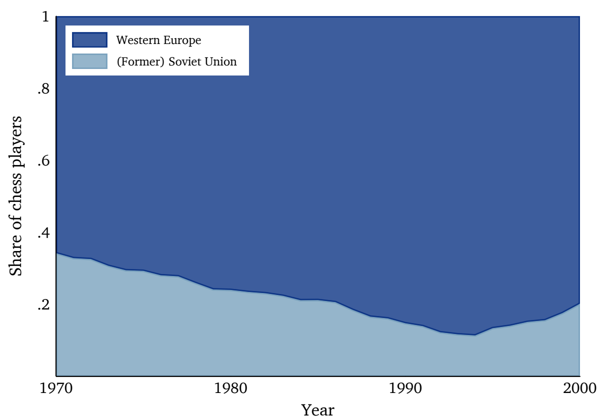
ONLINE APPENDIX

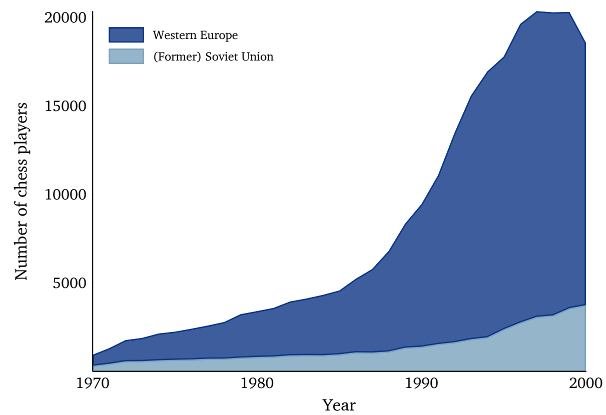
A Appendix: Figures

Figure A-1: Chess players over time and region

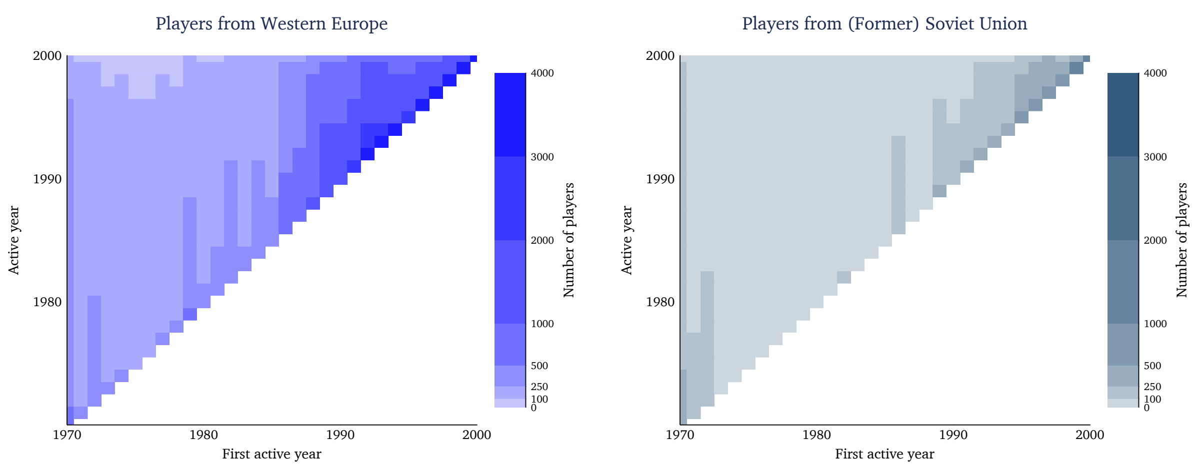
(a) Share of players by region



(b) Number of players by region



(c) Activity over time by player cohort

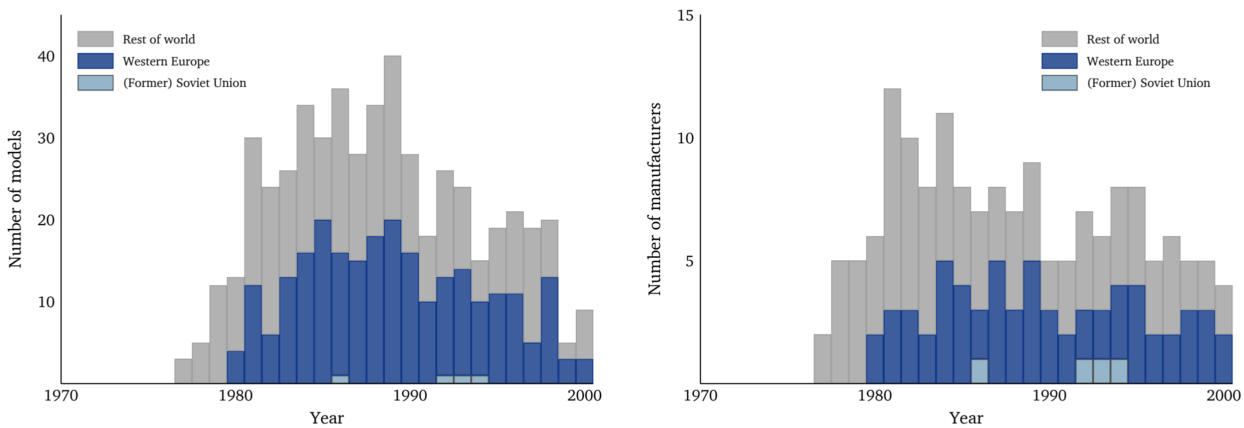


Notes: This set of figures illustrates the composition of our panel of human chess players over time and region. Figure A-1a depicts the share of active players from Western Europe and the (former) Soviet Union over time. Figure A-1b depicts the absolute