The Shapley Value of Classifiers in Ensemble Games

Benedek Rozemberczki
The University of Edinburgh
Edinburgh, United Kingdom
benedek.rozemberczki@ed.ac.uk

Rik Sarkar
The University of Edinburgh
Edinburgh, United Kingdom
rsarkar@inf.ed.ac.uk

ABSTRACT

How do we decide the fair value of individual classifiers in an ensemble model? We introduce a new class of transferable utility cooperative games to answer this question. The players in *ensemble games* are pre-trained binary classifiers which collaborate in an ensemble to correctly label points from a dataset. We design *Troupe* a scalable algorithm which designates payoffs to individual models based on the Shapley value of those in the ensemble game. We show that the approximate Shapley value of classifiers in these games is an adequate measure for selecting subgroup of highly predictive models. In addition, we introduce the Shapley entropy a new metric to quantify the heterogeneity of machine learning ensembles when it comes to model quality. We analytically prove that our Shapley value approximation algorithm is accurate and scales to large ensembles and big data. Experimental results on graph classification tasks establish that *Troupe* gives precise estimates of the Shapley value in ensemble games. We demonstrate that the Shapley value can be used for pruning large ensembles, show that complex classifiers have a prime role in correct and incorrect classification decisions and provide evidence that adversarial models receive a low valuation.

ACM Reference Format:
Benedek Rozemberczki and Rik Sarkar. 2018. The Shapley Value of Classifiers in Ensemble Games. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The advent of black box machine learning models raised fundamental questions about how input features and individual training data points contribute to the decisions of expert systems [11, 19]. There has also been interest in how the heterogeneity of models in an ensemble results in heterogeneous contributions of those to the classification decisions of the ensemble [10, 37]. For example one would assume that computer vision, credit scoring and fraud detection systems which were trained on varying quality proprietary datasets output labels for data points with varying accuracy. Another source of varying model performance can be the complexity of models e.g. the number of weights in a neural network or the depth of a classification tree.

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

Quantifying the contributions of models to an ensemble is paramount for practical reasons. Given the model valuations, the gains of the task can be attributed to specific models, large ensembles can be reduced to smaller ones without losing accuracy [15, 22] and performance heterogeneity of ensembles can be gauged [10]. This raises the natural question: How can we measure the contributions of models to the decisions of the ensemble in an efficient, model type agnostic, axiomatic and data driven way?

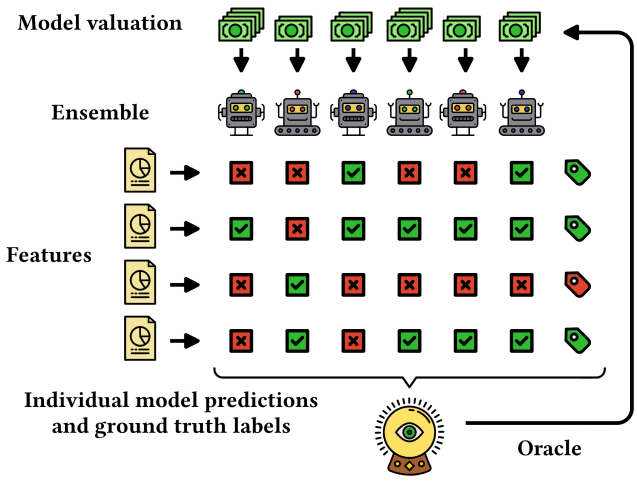

Figure 1: An overview of the model valuation problem. Models in the ensemble receive features for a set of data points and score those. Using the predictions and ground truth labels the oracle quantifies the worth of models.

We frame this question as the model valuation problem in an ensemble which we figuratively describe in Figure 1. The solution to this problem requires a general data-driven analytical framework to assess the worth of individual classifiers in the ensemble. Each classifier in the ensemble receives the features of data points. The classifiers output for each data point a probability distribution over the potential classes. Using these propensities an oracle which has access to the ground truth labels quantifies the worth of models in the ensemble. These importance metrics can be used to make decisions – e.g. pruning the ensemble and allocation of payoffs. A considerable design advantage of a framework like this is that the machine learning model owners do not have access to the labels.

Present work. We introduce *ensemble games*, a class of transferable utility cooperative games [25]. In these games binary classifiers which form an ensemble play a voting game to assign a binary label to a data point by utilizing the features of the data point. Building on the ensemble games we derive *dual ensemble games* in which

the classifiers cooperate in order to misclassify a data point. We do this to characterize the role of models in incorrect decisions.

We argue that the Shapley value [31], a solution concept from co-operative game theory, is a model importance metric. The Shapley value of a classifier in the ensemble game defined for a data point can be interpreted as the probability of the model becoming the pivotal voter in a uniformly sampled random permutation of classifiers. Computing the exact Shapley values in an ensemble game would take factorial time in the number of classifiers. In order to alleviate this we exploit an accurate approximation algorithm of the individual Shapley values which was tailored to voting games [7]. We propose *Troupe*, an algorithm which approximates the average of Shapley values in ensemble games and dual games using data.

We utilize the average Shapley values as measures of model importance in the ensemble. The Shapley values are interpretable as an importance distribution over classifiers, hence the information entropy of this distribution is a straightforward measure of ensemble heterogeneity when it comes to model quality. Using the newly introduced *Shapley entropy* we are able to quantify heterogeneity with respect to correct and incorrect decisions.

We evaluate *Troupe* by performing various classification tasks. Using data from real world webgraphs (Reddit, GitHub, Twitch) we demonstrate that *Troupe* outputs high quality estimates of the Shapley value and the ensemble heterogeneity metrics. We validate that the Shapley value estimates of *Troupe* can be used as a decision metric to build space efficient and accurate ensembles. Our results establish that more complex models in an ensemble have a prime role in both correct and incorrect decisions.

Main contributions. Specifically the contributions of our work can be summarized as:

(1) We propose ensemble games and their dual games to model the contribution of individual classifiers to the decisions of voting based ensembles.
(2) We design *Troupe* an approximate Shapley value based algorithm to quantify the role of classifiers in decisions.
(3) We define ensemble heterogeneity metrics using the entropy of approximate Shapley values outputted by *Troupe*.
(4) We provide a probabilistic bound for the approximation error of average Shapley values estimated from labeled data.
(5) We empirically evaluate *Troupe* for model valuation and forward ensemble building on graph classification tasks.

The rest of this work has the following structure. In Section 2 we discuss related work on the Shapley value, its approximations and applications in machine learning. We introduce the concept of ensemble games in Section 3 and discuss Shapley value based model valuation in Section 4 with theoretical results. We evaluate the proposed algorithm experimentally in Section 5. We summarize our findings in Section 6 and discuss future work. The reference implementation of *Troupe* is available at https://github.com/benedekrozemberczki/shapley.

2 RELATED WORK

The *Shapley* value [31] is a solution technique which distributes the gains of the grand coalition in a transferable utility cooperative game [25]. It is widely known for its axiomatic properties [4] such as *efficiency* and its factorial time exact computation which makes it intractable for games with a large number of players. Hence, general [20, 27] and game class specific [7] approximation techniques have been proposed to obtain estimates of it. In Table 1 we compare various approximation schemes in terms of having certain desired properties.

The Monte Carlo permutation sampling (*MC*) and its truncated variant (*TMC*) generate random permutations of the players to estimate the Shapley value with the average of marginal contributions [20, 21, 39]. The errors of these approximations are bounded, but the stochastic sampling gives a non-exact estimate. In order to create a tractable approximation [27] proposes a multilinear extension (*MLE*) of the Shapley value which gives exact estimates in linear time. A variant of this technique [16, 17] calculates the value of large players explicitly and applies the *MLE* technique to small ones. The only approximation technique tailored to weighted voting games is the expected marginal contributions method (*EMC*) which estimates the Shapley values based on contributions to varying size coalitions. Our proposed algorithm *Troupe* builds on *EMC* as it is error bound, exact, and specific to our setting.

Table 1: Comparison of Shapley value computation and approximation techniques in terms of having (✔) and missing (✘) desiderata; complexities with respect to the number of players m and permutations p.

