

Coordination vs Competency:

Variation in metagame outcomes in amateur and professional Counter-Strike.

Taha Ziad, [REDACTED]

Project Motivation

Video games, as software, are extensively data rich by their nature. Counter-Strike being no different provided a unique opportunity to review granular match statistics, given the ability to parse their native data files “Demos” or .dem files. This proprietary data type is a full statistical breakdown of a given Counter-Strike match. The expanse of data, along with the novelty of using such a unique python package “awpy”, enabled us to produce something new to the ever growing zeitgeist of Counter-Strike data analysis.

Primary Data Source: Professional Match Data from HLTV

For professional matches the premier source of demo files is <https://www.hltv.org/>. A historical repository, alongside a third party news site for Counter-Strike. HLTV works closely with tournament organizers and teams to make demo files readily available to the public. They do not have an open API and therefore files are generally manually gathered as is done for this report, but methods for scraping while feasible goes against their terms of service.

2.2. You are not allowed to: commercially exploit the Website or any of its content in any manner; modify in any manner any part of the Website or content, except for Your Content as described below; conduct, facilitate or organize data mining or web scraping in relation to the Website and/or any of the content; or construct and build a similar or competitive website, product, or service. - <https://www.hltv.org/terms>

As a result of these limitations, special care was taken to manually download each demo used in our analysis, which ultimately limited the scope of our data to two teams and a single tournament. Despite these constraints, from this small selection of matches we could gather the following:

DataFrame Name	Demos	Rows	Columns
bomb	22	2678	9
damages	22	12206	27
footsteps	18	318179	15
grenades	18	37538592	9
infernos	18	1750	15
kills	18	2426	47
player_round_totals	18	530	4
rounds	18	368	9
server_cvars	18	1068	3
shots	18	56066	14
smokes	18	2188	15
ticks	18	26841040	10

The data above contains statistics from roughly 12gb of demo files gathered from HLTV. This was every match played by “Vitality” and “Mouz” (ranked 1 and 2 respectively), from a recent tournament “IEM Dallas” this past May. In order to parse these files the most important dependency was use of the “awpy” python package (<https://github.com/pnxenopoulos/awpy>). This package allows , given any demo file, generation of the various polars data frames shown above.

Counter-Strike LAN matches are played on servers which operate at 128 ticks, meaning 128 times per second, data is gathered. As a result, in order to make our notebook readily available, data such as the “grenades” data frame was reduced by a factor of eight to allow us to upload the files to github for collaboration and efficiency purposes.

```
if "tick" in df.columns:
    df_filtered = df.filter(pl.col("tick") % mod_value == 0)
    df_filtered.write_csv(file_path)
```

Secondary Data Source: Public Matchmaking Data From Kaggle

As a result of Counter-Strikes rising popularity, and the availability of packages such as awpy, many fans of the game have taken similar advantage of the environment to contribute public data sets for others to use. One of many examples was a Kaggle data set “Counter-Strike 2 Statistics” (<https://www.kaggle.com/datasets/computingvictor/counter-strike-2-statistics>) which provides Counter-Strike data on casual matches of the top 100 players as of January 2024. The data provided in this source, while less exhaustive, provides a larger variety of players and a larger timeline to review against are primary sources.

Much of what is provided by this data source is aggregated (e.g. weapon usage rates and map win rates). As a result, in order to compare these statistics to our primary data set, care was taken to replicate the aggregation done in the Kaggle set for our demos. Due to this aggregation the dataset is much smaller at only around 6kb, despite this, it provides a fairly robust framework for analysis. Video game analytics is itself quite a sparse interest, even finding data for use can be a challenge, and so it’s rare to find such a comprehensive set even if limited in scope by player or timeline.

Data Manipulation Step 1: Parsing a Counter-Strike 2 Demo

The first step to parsing a demo is getting a .dem file. This can be as easy as playing a match yourself and using the console in game to begin recording a demo, then finding the file in your game folder. The real challenge begins when trying to accrue a large collection of demos. Thankfully, HLTV sponsors a collection of demo files for matches. As a result, while tedious, it was mostly trivial to get the demo files specifically. Provided at the end of this document are links to the match pages used to download the demos.

Once you have your collection of demos it is time to install “awpy”. Awpy is only the current forefront of a long lineage of comprehensive demo parsers built over the past 10 years. It runs off another parser “demoparser2” which runs on Go. As such before attempting to run awpy parsing it’s required to install Go.

Once installed parsing demos is quite simple:

- Create a demo object from demo filepath
 - `dem = Demo(demo_path, verbose=verbose)`
- Call the parse method to populate the object with attributes containing the various element dataframes
 - `dem.parse()`
 - This is where Go is used

- Once parsed, based on the name you can pull a particular data frame by using its name to get that attribute (now a polars dataframe) for a particular demo:

- `kills_df = dem.kills`

- Now you have a dataframe you can use it like any other, in our case we saved these as csv and uploaded them to github. But I think in future, it might be better to use parquet files so we can retain more information without having files as large as the .csv's.

