SCOPE: Selective Cross-Validation over Parameters for Elo

Alexander J. Bisberg,[1] **Rogelio E. Cardona-Rivera**[1,2]
[1]School of Computing
[2]Entertainment Arts and Engineering Program
University of Utah
Salt Lake City, UT 84112 USA
alex.bisberg@utah.edu, rogelio@eae.utah.edu

Abstract

It is crucial to develop reusable methods for explaining and evaluating esports given their popularity and diversity. Quantifying skill in an esport has the potential to improve win prediction, matchmaking, and storytelling for these games. Arpad Elo's skill modeling system for chess has been adapted to many games and sports. In each instance, the modeler is challenged with tuning parameters to optimize for some metric, usually accuracy. Often these approaches are one-off and lack consistency. We propose SCOPE, a framework that uses grid search cross-validation to select optimal parameters for Elo based on *accuracy*, *calibration*, or *log loss*. We demonstrate this method on a season of Call of Duty World League, a first-person shooter esport, and we demonstrate comparable performance to other more complex, state-of-the-art methods.

1 Introduction

In an early attempt to understand esports, Wagner (Wagner 2006) provided a perspective on the nature of Counter-Strike (Valve Corporation 2000):

> ... teams are faced with an extremely well defined virtual environment in which the only way of winning a match is to find and execute strategies that outperform the strategies of the opposing team.

Playing these games in competitive contexts takes unique and advanced expression of **skill**, or ability and capacity to execute activities that overcome challenges around ideas, things, or people. Precisely quantifying skill is of interest for a variety of reasons. Spectators enjoy games that elicit a sense of drama (Winn 2015); teams of similar skill play games that have a higher degree of uncertainty in outcome, eliciting a greater sense of drama (LeBlanc 2006). Players would rather play close matches than not, thus being relevant for matchmaking (Chen et al. 2017). How to best quantify skill is an open problem.

In the 1950s Arpad Elo, a Hungarian chess player and mathematician, invented the Elo (1978) model to quantify skill – and win probability – of chess players as they progressed through tournaments. His system would go on to be adopted by the US Chess Federation (USCF) and a modified version is still used today (Glickman 1995). Elo's method and modifications of pairwise comparison have been extended past individual board games to team sports. For example, Nate Silver (Silver 2012) and his blog FiveThirtyEight have applied the Elo model to predict outcomes in Major League Baseball, English Premier League soccer, professional tennis, and more. Further, OpenDota (Cui, Chung, and Hanson-Holtry 2018) calculates Elo ratings for professional Dota2 (Valve Corporation 2013) matches.

Recently, pure Elo-based models have been eschewed in favor of more elaborate models. For instance, Microsoft developed TrueSkill™ (Herbrich, Minka, and Graepel 2007) and its successor TrueSkill2 (Minka, Cleven, and Zaykov 2018), each a probabilistic inference-based system to automatically learn a player's skill rating. When applied to the problem of win prediction on Halo 2 (Bungie 2004) multiplayer online matches, the accuracy of these models are 52% and 68% respectively (Minka, Cleven, and Zaykov 2018). They claim the TrueSkill models are an improvement over Elo (Winn and Bishop 2019) given "it is difficult to see what assumptions are being made by making changes to the [Elo] algorithm" (p. 152) and that it is "hard to predict how well the resulting [Elo] algorithm will work" (p. 152). In other work, Delalleau (2012) developed a neural network-based approach for modeling skill and making predictions for balance and player enjoyment in Ghost Recon Phantoms (née Ghost Recon Online) (Ubisoft Singapore 2014).

However, *these more elaborate models sacrifice interpretability for accuracy*. We posit that if something is not immediately working in one of these models, they are no easier to "see what assumptions are being made" nor "predict how the resulting algorithm will work."

In this work, we demonstrate that Elo-based models can achieve comparable levels of accuracy while retaining their interpretability. We agree with the authors of these more elaborate models that it is necessary to have robust and adaptable systems to characterize player skill (especially in nascent areas, like esports) and we recognize that although Elo has already been use to quantify skill, the methods for adapting Elo are typically unsystematic and developed on a case-by-case basis. To address these limitations, we present a framework based on Selective Cross-validation Over Parameters for Elo, or SCOPE, an automatic parameter-tuning and cross-validating framework for Elo, which (a) clearly identifies the model's parameters, (b) systematically optimizes the parameters by grid-searching in a space of values, and (c) supports three different modes of optimization, over *accuracy*, *calibration*, and *log loss*.

