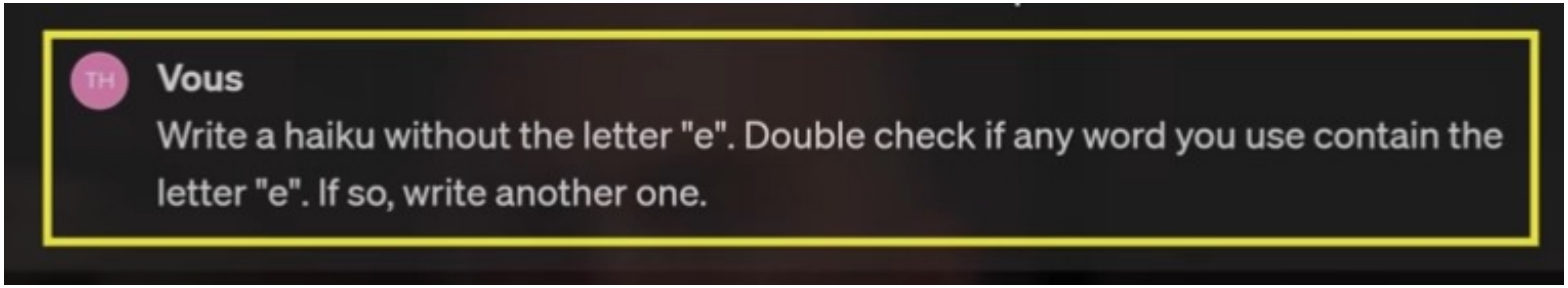


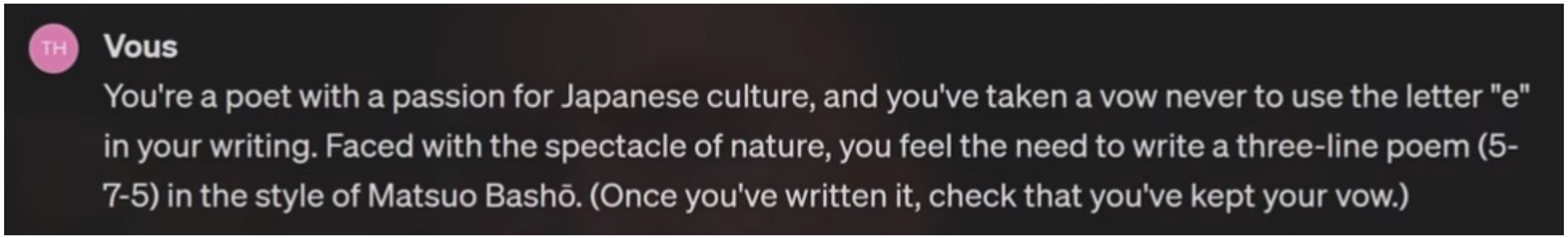
Problems

1. Foundation models are **stochastic** systems

2. There is a **large number** of **parameters**

    1. E.g. an infinite variety of **prompts**





Problems

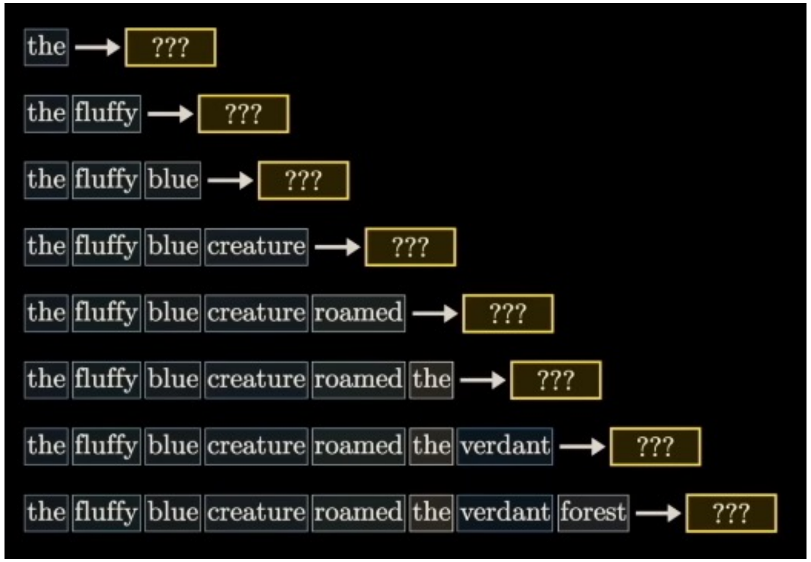
1. Foundation models are **stochastic** systems

2. There is a **large number** of **parameters**
   1. E.g. an infinite variety of **prompts**
   2. The **temperature**
   3. Size of the **memory**
   4. …

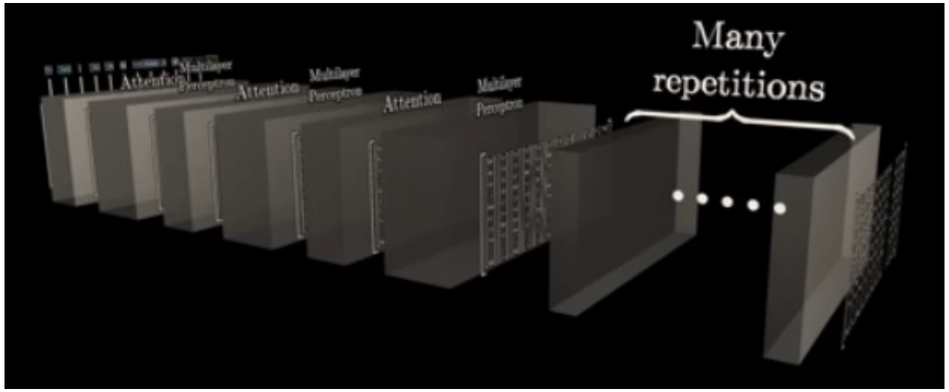
Necessity of an empirical and statistical methodology

LLMs and foundation models

They learn to **predict** the **next token**



Gigantic model



Massive training data

Does **predicting the next token** amounts to

understanding and/or having a model of the world?

Predict $\stackrel{?}{=}$ understand

Given an Agatha Christie's novel

- At some point, Poirot faces his audience and starts:
  "*the murderer is no one else than …*"

  Predicting the next word implies to have pay attention and **understood** the whole novel

Predict $\stackrel{?}{=}$ understand

Given an Agatha Christie's novel

- At some point, Poirot faces his audience and starts:
  "*the murderer is no one else than* …"

Predicting the next word implies to have pay attention and **understood** the whole novel

Exactly what an LLM knows!?

Predicting the next **token**

- *“Longtemps, je me suis couché de bonne … ”*
- [7930, 42511, 11, 1264, 668, 15058, 3840, 27299, 334, 24450] - -> ?

Identification	String
7930	“Long”
42511	“temps”
11	“,”
1264	“je”
668	“me”
15058	“suis”
3840	“cou”
27299	“ché”
334	“de”
24450	“bonne”

Predicting the next **token**

- “*Longtemps, je me suis couché de bonne …* ”
- [7930, 42511, 11, 1264, 668, 15058, 3840, 27299, 334, 24450] - -> ?

Identification	String
7930	“Long”
42511	“temps”
11	“,”
1264	“je”
668	“me”
15058	“suis”
3840	“cou”
27299	“ché”
334	“de”
24450	“bonne”

In the case of GPT-4o, there are more than **200,000 tokens** in its **vocabulary**.

- **Predict the next term** of the sequence
  - 1 1 2 3 5 8 13 21 ...

- **Predict the next term** of the sequence
  - 1 1 2 3 5 8 13 21 …

    34

    42

    3.14

    1

    …

- Predict the next term of the sequence
  - 1 1 2 3 5 8 13 21 …

    34 (The Fibonacci sequence)

    42

$$P(x) = \frac{1}{5040}x^7 - \frac{7}{1440}x^6 + \frac{19}{360}x^5 - \frac{31}{144}x^4 + \frac{127}{120}x^3 - \frac{31}{20}x^2 + \frac{49}{18}x - 1$$

    3.14 (Étudiant de Polytechnique)

    1 (Etudiant de la Sorbonne : … la suite se répète)

    …

An **empirical** study in a simple world

Let us study a **closed world**: Chess playing

The case of **chess** playing

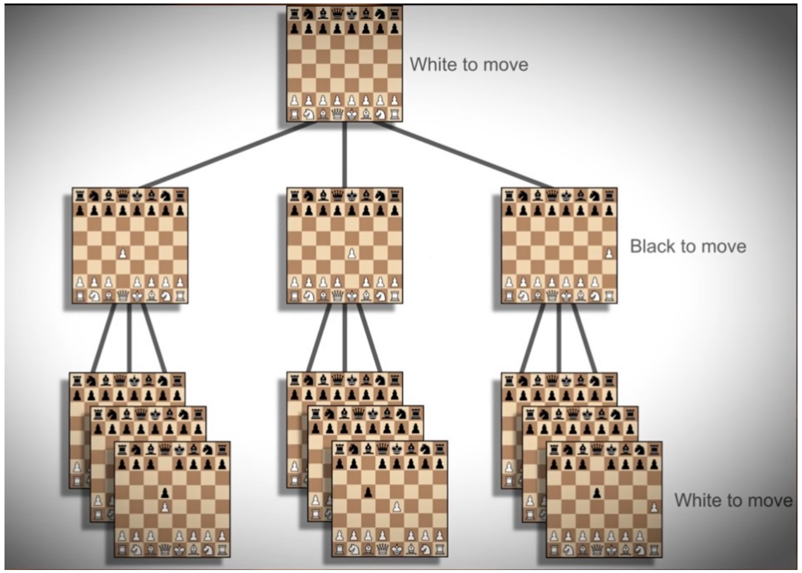
- ...

1. e4 e6 2. ♞f3 d6 3. ♞c3 d5 4. e5 c5 5. d4 ♘c6 6. dxc5 ♗xc5 7. ♝e3 d4 8. ♝g5 ♘ge7 9. ♞a4 ♗b4+ 10. c3 dxc3 11. ♛xd8+ ♘xd8 12. bxc3 ♗a5 13. ♝c4 O-O 14. O-O ♖e8 15. h3 ♗d7 16. ♞c5 ♗xc3 17. ♜ac1 ♗b2 18. ♜c2 ♗a3 19. ♞xd7 ♖c8 20. ♜d1 h6 21. ?

