Incentivizing Collaboration in Machine Learning via Synthetic Data Rewards

Sebastian Shenghong Tay,[1,2] Xinyi Xu,[1,2] Chuan Sheng Foo,[2] Bryan Kian Hsiang Low[1]

[1]Department of Computer Science, National University of Singapore, Singapore
[2]Institute for Infocomm Research, A*STAR, Singapore
sebastian.tay@u.nus.edu, xuxinyi@comp.nus.edu.sg, foo_chuan_sheng@i2r.a-star.edu.sg, lowkh@comp.nus.edu.sg

Abstract

This paper presents a novel *collaborative generative modeling* (CGM) framework that incentivizes collaboration among self-interested parties to contribute data to a pool for training a generative model (e.g., GAN), from which synthetic data are drawn and distributed to the parties as rewards commensurate to their contributions. Distributing synthetic data as rewards (instead of trained models or money) offers task- and model-agnostic benefits for downstream learning tasks and is less likely to violate data privacy regulation. To realize the framework, we firstly propose a data valuation function using *maximum mean discrepancy* (MMD) that values data based on its quantity and quality in terms of its closeness to the true data distribution and provide theoretical results guiding the kernel choice in our MMD-based data valuation function. Then, we formulate the reward scheme as a linear optimization problem that when solved, guarantees certain incentives such as fairness in the CGM framework. We devise a weighted sampling algorithm for generating synthetic data to be distributed to each party as reward such that the value of its data and the synthetic data combined matches its assigned reward value by the reward scheme. We empirically show using simulated and real-world datasets that the parties' synthetic data rewards are commensurate to their contributions.

1 Introduction

For the state-of-the-art deep learning models, training with a large quantity of data is important to prevent overfitting and achieve good generalization. So, when there are multiple parties with each owning a dataset sampled from the same distribution, pooling their datasets and training on the pooled dataset would yield an improved *machine learning* (ML) model for every participating party. For example, banks that use ML models to predict their customers' credit ratings (Tsai and Chen 2010) would benefit from pooling their datasets as every bank can now train its ML model on a much larger dataset with more unique customers. This benefit would be even more pronounced in applications where data is difficult/costly to obtain and every party has limited data, such as in medical imaging (Sandfort et al. 2019).

However, data sharing/pooling is challenging in practice due to issues of data privacy (Devereaux et al. 2016) and possibly inequitable benefits from such a form of collaboration (Lo and DeMets 2016). To elaborate on the latter, parties would be more willing to participate in the collaboration if *fairness* is guaranteed. For example, if only one party contributes high-quality data but all parties receive equal rewards, then clearly the other parties benefit disproportionately from the collaboration and that contributing party has no incentive to participate, especially when all parties are self-interested. One may define fairness as every party receiving a reward commensurate to its contribution (however contribution is measured), which incentivizes not only participation but also non-trivial contributions from all parties.

To resolve the above issues, the notion of *collaborative ML* (also referred to as *multi-party ML* (Chen et al. 2020)) allows multiple self-interested parties to mutually benefit from collaboration in data sharing/pooling by incentivizing non-trivial contributions from them while accounting for fairness and data privacy. A prior work of collaborative ML (Sim et al. 2020) has focused on the supervised learning setting where every party contributes training data and receives a model as reward with predictive performance commensurate to its contribution, while another work (Ohrimenko, Tople, and Tschiatschek 2019) has developed a marketplace where parties pay money for better performing ML models on their specific learning tasks and receive money when their contributed data improve the ML models of other parties. A key limitation of these works is that trained ML models are distributed to the parties as rewards, which limits each party's flexibility to experiment with different model architectures and hyperparameters. If more competitive architectures emerge in the future, the parties cannot take advantage of these new architectures without reinitiating the collaboration. Another limitation of distributing trained ML models as rewards is that it precludes the possibility of performing a different learning task on the same dataset as the ML model is tied to a specific task.

One way of overcoming the above limitations is to distribute synthetic data to the parties as rewards (in short, *synthetic data rewards*) instead of trained models. It has been demonstrated that augmenting real data with synthetic data can improve model performance: For example, some works (Bowles et al. 2018; Frid-Adar et al. 2018; Sandfort et al. 2019) have used *generative adversarial networks* (GANs) for data augmentation to improve classification per-

formance on various medical imaging applications such as liver lesions or brain scan segmentations. Distributing synthetic data as rewards is less likely to violate data privacy regulations, unlike sharing real data directly. Also, there is no assumption on whether all parties share a common downstream learning task, the task of interest to each party (e.g., supervised or unsupervised, classification or regression), or the type of ML model used by each party. In particular, with the synthetic data reward, each party can now optimize over model architectures and hyperparameters, train new model architectures emerging in the future, and train separate ML models for different learning tasks.

As a departure from the restriction to supervised learning, this paper presents a novel *collaborative generative modeling* (CGM) framework that incentivizes collaboration among self-interested parties to contribute data to a pool for training an unsupervised generative model (e.g., a GAN), from which synthetic data are drawn and distributed to the parties as rewards (i.e., commensurate to their contributions) instead of sharing real data directly. Like previous works on collaborative ML (Ohrimenko, Tople, and Tschiatschek 2019; Sim et al. 2020), our CGM framework only requires a trusted mediator to train the generative model on the pooled dataset but differs in offering the above-mentioned task- and model-agnostic benefits of synthetic data rewards. Our framework does not consider monetary payment and hence enables participation from parties such as startups or nonprofit organizations with data but limited/no funds. Our work here provides the following specific novel contributions:

- We propose a task- and model-agnostic data valuation function using *maximum mean discrepancy* (MMD) that values (real and/or synthetic) data based on its quantity and quality in terms of its closeness to the true data distribution (Sec. 3), and provide theoretical results guiding the choice of the kernel in our MMD-based data valuation function (Sec. 5);
- We formulate the reward scheme as a linear optimization problem that when solved, guarantees certain incentives such as fairness in the CGM framework (Sec. 4.1);
- We devise a weighted sampling algorithm for generating synthetic data to be distributed to each party as reward such that the value of its data and the synthetic data combined matches its assigned reward value by the reward scheme (Sec. 4.2), and empirically show using simulated and real-world datasets that the parties' synthetic data rewards are commensurate to their contributions (Sec. 6).

Related Work. Collaborative ML is a rich and novel field which uses solution concepts from cooperative game theory and mechanism design. The Shapley value is a commonly adopted solution concept to formalize a notion of fairness in quantifying the contributions of self-interested parties (e.g., via their shared data) (Ohrimenko, Tople, and Tschiatschek 2019; Sim et al. 2020). This line of research inspires several data valuation methods using the Shapley value (Ghorbani, Kim, and Zou 2020; Ghorbani and Zou 2019; Jia et al. 2020; Wang et al. 2020), the core (Yan and Procaccia 2021), influence functions (Richardson, Filos-Ratsikas, and Faltings 2020b), and volume (Xu et al. 2021b). Previous works have used concepts from mechanism design to elicit truthful reporting (Chen et al. 2020; Richardson, Filos-Ratsikas, and Faltings 2020a) and to incentivize sharing data and/or model parameters in *federated learning* (Cong et al. 2020; Kang et al. 2019a,b; Lyu et al. 2020; Yu et al. 2020; Zhan et al. 2020; Xu et al. 2021a). Other works have addressed data privacy (Ding et al. 2021; Hu et al. 2019), adversarial robustness (Hayes and Ohrimenko 2018; So, Guler, and Avestimehr 2020), communication efficiency (Ding et al. 2021), and fairness in Bayesian optimization (Sim et al. 2021). Compared to existing works which have mainly focused on supervised learning, our work investigates a novel task- and model-agnostic setting through the CGM framework that distributes synthetic data as rewards, which to the best of our knowledge has not been considered in the literature.

