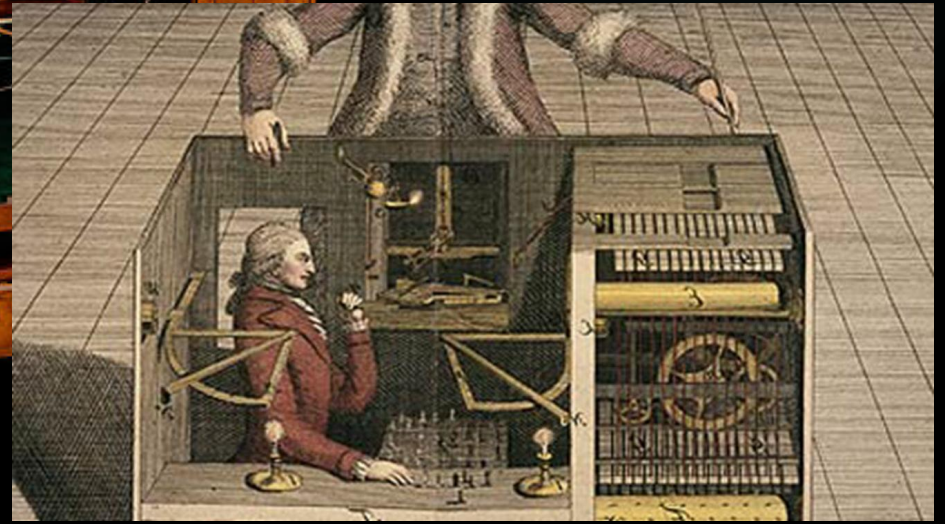


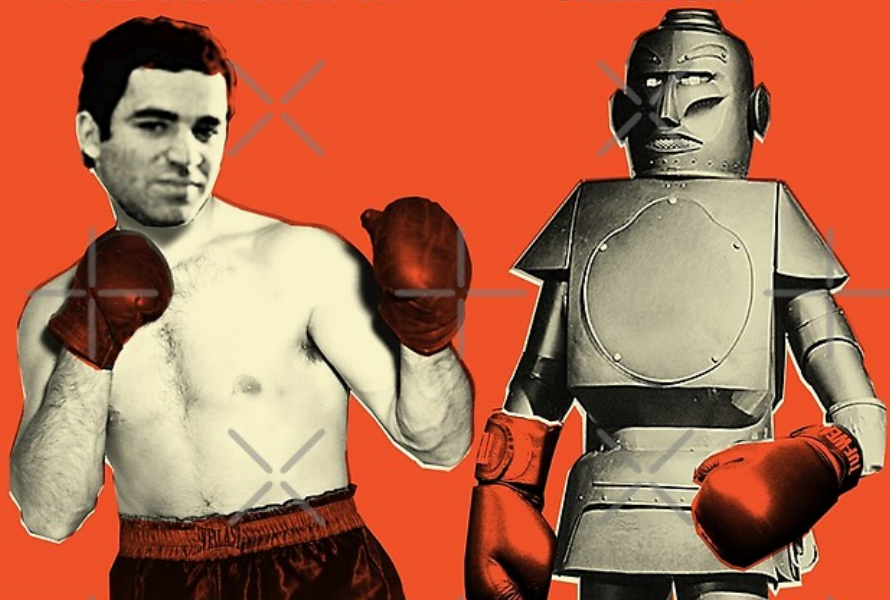
# Detecting fair play violations in chess using Neural networks

ZURA KEVANISHVILI



WORLD'S HEAVYWEIGHT CHAMPIONSHIP

GARRY  
**KASPAROV** vs. **DEEPBLUE**  
WORLD CHESS CHAMPION — CHALLENGER —



**FEB. 10th 1996**  
PHILADELPHIA



**MAY. 11th 1997**  
NEW YORK CITY



 **Magnus Carlsen** ✓  
@MagnusCarlsen


I've withdrawn from the tournament. I've enjoyed playing in the @STLChessClub, and I'll be back in the future



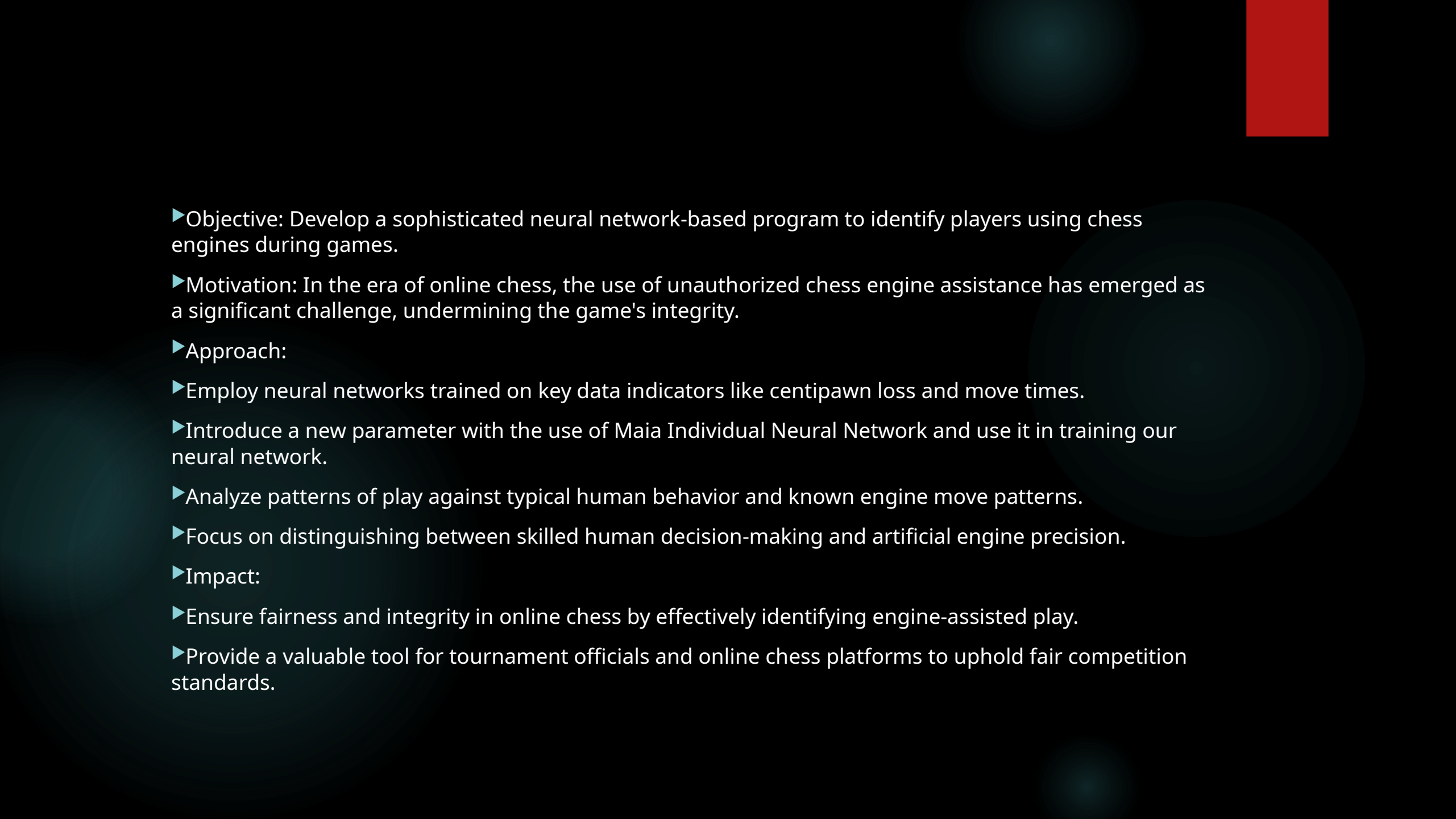
youtube.com  
If I speak I am in big  
Twitter-[https://twitter/](https://twitter.com/STLChessClub)