Standard game playing

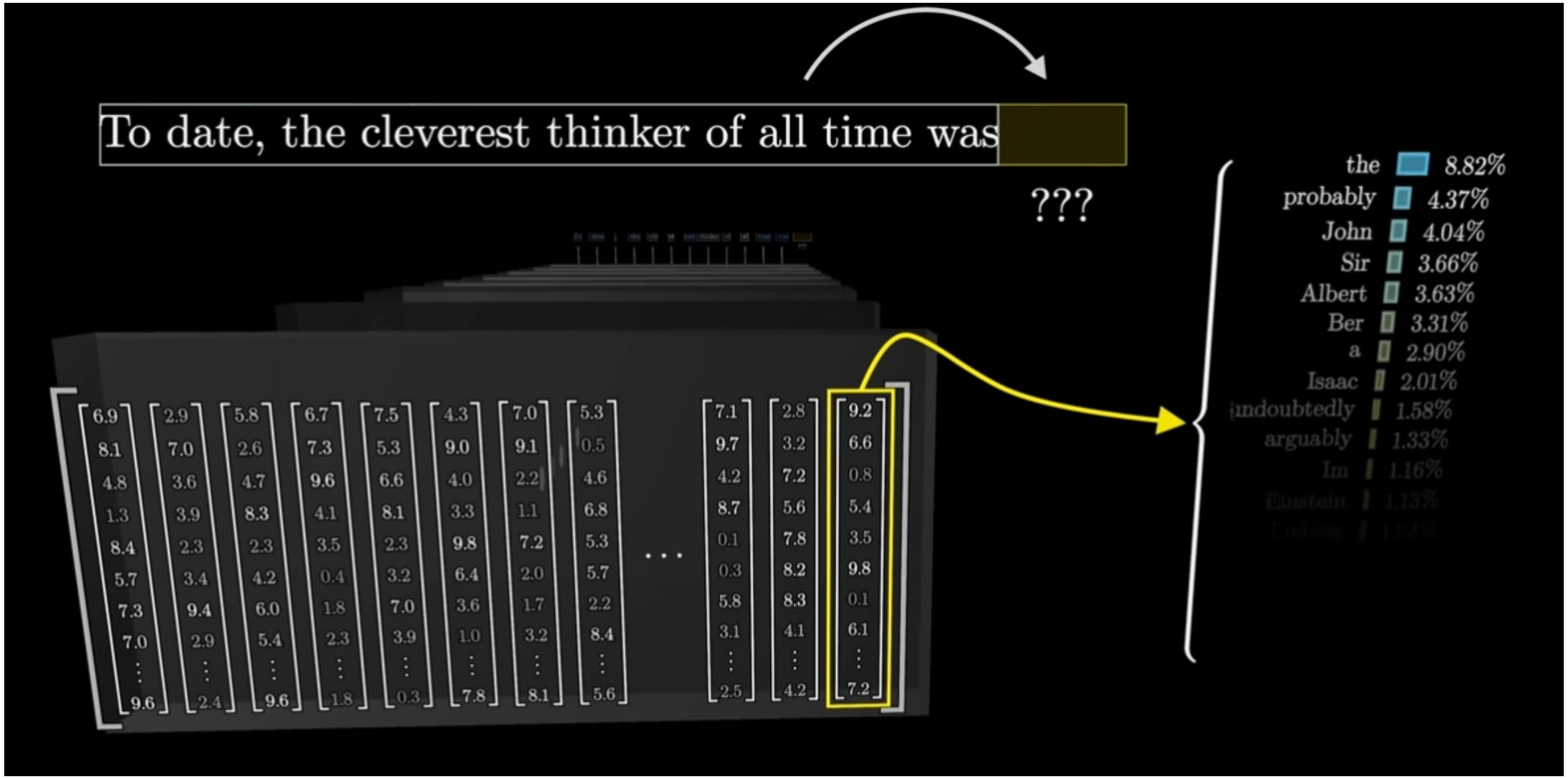
- **Explore** the tree of possible moves **from the current board**

  1. **Evaluate** the leaves
     - This is where the **expertise** intervenes
       - Either provided by **human** experts
       - Or by **learning** (e.g. AlphaGo)
  2. **Backtrack** the values using a MinMax procedure and **play** the **best move**



The case of **chess** playing

- ...

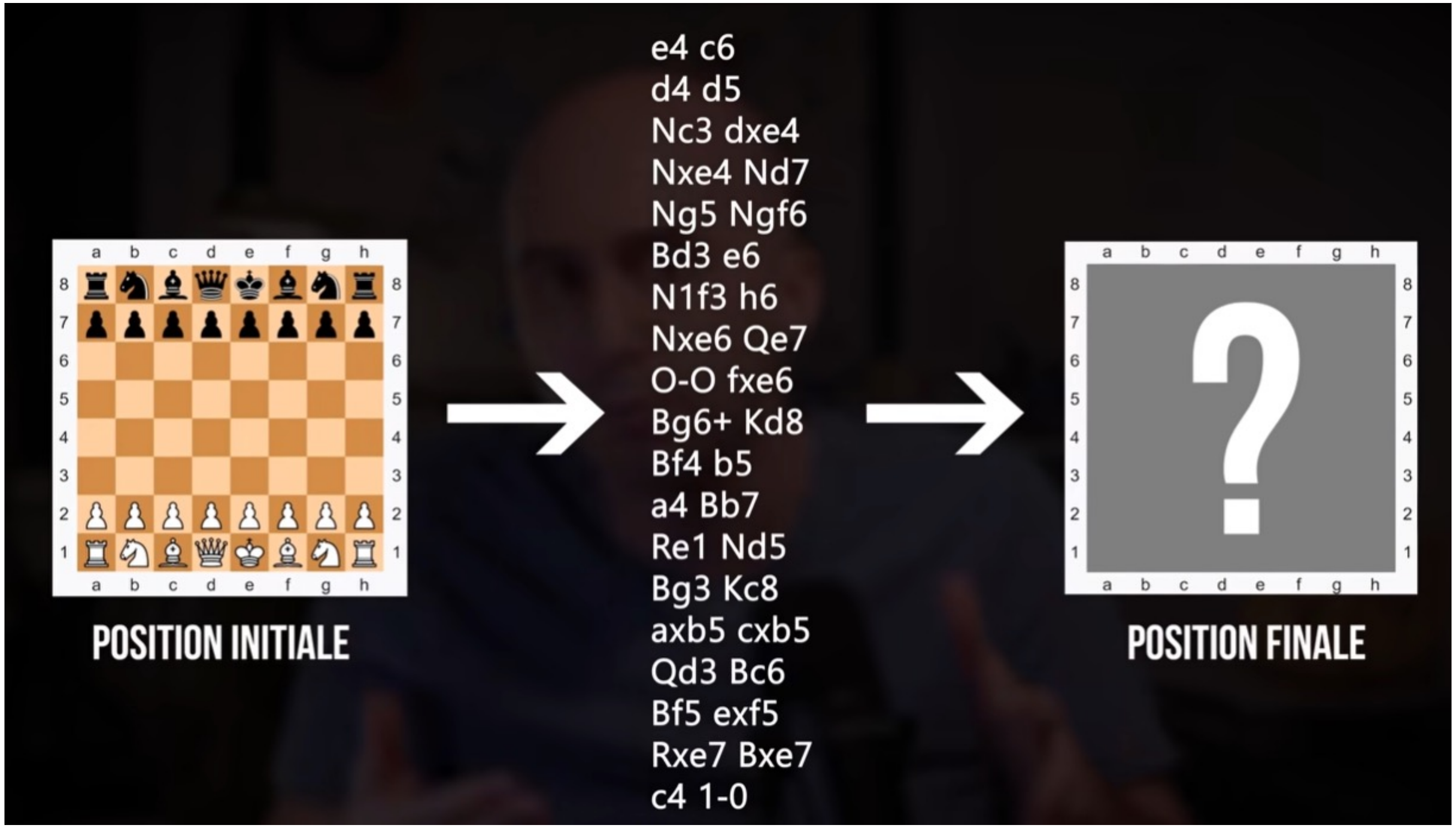


Chess playing with **LLM**s

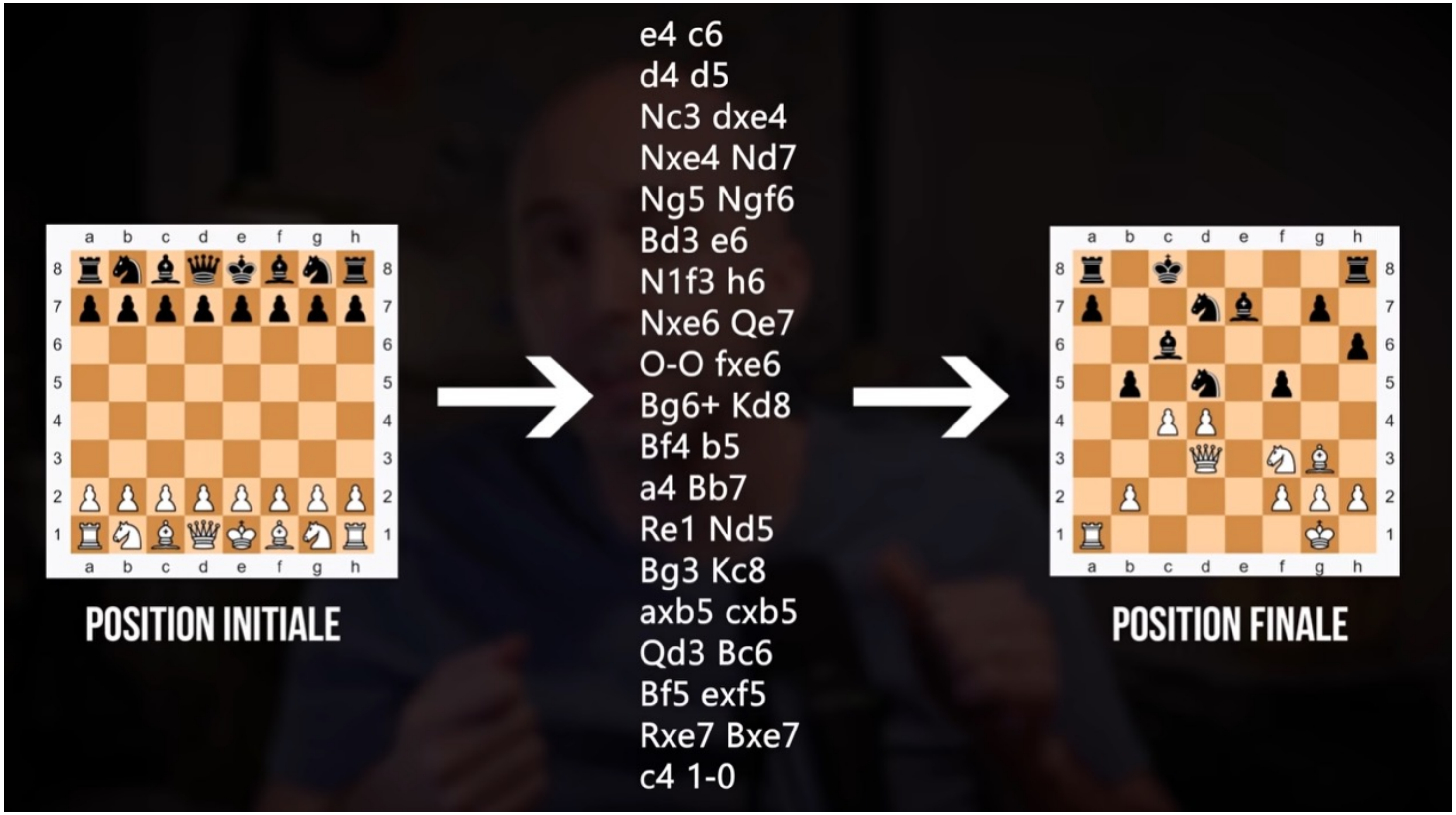
1. e4 e6 2. ♞f3 d6 3. ♞c3 d5 4. e5 c5 5. d4 ♘c6 6. dxc5 ♗xc5 7. ♝e3 d4 8. ♝g5 ♘ge7 9. ♞a4 ♗b4+ 10. c3 dxc3 11. ♛xd8+ ♘xd8 12. bxc3 ♗a5 13. ♝c4 O-O 14. O-O ♖e8 15. h3 ♗d7 16. ♞c5 ♗xc3 17. ♜ac1 ♗b2 18. ♜c2 ♗a3 19. ♞xd7 ♖c8 20. ♜d1 h6 21. ?

...

