Machine Learning for Optimal Economic Design

Paul Dütting, Zhe Feng, Noah Golowich, Harikrishna Narasimhan, David C. Parkes and Sai Srivatsa Ravindranath

Abstract This position paper anticipates ways in which the disruptive developments in machine learning over the past few years could be leveraged for a new generation of computational methods that automate the process of designing optimal economic mechanisms.

1 Introduction

Mechanism design is the problem of designing incentives to achieve an outcome that satisfies desired objectives in the presence of self-interested participants. Because the participants are assumed to act rationally, and play an equilibrium, it can also

We would like to thank Yang Cai, Vincent Conitzer, Constantinos Daskalakis, Scott Kominers, Alexander Rush, and participants at seminars at the Simons Institute for the Theory of Computing, Dagstuhl, London School of Economics, IJCAI'18, AAMAS'18, The Technion, the EC'18 WADE workshop, the German Economic Association annual meeting, Google, MIT/Harvard Theory Seminar, WWW'19 workshop, and HBS for their helpful feedback.

P. Dütting
Department of Mathematics, London School of Economics, Houghton Street,
London WC2A 2AE, UK
e-mail: p.d.duetting@lse.ac.uk

Z. Feng · N. Golowich · H. Narasimhan · D. C. Parkes (✉) · S. S. Ravindranath
Paulson School of Engineering and Applied Sciences, Harvard University,
33 Oxford Street, Cambridge, MA 02138, USA
e-mail: parkes@eecs.harvard.edu

Z. Feng
e-mail: zhe_feng@g.harvard.edu

N. Golowich
e-mail: ngolowich@college.harvard.edu

H. Narasimhan
e-mail: hnarasimhan@g.harvard.edu

S. S. Ravindranath
e-mail: saisr@g.harvard.edu

J.-F. Laslier et al. (eds.), *The Future of Economic Design*, Studies in Economic Design, https://doi.org/10.1007/978-3-030-18050-8_70

be thought about as the problem of inverse game theory. A mechanism designer creates the rules of a game, by which an outcome will be selected based on messages sent by participants. Mechanism design has developed into a beautiful theory that has influenced thinking across a range of problems, including auctions and voting procedures, but despite more than 40 years of intense research, several fundamental questions remain open.

Revenue-optimal auction design is the prime example, both for having elegant theoretical results and also for seemingly simple and important-to-practice cases that remain unsolved. The central result is the characterization of revenue-optimal single-item auctions as *virtual value maximizers* (Myerson 1981). We know, for example, that second price auctions with a suitably chosen reserve price are optimal when selling to bidders with i.i.d. values, and how to prioritize one bidder over another in settings with bidder asymmetry. But Myerson's theory is as beautiful as it is rare. Indeed, the design of optimal auctions for multiple items is much more difficult, and has defied a thorough theoretical understanding.

The contours of the available analytical results bear witness to the severe analytical challenges in going beyond single-item auctions. Even the design of the optimal auction for selling two items to just a single buyer is not fully understood.[1] For a single additive buyer with values on items i.i.d. $U(0, 1)$, Manelli and Vincent (2006) handle three items, and Giannakopoulos and Koutsoupias (2014) up to six items. Yao (2017) provides the optimal design for any number of additive bidders and two items, buy only as long as item values can take on one of two possible values. Decades after Myerson's result, we do not have a precise description of optimal auctions with two or more bidders and more than two items.

A promising alternative is to use computers to solve problems of optimal economic design. The framework of *automated mechanism design* (Conitzer and Sandholm 2002, 2003) suggests to use algorithms for the design of optimal mechanisms. Early approaches required an explicit representation of all possible type profiles, which is exponential in the number of agents and does not scale (see also Albert et al. 2017). Others have proposed more restricted approaches, that search through a parametric family of mechanisms (Guo and Conitzer 2009, 2010; Sandholm and Likhodedov 2015; Narasimhan et al. 2016).

In recent years, efficient algorithms have been developed for the design of optimal, *Bayesian incentive compatible (BIC)* auctions in multi-bidder, multi-item settings (Cai and Daskalakis 2015; Alaei et al. 2012, 2013; Cai et al. 2012a, b, 2013a; Bhalgat et al. 2013; Cai and Huang 2013; Daskalakis et al. 2017). But while there is a characterization of optimal mechanisms as virtual-value maximizers (Cai et al.

[1] Results are known for additive i.i.d. $U(0, 1)$ values on items (Manelli and Vincent 2006), additive, independent and asymmetric distributions on item values (Daskalakis et al. 2017; Giannakopoulos and Koutsoupias 2015; Thirumulanathan et al. 2016), additive, i.i.d. exponentially distributed item values (Daskalakis et al. 2017) and extended to multiple items (Giannakopoulos 2015), additive, i.i.d. Pareto distributions on item values (Hart and Nisan 2012), unit-demand valuations with item values i.i.d. $U(c, c+1)$, $c > 0$ (Pavlov 2011), and unit-demand, independent, uniform and asymmetric distributions on item values (Thirumulanathan et al. 2017).

2012a, 2013b), relatively little is known about the structure of optimal mechanisms; see Daskalakis (2015) for an overview.

Moreover, these algorithms leverage a reduced-form representation that makes them unsuitable for the design of *dominant-strategy incentive compatible* (DSIC) mechanisms, and similar progress has not been made for this setting. DSIC is of special interest because of the robustness it provides, relative to BIC. The recent literature has focused instead on understanding when simple mechanisms can approximate the performance of optimal designs.[2]

Where do we go from here? Thanks to the disruptive developments in machine learning, we believe that there is a powerful opportunity to use its tools for the design of optimal economic mechanisms. The essential idea is to repurpose the training problem from machine learning for the purpose of optimal design. In what follows, we will highlight some recent results that we have in support of this agenda. The question we ask is:

> *Can machine learning be used to design optimal economic mechanisms, including optimal DSIC mechanisms, and without the need to leverage characterization results?*

The illustrative examples will be drawn from optimal auction design, including optimal design with private budget constraints, as well as a problem in social choice—the multi-facility location problem. We believe the framework is considerably more general, and will extend to address problems in matching and non-linear pricing, for example.

2 Adopting the Lens of Machine Learning

To understand the opportunity, we start with optimization-based formulations for each of the problems of mechanism design and machine learning.

A typical problem in mechanism design is to find a function from inputs (a type profile) to outputs (say an allocation and payments) that maximizes the expected value of an objective, defined for a distribution on inputs. Global constraints are also imposed, for example *incentive compatibility* (IC).[3] Illustrating this for the design of an *allocation rule* g and *payment rule* p (mapping reported types to an allocation and payments, respectively) of an auction, we would solve:

[2] Working in increasingly general settings, relevant results on DSIC auction design include Chawla et al. (2007, 2010), Alaei (2014), Kleinberg and Weinberg (2012), Hart and Nisan (2012), Li and Yao (2013), Babaioff et al. (2014), Yao (2015), Rubinstein and Weinberg (2015), Cai et al. (2016), Cai and Zhao (2017), Dütting et al. (2017). These mechanisms are simple, and reveal the structural ingredients that are important for the design of mechanisms with good revenue properties.

[3] IC means that no agent can benefit, in equilibrium, by misreporting its type, and can hold in a dominant-strategy equilibrium (DSIC) or a Bayes-Nash equilibrium (BIC). We will generally be interested in DSIC.

$$\max_{g,p} \mathbf{E}_{v\sim F_V} \; \mathcal{O}(v; g, p) \quad (1)$$
$$\text{s.t.} \quad (g, p) \in IC.$$

This maximizes the expected value of objective $\mathcal{O}(v; g, p)$, where $v=(v_1, \ldots, v_n)$ denotes the type profile for n bidders, and F_V the distribution from which type profiles are sampled. For revenue optimality, the objective would be $\mathcal{O}(v; g, p) = \sum_{i=1}^{n} p_i(v)$. Here, IC denotes the set of IC rules. Following Myerson (1981), this kind of problem can be solved in simple cases through a characterization of allocation rules for which there exists a payment rule that provides IC, allowing the objective to be expressed in terms of the allocation rule alone, and then proceeding analytically. But this approach is very challenging to extend to general, multi-item problems.

