A Permutation-Equivariant Neural Network Architecture For Auction Design

Jad Rahme *[a], Samy Jelassi[a,b], Joan Bruna[b, c, d], and S. Matthew Weinberg[a]

[a]Princeton University, Princeton
[b]Courant Institute of Mathematical Sciences, New York University, New York
[c]Center for Data Science, New York University, New York
[d]Institute for Advanced Study, Princeton

June 11, 2020

Abstract

Designing an incentive compatible auction that maximizes expected revenue is a central problem in Auction Design. Theoretical approaches to the problem have hit some limits in the past decades and analytical solutions are known for only a few simple settings. Computational approaches to the problem through the use of LPs have their own set of limitations. Building on the success of deep learning, a new approach was recently proposed by Duetting et al. (2019) in which the auction is modeled by a feed-forward neural network and the design problem is framed as a learning problem. The neural architectures used in that work are general purpose and do not take advantage of any of the symmetries the problem could present, such as permutation equivariance. In this work, we consider auction design problems that have permutation-equivariant symmetry and construct a neural architecture that is capable of perfectly recovering the permutation-equivariant optimal mechanism, which we show is not possible with the previous architecture. We demonstrate that permutation-equivariant architectures are not only capable of recovering previous results, they also have better generalization properties.

1 Introduction

Designing truthful auctions is one of the core problems that arise in economics. Concrete examples of auctions include sales of treasury bills by the US government, art sales by Christie's or Google Ads. Following seminal work of Vickrey (Vickrey, 1961) and Myerson (Myerson, 1981), auctions are typically studied in the *independent private valuations* model: each bidder has a valuation function over items, and their payoff depends only on the items they receive. Moreover, the auctioneer knows aggregate information about the population that each bidder comes from, modeled as a distribution over valuation functions, but does not know precisely each bidder's valuation. Auction design is challenging since the valuations are private and bidders need to be encouraged to report their valuations truthfully. The auctioneer aims at designing an incentive compatible auction that maximizes revenue.

*Corresponding author: jrahme@princeton.edu

While auction design has existed as a subfield of economic theory for several decades, complete characterizations of the optimal auction only exist for a few settings. Myerson resolved the optimal auction design problem when there is a single item for sale (Myerson, 1981). However, the problem is not completely understood even in the extremely simple setting with just a single bidder and two items. While there have been some partial characterizations (Manelli and Vincent, 2006, 2010; Pavlov, 2011; Wang and Tang, 2014; Daskalakis et al., 2017), and algorithmic solutions with provable guarantees (Alaei, 2011; Alaei et al., 2012, 2013; Cai et al., 2012a,b), neither the analytic nor algorithmic approach currently appears tractable for seemingly small instances.

Another line of work to confront this theoretical hurdle consists in building automated methods to find the optimal auction. Early works (Conitzer and Sandholm, 2002, 2004) framed the problem as a linear program. However, this approach suffers from severe scalablility issues as the number of constraints and variables is exponential in the number of bidders and items (Guo and Conitzer, 2010). Later, Sandholm and Likhodedov (2015) designed algorithms to find the optimal auction. While scalable, they are however limited to specific classes of auctions known to be incentive compatible. A more recent research direction consists in building deep learning architectures that design auctions from samples of bidder valuations. Duetting et al. (2019) proposed RegretNet, a feed-forward architecture to find near-optimal results in several known multi-item settings and obtain new mechanisms in unknown cases. This architecture however is not data efficient and can require a large number of valuation samples to learn an optimal auction in some cases. This inefficiency is not specific to RegretNet but is characteristic of neural network architectures that do not incorporate any inductive bias.

In this paper, we build a deep learning architecture for *multi-bidder symmetric auctions*. These are auctions which are invariant to relabeling the items or bidders. More specifically, such auctions are *anonymous* (in that they can be executed without any information about the bidders, or labeling them) and *item-symmetric* (in that it only matters what bids are made for an item, and not its a priori label).

It is now well-known that when bidders come from the same population that the optimal auction itself is anonymous. Similarly, if items are *a priori indistinguishable* (e.g. different colors of the same car — individuals certainly value a red vs. blue car differently, but there is nothing objectively more/less valuable about a red vs. blue car), the optimal auction is itself item-symmetric. So in such settings, our approach will still approach the true optimum (but with better generalization, and in a way which retains this structure, see Contributions below). Even without these conditions, the optimal auction is often symmetric anyway: for example, "bundling together" (the auction which allows bidders to pay a fixed price for all items, or receive nothing) is item-symmetric, and is often optimal even when the items are a priori distinguishable.

Beyond their frequent optimality, such auctions are desirable objects of study *even when they are suboptimal*. For example, seminal work of Hartline and Roughgarden which pioneered the study of "simple vs. optimal auctions" analyzes the approximation guarantees achievable by anonymous auctions (Hartline and Roughgarden, 2009), and exciting recent work continues to improve these guarantees (Alaei et al., 2015; Jin et al., 2019a,c,b). Similarly, Daskalakis and Weinberg develop algorithms for item-symmetric instances (Daskalakis and Weinberg, 2012), and exciting recent work show how to leverage item-symmetric to achieve near-optimal auctions in completely general settings (Kothari et al., 2019). To summarize: symmetric auctions are known to be optimal in many settings of interest (even those which are not themselves symmetric). Even in settings where they are not optimal, they are known to yield near-optimal auctions. And even when they are only approximately optimal, seminal work has identified them as important objects of study owing to their simplicity.[1] While applying existing feed-forward architectures as RegretNet to symmetric auctions is possible, we show in §3 that RegretNet struggles to find symmetric auctions, *even when the optimum is symmetric*. To be clear, the architecture's performance is indeed quite close to optimal, but the resulting auction is not "close to symmetric". This paper proposes an architecture that outputs a symmetric auction symmetry by design.

Contributions

This paper presents a neural network architecture, EquivariantNet, that outputs symmetric auctions. This architecture is parameter-efficient and is able to recover some of the optimal results in the symmetric auctions literature. Our approach outlines three important benefits:

- *Symmetry*: our architecture outputs a symmetric auction by design. It is immune to permutation-sensitivity and exploitability as defined in 3.1.
- *Sample generalization*: Because we use domain knowledge, our architecture converges to the optimum with fewer valuation samples.
- *Out-of-setting generalization*: Our architecture does not require hard-coding the number of bidders or items during training — training our architecture on instances with n bidders and m items produces a well-defined auction even for instances with n' bidders and m' items. Somewhat surprisingly, we show in §4 some examples where our architecture trained on 1 bidder with 5 items generalizes well even to 1 bidder and m items, for any $m \in \{2, 10\}$.

EquivariantNet is an adaption of the deep sets architecture (Hartford et al., 2018) to symmetric auctions. We highlight that the novelty of this paper is not on proposing a new architecture but rather on adapting an existing method to return symmetric auctions and providing new understanding on these auctions. Our architecture can be seen as a tool for researchers to confirm or refute hypotheses and for this reason, we run our experiments on synthetic data. The paper decomposes as follows. §2 introduces the standard notions of auction design. §3 presents our permutation-equivariant architecture to encode symmetric auctions. Finally, §4 presents numerical evidence for the effectiveness of our approach.

Related work

Auction design and machine learning. Machine learning and computational learning theory have been used in several ways to design auctions from samples of bidder valuations. Some works have focused sample complexity results for designing optimal revenue-maximizing auctions. This has been established in single-parameter settings (Dhangwatnotai et al., 2015; Cole and Roughgarden, 2014; Morgenstern and Roughgarden, 2015; Medina and Mohri, 2014; Huang et al., 2018; Devanur et al., 2016; Hartline and Taggart, 2019; Roughgarden and Schrijvers, 2016; Gonczarowski and Nisan, 2017; Guo et al., 2019), multi-item auctions (Dughmi et al., 2014; Gonczarowski and Weinberg, 2018), combinatorial auctions (Balcan et al., 2016; Morgenstern and Roughgarden, 2016; Syrgkanis, 2017) and allocation mechanisms (Narasimhan and Parkes, 2016). Machine learning has also been used to optimize different aspects of mechanisms (Lahaie, 2011; Dütting et al., 2015). All these aforementioned differ from ours as we resort to deep learning for finding optimal auctions.

[1] In modern discussion of auctions, they are also desirable due to fairness considerations.

Auction design and deep learning. While Duetting et al. (2019) is the first paper to design auctions through deep learning, several other paper followed-up this work. Feng et al. (2018) extended it to budget constrained bidders, Golowich et al. (2018) to the facility location problem. Tacchetti et al. (2019) built architectures based on the Vickrey-Clarke-Groves auctions. Recently, Shen et al. (2019) and Duetting et al. (2019) proposed architectures that *exactly* satisfy incentive compatibility but are specific to *single-bidder* settings. In this paper, we aim at *multi-bidder* settings and build permutation-equivariant networks that return nearly incentive compatibility symmetric auctions.

2 Symmetries and learning problem in auction design

We review the framework of auction design and the problem of finding truthful mechanisms. We then present symmetric auctions and similarly to Duetting et al. (2019), frame auction design as a learning problem.

