Inferring Heterogeneous Private Valuations from Offline Market Data via Entropic Risk-Sensitive Utility Maximization

Xingyu Qian, Haoran Yu*

School of Computer Science & Technology, Beijing Institute of Technology
3120241000@bit.edu.cn, yhrhawk@gmail.com

Abstract

Inferring humans' private valuations for goods from their observed market behavior is essential for evaluating market efficiency and improving trading mechanism design. A core challenge lies in uncovering the human decision function that maps private valuations and observed market states to actions. In complex market settings where humans make sequential decisions in stochastic environments, neural networks offer the flexibility to model this decision function. However, training them without access to private valuations or environment dynamics remains challenging. We tackle this challenge and study how to infer heterogeneous human valuations from offline decision data in continuous double auctions. We propose learning the decision function via risk-sensitive utility maximization. First, we train a generative model on offline bid and ask data to simulate individual trading behavior. Using this generative model, we instantiate simulated markets composed of randomly generated buyers and sellers. Second, we introduce an agent into these simulated markets and use reinforcement learning to learn a risk-sensitive utility-maximizing decision function for the agent. Third, we formulate a bilevel optimization to jointly recover private valuations and risk preference parameters. Our extensive experiments on a large-scale continuous double auction dataset demonstrate that our framework significantly reduces errors in inferring real human valuations.

1 Introduction

Private valuation inference refers to estimating humans' private valuations for goods based on their strategic behavior in market settings (such as ascending auctions (Aradillas-Lopez, Gandhi, and Quint 2013), repeated auctions (Noti and Syrgkanis 2021), and sequential bargaining (Cui and Yu 2023)). Accurately inferring these valuations helps reveal human behavioral patterns in markets, evaluate the efficiency of trading mechanisms, and guide the optimization of these mechanisms.

A key step in private valuation inference is recovering the human decision function, which maps private valuations and the market states observed by humans to their decisions. In some simple market settings, this function is relatively straightforward to model. For example, in repeated auctions, each bidder submits a bid in every round, and the auction resets between rounds. It is common to assume that bidders follow a no-regret learning strategy and explicitly model their bidding function (Noti and Syrgkanis 2021).

In market settings such as sequential bargaining and continuous double auctions, each individual must forecast evolving market states and make decisions accordingly, which makes it more difficult to recover the human decision function. Given their flexibility in capturing complex patterns, neural networks can be used to model the human decision function in these market settings. One key challenge in training these neural networks is that human private valuations are unobservable, so standard supervised learning cannot be applied. Recent studies designed loss functions that impose economic constraints (e.g., humans never choose strictly dominated decisions) to guide neural network training (Cui and Yu 2023).

In this paper, we address heterogeneous private valuation inference in continuous double auctions (CDAs) and develop a novel approach for training the neural network that models the decision function without access to valuation data. Our key idea for guiding network training is to impose that each individual's decision function maximizes its utility under a specified risk measure, given its belief about market evolution. The dynamics of market states in CDAs are highly stochastic. Uncertainty about future states directly influences human decisions. By incorporating risk preference into the utility model, we can more accurately recover the underlying decision function.

Our valuation inference framework comprises three steps. First, we train a generative model on offline data of buyer bids and seller asks. The model predicts future bids and asks given the current market state, and thus serves as each individual's belief about market evolution. Second, we apply reinforcement learning to learn a decision function that maximizes utility under the entropic risk measure. During training, we simulate the CDA environment using the generative model obtained in the first step. Third, we formulate a bilevel optimization problem to infer each individual's private valuation and risk preference, leveraging the learned decision function alongside the observed behavioral data.

The main contributions of our work are as follows.

- **Novel Valuation Inference Framework:** We propose a framework that infers heterogeneous private valuations

*Corresponding author.

using only offline human behavioral data and applies to general market mechanisms. By learning the human decision function through risk-sensitive utility maximization, our principled approach offers richer structural guidance than existing methods, enabling more accurate learning of the human decision pattern.

- **Human Risk Preference Estimation:** We adopt an entropic risk-sensitive utility model and estimate each individual's risk preference parameter alongside its valuation. Accurate risk preference estimation can uncover human behavior patterns under uncertainty and also mitigate systematic bias in private valuation inference.
- **Evaluation Using Human Behavioral Data:** We evaluate the inference accuracy of our framework using a large dataset on real human behavior in CDA experiments. Compared to baseline methods, our framework achieves significantly lower inference error.

2 Related Work

2.1 Private Information Inference

To infer the private information of decision makers from their strategic behavior, a key challenge is recovering the decision function that governs their actions. One main approach, rooted in inverse game theory, assumes that decision makers are rational and their decisions satisfy equilibrium conditions. For example, (Bertsimas, Gupta, and Paschalidis 2015; Maddux et al. 2023) formulated the inference problem as identifying private information that minimizes deviations from Nash equilibrium conditions in the observed behavior. Beyond Nash equilibrium, alternative concepts, such as Bayesian Nash equilibrium (Larsen and Zhang 2018) and Quantal Response Equilibrium (Yu et al. 2022b; Wu et al. 2022), have been applied to account for incomplete information and bounded rationality in decision making. In stochastic settings involving private information and multi-round decisions, computing equilibria becomes intractable, rendering the inverse game theoretic approach difficult to apply.

Some studies have focused on repeated games, where the same static game is played in each round (Noti and Syrgkanis 2021; Zhang et al. 2025). A common approach is to assume a specific form for the decision function. For example, (Yu et al. 2022a) adopted logit-response dynamics, where each individual selects actions with probabilities proportional to their utilities computed from the previous round's outcomes. (Nisan and Noti 2017) assumed that each individual uses a no-regret learning strategy that minimizes its cumulative regret over rounds.

