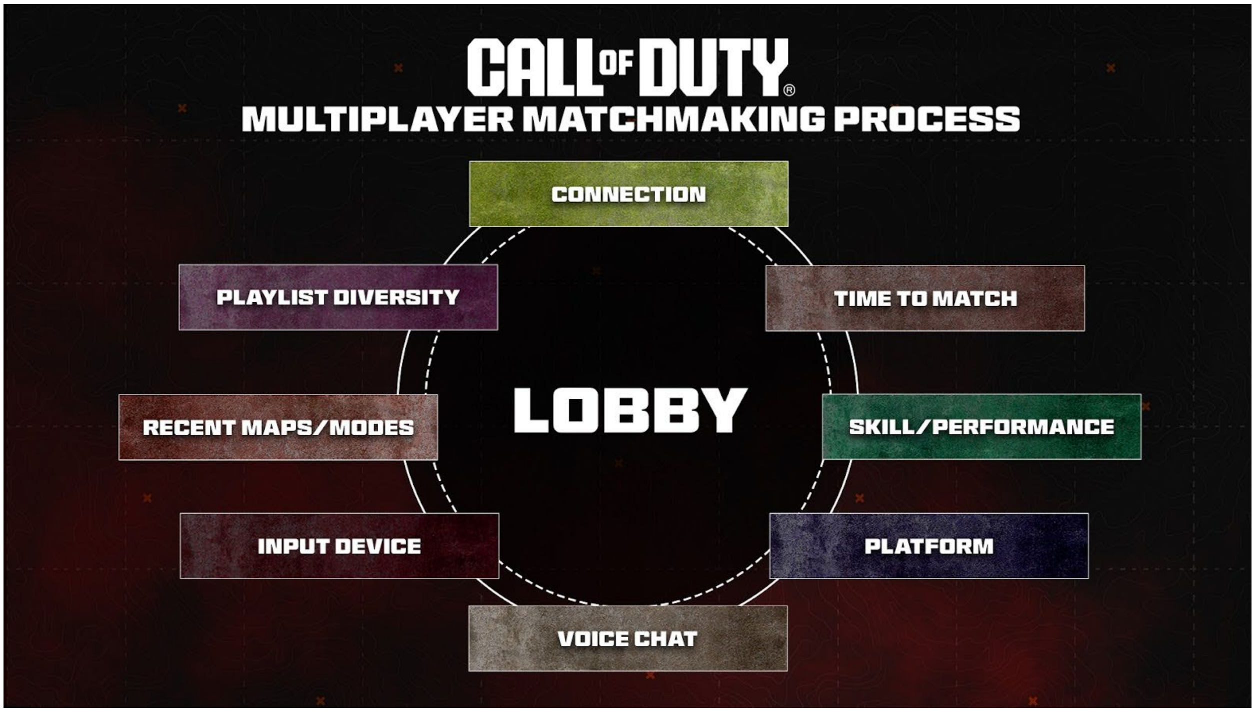
# Matchmaking Series:
# The Role of Skill in Matchmaking

## Overview

*Call of Duty* matchmaking is a complex and multifaceted domain. On April 4, 2024, we released the first in a series of white papers exploring the impact and prioritization of ping in matchmaking [1]. In this document, we will discuss the topic of skill in core multiplayer matchmaking, its implementation, and how we have observed positive results from the fine tuning of skill in the matchmaking algorithm used by *Call of Duty*.

As outlined in the Call of Duty Blog [2], skill is just one factor in the multidimensional algorithm of *Call of Duty* matchmaking. The other factors include:



1. CONNECTION – As the community will attest, Ping is King. Connection is the most critical and heavily weighted factor in the matchmaking process.
2. TIME TO MATCH – This factor is the second most critical to the matchmaking process. We all want to spend time playing the game rather than waiting for matches to start.
3. The following factors are also critical to the matchmaking process:
    - PLAYLIST DIVERSITY – The number of playlists available for players to choose from.

- RECENT MAPS/MODES – Considering maps you have recently played on as well as your mode preferences, editable in Quick Play settings.
- SKILL/PERFORMANCE – This is used to give our players – a global community with a wide skill range – the opportunity to have an impact in every match.
- INPUT DEVICE – Controller or mouse and keyboard.
- PLATFORM – The device (PC, Console) that you are playing on.
- VOICE CHAT – Enabled or disabled.

Connection quality and time to match are top priorities in *Call of Duty* matchmaking [1]. Skill is considered during the grouping of players to form a lobby and in team balancing at intermission. As discussed in depth below, skill targets are loosened faster than delta ping (lobby connection quality) targets when forming a lobby.

## Terminology

| | |
|---|---|
| Dedicated Server | A game host running in a data center. |
| Ping | The time taken for a network packet to make a round trip from the game client to the dedicated server. |
| Delta Ping | The difference between a player's lowest ping data center and their ping to any given other data center. |
| Party | A group of one or more players who have chosen to play together, treated by the matchmaker as an atomic group. |
| Lobby | A collection of parties that are in the process of being assembled to play a match, in the process of playing a match, or in the process of finishing a match. |
| Team | A partition of the lobby that is working together toward a shared objective and shared match outcome. Parties are typically kept intact within teams. |
| Raw Skill | A single value representing a player's performance relative to the rest of the player population. |
| Skill Percentile | A value which represents where in the population a player's raw skill lies. |

| | |
|---|---|
| Skill Disparity | The difference between the best and worst skilled player in a party or lobby. Typically, in the form of a skill percentile difference. |
| KPI | Key Performance Indicator. These are quantifiable metrics that measure performance against a specific objective. |
| TDM | Team Deathmatch, a multiplayer game mode that divides the lobby into two teams, and the team that scores the most kills wins the match. This is one of the most popular modes in the *Call of Duty* franchise. |
| KPM | Kills Per Minute. This KPI tracks the average number of kills a player achieves per minute during a match. |
| SPM | Score Per Minute. This KPI tracks the average score per minute players achieve during a match. In *Call of Duty*, a player's score is based on a combination of kills and completing match objectives. |

# What is Skill?

For the purposes of core multiplayer matchmaking, we generally define skill as how well a player can be expected to perform against the rest of the population in a given game mode, based on their previously observed performance. At a technical level we are interested in a value with the following properties:

1. It should be constrained between two numbers, otherwise it is difficult to reason about the space of all possible skill values, making analysis of the distribution more difficult.
2. It should be highly predictive: if we base your skill on a specific in-match performance metric (such as "kills per unit time"), it should also be a reliable predictor of your future performance as measured by this metric.
3. It should be summable such that the average skill of multiple players is predictive of their combined skill. This allows for very efficient and predictive team balancing. Team balancing is very important for forming games where the outcome is unpredictable. Blowouts result in players leaving the game which adversely affects the player pool. Team balance itself is covered in more detail later in the document.
4. It should be capable of adapting to a player's ever-changing performance quickly.
5. It should be resilient: the overall skill distribution should remain accurate in all situations. Simple skill algorithms can shift, inflate, deflate and even collapse when exposed to large population changes such as influxes of fresh players.

## How is Skill Calculated?

In *Call of Duty*, we calculate skill based on a player's relative performance on a specific metric. After each match, we compute this performance metric for each player. All players in the match are then compared to one another, regardless of team. Based on these comparisons each player's recorded skill value is then updated. The value of this skill adjustment is inversely proportional to the likelihood of a player achieving the outcome they did against the other players in the lobby. Note that the performance metric used only ever involves match performance; player progression or total time spent playing the game are not factored into skill.

This skill calculation involves several carefully selected parameters to achieve the five desired properties, referenced above.

Seemingly sufficient performance metrics can have large downsides, that we'll explore next. Let us evaluate some simple performance metrics and see their potential pitfalls when applied to TDM.

1. *Match Total Kills.* This value tells us how well a player did relative to the other opponents in their lobby at the main objective of the game. However, it has poor cardinality, as many players can achieve the same number of kills. This makes updating skill difficult, as many players will appear equally good based on this performance metric. It also does not reflect a player's ability to survive, which is an important outcome in *Call of Duty* as well. For example, a player with 10 kills and one death, is better than a player with 10 kills and 20 deaths.
2. *Kill / Death Ratio.* This value has much better cardinality, and it reflects both the primary and secondary objective of the game-mode. However, it does not account for self-kills, which is an easy mechanism of reverse boosting (artificial dropping your skill to get easier matches).
3. *Kills / (Deaths by enemy).* This value ensures players cannot artificially drop their skill by simply self-killing. However, a large problem remains; the magnitude of this value is the same for a player with 10 kills and one death regardless of if they played the full match or joined in the last minute.

We need to adjust for all the factors that contribute to or detract from a team's performance while being resilient towards gaming the system. To achieve this, we are constantly iterating on our performance metrics to optimize the player experience per game-mode.