A typical problem in machine learning is to find a function from inputs (a vector of features) to outputs (say an image label) that minimizes the expected value of an objective. A typical objective is to minimize the expected loss on input-output pairs sampled from some distribution, where the loss might be defined to be 0 if the predicted output is correct and 1 if it is incorrect. For parametric models, with function f^w defined through parameters $w \in \mathbb{R}^d$ (for some $d \geq 1$), we would solve

$$\min_{w} \mathbf{E}_{(x,y)\sim F_{XY}} \; \mathcal{L}(x, y; f^w). \quad (2)$$

The objective is to minimize expected loss, for *loss function* $\mathcal{L}(x, y; f^w)$, where (x, y) is an input-output pair sampled i.i.d. from some distribution F_{XY}. The input-output pair could be *feature vector* $x \in \mathbb{R}^k$ for $k \geq 1$, and *target value* $y \in \mathbb{R}$, respectively. Here, $f^w : \mathbb{R}^k \mapsto \mathbb{R}$ is a parameterized function (the target can also be categorical, in which case f^w would map to a finite set). A typical approach to solve (2) is to use training data sampled from F_{XY}, together with an optimization method such as stochastic gradient descent to minimize the loss on the training data (perhaps along with regularization, to prefer simple solutions over complex solutions that might over-fit to the training data).

Comparing formulations (1) and (2), and considering the particular setting of revenue-optimal auction design, this suggests the following representation of a problem of optimal economic design as one of machine learning:

Feature vector	$x \longrightarrow (v_1, \ldots, v_n)$
Target value	not needed
Hypothesis	$f^w \longrightarrow (g^w, p^w)$ (parameterized allocation rule and payment rule)
Loss function	$\mathcal{L}(v; g^w, p^w) = -\sum_{i=1}^{n} p_i^w(v)$
Constraints (new)	IC

The loss function becomes the negated revenue, and thus minimizing expected loss is equivalent to maximizing expected revenue. There is no need for labeled training data: rather, the required training data is samples of type profiles, and the loss function is defined to directly capture the economic objective (e.g., negated revenue). For this reason, there is no object that corresponds to the target value.

The technical challenge, relative to standard training problems in machine learning, is to formulate the IC constraint. In some settings, IC can be directly achieved by constraining the set of functions (the hypothesis class). In other settings, we have found it useful to work with quantities that capture the degree of violation of the constraint. Fixing the bids of others, the *ex post regret* to a bidder is the maximum increase in the bidder's utility, considering all possible non-truthful bids. The *expected ex post regret* for bidder i, given mechanism parameters w, is defined as

$$rgt_i(w) \;=\; \mathbf{E}_{v \sim F_V}\Big[\max_{v_i' \in V_i} u_i(v_i', v_{-i}; v_i, g^w, p^w) - u_i(v_i, v_{-i}; v_i, g^w, p^w)\Big], \qquad (3)$$

where V_i is the valuation domain for bidder i, and $u_i(v_i', v_{-i}; v_i, g^w, p^w)$ is the utility (value minus price) to bidder i with valuation v_i when reporting v_i', when others report $v_{-i} = (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$, and with allocation and payment rule g^w and p^w, respectively.

For a suitably expressive, parameterized set of functions g^w and p^w, the problem of optimal auction design can be formulated as:

$$\min_w \; \mathbf{E}_{v \sim F_V}\, \mathcal{L}(v; g^w, p^w) \qquad (4)$$
$$\text{s.t. } rgt_i(w) \;=\; 0, \quad \forall i \in N.$$

This allows for ex post regret only on measure zero events. We will additionally require *individual rationality*, a property that every agent has a weak incentive to participate in a mechanism. This can be ensured by restricting our search space to a class of parametrized mechanisms (g^w, p^w) that charge no agent more than its expected utility for an allocation.

Let us suppose this can be made to work— that machine learning can be used in this way for optimal economic design. Before continuing, we will discuss some objections that could be raised about this research agenda:

(1) "As theorists, we care about understanding the structure of optimal designs, we're not interested in black-box solutions." In fact, we expect that a machine learning framework can provide a useful complement to theory, used for example to support or refute conjectures on the structure of optimal designs, or to identify parts of the theory landscape where current designs are far from optimal. Asking that learned designs are interpretable is also an interesting research agenda in its own right, and one that should find synergy with a growing attention to interpretability in machine learning (Doshi-Velez et al. 2015; Wang and Rudin 2015; Caruana et al. 2015; Ribeiro et al. 2016; Smilkov et al. 2016; Raghu et al. 2016; Andrew Slavin Ross 2017).

(2) "Simplicity is important. Participants need to understand mechanisms." While this is undoubtedly important in some settings, we believe that participants in many kinds of economic mechanisms will be increasingly automated (consider, for example the use of automated bidding for advertising and other problems in marketing, and automated trading in finance). Mechanisms populated by automated agents do not need to be simple in the same way as those intended for use by people. Rather, it

seems to use that robust game-theoretic properties such as DSIC are more important than descriptive simplicity, and especially if a mechanism is accompanied by a proof of its economic properties.[4]

(3) "What if incentive compatibility is only approximately achieved? What good is this from an equilibrium perspective?" We have some sympathy for this concern, in that when the expected, *ex post* regret of a learned mechanism is small but positive, some types may still have a large incentive to deviate. But this is only a first step. Going forward, we can think about other notions of approximate DSIC.[5] Moreover, this concern can be tempered by also imposing additional structural properties that are necessary for IC, thus tightening the approximation.[6]

(4) "What if there are computation-theoretic or learning-theoretic barriers to optimal design?" Any such barrier is intrinsic, and holds whether the design problem is left to human ingenuity or formulated in a way that is amenable to solution by an algorithm. Barriers, where they exist, will require the design of second-best mechanisms, that are optimal given not only incentive constraints but also these computational or learning-theoretic constraints.[7] As such, we see this not as an objection to using machine learning for optimal economic design, but as a broader objection to the agenda of optimal economic design.

(5) "What if it is the rules of the optimal mechanism entail solving an intractable computational problem?" We think this presents the most serious complaint, in that we already know of settings such as those of combinatorial auctions where the allocation rule requires solving an **NP**-hard optimization problem (Rothkopf et al. 1998). Still, because we may be interested in solving problems of a fixed size (in terms of the number of items and bidders), these kinds of complexity barriers do not immediately bite. Moreover, many complexity barriers are worst-case, and there is an increasing attention to using neural networks to solve problems of combinatorial optimization for distributions on inputs (Niepert et al. 2016; Vinyals et al. 2015; Orhan and Ma 2017), and progress there will also benefit the use of machine learning for automated economic design.

(6) "What if the learned design is brittle, with its incentives or optimality properties not robust to a small change in the type distribution?" On one hand, DSIC designs are intrinsically more robust than BIC designs in that incentive compatibility does not depend on the distribution. On the other hand, empirical observations

[4]See Parkes and Wellman (2015) for a discussion on the role of AI in the mediation of economic transactions.

[5]See Carroll (2013), Mennle and Seuken (2014), Lubin and Parkes (2012), Mennle and Seuken (2017) for some discussion of approximate notions of incentive compatibility.

[6]For example, we could also penalize failure of weak-monotonicity (Bikhchandani et al. 2006), or insist that the implied pricing-function is agent independent (with prices to an agent that are do not depend on its report, conditioned on an allocation).

