On the Effects of Simulating Human Decisions in Game Analysis

Vanessa Volz
Game AI research group
Queen Mary University
London, UK
v.volz@qmul.ac.uk

Boris Naujoks
IDE+A
TH Köln – University of Applied Sciences
Gummersbach, Germany
boris.naujoks@th-koeln.de

***Abstract*—The analysis of games or game content, e.g. in the context of AI-assisted game design and search-based PCG, is often based on playthroughs that are generated by simulating human decisions using AI players. With this paper, we hope to encourage more systematic analyses of uncertainties and errors potentially introduced by simulating human decisions in this context. To this end, we conduct a case study on StarCraft II to demonstrate the potential effects of these uncertainties. We construct two usecases from an analysis of existing approaches in research that employ simulations of human decisions through game AI. We are able to demonstrate large impacts of the simulations and finally discuss how the resulting uncertainties can be controlled in future work.**

***Index Terms*—Simulated Human Decision-Makers, Game Analysis, PCG, AI-assisted Game Design, StarCraft II**

I. Introduction

In many real-world problems, human decision-makers are a part of the system under investigation. This is true, for example, for optimising public transport or stock market strategies. Many of the state-of-the art approaches employ simulations to provide data for natural computing methods to approach these problems. Simulations are often used in applications where actually evaluating potential solutions is prohibitively expensive or even dangerous. However, relying on simulations obviously introduces an error and its effects on the fitness landscape and thus on the computed solutions are not clear. These effects should be investigated further, which we intend to do with this paper with an application to games.

In research on game analysis, artificial intelligence (AI) agents are commonly used to simulate human players taking actions in a game. The simulations are for example used to train and test AI agents (*self-play*), as well as for evaluating games and their content in the context of AI-assisted game design. For instance, AI agents have been used to identify patterns and problems in game design [2] as well as to evaluate search-based procedurally generated content [13]. Furthermore, Google DeepMind has achieved publicly acclaimed success on games such as GO and StarCraft II with their game-playing AI agents that were trained through self-play [11, 15].

However, it is also apparent that for results based solely on AI agent playthroughs, it is difficult to draw conclusions for games with human players. While the game-playing AIs developed by Google DeepMind are demonstrably able to beat professional human players, they do exhibit behaviour very unlike human players.[1] This means that playing patterns can potentially differ significantly between human and AI players. However, these patterns are often the sole source of information of research in game analysis, mainly due to the lack of data on human players (cf. survey in [17]). Natural computing methods based on such AI simulations (e.g [18]) can thus be misled indefinitely.

Therefore, with this paper, we hope to encourage more systematic analyses of uncertainties and errors potentially introduced by simulating human decisions in the context of the analysis of games (content). To this end, we first present two usecases in section II that we use to explore these uncertainties and errors in a clear and quantifiable fashion. The usecases further provide context and thus motivation for this analysis. We also conduct a review of literature employing simulation of human decisions in games in section III. We give an overview over the various applications of such AI simulations and find that their effects are rarely investigated. Following this, we detail several experiments based on the usecases that serve as a first investigation into the impact of AI simulations in section IV. As expected, we find that these effects are considerable and conclude in section V that their analysis is crucial when using AI simulations. We finally discuss several directions for future research regarding simulating human decisions and controlling its effects.

II. StarCraft II Win Prediction: Usecases

In the following, we present two usecases that we use to demonstrate the effects of simulating human decisions in game analysis in this paper. Both usecases are related to win prediction in StarCraft II, a real-time strategy (RTS) game described in more detail in section II-A. StarCraft II was chosen as an example application due to its considerable complexity and the availability of suitable datasets and analysis software. The experiments are based on an extensive dataset of StarCraft II replays, from human as well as AI players detailed in section

[1] See for example the plot on Actions Per Minute (APM) in [15]

978-1-7281-1884-0/19/$31.00 ©2019 IEEE

Fig. 1. Protoss base with multiple units and buildings.

II-B. We describe both usecases along with their separate motivations in sections II-C and II-D, respectively.

A. StarCraft II

StarCraft II[2] is a popular RTS game with a science-fiction theme released in 2010. It was designed as an E-Sport and has a massive following, regular tournaments (e.g. World Championship Series) and professional players.

StarCraft II features 3 playable races (Terran, Protoss, Zerg), 3 types of resources (minerals, vespene gas and food/supply), and several game modes (1v1, 2v2, 3v3, 4v4 and campaign), but this paper will focus exclusively on 1v1, the most popular game mode. Figure II shows a Protoss base with several buildings and units at an early stage of a game.