3:00 PM · Sep 5, 2022 · Twitter for Android

**Musk** ✓ @elonmusk · 9m  
"I've lent hits a target no one else can hit, genius is a target no one can see (cause it's in ur butt)"  
Schopenhauer

 **Babble** @Babble\_\_\_\_ · 18h  
Currently obsessed with the notion that Hans Niemann has been cheating at the Sinquefeld Cup chess tournament using wireless anal beads that vibrate him the correct moves.



- 
- ▶Objective: Develop a sophisticated neural network-based program to identify players using chess engines during games.
  - ▶Motivation: In the era of online chess, the use of unauthorized chess engine assistance has emerged as a significant challenge, undermining the game's integrity.
  - ▶Approach:
    - ▶Employ neural networks trained on key data indicators like centipawn loss and move times.
    - ▶Introduce a new parameter with the use of Maia Individual Neural Network and use it in training our neural network.
    - ▶Analyze patterns of play against typical human behavior and known engine move patterns.
    - ▶Focus on distinguishing between skilled human decision-making and artificial engine precision.
  - ▶Impact:
    - ▶Ensure fairness and integrity in online chess by effectively identifying engine-assisted play.
    - ▶Provide a valuable tool for tournament officials and online chess platforms to uphold fair competition standards.

# Recent breakthroughs in the Hans Niemann case

## How I Found Perfect Correlation between Chess Player Rating and ACPL and STDCPL

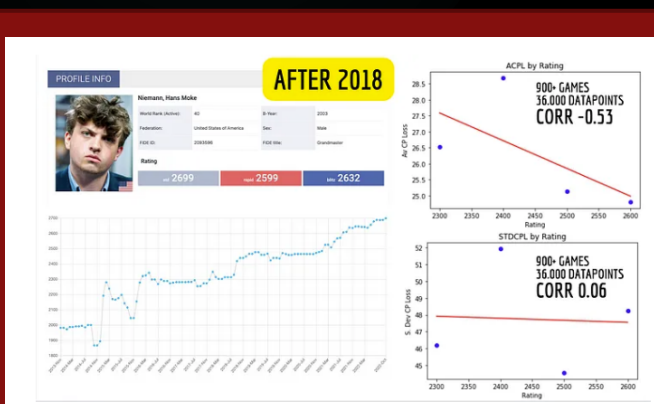
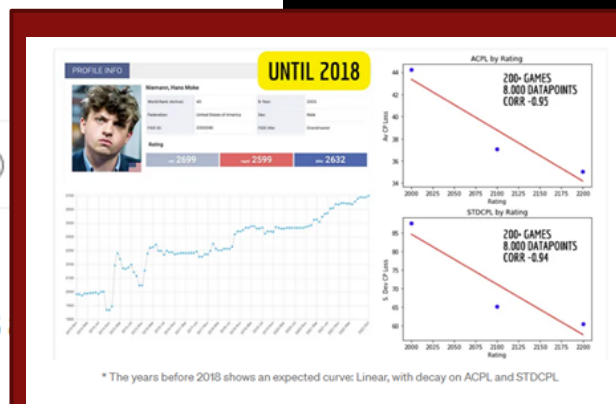
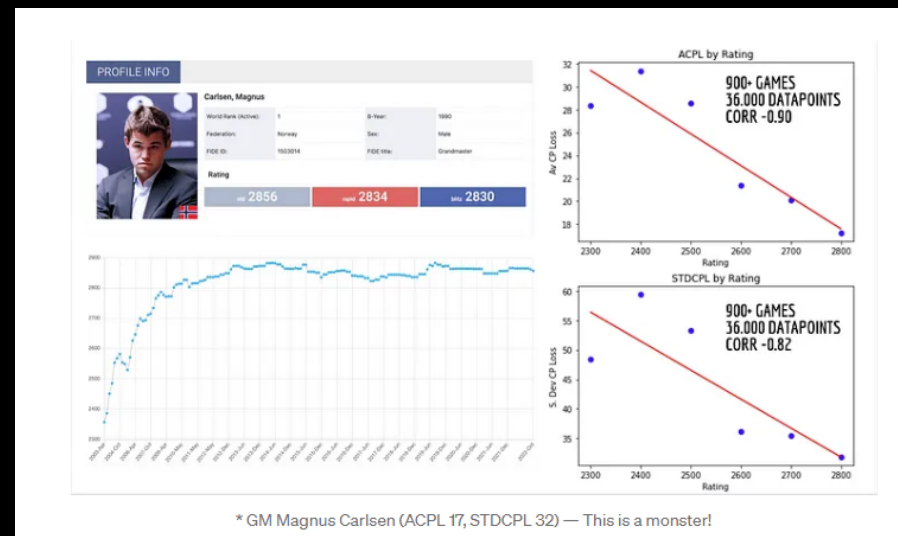


