Optimal Auctions through Deep Learning ♠

Paul Dütting [*1] **Zhe Feng** [*2] **Harikrishna Narasimham** [*2] **David C. Parkes** [*2] **Sai S. Ravindranath** [*2]

Abstract

Designing an incentive compatible auction that maximizes expected revenue is an intricate task. The single-item case was resolved in a seminal piece of work by Myerson in 1981. Even after 30-40 years of intense research the problem remains unsolved for seemingly simple multi-bidder, multi-item settings. In this work, we initiate the exploration of the use of tools from deep learning for the automated design of optimal auctions. We model an auction as a multi-layer neural network, frame optimal auction design as a constrained learning problem, and show how it can be solved using standard pipelines. We prove generalization bounds and present extensive experiments, recovering essentially all known analytical solutions for multi-item settings, and obtaining novel mechanisms for settings in which the optimal mechanism is unknown.

1. Introduction

Optimal auction design is one of the cornerstones of economic theory. It is of great practical importance, as auctions are used across industries and by the public sector to organize the sale of their products and services. Concrete examples are the US FCC Incentive Auction, the sponsored search auctions conducted by web search engines such as Google, or the auctions run on platforms such as eBay. In the standard *independent private valuations* model, each bidder has a valuation function over subsets of items, drawn independently from not necessarily identical distributions. It is assumed that the *auctioneer knows the distributions* and can (and will) use this information in designing the auction. A major difficulty in designing auctions is that valuations are *private* and bidders need to be incentivized to report their valuations truthfully. The goal is to learn an incentive compatible auction that maximizes revenue.

In a seminal piece of work, Myerson resolved the optimal auction design problem when there is a *single item* for sale (Myerson, 1981). Quite astonishingly, even after 30-40 years of intense research, the problem is not completely resolved even for a simple setting with *two bidders and two items*. While there have been some elegant partial characterization results (Manelli & Vincent, 2006; Pavlov, 2011; Haghpanah & Hartline, 2015; Giannakopoulos & Koutsoupias, 2015; Daskalakis et al., 2017; Yao, 2017), and an impressive sequence of recent algorithmic results (Cai et al., 2012b;a; 2013; Hart & Nisan, 2017; Babaioff et al., 2014; Yao, 2015; Cai & Zhao, 2017; Chawla et al., 2010), most of them apply to the weaker notion of Bayesian incentive compatibility (BIC). Our focus is on designing auctions that satisfy dominant-strategy incentive compatibility (DSIC), which is the more robust and desirable notion of incentive compatibility.

A recent, concurrent line of work started to bring in tools from machine learning and computational learning theory to design auctions from samples of bidder valuations. Much of the effort here has focused on analyzing the *sample complexity* of designing revenue-maximizing auctions (see e.g. Cole & Roughgarden (2014); Mohri & Medina (2016)). A handful of works has leveraged machine learning to optimize different aspects of mechanisms (Lahaie, 2011; Dütting et al., 2014; Narasimhan et al., 2016), but none of these offers the generality and flexibility of our approach. There have also been computational approaches to auction design, under the agenda of *automated mechanism design* (Conitzer & Sandholm, 2002; 2004; Sandholm & Likhodedov, 2015), but these are limited to specialized classes of auctions known to be incentive compatible.

Our contribution. In this work we provide the first, general purpose, end-to-end approach for solving the multi-item auction design problem. We use multi-layer neural networks to encode auction mechanisms, with bidder valuations being the input and allocation and payment decisions being the output. We then train the networks using samples from

---

♠ We thank Zihe Wang (Shanghai University of Finance and Economics) for pointing out that the combinatorial feasible definition in the published version of this paper need not imply an integer decomposition. This affects settings (IV) and (V). To address this issue, we add Appendix D, which includes additional constraints and shows that the experimental results remain almost unchanged.

[*]Equal contribution [1]London School of Economics [2]Harvard University. Correspondence to: Zhe Feng <zhe_feng@g.harvard.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

the value distributions, so as to maximize expected revenue subject to constraints for incentive compatibility.

To be able to tackle this problem using standard pipelines, we restate the incentive compatibility constraint as requiring the *expected ex post regret* for the auction to be zero. We adopt the *Augmented Lagrangian Method* to solve the resulting constrained optimization problem, where in each iteration we push gradients through the regret term, by solving an inner optimization problem to find the optimal misreport for each bidder and valuation profile.

We describe network architectures for bidders with additive, unit-demand, and combinatorial valuations, and present extensive experiments that show that:

(a) Our approach is capable of recovering essentially all analytical solutions for multi-item settings that have been obtained over the past 30-40 years by finding auctions with almost optimal revenue and vanishingly small regret that match the allocation and payment rules of the theoretically optimal auctions to surprising accuracy.

(b) Our approach finds high-revenue auctions with negligibly small regret in settings in which the optimal auction is unknown, matching or outperforming state-of-the-art computational results (Sandholm & Likhodedov, 2015).

(c) Whereas the largest setting presently studied in the analytical literature is one with 2 bidders and 2 items, our approach learns auctions for larger settings, such as a 5 bidder, 10 items setting, where optimal auctions have been hard to design, and finds low regret auctions that yield higher revenue than strong baselines.

We also prove a novel *generalization bound*, which implies that, with high probability, for our architectures high revenue and low regret on the training data translates into high revenue and low regret on freshly sampled valuations.

Discussion. By focusing on expected ex post regret we adopt a quantifiable relaxation of dominant-strategy incentive compatibility, first introduced in (Dütting et al., 2014). Our experiments suggest that this relaxation is an effective tool for approximating the optimal DSIC auctions.

While not strictly limited to neural networks our approach benefits from the expressive power of neural networks and the ability to enforce complex constraints in the training problem using the standard pipeline. A key advantage of our method over state-of-the-art automated mechanism design approaches (such as (Sandholm & Likhodedov, 2015)) is that we optimize over a broader class of not necessarily incentive compatible mechanisms, and are only constrained by the expressivity of the neural network architecture.

While the original work on automated auction design framed the problem as a linear program (LP) (Conitzer & Sandholm, 2002; 2004), follow-up works have acknowledged that this approach has severe scalablility issues as it requires a number of constraints and variables that is exponential in the number of agents and items (Guo & Conitzer, 2010). We find that even for small setting with 2 bidders and 3 items (and a discretization of the value into 5 bins per item) the LP takes 69 hours to complete since the LP needs to handle $\approx 10^5$ decision variables and $\approx 4 \times 10^6$ constraints. For the same setting, our approach found an auction with lower regret in just over 9 hours (see Table 1).

Further related work. Prior sample complexity results are available for the design of optimal single-item auctions (Cole & Roughgarden, 2014; Mohri & Medina, 2016; Huang et al., 2015), single bidder, multi-item auctions (Dughmi et al., 2014), general single-parameter settings (Morgenstern & Roughgarden, 2015), combinatorial auctions (Balcan et al., 2016; Morgenstern & Roughgarden, 2016; Syrgkanis, 2017), and allocation mechanisms (both with and without money) (Narasimhan & Parkes, 2016). Several other research groups have recently picked up deep nets and inference tools and applied them to economic problems, different from the one we consider here. These include the use of neural networks to predict behavior of human participants in strategic scenarios (Hartford et al., 2016), an automated equilibrium analysis of mechanisms (Thompson et al., 2017), deep nets for causal inference (Hartford et al., 2017; Louizos et al., 2017), and deep reinforcement learning for solving combinatorial games (Raghu et al., 2018).[1]

2. Auction Design as a Learning Problem

Auction design basics. We consider a setting with a set of n bidders $N = \{1, \ldots, n\}$ and m items $M = \{1, \ldots, m\}$. Each bidder i has a valuation function $v_i : 2^M \to \mathbb{R}_{\geq 0}$, where $v_i(S)$ denotes how much the bidder values the subset of items $S \subseteq M$. In the simplest case, a bidder may have additive valuations, where she has a value for individual items in M, and her value for a subset of items $S \subseteq M$: $v_i(S) = \sum_{j \in S} v_i(\{j\})$. Bidder i's valuation function is drawn independently from a distribution F_i over possible valuation functions V_i. We write $v = (v_1, \ldots, v_n)$ for a profile of valuations, and denote $V = \prod_{i=1}^{n} V_i$.

The auctioneer knows the distributions $F = (F_1, \ldots, F_n)$, but does not know the bidders' realized valuation v. The bidders report their valuations (perhaps untruthfully), and an auction decides on an allocation of items to the bidders and charges a payment to them. We denote an auction (g, p) as a pair of allocation rules $g_i : V \to 2^M$ and payment rules

[1] There has also been follow-up work to the present paper that extends our approach to budget constrained bidders (Feng et al., 2018) and to the facility location problem (Golowich et al., 2018), and that develops specialized architectures for single bidder settings that satisfy IC (Shen et al., 2019).

$p_i : V \to \mathbb{R}_{\geq 0}$ (these rules can be randomized). Given bids $b = (b_1, \ldots, b_n) \in V$, the auction computes an allocation $g(b)$ and payments $p(b)$.

A bidder with valuation v_i receives a utility $u_i(v_i, b) = v_i(g_i(b)) - p_i(b)$ for report of bid profile b. Bidders are strategic and seek to maximize their utility, and may report bids that are different from their valuations. Let v_{-i} denote the valuation profile $v = (v_1, \ldots, v_n)$ without element v_i, similarly for b_{-i}, and let $V_{-i} = \prod_{j \neq i} V_j$ denote the possible valuation profiles of bidders other than bidder i. An auction is *dominant strategy incentive compatible* (DSIC), if each bidder's utility is maximized by reporting truthfully no matter what the other bidders report. In other words, $u_i(v_i, (v_i, b_{-i})) \geq u_i(v_i, (b_i, b_{-i}))$ for every bidder i, every valuation $v_i \in V_i$, every bid $b_i \in V_i$, and all bids $b_{-i} \in V_{-i}$ from others. An auction is (ex post) *individually rational* (IR) if each bidder receives a non-zero utility, i.e. $u_i(v_i, (v_i, b_{-i})) \geq 0 \;\; \forall i \in N$, $v_i \in V_i$, and $b_{-i} \in V_{-i}$.

In a DSIC auction, it is in the best interest of each bidder to report truthfully, and so the revenue on valuation profile v is $\sum_i p_i(v)$. Optimal auction design seeks to identify a DSIC auction that maximizes expected revenue.

Formulation as a learning problem. We pose the problem of optimal auction design as a learning problem, where in the place of a loss function that measures error against a target label, we adopt the negated, expected revenue on valuations drawn from F. We are given a parametric class of auctions, $(g^w, p^w) \in \mathcal{M}$, for parameters $w \in \mathbb{R}^d$ (some $d \in \mathbb{N}$), and a sample of bidder valuation profiles $S = \{v^{(1)}, \ldots, v^{(L)}\}$ drawn i.i.d. from F.[2] The goal is to find an auction that minimizes the negated, expected revenue $-\sum_{i \in N} p_i^w(v)$, among all auctions in $\mathcal{M}$ that satisfy incentive compatibility.

In particular, we introduce constraints in the learning problem to ensure that the chosen auction satisfies incentive compatibility. For this, we define the *ex post* regret for each bidder to measure the extent to which an auction violates incentive compatibility. Fixing the bids of others, the ex post regret for a bidder is the maximum increase in her utility, considering all possible non-truthful bids. We will be interested in the *expected ex post regret* for bidder i:

$$rgt_i(w) = \mathbf{E}\Big[\max_{v_i' \in V_i} u_i^w(v_i; (v_i', v_{-i})) - u_i^w(v_i; (v_i, v_{-i}))\Big],$$

where the expectation is over $v \sim F$ and $u_i^w(v_i, b) = v_i(g_i^w(b)) - p_i^w(b)$ for given model parameters w. We assume that F has full support on the space of valuation profiles V, and recognizing that the regret is non-negative, an auction satisfies DSIC if and only if $rgt_i(w) = 0, \forall i \in N$.

[2]Note that there is no need to compute equilibrium inputs— we sample true profiles, and seek to learn rules that are IC.

Given this, we re-formulate the learning problem as minimizing the expected loss, i.e., the expected negated revenue s.t. the expected ex post regret being 0 for each bidder:

$$\min_{w \in \mathbb{R}^d} \mathbf{E}_{v \sim F}\Big[-\sum_{i \in N} p_i^w(v)\Big] \quad \text{s.t.} \quad rgt_i(w) = 0, \; \forall i \in N.$$

Given a sample S of L valuation profiles from F, we estimate the empirical ex post regret for bidder i as:

$$\widehat{rgt}_i(w) = \frac{1}{L}\sum_{\ell=1}^{L} \max_{v_i' \in V_i} u_i^w\big(v_i^{(\ell)}; \big(v_i', v_{-i}^{(\ell)}\big)\big) - u_i^w(v_i^{(\ell)}; v^{(\ell)}), \quad (1)$$

and seek to minimize the empirical loss subject to the empirical regret being zero for all bidders:

$$\min_{w \in \mathbb{R}^d} \; -\tfrac{1}{L}\textstyle\sum_{\ell=1}^{L}\sum_{i=1}^{n} p_i^w(v^{(\ell)}) \quad \text{s.t.} \quad \widehat{rgt}_i(w) = 0, \; \forall i \in N. \quad (2)$$

Individual Rationality. We will additionally require the designed auction to satisfy IR, which can be ensured by restricting our search space to a class of parametrized auctions (g^w, p^w) that charge no bidder more than her expected utility for an allocation. In Section 3, we will model the allocation and payment rules as neural networks and incorporate the IR requirement in the architecture.

Generalization bound. We bound the gap between the expected regret and the empirical regret in terms of the number of sampled valuations profiles. We show a similar result for revenue. Our bounds hold for any auction chosen from a finite capacity class, and imply that solving for (2) with a large sample yields an auction with near-optimal expected revenue and close-to-zero expected regret (we note that in practice, we may not be able to solve (2) exactly).

