

Y1V11 Chess Feats Analysis

By TheHuntsman1001

Special Thanks to: Gabybel76 and DAX PMFT.

In Y1V11, during the Event Selection Exam, Ayanokouji performs a series of active calculations using his Cognitive Ability. These are now known generally as the “chess feats” and they're currently one of his most misunderstood feats in all of COTE. Here it will be explained what the Chess Feats are, their validity, and how they should be scaled based on the detailed information provided in the LN and common knowledge of the game of chess and chess computers.

During the exam, Ayanokouji plays a game of chess against Sakayanagi. The game was played indirectly through headset communication of one's moves due to the rules of ESE. As such, the ones physically doing the moves and listening to the commands of their leaders through headsets during the 2nd half of the game, were Horikita and Hashimoto. During this match Tsukishiro was also intervening by changing some of Ayanokouji’s moves to ensure he would lose the match. There are two main lines of analysis for this overall feat.

———————————————————————————————

The first string of analysis centers around how far into the future Ayanokouji calculated the moves Sakayanagi would make and thus what moves he should make. At two different times during the match he calculated many moves ahead, and the number of moves he calculated was estimated using the amount of time each of the players took to make their moves and the allotted time of their match.

The second string of analysis for this feat involves Tsukishiro’s use of multiple personal computers running a chess engine tracking the match between Ayanokouji and Sakayanagi. After the match he informed Ayanokouji that at least one of his moves had been better than even the best possible move his chess engine could have calculated. This could only be possible if he either calculated a greater number of board positions than the chess engine and found that his move led to better outcomes more often than the chess engine could calculate. Or it meant that he could analyze a relative number of board positions compared to the engine but could more efficiently analyze and evaluate the position and determine the best move.

Firstly to address some misconceptions with the chess feat:
This does not scale his FSIQ based on the premise that being good at chess or being a master of chess has anything to do with intelligence. While several notable GMs and even IMs are very intelligent people there isn’t proof that chess directly affects this. Furthermore, not all high level players would meet this consideration so linking chess to intelligence is NOT the point of this feat. What this feat does

examine is the calculations Ayanokouji makes during the match and visualization necessary to accomplish that. These are used to scale several components of his FSIQ since they cover the requisite subtests.

The primary criticism most people have with scaling is not knowing the exact moves Ayanokouji and Sakayanagi made during the match. However, knowing that is NOT necessary for analyzing this portion of the feat. Since first off it's likely nobody reading this doc or analyzing this feat is advanced enough in the game of chess for that information to mean anything in how this is scaled. What I mean by this is it doesn't matter if Ayanokouji's best move was Nf3, Bc6, or Qd2. That information wouldn't change anything about the feat itself. Also since the COTE author isn't a chess Master, creating a board position worthy of that isn't his specialty but that doesn't detract from the feat itself. Knowing the exact moves Ayanokouji and Sakayanagi made would only matter if you analyzed their chess match, not Ayanokouji's feat within it. Therefore, based on proven-ability scaling, everything contained in the chess feat is valid.

The other main criticism of the Chess feat is not knowing the exact number of moves the two made during the game, although this is more of a misconception. While the exact number was never stated we do know the length of time the match took and how quickly each of them were taking to make moves during the different stages of the game. With that information using simple math we can estimate the number of moves made throughout the match. While that information will be semi-important for analyzing this portion of the Chess feat it is not vital.

Before looking at the main analysis of Ayanokouji's moves during the match we can also address some misconceptions about his chess skills in general.

Feat 1: Calculations of how many moves in advance Ayanokouji predicted and how long he took to make the calculations:

> Disclaimer: *Due to the author's writing, calculus is used to take median values as their base during the chess match. The author, Kinugasa Shougo, managed to structure this chess match in a way that doesn't make it difficult to interpret. The key is to connect the dots and read between the lines while staying true to the original text.*
>
> The author did cut the match in two halves of approximately 17/13 min each, here are the timeline narratives to support this:

> Eventually, during the second half of the match, Sakayanagi's hand stopped.
>
> Under more normal circumstances, I already knew what move Sakayanagi would make next.
>
> But— she mysteriously became lost in thought instead.

> Even after the match went past the halfway point, instead of opening the gap between us and spreading my tiny lead, it took the full extent of my ability to prevent her from closing in on me.

Both scans note that a little past the halfway mark of Ayanokouji and Sakayanagi's 30 min time clock there was a shift in the match. 17 min for the first half and 13 min for the second half were estimated based on that fact.

Now for the first calculation, this informs us that each moves took approximately 10-20 seconds, this value is used to calculate this feat conservatively:

> Each move we made took anywhere from 10 to 20 seconds. And each time I pressed the enter key to send our instructions, the countdown timer would stop decreasing.

To put simply, Ayanokouji made 'x' number of moves made in a time span of 17 min → 'x' = 60/15 = 4 moves per minute x 17 minutes = 34 moves each | 'x' = 34, so Ayanokouji and Sakayanagi each made about 34 moves in this span of 17 min which total to 68 moves. It is directly stated that they both took ~15s for each move. As seen above, Ayanokouji affirmed that at the end of the 17 minute span, he already KNEW what move Arisu would make under normal circumstances (meaning without Tsukishiro's intervention), which means that he had predicted her 34 moves up to that point, and already had a winning strategy, proving that he predicted more than these 68 moves in total, here is the proof:

completely give up on the path to the end of the game that I had initially envisioned.

And then, I had to build a new path. One that would lead me to victory.

Just before the two-minute mark, I typed something into my keyboard and sent my instruction to Horikita.

In the end though, Ayanokouji needed to recalculate his plan since Tsukishiro intervened for the first time by changing one of his moves.

Now how long did it take for Ayanokouji to calculate this “winning line”? We don’t have an exact time but based on what we know about the stages of the match we can make a reasonable estimation.

Since this dialogue starts at the beginning of Ayanokouji’s 30 min time limit, and Ayanokouji was both managing his time limit for the match while also taking enough time to calculate a winning line of moves, taking more than a few minutes wouldn’t make any sense. Also given that there wasn’t a point during those moves where Ayanokouji wasn’t making his moves at regular 10-20 second increments he must have finished his calculations early on in his time limit and then continued each of his moves after regular intervals double checking everything was going according to plan.

