Advancing Differentiable Economics: A Neural Network Framework for Revenue-Maximizing Combinatorial Auction Mechanisms

Mai Pham , **Vikrant Vaze** and **Peter Chin**

Thayer School of Engineering at Dartmouth

mai.p.pham.th@dartmouth.edu,

Abstract

Differentiable economics, which uses neural networks as function approximators and gradient-based optimization in automated mechanism design (AMD), marked a significant breakthrough with the introduction of RegretNet [Duetting *et al.*, 2019]. It combines the flexibility of deep learning with a regret-based approach to relax incentive compatibility, allowing for approximations of revenue-maximizing auctions. However, applying these techniques to combinatorial auctions (CAs) - where bidders value bundles rather than individual items, capturing item interdependencies - remains a challenge, primarily due to the lack of methodologies that can effectively deal with combinatorial constraints. To tackle this, we propose two architectures: CANet, a fully connected neural network, and CAFormer, a transformer-based model designed to learn optimal randomized mechanisms. Unlike existing methods in traditional AMD, our approach is more scalable and free of assumptions about the structures of allowable bundles or bidder valuations. We demonstrate that our models match current methods in non-combinatorial settings and set new benchmarks for CAs. Specifically, our models consistently outperform benchmark mechanisms derived from heuristic approaches and provide empirical solutions where analytical results are unavailable. This work bridges the gap in applying differentiable economics to combinatorial auctions, offering a scalable and flexible framework for designing revenue-maximizing mechanisms.

1 Introduction

Mechanism design, a field of economics, focuses on creating incentives and interaction rules among self-interested agents to achieve specific objectives for the group. It differs from traditional game theory by starting with a desired outcome and designing the rules so that agents acting in their self-interest naturally lead to that outcome. A key application of mechanism design is to create auction rules, which play a significant role in economic activities such as the fine art market, adver-

tising on search engines or e-commerce platforms [Bajari and Hortaçsu, 2003; Edelman *et al.*, 2007].

The Vickrey-Clarke-Groves (VCG) mechanism is optimal and strategy-proof when the goal is to maximize total bidder welfare [Vickrey, 1961; Clarke, 1971; Groves, 1973]. However, revenue maximization poses a greater challenge. Myerson [1981] resolved the revenue-maximizing strategy-proof auction for a single item, later extended by Maskin *et al.* [1989] to multiple copies of a single item. The general revenue-maximizing auction for multiple items remains unsolved decades later, with only specific two-item cases addressed [Armstrong, 2000; Avery and Hendershott, 2000].

The lack of theoretical progress in revenue-maximizing multi-item auctions inspired the development of Automated Mechanism Design (AMD) [Sandholm, 2003]. AMD uses computational methods to tailor mechanisms to specific problem instances, with the use of heuristic approaches [Sandholm, 2002; Sandholm and Likhodedov, 2015], but suffers from the curse of dimensionality in large-scale settings. More recently, leveraging the power of deep learning, differentiable economics uses deep neural networks as function approximators to learn optimal auctions. This approach, introduced with RegretNet [Duetting *et al.*, 2019], offers a scalable alternative to traditional AMD methods.

Since its introduction, RegretNet has inspired a variety of extensions and improvements [Rahme *et al.*, 2021b,a; Ivanov *et al.*, 2022]. The complexity of multi-item auctions is further amplified in combinatorial auctions, where bidders express valuations for bundles rather than individual items, capturing complementarities or substitutabilities. Unlike traditional auctions, the interdependencies between items in combinatorial auctions introduce additional challenges. One such challenge is winner determination, a problem that is well known to be NP-hard, which involves allocating bundles to agents while ensuring that none of the bundles overlap. Although the later part of the RegretNet study discusses the 2-item, 2-bidder auction in a combinatorial setting [Duetting *et al.*, 2019], the problem is still wide open, primarily due to the lack of methodologies to effectively constrain the allocation space to satisfy combinatorial feasibility.

This paper builds on recent advances in auction theory and machine learning to design combinatorial revenue-optimal auctions. It makes the following contributions:

1. It provides an empirical method to find **near-optimal**

solutions to revenue-maximizing CA, addressing problems where analytical solutions are not available. This method is not limited by assumptions about allowable bundle structures or bidder valuations. Similar to the previous work in differentiable economics [Duetting *et al.*, 2019], the method is approximately strategy-proof.

2. Unlike previous CA studies, [Sandholm and Likhodedov, 2015; Tacchetti *et al.*, 2022; Jamshidi *et al.*, 2024], our models identify **randomized (lottery) mechanisms**, extending the RegretNet family to CA problems. It is well known that randomization can increase revenue in CAs. Our empirical experiments show substantially higher performance than deterministic mechanisms.

3. It introduces two architectures: **CANet** and **CAFormer**, extending prior designs from non-combinatorial auctions to a CA setting. While CANet is sensitive to the order and structure of the input, CAFormer is permutation-equivariant, offering better generalization in scenarios with varying input configurations. In addition, it implements several techniques to improve stability in training and tackle the vanishing gradient problem.

4. It provides an example of successful use of **gradient based methods** for **constrained optimization problems**, which are known to be challenging. In our particular case, the constraints are related to combinatorial feasibility. This approach can be extended to other differentiable combinatorial optimization problems.

Section 2 introduces the problem statement. Section 3 presents the proposed method. Section 4 provides the results and Section 5 concludes with a discussion and the next steps.

2 Combinatorial Auction Design

2.1 Optimal CA as a linear program

General auction setting We study a setting with one seller, a set N of n bidders, and a set M of m items for sale. In an auction, bidders submit bids for the bundles of items given the auction rule. An auction rule is a pair: an allocation rule $g(.)$ and a payment rule $p(.)$, where g_i is the bundle of items received by the bidder i and p_i is the payment by the bidder i.

