NeuralAC: Learning Cooperation and Competition Effects for Match Outcome Prediction

Yin Gu[1], Qi Liu[1]*, Kai Zhang[1], Zhenya Huang[1], Runze Wu[2], Jianrong Tao[2]

[1] Anhui Province Key Laboratory of Big Data Analysis and Application, School of Data Science & School of Computer Science and Technology, University of Science and Technology of China

[2] Fuxi AI Lab, NetEase Inc., Hangzhou, China

{gy128, kkzhang0808}@mail.ustc.edu.cn, {qiliuql, huangzhy}@ustc.edu.cn, {wurunze1, hztaojianrong}@corp.netease.com

Abstract

Match outcome prediction in group comparison setting is a challenging but important task. Existing works mainly focus on learning individual effects or mining limited interactions between teammates, which is not sufficient for capturing complex interactions between teammates as well as between opponents. Besides, the importance of interacting with different characters is still largely underexplored. To this end, we propose a novel ***Neural A****ttentional* ***C****ooperation-competition model* (***NeuralAC***), which incorporates weighted-cooperation effects (i.e., intra-team interactions) and weighted-competition effects (i.e., inter-team interactions) for predicting match outcomes. Specifically, we first project individuals to latent vectors and learn complex interactions through deep neural networks. Then, we design two novel attention-based mechanisms to capture the importance of intra-team and inter-team interactions, which enhance NeuralAC with both accuracy and interpretability. Furthermore, we demonstrate NeuralAC can generalize several previous works. To evaluate the performances of NeuralAC, we conduct extensive experiments on four E-sports datasets. The experimental results clearly verify the effectiveness of NeuralAC compared with several state-of-the-art methods.

Introduction

Group comparison, usually involving two teams competing with each other (e.g., Figure 1), is ubiquitous in sports and online games, such as football, *Dota2*, and *League of Legends*. In the last decade, the popularity of online competitive games has exploded and there are more than 800 million online game players. A large number of players create great commercial value coupled with some technical challenges. One of the crucial problems, i.e., match outcome prediction, has attracted considerable research attention since it plays a key role in creating fair matches for players and increasing the teams' probability of winning (Chen et al. 2018).

In the literature, many existing methods in group comparison (Herbrich, Minka, and Graepel 2007; Huang, Lin, and Weng 2008) focus on learning individual effects from outcomes of group comparisons. Despite the popularity of these methods, they omit interplays between players within a team. In their assumption, team members are independent of each other, that is, a team's ability is modeled as the sum of the team members' score. To tackle this limitation, neural network-based methods (Delalleau et al. 2012; Gong et al. 2020) were proposed to capture the intra-team interactions. However, they focus on obtaining the team representation by aggregating single team member's representations, which preserves little low-level information, and thus it is hard to evaluate individuals' contributions to the teamwork. Meanwhile, factorization machines (FM) (Rendle 2010) was adopted to group comparison (Li et al. 2018a), where the cooperation effect (i.e., intra-team interaction) was modeled as the inner product of two latent vectors. Though low-level information was preserved, they can not model non-linear interactions due to the limitation of FM method.

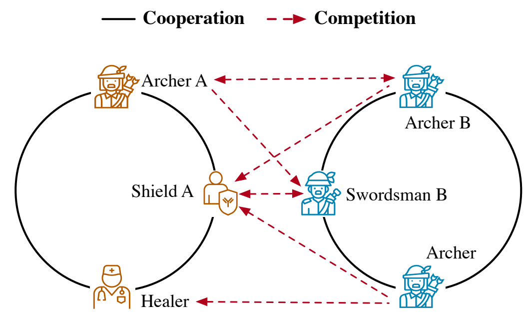


Figure 1: An example of group comparisons.

Indeed, both cooperation and competition are very common in human society (Bengtsson and Kock 1999; Bar-Yam 2003; Tauer and Harackiewicz 2004; MacRae 2018), which can be highly complex. For example, as shown in Figure 1, two teams fight each other in group comparison (e.g., battlefield), which involves multiple interactions, including intra-team interactions (e.g., the shield soldier A protects teammates, the healer cures teammates), and inter-team interactions (e.g., the archer A shoots the swordsman B, the shield soldier resists the swordsman's attack). Everyone on the battle has different strengths and weaknesses, making them perform differently when against different opponents. Meanwhile, teammates could complement each other through cooperation, which makes the group comparison highly intri-

*Corresponding Author.

cate. To make the prediction accurate, it's necessary to incorporate comprehensive interactions. Nevertheless, how to model such cooperation and competition effects simultaneously remains a challenge.

Another limitation of existing approaches is that they do not consider the importance of interactions. Considering an ancient war where two armies fight each other, soldiers focus on finding opportunities to kill enemy generals. Meanwhile, soldiers try to protect generals on their side. It's clear that generals play a key role in wars, and interacting with generals has a larger influence on the outcome of the war. Therefore, in this group comparison scenario, generals should receive more attention. In other words, interactions with different characters have different attention scores, as they contribute differently to the match outcome. Thus, modeling inter-team and intra-team attention distributions is a nontrivial task and yet remains a challenge.

