Research Article

Open Access

Reevaluation of artificial intelligence engine alpha zero, a self-learning algorithm, reveals lack of proof of best engine, and an advancement of artificial intelligence via multiple roots

Abstract

Artificial Intelligence (AI) is at the heart of IT-research and pioneers automated problem-solving. A recent breakthrough that could shift the AI-field was reported by Silver and colleagues of the famous Google Deep mind developer team around Demis Hassabis. They have reported that a new adaption of their in-house generalized neural network (NN) machine learning algorithm, termed *AlphaZero*, has achieved to outperform the world's best-rated chess engine *Stockfish*, already after four hours of self-learning, and starting only with the rules of chess. *AlphaZero* originates from the superhuman *Alpha Go* program and could beat the best chess engine via blank slate tabula rasa self-play reinforcement machine learning, i.e. only by learning from many games played against itself. This worldwide strongest AI-performance claim has been drawn from a 100-game match between *AlphaZero* and *Stockfish* engines and has attracted much attention by the media, especially in the world of chess, which has been historically a key domain of AI. *AlphaZero* did not lose one game and won 28 times, while the remainders of the 100 games were draws. General reinforcement learning is very promising for many applications of mathematical solution finding in complexity. However, the requirement to independently verify if this breakthrough AI claim can be made poses some major difficulties and raises some inevitable doubts if a final proof has been given. Machine and method details are not available and only10 example games were given. This research starts with a reproducibility testing of all 10 example games and reveals that *AlphaZero* shows signs of human openings and might have outperformed *Stockfish* due to an irregular underperformance of *Stockfish8*, like post-opening novelties or subperfect game moves, like in game moves and post-opening novelites. At this juncture, the testing revealed that AI quiescence searches could be improved via multiple roots for both engines, which could boost all future AI performances. In light of a lack of tournament conditions and an independent referee, comparability challenges of software and hardware configurations such as *AlphaZero's* TFLOP super-calculation-power, this work suggests that a final best AI-engine-claim requires further proof. Overclaim biases are found in all sciences of today due to the publishing imperatives and wish to be first.

Keywords: Artificial intelligence, ai, best, chess, engine, stockfish, *AlphaZero*, google, deepmind, elo, bias, machine, learning, tflops, root, tree, algorithm

Volume 1 Issue 1 - 2018

Roman Anton
The University of Truth and Common Sense, Department of Theoretical Sciences, Germany

Correspondence: Roman Anton, The University of Truth and Common Sense, Department of Theoretical Sciences, Germany, Email mail.roman.anton@gmail.com

Received: February 02, 2018 | **Published:** March 07, 2018

Abbreviations: AI, Artificial Intelligence; CPU, Central Processing Unit; Depth, The deepness of the calculation often given in ply as the sequence of moves calculated ahead; ECO, ECO codes is a classification system of chess openings. ECO abbreviates Encyclopaedia of Chess Openings code. E.g. see free online versions of the Encyclopaedia;7 FLOPs, Floating Point Operations per second, a measure of computing performance; KPI, Key Performance Indicator; MTCS, Monte-Carlo Tree Search; NN, Neural Network; Ply, Half move in chess of either black or white, e.g. 1. e4 is one ply and 1. …e5 is a second play of the first move (here 1.); Root, the root of the search tree is the ply in which the calculation is started at a depth of zero; TPU, Tensor Processing Unit; TFLOP, Terra FLOPs

Introduction

Games are an ideal model system to research and develop artificial intelligence (AI). In chess, information is perfect and symmetric, and decision-making is logical and ranges from boolean to fuzzy logic, and is alternately challenged by sequential complex puzzles and non-trivial positions throughout the iterative game. This provides an ideal IT assay model system to develop and test AI and machine learning in the field of decision-making and machine learning via continuous improvement. It represents an ideal and major challenge for the development of AI software and general reinforcement self-learning programs due to the simultaneously high levels of game complexity and the vast amount of branching positions and a lower bound of about 10^{120} possible games according to the Shannon number. Hence, engines and players must somehow find the best ways and algorithms to manage certainty and uncertainty in their evaluations. A recent report by Silver and colleagues[1] has claimed a big major breakthrough that has been widely perceived as such by most of the noticed experts in the field and by the media. Many speak of a major milestone that has been reached by developing a pure machine self-learning algorithm that can outperform even the best conventional chess engine, which is an

equally significant and relevant benchmark. According to the authors, this world-breakthrough has been achieved by simply advancing and adopting the *Alpha Go* program to Chess, next to Shogi, a Japanese Chess form, and Go, games where it also won outstandingly.[1] In the game of chess, *AlphaZero* clearly outperformed *Stockfish8* in 100 games after only a short period of self-training, starting with the rules of chess and by pure reinforcement learning algorithms and simulated self-play. In the field of AU, this would represent a striking world-breakthrough of efficient and effective machine learning.

The claim has been made that already after 4 hours and 300k steps, *AlphaZero* would have been able to outperform *Stockfish8*, basically by starting and learning from scratch.[1] Hereby, *AlphaZero* would have only searched 80.000 positions per second, while *Stockfish8* would have searched 70 million.[1] Astoundingly, this is 875-times fewer calculations per second due to self-learning that would indicate another novelty of AI condensing game information into a human-like concluding, learning and intuition, as if variation pruning is optimized game by game. This would be the source code that everybody in the AI field was searching for, but has it actually been achieved in all relevant ways and not in a different way, e.g. via accumulated book-like knowledge? This work will re-evaluate if the claim of the best AI-chess-engine can really already be made by testing for reproducibility.

In a 100-game-match the new *AlphaZero* algorithm was shown to beat and obliterate *Stockfish8*, one of the world's best chess engines,[2] which won the 2016 TCEC computer chess championship. Today, an updated version of *Stockfish8* has been released, called *asmFish*,[3] which is again in December 2017 became the best engine of today, also in a direct comparison of all latest versions and other engines, and "in an official engine tournament". In the original publication, *AlphaZero* has not been subjected to an official independent testing[1] like on CCRL or tournaments, or elsewhere to the author's knowledge. Only 10 of the 100 games were made available,[1] which are reanalyzed in this work by multiple re-evaluations of every move played by *Stockfish8*.

Science and publishing have become very biased today,[4–6] especially since 2000, according to most estimates. It could be possible that an overclaim and many other systemic biases are inherent to all or almost all publications. Thus, as a first objective, this hypothesis shall be tested in the field of AI, a very clear-cut field of unambiguous IT and mathematics, by reevaluating if the best-engine-claim can already be made, or not, while also having a very constructive question in mind of how to improve AI and machine learning, as the second objective.

Materials and methods

a. Materials

Chess Engines: Stockfish7 64 bit and Stockfish 8 64 bit 4 CPUs and Fritz Chess software was used to recalculate the individual positions of the 10 example games,[1] as this was the only information that we have yet received from the publication. The engine *AlphaZero* in its latest version was not available to the author and all downloadable forms were not in the latest version. Opening book: Fritz11.ctg 06.11.2007 11:00 CTG-file 263.816 KB; Personal Computer: RAM: 3,44 GB usable; 64-bit operating system, x64-based processor, AMD A6-6310 APU Radeon R4 Graphics 1,80 GHz, HP Notebook; System: Windows 10 Home Edition; Chess Software: Fritz Twelve Beginner Edition 2011 (USK 0 years) ChessBase GmbH ISBN-978-3-86681-248-2; Screenshot and Office Software and further irrelevant tools; Chess games database were derived from 365Chess.com[7] and from Chessify with 6M chess positions.

