What Your Engine is Trying to Tell You

Part 1: What's a pawn?

BY **GM LARRY KAUFMAN**

IT USED TO BE SO SIMPLE.

They had different names, but way back in the 20th century, the best chess engines were quite similar. They used the same search method — a tried-and-true programming technique called alpha-beta pruning. They evaluated the terminal positions by counting up the material, and then modifying the scores for hand-coded positional factors like square control, pawn structure, and safety. And they only ran on one core. (Your computer likely only had one core back then.)

Naturally there were some stylistic differences. Fritz would value things differently than Junior, while Shredder and Hiarcs were not nearly as speculative as CS Tal. But generally speaking, you could just get the top-rated engine and feel confident in your understanding of the evaluations it spit back out. A score of +1.30 meant that White was up a pawn with a bit of an additional positional advantage. The engines were strong tactically, but pretty weak positionally, so their advice could not always be trusted.

Since then, things have changed.

Near the turn of the century the first multi-core processors became available, making search faster but also (because of parallelization) unpredictable. Repeating a search on multiple cores will often give a different move and evaluation.

Search algorithms were improved. Searches got much deeper with new techniques for pruning unpromising lines, making the engines more "selective" and stronger.

Evaluation functions also became much more sophisticated. Material advantage, for example, started to be treated differently based on game stage. If, for example, you are up a healthy, undoubled pawn in a pawn ending (with at least two pawns for the better side), and there is no compensation for the defender, you're usually winning quite

PHOTO: COURTESY SLCC / LENNART OOTES

easily unless the defender has some obvious tactical save. Often it takes just minimal skill to promote the pawn and deliver checkmate, so the result is that a pawn-up endgame may actually be more easily won than a queen-up middlegame.

This is one of the reasons that strong engines began in those years to show values far above +1.00 for such endgames, while returning scores of below +1.00 for a clean pawn advantage in the opening. While such positions are theoretically winning, all being equal, the defending side always has decent drawing chances, especially in human play.

But the first big breakthrough came in 2017 with AlphaZero. Using a different processor type (GPUs instead of CPUs), neural networks for evaluation instead of human-derived rules, and a radically different type of search ("MCTS" or Monte Carlo tree search) that is probabilistic in nature, it was able to crush the best classical engines of its time.

AlphaZero was never made public, but soon enough an open-source alternative called Leela (Lc0) was launched. Leela was designed on the principles underlying AlphaZero, and was nearly identical, save the fact that it could run on consumer hardware. It played much different chess than the top alpha-beta engines of the day, somewhat weaker tactically but much stronger positionally, and for a couple of years it was the engine of choice for anyone who could afford a computer with a powerful GPU.

Then came the next big breakthrough, the one whose effects we are witnessing now. Programmers in the Shogi (Japanese chess) world developed smaller neural networks suitable for CPUs that replaced their evaluation functions. These "neural networks updated efficiently," or NNUEs, were remarkably successful for Shogi engines, and it was claimed that this programming technique would add 100 rating points to the top engines.

Most engine enthusiasts were skeptical, but with my experience as one of the top non-Japanese shogi players, I was quickly convinced by the radical strength increase this new technique wrought in Shogi engines. Soon chess programmers began to tinker with NNUEs, and by the end of 2020, both Stockfish and Komodo had seen that predicted 100 Elo point boost. Later enhancements added greatly to that gain.

The NNUE evaluation function can be seen as a hybrid between classic engines and those like AlphaZero or Leela. It runs on the CPU, and uses the traditional alpha-beta search to zoom through analysis, but it uses neural networks trained on high-quality data over many iterations to work as its evaluative brain, determining the "score" of the position. Using training techniques not unlike those of Leela, the result is a much smaller network that may not be as "smart" as Leela's, but it does not burden the CPU and slow search down. The result is an engine that retains nearly all the tactical strength of classical engines, while also keeping much of the positional strength of Leela.

These NNUE engines quickly took the lead in engine tournaments and rating lists, and today all the top-ranked engines use neural networks for their evaluation functions. CPU based engines like the open-source Stockfish are currently in the lead, followed by Komodo (which I worked on for years) and Torch, a new, private engine from *Chess.com*. The relative placement of Leela among these three depends greatly on hardware and test conditions.

One of the curious effects of the shift to NNUEs was a disconnection between evaluation and material. By the time of Stockfish 15, for example, users would encounter positions that would have gotten a +1.00 evaluation in the old days — a simple pawn-up with no positional advantage for either side in the opening — and be surprised when Stockfish reported a score of +2.00 or more. Other positions that looked like marginal, difficult wins to the human eye would receive absurd evaluations like +5.

A pawn was no longer a pawn. And users were getting confused.

Starting with version 15.1, the Stockfish community opted for a new standard, one that the other major engines have also adopted. Unmooring a strict relationship between material and evaluation, a score of +1.00 is now defined as a position where White is expected to win 50% of the games from that position, while Black will either draw or win the other half. There is some difference in how this standard is applied — Stockfish pegs this to move 32, while Komodo looks to the opening, and Leela applies it everywhere — but that is just a detail.

The main thing to know is that any evaluation above 1.00 means that the engine believes the position is more likely than not won with perfect play. A score between 0 and 1 means that the position is evaluated with advantage to one side, but more likely than not to be drawn with perfect play. Since it generally takes about a 0.7 pawn advantage to reach this 50/50 dividing line at the super-GM level, we might say that a clean pawn-up should show an evaluation of +1.4 on average — 1/0.7 = 1.43.