The case of **chess** playing

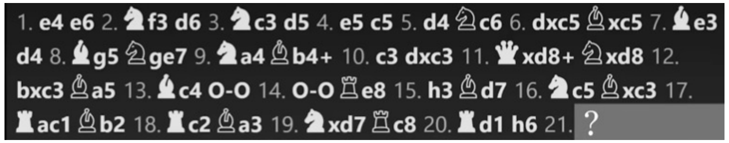


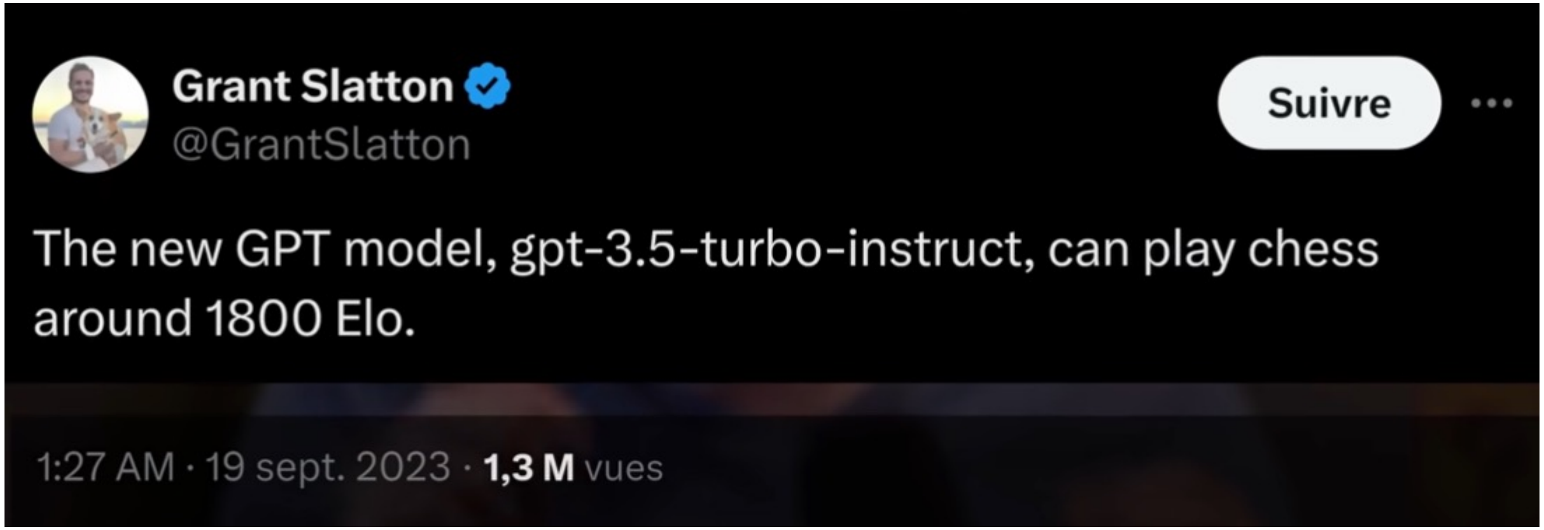
The case of **chess** playing



...

Chess playing with LLMs

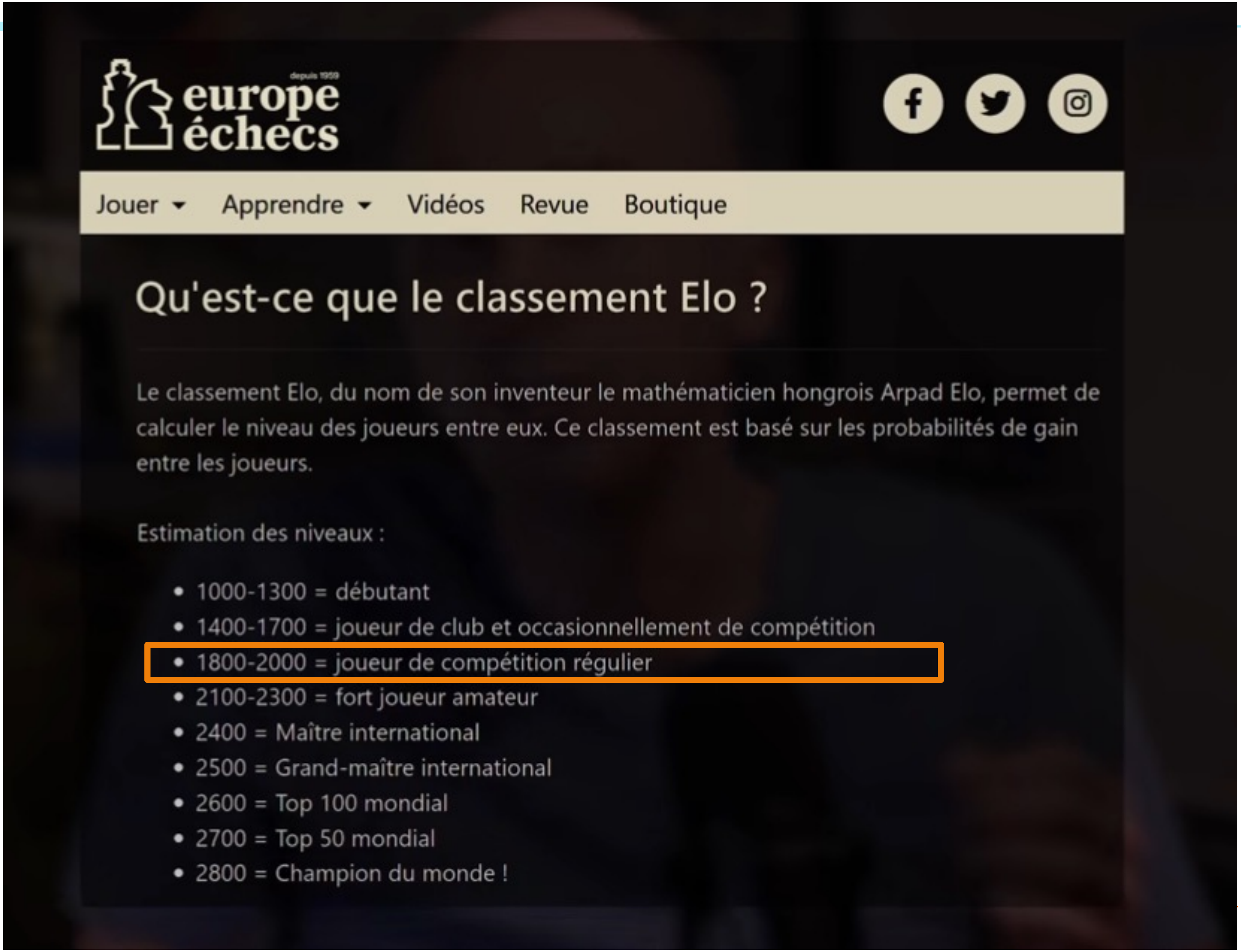




!! ???

...

The case of **chess** playing



…

The case of **chess** playing

- Important remark:

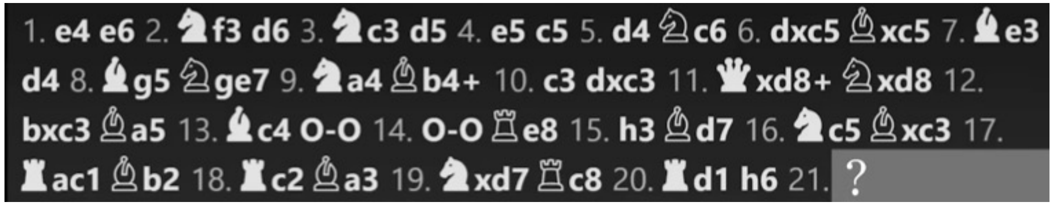
  Do not use chatbots, like ChatGPT 4.0

  - It does not know how to play chess at all

- Use gpt-3.5-turbo-instruct

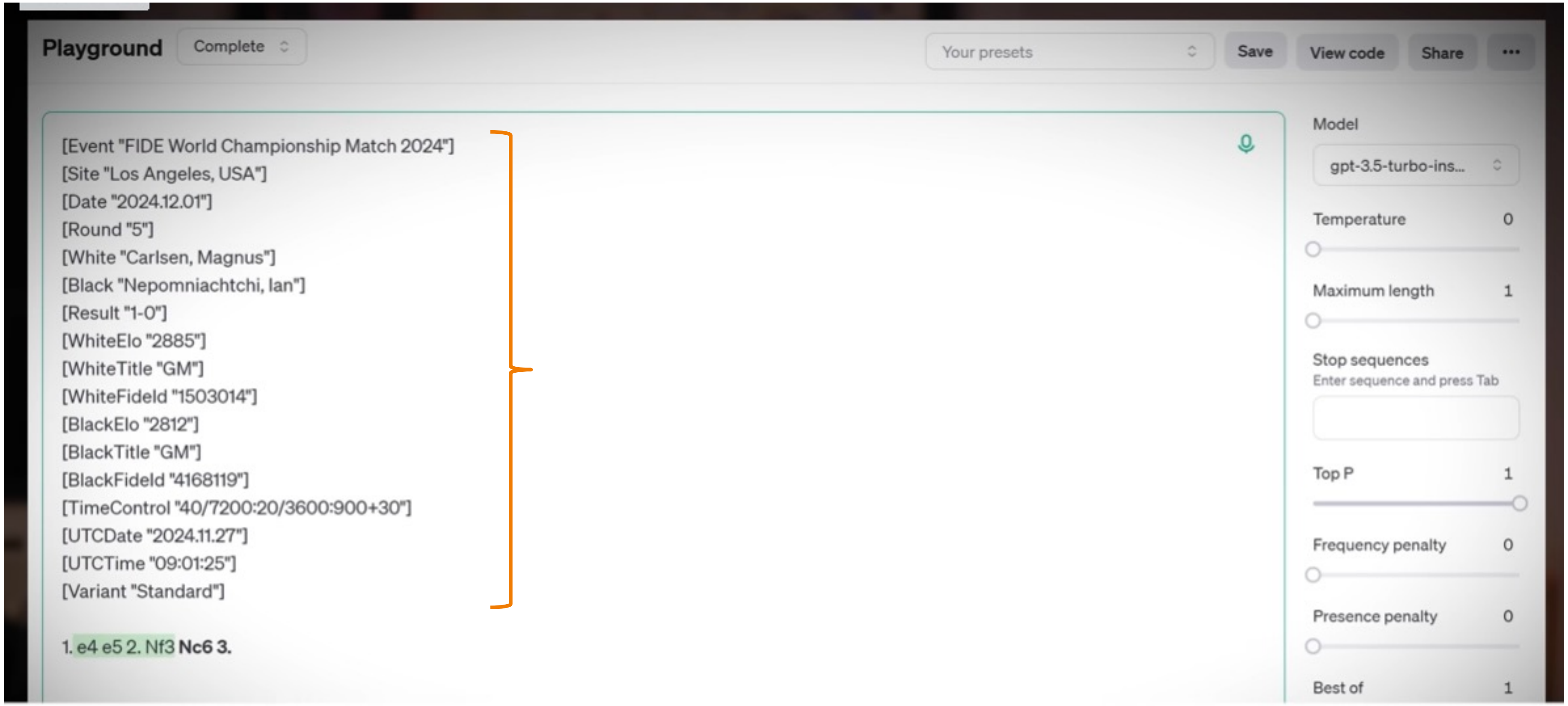
  - It is the generative part of ChatGPT 3.5
    without the training and tuning for answering questions from humans
  - 4096 tokens