2 Problem Statement and Notations

The CGM framework comprises a set of honest, non-malicious parties $N := \{1, \ldots, n\}$ and their corresponding datasets $D_1, \ldots, D_n$. Let $\mathcal{D}$ be the true data distribution s.t. each party i may only be able to sample its dataset D_i from a restricted subset of the support of $\mathcal{D}$. Every party i sends D_i to a trusted mediator who trains a generative model (e.g., GAN, variational autoencoder, or flow-based model) on the pooled dataset $\bigcup_{i \in N} D_i$ to produce a distribution $\mathcal{G}$ from which the mediator is able to draw samples. Informally, $\mathcal{G}$ represents an approximation of $\mathcal{D}$. The mediator then generates a large *synthetic dataset* G s.t. each synthetic data point in G is drawn i.i.d. from $\mathcal{G}$. The reward to each party i will be a subset G_i (of synthetic data points) of G and is thus said to be *freely replicable*.[1] In this paper, we use the following definitions from cooperative game theory (Chalkiadakis, Elkind, and Wooldridge 2011): A *coalition* C is a subset of parties (i.e., $C \subseteq N$). The *grand coalition* is the set N of all parties. A *coalition structure* CS is a partition of the parties into disjoint coalitions s.t. $\bigcup_{C \in CS} C = N$, $C \cap C' = \emptyset$ for all $C, C' \in CS$ and $C \neq C'$, and each party cooperates only with parties in the same coalition. A *characteristic function* $v_c : 2^N \rightarrow \mathbb{R}$ maps each coalition to a (real) value of the coalition. Finally, the *reward vector* $(r_1, \ldots, r_n) \in \mathbb{R}^n$ denotes the final reward values assigned to parties $1, \ldots, n$.

Our work here considers the problem of CGM defined as follows: Given the parties' datasets $D_1, \ldots, D_n$ and an appropriate *data valuation* function v (quantitatively capturing the practical assumptions A, B, and C in Sec. 3 on the desired qualities of a dataset), determine the reward vector $(r_1, \ldots, r_n)$ that guarantees certain incentives (Sec. 4), and then distribute subsets of synthetic data points $G_1, \ldots, G_n \subseteq G$ to the respective parties $1, \ldots, n$ as rewards s.t. $v(D_i \cup G_i) = r_i$ (Sec. 4.2).

3 Data Valuation with Maximum Mean Discrepancy (MMD)

Existing metrics for evaluating the approximation quality of generative models do so by measuring some form of dis-

[1] Like digital goods, model or data reward can be replicated at no marginal cost and given to more parties (Sim et al. 2020).

tance between the generated and the true distributions (Borji 2019). One such distance measure is the *maximum mean discrepancy* (MMD) which is a statistic to test whether two distributions $\mathcal{D}'$ and $\mathcal{D}$ are different by measuring the difference of their expected function values based on samples drawn from these distributions (Gretton et al. 2012):

$$\text{MMD}(\mathcal{F},\mathcal{D}',\mathcal{D}) := \sup_{f\in\mathcal{F}} (\mathbb{E}_{x\sim\mathcal{D}}[f(x)] - \mathbb{E}_{x'\sim\mathcal{D}'}[f(x')])$$

where $\mathcal{F}$ is the class of functions f in the unit ball of the reproducing kernel Hilbert space associated with a kernel function k. We defer the discussion on kernels appropriate for use with MMD to (Tay et al. 2021), and will discuss the choice of kernel function k in Sec. 5. Note that $\text{MMD}(\mathcal{F},\mathcal{D}',\mathcal{D}) = 0$ iff $\mathcal{D}' = \mathcal{D}$ (Gretton et al. 2012). Let the *reference dataset* $T := D_1 \cup \ldots \cup D_n \cup G$ denote a union of the pooled dataset with the synthetic dataset and hence represents all available data in our problem setting. Let $t := |T|$ and S be any arbitrary subset of T where $s := |S|$. The unbiased estimate $\text{MMD}^2_u(\mathcal{F},S,T)$ and biased estimate $\text{MMD}^2_b(\mathcal{F},S,T)$ of the squared MMD can be obtained in the form of matrix Frobenius inner products, as shown in (Gretton et al. 2012):

$$\begin{aligned}\text{MMD}^2_u(\mathcal{F},S,T) &= \langle (s(s-1))^{-1}\mathbf{1}_{[x,x'\in S, x\neq x']} - \\ &\quad 2(st)^{-1}\mathbf{1}_{[x\in S,x'\in T]} + (t(t-1))^{-1}\mathbf{1}_{[x,x'\in T, x\neq x']}, \mathbf{K}\rangle \\ \text{MMD}^2_b(\mathcal{F},S,T) &= \langle s^{-2}\mathbf{1}_{[x,x'\in S]} - \\ &\quad 2(st)^{-1}\mathbf{1}_{[x\in S,x'\in T]} + t^{-2}\mathbf{1}_{[x,x'\in T]}, \mathbf{K}\rangle \end{aligned} \quad (1)$$

where $\mathbf{1}_A$ is a matrix with components $1(x,x')$ for all $x,x' \in T$ such that $1(x,x')$ is an indicator function of value 1 if condition A holds and 0 otherwise, and $\mathbf{K}$ is a matrix with components $k(x,x')$ for all $x,x' \in T$.

Our *data valuation* function exploits the negative $\text{MMD}^2_b(\mathcal{F},S,T)$ (1) w.r.t. reference dataset T:[2]

$$\begin{aligned} v(S) &:= \left\langle t^{-2}\mathbf{1}_{[x,x'\in T]}, \mathbf{K}\right\rangle - \text{MMD}^2_b(\mathcal{F},S,T) \\ &= \left\langle 2(st)^{-1}\mathbf{1}_{[x\in S,x'\in T]} - s^{-2}\mathbf{1}_{[x,x'\in S]}, \mathbf{K}\right\rangle \end{aligned} \quad (2)$$

which is a reasonable choice for our problem setting under the following practical assumptions:

(A) Every party benefits from having data drawn from $\mathcal{D}$ besides having just its dataset D_i since D_i may only be sampled from a restricted subset of the support of $\mathcal{D}$ (Sec. 2). We discuss its validity in (Tay et al. 2021).

(B) The empirical distribution associated with the reference dataset T (i.e., the pooled dataset and synthetic dataset) approximates the true data distribution $\mathcal{D}$ well. This principle of approximating the ground truth with an aggregate has precedence in multi-party ML (Blanchard et al. 2017).

