PRICES, BIDS, VALUES: EVERYTHING, EVERYWHERE, ALL AT ONCE

Ermis Soumalias[1,2]*, **Jakob Heiss**[2,3,4]*, **Jakob Weissteiner**[1,2,5], **Sven Seuken**[1,2]
[1]University of Zurich (Department of Informatics)
[2]ETH AI Center
[3]ETH Zurich (Department of Mathematics)
[4]UC Berkeley (Department of Statistics)
[5]UBS Zurich
{ermis,weissteiner,seuken}@ifi.uzh.ch, jakob.heiss@berkeley.edu

ABSTRACT

We study the design of *iterative combinatorial auctions (ICAs)*. The main challenge in this domain is that the bundle space grows exponentially in the number of items. To address this, several papers have recently proposed machine learning (ML)-based preference elicitation algorithms that aim to elicit only the most important information from bidders to maximize efficiency. The SOTA ML-based algorithms elicit bidders' preferences via *value queries* (i.e., "What is your value for the bundle $\{A, B\}$?"). However, the most popular iterative combinatorial auction in practice elicits information via more practical *demand queries* (i.e., "At prices p, what is your most preferred bundle of items?"). In this paper, we examine the advantages of value and demand queries from both an auction design and an ML perspective. We propose a novel ML algorithm that provably integrates the full information from both query types. As suggested by our theoretical analysis, our experimental results verify that combining demand and value queries results in significantly better learning performance. Building on these insights, we present MLHCA, the most efficient ICA ever designed. MLHCA substantially outperforms the previous SOTA in realistic auction settings, delivering large efficiency gains. Compared to the previous SOTA, MLHCA reduces efficiency loss by up to a factor of 10, and in the most challenging and realistic domain, MLHCA outperforms the previous SOTA using 30% fewer queries. Thus, MLHCA achieves efficiency improvements that translate to welfare gains of hundreds of millions of USD, while also reducing the cognitive load on the bidders, establishing a new benchmark both for practicability and for economic impact.

1 INTRODUCTION

Combinatorial auctions (CAs) are used to allocate multiple items among several bidders who may view those items as complements or substitutes. In a CA, bidders are can submit bids for whole *bundles/packages* of items. CAs have enjoyed widespread adoption in practice, with their applications ranging from allocating spectrum licences (Cramton, 2013) to TV ad slots (Goetzendorff et al., 2015) and airport landing/take-off slots (Rassenti et al., 1982).

The key challenge in CAs is that the bundle space grows exponentially in the number of items, making it impossible for bidders to report their full value function in all but the smallest domains. Moreover, Nisan & Segal (2006) showed that for arbitrary value functions, CAs require an exponential number of bids in order to guarantee full efficiency. Thus, practical CA mechanisms cannot provide efficiency guarantees in real world settings with more than a modest number of items. Instead, the focus has shifted towards *iterative combinatorial auctions (ICAs)*, where bidders interact with the auctioneer over a series of rounds, providing only a limited (i.e., practically feasible) amount of information, with the aim to maximize the efficiency of the final allocation.

*These authors contributed equally.

The most established ICA following this interaction paradigm is the *combinatorial clock auction (CCA)* (Ausubel et al., 2006). The CCA has been extensively used for allocating spectrum licenses, generating over *USD* 20 *billion* in revenue between 2012 and 2014 alone (Ausubel & Baranov, 2017). Speed of convergence is a critical consideration for any ICA since each round entails costly computations and business modelling for the bidders (Kwasnica et al., 2005; Milgrom & Segal, 2017; Bichler et al., 2017). Large spectrum auctions following the CCA format can take more than 100 bidding rounds. In order to decrease the number of rounds, many CAs in practice use aggressive price update rules (e.g., increasing prices by up to 10% each round), which can harm efficiency (Ausubel & Baranov, 2017). Thus, it remains a challenging problem to design a practical ICA that is efficient and converges in a small number of rounds. Specifically, given the value of resources allocated in such real-world ICAs, increasing their efficiency by even one percentage point already translates into welfare gains of hundreds of millions of dollars.

1.1 ML-Powered Iterative Combinatorial Auctions

To address this challenge, researchers have proposed various ways of using *machine learning (ML)* to improve the efficiency of ICAs. The seminal works by Blum et al. (2004) and Lahaie & Parkes (2004) were the first to frame preference elicitation in CAs as a learning problem. In more recent years, Brero et al. (2018; 2021), Weissteiner & Seuken (2020); Weissteiner et al. (2022b;a; 2023) proposed ML-powered ICAs. At the heart of those approaches lies an ML-powered preference elicitation algorithm that uses an ML model to learn each bidder's value function to generate an informative *value query* (i.e., "What is your value for the bundle $\{A, B\}$?"), which in turn refines that bidder's ML model.[1]

While those value-query based ML-powered ICAs lead to significant efficiency gains redefining the *state-of-the-art (SOTA)* efficiency results in many realistic auction domains, those approaches suffer from one common practical limitation: they fundamentally rely throughout the whole ICA on *value queries (VQs)*. Prior research in auction design has identified *demand queries (DQs)* as the best way to run an auction (Cramton, 2013). Their advantages compared to value queries include elimination of tacit collusion and bid signaling, as well as simplified bidder decision-making that keeps the bidders focused on what is most relevant: the relationship between prices and aggregate demand. Additionally, value queries are cognitively complex, and thus typically should be only used sparsely in real-world ICAs. For these reasons, DQs are the most prominent interaction paradigm for auctions in practice. Following this rationale, Soumalias et al. (2024b) addressed the common limitation of prior work by designing the first practical ML-powered ICA that elicits information from bidders via DQs instead of VQs and only makes use of VQs in supplementary rounds, when bidders have already obtained a clearer picture on which bundles they can realistically hope to clinch and how much they should approximately value such bundles.

While this DQ-based ICA represented a significant leap towards making ML-powered ICAs practical and at the same time outperformed the baseline CCA that is typically used in real-world applications, it still suffered from the following two important deficiencies: First, it could not reach the SOTA efficiency of the impractical VQ-based ML-powered ICAs. Second, to improve efficiency, just like the CCA, it required the use of a *supplementary round*, in which the bidders must decide on which additional *value bids* to submit to the mechanism, a cognitive complicated task for the bidders.

The present paper closes these two last gaps in the realm of ICAs by designing a hybrid ML-powered ICA that combines DQ-based rounds with a sophisticated yet practical VQ-based supplementary round. Importantly, this hybrid ML-powered ICA clearly outperforms the previous SOTA ICA while still being practical in real-world applications.

1.2 Our Contributions

In this paper, we introduce the *Machine Learning-powered Hybrid Combinatorial Auction (ML-HCA)*, a practical ICA that achieves unprecedented efficiency. Our contributions are as follows:

[1] From an optimization task perspective this setting can be viewed as a combinatorial Bayesian optimization problem.

1. In Section 3, we provide a theoretical foundation and illustrative examples that demonstrate the advantages and limitations of DQs and VQs as input mechanisms for auctions and learning algorithms.
2. In Section 4, we introduce a learning algorithm capable of leveraging both types of queries. We provide strong experimental evidence of the learning benefits of combining both query types, as well as the advantages of starting an auction with DQs instead of VQs.
3. In Section 5 we combine our auction and ML insights to develop MLHCA, the first ICA to incorporate both sophisticated DQ and VQ generation algorithm. Simulations in realistic domains show that MLHCA significantly outperforms the previous SOTA, achieving higher efficiency with 40% fewer queries (Section 6), setting a new benchmark for both efficiency and practicality.

1.3 Further Related work

In the field of *automated mechanism design*, Dütting et al. (2015; 2019), Golowich et al. (2018) and Narasimhan et al. (2016) used ML to learn new mechanisms from data, while Cole & Roughgarden (2014); Morgenstern & Roughgarden (2015) and Balcan et al. (2023) bounded the sample complexity of learning approximately optimal mechanisms. In contrast to this prior work, our design incorporates an ML algorithm into the mechanism itself, i.e., the ML algorithm is part of the mechanism. Lahaie & Lubin (2019) suggest an adaptive price update rule that increases price expressivity as the rounds progress in order to improve efficiency and speed of convergence. Unlike that work, we aim to improve preference elicitation in the main rounds while still using linear prices. Preference elicitation is a key market design challenge outside of CAs too. Soumalias et al. (2024a) introduce an ML-powered mechanism for course allocation that improves preference elicitation by asking students comparison queries.

Despite the prominence of DQs in real-world applications, the only prior work apart from Soumalias et al. (2024b) on ML-based DQs that we are aware of is that of Brero & Lahaie (2018) and Brero et al. (2019), who proposed integrating ML in a price-based ICA to generate the next price vector in order to achieve faster convergence. However, this prior work does not exploit any notion of similarity between bundles that contain overlapping items, only incorporates a fraction of the information revealed by the agents' bidding (i.e., for the bundle an agent bids on, her value for that bundle must be larger than its price), and is computationally intractable already in medium-sized auction domains. See Appendix D for further related work.

1.4 Practical Considerations and Incentives

MLHCA integrates both ML-powered DQ and VQ rounds. In DQ-based auctions like the CCA or ML-CCA, ensuring truthful bidding depends heavily on well-chosen *activity rules* and *payment rules*. In Appendix A.3, we provide a detailed discussion of the most common activity rules used in the CCA to align incentives, and detail how MLHCA can also leverage these rules to achieve the same goal.

The VQ rounds in MLHCA extend the MLCA framework (Brero et al., 2021) by incorporating information from earlier DQ rounds into bidders' ML models. Brero et al. (2021) argued that MLCA offers strong practical incentives, and under two additional assumptions, truthful bidding is an ex-post Nash equilibrium. In Appendix A.4 we provide a detailed discussion of these arguments, and detail why they also apply to MLHCA's VQ rounds.

By effectively combining activity rules in the DQ rounds and leveraging the established incentive structure of MLCA in the VQ rounds, MLHCA achieves a robust incentive alignment across all its stages.

2 Preliminaries

2.1 Formal Model for ICAs

We consider *multiset* CA domains with a set $N = \{1, \dots, n\}$ of bidders and a set $M = \{1, \dots, m\}$ of distinct items with corresponding *capacities*, i.e., number of available copies, $c = (c_1, \dots, c_m) \in$

$\mathbb{N}^m$. We denote by $x \in \mathcal{X} = \{0, \dots, c_1\} \times \dots \times \{0, \dots, c_m\}$ a bundle of items represented as a positive integer vector, where $x_j = k$ iff item $j \in M$ is contained k-times in x. The bidders' true preferences over bundles are represented by their (private) value functions $v_i : \mathcal{X} \to \mathbb{R}_{\geq 0},\ i \in N$, i.e., $v_i(x)$ represents bidder i's true value for bundle $x \in \mathcal{X}$. We collect the value functions v_i in the vector $v = (v_i)_{i \in N}$. By $a = (a_1, \dots, a_n) \in \mathcal{X}^n$ we denote an allocation of bundles to bidders, where a_i is the bundle bidder i obtains. We denote the set of *feasible* allocations by $\mathcal{F} = \left\{a \in \mathcal{X}^n : \sum_{i \in N} a_{ij} \leq c_j,\ \forall j \in M\right\}$. We assume that bidders have quasilinear utility functions u_i of the form $u_i(a_i) = v_i(a_i) - \pi_i$ where v_i can be highly non-linear and $\pi_i \in \mathbb{R}_{\geq 0}$ denotes the bidder's payment. This implies that the (true) *social welfare* $V(a)$ of an allocation a is equal to the sum of all bidders' values $\sum_{i \in N} v_i(a_i)$.[2] We let $a^* \in \arg\max_{a \in \mathcal{F}} V(a)$ denote a social-welfare maximizing, i.e., *efficient*, allocation. The *efficiency* of any allocation $a \in \mathcal{F}$ is $V(a)/V(a^*)$.

An ICA *mechanism* defines how the bidders interact with the auctioneer and how the allocation and payments are determined. In this paper, we consider ICAs that iteratively ask bidders both *linear demand queries* (DQs) and *value queries* (VQs).

Definition 1 (Linear Demand Query). *In a linear demand query, the auctioneer presents a vector of item prices $p \in \mathbb{R}^m_{\geq 0}$ and each bidder i responds with her utility-maximizing bundle, i.e.,*

$$x_i^*(p) \in \underset{x \in \mathcal{X}}{\arg\max} \left\{v_i(x) - \langle p, x \rangle\right\}\ \ i \in N, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean scalar product in $\mathbb{R}^m$.

Definition 2 (Value Query). *In a value query, the auctioneer presents to bidder i a bundle of items x and bidder i responds with her value at those prices, i.e., $v_i(x) \in \mathbb{R}_{\geq 0}$.*

For bidder $i \in N$, let her $K \in \mathbb{N}$ elicited DQs be denoted as $R_i^{DQ} = \{(x_i^*(p^r), p^r)\}_{r=1}^K$ and her $L \in \mathbb{N}$ elicited VQs as $R_i^{VQ} = \left\{\left(x_i^l, v_i(x_i^l)\right)\right\}_{l=1}^L$. Bidder i's reports are denoted as $R_i = R_i^{DQ} \cup R_i^{VQ}$. Let $R = (R_1, \dots, R_n)$ be the tuple of elicited query data from all bidders.

In any auction that uses DQs, an important notion is the bidder's *inferred value*. A bidder's inferred value for a bundle is the maximum lower bundle on her value that the auctioneer can deduce, based on that bidder's reports. A bidder's inferred value for a bundle is weakly lower than her true value, with the equality holding in case the bidder answered the corresponding VQ for that bundle. Formally:

Definition 3 (Inferred Value). *Bidder i's inferred value for bundle $x \in \mathcal{X}$ given her reports R_i is*

$$\widetilde{v}_i(x; R_i) = \begin{cases} v_i(x) & \textit{if } (x, v_i(x)) \in R_i^{VQ}, \\ \max\left\{\left\{\langle x, p^r \rangle : (x, p^r) \in R_i^{DQ}\right\} \cup \{0\}\right\} & \textit{otherwise.} \end{cases} \tag{2}$$

The ICA's final allocation $a^*(R) \in \mathcal{F}$ and payments $\pi_i \coloneqq \pi_i(R) \in \mathbb{R}^n_{\geq 0}$ are computed based on the elicited reports R *only*. Concretely, $a^*(R) \in \mathcal{F}$ is determined by solving the *Winner Determination Problem (WDP):*

$$a^*(R) \in \underset{a \in \mathcal{F}}{\arg\max} \sum_{i \in N} \widetilde{v}_i(a_i; R_i), \tag{3}$$

where $\sum_{i \in N} \widetilde{v}_i(a_i; R_i)$ is the allocation's *inferred social welfare*, a lower bound for its *social welfare* $\sum_{i \in N} v_i(a_i)$.

2.2 Benchmark ICAs

In this section, we briefly introduce the three main benchmarks considered in this paper.

CCA The most established ICA is the *Combinatorial Clock Auction (CCA)* (Ausubel et al., 2006). The CCA consists of two phases. The initial *clock phase* proceeds in rounds. In each round r, the auctioneer sets anonymous item prices $p^r \in \mathbb{R}^m_{\geq 0}$, prompting each bidder to respond to a DQ, declaring her utility-maximizing bundle at p^r. In the next round, the prices of over-demanded items

[2] Note that $V(a) = \sum_{i \in N} u_i(a_i) + u_{\text{auctioneer}}(a) = \sum_{i \in N} (v_i(a_i) - \pi_i) + \sum_{i \in N} \pi_i = \sum_{i \in N} v_i(a_i)$.

are increased by a fixed percentage, until over-demand is eliminated. The second phase of the CCA, known as the *supplementary round*, allows bidders to report their valuations for additional bundles, governed by specific *activity rules* to promote incentive alignment. The *clock bids raised heuristic* suggests that bidders report their values for all bundles they requested during the clock phase. The final allocation is determined by solving the WDP based on all reports from both phases, as in Equation (3).

ML-CCA The most efficient DQ-based ICA is the *Machine Learning-powered Combinatorial Clock Auction (ML-CCA)* (Soumalias et al., 2024b). ML-CCA has the same interaction paradigm as the CCA, but with a substantially more refined DQ-generation algorithm in its clock phase. In each round, an ML model is trained to estimate each bidder's value function based on previously submitted DQ responses. The auctioneer then solves a convex optimization problem to determine the prices with the greatest clearing potential for those value function estimates.

BOCA The SOTA ICA in terms of efficiency is the VQ-based *Bayesian optimization-based combinatorial auction (BOCA)* (Weissteiner et al., 2023). The main idea of BOCA is that in each round, the auctioneer creates an estimate of the upper uncertainty bound of the value function of each agent based on her past responses. Then, the auctioneer solves an ML-based WDP to find the feasible allocation with the highest upper bound on its estimated social welfare, and queries each agent her value for her bundle in that allocation. This allows the mechanism to balance between exploring and exploiting the bundle space during its preference elicitation phase.