Non-additive valuations We define K as the set (indexed by S) of bundles of size $k = 2^m - 1$, which comprises *all possible non-empty combinations* of items for each bidder. The *valuation profile* $v = (v_1, v_2, \ldots, v_n)$ is private information and is unknown to the auctioneer. Let V_i denote the set of all possible valuation functions for the bidder i. In a CA, the *valuation function* for the bidder i is given by the vector $v_i = (v_{i1}, v_{i2}, \ldots, v_{ik})$, where each element specifies the value that bidder i attaches to a certain bundle $S \in K$ of items from M. Each valuation function v_i is sampled from a continuous probability density F_i, and F_i is positive on all V_i. The auction is *symmetric* if $F_i = F \ \forall i \in N$, and *asymmetric* if $F_i \neq F_j$ for $i \neq j$.

Quasi-linear utilities We make the standard assumption that each bidder i has a *quasi-linear utility function*, that is, a bidder i with valuation v_i for the bid profile b receives utility $u_i(v_i; b) = v_i(g_i(b)) - p_i(b)$, where $v_i(g)$ is bidder i's valuation for allocation g, p_i is the payment by the bidder i.

Incentive Compatibility and Regret A mechanism is a *dominant-strategy incentive compatible (DSIC)* mechanism if each bidder's utility is maximized when reporting true valuation, regardless of what other bidders report.

$$u_i(v_i; (v_i, b_{-i})) \geq u_i(v_i; (b_i, b_{-i})) \, \forall b_i \geq 0, \ \forall i \in N \quad (1)$$

To quantify the violation of DSIC, *expected ex-post regret* for each bidder i is defined by the following equation.

$$\mathrm{rgt}_i = \mathbf{E}\left[\max_{v'} u_i(v_i; (v', v_{-i})) - u_i(v_i; (v_i, v_{-i}))\right] \quad (2)$$

Individual Rationality (IR) A mechanism is *ex-post individual rational* if each bidder receives a nonnegative utility when participating truthfully.

$$u_i(v_i; (v_i, b_{-i})) \geq 0 \, \forall b_{-i}, \ \forall i \in N \quad (3)$$

Combinatorial Feasibility Let z_{iS} specify if a bundle S is allocated to bidder i ($z_{iS} = 1$) or not ($z_{iS} = 0$). The standard CA formulation requires the following constraints to hold:

$$\sum_{i \in N} \sum_{S \in K : j \in S} z_{iS} \leq 1 \quad \forall j \in M \quad (4)$$

$$\sum_{S \in K} z_{iS} \leq 1 \quad \forall i \in N \quad (5)$$

$$z_{iS} \in \{0, 1\} \quad \forall i \in N, S \in K \quad (6)$$

Constraint (4) ensures that each item j is allocated to at most one bidder, implicitly guaranteeing that no two bidders can receive overlapping bundles. Constraint (5) ensures that each bidder i is assigned at most one bundle.

To make the solution space differentiable, we relax the binary constraint (6) by the following theorem:

Theorem 1 (Birkhoff, 1916). *Every doubly stochastic matrix is a convex combination of permutation matrices.*

This ensures that a randomized allocation represented by a doubly-stochastic matrix can be decomposed into a lottery over deterministic one-to-one assignments. Note that since each bundle consists of a unique subset of items, constraint (4) implies that each bundle can be allocated at most once. So relaxing the binary constraint (6)

$$z_{iS} \in [0, 1] \quad \forall i \in N, S \in K \quad (7)$$

enables the representation of lottery decomposable randomized allocations.

We refer to the inequalities (4), (5), and (7) as *combinatorial constraints*. Any allocation satisfying these constraints is considered *combinatorially feasible*.

Revenue Maximization The primary objective of the auctioneer in a CA is to maximize revenue while respecting the constraints of the mechanism. For a given valuation profile v, the revenue for a given payment rule $p(v)$ is defined as:

$$\mathrm{rev} = \sum_{i \in N} p_i(v).$$

The revenue maximization problem in CAs can be formulated as a linear program over the allocation and payment rules. The linear program is expressed as:

$$\max_{\mathbf{g}, \mathbf{p}} \sum_{i \in N} p_i(v)$$

subject to (1), (3), (4), (5), (7) and $p_i(v) \geq 0, \ \forall i \in N$. For brevity, in the sections below, we use the same notations $u, \mathrm{rev}, \mathrm{rgt}$ for empirical quantities in the learning problem.

2.2 Auction Design as a learning problem

RegretNet [Duetting *et al.*, 2019] provides a deep learning framework for optimal auction design, addressing multi-bidder, multi-item scenarios where analytical solutions are unknown. It encodes auction allocation and payment rules within neural network weights.

The architecture comprises two networks: the allocation network $\mathbf{A}^{\text{net}}$ and the payment network $\mathbf{P}^{\text{net}}$. Both process a bid vector $\mathbf{b}^{nm}$ through fully-connected layers. The Allocation Network maps bids to allocation probabilities $\mathbf{A}^{\text{net}}(\mathbf{b}^{nm}) = \mathbf{Z}^{nm}$, where $z_{ij} = \frac{e^{\tilde{z}_{ij}}}{\sum_{i=1}^{n+1} e^{\tilde{z}_{ij}}}$ and allows an item to remain unallocated by introducing a dummy participant. The payment network maps bids to payment allocations $\mathbf{P}^{\text{net}}(\mathbf{b}^{nm}) = \tilde{\mathbf{p}}^n$, where $\tilde{p}_i$ is scaled using sigmoid, and $p_i = \tilde{p}_i \sum_{j=1}^{m} z_{ij} b_{ij}$, adhering to IR constraints (3).

RegretNet minimizes empirical negative revenue plus the regret penalty over profiles V:

$$\min_{\mathbf{w},\lambda,\rho} \left(-\sum_{i\in N} \mathbb{E}[P_i(v;\mathbf{w})] + \lambda_i \text{rgt}_i + \frac{\rho}{2}\text{rgt}_i^2 \right) \quad (8)$$

where regret $\text{rgt}_i(v_i', v; \mathbf{w}) = \max_{v_i'}(u(v_i, (v_i', v_{-i}); \mathbf{w}) - u(v_i, v; \mathbf{w}))$. Regret is iteratively minimized using augmented Lagrangian and gradient descent:

$$\mathcal{L}_{\text{inner}}(v_i') = -u(v_i, (v_i', v_{-i}); \mathbf{w}) \quad (9)$$

and the Lagrange multipliers λ_i and ρ are updated as $\lambda_i \leftarrow \lambda_i + \rho \tilde{R}_i$, $\rho \leftarrow \rho + \rho\Delta$.