We measure the capacity of an auction class using a definition of covering numbers used in the ranking literature (Rudin & Schapire, 2009). We define the $\ell_{\infty,1}$ distance between auctions $(g, p), (g', p') \in \mathcal{M}$ as $\max_{v \in V} \sum_{i,j} |g_{ij}(v) - g'_{ij}(v)| + \sum_i |p_i(v) - p'_i(v)|$. For any $\epsilon > 0$, let $\mathcal{N}_\infty(\mathcal{M}, \epsilon)$ be the minimum number of balls of radius ϵ required to cover $\mathcal{M}$ under the $\ell_{\infty,1}$ distance.

Theorem 1. *For each bidder i, assume w.l.o.g. the valuation function $v_i(S) \leq 1, \forall S \subseteq M$. Let $\mathcal{M}$ be a class of auctions that satisfy individual rationality. Fix $\delta \in (0, 1)$. With probability at least $1 - \delta$ over draw of sample $\mathcal{S}$ of L profiles from F, for any $(g^w, p^w) \in \mathcal{M}$,*

$$\mathbf{E}_{v \sim F}\Big[-\sum_{i \in N} p_i^w(v)\Big] \leq -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i=1}^{n} p_i^w(v^{(\ell)}) + 2n\Delta_L + Cn\sqrt{\frac{\log(1/\delta)}{L}}$$

and

$$\frac{1}{n}\sum_{i=1}^{n} rgt_i(w) \;\leq\; \frac{1}{n}\sum_{i=1}^{n}\widehat{rgt}_i(w) \,+\, 2\Delta_L \,+\, C'\sqrt{\frac{\log(1/\delta)}{L}},$$

where $\Delta_L = \inf_{\epsilon>0}\left\{\frac{\epsilon}{n} + 2\sqrt{\frac{2\log(\mathcal{N}_\infty(\mathcal{M},\epsilon/2))}{L}}\right\}$ *and* C, C' *are distribution-independent constants.*

See the appendix for the proof. If the term Δ_L in the above bound goes down to 0 as the sample size L increases, the above bounds go to 0 as $L \to \infty$. In Theorem 2 in Section 3, we bound Δ_L for the neural network architectures we present in this paper.

3. Neural Network Architecture

We describe neural network architectures, which we refer to as *RegretNet*, for modeling multi-item auctions. We consider bidders with additive, unit-demand, and general combinatorial valuations. The architectures contain two logically distinct components: the allocation and payment networks.

Additive valuations. A bidder has additive valuations if the bidder's value for a bundle of items $S \subseteq M$ is the sum of her value for the individual items in S, i.e. $v_i(S) = \sum_{j\in S} v_i(j)$. In this case, the bidders report only their valuations for individual items. The architecture for this setting models a randomized allocation network $g^w : \mathbb{R}^{nm} \to [0,1]^{nm}$ and a payment network $p^w : \mathbb{R}^{nm} \to \mathbb{R}^n_{\geq 0}$, both of which are modeled as feed-forward, fully-connected networks with *tanh* activations. The input layer of the networks consists of bids b_{ij} representing the valuation of bidder i for item j.

The allocation network outputs a vector of allocation probabilities $z_{1j} = g_{1j}(b), \ldots, z_{nj} = g_{nj}(b)$, for each item $j \in [m]$. To ensure feasibility, i.e. that the probability of an item being allocated is at most 1, the allocations are computed using a *softmax activation function*, so that for all items j, $\sum_{i=1}^{n} z_{ij} \leq 1$. To accommodate the possibility of an item not being assigned to any bidder, we include a dummy node in the softmax computation which holds the residual allocation probabilities. Bundling of items is possible because the output units allocating items to the same bidder can be correlated. The payment network outputs a payment for each bidder that denotes the amount the bidder should pay in expectation, for this particular bid profile.

To ensure that the auction satisfies *individual rationality*, i.e. does not charge a bidder more than her expected value for the allocation, the network first computes a fractional payment $\tilde{p}_i \in [0,1]$ for each bidder i using a sigmoidal unit, and outputs a payment $p_i = \tilde{p}_i \sum_{j=1}^{m} z_{ij}\, b_{ij}$, where z_{ij}'s are outputs from the allocation network. An overview of the architecture is shown in Figure 1, where the revenue and regret are computed as functions of the parameters of the allocation and payment networks.

Unit-demand valuations. A bidder has unit-demand valuations when the bidder's value for a bundle of items $S \subseteq M$ is the maximum value she assigns to any one item in the bundle, i.e. $v_i(S) = \max_{j\in S} v_i(j)$. The allocation network for unit-demand bidders is the feed-forward network shown in Figure 2. For revenue maximization in this setting, it can be shown that it is sufficient to consider allocation rules that assign at most one item to each bidder.[3] In the case of randomized allocation rules, this would require that the total allocation for each bidder is at most 1, i.e. $\sum_j z_{ij} \leq 1,\ \forall i \in [n]$. We would also require that no item is over-allocated, i.e. $\sum_i z_{ij} \leq 1,\ \forall j \in [m]$. Hence, we design allocation networks for which the matrix of output probabilities $[z_{ij}]_{i,j=1}^{n}$ is doubly stochastic.[4]

In particular, we have the allocation network compute two sets of scores s_{ij}'s and s'_{ij}'s, with the first set of scores normalized along the rows, and the second set of scores normalized along the columns. Both normalizations can be performed by passing these scores through softmax functions. The allocation for bidder i and item j is then computed as the minimum of the corresponding normalized scores:

$$z_{ij} \,=\, \varphi^{DS}_{ij}(s, s') \,=\, \min\left\{\frac{e^{s_{ij}}}{\sum_{k=1}^{n+1} e^{s_{kj}}},\ \frac{e^{s'_{ij}}}{\sum_{k=1}^{m+1} e^{s'_{jk}}}\right\},$$

where indices $n+1$ and $m+1$ denote dummy inputs that correspond to an item not being allocated to any bidder, and a bidder not being allocated any item respectively.

Lemma 1. *$\varphi^{DS}(s,s')$ is doubly stochastic $\forall\, s, s' \in \mathbb{R}^{nm}$. For any doubly stochastic allocation $z \in [0,1]^{nm}$, $\exists\, s, s' \in \mathbb{R}^{nm}$, for which $z = \varphi^{DS}(s,s')$.*

The payment network is the same as in Figure 1.

Combinatorial valuations. We also consider bidders with general, combinatorial valuations. In the present work, we develop this architecture only for small number of items.[5] In this case, each bidder i reports a bid $b_{i,S}$ for every bundle of items $S \subseteq M$ (except the empty bundle, for which her valuation is taken as zero). The allocation network has an output $z_{i,S} \in [0,1]$ for each bidder i and bundle S, denoting the probability that the bidder is allocated the bundle. To prevent the items from being over-allocated, we require that the probability that an item appears in a bundle allocated

[3] This holds by a simple reduction argument: for any IC auction that allocates multiple items, one can construct an IC auction with the same revenue by retaining only the most-preferred item among those allocated to the bidder.

[4] A randomized allocation represented by a doubly-stochastic matrix can be decomposed into a lottery over deterministic one-to-one assignments (Birkhoff, 1946; von Neumann, 1953).

[5] With more items, combinatorial valuations can be succinctly represented using appropriate bidding languages; see, e.g. (Boutilier & Hoos, 2001).

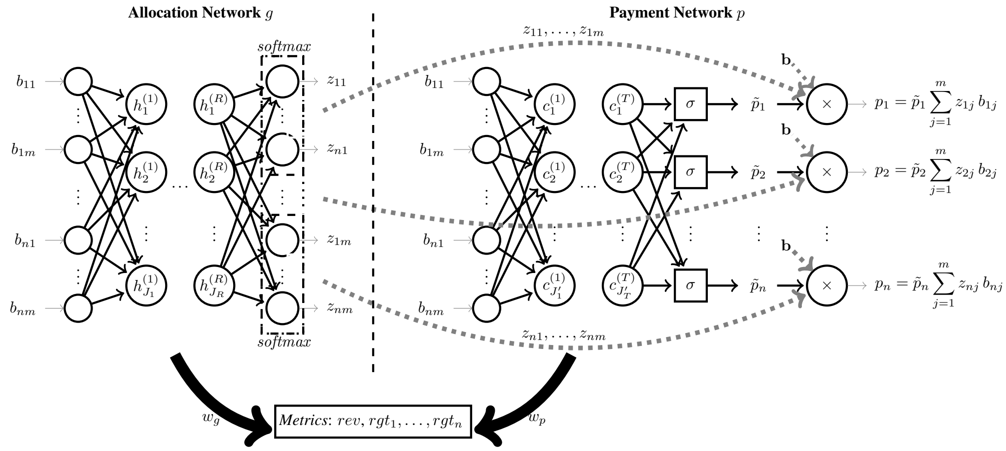
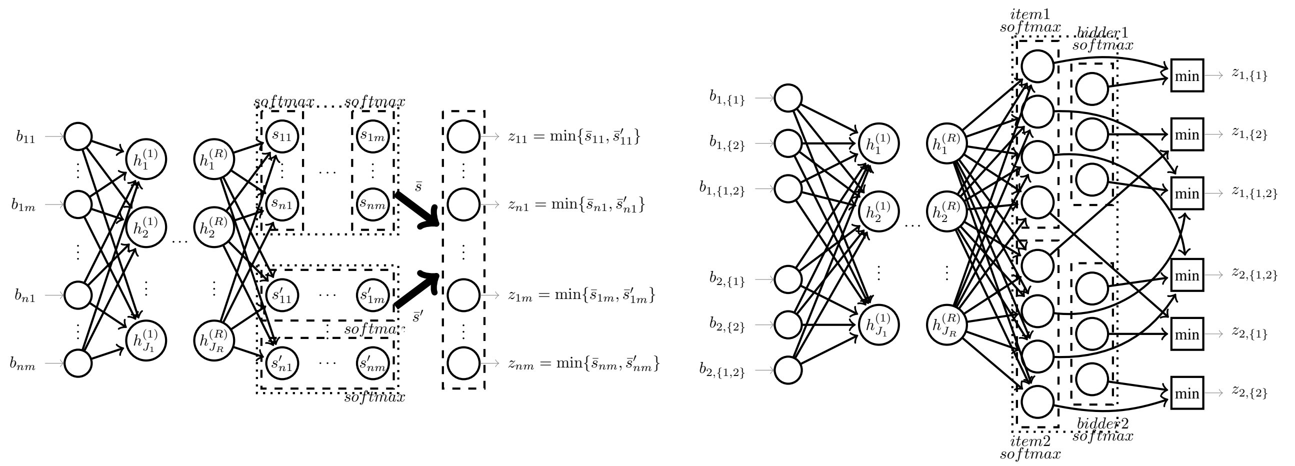


Figure 1: The allocation and payment networks for a setting with n additive bidders and m items. The inputs are bids from bidders for each item. The rev and each rgt_i are defined as a function of the parameters of the allocation and payment networks $w = (w_g, w_p)$.



(a) (b)

Figure 2: The allocation network for settings with (a) n *unit-demand* bidders and m items; and (b) 2 *combinatorial* bidders and 2 items. In (a), $\bar{s}_{ij} = e^{s_{ij}} / \sum_{k=1}^{n+1} e^{s_{kj}}$ and $\bar{s}'_{ij} = e^{s'_{ij}} / \sum_{k=1}^{m+1} e^{s'_{jk}}$. The payment networks for these settings are same as in Figure 1.

to some bidder is at most 1. We also require that the total allocation to a bidder is at most 1:

$$\sum_{i \in N} \sum_{S \subseteq M : j \in S} z_{i,S} \leq 1, \forall j \in M; \tag{3}$$

$$\sum_{S \subseteq M} z_{i,S} \leq 1, \forall i \in N. \tag{4}$$

We refer to an allocation that satisfies constraints (3)–(4) as being *combinatorial feasible*. To enforce these constraints, we will have the allocation network compute a set of scores for each item and a set of scores for each agent. Specifically, there is a group of bidder-wise scores $s_{i,S}, \forall S \subseteq M$ for each bidder $i \in N$, and a group of item-wise scores $s^{(j)}_{i,S}, \forall i \in N, S \subseteq M$ for each item $j \in M$. Each group of scores is normalized using a softmax function: $\bar{s}_{i,S} = \exp(s_{i,S}) / \sum_{S'} \exp(s_{i,S'})$ and $\bar{s}^{(j)}_{i,S} = \exp(s^{(j)}_{i,S}) / \sum_{i',S'} \exp(s^{(j)}_{i',S'})$. The allocation for bidder i and bundle $S \subseteq M$ is defined as the minimum of the normalized bidder-wise score $\bar{s}_{i,S}$ for i and the normalized item-wise scores $\bar{s}^{(j)}_{i,S}$ for each $j \in S$:

$$z_{i,S} = \varphi^{CF}_{i,S}(s, s^{(1)}, \ldots, s^{(m)}) = \min\left\{\bar{s}_{i,S}, \bar{s}^{(j)}_{i,S} : j \in S\right\}.$$

Lemma 2. *$\varphi^{CF}(s, s^{(1)}, \ldots, s^{(m)})$ is combinatorial feasible $\forall\, s, s^{(1)}, \ldots, s^{(m)} \in \mathbb{R}^{n2^m}$. For any combinatorial feasible allocation $z \in [0,1]^{n2^m}$, $\exists\, s, s^{(1)}, \ldots, s^{(m)} \in \mathbb{R}^{n2^m}$, for which $z = \varphi^{CF}(s, s^{(1)}, \ldots, s^{(m)})$.*

Figure 2(b) shows the network architecture for a setting with 2 bidders and 2 items. For ease of exposition, we ignore the empty bundle in our discussion. For each bidder $i \in \{1, 2\}$, the network computes three scores $s_{i,\{1\}}, s_{i,\{2\}}$, and $s_{i,\{1,2\}}$, one for each bundle that she can be assigned, and normalizes them using a softmax function. The network also computes four scores for item 1: $s^1_{1,\{1\}}, s^1_{2,\{1\}}, s^1_{1,\{1,2\}}$, and $s^1_{2,\{1,2\}}$, one for each assignment where item 1 is present, and similarly, four scores for item 2: $s^2_{1,\{2\}}, s^2_{2,\{2\}}, s^2_{1,\{1,2\}}$, and $s^2_{2,\{1,2\}}$. Each set of scores is then normalized by separate softmax functions. The final allocation for each bidder i is: $z_{i,\{1\}} = \min\{\bar{s}_{i,\{1\}}, \bar{s}^1_{i,\{1\}}\}$, $z_{i,\{2\}} = \min\{\bar{s}_{i,\{2\}}, \bar{s}^2_{i,\{2\}}\}$, and $z_{i,\{1,2\}} = \min\{\bar{s}_{i,\{1,2\}}, \bar{s}^1_{i,\{1,2\}}, \bar{s}^2_{i,\{1,2\}}\}$.