2.3 ML Framework

The ML models used by ML-CCA, and as basis for the construction of the uncertainty bound estimates in BOCA are *monotone-value neural networks (MVNNs)* $\mathcal{M}_\theta : \mathcal{X} \to \mathbb{R}$ (Weissteiner et al., 2022a). MVNNs are a recently introduced class of NNs specifically designed to represent *monotone combinatorial* valuations. MVNNs have also had success in combinatorial allocation domains without money, e.g. for course allocation Soumalias et al. (2024a). Soumalias et al. (2024b) introduced *multiset MVNNs (mMVNNs)*, an extension of MVNNs that also incorporates at a structural level the information that some items in the auction are identical copies of each other. In this work, we instantiate our ML models using mMVNNs, and denote agent i's model as $\mathcal{M}_i : \mathcal{X} \to \mathbb{R}$. Within this work, we will refer to all mMVNNs simply as MVNNs. We provide more details on MVNNs in Appendix I.

3 Advantages of DQs and VQs and why one should combine them

In this section, we examine the limitations of using only VQs or only DQs in auctions and highlight the benefits of combining them. All deferred proofs can be found in Appendix E.

3.1 Disadvantages of only using VQs

Almost all ML-powered VQ-based auctions including the current SOTA, BOCA (Weissteiner et al., 2023) first ask each bidder multiple random VQs (i.e., VQs for randomly selected bundles). These VQs are necessary to initialize the ML estimates of the bidder's value functions. In practice, it is very hard for bidders to answer random VQs since they are not aligned with their preferences.[3] The most popular ICAs in practice (e.g., the CCA) ask the bidders DQs, which have been argued can be answered by the bidders sufficiently well (Cramton, 2013).[4]

Even if bidders manage to respond perfectly to random VQs, the information obtained is limited. This is because, in large combinatorial domains, bidders typically have high values for only a small subset of possible bundles, making the probability of querying one of these high-value bundles at

[3] To provide some intuition, imagine you go to the supermarket because you want to bake a birthday cake for your friend and then you are asked your value for 30 frying pans plus 500 coconuts. It might be hard to estimate your value for such a random combination of items.

[4] In our practical supermarket example, now imagine that you view the price tags for the same items. It is quite doable to decide which items you want to buy and in which quantities.

random exceedingly low. On the other hand, querying bidders with DQs at a random price vector is more likely to prompt responses that reveal their high-value bundles. Formally:

Lemma 1. *The difference in expected social welfare between an auction that uses a single random demand query and an auction that uses $k \ll 2^m$ random value queries can be arbitrarily large.*

Remark 1. *This limitation of random VQs is evidenced in practice. Empirical comparisons between VQ-based ML-powered mechanisms, such as Weissteiner et al. (2023), and DQ-based mechanisms, such as Soumalias et al. (2024b), reveal that efficiency after initial queries is significantly lower for VQ-based approaches across all tested domains (see Figure 4 in Section 6).*

Beyond auction efficiency, the limited information provided by random VQs poses challenges for learning algorithms in ML-powered ICAS. In contrast, DQs provide global information about bidder preferences across the entire bundle space. When bidder i responds to a DQ at prices p, she solves the optimization problem: $x_i^*(p) \in \arg\max_{x \in \mathcal{X}} \{v_i(x) - \langle p, x \rangle\}$, which reveals valuable information about her preferences across all possible bundles. Strong evidence for this is presented inSection 4.2, where we show that the network trained only on DQs exhibits better generalization performance than one trained on random VQs.

Additionally, if DQ prices are sufficiently low, bidders respond with their value-maximizing bundles, which may be hard to recover through VQs alone. By incorporating this information, the learning algorithm can more effectively identify critical regions in the allocation space and subsequently focus on refining those areas. This advantage is further supported by our experiments (Figure 4 in Section 6). We show that in our ML-powered hybrid auction, the first ML-powered VQ after a series of DQs achieves significantly higher efficiency compared to the first ML-powered VQ after an equivalent number of random VQs in the current SOTA VQ-based auction.

Moreover, even if the auction finds an efficient allocation by using VQs, it cannot terminate early as there is no way for the auctioneer to certify that the auction has reached 100% efficiency. In contrast, for DQ-based auctions there is an easy condition that allows the auction to terminate early:

Proposition 1. *If clearing prices exist, an auction using DQs can provide a guarantee of optimal efficiency and terminate early.*

Proof. If clearing prices have been found, the corresponding allocation constitutes a *Walrasian equilibrium*, and thus has an efficiency equal to 100%. See (Soumalias et al., 2024b, Appendix C.1) for a detailed proof. □

Remark 2. *This is indeed an issue in practice. In Section 6, we experimentally show that, in realistic domains, our MLHCA can often reach* 100% *efficiency before the common maximum number of* 100 *rounds used by most ML-powered ICAs (e.g. Weissteiner & Seuken (2020); Weissteiner et al. (2022b;a; 2023); Soumalias et al. (2024b)) is reached.*

3.2 Disadvantages of only using DQs

In this section, we show the disadvantages of using DQs to elicit the bidders' preferences.

The first major disadvantage of an auction employing only DQs is that the auction's efficiency can actually drop by adding more DQs.

Lemma 2. *In a DQ-based ICA, adding DQs can actually reduce efficiency. A single DQ can cause an efficiency drop arbitrarily close to* 100%. *By comparison, in a VQ-based ICA, adding additional queries can never reduce efficiency (assuming truthful bidding).*

Proof. Let $m = 2$, $n = 2$, $c_1 = 1$, $c_2 = 1$,

$$v_1 = \max\{400 \cdot \mathbb{1}_{x_1 \geq 1}, 2 \cdot \mathbb{1}_{x_2 \geq 1}\} \text{ and}$$
$$v_2 = 1.1 \cdot \mathbb{1}_{x_1 \geq 1}.$$

Suppose the auction has asked two DQs. The first DQ $p = (1, 1)$ is responded by both bidders with $(1, 0) \in \arg\max_{x \in \mathcal{X}} \{v_i(x) - \langle p, x \rangle\}$. The second DQ $p = (1.2, 1)$ is responded by bidder 1 with $(1, 0) \in \arg\max_{x \in \mathcal{X}} \{v_1(x) - \langle p, x \rangle\}$ and by bidder 2 with $(0, 0) \in \arg\max_{x \in \mathcal{X}} \{v_2(x) - \langle p, x \rangle\}$.

After these 2 DQs the WDP based on the inferred values (see Equation (3)), would assign item 1 to bidder 1 (resulting in an inferred social welfare of $1.2 + 0 = 1.2$). This is the efficient allocation with a true SCW of 400, i.e., an efficiency equal to 100%.

Now suppose that a third DQ $p = (401, 1)$ is added to the auction. Bidder 1's demand response is $(0, 1) \in \arg\max_{x \in \mathcal{X}} \{v_1(x) - \langle p, x\rangle\}$ and bidder 2's response is $(0, 0) \in \arg\max_{x \in \mathcal{X}} \{v_2(x) - \langle p, x\rangle\}$. The WDP would now assign item 2 to bidder 1 and item 1 to bidder 2, resulting in an inferred SCW of $1+1 = 2$). This would result only in an efficiency of $\frac{2+1.1}{400} < 1\%$.

While the inferred SCW obviously cannot decrease in any round (since the set we maximize over cannot decrease in any round and inferred values cannot decrease), we have shown here that the true SCW can decrease substantially. In this example, the SCW dropped by more than 99%. One could easily modify this example to even obtain an efficiency drop arbitrarily close to 100% if one decreases the values 1,1.2 and 2 (the prices and the values inside the value functions) by any small factor or increases the numbers 400 and 401, by any large factor. Then the proof would still work, which shows that the efficiency can even fall from 100% to values arbitrarily close to 0%.

On the other hand, if we only ask VQs, there is no difference between inferred SCW and true SCW (assuming truthful bidding), which results in non-decreasing SCW. □

Remark 3. *This is a significant issue in practice. In Section 6 we experimentally show that in the most realistic spectrum auction domain, the CCA's efficiency drops by over* 7% *with the introduction of more DQs. In a second realistic domain, the CCA actually has higher efficiency after just* 5 *DQs compared to after* 100*. This efficiency degradation is not only a concern for the CCA but also affects ML-powered DQ-based ICAs in similar ways.*

In the next lemma, we show that the same issue arises in an auction that uses both DQs and VQs:

Lemma 3. *In an auction that first uses DQs and then VQs, adding VQs can actually reduce efficiency. The efficiency drop can even be arbitrarily close to 100%.*

Proof. Consider the setting from the proof of Lemma 2 including the first 2 DQs. Recall that in this setting after these 2 DQs, the WDP would achieve 100% efficiency. Now instead of the third DQ, we ask the following VQ: We ask bidder 1 for her value of the bundle $(0, 1)$ and we ask bidder 2 for her value of the bundle $(1, 0)$. Then the WDP based on these 3 rounds would assign item 2 to bidder 1, and item 1 to bidder 2, as we explain in the following. The inferred SCW $v_1(0, 1) + v_2(1, 0) = 2 + 1.1$ (which is equal to the true SCW of this allocation) is higher than the inferred SCW of all other allocations consisting of elicited bundles: For bidder 1 the DQ responses were always $(1, 0)$ with inferred value 1.2, and the VQ elicited $v_1(0, 1) = 2$. For bidder 2, the DQ responses were $(1, 0)$ with inferred value 1 and $(0, 0)$ with inferred value 0, and the VQ response was $v_2(1, 0) = 1.1$. So we see that the highest inferred SCW among all feasible allocations is achieved by assigning item 2 to bidder 1 and item 1 to bidder 2 (e.g., assigning it the other way around would only achieve an inferred SCW of $1.2 + 0$, while the true SCW $v_1(1, 0) + v_2(0, 1) = 400 + 0$ would be much larger).

So the efficiency dropped from 100% to $\frac{2+1.1}{400} < 1\%$ after the VQ (i.e., the efficiency drops by more than 99%). □

Even though Lemma 3 shows that an auction using DQs followed by VQs can still experience an arbitrarily large efficiency drop, we can completely address this issue using a *single* carefully designed VQ, which we call the "bridge bid."

Definition 4 (Bridge bid)**.** *The bridge bid asks each bidder her value for the bundle she would have been allocated according to the WDP after the last DQ.*

Lemma 4. *In an ICA that first asks DQs and then VQs, by first using a single specific VQ, the bridge bid from Definition 4, the auction can ensure its efficiency is at least as high as the efficiency achieved by its DQs alone.*

Proof. The bridge bid itself can obviously not decrease efficiency, because it simply replaces the inferred SCW of the winning allocation of the previous WDP with the true SCW of exactly the same allocation. In other words, the inferred values of the bundles of the previously WDP-winning allocation can be increased or stay the same, while all the other inferred values stay the same. Thus the winning allocation stays the winning allocation when the bridge bid is added. For the remainder of the proof, we will show that all the VQs after the bridge bid can also not decrease the efficiency. In every further WDP another allocation can only outperform the bridge bid allocation if it has a higher inferred[5] social welfare. However, if it has a higher inferred SCW, it's true SCW cannot be lower than the one of the bridge bid. And as we have shown in the beginning of the proof, the SCW of the allocation of the bridge bid is equal to the SCW of the last WDP winner after the last DQ. Thus, for any VQ after the bridge bid the SCW cannot be worse than the winning allocation of the WDP right after the last DQ. □

Remark 4. *Again, this in practice is highly impactful. In our experimental section (Section 6), we show that in the most realistic domain, our MLHCA without this bridge bid loses* 7 *percentage points of efficiency. The auction needs another* 20 *VQs to recover its DQ-only efficiency. By using just a single specialized VQ, the bridge bid, we can completely alleviate this problem. For a more detailed discussion, see Section 6.2.*

The next theorem shows an even more fundamental limitation of only asking DQs. Specifically, asking only DQs can result in low efficiency, *even* in the limit where we ask *all* possible DQs.

Theorem 1. *A DQ-based auction cannot guarantee even* 55% *efficiency, even if it asks all (unaccountably many) possible DQs (i.e., a DQ for every price vector in* $[0,\infty)^m$*). This remains true even if the bidders additionally report their true values for all bundles they requested in those DQs.*

Proof. Let $n = 2$, $m = 1$, $c_1 = 10$, $v_1(x) = 100\mathbb{1}_{x_1 \geq 1}$, $v_2(x) = 9x_1 + \frac{1}{25}x_1^2$. Here, the unique efficient allocation would assign 1 item to bidder 1 and the remaining 9 items to bidder 2, resulting in an SCW of $v_1(1) + v_2(9) = 100 + 9 \cdot 9 + \frac{9^2}{25} = 184.24$. However, there is no DQ $p \in \mathbb{R}^m_{\geq 0}$ that bidder 2 would answer with $x_2^*(p) = (9)$:

- If the price p is below $\frac{94}{10}$, bidder 2 will answer with $x_2^*(p) = (10)$;
- if the price $p = \frac{94}{10}$, bidder 2 will answer with either $x_2^*(p) = (10)$ or $x_2^*(p) = (0)$;
- if the price p is higher than $\frac{94}{10}$, bidder 2 will answer with $x_2^*(p) = (0)$.

Therefore, the WDP cannot assign 9 items to bidder 2 if only DQs were asked no matter how many DQs were asked. Raising the bids for those bundles would also not help, because this would still not give us any value for 9 items for bidder 2. The best SCW that such WDPs based on DQ responses (and raised DQ responses) can achieve is thus 100, which results in an efficiency of $\frac{100}{184.24} \approx 54.28\%$. □

Thus, every method that only asks DQs (e.g., CCA or Soumalias et al. (2024b)) will result in inefficient allocations even in the limit of infinitely many iterations in the case of certain value functions (even if raised clock bids are added in the supplementary round).

Remark 5. *The issue highlighted in Theorem 1 also arises in practical settings. In Section 6, we experimentally show that in the most realistic domain, MRVM, the final 50 DQs of ML-CCA (Soumalias et al., 2024b), the current SOTA DQ-based ICA, only increase efficiency by 0.3% points. If the bidders also report their true values for all bundles they requested, this only causes an efficiency increase of less than* 0.2% *points. In contrast, for MLHCA, the last* 30 *VQs cause an efficiency increase of over* 1.8% *points. For the other domains, we see a qualitatively similar picture in Figure 4.*

In Theorem 1, we showed that a DQ-based auction cannot guarantee full efficiency. Intuitively, the driving force behind this limitation is that despite the broad information that DQs provide, they cannot fully reveal a bidder's value function. In Example 1, we show a practical example where both linear and non-linear value functions would result in exactly the same response to any DQ by the bidder. However, the same is not true for a VQ-based auction, leading to the following result:

[5] Note that after every VQ it is still possible that the WDP combines bundles queried during any VQ with bundles that were DQ responses for any old DQ. Thus even after some VQs the inferred SCW of the DWP-winning allocation can be strictly smaller than its true SCW.

Lemma 5. *Asking each bidder $\prod_{j=1}^{m} c_j$ different VQs guarantees that the allocation will be efficient.*

Proof. If the bidders give us their values for all possible bundles, then we have access to their complete value functions. Then the WDP is equivalent to optimizing the SCW. □

Thus, Weissteiner & Seuken (2020); Weissteiner et al. (2022b;a; 2023) and the hybrid method that we introduce in this paper have a guarantee to converge to an efficient allocation in the limit of infinitely many iterations.[6]

3.3 The Advantages of Combining DQs and VQs

DQs are cognitively simpler than VQs early in the auction. All ML-powered, VQ-based ICAs in the literature begin by asking bidders their values for uniformly at random selected bundles to initialize the ML models. In contrast, the SOTA ML-powered DQ-based approach (Brero & Lahaie, 2018; Brero et al., 2019; Soumalias et al., 2024b) starts by asking bidders for their preferred bundles at low initial prices that gradually increase over rounds. From a practical standpoint, it is nearly impossible for bidders to accurately assess VQs for randomly chosen bundles, whereas responding to DQs with low prices is far easier.[7] As the auction progresses and the bidders' ML models become more accurate, a VQ-based ML-powered ICA can ask targeted VQs that align better with bidder interests, making them easier to answer.[8]

DQs are more effective in the early stages of the auction. Initially, the auctioneer lacks knowledge of which bundles align with bidders' interests. Beginning with DQs allows the auctioneer to gather early insights about the bidders' preferences over the whole bundle space, facilitating the use of more targeted queries later on. This practice is well-established in the combinatorial auction community. For instance, the initial DQ phase in the CCA is often referred to as a "price discovery phase" (Ausubel et al., 2006). We argue that the same concept holds even in ML-powered auctions. Our experiments in Section 6 confirm that DQ-based approaches (e.g., ML-CCA (Soumalias et al., 2024b)) outperform VQ-based approaches (Weissteiner & Seuken, 2020; Weissteiner et al., 2022b;a; 2023) during the early rounds of the auction. However, as suggested by Theorem 1 and Lemma 5 , VQ-based approaches eventually surpass DQ-based mechanisms in later iterations.

A key contributing factor as to why VQ-based ML-powered approaches perform better than DQ-based approaches is that they can take into account the WDP, i.e., the downstream optimization problem that will determine the final allocation.[9] In contrast, responses to a single DQ often lead to over-demand for certain items or leave some items unassigned (under-demand). In Example 1, bidder 2 lacks information to know she should bid for 9 items. Only the auctioneer, having information from all bidders, knows that assigning 9 items to bidder 2 would complement bidder 1's preferences. The auctioneer can leverage this aggregated knowledge by asking bidder 2 a VQ for 9 items, whereas DQs alone would not provide this opportunity.