Based on this information we can reason that Ayanokouji must have spent less than about 3-5 minutes making these predictions. If we use 5 minutes (which is equal to 300 seconds) we can calculate approximately how much time he spent calculating each move. We confirmed that Ayanokouji had calculated more than 68 moves ahead, but we can’t determine exactly how many. For the purpose of this calculation we will assume he only calculated 68 moves because Tsukishiro’s intervention forced him to change his strategy at that point.

5 min=300s | 300/68~4.41s per prediction

This means that he could do the following in these 5 mins:

- Make 1 move calculation every 4.41s
- Analyze Arisu’s aggressive playstyle without seeing her, (there wasn't any visual contact between them), as they were sending instructions through the monitors.

- Processing in his chessboard in his mind along with his premoves/predictions.
- Maintaining the process for a whole span of 300s

That covers the first part of their chess match spanning about 17 min out of the allotted 30 min for commander participation. Now moving on to the second part covering the about final 13 min.

Ayanokouji stopped playing after Arisu responded strongly to his last move, which marked the 1st period's end.

> One, two minutes passed. Time flew by without stopping.
>
> All of the time I had saved up was slowly ticking itself away.

> [You've only got around five minutes left…!]
>
> Of course, I was well aware of how much time I had.

> No matter how much time I took reading into what move to make, it would still be worth it.

> There were less than three minutes left. I was forced to wipe my slate clean, to completely give up on the path to the end of the game that I had initially envisioned.
>
> And then, I had to build a new path. One that would lead me to victory.
>
> Just before the two-minute mark, I typed something into my keyboard and sent my instruction to Horikita.

Here Ayanokouji took time to repose his mind, then used about a one-minute span to calculate a new winning path, he cleaned his whole brain about his previous winning path, and took 1 min to create anew. (from his 3rd to 2nd remaining minute)

After this, Ayanokouji took no more than 2 seconds to confront arisu's moves that respectively took 30 seconds, 60 seconds, and longer toward the end of the match.

> Her first response took 30 seconds, as did the next. And the move after that took her a full minute.
>
> I, on the other hand, was responding to her moves within one or two seconds each.
>
> The two of us were now walking down a path that would lead to my victory.
>
> The endgame was approaching. The outcome would be decided soon.
>
> With my next move, checkmate would be all but certain.
>
> She still had moves she could make to escape it, but those were few in number.
>
> If she messed up, she would lose her last way out.

If we assume that there was exactly 2 minutes left on his timer and he moved every 2 seconds we can again use math to estimate the number of moves he calculated here.

x=120/2, x=60 moves. Ayanokouji and Arisu would each make 60 moves totaling 120.

Ayanokouji eventually ran out of time before he could win the game since Tsukishiro changed more of his moves during this second phase and delayed the time it took those instructions to reach Horikita. We know that he mentioned that one move specifically took about 30 seconds to reach Horikita. Since he was making each move in just a few seconds, having this happen multiple times would bring too much attention to Tsukishiro. So in that case if we remove 30 seconds from Ayanokouji's remaining time x=90/2, x=45 he may have only made 45 moves during this final part of the game. This move count makes sense as endgame often involve a series of late pawn moves which would cause the total number of moves to be higher. Either way we can be sure that even in this much shorter amount of time he made even more moves than during his beginning section However, since he ended up running out of time due to Tsukishiro's interference his second half calculation here could have been even further in advance than his original plan of

60 moves for him + 60 moves for Arisu which would be 120 total moves calculated. For the purpose of this feat we assume the conservative value of only 45 moves instead of 60, meaning a total of 90 moves.

As Ayanokouji's move was again altered by Tsukishiro, Arisu was able to win the match. Now she just had 5 moves to make in order to win; and majestically finished the game with her queen sacrifice, providing her the "victory", in these few intense seconds.

> During the match, there was an instruction I had sent to Horikita after a lengthy period of consideration.
>
> But, it took around 30 seconds for the instruction to make it from my keyboard to Horikita's headset.

> Over the course of the next two to three moves, it felt as though the match had deviated from my path.

> "It was a fun match. It really is quite regrettable that it has to end."
>
> She had less than 40 seconds. Her voice was calm, and as she spoke, I could hear the sound of her typing something into her keyboard.
>
> Instead of acknowledging her surrender, her words were instead meant to highlight the confidence she had in the ferocious move she had come up with.
>
> […Atta girl, princess!]
>
> It was a comeback from the brink of death by Hashimoto, or rather… by Sakayanagi, the one who was standing behind him.

> There were 5 moves left… and then 4… 3… Before finally…
>
> Sakayanagi forced the checkmate via queen sacrifice.

This marked the end of Ayanokouji and Sakayanagi's chess match. At 2 different points during the match Ayanokouji made calculations which would have ended in a checkmate against Arisu. The first time he calculated at least 68 moves ahead and the second time at least 90 moves ahead.

In conclusion for feat 1, here is a summary of Ayanokouji's feats during the match, as well as establishing a range from "low-ball to extreme-ball scaling ladder:

- 17 minutes time frame '1st phase of the game'
- 68 moves in advance calculated by Ayanokouji
- 1 minute time frame '2nd phase of the game' (With prep time before)
- 90 moves in advance calculated (likely 120+)

Considering that game's progression was halfway done when they began to handle the reins. And that professional players' endgames are full of temporization and gauging moves, we can assume that there were still a fair amount of pieces/pawns on board.

I consider ⅓ of the game's 1st part to be the beginning, + I already proved that Ayanokouji managed to process everything at this specific beginning ; 5 min is the time I'll set for the low-ball, while 3 min will be the mid-high ball.

- 5 min timespan (Very Low-Ball) = ~4.41s/prediction
- 4 min timespan (Low-Ball) = ~3.53s/prediction
- 3 min timespan (MID Ball) = ~2.65s/prediction
- 2 min timespan (High-Ball) = ~1.76s/prediction
- 1 min timespan (Very High-Ball) = ~0.90s/per prediction.

In order for Ayanokouji to analyze and select the best move he can make in a situation he needs to analyze the different possible board states that can occur from the board's current position and make a move that creates a more favorable position. This is accomplished by running a simulation of possible moves and choosing the best one, or using pattern recognition to identify patterns and situations from previous games played or other analyzed games. The best players use a combination of both. His perfect memory and high pattern recognition

abilities help him greatly in finding the best possible move when analyzing board positions in his head.