Method	Voting	Bound	Exact	Space	Time
Explicit	✔	✔	✔	$O(m)$	$O(m!)$
MC [21]	✘	✔	✘	$O(m)$	$O(mp)$
TMC [11]	✘	✔	✘	$O(m)$	$O(mp)$
MLE [27]	✘	✘	✔	$O(m)$	$O(m)$
MMLE [16, 17]	✘	✘	✔	$O(m)$	$O(m!)$
EMC [7]	✔	✔	✔	$O(m)$	$O(m^2)$

The earliest use of the Shapley value in machine learning was for measuring feature importance in linear models [18, 23, 28]. In the feature selection setting the features are seen as players that cooperate to achieve high goodness of fit. Various discussed approximation schemes [20, 33] have been exploited to make feature importance quantification in high dimensional spaces feasible [19, 34, 35] when explicit computation is not tractable. Another machine learning domain for applying the Shapley value was the pruning of neural networks [1, 12, 32]. In this context approximate Shapley values of hidden layer neurons are used to downsize overparametrized classifiers. It is worth noting that pruning neurons [32] is feature selection – representative hidden layer features are chosen. Finally, there has been increasing interest in the equitable valuation of data points with game theoretic tools [14]. In such settings the estimated Shapley values are used to gauge the influence of individual points on a supervised model. These approximate scores are obtained with group testing of features [14] and permutation sampling [3, 11].

3 ENSEMBLES GAMES

We introduce novel classes of co-operative games and examine axiomatic properties of solution concepts which can be applied to these games. We overview an exact solution of these games, the Shapley value [31], and discuss various approximations schemes of the Shapley value based solution [7, 21, 27].

3.1 Defining ensemble games

We are going to define a class of games in which binary classifier models (players) co-operate in order to label a single data point correctly in a voting based ensemble of classifiers. A second class of games is derived in which the games co-operate to misclassify the data point.

Definition 1. ***Labeled data point.*** *Let $(\boldsymbol{x}, y)$ be a labeled data point where $\boldsymbol{x} \in \mathbb{R}^d$ is the feature vector and $y \in \{0, 1\}$ is the corresponding binary label.*

Definition 2. ***Positive classification probability.*** *Let $(\boldsymbol{x}, y)$ be a labeled data point and M be a binary classifier, $P(y = 1 \mid M, \boldsymbol{x})$ is the probability of the data point having a positive label output by classifier M.*

Our work considers arbitrary binary classifier models (e.g. classification trees, support vector machines, neural networks) that operate on the same input feature vector $\mathbf{x}$. We are agnostic with respect to the exact type of the model, we only assume that M can estimate the probability of the data point having a positive label. We do not assume that the model owner can access the label, rather that given the features a probability of the positive class can be output by the model.

Definition 3. ***Ensemble.*** *An ensemble is the set $\mathcal{M}$ that consists of binary classifier models $M \in \mathcal{M}$ which can each output a probability for a data point $\boldsymbol{x} \in \mathbb{R}^d$ having a positive label. The size of the ensemble m equals $|\mathcal{M}|$ which is the cardinality of the binary classifier set.*

We assume that the decision about the predicted label of a data point is based on the arithmetic average of the output probabilities. Formally this means that:

$$P(y = 1 \mid \mathcal{M}, \mathbf{x}) = \sum_{M \in \mathcal{M}} P(y = 1 \mid M, \mathbf{x})/m.$$

Using this probability the decision rule for labeling the data point is:

$$\widehat{y} = \begin{cases} 1 & \text{if } P(y = 1 \mid \mathcal{M}, \mathbf{x}) \geq \gamma, \\ 0 & \text{otherwise.} \end{cases}$$

where $0 \leq \gamma \leq 1$ is the decision cutoff value and $\widehat{y}$ is the predicted label of the data point.

Definition 4. ***Sub-ensemble.*** *A sub-ensemble $\mathcal{S}$ is a subset $\mathcal{S} \subseteq \mathcal{M}$ of binary classifier models.*

Definition 5. ***Individual model weight.*** *The individual weight of the vote for M, in sub-ensemble $\mathcal{S} \subseteq \mathcal{M}$ for data point $(y, \boldsymbol{x})$ is defined as:*

$$w_M = \begin{cases} P(y = 1 \mid M, \boldsymbol{x})/m & \text{if } y = 1, \\ P(y = 0 \mid M, \boldsymbol{x})/m & \text{otherwise.} \end{cases}$$

The individual model weight of the binary classifier $M \in \mathcal{M}$ is bounded, such that $0 \leq w_M \leq 1/m$, because the classification probabilities themselves are bounded. It is also worth emphasizing that the weight of $M \in \mathcal{S}$ depends on m, the number of models in the large ensemble (which is $|\mathcal{M}|$) and not on $|\mathcal{S}|$, which is the number of classifiers in the sub-ensemble.

Definition 6. ***Ensemble game.*** *Let $\mathcal{M}$ be a set of binary classifiers. An ensemble game for a labeled data point $(y, \boldsymbol{x})$ is then a co-operative game $G = (\mathcal{M}, v)$ in which:*

$$v(\mathcal{S}) = \begin{cases} 1 & \text{if } w(\mathcal{S}) \geq \gamma, \\ 0 & \text{otherwise.} \end{cases}$$

for a binary classifier ensemble vote score $0 \leq w(\mathcal{S}) \leq 1$ where $w(\mathcal{S}) = \sum_{M \in \mathcal{S}} w_M$ for any sub-ensemble $\mathcal{S} \subseteq \mathcal{M}$ and cutoff value $0 \leq \gamma \leq 1$.

This definition is the central idea in our work. The models in the ensemble play a cooperative voting game to classify the data point correctly. When the data point is classified correctly the payoff is 1, an incorrect classification results in a payoff of 0. Each model casts a weighted vote about the data point and our goal is going to be to quantify the value of individual models in the final decision. In other words, we would like to measure how individual binary classifiers *contribute on average* to the correct classification of a specific data point.

Definition 7. ***Dual ensemble game.*** *Let $\mathcal{M}$ be a set of binary classifiers. A dual ensemble game for a labeled data point $(y, \boldsymbol{x})$ is then a co-operative game $G = (\mathcal{M}, \widetilde{v})$ in which:*

$$\widetilde{v}(\mathcal{S}) = \begin{cases} 1 & \text{if } \widetilde{w}(\mathcal{S}) \geq \widetilde{\gamma}, \\ 0 & \text{otherwise.} \end{cases}$$

for a binary classifier ensemble vote score $0 \leq \widetilde{w}(\mathcal{S}) \leq 1$ where $\widetilde{w}(\mathcal{S}) = \sum_{M \in \mathcal{S}} (1/m - w_M)$ for any sub-ensemble $\mathcal{S} \subseteq \mathcal{M}$ and inverse cutoff value $0 \leq \widetilde{\gamma} \leq 1$ defined by $\widetilde{\gamma} = 1 - \gamma$.

If the sum of classification weights for the binary classifiers is below the cutoff value the models in the ensemble misclassify the point, lose the ensemble game and as a consequence receive a payoff that is zero. In such scenarios it is interesting to ask: how can we describe the role of models in the misclassification? The dual ensemble game is derived from the original ensemble game in order to characterize this situation.

Definition 8. ***Simplified ensemble game.*** *An ensemble game in simplified form is described by the cutoff value – weight-vector tuple $(\gamma, [w_1, \ldots, w_m])$.*

Definition 9. ***Simplified dual ensemble game.*** *Given a simplified form ensemble game $(\gamma, [w_1, \ldots, w_m])$, the corresponding simplified dual ensemble game is defined by the cutoff value – weight vector tuple:*

$$(\widetilde{\gamma}, [\widetilde{w}_1, \ldots, \widetilde{w}_m]) = (1 - \gamma, [1/m - w_1, \ldots, 1/m - w_m])$$

The simplified forms of ensemble and dual ensemble games are compact data structures which can describe the game without the models themselves and the enumeration of every sub-ensemble.

3.2 Solution concepts for model valuation

Earlier we defined the binary classification problem with an ensemble as a weighted voting game, which is a type of co-operative game. Now we will argue that *solution concepts* of co-operative games are suitable for the valuation of individual models which form the binary classifier ensemble.