(C) Having more data is at least never worse off, which is generally true for ML problems (precluding cases such as excessively noisy data or adversarial data) and investigated in computational learning theory in the form of sample complexity (Bousquet, Boucheron, and Lugosi 2003).

We will now show that under such practical assumptions, $v(S)$ (2) w.r.t. reference dataset T is a reasonable choice for data valuation:

[2] A similar form to (2) is considered in another work with a different focus on interpretable ML (Kim, Khanna, and Koyejo 2016).

Proposition 1. *Let k^* be the value of every diagonal component of $\mathbf{K}$ s.t. $k^* := k(x,x) \geq k(x,x')$ for all $x,x' \in T$, and $\sigma_S := \left\langle s^{-2}\mathbf{1}_{[x,x'\in S]}, \mathbf{K}\right\rangle$. Then, $v(S)$ (2) can be re-expressed as*

$$v(S) = (s-1)^{-1}(\sigma_S - k^*) - \text{MMD}^2_u(\mathcal{F},S,T) + c \quad (3)$$

where c is a constant (i.e., independent of S).

Since σ_S is an average of kernel components $k(x,x')$ for all $x,x' \in S$, $\sigma_S \leq k^*$. It follows that the value $v(S)$ (3) of dataset S appears to weakly increase as s increases (hence satisfying assumption C) and/or as $\text{MMD}^2_u(\mathcal{F},S,T)$ decreases (thus satisfying assumptions A and B, since $\text{MMD}^2_u(\mathcal{F},S,T)$ is an unbiased estimate of the squared MMD between the distributions associated with S and T). But, this interpretation is not entirely correct as the value of σ_S may fluctuate with an increasing s, which depends on what data points are added to S. The result below gives a more precise interpretation if the value of every off-diagonal component of $\mathbf{K}$ can be bounded:

Corollary 1. *Suppose that there exist some constants γ and η s.t. $\gamma \leq k(x,x') \leq \eta \leq k^*$ for all $x,x' \in T$ and $x \neq x'$.*

$$\begin{aligned} s^{-1}(\gamma - k^*) - \text{MMD}^2_u(\mathcal{F},S,T) + c \leq v(S) \\ \leq s^{-1}(\eta - k^*) - \text{MMD}^2_u(\mathcal{F},S,T) + c\,. \end{aligned} \quad (4)$$

Since $\gamma \leq \eta \leq k^*$, as s increases and/or $\text{MMD}^2_u(\mathcal{F},S,T)$ decreases, the upper and lower bounds of $v(S)$ in (4) both weakly increase. So, given that the above practical assumptions hold, $v(S)$ is a reasonable choice for data valuation as it accounts for both the dataset quantity s and quality in terms of closeness to the empirical distribution associated with reference dataset T via $\text{MMD}^2_u(\mathcal{F},S,T)$. Also, $v(S)$ is downstream *task-agnostic* (i.e., no assumption on how each party uses its synthetic data reward) and *model-agnostic* (i.e., no restriction to the type of ML model adopted by each party) which are desirable properties as they afford flexibility to the parties. We will discuss in Sec. 5 how γ and η can be set to guarantee a non-negative and monotone $v(S)$.

Finally, our *characteristic function* for data valuation is defined as $v_c(C) := v(\bigcup_{i\in C} D_i)$ which will be used to determine the expected marginal contributions of parties $1,\ldots,n$ to CGM via the Shapley value and in turn their reward values $(r_1,\ldots,r_n)$ (Sec. 2), as detailed next.

4 Reward Scheme for Guaranteeing Incentives in CGM Framework

To incentivize collaboration among all parties in the grand coalition, their assigned rewards have to satisfy certain incentive conditions established in cooperative game theory. However, classical cooperative game theory cannot be directly applied to our problem setting involving freely replicable synthetic data reward[1]. Inspired by the reward scheme of Sim et al. (2020) for Bayesian supervised learning that is designed to guarantee certain incentives under freely replicable model reward[1], we will propose here a novel reward scheme that meets *appropriately modified* incentive conditions to suit our CGM framework.

We begin by considering the *Shapley value* ϕ_i of party i, which quantifies its expected *marginal contribution* when it joins the other parties preceding it in any permutation:

$$\phi_i := (1/n!) \sum_{\pi \in \Pi_N} [v_c(C_{\pi,i} \cup \{i\}) - v_c(C_{\pi,i})] \quad (5)$$

where the characteristic function v_c for data valuation is previously defined in Sec. 3, Π_N is the set of all possible permutations of N, and $C_{\pi,i}$ is the coalition of parties preceding i in permutation π (Chalkiadakis, Elkind, and Wooldridge 2011). The notion of marginal contribution (and hence Shapley value) plays a significant role in the properties of (**F3**) strict desirability and (**F4**) strict monotonicity that define the (**R5**) fairness incentive in (Sim et al. 2020):[3] In our work, the implication of F3 is that if the marginal contributions of parties i and j only differ for coalition C (i.e., $v_c(C \cup \{i\}) > v_c(C \cup \{j\})$), then it is only fair for party i to be assigned a larger reward value r_i; its effect on our modified F4 will be discussed later in Sec. 4.1.

Besides R5, the reward scheme of Sim et al. (2020) has considered other desirable incentives when forming the grand coalition N: (**R1**) Non-negativity: $\forall i \in N \;\; r_i \geq 0$; (**R2**) Feasibility: $\forall i \in N \;\; r_i \leq v_c(N)$; (**R3**) Weak efficiency: $\exists i \in N \;\; r_i = v_c(N)$; (**R4**) Individual rationality: $\forall i \in N \;\; r_i \geq v_c(\{i\})$; (**R6**) Stability: $\forall C \subseteq N \;\; \forall i \in C \;\; (\phi_i = \max_{j \in C} \phi_j) \Rightarrow \;\; v_c(C) \leq r_i$; and (**R7**) Group welfare involves maximizing $\sum_{i \in N} r_i$. Intuitively, R4 says that the reward value assigned to each party i should be at least the value of its dataset D_i, which makes it prefer collaboration in N than working alone. R6 states that the grand coalition is stable if for every coalition $C \subseteq N$, the reward value assigned to the party with largest Shapley value is at least the value of datasets $\bigcup_{i \in C} D_i$, which prevents all parties in coalition C from simultaneously breaking away and obtaining larger reward values. We will describe the intuition underlying our modified R2 and R3 in Sec. 4.1.

Given that v_c is non-negative and monotonically increasing (see Sec. 5 for sufficient conditions that guarantee these properties), the reward scheme of Sim et al. (2020) exploits the notion of *ρ-Shapley fair reward values* $r_i := (\phi_i/\phi^*)^\rho \times v_c(N)$ for each party $i \in N$ with an adjustable parameter ρ to trade off between satisfying the above incentive conditions. For your convenience, we've reproduced their main result and full definitions of incentive conditions in (Tay et al. 2021) and consolidated our discussion of the key differences with the work of Sim et al. (2020) in (Tay et al. 2021).