Neural networks have been employed in several valuation inference studies to model complex functional relationships. (Ling, Fang, and Kolter 2019) leveraged a neural network to model the mapping from observed individual features to private valuations. (Cui and Yu 2024) used a neural network to model the seller asking function in continuous double auctions. The study assumed that the statistical knowledge about the ask-cost ratio is known and focused on using this knowledge to guide network training. In contrast, our work does not assume the availability of this statistical knowledge. We propose a fundamentally different paradigm and train the network via risk-sensitive utility maximization.

2.2 Inverse Reinforcement Learning

Inverse reinforcement learning aims to recover a decision maker's reward function by assuming that it follows a stochastic reward-maximizing policy (Adams, Cody, and Beling 2022). One classic approach is maximum entropy inverse reinforcement learning (Ziebart et al. 2008; Boularias, Kober, and Peters 2011). When the reward function is linear in a feature vector, the approach essentially finds the reward weights under which the induced policy maximizes the entropy while the expected features under the policy match the observed features. More recently, (Zeng et al. 2022; Liu and Zhu 2024; Zeng, Hong, and Garcia 2025) have investigated the maximum likelihood inverse reinforcement learning approach. It directly seeks reward parameters where the induced policy maximizes the likelihood of the observed actions.

Our work differs from the above studies in the following aspects. First, we focus on inferring the heterogeneous private valuations of hundreds of individual participants in continuous double auctions. Inverse reinforcement learning estimates a reward function for a single decision maker. Directly applying it to our problem requires training hundreds of separate policies, which is computationally infeasible. Second, we incorporate the entropic risk measure when modeling the participant maximization objective and study the joint inference of private valuations and risk preferences. Third, we address the setting where only offline behavioral data are available, whereas the above studies require access to the underlying stochastic environment.

3 Preliminaries

3.1 Continuous Double Auctions

In a Continuous Double Auction (CDA), multiple buyers and sellers submit bids and asks to trade discrete units of a homogeneous commodity. The trading process evolves over multiple time slots. In each time slot, buyers may submit bids indicating their willingness to buy one unit of the commodity at specified prices, while sellers may submit asks indicating their willingness to sell. All submitted bids and asks are publicly visible to market participants. There is a cap of K units on the total number that each buyer or seller may trade in the market.

Buyers and sellers may freely adjust their bids and asks. Transactions occur immediately when a buyer's bid meets a seller's ask. The market closes once the maximum number of time slots is reached.

Figure 1 illustrates a CDA. Blue blocks denote seller asks, and green blocks denote buyer bids. Purple dotted lines indicate price adjustments made by sellers and buyers. Two transactions occur in the illustrated example.

3.2 Buyer Valuation Inference Problem

To simplify the presentation, this paper focuses on inferring buyer valuations. The seller valuation inference problem can be defined and solved in a similar way.

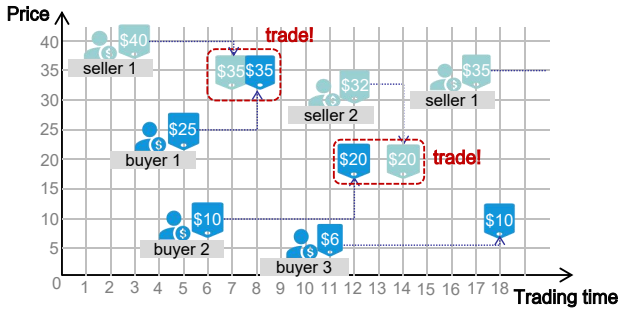

Figure 1: Illustration of a Continuous Double Auction.

Let $\boldsymbol{v}_n = (v_{n1}, v_{n2}, \ldots, v_{nK})$ denote the private valuation vector of buyer n, where v_{nk} is the highest price that the buyer is willing to pay for the k-th unit.

Let $\mathcal{N}$ be the set of buyer indices across all markets. The buyer private valuation inference problem aims to infer $\{\boldsymbol{v}_n\}_{n\in\mathcal{N}}$, given the complete records of buyer bids and seller asks across all markets.

3.3 Entropic Risk Measure

The entropic risk measure is widely used to model the attitudes of decision makers toward risk in various trading environments (Follmer and Knispel 2011; Nass, Belousov, and Peters 2019). Let R denote a random return whose distribution is parameterized by $\boldsymbol{\psi}$. Under the entropic risk measure, a decision maker evaluates outcomes using the following function:

$$U(\boldsymbol{\psi}, \gamma) = -\frac{1}{\gamma} \log \mathbb{E}_{R\sim p_{\boldsymbol{\psi}}} \left[e^{-\gamma R}\right]. \tag{1}$$

Here, the parameter γ captures the risk preference. When $\gamma > 0$, the decision maker is risk-averse and prefers returns with lower variance; when $\gamma < 0$, it is risk-seeking. As $\gamma \to 0$, the decision maker becomes risk-neutral and $U(\boldsymbol{\psi}, \gamma)$ converges to the expected return $\mathbb{E}[R]$, which has been widely used in economic modeling (Luo et al. 2018; Luo and Jennings 2020).

The well-known mean–variance risk measure can be regarded as a simplification of the entropic risk measure. To see this, we can apply a Taylor expansion to the logarithmic and exponential functions in (1) and obtain

$$U(\boldsymbol{\psi}, \gamma) \approx \mathbb{E}[R] - \frac{\gamma}{2}\mathrm{Var}[R], \tag{2}$$

where $\mathrm{Var}[R] = \mathbb{E}\left[R^2\right] - (\mathbb{E}[R])^2$ is the variance of R. When $\gamma > 0$, the utility decreases with the variance, implying that the decision maker is risk-averse.

