# Differentiable Economics for Randomized Affine Maximizer Auctions

Michael Curry[1] Tuomas Sandholm[2 3 4 5] John Dickerson[1]

## Abstract

A recent approach to automated mechanism design, differentiable economics, represents auctions by rich function approximators and optimizes their performance by gradient descent. The ideal auction architecture for differentiable economics would be perfectly strategyproof, support multiple bidders and items, and be rich enough to represent the optimal (i.e. revenue-maximizing) mechanism. So far, such an architecture does not exist. There are single-bidder approaches (MenuNet, RochetNet) which are always strategyproof and can represent optimal mechanisms. RegretNet is multi-bidder and can approximate any mechanism, but is only approximately strategyproof. We present an architecture that supports multiple bidders and is perfectly strategyproof, but cannot necessarily represent the optimal mechanism. This architecture is the classic affine maximizer auction (AMA), modified to offer lotteries. By using the gradient-based optimization tools of differentiable economics, we can now train lottery AMAs, competing with or outperforming prior approaches in revenue.

## 1. Introduction

Auctions are a widely-used mechanism for allocating scarce items that are for sale, in which a centralized auctioneer solicits bids from auction participants, and based on those bids, allocates the items (possibly keeping some of them) and charges some payments. The auctioneer may wish to design the auction to achieve some goal. The usual assumption is that the auctioneer has access to a prior distribution over bidders' valuations. Typically, it is also desired that the auction be strategyproof, that is, there should be no incentive for bidders to be untruthful in their bids about their valuations.

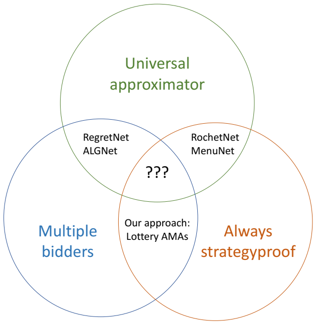

Figure 1: Our architecture in relation to other techniques from differentiable economics for multi-item revenue-maximizing auction design. The "holy grail" in the middle of the Venn diagram—that is, techniques that can represent (i) any auction for (ii) general numbers of bidders and items while (iii) guaranteeing strategyproofness—has not been achieved; however, we show that our method achieves (iii) strategyproofness-by-design for (ii) general numbers of items and bidders while still improving revenue over baselines.

When the auctioneer wants to maximize the total welfare of the bidders, the *Vickrey-Clarke-Groves (VCG)* mechanism, which is always strategyproof, is also optimal (Vickrey, 1961; Clarke, 1971; Groves, 1973). When the auctioneer instead wants to maximize her revenue (or profit), the problem is significantly more challenging.

Myerson (1981) settled the revenue-maximizing strategyproof auction problem when there is one item for sale. Maskin & Riley (1989) generalized that mechanism to the case of multiple copies of a single item. However, four decades later, the multi-item revenue-maximizing auction is still unknown. Special cases of the two-item setting have been solved (Armstrong, 2000; Avery & Hendershott, 2000). There is some theory of strong duality (Daskalakis et al., 2017; Kash & Frongillo, 2016) for selling multiple items to a single agent. There have also been some successes for the weaker notion of Bayesian incentive compatibility (Cai et al., 2012b;a; 2013). But for designing dominant-strategy incentive compatible mechanisms that sell multiple items to multiple agents there has been little progress despite decades of research. Yao (2017) presents a result for one special case, giving an explicit example of a

[1] University of Maryland [2] Carnegie Mellon University [3] Optimized Markets, Inc. [4] Strategic Machine, Inc. [5] Strategy Robot, Inc.. Correspondence to: Michael Curry <curry@cs.umd.edu>.

Preprint.

revenue gap between the best dominant-strategy incentive compatible mechanism and the best Bayes-Nash incentive compatible mechanism. Nevertheless, the problem is wide open. Even for the seemingly trivial case of two agents with i.i.d. uniform valuations over two items, the optimal selling mechanism is not known.

In part motivated by the fact that the theory on this question has essentially gotten stuck for decades, Conitzer & Sandholm (2002); Sandholm (2003) introduced the idea of *automated mechanism design (AMD)*: designing the mechanism computationally for the problem instance at hand, as opposed to trying to analytically derive a general form for the revenue-maximizing multi-item auction. AMD has since become a popular research topic. Three different high-level approaches to AMD have been introduced: 1) designing the mechanism from scratch in tabular form (Conitzer & Sandholm, 2002), 2) conducting search over the parameters of a mechanism class where all the mechanisms in the class have some desirable properties such as strategyproofness and individual rationality (the latter incentivizes buyers to participate) (Likhodedov & Sandholm, 2004; 2005; Sandholm & Likhodedov, 2015), and 3) *incremental mechanism design* where the design starts from some (typically well-known but not strategyproof) mechanism and then keeps making changes to the mechanism to improve it (Conitzer & Sandholm, 2007).

A recent form of incremental mechanism design that capitalizes on the modern power of deep learning is called *differentiable economics*. Duetting et al. (2019) introduced the use of deep neural networks as function approximators to learn auctions. Their RegretNet architecture learns approximately strategyproof auctions for multi-bidder multi-item auctions. MenuNet (Shen et al., 2019) and RochetNet (Duetting et al., 2019) are restricted to a single bidder, but enforce strategyproofness at the architectural level.