## How Does Skill Change Over Time?

Player skill can vary over time for a variety of reasons. This might be because someone is experimenting with a new loadout, they haven't played recently, or they are simply tired or distracted. It is therefore important that a player's skill value is updated on an ongoing basis, and that it can be updated and reach equilibrium quickly. Overcorrection can lead to large fluctuations in the skill of players that someone is matched with and against and can result in unfair matches. However, when a player's abilities are stable, it is equally important that skill calculations find the stable midpoint quickly. These two goals, stability and rapid correction, are largely at odds. A balance must be found between the stability and flexibility that best suits each core multiplayer game mode in *Call of Duty*.

However, even if skill could be tracked perfectly and all matches were made with completely equal opponents, many players will still experience significant loss or win streaks. For instance, in any string of five perfectly equal games, the equivalent of a binomial distribution of a coin flipped five times, about 3% of players will experience a five-game loss streak, and about 3% will experience a five-game win streak.

# Why Even Track Skill?

One of the core design principles of *Call of Duty* is **Player First**. Players of all levels should have a fun and competitive experience with the game. Team balance is the first and most important reason to track skill. If we don't know how we expect players to perform in a match, then we can't provide a balanced in-match experience for players. This results in blowouts, which we know are not fun for players on the losing end. We have found that balancing skill against other matchmaking factors quantifiably increases the extent to which most players play and enjoy *Call of Duty*. When skill is utilized in matchmaking, 80-90% of players experience better end-of-match placement, stick with the game longer and quit matches less frequently.

All these factors strongly encourage the long-term health of the *Call of Duty* player base, helping the title avoid the feedback loop of low-to-average skill players continually leaving the game as the average skill of the population rises. By avoiding this feedback mechanism, the remaining 10-20% of the player population benefits. If low skill players engage with our titles less, then higher and higher skilled players become the new low skill players (relatively speaking). As a result, they then experience the negative outcomes of being the lowest skilled players in the core multiplayer population, likely resulting in those players then returning at reduced rates. This ultimately becomes a feedback loop, likely resulting in a player population of only the best of the best, and a very unwelcoming experience for any new players. As this would adversely impact the overall player pool, the net result would be a negative experience for all players.

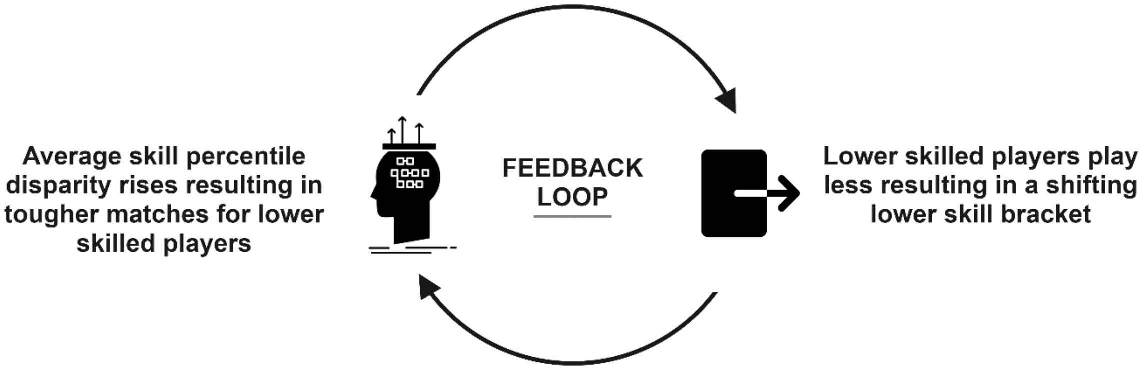

Average skill percentile disparity rises resulting in tougher matches for lower skilled players
FEEDBACK LOOP
Lower skilled players play less resulting in a shifting lower skill bracket

**Figure 1.**
*An illustration of the negative feedback loop of low skill players leaving the player base*

## Team Balance

Team balance is vital for ensuring games are fair for our players. The goal is to make the outcome of a match as unpredictable as possible. This reduces the probability of blowouts occurring, which are known to negatively correlate with self-reported “fun.” In the absence of team balance, larger parties end up with a significant advantage, where even a slightly above average party would be statistically likely to be above the average team sampled from the population. For instance, a six-player party who are all in the 60th skill percentile would be rated higher than approximately 80% of randomly sampled six-player teams.

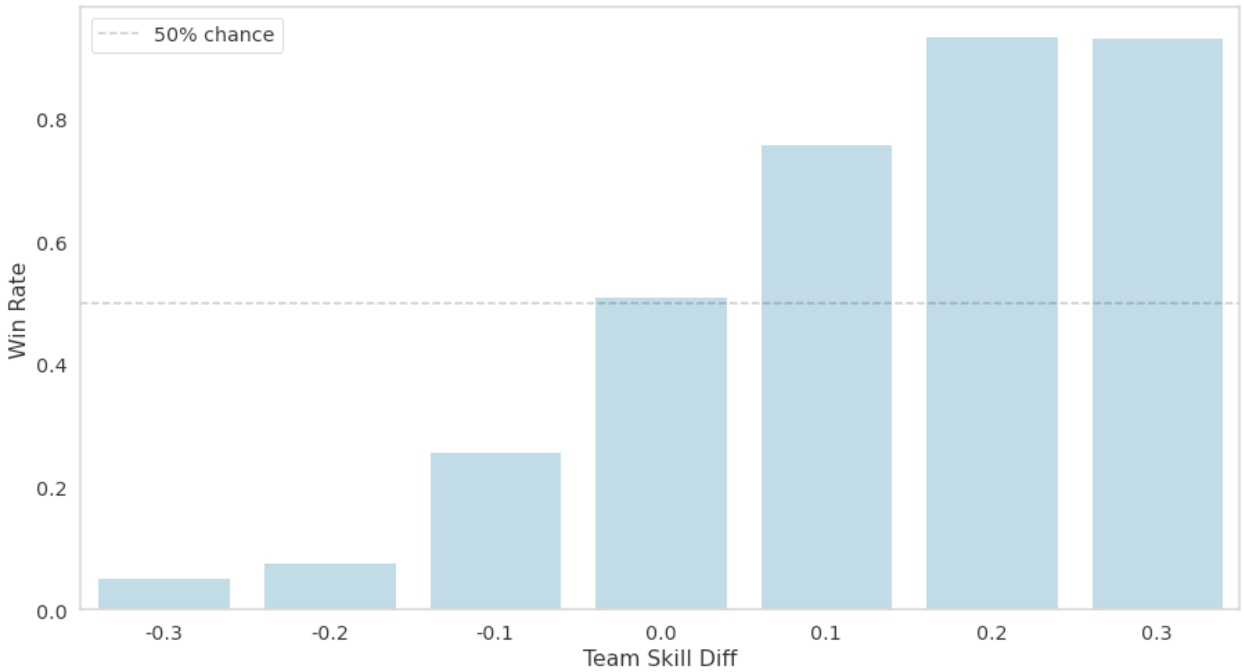


**Figure 2.**
*The observed win rate of a team in TDM given the differential between the sum skill of both teams. The X axis is in raw skill.*

In **Figure 2** we can see that win rates are significantly affected by small team skill differences. For instance, let's consider a lobby of 12 50th percentile players. If just one of those players was an 80th percentile player instead, corresponding to a 0.1 increase in raw team skill, that players team would have a 70+% chance of winning.

## What Impact Does Skill Have on The Player Experience?

We are always working to improve the quality of matchmaking in *Call of Duty* and rely on data driven approaches to evaluate our success. There are two primary methods by which we've come to understand the impact of skill in matchmaking:

1. Testing of different skill matching approaches.
2. Comparing match outcomes between titles in the *Call of Duty* franchise that have different skill implementations.

When discussing this, we talk about tightening and loosening our skill constraints. This is adjusting two parameters in our system:

1. Allowing for the average skill of a party being added to a lobby to be farther from the average skill of the lobby (loosening) or requiring it to be closer (tightening).

2. Allowing for a lobby's percentile skill disparity to drift further as a result of adding a new party (loosening) or restricting how far we let this drift (tightening).

For more details on how these two dimensions work see How Is Skill Incorporated into Matchmaking? section below.

## Testing of Different Skill Matching Approaches

We continually run tests on various parts of the matchmaking system to find optimal configurations to improve fun while maintaining efficient matchmaking. While there is no direct measure for ‘fun’, we use data that indicates that players are enjoying the game, such as how long they continue to play the game, match-level quit rates, player surveys and match outcomes.

Call of Duty is a game where players can play together in parties. In some experimentation methodologies, this results in some mixing of the cohorts and the analysis of results can be complex. Testing matchmaking at this scale is a very interesting subject, independent of the discussion of skill. This is a topic we will discuss further in a future white paper focusing on experimentation methods.

As an example, in early 2024, we ran the Deprioritize Skill Test in Call of Duty®: *Modern Warfare® III*, where we used our A/B test framework to loosen the constraints on skill in matchmaking. It’s important to note that skill, as a factor in matchmaking, was decreased for this test, but not removed entirely from the matching algorithm. Based on our history of testing, completely removing skill from matchmaking would amplify the observed effects. This experiment is a repeat of a type of test that we have run at various times throughout the last five years. We ran the 2024 test in North America and established a treatment group of 50% of the population. For the treatment group we loosened the skill constraints. The other half of the population was left with the standard configuration.