Data Manipulation Methods Step 2

In using Counter-Strike demo as a data source, we were lucky enough to not have to deal with missing, incomplete, or incorrect data as the game collects all information including player location, guns used, attacker, victim, and etc. However, it took a few steps to combine parsed data into dataframes for comparison.

In separating the files into the right dataframes, it was easy enough to differentiate between “kills” files from “rounds” files and “mouz” files from “vitality” files. The map and opponent names were in the file name, so proper splitting at “-vs-” for opponent and another split at “-” allowed us to find the proper opponents and maps played. Where we ran into difficulty was setting the opponent when Mouz and Vitality were playing against each other. Having multiple nested if statements to differentiate files and teams, the code was erroneously setting Vitality’s opponent as itself when the filename contained both Mouz and Vitality. To fix the issue, we added `team_name` and `file_type` arguments into the function, which was used to for example differentiate between Mouz’s kill dataframe from Vitality’s kill dataframe. This allowed for more individualized functionality while allowing more universal parsing of different types of files. To grab files, the user simply had to include team name, file type, and roster (if they wanted only team specific data)

- `df_kills_vitality = get_df("vitality", "kills", roster = vitality_roster)`
- `df_grenades_mouz = get_df("mouz", "grenades", roster = mouz_roster)`

Because all the corresponding files had the same columns, each was simply concatenated using `pd.concat` without having to adjust any column names. The “`assister_steamid`”, “`victim_steamid`”, and “`attacker_steamid`” columns were dropped for privacy concerns.

Our second dataset which included data for Top 100 players had some adjustments necessary for analysis. Since we planned on just adding pro data in addition to the Top 100 data, we didn’t adjust any formatting for the second dataset but we adjusted the columns with percentages from string to float and added the label “Top100” to differentiate between the pros and top 100 players. Since we used map and weapons statistics datasets for Top 100 players, there were specific changes made to each dataset on top of what is mentioned above. Specifically map names and weapon names were different from our pro match datasets. There was no pretty or convenient way to match the names other than to code each replacement using `.replace()` or `.map()` functions.

Where major data manipulation occurred was in pro datasets for both maps, weapons, and grenade analysis. They took similar steps in analysis but processing steps were slightly different. For maps analysis, we first had to find the side (attacker or defender) the teams were on using:

- `df_kills.groupby(['map', 'opponent', 'round_num'])['attacker_side'].first().reset_index()`

Then merged on existing rounds dataframe on map, opponent, and rounds. We then applied `groupby` on the rounds df on map and team’s side and aggregate on `team_won` column to calculate the number of rounds won and total number of rounds played. The dataframe was pivoted to have the same format as the Top 100 maps dataset. This only calculated per side win and number of rounds played data. Using simple arithmetics, we calculated per side win percentages which allowed for direct comparison to the second dataset and added a “Team” column to differentiate the teams.

For the weapons comparison, the steps were similar to comparing the map win rates. We first cleaned up the damages dataframe by grouping player, weapon, and hitgroup to calculate the percent of times a player hit a certain body part given a weapon. The following dataframe was pivoted to have the hitgroup as columns. This gave a similarly formatted dataframe to the Top100 weapons dataframe. To concatenate the two dataframes, we first had

to change the columns in Top100 dataframe from string to float and match the weapon name formats. The Top100 dataframe had proper weapon names you'd find at a gun range whereas the pro's dataframe had all lowercase and sometimes ambiguous names. There was no easy way to format these other than to hard code each replacement. Unlike the maps dataset, there was one more step in analyzing the data. The pro dataset had "right_arm", "left_arm", "stomach" and "neck" in hit locations which were not present in the Top-100 dataset. Upon further deliberation, we combined arms and neck with the chest and dropped "stomach" from our analysis. This was done due to their damage multiplier. The arms and neck, like the chest, had 1x damage multiplier but the stomach had 1.25x damage multiplier. Although the stomach accounted for far fewer data points than chest due to most players aiming for the head, we believed that it was important to differentiate two different damage multipliers

The utility analysis brought a new set of challenges in formatting datasets. It required 4 different datasets (rounds, grenade, kills, and damages) to be formatted the same. The round_df was used as the basis for the merged dataframe as it contained round number, map, opponent, and team's side information. The grenade_df contained grenade information at each tick that the demo was parsed. This meant that there was a lot of unnecessary location information for the same grenade when it was being thrown. So, it was filtered down to have unique "entity_id":

- `df_grenades_vitality.drop_duplicates(subset = ["entity_id"])`

Since location information was not necessary, this reduced the size of the dataframe from +20,000 rows to <300 rows. The grenade_df was grouped by map, opponent, round number, and grenade then unstacked to count the instances of each grenade per round. The kill_df was only used to count for the number of flash assists in a round. The damage_df contained all instances of damage that were incurred by players, so we filtered this to only grenades. Just like the damage_df, it was grouped by map, opponent, and round_num then unstacked to have the same format as the rest. Then all four dataframes were merged on map, opponent, and round_num and Nan values were filled with 0 indicating no damage done or no grenades thrown. This created an all encompassing dataframe containing the number of each grenade thrown and damage done per grenade and flash assist in a round.