So how do we interpret these new evaluations, now that they are not strictly speaking in pawn units?

I propose a simple way to think about them, one that might not be technically correct, but is good enough for over-the-board use.

If the evaluation is above +1.00, the position is likely already won with perfect play. The number just indicates how certainly this is so, with higher numbers showing increasingly more certainty. Anything above +2.00 indicates high confidence in the winning assessment.

Note that this does not tell us anything about the ease of winning. The engine might report one position to be +10, but you might have to find five consecutive brilliant moves to justify the evaluation! This is especially true with the NNUE engines; with Leela, there is more of an attempt in the search to assess the difficulty in finding the win. (More on this later.)

If the evaluation is between 0.00 and 1.00, you should think of it as representing the probability that the game will likely reach theoretically winning status before reaching total equality status, given that the players are human and likely to err.

For example: if the position is evaluated as +0.50, it should be equally likely to reach 0.00 (totally equal) or +1.00 first, given the random errors made by humans. (This would not be true if the potential errors were tiny, as they are in engine versus engine play.) If the position is +0.80, it is


GM Larry Kaufman

PHOTO: COURTESY DAVID LLADA

obviously more likely to wander up to +1 than down to 0.00.

One of the practical ramifications of this line of thinking, and one that we will discuss extensively in the second installment of this article, is engine evaluation and opening choice. Someone playing Black in ICCF correspondence chess, where engine us is fully legal, might be willing to play positions that the engine evaluates as -0.70, knowing that it is probably drawable by the engine. A human playing over-the-board chess should avoid such positions, unless they believe that the positions are harder to play in practical terms for the opponent than themselves.

This is where human judgment becomes important. Consider a situation where White sacrifices a pawn for superior development. Is it easier to find the right moves for White with the dynamic advantage, or for Black with the static / material advantage? Is the defending side able to find the best defensive ideas easily, or are they counter-intuitive? There is no general answer; any assessment is necessarily concrete and position-dependent.

It's also important in this context to understand how the top engines are different from one another, and how the hardware used can affect what the engines output.

Most serious players use Stockfish now for analyzing games, since it generally tops the rating lists and wins elite engine tournaments. Because it is open-source software and can run on many different kinds of processors, it's everywhere — on cellphones and iPads, on laptops and "in the cloud." That kind of ubiquity, however, can obscure the truth. The specific version of Stockfish being used, and the hardware it runs on, can affect the quality of its analysis.

The Stockfish on your cellphone may be Stockfish 11, which was the last version with a human-crafted evaluation (HCE) and without NNUE. *Chess.com* offers Stockfish 16 lite (using legacy HCE) and "normal" Stockfish 16, along with Komodo lite (HCE only) and two versions of their private engine Torch: one with a full-sized NNUE evaluation, and one with a distilled, smaller version.

It's much the same on Lichess, where users can choose between Stockfish 11 HCE, Stockfish 14 NNUE, and two versions of Stockfish 16: a lite version with a 7 megabyte evaluation network, and one with the standard 40 megabyte NNUE evaluation file.

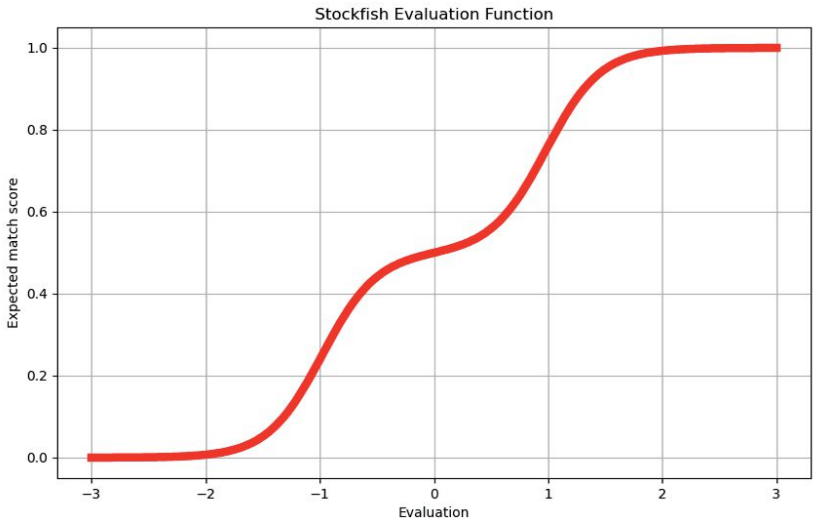
Users can adjust the number of lines being analyzed on both *Chess.com* and Lichess. They can also modify the number of threads / cores (most modern CPUs offer multi-threading cores) being used for analysis, and the size of a modest hash table, on Lichess. But to do any of this, a user must actively enter the settings dialog, when the fact of the matter is that few users will go to these lengths. Why bother when you can just turn the engine on and watch the analysis start?

Up-to-date engines offer the most accurate analysis, and more powerful hardware allows you to get that accurate analysis faster. The Stockfish that runs on your phone will be very slow compared to what you will get on a PC with four, eight, or 16 cores. It's true that you can get nearly the same quality analysis using one core on your cellphone as you could on a 128-core / 256-thread Threadripper, but it would take about 100 times as long. What would take five minutes to find on your phone would take just seconds on a fast PC.

