Trade-off between Payoff and Model Rewards in Fair Collaborative Machine Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

This paper investigates the problem of fairly trading off between payoff and model rewards in collaborative *machine learning* (ML) where parties aggregate their datasets together to obtain improved ML models over that of each party. Supposing parties can afford the optimal model trained on the aggregated dataset, we propose an allocation scheme that distributes the payoff fairly. Notably, the same scheme can be derived from two different approaches based on (1) desirable properties of the parties' payoffs or (2) that of the underlying payoff flows from one party to another. While the former is conceptually simpler, the latter can be used to handle the practical constraint on the budgets of parties. In particular, we propose desirable properties for achieving a fair adjustment of the payoff flows that can trade off between the model reward's performance and the payoff reward. We empirically demonstrate that our proposed scheme is a sensible solution in several scenarios of collaborative ML with different budget constraints.

1 Introduction

While *machine learning* (ML) has proven its usefulness in a wide range of applications, a significant amount of diverse training data is often required to achieve desirable performance. Although this may not be an issue for very large corporations, it can hardly be met by other businesses. In practice, the dataset coming from an individual data owner, called a *party*, (e.g., a company) is often limited in size (e.g., due to the size of the customer base) and diversity (e.g., due to the constraints in the geographical location and/or the customer demographics). Hence, there is a strong urge for parties to obtain high-quality ML models by performing *collaborative ML*: training models on their aggregated dataset which is often of much larger quantity and higher diversity than those of individual parties.

However, the goal of obtaining improved models may not outweigh the concern of *fairness* if all parties receive the same model while their contributions differ (due to the difference in the size and quality of different parties' datasets). For instance, a party with a larger contribution feels "under-rewarded" if all parties receive models of *equal performances* without any other *compensation/rewards for the difference in their contributions*. Hence, a "fair" reward allocation scheme is desirable to give all parties enough incentives to join the collaboration. Let us classify existing works on fair reward allocation schemes into 2 groups depending on whether there exists an external source of reward to be distributed among parties. This source of reward can be thought of as an *external buyer* of the model who is interested in the model but does not contribute data, so he/she is willing to pay a reward to parties in the collaboration to obtain the model.

As an example, a company distributes monetary payoffs to its users for contributing user data to train a model. Then, one can view users as parties who collaborate to train the model and the company as an external buyer of the model. In this case, one can compensate for the difference in users' contributions by fairly distributing the total monetary payoff from the company to users according

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

to the value of each user's data to the model. This value can be measured by a data valuation method, e.g., the Shapley value [4] and its variants to incorporate the underlying data distribution [3], replication-robustness [5], and the federated learning setting [16].

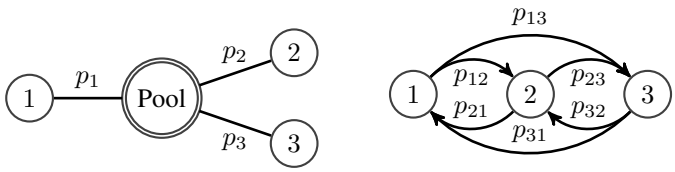
In contrast, we focus on the other group of works when the obtained model from the collaboration of several parties is not sold to any external buyer (e.g., due to privacy or the fear of losing the competitive advantage), i.e., there is not any external source of reward to be distributed among parties. In the literature, fairness is maintained by (i) requiring parties to pay a participation fee to be distributed back to all parties [12] or (ii) ensuring that a party with less valuable data receives a model with lower performance than those with more valuable data [14]. In brief, parties receive different *monetary payoffs* (produced by the participation fee) in the former and different models (called *model rewards*) in the latter. However, there are 2 main drawbacks: the trading-off between monetary payoffs vs. model rewards and the uniqueness of the allocation scheme. **First**, the former approach [12] prevents parties with an insufficient monetary budget from joining the collaboration regardless of the usefulness of their datasets to other parties. The latter approach [14] prevents parties with low-quality datasets from obtaining a good model regardless of their monetary expense capability. Both cases are undesirable. **Second**, the uniqueness of the allocation schemes in the works of [12, 14] is not discussed. Hence, it raises the question of whether there are other allocation schemes that satisfy the same fairness properties in [12, 14]. If such schemes exist, there can be undesirable disagreement among parties in choosing the allocation scheme to use. On the contrary, if an allocation scheme is uniquely defined from a set of fairness properties (e.g., the celebrated Shapley value [13]), parties which agree on these properties are certain to adopt the (unique) allocation scheme.

In this work, we resolve the above 2 problems by proposing fairness properties that define not only a unique allocation scheme of the monetary payoff but also a unique way of trading off between payoff and model rewards. et is noted that as both payoff and model rewards are allowed, our problem subsumes the two problems in [12, 14]. Nevertheless, our solution is not a generalization of existing solutions [12, 14] because even in the special cases of our problem that are equivalent to those in [12, 14], our solution is different from that of [12, 14]. In particular, our solution satisfies the linearity property while the works of [12, 14] do not. The outline of our paper is as follows.

In Sec. 3, we consider parties without budget constraints, which implies that all parties can afford the highest-quality model from the collaboration. Thus, the scheme is reduced to only allocating monetary payoffs to parties. In this case, we propose 3 desirable properties which uniquely determine the fair payoff for each party (in Sec. 3.1). The key observation is that payoffs should *fairly compensate for the difference between the contributions of parties to the models they receive*. While these payoffs suffice to characterize a payoff allocation scheme, it does not give us insights into the underlying payoff flows transferred from one party to another. The latter is crucial for us to trade off between payoff and model rewards when parties have budget constraints. Therefore, in Sec. 3.2, we propose 4 desirable properties for fair payoff flows from one party to another by drawing inspiration from the Shapley value [13]. These properties uniquely determine a new solution called *the conditional Shapley value* which specifies the payoff flows from one party to another. Interestingly, we arrive at the same set of payoffs as those in Sec. 3.1. Hence, the fairness of our proposed payoff allocation scheme can be consistently explained with both the proposed properties of the payoffs (Sec. 3.1) and those of the underlying payoff flows transferred from one party to another (Sec. 3.2).

