Optimal Auctions through Deep Learning*

Paul Dütting[a], Zhe Feng[b], Harikrishna Narasimhan[c], David C. Parkes[b], and Sai Srivatsa Ravindranath[b]

[a]Department of Mathematics, London School of Economics
p.d.duetting@lse.ac.uk
[b]John A. Paulson School of Engineering and Applied Sciences, Harvard University
zhe_feng,parkes,saisr@g.harvard.edu
[c]Google Research, Mountain View
hnarasimham@google.com

February 29, 2020

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 settings with two or more items. In this work, we initiate the exploration of the use of tools from deep learning for the automated design of optimal auctions. We model an auction as a multi-layer neural network, frame optimal auction design as a constrained learning problem, and show how it can be solved using standard machine learning pipelines. We prove generalization bounds and present extensive experiments, recovering essentially all known analytical solutions for multi-item settings, and obtaining novel mechanisms for settings in which the optimal mechanism is unknown.

1 Introduction

Optimal auction design is one of the cornerstones of economic theory. It is of great practical importance, as auctions are used across industries and by the public sector to organize the sale of their products and services. Concrete examples are the US FCC Incentive Auction, the sponsored search auctions conducted by web search engines such as Google, and the auctions run on platforms such as eBay. In the standard *independent private valuations* model, each bidder has a valuation

*Part of the work of the first author was completed while visiting Google Research. This work is supported in part through NSF award CCF-1841550, as well as a Google Fellowship for Zhe Feng. We would like to thank Dirk Bergemann, Yang Cai, Vincent Conitzer, Yannai Gonczarowski, Constantinos Daskalakis, Glenn Ellison, Sergiu Hart, Ron Lavi, Kevin Leyton-Brown, Shengwu Li, Noam Nisan, Parag Pathak, Alexander Rush, Karl Schlag, Alex Wolitzky, participants in the Economics and Computation Reunion Workshop at the Simons Institute, the NIPS'17 Workshop on Learning in the Presence of Strategic Behavior, a Dagstuhl Workshop on Computational Learning Theory meets Game Theory, the EC'18 Workshop on Algorithmic Game Theory and Data Science, the Annual Congress of the German Economic Association, participants in seminars at LSE, Technion, Hebrew, Google, HBS, MIT, and the anonymous reviewers on earlier versions of this paper for their helpful feedback. The first version of this paper was posted on arXiv on June 12, 2017. An extended abstract appeared in the Proceedings of the 36th International Conference on Machine Learning. The source code for all experiments is available from Github at https://github.com/saisrivatsan/deep-opt-auctions.

function over subsets of items, drawn independently from not necessarily identical distributions. It is assumed that the auctioneer knows the distributions and can (and will) use this information in designing the auction. A major difficulty in designing auctions is that valuations are private and bidders need to be incentivized to report their valuations truthfully. The goal is to learn an incentive compatible auction that maximizes revenue.

In a seminal piece of work, Myerson resolved the optimal auction design problem when there is a single item for sale [Myerson, 1981]. Quite astonishingly, even after 30-40 years of intense research, the problem is not completely resolved even for a simple setting with two bidders and two items. While there have been some elegant partial characterization results [Manelli and Vincent, 2006, Pavlov, 2011, Haghpanah and Hartline, 2019, Giannakopoulos and Koutsoupias, 2018, Daskalakis et al., 2017, Yao, 2017], and an impressive sequence of recent algorithmic results [Cai et al., 2012b,a, 2013, Hart and Nisan, 2017, Babaioff et al., 2014, Yao, 2015, Cai and Zhao, 2017, Chawla et al., 2010], most of them apply to the weaker notion of Bayesian incentive compatibility (BIC). Our focus is on designing auctions that satisfy dominant-strategy incentive compatibility (DSIC), 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 [Cole and Roughgarden, 2014, Mohri and Medina, 2016, Huang et al., 2018, Morgenstern and Roughgarden, 2015, Gonczarowski and Nisan, 2017, Morgenstern and Roughgarden, 2016, Syrgkanis, 2017, Gonczarowski and Weinberg, 2018, Balcan et al., 2016]. A handful of works has leveraged machine learning to optimize different aspects of mechanisms [Lahaie, 2011, Dütting et al., 2014, Narasimhan et al., 2016], but none of these offers the generality and flexibility of our approach. There have also been computational approaches to auction design, under the agenda of *automated mechanism design* [Conitzer and Sandholm, 2002, 2004, Sandholm and Likhodedov, 2015], but where scalable, they are limited to specialized classes of auctions known to be incentive compatible.

1.1 Our Contribution

In this work we provide the first, general purpose, end-to-end approach for solving the multi-item auction design problem. We use multi-layer neural networks to encode auction mechanisms, with bidder valuations being the input and allocation and payment decisions being the output. We then train the networks using samples from the value distributions, so as to maximize expected revenue subject to constraints for incentive compatibility.

We propose two different approaches to handling IC constraints. In the first, we leverage characterization results for IC mechanisms, and constrain the network architecture appropriately. We specifically show how to exploit Rochet's characterization result for single-bidder multi-item settings [Rochet, 1987], which states that DSIC mechanisms induce Lipschitz, non-decreasing, and *convex* utility functions.

Our second approach, replaces the IC constraints with the goal of minimizing expected ex post regret, and then lifts the constraints into the objective via the *augmented Langrangian* method. We minimize a combination of negated revenue, and a penalty term for IC violations. This approach is also applicable in multi-bidder multi-item settings for which we don't have tractable characterizations of IC mechanisms, but will generally only find mechanisms that are approximately incentive compatible.

We show through extensive experiments that our two approaches are capable of recovering essentially all analytical results that have been obtained over the past 30-40 years, and that deep learning is also a powerful tool for confirming or refuting hypotheses concerning the form of optimal auctions and can be used to find new designs.

We also present *generalization bounds* in the style of machine learning that provide confidence intervals on the expected revenue and expected ex post regret based on the empirical revenue and empirical regret during training, the complexity of the neural network used to encode the allocation and payment rules, and the number of samples used to train the network.

1.2 Discussion

In general, the optimization problems we face may be non-convex, and so gradient-based approaches may get stuck in local optima. Empirically, however, this has not been an obstacle to deep nets in other problem domains, and there is growing theoretical evidence in support of this "no local optima" phenomenon (see, e.g., [Choromanska et al., 2015, Kawaguchi, 2016, Patel et al., 2016]).

By focusing on expected ex post regret we adopt a quantifiable relaxation of dominant-strategy incentive compatibility, first introduced in [Dütting et al., 2014]. Our experiments suggest that this relaxation is an effective tool for approximating 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 using the standard pipeline. A key advantage of our method over other approaches to automated mechanism design such as [Sandholm and Likhodedov, 2015] is that we optimize over a broad class of mechanisms, constrained only by the expressivity of the neural network architecture.

While the original work on automated auction design framed the problem as a linear program (LP) [Conitzer and Sandholm, 2002, 2004], follow-up work acknowledged that this has severe scalablility issues as it requires a number of constraints and variables that is exponential in the number of agents and items [Guo and Conitzer, 2010]. We find that even for small setting with 2 bidders and 3 items (and a discretization of the value into 5 bins per item) the corresponding 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 low regret in just over 9 hours (see Table 10).

1.3 Further Related Work

Several other research groups have recently picked up deep nets and inference tools and applied them to economic problems, different from the one we consider here. These include the use of neural networks to predict behavior of human participants in strategic scenarios [Hartford et al., 2016, Fudenberg and Liang, 2019], an automated equilibrium analysis of mechanisms [Thompson et al., 2017], deep nets for causal inference [Hartford et al., 2017, Louizos et al., 2017], and deep reinforcement learning for solving combinatorial games [Raghu et al., 2018].[1]

1.4 Organization

Section 2 formulates the auction design problem as a learning problem, describes our two basic approaches, and states the generalization bound. Section 3 presents the network architectures, and instantiates the generalization bound for these networks. Section 4 describes the training and optimization procedures, and Section 5 the experiments. Section 6 concludes.

[1]There has also been follow-up work to the present paper that extends our approach to budget constrained bidders [Feng et al., 2018] and to the facility location problem [Golowich et al., 2018], and that develops specialized architectures for single bidder settings that satisfy IC [Shen et al., 2019] and for the purpose of minimizing agent payments [Tacchetti et al., 2019]. A short survey also appears as a chapter in [Dütting et al., 2019].

2 Auction Design as a Learning Problem

2.1 Auction Design Basics

We consider a setting with a set of n bidders $N = \{1, \ldots, n\}$ and m items $M = \{1, \ldots, m\}$. Each bidder i has a valuation function $v_i : 2^M \to \mathbb{R}_{\geq 0}$, where $v_i(S)$ denotes how much the bidder values the subset of items $S \subseteq M$. In the simplest case, a bidder may have *additive* valuations. In this case she has a value $v_i(\{j\})$ for each individual item $j \in M$, and her value for a subset of items $S \subseteq M$ is $v_i(S) = \sum_{j \in S} v_i(\{j\})$. If a bidder's value for a subset of items $S \subseteq M$ is $v_i(S) = \max_{j \in S} v_i(\{j\})$, we say this bidder has a *unit-demand* valuation. We also consider bidders with general combinatorial valuations, but defer the details to Appendix A.2.

Bidder i's valuation function is drawn independently from a distribution F_i over possible valuation functions V_i. We write $v = (v_1, \ldots, v_n)$ for a profile of valuations, and denote $V = \prod_{i=1}^n V_i$. The auctioneer knows the distributions $F = (F_1, \ldots, F_n)$, but does not know the bidders' realized valuation v. The bidders report their valuations (perhaps untruthfully), and an auction decides on an allocation of items to the bidders and charges a payment to them. We denote an auction (g, p) as a pair of allocation rules $g_i : V \to 2^M$ and payment rules $p_i : V \to \mathbb{R}_{\geq 0}$ (these rules can be randomized). Given bids $b = (b_1, \ldots, b_n) \in V$, the auction computes an allocation $g(b)$ and payments $p(b)$.

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

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

2.2 Formulation as a Learning Problem

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

We present two approaches for achieving IC. In the first, we leverage characterization results to constrain the search space so that all mechanisms within this class are IC. In the second, we replace the IC constraints with a differentiable approximation, and lift the constraints into the objective via the augmented Lagrangian method. The first approach affords a smaller search space and is exactly DSIC, but requires an IC characterization that can be encoded within a neural network architecture and applies to single-bidder multi-item settings. The second approach applies to multi-bidder multi-item settings and does not rely on the availability of suitable characterization results, but entails search through a larger parametric space and only achieves approximate IC.

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

2.2.1 Characterization-Based Approach

We begin by describing our first approach, in which we exploit characterizations of IC mechanisms to constrain the search space. We provide a construction for single-bidder multi-item settings based on Rochet [1987]'s characterization of IC mechanisms via induced utilities, which we refer to as *RochetNet*. We present this construction for additive preferences, but the construction can easily be extended to unit demand valuations. See Section 3.1. In Appendix A.1 we describe a second construction based on Myerson [1981]'s characterization result for single-bidder multi-item settings, which we refer to as *MyersonNet*.

To formally state Rochet's result we need the following notion of an induced utility function. The utility function $u : \mathbb{R}^m_{\geq 0} \to \mathbb{R}$ *induced* by a mechanism (g, p) for a single bidder with additive preferences is[3]:

$$u(v) = \sum_{j=1}^{m} g_j(v)\, v_j \; - \; p(v). \tag{1}$$

Rochet's result establishes the following connection between DSIC mechanisms and induced utility functions:

Theorem 2.1 (Rochet [1987])**.** *A utility function $u : \mathbb{R}^m_{\geq 0} \to \mathbb{R}$ is induced by a DSIC mechanism iff u is 1-Lipschitz w.r.t. the ℓ_1-norm, non-decreasing, and convex. Moreover, for such a utility function u, $\nabla u(v)$ exists almost everywhere in $\mathbb{R}^m_{\geq 0}$, 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.*

Further, for a mechanism to be IR, its induced utility function must be non-negative, i.e. $u(v) \geq 0, \forall v \in \mathbb{R}^m_{\geq 0}$.

To find the optimal mechanism, it thus suffices to search over all non-negative utility functions that satisfy the conditions in Theorem 2.1, 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 learning problem. The associated mechanism can then be recovered from the gradient of the learned neural network. We describe the neural network architectures for this approach in Section 3.1, and we present extensive experiments with this approach in Section 5 and Appendix B.

2.2.2 Characterization-Free Approach

Our second approach—which we refer to as *RegretNet*—does not rely on characterizations of IC mechanisms. Instead, it replaces the IC constraints with a differentiable approximation and lifts the IC constraints into the objective by augmenting the objective with a term that accounts for the extent to which the IC constraints are violated.

We propose to measure the extent to which an auction violates incentive compatibility through the following notion of ex post regret. 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. For mechanisms (g^w, p^w), we will be interested in the *expected ex post regret* for bidder i:

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

where the expectation is over $v \sim F$ and $u^w_i(v_i; b) = v_i(g^w_i(b)) - p^w_i(b)$ for model parameters w. We assume that F has full support on the space of valuation profiles V, and recognizing that the regret

[3] For a unit-demand bidder, the utility can also be represented via (1) with the additional constraint that $\sum_j g_j(v) \leq 1, \forall v$. We discuss this more in Section 3.1.

is non-negative, an auction satisfies DSIC if and only if $rgt_i(w) = 0, \forall i \in N$, except for measure zero events.

Given this, we re-formulate the learning problem as minimizing expected negated revenue subject to the expected ex post regret being zero for each bidder:

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

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

$$\widehat{rgt}_i(w) = \;\; \tfrac{1}{L} \textstyle\sum_{\ell=1}^{L} \Big[\max_{v_i' \in V_i} \; u_i^w\big(v_i^{(\ell)}; \big(v_i', v_{-i}^{(\ell)}\big)\big) - u_i^w\big(v_i^{(\ell)}; v^{(\ell)}\big) \Big], \tag{2}$$

and seek to minimize the empirical loss (negated revenue) subject to the empirical regret being zero for all bidders:

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

We additionally require the designed auction to satisfy IR, which can be ensured by restricting the search space to a class of parametrized auctions that charge no bidder more than her valuation for an allocation.

In Section 3 we will model the allocation and payment rules as neural networks and incorporate the IR requirement within the architecture. In Section 4 we describe how the IC constraints can be incorporated into the objective using Lagrange multipliers, so that the resulting neural nets can be trained with standard pipelines. Section 5 and Appendix B present extensive experiments.

2.3 Quantile-Based Regret

Our characterization-free approach will lead to mechanisms with low (typically vanishing) expected ex post regret. The bounds on the expected ex post regret also yield guarantees of the form "the probability that the ex post regret is larger than x is at most q."

Definition 2.1 (Quantile-based ex post regret)**.** *For each bidder i, and q with $0 < q < 1$, the q-quantile-based ex post regret, $rgt_i^q(w)$, induced by the probability distribution F on valuation profiles, is defined as the smallest x such that*

$$\mathbf{P}\left(\max_{v_i' \in V_i} u_i^w(v_i; (v_i', v_{-i})) - u_i^w(v_i; (v_i, v_{-i})) \geq x \right) \leq q.$$

We can bound the q-quantile based regret $rgt_i^q(w)$ by the expected ex post regret $rgt_i(w)$ as in the following lemma. The proof appears in Appendix D.1.