There weren't any major difficulties in prepping the datasets for analysis since there were no missing data points in parsed demo files and if there were any missing data points like the utility dataframe, we could simply replace them with 0 to indicate no utilities used. The only difficult part was checking to make sure that before merging or concatenating multiple dataframes, we had to individually check each groupby functions to confirm correct formatting and naming schemes.

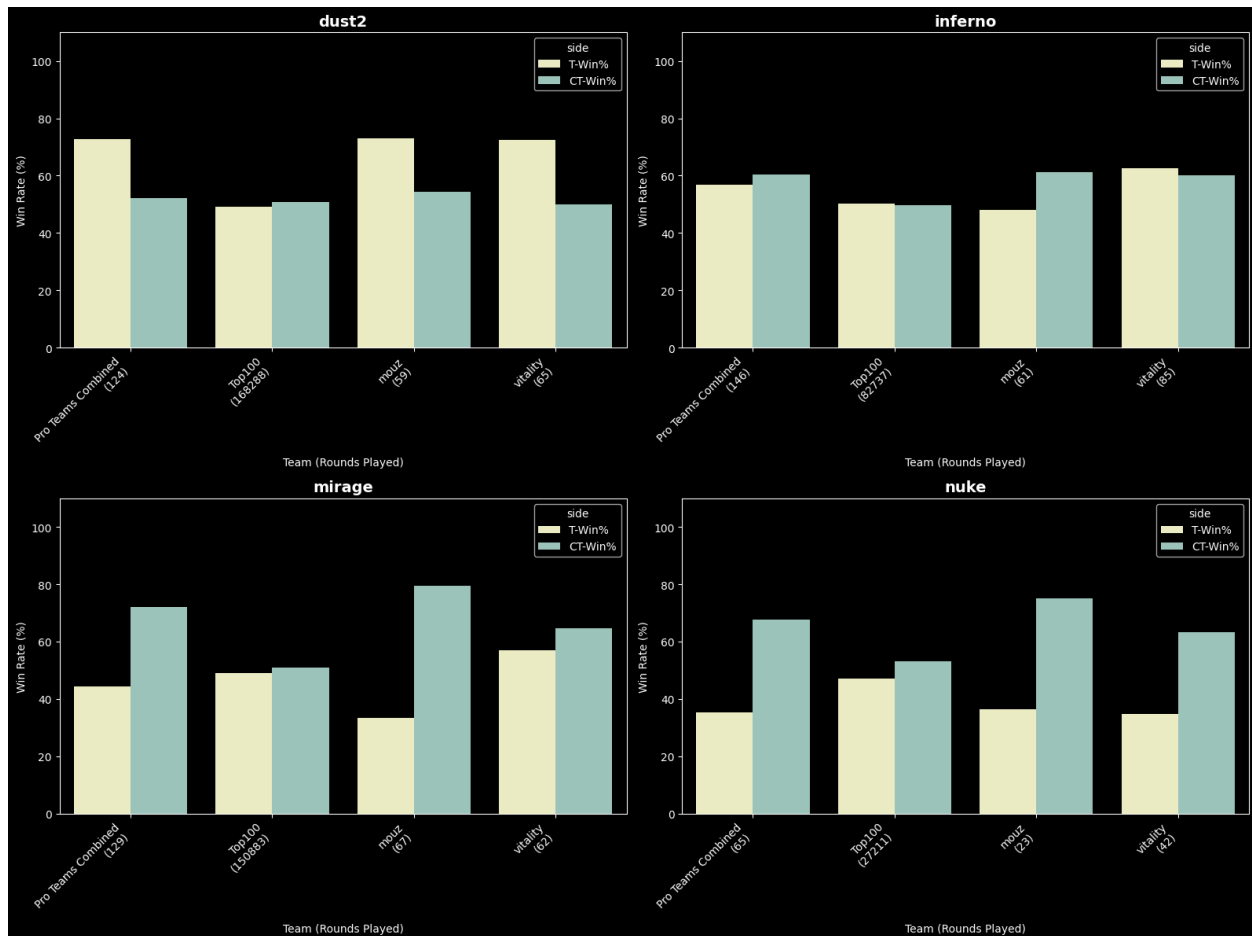
In creating visualizations, we had to melt the dataframes to have a proper axis. For instance in map visuals, we had to create per map, per team, per side data points by melting on `id_vars = ["map", "team", "Rounds-Played"]` and `value_vars = ["T-Win%", "CT-Win%"]`. Allowed us to separate the bar plots based on teams.

Map Win Rate Comparison

Comparing the T-Side (Terrorist) and CT-Side (Counter-Terrorist) win rates across different maps reveals insights into team-specific strengths and weaknesses, as well as how they stack up against the broader Top 100 aggregated data.