In Sec. 4, we consider parties with budget constraints, i.e., their expense cannot exceed their budgets. Consequently, there may exist parties that cannot afford the highest-quality model if their budgets are below the required amounts of payment. We propose to trade off between payoff and model rewards such that parties can sacrifice the model reward, i.e., obtaining a worse model, to meet the budget. In particular, we construct a set of desirable properties that uniquely define an adjustment of the allocation scheme to meet the budgets while retaining the properties of the fair payoff allocation.

In Sec. 5, we empirically demonstrate through several machine learning problems that our proposed allocation scheme exhibits several sensible characteristics. For example, even if parties do not have any budget, they are still motivated to join the collaboration because they are able to obtain improved models from the collaboration as long as they have valuable datasets. Furthermore, an increase in the budget of a party not only benefits itself but also benefits other parties in terms of improved models and/or higher monetary payoffs.



(a) Payoff reward representation. (b) Payoff flow representation.

Figure 1: Diagrams of the representations of a payoff allocation scheme.

2 Background

Let us consider a set of n parties, denoted as $\mathcal{N} \triangleq \{1, 2, 3 \ldots, n\}$, collaborating with each other by aggregating their datasets to train ML models. Any subset of $\mathcal{C} \subset \mathcal{N}$ is a *coalition* while $\mathcal{N}$ is the *grand coalition*. To ease the notation clutter, we overload the notation i to denote both party i and the singleton set $\{i\}$. The aggregated training dataset of a coalition $\mathcal{C}$ is denoted as $\mathcal{D}_\mathcal{C} \triangleq \cup_{i \in \mathcal{C}} \mathcal{D}_i$ where $\mathcal{D}_i$ denotes the dataset of party i. The performance of a model trained on $\mathcal{D}_\mathcal{C}$ is denoted as $s(\mathcal{C})$. We assume the *monotonicity property* of s with respect to (w.r.t.) $\mathcal{C}$: $\forall \mathcal{C} \subset \mathcal{C}' \subset \mathcal{N},\ s(\mathcal{C}') \geq s(\mathcal{C})$; and $s(\mathcal{N}) > 0$. In the experiments (in Sec. 5), the model performance is measured by the prediction accuracy of models on a validation set which is common across different parties. It reflects the assumption that parties solve the same ML problem. Our approach also works with other performance metrics such as the information gain in the work of [14].

The *value function*, denoted as v, is a mapping from a coalition $\mathcal{C} \in \mathcal{N}$ to its *value* $v(\mathcal{C})$ such that $v(\emptyset) = 0$. In this work, $v(\mathcal{C})$ is considered as the monetary payoff. It is the amount of money to buy a model with the same performance as the optimal model trained on $\mathcal{D}_\mathcal{C}$. Adopting the assumption from the work of [12], we define it using an exchange rate $\nu > 0$: $v(\mathcal{C}) \triangleq \nu s(\mathcal{C})$. The *monotonicity property* of v follows from that of s. More importantly, a model with the performance $s(\mathcal{C})$ can be exchanged fairly with a payoff $v(\mathcal{C}) \triangleq \nu s(\mathcal{C})$.

Given a value function v, the Shapley value is the solution to distributing the payoff $v(\mathcal{N})$ to parties in $\mathcal{N}$ that satisfies 4 desirable properties: efficiency, symmetry, linearity, and dummy party (see App. A). Hence, it is the foundation of many data valuation methods [4, 3, 6, 7, 17]. The Shapley value of a party i is defined as $\varphi_i(v) \triangleq \frac{1}{n} \sum_{\mathcal{C} \in \mathcal{N} \setminus i} \binom{n-1}{|\mathcal{C}|}^{-1} \Delta_v(i|\mathcal{C})$ where $\Delta_i(i|\mathcal{C}) \triangleq v(\mathcal{C} \cup i) - v(\mathcal{C})$ is the marginal contribution of party i to coalition $\mathcal{C}$. It is noted that to use the Shapley value, there should exist a payoff $v(\mathcal{N})$ to be distributed to all parties in $\mathcal{N}$, i.e., there exists an external source of reward (see the classification of existing works in Sec. 1). In contrast, we assume that such an external source of reward does not exist in this work, which renders the direct use of the Shapley value challenging. Thus, we propose a new perspective of distributing a payoff of value 0 to all parties in $\mathcal{N}$ (by allowing parties to receive negative payoff rewards) such that these payoff rewards *compensate for the difference in the contributions among parties*.

In this work, an *allocation scheme* specifies both payoff and model rewards that each party receives. We do not distinguish between two models having the same performance. Thus, for a party $i \in \mathcal{N}$, the scheme specifies the value (equivalently, the performance) of the model and the payoff value it receives. To receive a negative payoff is to make a payment.

We first consider the case where there are not any budget constraints, so all parties obtain the same model performance $s(\mathcal{N})$ (i.e., the optimal model trained on $\mathcal{D}_\mathcal{N}$) while they may receive different payoff rewards to account for the different contributions in Sec. 3. Then, in Sec. 4, we consider the case that parties have budget constraints such that they may not afford the model of performance $s(\mathcal{N})$. In this case, not only do parties receive payoffs constrained by the budgets, but they also receive different model rewards to maintain fairness.