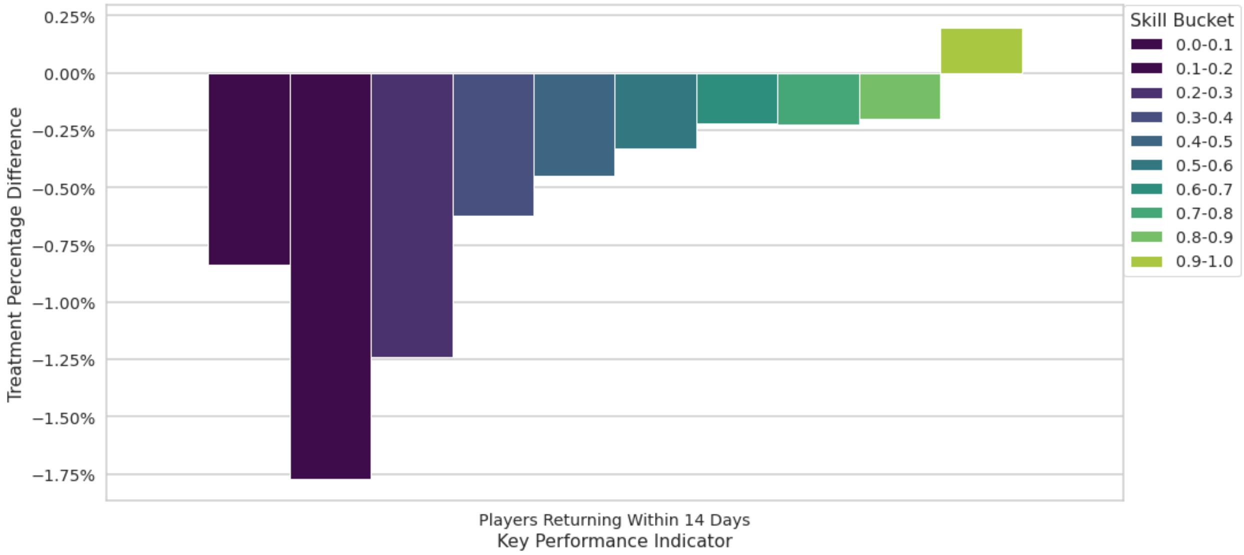


**Figure 3.**
*Difference in players returning within 14 days during the Deprioritize Skill Test*

In **Figure 3** we can see one of the results of the Deprioritize Skill Test. After a month of running this test, we categorized the treatment population into 10 equally sized groups across the skill population. Each bar represents the change in the labeled KPI for that 10th of the skill distribution compared to the control group. The skill distribution is determined using our internal skill algorithm that tracks how good we believe a player to be, as described above. As player skill is always fluctuating, we take the average of the skill values each user had during the test, then calculate the percentiles from these averaged values. For example, a player represented in the top 10% group in **Figure 3**, had an average skill value in the top 10% of all players seen during the experiment.

In **Figure 3** we can observe the percent difference in the number of players returning after 14 days between the treatment and control groups. With deprioritized skill, returning player rate was down significantly for 90% of players. The 10% of highest skilled players came back in increased numbers, but in aggregate, we see meaningfully fewer players coming back to the game. This effect may appear small, but this change was observable within the duration of the test. This will compound over time, just like interest, and will have a meaningful impact on our player population. This is a concern for all players, including the top 10%, as if this pattern is allowed to continue, players will exit the game in increased numbers. Eventually a top 10% player will become a top 20% player, and eventually a top 30% player, until only the very best players remain playing the game. Those original top players will become increasingly likely to not return to the game. Ultimately, this will result in a worse experience for all players, as there will be fewer and fewer players available to play with. Also, as noted above, this test only deprioritized skill in the matching rules. If it were completely removed, we would expect to see the player population erode rapidly in the span of a few months, resulting in a negative outcome for all our players.

We have also run experiments to tighten skill beyond our current configuration. This had inverse results, negatively impacting the high skill cohort. This change was not rolled out as a standard approach, as we continue to strive for a balance in our approach to matchmaking. We provide more detail on this test in our discussion of historical testing.

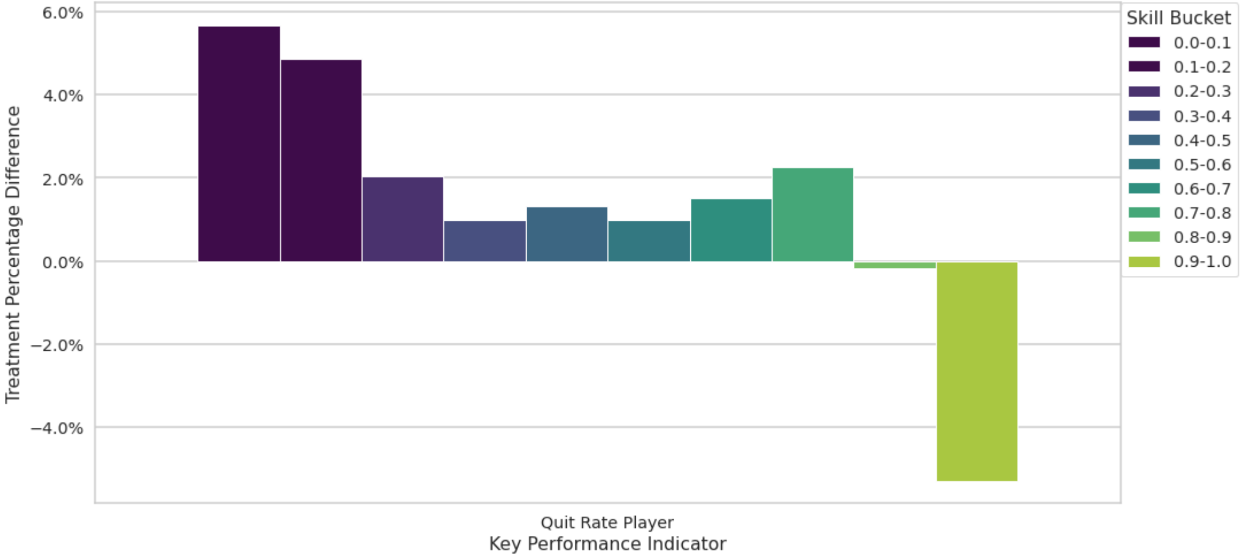


**Figure 4.**
*Difference in Quit Rate from the Deprioritized Skill Test*

Quit rate is the likelihood for a player to quit throughout a match. In **Figure 4,** we observe that the quit rate significantly increases across 80% of players, and only the top 10% see a meaningful decrease in quit rates. We have historically found that quit rates have a strong negative correlation with self-reported "fun" gathered through player surveys. This will be a short-term benefit for the top 10% of players, however. As the accelerated departure of players in the lower skill brackets takes hold, top 10% players will eventually drift down the skill distribution (as originally top 10% players will make up a larger and larger portion of the player base). As a result, we expect to see once top 10% players quit games at increasing rates as they become a 50th percentile player after much of the lower skill population has left the game.

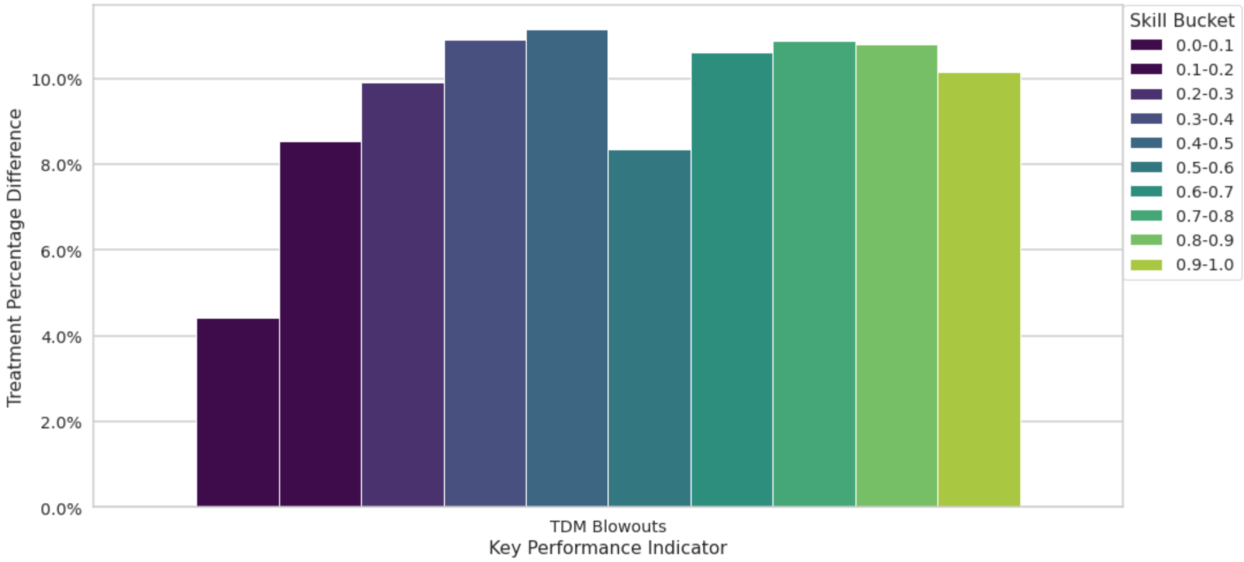


**Figure 5.**

*Difference in TDM Blowouts from the Deprioritized Skill Test*

In **Figure 5** we see the difference in the rate of blowouts occurring in TDM. A blowout is when a team in a lobby wins with a score delta greater than 30. This has increased for all players and has also been established as having a negative correlation with self-reported "fun." We see similar results in other game modes.