count of active players from Western Europe and the (former) Soviet Union over time. Figure A-1c depicts for each region separately the number of active players over time by their first year of activity.

Figure A-2: Chess computer models and manufacturers over time and region

(a) New models over time

(b) Manufacturers over time

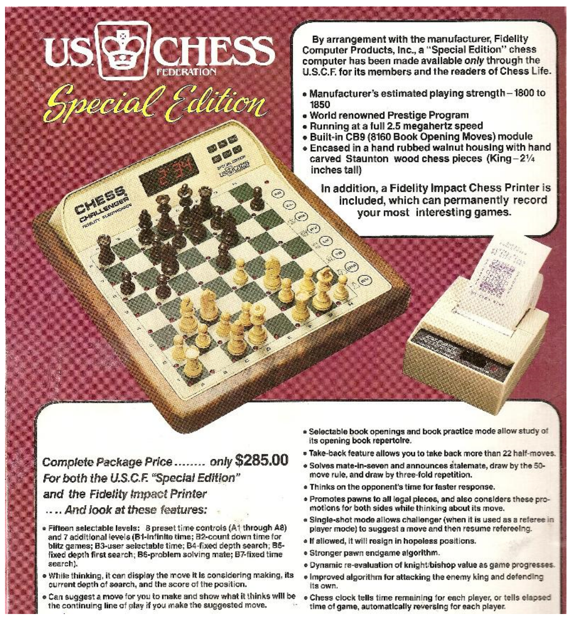


Notes: These two figures illustrate the number of chess computer models (Figure A-2a) and chess computer manufacturers (Figure A-2b) over time and region.

Figure A-3: The first personal (dedicated) chess computer *Challenger 1* (1977)