[7]Daskalakis et al. (2014) give a complexity result for optimal mechanism design. There is also a recent literature on the sample complexity of auctions and mechanisms, including revenue-optimal auctions (Elkind 2007; Cole and Roughgarden 2014; Dughmi et al. 2014; Morgenstern and Roughgarden 2015, 2016; Huang et al. 2015; Devanur et al. 2016; Narasimhan and Parkes 2016; Gonczarowski and Nisan 2017; Cai and Daskalakis 2017).

about the use of highly non-linear models in other settings, such as those of *deep learning* (Goodfellow et al. 2016), suggest that robustness to small perturbations in the inputs can indeed be a concern (Szegedy et al. 2014; Fawzi et al. 2018; Moosavi-Dezfooli et al. 2016). The robustness of learned models is gaining attention within machine learning (Chen et al. 2017; Shalev-Shwartz and Wexler 2016; Goodfellow et al. 2015; Abadi et al. 2016), and progress there will also bring benefits here. At the same time, it will be important to conduct thorough studies of learned mechanisms to validate their robustness.

3 Deep Learning for Optimal Auction Design

We have initiated the study of multi-layer, feed-forward neural networks for the design of optimal auctions (Dütting et al. 2019).[8] These networks provide differentiable, non-linear function approximations to auction rules, and the training problem—the problem of optimal design—is solved through stochastic gradient descent.[9]

We focus here on describing a "fully agnostic" approach, which proceeds without the use of characterization results and, because of this, holds the most promise in discovering new economic designs.[10] The input layer of the REGRETNET architecture represents bids, and the network has two logically distinct components: the allocation network and the payment network (see Fig. 1). The networks consist of multiple "hidden layers" (denoted $h^{(r)}$ and $c^{(t)}$ in the figure) and an output layer. Each unit in a hidden layer and each unit in an output layer may be a non-linear activation function, applied to a weighted sum of outputs from the previous layer. These weights form the parameters of the network.

The allocation rule g is modeled with R fully-connected hidden layers (we have used $R = 2$ and 100 units in each layer in our experiments), each with *tanh activations*, and a fully-connected output layer. For a given bid profile b, illustrated here as providing a number for each bidder for each of m items, the network outputs a vector of allocation probabilities $z_{1j}(b), \ldots, z_{nj}(b)$, for each item $j \in [m]$, through a

---

[8]The use of machine learning for mechanism design was earlier pioneered by Dütting et al. (2015), who use support vector machines to design payment rules for a given allocation rule (which can be designed to be scalable). But their framework can fail to even closely approximate incentive compatibility then the rule is not implementable, and does not support design objectives that are stated on payments. Earlier, Procaccia et al. (2009) studied the learnability of specific classes of voting rules, but without considering incentives; see also Xia (2013), who suggests a learning framework that incorporates specific axiomatic properties.

[9]Deep learning, which refers typically to the use of multi-layer neural networks, has gained a great deal of attention in recent years. This is because of the existence of large data sets, the development of tool chains that make experimentation easy, optimized hardware to speed-up training (GPUs), as well as massive investment from the private sector. Whether a network is considered 'deep' or not is a matter of taste.

[10]We have also explored network architectures that leverage characterization results; Myerson (1981) and Rochet (1987) for optimal auction design, and Moulin (1980) for facility location problems.

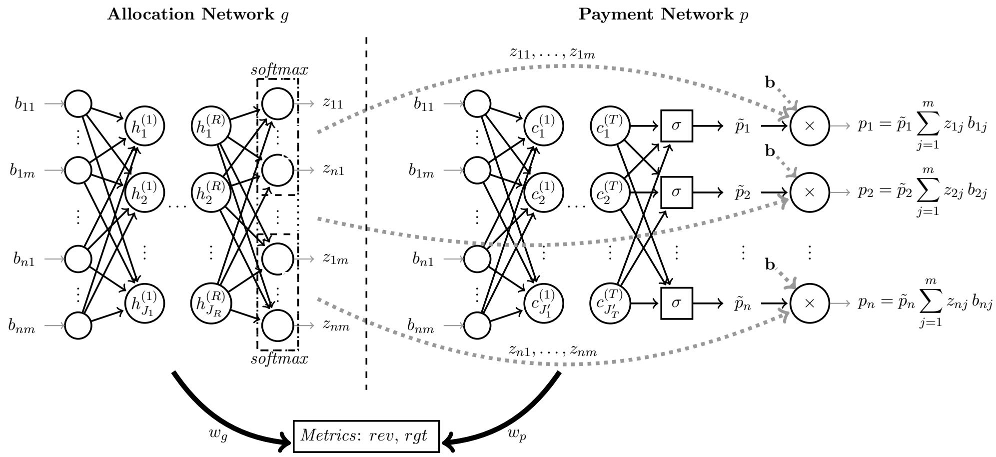


Fig. 1 RegretNet: The allocation network g and payment network p for a setting with multiple bidders $(1, \ldots, n)$ and multiple items $(1, \ldots, m)$ (Dütting et al. 2019). The rev and rgt are defined as a function of the parameters of the allocation and payment networks $w = (w_g, w_p)$

softmax activation function, with $\sum_{i=1}^{n} z_{ij}(b) \leq 1$ for each item $j \in [m]$.[11] Bundling of items is possible because the value on output units corresponding to allocating each of two different items to the same bidder can be correlated.

The payment rule is modeled using a feed-forward neural network with T fully-connected hidden layers (we use $T = 2$ and 100 units in each layer in our experiments), each with tanh activations, and a fully-connected output layer. The output layer defines the payment for each bidder i given a type profile. To ensure that the auction satisfies individual rationality (IR), i.e. does not charge a bidder more than its expected value for the allocation, the network first computes a fractional payment $\tilde{p}_i \in [0, 1]$ for each bidder i using a sigmoid unit, and outputs a payment $p_i = \tilde{p}_i \sum_{j=1}^{m} z_{ij}\, b_{ij}$, where z_{ij}'s are the allocation probabilities output by the allocation network.[12]

Altogether, we can adopt $w \in \mathbb{R}^d$ to denote the vector of parameters, including parameters in both the payment and allocation parts of the network.

In practice, the loss and regret involved in formulating (4) are estimated from a sample of value profiles, $S = \{v^{(1)}, \ldots, v^{(L)}\}$, drawn i.i.d. from F_V. In place of expected loss, we adopt the *empirical loss*, defined as

[11] The sigmoid activation function is $\sigma(z) = 1/(1 + e^{-z})$. The softmax activation function for item j is $softmax_i(s_{1j}, \ldots, s_{nj}, s_{n+1,j}) = e^{s_{ij}} / \sum_{k=1}^{n+1} e^{s_{kj}}$, where $s_{n+1,j}$ is a dummy input that corresponds to the item not being allocated to any bidder. In another variation, we handle unit-demand valuations of bidders by using an additional set of softmax activation functions, one per agent, and taking the minimum of these item-wise and agent-wise softmax components in defining the output layer.