The payment network for combinatorial bidders has the same structure as the one in Figure 1, computing a fractional payment $\tilde{p}_i \in [0,1]$ for each bidder i using a sigmoidal unit, and outputting a payment $p_i = \tilde{p}_i \sum_{S \subseteq M} z_{i,S}\, b_{ij}$, where $z_{i,S}$'s are the outputs from the allocation network.

Covering number bounds. We now bound the term Δ_L in the generalization bound in Theorem 1 for the neural networks presented above.

Theorem 2. *For RegretNet with R hidden layers, K nodes per hidden layer, d_a parameters in the allocation network, d_p parameters in the payment network, and the vector of all model parameters $\|w\|_1 \leq W$, the following are the bounds on the term Δ_L for different bidder valuation types:*

(a) additive valuations:

$$\Delta_L \leq O\Big(\sqrt{R(d_a+d_p)\log(LW\max\{K, mn\})/L}\Big),$$

(b) unit-demand valuations:

$$\Delta_L \leq O\Big(\sqrt{R(d_a+d_p)\log(LW\max\{K, mn\})/L}\Big),$$

(c) combinatorial valuations:

$$\Delta_L \leq O\Big(\sqrt{R(d_a+d_p)\log(LW\max\{K, n\,2^m\})/L}\Big).$$

The proof is given in the appendix. As the sample size $L \to \infty$, the term $\Delta_L \to 0$. The dependence of the above result on the number of layers, nodes and parameters in the network is similar to standard covering number bounds for neural networks (Anthony & Bartlett, 2009). Note that the logarithm in the bound for combinatorial valuations cancels the exponential dependence on the number of items m.

4. Optimization and Training

We use the augmented Lagrangian method to solve the constrained training problem in (2) over the space of neural autoworker parameters w. We first define the Lagrangian function for the optimization problem, augmented with a quadratic penalty term for violating the constraints:

$$\mathcal{C}_\rho(w;\lambda) \;=\; -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i\in N} p_i^w(v^{(\ell)}) \;+\; \sum_{i\in N}\lambda_i\,\widehat{rgt}_i(w) \;+\; \frac{\rho}{2}\sum_{i\in N}\Big(\widehat{rgt}_i(w)\Big)^2$$

where $\lambda \in \mathbb{R}^n$ is a vector of Lagrange multipliers, and $\rho > 0$ is a fixed parameter that controls the weight on the quadratic penalty. The solver alternates between the following updates in each iteration on the model parameters and the Lagrange multipliers: (a) $w^{new} \in \operatorname{argmin}_w \mathcal{C}_\rho(w^{old};\lambda^{old})$ and (b) $\lambda_i^{new} = \lambda_i^{old} + \rho\,\widehat{rgt}_i(w^{new}),\ \forall i \in N$.

Algorithm 1 RegretNet Training

Input: Minibatches $\mathcal{S}_1,\ldots,\mathcal{S}_T$ of size B
Parameters: $\forall t, \rho_t > 0, \gamma > 0, \eta > 0, R \in \mathbb{N}, K \in \mathbb{N}$
Initialize: $w^0 \in \mathbb{R}^d$, $\lambda^0 \in \mathbb{R}^n$
for $t = 0$ **to** T **do**
  Receive minibatch $\mathcal{S}_t = \{u^{(1)},\ldots,u^{(B)}\}$
  Initialize misreports ${v'}_i^{(\ell)} \in V_i, \forall \ell \in [B],\ i \in N$
  for $r = 0$ **to** R **do**
    $\forall \ell \in [B],\ i \in N:$
      ${v'}_i^{(\ell)} \leftarrow {v'}_i^{(\ell)} + \gamma \nabla_{v'_i}\, u_i^w\big(v_i^{(\ell)}; ({v'}_i^{(\ell)}, v_{-i}^{(\ell)})\big)$
  end for
  Compute regret gradient: $\forall \ell \in [B], i \in N$:
    $g^t_{\ell,i} =$
    $\nabla_w\big[u_i^w\big(v_i^{(\ell)}; ({v'}_i^{(\ell)}, v_{-i}^{(\ell)})\big) - u_i^w(v_i^{(\ell)}; v^{(\ell)})\big]\Big|_{w=w^t}$
  Compute Lagrangian gradient using (5) and update w^t:
    $w^{t+1} \leftarrow w^t - \eta \nabla_w\, \mathcal{C}_{\rho_t}(w^t, \lambda^t)$
  Update Lagrange multipliers once in Q iterations:
    if t is a multiple of Q
      $\lambda_i^{t+1} \leftarrow \lambda_i^t + \rho_t\, \widetilde{rgt}_i(w^{t+1}),\ \ \forall i \in N$
    else
      $\lambda^{t+1} \leftarrow \lambda^t$
end for

The solver is described in Algorithm 1. We divide the training sample $\mathcal{S}$ into mini-batches of size B, and perform several passes over the training samples (with random shuffling of the data after each pass). We denote the minibatch received at iteration t by $\mathcal{S}_t = \{u^{(1)},\ldots,u^{(B)}\}$. The update (a) on model parameters involves an unconstrained optimization of $\mathcal{C}_\rho$ over w and is performed using a gradient-based optimizer. Let $\widetilde{rgt}_i(w)$ denote the empirical regret in (1) computed on mini-batch $\mathcal{S}_t$. The gradient of $\mathcal{C}_\rho$ w.r.t. w for fixed λ^t is given by:

$$\nabla_w\, \mathcal{C}_\rho(w;\,\lambda^t) \;=\; -\frac{1}{B}\sum_{\ell=1}^{B}\sum_{i\in N}\nabla_w\, p_i^w(v^{(\ell)}) \;+\; \sum_{i\in N}\sum_{\ell=1}^{B}\lambda_i^t\, g_{\ell,i} \;+\; \rho\sum_{i\in N}\sum_{\ell=1}^{B}\widetilde{rgt}_i(w)\, g_{\ell,i} \quad (5)$$

where

$$g_{\ell,i} \;=\; \nabla_w\Big[\max_{v'_i\in V_i} u_i^w\big(v_i^{(\ell)}; (v'_i, v_{-i}^{(\ell)})\big) - u_i^w(v_i^{(\ell)}; v^{(\ell)})\Big].$$

Note that the terms $\widetilde{rgt}_i$ and $g_{\ell,i}$ in turn involve a "max" over misreports for each bidder i and valuation profile ℓ. We solve the inner maximization over misreports using another gradient based optimizer, and push the gradient through the utility differences at the optimal misreports. In particular, we maintain misreports ${v'}_i^{(\ell)}$ for each i and valuation profile ℓ. For every update on the model parameters w^t, perform R gradient updates to compute the optimal misreports: ${v'}_i^{(\ell)} =$

${v'}_i^{(\ell)} + \gamma \nabla_{v'_i} u_i^w \big(v_i^{(\ell)}; \big({v'}_i^{(\ell)}, v_{-i}^{(\ell)}\big)\big)$, for some $\gamma > 0$. In our experiments, we use the Adam optimizer (Kingma & Ba, 2014) for updates on model w and ${v'}_i^{(\ell)}$.

Since the optimization problem we seek to solve is non-convex, the solver is not guaranteed to reach a globally optimal solution. However, our method proves very effective in our experiments. The learned auctions incur very low regret and closely match the structure of the optimal auctions in settings where this is known.

5. Experimental Results

We demonstrate that our approach can recover near-optimal auctions for essentially all settings for which the optimal solution is known and that it can find new auctions for settings where there is no known analytical solution. We present the complete set of experiments in the appendix and include a representative subset of the results here.

Setup. We implemented our framework using the TensorFlow deep learning library.[6] We used the Glorot uniform initialization (Glorot & Bengio, 2010) for all networks and the tanh activation function at the hidden nodes. For all the experiments, we used a sample of 640,000 valuation profiles for training and a sample of 10,000 profiles for testing. The augmented Lagrangian solver was run for a maximum of 80 epochs with a minibatch size of 128. The value of ρ in augmented Lagrangian was set to 1.0 and incremented every 2 epochs. An update on w^t was performed for every minibatch using the Adam optimizer with learning rate 0.001. For each update on w^t, we ran $R = 25$ misreport updates steps with learning rate 0.1. At the end of 25 updates, the optimized misreports for the current minibatch were cached and used to initialize the misreports for the same minibatch in the next epoch. An update on λ^t was performed once in every 100 minibatches (i.e. $Q = 100$). Our experiments were run on a compute cluster with NVDIA GPU cores.

Evaluation. In addition to the revenue of the learned auction on a test set, we also evaluate the regret, averaged across all bidders and test valuation profiles, $rgt = \frac{1}{n}\sum_{i=1}^{n} \widehat{rgt}_i(f, p)$. Each $\widehat{rgt}_i$ has a 'max' of the utility function over bidder valuations $v'_i \in V_i$ (see (1)). We evaluate these terms by running gradient ascent on v'_i with a step-size of 0.1 for 2000 iterations (we test 1000 different random initial v'_i and report the one achieves the largest regret).

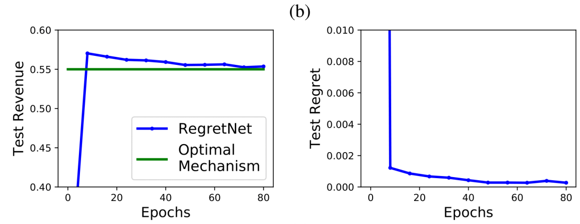
Single bidder. Even in the simple setting of single bidder auctions, there are analytical solutions only for special cases. We give the first computational approach that can handle the general design problem, and compare to the available analytical results. We show that not only are we able to learn auctions with near-optimal revenue, but we are also able to learn allocation rules that resemble the theoretically optimal rule with surprising accuracy.

Dist.	OPT	rev	rgt
(I)	0.550	0.554	< 0.001
(II)	2.137	2.137	< 0.001

(a)

Dist.	rev	rgt	VVCA	AMA_{bsym}
(III)	**0.878**	< 0.001	0.860	0.862
(IV)	**2.871**	< 0.001	2.741	2.765
(V)	**4.270**	< 0.001	4.209	3.748

(b)



(c)

Figure 3: (a)-(b): Test revenue and regret for (a) single bidder, 2 items and (b) 2 bidder, 2 items settings. (c): Plot of test revenue and regret as a function of training epochs for setting (I).

(I) Single bidder with additive valuations over 2 items, where the item values are drawn from $U[0, 1]$. The optimal auction is given by Manelli & Vincent (2006).

(II) Single bidder with unit-demand valuations over 2 items, where the item values are drawn from $U[2, 3]$. The optimal mechanism is given by Pavlov (2011).

Figure 3(a) presents the revenue and regret of the final auctions learned for settings (I) and (II) on the test set with an architecture with two hidden layers and 100 nodes per layer.[7] The revenue of the learned auctions is very close to the optimal revenue, with negligibly small regret. In some cases the learned auctions achieve revenue slightly above that of the optimal incentive compatible auction. This is possible because of the small, non-zero regret that they incur. The visualizations of the learned allocation rules in Figure 4(a)-(b) show that our approach also closely recovers the structure of the optimal auction. Figure 3(c) presents a plot of revenue and regret as a function of the training epochs. The solver adaptively tunes the Lagrange multiplier on the regret, focusing on the revenue in the initial iterations and on regret in later iterations.

Multiple bidders. We next compare to the state-of-the-art computational results of Sandholm and Likhodedov (Sandholm & Likhodedov, 2015) for settings for which the optimal auction is *not* known. These auctions are obtained by searching over a parametrized class of incentive compatible auctions. Unlike these prior methods, we do not need to search over a specific class of incentive compatible auction, and are limited only by the expressive power of the networks used. We show that this leads to novel auction designs that match or outperform the state-of-the-art mechanisms.

[6] https://github.com/saisrivatsan/deep-opt-auctions

[7] Based on evaluations on a held-out set, we found the gains to be negligible when we used more number of layers or nodes.

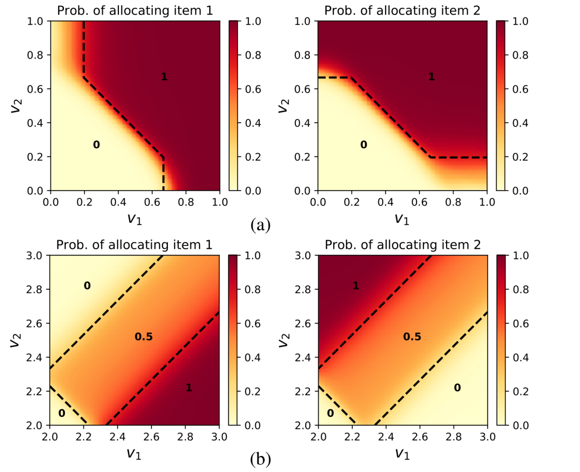


Figure 4: Allocation rules learned for single-bidder, two items settings: (a) I and (b) II. The solid regions describe the probability that the bidder is allocated item 1 (left) and item 2 (right) for different valuation inputs. The optimal auctions are described by the regions separated by the dashed black lines, with the numbers in black the optimal probability of allocation in the region.

(III) 2 additive bidders and 2 items, where bidders draw their value for each item from $U[0, 1]$.

(IV) 2 bidders and 2 items, with $v_{1,1}, v_{1,2}, v_{2,1}, v_{2,2} \sim U[1, 2]$, $v_{1,\{1,2\}} = v_{1,1} + v_{1,2} + C_1$ and $v_{2,\{1,2\}} = v_{2,1} + v_{2,2} + C_2$, where $C_1, C_2 \sim U[-1, 1]$.