Definition 10. ***Solution concept.*** *A solution concept defined for the ensemble game $G = (\mathcal{M}, v)$ is a function which assigns the real value $\Phi_M(\mathcal{M}, v) \in \mathbb{R}$ to each binary classifier $M \in \mathcal{M}$.*

The scalar Φ_M can be interpreted as the value of the individual binary classifier M in the ensemble $\mathcal{M}$. In the following we discuss axiomatic properties of solution concepts which are desiderata for model valuation functions. We also discuss the practical implications of Axioms - in the context of model valuation in binary ensemble games.

Axiom 1. ***Null classifier.*** *A solution concept has the null classifier property if $\forall\ \mathcal{S} \subseteq \mathcal{M} : v(\mathcal{S} \cup \{M\}) = v(\mathcal{S})) \rightarrow \Phi_M(\mathcal{M}, v) = 0$.*

Having the null classifier property means that a binary classifier which always has a zero marginal contribution in any sub-ensemble will have a zero payoff. Specifically, this means that the classifier never casts the deciding vote to correctly classify the data point when it is added to a sub-ensemble. Conversely, in the dual ensemble game the model never contributes to the misclassification of the data point.

Axiom 2. ***Efficiency.*** *A solution concept satisfies the efficiency property if $v(\mathcal{M}) = \sum_{M \in \mathcal{M}} \Phi_M(\mathcal{M}, v)$.*

A solution concept which satisfies the efficiency property decomposes the payoff of the ensemble such way that the value of individual models altogether equals to that of the ensemble.

Axiom 3. ***Symmetry.*** *A solution concept has the symmetry property if $\forall \mathcal{S} \subseteq \mathcal{M} \setminus \{M', M''\} : v(\mathcal{S} \cup \{M'\}) = v(\mathcal{S} \cup \{M''\})$ implies that $\Phi_{M'}(\mathcal{M}, v) = \Phi_{M''}(\mathcal{M}, v)$.*

In our setting the symmetry property of a solution concept means that two binary classifiers which always contribute to the possible sub-ensembles the same marginal contribution will receive the same valuation in a symmetric model valuation scheme.

Axiom 4. ***Linearity.*** *A solution concept has the linearity property if given two ensemble games $G = (\mathcal{M}, v)$ and $G' = (\mathcal{M}, v')$ for any binary classifier $M \in \mathcal{M}$ it holds that $\Phi_M(\mathcal{M}, v + v') = \Phi_M(\mathcal{M}, v) + \Phi_M(\mathcal{M}, v')$.*

The linearity property is advantageous as the evaluation of machine learning models is not based on the performance on a single data point. This property allows the solution of data point level ensemble games and the aggregation of the solution vectors to quantify the performance of classifiers in the ensemble based on the dataset.

3.3 The Shapley value

The Shapley value [31] of a classifier is the average marginal contribution of the model over the possible different permutations in which the ensemble can be formed [4]. It is a solution concept which satisfies Axioms 1-4 and the only solution concept which is uniquely characterized by Axioms 3 and 4.

Definition 11. ***Shapley value.*** *The Shapley value of binary classifier M in the ensemble $\mathcal{M}$, for the data point level ensemble game $G = (\mathcal{M}, v)$ is defined as*

$$\Phi_M(v) = \sum_{\mathcal{S} \subseteq \mathcal{M} \setminus \{M\}} \frac{|\mathcal{S}|!\ (|\mathcal{M}| - |\mathcal{S}| - 1)!}{|\mathcal{M}|!} (v(\mathcal{S} \cup \{M\}) - v(\mathcal{S})).$$

Calculating the exact Shapley value for every model in an ensemble game would take $O(m!)$ time which is computationally unfeasible in large scale settings. In order to mitigate this we discuss a range of approximation approaches in detail which can give Shapley value estimates in $O(m)$ and $O(m^2)$ time.

3.3.1 Multilinear extension (MLE) approximation of the Shapley value. The *MLE* approximation of the Shapley value in a voting game [7, 27] can be used to estimate the Shapley value in the ensemble game and its dual game. Let us define the expected value and the variance of the classifier weights as: $\mu = \sum_{i=1}^{m} w_j/m$ and $\nu = \sum_{i=1}^{m} (w_j - \mu)^2/m$. For a classifier $M \in \mathcal{M}$ the multi-linear approximation of the unnormalized Shapley value is computed by:

$$\widehat{\Phi}_M \propto \int_{-\infty}^{\gamma} \frac{1}{\sqrt{2\pi\nu}} \exp\left(-\frac{(x-\mu)^2}{2\nu}\right) \mathrm{d}x - \int_{-\infty}^{\gamma - w_M} \frac{1}{\sqrt{2\pi\nu}} \exp\left(-\frac{(x-\mu)^2}{2\nu}\right) \mathrm{d}x.$$

This approximation assumes that the size of the game is large (many classifiers in the ensemble in our case) and also that μ has an *approximate normal distribution.* Calculating all of the approximate Shapley values by *MLE* takes $O(m)$ time.

3.3.2 Monte Carlo (MC) approximation of the Shapley value. The MC approximation [20, 21] given the ensemble $\mathcal{M}$ estimates the Shapley value of the model $M \in \mathcal{M}$ by the average marginal contribution over uniformly sampled permutations.

$$\hat{\Phi}_M = \mathbb{E}_{\theta \sim \Theta}[v(\mathcal{S}_\theta^M \cup \{M\}) - v(\mathcal{S}_\theta^M)] \tag{1}$$

In Equation (1) Θ is a uniform distribution over the $m!$ permutations of the binary classifiers and $\mathcal{S}_\theta^M$ is the subset of models that appear before the classifier M in permutation θ. Approximating the Shapley value requires the generation of p classifier permutations, marginal contribution calculations to the ensemble and averaging those contributions – this takes $O(mp)$ time.

Data: $[w_1, \ldots, w_m]$ – Weights of binary classifiers.
γ – Cutoff value.
δ – Numerical stability parameter.
μ – Expected value of weights.
ν – Variance of weights.
Result: $(\widehat{\Phi}_1, \ldots, \widehat{\Phi}_m)$ – Approximate Shapley values.

for $j \in \{1, \ldots, m\}$ **do**
  $\widehat{\Phi}_j \leftarrow 0$
  for $k \in \{1, \ldots, m-1\}$ **do**
    $a \leftarrow (\gamma - w_j/k$
    $b \leftarrow (\gamma - \delta)/k$
    $\widehat{\Phi}_j \leftarrow \widehat{\Phi}_j + \frac{1}{\sqrt{2\pi\nu/k}} \int_a^b \exp(-k\frac{(x-\mu)^2}{2\nu})\mathrm{d}x$
  end
end

Algorithm 1: Expected marginal contribution approximation of Shapley values based on [7].

3.3.3 Voting game approximation of the Shapley value. The newly introduced ensemble games are a variant of voting games [26], hence we can use the *Expected Marginal Contributions (EMC)* approximation [7]. This procedure sums the expected marginal contributions of a model to fixed size ensembles – it is summarized with pseudo-code by Algorithm 1.

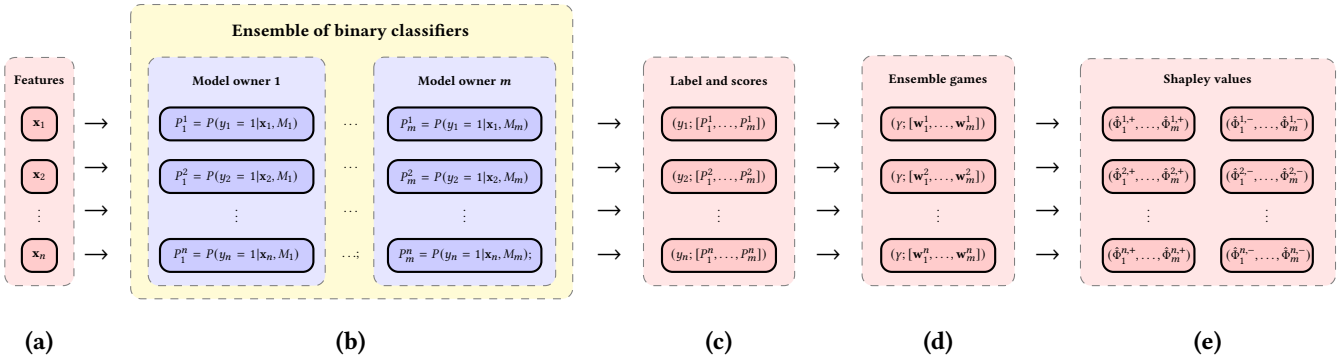

Figure 2: The approximate average Shapley value calculation pipeline. (a) Given the set of labeled datapoints the data owner sends the features of data points to the binary classifiers in the ensemble. (b) The models in the the ensemble score each of the data points independently and send the probability scores back to the oracle. (c) Given the label and the scores the oracle defines the ensemble game on each of the data points. (d-e) The ensemble games and the dual ensemble games are solved by the oracle and approximate Shapley values of models are calculated in each of the data point level ensemble games.