To address the above challenges, in this paper, we propose a novel ***Neural A****ttentional* ***C****ooperation-competition model* (***NeuralAC***), which incorporates weighted-cooperation effects and weighted-competition effects for predicting match outcomes. Different from previous approaches, we choose the element-wise product of two latent vectors as the input of deep neural networks (DNNs) to get the corresponding score, which greatly facilitates deep layers to learn meaningful second-order interactions, while still preserving its interpretability. First, by deploying DNNs on both intra-team and inter-team interactions, we get pairwise cooperation scores and pairwise competition scores. Then, we design two attention mechanisms to capture the importance of intra-team and inter-team interactions, which enhance NeuralAC with both accuracy and interpretability. Furthermore, we demonstrate NeuralAC is general and expressive, which can generalize several previous works. The main contributions of this work are as follows:

- We consider both intra-team interactions and inter-team interactions, and we propose to model comprehensive interactions with neural networks for learning complex cooperation and competition effects.
- We further propose two attention mechanisms to enhance NeuralAC, which provide strong interpretability about the importances of interactions.
- Extensive experiments on four real E-sports datasets show the effectiveness of NeuralAC. The code and datasets are avaliable at https://github.com/bigdata-ustc/NAC.

Related Work

Group Comparison

Many existing works (Herbrich, Minka, and Graepel 2007; Huang, Lin, and Weng 2008) in this area focus on learning individual effects from group comparison. They assume the player's performance is independent of teammates, and the ability of the team is represented as the summation of the team members' scores. This assumption may not hold true in the real world, because some players may perform well when they team-up together. To address this limitation, some methods (DeLong et al. 2011; Semenov et al. 2016; Li et al. 2018a) are proposed to model the cooperation effects in the team composition. For instance, Li et al.(2018a) exploited factorization machine to model interplay between teammates. Their methods may not be expressive enough due to intra-team interactions is modeled in a linear way. Deep learning is also adopted (Gong et al. 2020; Delalleau et al. 2012) to capture intra-team interactions. Gong et al.(2020) also proposed a novel technique of learning the representations of individuals from relation graphs. However, these works mainly utilize DNNs for aggregating players' representations to obtain team representations. Despite non-linear interactions is modeled, their methods capture limited information at the low level. Besides, due to the inherent traits of DNNs, these methods lack interpretability and it's hard to assess individuals' contributions to team works.

The existing works either focus on learning individual effects or modeling limited cooperation effects. Besides, competition effects and the importance of interactions are still largely under-explored. Meanwhile, some methods (Delalleau et al. 2012; Minka, Cleven, and Zaykov 2018; Gong et al. 2020) utilized in-game features to get more accurate predictions. However, those features are usually designed by experts in the domain, hence case-specific. We focus on a more general task with no domain knowledge required. Therefore, we don't utilize any in-game features.

Cooperation and Competition

Cooperation and competition are important factors in other fields, which are widely studied. For example, Dai et al. (2020) predicted cooperation and competition relationships among companies in a company relation network. Usmani et al. (2020) analyzed the competitiveness of commercial products in the market. Some works (Lowe et al. 2017; Wray, Kumar, and Zilberstein 2018) explored to model decisions making process in the multi-agent cooperative-competitive environment. Although cooperation and competition have been studied in other fields, very little work in group comparison has fully explored the impact of cooperation and competition effects.

NeuralAC Model

In this section, we first formally introduce match outcome prediction task. Then, we give an overview of NeuralAC. After that, we details basic NeuralAC and attention mechanisms. Finally, we demonstrate the generality of NeuralAC.

Problem Definition

Suppose there are n individuals $\{1, 2, ..., n\}$, M observable matches. Each match involves two teams T_A and T_B, each of them is a subset of $\{1, 2, ..., n\}$, and the match outcomes of the M matches is denoted as $\{y_1, y_2, ..., y_M\}$. In this paper, we focus on the problem of binary match outcome prediction, each match outcome is either win or lose. We assume that there is no draw. Let $y_m = 1$ if T_A beat T_B in a match $m \in [1, M]$, otherwise $y_m = 0$. Given a match between T_A and T_B, our goal is to predict the match outcome $\hat{y} \in [0, 1]$.

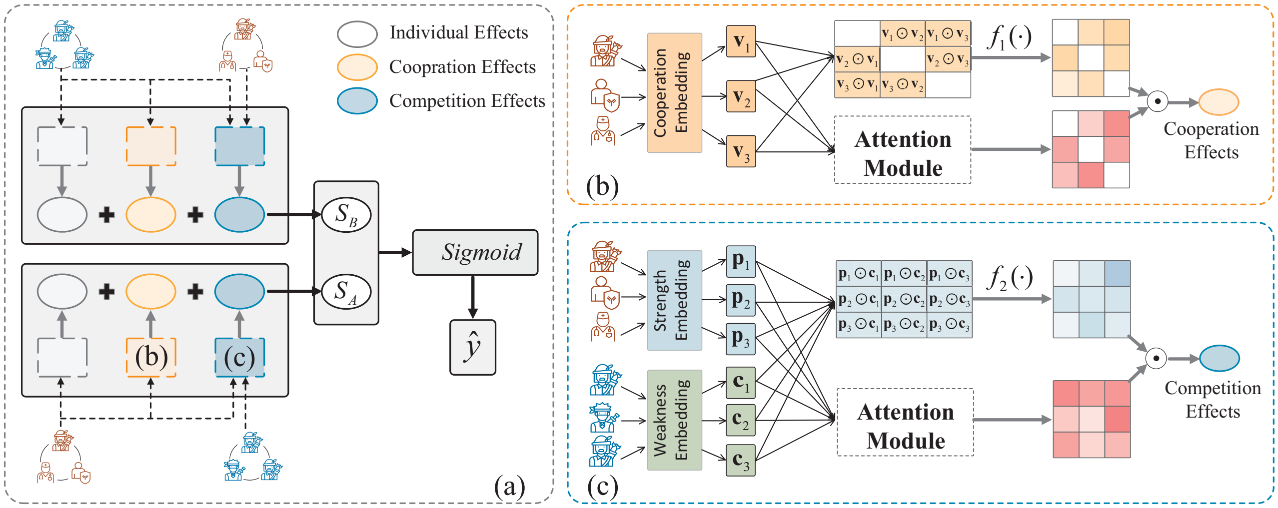


Figure 2: NeauralAC model architecture. (a) shows the overview of the model; (b) is cooperation effects part; (c) is competition effects part. Note that, for clarity purpose, we omit the individual effects part in this figure.

