Fast Shapley Value Computation in Data Assemblage Tasks as Cooperative Simple Games

XUAN LUO, Simon Fraser University, Canada
JIAN PEI, Duke University, USA
CHENG XU, Hong Kong Baptist University, Hong Kong
WENJIE ZHANG, University of New South Wales, Australia
JIANLIANG XU, Hong Kong Baptist University, Hong Kong

In this paper, we tackle the challenging problem of Shapley value computation in data markets in a novel setting of data assemblage tasks with binary utility functions among data owners. By modeling these scenarios as cooperative simple games, we leverage pivotal probabilities to transform the computation into a problem of counting beneficiaries. Moreover, we make an insightful observation that the Shapley values can be computed using subsets of minimal syntheses within the inclusion-exclusion framework in combinatorics. Based on this insight, we develop a game decomposition approach and utilize techniques in Boolean function decomposition into disjunctive normal form. One interesting property of our method is that the time complexity depends only on the data owners participating in those minimal syntheses, rather than all the data owners. Extensive experiments with real data sets demonstrate a significant efficiency improvement for computing the Shapley values in data assemblage tasks modeled as simple games.

CCS Concepts: • **Information systems → Data management systems**.

Additional Key Words and Phrases: Shapley Value, Data Market, Data Assemblage, Simple Game

ACM Reference Format:
Xuan Luo, Jian Pei, Cheng Xu, Wenjie Zhang, and Jianliang Xu. 2024. Fast Shapley Value Computation in Data Assemblage Tasks as Cooperative Simple Games. *Proc. ACM Manag. Data* 2, 1 (SIGMOD), Article 56 (February 2024), 28 pages. https://doi.org/10.1145/3639311

1 INTRODUCTION

The power of big data largely stems from its many secondary uses, such as enabling machine learning and AI models [33, 38, 42], recommender systems [3, 65], causal inference [34, 60, 61], and data-driven decision-making applications [63]. However, a significant challenge lies in incentivizing and facilitating large-scale data sharing and collaboration. Data markets [15, 27, 39, 59, 67] are emerging as a promising solution to enable and facilitate data sharing among potential data owners and consumers.

Essentially, a data market serves as an online platform where parties with data demands can acquire data sets or data services, while data owners can exchange their data and services for revenue or compensation in various ways. Numerous active data markets already exist, such as AWS

Authors' addresses: Xuan Luo, Simon Fraser University, Technology and Science Complex I, Burnaby, BC, Canada, xuan_luo@sfu.ca; Jian Pei, Duke University, Duke Box 90129, Durham, NC, USA, 27708-0129, j.pei@duke.edu; Cheng Xu, Hong Kong Baptist University, Kowloon, Hong Kong, chengxu@comp.hkbu.edu.hk; Wenjie Zhang, University of New South Wales, Sydney, Australia, wenjie.zhang@unsw.edu.au; Jianliang Xu, Hong Kong Baptist University, Kowloon, Hong Kong, xujl@comp.hkbu.edu.hk.

2836-6573/2024/2-ART56 $15.00
https://doi.org/10.1145/3639311

Data Exchange, Windows Azure Marketplace, Dawex, Datarade, dmi.io, WorldQuant, Xignite, and BlueTalon [39]. Due to diverse needs in business models, vertical domains, government regulations, and industry applications, there are different types of data markets.

At the core of every data market lies a data valuation module. In a data market, a group of data owners collaborate to complete a specific task requested by a buyer, such as assembling a data set for data analytics or machine learning. In this paper, we focus on *data assemblage tasks* [45], which involve using multiple data sets from various data owners to create a desired data set. Data assemblage tasks are at the core of data integration [10, 16, 24, 40, 53, 64]. In this context, data valuation, in essence, assigns a score to a data owner, reflecting their contribution to a data assemblage task. Formally, given a task, a set of data owners and their corresponding data sets $O = \{o_1, \ldots, o_n\}$, a data valuation (function) $\psi : O \rightarrow [0, 1]$ is defined such that $\sum_{i=1}^{n} \psi(o_i) = 1$.

Data valuation plays a crucial role in ensuring fairness, effectiveness, and efficiency in data markets. Different data markets may have different requirements for data valuation [62], such as truthfulness [2], revenue maximization, fairness [2, 71], arbitrage-freeness [43], privacy-preservation [1, 4, 5, 21, 28, 30, 35, 57, 58, 72], and computational efficiency [2, 6, 32].

In this paper, we explore one type of fundamental data valuation scenarios in data assemblage tasks within data markets, where the utility of a coalition among data owners is limited to binary values of 0 or 1. In this setting, if a coalition successfully produces a data set that meets the data buyer's requirements, the utility of the coalition is assigned value 1; otherwise, value 0. This binary framework allows for transactions where the data set is either purchased in its entirety or not purchased at all and finds applications in various scenarios. For instance, in some existing data markets like DataRade, a data buyer typically has the option to either purchase a complete data set or choose not to make any purchase, rather than paying a fraction of the cost for a partial portion of the data. The main focus of this paper is to address the question of how to assess the contribution of each data owner in a fair and efficient manner.

The simple binary setting can be modeled as a simple game in cooperative game theory [13]. The Shapley fairness principle [71] is the most fundamental and widely used approach to achieving fairness in this context. The Shapley value, as the unique solution that embodies Shapley fairness, measures the expected marginal contribution of a participant over all possible coalitions of other participants. However, due to the combinatorial nature of the Shapley value, computing the exact Shapley value is #P-hard [22, 25]. Faigle and Kern [25] demonstrated that even in a simple game, the computation of the Shapley value is #P-hard in general.

To address this challenge, some existing studies have focused on approximating the Shapley value [47, 48]. These approximation methods often utilize Monte Carlo simulation and require a substantial number of samples to reduce the error to an acceptable level. As a result, these approximation methods can still be time-consuming. Other studies have made assumptions about additional properties of utility functions [29, 36, 41, 45]. For example, Luo et al. [45] assumed that the utility of a target data set can be decomposed, allowing for the decomposition of Shapley value computation. However, to the best of our knowledge, the general problem of computing the Shapley value for data assemblage tasks as a simple game has not been modeled or explored in the existing literature.

In this paper, we tackle the challenge of efficiently computing the Shapley value in data assemblage tasks as simple games and make several contributions. After the problem definition and the review of related work in Section 2, we firstly leverage the concept of pivotal probability in simple games and transform the computation of pivotal probabilities in data assemblage tasks into a problem of counting beneficiaries using minimal syntheses in data assemblage. This transformation allows us to devise an algorithm for computing the Shapley value, which has a time complexity that depends only on the data owners participating in those minimal syntheses, rather than all the

data owners. Secondly, considering that data sets are often assembled in a structured manner, such as through operations like join, concatenation, projection, and selection, we make a significant observation that the Shapley value can be computed using subsets of minimal syntheses within the inclusion-exclusion framework in combinatorics (Section 3). Drawing from this insight, we develop a game decomposition approach and utilize cutting-edge techniques in Boolean function decomposition into disjunctive normal form (Section 4). Thirdly, through extensive experiments, we provide compelling evidence that our novel approaches greatly enhance the efficiency of Shapley value computation in data assemblage tasks modeled as simple games (Sections 5 and 6). Lastly, we conclude the paper and discuss future directions (Section 7). Limited by space, we omit most proofs, which can be found in the full version technical report [46].

2 PROBLEM DESCRIPTION AND RELATED WORK

2.1 Data Assemblage Tasks as Simple Games

In cooperative game theory [13], a **characteristic function game** G is defined by a pair (O, v), where O is a set of players and $v : 2^O \rightarrow \mathbb{R}$ a **characteristic function**, which maps each **coalition** $C \subseteq O$ to a real number $v(C) \in \mathbb{R}$ called the **value** of the coalition. The whole set of players O itself as a coalition is called the **grand coalition**. A characteristic function game $G = (O, v)$ is **monotone** if for any coalitions C and C' such that $C \subseteq C' \subseteq O$, $v(C) \leq v(C')$.

A characteristic function game $G = (O, v)$ is a **simple game** if it is monotone and the characteristic function only takes values 0 and 1, that is, $v(C) \in \{0, 1\}$ for any coalition $C \subseteq O$. In other words, in a simple game, a coalition either succeeds (i.e., achieving the goal) or fails. A classic example of cooperative simple games is voting games [14]. Given a finite set of voters O as the players, each vote has the same weight, say 1. Let $q \geq 0$ be a quota. The characteristic function $v : 2^O \rightarrow \{0, 1\}$ is defined as, for any coalition $C \subseteq O$, $v(C) = 1$ if $|C| \geq q$; and 0 otherwise. In this paper, we focus on cooperative simple games. Thus, hereafter, without specific mentioning, the term "game" refers to cooperative simple game.

Consider a set of **data owners** O, each having a data set. In a **data assemblage task**, the data owners collaborate and try to produce a **target data set** D that a data buyer wants to acquire. Since the data owners use their data sets to conduct the data assemblage task, for the sake of brevity, we overload symbol $o \in O$ to denote a data owner as well as the data set that the owner has. In the rest of the paper, we use the terms interchangeably and do not distinguish data owners and the corresponding data sets.

In the context of a data assemblage task, a coalition $C \subseteq O$ is a subset of data owners. If the data sets in a coalition C can be used to produce the target data set D, then C is called a **synthesis**. In general, there may exist multiple syntheses in a data assemblage task. Moreover, adding more data owners to a synthesis does not prevent the target data set from being produced. Therefore, a superset of a synthesis is also a synthesis.

Example 1 (Data assemblage task). Suppose a data buyer wants to acquire the data about the house value and the resident names at street address a. The target data schema is $R =$ (address, resident, value). A data owner o_1 has the home value data in the schema $R_1 =$ (address, value). Data owners o_2, o_3, and o_4 have some partial information about the residents living in those homes in schema $R_2 =$ (address, resident). The target data set can be produced using the assemblage plan $(o_1 \bowtie (o_2 \cup o_3 \cup o_4))$. $\{o_1, o_2, o_3, o_4\}$ is a synthesis, since it produces the target data set.

Suppose $o_1 = \{(a, 300k)\}$, $o_2 = \{(a, John), (a, Amy)\}$, $o_3 = \{(a, John), (a, Cathy)\}$, and $o_4 = \{(a, Amy), (a, Cathy)\}$. Since $(o_1 \bowtie (o_2 \cup o_3 \cup o_4)) = (o_1 \bowtie (o_2 \cup o_3))$, $\{o_1, o_2, o_3\}$ is also a synthesis. So are $\{o_1, o_2, o_4\}$ and $\{o_1, o_3, o_4\}$. □

We can model a data assemblage task as a simple game using syntheses.

Definition 1 (Data assemblage). A **data assemblage task** is a simple game (O, v), where O is a set of data owners and the value of a coalition is defined as, for any coalition $C \subseteq O$, $v(C) = 1$ if C is a synthesis; and 0 otherwise. We also call a data owner a player. □

Simple games have been extensively researched in literature [14, 54]. The major challenge is the representation. A straightforward approach is to list every possible coalition and its corresponding value or every successful coalition, but this becomes exponential in the number of data owners, making it inefficient. To address this challenge, many studies have explored more succinct representations, such as characteristic functions. One example is the use of monotone boolean functions to represent simple games [54]. In this representation, the characteristic function $v(\cdot)$ is a boolean function, which takes a set of data owners as input and considers each data owner as either being part of the coalition (TRUE) or not (FALSE). The coalition is considered winning if the characteristic function returns TRUE and otherwise losing. A characteristic function in a data assemblage task as a simple game can be written in disjunctive normal form (DNF), where each conjunction term is a synthesis [19, 54].

2.2 Shapley Value

In a data assemblage task, how should we fairly assess the contribution made by every data owner in assembling the target data set? Due to the monotonicity of the data assemblage game, data owners are motivated to cooperate and form the grand coalition in data assemblage tasks. Therefore, the Shapley value [71] becomes a natural choice.

Definition 2 (Shapley value [71]). Given a characteristic function game (O, v), the **Shapley value** of a player $o \in O$ is

$$\psi(o) = \frac{1}{|O|} \sum_{C \subseteq O \setminus \{o\}} \frac{v(C \cup \{o\}) - v(C)}{\binom{|O|-1}{|C|}} = \frac{1}{|O|!} \sum_{\pi \in \Pi_O} (v(P_o^\pi \cup \{o\}) - v(P_o^\pi)) \tag{1}$$

where Π_O is the set of all possible permutations of the players in O and P_o^π is the set of players preceding o in permutation $\pi \in \Pi_O$. □

Shapley value is the only payoff division scheme that satisfies the **Shapley fairness** [71], which consists of the properties of efficiency, no payoff for dummy, symmetry, and additivity. Given the attractive properties of the Shapley value, we are interested in computing the Shapley value for every data owner in a data assemblage task as a simple game. However, computing the Shapley value using Equation 1 is often very costly and cannot scale up to a large set of players due to the combinatorial nature [29, 37].

2.3 Related Work

With the increasing popularity of data science, more and more data markets (such as Dawex[1], Snowflake data marketplace[2], and BDEX[3]) have been established to facilitate the exchange of data between data suppliers and data consumers [27, 67]. Seven categories of participants have been identified in these data markets [55].

A key issue in data markets is revenue allocation [17]. Some popular methods include the core [31] and the leave-one-out method [18]. However, both the core and the leave-one-out method do not fulfill all fairness properties.

Due to the combinatoric nature, the computation of the exact Shapley value can be very expensive. To tackle this challenge, one approach is to approximate the Shapley value using the Monte Carlo

[1]https://www.dawex.com/en/, accessed on May 9, 2021.
[2]https://www.snowflake.com/data-marketplace/, accessed on May 9, 2021.
[3]https://www.bdex.com, accessed on May 9, 2021.

sampling method [47]. Other studies have explored additional assumptions, such as considering the uniqueness or novelty of data items [41], assuming that adding a small number of training data points has a negligible impact on model performance [29], exploiting the locality of utility in certain models [36], and applying the independent utility assumption [45]. Our study distinguishes from the existing work by exploring the opportunity of fast Shapley value computation enabled by some interesting properties of simple games. Comparing with those studies, this paper does not require any additional assumption and focuses on the computation of the exact Shapley value in a simple game.

Shapley [70] proposed that a simple game can be decomposed into smaller sub-games and such a decomposition can be represented using a decomposition tree. Since the characteristic function of a simple game can be represented as a monotone boolean function, the decomposition tree can be obtained via the decomposition of monotone boolean function. To compute the decomposition tree, Bioch [7, 8] proposed a decomposition algorithm based on generalized Shannon decomposition [11, 12, 69] and showed that the tree can be obtained in polynomial time. However, the previous studies on game decomposition mainly focus on the representation of simple games using smaller sub-games after game decomposition and how to obtain the decomposition tree, but do not address how to use game decomposition to speed up Shapley value computation. To the best of our knowledge, we are the first to use decomposition of simple games to accelerate Shapley value computation.

Our study is remotely related to quantifying the contribution of database tuples to query answering in Shapley value [23, 44, 51, 52, 66]. The key difference is that those studies compute the exact Shapley value only for the queries where probability computation is tractable [20, 23], but our study puts no restriction on query types and can be applied to all queries in general.

3 COMPUTING SHAPLEY VALUE USING SYNTHESES

In this section, we first review the pivotal interpretation of the Shapley value [26, 49]. Then, we transform the problem of Shapley value computation to computing pivotal probabilities using syntheses.