Lemma 2.1. *For any fixed q, $0 < q < 1$, and bidder i, we can bound the q-quantile-based ex post regret by*

$$rgt_i^q(w) \leq \frac{rgt_i(w)}{q}.$$

Using this lemma we can show, for example, that when the expected ex post regret is 0.001, then the probability that the ex post regret exceeds 0.01 is at most 10%.

2.4 Generalization Bound

We conclude this section with two generalization bounds. We provide a lower bound on the expected revenue and an upper bound on the expected ex post regret in terms of the empirical revenue and empirical regret during training, the complexity (or capacity) of the auction class that we optimize over, and the number of sampled valuation profiles.

We measure the capacity of an auction class $\mathcal{M}$ using a definition of *covering numbers* used in the ranking literature [Rudin and Schapire, 2009]. Define the $\ell_{\infty,1}$ distance between auctions $(g, p), (g', p') \in \mathcal{M}$ as

$$\max_{v \in V} \sum_{i \in N, j \in M} |g_{ij}(v) - g'_{ij}(v)| + \sum_{i \in N} |p_i(v) - p'_i(v)|.$$

For any $\epsilon > 0$, let $\mathcal{N}_\infty(\mathcal{M}, \epsilon)$ be the minimum number of balls of radius ϵ required to cover $\mathcal{M}$ under the $\ell_{\infty,1}$ distance.

Theorem 2.2. *For each bidder i, assume w.l.o.g. that the valuation function v_i satisfies $v_i(S) \leq 1, \forall S \subseteq M$. Let $\mathcal{M}$ be a class of auctions that satisfy individual rationality. Fix $\delta \in (0, 1)$. With probability at least $1 - \delta$ over draw of sample $\mathcal{S}$ of L profiles from F, for any $(g^w, p^w) \in \mathcal{M}$,*

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

and

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

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

See Appendix D.2 for the proof. If the term Δ_L in the above bound goes to zero as the sample size L increases then the above bounds go to zero as $L \to \infty$. In Theorem 3.2 in Section 3, we bound Δ_L for the neural network architectures we present in this paper.

3 Neural Network Architecture

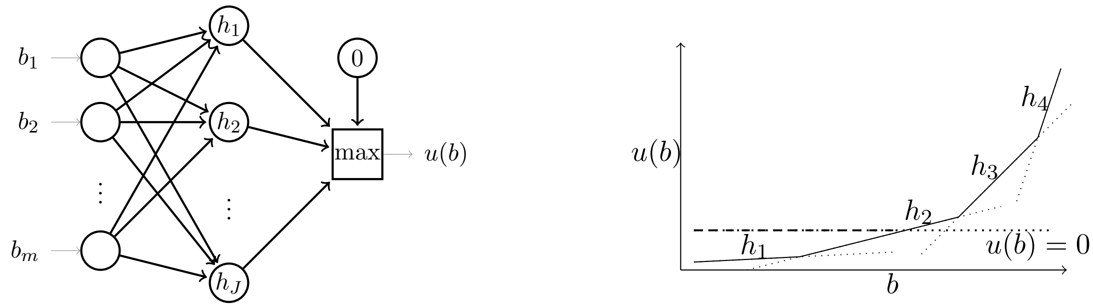
We describe the RochetNet architecture for single-bidder multi-item settings in Section 3.1, and the RegretNet architecture for multi-bidder multi-item settings in Section 3.2. We focus on additive and unit-demand preferences. We discuss how to extend the constructions to capture combinatorial valuations for multi-bidder, multi-item settings in Appendix A.2.

3.1 The RochetNet Architecture

Recall that in the single-bidder, multi-item setting we seek to encode utility functions that satisfy the requirements of Theorem 2.1. The associated auction mechanism can be deduced from the gradient of the utility function.

We first describe the construction for additive valuations. To model a non-negative, monotone, convex, Lipschitz utility function, we use the maximum of J linear functions with non-negative coefficients and zero:

$$u^{\alpha,\beta}(v) \;=\; \max\left\{\max_{j \in [J]}\{\alpha_j \cdot v \;+\; \beta_j\},\, 0\right\}, \tag{4}$$



(a) **(b)**

Figure 1: RochetNet: (a) Neural network representation of a non-negative, monotone, convex induced utility function; here $h_j(b) = \alpha_j \cdot b + \beta_j$ for $b \in \mathbb{R}^m$ and $\alpha_j \in [0,1]^m$. (b) An example of a utility function represented by RochetNet for one item.

where parameters $w = (\alpha, \beta)$, with $\alpha_j \in [0,1]^m$ and $\beta_j \in \mathbb{R}$ for $j \in [J]$. By bounding the hyperplane coefficients to $[0,1]$, we guarantee that the function is 1-Lipschitz. The following theorem verifies that the utility modeled by RochetNet satisfies Rochet's characterization (Theorem 2.1). The proof is given in Appendix D.3.

Theorem 3.1. *For any $\alpha \in [0,1]^{mJ}$ 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 ℓ_1-norm.*

The utility function, represented as a single layer neural network, is illustrated in Figure 1(a), where each $h_j(b) = \alpha_j \cdot b + \beta_j$ for bid $b \in \mathbb{R}^m$. Figure 1(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^w, p^w), with $w = (\alpha, \beta)$, can be derived from the gradient of the utility function, with the allocation rule given by:

$$g^w(b) = \nabla u^{\alpha,\beta}(b), \tag{5}$$

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

$$p^w(b) \;=\; \nabla u^{\alpha,\beta}(b) \cdot b \;-\; u^{\alpha,\beta}(b). \tag{6}$$

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

$$-\mathbf{E}_{v \sim F}\big[\nabla u^{\alpha,\beta}(v) \cdot v - u^{\alpha,\beta}(v)\big] = \mathbf{E}_{v \sim F}\big[\beta_{j^*(v)}\big]. \tag{7}$$

To ensure that the objective is a continuous function of the parameters α and β (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:

$$\mathcal{L}(\alpha, \beta) \;=\; -\mathbf{E}_{v \sim F}\big[\sum_{j \in [J]} \beta_j \widetilde{\nabla}_j(v)\big], \tag{8}$$

where

$$\widetilde{\nabla}_j(v) \;=\; \operatorname{softmax}_j\big(\kappa \cdot (\alpha_1 \cdot v + \beta_1), \ldots, \kappa \cdot (\alpha_J \cdot v + \beta_J)\big) \tag{9}$$

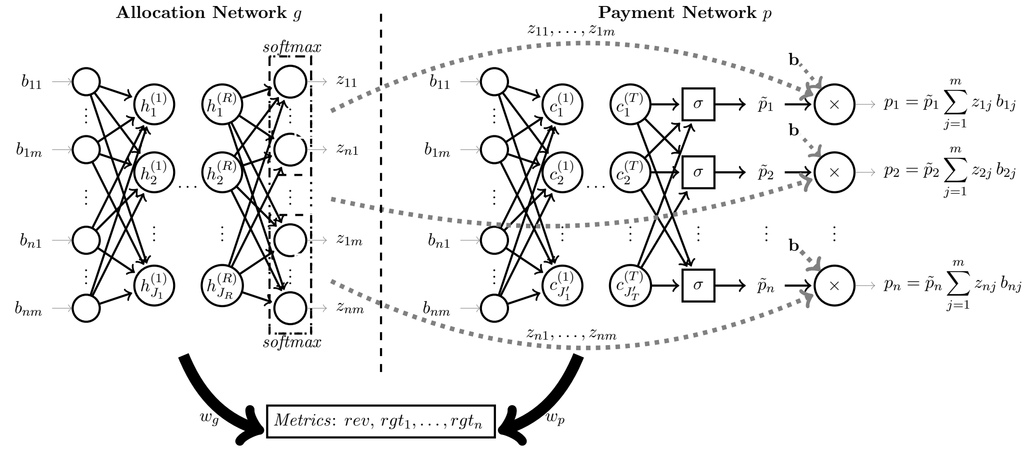


Figure 2: RegretNet: The allocation and payment networks for a setting with n additive bidders and m items. The inputs are bids from each bidder for each item. The revenue rev and expected ex post rgt_i are defined as a function of the parameters of the allocation and payment networks $w = (w_g, w_p)$.

and $\kappa > 0$ is a constant that controls the quality of the approximation.[4] We seek to optimize the parameters of the neural network $\alpha \in [0,1]^{mJ}, \beta \in \mathbb{R}^J$ to minimize loss. Given a sample $\mathcal{S} = \{v^{(1)}, \ldots, v^{(L)}\}$ drawn from F, we optimize an empirical version of the loss.

This approach easily extends to a single bidder with a unit-demand valuation. In this case, the sum of the allocation probabilities cannot exceed one. This is enforced by restricting the coefficients for each hyperplane to sum up to at most one, i.e. $\sum_{k=1}^{m} \alpha_{jk} \leq 1, \forall j \in [J]$, and $\alpha_{jk} \geq 0, \forall j \in J, k \in [m]$.[5] 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.[6]

An interpretation of the RochetNet 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 bid. Each linear function $h_j(b) = \alpha_j \cdot b + \beta_j$ in RochetNet corresponds to an option on the menu, with the allocation probabilities and payments encoded through the parameters α_j and β_j respectively. Recently, Shen et al. [2019] extended RochetNet to more general settings, including non-linear utility function settings.

3.2 The RegretNet Architecture

We next describe the basic architecture for the characterization-free, RegretNet approach. Recall that in this case the goal is to train neural networks that explicitly encode the allocation and payment rule of the mechanism. The architectures generally consist of two logically distinct components: the allocation and payment networks. These components are trained together and the outputs of these networks are used to compute the regret and revenue of the auction.

3.2.1 Additive Valuations

An overview of the RegretNet architecture for additive valuations is given in Figure 2. The allocation network encodes a randomized allocation rule $g^w : \mathbb{R}^{nm} \to [0,1]^{nm}$ and the payment network encodes a payment rule $p^w : \mathbb{R}^{nm} \to \mathbb{R}^n_{\geq 0}$, both of which are modeled as feed-forward,

[4]The softmax function, $\text{softmax}_j(\kappa x_1, \ldots, \kappa x_J) = e^{\kappa x_j} / \sum_{j'} e^{\kappa x_{j'}}$, takes as input J real numbers and returns a probability distribution consisting of J probabilities, proportional to the exponential of the inputs.

[5]To achieve this contraint, we can re-parameterize α_{jk} as $\text{softmax}_k(\gamma_{j1}, \cdots, \gamma_{jm})$, where $\gamma_{jk} \in \mathbb{R}, \forall j \in J, k \in m$.

[6]The original characterization of Rochet [1987] applies to general, convex outcome spaces, as is the case here.

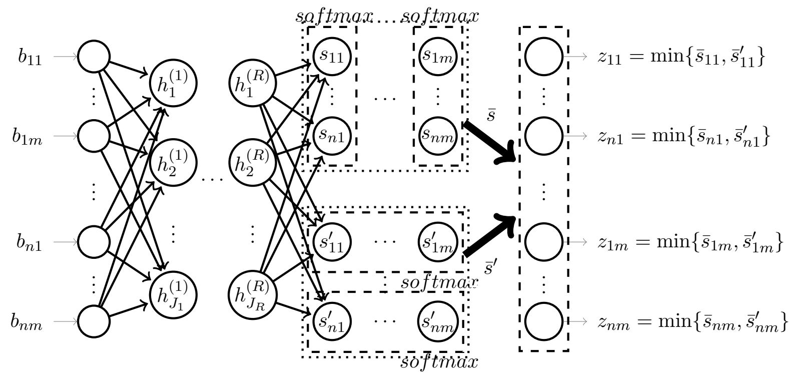


Figure 3: RegretNet: The allocation network for settings with n *unit-demand* bidders and m items.

fully-connected networks with a tanh activation function in each of the hidden nodes. The input layer of the networks consists of bids $b_{ij} \geq 0$ representing the valuation of bidder i for item j.

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

To ensure that the auction satisfies *IR*, i.e., does not charge a bidder more than her expected value for the allocation, the network first computes a normalized payment $\tilde{p}_i \in [0, 1]$ for each bidder i using a sigmoidal unit, and then outputs a payment $p_i = \tilde{p}_i(\sum_{j=1}^{m} z_{ij}\, b_{ij})$, where the z_{ij}'s are the outputs from the allocation network.

3.2.2 Unit-Demand Valuations

The allocation network for unit-demand bidders is the feed-forward network shown in Figure 3. For revenue maximization in this setting, it is sufficient to consider allocation rules that assign at most one item to each bidder.[7] In the case of randomized allocation rules, this requires that the total allocation probability to each bidder is at most one, 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.[8]

In particular, we have the allocation network compute two sets of scores s_{ij}'s and s'_{ij}'s. Let s, $s' \in \mathbb{R}^{nm}$ denote the corresponding matrices. The first set of scores are 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:

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

[7] 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 a bidder.

[8] This is a more general definition for doubly stochastic than is typical. Doubly stochastic is usually defined on a square matrix with the sum of rows and the sum of columns equal to 1.

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.

We first show that $\varphi^{DS}(s, s')$ as constructed is doubly stochastic, and that we do not lose in generality by the constructive approach that we take. See Appendix D.4 for a proof.

Lemma 3.1. *The matrix $\varphi^{DS}(s, s')$ is doubly stochastic $\forall\, s, s' \in \mathbb{R}^{nm}$. For any doubly stochastic matrix $z \in [0,1]^{nm}$, $\exists\, s, s' \in \mathbb{R}^{nm}$, for which $z = \varphi^{DS}(s, s')$.*

It remains to show that doubly-stochastic matrices correspond to lotteries over one-to-one assignments. This is a special case of the *bihierarchy* structure proposed in [Budish et al., 2013] (Theorem 1), which we state in the following lemma for completeness.[9]

Lemma 3.2 (Budish et al. [2013])**.** *Any doubly stochastic matrix $A \in \mathbb{R}^{n\times m}$ can be represented as a convex combination of matrices $B^1, \ldots, B^k$ where each $B^\ell \in \{0,1\}^{n\times m}$ and $\sum_{j\in[m]} B_{ij} \le 1$, $\forall i \in [n]$ and $\sum_{i\in[n]} B_{ij} \le 1$, $\forall j \in [m]$.*

The payment network for unit-demand valuations is the same as for the case of additive valuations (see Figure 2).

3.3 Covering Number Bounds

We conclude this section by instantiating our generalization bound from Section 2.4 for the RegretNet architectures, where we have both a regret and revenue term. Analogous results can be derived for RochetNet, where we only have a revenue term.

Theorem 3.2. *For RegretNet with R hidden layers, K nodes per hidden layer, d_g parameters in the allocation network, d_p parameters in the payment network, m items, n bidders, a sample size of L, and the vector of all model parameters w satisfying $\|w\|_1 \le W$ [10] the following are valid bounds for the Δ_L term defined in Theorem 2.2, for different bidder valuation types:*

(a) additive valuations:

$\Delta_L \le O\big(\sqrt{R(d_g + d_p)\log(LW\max\{K, mn\})/L}\big)$,

(b) unit-demand valuations:

$\Delta_L \le O\big(\sqrt{R(d_g + d_p)\log(LW\max\{K, mn\})/L}\big)$,

The proof is given in Appendix D.7. As the sample size $L \to \infty$, the term $\Delta_L \to 0$. The dependence of the above result on the number of layers, nodes, and parameters in the network is similar to standard covering number bounds for neural networks [Anthony and Bartlett, 2009].