4.1 A Modified Reward Scheme with Rectified ρ-Shapley Fair Reward Values

Under the CGM framework, each party i initially has dataset D_i (Sec. 2) and would thus be assigned at least a reward value of $r_i = v_c(\{i\}) = v(D_i)$, i.e., when $G_i = \emptyset$. This is a subtle yet important difference with the reward scheme of Sim et al. (2020), the latter of which allows a party to be assigned a reward value of 0. So, we introduce a *rectified* form of the above ρ-Shapley fair reward values:

$$r_i := \max \{v_c(\{i\}), (\phi_i/\phi^*)^\rho \times v^*\} \quad (6)$$

for each party $i \in N$ where v^* is the maximum reward value (i.e., $v^* \geq r_i$ for any party $i \in N$), as discussed below (notice from Theorem 1 that $v^* = v_c(N)$ in (Sim et al. 2020)). When the grand coalition N forms, R4 is trivially satisfied since each party i has at least its dataset D_i, hence distinguishing our modified reward scheme from that of Sim et al. (2020) whose R4 may be violated. So, for our reward scheme, no party will be worse off by participating in the collaboration. However, other non-trivial issues ensue:

Proposition 2. *If $v^* = v_c(N)$ and ρ satisfies $(\phi_i/\phi^*)^\rho \times v^* < v_c(\{i\})$ for some party $i \in N$, then $(r_1, \ldots, r_n)$ (6) may not satisfy R5 due to possibly violating F3.*

Furthermore, recall from Sec. 2 that under the CGM framework, the mediator generates a synthetic dataset G from which subsets of synthetic data points are sampled to distribute to the parties as rewards. This leads to a few important implications. Firstly, since every party can at most be rewarded the entire synthetic dataset G, the largest possible reward value $v(D_i \cup G)$ may differ across parties $i = 1, \ldots, n$. In contrast, for the reward scheme of Sim et al. (2020), the largest possible reward value $v_c(N)$ is the same across all parties. Note that in our work, $v(D_i \cup G) > v_c(N)$ is possible. All these motivate the need to consider a generalized notion of the maximum reward value v^* (i.e., $v^* \geq r_i$ for any party $i \in N$) in our modified reward scheme; we will discuss below how v^* can be optimized via a linear program. As a result, R2 and R3 have to be redefined to reflect the possibility of $v(D_i \cup G) > v_c(N)$ and ensure at least one party being assigned the maximum reward value v^* instead of the possibly smaller $v_c(N)$, respectively:

Definition 1 (**R2: CGM Feasibility).** No party in the grand coalition should be assigned a reward value larger than that of its dataset and the synthetic dataset combined:

$$\forall i \in N \;\; r_i \leq v(D_i \cup G) \; .$$

Definition 2 (**R3: CGM Weak Efficiency).** At least a party in the grand coalition should be assigned the maximum reward value: $\exists i \in N \;\; r_i = v^* \; .$

We need to redefine property F4 defining R5 to account for the notion of maximum reward value v^*:

Definition 3 (**F4: CGM Strict Monotonicity).** Let v_c and v_c' denote any two characteristic functions for data valuation with the same domain 2^N, r_i and r_i' be the corresponding reward values assigned to party i, and v'^* be the maximum reward value under v_c'. If the marginal contribution of party i is larger under v_c' than v_c (e.g., by including a larger dataset) for at least a coalition, *ceteris paribus*, then party i should be assigned a larger reward value under v_c' than v_c:

$$\forall i \in N \;\; [\exists C \subseteq N \setminus \{i\} \;\; v_c'(C \cup \{i\}) > v_c(C \cup \{i\})]$$
$$\wedge \;\; [\forall B \subseteq N \setminus \{i\} \;\; v_c'(B \cup \{i\}) \geq v_c(B \cup \{i\})]$$
$$\wedge [\forall A \subseteq N \setminus \{i\} \;\; v_c'(A) = v_c(A)] \wedge (v'^* > r_i) \Rightarrow r_i' > r_i \; .$$

[3] The other two properties: (**F1**) uselessness and (**F2**) symmetry defining R5 in (Sim et al. 2020) are standard axioms of Shapley value (Shapley 1953) and commonly used in works on data valuation (Ghorbani and Zou 2019; Jia et al. 2020; Ohrimenko, Tople, and Tschiatschek 2019). Due to lack of space, we have reproduced the formal definitions of properties F1 to F4 in (Tay et al. 2021).

The following result verifies that the rectified ρ-Shapley fair reward values (6) in our modified reward scheme satisfy the above redefined incentive conditions R2, R3, R5 and previously defined ones by selecting appropriate ρ and v^*:

Proposition 3. *Let $0 \leq \rho \leq 1$. Using the new definitions of R2, R3, and F4 in Definitions 1, 2, and 3, the rectified ρ-Shapley fair reward values $(r_1, \ldots, r_n)$ (6) satisfy*

(a) R1 to R4 if ρ and v^* are set to satisfy
$\forall i \in N \;\; (v_c(\{i\}) \leq v^*) \wedge ((\phi_i/\phi^*)^\rho \times v^* \leq v(D_i \cup G))$,

(b) R1 to R5 if $\rho > 0$ and v^* are set to satisfy
$\forall i \in N \;\; v_c(\{i\}) \leq (\phi_i/\phi^*)^\rho \times v^* \leq v(D_i \cup G)$, *and*

(c) R1 to R6 if $\rho > 0$ and v^* are set to satisfy
$\forall i \in N \;\; v_c(C_i) \leq (\phi_i/\phi^*)^\rho \times v^* \leq v(D_i \cup G)$.

On the other hand, R7 (i.e., group welfare) may not be achieved since $\sum_{i \in N} r_i$ is maximized by $r_i = v(D_i \cup G)$ for each party $i \in N$ which may not be satisfied by any pair of feasible values of ρ and v^* given some synthetic dataset $G \neq \emptyset$. We will instead do our best to increase $\sum_{i \in N} r_i$ while giving precedence to satisfying the other incentive conditions in Proposition 3, as detailed next.