Maps	T-Win%	CT-Win%	Team
Dust 2	49.2	50.8	Top100
Mirage	49.1	50.9	Top100
Inferno	50.3	49.7	Top100
nuke	47.0	53.0	Top100

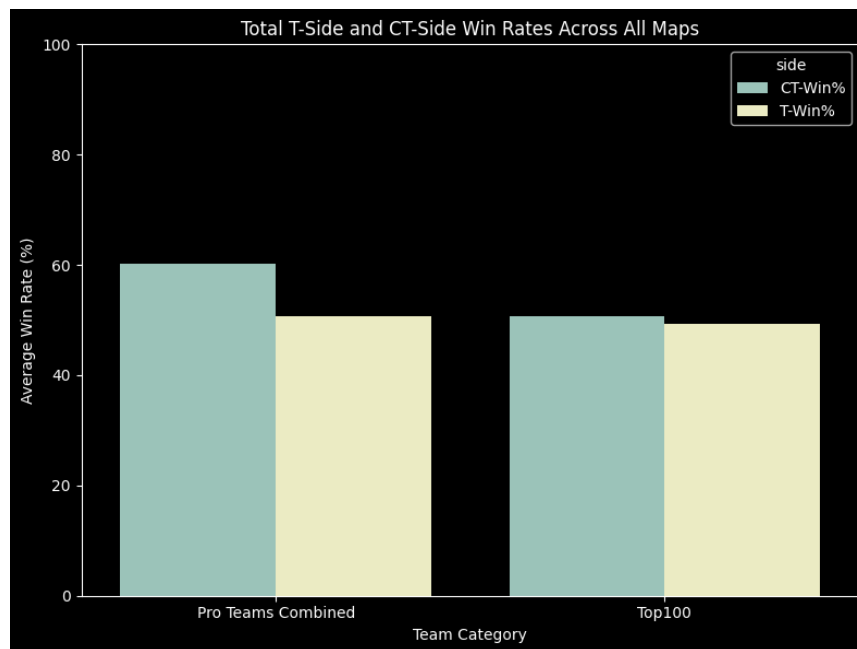
Dust 2	70.4	50.0	MOUZ
Inferno	41.4	59.5	MOUZ
Mirage	28.9	79.4	MOUZ
nuke	33.3	75.0	MOUZ
Dust 2	65.6	50.0	Vitality
Inferno	61.0	60.0	Vitality
Mirage	51.6	59.5	Vitality
nuke	33.3	63.2	Vitality



Note: In professional matches because there are 7 available maps, and a maximum of 5 will be played, a pick/ban phase is introduced. In casual matches the players pick a preference before joining, and it is randomly decided when a match is found.

Dust 2 is very slightly attacker favored (T-Win%) for the entire league, this is reflected for the two organizations present. With a coordinated offense teams generally can overwhelm a defensive strategy that involves splitting resources across the two sites of interest. In public matches where teams are made mostly randomly apart from being within a skill range, disconnects in metagame strategy can cause coordination to falter. Leading to slower rounds where the defensive side can bolster areas of the map that would have already been overtaken if done decisively.

Inferno on the other hand is one of the most balanced maps in the pool. With the attacker advantage of 50.4% win rate across the entire league. Vitality being the best team, their extremely similar win rate across both attacking and defending seemingly reflects a deep understanding of how the meta for this map plays out. We would expect as the sample size increases, as with our top 100 players, that the win rate would also trend towards this mean and this is further reflected in the graph above. Mouz large discrepancy could indicate they are still developing their attacking strategy. And we'd expect given a larger data set that overtime their two sides may trend towards each other. For reference, Vitality's first pick is inferno, they currently hold a 93% win rate on this map in the past 3 months.

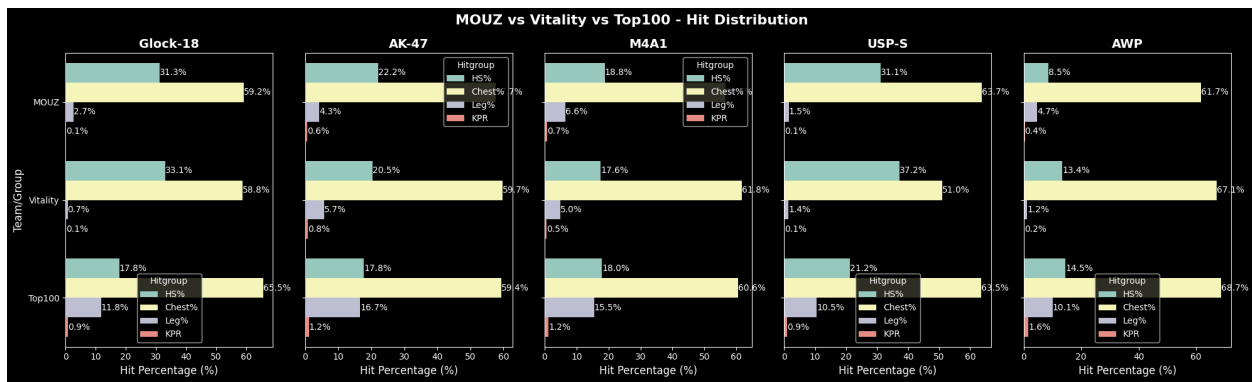
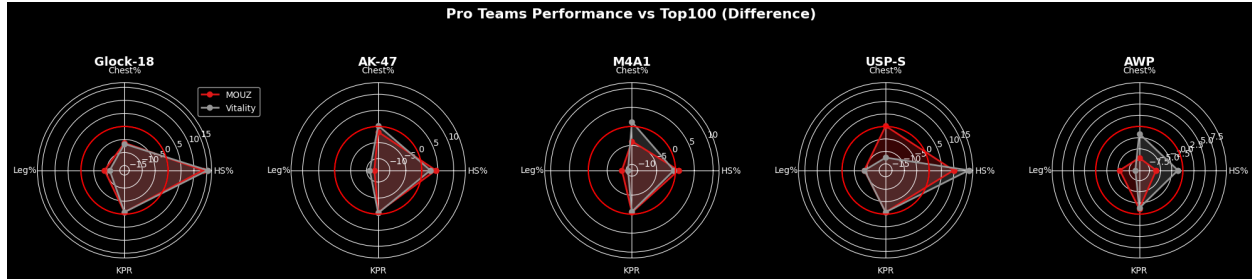