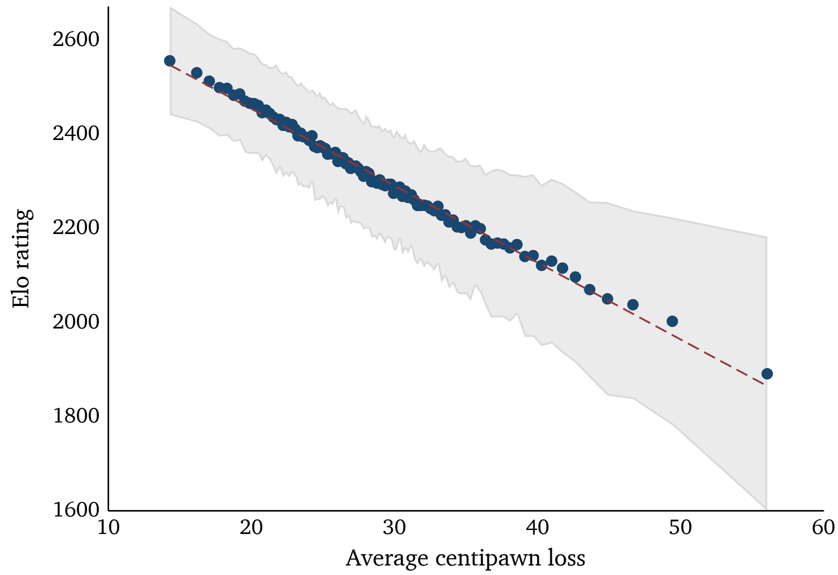
Notes: This figure shows a picture of the first personal chess computer: the *Challenger 1* by *Fidelity Electronics*. Source: https://www.computerhistory.org.

Figure A-4: USCF promotion of a chess computer for members (1982)



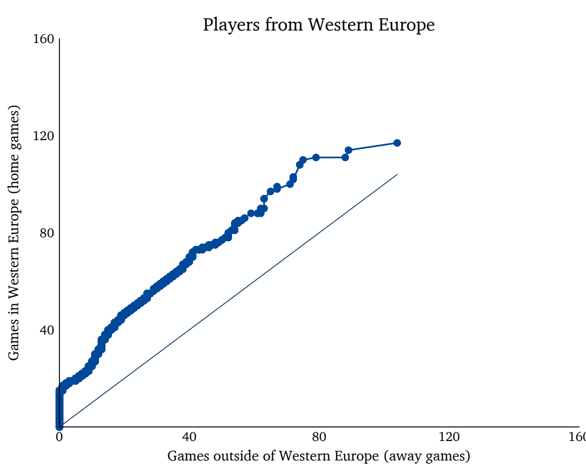
Notes: This figure shows a personal chess computer model advertised by the United States Chess Federation (USCF) to its members. Source: https://www.computerhistory.org.

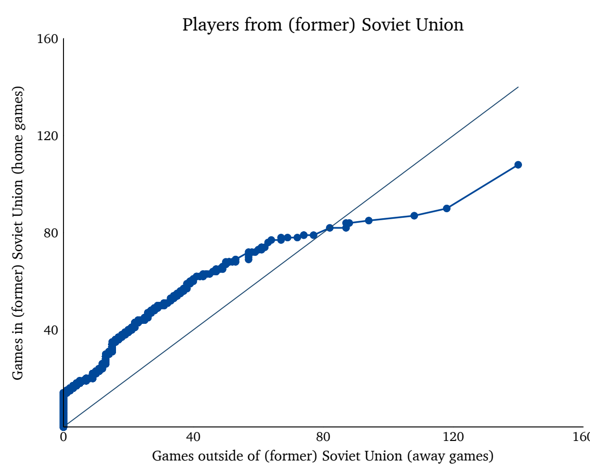
Figure A-5: Relationship between Elo rating and average centipawn loss



Notes: This figure plots the mean ELO rating and the mean average centipawn loss for 100 bins. The dashed line depicts the fit line. The grey area indicates the band of one standard deviation above and below the mean in each bin. The disaggregated level of observation is the human chess player by year. Average centipawn losses based on fewer than 250 moves per year excluded.