4 Optimization and Training

We next describe how we train the neural network architectures presented in the previous section. We focus on the RegretNet architectures where we have to take care of the incentives directly. The approach that we take for RochetNet is the standard (projected) stochastic gradient descent[11] (SGD) for loss function $\mathcal{L}(\alpha, \beta)$ in Equation 8.

For RegretNet 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

[9] Budish et al. [2013] also propose a polynomial algorithm to decompose the doubly stochastic matrix.

[10] Recall that $\|\cdot\|_1$ is the induced matrix norm, i.e. $\|w\|_1 = \max_j \sum_i |w_{ij}|$.

[11] During training for additive valuations setting in RochetNet, we project each weight α_{jk} into $[0,1]$ to guarantee feasibility.

Algorithm 1 RegretNet Training

Input: Minibatches $\mathcal{S}_1, \ldots, \mathcal{S}_T$ of size B
Parameters: $\forall t, \rho_t > 0$, $\gamma > 0$, $\eta > 0$, $\Gamma \in \mathbb{N}$, $K \in \mathbb{N}$
Initialize: $w^0 \in \mathbb{R}^d$, $\lambda^0 \in \mathbb{R}^n$
for $t = 0$ **to** T **do**
  Receive minibatch $\mathcal{S}_t = \{v^{(1)}, \ldots, v^{(B)}\}$
  Initialize misreports ${v'}_i^{(\ell)} \in V_i, \forall \ell \in [B],\ i \in N$
  for $r = 0$ **to** Γ **do**
    $\forall \ell \in [B],\ i \in N$:
      ${v'}_i^{(\ell)} \leftarrow {v'}_i^{(\ell)} + \gamma \nabla_{v'_i} u_i^w\big(v_i^{(\ell)}; \big({v'}_i^{(\ell)}, v_{-i}^{(\ell)}\big)\big)$
  end for
  Compute regret gradient: $\forall \ell \in [B], i \in N$:
    $g_{\ell,i}^t =$
      $\nabla_w \Big[u_i^w\big(v_i^{(\ell)}; \big({v'}_i^{(\ell)}, v_{-i}^{(\ell)}\big)\big) - u_i^w\big(v_i^{(\ell)}; v^{(\ell)}\big)\Big] \Big|_{w=w^t}$
  Compute Lagrangian gradient using (10) and update w^t:
    $w^{t+1} \leftarrow w^t - \eta \nabla_w \mathcal{C}_{\rho_t}(w^t, \lambda^t)$
  Update Lagrange multipliers once in Q iterations:
    if t is a multiple of Q
      $\lambda_i^{t+1} \leftarrow \lambda_i^t + \rho_t\, \widetilde{rgt}_i(w^{t+1}), \quad \forall i \in N$
    else
      $\lambda^{t+1} \leftarrow \lambda^t$
end for

function for the optimization problem, augmented with a quadratic penalty term for violating the constraints:

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

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

The solver is described in Algorithm 1. We divide the training sample $\mathcal{S}$ into minibatches 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 $\mathcal{S}_t = \{v^{(1)}, \ldots, v^{(B)}\}$. The update (a) on model parameters involves an unconstrained optimization of $\mathcal{C}_\rho$ over w and is performed using a gradient-based optimizer. Let $\widetilde{rgt}_i(w)$ denote the empirical regret in (2) computed on minibatch $\mathcal{S}_t$. The gradient of $\mathcal{C}_\rho$ w.r.t. w for fixed λ^t is given by:

$$\nabla_w\, \mathcal{C}_\rho(w;\, \lambda^t) \;=\; -\frac{1}{B}\sum_{\ell=1}^{B}\sum_{i \in N} \nabla_w\, p_i^w(v^{(\ell)}) + \sum_{i \in N}\sum_{\ell=1}^{B} \lambda_i^t\, g_{\ell,i} \;+\; \rho \sum_{i \in N}\sum_{\ell=1}^{B} \widetilde{rgt}_i(w)\, g_{\ell,i} \tag{10}$$

where

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

The terms $\widetilde{rgt}_i$ and $g_{\ell,i}$ in turn involve a "max" over misreports for each bidder i and valuation profile ℓ. We solve this inner maximization over misreports using another gradient based optimizer. In particular, we maintain misreports $v'^{(\ell)}_i$ for each i and valuation profile ℓ. For every update on the model parameters w^t, we perform Γ gradient updates to compute the optimal misreports: $v'^{(\ell)}_i = v'^{(\ell)}_i + \gamma \nabla_{v'_i} u^w_i\big(v^{(\ell)}_i; \big(v'^{(\ell)}_i, v^{(\ell)}_{-i}\big)\big)$, for some $\gamma > 0$. We show a visualization of these iterations in Appendix C. In our experiments, we use the Adam optimizer [Kingma and Ba, 2014] for updates on model parameters w and misreports $v'^{(\ell)}_i$.[12]

Since the optimization problem 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 optimal auctions in settings where this is known.

5 Experiments

We demonstrate that our approach can recover near-optimal auctions for essentially all settings for which the optimal solution is known, that it is an effective tool for confirming or refuting hypotheses about optimal designs, and that it can find new auctions for settings where there is no known analytical solution. We present a representative subset of the results here, and provide additional experimental results in Appendix B.

5.1 Setup

We implemented our framework using the TensorFlow deep learning library.[13] For RochetNet we initialized parameters α and β in Equation (4) using a random uniform initializer over the interval [0,1] and a zero initializer, respectively. For RegretNet we used the tanh activation function at the hidden nodes, and Glorot uniform initialization [Glorot and Bengio, 2010]. We performed cross validation to decide on the number of hidden layers and the number of nodes in each hidden layer. We include exemplary numbers that illustrate the tradeoffs in Section 5.6.

We trained RochetNet on 2^{15} valuation profiles sampled every iteration in an online manner. We used the Adam optimizer with a learning rate of 0.1 for 20,000 iterations for making the updates. The parameter κ in Equation (9) was set to 1,000. Unless specified otherwise we used a max network over 1,000 linear functions to model the induced utility functions, and report our results on a sample of 10,000 profiles.

For RegretNet we used a sample of 640,000 valuation profiles for training and a sample of 10,000 profiles for testing. The augmented Lagrangian solver was run for a maximum of 80 epochs (full passes over the training set) with a minibatch size of 128. The value of ρ in the augmented Lagrangian was set to 1.0 and incremented every two 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 $\Gamma = 25$ misreport updates steps with learning rate 0.1. At the end of 25 updates, the optimized misreports for the current minibatch were cached and used to initialize the misreports for the same minibatch in the next epoch. An update on λ^t was performed once every 100 minibatches (i.e., $Q = 100$).

We ran all our experiments on a compute cluster with NVDIA Graphics Processing Unit (GPU) cores.

[12] Adam is a variant of SGD, which involves a momentum term to update weights. Lines 9 and 15 in the pseudo-code of Algorithm 1 are for a standard SGD algorithm.

[13] All code is available through the GitHub repository at https://github.com/saisrivatsan/deep-opt-auctions.

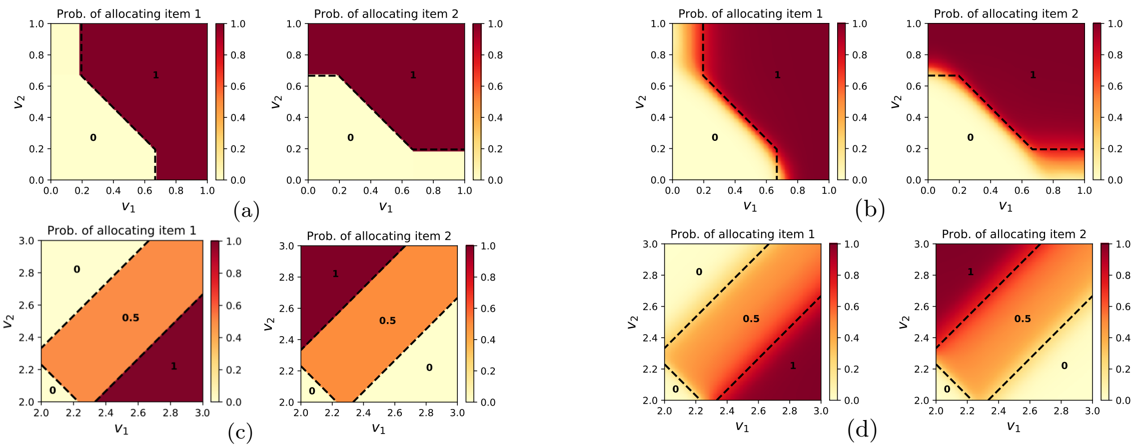


Figure 4: Side-by-side comparison of allocation rules learned by RochetNet and RegretNet for single bidder, two items settings. Panels (a) and (b) are for Setting A and Panels (c) and (d) are for Setting B. The panels describe the probability that the bidder is allocated item 1 (left) and item 2 (right) 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.

5.2 Evaluation

In addition to the revenue of the learned auction on a test set, we also evaluate the regret achieved by RegretNet, averaged across all bidders and test valuation profiles, i.e., $rgt = \frac{1}{n}\sum_{i=1}^{n} \widehat{rgt}_i(g^w, p^w)$. Each $\widehat{rgt}_i$ has an inner "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 2,000 iterations (we test 1,000 different random initial v'_i and report the one that achieves the largest regret).

For some of the experiments we also report the total time it took to train the network. This time is incurred during offline training, while the allocation and payments can be computed in a few milliseconds once the network is trained.

5.3 The Manelli-Vincent and Pavlov Auctions

As a representative example of the exhaustive set of analytical results that we can recover with our approach we discuss the Manelli-Vincent and Pavlov auctions [Manelli and Vincent, 2006, Pavlov, 2011]. We specifically consider the following single-bidder, two-item settings:

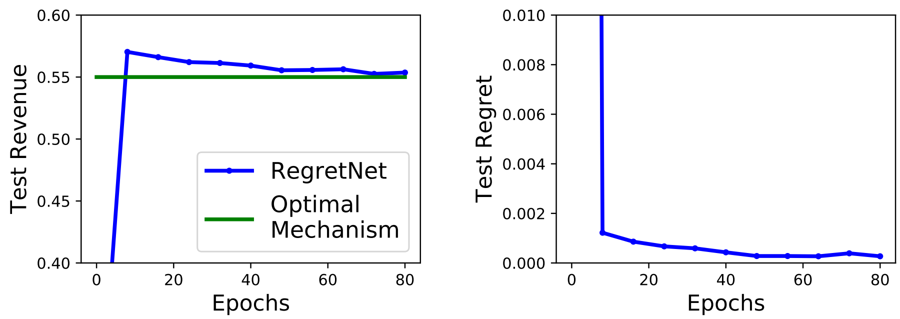
A. Single bidder with additive valuations over two items, where the item values are independent draws from $U[0, 1]$.
B. Single bidder with unit-demand valuations over two items, where the item values are independent draws from $U[2, 3]$.

The optimal design for the first setting is given by Manelli and Vincent [2006], who show that the optimal mechanism is deterministic and offers the bidder three options: receive both items and pay $(4 - \sqrt{2})/3$, receive item 1 and pay $2/3$, or receive item 2 and pay $2/3$. For the second setting Pavlov [2011] shows that it is optimal to offer a fair lottery $(\frac{1}{2}, \frac{1}{2})$ over the items (at a discount), or to purchase any item at a fixed price. For the parameters here the price for the lottery is $\frac{1}{6}(8 + \sqrt{22}) \approx 2.115$ and the price for an individual item is $\frac{1}{6} + \frac{1}{6}(8 + \sqrt{22}) \approx 2.282$.

We used two hidden layers with 100 hidden nodes in RegretNet for these settings. A visualization of the optimal allocation rule and those learned by RochetNet and RegretNet is given in Figure 4. Figure 5(a) gives the optimal revenue, the revenue and regret obtained by RegretNet, and the

Distribution	Opt	RegretNet		RochetNet
	rev	*rev*	*rgt*	*rev*
Setting A	0.550	0.554	< 0.001	0.550
Setting B	2.137	2.137	< 0.001	2.136

(a)



(b)

Figure 5: (a): Test revenue and regret for RegretNet and revenue for RochetNet for Settings A and B. (b): Plot of test revenue and regret as a function of training epochs for Setting A with RegretNet.

revenue obtained by RochetNet. Figure 5(b) shows how these terms evolve over time during training in RegretNet.

We find that both approaches essentially recover the optimal design, not only in terms of revenue, but also in terms of the allocation rule and transfers. The auctions learned by RochetNet are exactly DSIC and match the optimal revenue precisely, with sharp decision boundaries in the allocation and payment rule. The decision boundaries for RegretNet are smoother, but still remarkably accurate. The revenue achieved by RegretNet matches the optimal revenue up to a $< 1\%$ error term and the regret it incurs is < 0.001. The plots of the test revenue and regret show that the augmented Lagrangian method is effective in driving the test revenue and the test regret towards optimal levels.

The additional domain knowledge incorporated into the RochetNet architecture leads to exactly DSIC mechanisms that match the optimal design more accurately, and speeds up computation (the training took about 10 minutes compared to 11 hours). On the other hand, we find it surprising how well RegretNet performs given that it starts with no domain knowledge at all.

We present and discuss a host of additional experiments with single-bidder, two-item settings in Appendix B.

5.4 The Straight-Jacket Auction

Extending the analytical result of Manelli and Vincent [2006] to a single bidder, and an arbitrary number of items (even with additive preferences, all uniform on $[0,1]$) has proven elusive. It is not even clear whether the optimal mechanism is deterministic or requires randomization.

A recent breakthrough came with Giannakopoulos and Koutsoupias [2018], who were able to find a pattern in the results for two items and three items. The proposed mechanism—the Straight-Jacket Auction (SJA)—offers bundles of items at fixed prices. The key to finding these prices is to view the best-response regions as a subdivision of the m-dimensional cube, and observe that there is an intrinsic relationship between the price of a bundle of items and the volume of the respective best-response region.

Items	SJA (*rev*)	RochetNet (*rev*)
2	0.549187	0.549175
3	0.875466	0.875464
4	1.219507	1.219505
5	1.576457	1.576455
6	1.943239	1.943216
7	2.318032	2.318032
8	2.699307	2.699305
9	3.086125	3.086125
10	3.477781	3.477722

Figure 6: Revenue of the Straight-Jacket Auction (SJA) computed via the recursive formula in [Giannakopoulos and Koutsoupias, 2018], and that of the auction learned by RochetNet, for various numbers of items m. The SJA is known to be optimal for up to six items, and conjectured to be optimal for any number of items.

Giannakopoulos and Koutsoupias gave a recursive algorithm for finding the subdivision and the prices, and used LP duality to prove that the SJA is optimal for $m \leq 6$ items.[14] They also conjecture that the SJA remains optimal for general m, but were unable to prove it.

Figure 6 gives the revenue of the SJA, and that found by RochetNet for $m \leq 10$ items. We used a test sample of 2^{30} valuation profiles (instead of 10,000) to compute these numbers for higher precision. It shows that RochetNet finds the optimal revenue for $m \leq 6$ items, and that it finds DSIC auctions whose revenue matches that of the SJA for $m = 7, 8, 9$, and 10 items. Closer inspection reveals that the allocation and payment rules learned by RochetNet essentially match those predicted by Giannakopoulos and Koutsoupias for all $m \leq 10$.

