Differentiable Economics for Randomized Affine Maximizer Auctions

Michael Curry[1] , **Tuomas Sandholm**[2] and **John Dickerson**[3]
[1] University of Zurich, ETH AI Center
[2] Carnegie Mellon University, Optimized Markets, Inc., Strategic Machine, Inc., Strategy Robot, Inc.
[3] University of Maryland
curry@ifi.uzh.ch, sandholm@cs.cmu.edu, johnd@umd.edu

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.

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 and 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 and Hendershott, 2000]. There is some theory of strong duality [Daskalakis *et al.*, 2017; Kash and Frongillo, 2016] for selling multiple items to a single agent. In some such cases it may be advantageous to offer *lotteries* – i.e. award certain items to participants with some fractional probability. There have also been some successes for the weaker notion of Bayesian incentive compatibility [Cai *et al.*, 2012a; Cai *et al.*, 2012b; Cai *et al.*, 2013; Kolesnikov *et al.*, 2022]. 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 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 and Sandholm [2002; 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 and 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 and Sandholm, 2004; Likhodedov and Sandholm, 2005; Sandholm and 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 and Sandholm, 2007].

A recent form of incremental mechanism design that capitalizes on the modern power of deep learning is called *differ-*

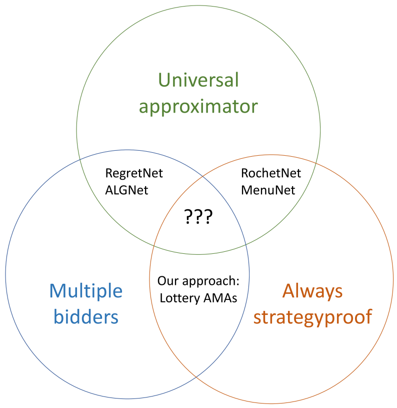

Figure 1: Our architecture in relation to other techniques from differentiable economics for multi-item revenue-maximizing auction design. The "holy grail" in the middle of the Venn diagram—that is, techniques that can represent (i) any auction for (ii) general numbers of bidders and items while (iii) 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.

entiable economics. Dütting et al. [2019] introduced the use of deep neural networks as function approximators to learn auctions. Their RegretNet architecture directly learns approximately strategyproof allocation and payment rules 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 (Figure 1). 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 and Sandholm, 2004; Likhodedov and Sandholm, 2005; Sandholm and 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; Balcan *et al.*, 2018; Balcan *et al.*, 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 strategyproof 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

Differentiable economics Dütting 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 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.

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.

Strategyproof architectures Alongside RegretNet, Dütting 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 and Frongillo, 2016]. The authors of Dütting et al. [2019] and [Shen *et al.*, 2019] apply these results to their learned auctions, and discover some previously-unknown optimal auctions.

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 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; Ivanov *et al.*, 2022; Duan *et al.*, 2022], 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 outcomes [Tacchetti *et al.*, 2019; Weissteiner and Seuken, 2020; Weissteiner *et al.*, 2020; Brero *et al.*, 2019b; Brero *et al.*, 2019a; Bachrach *et al.*, 2021; Soumalias *et al.*, 2022].

Automated mechanism design and learning theory for auctions Affine maximizer auctions (AMAs) are classic tools for automated mechanism design [Sandholm and Likhodedov, 2015; Likhodedov and Sandholm, 2004; Likhodedov and Sandholm, 2005]. In essence, AMAs are 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.

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

Previous work considers the problem of learning high-performing AMAs from samples using gradient based methods [Sandholm and Likhodedov, 2015; Likhodedov and 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; Balcan *et al.*, 2018; Balcan *et al.*, 2021]. Tang and 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.

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 and 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.

Approximation guarantees for simple mechanisms There has been an immense amount of work in the area of "simple vs. optimal" mechanisms [Hartline and Roughgarden, 2009]. A characteristic result is that of Yao [2014], which shows that when selling multiple items, choosing the better of selling items separately via a Myerson auction, and running a VCG mechanism with optimal per-bidder entry fees, gives a 69-approximation of the optimal BIC revenue. This result was later improved by Cai et al. [2016] to an 8-approximation for additive buyers. (This is a vast area of research with many results, e.g [Chawla *et al.*, 2010; Cai and Zhao, 2017], and we cannot do it justice here, but Cai et al. [2016] also gives a thorough survey of known results.)