The results of RegretNet are further extended to address combinatorial auctions in a two-item, two-bidder setting by adding the constraint:

$$\forall i, \quad z_{i,\{1\}} + z_{i,\{2\}} \le 1 - \sum_{i=1}^{n} z_{i,\{1,2\}}$$

To accommodate this constraint, two softmax layers are used; one outputs a set of bidder scores, and the other a set of item scores, denoted by $\bar{s}_{i,S}$ and $\bar{s}_{i,S}^{(j)}$. The set of item scores is then used to compute a normalized set of scores for all i and S, given by

$$\bar{\bar{s}}_{iS} = \begin{cases} \bar{s}_{iS}^{(i)} & S = \{1\}, \{2\} \\ \min\{\bar{s}_{iS}^{(k)}\} & S = \{1,2\} \end{cases}$$

and the final allocation $z_{i,S}$ is determined by the minimum of the bidder score and normalized item score.

$$z_{iS} = \min\left(\bar{s}_{iS}, \bar{\bar{s}}_{iS}\right)$$

2.3 Related Work

The seminal works of Vickrey [1961] introduced auction mechanisms for single items, which were later extended to multi-item settings by developing the Vickrey-Clarke-Groves (VCG) mechanism [Vickrey, 1961; Clarke, 1971; Groves, 1973]. Although VCG achieves efficiency in dominant strategies, it often fails to maximize revenue. Myerson [1981] laid the theoretical foundation for revenue-maximizing auctions.

The study of combinatorial auctions originated from the need to allocate resources among bidders with complex preferences over bundles of items. Cramton *et al.* [2005] provide a detailed review of combinatorial auctions. Optimal combinatorial auctions are NP-hard because of the exponential number of possible bundles and the Incentive Compatibility (IC) and IR constraints. Heuristic and exact methods have been studied to solve the winner determination problem as a combinatorial optimization problem [Sandholm, 2002; Wu and Hao, 2016; Ray *et al.*, 2021]. In the early work, both the winner determination and pricing problems were tackled using integer programming and approximation algorithms. Lavi and Swamy [2011] developed polynomial-time algorithms for approximately optimizing VCG-based mechanisms in certain structured settings. Another direction, extending the results of Roberts [1979], is known as Affine Maximizer Auctions (AMAs). These are variations of the VCG mechanism that adjust the allocation process by assigning positive weights to each bidder's welfare and incorporating boosts for different allocations, potentially leading to higher revenue than the VCG mechanism [Sandholm and Likhodedov, 2015; Curry *et al.*, 2023].

Recent advances in differentiable economics have extended its applications to various domains, reflecting the growing synergy between machine learning and economic theory. These works leverage neural architectures and differentiable approaches to address classical and emerging problems in auction design and beyond. Several papers have extended the ideas introduced in RegretNet. Rahme *et al.* [2021b] model optimal auction as a two-player game and proposes a loss function with fewer hyperparameters. Rahme *et al.* [2021a] and Ivanov *et al.* [2022] improve the results using different architectures. Others have applied this framework to auctions with fairness or budget constraints [Kuo *et al.*, 2020; Feng *et al.*, 2018], or adapted similar methods to tackle broader mechanism design challenges [Golowich *et al.*, 2018; Ravindranath *et al.*, 2023]. Several papers leverage machine learning-based approaches in combinatorial auctions. Brero *et al.* [2018] used a machine learning-based elicitation algorithm to identify which values to query from the bidders. Tacchetti *et al.* [2022] propose a network architecture to learn Groves payment rules in combinatorial auctions with certain bidding languages. Jamshidi *et al.* [2024] present a machine learning-powered combinatorial auction based on the principles of differential privacy. Brero *et al.* [2021] present a machine learning-powered iterative combinatorial auction. Ravindranath *et al.* [2024] use deep Reinforcement Learning for sequential combinatorial auctions.

3 Proposed Method

After formulating CAs as a learning problem, we will propose models that facilitate end-to-end learning through gradient-based optimization techniques.