Example 1. *In the example from the proof of Theorem 1, after sufficiently many DQs have been asked, a single VQ would suffice to increase the social welfare from $\approx 56.45\%$ to 100%. MLHCA would ask this VQ in its first VQ round, provided that enough DQs had been asked beforehand, as we explain in the remainder of this example. v_1 can be very precisely reconstructed from DQs $p = \epsilon$ (which is responded by $x = (1)$), $p = 100 - \epsilon$ (which is responded by $x = (1)$), and $p = 100 + \epsilon$*

[6] Note that all these methods always enforce to ask a new VQ in any round, i.e., if the WDP suggests to ask a bidder a VQ for a bundle she was already asked for in a previous round, then we solve a constrained WDP instead with the constraint that this bidder is not allowed to be asked for any previously asked bundle again.

[7] In the example from Footnotes 3 and 4, imagine being asked your value for a bundle of 30 frying pans and 500 coconuts. It's hard to assess such a random combination. Now, imagine shopping at a supermarket with a 50% discount across all items; it's easier to determine what items you want under these conditions.

[8] Continuing with our example, imagine being asked for the value of ingredients specifically for a strawberry cake in one iteration and for a blueberry cake in the next. If your goal is to bake a cake, these targeted VQs are much easier to respond to.

[9] By definition, all the bundles in a VQ form a feasible allocation. Furthermore, VQs typically allocate (almost) all items to bidders, as they maximize the estimated social welfare. The MVNN architecture ensures monotonicity in the estimated value functions. If the estimated value functions were strictly monotonic, the solution to the MILPs determining the next VQ would always allocate all items.

(which is responded by $x = (0)$). The last two DQs reveal that $100 - \epsilon \leq v_1(1) \leq 100 + \epsilon$. And the first DQ reveals that $v_1(x) - v_1(1) \leq (x-1)\epsilon$ for any $x > 1$. Combining these information reveals that $v_1 \leq 100\mathbb{1}_{\cdot \geq 1} + 10\epsilon$ and with the help of monotonicity these 3 DQs reveal that $v_1 \geq 100\mathbb{1}_{\cdot \geq 1} - \epsilon$. So, we can reconstruct the true v_1 up to 10ϵ. For bidder 2, from DQs $p = \frac{94-\epsilon}{10}$ (which is responded by $x = 10$) and $p = \frac{94+\epsilon}{10}$ (which is responded by $x = 0$), we can only reconstruct that $v_2(\cdot) \leq \frac{94+\epsilon}{10}(\cdot)$ and that $v_2(10) \geq 94 - \epsilon$. E.g., the linear function $\frac{94}{10}(\cdot)$ would not contradict any possible DQ response from bidder 2. Our ML algorithm should not have any problem with estimating v_1 sufficiently well. If additionally, our ML algorithm estimates v_2 (approximately) as this linear function $\frac{94}{10}(\cdot)$, then the WDP would directly assign 1 item to bidder 1 and 9 items to bidder 2, which is the efficient allocation. In theory, MVNNs could also express functions that achieve 0 training loss on all DQs for bidder 2 but do not result in an efficient allocation. However, these functions would be highly non-linear and for many NN architectures it is shown that they prefer functions which are in a certain sense close to linear (Heiss et al., 2019; 2023; 2021; Heiss, 2024). In Appendix I.1 we explain, why our MVNNs would learn a linear approximation of v_2. Therefore the WDP would result in the efficient allocation in this example.

Remark 6. *Note that this example is not pathological. In Section 6, we will show that in realistic domains using 40 DQs and only 2 VQs, our MLHCA can achieve higher efficiency than the SOTA DQ-based mechanism using 100 queries.*

Our MLHCA is the first auction to integrate both a sophisticated DQ and VQ generation algorithm. By leveraging insights from auction theory and starting with DQs before transitioning to VQs, MLHCA achieves state-of-the-art efficiency in all rounds and demonstrates significantly improved final efficiency across all domains compared to the current state-of-the-art.

Moreover, we argue that the combination of DQs and VQs is particularly powerful for learning bidders' value functions, as the information from these two query types complements each other nicely (see Section 4).

4 Mixed Query Learning

In this section, we introduce our mixed training algorithm and provide experimental evidence supporting our theoretical analysis from Section 3. Specifically, we demonstrate the learning benefits of initializing auctions with DQs rather than VQs and highlight how combining DQs with VQs leads to superior learning performance.

4.1 Mixed Training Algorithm

To leverage the advantages of both DQs and VQs, we propose a two-stage training algorithm. In each epoch, the ML model is first trained on all DQ responses using the loss function from (Soumalias et al., 2024b). The main idea behind this loss is that for each DQ, an optimization problem is solved to predict the bidder's utility-maximizing bundle at the given prices, treating her ML model as her true value function. In case the predicted reply disagrees with the bidder's true reply, the loss is the difference in predicted utility between these 2 bundles. This loss provably incorporates the full information that the DQ responses provide. Then, the model is trained on the VQ responses using a standard regression loss. For more details, see Appendix F.

4.2 Experimental Analysis

In this section, we demonstrate the learning benefits of initializing auctions with DQs rather than VQs and highlight how combining both query types leads to superior learning performance.

We conduct the following experiment: We perform *hyperparameter optimization (HPO)* to train an mMVNN for the most critical bidder type in the most realistic domain—the national bidder in the MRVM domain. In Appendix G we present the same experiment for all other domains. Our HPO procedure is the following. For a single bidder of that type, we generate three distinct training sets:

1. The first training set contains 40 DQs simulating 40 CCA clock rounds, along with 20 VQs for bundles chosen uniformly at random.

Optimization	Train Points		$\mathbf{R^2}$		KT		MAE scaled		$\mathbf{R_c^2}$	
Metric	VQs	DQs	$\mathcal{T}_r$	$\mathcal{T}_p$	$\mathcal{T}_r$	$\mathcal{T}_p$	$\mathcal{T}_r$	$\mathcal{T}_p$	$\mathcal{T}_r$	$\mathcal{T}_p$
R^2 on $\mathcal{V}_r$	20	40	0.84	0.42	0.79	0.80	0.037	0.044	0.84	0.80
	60	0	0.73	-10.07	0.68	0.64	0.052	0.236	0.74	0.20
	0	60	0.24	-3.07	0.77	0.77	0.103	0.128	0.83	0.76
R^2 on $\mathcal{V}_p$	20	40	0.82	0.01	0.79	0.80	0.041	0.062	0.84	0.83
	60	0	0.76	-3.40	0.72	0.62	0.049	0.141	0.77	0.05
	0	60	-0.05	-6.24	0.78	0.72	0.103	0.154	0.84	0.69

Table 1: Learning comparison of training only on DQs, only on VQs, or on both. Shown are averages over ten instances for the winning configuration of each HPO procedure. Winners are marked in gray.

2. The second training set consists of 60 DQs, simulating 60 CCA clock rounds, with no VQs.

3. The third training set contains 60 VQs and no DQs.

We evaluate the generalization performance of the trained models on two distinct sets: A *random bundle set* ($\mathcal{V}_r$), which consists of 50,000 bundles sampled uniformly at random from the bundle space. A *random price-driven set* ($\mathcal{V}_p$), which consists of the bundles requested by the bidder in 200 randomly generated price vectors $\{p^r\}_{r=1}^{200}$, where each item's price is drawn uniformly between 0 and three times its average value for that bidder type. $\mathcal{V}_r$ evaluates generalization performance over the entire bundle space, while $\mathcal{V}_p$ focuses on the bidder's utility-maximizing bundles for various prices.

For each HPO configuration, we average the performance across 10 bidders of the same type. The best-performing configuration for each validation set is selected based on the coefficient of determination.

For the selected configurations, we evaluate performance on 10 separate test seeds representing new bidders, generating the test sets $\mathcal{T}_r$ and $\mathcal{T}_p$ in the same way as for the validation sets. For each test set, we report the *coefficient of determination (R^2)*, *Kendall Tau (KT)*, scaled *Mean Absolute Error (scaled MAE)* normalized with respect to the average value of a bundle in that domain and R^2 centered (R_c^2), a shift invariant version of R^2. An R_c^2 value of 1 indicates that the ML model has learned the bidder's value function perfectly, up to a constant shift. By comparing R_c^2 with the standard R^2, we can assess, for the bundles tested, the shift magnitude in the learned value function.[10]

Each HPO procedure was conducted under identical conditions, including the same test instances, random seeds, hyperparameter search space, and total computation time. For more details on the HPO process, see appendix K.1.

Table 1 shows that training on a mixture of DQs and VQs consistently outperforms training on either query type alone. This is evident across all metrics, and especially for the utility-maximizing bundles of test set $\mathcal{T}_p$, where mixed training yields almost three times lower MAE compared to other approaches.

Furthermore, the mixed-query model was the only one able to approximately learn the correct mean value for both validation sets, as reflected by the small difference between its R_c^2 and standard R^2. In contrast, models trained solely on DQs or VQs showed a much larger discrepancy between these two metrics for at least one of the validation sets. As explained in Section 3, when training only on DQs, the model only has relative information about bundle values and thus the value function is not uniquely identifiable, preventing the network from learning it accurately. On the other hand, models trained solely on VQs experience a distributional shift between the two test sets—one set focuses on utility-maximizing bundles, while the other contains bundles selected uniformly at random. Since

[10] Note that this shift is not perfectly constant as (m)MVNNs map the zero bundle to zero.

the VQ training set is drawn uniformly at random and lacks utility-maximizing bundles, the model fails to capture the bidder's value function for these critical bundles.[11]

In Table 1 we observe that the models trained only on DQs exhibit much better generalization performance in the bundles of $\mathcal{T}_p$ than the models trained on random VQs, despite of their lack of absolute value information. The reason for the better generalization performance is the strong distributional shift between the bundles of the two sets. But from an allocative value perspective, the bundles in the set $\mathcal{T}_p$ are those for which the bidders have high utility, and thus value. Thus, this is the critical area of the allocation space where the auctioneer wants the models to perform well. This gives strong empirical motivation as to why starting the learning process with DQs is more effective than starting it with VQs. In Section 6 we will show that the efficiency after the first ML-powered VQ is, across all domains, much higher for the model trained on DQs compared to the one trained on random VQs. The reason behind this improvement is precisely the fact that the DQ-trained models have learned a better approximation of the bidders' value functions in the most critical part of the allocation space. In fact, the learning performance is so much better that, in two out of the four realistic domains tested, a *single* ML-powered VQ in the DQ-trained networks suffices to achieve better auction efficiency than the VQ-trained networks using 60 ML-powered VQs.

Comparing the models trained only on DQs with random VQs in Table 1 provides strong empirical evidence of the two main, orthogonal learning advantages of starting an ML-powered auction with DQs compared to random VQs. The first advantage is that CCA DQs provide global information about the bundle space, which promotes exploration of the allocation space. This global information that DQs provide is evident from the higher KT that the DQ-trained network can achieve across both test sets compared to the VQ-trained one. The reason for this increased performance is that, as explained in Section 3, DQs provide global relative information about the entire allocation space.

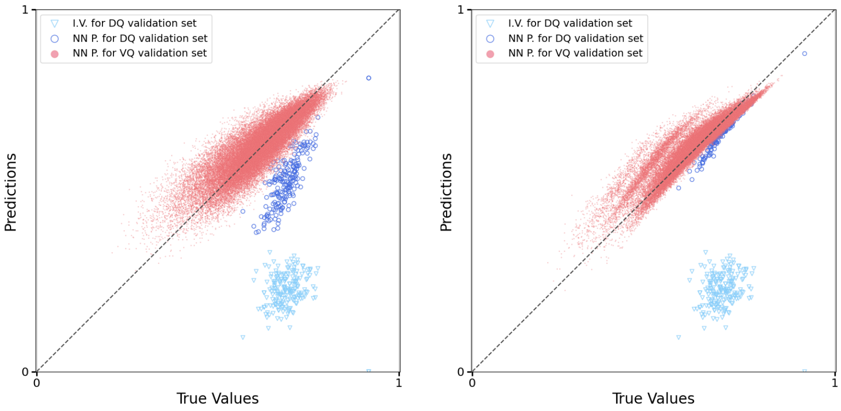
The second learning advantage of starting an auction with CCA DQs is that they provide particularly much information about the critical, high valued areas of the allocation space right from the start. This is evident from the fact that the models trained only on DQs exhibit much better generalization performance in the bundles of $\mathcal{T}_p$ than the models trained on random VQs, despite their lack of absolute value information. The reason for the better generalization performance is the strong distributional shift between the bundles of the two sets. But from an allocative value perspective, the bundles in the set $\mathcal{T}_p$ are those for which the bidders have high utility, and thus value. Thus, this is the critical area of the allocation space where the auctioneer wants the models to perform well.

These two learning advantages are so critical that, as we will demonstrate in Section 6, the efficiency gains after the first ML-powered VQs is, across all domains, much higher for the model trained on DQs compared to the one trained on random VQs. In fact, the learning performance is so much better that, in two out of the four domains, our hybrid auction (Section 5) using just two ML-powered VQs, following training on 40 DQs, achieves higher efficiency than the SOTA VQ-based mechanism using 40 random VQs and 60 ML-powered VQs.

In Figure 1, we present prediction vs. true value plots for the top-performing configurations with respect to R^2 on $\mathcal{V}_r$ from Table 1. We compare the model trained on 40 DQs and 20 VQs against the one trained on 60 VQs, corresponding to the first and second rows of Table 1. Bundles from $\mathcal{T}_c$ are represented by red circles, while those from $\mathcal{T}_p$ are shown in blue. For bundles in $\mathcal{T}_p$, we also plot their *inferred values*, reflecting their price when the bidder requested them.

In Figure 1a we observe that the model trained solely on VQs consistently under predicts values for bundles in $\mathcal{T}_p$. Furthermore, there is a very large spread in the predicted values of these bundles. These bundles are out of distribution for the network, and thus it cannot generalize to them. If we examine the inferred values for the same bundles, we observe a substantial deviation from the true diagonal. The vertical distance between each bundle's inferred value and the true diagonal line corresponds to the bidder's utility when requesting that bundle - the quantity she is maximizing. In contrast, as shown in Figure 1b, the model trained on the mixed dataset is able to place the bundles of $\mathcal{T}_p$ in an almost perfect parallel line to the true diagonal, and with a much smaller shift. These bundles are not out of distribution for that network, which means it can perform better. For the bundles of $\mathcal{T}_c$, we observe that the predictions of both models are centered around the true diagonal, indicating that both networks have learned the correct mean value. However, again we can

[11] Note that at the start of an ML-powered, VQ-based auction, the ML models are not yet sufficiently accurate, preventing the auctioneer from asking VQs for utility- or value-maximizing bundles.



(a) (b)

Figure 1: Comparison of scaled prediction vs. true values for an mMVNN trained with different query types for the national bidder in the MRVM domain. (a) Training with 60 demand queries. (b) Training with 40 demand queries and 20 value queries.

observe that for the network trained on the mixed dataset, its predictions on $\mathcal{T}_c$ are again more tightly clustered in a line around the true diagonal, as was also suggested by the stronger MAE and KT in Table 1. These observations illustrate the powerful synergy between DQs and VQs. The *global, relative information* provided by DQs enables the network to align its predictions roughly along a consistent trajectory—essentially forming a parallel line to the true diagonal. The *absolute value information* from the VQs then fine-tunes this alignment, effectively positioning the line exactly on the true diagonal, ensuring the predicted values match the true values accurately.

5 The Mechanism

In this section, we describe our *ML-powered Hybrid Combinatorial Auction (MLHCA)*, which combines the auction and ML insights of how to combine DQs and VQs from Sections 3 and 4.

We present a simplified version of MLHCA in Algorithm 1. In Lines 3 to 6, we generate the first $Q^{\text{CCA}} \in \mathbb{N}$ DQs using the same price update rule as the CCA (with larger price increments). In each of the next $Q^{\text{DQ}} \in \mathbb{N}$ ML-powered rounds, we first train, for each bidder, an mMVNN on her demand responses (Line 9). Next, in Line 10, we call NextPrice (Soumalias et al., 2024b) to generate the next DQ based on the agents' trained mMVNNs (see Appendix C). If MLHCA has found market-clearing prices, then the corresponding allocation is efficient and is returned, along with payments $\pi(R)$ according to the deployed payment rule (Line 16). MLHCA is plug-and-play compatible with many different payment rules. If, by the end of the ML-powered DQs the market has not cleared we switch to VQ rounds. In the first VQ round (Line 18) we ask each bidder for her *bridge bid* (see Definition 4). This single VQ bid ensures that the MLHCA's efficiency is lower bounded by the efficiency after just the DQ rounds (Lemma 4). In the final $Q^{\text{VQ}} - 1$ VQ rounds, for each bidder, we query her her value for the bundle she is allocated in the predicted optimal allocation (based on all ML models), under the constraint that she has not answered a VQ for that bundle in the past.[12] The final allocation and payments are then determined based on all reports (Lines 25 to 26). Note that ML-CCA can be combined with various possible payment rules $\pi(R)$, such as VCG or VCG-nearest. We present the detailed description of the mechanism in Appendix H.