Figure A-6: Home and away games of human chess players by region

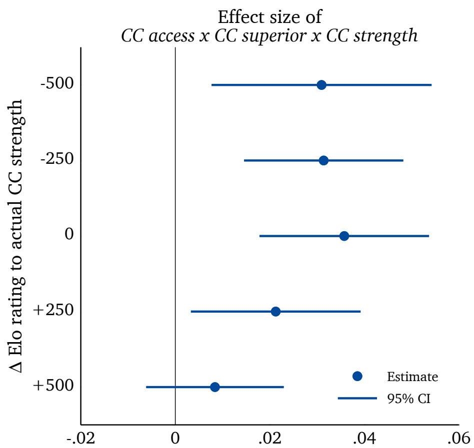




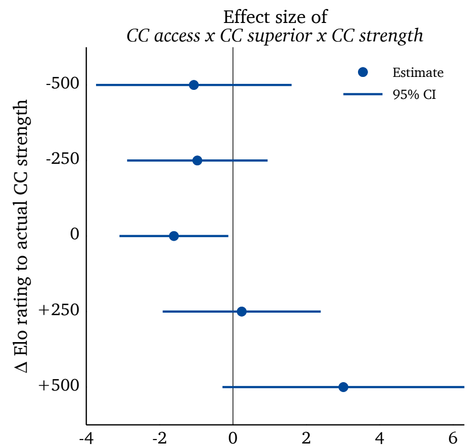
Notes: This figure illustrates the distribution of home and away tournament games of human chess players by each region separately. The figure is a quantile-quantile plot that compares the distributions of tournament games in the domestic region (Western Europe and the (former) Soviet Union, respectively) and games outside of it. Points above the 45 degrees line indicate a larger share of games played in the player's domestic region. Only tournament games before 1989 considered.

Figure A-7: Estimates by manipulated chess computer strength (placebo regressions)

(a) Elo rating



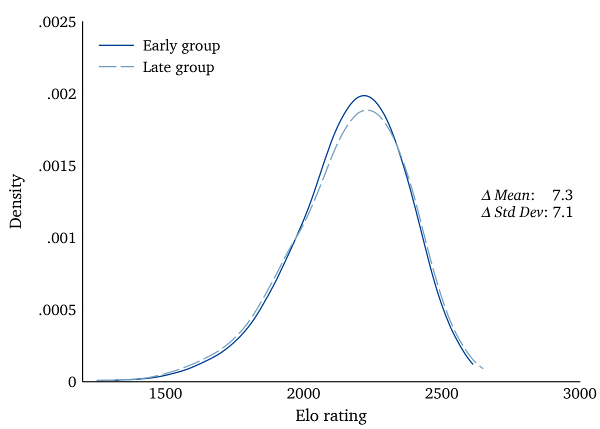
(b) Average centipawn loss



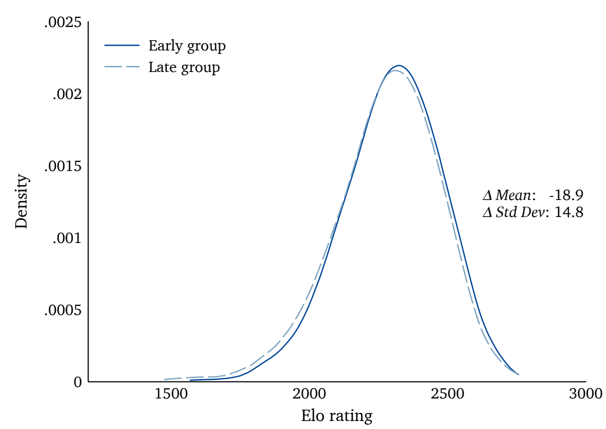
Notes: These two figures provide a comparison of estimates based on regressions with differently manipulated chess computer strength. Figure A-7a states the effect size of *Chess computer (CC) access x Chess computer (CC) superior x Chess computer (CC) strength* on player performance (Elo rating). Figure A-7b states the effect size of *Chess computer (CC) access x Chess computer (CC) superior x Chess computer (CC) strength* on player performance (average centipawn loss). The estimates are based on regressions as specified in equation (2). In each model, *Chess computer strength* is manipulated by adding (subtracting) 250 or 500 Elo points to (from) the chess computer's actual rating. Note that the variables *Chess computer access* × *Chess computer strength*, *Chess computer superior*, and *Chess computer access* × *Chess computer superior* are a function of the manipulated computer strength.