The player who successfully destroys all their opponent's buildings wins the game. The game also ends if a player concedes or a stalemate is detected by the game. Players have to carefully balance resource gathering efforts (economy), production of military units (army), and upgrades (technology). This balancing act is often called macromanagement, whereas the control of singular units is called micromanagement.

StarCraft: Brood War, an earlier release in the same series, has been used extensively in research as a benchmark and competition framework for AI agents [14]. More recently, in 2017, the *StarCraft II Learning Environment (SC2LE)* [16] was published. It provides an interface for AI agents to interact with a multi-platform version of StarCraft II and supports the analysis of previously recorded games. Moreover, it offers an interface, through which a large set of gamestate observations[3] can be tracked for every game tick in real-time. It also allows the analysis of replays, which are files that store complete games. Software for accessing the data is publicly available[4].

Google DeepMind recently organised a demonstration showing impressive progress in the challenge of developing proficient AI players for the game [15].

B. Datasets

1) Data Mining: In this paper, we are using three replay datasets for our analysis, which are described in the following:

- LADDER: 4955 1v1 ladder games (human vs. human) randomly sampled from publicly available replay packs. Ladder games count towards a player's ranking, which one generally seeks to improve.
- WCS: 419 1v1 games (human vs. human), played during the World Championship Series (WCS) tournament in Leipzig, Germany, January 26th-28th 2018[5].
- AI: 651 1v1 games (AI vs. AI) from the StarCraft II AI ladder[6]

It is important to note that, while the players whose games are contained in the LADDER dataset are not absolute beginners, their proficiency is expected to differ significantly from the (semi-)professional players in a WCS tournament. (Non-cheating) AI players were considered to be less proficient than most human players until the recent successes of AlphaStar [15]. However, StarCraft AI competitions and the SC2LE still continue to have traction as several challenges, especially related to macromanagement, are still open.

2) Features: For each of the games in the datasets, we collect several features that describe player progress. The features we were able to collect and use for further analysis are listed along with their interpretation in the following.

General features (metadata)

Map name	unique identifier of map
Race	Protoss, Terran, Zerg
Result	win, loss, tie, undecided
APM	actions per minute
Game duration	number of game ticks

The collected features contain some general information like the name of the map the match was played on, and how many game ticks it lasted. Additionally, the assigned race and result of both players are saved along with the APM statistic.

	Resource features (stats)
collection rate	resources collected per minute
current	number of unspent resources
used	number of spent resources (total)
killed	enemy units, buildings destroyed by player
lost	own units, buildings destroyed
friendly fire	own units, buildings destroyed by player

All resource features are available for both minerals and vespene gas separately and measured in these resource units. They describe the collection status of the respective resource, as well as building and units expressed in terms of their resource costs as an aggregated measure. The last four features are divided into three more categories (economy, army and technology) that indicate the type of building or unit the resource was spent on. Expert players are often able to identify player strategy and progress based on these resource features. We thus assume that this data contains enough information to train win prediction models on.

3) Preprocessing: As our goal is to find a generic characterisation of these real-world datasets, it is important to remove outliers and data artefacts in order to minimise misleading signals. An important example of such artefacts are games

[2] https://starcraft2.com

[3] https://github.com/deepmind/pysc2/blob/master/docs/environment.md

[4] for more information check https://github.com/deepmind/pysc2/blob/master/docs/environment.md, https://github.com/deepmind/pysc2, and https://github.com/Blizzard/s2protocol

[5] https://wcs.starcraft2.com/en-us/tournament/3895/

[6] http://sc2ai.net/

where players were away from their keyboard, and games where players lose on purpose. While such behaviour might result in an interesting characteristic, it is extremely unlikely to occur in AI or tournament games. Since the goal of this paper is to compare models across different datasets, we remove the corresponding replays from the datasets.

Following recommendations from previous work [19], we thus remove games

1) where at least one player performed 0 actions per minute,
2) that lasted 30 seconds or less,
3) where at least one player spent less than 50 minerals and already destroyed one of their own buildings.

After preprocessing, we are left with 4410 LADDER games, 419 WCS games and 651 AI games.

C. Usecase I: Map Balance

As StarCraft II is an E-Sport, it is of course important that maps played in tournaments and leagues are *balanced*, i.e. no race or playing style is severely favoured or disadvantaged. For example, maps with few choke points are often said to favour Zerg, as players of this race often rely on large armies with light units that can swarm the opponent[7]. At the same time, maps should also be diverse enough to encourage varied playing experiences.