2.1 Auction design and symmetries

Auction design. We consider the setting of additive auctions with n bidders with $N = \{1, \ldots, n\}$ and m items with $M = \{1, \ldots, m\}$. Each bidder i is has value v_{ij} for item j, and values the set S of items at $\sum_{j \in S} v_{ij}$. Such valuations are called *additive*, and are perhaps the most well-studied valuations in multi-item auction design (Hart and Nisan, 2012, 2013; Li and Yao, 2013; Babaioff et al., 2014; Daskalakis et al., 2014; Hart and Reny, 2015; Cai et al., 2016; Daskalakis et al., 2017; Beyhaghi and Weinberg, 2019).

The designer does not know the full valuation profile $V = (v_{ij})_{i \in N, j \in M}$, but just a distribution from which they are drawn. Specifically, the valuation vector of bidder i for each of the m items $\vec{v}_i = (v_{i1}, \ldots, v_{im})$ is drawn from a distribution D_i over $\mathbb{R}^m$ (and then, V is drawn from $D := \times_i D_i$). The designer asks the bidders to report their valuations (potentially untruthfully), then decides on an allocation of items to the bidders and charges a payment to them.

Definition 1. *An auction is a pair (g, p) consisting of a randomized allocation rule $g = (g_1, \ldots, g_n)$ where $g_i \colon \mathbb{R}^{n \times m} \to [0, 1]^m$ such that for all V, and all j, $\sum_i (g_i(V))_j \leqslant 1$ and payment rules $p = (p_1, \ldots, p_n)$ where $p_i \colon \mathbb{R}^{n \times m} \to \mathbb{R}_{\geqslant 0}$.*

Given reported bids $B = (b_{ij})_{i \in N, j \in M}$, the auction computes an allocation probability $g(B)$ and payments $p(B)$. $[g_i(B)]_j$ is the probability that bidder i received object j and $p_i(B)$ is the price bidder i has to pay to the mechanism. In what follows, $\mathcal{M}$ denotes the class of all possible auctions.

Definition 2. *The utility of bidder i is defined by $u_i(\vec{v}_i, B) = \sum_{j=1}^{m} [g_i(B)]_j v_{ij} - p_i(B)$.*

Bidders seek to maximize their utility and may report bids that are different from their valuations. Let V_{-i} be the valuation profile without element $\vec{v}_i$, similarly for B_{-i} and $D_{-i} = \times_{j \neq i} D_j$. We aim at auctions that invite bidders to bid their true valuations through the notion of incentive compatibility.

Definition 3. *An auction (g, p) is dominant strategy incentive compatible (DSIC) if each bidder's utility is maximized by reporting truthfully no matter what the other bidders report. For every bidder i, valuation $\vec{v}_i \in D_i$, bid $\vec{b}_i{}' \in D_i$ and bids $B_{-i} \in D_{-i}$, $\quad u_i(\vec{v}_i, (\vec{v}_i, B_{-i})) \geqslant u_i(\vec{v}_i, (\vec{b}_i{}', B_{-i}))$.*

Additionally, we aim at auctions where each bidder receives a non-negative utility.

Definition 4. *An auction is individually rational (IR) if for all $i \in N$, $\vec{v}_i \in D_i$ and $B_{-i} \in D_{-i}$,*

$$u_i(\vec{v}_i, (\vec{v}_i, B_{-i})) \geqslant 0. \tag{IR}$$

In a DSIC auction, the bidders have the incentive to truthfully report their valuations and therefore, the revenue on valuation profile V is defined as $\sum_{i=1}^{n} p_i(V)$. Optimal auction design aims at finding a DSIC auction that maximizes the expected revenue $rev := \mathbb{E}_{V\sim D}[\sum_{i=1}^{n} p_i(V)]$.

Linear program. We frame the problem of optimal auction design as an optimization problem where we seek an auction that minimizes the negated expected revenue among all IR and DSIC auctions. Since there is no known characterization of DSIC mechanisms in the multi-bidder setting, we resort to the relaxed notion of *ex-post regret*. It measures the extent to which an auction violates DSIC, for each bidder.

Definition 5. *The ex-post regret for a bidder i is the maximum increase in his utility when considering all his possible bids and fixing the bids of others. For a valuation profile V, the ex-post regret for a bidder i is $rgt_i(V) = \max_{\vec{v}_i{}' \in \mathbb{R}^m} u_i(\vec{v}_i; (\vec{v}_i{}', V_{-i})) - u_i(\vec{v}_i; (\vec{v}_i, V_{-i}))$. In particular, DSIC is equivalent to*

$$rgt_i(V) = 0, \ \forall i \in N. \tag{IC}$$

Therefore, by setting (IC) and (IR) as constraints, finding an optimal auction is equivalent to the following linear program

$$\begin{aligned} \min_{(g,p)\in\mathcal{M}} -\mathbb{E}_{V\sim D}\left[\sum_{i=1}^{n} p_i(V)\right] \quad \text{s.t.} \quad & rgt_i(V) = 0, && \forall i \in N, \ \forall V \in D, \\ & u_i(\vec{v}_i, (\vec{v}_i, B_{-i})) \geqslant 0, && \forall i \in N, \ \vec{v}_i \in D_i, B_{-i} \in D_{-i}. \end{aligned} \tag{LP}$$

Symmetric auctions. (LP) is intractable due to the exponential number of constraints. However, in the setting of *symmetric* auctions, it is possible to reduce the search space of the problem as shown in Theorem 1. We first define the notions of bidder- and item-symmetries.

Definition 6. *The valuation distribution D is bidder-symmetric if for any permutation of the bidders $\varphi_b \colon N \to N$, the permuted distribution $D_{\varphi_b} := D_{\varphi_b(1)} \times \cdots \times D_{\varphi_b(n)}$ satifies: $D_{\varphi_b} = D$.*

Bidder-symmetry intuitively means that the bidders are a priori indistinguishable (although individual bidders will be different). This holds for instance in auctions where the identity of the bidders is anonymous, or if $D_i = D_j$ for all i, j (bidders are i.i.d.).

Definition 7. *Bidder i's valuation distribution D_i is item-symmetric if for any items $x_1, \dots, x_m$ and any permutation $\varphi_o \colon M \to M$, $D_i(x_{\varphi_o(1)}, \dots, x_{\varphi_o(m)}) = D_i(x_1, \dots, x_m)$.*

Intuitively, item-symmetry means that the items are also indistinguishable but not identical. It holds when the distributions over the items are i.i.d. but this is not a necessary condition. Indeed, the distribution $\{(a, b, c) \in \mathcal{U}(0, 1)^{\otimes 3} : a + b + c = 1\}$ is not i.i.d. but is item-symmetric.

Definition 8. *An auction is symmetric if its valuation distributions are bidder- and item-symmetric.*

We now define the notion of permutation-equivariance that is important in symmetric auctions.

Definition 9. *The functions g and p are permutation-equivariant if for any two permutation matrices $\Pi_n \in \{0,1\}^{n\times n}$ and $\Pi_m \in \{0,1\}^{m\times m}$, and any valuation matrix V, we have $g(\Pi_n \, V \, \Pi_m) = \Pi_n \, g(V) \, \Pi_m$ and $p(\Pi_n \, V \, \Pi_m) = \Pi_n \, p(V)$.*

Theorem 1. *When the auction is symmetric, there exists an optimal solution to (LP) that is permutation-equivariant.*

Theorem 1 is originally proved in Daskalakis and Weinberg (2012) and its proof is reminded in App. B for completeness. It encourages to reduce the search space in (LP) by only optimizing over permutation-equivariant allocations and payments. We implement this idea in Section 3 where we build equivariant neural network architectures. Before, we frame auction design as a learning problem.

2.2 Auction design as a learning problem

Similarly to Duetting et al. (2019), we formulate auction design as a learning problem. We learn a parametric set of auctions (g^w, p^w) where $w \in \mathbb{R}^d$ parameters and $d \in \mathbb{N}$. Directly solving (LP) is challenging in practice. Indeed, the auctioneer must have access to the bidder valuations which are unavailable to her. Since she has access to the valuation distribution, we relax (LP) and replace the IC constraint for all $V \in D$ by the expected constraint $\mathbb{E}_{V\sim D}[rgt_i(V)] = 0$ for all $i \in N$.. In practice, the expectation terms are computed by sampling L bidder valuation profiles drawn i.i.d. from D. The empirical ex-post regret for bidder i is

$$\widehat{rgt}_i(w) = \frac{1}{L}\sum_{\ell=1}^{L} \max_{\vec{v}_i{}' \in \mathbb{R}^m} u_i^w(\vec{v}_i^{(\ell)}; (\vec{v}_i{}', V_{-i}^{(\ell)})) - u_i(\vec{v}_i^{(\ell)}; (\vec{v}_i^{(\ell)}, V_{-i}^{(\ell)})), \tag{$\hat{R}$}$$

where $u_i^w(\vec{v}_i, B) := \sum_{j=1}^{m} [g_i^w(B)]_j v_{ij} - p_i^w(B)$ is the utility of bidder i under the parametric set of auctions (g^w, p^w). Therefore, the learning formulation of (LP) is

$$\min_{w\in\mathbb{R}^d} -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i=1}^{n} p_i^w(V^{(\ell)}) \quad \text{s.t.} \quad \widehat{rgt}_i(w) = 0, \ \forall i \in N. \tag{$\widehat{\mathrm{LP}}$}$$

Duetting et al. (2019) justify the validity of this reduction from (LP) to ($\widehat{\mathrm{LP}}$) by showing that the gap between the expected regret and the empirical regret is small as the number of samples increases. Additionally to being DSIC, the auction must satisfy IR. The learning problem ($\widehat{\mathrm{LP}}$) does not ensure this but we will show how to include this requirement in the architecture in §3.