The algorithm iterates over the individual model weights and initializes the approximate Shapley values as zeros (lines 1-2). For each ensemble size it calculates the expected contribution of the model to the ensemble. This expected contribution is the probability that classifier becomes the marginal voter. These marginal contributions are added to the approximate Shapley value of the model (lines 3-8). Using this voting game specific Shapley value approximation technique takes $O(m^2)$ time. However, it is the most accurate Shapley value approximation technique in terms of absolute error of the Shapley value estimates [7].

4 MODEL VALUATION WITH THE SHAPLEY VALUE OF ENSEMBLE GAMES

In this section we overview *Troupe* the mechanism of the expected marginal contributions based algorithm used for model valuation in ensembles. The flowchart in Figure 2 gives a high-level summary of our model valuation approach. We also point out the theoretical properties of the proposed model valuation framework later.

4.1 The model valuation algorithm

The Shapley value based model evaluation procedure described by Algorithm 2 uses $\mathcal{D} = \{(\mathbf{x}_1, y_1); \ldots; (\mathbf{x}_n, y_n)\}$, a set of n datapoints with binary labels, for evaluating the ensemble $\mathcal{M}$ given a cutoff value $0 \leq \gamma \leq 1$ and a numerical stability parameter $\delta \in \mathbb{R}^{0+}$.

We set m the number of models and n the number of labeled data points used in the model valuation (lines 1-2). We iterate over all of the data points (line 3) and given a data point for all of the models we compute the individual model weights in an ensemble game using the label, features and the model itself (line 4) – see Algorithm 3 in Appendix A for the exact details. Using the individual model weights in the ensemble game we calculate the expected value and variance of the data point (lines 5-6).

If the sum of individual model weights is higher than the cutoff value (line 7) the Shapley values in the ensemble game are approximated by Algorithm 1 for the data point (line 8). The Shapley values in the dual ensemble game are defined by a vector of zeros (line 9). If the point is misclassified (line 10) a *dual ensemble game* is defined for the data point. This requires the redefined cutoff value $\widetilde{\gamma}$, expected value $\widetilde{\mu}$ and variance $\widetilde{\nu}$ of the individual model weight vector (lines 11-13). We exploit the linearity of expectation

Data: $\mathcal{D}$ – Labeled data points.
$\mathcal{M}$ – Set of binary classifiers.
γ – Cutoff value.
δ – Numerical stability parameter.
Result: $\underbrace{\left\{(\widehat{\Phi}_0^{1,+}, \ldots, \widehat{\Phi}_m^{1,+}), \ldots, (\widehat{\Phi}_0^{n,+}, \ldots, \widehat{\Phi}_m^{n,+})\right\}}_{\text{Shapley value vectors for ensemble games.}}$
$\underbrace{\left\{(\widehat{\Phi}_0^{1,-}, \ldots, \widehat{\Phi}_m^{1,-}), \ldots, (\widehat{\Phi}_0^{n,+}, \ldots, \widehat{\Phi}_m^{n,+})\right\}}_{\text{Shapley value vectors for dual ensemble games.}}$

1 $m \leftarrow |\mathcal{M}|$
2 $n \leftarrow |\mathcal{D}|$
3 **for** $i \in \{1, \ldots, n\}$ **do**
4 $\quad [w_1^i, \ldots, w_m^i] \leftarrow$ **Get Weights**$((\mathbf{x}_i, y_i); \{M_1, \ldots, M_m\})$
5 $\quad \mu \leftarrow \sum_{j=1}^{m} w_j^i / m$
6 $\quad \nu \leftarrow \sum_{j=1}^{m} (w_j^i - \mu)^2 / m$
7 $\quad$ **if** $\mu > \gamma / m$ **then**
8 $\quad\quad (\widehat{\Phi}_0^{i,+}, \ldots, \widehat{\Phi}_m^{i,+}) \leftarrow$ **Shapley** $([w_1^i, \ldots, w_m^i]; \gamma; \delta; \mu; \nu)$
9 $\quad\quad (\widehat{\Phi}_0^{i,-}, \ldots, \widehat{\Phi}_m^{i,-}) \leftarrow \mathbf{0}$
10 $\quad$ **else**
11 $\quad\quad \widetilde{\gamma} \leftarrow 1 - \gamma$
12 $\quad\quad \widetilde{\mu} \leftarrow 1/m - \mu$
13 $\quad\quad \widetilde{\nu} \leftarrow \nu$
14 $\quad\quad [\widetilde{w}_1^i, \ldots, \widetilde{w}_m^i] \leftarrow [1/m - w_1^i, \ldots, 1/m - w_m^i]$
15 $\quad\quad (\widehat{\Phi}_0^{i,+}, \ldots, \widehat{\Phi}_m^{i,+}) \leftarrow \mathbf{0}$
16 $\quad\quad (\widehat{\Phi}_0^{i,-}, \ldots, \widehat{\Phi}_m^{i,-}) \leftarrow$ **Shapley** $([\widetilde{w}_1, \ldots, \widetilde{w}_m]; \widetilde{\gamma}; \delta; \widetilde{\mu}; \widetilde{\nu})$
17 $\quad$ **end**
18 **end**

Algorithm 2: *Troupe:* Calculating the approximate Shapley value of the classifiers in ensemble and dual ensemble games.

and the location invariance of the variance. The individual model weight vector is redefined (line 14) in order to quantifiy the role of models in the erroneous decision. The original ensemble game Shapley values are initialized with zeros (line 15) and using the new parametrization of the game we approximate the dual ensemble game Shapley values for the models in the ensemble (line 16) by the use of Algorithm 1. The algorithm outputs the data point level ensemble and dual ensemble game Shapley values for every model in the ensemble.

4.2 Measuring ensemble heterogeneity

The data point level Shapley values of ensemble games and their dual can be aggregated to measure the importance of models in ensemble level decisions. Using the information entropy of the Shapley values we also define quantities which summarize the heterogeneity of ensembles.

4.2.1 The average conditional Shapley value. In order to quantify the role of models in classification and misclassification we calculate the average Shapley value of models in the ensemble and dual ensemble games conditional on the success of classification. The sets $\mathcal{N}^+$ and $\mathcal{N}^-$ contain the indices of classified and misclassified data points. Using the cardinality of these sets $n^+ = |\mathcal{N}^+|$, $n^- = |\mathcal{N}^-|$ and the data point level Shapley values output by Algorithm 1 we can estimate the *conditional* role of models in classification and misclassification by averaging the approximate Shapley values using Equations (2) and (3).

$$(\overline{\Phi}_1^+, \ldots, \overline{\Phi}_m^+) = \sum_{i \in \mathcal{N}^+} (\widehat{\Phi}_0^{i,+}, \ldots, \widehat{\Phi}_m^{i,+})/n^+ \tag{2}$$

$$(\overline{\Phi}_1^-, \ldots, \overline{\Phi}_m^-) = \sum_{i \in \mathcal{N}^-} (\widehat{\Phi}_0^{i,-}, \ldots, \widehat{\Phi}_m^{i,-})/n^- \tag{3}$$

If a component of the average Shapley value vector in Equation (2) is large compared to other components the corresponding model has an important role in the correct classification decisions of the ensemble. Correspondingly, a large component in Equation (3) corresponds to a model which is responsible for a large number of misclassifications.