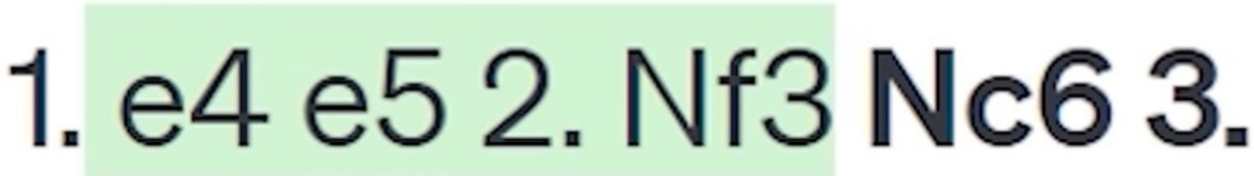
- It is not sufficient to use prompts like:



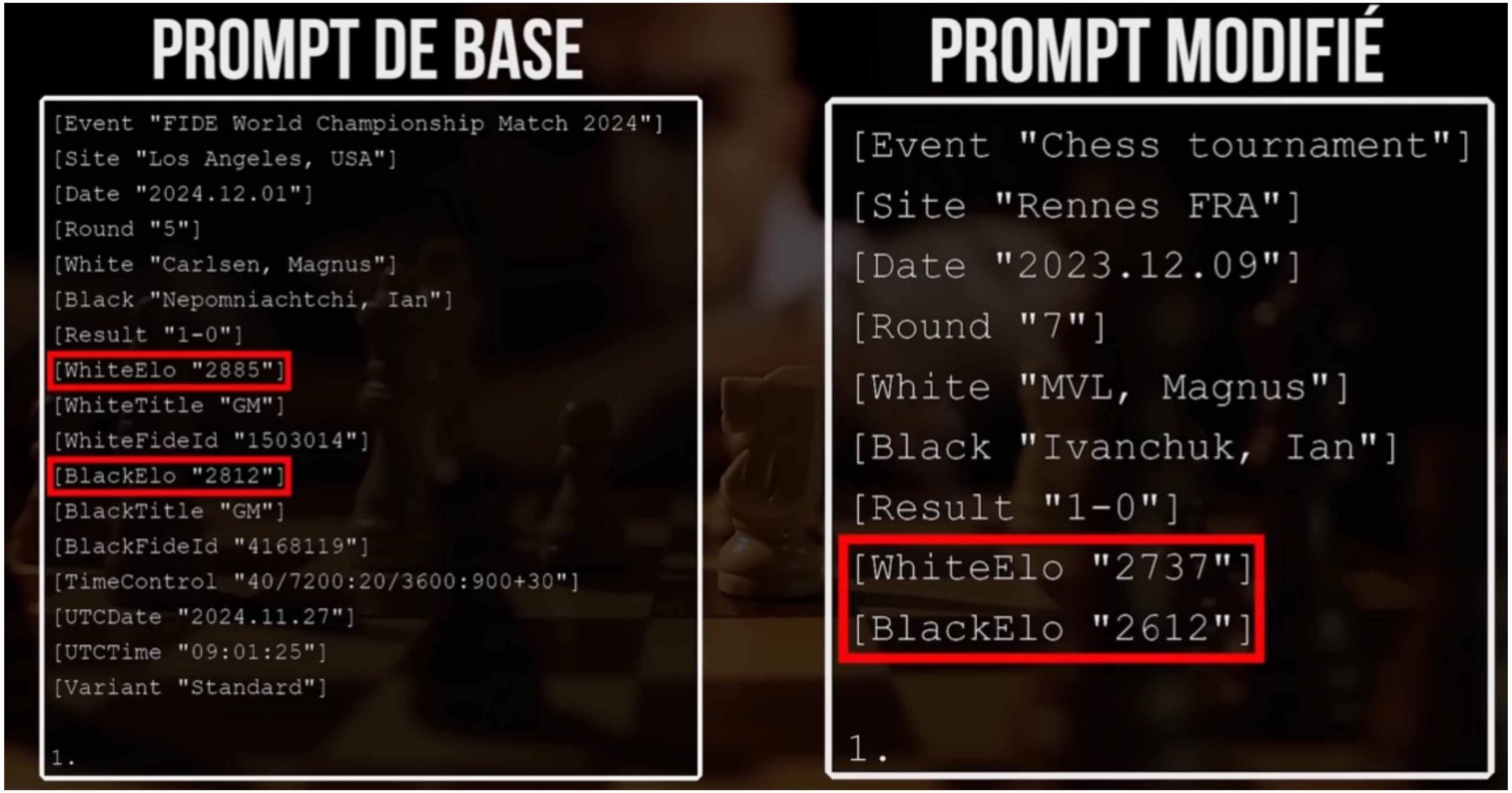
The case of **chess** playing: prompts

- Use gpt-3.5-turbo-instruct and PGN (Portable Game Notation) heading





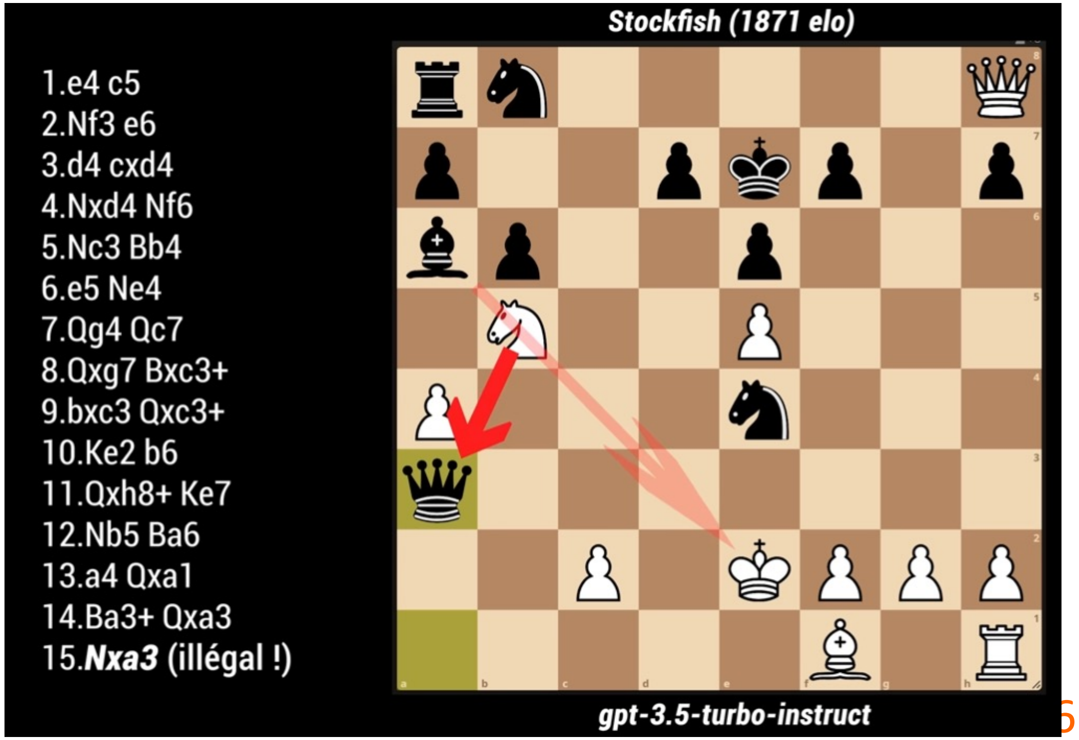
- … makes a small difference here between



Performances

Study by Mathieu Acher

- 439 plays. ~22 000 moves.
- 8 illegal moves (0.04% of all moves)

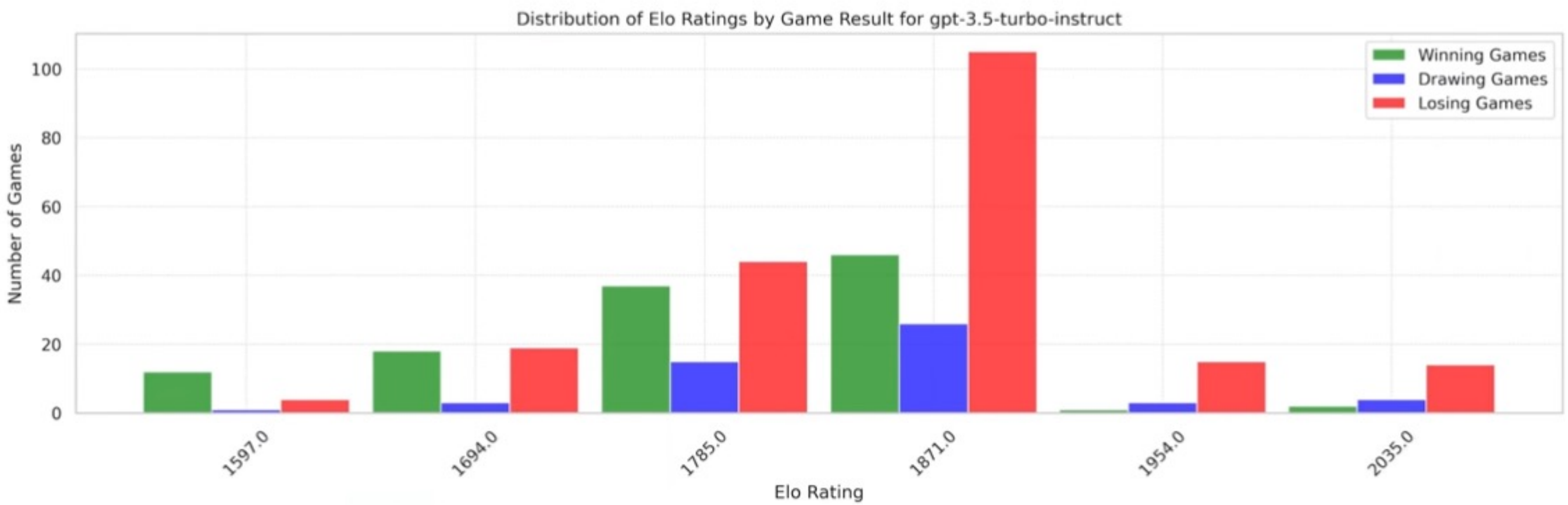


Performances

Study by Mathieu Acher (2023)

- 439 plays. ~22 000 moves.
- 8 illegal moves (0.04% of all moves)
- ~**1743 ELO** rating!!!

`gpt-3.5-turbo-instruct` is capable of winning games against stronger Elo opponents (even more than 2000 Elo!), but it's not that frequent. Here is the distribution of scores against SF at different skills.



- This is a **very impressive** performance!

- But the **rules** of the game were **never given** to the LLM!

Does gpt 3.5-turbo-instruct has a model of the chess world?

And, if yes, which one?

So, does gpt 3.5-turbo-instruct has a model of the chess world?

And, if yes, which one?

How to **approach** these questions?

1. Just a stochastic parrot?

Chess database

– ...As of December, 26th, 2025

lichess.org

open database

Database exports are released under the **Creative Commons CC0 license**.
Use them for research, commercial purpose, publication, anything you like.
You can download, modify and redistribute them, without asking for permission.

CHESS GAMES VARIANTS BROADCASTS PUZZLES EVALUATIONS

7,318,035,930 standard rated games, played on lichess.org, in PGN format. Each file contains the games for one month only; they are not cumulative.