3.1 Pivotal Probability and Shapley Value

In a data assemblage game, we can divide all coalitions into two categories: the **losing coalitions**, which cannot produce the target data set, and the **syntheses**, which are winning coalitions that produce the target data set. A data owner o is **pivotal** for a losing coalition C if $C \cup \{o\}$ is a synthesis. Using this notion, we can rewrite Equation 1 into

$$\psi(o) = \frac{1}{|O|!}|\{\pi \in \Pi_O | v(P_o^\pi) = 0, v(P_o^\pi \cup \{o\}) = 1\}|, \tag{2}$$

which gives the intuition of pivotal probability [26, 49].

Definition 3 (Pivotal probability). Given a set of data owners O and permutation $\pi \in \Pi_O$, a data owner o is **pivotal** for π and π is a **beneficiary** of o if $v(P_o^\pi) = 0$ and $v(P_o^\pi \cup \{o\}) = 1$. Denote by $\mathcal{B}_o = \{\pi \in \Pi_O | v(P_o^\pi) = 0, v(P_o^\pi \cup \{o\}) = 1\}$ the set of beneficiaries of o. The **pivotal probability** of o is the probability that o is pivotal for a permutation, that is, $\frac{|\mathcal{B}_o|}{|\Pi_O|} = \frac{|\mathcal{B}_o|}{|O|!}$. □

Proposition 4. *[49] In a simple game $G = (O, v)$, for any player $o \in O$, the Shapley value $\psi(o)$ equals the pivotal probability of o.* □

Based on Proposition 4, the problem of computing Shapley value for a data owner in a data assemblage task can be transformed into computing the pivotal probability of the data owner. According to Equation 2, to calculate the pivotal probability of a data owner o, we only need to count the number of permutations for which o is pivotal, that is, the size of $\mathcal{B}_o$ in Definition 3.

3.2 Counting Beneficiaries Using Minimal Syntheses

Any superset of a synthesis can produce the target data set and thus is also a synthesis. This superset monotonicity leads to a huge combinatorial space that contains many syntheses. To count the number of permutations for which a data owner is pivotal, we are interested in the minimal subsets of data owners that are syntheses. A synthesis C is **minimal** if for any coalition $C' \subset C$, C' is not a synthesis. We have the following result immediately.

Proposition 5. *If C is a minimal synthesis, then for every data owner $o \in C$, o is pivotal for $C \setminus \{o\}$.* □

A data owner o is a **dummy** if there does not exist any minimal synthesis C such that $o \in C$. According to Equation 2, $\psi(o) = 0$. Thus, we only need to compute the Shapley value for those data owners appearing in minimal syntheses. Since dummies do not contribute to Shapley computation, we can safely remove all dummies from our consideration. In this paper, without further mentioning we assume that *every data owner is not a dummy.*

What is the relation between beneficiary permutations and minimal syntheses? We have the following observation.

Theorem 6. *In a data assemblage game where O is the set of data owners, for a data owner $o \in O$ and a permutation $\pi \in \Pi_O$, π is a beneficiary of o if and only if $P_o^\pi \cup \{o\}$ is a synthesis and, for every minimal synthesis $C \subseteq P_o^\pi \cup \{o\}$, $o \in C$.* □

Theorem 6 points to a useful direction of computing the Shapley value of a data owner o. To calculate the size of $\mathcal{B}_o$, we only need to consider the minimal syntheses containing o and the prefixes of permutations where those minimal syntheses appear. We model such prefixes as the permutation set as follows.

For a subset of data owners $C \subseteq O$ and a permutation $\pi \in \Pi_O$, let P_C^π be the minimum prefix of π that contains all owners in C. For a data owner $o \in C$, let $P_{C\prec o} = \{\pi \mid \text{the last owner in } P_C^\pi \text{ is } o\}$ be the set of permutations where all other owners in C precede o. For a minimal synthesis C and a data owner $o \in C$, the **permutation set** of C with respect to o is $PS(C \prec o) = \{\pi \in P_{C\prec o} \mid P_C^\pi \setminus \{o\}$ does not contain any minimal synthesis$\}$. Following Theorem 6, in a data assemblage game, for any data owner o, $\mathcal{B}_o = \cup_{\text{minimal synthesis } C \text{ s.t. } o\in C} PS(C \prec o)$. Thus, we can count the beneficiaries of o using minimal syntheses. However, it is costly to compute $|\mathcal{B}_o|$ using this formula directly. Since $PS(C \prec o)$ is a subset of $P_{C\prec o}$, we can count $|PS(C \prec o)|$ through considering the permutations in $P_{C\prec o}$.

Example 2 (Permutation Set). Suppose a set of data owners $O = \{o_1, o_2, o_3\}$ can produce a target data set in coalition and the minimal syntheses are $C_1 = \{o_1, o_2\}$, $C_2 = \{o_1, o_3\}$, and $C_3 = \{o_2, o_3\}$. Then, for data owner o_1, $\mathcal{B}_{o_1} = PS(C_1 \prec o_1) \cup PS(C_2 \prec o_1)$.

How can we obtain $PS(C_1 \prec o_1)$ from $P_{C_1\prec o_1} = \{o_3o_2o_1, o_2o_3o_1, o_2o_1o_3\}$? According to the definitions, we need to filter out those permutations in $P_{C_1\prec o_1}$ whose prefixes contain a minimal synthesis before o_1. Apparently, a minimal synthesis can appear before o_1 only if it does not contain o_1. In this example, C_3 is the only minimal synthesis that does not contain o_1. Every permutation in $P_{(C_1\cup C_3)\prec o_1}$ also belongs to $P_{C_1\prec o_1}$ but does not belong to $PS(C_1 \prec o_1)$. Thus, $PS(C_1 \prec o_1) = P_{C_1\prec o_1} \setminus P_{(C_1\cup C_3)\prec o_1}$. We can easily calculate $P_{(C_1\cup C_3)\prec o_1} = \{o_2o_3o_1, o_3o_2o_1\}$. Similarly, $PS(C_2 \prec o_1) = P_{C_2\prec o_1} \setminus P_{(C_2\cup C_3)\prec o_1}$. Thus, $\mathcal{B}_{o_1} = \bigcup_{C\in\{C_1,C_2\}} P_{C\prec o_1} \setminus \bigcup_{C\in\{C_1\cup C_3, C_2\cup C_3\}} P_{C\prec o_1} = \{o_2o_1o_3, o_3o_1o_2\}$. □

Let S be the set of all minimal syntheses. For any data owner o, denote by S_o the set of minimal syntheses that contain o and by $S_{\overline{o}}$ the set of minimal syntheses that do not contain o. Example 2 leads to the following general result.

Corollary 1. *In a data assemblage game, for any data owner o, $\mathcal{B}_o = \bigcup_{C\in S_o} P_{C<o} \setminus \bigcup_{C\in\{C_x\cup C_y | C_x\in S_o, C_y\in S_{\overline{o}}\}} P_{C<o} = \bigcup_{C\in S_o} P_{C<o} \setminus \bigcup_{C\in\{C_x\cup C_y|(C_x,C_y)\in S_o\times S_{\overline{o}}\}} P_{C<o}$.* □

Corollary 1 transforms the problem of counting $|\mathcal{B}_o|$ into computing $P_{C<o}$. Here, we need to consider every coalition $C \in \{C_x \cup C_y \mid (C_x, C_y) \in S_o \times S_{\overline{o}}\}$. It is easy to notice that, for any $C_1, C_2 \in \{C_x \cup C_y \mid (C_x, C_y) \in S_o \times S_{\overline{o}}\}$, if $C_1 \subset C_2$, then $P_{C_1<o} \supset P_{C_2<o}$, and thus, we do not need to consider C_2 in calculating $\mathcal{B}_o$ using Corollary 1. Formally, given a set S of coalitions, let $minimal(S) = \{C \mid C \in S, \nexists C' \in S \text{ s.t. } C \supset C'\}$ be the **set of minimal coalitions** in S. Then, we can rewrite $\mathcal{B}_o = \bigcup_{C\in S_o} P_{C<o} \setminus \bigcup_{C\in minimal(\{C_x\cup C_y|(C_x,C_y)\in S_o\times S_{\overline{o}}\})} P_{C<o}$.

For multiple sets of coalitions $S_1, \ldots, S_n \subseteq 2^O$, denote by $\mathcal{M}_{S_1\times\cdots\times S_n} = minimal(\{\cup_{i=1}^n C_i \mid (C_1, \ldots, C_n) \in S_1 \times \cdots \times S_n\})$ the set of minimal coalitions formed by concatenating one coalition from each S_i $(1 \le i \le n)$. When $n = 1$, we define $\mathcal{M}_{S_1} = minimal(S_1)$. From Corollary 1, we have

$$|B_o| = |\cup_{C\in S_o} P_{C<o}| - |\cup_{C\in\mathcal{M}_{S_o\times S_{\overline{o}}}} P_{C<o}| \tag{3}$$

According to the definition of $P_{C<o}$, $|P_{C<o}| = \binom{|O|}{|C|}(|C|-1)!(|O|-|C|)! = \frac{|O|!}{(|O|-|C|)!|C|!}(|C|-1)!(|O|-|C|)! = \frac{|O|!}{|C|}$. Besides, $\bigcap_{C\in S_o} P_{C<o} = P_{(\cup_{C\in S_o} C)<o}$ holds by definition. Applying the set union cardinality formula, we have $|\cup_{C\in S_o} P_{C<o}| = |O|! \cdot \sum_{\substack{X\subseteq S_o \\ |X|\ge 1}} \frac{(-1)^{|X|-1}}{|\cup_{C_x\in X} C_x|}$.

Given a set of coalitions X, denote by $Owner(X)$ the set of data owners participating in at least one coalition in X, that is, $Owner(X) = \cup_{C_x\in X} C_x$. Then, we have $|\cup_{C\in S_o} P_{C<o}| = |O|! \cdot \sum_{\substack{X\subseteq S_o \\ |X|\ge 1}} \frac{(-1)^{|X|-1}}{|Owner(X)|}$. Similarly, $|\cup_{C\in\mathcal{M}_{S_o\times S_{\overline{o}}}} P_{C<o}| = |O|! \cdot \sum_{\substack{X\subseteq\mathcal{M}_{S_o\times S_{\overline{o}}} \\ |X|\ge 1}} \frac{(-1)^{|X|-1}}{|Owner(X)|}$. Plugging into Equation 3, we get

$$|B_o| = |O|! \cdot \left(\sum_{\substack{X\subseteq S_o \\ |X|\ge 1}} \frac{(-1)^{|X|-1}}{|Owner(X)|} - \sum_{\substack{X\subseteq\mathcal{M}_{S_o\times S_{\overline{o}}} \\ |X|\ge 1}} \frac{(-1)^{|X|-1}}{|Owner(X)|} \right).$$

According to the definition of pivotal probabilities, we have

Theorem 7. *Given a set of owners O and a set of minimal syntheses in a data assemblage game, let $O' \subseteq O$ be the set of owners in minimal syntheses. Then, $\forall o \in O \setminus O', \psi(o) = 0$ and $\forall o \in O'$,*

$$\psi(o) = \sum_{\substack{X\subseteq S_o \\ |X|\ge 1}} \frac{(-1)^{|X|-1}}{|Owner(X)|} - \sum_{\substack{X\subseteq\mathcal{M}_{S_o\times S_{\overline{o}}} \\ |X|\ge 1}} \frac{(-1)^{|X|-1}}{|Owner(X)|}. \tag{4}$$

The complexity of computing the Shapley value using Theorem 7 is exponential to $\max(|S_o|, |\mathcal{M}_{S_o\times S_{\overline{o}}}|)$, but does not rely on the number of data owners.

4 DATA ASSEMBLAGE GAME DECOMPOSITION

In this section, we develop a game decomposition approach. We first describe the ideas and then illustrate the intuition using examples. Last, we present the algorithm.

4.1 Key Ideas

For the sake of brevity, hereafter we write a coalition as a sequence of data owners, such as $C = o_1o_2$ as a shorthand for $C = \{o_1, o_2\}$. Without specific mention, we also write $o_1 \cup o_2$ as a shorthand of $\{o_1\} \cup \{o_2\} = \{o_1, o_2\}$, which is a set of two coalitions where each contains only one data owner.

Using Theorem 7 to calculate the Shapley value needs to consider all possible subsets of S_o and $\mathcal{M}_{S_o\times S_{\overline{o}}}$. When there are many minimal syntheses, either S_o or $S_{\overline{o}}$ is large, and thus $\max(|S_o|, |\mathcal{M}_{S_o\times S_{\overline{o}}}|)$ is large. The computation is costly. Since syntheses are combinations of

data owners, likely the number of data owners is significantly smaller than the number of syntheses, that is, $|O| \ll \min(2^{|S_o|}-1, 2^{|\mathcal{M}_{S_o\times S_{\bar{o}}}|}-1)$. Instead of enumerating all possible subsets of S_o or $\mathcal{M}_{S_o\times S_{\bar{o}}}$, we can rearrange the summation in Equation 4 by dividing the subsets of S_o or $\mathcal{M}_{S_o\times S_{\bar{o}}}$ into up to $|O|$ groups according to the number of data owners participating in those subsets, that is, $\psi(o) = \sum_{k=1}^{|O|}\left(\sum_{\substack{X\subseteq S_o\\ |Owner(X)|=k}}\frac{(-1)^{|X|-1}}{|Owner(X)|} - \sum_{\substack{X\subseteq \mathcal{M}_{S_o\times S_{\bar{o}}}\\ |Owner(X)|=k}}\frac{(-1)^{|X|-1}}{|Owner(X)|}\right) = \sum_{k=1}^{|O|}\frac{1}{k}\left(\sum_{\substack{X\subseteq S_o\\ |Owner(X)|=k}}(-1)^{|X|-1} - \sum_{\substack{X\subseteq \mathcal{M}_{S_o\times S_{\bar{o}}}\\ |Owner(X)|=k}}(-1)^{|X|-1}\right)$. For a set of coalitions S and a number k ($k \geq 0$), the **inclusion-exclusion coefficient** (or *IEC* in short)[4] of S with respect to k is $IEC(S,k) = \sum_{\substack{X\subseteq S\\ |Owner(X)|=k}}(-1)^{|X|-1}$. If there is no subset of S with k data owners, we define $IEC(S,k) = 0$ for $k \geq 1$, and $IEC(S,0) = -1$. We immediately have

$$\psi(o) = \sum_{k=1}^{|O|}\frac{IEC(S_o,k) - IEC(\mathcal{M}_{S_o\times S_{\bar{o}}},k)}{k} \tag{5}$$

To compute $IEC(S_o,k)$ and $IEC(\mathcal{M}_{S_o\times S_{\bar{o}}},k)$, we try to find non-empty subsets of S_o or $\mathcal{M}_{S_o\times S_{\bar{o}}}$ with k data owners that can be composed using smaller coalitions. In many situations, due to the redundancy among different data owners' data sets, there may exist some structural patterns in minimal syntheses. For example, in Example 1, every minimal synthesis contains o_1 and picks two from o_2, o_3, and o_4. This insight inspires a divide-and-conquer strategy: we can divide the set of minimal syntheses into exclusive subsets such that each subset can be constructed using the combinations of multiple smaller sub-coalitions. After the decomposition, we can compose S_o and $\mathcal{M}_{S_o\times S_{\bar{o}}}$ with smaller coalitions and compute $IEC(S_o,k)$ and $IEC(\mathcal{M}_{S_o\times S_{\bar{o}}},k)$ faster by enumerating smaller coalitions. This strategy is more effective for data assemblage tasks, since typically data sets are assembled in some structured way, such as join, selection, and projection. In this section, we develop the strategy and implement it using game decomposition.