Scaling of the Chess Predictions:

Ayanokouji was able to visualize multiple possibilities, choose the best one and predict his opponent's move in a magnitude of 1 sec - 4 sec per move during the beginning portion of his play. He managed to keep this multi-tasking and heavy process going for a few minutes (3-5 mins) while simultaneously formulating his long-term move strategy, which is borderline inhuman, even in a fictional context.

Finally I would add that each process which led to each predicted move is considered deductive reasoning, here is the rough process:

- Ayanokouji accurately analyzes the game's status:
- What is the best move I could do ? -> what moves Arisu would play based on what I play and her current understanding of the match?
- Calculate for each move the accurate solution by anticipating arisu's next moves.
- Deduces his winning path, based on his understanding of Arisu's abilities.

While Ayanokouji's active predictions during the chess feat showcase a level of FSIQ necessary for their execution, the ability to make predictions in chess, has been shown to not have such a major correlation to FSIQ and indexes such as WMI and PSI. However, it is still a very strong example of Ayanokouji's Analysis and Prediction skills.

The reason for this is a bit vague. Making predictions about your opponents future moves is something that GMs do often even if to lesser degrees. However this doesn't necessarily correlate to FSIQ as GMs such as Hikaru Nakamura don't even possess a genius intellect. Mainly this ties back to not scaling Ayanokouji's FSIQ based on his chess skill, but instead by his actions within the feat.

His ability to make such predictions against an opponent as strong as Arisu showcases that Ayanokouji has at a minimum a very high level GM rating, these predictions are not the most impressive part of the Chess Feat WHEN it comes to FSIQ specifically. Even so, being able to accurately predict the moves of a GM level opponent makes a strong case for Ayanokouji's Analysis and Prediction skills.

GMs such as Magnus, Hikaru and others have shown an ability to predict moves in a similar way with the main difference being the length of the predictions.

Feat 2: Calculations which exceeded the calculation capacity of Tsukishiro's multiple computers:

Now to look at how we can evaluate Ayanokouji's chess feat without this specific information. Simply put, we take the information Tsukishiro provided and contextualize it.

> During our game, there had been a number of instructions I'd issued to Horikita after a lengthy period of consideration. But even after inputting everything on the keyboard and pressing enter, there was almost a thirty-second delay before the instructions were received. Up until that point, the time lag before my instructions were received had only been about ten seconds.

> "Properly speaking, the winner of that chess match should've been Ayanokōji-kun."

> "He was planning to use a different move, at the time. An even better move than the best move we could come up with. I had a significant number of personnel and machines to account for his options, but we were still forced into making an extremely difficult decision."
>
> If they'd made an overtly careless move, it would've been obvious to anyone watching that something was off. To avoid that happening, he'd been forced to come up with a complex move.

As seen here, Tsukishiro explained that Ayanokouji had made a move that was "even better than the best move they could have come up with" while having an entire team working on analysis of the game. That team also included the use of multiple computers assisting the evaluation of the game.

Knowing this we can look into what making such a move actually means.

While we don't know the exact computers Tsukishiro used or the exact chess engine he was running, but given the fact that COTE is confirmed to take place in the future we know that the technology he has access to should be at least relative to our current technology and likely more advanced. COTE takes place at least around 10 years in the future so computer technology should be much more advanced than today's but since we don't know how powerful computers will be 20 years from now we will create a calculation of just how impressive this feat is by using current technology as a low ball estimate.

For reference on how quickly computers can advance we can look at a chess related example. In 1996 IBM created a supercomputer called Deep Blue designed to play chess. It played a series of matches vs the World Champion at the time Gary Kasparov. Kasparov managed to win but less than a year later they played a rematch which Deep Blue won fairly easily. This was the last time human ability exceeded computers at chess. About 18 years ago that computer could calculate approximately 200 million board positions per second based on its hardware. When a computer plays chess there are 2 main components, the hardware and the software. The hardware determines the calculation ability of the computer, meaning how many positions it can analyze per second. The software takes those analyzed positions and evaluates them. For example if today Stockfish played Deep Blue it would win since it is much better on the software side and evaluates positions far more efficiently.

Deep Blue had 11.38 GFLOPS of processing power which was massive at the time. However, by today's standards that would be incredibly subpar. Now computing power is measured in TFLOPS with 1 TFLOP = 1000 GFLOPS. An iPhone has around 10-15 TFLOPS of computing power while an average desktop can have 60 TFLOPS or more. This means over about a 15 year span, now an average phone has about 1000-2000x the computing power of Deep Blue while a PC has more than 5000 times the computing power of Deep Blue. At this rate the computers in COTE could realistically have thousands of times the computing power of current hardware in just 10-15 years.

- I found that the supercomputer Deep Blue "was capable of evaluating 200 million positions per second"
- But Deep Blue was old, supercomputers have became much more powerful since 1997, and Deep Blue wasted time evaluating every positions.
- Therefore 200 million is only a lower bound for the answer to this question.

Since Deep Blue had a performance of 11.38 GFLOPS, and the current most powerful supercomputer (Tianhe-2) has a performance of 33.86 PFLOPS = 33 860 000 GFLOPS, we could estimate that the answer would be closer to 200 000 000 * 33 900 000 / 11.38 = 6 * 10^14 = 600 trillion positions per second. That seems like a lot, but taking into account that in average a position has around 35 legal moves, given 3 minutes such a computer would only be able to calculate every single positions up to a depth of 11 plies. Since Deep Blue wasted time evaluating every single of those 200 million positions per second, I believe Deep Blue should have been able to calculate approximately a billion positions per second if it didn't have to evaluate them.

The misconception that many people believe regarding this feat being that Ayanokouji outperformed a supercomputer which could play itself 20 million times a second is laughable. Not only is this number never mentioned in the Light Novels, even computer's 20 years ago could compute 10 times that. Given that COTE takes place in the future such a number is extremely low. Especially since a chess engine running on an iPhone could calculate anywhere from 200 billion to 400 billion positions per second. This certainly disproves the 20 million moves per second fallacy.

