FQora: Towards Fair Quality-based Data Market for Machine Learning

Songyue Guo
HKUST(GZ) & HIT(SZ)
sguo349@connect.hkust-gz.edu.cn

Maocheng Li
HKUST
csmichael@cse.ust.hk

Zhao Chen
HKUST
chenzhao@ust.hk

Chen Cao
HKUST
cao@ust.hk

Lei Chen
HKUST(GZ) & HKUST
leichen@cse.ust.hk

ABSTRACT

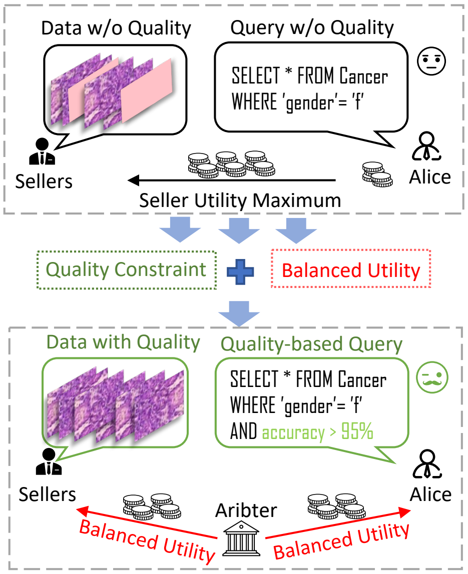
High-quality training data is critical for the development of highly accurate and robust machine learning models, particularly as foundational models emerge. Nevertheless, acquiring such data may incur significant expenses and consume a substantial amount of time, resulting in an increasing fascination with data marketplaces. However, prior research on data marketplaces encounters two key challenges. Firstly, the majority of marketplaces lack the ability for buyers to gain control over the quality of the data they intend to purchase. Secondly, the basic optimization objective of these marketplaces is to maximize the revenues of sellers, potentially resulting in a decrease in buyer purchases and consequently impacting the buyer's future profitability. To address these challenges, we propose a novel; fair and quality-based data marketplace **FQora**, which not only enables buyers to add quality constraints to queries but also achieves an overall utility balance in the marketplace. Extensive experiments on four datasets provide empirical support for our theoretical analysis and confirm the superior performance of our proposed FQora.

PVLDB Reference Format:
Songyue Guo, Maocheng Li, Zhao Chen, Chen Cao, and Lei Chen. FQora: Towards Fair Quality-based Data Market for Machine Learning. PVLDB, 14(1): XXX-XXX, 2020.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/Songyue-Guo/FQora.

1 Introduction

Machine learning has made significant progress in recent years, especially in the fields of computer vision [14, 20] and natural language processing [36] with the rise of foundational models like GPT-4 [30], Llama3 [28] etc. Training a model that works well usually requires a lot of training data. However, obtaining numerous data under GDPR [1], PIPL [2], and other protection criteria takes a lot of work. This prompts the birth of the data marketplace [4].



```
SELECT * FROM Cancer
WHERE 'gender'= 'f'
```

```
SELECT * FROM Cancer
WHERE 'gender'= 'f'
AND accuracy > 95%
```

Figure 1: Comparison between traditional data market (on the top) and a fair and quality-based data market (on the bottom) on query constraint and utility allocation.

Several firms, such as Microsoft Azure Marketplace [3], QLik Data Marketplace [32], WorldQuant [37], and Dawex [9], have created straightforward systems for trading data. Furthermore, numerous contemporary strategies have been employed in the field of data pricing, encompassing economic-based pricing [12, 39, 40], query-based pricing [7, 10, 18], and model-based pricing [8]. Previous research on query-based pricing primarily concentrated on the view-based approach, wherein clients are charged depending on the particular subset of data or query they have requested. In model-based pricing, the transaction asset is represented by the model, and its value is determined by evaluating the model's performance.

Motivation Example. We next provide a motivational example in Fig. 1 to show the need for quality requirements of buyers and a fair data market. For simplicity, we still refer to the SQL format to represent queries of buyers in the following of the paper.

Example 1.1. Let's consider Alice, a data scientist who aims to develop a cross-modal pathology foundation model focused on female-related issues. The purpose of this model is to assist clinicians in efficiently diagnosing illnesses in female patients. The saying "garbage in, garbage out" remains valid in the field of foundation model training, highlighting the crucial significance of data

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

quality in determining the overall effectiveness of the models. However, in the traditional data market, Alice can only request a query like Q_1 = *SELECT * FROM Cancer WHERE 'gender'= 'f'*, which is unable to haphazardly choose the data due to the possibility that it may be counterfeit, irrelevant, or inaccurate. Insufficient assurance of data quality can lead to the trained model producing absurdly misdiagnosed findings while assisting doctors in diagnosis. Hence, it is imperative to have a precise quality criterion for Alice's candidate dataset. In order to meet her request, she can submit a query with a quality constraint such as Q_2 = *SELECT * FROM Cancer WHERE 'gender'= 'f' AND accuracy > 95%*. Therefore, Alice possesses the capability to fulfill the requirements of high data quality. Furthermore, Alice places significant importance on price as well. Nevertheless, prior data markets typically take the seller utility maximum problem as the optimization objective. Assume the expectation value of the foundation model is $4, and the seller's price for query Q_1 is $3. Thus, the seller's utility is $3. Then, Alice's utility is only $1 in the traditional unfair data market. However, Alice doesn't want her input/output ratio to be below 1 for a long time. Thus, a price of $2 for Q_2 from the arbiter is considered as an optimal price of the fair data market for balanced utility allocation.

Gaps and Challenges. In a nutshell, the conventional data market faces two significant problems: firstly, buyer-initiated queries sometimes lack explicit specifications about data quality, and secondly, the original optimization objective of the market is to maximize seller profits. This highlights the importance of developing a data market infrastructure that not only caters to queries with specific data quality criteria but also emphasizes fair utility for all participants. Noting that fairness is also mentioned in [31], fairness in previous work typically focuses on the fair allocation between sellers, not the whole data market. Though efforts have been made to develop different variants of data markets [7, 8, 10, 32], how can we design a data market that can deal with a quality-constrained query and make a fair utility distribution between participants? We summarize the gaps and challenges as follows.

- *Handling quality-constrained queries.* In previous works, only query-based pricing can filter data and complete pricing according to user requirements, but query pricing only supports simple SQL statements and does not support queries with quality constraints. Moreover, the query-based approach is mostly based on evaluating a randomly generated support set, and the generation of the support set usually involves replacing a certain attribute of a single piece of data, which hardly leads to a change in the quality of the dataset. Hence, it is time-consuming to explicitly incorporate quality as an attribute into the support set and adhere to the prior structure to finalize the query pricing. Thus, the challenge is: *How to design a framework to handle queries with quality constraints*?
- *Ensuring fair utility distribution among participants.* The utility balanced problem of a market is a multi-objective optimization (MOO) problem, for which there are many solutions to obtain the final Pareto-optimal solution, but the Pareto-optimal solution set obtained by previous methods is usually distributed arbitrarily on the Pareto frontier without considering the balanced relationship between the individual objectives. However, our goal is to find a balanced Pareto optimal solution for each participant in the market, which is *NP-hard*. The challenge is: *How can we efficiently find a balanced Pareto optimal solution to build a fair data market?*

In this paper, we propose a fair and quality-based data market **FQora** to address the identified challenges. In order to address the first challenge, we developed a quality assessment function to evaluate the quality of the queried dataset. Additionally, we devised a quality-based pricing function to ensure that the market is arbitrage-free, which is contingent upon the user's specified quality requirements. We partition the fairness into long-term low risks and short-term balance for the second challenge. We first introduce a mean variance frontier to adaptively adjust the weights of the various participants' utilities during the long-term trading process of the market. Secondly, to ensure balance during the optimization process of a single trade, we propose a balanced Pareto optimal solution. Through adjusting the market price in the current transaction, we present a balanced Pareto optimization algorithm that equalizes the utility of all market participants.

Our contributions can be summarized as follows:

- We proposed a **F**air and **Q**uality-based data ma**r**ket for m**a**chine learning **FQora**. To the best of our knowledge, this is the first quality-based data market considering balanced utility allocation between sellers, buyers, and the arbiter.
- We theoretically analyzed the benefit of the revenue allocation for the arbiter, which can promote market demand and maintain market equity.
- We proposed a mean variance frontier and a Balanced Pareto Optimization algorithm via adaptively reweighted gradients to build a short-term balanced and long-term low-risk data market.
- Extensive experiments provide support for our theoretical analysis and affirm the superior performance of our proposed FQora.

The rest of the paper is organized as follows. We first introduce the preliminaries of the data market in Sec. 2. Then, we cover the details of our quality-based pricing framework in Sec. 3. We show our experimental results in Sec. 4, discuss related works in Sec. 5, and conclude in Sec. 6.

2 Preliminaries

In this section, we review the concept of the data market in Sec. 2.1 and prove the validation of the arbiter's revenue allocation in Sec. 2.2. Next, we formulate the willingness of the buyer as a willing function to represent the probability of a buyer's purchase in Sec. 2.3. Then, we list several quality functions that can be used for computing the quality scores in Sec. 2.4. Finally, we define the optimization objective of the fair data market, which is an NP-hard problem to solve, in Sec. 2.5. Table 1 summarizes the notations used in this paper.

2.1 Data Market

Consider a data market $\mathcal{M}$, which has a series of sellers ($s_i \in S$, $1 \leq i \leq N_s$), buyers ($b_i \in B$, $1 \leq i \leq N_b$), and an arbiter α facilitating data transactions between buyers and sellers. Sellers and buyers can be individuals, teams, divisions, or whole organizations.
Seller Settings. Each seller s_i in the market has his/her own dataset $\mathcal{D}_i = \{\mathrm{X}_i, \boldsymbol{y}_i\}_i^{N_s}$, where $\mathrm{X}_i = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k\}_{k=1}^{C_x}$ is the feature of data and $\boldsymbol{y}_i \in \mathbb{R}^{d_y}$ is the label for the corresponding X_i. Each $\boldsymbol{x}_k$ denotes

Table 1: The summary of major notations.

Notation	Definition
$\mathcal{M}$	the data market
s_i	the i-th seller in $\mathcal{M}$
N_s	the number of sellers in $\mathcal{M}$
$\mathcal{D}_i$	data possessed by s_i
$\mathcal{S}$	the support set of pricing anchor points
b_i	the i-th buyer in $\mathcal{M}$
N_b	the number of buyers in $\mathcal{M}$
Q_i	the query requested by b_i
v_i	valuation of b_i for the query Q_{ij}
α	the arbiter in $\mathcal{M}$
$\pi(\cdot)$	the quality-based pricing function
U_{s_i}	the utility of seller s_i
U_{b_i}	the utility of buyer b_i
U_α	the utility of arbiter α
$\mathbf{B}$	the mean variance weight of MVC
d	the update direction of pricing function $\pi(\cdot)$
$\mathbf{w}$	the balanced weight of BPO

one attribute of $\mathcal{D}_i$. For example, image, gender, and age are usually known as $\mathbf{X}_i$ in medical images and C_x is the number of attributes of $\mathbf{X}_i$. And we assume that the seller is truthful to the arbiter. In other words, the arbiter can access the data possessed by the seller. The common price support set $\mathcal{S}$ is shared by every seller to assist the arbiter in determining a reasonable price.