Let us illustrate the intuition using three examples, each representing a different decomposition scenario. Each example is introduced in four steps. The first step is *minimal synthesis decomposition*, where we partition minimal syntheses by identifying a structural pattern. This pattern allows us to partition both S_o and $\mathcal{M}_{S_o\times S_{\bar{o}}}$ into smaller coalitions. In the second step, *IEC computation decomposition*, we elucidate how to decompose the computation of $IEC(S_o,k)$ and $IEC(\mathcal{M}_{S_o\times S_{\bar{o}}},k)$ by leveraging the insights from minimal synthesis decomposition. The general idea is to divide the subsets of S_o and $\mathcal{M}_{S_o\times S_{\bar{o}}}$ with k data owners into distinct groups, where each group can be evaluated in closed-form or shares the same computation process. In the third step, *Shapley value computation*, we detail the Shapley value calculation for a data owner in the provided example, highlighting the reduced computational complexity achieved through IEC computation decomposition. In the fourth step, *generalization*, we generalize our discussions based on the insights gained from the example.

4.2 Vertical Decomposition

We begin with the vertical decomposition, where a set of minimal syntheses can be decomposed into the Cartesian product of a group of smaller coalitions such that each pair of these coalitions does not share any common data owners.

[4]We coin the name because that each IEC represents a coefficient in the well-known inclusion-exclusion principle in combinatorics [9].

Minimal Synthesis Decomposition. Consider a set of data owners $O = \{o_1, \ldots, o_6\}$ producing a target data set in coalition and the minimal syntheses are $C_1 = o_1o_2o_5$, $C_2 = o_1o_2o_6$, $C_3 = o_1o_3o_5$, $C_4 = o_1o_3o_6$, $C_5 = o_4o_5$, and $C_6 = o_4o_6$. Interestingly, the set of all minimal syntheses can be decomposed into the Cartesian product of two groups of coalitions, $\{o_1o_2, o_1o_3, o_4\} \times \{o_5, o_6\}$, that is, every minimal synthesis contains one among o_1o_2, o_1o_3, and o_4, and one between o_5 and o_6. Accordingly we can decompose the minimal syntheses into two groups of smaller coalitions, $S_1 = \{o_1o_2, o_1o_3, o_4\}$, $S_2 = \{o_5, o_6\}$. The sets of data owners participating in the two groups are $O_1 = Owner(S_1) = \{o_1, o_2, o_3, o_4\}$ and $O_2 = Owner(S_2) = \{o_5, o_6\}$, respectively. The minimal syntheses can be constructed in a structured way $\mathcal{M}_{S_1 \times S_2}$.

Owner o_1 only participates in some coalitions in S_1. Denote by $S_{1_{o_1}}$ the set of coalitions in S_1 that contain o_1 and by $S_{1_{\overline{o_1}}}$ the set of coalitions in S_1 that do not contain o_1. Then, we have $S_{o_1} = \mathcal{M}_{S_{1_{o_1}} \times S_2} = \{C_1, C_2, C_3, C_4\}$ and $\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}} = \mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}} \times S_2} = \{o_1o_2o_4o_5, o_1o_2o_4o_6, o_1o_3o_4o_5, o_1o_3o_4o_6\}$.

IEC Computation Decomposition. To compute $\psi(o_1)$ using Equation 5, we need to compute $IEC(S_{o_1}, k) - IEC(\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}, k)$ for $k \in [1, |O|]$.

We can compute $IEC(S_{o_1}, k)$ using the decomposition $S_{o_1} = \mathcal{M}_{S_{1_{o_1}} \times S_2}$. Instead of enumerating all non-empty subsets of S_{o_1}, we can efficiently compute $IEC(S_{o_1}, k)$ by dividing the coalitions in S_{o_1} with k data owners into several groups according to their projections onto $S_{1_{o_1}}$ and S_2. This division allows us to evaluate each group quickly using closed-form expressions.

For each non-empty subset $W \subseteq S_{o_1} = \mathcal{M}_{S_{1_{o_1}} \times S_2}$ with k data owners, W can be decomposed into two parts, $W[S_{1_{o_1}}] \subseteq S_{1_{o_1}}$ and $W[S_2] \subseteq S_2$, where $W[S_{1_{o_1}}]$ and $W[S_2]$ are the subsets of coalitions in $S_{1_{o_1}}$ and S_2 that compose W, respectively. Formally, for any subset $W \subseteq \mathcal{M}_{T_1 \oslash \cdots \oslash T_l}$, where $T_1, \ldots, T_l$ are sets of coalitions and $\oslash \in \{\times, \cup\}$[5], the **projection** of W on T_i is defined as $W[T_i] = \{C \mid C \in T_i, \exists C' \in W \text{ s.t. } C \subseteq C'\}$.

Clearly, the k data owners in W must be from either $W[S_{1_{o_1}}]$ or $W[S_2]$. Thus, $|\,Owner(W[S_{1_{o_1}}])| \geq 1$, $|\,Owner(W[S_2])| \geq 1$, and $|\,Owner(W[S_{1_{o_1}}])| + |\,Owner(W[S_2])| = k$. Therefore, we can divide all non-empty subsets $W \subseteq S_{o_1}$ with k data owners into $k-1$ groups according to the number of data owners in $W[S_{o_1}]$. We have

$$\begin{aligned} IEC(S_{o_1}, k) &= IEC(\mathcal{M}_{S_{1_{o_1}} \times S_2}, k) \\ &= \sum_{W \subseteq \mathcal{M}_{S_{1_{o_1}} \times S_2}, |\,Owner(W)| = k} (-1)^{|W|-1} \\ &= \sum_{k_1=1}^{k-1} \sum_{\substack{W \subseteq \mathcal{M}_{S_{1_{o_1}} \times S_2} \\ |\,Owner(W)| = k, |\,Owner(W[S_{1_{o_1}}])| = k_1}} (-1)^{|W|-1} \end{aligned} \tag{6}$$

On the right-hand side of Equation 6, the IEC of each group of subsets $W \subseteq \mathcal{M}_{S_{1_{o_1}} \times S_2}$, where $|\,Owner(W)| = k$ and $|\,Owner(W[S_{o_1}])| = k_1$, can be computed in closed form. That is, for $k \in [1, |O_1|]$ and $k_1 \in [1, k-1]$, when $S_{1_{o_1}}$ and S_2 do not share any common data owner, we have

$$\sum_{\substack{W \subseteq \mathcal{M}_{S_{1_{o_1}} \times S_2}, |\,Owner(W)| = k \\ |\,Owner(W[S_{1_{o_1}}])| = k_1}} (-1)^{|W|-1} = IEC(S_{1_{o_1}}, k_1) \cdot IEC(S_2, k - k_1) \tag{7}$$

As an illustration of Equation 7, consider $k = 3$ and $k_1 = 2$. $W_1 = \{o_1o_2o_5\}$, $W_2 = \{o_1o_2o_6\}$, $W_3 = \{o_1o_3o_5\}$, and $W_4 = \{o_1o_3o_6\}$ are the 4 non-empty subsets of S_{o_1} with 3 data owners where $|\,Owner(W[S_{1_{o_1}}])| = 2$. $IEC(S_{1_{o_1}}, 2) \cdot IEC(S_2, 3-2) = 4 \times (-1)^{1-1} = 4$. Now plugging Equation 7

[5] $\cup$ will be used in horizontal decomposition (Section 4.3).

into Equation 6, we have

$$\begin{aligned} IEC(S_{o_1},k) &= IEC(\mathcal{M}_{S_{1_{o_1}}\times S_2},k) \\ &= \sum_{k_1=1}^{k-1}\left(IEC(S_{1_{o_1}},k_1)\cdot IEC(S_2,k-k_1)\right) \\ &= \sum_{k_1+k_2=k,k_1\geq 1,\ k_2\geq 1}\left(IEC(S_{1_{o_1}},k_1)\cdot IEC(S_2,k_2)\right) \end{aligned} \tag{8}$$

Equation 8 shows how we break down the IEC computation for non-empty subsets of S_{o_1} with k data owners into the IEC computation of smaller coalitions, $S_{1_{o_1}}$ and S_2, using the decomposition $S_{o_1} = \mathcal{M}_{S_{1_{o_1}}\times S_2}$. Similarly, we have $IEC(\mathcal{M}_{S_{o_1}\times S_{\overline{o_1}}},k) = IEC(\mathcal{M}_{S_{1_{o_1}}\times S_{1_{\overline{o_1}}}\times S_2},k) = \sum_{\substack{k_1+k_2=k\\ k_1\geq 1,\ k_2\geq 1}}\left(IEC(\mathcal{M}_{S_{1_{o_1}}\times S_{1_{\overline{o_1}}}},k_1)\cdot IEC(S_2,k_2)\right)$.

Combing the above two equations, we get

$$IEC(S_{o_1},k) - IEC(\mathcal{M}_{S_{o_1}\times S_{\overline{o_1}}},k) = \sum_{\substack{k_1+k_2=k\\ k_1\geq 1,\ k_2\geq 1}}\left(\left(IEC(S_{1_{o_1}},k_1) - IEC(\mathcal{M}_{S_{1_{o_1}}\times S_{1_{\overline{o_1}}}},k_1)\right)\cdot IEC(S_2,k_2)\right) \tag{9}$$

Plugging Equation 9 into Equation 5, we get

$$\psi(o_1) = \sum_{k_1=1}^{|O_1|}\sum_{k_2=1}^{|O|-|O_1|}\frac{\left(IEC(S_{1_{o_1}},k_1) - IEC(\mathcal{M}_{S_{1_{o_1}}\times S_{1_{\overline{o_1}}}},k_1)\right)\cdot IEC(S_2,k_2)}{k_1+k_2} \tag{10}$$

Shapley Value Computation. In Equation 10, we enumerate all non-empty subsets of $S_{1_{o_1}}$, $\mathcal{M}_{S_{1_{o_1}}\times S_{1_{\overline{o_1}}}}$, and S_2, and consider $|O_1|\cdot(|O|-|O_1|)$ possible combinations of $IEC(S_{1_{o_1}},k_1) - IEC(\mathcal{M}_{S_{1_{o_1}}\times S_{1_{\overline{o_1}}}},k_1)$ ($k_1\in[1,|O_1|]$) and $IEC(S_2,k_2)$ ($k_2\in[1,|O|-|O_1|]$). Instead of enumerating all $(2^{|S_{o_1}|}-1)+(2^{\left|\mathcal{M}_{S_{o_1}\times S_{\overline{o_1}}}\right|}-1) = 30$ non-empty subsets of S_{o_1} and $\mathcal{M}_{S_{o_1}\times S_{\overline{o_1}}}$ using Equation 4, we only need to consider $(2^{\left|S_{1_{o_1}}\right|}-1)+(2^{\left|\mathcal{M}_{S_{1_{o_1}}\times S_{1_{\overline{o_1}}}}\right|}-1)+(2^{|S_2|}-1)+|O_1|\cdot(|O|-|O_1|) = 17$ terms in total using Equation 10. The reduction is significant when $|S_{o_1}|$ or $\left|\mathcal{M}_{S_{o_1}\times S_{\overline{o_1}}}\right|$ is large.

Using Equation 10 we can compute $\psi(o_1) = \frac{(2-0)\times 2}{2+1} + \frac{(2-0)\times(-1)}{2+2} + \frac{(-1-2)\times 2}{3+1} + \frac{(-1-2)\times(-1)}{3+2} + \frac{(0-(-1))\times 2}{4+1} + \frac{(0-(-1))\times(-1)}{4+2} = \frac{1}{6}$.

In Equation 10, $IEC(S_2,k_2)$ only refers to the data owners in $O\setminus O_1$. All data owners in O_1 share the same $IEC(S_2,k_2)$ value. Thus, we can precompute and reuse $IEC(S_2,k_2)$ for $k_2\in[1,|O|-|O_1|]$ when calculating the Shapley values for the data owners in O_1.

Generalization. We can generalize the above discussion. Partitioning minimal syntheses into the Cartesian product of non-overlapping coalitions, each devoid of shared data owners, allows us to decompose both S_o and $\mathcal{M}_{S_o\times S_{\overline{o}}}$ into smaller coalitions. This, in turn, enables the decomposition of the computation of $IEC(S_o,k) - IEC(\mathcal{M}_{S_o\times S_{\overline{o}}},k)$ into the IEC computation of smaller coalitions.

Lemma 1. *Consider a data assemblage game (O,v) without dummies and the set of minimal syntheses S, where S can be decomposed into n groups of coalitions $S_1,\ldots,S_n$ such that (1) for any $1\leq i<j\leq n$, $O_i\cap O_j=\emptyset$ and $\cup_{i=1}^{n}O_i = O$, where $O_i = Owner(S_i)$ $(1\leq i\leq n)$; and (2) $S = \mathcal{M}_{S_1\times\cdots\times S_n}$. For a group of coalitions S_i $(1\leq i\leq n)$, let $V_{\overline{S_i}} = \mathcal{M}_{S_1\times\cdots\times S_{i-1}\times S_{i+1}\cdots\times S_n}$. Then, for a data owner $o\in O_i$ and $k\in[1,|O|]$,*

$$IEC(S_o,k) - IEC(\mathcal{M}_{S_o\times S_{\overline{o}}},k) = \sum_{\substack{k_1+k_2=k\\ k_1\geq 1,\ k_2\geq 1}}\left(\left(IEC(S_{i_o},k_1) - IEC(\mathcal{M}_{S_{i_o}\times S_{i_{\overline{o}}}},k_1)\right)\cdot IEC(V_{\overline{S_i}},k_2)\right), \tag{11}$$

where

$$IEC(V_{\overline{S_i}}, k_2) = IEC(\mathcal{M}_{S_1 \times \cdots \times S_{i-1} \times S_{i+1} \cdots \times S_n}, k_2) = \sum_{\substack{\sum_{j=1, j\neq i}^{n} k'_j = k_2 \\ \forall j \in [1,n] \, and \, j \neq i, k'_j \geq 1}} \prod_{j=1, j\neq i}^{n} IEC(S_j, k'_j) \tag{12}$$

Proof sketch. Equation 11 follows Equation 9 by replacing S_2 by $V_{\overline{S_i}}$. Equation 12 is a generalization of Equation 8. □

4.3 Horizontal Decomposition

Now we look at the horizontal decomposition, where a set of minimal syntheses can be partitioned into exclusive subsets such that each pair of these subsets does not share any common data owners.

Minimal Synthesis Decomposition. Consider a set of data owners $O = \{o_1, \ldots, o_8\}$ producing a target data set in coalition and the minimal syntheses are $C_1 = o_1o_2$, $C_2 = o_1o_3$, $C_3 = o_2o_3$, $C_4 = o_4o_5$, $C_5 = o_5o_6$, $C_6 = o_6o_7$, and $C_7 = o_8$. The set of minimal syntheses can be partitioned into three groups, $S_1 = \{o_1o_2, o_1o_3, o_2o_3\}$, $S_2 = \{o_4o_5, o_5o_6, o_6o_7\}$, and $S_3 = \{o_8\}$, such that each data owner participates in only the minimal syntheses of one group and every minimal synthesis belongs to only one group. Let $O_i = Owner(S_i)$ $(1 \leq i \leq 3)$, that is, $O_1 = \{o_1, o_2, o_3\}$, $O_2 = \{o_4, o_5, o_6, o_7\}$, and $O_3 = \{o_8\}$. For $o_1 \in O_1$, $S_{1_{o_1}} = \{o_1o_2, o_1o_3\}$ and $S_{1_{\overline{o_1}}} = \{o_2o_3\}$. $S_{o_1} = S_{1_{o_1}}$ and $\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}} = \mathcal{M}_{S_{1_{o_1}} \times (S_{1_{\overline{o_1}}} \cup S_2 \cup S_3)}$. Let $L_{\overline{S_1}} = S_2 \cup S_3$ be the set of minimal syntheses that do not contain any data owner in O_1. Then, $\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}} = \mathcal{M}_{S_{1_{o_1}} \times (S_{1_{\overline{o_1}}} \cup L_{\overline{S_1}})}$.