[12] The output p_i that corresponds to bidder i is the amount the bidder should pay in expectation, for a particular bid profile. This can be converted into an equivalent lottery on payments, such that a bidder's payment is no greater than her value for any realized allocation (the property of *ex post* IR).

$$\widehat{\mathcal{L}}(g^w, p^w) = -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i=1}^{n} p_i^w(v^{(\ell)}). \tag{5}$$

To estimate the regret, we use additional samples of valuation profiles S_ℓ drawn i.i.d. from F_V for each profile $v^{(\ell)}$ in S, and compute the maximum utility gain over these alternate profiles. The regret penalty is estimated as:

$$\widehat{rgt}_i(w) = \frac{1}{L}\sum_{\ell=1}^{L} \max_{v' \in S_\ell}\left(u_i(v_i', v_{-i}^{(\ell)}; v_i^{(\ell)}, g^w, p^w) - u_i(v^{(\ell)}; v_i^{(\ell)}, g^w, p^w)\right). \tag{6}$$

For large samples S and S_ℓ (for each ℓ), a mechanism with very low empirical regret, will, with high probability, have very low regret.[13]

The training problem becomes:

$$\begin{aligned} &\min_w \ \widehat{\mathcal{L}}(g^w, p^w) \\ &\text{s.t.}\ \ \widehat{rgt}_i(w) = 0, \quad \forall i \in N. \end{aligned} \tag{7}$$

We can optimize (7) via the method of *augmented Lagrangian optimization*. This uses a sequence of unconstrained optimization problems, where the regret constraints are enforced through a weighted term in the objective. The solver works with the Lagrangian function, augmented with a quadratic penalty term for violating the constraints:

$$\mathcal{C}_\rho(w; \lambda_{rgt}) = \widehat{\mathcal{L}}(g^w, p^w) + \sum_{i \in N} \lambda_{rgt,i}\, \widehat{rgt}_i(w) + \frac{\rho}{2}\left(\sum_{i\in N} \widehat{rgt}_i(w)^2\right), \tag{8}$$

where $\lambda_{rgt} \in \mathbb{R}^n$ is a vector of Lagrange multipliers, and $\rho > 0$ is a fixed parameter that controls the weight on the quadratic penalty. The solver operates across multiple iterations, and performs the following updates in each iteration t:

$$w^{t+1} \in \text{argmin}_w\ \mathcal{C}_\rho(w;\ \lambda_{rgt}^t) \tag{9}$$

$$\lambda_{rgt,i}^{t+1} = \lambda_{rgt,i}^t + \rho\, \widehat{rgt}_i(w^{t+1}), \quad \forall i \in N, \tag{10}$$

where the inner optimization in (9) is approximated through multiple iterations of stochastic subgradient descent; in particular, the gradient is pushed through the loss function as well as the empirical measure of regret. The Lagrange multipliers are initialized to zero.[14]

[13] In more recent work (Dütting et al. 2019) we take an adversarial-style approach, using a gradient-based approach for estimating regret for a given profile. The gradient-based approach requires that the valuation space is continuous and the utility function is differentiable, but is more scalable and stable for larger settings.

[14] With a suitably large penalty parameter ρ, the method of augmented Lagrangian is guaranteed to converge to a (locally) optimal solution to the original problem (Wright and Nocedal 1999). In practice we find that even for small values of ρ and enough iterations, the solver converges to auction designs that yield near-optimal revenue while closely satisfying the regret constraints.

3.1 Illustrative Results

Through this approach, almost optimal auctions with almost zero expected *ex post* regret can be obtained across a number of different economic environments.

For the results presented here, we set $\rho = 0.05$ and sample 5000 training and 5000 value profiles i.i.d from a known distribution. We use the *TensorFlow* deep learning library, solving the inner optimization in the augmented Lagrangian method using the ADAM solver (Kingma and Ba 2015) with learning rate 0.001 and mini-batch size 64. All the experiments are run on a cluster of NVDIA GPU cores.

We first present results for the following two item, single-bidder settings, for which there exist theoretical results (this provides an optimal benchmark):

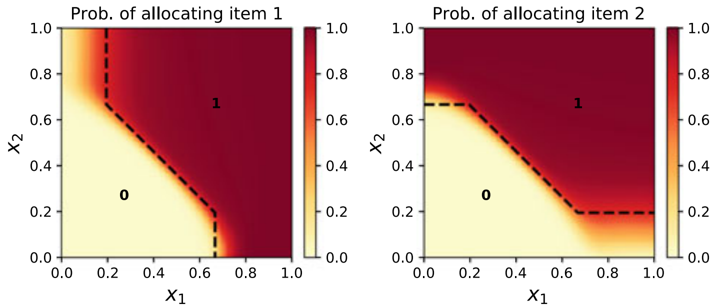
- 2 items, single additive bidder, with item values $x_1, x_2 \sim U[0, 1]$. See Fig. 2a. The optimal DSIC mechanism is due to Manelli and Vincent (2006).
- 2 items, single additive bidder, with item values $x_1 \sim U[4, 16]$ and $x_2 \sim U[4, 7]$. The optimal DSIC mechanism is due to Daskalakis et al. (2017).
- 2 items, single unit-demand bidder, with item values $x_1, x_2 \sim U[2, 3]$. The optimal DSIC mechanism is due to Pavlov (2011).
- 2 items, single unit-demand bidder, with item values $x_1, x_2 \sim U[0, 1]$. The optimal DSIC mechanism is due to Pavlov (2011).

Table 1 summarizes the revenue and regret for the learned mechanisms (all measured on data sampled from F_V and distinct from training data, and with regret normalized to be stated per-agent). The revenue from the learned auctions is very close to the optimal designs from the theoretical literature. In two cases, the revenue from REGRETNET is slightly higher than optimal. This can be explained by the non-zero regret, which makes these auctions not quite DSIC when training was terminated.

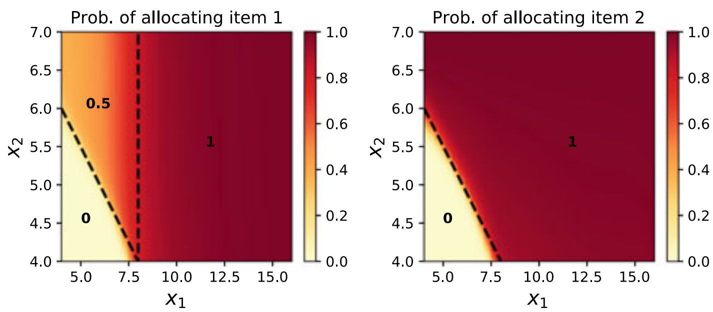
Figure 2a–d provide a visualization of the allocation rules in the learned mechanisms, comparing them with the optimal rules. In each case, we plot the probability of allocating item 1 and item 2 to the bidder in the learned mechanism, as a function of the bidder's value on each item. The design of the optimal allocation rule

Table 1 Revenue and regret for REGRETNET, comparing to the expected revenue of the optimal DSIC auction. We also state the normalized revenue, as a fraction of the revenue from the optimal auction (Dütting et al. 2019)

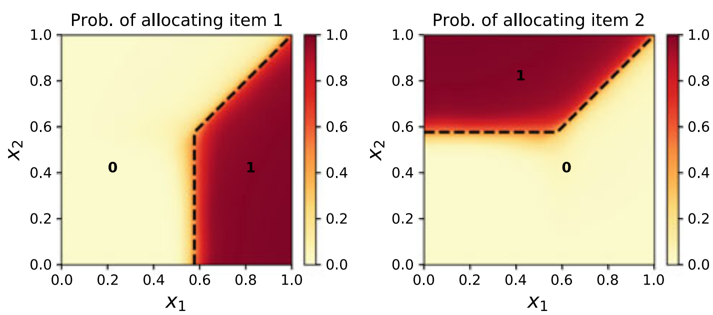
Auction environment	Optimal	REGRETNET	
	rev	rev (norm)	Regret
2 item, 1 additive bidder, $x_1, x_2 \sim U[0, 1]$	0.550	0.557 (101.3%)	<0.001
2 item, 1 additive bidder, $x_1 \sim U[4, 16]$, $x_2 \sim [4, 7]$	9.781	9.722 (99.4%)	<0.004
2 item, 1 unit-demand bidder, $x_1, x_2 \sim U[0, 1]$	0.384	0.386 (100.5%)	<0.001
2 item, 1 unit-demand bidder, uniform, $x_1, x_2 \sim U[2, 3]$	2.137	2.124 (99.4%)	<0.001



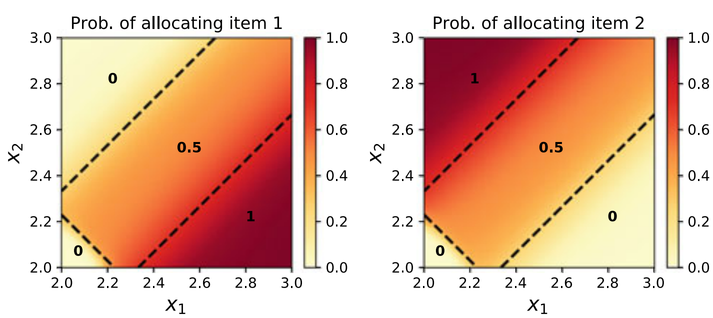
(a) Single bidder, additive, item values $x_1, x_2 \sim U[0, 1]$