Arbiter Settings. Typically, the arbiter α plays the trusted third-party role in the data transaction process. Moreover, the arbiter knows the details of data from the sellers and produces the price of the data for buyers by using a quality-based pricing function $\pi(\cdot)$. The arbiter can efficiently collect the requested data and accurately assess the quality of the data suitable for machine learning on query data and price it appropriately.

Buyer Settings. Each buyer b_i in the market will submit a query Q_i with quality constraints and a valuation v_i based on their needs to the arbiter α. Moreover, all buyers are aware of the data quality requirements for dataset $\mathcal{D}_i$.

2.2 Validation of the Arbiter's Revenue Allocation

The conventional data market paradigm primarily consists of two primary stakeholders: sellers and buyers. Sellers usually also assume the duties of arbiters [13]. Compared with previous work [7, 8, 10, 18], we assume an independent arbiter that allocates the revenue obtained by the data market. Therefore, we theoretically validate the assumption in this part. Firstly, we introduce the maximized revenue obtained by the seller and the minimum budget used by the buyer to purchase the same data when there is no arbiter in the data market. Moreover, we compute the difference when the arbiter joins the data market.

Utility Definition. We formulate the utility of seller s_i as U_{s_i}, and the utility of buyer b_i for wanting to buy the demand data as U_{b_i}. Considering the fact that only one seller cannot satisfy the query of b_i's demand, we separate Q_i into Q_{ij}, which means a query from seller s_i to buyer b_j. Similarly, $\pi(Q_{ij})$ denotes the price of query Q_{ij} paid by b_i to s_i. The collection cost of producing query data Q_{ij} is formulated as $C(Q_{ij})$. Similarly, V_i denotes the value of the query data of data quality Q_{ij} to buyer b_i. $\pi^\alpha(Q_{ij})$ is a function representing the arbiter's assessment of the data quality and price fairness. According to the existence of the arbiter, query Q_{ij} may have changed. Thus, we use Q'_{ij} to distinguish different queries owned in such two different scenarios. During the dynamic trade process, we use t to denote the t-th iteration of the purchase process.

Definition 2.1 (Utility of Seller and Buyer w/o Arbiter). In the scenario without an arbiter: the utilities of the buyer and the seller are given by:

$$U_{b_i} = \sum_{t,j} V(Q^t_{ji}) - \pi(Q^t_{ji}),\ U_{s_i} = \sum_{t,j} \pi(Q^t_{ij}) - C(Q^t_{ij}).$$

Definition 2.2 (Utility of Seller and Buyer with Arbiter). The utilities of the buyer and seller in the scenario involving an arbiter are now formulated as follows:

$$U^\alpha_{b_i} = \sum_{t,j} V(Q^{t\prime}_{ji}) - \pi^\alpha(Q^{t\prime}_{ji}),\ U^\alpha_{s_i} = \sum_{t,j} \pi^\alpha(Q^{t\prime}_{ij}) - C(Q^{t\prime}_{ij}).$$

Arbiter Benefits. In the absence of an arbiter in the market, transactions occur directly between buyers and sellers. However, reaching agreements can be challenging because the seller s_i may struggle to set reasonable prices. Consequently, the market is characterized by information asymmetry. Buyers cannot perfectly assess the quality of data, leading to potential market inefficiency (e.g., adverse selection). Moreover, buyers b_i, may seek extensive datasets within specific domains to train foundational models. Simultaneously, buyers in the machine learning domain often harbor specific quality requirements for data, potentially necessitating the purchase of large volumes of irrelevant data to attain desired quality standards.

Assumption 1. *For the data market $\mathcal{M}$, the introduction of the arbiter α will effectively enhance the demand for all queries Q_{ij} than without the arbiter, thus the price and value of the query data for both sellers and buyers are both higher than before:*

$$s_i : \sum_{t,j} \pi^\alpha(Q'_{ij}) > \sum_{t,j} \pi(Q_{ij}), b_i : \sum_{t,j} V(Q'_{ji}) > \sum_{t,j} V(Q_{ji}).$$

Assumption 2. *For s_i and b_i in market $\mathcal{M}$, by increasing the demand for query data, we assume that the improvement of the benefits for both s_i and b_i is greater than the cost of collecting data and purchasing data, respectively:*

$$s_i : \sum_{t,j} \pi^\alpha(Q^{t\prime}_{ij}) - \pi^\alpha(Q^t_{ij}) \geq \sum_{t,j} C(Q^{t\prime}_{ij}) - C(Q^t_{ij}),$$

$$b_i : \sum_{t,j} V(Q^{t\prime}_{ji}) - V(Q^t_{ji}) \geq \sum_{t,j} \pi^\alpha(Q^{t\prime}_{ji}) - \pi^\alpha(Q^t_{ji}).$$

Thus, we find that if the arbiter exists, the utility of sellers and buyers are both larger than there is not.

Theorem 2.3. *For the buyer b_i, the utility after the inclusion of the arbiter is greater than that without the arbiter: $U^\alpha_{b_i} \geq U_{b_i}$. Meanwhile, for sellers, the utility after the inclusion of the arbiter is also greater than that without the arbiter: $U^\alpha_{s_i} \geq U_{s_i}$.*

Proof. Consider a data market $\mathcal{M}$ with multiple sellers s_i and rational buyers b_i and the presence of information asymmetry.

For the buyer b_i, the difference of the utility of b_i is formulated as:

$$\begin{aligned} U^{\alpha}_{b_i} - U_{b_i} &= \sum_{t,j} [V(Q^{t\prime}_{ji}) - \pi^{\alpha}(Q^{t\prime}_{ji})] - [V(Q^{t}_{ji}) - \pi(Q^{t}_{ji})] \\ &= \sum_{t,j} \underbrace{[V(Q^{t\prime}_{ji}) - V(Q^{t}_{ji})]}_{\text{Data Value Increase}} - \underbrace{[\pi^{\alpha}(Q^{t\prime}_{ji}) - \pi(Q^{t}_{ji})]}_{\text{Data Price Increase}} . \end{aligned} \tag{1}$$

For the seller s_i, the difference of the utility of s_i is formulated as:

$$\begin{aligned} U^{\alpha}_{s_i} - U_{s_i} &= \sum_{t,j} [\pi^{\alpha}(Q^{t\prime}_{ij}) - C(Q^{t\prime}_{ij})] - [\pi(Q^{t}_{ij}) - C(Q^{t}_{ij})] \\ &= \sum_{t,j} \underbrace{[\pi^{\alpha}(Q^{t\prime}_{ij}) - \pi(Q^{t}_{ij})]}_{\text{Data Price Increase}} - \underbrace{[C(Q^{t}_{ij}) - C(Q^{t}_{ij})]}_{\text{Collection Cost Increase}} . \end{aligned} \tag{2}$$

Obviously, the structure of the difference between the two utilities is similar. However, there are a few differences. First, for the *Data Price Increase*, the increment is not the same for s_i and b_i because s_i sells the data to b_j and b_i buys the data from s_j. Therefore, Eq. (1) and Eq. (2) all have the *Data Price Increase* part, but not a trade-off part for each other.

Based on the Assumption 1, we can conclude that each part of Eq. (1) and Eq. (2) are non-negative due to the increasing demand of buyers' queries.

$$\sum_{t,j} V(Q^{t\prime}_{ji}) - V(Q^{t}_{ji}) \geq 0, \sum_{t,j} \pi^{\alpha}(Q^{t\prime}_{ji}) - \pi(Q^{t}_{ji}) \geq 0.$$
$$\sum_{t,j} \pi^{\alpha}(Q^{t\prime}_{ij}) - \pi(Q^{t}_{ij}) \geq 0, \sum_{t,j} C(Q^{t}_{ij}) - C(Q^{t}_{ij}) \geq 0.$$

We still do not know whether the difference is positive or negative. Next, we prove the magnitude relationship of each part in Eq. (1) and Eq. (2). Based on Assumption 2, we can get the result of each equation:

$$\begin{aligned} &\sum_{t,j} [V(Q^{t\prime}_{ji}) - V(Q^{t}_{ji})] - [\pi^{\alpha}(Q^{t\prime}_{ji}) - \pi(Q^{t}_{ji})] \geq 0 \\ \Rightarrow\ & U^{\alpha}_{b_i} - U_{b_i} \geq 0 \Rightarrow U^{\alpha}_{b_i} \geq U_{b_i}. \\ &\sum_{t,j} [\pi^{\alpha}(Q^{t\prime}_{ij}) - \pi^{\alpha}(Q^{t}_{ij})] - [C(Q^{t\prime}_{ij}) - C(Q^{t}_{ij})] \geq 0 \\ \Rightarrow\ & U^{\alpha}_{s_i} - U_{s_i} \geq 0 \Rightarrow U^{\alpha}_{s_i} \geq U_{s_i}, \end{aligned}$$

This concludes our correctness proof. □

Conclusion. Therefore, the presence of an unbiased arbiter α in a multi-seller data market increases the utility for both buyers and sellers by ensuring fair pricing, reflecting the true value of data, and encouraging a competitive market environment. Note that in the rest of the paper, we overwrite the quality-based pricing function $\pi(\cdot)$ as the pricing function when the arbiter exists.

2.3 Willingness Function

In addition to failing to take into account the revenue allocation for the arbiter, in the previous work [7, 10, 18], they also regarded the willingness of buyers as zero when the price of a query $\pi(Q_i, \mathcal{D})$ is larger than the valuation of buyer v_i. However, this is not practical in the real-world transaction process. For example, Alice wants to buy high-quality training data, but the price of the data is \$1 higher than the previous budget. In practice, Alice will still buy the demand data even if the price of the data is a little higher than the original budget. Therefore, we need to introduce a Willingness Function (WF) for modeling the probability that buyers want to purchase the data.

When the pricing of the query $\pi(Q_i, \mathcal{D})$ is gotten, the willingness of the buyer to purchase this query Q_i is based on their valuation v_i. In general, the willingness function $\mathbb{H}(x) : \mathbb{R} \rightarrow [0, 1]$ can be formulated as follows:

$$\mathbb{H}(x) = \begin{cases} 1, & x < 0. \\ \kappa, & x \geq 0. \end{cases} \tag{3}$$

The difference between the price and valuation of queried data $p_i - v_i$ is the input x. We assume that the willingness of buyers is a continuous random variable between 0 and 1 when x. And κ is a random variable. Due to willingness sometimes equals probability, we refer to the work [27] and assume the probability random variable π follows (α, β) Beta distribution.