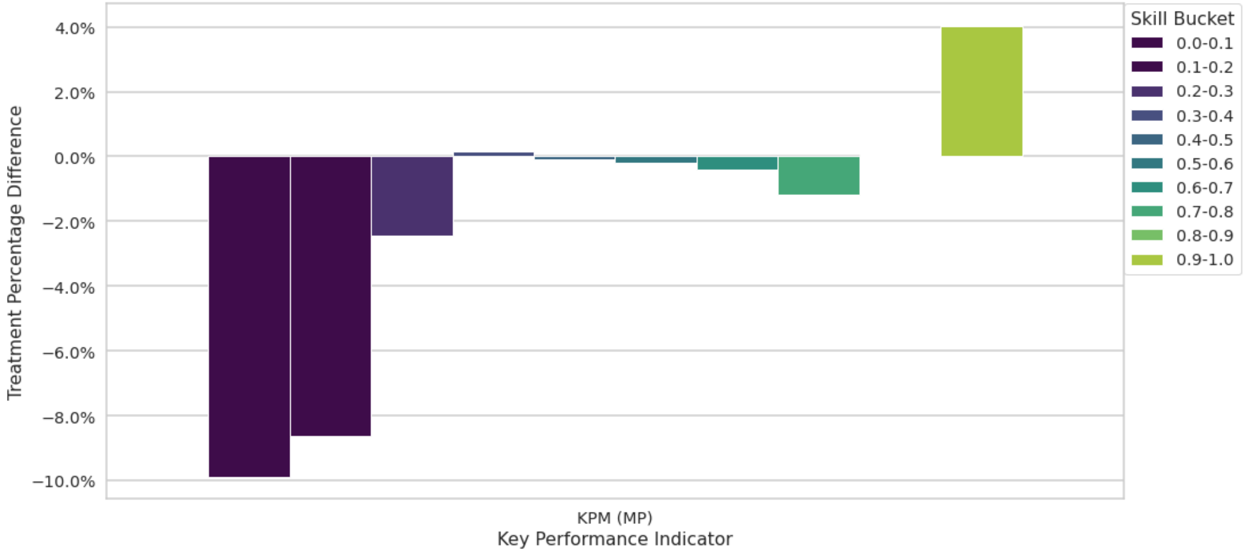


**Figure 6.**

*Difference in rate of Kill Per Minute (KPM) from the Deprioritized Skill Test*

Kills Per Minute is down significantly for the bottom 20-30% of players. The next 60% of players have no significant change, and the top 10% see significantly higher KPM. As with the other KPIs, the accelerated rate of low-skill players not returning to the game will result in players shifting to the left on this distribution over time.

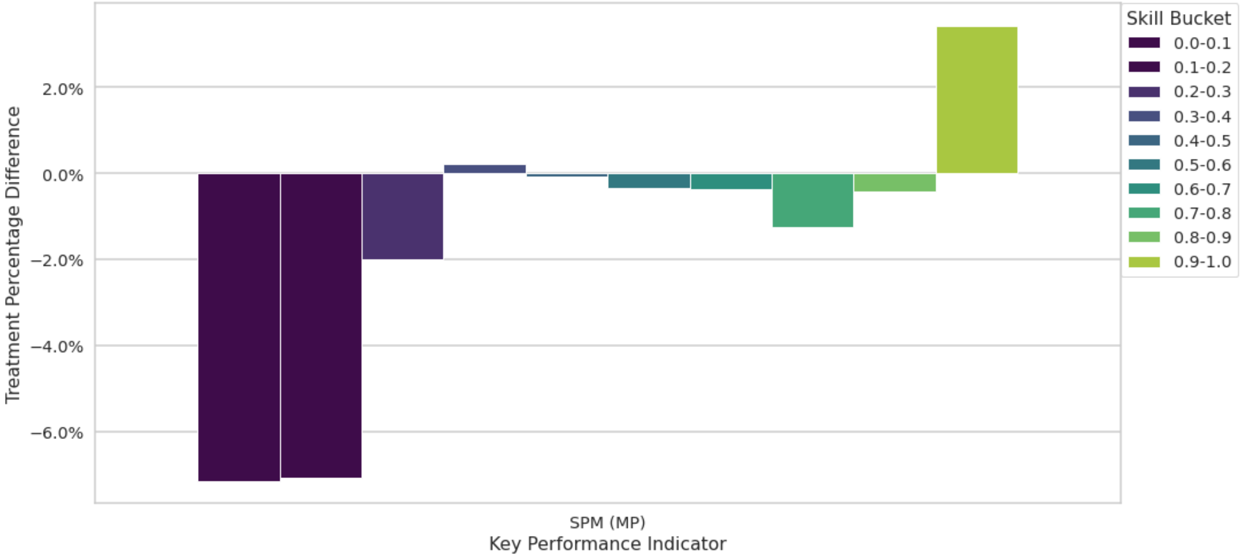


**Figure 7.**
*Difference in rate of Score Per Minute (SPM) from the Deprioritized Skill Test*

Like KPM, SPM follows a similar trend. The low-skill players perform worse, while the top 10% can dominate. As with KPM, we expect to see players shift to the left on this distribution over time, as low-skill players return to the game at lower rates.

The use of killstreaks and increased KPM and SPM shows that the wider lobby skill percentile disparity is disproportionality leveraged by the top 10% of players. Unfortunately, this increased performance comes at the cost of much greater impact to the much larger 30% of the population toward the bottom of the skill distribution.

## Comparing Match Outcomes Between Different Call of Duty Titles

Our other opportunity to measure the impact of using skill as a factor in matchmaking is from one game to another in *Call of Duty*. There's variability in core multiplayer skill tightness/looseness across titles in the franchise, because they tune for skill differently relative to the other matching criteria. We compare across games by amalgamating player match outcomes between two different games with different approaches to skill: one tighter, one looser. Match outcome is a broad metric encompassing many factors:

- leaderboard placement, regardless of team
- interactions with game systems, like killstreaks, and
- interactions with the objective, such as hardpoints

We can then look at match outcome differences between two games, across the skill distribution.

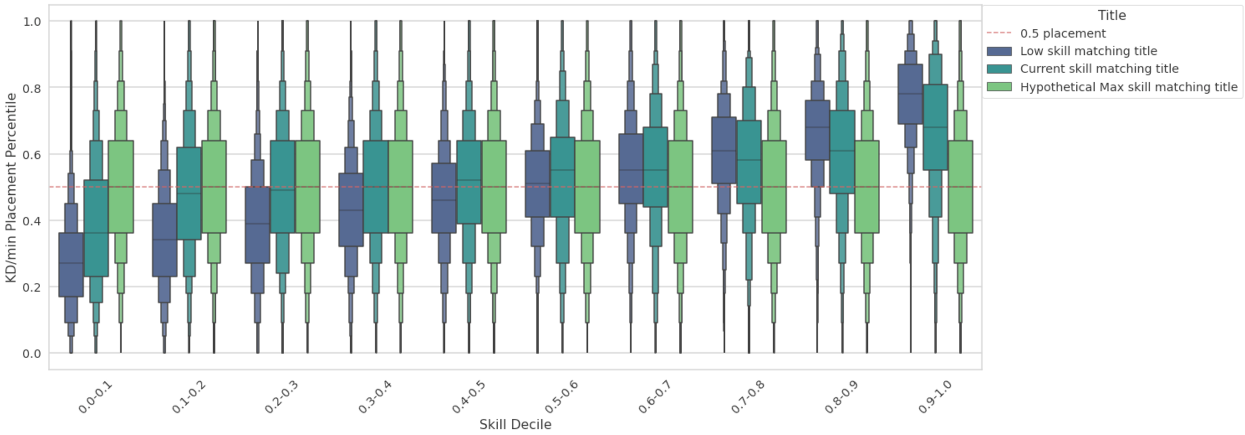


**Figure 8.**
*Letter-value plot of the observed distributions of KD/minute placement percentiles across the skill distribution between a Call of Duty title with low skill matching and current skill matching. This also includes a reference of what a hypothetical max skill matching across the distribution would look like.*

In **Figure 8** we can observe the effect of skill grouping on the achieved KD/Minute placement percentile. The range of potential outcomes an individual player achieves is widened in the title with tighter skill. In the title with low skill matching, a bottom decile player will place in the bottom half of a TDM match close to 90% of the time. In the title with tighter skill, a bottom decile player only experiences this about 75% of the time. Using skill in matchmaking does not necessarily flatten the outcome graph, it reduces the severity of the slope. A completely flat outcome is included as reference. Even with more consideration for skill in matchmaking, higher skill players perform better than lower skill players by a significant margin, and still perform far better than they would if skill disparity was the top priority of the *Call of Duty* matchmaking system, which it is not.

