Machine Learning-Aided Economic Design

HARVARD UNIVERSITY

Graduate School of Arts and Sciences

DISSERTATION ACCEPTANCE CERTIFICATE

The undersigned, appointed by the

Harvard John A. Paulson School of Engineering and Applied Sciences
have examined a dissertation entitled:

"Machine Learning-Aided Economic Design"

presented by: Zhe Feng

Signature ______________________________

Typed name: Professor D. Parkes

Signature ______________________________

Typed name: Professor Y. Chen

Signature ______________________________

Typed name: Professor S. Kominers

Signature Vincent Conitzer ______________________________

Typed name: Professor V. Conitzer

April 28, 2021

Machine Learning-Aided Economic Design

A dissertation presented

by

Zhe Feng

to

John A. Paulson School of Engineering and Applied Sciences

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Computer Science

Harvard University

Cambridge, Massachusetts

April 2021

Dissertation Advisor:
Prof. David C. Parkes

Author:
Zhe Feng

Machine Learning-Aided Economic Design

Abstract

Nowadays, online markets (e.g. online advertising market and online two-sided markets) grow larger and larger everyday. Designing an efficient and near-optimal market is an intricate task. Market designers are facing challenges not only in regard to scalability, but also coming from the use of data to better understand the behavior of strategic participants. At the same time, these participants are trying to understand how these markets work and to maximize reward. For these reasons, we continue to need improved frameworks for the design of online markets. One challenge for market design is to make effective use of data in order to design better markets. For the players, a central problem is how to optimize their strategy, adaptively learning from feedback and incorporating this along with other side information.

To handle these challenges, my thesis focuses on two topics, *Economic Design via Machine Learning* and *Learning in Online Markets*. For the first topic, I propose a unified computational framework for data-driven mechanism design that can help a mechanism designer to automatically design a good mechanism to satisfy incentive constraints and achieve a desired objective (e.g. revenue, social welfare). I provide different approaches to guarantee Incentive Compatibility and prove the generalization bounds. This deep-learning framework is very general and can be extended to handle other constraints, e.g., private budget constraints. In addition, I investigate how to transform an approximately incentive compatible mechanism to a fully BIC mechanism without loss of welfare and with only negligible loss of revenue. For the second topic, I analyze the convergence of the outcome achieved by strategic bidders when they adopt mean-based learning algorithms to bid in repeated auctions. I also propose a new online learning algorithm for a bidder to use when bidding in repeated auctions, where the bidder's own value, evolving in an arbitrary manner, and observed only if the bidder wins an auction. This algorithm has exponentially faster convergence in terms of its dependence on the action space than the generic bandit algorithm.

Contents

Abstract . . . iii
Acknowledgments . . . ix
List of Tables . . . x
List of Figures . . . xi
List of Algorithms . . . xv
Citation to Previously Published Work . . . xvi

Introduction **1**

I Deep Learning for Auction Design **7**

1 Optimal Auctions through Deep Learning **8**
1.1 Introduction . . . 8
1.1.1 Our Contribution . . . 9
1.1.2 Discussion . . . 10
1.1.3 Further Related Work . . . 11
1.1.4 Organization . . . 11
1.2 Auction Design as a Learning Problem . . . 11
1.2.1 Auction Design Basics . . . 11
1.2.2 Formulation as a Learning Problem . . . 13
1.2.3 Quantile-Based Regret . . . 16
1.2.4 Generalization Bound . . . 16
1.3 Neural Network Architecture . . . 17
1.3.1 The RochetNet Architecture . . . 18
1.3.2 The RegretNet Architecture . . . 20
1.3.3 Covering Number Bounds . . . 23
1.4 Optimization and Training . . . 24
1.5 Experiments . . . 26
1.5.1 Setup . . . 26
1.5.2 Evaluation . . . 27
1.5.3 The Manelli-Vincent and Pavlov Auctions . . . 27

1.5.4 The Straight-Jacket Auction 30
1.5.5 Discovering New Optimal Designs 31
1.5.6 Gradient-Based Regret Approximation 33
1.5.7 Scaling Up 34
1.5.8 Comparison to LP 34
1.6 Conclusion 36

II Extensions of RegretNet **37**

2 Deep Learning for Revenue-Optimal Auctions with Budgets **38**
2.1 Introduction 38
2.1.1 Main Contributions 39
2.1.2 Related Work 40
2.2 Problem Setup 41
2.2.1 Single-item auctions 41
2.2.2 Multi-item auctions 44
2.3 The Budgeted RegretNet Framework 45
2.3.1 Network architecture 46
2.3.2 Training problem 47
2.3.3 Handling other kinds of IC constraints 50
2.4 Experimental Results 51
2.4.1 Optimal DSIC auctions 52
2.4.2 Optimal BIC auctions 55
2.5 Conclusion 56

3 Welfare-Preserving ε-BIC to BIC Transformation with Negligible Revenue Loss **62**
3.1 Introduction 62
3.2 Warm-up: Single agent with Uniform Type Distribution 75
3.2.1 Lower Bound on Revenue Loss 77
3.2.2 Tighter Bound of Revenue Loss for Settings with Finite Menus 78
3.3 Single Agent with General Type Distribution 78
3.3.1 ε-BIC to BIC Transformation 79
3.3.2 Impossibility Result for ε-EEIC Transformation 79
3.4 Multiple Agents with Independent Private Types 80
3.4.1 Impossibility Results 81
3.5 Application to Automated Mechanism Design 81
3.6 Conclusion and Open Questions 83

III Learning to Bid in Repeated Auctions **85**

4 Convergence Analysis of No-Regret Bidding Algorithms in Repeated Auctions **86**
4.1 Introduction . . . 86
4.1.1 Our Results . . . 89
4.1.2 Our Techniques . . . 90
4.1.3 Related Works . . . 91
4.2 Model and Notations . . . 92
4.2.1 Learning Algorithms of Mean-Based Bidders . . . 93
4.3 Second Price Auctions with Mean-based Bidders . . . 94
4.3.1 Proof of Theorem 4.1 . . . 95
4.4 Generalizations to Other Auctions . . . 98
4.4.1 First Price Auctions . . . 98
4.4.2 Multi-Position VCG Auctions . . . 99
4.5 Experiments . . . 100