We take this as strong additional evidence that the conjecture of Giannakopoulos and Koutsoupias is correct.

For the experiments in this subsection, we used a max network over 10,000 linear functions (instead of 1,000). We followed up on the usual training phase with an additional 20 iterations of training using Adam optimizer with learning rate 0.001 and a minibatch size of 2^{30}. We also found it useful to impose item-symmetry on the learned auction, especially for $m = 9$ and 10 items, as this helped with accuracy and reduced training time. Imposing symmetry comes without loss of generality for auctions with an item-symmetric distribution [Daskalakis and Weinberg, 2012]. With these modifications it took about 13 hours to train the networks.

5.5 Discovering New Optimal Designs

We next demonstrate the potential of RochetNet to discover new optimal designs. For this, we consider a single bidder with additive but correlated valuations for two items as follows:

C. One additive bidder and two items, where the bidder's valuation is drawn uniformly from the triangle $T = \{(v_1, v_2) | \frac{v_1}{c} + v_2 \leq 2, v_1 \geq 0, v_2 \geq 1\}$ where $c > 0$ is a free parameter.

There is no analytical result for the optimal auction design for this setting. We ran RochetNet for different values of c to discover the optimal auction. The mechanisms learned by RochetNet for $c = 0.5, 1, 3$, and 5 are shown in Figure 8. Based on this, we conjectured that the optimal mechanism contains two menu items for $c \leq 1$, namely $\{(0,0), 0\}$ and $\{(1,1), \frac{2+\sqrt{1+3c}}{3}\}$, and three menu items for $c > 1$, namely $\{(0,0), 0\}$, $\{(1/c, 1), 4/3\}$, and $\{(1,1), 1 + c/3\}$, giving the optimal

[14] The duality argument developed by Giannakopoulos and Koutsoupias is similar but incomparable to the duality approach of Daskalakis et al. [2013]. We will return to the latter in Section 5.5.

c	Opt (rev)	RochetNet (rev)
0.500	1.104783	1.104777
1.000	1.185768	1.185769
3.000	1.482129	1.482147
5.000	1.778425	1.778525

Figure 7: Revenue of the newly discovered optimal mechanism and that of RochetNet, for Setting C with varying parameter c.

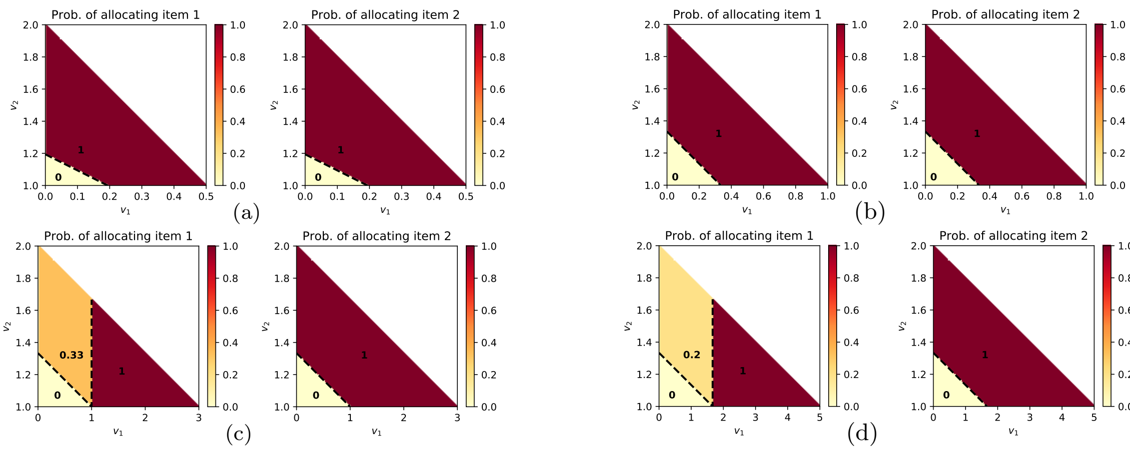


Figure 8: Allocation rules learned by RochetNet for Setting C. The panels describe the probability that the bidder is allocated item 1 (left) and item 2 (right) for $c = 0.5, 1, 3$, and 5. The auctions proposed in Theorem 5.1 are described by the regions separated by the dashed black lines, with the numbers in black the optimal probability of allocation in the region.

allocation and payment in each region. In particular, as c transitions from values less than or equal to 1 to values larger than 1, the optimal mechanism transitions from being deterministic to being randomized. Figure 7 gives the revenue achieved by RochetNet and the conjectured optimal format for a range of parameters c computed on 10^6 valuation profiles.

We validate the optimality of this auction through duality theory [Daskalakis et al., 2013] in Theorem 5.1. The proof is given in Appendix D.8.

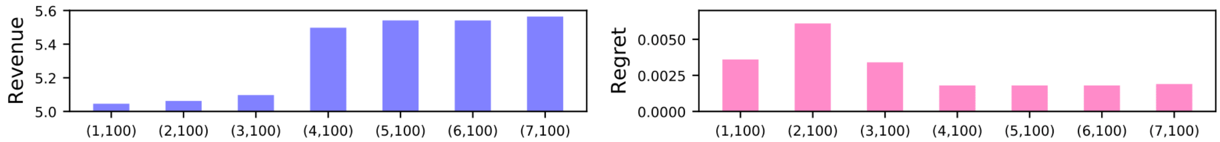
Theorem 5.1. *For any $c > 0$, suppose the bidder's valuation is uniformly distributed over set $T = \{(v_1, v_2) | \frac{v_1}{c} + v_2 \leq 2, v_1 \geq 0, v_2 \geq 1\}$. Then the optimal auction contains two menu items $\{(0,0), 0\}$ and $\{(1,1), \frac{2+\sqrt{1+3c}}{3}\}$ when $c \leq 1$, and three menu items $\{(0,0), 0\}$, $\{(1/c, 1), 4/3\}$, and $\{(1,1), 1 + c/3\}$ otherwise.*

5.6 Scaling Up

We next consider settings with up to five bidders and up to ten items. This is several orders of magnitude more complex than existing analytical or computational results. It is also a natural playground for RegretNet as no tractable characterizations of IC mechanisms are known for these settings.

We specifically consider the following two settings, which generalize the basic setting considered in [Manelli and Vincent, 2006] and [Giannakopoulos and Koutsoupias, 2018] to more than one bidder:

D. Three additive bidders and ten items, where bidders draw their value for each item independently from $U[0, 1]$.



(a)

Setting	RegretNet *rev*	RegretNet *rgt*	Item-wise Myerson	Bundled Myerson
D: 3×10	5.541	< 0.002	5.310	5.009
E: 5×10	6.778	< 0.005	6.716	5.453

(b)

Figure 9: (a) Revenue and regret of RegretNet on the validation set for auctions learned for Setting D using different architectures, where (R, K) denotes R hidden layers and K nodes per layer. (b) Test revenue and regret for Settings D and E, for the (5, 100) architecture.

Setting	Method	*rev*	*rgt*	*IR viol.*	Run-time
2×3	RegretNet	1.291	< 0.001	0	∼9 hrs
	LP (5 bins/value)	1.53	0.019	0.027	69 hrs

Figure 10: Test revenue, regret, IR violation, and running-time for RegretNet and an LP-based approach for a two bidder, three items setting with additive uniform valuations.

E. Five additive bidders and ten items, where bidders draw their value for each item independently from $U[0, 1]$.

The optimal auction for these settings is not known. However, running a separate Myerson auction for each item is optimal in the limit of the number of bidders [Palfrey, 1983]. For a regime with a small number of bidders, this provides a strong benchmark. We also compare to selling the grand bundle via a Myerson auction.

For Setting D, we show in Figure 9(a) the revenue and regret of the learned auction on a validation sample of 10,000 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 Setting D and Setting E. Figure 9(b) shows that the learned auctions yield higher revenue compared to the baselines, and do so with tiny regret.

5.7 Comparison to LP

Finally, we compare the running time of RegretNet with the LP approach proposed in [Conitzer and Sandholm, 2002, 2004]. To be able to run the LP, we consider a smaller setting with two additive bidders and three items, with item values drawn independently from $U[0, 1]$. For RegretNet we used two hidden layers, and 100 nodes per hidden layer. The LP was solved with the commercial solver *Gurobi*. We handled continuous valuations by discretizing the value into five bins per item (resulting in $\approx 10^5$ decision variables and $\approx 4 \times 10^6$ constraints) and rounding a continuous input valuation profile to the nearest discrete profile for evaluation.

The results are shown in Figure 10. 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 were not able to run an LP for this setting for finer discretizations in more than one week of compute time. In contrast, RegretNet yields much lower regret and no IR violations (as the neural network satisfies IR by design), and does so in just around nine hours. In fact, even for the larger Settings D–E, the running time of RegretNet was less than 13 hours.

6 Conclusion

Neural networks have been deployed successfully for exploration in other contexts, e.g., for the discovery of new drugs [Gómez-Bombarelli et al., 2018]. We believe that there is ample opportunity for applying deep learning in the context of economic design. We have demonstrated how standard pipelines can re-discover and surpass the analytical and computational progress in optimal auction design that has been made over the past 30-40 years. While our approach can easily solve problems that are orders of magnitude more complex than could previously be solved with the standard LP-based approach, a natural next step would be to scale this approach further to industry scale (e.g., through standardized benchmarking suites and innovations in network architecture). We also see promise for the framework in the present paper in advancing economic theory, for example in supporting or refuting conjectures and as an assistant in guiding new economic discovery.

References

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

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

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

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

E. Budish, Y.-K. Che, F. Kojima, and P. Milgrom. Designing random allocation mechanisms: Theory and applications. *American Economic Review*, 103(2):585–623, April 2013.

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

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

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

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

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

A. Choromanska, Y. LeCun, and G. Ben Arous. The landscape of the loss surfaces of multilayer networks. In *Proceedings of The 28th Conference on Learning Theory, Paris, France, July 3-6, 2015*, pages 1756–1760, 2015.

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

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

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

C. Daskalakis and S. M. Weinberg. Symmetries and optimal multi-dimensional mechanism design. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, Proceedings of the 13th ACM Conference on Economics and Computation, pages 370–387, 2012.

C. Daskalakis, A. Deckelbaum, and C. Tzamos. Mechanism design via optimal transport. In *Proceedings of the 14th ACM Conference on Electronic Commerce*, pages 269–286, 2013.

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

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

P. Dütting, Z. Feng, N. Golowich, H. Narasimhan, D. C. Parkes, and S. Ravindranath. Machine learning for optimal economic design. In J.-F. Laslier, H. Moulin, M. R. Sanver, and W. S. Zwicker, editors, *The Future of Economic Design*, volume 70, pages 495–515. Springer, 2019.

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

D. Fudenberg and A. Liang. Predicting and understanding initial play. *American Economic Review*, 2019. Forthcoming.

Y. Giannakopoulos and E. Koutsoupias. Duality and optimality of auctions for uniform distributions. In *SIAM Journal on Computing*, volume 47, pages 121–165, 2018.

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

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

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

Y. A. Gonczarowski and N. Nisan. Efficient empirical revenue maximization in single-parameter auction environments. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, pages 856–868, 2017.

Y. A. Gonczarowski and S. M. Weinberg. The sample complexity of up-to-ϵ multi-dimensional revenue maximization. In *59th IEEE Annual Symposium on Foundations of Computer Science*, pages 416–426, 2018.

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

N. Haghpanah and J. Hartline. When is pure bundling optimal? *Review of Economic Studies*, 2019. Revise and resubmit.

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

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

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

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

P. Jehiel, M. Meyer-ter-Vehn, and B. Moldovanu. Mixed bundling auctions. *Journal of Economic Theory*, 134(1):494–512, 2007.

K. Kawaguchi. Deep learning without poor local minima. In *Proceedings of the 30th Conference on Neural Information Processing Systems*, pages 586–594, 2016.

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

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

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

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

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

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

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

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

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

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

A. Patel, T. Nguyen, and R. Baraniuk. A probabilistic framework for deep learning. In *Proceedings of the 30th Conference on Neural Information Processing Systems*, pages 2550–2558, 2016.

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

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

J.-C. Rochet. A necessary and sufficient condition for rationalizability in a quasilinear context. *Journal of Mathematical Economics*, 16:191–200, 1987.

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

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

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

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

J. Sill. Monotonic networks. In *Proceedings of the 12th Conference on Neural Information Processing Systems*, pages 661–667, 1998.

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

A. Tacchetti, D.J. Strouse, M. Garnelo, T. Graepel, and Y. Bachrach. A neural architecture for designing truthful and efficient auctions. *CoRR*, abs/1907.05181, 2019.

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

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

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

A Additional Architectures

In this appendix we present our network architectures for multi-bidder single-item settings and for a general multi-bidder multi-item setting with combinatorial valuations.

A.1 The MyersonNet Approach

We start by describing an architecture that yields optimal DSIC auction for selling a single item to multiple buyers.

In the single-item setting, each bidder holds a private value $v_i \in \mathbb{R}_{\geq 0}$ for the item. We consider a randomized auction (g, p) that maps a reported bid profile $b \in \mathbb{R}^n_{\geq 0}$ to a vector of allocation probabilities $g(b) \in \mathbb{R}^n_{\geq 0}$, 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 $t_i : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$. The expected revenue of the auction, when bidders are truthful, is given by:

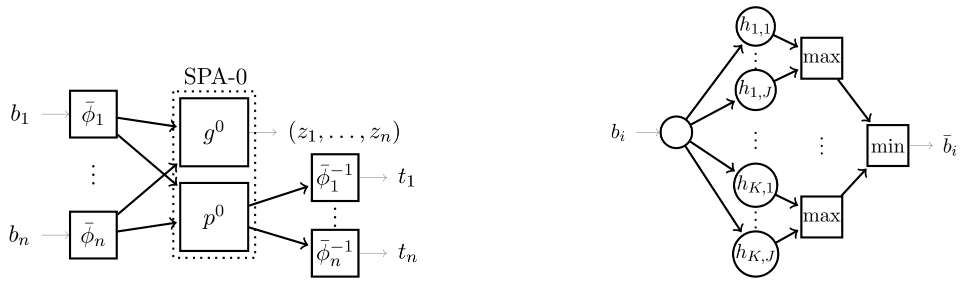
$$rev(g, p) \;=\; \mathbf{E}_{v\sim F}\Bigg[\sum_{i=1}^{n} g_i(v)\, t_i(v)\Bigg]. \tag{11}$$

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

Theorem A.1 (Myerson [1981])**.** *There exist a collection of monotonically non-decreasing functions, $\bar{\phi}_i : \mathbb{R}_{\geq 0} \to \mathbb{R}$ called the ironed virtual valuation functions such that the optimal BIC auction for selling a single item is the DSIC auction that assigns the item to the buyer with the highest ironed virtual value $\bar{\phi}_i(v_i)$ provided that this value is non-negative, with ties broken in an arbitrary value-independent manner, and charges the bidders according to $p_i(v_i) = v_i g_i(v_i) - \int_0^{v_i} g_i(t)\, dt$.*

For distribution F_i with density f_i the virtual valuation function is $\psi_i(v_i) = v_i - (1 - F(v_i))/f(v_i)$. A distribution F_i with density f_i is regular if ψ_i is monotonically non-decreasing. For regular distributions $F_1, \ldots, F_n$ no ironing is required and $\bar{\phi}_i = \psi_i$ for all i.