IEC Computation Decomposition. To compute $\psi(o_1)$ using Equation 5, we need to compute $OS(S_{o_1}, k) - IEC(\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}, k)$ for $k \in [1, |O|]$. Clearly, $IEC(S_{o_1}, k) = IEC(S_{1_{o_1}}, k)$.

Since $S_{1_{o_1}}$ and $S_{1_{\overline{o_1}}} \cup L_{\overline{S_1}}$ may share common data owners, the computation of $IEC(\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}, k)$ cannot be decomposed directly using Equation 8. Instead, we can compute $IEC(\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}, k)$ using the decomposition $\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}} = \mathcal{M}_{S_{1_{o_1}} \times (S_{1_{\overline{o_1}}} \cup L_{\overline{S_1}})} = \mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}} \cup \mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}$. We can categorize all non-empty subsets of $\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}$ with k data owners into distinct cases according to whether their projections onto $\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}$ and $\mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}$ are empty or not. Through this categorization, we observe that the subsets falling into the same case share the computation process.

Each non-empty subset $W \subseteq \mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}$ with k data owners can be decomposed into two parts, $W[\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}] \subseteq \mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}$ and $W[\mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}] \subseteq \mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}$, where $W[\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}]$ and $W[\mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}]$ are subsets of coalitions in $\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}$ and $\mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}$ that compose W, respectively. The projections $W[\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}]$ and $W[\mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}]$ cannot be empty at the same time. Depending on whether they are empty or not, all possible non-empty subsets $W \subseteq \mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}$ with k data owners can be divided into 3 cases, $W[\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}] \neq \emptyset$ and $W[\mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}] \neq \emptyset$; $W[\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}] = \emptyset$ and $W[\mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}] \neq \emptyset$; and $W[\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}] \neq \emptyset$ and $W[\mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}] = \emptyset$. All subsets falling into the same case share the same computation process. That is, for $k \in [1, |O|]$,

$$\begin{aligned} &IEC(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}} \cup \mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}, k) \\ &= IEC(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}, k) + IEC(\mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}, k) - IEC(\mathcal{M}_{(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}) \times (\mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}})}, k) \end{aligned} \tag{13}$$

As an illustration of Equation 13, when $k = 3$, $\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}} = \{o_1o_2o_3\}$ and $\mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}} = \{o_1o_2o_8, o_1o_3o_8, o_1o_2o_4o_5, o_1o_2o_5o_6, o_1o_2o_6o_7, o_1o_3o_4o_5, o_1o_3o_5o_6, o_1o_3o_6o_7\}$. There are 3 possible subsets of

$\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}} \cup \mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}$ with 3 data owners, that is, $\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}$ itself and the two single coalition subsets of $\mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}$, namely $\{o_1o_2o_8\}$ and $\{o_1o_3o_8\}$. Since $\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}$ has exact 3 data owners, the first term in Equation 13 $IEC(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}, 3) = (-1)^{1-1} = 1$. Similarly, the second term $IEC(\mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}, 3) = 2 \times (-1)^{1-1} = 2$. As there is no subset of $\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}} \cup \mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}$ that has 3 data owners and contains coalitions from both $\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}$ and $\mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}$, the third term $IEC(\mathcal{M}_{(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}) \times (\mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}})}, 3) = 0$. Thus, we have $IEC(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}} \cup \mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}, 3) = 3$.

The computation of the second and the third terms on the right-hand side of Equation 13 can be further simplified. We have

$$\begin{aligned} IEC(\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}, k) &= IEC(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}} \cup \mathcal{M}_{S_{1_{o_1}} \times L_{\overline{S_1}}}, k) \\ &= IEC(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}, k) + \sum_{\substack{k_1+k_2=k \\ k_1 \geq 1,\ k_2 \geq 1}} IEC(S_{1_{o_1}}, k_1) \cdot IEC(L_{\overline{S_1}}, k_2) \\ &\quad - \sum_{k_1+k_2=k,\ k_1 \geq 1,\ k_2 \geq 1} IEC(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}, k_1) \cdot IEC(L_{\overline{S_1}}, k_2) \end{aligned} \tag{14}$$

Since $IEC(S_{o_1}, k) = IEC(S_{1_{o_1}}, k)$, we have

$$\begin{aligned} &IEC(S_{o_1}, k) - IEC(\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}, k) \\ &= IEC(S_{1_{o_1}}, k) - IEC(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}, k) - \sum_{\substack{k_1+k_2=k \\ k_1 \geq 1,\ k_2 \geq 1}} \Big(\big(IEC(S_{1_{o_1}}, k_1) - IEC(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}, k_1) \big) \cdot IEC(L_{\overline{S_1}}, k_2) \Big) \\ &= \sum_{\substack{k_1+k_2=k \\ k_1 \geq 1,\ k_2 \geq 0}} \Big(\big(IEC(S_{1_{o_1}}, k_1) - IEC(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}, k_1) \big) \cdot \big(- IEC(L_{\overline{S_1}}, k_2) \big) \Big) \end{aligned} \tag{15}$$

Equation 15 shows how we break down the computation of $IEC(S_{o_1}, k) - IEC(\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}, k)$ into the IEC computation of smaller coalitions, $S_{1_{o_1}}$, $\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}$, and $L_{\overline{S_1}}$, by leveraging minimal synthesis decomposition. Notably, the computation of $IEC(L_{\overline{S_1}}, k_2)$ can be further decomposed using the decomposition $L_{\overline{S_1}} = S_2 \cup S_3$. Since S_2 and S_3 share no common data owner, for any $k_2 \geq 0$, we have

$$IEC(L_{\overline{S_1}}, k_2) = - \sum_{k'_2+k'_3=k_2,\ k'_2 \geq 0,\ k'_3 \geq 0} \big(IEC(S_2, k'_2) \cdot IEC(S_3, k'_3) \big) \tag{16}$$

Plugging Equations 15 and 16 into Equation 5, we have

$$\psi(o_1) = \sum_{k_1=1}^{|O_1|} \sum_{k_2=0}^{|O|-|O_1|} \frac{\big(IEC(S_{1_{o_1}}, k_1) - IEC(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}, k_1) \big) \cdot \big(- IEC(L_{\overline{S_1}}, k_2) \big)}{k_1 + k_2}, \tag{17}$$

where $IEC(L_{\overline{S_1}}, k_2) = - \sum_{k'_2+k'_3=k_2,\ k'_2 \geq 0,\ k'_3 \geq 0} \big(IEC(S_2, k'_2) \cdot IEC(S_3, k'_3) \big)$.

Shapley Value Computation. In Equation 17, we first consider the computation of $IEC(L_{\overline{S_1}}, k_2)$. By definition, $IEC(L_{\overline{S_1}}, 0) = -1$, so we only need to compute $IEC(L_{\overline{S_1}}, k_2)$ for $k_2 \geq 1$. Additionally, $IEC(S_2, 0) = IEC(S_3, 0) = -1$, which means we only need to enumerate all non-empty subsets of S_2 and S_3. We consider $|O_2| \cdot |O_3|$ possible combinations of $IEC(S_2, k'_2)$ ($k'_2 \in [1, |O_2|]$) and $IEC(S_3, k'_3)$ ($k'_3 \in [1, |O_3|]$). We exclude combinations when $k'_2 = 0$ or $k'_3 = 0$ since they result in $- IEC(S_3, k'_3)$ and $- IEC(S_2, k'_2)$, respectively, which are computed during the enumeration of non-empty subsets of S_3 and S_2. Thus, to compute $IEC(L_{\overline{S_1}}, k_2)$ for $k_2 \in [1, |O| - |O_1|]$, instead of enumerating all $2^{|L_{\overline{S_1}}|} - 1 = 15$ non-empty subsets of $L_{\overline{S_1}}$, we only need to consider $2^{|S_2|} - 1 + 2^{|S_3|} - 1 + |O_2| \cdot |O_3| = 12$ terms using Equation 16. The reduction is significant when $|L_{\overline{S_1}}|$ is large.

In Equation 17, in addition to the above 12 terms to compute $IEC(L_{\overline{S_1}}, k_2)$ for $k_2 \in [0, |O| - |O_1|]$, we enumerate all non-empty subsets of $S_{1_{o_1}}$ and $\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}$, respectively, and consider all $|O_1| \cdot (|O| - |O_1| + 1)$ possible combinations of $IEC(S_{1_{o_1}}, k_1) - IEC(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}, k_1)$ ($k_1 \in [1, |O_1|]$) and $IEC(L_{\overline{S_1}}, k_2)$. Instead of enumerating all $(2^{|S_{o_1}|} - 1) + (2^{\left|\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}\right|} - 1) = 514$ non-empty subsets of S_{o_1} and $\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}$ using Equation 4, we only need to consider $(2^{|S_{1_{o_1}}|} - 1) + (2^{\left|\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}\right|} - 1) + 12 + |O_1| \cdot (|O| - |O_1| + 1) = 34$ terms in total using Equation 17. The reduction is significant when $|S_{o_1}|$ or $|\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}|$ is large.

Now using Equation 17 we can compute $\psi(o_1) = \frac{(2-0)\times 1}{2+0} + \frac{(2-0)\times(-1)}{2+1} + \frac{(2-0)\times(-3)}{2+2} + \frac{(2-0)\times 5}{2+3} + \frac{(2-0)\times(-2)}{2+4} + \frac{(-1-1)\times 1}{3+0} + \frac{(-1-1)\times(-1)}{3+1} + \frac{(-1-1)\times(-3)}{3+2} + \frac{(-1-1)\times(-5)}{3+3} + \frac{(-1-1)\times(-2)}{3+4} = \frac{11}{105}$.

In Equation 17, $IEC(L_{\overline{S_1}}, k_2)$ only refers to the data owners in $O \setminus O_1$. All data owners in O_1 share the same $IEC(L_{\overline{S_1}}, k_2)$ value. We can precompute and reuse $IEC(L_{\overline{S_1}}, k_2)$ for $k_2 \in [0, |O| - |O_1|]$ when calculating the Shapley values for data owners in O_1.

Generalization. We can generalize the above discussion. Partitioning minimal syntheses into distinct, non-overlapping (in terms of data owners) subsets allows us to partition both S_o and $\mathcal{M}_{S_o \times S_{\overline{o}}}$ into smaller coalitions. Subsequently, this allows the computation of $IEC(S_o, k) - IEC(\mathcal{M}_{S_o \times S_{\overline{o}}}, k)$ to be decomposed using IEC within these smaller coalitions.

Lemma 2. *Consider a data assemblage game (O, v) without dummies and the set of minimal syntheses S, where S can be decomposed into n groups of coalitions $S_1, \ldots, S_n$ such that (1) for any $1 \le i < j \le n$, $O_i \cap O_j = \emptyset$ and $\cup_{i=1}^{n} O_i = O$, where $O_i = Owner(S_i)$ $(1 \le i \le n)$; and (2) $S = \cup_{i=1}^{n} S_i$. For a group of coalitions S_i $(1 \le i \le n)$, let $L_{\overline{S_i}} = S_1 \cup \cdots \cup S_{i-1} \cup S_{i+1} \cdots \cup S_n$. Then, for a data owner $o \in O_i (1 \le i \le n)$ and $k \in [1, |O|]$,*

$$IEC(S_o, k) - IEC(\mathcal{M}_{S_o \times S_{\overline{o}}}, k) = \sum_{\substack{k_1 + k_2 = k \\ k_1 \ge 1,\ k_2 \ge 0}} \Big(\big(IEC(S_{i_o}, k_1) - IEC(\mathcal{M}_{S_{i_o} \times S_{i_{\overline{o}}}}, k_1) \big) \cdot \big(- IEC(L_{\overline{S_i}}, k_2) \big) \Big), \tag{18}$$

where

$$\begin{aligned} IEC(L_{\overline{S_i}}, k_2) &= IEC(S_1 \cup \cdots \cup S_{i-1} \cup S_{i+1} \cdots \cup S_n, k_2) \\ &= (-1)^{n-2} \sum_{\substack{\sum_{j=1, j\neq i}^{n} k'_j = k_2 \\ \forall j \in [1,n] \text{ and } j \neq i, k'_j \ge 0}} \prod_{j=1, j\neq i}^{n} IEC(S_j, k'_j) \end{aligned} \tag{19}$$

Proof sketch. Equation 18 can be obtained following Equation 15. Equation 19 is a generalization of Equation 16. □

4.4 Hybrid Decomposition

Now let us consider a more sophisticated case where a set of minimal syntheses cannot be directly partitioned vertically or horizontally in the ways shown in Sections 4.2 and 4.3. We divide the minimal syntheses into exclusive subsets that can be further decomposed using vertical decomposition. Different from horizontal decomposition, in this partitioning we allow that some exclusive subsets may have data owners in common as long as upon decomposing each pair of all resulting smaller coalitions shall have no data owners in common.

Minimal Synthesis Decomposition. Consider a set of data owners $O = \{o_1, \ldots, o_5\}$ producing a target data set in coalition, and the minimal syntheses are $C_1 = o_1o_3$, $C_2 = o_1o_4$, $C_3 = o_2o_3$, $C_4 = o_2o_4$, $C_5 = o_1o_5$, $C_6 = o_2o_5$, $C_7 = o_3o_5$, and $C_8 = o_4o_5$. We can partition the minimal syntheses into three exclusive subsets: $\{C_1, C_2, C_3, C_4\}$, $\{C_5, C_6\}$, and $\{C_7, C_8\}$ such that each subset can be further decomposed as the Cartesian product of three sets of smaller coalitions, namely, $S_1 =$

$\{o_1, o_2\}$, $S_2 = \{o_3, o_4\}$, and $S_3 = \{o_5\}$, that is, $\{C_1, C_2, C_4, C_4\} = \mathcal{M}_{S_1 \times S_2}$, $\{C_5, C_6\} = \mathcal{M}_{S_1 \times S_3}$, and $\{C_6, C_7\} = \mathcal{M}_{S_2 \times S_3}$. Let $O_i = Owner(S_i)$ $(1 \le i \le 3)$, that is, $O_1 = \{o_1, o_2\}$, $O_2 = \{o_3, o_4\}$, and $O_3 = \{o_5\}$. Under this decomposition, the minimal syntheses can be constructed in a structured way, that is, $\mathcal{M}_{S_1 \times S_2} \cup \mathcal{M}_{S_1 \times S_3} \cup \mathcal{M}_{S_2 \times S_3}$.

For $o_1 \in O_1$, $S_{1_{o_1}} = \{o_1\}$ and $S_{1_{\overline{o_1}}} = \{o_2\}$. Let $V_{\overline{S_1}} = S_2 \cup S_3$ be the set of coalitions with data owners in $O \setminus O_1$ that can compose minimal syntheses with each coalition in S_1, and $L_{\overline{S_1}} = \mathcal{M}_{S_2 \times S_3}$ the set of minimal syntheses that do not contain any data owner in S_1. We have $S_{o_1} = \mathcal{M}_{S_{1_{o_1}} \times V_{\overline{S_1}}}$, $S_{\overline{o_1}} = \mathcal{M}_{S_{1_{\overline{o_1}}} \times V_{\overline{S_1}}} \cup L_{\overline{S_1}}$, and $\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}} = \mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}} \times V_{\overline{S_1}}} \cup \mathcal{M}_{S_{1_{o_1}} \times V_{\overline{S_1}} \times L_{\overline{S_1}}}$.