5 Learning to Bid Without Knowing your Value **103**
5.1 Introduction . . . 103
5.2 Learning in Auctions without Knowing your Value . . . 109
5.3 Abstraction: Learning with Win-Only Feedback . . . 110
5.4 Beyond Binary Outcomes: Outcome-Based Feedback . . . 113
5.4.1 Batch Rewards Per-Iteration and Sponsored Search Auctions . . . 115
5.5 Continuous Actions with Piecewise-Lipschitz Rewards . . . 118
5.5.1 Sponsored Search with Lipschitz Utilities . . . 120
5.6 Further Extensions . . . 121
5.6.1 Switching Regret and Implications for Price of Anarchy . . . 121
5.6.2 Feedback Graphs over Outcomes . . . 122
5.7 Experimental Results . . . 123
5.8 Discussion . . . 128

6 Conclusion **129**

References **132**

Appendix A Appendix to Chapter 1 **152**
A.1 Additional Architectures . . . 152
A.1.1 The MyersonNet Approach . . . 152
A.1.2 RegretNet for Combinatorial Valuations . . . 156
A.2 Additional Experiments . . . 159
A.2.1 Experiments with MyersonNet . . . 159

A.2.2 Additional Experiments with RochetNet and RegretNet 160
A.2.3 Experiments with RegretNet with Combinatorial Valuations 161
A.3 Omitted Proofs 163
A.3.1 Proof of Lemma 1.1 163
A.3.2 Proof of Theorem 1.2 164
A.3.3 Proof of Theorem 1.3 170
A.3.4 Proof of Lemma 1.2 171
A.3.5 Proof of Lemma A.1 171
A.3.6 Proof of Theorem 1.4 171
A.3.7 Proof of Theorem 1.5 178

Appendix B Appendix to Chapter 3 **183**
B.1 Details of Replica-Surrogate Mechanism 183
B.2 Omitted Properties of Our Transformation 183
B.3 Omitted Definitions 185
B.4 Omitted Proofs 186
B.4.1 Proof of Claim 3.2 186
B.4.2 Proof of Claim 3.3 186
B.4.3 Proof of Theorem 3.5 187
B.4.4 Proof of Theorem 3.6 188
B.4.5 Proof of Theorem 3.7 191
B.4.6 Proof of Theorem 3.8 191
B.4.7 Proof of Theorem 3.9 194
B.4.8 Proof of Theorem 3.10 194
B.4.9 Proof of Theorem 3.11 195
B.5 Omitted Details of Applications 196
B.5.1 LP-based Approach 196
B.5.2 RegretNet Approach 196

Appendix C Appendix to Chapter 4 **198**
C.1 Missing Proofs 198
C.1.1 Proof of Lemma 4.1 198
C.1.2 Proof of Lemma 4.2 199
C.1.3 Proof of Lemma 4.3 199
C.1.4 Proof of Claim 4.3 200
C.1.5 Proof of Theorem 4.4 200
C.1.6 Proof of Theorem 4.5 205
C.2 Additional Experiments 206
C.2.1 Experiment Details 207

Appendix D Appendix to Chapter 5 **209**
D.1 Omitted Algorithms 209
D.1.1 Outcome-based feedback graph over outcomes 210
D.2 Omitted proofs from Section 5.4 210
D.2.1 Comparison with Results in [WPR16]. 211
D.3 Notes on Subsection 5.4.1 212
D.4 Omitted Proofs from Section 5.4.1 213
D.5 Omitted Proofs from Section 5.5 214
D.5.1 Omitted Proofs from Section 5.5.1 216
D.6 Omitted proofs from section 5.6.1 217
D.6.1 Switching Regret and PoA 217
D.6.2 Feedback Graphs over Outcomes 219
D.7 Omitted proof for the regret of the exponential weights update 222

Acknowledgments

I would not have been able to write this thesis without the constant support and guidance of my amazing advisor, David C. Parkes. It is extraordinary how much David has influence my research interests while still encouraging me to explore new research areas. David is always being available when I asked for help in my research and life. It was my honor to be advised by David during my PhD journey and I will be forever grateful for his support.

I would also give my special thanks to Eric Sodomka, Jinchao Ye, Sébastien Lahaie, Ashwinkumar Badanidiyuru who hosting me for three summers at Facebook Research and Google Research. Their support and kindness made me decide to continue my research at industry after graduation. Thanks to the other members of thesis committee, Prof. Yiling Chen, Prof. Vincent Conitzer, and Prof. Scott Kominers for many helpful discussions and valuable comments. A huge thanks to Google for the PhD fellowship support.

I was fortunately to be able to work with many excellent researchers when pursuing my PhD, Paul Dütting, Guru Guruganesh, Chris Liaw, Aranyak Mehta, Harikrishna Narasimhan, Chara Podimata, Sai S. Ravindranath, Jon Schneider, Okke Schrijvers, Abhishek Sethi, Vasilis Syrgkanis, and Haifeng Xu. I extremely enjoyed our discussions and hope to collaborate more in the future. I am happy for having a warmful community of graduate students at EconCS group, particularly, Hongyao, Debmalya, Paul, Chara, Juntao, Sophie, and Sai. I really miss the life that we can hangout together in person. Especially, I would like to thank Debmalya for sharing his thesis latex template with me to make my life much easier.

I would also like to thank my undergraduate supervisor Prof. Xiaotie Deng, who taught me Game Theory and encouraged me to pursue research along with this research direction. I had the fortune of discussing problems with other excellent researchers throughout this journey: Santiago Balseiro, Yuan Deng, Yannai A. Gonczarowski, Nicole Immorlica, Yifeng Teng, Di Wang, Muhan Zhang and Mingfei Zhao.

Finally, I would like to give my great thanks to my girlfriend, Zihuan Qiao, and my parents, Shubo Feng and Ruilin Zhang, for believing me that I could finish a PhD and supporting me all the way. I would not have been able to finish my PhD without their support.