Optimizing ρ and v^* via a Linear Program. After computing the Shapley value ϕ_i of each party i (5), we have to optimize the values of ρ and v^* before assigning the resulting rectified ρ-Shapley fair reward values $(r_1, \ldots, r_n)$ (6) to parties $1, \ldots, n$. Let $\alpha_i := \phi_i/\phi^*$ denote the normalized Shapley value of party i, $v_i^{\min} := v_c(\{i\})$, and $v_i^{\max} := v(D_i \cup G)$. We desire v^* to be as large as possible to increase $\sum_{i \in N} r_i$ (group welfare). Also, if we like $(r_1, \ldots, r_n)$ (6) to be closer in proportion to $(\alpha_1, \ldots, \alpha_n)$ (i.e., expected marginal contributions of parties $1, \ldots, n$) or purely Shapley fair (i.e., $\rho = 1$), then ρ should be as close to 1 as possible.[4] Together with Proposition 3b, it follows that the optimization problem can be framed as $\max_{v^*, \rho}(\log v^* + \epsilon\rho)$ subject to the constraints of $\forall i \in N \;\; v_i^{\min} \leq v^* \alpha_i^\rho \leq v_i^{\max}$ and $0 \leq \rho \leq 1$ where ϵ is a weight controlling the relative importance of ρ.[5] To additionally satisfy R6 (i.e., Proposition 3c), we can set $v_i^{\min} := v_c(C_i)$ instead. Such a problem can be formulated as a *linear program* (LP) in inequality form that can be solved using standard LP solvers: $\min_{\mathbf{x}} \mathbf{c}^\top \mathbf{x}$ subject to the constraint of $\mathbf{A}\mathbf{x} \preceq \mathbf{b}$ where $\mathbf{x} := (\log v^*, \rho)^\top$, $\mathbf{c} := (-1, -\epsilon)^\top$, $\mathbf{b} := (\log v_1^{\max}, \ldots, \log v_n^{\max}, -\log v_1^{\min}, \ldots, -\log v_n^{\min}, 1, 0)^\top$, and $\mathbf{A}$ is a matrix of size $2n + 2$ by 2 with the first column $(1, \ldots, 1, -1, \ldots, -1, 0, 0)^\top$ and the second column $(\log \alpha_1, \ldots, \log \alpha_n, -\log \alpha_1, \ldots, -\log \alpha_n, 1, -1)^\top$. This formulation also informs us of a suitable choice of the synthetic dataset G: A sufficient but not necessary condition for the feasible set of the LP to be non-empty is $\min_{i \in N} v_i^{\max} \geq \max_{i \in N} v_i^{\min}$. When generating the synthetic dataset G, we may thus increase the size of G until this condition is satisfied; we provide an intuition for why this works in (Tay et al. 2021).

[4] Alternatively, one may consider decreasing ρ to increase $\sum_{i \in N} r_i$ (i.e., group welfare).

[5] We consider $\log v^*$ instead of v^* and the constraint of $\rho > 0$ in Proposition 3b is relaxed to $\rho \geq 0$ in our optimization problem to facilitate its reformulation as a linear program. In our experiments, we have never observed $\rho = 0$ since this term in the objective function is to be maximized.

4.2 Distributing Synthetic Data Rewards to Parties via Weighted Sampling

After assigning the rectified ρ-Shapley reward value r_i to each party $i \in N$ (Sec. 4.1), we greedily sample synthetic data points from G to be distributed to each party i as reward until the resulting $v(D_i \cup G_i)$ reaches the reward value r_i (Sec. 2).[6] Specifically, let $\Delta_x := v(D_i \cup G_i \cup \{x\}) - v(D_i \cup G_i)$ denote the marginal increase in the value $v(D_i \cup G_i)$ of its dataset D_i combined with its current synthetic dataset G_i by sampling the synthetic data point x. In each iteration of our weighted sampling algorithm for distributing synthetic data reward to party i (Algo. 1 in (Tay et al. 2021)), we firstly perform min-max normalization to rescale Δ_x to $\bar{\Delta}_x$ for all synthetic data points $x \in G \setminus G_i$ to lie within the $[0, 1]$ interval. We compute the probability of each synthetic data point x being sampled using the softmax function: $p(x) = \exp(\beta\bar{\Delta}_x)/\sum_{x' \in G \setminus G_i} \exp(\beta\bar{\Delta}_{x'})$ where $\beta \in [0, \infty)$ is the inverse temperature hyperparameter. Finally, we sample x based on $p(x)$ and add it to G_i. We repeat this process until $v(D_i \cup G_i)$ reaches r_i.

As $\beta \to \infty$, the synthetic data points x sampled by our algorithm tend to have larger Δ_x. This leads to fewer sampled synthetic points G_i as reward and thus a smaller $|D_i \cup G_i|$ when the resulting $v(D_i \cup G_i)$ reaches the assigned reward value r_i and the sampling ends. This in turn results in a smaller $\text{MMD}_u^2(\mathcal{F}, D_i \cup G_i, T)$, by Proposition 1. As $\beta \to 0$, the sampled synthetic points tend to have smaller Δ_x; at $\beta = 0$, our algorithm performs random sampling since all synthetic points are weighted equally. By the same reasoning, this leads to a larger $|D_i \cup G_i|$ and thus a larger $\text{MMD}_u^2(\mathcal{F}, D_i \cup G_i, T)$. So, β implicitly controls the trade-off between the no. of sampled synthetic points G_i vs. closeness to the distribution associated with reference dataset T.

Computing v using (2) incurs $\mathcal{O}(s(s+t))$ time. Instead of naively recomputing v for every synthetic data point x, the time needed to compute Δ_x can be reduced by performing a sequential update of v. By storing the values of $\langle \mathbf{1}_{[x \in S, x' \in T]}, \mathbf{K} \rangle$ and $\langle \mathbf{1}_{[x, x' \in S]}, \mathbf{K} \rangle$ at every iteration where $S = D_i \cup G_i$ (i.e., $s = |D_i \cup G_i|$), Δ_x can be recomputed for each x in $\mathcal{O}(s + t)$ time. The weighted sampling algorithm overall incurs $\mathcal{O}(n|G|^2(s+t)d)$ time. For computational details, refer to (Tay et al. 2021).

5 Kernel Selection

Recall from Sec. 3 that our data valuation function (2) depends on the choice of kernel function k which we will discuss here. The $\log$ on $v(S)$ for different subsets $S \subseteq T$ being used in the LP (Sec. 4.1) requires $v(S)$ to be non-negative for all such subsets S. The result below gives a sufficient condition on k to guarantee the non-negativity of $v(S)$:

[6] Though $v(D_i \cup G_i)$ may slightly exceed the assigned reward value r_i when sampling terminates due to discreteness of synthetic data points, such a margin diminishes when sufficiently large $|G_i|$ and $|G|$ are considered, as observed in our experiments (Sec. 6).

Proposition 4 (**Lower bound of k for non-negative $v(S)$**). *Suppose that there exist some constants γ and η s.t. $\gamma \leq k(x, x') \leq \eta \leq k^*$ for all $x, x' \in T$ and $x \neq x'$. Then,*

$$\forall S \subseteq T \;\; [\gamma = (t-2s)(k^* + (s-1)\eta)/(2s(t-s))] \Rightarrow v(S) \geq 0 \,. \quad (7)$$

Ideally, we also want $v(S)$ to be monotonically increasing as the addition of a data point to a dataset should not decrease its value, as discussed in assumption C (Sec. 3). The work of Kim, Khanna, and Koyejo (2016) provides a sufficient condition on k for v to be a monotonic function:

Theorem 1 (**Upper bound of k for monotone $v(S)$ (Kim, Khanna, and Koyejo 2016)**). *Suppose that there exists some constant η s.t. $k(x, x') \leq \eta \leq k^*$ for all $x, x' \in T$ and $x \neq x'$. Then,*

$$\forall S \subseteq T \;\; [\eta = tk^*/((s+1)(s(t-2)+t))] \Rightarrow [\forall x \in T \setminus S \;\; v(S \cup \{x\}) \geq v(S)] \,. \quad (8)$$