## Other Historical Testing

The above are just two examples of ways we can see the impact of skill on *Call of Duty* matchmaking. Skill as a consideration has been a factor in matchmaking for *Call of Duty* from as early as *Call of Duty 4: Modern Warfare.* In the early years of the franchise our ability to formally experiment was limited, and so we iterated game by game on our matchmaking approach. Since the release of *Call of Duty: Modern Warfare (2019)* our testing capabilities have improved substantially. We are now able to run experiments with modern testing methodologies, which we will explore in an upcoming entry in our white paper series, later this year (target timing may shift).

We can see that loosening skill negatively impacts our ability to keep players interested in our game. In a test similar to the Deprioritized Skill Test discussed above, we were able to see a significant decrease in the number of players playing *Call of Duty: Modern Warfare (2019)* and an increase in the overall match quit rate, when treated with a looser skill matching. Subsequent attempts to protect only the bottom 25% of players and allow for looser matchmaking for the remaining 75% of players also had clear negative effects on player counts in two weeks, with increased quit rates, and reductions in total hours played. Both of which are well established as negative indicators of self-reported "fun."

Another example was a test to tighten skill in *Call of Duty: Modern Warfare III*. This had inverse results consistent with the results of the loosening test. Quit rate was down for 90% of players and we saw other improvements in the experience of low-skill players (KPM and SPM). However, we observed negative impacts for high-skill players. As a result, this change was not rolled out as a standard approach in *Call of Duty: Modern Warfare III*, as we continue to strive for a balance in our approach to matchmaking.

Our goal has always been to make *Call of Duty* as enjoyable for as many players as we can, and we'll continue to experiment with how we can provide a better experience for all our players.

# How is Skill Incorporated into Matchmaking?

Matchmaking targets are loosened over time in a pattern. We call these loosening patterns backoffs. As a search ages, the system becomes more willing to accept looser restrictions across all dimensions, as the absence of a match over time is an indicator that not enough players are available to form a match with the current targets. The rate of these backoffs and the volume of available searches determines time to match. We have always backed off on skill more quickly than other matchmaking dimensions like Delta Ping, as outlined in our first white paper [1]. Exactly how much is dependent on the game mode and game type. Below is a detailed description of how skill is used in the matchmaking algorithm.

## Skill Percentiles

We refer to the skill values used during team balancing as a player's *Raw Skill*. Raw skill is a normal distribution between -1.0 and 1.0, but for the purpose of skill grouping we would rather have a normalized uniform value. This can be achieved by converting raw skill into a percentile. A system constantly tracks population skill values, and we convert each player's skill to a corresponding skill percentile.

The benefits of a skill percentile are that by default any matchmaking rules based on these values apply to all players equally, e.g. the bottom 30% and top 30% of the skill population get similar matchmaking times. The downside of skill percentiles is that they are less indicative of a player's skill level, so raw skill is used for team balancing.

## Skill Grouping

Skill grouping is a key factor to matchmaking with subtle differences across all our game modes. The goal of skill grouping is to keep similarly skilled players together and to find optimal opponents for parties that have large skill differences in the best and worst players. *Call of Duty* imposes no restrictions on how wide this skill gap can be for parties outside of ranked play and thus the best and worst players in the world can group up and our objective is to deliver a fun, fair match.

We have three overlapping systems that attempt to optimize skill grouping: A heuristic selection process, a skill grouping rule, and a skill disparity minimization rule. The combination of these systems achieves the intended goal. The skill similarity rule aims to keep the effective skill of parties in the lobby similar and the skill disparity rule tries to group parties with similar skill disparity.

### Heuristic Selection Process

This system aims to optimize the order in which candidates around a matchmaking player are selected during the matchmaking process.

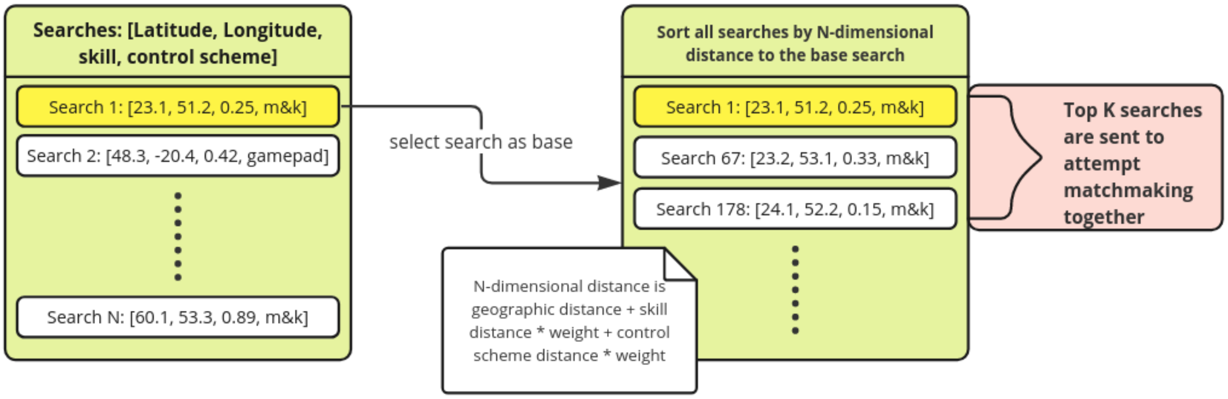

Searches: [Latitude, Longitude, skill, control scheme]
Search 1: [23.1, 51.2, 0.25, m&k]
Search 2: [48.3, -20.4, 0.42, gamepad]
Search N: [60.1, 53.3, 0.89, m&k]
select search as base
Sort all searches by N-dimensional distance to the base search
Search 1: [23.1, 51.2, 0.25, m&k]
Search 67: [23.2, 53.1, 0.33, m&k]
Search 178: [24.1, 52.2, 0.15, m&k]
N-dimensional distance is geographic distance + skill distance * weight + control scheme distance * weight
Top K searches are sent to attempt matchmaking together

**Figure 9.**

*Diagram of heuristic selection of candidates for a single search*

Every five seconds the system attempts to match all players searching. This starts by iterating over each search and selecting a subset of other searches that are likely candidates for lobby formation. In **Figure 9**, we can see how this process works. Each search is categorized by their geolocation (which is a proxy for similar DC ping), skill, and control scheme; these factors make up a player's N-dimensional location. We then sort the list of available candidates by N-dimensional distance which is computed as follows:

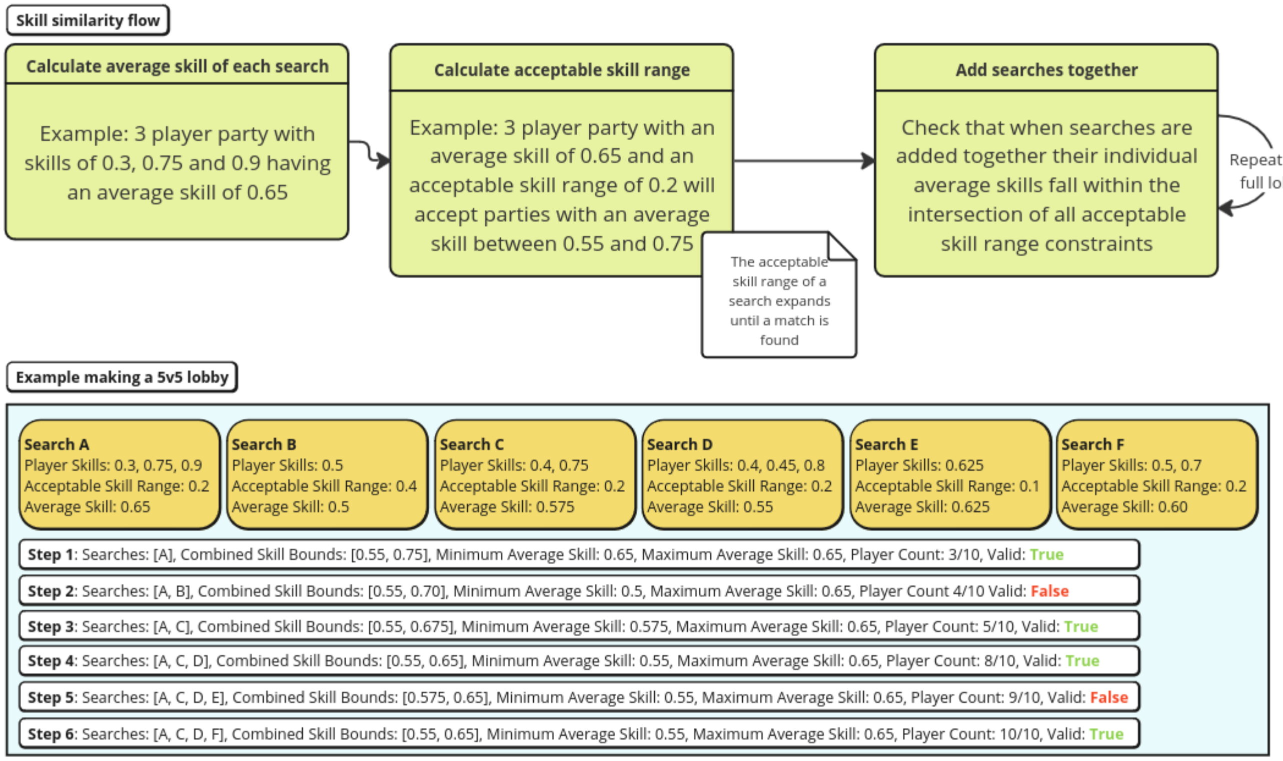
1. Player geolocation is stored as latitude and longitude. We use great circle approximation to find the geographic distance between two searches.
2. Skill is an additional dimension between 0.0 and 1.0, representing the search's average skill percentile. The skill distance is simply the difference between two searches' average skill percentile, which is then multiplied by a weight to align it to geographical distance. Skill is slightly weighted such that when multiple candidates are similarly close geographically, we will consider those with similar skill as a next step.
3. Control scheme is the final dimension which simply adds a set geographical distance penalty for control scheme similarity.
4. All three distances are added to get the final distance between two candidates.

