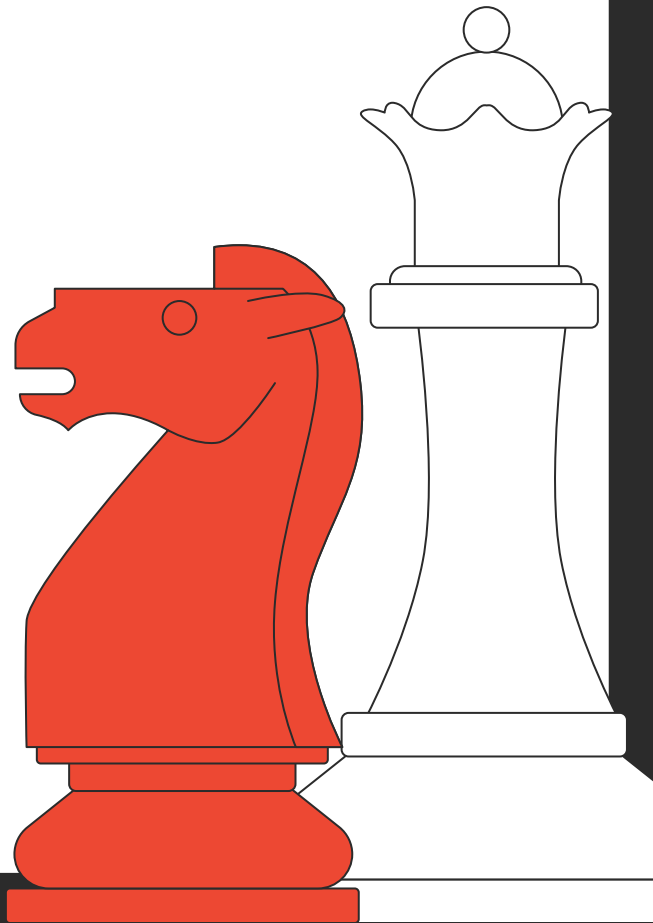
Probing a ChessLLM

ANLP - Project Presentation

Silvano Vento Maddonni



Introduction

Idea:

→ Take a LLM and explore how it represents the world state underlying the text in a task not (directly) related to NLP.

→ Chess

- Chess provides a simple, constrained, and deterministic domain where the exact world state is known.
- Chess games can also be transcribed exactly and unambiguously using chess notations
- The form of chess notations allows us to probe our language models for aspects of the board.

Chess as a Testbed for Language Model State Tracking

Shubham Toshniwal[1], **Sam Wiseman**[2], **Karen Livescu**[1], **Kevin Gimpel**[1]

[1]Toyota Technological Institute at Chicago
[2]Duke University
{shtoshni, klivescu, kgimpel}@ttic.edu, swiseman@cs.duke.edu

Watching a Language Model Learning Chess

Andreas Stöckl
University of Applied Sciences Upper Austria
Digital Media Department
Hagenberg, Austria
andreas.stoeckl@fh-hagenberg.at

Abstract

analyse how a transformer-based language
el learns the rules of chess from text data
orded games. We show how it is possible
stigate how the model capacity and the
e number of training data influence the
success of a language model with the
ess-specific metrics. With these met-
how that more games used for train-
udied range offers significantly bet-
or the same training time. How-
size does not show such a clear
s also interesting to observe that
ation metrics for language mod-
accuracy and perplexity, give
f this here. Further examina-
odels reveals how they store
t board state in the activa-
roups, and how the over-
ious moves influen-
ves.

model capacity is able to learn the rules of arith-
metic to a certain degree by training it with the
data of crawled websites. In the process, elemen-
tary operations were learned in a certain number
space, but not beyond. Is this limitation due to the
lack of capacity of the model, insufficient training
time or training data that did not contain sufficient
information?

In Nogueira et al. (2021), it was demonstrated
that regardless of the number of parameters and
training examples, *Transformer* Vaswani et al.
(2017) models are unable to learn addition rules
that are independent of the length of the num
seen during training.