3 Permutation-equivariant neural network architecture

We first show that feed-forward architectures as RegretNet (Duetting et al., 2019) may struggle to find a symmetric solution in auctions where the optimal solution is known to be symmetric. We then describe our neural network architecture, EquivariantNet that learns symmetric auctions. EquivariantNet is build using exchangeable matrix layers (Hartford et al., 2018).

3.1 Feed-forward nets and permutation-equivariance

In the following experiments we use the RegretNet architecture with the exact same training procedure and parameters as found in Duetting et al. (2019) .

Permutation-sensitivity. Given L bidders valuation samples $\{B^{(1)}, \ldots, B^{(L)}\} \in \mathbb{R}^{n\times m}$, we generate for each bid matrix $B^{(\ell)}$ all its possible permutations $B^{(\ell)}_{\Pi_n,\Pi_m} := \Pi_n B^{(\ell)} \Pi_m$, where $\Pi_n \in \{0,1\}^{n\times n}$ and $\Pi_m \in \{0,1\}^{m\times m}$ are permutation matrices. We then compute the revenue for each one of these bid matrices and obtain a revenue matrix $R \in \mathbb{R}^{n!m!\times L}$. Finally, we compute $h_R \in \mathbb{R}^L$ where $[h_R]_j = \max_{i\in[n!m!]} R_{ij} - \min_{i\in[n!m!]} R_{ij}$. The distribution given by the entries of h_R is a measure of how close the auction is to permutation-equivariance. A symmetric mechanism satisfies $h_R = (0, \ldots, 0)^\top$. Our numerical investigation considers the following auction settings:

– (I) One bidder and two items, the item values are drawn from $\mathcal{U}[0,1]$. Optimal revenue: 0.55 Manelli and Vincent (2006).

– (II) Four bidders and five items, the item values are drawn from $\mathcal{U}[0,1]$.

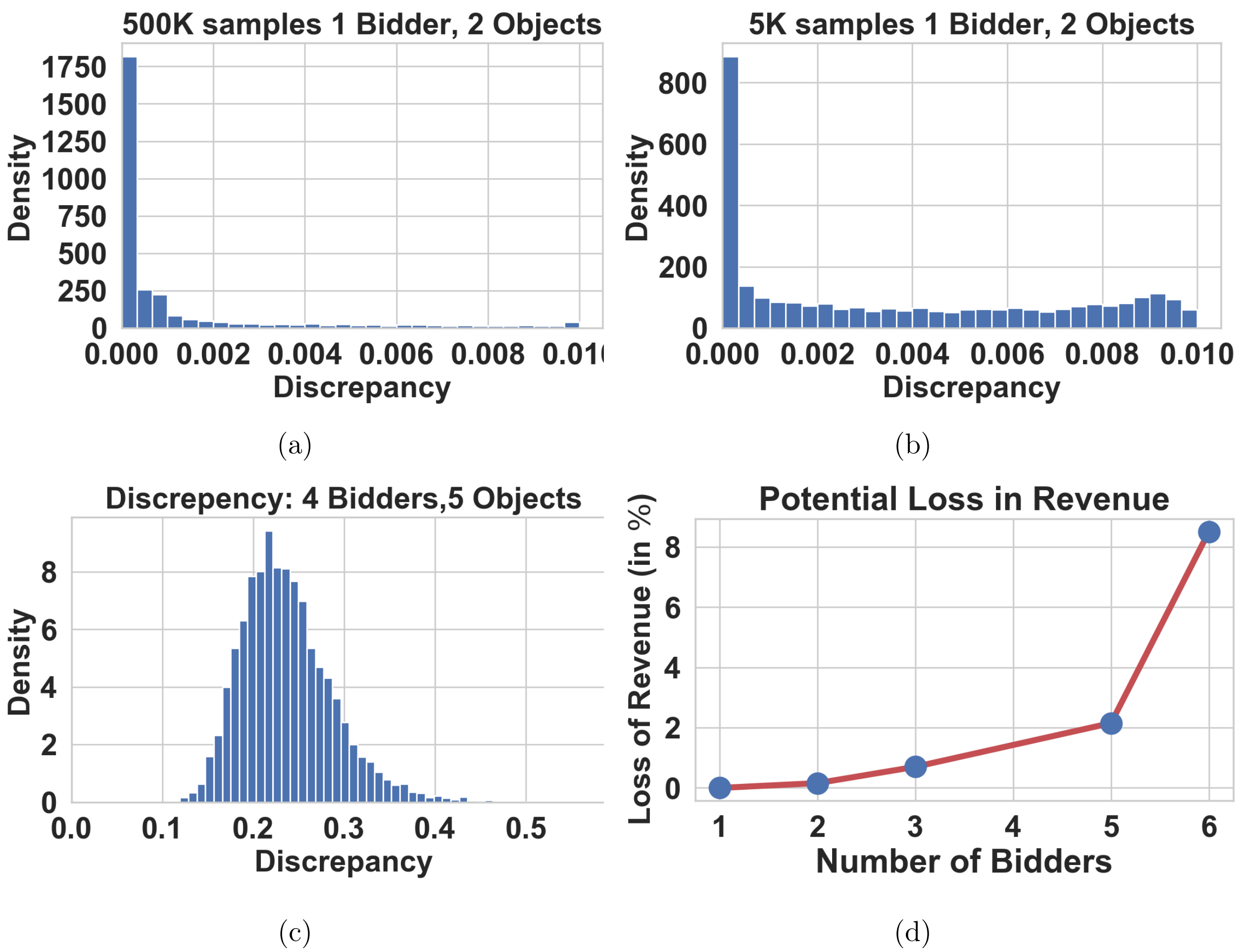


Figure 1: (a)-(b): Distribution h_R when varying the number of training samples (a) 500 000 (b) 5000 samples. (c): Histogram of the distribution h_R for setting (II). (d): Maximum revenue loss when varying the number of bidders for setting (III$_n$)
.

Fig. 1 (a)-(b) presents the distribution of h_R of the optimal auction learned for setting (I) when varying the number of samples L. When L is large, the distribution is almost concentrated at zero and therefore the network is almost able to recover the permutation-equivariant solution. When L is small, h_R is less concentrated around zero and therefore, the solution obtained is non permutation-equivariant.

As the problem's dimensions increase, this lack of permutation-invariance becomes more dramatic. Fig. 1 (c) shows h_R for the optimal auction mechanism learned for setting (II) when trained with $5 \cdot 10^5$ samples. Contrary to (I), almost no entry of h_R is located around zero, they are concentrated around between 0.1 and 0.4 i.e. between 3.8% and 15% of the estimated optimal revenue.

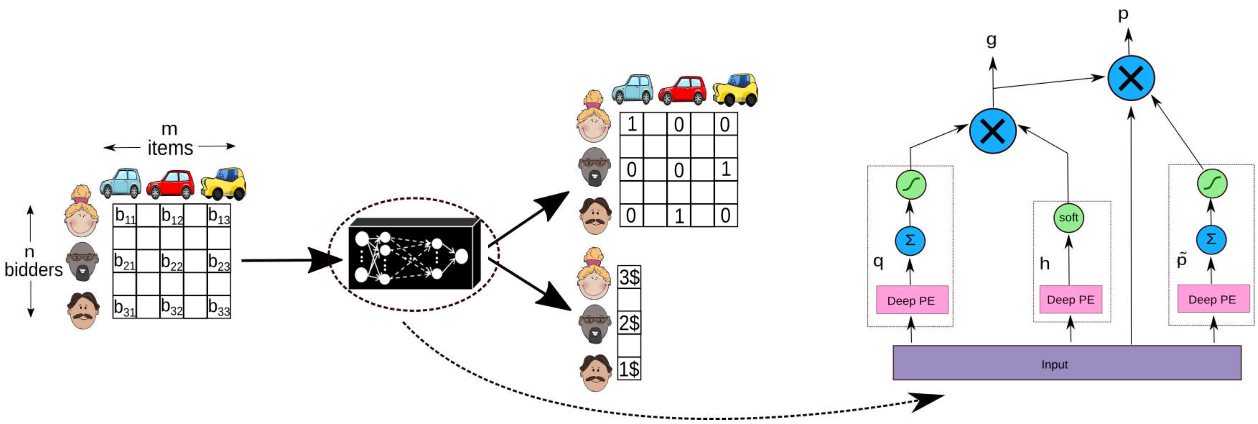


Figure 2: *Left:* Auction design setting. *Right:* EquivariantNet: Deep permutation-equivariant architecture for auction design. Deep PE denotes the deep permutation-equivariant architecture described in §3.2, $\sum$ the sum over rows/columns operations, $\times$ the multiplication operations, soft stands for soft-max and the curve for sigmoid. The network outputs an allocation g and a payment p.

