Understanding Deep Learning for Two-Sided Matchings

Aditya Kumar Shaw
ksaditya@iisc.ac.in

Arun Kumar
arunkumar12@iisc.ac.in

Prabhav Verma
prabhavverma@iisc.ac.in

Shishir Madhab Roy
shishirm@iisc.ac.in

Abstract

We investigate the integration of deep learning methods into the development of two-sided matching mechanisms, exploring the trade-off between strategy-proofness and stability. This study introduces differentiable surrogates to quantify ordinal strategy-proofness and stability, enabling neural networks to map preference profiles to valid matchings. Experimental results demonstrate precise control over the trade-off through parameter tuning, surpassing traditional approaches. By leveraging recommended papers and relevant literature, we examine the potential of machine learning pipelines in market design, aiming to replicate and enhance existing results.

1 Introduction

With the rise in computational power and application of deep learning techniques in many areas and problems, deep learning has become ubiquitous. The goal of this project is to study the applications of these methods in game theoretic mechanism design , through the problem of two sided matching and auction design and to try and reproduce and improve upon their results.

Though the problem of application of deep learning in auction design has some literature and research done into it, deep learning in two sided matching is fairly new and not much literature is available on it.

The main paper under review assigned to us also draws motivation from the works of the likes of Dutting et al and srivatsa et al.

The *two-sided matching problem* involves pairing two sets of agents, each with a strict preference order over the other set, aiming to maximize overall stability. The applications include matching students and colleges, workers and firms, marriageable men and women and many more.

Stability in two-sided matching refers to a situation where there are no two agents, one from each side, who prefer to match with each other over their current matches. *Strategy Proofness (SP)* in two-sided matching ensures that no agent can manipulate their preferences to improve their outcome without worsening someone else's. *Individual Rationality (IR)* in two-sided matching ensures every agent is assigned a partner they prefer over being unmatched, preventing regrets for participating.

It is proven that stability and SP cannot be simultaneously achieved in two-sided matching Dubins and Freedman [1981], Roth [1982]. Thus, a trade-off must exist between stability and SP. However, this trade-off is poorly understood beyond the existing solutions and Ravindranath et al. [2023] successfully tried to get a better understanding of this trade-off using deep learning.

Preprint. Under review.

2 Preliminaries

For notations, refer to the appendix A

Here we define some terms and sate theorems and results that are deemed necessary to understand the report.

2.1 Deferred Acceptance (DA)

Gale and Shapley [1962] introduced a simple matching mechanism for two-sided matching problem - DA - which is stable (for both sides) but not SP (for one side).

1. $\forall w \in W : w$ creates a list of its acceptable firms called remaining firms.
2. $\forall w \in W : w$ proposes to its most preferred acceptable remaining firm.
3. $\forall f \in F : f$ tentatively accepts its best proposal, if any, and rejects other proposals.
4. $\forall w \in W : w$ if rejected by firm f, removes f from its remaining firms list.
5. Repeat steps $[2, 3, 4, 5]$ until no changes occur in matchings.

This is the *worker-proposing DA algorithm*, firm-proposing DA algorithm can be defined analogously.

2.2 Random Serial Dictatorship (RSD)

Abdulkadiroglu et al. [2006] introduced another matching mechanism - RSD - which when used to solve two-sided matching problem is SP but not stable.

1. Sample a priority order, π[1], on the set $W \cup F$ uniformly at random.
2. For k = $\{1, ..., |\pi|\}$:
   (a) If π_k is not yet matched, then match π_k to its most preferred unmatched agent or to $\perp$ if all acceptable agents are already matched to some other agent.

2.3 Top trading cycles (TTC)

The TTC mechanism introduced in Shapley and Scarf [1974] when used to solve two-sided matching problem is neither SP (SP for one side) nor stable. Following is the *worker-proposing TTC algorithm*, firm-proposing TTC algorithm can be defined analogously.

1. $\forall a \in W \cup F : a$ creates a list of its acceptable agents called remaining agents.
2. Create a directed graph G with each unmatched agent a pointing to its most preferred agent. The agent a can point to itself if its remaining agent list is empty.
3. Every worker w that is part of a cycle in G is matched to a firm (or itself) it points to.
4. All unmatched agents remove from their remaining agents list the agents that got matched in this round.
5. Repeat steps $[2, 3, 4, 5]$ until all workers are matched.

2.4 Randomized Matching Mechanism

Randomized Matching Mechanism, g, maps preference profiles, $\succ$, to distributions on matching, $g(\succ) \in \Delta(\mathcal{B})$. $r \in [0, 1]^{|\overline{W}| \times |\overline{F}|}$ is the marginal probability, r_{wf}, with which the worker w is matched with firm f, for each $w \in \overline{W}, f \in \overline{F}$.

Definition 1 (Ex ante justified envy) *A randomized matching r causes ex ante justified envy if -*

- *some worker w prefers f over some fractionally matched firm f' (including $f' = \perp$) and firm f prefers w over some fractionally matched worker w' (including $w' = \perp$).*

[1] π is a permutation of the agents $a \in W \cup F$ denoting the priority of agents

- *some worker w finds a fractionally matched $f' \in F$ unacceptable i.e. $r_{wf'} > 0, \perp \succ_w f'$ or some firm f finds a fractionally matched $w' \in W$ unacceptable i.e. $r_{w'f} > 0, \perp \succ_f w'$.*

Definition 2 (Ex ante stable) *A randomized matching, r, is ex ante stable if and only if it does not cause any ex ante justified envy.*

Definition 3 (Utility Function) *For worker w we define, $u_w : \overline{F} \to \Re$, a $\succ_w$-utility function where $u_w(f) > u_w(f') \iff f \succ_w f', \forall f, f' \in \overline{F}$. Analogously, we define $\succ_f$-utility function for firm f.*

Definition 4 (Ordinal SP) *A randomized matching mechanism, g, satisfies ordinal SP if and only if $\forall a \in W \cup F, \forall \succ \in P, \forall \succ_a$-utility function, and for all reports $\succ'_a$ we have -*

$$\mathbb{E}_{\mu \sim g(\succ_a, \succ_{-a})}[u_i(\mu(i))] \geq \mathbb{E}_{\mu \sim g(\succ'_a, \succ_{-a})}[u_i(\mu(i))] \tag{1}$$

Definition 5 (First order Stochastic Dominance (FOSD)) *A randomized matching mechanism, g, satisfies FOSD if and only if $\forall w \in W, \forall f' \in \overline{F}$ such that $f' \succ_w \perp, \forall \succ_{-w}$, we have -*

$$\sum_{f \in F : f \succ_w f'} g_{wf}(\succ_w, \succ_{-w}) \geq \sum_{f \in F : f \succ_w f'} g_{wf}(\succ'_w, \succ_{-w}) \tag{2}$$

The *Stability Violation (STV())* acts as a differentiable surrogate for stability and it is given by -

$$STV(g^\theta) = \mathbb{E}_{\succ}[stv(g^\theta, \succ)] \tag{3}$$

See B.

Theorem 1 *A randomized mechanism, g^θ, is ex ante stable up to zero-measure events if and only if $STV(g^\theta) = 0$.*

The *Ordinal SP Violation (RGT())* acts as a differentiable surrogate for ordinal SP and it is given by -

$$RGT(g^\theta) = \mathbb{E}_{\succ}[regret(g^\theta, \succ)] \tag{4}$$

See C

Theorem 2 *A randomized mechanism, g^θ, is ordinal SP up to zero-measure events if and only if $RGT(g^\theta) = 0$.*

Following is the *loss function* proposed, where $\lambda \in [0, 1]$ is the trade-off between stability and SP -

$$\min_\theta \lambda * STV(g^\theta) + (1 - \lambda) * RGT(g^\theta) \tag{5}$$

3 Related Works and Motivation

Historically, the *automated mechanism design (AMD)* approach framed problems (including auctions & two-sided matching) as a linear program. However, this formulation suffers from exponential growth in complexity with the increase in number of agents and items.

Recent works, Dütting et al. [2022], Curry et al. [2020], Shen et al. [2021], Rahme et al. [2021], overcame this limitation by using deep neural networks for solving economic design problems, in particular they explored the *design of auctions* using deep learning frameworks.