If the virtual valuation functions $\psi_1, \ldots, \psi_n$ are furthermore monotonically increasing and not only monotonically non-decreasing, the optimal 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, denoted (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 winning bidder i. In fact, this auction is DSIC for any choice of strictly monotone transformations of the values:



(a) **(b)**

Figure 11: (a) MyersonNet: The network applies monotone transformations $\bar{\phi}_1, \ldots, \bar{\phi}_n$ to the input bids, passes the virtual values to the SPA-0 network in Figure 12, and applies the inverse transformations $\bar{\phi}_1^{-1}, \ldots, \bar{\phi}_n^{-1}$ to the payment outputs. (b) Monotone virtual value function $\bar{\phi}_i$, where $h_{kj}(b_i) = e^{\alpha^i_{kj}} b_i + \beta^i_{kj}$.

Theorem A.2. *For any set of strictly monotonically increasing functions $\bar{\phi}_1, \ldots, \bar{\phi}_n$, 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, where (g^0, p^0) is the allocation and payment rule of a second price auction with zero reserve.*

For regular distributions with monotonically increasing virtual value functions designing an optimal DSIC auction thus reduces to finding the right strictly monotone transformations and corresponding inverses, and modeling a second price auction with zero reserve.

We present a high-level overview of a neural network architecture that achieves this in Figure 11(a), and describe the components of this network in more detail in Section A.1.1 and Section A.1.2 below.

Our MyersonNet is tailored to monotonically increasing virtual value functions. For regular distributions with virtual value functions that are not strictly increasing and for irregular distributions this approach only yields approximately optimal auctions.

A.1.1 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 $w^i_{kj} \in \mathbb{R}_{>0}$, $k = 1, \ldots, K$, $j = 1, \ldots, J$ and intercepts $\beta^i_{kj} \in \mathbb{R}$, $k = 1, \ldots, K$, $j = 1, \ldots, J$, we define:

$$\bar{\phi}_i(b_i) \;=\; \min_{k \in [K]} \max_{j \in [J]} w^i_{kj}\, b_i + \beta^i_{kj}.$$

Since each of the above linear function is strictly non-decreasing, so is $\bar{\phi}_i$. In practice, we can set each $w^i_{kj} = e^{\alpha^i_{kj}}$ for parameters $\alpha^i_{kj} \in [-B, B]$ in a bounded range. A graphical representation of the neural network used for this transform is shown in Figure 11(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 [Sill, 1998]. 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:

$$\bar{\phi}_i^{-1}(y) \;=\; \max_{k \in [K]} \min_{j \in [J]} e^{-\alpha^i_{kj}}(y - \beta^i_{kj}).$$

A.1.2 Modeling SPA with Zero Reserve

We also need to model a SPA with zero reserve (SPA-0) within the neural network structure. For the purpose of training, we employ a smooth approximation to the allocation rule using a

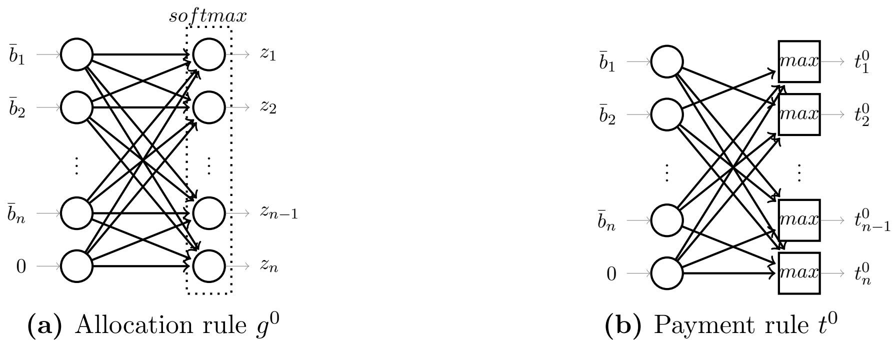


Figure 12: MyersonNet: SPA-0 network for (approximately) modeling a second price auction with zero reserve price. The inputs are (virtual) bids $\bar{b}_1, \ldots, \bar{b}_n$ and the output is a vector of assignment probabilities $z_1, \ldots, z_n$ and prices (conditioned on allocation) $t_1^0, \ldots, t_n^0$.

neural network. Once we learn value functions using this approximate allocation rule, we use them together with an exact SPA with zero reserve to construct the final auction.

The SPA-0 allocation rule g^0 can be approximated using a 'softmax' function on the virtual values $\bar{b}_1, \ldots, \bar{b}_n$ and an additional dummy input $\bar{b}_{n+1} = 0$:

$$g_i^0(\bar{b}) \;=\; \frac{e^{\kappa \bar{b}_i}}{\sum_{j=1}^{n+1} e^{\kappa \bar{b}_j}}, \;\; i \in N, \tag{12}$$

where $\kappa > 0$ is a constant fixed a priori, and determines the quality of the approximation. The higher the value of κ, the better the approximation but the less smooth the resulting allocation function.

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

$$t_i^0(\bar{b}) \;=\; \max\big\{ \max_{j \neq i} \bar{b}_j,\, 0 \big\}, \;\; i \in N. \tag{13}$$

Let $g^{\alpha,\beta}$ and $t^{\alpha,\beta}$ denote the allocation and conditional payment rules for the overall auction in Figure 11(a), where (α, β) are the parameters of the forward monotone transform. Given a sample of valuation profiles $\mathcal{S} = \{v^{(1)}, \ldots, v^{(L)}\}$ drawn i.i.d. from F, we optimize the parameters using the negated revenue on $\mathcal{S}$ as the error function, where the revenue is approximated as:

$$\widehat{rev}(g, t) \;=\; \frac{1}{L} \sum_{\ell=1}^{L} \sum_{i=1}^{n} g_i^{\alpha,\beta}(v^{(\ell)})\, t_i^{\alpha,\beta}(v^{(\ell)}). \tag{14}$$

We solve this training problem using a minibatch stochastic gradient descent solver.

A.2 RegretNet for Combinatorial Valuations

We next show how to adjust the RegretNet architecture so that it can handle bidders with general, combinatorial valuations. In the present work, we develop this architecture only for small number of items.[15] In this case, each bidder i reports a bid $b_{i,S}$ for every bundle of items $S \subseteq M$ (except the empty bundle, for which her valuation is taken as zero). The allocation network has an output

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

$z_{i,S} \in [0, 1]$ 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 one. We also require that the total allocation to a bidder is at most one:

$$\sum_{i \in N} \sum_{S \subseteq M : j \in S} z_{i,S} \leq 1, \forall j \in M; \tag{15}$$

$$\sum_{S \subseteq M} z_{i,S} \leq 1, \forall i \in N. \tag{16}$$

We refer to an allocation that satisfies constraints (15)–(16) as being *combinatorial feasible*. To enforce these constraints, the allocation network computes a set of scores for each bidder and a set of scores for each item. Specifically, there is a group of bidder-wise scores $s_{i,S}, \forall S \subseteq M$ for each bidder $i \in N$, and a group of item-wise scores $s^{(j)}_{i,S}, \forall i \in N$, $S \subseteq M$ for each item $j \in M$. Let $s, s^{(1)}, \ldots, s^{(m)} \in \mathbb{R}^{n \times 2^m}$ denote these bidder scores and item scores. Each group of scores is normalized using a softmax function: $\bar{s}_{i,S} = \exp(s_{i,S}) / \sum_{S'} \exp(s_{i,S'})$ and $\bar{s}^{(j)}_{i,S} = \exp(s^{(j)}_{i,S}) / \sum_{i',S'} \exp(s^{(j)}_{i',S'})$. The allocation for bidder i and bundle $S \subseteq M$ is defined as the minimum of the normalized bidder-wise score $\bar{s}_{i,S}$ and the normalized item-wise scores $\bar{s}^{(j)}_{i,S}$ for each $j \in S$:

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

Similar to the unit-demand setting, we first show that $\varphi^{CF}(s, s^{(1)}, \ldots, s^{(m)})$ is combinatorial feasible and that our constructive approach is without loss of generality. See Appendix D.5 for a proof.

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

Then we show that any combinatorial feasible matrix corresponds to lotteries over one-to-one assignments. This can be shown through the following decomposition lemma for combinatorial feasible matrices. See Appendix D.6 for the proof.[16]

Lemma A.2. *Any combinatorial matrix $A \in \mathbb{R}^{n \times 2^m}$ can be represented as a convex combination of matrices $B^1, \ldots, B^k$ where each $B^\ell \in \{0, 1\}^{n \times 2^m}$, and for each B^ℓ, $\sum_{i \in N} \sum_{S \subseteq M : j \in S} B^\ell_{i,S} \leq 1, \forall j \in M$, and $\sum_{S \subseteq M} B^\ell_{i,S} \leq 1, \forall i \in N$.*

Figure 13 shows the network architecture for a setting with two bidders and two items. For ease of exposition, we ignore the empty bundle. 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 can be assigned. The network also computes four scores for item 1: $s^1_{1,\{1\}}, s^1_{2,\{1\}}, s^1_{1,\{1,2\}}$, and $s^1_{2,\{1,2\}}$, one for each assignment where item 1 is present, and similarly, four scores for item 2: $s^2_{1,\{2\}}, s^2_{2,\{2\}}, s^2_{1,\{1,2\}}$, and $s^2_{2,\{1,2\}}$. Each set of scores is normalized by separate softmax functions. The final allocation for each bidder i is:

$$\begin{aligned} z_{i,\{1\}} &= \min\{\bar{s}_{i,\{1\}},\, \bar{s}^1_{i,\{1\}}\}, \\ z_{i,\{2\}} &= \min\{\bar{s}_{i,\{2\}},\, \bar{s}^2_{i,\{2\}}\}, \text{and} \\ z_{i,\{1,2\}} &= \min\{\bar{s}_{i,\{1,2\}},\, \bar{s}^1_{i,\{1,2\}},\, \bar{s}^2_{i,\{1,2\}}\}. \end{aligned}$$

The payment network for combinatorial bidders has the same structure as the one in Figure 2, computing a fractional payment $\tilde{p}_i \in [0, 1]$ for each bidder i using a sigmoidal unit, and outputting a payment $p_i = \tilde{p}_i \sum_{S \subseteq M} z_{i,S}\, b_{i,S}$.

[16] Unlike in the unit-demand case, the proof of this lemma does not yield an algorithm whose running time is polynomial in n and m.

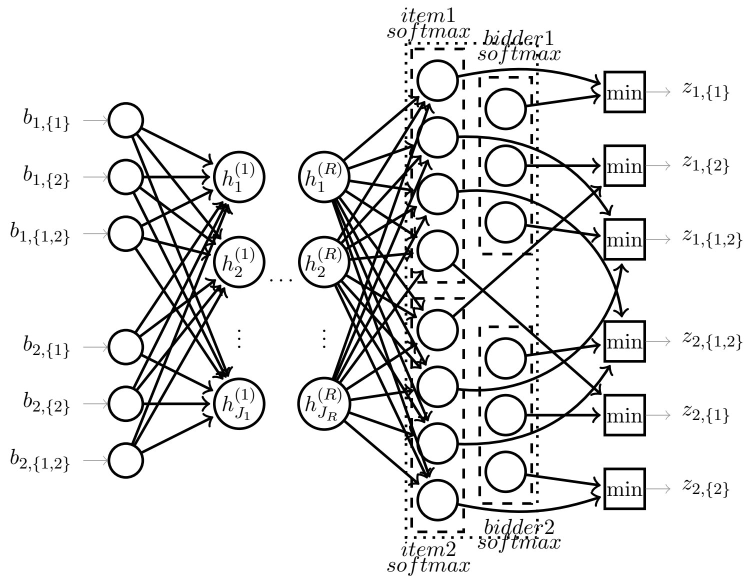


Figure 13: RegretNet: The allocation network for settings with two combinatorial bidders and two items.

B Additional Experiments

We present a broad range of additional experiments for the two main architectures used in the body of the paper, and additional ones for the architectures presented in Appendix A

B.1 Experiments with MyersonNet

We first evaluate the MyersonNet architecture introduced in Appendix A.1 for designing single-item auctions. We focus on settings with a small number of bidders because this is where revenue-optimal auctions are meaningfully different from efficient auctions. We present experimental results for the following four settings:

F. Three bidders with independent, regular, and symmetrically distributed valuations $v_i \sim U[0, 1]$.

G. Five bidders with independent, regular, and asymmetrically distributed valuations $v_i \sim U[0, i]$.

H. Three bidders with independent, regular, and symmetrically distributed valuations $v_i \sim Exp(3)$.

I. Three bidders with independent irregular distributions $F_{\text{irregular}}$, where each v_i is drawn from $U[0, 3]$ with probability 3/4 and from $U[3, 8]$ with probability 1/4.

We note that the optimal auctions for the first three distributions involve virtual value functions $\bar{\phi}_i$ that are strictly monotone. For the fourth and final distribution the optimal auction uses ironed virtual value functions that are not strictly monotone.

For the training set and test set we used 1,000 valuation profiles sampled i.i.d. from the respective valuation distribution. We modeled each transform $\bar{\phi}_i$ in the MyersonNet architecture using 5 sets of 10 linear functions, and we used $\kappa = 10^3$.

The results are summarized in Figure 14. 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.

Distribution	n	Opt	SPA	MyersonNet
		rev	*rev*	*rev*
Setting F	3	0.531	0.500	0.531
Setting G	5	2.314	2.025	2.305
Setting H	3	2.749	2.500	2.747
Setting I	3	2.368	2.210	2.355

Figure 14: The revenue of the single-item auctions obtained with MyersonNet.

B.2 Additional Experiments with RochetNet and RegretNet

In addition to the experiments with RochetNet and RegretNet on the single bidder, multi-item settings in Section 5.3 we also considered the following settings:

J. Single additive bidder with independent 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. [2017].

K. 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 due to Haghpanah and Hartline [2019].

L. Single unit-demand bidder with independent preferences over two items, where the item values $v_1, v_2 \sim U[0, 1]$. See [Pavlov, 2011] for the optimal mechanism.

We used RegretNet architectures with two hidden layers with 100 nodes each. The optimal allocation rules as well as a side-by-side comparison of those found by RochetNet and RegretNet are given in Figure 15. Figure 16 gives the revenue and regret achieved by RegretNet and the revenue achieved by RochetNet.

We find that in all three settings RochetNet recovers the optimal mechanism basically exactly, while RegretNet finds an auction that matches the optimal design to surprising accuracy.

B.3 Experiments with RegretNet with Combinatorial Valuations

We next compare our RegretNet architecture for combinatorial valuations described in Section A.2 to the computational results of Sandholm and Likhodedov [2015] for the following settings for which the optimal auction is not known:

M. Two additive bidders and two items, where bidders draw their value for each item independently from $U[0, 1]$.[17]

N. Two bidders and two items, with item valuations $v_{1,1}, v_{1,2}, v_{2,1}, v_{2,2}$ drawn independently from $U[1, 2]$ and set valuations $v_{1,\{1,2\}} = v_{1,1} + v_{1,2} + C_1$ and $v_{2,\{1,2\}} = v_{2,1} + v_{2,2} + C_2$, where C_1, C_2 are drawn independently from $U[-1, 1]$.