6 Experiments

In this section, we experimentally evaluate MLHCA. We compare its efficiency against BOCA (Weissteiner et al., 2023) and ML-CCA (Soumalias et al., 2024a) the SOTA VQ-based and DQ-based ICAs, respectively.

[12] This is the VQ generation algorithm that was first suggested in Brero et al. (2021) and used in all follow-up work.

6.1 Experiment Setup

To generate synthetic CA instances, we use the *spectrum auction test suite (SATS)* (Weiss et al., 2017). SATS gives us access to the true optimal allocation $a^* \in \mathcal{F}$, which we use to measure the efficiency loss, i.e., $1 - V(a^*)/V(a^*(R))$ when eliciting reports R. As in all mechanisms we compare against (e.g. (Soumalias et al., 2024b; Weissteiner et al., 2023)), we focus of efficiency (and not revenue). The main application of ICAs is in spectrum allocation, a government-run operation with a mandate to maximize welfare (Cramton, 2013). See Appendix J for a discussion of the corresponding results on revenue. To enable a fair comparison against prior work, we use 100 total queries for all auction mechanisms. Those are 100 VQs for BOCA, 100 DQs for ML-CCA, and 40 DQs and 60 VQs for MLHCA. For BOCA and ML-CCA, we use the best mechanism configuration and hyperparameters as reported in their corresponding papers.

For MLHCA's VQ rounds, we performed HPO separately for each bidder type in each domain, as detailed in Section 4.2. For the DQ rounds, we adopted the HPO parameters reported by Soumalias et al. (2024b), since our learning algorithm, when restricted to DQs, is equivalent to theirs. For further details, please refer to Appendix K.1.

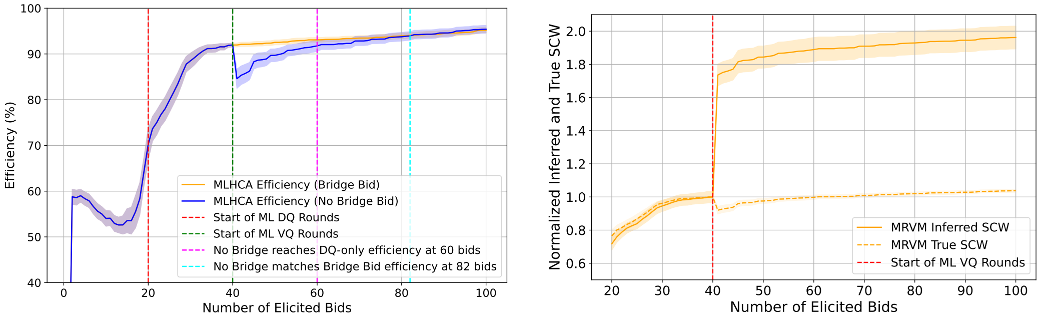
6.2 Evaluating the Effectiveness of the Bridge Bid

In this section, we experimentally evaluate the effectiveness of the bridge bid from Section 3.

In Figure 2a, we plot MLHCA's efficiency in the MRVM domain as a function of the number of elicited bids, comparing performance with and without the bridge bid. Without the bridge bid, we observe a significant efficiency drop of 7.3% points when MLHCA transitions to its VQ rounds. This

Algorithm 1: MLHCA($Q^{\text{CCA}}, Q^{\text{DQ}}, Q^{\text{VQ}}, \pi$)

```
Parameters: $Q^{\text{CCA}}$, $Q^{\text{DQ}}$, $Q^{\text{VQ}}$ and $\pi$
$R^{VQ} \leftarrow (\{\})_{i=1}^N$
$R^{DQ} \leftarrow (\{\})_{i=1}^N$
for $r = 1, ..., Q^{CCA}$ do                                        ▷ Draw $Q^{\text{CCA}}$ initial prices
    $p^r \leftarrow CCA(R^{DQ})$
    foreach $i \in N$ do                                             ▷ Initial demand query responses
        $R_i^{DQ} \leftarrow R_i^{DQ} \cup \{(x_i^*(p^r), p^r)\}$
for $r = Q^{CCA} + 1, ..., Q^{CCA} + Q^{DQ}$ do                       ▷ ML-powered DQs
    foreach $i \in N$ do
        $\mathcal{M}_i^\theta \leftarrow$ MixedTraining($R_i^{DQ} \cup R_i^{VQ}$)        ▷ Algorithm 4
    $p^r \leftarrow$ NextPrice($(\mathcal{M}_i^\theta)_{i=1}^n$)                        ▷ Appendix C
    foreach $i \in N$ do                                             ▷ Demand query responses for $p^r$
        $R_i^{DQ} \leftarrow R_i^{DQ} \cup \{(x_i^*(p^r), p^r)\}$
    if $\sum_{i=1}^{n} (x_i^*(p^k))_j = c_j \; \forall j \in M$ then               ▷ Market-clearing prices found
        Set final allocation $a^*(R^{DQ} \cup R^{VQ}) \leftarrow (x_i^*(p^r))_{i=1}^n$
        Calculate payments $\pi(R^{DQ} \cup R^{VQ}) \leftarrow (\pi_i(R^{DQ} \cup R^{VQ}))_{i=1}^n$
        return $a^*(R^{DQ} \cup R^{VQ})$ and $\pi(R^{DQ} \cup R^{VQ})$
foreach $i \in N$ do                                                 ▷ Bridge bid
    $R_i^{VQ} \leftarrow R_i^{VQ} \cup \{(a_i^*(R^{DQ} \cup R^{VQ}), v_i(a_i^*(R^{DQ} \cup R^{VQ})))\}$
for $r = Q^{CCA} + Q^{DQ} + 2, ..., Q^{CCA} + Q^{DQ} + Q^{VQ}$ do      ▷ ML-powered VQs
    foreach $i \in N$ do
        $\mathcal{M}_i^\theta \leftarrow$ MixedTraining($R_i^{DQ} \cup R_i^{VQ}$)        ▷ Section 4.1
    $a \leftarrow$ NextAllocation $\left(\left(\mathcal{M}_i^\theta\right)_{i=1}^n\right), R^{DQ} \cup R^{VQ}\right)$       ▷ Appendix H
    foreach $i \in N$ do
        $R_i^{VQ} \leftarrow R_i^{VQ} \cup \{(a_i, v_i(a_i))\}$                    ▷ Value query responses
Calculate final allocation $a^*(R^{DQ} \cup R^{VQ})$ as in Equation (3)
Calculate payments $\pi(R^{DQ} \cup R^{VQ})$                          ▷ E.g., VCG (Appendix A)
return $a^*(R^{DQ} \cup R^{VQ})$ and $\pi(R^{DQ} \cup R^{VQ})$
```



(a) Efficiency of MLHCA with and without the bridge bid (Definition 4) in the MRVM domain.

(b) Normalized inferred and true SCW of MLHCA without the bridge bid in the MRVM domain. Both quantities are normalized with respect to their average values at the start of the ML-based VQs.

Figure 2: Comparison of MLHCA's performance in the MRVM domain: (a) Efficiency with and without the bridge bid; (b) Normalized inferred and true SCW. Shown are averages with 95% CIs.

is consistent with our theoretical results in Lemma 3, where we showed that efficiency can decrease when the first VQ is introduced after DQs. In the MRVM domain, the most realistic setting, this effect is particularly pronounced. Notably, the auction requires 20 of our powerful ML-powered VQs just to recover the efficiency lost by the introduction of the first VQ. By contrast, using the bridge bid (Definition 4) completely mitigates this efficiency drop, as predicted by Lemma 4. However, as Figure 2a shows, if enough VQs are elicited, MLHCA without the bridge bid can eventually recover its efficiency, and both approaches converge to similar performance levels.

However, given that the auctioneer cannot determine the true efficiency of the auction at runtime, it is prudent to use the bridge bid version, which ensures consistent performance throughout the auction and significantly outperforms the alternative for the majority of rounds. Therefore, we consider this version the default approach for our MLHCA.

To better understand the cause of this efficiency drop, we refer to Figure 2b, where we plot the normalized inferred and true social welfare of MLHCA without the bridge bid in the MRVM domain. Both quantities are normalized to their values at the start of the ML-powered VQ rounds. At this point, we observe a stark contrast: the first VQ increases inferred social welfare by over 70%, while decreasing true social welfare by more than 7%. Before the ML-powered VQs, agents' reports were limited to their responses to DQs, and the auction's inferred social welfare was calculated based on the prices of the allocated bundles, as described in Equation (3). Due to the relatively low competition in MRVM, there was a substantial gap between the agents' true values and the inferred values based on their DQ responses.[13]

When the auction transitioned to VQs, agents responded with their true values for the queried bundles, leading to a sharp increase in inferred social welfare. However, the bidders' true values for the bundles they received during the DQ rounds were much higher than their inferred values, which the WDP failed to capture. As a result, transitioning to ML-powered VQs without the bridge bid caused a sharp increase in *inferred* social welfare alongside a drop in *true* social welfare.

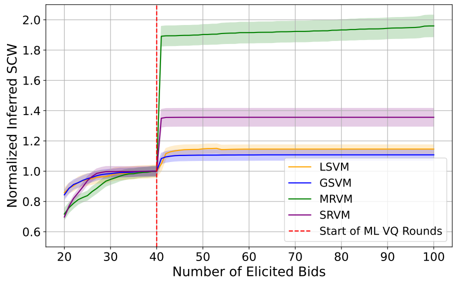


Figure 3: Normalized Inferred Social Welfare of MLHCA with and without the bridge bid (Definition 4) in the MRVM domain. Shown are averages with 95% CIs.

This efficiency drop when transitioning to VQs is less pronounced in other domains. In Figure 3, we plot MLHCA's SCW for all domains as a function of the number of elicited bids, normal-

[13] Low competition in the auction can be gauged from its revenue, as, in the absence of reserve prices, revenue is primarily driven by competition among bidders. MRVM has the lowest ratio of revenue to welfare across all domains by a factor of nearly 2; see Appendix J.

	Efficiency Loss in %					Queries to Reject Null Hypothesis		
Domain	MLHCA	BOCA	ML-CCA$_{\text{clock}}$	ML-CCA$_{\text{raised}}$	CCA	BOCA $\geq$ MLHCA	ML-CCA$_{\text{clock}}$ $\geq$ MLHCA	ML-CCA$_{\text{raised}}$ $\geq$ MLHCA
GSVM	0.00 ± 0.00	—	1.77 ± 0.68	1.07 ± 0.37	9.60 ± 1.49	—	42	42
LSVM	0.04 ± 0.07	0.39 ± 0.31	8.36 ± 1.70	3.61 ± 0.77	17.44 ± 1.60	58	42	43
SRVM	0.00 ± 0.00	0.06 ± 0.02	0.41 ± 0.11	0.07 ± 0.02	0.37 ± 0.11	42	42	42
MRVM	4.81 ± 0.57	7.77 ± 0.35	6.94 ± 0.24	6.68 ± 0.22	7.53 ± 0.48	54	74	79

Table 2: MLHCA (40DQs + 60VQs) vs BOCA (100VQs), ML-CCA (ML-CCA$_{\text{clock}}$) (100DQs) and ML-CCA with raised clock bids (ML-CCA$_{\text{raised}}$) (100DQs and up to 100VQs). Shown are averages and a 95% CI. Winners based on a t-test with significance level of 5% are marked in grey.

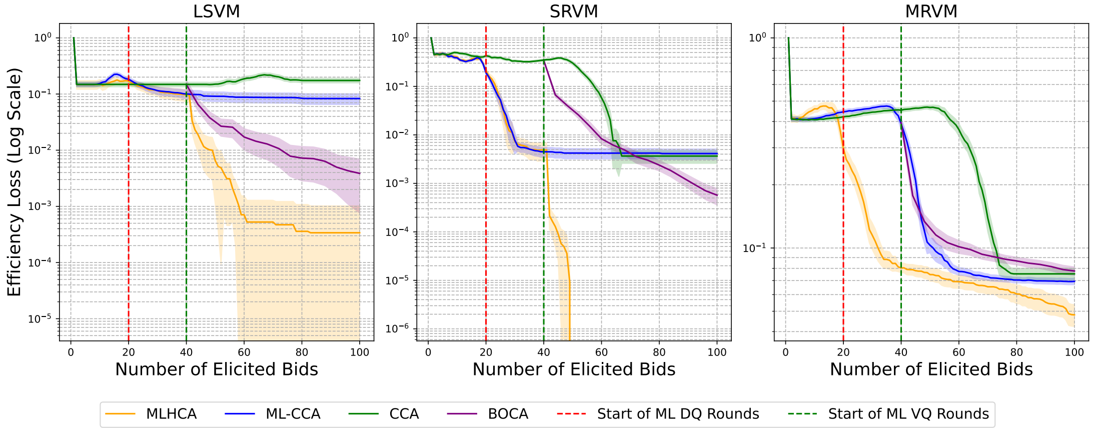


Figure 4: Efficiency loss paths (i.e., regret plots) of MLHCA compared to BOCA, ML-CCA and CCA. Shown are averages over 50 instances with 95% CIs.

ized to the start of the ML-powered VQ rounds. In these domains, the higher level of competition leads to inferred values for the queried bundles during the DQ rounds being much closer to the true values. Consequently, the bridge bid is less critical in these settings.

6.3 Efficiency Results

In this section, we present the efficiency of MLHCA, comparing its performance against the current alternatives discussed in Section 2.2. These results build on the theoretical insights discussed in Sections 3 and 4, showcasing both the advantages of starting with DQs and the efficacy of our hybrid approach.

In Table 2, we show the average efficiency loss of each mechanism after 100 queries. For ML-CCA, we also report results if it were supplemented with the clock bids raised heuristic (see Section 2.2), which would involve up to an additional 100 VQs per bidder.[14] Finally, we report the number of queries that MLHCA requires to outperform the final efficiency of each other mechanism, i.e., in GSVM, with 42 queries (40 DQs and 2 VQs) MLHCA statistically outperforms ML-CCA, even if ML-CCA were supplemented with 100 VQs from the clock bids raised heuristic.

In Table 2, we observe that across all domains, MLHCA significantly outperforms all other mechanisms. First, MLHCA is the *only* mechanism that can achieve a perfect 100% efficiency in SRVM. As a matter of fact, it can do this using less than 60 queries, while the other auctions cannot do that even with 100 queries. In the LSVM domain, MLHCA achieves a 10-fold reduction in efficiency loss compared to BOCA, the previous SOTA. But the most realistic domain is MRVM, designed to simulate the data from the 2014 Canadian spectrum auction (Weiss et al., 2017). In that domain, MLHCA is the first mechanism to cause a significant efficiency increase versus the CCA, increasing efficiency compared to all other mechanisms by over 2.5% points. If MLHCA were used in the latest Canadian Spectrum Auction, based on the value of goods traded (Innovation, Science and Economic

[14] In the clock bids raised heuristic, the bidders only need to report their value for each *unique* bundle they bid on during the auction, which, for 100 DQs, can be up to 100 bundles.

Development Canada, 2023), this alone would have translated in welfare gains of over 50 million USD versus the three other mechanisms studied here.

But in these auctions, speed of convergence is also important. In all domains, MLHCA needs at most 58 queries (40 DQs and 18 VQs) to statistically outperform both the BOCA and the ML-CCA final efficiency, which would require 100 VQs and 100 DQs, respectively. Additionally, in three out of four domains, MLHCA outperforms the 100 DQ efficiency of ML-CCA using only 40 DQs and 2 VQs. This result aligns with our theoretical analysis in Section 3.3, where we provided an example showcasing that, after the DQs have served to learn the bidder's value functions sufficiently well, a single VQ can lead to 100% efficiency.

Figure 4 shows the efficiency loss path for all domains. We see that MLHCA's superior (average) performance does not only hold at the end of the auction, but also for a large range of queries: in *all* domains, MLHCA outperforms ML-CCA in queries $\{42, \ldots, 100\}$ and BOCA in queries $\{1, \ldots, 100\}$. These results align with our theoretical analysis. Up to query 40, MLHCA is identical to ML-CCA, so the two mechanisms achieve identical efficiency. After query 40, the superior learning performance of combining DQs with VQs (as shown in Section 4) combined with the superior efficiency of asking VQs (Section 3) means that MLHCA outperforms ML-CCA. Up to query 40, MLHCA outperforms BOCA because of the auction and learning advantages of DQs at the early stages of the auction when the ML models are not yet sufficiently accurate. After round 40 MLHCA outperforms BOCA because of the far superior learning performance of networks trained on both DQs and VQs compared to only VQs.