Other related work includes using SVMs to search for stable matching without considering SP, and stable matching in context of bandid problems where agents' preferences are unknown a priori.

In this section we describe some previous classical results that prove to be helpful in understanding the problem and study some deep neural techniques that have been deployed to solve similar mechanism design problems and have motivated the application of Deep Learning in Matching Problems.

3.1 College Admissions and the Stability of Marriage

3.1.1 Introduction & Background

Gale and Shapley [1962] explores the problem of stable matching in the context of college admissions and marriage. The paper presents the Gale-Shapley algorithm, alias Deferred Acceptance (DA) algorithm, which provides an elegant solution to the stable matching problem by guaranteeing that no participant has an incentive to deviate from the assigned matching.

The paper begins by introducing the concept of stability in matching scenarios and outlines the challenges associated with designing mechanisms that produce stable matches. It then presents the Gale-Shapley algorithm, which iteratively matches colleges to students (or men to women in the original context) based on their preferences until a stable matching is achieved.

3.1.2 Experiments

While the paper does not include experimental results in the traditional sense, it provides theoretical proofs of the stability and optimality of the Gale-Shapley algorithm. Through mathematical analysis, Gale and Shapley demonstrate that the algorithm always produces stable matches and that it maximizes the preferences of one side of the matching while maintaining stability.

3.1.3 Conclusion and Future Work

The paper offers a foundational contribution to the field of matching theory, providing a robust algorithm for solving the stable matching problem. Future work in this area could explore extensions and applications of the Gale-Shapley algorithm in various real-world scenarios beyond college admissions and marriage, potentially leading to advancements in mechanism design and allocation problems.

3.1.4 Broader Impact

While the immediate impact of this work may be primarily academic, the principles outlined in the paper have implications for real-world applications such as college admissions, job placements, and organ donations. By providing a mechanism for ensuring stability and fairness in matching scenarios, the Gale-Shapley algorithm contributes to the development of more efficient and equitable allocation systems.

3.2 Machiavelli and The Gale-Shapley Algorithm

This paper seeks to generalize the results of [Gale and Shapley, 1962] and establish that their algorithm gives each student the best university available in a stable system of assignments.

Here the authors prove that students cannot improve their fate by lying about their preferences, no coalition of students can simultaneously improve the lot of all its members if those outside the coalition state their true preferences.

3.3 Changing The Boston School Choice Mechanism

3.3.1 Introduction & Background

In July 2005, the Boston School Committee decided to replace the existing Boston school choice mechanism with a deferred acceptance mechanism, aiming to simplify strategic choices for parents. The previous mechanism, a priority matching system, faced criticism due to its susceptibility to strategic manipulation.

3.3.2 Experiments

This paper presents empirical evidence against the prior Boston mechanism and advocates for the adoption of a strategy-proof mechanism. They found that under the previous system, some parents strategically considered school capacity constraints, while others did not. Consequently, many students were left unassigned despite having viable choices under a different strategy.

The deferred acceptance algorithm, initially studied by Gale and Shapley [1962] in two-sided matching markets, guarantees stable matchings, where no unmatched agents prefer each other over their assigned matches. In contrast, the Boston mechanism lacks strategy-proofness, allowing students to improve their assignments through preference misrepresentation.

3.3.3 Conclusion and Future Work

The adoption of the deferred acceptance mechanism in Boston addresses concerns regarding strategic manipulation and ensures stable assignments. This shift highlights the importance of strategy-proof mechanisms in school choice allocation.

Future research could focus on evaluating the long-term effects of the new mechanism on student outcomes and educational equity. Additionally, exploring variations of the deferred acceptance algorithm tailored to specific contexts may offer further insights into optimizing school choice mechanisms.

3.3.4 Broader Impact

The transition in the Boston school choice mechanism underscores the broader significance of strategy-proof mechanisms in allocation systems. By mitigating strategic behavior and promoting stability, such mechanisms contribute to fairer and more efficient resource distribution in various domains beyond education. This shift sets a precedent for improving allocation mechanisms in other contexts, fostering equitable outcomes and enhancing decision-making processes.

3.4 Strategy-Proofness versus Efficiency in Matching with Indifferences: Redesigning the NYC High School Match

3.4.1 Introduction & Background

The NYC High School Match problem [Abdulkadiroğlu et al., 2005] involves assigning students to high schools based on their preferences and school capacities. Abdulkadiroğlu et al. [2009] identified several important concerns and constraints in the NYC High School match and proposed solutions to address them.

3.4.2 Concerns and Proposed Solutions

Stability of the Matching In two-sided matching models, stability is crucial. A matching is stable if it is individually rational and there is no blocking pair where a student and a school both prefer each other over their current assignment. Some schools in the NYC system actively rank students, leading to the possibility of strategic manipulation if a blocking pair exists. To address this, the authors propose a stable matching algorithm that considers both schools' and students' preferences, ensuring that any attempt to circumvent the match is futile.

Efficiency for Students The old system in NYC was inefficient, often resulting in students being rejected by all their choices. The new system aims to improve efficiency by using deferred acceptance algorithms that identify stable matches optimal for one side of the market. By prioritizing student welfare, the redesigned mechanism ensures that students are assigned to schools they prefer as much as possible, thereby promoting efficiency in the matching process.

Strategy-Proofness for Students In the old system, some schools gave higher priority to students who ranked them as first or second choice, leading to strategic decisions by students while ranking schools. To mitigate this, the new system aims for strategy-proofness, where it is a dominant strategy for students to state their true preferences. While no stable mechanism is completely strategy-proof for all agents, the student-proposing deferred acceptance mechanism used in the redesign ensures strategy-proofness for students in case of strict preferences.

Tiebreaking Mechanisms Tiebreaking is necessary to resolve indifference in school preferences, but it introduces artificial stability constraints that may harm student welfare [Erdil and Ergin, 2008]. The authors consider different tiebreaking methods and opt for single tiebreaking, which has superior welfare properties compared to multiple tiebreaking. Although single tiebreaking may not always lead to student-optimal stable matching, it strikes a balance between stability and strategy-proofness.

Trade-off between Strategy-Proofness and Efficiency The authors analyze the trade-off between strategy-proofness and student welfare empirically, revealing significant costs associated with imposing strategy-proofness. While strategy-proof mechanisms ensure fairness and transparency, they may lead to efficiency losses. Understanding this trade-off is crucial for designing effective matching mechanisms.

3.4.3 Conclusion and Future Work

While the empirical evidence highlights substantial efficiency costs associated with strategy-proofness, quantifying the costs of lacking a strategy-proof mechanism remains challenging. The iterative process of designing the NYC high school match exemplifies the reliance on available theory alongside data-driven decision-making. However, it also reveals gaps where theoretical frameworks fall short, necessitating further development to address the complexities of matching systems. This paper contributes to bridging these gaps by developing new theory and providing empirical support for design decisions made in practice. Moving forward, there is a clear need for continued theoretical exploration to inform the design and redesign of school matching systems, addressing emerging challenges and ensuring efficiency, fairness, and transparency.

3.4.4 Broader Impact

This paper's insights extend beyond school choice mechanisms, resonating with economists designing practical markets. By emphasizing the importance of theoretical frameworks in guiding design decisions, it underscores interdisciplinary collaboration's significance. The iterative process of designing the NYC high school match illustrates real-world complexities and challenges. Consequently, the paper's findings have broader implications for designing allocation mechanisms across domains, promoting equity and efficiency. Addressing theoretical gaps and informing practical decisions, this research advances mechanism design, shaping policies for equitable resource allocation.

3.5 Optimal Auctions through Deep Learning

This paper shows the first step to implement an auction through machine learning technique. They have also introduced an auction as a learning problem. (Dütting et al. [2019]) provides general purpose, end-to-end approach for solving the multi-item auction design problem using multi-layer neural network with bidder valuations being input and allocation and payment decision being the output. The goal is maximizing an expected revenue. It also ensure that chosen auction satisfies incentive compatibility. They also prove a novel generalization bound, which implies that, with high probability, for their architectures high revenue and low regret on the training data translates into high revenue and low regret on freshly sampled valuations. This paper have a three different implementation of neural network architecture for three different valuations of bidders. Which they refer to as RegretNet. The different valuations of bidder are additive, unit-demand, and general combinatorial valuations.