3 Parties without Budget Constraints

In this section, parties have no budget constraints, so they can expend any amount of payoff to receive the optimal model trained on $\mathcal{D}_\mathcal{N}$ (i.e., with the performance of $s(\mathcal{N})$). Hence, the problem is reduced

to an allocation scheme of only the payoff reward. In particular, different parties receive different (and possibly negative) payoff rewards to compensate for the difference in their contributions.

Payoff reward vector. Given a value function v, let $p_i(v)$ denote the payoff reward that party i receives. Then, a payoff allocation scheme specifies a payoff reward vector $\mathbf{p}(v) \triangleq (p_i(v))_{i=1}^n$. While party i is rewarded if $p_i(v) > 0$, it makes a payment of value $-p_i(v)$ if $p_i(v) < 0$. In practice, it can be implemented as a 2-step process: first, parties with negative payoffs make payments to an intermediate pool; second, parties with positive payoffs take their payoffs from the pool (Fig. 1a).

Payoff flow. One may consider defining a payoff allocation scheme by the payoff transferred from one party to another which we call *payoff flows* (Fig. 1b). However, there can be different ways of transferring payoffs among parties which result in the same payoff reward vector. For example, a payoff reward vector remains unchanged if we increase each payoff flow from party i to party $\tau(i)$ by the same amount where $\tau(i) = i + 1$ for all $i < n$ and $\tau(n) = 1$ (i.e., increasing the payoff along a closed loop). Furthermore, regardless of how payoffs are transferred among parties, what matters, in the end, is the payoff reward each party receives which is determined by the payoff reward vector $\mathbf{p}$. Therefore, in the next section, we propose 3 desirable properties that uniquely determine a fair payoff reward vector (hence, a payoff allocation scheme).

3.1 Fair Payoff Reward Vector

We propose the following 3 desirable properties for the fairness of a payoff reward vector $\mathbf{p}$: balance, linear value, and dummy party.

Balance (BL). Recall that there is not any external source that pumps additional payoffs into $\mathcal{N}$ (or takes payoffs from $\mathcal{N}$). Therefore, the *balance* property states that $\sum_{i\in\mathcal{N}} p_i(v) = 0$.

In this work, we adopt the Shapley value as the measure of a party's contribution. In the literature, the Shapley values are often scaled to ensure all of the rewards are distributed (i.e., being efficient) [12, 14]. Nonetheless, we cannot employ this scaling transformation due to the above balance property. If all parties have positive Shapley values, any payoff rewards proportional to the Shapley values (by scaling) are either all positive or all negative, which violates the balance property.

Linear value (LV). Therefore, instead of requiring the payoff reward to be proportional to the Shapley value, we require the difference in payoff rewards to be proportional to that in the Shapley values:

$$\exists \beta > 0,\ \forall \{i, j\} \subset \mathcal{N}, \quad p_i(v) - p_j(v) = \beta\left(\varphi_i(v) - \varphi_j(v)\right)\ . \tag{1}$$

We call it the *linear value* property since it implies $p_i(v)$ is a linear function of $\varphi_i(v)$. Unlike the scaling transformation, the linear transformation allows a party with a positive Shapley value to have a negative payoff reward. Furthermore, it realizes our intuition of fairness: *the payoff rewards compensate the difference in the contributions between parties*. In particular, a big difference in the contributions (measured by the Shapley value) of two parties leads to a big difference in their payoff rewards. Besides, parties with higher contributions receive higher payoff rewards (since $\beta > 0$).

Dummy party (DP). Party $i \in \mathcal{N}$ is a dummy party if $v(\mathcal{C} \cup i) = v(\mathcal{C})$ for all $\mathcal{C} \subset \mathcal{N} \setminus i$. It means that a dummy party can be considered as an external buyer that does not contribute any data and is only interested in buying the model. Therefore, we expect a dummy party to pay the whole value of the model it receives from the collaboration, i.e., $v(\mathcal{N})$ (since parties have no budget constraints). In other words, if party i is a dummy party, then it makes a payment of value $v(\mathcal{N})$, i.e., $p_i(v) = -v(\mathcal{N})$.

The following lemma shows that these properties characterize a unique payoff reward vector, which is proven in App. B.

Lemma 3.1. *The vector $(p_i(v))_{i=1}^n$ of payoff rewards that satisfies the balance, the linear value, and the dummy party properties only exists for $\beta = n$ (in (1)). Furthermore, when $\beta = n$, such a payoff reward vector is uniquely determined:*

$$\forall i \in \mathcal{N},\ p_i(v) = n\varphi_i(v) - v(\mathcal{N})\ . \tag{2}$$

Corollary 3.2. *If we allocate payoff rewards to parties in $\mathcal{N}$ according to (2), then a party has a negative payoff reward (i.e., making a payment) if and only if its Shapley value is less than the average of the Shapley values of all parties, i.e., $v(\mathcal{N})/n$. This implies that a party receives a negative payoff if its contribution is less than the average of the contributions of all parties.*

In (2), the payoff reward $p_i(v)$ of a party is defined based on the Shapley value of that party and $v(\mathcal{N})$, so the symmetry and linearity properties (w.r.t v) of $p_i(v)$ follow from those of the Shapley value. We also study another property, namely *replication-robustness* [1, 5], in App. C where we show that the proposed payoff allocation scheme is replication-robust for parties with negative payoffs.

Remark 3.3. The Shapley value represents the contribution of a party to the optimal model trained on $\mathcal{D}_\mathcal{N}$. However, in our scenario, there should be n such models (all of them are the same) distributed to n parties. Thus, the contribution of a party to n models should be $n\varphi_i(v)$, which explains the scaling factor of $\beta = n$ in Lemma 3.1.

We recall that a payoff allocation scheme is fully determined by the payoff reward vector $\mathbf{p}$. However, the above properties of $\mathbf{p}$ do not give us enough information to handle budget constraints as explained later in Sec. 4. Thus, the next section describes a particular way to specify payoff flows from one party to another such that it gives rise to the same desirable payoff vector as the one in Lemma 3.1.