b. Methods

The 10 example games that were published by Silver and colleagues of the outstanding Google Deep Mind Machine Learning Artificial Intelligence team[1] were transferred into PGN format using chessbase.com and uploaded into the Fritz Twelve Beginner Edition software. Chess Engines, *Stockfish7* and *Stockfish8* were uploaded to Fritz Twelve Beginner Edition software and used with the following parameters: hash table size of 2139 MB, 64-bit versions, permanent brain to enable in some cases pre-calculations, which is using the calculations that were performed during the opponent's and own time; Further, maybe less relevant *Stockfish8* 64-bit version parameters are only mentioned for a further precision: Contempt: O, Threads: 4, Skill Level: 20, Move overhead: 30, Minimum Thinking Time: 20, Slow Mover: 89, nodestime: 0, Syzygy Probe Depth 1, Sysygy Probe Limit: 6, activated Syzygy50MoveRule; *Stockfish7* 64-bit version original parameters included: Contempt: O, Threads: 4, Skill Level: 20, Move overhead: 30, Minimum Thinking Time: 20, Slow Mover: 84, nodestime: 0, Syzygy Probe Depth 1, Sysygy Probe Limit: 6; activated Syzygy50MoveRule; 1 or 4 CPUs (central processing units); only a few available details about *AlphaZero* and *Stockfish8* parameters were found in, and can be obtained from, the original publication of *AlphaZero*,[1] which poses another challenge to reproducibility. An automated and manual move-by-move re-evaluation screening was performed with Fritz Twelve and *Stockfish8*.

c. AI methods

MCTS is a best first search algorithm with four phases: selection, expansion, simulation, and backpropagation, based on random playouts. The original publication reveals that the Neural-Network parameters of *AlphaZero* "are adjusted by gradient descent on a loss function l that sums over mean-squared error and cross-entropy losses respectively,"[1]

$$(p,v) = f_\theta(S), \quad l = (z-v)^2 - \pi^T \log p + c\| \theta \|^2$$

„where c is a parameter controlling the level of L_2 weight regularization. The updates parameters are used in subsequent games of self-play".[1] This helps achieve *AlphaZero's* learning together with root visit counts.[1] *Stockfish8* uses advanced alpha-beta searches and alpha-beta pruning that focuses quiescent searches on the most relevant combinations. It begins calculations based on the root of the current position, which is set to root ply zero.[2] Stockfish's more static AI is a non-learning algorithm that uses specific evaluation algorithms are used for opening, middle games, and endgame, and special positions.[2] Still, hash tables bear the potential to enable some side-effect short-term memory learning under permanent brain conditions[2] that is used as a method in the reevaluation of the example games.

Results and discussion

Re-evaluation of AI match statistics

In the game of chess, *AlphaZero* has recently been reported to outperform *Stockfish8*, after only 4 hours of learning via self-play. *AlphaZero* could outperform *Stockfish8* in an initial experiment already after 300k steps. The centerpiece of the claim is a 100-game match in which *AlphaZero* won against *Stockfish8*. Here the training phase has resulted in about 700k steps of tabula rasa learning under

tournament time controls of 1 minute per move, each MTCS was calculated on a single machine with 4 tensor processing units (TPUs), training started with randomly initialized parameters using 5.000 first-generation TPUs to generate self-play games and 64 second-generation TPUs to train the neural networks.[1] TPUs are built and assembled in a specific machine learning hardware configuration of integrated circuits developed by Google (ASICs) and mainly for AI applications.

In this 100-game match, *Stockfish8* has been? reported to have lost 28 times, 25 times with black and 3 times with white, while *AlphaZero* did not lose any game, and won 28 times, and drew 72 times, 25 times with white and 47 times with black (taken from Table 1 in Silver et al.)[1]. All of the 100 original games[1] were played in a direct comparison as an engine match of *Stockfish8* (official Linux release, 64 CPU threads, hash table size of 1GB, and according to the authors, at their strongest skill level) versus *AlphaZero* (1.1) 04.12.17 using a single machine with 4 TPUs,[1] which is, in fact, a comparison of very different software and hardware configurations.

Normally, on average, top games of the world's best 2003 different engines result in a win distribution of white-draw-black (Figure 1) of about 34,3%-40,3%-25,4% favoring white with 54,4% versus 45,6% for black.[3] *AlphaZero*, however, reveals a new, unexpected 'atypical' distribution pattern of 25%-3%-72% favoring white by 61% versus black by only 39% from the perspective of *AlphaZero*. From Stockfish's perspective, the pattern appears even more 'atypical' as 0%-72%-0% (Figure 1), only achieving draws against the learning *AlphaZero* engine. Put in other words, *AlphaZero*, as white, would have better exploited the first-mover advantage than normally seen by the sum of top engines (Figure 1). To better make use of a starting position, it would be required to figure out the most forcing lines in the widely 'unexpectable' long-run of the game. This would indicate that *AlphaZero* had learned and abstracted rules from its training phase and derived an understanding-like intuition by calculation higher quality lines and trees, as it is known to perform 875-times "fewer calculations per second".[1] Where should the performance stem from? Could a better focus on key lines also be achieved by self-learned book hints has AI machine learning gone perfect, human, or both?

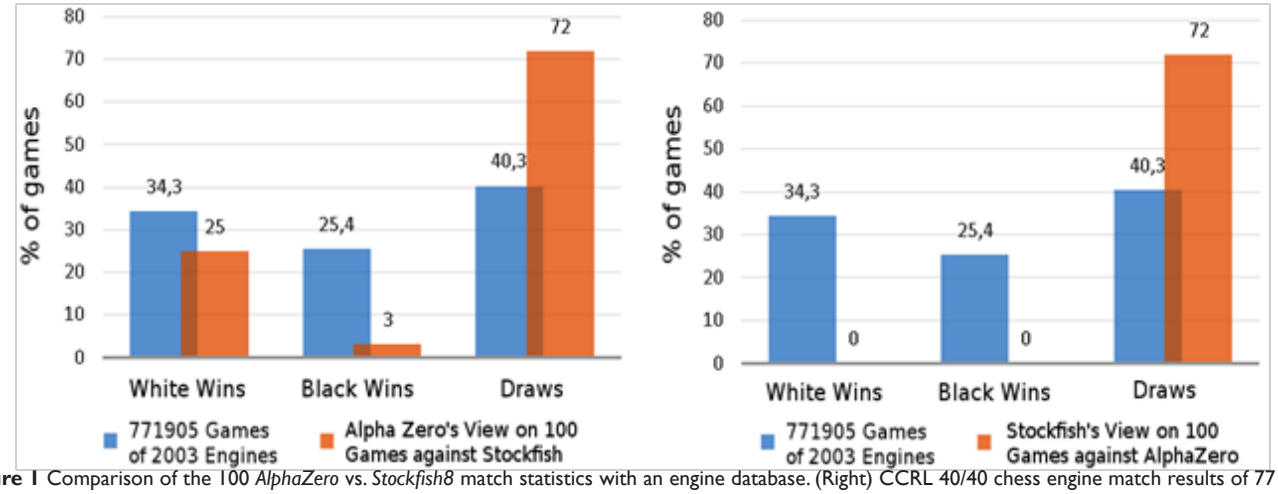

Figure 1 Comparison of the 100 *AlphaZero* vs. *Stockfish8* match statistics with an engine database. (Right) CCRL 40/40 chess engine match results of 771.905 games and newest 2003 machines versions, starting from December 2005 till December 2017, compared with 100 published games of *AlphaZero* versus *Stockfish* version 8(Table 1, Silver et al.,[1] 2017). Note the very strong shift from black wins into draws in comparison to the 2003 top engines data. (Left) The black-vs-white-score distribution pattern is fully absent if the statistics are viewed from the perspective of *Stockfish8* that did not win one game and drew 72 times. This distribution of white versus black wins is unexpectedly high and would suggest that a starting position could be more exploited than it is today by humans or most engines. This strong exploitation of a beginner advantage can be unexpected for a null-sum game.

