

# Optimal Auctions through Deep Learning\*

Paul Dütting<sup>1</sup>, Zhe Feng<sup>2</sup>, Harikrishna Narasimhan<sup>2</sup>, David C. Parkes<sup>2</sup>, and Sai Srivatsa Ravindranath<sup>2</sup>

<sup>1</sup>Department of Mathematics, London School of Economics  
p.d.duetting@lse.ac.uk

<sup>2</sup>John A. Paulson School of Engineering and Applied Sciences, Harvard University  
zhe\_feng,hnarasimhan,parkes,saisr@g.harvard.edu

March 3, 2019<sup>†</sup>

## 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 multi-item 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. Moreover, this can be done without appealing to *characterization results*, and even if the only feedback during training is revenue and regret. 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. We further show how characterization results, even rather implicit ones such as Rochet’s characterization through induced utilities and their gradients, can be leveraged to provide a tool to improve our theoretical understanding of the structure of optimal auctions.

## 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.

---

\*We would like to thank Yang Cai, Vincent Conitzer, Constantinos Daskalakis, Alexander Rush, participants in the Simons Institute Economics and Computation reunion workshop, the EC’18 WADE workshop, and the German Economic Association annual meeting, participants in seminars at Dagstuhl, LSE, Google, and HBS, and the anonymous reviewers on an earlier version of this paper for their helpful feedback.

<sup>†</sup>The first version of this paper was posted on arXiv on June 12, 2017. Preliminary versions were presented at the 3rd Workshop on Algorithmic Game Theory and Data Science and the NIPS’17 Workshop on Learning in the Presence of Strategic Behavior. Our code is open sourced in [Github](#).

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 [37]. 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 [33, 41, 25, 20, 16, 54], and an impressive sequence of recent algorithmic results [7, 6, 8, 26, 2, 53, 9, 10], most of them apply to the weaker notion of Bayesian incentive compatibility (BIC). Our focus is on designing auctions that satisfy dominant-strategy incentive compatible (DSIC), 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 and Roughgarden [11], Mohri and Medina [34]). A handful of works has leveraged machine learning to optimize different aspects of mechanisms [31, 18, 38], 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* [12, 13, 45], but these are limited to specialized classes of auctions known to be incentive compatible.

## 1.1 The Optimal Auction Design Problem

We consider a setting with a set of  $n$  bidders  $N = \{1, \dots, n\}$  and  $m$  items  $M = \{1, \dots, m\}$ . Each bidder  $i$  has a valuation function  $v_i : 2^M \rightarrow \mathbb{R}_{\geq 0}$ , 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  $v = (v_1, \dots, v_n)$  for a profile of valuations, and denote  $V = \prod_{i=1}^n V_i$ .

The auctioneer knows the distributions  $F = (F_1, \dots, F_n)$ , 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  $g_i : V \rightarrow 2^M$  and payment rules  $p_i : V \rightarrow \mathbb{R}_{\geq 0}$  (these rules can be randomized). Given bids  $b = (b_1, \dots, b_n) \in V$ , 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  $v = (v_1, \dots, v_n)$  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.  $u_i(v_i, (v_i, b_{-i})) \geq 0 \quad \forall i \in N, v_i \in V_i, \text{ and } b_{-i} \in V_{-i}$ .

In a DSIC and IR 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. It thus seeks to solve the following constrained optimization

problem:

$$\max_{(g,p)} \mathbf{E}_{v \sim F} \left[ \sum_i p_i(v) \right] \quad \text{s.t.} \quad (g,p) \in IC \quad (1)$$

where  $IC$  denotes the set of auctions that satisfy incentive compatibility. We refer the reader to a recent survey by [14] in regard to analytical solutions to this problem, and its relaxation where the DSIC constraint is replaced with the weaker BIC constraint, as well as for an overview of the many new challenges that arise in settings with multiple items.

## 1.2 Auction Design 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  $S = \{v^{(1)}, \dots, v^{(L)}\}$  drawn i.i.d. from  $F$ .<sup>1</sup> 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  $i$ :

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

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  $rgt_i(w) = 0, \forall i \in N$ .

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:

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

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

$$\widehat{rgt}_i(w) = \frac{1}{L} \sum_{\ell=1}^L \max_{v'_i \in V_i} u_i^w(v_i^{(\ell)}; (v'_i, v_{-i}^{(\ell)})) - u_i^w(v_i^{(\ell)}; v^{(\ell)}), \quad (2)$$

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

$$\begin{aligned} \min_{w \in \mathbb{R}^d} & -\frac{1}{L} \sum_{\ell=1}^L \sum_{i=1}^n p_i^w(v^{(\ell)}) \\ \text{s.t.} & \widehat{rgt}_i(w) = 0, \quad \forall i \in N. \end{aligned} \quad (3)$$

---

<sup>1</sup>Note that there is no need to compute equilibrium inputs— we sample true profiles, and seek to learn rules that are IC.

**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 parameterized auctions  $(g^w, p^w)$  that charge no bidder more than her expected utility for an allocation. In Section 2 and Section 4, we will model the allocation and payment rules as neural networks and incorporate the IR requirement within the architecture.

### 1.3 Our Approach

In this work, we consider two basic end-to-end approaches for solving the multi-item auction design problem, which differ in the way they incorporate the incentive constraints.

**Fully Agnostic Approach.** The first—and more general—approach that we explore does not depend on characterizations of DSIC and payment rules. It therefore bears the biggest promise for being capable of handling cases for which analytical approaches to optimal design get stuck. We use fully-connected multi-layer neural networks to encode auction mechanisms, with bidder valuations being the input and allocation and payment decisions being the output, which we refer to as *RegretNet*. We then train the networks using samples from 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. We propose two different methods to approximate regret during training: *Gradient-based approach*, 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, and *Sample-based approach*, where we use additional *fixed* samples of misreports drawn uniformly from valuation space and compute the maximum over these misreports.

**Characterization-Based Approach.** The second approach that we consider achieves DSIC by hard-coding known characterization results into the network architecture. This approach is less general, as a result of its appeal to characterization results, where the one we make use of only holds for a single bidder case. We include results from this approach because it holds promise as a tool to guide our theoretical understanding of the structure of optimal auctions. As a case in point, we leverage a characterization due to Rochet [43] for single bidder, multi-item auctions. According to this characterization, an auction for additive values on items is DSIC if and only if its induced utility  $u$  is 1-Lipschitz with respect to the  $\ell_1$ -norm, non-decreasing, and convex. An analogous characterization exists for a buyer with unit-demand valuations. Our network architecture (*RochetNet*) exploits this characterization result, not by directly learning the optimal allocation rule and payment rule, but rather by learning the induced utility function<sup>2</sup>.

### 1.4 Our Results

We explore, experimentally, whether our approaches can recover known optimal designs multi-item settings, as well as new designs for settings where such results are not known.

---

<sup>2</sup>As another illustration of how characterization results can be leveraged, we also develop an architecture for multi-bidder, single-item settings. This network, referred to as *MyersonNet*, mimics Myerson’s characterization for single-item auctions [37], and learns monotone transforms of bidder values. We present this approach in Appendix B and evaluate it in Appendix B.1.

**Results for the fully agnostic approach.** 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 (including Manelli and Vincent [33], Pavlov [41], Haghpahan and Hartline [25] and Daskalakis et al. [16]) 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 [45].

(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 too 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 low revenue and regret on the training data translates into low revenue and regret on newly sampled valuations.

**Results for the characterization-based approach.** Our second set of results demonstrate the use of RochetNet to recover the optimal mechanisms for the single bidder, two item settings described above. Plots of the learned allocation rules reveal that we learn to match almost precisely the theoretically optimal ones, although we have not encoded the specific (allocation) structure of the optimal mechanisms explicitly into the network. As a demonstration of its utility as a theoretical tool, we provide support for a conjecture of Daskalakis et al. [16] on the structure of optimal auction for a single additive bidder, two item setting with independent uniform valuations.

## 1.5 Discussion

By focusing on expected ex post regret we adopt a quantifiable relaxation of dominant-strategy incentive compatibility, first introduced in [18]. 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 [45]) 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) [12, 13], follow-up works have acknowledged that this approach has severe scalability issues as it requires a number of constraints and variables that is exponential in the number of agents and items [23]. 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).

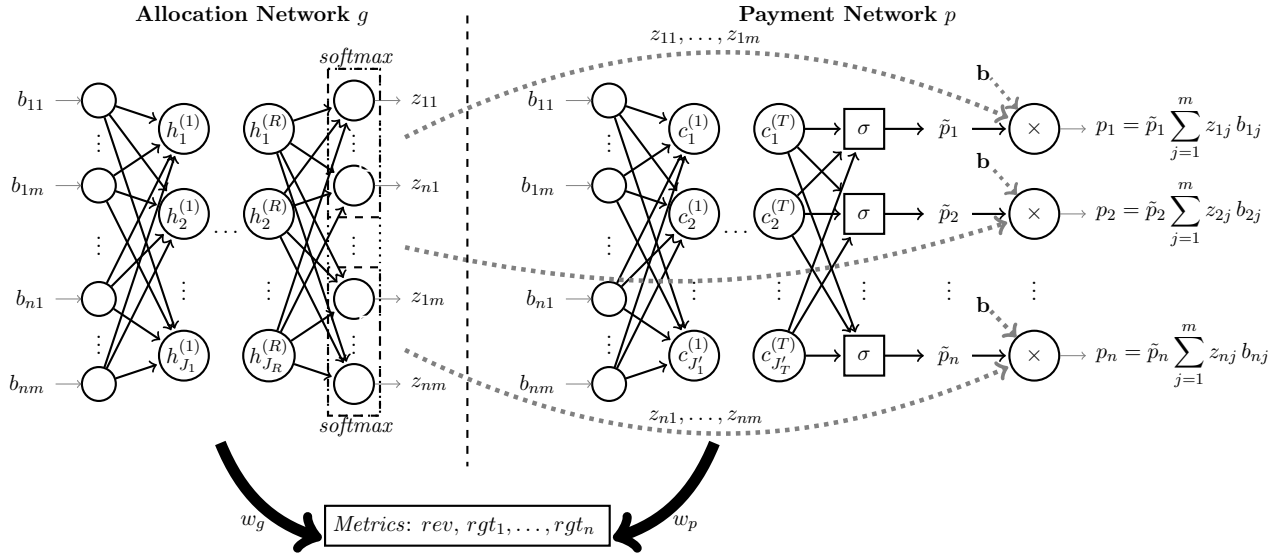
## 1.6 Further Related Work

There are sample complexity results in the literature for the design of optimal single-item auctions [11, 34, 29], single bidder, multi-item auctions [17], general single-parameter settings [35], combinatorial auctions [3, 36, 49], and allocation mechanisms (both with and without money) [39].

In addition, 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 [28], an automated equilibrium analysis of mechanisms [50], deep nets for causal inference [27, 32], and deep reinforcement learning for solving combinatorial games [42]. There has also been follow-up work to the present paper that extends our approach to budget constrained bidders [19] and to the facility location problem [22], and that develops specialized architectures for single bidder settings that satisfy IC [47] by generalizing the *RochetNet* architecture.

## 1.7 Road Map

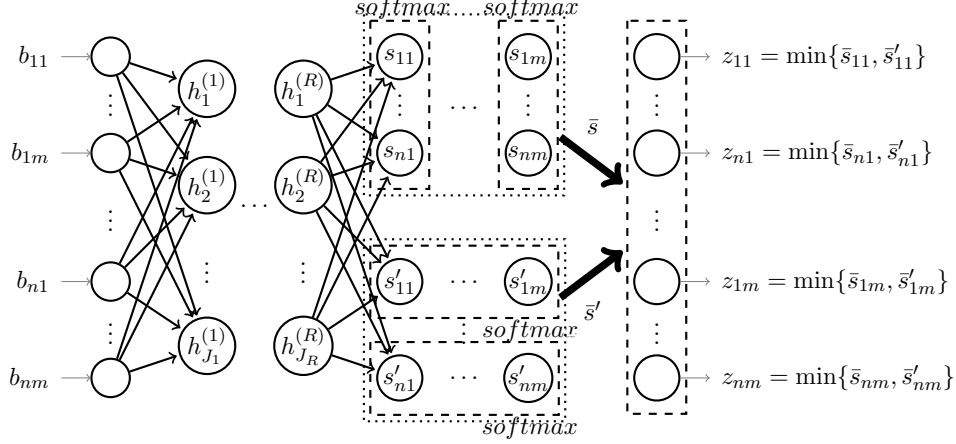
The rest of the paper is organized as follows. The *RegretNet* Framework and the generalization bound of revenue and regret is shown in Section 2. The experimental results for RegretNet is in Section 3. The characterization-based approach for single bidder, multi-items auctions (*RochetNet* framework) and the experimental results are given in Section 4. Another characterization-based approach for single-item auctions, referred to as *MyersonNet*, is shown in Appendix B. The sample-based approach for optimization in RegretNet is shown in Appendix E. We conclude our paper in Section 5. The proofs are in Appendix D.



**Figure 1:** The allocation and payment networks for a setting with  $n$  additive bidders and  $m$  items. The  $\text{rev}$  and each  $\text{rgt}_i$  are defined as a function of the parameters of the allocation and payment networks  $w = (w_g, w_p)$ .

## 2 The RegretNet Framework

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.



**Figure 2:** The allocation network for settings with  $n$  unit-demand bidders and  $m$  items, where  $\bar{s}_{ij} = e^{s_{ij}} / \sum_{k=1}^{n+1} e^{s_{kj}}$ . The payment network for this setting is the same as in Figure 1.

The architectures contain two logically distinct components: the allocation and payment networks. The outputs of these networks are used to compute the regret and revenue of the auction.

## 2.1 Network Architecture for Additive Valuations

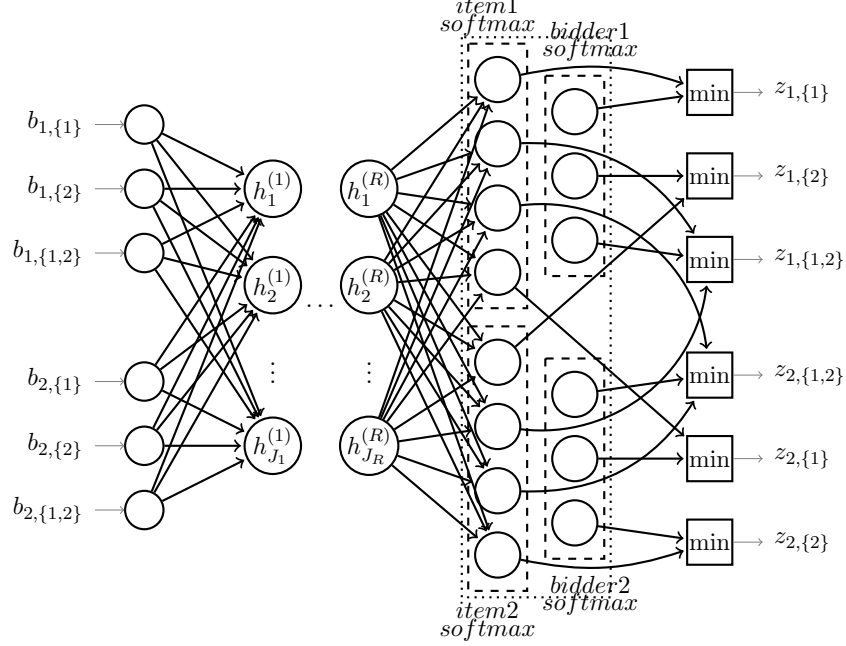
A bidder has additive valuations if the bidder's value for a bundle of items  $S \subseteq M$  is the sum of her value for the individual items in  $S$ , i.e.  $v_i(S) = \sum_{j \in S} v_i(j)$ . In this case, the bidders report only their valuations for individual items. The architecture for this setting models a randomized allocation network  $g^w : \mathbb{R}^{nm} \rightarrow [0, 1]^{nm}$  and a payment network  $p^w : \mathbb{R}^{nm} \rightarrow \mathbb{R}_{\geq 0}^n$ , 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  $z_{1j} = g_{1j}(b), \dots, z_{nj} = g_{nj}(b)$ , 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 sigmoid 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.