3.5.1 Setup and Notations

Consider a setting with a set of n bidders N = 1, ... , n and m items M = 1, ... , m. Each bidder i has a valuation function $v_i : 2^M \rightarrow R \geq 0, where\, v_i(S)$ denotes how much bidder has values for subset of items $S \subseteq M$ .Additive valuation for each bidders $v_i(S) = \sum_{j \in S} v_i(j)$ for a profile of valuations $v_i = (v_1, ..., v_m)$.let bidder i's valuation function is V than allocation rules would be $g_i : V \rightarrow 2^M$ payment rule would be $p_i : V \rightarrow R \geq 0$ .Given bids $b = (b_1, ..., b_n) \in V$, auction computes an allocation g(b) and payments p(b). A bidder with valuation v_i receives a utility $u_i(v_i, b) = v_i(g_i(b)) - p_i(b)$ for reporting a bid profile b.

3.5.2 Loss function

They adopt the minimization of negated, expected revenue on valuations as loss function.

$$\min \left(- \sum_{i \in N} P_i^w(v) \right) \tag{6}$$

To ensure that chosen auction satisfies incentive compatibility they have fixed the bids of others that called ex post regret $rgt_i(w)$. And they found that regret is non-negative, an auction satisfies dominant strategy incentive compatible (DSIC) if and only if $rgt_i(w) = 0, \forall i \in N$.

They re-formulate the learning problem as minimizing the expected loss, i.e., the expected negated revenue s.t. the expected ex post regret being 0 for each bidder:

$$\min \left(- \sum_{i \in N} P_i^w(v) \right) \text{ s.t. } rgt_i(w) = 0, \forall i \in N. \tag{7}$$

- They adopt the Augmented Lagrangian Method to solve the resulting constrained optimization problem, where in each iteration they push gradients through the regret term.

3.5.3 Neural Network Architecture

Additive Valuations: A bidder has additive valuations if the bidder's value for a bundle of items $S \subseteq M$ is the sum of her value for the individual items in S, i.e. $v_i(S) = \sum_{j \in S} v_i(j)$. In this case, the bidders report only their valuations for individual items. The architecture for allocation network $g^w : R^{nm} \to [0,1]^{nm}$ and a payment network $p^w : R^{nm} \to R^n \geq 0$, both are modeled with tanh activations. The input layer of the networks consists of bids b_{ij} representing the valuation of bidder i for item j. The allocation network outputs a vector of allocation probabilities $z_{1j} = g_{1j}(b), ..., z_{nj} = g_{nj}(b)$, for each item j $\in$[m]. Allocations are computed using a softmax activation function, so that for all items j,$\sum_{i=1}^{n} z_{ij} \leq 1$.

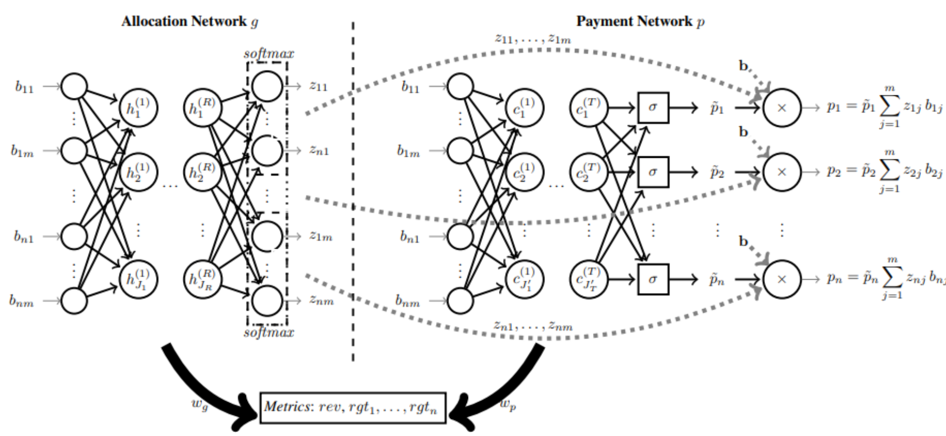


Figure 1: The allocation and payment networks for a setting with n additive bidders and m items.

Unit-demand valuations: A bidder has unit-demand valuations when the bidder's value for a bundle of items S $\subseteq$ M is the maximum value she assigns to any one item in the bundle, i.e. $v_i(S) = max_{j \in S} v_i(j)$. Also the total allocation for each bidder is at most 1, i.e. $\sum_j z_{ij} \leq 1, \forall i \in [n]$. It would also require that no item is over-allocated, i.e. $\sum_i z_{ij} \leq 1, \forall j \in [m]$. Hence, it design allocation networks for which the matrix of output probabilities $[z_{ij}]_{i,j=1}^{n}$ is doubly stochastic.

In particular, it has the allocation network compute two sets of scores s_{ij}'s and s'_{ij}'s, both normalization can be performed by passing these scores through softmax functions. The allocation for bidder i and item j is then computed as the minimum of the corresponding normalized scores:

$$Z_{ij} = \varphi_{ij}^{DS}(s, s') = min\left\{ \frac{e^{S_{ij}}}{\sum_{k=1}^{n+1} e^{S_{kj}}}, \frac{e^{S'_{ij}}}{\sum_{k=1}^{m+1} e^{S'_{jk}}} \right\}, \tag{8}$$

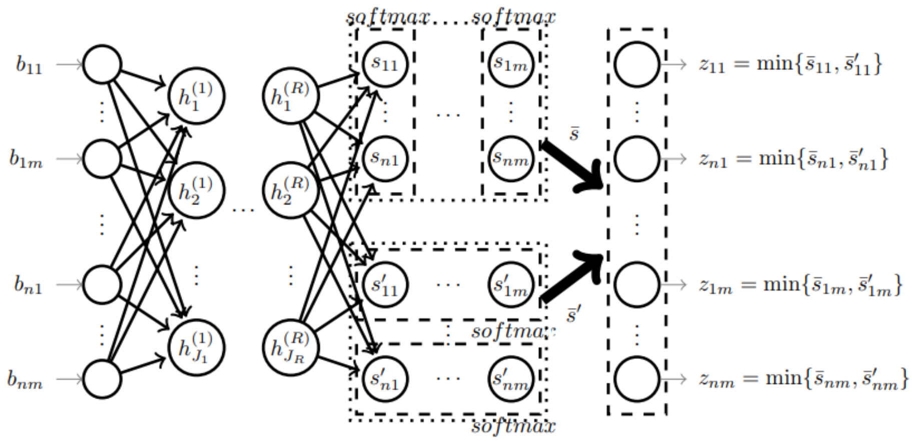


Figure 2: n unit-demand bidders and m items

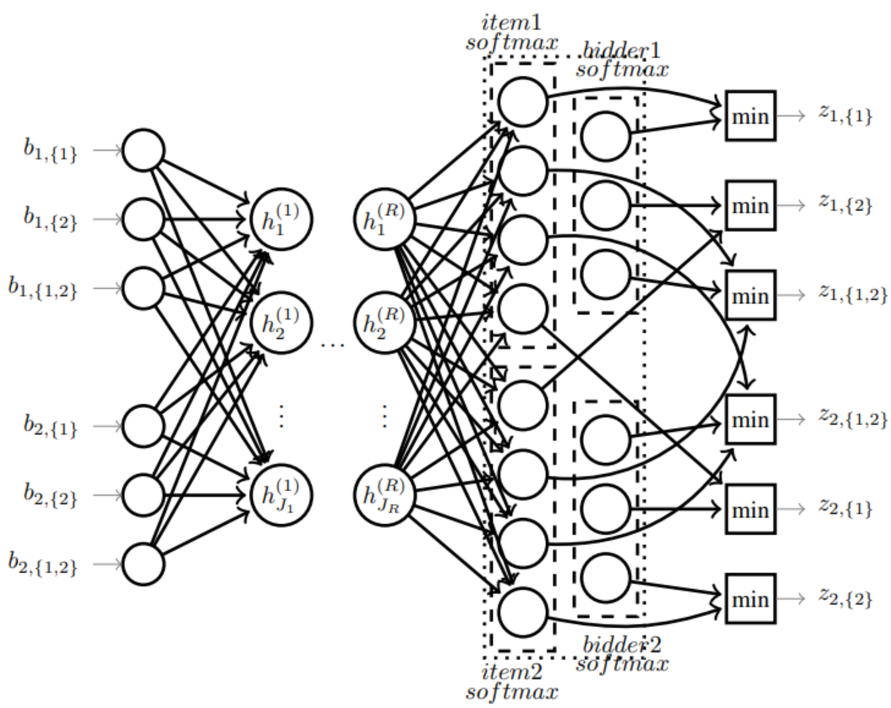


Figure 3: 2 combinatorial bidders and 2 items

Combinatorial valuations: In this case, each bidder i reports a bid $b_{i,S}$ for every bundle of items $S \subseteq M$. The allocation network has an output $z_{i,S} \in [0, 1]$ for each bidder i and bundle S, denoting the probability that the bidder is allocated the bundle. They also prevent the items from being over-allocated, item appears in bundle allocated to some bidder is at most 1 by using same constraints from architecture of unit-demand valuations. The allocation for bidder i and bundle $S \subseteq M$ is defined as the minimum of the normalized bidder-wise score $\overline{s}_{i,S}$ for i and the normalized item-wise scores $\overline{s}^{(j)}_{i,S}$ for each $j \in S$:

$$z_{i,S} = \varphi^{CF}_{i,S}(s, s^{(1)}, ..., s^{(m)}) = \min\{\overline{s}_{i,S}, \overline{s}^{(j)}_{i,S} : j \in S\} \tag{9}$$

3.5.4 Conclusion

Neural networks have been deployed successfully for exploration. It believe that there is ample opportunity for applying deep learning in the context of economic design.

3.6 Certifying Strategy-proof Auction Networks

3.6.1 Introduction & Background

Auctions play a crucial role in allocating goods efficiently, with significant applications in various domains such as online advertising, spectrum auctions, and more. However, designing auctions that maximize revenue while ensuring strategic agents truthfully reveal their private valuations remains a challenging task. Myerson [1981] provided insights into revenue-maximizing strategyproof auctions for single-item scenarios, but extending these principles to multiple items and agents remains an open problem. Recent advancements in "differentiable economics" leverage deep learning techniques to learn auction mechanisms, with the RegretNet [Dütting et al., 2022] architecture emerging as a promising approach. However, ensuring the strategyproofness of these learned mechanisms is not rigorously verified. Curry et al. [2020] proposes techniques to certify the strategyproofness of auction mechanisms under specific valuation profiles using methods from neural network verification literature.

The paper introduces the problem of automated mechanism design and describes the RegretNet approach for learning auction mechanisms. It also discusses neural network verification techniques adapted for the auction setting. RegretNet parameterizes mechanisms as neural networks and trains them to minimize regret between strategic and truthful bids, resembling adversarial training [Madry et al., 2018]. The goal is to approximate strategyproofness, but the efficacy of this approach remains uncertain due to potential suboptimal stationary points in neural network training [Athalye et al., 2018].

3.6.2 Experiments

Experiments are conducted on two auction settings: 1 agent, 2 items, and 2 agents, 2 items. Various network architectures are trained and evaluated for regret, revenue, and solve time. Results indicate that certified regrets tend to reveal strategic behaviors missed by gradient-based methods. Trained revenues exceed baseline benchmarks, and empirical evidence supports the effectiveness of individual rationality penalties. Solve time analysis highlights challenges in scalability, particularly with increasing numbers of items.

3.6.3 Conclusion & Future Work

The paper presents a method for certifying strategyproofness in learned auction mechanisms, paving the way for more rigorous auction design. Future work involves improving scalability and performance of learned mechanisms, potentially through advanced training algorithms and architectural modifications. Additionally, integrating certifiability techniques with existing generalization bounds and estimating expected strategyproofness could further enhance confidence in learned auction mechanisms.

3.6.4 Broader Impact

While the immediate social impact of this work may be limited, the broader project of strategyproof mechanism design holds promise for promoting fairness and accessibility in economic mechanisms. Ensuring strategyproofness can alleviate burdens on participants and prevent exploitation by sophisticated agents. By contributing to the development of strategyproof mechanisms, this work aligns with broader goals of enhancing fairness and accessibility in economic systems.

3.7 A Permutation-Equivariant Neural Network Architecture For Auction Design

3.7.1 Introduction & Background

Designing an incentive compatible auction that maximizes expected revenue is a central problem in Auction Design. Theoretical approaches to the problem have hit limits in the past decades and analytical solutions are known for only a few simple settings. Building on the success of deep learning, a new approach was recently proposed by [Dütting et al., 2022] (RegretNet), in which the auction is modeled by a feed-forward neural network and the design problem is framed as a learning problem. The neural architectures used in that work are general purpose and do not take advantage of any of the symmetries the problem could present, such as permutation equivariance.

The authors built a deep learning architecture for multi-bidder symmetric auctions. These are auctions which are invariant to relabeling the items or bidders, they are anonymous and item-symmetric.symmetric auctions are known to be optimal in many settings of interest (even those which are not themselves symmetric). Even in settings where they are not optimal, they are known to yield near-optimal auctions. And even when they are only approximately optimal, seminal work has identified them as important objects of study owing to their simplicity.

While applying existing feed-forward architectures as RegretNet to symmetric auctions is possible, RegretNet struggles to find symmetric auctions, even when the optimum is symmetric. The authors propose Equivariant.NetEquivariantNet is an adaption of the deep sets architecture [Jason Hartford and Ravanbakhsh., 2018] to symmetric auctions, that outputs symmetric auctions, with following three features:

Symmetry: This architecture outputs a symmetric auction by design. It is immune to permutation-sensitivity, related to fairness.

Sample generalization: Because we use domain knowledge, our architecture converges to the optimum with fewer valuation samples.

Out-of-setting generalization: the architecture does not require hard-coding the number of bidders or items during training - training on instances with n bidders and m items produces a well-defined auction even for instances with n' bidders and m' items.

3.7.2 Architecture

EquivariantNet learns symmetric auctions.EquivariantNet is build using exchangeable matrix layers [Jason Hartford and Ravanbakhsh., 2018].The input is a bid matrix $B = (b_{i,j}) \in R^{n\times m}$ drawn from a bidder-symmetric and item-symmetric distribution.

They aim at learning a randomized allocation neural network $g^w : R^{n\times m} \to [0,1]^{n\times m}$ and a payment network $p^w : R^{n\times m} \to R^n > 0$.It has three modules,

The first network outputs a vector $q^w(B) \in [0,1]^m$ such that entry $q^w_j(B)$ is the probability that item j is allocated to any of the n bidders.

The architecture consists of three modules. The first one is a deep permutation-equivariant network with tanh activation functions. The output of that module is a matrix $Q \in R^{n\times m}$.

The second module transforms Q into a vector R^m by taking the average over the rows of Q. We finally apply the sigmoid function to the result to ensure that $q^w(B) \in [0,1]^m$. This architecture ensures that $q^w(B)$ is invariant with respect to bidder permutations and equivariant with respect to items permutations.

The second network outputs a matrix $h(B) \in [0,1]^{n\times m}$ where $h^w_{i,j}$ is the probability that item j is allocated to bidder i conditioned on item j being allocated. The architecture consists of a deep permutation-equivariant network with tanh activation functions followed by softmax activation function so that $\sum_{i=1}^{n} h^w_{i,j}(B) = 1$.

By combining the outputs of q^w and h^w, we compute the allocation function $g^w : R^{n\times m} \to [0,1]^{n\times m}$ where $g_{ij}(B)$ is the probability that the allocated item j is given to bidder i.

The third network outputs a vector $p(B) \in R^n > 0$ where ~p_i^w is the fraction of bidder's i utility that she has to pay to the mechanism. Given the allocation function g^w, bidder i has to pay an amount $pi = \widetilde{p}_i(B) \sum_{j=1}^{m} g_{i,j}^w(B) B_{ij}$

3.7.3 Training-Loss and Optimization

Similarly to Dütting et al. [2022], the authors formulate auction design as a learning problem. The empirical ex-post regret for bidder i is the maximum increase in his utility when considering all his possible bids and fixing the bids of others.