If Tsukishiro was running a chess engine on average computer hardware as of today's standards (2023) with about 60 TFLOPS of computing power, his machines could calculate about 1.014 Trillion positions per second each at full capacity. For simplicity we will round to 1 Trillion.

So now that we have a better understanding of how much computing power modern computers have and that the ones Tsukishiro used could even be more advanced, the fact that Ayanokouji's move was better than the best move they could come up with is becoming more impressive. But how do we know that Ayanokouji out calculated those computers when finding his move? To understand this we need to quickly address how an engine determines the best move.

Chess engines can only analyze a certain number board positions given the hardware it runs on and the more powerful that is the more it can analyze. A common way that this is addressed is websites such as Chess.com or Lichess.com host a chess engine connected to an online server. This allows an engine to access

an entire system of computers hosted elsewhere. This is how many people can utilize an online engine like that at once since each user is only able to access a fraction of that total computing power. However, running an engine directly on a computer is often more efficient and more powerful than the sliver of the server you can access. However the actual evaluation is done by the software/engine itself. A chess engine starts with a process called alpha-beta pruning to quickly discard objectively bad moves from the start.

In further detail, the process of alpha-beta pruning is used to reduce the total number of calculations they need to do in order to find the best move. The computers create move trees from each possible move that branches out for each possible move from the starting move. If it evaluates a tree as not having a chance of producing a better position than the best one so far it prunes the tree. This helps engines not have to calculate for hours to find the best move. This is how an engine discards objectively bad moves without needing to spend resources to evaluate them. This process is taken into account when looking at how many move combinations per second engines can calculate. That means if an engine calculates 1 trillion positions per second that means 1 trillion “good” positions, not including the bad ones which were discarded. This is important because Ayanokouji and the engine would eliminate the same kinds of moves.

Since I have explained that the engine has a limited amount of calculation power to use the smarter the engine the better/more efficiently it can work through this process. The more moves it can discard here the more in depth on the good moves it can analyze. For example, If a computer has 50 legal moves they can make we can assume 25-30 of those moves are objectively bad. Exactly how bad would depend on each one individually but the more of these the engine can eliminate means it can look at the other 20-25 moves in more depth. If it had to analyze 50 moves it may only be able to look 10 moves ahead but with only 20 moves it can look 15-20 moves ahead. Since in both cases looking ahead would determine the bad moves and the engine would not make them. This process saves the engine time and processing power to be used on better moves. However, by eliminating objectively bad moves that doesn’t mean the engine calculates fewer moves. It just means it looks deeper at better ones.

Additionally, it is possible that Tsukishiro was using something similar since connecting a chess computer to a server to boost processing power is a common practice. However, while making such an assumption would mean Ayanokouji

would have needed to make a MUCH larger calculation there is no evidence Tsukishiro used a server so it won't be included in the calculations for the feat.

The next stage involves taking the moves which are not objectively bad and rating them. This rating involves looking at how the pieces are placed, what pieces are left on the board, and how those pieces can interact with others. Then the engine looks ahead as far as it can based on its calculation power and will find the path where the farthest move it can calculate has the best rating. A computer will determine that a move which gives a short term loss for long term benefit is better than short term gain. Meaning that a chess engine will always prioritize long-term benefit when analyzing moves because those moves give the best chance to ultimately win the game.

So from this we know how the chess engine determines what it believes to be the best move each time and we can see how Ayanokouji would be able to find a move that was better than its best move. The engine would evaluate the position and find its "best move" it thinks Ayanokouji should make, then Ayanokouji makes his move which would be different from what the engine decided, meaning the engine thought its move was better than the move Ayanokouji came up with. Afterward once the game progressed it would then realize that Ayanokouji's move was better than the best move it could come up with at that point. This would be the only way Ayanokouji's move could be better than the engine's best move and ensure that Tsukishiro would know that. If the engine didn't later realize his move was better Tsukishiro wouldn't know and wouldn't have been able to tell Ayanokouji about it later.

So in order for Ayanokouji to make a move better than the engine, he either needed to calculate a higher number of total board positions than the combined computers were able to, meaning he could analyze the game at a higher depth than the engine. Or he could have evaluated the position better and more efficiently than the engine could, leading to finding a better move than the engine at equal depth. However, Ayanokouji finding his move by evaluating better than the computers would be impossible if he didnt' at least match the calculation of the computer (although he wouldn't need to exceed it) because as explained earlier the engine will always prioritize long term benefits in its moves. If Ayanokouji found a "better" move but without calculating a similar number of positions and therefore calculating a lower depth than the engine his move may seem better in the short-term but wouldn't be better overall. For example a move that appears to win a piece from your opponent looks good at first, but if that move was actually a sacrifice from the opponent to win something better a few moves later that move

isn't actually the "best move". That same principle applies here, without matching the depth of the computers, even if Ayanokouji exceeded their evaluation level he would still need to match the calculation. That being said, this method of making his move is less likely for several reasons. Firstly, computers are designed to process data with known variables and they can do this far more efficiently than humans, if Ayanokouji is able to do this it would put his thinking, reasoning, and analysis skills far into the realm of superhuman. Second, Chess is a game where each player has perfect information which makes computers ideal for analyzing it. This means that each player knows every possible move they and their opponent can make at any given time. Because the engine is rating each possible move based on this complete set of information the engine's ability to rate each move is perfect based on the given information. This is a main reason why if 2 chess engines play each other the game will end in a draw. Since each engine can evaluate the game from their own point of view and the opponents it can determine the best move every time. The only way to break the tie is to calculate more positions than the opponent. This is the difference between victory and defeat for chess engines. Third, because Tsukishiro knew that Ayanokouji had made a move better than the engine he must have gotten that information from his computers. Meaning the computers must have later calculated that Ayanokouji's move was better than the one they came up with. This further supports Ayanokouji calculating more positions than the engine since after that move as the engine continued to calculate in the game it would begin to evaluate Ayanokouji's position better after a few moves due to not correctly identifying the best move several turns ago.

Finding his move through sheer calculation is the only one that makes sense and optimally at that, it would explain how Tsukishiro knew about how good his move was and that the engine would have figured out that Ayanokouji's move was better in hindsight.