(b) Single bidder, additive, item values $x_1 \sim U[4, 16]$, $x_2 \sim [4, 7]$



(c) Single bidder, unit-demand, item values $x_1, x_2 \sim U[0, 1]$



(d) Single bidder, unit-demand, item values $x_1, x_2 \sim U[2, 3]$

Fig. 2 A comparison of the allocation rules learned by REGRETNET to those of optimal auctions (Dütting et al. 2019). These are all single bidder environments. We plot the probability of allocating item 1 and item 2, as a function of the bidder's value on each item. The design of the optimal allocation rule is superimposed, with different allocation regions separated by dashed lines (the number in a region gives the probability the item is allocated in the optimal solution)

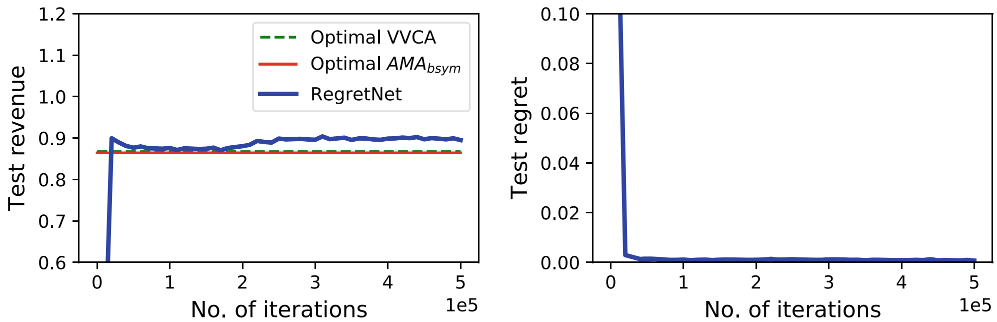


Fig. 3 The test revenue and regret of RegretNet as a function of solver iterations for the two items, two additive bidders setting, where item values are i.i.d. uniform on [0, 1]. (Dütting et al. 2019)

is superimposed, with different allocation regions separated by dashed lines (with the number in a region giving the probability the item is allocated in the optimal solution). We can confirm visually a very close correspondence between the result of machine learning and the optimal design.

We have also used this approach to design auctions for economic environments that are out of reach of theoretical analysis.

These include the single, additive bidder environment with ten items (there is no analytical solution with more than six items), as well as a setting with two items and two additive bidders, where item values are i.i.d. uniform on interval [0, 1]. Figure 3 illustrates the effect of additional training iterations on revenue and regret in this example, where revenue and regret are computed on hold out (test) data.[15] This progress of successively lower regret across training iterations is representative of the experimental results reported in Dütting et al. (2019). In this case, we learn an auction with effectively zero regret, and expected revenue of 0.895 (compared with two state-of-art results from the computational literature, namely 0.867 for the optimal *VVCA auction* and 0.864 for the optimal *symmetric AMA auction* (Sandholm and Likhodedov 2015).

4 Extensions of the Deep Learning Approach to Other Domains

To illustrate the generality of the framework, we briefly describe two different extensions.

First, we are exploring the use of machine learning for the design of optimal auctions in the presence of private *budget constraints* (Feng et al. 2018). This is a particularly difficult problem, as can be understood from the contours of the few

[15] A training iteration is one mini-batch gradient updated in the solver that we use for stochastic gradient descent.

theoretical results that are available. Even the optimal DSIC single-item, multiple bidder problem is not fully understood.[16]

The REGRETNET architecture can be extended to handle bidders with budget constraints, as well as to allow for BIC rather than DSIC where that is of interest. In regard to private budgets, these are handled by introducing an additional penalty (a *budget penalty*) to penalize payments above reported budget, and regret is modified to only consider deviations for which the payment is no greater than a bidder's true budget. In regard to BIC, a bidder's *interim* regret is estimated as the maximum, over a set of possible misreports, of the average increase in utility from deviation given a set of possible reports by others.

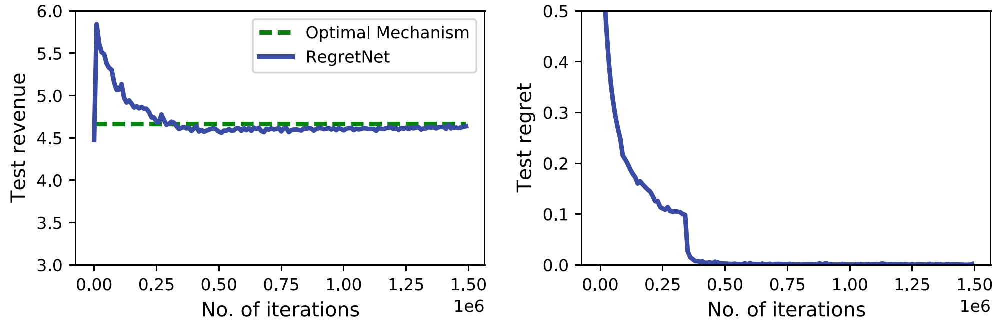
We obtain positive results for various auction environments, including settings for which there are no theoretical results for optimal design (Feng et al. 2018). An illustrative result is shown in Fig. 4. This is for auctioning a single item to two bidders, each with value uniform on set $\{1, 2, \ldots, 10\}$, where the first bidder is unconstrained and the second bidder has a budget of 4. In this case, we only consider the case of downward misreports of budget (conditional DSIC) and the optimal result is shown in Malakhov and Vohra (2008). Figure 4a shows the test revenue and *ex post* regret for the mechanism learned by REGRETNET, as a function of the number of solver iterations. The trained mechanism yields revenue very close to the optimal revenue, while yielding negligible regret. Figure 4b, c show that the learned allocation rule closely matches the optimal allocation rule.

Sticking within the context of auctions, we believe the REGRETNET architecture (or related approaches) will allow rapid experimentation in the following of directions:

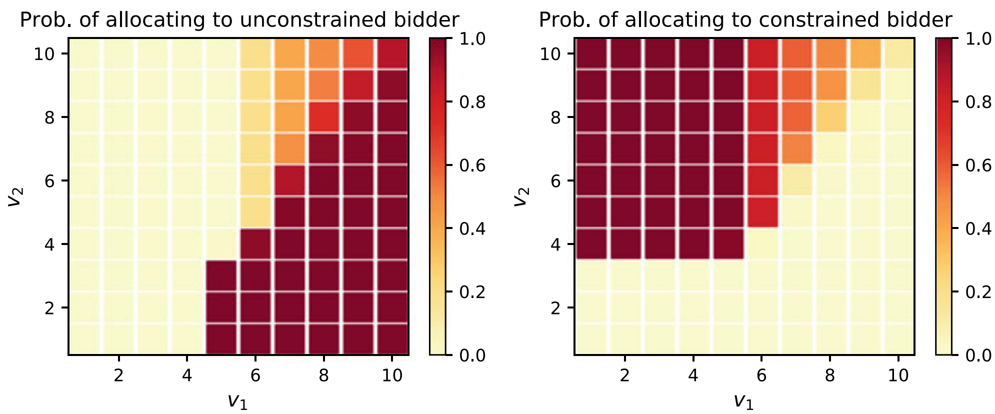
- Correlated, private values; interdependent values.
- Comparing the revenue properties of DSIC and BIC auctions.
- Various kinds of budgeted settings, including budgets that depend on outcomes, and settings where there are correlations between budgets and values.
- Auctions that are robust against deviations by groups of bidders (i.e., properties such as *group strategy-proofness* and its relaxations).
- Auctions that are *envy-free*, so that one bidder does not envy the allocation of another.
- Auctions that satisfy *core* properties, so that no group of participants can do better by breaking away from the auction.
- Revenue-optimal, combinatorial auctions.