2.4 Data Quality Assessment

To address the first challenge posed by the query with quality constraints, it is necessary to employ techniques for assessing the data's quality. For data quality assessment, there are some typical methods, such as label balance, noise transition [15, 21, 38], label disturbance probabilities [29, 35], etc. Next, we briefly introduce two types of algorithms with the capacity to quantify data quality. **Label Balance** estimates the quality of data by the degree of balance between the number of data labels. If the label distribution closes to the uniform distribution, the label balance algorithm will give a higher score to the corresponding dataset. The specific quantified assessment formula is as follows:

$$S_{balance} = \left(1 - \sum_{c=1}^{C} \max\left(\frac{\frac{N_c}{N} - \frac{1}{C}}{1 - \frac{1}{C}}, 0\right)\right)^2 . \tag{4}$$

Sieving Frameworks integrate the model's confidence scores during training to assess label disturbance probabilities for each sample. These methods [29, 35] involve setting a reliable threshold to compare sample confidences and obtain a noise-free sample set. We denote the clean score as follows:

$$S_{clean} = \frac{|\{(x_i, \bar{y}_i) \in \mathcal{D} | f(x_i) > \tau_i, y'_i = \bar{y}_i\}|}{|\mathcal{D}|}, \tag{5}$$

where τ_i is a confidence threshold for each sample in $\mathcal{D}$.

2.5 Fair Data Market Optimization Objective

After introducing a set of definitions of our proposed fair data market $\mathcal{M}$, we proceed to redefine the optimization objective of the fair data market $\mathcal{M}$. To calculate this objective, it is necessary to establish the utilities of each participant in the market. In the case of arbiter α, its utility is defined as follows:

$$U_{\alpha} = \rho \sum_{i,j} p_{ij} \cdot \mathbb{H}(p_{ij} - v_{ij}), \tag{6}$$

where U_{α} denotes the utility of the arbiter α. ρ is the commission rate to control the revenue obtained by α. $\mathbb{H}\{\cdot\}$ The willing function, as previously indicated in Sec. 2.3, measures the level of willingness exhibited by purchasers in their purchase decisions.

Therefore, we can calculate the utility of arbiter α based on each sold data products $\pi(Q_{ij}) \cdot \mathbb{H}(p_{ij} - v_{ij})$ and the control parameter ρ. For simplification, we rewrite the utility of sellers according to Eq. (2) without considering the cost of collection data:

$$U_s = \sum_{ij} p_{i,j} \cdot \mathbb{H}(p_{ij} - v_{ij}) - U_\alpha, \tag{7}$$

For the pricing constraints, we follow the assumption related to [10]. We also assume the sellers will provide a set of support price points: $\mathcal{S} = \{(q_1, \pi_1), (q_2, \pi_2), \ldots, (q_k, \pi_k)\}$, where (q_k, π_k) is a price point defining the price π_k that the price function $\pi(\cdot)$ should approximate based on the quality score q_k in the transaction process.

$$T((q_1, p_1), (q_2, p_2), \cdots, (q_k, p_k)) = -\sum_k |p_i - \pi(q_i)|. \tag{8}$$

Thus, our final optimal objective can be formulated as:

$$\begin{aligned} \arg\max_\pi \quad & (U_s, U_\alpha, U_b, T) \\ \textbf{s.t.} \quad & \pi(Q_i) > \pi(Q_j), i > j \\ & S(Q_i) > S(Q_j), i > j \end{aligned} \tag{9}$$

where the constraint of the multi-objective optimization is the quality monotone, illustrated in Sec 3.2. Thus, the optimization objective of the fair data market turns into a MOO (multi-objective optimization) problem. A Pareto solution set is a set of solutions distributed on the Pareto frontier that satisfies the requirement of optimization objective. We next introduce the definitions of Pareto Dominance and Pareto Optimality in the fair data market $\mathcal{M}$.

Definition 2.4 (Pareto Dominance). Given two prices functions {π^*, π}, a pricing function π^* Pareto dominates π ($\pi^* \twoheadrightarrow \pi$) for Data Market $\mathcal{M}$ when two conditions are met:

(i) *No participants in the market have a strict preference that π to π^*, it implies, for U_b, U_s, and U_α, $U_b(\pi^*) \leq U_b(\pi)$, $U_s(\pi^*) \leq U_s(\pi)$, $U_\alpha(\pi^*) \leq U_\alpha(\pi)$.*

(ii) *At least one participant distinctly favors π^* to π, that is, $U_b(\pi^*) < U_b(\pi)$, $U_s(\pi^*) < U_s(\pi)$, $U_\alpha(\pi^*) < U_\alpha(\pi)$.*

Definition 2.5 (Pareto Optimality). π^* is a Pareto optimal pricing function and $\{U_b(\pi^*), U_s(\pi^*)$, and $U_\alpha(\pi^*)\}$ is a set of optimal utilities for buyers, sellers, and the arbiter if it does not exist $\bar{\pi} \twoheadrightarrow \pi^*$. This means a pricing function solution, which no other solution dominates, is termed as Pareto optimal.

Therefore, we now formally define the optimization problem studied in this work as below:

Definition 2.6 (Fair and Quality-based Data Market Optimization Problem). Given a data market $\mathcal{M}$, assume there exist two sets: $\{s_i\}_{i=1}^{N_s}$ for sellers and $\{b_i\}_{i=1}^{N_b}$ for buyers. The duty of the arbiter α is to collect the price-sorted queries Q_i from b_i and search the demanded data from s_i. The problem of building a fair and quality-based data market is to find an optimal quality-based pricing function π^* for the arbiter α, such that there is no $\bar{\pi} \twoheadrightarrow \pi^*$for the optimization objective of maximum the utilities of all participants: $\arg\max_\pi(U_s, U_\alpha, U_b, T)$, *which enable the balanced distribution of utilities between market participants in $\mathcal{M}$.*

3 Fair and Quality-based Pricing Framework

In this section, we first introduce the basic pipeline of FQora. Then, we illustrate the prerequisites for a quality-based pricing function (QPF) and propose the QPF used in FQora. In addition, we introduce mean variance constraint to help the low-risk development from a long-term perspective and propose a balanced Pareto optimization algorithm to solve the fair allocation of utilities between participants from a short-term perspective. Finally, we give the theoretical guarantees for our proposed FQora.

3.1 The pipeline of FQora

In the setting of FQora, we can divide the participants into three parts: sellers, buyers, and the arbiter. As shown in Fig. 2, in a process of trade, buyer b_i first gives a quality-constrained query to the arbiter α. Subsequently, the arbiter proceeds to choose the data that meets the specified query and uses the quality-based pricing function π, which is defined in Sec. 3.3 and price anchor $\mathcal{S}$ from seller s_i to price the query and compute the utilities for all participants. Next, the arbiter α allocates balance utilities to all participants by mean variance constraint and balanced Pareto optimization (BPO), which are defined in Sec. 3.4 and Sec. 3.6. Finally, the arbiter obtains an updated quality-based pricing function π.

3.2 Desiderata for Quality-based Pricing Function

For practical application, the quality-based pricing function needs to handle all quality requirements proposed by buyer b_i, which means our quality-based pricing function should satisfy the properties of the pricing function, where the pricing function $\pi(Q, \mathcal{D}) \rightarrow \mathbb{R}^+$ indicates how much money buyers need to pay to obtain the query data. For the market to function effectively, these pricing functions must adhere to a set of desired properties, ensuring certain guarantees for both the seller and the buyer. In particular, the quality-based pricing function needs to have the following desiderata:

Quality Monotone. We want to make sure that if for a smaller quality score q, then the price of the data is smaller (or equal) for the higher quality score. Otherwise, a buyer who wants to buy the data with a higher quality score can purchase it for a lower price. The formal definition is as follows:

Definition 3.1 (Quality Monotone). A pricing function π is quality-monotone in dataset $\mathcal{D}$ if for every query $Q_1, Q_2 \in \Omega$, $S(Q_1) \geq S(Q_2)$implies that $\pi(Q_1, \mathcal{D}) \geq \pi(Q_2, \mathcal{D})$.

The concept of quality monotonicity suggests that in cases where two separate queries possess identical quality scores, denoted as $S(Q_1) = S(Q_2)$, their pricing should correspondingly be equal, thus $\pi(Q_1, \mathcal{D}) = \pi(Q_2, \mathcal{D})$. This property indicates that the pricing is influenced not by the specific parameters of the mechanism but rather by the quality inherent in the queried data.

Non-Negativity. Clearly, the quality-based pricing function has to be non-negative, since the buyer should not be able to make money from the arbiter by obtaining query data.

Definition 3.2 (Non-Negativity). A pricing function π, is non-negative in dataset $\mathcal{D}$ iff for every query $Q \in \Omega$, $\pi(Q, \mathcal{D}) \geq 0$.

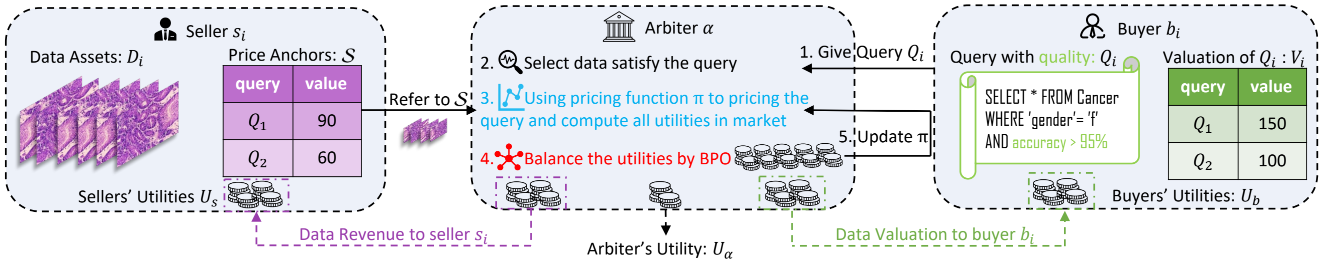

Seller s_i
Data Assets: D_i
Price Anchors: $\mathcal{S}$
query | value
Q_1 | 90
Q_2 | 60
Sellers' Utilities U_s
Refer to $\mathcal{S}$
Arbiter α
2. Select data satisfy the query
3. Using pricing function π to pricing the query and compute all utilities in market
4. Balance the utilities by BPO
1. Give Query Q_i
5. Update π
Buyer b_i
Query with quality: Q_i
SELECT * FROM Cancer
WHERE 'gender'= 'f'
AND accuracy > 95%
Valuation of Q_i : V_i
query | value
Q_1 | 150
Q_2 | 100
Buyers' Utilities: U_b
Data Revenue to seller s_i
Arbiter's Utility: U_α
Data Valuation to buyer b_i

Figure 2: The framework of our proposed FQora. The whole process of data transaction can be seen as five steps in one trade. Please refer to Sec. 3.1 for the explanation.

