Optimal Auctions through Deep Learning

[Extended Abstract]

Paul Dütting*
Department of Mathematics,
London School of Economics
Houghton Street
London WC2A 2AE, UK
p.d.duetting@lse.ac.uk

Zhe Feng†
SEAS, Harvard University
33 Oxford Street
Cambridge, MA 02138, USA
zhe_feng@g.harvard.edu

Harikrishna Narasimhan
Google Research
1600 Amphitheatre Way
Mountain View, CA 94043, USA
hnarasimham@google.com

David C. Parkes
SEAS, Harvard University
33 Oxford Street
Cambridge, MA 02138, USA
parkes@eecs.harvard.edu

Sai S. Ravindranath
SEAS, Harvard University
33 Oxford Street
Cambridge, MA 02138, USA
saisr@g.harvard.edu

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. We overview recent research results that show how tools from deep learning are shaping up to be become a powerful tool for the automated design of optimal auctions. In this approach, an auction is modeled as a multi-layer neural network, with optimal auction design framed as a constrained learning problem, and solved through standard machine learning pipelines. Through this approach, it is possible to recover essentially all known analytical solutions for multi-item settings, and obtain 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 function over subsets of items, drawn independently from not necessarily identical distributions. The auctioneer knows the value distribution, but not the actual valuations (willingness to pay) of bidders. The bidders may act strategically, and report untruthfully if this to their benefit. One way to circumvent this is to require that it is in each agent's best interest to report its value truthfully. The goal then 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 [17]. 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. Our focus is on designing auctions that satisfy *dominant-strategy incentive compatibility (DSIC)*, which is a robust and desirable notion of incentive alignment. While there have been some elegant partial characterization results [15, 20, 5, 9], and an impressive sequence of algorithmic results, e.g. [1, 2], these apply to the weaker notion of *Bayesian incentive compatibility (BIC)* except for the setting with one bidder, when DSIC and BIC coincide.

In [6], we have introduced the first, general purpose, end-to-end approach for solving the multi-item auction design problem. In particular, we use multi-layer neural networks to encode auction mechanisms, with bidder valuations forming the input and allocation and payment decisions forming the output. The networks are trained using samples from the value distributions, so as to maximize expected revenue subject to constraints for incentive compatibility. Earlier work has suggested to use algorithms to automate the design of mechanisms [3], but where scalable this earlier work had to restrict the search space to auction designs that are known to be incentive compatible [12, 22]. The deep learning approach, in contrast, enables searching over broad classes of not necessarily truthful mechanisms. Other related work has leveraged machine learning to optimize different aspects of mechanisms [7, 18], but none of these offers the generality and flexibility of our approach.

An extended abstract was published in the Proceedings of the 36th International Conference on Machine Learning, 2019. A full version of this paper is available at https://arxiv.org/abs/1706.03459. All code is available through the GitHub repository at https://github.com/saisrivatsan/deep-opt-auctions.

*Part of the work of the author was completed while visiting Google Research.

†The author is supported in part through NSF award CCF-1841550 and Google Fellowship.

Our framework provides two different approaches to handling IC constraints. In the first, we leverage results from economic theory that characterize DSIC mechanisms, and model the network architecture appropriately. We specifically show how to exploit Rochet's characterization result for single-bidder multi-item settings [21], which states that DSIC mechanisms induce Lipschitz, non-decreasing, and convex utility functions. In the second, we lift the IC constraints into the objective via the *augmented Langrangian* method, which has the effect of introducing a penalty term for IC violations. This approach, which we refer to as *RegretNet*, 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.

In this Research Highlight we describe the general approach, and present a selection of experimental results in support of our general finding that these 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. In the full version of the paper, we also prove *generalization bounds* that provide confidence intervals on the expected revenue and expected violation of IC based on empirical properties obtained 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. Others have provided generalization bounds for training revenue-maximizing auctions in simpler settings; see, e.g. [16].

Follow-up work has extended our approach to handle *budget-constrained bidders* [8] as well as to a problem in social choice, the so-called *facility location problem* [11], and obtained specialized architectures for single bidder settings that satisfy IC [23].

2. OPTIMAL AUCTION DESIGN

We start by stating the optimal auction design problem, and providing a few illustrative examples.

In the general version of the problem, we are given 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 our full version.

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.