Figure 4 clearly demonstrates how our results align with the theoretical insights from Sections 3 and 4. First, the efficiency loss path of CCA highlights the non-monotonicity of DQ-based mechanisms, as proven in Lemma 2. Notably, in the LSVM domain, the CCA achieves higher efficiency after just 5 DQs compared to after 100. Next, by comparing the efficiency of BOCA with that of ML-CCA after 40 elicited bids, we observe the much lower efficiency of random VQs compared to DQs, as highlighted in Lemma 1. This discrepancy is especially pronounced in the MRVM domain, where the efficiency loss of BOCA after the initial random VQs is orders of magnitude worse compared to mechanisms that employ ML-powered VQs. Finally, comparing ML-CCA with ML-HCA highlights the potential efficiency gains of supplementing DQs with VQs. Both ML-CCA and MLHCA show identical efficiency loss paths for the first 40 elicited bids, as they employ the same network configuration and identical DQs during these rounds. However, once MLHCA switches to ML-powered VQs after 40 queries, we see an immediate and significant drop in efficiency loss—by several orders of magnitude in both the LSVM and SRVM domains. In contrast, the efficiency of the DQ-based ML-CCA remains stagnant. This observation aligns with our theoretical results in Theorem 1. Once the ML models have effectively learned the bidders' value functions, the introduction of VQs yields a dramatic reduction in efficiency loss. However, the DQ-based ML-CCA, despite having well-trained models, lacks the ability to leverage this information further, preventing any significant efficiency improvement beyond this point.

In summary, MLHCA outperforms both DQ-based and VQ-based SOTA mechanisms in terms of both efficiency and speed of convergence, achieving high efficiency with fewer queries. This makes MLHCA a powerful and practical choice for real-world auction scenarios where high efficiency and rapid convergence are crucial.

7 Conclusion

We have introduced MLHCA, the first ICA that combines ML-powered VQ and DQ generation algorithms. MLHCA provably incorporates the full information that both query types provide, and leverages the theoretical and practical insights developed in this work to combine these queries effectively and achieve unprecedented efficiency-clearly surpassing current SOTA mechanisms.

Our results demonstrate that MLHCA consistently outperforms previous SOTA mechanisms across all tested domains, achieving substantial efficiency gains with significantly fewer queries. Notably, MLHCA reduces efficiency loss by up to a factor of 10 compared to the previous SOTA while surpassing all previous mechanisms with at most 74% of their queries. In the most realistic domain, MLHCA's efficiency gains translate into welfare improvements exceeding 50 million USD in a single auction instance. Importantly, MLHCA achieves these gains while simplifying bidder participation: compared to the previous SOTA VQ-based mechanism, MLHCA primarily relies on the more practical DQs, requiring only a few cognitively demanding VQs to reach similar efficiency levels. Compared to the SOTA DQ-based mechanism, MLHCA can achieve equivalent efficiency with just 40% of the DQs and only two VQs in most domains, eliminating the need for a supplementary round and thus streamlining the bidding process.

In conclusion, by effectively integrating both query types, MLHCA sets a new benchmark in both allocative efficiency and speed of convergence. This work lays the foundation for future combinatorial auction designs, where ML techniques not only enhance efficiency but also simplify the bidding process, ultimately increasing bidder participation and thus impact potential.

REFERENCES

Lawrence M Ausubel and Oleg Baranov. A practical guide to the combinatorial clock auction. *Economic Journal*, 127(605):F334–F350, 2017.

Lawrence M Ausubel and Oleg Baranov. Iterative vickrey pricing in dynamic auctions, 2019.

Lawrence M. Ausubel and Oleg Baranov. Revealed preference and activity rules in dynamic auctions. *International Economic Review*, 61(2):471–502, 2020. doi: https://doi.org/10.1111/iere.12431. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/iere.12431.

Lawrence M. Ausubel and Oleg V. Baranov. Market design and the evolution of the combinatorial clock auction. *The American Economic Review*, 104(5):446–451, 2014. ISSN 00028282. URL http://www.jstor.org/stable/42920978.

Lawrence M Ausubel, Peter Cramton, and Paul Milgrom. The clock-proxy auction: A practical combinatorial auction design. In Peter Cramton, Yoav Shoham, and Richard Steinberg (eds.), *Combinatorial Auctions*, pp. 115–138. MIT Press, 2006.

Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. Generalization guarantees for multi-item profit maximization: Pricing, auctions, and randomized mechanisms, 2023.

Martin Bichler, Zhen Hao, and Gediminas Adomavicius. *Coalition-based pricing in ascending combinatorial auctions*, pp. 493–528. Cambridge University Press, October 2017. ISBN 9781107135345. doi: 10.1017/9781316471609.025.

Sushil Bikhchandani and Joseph M Ostroy. The package assignment model. *Journal of Economic theory*, 107(2):377–406, 2002.

Avrim Blum, Jeffrey Jackson, Tuomas Sandholm, and Martin Zinkevich. Preference elicitation and query learning. *Journal of Machine Learning Research*, 5:649–667, 2004.

Gianluca Brero and Sébastien Lahaie. A bayesian clearing mechanism for combinatorial auctions. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018.

Gianluca Brero, Benjamin Lubin, and Sven Seuken. Combinatorial auctions via machine learning-based preference elicitation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018.

Gianluca Brero, Sébastien Lahaie, and Sven Seuken. Fast iterative combinatorial auctions via bayesian learning. In *Proceedings of the 33rd AAAI Conference of Artificial Intelligence*, 2019.

Gianluca Brero, Benjamin Lubin, and Sven Seuken. Machine learning-powered iterative combinatorial auctions. *arXiv preprint arXiv:1911.08042*, Jan 2021.

Richard Cole and Tim Roughgarden. The sample complexity of revenue maximization. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pp. 243–252, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450327107. doi: 10.1145/2591796.2591867. URL https://doi.org/10.1145/2591796.2591867.

Peter Cramton. Spectrum auction design. *Review of Industrial Organization*, 42(2):161–190, 2013.

Paul Dütting, Felix Fischer, Pichayut Jirapinyo, John K Lai, Benjamin Lubin, and David C Parkes. Payment rules through discriminant-based classifiers. *ACM Transactions on Economics and Computation*, 3(1):5, 2015.

Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David C Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Benjamin Estermann, Stefan Kramer, Roger Wattenhofer, and Ye Wang. Deep learning-powered iterative combinatorial auctions with active learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pp. 2919–2921, 2023.

Andor Goetzendorff, Martin Bichler, Pasha Shabalin, and Robert W. Day. Compact bid languages and core pricing in large multi-item auctions. *Management Science*, 61(7):1684–1703, 2015. doi: 10.1287/mnsc.2014.2076. URL https://doi.org/10.1287/mnsc.2014.2076.

Noah Golowich, Harikrishna Narasimhan, and David C Parkes. Deep learning for multi-facility location mechanism design. In *Proceedings of the Twenty-seventh International Joint Conference on Artificial Intelligence and the Twenty-third European Conference on Artificial Intelligence*, pp. 261–267, 2018.

Jakob Heiss. *Inductive Bias of Neural Networks and Selected Applications*. Doctoral thesis, ETH Zurich, Zurich, 2024. URL https://www.research-collection.ethz.ch/handle/20.500.11850/699241.

Jakob Heiss, Josef Teichmann, and Hanna Wutte. How implicit regularization of Neural Networks affects the learned function – Part I, November 2019. URL https://arxiv.org/abs/1911.02903.

Jakob Heiss, Josef Teichmann, and Hanna Wutte. How infinitely wide neural networks can benefit from multi-task learning - an exact macroscopic characterization. *arXiv preprint arXiv:2112.15577*, 2021. doi: 10.3929/ETHZ-B-000550890. URL https://arxiv.org/abs/2112.15577.

Jakob Heiss, Josef Teichmann, and Hanna Wutte. How (implicit) regularization of relu neural networks characterizes the learned function – part ii: the multi-d case of two layers with random first layer, 2023. URL https://arxiv.org/abs/2303.11454.

Innovation, Science and Economic Development Canada. 3800 mhz auction - provisional results, 2023. URL https://ised-isde.canada.ca/site/spectrum-management-telecommunications/en/spectrum-allocation/3800-mhz-auction-provisional-results#t1. Accessed: 2024-10-08.

Anthony M. Kwasnica, John O. Ledyard, Dave Porter, and Christine DeMartini. A new and improved design for multiobject iterative auctions. *Management Science*, 51(3):419–434, 2005. ISSN 00251909, 15265501. URL http://www.jstor.org/stable/20110340.

Sébastien Lahaie and Benjamin Lubin. Adaptive-price combinatorial auctions. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC ’19, pp. 749–750, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367929. doi: 10.1145/3328526.3329615. URL https://doi.org/10.1145/3328526.3329615.

Sebastien M Lahaie and David C Parkes. Applying learning algorithms to preference elicitation. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, 2004.

Benjamin Lubin, Sven Seuken, Manuel Beyeler, and Gianluca Brero. imlca: Machine learning-powered iterative combinatorial auctions with interval bidding, 2021. URL https://arxiv.org/abs/2009.13605.

Ryota Maruo and Hisashi Kashima. Efficient preference elicitation in iterative combinatorial auctions with many participants, 2024. URL https://arxiv.org/abs/2403.19075.

Paul Milgrom and Ilya Segal. Designing the us incentive auction. *Handbook of spectrum auction design*, pp. 803–812, 2017.

Jamie Morgenstern and Tim Roughgarden. The pseudo-dimension of near-optimal auctions. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS’15, pp. 136–144, Cambridge, MA, USA, 2015. MIT Press.

Harikrishna Narasimhan, Shivani Brinda Agarwal, and David C Parkes. Automated mechanism design without money via machine learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016.

Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129(1):192–224, 2006.

Greg Ongie, Rebecca Willett, Daniel Soudry, and Nathan Srebro. A function space view of bounded norm infinite width relu nets: The multivariate case. *arXiv preprint arXiv:1910.01635*, 2019. URL https://arxiv.org/pdf/1910.01635.pdf.

Rahul Parhi and Robert D Nowak. What kinds of functions do deep neural networks learn? insights from variational spline theory. *SIAM Journal on Mathematics of Data Science*, 4(2):464–489, 2022.

Stephen J Rassenti, Vernon L Smith, and Robert L Bulfin. A combinatorial auction mechanism for airport time slot allocation. *The Bell Journal of Economics*, pp. 402–417, 1982.

Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? *arXiv preprint arXiv:1902.05040*, 2019. URL https://arxiv.org/abs/1902.05040.

Ermis Soumalias, Behnoosh Zamanlooy, Jakob Weissteiner, and Sven Seuken. Machine learning-powered course allocation. *arXiv preprint arXiv:2210.00954*, 2024a.

Ermis Nikiforos Soumalias, Jakob Weissteiner, Jakob Heiss, and Sven Seuken. Machine learning-powered combinatorial clock auction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(9):9891–9900, Mar. 2024b. doi: 10.1609/aaai.v38i9.28850. URL https://ojs.aaai.org/index.php/AAAI/article/view/28850.

Michael Weiss, Benjamin Lubin, and Sven Seuken. Sats: A universal spectrum auction test suite. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pp. 51–59, 2017.

Jakob Weissteiner. *Integrating advanced machine learning methods into market mechanisms*. PhD thesis, University of Zurich, 2023.

Jakob Weissteiner and Sven Seuken. Deep learning—powered iterative combinatorial auctions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02):2284–2293, Apr. 2020. doi: 10.1609/aaai.v34i02.5606. URL https://ojs.aaai.org/index.php/AAAI/article/view/5606.

Jakob Weissteiner, Jakob Heiss, Julien Siems, and Sven Seuken. Monotone-value neural networks: Exploiting preference monotonicity in combinatorial assignment. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 541–548. International Joint Conferences on Artificial Intelligence Organization, 7 2022a. doi: 10.24963/ijcai.2022/77. URL https://doi.org/10.24963/ijcai.2022/77. Main Track.

Jakob Weissteiner, Chris Wendler, Sven Seuken, Ben Lubin, and Markus Püschel. Fourier analysis-based iterative combinatorial auctions. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 549–556. International Joint Conferences on Artificial Intelligence Organization, 7 2022b. doi: 10.24963/ijcai.2022/78. URL https://doi.org/10.24963/ijcai.2022/78. Main Track.

Jakob Weissteiner, Jakob Heiss, Julien Siems, and Sven Seuken. Bayesian optimization-based combinatorial assignment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37, 2023.

Francis Williams, Matthew Trager, Daniele Panozzo, Claudio Silva, Denis Zorin, and Joan Bruna. Gradient dynamics of shallow univariate relu networks. In *Advances in Neural Information Processing Systems*, pp. 8378–8387, 2019. URL http://papers.nips.cc/paper/9046-gradient-dynamics-of-shallow-univariate-relu-networks.pdf.

A Payment and Activity Rules

In this section, we reprint the VCG and VCG-nearest payment rules, as well as give an overview of activity rules for the CCA, and argue why the most prominent choices are also applicable to our MLHCA.

A.1 VCG Payments

Definition 5. (VCG Payments from Demand And Value Query Data) *Let $R = (R_1, \dots, R_n)$ denote an elicited set of both demand and value query data from each bidder and let $R_{-i} := (R_1, \dots, R_{i-1}, R_{i+1}, \dots, R_n)$. We then calculate the VCG payments $\pi^{VCG}(R) = (\pi_1^{VCG}(R) \dots, \pi_n^{VCG}(R)) \in \mathbb{R}^n_{\geq 0}$ as follows:*

$$\pi_i^{VCG}(R) := \sum_{j \in N \setminus \{i\}} \widetilde{v}_j\left(a^*(R_{-i})_j; R_j\right) - \sum_{j \in N \setminus \{i\}} \widetilde{v}_j\left(a^*(R)_j; R_j\right). \tag{4}$$

where $a^*(R_{-i})$ is the allocation that maximizes the inferred social welfare when excluding bidder i, i.e.,

$$a^*(R_{-i}) \in \operatorname*{arg\,max}_{a \in \mathcal{F}} \sum_{j \in N \setminus \{i\}} \widetilde{v}_j(a_j; R_j), \tag{5}$$

and $a^*(R)$ is the inferred social welfare-maximizing allocation (see Equation (3)).

Thus, when using VCG payments, bidder i's utility is:

$$\begin{aligned} u_i &= v_i(a^*(R)_i) - \pi_i^{\text{VCG}}(R) \\ &= v_i(a^*(R)_i) + \sum_{j \in N \setminus \{i\}} \widetilde{v}_j\left(a^*(R)_j; R_j\right) - \sum_{j \in N \setminus \{i\}} \widetilde{v}_j\left(a^*(R_{-i})_j; R_j\right). \end{aligned}$$

A.2 VCG-Nearest Payments

To define the VCG-nearest payments, we must first introduce the core:

Definition 6. (The Core) *An outcome $(a, \pi) \in \mathcal{F} \times \mathbb{R}^n_{\geq 0}$ (i.e., a tuple of a feasible allocation a and payments π) is in the core if it satisfies the following two properties:*

1. *The outcome is* individual rational, *i.e, $u_i = v_i(a_i) - \pi_i \geq 0$ for all $i \in N$*
2. *The core constraints*

$$\forall\, L \subseteq N \sum_{i \in N \setminus L} \pi_i(R) \geq \max_{a' \in \mathcal{F}} \sum_{i \in L} v_i(a'_i) - \sum_{i \in L} v_i(a_i) \tag{6}$$

where $v_i(a_i)$ is bidder i's value for bundle a_i and $\mathcal{F}$ is the set of feasible allocations.

In words, a payment vector π (together with a feasible allocation a) is in the core if no coalition of bidders $L \subset N$ is willing to pay more for the items than the mechanism is charging the winners. Note that by replacing the true values $v_i(a_i)$ with the bidders' (possibly untruthful) *inferred values* based on their reports $\widetilde{v}_i(a_i; R_i)$ in Definition 6 one can equivalently define the *revealed core*.

Now, we can define

Definition 7. (Minimum Revenue Core) *Among all payment vectors in the (revealed) core, the (revealed) minimum revenue core is the set of payment vectors with smallest L_1-norm, i.e., which minimize the sum of the payments of all bidders.*

We can now define VCG-nearest payments:

Definition 8. (VCG-Nearest Payments) *Given an allocation a_R for bidder reports R, the VCG-nearest payments $\pi^{VCG\text{-}nearest}(R)$ are defined as the vector of payments in the (revealed) minimum revenue core that minimizes the L_2-norm to the VCG payment vector $\pi^{VCG}(R)$.*

A.3 On the Importance of Activity Rules to Align Incentives

In the CCA, activity rules serve multiple purposes. First, they help accelerate the auction process. Second, they reduce ”bid-sniping” opportunities—bidders concealing their true intentions until the very last rounds of the auction.[15] Third, they limit surprise bids in the supplementary round of the CCA, significantly reducing a bidder's ability to drive up opponents' payments by overbidding on bundles they cannot win (Ausubel & Baranov, 2017). There are two types of activity rules that are implemented in a CCA:

1. *Clock phase activity rules*, which limit the bundles that an agent can bid on during the clock phase, based on their bids in previous clock rounds.
2. *Supplementary round activity rules*, which restrict the amounts that an agent can bid on specific sets of items during the supplementary round.

Traditionally, most clock phase activity rules in the CCA have relied on either revealed-preference principles or a *points-based system*, where points are assigned to each item before the auction, and bidders are only allowed to submit monotonically non-increasing bids in terms of points. In other words, as prices rise across rounds, bidders cannot submit bids for larger sets of items. Both of these approaches, as well as hybrid combinations thereof, were shown to actually further interfere with truthful bidding in some cases (Ausubel & Baranov, 2014; 2020).