(V) 2 bidders and 2 items, with $v_{1,1}, v_{1,2} \sim U[1, 2]$, $v_{2,1}, v_{2,2} \sim U[1, 5]$, $v_{1,\{1,2\}} = v_{1,1} + v_{1,2} + C_1$ and $v_{2,\{1,2\}} = v_{2,1} + v_{2,2} + C_2$, where $C_1, C_2 \sim U[-1, 1]$.

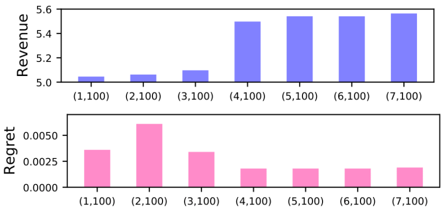
We adopt the same experimental setup as in settings (I)-(II). We compare the trained mechanism with the optimal auctions from the VVCA and AMA_{bsym} families of incentive compatible auctions from (Sandholm & Likhodedov, 2015). Figure 3(b) summarizes our results. Our approach leads to significant revenue improvements and tiny regret. Comparing with Figure 3(a), where the regret of (I) afforded a revenue advantage over OPT of around 0.004 or 0.72%, it seems highly unlikely that the tiny non-zero regret explains the revenue advantages over these prior results

Scaling up. We also consider settings with up to 5 bidders and 10 items. Due the exponential nature of the problem this is several orders of magnitude more complex than what the existing analytical literature can handle. For the settings that we study running a separate Myerson auction for each item is optimal in the limit of number of bidders (Palfrey, 1983). This yields a very strong but still improvable benchmark.

(VI) 3 additive bidders and 10 items, where bidders draw their value for each item from $U[0, 1]$.

(VII) 5 additive bidders and 10 items, where bidders draw their value for each item from $U[0, 1]$.

For setting (VI), we show in Figure 5(a) the revenue and regret of the learned auction on a validation sample of 10000



(a)

Distribution	rev	rgt	Item-wise Myerson	Bundled Myerson
VI: 3×10	**5.541**	< 0.002	5.310	5.009
VII: 5×10	**6.778**	< 0.005	6.716	5.453

(b)

Figure 5: (a) Revenue and regret on validation set for auctions learned for setting (VI) using different architectures. (b) Test revenue and regret for setting (VI) - (VII).

Dist.	Method	rev	rgt	*IR viol.*	Run-time
2×3	RegretNet	1.291	**< 0.001**	**0**	**$\sim$9 hrs**
	LP (D: 5 bins/value)	**1.53**	0.019	0.027	69 hrs

Table 1: Test revenue, regret and IR viol., run-time for RegretNet and LP for a 2 bidder, 3 items setting with uniform valuations.

profiles, obtained with different architectures. Here (R, K) denotes an architecture with R hidden layers and K nodes per layer. The $(5, 100)$ architecture has the lowest regret among all the 100-node networks for both settings above. Figure 5(b) shows that the final learned auctions yield higher revenue (with tiny regret) compared to the baselines.

Comparison to LP. We also compare the running time of our algorithm with the LP approach proposed (Conitzer & Sandholm, 2002; 2004). To be able to run the LP to completion, we consider a smaller setting with 2 additive bidders and 3 items, with item values drawn from from $U[0, 1]$. The LP is solved with the commercial solver *Gurobi*. We handle continuous valuations by discretizing the value into 5 bins per item (resulting in $\approx 10^5$ decision variables and $\approx 4 \times 10^6$ constraints) and then rounding a continuous input valuation profile to the nearest discrete profile (for evaluation). See the appendix for further discussion on LP.

The results are shown in Table 1. We also report the violations in IR constraints incurred by the LP on the test set; for L valuation profiles, this is measured by $\frac{1}{Ln} \sum_{\ell=1}^{L} \sum_{i \in N} \max\{u_i(v^{(\ell)}), 0\}$. Due to the coarse discretization, the LP approach suffers significant IR violations (and as a result yields higher revenue). We are not able to run a LP for this setting in more than 1 week of compute time for finer discretizations. In contrast, our approach yields much lower regret and no IR violations (as the neural networks satisfy IR by design), in just around 9 hours. In fact, even for the larger settings (VI)–(VII), the running time of our algorithm was less than 13 hours.

6. Conclusion

Neural networks have been deployed successfully for exploration in other contexts, e.g., for the discovery of new drugs (Gómez-Bombarelli et al., 2018). We believe that there is ample opportunity for applying deep learning in the context of economic design. We have demonstrated how standard pipelines can re-discover and surpass the analytical and computational progress in optimal auction design that has been made over the past 30-40 years. While our approach can easily solve problems that are orders of magnitudes more complex than what could previously be solved with the standard LP-based approach, a natural next step would be to scale this approach further up to industry scale. We envision progress at scale will come through addressing the benchmarking question (e.g., through standardized benchmarking suites), and through innovations in the network architecture.

Acknowledgments

This research is supported in part by NSF EAGER #124110.

References

Anthony, M. and Bartlett, P. L. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1st edition, 2009.

Babaioff, M., Immorlica, N., Lucier, B., and Weinberg, S. M. A simple and approximately optimal mechanism for an additive buyer. In *Proceedings of the 55th IEEE Symposium on Foundations of Computer Science*, pp. 21–30, 2014.

Balcan, M.-F., Sandholm, T., and Vitercik, E. Sample complexity of automated mechanism design. In *Proceedings of the 29th Conference on Neural Information Processing Systems*, pp. 2083–2091, 2016.

Birkhoff, G. Tres observaciones sobre el algebra lineal. *Universidad Nacional de Tucumán, Revista A*, 5:147–151, 1946.

Boutilier, C. and Hoos, H. H. Bidding languages for combinatorial auctions. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pp. 1211–1217, 2001.

Cai, Y. and Zhao, M. Simple mechanisms for subadditive buyers via duality. In *Proceedings of the 49th ACM Symposium on Theory of Computing*, pp. 170–183, 2017.

Cai, Y., Daskalakis, C., and Weinberg, M. S. Optimal multi-dimensional mechanism design: Reducing revenue to welfare maximization. In *Proceedings of the 53rd IEEE Symposium on Foundations of Computer Science*, pp. 130–139, 2012a.

Cai, Y., Daskalakis, C., and Weinberg, S. M. An algorithmic characterization of multi-dimensional mechanisms. In *Proceedings of the 44th ACM Symposium on Theory of Computing*, pp. 459–478, 2012b.

Cai, Y., Daskalakis, C., and Weinberg, S. M. Understanding incentives: Mechanism design becomes algorithm design. In *Proceedings of the 54th IEEE Symposium on Foundations of Computer Science*, pp. 618–627, 2013.

Chawla, S., Hartline, J. D., Malec, D. L., and Sivan, B. Multi-parameter mechanism design and sequential posted pricing. In *Proceedings of the 42th ACM Symposium on Theory of Computing*, pp. 311–320, 2010.

Cole, R. and Roughgarden, T. The sample complexity of revenue maximization. In *Proceedings of the 46th ACM Symposium on Theory of Computing*, pp. 243–252, 2014.

Conitzer, V. and Sandholm, T. Complexity of mechanism design. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pp. 103–110, 2002.

Conitzer, V. and Sandholm, T. Self-interested automated mechanism design and implications for optimal combinatorial auctions. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pp. 132–141, 2004.

Daskalakis, C. Multi-item auctions defying intuition? *SIGecom Exchanges*, 14(1):41–75, 2015.

Daskalakis, C., Deckelbaum, A., and Tzamos, C. Strong duality for a multiple-good monopolist. *Econometrica*, 85:735–767, 2017.

Dughmi, S., Han, L., and Nisan, N. Sampling and representation complexity of revenue maximization. In *Proceedings of the 10th Conference on Web and Internet Economics*, pp. 277–291, 2014.

Dütting, P., Fischer, F., Jirapinyo, P., Lai, J., Lubin, B., and Parkes, D. C. Payment rules through discriminant-based classifiers. *ACM Transactions on Economics and Computation*, 3(1):5, 2014.

Feng, Z., Narasimhan, H., and Parkes, D. C. Deep learning for revenue-optimal auctions with budgets. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, pp. 354–362, 2018.

Giannakopoulos, Y. and Koutsoupias, E. Selling two goods optimally. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming*, pp. 650–662, 2015.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010.

Golowich, N., Narasimhan, H., and Parkes, D. C. Deep learning for multi-facility location mechanism design. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 261–267, 2018.

Guo, M. and Conitzer, V. Computationally feasible automated mechanism design: General approach and case studies. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

Haghpanah, N. and Hartline, J. Multi-dimensional virtual values and second-degree price discrimination. *CoRR*, abs/1404.1341, 2015.

Hart, S. and Nisan, N. Approximate revenue maximization with multiple items. *Journal of Economic Theory*, 172: 313–347, 2017.

Hartford, J. S., Wright, J. R., and Leyton-Brown, K. Deep learning for predicting human strategic behavior. In *Proceedings of the 29th Conference on Neural Information Processing Systems*, pp. 2424–2432, 2016.

Hartford, J. S., Lewis, G., Leyton-Brown, K., and Taddy, M. Deep IV: A flexible approach for counterfactual prediction. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1414–1423, 2017.

Huang, Z., Mansour, Y., and Roughgarden, T. Making the most of your samples. In *Proceedings of the 16th ACM Conference on Economics and Computation*, pp. 45–60, 2015.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Lahaie, S. A kernel-based iterative combinatorial auction. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pp. 695–700, 2011.

Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R. S., and Welling, M. Causal effect inference with deep latent-variable models. In *Proceedings of the 30th Conference on Neural Information Processing Systems*, pp. 6449–6459, 2017.

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

Mohri, M. and Medina, A. M. Learning algorithms for second-price auctions with reserve. *Journal of Machine Learning Research*, 17:74:1–74:25, 2016.

Morgenstern, J. and Roughgarden, T. On the pseudo-dimension of nearly optimal auctions. In *Proceedings of the 28th Conference on Neural Information Processing Systems*, pp. 136–144, 2015.

Morgenstern, J. and Roughgarden, T. Learning simple auctions. In *Proceedings of the 29th Conference on Learning Theory*, pp. 1298–1318, 2016.

Myerson, R. Optimal auction design. *Mathematics of Operations Research*, 6:58–73, 1981.

Narasimhan, H. and Parkes, D. C. A general statistical framework for designing strategy-proof assignment mechanisms. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2016.

Narasimhan, H., Agarwal, S., and Parkes, D. C. Automated mechanism design without money via machine learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pp. 433–439, 2016.

Palfrey, T. Bundling decisions by a multiproduct monopolist with incomplete information. *Econometrica*, 51(2):463–83, 1983.

Pavlov, G. Optimal mechanism for selling two goods. *B.E. Journal of Theoretical Economics*, 11:1–35, 2011.

Raghu, M., Irpan, A., Andreas, J., Kleinberg, R., Le, Q. V., and Kleinberg, J. M. Can deep reinforcement learning solve Erdos-Selfridge-Spencer games? In *Proceedings of the 35th International Conference on Machine Learning*, pp. 4235–4243, 2018.

Rudin, C. and Schapire, R. E. Margin-based ranking and an equivalence between adaboost and rankboost. *Journal of Machine Learning Research*, 10:2193–2232, 2009.

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

Shalev-Shwartz, S. and Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014.

Shen, W., Tang, P., and Zuo, S. Automated mechanism design via neural networks. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, 2019. Forthcoming.

Syrgkanis, V. A sample complexity measure with applications to learning optimal auctions. In *Proceedings of the 20th Conference on Neural Information Processing Systems*, pp. 5358–5365, 2017.

Thompson, D., Newman, N., and Leyton-Brown, K. The positronic economist: A computational system for analyzing economic mechanisms. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pp. 720–727, 2017.

von Neumann, J. A certain zero-sum two-person game equivalent to an optimal assignment problem. *Annals of Mathematical Studies*, 28:5–12, 1953.

Yao, A. C.-C. An *n*-to-1 bidder reduction for multi-item auctions and its applications. In *Proceedings of the 26th ACM-SIAM Symposium on Discrete Algorithms*, pp. 92–109, 2015.

Yao, A. C.-C. Dominant-strategy versus bayesian multi-item auctions: Maximum revenue determination and comparison. In *Proceedings of the 18th ACM Conference on Economics and Computation*, pp. 3–20, 2017.

Optimal Auctions through Deep Learning

Appendix

A. Omitted Proofs

A.1. Proof of Lemma 1 and Proof of Lemma 2

Proof of Lemma 1. First, given the property of Softmax function and the min operation, $\varphi^{DS}(s, s')$ ensures that the row sums and column sums for the resulting allocation matrix do not exceed 1. In fact, for any doubly stochastic allocation z, there exists scores s and s', for which the min of normalized scores recovers z (e.g. $s_{ij} = s'_{ij} = log(z_{ij}) + c$ for any $c \in \mathbb{R}$). □

Proof of Lemma 2. Similar to Lemma 1, $\varphi^{CF}(s, s^{(1)}, \ldots, s^{(m)})$ trivially satisfies the combinatorial feasibility (constraints (3)–(4)). For any allocation z that satisfies the combinatorial feasibility, the following scores

$$\forall j \;=\; 1, \cdots, m, \quad s_{i,S} \;=\; s^{(j)}_{i,S} \;=\; \log(z_{i,S}) + c,$$

makes $\varphi^{CF}(s, s^{(1)}, \ldots, s^{(m)})$ recover z. □

A.2. Proof of Theorem 1

We present the proof for auctions with general, randomized allocation rules. A randomized allocation rule $g_i : V \to [0, 1]^{2^M}$ maps valuation profiles to a vector of allocation probabilities for bidder i. Here $g_{i,S}(v) \in [0, 1]$ denote the probability that the allocation rule assigns subset of items $S \subseteq M$ to bidder i, and $\sum_{S \subseteq M} g_{i,S}(v) \le 1$. Note that this encompasses the allocation rules we consider for additive and unit-demand valuations, which only output allocation probabilities for individual items. The payment function $p : V \to R^n$ maps valuation profiles to a payment for each bidder $p_i(v) \in \mathbb{R}$. For ease of exposition, we omit the superscripts "w". As before, $\mathcal{M}$ is a class of auctions (g, p).