The REGRETNET architecture can also be used for problems of mechanism design without money. To illustrate this, we have results for the *K-facility location problem* (Golowich et al. 2018). This problem generalizes the single-facility, 1-dimensional location problem under single-peaked preferences (Moulin 1980) to

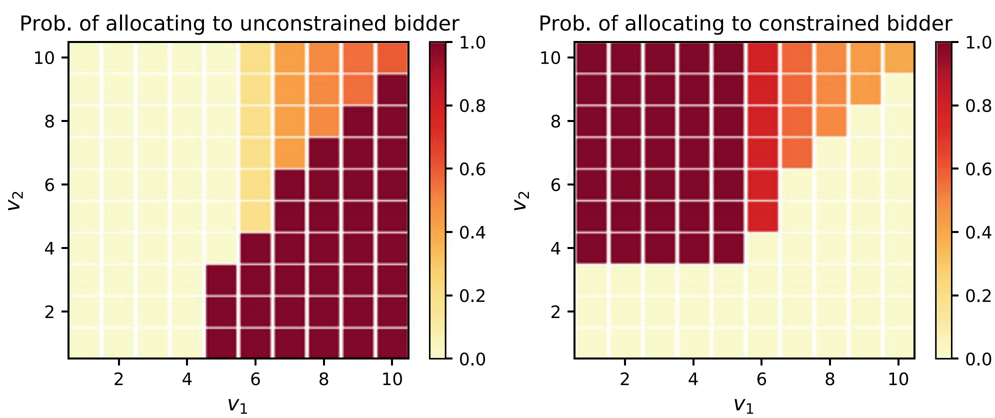
[16]Pai and Vohra (2014) design the optimal, single-item BIC auction, while Malakhov and Vohra (2008) provide the state-of-the-art result for the optimal, single-item DSIC auction (for two bidders, and with a weaker "conditional" form of DSIC). These results build on earlier results for more stylized settings (Che and Gale 1998, 2000; Maskin 2000; Laffont and Robert 1996). There are also a few approximation results for DSIC and BIC designs (Borgs et al. 2005; Bhattacharya et al. 2012; Chawla et al. 2011).



(a) Revenue and regret as a function of iteration count



(b) Learned allocation rule



(c) Optimal allocation rule

Fig. 4 REGRETNET (extended to handle private budgets) for a single item, two bidder auction, where bidder values $v_1, v_2 \sim \mathit{Unif}\{1, 2, ..., 10\}$, bidder 1 is unconstrained, and bidder 2 has a budget of 4. (b) and (c) compare the trained allocation rule and the optimal allocation rule, illustrating the probability of assigning the item to each bidder for different values (v_1, v_2). Based on based on Fig. 4 in Feng et al. (2018)

consider multiple facilities. An agent's utility depends on the distance between its peak and the closest facility. No general characterization result is available for DSIC mechanisms for facility location with $K \geq 2$ facilities.[17] Moreover, a series of negative results show the impossibility of achieving a good, worst-case approximation to the optimal social cost (Procaccia and Tennenholtz 2013; Fotakis and Tzamos 2014, 2016). Here, social cost is the total, negated distance from each agent to its closest facility.

We assume that each agent has a single-peaked utility function, i.i.d. drawn from some distribution, and look to minimize the expected social cost, i.e. the sum of the agents' costs. The architecture is modified to REGRETNET-NM, where the input layer is defined to include one unit for the position of each agent's peak and the output layer includes one unit for each facility, representing its location.[18] The training problem is solved through augmented Lagrangian.

We compare the social cost with that of the best percentile, dictatorial, and constant rules, as well as the optimal (non DSIC) rule.[19] We vary the number of facilities and number of agents. In each case, agent peaks are i.i.d. sampled, uniform on [0, 1]. In one variation, weighted social cost is also considered, where the designer may associate a different weight with each agent. Table 2 summarizes the results for the case of $K = 4, n = 5$, as well as $K = 3, n = 9$ for a weighted objective. The expected, per-agent *ex post* regret is low, and the performance of REGRETNET-NM is competitive with the best percentile rule for the unweighted case and better than the best percentile rule for the weighted case. Figure 5 illustrates the social choice rule that is learned in each of these two environments. This shows the histogram on percentiles of reports for each of the facilities.[20]

For $K = 4$ and $n = 5$, the concentration around $i/4$ for $i \in \{0, 1, 2, 3, 4\}$ in Fig. 5a indicates that the behavior of the learned rule is close to that of a percentile rule, almost always choosing the min and the max peaks, but making different choices about which reports to use for the other two facilities. For $K = 3$ and $n = 9$, and with a weighted objective that places a high weight on agents 1 and 2, we see in Fig. 5b that

---

[17] Ehlers and Gordon (2011) and Heo (2013) provide characterizations for the special case of $K = 2$ under additional assumptions. Heo (2013) assumes anonymity and an additional property, *users only*, which means that agents cannot influence the locations of facilities they will not use. Ehlers and Gordon (2011) assume that agents have *lexmax* preferences over facilities, and thus do not only care about the peak closest to them.

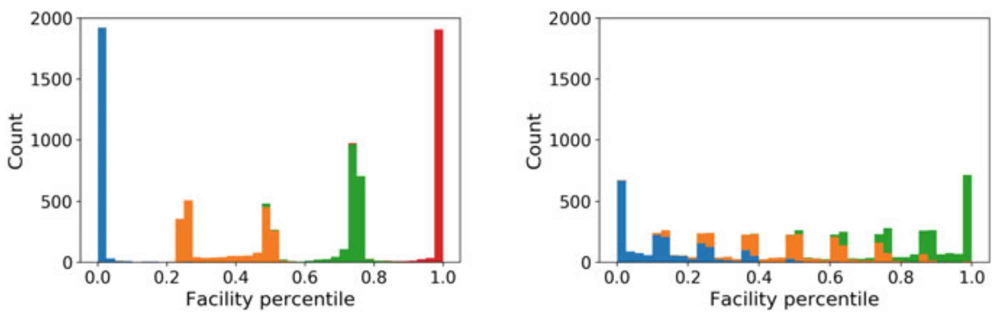
[18] For the single facility location problem, it is w.l.o.g. to consider mechanisms that operate on agent peaks (Border and Jordan 1983). This extends also to a more general "voting under constraints" setting (Barberà et al. 1997). For multiple facilities there are DSIC mechanisms that do not depend only on agent peaks. For example, one can consider the example of $K = 2$ facilities and $n = 2$, where one facility is placed at the peak of agent 1 and the other at some location that depends on the shape of agent 1's report. Still, we retain this simple representation in our current work.

[19] In a multi-facility, dictatorial rule, each facility is determined by a separate dictatorial rule. This is equivalent to having a serial-dictatorial rule for all K facilities. In a multi-facility, percentile rule, each facility is determined by a separate percentile rule (Sui et al. 2013). A constant rule places each facility in the same location all the time.

[20] If $p_1, \ldots, p_n$ are the agent peaks in sorted order, then a facility at location x has percentile 0 if $x \leq p_1$, has percentile 1 if $x \geq p_n$, and has percentile $\frac{i-1}{n-1} + \frac{x-p_i}{(n-1)(p_{i+1}-p_i)}$ if $p_i \leq x < p_{i+1}$.

