Discriminative neural network for hero selection in professional Heroes of the Storm and DOTA 2

Daniel Gourdeau, Louis Archambault

***Abstract*—Multiplayer online battle arena games (MOBAs) are one of the most popular types of online games. Annual tournaments draw large online viewership and reward the winning teams with large monetary prizes. Character selection prior to the start of the game (draft) plays a major role in the way the game is played and can give a large advantage to either team. Hence, professional teams try to maximize their winning chances by selecting the optimal team composition to counter their opponents. However, drafting is a complex process that requires deep game knowledge and preparation, which makes it stressful and error-prone. In this paper, we present an automatic drafter system based on the suggestions of a discriminative neural network and evaluate how it performs on the MOBAs Heroes of the Storm and DOTA 2. We propose a method to appropriately exploit very heterogeneous datasets that aggregates data from various versions of the games. Drafter testing on professional games shows that the actual selected hero was present in the top 3 determined by our drafting tool 30.4% of the time for HotS and 17.6% for DOTA 2. The performance obtained by this method exceed all previously reported results.**

***Keywords* —** MOBA, esports, Heroes of the Storm, DOTA 2, Neural Network

I. INTRODUCTION

Multiplayer Online Battle Arena (MOBA) is an increasingly popular type of video game in which teams of 5 players face off on a virtual battleground. Each player controls a unique hero, selected before the start of the game. Players must cooperate with their team to take out the enemy by destroying a core structure located deep in enemy territory. The major MOBA games include titans like DOTA 2 (Valve Corporation) and League of Legends (Riot Games), which gather millions of players and viewers online [1]. Smaller MOBAs like Heroes of the Storm (Blizzard Entertainment) also attract a large number of players and nurture a thriving professional scene with more than 5 million dollars annually awarded in prize money for competing in online and offline tournaments in 2017 and 2018 [2]. Also, these games are increasingly being studied by researchers because they offer a wealth of data on important topics such as the player experience and toxic online behaviors [3].

The competitive mode of all of these games share a similar characteristic: prior to the start of the game, heroes are selected alternatingly by each team until each team is complete. This is known as the draft phase. Table I shows the drafting order for both teams. Each hero has strengths and weaknesses that make it unique. Therefore, optimal hero selection aims for synergistic relationships with other heroes on the team and to counter/mitigate the strengths of the opponent. In addition, the selection of an effective team of heroes is influenced by a relatively strict metagame. For example, the absence of heroes classified as *support* or *tank* can seriously handicap a team's chances to win the game. Therefore, the draft phase of a game is crucial because it can give a major advantage to one of the two teams.

However, it is a very complex task because of the large number of heroes available (>80 for Heroes of the Storm and >100 for DOTA and League of Legends). At a time where MOBA tournament stakes can reach millions of dollars, professional teams look for any advantages they can get in drafts against their opponents. Hours of manual draft preparation can be made before an important match, and teams are increasingly turning to professional coaches to assist them. Team Gen. G., 2018 World Heroes of the Storm champion, attributed much of its success to entrusting the draft phase to a retired player turned coach [4]. In order to make reliable and consistent drafting for their team, coaches are looking for data-driven tools that go beyond first-order statistics like heroes win rate.

To complicate things further, MOBAs are constantly evolving games, as developers roll out regular balance patches updating hero skills, tweaking hero characteristics to address potential imbalances and introducing new heroes. Heroes can see dramatic rise or fall in popularity following a balance patch, sometimes because a stronger replacement was introduced or because its strengths and weaknesses shifted. This makes it hard for an automated system to train on data from multiple patches, because a good recommendation for patch $N-1$ might be a bad one in patch N. This shifting metagame was identified as an obstacle for combining data from different patches by Summerville *et al.* [5]. In HotS, another layer of complexity is added to the drafting phase due to the 9 distinct battlegrounds that are available in competitive play. Each battleground features a unique map layout and a special objective that requires coordination and teamwork to capture for a large advantage. Drafting priorities change depending on which battleground the game takes place on, because different skills are needed to prevail on the objective. For example, high mobility heroes that can move quickly across maps like *Falstad* or *Dehaka* are regularly drafted in larger battlegrounds like *Cursed Hollow*, but are rarely seen on smaller ones like *Battlefield of Eternity*.

This paper will focus on predicting hero selection in professional games of Heroes of the Storm (HotS) and DOTA 2. We propose a single model aggregating data from different patches and different maps in order to increase the sample size for hero selection prediction.

- We introduce discriminative networks as an alternative

approach for optimal drafting. Exploitation of this model does not require expensive search strategies like MCTS and discriminator score can be used as an estimator of draft quality.
- We introduce a method to integrate patch and map information into a single model, allowing the use of heterogeneous datasets for training.
- We report state-of-the-art results on the problem of predicting the next pick in professional matches in 2 major MOBAs.