## 2.2 Network Architecture for 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(a). For revenue maximization in this setting, it can be shown that it is sufficient to consider allocation rules that



**Figure 3:** The allocation network for settings with 2 *combinatorial* bidders and 2 items, where  $\bar{s}'_{ij} = e^{s'_{ij}} / \sum_{k=1}^{m+1} e^{s'_{jk}}$ . The payment networks for this setting is the same as in Figure 1.

assign at most one item to each bidder.<sup>3</sup> 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.<sup>4</sup>

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:

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

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.

### 2.3 Network Architecture for Combinatorial Valuations

We also consider bidders with general, combinatorial valuations. In the present work, we develop this architecture only for small number of items. In this case, each bidder  $i$  reports a bid  $b_{i,S}$  for

<sup>3</sup> 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.

<sup>4</sup> A randomized allocation represented by a doubly-stochastic matrix can be decomposed into a lottery over deterministic one-to-one assignments [5, 51].



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

$$\sum_{i \in N} \sum_{S \subseteq M: j \in S} z_{i,S} \leq 1, \forall j \in M; \quad (4)$$

$$\sum_{S \subseteq M} z_{i,S} \leq 1, \forall i \in N. \quad (5)$$

We refer to an allocation that satisfies constraints (4)–(5) 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  $s_{i,S}^{(j)}, \forall i \in N, S \subseteq M$  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  $\bar{s}_{i,S}^{(j)} = \exp(s_{i,S}^{(j)}) / \sum_{S', S''} \exp(s_{i,S'}^{(j)})$ . 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  $\bar{s}_{i,S}^{(j)}$  for each  $j \in S$ :

$$z_{i,S} = \varphi_{i,S}^{CF}(s, s^{(1)}, \dots, s^{(m)}) = \min \{ \bar{s}_{i,S}, \bar{s}_{i,S}^{(j)} : j \in S \}.$$

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

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:  $s_{1,\{1\}}^1, s_{2,\{1\}}^1, s_{1,\{1,2\}}^1$ , and  $s_{2,\{1,2\}}^1$ , one for each assignment where item 1 is present, and similarly, four scores for item 2:  $s_{1,\{2\}}^2, s_{2,\{2\}}^2, s_{1,\{1,2\}}^2$ , and  $s_{2,\{1,2\}}^2$ . Each set of scores is then normalized by separate softmax functions. The final allocation for each bidder  $i$  is:  $z_{i,\{1\}} = \min\{\bar{s}_{i,\{1\}}, \bar{s}_{i,\{1\}}^1\}$ ,  $z_{i,\{2\}} = \min\{\bar{s}_{i,\{2\}}, \bar{s}_{i,\{2\}}^2\}$ , and  $z_{i,\{1,2\}} = \min\{\bar{s}_{i,\{1,2\}}, \bar{s}_{i,\{1,2\}}^1, \bar{s}_{i,\{1,2\}}^2\}$ .

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.

## 2.4 Generalization Bound

We provide a generalization bound for the revenue and regret. We bound the gap between the empirical regret and the expected regret in terms of the sample size, for any auction chosen from a finite capacity class. We show a similar result for the revenue.

To deal with the non-standard ‘max’ structure in the regret, we measure the capacity of an auction class using a definition of covering numbers used in the ranking literature [44]. 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  $\epsilon$  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  $(g^w, p^w) \in \mathcal{M}$ ,

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

$$\text{and } \forall i \in N, \quad \frac{1}{n} \sum_{i=1}^n \text{rgt}_i(w) \leq \frac{1}{n} \sum_{i=1}^n \widehat{\text{rgt}}_i(w) + 2\Delta_L + C'\sqrt{\frac{\log(1/\delta)}{L}},$$

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

See Appendix for the proof. If the term  $\Delta_L$  in the above bound goes down to 0 as the sample size  $L$  increases, then the difference between the expected and empirical regret is upper bounded by a term that goes to 0 as  $L \rightarrow \infty$ .

**Covering Number Bounds.** We now bound the term  $\Delta_L$  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  $\Delta_L$  for different bidder valuation types:

- (a) additive valuations:  $\Delta_L \leq O(\sqrt{R(d_a + d_p) \log(LW \max\{K, mn\})/L})$ ,
- (b) unit-demand valuations:  $\Delta_L \leq O(\sqrt{R(d_a + d_p) \log(LW \max\{K, mn\})/L})$ ,
- (c) combinatorial valuations:  $\Delta_L \leq O(\sqrt{R(d_a + d_p) \log(LW \max\{K, n 2^m\})/L})$ .

See Appendix for the proof. As the sample size  $L \rightarrow \infty$ , the term  $\Delta_L \rightarrow 0$ . 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 [1]. Note that the logarithm in the bound for combinatorial valuations cancels the exponential dependence on the number of items  $m$ .

## 2.5 Optimization and Training

We use the augmented Lagrangian method to solve the constrained training problem in (3) over the space of neural network parameters  $w$ . We first define the Lagrangian function for the optimization problem, augmented with a quadratic penalty term for violating the constraints:

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

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)  $w^{\text{new}} \in \arg\min_w \mathcal{C}_\rho(w^{\text{old}}; \lambda^{\text{old}})$  and (b)  $\lambda_i^{\text{new}} = \lambda_i^{\text{old}} + \rho \widehat{\text{rgt}}_i(w^{\text{new}})$ ,  $\forall i \in N$ .

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  $\mathcal{S}_t = \{u^{(1)}, \dots, u^{(B)}\}$ . 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

---

**Algorithm 1** RegretNet Training

---

**Input:** Minibatches  $\mathcal{S}_1, \dots, \mathcal{S}_T$  of size  $B$   
**Parameters:**  $\forall t, \rho_t > 0, \gamma > 0, \eta > 0, R \in \mathbb{N}, Z \in \mathbb{N}$   
**Initialize:**  $w^0 \in \mathbb{R}^d, \lambda^0 \in \mathbb{R}^n$   
**for**  $t = 0$  **to**  $T$  **do**  
  Receive minibatch  $\mathcal{S}_t = \{u^{(1)}, \dots, u^{(B)}\}$   
  Initialize misreports  $v_i^{(\ell)} \in V_i, \forall \ell \in [B], i \in N$   
  **for**  $r = 0$  **to**  $R$  **do**  
     $\forall \ell \in [B], i \in N : v_i^{(\ell)} \leftarrow v_i^{(\ell)} + \gamma \nabla_{v_i'} u_i^w(v_i^{(\ell)}; (v_i^{(\ell)}, v_{-i}^{(\ell)}))$   
  **end for**  
  Compute regret gradient:  $\forall \ell \in [B], i \in N$ :  
     $g_{\ell,i}^t = \nabla_w [u_i^w(v_i^{(\ell)}; (v_i^{(\ell)}, v_{-i}^{(\ell)})) - u_i^w(v_i^{(\ell)}; v^{(\ell)})] \Big|_{w=w^t}$   
  Compute Lagrangian gradient using (6) 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  $Z$  iterations:  
    **if**  $t$  is a multiple of  $Z$   
       $\lambda_i^{t+1} \leftarrow \lambda_i^t + \rho_t \widehat{rgt}_i(w^{t+1}), \forall i \in N$   
    **else**  
       $\lambda^{t+1} \leftarrow \lambda^t$   
  **end for**

---

$\widehat{rgt}_i(\mathcal{S}_t; w)$  denote the empirical regret  $\widehat{rgt}_i(w)$  in (2) computed on mini-batch  $\mathcal{S}_t$ . The gradient of  $\mathcal{C}_\rho$  w.r.t.  $w$  for fixed  $\lambda^t$  is given by:

$$\nabla_w \mathcal{C}_\rho(w; \lambda^t) = -\frac{1}{B} \sum_{\ell=1}^B \sum_{i \in N} \nabla_w p_i^w(v^{(\ell)}) + \sum_{i \in N} \sum_{\ell=1}^B \lambda_i^t g_{\ell,i} + \rho \sum_{i \in N} \sum_{\ell=1}^B \widehat{rgt}_i(\mathcal{S}_t; w) g_{\ell,i} \quad (6)$$

where

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

The terms  $\widehat{rgt}_i(\mathcal{S}_t; w)$  and  $g_{\ell,i}$  in turn involve a “max” over misreports for each bidder  $i$  and valuation profile  $\ell$ . We propose two different approaches to approximately solve this inner maximization over misreports:

**Gradient-based approach for misreports.** We solve the inner maximization over misreports using another gradient based optimizer, and push the gradient through the utility differences at the optimal misreports. This has a flavor of adversarial learning. In particular, we maintain misreports  $v_i^{(\ell)}$  for each  $i$  and valuation profile  $\ell$ . For every update on the model parameters  $w^t$ , perform  $R$  gradient updates to compute the optimal misreports:  $v_i^{(\ell)} = v_i^{(\ell)} + \gamma \nabla_{v_i'} u_i^w(v_i^{(\ell)}; (v_i^{(\ell)}, v_{-i}^{(\ell)}))$ , for some  $\gamma > 0$ . For  $g_{\ell,i}$ , we set  $V_i = v_i^{(\ell)}$  for each  $i$  and valuation profile  $\ell$ . In our experiments, we use the Adam optimizer [30] for updates on model  $w$  and  $v_i^{(\ell)}$ . The details of this solver are described in Algorithm 1.

**Sample-based approach for misreports.** We calculate this ‘max’ approximately using additional *fixed* samples of valuation profiles  $S^{(\ell)}$  drawn uniformly from  $V$  for each profile  $v^{(\ell)}$  in  $S$ ,

and computing the maximum over these profiles. Compared with Algorithm 1, we do not need to compute a new  $v_i^{(\ell)}$  for each profile  $\ell$  at each iteration, but only set  $V_i = S_i^{(\ell)}$  for each  $g_{\ell,i}$  using this approach to update  $w$ .

In general, we find the gradient-based approach is more scalable and stable for larger settings, and it requires that the valuation space is continuous and the utility function is differentiable. The sample-based approach is more efficient in small settings, and it can adequately handle the discrete valuation cases, e.g. Yao’s work [54]. For the main body of the paper, we will focus on the gradient-based approach for misreports. We provide experimental results for the sample-based approach, to validate its efficiency and limitations, in Appendix E.1, and prove a generalization bound for this approach to approximate regret in Appendix E.2.

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. For more discussion about Augmented Lagrangian, see Appendix A.

### 3 Experimental Results for RegretNet

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.

#### 3.1 Experimental Setup and Evaluation

**Setup.** We use the TensorFlow deep learning library to implement our learning algorithms. We used the Glorot uniform initialization [21] for all networks and the tanh activation function at the hidden nodes.<sup>5</sup> For all the experiments, we use a sample of 640,000 valuation profiles for training<sup>6</sup> 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  $\rho$  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  $\lambda^t$  was performed once in every 100 minibatches (i.e.  $Z = 100$ ). Our experiments were run on a compute cluster with NVIDIA GPU cores.<sup>7</sup>

**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 (2)). 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).

<sup>5</sup>ReLU activations yield comparable results for smaller settings, but tanh works better for larger settings (X) - (XII).

<sup>6</sup>For small settings (I) - (V), the performance of RegretNet with smaller training samples (around 5000) are also well-behaved.

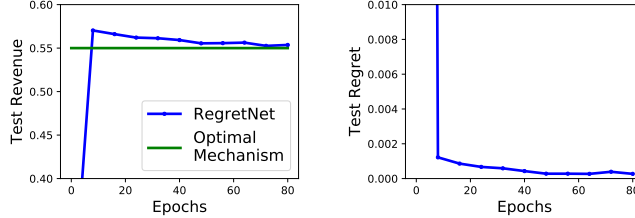
<sup>7</sup>A single iteration of augmented Lagrangian took on an average 1–17 seconds across experiments.

Distribution	Opt <i>irp</i>	RegretNet		RochetNet
		<i>rev</i>	<i>rgt</i>	<i>rev</i>
Setting (I): One additive bidder, $v_1, v_2 \sim U[0, 1]$	0.550	0.554	< 0.001	0.550
Setting (II): One additive bidder, $v_1 \sim [4, 16], v_2 \sim U[4, 7]$	9.781	9.734	< 0.001	9.779
Setting (III): One additive bidder, $v_1, v_2$ drawn uniformly from a unit triangle	0.388	0.392	< 0.001	0.388
Setting (IV): One unit-demand bidder, $v_1, v_2 \sim U[0, 1]$	0.384	0.384	< 0.001	0.384
Setting (V): One unit-demand bidder, $v_1, v_2 \sim U[2, 3]$	2.137	2.137	< 0.001	2.136

(a)

Distribution	<i>rev</i>	<i>rgt</i>	VVCA	AMA <sub>bsym</sub>
Setting (VI)	<b>0.878</b>	< 0.001	0.860	0.862
Setting (VII)	<b>2.871</b>	< 0.001	2.741	2.765
Setting (VIII)	<b>4.270</b>	< 0.001	4.209	3.748

(b)



(c)

**Figure 4:** (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).

### 3.2 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. We consider the following settings:

- (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 and Vincent [33].
- (II) Single additive bidder with preferences over two non-identically distributed items, where  $v_1 \sim U[4, 16]$  and  $v_2 \sim U[4, 7]$ . The optimal mechanism is given by Daskalakis et al. [16].
- (III) 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  $(1, 0)$ . The optimal mechanism is given by Haghpanah and Hartline [25].
- (IV) Single unit-demand bidder with preferences over two items, where the item values  $v_1, v_2 \sim U[0, 1]$ . The optimal mechanism is given by Pavlov [41].
- (V) 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 [41].

Figure 4(a) presents the revenue and regret of the final auctions learned for settings (I) - (V) on the test set with an architecture with two hidden nodes and 100 nodes per layer.<sup>8</sup> 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 5(a)-(e) show that our approach also closely recovers the structure of the optimal auctions.

Figure 4(c) presents a plot of revenue and regret as a function of the training epochs (number of passes through the whole training data). The solver adaptively tunes the Lagrange multiplier

<sup>8</sup>Based on evaluations on a held-out set, we found the gains to be negligible when we used more number of layers or nodes.

Distribution	Method	<i>rev</i>	<i>rgt</i>	<i>IR viol.</i>	Run-time
2 additive bidders, 3 items with $v_{ij} \sim U[0, 1]$	RegretNet	1.291	< 0.001	<b>0</b>	<b>~9 hrs</b>
	LP (D: 5 bins/value)	<b>1.53</b>	0.019	0.027	69 hrs

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

on the regret, focusing on the revenue in the initial iterations and on regret in later iterations.

### 3.3 Multiple bidders

We next compare to the state-of-the-art computational results of Sandholm and Likhodedov [45] for settings for which the optimal auction is *not* known. These auctions are obtained by searching over a parameterized 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.