For a valuation profile V, the ex-post regret for a bidder i is -

$$\hat{rgt_i}(w) = \frac{1}{L} \sum_{l=1}^{L} max_{\vec{v_i}' \in R^m} u_i^w(v_i^{\vec{(l)}}; (\widetilde{\vec{v_i}}, V_{-i}^{(l)})) - u_i^w(v_i^{\vec{(l)}}; (\vec{v_i}, V_{-i}^{(l)})) \tag{10}$$

and the learning formulation is -

$$min_{w \in R^d} - \frac{1}{L} \sum_{l=1}^{L} \sum_{i=1}^{n} p_i^w(V^{(l)}) \text{ s.t. } r\hat{g}t_i(w) = 0 \forall i \in N \tag{11}$$

The optimization and training procedure of Equivariant Net is similar to Dütting et al. [2022],where they apply the augmented Lagrangian method to the loss function.

The Lagrangian with a quadratic penalty is -

$$\mathcal{L}_p(w; \lambda) = -\frac{1}{L} \sum_{l=1}^{L} \sum_{i=1}^{n} p_i^w(V^{(l)}) + \sum_{i \in N} \lambda_i r\hat{g}t_i(w) + \frac{\rho}{2} \sum_{i \in N} (r\hat{g}t_i(w))^2 \tag{12}$$

where $\lambda \in R^n$ is a vector of Lagrange multipliers and $\rho > 0$ is a fixed parameter controlling the weight of the quadratic penalty. The solver alternates between the updates on model parameters and Lagrange multipliers.

4 Methodology Under Study: Deep Learning for Two Sided Matching

This is the main paper that has been assigned to us to study, having explored the works that motivated this and having analysed several necessary classical results and various NN architectures targetted a this problem, we now move on to the analysis of this paper.

Ravindranath et al. [2023] explored randomized matching mechanisms, for which the strongest SP concept is Ordinal SP, equivalent to FOSD. They formulated the two-sided matching problem as a deep learning task. The Neural Network (NN) function, $g^\theta : P \to \Delta(\mathcal{B})$, parameterized by θ, maps preference profile $\succ$ to matching distribution r.

The paper represented the ordinal preference order as a vector with constant offset utility since NN can't process ordinal preferences directly. Furthermore, they identified suitable, differentiable surrogates for approximate SP and stability. The NN's final output, $r \in [0, 1]^{n \times m}$, represents marginal probabilities in a randomized matching for the input profile. Notably, $r_{wf} = 0$ only when the match is unacceptable and this is realized using a boolean mask β.

4.1 Introduction

Two-sided matching markets, classically used for settings such as high-school matching, medical residents matching, and law clerk matching, and more recently used in online platforms such as Uber, Lyft, Airbnb, and dating apps, play a significant role in today's world.As a result, there is a significant interest in designing better mechanisms for two-sided matching.

The seminal work of Gale and Shapley [1962] introduces a simple mechanism for stable, one-to-one matching in two-sided markets—deferred-acceptance (DA). The DA mechanism is stable, however, the DA mechanism is not strategy-proof (SP), and a participant can sometimes misreport their preferences to obtain a better outcome (although it is SP for participants on one side of the market).

In general, it is well-known that there must necessarily be a trade-off between stability and strategy-proofness, it is provably impossible for a mechanism to achieve both stability and strategy-proofness Dubins and Freedman [1981], Roth [1982].

A second example of a matching mechanism is random serial dictatorship (RSD) Abdulkadiroglu et al. [2006], which is typically adopted for one-sided assignment problems rather than two-sided matching. When adapted to two-sided matching, RSD is SP but not stable. In fact, a participant may even prefer to remain unmatched than participate in the outcome of the matching.

A third example of a matching mechanism is the top trading cycles (TTC) mechanism Shapley and Scarf [1974], also typically adopted for one-sided assignment problems rather than problems of two-sided matching. In application to two-sided matching, TTC is neither SP nor stable (although it is SP for participants on one side of the market).

This paper initiates the study of Deep Learning Techniques for two sided matching, a key area of interest is to study the possibility of new trade-offs between strategy-proofness and stability.

It is known that these cannot be achieved simultaneously, but the efficient frontier is not understood, the authors introduce novel surrogates for quantifying these.

They work with randomized matching mechanisms, for which the strongest SP concept is ordinal strategy-proofness.This aligns incentives with truthful reporting, whatever an agent's utility function. Ordinal SP is equivalent to the property of first-order stochastic dominance (FOSD) Erdil [2014], which suitably defines the property that an agent has a better chance of getting their top, top-two, top-three, and so forth choices when they report truthfully.

As a surrogate for SP, they quantify during training the degree to which FOSD is violated, they also define a suitable surrogate to quantify the degree to which stability is violated. This surrogate aligns with the notion of ex ante stability—the strongest stability concept for randomized matching.

These surrogates are then used to train the neural networks for the task.They show that the efficient frontier characterized by these mechanisms is substantially better than what is obtained by a convex combination of the classical baseline techniques of deferred acceptance, randomized serial dictatorship and top trading cycles.

The authors use neural networks to represent the rules of a matching mechanism, mapping preference reports to a distribution over feasible matchings, and show how unsupervised learning can be used, to pipeline characterization the efficient frontier for the design trade-off between stability and SP. For this they adopt an adversarial learning approach, augmenting the data with defeating misreports that violate first order stochastic dominance (FOSD) equivalent to SP in this setting, suitable surrogates to quantify the degree of stability violation are also defined which align with ex-ante stability.

4.2 Setup

Let W denote a set of n workers and F denote a set of m firms. A feasible matching, μ, is a set of (worker, firm) pairs, with each worker and firm participating in at most one match.

Let B denote the set of all matchings. If $(w, f) \in \mu$, then μ matches w to f, and we write $\mu(w) = f$ and $\mu(f) = w$. If a worker or firm remains unmatched, we say it is matched to $\emptyset$.

We write $(w, \perp) \in \mu$ (resp. $(\perp, f) \in \mu$). Each worker has a strict preference order, $\succ_w$, over the set $F = F \cup \{\perp\}$. Each firm has a strict preference order, $\succ_f$, over the set $W = W \cup \{\perp\}$.

Worker w (firm f) prefers remaining unmatched to being matched with a firm (worker) ranked below $\perp$ (the agents ranked below $\perp$ are said to be unacceptable). If worker w prefers firm f to f', then we write $f \succ_w f'$, similarly for a firm's preferences.

Let P denote the set of all preference profiles, with $\succ = (\succ_1, ..., \succ_n, \succ_{n+1}, ..., \succ_{n+m}) \in P$ denoting a preference profile comprising of the preference order of the n workers and then the m firms.

A pair (w, f) forms a blocking pair for matching μ if w and f prefer each other to their partners in μ (or μ in the case that one or both are unmatched).

A matching μ is stable if and only if there are no blocking pairs.

A matching μ is individually rational (IR) if and only if it is not blocked by any individual; i.e., no agent finds its match unacceptable and prefers $\perp$

The authors work with randomized matching mechanisms, g, that map preference profiles, $\perp$, to distributions on matchings, denoted $g(\perp) \in \triangle(B)$ (the probability simplex on matchings). Let $r \in [0,1]^{(n+1)\times(m+1)}$denote the marginal probability, $r_{wf} \geq 0$, with which worker w is matched with firm f , for each $w \in W$ and $f \in F$.

4.3 Architecture

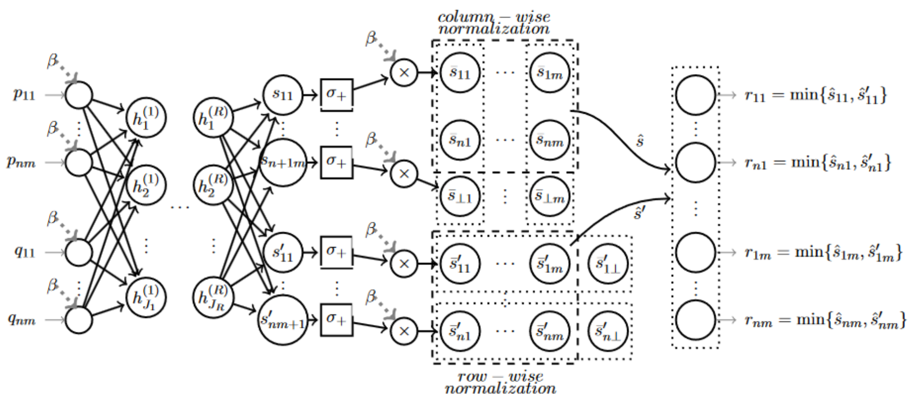


Figure 4: Matching network g for a set of n workers and m firms. Given inputs $p, q \in R^{n\times m}$ the matching network is a feed-forward network with R hidden layers that uses softplus activation to generate non-negative scores and normalization to compute the randomized matching. We additionally generate a Boolean mask matrix, β, and multiply it with the score matrix before normalization to ensure IR by making the probability of matches that are unacceptable zero.