Figure A-8: Skill distribution over time and region

(a) Western Europe

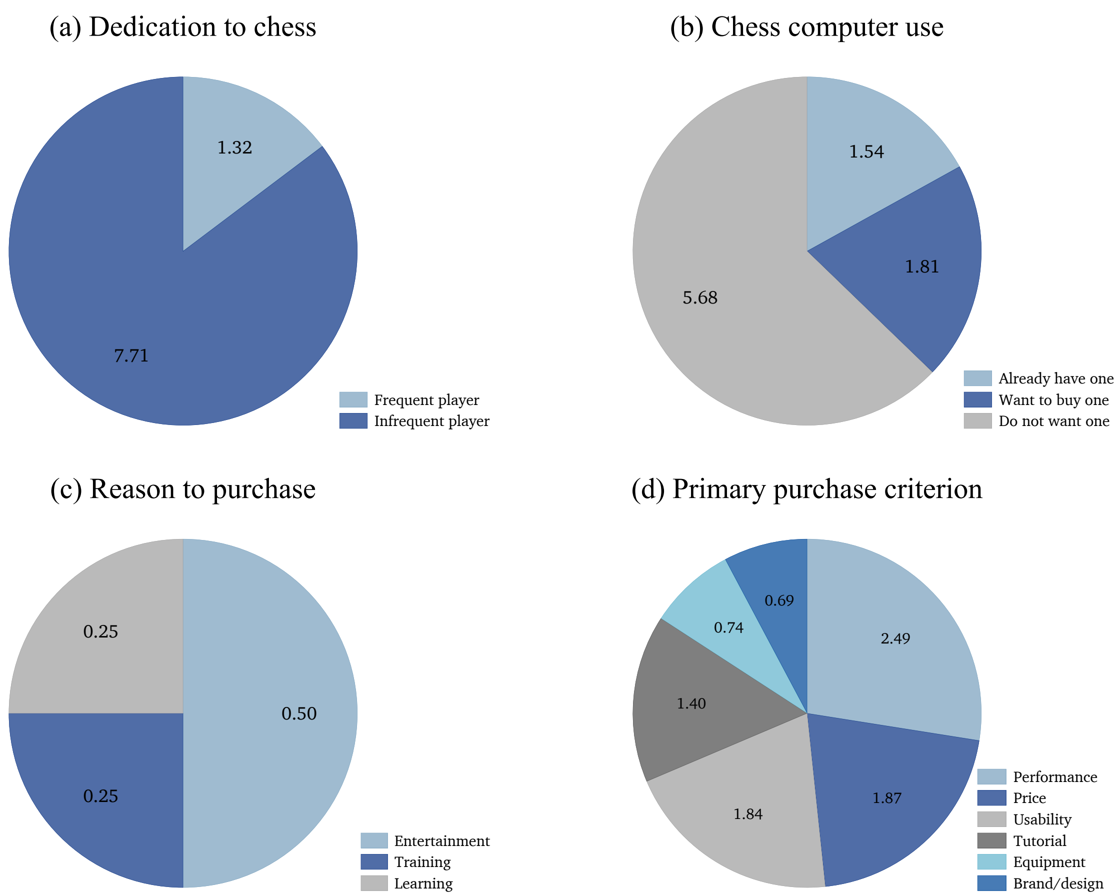


(b) Soviet Union



Notes: These two figures plot the skill distribution as kernel densities for Western and Soviet chess players active in 1980 (early group) and active in 1988 (late group). Figure A-8a depicts the skill distribution of active players from Western Europe. Figure A-8b depicts the skill distribution of active players from the Soviet Union. Δ*Mean* represents the difference in the mean Elo rating between the early group and the late group. Δ*Std Dev* represents the difference in the standard deviation of Elo rating between the early group and the late group.

Figure A-9: Results of a 1987 market survey on West German chess players (in millions)



Notes: Market analysis on behalf of chess computer manufacturer *Hegener+Glaser Aktiengesellschaft*. Survey conducted between June and August 1987. Results based on 2,014 respondents. The survey estimates a population of about 9 million chess players in West Germany.