IEC Computation Decomposition. To compute $\psi(o_1)$ using Equation 5, we need to compute $IEC(S_{o_1}, k) - IEC(\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}, k)$ for $k \in [1, |O|]$. We can decompose the computation of $IEC(S_{o_1}, k) - IEC(\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}, k)$ for $k \in [1, |O|]$ using the results obtained in Sections 4.2 and 4.3. We have

$$\begin{aligned} &IEC(S_{o_1}, k) - IEC(\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}, k) \\ &= \sum_{k_1+k_2=k,\ k_1 \ge 1,\ k_2 \ge 1} \Big(\big(IEC(S_{1_{o_1}}, k_1) - IEC(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}, k_1) \big) \cdot \big(IEC(V_{\overline{S_1}}, k_2) - IEC(\mathcal{M}_{V_{\overline{S_1}} \times L_{\overline{S_1}}}, k_2) \big) \Big) \end{aligned} \tag{20}$$

On the right-hand side of Equation 20, we can further decompose the computation of $IEC(V_{\overline{S_1}}, k_2) - IEC(\mathcal{M}_{V_{\overline{S_1}} \times L_{\overline{S_1}}}, k_2)$ using the decompositions $V_{\overline{S_1}} = S_2 \cup S_3$ and $L_{\overline{S_1}} = \mathcal{M}_{S_2 \times S_3}$. Plugging Equation 16 into $IEC(V_{\overline{S_1}}, k_2)$ and Equation 8 into $IEC(\mathcal{M}_{V_{\overline{S_1}} \times L_{\overline{S_1}}}, k_2)$, we have $IEC(V_{\overline{S_1}}, k_2) - IEC(\mathcal{M}_{V_{\overline{S_1}} \times L_{\overline{S_1}}}, k_2) = IEC(S_2, k_2) + IEC(S_3, k_2) - 2 \cdot \sum_{\substack{k_2' + k_3' = k_2 \\ k_2' \ge 1,\ k_3' \ge 1}} \big(IEC(S_2, k_2') \cdot IEC(S_3, k_3') \big)$. Plugging this equation and Equation 20 into Equation 5, we have

$$\psi(o_1) = \sum_{k_1=1}^{|O_1|} \sum_{k_2=1}^{|O|-|O_1|} \left(\Big(IEC(S_{1_{o_1}}, k_1) - IEC(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}, k_1) \Big) \cdot \frac{IEC(V_{\overline{S_1}}, k_2) - IEC(\mathcal{M}_{V_{\overline{S_1}} \times L_{\overline{S_1}}}, k_2)}{k_1 + k_2} \right), \tag{21}$$

where $IEC(V_{\overline{S_1}}, k_2) - IEC(\mathcal{M}_{V_{\overline{S_1}} \times L_{\overline{S_1}}}, k_2) = IEC(S_2, k_2) + IEC(S_3, k_2) - 2 \cdot \sum_{\substack{k_2' + k_3' = k_2 \\ k_2' \ge 1,\ k_3' \ge 1}} \Big(IEC(S_2, k_2') \cdot IEC(S_3, k_3') \Big)$.

Shapley Value Computation. In Equation 21, to compute $IEC(V_{\overline{S_1}}, k_2) - IEC(\mathcal{M}_{V_{\overline{S_1}} \times L_{\overline{S_1}}}, k_2)$, we enumerate all non-empty subsets of S_2 and S_3, and consider $|O_2| \cdot |O_3|$ possible combinations of $IEC(S_2, k_2')$ $(k_2' \in [1, |O_2|])$ and $IEC(S_3, k_3')$ $(k_3' \in [1, |O_3|])$. Thus, instead of enumerating all $2^{|V_{\overline{S_1}}|} - 1 + 2^{|\mathcal{M}_{V_{\overline{S_1}} \times L_{\overline{S_1}}}|} - 1 = 10$ non-empty subsets of $V_{\overline{S_1}}$ and $\mathcal{M}_{V_{\overline{S_1}} \times L_{\overline{S_1}}}$, we only need to consider $2^{|S_2|} - 1 + 2^{|S_3|} - 1 + |O_2| \cdot |O_3| = 6$ terms in total in the computation. The reduction is significant when $|V_{\overline{S_1}}|$ or $\left|\mathcal{M}_{V_{\overline{S_1}} \times L_{\overline{S_1}}}\right|$ is large.

In Equation 21, in addition to the 6 terms in computing $IEC(V_{\overline{S_1}}, k_2) - IEC(\mathcal{M}_{V_{\overline{S_1}} \times L_{\overline{S_1}}}, k_2)$ $(k_2 \in [1, |O| - |O_1|])$, we enumerate all non-empty subsets of $S_{1_{o_1}}$ and $\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}$ and consider $|O_1| \cdot (|O| - |O_1|)$ possible combinations of $IEC(S_{1_{o_1}}, k_1) - IEC(\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}, k_1)$ $(k_1 \in [1, |O_1|])$ and $IEC(V_{\overline{S_1}}, k_2) - IEC(\mathcal{M}_{V_{\overline{S_1}} \times L_{\overline{S_1}}}, k_2)$. Instead of enumerating all $(2^{|S_{o_1}|} - 1) + (2^{|\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}|} - 1) = 38$ non-empty subsets of S_{o_1} and $\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}$ using Equation 4, we only need to consider $(2^{|S_{1_{o_1}}|} - 1) + (2^{|\mathcal{M}_{S_{1_{o_1}} \times S_{1_{\overline{o_1}}}}|} - 1) + 6 + |O_1| \cdot (|O| - |O_1|) = 14$ terms in total using Equation 21. The reduction is significant when $|S_{o_1}|$ or $|\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}|$ is large.

Now we can use Equation 21 to compute $\psi(o_1) = \frac{(1-0)\times 3}{1+1} + \frac{(1-0)\times(-5)}{1+2} + \frac{(1-0)\times 2}{1+3} + \frac{(0-1)\times 3}{2+1} + \frac{(0-1)\times(-5)}{2+2} + \frac{(0-1)\times 2}{2+3} = \frac{11}{60}$.

In Equation 21, $IEC(V_{\overline{S_1}}, k_2) - IEC(\mathcal{M}_{V_{\overline{S_1}} \times L_{\overline{S_1}}}, k_2)$ only refers to the data owners in $O \setminus O_1$. All data owners in O_1 share the same $IEC(V_{\overline{S_1}}, k_2) - IEC(\mathcal{M}_{V_{\overline{S_1}} \times L_{\overline{S_1}}}, k_2)$ value. We can precompute and

reuse $IEC(V_{\overline{S_1}}, k_2) - IEC(\mathcal{M}_{V_{\overline{S_1}} \times L_{\overline{S_1}}}, k_2)$ for $k_2 \in [0, |O| - |O_1|]$ when calculating Shapley values for data owners in O_1.

Generalization. Again, let us generalize the above discussion. Partitioning minimal syntheses into exclusive subsets with vertical decomposition potential enables us to decompose both S_o and $\mathcal{M}_{S_o \times S_{\overline{o}}}$ into smaller coalitions. This, in turn, facilitates the decomposition of $IEC(S_o, k) - IEC(\mathcal{M}_{S_o \times S_{\overline{o}}}, k)$ through the use of IEC within these smaller coalitions.

Theorem 8. *Consider a data assemblage game (O, v) without dummies and the set of minimal syntheses S, where S can be decomposed into n groups of coalitions $S_1, \ldots, S_n$ such that (1) for any $1 \le i < j \le n$, $O_i \cap O_j = \emptyset$ and $\cup_{i=1}^{n} O_i = O$, where $O_i = Owner(S_i)$ $(1 \le i \le n)$; and (2) there exists a unique non-empty subset $\mathcal{N} \subseteq 2^{\{1,\ldots,n\}} \setminus \emptyset$ where $S = \cup_{\{j_1,\ldots,j_l\} \in \mathcal{N}} \mathcal{M}_{S_{j_1} \times \cdots \times S_{j_l}}$. For any S_i $(1 \le i \le n)$, denote by $\mathcal{N}_i = \{X \mid X \in \mathcal{N} \text{ and } i \in X\}$ and $\mathcal{N}_{\overline{i}} = \{X \mid X \in \mathcal{N} \text{ and } i \notin X\}$, respectively, the set of elements in $\mathcal{N}$ that do and do not contain i. Denote by $\mathcal{N}_{i_c} = \{X \setminus \{i\} \mid X \in \mathcal{N}_i\}$ the set of elements in $\mathcal{N}_i$ by excluding i. Then, we have*

$$\begin{aligned} & IEC(S_o, k) - IEC(\mathcal{M}_{S_o \times S_{\overline{o}}}, k) \\ = & \sum_{\substack{k_1 + k_2 = k \\ k_1 \ge 1,\ k_2 \ge 1}} \Big(\big(IEC(S_{i_o}, k_1) - IEC(\mathcal{M}_{S_{i_o} \times S_{i_{\overline{o}}}}, k_1) \big) \cdot \big(IEC(V_{\overline{S_i}}, k_2) - IEC(\mathcal{M}_{V_{\overline{S_i}} \times L_{\overline{S_i}}}, k_2) \big) \Big), \end{aligned} \tag{22}$$

where

$$\begin{aligned} IEC(V_{\overline{S_i}}, k_2) &= IEC(\cup_{\{j_1,\ldots,j_l\} \in \mathcal{N}_{i_c}} \mathcal{M}_{S_{j_1} \times \cdots \times S_{j_l}}, k_2) \\ &= \sum_{\substack{X \subseteq \mathcal{N}_{i_c} \\ |X| \ge 1}} \Big((-1)^{|X|-1} \sum_{\substack{let\ \{t_1,\ldots,t_l\} = \cup_{Z \in X} Z \\ k'_{t_1} + \cdots + k'_{t_l} = k_2 \\ \forall 1 \le j \le l, k'_{t_j} \ge 1}} \prod_{j=1}^{l} IEC(S_{t_j}, k'_{t_j}) \Big), \end{aligned} \tag{23}$$

and similarly,

$$\begin{aligned} IEC(\mathcal{M}_{V_{\overline{S_i}} \times L_{\overline{S_i}}}, k_2) &= IEC(\cup_{\{j_1,\ldots,j_l\} \in \mathcal{M}_{\mathcal{N}_{i_c} \times \mathcal{N}_{\overline{i}}}} \mathcal{M}_{S_{j_1} \times \cdots \times S_{j_l}}, k_2) \\ &= \sum_{\substack{X \subseteq \mathcal{M}_{\mathcal{N}_{i_c} \times \mathcal{N}_{\overline{i}}} \\ |X| \ge 1}} \Big((-1)^{|X|-1} \sum_{\substack{let\ \{t_1,\ldots,t_l\} = \cup_{Z \in X} Z \\ k'_{t_1} + \cdots + k'_{t_l} = k_2 \\ \forall 1 \le j \le l, k'_{t_j} \ge 1}} \prod_{j=1}^{l} IEC(S_{t_j}, k'_{t_j}) \Big). \end{aligned} \tag{24}$$

Proof sketch. Equation 22 can be obtained following Equation 20. Equation 23 is a generalization of Equation 14. Equation 24 can be obtained by computing $IEC(\cup_{\{j_1,\ldots,j_l\} \in \mathcal{M}_{\mathcal{N}_{i_c} \times \mathcal{N}_{\overline{i}}}} \mathcal{M}_{S_{j_1} \times \cdots \times S_{j_l}}, k_2)$ using Equation 23. □

Note that Equations 12 and 19 are two special cases of Equation 23 for any $k_2 \ge 1$. In the hybrid decomposition, when $V_{\overline{S_i}} = \mathcal{M}_{S_1 \times \cdots S_{i-1} \times S_{i+1} \cdots \times S_n}$, $\mathcal{N}_{i_c} = \{\{1, \ldots, i, i+1, \ldots, n\}\}$. We can obtain Equation 12 by plugging $\mathcal{N}_{i_c} = \{\{1, \ldots, i, i+1, \ldots, n\}\}$ into Equation 23. Similarly, when $V_{\overline{S_i}} = \cup_{j=1, j \ne i}^{n} S_j$, $\mathcal{N}_{i_c} = \{\{1\}, \ldots, \{i\}, \{i+1\}, \ldots, \{n\}\}$. We can obtain Equation 19 by plugging $\mathcal{N}_{i_c}$ = $\{\{1\}, \ldots, \{i\}, \{i+1\}, \ldots, \{n\}\}$ into Equation 23.

4.5 Data Assemblage Game Decomposition

The previous discussion illustrates our ideas of decomposing minimal syntheses to speed up the Shapley value computation. Now, we formulate game decomposition, which implements this process.

Definition 9 (Game Decomposition). Given a data assemblage game (O, v) without dummies and the set of minimal syntheses S. A set of data assemblage games $D = \{(O_1, v_1), \ldots, (O_n, v_n)\}$, where every game $(O_i, v_i)(1 \leq i \leq n)$ does not have dummies and the corresponding set of minimal syntheses is S_i, is a **decomposition** of (O, v) if (**owner partitioning**) $O = \cup_{i=1}^{n} O_i$, where $O_i = Owner(S_i)$ $(1 \leq i \leq n)$, and $O_i \cap O_j = \varnothing$ for $1 \leq i < j \leq n$; and (**modularity**) there exists a unique non-empty subset $\mathcal{N} \subseteq 2^{\{1,\ldots,n\}} \setminus \varnothing$ such that $S = \cup_{\{j_1,\ldots,j_t\} \in \mathcal{N}} \mathcal{M}_{S_{j_1} \times \cdots \times S_{j_t}}$. $\mathcal{N}$ is called the **decomposition mapping** (DM in short) from S to $S_1, \ldots, S_n$, denote by $DM(S \rightarrow \{S_1, \ldots, S_n\})$. A decomposition D is **grounded** if every $|O_i| = 1$ $(1 \leq i \leq n)$. A decomposition D is **trivial** if $n = 1$. □

Proposition 10 (Grounded decomposition). *Every data assemblage game without dummies has a unique grounded decomposition.*

A decomposition can be one of the following four cases. If $S = \mathcal{M}_{S_1 \times \cdots \times S_n}$ and $n > 1$, D is a **vertical decomposition**; if $S = S_1 \cup \cdots \cup S_n$ and $n > 1$, D is a **horizontal decomposition**; if D is not horizontal, vertical, or trivial, then it is a **hybrid decomposition**; otherwise, D is trivial. For a grounded decomposition D, if D is vertical or horizontal, then it can help to speed up the Shapley value computation as illustrated in Sections 4.2 and 4.3. Otherwise, it does not help to accelerate the Shapley value computation.

In general, a game may have multiple decompositions. It can be shown that a game cannot have a vertical decomposition and a horizontal decomposition at the same time. If a game has a vertical or horizontal decomposition (not necessary grounded), then it is **vertically** or **horizontally decomposable**, respectively. Otherwise, it is **holistic decomposable**. A game (O, v) can be in only one of the following three mutually exclusive categories: vertically decomposable, horizontally decomposable, and holistic decomposable. A thorough treatment of decomposability of data assemblage games is an interesting problem, but is well beyond the capacity of this paper. It will be addressed in a separate study.