Example 1 (Vickrey auction [25]). *A classic result in auction theory concerns the sale of a single item to n bidders. It states that the following auction—the so-called Vickrey or second-price auction—is DSIC and maximizes social welfare: Collect a bid b_i from each bidder, assign the item to the bidder with the highest bid (breaking ties in an arbitrary but fixed manner), and make the bidder pay the second highest bid.*

Example 2 (Myerson auction [17]). *A simple example shows that the Vickrey auction does not maximize revenue: Suppose there are two bidders with $v_i \in U[0, 1]$, then its expected revenue is $1/3$. Higher revenue can be achieved with a second-price auction with reserve r: As before collect bids b_i, allocate to the highest bid but only if this bid is at least r, make the winning bidder (if any) pay the maximum of the runner-up bid and r. It's straightforward to verify that this auction is DSIC and that choosing $r = 1/2$ leads to an expected revenue of $5/12 > 1/3$.*

In the simple example with a single item and uniform valuations, a second-price auction with reserve $1/2$ is in fact the optimal auction. This auction illustrates a special case of Myerson's theory for the design of revenue-optimal, single item auctions [17]. Comparable results are not available for selling multiple items, even when we are trying to sell them to a single bidder!

3. THE LEARNING PROBLEM

At the core of our approach is the following reinterpretation of the optimal auction design problem 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.

More concretely, the problem we seek to solve is the following: 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. Our 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 consider two distinct approaches for achieving IC. In the first approach, we make use of characterization results. When it is possible to encode them within a neural network architecture, these characterizations from economic theory usefully constrain the search space and provide exact DSIC. At the same time, the particular characterization that we use is limited in that it applies only to single-bidder settings. The second approach that we take is more general, applying to multi-bidder settings, and does not rely on the availability of suitable characterization results. On the other hand, this approach entails search through a larger parametric space, and only achieves approximate IC.

We describe the first approach in Section 4, and return to the second approach in Second 5.

4. THE ROCHETNET FRAMEWORK

We have developed two different frameworks that achieve exact IC through appropriate constraints on the neural network architecture. One framework that we have studied, referred to as *MyersonNet*, is inspired by Myerson's lemma [17], and can be used for the study of multi-bidder, single-item auctions (see full version). Another framework, which we refer to as *RochetNet* and which we describe in more detail next, architects a network for single-bidder multi-item settings, and all mechanisms that can be expressed by the network satisfy the respective necessary and sufficient conditions for DSIC provided by Rochet [21]. We give the construction for additive preferences, but this can easily be extended to unit demand valuations.

4.1 Characterization

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 given by:

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

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

THEOREM 4.1 (ROCHET [21]). *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. $u(v) \geq 0, \forall v \in \mathbb{R}^m_{\geq 0}$.

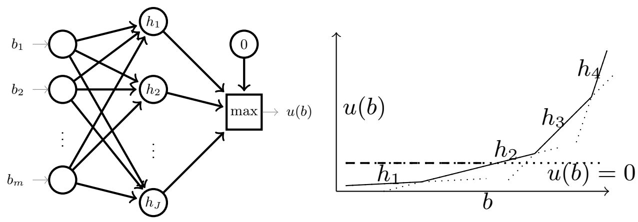
To find the optimal mechanism, it thus suffices to search over all non-negative utility functions that satisfy the conditions in Theorem 4.1, and pick the one that maximizes expected revenue.

4.2 The RochetNet Architecture

To model a non-negative, monotone, convex, Lipschitz utility function, we use the maximum of J linear functions with non-negative coefficients and zero:

$$u^{\alpha,\beta}(v) \;=\; \max\left\{ \max_{j\in[J]} \{\alpha_j \cdot v \,+\, \beta_j\},\, 0 \right\}, \tag{2}$$

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 4.1).



(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.

THEOREM 4.2. *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:

$$g^w(b) = \nabla u^{\alpha,\beta}(b), \tag{3}$$

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

$$p^w(b) \;=\; \nabla u^{\alpha,\beta}(b) \cdot b \;-\; u^{\alpha,\beta}(b). \tag{4}$$

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

$$-\mathbf{E}_{v\sim F}\left[\nabla u^{\alpha,\beta}(v) \cdot v - u^{\alpha,\beta}(v)\right] = \mathbf{E}_{v\sim F}\left[\beta_{j^*(v)}\right]. \tag{5}$$