Rafael Leite · Follow

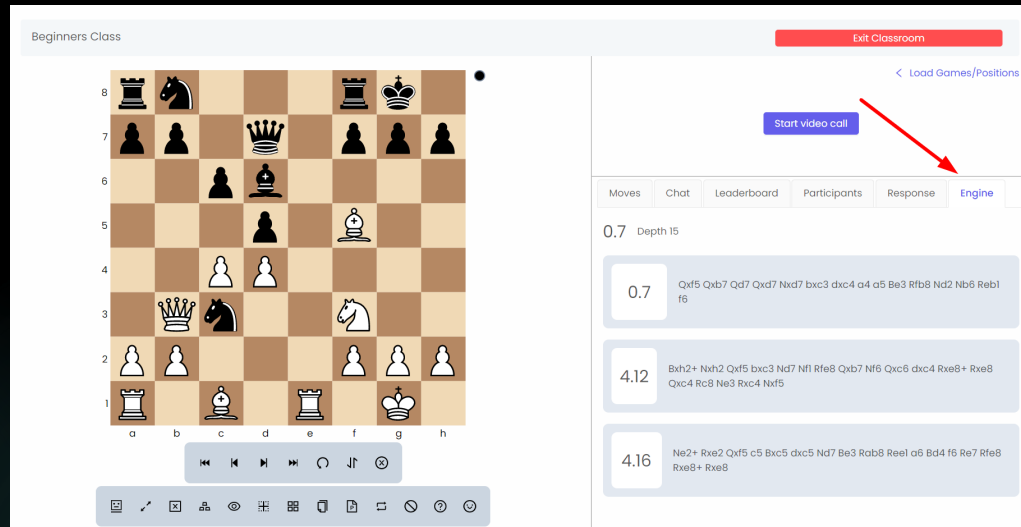
8 min read · Oct 4, 2022

648 18

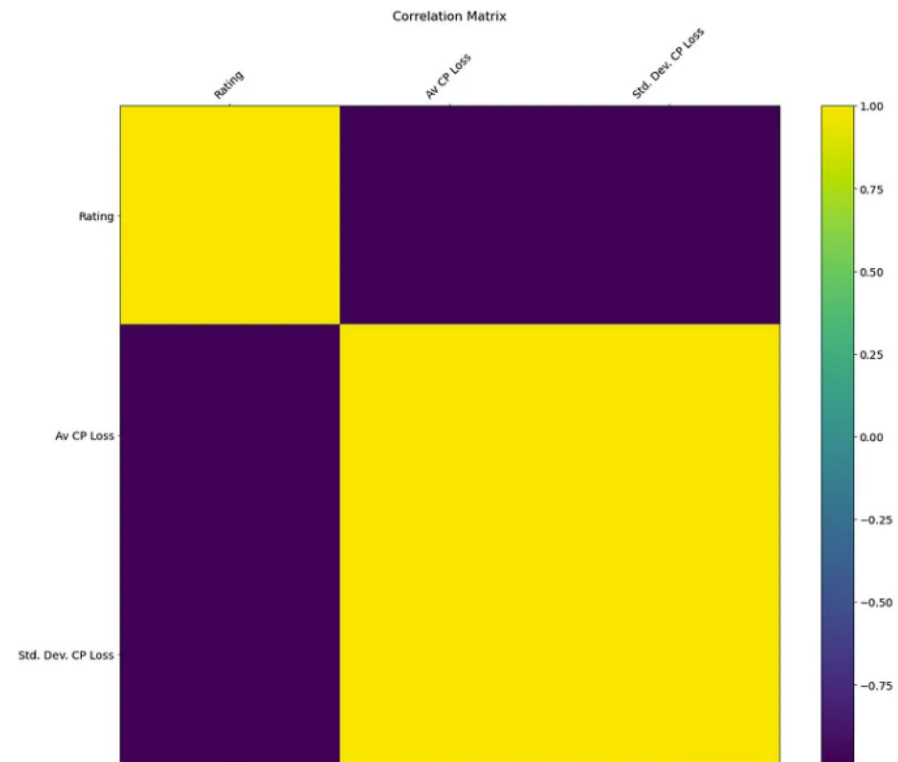
Also, how I found out that Chess Player Hans Niemann has 2500–2550 Strength, even being Rated near 2700



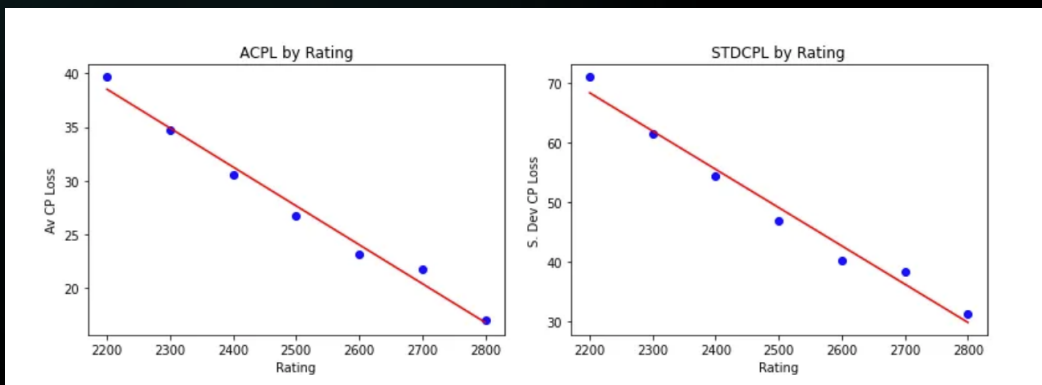
# What is Centi-Pawn Loss?



Correlation Matrix between Rating, ACPL and STDCPLOSS			
Rating	1	-0,992634	-0,988607
Av CP Loss	-0,992634	1	0,9988
Std, Dev, CP Loss	-0,988607	0,9988	1



\* Pearson correlation of -0.99 between Rating and ACPL and between Rating and STDCP

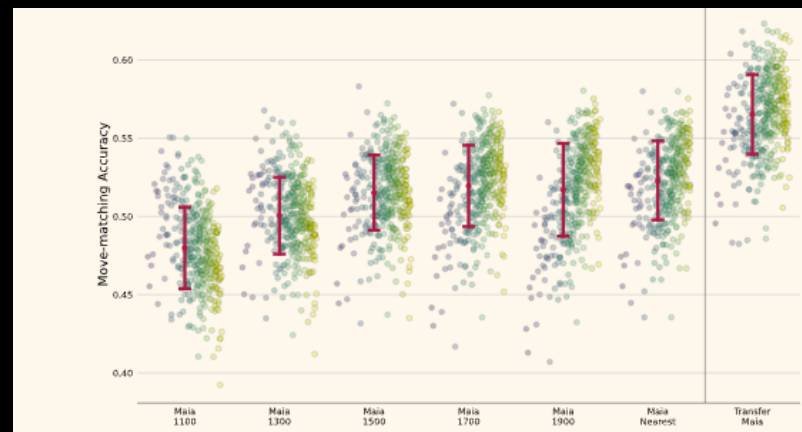
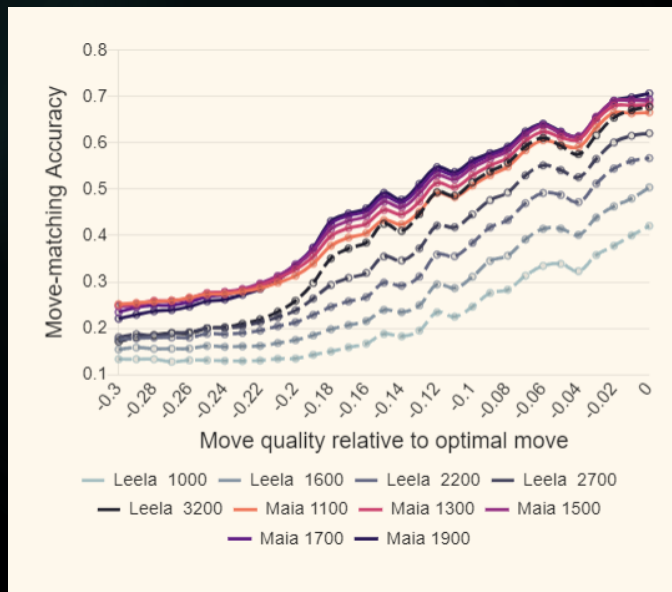
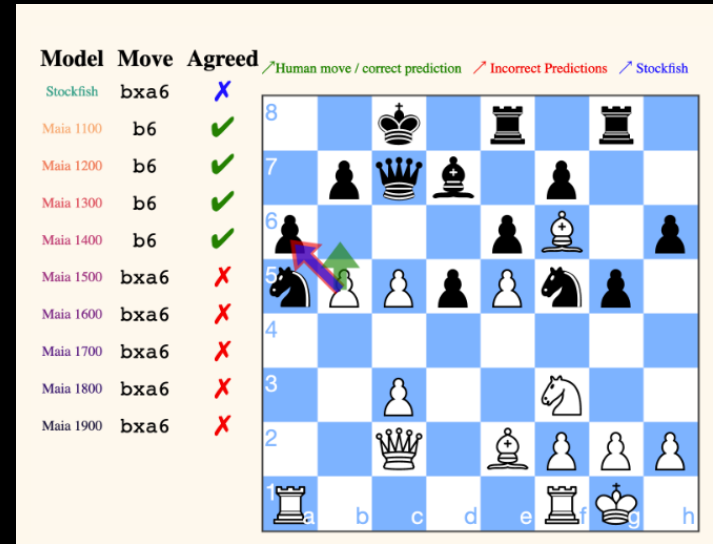
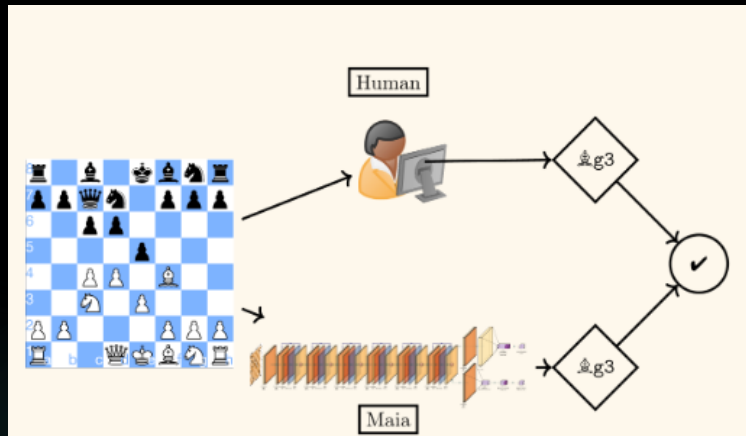