Re-evaluation screening for atypical moves

In a next step, all 10 example games of the 100-match engine contest of *AlphaZero* versus *Stockfish8*[1] were reanalyzed (Figure 2) (Figure 3) to identify potential causes or reasons for this atypical black-vs-white winning pattern (Figure 1). This reevaluation should reveal if there is a potential reason or bias why the best engine around that time, was unexpectedly outperformed so strikingly and if *AlphaZero*'s strengths and Stockfish's weaknesses can be found.

Using *Stockfish8* for the reanalysis of all moves of the 10 example games between *Stockfish8* and *AlphaZero* could identify potential blunders of very few but likely decisive *Stockfish8* moves during the 100-games match (Figure 2) (Figure 3) (Table 1). This indicates that *Stockfish8* might have unusually underperformed in ca. 1 move per game.

Potential *Stockfish8* weaknesses were identified via automated move-by-move evaluation screening of all 10 example games. An evaluation landscape of all 10 example games is given in Figure 2 & Figure 3 in centipawns (cps). These move-by-move cp-evaluation landscapes were then used for manually screening of atypical or weak moves.

While it seems likely that potentially underperforming or biased *Stockfish8* moves, which are termed potential *Stockfish8* blunders in Table 1 (i.e. potential or debatable *Stockfish8* weaknesses or atypical novelties), seem not to be a robust stable first choice of a normal operating *Stockfish8* engine under the re-evaluation settings, there is still a remaining small probability that *Stockfish8* might have played them under different parameters or conditions, which were not entirely defined in the publication. Still, most moves will likely reproducibly vary also if another re-evaluation would try to reproduce some of the original moves. Most if not all of the highlighted blunder moves in Table 1 raise new inevitable questions about the reproducibility of this 100-match game result and the AI-performance in chess, as at least one move per game could be indicated as a potential blunder or

bearing an atypical likelihood, and others potentially also could. At first glance, it seems possible that these unexpected Stockfish blunders or novelties (Table 1) would not have been very typically played in an official engine tournament under controlled engine settings, best engine parameters and full information details of the hardware and software configurations, with an independent referee, and without conflicts of interests or systemic biases.[4,8]

Table 1 *Stockfish8* re-analysis of all Stockfish 8 moves in the 10 example games, as a summary table; best estimates of all re-evaluations are given, no liability assumed as some engine parameters were elusive[1]

Game	Stockfish 8 blunder?	Decisiveness	64,2139MB Hash^{o}	64, 1 GB Hash 1 or 4 CPUs^{o}	Stockfish 8, first choice^{o}
1	35. Nc5?	yes	blunder (N>4)	blunder (N=3)	35. Rc1!
2	34.Qd3?	yes	blunder (N>4)	blunder (N=3)	34. Nd2!
3	15. (…) Re8?	yes	blunder (N>4)	blunder (N=3)	15. (…) Qd8!
4	13. (…) a5?	yes	blunder (N>4)	blunder (N=3*)	13. (…) Nb6?!
5	22. (…) Nc5?	yes	blunder (N>4)	blunder (N=3)	22. (…) hxg5!
6	34. (…) Nd4?	yes	blunder (N>4)	blunder (N=3)	34. (…) b5!
7	31. (…) Rfe8?	yes	blunder (N>4)	blunder (N=3*)	31. (…) h5!
8	23. (…) Qd8? 35. (…) Qf7? (…) 36. c3?	yes, yes, yes	blunder (N>4, N>4, N>4)	blunder (N=3, 3, 3)	23. (…) Ra2!, 35. (…) Qe8!
9	17. (…) b6?, 40. (…) b3?	yes, no	blunder (N>4*, N>4)	blunder (N=3*, 3)	17. (…) Bb4! 40. (…) Bc8!
10	23. b5?	yes	blunder (N>4)	blunder (N=3*)	23. Kg7!

^{o}1 min calculation time no pre-calculation * can seldomly be a non-blunder or has remaining likelihood to be a non-blunder

Note: In all 10 available example games, a move-by-move-screening made it possible to identify potentially atypical moves or potential *Stockfish8* weaknesses. These *Stockfish8* game-moves could not be clearly reproduced by *Stockfish8* re-evaluations in several attempts under related engine parameters. This indicates that *Stockfish8* might have been outperformed by *Alpha Zero* in the 100-games-match due to an untypical underperformance or weakness of *Stockfish8* during the non-controlled, non-independent, and non-tournament-like experimental conditions that potentially bear more risks of biases.[4] Games, game numbers, and notation were taken from the original publication.[1] See supplementary material link and Figures 2 & Figures 3 for more details and information on the chess moves and the respective chess games for an independent engine reanalysis.

Lack of reproducibility that AlphaZero is the best engine

As a result, this work must suggest that it is not finally proven if *AlphaZero* has already achieved to be the best chess engine worldwide, which should be reassessed in a new and official engine tournament match. Nevertheless, *AlphaZero* has achieved a very strong play that could be better than *Stockfish8* and further interesting findings.[1] Access to the engine and all 100 games, however, is still needed for a best engine claim, as much as an independent tournament engine match, and reproducibility is not assured, like for most scientific publications of today, due to a lack of the incentives. Potential *Stockfish8* blunders might represent what was recently termed strategic biases[9] in all sciences.[4,5]

A claim to be the best angine must anyhow be proven under official settings. *AlphaZero* seems to have the potential but will it win against a normally performing Stockfish or now *Asm Fish*? The best chess players also have to play more than one match alone at home. Thus, it is absolutely required to make the *AlphaZero* engine available as an open source, like it has been done with most of the top-rated engines including the latest versions of *Stockfish*, Houndi, and others, or to go with it to official engine tournaments like the CCRL or alike. Once a first position can be shown i.e. in an independent engine match, a claim can be made, otherwise some doubts will or can prevail (Figure 2) (Figure 3).

Comparability challenges of AI matches

How to compare AI apples with AI pears? This becomes a more technical question that must be resolved in its details in order to ask the right experimental questions. It becomes clear that the engine parameters, hardware, and software configurations are also another key part of these questions. While hardware parameters are at a high level of performance, they could be resource factors in saturation and less relevant for the final engine performance, which has been recently suggested also by chess grandmasters. This can be but could be also different due to the very different approaches used. For example, the speed of the calculation could have an impact on the performing operating chess engine software and its timing of the respective first choice moves at a given time interval. To test IA software ability, self-learning or pre-programmed, it would be important to keep the hardware and all variables constant, in order to test for engine abilities. If this is, however, not possible, one still could compare two systems of software and hardware on the board. But here the KPI calculation/s could potentially bear a different meaning.

How to achieve an equal firepower or equal calculation power between so different approaches? Are we comparing software or hardware or, in fact, always both? *AlphaZero* uses special hardware developed by Google that was running on a single machine with 4 TPUs, which are Google's own Tensor Processing Units rather than the available CPUs, Central Processing Units. In the NN case of second-generation TPUs this would correspond to 180 TFLOPs or TeraFLOPs, 10^{12} floating point operations per second – at Google, one new cloud TPU delivers up to a 180 TFLOP floating point performance for machine learning engine models - so it is difficult to compare.

On the other hand, the best CPU computers do normally not reach this very high TFLOP-level of computation power and one could assume that *Stockfish8* was running on a 100-fold slower hardware, but these details have not been clearly specified in the publication.[1] In light of this, it becomes questionable if *AlphaZero* really derived

and abstracted findings in a "human-like" intuitive manner, as the hardware is probably more than 100-times faster,[1] as compared to a single machine with "64 CPU threads" and a hash size of only 1 GB.[1] This view is further sustained by table 1, which identifies many underperforming moves of *Stockfish8* on a much slower laptop pc with a lower-end mass product standard's minimal configuration (see also materials and methods).