We can thus set an upper bound η (8) and a lower bound γ (7) of every off-diagonal component of $\mathbf{K}$ to guarantee the monotonicity and non-negativity of $v(S)$, respectively. Unfortunately, no kernel exists to satisfy both sufficient conditions in Theorem 1 and Proposition 4 at the same time if the size of S is less than half of that of the reference dataset T:

Proposition 5. *Let γ and η be set according to* (7) *and* (8). *If $s < (t/2 - 1)$, then $\gamma > \eta$.*

We prefer to guarantee the non-negativity of $v(S)$ (over monotonicity) for implementing the LP and hence only satisfy the lower bound of k (Proposition 4). Trivially setting all components of $\mathbf{K}$ to k^* satisfies this lower bound but is not useful as it values all datasets S of the same size s to be the same. Also, when the off-diagonal components of $\mathbf{K}$ are large, a non-monotonic behavior of $v(S)$ has been empirically observed, which agrees with our intuition formalized in Theorem 1 that a monotone $v(S)$ is guaranteed by an upper bound η (8) of every off-diagonal component of $\mathbf{K}$. To strike a middle ground, we use a simple binary search algorithm to find the min. length-scale of a kernel s.t. $v(D_1), \ldots, v(D_n)$ are non-negative. We have observed in our experiments that this results in an approximately monotone v and roughly 76% of all synthetic data points added causing an increase in v. We have also empirically observed that the synthetic data points are more likely to result in a decrease in v as more data points are added and s increases, which aligns with our intuition given by Theorem 1 that the upper bound η (8) to guarantee a monotone $v(S)$ decreases with a growing s and thus becomes harder to satisfy.

6 Experiments and Discussion

This section empirically evaluates the performance of our CGM framework using simulated and real-world datasets:

(a) **Simulated credit ratings.** We simulate a scenario where banks collaborate and share customer's *credit ratings* (CR) indirectly to improve their predictions on the likelihood of default (Tsai and Chen 2010). The banks serve different regions and hence own different subsets of the overall data distribution, but would like to predict well on the entire population for future expansion. Credit ratings are simulated using a 2-D Gaussian mixture model dataset with 5 clusters (classes) where the first dimension is the credit score and the second dimension is a measure of the likelihood of default.

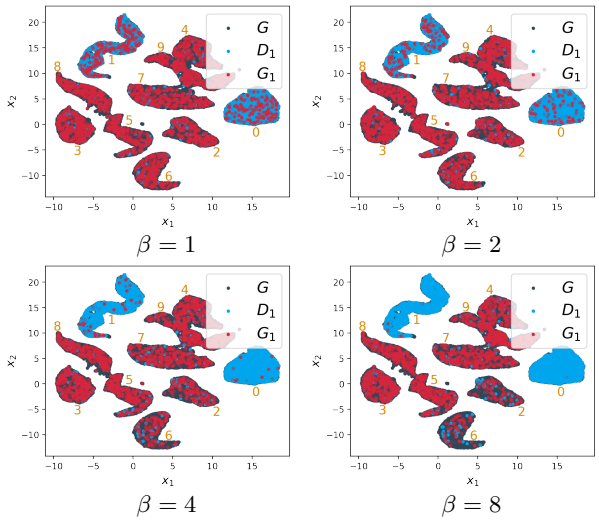
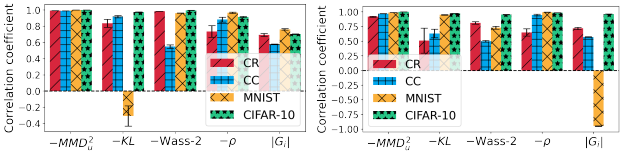


Figure 1: Synthetic data points G_1 (visualized in 2-D embedding using UMAP (McInnes, Healy, and Melville 2018)) as reward to party 1 with varying β in equal disjoint split. Each cluster has majority of the MNIST digit in yellow.

(b) **Credit card fraud dataset.** We use the real-world *credit card* (CC) fraud dataset (Dal Pozzolo et al. 2015) containing European credit card transactions such that most variables are transformed using PCA to yield 28 principal components as features and an 'Amount' variable denoting the amount transacted. We select the first 4 principal components to create a 4-D dataset, and separate the dataset into 5 classes according to Amount percentiles so as to simulate collaborating banks serving different populations that tend to make transactions within certain ranges of amounts. Synthetic data are obtained by sampling from a distribution fit to the CC dataset with kernel density estimation.

(c) **Simulated medical imaging.** Synthetic image data is commonly used to improve performance on downstream ML tasks such as in medical imaging (Bowles et al. 2018; Frid-Adar et al. 2018; Sandfort et al. 2019). We simulate a scenario where hospitals serving different populations share patients' data indirectly to improve predictions on medical imaging classification tasks on the whole population using the real-world MNIST (LeCun et al. 1998) and CIFAR-10 (Krizhevsky 2009) image datasets as surrogates. Synthetic data are obtained by sampling from pre-trained MMD GANs (Bińkowski et al. 2018). We perform dimensionality reduction on the surrogate MNIST and CIFAR-10 image datasets to create 8-D datasets, detailed in (Tay et al. 2021).

CR and CC have 5 classes, while MNIST and CIFAR-10 have 10 classes. For all datasets, we simulate 5 parties, and split the data among the 5 parties in 2 ways to simulate different settings of data sharing. The first split, which we refer to as 'equal disjoint', is when each party has a large majority of data in 1 class for CR and CC (2 for MNIST and



(a) Equal disjoint split (b) Unequal split

Figure 2: Correlation of (negative of) metrics and $|G_i|$ with α_i (higher is better).

CIFAR-10) and a small quantity of data in the other classes, and these majority classes are non-overlapping to simulate real-world settings where every party contributes data from a different restricted subset of the support of the data distribution. The second split, which we refer to as 'unequal', is when the first 2 parties have a uniform distribution of data over all classes while the remaining 3 parties have a large majority of data in 3 classes (6 for MNIST and CIFAR-10) and a small quantity of data in the rest of the classes to simulate real-world settings where some parties have 'higher-quality' data than the other parties in terms of the coverage of the support of the data distribution. However, our CGM framework is *not given these class labels* to simulate real-world scenarios where the class differences among parties are unknown. We use the squared exponential kernel with its length-scale computed using the binary search algorithm in Sec. 5. Our full CGM framework, which includes computing the normalized Shapley values $\alpha_1, \ldots, \alpha_n$ (i.e., expected marginal contributions) of parties $1, \ldots, n$, solving the LP to obtain their assigned rectified ρ-Shapley fair reward values $(r_1, \ldots, r_n)$, and running the weighted sampling algorithm for generating synthetic data points $G_1, \ldots, G_n$ to be distributed to them as rewards (Sec. 4), is applied across all datasets and splits. (Tay et al. 2021) provides full details of the experimental settings, additional results, and visualizations of the synthetic data rewards. As none of the prior work has previously considered synthetic data rewards, our results below set the baseline for future work.