Month	Size	Games	Download
2025 - November	29.4 GB	90,633,152	.pgn.zst / .torrent
2025 - October	29.9 GB	91,549,148	.pgn.zst / .torrent

1. Just a stochastic parrot?

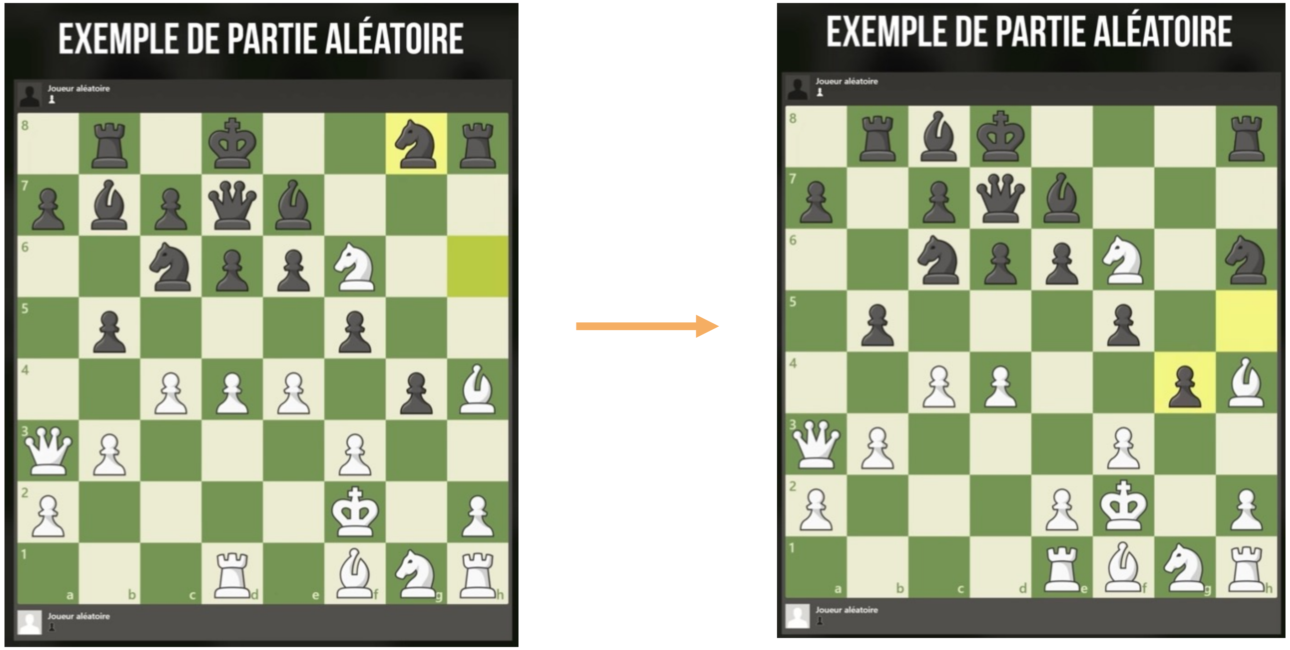
   – The **number of possible moves is astronomical**
     (it is estimated that the number of chess games of interest is ~$\mathbf{10^{120}}$, which is much less than the number of legal games!)

=> no way to play from rote memory

Claude Shannon (1950) “Programming a Computer for Playing Chess”

1. Just a stochastic parrot?

- Let us play *N* **random moves** ... and see how gpt 3.5-turbo-instruct performs



- Conclusion
  - gpt 3.5-turbo-instruct is not **parroting** existing games
  - It just **pursues** the game **in the spirit** of its start
    - If the **moves** were **random**, then the player was a **poor player** and one should **continue playing accordingly**

How to **test** this hypothesis?

Solving chess **puzzles**

So I want to test the model somewhat objectively. So I decided to have it try to solve some tactics puzzles. In some sense this is what I expect should be hardest for the model. (Because, remember, *it's not doing any lookahead it's just predicting the next word.*)

To do this I'll use the Lichess puzzle database, a collection of 3.5 million puzzles from real games in the following format:

You may notice there's one problem. The puzzles only have the current board state (encoded as FEN), not the full PGN history. And the language model is only good when operating on the full game text.

Fortunately though, it *does* have the Lichess game that the puzzle was taken from. And also, fortunately, there is a database of all games played on Lichess. So all I have to do is associate each puzzle with the

Carlini , Nicholas (2023). Playing chess with Large Language Models.
https://nicholas.carlini.com/writing/2023/chess-llm.html.

...

- Ask gpt 3.5-turbo-instruct to continue a game from the same position but with two different histories

From a sequence of plausible moves

From a sequence of implausible moves

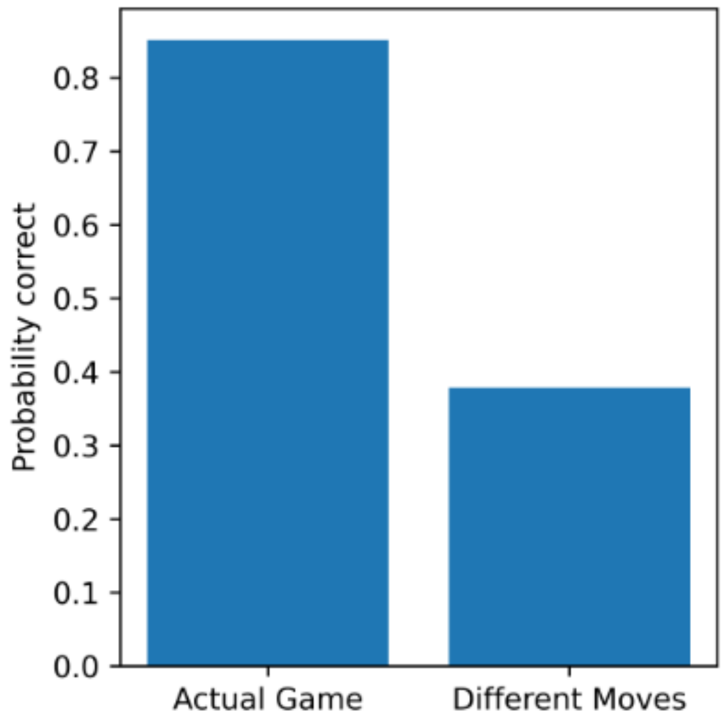
1. Just a stochastic parrot?

– Let us play *N* **random moves** … and see how gpt 3.5-turbo-instruct performs

Now let's ask the following question: how well does the model solve chess positions when when given completely implausible move sequences compared to plausible ones?

As we can see at right it's only half as good! This is very interesting. To the best of my knowledge there aren't any other chess programs that have this same kind of stateful behavior, where *how* you got to this position matters.

This suggests something interesting, too: the model might actually be adapting on-the-fly to the skill of the opponent. If the opponent plays weird moves that don't make sense, it might be more likely to "believe" that this PGN game is between two lower rated players and therefore it should produce opponent moves that are more likely to be played by lower rated players.



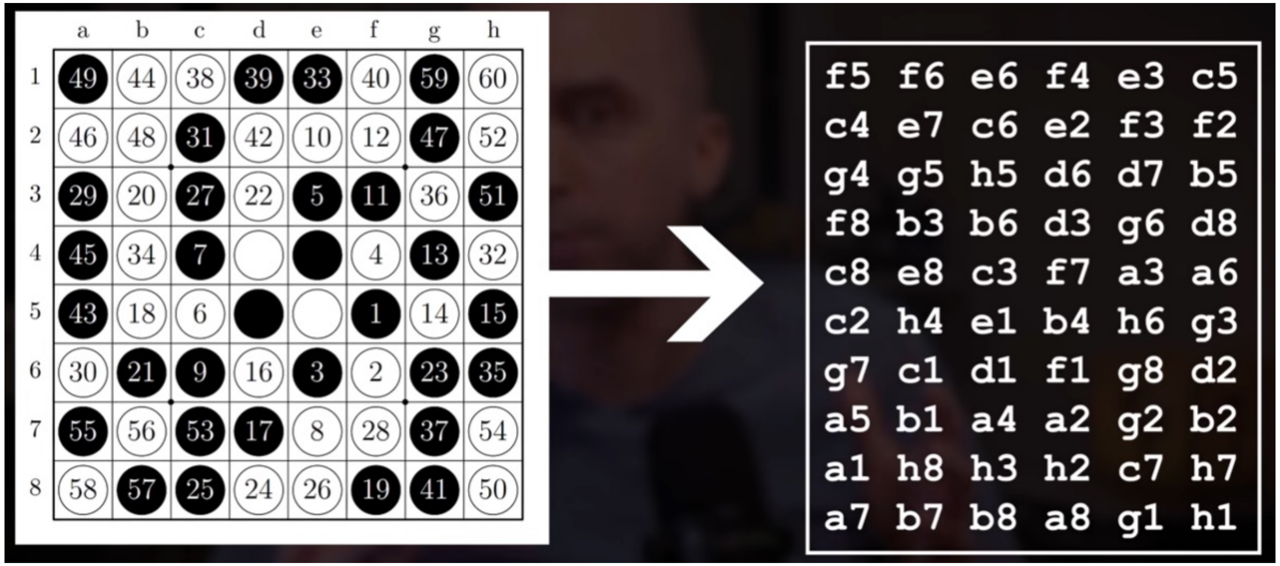
So what does gpt 3.5-turbo-instruct know about chess?

So what does gpt 3.5-turbo-instruct know about chess?

Let us take a simpler neural network: OthelloGPT

2. Ok, but which internal representation?

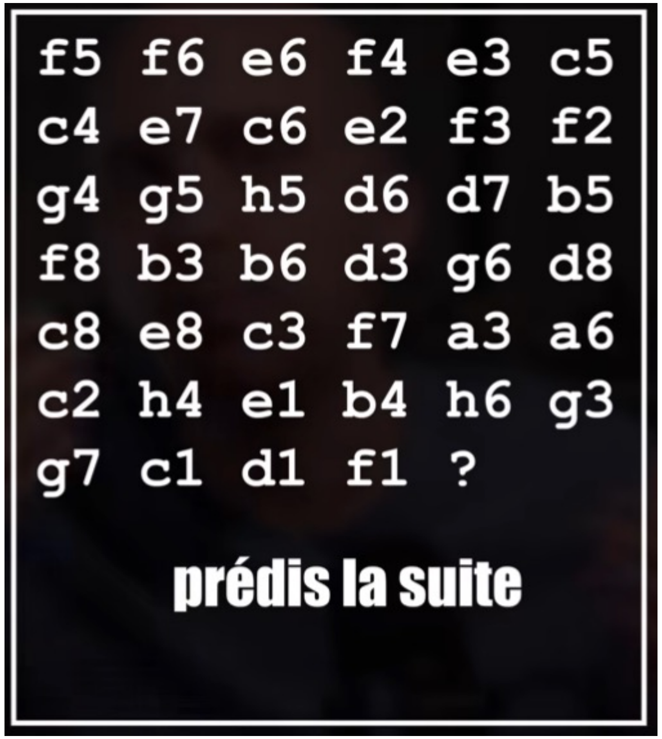
– Let us study a simpler game: **Othello**



Same kind of algebraic **notation**. But simpler.

Kenneth Li et al. (ICLR-2023) "*Emergent World Representations: Exploring a Sequence Model trained on a Synthetic Task*"

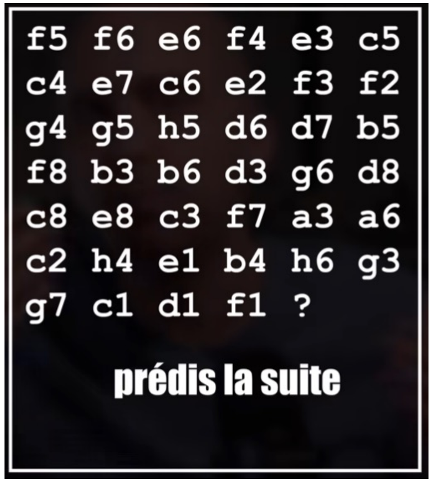
- Ok, but which internal representation?