Model Overview

In this paper, we focus on group comparison. We assume each team has a score indicating team's ability. Inspired by Blade-Chest (Chen and Joachims 2016), we formulate the probability of team T_A defeating team T_B as:

$$\begin{aligned} P(A \: beats \: B) &= \frac{\exp(S_A)}{\exp(S_A)+\exp(S_B)}, \\ &= \frac{1}{1+\exp(-(S_A - S_B))}, \\ &= \sigma(\Delta(A,B)), \end{aligned} \tag{1}$$

where S_A is T_A's score that represents the overall ability of the team, σ is the sigmoid function. $\Delta(A, B)$ denotes the edge that T_A have when match up against T_B. When $\Delta(A, B) \rightarrow 0$, both teams have the equal odds to win. When $\Delta(A, B) \rightarrow +\infty$, $P(A$ beats $B) \rightarrow 1$, B almost has no chance to defeat A, and vice versa.

As mentioned above, there are multiple complex interactions in group comparisons (e.g., cooperation between teammates and competition between opponents). Generally, if team members get high individual ability or two members cooperate well, the overall ability of the team can be improved. Besides, if a player in T_A has more advantages when competing with his opponents, then the overall ability of T_A can be further improved. Therefore, in NeuralAC, the overall ability of the team consists of three parts: individual effects, cooperation effects and competition effects. Take T_A versus T_B as an example, we formulate T_A's score as:

$$S_A = \sum_{i \in T_A} w_i + F_{\text{coop}}(T_A) + F_{\text{comp}}(T_A, T_B), \tag{2}$$

where w_i indicates i's individual ability, which is model parameter. The first term of S_A models individual effects. The second term $F_{\text{coop}}(T_A)$ and the third term $F_{\text{comp}}(T_A, T_B)$ models cooperation effects and competition effects, respectively. Figure 2 shows the framework and two main components of NeuralAC.

Basic NeuralAC

In this subsection, we illustrate the detail of NeuralAC without two attention mechanisms (e.g., cooperation effects part, competition effects part).

Cooperation Effects. Since everyone has different cooperation characteristics, the cooperation effect usually differs when working with different teammates. Inspired by (Li et al. 2018a; Gong et al. 2020), in NeuralAC, we assume each individual i has an embedding vector $\mathbf{v}_i \in \mathbb{R}^k$, namely cooperation vector, representing his cooperation characteristics. The cooperation effects $F_{\text{coop}}(T_A)$ is formulated as:

$$F_{\text{coop}}(T_A) = \sum_{i \in T_A} \sum_{j \in T_A, i \neq j} f_1(\mathbf{v}_i \odot \mathbf{v}_j), \tag{3}$$

where $\odot$ denotes the element-wise product, $\mathbf{v}_i$ and $\mathbf{v}_j$ are learnable parameters, f_1 refers to the MLP with non-linear activation function, which are capable of learning higher-order and non-linear interactions between teammtes. The output of $f_1(\mathbf{v}_i \odot \mathbf{v}_j)$ is a scalar value, which is the cooperation score between i and j.

Competition Effects. When a player attack another, the competition result depends on the offensive's strength and the defensive's weakness, and vice versa. Inspired by Blade-Chest, in NeuralAC, each individual i has two distinctive embedding vectors $\mathbf{p}_i \in \mathbb{R}^k$, $\mathbf{c}_i \in \mathbb{R}^k$, namely strength vector and weakness vector, respectively. To simplify the setting, we assume $\mathbf{v}_i$, $\mathbf{p}_i$, $\mathbf{c}_i$ share the same size k. Then, the competition effects $F_{\text{comp}}(T_A, T_B)$ is formulated as:

$$F_{\text{comp}}(T_A, T_B) = \sum_{i \in T_A} \sum_{j \in T_B} f_2(\mathbf{p}_i \odot \mathbf{c}_j), \tag{4}$$

where $\odot$ denotes the element-wise product, $\mathbf{p}_j$ and $\mathbf{c}_i$ are learnable parameters, f_2 refers to a MLP with non-linear activation function, which can model non-linear interactions between opponents. The output of $f_2(\mathbf{p}_i \odot \mathbf{c}_j)$ is a scalar value, indicating a competition score when i compete against j. To summarize, we give the formulation of S_A as:

$$\begin{aligned} S_A = &\sum_{i \in T_A} w_i + \sum_{i \in T_A} \sum_{j \in T_A, i \neq j} f_1(\mathbf{v}_i \odot \mathbf{v}_j) \\ &+ \sum_{i \in T_A} \sum_{j \in T_B} f_2(\mathbf{p}_i \odot \mathbf{c}_j). \end{aligned} \tag{5}$$

DNNs Components. In our setting, f_1 and f_2 share the same network structure. Here, we elaborate the design of f_2. To simplify the description, we denote $\mathbf{p}_i \odot \mathbf{c}_j$ as $\mathbf{x}$, and then feed $\mathbf{x}$ to the MLP. Similar to NFM (He and Chua 2017), the process can be formulated as:

$$\begin{aligned} \mathbf{z}_1 &= \sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1), \\ \mathbf{z}_2 &= \sigma(\mathbf{W}_2 \mathbf{z}_1 + \mathbf{b}_2), \\ &\cdots \qquad \cdots \\ \mathbf{z}_L &= \sigma(\mathbf{W}_L \mathbf{z}_{L-1} + \mathbf{b}_L), \end{aligned} \tag{6}$$

where L indicates the number of hidden layers, $\mathbf{W}_L$, $\mathbf{b}_L$ denote the weight matrix and bias for the L-th layer. $\sigma(\cdot)$ is $ReLU$ activation function.