While we agree with Winn and Bishop that it is difficult to predict Elo's success given its *baseline* configuration, SCOPE explicitly identifies the variables involved and systematically explores how those variables affect the Elo model's prediction. By changing multiple parameters and back testing them, it is easier to predict how Elo will work.

2 Background & Related Work

Elo developed his model on the basis of pairwise comparison techniques within statistics. These techniques, pioneered by Thurstone (1927), characterize the degree of preference between two objects. Although intended for the social sciences, Elo saw that this method could also be applied to games insofar as a win is preferred to a loss. Analogously, a larger difference in score leads to a larger magnitude of preference. Thurstone assumes that each score is fit to a Gaussian distribution. Using this assumption, the probability that one score is greater than the other can be directly calculated.

2.1 Setting Up Elo

Let $G(\bar{x}_a^i, \bar{s}^2)$ represent Team A's (T_a) initial normally-distributed ability (*i.e.* Elo) score with mean $\bar{x}_a$ and variance $\bar{s}^2$ prior to game i; let $G(\bar{x}_b^i, \bar{s}^2)$ represent the same for Team B. The difference between these two ability scores is a Gaussian with mean $\bar{x}_a^i - \bar{x}_b^i$. By definition from pairwise comparison, if $\bar{x}_a^i > \bar{x}_b^i$ then T_a wins game i. The probability that $\bar{x}_a^i - \bar{x}_b^i > 0$ can be approximated by the area under the normal probability density function, or the point on the cumulative density function at the value $\bar{x}_a^i - \bar{x}_b^i$. Let W_t^i represent the event that Team t (T_t) wins game i. Then:

$$\Pr(W_a^i) = \frac{1}{1 + 10^{(\bar{x}_a^i - \bar{x}_b^i)/w90}} \tag{1}$$

$$\Pr(W_b^i) = \frac{1}{1 + 10^{(\bar{x}_b^i - \bar{x}_a^i)/w90}} = 1 - \Pr(W_a^i) \tag{2}$$

The win probability is calculated using Eq. 1 and 2 above for each team respectively. They are designed to be reflexive so $\Pr(W_a^i) = 1 - \Pr(W_b^i)$. Elo decided to use base 10 instead of Euler's constant (e) because it is easier to calculate win probabilities for specific score differences. This allows the variable $w90$ to represent the point at which a mean score difference has a 90% chance of winning. In Elo's (1978) original model $w90 = 400$. In that case, if $\bar{x}_a^i$ is 400 points above $\bar{x}_b^i$, T_a will have a 90% chance to beat T_b in game i.

Not only does Elo give us a conception of the probability for one team to beat the other, but he proposes a way to update a team's (mean) score after a result. After the completion of a game, each team's Elo score is incremented by the difference in the expected score (E[S]) and actual score (S). This quantity is then scaled by some parameter K.

$$\bar{x}_a^{i+1} = \bar{x}_a^i + K(S_a^i - \mathrm{E}[S]_a^i) \tag{3}$$

$$\bar{x}_b^{i+1} = \bar{x}_b^i + K(S_b^i - \mathrm{E}[S]_b^i) \tag{4}$$

$$S_t^i = \begin{cases} 1 & \text{if } T_t \text{ wins game } i \\ 0 & \text{otherwise} \end{cases} \qquad \mathrm{E}[S]_t^i = \Pr(W_t^i)$$

In other words, a team receives a score $S = 1$ if that team actually won the game, otherwise $S = 0$. The expected score is the probability for that team to have won the game based on the previous score, calculated from Eq. 1 or 2. The idea is that if an outcome is more surprising, a team's score should be increased in the direction of that surprise. In the original Elo model, K is a constant that represents the importance or significance of a game on the actual score of a player; the factor is thus the weight of the game being played. In our model, K takes on a learned baseline value and is modified on a per-game basis. This is elaborated in Section 3.2.

2.2 Adaptations of the Elo Model

The original Elo rating system is fairly simple and several parties have introduced adaptations to model more complicated phenomena; we review some adaptations that we found relevant for modeling esports.