## 2. Our Contributions

Ideally, we would like an auction architecture that 1) supports multiple agents and items, 2) is perfectly strategyproof by construction, and 3) is always rich enough to represent the true optimal auction, given enough parameters. Such an architecture does not yet exist. RegretNet achieves 1 and 3 only; RochetNet and MenuNet achieve 2 and 3. In our work, we present an approach that achieves 1 and 2, though not 3 – **a multi-bidder, multi-item auction architecture which is always perfectly strategyproof**.

Consider a classic tool for automated mechanism design – the family of *affine maximizer auctions (AMAs)* (Roberts, 1979). AMAs are essentially versions of the VCG mechanism, modified by associating a positive “weight” to each bidder’s welfare and adding potentially different “boosts” to all the possible allocations. AMAs are always strategyproof and individually rational like VCG, but revenue can be significantly increased over VCG by tuning these parameters (weights and boosts). Importantly, this can be done by just using *samples* of the valuation distribution (Likhodedov & Sandholm, 2004; 2005; Sandholm & Likhodedov, 2015) rather than the traditional mechanism design approach of taking the full valuation distribution as input, which would be prohibitively complex in these combinatorial settings. Later work considers the number of samples needed for this in a learning-theoretic sense (Balcan et al., 2016; 2018; 2021).

Our contribution is to revisit the problem of learning AMAs, now with differentiable economics. One can view the paper from at least the following perspectives:

1. It can be seen as an extension of previous work on learning AMAs, now **allowing for lottery allocations**. This means not only learning the weights and boosts, but also learning over the (continuous) set of lotteries to offer. Randomization can increase revenue.

2. It can be seen as a **multi-bidder generalization of RochetNet and MenuNet**. Restricting our lottery AMAs to a single bidder essentially recovers these architectures, and for multiple bidders, strategyproofness is still guaranteed by construction. (However, for general multi-bidder combinatorial auction settings, AMAs cannot represent every truthful mechanism; there is no guarantee they can represent an optimal one.)

3. It provides a **more interpretable** family of mechanisms to learn using differentiable economics. RegretNet-style auctions are opaque: they map bid profiles to outcomes in an arbitrary way. In contrast, the rules for determining outcomes of an AMA are easy to explain. Moreover, by the end of training, our learned mechanisms typically have a small number of possible outcomes which are easily summarized.

## 3. Related work

### 3.1. Differentiable economics

**RegretNet** (Duetting et al., 2019) use the tools of modern deep learning to learn revenue-maximizing mechanisms. In particular, they present the RegretNet neural architecture. The idea is to treat an auction mechanism as a function mapping bid profiles to allocations and payments, and directly approximate this function using a neural network. The loss function consists of a term for revenue maximization, and another term for minimizing *regret* – violations

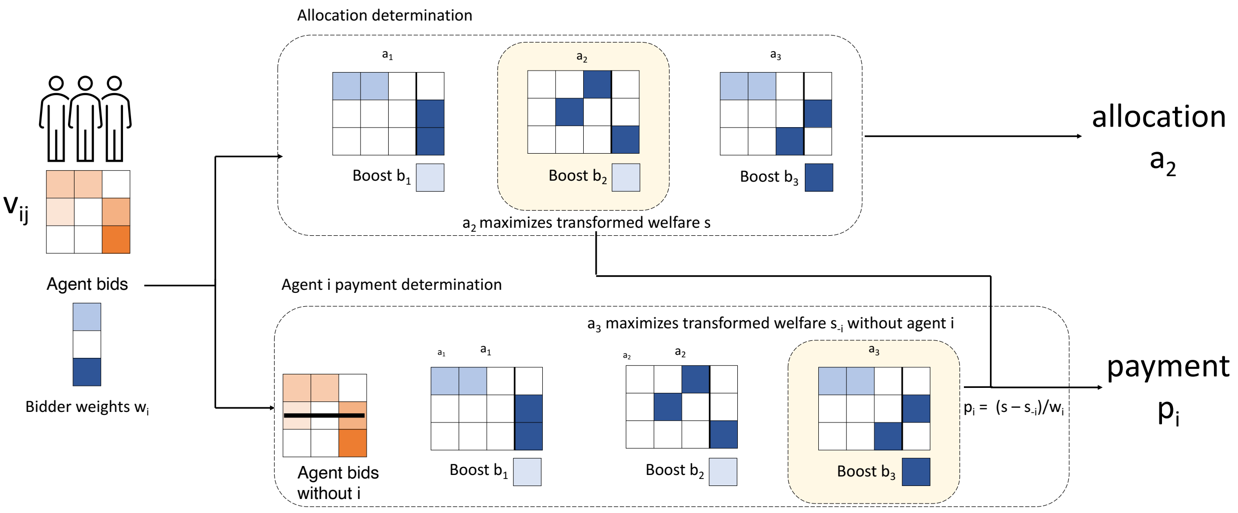

Figure 2: A schematic of an affine maximizer auction. The learned parameters—allocations, bidder weights, and allocation boosts—are represented in blue. (In a traditional AMA, allocations are not learned.) The chosen allocation maximizes transformed total welfare; the payment for each bidder $i$ is the difference in transformed welfare between the chosen allocation, and what would have been chosen without taking bidder $i$ into account.

of strategyproofness. RegretNet works quite well, approximately recovering some known optimal auctions and outperforming other approaches.

However, its approach has several limitations. In particular, the learned auctions are only approximately strategyproof – there is still some small presence of regret, and moreover the presence of regret can only be measured empirically. Curry et al. (2020) provides a way to exactly compute regret, which mitigates this latter limitation. But the former problem remains – a mechanism learned using the RegretNet approach is not guaranteed to be perfectly strategyproof.