To ensure the interpretability of NeuralAC (e.g. it does not make sense when cooperation and competition scores are negative), we set the activation function of the output layer to be $ReLU$.:

$$O_{ij} = ReLU(\mathbf{W}_o \mathbf{z}_L + \mathbf{b}_o), \tag{7}$$

where O_{ij} is the competition score when i against j.

NeuralAC

In this subsection, we show how to enhance basic NeuralAC with two attention mechanisms. Attention mechanisms have been widely used in many tasks, such as computer vision (Chen et al. 2017; Liu et al. 2018), natural language processing (Zhang et al. 2019), and recommendation system (Xiao et al. 2017; Li et al. 2018b). In a team competition case, we often pay more attention to the key person in our team. Similarly, we usually focus on the key person in the opponent team and look for a chance to defeat him. Since cooperating or competing with the key person has a greater influence on the match outcome, not all interactions should share the same weight as they contribute differently to the final game outcome. Motivated by this intuition, we propose to deploy the attention modules on cooperation effects and competition effects as:

$$\begin{aligned} S_A = &\sum_{i \in T_A} w_i + \sum_{i \in T_A} \sum_{j \in T_A, j \neq i} a_{ij}^{\text{coop}} f_1(\mathbf{v}_i \odot \mathbf{v}_j) \\ &+ \sum_{i \in T_A} \sum_{j \in T_B} a_{ij}^{\text{comp}} f_2(\mathbf{p}_i \odot \mathbf{c}_j), \end{aligned} \tag{8}$$

where a_{ij}^{coop}, a_{ij}^{comp} is intra-team and inter-team attention score respectively, which can be interpreted as the importance of the interaction in contributing to the game outcome.

Attention Components. To make attention modules generalized to unseen pairs and asymmetry (e.g., i's attention to j is usually different from j's attention to i), we formulate them as follows:

$$\begin{aligned} r_{ij}^{\text{coop}} &= \mathbf{v}_i^T \mathbf{W}_{\text{coop}} \mathbf{v}_j, \\ a_{ij}^{\text{coop}} &= \frac{\exp(r_{ij}^{\text{coop}})}{\sum_{j \in T_A, j \neq i} \exp(r_{ij}^{\text{coop}})}, \end{aligned} \tag{9}$$

$$\begin{aligned} r_{ij}^{\text{comp}} &= \mathbf{p}_i^T \mathbf{W}_{\text{comp}} \mathbf{c}_j, \\ a_{ij}^{\text{comp}} &= \frac{\exp(r_{ij}^{\text{comp}})}{\sum_{j \in T_B} \exp(r_{ij}^{\text{comp}})}, \end{aligned} \tag{10}$$

where $\mathbf{W}_{\text{coop}} \in \mathbb{R}^{k \times k}$, $\mathbf{W}_{\text{comp}} \in \mathbb{R}^{k \times k}$ are learnable model parameters , r_{ij}^{coop} and r_{ij}^{comp} denote attention values. The inputs to r_{ij}^{coop} are two teammates' cooperation vectors (e.g., $\mathbf{v}_i$ and $\mathbf{v}_j$), and the inputs to r_{ij}^{comp} are one's strength vector (e.g., $\mathbf{p}_i$) and his opponent's weakness vector (e.g., $\mathbf{c}_i$). The higher the r_{ij}^{coop}, the more attention j will receive from his teammate i. The higher the r_{ij}^{comp}, the more attention j will receive from his opponent i.

Training Strategy

Given M observed matches, let y_i denote the i-th match outcome, $\hat{y}_i$ denote corresponding prediction (i.e., $P(A\ beats\ B)$). The loss function is cross entropy between model output $\hat{y}$ and true label y:

$$\mathcal{L} = -\sum_{i=1}^{M} (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)). \tag{11}$$

In this way, we can learn NeuralAC by directly minimizing the loss function $\mathcal{L}$.

Generality of NeuralAC

In this subsection, we demonstrate the generality of NeuralAC, and pervious works can be seen as special cases of NeuralAC. To be specific, we first simplify NeuralAC by removing attention modules and non-linear activation function, then set hidden layer number to 0. In this way, T_A's score is formulated as:

$$\begin{aligned} S_A = &\sum_{i \in T_A} w_i + \sum_{i \in T_A} \sum_{j \in T_A, i \neq j} \mathbf{h}_1^T (\mathbf{v}_i \odot \mathbf{v}_j) \\ &+ \sum_{i \in T_A} \sum_{j \in T_B} \mathbf{h}_2^T (\mathbf{p}_i \odot \mathbf{c}_j), \end{aligned} \tag{12}$$

where vector $\mathbf{h}_1 \in \mathbb{R}^k$, $\mathbf{h}_2 \in \mathbb{R}^k$ denotes neuron weights of the output laryer.