II. Related Works

Previous works have studied the drafting phase in DOTA and League of Legends in terms of determining the winner of amateur matches after the drafting phase using machine-learning models, using the selected heroes as input [6]–[10].No previous work has studied drafting in HotS, partly because it is a newer game. Once trained, the usefulness of those models is to make optimal draft choices based on this win rate estimator.

The possibility of predicting the winner from draft and using this model for drafting was first introduced by Conley *et al.* [6]. Next, the influence of hero choice and player experience was studied by Pobiedina *et al.* [8]. Classifier choice is usually crucial in any machine-learning application, and Semenov *et al.* [7] explore the importance of this when predicting the winner of DOTA 2 games. Since model performance is so dependent upon hyperparameters tuning, Porokhnenko *et al.* tune multiple classifiers with different hyperparameters applied to win rate prediction. Also, Wang *et al.* [10] introduce a new representation for each hero based on overall in-game statistics to inform the model on crucial features such as average gold earnings. Making these optimal choices using win rate prediction models require exploration of the possibility tree from a partially completed draft, which is realized by Chen *et al.* [11] using Monte-Carlo Tree Search (MCTS). Finally, machine-learning models can be trained to imitate experts [5], who are assumed to follow optimal drafting strategy. In the latter work, recurrent neural networks with memory units (LSTMs) were used to predict the next hero selected based on the previous heroes selected in professional DOTA 2 matches.

Reinforcement learning is a very successful approach to general game-playing and has been applied to the game of Go and DOTA 2 to create agents such as AlphaGo [12] and Open AI 5 [13] that surpassed human performance in their respective game. The key to successful reinforcement learning is to obtain a reliable evaluation function, which is hard to obtain for drafts without playing out the games and requires professional level game-playing agents on accelerated hardware. Open AI 5 actually devised a drafting scheme where the agents' expected win rate at the start of the game was evaluated for all possible drafts (with a restricted hero pool of 18 heroes) and a tree-search algorithm was used to optimally select from those. This "brute-force" method would not scale well to the full hero pool and heavily relies on the availability of professional level game-playing agents. Also, they report no formal comparison with professional drafting. Because of the resources required and limitations associated with reinforcement learning, we turn our attention to predicting picks in professional MOBAs play using a different approach.

Similarly to Summerville *et al.* [5], our work is based on a dataset of professional MOBA matches. However, instead of directly trying to predict the next pick with LSTMs as they suggested, we propose to learn a divergence measure for the manifold on which professional drafts lie or are close to using discriminative neural networks. We then approach drafting as minimizing the divergence to this manifold at all points during the draft.

The intuition behind this approach is that a discriminative network provides us with an explicit quality metric for drafts: bad drafts will have bad discriminative scores, whereas predicting the next pick gives no information about the state of the draft. Moreover, a discriminative model provides not only a ranking of possible choices, but also the magnitude of change in discriminator score associated with these. Another advantage of our method is that this model enforces the commutativity property of the drafting phase, while the LSTM method is sensitive to the order in which previous heroes have been drafted.

III. Methodology

A. Datasets

The dataset used to train the network comes from the masterleague.net website which hosts all the replays files of HotS professional games since the beginning of the official *Heroes Global Championship* (HGC) organized by Blizzard. As of December 2018, it contains 7630 games starting in March 2016 for 110 074 examples of hero picks and bans. The web site provides an API, available under a creative commons license (CC BY-NC-SA 4.0). The following data was extracted via the API: the ordered picked and banned heroes, the patch number and the map on which the game was played. The dataset is composed of these features about every available game in the database. Notably, this dataset ranges across 54 different balance patches and more than 30 new heroes were introduced in this period, in addition to more than 50 significant changes to some heroes' core mechanics. Drafting rules were also modified during this period, as a third ban was introduced in the first ban phase in June 2018. The draft data is arranged in *DraftStates* (X^t) which are vectors twice the length of the hero pool (82 playable heroes). The index of the *DraftStates* denotes the number of non-zero elements. The first half of the vector represents the heroes picked and banned for team A, while other half is for team B. The *DraftStates* are set to 0 at the start of the draft. Selecting a hero for a team will set the vector element associated with that hero on that team's half at 1, or at -1 for a ban. The *DraftStates* are built incrementally, following the order presented in table I: for each game in the database, there are 14 or 16 *DraftStates* associated (depending on the number of bans). For instance, for each game there will be the empty *DraftStates* (X^0), one where the first ban for team A is set to -1 (X^1), then another where the bans for both teams are set to -1 (X^2). The draft

	Ban	Ban	Ban	Ban	Pick	Pick	Pick	Pick	Pick	Ban	Ban	Pick	Pick	Pick	Pick	Pick
Team A	X		X		✓			✓	✓		X			✓	✓	
Team B		X		X		✓	✓			X		✓	✓			✓

TABLE I
DRAFTING ORDER IN A HOTS GAME.

goes on, following the selection rules, until the end of the draft where 10 elements of this vector are set to 1 and 4 or 6 elements are set to -1 depending on the patch number (for each hero selected and banned by both teams). Patch and map information are stored for each *DraftStates* in separate one-hot encoded vectors. The dataset is partitioned in training and testing set, with 5% of the games uniformly sampled and held out for testing.