- (VI) 2 additive bidders and 2 items, where bidders draw their value for each item from  $U[0, 1]$ .
- (VII) 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  $C_1, C_2 \sim U[-1, 1]$ .
- (VIII) 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)-(V). We compare the trained mechanism with the optimal auctions from the VVCA and  $\text{AMA}_{\text{bsym}}$  families of incentive compatible auctions from [45]. Figure 4(b) summarizes our results. Our approach leads to significant revenue improvements and tiny regret. Comparing with Figure 4(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

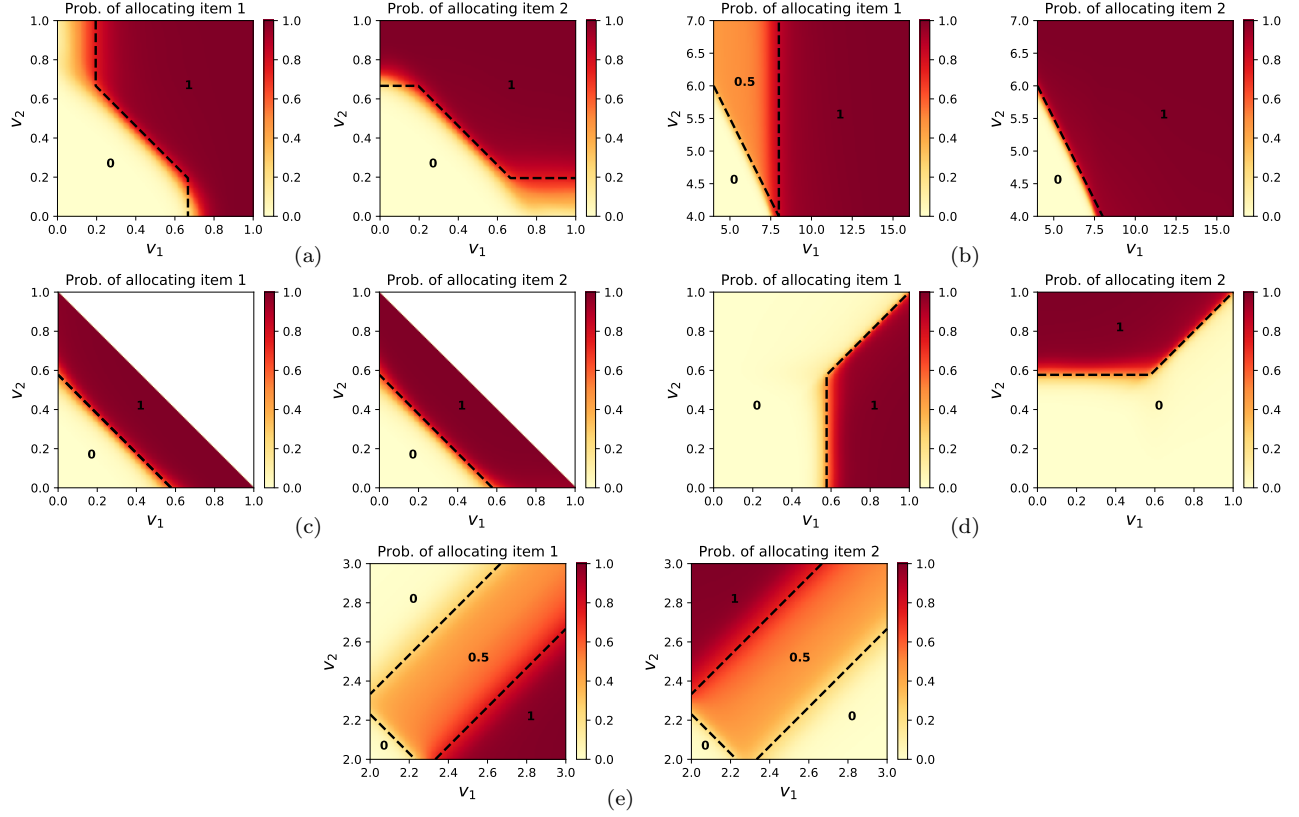
### 3.4 Scaling up

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 (Item-wise Myerson) is optimal in the limit of number of bidders [40]<sup>9</sup>. This yields a very strong but still improvable benchmark.

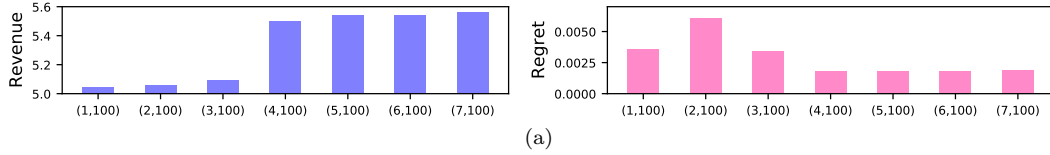
- (IX) Single additive bidder with preferences over ten items, where each  $v_i \sim U[0, 1]$ .
- (X) 3 additive bidders and 10 items, where bidders draw their value for each item from  $U[0, 1]$ .
- (XI) 5 additive bidders and 10 items, where bidders draw their value for each item from  $U[0, 1]$ .

For setting (X), we show in Figure 6(a) the revenue and regret of the learned auction on a validation sample of 10000 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 6(b) shows that the final learned auctions of settings (X) and (XI) yield higher revenue (with tiny regret) compared to the baselines. Table 2 shows the results for setting (IX) learned by RegretNet.

<sup>9</sup>We also compare with a Myerson auction (Bundled Myerson) to sell the entire bundle of items as one unit, which is optimal in the limit of number of items [40].



**Figure 5:** Allocation rules learned by RegretNet for single-bidder, two items settings: (a) Setting (I), (b) Setting (II), (c) Setting (III), (d) Setting (IV), and (e) Setting (V). 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.



Distribution	$rev$	$rgt$	Item-wise Myerson	Bundled Myerson
Setting (X): 3 additive bidders, 10 items with $v_{ij} \sim U[0, 1]$	<b>5.541</b>	< 0.002	5.310	5.009
Setting (XI): 5 additive bidders, 10 items with $v_{ij} \sim U[0, 1]$	<b>6.778</b>	< 0.005	6.716	5.453

(b)

**Figure 6:** (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).

### 3.5 Comparison to LP

We also compare the running time of our algorithm with the LP approach proposed by [12, 13]. 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  $U[0, 1]$ . 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 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 (X) - (XI), the running time of our algorithm was less than 13 hours.

## 4 The RochetNet Framework

In this section, we show how to leverage characterization results to develop essentially optimal designs that are, moreover, exactly DSIC and IR. We focus on the setting of a single bidder with preferences over multiple items, where there is no analytical solution to the problem even with additive values and more than six items [14]. In fact, we only have an implicit characterization of the optimal DSIC mechanism for this setting. Here we use neural networks to search over a specific class of fully-DSIC mechanisms that are known to contain the optimal auction. We think this specialized approach will be of interest as a tool with which to study and test conjectures about the structure of optimal auctions in poorly-understood settings. In the Appendix B, we also develop the specialized *MyersonNet* architecture for the single-item setting.<sup>10</sup>

### 4.1 The Network Architecture

We start with the case where the bidder’s preferences are *additive*. We make use of the characterization for Rochet [43] for a single bidder problem in terms of the bidder’s induced utility and its gradient.<sup>11</sup> The utility function  $u : \mathbb{R}_{\geq 0}^m \rightarrow \mathbb{R}$  induced by a mechanism  $(g, p)$  for a single bidder is:

$$u(v) = \sum_{j=1}^m g_j(v) v_j - p(v). \quad (7)$$

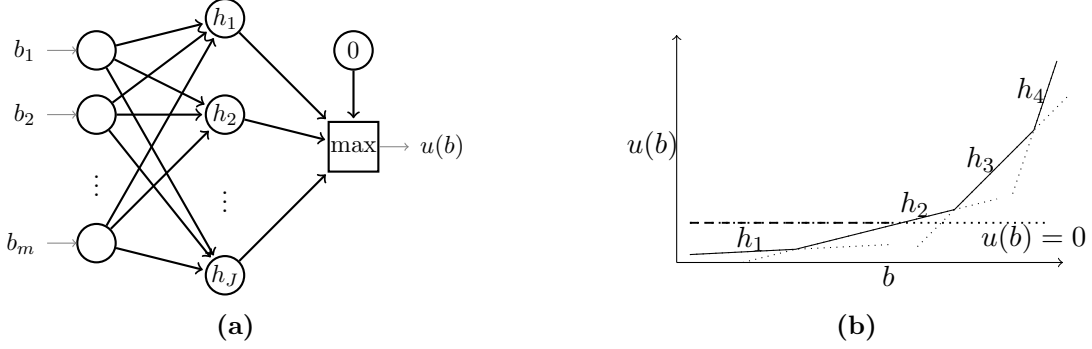
This is the bidder’s utility for bidding truthfully when her valuation is  $v$ . We say that the utility function is monotonically non-decreasing if  $u(v) \leq u(v')$  whenever  $v_j \leq v'_j, \forall j \in M$ . The following theorem explains the connection between a DSIC mechanism and its induced utility function:

**Theorem 3** (Rochet [43]). *A utility function  $u : \mathbb{R}_{\geq 0}^m \rightarrow \mathbb{R}$  is induced by a DSIC mechanism iff  $u$  is 1-Lipschitz w.r.t. the  $\ell_1$ -norm, non-decreasing, and convex. Moreover, for such a utility function  $u$ ,  $\nabla u(v)$  exists almost everywhere in  $\mathbb{R}_{\geq 0}^m$ , and wherever it exists,  $\nabla u(v)$  gives the allocation probabilities for valuation  $v$ , and  $\nabla u(v) \cdot v - u(v)$  is the corresponding payment.*

<sup>10</sup>Myerson characterized the structure of the optimal auction *via* monotone virtual value transformations. We develop a neural network that mimics this characterization, by modeling the monotone transforms as neural networks [48]. We show that MyersonNet can recover the optimal auction for a variety of valuation distributions.

<sup>11</sup>Daskalakis et al. [15] and Daskalakis et al. [16] also make use of Rochet’s characterization in their duality-based characterizations for the single-bidder multi item problem.





**Figure 7:** RochetNet: (a) Neural network representation of a non-negative, monotone, convex induced utility function; here  $h_j(b) = w_j \cdot b + \beta_j$ . (b) An example of a utility function represented by RochetNet for  $m = 1$ .

Further, for a mechanism to be IR, its induced utility function must be non-negative, i.e.  $u(v) \geq 0, \forall v \in \mathbb{R}_{\geq 0}^m$ . To find the optimal mechanism, we need to search over all non-negative utility functions that satisfy the conditions in Theorem 4, and pick the one that maximizes expected revenue. This can be done by modeling the utility function as a neural network, and formulating the above optimization as a neural network learning problem. The associated mechanism can then be recovered from the gradient of the learned neural network.

To model a non-negative, monotone, convex, Lipschitz utility function, we use a max of  $J$  linear functions with non-negative coefficients, and 0:

$$u_{\alpha,\beta}(v) = \max \left\{ \max_{j \in [J]} \{w_j \cdot v + \beta_j\}, 0 \right\}, \quad (8)$$

where each  $w_{jk} = 1/(1 + e^{-\alpha_{jk}})$ , for  $\alpha_{jk} \in \mathbb{R}, j \in [J], k \in M$ , and  $\beta_j \in \mathbb{R}$ . By bounding the hyperplane coefficients to  $[0, 1]$ , we guarantee that the function is 1-Lipschitz.

**Theorem 4.** *For any  $\alpha \in \mathbb{R}^{mJ}$  and  $\beta \in \mathbb{R}^J$ , the function  $u_{\alpha,\beta}$  is non-negative, monotonically non-decreasing, convex and 1-Lipschitz w.r.t. the  $\ell_1$ -norm.*

The utility function, represented as a single layer neural network, is illustrated in Figure 7(a), where each  $h_j(b) = w_j \cdot b + \beta_j$ . Figure 7(b) shows an example of a utility function represented by RochetNet for  $m = 1$ . By using a large number of hyperplanes, one can use this neural network architecture to search over a sufficiently rich class of monotone, convex 1-Lipschitz utility functions. Once trained, the mechanism  $(g, p)$  can be derived from the gradient of the utility function, with the allocation rule given by:

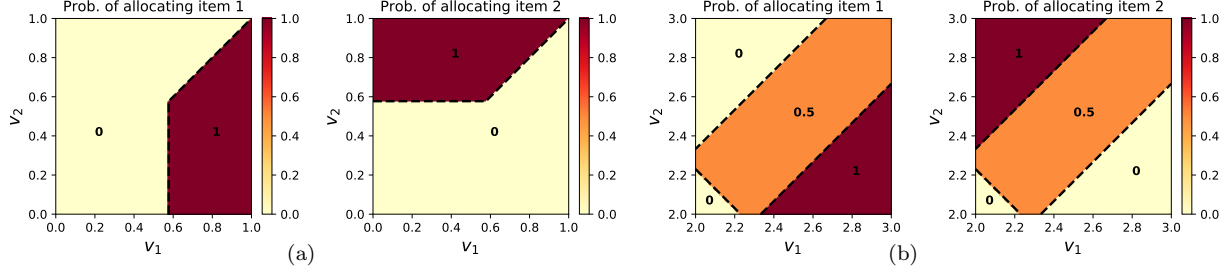
$$g(b) = \nabla u_{\alpha,\beta}(b), \quad (9)$$

and the payment rule is given by the difference between the expected value to the bidder from the allocation and the bidder's utility:

$$p(b) = \nabla u_{\alpha,\beta}(b) \cdot b - u_{\alpha,\beta}(b). \quad (10)$$

Here the utility gradient can be computed as:  $\nabla_j u_{\alpha,\beta}(b) = w_{j^*}$ , for  $j^* \in \operatorname{argmax}_{j \in [J]} \{w_j \cdot b + \beta_j\}$ . We seek to minimize the negated, expected revenue:

$$-\mathbf{E}_{v \sim F} [\nabla u_{\alpha,\beta}(v) \cdot v - u_{\alpha,\beta}(v)]. \quad (11)$$



**Figure 8:** Allocation rules learned by RocketNet for single unit-demand bidder, two items settings: (a) Setting (IV) and (b) Setting (V). 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.

To ensure that the objective is a continuous function of the parameters  $\alpha$  and  $\beta$  (so that the parameters can be optimized efficiently), the gradient term is computed approximately by using a softmax operation in place of the argmax. The loss function that we use is given by the negated revenue with approximate gradients:

$$\mathcal{L}(\alpha, \beta) = -\mathbf{E}_{v \sim F} [\tilde{\nabla} u_{\alpha, \beta}(v) \cdot v - u_{\alpha, \beta}(v)], \quad (12)$$

where

$$\tilde{\nabla}_k u_{\alpha, \beta}(v) = \sum_{j \in [J]} w_{jk} \cdot \text{softmax}_j(\kappa \cdot (w_1 \cdot v + \beta_1), \dots, \kappa \cdot (w_J \cdot v + \beta_J)) \quad (13)$$

and  $\kappa > 0$  is a constant that controls the quality of the approximation. We seek to optimize the parameters of the neural network  $\alpha \in \mathbb{R}^{mJ}, \beta \in \mathbb{R}^J$  to minimize loss  $\mathcal{L}$ . Given a sample  $S = \{v^{(1)}, \dots, v^{(L)}\}$  drawn from  $F$ , we optimize an empirical version of the loss.

Our approach easily extends to a bidder with *unit-demand* valuations. In this case, the sum of the allocation probabilities cannot exceed 1. This is enforced by restricting the coefficients for each hyperplane to sum up to at most 1, i.e.  $\sum_{k=1}^m w_{jk} \leq 1, \forall j \in [J]$ . It can be verified that even with this restriction, the induced utility function continuous to be monotone, convex and Lipschitz, ensuring that the resulting mechanism is DSIC.<sup>12</sup>

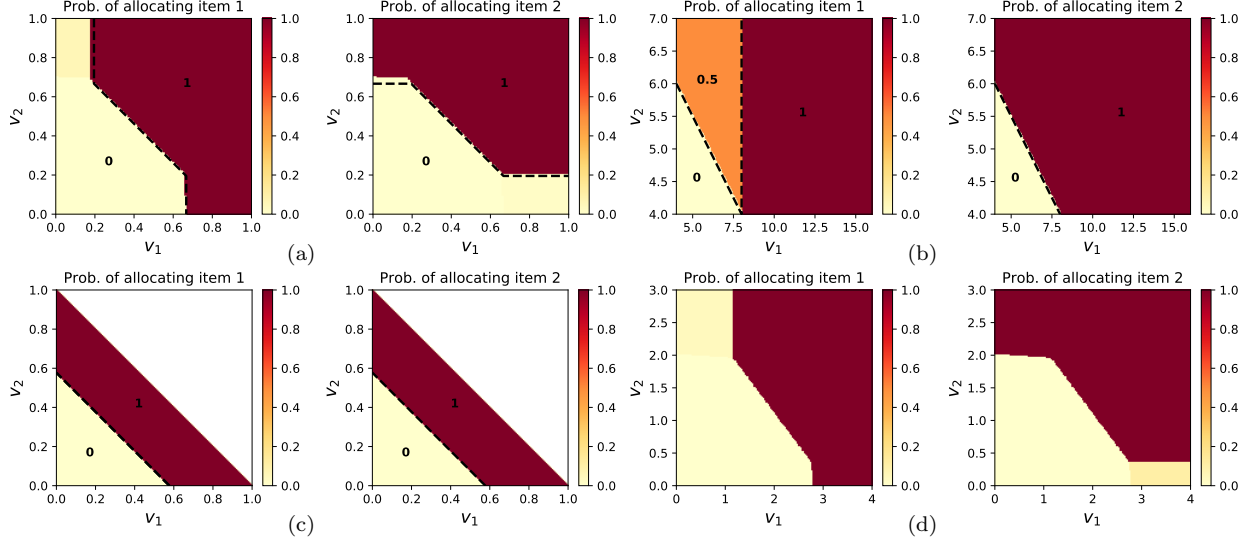
A possible interpretation of the RocketNet architecture is that the network maintains a menu of (randomized) allocations and prices, and chooses the option from the menu that maximizes the bidder's utility based on the bidder's bid. Each linear function  $h_j(b) = w_j \cdot b + \beta_j$  in RocketNet corresponds to an option on the menu, with the allocation probabilities and payments encoded through the parameters  $w_j$  and  $\beta_j$  respectively. Recently, [47] extends our RocketNet to more general settings, including non-linear utility function setting.