Correlation with Engine



# Further investigation Human factor using Maia





Correlation with  
Individual human  
Style



Correlation with  
Engine



ULTRABULLET  
1648 🏆 2 517 games

BULLET  
2100 🏆 40 2,441 games

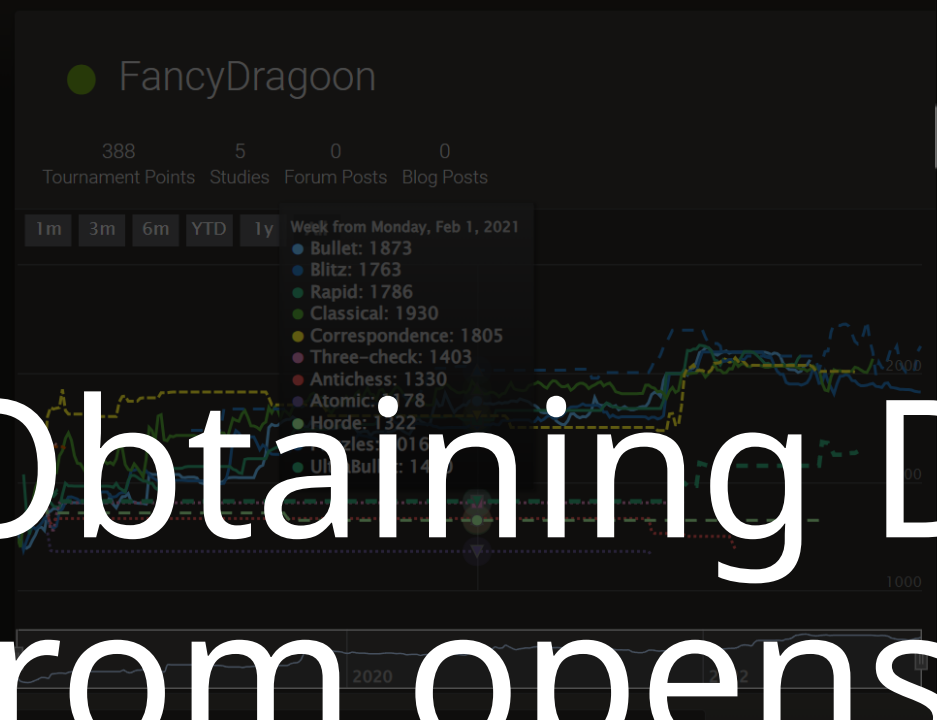
BLITZ  
1916 🏆 5 2,696 games  
Rank: 103,171

RAPID  
2100 🏆 29 906 games

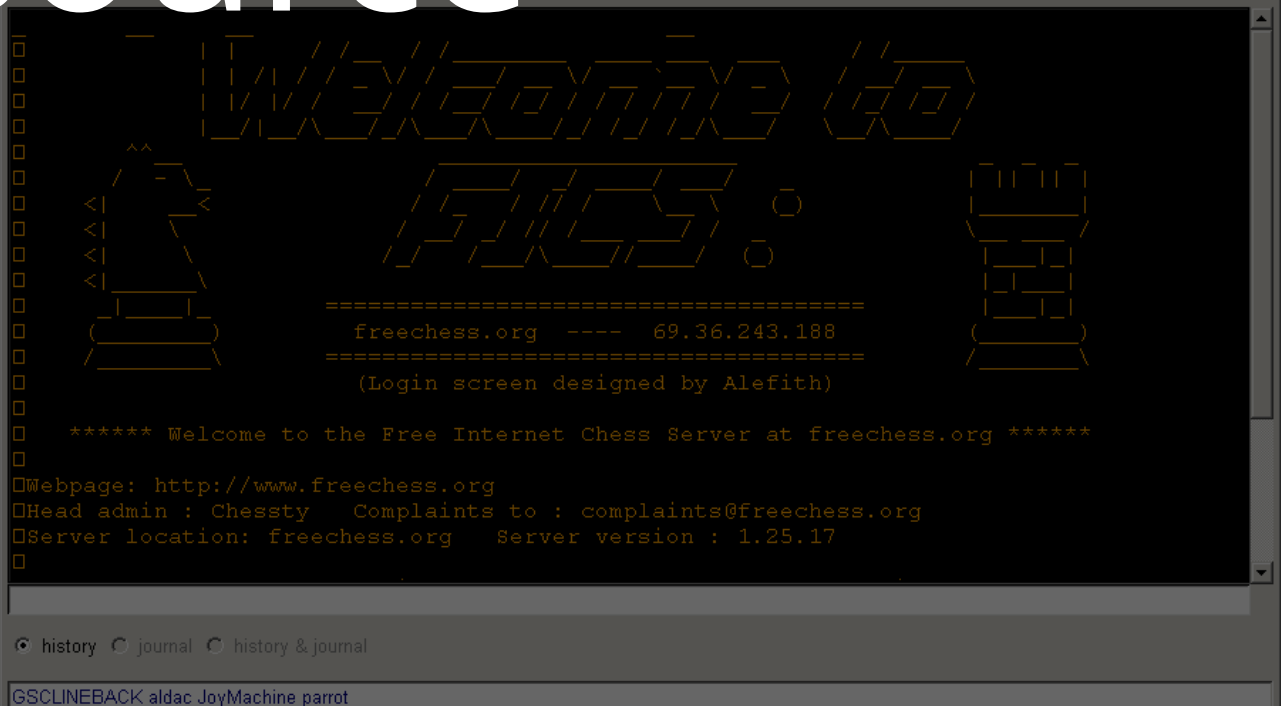
CLASSICAL  
2100 🏆 51 5,346 games

CORRESPONDENCE  
1994? 🏆 51 61 games

CRAZYHOUSE  
? 6 games



# Obtaining Data from opensource platforms



# Data

```
▶ with open('work (2).csv', encoding="utf8", errors='ignore') as f:  
    df = pd.read_csv(f)  
    print (df)  
    print (df.tail(10))  
    print (df.shape)
```

```
↳
```