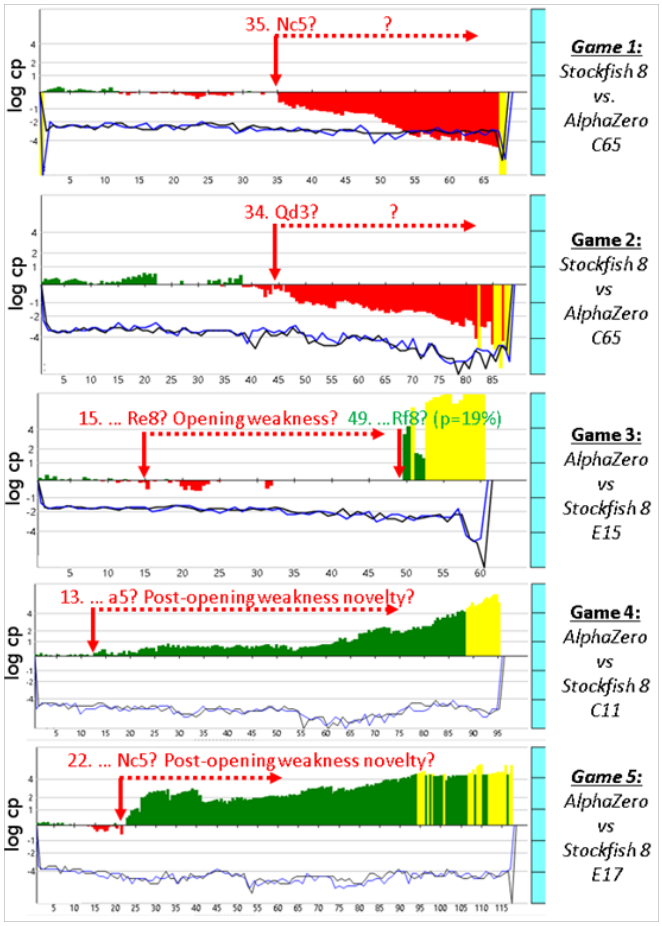

Figure 2 Stockfish 8 re-analysis of every move in example game 1-5. Engine parameters: 64-bit, 4 CPUs, 2139 MB. Re-evaluation profile of every position reveals the balances during the game and decisiveness of the potential biases or blunders. Green: Advantage for white; red: advantage for black; yellow: goes beyond scale; the scale is given in centipawns, which equals 1/100 of a pawn on the vertical axis; move number is given on the horizontal axis; the search depth of the program is given below the horizontal axis. The game notation and re-analysis evaluation of respective moves are found in the supplementary material link. Please note several post-opening weakness novelties that are not very common for human or Stockfish in the games in which Stockfish plays black.

Despite these points, there have been not many or no obvious experts that were allowed to reveal this to the media or to the public. Contemporary censorship of all hidden experts opinions causes extreme biases and dominant one-sided views.[4,5,8] It is almost impossible to publish independent views in today's journals.

Secondly, from the software side, there are the same potential issues of a lack of computational comparability. If *Stockfish8* utilizes 70 Million (M) calculations per second and *AlphaZero* only 80 thousand positions per second[1] the question arises of how, and how evenhandedly, "positions per second" is defined. What are the 180 TFLOPs good for if it calculates in sum so much less, are calculation results stored on hard drive books? Why do you need a 100-fold better machine if you need 875-times fewer calculations per second? What is the remaining hardware and software doing and calculating? Is this a "more human-like approach" or a lack of comparability?

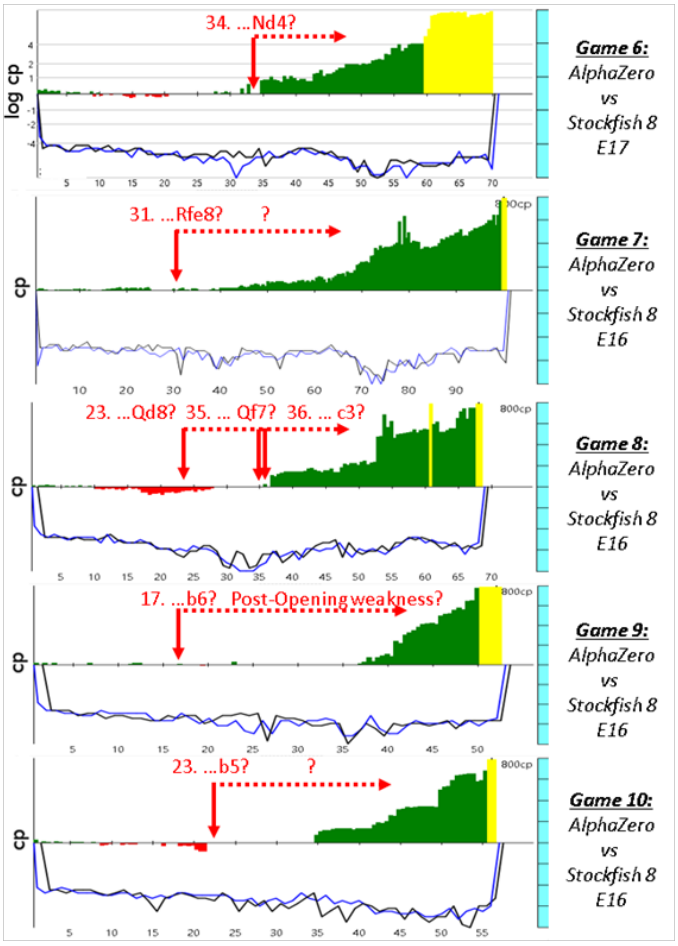

Figure 3 *Stockfish8* re-analysis of every move in example game 6-10. Engine parameters: 64-bit, 4 CPUs, 2139 MB. Re-evaluation profile of every position reveals the balances during the game and decisiveness of the potential biases or blunders. Green: Advantage for white; red: advantage for black; yellow: goes beyond scale; the scale is given in centipawns, which equals 1/100 of a pawn on the vertical axis; move number is given on the horizontal axis; the search depth of the program is given below the horizontal axis. The game notation and re-analysis evaluation of respective moves are found in the supplementary material link. Please note several post-opening weakness novelties that are not very common for human or Stockfish in the games in which Stockfish plays black.

Machine learning bears very powerful possibilities and still has a very big future, and *AlphaZero* is at the very heart of these new innovations. Even if the differences, in a 100-game match were only slight or equal, one could still speak of a great breakthrough and milestone in machine learning history and Neural Networks (NN) due to the efficiency and effectiveness of machine learning. But this evidence is elusive until it is reproducible, which bears major implication for all sciences, as the majority of all publications are biased[4,5,9] and cannot be reproduced in any way. 90% of scientist agree that there is a reproducibility crisis and only 3% do not see it.[10]

Citation: Anton R. Reevaluation of artificial intelligence engine alpha zero, a self-learning algorithm, reveals lack of proof of best engine, and an advancement of artificial intelligence via multiple roots. *Open Acc J Math Theor Phy.* 2018;1(1):22–31. DOI: 10.15406/oajmtp.2018.01.00005

A closer look at the general differences by comparing the overall hardware and software features of the individual approaches of the two different engines might help to estimate the level of comparability (Table 2).