## 4.2 Experimental Results for RocketNet

**Setup.** For all the experiments that we report in this section, we use a sample of 640,000 valuation profiles for training and a sample of 10,000 profiles for testing. We use Adam solver for training, with learning rate 0.001 and mini-batch size 128. During the training, we set  $\kappa = 1000$  to approximate argmax by softmax in Equation (13).

We first consider the same distributions used to evaluate the RegretNet architecture in Section 3. We model the induced utility function as a max network over 1000 linear functions. The results

<sup>12</sup>The original characterization of Rocket [43] applies to general, convex outcome spaces (which—as one can verify—is indeed the case here).



**Figure 9:** Allocation rules learned by RochetNet for single additive bidder, two items settings: (a) Setting (I), (b) Setting (II), (c) Setting (IIIss), and (d) Setting (XII). For (a), (b) and (c), 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. For (d), the subset of valuations  $(v_1, v_2)$  where the bidder receives neither item forms a pentagonal shape.

Distribution	Item-wise Myerson <i>rev</i>	Bundled Myerson <i>rev</i>	RegretNet		RochetNet <i>rev</i>
			<i>rev</i>	<i>rgt</i>	
Setting (IX): $v_i \sim U[0, 1]$	2.495	3.457	<b>3.461</b>	< 0.003	3.459
Setting (XII): $v_1 \sim U[0, 4], v_2 \sim U[0, 3]$	1.877	1.749	<b>1.911</b>	< 0.001	1.912

**Table 2:** Revenue of auctions for single additive bidder, 10 items with  $v_i \sim U[0, 1]$  and single additive bidder, 2 items with  $v_1 \sim U[0, 4], v_2 \sim U[0, 3]$ , obtained with RegretNet and RochetNet.

provided in Figure 4 (a) (Section 3) show that the revenue of the trained mechanisms are close to the optimal revenue. Table 2 shows the results for the single bidder, 10 item distribution, where the revenue from RochetNet is better than the baselines. In Figure 8, we show the allocation rules learned by RochetNet for the single unit-demand bidder, multi-items settings. RochetNet almost exactly recovers the optimal mechanisms proposed by Pavlov [41].

Finally, we use RochetNet to test a conjecture of Daskalakis et al. [15] for a single additive bidder, two item settings where the item valuations are drawn independently from uniform distributions. We consider Setting (XII) as follows,

- (XII) Single additive bidder with preferences over two items, where  $v_1 \sim U[0, 4]$  and  $v_2 \sim U[0, 3]$ . The optimal mechanism is still unknown.

Daskalakis et al. [15] conjecture that the optimal mechanism for these settings will assign zero utility to a subset of valuations that have a pentagonal shape. We applied RochetNet to various valuation distributions of the above form (including settings (I), (II) and (III)), and find that the learned mechanisms have allocation plots where the zero utility regions do indeed have a (full or degenerate) pentagonal shape. For example, in Figure 9(d), we show the allocation plots for a mechanism learned by RochetNet for setting (XII) with valuations  $v_1 \sim U[0, 4]$  and  $v_2 \sim U[0, 3]$  (the results learned by RegretNet and RochetNet is shown in Table 2.). There is a pentagonal region where the bidder receives neither item, thus lending evidence in support of Daskalakis et

al.’s conjecture. This demonstrates the use of RochetNet as a tool for gaining insights into the structure of optimal auctions.

Generally, RochetNet is able to yield sharper decision boundaries than RegretNet, and match the optimal mechanism more closely. This is because for the valuation distributions considered, the optimal mechanism can be described by a finite menu of allocations and payments, and RochetNet effectively recovers the optimal menu of options for these distributions. With this carefully designed structure, the running time of RochetNet for each of the above experiments is less than 10 minutes, and much faster than RegretNet.

## 5 Conclusion

Neural networks have been successfully used for data-driven discovery in other contexts, e.g., for the discovery of new drugs [24]. 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 magnitude 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 additional innovations in the network architecture.

## 6 Acknowledgment

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

## References

- [1] Martin Anthony and Peter L. Bartlett. 2009. *Neural Network Learning: Theoretical Foundations* (1st ed.). Cambridge University Press.
- [2] M. Babaioff, N. Immorlica, B. Lucier, and S. M. Weinberg. 2014. A Simple and Approximately Optimal Mechanism for an Additive Buyer. In *Proceedings of the 55th IEEE Symposium on Foundations of Computer Science*. 21–30.
- [3] M-F. Balcan, T. Sandholm, and E. Vitercik. 2016. Sample Complexity of Automated Mechanism Design. In *Proceedings of the 29th Conference on Neural Information Processing Systems*. 2083–2091.
- [4] D. P. Bertsekas. 2014. *Constrained optimization and Lagrange multiplier methods*. Academic press.
- [5] G. Birkhoff. 1946. Tres observaciones sobre el algebra lineal. *Universidad Nacional de Tucumán, Revista A* 5 (1946), 147–151.
- [6] Y. Cai, C. Daskalakis, and M. S. Weinberg. 2012. Optimal Multi-dimensional Mechanism Design: Reducing Revenue to Welfare Maximization. In *Proceedings of the 53rd IEEE Symposium on Foundations of Computer Science*. 130–139.

- [7] Y. Cai, C. Daskalakis, and S. M. Weinberg. 2012. An algorithmic characterization of multi-dimensional mechanisms. In *Proceedings of the 44th ACM Symposium on Theory of Computing*. 459–478.
- [8] Y. Cai, C. Daskalakis, and S. M. Weinberg. 2013. Understanding Incentives: Mechanism Design Becomes Algorithm Design. In *Proceedings of the 54th IEEE Symposium on Foundations of Computer Science*. 618–627.
- [9] Y. Cai and M. Zhao. 2017. Simple Mechanisms for Subadditive Buyers via Duality. In *Proceedings of the 49th ACM Symposium on Theory of Computing*. 170–183.
- [10] S. Chawla, J. D. Hartline, D. L. Malec, and B. Sivan. 2010. Multi-parameter mechanism design and sequential posted pricing. In *Proceedings of the 42th ACM Symposium on Theory of Computing*. 311–320.
- [11] R. Cole and T. Roughgarden. 2014. The sample complexity of revenue maximization. In *Proceedings of the 46th ACM Symposium on Theory of Computing*. 243–252.
- [12] V. Conitzer and T. Sandholm. 2002. Complexity of Mechanism Design. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*. 103–110.
- [13] V. Conitzer and T. Sandholm. 2004. Self-interested automated mechanism design and implications for optimal combinatorial auctions. In *Proceedings of the 5th ACM Conference on Electronic Commerce*. 132–141.
- [14] C. Daskalakis. 2015. Multi-item auctions defying intuition? *SIGecom Exchanges* 14, 1 (2015), 41–75.
- [15] C. Daskalakis, A. Deckelbaum, and C. Tzamos. 2013. Mechanism design via optimal transport. In *Proceedings of the 14th ACM Conference on Electronic Commerce*. 269–286.
- [16] C. Daskalakis, A. Deckelbaum, and C. Tzamos. 2017. Strong Duality for a Multiple-Good Monopolist. *Econometrica* 85 (2017), 735–767. Issue 3.
- [17] S. Dughmi, L. Han, and N. Nisan. 2014. Sampling and Representation Complexity of Revenue Maximization. In *Proceedings of the 10th Conference on Web and Internet Economics*. 277–291.
- [18] P. Dütting, F. Fischer, P. Jirapinyo, J. Lai, B. Lubin, and D. C. Parkes. 2014. Payment Rules through Discriminant-Based Classifiers. *ACM Transactions on Economics and Computation* 3, 1 (2014), 5.
- [19] Zhe Feng, Harikrishna Narasimhan, and David C. Parkes. 2018. Deep Learning for Revenue-Optimal Auctions with Budgets. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*. 354–362.
- [20] Y. Giannakopoulos and E. Koutsoupias. 2015. Selling Two Goods Optimally. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming*. 650–662.
- [21] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feed-forward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*.

- [22] Noah Golowich, Harikrishna Narasimhan, and David C. Parkes. 2018. Deep Learning for Multi-Facility Location Mechanism Design. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 261–267.
- [23] Mingyu Guo and Vincent Conitzer. 2010. Computationally Feasible Automated Mechanism Design: General Approach and Case Studies. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*.
- [24] Rafael Gmez-Bombarelli, Jennifer N. Wei, David Duvenaud, Jos Miguel Hernandez-Lobato, Benjamn Snchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Aln Aspuru-Guzik. 2018. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science* 4, 2 (2018), 268–276.
- [25] N. Haghpanah and J. Hartline. 2015. Multi-dimensional Virtual Values and Second-degree Price Discrimination. *CoRR* abs/1404.1341 (2015).
- [26] S. Hart and N. Nisan. 2017. Approximate revenue maximization with multiple items. *Journal of Economic Theory* 172 (2017), 313–347.
- [27] J. S. Hartford, G. Lewis, K. Leyton-Brown, and M. Taddy. 2017. Deep IV: A Flexible Approach for Counterfactual Prediction. In *Proceedings of the 34th International Conference on Machine Learning*. 1414–1423.
- [28] Jason S Hartford, James R Wright, and Kevin Leyton-Brown. 2016. Deep learning for predicting human strategic behavior. In *Proceedings of the 29th Conference on Neural Information Processing Systems*. 2424–2432.
- [29] Z. Huang, Y. Mansour, and T. Roughgarden. 2015. Making the Most of Your Samples. In *Proceedings of the 16th ACM Conference on Economics and Computation*. 45–60.
- [30] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014).
- [31] Sébastien Lahaie. 2011. A Kernel-Based Iterative Combinatorial Auction. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*. 695–700.
- [32] C. Louizos, U. Shalit, J. M. Mooij, D. Sontag, R. S. Zemel, and M. Welling. 2017. Causal Effect Inference with Deep Latent-Variable Models. In *Proceedings of the 30th Conference on Neural Information Processing Systems*. 6449–6459.
- [33] A. Manelli and D. Vincent. 2006. Bundling as an optimal selling mechanism for a multiple-good monopolist. *Journal of Economic Theory* 127, 1 (2006), 1–35.
- [34] M. Mohri and A. M. Medina. 2016. Learning Algorithms for Second-Price Auctions with Reserve. *Journal of Machine Learning Research* 17 (2016), 74:1–74:25.
- [35] J. Morgenstern and T. Roughgarden. 2015. On the Pseudo-Dimension of Nearly Optimal Auctions. In *Proceedings of the 28th Conference on Neural Information Processing Systems*. 136–144.
- [36] J. Morgenstern and T. Roughgarden. 2016. Learning simple auctions. In *Proceedings of the 29th Conference on Learning Theory*. 1298–1318.

- [37] R. Myerson. 1981. Optimal Auction Design. *Mathematics of Operations Research* 6 (1981), 58–73.
- [38] H. Narasimhan, S. Agarwal, and D. C. Parkes. 2016. Automated Mechanism Design without Money via Machine Learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. 433–439.
- [39] H. Narasimhan and D. C. Parkes. 2016. A General Statistical Framework for Designing Strategy-proof Assignment Mechanisms. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.
- [40] Thomas Palfrey. 1983. Bundling Decisions by a Multiproduct Monopolist with Incomplete Information. *Econometrica* 51, 2 (1983), 463–83.
- [41] G. Pavlov. 2011. Optimal Mechanism for Selling Two Goods. *B.E. Journal of Theoretical Economics* 11 (2011), 1–35. Issue 1.
- [42] M. Raghu, A. Irpan, J. Andreas, R. Kleinberg, Q. V. Le, and J. M. Kleinberg. 2018. Can Deep Reinforcement Learning Solve Erdos-Selfridge-Spencer Games?. In *Proceedings of the 35th International Conference on Machine Learning*. 4235–4243.
- [43] J.-C. Rochet. 1987. A Necessary and Sufficient Condition for Rationalizability in a Quasilinear Context. *Journal of Mathematical Economics* 16 (1987), 191–200.
- [44] C. Rudin and R. E. Schapire. 2009. Margin-based Ranking and an Equivalence between AdaBoost and RankBoost. *Journal of Machine Learning Research* 10 (2009), 2193–2232.
- [45] T. Sandholm and A. Likhodedov. 2015. Automated Design of Revenue-Maximizing Combinatorial Auctions. *Operations Research* 63, 5 (2015), 1000–1025.
- [46] Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA.
- [47] Weiran Shen, Pingzhong Tang, and Song Zuo. 2018. Computer-aided mechanism design: designing revenue-optimal mechanisms via neural networks. *CoRR* abs/1805.03382 (2018). arXiv:1805.03382
- [48] J. Sill. 1998. Monotonic networks. In *Proceedings of the 12th Conference on Neural Information Processing Systems*. 661–667.
- [49] Vasilis Syrgkanis. 2017. A Sample Complexity Measure with Applications to Learning Optimal Auctions. In *Proceedings of the 20th Conference on Neural Information Processing Systems*. 5358–5365.
- [50] D. Thompson, N. Newman, and K. Leyton-Brown. 2017. The Positronic Economist: A Computational System for Analyzing Economic Mechanisms. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 720–727.
- [51] J. von Neumann. 1953. A certain zero-sum two-person game equivalent to an optimal assignment problem. *Annals of Mathematical Studies* 28 (1953), 5–12.
- [52] S. Wright and J. Nocedal. 1999. Numerical optimization. *Springer* (1999).

- [53] A. C-C. Yao. 2015. An  $n$ -to-1 Bidder Reduction for Multi-item Auctions and its Applications. In *Proceedings of the 26th ACM-SIAM Symposium on Discrete Algorithms*. 92–109.
- [54] A. C-C. Yao. 2017. Dominant-Strategy versus Bayesian Multi-item Auctions: Maximum Revenue Determination and Comparison. In *Proceedings of the 18th ACM Conference on Economics and Computation*. 3–20.

## Appendix

### A Augmented Lagrangian Method for Constrained Optimization

We give a brief description of the Augmented Lagrangian method for solving constrained optimization problems [4]. We use this method for solving neural network training problems involving equality constraints.

Consider the following optimization problem with  $s$  equality constraints:

$$\begin{aligned} \min_{w \in \mathbb{R}^d} \quad & C(w) \\ \text{s.t.} \quad & g_j(w) = 0, \forall j = 1, \dots, s. \end{aligned} \tag{14}$$

The augmented Lagrangian method formulates an unconstrained objective, involving the Lagrangian for the above problem, augmented with additional quadratic penalty terms that penalize violations in the equality constraints:

$$\mathcal{C}_\rho(w, \lambda) = C(w) + \sum_{j=1}^s \lambda_j g_j(w) + \frac{\rho}{2} \sum_{j=1}^s (g_j(w))^2,$$

where  $\lambda = [\lambda_1, \dots, \lambda_s]$  is a vector of Lagrange multipliers associated with the equality constraints, and  $\rho > 0$  is a parameter that controls the weight on the penalty terms for violating the constraints. The method then performs the following sequence of updates:

$$w^{t+1} \in \operatorname{argmin}_{w \in \mathbb{R}^d} \mathcal{C}_\rho(w, \lambda^t) \tag{15}$$

$$\lambda_j^{t+1} = \lambda_j^t + \rho g_j(w^{t+1}). \tag{16}$$

One can set the penalty parameter  $\rho$  to a very large value (i.e. set a high cost for violating the equality constraints), so that method converges to a (locally) optimal solution to the original constrained problem (14). However, in practice, this can lead to numerical issues in applying the solver updates. Alternatively, the theory shows that under some conditions on the iterates of the solver, any value of  $\rho$  above a certain threshold will take the solver close to a locally optimal solution to (14) (see e.g. Theorem 17.6 in [52]).

In our experiments, we apply the augmented Lagrangian method to solve neural network revenue optimization problems, where we implement the inner optimization within the solver updates using mini-batch stochastic subgradient descent. We find that even for small values of  $\rho$ , with sufficient number of iterations, the solver converges to auction designs that yield near-optimal revenue while closely satisfying the regret constraints (see experimental results in Sections 3).

Finally, we point out that the described method can also be applied to optimization problems with inequality constraints  $h_j(w) \leq 0$  by formulating equivalent equality constraints of the form  $\max\{0, h_j(w)\} = 0$ .