The number of cores being used can also, in marginal cases, affect the analysis itself. Stockfish is programmed to prune more or less moves from its search depending on how many threads it has available. When running on just one thread, it will do much more pruning than it will running on 16 threads. With less pruning comes more chance that the engine can accurately assess moves that it would otherwise initially reject.

What does all this mean? Speed is linked to quality in computer chess. More cores means more speed, and up-to-date engines can take full advantage of increased computing power. Stockfish 11 on your iPad is good enough for a tactical game review, but if you want really high-quality positional suggestions, you should be using Stockfish 16 or later, or Dragon 3.3, running on a good PC.

Here's an example where the computing power mattered.



Left: This graph, based on normalized WDL data, shows how Stockfish evaluations map onto expected match scores or outcomes. Scores of +2 or -2 are seen as near-wins!

ONE OR THE OTHER

GM Leinier Dominguez (2745)
GM Pranav Anand (2520)
10th Sunway Sitges (5), Sitges, 12.16.2023

BLACK TO MOVE

This is a pivotal moment that two annotators — YouTube's Agadmator and Colin McGourtey for *Chess.com* — overlooked. In both cases I suspect it was because they didn't let the engine run long enough.

It turns out that **28. ... Kh7** is a clear mistake, allowing White to stay in the game after the game continuation **29. Rxc1 Rxc1+ 30. Kh2 d5 31. Qg3 Rc3 32. f3 Rd8 33. Nb5** (compare 33. Qg4 Bd6+ 34. g3 Be5 35. fxe4 Bxd4 36. exd5 is equal, but with the king on h8, Black would be winning) **33. ... Bf5** (compare 33. ... Bd3 34. Nxc3 bxc3 35. f4 Bf5 36. Qxc3 and White wins) **34. Qe1 Rcc8 35. Be6 Bxe6 36. Qxe6 Kh8 37. f4 Bc5 38. f5 d4**

IMAGE: WTH THE ASSISTANCE OF THE STOCKFISH DISCORD

39. f6 d3 40. fxg7+ Kxg7 41. Qe5+ Kg8 42. Qe6+ Kg7 43. Qe5+ Kg8, draw.

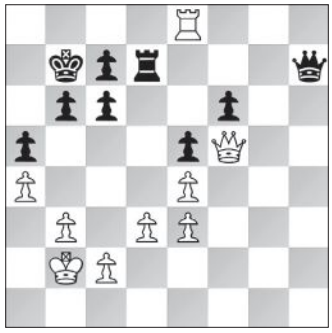
Writing for *Chess Life Online*, JJ Lang discovered that **28. ... Kh8!** actually gives Black a significant advantage. The king placement makes a difference in two variations. After **29. Rxc1 Rxc1+ 30. Kh2 d5 31. Qg3** (*31. Bxe8?* loses to *31. ... Bd6+*) **31. ... Rc3 32. f3 Rd8 33. Nb5** (if instead *33. Qg4 Bd6+ 34. g3 Be5 35. fxe4 Bxd4 36. exd5 Be5* and the bishop is safe due to the lack of a check!) **33. ... Bd3! 34. Nxc3 bxc3 35. f4 c2** and with the king on h8, White cannot take the d3-bishop with check!

Here the default engine on *Chess.com*, Torch, initially evaluates both 28. ... Kh7 and 28. ... Kh8 as roughly equal. Only when it gets to depth 34 or higher does it recognize the difference, and reaching that depth takes time on a laptop or portable device. A quick scan or automated analysis would not pick it up.

Moving on: what value are tablebases? (Tablebases are presolved endgame databases that engines can use to accurately assess positions with up to seven pieces in the search.) In general, I think they are overrated, because they don't raise the quality of engine assessments more than a few Elo points — engines play the endgame very well now without them. But they can come in handy. Consider this position from a recent game between Magnus Carlsen and David Anton Guijarro.

TO TAKE OR NOT TO TAKE

GM Magnus Carlsen (2847)
GM David Anton Guijarro (2673)
Carlsen Inv (5), *Chess24.com*, 03.13.2021



BLACK TO MOVE

If you were to analyze this game on *365chess.com* with their embedded version of Stockfish 16, you would (wrongly) believe that White was better but not necessarily winning after either 41. ... Qg7 (+0.37 at depth 42) or 41. ... Qxf5 (+0.72 at depth 42). After testing, I believe that the *365chess* engine has the NNUE evaluation turned off but does not disclose this fact.

The embedded Stockfishes (which do not include tablebases) on Lichess using four threads and a 512 megabyte hash table tell a different story: +0.4 and +2.3 at depth 42 for Stockfish 16, and +0.3 and +2.9 for Stockfish 16 lite at depth 45. They see 41. ... Qxf5 as a serious mistake.

Running on 16 threads with a 16 gigabyte hash table, the latest Stockfish equipped with six and selected seven-man tablebases is more critical still, deeming 41. ... Qg7 to be well within the drawing range at +0.35 (depth 40), while 41. ... Qxf5 is an egregious error at +4.64 (depth 40).