4 Entropic Risk-Sensitive Private Valuation Inference Framework

4.1 Buyer Bidding Strategy

In this subsection, we define the buyer bidding strategy and introduce the key idea of our valuation inference framework.

Bidding Strategy Definition Let b_n^t denote the bid of buyer n at time slot t. It is drawn from a discrete set of monetary values (e.g., integer dollar amounts). In the example in Figure 1, buyer 1 submits a bid of \$25 at $t = 5$ and increases the bid to \$35 at $t = 8$. The values of b_1^t for $t = 0, 1, \ldots, 8$ are $0, 0, 0, 0, 0, 25, 25, 25, 35$. Here, $b_1^t = 0$ indicates that the buyer does not submit a valid bid at time t, reflecting an unwillingness to pay any price.

Buyer n's choice of b_n^t depends on both its private valuation $\boldsymbol{v}_n$ and its observation $\boldsymbol{o}_n^t$. Specifically, the observation $\boldsymbol{o}_n^t$ includes (i) the buyer's own historical bids prior to time t, (ii) all open bids from other buyers and all open asks, and (iii) the transaction prices of all historical trades.

We use π to denote the buyer bidding strategy, which defines a probability distribution over possible actions. Given the private valuation $\boldsymbol{v}_n$ and observation $\boldsymbol{o}_n^t$, buyer n chooses b_n^t by sampling according to $\pi(b_n^t | \boldsymbol{v}_n, \boldsymbol{o}_n^t)$.

Idea of Valuation Inference Given the knowledge of b_n^t and $\boldsymbol{o}_n^t$ across t, we can infer $\boldsymbol{v}_n$ once we obtain the bidding strategy π. Our key idea for obtaining π is to assume that buyer n follows a bidding strategy that maximizes its utility under the entropic risk measure in the stochastic CDA environment.

As illustrated in Figure 2, our Entropic Risk-Sensitive Valuation Inference (**ERS-VI**) framework consists of three steps. In Section 4.2, we use offline data on buyer bids and seller asks to learn the CDA environment, characterizing the dynamics of buyer and seller decisions. In Section 4.3, we train the bidding strategy π so that it maximizes a bidding agent's utility in the learned CDA environment. In Section 4.4, we use the learned bidding strategy π, along with the buyer bids and observations, to infer each buyer's valuation vector.

4.2 Learning CDA Environment from Data

In this subsection, we construct a simulator for the CDA environment. Using the observed buyer bids and seller asks in the offline dataset, we employ supervised learning to train a generative model, which simulates the behavior of each buyer and seller. For example, to simulate a buyer, the generative model takes the observation $\boldsymbol{o}_n^t$ as input and predicts the bid b_n^t.[1] Once trained, we use this generative model to simulate the CDA environment. In Section 4.3, we use the simulated environment to train a utility-maximizing bidding strategy.

Generative Model The generative model's input features include the individual's role (buyer or seller), the individual's historical behavior prior to time t, all open bids and asks, and historical transaction prices. We process these features in two stages: **(i) sequence processing:** we feed the time-series features (e.g., historical behavior) into an LSTM to generate a fixed-length embedding; **(ii) feature fusion:** we concatenate the embedding with the remaining features to form a fused representation and propagate it through three

[1] This generative model differs from the buyer bidding strategy $\pi(b_n^t | \boldsymbol{v}_n, \boldsymbol{o}_n^t)$ defined above, as the generative model does not require the unobserved valuation $\boldsymbol{v}_n$ as input.

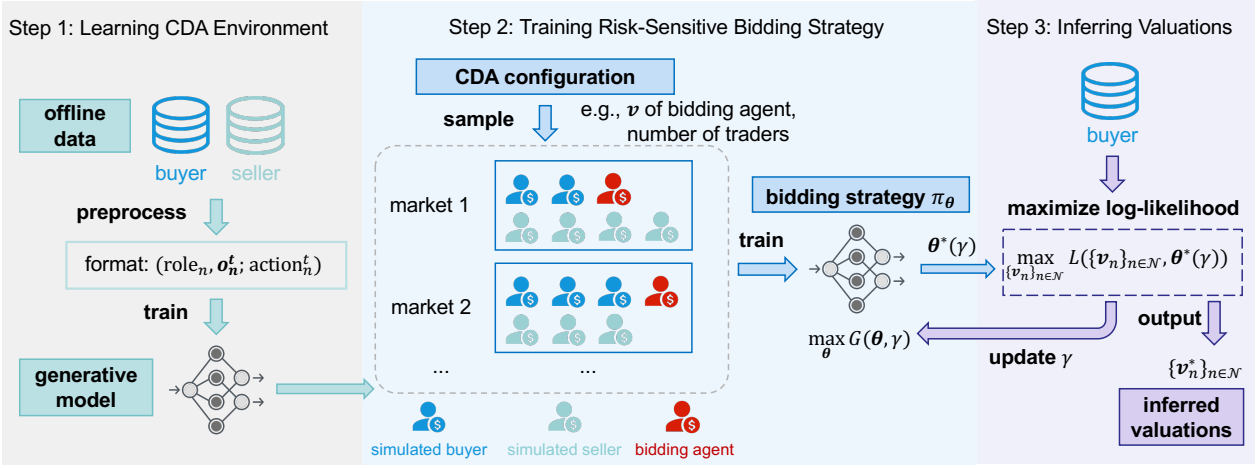

Figure 2: Our Entropic Risk-Sensitive Valuation Inference (**ERS-VI**) Framework.