This line of work is somewhat distinct from automated mechanism design. While the approximation ratios do apply to the settings we consider, they are too loose to be meaningful for our purposes. Overall, the focus of automated mechanism design is on computing concrete mechanisms for specific valuation distributions, rather than finding worst-case approximation ratios.

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.

This naturally raises the question of to what extent an AMA (including our learned AMAs) can be guaranteed to approximate the optimal revenue. To the best of our knowledge, there has been little work on this interesting question despite decades of study of affine maximizer mechanisms. We see it as complementary to, but outside the scope of, our work, which focuses on providing concrete techniques to compute good auctions for specific problem instances.

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. We assume valuations are additive over items – meaning that the value of a bundle containing multiple items is the sum of each individual item's value. We also allow for unit-demand – bidders only want at most 1 item – which we model as additive valuations, with an additional constraint that no allocation will give a bidder more than 1 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 $\sum_i a_{ij} \leq 1$, 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: $\sum_j a_{ij} \leq 1$. 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 the baselines for comparison 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. For a more detailed derivation, see the supplemental material. 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. Having done this, we have a set of menu items/allocations, and the boosts for each allocation correspond to the payment charged for that menu item. Thus our approach of learning allocations and boosts by gradient descent directly generalizes RochetNet [Duetting *et al.*, 2019] and MenuNet [Shen *et al.*, 2019] (with additive utilities). For more details, see the supplemental material.

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.) As is typical in automated mechanism design, 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, $\arg\max_k f(a_k) \approx \langle \mathrm{softmax}_\tau(f(a_1), \cdots, f(a_K)), \boldsymbol{a} \rangle$ and $\max_k f(a_k) \approx \langle \mathrm{softmax}_\tau(f(a_1), \cdots, f(a_K)), \boldsymbol{f}(\boldsymbol{a}) \rangle$. As the softmax temperature parameter τ 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 Dütting et al. [2019] , for additive allocations, each allocation is represented by 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. (For diagrams and pseudocode for training, please see the supplemental material.)

6 Results

6.1 Hyperparameters and Training

For lottery AMAs, we allow 2048 or 4096 allocations. The inverse of the softmax temperature parameter 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, on cluster nodes with 2080Ti GPUs. 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. 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 and Likhodedov, 2015]. Following other automated mechanism design work, we compare to theoretical revenues from Myerson auctions of separate items and of the grand bundle. We also include the lowest-regret version of the RegretFormer approach of Ivanov et al. [2022]. (Other versions trade off slightly relaxed enforcement of regret for even higher revenue than ALGNet and RegretNet.) Since we only use previously reported results, please refer to the previous papers for training details of the baselines.

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].

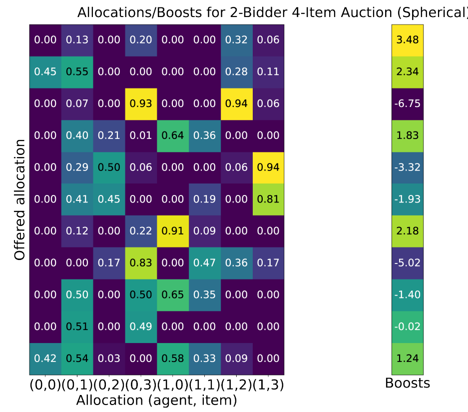

Figure 2: The ten lottery allocations (and their boosts) actually used after training an auction. (The auction has many more parameters, but the 2038 other allocations are never chosen for any of the sampled bids.) One can see that the mechanism does typically offer lotteries.

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 2 shows the final offered allocations and boosts from a representative mechanism – the auction is actually taking advantage of randomization.

2-bidder, 2-item uniform setting 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 and Likhodedov, 2015; Tang and Sandholm, 2012] as well as the RegretNet neural network approach (which performs better but is not perfectly strategyproof) [Duetting *et al.*, 2019].

We outperform the lowest-regret variant of the RegretFormer [Ivanov *et al.*, 2022], while also having 0 regret. 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 3.

Another interesting observation has to do with similarities to the MBARP auction of [Tang and Sandholm, 2012]. When selling both items, boosts are higher if they go to the same agent. They are higher still for withholding one item and highest for withholding both. This is the same basic pattern as with the MBARP. It is plausible but not certain that this mechanism approaches the optimal MBARP.

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. The worst lottery mechanism outperforms the best deterministic mechanism. Moreover, the lottery mechanisms do actually learn to randomize.