Exploitability. Finally, to highlight how important equivariant solutions are, we analyze the worst-revenue loss that the auctioneer can incur when the bidders act adversarially. Indeed, since different permutations can result in different revenues for the auction, cooperative bidders could pick among the $n!$ possible permutations of their labels the one that minimized the revenue of the mechanism and present themselves in that order. Instead of getting a revenue of $R_{opt} = \mathbb{E}_{V\sim D}\left[\sum_{i=1}^n p_i(V)\right]$, the auctioneer would get a revenue of $R_{adv} = \mathbb{E}_{V\sim D}\left[\min_{\Pi_n}\{\sum_{i=1}^n p_i(\Pi_n V)\}\right]$. The percentage of revenue loss is given by $l = 100 \times \frac{R_{opt}-R_{adv}}{R_{opt}}$. We compute l in in the following family of settings:

– (III_n) n additive bidders and ten item where the item values are drawn from $\mathcal{U}[0,1]$.

In Fig. 1 (d) we plot $l(n)$ the loss in revenue as a function of n. As the number of bidders increases, the loss becomes more substantial getting over the 8% with only 6 bidders. Permutation-sensitivity and exploitability of fully connected architectures highlight the importance of aiming for symmetric auctions. To this end, we design a permutation-equivariant architecture.

3.2 Architecture for symmetric auctions (EquivariantNet)

Our input is a bid matrix $B = (b_{i,j}) \in \mathbb{R}^{n\times m}$ drawn from a bidder-symmetric and item-symmetric distribution. We aim at learning a randomized allocation neural network $g^w \colon \mathbb{R}^{n\times m} \to [0,1]^{n\times m}$ and a payment network $p^w \colon \mathbb{R}^{n\times m} \to \mathbb{R}^n_{\geqslant 0}$. The symmetries of the distribution from which B is drawn and Theorem 1 motivates us to model g^w and p^w as permutation-equivariant functions. To this end, we use *exchangeable matrix layers* (Hartford et al., 2018) and their definition is reminded in App. A. We now describe the three modules of the allocation and payment networks Fig. 2.

The first network outputs a vector $q^w(B) \in [0,1]^m$ such that entry $q^w_j(B)$ is the probability that item j is allocated to any of the n bidders. The architecture consists of three modules. The first one is a deep permutation-equivariant network with tanh activation functions. The output of that module is a matrix $Q \in \mathbb{R}^{n\times m}$. The second module transforms Q into a vector $\mathbb{R}^m$ by taking the average over the rows of Q. We finally apply the sigmoid function to the result to ensure that $q^w(B) \in [0,1]^m$. This architecture ensures that $q^w(B)$ is invariant with respect to bidder permutations and equivariant with respect to items permutations.

The second network outputs a matrix $h(B) \in [0,1]^{n\times m}$ where h^w_{ij} is the probability that item j is allocated to bidder i conditioned on item j being allocated. The architecture consists of a deep permutation-equivariant network with tanh activation functions followed by softmax activation function so that $\sum_{i=1}^n h^w_{ij}(B) = 1$. This architecture ensures that q^w equivariant with respect to object and bidder permutations.

By combining the outputs of q^w and h^w, we compute the allocation function $g^w \colon \mathbb{R}^{n\times m} \to [0,1]^{n\times m}$ where $g_{ij}(B)$ is the probability that the allocated item j is given to bidder i. Indeed, using conditional probabilities, we have $g^w_{ij}(B) = q^w_j(B)h^w_{ij}(B)$. Note that g^w is a permutation-equivariant function.

The third network outputs a vector $p(B) \in \mathbb{R}^n_{\geqslant 0}$ where $\tilde{p}^w_i$ is the fraction of bidder's i utility that she has to pay to the mechanism. Given the allocation function g^w, bidder i has to pay an amount $p_i = \tilde{p}_i(B)\sum_{j=1}^m g^w_{ij}(B)B_{ij}$. Individual rationality is ensured by having $\tilde{p}_i \in [0,1]$. The architecture of $\tilde{p}^w$ is almost similar to the one of q^w. Instead of averaging over the rows of the matrix output by the permutation-equivariant architecture, we average over the columns.

3.3 Optimization and training

The optimization and training procedure of EquivariantNet is similar to Duetting et al. (2019). For this reason, we briefly mention the outline of this procedure and remind the details in App. C. We apply the augmented Lagrangian method to ($\hat{R}$). The Lagrangian with a quadratic penalty is:

$$\mathcal{L}_\rho(w;\lambda) = -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i\in N} p^w_i(V^{(\ell)}) + \sum_{i\in N}\lambda_i \widehat{rgt}_i(w) + \frac{\rho}{2}\left(\sum_{i\in N}\widehat{rgt}_i(w)\right)^2,$$

where $\lambda \in \mathbb{R}^n$ is a vector of Lagrange multipliers and $\rho > 0$ is a fixed parameter controlling the weight of the quadratic penalty. The solver alternates between the updates on model parameters and Lagrange multipliers: $w^{new} \in \mathrm{argmax}_w \mathcal{L}_\rho(w^{old}, \lambda^{old})$ and $\lambda^{new}_i = \lambda^{old}_i + \rho \cdot \widehat{rgt}_i(w^{new})$, $\forall i \in N$.

4 Experimental Results

We start by showing the effectiveness of our architecture in symmetric and asymmetric auctions. We then highlight its sample-efficiency for training and its ability to extrapolate to other settings. More details about the setup and training can be found in App. C and App. D.

Evaluation. In addition to the revenue of the learned auction on a test set, we also evaluate the corresponding empirical average regret over bidders $\widehat{rgt} = \frac{1}{n}\sum_{i=1}^n \widehat{rgt}_i$. We evaluate these terms by running gradient ascent on v'_i with a step-size of 0.001 for $\{300, 500\}$ iterations (we test $\{100, 300\}$ different random initial v'_i and report the one achieves the largest regret).

Known optimal solution. We first consider instances of single bidder multi-item auctions where the optimal mechanism is known to be symmetric. While independent private value auction as (I) fall in this category, the following item-asymmetric auction has surprisingly an optimal symmetric solution.

- (IV) One bidder and two items where the item values are independently drawn according to the probability densities $f_1(x) = 5/(1+x)^6$ and $f_2(y) = 6/(1+y)^7$. Optimal solution in Daskalakis et al. (2017).

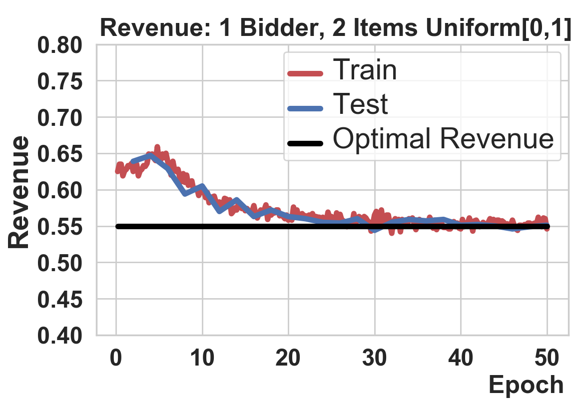
Dist.	rev	rgt	OPT
(I)	0.551	0.00013	0.550
(IV)	0.173	0.00003	0.1706
(V)	0.873	0.001	0.860

(a)

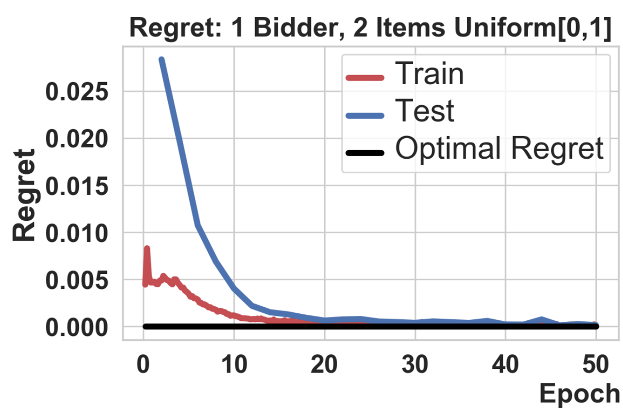
	EquivariantNet		RegretNet	
λ_2	rev	rgt	rev_F	rgt_F
0.01	0.37	0.0006	0.39	0.0003
0.1	0.41	0.0004	0.41	0.0007
1	0.86	0.0005	0.84	0.0012
10	3.98	0.0081	3.96	0.0056

(b)

Figure 3: (a): Test revenue and regret for (a) single bidder, 2 items and 2 bidders, 2 items settings. For this latter, OPT is the optimal revenue from VVCA and AMA_{bsym} families of auctions (Sandholm and Likhodedov, 2015). (b): Test revenue and regret for setting (VI) when varying λ_2 and $\lambda_1 = 1$. rev_F and rgt_F are computed with RegretNet (Duetting et al., 2019).



(a)



(b)

Figure 4: (a)-(b): Train and test revenues and regrets as a function of epochs for setting (I).

The two first lines in Fig. 3(a) report the revenue and regret of the mechanism learned by our model. The revenue is very close to the optimal one, and the regret is negligible. Remark that the learned auction may achieve a revenue slightly above the optimal incentive compatible auction. This is possible because although small, the regret is non-zero. Fig. 4(a)-(b) presents a plot of revenue and regret as a function of training epochs for the setting (I).

Unknown optimal solution. Our architecture is also able to recover a permutation-equivariant solution in settings for which the optimum is not known analytically such as:

– (V) Two additive bidders and two items where bidders draw their value for each item from $\mathcal{U}[0,1]$.

We compare our solution to the optimal auctions from the VVCA and AMA_{bsym} families of incentive compatible auctions from (Sandholm and Likhodedov, 2015). The last line of Fig. 3(a) summarizes our results.