## B MyersonNet for Single-item Auctions

We consider a setting with a single item to be sold, i.e.  $m = 1$ , and each bidder holds a private value  $v_i \in \mathbb{R}_{\geq 0}$  for the item. We consider a randomized mechanism  $(g, p)$  that maps a reported bid profile  $b \in \mathbb{R}_{\geq 0}^n$  to a vector of allocation probabilities  $g(b) \in \mathbb{R}_{\geq 0}^n$ , where  $g_i(b) \in \mathbb{R}_{\geq 0}$  denotes the probability that bidder  $i$  is allocated the item and  $\sum_{i=1}^n g_i(b) \leq 1$ . We shall represent the payment rule  $p_i$  via a price conditioned on the item being allocated to bidder  $i$ , i.e.  $p_i(b) = g_i(b) t_i(b)$  for some conditional payment function  $t_i : \mathbb{R}_{\geq 0}^n \rightarrow \mathbb{R}_{\geq 0}$ . The expected revenue of the mechanism, when bidders are truthful, is given by:

$$\text{rev}(g, p) = \mathbf{E}_{v \sim F} \left[ \sum_{i=1}^n g_i(v) t_i(v) \right]. \quad (17)$$

The structure of the revenue-optimal auction is well understood for this setting.

**Theorem 5** (Myerson [37]). *There exist a collection of monotonically increasing functions  $\bar{\phi}_i : V_i \rightarrow \mathbb{R}$  called the ironed virtual valuation functions such that the optimal BIC mechanism for selling a single item is the DSIC mechanism that assigns the item to the buyer  $i$  with the highest ironed virtual value  $\bar{\phi}_i(v_i)$  assuming this quantity is positive and charges the winning bidder the smallest bid that ensures that the bidder is winning.*

From Myerson's characterization, the optimal auction for regular distributions is deterministic and can be described by a set of strictly monotone virtual value transformations  $\bar{\phi}_1, \dots, \bar{\phi}_n : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ . The auction can be viewed as applying the monotone transformations to the input bids  $\bar{b}_i = \bar{\phi}_i(b_i)$ , feeding the computed virtual values to a second price auction (SPA) with zero reserve price  $(g^0, p^0)$ , making an allocation according to  $g^0(\bar{b})$ , and charging a payment  $\bar{\phi}_i^{-1}(p_i^0(\bar{b}))$  for agent  $i$ . In fact, this auction is DSIC for any choice of the strictly monotone virtual value functions:

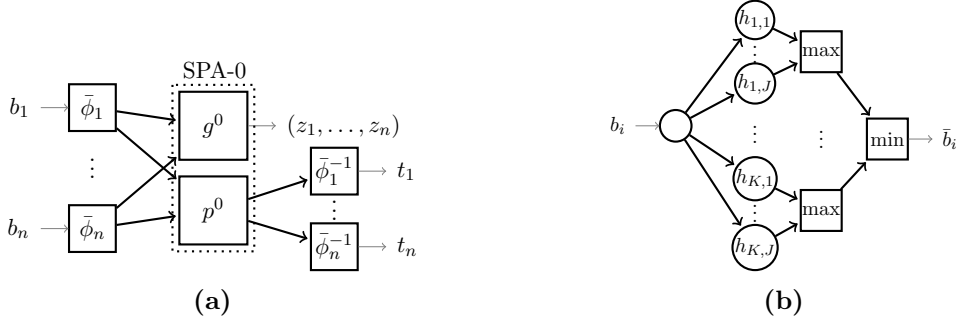
**Theorem 6.** *For any set of strictly monotonically increasing functions  $\bar{\phi}_1, \dots, \bar{\phi}_n : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ , an auction defined by outcome rule  $g_i = g_i^0 \circ \bar{\phi}$  and payment rule  $p_i = \bar{\phi}_i^{-1} \circ p_i^0 \circ \bar{\phi}$  is DSIC and IR.*

Thus designing the optimal DSIC auction for a regular distribution reduces to finding a set of strictly monotone virtual value functions that, when composed with the second price auction with zero reserve, yields maximum expected revenue. In the case of irregular distributions, the optimal mechanism is characterized by ironed virtual value transformations, which need not be *strictly* monotone or invertible. Hence the prescribed template of using strictly monotone transforms in conjunction with a SPA with zero reserve may not exactly recover the optimal mechanism. We shall see that the proposed approach can still be used to design mechanisms that yield revenue very close to the optimal revenue in the irregular case. See Figure 10(a) for the overall design of the neural network in this setting.

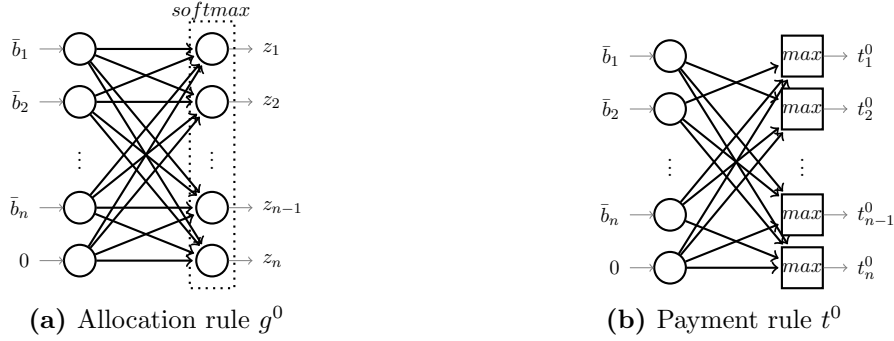
*Modeling monotone transforms.* We model each virtual value function  $\bar{\phi}_i$  as a two-layer feed-forward network with min and max operations over linear functions. For  $K$  groups of  $J$  linear functions, with strictly positive slopes  $w_{kj}^i \in \mathbb{R}_{>0}$ ,  $k = 1, \dots, K$ ,  $j = 1, \dots, J$  and intercepts  $\beta_{kj}^i \in \mathbb{R}$ ,  $k = 1, \dots, K$ ,  $j = 1, \dots, J$ , we define:

$$\bar{\phi}_i(b_i) = \min_{k \in [K]} \max_{j \in [J]} w_{kj}^i b_i + \beta_{kj}^i.$$

Since each of the above linear function is strictly non-decreasing, so is  $\bar{\phi}_i$ . In practice, we can set each  $w_{kj}^i = e^{\alpha_{kj}^i}$  for parameters  $\alpha_{kj}^i \in [-B, B]$  in a bounded range. A graphical representation of



**Figure 10:** (a) MyersonNet: The network applies monotone transformations  $\bar{\phi}_1, \dots, \bar{\phi}_n$  to the input bids, passes the virtual values to the SPA-0 network in Figure 11, and applies the inverse transformations  $\bar{\phi}_1^{-1}, \dots, \bar{\phi}_n^{-1}$  to the payment outputs; (b) Monotone virtual value function  $\bar{\phi}_i$ , where  $h_{kj}(b_i) = e^{\alpha_{kj}^i b_i} + \beta_{kj}^i$ .



**Figure 11:** SPA-0 network for (approximately) modeling a second price auction with zero reserve price. The inputs are (virtual) bids  $\bar{b}_1, \dots, \bar{b}_n$  and the output is a vector of assignment probabilities  $z_1, \dots, z_n$  and prices (conditioned on allocation)  $t_1^0, \dots, t_n^0$ .

the neural network used for this transform is shown in Figure 10(b). For sufficiently large  $K$  and  $J$ , this neural network can be used to approximate any continuous, bounded monotone function (that satisfies a mild regularity condition) to an arbitrary degree of accuracy [48]. A particular advantage of this representation is that the inverse transform  $\bar{\phi}^{-1}$  can be directly obtained from the parameters for the forward transform:

$$\bar{\phi}_i^{-1}(y) = \max_{k \in [K]} \min_{j \in [J]} e^{-\alpha_{kj}^i} (y - \beta_{kj}^i).$$

*Modeling SPA with zero reserve.* We also need to model a SPA with zero reserve (SPA-0) within the neural network structure. A neural network is usually a continuous function of its inputs, so that its parameters can be optimized efficiently. Since the allocation rule here is a discrete mapping (from bidder bids to the winning bidder), for the purpose of training, we employ a smooth approximation to the allocation rule using a neural network. Once we obtain the optimal virtual value functions using the approximate allocation rule, we use them in conjunction with an exact SPA with zero reserve, to construct the final mechanism.

The SPA-0 allocation rule  $g^0$  allocates the item to the bidder with the highest virtual value, if the virtual value is greater than 0, and leaves the item unallocated otherwise. This can be approximated using a ‘softmax’ function on the virtual values  $\bar{b}_1, \dots, \bar{b}_n$  and an additional dummy

Distribution	$n$	Opt <i>rev</i>	SPA <i>rev</i>	MyersonNet <i>rev</i>
Symmetric Uniform: $v_i \sim U[0, 1]$	3	0.531	0.500	0.531
Asymmetric Uniform: $v_i \sim U[0, i]$	5	2.314	2.025	2.305
Exponential: $v_i \sim \text{Exp}(3)$	3	2.749	2.500	2.747
Irregular: $v_i \sim F_{\text{irregular}}$	3	2.368	2.210	2.355

**Table 3:** The revenue of the single-item auctions obtained with RegretNet.

input  $\bar{b}_{n+1} = 0$ :

$$g_i^0(\bar{b}) = \frac{e^{\kappa \bar{b}_i}}{\sum_{j=1}^{n+1} e^{\kappa \bar{b}_j}}, \quad i \in N, \quad (18)$$

where  $\kappa > 0$  is a constant fixed a priori, and determines the quality of the approximation. The higher the value of  $\kappa$ , the better is the approximation, but the less smooth is the resulting allocation function (and thus it becomes harder to optimize).

The SPA-0 payment to bidder  $i$  (conditioned on being allocated) is the maximum of the virtual values from the other bidders, and zero:

$$t_i^0(\bar{b}) = \max \left\{ \max_{j \neq i} \bar{b}_j, 0 \right\}, \quad i \in N. \quad (19)$$

Let  $g^{\alpha, \beta}$  and  $t^{\alpha, \beta}$  denote the allocation and conditional payment rules for the overall mechanism in Figure 10(a), where  $(\alpha, \beta)$  are the parameters of the forward monotone transform. Given a sample of valuation profiles  $S = \{v^{(1)}, \dots, v^{(L)}\}$  drawn i.i.d. from  $F$ , we optimize the parameters using the (negative) revenue on  $S$  as the error function:

$$\widehat{\text{rev}}(f, p) = \frac{1}{L} \sum_{\ell=1}^L \sum_{i=1}^n g_i^{\alpha, \beta}(v^{(\ell)}) t_i^{\alpha, \beta}(v^{(\ell)}). \quad (20)$$

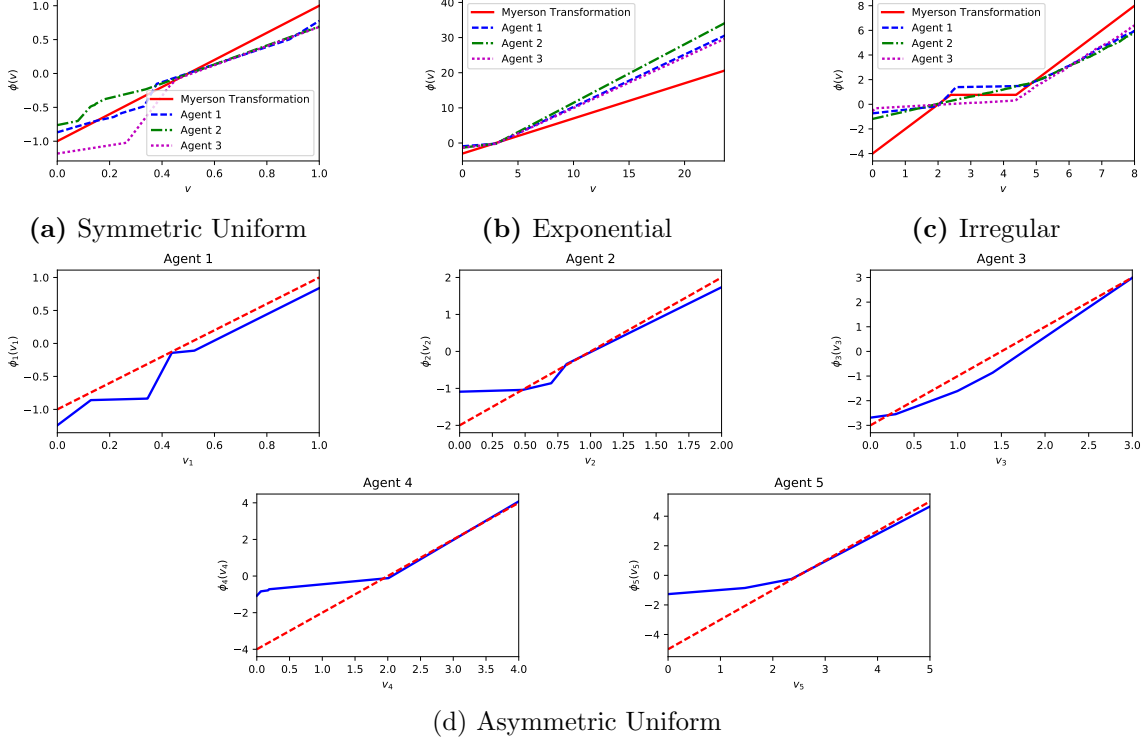
We solve this optimization problem using a mini-batch stochastic gradient descent solver.

## B.1 Experimental Results for MyersonNet

We evaluate the MyersonNet for designing single-item auctions on three regular distributions: (a) symmetric uniform distribution with 3 bidders and each  $v_i \sim U[0, 1]$ , (b) asymmetric uniform distribution with 5 bidders and each  $v_i \sim U[0, i]$ , and (c) exponential distribution with 3 bidders and each  $v_i \sim \text{Exp}(3)$ . We study auctions with a small number of bidders because this is where revenue-optimal auctions are meaningfully different from efficient auctions. The optimal auctions for these distributions involve virtual value functions  $\bar{\phi}_i$  that are strictly monotone. We also consider an irregular distribution  $F_{\text{irregular}}$ , where each  $v_i$  is drawn from  $U[0, 3]$  with probability 3/4 and from  $U[3, 8]$  with probability 1/4. In this case, the optimal auction uses ironed virtual value functions that are not strictly monotone. The training set and test set each have 1000 valuation profiles, sampled i.i.d. from the respective valuation distribution. We model each transform  $\bar{\phi}_i$  in the MyersonNet architecture using 5 sets of 10 linear functions, and set  $\kappa = 10^3$ .

The results are summarized in Table 3. For comparison, we also report the revenue obtained by the optimal Myerson auction and the second price auction (SPA) without reserve. The auctions learned by the neural network yield revenue close to the optimal.

The monotone virtual value transform inferred by the neural network for each bidder is shown in Figure 12. It can be verified that two auctions with virtual value transforms  $\bar{\phi}_1, \dots, \bar{\phi}_n$  and



**Figure 12:** Virtual value transformations learned by MyersonNet for each agent and of the optimal auction. The zero intercepts of most of the learned transforms are close to that of the optimal transforms. In many cases, the slopes of the learned transforms above the zero intercept are roughly a positive scalar multiple of the optimal transform, with the scalar being almost the same across agents.

$\bar{\phi}'_1, \dots, \bar{\phi}'_n$  are equivalent if for each bidder  $i$ ,  $\bar{\phi}_i$  and  $\bar{\phi}'_i$  have the same zero intercept, and there exists a scalar  $\gamma > 0$  such that for each bidder  $i$ ,  $\bar{\phi}_i(x) = \gamma \bar{\phi}'_i(x)$  for all  $x$  above the zero intercept. With the symmetric uniform and exponential distributions, the learned transform for each bidder has the same zero intercept as the optimal transform for the bidder. Also, the learned transforms have very similar slopes above the zero intercept. This is also seen with most agents in the asymmetric uniform setting, where the learned transform above the zero intercept has almost the same slope as the optimal transform. This indicates that the learned auctions for these distributions closely resemble the optimal auctions.

With the irregular distribution, the learned transform for each bidder has almost the same zero intercept as the optimal transform, but the learned transforms above the zero intercept are not all the same scalar multiple of the optimal transform. However notice that the learned transforms closely mimic the ironed flat portion of the optimal transform (albeit with a small non-zero slope because of the strictness imposed on the transforms by the network architecture). The slight mismatch between the learned and optimal transforms possibly explains why the revenue of the learned auction is not as close to the optimal revenue as that for the other distributions.