AlphaZero uses MTCS (Monte-Carlo Tree Search) instead of Stockfish's alpha-beta searches with domain-specific enhancements[1] that most of all conventional non-learner AI engines are using (Table 2). The search focus is achieved by deep NN for *AlphaZero*, while it is alpha-beta pruning in *Stockfish8* that deprioritizes those by internal score disproven tree branch variations or lines. Non-linear approximating deep NN positional evaluation is used in machine learning while Minimax evaluation is in place for most of the top-level competitive traditional engines like *Stockfish8*. All further software differences are listed in Table 2 to give an overview. During the learning phase, each MTCS used 800 simulations.[1] Learning happens by selecting moves in proportion to root visit count.[1]

Table 2 Comparison of IT hardware and software parameters of *Stockfish8* and *AlphaZero*[1]

AI Parameters	AlphaZero	Stockfish 8
Hardware	180 Tf LOPs (4 TPUs)	64 CPU threats
	">100 times faster"	1 GB Hash Size
Search Engine	MCTS	alpha beta
	80.000 positions/s	70.000.000 positions/s
	averages approximations	explicit calculations
Search Focus	deep NN focus	alpha-beta pruning
	quality searches	quantity searches
Evaluation	deep NN evaluation	Minimax, handcrafted
	averaging of position for	evaluation of subtrees
	evaluation within subtree;	without averaging;
	non-linear function	linear function
	approximation	approximation
Root of Subtree	averaged approximation errors	explicit approximation errors
	cancels out	fully represented at root of subtr ee
	averaged approximation of best moves	averaged approximation of best moves
	might cancel out but in less noise	might not cancel out but in more noise
Availability	hardware is not readily available nor standard	hardware is available and standard
	software is not available as fully open source	software is available as fully open source
Reproducibility	is not reproducible by others, no tournament	is reproducible by others, tournament
	engine reproducibility per move is not known	reproducibility per move is medium to high
Learning	general reinforcement machine self-learning	programming, trial and error
	no consecutive engine version matches	consecutive engine version matches

But how long did the learning phase last before the 100-game match has started? The abstract reveals it took *AlphaZero* 24 hours of learning to outperform *Stockfish8*.[1] In the paper, we also find a Figure 1 that states that it only took 4 hours and 300k steps based on artificial Elo-rankings. Next, it says that "We evaluated the fully trained instances of *AlphaZero* against *Stockfish8* (…)",[1] which, would indicate that *AlphaZero* might have trained for longer, for basically an unspecified time at least in theory. It seems to be not fully specified, is it 4h, 24h, or years – against itself, or did it also learn by playing *Stockfish*? What was the role of the opening books?

Multiple roots can boost artificial intelligence

Did a human-like thinking evolve or develop during this training period is thus an important and intriguing question that is often and widely asked by experts and the media. How does *AlphaZero* learn top-level chess play so fast? How can it restrict the number of positions per second to 80.000 while *Stockfish* utilizes 70 million per second? *Stockfish8* reduces and extends the search depth by alpha-beta pruning for bad or promising variations, respectively. But what about the search depth of *AlphaZero*? The answers could depend on search depth that is not clearly stated in the publication.[1] For example, what if calculation means sequence tree length of 50 for one machine and 25 for the other? Are all simulations played to the foreseeable end? Search depth is the number of half-moves, so-called "plies", the search function nominally looks ahead, between the "root and the horizon node" at the depth of zero where "quiescence searches" are performed. As an analogy, these quiescence searches maybe resemble what is called in the human chess school of thought the "candidate moves", lines and variations in trees.

Search depth could be thus a very important and crucial aspect of AI and machine learning. It is frequently viewed to be a basic feature of intelligence to focus on most promising facts and moves and to exclude zero-chance opportunities with certainty. *Stockfish* uses killer-

move and counter move heuristics and *AlphaZero* semi-random tree play out evaluation and scoring at the root and subtrees[1] that might be as crucial as search depth.

If entire games per move are calculated one would reach a maximum tree variation depth but this is only possible for high-quality candidate moves or via super-calculation-power in TFLOPs. For both engines, the quality of candidate moves is crucial, for quiescent searches at the root for *Stockfish* or root visit scores and subtree for *AlphaZero*. The centerpiece of quality begins at a root for both engines. The root is the basis for both approaches.

But this would mean that you can improve the performance of artificial intelligence engines by optimizing the candidate moves for the quiescent searches (for Stockfish-type engines) or the tree root visit scores (for NN-engines like *AlphaZero*). In both cases, *Stockfish8* and *AlphaZero*, the performance could be theoretically optimized by stabilizing the root tree via multiple roots: more roots more second order stability but also not too many, only a few as a test, to not drop calculation power and depth. Whenever an engine searches for candidate moves or top evaluated trees it has to perform a time-consuming computing work in an unpredictable landscape of future move events that bears both very simple and also very difficult variations in a vast amount.

To best navigate through this extraordinary tree landscapes universes the engine must overcome the resistance of these difficulties that require longer computing times. One simple way to increase the stability of the root and trees would be to allow the engines to have multiple roots at once in a testing manner and not just one root, as one root could be less stable and more prone to fall into a future "difficulty trap" that might be seldom but is possible. More roots at the artificial depth of zero (only one root would be truly and contemporarily zero) would stand on multiple feet like a chair or table needs more than one table leg roots. Candidate moves and tree visits would be simply harmonized in a permanent brain environment in order to not oversee a very important variation or line. Seldom, one root might be keeping the machine focus too busy in calculating the many very difficult variations and tree branches. Here, a short second or third root might provide a new additional landscape, a short engine-restart from an adjacent position, to build on and to work within the searches, as navigation through multiple roots might advance solution finding by helping to find the potentially overseen important lines. This assumes that *AlphaZero* also uses an evaluation function in its play and not only *de novo* derived chess understanding, feel free to correct the author if this is different, it is just not imaginable for the author that an NN could do this without a live evaluation function. To find empirical indicatory evidence for the theory shown in Figure 4 that could help to boost AI and AI predictive "move searches", all *AlphaZero* games were screened and investigated with regard to this question using a permanent brain function combining multiple roots for a position that could serve as a read-out.

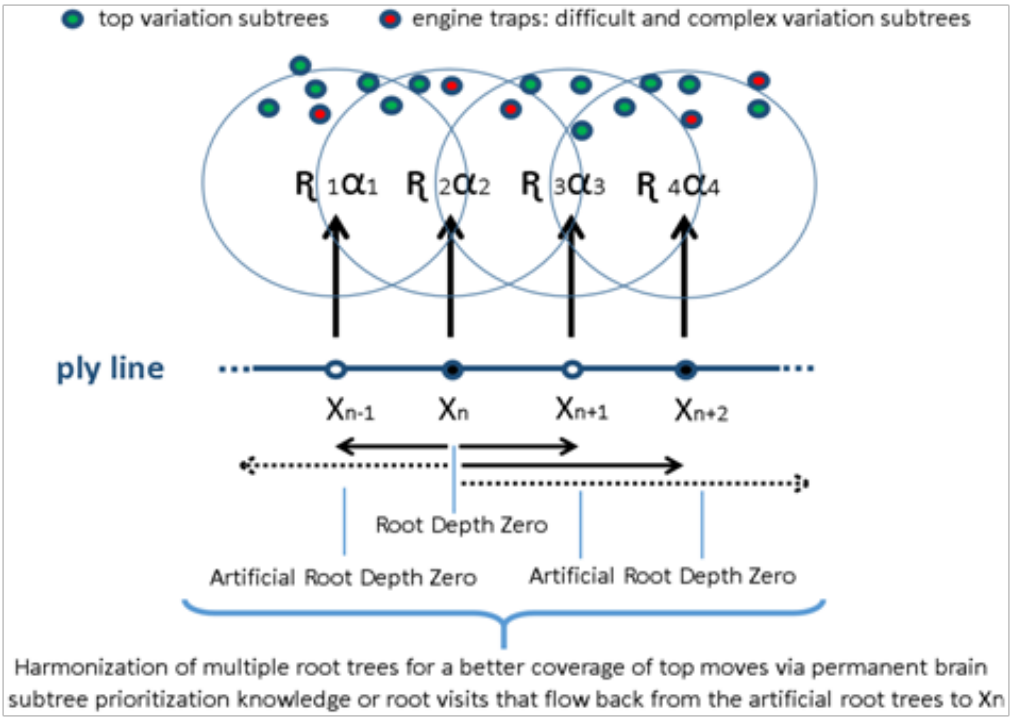

Figure 4 Theoretical model for AI engines (e.g. Stockfish and *AlphaZero* in the game of chess) to use more than one but a few multiple staggered roots to provide more stability in the calculations stemming from multiple starts that improve the second-order probabilities to identify the best move in some circumstances of future variations. Every AI calculation that starts at a set ply depth of zero can deal with extreme quantities of variations and high levels of unpredictability. In the theoretical model, if multiple roots are used and computing is harmonized subsequently it might become more likely to not oversee a best move and subtree as new screening approaches arise by starting at different levels e.g. for the evaluation and quiescence searches. R: hypothetical resistance of the future landscape of heterogeneous complexity trees with both simple and difficult variations. α: pseudo-adjustment variables for the engine approach whether in a semi-random tree-walk or in handcrafted heuristics or alike. Noteworthy, adjacent roots can be added from the past and future, as both could provide more stability for AI.