Non-symmetric optimal solution. Our architecture returns satisfactory results in asymmetric auctions. (VI) is a setting where there may not be permutation-equivariant solutions.

– (VI) Two bidders and two items where the item values are independently drawn according to the probability densities $f_1(x) = \lambda_1^{-1} e^{-\lambda_1 x}$ and $f_2(y) = \lambda_2^{-1} e^{-\lambda_2 y}$, where $\lambda_1, \lambda_2 > 0$.

Fig. 3(b) shows the revenue and regret of the final auctions learned for setting (VI). When $\lambda_1 = \lambda_2$, the auction is symmetric and so, the revenue of the learned auction is very close to the optimal revenue, with negligibly small regret. However, as we increase the gap between λ_1 and λ_2, the asymmetry becomes dominant and the optimal auction does not satisfy permutation-equivariance. We remark that our architecture does output a solution with near-optimal revenue and small regret.

Sample-efficiency. Our permutation-equivariant architecture exhibits solid generalization properties when compared to the feed-forward architecture RegretNet. When enough data is available at training, both architectures generalize well to unseen data and the gap between the training and test losses goes to zero. However, when fewer training samples are available, our equivariant architecture generalizes while RegretNet struggles to. This may be explained by the inductive bias in our architecture. We demonstrate this for auction (V) with a training set of 20 samples and plot the training and test losses as a function of time (measures in epochs) for both architectures in Fig. 5(a).

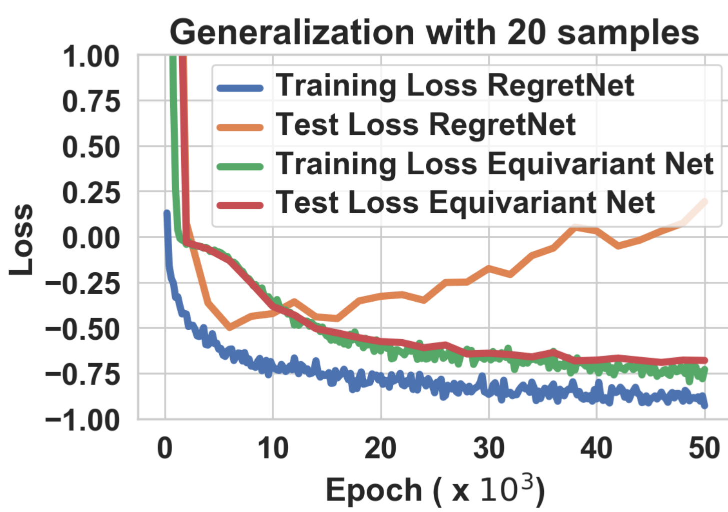


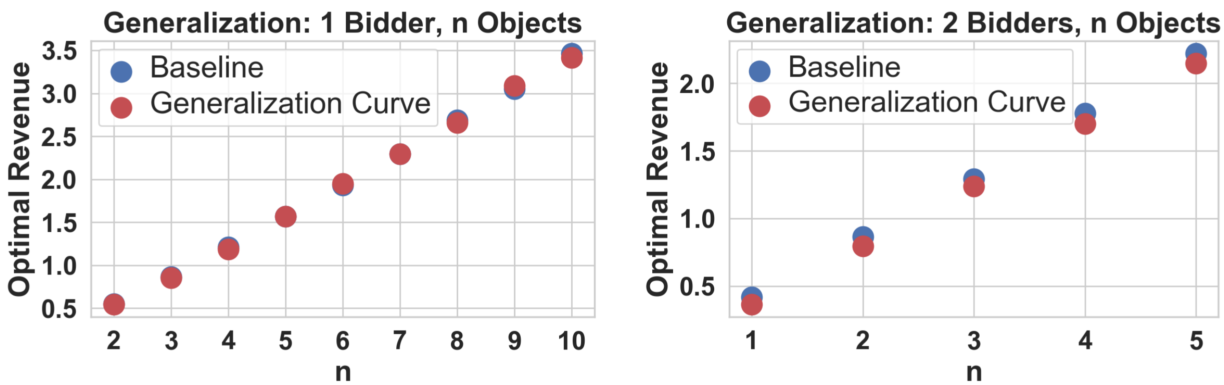
Figure 5: Train and test losses (V) with 20 training samples.

Out-of-setting generalization. The number of parameters in our permutation-equivariant architecture does not depend on the size of the input. Given an architecture that was trained on samples of size (n, m), it is possible to evaluate it on samples of any size (n', m'). This evaluation is not well defined for feed-forward architectures where the dimension of the weights depends on the input size. We use this advantage to check whether models trained in a fixed setting perform well in totally different ones.

– (α) Train an equivariant architecture on 1 bidder, 5 items and test it on 1 bidder, n items for $n = 2 \cdots 10$. All the items values are sampled independently from $\mathcal{U}[0, 1]$.

– (β) Train an equivariant architecture on 2 bidders, 3 objects and test it on 2 bidders, n objects for $n = 2 \cdots 6$. All the items values are sampled independently from $\mathcal{U}[0, 1]$.

Fig. 6(a)-(b) reports the test revenue that we get for different values of n in (α) and (β) and compares it to the empirical optimal revenue. Our baseline for that is RegretNet. Surprisingly, our model does generalize well. It is worth mentioning that knowing how to solve a larger problem such as 1×5 does not automatically result in a capacity to solve a smaller one such as 1×2; the generalization does happen on both ends. Our approach looks promising regarding out of setting generalization. It generalizes well when the number of objects varies and the number of bidders remain constants. However, generalization to settings where the number of bidders varies is more difficult due to the complex interactions between bidders. We do not observe good generalization with our current method.



(a) (b)

Figure 6: (a): Generalization revenue versus RegretNet for experiment (α). (b): Generalization revenue versus RegretNet for experiment (β).

Conclusion

We have explored the effect of adding domain knowledge in neural network architectures for auction design. We built a permutation-equivariant architecture to design symmetric auctions and highlighted its multiple advantages. It recovers several known optimal results and provides competitive results in asymmetric auctions. Compared to fully connected architectures, it is more sample efficient and is able to generalize to settings it was not trained on. In a nutshell, this paper insists on the importance of bringing domain-knowledge to the deep learning approaches for auctions.

Our architecture presents some limitations. It assumes that all the bidders and items are permutation-equivariant. However, in some real-world auctions, the item/bidder-symmetry only holds for a group of bidders/items. More advanced architectures such as Equivariant Graph Networks Maron et al. (2018) may solve this issue. Another limitation is that we only consider additive valuations. An interesting direction would be to extend our approach to other settings as unit-demand or combinatorial auctions.

Acknowledgements. We would like to thank Ryan P. Adams, Devon Graham and Andrea Tacchetti for helpful discussions. The work of Jad Rahme was funded by a Princeton SEAS Innovation Grant. Samy Jelassi and Joan Bruna's work are partially supported by the Alfred P. Sloan Foundation, NSF RI-1816753, NSF CAREER CIF 1845360, NSF CHS-1901091, Samsung Electronics, and the Institute for Advanced Study. The work of S. Matthew Weinberg was supported by NSF CCF-1717899.

References

Saeed Alaei. Bayesian Combinatorial Auctions: Expanding Single Buyer Mechanisms to Many Buyers. In *the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2011.

Saeed Alaei, Hu Fu, Nima Haghpanah, Jason Hartline, and Azarakhsh Malekian. Bayesian Optimal Auctions via Multi- to Single-agent Reduction. In *the 13th ACM Conference on Electronic Commerce (EC)*, 2012.

Saeed Alaei, Hu Fu, Nima Haghpanah, and Jason Hartline. The Simple Economics of Approximately Optimal Auctions. In *the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2013.

Saeed Alaei, Jason D. Hartline, Rad Niazadeh, Emmanouil Pountourakis, and Yang Yuan. Optimal auctions vs. anonymous pricing. *CoRR*, abs/1507.02615, 2015. URL http://arxiv.org/abs/1507.02615.

Moshe Babaioff, Nicole Immorlica, Brendan Lucier, and S. Matthew Weinberg. A simple and approximately optimal mechanism for an additive buyer. *SIGecom Exchanges*, 13(2):31–35, 2014. doi: 10.1145/2728732.2728736. URL http://doi.acm.org/10.1145/2728732.2728736.

Maria-Florina F Balcan, Tuomas Sandholm, and Ellen Vitercik. Sample complexity of automated mechanism design. In *Advances in Neural Information Processing Systems*, pages 2083–2091, 2016.

Hedyeh Beyhaghi and S. Matthew Weinberg. Optimal (and benchmark-optimal) competition complexity for additive buyers over independent items. In *Proceedings of the 51st ACM Symposium on Theory of Computing Conference (STOC)*, 2019.

Yang Cai, Constantinos Daskalakis, and S. Matthew Weinberg. An algorithmic characterization of multi-dimensional mechanisms. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 459–478, 2012a. doi: 10.1145/2213977.2214021. URL https://doi.org/10.1145/2213977.2214021.

Yang Cai, Constantinos Daskalakis, and S. Matthew Weinberg. Optimal multi-dimensional mechanism design: Reducing revenue to welfare maximization. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 130–139, 2012b. doi: 10.1109/FOCS.2012.88. URL https://doi.org/10.1109/FOCS.2012.88.