Mirage and Nuke are both heavily CT sided (52.5% and 54.5% respectively) across the league, and as expected the two top teams tend to reflect the design through their win rates. One notable difference between the two maps is that **Nuke** is the only map to have bomb sites on different floors. This novelty makes it quite unpopular in casual play, and therefore not well understood. And so this imbalance for attacking/defending isn't quite as strongly reflected in the win rates for the top 100 as it appears for teams like Mouz and Vitality. Nuke is Mouz's first pick, and it seems to be a good choice as they perform incredibly well on it time and time again.

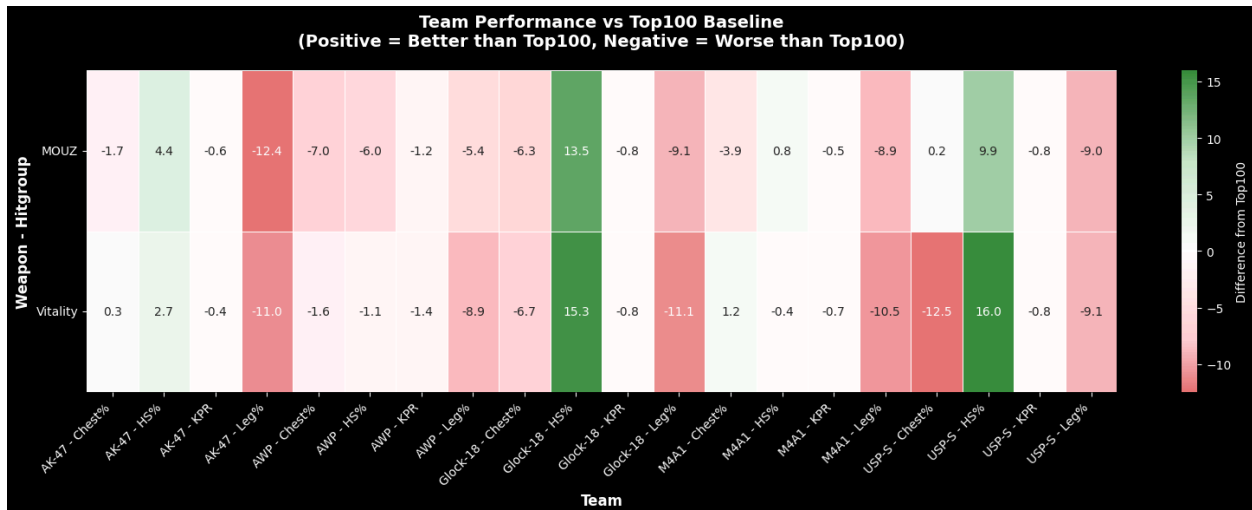
Overall: Map makers tend to agree that the more balanced a map is, the "better" it is. Whether this is true is up for debate, but it's a common statistic to review when making changes to a map over time. The idealized reason for this is reflected in the stats above where maps that are not balanced for both attacking and defending. Professional teams know how to take advantage of these inconsistencies between sides, and exacerbate the issues on the biggest stages. These teams push the limits of what is possible, and given an inch they time and time again, take a mile.

Weapon Comparison

Analysis of various weapon performances of Team Vitality and MOUZ IEM Dallas matches against a baseline from the Top 100 CS2 players. The metrics being analyzed are headshot percentages (HS%), chest percentage (Chest%), legshot percentage (Leg%), and kills per round (KPR) collectively referred to as hit groups.