List of Tables

2.1 Test metrics for Budgeted RegretNet auctions. Here (C) refers to conditional IC. For continuous valuation distributions, we also report within parenthesis the regret estimated using a larger misreport sample (i.e. with 1000 misreports for each type profile). 53

2.2 Test regret for Budgeted RegretNet under Setting IV with misreport samples of different sizes for each type profile. 53

List of Figures

1.1 RochetNet: (a) Neural network representation of a non-negative, monotone, convex induced utility function; here $h_j(b) = \alpha_j \cdot b + \beta_j$ for $b \in \mathbb{R}^m$ and $\alpha_j \in [0,1]^m$. (b) An example of a utility function represented by RochetNet for one item. 18

1.2 RegretNet: The allocation and payment networks for a setting with n additive bidders and m items. The inputs are bids from each bidder for each item. The revenue rev and expected ex post rgt_i are defined as a function of the parameters of the allocation and payment networks $w = (w_g, w_p)$. 21

1.3 RegretNet: The allocation network for settings with n *unit-demand* bidders and m items. 22

1.4 Side-by-side comparison of allocation rules learned by RochetNet and RegretNet for single bidder, two items settings. Panels (a) and (b) are for Setting A and Panels (c) and (d) are for Setting B. The panels describe the probability that the bidder is allocated item 1 (left) and item 2 (right) for different valuation inputs. The optimal auctions are described by the regions separated by the dashed black lines, with the numbers in black the optimal probability of allocation in the region. 28

1.5 (a): Test revenue and regret for RegretNet and revenue for RochetNet for Settings A and B. (b): Plot of test revenue and regret as a function of training epochs for Setting A with RegretNet. 29

1.6 Revenue of the Straight-Jacket Auction (SJA) computed via the recursive formula in [GK18], and that of the auction learned by RochetNet, for various numbers of items m. The SJA is known to be optimal for up to six items, and conjectured to be optimal for any number of items. 30

1.7 Revenue of the newly discovered optimal mechanism and that of RochetNet, for Setting C with varying parameter c. 32

1.8 Allocation rules learned by RochetNet for Setting C. The panels describe the probability that the bidder is allocated item 1 (left) and item 2 (right) for $c = 0.5, 1, 3$, and 5. The auctions proposed in Theorem 1.5 are described by the regions separated by the dashed black lines, with the numbers in black the optimal probability of allocation in the region. 32

1.9 Visualization of the gradient-based approach to regret approximation for a well-trained auction for Setting A. The top left figure shows the true valuation (green dot) and ten random initial misreports (red dots). The remaining figures display 20 steps of gradient descent, showing one in every four steps. 33
1.10 (a) Revenue and regret of RegretNet on the validation set for auctions learned for Setting D using different architectures, where (R, K) denotes R hidden layers and K nodes per layer. (b) Test revenue and regret for Settings D and E, for the (5, 100) architecture. 35
1.11 Test revenue, regret, IR violation, and running-time for RegretNet and an LP-based approach for a two bidder, three items setting with additive uniform valuations. 35

2.1 Budgeted RegretNet: (a) Allocation rule a and (b) Payment rule p for a setting with m *identical* items and n *additive* buyers. 45
2.2 Budgeted RegretNet: (a) Allocation rule a and (b) Payment rule p for a setting with m *distinct* items and n *unit-demand* buyers. 45
2.3 The auction learned under DSIC for Setting I with a single item and single bidder, where $v_1 \sim Unif[0,1]$ and $b_1 \sim Unif[0,1]$. The solid regions in (b) and (c) depict the probability of the item being allocated to the bidder. 54
2.4 The auction learned under conditional DSIC for Setting II with a single item and two bidders, where $v_1, v_2 \sim \text{Unif}\{1, 2, ..., 10\}$, bidder 1 is unconstrained, and bidder 2 has a budget of 4. 58
2.5 Revenue and regret for the DSIC auction learned under Setting III with four identical items and two additive bidders, where bidder i's value for each item $v_i \sim Unif[0,1]$ and $b_i \sim Unif[0,1]$. 59
2.6 Revenue and regret for the DSIC auction learned under Setting IV with two items and two unit-demand bidders, where bidder i's value for item j $v_{ij} \sim \text{Unif}[0,1]$ and $b_i \sim \text{Unif}[0,1]$. 59
2.7 A semi-logarithmic plot of test regret as a function of the number of iterations for different misreport sample sizes for the DSIC auction learned under Setting IV. 59
2.8 Auction learned under BIC for Setting II with a single item and two bidders, where $v_1, v_2 \sim \text{Unif}\{1, 2, ..., 10\}$, bidder 1 is unconstrained and bidder 2 has a budget of 4. 60
2.9 Revenue and regret of auction learned under BIC for Setting V with a single item and two bidders, where $v_1, v_2 \sim \text{Unif}[0,1]$ and $b_1, b_2 \sim \text{Unif}\{0.22, 0.42\}$. . . . 61

3.1 Visualization of the transformation for a single agent with a uniform type distribution: we start from a type graph $G(\mathcal{T}, E)$, where each edge $(t^{(1)}, t^{(2)})$ represents the agent weakly prefers the allocation and payment of type $t^{(2)}$ rather than his true type $t^{(1)}$. The weight of each edge is denoted in Eq. (3.4). In the graph, we use solid lines to represent the positive-weight edges, and dashed lines to represent zero-weight edges. We first find a shortest cycle, and rotate the allocation and payment along the cycle and update the graph (Step 1). We keep doing Step 1 to remove all cycles. Then we pick a source node $t^{(1)}$, and decrease the payment of type $t^{(1)}$ and all the ancestors of $t^{(1)}$ until we reduce the weight of one outgoing edge from $t^{(1)}$ to zero or we create a new zero-weight edge from t' to $t^{(1)}$ or one of the ancestors of $t^{(1)}$ (Step 2). 74
3.2 ε-BIC/ε-EEIC to BIC transformation for single agent with uniform type distribution 76