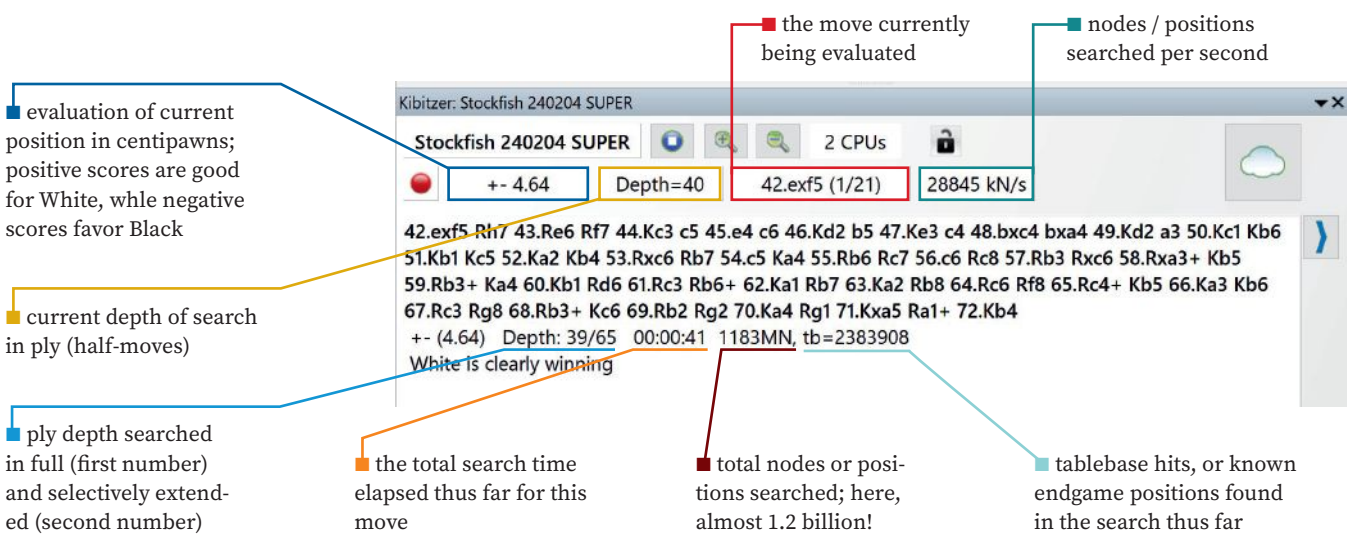
The point? The accuracy of engine analysis grows with increased resources. The Stockfish embedded in a website might only run on one core with limited hash tables, resulting in misleading output. More processing power and longer search time can overcome these problems, and the NNUE evaluation function adds to analytical accuracy immensely.

(In case you were wondering: in the game Anton Guijarro took the queen with disastrous results.)

What about Leela? Despite it needing specialized hardware for optimal use — an RTX 2060 graphics card or higher — I think it offers two advantages over Stockfish. The first is that it can analyze multiple lines without any cost. Analyzing in "multi-PV" is an important tool, as showing the difference between the first- and second-best moves can immediately point out forced continuations. With Stockfish, this can slow the search significantly. With Leela, there is no computational cost in doing so.

WHAT DOES ALL THAT STUFF MEAN?

Have you ever wondered what the engine window is trying to tell you? Our editor took a screenshot of his ChessBase screen while working on Carlsen – Anton Guijarro after 41. ... Qxf5. Here's what all those numbers actually refer to.



ART: NATASHA ROBERTS

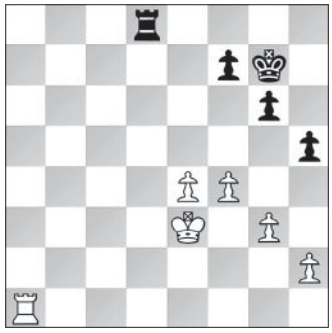
The second benefit is potentially more important. Users should understand that Stockfish and Leela are trying to answer different questions in their searches. Stockfish tries to find the strongest moves for both sides (assuming perfect play) at every turn, while Leela, using its "Monte Carlo" style search, inquires after the best moves against good but unpredictable play.

One common way that this difference shows up is the famous 0.00 evaluation from Stockfish, which is given when Stockfish "believes" that best play ends with a draw by repetition. Leela proceeds differently. Even if it expects a repetition, the side that more likely to go wrong will usually show a mildly negative score.

Imagine a situation where White is a rook down, looking for a difficult perpetual that requires accuracy over a number of checks. Leela might evaluate this as -0.10, indicating a probable, if non-trivial, draw where Black is the only one who might win. This is very helpful for a practical player, although it can be annoying to see a -0.03 evaluation when the perpetual is obvious.

Understanding engine output, and how Stockfish is different from Leela, is very helpful for a practical player. With experience, you will learn which evaluations indicate draws you can readily spot, and which are more difficult. There are countless positions in chess that are clearly drawn with perfect play, but which are also clearly easier for one side to win. Stockfish will show all zeros in such positions, while Leela will usually return a decent plus score for the "easier to play" side, even after a long think. Here is a simple example:

FOUR VERSUS THREE



WHITE TO MOVE

This four versus three rook endgame is known to theory as a draw, but it is well worth White's time to play out, as Black has to keep finding good moves to draw. On my modern laptop, not using tablebases, the current version of Stockfish shows a +0.04 score almost immediately, dropping to 0.00 after a couple of minutes. The current Leela shows +0.19, and +0.18 after a few minutes time.

Stockfish is telling us what the result of the game will be with 3500-level play, while Leela is telling us the likely result of the game with "only" human GM-level play. Stockfish says "draw," while Leela says "should be a draw, but Black may lose if they don't play accurately."

It's for this reason that, for most over-the-board players, I recommend using a Monte Carlo style engine like Leela if you have a modern GPU, and a MCTS-enabled engine like Komodo if you don't. Naturally you can (and should!) consult Stockfish as well, but you will have to make your own judgments about the difficulty of the moves for both sides.