O. Two bidders and two items, with item valuations $v_{1,1}, v_{1,2}$ drawn independently from $U[1, 2]$, item valuations $v_{2,1}, v_{2,2}$ drawn independently from $U[1, 5]$, and set valuations $v_{1,\{1,2\}} = v_{1,1} + v_{1,2} + C_1$ and $v_{2,\{1,2\}} = v_{2,1} + v_{2,2} + C_2$, where C_1, C_2 are drawn independently from $U[-1, 1]$.

These settings correspond to Settings I.-III. described in Section 3.4 of Sandholm and Likhodedov [2015]. These authors conducted extensive experiments with several different classes of incentive compatible mechanisms, and different heuristics for setting the parameters of these auctions. They

[17] This setting can be handled by the non-combinatorial RegretNet architecture and is included here for comparison to Sandholm and Likhodedov [2015].

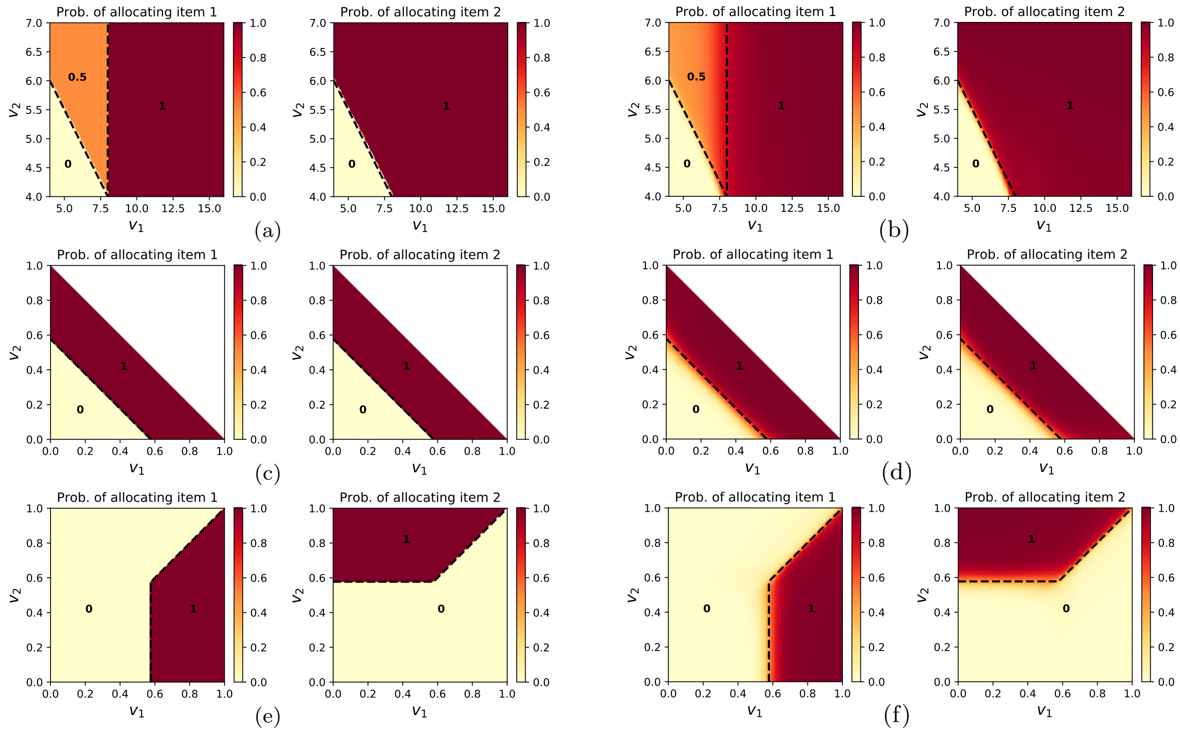


Figure 15: Side-by-side comparison of the allocation rules learned by RochetNet and RegretNet for single bidder, two items settings. Panels (a) and (b) are for Setting J, Panels (c) and (d) are for Setting K, and Panels (e) and (f) for Setting L. Panels describe the learned allocations for the two items (item 1 on the left, item 2 on the right). Optimal mechanisms are indicated via dashed lines and allocation probabilities in each region.

observed the highest revenue for two classes of mechanisms that generalize mixed bundling auctions and λ-auctions [Jehiel et al., 2007].

These two classes of mechanisms are the *Virtual Value Combinatorial Auctions* (VVCA) and *Affine Maximizer Auctions* (AMA). They also considered a restriction of AMA to bidder-symmetric auction (AMA_{bsym}). We use VVCA^*, AMA^*, and $\text{AMA}^*_{\text{bsym}}$ to denote the best mechanism in the respective class, as reported by Sandholm and Likhodedov and found using a heuristic grid search technique.

For Setting M and N, Sandholm and Likhodedov observed the highest revenue for $\text{AMA}^*_{\text{bsym}}$, and for Setting O the best performing mechanism was VVCA^*. Figure 17 compares the performance of RegretNet to that of these best performing, benchmark mechanisms. To compute the revenue of the benchmark mechanisms we used the parameters reported in [Sandholm and Likhodedov, 2015] (Table 2, p. 1011), and evaluated the respective mechanisms on the same test set used for RegretNet. Note that RegretNet is able to learn new auctions with improved revenue and tiny regret.

C Gradient-Based Regret Approximation

In Section 4, we describe a gradient-based approach to estimating a bidder's regret. We present a visualization of this approach in Figure 18 for a well-trained mechanism that has (almost) zero regret for Setting A. We consider a bidder with true valuation $(v_1, v_2) = (0.1, 0.8)$, represented as a green dot. The heat map represents the utility difference $u((v_1, v_2); (b_1, b_2)) - u((v_1, v_2); (v_1, v_2))$ for misreports $(b_1, b_2) \in [0, 1]^2$, with shades of yellow corresponding to low utility differences and

Distribution	Opt	RegretNet		RochetNet
	rev	*rev*	*rgt*	*rev*
Setting J	9.781	9.734	< 0.001	9.779
Setting K	0.388	0.392	< 0.001	0.388
Setting L	0.384	0.384	< 0.001	0.384

Figure 16: Test revenue and regret achieved by RegretNet and revenue achieved by RochetNet for Settings J–L.

Distribution	RegretNet		VVCA*	$\text{AMA}^*_{\text{bsym}}$
	rev	*rgt*	*rev*	*rev*
Setting M	0.878	< 0.001	—	0.862
Setting N	2.871	< 0.001	—	2.765
Setting O	4.270	< 0.001	4.209	—

Figure 17: Test revenue and regret for RegretNet for Settings M–O, and comparison with the best performing VVCA and AMA_{bsym} auctions as reported by Sandholm and Likhodedov [2015].

shades of blue corresponding to high utility differences. To estimate the regret of the bidder which is essentially zero in this case, we draw 10 random initial misreports from the underlying valuation distribution, represented as red dots, and then perform a sequence of gradient-descent steps on these random misreports. The figure shows the random initial misreports and then the 20 gradient-descent steps, plotting one in every four steps.

D Omitted Proofs

In this appendix we present formal proofs for all theorems and lemmas that were stated in the body of the paper or the other appendices. We first introduce some notation. We denote the inner product between vectors $a, b \in \mathbb{R}^d$ as $\langle a, b\rangle = \sum_{i=1}^d a_i b_i$. We denote the ℓ_1 norm for a vector x by $\|x\|_1$ and the induced ℓ_1 norm for a matrix $A \in \mathbb{R}^{k\times t}$ by $\|A\|_1 = \max_{1\le j\le t} \sum_{i=1}^k A_{ij}$.

D.1 Proof of Lemma 2.1

Let $f_i(v; w) := \max_{v'_i \in V_i} u_i^w(v_i; (v'_i, v_{-i})) - u_i^w(v_i; (v_i, v_{-i}))$. Then we have $rgt_i(w) = \mathbf{E}_{v\sim F}[f_i(v; w)]$. Rewriting the expected value, we have

$$rgt_i(w) = \int_0^\infty \mathbf{P}(f_i(v; w) \ge x)dx \ge \int_0^{rgt_i^q(w)} \mathbf{P}(f_i(v; w) \ge x)dx \ge q \cdot rgt_i^q(w),$$

where the last inequality holds because for any $0 < x < rgt_i^q(w)$, $\mathbf{P}(f_i(v; w) \ge x) \ge \mathbf{P}(f_i(v; w) \ge rgt_i^q(w)) = q$. □

D.2 Proof of Theorem 2.2

We present the proof for auctions with general, randomized allocation rules. A randomized allocation rule $g_i : V \to [0, 1]^{2^M}$ maps valuation profiles to a vector of allocation probabilities for bidder i, where $g_{i,S}(v) \in [0, 1]$ denotes the probability that the allocation rule assigns subset of items $S \subseteq M$ to bidder i, and $\sum_{S\subseteq M} g_{i,S}(v) \le 1$. This encompasses both the allocation rules for the combinatorial setting, and the allocation rules for the additive and unit-demand settings,

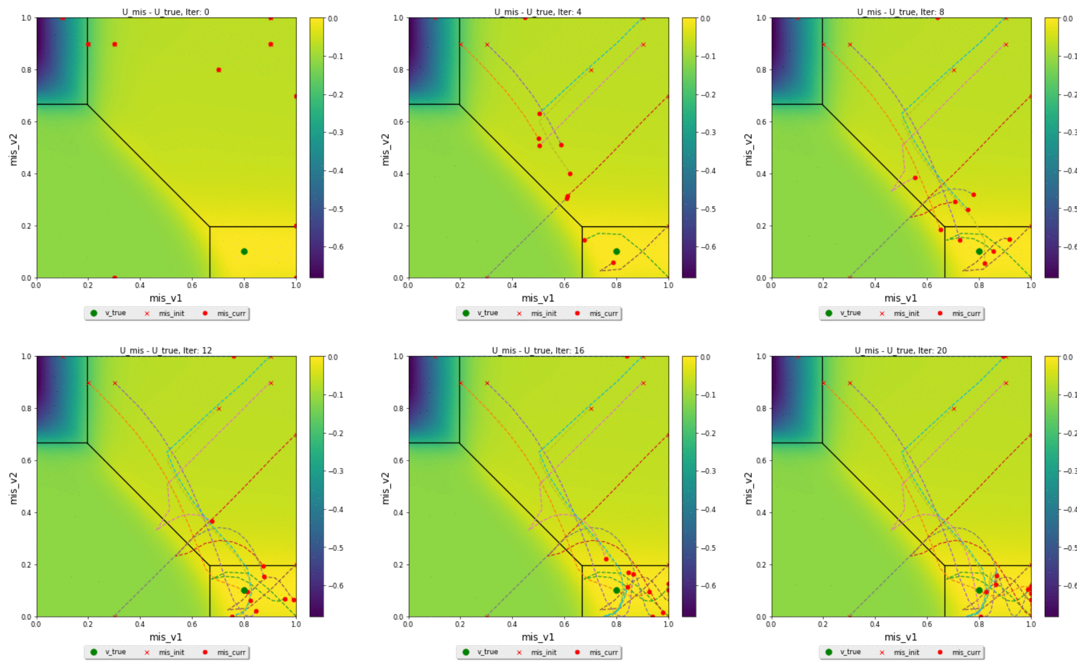


Figure 18: Visualization of the gradient-based approach to regret approximation for a well-trained auction for Setting A. The top left figure shows the true valuation (green dot) and ten random initial misreports (red dots). The remaining figures display 20 steps of gradient descent, showing one in every four steps.

which only output allocation probabilities for individual items. The payment function $p : V \to R^n$ maps valuation profiles to a payment for each bidder $p_i(v) \in \mathbb{R}$. For ease of exposition, we omit the superscripts "w". Recall that $\mathcal{M}$ is a class of auctions consisting of allocation and payment rules (g, p). As noted in the theorem statement, we will assume w.l.o.g. that for each bidder i, $v_i(S) \leq 1$, $\forall S \subseteq M$.

D.2.1 Definitions

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

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

and let $\mathcal{U}$ be the class of profile of utility functions defined on $\mathcal{M}$, i.e., the class of tuples $(u_1, \ldots, u_n)$ where each $u_i : V_i \times V \to \mathbb{R}$ and $u_i(v_i, b) = v_i(g(b)) - p_i(b), \forall i \in N$ for some $(g, p) \in \mathcal{M}$.

We will sometimes find it useful to represent the utility function as an inner product, i.e., treating v_i as a real-valued vector of length 2^M, we may write $u_i(v_i, b) = \langle v_i, g_i(b)\rangle - p_i(b)$.

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

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

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

Define the $\ell_{\infty,1}$ distance between two utility functions u and u' as

$$\max_{v,v'} \sum_i |u_i(v_i, (v_i', v_{-i})) - u_i(v_i, (v_i', v_{-i}))|$$

and let $\mathcal{N}_\infty(\mathcal{U}, \epsilon)$ denote the minimum number of balls of radius ϵ to cover $\mathcal{U}$ under this distance. Similarly, define the distance between u_i and u'_i as $\max_{v,v'_i} |u_i(v_i, (v'_i, v_{-i})) - u'_i(v_i, (v'_i, v_{-i}))|$, and let $\mathcal{N}_\infty(\mathcal{U}_i, \epsilon)$ denote the minimum number of balls of radius ϵ to cover $\mathcal{U}_i$ under this distance. Similarly, we define covering numbers $\mathcal{N}_\infty(\text{rgt} \circ \mathcal{U}_i, \epsilon)$ 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, $\mathcal{G}_i = \{g_i : V \to 2^M \,|\, g \in \mathcal{G}\}$. Similarly, we denote the class of payment functions by $\mathcal{P}$ and $\mathcal{P}_i = \{p_i : V \to \mathbb{R} \,|\, p \in \mathcal{P}\}$. 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 $\ell_{\infty,1}$ distance.

D.2.2 Auxiliary Lemma

We will use a lemma from [Shalev-Shwartz and Ben-David, 2014]. Let $\mathcal{F}$ denote a class of bounded functions $f : Z \to [-c, c]$ defined on an input space Z, for some $c > 0$. Let D be a distribution over Z and $\mathcal{S} = \{z_1, \ldots, z_L\}$ be a sample drawn i.i.d. from D. We are interested in the gap between the expected value of a function f and the average value of the function on sample S, and would like to bound this gap uniformly for all functions in $\mathcal{F}$. For this, we measure the capacity of the function class $\mathcal{F}$ using the empirical Rademacher complexity on sample S, defined below:

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

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

Lemma D.1 (Shalev-Shwartz and Ben-David [2014]). *Let $\mathcal{S} = \{z_1, \ldots, z_L\}$ be a sample drawn i.i.d. from some distribution D over Z. Then with probability of at least $1 - \delta$ over draw of $\mathcal{S}$ from D, for all $f \in \mathcal{F}$,*

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

D.2.3 Generalization Bound for Revenue

We first prove the generalization bound for revenue. For this, we define the following auxiliary function class, where each $f : V \to \mathbb{R}_{\geq 0}$measures the total payments from some mechanism in $\mathcal{M}$:

$$\text{rev} \circ \mathcal{M} \,=\, \left\{f : V \to \mathbb{R}_{\geq 0} \,\middle|\, f(v) \,=\, \textstyle\sum_{i=1}^n p_i(v), \text{ for some } (g, p) \in \mathcal{M}\right\}.$$

Note each function f in this class corresponds to a mechanism (g, p) in $\mathcal{M}$, and the expected value $\mathbf{E}_{v\sim D}[f(v)]$ gives the expected revenue from that mechanism. The proof then follows by an application of the uniform convergence bound in Lemma D.1 to the above function class, and by further bounding the Rademacher complexity term in this bound by the covering number of the auction class $\mathcal{M}$.