So now that we see that Ayanokouji making a larger calculation than the engine is how he would have found his move we can also see that without making the calculation finding that move would be impossible. As mentioned earlier, the computers are discarding objectively bad moves but analyzing the rest to find the "best" move from a given position. This means that the engine would have considered the move Ayanokouji made as a possibility although ultimately making a different one. However, why did the engine choose a different move? Those reasons are the same 2 reasons from before. Either Ayanokouji evaluated the move as better based on the situation which as was already explained not a possible way his move could be "better than the engine" or he calculated more moves and realized that in the longer-term this move is better than the move the engine

found. This not only proves Ayanokouji needed to calculate more positions than the computers, but that he calculated at a higher depth.

So how many board positions did Ayanokouji have to calculate to find his move? We can estimate based on the number of positions the computers Tsukishiro employed were able to calculate while knowing that Ayanokouji would need to exceed that calculation to find his move.

		Number of Computers		
		2	3	4
TFLOPS	Positions Per second	2.00E+12	3.00E+12	4.00E+12
60	1 Trillion	2,000,000,000,000	3,000,000,000,000	4,000,000,000,000

Here we can see the total size of the calculation given the information we have. While the calculation size changes based on how many computers Tsukishiro was using, if we take a value of 60 TFLOPS for an average computer the engine would have been calculating somewhere between 2 and 4 trillion positions per second.

However, this assumes that the computer makes its calculations in exactly 1 second which we know isn't the case, since Ayanokouji mentioned that this move specifically took much longer than normal to send through to Horikita. We also need to consider the context of when his move was made.

Tsukishiro only specifically mentioned one of Ayanokouji's moves being better than the engine so we won't assume it would be the case for his other moves, but we know the move Tsukishiro had to change to ensure Sakayanagi won was in the final minute of their game. During that time Ayanokouji had formulated his second string of moves after Tsukishiro altered his move during the earlier part of the game. So this move from Ayanokouji was at the end of his winning line in the final 2 minutes of the game. Here Ayanokouji was making his moves every 1-2 seconds until he ran out of time, and at the beginning of that period he had calculated this line of moves in advance.

I had less than three minutes left. The pattern I'd envisioned for the game's end was wiped clean, returning me to a blank slate. I started building a new path toward victory. Just before the two-minute mark…

My fingers tapped the keyboard, and I gave instructions to Horikita, who sprang into action like she had been waiting for me. The piece flew about the board with intense energy. Hashimoto looked flustered once again.

Her first response took 30 seconds, as did the next. And the move after that took her a full minute.

I, on the other hand, was responding to her moves within one or two seconds each.

The two of us were now walking down a path that would lead to my victory.

The endgame was approaching. The outcome would be decided soon.

With my next move, checkmate would be all but certain.

She still had moves she could make to escape it, but those were few in number.

If she messed up, she would lose her last way out.

Here we see Ayanokouji mention that in the final 2 minutes of the match he "built a new path toward victory" meaning he calculated a new series of moves and was making each of those moves in 1 or 2 seconds each time. Meaning that Ayanokouji's series of moves could be up to 60 moves long, since he would need to be able to make all of the moves before his time ran out and he was making moves at least every 2 seconds. Even if we assume a far more conservative estimate of a 30 move plan it changes the calculation a lot. Since all his moves including his move which was better than the engine were all calculated in advance it actually means that the calculation Ayanokouji would need to make had to include all the moves leading up to his best move. Essentially Ayanokouji would have needed to calculate

all his moves up front while the engine would be able to recalculate after each move. Furthermore we know the engine wasn't spending 1 sec on each move and specifically on his best move he mentioned it took about 30 seconds before it reached Horikita. That means the engine calculated for at least 30 seconds on his "best move" meaning it had time to process even more board states.

> I came up with my move. I quickly typed my instructions on the keyboard and hit Enter. The counter stopped. Horikita waited for my message, like she was praying. About thirty seconds after I had sent my instructions, Horikita's eyes widened. The long-awaited signal had apparently reached her via her headset.

Taking those variables into account Ayanokouji's calculation is far more impressive and looks like the following.

> "He was planning to use a different move, at the time. An even better move than the best move we could come up with. I had a significant number of personnel and machines to account for his options, but we were still forced into making an extremely difficult decision."
>
> If they'd made an overtly careless move, it would've been obvious to anyone watching that something was off. To avoid that happening, he'd been forced to come up with a complex move.

If we assume that Tsukishiro only had 3 computers at the same 60 TFLOPS of computational power the calculation would be as follows:

3 Trillion Positions Per Second	x	30 Seconds of Calculation	=	90 Trillion Total Board States

However, we can assume as a low ball that Tsukishiro had only 2 computers at 60 TFLOPS of computational power to provide a conservative estimate for the calculation. That means his computers can calculate a combined 2 Trillion board positions per second. If we don't take Ayanokouji's predicted moves into account for this calculation and just calculate the number of positions the engine would calculate during his last move we find that Ayanokouji's calculations would need to exceed 60 Trillion total board states during this move.

2 Trillion Positions Per Second	x	30 Seconds of Calculation	=	60 Trillion Total Board States

Here we take into account that the engine calculated for 30 seconds on the final move Tsukishiro changed and we see the additional calculation required so that Ayanokouji's series of moves would end with the move better than the engine. If we consider that this move along with all his other moves during the last 2 minutes of the game were calculated in advance at the 2 minute mark the calculation becomes even larger.

By calculating this move in advance it would mean that Ayanokouji's calculations for this entire series of moves which ended with his move which was "better than the engine" would have to include the engines calculations during the preceding moves too. This doesn't mean that every one of his moves was better than the engine since that would drastically change how this is calculated, but by calculating how many total positions the engine calculated during this line ending in his best move we see the required minimum calculation Ayanokouji needed to make in order to accomplish this.

Given that this series of moves could be up to 60 moves long based on the timeframe we will again use conservative numbers to calculate. Here we calculate each move based again on only 2 computers calculating at 2 trillion positions per second. Then since Ayanokouji made each move in 1-2 seconds each computer could calculate for 2 seconds on each move with the expectation of the final move where it calculated for the stated 30 seconds.

10 Moves	20 Moves	30 Moves
70,000,000,000,000	110,000,000,000,000	150,000,000,000,000

Here we are looking at the length of moves Ayanokouji calculated in advance leading up to his best move at the end. This is important because as explained earlier Ayanokouji calculated all his moves from this part of the match in advance. Depending on how many moves in advance he calculated his "best move" the required size of the calculation also increases.