Map imbalance issues in tournaments are usually mitigated by playing multiple games in each match-up, as well as allowing players to veto maps. Despite these provisions, the selection of maps is usually heavily criticised when introduced at the start of each season. It is thus of great interest to ensure the balance of all newly introduced maps. As extensive play-tests with human players are usually expensive, the question we pose in this usecase is: **Can map balance in StarCraft II be predicted based on simulations of human decisions?**

Similar usecases have been investigated in related work, such as in [6] where data from AI players is collected to support the creation of balanced maps in a shooter. AI simulations have also been used to optimise balance in decks for the card game TopTrumps in [18].

D. Usecase II: Embodied Win Prediction

Analysing games is not only important in the context of game design, but many state-of-the-art game AI approaches rely on the ability to assess how promising a given situation in a game is. This is why predicting the outcome of a game given observations of the game state is a popular topic in games research, including for StarCraft II (cf. [19]). This is especially true for embodied win prediction models, i.e. models that only rely on observations available to the player during the game.

The central question of this usecase is: **Can win prediction models trained on data from AI simulations be applied to games against human players?** This is interesting because game AIs are usually trained through self-play, i.e. by simulating human decisions, but are then pitted against human players (cf. [15]). To answer this question, we will mainly be relying on previous work on embodied win prediction in StarCraft II published in [19]. For the examples in this paper, we only consider the values at the very last game tick.

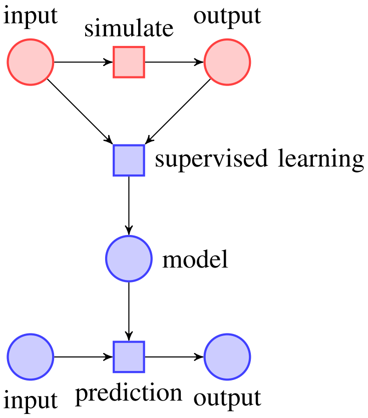

Fig. 2. Schematic overview of AI simulations in game analysis research. Direct approaches are depicted in red, where the output of a simulation is used directly. Model-based approaches (depicted in blue) use these simulations as training data to construct a model, whose predictions become the output.

III. Simulating Human Decisions in Games

A. Overview

In the following subsections III-B and III-C, we present a literature review of research that involves the analysis of games. Most of these publications are in the field of procedural content generation (PCG) or AI-assisted game design, where simulations are used for the automatic evaluation of games or game content. Most analysis approaches aim at finding explanations for how a given output variable changes based on an input. In StarCraft II, for example, a potential topic of analysis is the relationship between the winrates of Zerg players and the map played (see usecase I, section II-C). Answering this question could greatly help when identifying balancing issues in newly designed maps.

In the literature survey, we identified two main approaches of using AI simulations in game analysis research:

- Direct: The results of the simulation are used directly in the analysis
- Model-Based: The simulation results are used as training data for a machine learning model trained with supervised learning techniques.

The different approaches are also visualised in figure 2, where the direct approach is depicted in red and the model-based one in blue. Circles represent data objects, while squares refer to processes. The type of input and output data used of course depends on the question that is investigated in a given instance. This also holds true for the choice of AI players and the machine learning approach.

Both of the identified approaches rely on the assumption that observations made from AI playthroughs can be generalised to human gameplay. Continuing the example from above; a high winrate for AI Zerg players does not necessarily translate to a general Zerg advantage. Another possible explanation could

[7] https://liquipedia.net/starcraft2/Choke_Point

be that the implemented AI players are simply more proficient with Zerg gameplay, e.g. due to abusing their super-human micromanagement capabilities. In addition, if simulation data is used to train a model on win prediction (see usecase II, section II-D), different game-playing behaviour would result in different distributions of input data. This would be a considerable challenge for training a suitable machine learning model. Unfortunately, in most game analysis applications, it is not immediately clear whether generalisations as described are valid. In the following, we describe how the generalisability assumption for simulating human decision in games is handled in related work.

It is important to note that generalisability is not the only type of assumption regularly found in literature on game analysis. One underlying assumption, for example, is that complex abstract concepts such as *strategic depth* can be extracted from playthrough data and interpreted correctly (see e.g. [7]). Further examples of uncertainties are discussed in [17] in the context of a benchmark consisting of game problems. As discussed there, an additional complexity is that the various uncertainties are difficult to estimate and non-symmetric. They likely also interact in a cumulative fashion. More work is required towards investigating uncertainties in the game context, e.g. using benchmarks as suggested in [17].

In order to be able to provide a clear focus, we restrict the literature overview and discussions in the following sections to the generalisability assumptions caused by simulating human decisions with AI players.