Applying Lemma D.1 to the auxiliary function class $\text{rev} \circ \mathcal{M}$, we get with probability of at least $1 - \delta$ over draw of L valuation profiles S from D, for any $f \in \text{rev} \circ \mathcal{M}$,

$$\begin{aligned} \mathbf{E}_{v\sim F}\Big[-\sum_{i\in N} p_i(v)\Big] \leq\; & -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i=1}^{n} p_i(v^{(\ell)}) \\ & + 2\hat{R}_L(\text{rev} \circ \mathcal{M}) \;+ Cn\sqrt{\frac{\log(1/\delta)}{L}}, \end{aligned} \tag{17}$$

All that remains is to bound the above empirical Rademacher complexity $\hat{R}_L(\mathrm{rev} \circ \mathcal{M})$ in terms of the covering number of the payment class $\mathcal{P}$ and in turn in terms of the covering number of the auction class $\mathcal{M}$. Since we assume that the auctions in $\mathcal{M}$ satisfy individual rationality and $v(S) \leq 1, \forall S \subseteq M$, we have for any v, $p_i(v) \leq 1$.

By the definition of the covering number for the payment class, there exists a cover $\hat{\mathcal{P}}$ for $\mathcal{P}$ of size $|\hat{\mathcal{P}}| \leq \mathcal{N}_\infty(\mathcal{P}, \epsilon)$ such that for any $p \in \mathcal{P}$, there is a $f_p \in \hat{\mathcal{P}}$ with $\max_v \sum_i |p_i(v) - f_{p_i}(v)| \leq \epsilon$. We thus have:

$$
\begin{aligned}
\hat{\mathcal{R}}_L(\mathrm{rev} \circ \mathcal{M}) &= \frac{1}{L}\mathbf{E}_\sigma\left[\sup_p \sum_{\ell=1}^{L} \sigma_\ell \cdot \sum_i p_i(v^{(\ell)})\right] \\
&= \frac{1}{L}\mathbf{E}_\sigma\left[\sup_p \sum_{\ell=1}^{L} \sigma_\ell \cdot \sum_i f_{p_i}(v^{(\ell)})\right] + \frac{1}{L}\mathbf{E}_\sigma\left[\sup_p \sum_{\ell=1}^{L} \sigma_\ell \cdot \sum_i p_i(v^{(\ell)}) - f_{p_i}(v^{(\ell)})\right] \\
&\leq \frac{1}{L}\mathbf{E}_\sigma\left[\sup_{\hat{p} \in \hat{\mathcal{P}}} \sum_{\ell=1}^{L} \sigma_\ell \cdot \sum_i \hat{p}_i(v^{(\ell)})\right] + \frac{1}{L}\mathbf{E}_\sigma \|\sigma\|_1 \epsilon \\
&\leq \sqrt{\sum_\ell (\sum_i \hat{p}_i(v^\ell))^2}\sqrt{\frac{2\log(\mathcal{N}_\infty(\mathcal{P}, \epsilon))}{L}} + \epsilon \\
&\leq 2n\sqrt{\frac{2\log(\mathcal{N}_\infty(\mathcal{P}, \epsilon))}{L}} + \epsilon, \qquad (18)
\end{aligned}
$$

where the second-last inequality follows from Massart's lemma, and the last inequality holds because

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

We further observe that $\mathcal{N}_\infty(\mathcal{P}, \epsilon) \leq \mathcal{N}_\infty(\mathcal{M}, \epsilon)$. By the definition of the covering number for the auction class $\mathcal{M}$, there exists a cover $\hat{\mathcal{M}}$ for $\mathcal{M}$ of size $|\hat{\mathcal{M}}| \leq \mathcal{N}_\infty(\mathcal{M}, \epsilon)$ such that for any $(g, p) \in \mathcal{M}$, there is a $(\hat{g}, \hat{p}) \in \hat{\mathcal{M}}$ such that for all v,

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

This also implies that $\sum_i |p_i(v) - \hat{p}_i(v)| \leq \epsilon$, and shows the existence of a cover for $\mathcal{P}$ of size at most $\mathcal{N}_\infty(\mathcal{M}, \epsilon)$.

Substituting the bound on the Radamacher complexity term in (18) in (17) and using the fact that $\mathcal{N}_\infty(\mathcal{P}, \epsilon) \leq \mathcal{N}_\infty(\mathcal{M}, \epsilon)$, we get:

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

which completes the proof.

D.2.4 Generalization Bound for Regret

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:

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

The proof then proceeds in three steps:

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

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

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

An application of Lemma D.1 then completes the proof. We prove each of the above steps below.

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

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

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

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

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

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,

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

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

This completes the proof of Step 1. □

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

Proof. Recall that 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 $(\hat{g}, \hat{p}) \in \hat{M}$ such that

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

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

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

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

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

This completes the proof of Step 2. □

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

Proof. 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 $\hat{u}$ such that 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 $\mathcal{N}_\infty(\overline{\mathrm{rgt}} \circ \mathcal{U}, \epsilon) \leq \mathcal{N}_\infty(\mathrm{rgt} \circ \mathcal{U}, \epsilon)$. Following Step 1, it is easy to show $\mathcal{N}_\infty(\mathrm{rgt} \circ \mathcal{U}, \epsilon) \leq \mathcal{N}_\infty(\mathcal{U}, \epsilon/2)$.

Together with Step 2 this completes the proof of Step 3. □

Based on the same arguments as in Section D.2.3, we can thus bound the empirical Rademacher complexity as:

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

Applying Lemma D.1, completes the proof of the generalization bound for regret. □

D.3 Proof of Theorem 3.1

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

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

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

D.4 Proof of Lemma 3.1

First, given the property of the 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 $c \in \mathbb{R}$). □

D.5 Proof of Lemma A.1

Similar to Lemma 3.1, $\varphi^{CF}(s, s^{(1)}, \ldots, s^{(m)})$ trivially satisfies the combinatorial feasibility (constraints (15)–(16)). For any allocation z that satisfies the combinatorial feasibility, the following scores

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

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

D.6 Proof of Lemma A.2

The proof is structured as follows, we first show a reduction from any combinatorial feasible matrix A to a doubly stochastic matrix A'. Then we apply the same proof technique in Lemma 3.2 to show the existence of decomposition.

Given a combinatorial feasible matrix $A = (A_{i,S})_{i\in N, S\subseteq M}$, let $\{S^j_k\}_k, \forall k \in [2^{m-1} - 1]$ represent all subsets belonging to M, s.t. $S^j_k \neq \{j\}$ and $j \in S^j_k$, we construct a corresponding matrix $A' \in \mathbb{R}^{(n\cdot m(2^{m-1}-1)+n)\times 2^m}$ in the following way,

- $\forall 1 \leq i \leq n, \; S \subseteq M, A'_{i,S} = A_{i,S}$.
- $\forall i > n, \; j \in [m]$,
  - If $n(j-1)(2^{m-1}-1)+n < i \leq nj(2^{m-1}-1)+n$, and $n(j-1)(k-1)+n < i < njk+n$, we denote
  $$A'_{i,\{j\}} = A_{i',S^j_k},$$
  where $i' \equiv i \mod i, i' \in [n]$.
- Otherwise, $A'_{i,S} = 0$.

We visualize this construction for 2-item, 2-bidder combinatorial valuations in Figure 19. We then show if A' is doubly stochastic, A is combinatorial feasible.

$$\begin{bmatrix} A_{1,\emptyset} & A_{1,\{1\}} & A_{1,\{2\}} & A_{1,\{1,2\}} \\ A_{2,\emptyset} & A_{2,\{1\}} & A_{2,\{2\}} & A_{2,\{1,2\}} \end{bmatrix} \Rightarrow A' = \begin{bmatrix} A_{1,\emptyset} & A_{1,\{1\}} & A_{1,\{2\}} & A_{1,\{1,2\}} \\ A_{2,\emptyset} & A_{2,\{1\}} & A_{2,\{2\}} & A_{2,\{1,2\}} \\ 0 & A_{1,\{1,2\}} & 0 & 0 \\ 0 & A_{2,\{1,2\}} & 0 & 0 \\ 0 & 0 & A_{1,\{1,2\}} & 0 \\ 0 & 0 & A_{2,\{1,2\}} & 0 \end{bmatrix}$$

Figure 19: Reduction from a combinatorial feasible matrix A to a doubly stochastic matrix A' for 2-item, 2-bidder combinatorial valuations.

First, we copy A to the first n rows in A', since A' is doubly stochastic, then $\forall i \in [n], \sum_{S\subseteq M} A_{i,S} = \sum_{S\subseteq M} A'_{i,S} \leq 1$. Second, for each item j, we have $\sum_i A'_{i,\{j\}} \leq 1$ because A' is doubly stochastic. Based on our construction,

$$\forall j \in [m], \sum_i A'_{i,\{j\}} = \sum_{i\in N} \sum_{S\subseteq M: j\in S} A_{i,S} \leq 1.$$

Finally, we show for all the other rows and columns in A', we don't import additional constraints for A. For the columns $S \neq \{j\}, \forall j \in [m]$, all the new entries are 0, the doubly stochasticity of A' is guaranteed by feasibility of A. For the rows $i > n$, there is only one non-zero entry in each row, the doubly stochasticity only requires this entry is upper bounded by 1, which is trivially satisfied in A.

By Lemma 3.2, A' can be represented as a convex combination of matrices $\bar{B}^1, \bar{B}^2, \cdots, \bar{B}^k$ where $\bar{B}^\ell \in \mathbb{R}^{n\cdot m(2^{m-1}-1)+n)\times 2^m}$. Let B^ℓ be the truncation of the first n rows of $\bar{B}^\ell$, it is straightforward to show that A can be represented as the convex combination of matrices $B^1, \cdots, B^k$. Since each $\bar{B}^\ell$ is doubly stochastic, B^ℓ is combinatorial feasible based on our proof above. □

D.7 Proof of Theorem 3.2

In Theorem 3.2, we only show the bounds on Δ_L for RegretNet with additive and unit-demand bidders. We restate this theorem so that it also bounds Δ_L for the general combinatorial valuations setting. Recall that the ℓ_1 norm for a vector x is denoted by $\|x\|_1$ and the induced ℓ_1 norm for a matrix $A \in \mathbb{R}^{k\times t}$ is denoted by $\|A\|_1 = \max_{1\leq j\leq t} \sum_{i=1}^k A_{ij}$.

Theorem D.1. *For RegretNet with R hidden layers, K nodes per hidden layer, d_g parameters in the allocation network, d_p parameters in the payment network, and the vector of all model parameters $\|w\|_1 \leq W$, the following are the bounds on the term Δ_L for different bidder valuation types:*

(a) additive valuations:
$\Delta_L \leq O\big(\sqrt{R(d_g + d_p)\log(LW\max\{K, mn\})/L}\big)$,

(b) unit-demand valuations:
$\Delta_L \leq O\big(\sqrt{R(d_g + d_p)\log(LW\max\{K, mn\})/L}\big)$,

(c) combinatorial valuations:
$\Delta_L \leq O\big(\sqrt{R(d_g + d_p)\log(LW\max\{K, n\,2^m\})/L}\big)$.

We first bound the covering number for a general feed-forward neural network and specialize it to the three architectures we present in Section 3 and Appendix A.2.

Lemma D.2. *Let $\mathcal{F}_k$ be a class of feed-forward neural networks that maps an input vector $x \in \mathbb{R}^{d_0}$ to an output vector $y \in \mathbb{R}^{d_k}$, with each layer ℓ containing T_ℓ nodes and computing $z \mapsto \phi_\ell(w^\ell z)$,* *where each $w^\ell \in \mathbb{R}^{T_\ell \times T_{\ell-1}}$ and $\phi_\ell : \mathbb{R}^{T_\ell} \to [-B, +B]^{T_\ell}$. Further let, for each network in $\mathcal{F}_k$, let the parameter matrices $\|w^\ell\|_1 \leq W$ 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 ϵ_0/d, where we will choose $\epsilon_0 > 0$ at the end of the proof. We will use $\hat{\mathcal{F}}_k$ to denote the subset of neural networks in $\mathcal{F}_k$ whose parameters are in the range $\{-(\lceil Wd/\epsilon_0 \rceil - 1)\,\epsilon_0/d, \ldots, -\epsilon_0/d, 0, \epsilon_0/d, \ldots, \lceil Wd/\epsilon_0 \rceil \epsilon_0/d\}$. The size of $\hat{\mathcal{F}}_k$ is at most $\lceil 2dW/\epsilon_0 \rceil^d$. We shall now show that $\hat{\mathcal{F}}_k$ is an ϵ-cover for $\mathcal{F}_k$.

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

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

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

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

$$\|w_\ell - \hat{w}_\ell\|_1 = \max_{1 \leq j \leq T_{\ell-1}} \sum_{i=1}^{T_\ell} |w^\ell_{\ell,i,j} - \hat{w}_{\ell,i,j}| \leq T_\ell \epsilon_0/d \leq \epsilon_0. \tag{19}$$

We then have:

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

where the second line follows from our assumption on ϕ_{k+1}, and the sixth line follows from our inductive hypothesis and from (19). By choosing $\epsilon_0 = \frac{\epsilon}{B(2\Phi W)^k}$, we complete the proof. □

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

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

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

$$\sup_b \sum_{i,j} |g_{ij}(b) - \hat{g}_{ij}(b)| \leq \epsilon/3 \tag{20}$$

$$\sup_b \sum_{i} |\alpha_i(b) - \hat{\alpha}_i(b)| \leq \epsilon/3. \tag{21}$$

We now show that the class of mechanism $\hat{\mathcal{M}} = \{(\hat{g}, \hat{\alpha}) \,|\, \hat{g} \in \hat{\mathcal{G}}, \text{ and } \hat{p}(b) = \hat{\alpha}_i(b) \cdot \langle v_i, \hat{g}_i(b) \rangle\}$ is an ϵ-cover for $\mathcal{M}$ under the $\ell_{1,\infty}$ distance. For any mechanism in $(g,p) \in \mathcal{M}$, let $(\hat{g}, \hat{p}) \in \hat{\mathcal{M}}$ be a mechanism in $\hat{\mathcal{M}}$ that satisfies (21). We have:

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

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. □

We are now ready to prove covering number bounds for the three architectures in Section 3 and Appendix A.2.

Proof of Theorem D.1. All three architectures use the same feed-forward architecture for computing fractional payments, consisting of K hidden layers with tanh activation functions. We also have by our assumption that the ℓ_1 norm of the vector of all model parameters is at most W, for each $\ell = 1, \ldots, R+1$, $\|w_\ell\|_1 \leq W$. Using that fact that the tanh activation functions are 1-Lipschitz and bounded in $[-1, 1]$, and there are at most $\max\{K, n\}$ number of nodes in any layer of the payment network, we have by an application of Lemma D.2 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 D.2. For ease of exposition, we omit the dummy scores used in the final layer of neural network architectures.