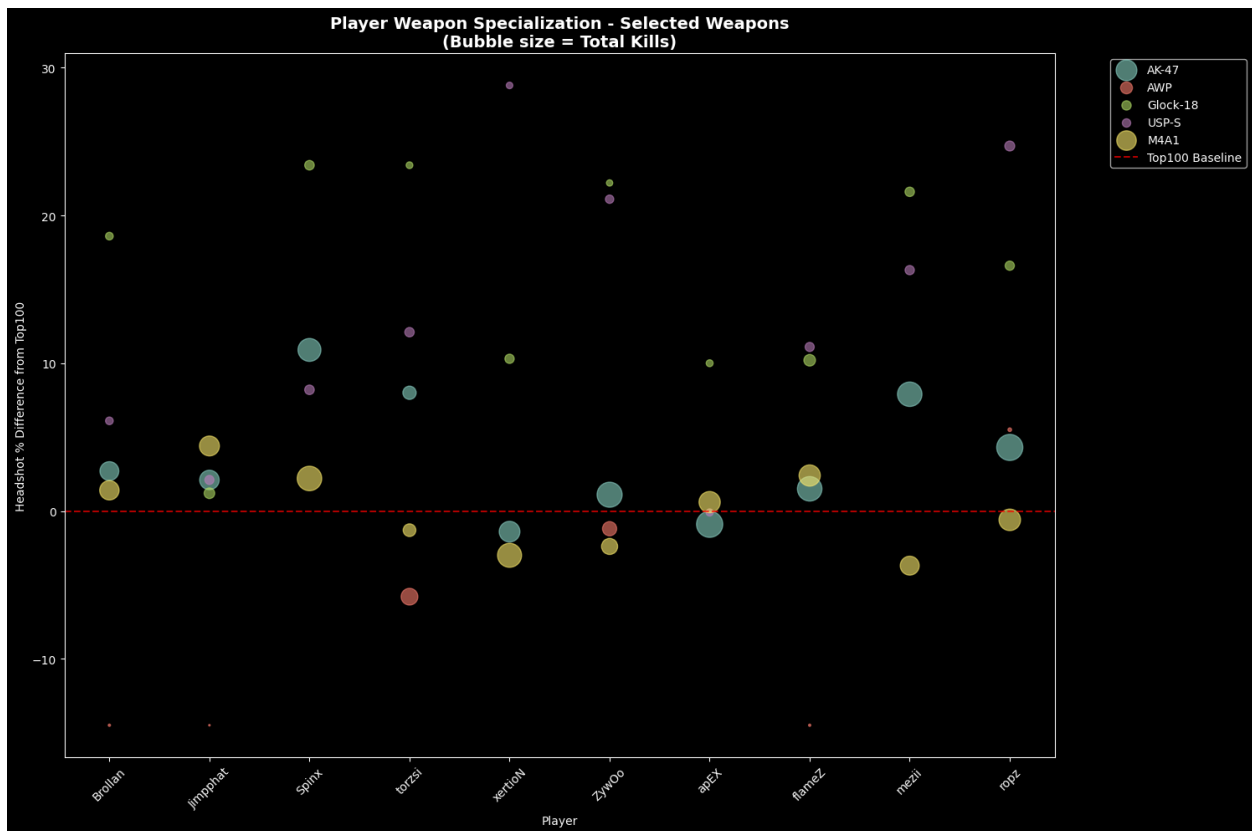


We can see that across pistols and the AK-47, pros have significantly higher headshot percentage compared to the top 100. This suggests overall better aim and more crucially crosshair placement. Crosshair placement aims to provide consistency in aim by positioning your crosshair where your opponent will be prior to seeing them. Pistols and the AK-47 also happen to be the most frequently used weapons in our data with both the pros and the top 100 players, generally staying in the top five weapons as shown above. This has to do with the default loadouts of each side's team as well as standard buys for in-game weapons. The concept of saving also plays a significant role in the weapon chosen. "Saving" is where a player may be in an unwinnable scenario and chooses to save their loadout and their money to make the next round easier for their team. Both pro teams generally have lower Leg% compared to the top 100 baseline. This suggests more precise aiming, as hitting the leg costs time and gives the enemy time to react to the engagement. Surprisingly, professional players have a lower KPR than the average top 100 player. This



is most likely due to a limitation in how the data is gathered. At the professional level, kills are a zero-sum game, as players on the same team cannot get the kills for players who have already died in a particular round. Meanwhile in our top 100 data set, all of these players could be playing in completely different games, and so there is more opportunity for a top 100 player to achieve higher average kills than a pro.

The heatmap reinforces these observations, clearly illustrating strong positive differences in pistol and AK-47 headshots for MOUZ and Vitality, while consistently showing negative differences in leg shots across all weapons, signifying precise aiming. Individual player data further highlights this, with many pros, notably "ropz" and "ZywOo," demonstrating headshot percentages well above the top 100 baseline, especially with pistols. The spider chart visually confirms these trends, with the shapes pulling outwards for headshot percentages and inwards for leg percentages and KPR, except for the AWP, where the pattern for headshots and chest shots is notably inverted for pro teams. This is likely to do with the use of the AWP, the AWP is a one-shot kill to the body and head, because the body is a larger target than a head the pro teams favor the body while using the AWP. This falls in line with the difference we have seen, where the top 100 lack the coordination that amplifies the raw talent of the pro players. The trends we see from the top 100 are amplified in a professional setting.