The method is further tested on an additional DOTA 2 dataset provided by the administrator of the website datdota.com. This dataset is made of 10250 games drafted in professional events between December 2016 and January 2019. This dataset ranges from patch 7.0 to 7.20. DOTA 2 features a single map, so this feature is not needed here. The drafts are encoded in the same way as the HotS dataset, but the input vector is longer to account for the greater number of available characters. Also, complete drafts in DOTA 2 have more elements than in HotS, with 6 bans per team instead of 3, for a total of 22 steps in the draft instead of 16.

Another similar DOTA 2 dataset focused only on a single patch is also used in this work and is the same as the one used by Summerville *et al* [5]. It consists of 1518 DOTA 2 matches gathered during patch 6.85 and enables direct comparison of results between the methodologies.

B. Discrete generative adversarial network

The datasets described above provide the discriminator plenty of positive examples of professional drafts, but the negative examples need to be generated. We focus on the generation of negative examples that lie near the distribution of professional drafts in order to provide meaningful examples. Generative adversarial networks (GANs, first introduced by Goodfellow *et al* [14]) seem like a good candidate for this task. In particular, conditional GANs [15] are of interest, because they are conditioned with an input instead of random noise. Conditional GANs have had big impacts, especially in the image translation field with the Pix2Pix network [16]. In this paradigm, a generator network is trained to select one of the discrete choices for a given X^t associated with a map and a patch. On the other hand, a discriminator network tries to distinguish the real drafts from the generated draft. The generated examples are of the form X^{t+1} and denoted X_g^{t+1}. Negative examples are created by the generator sampling one of the possibles choices for a given X^t in the training dataset, excluding the real X^{t+1} for this example (even though multiple identical X^t can have a different X^{t+1}). The real X^{t+1} is excluded from the generator output to avoid a situation where the discriminator is faced with two contradictory examples. The discriminator output is a single sigmoid-activated value representing classification between generated examples (X_g^{t+1}) and true examples (X^{t+1}). A particularity of the GAN training presented here is the limited number of discrete choices available for the generator. Even with the exclusion of the real X^{t+1}, it could be impossible to distinguish between real and generated examples, which can lead to pathological GAN behavior. In order to verify this intuition, another version of the discriminator network is also trained with random sampling of the possible choices at X^t to create negative examples without using the generator network.

C. Architecture

Our implementation of the generator and the discriminator is coded using the Tensorflow library [17]. The networks are multilayer perceptrons with 3 hidden layers. The number of neurons per layer is $\{1024, 512, 128\}$ respectively. Both networks receive an input composed of the three input vectors described previously: X^t, maps and patches. The input vectors are concatenated into a single vector and fed to the neural network. Generator output is the same input X^t, with an additional pick or ban (one element set to 1 or -1). Because discrete operations block gradients' flow, the generator's output layer uses a relaxed one-hot categorical distribution, which is a continuous approximation to a one-hot categorical encoding, also called Gumbel-Softmax [18] . With a low enough temperature, samples from the Gumbel-Softmax distribution become one-hot but still support gradients' flow. This distribution is commonly used when creating discrete GANs [19]. Drafting logic is coded into the generator graph as run-time Tensorflow logical operations to prevent invalid choices like: picking an already picked or banned hero, picking a hero for the wrong team or banning at the wrong time. Training is stabilized using various common tricks, a good compilation of which is presented by Chintala *et al* [20]. We use different mini-batches for real and generated data when training the discriminator and leaky ReLU is the activation function for hidden layers to avoid sparse gradients. The ADAM optimizer is used for training with default parameters [21] and labels are soft and defined as 0.05 for real drafts and 0.95 for generated drafts. All weights are L1-regularized to avoid exploding gradients and heavy dropout (40% dropped neurons) is used. The discriminative network is trained for 2×10^8 iterations with a learning rate of 1×10^{-4} and mini-batch size of 100. Different mini-batches are used for real and generated data, as is recommended for GANs. This also balances the training, with the same number of positive and negative examples presented to the discriminator. Binary cross-entropy loss function is used for both generator and discriminator. The depth of the network was optimized by evaluating the discriminator's performance on random valid end-point drafts for 2, 3 and 4 layers respectively. Random valid end-point drafts refer to finished drafts where drafting rules are followed. Thus, 5 heroes are selected on both sides, the proper amount of heroes have been banned and no hero is chosen or banned more than once. Degradation of the accuracy on random valid end-point drafts was observed for 2 layers, while it remained similar between 3 and 4 layers. As such, the shallower model was preferred. When applied to the DOTA 2 dataset, the same network and the same parameters are re-used without tuning, except for the number of iterations that was increased to 6×10^8 to reach convergence.