B Appendix: Tables

Table B-1: Correlation between average centipawn loss (white/black) and chess computer access

Sample: All players DV:	(1) Chess Computer access
Average centipawn loss (white)	−0.00013
	(0.00007)
Average centipawn loss (black)	−0.00002
	(0.00005)
Player FE	Yes
Player age FE	Yes
Year FE	Yes
Observations	151475
Players	17633
Log-likelihood	88365

Notes: Column (1) shows the estimates of a linear model regression with high-dimensional fixed effects. The unit of observation is the individual chess player by year. Robust standard errors clustered at player level in parentheses.

Table B-2: Chess performance over time for different player subsamples

Sample: All players with:	(1) ≥ 90 games	(2)	(3) ≥ 9 years active	(4)	(5) ≥ 2000 Elo	(6)
	Elo rating	Average centipawn loss	Elo rating	Average centipawn loss	Elo rating	Average centipawn loss
Chess computer (CC) access	0.013	−0.833	0.011	−0.385	0.012	−0.361
	(0.004)	(0.359)	(0.003)	(0.306)	(0.002)	(0.310)
Player FE	Yes	Yes	Yes	Yes	Yes	Yes
Player age FE	Yes	Yes	Yes	Yes	Yes	Yes
Year FE	Yes	Yes	Yes	Yes	Yes	Yes
Observations	52542	52542	107474	107474	126951	126951
Players	3355	3355	6461	6461	13526	13526
Log-likelihood	67001	−182791	148092	−390591	209530	−456276

Notes: Columns (1) to (6) show the estimates of linear model regressions with high-dimensional fixed effects. The unit of observation is the individual chess player by year. Robust standard errors clustered at player level in parentheses.

Table B-3: Chess performance over time for restricted observation windows

Sample: All players with:	(1) 1970-1988	(2)	(3) 1977-2000	(4)	(5) 1985-1995	(6)
	Elo rating	Average centipawn loss	Elo rating	Average centipawn loss	Elo rating	Average centipawn loss
Chess computer (CC) access	0.014	−1.110	0.010	−1.099	0.011	−0.362
	(0.004)	(0.567)	(0.003)	(0.342)	(0.003)	(0.302)
Player FE	Yes	Yes	Yes	Yes	Yes	Yes
Player age FE	Yes	Yes	Yes	Yes	Yes	Yes
Year FE	Yes	Yes	Yes	Yes	Yes	Yes
Observations	42746	42746	163123	163123	170968	170968
Players	4579	4579	19164	19164	19285	19285
Log-likelihood	64369	−150052	219780	−592681	228875	−623057

Notes: Columns (1) to (6) show the estimates of linear model regressions with high-dimensional fixed effects. The unit of observation is the individual chess player by year. Robust standard errors clustered at player level in parentheses.

Table B-4: Chess performance over time by each region separately

Sample:	(1) Soviet players	(2)	(3) Western players	(4)
	Elo rating	Average centipawn loss	Elo rating	Average centipawn loss
Chess computer (CC) access × CC superior	0.028	0.091	−0.014	−0.266
	(0.009)	(0.737)	(0.002)	(0.177)
CC access × CC superior × CC strength	0.065	−1.349	0.042	−2.273
	(0.021)	(2.341)	(0.010)	(0.824)
Player FE	Yes	Yes	Yes	Yes
Player age FE	Yes	Yes	Yes	Yes
Year FE	Yes	Yes	Yes	Yes
Observations	30606	30606	140362	140362
Players	3433	3433	15852	15852
Log-likelihood	41133	−105816	188239	−515959

Notes: Columns (1) to (4) show the estimates of linear model regressions with high-dimensional fixed effects. The unit of observation is the individual chess player by year. Robust standard errors clustered at player level in parentheses.

C Technical Appendix: Evaluating Moves

We use the *python-chess* module to implement an evaluation algorithm and *Stockfish 10* as a chess engine to evaluate the performance of chess players at the move level. The resulting *average centipawn loss* measure follows the definition of Guid and Brakto (2011).[1] We first give a motivation for why we use this measure and then detail its construction.