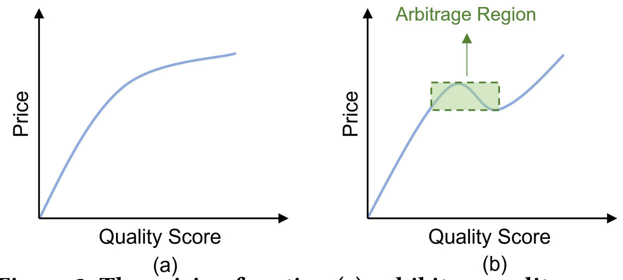


Figure 3: The pricing function (a) exhibits a quality monotone. The pricing function (b) fails to meet arbitrage-free situations. Undoubtedly, the entirety of the darkened region depicted is wasteful for the seller, as it results in a loss of prospective money.

Buyer-Oriented Arbitrage-Free. The quality-based pricing function must prevent arbitrage as a key feature. To understand the importance of this attribute, take an example.

Suppose b_i, a buyer, wants a single instance of high-quality data at a significant price. If b_i can merge multiple instances of similar data at a lower cost than the original instance, b_i may prefer the collection over the original instance, resulting in higher quality than the market standard. This is arbitrage. Effective quality-based pricing functions require avoiding arbitrage, which is either nonexistent or highly unlikely. More details on this idea are provided in the following definitions:

Definition 3.3 (Subadditivity). We define that a quality-based pricing function π is subadditive in dataset $\mathcal{D}$ if there are query requests $Q, Q_1, Q_2, \dots, Q_k$ and a mapping function h: $Q^k \rightarrow Q$ such that $\sum_{i=1}^{k} \pi(Q_i, \mathcal{D}) < \pi(Q, \mathcal{D})$ and $S(\bigcup_{i=1}^{k} Q_i) < S(Q)$, where $\bigcup_{i=1}^{k} Q_i$ is the "combined" dataset aggregated by $Q_1, Q_2, \dots, Q_k$.

In this scenario, a buyer aims to aggregate various query instances to create a new instance that is both high-quality and exhibits a significantly reduced price. This objective is akin to increasing the quality of purchased data while ensuring efficiency and effectiveness within a machine learning framework. It's important to note that the sole limitation on the buyer's capability is the requirement to maintain data quality, with no restrictions imposed on their computational resources.

Definition 3.4 (Arbitrage-Free). A pricing function π is arbitrage-free in dataset $\mathcal{D}$ if it satisfies both quality-monotone and subadditive.

3.3 Quality-based Pricing Function

Unlike traditional pricing strategies that might overlook the significance of data quality, the Quality-based Pricing Function (QPF) introduces a nuanced approach, offering a more equitable and realistic valuation of data based on its extrinsic and intrinsic qualities.

At its core, QPF is predicated on the principle that not all data is created equal. The value of data is intrinsically linked to its quality, which can vary greatly depending on factors such as Sec. 2.4 mentioned label balance, size, and accuracy computed by sieving frameworks. QPF seeks to quantify these attributes and incorporate them into the pricing model. The general formula for QPF could be represented as:

$$p_{ij} = \pi(Q_{ij}, \mathcal{D}) = \pi(S(Q_{ij}, \mathcal{D}, \mathbf{w}^q)), \tag{10}$$

where Q_{ij} is the query from buyer b_i to seller s_j, $\pi(\cdot)$ is the quality-based pricing function that computes the price of quality query Q_{ij}. Indeed, $\pi(\cdot)$ is a mapping function between quality score and price. Moreover, $S(\cdot)$ is the quality score function computing the quality score of Q_{ij} is defined as follows:

$$S(Q_{ij}, \mathcal{D}, \mathbf{w}^q) = \sum_k w_k^q \times \phi_k(Q_{ij}, \mathcal{D}) = q_{ij}, \tag{11}$$

where $\mathbf{w}^q$ is the weight vector of different quality assessments and w_k^q is the specific weight for k-th quality for query Q_{ij}. And $\phi_k(\cdot)$ is the k-th required quality function from buyer b_i as mentioned in Sec. 2.4. In addition, the final quality score of query Q_{ij} is represented by q_{ij}.

The allocation of weights among various quality evaluations can be delineated in two ways: firstly, through the buyer's query to obtain, wherein the user establishes an optimal weight scheme for the quality of the different dimensions based on their own criteria; The second scenario is one in which the purchaser does not designate a particular weight, but rather the arbitrator takes into account the quality of the various dimensions in line with the sequential addition of the query's degree of importance.

3.4 Mean Variance Constraint

The second challenge of the traditional data market is to build a fair data market that can make a balanced allocation of utilities between participants. However, before considering the solution to equalize the distributional benefits, we should first consider how to make the market work efficiently and effectively in the long run. Therefore, we have to consider the weight of different objectives to maximize the total circulation value of the data market.

In order to make the data market $\mathcal{M}$ work well in the long term, the best method is to decrease the risk of the data market. For measuring the degree of the risk of the data market, we first regard the whole process of data transactions in $\mathcal{M}$ as a dynamic process during T iterations. Therefore, we can obtain the average of each utility objective by

$$\mu_n^t = \frac{1}{i-1}\sum_{i=1}^{t} U_s^{(i-1)}, \;\; n = b, s, \text{and } \alpha. \tag{12}$$

Similarly, the corresponding covariance of different participants' utilities can be obtained by

$$\begin{aligned} Cov(U_m, U_n)^t =& \frac{1}{t-2}\sum_{i=1}^{t}(U_m^{i-1} - \mu_m^{i-2})(U_n^{i-1} - \mu_n^{i-2}), \\ & where\ m \neq n \text{ AND } m, n = b, s, \text{and } \alpha. \end{aligned} \tag{13}$$

Reducing the variance of utilities not only enhances individual satisfaction but also plays a pivotal role in mitigating the risk associated with data markets. Moreover, a decrease in utility variance fosters greater predictability and stability within the data market, facilitating smoother transactions and reducing the potential for market volatility. Ultimately, this contributes to a more resilient and efficient data ecosystem, where risks are more effectively managed, and stakeholders can confidently navigate the complexities of data-driven economies. Thus, we denote $\mathbb{E}(\mathbf{U})^t = [\mu_s^t, \mu_b^t, \mu_\alpha^t]$, and define the weight of sellers, buyers, and the arbiter as $\mathbf{B} = [\beta_1, \ldots, \beta_i]$, and $\Sigma \in \mathbb{R}^{m\times m}$ denotes the covariance of each utility of different participants in the market. Therefore, we can utilize Mean Variance Efficiency [26] to reduce the variance of the optimization of this data market for low-risk development by the following equation:

$$\min_{\mathbf{B}} \frac{1}{2}\mathbf{B}^\top \Sigma \mathbf{B}, \quad \textbf{s.t. } \mathbf{B}^\top \mathbb{E}(\mathbf{U}) = U_0, \; \mathbf{B}^\top \mathbf{1} = 1, \tag{14}$$

where Σ denotes the covariance matrix for participants in the data market, U_0 represents the expected utility value of the data market.

We utilize *Lagrangian Multiplier* method to solve the above mean variance problem, the detailed equation is defined as:

$$L(\mathbf{B}, \lambda_1, \lambda_2) = \frac{1}{2}\mathbf{B}^\top \Sigma \mathbf{B} - \lambda_1(\mathbf{B}^\top \mathbb{E}(\mathbf{U}) - U_0) - \lambda_2(\mathbf{B}^\top \mathbf{1} - 1). \tag{15}$$

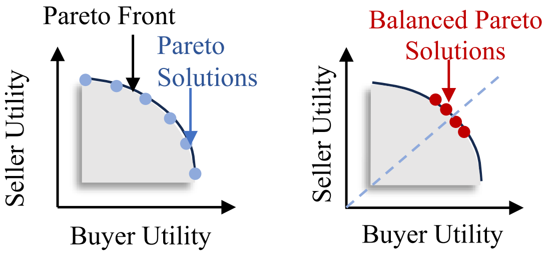
The gradient of L to $\mathbf{B}$, λ_1 and λ_2 can be computed as follows:

$$\begin{aligned} &\frac{\partial L}{\partial \mathbf{B}} = \Sigma\mathbf{B} + \lambda_1 \mathbb{E}(\mathbf{U}) + \lambda_2 \mathbf{I}, \\ &\frac{\partial L}{\partial \lambda_1} = \mathbf{B}^\top \mathbb{E}(\mathbf{U}) - U_0, \; \frac{\partial L}{\partial \lambda_2} = \mathbf{B}^\top \mathbf{1} - 1. \end{aligned} \tag{16}$$

We set the corresponding gradient $\frac{\partial L}{\partial \mathbf{B}}$, $\frac{\partial L}{\partial \lambda_1}$ and $\frac{\partial L}{\partial \lambda_2}$ to zero, thus can obtain the $\mathbf{B}$ by solving the following linear system of equations:

$$\begin{bmatrix} \Sigma & -\mathbf{U} & -\mathbf{1} \\ \mathbf{U}^\top & 0 & 0 \\ \mathbf{1}^\top & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{B} \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0 \\ U_0 \\ 1 \end{bmatrix}. \tag{17}$$

Thus, we can obtain the preferred mean variance weight $\mathbf{B}$ to control the long-term low-risk development of the data market $\mathcal{M}$ by keeping a lower level variance of utilities.



(a) Pareto Optimality (b) Balanced Solutions

Figure 4: Comparative Visualization of Multi-Objective Optimization Techniques. (a) Demonstrates the dispersion of random Pareto optimal solutions. (b) Illustrates the strategic selection of balanced Pareto optimal solutions within our BPO, emphasizing their alignment with the primary objective of Average Utility.

3.5 Balanced Pareto Optimal Solutions

With the low-risk development of the data market guaranteed by the mean variance constraint, we can now consider how to find a balanced Pareto optimal solution for the data market. For a simple explanation of gradient-based optimization in this work, we first transfer this maximum optimization objective into a minimization problem and solve the equality problem by gradient-based methods, like MGDA [33].

$$\begin{aligned} \arg\min_{\pi} \quad & (-U_s, -U_\alpha, -U_b, -T) \\ \textbf{s.t.} \quad & \pi(Q_i) > \pi(Q_j), i > j, \\ & S(Q_i) > S(Q_j), i > j. \end{aligned} \tag{18}$$

As we mentioned in Sec. 2.5, the aggregate of all solutions deemed Pareto optimal is referred to as the Pareto set, and its corresponding representation in the utility space is known as the Pareto front, as described in Fig. 4. This paper concentrates on employing gradient-based multi-objective optimization to find an optimal and balanced pricing function among participants. This approach is aimed at approximating the Pareto front in a manner that maximizes the average utilities.