### 3.2. Characterizing strategyproof mechanisms

Rochet (1987) shows that for any mechanism *with a single agent*, strategyproof mechanisms can be identified with convex utility functions (as a function of the agent's true type). Any strategyproof pair of allocation and payment rules will induce a convex utility function. An allocation rule can be derived from any convex utility function by simply taking its gradient (which also fixes the payment rule).

Characterizing strategyproof mechanisms for multiple agents is not so straightforward. Rochet's characterization still holds in this case: fixing other bids, agent $i$'s utility must be convex as a function of their type, and this must hold for all agents and for any choice of opponent bids. However, coming up with some universal approximator for the entire class of functions that has this property is difficult.

### 3.3. Strategyproof architectures

Alongside RegretNet, Duetting et al. (2019) also presents the RochetNet architecture, which is restricted to a single bidder but is perfectly strategyproof.[1] Shen et al. (2019) concurrently present MenuNet, another single-bidder architecture which is perfectly strategyproof.

Both MenuNet and RochetNet offer possibly-randomized sets of menu items at different prices. The bidder maximizes over all offered menu items, inducing a convex utility as a function of the bidder's type. As such, MenuNet and RochetNet will always represent a strategyproof mechanism for any setting of their parameters. And given enough parameters, they are universal approximators for strategyproof mechanisms.

For single-bidder auction design, there is a strong duality result which can be used to prove optimality of a proposed mechanism (Daskalakis et al., 2017; Kash & Frongillo, 2016). The authors of Duetting et al. (2019) and Shen et al. (2019) apply these results to their learned auctions, and discover some previously-unknown optimal auctions.

### 3.4. Further work in differentiable economics

Many papers have built on RegretNet. ALGNet (Rahme et al., 2021b) gives an improved loss function, which has fewer hyperparameters, and an improved training algorithm. We use it as a point of comparison below. Other

[1] In the appendix, they also present MyersonNet, which is restricted to 1 item.

papers apply the same general approach to auctions with fairness or budget constraints (Kuo et al., 2020; Peri et al., 2021; Feng et al., 2018), add new inductive biases to the architecture (Curry et al., 2021; Rahme et al., 2021a), or apply similar techniques to other mechanism design problems (Ravindranath et al., 2021; Golowich et al., 2018; Brero et al., 2021).

Another line of work uses neural networks to model agent preferences over possible bundles (Tacchetti et al., 2019; Weissteiner & Seuken, 2020; Brero et al., 2019b;a; Bachrach et al., 2021). Bichler et al. (2021) uses ML techniques to compute equilibrium strategies for non-incentive-compatible auctions.

### 3.5. Automated mechanism design and learning theory for auctions

Affine maximizer auctions (AMAs) are classic tools for automated mechanism design (Sandholm & Likhodedov, 2015; Likhodedov & Sandholm, 2004; 2005). In essence, AMAs are just weighted versions of the celebrated Vickrey-Clarke-Groves (VCG) mechanism (Vickrey, 1961; Clarke, 1971; Groves, 1973). VCG chooses the welfare-maximizing allocation; an AMA maximizes a rescaled and shifted version of the welfare. By choosing the parameters of the AMA carefully, performance on metrics other than welfare maximization can be improved without sacrificing strategyproofness. Previous work considers the problem of learning high-performing AMAs from samples using gradient based methods (Sandholm & Likhodedov, 2015; Likhodedov & Sandholm, 2004), albeit using different techniques and without considering lotteries. Guo et al. (2017) computes AMA parameters via linear programming for a particular problem setting. Other works consider the sample complexity of learning AMAs, treating them as a parameterized function class (Balcan et al., 2016; 2018; 2021). Tang & Sandholm (2012) considers a subset of AMAs for which the optimal revenue can be computed in closed form. Deng et al. (2021) tunes the parameters of a class of AMAs to improve performance in an online advertising application.

### 3.6. Lotteries and menu size complexity

There are a number of theoretical results showing that offering lotteries can improve revenue (Briest et al., 2010; Pavlov, 2011; Daskalakis et al., 2017). Hart & Nisan (2019) analyze this phenomenon and give an interesting perspective – in the most general sense, it is not offering lotteries *per se* that improves revenue. Rather, it is that offering more menu items can improve revenue by allowing finer price discrimination, and there are always fewer deterministic allocations than possible lotteries.

These results, however, give worst-case revenue gaps across whole classes of valuations, not a guarantee for any specific instance. As discussed below, we find that even when our mechanisms can improve their revenue by offering lotteries, they offer relatively few menu items, so performance improvements are not due to increased menu size.

### 3.7. Expressiveness of AMAs and Roberts's Theorem

To what extent can the class of affine maximizer auctions actually express the optimal strategyproof auction? As mentioned, Rochet (1987) shows that all single-agent strategyproof mechanisms can be identified with convex functions. For multi-agent multi-item settings with unrestricted valuations (meaning every agent may get any positive or negative utility from any outcome, and may even care about which particular items other agents receive), Roberts (1979) shows that every strategyproof mechanism must take the form of an AMA.

The settings we consider here do not have unrestricted valuations, so Robert's theorem does not apply. In particular, Roberts's theorem does not hold for deterministic combinatorial auctions where valuations are monotonically increasing in receiving more items, and the empty set has zero value. All the valuations we consider have these properties. On the other hand, for many settings, (Lavi et al., 2003) shows that any implementable allocation rule which satisfies certain natural conditions must be "almost" an AMA in a certain technical sense.