I hope you have learned something about today's chess engines in this article. Next month, we'll apply this knowledge to opening study, learn to interpret what modern engines are telling us about key opening tabiya, and try to deduce some principles of contemporary opening play.

A GLOSSARY OF TERMS

Cores / threads – modern processors differentiate between physical cores and "threads," or virtual instruction streams. Some cores are singled threaded, and some are dual threaded.
CPU – central processing unit; the calculative engine in your computer.
Evaluation – the part of a chess program that assesses each position in a game tree and assigns it a numerical value.
GPU – graphics processing unit; generally speaking, your graphics card.
Hash table – the memory used by a chess engine to store evaluations and refer to them in search.
HCE – human (or hand) crafted evaluation; the older "brain" for chess engines, using recognizable positional terms to score positions.
Neural network – an AI method of machine learning that trains computers to deal with data inputs through intensive analysis of training data. They are used for evaluation functions in chess engines.
NNUE – "neural network updated efficiently;" the new "brain" for CPU-based chess engines.
Search – the part of a chess program that sorts and chooses moves in a game tree.
Parallelization – a situation where one algorithm or function "runs" on multiple cores or threads.

Left: GM Pranav Anand (L) and GM Leinier Dominguez (R) analyze their game (as discussed on page 18) the old-fashioned way: with no silicon assistance allowed!

IMAGE: TWITTER

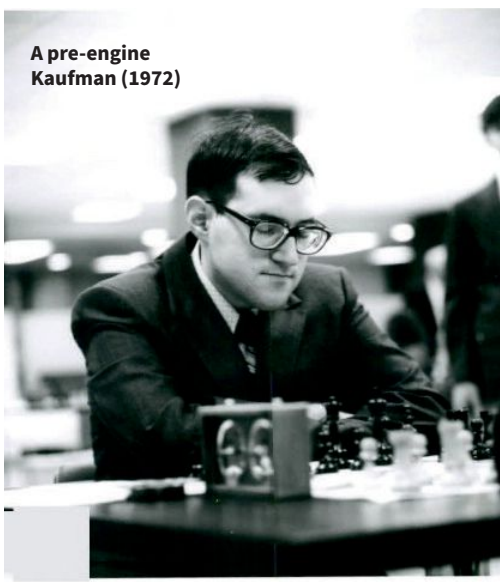
What Your Engine is Trying to Tell You

Part 2: In the opening, space matters.

BY **GM LARRY KAUFMAN**

LAST MONTH WE TOOK AN IN-depth look at how modern engines work. More precisely, we discussed how the old standard of +1.00 being equivalent to "being up a pawn with no additional positional advantage" no longer holds. In its place, today's engines, using different versions of neural nets for their evaluation functions, have unhitched any strict relationship between material and evaluation; instead, a score of +1.00 is now defined as a position where White is expected to win 50% of games from that position, with Black drawing or winning the other half. The result is a normalized evaluation that does not drift as neural nets evolve and change, but one that also "feels right" to human users.

While I recommend you read last month's installment for more on this important shift in computer chess, and on the key differences between today's top engines, this month I want to focus on the opening and think about what these new engines are revealing in that phase of the game.



A pre-engine Kaufman (1972)

This last part is important. We knew well before Crafty or Fritz were developed that certain positions were bad. As engines have improved, they have helped humans refine opening theory, and with the great leap taken in recent years via AI-enhanced evaluation functions, certain truths about a number of opening positions are being clarified faster than ever.

Consider the Berlin Defense — **1. e4 e5 2. Nf3 Nc6 3. Bb5 Nf6 4. 0-0 Nxe4 5. d4 Nd6 6. Bxc6 dxc6 7. dxe5 Nf5 8. Qxd8+ Kxd8** — which GM Vladimir Kramnik used to such great effect in his 2000 world championship victory over GM Garry Kasparov.

Here the top engines of the day (Fritz 6 and Junior 6) both saw White as being better by a score of around +0.35 to +0.40. Today, Leela and Stockfish recognize what Kramnik did — White's advantage is ephemeral at +0.10 to +0.15.

The "Elista Endgame" — **1. d4 d5 2. c4 c6 3. Nf3 Nf6 4. Nc3 dxc4 5. a4 Bf5 6. Ne5 e6 7. f3 c5 8. e4 Bg6 9. Be3 cxd4 10. Qxd4 Qxd4 11. Bxd4 Nfd7 12. Nxd7 Nxd7 13. Bxc4**, as played by Kramnik and Anand with Black in world championship matches — is a more recent example.

Hiarcs 13.1, used by Anand in his match preparation in 2010, sees this position as manageable for Black, with White ahead by +0.35 or so. Contemporary engines deem it borderline lost, with Leela giving White a stable, serious advantage of around +0.55.

It turns out that many openings that were popular in the last century now fall into this "dubious" category, which explains why they are rarely seen at elite levels except in blitz or rapid play. But perhaps we should be more precise here. What is the dividing line in the opening between (from Black's perspective) theoretically correct and theoretically dubious?

Last month I suggested that an advantage of roughly 0.7 pawns is the dividing line between a win and a draw with correct play. Humans, however, rarely play fully correct games. I submit that if the opening phase of a game ends with an engine evaluation of anywhere above +0.50, it is more likely than not that the game will reach expected winning territory (over +1.00) than reach dead equality (0.00) by move 30. In other words, an opening ending with an evaluation of +0.50 should be considered dubious for Black, as they are likely to be losing at some point given the reality of human fallibility. The opponent may not find the way to the win, but in principle, such positions are undesirable.