The top $K$ candidates sorted by this distance are then selected to be sent on to the matchmaker to try to form a lobby together. $K$ is a specific value unique to each game-mode tuned to strike a balance between computational efficiency and optimal matchmaking.

This process is necessary for efficiently finding groups of searches that can likely form a lobby together. Take for instance a group of 300 players, there are over 887,827,414,757,477,464,725 unique 12 player lobby configurations possible. Exhaustively finding the best amongst these is computationally impossible on the time scale of a single search. Therefore, we must rely on heuristics to order and prune the list of candidates such that each sequential search considered is the most likely to lead to a near optimal result.

## Skill Similarity Rules

These matchmaking rules attempt to minimize the difference in the average skill of parties in a lobby. In effect this acts to ensure the skill distribution in a lobby is roughly centered on the average skill of the lobby. As with delta ping this is a constraint that is loosened the longer a player's search runs. The amount of skill similarity a search will accept is also modified by other factors such as which game mode is selected, if there is a high-quality lobby open for joins, and how many players are playing in a specific region.


Skill similarity flow
Calculate average skill of each search
Example: 3 player party with skills of 0.3, 0.75 and 0.9 having an average skill of 0.65
Calculate acceptable skill range
Example: 3 player party with an average skill of 0.65 and an acceptable skill range of 0.2 will accept parties with an average skill between 0.55 and 0.75
The acceptable skill range of a search expands until a match is found
Add searches together
Check that when searches are added together their individual average skills fall within the intersection of all acceptable skill range constraints
Repeat until full lobby
Example making a 5v5 lobby

**Search A**
Player Skills: 0.3, 0.75, 0.9
Acceptable Skill Range: 0.2
Average Skill: 0.65

**Search B**
Player Skills: 0.5
Acceptable Skill Range: 0.4
Average Skill: 0.5

**Search C**
Player Skills: 0.4, 0.75
Acceptable Skill Range: 0.2
Average Skill: 0.575

**Search D**
Player Skills: 0.4, 0.45, 0.8
Acceptable Skill Range: 0.2
Average Skill: 0.55

**Search E**
Player Skills: 0.625
Acceptable Skill Range: 0.1
Average Skill: 0.625

**Search F**
Player Skills: 0.5, 0.7
Acceptable Skill Range: 0.2
Average Skill: 0.60

**Step 1**: Searches: [A], Combined Skill Bounds: [0.55, 0.75], Minimum Average Skill: 0.65, Maximum Average Skill: 0.65, Player Count: 3/10, Valid: True

**Step 2**: Searches: [A, B], Combined Skill Bounds: [0.55, 0.70], Minimum Average Skill: 0.5, Maximum Average Skill: 0.65, Player Count 4/10 Valid: False

**Step 3**: Searches: [A, C], Combined Skill Bounds: [0.55, 0.675], Minimum Average Skill: 0.575, Maximum Average Skill: 0.65, Player Count: 5/10, Valid: True

**Step 4**: Searches: [A, C, D], Combined Skill Bounds: [0.55, 0.65], Minimum Average Skill: 0.55, Maximum Average Skill: 0.65, Player Count: 8/10, Valid: True

**Step 5**: Searches: [A, C, D, E], Combined Skill Bounds: [0.575, 0.65], Minimum Average Skill: 0.55, Maximum Average Skill: 0.65, Player Count: 9/10, Valid: False

**Step 6**: Searches: [A, C, D, F], Combined Skill Bounds: [0.55, 0.65], Minimum Average Skill: 0.55, Maximum Average Skill: 0.65, Player Count: 10/10, Valid: True

**Figure 10.**
*Skill similarity rule flow and example of 5v5 lobby being formed*

In **Figure 10** we can see the flow of the skill similarity rule. Each search has its own average skill value, and a skill range centered on itself which constrains who the search is willing to match with.

In the example we see the process of forming a 5v5 lobby with various party searches and differing acceptable skill ranges.

1. At step 1, Search A at 0.65 skill and a 0.3 skill range will accept other searches between 0.55-0.75 skill.
2. At step 2 we attempt to add Search B at 0.5 skill and a 0.4 skill range. The intersection of both skill ranges is [0.55, 0.70]. Search B sits below the range accepted by Search A and is therefore invalid.
3. At step 3 we attempt to add Search C at 0.575 skill. The intersection of both skill ranges is [0.55, 0.675]. Both searches sit within this range and therefore Search C can be added.
4. At step 4 we attempt to add Search D at 0.55 skill. The intersection of all skill ranges is [0.55, 0.65]. All searches sit within this range and therefore Search D can be added.

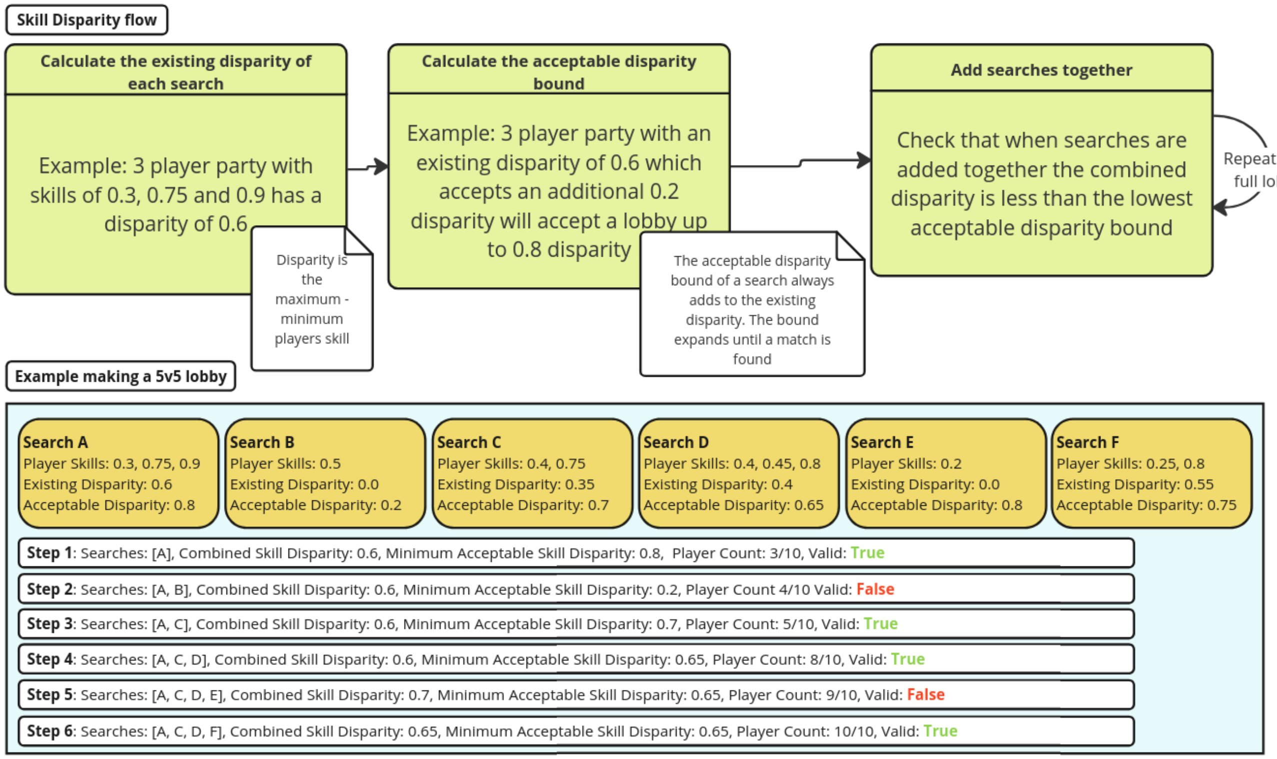
5. At step 5 we attempt to add Search E at 0.625 skill. The intersection of all skill ranges is [0.575, 0.65]. Search E has too restrictive of a skill range and will not accept Search D and is therefore invalid.
6. At step 6 we attempt to add Search F at 0.6 skill. The intersection of all skill ranges is [0.55, 0.65]. All searches sit within this range and therefore Search F can be added. With the addition of this search all 10 player slots are filled and the lobby is ready to be formed.