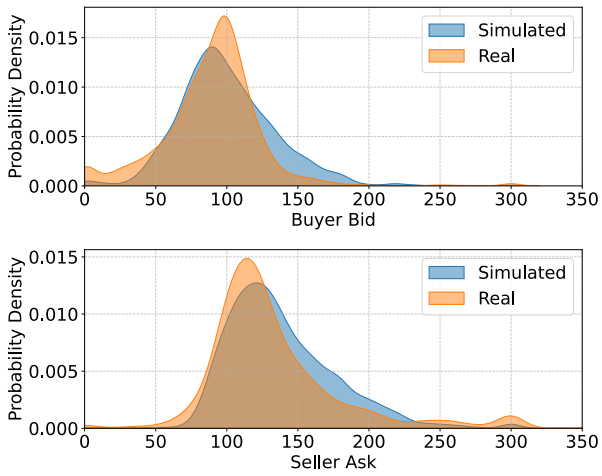

Figure 3: Comparison of Simulated and Real Behavior.

fully connected layers. The generative model outputs the predicted next action of the individual.

We train the generative model on a large-scale CDA dataset containing over 9,000 experimental double auction sessions (Lin et al. 2020) (more details will be introduced in Section 5). After preprocessing, we obtain 6,121,902 samples of individual behavior, compatible with the generative model's input-output format. We partition these samples into training (80%), validation (10%), and testing (10%) subsets. We determine the hyperparameters through a grid search that minimizes the validation loss.

Simulation Performance We demonstrate the effectiveness of the trained generative model in simulating CDA environments. For each CDA session in the dataset, we record the number of buyers and sellers, along with their bids and asks over the first 5 time slots, and use this information to initialize the simulator. Starting from $t = 5$, we simulate buyer bids and seller asks autoregressively until the end of the session. In each simulation, we apply the generative model separately for each individual at every time slot to generate its simulated bid or ask. We perform this simulation for different CDA sessions in the dataset. Finally, we aggregate all simulated bids and asks, and estimate their probability density functions using kernel density estimation.

In Figure 3, blue curves show the estimated densities of simulated buyer bids and seller asks. For comparison, orange curves show the densities estimated from real human behavior in the original dataset. The close alignment between the blue and orange curves demonstrates that our simulator reproduces real human behavior with high fidelity. Achieving this level of alignment is non-trivial, because small prediction errors in early time slots can compound over time. Despite this challenge, our generative model maintains stability and accuracy throughout the simulation, confirming its suitability for simulating CDA environments and functioning as an interactive simulator.

4.3 Training Risk-Sensitive Bidding Strategy

Bidding Agent in Simulated CDAs In this subsection, we train a bidding agent via reinforcement learning in the CDA environment simulated by the generative model. The agent's policy is modeled by a $\boldsymbol{\theta}$-parameterized neural network and denoted as $\pi_{\boldsymbol{\theta}}$. We aim to optimize $\boldsymbol{\theta}$ so that the policy maximizes the agent's utility given its private valuation $\boldsymbol{v}$.

We next detail the bidding agent's interaction with the simulated CDA environment. At the beginning of each simulated CDA session, we assign the agent a randomly sampled valuation $\boldsymbol{v}$ and randomly set the numbers of simulated buyers and sellers in the market. At each time slot t, the agent observes $\boldsymbol{o}^t$ and samples an action b^t according to $\pi_{\boldsymbol{\theta}}(b^t|\boldsymbol{v}, \boldsymbol{o}^t)$. Meanwhile, the generative model predicts the action of each simulated buyer or seller. The agent receives an immediate reward r^t and the next observation $\boldsymbol{o}^{t+1}$. This cycle repeats until the end of the session. During each time slot t, the agent records the experience tuple $(\boldsymbol{o}^t, b^t, r^t, \boldsymbol{o}^{t+1})$, which will be used to update its policy $\pi_{\boldsymbol{\theta}}$.

Immediate Reward Design The agent's immediate reward r^t depends on its valuation, bid, and the market state. If the agent successfully buys one unit, its reward equals its valuation minus the transaction price. Because the trading rule allows the agent to buy at most K units per session, the agent fails to trade in most time slots, resulting in sparse learning signals that hinder policy training.

To encourage more active exploration, we introduce a small bonus reward during non-trading slots. Specifically, whenever the lowest seller ask falls below the agent's valuation and the agent bids above a predefined threshold, we award a minor positive reward. Likewise, we impose a small penalty if the agent stays inactive for several consecutive time slots. The denser reward signals can accelerate policy convergence during training.

Risk-Sensitive Utility Maximization Let T denote the session length and $\tau = \left(\boldsymbol{o}^0, b^0, \ldots, \boldsymbol{o}^{T-1}, b^{T-1}, \boldsymbol{o}^T\right)$ denote a trajectory. We define the discounted cumulative reward of the trajectory τ as $R(\tau) = \sum_{t=0}^{T-1} \alpha^t r^t$, where α is the discount factor. Based on the entropic risk measure introduced in Section 3.3, we formulate the bidding agent's objective as maximizing the following utility function:

$$U(\boldsymbol{\theta}, \gamma) = -\frac{1}{\gamma} \log \mathbb{E}_{\tau \sim \pi_{\boldsymbol{\theta}}} \left[e^{-\gamma R(\tau)} \right]. \tag{3}$$

The parameter γ models the risk preference. When $\gamma > 0$, the agent is conservative and tends to submit higher bids to secure trades; when $\gamma < 0$, the agent is aggressive and tends to submit lower bids to chase higher rewards.

We apply the policy gradient method to maximize the utility. The gradient of the utility function with respect to $\boldsymbol{\theta}$ can be derived as follows:

$$\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}, \gamma) = -\frac{\mathbb{E}\left[\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}\left(b^t | \boldsymbol{v}, \boldsymbol{o}^t\right) e^{-\gamma R(\tau)}\right]}{\gamma \mathbb{E}\left[e^{-\gamma R(\tau)}\right]}. \tag{4}$$

In the implementation, we approximate the expectations in the above expression via the Monte Carlo method.