4	9	Miyasaka, Marcus M	Niemann, Hans Moke	1/2-1/2	2134	2302	34
...	..	...	...	...	...	...	..
6404	7	Caruana, Fabiano	Erdos, Viktor	36526	2783	2651	10
6405	7	Caruana, Fabiano	Gajewski, Grzegorz	1/2-1/2	2783	2631	20
6406	7	Georgiev, Vladimir	Caruana, Fabiano	0-1	2553	2783	42
6407	9	Grischuk, Alexander	Caruana, Fabiano	0-1	2792	2791	27
6408	9	Caruana, Fabiano	Svidler, Peter	36526	2791	2753	22

```
[Event "Philadelphia op 12th"]\n[Site "Philadelphia"]\n[Date "2018.03.28"]\  
6399 [Event "Gashimov Memorial-A 1st"]\n[Site "Sham...  
6400 [Event "Gashimov Memorial-A 1st"]\n[Site "Sham...  
6401 [Event "Gashimov Memorial-A 1st"]\n[Site "Sham...  
6402 [Event "Gashimov Memorial-A 1st"]\n[Site "Sham...  
6403 [Event "Gashimov Memorial-A 1st"]\n[Site "Sham...  
6404 [Event "ITA-chT 46th"]\n[Site "Condino"]\n[Dat...  
6405 [Event "ITA-chT 46th"]\n[Site "Condino"]\n[Dat...  
6406 [Event "ITA-chT 46th"]\n[Site "Condino"]\n[Dat...  
6407 [Event "Norway Chess 2nd"]\n[Site "Stavanger"]...  
6408 [Event "Norway Chess 2nd"]\n[Site "Stavanger"]...
```

```
Normal  
6399 Normal  
6400 Normal  
6401 Normal  
6402 Normal  
6403 Normal  
6404 Normal
```