However, Ausubel & Baranov (2019) showed that basing clock phase activity rules entirely on the *generalized axiom of revealed preference (GARP)* can dynamically approximate VCG payoffs, thus improving the bidding incentives of the CCA. GARP imposes revealed-preference constraints (see Definition 9) on the bidder's demand responses. The GARP activity rule requires that the bidder demonstrates rational behavior in her demand choices, without necessitating a monotonic price trajectory. As a result, it can also be applied during the ML-powered DQ phase of MLHCA, allowing our mechanism to enjoy similar improvements in bidding incentives.

For the supplementary round, the CCA's most prominent activity rules are again based on a combination of points-based systems and revealed-preference ideas, which we outline below:

Definition 9. (Revealed-preference constraint) *The revealed-preference constraint for bundle $x \in X$ with respect to clock round r is*

$$b_i(x) \leq b_i(x^r) + \langle p^r, x - x^r \rangle \,, \tag{7}$$

where $b_i(x) \in \mathbb{R}_{\geq 0}$ is bidder i's bid for bundle $x \in X$ in the supplementary round, $x^r \in \mathcal{X}$ is the bundle demanded by the agent at clock round r, $b_i(x^r) \in \mathbb{R}_{\geq 0}$ is the final bid for bundle $x^r \in \mathcal{X}$ and $p^r \in \mathbb{R}^m_{\geq 0}$ is the linear price vector of clock round r.

Intuitively, the revealed-preference constraint ensures that a bidder cannot claim a higher value for bundle x relative to bundle x^r, given that they expressed a preference for bundle x^r at the given prices p^r (see Equation (1)). The difference between the three most prominent supplementary round activity rules is with respect to *which clock rounds* the revealed-preference constraint should be satisfied. Specifically:

1. *Final Cap:* A bid for bundle $x \in \mathcal{X}$ should satisfy the *revealed-preference constraint (Definition 9)* with respect to the *final* clock round's price $p^{Q^{\max}} \in \mathbb{R}_{\geq 0}$ and bundle $x^{Q^{\max}} \in \mathcal{X}$.
2. *Relative Cap:* A bid for bundle $x \in \mathcal{X}$ should satisfy the *revealed-preference constraint (Definition 9)* with respect to the last clock round for which the bidder was eligible for that bundle $x \in \mathcal{X}$, based on the points-based system.
3. *Intermediate Cap:* A bid for bundle $x \in \mathcal{X}$ should satisfy the *revealed-preference constraint (Definition 9)* with respect to all eligibility-reducing rounds, starting from the last clock round for which the bidder was eligible for $x \in \mathcal{X}$ based on the point system.

Ausubel & Baranov (2017) showed that combining the *Final Cap* and *Relative Cap* activity rules leads to the largest amount of reduction in bid-sniping opportunities for the UK 4G auction, as

[15] The notion of ”bid-sniping” originated in eBay auctions with predetermined ending times, where high-value bidders could reduce their payments by submitting bids at the very last moment.

measured by the theoretical bid amount that each bidder would need to increase her bid by in the supplementary round in order to protect her final clock round bundle. Finally, note that the *Final-* and *Intermediate Cap* activity rules can also be applied to the ML-powered DQ phase of our MLHCA.[16]

To conclude, both the DQ and VQ phases of MLHCA are compatible with the most prominent activity rules of the CCA, and MLHCA also remains compatible with the commonly used VCG-nearest pricing rule (Definition 8). Combined with MLHCA's similar interaction paradigm to the CCA, these aspects provide strong evidence that our mechanism can leverage activity rules to effectively mitigate bidder misreporting opportunities, much like the classical CCA.

A.4 On the Importance of Marginal Economies to Align Incentives

In this section, we review the key arguments from Brero et al. (2021) on why MLCA provides strong incentives for truthful reporting in practice. These arguments extend to any ML-powered ICA that employs the same VQ-generation algorithm, including MLHCA.

Bidder i's utility in MLCA (and MLHCA) under VCG payments (see Definition 5) can be expressed as:

$$\begin{aligned} u_i &= v_i(a^*(R)_i) - \pi_i^{\text{VCG}}(R) \\ &= \underbrace{v_i(a^*(R)_i) + \sum_{j \in N \setminus \{i\}} \widetilde{v}_j\left(a^*(R)_j; R_j\right)}_{\text{(a)}} - \underbrace{\sum_{j \in N \setminus \{i\}} \widetilde{v}_j\left(a^*(R_{-i})_j; R_j\right)}_{\text{(b) Inferred SW of marginal economy}} . \end{aligned}$$

Any beneficial misreport by bidder i must increase the difference (a) $-$ (b).

MLCA has two features that mitigate manipulations. First, MLCA explicitly queries each bidder's marginal economy (Algorithm 3, Line 5), which implies that (b) is practically independent of bidder i's reports. Experimental evidence supporting this claim is provided in Section 7.3 of Brero et al. (2021). Second, MLCA (and also MLHCA) enables bidders to "push" information to the auction which they deem useful. This mitigates certain manipulations that target (a), as it allows bidders to increase (a) with truthful information. Brero et al. (2021) argue that any remaining manipulation would be implausible as it would require almost complete information.

Under further assumptions, we can also derive two theoretical incentive guarantees:

- Assumption 1 requires that, for all bidders $i \in N$, if all other bidders report truthfully, then the reported social welfare of bidder i's marginal economy (i.e., term (b)) is *independent* of her value reports.
- Assumption 2 requires that, if all bidders $i \in N$ bid truthfully, then MLCA *finds an efficient allocation*.

Result 1: Social Welfare Alignment Under Assumption 1, and given that all other bidders are truthful, MLCA is *social welfare aligned*. This means that the only way for a bidder to increase her true utility is by increasing the reported social welfare of $a^*(R)$ in the main economy (i.e., term (a)), which, in this case, equals the true social welfare of $a^*(R)$ (Brero et al., 2021, Proposition 3). The same is true for the VQ phase of MLHCA, as it employs the same allocation and payment rules.

Result 2: Ex-Post Nash Equilibrium If both Assumption 1 and Assumption 2 hold, then bidding truthfully constitutes an ex-post Nash equilibrium in MLCA (Brero et al., 2021, Proposition 4). The same is true for the VQ phase of MLHCA, as it employs the same allocation and payment rules.

To conclude, MLHCA's compatibility with both *activity rules* during its DQ rounds and *marginal economies* during its VQ rounds, as well as its compatibility with VCG and VCG-nearest payments, provides strong evidence that MLHCA can effectively mitigate opportunities for bidder misreporting.

[16] Soumalias et al. (2024b) argued that with the modification for the *Relative Cap* rule that the revealed-preference constraint should hold for the Q^{CCA} rounds that follow the same price update rule as the CCA, and then the ML-powered clock rounds should be treated as corresponding to the same amount of points, since the prices in these rounds on aggregate stay very close to the prices of the last Q^{init} round.

B A Machine Learning-Powered ICA

In this section, we present in detail the *machine learning-powered combinatorial auction (MLCA)* by Brero et al. (2021).

At the core of MLCA is a *query module* (Algorithm 2), which, for each bidder $i \in I \subseteq N$, determines a new value query q_i. First, in the *estimation step* (Line 1), an ML algorithm $\mathcal{A}_i$ is used to learn bidder i's valuation from reports R_i. Next, in the *optimization step* (Line 2), an *ML-based WDP* is solved to find a candidate q of value queries. In principle, any ML algorithm $\mathcal{A}_i$ that allows for solving the corresponding ML-based WDP in a fast way could be used. Finally, if q_i has already been queried before (Line 4), another, more restricted ML-based WDP (Line 6) is solved and q_i is updated correspondingly. This ensures that all final queries q are new.

Algorithm 2: NextQueries(I, R) (Brero et al. 2021)

```
Inputs: Index set of bidders $I$ and reported values $R$
1 foreach $i \in I$ do Fit $\mathcal{A}_i$ on $R_i$: $\mathcal{A}_i[R_i]$                    ▷ Estimation step
2 Solve $q \in \arg\max_{a \in \mathcal{F}} \sum_{i \in I} \mathcal{A}_i[R_i](a_i)$                 ▷ Optimization step
3 foreach $i \in I$ do
4     if $(q_i, v_i(q_i)) \in R_i$ then                                  ▷ Bundle already queried
5         Define $\mathcal{F}' = \{a \in \mathcal{F} : a_i \neq x, \forall (x, v_i(x)) \in R_i\}$
6         Re-solve $q' \in \arg\max_{a \in \mathcal{F}'} \sum_{l \in I} \mathcal{A}_l[R_l](a_l)$
7         Update $q_i = q'_i$
8 return Profile of new queries $q = (q_1, \ldots, q_n)$
```

In Algorithm 3, we present MLCA. In the following, let $R_{-i} = (R_1, \ldots, R_{i-1}, R_{i+1}, \ldots, R_n)$. MLCA proceeds in rounds until a maximum number of queries per bidder Q^{max} is reached. In each round, it calls Algorithm 2 $(Q^{\text{round}} - 1)n + 1$ times: for each bidder $i \in N$, $Q^{\text{round}} - 1$ times excluding a different bidder $j \neq i$ (Lines 5–10, sampled *marginal economies*) and once including all bidders (Line 11, *main economy*). In total each bidder is queried Q^{round} bundles per round in MLCA. At the end of each round, the mechanism receives reports R^{new} from all bidders for the newly generated queries q^{new} and updates the overall elicited reports R (Lines 12–14). In Lines 16–17, MLCA computes an allocation a^*_R that maximizes the *reported* social welfare (see Equation (3)) and determines VCG payments $p(R)$ based on the reported values R (see Appendix Definition 5).

Algorithm 3: MLCA($Q^{\text{init}}, Q^{\text{max}}, Q^{\text{round}}$) (Brero et al. 2021)

```
Params: $Q^{\text{init}}, Q^{\text{max}}, Q^{\text{round}}$ initial, max and #queries/round
foreach $i \in N$ do
    Receive reports $R_i$ for $Q^{\text{init}}$ randomly drawn bundles
for $k = 1, ..., \lfloor (Q^{max} - Q^{init})/Q^{round} \rfloor$ do                      ▷Round iterator
    foreach $i \in N$ do                                                                ▷Marginal economy queries
        Draw uniformly without replacement ($Q^{\text{round}}$−1) bidders from $N \setminus \{i\}$ and store them in $\tilde{N}$
        foreach $j \in \tilde{N}$ do
            $q^{\text{new}} = q^{\text{new}} \cup \mathit{NextQueries}(N \setminus \{j\}, R_{-j})$
    $q^{\text{new}} = \mathit{NextQueries}(N, R)$                                          ▷Main economy queries
    foreach $i \in N$ do
        Receive reports $R_i^{\text{new}}$ for $q_i^{\text{new}}$, set $R_i = R_i \cup R_i^{\text{new}}$
Given elicited reports $R$ compute $a_R^*$ as in Equation (3)
Given elicited reports $R$ compute VCG-payments $p(R)$
return Final allocation $a_R^*$ and payments $p(R)$
```

C ML-POWERED DEMAND QUERY GENERATION

In this section, we reprint the ML-powered demand query generation algorithm from Soumalias et al. (2024b). The critical notions behind the idea are those of indirect utility and revenue and clearing prices.

Definition 10 (Indirect Utility and Revenue). *For linear prices $p \in \mathbb{R}^m_{\geq 0}$, a bidder's indirect utility U and the seller's indirect revenue R are defined as*

$$U(p, v_i) := \max_{x \in \mathcal{X}} \{v_i(x) - \langle p, x \rangle\} \text{ and } \tag{8}$$

$$R(p) := \max_{a \in \mathcal{F}} \left\{ \sum_{i \in N} \langle p, a_i \rangle \right\} \overset{1}{=} \sum_{j \in M} c_j p_j, \tag{9}$$

i.e., at prices p, Equations (8) and (9) are the maximum utility a bidder can achieve for all $x \in \mathcal{X}$ and the maximum revenue the seller can achieve among all feasible allocations.

Definition 11 (Clearing Prices). *Prices $p \in \mathbb{R}^m_{\geq 0}$ are* clearing prices *if there exists an allocation $a(p) \in \mathcal{F}$ such that*

1. *for each bidder i, the bundle $a_i(p)$ maximizes her utility, i.e., $v_i(a_i(p)) - \langle p, a_i(p) \rangle = U(p, v_i), \forall i \in N$, and*
2. *the allocation $a(p) \in \mathcal{F}$ maximizes the sellers revenue, i.e., $\sum_{i \in N} \langle p, a_i(p) \rangle = R(p)$.*[17]

Theorem 2 extends Bikhchandani & Ostroy (2002, Theorem 3.1), establishing a connection between the aforementioned definitions:

Theorem 2 (Soumalias et al. (2024b)). *Consider the notation from Definitions 10 and 11 and the objective function $W(p, v) := R(p) + \sum_{i \in N} U(p, v_i)$. Then it holds that, if a linear clearing price vector exists, every price vector*

$$p' \in \operatorname*{arg\,min}_{\tilde{p} \in \mathbb{R}^m_{\geq 0}} \quad W(\tilde{p}, v) \tag{10a}$$

$$\text{such that} \quad (x_i^*(\tilde{p}))_{i \in N} \in \mathcal{F} \tag{10b}$$

is a clearing price vector and the corresponding allocation $a(p') \in \mathcal{F}$ is efficient.[18]

Theorem 2 does not claim the existence of *linear clearing prices (LCPs)* $p \in \mathbb{R}^m_{\geq 0}$. For general value functions v, LCPs may not exist (Bikhchandani & Ostroy, 2002). However, in the case that LCPs do exist, Theorem 2 shows that *all* minimizers of equation 10 are LCPs and their corresponding allocation is efficient. This is at the core of their ML-powered demand query generation algorithm.

Their key idea to generate ML-powered demand queries is the following: As an approximation for the true value function v_i, they use for each bidder a distinct mMVNN $\mathcal{M}_i^\theta : \mathcal{X} \to \mathbb{R}_{\geq 0}$ that has been trained on the bidder's elicited DQ data R_i. Motivated by Theorem 2, they then try to find the DQ $p \in \mathbb{R}^m_{\geq 0}$ minimizing $W(p, \left(\mathcal{M}_i^\theta\right)_{i=1}^n)$ subject to the feasibility constraint equation 10b. This way, we find demand queries $p \in \mathbb{R}^m_{\geq 0}$ which, given the already observed demand responses R, have high clearing potential.

Note that equation 10 is a hard, bi-level optimization problem. Instead, Theorem 3 allows them to minimize the problem via gradient descent:

Theorem 3 ((Soumalias et al., 2024b)). *Let $\left(\mathcal{M}_i^\theta\right)_{i=1}^n$ be a tuple of trained mMVNNs and let $\hat{x}_i^*(p) \in \arg\max_{x \in \mathcal{X}} \left\{\mathcal{M}_i^\theta(x) - \langle p, x \rangle\right\}$ denote each bidder's predicted utility maximizing bundle w.r.t. $\mathcal{M}_i^\theta$. Then it holds that $p \mapsto W(p, \left(\mathcal{M}_i^\theta\right)_{i=1}^n)$ is* convex, Lipschitz-continuous *and* a.e.

[17] For linear prices, this maximum is achieved by selling every item, i.e., $\forall j \in M : \sum_{i \in N} (a_i)_j = c_j$.

[18] More precisely, constraint equation 10b should be reformulated as

$$\exists \, (x_i^*(\tilde{p}))_{i \in N} \in \bigtimes_{i \in N} \mathcal{X}_i^*(\tilde{p}) : (x_i^*(\tilde{p}))_{i \in N} \in \mathcal{F},$$

where $\mathcal{X}_i^*(\tilde{p}) := \arg\max_{x \in \mathcal{X}} \{v_i(x) - \langle \tilde{p}, x \rangle\}$, since in theory, $x_i^*(\tilde{p})$ does not always have to be unique.

differentiable. *Moreover,*

$$c - \sum_{i \in N} \hat{x}_i^*(p) \in \nabla_p^{\text{sub}} W(p, \left(\mathcal{M}_i^\theta\right)_{i=1}^n) \tag{11}$$

is always a sub-gradient and a.e. a classical gradient.

With Theorem 3, we obtain the following update rule of classical GD $p_j^{\text{new}} \overset{a.e.}{=} p_j - \gamma(c_j - \sum_{i \in N} (\hat{x}_i^*(p))_j), \forall j \in M$. Interestingly, this equation has an intuitive economic interpretation. If the j^{th} item is over/under-demanded based on the predicted utility-maximizing bundles $\hat{x}_i^*(p)$, then its new price p_j^{new} is increased/decreased by the learning rate times its over/under-demand. To enforce constraint equation 10b in GD, they asymmetrically increase the prices $1 + \mu \in \mathbb{R}_{\geq 0}$ times more in case of over-demand than they decrease them in case of under-demand. This leads to the final update rule:

$$p_j^{\text{new}} \overset{a.e.}{=} p_j - \tilde{\gamma}_j(c_j - \sum_{i \in N} (\hat{x}_i^*(p))_j), \forall j \in M, \tag{12a}$$

$$\tilde{\gamma}_j := \begin{cases} \gamma \cdot (1 + \mu) & , c_j < \sum_{i \in N} (\hat{x}_i^*(p))_j \\ \gamma & , \text{else} \end{cases} \tag{12b}$$

D Extended Literature Review

In addition to the related work mentioned in Section 1, we also want to mention some further recent work an ML-based ICAs.