B. Direct Approaches

There are a variety of publications that use AI agents to evaluate abstract concepts like the ones mentioned above. Among other metrics, [4] and [18] evaluate *closeness*, i.e. how close the game ended, using AI agents. In [12], the enjoyability of Pacman ghost teams is expressed as a weighted sum of several measures that express challenge and spatial, as well as behavioural diversity, based on gameplay data. In [2], performance along with further behavioural statistics such as frequency of specific actions, are analysed based on AI playthroughs for the board game *Ticket to Ride*.

While some of the publications listed above are mainly based on observations and previous experiences of a designer, several concept are also developed using formal theories, such as the theory of fun [9]. *Relative performance profiles* and *learnability* are both approaches that are based on this principle. According to the theory, human players enjoy learning new patterns, information and strategies.

It is assumed that the degree to which a given game allows and supports this learning progress can be approximated via its ability to differentiate between the performances of players in terms of in-game reward. Relative performance profiles measure this differentiation. They have been used to evaluate video games in context of the *General Video Game Playing* platform (GVGAI)[8] in [8]. Lantz et al. use relative performance profiles as a way to measure strategic depth in [7]. Isaksen et al. simulate varying player skill by adding randomised reactionary delays to their AI playing *Flappy Bird* and use the resulting score distribution to determine the level difficulty in [3].

There also exist more systematic frameworks that seek to support the usage of AI simulations for game analysis and evaluation. One such framework is *restricted play*, which was proposed by A. Jaffe for his work on game balancing. The concept of restricted play was later extended in [5] and constitutes evaluating game features by evaluating win probabilities of agents that are limited in specific ways. For example, if the impact of a certain action A is supposed to be evaluated, the win probabilities of an agent that is restricted in the sense that it is not allowed to execute A against an unrestricted agent are measured. In this case, if the win rate is around 50%, A would be shown to have no observable impact.

While some of the publications described above rely on theoretic models, it is not immediately clear whether results from AI simulations are transferrable to the actual games played by humans. To ensure that, the AI would need to be implemented in such a way that it behaves human-like. However, finding accurate and generalisable player models is still an unsolved problem in research on player modelling.

C. Model-Based Approaches

Model-based approaches, where the models are trained from generated data are less ubiquitous. One example is the work in [6], where a neural network is trained on a large amount of data generated through AI simulations. The resulting model is used to guide an algorithm for automated level design. This is done in order to save computational effort that would otherwise be necessary to evaluate each level design considered. In [10], a model employing neuroevolutionary pairwise preference learning and automatic feature selection is trained to predict engagement, frustration and challenge based on recorded gameplay data. However, basing models exclusively on simulated data also assumes that the datasets the models are trained on and applied to are similarly distributed.

In order to avoid errors in this regard, some researcher also integrate human gameplay data. One example is [20], where the authors propose an active learning method to automate playtests intended to tune low-level parameters. They test their approach on a shoot-'em-up game. In their paper, Zook et al. train regression and classification models to predict the value of a game metric or subjective response from game design parameters as an input. These models are trained on information from playtests by human players. The authors intend to minimise the number of playtests required to generate training data by using active learning methods to efficiently select game setups for human playtests.

However, approaches like this one bear another set of difficulties. First of all, enough appropriate data from human playtests needs to be available. This is made difficult because games in general combined with human nature make for a very noisy testbed. This means that each data point would need to

[8] gvgai.net

consist of multiple playthroughs, ideally of different people with different interests, playing styles and skill sets. In case there is not enough data, especially if we are investigating a multi-dimensional search space of game configurations, this method will extrapolate from the obtained knowledge. In addition, it is often unclear whether interpolating between observed data points such as levels creates meaningful results. Defining suitable distance measures is similarly complicated. As a result, modelling assumptions might be violated.

IV. Experiments

In the following, we first compare our artificially generated dataset called AI with our main real-world dataset, LADDER, using a descriptive analysis in section IV-A. The purpose of this analysis is to test whether the generalisability assumptions (described in section III-A) are likely to hold for the datasets used for our experimental analysis (see section II-B). Following that, we describe the results for both usecases described in sections II-C and II-D and discuss the observed effects of replacing human decisions with AI simulations.

A. Descriptive Analysis

All experiments are performed on data only and using a Kolmogorov-Smirnov test [1]. We are using the standard confidence level of $\alpha = 0.05$ for all tests. The results are supported visually by histograms and barplots in figure 3. The figures plot the value distribution of a specific feature displayed on the x-axis (i.e. used_vespene_technology in figure 3, 3rd row left). Only relative frequencies are displayed to allow a comparison between datasets of different sizes. The different datasets visualised in the figures are colour coded (blue: LADDER, red: AI) and can overlap, resulting in a purple colouring.