Mark Glickman (Glickman 1995; Glickman and Jones 1999) is responsible for many of the modifications to the original Elo model applied to chess. His adaptations include (a) adjusting parameters based on historical data, (b) representing changes in skill over time, (c) modeling variance in skill, and (d) accounting for game-specific influences; *e.g.* in chess, the player controlling the *white* pieces has an advantage simply because they go first. He named this updated version the *Glicko* model.

FiveThirtyEight introduced modifications like (a) game-specific differences that favor a particular team; *e.g.* in baseball, home field advantage, park effects, starting pitcher (Boice 2018) and in tennis, number of games played (Morris and Bialik 2015) and (b) the expansion of Elo to account for multiple players – who could switch teams during a seasons of play. SCOPE outlines the possible modifications to an Elo model and optimizes the parameters correctly. This is elaborated further in Section 3.

2.3 Model Assessment Metrics

Kovalchik (2016) developed a framework to evaluate four different tennis prediction models (including FiveThirtyEight's Elo model) by training them all on a year's worth of data and using them to predict the subsequent year's results. We adapted this method to assess our own model. The framework compares models using the metrics *accuracy*, *calibration*, *log loss*, and *discrimination*. We focus on the first 3 and leave discrimination for future work. In the following equations, n is the total number of series' played, and S_x and $\mathrm{E}[S]_x$ refer to the actual score and expected score for a *series* (as opposed to a single game, as before).

Accuracy For this metric, we count a *correct* prediction when the team that is expected to win over 50% of the time actually wins. The *accuracy* of the model is the total number of correct predictions divided by the total number of predictions we made over every series.

$$correct = \begin{cases} 1 & \text{if } \mathrm{E}[S]_x > 0.5 \wedge S_x = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$accuracy = \frac{\sum^{n} correct}{n} \tag{5}$$

Calibration This is the sum of the win probabilities for each team predicted to have over a 50% chance of winning divided by the number of times those teams actually won. If the ratio is over 1, better teams are predicted to win more often than they actually do, measuring over-prediction.

$$correct = \begin{cases} 1 & \text{if } \mathrm{E}[S]_a > \mathrm{E}[S]_b \wedge S_a = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$calibration = \frac{\sum^{n} \mathrm{E}[S]_a}{\sum^{n} correct} \tag{6}$$

Log Loss This is a ratio of two quantities. The numerator is the sum of (a) the product of when the player actually won times the log of the prediction accuracy and (b) the product of when the player lost times the log of the prediction. Log loss closer to 0 is better.

$$logloss = \frac{-\sum^{n} S_a \log \mathrm{E}[S]_a + S_b \log \mathrm{E}[S]_b}{n} \tag{7}$$

3 Framework

In this section we outline the aforementioned Elo model adaptations and the method to use the model assessment metrics. Together these steps comprise SCOPE.

3.1 Elo Initialization

Part of the power in the Elo model lies with its ability to "remember" history, applying updates over time. What happens where there is no history? Elo acknowledges the challenges of starting a rating system from scratch:

> The working formula of the Elo system presume an existing pool of rated players. A special problem arises when an isolated group of players is to be rated for the first time... (Elo 1978, p. 53)

Glickman and Jones (1999) describe a provisional rating system where scores stay fixed for the first 20 games.

Our dataset (described in Section 4.1) contained a small number of games, such that we did not want to waste any data on the provisional phase. To manage this, we developed a technique based on mapping past team performance onto a rank and then mapping the rank onto an Elo score for the subsequent season of play. There are no global rankings released for the CWL, so we calculated a team's rank as the difference between number of games won and number of games lost in the last tournament of the previous season, *i.e.* teams who won the most games were assumed to be the most skilled. For g games played, team t's rank T_t^{rank} is:

$$\text{won} = \sum_{i=0}^{g} S_t^i \qquad T_t^{\text{rank}} = \text{won} - (g - \text{won})$$

As Elo notes, "The rating scale itself - its range of numbers - is, like any scale without reproducible fixed points, necessarily an open-ended floating scale" (Elo 1978, p.18). Thus, we fit the ranking to a normal distribution with a $\mu = 1500$ and standard deviation of $\sigma = 100$, per Elo. Any team who did not play in that tournament was rated in the bottom 12.5% since most major teams compete in the final tournament.