Assessing contributions of parties. We assess whether our CGM framework can appropriately quantify the expected marginal contributions of the parties via their Shapley values (Sec. 4). Results are reported in (Tay et al. 2021): As expected, very large α_i's are observed for parties in the unequal split with full class distribution, while the α_i's are typically more evenly spread in the equal disjoint split.

Role of inverse temperature hyperparameter β. To substantiate our claim that β in the weighted sampling algorithm (Sec. 4.2) controls the trade-off between the no. of synthetic data points as rewards vs. negative unbiased MMD (i.e., closeness to empirical distribution associated with T), we report the correlation of β with them in (Tay et al. 2021): β is observed to be highly negatively correlated with the no. of synthetic data points and highly positively correlated with negative unbiased MMD, which aligns with our reasoning in Sec. 4.2. Also, Fig. 1 shows that as β increases, the algorithm samples fewer synthetic data points but they are more dissimilar from a party's original dataset.

Are synthetic data rewards distributed to parties and their downstream ML task performances commensurate to their contributions? We firstly assess whether our CGM framework can distribute synthetic data points G_i to each party i as reward such that the closeness of the empirical distributions associated with $D_i \cup G_i$ vs. reference dataset T correlates well with its expected marginal contribution via the normalized Shapley value α_i. We quantify such a closeness using 4 metrics (which we take the negative of so that higher is better): (a) unbiased MMD estimate (1), (b) an estimate of reverse Kullback-Leibler divergence based on k-nearest neighbors (Pérez-Cruz 2008) averaged over $k = 2, \ldots, 6$, (c) Wasserstein-2 distance between multivariate Gaussians fit to $D_i \cup G_i$ vs. T (i.e., how Fréchet Inception distance for evaluating GANs is computed (Heusel et al. 2017)), and (d) class imbalance ρ calculated with $\rho_i := (1/m) \sum_{y=1}^{m} p_y^2$ where m is the no. of classes and p_y is the proportion of data points in party i's combined dataset $D_i \cup G_i$ belonging to class y. In all datasets, T is equally distributed among the classes and hence achieves a minimum for ρ. We also measure the correlation of the no. $|G_i|$ of synthetic data points as reward to party i with α_i. Fig. 2 shows results of the mean and standard error of the correlations over varying $\beta = 1, 2, 4, 8$ in the weighted sampling. It can be observed that across all splits, datasets, and metrics, the negative of the metrics and $|G_i|$ mostly display highly positive correlations with α_i, as desired. We defer the discussion of the few negative correlations to (Tay et al. 2021).

After distributing the synthetic data rewards to the parties, we assess whether their performances on downstream ML tasks (from augmenting their real data with synthetic data) correlate well with their expected marginal contributions via α_i. We simulate supervised learning scenarios where each party trains an SVM on its real and synthetic data and predicts the class labels on unseen real data. For the real-world CC, MNIST, and CIFAR-10 datasets, the correlations of their classification accuracies with α_i (averaged over β) are, respectively, 0.523, 0.459, and 0.174 in the equal disjoint split, and 0.791, 0.338, and 0.835 in the unequal split. We observe positive correlations overall, thus confirming our hypothesis that the parties' downstream ML task performances are commensurate to their contributions.

7 Conclusion

This paper has described a novel CGM framework that incentivizes collaboration among self-interested parties to contribute data to a pool for training a generative model, from which synthetic data are drawn and distributed to the parties as rewards commensurate to their contributions. The CGM framework comprises an MMD-based data valuation function whose bounds weakly increase with a growing dataset quantity and an improved closeness of the empirical distributions associated with the dataset vs. the reference dataset, a reward scheme formulated as an LP for guaranteeing incentives like fairness, and a weighted sampling algorithm with the flexibility of controlling the trade-off between no. of synthetic data points as reward vs. the closeness described above. For future work, we will consider deep kernels to automatically learn useful representations for data valuation and prove stronger guarantees on the non-negativity and monotonicity of our data valuation function.

Acknowledgments. This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (Award No: AISG2-RP-2020-018). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. Sebastian Tay is supported by the Institute for Infocomm Research of Agency for Science, Technology and Research (A*STAR).

References

Bińkowski, M.; Sutherland, D. J.; Arbel, M.; and Gretton, A. 2018. Demystifying MMD GANs. In *Proc. ICLR*.

Blanchard, P.; El Mhamdi, E. M.; Guerraoui, R.; and Stainer, J. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proc. NeurIPS*, 118–128.

Borji, A. 2019. Pros and cons of GAN evaluation measures. *Computer Vision and Image Understanding*, 179: 41–65.

Bousquet, O.; Boucheron, S.; and Lugosi, G. 2003. Introduction to statistical learning theory. In Bousquet, O.; von Luxburg, U.; and Rätsch, G., eds., *Advanced Lectures on Machine Learning*, volume 3176 of *Lecture Notes in Computer Science*, 169–207. Springer, Berlin, Heidelberg.

Bowles, C.; Chen, L.; Guerrero, R.; Bentley, P.; Gunn, R.; Hammers, A.; Dickie, D. A.; Hernández, M. V.; Wardlaw, J.; and Rueckert, D. 2018. GAN augmentation: Augmenting training data using generative adversarial networks. arXiv:1810.10863.

Chalkiadakis, G.; Elkind, E.; and Wooldridge, M. 2011. Computational Aspects of Cooperative Game Theory. In Brachman, R. J.; Cohen, W. W.; and Dietterich, T. G., eds., *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool Publishers.

Chen, M.; Liu, Y.; Shen, W.; Shen, Y.; Tang, P.; and Yang, Q. 2020. Mechanism Design for Multi-Party Machine Learning. arXiv:2001.08996.

Cong, M.; Yu, H.; Weng, X.; and Yiu, S. 2020. A Game-Theoretic Framework for Incentive Mechanism Design in Federated Learning. In Yang, Q.; Fan, L.; and Yu, H., eds., *Federated Learning*, volume 12500 of *Lecture Notes in Computer Science*, 205–222. Springer, Cham.

Dal Pozzolo, A.; Caelen, O.; Johnson, R. A.; and Bontempi, G. 2015. Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence*, 159–166. IEEE.

Devereaux, P. J.; Guyatt, G.; Gerstein, H.; Connolly, S.; and Yusuf, S. 2016. Toward Fairness in Data Sharing. *New England Journal of Medicine*, 375(5): 405–407.

Ding, J.; Liang, G.; Bi, J.; and Pan, M. 2021. Differentially Private and Communication Efficient Collaborative Learning. In *Proc. AAAI*.

Frid-Adar, M.; Diamant, I.; Klang, E.; Amitai, M.; Goldberger, J.; and Greenspan, H. 2018. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, 321: 321–331.

Fukumizu, K.; Gretton, A.; Sun, X.; and Schölkopf, B. 2007. Kernel measures of conditional dependence. In *Proc. NeurIPS*, volume 20, 489–496.

Fukumizu, K.; Sriperumbudur, B. K.; Gretton, A.; and Schölkopf, B. 2008. Characteristic Kernels on Groups and Semigroups. In *Proc. NeurIPS*, 473–480.

Ghorbani, A.; Kim, M. P.; and Zou, J. 2020. A Distributional Framework for Data Valuation. In *Proc. ICML*.