3.1 Constraint Enforcement

Recall from Section 2 that the allocation must satisfy constraints (4), (5) and (7). We decompose the allocation matrix
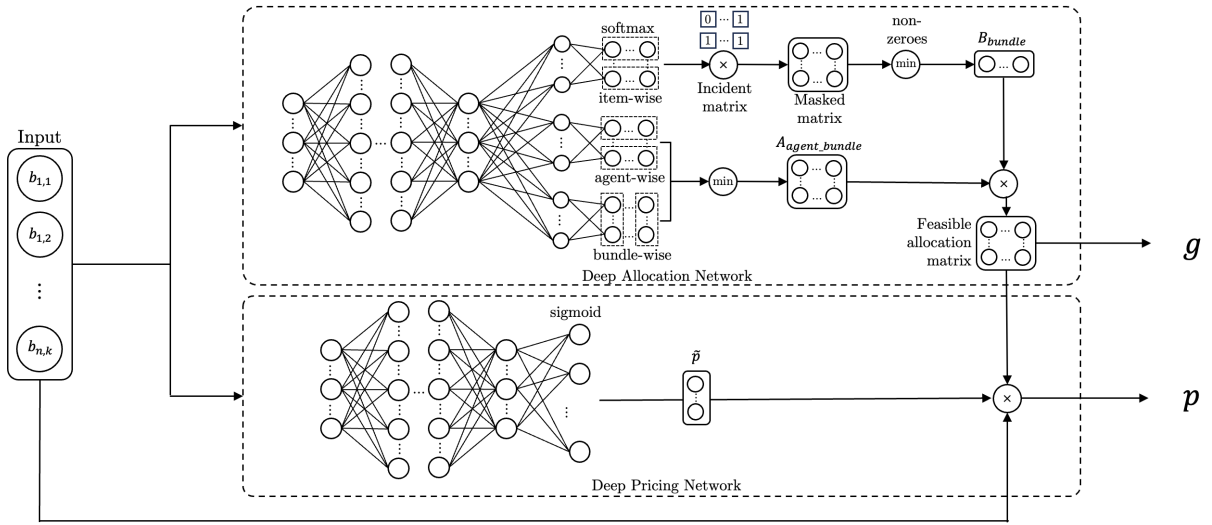
(a) The CANet architecture for n bidders and m items takes $n \times k$ bid vectors as input. It consists of two fully connected network modules: the allocation network, which outputs a feasible allocation matrix g, and the pricing network, which generates a pricing vector $\tilde{p}$.
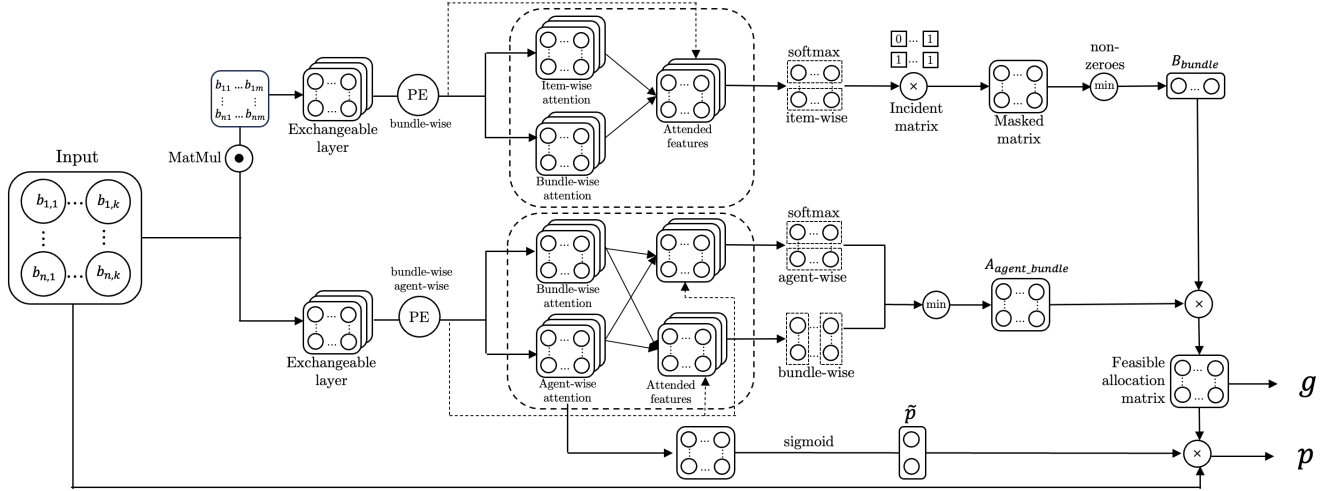
(b) The CAFormer architecture processes $n \times k$ bid matrices using attention mechanism to encode permutation-equivariant item-bundle and agent-bundle representations. It outputs a feasible allocation matrix g, with the pricing vector $\tilde{p}$ derived from the agent embedding.

Figure 1: CANet and CAFormer Architectures.

$\mathbf{Z} \in \mathbb{R}^{n \times k}$ into the product of two matrices:

$$\mathbf{Z} = \mathbf{B}^{\text{bundle}} \cdot \mathbf{A}^{\text{agent-bundle}},$$

where $\mathbf{A}^{\text{agent-bundle}}$ and $\mathbf{B}^{\text{bundle}}$ are defined below. This approach is motivated by a two-step allocation process: first, the allocation of items to the bundles, satisfying the item-wise constraint, and then the allocation of the bundles to the bidders, satisfying the other constraints. Each entry in the final allocation matrix represents the probability of allocating a bundle to a bidder, which is calculated by the product of the probability that the bundle is allocated B_S^{bundle} and the probability that the bidder receives the bundle, given that the bundle is allocated $A_{iS}^{\text{agent-bundle}}$.

The item-to-bundle allocation matrix $\mathbf{B}^{\text{bundle}}$ is derived from the output of the Allocation Network, normalized to represent item-wise probabilities. To construct $\mathbf{B}^{\text{bundle}}$, we define

the incident matrix $\mathbf{I} \in \mathbb{R}^{m \times k}$, where:

$$I_{iS} = \begin{cases} 1 & \text{if item } i \text{ is included in bundle } S, \\ 0 & \text{otherwise.} \end{cases}$$

The initial bundle allocation matrix $\mathbf{B} \in \mathbb{R}^{m \times k}$ is adjusted by mapping bundles to items with a softmax function across m items:

$$\mathbf{B}^{\text{adjusted}} = \text{softmax}_M(\mathbf{B} \cdot \mathbf{I}).$$

To ensure valid item-to-bundle allocations, non-positive values in $\mathbf{B}^{\text{adjusted}}$ are replaced with a large constant M:

$$\mathbf{B}^{\text{masked}} = \begin{cases} \mathbf{B}^{\text{adjusted}}, & \text{if } \mathbf{B}^{\text{adjusted}} > 0, \\ M, & \text{otherwise.} \end{cases}$$

For each item i, we compute the minimum value across all bundles and reshape $\mathbf{B}_{\min}$ to include the agent dimension:

$$\mathbf{B}^{\text{bundle}} = \text{Unsqueeze}(\min_S \mathbf{B}^{\text{masked}}),$$

The bundle-to-agent allocation matrix $\mathbf{A}_{\text{agent-bundle}}$ is computed from the allocation network outputs $\mathbf{A}$ and $\mathbf{A}'$ using the softmax function along the agent and the bundle dimension.

$$\mathbf{A}^{\text{agent}} = \text{softmax}_N(\mathbf{A}), \quad \mathbf{A}^{\text{bundle}} = \text{softmax}_K(\mathbf{A}')$$

then taking element-wise minimum of the two matrices

$$\mathbf{A}^{\text{agent-bundle}} = \min(\mathbf{A}^{\text{agent}}, \mathbf{A}^{\text{bundle}})$$

The final allocation matrix $\mathbf{Z}$ is given by:

$$\mathbf{Z} = \mathbf{B}^{\text{bundle}} \cdot \mathbf{A}^{\text{agent-bundle}},$$

where each entry z_{iS} represents the probability of allocating bundle S to agent i.