Here a neural network is used to represent a matching mechanism. Let $g^\theta : P \to \triangle(B)$ denote the mechanism, for parameters $\theta \in R^d$. The input is a preference profile, and the output defines a distribution on matchings. We use a feed-forward neural network with R = 4 fully connected hidden layers, J = 256 units in each layer, leaky ReLU activations, and a fully connected output layer. To represent an agent's preference order in the input, we adopt a utility for each agent on the other side of the market that has a constant offset in utility across successive agents in the preference order.

4.4 Loss-Function and Optimization

For a mechanism parameterized as g^θ, the training problem formulated is -

$$min_\theta \lambda \times stv(g^\theta) + (1-\lambda) \times rgt(g^\theta) \tag{13}$$

where $\lambda \in [0,1]$ controls the trade-off between approximate stability and approximate SP, $stv(.)$ and $rgt(.)$ are the expected stability violation and regret respectively.

4.5 Welfare

The welfare of a learnt mechanism g on profile $\succ$ can be defined as -

$$welfare(g, \succ) = \frac{1}{2}\left(\frac{1}{n} + \frac{1}{m}\right) \sum_{w\in W} \sum_{f\in F} g_{wf}\left(p^{\succ}_{wf} + q^{\succ}_{wf}\right) \tag{14}$$

5 Experiment Details

Ravindranath et al. [2023] had trained an NN model for different values of correlation between the preferences of the agents $c \in \{0.00, 0.25, 0.50, 0.75\}$ and for different values of lambda (see 13) $\lambda \in \{0.0, 0.1, ..., 0.9, 1.0\}$ and compared the results with those of convex combinations of DA-RSD, DA-TTC and TTC-RSD.

Ravindranath et al. [2023] trained each NN model for 50,000 epochs with learning rate starting at 0.005 and halving it every 10,000 and 25,000 epochs. Authors used Adam optimizer to minimize this loss. Experimental results are as shown in the following plots -

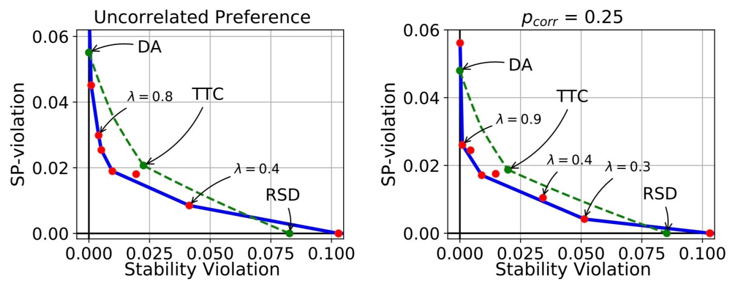


Figure 5: Comparing Stability violations and Strategy-proofness violations of the learnt mechanisms with DA, TTC and RSD for different values of trade-off parameter lambda λ.

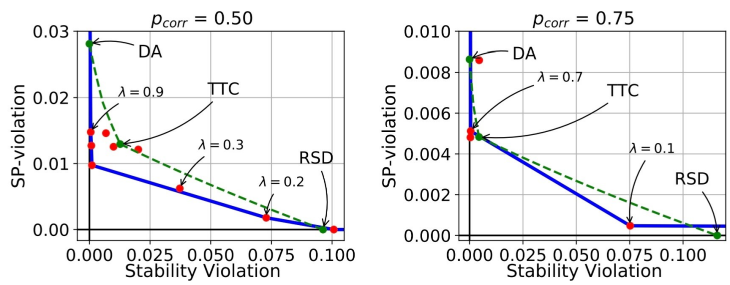


Figure 6: Comparing Stability violations and Strategy-proofness violations of the learnt mechanisms with DA, TTC and RSD for different values of trade-off parameter lambda λ.

For $\lambda = 0$, the learnt mechanisms' performance closely resembles RSD with very low regret but poor stability. As λ increases, the learnt mechanism approximates DA. At intermediate λ values, it outperforms convex combinations of DA-RSD, DA-TTC and TTC-RSD.

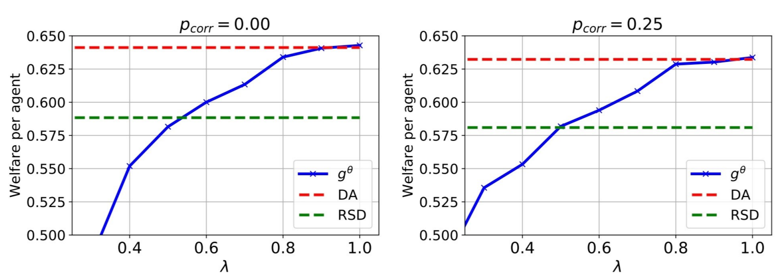


Figure 7: Comparing Welfare per agent of the learnt mechanisms with DA, TTC and RSD for different values of trade-off parameter lambda λ

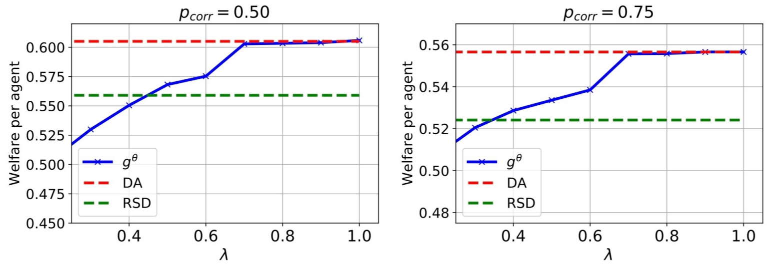


Figure 8: Comparing Welfare per agent of the learnt mechanisms with DA, TTC and RSD for different values of trade-off parameter lambda λ

Note : As TTC and RSD algorithms do not guarantee IR, the IR violations had been added to the Stability violations of these algorithms' results. With this being said, no other learnt mechanism or DA fails IR.

6 Extending the work

We are able to extend the work in a novel way. Details are in Appendix along with further possible future work details.

References

Atila Abdulkadiroglu, Parag Pathak, Alvin E Roth, and Tayfun Sonmez. Changing the boston school choice mechanism. Working Paper 11965, National Bureau of Economic Research, January 2006.

Atila Abdulkadiroğlu, Parag A. Pathak, and Alvin E. Roth. The new york city high school match. *American Economic Review*, 95(2):364–367, May 2005. doi: 10.1257/000282805774670167. URL `https://www.aeaweb.org/articles?id=10.1257/000282805774670167`.

Atila Abdulkadiroğlu, Parag A. Pathak, and Alvin E. Roth. Strategy-proofness versus efficiency in matching with indifferences: Redesigning the nyc high school match. *The American Economic Review*, 99(5):1954–1978, 2009. ISSN 00028282, 19447981. URL `http://www.jstor.org/stable/25592543`.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, 2018.

Michael J. Curry, Ping-Yeh Chiang, Tom Goldstein, and John Dickerson. Certifying strategyproof auction networks, 2020.

L E Dubins and D A Freedman. Machiavelli and the gale-shapley algorithm. *Am. Math. Mon.*, 88(7): 485, August 1981.

Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning. In *International Conference on Machine Learning*, pages 1706–1715. PMLR, 2019.

Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David C. Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning: Advances in differentiable economics, 2022.

A. Erdil. strategy-proof stochastic assignment. *Journal of Economic Theory*, 151:146–162, 2014.

Aytek Erdil and Haluk Ergin. What's the matter with tie-breaking? improving efficiency in school choice. *The American Economic Review*, 98(3):669–689, 2008. ISSN 00028282. URL `http://www.jstor.org/stable/29730091`.