Generalized Bradly-Terry. Generalized BT (Huang, Lin, and Weng 2008) consider individual effect, while neglecting cooperation effect and competition effect. By fixing $\mathbf{h}_1$ and $\mathbf{h}_2$ to constant zero vectors, we can get the Generalized BT model, where S_A is defined as:

$$S_A = \sum_{i \in T_A} w_i. \tag{13}$$

Factorization Machine. FM (Li et al. 2018a) models pairwise intra-team interactions by inner product of two latent vectors, the team score can be represented as:

$$S_A = \sum_{i \in T_A} w_i + \sum_{i \in T_A} \sum_{j \in T_A, i \neq j} \mathbf{v}_i^T \mathbf{v}_j. \quad (14)$$

By forcing $\mathbf{h}_2$ to be a constant zero vector, and fixing $\mathbf{h}_1$ to be constant one vector, we can get the FM model exactly.

Blade-Chest-Inner. Blade-Chest (Chen and Joachims 2016) model each player i with an absolute ability value w_i, strength vector $\mathbf{p}_i$, and weakness vector $\mathbf{c}_i$. Blade-Chest consider interaction between a opponent, but it is designed for 1v1 circumstance. In Blade-Chest-Inner model, take player a versus player b as an example, a's score is modeled as:

$$S_a = w_a + \mathbf{p}_a^T \mathbf{c}_b. \quad (15)$$

By setting the team size of both sides to 1 and let $\mathbf{h}_2$ to be constant one vector, our model can be reduced to the following formula, which is the same with Blade-Chest model:

$$S_A = \sum_{i \in \{a\}} w_i + 0 + \sum_{i \in \{a\}} \sum_{j \in \{b\}} \mathbf{p}_i^T \mathbf{c}_j. \quad (16)$$

Experiments

Dataset Description

Online games are an ideal testbed and can provide a lot of group comparison data. We use four E-sports datasets to evaluate the utility of our model. The basic statistics of all the datasets are summarized in Table 1.

Dota2 is a famous Multiplayer Online Battle Arena (MOBA) game. In each game, two teams fight each other on the map. Each player controls a different virtual character named hero throughout the whole game. We downloaded ranked matches from yasp.co[1] and Varena[2], which were played in the years of 2015 and 2018 respectively.

League of Legends (LOL) shares a similar game pattern as Dota2. We crawled the recent matches from RiotGame[3]. For *LOL* dataset, we investigate a special game mode, where players are forced to fight on one single line, instead of three lines in Dota2.

We filter out matches that played less than 15 minutes for Dota2 and 8 minutes for LoL. Due to the existence of matchmaking systems, players on both sides have relatively close skills. Therefore, we treat each hero as an individual.

Teamfight Tactics (TFT) is a round-based strategy game that players compete against seven other opponents by constructing and optimizing team compositions to be the last one standing. A Team is composed of heroes selected by the player, each hero has different synergies and equipment. Different from MOBA games, players do not have control of the deployed heroes during the combat time. In each round, the player will fight against a random opponents. We crawl the ranked TFT match records via RiotGame API. We sample group comparisons according to players' last survival round.

[1] https://github.com/odota/core/wiki/JSON-Data-Dump

[2] https://open.varena.com/documentation/dota2/

[3] https://developer.riotgames.com/apis#match-v4

Dataset	Matches	#Heroes	Mode
Dota2015	800,000	110	5v5
Dota2018	580,270	116	5v5
LoL	754,700	148	5v5
TFT	800,000	188	N1vN2

Table 1: Statistics of the datasets.

Baseline Methods

- Logistic Regression (LR) (Ng and Jordan 2002): A linear classifier with L2 regularization. We use the same data input format as Semenov et al. (2016).
- Generalized Bradly-Terry (BT) (Huang, Lin, and Weng 2008): Another linear model, which consider only individual effects.
- TrueSkill (Herbrich, Minka, and Graepel 2007): An algorithm based on probability graph, which is widely used in online games for matchmaking.
- LightGBM (LGB) (Ke et al. 2017) : A highly efficient implementation of GBDT, which achieve state of the art performance in many data science competitions.
- HOI (Li et al. 2018a): A factorization machines (FM) (Rendle 2010) based model that takes pair-wise interactions of teammates into account.
- OptMatch (Gong et al. 2020): A method based on multi-head self-attention (Vaswani et al. 2017), where each hero has own embeddings and feed into the module to get the team representation for predicting match outcomes. Since we don't utilize any in-game feature except hero IDs, we remove the feature module of OptMatch. Besides, OptMatch assumes teams on two sides have the same size, therefore, we don't apply OptMatch on TFT dataset.

Model Variants

To examine the effectiveness of each component in NeuralAC, we conducted a series of ablation experiments.

- *no-coop*: A variant of NeuralAC that does not model the cooperation effect, i.e., remove f_1.
- *no-comp*: A variant of NeuralAC that does not model the competition effect, i.e., remove f_2.
- *no-att*: A variant of NeuralAC that all attention modules are removed, i.e., remove a_{ij}^{coop} and a_{ij}^{comp}.