4.1 Simulation results for second price auctions 100
4.2 Simulation Results for first price auctions 101

5.1 Example interfaces of bid simulators of two major search engines, Google Adwords (left) and BingAds (right), that enables learning the allocation and the payment function. (sources [Sta14; Lan14]) 106
5.2 Regret of WIN-EXP vs EXP3 for different discretizations ϵ (CTR $\sim U[0.5, 1]$). . . 125
5.3 Regret of WIN-EXP vs EXP3 for different CTR distributions and stochastic adversaries, $\varepsilon = 0.001$. 125
5.4 Regret of WIN-EXP vs EXP3 for different CTR distributions and adaptive EXP3 adversaries, $\varepsilon = 0.001$. 126
5.5 Regret of WIN-EXP vs EXP3 for different CTR distributions and adaptive WIN-EXP adversaries, $\varepsilon = 0.001$. 126
5.6 Regret of WIN-EXP vs EXP3 with noise $\sim \mathcal{N}\left(0, \frac{1}{m}\right)$ for stochastic adversaries, $\varepsilon = 0.001$. 127
5.7 Regret of WIN-EXP vs EXP3 with noise $\sim \mathcal{N}\left(0, \frac{1}{m}\right)$ for adaptive EXP3 adversaries, $\varepsilon = 0.001$. 127
5.8 Regret of WIN-EXP vs EXP3 with noise $\sim \mathcal{N}\left(0, \frac{1}{m}\right)$ for adaptive WINEXP adversaries, $\varepsilon = 0.001$. 128

A.1 (a) MyersonNet: The network applies monotone transformations $\bar{\phi}_1, \ldots, \bar{\phi}_n$ to the input bids, passes the virtual values to the SPA-0 network in Figure A.2, and applies the inverse transformations $\bar{\phi}_1^{-1}, \ldots, \bar{\phi}_n^{-1}$ to the payment outputs. (b) Monotone virtual value function $\bar{\phi}_i$, where $h_{kj}(b_i) = e^{\alpha_{kj}^i} b_i + \beta_{kj}^i$. 154

A.2 MyersonNet: SPA-0 network for (approximately) modeling a second price auction with zero reserve price. The inputs are (virtual) bids $\bar{b}_1, \ldots, \bar{b}_n$ and the output is a vector of assignment probabilities $z_1, \ldots, z_n$ and prices (conditioned on allocation) $t^0_1, \ldots, t^0_n$. 155
A.3 The revenue of the single-item auctions obtained with MyersonNet. 160
A.4 Side-by-side comparison of the allocation rules learned by RochetNet and RegretNet for single bidder, two items settings. Panels (a) and (b) are for Setting J, Panels (c) and (d) are for Setting K, and Panels (e) and (f) for Setting L. Panels describe the learned allocations for the two items (item 1 on the left, item 2 on the right). Optimal mechanisms are indicated via dashed lines and allocation probabilities in each region. 161
A.5 Test revenue and regret achieved by RegretNet and revenue achieved by RochetNet for Settings J–L. 162
A.6 Test revenue and regret for RegretNet for Settings M–O, and comparison with the best performing VVCA and AMA_{bsym} auctions as reported by Sandholm and Likhodedov [SL15]. 163
A.7 The transport of transformed measure of each region in Setting C. 179

B.1 Replica-Surrogate Matching Mechanism. 184
B.2 ε-BIC to BIC transformation for single agent with general type distribution. 190

C.1 Training curve of mean reward of each bidder (left) and roll-out bidding strategy of each bidder (right) in the exploitation phase of contextual ε-Greedy algorithm in first price auctions for three bidders. 207
C.2 The roll-out of the optimal bidding strategy of each bidder with deep state representations in second price auction (left) and first price auction (right). 208

List of Algorithms

1 RegretNet Training 25

2 Mean-based (Contextual) Learning Algorithm of Bidder i 94

3 WIN-EXP algorithm for learning with win-only feedback 111

4 WIN-EXP algorithm for learning with outcome-based feedback 114

5 WIN-EXP algorithm for learning with outcome-based batch-reward feedback 116

6 WIN-EXP-G algorithm for learning with outcome-based feedback and a feedback graph over outcomes 210

Citation to Previously Published Work

1. A significant part of the work presented in Chapter 1 is based on the following publication:

   Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David C. Parkes and Sai S. Ravindranath. "Optimal Auctions through Deep Learning". In *Proceedings of the 36th International Conference on Machine Learning (ICML-19)*, PMLR 97:1706-1715, 2019.

2. The work presented in Chapter 2 is based on the following publication:

   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)*, pp. 354-362, 2018.

3. The work presented in Chapter 3 is based on the following paper, which is presently posted in ArXiv:

   Vincent Conitzer, Zhe Feng, David C. Parkes, and Eric Sodomka. "Welfare-Preserving ε-BIC to BIC Transformation with Negligible Revenue Loss". In *CoRR*, ArXiv:2007.09579, 2020.

4. The work presented in Chapter 4 is based on the following publication:

   Zhe Feng, Guru Guruganesh, Christopher Liaw, Aranyak Mehta, and Abhishek Sethi. "Convergence Analysis of No-Regret Bidding Algorithms in Repeated Auctions". In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI-21)*, to appear, 2021.

5. The work presented in Chapter 5 is based on the following publication:

   Zhe Feng, Chara Podimata, and Vasilis Syrgkanis. "Learning to Bid Without Knowing Your Value". In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC-18)*, pp. 505-522, 2018.

   This work may also appear in my collaborator, Chara Podimata's, dissertation.

Introduction

Nowadays, online markets (e.g. online advertising market and online two-sided markets) grow larger and larger everyday. Designing an efficient and near-optimal market is an intricate task. Market designers are facing challenges not only in regard to scalability, but also from how to use data to better understand the behavior of strategic players in the market. At the same time, participants are trying to understand how these markets work and to maximize their own reward. For these reasons, we continue to need improved frameworks for their design.

One challenge for market design is in regard to how to utilize data in order to design better markets. I think about this as *data-driven market design*. For the players, a central problem is how to optimize their strategy, adaptively learning from feedback while also incorporating this with other side information. These concerns motivate the following fast developing research areas: automated mechanism design [CS02], learning in online markets [Blu+04], and statistical machine learning in economics and social science, e.g. [Ath18].