While the plots show the complete datasets, i.e. feature values for both players in a game for the sake of completeness, the tests are done only on feature vectors where the corresponding player won the game. This ensures that the data is independently distributed as is assumed by the Kolmogorov-Smirnov Test. It should also make the values of the features more comparable, especially since there are considerably more undecided or tied games in the AI datasets that produce outliers, especially in terms of game duration.

As both usecases addressed in this paper relate to win predictions (see sections II-C and II-D), in this section, we specifically analyse the distribution of features that are commonly associated with player success during a game (game duration, resource statistics), as well as with player skill (APM). The results are visualised in figure 3.

We observe that, while human players seem to play the races about equally (with slightly more Terran players, cf. figure 3 top left), research seems to focus on Zerg players. There is also a striking difference in terms of game result, since 20% of AI games end without a winner (cf. figure 3 top right), which is very unusual for games played by humans (LADDER as well as WCS). It is interesting to see that APM does not translate to proficiency in case of AI players (cf. figure 3 2nd row left). They perform worse than human players from LADDER, but achieve a much larger APM.[9] While AIs are of course not limited by human reaction times, the agents seem to be performing a great number of actions that are either not meaningful or possibly counterproductive. This behaviour is especially true for exploratory algorithms.

With only very few exceptions, the resource-related features from the AI and LADDER datasets are significantly differently distributed. In figure 3, we show some examples of differently distributed features, namely mineral collection rate (2nd row right), vespene gas used for technology and minerals used for economy (3rd row left and right, respectively) as well as minerals used for technology (bottom left). As a counterexample, we also added vespene gas lost by economy (bottom right).

The observations described above from the visual comparison of distributions presented in figure 3 are also strongly supported by the corresponding p-values received from a Kolmogorov-Smirnov test. The hypothesis that both datasets share the same cumulative distribution function is rejected in almost all cases. Typical p-values received were $9.592 \cdot 10^{-14}$ for the chosen race, or less than $2.2 \cdot 10^{-16}$ for APM, minerals collection rate, vespene gas used for technology, and minerals used for technology. As expected, the hypothesis was accepted for economy lost measured in vespene gas (bottom right in figure 3) with a p-value of 0.7116.

Based on the above descriptive analysis, we thus conclude that the LADDER and AI datasets follow different distributions.

B. Usecase I: Map Balance

For this usecase, we investigate the different datasets in terms of observable patterns regarding the correlations between map and race choice and the resulting winrate for the player in question. To do this, we select from the LADDER and AI datasets those maps that occur in both (M1: Abyssal Reef LE, M2: Ascension to Aiur LE, M3: Interloper LE, M4: Mech Depot LE, M5: Odyssey LE). WCS is not used in the following experiments, as the map pool for the tournament was completely different. For each of those maps and datasets, we compute the respective winrates of players of all three races. The results are presented in table I, with a colour gradient used as a visual aide to spot patterns.

It is quite striking that based on the LADDER data, all maps seem relatively balanced with respect to race. All of the races have a winrate of around 50%. In contrast, the AI dataset seems to show that Terran has a clear disadvantage on most maps (except for M5: Odyssey LE). From this simple comparison, it is already evident that the data obtained from the AI dataset could not be used to predict map balance for human league games. This is because none of the observed patterns in the data is reflected in the LADDER dataset.

A potential source of this lack of correlation is the different distributions of the data as described in the previous section. AI games end in ties or as undecided far more often than LADDER

[9] The APM distribution of Goolge DeepMind's AlphaStar AI is also very different from that of its human pro-player opponents [15].