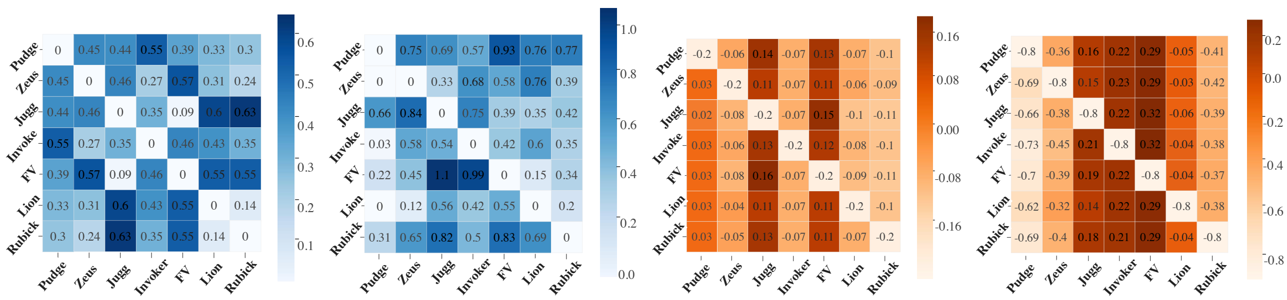
Experimental Setup

For NeuralAC model, the dimension of hidden layers is set to 50, and ReLu is used as activation function. We initialize the parameter with *Kaiming* initialization (He et al. 2015). Besides, Dropout (Srivastava et al. 2014) technique is also applied with the drop probability set to 0.2.

For every dataset, we randomly divided samples into 80% for training, 10% for validating, and 10% for testing. We choose Area Under ROC (AUC) (Bradley 1997) and Accuracy (Acc) as the evaluation metrics. For HOI and

Model	Dota2015		Dota2018		LoL		TFT	
	AUC	Acc	AUC	Acc	AUC	Acc	AUC	Acc
BT	0.6330	0.5955	0.6116	0.5784	0.6347	0.5969	0.7634	0.6935
LR	0.6330	0.5956	0.6116	0.5784	0.6347	0.5969	0.7634	0.6935
TrueSkill	0.6110	0.5789	0.5805	0.5577	0.6129	0.5811	0.7506	0.6832
LGB	0.6445	0.6035	0.6224	0.5929	0.6411	0.6028	0.8015*	0.7234*
HOI	0.6373	0.5989	0.6144	0.5821	0.6337	0.5965	0.7728	0.6989
OptMatch	0.6325	0.5961	0.6173	0.5851	0.6523	0.6101	-	-
NeuralAC	**0.6615**	**0.6156**	**0.6411**	**0.6012**	**0.6663**	**0.6209**	**0.8082**	**0.7279**
no-coop	0.6525	0.6086	0.6333	0.5951	0.6531	0.6110	0.7992	0.7215
no-comp	0.6444	0.6051	0.6203	0.5841	0.6546*	0.6115*	0.7740	0.7000
no-att	0.6606*	0.6150*	0.6396*	0.5991*	0.6480	0.6070	0.7780	0.7037

Table 2: Experimental results on match outcome prediction. (The second best methods are denoted with *)



(a) Cooperation scores. (b) Competition scores. (c) Cooperation attention values. (d) Competition attention values.

Figure 3: Note that the cooperation scores matrix is symmetry, the other three matrices are asymmetric since the impact of hero i on hero j is usually different from the impact of hero j on hero i. The diagonal is left blank because one cannot interact with himself. Take subfigure (b) for illustration, the value in i-th row and j-th column, denote the competition score when hero in i-th row competes against hero in j-th column.

NeuralAC, we set embedding size k to 20. As for OptMatch, the embedding size was searched in [20, 40, 80, 160, 320] to get the best performance on validating. We choose Adam (Kingma and Ba 2014) as the optimizer, with 0.001 of learning rate and 0.0001 of weight decay coefficient, for HOI, OptMatch and NeuralAC. Besides, the batch size is set to 256 for HOI, OptMatch and NeuralAC on all datasets.

LR, TrueSkill and LGB are implemented by open source packages sklearn, trueskill, LightGBM, respectively. HOI, NeuralAC and OptMatch are implemented by PyTorch package (Paszke et al. 2019). All experiments are implemented by Python and are trained on a Linux server with Intel Xeon E5-2650 CPUs and a TITAN Xp GPU.

Experimental Results

Table 2 shows the experimental results of all methods on the prediction task. First, NeuralAC outperforms all the other baselines on all datasets, indicating the effectiveness of our model. Second, by incorporating cooperation effects, HOI performs better than BT, LR and Trueskill on most datasets. This proves the existence of the cooperative effect in group comparison. Third, no-comp performs better than HOI, which indicates that the inner product may fail to model complex intra-team interactions. Fourth, no-comp outperforms OptMatch in 2 out of 3 datasets. One possible reason may be that despite OptMatch modeling higher-order interactions, it preserves little low-level information. Finally, compared with other variants, NeuralAC performs better, which suggests that incorporating competition effects, and attention mechanisms improves the accuracy of prediction.

Model Interpretability

To evaluate the interpretability of NeuralAC (i.e., whether the cooperation effects, competition effects and attention distribution are reasonable), we choose the most 7 popular heroes in Dota2018, then calculate their pair-wise cooperation scores, competition scores, and attention values separately. The corresponding results are shown in Figure 3.

Cooperation Score. Intuitively, if two individuals i and j perform better when they play together, they are more likely to get a higher cooperation score. Similarly, if i suppress j more when j is i's opponent, i is more likely to get a

	indivi. effects	coop. effects	comp. effects
T_A	0.5284	1.9886	3.5600 ($T_A \rightarrow T_B$)
T_B	0.5494	2.0271	2.2458 ($T_B \rightarrow T_A$)

Table 3: Effects of two teams.

T_A's competition scores			T_B's competition scores		
T_A	T_B	value	T_B	T_A	value
Spectre	Riki	1.5	Phoenix	Spectre	0.82
Spectre	Luna	1.1	Riki	Spectre	0.75
Pudge	Pugna	1.1	Pugna	Spectre	0.64
	...			...	
Spectre	Phoenix	0.26	Luna	Zeus	0
Zeus	Pugna	0.2	Skywrath	Pudge	0

Table 4: Competition scores of two teams.