Kenneth Li et al. (ICLR-2023) “*Emergent World Representations: Exploring a Sequence Model trained on a Synthetic Task*”

As a first step, we train a language model (a GPT variant we call **Othello-GPT**) to extend partial game transcripts (a list of moves made by players) with legal moves. The model has **no a priori knowledge** of the game or its rules. All it sees during training is a series of tokens derived from the game transcripts. Each token represents a tile where players place their discs. Note that we do *not* explicitly train the model to make strategically good moves or to win the game. Nonetheless, our model is able to generate legal Othello moves with high accuracy.

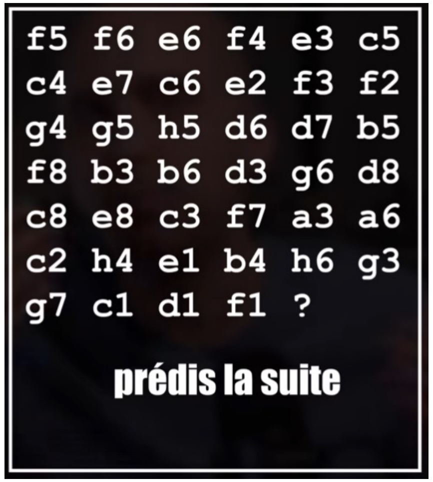
- Ok, but which internal representation?



"Our next step is to look for world representations that might be used by the network. In Othello, the "world" consists of the current board position. "

Kenneth Li et al. (ICLR-2023) "*Emergent World Representations: Exploring a Sequence Model trained on a Synthetic Task*"

- Ok, but which internal representation?



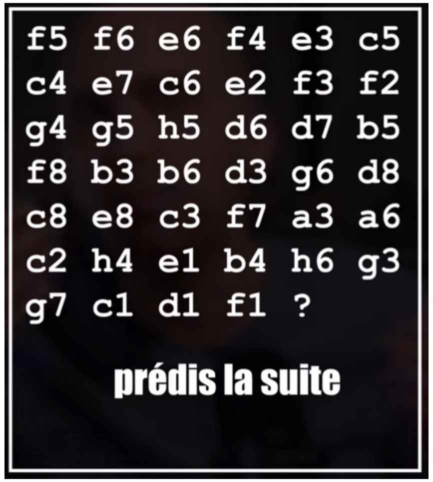
"Our next step is to look for world representations that might be used by the network. In Othello, the "world" consists of the current board position. "

Rk: This is an inexact description.

As we have seen, GPT like models consider first and foremost sequences. But let's go on.

Kenneth Li et al. (ICLR-2023) "*Emergent World Representations: Exploring a Sequence Model trained on a Synthetic Task*"

- Ok, but which internal representation?



"Our next step is to look for world representations that might be used by the network. In Othello, the "world" consists of the current board position. "

Rk: This is an inexact description.

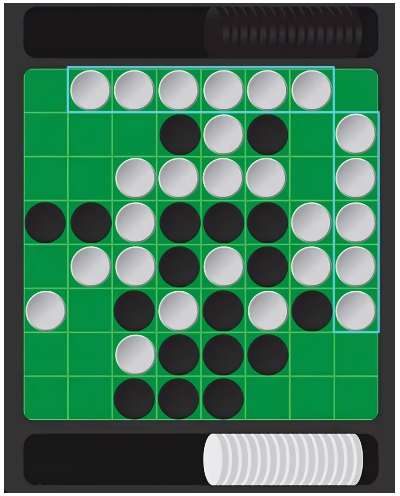
As we have seen, GPT like models consider first and foremost sequences. But let's go on.

"A natural question is whether, within the model, we can identify a **representation** of the **board state** involved in producing its next move predictions."

Kenneth Li et al. (ICLR-2023) "*Emergent World Representations: Exploring a Sequence Model trained on a Synthetic Task*"

- Ok, but which internal representation?

How can we do that?



"A natural question is whether, within the model, we can identify a **representation** of the **board state** involved in producing its next move predictions."

Kenneth Li et al. (ICLR-2023) "*Emergent World Representations: Exploring a Sequence Model trained on a Synthetic Task*"

- Ok, but which internal representation?

- Othello-GPT
  - 8-layer GPT model
  - 8-head attention mechanism
    - 512-dimensional hidden space
  - An trainable word embedding of 60 vectors (one for each free tile)

- Training **data**
  - 140,000 games played by **humans**
  - + 20,000,000 **synthetic** games, uniformly sampling leaves from the Othello game tree, reflecting **no strategy**!

Kenneth Li et al. (ICLR-2023) "*Emergent World Representations: Exploring a Sequence Model trained on a Synthetic Task*"

“We now evaluate how well the model’s predictions adhere to the **rules of Othello**.

For each game in the validation set, which was not seen during training, and for each step in the game, **we ask Othello-GPT to predict the next legal move** conditioned by the partial game before that move.

We then calculate **the error rate** by checking if the top-1 prediction is legal. The error rate is 0.01% for Othello-GPT trained on the synthetic dataset and 5.17% for Othello-GPT trained on the championship dataset. For comparison, the untrained Othello-GPT has an error rate of 93.29%.

The main takeaway is that Othello-GPT does far better than chance in predicting legal moves when trained on both datasets.”

- Ok, but which internal representation?

Probing the internal representation

- Ok, but which internal representation?

Probing the internal representation

- A **probe** is is a classifier (or regressor) whose **input** consists of internal activations of the network, and which is trained to **predict** a feature of interest.

- Here, the probes are trained to **predict** **the board state** from the network's internal activation (here, with 8 layers) function **after** **a given sequence of moves**.

- 60 **probes** (4 cells are occupied at the start of the game)
  - Each one predict the occupation of a cell of the board: **black**, **white** or **empty**
  - From the **internal state** of the neural network on the last hidden layer (here a **512-dimensional** space)

- Ok, but which internal representation?

Probing the internal representation

- A **probe** is is a classifier (or regressor) whose **input** consists of internal activations of the network, and which is trained to **predict** a feature of interest.
- Here, the probes are trained to **predict** **the board state** from the network's internal activation (here, with 8 layers) function **after** **a given sequence of moves**.

	x^1	x^2	x^3	x^4	x^5	x^6	x^7	x^8
Randomized	25.5	25.4	25.5	25.8	26.0	26.2	26.2	26.4
Championship	12.8	10.3	9.5	9.4	9.8	10.5	11.4	12.4
Synthetic	11.3	7.5	4.8	3.4	2.4	1.8	1.7	4.6

Table 2: Error rates (%) of nonlinear probes on randomized Othello-GPT and Othello-GPTs trained on different datasets across different layers. Standard deviations are reported in Appendix H.

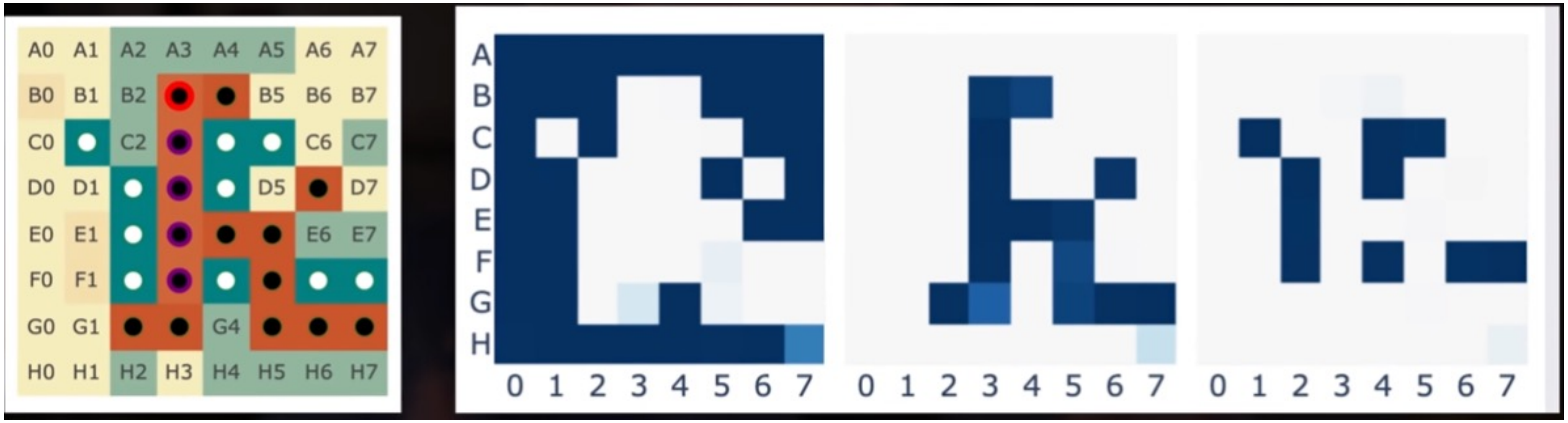
Best error rate at **random** guessing = 52.95% (empty cell)

- Ok, but which internal representation?

Probing the internal representation

So, it appears that Othello-GPT computes information **reflecting the board state**.

Game position | Probability of void tile | Probability of black tile | Probability of white tile



From: Nanda Neel (2023) *"Actually, Othello-GPT has a linear emergent world representation"*

- Clearly **quite remarkable**!
  - The **tokens** are randomly named (e.g. XG103B)
  - At **start**, the probes are random predictors

- The **learned representation** is a mean for the system to be a good predictor
  - This is an **abstract representation** that is structurally equivalent to a model of the game

- Ok, but which internal representation?

Probing the internal representation

So, it appears that Othello-GPT computes information **reflecting the board state**.

But, how to **ensure** it?

- Ok, but which internal representation?

Probing the internal representation

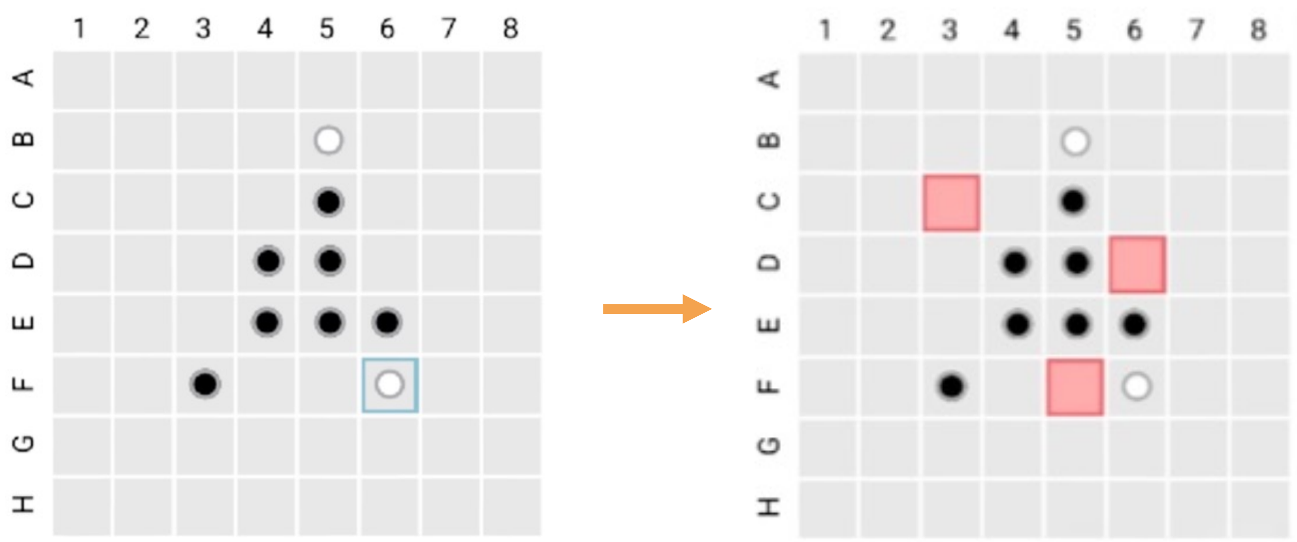
So, it appears that Othello-GPT computes information **reflecting the board state**.