Some studies have applied value-based reinforcement learning methods to maximize the utility in (3) (Fei et al. 2020; Hau, Petrik, and Ghavamzadeh 2023). These methods require solving nonlinear Bellman equations with a log–exp structure, which is challenging in non-tabular Markov decision processes, as in our high-dimensional setting.

Training Objective Design We train the bidding agent by maximizing an objective function that combines the utility function with two regularization terms. First, we use $\{p_j\}_{j=1}^J$ to denote the feasible set of b^t, where p_j is the j-th price value. We introduce the following smoothness regularizer to encourage a smooth bidding strategy:

$$S(\boldsymbol{\theta}) = \mathbb{E}\left[\sum_{t=0}^{T-1} \sum_{j=1}^{J-1} \left(\pi_{\boldsymbol{\theta}}(p_j | \boldsymbol{v}, \boldsymbol{o}^t) - \pi_{\boldsymbol{\theta}}(p_{j+1} | \boldsymbol{v}, \boldsymbol{o}^t)\right)^2\right],$$

where the expectation is over the randomness of all $\boldsymbol{o}^t$. Minimizing the above regularizer penalizes large probability jumps between adjacent prices. As a result, the trained bidding strategy can better mimic real human behavior and thereby improve the accuracy of private valuation inference.

Second, we introduce the following entropy regularizer:

$$H(\boldsymbol{\theta}) = -\mathbb{E}\left[\sum_{t=0}^{T-1} \sum_{j=1}^{J} \pi_{\boldsymbol{\theta}}(p_j | \boldsymbol{v}, \boldsymbol{o}^t) \log \pi_{\boldsymbol{\theta}}(p_j | \boldsymbol{v}, \boldsymbol{o}^t)\right].$$

It measures the randomness in the agent's bid distribution over the set $\{p_j\}_{j=1}^J$. By increasing the entropy, we encourage adequate exploration of actions and prevent the bidding policy from collapsing into a deterministic policy too early during training.

Note that including an entropy regularizer is distinct from maximizing the utility with an entropic risk measure. According to (Haarnoja et al. 2017), an entropy regularizer raises the policy entropy at every timestep and therefore promotes per-step exploration. In contrast, (3) concerns the risk sensitivity at the trajectory level.

We define the training objective function as

$$G(\boldsymbol{\theta}, \gamma) = U(\boldsymbol{\theta}, \gamma) - \lambda_1 S(\boldsymbol{\theta}) + \lambda_2 H(\boldsymbol{\theta}), \tag{5}$$

where λ_1 and λ_2 are positive tunable hyperparameters that weight the smoothness and entropy regularizers. We maximize the objective using the Adam optimizer, where the gradient of the utility term is given in (4). We use $\boldsymbol{\theta}^*(\gamma)$ to denote the optimal policy parameters given the risk preference parameter γ.

In our implementation, the bidding policy network $\pi_{\boldsymbol{\theta}}(b^t | \boldsymbol{v}, \boldsymbol{o}^t)$ adopts a two-stage architecture similar to that of the generative model. It first encodes time-series features via an LSTM and then fuses them with the remaining features through fully connected layers. To accommodate the bidder's private valuation $\boldsymbol{v}$ (which is not an input to the generative model), we simply expand the fused feature vector with $\boldsymbol{v}$. During training, at the beginning of each simulated session, we randomly sample $\boldsymbol{v}$ from the full range of possible valuations. This ensures that the policy learns to bid effectively for any valuation in its support.

4.4 Inferring Risk Preferences and Valuations

Recall that our goal is to infer $\{\boldsymbol{v}_n\}_{n \in \mathcal{N}}$, i.e., the valuations of all buyers. To measure the consistency between these inferred valuations, the learned policy parameters $\boldsymbol{\theta}$, and the observed bidding actions, we use the following log-likelihood function:

$$L\left(\{\boldsymbol{v}_n\}_{n \in \mathcal{N}}, \boldsymbol{\theta}\right) = \sum_{n \in \mathcal{N}} \sum_{t=0}^{T-1} \log \pi_{\boldsymbol{\theta}}\left(b_n^t | \boldsymbol{v}_n, \boldsymbol{o}_n^t\right). \tag{6}$$

The log-likelihood is large when the policy $\pi_{\boldsymbol{\theta}}$ assigns a high probability to buyer n's actual bid b_n^t given the inferred valuation $\boldsymbol{v}_n$ and observation $\boldsymbol{o}_n^t$.

As described in Section 4.3, for a given risk preference γ, the optimal policy parameters $\boldsymbol{\theta}^*(\gamma)$ are obtained by maximizing $G(\boldsymbol{\theta}, \gamma)$ in (5). Therefore, we formulate the joint inference of $\{\boldsymbol{v}_n\}_{n \in \mathcal{N}}$ and γ as follows:

$$\max \; L\left(\{\boldsymbol{v}_n\}_{n \in \mathcal{N}}, \boldsymbol{\theta}^*(\gamma)\right) \tag{7}$$

$$\text{s.t.} \quad \boldsymbol{\theta}^*(\gamma) \in \arg\max_{\boldsymbol{\theta}} G(\boldsymbol{\theta}, \gamma), \tag{8}$$

$$\text{var.} \; \gamma, \{\boldsymbol{v}_n\}_{n \in \mathcal{N}}. \tag{9}$$

Algorithm 1: Entropic Risk-Sensitive Valuation Inference

Input: Offline datasets of buyer bids and seller asks, and hyperparameters, such as $\alpha, \lambda_1, \lambda_2$, and M.
Output: A set of inferred valuations, i.e., $\{\boldsymbol{v}_n^*\}_{n\in\mathcal{N}}$.