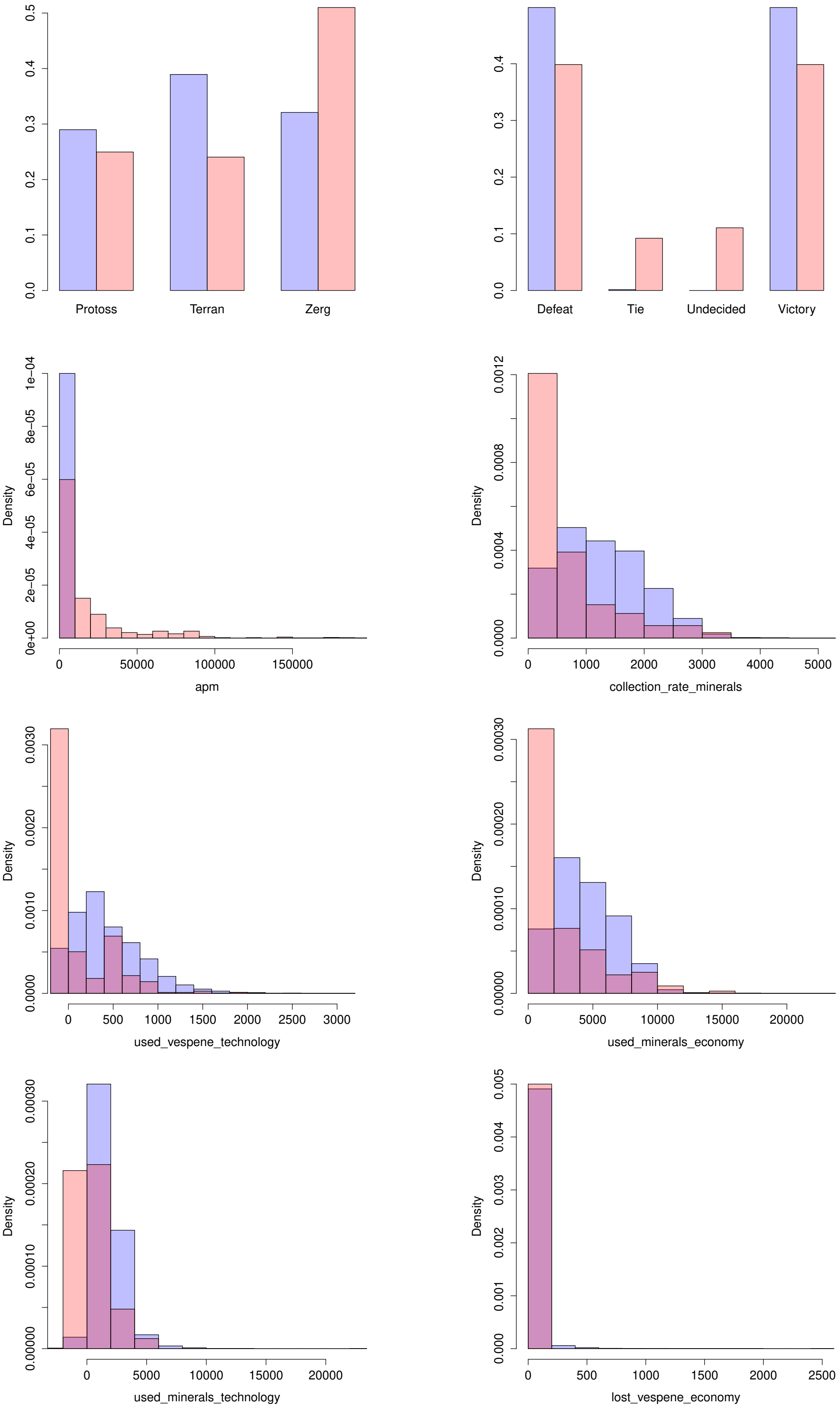

Fig. 3. Comparison of features in LADDER and AI. Dataset LADDER is displayed in blue, AI in red. Features displayed from left to right, top to bottom are assigned race, result, APM, mineral collection rate as well as minerals spent on army, economy and technology, respectively. The last graphic in the lower right corner shows economy lost measured in vespene gas.

TABLE I

WINRATES BY RACE ON DIFFERENT MAPS FROM THE LADDER AND AI DATASETS. COLOUR GRADIENT ADDED AS VISUAL AIDE.

Map		LADDER			AI	
	Protoss	Terran	Zerg	Protoss	Terran	Zerg
M1	0.5	0.51	0.48	0.58	0.31	0.32
M2	0.49	0.49	0.52	0.55	0.31	0.43
M3	0.51	0.47	0.52	0.44	0.26	0.47
M4	0.54	0.44	0.53	0.61	0.27	0.45
M5	0.5	0.48	0.52	0.12	0.5	0.35

games (see figure 3 top right). In addition, the distribution of data on the different races is also imbalanced for the AI dataset (see figure 3 top left). Another issue are the different magnitudes of the datasets: There were only between 210-224 samples for each map in the AI dataset, while the LADDER dataset contains between 1153-1542 samples per map.

C. Usecase II: Embodied Win Prediction

For our second usecase, we train several machine learning models for embodied win prediction following previous research in [19]. The models are trained on different data sets and we investigate their performance in the following.