D. Model evaluation

Once the discriminator is trained, drafting can be approached as picking the hero that minimizes the discriminator output. Since the number of possibilities is limited, it is possible to explore all the possible hero choices in order to minimize the discriminator output, which we'll denote X_d^{t+1}. The discriminator's quality is evaluated by:

(1) Testing its classification accuracy on random valid end-point drafts (in the sense that drafting rules are followed). The reasoning behind this is that a good discriminator should classify the vast majority of these as generated drafts.

(2) Testing its accuracy on the held-out testing set of professional games.

(3) Comparing the picked heroes in the testing set versus the top choices minimizing the discriminator output. The set of n possible choices minimizing the discriminator is denoted $X_{d_n}^{t+1}$. In a practical scenario, if $X^{t+1} \subset X_{d_n}^{t+1}$ for an appropriate n, the set of heroes that a player using our drafting tool has to consider could be reduced to n instead of every hero. The greater n is, the greater the possibility that $X^{t+1} \subset X_{d_n}^{t+1}$.

(4) Assessing model's learning curve when a new patch is introduced. To do so, a model is trained using all data up until a given patch and is tested on the next 50 games of this new patch, using the previous patch's weights. Then, games from this new patch are added incrementally to the training process and the model is always tested on the next 50 games. That way, it is possible to obtain the minimum number of games necessary for the model to adjust after a new patch. We test this methodology on minor patches 7.19 for DOTA 2 and patch 2.37.1 for HotS. Despite being a later patch, patch 7.20 for DOTA 2 was not used for this assessment because it was a major patch with a very large number of balance changes.

IV. Results

A. Hero clustering

In order to gain an understanding of the discriminator's process, it is useful to visualize the learned network weights. Fortunately, the learned weights of the trained network linking the input to the first hidden layer has a semantic meaning due to the discrete properties of the input. Indeed, the set of weights linking each element of the input to the first hidden layer represents a learned representation of the hero associated with that element. Thus, each hero has a set of weights associated and it is logical that heroes with similar roles should have similar weights. To test this hypothesis, a Ward hierarchical clustering algorithm is applied to the first half (representing heroes on team A) of the weights linking the input to the first hidden layer. The clustering results are shown in Figure 1. Four principal clusters are illustrated and are coherent with commonly accepted roles in HotS. The blue cluster is composed exclusively of *healers*, whose task is restoring their allies health and protecting them with defensive abilities. The green cluster is mostly composed of melee assassins, that need to close in on their enemies to deal damage. Next to it, the black cluster is composed of *tanks*, that are melee heroes that protect their allies by disabling their opponents. Thus, proximity between these roles is expected, and confirms the quality of the representations learned by the discriminator. Next, the red cluster is made entirely of ranged heroes that can poke their enemies from afar. Finally, the other greyed-out clusters are made of heroes that were not often picked in professional games, so the small sample size makes it hard for the network to learn meaningful weights. Indeed, the heroes in the shaded clusters account for 35.3 % of the available heroes, but less than 5% of the picks and bans in the dataset.

B. Model performance

	HotS Accuracy	
Model type	GAN	NN
Random drafts	0.9997	**0.999994**
Pro drafts	0.784	**0.843**
Top 10 choice	0.488	**0.624**
Top 5 choice	0.328	**0.435**
Top 3 choice	0.238	**0.304**
Top 1 choice	0.100	**0.131**

TABLE II
HotS discriminator accuracy in various scenarios

	DOTA 2 Accuracy			
Model type	GAN	NN	NN 7.07	NN 6.85
Random drafts	0.812	0.901	0.9992	**0.9994**
Pro drafts	0.668	**0.722**	0.489	0.683
Top 10 choice	0.352	0.393	0.316	**0.495**
Top 5 choice	0.227	0.259	0.203	**0.321**
Top 3 choice	0.154	0.176	0.144	**0.216**
Top 1 choice	0.062	0.075	0.051	**0.100**

TABLE III
DOTA 2 discriminator accuracy in various scenarios

Tables II and III presents the discriminator accuracy in the situations presented in the methodology. Accuracy on the top discriminator choices was calculated for values of n of $\{1,3,5,10\}$.The best values among all models are highlighted in bold. We observe that in both HotS and DOTA 2, the GAN approach underperformed compared to a discriminator trained with random sampling of the possibles X_g^{t+1} (NN in Tables II and III). This behavior is caused by the discrete nature of the hero picks and the fact that negative examples were generated using the previous state. Therefore, GAN generated examples are too similar to real examples and the discriminator is faced with contradictory examples.