4.2.2 The Shapley entropy. The Shapley values of the ensemble games can be interpreted as a probability distribution over the classifiers in the ensemble. Using the information entropy of average Shaple values of ensemble and dual ensemble games we can define metrics that quantify the heterogeneity of ensembles with respect to classification and misclassification decisions.

$$H^+ = -\sum_{j=1}^{m} \overline{\Phi}_j^+ \log(\overline{\Phi}_j^+) \quad \text{and} \quad H^- = -\sum_{j=1}^{m} \overline{\Phi}_j^- \log(\overline{\Phi}_j^-)$$

If H^+ is high it means that the all of the models are responsible for classification decisions to the same extent. A low H^+ value implies that a subset of models in the ensemble have a dominant role in correct classification decisions. The H^- score can be interpreted analogously with respect to misclassifications decisions.

4.3 Theoretical properties

Our framework utilizes labeled data instances to approximate the importance of classifiers in the ensemble and this has important implications. In the following we discuss how the size of the dataset and the number classifiers affects the Shapley value approximation error and the runtime of *Troupe*.

4.3.1 Bounding the average approximation error. Our discussion focuses on the average conditional Shapley value in ensemble games. However, analogous results can be obtained for the Shapley values computed from dual ensemble games.

Definition 12. ***Approximation error.*** *The Shapley value approximation error of model $M \in \mathcal{M}$ in an ensemble game is defined as $\Delta\Phi_M^+ = \widehat{\Phi}_M^+ - \Phi_M^+$.*

Definition 13. ***Average approximation error.*** *Let use denote the Shapley value approximation errors of model $M \in \mathcal{M}$ calculated from the dataset $\mathcal{D}$ of correctly classified points as $\Delta\Phi_M^{1,+}, \ldots, \Delta\Phi_M^{n,+}$. The average approximation error is defined by $\overline{\Delta\Phi}_M^+ = \sum_{i=1}^n \Delta\Phi_M^{i,+}/n$.*

Theorem 1. ***Average conditional Shapley value error bound.*** *If the Shapley value approximation errors $\Delta\Phi_M^{1,+}, \ldots, \Delta\Phi_M^{n,+}$ of model $M \in \mathcal{M}$ calculated by Algorithm 2 from the dataset $\mathcal{D}$ are independent random variables than for any $\varepsilon \in \mathbb{R}^+$ Inequality* (4) *holds.*

$$P(|\overline{\Delta\Phi}_M^+ - \mathbb{E}[\overline{\Delta\Phi}_M^+]| \geq \varepsilon) \leq 2\exp\left(-\sqrt{\frac{n^2 \cdot m \cdot \varepsilon^4 \cdot \pi}{8}}\right) \tag{4}$$

Proof. Let us first note the fact that every absolute Shapley approximation value is bounded by the inequality described in Lemma 1.

Lemma 1. ***Approximate Shapley value bound.*** *As [7] states the approximation error of the Shapley value in a single voting game is bounded by Inequality* (5) *when the expected marginal contributions approximation is used.*

$$-\sqrt{\frac{8}{m\pi}} \leq \Delta\Phi_M^+ \leq \sqrt{\frac{8}{m\pi}} \tag{5}$$

Using Lemma 1, the fact that *Troupe* is based on the expected marginal contributions approximation and that the Shapley values of a model in different ensemble games are independent random variables we can use Hoeffding's second inequality [13] for bounded non zero-mean random variables:

$$P(|\overline{\Delta\Phi}_M^+ - \mathbb{E}[\overline{\Delta\Phi}_M^+]| \geq \varepsilon) \leq 2\exp\left(-\frac{2\varepsilon^2 n^2}{\sum_{i=1}^{n}\left[\left(\sqrt{\frac{8}{m\pi}}\right) - \left(-\sqrt{\frac{8}{m\pi}}\right)\right]}\right)$$

$$P(|\overline{\Delta\Phi}_M^+ - \mathbb{E}[\overline{\Delta\Phi}_M^+]| \geq \varepsilon) \leq 2\exp\left(-\frac{2\varepsilon^2 n^2}{2n\sqrt{\frac{8}{m\pi}}}\right)$$

□

Theorem 2. ***Confidence interval of the expected average approximation error.*** *In order to acquire an $(1-\alpha)$-confidence interval of $\mathbb{E}[\overline{\Delta\Phi}_M^+] \pm \varepsilon$ one needs a labeled dataset $\mathcal{D}$ of correctly classified data points for which n the cardinality of $\mathcal{D}$, satisfies Inequality* (6).

$$n \geq \sqrt{\frac{8\ln^2\left(\frac{\alpha}{2}\right)}{\varepsilon^4 m\pi}} \tag{6}$$

Proof. The probability $P(|\overline{\Delta\Phi}_M^+ - \mathbb{E}[\overline{\Delta\Phi}_M^+]| \geq \varepsilon)$ in Theorem 1 equals to the level of significance for the confidence interval $\mathbb{E}[\overline{\Delta\Phi_M}^+] \pm \varepsilon$. Which means that Inequality (7) holds for the significance level α.

$$\alpha \leq 2\exp\left(-\sqrt{\frac{n^2 \cdot m \cdot \varepsilon^4 \cdot \pi}{8}}\right). \tag{7}$$

Solving inequality (7) for n yields the cardinality of the dataset (number of correctly classified data points) required for obtaining the confidence interval described in Theorem 2. □

The inequality presented in Theorem 2 has two important consequences regarding the bound:

(1) Larger ensembles require less data in order to give confident estimates of the Shapley value for individual models.
(2) The dataset size requirement is sublinear in terms of confidence level and quadratic in the precision of the Shapley value approximation.

4.3.2 Runtime and memory complexity. The runtime and memory complexity of the proposed model valuation framework depends on the complexity of the main evaluation phases. We assume that our framework operates in a single-core non distributed setting.

Scoring and game definition. The scoring of a data point takes $O(1)$ time. Scoring the data point with all models takes $O(m)$. Scoring the whole dataset and defining games both takes $O(nm)$ time and $O(nm)$ space respectively.

Approximation and overall complexity. Calculating the expected marginal contribution of a model to a fixed size ensemble takes $O(1)$ time. Doing this for all of the ensemble sizes takes $O(m)$ time. Approximating the Shapley value for all models requires $O(m^2)$ time and $O(m)$ space. Given a dataset of n points this implies a need for $O(nm^2)$ time and $O(nm)$ space. This is also the overall time and space complexity of the proposed framework.

5 EXPERIMENTAL EVALUATION

In this section, we show that the *Troupe* approximates the average of approximate Shapley values precisely. We provide evidence that Shapley values are a useful decision metric for ensemble creation. Our results illustrate that model importance and complexity are correlated, and that Shapley values are able to identify adversarial models in the ensemble.

Our evaluation is based on various real world binary graph classification tasks [29]. Specifically, we use datasets collected from *Reddit*, *Twitch* and *GitHub* – the descriptive statistics of these datasets are enclosed in Appendix B as Table 4.

5.1 The precision of approximation

The Shapley value approximation performance of *Troupe* was compared to that of various other estimation schemes [20, 27] discussed earlier. We used an ensemble of logistic regressions where each classifier was trained on features extracted with a whole graph embedding technique [30, 36, 38]. We utilized 50% of the graphs for training and calculated the average conditional Shapley values of ensemble games and dual ensemble games using the remaining 50% of the data. The experimental details are discussed in Appendix C.

5.1.1 Approximate model importance measurement. In Table 3 we summarized the absolute percentage error values (compared to exact Shapley values in ensemble games) obtained with the various approximations schemes. (i) Our empirical results support that *Troupe* consistently computes accurate estimates of the ground truth model evaluations across datasets and classifiers in the ensemble. (ii) The high quality of estimates suggests that the approximate Shapley values of models extracted by *Troupe* can serve as a proxy decision metric for ensemble building and model selection.

5.1.2 Approximate ensemble heterogeneity measurement. Using the previous conditional average Shapley values of ensemble and dual ensemble games we calculated the Shapley entropy scores – see Table 2. Our empirical findings suggest that ensemble heterogeneity varies when it comes to correct and incorrect decisions. We also see evidence that: (i) *Troupe* gives the best estimates of the exact ensemble heterogeneity scores; (ii) *MLE* consistently underestimates the ensemble heterogeneity while *MC* overestimates the ensemble heterogeneity (which is a result of the non-exact computation).