Citation: Anton R. Reevaluation of artificial intelligence engine alpha zero, a self-learning algorithm, reveals lack of proof of best engine, and an advancement of artificial intelligence via multiple roots. *Open Acc J Math Theor Phy.* 2018;1(1):22–31. DOI: 10.15406/oajmtp.2018.01.00005

Empirical test if multiple roots boost AI

To empirically prove the hypothesis (Figure 4) of the theoretically derived technical AI idea how to improve both AI in chess engines, all games were screened for a good read-out position to test the hypothetical model. Example Game 3 of *AlphaZero* vs *Stockfish*[1] was co-incidentally identified re-evaluated by *Stockfish8* 64-bit 4 CPUs and 2139 MB hash: in move 49 *Stockfish8* has originally played Rf8 in the 100-game-match versus *AlphaZero*.[1] At first glance, this seems not to be the very best move even for a human player because, someone might argue, as it further and unnecessarily crams the king and queen and obviously suffocates their flexibility and options as it might to much limit their potential future moves (Figure 5). But this seems not to be a Stockfish8 blunder but a real *Stockfish8* move that drives a loss of the game. In fact, in a 1-minute per move calculation, *Stockfish8* could independently reproduce to find Rf8 as the first choice with a significant likelihood of ca. 19% of all cases (N=16), an unexpected reproducibility that has led to a new finding. The obviously better move 49. …Kf8! is not always found by *Stockfish8* (64-bit, 4 CPUs, 1 or 2139 MB hash), which has played the right move Kf8 in only about ~50% of cases (Figure 5). Now, if multiple roots are simulated by a permanent brain function and a pre-calculation is performed of the X_{n-1} root ply, it yielded a significant improvement in the overall Stockfish8 performance in this position and also in several other games and positions (not shown).

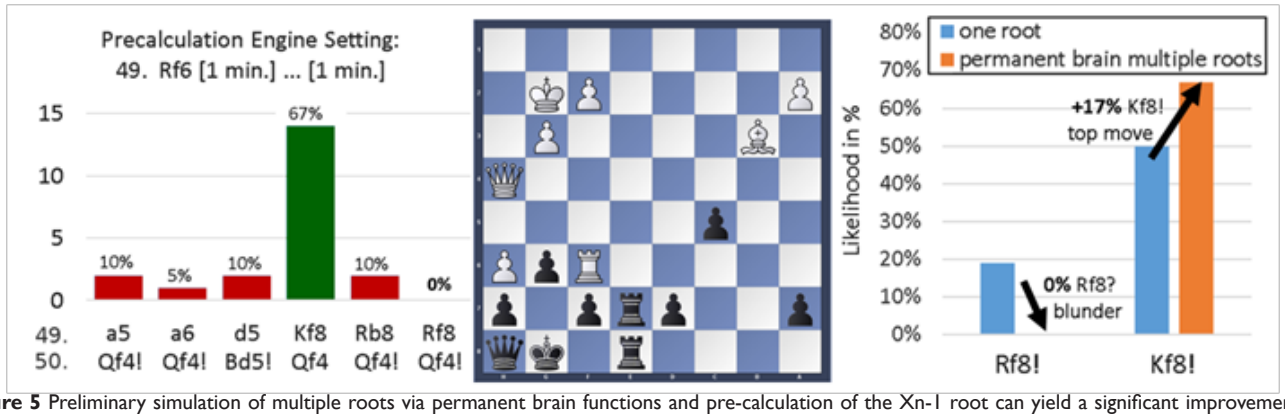

Figure 5 Preliminary simulation of multiple roots via permanent brain functions and pre-calculation of the Xn-1 root can yield a significant improvement in *Stockfish8* engine performance in a read-out position (49./50.; game 3), as sometimes variations can be disproven by computing subsequent plies. This might suggest new benefits for AI in general if multiple roots are used that could simulate an effect of using multiple cores. (left) pre-calculation of move 49. Rf6 using a permanent brain function can yield an improvement of *Stockfish 8* 64-bit 4 CPUs moves. (center) board at position 49. Rf6 of game 3.[1] (right) multiple roots could slightly help to find the best move (p=+17%) and it also helps to faster disprove a losing move, here Rf8 (p=0%).

Still, this "AI boost effect" could be also simply explained by a longer calculation time and/or by a single root and variation trap in the future (the hypothetical model in Figure 4). To exclude this possibility, a longer calculation time was tested without the use of multiple roots and did not improve the result (N=4; 0% Kf8 2 min., 1 root, no permanent brain): thus it is not the time that has improved the predictive AI search but "multiple roots".

More calculation time also did not play a role in several additional positions in which *Stockfish8* calculated many 0.00-evaluation-type candidate moves and lines (not shown). In this very specific position (move 49. … of game 3), *AlphaZero* has seemingly uncovered an interesting roughly 50%-weakness of *Stockfish8* that might help the developers of *Stockfish*, *AlphaZero*, engines and AI, to improve future versions in some way, which cannot be finally proven here. Also, placing the queen earlier on 45. … Qh8 could be questioned. Prolonged indifferences during the calculation (0.00) seem to cause this ca. 50%-level type of a natural Stockfish error. These indifferences stem from an incomplete candidate move portfolio while the quiescence search is trapped in a difficult-to-assess variation and seemingly a wish to delay a rejection of a previous estimate (Figure 4), and multiple roots could be the solution here.

As it is not the time given for the calculation, but the root start of the quiescence search with a potential lack of all candidate moves and lack of all evaluations (often seen as 0.00 values), it is likely that a switching to multiple roots could solve this instability issue of such and other AI searches to approach a difficult tree landscape. This can be shown in both, (a) in the pre-calculation (X_{n-1}) and it can be also shown in (b) the post-calculation (X_{n+1}) mode using permanent brain linked roots (Figure 6). Multiple starter simulations of adjacent positions could thus improve the searches. Empirical indicatory evidence was found for the model but it requires further testing.

Figure 6 reveals the benefit of including a future root, Figure 6A shows that move 49. … Kf8 would be the most practical and probably best solution as identified by *Stockfish8*, a human player, and *asmFish*, the recently best Stockfish engine. Without any pre- or post-calculation, *Stockfish8* can be sometimes indifferent as seen by 0.00 values for all variations and it oversees the best solution, i.e. the Kg8 line. However, if a pre- or post-calculation is added via the permanent brain function, *Stockfish8* is almost fully able to see that Rf8(?) is an unfavorable move and increases the likelihood of Kf8 up to 100% (Figure 6B). Hence, it seems to be a quiescence initialization weakness causing an alpha-beta pruning weakness to find the most promising lines. AI using multiple roots at a depth of -1, 0, +1, +2 thus "can much advance the search results", which might be interesting for Google too. Google has actually already started doing this by integrating our search preferences for advertisements that could be further advances in this AI way using multiple roots. Thus, the model in Figure 4 is helpful, as predicting future moves and pre-calculations could help *Stockfish8*, *AlphaZero*, chess and search engines, and AI, in general, to find the best move in an indifferent way: multiple roots can boost the performance, which is supposed to be just a draft not an "overclaim", but the "overclaim" mechanisms of publishing become apparent. The future will tell if multiple roots can boost AI performance, and only the future can tell if *AlphaZero* is the best engine one may argue?