Yang Cai, Nikhil Devanur, and S. Matthew Weinberg. A duality based unified approach to bayesian mechanism design. In *Proceedings of the 48th ACM Conference on Theory of Computation(STOC)*, 2016.

Richard Cole and Tim Roughgarden. The sample complexity of revenue maximization. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 243–252, 2014.

Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. *arXiv preprint cs/0205075*, 2002.

Vincent Conitzer and Tuomas Sandholm. Self-interested automated mechanism design and implications for optimal combinatorial auctions. In *Proceedings of the 5th ACM conference on Electronic commerce*, pages 132–141, 2004.

Constantinos Daskalakis and Seth Matthew Weinberg. Symmetries and optimal multi-dimensional mechanism design. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 370–387, 2012.

Constantinos Daskalakis, Alan Deckelbaum, and Christos Tzamos. The Complexity of Optimal Mechanism Design. In *the 25th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2014.

Constantinos Daskalakis, Alan Deckelbaum, and Christos Tzamos. Strong duality for a multiple-good monopolist. *Econometrica*, 85(3):735–767, 2017.

Nikhil R. Devanur, Zhiyi Huang, and Christos-Alexandros Psomas. The sample complexity of auctions with side information. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 426–439, 2016. doi: 10.1145/2897518.2897553. URL http://doi.acm.org/10.1145/2897518.2897553.

Peerapong Dhangwatnotai, Tim Roughgarden, and Qiqi Yan. Revenue maximization with a single sample. *Games and Economic Behavior*, 91:318–333, 2015. doi: 10.1016/j.geb.2014.03.011. URL https://doi.org/10.1016/j.geb.2014.03.011.

Paul Duetting, Zhe Feng, Harikrishna Narasimhan, David Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1706–1715, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/duetting19a.html.

Shaddin Dughmi, Li Han, and Noam Nisan. Sampling and representation complexity of revenue maximization. In *International Conference on Web and Internet Economics*, pages 277–291. Springer, 2014.

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 (TEAC)*, 3(1):1–41, 2015.

Zhe Feng, Harikrishna Narasimhan, and David C Parkes. Deep learning for revenue-optimal auctions with budgets. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, pages 354–362. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Noah Golowich, Harikrishna Narasimhan, and David C. Parkes. Deep learning for multi-facility location mechanism design. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2018)*, pages 261–267, 2018. URL https://econcs.seas.harvard.edu/files/econcs/files/golowich_ijcai18.pdf.

Yannai A. Gonczarowski and Noam Nisan. Efficient empirical revenue maximization in single-parameter auction environments. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 856–868, 2017. doi: 10.1145/3055399.3055427. URL http://doi.acm.org/10.1145/3055399.3055427.

Yannai A. Gonczarowski and S. Matthew Weinberg. The sample complexity of up-to-ε multidimensional revenue maximization. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, 2018.

Chenghao Guo, Zhiyi Huang, and Xinzhi Zhang. Settling the sample complexity of single-parameter revenue maximization. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019.*, pages 662–673, 2019. doi: 10.1145/3313276.3316325. URL https://doi.org/10.1145/3313276.3316325.

Mingyu Guo and Vincent Conitzer. Computationally feasible automated mechanism design: General approach and case studies. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

Sergiu Hart and Noam Nisan. Approximate Revenue Maximization with Multiple Items. In *the 13th ACM Conference on Electronic Commerce (EC)*, 2012.

Sergiu Hart and Noam Nisan. The menu-size complexity of auctions. In *the 14th ACM Conference on Electronic Commerce (EC)*, 2013.

Sergiu Hart and Philip J. Reny. Maximizing Revenue with Multiple Goods: Nonmonotonicity and Other Observations. *Theoretical Economics*, 10(3):893–922, 2015.

Jason Hartford, Devon R Graham, Kevin Leyton-Brown, and Siamak Ravanbakhsh. Deep models of interactions across sets. *arXiv preprint arXiv:1803.02879*, 2018.

Jason D. Hartline and Tim Roughgarden. Simple versus optimal mechanisms. In *ACM Conference on Electronic Commerce*, pages 225–234, 2009.

Jason D. Hartline and Samuel Taggart. Sample complexity for non-truthful mechanisms. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019.*, pages 399–416, 2019. doi: 10.1145/3328526.3329632. URL https://doi.org/10.1145/3328526.3329632.

Zhiyi Huang, Yishay Mansour, and Tim Roughgarden. Making the most of your samples. *SIAM Journal on Computing*, 47(3):651–674, 2018.

Yaonan Jin, Weian Li, and Qi Qi. *On the Approximability of Simple Mechanisms for MHR Distributions*, pages 228–240. 12 2019a. ISBN 978-3-030-35388-9. doi: 10.1007/978-3-030-35389-6_17.

Yaonan Jin, Pinyan Lu, Qi Qi, Zhihao Gavin Tang, and Tao Xiao. Tight approximation ratio of anonymous pricing. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 674–685. ACM, 2019b. doi: 10.1145/3313276.3316331. URL https://doi.org/10.1145/3313276.3316331.

Yaonan Jin, Pinyan Lu, Zhihao Gavin Tang, and Tao Xiao. Tight revenue gaps among simple mechanisms. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 209–228. SIAM, 2019c.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Pravesh Kothari, Divyarthi Mohan, Ariel Schvartzman, Sahil Singla, and S Matthew Weinberg. Approximation schemes for a buyer with independent items via symmetries. *arXiv preprint arXiv:1905.05231*, 2019.

Sébastien Lahaie. A kernel-based iterative combinatorial auction. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

Xinye Li and Andrew Chi-Chih Yao. On revenue maximization for selling multiple independently distributed items. *Proceedings of the National Academy of Sciences*, 110(28):11232–11237, 2013.

Alejandro Manelli and Daniel Vincent. Bundling as an optimal selling mechanism for a multiple-good monopolist. *Journal of Economic Theory*, 127(1):1–35, 2006.

Alejandro M Manelli and Daniel R Vincent. Bayesian and dominant-strategy implementation in the independent private-values model. *Econometrica*, 78(6):1905–1938, 2010.

Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. *arXiv preprint arXiv:1812.09902*, 2018.

Andres M Medina and Mehryar Mohri. Learning theory and algorithms for revenue optimization in second price auctions with reserve. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 262–270, 2014.

Jamie Morgenstern and Tim Roughgarden. Learning simple auctions. In *Conference on Learning Theory*, pages 1298–1318, 2016.

Jamie H Morgenstern and Tim Roughgarden. On the pseudo-dimension of nearly optimal auctions. In *Advances in Neural Information Processing Systems*, pages 136–144, 2015.

Roger B Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, 1981.

Harikrishna Narasimhan and David C Parkes. A general statistical framework for designing strategy-proof assignment mechanisms. In *UAI'16 Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, 2016.

Gregory Pavlov. Optimal mechanism for selling two goods. *The B.E. Journal of Theoretical Economics*, 11(3), 2011.

Tim Roughgarden and Okke Schrijvers. Ironing in the dark. In *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, pages 1–18, 2016. doi: 10.1145/2940716.2940723. URL http://doi.acm.org/10.1145/2940716.2940723.

Tuomas Sandholm and Anton Likhodedov. Automated design of revenue-maximizing combinatorial auctions. *Operations Research*, 63(5):1000–1025, 2015.

Weiran Shen, Pingzhong Tang, and Song Zuo. Automated mechanism design via neural networks. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, pages 215–223. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

Vasilis Syrgkanis. A sample complexity measure with applications to learning optimal auctions. In *Advances in Neural Information Processing Systems*, pages 5352–5359, 2017.

Andrea Tacchetti, DJ Strouse, Marta Garnelo, Thore Graepel, and Yoram Bachrach. A neural architecture for designing truthful and efficient auctions. *arXiv preprint arXiv:1907.05181*, 2019.

William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

Zihe Wang and Pingzhong Tang. Optimal mechanisms with simple menus. In *ACM Conference on Economics and Computation, EC '14, Stanford , CA, USA, June 8-12, 2014*, pages 227–240, 2014. doi: 10.1145/2600057.2602863. URL https://doi.org/10.1145/2600057.2602863.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pages 3391–3401, 2017.

A Permutation-equivariant network

In this section, we remind *exchangeable matrix layers* introduced by (Hartford et al., 2018), that are a generalization of the deep sets architecture (Zaheer et al., 2017). We briefly describe this architecture here and invite the reader to look at the original paper for details.

The architecture consists in several layers and each of them is constituted of multiple channels. Each layer takes as input K channels and outputs O channels. Let $B^{(k)}$ and $Y^{(o)}$ denote such channels. Cross-channel interactions are fully connected: we have five unique parameters $w_1^{(k,o)}, \ldots, w_4^{(k,o)}$ for each combination of input-output channels and the bias parameter w_5 does not depend on the input. The element (i, j) of the o-th output channel is

$$Y_{i,j}^{(o)} = \sigma\left(\sum_{k=1}^{K} w_1^{(k,o)} B_{i,j}^{(k)} + \frac{w_2^{(k,o)}}{n}\sum_{i'} B_{i',j}^{(k)} + \frac{w_3^{(k,o)}}{m}\sum_{j'} B_{i,j'}^{(k)} + \frac{w_4^{(k,o)}}{nm}\sum_{i',j'} B_{i',j'}^{(k)} + w_5^{(o)}\right) \tag{1}$$

Each layer preserves permutation-equivariance since we only apply sums over lines and columns of $X^{(k)}$ in (1). By stacking multiple layers (1), we build a deep permutation-equivariant network.