1: Train a generative model that predicts a buyer's bid or a seller's ask given the market observation.
2: Choose γ^0 randomly and set $m = 0$.
3: **for** $m \leq M$ **do**
4: **if** $m = 0$ **then**
5: Train a bidding agent's policy network from scratch in simulated markets to obtain $\boldsymbol{\theta}^*(\gamma^m)$, i.e., the vector maximizing the objective in (5).
6: **else**
7: Identify $\tilde{\gamma} \in \{\gamma^0, \ldots, \gamma^{m-1}\}$ that is closest to γ^m.
8: Initialize the policy network using $\boldsymbol{\theta}^*(\tilde{\gamma})$ and fine-tune it to obtain $\boldsymbol{\theta}^*(\gamma^m)$.
9: **end if**
10: Use gradient ascent to compute $\{\boldsymbol{v}_n^*\}_{n\in\mathcal{N}}$, i.e., the valuations maximizing the log-likelihood in (6), and record the resulting optimal log-likelihood as l^m.
11: Select γ^{m+1} based on $\left(\gamma^0, l^0\right), \ldots, \left(\gamma^m, l^m\right)$.
12: $m \leftarrow m + 1$.
13: **end for**
14: **return** $\{\boldsymbol{v}_n^*\}_{n\in\mathcal{N}}$ computed in the final iteration.

It is a bilevel optimization problem. For a fixed γ, it is straightforward to maximize the log-likelihood function over $\{\boldsymbol{v}_n\}_{n\in\mathcal{N}}$ via gradient ascent. However, it is challenging to maximize the log-likelihood function over γ using gradient-based methods. Based on the chain rule, we can derive the following equation:

$$\frac{dL\left(\{\boldsymbol{v}_n\}_{n\in\mathcal{N}}, \boldsymbol{\theta}^*(\gamma)\right)}{d\gamma} = \frac{\partial L\left(\{\boldsymbol{v}_n\}_{n\in\mathcal{N}}, \boldsymbol{\theta}\right)}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*(\gamma)} \cdot \frac{d\boldsymbol{\theta}^*(\gamma)}{d\gamma}.$$

Since computing $\boldsymbol{\theta}^*(\gamma)$ involves solving a complex inner maximization problem, computing $\frac{d\boldsymbol{\theta}^*(\gamma)}{d\gamma}$ is infeasible.

We tackle the bilevel problem using a two-stage optimization scheme. In the outer loop, we apply a gradient-free method (e.g., Bayesian optimization) to search for γ. In the inner loop, we update $\{\boldsymbol{v}_n\}_{n\in\mathcal{N}}$ via gradient ascent.

Algorithm 1 outlines the complete inference procedure. In line 3, M denotes the number of γ values to evaluate. To accelerate repeated training of the bidding agent (in line 8), we warm-start each training by fine-tuning the policy network from the parameters obtained in previous iterations.

5 Experiments

5.1 Experimental Setup

The dataset comes from CDA experiments conducted on the MobLab platform (Lin et al. 2020). The dataset records complete bidding and asking behavior of participants over 9,000 CDA markets. In each market, the experimental designer assigned a private valuation to every participant, providing ground-truth data for evaluating valuation inference methods. The number of units that each buyer or seller could trade ranged from 1 to 3 across different markets.

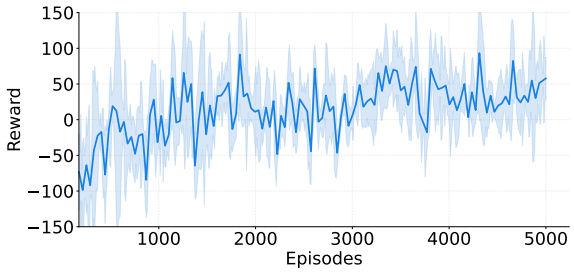

Figure 4: Evolution of Bidding Policy During Training.

We evaluate the inference accuracy of our **ERS-VI** framework on the dataset and compare it with the following baseline methods:

- **CE**: It addresses the inference problem by using supervised learning to train a decision function that takes both the valuation and observation as inputs. Because true valuations are unavailable at training time, they are substituted with values sampled from their feasible range.
- **BLUE** (Cui and Yu 2023): It extends **CE** by incorporating a rationality constraint during training, penalizing decision functions that select strictly dominated actions.
- **SL**: It learns a direct mapping from buyer observations and bids to valuations. To guide training, it enforces that predicted valuations lie below the corresponding bids.
- **DL** (He et al. 2016): Unlike **SL**, it employs a dual learning framework that alternately tackles two interconnected tasks: (i) valuation inference, which estimates valuations from observations and bids, and (ii) behavior prediction, which forecasts bids from valuations and observations. It lets each task mutually constrain the other through iterative network training.
- **PGM** (Pirotta and Restelli 2016): It searches for the valuation that minimizes the norm of the bidding policy's gradient. This is motivated by the principle that the optimal bidding policy has a zero gradient when it takes the true valuation as input.
- **Mean**: It estimates valuations by computing the mean of all feasible valuations (defined as the values below the bids). This method does not require any network training.
- **Random**: It estimates valuations by uniformly sampling from all feasible valuations.
- **RvS** (Emmons et al. 2022): It first trains a supervised decision function that maps immediate reward, valuation, and observation to an action. To infer the valuation from observed observations and actions, it samples a positive immediate reward and then searches for the valuation that maximizes the likelihood of the given actions under the trained decision function.