We will assume that the allocation and payment rules in $\mathcal{M}$ are continuous and that the set of valuation profiles V is a compact set.

Notation. For any vectors $a, b \in \mathbb{R}^d$, the inner product is denoted as $\langle a, b\rangle \;=\; \sum_{i=1}^{d} a_i b_i$. For any matrix $A \in \mathbb{R}^{k\times\ell}$, the L_1 norm is given by $\|A\|_1 \;=\; \max_{1\le j\le\ell} \sum_{i=1}^{k} A_{ij}$.

Let $\mathcal{U}_i$ be the class of utility functions for bidder i defined on auctions in $\mathcal{M}$, i.e.:

$$\mathcal{U}_i \;=\; \Big\{u_i : V_i \times V \to \mathbb{R} \,\Big|\, u_i(v_i, b) \;=\; v_i(g(b)) \;-\; p_i(b) \text{ for some } (g, p) \in \mathcal{M}\Big\}.$$

and let $\mathcal{U}$ be the class of profile of utility functions defined on $\mathcal{M}$, i.e. the class of tuples $(u_1, \ldots, u_n)$ where each $u_i : V_i \times V \to \mathbb{R}$ and $u_i(v_i, b) \;=\; v_i(g(b)) \;-\; p_i(b), \forall i \in N$ for some $(g, p) \in \mathcal{M}$. We will sometimes find it useful to represent the utility function as an inner product, i.e. treating v_i as a real-valued vector of length 2^M, we may write $u_i(v_i, b) = \langle v_i, g_i(b)\rangle - p_i(b)$.

Let $\mathrm{rgt} \circ \mathcal{U}_i$ be the class of all regret functions for bidder i defined on utility functions in $\mathcal{U}_i$:

$$\mathrm{rgt} \circ \mathcal{U}_i = \left\{f_i : V \to \mathbb{R} \,\middle|\, f_i(v) \;=\; \max_{v'_i} u_i(v_i, (v'_i, v_{-i})) \;-\; u_i(v_i, v) \text{ for some } u_i \in \mathcal{U}_i\right\}$$

and as before, let $\mathrm{rgt} \circ \mathcal{U}$ be defined as the class of profiles of regret functions.

Define the $\ell_{\infty,1}$ distance between two utility functions u and u' as $\max_{v,v'} \sum_i |u_i(v_i, (v'_i, v_{-i})) - u_i(v_i, (v'_i, v_{-i}))|$ and $\mathcal{N}_\infty(\mathcal{U}, \epsilon)$ is the minimum number of balls of radius ϵ to cover $\mathcal{U}$ under this distance. Similarly, define the distance between u_i and u'_i as $\max_{v,v'_i} |u_i(v_i, (v'_i, v_{-i})) - u'_i(v_i, (v'_i, v_{-i}))|$, and let $\mathcal{N}_\infty(\mathcal{U}_i, \epsilon)$ denote the minimum number of balls of radius ϵ to cover$\mathcal{U}_i$ under this distance. Similarly, we define covering numbers $\mathcal{N}_\infty(\mathrm{rgt} \circ \mathcal{U}_i, \epsilon)$ and $\mathcal{N}_\infty(\mathrm{rgt} \circ \mathcal{U}, \epsilon)$ for the function classes $\mathrm{rgt} \circ \mathcal{U}_i$ and $\mathrm{rgt} \circ \mathcal{U}$ respectively.

Moreover, we denote the class of allocation functions as $\mathcal{G}$ and for each bidder i, $\mathcal{G}_i \;=\; \{g_i : V \to 2^M \mid g \in \mathcal{G}\}$. Similarly, we denote the class of payment functions by $\mathcal{P}$ and $\mathcal{P}_i \;=\; \{p_i : V \to \mathbb{R} \mid p \in \mathcal{P}\}$. We denote the covering number of $\mathcal{P}$ as $\mathcal{N}_\infty(\mathcal{P}, \epsilon)$ under the $\ell_{\infty,1}$ distance and the covering number for $\mathcal{P}_i$ using $\mathcal{N}_\infty(\mathcal{P}_i, \epsilon)$ under the ℓ_∞ distance.

We first state the following lemma from (Shalev-Shwartz & Ben-David, 2014). Let $\mathcal{F}$ be a class of functions $f : Z \to [-c, c]$ for some input space Z and $c > 0$. Given a sample $\mathcal{S} = \{z_1, \ldots, z_L\}$ of points from Z, define the empirical Rademacher complexity of $\mathcal{F}$ as:

$$\hat{\mathcal{R}}_L(\mathcal{F}) := \frac{1}{L}\mathbf{E}_\sigma\left[\sup_{f\in\mathcal{F}}\sum_{z_i\in S}\sigma_i f(z_i)\right],$$

where $\sigma \in \{-1, 1\}^L$ and each σ_i is drawn i.i.d from a uniform distribution on $\{-1, 1\}$.

Lemma 3 (Generalization bound in terms of Rademacher complexity)**.** *Let $\mathcal{S} = \{z_1, \ldots, z_L\}$ be a sample drawn i.i.d. from some distribution D over Z. Then with probability of at least $1 - \delta$ over draw of $\mathcal{S}$ from D, for all $f \in \mathcal{F}$,*

$$\mathbf{E}_{z\in D}[f(z)] \leq \frac{1}{L}\sum_{i=1}^{L} f(z_i) \,+\, 2\hat{\mathcal{R}}_L(\mathcal{F}) \,+\, 4c\sqrt{\frac{2\log(4/\delta)}{L}}.$$

We are now ready to prove Theorem 1. We begin with the first part, namely a generalization bound for revenue.

Proof of Theorem 1 (Part 1). The proof involves a direct application of Lemma 3 to the class of revenue functions defined on $\mathcal{M}$:

$$\mathrm{rev}\circ\mathcal{M} \,=\, \left\{f : V \to \mathbb{R} \,\middle|\, f(v) \,=\, \textstyle\sum_{i=1}^{n} p_i(v),\ \text{for some } (g, p) \in \mathcal{M}\right\},$$

and bounds the Rademacher complexity term for this class in terms of the covering number for the payment class $\mathcal{P}$, which in turn is bounded by the covering number for the auction class for $\mathcal{M}$.

Since we assume that the auctions in $\mathcal{M}$ satisfy individual rationality and the valuation functions are bounded in $[0, 1]$, we have for any v, $p_i(v) \leq 1$. By definition of the covering number $\mathcal{N}_\infty(\mathcal{P}, \epsilon)$ for the payment class, for any $p \in \mathcal{P}$, there exists a $f_p \in \hat{\mathcal{P}}$ where $|\hat{\mathcal{P}}| \leq \mathcal{N}_\infty(\mathcal{P}, \epsilon)$, such that $\max_v \sum_i |p_i(v) - f_{p_i}(v)| \leq \epsilon$. First we bound the Rademacher complexity, for a given $\epsilon \in (0, 1)$,

$$\begin{aligned}
\hat{\mathcal{R}}_L(\mathrm{rev}\circ\mathcal{M}) &= \frac{1}{L}\mathbf{E}_\sigma\left[\sup_p \sum_{\ell=1}^{L}\sigma_\ell \cdot \sum_i p_i(v^{(\ell)})\right] \\
&= \frac{1}{L}\mathbf{E}_\sigma\left[\sup_p \sum_{\ell=1}^{L}\sigma_\ell \cdot \sum_i f_{p_i}(v^{(\ell)})\right] + \frac{1}{L}\mathbf{E}_\sigma\left[\sup_p \sum_{\ell=1}^{L}\sigma_\ell \cdot \sum_i p_i(v^{(\ell)}) - f_{p_i}(v^{(\ell)})\right] \\
&\leq \frac{1}{L}\mathbf{E}_\sigma\left[\sup_{\hat{p}\in\hat{\mathcal{P}}} \sum_{\ell=1}^{L}\sigma_\ell \cdot \sum_i \hat{p}_i(v^{(\ell)})\right] + \frac{1}{L}\mathbf{E}_\sigma\|\sigma\|_1\epsilon \\
&\leq \sqrt{\sum_\ell\Big(\sum_i \hat{p}_i(v^\ell)\Big)^2}\sqrt{\frac{2\log(\mathcal{N}_\infty(\mathcal{P}, \epsilon))}{L}} + \epsilon \quad \text{(By Massart's Lemma)} \\
&\leq 2n\sqrt{\frac{2\log(\mathcal{N}_\infty(\mathcal{P}, \epsilon))}{L}} + \epsilon.
\end{aligned}$$

The last inequality is because

$$\sqrt{\sum_\ell\left(\sum_i \hat{p}_i(v^\ell)\right)^2} \leq \sqrt{\sum_\ell\left(\sum_i p_i(v^\ell) + n\epsilon\right)^2} \leq 2n\sqrt{L}.$$

Next we show $\mathcal{N}_\infty(\mathcal{P}, \epsilon) \leq \mathcal{N}_\infty(\mathcal{M}, \epsilon)$, for any $(g, p) \in \mathcal{M}$, take $(\hat{g}, \hat{p})$ s.t. for all v

$$\sum_{i,j}|g_{ij}(v) - \hat{g}_{ij}(v)| + \sum_i |p_i(v) - \hat{p}_i(v)| \leq \epsilon.$$

Thus for any $p \in \mathcal{P}$, for all v, $\sum_i |p_i(v) - \hat{p}_i(v)| \leq \epsilon$, which implies $\mathcal{N}_\infty(\mathcal{P}, \epsilon) \leq \mathcal{N}_\infty(\mathcal{M}, \epsilon)$. Applying Lemma 3 and $\sum_i p_i(v) \leq n$ for any v, with probability of at least $1 - \delta$,

$$\mathbf{E}_{v\sim F}\Big[-\sum_{i\in N} p_i(v)\Big] \leq -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i=1}^{n} p_i(v^{(\ell)}) + 2\cdot\inf_{\epsilon>0}\Big\{\epsilon + 2n\sqrt{\frac{2\log(\mathcal{N}_\infty(\mathcal{M},\,\epsilon))}{L}}\Big\} + Cn\sqrt{\frac{\log(1/\delta)}{L}}.$$

This completes the proof for the first part. □

We move to the second part, namely a generalization bound for regret, which is the more challenging part of the proof.

Proof of Theorem 1 (Part 2). We first define the class of sum regret functions:

$$\overline{\mathrm{rgt}} \circ \mathcal{U} = \left\{ f : V \to \mathbb{R} \;\middle|\; f(v) \;=\; \sum_{i=1}^{n} r_i(v) \text{ for some } (r_1, \ldots, r_n) \in \mathrm{rgt} \circ \mathcal{U} \right\}.$$

The proof then proceeds in three steps:

(1) bounding the covering number for each regret class $\mathrm{rgt} \circ \mathcal{U}_i$ in terms of the covering number for individual utility classes $\mathcal{U}_i$,

(2) bounding the covering number for the combined utility class $\mathcal{U}$ in terms of the covering number for $\mathcal{M}$, and

(3) bounding the covering number for the sum regret class $\overline{\mathrm{rgt}} \circ \mathcal{U}$ in terms of the covering number for the (combined) utility class $\mathcal{M}$.

An application of Lemma 3 then completes the proof. We prove each of the above steps below.

Step 1. $\mathcal{N}_\infty(\mathrm{rgt} \circ \mathcal{U}_i, \epsilon) \leq \mathcal{N}_\infty(\mathcal{U}_i, \epsilon/2)$.

By definition of covering number $\mathcal{N}_\infty(\mathcal{U}_i, \epsilon)$, there exists $\hat{\mathcal{U}}_i$ with size at most $\mathcal{N}_\infty(\mathcal{U}_i, \epsilon/2)$ such that for any $u_i \in \mathcal{U}_i$, there exists a $\hat{u}_i \in \hat{\mathcal{U}}_i$ with

$$\sup_{v, v_i'} |u_i(v_i, (v_i', v_{-i})) - \hat{u}_i(v_i, (v_i', v_{-i}))| \leq \epsilon/2.$$

For any $u_i \in \mathcal{U}_i$, taking $\hat{u}_i \in \hat{\mathcal{U}}_i$ satisfying the above condition, then for any v,

$$\begin{aligned}
&\left|\max_{v_i'\in V}\big(u_i(v_i,(v_i',v_{-i})) - u_i(v_i,(v_i,v_{-i}))\big) - \max_{\bar{v}_i\in V}\big(\hat{u}_i(v_i,(\bar{v}_i,v_{-i})) - \hat{u}_i(v_i,(v_i,v_{-i}))\big)\right| \\
\leq\; & \left|\max_{v_i'} u_i(v_i,(v_i',v_{-i})) - \max_{\bar{v}_i}\hat{u}_i(v_i,(\bar{v}_i,v_{-i})) + \hat{u}_i(v_i,(v_i,v_{-i})) - u_i(v_i,(v_i,v_{-i}))\right| \\
\leq\; & \left|\max_{v_i'} u_i(v_i,(v_i',v_{-i})) - \max_{\bar{v}_i}\hat{u}_i(v_i,(\bar{v}_i,v_{-i}))\right| + |\hat{u}_i(v_i,(v_i,v_{-i})) - u_i(v_i,(v_i,v_{-i}))| \\
\leq\; & \left|\max_{v_i'} u_i(v_i,(v_i',v_{-i})) - \max_{\bar{v}_i}\hat{u}_i(v_i,(\bar{v}_i,v_{-i}))\right| + \epsilon/2.
\end{aligned}$$

Let $v_i^* \in \arg\max_{v_i'} u_i(v_i, (v_i', v_{-i}))$ and $\hat{v}_i^* \in \arg\max_{\bar{v}_i} \hat{u}_i(v_i, (\bar{v}_i, v_{-i}))$, then

$$\begin{aligned}
&\max_{v_i'} u_i(v_i,(v_i',v_{-i})) = u_i(v_i^*,v_{-i}) \leq \hat{u}_i(v_i^*,v_{-i}) + \epsilon/2 \leq \hat{u}_i(\hat{v}_i^*,v_{-i}) + \epsilon/2 = \max_{\bar{v}_i}\hat{u}_i(v_i,(\bar{v}_i,v_{-i})) + \epsilon/2, \text{ and} \\
&\max_{\bar{v}_i}\hat{u}_i(v_i,(\bar{v}_i,v_{-i})) = \hat{u}_i(\hat{v}_i^*,v_{-i}) \leq u_i(\hat{v}_i^*,v_{-i}) + \epsilon/2 \leq u_i(v_i^*,v_{-i}) + \epsilon/2 = \max_{v_i'} u_i(v_i,(v_i',v_{-i})) + \epsilon/2.
\end{aligned} \tag{6}$$

Thus, for all $u_i \in \mathcal{U}_i$, there exists $\hat{u}_i \in \hat{\mathcal{U}}_i$ such that for any valuation profile v,

$$\left|\max_{v_i'}\big(u_i(v_i,(v_i',v_{-i})) - u_i(v_i,(v_i,v_{-i}))\big) - \max_{\bar{v}_i}\big(\hat{u}_i(v_i,(\bar{v}_i,v_{-i})) - \hat{u}_i(v_i,(v_i,v_{-i}))\big)\right| \leq \epsilon,$$

which implies $\mathcal{N}_\infty(\mathrm{rgt} \circ \mathcal{U}_i, \epsilon) \leq \mathcal{N}_\infty(\mathcal{U}_i, \epsilon/2)$.