Again it is important that these numbers are still low end estimates since here we assume Tsukishiro only used 2 computers when he could have easily been using more, and we assume that the computers could only calculate at a level equal to an average computer today. That wouldn't account for COTE taking place in the future.

So from this we can see that at the very least Ayanokouji needed to make a calculation of 60-70 Trillion board states and that number could be much higher if Tsukishiro used more than 2 computers or those computers could calculate above the level of average computers today. Since COTE takes place in the future and these computers were funded by either the White Room or the Japanese government they could be much better than average.

Summary of the Engine Calculation Feat

In conclusion Ayanokouji was able to at least once make a calculation which found him a move which was better than the best move Tsukishiro's team and their computers could find. Depending on how powerful those computers are the exact size of the calculation differs but using an average computer to estimate a low ball for Tsukishiro's computers. From that, we can see the size of Ayanokouji's calculation amounts to about 1 Trillion moves per second per computer.

With only 2 computers: 2 Trillion Moves per second

With 3 computers: 3 Trillion Moves per second

Then given the fact that the computers had at least 30 seconds to calculate on the move which Tsukishiro changed, the total size of Ayanokouji's calculation would be at a minimum 60-90 Trillion board states.

Additionally, depending on the length of the series of moves Ayanokouji made leading up to that move the total size of the calculation could be far higher still. Using the conservative estimate of even 30 moves instead of a likely 90+ we see his total calculation would exceed 150 Trillion Board states.

WR Chess Statements

> As a child, I learned how to play chess in the White Room.
>
> I had played against a great number of so-called professional instructors, but she was better than all of them.

In the infamous environment known as the White Room, Ayanokoji received his training on his chess ability through the hired professionals the WR institution searched for in large numbers for the instruction of every child to a proper degree. Ayanokoji states that he had played against a multitude of professionals as a child, and that in comparison, Sakayanagi was better than each one of them.
Despite having not played Chess anytime within the last year and a half since leaving the White Room, Ayanokoji was able to match and prove superior to Arisu, this implies that he is at a level incomparable to the Professionals he used to face as a child, as even if he only maintained himself at his WR skill level, he was still able to beat someone superior to all of his instructors.

The minimum qualification for a professional ranking at chess would be 2400-2600 FIDE rating, which is considered International Master to Grandmaster ranking. In order to be a professional you have to be “engaged in a specified activity as one's main paid occupation rather than as a pastime” and only high level players could meet such a qualification.

This means that based on Ayanokouji’s evaluation, Arisu is at least a high ranking Grandmaster level. In order for her to be better than all the professionals he played as a child she would need to be skilled enough to easily rank above IMs and some GMs. This would put her rating likely above 2700.

While this can give some context to the level of skill both players have, neither Ayanokouji or Arisu’s estimated FIDE rating is a good enough benchmark to scale their intelligence.

FSIQ Scaling of the Chess Board Calculations:

Ayanokouji’s chess feats are examples of his ridiculous FSIQ, and specifically showcases his inhuman PSI, WMI, PRI, and CPI. To explain this first we will give a brief explanation of these FSIQ categories and how the chess feats showcase them.

PSI – Processing Speed Index generally suggests a greater ability to process information quickly and efficiently. This may be associated with quicker decision-making, better performance on tasks that require visual-motor coordination, and enhanced performance in situations that demand rapid processing of information. Here Ayanokouji's ability to rapidly make calculations of massive size demonstrate his inhuman PSI. The sheer scale of his calculations (Specifically in Feat 2) makes the fact that he did it so quickly impressive. Given that Ayanokouji's calculation abilities surpassed the chess engine and he was able to calculate board positions faster than the trillions of positions per second the computers could, his PSI is monstrous.

WMI - Working memory is a system in the human brain that temporarily holds and manipulates information needed to perform cognitive tasks. It plays a crucial role in various cognitive processes, including problem-solving, reasoning, decision-making, and language comprehension. WMI is measured by subtests which call for the ability to recall information from short term memory in a specific order. The order is important as the test will have the participant memorize a sequence and repeat it either forward or backward. As Ayanokouji was creating his winning lines during the different phases of the game, he needed to store the visual information in his working memory. Since Ayanokouji has a perfect memory his WMI may not have an upper limit per the Long-Term WMI analysis, and this is a good example of that. This means he needs to manage between 60 trillion and 90 trillion individual pieces of visual data and keep that data organized since all those individual positions are ordered specifically as one move after another. Again, given the sheer scale of this feat Ayanokouji maintaining this in his Working Memory is very impressive.

Long-Term WMI Analysis - https://docs.google.com/document/d/1-p3xPm0FNlB8Y6j1z3dy21mwGnem4yq_bMHkQ5YMgrY/edit?usp=sharing

PRI – Perceptual Reasoning Index is intended to assess an individual's non-verbal reasoning and problem-solving abilities related to visual and spatial information. The Perceptual Reasoning Index typically includes subtests that evaluate skills such as visual processing, pattern recognition, and spatial reasoning. This is the category that contains Visual Spatial Index (VSI). This is important because Ayanokouji needed to use visual data to process this quantity of information. This

process can't just be using verbal data such as the verbal chess notation (i.e., d4: Qx4) which comprises letters and digits, because according to research on working memory, verbal memory spans are limited by a few chunks ranging from 4-7 different chunks in parallel, but the research found no fixed limit on visual data.

Working memory is widely acknowledged as having limited capacity. An early quantification of the capacity limit associated with short-term memory was the "magical number seven" suggested by Miller in 1956.[20] Miller claimed that the information-processing capacity of young adults is around seven elements, referred to as "chunks", regardless of whether the elements are digits, letters, words, or other units. Later research revealed this number depends on the category of chunks used (e.g., span may be around seven for digits, six for letters, and five for words), and even on features of the chunks within a category. For instance, attention span is lower for longer words than short words. In general, memory span for verbal contents (digits, letters, words, etc.)