The pseudocode of these methods, codes, and data are available at: https://github.com/qianxingyuyuyu/CDAinfer.

5.2 Experimental Results

Training of Bidding Policy Figure 4 illustrates the bidding policy's training process during the first iteration of our

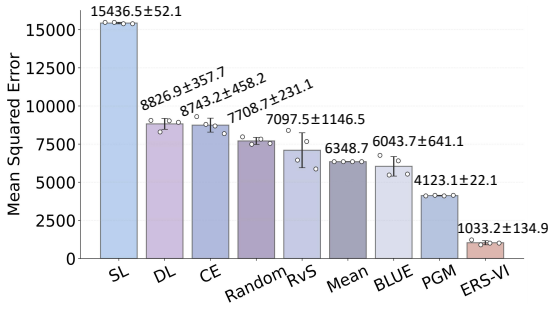

Figure 5: Mean Squared Errors of Different Methods.

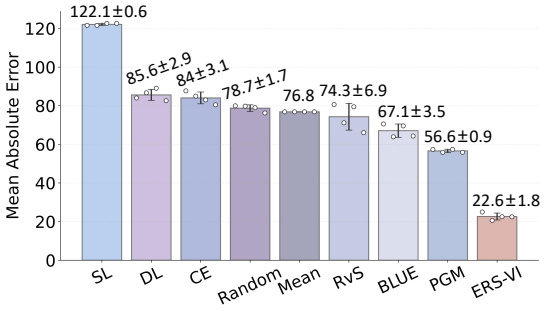

Figure 6: Mean Absolute Errors of Different Methods.

framework. The plot shows discounted cumulative reward versus episode count, averaged over 4 runs (the solid line denotes the mean and the shaded region denotes the standard deviation). In the early episodes, rewards are negative because the agent may bid above its valuation. As training progresses, rewards generally increase despite fluctuations in the curve. These fluctuations arise from our design to train a single agent that can adapt to each sampled valuation vector $\boldsymbol{v}$. Since different episodes sample different $\boldsymbol{v}$ values, episodes with lower valuations yield smaller rewards.

Comparison with Baselines We next compare the inference performance of different methods. We first evaluate each method using the mean squared error (MSE) between inferred and true private valuations. We run each method 4 times and show the average MSE and its standard deviation in Figure 5. Each white circle represents the MSE under an individual experimental run.

Inference methods are displayed from left to right in descending order of MSE. Our **ERS-VI** achieves the lowest average MSE (1033.2) and significantly outperforms all baseline methods. A key insight from Figure 5 is that imposing appropriate structural constraints consistently enhances inference accuracy. For example, **DL** outperforms **SL** by introducing an auxiliary task that guides the learning process. **BLUE** improves upon **CE** by enforcing a rationality constraint on the decision function learning process. Among the baselines, **PGM** achieves the best result. It assumes that the optimal bidding policy exhibits a zero policy gradient, which is similar to the utility maximization principle applied by our **ERS-VI** framework.

In Figure 6, we plot the mean absolute errors (MAEs) of different inference methods. The ordering is consistent with that of the MSE results, and our **ERS-VI** substantially outperforms all baselines.

Impact of Risk Preference Estimation We assess how estimating risk preference affects valuation inference. In Figure 7, we compare the inference results under two values of the risk preference parameter γ. We train a bidding agent for $\gamma = -0.025$ and $\gamma = 0.025$. Then, we use these bidding agents to infer buyer valuations. For ease of visualization, we randomly sample 300 valuations and plot them in the left ($\gamma = -0.025$) and right ($\gamma = 0.025$) panels. Each point represents a valuation, with its horizontal coordinate showing the true value and its vertical coordinate showing

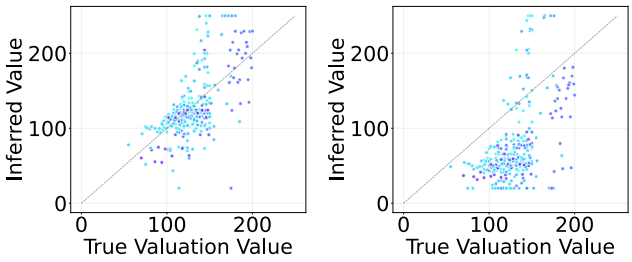

Figure 7: Sampled Inference Results Under Different γ Values (Left: $\gamma = -0.025$; Right: $\gamma = 0.025$).

the inferred value. Points near the 45-degree line indicate accurate inference, as the inferred values are close to the true values. Each point is assigned a color ranging from light blue to dark purple, where points with the same color across the two panels correspond to the same valuation being inferred.

Figure 7 demonstrates that inference under $\gamma = -0.025$ is more accurate, suggesting that a risk-seeking bidding agent better captures actual buyer behavior. In the right panel ($\gamma = 0.025$), most points fall below the 45-degree line, indicating a systematic underestimation of valuations. The reason for the underestimation is as follows. When $\gamma = 0.025$, the bidding agent is risk-averse and tends to place bids close to its true valuation to secure the purchase. For example, a risk-averse agent with a true valuation of 100 may bid 90. In contrast, an actual buyer with a true valuation of 200 is risk-seeking and may also bid 90. If we infer valuations using $\gamma = 0.025$, a bid of 90 is interpreted as corresponding to a valuation of 100, thus underestimating the actual valuation of 200. The results in Figure 7 indicate the importance of estimating risk preference while inferring private valuations.

6 Conclusion

In this paper, we proposed a three-step framework for inferring heterogeneous private valuations via entropic risk-sensitive utility maximization. Experiments on a large-scale dataset demonstrate that our framework yields more accurate inferred valuations than baseline methods, including the method that learns the bidding function by enforcing a rationality constraint instead of utility maximization. Our framework is not limited to continuous double auctions and can be applied to infer heterogeneous valuations from offline data in general stochastic market environments.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62572049 and No. 62202050) and the Beijing Institute of Technology Research Fund Program for Young Scholars. We gratefully acknowledge the authors of (Lin et al. 2020) for releasing the CDA dataset to the public.