3.2 Fair Payoff Flows

Let us decompose the payoff reward $p_i(v) \in \mathbb{R}$ into the *total cost flow* $p_{i,-}(v) \geq 0$ (from party i to others) that party i needs to pay to obtain the optimal model trained on $\mathcal{D}_\mathcal{N}$ and the *total revenue flow* $p_{i,+}(v) \geq 0$ (from others to party i) that party i obtains by contributing its dataset to the models other parties receive (the revenue does not deduct any cost):

$$p_i(v) = p_{i,+}(v) - p_{i,-}(v) \ . \tag{3}$$

Since $p_{i,-}$ is the total cost that party i incurs to acquire the additional performance of the model from the datasets of other parties $\mathcal{N} \setminus i$, the total cost flow can be decomposed into

$$p_{i,-}(v) = \textstyle\sum_{j \in \mathcal{N} \setminus i} p_{ij}$$

where the payoff flow p_{ij} denotes the cost that party i pays to party j for party j's contribution. Equivalently, it is the revenue that party j obtains from party i. Hence, the total revenue flow is decomposed into

$$p_{i,+}(v) = \textstyle\sum_{j \in \mathcal{N} \setminus i} p_{ji} \ .$$

Note that the payoff flow $p_{ij}(v)$ can be viewed as either a cost of party i or a revenue of party j. Therefore, to construct all payoff flows between parties, we choose to only examine the costs of all parties because it is easier to argue about the total cost of a party (see the *fair cost* property below) than the total revenue of a party.

In particular, for a party $i \in \mathcal{N}$, we are interested in the (cost) payoff flows $(p_{ij}(v))_{j \in \mathcal{N} \setminus i}$. As there are many different ways of transferring payoff flows from one party to another that result in the same payoff reward vector $\mathbf{p}$, it is an ill-posed problem to reconstruct payoff flows solely from $\mathbf{p}$ in Lemma 3.1. Therefore, we propose the following desirable properties of fair payoff flows that are able to not only uniquely determine $(p_{ij}(v))_{j \in \mathcal{N} \setminus i}$ ($p_{ii}(v) = 0$ in our construction), but also result in the same payoff reward vector in Lemma 3.1.

Fair cost. As explained above, $p_{i,-}(v)$ is the cost that party i incurs to acquire the additional performance of the model from the datasets of other parties $\mathcal{N} \setminus i$. This additional performance is the difference between the performance $s(\mathcal{N})$ of the obtained model trained on $\mathcal{D}_\mathcal{N}$ and the performance $s(i)$ of the model trained on the dataset of party i. Equivalently, the increase in the value of the model is $\nu(s(\mathcal{N}) - s(i)) = v(\mathcal{N}) - v(i)$. Therefore, we suggest the *fair cost* property: the total cost flow $p_{i,-}(v)$ of party i should be equal to the value of the additional performance obtained by party i:

$$p_{i,-}(v) = v(\mathcal{N}) - v(i), \text{ equivalently, } \textstyle\sum_{j \in \mathcal{N} \setminus i} p_{ij}(v) = v(\mathcal{N}) - v(i) \ . \tag{4}$$

Conditional symmetry. If two parties j, k in $\mathcal{N} \setminus i$ have the same marginal contribution to coalition $\mathcal{C} \cup i$ for any $\mathcal{C} \subset \mathcal{N} \setminus \{i, j, k\}$, then the cost party i pays to party j is the same as that party i pays to party k, i.e., $p_{ij}(v) = p_{ik}(v)$. It implies that the cost of party i does not depend on the marginal contributions of other parties to coalitions that do not contain i.

Linearity. Let us consider the value function as a vector of 2^n real numbers with the usual addition and scalar multiplication operations, then for value functions v, w, and $\alpha \in \mathbb{R}$,

$$p_{ij}(v + w) = p_{ij}(v) + p_{ij}(w) \ , \quad p_{ij}(\alpha v) = \alpha p_{ij}(v) \ .$$

Conditional dummy party. A conditional dummy party $j \in \mathcal{N} \setminus i$ given a party i satisfies $v(\mathcal{C} \cup \{i, j\}) = v(\mathcal{C} \cup i)$ for all $\mathcal{C} \subset \mathcal{N} \setminus \{i, j\}$. For a conditional dummy party j given party i, $p_{ij}(v) = 0$. It means that party i does not pay any payoff to a party that does not contribute to any coalition containing i.

The above 4 properties uniquely determine the payoff flows according to the following lemma (proven in App. D).

Lemma 3.4. *There exists a unique solution $(p_{ij}(v))_{j \in \mathcal{N} \setminus i}$ that satisfies the above 4 properties. We call this unique solution the* conditional Shapley value *given party i and denote it as $(\varphi_{j|i}(v))_{j \in \mathcal{N} \setminus i}$. It is defined as follows:*

$$\varphi_{j|i}(v) = \tfrac{1}{n-1} \textstyle\sum_{\mathcal{C} \in \mathcal{N} \setminus \{i,j\}} \binom{n-2}{|\mathcal{C}|}^{-1} \Delta_v(j|\mathcal{C} \cup i) \; . \tag{5}$$

Remark 3.5. Since the formulation of the conditional Shapley value is similar to that of the Shapley value, we can apply existing Shapley-value approximation methods [2, 4, 11] to the conditional Shapley value.