Lemma 1. *Matrix $\mathbf{Z}$ is combinatorially feasible.*

Proof. The matrix $\mathbf{A}^{\text{agent-bundle}}$ represents the bundle-to-agent allocation. Each entry $A_{iS}^{\text{agent-bundle}}$ is computed as:

$$A_{iS}^{\text{agent-bundle}} = \min\left(\frac{e^{a_{iS}/\theta}}{\sum_{i'} e^{a_{i'S}/\theta}}, \frac{e^{a'_{jS}/\theta}}{\sum_{S'} e^{a'_{iS'}/\theta}} \right).^1$$

This ensures that

$$\sum_i A_{iS}^{\text{agent-bundle}} \leq 1, \forall S \in K \tag{10}$$

$$\sum_S A_{iS}^{\text{agent-bundle}} \leq 1, \forall i \in N \tag{11}$$

From the definition of z_{iS}, the allocation of any bundle S is limited by its least-available item:

$$\sum_i \sum_{S \ni j} z_{iS} \leq \sum_i \sum_{S \ni j} \min_{j \in S}\{B_{jS}^{\text{adjusted}}\} \cdot A_{iS}^{\text{agent-bundle}}.$$

Since B^{adjusted} is normalized with softmax along item-wise dimension to ensure that no item j is allocated more than once:

$$\sum_{S \ni j} \min_{j \in S}\{B_{jS}^{\text{adjusted}}\} \leq \sum_{S \in K} \min_{j \in S}\{B_{jS}^{\text{adjusted}}\} \leq 1, \quad \forall j \in M.$$

Thus,

$$\sum_i \sum_{S \ni j} z_{iS} \leq \sum_i \sum_{S \ni j} A_{iS}^{\text{agent-bundle}}.$$

For a fixed item j, we can write:

$$\sum_i \sum_{S \ni j} A_{iS}^{\text{agent-bundle}} \leq \sum_{S \ni j} \sum_i A_{iS}^{\text{agent-bundle}}.$$

From (10), we have:

$$\sum_i \sum_{S \ni j} A_{iS}^{\text{agent-bundle}} \leq \sum_{S \ni j} 1.$$

From (11), the number of bundles S containing j is at most the total number of bundles k, constraint (4) is satisfied:

$$\sum_i \sum_{S \ni j} A_{iS}^{\text{agent-bundle}} \leq 1.$$

Thus, $\mathbf{Z}$ is combinatorially feasible. $\square$

---

[1]In all of our implementations, the softmax function incorporates a *temperature* parameter $\theta \in \mathbb{R}^+$, inspired by the Boltzmann distribution in statistical mechanics, which controls the concentration of probability mass around the highest logit.

3.2 CANet Architecture

Motivated by RegretNet [Duetting *et al.*, 2019], we present CANet, a deep neural network architecture designed for combinatorial auction mechanisms. The CANet architecture, visualized in Figure 1a, comprises two modules: the Deep Allocation Network and the Deep Pricing Network. The input to CANet, denoted by $\mathbf{b}$, is a vector $\mathbf{b} \in \mathbb{R}^{n \cdot k}$, where $b_{iS} \sim F_i$ is the bid submitted by agent i for bundle S.

The Deep Allocation Network computes a feasible allocation matrix $\mathbf{Z} \in \mathbb{R}^{n \times k}$ ensuring that both individual item constraints and global allocation constraints are satisfied. The input $\mathbf{b}$ is passed through fully connected layers with non-linear activations. The network produces three output vectors $\mathbf{A} \in \mathbb{R}^{n \times k}$ and $\mathbf{A}' \in \mathbb{R}^{n \times k}$ representing an unnormalized agent-bundle allocation; and $\mathbf{B} \in \mathbb{R}^{m \times k}$ representing an unnormalized bundle-item allocation. These outputs are combined following the process described in section 3.1 to form a feasible allocation matrix $\mathbf{Z}$.

The Deep Pricing Network computes the pricing vector $\tilde{\mathbf{p}} \in \mathbb{R}^n$, which represents a discount applied to the bids. The input $\mathbf{b}$ is processed through fully connected layers with sigmoid transformation, to ensure $\tilde{p}_i \in [0, 1]$. The final price is vector $\mathbf{p}$ with $p_i = \tilde{p}_i \sum_{S=1}^k z_{iS} b_{iS}$.

3.3 CAFormer Architecture

To enhance expressivity and ensure permutation-equivariance and context-awareness, we introduce CAFormer. This architecture combines exchangeable layers (proposed in Zaheer *et al.* [2018] and used in EquivariantNet Rahme *et al.* [2021a]) with stacked attention layers [Vaswani *et al.*, 2017], as seen in RegretFormer [Ivanov *et al.*, 2022]. The architecture is illustrated in Figure 1b. The attention mechanism is permutation-invariant, while exchangeable layers ensure permutation-equivariance—properties particularly desirable in symmetric auctions. For asymmetric settings or preserving input order across the bundle dimension, we apply agent-wise and bundle-wise positional encoding as presented in [Vaswani *et al.*, 2017] after the exchangeable layers.

The input to CAFormer is a two-dimensional matrix $\mathbf{b} \in \mathbb{R}^{n \times k}$. Unlike CANet that can output an item-bundle allocation at the last layer, CAFormer must be designed properly to maintain insensitivity to the size of the problem. We achieve it by transforming the bid matrix into a representation of item-bundle allocation $\mathbf{b}' \in \mathbb{R}^{m \times k}$ through matrix multiplication with the individual item bid matrix $\mathbf{b}^{\text{item}} \in \mathbb{R}^{n \times m}$.