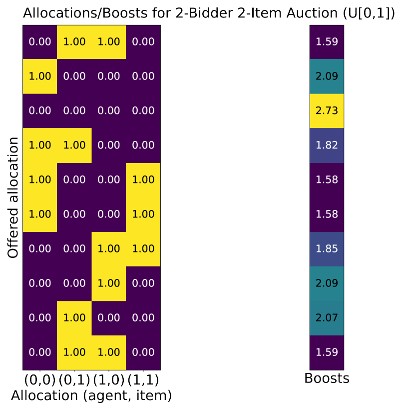

Figure 3: All allocations actually used after training for a 2x2 U[0,1] additive auction (the same setting as compared to in [Likhodedov and Sandholm, 2004] and Dütting et al. [2019]). Here, although the mechanism space is that of randomized mechanisms, the algorithm learns to offer deterministic allocations. The revenue is comparable to results in [Likhodedov and Sandholm, 2004].

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
RegretFormer (low regret budget)	0.861	0.00006

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 and Likhodedov [2015]. MBARP is a subset of AMA from Tang and 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. We manage to outperform the lowest-regret RegretFormer from Ivanov et al. [2022]. Note that Sandholm and Likhodedov [2015] present many variants, some of which beat our revenue, although all are comparable.

3-bidder, 10-item uniform setting Finally, we consider one of the much larger auction settings from Dütting et al. [2019] – 3 additive bidders with 10 $U[0,1]$ items. This is a setting for which previous AMA approaches cannot be applied as-is. 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 related approaches, 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 others may be due to non-zero regret. 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.

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.

Effects of parameter initialization Motivated by the *lottery ticket hypothesis* in neural network training [Frankle and 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.

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
RegretFormer (low regret budget)	5.745	0.00022

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 and others (although these approaches have a small strategyproofness violation), but outperforms the separate Myerson and grand bundling baselines.

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

Auction	Min	Max	# Init.	# Deterministic
Lottery spherical	8	15	2048	20
Deterministic spherical	6	9	20	20
2x2 U[0,1]	7	10	4096	9
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. For all but the smallest setting, 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. We also test 4 different random initializations of the same small number of allocations, and find lower performance.

in performance between the good initialization and randomly-initialized models with the same number of parameters.

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/item settings.

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 and Carbin, 2019] in deep learning. Indeed, our experimental results in section 6.2 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.

8 Future Research

We focused on auctions because there is a large body of techniques in automated mechanism design and differentiable economics which provide useful baselines for performance. We expect that the approach described here could be extended to other mechanism design problems as long as 1) VCG-style mechanisms can be used, 2) feasible mechanism outcomes can be parameterized in a way amenable to gradient-based learning, and 3) 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.

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 NSF grants IIS-1901403 and CCF-1733556 and ARO grant W911NF-22-1-0266. Curry also received funding from the European Research Council (ERC) under the EU's Horizon 2020 research and innovation programme (Grant agreement No. 805542). We thank Ping-yeh Chiang, Jonas Geiping, Uro Lyi, David Miller, Neehar Peri, and the anonymous reviewers.

References

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

[Avery and Hendershott, 2000] Christopher Avery and Terrence Hendershott. Bundling and optimal auctions of multiple products. *Review of Economic Studies*, 67:483–497, 2000.

[Bachrach *et al.*, 2021] Yoram Bachrach, Ian Gemp, Marta Garnelo, Janos Kramar, Tom Eccles, Dan Rosenbaum, and Thore Graepel. A neural network auction for group decision making over a continuous space. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[Balcan *et al.*, 2016] Maria-Florina F Balcan, Tuomas Sandholm, and Ellen Vitercik. Sample complexity of automated mechanism design. In *NeurIPS*, 2016.

[Balcan *et al.*, 2018] Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. A general theory of sample complexity for multi-item profit maximization. In *Economics and Computation (EC)*, 2018.

[Balcan *et al.*, 2021] Maria-Florina Balcan, Dan DeBlasio, Travis Dick, Carl Kingsford, Tuomas Sandholm, and Ellen Vitercik. 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.

[Bradbury *et al.*, 2018] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal

Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[Brero *et al.*, 2019a] Gianluca Brero, Sébastien Lahaie, and Sven Seuken. Fast iterative combinatorial auctions via bayesian learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[Brero *et al.*, 2019b] Gianluca Brero, Benjamin Lubin, and Sven Seuken. Machine learning-powered iterative combinatorial auctions. *arXiv preprint arXiv:1911.08042*, 2019.