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:

$$\mathcal{L}(\alpha, \beta) \;=\; -\mathbf{E}_{v\sim F}\Big[\sum_{j\in[J]} \beta_j \widetilde{\nabla}_j(v)\Big], \tag{6}$$

where

$$\widetilde{\nabla}_j(v) \;=\; \mathrm{softmax}_j\big(\kappa \cdot (\alpha_1 \cdot v + \beta_1), \ldots, \kappa \cdot (\alpha_J \cdot v + \beta_J)\big) \tag{7}$$

and $\kappa > 0$ is a constant that controls the quality of the approximation. The softmax function, $\mathrm{softmax}_j(\kappa x_1, \ldots, \kappa x_J) = e^{\kappa x_j} / \sum_{j'} e^{\kappa x_{j'}}$, takes as input J real numbers and returns a probability distribution consisting of J probabilities, proportional to the exponential of the inputs.

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.

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.

4.3 Training

We train RochetNet using samples drawn from simulated bidder value distributions. The approach used for training is the standard projected stochastic gradient descent (SGD) to minimize the loss function $\mathcal{L}(\alpha, \beta)$ in Equation (6). We estimate gradients for the loss using 2^{15} valuation profiles sampled in an online manner. When training for the additive valuations setting, we additionally project each weight α_{jk} onto $[0,1]$ to guarantee feasibility.

5. THE REGRETNET FRAMEWORK

We next describe our second approach to handling IC constraints and the corresponding framework, which we refer to as *RegretNet*. Unlike the first approach, this second approach does not rely on characterizations of IC mechanisms. Instead, we replace the IC constraints with a differentiable approximation and lift 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. Here, we provide an overview of the special case in which bidders have additive values for items, but the framework also handles more general settings.

5.1 Expected ex post regret

We can measure the extent to which an auction violates incentive compatibility through a particular variation on *ex post* regret introduced in [7]. 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:

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

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

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

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

$$\widehat{rgt}_i(w) = \frac{1}{L}\sum_{\ell=1}^{L}\Big[\max_{v'_i \in V_i} u_i^w\big(v_i^{(\ell)}; (v'_i, v_{-i}^{(\ell)})\big) - u_i^w\big(v_i^{(\ell)}; v^{(\ell)}\big)\Big], \quad (8)$$

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

$$\min_{w \in \mathbb{R}^d} \quad -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i=1}^{n} p_i^w(v^{(\ell)})$$
$$\text{s.t.} \quad \widehat{rgt}_i(w) = 0, \quad \forall i \in N. \quad (9)$$

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.

5.2 The RegretNet Architecture

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.

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, 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.

5.3 Training

For RegretNet we have used the *augmented Lagrangian method* to solve the constrained training problem in (9) over the space of neural network parameters w.

We first define the Lagrangian function for the optimization problem, augmented with a quadratic penalty term for

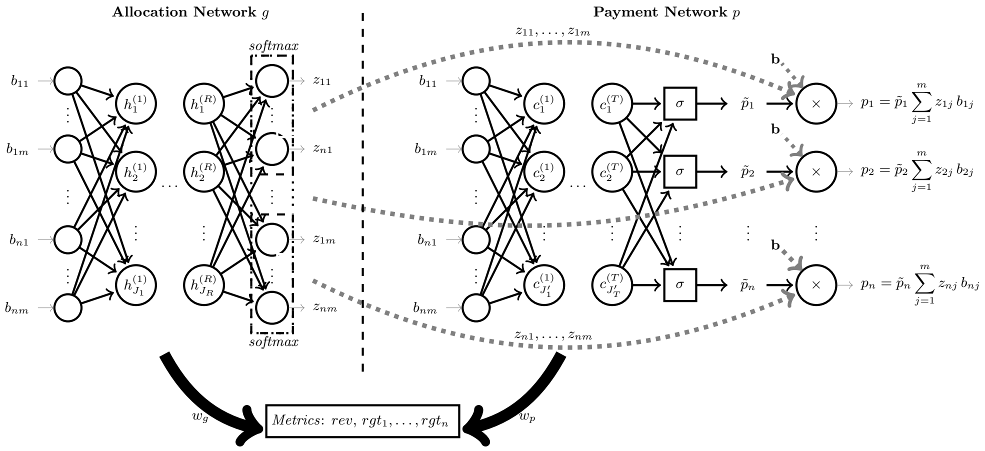


Figure 2: RegretNet: The allocation and payment networks for a setting with n additive bidders and m items. The inputs are bids from each bidder for each item. The revenue rev and expected ex post rgt_i are defined as a function of the parameters of the allocation and payment networks $w = (w_g, w_p)$.