Decomposition of the Shapley value. The Shapley value of a party can be decomposed into the conditional Shapley values and its value as shown in App. E:

$$n\varphi_i(v) = v(i) + \textstyle\sum_{j \in \mathcal{N} \setminus i} \varphi_{i|j}(v) \; . \tag{6}$$

We recall in Remark 3.3 that a party contributes to n models (each of which is distributed to one party in $\mathcal{N}$). Hence, the total contribution of a party is $n\varphi_i(v)$ which is the LHS of (6). On the other hand, the RHS of (6) is the sum of the contributions of party i to other parties $j \in \mathcal{N} \setminus i$ measured by the conditional Shapley values $\varphi_{i|j}(v)$ and its own value $v(i)$. The latter is interpreted as the contribution of party i to the model it receives from the collaboration. It is $v(i)$ because party i can always obtain a model of $v(i)$ from its dataset. In brief, the contribution of party i (to n models for n parties) measured by the Shapley value can be decomposed into the contributions of party i to the models that other parties receive measured by the conditional Shapley values and that to the model party i receives.

We set $p_{ii} = 0$, so by setting $p_{ij}(v) = \varphi_{j|i}(v)$ for $i \neq j$, all payoff flows are fully specified. Then, given a party i, the payoff reward $p_i(v)$ is

$$p_i(v) = p_{i,+}(v) - p_{i,-}(v) = n\varphi_i(v) - v(\mathcal{N}) \tag{7}$$

where the total cost flow is $p_{i,-}(v) \triangleq \sum_{j \in \mathcal{N}} p_{ij}(v) = v(\mathcal{N}) - v(i)$ (see the fair cost property) and the total revenue flow is $p_{i,+}(v) \triangleq \sum_{j \in \mathcal{N}} p_{ji}(v) = n\varphi_i(v) - v(i)$ (from (6)). The payoff rewards specified by (7) are exactly the same as that in Lemma 3.1. Thus, they also satisfy the 3 desirable properties in Sec. 3.1: balance, linear value, and dummy party.

Remark 3.6 (Omnipotent party). Furthermore, if we define an *omnipotent party* as a party i that satisfies $v(\mathcal{C} \cup i) = v(i)$ for all $\mathcal{C} \subset \mathcal{N} \setminus i$. Then, party i does not pay to any other parties, i.e., $p_{i,-} = 0$. It is because other parties $j \in \mathcal{N} \setminus i$ are conditional dummy parties given party i.

In brief, when parties have no budget constraints, we present a fair payoff allocation scheme that arises from either a set of desirable properties of the payoff rewards (Sec. 3.1) or that of the underlying payoff flows (Sec. 3.2). While the former is conceptually simpler, the latter can be utilized in the next section to address the budget constraints of parties.

4 Parties with Budget Constraints

In the previous section, we assume all parties have no budget constraints, so they can afford the optimal model with the performance $s(\mathcal{N})$ (trained on $\mathcal{D}_{\mathcal{N}}$). However, in practice, parties often have budget constraints. Let $b_i \geq 0$ denote the budget of party i. When $-p_i(v) > b_i$, party i cannot afford the optimal model trained on $\mathcal{D}_{\mathcal{N}}$. To allow party i to join the collaboration without sacrificing fairness, we would like to reduce the performance of the model it receives such that its corresponding payoff reward satisfies the budget constraint. This implies that different parties receive models of different performances. We call it the *heterogeneous model rewards*. As we discuss in the following

paragraph, while the reward vector representation based on the Shapley value is not suitable, the payoff flows based on our proposed conditional Shapley value are able to handle this heterogeneity.

The heterogeneous model rewards lead to different contributions of a party to models received by different parties. Since the Shapley value of a party is a single measure of its contribution to the model of value $v(\mathcal{N})$, it is unable to keep track of the different contributions of a party (e.g., contributions to models of value $< v(\mathcal{N})$). Thus, it is challenging to incorporate the budget constraint into the fairness properties of the payoff reward vector in Sec. 3.1 because they rely on the Shapley value which does not distinguish the contributions of one party to different models that other parties receive. In contrast, this can be achieved by taking into account the payoff flows among parties, e.g., with our proposed conditional Shapley value.

Unlike existing works that directly adjust the Shapley value [5, 14, 15], which potentially invalidates some fairness properties (e.g., the linearity property), we choose to adjust the value function without altering the 4 fairness properties that uniquely define the payoff flows (the conditional Shapley values) from the adjusted value function.

Since different parties have different budget constraints, we adjust the value function w.r.t. different parties differently. To do this, we modify slightly the notation of the value function to denote an n-output function: $\overrightarrow{v} = (v_i)_{i=1}^n$ where v_i is the value function w.r.t. party i such that $\overrightarrow{v}(\mathcal{C}) = (v_i(\mathcal{C}))_{i=1}^n, \forall \mathcal{C} \subset \mathcal{N}$. Then, by defining $p_{ij}(\overrightarrow{v}) = p_{ij}(v_i)$ and $p_{i,-}(\overrightarrow{v}) = p_{i,-}(v_i) = v_i(\mathcal{N}) - v_i(i)$ the 4 fairness properties in Sec. 3.1 uniquely define the conditional Shapley value $\varphi_{j|i}(\overrightarrow{v}) = \varphi_{j|i}(v_i)$ (Lemma 3.4). This formulation allows different parties receive models of different performances, e.g., $v_i(\mathcal{N}) \neq v_j(\mathcal{N})$ where $v_i(\mathcal{N})$ is the value of the model that party i receives. Let $v_i(\mathcal{C}) = v_j(\mathcal{C}) = v(\mathcal{C})$ for all i, j, and $\mathcal{C} \subset \mathcal{N}$, then $\varphi_{ij}(\overrightarrow{v}) = \varphi_{ij}(v)$ is the payoff flow when all parties have no budget constraint (i.e., the case in Sec. 3.1). Given v and the budget $\boldsymbol{b} \triangleq (b_i)_{i=1}^n$, we would like to find the adjusted value function $\overrightarrow{v}' \triangleq (v_i')_{i=1}^n$ such that the payoff flows $\varphi_{ij}(\overrightarrow{v}') = \varphi_{ij}(v_i')$ meet the budget constraints, i.e., $-p_i(v_i') \leq b_i,\ \forall i \in \mathcal{N}$.