We thus train a simple Artificial Neural Network (ANN - 1 hidden layer, 10 neurons) to predict the winner of a match given the remaining features as input. We report the mean and standard deviation of the prediction accuracies observed in 30 independent tests. For a baseline, we first trained such a predictor separately on each of the datasets using cross-validation and a 90/10 split. Results are presented in table II. The obtained mean accuracies are very high, with a small standard deviation. This was expected, as the data describes the gamestate at the end of the game and no generalisation is required.

TABLE II

WINNER PREDICTION ACCURACY FOR BASELINE EXPERIMENT: PREDICTOR TRAINED ON EACH DATA SET SEPARATELY. MEAN PREDICTION ACCURACY (MEAN) AND STANDARD DEVIATIONS (SD) ARE PROVIDED.

dataset	mean	SD
LADDER	0.939531	0.008019
AI	0.975128	0.014095
WCS	0.920238	0.035114

However, in the usecase described in section II-D, the model would be trained on artificially generated data, but applied to predict the winner of human vs. human matches, or vice versa. We thus conduct a second set of experiments, where we train the ANN on one data set and test it on a different one. In table III, we list the different combinations of training and test set in the first two columns, as well as the obtained mean prediction accuracy and standard deviation in the last two columns.

We observe that in the experiments involving both LADDER and AI (rows 1-2), the trained predictors achieve accuracies of around 0.5. For the LADDER dataset, this is barely better than chance, as there are almost no undecided or tied games. However, the baseline experiment already demonstrates that much better prediction accuracies are achievable (cf. table II). The models are thus clearly not able to generalise well.

TABLE III

WINNER PREDICTION ACCURACY FOR GENERALISATION EXPERIMENT: DIFFERENT DATASETS FOR TRAINING AND TESTS. MEAN VALUES (MEAN) AND STANDARD DEVIATIONS (SD) ARE PROVIDED.

trained on	tested on	mean	SD
LADDER	AI	0.529391	0.059723
AI	LADDER	0.510401	0.011530
LADDER	WCS	0.950040	0.006451
WCS	LADDER	0.869531	0.019162

In order to investigate further whether the generalisation issues stem from the fact that AI simulations are used as a source for the data, we repeat the experiment with the WCS dataset instead of AI. The players in the WCS are much more experienced and should behave significantly differently than players in LADDER, including using different strategies.

The results in table III (rows 3-4) show that generalisation works significantly better in this case. The prediction accuracy on the WCS dataset was even improved when compared to the baseline experiment. This might be due to the small number of games in WCS. This conjecture would also be supported by the fact that the WCS predictor has the highest standard deviation in table II. Even the predictor trained on WCS data is able to achieve 86% accuracy on the LADDER dataset.

V. LESSONS LEARNED

The analysis of games or game content, e.g. in the context of AI-assisted game design and search-based PCG, is often based on playthroughs generated via simulations of human decisions using AI players. With this paper, we hope to encourage researchers to investigate more systematically what effects these simulations have on the results of a given application.

To this end, we investigate existing literature that relies on such AI simulations and identify two main approaches of using the generated data: *direct* and *model-based* (see section III-A). We find that the main source of potential errors is the assumption of generalisability. We develop two usecases related to StarCraft II win prediction that represent these approaches and motivate them in sections II-C and II-D. We investigate these usecases experimentally based on 3 datasets.

The descriptive analysis of the data already indicates a stark difference between behaviour of human players and AI agents. We thus conclude that the generalisability assumption was incorrect. In the experiments related to the usecases, we were able to demonstrate clear effects of simulating human decisions on the results we obtained in our case study.

Since we were not able to test a multitude of different settings and games, we cannot generalise our findings to all approaches based on data from AI simulations. Additionally, the behaviour of the AlphaStar AI is arguably more similar to the strategies developed by human players than previous AIs in the bot ladder. If replays from the AlphaStar internal league would be made available publicly in sufficient quantity, the resulting dataset should be included in the above experiments.

Nevertheless, our usecases provide counter-examples to the generalisability assumptions often made in literature. We thus strongly recommend that any methodology that relies on generalisation of patterns observed in simulated data should be carefully tested in terms of the error this approach introduces into the results. This is true for applications in AI-assisted game design and search-based PCG, as well as game AI relying on self-play.

One potential way of mitigating these errors is through applying more sophisticated player modelling approaches as mentioned in section III-B. However, many open research questions still remain, including how to quantify the differences between two observed player strategies, and how to ensure *test coverage*, i.e. that a suitable range of player strategies is represented.