## 4. Affine Maximizer Auctions

### 4.1. Combinatorial Auction Setting

Consider a setting in which $m$ auction participants are bidding on $n$ items. Each bidder has a private type $v_i \in \mathbb{R}^n$ denoting how much they value each item.

Allocations consist of matrices $a \in \mathbb{R}_+^{mn}$, where $a_{ij}$ denotes the amount of item $j$ given to bidder $i$. We require that $\sum_j a_{ij} \leq 1$, so that no item is overallocated. For deterministic auctions, we require that $a_{ij} \in \{0, 1\}$. For unit-demand auctions, we also require that every bidder receives at most 1 item: $\sum_i a_{ij} \leq 1$. Denote the set of feasible allocations for a given setting by $A \subset \mathbb{R}^{mn}$. We will often treat $A$ as a set with elements $a_k$. Payments $p_i$ are simply positive scalars. Given an allocation, bidder $i$ receives utility $u_i = \sum_j a_{ij} v_{ij} - p_i$.

The regret for player $i$ under a given bid profile is defined as the difference in utility between bidding truthfully and the best strategic misreport:

$$\mathrm{rgt}_i(v) = \max_{b_i} u_i(b_i, v_{-i}) - u_i(v_i)$$

When regret is 0 for every player, and for every bid profile, the auction is dominant-strategy incentive compatible (DSIC). In this work, all our auctions have guaranteed zero regret, but some of our baselines may have positive regret.

In addition to requiring our auctions to be DSIC, we also require individual rationality (IR) – that is, $u_i \geq 0$ for every bidder, or equivalently, no truthful bidder will ever pay more than the value of the items they receive.

### 4.2. Affine Maximizer Auction Mechanism

Affine maximizer auctions have parameters consisting of weights $w_i$ for each bidder and boosts $b_k$ associated with each allocation $a_k$. Given some bids $\boldsymbol{v}$ for each bidder, the affine maximizer auction chooses the allocation (and boost) $a_k, b_k$ that will maximize the weighted, boosted welfare:

$$k^* = \arg\max_k \sum_i w_i \sum_j (a_k)_{ij} \boldsymbol{v}_{ij} + b_k \tag{1}$$

Let $a(\boldsymbol{v}) = a_{k^*}, b(\boldsymbol{v}) = b_{k^*}$.

Then, to compute a payment $p_i$ for bidder $i$, it considers the counterfactual auction result where bidder $i$ did not participate. The total decrease in all other bidder' welfare (weighted and boosted) between this counterfactual auction and the new auction is $p_i$:

$$\begin{aligned} p_i = \frac{1}{w_i} &\left( \sum_{\ell \neq i} \sum_j w_\ell a(\boldsymbol{v}_{-i})_{\ell j} \boldsymbol{v}_{\ell j} + b(\boldsymbol{v}_{-i}) \right) \\ - \frac{1}{w_i} &\left( \sum_{\ell \neq i} \sum_j w_\ell a(\boldsymbol{v})_{\ell j} \boldsymbol{v}_{\ell j} + b(\boldsymbol{v}) \right) \end{aligned} \tag{2}$$

As mentioned above, AMAs (like the VCG mechanism) are always DSIC. To see why this is the case, observe that for any fixed set of bids $v_{-i}$, agent $i$'s utility $u_i(v_i) = \sum_j a(v_i, v_{-i})_{ij} v_{ij} - p_i(v_i, v_{-i})$ will be a pointwise maximum over a set of affine functions (one per possible allocation), and thus convex.

The choice of the above payment rule also ensures IR. We additionally require that allocating nothing and charging nothing always be among the possible outcomes $a_k$, although this is not strictly required to ensure IR.

**Our Approach Generalizes RochetNet and MenuNet** When there is only one bidder, without loss of generality we can fix the weights to one and assume welfare when the single bidder is removed is zero, recovering the max-over-affine representation of a strategyproof single-bidder mechanism. Thus our approach of learning allocations and boosts by gradient descent directly generalizes RochetNet (Duetting et al., 2019) and MenuNet (Shen et al., 2019).

## 5. Learning Affine Maximizers Via Differentiable Economics

AMAs have three types of parameters: the bidder weights $w_i$, the boosts $b_k$, and the allocations $a_k$. (Treating the allocations of AMAs as learned parameters along with the weights and boosts is a contribution of our work.) We assume access to sampled truthful valuations, and learn these parameters jointly via gradient descent on the objective $-\sum_i p_i$.

During training, we use the softmax function as a differentiable surrogate for the max and argmax operations: that is, $\arg\max_k f(a_k) \approx \langle \text{softmax}_\tau(f(a_1), \cdots, f(a_k)), \boldsymbol{a} \rangle$ and $\max_k f(a_k) \approx \langle \text{softmax}_\tau(f(a_1), \cdots, f(a_k)), \boldsymbol{f}(\boldsymbol{a}) \rangle$ As the softmax temperature parameter $\tau$ approaches 0, this approach recovers the true argmax.

Using this soft version of the AMA definition, we directly compute the total payment and differentiate it with respect to the parameters via the Jax autograd system (Bradbury et al., 2018) along with Haiku (Hennigan et al., 2020) and optax (Hessel et al., 2020). At test time, we use the learned parameters in the exact AMA definition, using the regular max operator.