Let us consider party i which cannot afford the optimal model of value $v(\mathcal{N})$ due to its budget, i.e., $b_i < -p_i(v)$. For party i to meet its budget, we need to increase $p_i(v)$ (equivalently, decreasing $-p_i(v)$) such that the adjusted payoff reward $p_i(v_i') \geq -b_i > p_i(v)$. From the decomposition of payoff rewards into payoff flows in Sec. 3.2, the payoff reward is $p_i(v) = p_{i,+}(v) - p_{i,-}(v)$ (7). It is not possible for party i to increase its total revenue flow $p_{i,+}(v)$ since it comes from other parties. Hence, the only way for party i to increase $p_i(v)$ is by decreasing its total cost flow $p_{i,-}(v)$ to other parties $\mathcal{N} \setminus i$ (i.e., $p_{i,-}(v_i') < p_{i,-}(v)$).

As explained in the fair cost property in Sec. 3.2, $p_{i,-}(v)$ is the cost that party i incurs to acquire the increase in the model performance $v(\mathcal{N}) - v(i)$. Hence, decreasing $p_{i,-}(v)$ can be achieved by decreasing $v(\mathcal{N})$, i.e., the value function evaluated at $\mathcal{N}$. In other words, party i receives a model of reduced performance. However, adjusting solely $v(\mathcal{N})$ may lead to unfairness, so we propose the following properties to define a fair adjustment of the value function.

Feasibility & Individual Rationality. Party i is interested in reducing its cost due to the increase in the value $v(\mathcal{N}) - v(i)$. Therefore, a rational party should receive a model with the (adjusted) value $v_i'(\mathcal{N})$ that is (**feasibility**) at most $v(\mathcal{N})$; and (**individual rationality**) at least $v(i)$ (i.e., not worse than the value $v(i)$ of the model it can obtain by itself). The same argument should apply to any coalition containing i:

$$\forall i \in \mathcal{N},\ \forall \mathcal{C} \subset \mathcal{N} \setminus i\ , \quad v(\mathcal{C} \cup i) \geq v_i'(\mathcal{C} \cup i) \geq v(i)\ . \tag{8}$$

Marginal contribution consistency. As an alternative perspective, party i reduces its cost by reducing the contribution of other parties to the increase in the performance of the model it receives. This can be measured by the conditional Shapley values of other parties given i which consist of the marginal contributions of other parties to any coalition containing i. Therefore, the *marginal contribution consistency* states that the marginal contributions of a party $j \neq i$ to any coalition containing i should be consistently reduced by the same factor:

$$\forall i \in \mathcal{N},\ \exists \lambda_i > 0\ , \forall j \in \mathcal{N} \setminus i\ , \forall \mathcal{C} \subset \mathcal{N} \setminus \{i, j\}\ , \Delta_{v_i'}(j|\mathcal{C} \cup i) = \lambda_i \Delta_v(j|\mathcal{C} \cup i)\ . \tag{9}$$

It means that the adjustment should not unfairly bias the marginal contribution of a party to a coalition.

Weak efficiency. There always exists a party in $\mathcal{N}$ that receives the optimal model trained on $\mathcal{D}_{\mathcal{N}}$ (of value $v(\mathcal{N})$) [14].

The above properties and the budget constraints uniquely define a fair adjustment of the value function as shown in the following lemma (proven in App. F).

Lemma 4.1. *Let $\boldsymbol{\lambda}^* \triangleq (\lambda_i^*)_{i=1}^n$ be the solution to the following linear programming problem:*

$$\begin{array}{lll} \textit{maximize} & \sum_{\{i,j\}\subset\mathcal{N}} \lambda_i \varphi_{j|i}(v) & \\ \textit{subject to} & \lambda_i \sum_{j\in\mathcal{N}\setminus i} \varphi_{j|i}(v) - \sum_{k\in\mathcal{N}\setminus i} \lambda_k \varphi_{i|k}(v) \le b_i, & \forall i \in \mathcal{N} \\ & \lambda_i \in [0,1], & \forall i \in \mathcal{N}\,. \end{array} \tag{P}$$

The above 3 properties and the budget constraints uniquely define the adjusted value function $\vec{v}' = \boldsymbol{\lambda}^* \vec{v} \triangleq (\lambda_i^* v)_{i=1}^n$ where the value function $\lambda_i^* v$ denotes a scaled value function of v by multiplying it with λ_i^*, i.e., the evaluation of $\lambda_i^* v$ at any $\mathcal{C} \subset \mathcal{N}$ is $\lambda_i^* v(\mathcal{C})$.

Given the above adjusted value function, the 4 fairness properties in Sec. 3.1 uniquely define the payoff flows $p_{ij}(\boldsymbol{\lambda}^* \vec{v})$ as the conditional Shapley value $\varphi_{j|i}(\boldsymbol{\lambda}^* \vec{v}) = \lambda_i^* \varphi_{j|i}(v)$ for all $\{i,j\} \subset \mathcal{N}$ due to the linearity property. Hence, the total revenue flow and the total cost flow by distributing the payoff according to the conditional Shapley values $\varphi_{j|i}(\boldsymbol{\lambda}^* \vec{v})$ for all $\{i,j\} \subset \mathcal{N}$ are

$$p_{i,+}(\boldsymbol{\lambda}^* \vec{v}) = \sum_{j\in\mathcal{N}\setminus i} \lambda_j^* \varphi_{i|j}(v) \text{ and } p_{i,-}(\boldsymbol{\lambda}^* \vec{v}) = p_{i,-}(\lambda_i^* v) = \lambda_i^* \sum_{j\in\mathcal{N}\setminus i} \varphi_{j|i}(v), \text{ respectively.}$$