Background

The measure introduced by Guid and Brakto (2011) allows for the comparison of chess players even if they never directly played against each other. The lack of encounters may be due to local clustering, distance, or because their timespans of being an active player do not overlap. This property holds because the evaluation is independent of the overall quality of play of the environment in which the player to be evaluated interacts and is solely based on move-level performance.

The measure allows for absolute comparison as it uses an external evaluator that is stable across locations and times. To this end, they quantify the advantage of each player over his/her opponent in the position preceding and the one following a move. After the difference in advantage that results from the move is calculated, this number will be compared to the evaluation of the hypothetical, best possible move as suggested by the engine. The resulting move-level skill measure can be interpreted as the number of advantage points a player misses to those of a chess computer that was confronted with the exact same position.

For our analysis, we chose Stockfish 10, which is considered the strongest chess engine available. The analysis excludes the opening game, which is based mostly on theory and still challenges current chess programs in delivering accurate evaluations. Evaluation in each game therefore starts at the 12th move of the white player.

[1] Guid, M., & Bratko, I. (2011). Using heuristic-search based engines for estimating human skill at chess. *ICGA Journal*, 34(2), 71-81.

Method

Our initial dataset contains game-level observations about all the moves played. The algorithm will loop through all games $g \in \{1, 2, \ldots n\}$ and all halfmoves of game g that have indices $i \in \{1, 2, \ldots, m_g\}$. Each single move of a game is called a halfmove or ply, whereas a fullmove always contains two moves, one by the white and one by the black player. We only use the notion of a halfmove in the subsequent chapters.

As a basis, we define $position_{ig}$ as a board position before halfmove number i of game g is played. $evaluation_{igp}$ represents the chess engine's evaluation of $position_{ig}$ from perspective of player $p \in \{0, 1\}$, where:

$$evaluation_{igp} = f_p(position_{ig}, \theta).$$

The function f_p represents Stockfish's internal evaluation algorithm that takes a board position as an input and returns a score, representing the advantage from the perspective of player p over its opponent. θ contains engine parameters such as the chosen search depth δ and contempt factor κ. The type of score the engine will return depends on whether there are end nodes in the current search tree. If there are, the return value will be a mate score, representing the number of moves until the player is expected to mate his/her opponent in case of a positive value, or the number of moves until he/she is expected to be mated in case of negative values. The usual case is when there are no end nodes present. The engine will then return a centipawn score that represents the advantage of a player measured as a hundredth of a pawn. Since we want to calculate differences in advantage, we need to use the same units and therefore use another function to convert mate- into centipawn-scores:

$$\begin{aligned} advantage_{igp} &= g(f_p(position_{ig}, \theta)) \\ &= g(evaluation_{igp}), \end{aligned}$$

where

$$g(evaluation_{igp}) = \begin{cases} 32768 - evaluation_{igp} * 1000, & \text{if} \quad evaluation_{igp} \in \mathbb{Z} \setminus \{0\} \times \{MATE\} \\ evaluation_{igp}, & \text{if} \quad evaluation_{igp} \in \mathbb{Z} \times \{CP\} \end{cases}$$

The constant in this linear function is the maximum value of a 16-bit integer measured in centipawns. We will come back to why the exact number is of less importance. For each mate-score in $evaluation_{igp}$, we subtract 1,000 centipawns from this number, representing the willingness to sacrifice the queen (which is worth 900 CP) in order to delay a checkmate by one move. The engine also returns a suggestion for the best possible move to play in the current board position.