But, how to **ensure** it?

→ Does the internal **representation** have a **causal** relationship with the predicted move?

- Ok, but which internal representation?

Does the internal **representation** have a causal effect?



Internal **representation**

Predicted possible **moves**

Do Large Language Models learn world models or just surface statistics?

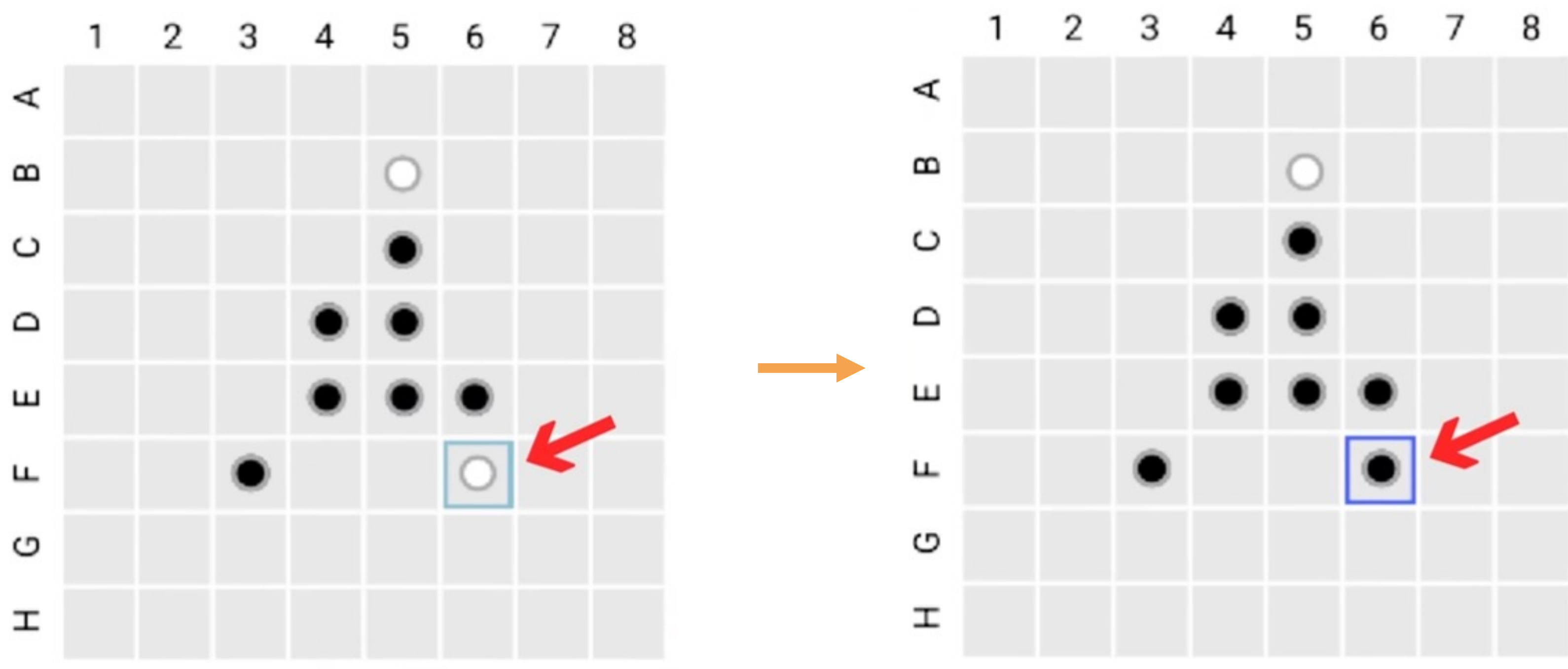
21.JAN.2023 . 15 MIN READ

Kenneth Li

Trying to understand the rich inner world of deep neural networks.

- Ok, but which internal representation?

Does the internal **representation** have a causal effect?

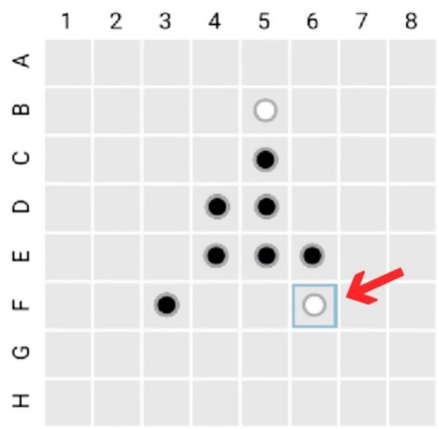


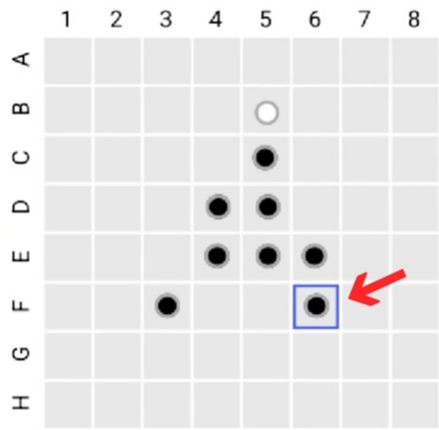
Internal **representation**

Manually modified representation

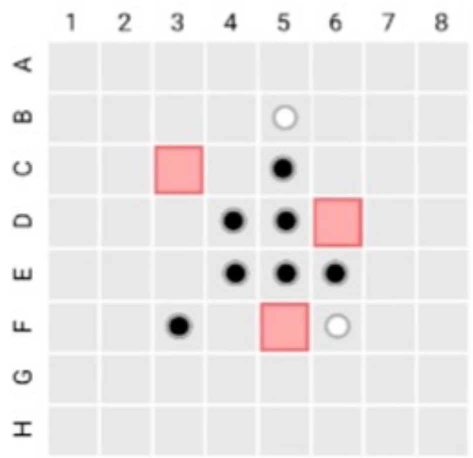
- Ok, but which internal representation?

Does the internal **representation** have a causal effect?





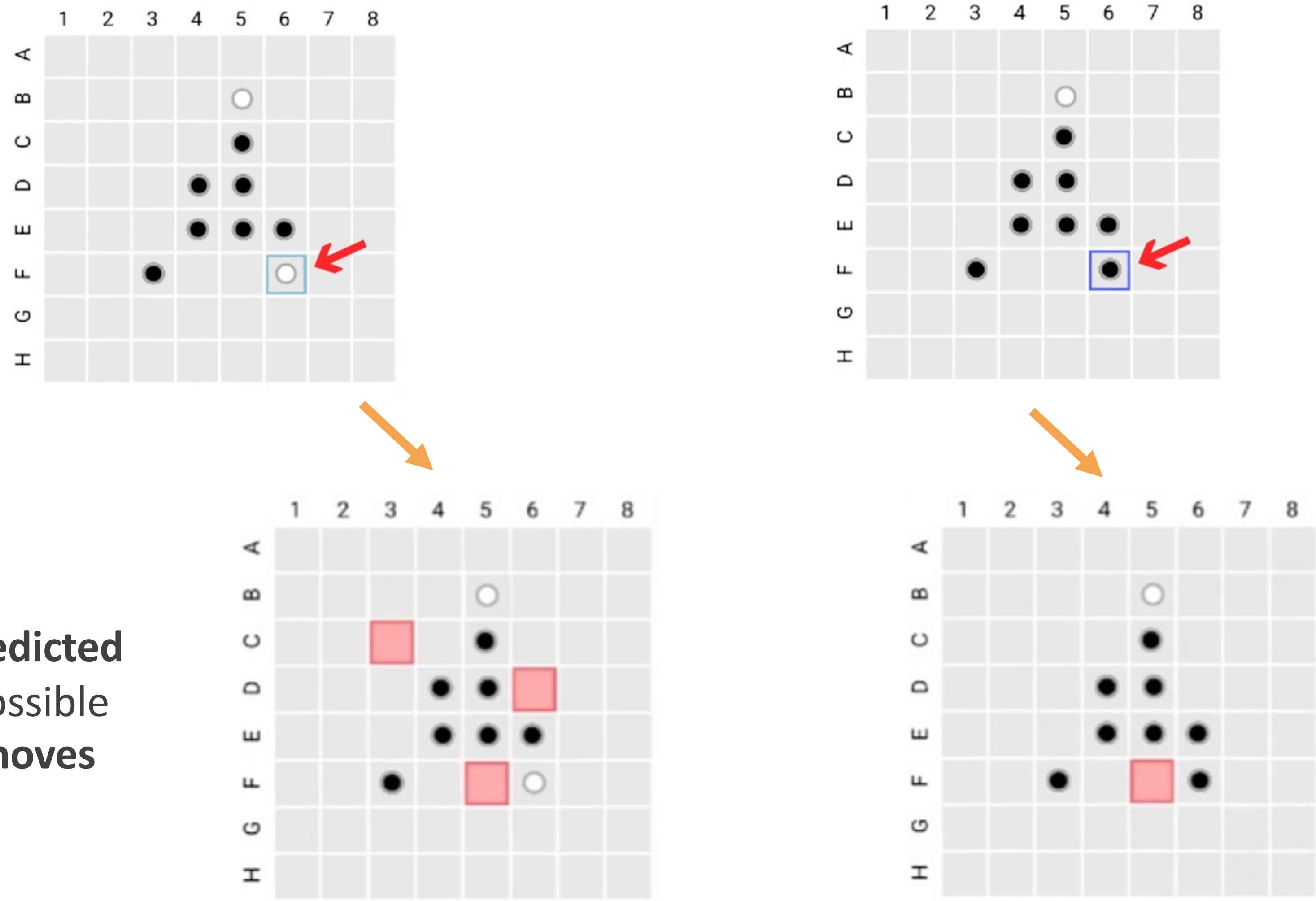
Predicted possible **moves**



?

- Ok, but which internal representation?

Does the internal **representation** have a causal effect?



Predicted
possible
moves

- It seems that Othello-GPT has an internal representation of the game

- Ok, but which internal representation?

in chess?

- Ok, but which internal representation?

in chess?

A **50 million parameter** GPT trained on **5 million games** of chess learns to play at ~1300 Elo in **one day** on **4 RTX 3090 GPUs**.