Table 2 (Weighted) social cost and regret for REGRETNET-NM, comparing to the best percentile, best dictatorial, best constant, and optimal (non-DSIC) rules. K is the number of facilities, n the number of agents (Golowich et al. 2018)

Environment	Percentile	Dictatorial	Constant	Optimal	REGRETNET	
	Social cost	Social cost	Social cost	Social cost	Social cost	Regret
$K = 4, n = 5$	**0.017**	0.024	0.064	0.0083	0.018	0.0024
$K = 3, n = 9$, weighted	0.056	0.053	0.085	0.032	**0.041**	0.0005



(a) $K = 4$ facilities, $n = 5$ agents (b) $K = 3$ facilities, $n = 9$ agents

Fig. 5 Histograms of facility percentiles chosen by REGRETNET-NM. For each instance, the locations selected by the network are sorted and shown in different colors. Blue is the percentile of the left-most facility, then orange, then green, then red (for $K = 4$). Figure 5a is from Golowich et al. (2018)

the learned rule frequently places each facility at a location that corresponds to one of the reported peaks. In fact, we can confirm that the network learns to approximately treat agents 1 and 2 as dictators.

More generally, we anticipate that this framework can be successfully applied to the following variations on facility location, as well as more general problems of mechanism design without money:

- Facility location problems in multiple dimensions, as well as problems with multiple capacitated facilities (see Aziz et al. 2018 for a single facility version).
- Matching problems, for example optimizing expected welfare in one-sided and two-sided matching problems (including considerations of incentive alignment or stability).
- Voting problems, for example to maximize expected welfare subject to various axiomatic properties.

5 Looking Forward

We believe that computational methods, and machine learning in particular, can power a new generation of optimal economic design. We look to learn optimal designs from data that we generate that represents the distribution on agent types, and capture economic constraints such as incentive compatibility within the training problem, and an objective such as revenue via an appropriate loss function. There are many fundamental research questions to pursue, including questions that are responsive to the earlier discussion— robustness, interpretability, and understanding computation-theoretic and learning-theoretic barriers. The approach is not limited to auction design, but can extend to other problems of economic design such as those of optimal contract theory as well as to problems without money. Success will enable bespoke, optimal designs to be deployed in all corners of the rapidly emerging digital economy. Indeed, automated design looks like a necessary response to a future in which we should expect the increasing adoption of algorithmic and AI methods for economic decision making. Advances in the use of computational methods for design will also provide tools with which to validate and advance the theory of optimal economic design.

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

References

Abadi, M., Chu, A., Goodfellow, I. J., McMahan, H. B., Mironov, I., Talwar, K., et al. (2016). Deep learning with differential privacy. *Proceedings of the ACM Conference on Computer and Communications Security* (pp. 308–318).

Alaei, S., Fu, H., Haghpanah, N., & Hartline, J. D. (2013). The simple economics of approximately optimal auctions. *Proceedings of the 54th IEEE Symposium on Foundations of Computer Science* (pp. 628–637).

Alaei, S., Fu, H., Haghpanah, N., Hartline, J. D., & Malekian, A. (2012). Bayesian optimal auctions via multi- to single-agent reduction. *Proceedings of the 13th ACM Conference on Electronic Commerce* (p. 17).

Alaei, S. (2014). Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM Journal on Computing*, *43*, 930–972.

Albert, M., Conitzer, V., & Stone, P. (2017). Automated design of robust mechanisms. *Proceedings of the 31st AAAI Conference on Artificial Intelligence* (pp. 298–304).

Aziz, H., Chan, H., Lee, B. E., & Parkes, D. C. (2018). Mechanism design without money for common goods. arXiv:1806.00960.

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

Barberà, S., Massó, J., & Neme, A. (1997). Voting under constraints. *Journal of Economic Theory*, *76*, 298–321.

Bhalgat, A., Gollapudi, S., & Munagala, K. (2013). Optimal auctions via the multiplicative weight method. *Proceedings of the 14th ACM Conference on Economics and Computation* (pp. 73–90).

Bhattacharya, S., Goel, G., Gollapudi, S., & Munagala, K. (2012). Budget-constrained auctions with heterogeneous items. *Theory of Computing*, *8*, 429–460.

Bikhchandani, S., Chatterji, S., Lavi, R., Mu'alem, A., Nisan, N., & Sen, A. (2006). Weak monotonicity characterizes deterministic dominant-strategy implementation. *Econometrica*, *74*, 1109–1132.

Border, K. C., & Jordan, J. S. (1983). Straightforward elections, unanimity and phantom voters. *Review of Economic Studies*, *50*, 153.

Borgs, C., Chayes, J., Immorlica, N., Mahdian, M., & Saberi, A. (2005). Multi-unit auctions with budget-constrained bidders. *Proceedings of the 6th ACM Conference on Electronic Commerce* (pp. 44–51).

Cai, Y., & Daskalakis, C. (2017). Learning Multi-Item Auctions with (or without) Samples. *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science* (pp. 516–527).

Cai, Y., & Huang, Z. (2013). Simple and nearly optimal multi-item auctions. *Proceedings of the 24th ACM-SIAM Symposium on Discrete Algorithms* (pp. 564–577).

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

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

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

Cai, Y., Daskalakis, C., & Weinberg, S. M. (2013a). Reducing revenue to welfare maximization: approximation algorithms and other generalizations. *Proceedings of the 24th ACM-SIAM Symposium on Discrete Algorithms* (pp. 578–595).

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

Cai, Y., Devanur, N. R., & Weinberg, S. M. (2016). A duality based unified approach to bayesian mechanism design. *Proceedings of the 48th ACM Symposium on Theory of Computing* (pp. 926–939).

Cai, Y., & Daskalakis, C. (2015). Extreme value theorems for optimal multidimensional pricing. *Games and Economic Behavior*, *92*, 266–305.

Carroll, G. (2013). *A Quantitative Approach to Incentives: Application to Voting Rules*. Technical report, Microsoft Research and Stanford University.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. *Proceedings of the 21sth ACM Conference on Knowledge Discovery and Data Mining* (pp. 1721–1730).

Chawla, S., Hartline, J.D., & Kleinberg, R. D. (2007). Algorithmic pricing via virtual valuations. *Proceedings of the 8th ACM Conference on Electronic Commerce* (pp. 243–251).

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

Chawla, S., Malec, D. L., & Malekian, A. (2011). Bayesian mechanism design for budget-constrained agents. *Proceedings of the 12th ACM Conference on Electronic Commerce* (pp. 253–262).

Che, Y.-K., & Gale, I. (1998). Standard auctions with financially constrained bidders. *The Review of Economic Studies*, *65*, 1–21.

Che, Y.-K., & Gale, I. (2000). The optimal mechanism for selling to a budget-constrained buyer. *Journal of Economic Theory*, *92*, 198–233.

Chen, R., Lucier, B., Singer, Y., & Syrgkanis, V. (2017). Robust optimization for non-convex objectives. *Proceedings of the 21st Conference on Neural Information Processing Systems* (pp. 4708–4717).

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

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

Conitzer, V., & Sandholm, T. (2003). Applications of automated mechanism design. *Proceedings of the 4th Bayesian Modelling Applications Workshop*.

Daskalakis, C., Deckelbaum, A., & Tzamos, C. (2014). The complexity of optimal mechanism design. *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 1302–1318).

Daskalakis, C., Deckelbaum, A., & Tzamos, C. (2017). Strong duality for a multiple-good monopolist. *Econometrica* (pp. 735–767).

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

Devanur, N. R., Huang, Z., & Psomas, C. -A. (2016). The sample complexity of auctions with side information. *Proceedings of the 48th ACM Symposium on Theory of Computing* (pp. 426–439).

Doshi-Velez, F., Wallace, B. C. & Adams, R. (2015). Graph-sparse LDA: A topic model with structured sparsity. *Proceedings of the 29th AAAI Conference on Artificial Intelligence* (pp. 2575–2581).

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

Dütting, P., Feldman, M., Kesselheim, T., & Lucier, B. (2017). Prophet inequalities made easy: stochastic optimization by pricing non-stochastic inputs. *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science* (pp. 540–551).