Citation: Anton R. Reevaluation of artificial intelligence engine alpha zero, a self-learning algorithm, reveals lack of proof of best engine, and an advancement of artificial intelligence via multiple roots. *Open Acc J Math Theor Phy.* 2018;1(1):22–31. DOI: 10.15406/oajmtp.2018.01.00005

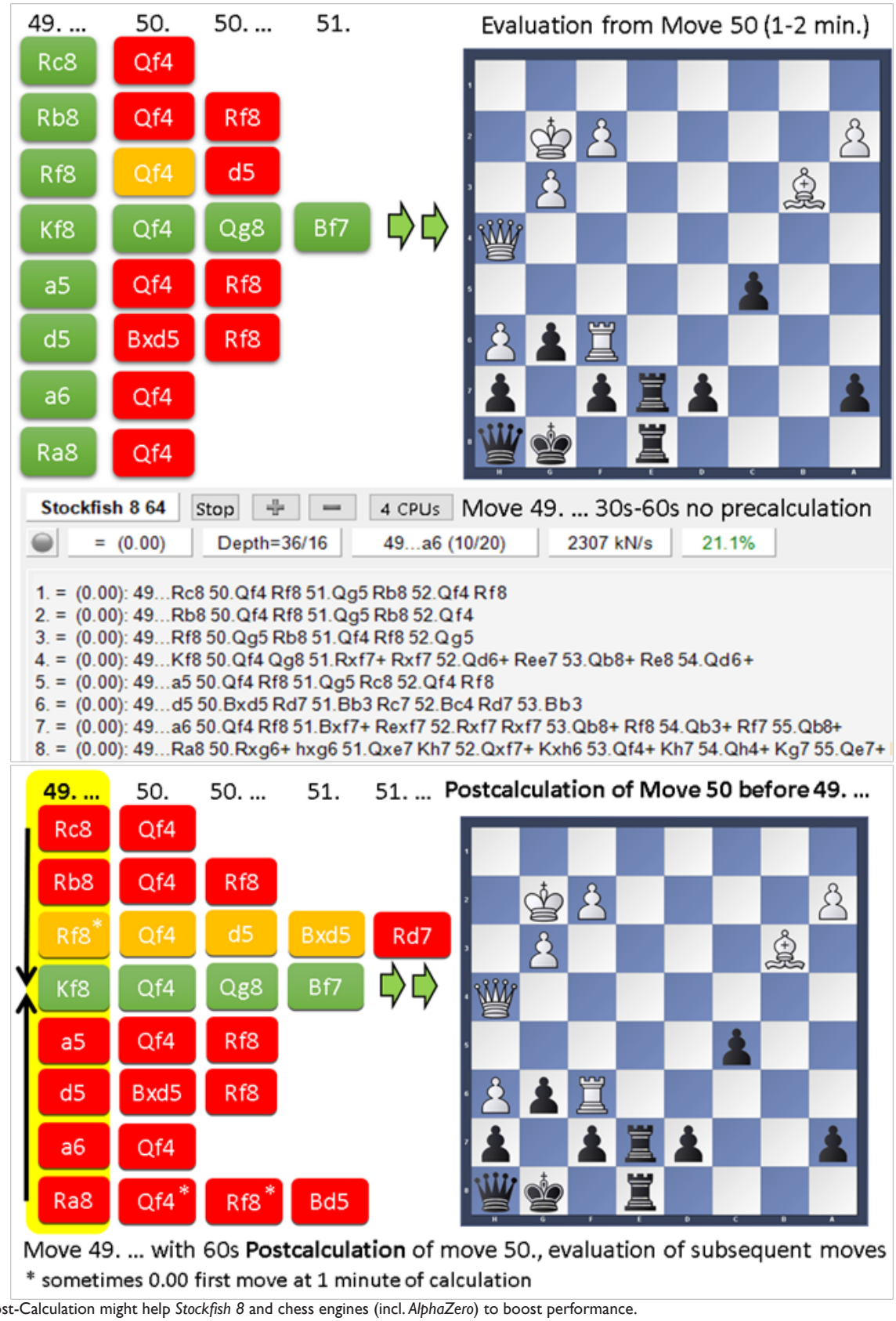

Figure 6 Post-Calculation might help *Stockfish 8* and chess engines (incl. *AlphaZero*) to boost performance.

Artificial intelligence and the openings

Finally, also the openings of the chess game are crucial and decisive and an interesting case to study AI and searches of the hay needle in vast amounts of complexity and universes of combinations. Table 1, Figure 2 & Figure 3 have revealed that *Stockfish8* sometimes had an unexpected opening or post opening weakness. Hence, a final proof is still missing that *AlphaZero* can beat *Stockfish8*'s high-performance play, and it would be important to retest this in a reproducible and unbiased fashion in an independent engine match tournament. But what about *AlphaZero*? How well did *AlphaZero* perform in these decisive openings? The answer is very well, judging from a human perspective that is built upon a human-centric game database of the best human master games played.

The winning chances of each opening are summarized in Table 3. Interestingly, in the 10 example games given, *AlphaZero* could best win as black in the C65 openings of the Ruy-Lopez/Berlin Defense, which could indicate a weaker start for white but not necessarily. Of the 8 winning games as white *AlphaZero* favored the Queen's Indian Defense (Table 3). What do we know from human games in these opening positions is also given in Table 3. For instance, in game 3, the opening can be viewed up to the move 15., there were 13 games in the human chess games database: 38% were won by white, 46% were a draw, and 15% were won by black. *AlphaZero* was competing with *Stockfish8* as "fully trained instance(s)" (Silver et al.,[1] 2017, page 4), however, officially without any opening book knowledge, as all was learned by self-play reinforcement machine learning of the deep NN.[1] The question arises, how well do the 10 *AlphaZero* opening choices (Table 3), only derived by *AlphaZero* e.g. the root visit counts, overlap with the best choices of top-level human openings i.e. the best win-chances know so far?

Table 3 Potential Opening Win-Chances-Bias in the 10 *AlphaZero* vs. *Stockfish8* games[1]

Game	white	draw	black	N Games	ECO Code	human book move	AlphaZero
1	21	62	15	32	C65	7. … 0-0	black 0-1
2	0	100	0	2	C65	8. … Bd6	black 0-1
3	38	46	15	13	E15	15. Qg4	white 1-0
4	60	20	20	5	C11	9. Bd3	white 1-0
5	41	41	16	174	E17	11. e4	white 1-0
6	41	41	16	174	E17	11. e4	white 1-0
7	33,3	66,7	0,0	3	E16	11. b4	white 1-0
8	66,0	0,0	33,0	3	E16	9. … Ne4	white 1-0
9	35,0	42,0	21,0	14	C11	6. … cxd4	white 1-0
10	46,0	35,0	18,0	512	E17	7. d5	white 1-0

The answer to this question is given in Figure 7 to some global extend: In the 8 games as white, *AlphaZero* was clearly selecting the better openings – at least from a 100-years of accumulated human chess knowledge perspective – and also compared to *Stockfish8*, irrespective of all the post-opening weakness novelties listed in Table 1. Moreover, the heightened opening chances for white and the lower wining chances for black can be shown to be significant ($p>0.05$, $N=8$) if compared to all games in the database ($N=3.382.780$). This shows that *AlphaZero* would have learned top openings that a century of all human top players had crystallized and precipitate over time. Or were all human games always inspired by engines that cause the overlap? *AlphaZero* would have reinvented this Gordian wheel of openings in breakthrough time of only hours by playing nobody more intelligent than itself, starting from scratch. One might argue that an evaluation function is still at work and that *Stockfish-8*'s AI does not even have to train at all to win against maybe all top players. But selectivity for these openings could have many reasons. Still, how can such a human-like book-knowledge choices pattern by Alpha be explained by a self-learning algorithm? Would the same openings be discovered by machines, or has there been an assistance? Or are these openings really the best for both human and engine play? Playing an engine is different than playing a human opponent, no?