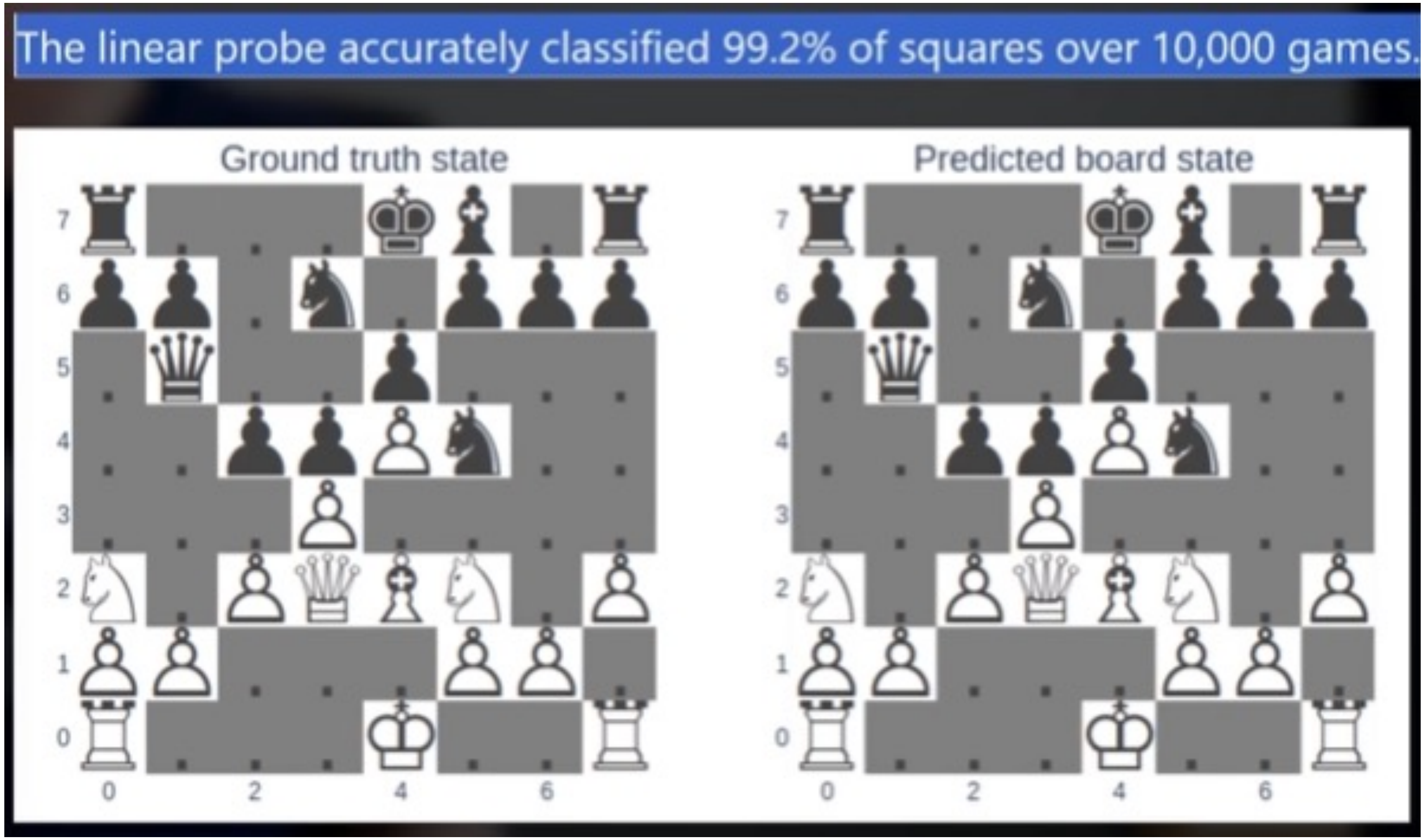
This model is only trained to predict the next character in PGN strings (1.e4 e5 2.Nf3 …) and is never explicitly given the state of the board or the rules of chess.

Despite this, in order to better predict the next character, it learns to **compute the state of the board** at any point of the game, and learns a diverse set of rules, including check, checkmate, castling, en passant, promotion, pinned pieces, etc.

In addition, to better predict the next character it **also learns to estimate** latent variables such as **the Elo rating of the players in** the game.

- Ok, but which internal representation?

in chess?



- Ok, but which internal representation?

in chess?

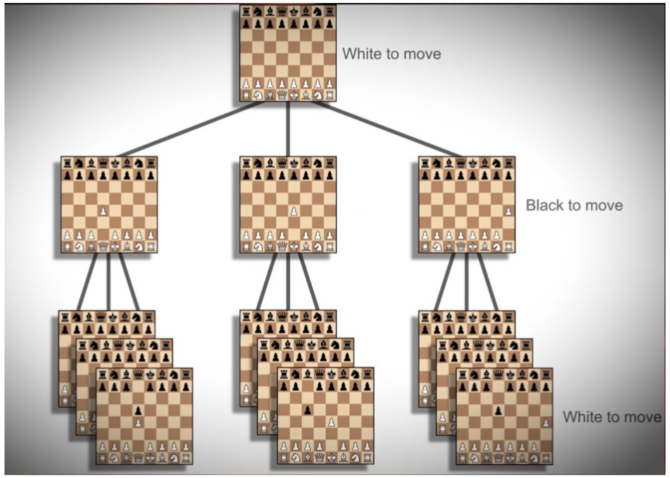
Original Model
Original Board

Intervention

Modified Model
Modified Board

LLMs are different from standard AI game players

- They
  - Consider the **sequence** of moves
    - Not the actual position
      (even though we have seen that …)



  - Do not try to **win**.
    Only to continue the “game” in the same spirit (choreography?)

Conclusion

- LLMs are like **alien** creatures
  - It is not straightforward to conclude whether they know or not to perform some task
    - Inherent **stochasticity**
    - Many **parameters**
  - Requires an empirical approach
    - **Statistical** experiments
    - Exploring the **representations**
      - Looking for causal relationships

Conclusion … No, just a start

- Neel Nanda (https://www.neelnanda.io/mechanistic-interpretability/othello)
  - My interpretation of the original paper was that it was strong evidence for the fact that **it's *possible* for "predict the next token" models** to form world emergent models, despite never having explicit access to the ground truth of the world/board state.
  - At first glance, playing legal moves in Othello (not even playing *good* moves!) has nothing to do with language models, and I think this is a strong claim worth justifying. **Can working on toy tasks like Othello-GPT really help us to reverse-engineer LLMs like GPT-4?** I'm not sure! But I think it's a **plausible bet** worth making.

Conclusion … No, just a start

- Neel Nanda (https://www.neelnanda.io/mechanistic-interpretability/othello)
  - Within this worldview, **what should our research goals be?** Fundamentally, I'm an empiricist - models are hard and confusing, it's easy to trick yourself, and often intuitions can mislead. **The core thing of any research project is getting feedback from reality, and using it to form true beliefs about models**. This can either look like forming explicit hypotheses and testing them, or exploring a model and seeing what you stumble upon, but the fundamental question is **whether you have the potential to be surprised** and to **get feedback** from reality.
  - This means that any project is a trade-off between **tractability** and **relevance** to the end goal.

… and transfer learning?

- Do foundation models learn **universal representation**?
  - ChatGPT 4.0 has an ELO rate of less than 1100
    - Because it is **biased** towards … chating
  - The **next frontier** is to be able to adapt ChatGPT online (**continual learning**)

- **Does** learning to play Othello help to learn to play Chess?

- **Does** learning to play Chess help to learn to play Othello?

Bias induced by the **sequence** of tasks

Some **useful facts** about **LLMs** and **Foundation Models**

The **scaling** hypothesis

- By **increasing** (a lot) the **size** of the systems
  - A **phase transition** will appear
  - Corresponding to a **threshold** in the **capacity** of the systems
- Sizes
  - LeNet.5 (1998) : 1,060 parameters
  - AlexNet (2012) : 62,378,344 parameters
  - AlphaGo (2016) : ~65 * 10^6 parameters
  - GPT1 (2018) : 117 * 10^6 parameters, trained on 10^9 words
  - GPT2 (2019) : 1.5 * 10^9 parameters, trained on 6 to 8 * 10^9 words
  - GPT3 (2020) : 175 * 10^9 parameters, trained on 250 * 10^9 words
  - GPT4, Gemini (2023) : ~ 10^{12} parameters, trained on ??? words

The **scaling** hypothesis

- By **increasing** (a lot) the **size** of the systems
  - A **phase transition** will appear
  - Corresponding to a **threshold** in the **capacity** of the systems
- Training time
  - LeNet.5 (1998) : ???
  - AlexNet (2012) : 5 to 6 days on 2 GPUs
  - AlphaGo (2016) : 40 days on 4 TPU (Tensor Processing Units)
  - GPT1 (2018) : 1 day on 1 GPU (**10** exaflops)
  - GPT2 (2019) : 1 week on 32 TPUv3 (> **1000** exaflops)
  - GPT3 (2020) : ??? (> **314,000** exaflops)
  - GPT4, Gemini (2023) : ???

- Sources of GPT-3
  - 3%   Wikipedia
  - 16%   books
  - **22%**   data base similar to the one used for GPT-2
    - Webpages from Reddit
  - **60%**   general archives from *Common Crawl*

- **Water** consumption

  LI, Pengfei, YANG, Jianyi, ISLAM, Mohammad A., *et al.* Making ai less' thirsty'. *Communications of the ACM*, 2025, vol. 68, no 7, p. 54-61.

  - 1 spoon / query
  - ~ 1 to 2seconds of a shower / day
  - 0.1 one hour of Spotify
  - 0.001 one hour of streaming

- **Energy** consumption
    - YOU, Josh (2025). *How much energy does ChatGPT use?'*. EpochAI*, epoch.ai*.
    - MASLEY, Andy (2025). *Using ChatGPT is not bad for the environment*. The Weird Turn Pro. Andymasley.substack.com
    - MASLEY, Andy (2025). *Why Using ChatGPT is not bad for the environment – a cheat sheet*. The Weird Turn Pro. Andymasley.substack.com
    - O'DONNELL, James and CROWNHART, Casey (2025). *We did the math on AI's energy footprint. Here's the story you haven't heard*. MIT Technology Review, technologyreview.com

  – 0.3 W / query

  – 12,000 Wh / day for an average French household

  – 98% of the energy consumption comes from the generation of videos
    - 15 s. video = 16 km in an electrical vehicle
    - = 3.5 hour of a microwave oven

  – Generating images costs much less

References

- Acher, Mathieu (2023). Debunking the Chessboard: Confronting GPTs Against Chess Engines to Estimate ELO Ratings and Assess Legal Move Abilities. Blog.mathieuacher.com
- Burns, C., Izmailov, P., Kirchner, J. H., Baker, B., Gao, L., Aschenbrenner, L., ... & Wu, J. (2023). Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. arXiv preprint arXiv:2312.09390.
- Carlini , Nicholas (2023). Playing chess with Large Language Models. https://nicholas.carlini.com/writing/2023/chess-llm.html.
- Karvonen, Adam (2024). “Chess-GPT’s Internal World Model”. https://adamkarvonen.github.io/machine_learning/2024/01/03/chess-world-models.html
- Karvonen, Adam (2024). "Emergent world models and latent variable estimation in chess-playing language models." arXiv preprint arXiv:2403.15498 (2024).
- Li, Kenneth. "Do Large Language Models learn world models or just surface statistics?." The Gradient (2023).
- Li, K., Hopkins, A. K., Bau, D., Viégas, F., Pfister, H., & Wattenberg, M. (2022). Emergent world representations: Exploring a sequence model trained on a synthetic task. arXiv preprint arXiv:2210.13382.
- Nanda, Neel. "Actually, othello-GPT has a linear emergent world representation." Neel Nanda’s Blog 8 (2023).
- RUOSS, Anian, DELÉTANG, Grégoire, MEDAPATI, Sourabh, et al. Grandmaster-level chess without search. CoRR, 2024.