Gradient-based MOO [33] seeks to identify a direction d that allows for the iterative discovery of a subsequent solution $\pi^{(t+1)}$ that dominates the predecessor one π^t which means $(U(\pi^{(t+1)}) \geq U(\pi^t))$ by moving against the computed direction d with step size η like $\pi^t = \pi^t - \eta d$. As [33] propose, employing the Multiple Gradient Descent Algorithm (MGDA) to achieve convergence towards a local Pareto optimum. This is accomplished through iterative application of the descent direction d, which is derived as follows:

$$\begin{aligned} d^* =& \arg\min_{d\in\mathbb{R}^m, \beta\in\mathbb{R}} \beta + \frac{1}{2}\|d\|^2 \\ & \textbf{s.t. } \nabla U_i \left(\pi^t\right)^\top d \leq \beta, \quad i = s, b \text{ or } \alpha, \end{aligned} \tag{19}$$

where d^* is the optimal direction capable of enhancing all utilities. Nevertheless, gradient-based multi-objective optimization approaches [24] aims to achieve a Pareto optimal solution by integrating gradients with weights that can be adjusted. However, these methods frequently fail to consider the true objective, which is balanced utility. In this work, we extend this methodology and introduce an innovative gradient-based strategy. This new approach is designed to balance gradients across different utilities effectively, thereby alleviating conflicts between utilities. Additionally, it aims to constrain the solution to ensure that it maximizes the weighted average utility.

For each agent b_i, s_i and α, we denote the gradients of U_b, U_s, U_α as g_b, g_s, g_α respectively. All these gradients, $g_i = \nabla U_i$ can be computed by the backward-propagation process from the utilities U_i. Moreover, g_i represents the optimal update direction for maximizing the utility U_i. Nevertheless, the irregularity in the preferred directional updates of price function π across agents presents a challenge, as divergent gradients may result in conflicts. Such conflicts can precipitate a stagnation in the optimization process, characterized by the excessive skew to certain participants in $\mathcal{M}$, e.g. buyers b_i or sellers s_i, at the expense of insufficient consideration of others. Intuitively, identifying an updated trajectory that not only minimizes gradient conflicts but also attains Pareto optimality is crucial for enhancing the efficacy of the MOO process.

Algorithm 1: FQora: Fair Quality-based Data Market

```
Input: Seller($s_i$): dataset $\mathcal{D}_i$, support price set $\mathcal{S}_i$; Buyer($b_i$):
       query $Q_i$, validation $v_i$; Arbiter($\alpha$): commission rate
       $\rho$, quality function $S(\cdot)$; Data Market ($\mathcal{M}$): iteration
       rounds $T$, size step $\eta$
Output: Pricing function $\pi^t$, seller utility $U_s^t$, buyer utility
        $U_b^t$, seller utility $U_\alpha^t$
Initialize $\pi^{(0)}$, $\mathbf{B} = [\frac{1}{m}, \dots, \frac{1}{m}]$;
Initial Utility Compute the original utility
    $U_s^{(0)}, U_b^{(0)}, U_\alpha^{(0)}$, Average of utility $\mu_n^t$;
for $t = 1, 2, \dots, T$ do
    $\pi^t \leftarrow \pi^{(t-1)}$;
    for $e = 1, 2, \dots, E$ do
        $d^* =$ BalancedParetoOptimization($\mathbf{B}$) ;
        $\pi_e^t \leftarrow \pi_{e-1}^t - \eta d^*$ ;
    end
    Obtain $U_s^t$, $U_b^t$, and $U_\alpha^t$                ▷ by Eq. (6) and (7);
    Compute $\mu, \sigma^2, \sum$ of $s_i, b_i$, and $\alpha$  ▷ by Eq. (12) and (13);
    Get the utility weight $\mathbf{B}$ by MVO               ▷ by Eq. (17);
end
return $\pi^T, U_s^T, U_b^T$, and $U_\alpha^T$ ;
Function BalancedParetoOptimization($\mathbf{B}$):
    $g_i \leftarrow \nabla_\pi U_i, i = b, s$ and $\alpha$ ;
    $g_a \leftarrow \frac{1}{m} \sum_i \beta_i g_i$;
    $[\gamma_1, \gamma_2, \dots, \gamma_i] = \arg\min_{d \in C} ||d||^2$;
    $g_{\gamma_i} \leftarrow \gamma_i g_i$;
    $[w_1, w_2, \dots, w_i] \leftarrow \arg\min_{\mathbf{w} \in \mathcal{W}^T} \frac{g_a^\top g_\mathrm{w} + \xi ||g_a||^2 ||g_\mathrm{w}||}{1 - \xi^2 ||g_a||^2}$;
    $d^* = \frac{g_a}{1-\xi^2 \|g_a\|^2} + \frac{\xi \|g_a\|^2 g_\mathrm{w}}{\left(1-\xi^2 \|g_a\|^2\right) \|g_\mathrm{w}\|}$;
return $d^*$;
```

3.6 Balanced Pareto Optimization Algorithm

We utilize the multiple gradient descent algorithm (MGDA) [33] to get a random Pareto optimal solution by finding a point of minimum-norm in the *Convex Hull* of all utilities' gradients C, and iteratively utilizing the descent direction d_{des}, defined as follows:

$$d_{des} = \arg\min_{d \in C} ||d||^2, \tag{20}$$

where C is a *Convex Hull* of gradients can be formulated as follows:

$$C = \{\mathcal{G}\gamma | \gamma \in \mathcal{W}^m\}, \ s.t. \mathcal{W}^m = \left\{ \mathcal{W}^m \in \mathbb{R}_+^m | \sum_{i=1}^m \gamma_i = 1 \right\} . \tag{21}$$

We define the matrix of the corresponding gradient to optimize specific agents' utility as $\mathcal{G} \in \mathbb{R}^m = \{g_b, g_s, g_\alpha\}$. $\mathcal{W}^m$ is a m-dimensional probability simplex.

Moreover, the optimized direction d_{des} can be rewritten as a convex combination of all gradients:

$$d_{des} = \sum_{i=1}^m \gamma_i g_i, \tag{22}$$

where $g_i = \nabla U_i(\pi)$ is the gradient of each sub-objective. It implies that, upon backpropagation towards an arbitrary Pareto optimal solution, the optimal gradient value for each utility objective is $g_{\gamma_i} = \gamma_i g_i$.

However, the strategy of deviating from the direction d_{des} fails to ensure that the resultant solution fulfills the specific demands of the fair Data Market. A quintessential goal within $\mathcal{M}$ is to reduce the gradient conflict inherent among concurrent participants, a measure crucial for augmenting efficacy across all objectives. To surmount this obstacle, it becomes necessary to identify and adopt a trajectory to facilitate a progression from an existing solution, thereby reconciling the need for the direction of updating price with the overarching objectives of $\mathcal{M}$. We aim to find the best direction, d, among the gradients used in updating our quality-based pricing function, to help improve the performance of the most unbalanced participants and guide the market toward the most balanced utility allocation based on the mean variance constraint method mentioned in Sec. 3.4. Our first step is to identify the gradient that is most different from d, meaning it has the largest angle compared to d, which helps us focus on improving where the market needs it most. This idea can be explained with the following formula:

$$\min_i \langle g_{\gamma_i}, d \rangle, \ \textbf{s.t.} \ g_{\gamma_i}^\top d \geq 0, i = 1, 2, \dots, m. \tag{23}$$

To improve the least effective gradient while balancing all gradients near the average gradient (defined as g_a), we need a careful approach. This involves adjusting gradients so that improving one doesn't harm others, aiming for a balanced solution that optimizes overall performance. We mathematically represent this gradient-balanced optimization problem via the dual problem (maximin optimization problem) as follows:

$$\max_{d \in \mathbb{R}^{N_g}} \min_{i \in [m]} \langle g_{\gamma_i}, d \rangle \quad \textbf{s.t.} \ g_a^\top d \geq 0, \ ||d - g_a|| \leq \xi g_a^\top d \quad , \tag{24}$$

where $g_{\gamma_i} = \gamma_i g_i$ is the weighted gradient computed by the Eq. (20), $\xi \in (0, 1]$ represents the fixed hyper-parameter that ensures consistent performance and reduces fluctuations of the optimization process. Due to the optimal direction d can also be constraint in the C of g_{γ_i}, we can easily obtain that

$$C^\gamma = \{\mathcal{G}_\gamma \mathbf{w} \mid \mathbf{w} \in \mathcal{W}^m\}, \tag{25}$$

where $\mathcal{G}_\gamma \in \mathbb{R}^{m \times N_g} = \{g_{\gamma_1}, g_{\gamma_2}, \ldots, g_{\gamma_m}\}$ is the gradient matrix of each agent in the data market $\mathcal{M}$, $\mathbf{w} = (w_1, w_2, \ldots, w_m)$ implies the weight of the optimal direction d composed by g_{γ_i}. Thus, the primal minimize problem $\min_i \langle g_{\gamma_i}, d \rangle$ can be transformed to the dual minimize problem $\min_{\mathbf{w}} \langle \sum_i w_i g_{\gamma_i}, d \rangle$. Then we can transform the above dual problem to the following type:

$$\max_{d \in \mathbb{R}^{N_g}} \min_{\mathbf{w} \in \mathcal{W}^m} \langle g_{\mathbf{w}}, d \rangle \quad \textbf{s.t. } g_a^\top d \geq 0,\ ||d - g_a|| \leq \xi g_a^\top d \quad . \tag{26}$$

Given that the objective function for d is concave, accompanied by linear constraints, and considering $\mathbf{w}$ belongs to a compact set $\mathcal{W}^m$, the application of Sion's minimax theorem [16] permits the interchange of the maximization and minimization operations without altering the solution to Eq. (27). This is formally expressed as follows:

$$\min_{\mathbf{w} \in \mathcal{W}^m} \max_{d \in \mathbb{R}^{N_g}} \langle g_{\mathbf{w}}, d \rangle \quad \textbf{s.t. } g_a^\top d \geq 0,\ ||d - g_a|| \leq \xi g_a^\top d \quad . \tag{27}$$

We can utilize *Lagrangian Multiplier* to solve the problem Eq. (26), the detail equation is defined as:

$$\min_{\mathbf{w} \in \mathcal{W}^m} \max_{d \in \mathbb{R}^{N_g}} g_{\mathbf{w}}^\top d - \lambda[||d - g_a||^2 - \xi^2 (g_a^\top d)^2]. \tag{28}$$

The optimal solution to the primal problem Eq. (13) can be obtained by solving the max-min problem defined by Eq. (28). Then we have

$$d^* = \frac{g_{\mathbf{w}} + \lambda^* g_a}{(1 - \xi^2 g_a^2)\lambda^*},\ \lambda^* = \frac{g_{\mathbf{w}}}{\xi ||g_a||^2}, \tag{29}$$

where d^* is the balanced and optimal update direction for the pricing function π, λ^* is the optimal Lagrange multiplier. And $g_{\mathbf{w}}$, denoted as $\sum_i^m w_i g_{\gamma_i}$, is the convex combination of g_{γ_i}. Therefore, we can obtain the optimal $\mathbf{w}^*$ by rewriting Eq. (24) to the following problem:

$$\arg \min_{\mathbf{w} \in \mathcal{W}^T} \frac{g_a^\top g_{\mathbf{w}} + \xi ||g_a||^2 ||g_{\mathbf{w}}||}{1 - \xi^2 ||g_a||^2}. \tag{30}$$

We summarized the pseudocode of the proposed FQora in Alg. 1.