# Setting up Maia

```
(transfer_chess) zuka@zuka-Z390-AORUS-ULTRA:~/maia-individual/FancyDragoon$ ls
pgns split train validate
(transfer_chess) zuka@zuka-Z390-AORUS-ULTRA:~/maia-individual/FancyDragoon$ ls pgns
train_black.pgn train_white.pgn validate_black.pgn validate_white.pgn
(transfer_chess) zuka@zuka-Z390-AORUS-ULTRA:~/maia-individual/FancyDragoon$ ls split
games_black.pgn.bz2 games_white.pgn.bz2 train_black.pgn.bz2 train_white.pgn.bz2 validate_black.pgn.bz2 validate_white.pgn.bz2
(transfer_chess) zuka@zuka-Z390-AORUS-ULTRA:~/maia-individual/FancyDragoon$ ls train
black white
(transfer_chess) zuka@zuka-Z390-AORUS-ULTRA:~/maia-individual/FancyDragoon$ ls train/black
supervised-0
(transfer_chess) zuka@zuka-Z390-AORUS-ULTRA:~/maia-individual/FancyDragoon$ ls train/white
supervised-0
(transfer_chess) zuka@zuka-Z390-AORUS-ULTRA:~/maia-individual/FancyDragoon$ ls validate/
black white
(transfer_chess) zuka@zuka-Z390-AORUS-ULTRA:~/maia-individual/FancyDragoon$ ls validate/black
supervised-0
(transfer_chess) zuka@zuka-Z390-AORUS-ULTRA:~/maia-individual/FancyDragoon$ ls validate/white
supervised-0
(transfer_chess) zuka@zuka-Z390-AORUS-ULTRA:~/maia-individual/FancyDragoon$ ls validate/black/supervised-0/
game_000000.gz game_000001.gz game_000002.gz game_000003.gz game_000004.gz
(transfer_chess) zuka@zuka-Z390-AORUS-ULTRA:~/maia-individual/FancyDragoon$ ls validate/white/supervised-0/
game_000000.gz game_000001.gz game_000002.gz game_000003.gz game_000004.gz
(transfer_chess) zuka@zuka-Z390-AORUS-ULTRA:~/maia-individual/FancyDragoon$ ls train
black white
(transfer_chess) zuka@zuka-Z390-AORUS-ULTRA:~/maia-individual/FancyDragoon$ ls train/black/supervised-0/
game_000000.gz game_000004.gz game_000008.gz game_000012.gz game_000016.gz game_000020.gz game_000024.gz game_000028.gz game_000032.gz game_000036.gz
game_000001.gz game_000005.gz game_000009.gz game_000013.gz game_000017.gz game_000021.gz game_000025.gz game_000029.gz game_000033.gz
game_000002.gz game_000006.gz game_000010.gz game_000014.gz game_000018.gz game_000022.gz game_000026.gz game_000030.gz game_000034.gz
game_000003.gz game_000007.gz game_000011.gz game_000015.gz game_000019.gz game_000023.gz game_000027.gz game_000031.gz game_000035.gz
(transfer_chess) zuka@zuka-Z390-AORUS-ULTRA:~/maia-individual/FancyDragoon$ ls train/white/supervised-0/
game_000000.gz game_000004.gz game_000008.gz game_000012.gz game_000016.gz game_000020.gz game_000024.gz game_000028.gz game_000032.gz
game_000001.gz game_000005.gz game_000009.gz game_000013.gz game_000017.gz game_000021.gz game_000025.gz game_000029.gz game_000033.gz
game_000002.gz game_000006.gz game_000010.gz game_000014.gz game_000018.gz game_000022.gz game_000026.gz game_000030.gz game_000034.gz
game_000003.gz game_000007.gz game_000011.gz game_000015.gz game_000019.gz game_000023.gz game_000027.gz game_000031.gz game_000035.gz
(transfer_chess) zuka@zuka-Z390-AORUS-ULTRA:~/maia-individual/FancyDragoon$
```

```
2024-05-07 03:32:11 2-training/train_transfer.py final_config.yaml
2024-05-07 03:32:11 Starting train_transfer
2024-05-07 03:32:11 dataset:
  name: FancyDragoon
  path: .
gpu: 0
model:
  back_prop_blocks: 99
  filters: 64
  keep_weights: true
  path: maia-1900
  residual_blocks: 6
  se_ratio: 8
training:
  batch_size: 256
  checkpoint_steps: 500
  lr_boundaries:
  - 35000
  - 80000
  - 110000
  lr_values:
  - 0.01
  - 0.001
  - 0.0001
  - 1.0e-05
  num_batch_splits: 1
  policy_loss_weight: 1.0
  precision: half
  shuffle_size: 256
  test_steps: 2000
  total_steps: 150000
  train_avg_report_steps: 50
  value_loss_weight: 1.0

2024-05-07 03:32:11 found ['./FancyDragoon/train'] chunk dirs
2024-05-07 03:32:11 found 73 chunks totalck chunks
sorting 37 B chunks...sorting 36 W chunks...[done]
game_000000.gz - game_000035.gz
game_000000.gz - game_000036.gz
2024-05-07 03:32:11 found ['./FancyDragoon/validate'] chunk dirs
2024-05-07 03:32:11 found 10 chunks total chunks
sorting 5 B chunks...sorting 5 W chunks...[done]
game_000000.gz - game_000002.gz
game_000000.gz - game_000004.gz
2024-05-07 03:32:11 Using 14 worker processes.
2024-05-07 03:32:11 14 tasks started
2024-05-07 03:32:12 Using 14 worker processes.
2024-05-07 03:32:12 14 tasks started
loading lower weights from /home/zuka/anaconda3/envs/transfer_chess/llb/python3.7/site-packages/backend-1.0.0-py3.7.egg/models/maia-1900/ckpt/ckpt-40
2024-05-07 03:32:16 Setting up lc0 stuff
Using 1 evaluation batches
2024-05-07 03:32:20 step 1, policy=1.41341 value=0.680479 policy accuracy=54.6875% value accuracy=68.3594% mse=0.10384
2024-05-07 03:32:28 step 50, lr=0.01 policy=1.46237 value=0.786972 mse=0.186035 reg=0.0504127 total=2.29975 (1114.53 pos/s)
2024-05-07 03:32:35 step 100, lr=0.01 policy=1.47145 value=0.782909 mse=0.189233 reg=0.050725 total=2.30509 (1640.88 pos/s)
2024-05-07 03:32:41 step 150, lr=0.01 policy=1.45381 value=0.770327 mse=0.185295 reg=0.051024 total=2.27516 (2170.45 pos/s)
2024-05-07 03:32:47 step 200, lr=0.01 policy=1.42974 value=0.764251 mse=0.186501 reg=0.0513232 total=2.24531 (2344.24 pos/s)
2024-05-07 03:32:52 step 250, lr=0.01 policy=1.4096 value=0.778965 mse=0.185798 reg=0.0516263 total=2.24019 (2253.36 pos/s)
2024-05-07 03:32:58 step 300, lr=0.01 policy=1.41192 value=0.761401 mse=0.183661 reg=0.0519247 total=2.22525 (2235.28 pos/s)
2024-05-07 03:33:04 step 350, lr=0.01 policy=1.38909 value=0.772159 mse=0.184294 reg=0.0522276 total=2.21348 (2198.81 pos/s)
2024-05-07 03:33:10 step 400, lr=0.01 policy=1.3446 value=0.756534 mse=0.183641 reg=0.0525236 total=2.15365 (2101.72 pos/s)
2024-05-07 03:33:15 step 450, lr=0.01 policy=1.37055 value=0.766511 mse=0.181481 reg=0.0528108 total=2.18987 (2361.01 pos/s)
2024-05-07 03:33:21 step 500, lr=0.01 policy=1.34606 value=0.746091 mse=0.181177 reg=0.0531001 total=2.14615 (2185.94 pos/s)
Model saved in file: models/final_config/ckpt-1
saved as 'models/final_config/ckpt-1-500.pb.gz' 1.2M
Weights saved in file: models/final_config/ckpt-1-500
2024-05-07 03:33:27 step 550, lr=0.01 policy=1.34087 value=0.743855 mse=0.182102 reg=0.0533975 total=2.13812 (2134.29 pos/s)
2024-05-07 03:33:33 step 600, lr=0.01 policy=1.32433 value=0.746509 mse=0.180094 reg=0.0537128 total=2.12456 (2297.84 pos/s)
2024-05-07 03:33:39 step 650, lr=0.01 policy=1.31271 value=0.756225 mse=0.182675 reg=0.0540237 total=2.12296 (2217.04 pos/s)
2024-05-07 03:33:44 step 700, lr=0.01 policy=1.32373 value=0.765104 mse=0.178234 reg=0.0543258 total=2.14316 (2253.65 pos/s)
2024-05-07 03:33:50 step 750, lr=0.01 policy=1.32932 value=0.741957 mse=0.177463 reg=0.0546403 total=2.12592 (2174.12 pos/s)
2024-05-07 03:33:56 step 800, lr=0.01 policy=1.27251 value=0.746284 mse=0.180030 reg=0.0549630 total=2.07375 (2235.28 pos/s)
```

# Final Configuration

# Used scripts for Data extraction

```
centipawn_metrics_extractor.py > ...
83     previous_state_evaluation = evaluationsAdjusted[index - 1]
84     current_state_evaluation = evaluationsAdjusted[index]
85     if index % 2 != 0:
86         white_centipawn_loss_list.append(previous_state_evaluation - current_state_evaluation)
87     else:
88         black_centipawn_loss_list.append(current_state_evaluation - previous_state_evaluation)
89     index += 1
90
91     white_centipawn_loss_list_adjusted = [0 if x < 0 else x for x in white_centipawn_loss_list]
92     black_centipawn_loss_list_adjusted = [0 if x < 0 else x for x in black_centipawn_loss_list]
93
94     white_average_centipawn_loss = round(sum(white_centipawn_loss_list_adjusted) / len(white_centipawn_loss_list_adjusted))
95     black_average_centipawn_loss = round(sum(black_centipawn_loss_list_adjusted) / len(black_centipawn_loss_list_adjusted))
96
97     print("White average centipawn loss: {}".format(white_average_centipawn_loss))
98     print("Black average centipawn loss: {}".format(black_average_centipawn_loss))
99
100     # Fill dataframe with game data and results
101     gameDf = pd.DataFrame(columns=worksheetColumns, index=range(1))
102
103     gameDf['Date'] = datetime.strptime(game.headers["Date"], "%Y.%m.%d")
104     gameDf['Event Name'] = game.headers["Event"]
105     gameDf['Event Rounds'] = game.headers["EventRounds"]
106     gameDf['Round'] = game.headers["Round"]
107     gameDf['Moves'] = math.ceil(int(game.headers["PlyCount"])/2)
108     gameDf.at[0, 'Evaluations List'] = evaluationsAdjusted
109     exporter = chess.pgn.StringExporter(headers=True, variations=True, comments=True)
110     pgn_string = game.accept(exporter)
111     gameDf.at[0, 'PGN'] = pgn_string
112     print(pgn_string)
113     gameDf['Analysis Depth'] = depth
114
```