These rules exist to primarily account for parties and to aid team balancing. Parties with high disparity are difficult to match fairly, take for instance a party of two players, **Alice** and **Bob**, **Bob** is an average player with 50% skill percentile and **Alice** is an elite player with 99% skill percentile. If we matchmake them with **Bob's** skill, **Alice** is practically guaranteed to be the best player in every lobby they join, more so than if she played solo. If we matchmake on **Alice's** skill, then **Bob** will likely be the worst player in every lobby they join. Thus, we must match them in the middle such that the worst player gets some opponents of equal footing, while minimizing the inherent advantage of the better player.

### Skill Disparity Rules

The skill disparity rules are concerned with minimizing the difference between the worst and the best player in a lobby. This rule works in tandem with the skill similarity rule to group parties with high disparity together where possible. As discussed above, parties with high skill disparity are inherently difficult to matchmake fairly, thus the more we can group similarly disparate parties together the less of an effect they have on the less disparate population.

The skill disparity rules loosen over time using the same mechanism as skill similarity and delta ping. Even though we can track and predict how long it may take to form a desirable match, this prediction can be off when fewer players search than expected. When this happens, looser constraints aid in the formation of a match in a reasonable length of time.



Skill Disparity flow

Calculate the existing disparity of each search

Example: 3 player party with skills of 0.3, 0.75 and 0.9 has a disparity of 0.6

Disparity is the maximum - minimum players skill

Calculate the acceptable disparity bound

Example: 3 player party with an existing disparity of 0.6 which accepts an additional 0.2 disparity will accept a lobby up to 0.8 disparity

The acceptable disparity bound of a search always adds to the existing disparity. The bound expands until a match is found

Add searches together

Check that when searches are added together the combined disparity is less than the lowest acceptable disparity bound

Repeat until full lobby

Example making a 5v5 lobby

**Search A**
Player Skills: 0.3, 0.75, 0.9
Existing Disparity: 0.6
Acceptable Disparity: 0.8

**Search B**
Player Skills: 0.5
Existing Disparity: 0.0
Acceptable Disparity: 0.2

**Search C**
Player Skills: 0.4, 0.75
Existing Disparity: 0.35
Acceptable Disparity: 0.7

**Search D**
Player Skills: 0.4, 0.45, 0.8
Existing Disparity: 0.4
Acceptable Disparity: 0.65

**Search E**
Player Skills: 0.2
Existing Disparity: 0.0
Acceptable Disparity: 0.8

**Search F**
Player Skills: 0.25, 0.8
Existing Disparity: 0.55
Acceptable Disparity: 0.75

**Step 1**: Searches: [A], Combined Skill Disparity: 0.6, Minimum Acceptable Skill Disparity: 0.8,  Player Count: 3/10, Valid: True

**Step 2**: Searches: [A, B], Combined Skill Disparity: 0.6, Minimum Acceptable Skill Disparity: 0.2, Player Count 4/10 Valid: False

**Step 3**: Searches: [A, C], Combined Skill Disparity: 0.6, Minimum Acceptable Skill Disparity: 0.7, Player Count: 5/10, Valid: True

**Step 4**: Searches: [A, C, D], Combined Skill Disparity: 0.6, Minimum Acceptable Skill Disparity: 0.65, Player Count: 8/10, Valid: True

**Step 5**: Searches: [A, C, D, E], Combined Skill Disparity: 0.7, Minimum Acceptable Skill Disparity: 0.65, Player Count: 9/10, Valid: False

**Step 6**: Searches: [A, C, D, F], Combined Skill Disparity: 0.65, Minimum Acceptable Skill Disparity: 0.65, Player Count: 10/10, Valid: True

**Figure 11.**
*Skill disparity rule flow and example of 5v5 lobby being formed*

In **Figure 11** we can see the flow of the skill disparity rule. Each search has its own skill disparity and an acceptable skill disparity bound which constrains who the search is willing to match with.

In the example we see the process of forming a 5v5 lobby with various party searches and differing skill disparities and acceptable disparity bounds.

1. At step 1, Search A has 0.6 disparity and will accept up to 0.8.
2. At step 2, we attempt to add Search B which will only accept 0.2 disparity. This is lower than the existing disparity of 0.6 and therefore the search isn't added.
3. At step 3, we attempt to add Search C which will accept up to 0.7 disparity. This is a reduced acceptable disparity relative to Search A but still higher than the 0.6 disparity of both searches combined and therefore Search C can be added.
4. At step 4, we attempt to add Search D which will accept up to 0.65 disparity. Similar to step 3, this reduced acceptable disparity is still less than the combined disparity of 0.6 and therefore Search D can be added.
5. At step 5, we attempt to add Search E which will accept up to 0.8 disparity. This search has a lower skill than any previous players added, and the actual disparity increases

to 0.7. This is higher than the minimum acceptable skill disparity of 0.65 and therefore the search is not added.

6. At step 6, we attempt to add Search F which will accept up to 0.75 disparity. This search also includes a low-skill player which increases the combined disparity to 0.65. This is just within the minimum acceptable skill disparity of 0.65 and therefore Search F can be added.

In the above example we can see how the skill disparity rule stops some searches from being included in a forming lobby with relatively high disparity. Search B would likely pass the skill similarity rule with the same searches, but it has a low existing disparity and has not been searching long so we can likely find it a tighter game. Search E has been searching for a long time, but adding it would exceed the bounds of the other searches already added. Again, there is a high likelihood that despite having very wide acceptable bounds, Search E could be added to a more appropriate game centered on its own skill, which the skill similarity rule will help enforce.

## Team Balance

Team balance is a multistep process, where each step is an NP-HARD problem and ideally contributes to the final goal of a balanced match while also avoiding biases against individual players.

### Grouping Phase

The grouping phase occurs whenever we are forming a new lobby or backfilling an existing one. During this phase we are pursuing three goals:

1. Prevent the formation of matches impossible to balance.
2. Prevent the formation of imbalanced incomplete matches.
3. Backfills never increase an existing team imbalance.

This problem is a variant of the k-partitioning problem [3]. For any prospective new lobby or backfill we are trying to find that there is a least one solution which satisfies the k-partitioning problem where the number of players is lower than or equal to the maximum team size and k is the number of teams in the game-mode.

Let's look at an example using the following format. A party of *N* players is denoted as {*N*}. The team balance process is represented using the =>. A team comprised of multiple parties is denoted with square brackets.

Example in a 6v6 game-mode:

- Searches: {3}, {2}, {2}, {1} => [{3}, {1}] vs [{2}, {2}]
  - → This is a valid team balance.
- Searches: {4}, {4}, {3} => [{4}, {3}] vs [{4}]
  - → There exists no way to team balance these players without creating a team greater than the maximum team size.

For game modes with two teams, we are using the Karmarkar-Karp heuristic [4] to get the potentially best team size differential of a set of searches. This heuristic is fast to compute and is guaranteed to find a result that satisfies goal (1) above.

However, knowing a set of searches is balanceable is not enough, for the purpose of goals (2) and (3) we also want to limit the team size differentials within a lobby. New lobbies are always created with a team size differential of zero, even when we make a lobby that is not completely filled. Backfills will accept a search if the existing team size differential is not worsened.

Example in a new lobby for a 6v6 game-mode:

- Searches: {3}, {2}, {2}, {2}, {1} => [{3}, {2}] vs [{1}, {2}, {2}]
  - → This is a valid team balance.
- Searches: {3}, {2}, {2}, {2}, {1}, {1} => [{3}, {2}, {1}] vs [{1}, {2}, {2}]
  - → This is an invalid team balance as there is a team differential. New lobbies are never created with team size differences.

The algorithms to enforce these goals need to run incredibly quickly, being executed thousands of times per second. During this phase, it is only computationally feasible to determine whether the teams in a lobby will be balanceable, the exact team compositions are only calculated when the lobby is first formed. The implication of this is that we must try to minimize team skill differentials by selecting candidates during the grouping phase that will be readily balanceable down the line. The skill component of the heuristic in tandem with the skill grouping rules aid the likelihood of closer team balance.

### Lobby Phase

Once a lobby has been formed the exact team composition can be computed. This occurs in two steps.

1. For modes up to 12v12 we do a fully exhaustive search to find every possible team composition. This list is pruned to the team compositions that have the lowest difference in size between the two teams.
2. The team composition with the smallest sum skill differential between the teams is then selected from the pruned list.