3.7 Theoretical Analysis

In this subsection, we theoretically analyze the equivalence of the dual problem and the convergence of our proposed Balanced Pareto Optimization.

Theorem 3.5 (Equivalence of the Dual Problem). *Assume that both the primal and dual problems possess optimal solutions. Let $k^* = \min_{\lambda, \mathbf{w}} \max_d L(\lambda, \mathbf{w}, d)$ and $v^* = \max_d \min_{\lambda, \mathbf{w}} L(\lambda, \mathbf{w}, d)$. Thus, $k^* = \min_{\lambda, \mathbf{w}} \max_d L(\lambda, \mathbf{w}, d) = \max_d \min_{\lambda, \mathbf{w}} L(\lambda, \mathbf{w}, d) = v^*$.*

Proof. We use the mentioned *Lagrangian function* in Eq. (28).

$$L(\lambda, \mathbf{w}, d) = g_{\mathbf{w}}^\top d - \lambda[||d - g_a||^2 - \xi^2 (g_a^\top d)^2].$$

Let $\mathcal{L}_m(d) = \max_d L(\lambda, \mathbf{w}, d)$ and $\mathcal{L}_n(\lambda, \mathbf{w}) = \min_{\lambda, \mathbf{w}} L(\lambda, \mathbf{w}, d)$. Then, we can obtain:

$$\min_{\lambda, \mathbf{w}} L(\lambda, \mathbf{w}, d) \leq L(\lambda, \mathbf{w}, d) \leq \max_d L(\lambda, \mathbf{w}, d).$$

Therefore, we get the following inequality:

$$L_n(\lambda, \mathbf{w}) \leq L_m(d).$$

Due to both primal problems and dual problems having optimal values, we get: $\max L_n(\lambda, \mathbf{w}) \leq \min L_m(d)$.

Then, we obtain:

$$v^* = \min_{\lambda, \mathbf{w}} L(\lambda, \mathbf{w}, d) \leq \min_{\lambda, \mathbf{w}} \max_d L(\lambda, \mathbf{w}, d) = k^*.$$

Since the dual problem is a convex programming and the solutions d^*, λ^*, and $\mathbf{w}^*$ meet Karush-KuhnTucker (KKT) [5, 11] conditions, we can get: $k^* = v^* = L(\lambda^*, \mathbf{w}^*, d^*)$.

That is, the optimal value defined by Eq. (27) is equal to an optimal value defined by Eq. (26). Therefore, we can solve the complex Maximin Optimization Problem in Eq. (26) by solving its dual problem. □

Lemma 3.6. *if utility function U_i is L-smooth and differentiable, which means ∇U is a L-Lipschitz continuous then $U(\pi') \leq U(\pi) + \nabla U(\pi)^\top (\pi' - \pi) + \frac{L}{2} ||\pi' - \pi||^2$*

Proof of Lemma 3.6.

$$\begin{aligned} U\left(\pi'\right) &= U(\pi) + \int_0^1 \nabla U\left(\pi + x\left(\pi' - \pi\right)\right)^\top \left(\pi' - \pi\right) dx \\ &= U(\pi) + \nabla U(\pi)^\top \left(\pi' - \pi\right) \\ &+ \int_0^1 \left(\nabla U\left(\pi + x\left(\pi' - \pi\right)\right) - \nabla U(\pi)\right)^\top \left(\pi' - \pi\right) dx \\ &\leq U(\pi) + \nabla U(\pi)^\top \left(\pi' - \pi\right) \\ &+ \int_0^1 \left\|\nabla U\left(\pi + x\left(\pi' - \pi\right)\right) - \nabla U(\pi)\right\| \left\|\pi' - \pi\right\| dx \end{aligned}$$

(Using the definition of Lipschitz-continuous)

$$\begin{aligned} &\leq U(\pi) + \nabla U(\pi)^\top \left(\pi' - \pi\right) + \int_0^1 tL \left\|\pi' - \pi\right\|^2 dt \\ &= U(\pi) + \nabla U(\pi)^\top \left(\pi' - \pi\right) + \frac{L}{2} \left\|\pi' - \pi\right\|^2. \end{aligned}$$

□

Theorem 3.7 (Convergence of Balanced Pareto Optimization). *Because utility functions U_i are convex and differentiable, and L-smooth with $L > 0$. The pricing function is updated by the equation $\pi^t = \pi^{(t-1)} - \eta_\pi d$, where d is defined in Eq. (29) and $\mu_\pi^t = \min_{i \in [k]} \frac{||d - g_a||}{c \cdot L \cdot d^2}$. All the utility functions $U_s(\pi), U_b(\pi), U_\alpha(\pi)$ converges to $U_s(\pi^*), U_b(\pi^*), U_\alpha(\pi^*)$.*

Proof. We denote $\{\pi_e^t\}_{e=1}^E$ be pricing functions generated by using the optimal balanced update direction: $\pi_{e+1}^t = \pi_e^t - \eta d^*$, where d^* can be computed by Eq. (29). Due to all ∇U_i are Lipschitz continuous, for each utility objective $\{U_i\}_{i \in [m]}$, we utilize Lemma 3.6,

$$\begin{aligned} U_i\left(\pi_{e+1}^t\right) &\leq U_i\left(\pi_e^t\right) + \nabla U_i\left(\pi_{e+1}^t\right)^\top \left(\pi_{e+1}^t - \pi_e^t\right) \\ &+ \frac{L}{2} \left\|\pi_{e+1}^t - \pi_e^t\right\|^2 \\ &= U_i\left(\pi_e^t\right) - \eta_\pi^t \nabla U_i\left(\pi_e^t\right)^\top d + \frac{L}{2} \left\|\eta_\pi^t d\right\|^2 \end{aligned}$$

(Using the constraint $\|d - g_a\| \leq \xi g_a^\top d$)

$$\begin{aligned} &\leq U_i\left(\pi_e^t\right) - \frac{\eta_\pi^t \|d - g_a\|}{\xi} + \frac{\left(\eta_\pi^t\right)^2}{2} L \|d\|^2 \\ &= U_i\left(\pi_e^t\right) - \frac{\eta_\pi^t \|d - g_a\|}{\xi} + \frac{\eta_\pi^t}{2} \min_j \frac{\|d - g_a\|}{\xi} \\ &\leq U_i\left(\pi_e^t\right) - \frac{\eta_\pi^t \|d - g_a\|}{2\xi} \leq U_i\left(\pi_e^t\right). \end{aligned}$$

This statement indicates that when the FQora is applied, the value of the objective function for all participants consistently increases with each iteration. □

Following this, the discussion will turn to a detailed examination of the step size, η_π, as outlined in Lemma 3.8. This analysis is essential for understanding how the step size affects the algorithm's performance and the efficiency in FQora optimization scenario.

Lemma 3.8. *The convergence of GD algorithm, when employing a step size η_π, is contingent upon specific conditions related to the magnitude of η_π, which means η_π is meticulously selected such that $0 < \eta_\pi < \frac{1}{L}$, where L is the Lipschitz smoothness constant of utility function.*

Proof. (I) We first prove the left part of the inequality $\eta_\pi > 0$.

$$\eta_\pi = \min_{i\in[k]} \frac{||d-g_a||}{\xi\cdot L\cdot ||d||^2}, \ \textbf{s.t. } \xi \in (0,1], L>0$$

Thus, we can obviously get the result $\eta_\pi > 0$.

(II) Next, we prove the right part of the inequality $\eta_\pi < \frac{1}{L}$.

$$\begin{aligned}\eta_\pi &= \min_{i\in[k]} \frac{||d-g_a||}{\xi\cdot L\cdot||d||^2} \quad (\text{using } ||d-g_0|| \le \xi\cdot g_a^\top d)\\ &\le \min_{i\in[k]} \frac{\xi g_a^\top d}{\xi \cdot L\cdot ||d||^2} = \frac{g_a^\top d}{L\cdot||d||^2}\\ &= \frac{||g_a||\cdot||d||\cos\varphi}{L\cdot||d||^2} = \frac{||g_a||\cos\varphi}{||d||}\cdot\frac{1}{L},\end{aligned}$$

where $\varphi \in [0^\circ, 90^\circ)$ defined the limitation of the angle between the optimal direction d and g_a. Generally, we impose penalties on the gradient norm to enhance generalization and stability. Consequently, we can obtain $||d||^2 - ||g_0||^2 > 0$ when $\xi \in (0,1]$.Therefore,

$$\eta_\pi \le \frac{||g_a||\cdot||d||\cos\varphi}{L\cdot||d||^2} = \frac{||g_a||\cos\varphi}{||d||}\cdot\frac{1}{L} < \frac{1}{L},$$

Thus, we get the rational step size scope is $\eta_\pi \in (0, \frac{1}{L})$. □

4 Experiments

In this section, we empirically evaluate the performance and efficiency of the proposed data market framework **FQora**. We aim to answer the following research questions:

- **Q1.** Can it provide a greater number of balanced utilities for each participant in the data market in comparison to other baselines across various settings (different query distributions and willingness distributions)?
- **Q2.** Can it strongly protect the desiderat of pricing function such that $\pi(s_i)$ is monotone to the quality score s_i?
- **Q3.** How does the mean variance weight **B** and the balanced weight **w** change to control FQora to find the most balanced optimization direction d to achieve a fair data market?
- **Q4.** How efficiently does FQora generate an optimal price function to get the optimal utilities under different step sizes η?
- **Q5.** How does the stability rate ξ impact FQora?

4.1 Datasets

Our experimental evaluation was conducted by executing simulated queries across four synthesis datasets based on typical machine learning datasets spanning various domains. A summary of the datasets' characteristics is provided in Table 2, offering insight into their diversity and scope.

- Tiny-ImageNet [22]: An abbreviated version of the ImageNet dataset, which was developed for the purpose of evaluating the effectiveness of image recognition systems.
- CIFAR10 [19]: A dataset aimed at benchmarking computer vision models, representing a diverse set of objects.
- CIFAR100 [19]: Similar to CIFAR-10 but designed for finer granularity in benchmarking image classification algorithms.
- MNIST [23]: A benchmark dataset for evaluating handwriting recognition technologies.