To test the ability of
rules, and ass

strides
he com-
scertain
state un-
er the task
ike natural
trained, and
at the appro-
y probing the
robing-related
ing data,
pieces and
rained solely
providing acc
can yield signi
nsformer language
re game history i.e.
attention results in a
ose this testbed as a
elopment and analysis

tion

age models have stretched
the simple self-supervised
becoming a fixture in state
s (Vaswani et al. 2017; De-

language modeling objective or introducing any ne
fiers.[1]

Due to the simplicity and precision of
uate language model predictions at
than merely comparing the
ample, even if the ne
ground truth m
is legal gi
be

ers
mo
train

Under review as a conference paper at COLM 2024

Emergent World Models and Latent Variable Estimation in Chess-Playing Language Models

Adam Karvonen
Independent
adam.karvonen@gmail.com

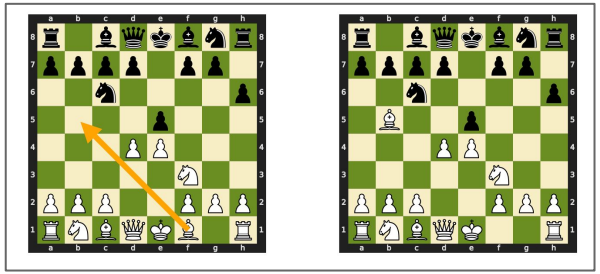
Abstract

Language models have shown unprecedented capabilities, sparking debate
over the source of their performance. Is it merely the outcome of learning
syntactic patterns and surface level statistics, or do they extract semantics
and a world model from the text? Prior work by Li et al. investigated this
by training a GPT model on synthetic, randomly generated Othello games
and found that the model learned an internal representation of the board
d found this work into the more complex domain of chess, training
extend this work into the more complex domain of chess, training
and investigating our model's internal representations using
ontrastive activations. The model is given no a prio
nd is solely trained on next character predicti
representations of board state. We valid
g them to make interventions o
ard state. Unlike Li et al's
the model also le
t the next c
impr

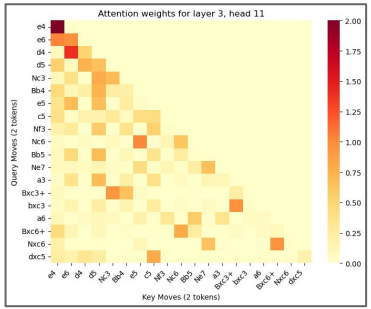
The project

The project is divided in three parts:

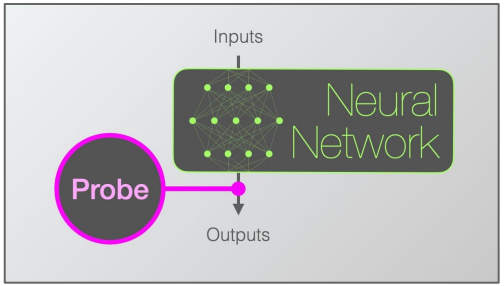
- Part 1: LLM next move prediction + legal moves



- Part 2: Attention Heatmaps

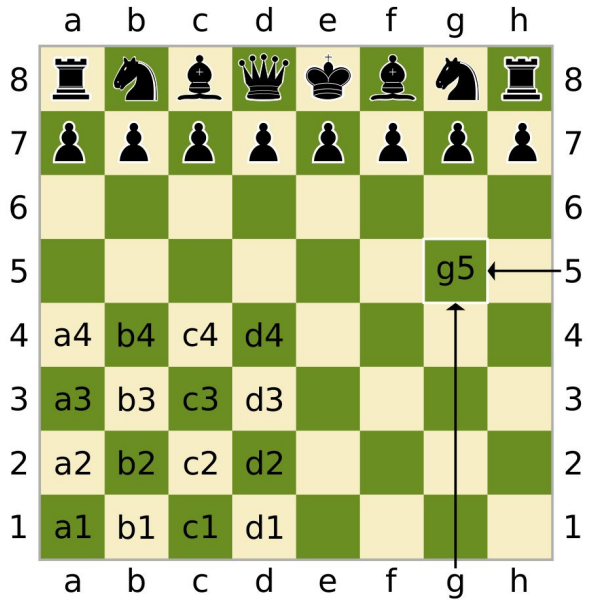


- Part 3: Probing Classifier for board pieces locations



Chess as text

- Chess games and moves can be transcribed in many different ways.
- Algebraic notation is the standard:



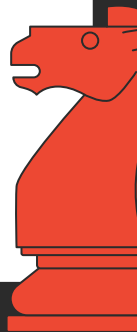
PGN vs UCI

- The sequence of moves composing a game are generally encapsulated using the Portable Game Notation (**PGN**).

```
1. e4 e5 2. Nf3 Nc6 3. Bb5 a6 4. Ba4 Nf6 5. O-O
    Be7 6. Re1 b5 7. Bb3 d6 8. c3 O-O 9. h3 Nb8
    10. d4 Nbd7 11. c4 c6 12. cxb5 axb5 13. Nc3
```

- We represent moves using Universal Chess Interface (**UCI**) notation, which combines the starting square and the destination square to represent a move.

```
e2e4 e7e5 g1f3 b8c6 f1b5 a7a6 b5a4 g8f6 e1g1
f8e7 f1e1 b7b5 a4b3 d7d6 c2c3 e8g8 h2h3 c6b8
d2d4 b8d7 c3c4 c7c6 c4b5 a6b5 b1c3
```



Tokenizer

- The games represented in UCI notation are tokenized using a simple regular expression based tokenizer, which considers a board square symbol such as b1 as a single token.
- This produces a vocabulary of 77 token types, which includes the 64 squares, piece type symbols, and other special symbols.