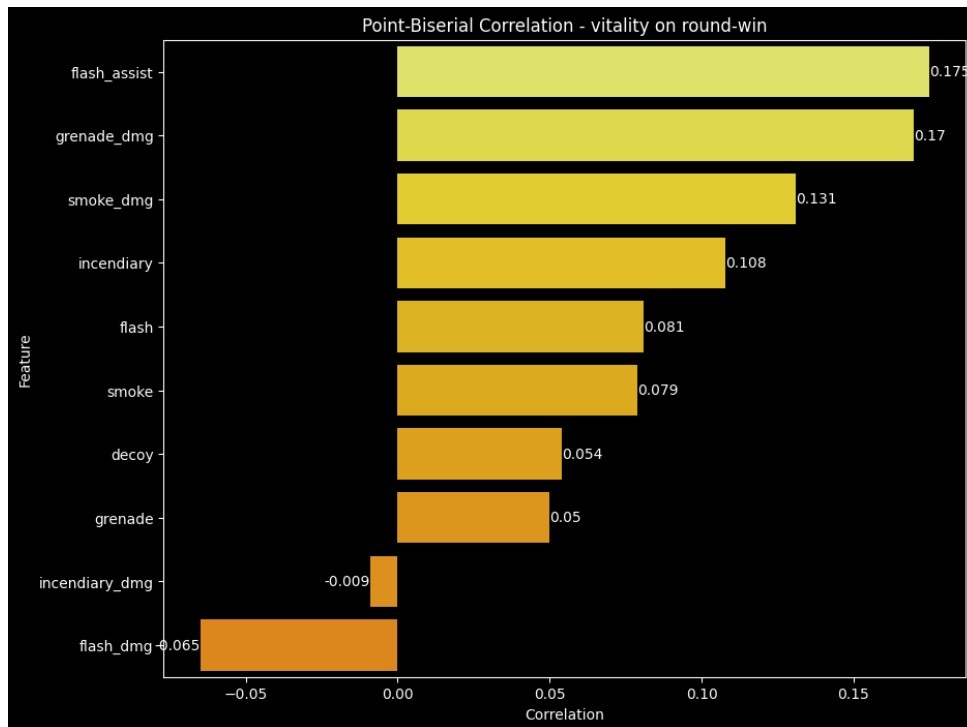


From the visual above we can see the difference in the players headshot percentages and total kills against the top 100 that give us the ability to look at the individual contributors to the pro teams out comes and performances. Across most of the players we can see a positive increase in headshot percentage over the top 100 with some notable exceptions. "torzsi", "xertioN", and "ZywOo" and have lower headshot percentages than the baseline with two weapons. For "torzsi" and "ZywOo" they are on AWP more then other players and being that the AWP is a high skill weapon that in the hands of pro is being used more for body shots rather than heads we can understand why they have lower headshot percentage. "xertioN" performed worse than most of the other pros and the top 100 with both the AK-47 and the M4A1, both are the primary weapons for the teams (T and CT). The

individual player variation that we see can be attributed to role specific play with entry fraggers (leads a push) having higher headshot percentages and more kills. While more supportive players have lower headshot percentages and similar kill counts. We can use this plot to distinguish the aiming precision, particularly in headshots for these players and how this differs from the top 100.