This thesis is motivated by the goal of using machine learning for market design and mechanism design, and the recognition that there is an absence today of a unified approach to data-driven market design. The main research questions I investigate in this thesis are in the following,

1. **Economic design via machine learning:** *Can machine learning solve problems of optimal economic design that are hard to theoretically analyze?*

2. **Learning in online markets:** *Can we optimize the performance of online markets through the design of the learning algorithms that are used by both users and market platform operators?*

For the first research question, the main challenge in applying machine learning to economic design comes from the need to handle the strategic behavior of market participants. For instance, a revenue-optimal auction needs to satisfy incentive constraints (e.g. incentive compatibility and individual rationality). To handle this challenge, I provide a unified computational framework for data-driven mechanism design that can help the mechanism designer to automatically design a mechanism to satisfy incentive constraints and achieve a desired objective (e.g. revenue, social welfare). The general computational framework is named as *RegretNet* (Chapter 1), and uses a multi-layer neural network to flexibly represent mechanisms and access stochastic gradient descent based optimization. I generalize RegretNet to handle other constraints in auction design with financially constrained buyers and handled BIC in Chapter 2. I also propose a way to transform an approximately incentive compatible mechanism to a fully Bayesian incentive compatible mechanism without loss of welfare and with only negligible loss of revenue in Chapter 3. The RegretNet framework is a general and powerful tool to support economists in discovering new theory and help practitioners to optimize target objectives such as revenue empirically.

For the second research question, I focus on the bidding algorithms of advertisers in repeated auctions. No-regret learning algorithms are commonly used in practice for advertisers to decide their bids in repeated auctions [NST15], however, the convergence of the no-regret learning algorithms in repeated auctions is not well understood. No-regret learning algorithms are known to converge to a *coarse correlated equilibrium* (CCE) in repeated general-sum games in a time-average sense [BM07]. But it was not known which specific CCE no-regret learning algorithms converge to in repeated auctions. In Chapter 4, I show that mean-based learning algorithms (a board class of no-regret learning algorithms) converge to a natural Bayesian Nash Equilibrium in repeated single-item second price auctions, repeated single-item first price auctions and repeated multi-position VCG auctions in a last-iterate sense. I also investigate how to design an efficient learning algorithm for bidders in repeated auctions when they don't know their valuation before bidding. This is presented in Chapter 5.

My thesis is divided into three parts.

Part I: Deep Learning for Auction Design

Optimal economic design, especially optimal auction design, is one of the cornerstones of economic theory and has also received a lot of attention in computer science in recent years. The most important question is that of designing a protocol for selling one or more items in order to maximize expected revenue. Myerson's seminal work [Mye81] solved the problem for a single item setting. Although there are several algorithmic advances in the design of optimal, Bayesian incentive compatible (BIC) multi-item auctions [Das15; CDW12a; CDW12b; CDW13; CZ17], the design of optimal, dominant strategy incentive compatible (DSIC) mechanisms has remained open. Progress here is important, because the equilibrium concept in DSIC auctions requires much weaker assumptions than that of BIC auctions.

Automated Mechanism Design (AMD), initiated by Conitzer and Sandholm [CS02], utilizes a computational approach, for example integer programming or linear programming, to automatically design optimal mechanisms. Recently, data-driven automated mechanism design has also started to gain attention in the machine learning literature [CR14; NAP16; BSV18; Alb+21], where we can optimize the parameterized mechanism from i.i.d. sampled valuation data.

Following the research line of data-driven automated mechanism design, and based on the rich representation and approximation to nonlinear function of deep neural networks, I propose the use of feed-forward multi-layer neural networks to approximately model optimal auctions in Chapter 1. I introduce three different architectures: *MyersonNet*, *RochetNet*, and *RegretNet*. The first two networks use characterization results from economic theory to guarantee DSIC, but are inflexible in that they can only handle single-item auctions and single-bidder auctions, respectively. In contrast, RegretNet is a general approach that uses negated, expected revenue as the loss function. Crucially, we must achieve incentive compatibility. RegretNet is trained subject to a constraint that the expected ex post regret to bidders for bidding truthfully is zero. This implies DSIC up to measure zero types. RegretNet can handle multi-item and multi-bidder settings. I have developed an *augmented Lagrangian* methodology to train RegretNet subject to the requirement of very low expected ex post regret. Simulations show that RegretNet can learn almost-optimal, almost incentive compatible auctions for essentially all settings for which there are known analytical solutions, and obtain novel mechanisms for settings in which the optimal mechanism is unknown. In addition, I have developed *generalization results* for both the expected revenue and regret in RegretNet. This work has opened an area of economic design through deep learning and inspired a lot of follow-up work, e.g., Shen, Tang, and Zuo [STZ19], Rahme et al. [Rah+20], Rahme, Jelassi, and Weinberg [RJW20], and Curry et al. [Cur+20].

Part II: Extensions of RegretNet

The RegretNet framework is very flexible, and I develop it for settings in which bidders are financially (budget) constrained. The problem of designing a revenue-maximizing auction for settings with private budgets is known to be very challenging. Even the single-item case is not fully understood, and there are no analytical results for optimal, DSIC, two-item auctions. In Chapter 2, I generalize the RegretNet architecture to handle private budget constraints, as well as to allow for BIC in order to facilitate comparisons with the existing, theoretical literature. I discover new auctions with high revenue for multi-unit auctions with private budgets, including settings with unit-demand bidders for which no analytical results are available. For benchmarking purposes, I also demonstrate that RegretNet can obtain essentially optimal designs for simpler settings where analytical solutions were available [CG00; MV08; PV14].

One weakness of RegretNet is that RegretNet outputs an approximately-IC mechanism (with tiny regret), which is not fully incentive compatible. A natural question is whether we can transform this approximately-IC mechanism to a fully IC mechanism without sacrificing the target objectives. To address this issue, in Chapter 3, I propose an approach to transform an approximately-IC mechanism into a fully BIC mechanism without any loss of welfare and with only negligible loss in revenue. I show that the revenue loss bound is tight given the requirement to maintain social welfare. This is the first ε-BIC to BIC transformation that preserves welfare while also providing negligible revenue loss. Previous ε-BIC to BIC transformations preserve social welfare but have no revenue guarantee [BH11], or suffer welfare loss while incurring a revenue loss with both a multiplicative and an additive term, e.g., Daskalakis and Weinberg [DW12], Cai and Zhao [CZ17], and Rubinstein and Weinberg [RW18]. The revenue loss achieved by our transformation is incomparable to these earlier approaches and can sometimes be significantly less. I also analyze ε-expected ex-post IC (ε-EEIC) mechanisms [Düt+14], i.e., the mechanism with the kinds of approximate guarantees provided by RegretNet and provide a welfare-preserving transformation with the same revenue loss guarantee for the special case of uniform type distributions. I give applications of this method to both linear-programming based and machine-learning based methods of automated mechanism design. I also show the impossibility of welfare-preserving, ε-EEIC to BIC transformations with negligible loss of revenue for non-uniform distributions.