B Proof of Theorem 1

Notation: For a matrix $B \in \mathbb{R}^{nm}$ we will denote the ith line by $B_i \in \mathbb{R}^m$ or $[B]_i \in \mathbb{R}^m$.Let D denote an equivariant distribution on $\mathbb{R}^{nm}$. Let $g\colon \mathbb{R}^{nm} \to \mathbb{R}^{nm}$ and $p\colon \mathbb{R}^{nm} \to \mathbb{R}^n$ be solutions to the following problem:

$$p = \text{argmax } \mathbb{E}_{B\sim D}\left[\sum_{i=1}^{n} p_i(B)\right]$$

subject to:

$$\langle\, [g(B)]_i\,,\, B_i\,\rangle \geqslant p_i(B),$$

and

$$\langle\, [g(B_i, B_{-i})]_i\,,\, B_i\,\rangle - p_i(B_i, B_{-i}) \geqslant \langle\, [g(B_i', B_{-i})]_i\,,\, B_i\,\rangle - p_i(B_i', B_{-i}), \quad \forall B_i' \in \mathbb{R}^m.$$

Let Π_n and Π_m be two permutation matrices of sizes n and m. In particular Π_n and Π_m are orthogonal matrices and in the following we use that $\Pi_n^{-1} = \Pi_n^T$ and $\Pi_m^{-1} = \Pi_m^T$. Let's define:

$$g^{\Pi_n,\Pi_m}(B) = \Pi_n^{-1}\, g(\Pi_n\, B\, \Pi_m)\, \Pi_m^{-1}$$
$$p^{\Pi_n,\Pi_m}(B) = \Pi_n^{-1}\, p(\Pi_n\, B\, \Pi_m).$$

Let's prove that if (g, p) is a solution to the problem then so is $(g^{\Pi_n,\Pi_m}, p^{\Pi_n,\Pi_m})$. First we show that $(g^{\Pi_n,\Pi_m}, p^{\Pi_n,\Pi_m})$ still satisfy the previous constraints.

$$\begin{aligned}
\langle [g^{\Pi_n,\Pi_m}(B)]_i\,,\, B_i\,\rangle &= \langle [\Pi_n^{-1} g(\Pi_n\, B\, \Pi_m)\Pi_m^{-1}]_i\,,\, B_i\,\rangle \\
&= \langle [\Pi_n^{-1} g(\Pi_n\, B\, \Pi_m)]_i \Pi_m^{-1}\,,\, B_i\,\rangle \\
&= \langle [\Pi_n^{-1} g(\Pi_n\, B\, \Pi_m)]_i\,,\, B_i \Pi_m\,\rangle \\
&= \langle [\Pi_n^{-1} g(\Pi_n\, B\, \Pi_m)]_i\,,\, [B\Pi_m]_i\,\rangle \\
&= \langle [\Pi_n^{-1} g(\Pi_n\, B\, \Pi_m)]_i\,,\, [B\Pi_m]_i\,\rangle.
\end{aligned}$$

Let's denote by φ the permutation on the indices corresponding to the Π_n permutation. then we have:

$$
\begin{aligned}
[\Pi_n^{-1} g(\Pi_n\, B\, \Pi_m)]_i &= [g(\Pi_n\, B\, \Pi_m)]_{\varphi^{-1}(i)} \\
[B\Pi_m]_i &= [\Pi_n B \Pi_m]_{\varphi^{-1}(i)}.
\end{aligned}
$$

This gives us that:

$$
\begin{aligned}
\langle [g^{\Pi_n,\Pi_m}(B)]_i \,,\, B_i \,\rangle &= \langle [\Pi_n^{-1} g(\Pi_n\, B\, \Pi_m)]_i \,,\, [B\Pi_m]_i \,\rangle \\
&= \langle [g(\Pi_n\, B\, \Pi_m)]_{\varphi^{-1}(i)} \,,\, [\Pi_n B \Pi_m]_{\varphi^{-1}(i)} \,\rangle \\
&\geqslant [p(\Pi_n B \Pi_m)]_{\varphi^{-1}(i)} \\
&= [\Pi_n^{-1} p(\Pi_n B \Pi_m)]_i \\
&= [p^{\Pi_n,\Pi_m}(B)]_i.
\end{aligned}
$$

This shows that $(g^{\Pi_n,\Pi_m}, p^{\Pi_n,\Pi_m})$ satisfies the first constraint. We now move to the second constraint.

Let's write $\tilde{B} = (B'_i, B_{-i})$. As a reminder, this is the matrix B where the ith line has been replaced with B'_i. We need to show that:

$$
\langle\, [g^{\Pi_n,\Pi_m}(B)]_i \,,\, B_i \,\rangle - p_i^{\Pi_n,\Pi_m}(B) \geqslant \langle\, [g^{\Pi_n,\Pi_m}(\tilde{B})]_i \,,\, B_i \,\rangle - p_i^{\Pi_n,\Pi_m}(\tilde{B}).
$$

Using the previous computations we find that:

$$
\langle\, [g^{\Pi_n,\Pi_m}(B)]_i \,,\, B_i \,\rangle - p_i^{\Pi_n,\Pi_m}(B) = \langle [g(\Pi_n\, B\, \Pi_m)]_{\varphi^{-1}(i)} \,,\, [\Pi_n B \Pi_m]_{\varphi^{-1}(i)} \,\rangle - [p(\Pi_n B \Pi_m)]_{\varphi^{-1}(i)},
$$

where φ is the permutation associated with Π_n. Since g and p satisfy the second constraint we have:

$$
\begin{aligned}
\langle\, [g^{\Pi_n,\Pi_m}(B)]_i \,,\, B_i \,\rangle - p_i^{\Pi_n,\Pi_m}(B) &= \langle [g(\Pi_n\, B\, \Pi_m)]_{\varphi^{-1}(i)} \,,\, [\Pi_n B \Pi_m]_{\varphi^{-1}(i)} \,\rangle - [p(\Pi_n B \Pi_m)]_{\varphi^{-1}(i)} \\
&\geqslant \langle [g(\Pi_n\, \tilde{B}\, \Pi_m)]_{\varphi^{-1}(i)} \,,\, [\Pi_n \tilde{B} \Pi_m]_{\varphi^{-1}(i)} \,\rangle - [p(\Pi_n \tilde{B} \Pi_m)]_{\varphi^{-1}(i)} \\
&= \langle\, [g^{\Pi_n,\Pi_m}(\tilde{B})]_i \,,\, B_i \,\rangle - p_i^{\Pi_n,\Pi_m}(\tilde{B}).
\end{aligned}
$$

This concludes the proof that $(g^{\Pi_n,\Pi_m}, p^{\Pi_n,\Pi_m})$ satisfy the constraints. Now we have to show that p^{Π_n,Π_m} is optimal.

$$
\begin{aligned}
\mathbb{E}_{B\sim D}\left[\sum_{i=1}^{n} p^{\Pi_n,\Pi_m}(B)\right] &= \mathbb{E}_{B\sim D}\left[\langle\, p^{\Pi_n,\Pi_m}(B) \,,\, \mathbf{1} \,\rangle\right] \\
&= \mathbb{E}_{B\sim D}\left[\langle\, \Pi_n^{-1}\, p(\Pi_n\, B\, \Pi_m) \,,\, \mathbf{1} \,\rangle\right] \\
&= \mathbb{E}_{B\sim D}\left[\langle\, p(\Pi_n\, B\, \Pi_m) \,,\, \mathbf{1} \,\rangle\right] \\
&= \mathbb{E}_{B\sim D}\left[\langle\, p(B) \,,\, \mathbf{1} \,\rangle\right] \\
&= \mathbb{E}_{B\sim D}\left[\sum_{i=1}^{n} p_i(B)\right],
\end{aligned}
$$

where we used that $\Pi_n^{-1} = \Pi_n^T$, $\Pi_n \mathbf{1} = \mathbf{1}$ and that $\Pi_n\, B\, \Pi_m \sim D$ since D is an equivariant distribution. This shows that if p is optimal then p^{Π_n,Π_m} is also optimal since they have the same expectation. We conclude that $(g^{\Pi_n,\Pi_m}, p^{\Pi_n,\Pi_m})$ is an optimal solution. Let's define

$$
\begin{aligned}
\tilde{g}(B) &= \mathbb{E}_{\Pi_n,\Pi_m}\left[g^{\Pi_n,\Pi_m}(B)\right] \\
\tilde{p}(B) &= \mathbb{E}_{\Pi_n,\Pi_m}\left[p^{\Pi_n,\Pi_m}(B)\right].
\end{aligned}
$$

Here, in the expectation, Π_n and Π_m are drawn uniformly at random. Since the problem and constraints are convex, $(\tilde{g}, \tilde{p})$ is also an optimal solution to the problem as a convex combination of optimal solutions. Let's prove that $\tilde{g}$ and $\tilde{p}$ are equivariant functions.