D. Gale and L.S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962. doi: https://doi.org/10.2307/2312726.

Kevin Leyton-Brown Jason Hartford, Devon R Graham and Siamak Ravanbakhsh. Deep models of interactions across sets. *arXiv preprint arXiv:1803.02879*, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL `https://openreview.net/forum?id=rJzIBfZAb`.

Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981. ISSN 0364765X, 15265471. URL `http://www.jstor.org/stable/3689266`.

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

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

Alvin E Roth. The economics of matching: Stability and incentives. *Math. Oper. Res.*, 7(4):617–628, November 1982.

Lloyd Shapley and Herbert Scarf. On cores and indivisibility. *Journal of Mathematical Economics*, 1 (1):23–37, 1974. ISSN 0304-4068. doi: https://doi.org/10.1016/0304-4068(74)90033-0.

Weiran Shen, Pingzhong Tang, and Song Zuo. Automated mechanism design via neural networks, 2021.

A Notations

$W = \{w_i\}_{i=1}^n$ is the set of n workers and $F = \{f_j\}_{j=1}^m$ is the set of m firms.

(w_i, f_j) is a match i.e. worker w_i is matched with firm f_j and vice versa. Furthermore, if a worker or firm is unmatched, we say it is matched with $\perp$.

μ is a feasible matching, $\mu = \{(w_i, f_j)_{w_i \in \overline{W}, f_j \in \overline{F}}\}$, such that each worker and each firm occurs in a match exactly once and $\overline{W} = W \cup \{\perp\}, \overline{F} = F \cup \{\perp\}$.

If $(w_i, f_j) \in \mu$, then we write $\mu(w_i) = f_j, \mu(f_j) = w_i$.

$\mathcal{B} = \{\mu\}$ is the set of all possible matching.

Each worker (firm) has a strict preference order, $\succ_w$ $(\succ_f)$, over $\overline{W}(\overline{F})$.

If a worker (firm) prefers firm f over f' (worker w over w'), then we write $f \succ_w f'$ $(w \succ_f w')$.

Given a strict preference order, $\succ_w$ $(\succ_f)$, of worker w (firm m), all firms f (workers w) such that $f \succ_w \perp$ $(w \succ_f \perp)$ are called acceptable firms (workers).

$\succ = \{\succ_1, ..., \succ_n, \succ_{n+1}, ..., \succ_{n+m}\}$ is a preference profile consisting of the preference orders of n workers followed by the preference orders of m firms.

$P = \{\succ\}$ is the set of all preference profiles.

B Stability Violation (STV())

$stv(g^\theta, \succ)$ is given by -

$$stv(g^\theta, \succ) = \frac{1}{2}\left(\frac{1}{m} + \frac{1}{n}\right) \sum_{w=1}^{n} \sum_{f=1}^{m} stv_{wf}(g^\theta, \succ)$$

where $stv_{wf}(g^\theta, \succ)$ is given by -

$$stv_{wf}(g^\theta, \succ) = \left(\sum_{w' \in \overline{W}} g^\theta_{w'f} * max(q^\succ_{wf} - q^\succ_{w'f}, 0) \right) * \left(\sum_{f' \in \overline{F}} g^\theta_{wf'} * max(p^\succ_{wf} - p^\succ_{wf'}, 0) \right)$$

where $p^\succ_{wf}, p^\succ_{wf'}, q^\succ_{wf}, q^\succ_{w'f}$ are the input representations of the preference orders.

C Ordinal SP Violation (RGT())

$regret(g^\theta, \succ)$ is given by -

$$regret(g^\theta, \succ) = \frac{1}{2}\left(\frac{1}{m} * \sum_{w \in W} regret_w(g^\theta, \succ) + \frac{1}{n} * \sum_{f \in F} regret_f(g^\theta, \succ) \right)$$

where $regret_w(g^\theta, \succ)$ is given by -

$$regret_w(g^\theta, \succ) = \max_{\succ'_w \in P} \left(\max_{f' \succ_w \perp} \sum_{f \succ_w f'} \Delta_{wf}(g^\theta, \succ'_w, \succ) \right)$$

and $regret_f(g^\theta, \succ)$ is given by -

$$regret_f(g^\theta, \succ) = \max_{\succ'_f \in P} \left(\max_{w' \succ_f \perp} \sum_{w \succ_f w'} \Delta_{wf}(g^\theta, \succ'_f, \succ) \right)$$

where $\Delta_{wf}(g^\theta, \succ'_f, \succ)$ is given by -

$$\Delta_{wf}(g^\theta, \succ'_f, \succ) = g^\theta_{wf}(\succ'_w, \succ_{-w}) - g^\theta_{wf}(\succ_w, \succ_{-w})$$

D Extended Experiment Details

Due to limitations of computational resources we have taken a smaller set of values for lambda, $\lambda \in \{0.00, 0.25, 0.50, 0.75, 1.00\}$.

While trying to reproduce the results, we realized that the models trained over different correlation values seemed to perform relatively well and out performed the classical algorithms' results on the test data having different correlation values than those on which they were originally trained. For example, the models that is trained on preferences with correlation $c = 0.50$ and lambda value $\lambda = 0.5$ and $\lambda = 0.75$ outperformed the results of convex combinations of DA-TTC, TTC-RSD and DA-RSD when all are tested on preference data with no correlation (or $c = 0.0$). See figure 9.

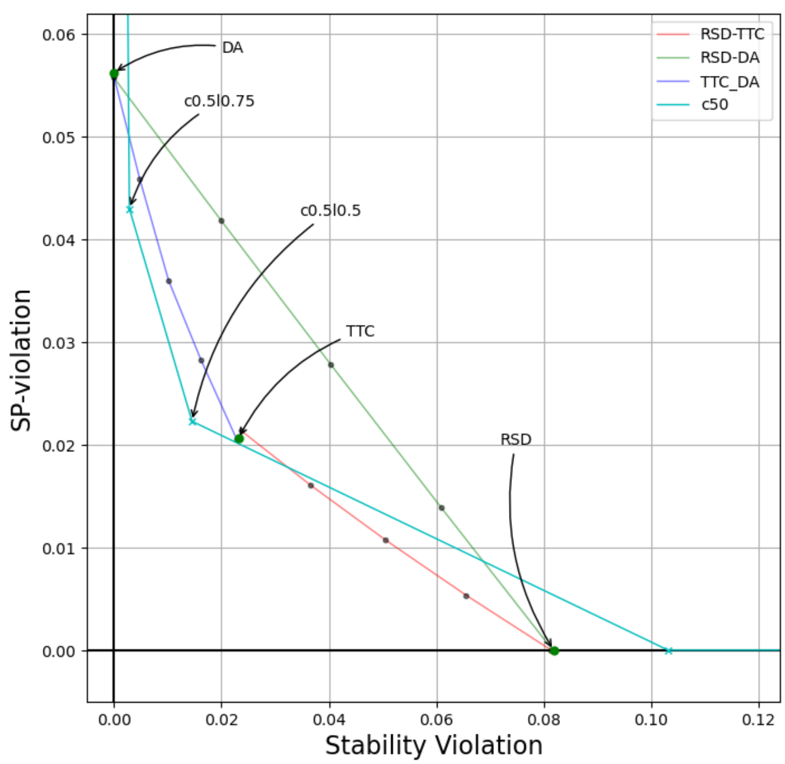


Figure 9: Models $c_{0.5}l_{0.5}$ and $c_{0.5}l_{0.75}$ giving better results on test data preferences with correlation $c = 0.0$ as compared to classical algorithms.

Following are some notations defined to make further discussion concise and less complicated -

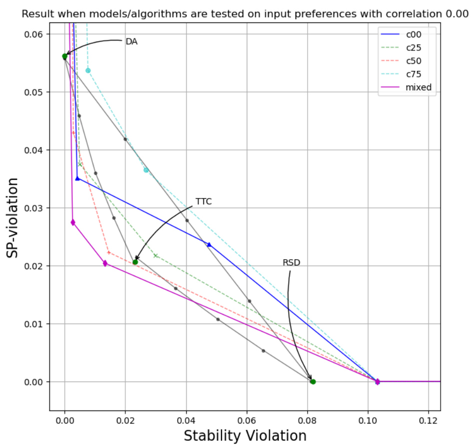
- Let $c_x l_y$ denote an NN model with lambda $\lambda = y$ trained on data containing preferences with correlation $c = x$.
- Let ml_y denote an NN model with lambda $\lambda = y$ trained on data containing preferences with correlation $c \in \{0.00, 0.25, 0.50, 0.75\}$.