[Brero *et al.*, 2021] Gianluca Brero, Alon Eden, Matthias Gerstgrasser, David Parkes, and Duncan Rheingans-Yoo. Reinforcement learning of sequential price mechanisms. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[Briest *et al.*, 2010] Patrick Briest, Shuchi Chawla, Robert Kleinberg, and S Matthew Weinberg. Pricing randomized allocations. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010.

[Cai and Zhao, 2017] Yang Cai and Mingfei Zhao. Simple mechanisms for subadditive buyers via duality. In *ACM Symposium on Theory of Computing (STOC)*, 2017.

[Cai *et al.*, 2012a] Yang Cai, Constantinos Daskalakis, and S Matthew Weinberg. An algorithmic characterization of multi-dimensional mechanisms. In *ACM Symposium on Theory of Computing (STOC)*, 2012.

[Cai *et al.*, 2012b] Yang Cai, Constantinos Daskalakis, and S Matthew Weinberg. Optimal multi-dimensional mechanism design: Reducing revenue to welfare maximization. In *Symposium on Foundations of Computer Science (FOCS)*, 2012.

[Cai *et al.*, 2013] Yang Cai, Constantinos Daskalakis, and S Matthew Weinberg. Understanding incentives: Mechanism design becomes algorithm design. In *Symposium on Foundations of Computer Science (FOCS)*, 2013.

[Cai *et al.*, 2016] Yang Cai, Nikhil R. Devanur, and S. Matthew Weinberg. A Duality-Based Unified Approach to Bayesian Mechanism Design. In *ACM Symposium on Theory of Computing (STOC)*, 2016.

[Chawla *et al.*, 2010] Shuchi Chawla, Jason D Hartline, David L Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *ACM Symposium on Theory of Computing (STOC)*, 2010.

[Clarke, 1971] Edward H Clarke. Multipart pricing of public goods. *Public Choice*, pages 17–33, 1971.

[Conitzer and Sandholm, 2002] Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. In *Uncertainty in Artificial Intelligence (UAI)*, 2002.

[Conitzer and Sandholm, 2007] Vincent Conitzer and Tuomas Sandholm. Incremental mechanism design. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[Curry *et al.*, 2020] Michael Curry, Ping-yeh Chiang, Tom Goldstein, and John P. Dickerson. Certifying strategyproof auction networks. In *NeurIPS*, 2020.

[Curry *et al.*, 2021] Michael J Curry, Uro Lyi, Tom Goldstein, and John Dickerson. Learning revenue-maximizing auctions with differentiable matching. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

[Daskalakis *et al.*, 2017] Constantinos Daskalakis, Alan Deckelbaum, and Christos Tzamos. Strong duality for a multiple-good monopolist. *Econometrica*, 85(3):735–767, 2017.

[Deng *et al.*, 2021] Yuan Deng, Jieming Mao, Vahab Mirrokni, and Song Zuo. Towards efficient auctions in an auto-bidding world. In *The Web Conference*, 2021.

[Duan *et al.*, 2022] Zhijian Duan, Jingwu Tang, Yutong Yin, Zhe Feng, Xiang Yan, Manzil Zaheer, and Xiaotie Deng. A context-integrated transformer-based neural network for auction design. *arXiv preprint arXiv:2201.12489*, 2022.

[Duetting *et al.*, 2019] Paul Duetting, Zhe Feng, Harikrishna Narasimhan, David C. Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning. In *ICML*, 2019.

[Feng *et al.*, 2018] Zhe Feng, Harikrishna Narasimhan, and David C. Parkes. Deep learning for revenue-optimal auctions with budgets. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2018.

[Frankle and Carbin, 2019] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR*, 2019.

[Golowich *et al.*, 2018] Noah Golowich, Harikrishna Narasimhan, and David C. Parkes. Deep learning for multi-facility location mechanism design. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[Groves, 1973] Theodore Groves. Incentives in teams. *Econometrica: Journal of the Econometric Society*, pages 617–631, 1973.

[Guo *et al.*, 2017] Mingyu Guo, Hideaki Hata, and Ali Babar. 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 and Nisan, 2019] Sergiu Hart and Noam Nisan. Selling multiple correlated goods: Revenue maximization and menu-size complexity. *Journal of Economic Theory*, 183:991–1029, 2019.

[Hartline and Roughgarden, 2009] Jason D Hartline and Tim Roughgarden. Simple versus optimal mechanisms. In *Economics and Computation (EC)*, 2009.