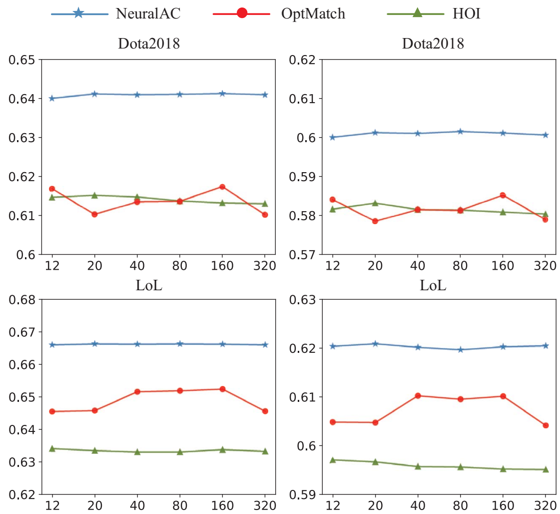


Figure 4: Test Acc and AUC w.r.t. embedding size k.

higher competition score over j. As shown in Figure 3(a), Jugg or FV get a high score when they play with Lion or Rubick. One likely explanation may be that both Jugg and FV are melee Damage Per Second (DPS) heroes, which means they can deal huge physical damage to an enemy in a short time, but they have a small attack range; Lion and Rubick are ranged wizards with stun spells, which means they can help Jugg or FV get close to their enemy, hence improve attack efficiency. Two different types of heroes can complement each other. Therefore, Jugg cooperates well with Rubick or Lion. Furthermore, we can observe that the cooperation score between Jugg and FV is extremely low, which validates our assumption from another aspect because heroes of the same type cooperate poorly.

Competition Score. In the first columns of Figure 3(b), some values are relatively low, which suggests that Pudge almost immune from Zeus, Invoker, and Lion. Because Zeus, Invoker, and Lion rely heavily on magic spells while Pudge has a high magic resistance.

Attention Distribution. As shown in Figure 3(c) and Figure 3(d), FV and Jugg get relatively high attention values in two figures, because they are key characters in the game. One strange finding is Invoker get low scores in Figure 3(c), but high scores in Figure 3(d). A possible reason for this abnormality is Invoker has 10 unique spells, which make him one of the most powerful heroes in dota2. Unlike FV or Jugg, even without the assists of teammates, Invoker could still play an role in the battle. In the first column of Figure 3(d), Pudge get lowest attention values from enemy, since Pudge is one of the strongest heroes with high health, making him the last person his opponent wants to attack.

Through the above analysis, we can infer that NeuralAC indeed learns meaningful and reasonable relationships between heroes. It is worth to point out that our proposed model is capable to learn such complex relationships merely based on game outcomes, with no prior knowledge (e.g., attack range) of heroes[4] is provided to NeuralAC.

Case Study

Here we present a match record on dataset Dota2018, where T_A beats T_B in the end. As shown in Table 3, T_A and T_B get quite close individual effects and cooperation effects. However, the difference between their competition effects is huge. In Table 4, We detail the three highest competition scores and the two lowest competition scores for two teams. Overall, we can observe heroes in T_A have more edges when competing against heroes in T_B. If a method fail to model competition effects, then it may give a wrong prediction.

Hyperparameters Effects

Since HOI, OptMatch, and NeuralAC have embedding layers, we conduct a group of experiments on Dota2018 and LoL to explore the impact of the embedding size, where other parameters (e.g., batch size, learning rate) are fixed. As shown in Figure 4, it is obvious that NeuralAC consistently performs the best under all parameter settings. However, the model does not learn better when embedding size increases.

Conclusions

In this paper, we proposed NeuralAC for match outcomes prediction. By modeling both attentional cooperation effects and attentional competition effects with neural networks, NeuralAC outperforms the state-of-the-art methods. Extensive experimental results on four datasets showed the effectiveness of NeuralAC. Besides, we demonstrated that NeuralAC could be seen as the generalization of several previous models. Finally, NeuralAC provides meaningful and reasonable relationships between individuals, which can be further used in team formations (Wright and Vorobeychik 2015), hero recommendation, and user performance prediction (Huang et al. 2020; Wu et al. 2020; Wang et al. 2020).

[4]To know more about heroes in Dota2, you can refer to the following websites: https://dota2.gamepedia.com/Heroes, https://www.dota2.com/heroes/?l=english

Acknowledgements

We would like to thank RiotGame for providing convenient API. We also want to thank Linxia Gong for her advice on this research. This research was partially supported by grants from the National Key Research and Development Program of China (No. 2018YFC0832101), the National Natural Science Foundation of China (Grants No. 61922073 and 61672483), and the Foundation of State Key Laboratory of Cognitive Intelligence (Grant No. iED2020-M004).

References

Bar-Yam, Y. 2003. Complex systems and sports: complex systems insights to building effective teams. *Cambridge (MA): NECSI* .

Bengtsson, M.; and Kock, S. 1999. Cooperation and competition in relationships between competitors in business networks. *Journal of business & industrial marketing* .

Bradley, A. P. 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition* 30(7): 1145–1159.

Chen, L.; Zhang, H.; Xiao, J.; Nie, L.; Shao, J.; Liu, W.; and Chua, T.-S. 2017. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of CVPR*, 5659–5667.

Chen, S.; and Joachims, T. 2016. Modeling intransitivity in matchup and comparison data. In *Proceedings of WSDM*, 227–236.