$$\mathbf{b}' = \text{MatMul}(\mathbf{b}^\top, \mathbf{b}^{\text{item}})^\top,$$

where $^\top$ denotes the transpose operation. Both $\mathbf{b}$ and $\mathbf{b}'$ are then expanded in a feature dimension that contain global information of bids with exchangeable layers, a key building block in architectures like EquivariantNet [Rahme *et al.*, 2021a] and RegretFormer [Ivanov *et al.*, 2022] which ensures permutation-equivariance. This layer processes a tensor $\mathbf{X} \in \mathbb{R}^{n \times k \times 1}$, into an output tensor $\mathbf{X}_{\text{ex}} \in \mathbb{R}^{n \times k \times d}$. The exchangeable layer computes the element of the output channels by aggregating input tensor B over elements, rows, columns, and globally, weighted by learnable parameters, and applies a non-linear activation σ. Details can be found in Deep Sets

[Zaheer *et al.*, 2018].

$$\mathbf{b}_{ex} = \text{Exchangeable}(\mathbf{b}), \quad \mathbf{b}'_{ex} = \text{Exchangeable}(\mathbf{b}')$$

After each exchangeable layer, we apply sequential attention-based blocks to the tensors $\mathbf{b}_{ex}$ and $\mathbf{b}'_{ex}$. Each block consists of multi-head self-attention layers with residual connections.

For each layer, the input tensor is reshaped to enable item-wise, bundle-wise, or agent-wise self-attention. Item-wise attention operates on reshaped input $\mathbf{b}'_{item} \in \mathbb{R}^{m \times d}$, bundle-wise attention operates on $\mathbf{b}_{bundle}$ and $\mathbf{b}'_{bundle} \in \mathbb{R}^{k \times d}$, and agent-wise attention operates on $\mathbf{b}_{agent} \in \mathbb{R}^{n \times d}$, where d is the feature dimensionality of the input. The outputs are concatenated, forming $\mathbf{b}'_{concat} \in \mathbb{R}^{m \times k \times d'_{concat}}$ and $\mathbf{b}_{concat} \in \mathbb{R}^{n \times k \times d_{concat}}$, where d_{concat} and d'_{concat} are the combined attended features. After the attention-blocks, we use fully-connected layers f^{item}_{fc}, f^{bundle}_{fc}, and f^{agent}_{fc} to map the features back to the original dimensionality 1:

$$\mathbf{B} = f^{item}_{fc}(\mathbf{b}'_{concat}),$$
$$\mathbf{A} = f^{agent}_{fc}(\mathbf{b}_{concat}), \quad \mathbf{A}' = f^{bundle}_{fc}(\mathbf{b}_{concat})$$

Again, following the procedure outlined in Section 3.1, we derive a feasible allocation matrix $\mathbf{Z}$. The pricing vector $\tilde{\mathbf{p}} \in \mathbb{R}^n$ is derived by applying a fully connected layer with sigmoid activation to the agent-wise attention output $\mathbf{b}_{agent}$, reducing its dimensionality to 1. The final prices are computed using the same method as in CANet.

3.4 Training Procedure

Similarly to RegretNet, our training process also includes an inner utility maximization (9) and an outer revenue maximization and regret minimization (8). For brevity, we refer the reader to the original paper by Duetting *et al.* [2019] for details of the adversarial training. Previous works [Duetting *et al.*, 2019; Rahme *et al.*, 2021a; Ivanov *et al.*, 2022] relies heavily on gradient-based Lagrangian optimization (see Section 2.2). This method is known to be sensitive to the choice of hyper-parameters. An example is in the outer loss (8), when λ_i becomes excessively large, the model disproportionately focuses on minimizing regret, and gradients for regret become negligible due to a scaling effect or poor numerical conditioning. To ensure a stable tradeoff, we constrain the loss weights such that their sum is 1. This simple balancing mechanism does not directly tackle the overemphasis on regret, but it ensures regret gradient does not shrink due to the excessive weight λ_i. Additionally, we apply a logarithmic transformation to the revenue term to avoid unbounded growth. This is in the spirit of multi-task learning, which seeks to optimize both objectives in the present of trade-offs. Our outer loss function is defined as:

$$\mathcal{L}_{outer} = -w_{rev} \log(1 + \text{rev}) + w_{rgt}\text{rgt},$$

where w_{rev} and w_{rgt} are task weights that represent the emphasis on revenue and regret, respectively.

Following Ivanov *et al.* [2022], we also set a regret budget, allowing controlled violations of DSIC by gradually decreasing the budget during training. The update for w_{rgt} is explicitly computed as:

$$g_t = \log(\text{rgt}) - \log(\bar{\text{rgt}}) - \log(1 + \alpha\text{rev}) \quad (12)$$

Algorithm 1 Adam Update Rule for w_{rev} and w_{rgt}

1: Initialize $m_w \leftarrow 0$, $v_w \leftarrow 0$, β_1, β_2, ϵ, γ_w, ρ
2: Compute gradient g_t for w_{rgt} with equation (12)
3: Update first and second moments
4: $\quad m_w \leftarrow \beta_1 m_w + (1 - \beta_1)g_t$
5: $\quad v_w \leftarrow \beta_2 v_w + (1 - \beta_2)g_t^2$
6: Bias correction
7: $\quad \hat{m}_w \leftarrow \dfrac{m_w}{1 - \beta_1^{\epsilon+1}}; \quad \hat{v}_w \leftarrow \dfrac{v_w}{(1 - \beta_2^{\epsilon+1}}$
8: Update w_{rgt}
9: $\quad w_{rgt} \leftarrow w_{rgt} + \gamma_w \dfrac{\hat{m}_w}{(\sqrt{\hat{v}_w} + \epsilon)}$
10: Normalize weights
11: $\quad w_{rgt} = \max\left(\dfrac{e^{w_{rgt}/\rho} - e^{-w_{rgt}/\rho}}{e^{w_{rgt}/\rho} + e^{-w_{rev}/\rho}}, 0\right)$
12: $\quad w_{rev} \leftarrow 1 - w_{rgt}$
13: return w_{rev}, w_{rgt}

---

where $\bar{\text{rgt}}$ is a target regret value. The parameter α is optional and adjusts the ratio between the target regret and revenue.

To ensure proper balance between revenue and regret, we normalize the weights with tanh function scaled with paramter ρ. Our w_{rev} and w_{rgt} are maintained in range $[0, 1]$. Using exponential moving averages for the first and second moments of g_t, we update w_{rgt} with a bias-corrected Adam-style rule with learning rate γ. We explicitly present the update rule for the weights in Algorithm 1.