For deterministic auctions, we fix the set $a_k$ to be the set of all feasible allocations. For lottery auctions, we randomly initialize a large (typically $|A| = 4096$) set of allocations – although by the end of training, very few of these are actually used (discussed below).

We parameterize these allocations $a_k$ to ensure that they are always feasible. Following the approach from (Duetting et al., 2019), for additive allocations, each allocation is represented an $m$ by $n + 1$ matrix of unrestricted parameters – the extra column is for a dummy item representing "no allocation". We take an item-wise softmax and truncate the dummy column to generate a feasible allocation. For unit-demand allocations, we follow the approach used in (Ravindranath et al., 2021), applying the softplus operation to two matrices of $m$ by $n$ parameters, normalizing row- and column-wise respectively, and taking the minimum of the result.

## 6. Results

### 6.1. Hyperparameters and training

For lottery AMAs, we allow either 2048 or 4096 allocations. The softmax temperature is 100; we use an Adam optimizer with learning rate of 0.01. We train all auctions for 9000 steps, with $2^{15}$ fresh valuation samples per gradient update. All reported test revenues are on 100000 sampled valuations. Because the valuation distributions are symmetric, in the cases tested below we fix bidder weights to 1.

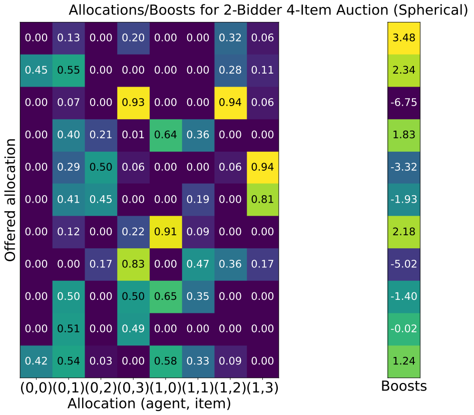

Figure 3: The ten lottery allocations (and their boosts) actually used after training an auction. (The auction has many more parameters, but the 2038 other allocations are never chosen for any of the sampled bids.) One can see that the mechanism does typically offer lotteries.

To determine which allocations are actually used, we sample 100000 test valuations, and include any allocation that was chosen for even one bid profile.

For baselines, we compare against previously reported results from RegretNet (Duetting et al., 2019), ALGNet (Rahme et al., 2021b), and AMAs trained using other methods (Sandholm & Likhodedov, 2015), as well as theoretical revenues from Myerson auctions of separate items and of the grand bundle.

## 6.2. Revenue performance

**Spherical distribution** In order to demonstrate a revenue improvement by offering lotteries, we consider a particular valuation distribution which we refer to as the "spherical distribution" for lack of a better name – this is a distribution on a number of discrete, random points, scaled and normalized according to the proof construction in (Briest et al., 2010).

We construct such a distribution for 4 items with 5 valuation points and consider a setting with two unit-demand bidders, each of whose valuations are sampled i.i.d from this distribution. We would expect a large gap between revenue extracted by lotteries and by a deterministic mechanism.

Indeed, we find that this is the case – when we train our lottery AMA with 2048 allocations on this distribution, it gets more than twice the revenue of a deterministic AMA (see 1). Figure 3 shows the final offered allocations and boosts from a representative mechanism – the auction is actually taking advantage of randomization.

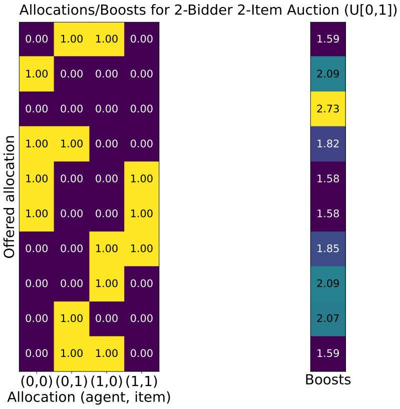

Figure 4: All allocations actually used after training for a 2x2 U[0,1] additive auction (the same setting as compared to in Likhodedov & Sandholm (2004) and Duetting et al. (2019)). Here, although the mechanism space is that of randomized mechanisms, the algorithm learns to offer deterministic allocations. The revenue is comparable to results in Likhodedov & Sandholm (2004).

**2 bidder, 2 item uniform** We also consider a 2 bidder, 2 item additive auction where item values are independently distributed on $U[0, 1]$. This seems like the most trivial possible multi-bidder multi-item auction setting, yet it is so far completely beyond current theory – this makes it an interesting test case for automated mechanism design.

We train a lottery AMA on this setting and find revenue competitive with both previous AMA approaches (Sandholm & Likhodedov, 2015; Tang & Sandholm, 2012) as well as the RegretNet neural network approach (which performs better but is not perfectly strategyproof) (Duetting et al., 2019).

An interesting observation, though, is that even though our lottery AMA is free to offer lotteries, it does not do so – all allocations actually offered by the end of training are deterministic, as seen in Figure 4.

**3 bidder, 10 item uniform** Finally, we consider one of the much larger auction settings from (Duetting et al., 2019) – 3 additive bidders with 10 $U[0, 1]$ items. We give our network parameters for 4096 allocations, many fewer than the number of possible deterministic allocations in this setting.