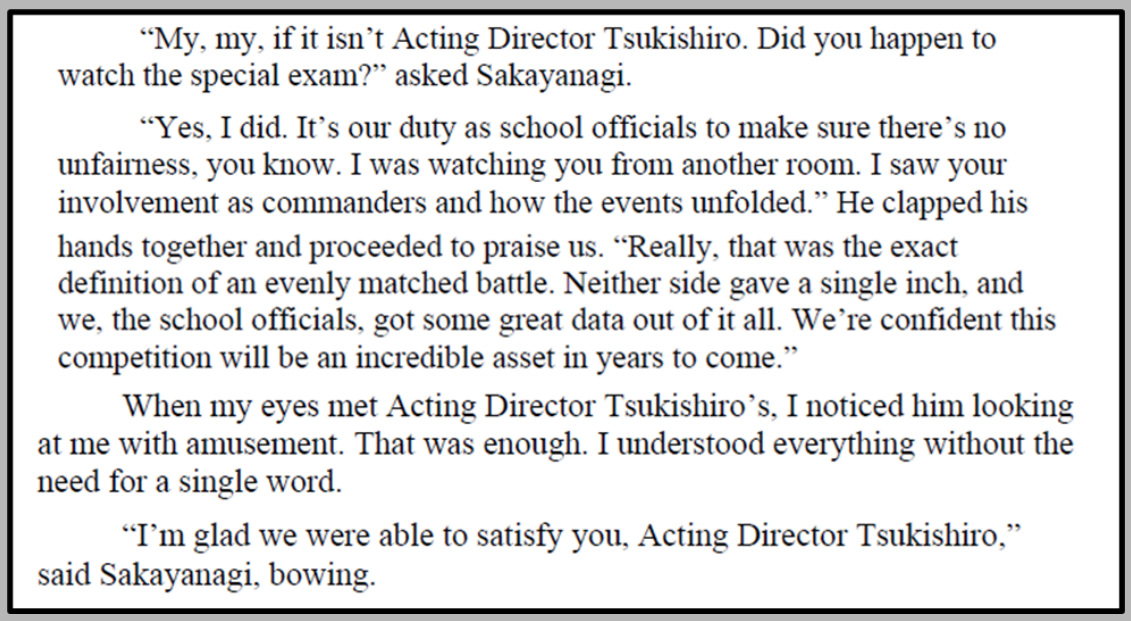
In the visual domain, some investigations report no fixed capacity limit with respect to the total number of items that can be held in working memory. Instead, the results argue for a limited resource that can be flexibly shared between items retained in memory (see below in Resource theories), with some items in the focus of attention being allocated more resource and recalled with greater precision.[24][25][26][27]

Proofs and Misconceptions Clearing

This final section is to further explain how during the Y1V11 chess match the only way Ayanokouji could have made a move that Tsukishiro described as being “better than the best move we could have made” is only possible by either directly out calculating the chess engine, or by more effectively analyzing board positions. Due to the circumstances of this game these are the only possible ways he could have accomplished this. Here are some potential ways people may believe the Chess Feat can be debunked but these are the reasons such debunks are invalid.

1. Making a move that confuses the engine.

Making a move designed to confuse a chess engine is certainly possible and isn’t that uncommon, however that is impossible under these circumstances. Ayanokouji didn’t know that Tsukishiro was using a chess engine during their match and therefore he couldn’t have made a move with that in mind. As seen here Ayanokouji and Sakayanagi meet Tsukishiro after their match ended and during their encounter Ayanokouji said that he understood what he had done. This was the moment that he realized that Tsukishiro had used an engine to interfere with his moves and that was how he sabotaged him. Ayanokouji had obviously already realized that he was being sabotaged, but this was when he figured out exactly how. Because this was after the match he couldn’t have known about the engine beforehand and couldn’t have used that knowledge to make a move specifically to confuse it.



“My, my, if it isn’t Acting Director Tsukishiro. Did you happen to watch the special exam?” asked Sakayanagi.

“Yes, I did. It’s our duty as school officials to make sure there’s no unfairness, you know. I was watching you from another room. I saw your involvement as commanders and how the events unfolded.” He clapped his hands together and proceeded to praise us. “Really, that was the exact definition of an evenly matched battle. Neither side gave a single inch, and we, the school officials, got some great data out of it all. We’re confident this competition will be an incredible asset in years to come.”

When my eyes met Acting Director Tsukishiro’s, I noticed him looking at me with amusement. That was enough. I understood everything without the need for a single word.

“I’m glad we were able to satisfy you, Acting Director Tsukishiro,” said Sakayanagi, bowing.

2. Making a move the engine doesn't see is common.

Making a move better than an engine is a misconception based on the fact that some people think a move classified as a "brilliant move" by a chess engine was actually better than its best calculated move. This is simply not the case. A brilliant move is the highest rated move a chess engine will classify a player move when analyzing it. It refers to a move it considers either extraordinarily clever, hard to find, or one that involves a calculated piece sacrifice. There is no correlation between brilliant moves and making a better move than an engine.

> *Brilliant (!!) moves and Great Moves are always the best or nearly best move in the position, but are also special in some way.* ***We replaced the old Brilliant algorithm with a simpler definition: a Brilliant move is when you find a good piece sacrifice.*** *There are some other conditions, like you should not be in a bad position after a Brilliant move and you should not be completely winning even if you had not found the move. Also, we are more generous in defining a piece sacrifice for newer players, compared with those who are higher rated.*

While the Chess.com definition is a fair application of the term "brilliant move", I feel that many moves classified as "brilliant" on Chess.com don't live up to the expectation. I think chess.com is missing the "hard-to-see" and "unexpected" parts of the widely accepted "brilliant" term. Many of the brilliant moves I've seen here would only be considered great move (!) at most in any other context.