## C Discussion of LP and Additional Experiments

In this section, we show more discussion of LP and additional experiments.

Distretization	Number of decision variables	Number of constraints
5 bins/value	$1.25 \times 10^5$	$3.91 \times 10^6$
6 bins/value	$3.73 \times 10^5$	$2.02 \times 10^7$
7 bins/value	$9.41 \times 10^5$	$8.07 \times 10^7$

**Table 4:** Number of decision variables and constraints of LP with different discretizations for a 2 bidder, 3 items setting with uniform valuations.

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

**Table 5:** Revenue of auctions for 2 unit-demand bidders, 2 items obtained with RegretNet (Algorithm 1). For the ascending auction, the price were raised in units of 0.3 (which was empirically tuned using a grid search.)

**Discussion on size of LP.** First, we provide more evidence about the efficiency of our RegretNet compared with LP. As mentioned in [12], 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  $n \cdot (D^m - 1)$  IC constraints (for each bidder, there are  $(D^m - 1)$  constraints) for each valuation profile. In addition, there are  $n$  bidder-wise and  $m$  item-wise allocation constraints. In Table 4, 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 Experiments** We consider an additional experiment for 2 unit-demand bidders, 2 items as follows,

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

For this setting (a), the optimal auction is again not known; we show in Table 5 that the learned auctions beat the well-know ascending auction with near-optimal arising pricing unit (tuned by grid-search).

## D Omitted Proofs

### D.1 Proof of Lemma 1 and 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  $c \in \mathbb{R}$ ).  $\square$

*Proof of Lemma 2.* Similar to Lemma 1,  $\varphi^{CF}(s, s^{(1)}, \dots, s^{(m)})$  trivially satisfies the combinatorial feasibility (constraints (4)–(5)). For any allocation  $z$  that satisfies the combinatorial feasibility, the following scores

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

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

## D.2 Proof of Theorem 1

We present the proof for auctions with general, randomized allocation rules. A randomized allocation rule  $g_i : V \rightarrow [0, 1]^{2^M}$  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  $\sum_{S \subseteq M} g_{i,S}(v) \leq 1$ . 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  $p : V \rightarrow \mathbb{R}^n$  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.

**Notations.** 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  $\|A\|_1 = \max_{1 \leq j \leq \ell} \sum_{i=1}^k A_{ij}$ .

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

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

and let  $\mathcal{U}$  be the class of profile of utility functions defined on  $\mathcal{M}$ , i.e. the class of tuples  $(u_1, \dots, u_n)$  where each  $u_i : V_i \times V \rightarrow \mathbb{R}$  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  $\text{rgt} \circ \mathcal{U}_i$  be the class of all regret functions for bidder  $i$  defined on utility functions in  $\mathcal{U}_i$ :

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

and as before, let  $\text{rgt} \circ \mathcal{U}$  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  $\epsilon$  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  $\epsilon$  to cover  $\mathcal{U}_i$  under this distance. Similarly, we define covering numbers  $\mathcal{N}_{\infty}(\text{rgt} \circ \mathcal{U}_i, \epsilon)$  and  $\mathcal{N}_{\infty}(\text{rgt} \circ \mathcal{U}, \epsilon)$  for the function classes  $\text{rgt} \circ \mathcal{U}_i$  and  $\text{rgt} \circ \mathcal{U}$  respectively.

Moreover, we denote the class of allocation functions as  $\mathcal{G}$  and for each bidder  $i$ ,  $\mathcal{G}_i = \{g_i : V \rightarrow 2^M \mid g \in \mathcal{G}\}$ . Similarly, we denote the class of payment functions by  $\mathcal{P}$  and  $\mathcal{P}_i = \{p_i : V \rightarrow \mathbb{R} \mid p \in \mathcal{P}\}$ . 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  $\ell_{\infty}$  distance.

We would find it useful to first state the following lemma from [46]. Let  $\mathcal{F}$  be a class of functions  $f : Z \rightarrow [-c, c]$  for some input space  $Z$  and  $c > 0$ . Given a sample  $\mathcal{S} = \{z_1, \dots, z_L\}$  of points from  $Z$ , define the empirical Rademacher Complexity of  $\mathcal{F}$  as:

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

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

**Lemma 3** (Generalization bound in terms of Rademacher complexity). *Let  $\mathcal{S} = \{z_1, \dots, z_L\}$  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.

### D.2.1 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  $\mathcal{M}$ :

$$\text{rev} \circ \mathcal{M} = \{f : V \rightarrow \mathbb{R} \mid f(v) = \sum_{i=1}^n p_i(v), \text{ for some } (g, p) \in \mathcal{M}\}.$$

and bound the Rademacher complexity term for this class first 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 auction 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  $\epsilon \in (0, 1)$ ,

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

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  $1 - \delta$ ,

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

This completes the proof for the first part.  $\square$

### D.2.2 Generalization Bound for Regret

*Proof of Theorem 1 (Part 2).*

We move to the second part, namely a generalization bound for regret, which is the more challenging part of the proof. We first define the class of sum regret functions:

$$\overline{\text{rgt}} \circ \mathcal{U} = \left\{ f : V \rightarrow \mathbb{R} \mid f(v) = \sum_{i=1}^n r_i(v) \text{ for some } (r_1, \dots, r_n) \in \text{rgt} \circ \mathcal{U} \right\}.$$

The proof then proceeds in three steps:

- (1) bounding the covering number for each regret class  $\text{rgt} \circ \mathcal{U}_i$  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  $\mathcal{M}$
- (3) bounding the covering number for the sum regret class  $\overline{\text{rgt}} \circ \mathcal{U}$  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.**  $\mathcal{N}_\infty(\text{rgt} \circ \mathcal{U}_i, \epsilon) \leq \mathcal{N}_\infty(\mathcal{U}_i, \epsilon/2)$ .

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  $v$ ,

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

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

$$\begin{aligned} \max_{v'_i} u_i(v_i, (v'_i, v_{-i})) &= u_i(v_i^*, v_{-i}) \leq \hat{u}_i(v_i^*, v_{-i}) + \epsilon/2 \leq \hat{u}_i(\hat{v}_i^*, v_{-i}) + \epsilon/2 = \max_{\bar{v}_i} \hat{u}_i(v_i, (\bar{v}_i, v_{-i})) + \epsilon/2 \\ \max_{\bar{v}_i} \hat{u}_i(v_i, (\bar{v}_i, v_{-i})) &= \hat{u}_i(\hat{v}_i^*, v_{-i}) \leq u_i(\hat{v}_i^*, v_{-i}) + \epsilon/2 \leq u_i(v_i^*, v_{-i}) + \epsilon/2 = \max_{v'_i} u_i(v_i, (v'_i, v_{-i})) + \epsilon/2 \end{aligned} \tag{21}$$



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  $v$

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

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

This completes the proof for Step 1.

**Step 2.**  $\mathcal{N}_\infty(\mathcal{U}, \epsilon) \leq \mathcal{N}_\infty(\mathcal{M}, \epsilon)$ , 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  $|\hat{\mathcal{M}}| \leq \mathcal{N}_\infty(\mathcal{M}, \epsilon)$  such that, there exists  $(\hat{g}, \hat{p}) \in \hat{\mathcal{M}}$  s.t.

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

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

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

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

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

This completes the proof for Step 2.

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

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  $v, v' \in V$ ,  $\sum_i |u_i(v_i, (v'_i, v_{-i})) - \hat{u}_i(v_i, (v'_i, v_{-i}))| \leq \epsilon$ . Therefore for all  $v \in V$ ,  $|\sum_i u_i(v_i, (v'_i, v_{-i})) - \sum_i \hat{u}_i(v_i, (v'_i, v_{-i}))| \leq \epsilon$ , from which it follows that  $\mathcal{N}_\infty(\overline{\text{rgt}} \circ \mathcal{U}, \epsilon) \leq \mathcal{N}_\infty(\text{rgt} \circ \mathcal{U}, \epsilon)$ . Following Step 1, it is easy to show  $\mathcal{N}_\infty(\text{rgt} \circ \mathcal{U}, \epsilon) \leq \mathcal{N}_\infty(\mathcal{U}, \epsilon/2)$ . This further with Step 2 completes the proof of Step 3.

Based on the same arguments in Section D.2.1, the empirical Rademacher Complexity is bounded as:

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

Applying Lemma 3, completes the proof for generalization bound for regret.  $\square$

### D.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 2.

**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  $\ell$  containing  $T_\ell$  nodes and computing  $z \mapsto \phi_\ell(w^\ell z)$ , where each  $w^\ell \in \mathbb{R}^{T_\ell \times T_{\ell-1}}$  and  $\phi_\ell : \mathbb{R}^{T_\ell} \rightarrow [-B, +B]^{T_\ell}$ . Further let, for each network in  $\mathcal{F}_k$ , let the parameter matrices  $\|w^\ell\|_1 \leq W$  and  $\|\phi_\ell(s) - \phi_\ell(s')\|_1 \leq \Phi\|s - s'\|_1$  for any  $s, s' \in \mathbb{R}^{T_{\ell-1}}$ .*

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

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  $\epsilon_0/d$ , 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  $\{-(\lceil Wd/\epsilon_0 \rceil - 1)\epsilon_0/d, \dots, -\epsilon_0/d, 0, \epsilon_0/d, \dots, \lceil Wd/\epsilon_0 \rceil \epsilon_0/d\}$ . 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  $\epsilon$ -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  $\{-(\lceil Wd/\epsilon_0 \rceil - 1)\epsilon_0/d, \dots, -\epsilon_0/d, 0, \epsilon_0/d, \dots, \lceil Wd/\epsilon_0 \rceil \epsilon_0/d\}$ . 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.

$$\|w_\ell - \hat{w}_\ell\|_1 = \max_{1 \leq j \leq T_{\ell-1}} \sum_{i=1}^{T_\ell} |w_{\ell,i,j}^\ell - \hat{w}_{\ell,i,j}^\ell| \leq T_\ell \epsilon_0/d \leq \epsilon_0. \quad (22)$$

We then have:

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

where the second line follows from our assumption on  $\phi_{k+1}$ , and the sixth line follows from our inductive hypothesis and from (22). By choosing  $\epsilon_0 = \frac{\epsilon}{B(2\Phi W)^k}$ , we complete the proof.  $\square$

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  $\alpha : \mathbb{R}^{m(n+1)} \rightarrow [0, 1]^n$  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  $\ell_\infty$  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  $\ell_\infty$  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

$$\sup_b \sum_{i,j} |g_{ij}(b) - \hat{g}_{ij}(b)| \leq \epsilon/3 \quad (23)$$

$$\sup_b \sum_i |\alpha_i(b) - \hat{\alpha}_i(b)| \leq \epsilon/3. \quad (24)$$

We now show that the class of mechanism  $\hat{\mathcal{M}} = \{(\hat{g}, \hat{\alpha}) \mid \hat{g} \in \hat{\mathcal{G}}, \text{ and } \hat{p}(b) = \hat{\alpha}_i(b) \cdot \langle v_i, \hat{g}_i(b) \rangle\}$  is an  $\epsilon$ -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 (24). We have:

$$\begin{aligned} & \sum_{i,j} |g_{ij}(b) - \hat{g}_{ij}(b)| + \sum_i |p_i(b) - \hat{p}_i(b)| \\ & \leq \epsilon/3 + \sum_i |\alpha_i(b) \cdot \langle b_i, g_i(b) \rangle - \hat{\alpha}_i(b) \cdot \langle b_i, \hat{g}_i(b) \rangle| \\ & \leq \epsilon/3 + \sum_i |(\alpha_i(b) - \hat{\alpha}_i(b)) \cdot \langle b_i, g_i(b) \rangle| + |\hat{\alpha}_i(b) \cdot (\langle b_i, g_i(b) \rangle - \langle b_i, \hat{g}_i(b) \rangle)| \\ & \leq \epsilon/3 + \sum_i |\alpha_i(b) - \hat{\alpha}_i(b)| + \sum_i \|b_i\|_\infty \cdot \|g_i(b) - \hat{g}_i(b)\|_1 \\ & \leq 2\epsilon/3 + \sum_{i,j} |g_{ij}(b) - \hat{g}_{ij}(b)| \leq \epsilon, \end{aligned}$$

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.  $\square$

We are now ready to prove covering number bounds for the three architectures in Section 2.

*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  $\ell = 1, \dots, R+1$ ,  $\|w_\ell\|_1 \leq W$ . 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  $\phi_{R+1} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  for the final layer for input  $s \in \mathbb{R}^{n \times m}$  can be described as:  $\phi_{R+1}(s) = [\text{softmax}(s_{1,1}, \dots, s_{n,1}), \dots, \text{softmax}(s_{1,m}, \dots, s_{n,m})]$ , where  $\text{softmax} : \mathbb{R}^n \rightarrow [0, 1]^n$  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  $s, s' \in \mathbb{R}^{n \times m}$ ,

$$\begin{aligned} \|\phi_{R+1}(s) - \phi_{R+1}(s')\|_1 &= \sum_j \|\text{softmax}(s_{1,j}, \dots, s_{n,j}) - \text{softmax}(s'_{1,j}, \dots, s'_{n,j})\|_1 \\ &\leq \sqrt{n} \sum_j \|\text{softmax}(s_{1,j}, \dots, s_{n,j}) - \text{softmax}(s'_{1,j}, \dots, s'_{n,j})\|_2 \\ &\leq \sqrt{n} \frac{\sqrt{n-1}}{n} \sum_j \sqrt{\sum_i \|s_{ij} - s'_{ij}\|^2} \\ &\leq \sum_j \sum_i |s_{ij} - s'_{ij}| \end{aligned} \tag{25}$$

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

The hidden layers  $\ell = 1, \dots, R$  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  $\ell = 1, \dots, R+1$ ,  $\|w_\ell\|_1 \leq W$ . 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  $\phi_{R+1} : \mathbb{R}^{2n} \rightarrow \mathbb{R}^n$  for the final layer for two sets of scores  $s, \bar{s} \in \mathbb{R}^{n \times m}$  can be described as:

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

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

$$\begin{aligned}
\|\phi_{R+1}(s, \tilde{s}) - \phi_{R+1}(s', \tilde{s}')\|_1 &= \sum_{i,j} \left| \min\{\text{softmax}_j(s_{i,1}, \dots, s_{i,m}), \text{softmax}_i(\tilde{s}_{1,j}, \dots, \tilde{s}_{n,j})\} \right. \\
&\quad \left. - \min\{\text{softmax}_j(s'_{i,1}, \dots, s'_{i,m}), \text{softmax}_i(\tilde{s}'_{1,j}, \dots, \tilde{s}'_{n,j})\} \right| \\
&\leq \sum_{i,j} \left| \max\{\text{softmax}_j(s_{i,1}, \dots, s_{i,m}) - \text{softmax}_j(s'_{i,1}, \dots, s'_{i,m}), \right. \\
&\quad \left. \text{softmax}_i(\tilde{s}_{1,j}, \dots, \tilde{s}_{n,j}) - \text{softmax}_i(\tilde{s}'_{1,j}, \dots, \tilde{s}'_{n,j})\} \right| \\
&\leq \sum_i \|\text{softmax}(s_{i,1}, \dots, s_{i,m}) - \text{softmax}(s'_{i,1}, \dots, s'_{i,m})\|_1 \\
&\quad + \sum_j \|\text{softmax}(\tilde{s}_{1,j}, \dots, \tilde{s}_{n,j}) - \text{softmax}(\tilde{s}'_{1,j}, \dots, \tilde{s}'_{n,j})\|_1 \\
&\leq \sum_{i,j} |s_{ij} - s'_{ij}| + \sum_{i,j} |\tilde{s}_{ij} - \tilde{s}'_{ij}|,
\end{aligned}$$

where the last step can be derived in the same way as (25).