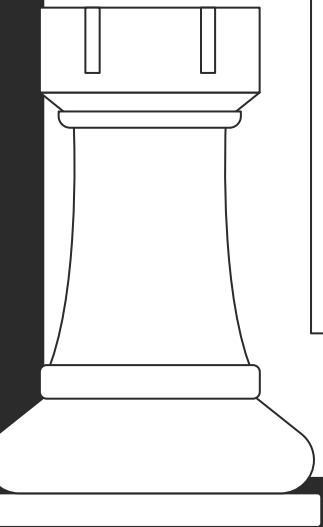
Type	Examples	Count
Square names	e4, d1	64
Piece type	P, K, Q, R, B, N	6
Promoted	q, r, b, n	4
Special symbols	BOS, EOS, PAD	3
Total		77

Model Vocabulary

Next Move Prediction + Legal Moves

- The starting model used is a GPT2-small. 12-layers, 12 heads, embedding 768.
- The model is pre-trained on 250K chess games taken from a full dataset of 2.9 million quality chess games. The whole training was done using UCI notation.

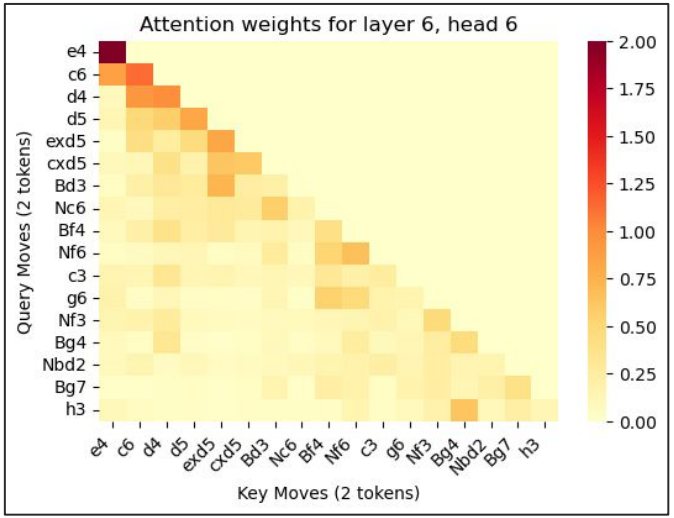
```
PGN seq:
1.Nf3 f5 2.b4 Nf6 3.Bb2 e6 4.b5 d5 5.g3 Bd7 6.a4
a6 7.c4 dxc4 8.Nc3 axb5 9.axb5
UCI seq:
g1f3 f7f5 b2b4 g8f6 c1b2 e7e6 b4b5 d7d5 g2g3
c8d7 a2a4 a7a6 c2c4 d5c4 b1c3 a6b5 a4b5
LM plays: a8a1
(In the game it was: a8a1)
```



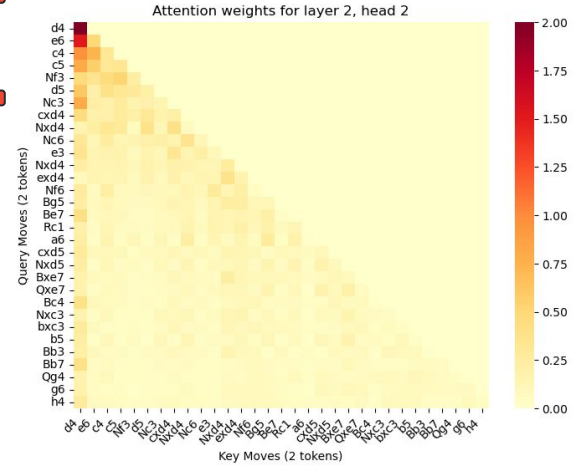
- Legal moves → 96% ratio.