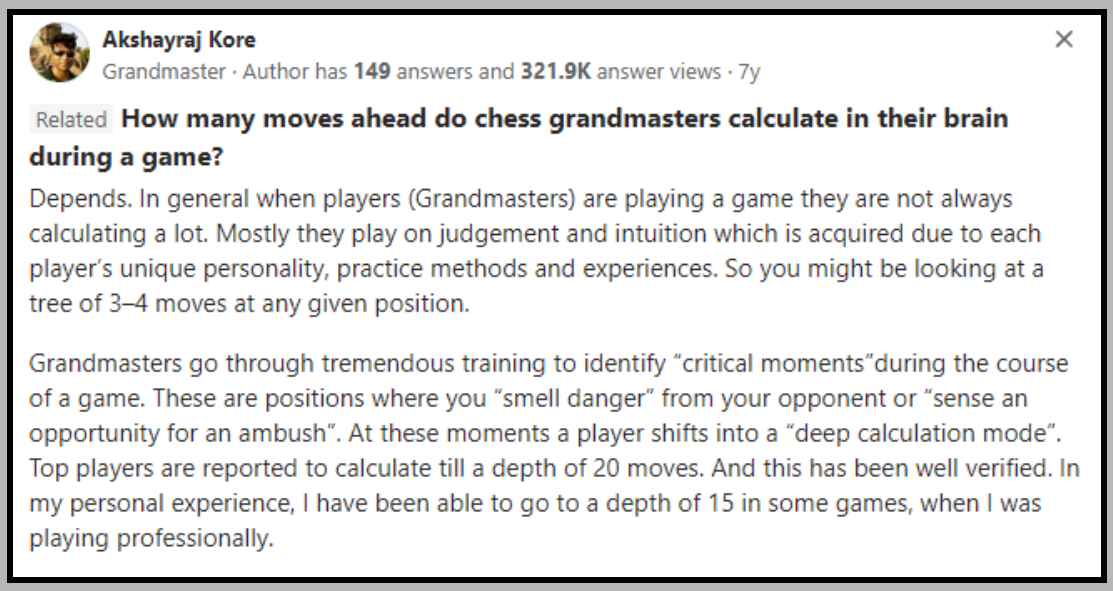
3. The calculation wasn't necessary to find that move.

This is the idea that Ayanokouji could have made his move that was better than the engine without the need for the same calculation. This has to do with him using his superior chess experience and pattern recognition to find the best move in a position. However, it is impossible for this to be the reason Ayanokouji was able to make the move that he did. Firstly, based on the pure number of games analyzed, a chess engine has massively more chess experience than Ayanokouji. The computer can use the positions from these analyzed games to find trends and apply those to unfamiliar positions. In addition, while Ayanokouji can certainly use his pattern recognition skills to aid him, that wouldn't be sufficient to accomplish this. Computers are discarding objectively bad moves but analyzing the rest to find the "best" move from a given position. This means that the engine would have considered the move

Ayanokouji made as a possibility although ultimately making a different one. However, why did the engine choose a different move? Either Ayanokouji evaluated the move as better based on the situation which is not a possible way his move could be "better than the engine" or he calculated more moves and realized that in the longer-term this move is better than the move the engine found. This not only proves Ayanokouji needed to calculate more positions than the computers, but that he calculated at a higher depth.

4. This is something that Chess Masters can do.

Regarding the Chess Game predictions (Feat 1), it is possible for Chess Masters to achieve similar kinds of feats. We observe in high level chess play, people such as Magnus Carlsen or Hikaru Nakamura talk about making predictions or pre-moves based on what they think their opponent will do. However, the length at which Ayanokouji made predictions being 68 moves in the first half of the match and between 90-120 in the second half toward the endgame are both significantly longer than anything performed by normal Grand Masters. If we consider his minimum of 45 move line in the second phase this is even more apparent. Bobby Fischer who is a former world champion has talked about a 15 move line he calculated to win a tournament match. This is considered an incredible feat against a GM level opponent. Making an argument that it is normal for a GM to calculate 50 or more moves in advance is just insane and well beyond normal human ability. Even in endgame situations where it becomes much easier to calculate lines since there are far fewer pieces on the board, making a 20-30 move calculation is very difficult.



Akshayraj Kore
Grandmaster · Author has **149** answers and **321.9K** answer views · 7y

Related **How many moves ahead do chess grandmasters calculate in their brain during a game?**

Depends. In general when players (Grandmasters) are playing a game they are not always calculating a lot. Mostly they play on judgement and intuition which is acquired due to each player's unique personality, practice methods and experiences. So you might be looking at a tree of 3–4 moves at any given position.

Grandmasters go through tremendous training to identify "critical moments"during the course of a game. These are positions where you "smell danger" from your opponent or "sense an opportunity for an ambush". At these moments a player shifts into a "deep calculation mode". Top players are reported to calculate till a depth of 20 moves. And this has been well verified. In my personal experience, I have been able to go to a depth of 15 in some games, when I was playing professionally.

- Grand Master Akshayraj Kore talking about how GMs calculate

Regarding the Board State Calculations, it is just false to claim Chess Masters could do this. Chess Masters haven't been able to compete against chess engines since the 1990's. There are instances of Chess Masters playing an engine while it is still being "trained" or while it is still in development, like when Magnus Carlsen played an older version of Stockfish and was able to win. However, once the engine was complete even the best players in the world stand no chance against an engine. The last time a human was able to beat a chess engine was the current World Champion Gary Kasparov vs IBMs Deep Blue in 1996. They once again played in 1997 and the computer won. Since then computers have crushed humans. For additional reference chess players are ranked on a rating scale by FIDE as a method to quantify their chess rating. While this isn't a perfect system it gives some clarity on how they compare to each other. The top Grandmasters in the world rank around 2800. When compared to stockfish, one of the most popular chess engines, it has a rating close to 4000, and potentially higher. Deep Blue however had a rating around 2700 which is substantially lower than Stockfish or Alpha-Zero. When talking about what it is like to play against an engine, the 2nd best chess player in the world, Hikaru Nakamura explained that he would have no chance. He said the difference between his abilities at chess and a novice is a similar gap compared to him and stockfish. The difference is about 1000 rating. A rank of 1800 is an amateur player and nowhere near GM level. Hikaru said compared to stockfish, he doesn't even understand the game of chess. He even said that if he worked together with Magnus Carlsen, the #1 rated player, they would still have no chance against the engine 2v1.

https://www.youtube.com/watch?v=0T-1sG-sZLU&list=LL&index=4

- Hikaru talking about playing an engine

5. That calculation isn't possible.

The final argument against this feat is the fact that this level of calculation just isn't possible for a human. This is an invalid argument for analyzing fictional characters. Just because something isn't possible by real world standards doesn't mean it is impossible in fiction. If this argument were valid then every fictional character would be downgraded to normal human level.