Using Equations 11, 18, and 22 to compute $IEC(S_{o_1}, k) - IEC(\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}, k)$ in Equation 5 needs to consider all possible subsets of S_j $(1 \leq j \leq n)$. When there are many coalitions in S_j, the computation can be still costly. We can recursively decompose a data assemblage game until reaching the grounded decomposition.

For example, consider a set of data owners $O = \{o_1, o_2, o_3, o_4, o_5\}$ producing a target data set in coalition, and the minimal syntheses are $C_1 = o_1o_3$, $C_2 = o_2o_3$, $C_3 = o_1o_4o_5$, $C_4 = o_2o_4o_5$, and $C_5 = o_3o_4o_5$. We can first decompose (O, v) into three sub-games (O_1, v_1), (O_2, v_2), and (O_3, v_3) using the hybrid decomposition, where $O_1 = \{o_1, o_2\}$, $O_2 = \{o_3\}$, and $O_3 = \{o_4, o_5\}$, and the corresponding sets of minimal syntheses are $S_1 = \{o_1, o_2\}$, $S_2 = \{o_3\}$, and $S_3 = \{o_4o_5\}$, respectively. Then, the set of minimal syntheses can be constructed as $\mathcal{M}_{S_1 \times S_2} \cup \mathcal{M}_{S_1 \times S_3} \cup \mathcal{M}_{S_2 \times S_3}$. Then, (O_1, v_1) and (O_3, v_3) can be further decomposed into two sub-games where each sub-game contains only one data owner using the horizontal decomposition and the vertical decomposition, respectively.

To record the result of recursive decomposition, we can build a **decomposition tree** $\mathcal{T}$ [70], where each tree node is a decomposition. To keep our notations simple, we also use symbol D to denote a node in a decomposition tree. For a node $D \in \mathcal{T}$, we record all children nodes as $D.children$, the parent node as $D.parent$, the set of data owners as $D.O$, and the set of minimal syntheses as $D.S$. Besides, we use the symbols $\otimes$, $\oplus$, and $\odot$ to represent the vertical decomposition, the horizontal decomposition, and the hybrid decomposition, respectively. The decomposition tree of (O, v) is as shown in Figure 1.

To compute $\psi(o_1)$ according to Equation 5, we need to compute $IEC(S_{o_1}, k) - IEC(\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}, k)$, which can be obtained using Equation 22. Given that D_1 is holistic decomposable, we have

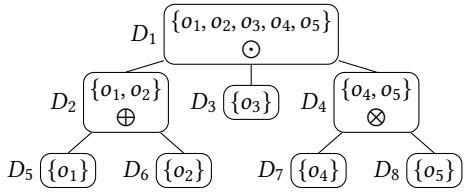


Fig. 1. Example of Decomposition Tree.

$IEC(S_{o_1}, k) - IEC(\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}, k) = IEC(D_1.S_{o_1}, k) - IEC(D_1.\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}, k)$[6] $= \sum_{\substack{k_1+k_2=k \\ k_1 \geq 1,\ k_2 \geq 1}} \Big(\big(IEC(D_1.V_{\overline{D_2.S}}, k_1) - IEC(D_1.\mathcal{M}_{V_{\overline{D_2.S}} \times L_{\overline{D_2.S}}}, k_1) \big) \cdot \big(IEC(D_2.S_{o_1}, k_2) - IEC(D_2.\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}}, k_2) \big) \Big)$.

Notably, $IEC(D_1.V_{\overline{D_2.S}}, k_1) - IEC(D_1.\mathcal{M}_{V_{\overline{D_2.S}} \times L_{\overline{D_2.S}}}, k_1)$ involves only the data owners in $Owner(D_1.S) \setminus Owner(D_2.S)$, that is, $D_1.O \setminus D_2.O$. Given a non-leaf node $D \in \mathcal{T}$ and one of its children D_c, for any data owner $o \in D_c.O$, denote by $IECE(D, D_c, k)$ the part of computation excluding data owners in $D_c.O$ using Equations 11, 18, and 22. Then, for $k \geq 1$,

$$IECE(D, D_c, k) = \begin{cases} IEC(D.V_{\overline{D_c.S}}, k) & \text{if } D \text{ is a vertical decomposition,} \\ -IEC(D.L_{\overline{D_c.S}}, k) & \text{if } D \text{ is a horizontal decomposition,} \\ IEC(D.V_{\overline{D_c.S}}, k) - IEC(D.\mathcal{M}_{V_{\overline{D_c.S}} \times L_{\overline{D_c.S}}}, k) & \text{if } D \text{ is a hybrid decomposition.} \end{cases}$$

Define $IECE(D, D_c, 0) = 0$ when D is a vertical or hybrid decomposition, and $IECE(D, D_c, 0) = 1$ when D is a horizontal decomposition.

Based on the above discussion, we have the following main result on recursive game decomposition and Shapley value computation.

Theorem 11. *Given a decomposition tree $\mathcal{T}$ with the set of data owners O and a data owner $o \in O$. Let D_l be the leaf node in $\mathcal{T}$ that contains o. For any non-leaf node $D \in \mathcal{T}$, let D_c be a child of D, its parent $D.parent$, and a number $k \geq 0$, let*

$$\gamma(D, D_c, k) = \begin{cases} IECE(D, D_c, k) & \text{if } D \text{ is root,} \\ \sum_{\substack{k_1+k_2=k \\ k_1 \geq 0,\ k_2 \geq 0}} \big(IECE(D, D_c, k_1) \cdot \gamma(D.parent, D, k_2) \big) & \text{otherwise.} \end{cases} \tag{25}$$

Then, $\psi(o) = \sum_{k=1}^{|O|-1} \frac{\gamma(D_l.parent, D_l, k)}{k+1}$. □

Algorithm 1 presents the recursive decomposition Shapley value (RDSV) approach using Theorem 11, which consists of three steps. In the first step, *decomposition tree construction* (Line 2), given the set of minimal syntheses, according to Corollary 8 in Bioch [7], a decomposition tree (called modular tree by Bioch [7]) can be generated in $O(|S| \cdot |O'|^5)$ time. Here, the minimal syntheses as input can be obtained by, for example, tracking the provenance of each tuple in a data set using tools like ProvSQL [68]. In the second step, *IEC Computation* (Lines 3 to 8), let $N_\oplus$, $N_\otimes$, and $N_\odot$ be the numbers of vertical, horizontal, and hybrid decompositions in the decomposition tree $\mathcal{T}$, respectively. The complexity of IEC computation is $O((N_\oplus + N_\otimes) m_f {m_o}^2 + N_\odot 2^{m_f} {m_o}^2)$. In the last step, *IECE Computation* (Lines 9 to 16), the complexity of IECE computation step is $O((N_\oplus + N_\otimes) {m_f}^2 {m_o}^2 + N_\odot 2^{m_f} m_f {m_o}^2)$. In summary, the complexity of Algorithm 1 is $O((N_\oplus + N_\otimes) {m_f}^2 {m_o}^2 + N_\odot 2^{m_f} m_f {m_o}^2 + |S| \cdot |O'|^5)$, where $N_\oplus + N_\otimes + N_\odot \leq |O'| - 1$. If there is no hybrid decomposition in $\mathcal{T}$, the complexity of Algorithm 1 is polynomial. Otherwise, it is exponential in the maximum fan-out of the decomposition tree.

[6]For simplicity, we redefine $D_1.\mathcal{M}_{S_{o_1} \times S_{\overline{o_1}}} = \mathcal{M}_{D_1.S_{o_1} \times D_1.S_{\overline{o_1}}}$, with D_1 before the dot operator signifying the constraint of minimal syntheses S_{o_1} under the decomposition D_1. This convention will be consistently followed thereafter.

Algorithm 1: RDSV: computing Shapely value with recursive decomposition.

Input: a set of data owners $O = \{o_1, \ldots, o_n\}$ and the corresponding set of minimal syntheses S
Output: the Shapley value $\psi(o_i)$ $(1 \leq i \leq n)$

remove dummies from O, let O' be the set of non-dummy owners;
generate the decomposition tree $\mathcal{T}$;
bottom-up traverse $\mathcal{T}$ **foreach** $D \in \mathcal{T}$ **do**
  if D *is not a root node* **then**
    if D *is a vertical, horizontal, and hybrid decomposition, respectively* **then**
      compute $IEC(D.S, k)$ for $k \in [1, |D.O|]$ using Equation 12, Equation 19, and Equation 23;
    else // D is a leaf node
      $IEC(D.S, 1) = 1$;
top-down traverse $\mathcal{T}$ **foreach** $D \in \mathcal{T}$ **do**
  if D *is not a leaf node* **then**
    foreach $D_c \in D.children$ **do**
      if D *is a vertical, horizontal, and hybrid decomposition, respectively* **then**
        compute $IECE(D, D_c, k)$ for $k \in [1, |D.O| - |D_c.O|]$ using Equation 12, Equation 19, and Equation 23;
      compute $\gamma(D, D_c, k)$ using Equation 25;
  else // D is a leaf node
    compute $\psi(o)(o \in D.O)$ using Theorem 11;

Last but not least, in a decomposition tree, the data owners in all leaf nodes under the same parent node have the identical Shapley value if the parent node is a vertical or horizontal decomposition. This result speeds up the Shapley value computation further.

5 EMPIRICAL EVALUATION

In this section, we evaluate the performance of our proposed method **RDSV** empirically against the baseline methods.

5.1 Experiment Setup

We compare with three baselines. The **traditional method (TRAD)** computes the exact Shapley value using Equation 1. The **permutation-based sampling method (PERM)** *approximates* Shapley value using the Monte-Carlo sampling method [47]. We use **PERM-1000** and **PERM-2000** to report the results when **PERM** takes 1,000 and 2,000 permutations as the sample, respectively. Although the **independent utility based method (IUSV)** [45] cannot solve data assemblage tasks as simple games in general, we also design specific experiments to compare with **IUSV** under the independent utility assumption. Our experiment settings are similar to [45]. In all baselines, utility is computed directly using minimal syntheses and thus they can take the full advantage of the availability of minimal syntheses.

All methods are implemented using the Rust programming language [50]. The artifacts materials can be found at https://github.com/IDEAL-Lab/shapley-value-simple-game. We use a commodity server with Intel Xeon 2.00GHz E7-4730 CPU and 125GB RAM, running Ubuntu 20.04 LTS to run the experiments. We enforce a timeout[7] of 7,200 seconds in our experiments. A program is terminated if the runtime exceeds the timeout threshold.

[7]We conduct 10 iterations for each experiment. If a timeout or an out-of-memory (OOM) issue occurs during any of these iterations, we record the runtime as a timeout.

Table 1. Default System Parameters

Parameters	Default value
Number of data owners per table k	500
Zipfan parameter α	3.0
Maximum number of copies m	4
Zipfan parameter β (used in UA)	3.0

In our experiments, we use a real data set TPC-H[8], which has 8 tables with 5, 25, 10,000, 150,000, 200,000, 800,000, 1,500,000, and 6,001,215 records, respectively. We assign the records in each table of the data set to the data owners in three steps.

First, an owner can have data from multiple tables. We decide the number of owners assigned to each table in the data set. We consider two settings. In the **EO** (for equal number of owners) setting we randomly choose k owners for each table except for the two smallest tables in the data set. Only 5 owners are assigned to each of those two tables as they have very few records. In the **UO** (for unequal number of owners) setting we choose k owners randomly for the largest table, 5 owners to each of the two smallest tables, and 10 owners to each of the other tables in the data set. In this setting, we explore the effect of splitting records in a large table among multiple owners on Shapley value computation.

When we assign owners to a table, each owner has the equal chance. Moreover, whether a owner has data from a table does not affect the chance that the owner has data from another table.

Second, after we assign owners to each table, we next decide the number of copies of each record in a table. The number of copies of tuples follows the Zipfian distribution [56] with parameter α for each table. Besides, we impose a restriction on the maximum number of copies that a tuple can have, denoted by m.

Last, we assign records to owners in each table. Two different settings are considered. The **EA** (for equal chance assignment) setting assigns records to owners uniformly. Specifically, if a record has l copies and there are k owners assigned to the table ($l \leq k$), each owner has a probability of $\frac{l}{k}$ of obtaining a copy of the record. We impose a constraint that each owner can hold at most one copy of a record. This ensures that the expected number of records held by each owner within a table is the same. The **UA** (for unequal chance assignment) setting assigns records to owners according to the Zipfian distribution with parameter β. Once again, we enforce that each data owner can hold at most one copy of a record. This approach leads to a small number of owners holding the majority of records in a table.

In total, we have four different settings, which can test how data owner distribution may affect Shapley value computation. Specifically, by EO versus UO, we can observe the effect of owner diversity in tables. By EA versus UA, we can observe the effect of dominating owners. Table 1 shows the default parameter values in our experiments.

In the TPC-H data set, a coalition plan executes equi-join queries among all tables and generates a coalition set with 6,001,215 tuples. To compare with **IUSV**, we set the utility of each tuple in the coalition set to 1. Then, the generation of each tuple in the coalition set can be regarded as a data assemblage game. Thus, we have 6,001,215 games in the experiments and **IUSV** can be applied [45].

We evaluate all methods using **Runtime**, the total clock time of computing the Shapley values for all data owners in all simple games. We also assess **Error rate**, the average error in percentage of Shapley value by **PERM**. That is, $error = \frac{1}{t}\sum_{i=1}^{t} \frac{\sum_{u\in\mathcal{U}} |\psi_i(u) - \widehat{\psi_i(u)}|}{\sum_{u\in\mathcal{U}} \psi_i(u)} \times 100\%$, where t is the number

[8]http://www.tpc.org/tpch/, accessed on July 1, 2022.

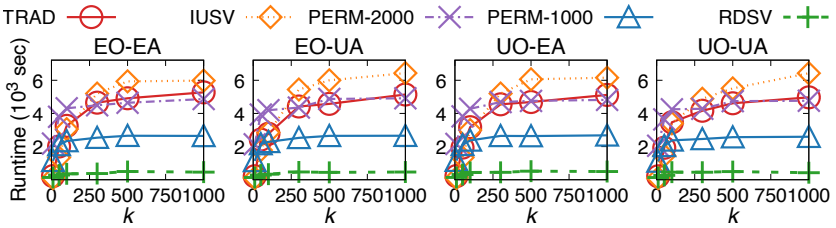


Fig. 2. Scalability on Number of Data Owners k per Table.

Table 2. Error Rate of PERM Under Default Parameters

	EO-EA	EO-UA	UO-EA	UO-UA
PERM-1000	6.64%	6.63%	6.64%	6.64%
PERM-2000	4.70%	4.69%	4.74%	4.70%

of games, $\psi_i(u)$ and $\widehat{\psi_i(u)}$ are the exact Shapley value and the one approximated by **PERM** in the i-th simple game, respectively. The higher error rate, the lower the quality of the approximated Shapley value by **PERM**. For **RDSV** we evaluate some additional metrics. The **vertical rate**, **horizontal rate**, and **hybrid rate** are the percentages of the total number of vertical decompositions, horizontal decompositions, and hybrid decompositions over the sum of the total number of all three decompositions in all games, respectively.