PHOTO: US CHESS ARCHIVES

In what follows, I will be citing evaluations from Leela (version 0.30, net sva5230000) on a powerful graphics card (RTX 4090 mobile), in order to compare results with the circa 2008 Rybka 3 used by Anand in his preparation for the Kramnik match. I do so for reasons enumerated last month: the "Monte Carlo" style search utilized by Leela provides evaluations that are more representative of grandmaster encounters over-the-board. Please feel free to reproduce these results with current Stockfish or Komodo; in my experience, using another engine doesn't change much.

Let's look at some examples of major openings that have fallen from favor in the era of neural nets, and then try to understand the reasons why they have done so.

KING'S INDIAN, MAR DEL PLATA VARIATION

1. d4 Nf6 2. c4 g6 3. Nc3 Bg7 4. e4 d6 5. Nf3 0-0 6. Be2 e5 7. 0-0 Nc6
Here 7. ... Na6 is now looking better at +0.45, which, while bad, is at least above the "dubious" mark for Black.

8. d5 Ne7
This was a major defense to 1. d4 throughout the last century.

Now the Bayonet Attack, beginning with **9. b4**, gives a +0.58 eval, well into the dubious range. The most popular variation runs **9. ... Nh5** (relatively better per Leela is *9. ... a5*) **10. Re1 f5**. Now the recent move **11. a4** scores highly for White; after **11. ... a5 12. bxa5 Rxa5 13. exf5 Nxf5 14. Bg5 Nf6 15. Bd3**

White has more space, more active pieces, and a +0.60 eval per Leela, while the 2008 Rybka 3 thinks White can only claim a +0.19 advantage! Of course this position is not lost for Black, but it is obviously more pleasant for White and it will not be easy for Black to hold. Don't expect many King's Indians in the Candidates' tournament!

PIRC DEFENSE

1. e4 d6 2. d4 Nf6 3. Nc3 g6
The Pirc and Modern together were moderately popular in the last century in high-level play; now they are generally only seen in Rapid/Blitz games. (The "Modern Defense," 1. ... g6, is apt to transpose eventually though there are independent options which are in general not objectively better.)

Here both 4. f4 and 4. Be3 give evals above +0.50; I'll show:

4. Be3 Bg7 5. Qd2 c6 6. a4 0-0 7. f3 (Top players of the last century would not have approved this combination of a2-a4 and f3-f3, but engines like it and we can see why now.) **7. ... Qa5 8. Nge2 e5 9. g4 Nbd7 10. g5 exd4 11. Bxd4 Nh5** (11. ... Ne8 should transpose) **12. Bxg7 Nxg7 13. 0-0-0**

Leela evaluates this as +0.52, while Rybka returns just +0.19. Both the a4- and d6-pawns are weak, but White has more space and a head start in the attacking race with g2-g4-g5 already played.

BENKO GAMBIT

1. d4 Nf6 2. c4 c5 3. d5 b5!?
This was quite popular in the last century — I was one of Pal Benko's earliest victims of the gambit in the 1960s! — but has almost disappeared from high level play.

4. cxb5 a6 5. bxa6
Here 5. e3, as I played versus Benko, is also good, giving back the pawn but keeping a positional edge that approaches the "dubious" line.

5. ... Bxa6
Similar is 5. ... g6, while 5. ... e6 may keep White's edge below the dubious line, but it moves outside the Benko Gambit proper, and it is not very pleasant for Black.

6. Nc3 g6 7. e4 Bxf1 8. Kxf1 g6 9. g3 Bg7 10. Nf3 0-0 11. Kg2 Nbd7 12. a4!

Rybka believes Black has some play for the pawn with a +0.50 evaluation, while Leela rightly sees this as +0.80. With moves like Ra1-a3, Qd1-c2, and Nc3-b5 likely, White is consolidating the extra pawn for just mild compensation, with a space advantage to boot. Black may be able to hold with perfect play, but it will be very difficult.

PHILIDOR'S DEFENSE

1. e4 e5 2. Nf3 d6 3. d4 exd4 4. Nxd4 Nf6 5. Nc3 Be7
This had a bit of popularity about 15 years ago — I took it up myself then — but now it is in the "dubious" bag.

6. Bf4 0-0 7. Qd2 Nc6 8. 0-0-0 Nxd4 9. Qxd4 a6 10. e5
Also fine for White is 10. f3 with a pawn storm coming.

10. ... dxe5 11. Qxe5 Bd6 12. Qd4 Bxf4+ 13. Qxf4 Qe7 14. Bc4 Be6 15. Rhe1 Rad8


GM Larry Kaufman today

PHOTO: COURTESY DAVID LLADA

16. Rxd8 Rxd8 17. Bxe6 fxe6 18. Ne4

This is evaluated by Leela at +0.58, or a 65% White score. Rybka is more sanguine at +0.32. To my eye, the black pawn on e6 is isolated and very weak, and if it is lost without compensation, White should win with pawns on both sides of the board. Perhaps Black can draw with perfect play, but no one would want to defend this over-the-board.

BUDAPEST GAMBIT

1. d4 Nf6 2. c4 e5
This gambit has always had a decent amount of popularity, and was strongly recommended in a book by GM Moskalenko a few years ago.

3. dxe5 Ng4
The Fajarowicz Gambit, 3. ... Ne4?, is clearly losing after 4. a3 per Leela at +1.33.