Table 2: Exact and approximate Shapley entropy values of ensemble and dual ensemble games on the whole graph embedding expert system based classification tasks. Bold numbers denote the best approximation.

		Exact	Troupe	MLE	$MC_{p=10^2}$	$MC_{p=10^3}$
Reddit	H^+	2.1860	**2.1864**	2.1971	2.1818	2.1929
	H^-	2.1907	**2.1911**	2.1972	2.1765	2.1860
Twitch	H^+	2.1963	**2.1962**	2.1972	2.1873	2.1949
	H^-	2.1953	**2.1956**	2.1972	2.1890	2.1947
GitHub	H^+	2.1953	**2.1954**	2.1972	2.1907	2.1944
	H^-	2.1961	**2.1961**	2.1971	2.1869	2.1966

5.2 Ensemble building

Our earlier results suggested that the approximate Shapley values output by *Troupe* (and other estimation methods) can be used as decision metrics for ensemble building. We demonstrate this by selecting a high performance subset of a random forest in a forward fashion using the estimated model valuation scores. The test performance (measured by the AUC scores) of these classifiers as a function of subensemble size is plotted on Figure 3. Our results suggest that Troupe has a material advantage over competing approximation schemes on the Twitch and Reddit datasets. The exact experimental details and the model selection procedure is discussed in Appendix D.

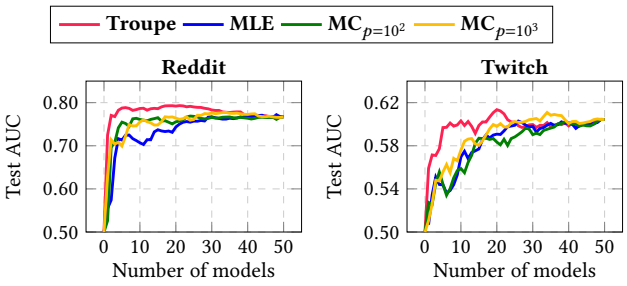

Figure 3: The test graph classification performance of binary classifier ensembles as a function of ensembles selected by our forward model building procedure. Models are added iteratively to the ensemble based on the average approximate Shapley values extracted from the ensemble games.

Table 3: Absolute percentage error of average conditional Shapley values obtained by approximation techniques (rows) for the graph classifiers (columns) in the ensemble game. Bold numbers note the lowest error on each dataset – classifier pair.

	Approximation	FEATHER	Graph2Vec	GL2Vec	NetLSD	SF	LDP	GeoScatter	IGE	FGSD
Reddit	**Troupe**	**1.23**	**2.35**	8.18	**0.99**	**2.64**	**2.31**	**1.64**	**4.85**	**1.49**
	MLE	3.20	23.61	32.62	4.19	5.34	7.97	7.12	5.42	7.61
	MC $p = 10^3$	12.57	30.94	13.26	8.67	32.76	12.32	11.62	16.36	12.78
	MC $p = 10^3$	4.71	5.38	**3.41**	1.67	3.03	5.51	4.34	4.97	3.82
Twitch	**Troupe**	**0.28**	**3.33**	**1.18**	**2.53**	**1.62**	**0.59**	**1.48**	**0.25**	1.19
	MLE	5.22	5.44	3.05	8.32	2.38	1.92	3.14	4.85	5.77
	MC $p = 10^2$	2.37	10.40	6.76	7.07	15.79	6.36	13.99	23.96	**0.39**
	MC $p = 10^3$	2.32	4.60	2.96	2.67	2.53	2.73	6.31	0.27	3.89
GitHub	**Troupe**	**2.68**	**0.18**	**2.61**	**1.41**	**1.49**	1.23	**1.88**	**2.36**	1.04
	MLE	9.22	5.12	3.76	3.27	9.71	7.46	5.08	4.04	0.73
	MC $p = 10^2$	5.91	9.37	4.67	9.82	8.78	8.34	13.66	28.95	**0.76**
	MC $p = 10^3$	3.35	6.09	7.70	3.26	2.84	**0.79**	6.67	2.51	1.12

5.3 Model complexity and influence

We fitted a voting expert which consisted of neural networks with heterogeneous model complexity. The exact experimental settings are detailed in Appendix E. The distribution of normalized average Shapley values is plotted on Figure 4 for the ensemble and dual ensemble games conditioned on the number of hidden layer neurons. These results imply that more complex models with a larger number of free parameters receive higher Shapley values in both classes of games. In simple terms complex models contribute to correct and incorrect classification decisions at a disproportionate rate. We provide additional evidence for this in Appendix E.

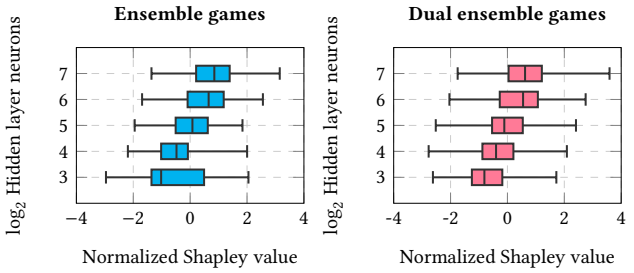

Figure 4: The distribution of normalized Shapley values for neural networks in ensemble and dual ensemble games conditional on the number of neurons (Reddit dataset).

5.4 Identifying adversarial models

Ensembles can be formed by the model owners submitting their classifiers voluntarily to a machine learning market. We investigated how adversarial behaviour of model owners affects the Shapley values of classifiers. In Figure 5 we plotted the mean Shapley value of models which are adversarial and which are not in a scenario where a fraction of model owners mixed their predictions with noise. Even with a negligible amount of adversarial noise the Shapley values of adversarial models drop in the ensemble games considerably – we provide full experimental details in Appendix F.

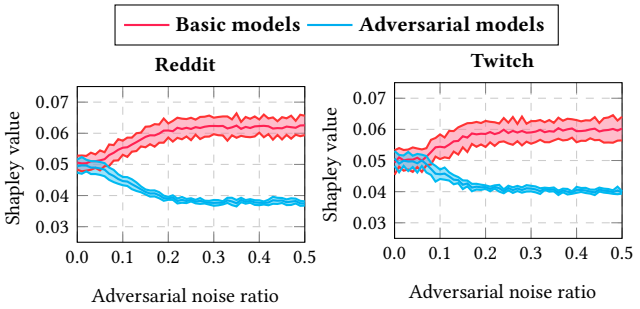

Figure 5: The mean Shapley value (standard errors added) of adversarial and base classifiers in ensemble games as a function of adversarial noise ratio mixed to predictions.

5.5 Scalability

We plotted the mean runtime of Shapley value approximations calculated from 10 experimental runs for an ensemble with $m = 2^5$ and dataset with $n = 2^8$ on Figure 6. All results were obtained with our open-source framework. These average runtimes of the approximation techniques are in line with the known and new theoretical results discussed in Sections 2 and 4. The precise estimates of Shapley values obtained with *Troupe* come at a time cost.

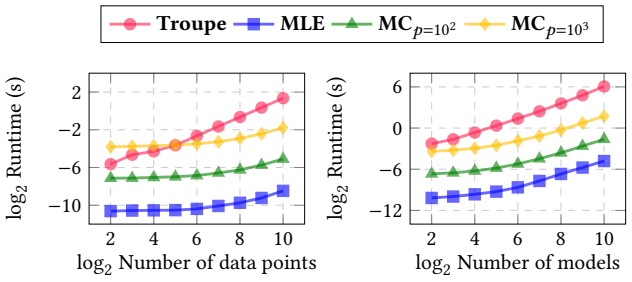

Figure 6: The runtime of Shapley value approximation in ensemble games as a function of dataset size and number of classifiers in the ensemble.