Part III: Learning to Bid in Repeated Auctions

In Part III, I focus on *learning to bid in auctions* and introduce two contributions, *convergence analysis of no-regret bidding algorithms in repeated auctions* and *learning to bid without knowing your value*, which are presented in Chapter 4 and Chapter 5, respectively.

The connection between games and no-regret algorithms has been widely studied in the literature. A fundamental result is that when all players play no-regret strategies, this produces a sequence of actions whose time-average is a coarse-correlated equilibrium of the game, e.g., [FL98; CL06; Nis+07]. However, little is known about convergence to particular equilibria. In Chapter 4, I study the convergence of no-regret bidding algorithms in auctions. Besides being of theoretical interest, bidding dynamics in auctions is also an important question from a practical viewpoint. I study repeated games in which a single item is sold at each time step and each bidder's value is i.i.d. drawn from an underlying distribution. We show that if each bidder uses a mean-based learning rule then the bidders converge with high probability to the truthful, dominant strategy equilibrium in a second price single-item auction, to the truthful, dominant strategy equilibrium in the VCG auction in the multi-slot setting, and to the efficient Bayesian Nash equilibrium in a first price single-item auction. The mean-based algorithms cover a wide variety of known no-regret algorithms such as Exp3, UCB, and ε-Greedy. Also, I analyze the point-wise convergence in such learning algorithms, as opposed to the time-average of the sequence. I also present experiments that corroborate my theoretical findings and also find a similar convergence for other bidding strategies such as Deep Q-Learning.

In Chapter 5, I address online learning in complex auction settings, such as sponsored search auctions, where the bidder's own value is unknown, evolving in an arbitrary manner, and observed only if the bidder wins an allocation. I leverage the structure of the utility of the bidder and the partial feedback from the auction, in order to provide algorithms with regret rates against the best fixed bid in hindsight that are *exponentially faster* in convergence in terms of dependence on the action space than what would have been derived by applying a generic bandit algorithm, and almost equivalent to what would have been achieved in the full information setting. Our results are achieved by analyzing a new online learning setting with outcome-based feedback, which generalizes from learning with feedback graphs. I provide an online learning algorithm for this setting, of independent interest, with regret that grows logarithmically with the number of actions and linearly only in the number of potential outcomes (the latter being very small in most auction settings). In Section 5.7, I show that this algorithm outperforms the bandit approach experimentally and that this performance is robust to dropping some of the theoretical assumptions or introducing noise into the feedback that the player receives.

Part I

Deep Learning for Auction Design

Chapter 1

Optimal Auctions through Deep Learning

1.1 Introduction

Optimal auction design is one of the cornerstones of economic theory. It is of great practical importance, as auctions are used across industries and by the public sector to organize the sale of their products and services. Concrete examples are the US FCC Incentive Auction, the sponsored search auctions conducted by web search engines such as Google, and the auctions run on platforms such as eBay. In the standard *independent private valuations* model, each bidder has a valuation function over subsets of items, drawn independently from not necessarily identical distributions. It is assumed that the auctioneer knows the distributions and can (and will) use this information in designing the auction. A major difficulty in designing auctions is that valuations are private and bidders need to be incentivized to report their valuations truthfully. The goal is to learn an incentive compatible auction that maximizes revenue.

In a seminal piece of work, Myerson resolved the optimal auction design problem when there is a single item for sale [Mye81]. Quite astonishingly, even after 30-40 years of intense research, the problem is not completely resolved even for a simple setting with two bidders and two items. While there have been some elegant partial characterization results [MV06; Pav11; HH19; GK18; DDT17; Yao17], and an impressive sequence of recent algorithmic results

[CDW12a; CDW12b; CDW13; HN17; Bab+14; Yao15; CZ17; Cha+10], most of them apply to the weaker notion of Bayesian incentive compatibility (BIC). Our focus is on designing auctions that satisfy dominant-strategy incentive compatibility (DSIC), the more robust and desirable notion of incentive compatibility.

A recent, concurrent line of work started to bring in tools from machine learning and computational learning theory to design auctions from samples of bidder valuations. Much of the effort here has focused on analyzing the *sample complexity* of designing revenue-maximizing auctions [CR14; MM16; HMR18; MR15; GN17; MR16; Syr17; GW18; BSV16]. A handful of works has leveraged machine learning to optimize different aspects of mechanisms [Lah11; Düt+14; NAP16], but none of these offers the generality and flexibility of our approach. There have also been computational approaches to auction design, under the agenda of *automated mechanism design* [CS02; CS04; SL15], but where scalable, they are limited to specialized classes of auctions known to be incentive compatible.

1.1.1 Our Contribution

In this work we provide the first, general purpose, end-to-end approach for solving the multi-item auction design problem. We use multi-layer neural networks to encode auction mechanisms, with bidder valuations being the input and allocation and payment decisions being the output. We then train the networks using samples from the value distributions, so as to maximize expected revenue subject to constraints for incentive compatibility.

We propose two different approaches to handling IC constraints. In the first, we leverage characterization results for IC mechanisms, and constrain the network architecture appropriately. We specifically show how to exploit Rochet's characterization result for single-bidder multi-item settings [Roc87], which states that DSIC mechanisms induce Lipschitz, non-decreasing, and *convex* utility functions.

Our second approach, replaces the IC constraints with the goal of minimizing expected ex post regret, and then lifts the constraints into the objective via the *augmented Langrangian* method. We minimize a combination of negated revenue, and a penalty term for IC violations. This approach is also applicable in multi-bidder multi-item settings for which we don't have tractable characterizations of IC mechanisms, but will generally only find mechanisms that are approximately incentive compatible.