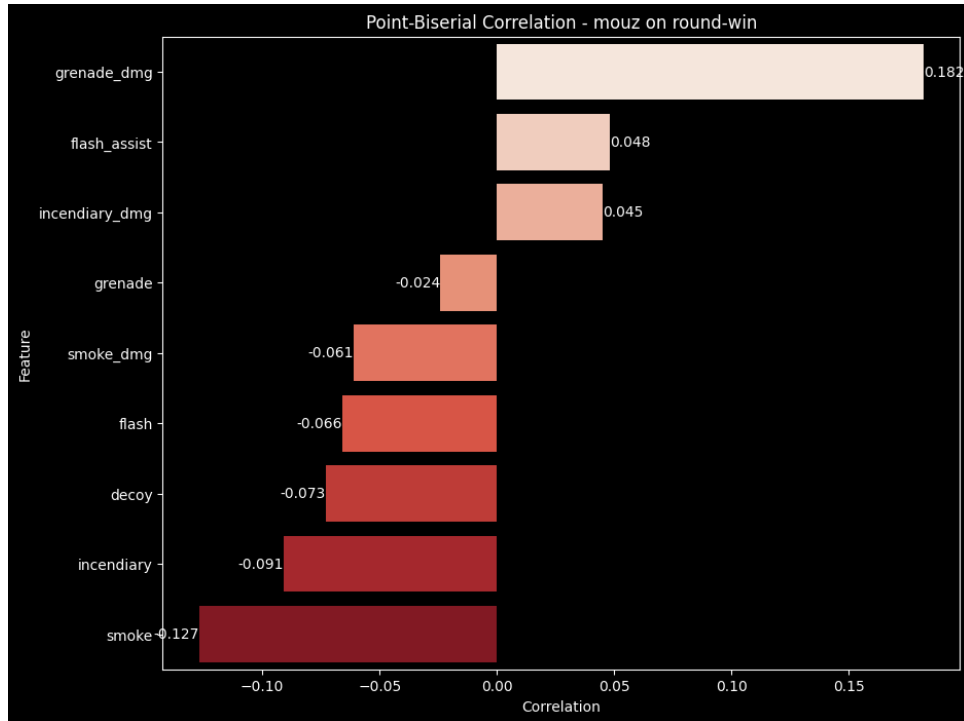
Utility's Effect on Round Win

One of the largest contention in differentiating teams consisting of top 100 players and top pro teams is how holistic a team is. Since holisticness is not a measurable trait, we used utility usage as a proxy to measure interconnectedness of the teams. With just 100 players, we can form nearly 7.5 million team combinations, meaning consistent team formation is nearly impossible with most players playing online. Also, the environment they play in is not consistent. The pros play in a well air conditioned studio environment with local servers (no internet lag) allowing the most distraction free environment to communicate and execute strategies. Whereas some online players may not have access to mic or their internet connection may be unstable. These factors can affect communication amongst the players and thus utility usage and round win rate. As seen from above, we found that across all maps, the pros had higher win rates for both CT and T sides compared to the Top 100 players. How much of that is due to utility usage?



X_dmg indicates X utility's damage in a round
X indicates number of X utility thrown in a round

From the correlation plot above, there is positive correlation between Vitality's utility usage and them winning the round. The winning team, Vitality, has a positive correlation between almost all utility usage and them winning the round, with the number of flash assist and grenade damages in a round having the highest correlation. This positive correlation of utility usage and round outcome is probably a good indication of their recent success in other tournaments as well as their cohesiveness. According to [HLTV.org](https://www.hltv.org), in the past 3 months, Vitality has ranked 2nd in flash assists per round and 4th in utility damage per round amongst top 20 teams enroute to winning 5 international tournaments including IEM Dallas.



Mouz on the other hand has a nearly opposite correlation plot compared to Vitality. For most of the utilities other than grenade and incendiary damage and flash assists, utility's usage has negatively affected their round outcome. This is consistent with HLTV ranking Mouz as 10th in utility damage per round and 14th in flash assists per round amongst top 20 teams in the past 3 months. Although the exact reason for their utility struggles can not be measured with the data we have, we can assume the reason to be in line with the timing of utility use (strategy). Mouz placing 2nd in IEM Dallas is not an easy feat, but knowing that their utility usage is subpar compared to Vitality, we can assume Mouz has created round winning opportunities through other measures.

Looking more closely at the numbers, the correlation between the utility and round-win is not strong for both Vitality and Mouz. Does this mean that utility is not a key component in winning the round? Probably not. In Counter-Strike, utilities are used to set up the round. For example, utility blocks an enemy's line of sight or takes out an enemy from an important position when attacking. And as a defender, you can nullify attackers' utilities by throwing an incendiary to block them from entering a site. Although the damages with the utilities may not overturn the outcome of a round, it creates more opportunities to get kills, which likely has the strongest correlation with the round win. This presumption is confirmed by flash assists having positive correlations with round win for both Mouz and Vitality. More flash assists a team has, they likely have more kills and better team coordination. Thus, we can't ignore utility usage just because the correlation is weak. Utility usage may not strongly affect the round outcome, but it can definitely alter a team's success in a tournament.

In answering if a pro team's utility usage is a crucial factor in differentiating them from the top100, the answer is not conclusive. The outcome has shown that utility definitely has some correlation with the round outcome. But with a small sample size of one tournament for two teams and using the utility as a proxy to cohesiveness may not be sufficient in understanding cohesiveness' effect on winning. There is more than just damage and number of utilities in a round such as timing, location (smokes could land in a wrong place), and what happens after the utilities have been used that impact the round outcome. We can conclude that there is correlation between utility and round but using it to measure cohesiveness is still up for further investigation as more information is needed to provide a concrete answer.

Links:

Match pages for demo files:

- <https://www.hltv.org/matches/2382379/mouz-vs-bcgame-iem-dallas-2025>
- <https://www.hltv.org/matches/2382683/liquid-vs-mouz-iem-dallas-2025>
- <https://www.hltv.org/matches/2382689/mouz-vs-falcons-iem-dallas-2025>
- <https://www.hltv.org/matches/2382375/vitality-vs-legacy-iem-dallas-2025>
- <https://www.hltv.org/matches/2382675/gamerlegion-vs-vitality-iem-dallas-2025>
- <https://www.hltv.org/matches/2382681/vitality-vs-the-mongolz-iem-dallas-2025>
- <https://www.hltv.org/matches/2382694/vitality-vs-falcons-iem-dallas-2025>
- <https://www.hltv.org/matches/2382693/mouz-vs-the-mongolz-iem-dallas-2025>
- <https://www.hltv.org/matches/2382695/mouz-vs-vitality-iem-dallas-2025>

Past 3 months stats per team

- <https://www.hltv.org/stats/teams/ftu?startDate=2025-03-11&endDate=2025-06-11&rankingFilter=Top20>