Algorithm 1 RegretNet Training

1: **Input:** Minibatches $\mathcal{S}_1, \ldots, \mathcal{S}_T$ of size B
2: **Parameters:** $\forall t, \rho_t > 0$, $\gamma > 0$, $\eta > 0$, $\Gamma \in \mathbb{N}$, $K \in \mathbb{N}$
3: **Initialize:** $w^0 \in \mathbb{R}^d$, $\lambda^0 \in \mathbb{R}^n$
4: **for** $t = 0$ **to** T **do**
5: Receive minibatch $\mathcal{S}_t = \{v^{(1)}, \ldots, v^{(B)}\}$
6: Initialize misreports $v'^{(\ell)}_i \in V_i, \forall \ell \in [B],\ i \in N$
7: **for** $r = 0$ **to** Γ **do**
8: $\forall \ell \in [B],\ i \in N:$
9: $v'^{(\ell)}_i \leftarrow v'^{(\ell)}_i + \gamma \nabla_{v'_i} u^w_i\big(v^{(\ell)}_i; (v'^{(\ell)}_i, v^{(\ell)}_{-i})\big)$
10: **end for**
11: Compute regret gradient: $\forall \ell \in [B], i \in N$:
12: $g^t_{\ell,i} =$
13: $\nabla_w \left[u^w_i\big(v^{(\ell)}_i; (v'^{(\ell)}_i, v^{(\ell)}_{-i})\big) - u^w_i\big(v^{(\ell)}_i; v^{(\ell)}\big)\right]\Big|_{w=w^t}$
14: Compute Lagrangian gradient (10) on $\mathcal{S}_t$ and update:
15: $w^{t+1} \leftarrow w^t - \eta \nabla_w \mathcal{C}_{\rho_t}(w^t, \lambda^t)$
16: Update Lagrange multipliers once in Q iterations:
17: **if** t is a multiple of Q
18: Compute $\widehat{rgt}$ on $\mathcal{S}_t$
19: $\lambda^{t+1}_i \leftarrow \lambda^t_i + \rho_t\, \widehat{rgt}_i(w^{t+1}), \quad \forall i \in N$
20: **else**
21: $\lambda^{t+1} \leftarrow \lambda^t$
22: **end for**

violating the constraints:

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

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

The solver is described in Algorithm 1. We divide the training sample $\mathcal{S}$ into minibatches of size B, estimate gradients on the minibatches, and perform several passes over the training samples. The update (a) on model parameters involves an unconstrained optimization of $\mathcal{C}_\rho$ over w and is performed using a gradient-based optimizer. The gradient of $\mathcal{C}_\rho$ w.r.t. w for fixed λ^t is given by:

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

where

$$g_{\ell,i} = \nabla_w\left[\max_{v'_i \in V_i} u^w_i\big(v^{(\ell)}_i; (v'_i, v^{(\ell)}_{-i})\big) - u^w_i\big(v^{(\ell)}_i; v^{(\ell)}\big)\right].$$

The terms $\widehat{rgt}_i$ 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 (lines 6–10).

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

6. EXPERIMENTS

We present and discuss a selection of experiments out of a broad range of experiments that we have conducted and that we describe in more detail in [6] and the full version. The experiments 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.

6.1 Setup

We implemented our framework using the TensorFlow deep learning library.

For RochetNet we initialized parameters α and β in Equation (2) 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 [10]. 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 6.6.

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 (7) 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.

6.2 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 (8)). 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.

6.3 The Manelli-Vincent Auction

As a representative example of the exhaustive set of analytical results that we can recover with our approach we discuss the Manelli-Vincent auction [15].

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

The optimal auction for this setting is given by Manelli and Vincent [15]. We used two hidden layers with 100 hidden nodes in RegretNet for this setting. A visualization of the optimal allocation rule and those learned by RochetNet and RegretNet is given in Figure 3. Figure 4(a) gives the optimal revenue, the revenue and regret obtained by RegretNet, and the revenue obtained by RochetNet. Figure 4(b) shows how these terms evolve over time during training in RegretNet.

Both approaches essentially recover the optimal design, not only in terms of revenue, but also in terms of the allocation rule and transfers. The auction learned by RochetNet is exactly DSIC and matches 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.