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

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

$$
\begin{aligned}
&\|\phi_{R+1}(s) - \phi_{R+1}(s')\|_1 \\
&\quad\leq \sum_j \left\|\text{softmax}(s_{1,j}, \ldots, s_{n,j}) - \text{softmax}(s'_{1,j}, \ldots, s'_{n,j})\right\|_1 \\
&\quad\leq \sqrt{n}\sum_j \left\|\text{softmax}(s_{1,j}, \ldots, s_{n,j}) - \text{softmax}(s'_{1,j}, \ldots, s'_{n,j})\right\|_2 \\
&\quad\leq \sqrt{n}\frac{\sqrt{n-1}}{n}\sum_j \sqrt{\sum_i \|s_{ij} - s'_{ij}\|^2} \\
&\quad\leq \sum_j \sum_i |s_{ij} - s'_{ij}| \qquad (22)
\end{aligned}
$$

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

The hidden layers $\ell = 1, \ldots, R$ are standard feed-forward layers with tanh activations. Since the tanh activation function is 1-Lipschitz, $\|\phi_\ell(s) - \phi_\ell(s')\|_1 \leq \|s - s'\|_1$. We also have by our assumption that the ℓ_1 norm of the vector of all model parameters is at most W, for each $\ell = 1, \ldots, R+1$, $\|w_\ell\|_1 \leq W$. Moreover, the output of each hidden layer node is in $[-1, 1]$, the output layer nodes is in $[0, 1]$, and the maximum number of nodes in any layer (including the output layer) is at most $\max\{K, mn\}$.

By an application of Lemma D.2 with $\Phi = 1$, $B = 1$, and $d = \max\{K, mn\}$ we have

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

where d_g is the number of parameters in allocation networks.

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

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

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

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

$$+ \sum_j \left\|\text{softmax}(\tilde{s}_{1,j}, \ldots, \tilde{s}_{n,j}) \,-\, \text{softmax}(\tilde{s}'_{1,j}, \ldots, \tilde{s}'_{n,j})\}\right\|_1$$

$$\leq \sum_{i,j} |s_{ij} - s'_{ij}| \,+\, \sum_{i,j} |\tilde{s}_{ij} - \tilde{s}'_{ij}|,$$

where the last step can be derived in the same way as (22).

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

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

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

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

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

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

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

where the last step can be derived in the same way as (22).

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

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

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

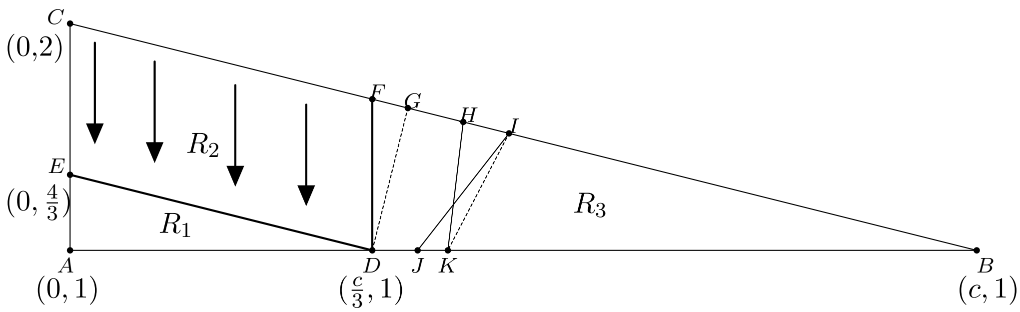


Figure 20: The transport of transformed measure of each region in Setting C.

.

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

(a) For additive bidders, choosing $\epsilon = \frac{1}{\sqrt{L}}$, we get

$$\Delta_L \leq O\left(\sqrt{R(d_p + d_g)\frac{\log(W \max\{K, mn\}L)}{L}}\right)$$

(b) For unit-demand bidders, choosing $\epsilon = \frac{1}{\sqrt{L}}$, we get

$$\Delta_L \leq O\left(\sqrt{R(d_p + d_g)\frac{\log((W \max\{K, mn\}L)}{L}}\right)$$

(c) For combinatorial bidders, choosing $\epsilon = \frac{1}{\sqrt{L}}$, we get

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

□

D.8 Proof of Theorem 5.1

We apply the duality theory of [Daskalakis et al., 2013] to verify the optimality of our proposed mechanism (motivated by empirical results of RochetNet). For the completeness of presentation, we provide a brief introduction of their approach here.

Let $f(v)$ be the joint valuation distribution of $v = (v_1, v_2, \cdots, v_m)$, V be the support of $f(v)$ and define the measure μ with the following density,

$$\mathbb{I}_{v=\bar{v}} + \mathbb{I}_{v\in\partial V} \cdot f(v)(v \cdot \hat{n}(v)) - (\nabla f(v) \cdot v + (m+1)f(v)) \tag{23}$$

where $\bar{v}$ is the "base valuation", i.e. $u(\bar{v}) = 0$, ∂V denotes the boundary of V, $\hat{n}(v)$ is the outer unit normal vector at point $v \in \partial V$, and m is the number of items. Let $\Gamma_+(X)$ be the unsigned (Radon) measures on X. Consider an unsinged measure $\gamma \in \Gamma_+(X \times X)$, let γ_1 and γ_2 be the two marginal measures of γ, i.e. $\gamma_1(A) = \gamma(A \times X)$ and $\gamma_2(A) = \gamma(X \times A)$ for all measurable sets

$A \subseteq X$. We say measure α dominates β if and only if for all (non-decreasing, convex) functions u, $\int u\, d\alpha \geq \int u\, d\beta$. Then by strong duality theory we have

$$\sup_u \int_V u\, d\mu = \inf_{\gamma \in \Gamma_+(V,V), \gamma_1 - \gamma_2 \succeq \mu} \int_{V \times V} \|v - v'\|_1\, d\gamma, \tag{24}$$

and both the supremum and infimum are achieved. Based on "complementary slackness" of linear programming, the optimal solution of Equation 24 needs to satisfy the following conditions.

Corollary D.1 (Daskalakis et al. [2017]). *Let u^* and γ^* be feasible for their respective problems in Equation 24, then $\int u^*\, d\mu = \int \|v - v'\|_1\, d\gamma^*$ if and only if the following two conditions hold:*

$$\int u^*\, d(\gamma_1^* - \gamma_2^*) = \int u^*\, d\mu$$

$$\int u^*(v) - u^*(v') = \|v - v'\|_1, \gamma^*\text{-almost surely.}$$

Then we prove the utility function u^* induced by the mechanism for setting C is optimal. Here we only focus on Settiong C with $c > 1$, for $c \leq 1$ the proof is analogous and we omit here[18]. The transformed measure μ of the valuation distribution is composed of:

1. A point mass of $+1$ at $(0, 1)$.
2. Mass -3 uniformly distributed throughout the triangle area (density $-\frac{6}{c}$).
3. Mass -2 uniformly distributed on lower edge of triangle (density $-\frac{2}{c}$).
4. Mass $+4$ uniformly distributed on right-upper edge of triangle (density $+\frac{4}{\sqrt{1+c^2}}$).

It is straightforward to verify that $\mu(R_1) = \mu(R_2) = \mu(R_3) = 0$. We will show there exists an optimal measure γ^* for the dual program of Theorem 2 (Equation 5) in [Daskalakis et al., 2013]. γ^* can be decomposed into $\gamma^* = \gamma^{R_1} + \gamma^{R_2} + \gamma^{R_3}$ with $\gamma^{R_1} \in \Gamma_+(R_1 \times R_1), \gamma^{R_2} \in \Gamma_+(R_2 \times R_2), \gamma^{R_3} \in \Gamma_+(R_3 \times R_3)$. We will show the feasibility of γ^*, such that

$$\gamma_1^{R_1} - \gamma_2^{R_1} \succeq \mu|_{R_1}; \quad \gamma_1^{R_2} - \gamma_2^{R_2} \succeq \mu|_{R_2}; \quad \gamma_1^{R_3} - \gamma_2^{R_3} \succeq \mu|_{R_3}. \tag{25}$$

Then we show the conditions of Corollary 1 in [Daskalakis et al., 2013] hold for each of the measures $\gamma^{R_1}, \gamma^{R_2}, \gamma^{R_3}$ separately, such that $\int u^* d(\gamma_1^A - \gamma_2^A) = \int_A u^* d\mu$ and $u^*(v) - u^*(v') = \|v - v'\|_1$ hold γ^A-almost surely for any $A = R_1, R_2$, and R_3. We visualize the transport of measure γ^* in Figure 20.

Construction of γ^{R_1}. $\mu_+|_{R_1}$ is concentrated on a single point $(0, 1)$ and $\mu_-|_{R_1}$ is distributed throughout a region which is coordinate-wise greater than $(0, 1)$, then it is obviously to show $0 \succeq \mu|_{R_1}$. We set γ^{R_1} to be zero measure, and we get $\gamma_1^{R_1} - \gamma_2^{R_1} = 0 \succeq \mu|_{R_1}$. In addition, $u^*(v) = 0, \forall v \in R_1$, then the conditions in Corollary 1 in [Daskalakis et al., 2013] hold trivially.

Construction of γ^{R_2}. $\mu_+|_{R_2}$ is uniformly distributed on upper edge CF of the triangle and $\mu_-|_{R_2}$ is uniformly distributed in R_2. Since we have $\mu(R_2) = 0$, we construct γ^{R_2} by "transporting" $\mu_+|_{R_2}$ into $\mu_-|_{R_2}$ downwards, that is $\gamma_1^{R_2} = \mu_+|_{R_2}, \gamma_2^{R_2} = \mu_-|_{R_2}$. Therefore, $\int u^* d(\gamma_1^{R_2} - \gamma_2^{R_2}) = \int u^* d\mu$ holds trivially. The measure γ^{R_2} is only concentrated on the pairs (v, v') such that $v_1 = v_1', v_2 \geq v_2'$. Thus for such pairs $(v'v')$, we have $u^*(v) - u^*(v') = (\frac{v_1}{c} + v_2 - \frac{4}{3}) - (\frac{v_1}{c} + v_2' - \frac{4}{3}) = ||v - v'||_1$.

Construction of γ^{R_3}. It is intricate to directly construct γ^{R_3} analytically, however, we will show there the optimal measure γ^{R_3} only transports mass from $\mu_+|_{R_3}$ to $\mu_-|_{R_3}$ leftwards and

[18] It is fairly similar to the proof for setting $c > 1$. If $c \leq 1$, there are only two regions to discuss, in which R_1 and R_2 are the regions correspond to allocation $(0, 0)$ and $(1, 1)$, respectively. Then we show the optimal $\gamma^* = \bar{\gamma}^{R_1} + \bar{\gamma}^{R_2}$ where $\bar{\gamma}^{R_1} = 0$ for region R_1 and show γ^{R_2} only "transports" mass of measure downwards and leftwards in region R_2, which is analgous to the analysis for γ^{R_3} for setting $c > 1$.

downwards. Let's consider a point H on edge BF with coordinates (v_1^H, v_2^H). Define the regions $R_L^H = \{v' \in R_3 | v_1' \le v_1^H\}$ and $R_U^H = \{v' \in R_3 | v_2' \ge v_2^H\}$. Let $\ell(\cdot)$ represent the length of segment, then we have $\ell(FH) < \frac{2}{3\sqrt{c^2+1}}$. Thus,

$$\mu(R_U^H) = \frac{4\ell(FH)}{\sqrt{c^2+1}} - \frac{6}{c} \cdot \frac{\ell^2(FH)c}{2(c^2+1)} = \frac{\ell(FH)}{\sqrt{c^2+1}} \cdot \left(4 - \frac{3\ell(FH)}{\sqrt{c^2+1}}\right) > 0$$

$$\begin{aligned}\mu(R_L^H) &= \frac{4\ell(FH)}{\sqrt{c^2+1}} - \frac{2}{c} \cdot \frac{\ell(FH)c}{\sqrt{c^2+1}} - \frac{6}{c} \cdot \left(\frac{2\ell(FH)c}{3\sqrt{c^2+1}} - \frac{\ell^2(FH)c}{2(c^2+1)}\right) \\ &= \frac{\ell(FH)}{\sqrt{c^2+1}} \cdot \left(\frac{3\ell(FH)}{\sqrt{c^2+1}} - 2\right) < 0\end{aligned}$$

Thus, there exists a unique line l_H with positive slope that intersects H and separate R_3 into two parts, R_U^H (above l_H) and R_B^H (below l_H), such that $\mu_+(R_U^H) = \mu_-(R_U^H)$. We will then show for any two points on edge BF, H and I, lines l_H and l_I will not intersect inside R_3. In Figure 20, on the contrary, we assume $l_H = HK$ and $l_I = IJ$ intersects inside R_3. Given the definition of l_H and l_I, we have

$$\mu_+(FHKD) = \mu_-(FHKD); \quad \mu_+(FIJD) = \mu_-(FIJD)$$

Since μ_+ is only distributed along the edge BF, we have

$$\mu_+(FIKD) = \mu_+(FIJD) = \mu_-(FIJD)$$

Notice μ_- is only distributed inside R_3 and edge DB, thus $\mu_-(FIKD) > \mu_-(FIJD)$. Given the above discussion, we have

$$\begin{aligned}\mu_+(HIK) &= \mu_+(FIJD) - \mu_+(FHKD) = \mu_-(FIJD) - \mu_-(FHKD) \\ &< \mu_-(FIKD) - \mu_-(FHKD) = \mu_-(HIK)\end{aligned} \tag{26}$$

On the other hand, let $S(HIK)$ be the area of triangle HIK, DG be the altitude of triangle DBF w.r.t BF, and h be the altitude of triangle HJK w.r.t the base HI.

$$\begin{aligned}\mu_-(HJK) &= \frac{6}{c} \cdot S(HIK) = \frac{6}{c} \cdot \frac{1}{2}\ell(HI)h \le \frac{3}{c} \cdot \ell(HI) \cdot \ell(DG) \\ &= \frac{3}{c} \cdot \frac{2c}{3\sqrt{c^2+1}} \cdot \ell(HI) = \frac{2}{\sqrt{c^2+1}} \cdot \ell(HI) \\ &< \frac{2}{\sqrt{c^2+1}} \cdot \ell(HI) = \mu_+(HIK),\end{aligned}$$

which is a contradiction of Equation 26. Thus, we show l_H and l_I doesn't intersect inside R_3. Let γ^{R_3} be the measure that transport mass from $\mu_+|_{R_3}$ to $\mu_-|_{R_3}$ through lines $\{l_H | H \in BF\}$. Then we have $\gamma_1^{R_3} = \mu_+|_{R_3}, \gamma_2^{R_3} = \mu_-|_{R_3}$, which leads to $\int u^* d(\gamma_1^{R_3} - \gamma_2^{R_3}) = \int u^* d\mu$. The measure γ^{R_3} is only concentrated on the pairs (v, v'), s.t. $v_1 \ge v_1'$ and $v_2 \ge v_2'$. Therefore, for such pairs (v, v'), we have $u^*(v) - u^*(v') = (v_1 + v_2 - \frac{c}{3} - 1) - (v_1' + v_2' - \frac{c}{3} - 1) = (v_1 - v_1') + (v_2 - v_2') = ||v - v'||_1$.

Finally, we show there must exist an optimal measure γ for the dual program of Theorem 2 in [Daskalakis et al., 2013]. □