Many team configurations are balanceable but do not allow a lot of flexibility to shuffle players around. The most obvious case for this is matchmaking a six-player party in a 6v6 player mode. Without incorporating skill at the matchmaking phase there is no guarantee that a formed lobby including a team sized party can be balanced effectively. Similar situations can easily arise with smaller parties as well; two three-player parties and three two-player parties can only be matchmade such that the two three-player parties are on the same team.

## Ranked Play

Skill is not isolated as a factor in matchmaking for Ranked Play chiefly due to game design. Ranked Play is designed to deliver an expressly competitive environment; accordingly, players must qualify for access to Ranked Play modes. Many players who have qualified for Ranked Play still choose to enter the game in non-ranked playlists. For new players and those who do not participate in Ranked Play, it’s important they can contribute meaningfully to their team and their own personal in-game achievements. The next Matchmaking Series white paper will further detail Ranked Play.

## How Does Skill Impact Other Matchmaking KPIs Across the Skill Spectrum?

One of the goals of our system is to give everyone a relatively similar matchmaking experience as mentioned in the introduction; a fair shot at achieving and experiencing the range of outcomes and events in *Call of Duty*. However, the population is highly asymmetric, with most parties, particularly disparate ones, sitting higher in the skill distribution. The practical result of this is that matchmaking at the higher skill level requires more population to form equivalently equitable matches. Note that in the previous white paper of the Matchmaking Series, discussing the role of Ping, we stated that skill level has no impact on the latency experience [1]. This was an oversimplification and should be clarified. Skill level has a small impact on matchmaking outcomes, including Delta Ping and search time, but it is minor and not strictly linear. Search time peaks around the 7th decile, but as illustrated in **Figure 13** absolute ping is consistent across the skill distribution and slightly decreases for higher skill players.

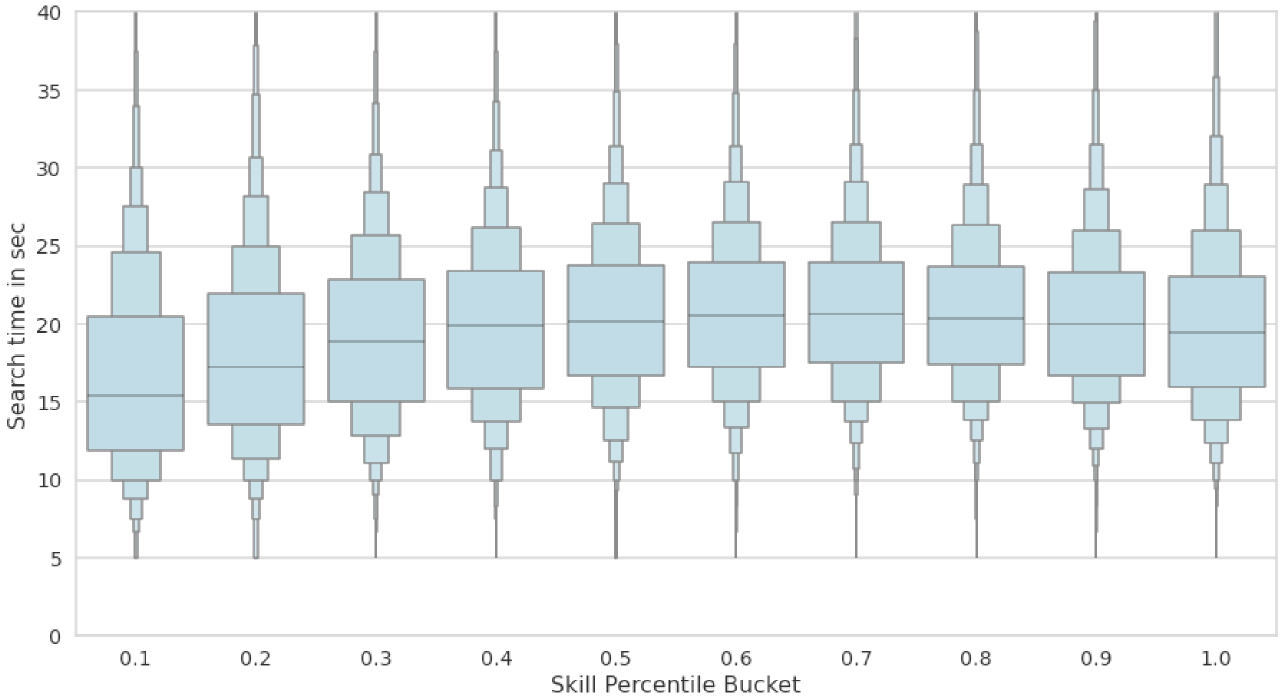


**Figure 12.**
*Letter-value plot of time spent search per match across the skill spectrum*

In **Figure 12** we can observe a relatively similar matchmaking search time across the skill spectrum. Note that search time strongly correlates with Delta Ping and skill disparity. There is a slight upward trend in the data that exists because of the distribution of parties within the player population. Higher skill players are more likely to play in parties which take longer to matchmake optimally.

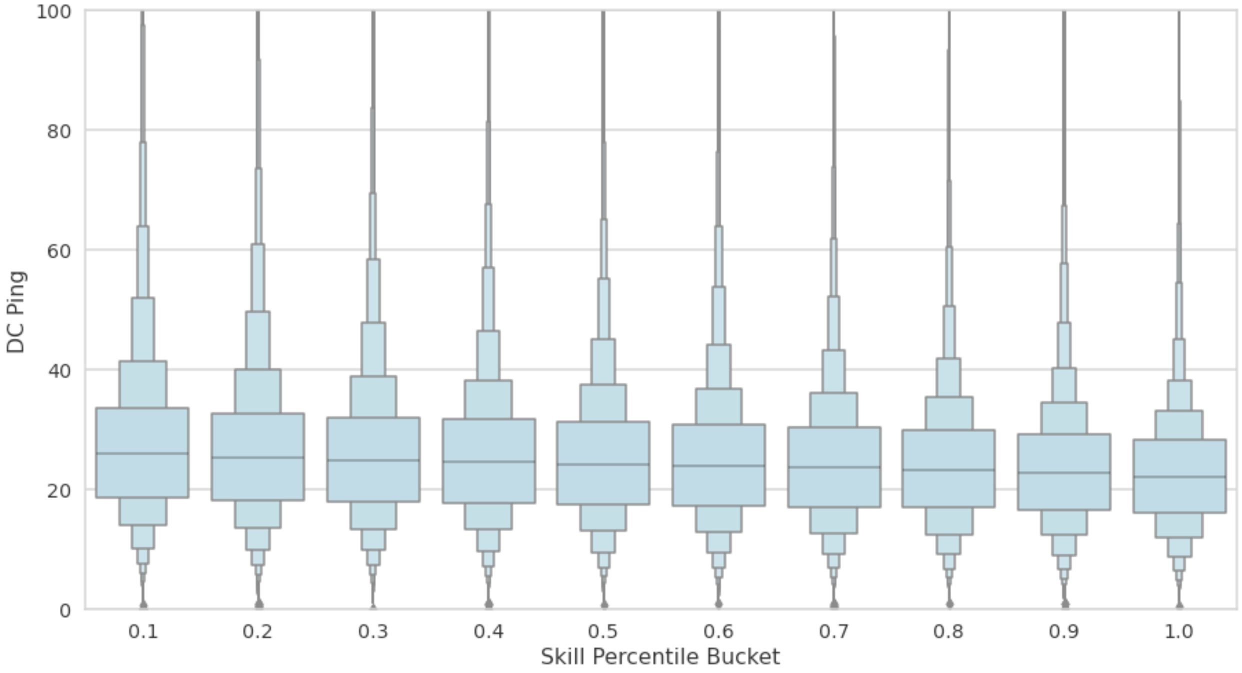


**Figure 13.**
*Letter-value plot of absolute ping across the skill spectrum*

**Figure 13** outlines the distribution of absolute ping across the skill distribution, as measured by pre-matchmaking QoS. The key observation here is that despite the distribution of search times, the absolute latency experience is consistent across the skill distribution and slightly decreasing with higher skill.

The rules of the *Call of Duty* core multiplayer matchmaking system are applied consistently across our entire skill distribution to provide as fun and fair an experience as possible. While this approach has been shown to support the long-term quality of our players' experience, we are always looking for ways to improve and we will continue to experiment in this area.

# References

[1] Activision Publishing, Inc. (2024, April 4). *Matchmaking Series: Ping*. Activision Research. https://research.activision.com/content/dam/atvi/activision/atvi-touchui/research/publications/docs/Call-of-Duty-Matchmaking-Series-PING.pdf

[2] Activision Publishing, Inc. (2024, April 4). *Call of Duty Update: An Inside Look at Matchmaking*. https://www.callofduty.com/blog/2024/01/call-of-duty-update-an-Inside-look-at-matchmaking

[3] Wikimedia Foundation. (2024a, January 26). *Multiway number partitioning*. Wikipedia. https://en.wikipedia.org/wiki/Multiway_number_partitioning

[4] Wikimedia Foundation. (2023, December 8). *Largest differencing method*. Wikipedia. https://en.wikipedia.org/wiki/Largest_differencing_method