# Data normalization

```
def encode_numeric_zscore(df, name, mean=None, sd=None):  
    if mean is None:  
        mean = df[name].mean()  
  
    if sd is None:  
        sd = df[name].std()  
  
    df[name] = (df[name] - mean) / sd
```

```
def encode_text_dummy(df, name):  
    dummies = pd.get_dummies(df[name])  
    for x in dummies.columns:  
        dummy_name = str(name) + "-" + str(x)  
        df[dummy_name] = dummies[x]  
    df.drop(name, axis=1, inplace=True)
```

```
df.columns=['Event Rounds', 'White Name', 'Black Name', 'Result', 'White ELO', 'Black ELO', 'Moves', 'White Av CP Loss',  
           'Black Av CP Loss', 'PGN', 'Final']
```

```
print (df.columns)
```

```
numbers=['Event Rounds', 'White ELO', 'Black ELO', 'Moves', 'White Av CP Loss',  
         'Black Av CP Loss']  
strings=['White Name', 'Black Name', 'Result', 'PGN']
```



# Neural network Demonstration

```
model = Sequential()
model.add(Dense(10, input_dim=x.shape[1], activation='relu'))
model.add(Dense(50, input_dim=x.shape[1], activation='relu'))
model.add(Dense(10, input_dim=x.shape[1], activation='relu'))
model.add(Dense(1, kernel_initializer='normal'))
model.add(Dense(y.shape[1], activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam')
monitor = EarlyStopping(monitor='val_loss', min_delta=1e-3,
                        patience=5, verbose=1, mode='auto',
                        restore_best_weights=True)

model.fit(x_train,y_train,validation_data=(x_test,y_test),
        callbacks=[monitor],verbose=2,epochs=10)

pred = model.predict(x_test)
pred = np.argmax(pred,axis=1)
y_eval = np.argmax(y_test,axis=1)
score = metrics.accuracy_score(y_eval, pred)
print(i,"Validation score: {}".format(score))
```

# Results without Maia Parameters vs With:

```
Epoch 1/10
151/151 - 3s - loss: 0.3005 - val_loss: 0.0923 - 3s/epoch - 23ms/step
Epoch 2/10
151/151 - 1s - loss: 0.0848 - val_loss: 0.0822 - 1s/epoch - 8ms/step
Epoch 3/10
151/151 - 1s - loss: 0.0637 - val_loss: 0.0726 - 915ms/epoch - 6ms/step
Epoch 4/10
151/151 - 1s - loss: 0.0425 - val_loss: 0.0743 - 942ms/epoch - 6ms/step
Epoch 5/10
151/151 - 1s - loss: 0.0260 - val_loss: 0.0773 - 920ms/epoch - 6ms/step
Epoch 6/10
151/151 - 1s - loss: 0.0170 - val_loss: 0.0880 - 957ms/epoch - 6ms/step
Epoch 7/10
151/151 - 1s - loss: 0.0077 - val_loss: 0.1181 - 956ms/epoch - 6ms/step
Epoch 8/10
Restoring model weights from the end of the best epoch: 3.
151/151 - 1s - loss: 0.0029 - val_loss: 0.1434 - 946ms/epoch - 6ms/step
Epoch 8: early stopping
<keras.src.callbacks.History at 0x798d5e554820>
```

```
pred = model.predict(x_test)
pred = np.argmax(pred,axis=1)
y_eval = np.argmax(y_test,axis=1)
score = metrics.accuracy_score(y_eval, pred)
print(i,"Validation score: {}".format(score))
```

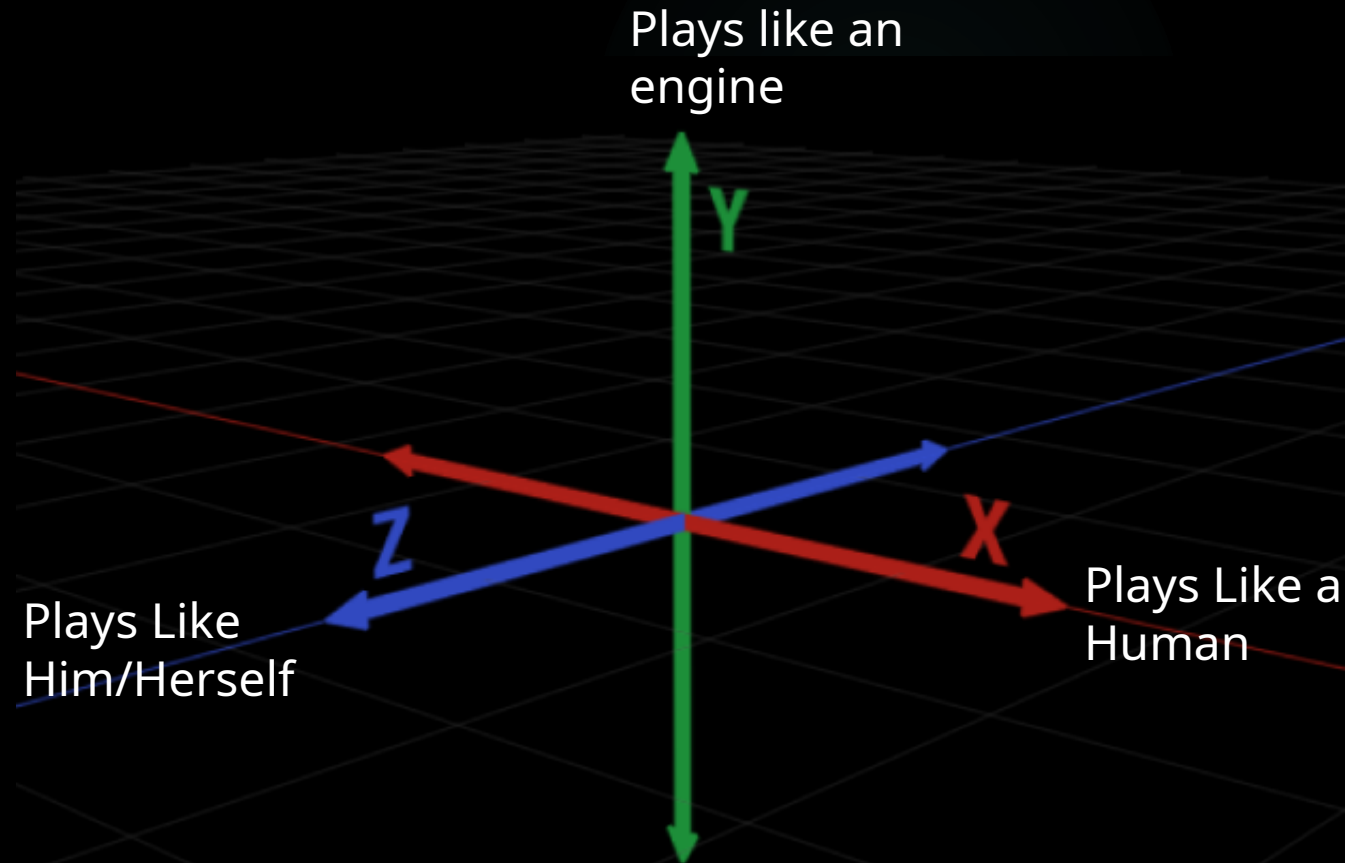
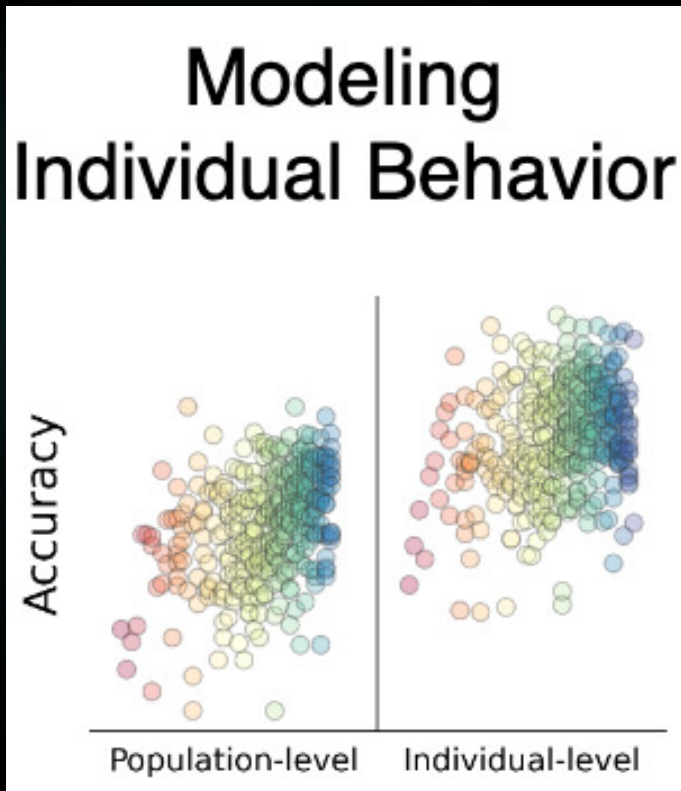
```
51/51 [=====] - 0s 4ms/step
PGN Validation score: 0.9800374298190893
```

```
Epoch 1/10
151/151 - 3s - loss: 0.3005 - val_loss: 0.0923 - 3s/epoch - 23ms/step
Epoch 2/10
151/151 - 1s - loss: 0.0848 - val_loss: 0.0822 - 1s/epoch - 8ms/step
Epoch 3/10
151/151 - 1s - loss: 0.0637 - val_loss: 0.0726 - 915ms/epoch - 6ms/step
Epoch 4/10
151/151 - 1s - loss: 0.0425 - val_loss: 0.0743 - 942ms/epoch - 6ms/step
Epoch 5/10
151/151 - 1s - loss: 0.0260 - val_loss: 0.0773 - 920ms/epoch - 6ms/step
Epoch 6/10
151/151 - 1s - loss: 0.0170 - val_loss: 0.0880 - 957ms/epoch - 6ms/step
Epoch 7/10
151/151 - 1s - loss: 0.0077 - val_loss: 0.1181 - 956ms/epoch - 6ms/step
Epoch 8/10
Restoring model weights from the end of the best epoch: 3.
151/151 - 1s - loss: 0.0029 - val_loss: 0.1434 - 946ms/epoch - 6ms/step
Epoch 8: early stopping
<keras.src.callbacks.History at 0x798d5e554820>
```

```
pred = model.predict(x_test)
pred = np.argmax(pred,axis=1)
y_eval = np.argmax(y_test,axis=1)
score = metrics.accuracy_score(y_eval, pred)
print(i,"Validation score: {}".format(score))
```

```
51/51 [=====] - 0s 4ms/step
PGN Validation score: 0.8000374298190893
```