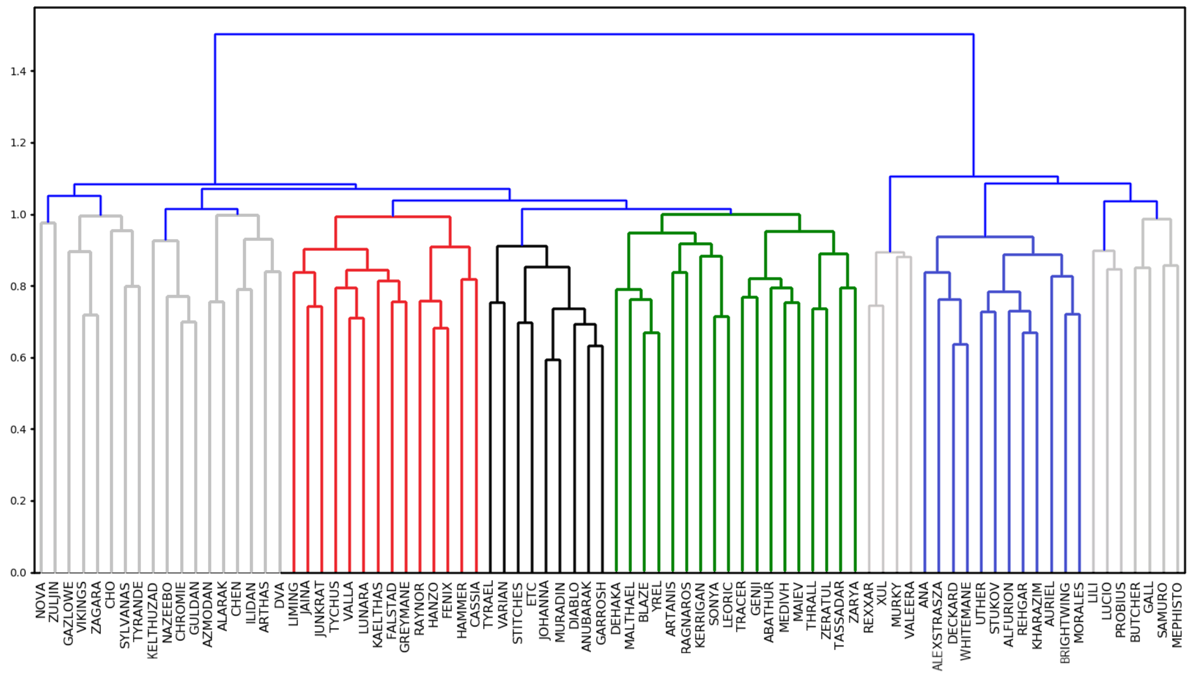

Fig. 1. Ward linkage hierarchical clustering of heroes-associated weights of the first layer of the discriminator. Clear clustering by hero role is visible in the graph. The blue cluster is made of healers, the black cluster contains only main tanks, while the green one next to it contains melee assassins and secondary tanks, both of which share similarities with main tanks. The red cluster contains only ranged assassins. The other greyed-out clusters comprise mostly characters that see little play in professional HotS, which means that the weights aren't well defined. A few heroes who see some play are out of place, for instance *Tracer*, who should be in the red cluster or *Lucio* and *Tyrande*, who should be in the blue cluster.

To the best of our knowledge, no work has been done on draft analysis in HotS before and the results are presented in Table II. Therefore, comparison will be attempted with similar work on DOTA 2 drafts by Summerville *et al* [5]. In order to provide a fair comparison, the same dataset is used with the same 11 fold cross-validation method. Table III also report results for DOTA 2 on the complete dataset (10 250 games compared with 1 518), in which the drafts are more heterogeneous as they range across several balance patches. Summerville *et al* [5] report accuracy on the top 3 recommended choices as 11.94%. By comparison, our method achieved an accuracy of 17.6% for DOTA 2 drafts without fine tuning the architecture or parameters. When trained on the same dataset, an even better score is obtained, with a top-3 accuracy of 21.6% compared to 11.94%. We also report results for a model trained only on patch 7.07 (1480 games), as a comparison with a similar number of games. The discriminator is also very specific towards professional drafts, with 84.3% accuracy on every DraftState of the testing set for HotS and 72.2% for DOTA 2. Random valid end-point drafts are also correctly identified as such with 99.9994% accuracy in HotS and 90.1% in DOTA 2, which shows that the professional drafts occupy only a tiny fraction of the valid drafts. The results for DOTA 2 could probably be improved by tuning the network parameters. This might help to bridge the gap between the performance observed across HotS and DOTA 2, but the draft complexity might simply be naturally higher for DOTA 2.

In order to investigate the impact of heroes pool size, the accuracy of the discriminator on the top 5 hero choice was produced for each patch, hypothesizing that an increasingly large number of available heroes would eventually reduce accuracy. From patches 0 to 55, more than 30 heroes were introduced in HotS, almost a 60% increase in available heroes. Figure 2 shows a stable accuracy with increasing hero pool, which means that the performance depends more on the game's complexity and balance than on the absolute number of available characters. Lower fluctuations in accuracy for HotS are observed starting at patch 30, mostly due to an increase in the number of games for these patches. Performance is more stable for DOTA 2, notably due to a higher number of games per patch. However, a steady increase in performance can be observed in the latest patches for DOTA 2, which may point to a metagame with less diversity than before.