This completes the proof for Step 1.

Step 2. $\mathcal{N}_\infty(\mathcal{U}, \epsilon) \le \mathcal{N}_\infty(\mathcal{M}, \epsilon)$, for all $i \in N$.

Recall the utility function of bidder i is $u_i(v_i, (v'_i, v_{-i})) = \langle v_i, g_i(v'_i, v_{-i})\rangle - p_i(v'_i, v_{-i})$. There exists a set $\hat{\mathcal{M}}$ with $|\hat{\mathcal{M}}| \le \mathcal{N}_\infty(\mathcal{M}, \epsilon)$ such that there exists $(\hat{g}, \hat{p}) \in \hat{M}$ with

$$\sup_{v\in V} \sum_{i,j} |g_{ij}(v) - \hat{g}_{ij}(v)| + \|p(v) - \hat{p}(v)\|_1 \le \epsilon.$$

We denote $\hat{u}_i(v_i, (v'_i, v_{-i})) = \langle v_i, \hat{g}_i(v'_i, v_{-i})\rangle - \hat{p}_i(v'_i, v_{-i})$, where we treat v_i as a real-valued vector of length 2^M.

For all $v \in V, v'_i \in V_i$,

$$\begin{aligned}
&|u_i(v_i, (v'_i, v_{-i})) - \hat{u}_i(v_i, (v'_i, v_{-i}))| \\
&\le \; |\langle v_i, g_i(v'_i, v_{-i})\rangle - \langle v_i, \hat{g}_i(v'_i, v_{-i})\rangle| + |p_i(v'_i, v_{-i}) - \hat{p}_i(v'_i, v_{-i})| \\
&\le \; \|v_i\|_\infty \cdot \|g_i(v'_i, v_{-i}) - \hat{g}_i(v'_i, v_{-i})\|_1 + |p_i(v'_i, v_{-i}) - \hat{p}_i(v'_i, v_{-i})| \\
&\le \; \sum_j |g_{ij}(v'_i, v_{-i}) - \hat{g}_{ij}(v'_i, v_{-i})| + |p_i(v'_i, v_{-i}) - \hat{p}_i(v'_i, v_{-i})| \,.
\end{aligned}$$

Therefore, for any $u \in \mathcal{U}$, take $\hat{u} = (\hat{g}, \hat{p}) \in \hat{\mathcal{M}}$, for all v, v',

$$\begin{aligned}
&\sum_i |u_i(v_i, (v'_i, v_{-i})) - \hat{u}_i(v_i, (v'_i, v_{-i}))| \\
&\qquad \le \sum_{ij} |g_{ij}(v'_i, v_{-i}) - \hat{g}_{ij}(v'_i, v_{-i})| + \sum_i |p_i(v'_i, v_{-i}) - \hat{p}_i(v'_i, v_{-i})| \le \epsilon.
\end{aligned}$$

This completes the proof for Step 2.

Step 3. $\mathcal{N}_\infty(\overline{\text{rgt}} \circ \mathcal{U}, \epsilon) \le \mathcal{N}_\infty(\mathcal{M}, \epsilon/2)$

By definition of $\mathcal{N}_\infty(\mathcal{U}, \epsilon)$, there exists $\hat{\mathcal{U}}$ with size at most $\mathcal{N}_\infty(\mathcal{U}, \epsilon)$, such that, for any $u \in \mathcal{U}$, there exists $\hat{u}$ s.t. for all $v, v' \in V$, $\sum_i |u_i(v_i, (v'_i, v_{-i})) - \hat{u}_i(v_i, (v'_i, v_{-i}))| \le \epsilon$. Therefore for all $v \in V$, $|\sum_i u_i(v_i, (v'_i, v_{-i})) - \sum_i \hat{u}_i(v_i, (v'_i, v_{-i}))| \le \epsilon$, from which it follows that $\mathcal{N}_\infty(\overline{\text{rgt}} \circ \mathcal{U}, \epsilon) \le \mathcal{N}_\infty(\text{rgt} \circ \mathcal{U}, \epsilon)$. Following Step 1, it is easy to show $\mathcal{N}_\infty(\text{rgt} \circ \mathcal{U}, \epsilon) \le \mathcal{N}_\infty(\mathcal{U}, \epsilon/2)$. This further with Step 2 completes the proof of Step 3.

Based on the same arguments as in the proof of Theorem 1 (Part 1) the empirical Rademacher complexity is bounded as:

$$\hat{\mathcal{R}}_L(\overline{\text{rgt}} \circ \mathcal{U}) \le \inf_{\epsilon > 0} \left(\epsilon + 2n\sqrt{\frac{2 \log \mathcal{N}_\infty(\overline{\text{rgt}} \circ \mathcal{U}, \epsilon)}{L}} \right) \le \inf_{\epsilon > 0} \left(\epsilon + 2n\sqrt{\frac{2 \log \mathcal{N}_\infty(\mathcal{M}, \epsilon/2)}{L}} \right).$$

Applying Lemma 3, completes the proof for generalization bound for regret. □

A.3. Proof of Theorem 2

We first bound the covering number for a general feed-forward neural network and specialize it to the three architectures we present in Section 3.

Lemma 4. *Let $\mathcal{F}_k$ be a class of feed-forward neural networks that maps an input vector $x \in \mathbb{R}^{d_0}$ to an output vector $y \in \mathbb{R}^{d_k}$, with each layer ℓ containing T_ℓ nodes and computing $z \mapsto \phi_\ell(w^\ell z)$, where each $w^\ell \in \mathbb{R}^{T_\ell \times T_{\ell-1}}$ and $\phi_\ell : \mathbb{R}^{T_\ell} \to [-B, +B]^{T_\ell}$. Further let, for each network in $\mathcal{F}_k$, let the parameter matrices $\|w^\ell\|_1 \le W$ and $\|\phi_\ell(s) - \phi_\ell(s')\|_1 \le \Phi\|s - s'\|_1$ for any $s, s' \in \mathbb{R}^{T_{\ell-1}}$.*

$$\mathcal{N}_\infty(\mathcal{F}_k, \epsilon) \le \left\lceil \frac{2Bd^2W(2\Phi W)^k}{\epsilon} \right\rceil^d,$$

where $T = \max_{\ell\in[k]} T_\ell$ and d is the total number of parameters in a network.

Proof. We shall construct an $\ell_{1,\infty}$ cover for $\mathcal{F}_k$ by discretizing each of the d parameters along $[-W, +W]$ at scale ϵ_0/d, where we will choose $\epsilon_0 > 0$ at the end of the proof. We will use $\hat{\mathcal{F}}_k$ to denote the subset of neural networks in $\mathcal{F}_k$ whose parameters are in the range $\{-(\lceil Wd/\epsilon_0 \rceil - 1)\,\epsilon_0/d, \ldots, -\epsilon_0/d, 0, \epsilon_0/d, \ldots, \lceil Wd/\epsilon_0 \rceil \epsilon_0/d\}$. Note that size of $\hat{\mathcal{F}}_k$ is at most $\lceil 2dW/\epsilon_0 \rceil^d$. We shall now show that $\hat{\mathcal{F}}_k$ is an ϵ-cover for $\mathcal{F}_k$.

We use mathematical induction on the number of layers k. We wish to show that for any $f \in \mathcal{F}_k$ there exists a $\hat{f} \in \hat{\mathcal{F}}_k$ such that:

$$\|f(x) - \hat{f}(x)\|_1 \leq Bd\epsilon_0(2\Phi W)^k.$$

Note that for $k = 0$, the statement holds trivially. Assume that the statement is true for $\mathcal{F}_k$. We now show that the statement holds for $\mathcal{F}_{k+1}$.

A function $f \in \mathcal{F}_{k+1}$ can be written as $f(z) = \phi_{k+1}(w_{k+1}H(z))$ for some $H \in \mathcal{F}_k$. Similarly, a function $\hat{f} \in \hat{\mathcal{F}}_{k+1}$ can be written as $\hat{f}(z) = \phi_{k+1}(\hat{w}_{k+1}\hat{H}(z))$ for some $\hat{H} \in \hat{\mathcal{F}}_k$ and $\hat{w}_{k+1}$ is a matrix of entries in $\{-(\lceil Wd/\epsilon_0 \rceil - 1)\,\epsilon_0/d, \ldots, -\epsilon_0/d, 0, \epsilon_0/d, \ldots, \lceil Wd/\epsilon_0 \rceil \epsilon_0/d\}$. Also note that for any parameter matrix $w^\ell \in \mathbb{R}^{T_\ell \times T_{\ell-1}}$, there is a matrix $\hat{w}^\ell$ with discrete entries s.t.

$$\|w_\ell - \hat{w}_\ell\|_1 = \max_{1 \leq j \leq T_{\ell-1}} \sum_{i=1}^{T_\ell} |w^\ell_{\ell,i,j} - \hat{w}_{\ell,i,j}| \leq T_\ell \epsilon_0/d \leq \epsilon_0. \tag{7}$$

We then have:

$$\begin{aligned}
\|f(x) - \hat{f}(x)\|_1 &= \|\phi_{k+1}(w_{k+1}H(x)) - \phi_{k+1}(\hat{w}_{k+1}\hat{H}(x))\|_1 \\
&\leq \Phi\|w_{k+1}H(x) - \hat{w}_{k+1}\hat{H}(x)\|_1 \\
&\leq \Phi\|w_{k+1}H(x) - w_{k+1}\hat{H}(x)\|_1 + \Phi\|w_{k+1}\hat{H}(x) - \hat{w}_{k+1}\hat{H}(x)\|_1 \\
&\leq \Phi\|w_{k+1}\|_1 \cdot \|H(x) - \hat{H}(x)\|_1 + \Phi\|w_{k+1} - \hat{w}_{k+1}\|_1 \cdot \|\hat{H}(x)\|_1 \\
&\leq \Phi W\|H(x) - \hat{H}(x)\|_1 + \Phi B\|w_{k+1} - \hat{w}_{k+1}\|_1 \\
&\leq Bd\epsilon_0 \Phi W(2\Phi W)^k + \Phi B d\epsilon_0 \\
&\leq Bd\epsilon_0(2\Phi W)^{k+1},
\end{aligned}$$

where the second line follows from our assumption on ϕ_{k+1}, and the sixth line follows from our inductive hypothesis and from (7). By choosing $\epsilon_0 = \frac{\epsilon}{B(2\Phi W)^k}$, we complete the proof. □

We next bound the covering number of the mechanism class in terms of the covering number for the class of allocation networks and for the class of payment networks. Recall that the payment networks computes a fraction $\alpha : \mathbb{R}^{m(n+1)} \to [0,1]^n$ and computes a payment $p_i(b) = \alpha_i(b) \cdot \langle v_i, g_i(b) \rangle$ for each bidder i. Let $\mathcal{G}$ be the class of allocation networks and $\mathcal{A}$ be the class of fractional payment functions used to construct auctions in $\mathcal{M}$. Let $\mathcal{N}_\infty(\mathcal{G}, \epsilon)$ and $\mathcal{N}_\infty(\mathcal{A}, \epsilon)$ be the corresponding covering numbers w.r.t. the ℓ_∞ norm. Then:

Lemma 5. $\mathcal{N}_\infty(\mathcal{M}, \epsilon) \leq \mathcal{N}_\infty(\mathcal{G}, \epsilon/3) \cdot \mathcal{N}_\infty(\mathcal{A}, \epsilon/3)$.