Ghorbani, A.; and Zou, J. 2019. Data Shapley: Equitable valuation of data for machine learning. In *Proc. ICML*, 4053–4065.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A kernel two-sample test. *Journal of Machine Learning Research*, 13(1): 723–773.

Hayes, J.; and Ohrimenko, O. 2018. Contamination Attacks and Mitigation in Multi-Party Machine Learning. In *Proc. NeurIPS*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Proc. NeurIPS*, 6626–6637.

Hu, Y.; Niu, D.; Yang, J.; and Zhou, S. 2019. FDML: A Collaborative Machine Learning Framework for Distributed Features. In *Proc. ACM SIGKDD*, Pages 2232–2240.

Jia, R.; Dao, D.; Wang, B.; Hubis, F. A.; Hynes, N.; Gurel, N. M.; Li, B.; Zhang, C.; Song, D.; and Spanos, C. 2020. Towards efficient data valuation based on the Shapley value. In *Proc. AISTATS*.

Kang, J.; Xiong, Z.; Niyato, D.; Xie, S.; and Zhang, J. 2019a. Incentive Mechanism for Reliable Federated Learning: A Joint Optimization Approach to Combining Reputation and Contract Theory. *IEEE Internet of Things Journal*, 6(6): 10700–10714.

Kang, J.; Xiong, Z.; Niyato, D.; Yu, H.; Liang, Y.-C.; and Kim, D. I. 2019b. Incentive Design for Efficient Federated Learning in Mobile Networks: A Contract Theory Approach. In *Proc. IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, 1–5.

Kim, B.; Khanna, R.; and Koyejo, O. O. 2016. Examples are not enough, learn to criticize! Criticism for interpretability. In *Proc. NeurIPS*, 2280–2288.

Krawczyk, B. 2016. Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence*, 5(4): 221–232.

Krizhevsky, A. 2009. *Learning multiple layers of features from tiny images*. Master's thesis, Department of Computer Science, University of Toronto.

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

Lo, B.; and DeMets, D. L. 2016. Incentives for Clinical Trialists to Share Data. *New England Journal of Medicine*, 375(12): 1112–1115.

Lyu, L.; Xu, X.; Wang, Q.; and Yu, H. 2020. Collaborative Fairness in Federated Learning. In Yang, Q.; Fan, L.; and

Yu, H., eds., *Federated Learning*, volume 12500 of *Lecture Notes in Computer Science*, 189–204. Springer, Cham.

Maleki, S.; Tran-Thanh, L.; Hines, G.; Rahwan, T.; and Rogers, A. 2013. Bounding the Estimation Error of Sampling-based Shapley Value Approximation. *CoRR*, abs/1306.4265.

McInnes, L.; Healy, J.; and Melville, J. 2018. UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv:1802.03426.

Micchelli, C. A.; Xu, Y.; and Zhang, H. 2006. Universal Kernels. *Journal of Machine Learning Research*, 7(12).

Ohrimenko, O.; Tople, S.; and Tschiatschek, S. 2019. Collaborative machine learning markets with data-replication-robust payments. arXiv:1911.09052.

Pérez-Cruz, F. 2008. Kullback-Leibler divergence estimation of continuous distributions. In *Proc. IEEE ISIT*, 1666–1670.

Richardson, A.; Filos-Ratsikas, A.; and Faltings, B. 2020a. Budget-Bounded Incentives for Federated Learning. In Yang, Q.; Fan, L.; and Yu, H., eds., *Federated Learning*, volume 12500 of *Lecture Notes in Computer Science*, 176–188. Springer, Cham.

Richardson, A.; Filos-Ratsikas, A.; and Faltings, B. 2020b. Incentivizing and Rewarding High-Quality Data via Influence Functions. In *Proc. ICML Workshop on Incentives in Machine Learning*.

Sandfort, V.; Yan, K.; Pickhardt, P. J.; and Summers, R. M. 2019. Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. *Scientific Reports*, 9.

Seitzer, M. 2020. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid. Version 0.1.1.

Shapley, L. S. 1953. A value for n-person games. In Kuhn, H. W.; and Tucker, A. W., eds., *Contributions to the Theory of Games*, volume 2, 307–317. Princeton Univ. Press.

Sim, R. H. L.; Zhang, Y.; Chan, M. C.; and Low, B. K. H. 2020. Collaborative Machine Learning with Incentive-Aware Model Rewards. In *Proc. ICML*, 8927–8936.

Sim, R. H. L.; Zhang, Y.; Low, B. K. H.; and Jaillet, P. 2021. Collaborative Bayesian Optimization with Fair Regret. In *Proc. ICML*, 9691–9701.

So, J.; Guler, B.; and Avestimehr, S. 2020. A Scalable Approach for Privacy-Preserving Collaborative Machine Learning. In *Proc. NeurIPS*, 8054–8066.

Sriperumbudur, B. K.; Gretton, A.; Fukumizu, K.; Lanckriet, G.; and Schölkopf, B. 2008. Injective Hilbert space embeddings of probability measures. In *21st Annual Conference on Learning Theory (COLT 2008)*, 111–122. Omnipress.

Tay, S. S.; Xu, X.; Foo, C. S.; and Low, B. K. H. 2021. Incentivizing Collaboration in Machine Learning via Synthetic Data Rewards. *arXiv preprint*.

Tsai, C.-F.; and Chen, M.-L. 2010. Credit rating by hybrid machine learning techniques. *Applied Soft Computing*, 10(2): 374–380.

Vaidya, P. M. 1989. Speeding-up linear programming using fast matrix multiplication. In *30th Annual Symposium on Foundations of Computer Science*, 332–337. IEEE Computer Society.

Wang, T.; Rausch, J.; Zhang, C.; Jia, R.; and Song, D. 2020. A Principled Approach to Data Valuation for Federated Learning. In Yang, Q.; Fan, L.; and Yu, H., eds., *Federated Learning*, volume 12500 of *Lecture Notes in Computer Science*, 153–167. Springer, Cham.

Xu, X.; Lyu, L.; Ma, X.; Miao, C.; Foo, C. S.; and Low, B. K. H. 2021a. Gradient Driven Rewards to Guarantee Fairness in Collaborative Machine Learning. In *Proc. NeurIPS*.

Xu, X.; Wu, Z.; Foo, C. S.; and Low, B. K. H. 2021b. Validation Free and Replication Robust Volume-based Data Valuation. In *Proc. NeurIPS*.

Yan, T.; and Procaccia, A. D. 2021. If You Like Shapley Then You'll Love the Core. In *Proc. AAAI*.

Yu, H.; Liu, Z.; Liu, Y.; Chen, T.; Cong, M.; Weng, X.; Niyato, D.; and Yang, Q. 2020. A Fairness-Aware Incentive Scheme for Federated Learning. In *Proc. AIES*.

Zhan, Y.; Li, P.; Qu, Z.; Zeng, D.; and Guo, S. 2020. A learning-based incentive mechanism for federated learning. *IEEE Internet of Things Journal*, 7(7): 6360–6368.