Figure 3 illustrates that the accuracy is decreasing with draft progression for both games. This behavior is expected and was first reported by [5]. It is explained by the fact that a small subset made of the strongest heroes are selected first, leaving room for more diversity later on in the draft. Additionally, the high accuracy on the first ban for the second team (observed in both DOTA 2 and HotS) is explained by the fact that banning the strongest hero usually falls on the second team to choose, because team A picks first and they should deny them this hero.

Figure 4 presents the performance curve of the model on a new patch with an increasing number of games in the training set for this patch. Stable performance is obtained around 50 games for DOTA 2 and 20 games for HotS. Since both of these were relatively minor patches, it can be expected that major patches would take slightly longer until convergence. In addition, the figure highlights a steady decrease in accuracy for DOTA 2 for larger amount of games. This is an unexpected behavior, unseen for HotS. Consulting the DOTA 2 patch notes, it can be seen that there were 3 smaller patches at regular intervals during the 7.19 patch (7.19b, 7.19c, 7.19d), but we only have information about the major patch number on the dataset, while this minor patch information was available for HotS. Therefore, the longer the major patch goes on, it becomes less representative of the current minor patch version and performance decreases.

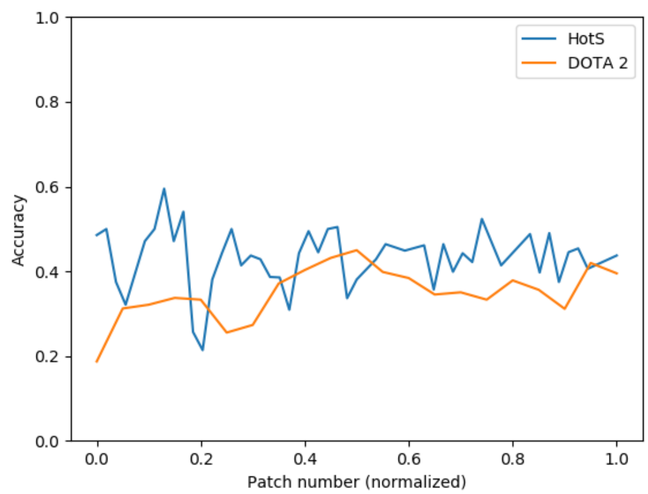

Fig. 2. Discriminator accuracy for the top 5 picks for every patch with professional games played. Larger variations are observed at the infancy of the professional league, with performance stabilizing around patch 30 near 40%. This is in part because of the metagame stabilizing and larger sample size for later patches. DOTA 2 performance is less noisy, in part because of higher number of games and lower number of patches. However, an upward trend is visible, with similar performance obtained on the current patch.

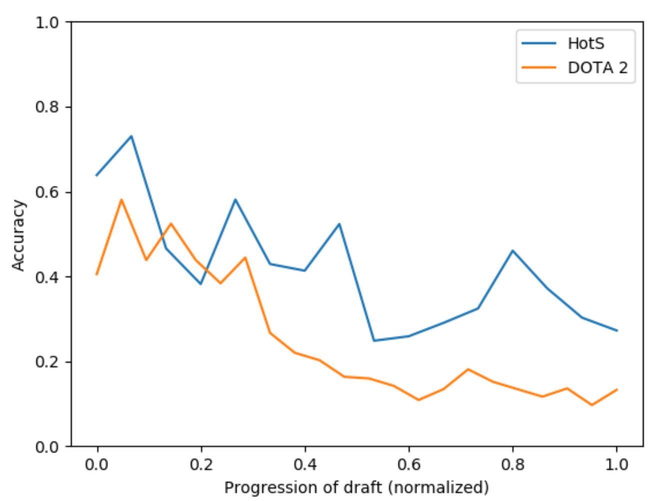

Fig. 3. Discriminator accuracy for the top 5 picks at each step of the drafting process. A significant decrease in accuracy is evident going further into the draft. This behavior is expected, as strongest heroes will be chosen first, and there is a greater diversity of reasonable choices later on in the draft.