Another way of controlling errors from the generalisation assumption could be to create mapping functions that are able to translate features observed in one dataset to another. The mapping function (e.g. a transition matrix) could be specified using an EA that minimises the multivariate statistical distance between the mapped AI data and the target data (e.g. Energy distance). However, depending on the application, such a function does not necessarily exist and would introduce another, albeit measurable, source of error.

Furthermore, the error introduced by model-based approaches, as discussed in section III-C, can be controlled by testing the models on data from human players, if available. Concerted efforts regarding collecting and sharing suitable datasets should be made in the community to facilitate these efforts. In addition, more research on the analysis of general patterns of data generated from AI simulations could support the choice of modelling approach. Finally, more interdisciplinary research approaches on surrogate modelling and uncertainty quantification in optimisation seems to be a worthwhile future step[10] in order to (a) reduce the number of data samples required and (b) understand the effects of the multiple sources of uncertainties present in game analysis problems.

References

[1] W. Conover. *Practical nonparametric statistics*. Wiley, NY, 3. edition, 1999.

[2] F. de Mesentier Silva, et al. AI-based Playtesting of Contemporary Board Games. In *Foundations of Digital Games Conference (FDG)*. ACM Press, New York, 2017.

[3] A. Isaksen, et al. Discovering Unique Game Variants. In *Computational Creativity and Games Workshop at the International Computational Creativity*, http: //game.engineering.nyu.edu/projects/exploring-game-space/, (accessed 30. May 2019), 2015.

[4] A. Isaksen, et al. Characterising Score Distributions in Dice Games. *Game & Puzzle Design*, 2(1):24–37, 2016.

[5] A. Jaffe. *Understanding Game Balance with Quantitative Methods*. Phd thesis, University of Washington, 2013.

[6] D. Karavolos, A. Liapis, and G. N. Yannakakis. Using a surrogate model of gameplay for automated level design. In *IEEE Computational Intelligence and Games (CIG)*. IEEE Press, Piscataway, NJ, 2018.

[7] F. Lantz, et al. Depth in Strategic Games. In *What's Next for AI in Games? Workshop of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI Press, Palo Alto, CA, 2017.

[8] T. S. Nielsen, et al. General video game evaluation using relative algorithm performance profiles. In A. Mora and G. Squillero, editors, *Applications of Evolutionary Computation (EvoApplications)*, pages 369–380. Springer, Cham, Switzerland, 2015.

[9] J. Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990-2010). *Trans. on Autonomous Mental Development*, 2(3):230–247, 2010.

[10] N. Shaker, et al. Fusing Visual and Behavioral Cues for Modeling User Experience in Games. *IEEE Transactions on Cybernetics*, 43(6):1519–1531, 2013.

[11] D. Silver, et al. Mastering the game of go without human knowledge. *Nature*, 550:354–359, 2017.

[12] W. Sombat, P. Rohlfshagen, and S. M. Lucas. Evaluating the enjoyability of the ghosts in Ms Pac-Man. In *IEEE Computational Intelligence and Games (CIG)*, pages 379–387. IEEE Press, Piscataway, NJ, 2012.

[13] J. Togelius, et al. The Mario AI Championship 2009–2012. *AAAI AI Magazine*, 34(3):89–92, 2013.

[14] M. Čertický and D. Churchill. The current state of starcraft AI competitions and bots. In *Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE)*. AAAI Press, Palo Alto, CA, 2017.

[15] O. Vinyals, et al. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/ (accessed 30. May 2019), 2019.

[16] O. Vinyals et al. Starcraft II: A new challenge for reinforcement learning. *CoRR*, abs/1708.04782, 2017. URL http://arxiv.org/abs/1708.04782.

[17] V. Volz. *Uncertainty Handling in Surrogate Assisted Optimisation of Games*. PhD thesis, TU Dortmund University, 2019.

[18] V. Volz, G. Rudolph, and B. Naujoks. Demonstrating the Feasibility of Automatic Game Balancing. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 269–276. ACM, NY, 2016.

[19] V. Volz, M. Preuss, and M. K. Bonde. Towards embodied starcraft II winner prediction. In *Computer Games Workshop at IJCAI 18*, https://pdfs.semanticscholar.org/87a6/71703c7ed169ab96528854edbeb9627df81c.pdf (accessed 30. May 2019), 2018.

[20] A. Zook, E. Fruchter, and M. O. Riedl. Automatic playtesting for game parameter tuning via active learning. In *Foundations of Digital Games (FDG)*, http://www.fdg2014.org/proceedings.html (accessed 30. May 2019). Society for the Advancement of the Science of Digital Games, Santa Cruz, CA, 2014.

[10] Recent suggestions in https://uqop.sciencesconf.org/246597