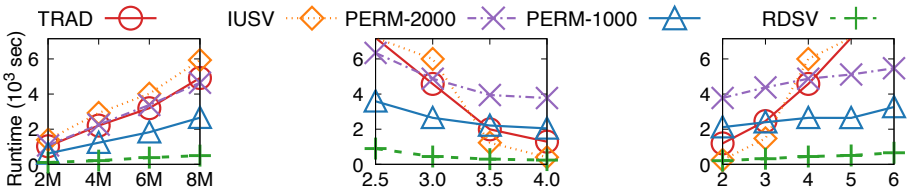
5.2 Scalability

We first consider the scalability with regards to k, the number of data owners per table (Figure 2). Among the three baselines, **PERM** with sample sizes 1,000 and 2,000 outperforms **TRAD** and **IUSV** in runtime. Here we choose the sample sizes 1000 and 2000 since a larger sample size causes **PERM** timeout. The approximation quality of **PERM** is low with these two sample sizes, with the error rate ranging from 4.69% to 6.64% under the default system parameters as shown in Table 2. Compared with **PERM**, **RDSV** outperforms the baselines by an order of magnitude and can compute the exact Shapley value in all cases.

In all the four settings, when the number of data owners per table k increases, more data owners hold some records in more tables. Thus, the number of minimal syntheses and the number of minimal synthesis owners in a game increase. The average fan-out and the number of decompositions in the decomposition tree of a game also increase. Those factors cause the increase of runtime in all methods.

Next, we test the scalability with regards to data set size, that is, the total number of records in a data set. We generate TPC-H data set with size 2 million (2M), 4 million (4M), 6 million (6M), and 8 million (8M) using the TPC-H benchmark kit[9]. Then, we compare the runtime of the methods under the default parameters in all the four settings. As shown in Figure 3a, the runtime of all methods increases linearly with respect to the data set size in the EO-UA setting. We also observe the similar trends in the other three settings and omit the details here due to limited space. In all the four settings, **RDSV** consistently outperforms all baselines. Notably, as the data set size increases, our model's performance improvement becomes even more pronounced. This demonstrates the scalability of **RDSV** with respect to the data set size.

[9]https://github.com/gregrahn/tpch-kit, accessed on May 20, 2023.



(a) Data Set Size (b) Zipfian Parameter α (c) Max # Copies m

Fig. 3. Effect of Data Set Size, Zipfian Parameter α, and Max # Copies m in EO-UA.

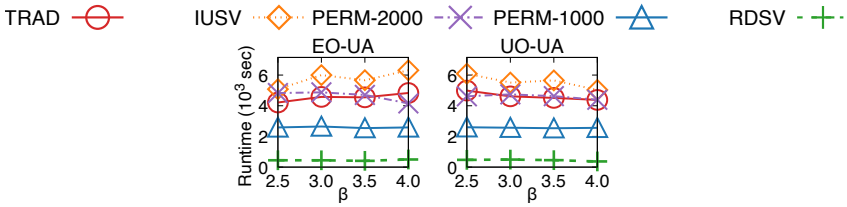


Fig. 4. Effect of Zipfian Parameter β.

5.3 Effect of Data Assignment Distribution

We first test the runtime with respect to the Zipfian parameter α. A larger α results in less copies of each record and less records assigned to a data owner, and leads to a decrease in the number of minimal syntheses and the number of minimal synthesis owners in a game. Such decreases lead to a significant reduction in the number of decompositions required. Therefore, the runtime of **RDSV** reduces when α increases as shown in Figure 3b[10]. Although the runtime of the baselines also decreases substantially when α increases, **RDSV** is still at least one order of magnitude faster than the baselines. The advantage of **RDSV** is clear. We observe similar trends in all the four settings. Limited by space, we only show the setting EO-UA here.

Next, we test the runtime with respect to the maximum number of copies m. A larger m means that more records are held by each data owner, resulting in more minimal syntheses and minimal synthesis owners in a game. This leads to the increase of the runtime of all methods, including **RDSV**, as shown in Figure 3c. Still, **RDSV** is at least one order of magnitude faster than the baselines. We observe similar trends in all the four settings. Limited by space, we only show the setting EO-UA here.

In the settings of UA, the Zipfian distribution with parameter β controls how records are distributed among owners in a biased manner. Figure 4 shows that in both the settings EO-UA and UO-UA, **RDSV** and the baselines are insensitive to β. This is because each owner can have at most one copy of a record, and both the number of minimal syntheses and the number of minimal synthesis owners are primarily determined by the number of copies assigned to each record, that is, by the Zipfian parameter α and the maximum number of copies m. Again, **RDSV** is orders of magnitude faster than the baselines.

[10]In this paper, a method's line extending past a plot's boundary signifies its runtime exceeding the timeout for the respective x-axis parameter. For instance, **TRAD**'s runtime exceeds the timeout at $\alpha = 2.5$.

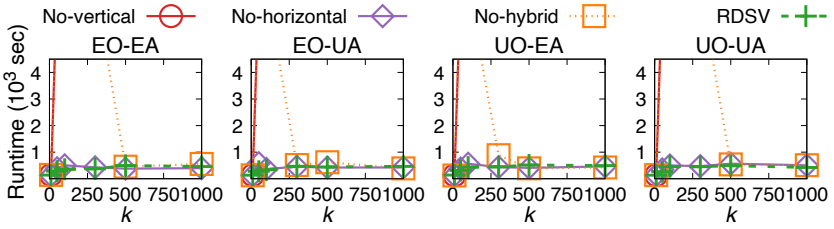


Fig. 5. Performance of Ablations w.r.t. Number of Data Owners k per Table.

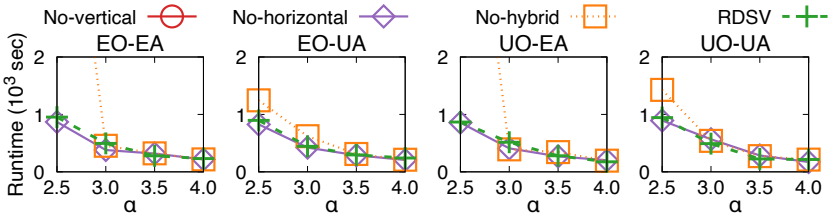


Fig. 6. Performance of Ablations w.r.t. Zipf Parameter α.

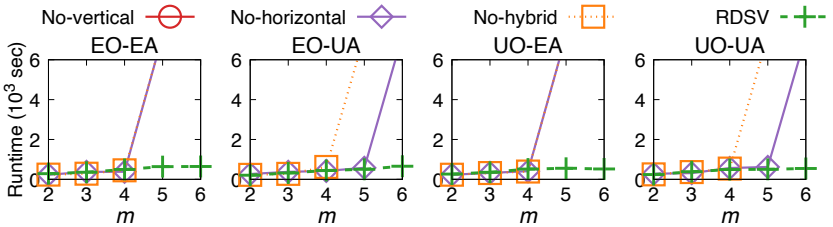


Fig. 7. Performance of Ablations w.r.t. Max # Copies m.

5.4 Ablation Study

RDSV employs three types of decompositions. We investigate how those decompositions affect the efficiency of the method by ablating one type of decomposition at a time. For example, "No-vertical" is the version of **RDSV** that does not conduct vertical decomposition. Figures 5, 6, and 7 report the performance of the ablations and our complete method with respect to various parameters.

Figure 5 shows the scalability of the ablations with respect to k, the number of data owners per table. No-vertical exceeds timeout threshold in all settings except for the cases when $k = 10$. When the data assemblage task is an equi-join query, the root of the decomposition tree is often a vertical decomposition. Thus, without the vertical decomposition, **RDSV** is unable to decompose almost all games in our experiments. Without a game decomposition, **RDSV** degenerates to Equation 5, where the computational cost is high when the number of minimal syntheses is large. When $k = 10$, the maximum number of minimal synthesis owners cannot exceed 10.

For the same reason, No-vertical exceeds the timeout threshold in all settings in Figures 6 and 7. No-vertical is barely able to compute the Shapley value in a brute force manner. This experiment shows that vertical decompositions are essential for data assemblage tasks with equi-join queries.

No-horizontal has very little performance loss with respect to the number of data owners k per table (Figure 5) and the Zipfian parameter α (Figure 6). Since we impose a constraint on the maximum number of copies per tuple (4 by default), the number of data owners involved in a

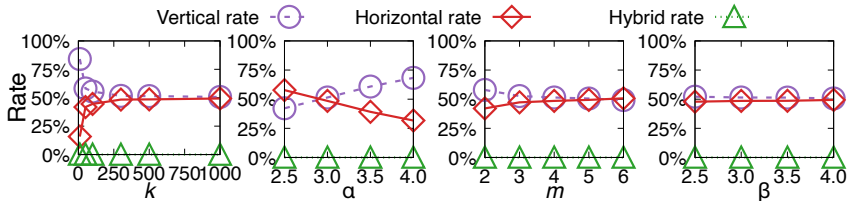


Fig. 8. Vertical Rate, Horizontal Rate, and Hybrid Rate w.r.t. Parameters in EO-UA.

horizontal decomposition is also constrained by the same parameter. When the number of owners is small, not conducting horizontal decompositions does not hurt the performance much, since computing the Shapley value by enumerating all coalitions is fast anyway. However, if we increase the maximum number of copies (Figure 7), the gap between No-horizontal and the complete method increases dramatically.

To investigate the role of hybrid decompositions in **RDSV**, Figure 8 plots the vertical rate, horizontal rate, and hybrid rate in the setting EO-UA with respect to parameters k, α, m, and β varies. The trends are similar in other settings and are omitted limited by space. Interestingly, the hybrid rate is quite low compared to the vertical rate and the horizontal rate in all settings. However, this does not mean that hybrid decompositions are not useful in **RDSV**. **RDSV** may exceed the timeout threshold when the number of data owners holding records is high and data is dense or when some records have a large number of copies, such as the cases when $k = 50$ in all the four settings of Figure 5 and $m = 6$ in all the four settings of Figure 7. In all those cases, there are only a small number of games with a large number of minimal syntheses that cannot be well decomposed without the hybrid decomposition. This result shows that, although the hybrid rate is low, the hybrid decomposition is powerful once it can be applied.

The ablations are not sensitive and do not lose much performance with respect to β for the same reason as discussed at the end of Section 5.3. Limited by space, we omit the details here.

6 A CASE STUDY

To test the performance of **RDSV** in real-world applications, we conduct a case study on real-world data assemblage tasks.

We use the European Soccer Database (ESD) from Kaggle[11], a data set that inherently contains data owner information. ESD is a meticulously compiled data set sourced from web crawling using different APIs, encompassing a wealth of information spanning players, teams, and match data from 2008 to 2016. ESD comprises a total of 7 tables.

As an example used in our case study, we find out for each team whether the team won at least one game in the 2016 season[12]. Accordingly, for each team that won at least once, there is a data assemblage game to assemble at least one record of the team's wins. Those tasks involve four tables in ESD: `Country`, `League`, `Team`, and `Match`, with 11, 11, 299, and 25,979 records, respectively. The task encompasses a sequence of table joins. A SQL representation of the task can be found in the full version technical report [46].

In table `Match`, there are 3,326 records related to the 2016 season. In tables `Team` and `Match`, the information about API-ids is provided, which indicates where the records are collected. Naturally, each API can be regarded as a data owner. In total, there are 3,689 unique APIs providing the data

[11]http://www.kaggle.com/hugomathien/soccer, accessed on Oct 1, 2023.
[12]https://www.kaggle.com/code/abdelrhmanragab/european-soccer, accessed on Oct 1, 2023.

Table 3. Runtime (unit: sec) of All Methods in the Case Study

TRAD	IUSV	PERM-2000	PERM-1000	RDSV
-	-	6.42	4.07	2.45

related to this query. Note that one record may be provided by more than one API. Accordingly, the total number of data owners in this case is 3,689.

There are 188 teams that won in the season. 2,843 data owners contribute to 2,471 minimal syntheses in those 188 data assemblage games. We compute the Shapley value for each data owner in each of those games. It is worth noting that in this setting, **IUSV** [45] can work and thus can serve as a baseline.

The total runtime of each method over all those 188 games is shown in Table 3[13]. Among the three baselines, both **TRAD** and **IUSV** cannot complete within 2 hours, and **PERM** significantly outperforms **TRAD** and **IUSV** by completing in just a few seconds. However, **PERM** exhibits low approximation quality with sample sizes 1,000 and 2,000, resulting in error rates of 8.13% and 5.80%, respectively. In contrast, **RDSV** not only outperforms all baselines in runtime but also computes the exact Shapley value. This demonstrates the efficiency and effectiveness of **RDSV** in real-world applications. Remarkably, all 188 games can be decomposed through a combination of only vertical decomposition and horizontal decomposition, with the vertical rate and the horizontal rate at 74.08% and 25.92%, respectively.

7 CONCLUSIONS AND DISCUSSIONS

In this paper, we tackle the problem of Shapley value computation in data assemblage tasks as cooperative games and use pivotal probabilities to count beneficiaries. By game decomposition, we significantly improve the efficiency and retain the exactness.

While our discussion here focuses on simple games, it offers insights and methods that can inspire ideas for more general games with non-binary utility functions. As future work, we may convert a non-binary utility function (e.g., a k-nary function) into a composition of multiple binary utility functions. Subsequently, our proposed method can be applied to each of these binary utility functions. Our proposed method leverages a decomposition algorithm [7, 8] that is applicable to general boolean functions, encompassing both monotone and non-monotone boolean functions. As another interesting direction, we may adopt a similar game decomposition approach to expedite the computation of Shapley values in scenarios where the utility function takes a non-monotone boolean form.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their detailed comments which helped to improve the paper during the revision process. Xuan Luo and Cheng Xu are supported in part of an NSERC Discovery Grant. Jian Pei is supported in part by a startup grant and a Beyond the Horizon grant by Duke University. Wenjie Zhang is supported by ARC Future Fellowship (FT210100303) and ARC Discovery Project (DP230101445). Jianliang Xu is supported by Hong Kong RGC CRF Project (C2004-21GF). All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

[13]In this table, a "-" indicates a timeout

REFERENCES

[1] Alessandro Acquisti, Curtis Taylor, and Liad Wagman. 2016. The Economics of Privacy. *Journal of Economic Literature* 54, 2 (June 2016), 442–92. https://doi.org/10.1257/jel.54.2.442

[2] Anish Agarwal, Munther A. Dahleh, and Tuhin Sarkar. 2019. A Marketplace for Data: An Algorithmic Solution. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, Anna Karlin, Nicole Immorlica, and Ramesh Johari (Eds.). ACM, 701–726. https://doi.org/10.1145/3328526.3329589

[3] Charu C. Aggarwal. 2016. *Recommender Systems: The Textbook* (1st ed.). Springer Publishing Company, Incorporated.

[4] Charu C. Aggarwal and Philip S. Yu. 2008. Privacy-Preserving Data Mining: A Survey. In *Handbook of Database Security: Applications and Trends*, Michael Gertz and Sushil Jajodia (Eds.). Springer US, Boston, MA, 431–460. https://doi.org/10.1007/978-0-387-48533-1_18

[5] William Aiello, Yuval Ishai, and Omer Reingold. 2001. Priced Oblivious Transfer: How to Sell Digital Goods. In *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding (Lecture Notes in Computer Science, Vol. 2045)*. Springer, 119–135. https://doi.org/10.1007/3-540-44987-6_8

[6] Magdalena Balazinska, Bill Howe, and Dan Suciu. 2011. Data Markets in the Cloud: An Opportunity for the Database Community. *Proc. VLDB Endow.* 4, 12 (2011), 1482–1485. http://www.vldb.org/pvldb/vol4/p1482-balazinska.pdf

[7] Jan C. Bioch. 2002. Modular Decomposition of Boolean Functions. https://ssrn.com/abstract=370984.