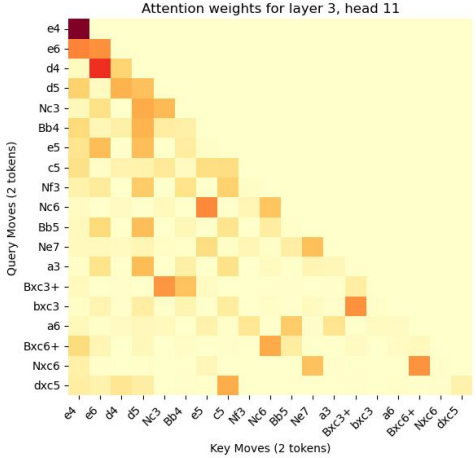
Attention Heatmaps

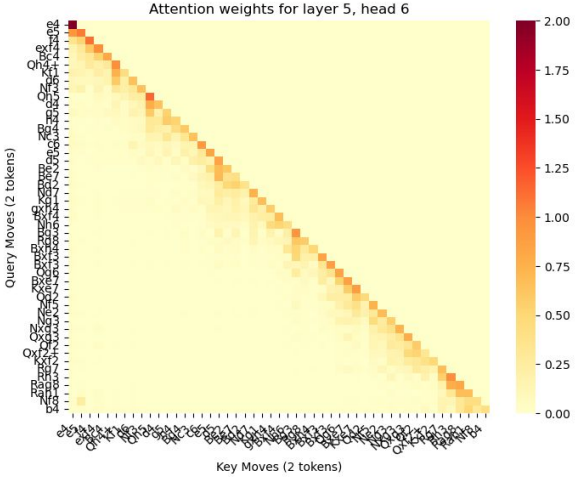
- Attention heatmaps are visual representations of where the neural network focuses its attention when processing input data.

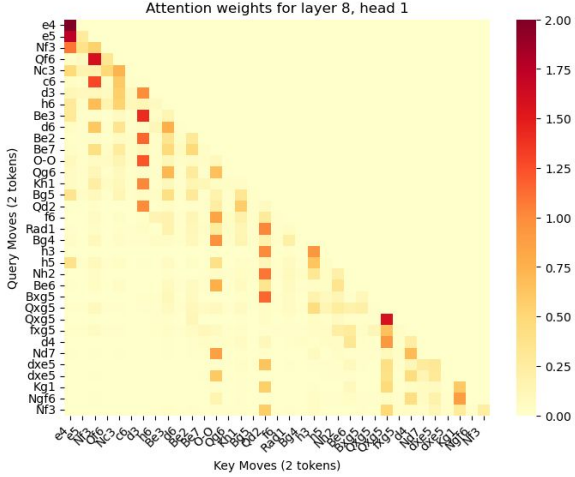


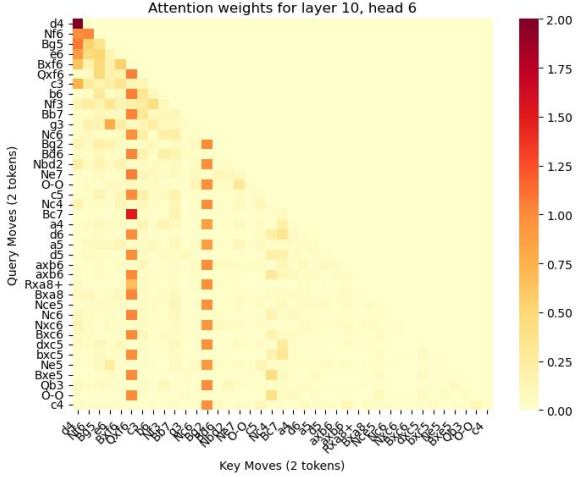
- What it is interesting to look for:
  - Patterns of attention (e.g., attention to recent moves, or to specific types of pieces).
  - Changes in attention patterns across different layers.
  - Any unexpected or interesting focus of attention that might give insights into how the model is processing the chess game.

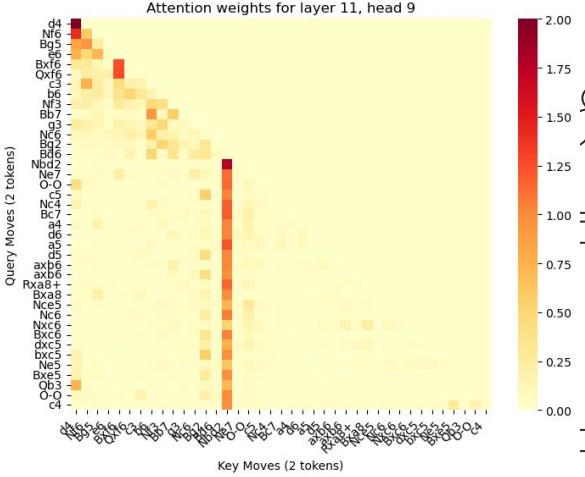

Attention weights for layer 2, head 2
Query Moves (2 tokens)
Key Moves (2 tokens)