# Future Plans: experimenting with Maia individual



# Future Plans: Integration with Irwin



IRWIN AI  
HUNTER

Updated 2 hours ago  
Hit "Evaluate" to update

PV 0 Rate by Ambiguity

82% 53% 53% 36% 38%  
1 2 3 4 5

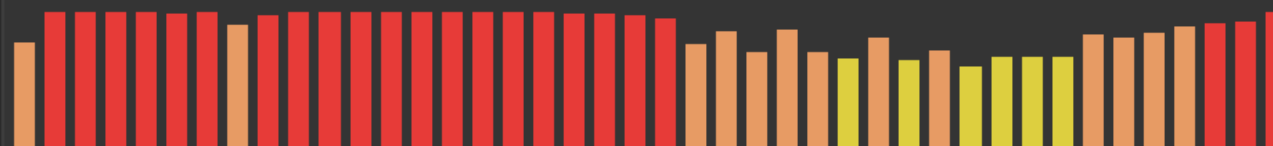
98%

Overall assessment



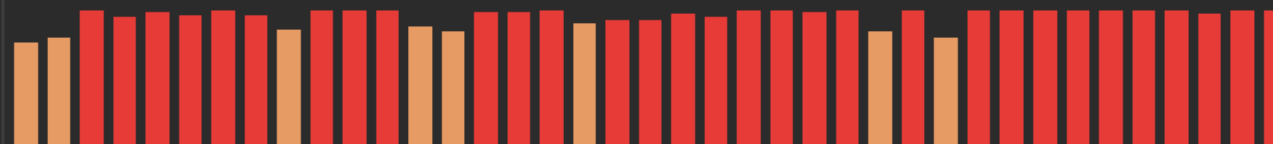
gous (2000) -11  
⌚ 10+0 - 23 days ago

98% assessment  
0% blurs



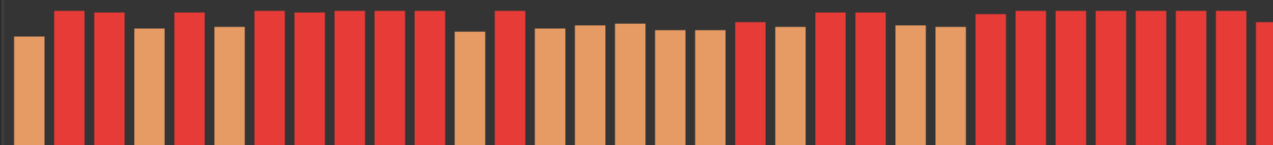
stanojkov (1797) -5  
⌚ 5+10 - 11 days ago

98% assessment  
0% blurs



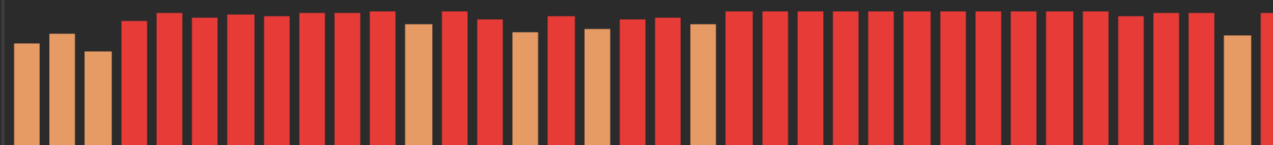
oquzlar (2159) -12  
⌚ 15+0 - 16 days ago

98% assessment  
0% blurs



Volodya51 (2009) -9  
⌚ 10+0 - 16 days ago

98% assessment  
0% blurs



BugaevAnton (1976) -8  
⌚ 20+0 - 11 days ago

97% assessment  
0% blurs

**THANK YOU FOR YOUR  
ATTENTION**



**ANY QUESTIONS?**