Queries Generation.These original datasets do not have queries from buyers or price settings. Therefore, we follow the setting in previous work [10] to give each dataset a maximum pricing. For example, the maximum price is set as 300 and 100 for Tiny-ImageNet and CIFAR10, respectively. For query production, we first refer to [8] to generate the buyer's validation following the normal distribution and assign them randomly to each buyer. For the query generation, we refer to [10] to generate the appropriate query according to the distribution of the validation.

Table 2: Datasets characteristics

Dataset	# Number	# Class	# Max Value
TinyImageNet	110000	200	300
CIFAR10	60000	10	100
CIFAR100	60000	100	200
MNIST	70000	10	100

4.2 Implementation Details

Baselines. To evaluate the advantages of our proposed **FQora** in terms of achieving balanced utility for sellers, buyers, and the arbiter, we conducted a comparative study with four other data markets. Each of these markets utilizes well-behaved pricing functions for previous queries without quality constraints.

- **Max** establishes a singular price for all queries based on the query dataset's maximum value.
- **Mid** imposes a uniform cost on all queries, ensuring that a minimum of 50% of the purchasers can financially procure the queries.
- **Linear** utilizes a linear interpolation technique that takes the lowest and highest values in the given dataset to establish the pricing of queries.
- **QIRANA** [10] provides a query pricing framework to value the price of a single or a bundle of queries.

Experiment Settings. Since FQora is a quality-based data market, we should find proper quality functions as we introduced in Sec. 2.4. Based on these four ML datasets listed in Table 2, we select two functions: Histogram of Oriented Gradients (HOG) from a feature perspective and the degree of label balance from a label perspective to map the data quality to a proper quality score. For the iteration setting, we shuffle the generated query of each buyer and allow each buyer to have one query in each iteration. For all the following experiments, we set $T = 6$, $E = 60$, and $\xi = 0.01$, and repeat the experiment five times to calculate the average. Moreover, all experiments are conducted on a machine with Nvidia 4090 and Intel(R) Core(TM) i7-13700K under Ubuntu 22.04.3 LTS.

Evaluation Metric. To evaluate the average utility assignment, we utilize the harmonic mean of all participants' utilities $U_{\text{avg}} = \frac{m}{\sum_i^m \frac{1}{U_i}}$ to measure the average utility of the data market because of its sensitivity to extreme values and unbalanced values.

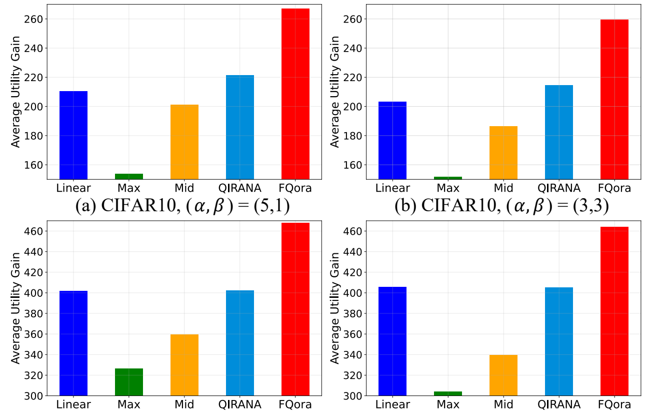
4.3 Comparison of Utility (Q1)

The first question is whether FQora can build a fair and low-risk data market. As shown in Table 3, FQora surpasses its counterparts in maximizing the balanced utility across all datasets. Even Qora, which is a simple version of FQora without Balanced Pareto Optimization, still performs better than other baselines on all simulated queries. Simultaneously, it is noteworthy that the Linear and QIRANA pricing strategies demonstrate comparable levels of utility gain. Then, we fix the query distribution and *vary the willingness function distribution* on CIFAR10 and CIFAR100 datasets to test the performance of FQora. As shown in Fig 5, we vary the parameters $\alpha = 5, \beta = 1$ to $\alpha = 3, \beta = 3$, all methods get a drop of the utility gain because of the increased purchase probability of buyers when prices are higher than valuations. Moreover, FQora achieves considerably higher balanced utility in comparison to the linear and QIRANA. This is because the linear approach fails to capitalize on the potential to offer queries to customers who are specifically interested in purchasing data with a moderate quality score. Similarly, we experiment with *variations in the distribution of queries*. We refer [8] to vary the query distribution from the normal distribution to the bimodal distribution. As shown in Fig 6, because of the change in the size of the bimodal peaks of the query value distribution, the single-value pricing approach will gain a large degree of degradation. However, FQora remains insensitive to changes in query distribution while consistently achieving optimal performance benefits.

We also conclude the visualization of utility assignment in Fig. 7. As shown in Fig. 7, utilities of sellers, buyers, and the arbiter are assigned more balanced by FQora. FQora's triangle is the most similar to an equilateral triangle in all baselines. This superior performance can be inferred as the proposed Balanced Pareto Optimization makes FQora get a balanced Pareto solution under the mean variance weight **B** constraint. The performance of Max and Mid is relatively lower in the results. Specifically, Max may skew to buyers because the max single value, and Mid may shift to sellers for a middle value for all queries. Therefore, the skewed value pricing in Max and Mid will cause a lower balanced utility allocation.

Table 3: Experiment Results on the Average Utility Gain (↑)

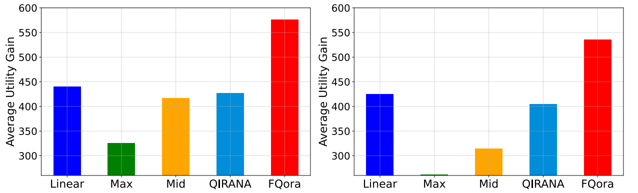
Pricing	CIFAR10	CIFAR100	TinyImageNet	MNIST
Linear	210.52	405.83	439.70	213.18
Max	153.76	304.30	325.75	158.84
Mid	221.53	339.51	416.79	166.56
QIRANA	201.91	405.14	426.70	220.62
Qora	231.27	418.52	511.53	223.21
FQora	**266.94**	**447.96**	**575.84**	**246.65**



(a) CIFAR10, $(\alpha, \beta) = (5,1)$ (b) CIFAR10, $(\alpha, \beta) = (3,3)$

(c) CIFAR100, $(\alpha, \beta) = (5,1)$ (d) CIFAR100, $(\alpha, \beta) = (3,3)$

Figure 5: The comparison of utility gain under different willingness settings.



(a) TinyImageNet, Normal Distribution (b) TinyImageNet, Bimodal Distribution

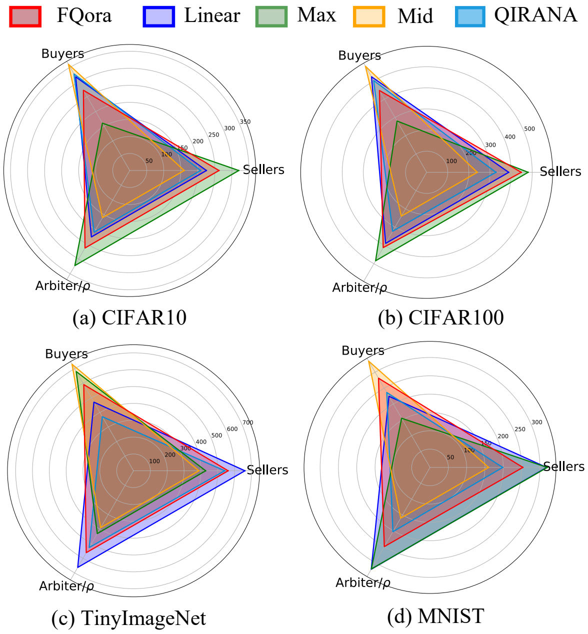
Figure 6: The comparison of utility gain under different query distributions.

4.4 Monotone Pricing to Quality Score (Q2)

The second question is whether it is true that the expected price behaves always monotonically as a function of the quality score of the queried data, which is the most important property for the pricing function defined in Sec. 3.2. We use CIFAR10 and CIFAR100 datasets to validate the generalization of the monotone of pricing function. Figure 8 confirms the existence of a quality monotonic relationship, thereby FQora satisfies buyer-oriented arbitrage-free. Other pricing frameworks like QIRANA or Linear give a linear quality score progression and also keep the quality monotone. However, FQora's prices are more consistent with the distribution of realistic commodity prices. Sellers normally make a lower price for smaller quantities, and the ease of trying them out can help attract buyers. In the middle score range, where most income comes from, prices move frequently, and the higher-quality data needed to represent intuitive benefits may not be clear.

4.5 Variation of Mean Variance Weight and Balanced Weight (Q3)

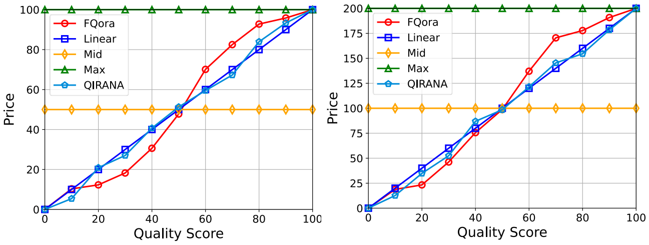
The third question is: How do the mean variance constraint and balanced Pareto optimization work to establish a market that is both fair and low-risk? We conduct the experiments for the mean variance weight **B** and the balanced weight **w** on CIFAR10 and TinyImageNet. As shown in Fig. 9, we can observe that the mean variance weight of different participants depends on the dataset because we shuffle the query of buyers. Nevertheless, the weights



(a) CIFAR10 (b) CIFAR100

(c) TinyImageNet (d) MNIST

Figure 7: The visual comparison of utility assignment under different datasets.



(a) CIFAR10 (b) CIFAR100

Figure 8: The comparison of price on (a) CIFAR10 and (b)

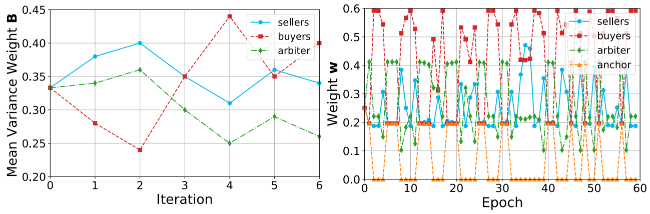
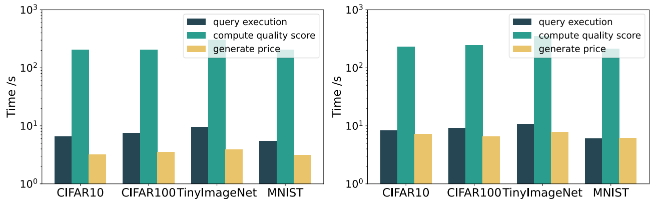
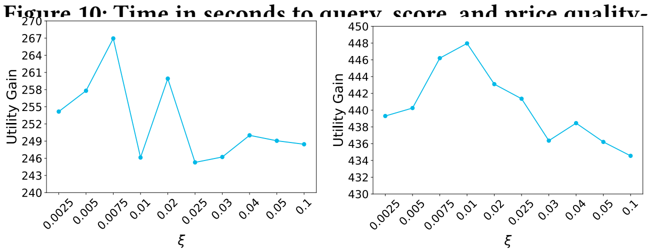


Figure 9: The variation of mean variance weight B and balanced weight w on TinyImageNet dataset.