In order to proceed, lets define m_{ig} as a move and part of the space of all possible moves $m_{ig} \in \mathscr{M}_{ig}$ in $position_{ig}$ that maps a given board position $\overline{position}_{ig}$ into $position_{i+1g}$:

$$\text{move } m_{ig} = \psi^{-1}|_{m_{ig}}(\overline{position}_{ig}, position_{i+1gp}), \text{with}$$

$$position_{i+1g} = \psi(\overline{position}_{ig}, m_{ig})$$

The best move $\widetilde{m}_{ig}$ is maximizing the engine's evaluation of $position_{i+1,g}$ for all possible moves $m_{ig} \in \mathscr{M}_{ig}$ in $\overline{position}_{ig}$ given engine parameters θ:

$$\widetilde{m}_{ig} = \underset{m_{ig} \in \mathscr{M}_{ig}}{\arg\max} f_p(\psi(\overline{position}_{ig}, m_{ig}), \theta)$$

The advantage of player p after playing this hypothetical move $\widetilde{m}_{ig}$ in $position_{ig}$ is:

$$\widetilde{advantage}_{i+1gp} = g(f_p(\psi(position_{ig}, \widetilde{m}_{ig}), \theta))$$

We then define a $loss_{igp}$ as being the evaluation of move m_{ig} according to Γ_p. This function simply takes the difference in the advantage of player p before and after playing that move, given the engine parameters θ:

$$loss_{igp} = \Gamma_p(m_{ig}, \theta)$$

$$\Gamma_p(m_{ig}, \theta) = g(f_p(position_{ig}, \theta)) - g(f_p(\psi(position_{ig}, m_{ig}), \theta))$$

$$= advantage_{igp} - advantage_{i+1gp}$$

The definitions above now allow us to define the move-level skill measure. It can be described as the evaluation of the move played but corrected by the evaluation of the best possible move:

$$
\begin{aligned}
moveCPL_{igp} &= \Gamma_p(m_{ig}, \theta) - \Gamma_p(\widetilde{m}_{ig}, \theta) \\
&= \{g(f_p(\overline{position}_{ig}, \theta)) - g(f_p(\psi(\overline{position}_{ig}, m_{ig}), \theta))\} \\
&\quad - \{g(f_p(\overline{position}_{ig}, \theta)) - g(f_p(\psi(\overline{position}_{ig}, \widetilde{m}_{ig}), \theta))\} \\
&= g(f_p(\psi(\overline{position}_{ig}, \widetilde{m}_{ig}), \theta)) - g(f_p(\psi(\overline{position}_{ig}, m_{ig}), \theta))
\end{aligned}
$$

The final skill measure of player p in game g is calculated by aggregating the move-level evaluations:

$$
\begin{aligned}
aCPL_{p,g} &= \sum_{i=0}^{m_g} \Phi(moveCPL_{igp}) * d_{i,g_p} \\
&= \sum_{i=0}^{m_g} \Phi(\Gamma_p(m_{ig}, \theta) - \Gamma_p(\widetilde{m}_{ig}, \theta)) * d_{ig},
\end{aligned}
$$

where

$$
d_{igp} = \begin{cases} 1, & \text{if } \Gamma_p(m_{ig}, \theta) \in [-200, 200] \wedge \Gamma_p(\widetilde{m}_{ig}, \theta) \in [-200, 200] \\ 0, & \text{otherwise} \end{cases}
$$

$$
\Phi(moveCPL_{igp}) = \begin{cases} -300, & \text{if} \quad moveCPL_{igp} < 300, \\ moveCPL_{igp}, & \text{if} \quad moveCPL_{igp} \in [-300, 300], \\ 300, & \text{if} \quad moveCPL_{igp} > 300. \end{cases}
$$

The binary variable d_{igp} equals one if both, the move played and the move suggested by the engine, are evaluated within the interval $[-200, 200]$. Only in this case will the evaluation be considered for the game-level skill measure. The reason for this is that players who have a strong advantage might not choose to play the best but a less risky move and therefore would get punished illegitimately. If a player considers his/her position to be lost, he/she similarly might on purpose play a worse move in an attempt to turn around the advantage, which again would result in an objectively illegitimate evaluation loss.

Additionally, for some mistakes, the player gets punished disproportionately high, resulting from the numerical difference between the erroneous move and the evaluated choice of the engine. Guid and Brakto (2011) therefore confine the output space of the evaluation function to the interval $[-300, 300]$, where 300 CPLs should represent a "huge mistake [...] through the eyes of a human expert."