4 Experimental Results

Set up Our framework is implemented in PyTorch and trained for 50,000 or 100,000 outer optimization iterations, depending on the problem size. All networks used Glorot uniform initialization [Glorot and Bengio, 2010] and tanh activations. We sampled 640,000 valuation profiles offline for training and 10,000 profiles online for testing. Each outer optimization update included 50 inner steps during training and 1,000 during validation. Typical hyperparameters include: tanh scaler $\rho = 2$, softmax temperature $\theta \in \{10, 15, 25\}$, network learning rates $\{0.0004, 0.0007\}$, regret learning rate $\gamma \in \{0.01, 0.02\}$ and the target regret-revenue adjustment factor $\alpha \in \{0.5, 1\}$. CANet's allocation and pricing networks use $\{3, 6\}$ layers with 100 hidden nodes each, while CAFormer has a single attention layer with 2 heads and 64 hidden features. We initialized $w_{rgt} = 1$, exponentially annealling regret targets from $\bar{\text{rgt}}_{start} = 0.05$ to $\bar{\text{rgt}}_{end} \in \{0.0008, 0.001, 0.002, 0.003\}$ in $2/3$ of the training iterationw. Experiments are conducted on three scales: 2 bidders, 2 items (3 bundles); 2 bidders, 3 items (7 bundles); and 2 bidders, 5 items (31 bundles).

Baselines We compare against the VCG mechanism [Vickrey, 1961; Clarke, 1971; Groves, 1973], previously reported results from RegretNet and RegretFormer [Ivanov *et al.*, 2022], Affine Maximizer Auction trained via grid search (AMA and VVCA), and local search methods (BLAMA, ABAMA, BBBVVCA) [Sandholm and Likhodedov, 2015]. We re-implement VCG mechanism and the search algorithms [Sandholm and Likhodedov, 2015]. The search algorithms

	2x2 (B)		2x3 (B)		2x5 (B)		2x2 (C)		2x3 (C)		2x5 (C)	
	Rev	Rgt	Rev	Rgt	Rev	Rgt	Rev	Rgt	Rev	Rgt	Rev	Rgt
VCG	2.405	0	3.537	0	5.838	0	2.847	0	4.239	0	6.987	0
AMA	2.760	0	-	-	-	-	4.240	0	-	-	-	0
VVCA	2.770	0	-	-	-	-	4.240	0	-	-	-	0
BLAMA	2.630	0	4.125	0	7.280	0	4.080	0	4.812	0	8.164	0
ABAMA	2.630	0	4.166	0	7.145	0	4.010	0	5.916	0	11.140	0
BBBVVCA	2.620	0	4.105	0	7.156	0	4.010	0	5.898	0	11.135	0
RegretNet	2.871	0.001	-	-	-	-	4.270	0.001	-	-	-	-
CANet	2.904	0.001	4.369	0.002	7.304	0.007	4.285	0.001	6.556	0.001	11.393	0.002
CAFormer	**2.919**	0.001	**4.388**	0.002	**7.318**	0.003	**4.403**	0.002	**6.693**	0.001	**11.534**	0.004

Table 1: Combinatorial results: Grid search methods (AMA and VVCA) become computationally infeasible in higher dimensions. RegretNet results are limited to 2×2 cases. CANet and CAFormer beat the benchmarks in all settings. Both CAFormer and CANet achieve negligible regret. But CAFormer consistently outperforms CANet in revenue.

are trained on 100 samples for 10 different random seeds, and evaluated on 1,000,000 samples.

Performance in non-combinatorial setting We compare the computational results for non-combinatorial setting, where bidders draw their value for each item from $U[0, 1]$ (setting A). In these experiments, we make additive valuation assumption. The revenue performance is reported in Table 2.

Our revenue performance is comparable to the machine learning-powered models, which outperform the heuristic designs. The slight underperformance in comparison to RegretNet and RegretFormer is due to the complex constrained optimization space. For instance, in the 2-agent, 5-item setting, RegretNet and RegretFormer optimize allocations at the agent-item level (2×5), while CANet and CAFormer optimize at the agent-bundle level (2×31), which introduces excessive complexity to non-combinatorial settings, preventing full convergence. Besides, regret estimation is less reliable in larger-scale settings, because adversarial optimization of the inner loss (9) might be inaccurate, which affects the convergence of optimal revenue. If the utility of the best misreport is underestimated, the revenue might be overestimated. Future work is encouraged to evaluate this approximation's accuracy.

Performance in combinatorial setting We compare the results in combinatorial setting. Let v_{ij}^{item} be the valuation of of bidder i for item j, individually. The valuation of bidder i for bundle S is drawn as $v_{iS} = \sum_{j \in S} v_{ij}^{\text{item}} + c_{iS}$ with $c_{iS} \sim U[-1, 1]$. For symmetric settings (B), $v_{ij}^{\text{item}} \sim U[1, 2], \forall i \in N$. For asymmetric settings (C), $v_{1j}^{\text{item}} \sim U[1, 2]$ and $v_{2j}^{\text{item}} \sim U[1, 5]$.

The results in Table 1 show that our models outperform heuristic benchmarks and RegretNet in all settings. As we specify the regret target as a proportion of revenue, the regret is higher in larger-scale settings. Both CAFormer and CANet achieve negligible regret in all experiments. However, CAFormer, with a greater number of parameters and expressivity, consistently outperforms CANet in revenue.

	2x2 (A)		2x3 (A)		2x5 (A)	
	Rev	Rgt	Rev	Rgt	Rev	Rgt
VCG	0.667	0	1	0	1.671	0
AMA	0.860	0	-	0	-	0
VVCA	0.866	0	-	0	-	0
BLAMA	0.786	0	1.222	0	2.231	0
ABAMA	0.786	0	1.255	0	2.251	0
BBBVVCA	0.776	0	1.238	0	2.242	0
RegretNet	0.878	0.001	1.317	0.001	2.339	0.001
RegretFormer	**0.908**	0.001	**1.416**	0.001	**2.453**	0.001
CANet	0.879	0.001	1.317	0.001	2.282	0.004
CAFormer	0.891	0.001	1.326	0.001	2.329	0.005

Table 2: Non-combinatorial results: Our revenue performance is comparable to that of RegretNet and RegretFormer, which outperform heuristic designs.