We show through extensive experiments that our two approaches are capable of recovering essentially all analytical results that have been obtained over the past 30-40 years, and that deep learning is also a powerful tool for confirming or refuting hypotheses concerning the form of optimal auctions and can be used to find new designs.

We also present *generalization bounds* in the style of machine learning that provide confidence intervals on the expected revenue and expected ex post regret based on the empirical revenue and empirical regret during training, the complexity of the neural network used to encode the allocation and payment rules, and the number of samples used to train the network.

1.1.2 Discussion

In general, the optimization problems we face may be non-convex, and so gradient-based approaches may get stuck in local optima. Empirically, however, this has not been an obstacle to deep nets in other problem domains, and there is growing theoretical evidence in support of this "no local optima" phenomenon (see, e.g., [CLA15; Kaw16; PNB16]).

By focusing on expected ex post regret we adopt a quantifiable relaxation of dominant-strategy incentive compatibility, first introduced in [Düt+14]. Our experiments suggest that this relaxation is an effective tool for approximating optimal DSIC auctions.

While not strictly limited to neural networks, our approach benefits from the expressive power of neural networks and the ability to enforce complex constraints using the standard pipeline. A key advantage of our method over other approaches to automated mechanism design such as [SL15] is that we optimize over a broad class of mechanisms, constrained only by the expressivity of the neural network architecture.

While the original work on automated auction design framed the problem as a linear program (LP) [CS02; CS04], follow-up work acknowledged that this has severe scalablility issues as it requires a number of constraints and variables that is exponential in the number of agents and items [GC10]. We find that even for small setting with 2 bidders and 3 items (and a discretization of the value into 5 bins per item) the corresponding LP takes 69 hours to complete since the LP needs to handle $\approx 10^5$ decision variables and $\approx 4 \times 10^6$ constraints. For the same setting, our approach found an auction with low regret in just over 9 hours (see Table 1.11). Our work shows that we are able to help economists to discover new theory (see Section 1.5.5) and help practitioners to maximize revenue empirically, by efficiently computing across a wide set of examples.

1.1.3 Further Related Work

Several other research groups have recently picked up deep nets and inference tools and applied them to economic problems, different from the one we consider here. These include the use of neural networks to predict behavior of human participants in strategic scenarios [HWLB16; FL19], an automated equilibrium analysis of mechanisms [TNLB17], deep nets for causal inference [Har+17; Lou+17], and deep reinforcement learning for solving combinatorial games [Rag+18].[1]

1.1.4 Organization

Section 1.2 formulates the auction design problem as a learning problem, describes our two basic approaches, and states the generalization bound. Section 1.3 presents the network architectures, and instantiates the generalization bound for these networks. Section 1.4 describes the training and optimization procedures, and Section 1.5 the experiments. Section 3.6 concludes.

1.2 Auction Design as a Learning Problem

1.2.1 Auction Design Basics

We consider a setting with a set of n bidders $N = \{1, \ldots, n\}$ and m items $M = \{1, \ldots, m\}$. Each bidder i has a valuation function $v_i : 2^M \to \mathbb{R}_{\geqslant 0}$, where $v_i(S)$ denotes how much the bidder values the subset of items $S \subseteq M$. In the simplest case, a bidder may have *additive* valuations.

[1] There has also been follow-up work to the present work that extends our approach to budget constrained bidders [FNP18] and to the facility location problem [GNP18], and that develops specialized architectures for single bidder settings that satisfy IC [STZ19] and for the purpose of minimizing agent payments [Tac+19]. A short survey also appears as a chapter in [Düt+19a].

In this case she has a value $v_i(\{j\})$ for each individual item $j \in M$, and her value for a subset of items $S \subseteq M$ is $v_i(S) = \sum_{j \in S} v_i(\{j\})$. If a bidder's value for a subset of items $S \subseteq M$ is $v_i(S) = \max_{j \in S} v_i(\{j\})$, we say this bidder has a *unit-demand* valuation. We also consider bidders with general combinatorial valuations, but defer the details to Appendix A.1.2.

Bidder i's valuation function is drawn independently from a distribution F_i over possible valuation functions V_i. We write $v = (v_1, \ldots, v_n)$ for a profile of valuations, and denote $V = \prod_{i=1}^{n} V_i$. The auctioneer knows the distributions $F = (F_1, \ldots, F_n)$, but does not know the bidders' realized valuation v. The bidders report their valuations (perhaps untruthfully), and an auction decides on an allocation of items to the bidders and charges a payment to them. We denote an auction (g, p) as a pair of allocation rules $g_i : V \to 2^M$ and payment rules $p_i : V \to \mathbb{R}_{\geqslant 0}$ (these rules can be randomized). Given bids $b = (b_1, \ldots, b_n) \in V$, the 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 a report of bid profile b. Let v_{-i} denote the valuation profile $v = (v_1, \ldots, v_n)$ without element v_i, similarly for b_{-i}, and let $V_{-i} = \prod_{j \neq i} V_j$ denote the possible valuation profiles of bidders other than bidder i. An auction is *dominant strategy incentive compatible* (DSIC) if each bidder's utility is maximized by reporting truthfully no matter what the other bidders report. In other words, $u_i(v_i; (v_i, b_{-i})) \geqslant u_i(v_i; (b_i, b_{-i}))$ for every bidder i, every valuation $v_i \in V_i$, every bid $b_i \in V_i$, and all bids $b_{-i} \in V_{-i}$ from others. An auction is ex post *individually rational* (IR) if each bidder receives a non-zero utility, i.e. $u_i(v_i; (v_i, b_{-i})) \geqslant 0 \ \forall i \in N$, $v_i \in V_i$, and $b_{-i} \in V_{-i}$.

In a DSIC auction, it is in the best interest of each bidder to report truthfully, and so the revenue on valuation profile v is $\sum_i p_i(v)$. Optimal auction design seeks to identify a DSIC auction that maximizes expected revenue. There is another weaker notion of incentive compatibility, *Bayesian Incentive Compatibility* (BIC) in the literature. An auction is BIC if each bidder's utility is maximized by reporting truthfully when the other bidders also report truthfully, i.e. $\mathbf{E}_{v_{-i}}[u_i(v_i; (v_i, v_{-i}))] \geqslant \mathbf{E}_{v_{-i}}[u_i(v_i; (b_i, v_{-i}))]$ for every bidder i, every valuation $v_i \in V_i$, every bid $b_i \in V_i$. In this work, we focus on DSIC auctions rather than BIC auctions, since DSIC auctions are more preferable in practice where the bidders need less prior information to bid.