Chen, Z.; Nguyen, T.-H. D.; Xu, Y.; Amato, C.; Cooper, S.; Sun, Y.; and El-Nasr, M. S. 2018. The art of drafting: a team-oriented hero recommendation system for multiplayer online battle arena games. In *Proceedings of the 12th ACM Conference on Recommender Systems*, 200–208.

Dai, L.; Yin, Y.; Qin, C.; Xu, T.; He, X.; Chen, E.; and Xiong, H. 2020. Enterprise Cooperation and Competition Analysis with a Sign-Oriented Preference Network. In *Proceedings of SIGKDD*, 774–782.

Delalleau, O.; Contal, E.; Thibodeau-Laufer, E.; Ferrari, R. C.; Bengio, Y.; and Zhang, F. 2012. Beyond skill rating: Advanced matchmaking in ghost recon online. *IEEE Transactions on Computational Intelligence and AI in Games* 4(3): 167–177.

DeLong, C.; Pathak, N.; Erickson, K.; Perrino, E.; Shim, K.; and Srivastava, J. 2011. TeamSkill: modeling team chemistry in online multi-player games. In *PAKDD*, 519–531. Springer.

Gong, L.; Feng, X.; Ye, D.; Li, H.; Wu, R.; Tao, J.; Fan, C.; and Cui, P. 2020. OptMatch: Optimized Matchmaking via Modeling the High-Order Interactions on the Arena. In *Proceedings of SIGKDD*, 2300–2310.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

He, X.; and Chua, T.-S. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 355–364.

Herbrich, R.; Minka, T.; and Graepel, T. 2007. TrueSkill: a Bayesian skill rating system. In *NeurIPS*, 569–576.

Huang, T.-K.; Lin, C.-J.; and Weng, R. C. 2008. Ranking individuals by group comparisons. *Journal of Machine Learning Research* 9(Oct): 2187–2216.

Huang, Z.; Liu, Q.; Chen, Y.; Wu, L.; Xiao, K.; Chen, E.; Ma, H.; and Hu, G. 2020. Learning or Forgetting? A Dynamic Approach for Tracking the Knowledge Proficiency of Students. *ACM Transactions on Information Systems (TOIS)* 38(2): 1–33.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *NeurIPS*, 3146–3154.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Li, Y.; Cheng, M.; Fujii, K.; Hsieh, F.; and Hsieh, C.-J. 2018a. Learning from group comparisons: exploiting higher order interactions. In *NeurIPS*, 4981–4990.

Li, Z.; Zhao, H.; Liu, Q.; Huang, Z.; Mei, T.; and Chen, E. 2018b. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of SIGKDD*, 1734–1743.

Liu, Q.; Huang, Z.; Huang, Z.; Liu, C.; Chen, E.; Su, Y.; and Hu, G. 2018. Finding similar exercises in online education systems. In *Proceedings of SIGKDD*, 1821–1830.

Lowe, R.; Wu, Y. I.; Tamar, A.; Harb, J.; Abbeel, O. P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *NeurIPS*, 6379–6390.

MacRae, S. A. 2018. Competition, cooperation, and an adversarial model of sport. *Journal of the Philosophy of Sport* 45(1): 53–67.

Minka, T.; Cleven, R.; and Zaykov, Y. 2018. Trueskill 2: An improved bayesian skill rating system. *Tech. Rep.* .

Ng, A. Y.; and Jordan, M. I. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *NeurIPS*, 841–848.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 8026–8037.

Rendle, S. 2010. Factorization machines. In *ICDM*, 995–1000. IEEE.

Semenov, A.; Romov, P.; Korolev, S.; Yashkov, D.; and Neklyudov, K. 2016. Performance of machine learning algorithms in predicting game outcome from drafts in Dota 2. In *International Conference on Analysis of Images, Social Networks and Texts*, 26–37. Springer.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1): 1929–1958.

Tauer, J. M.; and Harackiewicz, J. M. 2004. The effects of cooperation and competition on intrinsic motivation and performance. *Journal of personality and social psychology* 86(6): 849.

Usmani, S.; Bernagozzi, M.; Huang, Y.; Morales, M.; Sarvestani, A. S.; and Srivastava, B. 2020. Clarity: Data-Driven Automatic Assessment of Product Competitiveness. In *Proceedings of AAAI*, volume 34, 13204–13211.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*, 5998–6008.

Wang, F.; Liu, Q.; Chen, E.; Huang, Z.; Chen, Y.; Yin, Y.; Huang, Z.; and Wang, S. 2020. Neural cognitive diagnosis for intelligent education systems. In *Proceedings of AAAI*, volume 34, 6153–6161.

Wray, K. H.; Kumar, A.; and Zilberstein, S. 2018. Integrated Cooperation and Competition in Multi-Agent Decision-Making. In *Proceedings of AAAI*, 4751–4758.

Wright, M.; and Vorobeychik, Y. 2015. Mechanism design for team formation. In *Proceedings of AAAI*, 1050–1056.

Wu, R.; Deng, H.; Tao, J.; Fan, C.; Liu, Q.; and Chen, L. 2020. Deep Behavior Tracing with Multi-level Temporality Preserved Embedding. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2813–2820.

Xiao, J.; Ye, H.; He, X.; Zhang, H.; Wu, F.; and Chua, T.-S. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617* .

Zhang, K.; Zhang, H.; Liu, Q.; Zhao, H.; Zhu, H.; and Chen, E. 2019. Interactive attention transfer network for cross-domain sentiment classification. In *Proceedings of AAAI*, volume 33, 5773–5780.