3.2 K Baseline and Updates

There is not a specific formula to calculate K, but we can use some heuristics and techniques for finding the baseline K value (initialization), and then determining the amount to modify K for a particular match-up (updates). These per-game updates can make model predictions more accurate (Elo 1978; Glickman and Jones 1999).

We posit that K should be some small percentage of the population mean Elo scores μ and also below the population standard deviation σ. Otherwise, the scores would balloon out of proportion after a single game (and would require a manual reset of Elo scores as is done in prior work). Elo (1978) sets K in the range $16 - 30$ for chess. FiveThirtyEight (Boice 2018) and Lacy (2018) set K in the range $30 - 50$ for football and basketball ($\mu \approx 1500, \sigma \approx 100$). We experimentally tested possible values of K in these ranges. We underestimate K starting at 1 and increase up to 50, near the upper limit of where most have tested. The upper value is half of σ, so scores will not increase excessively.

We can use more data than just wins and losses to determine a score update. In general, this is captured by changing K on a per-game basis. Factors that could alter our belief about how much each team's score should change include margin of victory, magnitude of perceived skill, and change in skill over time. All updates to K are of the form:

$$K_{i+1} = K_i + (K_{\text{update}})$$

Where K_0 is the K value as initialized and K_{update} is determined by the methods described below.

Margin of Victory The intuition behind this K update is that one team is more skilled than another if the former beats the latter with a large margin of victory (MOV). Lacy (2018) implemented this in his Elo model for English Premier League soccer. His hypothesis is that MOV plays a larger role for close matches, but after a certain point the advantage trails off. To validate this assumption, we tested multiple MOV functions; each of these functions serves as a potential form of the K_{update} value:

$$K_{\text{update}} = \Delta w - 1 \tag{linear}$$
$$K_{\text{update}} = \log(150(\Delta w - 1) + 1) \tag{log}$$
$$K_{\text{update}} = \sqrt{100(\Delta w)} \tag{sqrt}$$
$$K_{\text{update}} = 3^{\Delta w} \tag{exp}$$

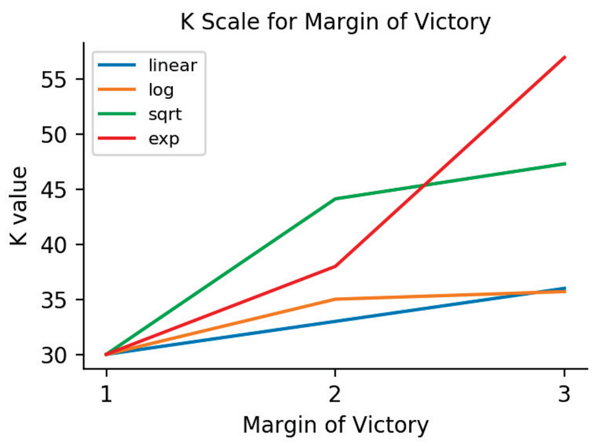

Figure 1: MOV functions with base $k = 30$.

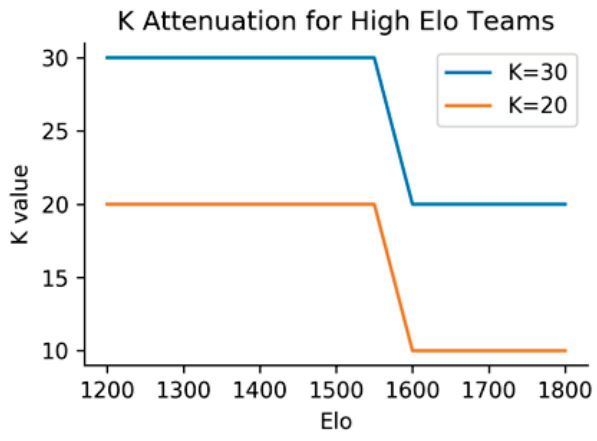

Figure 2: Reducing K at higher Elo scores has potential to increase model performance (Glickman and Jones 1999).

These equations were chosen because they kept K below σ, the population Elo standard deviation. We did not explicitly tune the coefficients for each of these equations in SCOPE. In future work, we hope to add these parameters to the cross-validation process within our model.