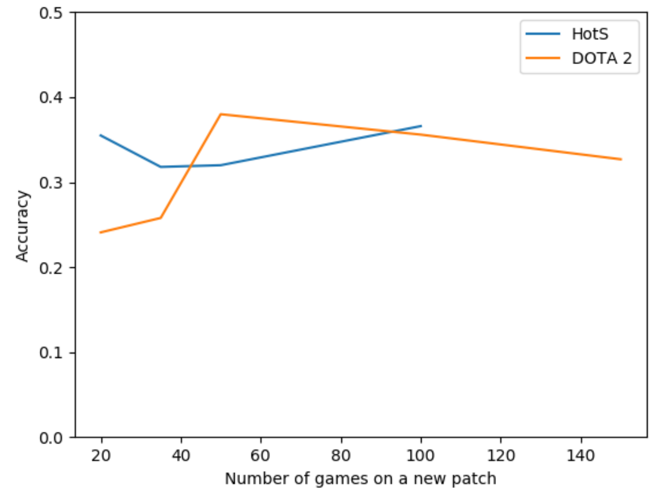

Fig. 4. Discriminator accuracy for the top 5 picks on a new patch for a increasing amount of training games on this new patch (patch 7.19 for DOTA 2 and patch 2.37.1 for HotS). Performance are already optimal after only 50 games on the new patch for DOTA 2 and 20 games on HotS.

HotS training couldn't reach 150 games because the number of games in the patch was only 170 (50 kept for testing).

	Picks					*Bans*		
Team A	Valla	Zeratul	Artanis	Garrosh	Deckard	Hammer	Medivh	Whitemane
Team B	Thrall	Diablo	Raynor	Yrel	**Healer?**	Liming	Genji	Hanzo

Hero name	**Role**	**Network value**
Kharazim	Healer	0.575
Rehgar	Healer	0.545
Lucio	Healer	0.542
Uther	Healer	0.538
Alexstrasza	Healer	0.527
Abathur	Support/Healer	0.518
Malfurion	Healer	0.515
Stukov	Healer	0.509
Junkrat	Assassin	0.498
Tassadar	Support	0.495

Fig. 5. Incomplete *DraftState* where Team B should pick a healer and top-10 ordered hero selection according to the discriminator. Choices highlighted in green are all healers and thus are reasonable. This draft takes place on the map *Volskaya Foundry* during the December 2018 balance patch.

As an example of the proposed method, Figure 5 presents an almost-complete *DraftState* in HotS where the last hero for team B is about to be picked. However, their team is missing a healer and it should logically be their next choice. Table 5 presents the top 10 choices according to the discriminator for this situation and the healers are colored in green. Strikingly, the top 8 heroes are all healers, showing that the network understands the basics of team building. Interestingly, it also presents the choice of *Abathur* (not usually classified as a healer), but replacing a conventional healer with *Abathur* is a relatively new and effective strategy when the team already has self-healing, which is the case here with all heroes on team B having some sort of self-healing. The last two choices in the top 10, *Junkrat* and *Tassadar*, would be poor picks in this situation.

V. Discussion

In this work, we presented a general approach to the problem of character drafting in 2 different MOBA esports. Our method used a discriminative neural network to predict which heroes professional players would have picked in this situation. We interpret the discriminator's output as the divergence to the manifold of professional drafts and propose to select heroes by minimizing the discriminator output. The proposed method describes a way to take into account the shifting metagame caused by game patches and new heroes, and shows stable performance over a long period of time that included major changes to HotS. The method, originally developed for HotS, generalized very well to DOTA 2 without any modification. This is encouraging for the application of this method to all MOBAs. We show that the network has learned useful representations of the heroes and the proposed picks in hypothetical situations demonstrate that basic team-building logic is respected. Computation of the next pick, alternatives and expected draft takes about a second, which enables its use in real time.

The proposed method is compared to a GAN approach to the problem, and pathological behaviors for discrete GANs are uncovered. However, generating negative examples using a GAN might be possible if the examples are generated from scratch. This way, the generated negative examples wouldn't overlap so much with the real distribution of professional games.

Significantly different scores are obtained depending on the dataset, which further supports Figure 2 in the sense that the performance fluctuates with the patch. Therefore, future work should make sure to compare performance on identical datasets, which calls for the release of a benchmark dataset. Performance within the same patch can also vary in DOTA 2 because of minor patches changes that are not available in the dataset. Therefore, a better dataset with all major and minor patch version should increase performance for DOTA 2.

The principal limitation of this work is that it does not aim to pick the best hero for a situation, only what the network thinks a professional would've picked. It is arguable that a professional would pick the best hero, but it also means that it is not really capable of innovative or surprising picks. Another drawback is the disruption of the model when a new patch is introduced. Whenever a new hero or patch is released, the model must be trained first on this new data because it can't anticipate what changes this will bring. However, it seems that only a very small amount of games are needed to adjust the model, with as little as 50 games required for DOTA 2 and 20 games for HotS until good performance is restored. these results were obtained on minor patches, with few balance changes, they can be considered a lower bound on the amount of games necessary. Larger patches will require a larger amount of games to adjust the model. In addition, in a practical scenario where the model would be used by a team, the model can also be trained on the drafts from this team's practice games (commonly called "scrims"), which would accelerate convergence of the performance.