Results are shown in Table 3, along with baselines. While we do not match the performance of RegretNet and ALGNet, we do at least exceed the performance of the separate Myerson and grand bundling approaches. Some, though probably not most, of the extra revenue gained by RegretNet and ALGNet may be due to non-zero regret.

| AMA Type | Max Revenue | Min Revenue | Mean Revenue | Std Revenue |
|---|---|---|---|---|
| Lottery | 2.158 | 1.87 | 2.06 | 0.098 |
| Deterministic | 1.462 | 0.627 | 0.842 | 0.279 |

Table 1: Results from 8 random parameter initializations, with 2048 allocations, on the spherical valuation distribution. In particular, the worst lottery mechanism outperforms the best deterministic mechanism. Moreover, for this setting, the lottery mechanisms do actually learn to randomize.

| Auction | Best Revenue | Regret |
|---|---|---|
| Lottery AMA (ours) | 0.868 | 0 |
| Combinatorial AMA | 0.862 | 0 |
| Separate Myerson | 0.833 | 0 |
| Grand Bundle | 0.839 | 0 |
| MBARP | 0.871 | 0 |
| RegretNet | 0.878 | $< 0.001$ |
| ALGNet | 0.879 | 0.00058 |

Table 2: Revenue comparison for 2 bidder, 2 item U[0,1] additive auction. Our approach is competitive with other approaches. Combinatorial AMA refers to results from Sandholm & Likhodedov (2015). MBARP is a subset of AMA from Tang & Sandholm (2012) where the optimal parameters have been computed (only for 2 items). RegretNet achieves higher revenue, but possibly due to a small strategyproofness violation. Note that Sandholm & Likhodedov (2015) present many variants, some of which beat our revenue, although all are comparable.

| Auction | Best Revenue | Regret |
|---|---|---|
| Lottery AMA (ours) | 5.345 | 0 |
| Separate Myerson | 5.31 | 0 |
| Grand bundle | 5.009 | 0 |
| RegretNet | 5.541 | 0.002 |
| ALGNet | 5.562 | 0.002 |

Table 3: Revenue comparison for 3 bidder, 10 item U[0,1] additive auction. We train a lottery AMA with 4096 allocations. It underperforms RegretNet (although RegretNet has a small strategyproofness violation), but outperforms the separate Myerson and grand bundling baselines.

We also attempted to train a lottery AMA for the 5 bidder, 10 item uniform case, but found that after several attempts it failed to outperform the separate Myerson baseline.

### 6.3. Number of allocations used

We observe that although our auctions are initialized with many parameters, the number of possible deterministic outcomes may be quite large, the number of allocations used on any actual valuation profile is typically quite small. Results are summarized in Table 4 for all experiments mentioned above.

### 6.4. Effects of parameter initialization

Motivated by the *lottery ticket hypothesis* in neural network training (Frankle & Carbin, 2019), we consider the effects of overparameterization and parameter initialization on performance.

First, we consider training from the same parameter initialization, under a different source of randomness for the data. We find that starting from the same initialization typically results in nearly the same allocation indices being chosen, with Jaccard similarities of .64, .67, and .82 across the indices chosen under the new random data. Starting from a different parameter initialization, there was no overlap in the indices chosen. We find that the results are quite similar, which suggests that parameter initialization is important in determining the end results.

We also consider the opposite approach: take the final actually-used allocations, look at what values those parameters took at initialization before training, and retrain using only those parameters. In other words, we train a model with very few parameters "from scratch", but with an initialization we hope will perform well – the "winning lottery ticket".

Results are summarized in Table 5. We indeed find a large gap in performance between the good initialization and randomly-initialized models with the same number of parameters.

| Auction | Min | Max | # at Initialization | # Deterministic |
|---|---|---|---|---|
| Lottery spherical | 8 | 15 | 2048 | 20 |
| Deterministic spherical | 6 | 9 | 20 | 20 |
| 2x2 U[0,1] | 7 | 10 | 4096 | 81 |
| 3x10 U[0,1] | 58 | 64 | 4096 | $2^{20}$ |

Table 4: The number of allocations actually used after 9000 steps of training, for the experiments given above. These quantities are smaller than the number of initial outcomes as well as the number of possible deterministic outcomes.

| | Mean Rev. | Best Rev. |
|---|---|---|
| Winning Ticket (2x2) | 0.870 | 0.872 |
| Small Random (2x2) | 0.772 | 0.777 |
| Winning Ticket (Spherical) | 1.836 | 1.842 |
| Small Random (Spherical) | 1.197 | 1.572 |

Table 5: We take the actually-used allocations from the best-performing 2x2 uniform and spherical models – the values of these parameters *before training* are the "winning ticket". We initialize a lottery AMA using the winning ticket initializations, and train on 4 random data seeds. To compare, we also test 4 different random initializations of the same small number of allocations, and find significantly lower performance.

## 7. Discussion

We see our approach as a first step towards strategyproof architectures for multi-agent differentiable economics. On the one hand, it is a natural generalization of RochetNet and MenuNet. On the other hand, it is also a natural generalization of classic work on AMAs.

Beyond the obvious advantage of perfect strategyproofness, there are other reasons one might prefer this approach over RegretNet. In particular, AMAs are interpretable – it's easy to simply inspect which allocations are being offered as possibilities. However, it's unclear when and whether our approach can actually represent the true optimal mechanism – that remains an open theory question. Regardless, we see it as a useful tool for automated mechanism design in multi-bidder multi-item settings.

### 7.1. Lottery ticket hypothesis

Our networks are quite sensitive to initialization – there's a relatively wide range of performance between reinitialized instances of the same architecture shown the same sequence of training data. Moreover, we found that starting out with a large number of parameters improves performance, even though by the end of training only a tiny number of these parameters were actually used.