While for a given lambda value $\lambda = y$ the loss function that we are optimizing is same for all the 4 models trained in Ravindranath et al. [2023], just the training data differs so we theorized that an NN model ml_y might work at least as good as an NN model $c_x l_y$ on test data containing preferences with correlation $c = x$ and somewhat better than NN models $c_{x'} l_y$ on test data containing preferences with correlation $c = x$. Furthermore, we would be training less number of models for the same task which greatly reduces the need of computational power and complexity.

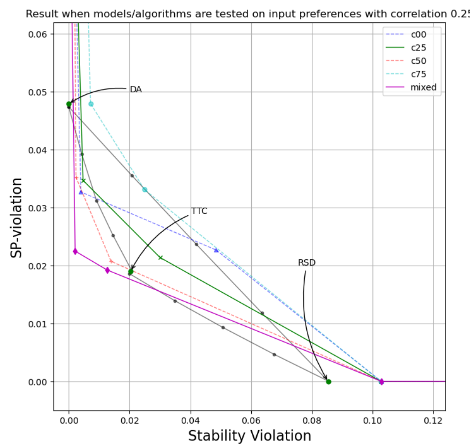
We employed identical methods for generating synthetic training and test datasets. Additionally, our NN models are optimized for minimizing the same loss function across both datasets.

We trained all NN models for 2,000 epochs with learning rate set to 0.01 to quickly reduce the loss initially, followed by 5,000 epochs with learning rate set to 0.001 followed by another 3,000 epochs with learning rate set to 0.0005 to gradually and steadily reduce the loss. We decided to utilize the Adam optimizer given its superior performance. Although we could not reproduce the exact results, which is quite obvious given the lesser number of epochs, we did manage to beat the classical algorithms for some values of lambda.

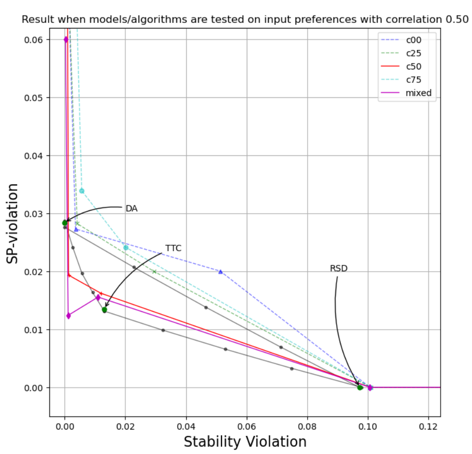
Experimental results are as shown in the following plots -



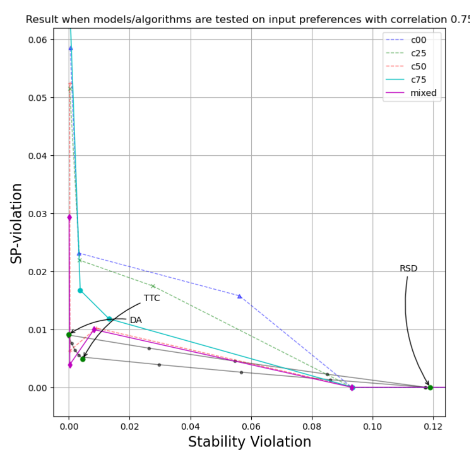
Comparing Stability violations and Strategy-proofness violations of all mechanisms with DA, TTC, RSD and their convex combinations for different values of trade-off parameter lambda (λ), computed on data containing preferences with correlation $c = 0.00$.



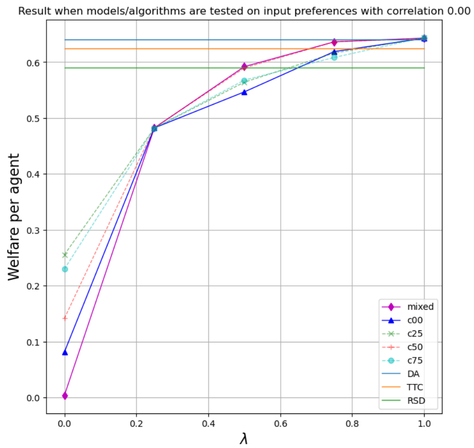
Comparing Stability violations and Strategy-proofness violations of all mechanisms with DA, TTC, RSD and their convex combinations for different values of trade-off parameter lambda (λ), computed on data containing preferences with correlation $c = 0.25$



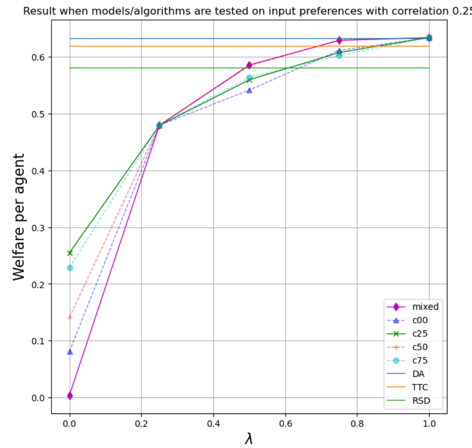
Comparing Stability violations and Strategy-proofness violations of all mechanisms with DA, TTC, RSD and their convex combinations for different values of trade-off parameter lambda (λ), computed on data containing preferences with correlation $c = 0.50$



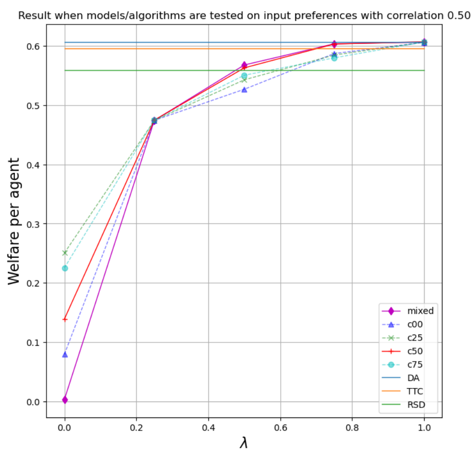
Comparing Stability violations and Strategy-proofness violations of all mechanisms with DA, TTC, RSD and their convex combinations for different values of trade-off parameter lambda (λ), computed on data containing preferences with correlation $c = 0.75$



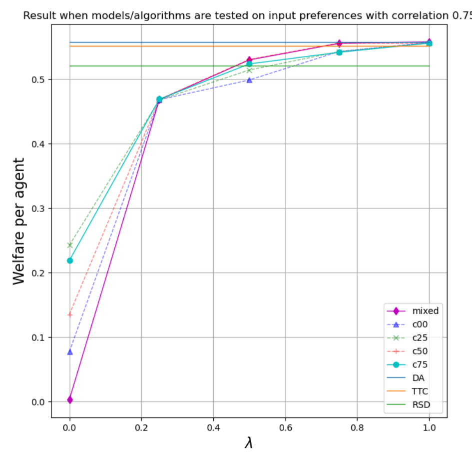
Comparing welfare per agent for all mechanisms, computed on data containing preferences with correlation $c = 0.00$.



Comparing welfare per agent for all mechanisms, computed on data containing preferences with correlation $c = 0.25$.



Comparing welfare per agent for all mechanisms, computed on data containing preferences with correlation $c = 0.50$.



Comparing welfare per agent for all mechanisms, computed on data containing preferences with correlation $c = 0.75$.

The source codes for all experiments along with the results are available on GitHub, click here to visit the repository.

E Future Work

While our work builds on the work done by Ravindranath et al. [2023], many more interesting experiments can be performed to further improve on this. Some of which could be -

- Rather than training on data containing preferences with a small set of different correlation values, one can train a model on data containing preferences with all possible correlation values.
- A model trained for n agents (on both sides) can in theory also work for m_1, m_2 agents where $m_1, m_2 \leq n$ if we introduce $n - m_1, n - m_2$ pseudo agents on both sides who always like to remain unmatched, provided it is trained using task-relevant preference data.