Score Cutoff Glickman and Jones (1999) has noted that attenuating ratings for higher Elo opponents is a good strategy for improving model performance. As demonstrated in Figure 2, one way to do this is with a piece-wise function and set a specific cutoff Elo score to decrease the K value.

$$K_{\text{update}} = \begin{cases} 0 & \text{if } \bar{x} \geq \text{cutoff} \\ -K_i + (K_{\text{scale}} * K_i) & \text{if } \bar{x} < \text{cutoff} \end{cases}$$

Once we are more sure that a team is skilled, we give them the benefit of the doubt going forward. High skill teams are less susceptible to the results of a single game. The cutoff

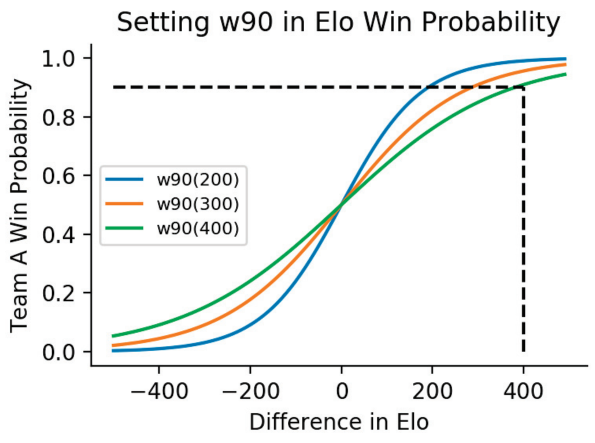

Figure 3: Win probability for a team with varying $w90$.

values are chosen every half standard deviation (50 points) above a team's Elo score ($\bar{x}$), starting at one standard deviation above μ (1600). In most training cases teams never reached scores over 1750, so this value represents no cutoff. This only happens on the high end and not the low end of scores. In Table 1, we refer to the point at which we make this decrease in K as **Cutoff**. The amount which K is reduced is called the **K Scale**.

3.3 Adjusting Win Probability Gap

Glickman and Jones (1999) found that when looking at historical data for the UCSF the existing model continuously over-predicted the strength of high ranked players. When he changed $w90$ from 400 to 591 his model would more accurately predict the winner. In this case, we tested values from 100 to 500 in increments of 100, starting at one standard deviation above from the mean Elo scores.

Tuning $w90$ enables us to control win probability based on skill disparity. Figure 3 illustrates different curves for the win probability function, and represents how tuning affects probability; in essence, tuning alters the curve's kurtosis. This makes sense, since kurtosis characterizes the possibility of extreme events occurring. A lower $w90$ means that we can be more confident in our predictions, since they're less susceptible to random events.

3.4 Skill Changes Over Time

Because some teams do not play in consecutive tournaments, their Elo may not be reflective of actual skill; some forecasters regress teams to the mean after a season or gap in play (Boice 2018). Regressing to the mean is a common technique to smooth out noisy data.

In Table 1, this is referred to as **Regression**. This is demonstrated in the equation below, where r represents the amount of regression:

$$\bar{x}^{i+1} = (1 - r)\bar{x}^i + r\mu$$

If a team's score is below μ, their score will increase; if it is above μ, it will decrease. For our tests we chose $r \in$

Algorithm 1 SCOPE

```
1: procedure UPDATEELO($elo_i, series, params$)
2:     $elo_f \leftarrow \{\emptyset\}$
3:     for $s$ in $series$ do:
4:         $elo_f \leftarrow update(elo_i, s, params)$        ▷ Eq. 3/4
5:         return $elo_f$
6: procedure EVALELO($elo_f, series$)
7:     $metrics \leftarrow \{\emptyset\}$
8:     $metrics \leftarrow accuracy(elo_f, series)$        ▷ Eq. 5
9:     $metrics \leftarrow calibration(elo_f, series)$     ▷ Eq. 6
10:    $metrics \leftarrow logLoss(elo_f, series)$         ▷ Eq. 7
11:    return $metrics$
12: procedure CROSSVALIDATE($Data, n$)
13:    $trainingSet \leftarrow \{\emptyset\}$
14:    for $i$ in $n$ do:
15:        $trainingSet \leftarrow \{trainingSet \cup Data[i]\}$
16:        $validation \leftarrow Data[i+1]$
17:        for $tr$ in $trainingSet$ do:
18:            $elo_f \leftarrow UpdateElo(elo_i, tr)$
19:            $metrics \leftarrow EvalElo(elo_f, validation)$
```

$\{0, 0.1, 0.2, 0.3\}$ because any values higher than that would bring the scores so close to the mean that valuable information about the team skill would be lost.