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

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

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

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

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

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

$$\begin{aligned}
&\|\phi_{R+1}(s, s^{(1)}, \dots, s^{(m)}) - \phi_{R+1}(s', s'^{(1)}, \dots, s'^{(m)})\|_1 \\
&= \sum_{i,S} \left| \min \left\{ \text{softmax}_S(s_{i,S'} : S' \subseteq M), \text{softmax}_S(s_{i,S'}^{(1)} : S' \subseteq M), \dots, \text{softmax}_S(s_{i,S'}^{(m)} : S' \subseteq M) \right\} \right. \\
&\quad \left. - \min \left\{ \text{softmax}_S(s'_{i,S'} : S' \subseteq M), \text{softmax}_S(s'_{i,S'}^{(1)} : S' \subseteq M), \dots, \text{softmax}_S(s'_{i,S'}^{(m)} : S' \subseteq M) \right\} \right| \\
&\leq \sum_{i,S} \max \left\{ |\text{softmax}_S(s_{i,S'} : S' \subseteq M) - \text{softmax}_S(s'_{i,S'} : S' \subseteq M)|, \right. \\
&\quad |\text{softmax}_S(s_{i,S'}^{(1)} : S' \subseteq M) - \text{softmax}_S(s'_{i,S'}^{(1)} : S' \subseteq M)|, \dots \\
&\quad \left. |\text{softmax}_S(s_{i,S'}^{(m)} : S' \subseteq M) - \text{softmax}_S(s'_{i,S'}^{(m)} : S' \subseteq M)| \right\} \\
&\leq \sum_i \|\text{softmax}(s_{i,S'} : S' \subseteq M) - \text{softmax}(s'_{i,S'} : S' \subseteq M)\|_1 \\
&\quad + \sum_{i,j} \|\text{softmax}(s_{i,S'}^{(j)} : S' \subseteq M) - \text{softmax}(s'_{i,S'}^{(j)} : S' \subseteq M)\|_1 \\
&\leq \sum_{i,S} |s_{i,S} - s'_{i,S}| + \sum_{i,j,S} |s_{i,S}^{(j)} - s'_{i,S}^{(j)}|,
\end{aligned}$$

where the last step can be derived in the same way as (25).

As with additive bidders, using additionally hidden layers  $\ell = 1, \dots, R$  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.  $\square$

We now bound  $\Delta_L$  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  $\epsilon$ :

(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

$$\Delta_L \leq O \left( \sqrt{R(d_p + d_a) \frac{\log(W \max\{K, n \cdot 2^m\}L)}{L}} \right).$$

#### D.4 Proof of Theorem 4

*Proof for Theorem 4.* The convexity of  $u_{\alpha, \beta}$  follows from the fact it is a ‘max’ of linear functions. We now show that  $u_{\alpha, \beta}$  is monotonically non-decreasing. Let  $h_j(v) = w_j \cdot v + \beta_j$ . Since  $w_j$  is non-negative in all entries, for any  $v_i \leq v'_i, \forall i \in M$ , we have  $h_j(v) \leq h_j(v')$ . Then

$$u_{\alpha, \beta}(v) = \max_{j \in [J]} h_j(v) = h_{j_*}(v) \leq h_{j_*}(v') \leq \max_{j \in [J]} h_j(v') = u_{\alpha, \beta}(v'),$$

where  $j_* \in \operatorname{argmin}_{j \in [J]} h_j(v)$ . It remains to be shown that  $u_{\alpha, \beta}$  is 1-Lipschitz. For any  $v, v' \in \mathbb{R}_{\geq 0}^m$ ,

$$\begin{aligned} |u_{\alpha, \beta}(v) - u_{\alpha, \beta}(v')| &= |\max_{j \in [J]} h_j(v) - \max_{j \in [J]} h_j(v')| \leq \max_{j \in [J]} |h_j(v') - h_j(v)| \\ &= \max_{j \in [J]} |w_j \cdot (v' - v)| \leq \max_{j \in [J]} \|w_j\|_\infty |v' - v|_1 \leq |v'_k - v_k|_1 \end{aligned}$$

where the last inequality holds because each component  $w_{jk} = \sigma(\alpha_{jk}) \leq 1$ .  $\square$

---

**Algorithm 2** RegretNet Training for Sample-based Approximate Regret

---

**Input:** Minibatches  $\mathcal{S}_1, \dots, \mathcal{S}_T$  of size  $B$ , and misreports  $S_i^{(\ell)}$  for each  $i, \ell$   
**Parameters:**  $\forall t, \rho_t > 0, \eta > 0, R \in \mathbb{N}, Z \in \mathbb{N}$   
**Initialize:**  $w^0 \in \mathbb{R}^d, \lambda^0 \in \mathbb{R}^n$   
**for**  $t = 0$  **to**  $T$  **do**  
    Receive minibatch  $\mathcal{S}_t = \{u^{(1)}, \dots, u^{(B)}\}$   
    Compute regret gradient:  $\forall \ell \in [B], i \in N$ :  
         $g_{\ell,i}^t = \nabla_w [\max_{v'_i \in S_i^{(\ell)}} u_i^w(v_i^{(\ell)}; (v'_i, v_{-i}^{(\ell)})) - u_i^w(v_i^{(\ell)}; v^{(\ell)})] \Big|_{w=w^t}$   
    Compute Lagrangian gradient using (6) 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  $Z$  iterations:  
        **if**  $t$  is a multiple of  $Z$   
             $\lambda_i^{t+1} \leftarrow \lambda_i^t + \rho_t \widetilde{rgt}_i(w^{t+1}), \quad \forall i \in N$   
        **else**  
             $\lambda^{t+1} \leftarrow \lambda^t$   
**end for**

---

## E Sample-based Approach for Optimization in RegretNet

In this section, we show the sample-based approach for computing a ‘max’ over misreports. In this solver, we compute the empirical regret approximately by replacing the max over misreports in the definition by a max over a sample of misreport profiles drawn uniformly from each  $V_i$ . Given a sample of  $L$  valuation profiles  $S$  drawn i.i.d. from  $F$ , and a sample of misreport profiles  $S_i^{(\ell)}$  drawn uniformly from  $V_i$  for each bidder  $i \in N$  and each profile  $v^{(\ell)} \in S$ , we define the approximate empirical regret for bidder  $i$  as:

$$\widetilde{rgt}_i(w) = \frac{1}{L} \sum_{\ell=1}^L \max_{v'_i \in S_i^{(\ell)}} \left( u_i^w(v_i^{(\ell)}; (v'_i, v_{-i}^{(\ell)})) - u_i^w(v_i^{(\ell)}; v^{(\ell)}) \right), \quad (26)$$

where  $u_i^w(v_i, b) = v_i(g_i^w(b)) - p_i^w(b)$ .

The solver is shown in Algorithm 2. We give the experimental results from using this solver, and compare them with the results from using the gradient-based approach for misreports, as provided in Section 3. In addition, we extend the generalization bound in Theorem 1 to this approximate empirical regret in Section E.2.

### E.1 Experiments

**Setup and Evaluation.** For all the experiments in this section, we generate 100 misreports from a uniform distribution for each valuation profile (i.e.  $|S_i^{(\ell)}| = 100, \forall i, \ell$ ) in the training data. All the other setups and evaluation are the same as in the experiments that make use of the gradient-based approach to misreports (Section 3.1).

**Experimental Results** First, we consider the small settings, which are the same as in Section 3 (settings (I) - (VIII)) and report the revenue and regret of the final auctions learned for these settings on the test set, adopting an architecture two hidden layers and 100 nodes per layer. The test results are summarized in Table 6 and Table 7. We find RegretNet using this sample-based

Distribution	Opt <i>irp</i>	RegretNet (Algorithm 2)	
		<i>rev</i>	<i>rgt</i>
Setting (I): One additive bidder, $v_1, v_2 \sim U[0, 1]$	0.550	0.557	< 0.001
Setting (II): One additive bidder, $v_1 \sim [4, 16], v_2 \sim U[4, 7]$	9.781	9.722	< 0.003
Setting (III): One additive bidder, $v_1, v_2$ drawn uniformly from a unit triangle	0.388	0.392	< 0.001
Setting (IV): One unit-demand bidder, $v_1, v_2 \sim U[0, 1]$	0.384	0.386	< 0.001
Setting (V): One unit-demand bidder, $v_1, v_2 \sim U[2, 3]$	2.137	2.124	< 0.001

**Table 6:** Test revenue and regret for settings (I) - (V), learned by RegretNet using sample-based approach to approximate the max over misreports. The optimal mechanisms for these settings are known.

Distribution	<i>rev</i>	<i>rgt</i>	VVCA	AMA <sub>bsym</sub>
Setting (VI)	<b>0.880</b>	< 0.001	0.860	0.862
Setting (VII)	<b>2.862</b>	< 0.001	2.741	2.765
Setting (VIII)	<b>4.236</b>	< 0.001	4.209	3.748

**Table 7:** Test revenue and regret for settings (VI) - (VIII), learned by RegretNet using sample-based approach to approximate the max over misreports. The state-of-the-art computational results is provided by Sandholm and Likhodedov [45] and we compare our RegretNet with the two optimal auctions from the VVCA and AMA<sub>bsym</sub> families from [45].

approach for regret (Algorithm 2) achieves very comparable results to the gradient-based approach in these settings. However, the training time is a lot smaller, with Algorithm 2 running in less than two hours for these settings, giving a  $4\times$  speed up compared with Algorithm 1. From the visualizations of the learned allocation rules for settings (I) - (V), Figure 13, we can see this alternate approach continues to closely recover the optimal structure.

On the other hand, for larger settings (e.g. settings (IX) - (XI)), the regret attained by Algorithm 2 is much higher than the regret attained by Algorithm 1 (with the gradient-based approach to regret). This is because we need more misreports to accurately approximate the max function in the regret for settings with more items. Based on our generalization bound for the approximate empirical regret (Equation 26 in Section E.2), the number of misreports  $Q$  for each valuation profile needs to be exponentially large with the number of items to achieve good generalization error. Therefore, this sample-based RegretNet is hard to scale up. Still, the approach is more efficient in smaller settings, and can handle the setting that the utility function is not differentiable (e.g., settings with discrete values).

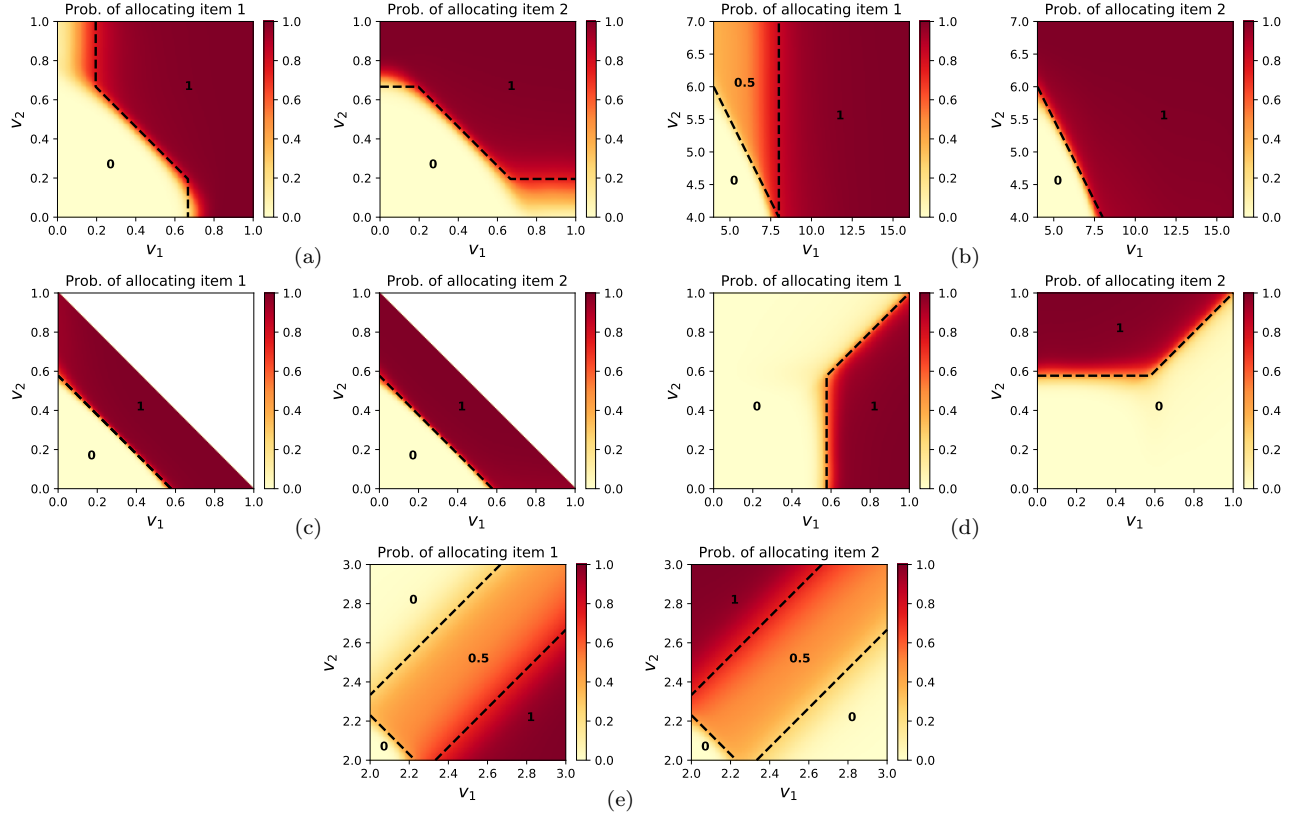
## E.2 Generalization Bound

We state a bound on the gap between the expected ex post regret and the approximate empirical regret in (26) in terms of the number of valuation profiles in  $S$  and the number of misreport profiles in  $S^{(\ell)}$ .

**Theorem 7.** *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  $(g^w, p^w)$  that satisfy individual rationality, where the allocation and payment functions satisfy the following Lipschitz property: for all  $i \in N, v'_i, v''_i \in V, v_{-i} \in V_{-i}, \|g_i^w(v'_i, v_{-i}) - g_i^w(v''_i, v_{-i})\|_1 \leq \mathcal{L}\|v'_i - v''_i\|_\infty$  and  $|p_i^w(v'_i, v_{-i}) - p_i^w(v''_i, v_{-i})| \leq \mathcal{L}\|v'_i - v''_i\|_\infty$ . Fix  $\delta \in (0, 1)$ . With probability at least  $1 - \delta$  over draw of sample  $S$  of  $L$  profiles from  $F$  and draw of each misreport sample  $S_i^{(\ell)}$  of size  $Q$  from a uniform distribution over  $V$ , for any  $(g^w, p^w) \in \mathcal{M}$ ,*

$$rgt_i(w) - \widetilde{rgt}_i(w) \leq 2 \inf_{\epsilon > 0} \left( \epsilon + 2 \left( \frac{2 \log(\mathcal{N}_\infty(\mathcal{M}, \epsilon/2))}{L} \right)^{1/2} + 2\mathcal{L} \left( \frac{\log(2nL\mathcal{N}_\infty(\mathcal{M}, \epsilon/4)/\delta)}{Q} \right)^{1/D} \right) + C \sqrt{\frac{\log(n/\delta)}{L}},$$





**Figure 13:** Allocation rules learned by RegretNet with Algorithm 2 as the solver, for single-bidder, two items settings: (a) Setting (I), (b) Setting (II), (c) Setting (III), (d) Setting (IV), and (e) Setting (V). 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.

where  $D = \max_{i \in N} \dim(V_i)$ ,  $\dim(V_i)$  is the dimension of each valuation in  $V_i$ , and  $C$  is a distribution-independent constant.

Upper bounding the ‘inf’ over  $\epsilon$  the same way we bounded the term  $\Delta_L$  in Theorem 2, we get a generalization bound in terms of the number of valuation profiles  $L$  in  $S$ , and the number of valuations  $Q$  in each misreport sample  $S_i^{(\ell)}$ .

### E.3 Proof of Theorem 7

We use the same notations as in Section D. As before, let  $\mathcal{U}_i$  be the class of utility functions for bidder  $i$  defined on auctions in  $\mathcal{M}$ , i.e.:

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

We will use  $U_i$  to denote a uniform distribution over  $V_i$ . For ease of exposition, we omit the subscript “ $w$ ” on the utility function.

For a fixed bidder, we show an uniform convergence result for a fixed bidder valuation profile in Lemma 6. We extend this result to a sample  $S$  of  $L$  valuation profiles drawn i.i.d. from distribution  $F$  in Lemma 7, where we bound the gap between the empirical regret where the ‘max’ over misreports is computed exactly, and the approximate empirical regret. Combining Lemma 7 and Theorem 1, followed by a union bound over all bidders completes the proof of Theorem 7.