6 CONCLUSIONS AND FUTURE DIRECTIONS

We proposed a new class of cooperative games called *ensemble games* in which binary classifiers cooperate in order to classify a data point correctly. We postulated that solving these games with the Shapley value results in a measure of individual classifier quality. We designed *Troupe* a voting game inspired approximation algorithm which computes the average of Shapley values for every classifier in the ensemble based on a dataset. Using these estimated model valuation scores we introduced metrics which describe the model heterogeneity of the ensemble with respect to predictive accuracy. We provided theoretical results about the sample size needed for precise estimates of the model quality.

We have demonstrated that our algorithm can provide accurate estimates of the Shapley value and the ensemble heterogeneity measures on real world social network data. We illustrated how the Shapley values of the models can be used to create small sized but highly accurate graph classification ensembles. We presented evidence that complex models have an important role in the classification decisions of ensembles. We showcased that our framework can identify adversarial models in the classification ensemble.

We think that our contribution opens up venues for novel theoretical and applied data mining research about ensembles. We theorize that our model valuation framework can be generalized to ensembles which consist of multi-class predictors using a one versus many approach. Our work provides an opportunity for the design and implementation of real world machine learning markets where the payoff of a model owner is a function of the individual model value in the ensemble.

ACKNOWLEDGEMENTS

Benedek Rozemberczki was supported by the Centre for Doctoral Training in Data Science, funded by EPSRC (grant EP/L016427/1). We would also like to thank Maria Astefanoaei, Matt Chapman-Rounds, Oliver Kiss, Jonathan Mallinson, James Owers, and Lauren Watson for their useful comments.

REFERENCES

[1] Marco Ancona, Cengiz Oztireli, and Markus Gross. 2019. Explaining Deep Neural Networks with a Polynomial Time Algorithm for Shapley Value Approximation. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Vol. 97. PMLR, 272–281.

[2] Chen Cai and Yusu Wang. 2018. A Simple Yet Effective Baseline for Non-Attributed Graph Classification. *ICLR 2019 Workshop on Representation Learning on Graphs and Manifolds* (2018).

[3] Javier Castro, Daniel Gómez, and Juan Tejada. 2009. Polynomial Calculation of the Shapley Value Based on Sampling. *Computers and Operations Research* 36, 5 (2009), 1726 – 1730.

[4] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. 2011. Computational Aspects of Cooperative Game Theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 5, 6 (2011), 1–168.

[5] Hong Chen and Hisashi Koga. 2019. GL2Vec: Graph Embedding Enriched by Line Graphs with Edge Features. In *International Conference on Neural Information Processing*. Springer, 3–14.

[6] Nathan de Lara and Pineau Edouard. 2018. A Simple Baseline Algorithm for Graph Classification. In *Advances in Neural Information Processing Systems*.

[7] Shaheen S Fatima, Michael Wooldridge, and Nicholas R Jennings. 2008. A Linear Approximation Method for the Shapley Value. *Artificial Intelligence* 172, 14 (2008), 1673–1699.

[8] Alexis Galland and Marc Lelarge. 2019. Invariant Embedding for Graph Classification. In *ICML Workshop on Learning and Reasoningwith Graph-Structured Data*.

[9] Feng Gao, Guy Wolf, and Matthew Hirn. 2019. Geometric Scattering for Graph Data Analysis. In *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97. 2122–2131.

[10] Mike Gashler, Christophe Giraud-Carrier, and Tony Martinez. 2008. Decision Tree Ensemble: Small Heterogeneous is Better than Large Homogeneous. In *Seventh International Conference on Machine Learning and Applications*. IEEE, 900–905.

[11] Amirata Ghorbani and James Zou. 2019. Data Shapley: Equitable Valuation of Data for Machine Learning. In *International Conference on Machine Learning*. 2242–2251.

[12] Amirata Ghorbani and James Zou. 2020. Neuron Shapley: Discovering the Responsible Neurons. *arXiv preprint arXiv:2002.09815* (2020).

[13] Wassily Hoeffding. 1994. Probability Inequalities for Sums of Bounded Random Variables. In *The Collected Works of Wassily Hoeffding*. Springer, 409–426.

[14] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J. Spanos. [n.d.]. Towards Efficient Data Valuation Based on the Shapley Value. In *Proceedings of Machine Learning Research*. 1167–1176.

[15] Aleksandar Lazarevic and Zoran Obradovic. 2001. Effective Pruning of Neural Network Classifier Ensembles. In *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, Vol. 2. IEEE, 796–801.

[16] Dennis Leech. 1998. *Computing Power Indices for Large Voting Games: A New Algorithm.* Technical Report.

[17] Dennis Leech. 2003. Computing Power Indices for Large Voting Games. *Management Science* 49, 6 (2003), 831–837.

[18] Stan Lipovetsky and Michael Conklin. 2001. Analysis of Regression in Game Theory Approach. *Applied Stochastic Models in Business and Industry* 17, 4 (2001), 319–330.

[19] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Curran Associates Inc., 4768–4777.

[20] Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. 2013. Bounding the Estimation Error of Sampling-Based Shapley Value Approximation. *arXiv preprint arXiv:1306.4265* (2013).

[21] Irwin Mann and Lloyd S Shapley. 1960. Values of Large Games, IV: Evaluating the Electoral College by Monte Carlo Techniques. (1960).

[22] Gonzalo Martínez-Muñoz and Alberto Suárez. 2006. Pruning in Ordered Bagging Ensembles. In *Proceedings of the 23rd International Conference on Machine Learning*. 609–616.

[23] Fatiha Mokdad, Djamel Bouchaffra, Nabil Zerrouki, and Azzedine Touazi. 2015. Determination of an Optimal Feature Selection Method Based on Maximum Shapley Value. In *2015 15th International Conference on Intelligent Systems Design and Applications (ISDA)*. IEEE, 116–121.

[24] Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, and Yang Liu. 2017. Graph2Vec: Learning Distributed Representations of Graphs. (2017).

[25] Ludwig Johann Neumann, Oskar Morgenstern, et al. 1947. *Theory of Games and Economic Behavior*. Vol. 60. Princeton University Press Princeton.

[26] Martin J Osborne and Ariel Rubinstein. 1994. *A Course in Game Theory.* MIT press.

[27] Guillermo Owen. 1972. Multilinear Extensions of Games. *Management Science* 18, 5-part-2 (1972), 64–79.

[28] Miklós Pintér. 2011. Regression Games. *Annals of Operations Research* 186, 1 (2011), 263–274.

[29] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. 2020. Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*. ACM, 3125–3132.

[30] Benedek Rozemberczki and Rik Sarkar. 2020. Characteristic Functions on Graphs: Birds of a Feather, from Statistical Descriptors to Parametric Models. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*. ACM, 1325–1334.

[31] Lloyd S Shapley. 1953. A Value for n-Person Games. *Contributions to the Theory of Games* 2, 28 (1953), 307–317.

[32] Julian Stier, Gabriele Gianini, Michael Granitzer, and Konstantin Ziegler. 2018. Analysing Neural Network Topologies: A Game Theoretic Approach. *Procedia Computer Science* 126 (2018), 234–243.

[33] Erik Strumbelj and Igor Kononenko. 2010. An Efficient Explanation of Individual Classifications Using Game Theory. *The Journal of Machine Learning Research* 11 (2010), 1–18.

[34] Xin Sun, Yanheng Liu, Jin Li, Jianqi Zhu, Xuejie Liu, and Huiling Chen. 2012. Using Cooperative Game Theory to Optimize the Feature Selection Problem. *Neurocomputing* 97 (2012), 86–93.

[35] Mukund Sundararajan and Amir Najmi. 2020. The Many Shapley Values for Model Explanation. In *International Conference on Machine Learning*. PMLR, 9269–9278.

[36] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, Alexander Bronstein, and Emmanuel Müller. 2018. NetLSD: Hearing the Shape of a Graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2347–2356.

[37] Kagan Tumer and Joydeep Ghosh. 1996. Error Correlation and Error Reduction in Ensemble Classifiers. *Connection Science* 8, 3-4 (1996), 385–404.
[38] Saurabh Verma and Zhi-Li Zhang. 2017. Hunt for the Unique, Stable, Sparse and Fast Feature Learning on Graphs. In *Advances in Neural Information Processing Systems*. 88–98.
[39] Gilad Zlotkin and Jeffrey S. Rosenschein. 1994. Coalition, Cryptography, and Stability: Mechanisms for Coalition Formation in Task Oriented Domains. In *AAAI*. 432–437.