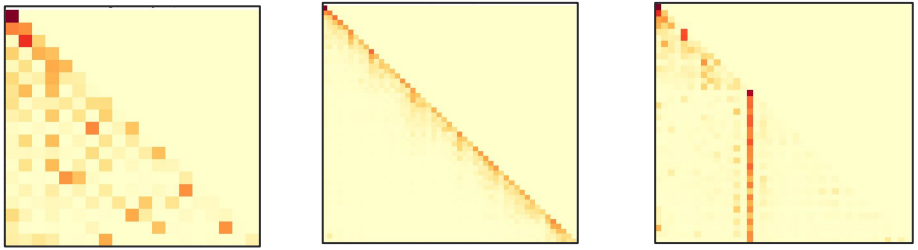

Attention weights for layer 3, head 11
Query Moves (2 tokens)
Key Moves (2 tokens)


Attention weights for layer 5, head 6
Query Moves (2 tokens)
Key Moves (2 tokens)


Attention weights for layer 8, head 1
Query Moves (2 tokens)
Key Moves (2 tokens)


Attention weights for layer 10, head 6
Query Moves (2 tokens)
Key Moves (2 tokens)


Attention weights for layer 11, head 9
Query Moves (2 tokens)
Key Moves (2 tokens)



Final Prediction = [1 to 4] + [5 to 7] + [8 to 12]

[layers 1-4] = General and Sparse

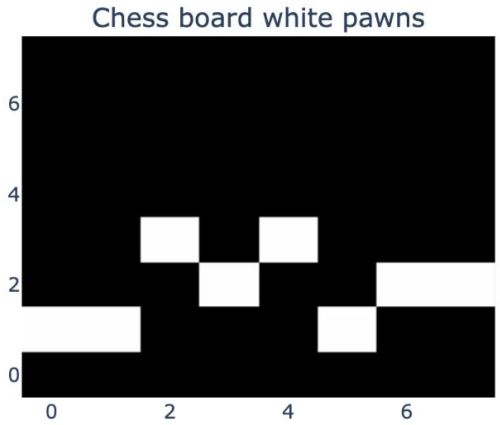
[layers 5-7] = Reinforced Diagonal
(local connectivity of moves)

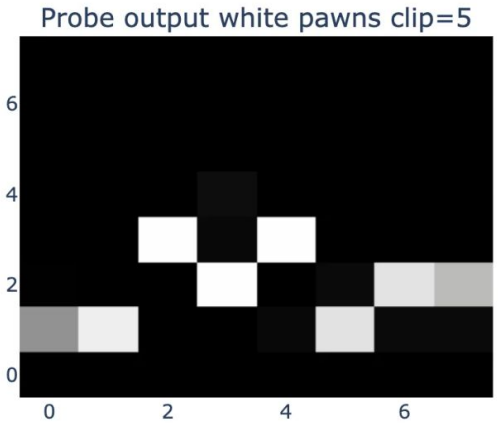
[layers 8-12] = Single moves attention

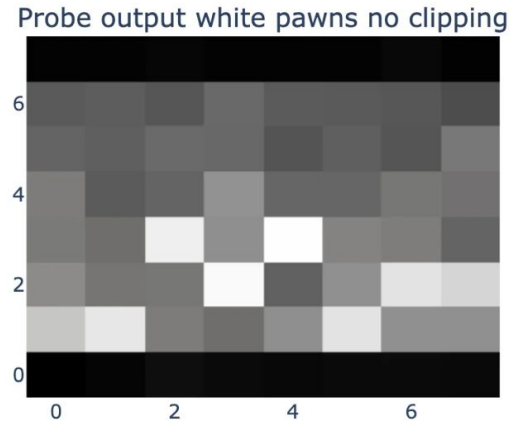
Internal Model World Representation

- Investigate the internal representations of language models in a more constrained setting.

- Train linear probes that recover the internal board state,







Internal Model World Representation

PiecePositionProbe:

- 768x64 linear classifier model with a sigmoid that outputs a 64 elements tensor with probabilities for each square.
- We take the model with 'frozen' weights and extract the features from a particular layer (often the last hidden layer).
- Then, we train the linear classifier on top of these features to see how well it can perform on this task.

Piece	Base	Trained Probe
Pawn	0.54	0.82
Rook	0.55	0.94
Knight	0.51	0.97
King	0.45	0.93

Classifier Probe

Thanks!

Silvano Vento Maddonni