(a) η = 0.01 (b) η = 0.03

Figure 10: Time in seconds to query, score, and price quality-



(a) CIFAR10 (b) TinyImageNet

Figure 11: Influence of different values of ξ

assigned to the participants exhibit minimal variation, consistently ranging from 20 to 40 percent. This observation underscores the low-risk development of the data market. It can be observed from Fig. 9 that the balanced weight **w** adaption process of our FQora. FQora can automatically learn the task weights without pre-defined heuristic constraints to balance the distribution of utilities among participants so that the market is not skewed in favor of one participant or another. In addition, the weight adaptation process of **w** is stable, and the search space is compact.

4.6 Efficiency (Q4)

In order to check the efficiency of our algorithm, we evaluate the execution time of FQora on all datasets. As shown in Fig. 10, we evaluate the runtime performance for three parts: query execution cost, computing cost for the quality scores of queries, and the time of pricing generation. Our experiments show that with all optimizations activated, we can price a query as fast (and most of the time faster) as computing the query on all datasets. However, we should note here that the time to compute the quality score of the queried data is also a time cost of FQora because computing HOG value of data is time-consuming for the whole algorithm.

4.7 Parameter Sensitivity (Q5)

To investigate the impact of using different values of ξ on the performance of our FQora, we conduct experiments on CIFAR10 and TinyImageNet, and the results are shown in Fig. 11. Noting that model with ξ = 0.0075 and ξ= 0.01 perform overall better than other values on these two datasets, respectively. In addition, FQora with a larger value of ξ yields unsatisfactory results for all participants in the data market on two datasets. One possible explanation is that the increased ξ causes the balanced update direction d to deviate significantly from the average utility g_a while adhering to the mean variance constraint outlined in Eq. (21). Consequently, as FQora identifies Pareto optimality that increasingly diverges from the balanced objective, it can have adverse effects on certain participants' utilities, resulting in a decline in overall balanced utility gain.

5 Related Work

In this section, we discuss related work on data market frameworks and multi-objective optimization.

Data Market Frameworks. In recent times, the issue of pricing data assets has gained significant scholarly attention. There are various frameworks for data pricing (see [13, 31] for the surveys). We divide all data pricing frameworks into two classes: query pricing and model pricing. In query pricing, a dataset has a fixed price. The challenge of query pricing is determining how to price relational queries on a given dataset without enabling arbitrage opportunities. Koutris et al. [18] show that the basic mathematical tool is query determinacy. Qirana [10] gives a more complex method with the maximization of sellers' revenue. Recent work [7] also considered how to maximize the broker's revenue under the same pricing model as above. The recent line of work [7, 10, 17, 18] has formally studied pricing schemes for assigning prices to relational queries. For model pricing, the goal of this scheme is to develop a technology to allow the purchase of ML models instead of training datasets in query pricing. [8] proposed a framework via a noise injection approach, which provably satisfies the desired arbitrage-free properties. FQora is more similar to query pricing, but they are fundamentally different. Buyers in FQora utilize a quality-modified query to get a quality-controlled dataset. Moreover, FQora introduces a balanced data market without only considering the maximization of seller revenue.

Multi-Objective Optimization. Multi-objective optimization (MOO) is important in operations research and computer science. It optimizes numerous competing goals simultaneously. MOO seeks Pareto optimum solutions. These solutions are efficient, meaning there is no better solution in the search space for all objectives. MOO problems can be solved with Evolutionary Algorithms (EAs) [34]. Multi-Objective Particle Swarm Optimization and Non-dominated Sorting Genetic Algorithm II [6] are also well-known examples. Recently, gradient balance approaches have been developed to reduce task conflicts and improve performance. Désideéri optimizes several objectives using MGDA [11]. Sener et al. [33] reframed the multi-objective problem as a multi-task problem. Through random selection, they seek a Pareto optimal solution. Mao et al. [25] introduced a Tchebycheff-based multi-task learning approach. These tactics aim for an arbitrary Pareto optimal answer without prioritizing it. In contrast, we present a highly successful approach to multi-objective optimization that aims to discover solutions that achieve both Pareto optimality and utility balancing.

6 Conclusion

In this paper, we proposed a novel quality-based data market FQora to address the quality-missing and unbalanced utility allocation challenges. Specifically, FQora empowers buyers to control data quality by attaching quality constraints on the query and ensures sustainable market growth and equitable utility distribution by incorporating the mean variance frontier and a Balanced Pareto Optimization algorithm. Moreover, we present a series of theoretical proofs to illustrate the effectiveness and superiority of our FQora. Experimental results on four datasets show that FQora achieves superior performance on quality-based data pricing and balanced utility allocation problems.

REFERENCES

[1] 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Official Journal of the European Union. https://gdpr-info.eu/ Accessed: 2022.08.28.

[2] 2020. Personal Information Protection Law of the People's Republic of China. Standing Committee of the National People's Congress. http://www.npc.gov.cn/npc/c30834/202108/f6adab14c7c14fef9c45a2f3b5f6e19a.shtml Accessed: 2022.08.28.

[3] Windows Azure. 2021. *Windows Azure Marketplace.* www.datamarket.azure.com

[4] Magdalena Balazinska, Bill Howe, and Dan Suciu. 2011. Data markets in the cloud: An opportunity for the database community. *Proceedings of the VLDB Endowment* 4, 12 (2011), 1482–1485.

[5] Dimitri P Bertsekas. 1997. Nonlinear programming. *Journal of the Operational Research Society* 48, 3 (1997), 334–334.

[6] Sankhadeep Chatterjee, Sarbartha Sarkar, Nilanjan Dey, Amira S Ashour, and Soumya Sen. 2018. Hybrid non-dominated sorting genetic algorithm: II-neural network approach. In *Advancements in Applied Metaheuristic Computing.* IGI Global, 264–286.

[7] Shuchi Chawla, Shaleen Deep, Paraschos Koutrisw, and Yifeng Teng. 2019. Revenue maximization for query pricing. *Proceedings of the VLDB Endowment* 13, 1 (2019), 1–14.

[8] 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.* 1535–1552.

[9] Dawex. 2020. *Data Exchange, reveal the power of your data ecosystem.* https://www.dawex.com/

[10] Shaleen Deep and Paraschos Koutris. 2017. QIRANA: A framework for scalable query pricing. In *Proceedings of the 2017 ACM International Conference on Management of Data.* 699–713.

[11] Jean-Antoine Désidéri. 2012. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique* 350, 5-6 (2012), 313–318.

[12] Eugene F Fama and Kenneth R French. 2016. Commodity futures prices: Some evidence on forecast power, premiums, and the theory of storage. In *The World Scientific Handbook of Futures Markets.* World Scientific, 79–102.

[13] Raul Castro Fernandez, Pranav Subramaniam, and Michael J Franklin. 2020. Data market platforms: trading data assets to solve data problems. *Proceedings of the VLDB Endowment* 13, 12 (2020), 1933–1947.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 770–778.

[15] Zhimeng Jiang, Kaixiong Zhou, Zirui Liu, Li Li, Rui Chen, Soo-Hyun Choi, and Xia Hu. 2021. An information fusion approach to learning with instance-dependent label noise. In *International Conference on Learning Representations.*

[16] Jürgen Kindler. 2005. A simple proof of Sion's minimax theorem. *The American Mathematical Monthly* 112, 4 (2005), 356–358.

[17] Paraschos Koutris, Prasang Upadhyaya, Magdalena Balazinska, Bill Howe, and Dan Suciu. 2012. Querymarket demonstration: Pricing for online data markets. *Proceedings of the VLDB Endowment* 5, 12 (2012), 1962–1965.

[18] Paraschos Koutris, Prasang Upadhyaya, Magdalena Balazinska, Bill Howe, and Dan Suciu. 2015. Query-based data pricing. *Journal of the ACM (JACM)* 62, 5 (2015), 1–44.

[19] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (2017), 84–90.

[21] Seong Min Kye, Kwanghee Choi, Joonyoung Yi, and Buru Chang. 2021. Learning with Noisy Labels by Efficient Transition Matrix Estimation to Combat Label Miscorrection. (11 2021). http://arxiv.org/abs/2111.14932

[22] Ya Le and Xuan Yang. 2015. Tiny imagenet visual recognition challenge. *CS 231N* 7, 7 (2015), 3.

[23] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[24] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. 2021. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems* 34 (2021), 18878–18890.

[25] Yuren Mao, Shuang Yun, Weiwei Liu, and Bo Du. 2020. Tchebycheff procedure for multi-task text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* 4217–4226.

[26] Harry M Markowits. 1952. Portfolio selection. *Journal of finance* 7, 1 (1952), 71–91.

[27] James B McDonald and Yexiao J Xu. 1995. A generalization of the beta distribution with applications. *Journal of Econometrics* 66, 1-2 (1995), 133–152.

[28] Meta. 2023. Llama 3: open source, free for research and commercial use. https://llama.meta.com/llama3/.

[29] Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. 2019. Confident Learning: Estimating Uncertainty in Dataset Labels. (10 2019). http://arxiv.org/abs/1911.00068

[30] OpenAI. 2023. GPT-4: OpenAI's Generative Pre-trained Transformer 4. https://openai.com/research/gpt-4.

[31] Jian Pei. 2020. A survey on data pricing: from economics to data science. *IEEE Transactions on knowledge and Data Engineering* 34, 10 (2020), 4586–4608.

[32] QLik. 2020. *QLik Data Market.* www.qlik.com/us/products/qlik-data-market

[33] Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems* 31 (2018).

[34] Lothar Thiele, Kaisa Miettinen, Pekka J Korhonen, and Julian Molina. 2009. A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary computation* 17, 3 (2009), 411–436.

[35] Reihaneh Torkzadehmahani, Reza Nasirigerdeh, Daniel Rueckert, and Georgios Kaissis. 2022. Label noise-robust learning using a confidence-based sieving

strategy.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[37] WorldQuant. 2020. *WorldQuant.* https://www.worldquant.com/

[38] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. 2020. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems* 33 (2020), 7597–7610.

[39] Chengzhen Xu, Kun Zhu, Changyan Yi, and Ran Wang. 2020. Data pricing for blockchain-based car sharing: A stackelberg game approach. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 1–5.

[40] Haipeng Yao, Tianle Mai, Jingjing Wang, Zhe Ji, Chunxiao Jiang, and Yi Qian. 2019. Resource trading in blockchain-based industrial Internet of Things. *IEEE Transactions on Industrial Informatics* 15, 6 (2019), 3602–3609.