$$\begin{aligned}
\tilde{g}(\Pi_n \, B \, \Pi_m) &= \mathbb{E}_{\Pi'_n, \Pi'_m} \left[g^{\Pi'_n, \Pi'_m}(\Pi_n \, B \, \Pi_m) \right] \\
&= \mathbb{E}_{\Pi'_n, \Pi'_m} \left[{\Pi'_n}^{-1} g(\Pi'_m \, \Pi_n \, B \, \Pi_m \, \Pi'_m) {\Pi'_m}^{-1} \right] \\
&= {\Pi_n}^{-1} \; \mathbb{E}_{\Pi'_n, \Pi'_m} \left[(\Pi'_n \Pi_n)^{-1} g(\Pi'_n \, \Pi_n \, B \, \Pi_m \, \Pi'_m)(\Pi_m \Pi'_m)^{-1} \right] \; {\Pi_m}^{-1}.
\end{aligned}$$

If Π'_n and Π'_m are uniform among permutation then so is $\Pi'_n \Pi_n$ and $\Pi'_m \Pi_m$. So through a change of variable we find that:

$$\begin{aligned}
\tilde{g}(\Pi_n \, B \, \Pi_m) &= {\Pi_n}^{-1} \; \mathbb{E}_{\Pi'_n, \Pi'_m} \left[{\Pi'_n}^{-1} g(\Pi'_n \, B \, \Pi'_m) {\Pi'_m}^{-1} \right] \; {\Pi_m}^{-1} \\
&= {\Pi_n}^{-1} \; \tilde{g}(B) \; {\Pi_m}^{-1}.
\end{aligned}$$

This shows that $\tilde{g}$ is equivariant. The proof that $\tilde{p}$ is equivariant is similar.

$$\begin{aligned}
\tilde{p}(\Pi_n \, B \, \Pi_m) &= \mathbb{E}_{\Pi'_n, \Pi'_m} \left[p^{\Pi'_n, \Pi'_m}(\Pi_n \, B \, \Pi_m) \right] \\
&= \mathbb{E}_{\Pi'_n, \Pi'_m} \left[{\Pi'_n}^{-1} p(\Pi'_m \, \Pi_n \, B \, \Pi_m \, \Pi'_m) \right] \\
&= {\Pi_n}^{-1} \; \mathbb{E}_{\Pi'_n, \Pi'_m} \left[(\Pi'_n \Pi_n)^{-1} p(\Pi'_n \, \Pi_n \, B \, \Pi_m \, \Pi'_m) \right].
\end{aligned}$$

By doing a change of variable as before we find:

$$\begin{aligned}
\tilde{p}(\Pi_n \, B \, \Pi_m) &= {\Pi_n}^{-1} \; \mathbb{E}_{\Pi'_n, \Pi'_m} \left[{\Pi'_n}^{-1} p(\Pi'_n \, B \, \Pi'_m) \right] \\
&= {\Pi_n}^{-1} \; \tilde{p}(B),
\end{aligned}$$

$(\tilde{g}, \tilde{p})$ is an equivariant optimal solution, this concludes the proof.

C Optimization and training procedures

Our training algorithm is the same as the one found in Duetting et al. (2019). We made that choice to better illustrate the intrinsic advantages of our permutation equivariant architecture. We include implementation details here for completeness and additional details can be found in the original paper.

We generate a training dataset of valuation profiles $\mathcal{S}$ that we then divide into mini-batches of size B. Typical sizes for $\mathcal{S}$ are $\{5000, 50000, 500000\}$ and typical batch sizes are $\{50, 500, 50000\}$. We train our networks over for several epochs (typically $\{50, 80\}$) and we apply a random shuffling of the training data for each new epoch. We denote the minibatch received at iteration t by $\mathcal{S}_t = \{V^{(1)}, \ldots, V^{(B)}\}$. The update on model parameters involves an unconstrained optimization of $\mathcal{L}_\rho$ over w and is performed using a gradient-based optimizer. Let $\widehat{rgt}_i(w)$ be the empirical regret in (R) computed on mini-batch $\mathcal{S}_t$. The gradient of $\mathcal{L}_\rho$ with respect to w is given by:

$$\begin{aligned}
\nabla_w \mathcal{L}_\rho(w) = &-\frac{1}{B} \sum_{\ell=1}^{B} \sum_{i \in N} \nabla_w p_i^w(V^{(\ell)}) \\
&+ \sum_{i \in N} \sum_{\ell=1}^{B} \lambda_i^t g_{\ell,i} + \rho_t \sum_{i \in N} \sum_{\ell=1}^{B} \widehat{rgt}_i(w) g_{\ell,i},
\end{aligned} \tag{2}$$

Algorithm 1 Training Algorithm

Input: Minibatches $\mathcal{S}_1, \ldots, \mathcal{S}_T$ of size B
Parameters: $\gamma > 0$, $\eta > 0$, $c > 0$, $R \in \mathbb{N}$, $T \in \mathbb{N}$, $T_\rho \in \mathbb{N}$, $T_\lambda \in \mathbb{N}$.
Initialize Parameters: $\rho^0 \in \mathbb{R}$, $w^0 \in \mathbb{R}^d$, $\lambda^0 \in \mathbb{R}^n$,
Initialize Misreports: $v_i'^{(\ell)} \in V_i, \ \forall \ell \in [B], \, i \in N$.

for $t = 0, \ldots, T$ **do**
  Receive minibatch $\mathcal{S}_t = \{V^{(1)}, \ldots, V^{(B)}\}$.
  for $r = 0, \ldots, R$ **do**

$$\forall \ell \in [B], \, i \in n :$$
$$v_i'^{(\ell)} \leftarrow v_i'^{(\ell)} + \gamma \nabla_{v_i'} u_i^{w_t}(v_i{}^{(\ell)}; (v_i'^{(\ell)}, V_{-i}^{(\ell)}))$$

  Get Lagrangian gradient using (2) and update w^t:
    $w^{t+1} \leftarrow w^t - \eta \nabla_w \mathcal{L}_{\rho^t}(w^t)$.

  Update ρ once in T_ρ iterations:
  if t is a multiple of T_ρ **then**
    $\rho^{t+1} \leftarrow \rho^t + c$
  else
    $\rho^{t+1} \leftarrow \rho^t$

  Update Lagrange multipliers once in T_λ iterations:
  if t is a multiple of T_λ **then**
    $\lambda_i^{t+1} \leftarrow \lambda_i^t + \rho^t \, \widehat{rgt}_i(w^t), \forall i \in N$
  else
    $\lambda^{t+1} \leftarrow \lambda^t$

where

$$g_{\ell,i} = \nabla_w \left[\max_{v_i' \in V_i} u_i^w(v_i^{(\ell)}; (v_i', V_{-i}^{(\ell)})) - u_i^w(v_i^{(\ell)}; (v_i^{(\ell)}, V_{-i}^{(\ell)})) \right].$$

The terms $\widehat{rgt}_i$ and $g_{\ell,i}$ requires us to compute the maximum over misreports for each bidder i and valuation profile ℓ. To compute this maximum we optimize the function $v_i' \to u_i^w(v_i^{(\ell)}; (v_i', V_{-i}^{(\ell)}))$ using another gradient based optimizer.

For each i and valuation profile ℓ, we maintain a misreports valuation $v_i'^{(\ell)}$. For every update on the model parameters w^t, we perform R gradient updates to compute the optimal misreports: $v_i'^{(\ell)} = v_i'^{(\ell)} + \gamma \nabla_{v_i'^{(\ell)}} u_i^w(v_i^{(\ell)}; (v_i'^{(\ell)}, V_{-i}^{(\ell)}))$, for some $\gamma > 0$. In our experiments, we use the Adam optimizer (Kingma and Ba, 2014) for updates on model w and $v_i'^{(\ell)}$. Typical values are $R = 25$ and $\gamma = 0.001$ for the training phase. During testing, we use a larger number of step sizes R_{test} to compute these optimal misreports and we try bigger number initialization, N_{init}, that are drawn from the same distribution of the valuations. Typical values are $R_{test} = \{200, 300\}$ and $N_{init} = \{100, 300\}$. When the valuations are constrained to an interval (for instance $[0, 1]$), this optimization inner loop becomes constrained and we make sure that the values we get for v_i' are realistic by projecting them to their domain after each gradient step.

The parameters λ^t and ρ_t in the Lagrangian are not constant but they are updated over time. ρ_t is initialized at a value ρ_0 is incremented every T_ρ iterations, $\rho_{t+1} \leftarrow \rho_t + c$. Typical values are $\rho_0 = \{0.25, 1\}$, $c = \{0.25, 1, 5\}$ and $T_\rho = \{2, 5\}$ epochs. λ_t is initialized at a value λ_0 is updates every T_λ iterations according to $\lambda_i^{t+1} \leftarrow \lambda_i^t + \rho_t \widehat{rgt}_i(w^t), \forall i \in N$. Typical values are $\lambda_i^0 = \{0.25, 1, 5\}$ and $T_\lambda = \{2\}$ iterations.

D Setup

We implemented our experiments using PyTorch. A typical deep exchangeable network consists of 3 hidden layers of 25 channels each. Depending on the experiment, we generated a dataset of $\{5000, 50000, 500000\}$ valuation profiles and chose mini batches of sizes $\{50, 500, 5000\}$ for training. The optimization of the augmented Lagrangian was typically run for $\{50, 80\}$ epochs. The value of ρ in the augmented Lagrangian was set to 1.0 and incremented every 2 epochs. An update on w^t was performed for every mini-batch using the Adam optimizer with a learning rate of 0.001. For each update w^t, we ran $R = 25$ misreport update steps with a learning rate of 0.001. An update on λ^t was performed once every 100 minibatches.