Proof. Let $\hat{\mathcal{G}} \subseteq \mathcal{G}$, $\hat{\mathcal{A}} \subseteq \mathcal{A}$ be ℓ_∞ covers for $\mathcal{G}$ and $\mathcal{A}$, i.e. for any $g \in \mathcal{G}$ and $\alpha \in \mathcal{A}$, there exists $\hat{g} \in \hat{\mathcal{G}}$ and $\hat{\alpha} \in \hat{\mathcal{A}}$ with

$$\sup_b \sum_{i,j} |g_{ij}(b) - \hat{g}_{ij}(b)| \leq \epsilon/3, \text{and} \tag{8}$$

$$\sup_b \sum_i |\alpha_i(b) - \hat{\alpha}_i(b)| \leq \epsilon/3. \tag{9}$$

We now show that the class of mechanism $\hat{\mathcal{M}} = \{(\hat{g}, \hat{\alpha}) \,|\, \hat{g} \in \hat{\mathcal{G}}, \text{ and } \hat{p}(b) = \hat{\alpha}_i(b) \cdot \langle v_i, \hat{g}_i(b) \rangle\}$ is an ϵ-cover for $\mathcal{M}$ under the $\ell_{1,\infty}$ distance. For any mechanism in $(g, p) \in \mathcal{M}$, let $(\hat{g}, \hat{p}) \in \hat{\mathcal{M}}$ be a mechanism in $\hat{\mathcal{M}}$ that satisfies (9). We have:

$$\sum_{i,j} |g_{ij}(b) - \hat{g}_{ij}(b)| + \sum_i |p_i(b) - \hat{p}_i(b)|$$

$$\leq \epsilon/3 + \sum_i |\alpha_i(b) \cdot \langle b_i, g_{i,\cdot}(b)\rangle - \hat{\alpha}_i(b) \cdot \langle b_i, \hat{g}_i(b)\rangle|$$

$$\leq \epsilon/3 + \sum_i |(\alpha_i(b) - \hat{\alpha}_i(b)) \cdot \langle b_i, g_i(b)\rangle| + |\hat{\alpha}_i(b) \cdot (\langle b_i, g_i(b)\rangle - \langle b_i, \hat{g}_{i,\cdot}(b)\rangle)|$$

$$\leq \epsilon/3 + \sum_i |\alpha_i(b) - \hat{\alpha}_i(b)| + \sum_i \|b_i\|_\infty \cdot \|g_i(b) - \hat{g}_i(b)\|_1$$

$$\leq 2\epsilon/3 + \sum_{i,j} |g_{ij}(b) - \hat{g}_{ij}(b)| \leq \epsilon,$$

where in the third inequality we use $\langle b_i, g_i(b)\rangle \leq 1$. The size of the cover $\hat{\mathcal{M}}$ is $|\hat{\mathcal{G}}||\hat{\mathcal{A}}|$, which completes the proof. □

We are now ready to prove covering number bounds for the three architectures in Section 3.

Proof of Theorem 2. All three architectures use the same feed-forward architecture for computing fractional payments, consisting of K hidden layers with tanh activation functions. We also have by our assumption that the L_1 norm of the vector of all model parameters is at most W, for each $\ell = 1, \ldots, R+1$, $\|w_\ell\|_1 \leq W$. Using that fact that the tanh activation functions are 1-Lipschitz and bounded in $[-1, 1]$, and there are at most $\max\{K, n\}$ number of nodes in any layer of the payment network, we have by an application of Lemma 4 the following bound on the covering number of the fractional payment networks $\mathcal{A}$ used in each case:

$$\mathcal{N}_\infty(\mathcal{A}, \epsilon) \leq \left\lceil \frac{\max(K, n)^2 (2W)^{R+1}}{\epsilon} \right\rceil^{d_p},$$

where d_p is the number of parameters in payment networks.

For the covering number of allocation networks $\mathcal{G}$, we consider each architecture separately. In each case, we bound the Lipschitz constant for the activation functions used in the layers of the allocation network and followed by an application of Lemma 4. For ease of exposition, we omit the dummy scores used in the final layer of neural network architectures.

Additive bidders. The output layer computes n allocation probabilities for each item j using a softmax function. The activation function $\phi_{R+1} : \mathbb{R}^n \to \mathbb{R}^n$ for the final layer for input $s \in \mathbb{R}^{n \times m}$ can be described as: $\phi_{R+1}(s) = [\text{softmax}(s_{1,1}, \ldots, s_{n,1}), \ldots, \text{softmax}(s_{1,m}, \ldots, s_{n,m})]$, where $\text{softmax} : \mathbb{R}^n \to [0,1]^n$ is defined for any $u \in \mathbb{R}^n$ as $\text{softmax}_i(u) = \exp(u_i)/\sum_{k=1}^n \exp(u_k)$.

We then have for any $s, s' \in \mathbb{R}^{n \times m}$,

$$\begin{aligned}
\|\phi_{R+1}(s) - \phi_{R+1}(s')\|_1 &= \sum_j \left\|\text{softmax}(s_{1,j}, \ldots, s_{n,j}) - \text{softmax}(s'_{1,j}, \ldots, s'_{n,j})\right\|_1 \\
&\leq \sqrt{n} \sum_j \left\|\text{softmax}(s_{1,j}, \ldots, s_{n,j}) - \text{softmax}(s'_{1,j}, \ldots, s'_{n,j})\right\|_2 \\
&\leq \sqrt{n} \frac{\sqrt{n-1}}{n} \sum_j \sqrt{\sum_i \|s_{ij} - s'_{ij}\|^2} \\
&\leq \sum_j \sum_i |s_{ij} - s'_{ij}|, \qquad (10)
\end{aligned}$$

where the third step follows by bounding the Frobenius norm of the Jacobian of the softmax function.

The hidden layers $\ell = 1, \ldots, R$ are standard feed-forward layers with tanh activations. Since the tanh activation function is 1-Lipschitz, $\|\phi_\ell(s) - \phi_\ell(s')\|_1 \leq \|s - s'\|_1$. We also have by our assumption that the L_1 norm of the vector of all model parameters is at most W, for each $\ell = 1, \ldots, R+1$, $\|w_\ell\|_1 \leq W$. Moreover, the output of each hidden layer node is in $[-1, 1]$, the output layer nodes is in $[0, 1]$, and the maximum number of nodes in any layer (including the output layer) is at most $\max\{K, mn\}$.

By an application of Lemma 4 with $\Phi = 1$, $B = 1$ and $d = \max K, mn$, we have

$$\mathcal{N}_\infty(\mathcal{G}, \epsilon) \leq \left\lceil \frac{\max\{K, mn\}^2 (2W)^{R+1}}{\epsilon} \right\rceil^{d_a},$$

where d_a is the number of parameters in allocation networks.

Unit-demand bidders. The output layer n allocation probabilities for each item j as an element-wise minimum of two softmax functions. The activation function $\phi_{R+1} : \mathbb{R}^2 n \to \mathbb{R}^n$ for the final layer for two sets of scores $s, \bar{s} \in \mathbb{R}^{n \times m}$ can be described as:

$$\phi_{R+1,i,j}(s, s') = \min\{\text{softmax}_j(s_{i,1}, \ldots, s_{i,m}),\, \text{softmax}_i(s'_{1,j}, \ldots, s'_{n,j})\}.$$

We then have for any $s, \tilde{s}, s', \tilde{s}' \in \mathbb{R}^{n \times m}$,

$$\begin{aligned}
\|\phi_{R+1}(s, \tilde{s}) - \phi_{R+1}(s', \tilde{s}')\|_1 &= \sum_{i,j} \Big| \min\{\text{softmax}_j(s_{i,1}, \ldots, s_{i,m}),\, \text{softmax}_i(\tilde{s}_{1,j}, \ldots, \tilde{s}_{n,j})\} \\
&\qquad - \min\{\text{softmax}_j(s'_{i,1}, \ldots, s'_{i,m}),\, \text{softmax}_i(\tilde{s}'_{1,j}, \ldots, \tilde{s}'_{n,j})\} \Big| \\
&\le \sum_{i,j} \Big| \max\{\text{softmax}_j(s_{i,1}, \ldots, s_{i,m}) - \text{softmax}_j(s'_{i,1}, \ldots, s'_{i,m}), \\
&\qquad \text{softmax}_i(\tilde{s}_{1,j}, \ldots, \tilde{s}_{n,j}) - \text{softmax}_i(\tilde{s}'_{1,j}, \ldots, \tilde{s}'_{n,j})\} \Big| \\
&\le \sum_{i} \big\| \text{softmax}(s_{i,1}, \ldots, s_{i,m}) - \text{softmax}(s'_{i,1}, \ldots, s'_{i,m}) \big\|_1 \\
&\qquad + \sum_{j} \big\| \text{softmax}(\tilde{s}_{1,j}, \ldots, \tilde{s}_{n,j}) - \text{softmax}(\tilde{s}'_{1,j}, \ldots, \tilde{s}'_{n,j})\} \big\|_1 \\
&\le \sum_{i,j} |s_{ij} - s'_{ij}| + \sum_{i,j} |\tilde{s}_{ij} - \tilde{s}'_{ij}|,
\end{aligned}$$

where the last step can be derived in the same way as (10).

As with additive bidders, using additionally hidden layers $\ell = 1, \ldots, R$ are standard feed-forward layers with tanh activations, we have from Lemma 4 with $\Phi = 1$, $B = 1$ and $d = \max\{K, mn\}$,

$$\mathcal{N}_\infty(\mathcal{G}, \epsilon) \le \left\lceil \frac{\max\{K, mn\}^2 (2W)^{R+1}}{\epsilon} \right\rceil^{d_a}.$$

Combinatorial bidders. The output layer outputs an allocation probability for each bidder i and bundle of items $S \subseteq M$. The activation function $\phi_{R+1} : \mathbb{R}^{(m+1)n2^m} \to \mathbb{R}^{n2^m}$ for this layer for $m+1$ sets of scores $s, s^{(1)}, \ldots, s^{(m)} \in \mathbb{R}^{n \times 2^m}$ is given by:

$$\begin{aligned}
&\phi_{R+1,i,S}(s, s^{(1)}, \ldots, s^{(m)}) \\
&\quad = \min\Big\{\text{softmax}_S(s_{i,S'} : S' \subseteq M),\, \text{softmax}_S(s^{(1)}_{i,S'} : S' \subseteq M),\, \ldots, \text{softmax}_S(s^{(m)}_{i,S'} : S' \subseteq M)\Big\},
\end{aligned}$$

where $\text{softmax}_S(a_{S'} : S' \subseteq M) = \exp(a_S) / \sum_{S' \subseteq M} \exp(a_{S'})$.

We then have for any $s, s^{(1)}, \ldots, s^{(m)}, s', s'^{(1)}, \ldots, s'^{(m)} \in \mathbb{R}^{n \times 2^m}$,

$$\begin{aligned}
&\|\phi_{R+1}(s, s^{(1)}, \ldots, s^{(m)}) - \phi_{R+1}(s', s'^{(1)}, \ldots, s'^{(m)})\|_1 \\
&= \sum_{i,S} \Big| \min\Big\{\text{softmax}_S(s_{i,S'} : S' \subseteq M),\, \text{softmax}_S(s^{(1)}_{i,S'} : S' \subseteq M),\, \ldots, \text{softmax}_S(s^{(m)}_{i,S'} : S' \subseteq M)\Big\} \\
&\quad - \min\Big\{\text{softmax}_S(s'_{i,S'} : S' \subseteq M),\, \text{softmax}_S(s'^{(1)}_{i,S'} : S' \subseteq M),\, \ldots, \text{softmax}_S(s'^{(m)}_{i,S'} : S' \subseteq M)\Big\} \Big| \\
&\le \sum_{i,S} \max\Big\{ \big|\text{softmax}_S(s_{i,S'} : S' \subseteq M) - \text{softmax}_S(s'_{i,S'} : S' \subseteq M)\big|, \\
&\qquad \big|\text{softmax}_S(s^{(1)}_{i,S'} : S' \subseteq M) - \text{softmax}_S(s'^{(1)}_{i,S'} : S' \subseteq M)\big|, \ldots \\
&\qquad \big|\text{softmax}_S(s^{(m)}_{i,S'} : S' \subseteq M) - \text{softmax}_S(s'^{(m)}_{i,S'} : S' \subseteq M)\big| \Big\} \\
&\le \sum_{i} \big\|\text{softmax}(s_{i,S'} : S' \subseteq M) - \text{softmax}(s'_{i,S'} : S' \subseteq M)\big\|_1 \\
&\qquad + \sum_{i,j} \big\|\text{softmax}(s^{(j)}_{i,S'} : S' \subseteq M) - \text{softmax}(s'^{(j)}_{i,S'} : S' \subseteq M)\big\|_1
\end{aligned}$$

Distretization	Number of decision variables	Number of constriants
5 bins/value	1.25×10^5	3.91×10^6
6 bins/value	3.73×10^5	2.02×10^7
7 bins/value	9.41×10^5	8.07×10^7

Table 2: Number of decision variables and constraints of LP with different discretizations for a 2 bidder, 3 items setting with uniform valuations.

$$\leq \sum_{i,S} |s_{i,S} - s'_{i,S}| + \sum_{i,j,S} |s^{(j)}_{i,S} - s'^{(j)}_{i,S}|,$$

where the last step can be derived in the same way as (10).

As with additive bidders, using additionally hidden layers $\ell = 1, \ldots, R$ are standard feed-forward layers with tanh activations, we have from Lemma 4 with $\Phi = 1$, $B = 1$ and $d = \max\{K, n \cdot 2^m\}$

$$\mathcal{N}_\infty(\mathcal{G}, \epsilon) \leq \left\lceil \frac{\max\{K, n \cdot 2^m\}^2 (2W)^{R+1}}{\epsilon} \right\rceil^{d_a},$$

where d_a is the number of parameters in allocation networks. □

We now bound Δ_L for the three architectures using the covering number bounds we derived above. In particular, we upper bound the the 'inf' over $\epsilon > 0$ by substituting a specific value of ϵ:

(a) For additive bidders, choosing $\epsilon = \frac{1}{\sqrt{L}}$, we get $\Delta_L \leq O\left(\sqrt{R(d_p + d_a)\frac{\log(W \max\{K, mn\} L)}{L}}\right)$.

(b) For unit-demand bidders, choosing $\epsilon = \frac{1}{\sqrt{L}}$, we get $\Delta_L \leq O\left(\sqrt{R(d_p + d_a)\frac{\log((W \max\{K, mn\} L)}{L}}\right)$.

(c) For combinatorial bidders, choosing $\epsilon = \frac{1}{\sqrt{L}}$, we get $\Delta_L \leq O\left(\sqrt{R(d_p + d_a)\frac{\log(W \max\{K, n \cdot 2^m\} L)}{L}}\right)$.

B. Omitted Details in Experiments

In this section, we show more details of the experiments in this paper.

Discussion on size of LP. First, we provide more evidence about the efficiency of our RegretNet compared with LP. As mentioned in (Conitzer & Sandholm, 2002), the number of decision variables and constraints are exponential in the number of bidders and items. We consider the setting with n additive bidders and m items and the value is divided into D bins per item. There are D^{mn} valuation profiles in total, each involving $(n + nm)$ variables (n payments and nm allocation probabilities). For the constraints, there are n IR constraints (for n bidders) and $n \cdot \big(D^m - 1\big)$ IC constraints (for each bidder, there are $\big(D^m - 1\big)$ constraints) for each valuation profile. In addition, there are n bidder-wise and m item-wise allocation constraints. In Table 2, we show the explosion of decision variables and constraints with finer discretization of the valuations for 2 bidders, 3 items setting. As we can see, the decision variables and constraints blow up extremely fast, even for a small setting with a coarse discretization over value.