[Hennigan *et al.*, 2020] Tom Hennigan, Trevor Cai, Tamara Norman, and Igor Babuschkin. Haiku: Sonnet for JAX, 2020.

[Hessel *et al.*, 2020] Matteo Hessel, David Budden, Fabio Viola, Mihaela Rosca, Eren Sezener, and Tom Hennigan. Optax: composable gradient transformation and optimisation, in jax!, 2020.

[Ivanov *et al.*, 2022] Dmitry Ivanov, Iskander Safiulin, Ksenia Balabaeva, and Igor Filippov. Optimal-er auctions through attention. In *NeurIPS*, 2022.

[Kash and Frongillo, 2016] Ian A Kash and Rafael M Frongillo. Optimal auctions with restricted allocations. In *Economics and Computation (EC)*, 2016.

[Kolesnikov *et al.*, 2022] Alexander V. Kolesnikov, Fedor Sandomirskiy, Aleh Tsyvinski, and Alexander P. Zimin. Beckmann's approach to multi-item multi-bidder auctions. (arXiv:2203.06837), September 2022.

[Kuo *et al.*, 2020] Kevin Kuo, Anthony Ostuni, Elizabeth Horishny, Michael J. Curry, Samuel Dooley, Ping yeh Chiang, Tom Goldstein, and John P. Dickerson. ProportionNet: Balancing fairness and revenue for auction design with deep learning. *arXiv preprint arXiv:2010.06398*, 2020.

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

[Likhodedov and Sandholm, 2004] Anton Likhodedov and Tuomas Sandholm. Methods for boosting revenue in combinatorial auctions. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2004.

[Likhodedov and Sandholm, 2005] Anton Likhodedov and Tuomas Sandholm. Approximating revenue-maximizing combinatorial auctions. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2005.

[Maskin and Riley, 1989] Eric Maskin and John Riley. Optimal multi-unit auctions. In Frank Hahn, editor, *The Economics of Missing Markets, Information, and Games*, chapter 14, pages 312–335. Clarendon Press, Oxford, 1989.

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

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

[Peri *et al.*, 2021] Neehar Peri, Michael J Curry, Samuel Dooley, and John P Dickerson. Preferencenet: Encoding human preferences in auction design with deep learning. In *NeurIPS*, 2021.

[Rahme *et al.*, 2021a] Jad Rahme, Samy Jelassi, Joan Bruna, and S Matthew Weinberg. A permutation-equivariant neural network architecture for auction design. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[Rahme *et al.*, 2021b] Jad Rahme, Samy Jelassi, and S Matthew Weinberg. Auction learning as a two-player game. In *ICLR*, 2021.

[Ravindranath *et al.*, 2021] Sai Srivatsa Ravindranath, Zhe Feng, Shira Li, Jonathan Ma, Scott D Kominers, and David C Parkes. Deep learning for two-sided matching. *arXiv preprint arXiv:2107.03427*, 2021.

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

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

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

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

[Shen *et al.*, 2019] Weiran Shen, Pingzhong Tang, and Song Zuo. Automated mechanism design via neural networks. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2019.

[Soumalias *et al.*, 2022] Ermis Soumalias, Behnoosh Zamanlooy, Jakob Weissteiner, and Sven Seuken. Machine learning-powered course allocation. *arXiv preprint arXiv:2210.00954*, 2022.

[Tacchetti *et al.*, 2019] Andrea Tacchetti, DJ Strouse, Marta Garnelo, Thore Graepel, and Yoram Bachrach. A neural architecture for designing truthful and efficient auctions. *arXiv preprint arXiv:1907.05181*, 2019.

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

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

[Weissteiner and Seuken, 2020] Jakob Weissteiner and Sven Seuken. Deep learning—powered iterative combinatorial auctions. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[Weissteiner *et al.*, 2020] Jakob Weissteiner, Chris Wendler, Sven Seuken, Ben Lubin, and Markus Püschel. Fourier analysis-based iterative combinatorial auctions. *arXiv preprint arXiv:2009.10749*, 2020.

[Yao, 2014] Andrew Chi-Chih Yao. An n-to-1 Bidder Reduction for Multi-item Auctions and its Applications. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2014.

[Yao, 2017] Andrew Chi-Chih Yao. Dominant-strategy versus bayesian multi-item auctions: Maximum revenue determination and comparison. In *Economics and Computation (EC)*, 2017.