Thus, the LHS of the first set of constraints in (P) are $-p_i(\boldsymbol{\lambda}\vec{v}) = p_{i,-}(\boldsymbol{\lambda}\vec{v}) - p_{i,+}(\boldsymbol{\lambda}\vec{v}),\ \forall i \in \mathcal{N}$, so they are the budget constraints. Furthermore, the value of the model obtained by party i according to the above adjusted value function is $v_i'(\mathcal{N}) = \lambda_i^* v(\mathcal{N}) = v(i) + p_{i,-}(\lambda_i^* v)$, which demonstrates the feasibility & individual rationality property (since $\lambda_i \in [0,1]$). To produce a model with a pre-specified performance (e.g., the validation accuracy), we suggest using early stopping. Starting from the optimal model trained on the dataset $\mathcal{D}_i$ of party i (with the performance $v(i)/\nu = s(i)$), we continue to train the model with the aggregated dataset $\mathcal{D}_\mathcal{N}$ and stop the training when the performance of the model is $\lambda_i^* v(\mathcal{N})/\nu$.

In addition, we can show that parties utilize as much of their budgets as possible to obtain a good model in App. G, which we call the *budget efficiency* property: If $-p_i(v) > b_i$, then $-p_i(\boldsymbol{\lambda}^* \vec{v}) = b_i$.

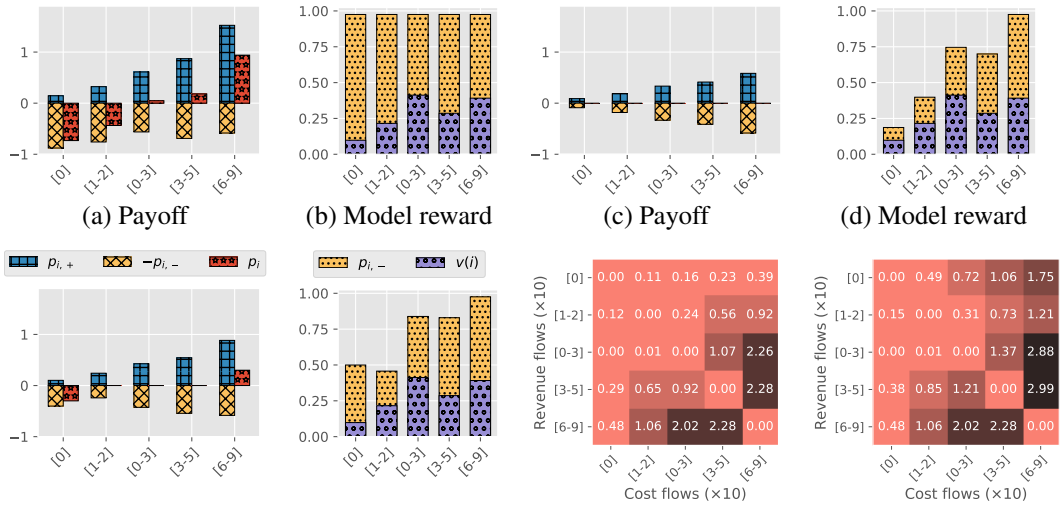
5 Experiments

This section empirically illustrates the proposed allocation scheme of both payoff and model rewards in collaborative ML. To easily interpret the results, we choose the MNIST dataset [9] which contains $70,000$ images of handwritten digits. The dataset is partitioned based on the digit labels. A party, denoted as $[s\text{-}e]$ ($s \le e$), owns a subset of the MNIST training dataset (of size $60,000$) that are labeled with digits $s, s+1, \ldots, e$. There are 5 parties in $\mathcal{N}$: [0] (i.e., [0-0]), [1-2], [0-3], [3-5], and [6-9]. For simplicity, we set $\nu = 1$. The model performance is measured by the prediction accuracy on a separate dataset of size $10,000$.

Additional experiments on the CIFAR-10 [8] dataset and the IMDB movie reviews dataset [10] dataset are in App. H. All experiments share the following desirable observations.

- When there are not any budget constraints, all parties receive models with the highest performance, and parties with more valuable datasets receive higher payoff rewards.
- Even if all parties' budgets are 0, they still benefit from the collaboration (hence, motivated to join the collaboration) by obtaining better models than their own.
- Increasing the budget of a party not only increases the model performance of the party but can also benefit other parties through improved model performance and/or higher payoffs.
- Weak efficiency: There is always a party receiving the best model of value $v(\mathcal{N})$.

Figs. 2a and 2b show the payoff and model rewards when there are not any budget constraints (i.e., $\mathbf{b} = \infty\mathbf{1}$), respectively. In Fig. 2a, party [6-9] receives the largest payoff reward because party [6-9] owns the dataset with the largest number of digits which is not overlapped with any other parties' datasets. Party [0-3] receives a larger payoff reward than that of parties [0] and [1-2] because party [0-3]'s dataset contains the datasets of both parties [0] and [1-2]. In Fig. 2b, all parties receive models with the highest performance as there are not any budget constraints. While the work of [12] can address the case of no budget constraints, it results in a payoff vector



(a) Payoff (b) Model reward (c) Payoff (d) Model reward

(e) Payoff (f) Model reward (g) Payoff flow (h) Payoff flow

Figure 2: Plots of the payoffs, model rewards, and payoff flows in the MNIST experiments with budget constraints: (a,b) $\mathbf{b} = \infty\mathbf{1}$, (c,d,g) $\mathbf{b} = \mathbf{0}$, and (e,f,h) $b_i = 0.3\mathbb{1}_{i=1}$.

$[-0.41, -0.26, -0.02, 0.12, 0.58]$ which is different from ours. It is noted that their solution does not satisfy the linear property while ours does.