4. e4 Nxe5 5. f4 Nec6 6. Nc3
White's space advantage used to be considered only enough for a small plus, but engines love space. Over time, they have convinced us that they are right!

6. ... Bc5 7. Qg4 0-0 8. f5 d6 9. Nf3 Re8 10. Nd5 Kh8 11. Bg5 f6 12. Bd2 Nd7 13. 0-0-0

White's space and development advantages are far more significant than the backward e4-pawn. Leela's evaluation of +0.58 is much closer to the truth than Rybka's +0.10.

SICILIAN DRAGON

1. e4 c5 2. Nf3 d6 3. d4 cxd4 4. Nxd4 Nf6 5. Nc3 g6
Always popular, but it fell out of fashion due to the Yugoslav Attack.

6. Be3 Bg7 7. f3 0-0 8. Qd2 Nc6 9. 0-0-0
Here 9. Bc4 was the main move in Fischer's heyday, but now it seems that the text move is the only way to prove it "dubious."

9. ... d5 10. exd5
10. Qe1 is also quite good.

10. ... Nxd5 11. Nxc6 bxc6 12. Bd4 Bxd4 13. Qxd4 Qb6 14. Na4 Qc7 15. Bc4 Rd8 16. Nc5

White has both better development and a better pawn structure. This is +0.51 for White per Leela, while Rybka thinks the position is basically equal at +0.08.

MODERN BENONI

1. d4 Nf6 2. c4 c5 3. d5 e6 4. Nc3 exd5 5. cxd5 d6
A favorite of dynamic players, especially GM Mikhail Tal in the last century. It now looks nearly lost.

6. e4 g6 7. f4
Note that when White has played an early Ng1-f3, this move is illegal; here, the Benoni isn't quite as bad.

7. ... Bg7 8. Bb5+ Nfd7 (other moves lose to 9. e5) **9. a4 0-0 10. Nf3 Na6 11. 0-0 Nb4 12. h3 a6 13. Bc4 f5 14. e5 dxe5 15. d6+ Kh8 16. Ng5 e4 17. Be3 Qf6 18. Qe2 Qxd6 19. Rfd1 Qc6 20. Nd5**

(see diagram top of next column)

This position is a striking example of how engine evaluations have progressed. Rybka thinks that the position is fully unclear, returning an evalaution of -0.11. Leela believes

this position is nearly winning for White at +0.85, and White scored 5/7 in my database.

It is difficult for humans to judge whether White's large lead in development or Black's two extra pawns are more weighty here, but modern engines quickly see that development is key here, not the pawn count.

RUY LOPEZ, STEINITZ DEFENSE

1. e4 e5 2. Nf3 Nc6 3. Bb5 d6
Good enough for the first World Champion, it retained some popularity into the 1900s, especially among amateurs, as it is easy to play and only concedes space.

4. d4 exd4 5. Nxd4 Bd7 6. Nc3 Nf6 7. Bxc6 bxc6 8. Qf3 Qb8 9. 0-0 Qb6 10. Be3 Ng4 11. Rab1 a5 12. a3

White's huge lead in development is worth far more than the bishop pair, making Black's opening dubious. Rybka says +0.36, while Leela gives White a +0.64 advantage.

So which defenses are deemed satisfactory now? The engines strongly believe that after 1. e4, pushing the king's pawn with 1. ... e5 2. Nf3 Nc6 is the only path to near-equality, but the Petroff, the Sicilian (Najdorf and Sveshnikov), and the Caro-Kann give evals of around +0.30, which is at least not near "dubious" territory. Speaking generally, White will get some tangible but modest advantage (usually space, central control,

or better pawn structure) in these lines, but not two small advantages or one large one.

The French usually ends up around +0.40 — still not "dubious," but starting to get uncomfortably close to it, and so it is not very popular anymore at high level. The main problem is that Black usually gets two "strikes": less space and a bad (light-squared) bishop.

Playing 1. ... Nc6 is also around +0.40 after 2. d4 but has little merit since if White plays 2. Nf3 Black's only non-dubious option is 2. ... e5, transposing to normal 1. e4 lines. Other moves are in or near the dubious zone.

Pretty much all Black gambits against 1. e4 are dubious per the computer, except the Marshall Gambit in the Ruy Lopez, which the engines now approve of! In summary, you have at least five defenses to 1. e4 that are not deemed dubious. This is sufficient to provide variety and cater to differing tastes.

Against 1. d4 only 1. ... d5 and 1. ... Nf6 are near-equal. Both 1. ... c6 and 1. ... e6 allow White to play 2. e4, without gaining anything tangible in return.

The Dutch, rather popular in the past among amateurs, didn't quite make my dubious list, but both 2. g3 and 2. Bg5 give White something like +0.45, very near that line.

The basic problem is that 1. ... f5 doesn't aid development, and in the opening every move is precious. Many tests have shown that if Black wastes even a single tempo at the start, he is dangerously close to losing, i.e., 1. e4, Black "passes" (illegal, I know), 2. d4 is around +0.90. It's true that 1. ... f5 is much better than "passing," as it does gain space, but the move is only half-useful.

After the two best Black first moves, the best defenses to the Queen's Gambit, namely the Queen's Gambit Accepted and Declined, Slav, and Nimzo-Indian all give comfortable evals below +0.20, with the Grunfeld only slightly above this. Players often concede the bishop-pair for mild compensation in Nimzo lines, and the others mostly concede a bit of space, but all in manageable amounts.