Estermann et al. (2023) use more diverse VQs for the initial VQs. They show that this diversity leads to higher efficiency than just asking initial VQs for i.i.d. uniformly random bundles. However, this does not solve the problem of it being cognitively very hard for bidders to answer these VQs that are not aligned with their preferences. Moreover, their efficiency results are outperformed by our MLHCA.

Maruo & Kashima (2024) uses multi-task learning to transfer to improve the generalization of the MVNNs by leveraging similarities among the value functions across bidders. This technique should also be compatible with our MLHCA. Thus, it would be an interesting direction for future work to incorporate multi-task learning into MLHCA and to evaluate how much this would improve efficiency. From a game theoretical perspective, one should think very carefully if multi-task learning could change the incentives of bidders. From a game-theoretical perspective, one would achieve incentive-alignment with SCW, if each bidder i cannot change the marginal efficiency of the economy $N \setminus \{i\}$. For MLCA, 3 out of 4 VQs actually query these marginal economies, such that $\mathcal{M}_i^\theta$ has no direct influence on these queries, which provides quite a strong game theoretical argument. Via multi-task learning, bidders have a more direct way to influence other bidders' models. While multi-task learning is a very promising direction to explore, one should be aware of potential game-theoretical risks imposed by multi-task learning.

Lubin et al. (2021) allow bidders to answer VQs with an interval over prices instead of an exact price. It would be interesting for future work to combine this approach with MLHCA.

Weissteiner (2023) and Heiss (2024, Section 4.4) provide a broader picture on ML-based ICAs.

E Proofs from Section 3

In this Section, we provide all deferred proofs from Section 3.

Lemma 1 Proof. Let $n = 2$ and $m = 100$ and $c_1 = c_2 = \cdots = c_m = 1$, i.e., the auction has 100 unique items. Bidder 1 has a value of zero for the empty set and a value of $\epsilon > 0$ for any non-empty set of items, while bidder 2 has a value of $V \to \infty$ for the full bundle, and a value of zero for any other bundle. Note that these are proper value functions, as they are both monotone and assign a value of zero to the empty set. The bundle space $\mathcal{X}$ has a size of 2^m. For the auction that asks random value queries, the probability that bidder 2 is queried her value for the full bundle conditioned on not having been asked that question in the previous k queries is $\frac{1}{2^m-k}$. Given that 2^m is many orders of magnitude larger than k, the probability of the auction not querying bidder 2 her value for the full bundle in k random value queries is:

$$\prod_{j=1}^{k}\left(1-\frac{1}{2^m-(j-1))}\right)=\prod_{j=1}^{k}\left(\frac{2^m-j}{2^m-(j-1))}\right)\approx\prod_{j=1}^{k}\left(\frac{2^m}{2^m}\right)=1 \tag{13}$$

If that query is asked to bidder 2, then bidder 2 will be allocated that bundle and bidder 1 will be allocated the empty bundle, and the social welfare of the final allocation will be equal to V. In any other case, bidder 1 will be allocated a non-empty bundle, and the social welfare of the allocation will be equal to ϵ.

Say we restrict the auction that asks random queries to just a single demand query per bidder. The expected total price for the full bundle is $m \cdot p$, where p is the expected value of the price of a single item. Given that $V \gg p \cdot m$, the probability that bidder 2 will not request the full bundle, even with just a single query, tends to 0. But if bidder 2 requests the full bundle, her inferred value for it will be on expectation $p \cdot m$, while bidder 1's inferred value for the bundle that she requested in that round will be at most her true value for that bundle, ϵ. Given that ϵ tends to zero, the probability that $p \cdot m$ is less than ϵ tends to zero. Thus, the expected value of the auction that asks a single random DQ tends to V. Taking $V \to \infty$ and $\epsilon \to 0$ completes the proof. □

F Training Algorithm Details

In this section, we provide the details on our training algorithm to combine DQs and VQs. To leverage the advantages of both DQs and VQs, we propose a straightforward two-stage training algorithm. In each epoch, the ML model is first trained on all DQ responses using the loss function from (Soumalias et al., 2024b) (Lines 4 to 6). The main idea behind this loss, is that for each DQ, an optimizaiton problem is solved to predict the bidder's utility-maximizing bundle at the given prices, treating her ML model as her true value function. In case the predicted reply disagrees with the bidder's true reply, the loss is the difference in predicted utilities between these 2 bundles, given the current prices. Next, the model is trained on the VQ responses using a standard regression loss (Lines 8 to 10) This mixed approach ensures that the model benefits from both the broad information of DQs and the precise value information from VQs.

Algorithm 4: MixedTraining

```
Input  : Demand query data $R_i^{DQ} = \{(x_i^*(p^r), p^r)\}_{r=1}^R$, Value query data $R_i^{VQ} = \{(x_i^l, v_i(x_i^l))\}_{l=1}^L$
         Epochs $T \in \mathbb{N}$, Learning Rate $\gamma > 0$, Cardinal loss function $F$
1  $\theta_0 \leftarrow$ init mMVNN                                  ▷ Weissteiner et al. (2023, S.3.2)
2  for $t = 0$ to $T - 1$ do
3      for $r = 1$ to $R$ do                                         ▷ Demand responses for prices
4          Solve $\hat{x}_i^*(p^r) \in \arg\max_{x \in \mathcal{X}} \mathcal{M}_i^{\theta_t}(x) - \langle p^r, x \rangle$
5          if $\hat{x}_i^*(p^r) \neq x_i^*(p^r)$ then                ▷ mMVNN is wrong
6              $L(\theta_t) \leftarrow (\mathcal{M}_i^{\theta_t}(\hat{x}_i^*(p^r)) - \langle p^r, \hat{x}_i^*(p^r) \rangle) - (\mathcal{M}_i^{\theta_t}(x_i^*(p^r)) - \langle p^r, x_i^*(p^r) \rangle)$   ▷ Add
                predicted utility difference to loss
7              $\theta_{t+1} \leftarrow \theta_t - \gamma(\nabla_\theta L(\theta))_{\theta=\theta_t}$   ▷ SGD step
8      for $l = 1$ to $L$ do                                         ▷ Value Queries
9          $L(\theta_t) \leftarrow F(\mathcal{M}_i^{\theta_t}(x_i^l), v_i(x_i^l))$   ▷ Cardinal Loss on VQs
10         $\theta_{t+1} \leftarrow \theta_t - \gamma(\nabla_\theta L(\theta))_{\theta=\theta_t}$   ▷ SGD step
11 return Trained mMVNN $\mathcal{M}_i^{\theta_T}$
```

G Learning Experiments for Other Domains

In Tables 3 to 5 we present the results of the learning experiment of Section 4.2 for all additional domains.

Across all domains, the network trained only on DQs demonstrates the worst generalization performance on the dataset $\mathcal{T}_r$. This is primarily due to two factors: the absence of absolute value information that VQs provide and the distributional shift between $\mathcal{T}_r$ and $\mathcal{T}_p$, with the DQ training data being more aligned with $\mathcal{T}_p$.

The performance of the network trained solely on VQs varies by domain. In the GSVM and SRVM domains, the learning task is relatively easy, as indicated by the already strong performance of previous ML-powered ICAs. In these domains, the networks trained only on VQs perform well across both test sets (Tables 3 and 5). However, in the more challenging LSVM domain—similarly to the MRVM domain discussed in Section 4.2—the network trained exclusively on VQs performs well on the $\mathcal{T}_r$ test set, which contains points from the same distribution as its training data, but performs worse on the utility-maximizing bundles of $\mathcal{T}_p$ compared to the network trained on both query types.

This inferior learning performance on the critical dataset $\mathcal{T}_p$ explains why MLHCA outperforms pure VQ-based ML-powered ICAs, such as Weissteiner et al. (2023); Weissteiner & Seuken (2020), in the LSVM domain.

Optimization	**Train Points**		$\mathbf{R^2}$		**KT**		**MAE scaled**		$\mathbf{R_c^2}$	
Metric	VQs	DQs	$\mathcal{T}_r$	$\mathcal{T}_p$	$\mathcal{T}_r$	$\mathcal{T}_p$	$\mathcal{T}_r$	$\mathcal{T}_p$	$\mathcal{T}_r$	$\mathcal{T}_p$
R^2 on $\mathcal{V}_r$	20	40	0.96	0.95	0.90	0.94	0.07	0.12	0.96	0.98
	60	0	0.99	0.98	0.96	0.98	0.03	0.05	0.99	0.98
	0	60	0.79	0.97	0.83	0.94	0.04	0.02	0.91	0.98
R^2 on $\mathcal{V}_p$	20	40	0.96	0.99	0.91	0.96	0.07	0.04	0.97	0.99
	60	0	0.99	0.98	0.96	0.98	0.03	0.02	0.99	0.98
	0	60	0.79	0.97	0.83	0.94	0.13	0.05	0.91	0.98

Table 3: Learning comparison of training only on DQs, only on VQs, or on both for the GSVM domain.

Optimization	**Train Points**		$\mathbf{R^2}$		**KT**		**MAE scaled**		$\mathbf{R_c^2}$	
Metric	VQs	DQs	$\mathcal{T}_r$	$\mathcal{T}_p$	$\mathcal{T}_r$	$\mathcal{T}_p$	$\mathcal{T}_r$	$\mathcal{T}_p$	$\mathcal{T}_r$	$\mathcal{T}_p$
R^2 on $\mathcal{V}_r$	20	40	0.38	0.88	0.65	0.80	0.46	0.33	0.44	0.91
	60	0	0.67	0.80	0.75	0.81	0.30	0.46	0.67	0.87
	0	60	-1.20	0.99	0.80	0.84	1.10	0.11	0.46	0.99
R^2 on $\mathcal{V}_r$	20	40	0.38	0.88	0.65	0.80	0.46	0.33	0.44	0.91
	60	0	0.65	0.82	0.81	0.88	0.25	0.38	0.66	0.87
	0	60	-2.97	0.96	0.77	0.85	1.51	0.22	0.42	0.97

Table 4: Learning comparison of training only on DQs, only on VQs, or on both for the LSVM domain.

Optimization	Train Points		$\mathbf{R^2}$		KT		MAE scaled		$\mathbf{R_c^2}$	
Metric	VQs	DQs	$\mathcal{T}_r$	$\mathcal{T}_p$	$\mathcal{T}_r$	$\mathcal{T}_p$	$\mathcal{T}_r$	$\mathcal{T}_p$	$\mathcal{T}_r$	$\mathcal{T}_p$
R^2 on $\mathcal{V}_r$	20	40	1.00	0.89	0.97	0.90	0.02	0.03	1.00	0.93
	60	0	1.00	0.96	0.99	0.97	0.00	0.01	1.00	0.97
	0	60	0.93	-0.13	0.96	0.92	0.11	0.10	0.97	0.94
R^2 on $\mathcal{V}_p$	20	40	1.00	0.94	0.98	0.92	0.01	0.02	1.00	0.94
	60	0	1.00	0.96	0.99	0.97	0.00	0.01	1.00	0.97
	0	60	0.91	0.02	0.96	0.86	0.12	0.09	0.95	0.89

Table 5: Learning comparison of training only on DQs, only on VQs, or on both for the SRVM domain.

H Detailed Auction Mechanism

In this section, we present a detailed description of MLHCA. The full auction mechanism is presented in Algorithm 5. In Lines 3 to 6, we generate the first Q^{CCA} DQs using the same price update rule as the CCA. In each of the next Q^{DQ} ML-powered rounds, we first train, for each bidder, an mMVNN on her demand responses (Line 9). Next, in Line 10, we call NextPrice (Soumalias et al., 2024b) to generate the next DQ based on the agents' trained mMVNNs (see Appendix C). If MLHCA has found market-clearing prices, then the corresponding allocation is efficient and is returned, along with payments $\pi(R)$ according to the deployed payment rule (Line 16). If, by the end of the ML-powered DQs the market has not cleared we switch to VQ rounds. In the first VQ round (Line 18) we ask each bidder for her *bridge bid* (see Definition 4). As proven in Lemma 4, this single VQ ensures that the MLHCA's efficiency is lower bounded by the efficiency after just the DQ rounds. The difference in the algorithm description compared to the version presented in Section 5 lies in the VQ rounds. Specifically, we make use of *marginal economies*. Once every Q^{round} VQ rounds, for each bidder, we query her value for the bundle she receives in the predicted optimal allocation (based on all ML models), under the constraint that the bidder in question receives a bundle for which she has not been queried in the past (Lines 22 to 25). This is as described in Section 5. But in the other $Q^{\text{round}} - 1$ rounds, for each bidder, we query her value for the bundle she receives in the predicted optimal allocation based only on the models of the *non-marginalized bidders* (Lines 26 to 30). Each time, for each bidder, we marginalize $Q^{\text{round}} - 1$ bidders uniformly at random without replacement. The marginal economies have been designed to improve the incentive properties of the auction (for a detailed analysis, see Brero et al. (2021)). Similar to all papers in this line of work, e.g. Brero et al. (2021); Weissteiner et al. (2022a; 2023), we set $Q^{\text{round}} = 4$ in all of our experiments. The final allocation and payments are then determined based on all reports (Lines 25 to 26). Note that ML-CCA can be combined with various possible payment rules $\pi(R)$, such as VCG or VCG-nearest.

Algorithm 5: MLHCA($Q^{\text{CCA}}, Q^{\text{DQ}}, Q^{\text{VQ}}, Q^{\text{round}}$)

```
Parameters: $Q^{\text{CCA}}, Q^{\text{DQ}}, Q^{\text{VQ}}, Q^{\text{round}}$ and $\pi$
$R^{VQ} \leftarrow (\{\})_{i=1}^{N}$
$R^{DQ} \leftarrow (\{\})_{i=1}^{N}$
for $r = 1, ..., Q^{CCA}$ do                                          ▷ Draw $Q^{\text{CCA}}$ initial prices
    $p^r \leftarrow CCA(R^{DQ})$
    foreach $i \in N$ do                                              ▷ Initial demand query responses
        $R_i^{DQ} \leftarrow R_i^{DQ} \cup \{(x_i^*(p^r), p^r)\}$
for $r = Q^{CCA} + 1, ..., Q^{CCA} + Q^{DQ}$ do                       ▷ ML-powered DQs
    foreach $i \in N$ do
        $\mathcal{M}_i^\theta \leftarrow$ MixedTraining($R_i^{DQ} \cup R_i^{VQ}$)      ▷ Section 4.1
    $p^r \leftarrow$ NextPrice($(\mathcal{M}_i^\theta)_{i=1}^n$)                   ▷ Appendix C
    foreach $i \in N$ do                                              ▷ Demand query responses for $p^r$
        $R_i^{DQ} \leftarrow R_i^{DQ} \cup \{(x_i^*(p^r), p^r)\}$
    if $\sum_{i=1}^{n} (x_i^*(p^k))_j = c_j \; \forall j \in M$ then   ▷ Market-clearing
        $a^*(R^{DQ} \cup R^{VQ}) \leftarrow (x_i^*(p^r))_{i=1}^n$       ▷ Set final allocation to clearing allocation
        Calculate payments $\pi(R^{DQ} \cup R^{VQ}) \leftarrow (\pi_i(R^{DQ} \cup R^{VQ}))_{i=1}^n$
        return $a^*(R^{DQ} \cup R^{VQ})$ and $\pi(R^{DQ} \cup R^{VQ})$
foreach $i \in N$ do                                                  ▷ Bridge bid
    $R_i \leftarrow R_i \cup \{(a_i^*(R), v_i(a_i^*(R)))\}$
for $r = Q^{CCA} + Q^{DQ} + 2, ..., Q^{CCA} + Q^{DQ} + Q^{VQ}$ do     ▷ ML-powered VQs
    foreach $i \in N$ do
        $\mathcal{M}_i^\theta \leftarrow$ MixedTraining($R_i^{DQ} \cup R_i^{VQ}$)      ▷ Section 4.1
    if $r \% Q^{round} = 0$ then                                      ▷ Query Main Economy
        foreach $i \in N$ do
            $x'(R) \leftarrow \arg\max_{x \in \mathcal{F}: x_i \notin R_i^{VQ}} \sum_{i' \in N} \mathcal{M}_i^\theta(x_{i'})$   ▷ Find predicted optimal allocation
            $x_i^*(R) \leftarrow x_i'(R)$
    else                                                              ▷ Query Marginal Economy
        foreach $i \in N$ do
            $\widetilde{N} \leftarrow$ draw uniformly at random $Q^{\text{round}} - 1$ bidders from $N \setminus \{i\}$
            $x'(R) \leftarrow \arg\max_{x \in \mathcal{F}: x_{i'} \notin R_{i'}^{VQ}} \sum_{i' \in \widetilde{N}} \mathcal{M}_i^\theta(x_{i'})$   ▷ Find predicted optimal allocation
              in marginal economy
            $x_i^*(R) \leftarrow x_i'(R)$
    foreach $i \in N$ do                                              ▷ Value query responses for $x^*(R)$
        $R_i \leftarrow R_i \cup \{(x_i^*(R), v_i(x_i^*(R)))\}$
Calculate final allocation $a^*(R)$ as in Equation (3)
Calculate payments $\pi(R)$                                           ▷ E.g., VCG (Appendix A)
return $a^*(R)$ and $\pi(R)$
```

I MVNN

The original definition (Weissteiner et al., 2022a) is a special case of the more general definition (Soumalias et al., 2024b) that we state here.