Figs. 2c and 2d show the payoff and model rewards when all parties are not allowed to spend any payoff (i.e, $\mathbf{b} = \mathbf{0}$), respectively. From the balance property, the payoff rewards are 0 for all parties (in Fig. 2c). However, in Fig. 2d, they still benefit from the collaboration by obtaining better models (of value $v(i) + p_{i,-}(\lambda_i v)$) than their own models (of value $v(i)$). In particular, we observe that party [6-9] receives the optimal model trained on the aggregated dataset $\mathcal{D}_{\mathcal{N}}$ (of value $v(\mathcal{N})$), as stated by the weak efficiency property. Thus, when monetary payoffs are not allowed, our allocation scheme is still able to incentivize parties to fairly collaborate using their data. Compared to the ρ-Shapley value [14] (which only works in this special case), the model reward allocation in Fig. 2d cannot be obtained from the ρ-Shapley value (by varying its parameters). In fact, unlike our proposed allocation scheme, when $\rho \neq 1$, the ρ-Shapley value does not satisfy the linear property.

Figs. 2e and 2f show the payoff and model rewards when only party [0] has a positive budget of 0.3 (other parties have budgets of 0), i.e., $b_i = 0.3\mathbb{1}_{i=1}$. While it is intuitive that with a positive budget, party [0] improves the performance of its obtained model in comparison to the previous case of no budget, it is interesting that parties [1-2], [0-3], and [3-5] also have the performance of their models improved (by comparing Fig. 2f and Fig. 2d). The reason is that revenues of parties [1-2], [0-3], and [3-5] from party [0] increase (comparing the first row in Fig. 2g and Fig. 2h). These increased revenues are used to further improve the models of [1-2], [0-3], and [3-5], e.g., by getting more contributions from party [6-9] (comparing the last column in Fig. 2g and Fig. 2h). Furthermore, not only does party [6-9] obtain the best model of value $v(\mathcal{N})$ as that in Fig. 2d, but it also obtains an additional payoff reward for its contributions to other parties in Fig. 2e. As seen in Figs. 2g and 2h, the cost flows of party [6-9] (the last rows) are the same between the two cases. Thus, according to our allocation scheme, increasing the budget of a party benefits not only itself but also other parties.

6 Conclusion

This paper presents an allocation scheme for both payoff and model rewards in fair collaboration ML. When there are not any budget constraints, all parties receive the same model trained on the aggregated dataset. On the other hand, they receive different payoff rewards to account for the difference in the contributions of different parties. We construct such a payoff allocation scheme that is fair according to (i) desirable properties of the parties' payoffs or (ii) that of the underlying payoff flows from one party to another. The latter is useful to handle the practical constraint on the budgets of parties. In particular, when there exist budget constraints, we propose a solution to trade off between payoff and model rewards through fair adjustments of the payoff flows. The experiments empirically show that our proposed scheme is reasonable in several collaborative ML scenarios with different budget constraints.

References

[1] A. Agarwal, M. Dahleh, and T. Sarkar. A marketplace for data: An algorithmic solution. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 701–726, 2019.

[2] J. Castro, D. Gómez, and J. Tejada. Polynomial calculation of the Shapley value based on sampling. *Computers & operations research*, 36(5):1726–1730, 2009.

[3] A. Ghorbani, M. Kim, and J. Zou. A distributional framework for data valuation. In *Proc. ICML*, pages 3535–3544. PMLR, 2020.

[4] A. Ghorbani and J. Zou. Data Shapley: Equitable valuation of data for machine learning. In *Proc. ICML*, pages 2242–2251, 2019.

[5] D. Han, S. Tople, A. Rogers, M. Wooldridge, O. Ohrimenko, and S. Tschiatschek. Replication-robust payoff-allocation with applications in machine learning marketplaces. *arXiv preprint arXiv:2006.14583*, 2020.

[6] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. M. Gurel, B. Li, C. Zhang, C. J. Spanos, and D. Song. Efficient task-specific data valuation for nearest neighbor algorithms. In *Proc. VLDB Endowment*, volume 12, pages 1610–1623, 2019.

[7] R. Jia, D. Dao, B. Wang, R. A. Hubis, N. Hynes, N. M. Gürel, B. Li, C. Zhang, D. Song, and C. J. Spanos. Towards efficient data valuation based on the Shapley value. In *AISTATS*, pages 1167–1176. PMLR, 2019.

[8] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[10] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150. Association for Computational Linguistics, June 2011.

[11] S. Maleki, L. Tran-Thanh, G. Hines, T. Rahwan, and A. Rogers. Bounding the estimation error of sampling-based Shapley value approximation. *arXiv preprint arXiv:1306.4265*, 2013.

[12] O. Ohrimenko, S. Tople, and S. Tschiatschek. Collaborative machine learning markets with data-replication-robust payments. *arXiv preprint arXiv:1911.09052*, 2019.

[13] L. S. Shapley. A value for n-person games. *Contributions to the theory of games*, 2(28):307–317, 1953.

[14] R. H. L. Sim, Y. Zhang, M. C. Chan, and B. K. H. Low. Collaborative machine learning with incentive-aware model rewards. In *Proc. ICML*, pages 8927–8936. PMLR, 2020.

[15] S. Tay, X. Xu, C. S. Foo, and B. K. H. Low. Incentivizing collaboration in machine learning via synthetic data rewards. 2022.

[16] X. Xu, L. Lyu, X. Ma, C. Miao, C. S. Foo, and B. K. H. Low. Gradient driven rewards to guarantee fairness in collaborative machine learning. volume 34, 2021.

[17] J. Yoon, S. Arik, and T. Pfister. Data valuation using reinforcement learning. In *Proc. ICML*, pages 10842–10851. PMLR, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] We include the proofs in the Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]