This method is quite different from other approaches, such as trying to maximize the probability of winning the game on the basis of heroes selected or draft prediction using LSTM networks [5]. However, win rate maximization is a harder problem requiring very large datasets ([7], [11] used a dataset of more than 5 million games) because of a multitude of confounding factors like failure to execute a winning strategy, players *trolling* (intentionally wanting to lose) and skill difference between the teams, to name a few. This kind of large dataset is not available for the professional scene, thus limiting the applicability of this method on professional play. Furthermore, exploration of the drafting possibilities using techniques like Monte-Carlo Tree Search [11] are very interesting from a game theory point of view, but can lead the draft into team compositions that are not well characterized by the win rate estimator due to low density in the training data. Therefore, limiting the research to areas that are densely populated and where estimator prediction should be reliable is an interesting avenue. It seems that combining a win rate-maximizing model explored using MCTS, but constrained with the proposed approach could yield the best of both worlds: maximizing the chance to win while ensuring that the model doesn't stray too far from what is professionally viable.

Finally, the proposed method is directly applicable to any of the major MOBAs and the trained model can be used in various scenarios. For instance, it could be used by professional coaches and players to quickly get possible picks to choose from and remove some of the pressure from drafting, by commentators and casters to bring insight into the drafting process and explain to their audience or by aspiring semi-professional teams to teach them the professional metagame.

VI. Conclusion

In this paper, we tackled the task of the drafting phase of professional games of Heroes of the Storm using a discriminative neural network. This method generalized easily to professional DOTA 2 drafting, yielding state-of-the-art results. This work is motivated by professional teams and coaches who seek help to optimize their drafting strategy in order to gain an edge on the competition. We describe our approach to this task and show promising results on two separate datasets of professional MOBA esports. We were able to demonstrate that the network learned meaningful representation between heroes.

Esports is a growing domain where data is becoming more and more available. This load of data is waiting to be fully exploited and with increasingly high stakes in major tournaments, teams will look for any advantage they can get. We show that our method can be easily applied to different MOBA games, given sufficient data. This would help coaches and teams make informed choices during the drafting phase.

VII. Acknowledgements

The authors would like to thank Ben Steenhuisen from DatDota for sharing the DOTA 2 dataset and Adam Summerville for sharing the DOTA 2 dataset used in his article.

REFERENCES

[1] Johannes Amstrup Andersen. Best moba games – lol, dota 2, hots & smite compared. *https://www.progamerreview.com/best-moba-games/*, 2018.

[2] Esports earning. Top games of 2017. *https://www.esportsearnings.com/history/2017/games*, 2018.

[3] Marçal Mora-Cantallops and Miguel-Angel Sicilia. Moba games: A literature review. *Entertainment computing*, 26:128–138, 2018.

[4] Invenglobal News Letter. Interview with gen.g sake and reset: The champions of the 2018 hgc. *https://www.invenglobal.com/articles/6720/interview-with-geng-sake-and-reset-the-champions-of-the-2018-hgc*, 2018.

[5] Adam Summerville, Michael Cook, and Ben Steenhuisen. Draft-analysis of the ancients: Predicting draft picks in dota 2 using machine learning. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2016.

[6] Kevin Conley and Daniel Perry. How does he saw me? a recommendation engine for picking heroes in dota 2. *Np, nd Web*, 7, 2013.

[7] Aleksandr Semenov, Peter Romov, Sergey Korolev, Daniil Yashkov, and Kirill Neklyudov. 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*, pages 26–37. Springer, 2016.

[8] Nataliia Pobiedina, Julia Neidhardt, Maria del Carmen Calatrava Moreno, Laszlo Grad-Gyenge, and Hannes Werthner. On successful team formation: Statistical analysis of a multiplayer online game. In *Business Informatics (CBI), 2013 IEEE 15th Conference on*, pages 55–62. IEEE, 2013.

[9] Iuliia Porokhnenko, Petr Polezhaev, and Alexander Shukhman. Machine learning approaches to choose heroes in dota 2. In *Proceedings of the 24th Conference of Open Innovations Association FRUCT*, page 48. FRUCT Oy, 2019.

[10] Nanzhi Wang, Lin Li, Linlong Xiao, Guocai Yang, and Yue Zhou. Outcome prediction of dota2 using machine learning methods. In *Proceedings of 2018 International Conference on Mathematics and Artificial Intelligence*, pages 61–67. ACM, 2018.

[11] Zhengxing Chen, Truong-Huy D Nguyen, Yuyu Xu, Christopher Amato, Seth Cooper, Yizhou Sun, and Magy Seif El-Nasr. 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*, pages 200–208. ACM, 2018.

[12] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[13] Open AI. how-openai-five-works. *https://openai.com/five/#how-openai-five-works*, 2019.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[15] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR 2017*, 2017.

[17] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[18] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *ICLR 2017*, 2016.

[19] Matt J Kusner and José Miguel Hernández-Lobato. Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*, 2016.

[20] Soumith Chintala, Emily Denton, Martin Arjovsky, and Michael Mathieu. How to train a gan? tips and tricks to make gans work. *https://github.com/soumith/ganhacks*, 2018.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.