References

Adams, S.; Cody, T.; and Beling, P. A. 2022. A survey of inverse reinforcement learning. *Artificial Intelligence Review*, 55(6): 4307–4346.

Aradillas-Lopez, A.; Gandhi, A.; and Quint, D. 2013. Identification and inference in ascending auctions with correlated private values. *Econometrica*, 81(2): 489–534.

Bertsimas, D.; Gupta, V.; and Paschalidis, I. C. 2015. Data-driven estimation in equilibrium using inverse optimization. *Mathematical Programming*, 153: 595–633.

Boularias, A.; Kober, J.; and Peters, J. 2011. Relative entropy inverse reinforcement learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 182–189.

Cui, L.; and Yu, H. 2023. Inferring private valuations from behavioral data in bilateral sequential bargaining. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 2624–2632.

Cui, L.; and Yu, H. 2024. Data-driven knowledge-aware inference of private information in continuous double auctions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 10012–10020.

Emmons, S.; Eysenbach, B.; Kostrikov, I.; and Levine, S. 2022. RvS: What is essential for offline RL via supervised learning? In *Proceedings of the International Conference on Learning Representations*.

Fei, Y.; Yang, Z.; Chen, Y.; Wang, Z.; and Xie, Q. 2020. Risk-sensitive reinforcement learning: Near-optimal risk-sample tradeoff in regret. *Advances in Neural Information Processing Systems*, 33: 22384–22395.

Follmer, H.; and Knispel, T. 2011. Entropic risk measures: Coherence vs. convexity, model ambiguity and robust large deviations. *Stochastics and Dynamics*, 11(2/3): 333–351.

Haarnoja, T.; Tang, H.; Abbeel, P.; and Levine, S. 2017. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, 1352–1361.

Hau, J. L.; Petrik, M.; and Ghavamzadeh, M. 2023. Entropic risk optimization in discounted MDPs. In *International Conference on Artificial Intelligence and Statistics*, 47–76.

He, D.; Xia, Y.; Qin, T.; Wang, L.; Yu, N.; Liu, T.-Y.; and Ma, W.-Y. 2016. Dual learning for machine translation. *Advances in neural information processing systems*, 29.

Larsen, B.; and Zhang, A. L. 2018. A mechanism design approach to identification and estimation. Technical report, National Bureau of Economic Research.

Lin, P.-H.; Brown, A. L.; Imai, T.; Wang, J. T.-y.; Wang, S. W.; and Camerer, C. F. 2020. Evidence of general economic principles of bargaining and trade from 2,000 classroom experiments. *Nature Human Behaviour*, 4(9): 917–927.

Ling, C. K.; Fang, F.; and Kolter, J. Z. 2019. Large scale learning of agent rationality in two-player zero-sum games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6104–6111.

Liu, S.; and Zhu, M. 2024. In-trajectory inverse reinforcement learning: Learn incrementally before an ongoing trajectory terminates. *Advances in Neural Information Processing Systems*, 37: 117164–117209.

Luo, Y.; and Jennings, N. R. 2020. A differential privacy mechanism that accounts for network effects for crowdsourcing systems. *Journal of Artificial Intelligence Research*, 69: 1127–1164.

Luo, Y.; Shah, N. B.; Huang, J.; and Walrand, J. 2018. Parametric prediction from parametric agents. *Operations Research*, 66(2): 313–326.

Maddux, A. M.; Pagan, N.; Belgioioso, G.; and Dorfler, F. 2023. Data-driven behaviour estimation in parametric games. *IFAC-PapersOnLine*, 56(2): 9330–9335.

Nass, D.; Belousov, B.; and Peters, J. 2019. Entropic risk measure in policy search. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1101–1106.

Nisan, N.; and Noti, G. 2017. An experimental evaluation of regret-based econometrics. In *Proceedings of the 26th International Conference on World Wide Web*, 73–81.

Noti, G.; and Syrgkanis, V. 2021. Bid prediction in repeated auctions with learning. In *Proceedings of the Web Conference*, 3953–3964.

Pirotta, M.; and Restelli, M. 2016. Inverse reinforcement learning through policy gradient minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Wu, J.; Shen, W.; Fang, F.; and Xu, H. 2022. Inverse game theory for stackelberg games: The blessing of bounded rationality. *Advances in Neural Information Processing Systems*, 35: 32186–32198.

Yu, S.; Brantingham, P. J.; Valasik, M.; and Vorobeychik, Y. 2022a. Learning binary multi-scale games on networks. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, 2310–2319.

Yu, Y.; Salfity, J.; Fridovich-Keil, D.; and Topcu, U. 2022b. Inverse matrix games with unique quantal response equilibrium. *IEEE Control Systems Letters*, 7: 643–648.

Zeng, S.; Hong, M.; and Garcia, A. 2025. Structural estimation of markov decision processes in high-dimensional state space with finite-time guarantees. *Operations Research*, 73(2): 720–737.

Zeng, S.; Li, C.; Garcia, A.; and Hong, M. 2022. Maximum-likelihood inverse reinforcement learning with finite-time guarantees. *Advances in Neural Information Processing Systems*, 35: 10122–10135.

Zhang, B. H.; Lin, T.; Chen, Y.; and Sandholm, T. 2025. Learning a Game by Paying the Agents. *arXiv preprint arXiv:2503.01976*.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; Dey, A. K.; et al. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 8, 1433–1438.