4 Validation

We chose to apply SCOPE to the problem of win prediction for esports, specifically studying the Call of Duty (COD) game franchise, played as an esport in the Call of Duty World League (CWL).

4.1 Selecting a Dataset

In 2018, the CWL was played on Call of Duty: WWII (Sledgehammer Games 2017). Teams competed in separate game modes: *Hardpoint*, *Search and Destroy*, and a third mode that changes based on the season; in the WWII season, it was *Capture the Flag*. These modes encompass a variety of play spaces and elicit distinct strategies. In addition to its popularity, Activision made CWL data publicly available (Shacklette 2018); the dataset includes data for CWL tournaments from late in the year 2017 to present day. In future work, we will to apply SCOPE to other datasets from different esports as they become available.

4.2 Applying SCOPE

After choosing a dataset and initializing scores, we applied SCOPE. The pseudo-code for the procedure is outlined in Algorithm 1. We define $UpdateElo$, a procedure to calculate the next Elo score using Elo's formula with the modifications mentioned above. Next $EvalElo$ will make a prediction given the win probability between the two teams and calculate the metrics we described above. Then the $CrossValidate$ procedure splits the dataset (comprised of series' of games) with the day forward chaining technique (elaborated upon in Section 4.4), then trains the model on the training set and tests it on the block of data for the validation set.

Table 1: Elo Model Parameter Grid Tested in Evaluation

Base K	1	5	10	20	30	40	50
MOV	lin	log	exp	sqrt			
Cutoff	1600	1650	1700	1750			
K Scale	0.1	0.25	0.5	0.75	0.9		
w90	100	200	300	400	500		
Regression	0	0.1	0.2	0.3			

4.3 Selecting Parameter Ranges

Table 1 shows the values that we searched through for optimizing our model. As mentioned, we chose base K values lower than σ such that Elo scores do not grow in a manner that outpaces actual team skill. We chose these functions for MOV to test whether the effect increases or decreases with magnitude of the victory. The cutoff values start at one standard deviation above the population mean (μ) and continue until they are 2.5x, a level that is unlikely to be reached by any one score, effectively nullifying the cutoff at that point. K scale, the amount which K is decreased at the cutoff, varies from 10% to 90% to explore a very drastic, or minimal change. The range for $w90$ translates to different the win probability for different skill discrepancies; low values will lead to larger differences in probability for teams closer in skill. Finally, regression values were bounded at 40% to prevent excessive information loss.

4.4 Cross-Validation

Performing cross-validation on our model will help ensure that we are not over-fitting. Elo uses time series data, so we must use different cross-validation methods than those for batch processing algorithms. For this reason we chose to use nested cross-validation, also known as day forward chaining or rolling-origin evaluation. This has been demonstrated to work well for time series data. (Bergmeir and Benítez 2012).

The idea behind this method is to split the training set t into n ordered folds (t_i) then for each epoch of training we add one of the folds to the data. We evaluate this training set on the subsequent fold v, the validation set. In this case we chose N (the total number of folds) as the number of tournaments in the season. Even though the number of series played in each tournament is not the same, the division of the training set by tournament is a natural splitting point for cross-validation blocks.

We decided to approach this problem using grid search, iterating over many possible parameter combinations to discover the best ones. Although using randomized parameter selection techniques have been shown to be robust and converge, grid search is one of the most widely used techniques and allows us to have a more curated and transparent parameter selection in our model (Huang, Mao, and Liu 2012). After preliminary testing we saw some of the parameters affected the model evaluations more than others. On a practical note, splitting up the parameters into 2 groups of 3 reduced the computation time per run, improving iteration for development of the algorithm.