Definition 12 (MVNN). *An MVNN $\mathcal{M}_i^\theta : \mathcal{X} \to \mathbb{R}_{\geq 0}$ for bidder $i \in N$ is defined as*

$$\mathcal{M}_i^\theta(x) := W^{i,K_i} \varphi_{0,t^{i,K_i-1}} \left(\ldots \varphi_{0,t^{i,1}}(W^{i,1}\,(Dx) + b^{i,1}) \ldots \right) \tag{14}$$

- *$K_i + 2 \in \mathbb{N}$ is the number of layers (K_i hidden layers),*
- *$\{\varphi_{0,t^{i,k}}\}_{k=1}^{K_i-1}$ are the MVNN-specific activation functions with cutoff $t^{i,k} > 0$, called* bounded ReLU (bReLU)*:*

$$\varphi_{0,t^{i,k}}(\cdot) := \min(t^{i,k}, \max(0, \cdot)) \tag{15}$$

- *$W^i := (W^{i,k})_{k=1}^{K_i}$ with $W^{i,k} \geq 0$ and $b^i := (b^{i,k})_{k=1}^{K_i-1}$ with $b^{i,k} \leq 0$ are the* non-negative *weights and* non-positive *biases of dimensions $d^{i,k} \times d^{i,k-1}$ and $d^{i,k}$, whose parameters are stored in $\theta = (W^i, b^i)$.*
- *$D := \operatorname{diag}\left(1/c_1, \ldots, 1/c_m\right)$ is the linear normalization layer that ensures $Dx \in [0,1]$ and is not trainable.*

Remark 7. *The index i of the MVNN $\mathcal{M}_i^\theta(x)$ emphasizes that we train an individual MVNN for every bidder i to approximate v_i. In the following, we sometimes omit the index i if we just want to make general arguments about the MVNN architecture without.*

Remark 8 (Linear Skip Connection). *Sometimes we also use linear skip connections as introduced in (Weissteiner et al., 2023, Definition F.1)*

Remark 9 (Initiaization). *We always use the initialization scheme from (Weissteiner et al., 2023, Section 3.2 and Appednix E), which offers crucial advantages over standard initialization schemes as discussed in (Weissteiner et al., 2023, Section 3.2 and Appednix E).*

I.1 On the inductive bias of MVNNs

Weissteiner et al. (2022a); Soumalias et al. (2024b) have shown that MVNNscan represent any monotonic normalized function on $\mathcal{X}$. However, for finitely many data points multiple different monotonic functions can fit the data equally well, but the training algorithm will choose only one of these functions. We want to understand according to which preferences the algorithm makes this choice, i.e., we want to understand its inductive bias.

For certain ReLU-NNs it has been shown that L2-regularization (also known as "weight decay") of the parameters θ corresponds to regularizing a Lp-norm of the second derivative of the function (Heiss et al., 2019; 2023; 2021; Heiss, 2024; Savarese et al., 2019; Ongie et al., 2019; Williams et al., 2019; Parhi & Nowak, 2022). Since the second derivative of linear functions is zero, these NNs prefer linear functions.

However, MVNNs use a different activation function (Weissteiner et al., 2022a). For MVNNs, no theoretical result about their second derivative has been proven so far. It is quite clear that the L2-regularization of the parameters of a MVNN does not exactly correspond to any Lp-norm of the second derivative. Weissteiner et al. (2023) modified the MVNN architecture by adding so-called linear skip connections (Weissteiner et al., 2023, Definition F.1) to obtain an inductive bias towards linear functions. If one uses unregularized linear skip connections but regularizes all other parameters, it is quite obvious that the optimal parameters will only have non-zero weights in the linear skip connections if a monotonic linear function can perfectly explain the data.[19]

In the setting of Example 1 (which is based on the example in the proof of Theorem 1) one can also prove that MVNNs with arbitrarily small L2-regularization, would always choose a function that is linear on $\mathcal{X}$ given any possible truthful DQ responses from bidder 2, even without linear skip connections.

[19] If the data can be perfectly explained by a linear function, then only using the linear skip connections can achieve zero training loss and zero regularization costs, while setting any parameter outside the linear skip connections to any non-zero value would lead to non-zero L2 regularization costs.

Proposition 2. *As in Example 1, let $n = 2$, $m = 1$, $c_1 = 10$ and v_2 such that whenever bidder 2 is queried a DQ she answers in the following way:*

- *If the price p is below $\frac{94}{10}$, bidder 2 will answer with $x_2^*(p) = (10)$;*
- *if the price $p = \frac{94}{10}$, bidder 2 will answer with either $x_2^*(p) = (10)$ or $x_2^*(p) = (0)$;*
- *if the price p is higher than $\frac{94}{10}$, bidder 2 will answer with $x_2^*(p) = (0)$.*

Let $\{p^1, \ldots, p^{n_{\mathrm{DQ}}^{\mathrm{train}}}\} \subset [0, \infty)$ be the subset of prices bidder 2 is queried. Let θ^* be any (local) minimizer of the L2-regularized loss from Soumalias et al. (2024b)

$$L^\lambda(\theta) := \sum_{r=1}^{n_{\mathrm{DQ}}^{\mathrm{train}}} \Big(\mathcal{M}_\theta(\hat{x}_2^*(p^r)) - \langle p^r, \hat{x}_2^*(p^r) \rangle - \big(\mathcal{M}_\theta(x_2^*(p^r)) - \langle p^r, x_2^*(p^r) \rangle \big) \Big)^+ + \lambda \left\|\theta\right\|_2^2,$$

where $\hat{x}_2^*(p^r) := \arg\min_{x \in \mathcal{X}} \left(\mathcal{M}_\theta(x) - \langle p^r, x \rangle \right)$. Then the MVNN $\mathcal{M}_{\theta^*} : \mathcal{X} \to \mathbb{R}$ is linear.

Proof. We define $\tilde{p} := \max \left\{ p^r : x^*(r) = (10), 1 \leq r \leq n_{\mathrm{DQ}}^{\mathrm{train}} \right\}$.[20]

1. First we show that $\mathcal{M}_{\theta^*}(10) \leq 10\tilde{p}$ via a contraposition argument. Let's assume $\mathcal{M}_{\theta^*}(10) \geq 10q > 10\tilde{p}$, then multiplying the last layer's weights by $1 - \delta > \frac{q}{\tilde{p}}$ would both reduce the data-loss-term L^0 (since the activation the hidden layers of MVNNs are always non-negative) and the regularization costs $\lambda \left\|\cdot\right\|_2^2$. Therefore, no local minima θ^* can satisfy $\mathcal{M}_{\theta^*}(10) > 10\tilde{p}$. Thus, we have shown that $\mathcal{M}_{\theta^*}(10) \leq 10\tilde{p}$ holds for any local minima θ^*.
2. Next, we show that all pre-activations of our $\mathcal{M}_{\theta^*}$ are smaller or equal to the cut-off of the corresponding bReLU activation function for any input $x \in \mathcal{X}$. Let's assume again the contraposition that at least one pre-activation is larger than the cut-off. In this case, we can scale down all the incoming weights of such a neuron without changing $\mathcal{M}_{\theta^*}(10)$. Scaling down these weights cannot increase the value of $\mathcal{M}_{\theta^*}(x)$ for any $x \in \mathcal{X}$, so it cannot increase the data-loss term L^0, but scaling down weights obviously decreased the regularization costs. Thus, via this counterposition argument, we have proven that all the pre-activations are smaller or equal to the cut-off for any local minima θ^*.
3. Next, we show that all biases of θ^* are zero. First, note that by item 2, we know that $\mathcal{M}_{\theta^*}$ is convex (since the bReLU is convex below the cut-off). By combining this fact with item 1, we obtain that $\mathcal{M}_{\theta^*}(x) \leq x\tilde{p}$, since MVNNs always satisfy $\mathcal{M}_\theta(0) = 0$. Let's assume the counterposition of at least one bias being strictly negative (as by definition biases can never be positive for MVNNs). Then we could increase the bias a little bit without increasing the data-loss-term L^0,[21] but increasing the bias reduces its regularization cost. Thus any local minima θ^* satisfies that the biases are zero.

By combining items 2 and 3 we obtain that $\mathcal{M}_{\theta^*}$ is linear. □

[20] If $\left\{ p^r : x^*(r) = (10), 1 \leq r \leq n_{\mathrm{DQ}}^{\mathrm{train}} \right\}$ is empty, we define $\tilde{p} := 0$. In Example 1, $\tilde{p} = \frac{94-\epsilon}{10}$.

[21] This argument relies on the fact, that we only queried finitely many DQs. If we asked infinitely many DQs that are dense around $p = \frac{94}{10}$, one would need to modify the argument by not only increasing the biases but simultaneously also decreasing certain weights.

J Revenue Results

	Efficiency Loss in %		**Relative Revenue in %**	
Domain	**MLHCA**	**BOCA**	**MLHCA**	**BOCA**
GSVM	0.00 ± 0.00	—	70.15 ± 4.43	—
LSVM	0.04 ± 0.07	0.39 ± 0.31	79.43 ± 3.05	73.53 ± 3.72
SRVM	0.00 ± 0.00	0.06 ± 0.02	56.05 ± 1.69	54.22 ± 1.46
MRVM	4.81 ± 0.57	7.77 ± 0.35	27.97 ± 2.16	42.04 ± 1.89

Table 6: Efficiency loss and relative revenue comparison between MLHCA (40DQs + 60VQs) and BOCA (100VQs). Shown are averages and a 95% CI. Winners marked in gray.

In Table 6, we present the relative revenue results of MLHCA and BOCA, both using VCG payments (see Appendix A.1). We define relative revenue as the percentage of optimal welfare recovered as revenue on a per-instance basis. For a detailed discussion of the corresponding efficiency results, please refer to Section 6.

Unlike efficiency, the best-performing mechanism in terms of revenue varies by domain. In the LSVM and SRVM domains, MLHCA generates higher revenue than BOCA, while in the MRVM domain, the opposite is true.

The explanation for MLHCA's higher revenue in the LSVM and SRVM domains is straightforward: MLHCA achieves higher efficiency than BOCA in these domains and still has at least 42 VQs remaining after matching BOCA's efficiency. This additional exploration afforded by those extra VQs enables MLHCA to identify many high-value allocations, ultimately driving up prices under the VCG payment rule.

The lower revenue of MLHCA compared to BOCA in the MRVM domain can be explained by examining the DQ rounds of MLHCA. In this domain, lower competition among bidders results in relatively low item prices, which reduces the inferred-to-true welfare ratio of the allocations based only on the DQs. This is illustrated in Figure 3. The low inferred value from these queries prevents them from driving up the VCG prices, even though they lead to allocations with high efficiency. As a result, in this domain, only the VQs contribute significantly to the auction's payments. Given that BOCA uses 100 VQs while MLHCA uses only 60, this difference leads to BOCA achieving higher revenue in the MRVM domain.

K Details on why one should start with DQs and end with VQs.

This appendix extends Section 3.3 by elaborating more on multiple reasons why ML-based ICAs benefit enormously from first asking DQs and then transitioning to VQs.

From a practical standpoint, it is nearly impossible for bidders to accurately answer VQs for bundles selected uniformly at random, while it is much easier for them to respond to VQs that are closely aligned with their preferences.[22] At the beginning of the auction, the auctioneer has no prior knowledge of which bundles align with the bidders' interests. By starting with DQs, the auctioneer can gather initial insights into bidders' preferences, enabling the selection of VQs that are later highly targeted and more relevant to each bidder's interests.

This practical consideration alone makes a compelling case for starting an auction with DQs. In the following, we argue that even beyond this practical reason, there are multiple theoretical advantages to beginning with DQs and concluding the auction with VQs.

Empirically, we observe that VQ-based approaches (Weissteiner & Seuken, 2020; Weissteiner et al., 2022b;a; 2023) tend to outperform DQ-based mechanisms in later iterations, as suggested by Theorem 1 and Lemma 5. However, DQs consistently perform better in the earlier rounds.

This leads to our hypothesis that DQs are significantly more valuable in the initial iterations of CAs, whereas VQs become more valuable in later stages. Our experiments, presented in Section 6, confirm this hypothesis. Our MLHCA, which begins by asking DQs and later transitions to VQs, achieves higher efficiency compared to using only DQs or only VQs for all rounds.

The intuitive explanation for this is as follows: from the outset, bidders respond to DQs with bundles that they find genuinely interesting, providing relevant information right from the start (see Lemma 1). It also makes sense to first gather global information[23] about the value functions through DQs to get an overview of bidders' preferences, and then use VQs to gather more precise, "local" information about the most relevant bundles once we have enough knowledge to identify them.

As the auction progresses, the limitations of DQs (see Theorem 1) become more apparent. In contrast, VQs do not suffer from these limitations and remain informative until an efficient allocation is found (see Lemma 5).

Example 1 illustrates how VQs lead to allocations that fit together effectively.[24] In contrast, responses to a single DQ often lead to over-demand for certain items or leave some items unassigned (under-demand). In Example 1, bidder 2 lacks information to know she should bid for 9 items. Only the auctioneer, having information from all bidders, knows that assigning 9 items to bidder 2 would complement bidder 1's preferences. The auctioneer can leverage this aggregated knowledge by asking bidder 2 a VQ for 9 items, whereas DQs alone would not provide this opportunity.

Moreover, we argue that the combination of DQs and VQs is particularly powerful for learning bidders' value functions, as the information from these two query types complements each other nicely (see Section 4).

[22] In the example from Footnotes 3 and 4, imagine being asked in one iteration to provide your value for a bundle of exactly the ingredients needed for a strawberry cake, and in the next iteration for a bundle needed for a blueberry cake. These VQs might be easier to answer if your initial intention was baking a cake.

[23] Each DQ provides information about every possible bundle in the bundle space, making DQs highly effective for exploring the entire space early on.

[24] By definition, all the bundles in a VQ form a feasible allocation. Furthermore, VQs typically allocate (almost) all items to bidders, as they maximize the estimated social welfare. The MVNN architecture ensures monotonicity in the estimated value functions. If the estimated value functions were strictly monotonic, the solution to the MILPs determining the next VQ would always allocate all items.

K.1 Hyperparameter Optimization Details

In this section, we provide details on our exact HPO methodology and the ranges that we used.

We separately optimized the HPs of the mMVNNs for each bidder type of each domain, using a different set of SATS seeds than for all other experiments in the paper. Specifically, for each bidder type, we first trained an mMVNN using as initial data points the demand responses of an agent of that type during 40 consecutive CCA clock rounds, and her value responses for 30 uniformly at random selected bundles, and then measured the generalization performance of the resulting network on a validation set that consisted of 50,000 uniformly at random bundles of items, similar to $\mathcal{V}_r$ in Section 4.2. The number of seeds used to evaluate each model was equal for all models and set to 10. Finally, for each bidder type, we selected the set of HPs that performed the best on this validation set with respect to the coefficient of determination (R^2). The full range of HPs tested for all agent types and all domains is shown in Table 7, while the winning configurations are shown in Table 8.

The winning configurations for both metrics are shown in Table 8

Hyperparameter	**HPO-Range**
Non-linear Hidden Layers	[1,2,3]
Neurons per Hidden Layer	[8, 10, 20, 30]
Learning Rate	(1e-4, 1e-2)
Epochs	[30, 50, 70, 100, 300, 500, 1000]
L2-Regularization	(1e-8, 1e-2)
Linear Skip Connections[25]	[True, False]
Cached DQ solution Frequency	[1, 2, 5, 10]
Batch Size for VQs	[1, 5, 10]

Table 7: HPO ranges for all domains.

Domain	Bidder Type	# Hidden Layers	# Hidden Units	Lin. Skip	Learning Rate	L2 Regularization	Epochs	Cached Solution Freq.	Batch Size
LSVM	Regional	1	20	False	0.001	0.000001	100	20	1
	National	1	30	True	0.0001	0.001	1000	10	5
GSVM	Regional	2	30	True	0.0001	0.001	1000	10	5
	National	2	30	False	0.0005	0.0001	200	10	1
MRVM	Local	3	20	True	0.001	0.00001	200	5	10
	Regional	1	30	True	0.0001	0.000001	1000	20	1
	National	2	20	False	0.001	0.000001	100	5	10
SRVM	Local	1	1000	True	0.01	0.0001	10	5	1
	Regional	1	30	False	0.005	0.000001	500	5	1
	High Frequency	1	10	True	0.005	0.00001	500	10	5
	National	1	10	True	0.005	0.00001	1000	5	10

Table 8: Winning HPO configurations for R^2

[25] For the definition of (m)MVNNs with a linear skip connection, please see Weissteiner et al. (2023, Definition F.1)