5 Discussion and Future Work

We view our approach as an initial step towards generalizing differentiable economics for combinatorial auction, though it remains an open theoretical question whether it can fully represent the true optimal mechanism. Although most revenue-maximizing heuristic methods struggle with combinatorial explosion in larger-scale settings, our approach offers greater efficiency. While our formulation and experiments focused on valuations with known distributions (particularly uniform) and considered all possible bundles, by design, this approach is inherently not limited to assumptions about the structures of bundles and valuation profiles. It can even be applied to online auctions and auctions whose distribution is unknown.

In practice, if items are divisible, fractional allocations directly represent feasible outcomes; however, if they are not, implementing the lottery in large-scale settings can be challenging. In some cases, initial use of randomized mechanisms can enable convergence to effective solutions, which can sub-

sequently be fine-tuned to obtain deterministic results.

Of the two proposed architectures, CAFormer not only outperforms in revenue, but also offers other advantages: permutation-equivariance, the ability to generalize to unseen settings with variable input size, and to incorporate contextual information. Moreover, we offer a technical perspective, which considers the multi-task nature of the problem, providing flexibility in training. However, the non-convexity of the min-max optimization poses a challenge for gradient-based methods, as they lack theoretical guarantees for convergence. This limits the power of differentiable economics in larger-scale problems. Exploring alternative loss-balancing techniques could yield further insights, but this is beyond the scope of our current work and is left for future investigation.

Although our focus is on auctions, we expect that the proposed approach could extend to other differentiable combinatorial optimization problems, as long as the feasible region can be parameterized for gradient-based learning and the objective function remains differentiable. Developing techniques to incorporate constraints more effectively in gradient-based optimization is a promising future research direction.

References

Mark Armstrong. Optimal multi-object auctions. *The Review of Economic Studies*, 67(3):455–481, 07 2000.

Christopher Avery and Terrence Hendershott. Bundling and optimal auctions of multiple products. *The Review of Economic Studies*, 67(3):483–497, 2000.

Patrick Bajari and Ali Hortaçsu. The winner's curse, reserve prices, and endogenous entry: Empirical insights from ebay auctions. *The RAND Journal of Economics*, 34(2):329–355, 2003.

Gianluca Brero, Benjamin Lubin, and Sven Seuken. Combinatorial auctions via machine learning-based preference elicitation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 128–136. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

Gianluca Brero, Benjamin Lubin, and Sven Seuken. Machine learning-powered iterative combinatorial auctions, 2021.

Edward H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

Peter Cramton, Yoav Shoham, and Richard Steinberg. *Combinatorial Auctions*. The MIT Press, December 2005.

Michael Curry, Tuomas Sandholm, and John Dickerson. Differentiable economics for randomized affine maximizer auctions. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI-2023, page 2633–2641. International Joint Conferences on Artificial Intelligence Organization, August 2023.

Paul Duetting, Zhe Feng, Harikrishna Narasimhan, David Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1706–1715. PMLR, 09–15 Jun 2019.

Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259, March 2007.

Zhe Feng, Harikrishna Narasimhan, and David C. Parkes. Deep learning for revenue-optimal auctions with budgets. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 354–362, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

Noah Golowich, Harikrishna Narasimhan, and David C. Parkes. Deep learning for multi-facility location mechanism design. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, IJCAI-2018, page 261–267. International Joint Conferences on Artificial Intelligence Organization, July 2018.

Theodore Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973.

Dmitry Ivanov, Iskander Safiulin, Igor Filippov, and Ksenia Balabaeva. Optimal-er auctions through attention, 2022.

Arash Jamshidi, Seyed Mohammad Hosseini, Seyed Mahdi Noormousavi, and Mahdi Jafari Siavoshani. Differentially private machine learning-powered combinatorial auction design, 2024.

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*, abs/2010.06398, 2020.

Ron Lavi and Chaitanya Swamy. Truthful and near-optimal mechanism design via linear programming. *J. ACM*, 58(6), December 2011.

Eric Maskin, J. Riley, and F. Hahn. *Optimal Multi-Unit Auctions*, pages 312–335. Oxford University Press, 1989. Reprinted in P. Klemperer, The Economic Theory of Auctions, London: Edward Elgar, 2000.

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

Jad Rahme, Samy Jelassi, Joan Bruna, and S. Matthew Weinberg. A permutation-equivariant neural network architecture for auction design, 2021.

Jad Rahme, Samy Jelassi, and S. Matthew Weinberg. Auction learning as a two-player game, 2021.

Sai Srivatsa Ravindranath, Zhe Feng, Shira Li, Jonathan Ma, Scott D. Kominers, and David C. Parkes. Deep learning for two-sided matching, 2023.

Sai Srivatsa Ravindranath, Zhe Feng, Di Wang, Manzil Zaheer, Aranyak Mehta, and David C. Parkes. Deep reinforcement learning for sequential combinatorial auctions, 2024.

Abhishek Ray, Mario Ventresca, and Karthik Kannan. A graph-based ant algorithm for the winner determination problem in combinatorial auctions. *Information Systems Research*, 32(4):1099–1114, December 2021.

K. Roberts. The characterization of implementable choice rules. In J.-J. Laffont, editor, *Aggregation and Revelation of Preferences*, pages 321–348. North Holland Publishing, 1979.

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

Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1–2):1–54, February 2002.

Tuomas Sandholm. Automated mechanism design: A new application area for search algorithms. In Francesca Rossi, editor, *Principles and Practice of Constraint Programming – CP 2003*, pages 19–36, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

Andrea Tacchetti, DJ Strouse, Marta Garnelo, Thore Graepel, and Yoram Bachrach. Learning truthful, efficient, and welfare maximizing auction rules, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

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

Qinghua Wu and Jin-Kao Hao. A clique-based exact method for optimal winner determination in combinatorial auctions. *Information Sciences*, 334–335:103–121, March 2016.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets, 2018.