A dependence on initialization, a benefit from overparameterization, and a final model which is effectively sparse all bring to mind the lottery ticket hypothesis (Frankle & Carbin, 2019) in deep learning. Indeed, our experimental results in section 6.4 suggest that some version of the lottery ticket hypothesis is in play here. We also observe that Curry et al. (2020) was able to significantly distill learned auction networks without harming performance. Future work in auction learning might further take advantage of this direction.

### 7.2. Strengths and Limitations

**Limitations compared to other differentiable economics approaches** The most obvious limitation of our work is that in most settings, there are probably strategyproof non-AMAs which outperform the best AMA, but we cannot learn these. This is most likely why ALGNet and RegretNet outperform our mechanisms in terms of revenue. On the other hand, we think that ensuring perfect strategyproofness is a real advantage, and a more flexible neural architecture that preserves this property while going beyond AMAs remains out of reach.

**Limitations compared to other AMA/AMD approaches** Beyond this, our approach works for settings with additive and unit-demand valuations, where the bidders' valuations are just vectors so that the total value of a bundle can be expressed as an inner product. It is not straightforwardly well-suited to complex combinatorial valuations where bidders may value bundles very differently depending on the presence of specific items.

Moreover, we solve the winner determination problem by explicitly computing the (weighted, boosted) welfare for every possible allocation. This works because at any given time, our auctions are only offering a restricted set of learned allocations, which is usually much smaller than the overall set of possible allocations. But in a setting where it was important to offer something closer to the full set of allocations, our approach would quickly become intractable.

**Strength: interpretability** A major advantage of our approach is in interpretability. Neural-network-based approaches are almost totally opaque – there's no way to summarize or explain the learned mechanism to the bidders, other than simply giving them the network parameters. By contrast, in our approach, we can simply describe the possible allocations, their boosts, and each bidder's weights, and it is immediately clear how the mechanism works.

## 8. Future Research

We focus on auctions because there is a large body of techniques in automated mechanism design and differentiable economics which provide useful baselines for performance. But VCG-style mechanisms can be used for mechanism design problems beyond auctions, or for more complex types of auctions than considered here.

We expect that the approach described here could be extended to other mechanism design problems as long as 1) feasible mechanism outcomes can be parameterized in a way amenable to gradient-based learning, and 2) the welfare of an outcome as a function of agent types can be computed in a way that preserves differentiability. Exploring the use of learned AMAs in new mechanism design settings is a fruitful direction for future work.

## 9. Acknowledgements

Curry and Dickerson were supported in part by NSF CAREER Award IIS-1846237, NSF D-ISN Award #2039862, NSF Award CCF-1852352, NIH R01 Award NLM-013039-01, NIST MSE Award #20126334, DARPA GARD #HR00112020007, DoD WHS Award #HQ003420F0035, ARPA-E Award #4334192, and a Google Faculty Research Award. Sandholm was supported by the National Science Foundation under grants IIS-1901403 and CCF-1733556. We thank Ping-yeh Chiang, Jonas Geiping, Uro Lyi, David Miller, and Neehar Peri for helpful comments and guidance on early versions of this work.

## References

Armstrong, M. Optimal multi-object auctions. *Review of Economic Studies*, 67:455–481, 2000.

Avery, C. and Hendershott, T. Bundling and optimal auctions of multiple products. *Review of Economic Studies*, 67:483–497, 2000.

Bachrach, Y., Gemp, I., Garnelo, M., Kramar, J., Eccles, T., Rosenbaum, D., and Graepel, T. A neural network auction for group decision making over a continuous space. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Balcan, M.-F., Sandholm, T., and Vitercik, E. A general theory of sample complexity for multi-item profit maximization. In *Economics and Computation (EC)*, 2018.

Balcan, M.-F., DeBlasio, D., Dick, T., Kingsford, C., Sandholm, T., and Vitercik, E. How much data is sufficient to learn high-performing algorithms? Generalization guarantees for data-driven algorithm design. In *ACM Symposium on Theory of Computing (STOC)*, 2021.

Balcan, M.-F. F., Sandholm, T., and Vitercik, E. Sample complexity of automated mechanism design. In *Neural Information Processing Systems (NeurIPS)*, 2016.

Bichler, M., Fichtl, M., Heidekrüger, S., Kohring, N., and Sutterer, P. Learning equilibria in symmetric auction games using artificial neural networks. *Nature Machine Intelligence*, 3(8):687–695, 2021.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Brero, G., Lahaie, S., and Seuken, S. Fast iterative combinatorial auctions via bayesian learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019a.

Brero, G., Lubin, B., and Seuken, S. Machine learning-powered iterative combinatorial auctions. *arXiv preprint arXiv:1911.08042*, 2019b.

Brero, G., Eden, A., Gerstgrasser, M., Parkes, D., and Rheingans-Yoo, D. Reinforcement learning of sequential price mechanisms. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Briest, P., Chawla, S., Kleinberg, R., and Weinberg, S. M. Pricing randomized allocations. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010.

Cai, Y., Daskalakis, C., and Weinberg, S. M. Optimal multi-dimensional mechanism design: Reducing revenue to welfare maximization. In *Symposium on Foundations of Computer Science (FOCS)*, 2012a.

Cai, Y., Daskalakis, C., and Weinberg, S. M. An algorithmic characterization of multi-dimensional mechanisms. In *ACM Symposium on Theory of Computing (STOC)*, 2012b.