[8] Jan C. Bioch. 2005. The complexity of modular decomposition of Boolean functions. *Discret. Appl. Math.* 149, 1-3 (2005), 1–13. https://doi.org/10.1016/j.dam.2003.12.010

[9] Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. 2009. Set Partitioning via Inclusion-Exclusion. *SIAM J. Comput.* 39, 2 (2009), 546–563. https://doi.org/10.1137/070683933 arXiv:https://doi.org/10.1137/070683933

[10] Jens Bleiholder, Sascha Szott, Melanie Herschel, and Felix Naumann. 2010. Complement union for data integration. In *Workshops Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, March 1-6, 2010, Long Beach, California, USA*. IEEE Computer Society, 183–186. https://doi.org/10.1109/ICDEW.2010.5452760

[11] George Boole. 1854. *An investigation of the laws of thought: on which are founded the mathematical theories of logic and probabilities*. Vol. 2. Walton and Maberly.

[12] Frank M. Brown. 1990. *Boolean reasoning - the logic of boolean equations*. Kluwer.

[13] Satya R. Chakravarty, Manipushpak Mitra, and Palash Sarkar. 2014. *A Course on Cooperative Game Theory*. Cambridge University Press. https://doi.org/10.1017/CBO9781107415997

[14] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. 2011. *Computational Aspects of Cooperative Game Theory (Synthesis Lectures on Artificial Inetlligence and Machine Learning)* (1st ed.). Morgan & Claypool Publishers.

[15] Lingjiao Chen, Paraschos Koutris, and Arun Kumar. 2019. Towards Model-based Pricing for Machine Learning in a Data Marketplace. In *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, Peter A. Boncz, Stefan Manegold, Anastasia Ailamaki, Amol Deshpande, and Tim Kraska (Eds.). ACM, 1535–1552. https://doi.org/10.1145/3299869.3300078

[16] Sara Cohen, Itzhak Fadida, Yaron Kanza, Benny Kimelfeld, and Yehoshua Sagiv. 2006. Full Disjunctions: Polynomial-Delay Iterators in Action. In *Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea, September 12-15, 2006*, Umeshwar Dayal, Kyu-Young Whang, David B. Lomet, Gustavo Alonso, Guy M. Lohman, Martin L. Kersten, Sang Kyun Cha, and Young-Kuk Kim (Eds.). ACM, 739–750. http://dl.acm.org/citation.cfm?id=1164191

[17] Zicun Cong, Xuan Luo, Jian Pei, Feida Zhu, and Yong Zhang. 2022. Data pricing in machine learning pipelines. *Knowl. Inf. Syst.* 64, 6 (2022), 1417–1455. https://doi.org/10.1007/s10115-022-01679-4

[18] R.D. Cook and Sanford Weisberg. 1980. Characterizations of an Empirical Influence Function for Detecting Influential Cases in Regression. *Technometrics* 22, 4 (1980), 495–508. https://doi.org/10.1080/00401706.1980.10486199 arXiv:https://www.tandfonline.com/doi/pdf/10.1080/00401706.1980.10486199

[19] Yves Crama and Peter L. Hammer. 2011. *Boolean Functions - Theory, Algorithms, and Applications*. Encyclopedia of mathematics and its applications, Vol. 142. Cambridge University Press. http://www.cambridge.org/gb/knowledge/isbn/item6222210/?site_locale=en_GB

[20] Nilesh N. Dalvi and Dan Suciu. 2007. Efficient query evaluation on probabilistic databases. *VLDB J.* 16, 4 (2007), 523–544. https://doi.org/10.1007/s00778-006-0004-3

[21] David Dao, Dan Alistarh, Claudiu Musat, and Ce Zhang. 2018. DataBright: Towards a Global Exchange for Decentralized Data Ownership and Trusted Computation. *CoRR* abs/1802.04780 (2018). arXiv:1802.04780 http://arxiv.org/abs/1802.04780

[22] Xiaotie Deng and Christos H. Papadimitriou. 1994. On the Complexity of Cooperative Solution Concepts. *Mathematics of Operations Research* 19, 2 (1994), 257–266. http://www.jstor.org/stable/3690220

[23] Daniel Deutch, Nave Frost, Benny Kimelfeld, and Mikaël Monet. 2021. Computing the Shapley Value of Facts in Query Answering. *CoRR* abs/2112.08874 (2021). arXiv:2112.08874 https://arxiv.org/abs/2112.08874

[24] Xin Luna Dong and Divesh Srivastava. 2015. *Big Data Integration.* Morgan & Claypool Publishers. https://doi.org/10.2200/S00578ED1V01Y201404DTM040

[25] Ulrich Faigle and Walter Kern. 1992. The Shapley value for cooperative games under precedence constraints. *International Journal of Game Theory* 21 (1992), 249–266.

[26] Dan S. Felsenthal and Moshé Machover. 1996. Alternative Forms of the Shapley Value and the Shapley-Shubik Index. *Public Choice* 87, 3/4 (1996), 315–318. http://www.jstor.org/stable/30027233

[27] Raul Castro Fernandez, Pranav Subramaniam, and Michael J. Franklin. 2020. Data Market Platforms: Trading Data Assets to Solve Data Problems. *Proc. VLDB Endow.* 13, 11 (2020), 1933–1947. http://www.vldb.org/pvldb/vol13/p1933-fernandez.pdf

[28] Lisa K. Fleischer and Yu-Han Lyu. 2012. Approximately Optimal Auctions for Selling Privacy When Costs Are Correlated with Data. In *Proceedings of the 13th ACM Conference on Electronic Commerce* (Valencia, Spain) *(EC'12)*. Association for Computing Machinery, New York, NY, USA, 568–585. https://doi.org/10.1145/2229012.2229054

[29] Amirata Ghorbani and James Y. Zou. 2019. Data Shapley: Equitable Valuation of Data for Machine Learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 2242–2251. http://proceedings.mlr.press/v97/ghorbani19c.html

[30] Arpita Ghosh, Katrina Ligett, Aaron Roth, and Grant Schoenebeck. 2014. Buying Private Data without Verification. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation* (Palo Alto, California, USA) *(EC'14)*. Association for Computing Machinery, New York, NY, USA, 931–948. https://doi.org/10.1145/2600057.2602902

[31] Donald B Gillies. 1959. Solutions to general non-zero-sum games. *Contributions to the Theory of Games* 4, 40 (1959), 47–85.

[32] Andrew V. Goldberg, Jason D. Hartline, and Andrew Wright. 2001. Competitive Auctions and Digital Goods. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms* (Washington, D.C., USA) *(SODA'01)*. Society for Industrial and Applied Mathematics, USA, 735–744.

[33] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning.* MIT press.

[34] Miguel A Hernán and James M Robins. 2010. Causal inference.

[35] Nick Hynes, David Dao, David Yan, Raymond Cheng, and Dawn Song. 2018. A Demonstration of Sterling: A Privacy-Preserving Data Marketplace. *Proc. VLDB Endow.* 11, 12 (Aug. 2018), 2086–2089. https://doi.org/10.14778/3229863.3236266

[36] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gürel, Bo Li, Ce Zhang, Costas J. Spanos, and Dawn Song. 2019. Efficient Task-Specific Data Valuation for Nearest Neighbor Algorithms. *Proc. VLDB Endow.* 12, 11 (2019), 1610–1623. https://doi.org/10.14778/3342263.3342637

[37] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J. Spanos. 2019. Towards Efficient Data Valuation Based on the Shapley Value. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan (Proceedings of Machine Learning Research, Vol. 89)*, Kamalika Chaudhuri and Masashi Sugiyama (Eds.). PMLR, 1167–1176. http://proceedings.mlr.press/v89/jia19a.html

[38] Michael I Jordan and Tom M Mitchell. 2015. Machine learning: Trends, perspectives, and prospects. *Science* 349, 6245 (2015), 255–260.

[39] Javen Kennedy, Pranav Subramaniam, Sainyam Galhotra, and Raul Castro Fernandez. 2022. Revisiting Online Data Markets in 2022: A Seller and Buyer Perspective. *SIGMOD Rec.* 51, 3 (nov 2022), 30–37. https://doi.org/10.1145/3572751.3572757

[40] Aamod Khatiwada, Roee Shraga, Wolfgang Gatterbauer, and Renée J. Miller. 2022. Integrating Data Lake Tables. *Proc. VLDB Endow.* 16, 4 (2022), 932–945. https://www.vldb.org/pvldb/vol16/p932-khatiwada.pdf

[41] Jon Kleinberg, Christos H Papadimitriou, and Prabhakar Raghavan. 2001. On the value of private information. In *Theoretical Aspects Of Rationality And Knowledge: Proceedings of the 8 th conference on Theoretical aspects of rationality and knowledge*, Vol. 8. Citeseer, 249–257.

[42] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[43] Chao Li, Daniel Yang Li, Gerome Miklau, and Dan Suciu. 2015. A Theory of Pricing Private Data. *ACM Trans. Database Syst.* 39, 4, Article 34 (Dec. 2015), 28 pages. https://doi.org/10.1145/2691190.2691191

[44] Ester Livshits, Leopoldo E. Bertossi, Benny Kimelfeld, and Moshe Sebag. 2020. The Shapley Value of Tuples in Query Answering. In *23rd International Conference on Database Theory, ICDT 2020, March 30-April 2, 2020, Copenhagen, Denmark (LIPIcs, Vol. 155)*, Carsten Lutz and Jean Christoph Jung (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 20:1–20:19. https://doi.org/10.4230/LIPIcs.ICDT.2020.20

[45] Xuan Luo, Jian Pei, Zicun Cong, and Cheng Xu. 2022. On Shapley Value in Data Assemblage Under Independent Utility. *Proc. VLDB Endow.* 15, 11 (2022), 2761–2773. https://www.vldb.org/pvldb/vol15/p2761-luo.pdf

[46] Xuan Luo, Jian Pei, Cheng Xu, Wenjie Zhang, and Jianliang Xu. 2024. Fast Shapley Value Computation in Data Assemblage Tasks as Cooperative Simple Games (Technical Report). https://github.com/IDEAL-Lab/shapley-value-simple-game/blob/main/technical_report.pdf
[47] Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. 2013. Bounding the Estimation Error of Sampling-based Shapley Value Approximation With/Without Stratifying. *CoRR* abs/1306.4265 (2013). arXiv:1306.4265 http://arxiv.org/abs/1306.4265
[48] Irwin Mann and Lloyd S Shapley. 1960. *Values of large games, IV: Evaluating the electoral college by Montecarlo techniques*. Rand Corporation.
[49] Irwin Mann and Lloyd S Shapley. 1964. The a priori voting strength of the electoral college. *Game theory and related approaches to social behavior* (1964), 151–164.
[50] Nicholas D Matsakis and Felix S Klock II. 2014. The rust language. In *ACM SIGAda Ada Letters*, Vol. 34. ACM, 103–104.
[51] Alexandra Meliou, Wolfgang Gatterbauer, Katherine F. Moore, and Dan Suciu. 2010. The Complexity of Causality and Responsibility for Query Answers and non-Answers. *Proc. VLDB Endow.* 4, 1 (2010), 34–45. https://doi.org/10.14778/1880172.1880176
[52] Alexandra Meliou, Sudeepa Roy, and Dan Suciu. 2014. Causality and Explanations in Databases. *Proc. VLDB Endow.* 7, 13 (2014), 1715–1716. https://doi.org/10.14778/2733004.2733070
[53] Renée J. Miller. 2018. Open Data Integration. *Proc. VLDB Endow.* 11, 12 (2018), 2130–2139. https://doi.org/10.14778/3229863.3240491
[54] Xavier Molinero, Fabián Riquelme, and Maria J. Serna. 2015. Forms of representation for simple games: Sizes, conversions and equivalences. *Math. Soc. Sci.* 76 (2015), 87–102. https://doi.org/10.1016/j.mathsocsci.2015.04.008
[55] Alexander Muschalle, Florian Stahl, Alexander Löser, and Gottfried Vossen. 2012. Pricing approaches for data markets. In *International workshop on business intelligence for the real-time enterprise*. Springer, 129–144.
[56] Mark EJ Newman. 2005. Power laws, Pareto distributions and Zipf's law. *Contemporary physics* 46, 5 (2005), 323–351.
[57] Kobbi Nissim, Salil Vadhan, and David Xiao. 2014. Redrawing the Boundaries on Purchasing Data from Privacy-Sensitive Individuals. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science* (Princeton, New Jersey, USA) *(ITCS'14)*. Association for Computing Machinery, New York, NY, USA, 411–422. https://doi.org/10.1145/2554797.2554835
[58] Chaoyue Niu, Zhenzhe Zheng, Fan Wu, Shaojie Tang, Xiaofeng Gao, and Guihai Chen. 2018. Unlocking the Value of Privacy: Trading Aggregate Statistics over Private Correlated Data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD'18)*. Association for Computing Machinery, New York, NY, USA, 2031–2040. https://doi.org/10.1145/3219819.3220013
[59] K. Pantelis and L. Aija. 2013. Understanding the value of (big) data. In *2013 IEEE International Conference on Big Data*. 38–42.
[60] Judea Pearl. 2009. Causal inference in statistics: An overview. *Statistics surveys* 3 (2009), 96–146.
[61] Judea Pearl. 2010. Causal inference. *Causality: objectives and assessment* (2010), 39–58.
[62] J. Pei. 2021. A Survey on Data Pricing: from Economics to Data Science. *IEEE Transactions on Knowledge & Data Engineering* 01 (dec 2021), 1–1. https://doi.org/10.1109/TKDE.2020.3045927
[63] Foster Provost and Tom Fawcett. 2013. Data science and its relationship to big data and data-driven decision making. *Big data* 1, 1 (2013), 51–59.
[64] Anand Rajaraman and Jeffrey D. Ullman. 1996. Integrating Information by Outerjoins and Full Disjunctions. In *Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 3-5, 1996, Montreal, Canada*, Richard Hull (Ed.). ACM Press, 238–248. https://doi.org/10.1145/237661.237717
[65] Paul Resnick and Hal R Varian. 1997. Recommender systems. *Commun. ACM* 40, 3 (1997), 56–58.
[66] Babak Salimi, Leopoldo E. Bertossi, Dan Suciu, and Guy Van den Broeck. 2016. Quantifying Causal Effects on Query Answering in Databases. In *8th USENIX Workshop on the Theory and Practice of Provenance, TaPP 2016, Washington, D.C., USA, June 8-9, 2016*, Sarah Cohen Boulakia (Ed.). USENIX Association. https://www.usenix.org/conference/tapp16/workshop-program/presentation/salimi
[67] Fabian Schomm, Florian Stahl, and Gottfried Vossen. 2013. Marketplaces for data: an initial survey. *ACM SIGMOD Record* 42, 1 (2013), 15–26.
[68] Pierre Senellart, Louis Jachiet, Silviu Maniu, and Yann Ramusat. 2018. ProvSQL: Provenance and Probability Management in PostgreSQL. *Proc. VLDB Endow.* 11, 12 (2018), 2034–2037. https://doi.org/10.14778/3229863.3236253
[69] Claude E. Shannon. 1949. The synthesis of two-terminal switching circuits. *Bell Syst. Tech. J.* 28, 1 (1949), 59–98. https://doi.org/10.1002/j.1538-7305.1949.tb03624.x
[70] LS Shapley. 1967. On committees. In *New Methods of Thought and Procedure: Contributions to the Symposium on Methodologies*. Springer, 246–270.
[71] Lloyd S. Shapley. 1952. *A Value for n-Person Games*. Technical Report P-295. RAND Corporation, Santa Monica, CA.

[72] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction APIs. In *Proceedings of the 25th USENIX Conference on Security Symposium* (Austin, TX, USA) *(SEC'16)*. USENIX Association, USA, 601–618.

Received July 2023; revised October 2023; accepted November 2023