Table 2: Best Parameters for K Value, Regression, and MOV

Parameters			Metrics		
Regression	Base K	MOV	Accuracy	Calibration	Log Loss
0.20	5	exp	**.680** ± .11	1.02 ± .16	.376 ± .049
0.30	5	exp	.663 ± .12	**.999** ± .15	.410 ± .051
None	50	exp	.629 ± .055	1.445 ± .24	**.094** ± .066

Table 3: Best Parameters for Cutoff, K Scale, and w90

Parameters			Metrics		
Cutoff	K Scale	w90	Accuracy	Calibration	Log Loss
1650	0.10	200	**.684** ± .11	1.01 ± .15	.374 ± .046
1650	0.10	200	.684 ± .11	**1.01** ± .15	.374 ± .046
1650	0.75	100	.662 ± .12	1.17 ± .17	**.253** ± .048

4.5 Model Evaluation Results and Discussion

We choose accuracy, calibration, and log loss as evaluation metrics because they give insight into the model behavior in general and fringe cases. Depending on the modeling goals, it may be useful to optimize for one of these metrics in particular. We ran grid search three times, once to optimize each of the metrics. In Tables 2 and 3, we highlight the maximum values achieved; each table represents a different set of parameters we tuned, split up to reduce computation time.

The best Base K value for accuracy is 5. This is low compared to FiveThirtyEight's MLB, NFL and Lacy's EPL rankings which were between $40 - 60$ (Boice 2018; Lacy 2018); it was the second lowest value we tested. Conversely, the best Base K value for log loss was actually 50, the highest possible Base K value in our model. This finding suggests in reality there are a few teams that are very high skill and a few teams that are very low skill. Since log loss generally tries to avoid choosing wrong, it makes sense that it predicts that good teams will get better when they win. In terms of accuracy, if teams are generally predicted to be closer in skill there is a smaller chance for larger errors and upsets are easier to explain. Regression back to the mean after each tournament removes a fair amount of disparity from the field. This is further confirmed by the large amount of regression present in the highest accuracy and calibration parameters, 20 and 30 percent respectively. In general we are plagued by small sample sizes, especially because many teams only play in a few tournaments. This is something that established leagues like MLB or EPL do not have to deal with.

The exponential scaling for MOV performed the best. This suggests that teams who are able to handily beat their opponents should be rewarded with a larger skill update. From the *exp* equation, a base value of 5 leads to a K value of 32 if a team wins 3-0.

Compared to the case of $K = 1$ without any regression or margin of victory scaling (56% accuracy), we achieved a 12% gain, up to 68% by tuning these parameters. The next set of parameters we validated for seemed less impactful, but important to test for. There was only a 0.4% increase in accuracy and a decrease in log loss and calibration by attenuating K at higher levels and adjusting the $w90$ criterion.

Reducing K above 1650 drove a small increase in accuracy. A commanding victory still had a comparably high amount of influence on a team's skill. $w90$ was held constant at 200 for the first run through of cross-validation, so there was not much change here. This hints that there is enough range in our model to account for differences between teams. If this coefficient was much higher, it would mean that it would take a much larger Elo difference to see a difference in win probability.

5 Conclusion

In this paper we were able to construct an accurate win prediction model for the CWL. SCOPE not only has the potential to be applied to esports, but other sports and games as well. Given the simplicity of this technique, we foresee it being used by analysts to communicate with fans about team skill in esports. We acknowledge that this work is presented as a generalizeable framework and should therefore be demonstrated on multiple esports datasets. Due to space limitations for this paper, we defer that for future work. We have primarily focused on identifying the necessary modeling assumptions needed to extend this framework to other sports and games.

Future directions to improve this model include making more sophisticated updates to Elo. This could be reflected by changing the granularity with which we update Elo, including separate Elo for each game mode or even per player. Another way to update the system would be to change the Elo when a player changes teams. FiveThirtyEight has integrated a component into their MLB Elo prediction system that changes pre-game Elo based on who is pitching (Boice 2018). We could implement a similar system that looks back at how a newly added player performed on their previous team and update their new team's Elo accordingly. FiveThirtyEight's MLB model also takes in to account park effects for different stadiums (Boice 2018). We could integrate the player side selection and map veto process to get a better picture of how map selection would affect Elo in COD, or make similar changes for other games.

When conducting our research into existing skill rating methods, we prioritized model transparency. If players do not understand how their skill ratings are composed, we posit this can cause unnecessary strife in the community. SCOPE should be applied to new esports, or existing ones, to understand the parameters that change in an Elo model, determine the best set of values for those parameters using cross-validation, and evaluate the effectiveness of the model in different ways. All told, SCOPE is a parsimonious model building framework that will enable engineers and data scientists to create better skill-based ranking systems.