Additional discussion of experiments. For small settings (I)–(V), we get similar performance as in Figure 3 with smaller training samples (around 5000). ReLU activations yield comparable results for smaller settings (I)–(V), but tanh works better for larger settings (VI)–(VII). Our RegretNet is scalable for auctions with more bidders and items. A single iteration of augmented Lagrangian took on an average 1–17 seconds across experiments. Even for the larger settings (VI)–(VII), the running time of our algorithm was less than 13 hours. For the settings (VI)–(VII) for which the optimal auction is *not* known, we also compare with a Myerson auction to sell the entire bundle of items as one unit, which is optimal in the limit of number of items (Palfrey, 1983).

Distribution	Opt	RegretNet	
	rev	rev	rgt
Setting (a): $v_1 \sim [4, 16], v_2 \sim U[4, 7]$	9.781	9.734	< 0.001
Setting (b): v_1, v_2 drawn uniformly from a unit triangle	0.388	0.392	< 0.001
Setting (c): $v_1, v_2 \sim U[0, 1]$	0.384	0.384	< 0.001

Table 3: Revenue of auctions for single additive bidder, two items obtained with RegretNet.

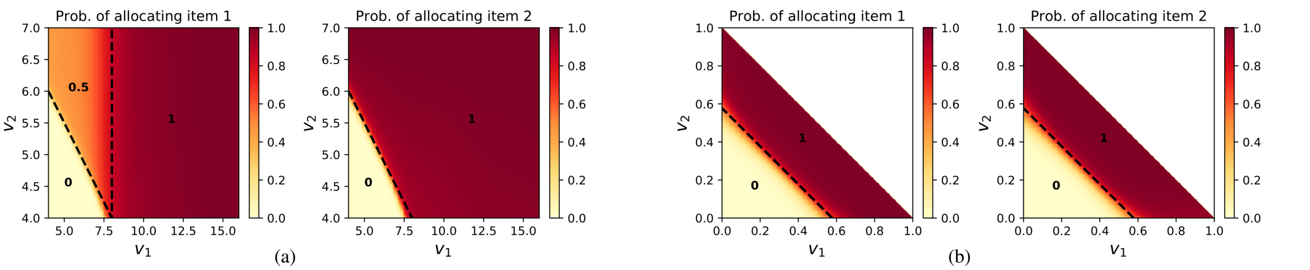


Figure 6: Allocation rule learned by RegretNet for (a) the single additive bidder, two items setting with values $v_1 \sim U[4, 16]$ and $v_2 \sim U[4, 7]$, and for (b) the single additive bidder, two items setting with values v_1, v_2 drawn jointly, uniformly from a triangle with vertices $(0, 0)$, $(0, 1)$ and $(1, 0)$, The optimal mechanisms due to (Daskalakis et al., 2017) for (a) and (Haghpanah & Hartline, 2015) for (b) are described by the regions separated by the dashed orange lines. The numbers in orange are the probability the item is allocated in a region.

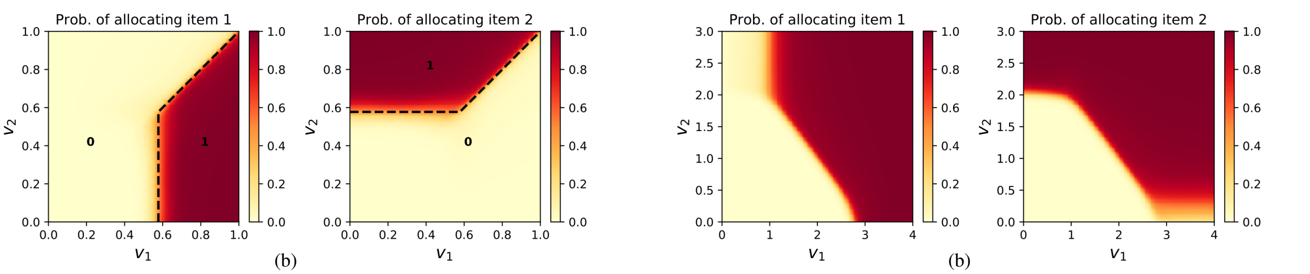


Figure 7: Allocation rule learned by RegretNet for (a) the single unit-demand bidder, two items setting with values $v_1, v_2 \sim U[0, 1]$ (optimal mechanism due to (Pavlov, 2011)), and for (b) the single additive bidder, two items setting with values $v_1 \sim U[0, 4], v_2 \sim U[0, 3]$. The subset of valuations (v_1, v_2) where the bidder receives neither item looks like a pentagonal shape.

Distribution	Item-wise Myerson	Bundled Myerson	RegretNet	
	rev	rev	rev	rgt
Setting (d): $v_i \sim U[0, 1]$	2.495	3.457	**3.461**	< 0.003
Setting (e): $v_1 \sim U[0, 4], v_2 \sim U[0, 3]$	1.877	1.749	**1.911**	< 0.001

Table 4: Revenue of auctions for single additive bidder, 10 items obtained with RegretNet and single additive bidder, 2 items with $v_1 \sim U[0, 4], v_2 \sim U[0, 3]$.

Distribution	Ascending auction	RegretNet	
	rev	rev	rgt
Setting (f): $v_1, v_2 \sim U[0, 1]$	0.179	**0.706**	< 0.001

Table 5: Revenue of auctions for 2 unit-demand bidders, 2 items obtained with RegretNet. For the ascending auction, the price were raised in units of 0.3 (which was empirically tuned using a grid search.)

C. Additional Experiments

In this section, we show the additional experiments for both the single bidder case and the mulitple bidders case. We consider the following settings:

(a) Single additive bidder with preferences over two non-identically distributed items, where $v_1 \sim U[4, 16]$ and $v_2 \sim U[4, 7]$.

(b) Single additive bidder with preferences over two items, where (v_1, v_2) are drawn jointly and uniformly from a unit triangle with vertices $(0, 0), (0, 1)$ and $(1, 0)$.

(c) Single unit-demand bidder with preferences over two items, where the item values $v_1, v_2 \sim U[0, 1]$,

(d) Single additive bidder with preferences over ten items, where each $v_i \sim U[0, 1]$.

(e) Single additive bidder with preferences over two items, where the item values $v_1 \sim U[0, 4], v_2 \sim U[0, 3]$,

(f) Two unit-demand bidders and two items, where the bidders draw their value for each item from identical uniform distributions over $[0, 1]$.

For setting (a), we show our *RegretNet* almost exactly recovers the optimal mechanism of (Daskalakis et al., 2017). For setting (b), we show that the approach almost exactly recovers the optimal mechanism of (Haghpanah & Hartline, 2015). For setting (c), we show that the approach almost exactly recovers the optimal mechanisms of (Pavlov, 2011). For settings (a), (b), (c), we show our results in Table 3, and we show the allocation plots for the three settings above in Figure 6 and Figure 7. To our knowledge, an analytical solution for the optimal mechanism for setting (d) is not available (Daskalakis, 2015). Here our approach finds a new mechanism that has higher revenue than both a Myerson auction on each item and a Myerson on the entire bundle, we show it in Table 4. For setting (e), we plot the allocation figures in Figure 7 and test the performance of our *RegretNet* compared with Myerson auction on each item and Myerson auction on the entire bundle in Table 4. For setting (f), the optimal auction is again not known; we show in Table 5 that the learned auctions beat reasonable baseline mechanisms.

D. Decomposition of Combinatorial Feasible Allocations

In Section 3, we defined a *combinatorial feasible* allocation. In this section, we show that the definition need not imply the existence of an integer decomposition and provide a stronger definition for the case of two items, a modified neural network architecture, and updated experimental results for settings (IV) and (V). The effect is a very slight reduction in the expected revenue from the optimized auction designs.

Definition 1. *A fractional combinatorial allocation z has an integer decomposition if and only if z can be represented as a convex combination of feasible, deterministic allocations.*

Example 1 shows that a combinatorial feasible allocation may not have an integer deccomposition, even for the case of two bidders and two items.

Example 1. *Consider a setting with two bidders and two items, and the following fractional, combinatorial feasible allocation:*

$$z = \begin{bmatrix} z_{1,\{1\}} & z_{1,\{2\}} & z_{1,\{1,2\}} \\ z_{2,\{1\}} & z_{2,\{2\}} & z_{2,\{1,2\}} \end{bmatrix} = \begin{bmatrix} 3/8 & 3/8 & 1/4 \\ 1/8 & 1/8 & 1/4 \end{bmatrix}$$

Any integer decomposition of this allocation z would need to have the following structure:

$$z = a \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} + b \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} + c \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + d \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + e \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + f \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} + g \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + h \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

where the coefficients sum to at most 1. Firstly, it is straightforward to see that $a = b = 1/4$. Given the construction, we must have $c + d = 3/8, e \geq 0$ and $f + g = 3/8, h \geq 0$. Thus, $a + b + c + d + e + f + g + h \geq 1/2 + 3/4 = 5/4$ for any decomposition. Hence, z is not implementable.

Dist.	*rev*	*rgt*	VVCA	AMA_{bsym}
(IV)	**2.860**	< 0.001	2.741	2.765
(V)	**4.269**	< 0.001	4.209	3.748

Figure 8: Modified test revenue and regret for the two bidder, two item combinatorial auction settings.

To ensure that a combinatorial feasible allocation has an integer decomposition we need to introduce additional constraints. For the two items case, we introduce the following constraint:

$$\forall i, z_{i,\{1\}} + z_{i,\{2\}} \leq 1 - \sum_{i'=1}^{n} z_{i',\{1,2\}}. \tag{11}$$

Theorem 3. *For $m = 2$, any combinatorial feasible allocation z with additional constraints (11) can be represented as a convex combination of matrices $B^1, \dots, B^k$ where each B^ℓ is a feasible, 0-1 allocation.*

Proof. Firstly, we observe in any deterministic allocation B^ℓ, if there exists an i, s.t. $B^\ell_{i,\{1,2\}} = 1$, then $\forall j \neq i, S : B^\ell_{j,S} = 0$. Therefore, we first decompose z into the following components,

$$z = \sum_{i=1}^{n} z_{i,\{1,2\}} \cdot B^i + C,$$

and

$$B^i_{j,S} = \begin{cases} 1 & \text{if } j = i, S = \{1,2\}, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

Then we want to argue that C can be represented as $\sum_{\ell=i+1}^{k} p_\ell \cdot B^\ell$, where $\sum_{\ell=i+1}^{k} p_\ell \leq 1 - \sum_{i=1}^{n} z_{i,\{1,2\}}$ and each B^ℓ is a feasible 0-1 allocation. Matrix C has all zeros in the last (items $\{1,2\}$) column, $\sum_i C_{i,\{1\}} \leq 1 - \sum_{i=1}^{n} z_{i,\{1,2\}}$, and $\sum_i C_{i,\{2\}} \leq 1 - \sum_{i=1}^{n} z_{i,\{1,2\}}$.

In addition, based on constraint (11), for each bidder i,

$$C_{i,\{1\}} + C_{i,\{2\}} = z_{i,\{1\}} + z_{i,\{2\}} \leq 1 - \sum_{i'=1}^{n} z_{i',\{1,2\}}.$$

Thus C is a doubly stochastic matrix with scaling factor $1 - \sum_{i'=1}^{n} z_{i',\{1,2\}}$. Therefore, we can always decompose C into a linear combination $\sum_{\ell=i+1}^{k} p_\ell \cdot B^\ell$, where $\sum_{\ell=i+1}^{k} p_\ell \leq 1 - \sum_{i'=1}^{n} z_{i',\{1,2\}}$ and each B^ℓ is a feasible 0-1 allocation. ☐

We leave to future work to characterize the additional constraints needed for the multi-item ($m > 2$) case.

D.1. Neural Network Architecture and Experimental Results

To accommodate the additional constraint (11) for the two items case we add an additional softmax layer for each bidder. In addition to the original (unnormalized) bidder-wise scores $s_{i,S}, \forall i \in N, S \subseteq M$ and item-wise scores $s^{(j)}_{i,S}, \forall i \in N, S \subseteq M, j \in M$ and their normalized counterparts $\bar{s}_{i,S}, \forall i \in N, S \subseteq M$ and $\bar{s}^{(j)}_{i,S}, \forall i \in N, S \subseteq M, j \in M$, the allocation network computes an additional set of scores for each bidder i, ${s'}^{(i)}_{i,\{1\}}, {s'}^{(i)}_{i,\{2\}}, {s'}^{(i)}_{1,\{1,2\}}, \cdots, {s'}^{(i)}_{n,\{1,2\}}$. These additional scores are then normalized using a softmax function as follows,

$$\forall i, k \in N, S \subseteq M, \quad \bar{s'}^{(i)}_{k,S} = \frac{\exp\left({s'}^{(i)}_{k,S}\right)}{\exp\left({s'}^{(i)}_{i,\{1\}}\right) + \exp\left({s'}^{(i)}_{i,\{2\}}\right) + \sum_k \exp\left({s'}^{(i)}_{k,\{1,2\}}\right)}.$$

To satisfy constraint (11) for each bidder i, we compute the normalized score $\bar{s'}_{i,S}$ for each i, S as,

$$\bar{s'}_{i,S} = \begin{cases} \bar{s'}^{(i)}_{i,S} & \text{if } S = \{1\} \text{ or } \{2\}, \text{ and} \\ \min\left\{\bar{s'}^{(k)}_{i,S} : k \in N\right\} & \text{if } S = \{1,2\}. \end{cases}$$

Then the final allocation for each bidder i is:

$$z_{i,S} = \min\left\{\bar{s}_{i,S}, \bar{s'}_{i,S}, \bar{s}^{(j)}_{i,S} : j \in S\right\}.$$

We repeat the experiments on the combinatorial auction settings (IV) and (V) with this modified architecture. We summarize the results of imposing this additional structure in Table 8. Compared with Figure 3(b) we see only a very small change in the expected revenue.