**Lemma 6.** *For any fixed bidder  $i$ , assume for all  $v'_i \in V_i, v''_i \in V_i, v \in V$  and  $u_i \in \mathcal{U}_i$ ,  $|u_i(v_i, (v'_i, v_{-i})) - u_i(v_i, (v''_i, v_{-i}))| \leq \mathcal{L} \|v'_i - v''_i\|_\infty$ . Let  $S_i$  be a sample of  $Q$  misreport generated uniformly over  $V_i$ . Then for any fixed  $v \in V$  and any  $\epsilon > 0$ ,*

$$\begin{aligned} \mathbf{P}_{S_i \sim U_i^Q} \left( \sup_{u_i \in \mathcal{U}_i} \left| \max_{v'_i \in S_i} u_i(v_i, (v'_i, v_{-i})) - \max_{\bar{v}_i \in V_i} u_i(v_i, (\bar{v}_i, v_{-i})) \right| \geq \epsilon \right) \\ \leq \mathcal{N}_\infty \left( \mathcal{U}_i, \frac{\epsilon}{4} \right) \exp \left( -Q \left( \frac{\epsilon}{2\mathcal{L}} \right)^{D_i} \right), \end{aligned}$$

where  $D_i = \dim(V_i)$ .

*Proof.* By definition of covering number  $\mathcal{N}_\infty(\mathcal{U}_i, \epsilon)$ , there exists  $\hat{\mathcal{U}}_i$  with at most  $|\hat{\mathcal{U}}_i| \leq \mathcal{N}_\infty(\mathcal{U}_i, \frac{\epsilon}{4})$ , such that for any  $u_i \in \mathcal{U}_i$ , there exists a  $\hat{u}_i \in \hat{\mathcal{U}}_i$  with

$$\sup_{v, v'_i} |\hat{u}_i(v_i, (v'_i, v_{-i})) - u_i(v_i, (v'_i, v_{-i}))| \leq \frac{\epsilon}{4}$$

Then for any  $u_i \in \mathcal{U}_i$  such that

$$\left| \max_{v'_i \in S_i} u_i(v_i, (v'_i, v_{-i})) - \max_{\bar{v}_i \in V_i} u_i(v_i, (\bar{v}_i, v_{-i})) \right| \geq \epsilon \quad (27)$$

For any  $\hat{u}_i \in \hat{\mathcal{U}}_i$ , s.t.  $\sup_{v, v'_i} |\hat{u}_i(v_i, (v'_i, v_{-i})) - u_i(v_i, (v'_i, v_{-i}))| \leq \frac{\epsilon}{4}$ , we have

$$\begin{aligned} \epsilon &\leq \left| \max_{v'_i \in S_i} u_i(v_i, (v'_i, v_{-i})) - \max_{\bar{v}_i \in V_i} u_i(v_i, (\bar{v}_i, v_{-i})) \right| \\ &\leq \left| \max_{v'_i \in S_i} \hat{u}_i(v_i, (v'_i, v_{-i})) - \max_{\bar{v}_i \in V_i} \hat{u}_i(v_i, (\bar{v}_i, v_{-i})) \right| + \left| \max_{v'_i \in S_i} u_i(v_i, (v'_i, v_{-i})) - \max_{v'_i \in S_i} \hat{u}_i(v_i, (v'_i, v_{-i})) \right| \\ &\quad + \left| \max_{\bar{v}_i \in V_i} u_i(v_i, (\bar{v}_i, v_{-i})) - \max_{\bar{v}_i \in V_i} \hat{u}_i(v_i, (\bar{v}_i, v_{-i})) \right| \\ &\leq \left| \max_{v'_i \in S_i} \hat{u}_i(v_i, (v'_i, v_{-i})) - \max_{\bar{v}_i \in V_i} \hat{u}_i(v_i, (\bar{v}_i, v_{-i})) \right| + \frac{\epsilon}{2}, \end{aligned}$$

where the last step can be derived the same way as (21).

Thus we can claim for any  $u_i \in \mathcal{U}_i$  be such that 27 holds, there exists a  $\hat{u}_i \in \hat{\mathcal{U}}_i$  such that  $|\max_{v'_i \in S_i} \hat{u}_i(v_i, (v'_i, v_{-i})) - \max_{\bar{v}_i \in V_i} \hat{u}_i(v_i, (\bar{v}_i, v_{-i}))| \geq \frac{\epsilon}{2}$ , then we have

$$\begin{aligned}
& \mathbf{P}_{S_i \sim U_i^Q} \left( \sup_{u_i \in \mathcal{U}_i} \left| \max_{v'_i \in S_i} u_i(v_i, (v'_i, v_{-i})) - \max_{\bar{v}_i \in V_i} u_i(v_i, (\bar{v}_i, v_{-i})) \right| \geq \epsilon \right) \\
& \leq \mathbf{P}_{S_i \sim U_i^Q} \left( \max_{\hat{u}_i \in \hat{\mathcal{U}}_i} \left| \max_{v'_i \in S_i} \hat{u}_i(v_i, (v'_i, v_{-i})) - \max_{\bar{v}_i \in V_i} \hat{u}_i(v_i, (\bar{v}_i, v_{-i})) \right| \geq \frac{\epsilon}{2} \right) \\
& \leq \sum_{\hat{u}_i \in \hat{\mathcal{U}}_i} \mathbf{P}_{S_i \sim U_i^Q} \left( \left| \max_{v'_i \in S_i} \hat{u}_i(v_i, (v'_i, v_{-i})) - \max_{\bar{v}_i \in V_i} \hat{u}_i(v_i, (\bar{v}_i, v_{-i})) \right| \geq \frac{\epsilon}{2} \right) \\
& \leq \mathcal{N}_\infty \left( \mathcal{U}_i, \frac{\epsilon}{4} \right) \max_{\hat{u}_i \in \hat{\mathcal{U}}_i} \mathbf{P}_{S_i \sim U_i^Q} \left( \left| \max_{v'_i \in S_i} \hat{u}_i(v_i, (v'_i, v_{-i})) - \max_{\bar{v}_i \in V_i} \hat{u}_i(v_i, (\bar{v}_i, v_{-i})) \right| \geq \frac{\epsilon}{2} \right)
\end{aligned}$$

For any  $\hat{u}_i \in \hat{\mathcal{U}}_i$ , let  $\bar{v}_i^* \in \arg \max_{v'_i \in V_i} \hat{u}_i(v_i, (\bar{v}_i, v_{-i}))$ , then we have

$$\begin{aligned}
& \mathbf{P}_{S_i \sim U_i^Q} \left( \left| \max_{v'_i \in S_i} \hat{u}_i(v_i, (v'_i, v_{-i})) - \max_{\bar{v}_i \in V_i} \hat{u}_i(v_i, (\bar{v}_i, v_{-i})) \right| \geq \frac{\epsilon}{2} \right) \\
& \leq \mathbf{P} \left( \text{There is no } v'_i \in S_i, \text{ such that } \|v'_i - \bar{v}_i^*\|_\infty \leq \frac{\epsilon}{2\mathcal{L}} \right) \\
& = \prod_{v'_i \in S_i} \mathbf{P} \left( \|v'_i - \bar{v}_i^*\|_\infty \geq \frac{\epsilon}{2\mathcal{L}} \right) \\
& \leq \prod_{v'_i \in S_i} \left( 1 - \mathbf{P}_{v'_{ij}} \left( \|v'_i - \bar{v}_i^*\|_\infty \leq \frac{\epsilon}{2\mathcal{L}} \right) \right) \\
& \leq \left( 1 - \left( \frac{\epsilon}{2\mathcal{L}} \right)^{D_i} \right)^Q \leq \exp \left( -Q \left( \frac{\epsilon}{2\mathcal{L}} \right)^{D_i} \right).
\end{aligned}$$

Thus we complete the proof.  $\square$

We extend Lemma 6 to a sample of  $L$  valuation profiles drawn i.i.d. from  $F$ .

**Lemma 7.** *For any fixed bidder  $i$ , assume for all  $v'_i \in V_i, v''_i \in V_i, v \in V$  and  $u_i \in \mathcal{U}_i$ ,  $|u_i(v_i, (v'_i, v_{-i})) - u_i(v_i, (v''_i, v_{-i}))| \leq \mathcal{L} \|v'_i - v''_i\|_\infty$ . Let  $S$  be a sample of  $L$  valuation profiles drawn i.i.d. from  $F$  and  $S_i^{(\ell)}$  be a sample of  $Q$  misreport values drawn uniformly over  $V_i$  for each valuation profile  $v^{(\ell)}$ . Then for any  $\epsilon > 0$ ,*

$$\begin{aligned}
& \mathbf{P} \left( \left| \frac{1}{L} \sum_{\ell=1}^L \max_{v'_i \in S_i^{(\ell)}} u_i \left( v_i^{(\ell)}, (v'_i, v_{-i}^{(\ell)}) \right) - \frac{1}{L} \sum_{\ell=1}^L \max_{\bar{v}_i \in V_i} u_i \left( v_i^{(\ell)}, (\bar{v}_i, v_{-i}^{(\ell)}) \right) \right| \geq \epsilon \right) \\
& \leq L \mathcal{N}_\infty \left( \mathcal{U}_i, \frac{\epsilon}{4} \right) \exp \left( -Q \left( \frac{\epsilon}{2\mathcal{L}} \right)^D \right),
\end{aligned}$$

where the probability is over random draws of  $S \sim F^L$  and  $S_i^{(\ell)} \sim U_i^Q$ , and  $D = \max_{i \in N} \dim(V_i)$ .

*Proof.*

$$\begin{aligned}
& \mathbf{P}_{S \sim F^L, S_i^{(\ell)} \sim U_i^Q, \ell=1, \dots, L} \left( \sup_{u_i \in \mathcal{U}_i} \left| \frac{1}{L} \sum_{\ell=1}^L \max_{v'_i \in S_i^{(\ell)}} u_i(v_i^{(\ell)}, (v'_i, v_{-i}^{(\ell)})) - \frac{1}{L} \sum_{\ell=1}^L \max_{\bar{v}_i \in V_i} u_i(v_i^{(\ell)}, (\bar{v}_i, v_{-i}^{(\ell)})) \right| \geq \epsilon \right) \\
& \leq \mathbf{P}_{S \sim F^L, S_i^{(\ell)} \sim U_i^Q, \ell=1, \dots, L} \left( \frac{1}{L} \sum_{\ell=1}^L \sup_{u_i \in \mathcal{U}_i} \left| \max_{v'_i \in S_i^{(\ell)}} u_i(v_i^{(\ell)}, (v'_i, v_{-i}^{(\ell)})) - \max_{\bar{v}_i \in V_i} u_i(v_i^{(\ell)}, (\bar{v}_i, v_{-i}^{(\ell)})) \right| \geq \epsilon \right) \\
& \leq \mathbf{P}_{S \sim F^L, S_i^{(\ell)} \sim U_i^Q, \ell=1, \dots, L} \left( \exists \ell \text{ s.t. } \sup_{u_i} \left| \max_{v'_i \in S_i^{(\ell)}} u_i(v_i^{(\ell)}, (v'_i, v_{-i}^{(\ell)})) - \max_{\bar{v}_i \in V_i} u_i(v_i^{(\ell)}, (\bar{v}_i, v_{-i}^{(\ell)})) \right| \geq \epsilon \right) \\
& \leq L \mathbf{P}_{v \sim F, S_i \sim U_i^Q} \left( \sup_{u_i \in \mathcal{U}_i} \left| \max_{v'_i \in S_i} u_i(v_i, (v'_i, v_{-i})) - \max_{\bar{v}_i \in V_i} u_i(v_i, (\bar{v}_i, v_{-i})) \right| \geq \epsilon \right) \\
& \leq L \sup_{v \in V} \mathbf{P}_{S_i \sim U_i^Q} \left( \sup_{u_i \in \mathcal{U}_i} \left| \max_{v'_i \in S_i} u_i(v_i, (v'_i, v_{-i})) - \max_{\bar{v}_i \in V_i} u_i(v_i, (\bar{v}_i, v_{-i})) \right| \geq \epsilon \right) \\
& \leq L \mathcal{N}_\infty \left( \mathcal{U}_i, \frac{\epsilon}{4} \right) \exp \left( -Q \left( \frac{\epsilon}{2\mathcal{L}} \right)^D \right),
\end{aligned}$$

where in the third inequality, we take a union bound over all profiles in  $S$  and the last inequality follows from Lemma 6.  $\square$

*Proof of Theorem 7.* We start by noting that since each auction in  $(g^w, p^w) \in \mathcal{M}$  has a allocation and payment functions that satisfies a Lipschitz property, we have  $\forall v'_i, v''_i \in V_i, v \in V$  and  $u_i \in \mathcal{U}_i$ ,

$$\begin{aligned}
|u_i(v_i, (v'_i, v_{-i})) - u_i(v_i, (v''_i, v_{-i}))| & \leq |v_i \cdot (g_i^w(v'_i, v_{-i}) - g_i^w(v''_i, v_{-i}))| + |p_i^w(v'_i, v_{-i}) - p_i^w(v''_i, v_{-i})| \\
& \leq \|v_i\|_\infty \cdot \|g_i^w(v'_i, v_{-i}) - g_i^w(v''_i, v_{-i})\|_1 + |p_i^w(v'_i, v_{-i}) - p_i^w(v''_i, v_{-i})| \\
& \leq 2\mathcal{L} \|v'_i - v''_i\|_\infty.
\end{aligned}$$

We decompose the left-hand side of the desired bound. For any  $(g^w, p^w) \in \mathcal{M}$

$$\begin{aligned}
& rgt_i(w) - \widetilde{rgt}_i(w) \\
& = \mathbf{E}_{v \sim F} \left[ \max_{\bar{v}_i \in V} (u_i(v_i, (\bar{v}_i, v_{-i})) - u_i(v_i, (v_i, v_{-i}))) \right] - \frac{1}{L} \sum_{\ell=1}^L \max_{v'_i \in S_i^{(\ell)}} u_i(v_i^{(\ell)}, (v'_i, v_{-i}^{(\ell)})) - u_i(v_i^{(\ell)}, (v_i^{(\ell)}, v_{-i}^{(\ell)})) \\
& = \underbrace{\mathbf{E}_{v \sim F} \left[ \max_{\bar{v}_i \in V} u_i(v_i, (\bar{v}_i, v_{-i})) - u_i(v_i, (v_i, v_{-i})) \right] - \frac{1}{L} \sum_{\ell=1}^L \max_{\bar{v}_i \in V_i} u_i(v_i^{(\ell)}, (\bar{v}_i, v_{-i}^{(\ell)})) - u_i(v_i^{(\ell)}, (v_i^{(\ell)}, v_{-i}^{(\ell)}))}_{\text{term}_1} \\
& \quad + \underbrace{\frac{1}{L} \sum_{\ell=1}^L \max_{\bar{v}_i \in V_i} u_i(v_i^{(\ell)}, (\bar{v}_i, v_{-i}^{(\ell)})) - \frac{1}{L} \sum_{\ell=1}^L \max_{v'_i \in S_i^{(\ell)}} u_i(v_i^{(\ell)}, (v'_i, v_{-i}^{(\ell)}))}_{\text{term}_2}.
\end{aligned}$$

Bounding term<sub>1</sub> using techniques from Theorem 1 and term<sub>2</sub> using Lemma 7, we have:

$$\begin{aligned}
& \mathbf{E}_v \left[ \max_{\bar{v}_i \in V_i} (u_i(v_i, (\bar{v}_i, v_{-i})) - u_i(v_i, (v_i, v_{-i}))) \right] - \frac{1}{L} \sum_{\ell=1}^L \max_{v'_i \in S_i^{(\ell)}} (u_i(v_i^{(\ell)}, (v'_i, v_{-i}^{(\ell)})) - u_i(v_i^{(\ell)}, (v_i^{(\ell)}, v_{-i}^{(\ell)}))) \\
& \leq 2 \inf_{\epsilon > 0} \left( \epsilon + 2 \left( \frac{2 \log(\mathcal{N}_\infty(\mathcal{M}, \epsilon/2))}{L} \right)^{1/2} + 2\mathcal{L} \left( \frac{\log(2L \mathcal{N}_\infty(\mathcal{M}, \epsilon/4)/\delta)}{Q} \right)^{1/D} \right) + C \sqrt{\frac{\log(n/\delta)}{L}},
\end{aligned}$$

From the proof of Theorem 1 (step 2), we know that  $\mathcal{N}_\infty(\mathcal{U}_i, \epsilon) \leq \mathcal{N}_\infty(\mathcal{U}, \epsilon) \leq \mathcal{N}_\infty(\mathcal{M}, \epsilon)$ . This, along with a union bound over each bidder completes the proof of Theorem 7.

□