Acknowledgments

This work was supported in part by an unrestricted gift grant by Activision Central Tech. We thank Justin Shacklette and Dean Wyatte from Activision for their help and feedback.

References

Bergmeir, C., and Benítez, J. M. 2012. On the use of cross-validation for time series predictor evaluation. *Information Sciences* 191:192–213.

Boice, J. 2018. How Our MLB Predictions Work. https://fivethirtyeight.com/methodology/how-our-mlb-predictions-work/. Last accessed: 2019-03-23.

Bungie. 2004. *Halo 2*. Microsoft Game Studios.

Chen, Z.; Xue, S.; Kolen, J.; Aghdaie, N.; Zaman, K. A.; Sun, Y.; and Seif El-Nasr, M. 2017. Eomm: An engagement optimized matchmaking framework. In *Proceedings of the 26th International Conference on World Wide Web*, 1143–1150.

Cui, A.; Chung, H.; and Hanson-Holtry, N. 2018. OpenDota. https://www.opendota.com. Last checked: 2019-03-23.

Delalleau, Contal, T.-L. F. B. Z. 2012. Beyond skill rating: Advanced matchmaking in ghost recon online. *IEEE Transactions on Computational Intelligence and AI in Games* 4(3):167–177.

Elo, A. 1978. *The Rating of Chessplayers, Past and Present*. Ishi Press International.

Glickman, M. E., and Jones, A. C. 1999. Rating the chess rating system. *Chance* 12:21–28.

Glickman, M. E. 1995. A comprehensive guide to chess ratings. *American Chess Journal* 3(1):59–102.

Herbrich, R.; Minka, T.; and Graepel, T. 2007. TrueSkill™: a Bayesian skill rating system. In *Advances in Neural Information Processing Systems*, 569–576.

Huang, Q.; Mao, J.; and Liu, Y. 2012. An improved grid search algorithm of svr parameters optimization. In *Proceedings of the 14th IEEE International Conference on Communication Technology*, 1022–1026.

Kovalchik, S. A. 2016. Searching for the goat of tennis win prediction. *Journal of Quantitative Analysis in Sports* 12(3):127–138.

Lacy, S. 2018. Implementing an elo rating system for european football. https://stuartlacy.co.uk/2017/08/31/implementing-an-elo-rating-system-for-european-football/. Last accessed: 2019-03-23.

LeBlanc, M. 2006. Tools for creating dramatic game dynamics. In *The game design reader: A rules of play anthology*. Cambridge, MA, USA: MIT Press. 438–459.

Minka, T.; Cleven, R.; and Zaykov, Y. 2018. Trueskill 2: An improved bayesian skill rating system. Technical Report MSR-TR-2018-8, Microsoft.

Morris, B., and Bialik, C. 2015. Serena Williams And The Difference Between All-Time Great And Greatest Of All Time. http://53eig.ht/1UkFK5s. Last accessed: 2019-03-23.

Shacklette, J. 2018. Call of Duty World League Data. https://github.com/Activision/cwl-data. Last checked: 2019-03-23.

Silver, N. 2012. *The Signal and the Noise: Why So Many Predictions Fail–But Some Don't*. New York, New York, USA: Penguin.

Sledgehammer Games. 2017. *Call of Duty: WWII*. Activision.

Thurstone, L. L. 1927. A law of comparative judgment. *Psychological review* 34(4):273.

Ubisoft Singapore. 2014. *Tom Clancy's Ghost Recon Phantoms*. Ubisoft.

Valve Corporation. 2000. *Counter-Strike*. Valve Corporation.

Valve Corporation. 2013. *Dota 2*. Valve Corporation.

Wagner, M. G. 2006. On the Scientific Relevance of eSports. In *Proceedings of the 2006 International Conference on Internet Computing & Conference on Computer Games Development*, 437–442.

Winn, J., and Bishop, C. M. 2019. *Model-Based Machine Learning (Early Access)*. Self-published. http://www.mbmlbook.com/index.html.

Winn, C. 2015. The well-played moba: How dota 2 and league of legends use dramatic dynamics. In *Proceedings of the 2015 DiGRA Conference: Diversity of Play*.