6.4 The Straight-Jacket Auction

Extending the analytical result of [15] 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

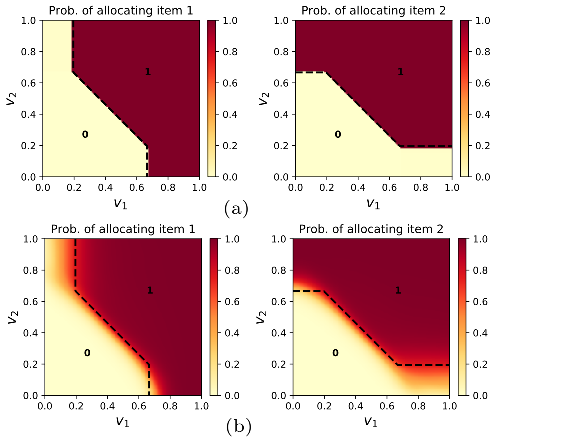
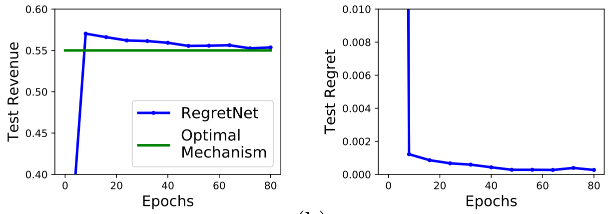


Figure 3: Side-by-side comparison of allocation rules learned by RochetNet (panels (a)) and RegretNet (panels (b)) for for Setting A. The panels describe the probability that the bidder is allocated item 1 (left) and item 2 (right) for different valuation inputs. The optimal auctions are described by the regions separated by the dashed black lines, with the numbers in black the optimal probability of allocation in the region.

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

(a)



(b)

Figure 4: (a): Test revenue and regret for RegretNet and revenue for RochetNet for Settings A. (b): Plot of test revenue and regret as a function of training epochs for Setting A with RegretNet.

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 5: Revenue of the Straight-Jacket Auction (SJA) computed via the recursive formula in [9], 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.

even clear whether the optimal mechanism is deterministic or requires randomization.

Giannakopoulos and Koutsoupias [9] proposed a Straight-Jacket Auction (SJA) and 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. These authors also conjecture that the SJA remains optimal for general m, but were unable to prove it.

Figure 5 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 [9] for all $m \leq 10$. We take this as strong additional evidence that the conjecture of Giannakopoulos and Koutsoupias [9] is correct.

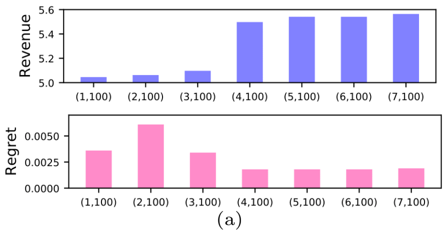
6.5 Discovering New Optimal Designs

RochetNet can also be used to discover new, optimal designs. For this, we consider a single bidder with additive but correlated valuations for two items as follows:

B. 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 (e.g., $0.5, 1, 3, 5$) to discover the optimal auction. 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 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. We validate the optimality of this auction through duality theory [4] in Theorem 6.1.

THEOREM 6.1. *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.*



(a)

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

(b)

Figure 6: (a) Revenue and regret of RegretNet on the validation set for auctions learned for Setting C using different architectures, where (R, K) denotes R hidden layers and K nodes per layer. (b) Test revenue and regret for Settings C and D, for the (5, 100) architecture.

6.6 Scaling Up

We have also considered 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.

The following two settings generalize the basic setting considered in [15] and [9] to more than one bidder:

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

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

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 [19]. 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 C, we show in Figure 6(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 C and Setting D. Figure 6(b) shows that the learned auctions yield higher revenue compared to the baselines, and do so with tiny regret.

7. CONCLUSION

The results from this research demonstrate 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, 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 this framework in advancing economic theory, for example in supporting or refuting conjectures and as an assistant in guiding new economic discovery.

More generally, we believe that our work (together with a handful of contemporary works such as [13, 24]) just opened the door to ML-assisted economic theory and practice, and we are looking forward to the advances that this agenda will bring along.

8. REFERENCES

[1] 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.

[2] 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, 2012.

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

[4] 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.

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

[6] P. Duetting, Z. Feng, H. Narasimhan, D. C. Parkes, and S. S. Ravindranath. Optimal auctions through deep learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1706–1715, 2019.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

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

[18] 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.

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

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

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

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

[23] 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.

[24] 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.

[25] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16:8–37, 1961.