Pretty much everything else is dubious or nearly so, except perhaps the Queen's Indian Defense, which is evaluated at around +0.30. As with king's pawn openings, we have about five satisfactory defenses to choose between versus the queen's pawn. The engines have greatly reduced the number of "approved" defenses to the two leading initial moves, but there is still ample variety to keep the opening phase of the game interesting.

What does this tell us about the age-old question of whether 1. d4 or 1. e4 is "best?" (Note that 1. Nf3 usually transposes to 1. d4 lines, and 1. c4, although a good move, is not generally thought to be "best.") We can see from the above that the path to near-equality is much narrower against 1. e4 — only 1. e4 e5 2. Nf3 Nc6 is in that range — but if it is chosen, White may have trouble getting even the little he can expect with 1. d4.

In other words, if White expects a variety of defenses, as is typical in open tournaments, 1. e4 is probably the best move statistically. But if White expects most opponents to reply to 1. e4 with 1. ... e5 2. Nf3 Nc6, as in top-level events, then the argument for 1. d4 becomes stronger. The recent general preference for 1. e4 can perhaps best be explained by the observation that it is much easier to prepare for one specific defense than for five!

What does all this mean for the practical tournament player? If you are picking an opening for a specific game to avoid preparation, or if you are playing in an event where preparation is not practical or likely, it's fine to play defenses in the "dubious" zone. I would avoid any that are in the "losing" zone (above +1.0), unless you expect to know them much better than your opponent.

If you are playing an opponent with Black who is likely to have prepared for you, beware of repeating a previously played "dubious" variation. They only need to check it with an engine before the game, and repeat their analysis, to likely reach a position where you are at serious risk of losing without doing anything obviously wrong. When you have the white pieces, there is far less reason to play anything "dubious." By simply playing engine-approved moves, you can expect to get a position where you are at least slightly better, and can play for a win, against an imperfect opponent.

One final question: why are so many defenses of the 20th century now deemed dubious by today's engines? This is difficult to answer succinctly. Leela, Komodo, and Stockfish — all equipped with neural nets for evaluative functions — have multiple and significant advantages over traditional engines with hand-crafted evaluations. They are better in closed positions. They can sense issues with *zugzwang* much more accurately. They have much better understanding of non-forcing play.

If I was asked to give one answer, however, I would say they more accurately understand the value of *space*.

Space in chess has many definitions, but for our purposes, there's one from a Seirawan book that is easy to apply. Space, he says (and here I'm paraphrasing), involves attacks on squares on the opponent's half of the board.

Black defenses like the Pirc / Modern, the Alekhine (1. e4 Nf6 2. e5 Nd5 3. d4 d6 4. c4 Nb6, also in the "dubious" zone at +0.61), or the Scandinavian (1. e4 d5 2. exd5 Qxd5 3. Nc3 Qa5 [or 3. ... Qd6] 4. d4 at +0.57) concede a clear space advantage to White. Fans of these openings will argue that this isn't anything tangible like material, so what's the problem? Engines may say that space matters, but is this true for humans playing

Kaufman in 1996

PHOTO: US CHESS ARCHIVES

over-the-board without silicon assistance?

Put simply: yes, it's true in human games. Statistics don't lie, and win rates against these defenses are usually much better for White than in the "approved" systems. More to the point, however, I'd like to give a concrete example of why I believe this.

Consider one of the main lines of the Smith-Morra Gambit in the Sicilian:

1. e4 c5 2. d4 cxd4 3. c3 dxc3 4. Nxc3 Nc6 5. Nf3 d6 6. Bc4 e6 7. 0-0 Be7 8. Qe2 Nf6 9. Rd1 e5 10. Be3 0-0

What does White have for the pawn? Both kings are safe. Black's extra pawn is backward, but central and safe, and there are no threats. The key point, however, is that White's pieces are mostly developed to better squares, controlling more "space." I count 10 attacks on the enemy half of the board by White's bishops, four by the knights, two by a pawn, and two by a rook, for a total of 18 attacks. Black, in contrast, has just eight such attacks. I don't know how many squares of space (defined this way) equal a pawn, but the difference of 10 feels like it should be enough.

The engines evaluate this position as being a tiny bit in Black's favor, but the results of games in my database show 57% for White in 59 games. Why the incongruity?

I suspect that White would score more than Black in this position, even at amateur levels, although the balance might approach 50% as more games are decided by random blunders. I also suspect that the player with less space is more likely to blunder, and less likely to be able to convert an extra pawn without mishap, so these engine evaluations remain relevant at low levels. All things being equal, it's harder to play cramped positions, and easier to play with space.

One interesting exception would be when Black is a pawn down but has mild compensation for it. In such cases Black's chances might be equal or better in amateur games, even if they are poor at the GM level. An example might be the Schliemann defense to the Ruy Lopez; for example:

1. e4 e5 2. Nf3 Nc6 3. Bb5 f5 4. d3 fxe4 5. dxe4 Nf6 6. 0-0 Bc5 7. Bxc6 bxc6 8. Nxe5 0-0

Leela returns an evaluation of +0.57, making this a "dubious" line, but only because compensation for the pawn is deemed insufficient. In any event, White can decline the pawn and settle for a small positional plus.

So what's the big takeaway? Well, if White can achieve a plus from a pawn-down position with just space to compensate, imagine how well White would do with the same space advantage, but equal material. Modern engines have taught us many lessons, and one of the most important is that space matters!