Cai, Y., Daskalakis, C., and Weinberg, S. M. Understanding incentives: Mechanism design becomes algorithm design. In *Symposium on Foundations of Computer Science (FOCS)*, 2013.

Clarke, E. H. Multipart pricing of public goods. *Public Choice*, pp. 17–33, 1971.

Conitzer, V. and Sandholm, T. Complexity of mechanism design. In *Uncertainty in Artificial Intelligence (UAI)*, 2002.

Conitzer, V. and Sandholm, T. Incremental mechanism design. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

Curry, M., Chiang, P.-y., Goldstein, T., and Dickerson, J. P. Certifying strategyproof auction networks. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Curry, M. J., Lyi, U., Goldstein, T., and Dickerson, J. Learning revenue-maximizing auctions with differentiable matching. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

Daskalakis, C., Deckelbaum, A., and Tzamos, C. Strong duality for a multiple-good monopolist. *Econometrica*, 85(3):735–767, 2017.

Deng, Y., Mao, J., Mirrokni, V., and Zuo, S. Towards efficient auctions in an auto-bidding world. In *The Web Conference*, 2021.

Duetting, P., Feng, Z., Narasimhan, H., Parkes, D. C., and Ravindranath, S. S. Optimal auctions through deep learning. In *International Conference on Machine Learning (ICML)*, 2019.

Feng, Z., Narasimhan, H., and Parkes, D. C. Deep learning for revenue-optimal auctions with budgets. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2018.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Golowich, N., Narasimhan, H., and Parkes, D. C. Deep learning for multi-facility location mechanism design. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

Groves, T. Incentives in teams. *Econometrica: Journal of the Econometric Society*, pp. 617–631, 1973.

Guo, M., Hata, H., and Babar, A. Optimizing affine maximizer auctions via linear programming: an application to revenue maximizing mechanism design for zero-day exploits markets. In *International Conference on Principles and Practice of Multi-Agent Systems*, 2017.

Hart, S. and Nisan, N. Selling multiple correlated goods: Revenue maximization and menu-size complexity. *Journal of Economic Theory*, 183:991–1029, 2019.

Hennigan, T., Cai, T., Norman, T., and Babuschkin, I. Haiku: Sonnet for JAX, 2020. URL http://github.com/deepmind/dm-haiku.

Hessel, M., Budden, D., Viola, F., Rosca, M., Sezener, E., and Hennigan, T. Optax: composable gradient transformation and optimisation, in jax!, 2020. URL http://github.com/deepmind/optax.

Kash, I. A. and Frongillo, R. M. Optimal auctions with restricted allocations. In *Economics and Computation (EC)*, 2016.

Kuo, K., Ostuni, A., Horishny, E., Curry, M. J., Dooley, S., yeh Chiang, P., Goldstein, T., and Dickerson, J. P. ProportionNet: Balancing fairness and revenue for auction design with deep learning. *arXiv preprint arXiv:2010.06398*, 2020.

Lavi, R., Mu'alem, A., and Nisan, N. Towards a characterization of truthful combinatorial auctions. In *Symposium on Foundations of Computer Science (FOCS)*, October 2003.

Likhodedov, A. and Sandholm, T. Methods for boosting revenue in combinatorial auctions. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2004.

Likhodedov, A. and Sandholm, T. Approximating revenue-maximizing combinatorial auctions. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2005.

Maskin, E. and Riley, J. Optimal multi-unit auctions. In Hahn, F. (ed.), *The Economics of Missing Markets, Information, and Games*, chapter 14, pp. 312–335. Clarendon Press, Oxford, 1989.

Myerson, R. B. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

Pavlov, G. Optimal mechanism for selling two goods. *The BE Journal of Theoretical Economics*, 11(1), 2011.

Peri, N., Curry, M. J., Dooley, S., and Dickerson, J. P. Preferencenet: Encoding human preferences in auction design with deep learning. In *Neural Information Processing Systems (NeurIPS)*, 2021.

Rahme, J., Jelassi, S., Bruna, J., and Weinberg, S. M. A permutation-equivariant neural network architecture for auction design. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021a.

Rahme, J., Jelassi, S., and Weinberg, S. M. Auction learning as a two-player game. In *International Conference on Learning Representations (ICLR)*, 2021b.

Ravindranath, S. S., Feng, Z., Li, S., Ma, J., Kominers, S. D., and Parkes, D. C. Deep learning for two-sided matching. *arXiv preprint arXiv:2107.03427*, 2021.

Roberts, K. The characterization of implementable choice rules. *Aggregation and Revelation of Preferences*, 12(2): 321–348, 1979.

Rochet, J.-C. A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of Mathematical Economics*, 16(2):191–200, 1987.

Sandholm, T. Automated mechanism design: A new application area for search algorithms. In *International Conference on Principles and Practice of Constraint Programming*, 2003.

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

Shen, W., Tang, P., and Zuo, S. Automated mechanism design via neural networks. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2019.

Tacchetti, A., Strouse, D., Garnelo, M., Graepel, T., and Bachrach, Y. A neural architecture for designing truthful and efficient auctions. *arXiv preprint arXiv:1907.05181*, 2019.

Tang, P. and Sandholm, T. Mixed-bundling auctions with reserve prices. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2012.

Vickrey, W. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961.

Weissteiner, J. and Seuken, S. Deep learning—powered iterative combinatorial auctions. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Yao, A. C.-C. Dominant-strategy versus bayesian multi-item auctions: Maximum revenue determination and comparison. In *Economics and Computation (EC)*, 2017.