Supposedly, *AlphaZero* did not use any opening book knowledge and statistic, it has gained an excellent feeling for openings in rather very short time, but how long were the fully trained instances really trained? This is not precisely explained in the publication[1] but 4h or 24h would be very short for this. The question arises whether *AlphaZero's* openings in these example games are significantly better than the global book win-statistics.

Figure 7 reveals a striking difference of 45% winning chance for white (*AlphaZero*) versus only 17,4% winning chances for black (*Stockfish8*), already after only 10 moves on average. If *AlphaZero* really did not use any opening book knowledge, this would mean that the 3.3 Million grandmaster games were not played for anything. Both machine learning and all human top games overlap in the most promising openings. But what about all the other known book openings e.g. with good winning chances for white? We cannot say at this point in time. In the learning phase of *AlphaZero*, there was a growing interest in E00 Queen's pawn games and a decline in Spanish games (Table2 of Silver et al.[1]). Still, engine learning developed a pickiness of promising human openings.

In summary, from an anthropocentric view, *AlphaZero* did extremely well in the openings and post-opening phase and *Stockfish8* was very slightly but decisively underperforming; this could be theoretically a human-bias. Stockfish, for example, in tendency fails to find the best human openings without the use of any book knowledge.

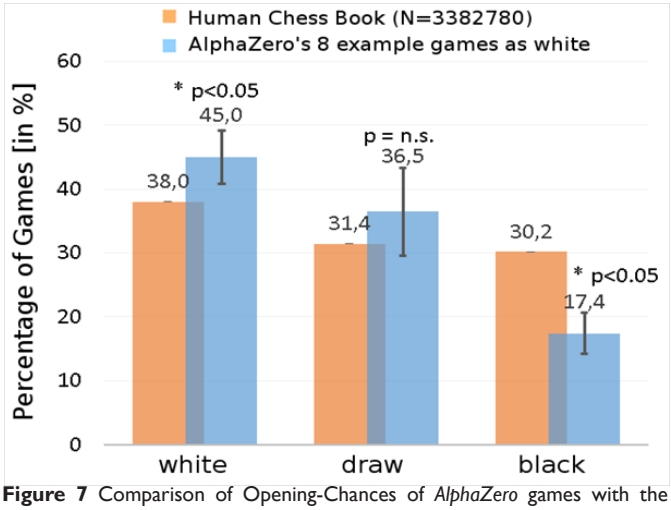

Figure 7 Comparison of Opening-Chances of *AlphaZero* games with the database of 3M human book games.

Conclusion

Indications for the hypothesis that all sciences of today are systematically biased could be found. Global biases an blockage of the majority of postdocs predominated every scientific progress of our time in all fields that are led by faculties: physics, biology, chemistry, philosophy, mathematics, medicine, all social sciences, psychology, law, economics and business studies. Bias is a major concern for all progress, the real truth, common sense and thus the future of all sciences.[4,5,8,10]

The re-analysis of probably one of the most outstanding breakthrough publications in Artificial Intelligence, in the recent scientific literature since some time, reveals some inevitable concerns about the reproducibility and comparability of the still very promising AI performance. Accessibility of a setting to independently reproduce the engine match and real tournament settings were not fully given, only 10 selected games of 100.[1] Re-evaluation screening of every move has revealed atypical moves of *Stockfish8* in almost every example match of *AlphaZero* and *Stockfish8*. Hence, this analysis must conclude that a final proof was not undoubtedly given that *AlphaZero* could or can beat the best chess engines in the world and it clearly requires several further proofs and more details and access. The best chess players in the world also need to prove this more than once in an independent tournament setting, and not only at home experimental conditions without a referee and asymmetric settings information.

Bias arises in uncontrolled settings, reproducibility issues arise systemically but this group is not to be blamed, as most publications do not even show the ambition to enable reproducibility in most of all sciences.[4,5,8,10] Still, reproducibility will secretly remain, and inescapably, an absolute requirement of all science that is not met no more worldwide. Who could still blame an individual working group only based on indications if everyone has gone wrong everywhere already? Hence, this conclusion proposes an overhaul of all sciences: (1) to stop all systematic biases, (2) to better reproducibility, and (3) to provide sustainable and independent career paths to all scientists Postdocs and PhDs without extreme pressures, and prevailing conflicts of interest are lurking around every corner.

A boost of Artificial Intelligence could be achieved if multiple roots are simulated for quiescence searches or any other searches or calculation. Two adjacent eyes see better than one and make a 2D landscape 3D. Similarly, two adjacent roots are better than one and can make the complexity landscapes appear more unbiased and dimensional for the permanent brain to work with. They not only give double information but also different starting positions that are adjacent and slightly shifted, by only very few past and future plys in chess. AI must leave the 2D PCB board to become more three dimensional and multiple roots might more cores for probably very many IT AI applications.

In opposite to Stockfish, *AlphaZero* is closer to a better multiple root solution, which might have yielded an AI advantage, as *AlphaZero* always simulates "multiple games" and not only variations, as compared to Stockfish, for root visit counts, as MTCS assesses series of simulated games that traverses a tree from root s_{root} to leaf.[1]

Hence, roots might be better assessed by *AlphaZero* and even with less of a bias due to the randomness of play out. This work proposes that roots could be simply simulated before the next root state is reached (Figure 4), and the calculation results of the previous root calculations should also be used. Even if the use of several future roots or predictive roots can be also suboptimal or even false, it sometimes can still yield a higher stability of the search on the zero-ply root due to permanent brain functions, which in a next step could be further advanced, in order to not miss out on a very important variation due to time wasting on what would be maybe called a "false in-depth focus" of an AI exercise.

The performance boost of AI via multiple roots could become a standard for "combinatorial solution finding in complexity" – but who knows if this will be an "over-claim", and what will be the best AI engine? Only the future of AI that is already approaching us.- Supplementary material is available using the following hyperlink.

Acknowledgements

I would like to acknowledge the University of Truth and Common Sense that is to be seen as a School of Life and Independence.

Conflict of interest

The author declares that there is no additional conflict of interest to the conflicts of interest found in all sciences.

References

1. David Silver, Thomas Hubert, Julian Schrittwieser, et al. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv vol*. 2017;1712(01815):1–19.
2. T Romstad, M Costabla, J Kiiski. Stockfish: A Strong Open Source Chess Engine. 2017.
3. CCRL. CCRL 40/40 - Chess Engine Comparison. 2017.
4. Hannah R Rothstein, Alexander J Sutton, Michael Borenstein. Publication Bias in Meta-Analysis – Prevention, Assessment and Adjustments. *Wiley*. 2005;376.
5. John PA Ioannidis. Why most published research findings are false. *PLoS Med*. 2005;2(8):124.
6. Michael J Mahoney. Publication prejudices: An experimental study of confirmatory bias in the peer review system. *Cognitive Therapy and Research*. 1977;1(2):161–175.

7. 365chess. Online Chess Games Database. 2017.
8. Christopher J Pannucci, Edwin G Wilkins. Identifying and Avoiding Bias in Research. *Plast Reconst Surg*. 2010;126(2):619–625.
9. Roman Anton. Falseness in the miRNA-Field as an Indicator of Strategic Bias in the Research System via Peer-Review and Publishing Eligibility. *Journal of Investigative Genomics*. 2017;4(3):1–10.
10. M Baker. 1,500 scientists lift the lid on reproducibility. *Nature*. 2016;533(7604):452–454.

Citation: Anton R. Reevaluation of artificial intelligence engine alpha zero, a self-learning algorithm, reveals lack of proof of best engine, and an advancement of artificial intelligence via multiple roots. *Open Acc J Math Theor Phy.* 2018;1(1):22–31. DOI: 10.15406/oajmtp.2018.01.00005