Dütting, P., Feng, Z., Narasimhan, H., Parkes, D. C., & Ravindranath, S. S. (2019). Optimal auctions through deep learning. *Proceedings of the 36th International Conference on Machine Learning* (first version, 2017)

Dütting, P., Fischer, F. A., Jirapinyo, P., Lai, J. K., Lubin, B., & Parkes, D. C. (2015). Payment rules through discriminant-based classifiers. *ACM Transactions on Economics and Computation*, *3*, 5:1–5:41.

Ehlers, L., & Gordon, S. (2011). *Strategy-Proof Provision of Two Public Goods: The LexMax Extension*. Technical report, Mimeo Université de Montréal.

Elkind, E. (2007). Designing and learning optimal finite support auctions. *Proceedings of the 18th ACM-SIAM Conference on Discrete Algorithms* (pp. 736–745).

Fawzi, A., Fawzi, O., & Frossard, P. (2018). Analysis of classifiers' robustness to adversarial perturbations. *Machine Learning*, *107*, 481–508.

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

Fotakis, D., & Tzamos, C. (2014). On the power of deterministic mechanisms for facility location games. *ACM Transactions on Economics and Computation*, *2*, 15:1–15:37.

Fotakis, D., & Tzamos, C. (2016). Strategyproof facility location for concave cost functions. *Algorithmica*, *76*, 143–167.

Giannakopoulos, Y. & Koutsoupias, E. (2014). Duality and optimality of auctions for uniform distributions. *Proceedings of the 15th ACM Conference on Economics and Computation* (pp. 259–276).

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

Giannakopoulos, Y. (2015). Bounding the optimal revenue of selling multiple goods. *Theoretical Computer Science*, *581*, 83–96.

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

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

Goodfellow, I. J., Bengio, Y., & Courville, A. C. (2016). *Deep Learning, Adaptive Computation and Machine Learning*. MIT Press: Cambridge.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. *Proceedings of the 3rd International Conference on Learning Representations*.

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

Guo, M., & Conitzer, V. (2009). Worst-case optimal redistribution of VCG payments in multi-unit auctions. *Games and Economic Behavior*, *67*, 69–98.

Hart, S., & Nisan, N. (2012). Approximate revenue maximization with multiple items. *Proceedings of the 13th ACM Conference on Economics and Computation* (p. 656).

Heo, E. J. (2013). Strategy-proof rules for two public goods: double median rules. *Social Choice and Welfare*, *41*, 895–922.

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

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations*.

Kleinberg, R., & Weinberg, S. M. (2012). Matroid prophet inequalities. *Proceedings of the 44th ACM Symposium on Theory of Computing* (pp. 123–136).

Laffont, J.-J., & Robert, J. (1996). Optimal auction with financially constrained buyers. *Economics Letters*, *52*, 181–186.

Li, X., & Yao, A. C.-C. (2013). On revenue maximization for selling multiple independently distributed items. *Proceedings of the National Academy of Sciences*, *110*, 11232–11237.

Lubin, B., & Parkes, D. C. (2012). Approximate strategyproofness. *Current Science*, *103*, 1021–1032.

Malakhov, A., & Vohra, R. V. (2008). Optimal auctions for asymmetrically budget constrained bidders. *Review of Economic Design*, *12*, 245.

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

Maskin, E. S. (2000). Auctions, development, and privatization: efficient auctions with liquidity-constrained buyers. *European Economic Review*, *44*, 667–681.

Mennle, T., & Seuken, S. (2014). An axiomatic approach to characterizing and relaxing strategyproofness of one-sided matching mechanisms. *Proceedings of the 15th ACM Conference on Economics and Computation* (pp. 37–38).

Mennle, T., & Seuken, S. (2017). Partial strategyproofness: Relaxing strategyproofness for the random assignment problem. arXiv:1401.3675v4.

Moosavi-Dezfooli, S., Fawzi, A., & Frossard, P. (2016). DeepFool: A simple and accurate method to fool deep neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2574–2582).

Morgenstern, J., & Roughgarden, T. (2015). On the pseudo-dimension of nearly optimal auctions. *Proceedings of the 29th Conference on Neural Information Processing Systems*.

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

Moulin, H. (1980). On strategy-proofness and single-peakedness. *Public Choice*, *35*, 55–74.

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

Narasimhan, H., & Parkes, D. C. (2016). A general statistical framework for designing strategy-proof assignment mechanisms. *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence* (pp. 527–536).

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

Niepert, M., Ahmed, M., & Kutzkov, K. (2016). Learning convolutional neural networks for graphs. *Proceedings of The 33rd International Conference on Machine Learning* (Vol. 48, pp. 2014–2023).

Orhan, A. E., & Ma, W. J. (2017). Efficient probabilistic inference in generic neural networks trained with non-probabilistic feedback. *Nature Communications*, *8*, 138.

Pai, M. M., & Vohra, R. (2014). Optimal auctions with financially constrained buyers. *Journal of Economic Theory*, *150*, 383–425.

Parkes, D. C., & Wellman, M. P. (2015). Economic reasoning and artificial intelligence. *Science*, *349*, 267–272.

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

Procaccia, A. D., & Tennenholtz, M. (2013). Approximate mechanism design without money. *ACM Transactions on Economics and Computation*, *1*, 1–26.

Procaccia, A., Zohar, A., Peleg, Y., & Rosenschein, J. (2009). The learnability of voting rules. *Artificial Intelligence*, *173*, 1133–1149.

Raghu, M., Poole, B., Kleinberg, J. M., Ganguli, S., & Sohl-Dickstein, J. (2016). Survey of expressivity in deep neural networks. arXiv:1611.08083.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should i trust you?": explaining the predictions of any classifier. *Proceedings of the 22nd ACM Conferecne on Knowlegde Discovery and Data Mining* (pp. 1135–1144).

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

Ross, A. S., Hughes, M. C., & Doshi-Velez, F. (2017). Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations. *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (pp. 2662–2670).

Rothkopf, M. H., Pekec, A., & Harstad, R. M. (1998). Computationally manageable combinational auctions. *Management Science*, *44*, 1131–1147.

Rubinstein, A., & Weinberg, S. M. (2015). Simple mechanisms for a subadditive buyer and applications to revenue monotonicity. *Proceedings of the 16th ACM Conference on Economics and Computation* (pp. 377–394).

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

Shalev-Shwartz, S., & Wexler, Y. (2016). Minimizing the maximal loss: how and why. *Proceedings of the 33rd International Conference on Machine learning* (pp. 793–801).

Smilkov, D., Thorat, N., Nicholson, C., Reif, E., Viégas, F. B., & Wattenberg, M. (2016). Embedding projector: interactive visualization and interpretation of embeddings. arXiv:1611.05469.

Sui, X., Boutilier, C., & Sandholm, T. (2013). Analysis and optimization of multi-dimensional percentile mechanisms. *Proceedings of the 23rd International Joint Conference on Artificial Intelligence* (pp. 367–374).

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., & Fergus, R. (2014). Intriguing properties of neural networks. *Proceedings of the 2nd International Conference on Learning Representations*.

Thirumulanathan, D., Sundaresan, R., & Narahari, Y. (2016). Optimal mechanism for selling two items to a single buyer having uniformly distributed valuations. *Proceedings of the 12th International Conference on Web and Internet Economics* (pp. 174–187).

Thirumulanathan, D., Sundaresan, R., & Narahari, Y. (2017): On optimal mechanisms in the two-item single-buyer unit-demand setting. arXiv:1705.01821.

Vinyals, O., Fortunato, M., & Jaitly, N. (2015). Pointer networks. *Proceedings of the 29th Conference on Neural Information Processing Systems* (pp. 2692–2700).

Wang, F., & Rudin, C. (2015). Falling rule lists. *Proceedings of the 18th Conference on Artificial Intelligence and Statistics*.

Wright, S., & Nocedal, J. (1999). Numerical optimization. *Springer Science*, *35*, 67–68.

Xia, L. (2013). Designing social choice mechanisms using machine learning. *Proceedings of the 13th Conference on Autonomous Agents and Multi-Agent Systems* (pp. 471–474).

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

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