A CONDITIONAL SCORING ALGORITHM

Data: $(\mathbf{x}, y)$ – Labeled data point.
$\{M_1, \ldots, M_m\}$ – Set of binary classifiers.
Result: $[w_1, \ldots, w_m]$ – Weights of individual models.
for $j \in \{1, \ldots, m\}$ **do**
  if $y = 1$ **then**
    $w_j \leftarrow P(y = 1 \mid M_j, \mathbf{x})/m$
  else
    $w_j \leftarrow P(y = 0 \mid M_j, \mathbf{x})/m$
  end
end

Algorithm 3: Calculating the individual model weights in an ensemble game given a data point and a classifier ensemble.

B GRAPH CLASSIFICATION DATASETS

Table 4: Descriptive statistics of the binary graph classification datasets taken from [29] used for the evaluation of our proposed framework. These datasets are fairly balanced, while the graphs are hetereogeneous with respect to size, density and diameter.

	Classes		Nodes		Density		Diameter	
Dataset	**Positive**	**Negative**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
Reddit	521	479	11	93	0.023	0.027	2	18
Twitch	520	480	14	52	0.039	0.714	2	2
GitHub	552	448	10	942	0.004	0.509	2	15

C APPROXIMATION EXPERIMENT DETAILS

The features of the graphs were extracted with whole graph embedding techniques implemented in the open source *Karate Club* framework [29]. Given a set of graphs $\mathcal{G} = (G_1, \ldots, G_n)$ whole graph embedding algorithms [30, 36] learn a mapping $g : \mathcal{G} \rightarrow \mathbb{R}^d$ which delineate the graphs $G \in \mathcal{G}$ to a d dimensional metric space. We utilized the following whole graph embedding and statistical finger printing techniques:

(1) **FEATHER** [30] uses the characteristic function of topological features as a graph level statistical descriptor.
(2) **Graph2Vec** [24] extracts tree features from the graph.
(3) **GL2Vec** [5] distills tree features from the dual graph.
(4) **NetLSD** [36] derives characteristic of graphs using the heat trace of the graph spectra.
(5) **SF** [6] utilizes the largest eigenvalues of the graph Laplacian matrix as an embedding.
(6) **LDP** [2] sketches the histogram of local degree distributions.
(7) **GeoScattering** [9] applies the scattering transform to various structural features (e.g. degree centrality).
(8) **IGE** [8] combines graph features from local degree distributions and scattering transforms.
(9) **FGSD** [38] sketches the Moore-Penrose spectrum of the normalized graph Laplacian with a histogram.

The embedding techniques used the *default settings* of the *Karate Club* library, each embedding dimension was column normalized and the graph features were fed to the *scikit-learn* implementation of logistic regression. This classifier was trained with ℓ_2 penalty cost, we chose an SGD optimizer and the regularization coefficient λ was set to be 10^{-2}.

D MODEL SELECTION EXPERIMENTAL DETAILS

From each graph we extracted Weisfeiler-Lehman tree features [24] and kept those topological patterns which appeared in at least 5 graphs. Using the counts of these features in graphs we define statistical descriptors. Using 40% of the graphs we trained a random forest with 50 classification trees, each tree was trained on 20 randomly sampled Weisfeiler-Lehman features – we used the default settings of *scikit-learn*. We calculated the average conditional Shapley value of classifiers in the ensemble games defined on using 30% of the data. We ordered the classifiers by the approximate Shapley values in decreasing order, created subensembles in a forward fashion and calculated the predictive performance of the resulting subensembles on the remaining 30% of the graphs.

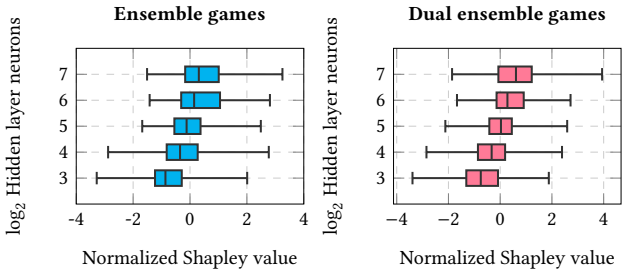

Figure 7: The distribution of normalized Shapley values for neural networks in ensemble and dual ensemble games conditional on the number of neurons (Twitch dataset).

E MODEL COMPLEXITY AND INFLUENCE EXPERIMENTAL DETAILS

We extracted the Weisfeiler-Lehman features which appeared in at least 5 graphs in the datasets. We defined the frequency based graph descriptors used in Appendix D. The models were trained with 50% of the dataset and the average Shapley values were calculated from the remaining 50% of graphs.

E.1 Neural network ensembles

We created an ensemble of $m = 10^3$ neural networks using *scikit-learn* – each of these had a single hidden layer. Each model received

20 randomly selected frequency features as input and had a randomly chosen number of hidden layer neurons – we uniformly sampled this hyperparameter from $\{2^3, 2^4, 2^5, 2^6, 2^7\}$. Individual neural networks were trained by minimizing the binary cross-entropy with SGD for 200 epochs with a learning rate of 10^{-2}. The results in Section 5 and the ones in Figure 7 demonstrate that complexity (number of free parameters) is correlated with relative model importance.

E.2 Random forest ensembles

We created a random forest ensemble of $m = 10^3$ classification trees using *scikit-learn*. Each tree in the ensemble received 20 randomly selected Wesifeiler-Lehman count features as input. We used the default settings of *scikit-learn* except for the maximal depth which we fixed to be 4. Using the Reddit and Twitch datasets we plotted on Figures 8 and 9 the mean normalized Shapley value of classification trees obtained by *Troupe* in the ensemble games and dual ensemble games conditioned on the number of leaves that the trees have.

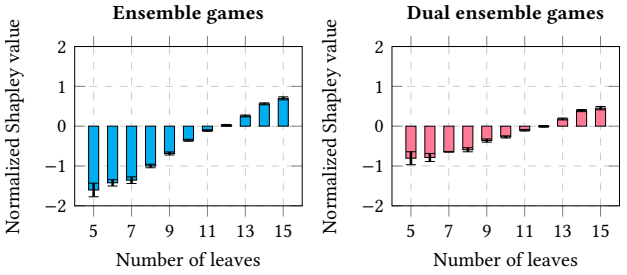

Figure 8: The mean and standard error of normalized Shapley values for classification trees in ensemble and dual ensemble games conditional on the number of leaves in the tree (Reddit dataset).

Our results support the claim made earlier that more complex models (higher number of tree leaves) contribute to correct and incorrect classification decisions with a higher probability. However, in this case the higher number of leaves might be a random artifact of sampling better quality features which are more discriminative.

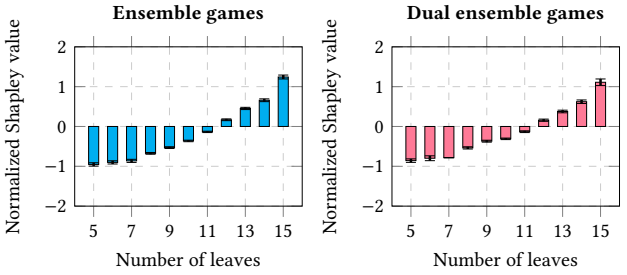

Figure 9: The mean and standard error of normalized Shapley values for classification trees in ensemble and dual ensemble games conditional on the number of leaves in the tree (Twitch dataset).

F IDENTIFYING ADVERSARIAL MODELS EXPERIMENTAL DETAILS

We used the Weisfeiler-Lehman tree features described in Appendices D and E and trained a random forest ensemble of 20 classifiers using 50% of the data – each of these models utilized a random subset of 20 features and had a maximal tree depth of 4. The Shapley values were calculated from the remaining 50% of the data using *Troupe*. We artificially corrupted the predictions for 10 of the classification trees by mixing the outputted probability values with noise that had $\mathcal{U}(0, 1)$ distribution. The corrupted predictions were a convex combination of the original prediction and the noise – we call the weight of the noise as the adversarial noise ratio. Given an adversarial noise ratio we calculated the mean Shapley value (with standard errors) for the adversarial and normally behaving classification trees in the ensemble.