1.2.2 Formulation as a Learning Problem

We pose the problem of optimal auction design as a learning problem, where in the place of a loss function that measures error against a target label, we adopt the negated, expected revenue on valuations drawn from F. We are given a parametric class of auctions, $(g^w, p^w) \in \mathcal{M}$, for parameters $w \in \mathbb{R}^d$ for some $d \in \mathbb{N}$, and a sample of bidder valuation profiles $\mathcal{S} = \{v^{(1)}, \ldots, v^{(L)}\}$ drawn i.i.d. from F.[2] The goal is to find an auction that minimizes the negated, expected revenue $-\mathbf{E}[\sum_{i \in N} p_i^w(v)]$, among all auctions in $\mathcal{M}$ that satisfy incentive compatibility.

We present two approaches for achieving IC. In the first, we leverage characterization results to constrain the search space so that all mechanisms within this class are IC. In the second, we replace the IC constraints with a differentiable approximation, and lift the constraints into the objective via the augmented Lagrangian method. The first approach affords a smaller search space and is exactly DSIC, but requires an IC characterization that can be encoded within a neural network architecture and applies to single-bidder multi-item settings. The second approach applies to multi-bidder multi-item settings and does not rely on the availability of suitable characterization results, but entails search through a larger parametric space and only achieves approximate IC.

Characterization-Based Approach

We begin by describing our first approach, in which we exploit characterizations of IC mechanisms to constrain the search space. We provide a construction for single-bidder multi-item settings based on Rochet [Roc87]'s characterization of IC mechanisms via induced utilities, which we refer to as *RochetNet*. For the single-bidder setting, there is no difference between DSIC and BIC constraints. We present this construction for additive preferences, but the construction can easily be extended to unit demand valuations. See Section 1.3.1. In Appendix A.1.1 we describe a second construction based on Myerson [Mye81]'s characterization result for single-bidder multi-item settings, which we refer to as *MyersonNet*.

To formally state Rochet's result we need the following notion of an induced utility function.

[2] There is no need to compute equilibrium inputs—we sample true profiles, and seek to learn rules that are IC.

The utility function $u : \mathbb{R}^m_{\geqslant 0} \to \mathbb{R}$ *induced* by a mechanism (g, p) for a single bidder with additive preferences is[3]:

$$u(v) = \sum_{j=1}^{m} g_j(v)\, v_j \;-\; p(v). \tag{1.1}$$

Rochet's result establishes the following connection between DSIC mechanisms and induced utility functions:

Theorem 1.1 (Rochet [Roc87]). *A utility function $u : \mathbb{R}^m_{\geqslant 0} \to \mathbb{R}$ is induced by a DSIC mechanism iff u is 1-Lipschitz w.r.t. the ℓ_1-norm, non-decreasing, and convex. Moreover, for such a utility function u, $\nabla u(v)$ exists almost everywhere in $\mathbb{R}^m_{\geqslant 0}$, and wherever it exists, $\nabla u(v)$ gives the allocation probabilities for valuation v and $\nabla u(v) \cdot v \;-\; u(v)$ is the corresponding payment.*

Further, for a mechanism to be ex post IR, its induced utility function must be non-negative, i.e. $u(v) \geqslant 0, \forall v \in \mathbb{R}^m_{\geqslant 0}$.

To find the optimal mechanism, it thus suffices to search over all non-negative utility functions that satisfy the conditions in Theorem 1.1, and pick the one that maximizes expected revenue.

This can be done by modeling the utility function as a neural network, and formulating the above optimization as a learning problem. The associated mechanism can then be recovered from the gradient of the learned neural network. We describe the neural network architectures for this approach in Section 1.3.1, and we present extensive experiments with this approach in Section 1.5 and Appendix A.2.

Characterization-Free Approach

Our second approach—which we refer to as *RegretNet*—does not rely on characterizations of IC mechanisms. Instead, it replaces the IC constraints with a differentiable approximation and lifts the IC constraints into the objective by augmenting the objective with a term that accounts for the extent to which the IC constraints are violated.

[3] For a unit-demand bidder, the utility can also be represented via (1.1) with the additional constraint that $\sum_j g_j(v) \leqslant 1, \forall v$. We discuss this more in Section 1.3.1.

We propose to measure the extent to which an auction violates incentive compatibility through the following notion of ex post regret. Fixing the bids of others, the ex post regret for a bidder is the maximum increase in her utility, considering all possible non-truthful bids. For mechanisms (g^w, p^w), we will be interested in the *expected ex post regret* for bidder i:

$$rgt_i(w) = \mathbf{E}\Big[\max_{v_i' \in V_i} u_i^w(v_i;(v_i', v_{-i})) - u_i^w(v_i;(v_i, v_{-i}))\Big],$$

where the expectation is over $v \sim F$ and $u_i^w(v_i;b) = v_i(g_i^w(b)) - p_i^w(b)$ for model parameters w. We assume that F has full support on the space of valuation profiles V, and recognizing that the regret is non-negative, an auction satisfies DSIC if and only if $rgt_i(w) = 0, \forall i \in N$, except for measure zero events. In this work, we focus on DSIC constriant and our RegretNet can be adapted to handle BIC constraint, see in [FNP18].

Given this, we re-formulate the learning problem as minimizing expected negated revenue subject to the expected ex post regret being zero for each bidder:

$$\begin{aligned} \min_{w \in \mathbb{R}^d} \quad & \mathbf{E}_{v\sim F}\Big[-\textstyle\sum_{i\in N} p_i^w(v)\Big] \\ \text{s.t.} \quad & rgt_i(w) = 0, \ \forall i \in N. \end{aligned}$$

Given a sample $\mathcal{S}$ of L valuation profiles from F, we estimate the empirical ex post regret for bidder i as:

$$\widehat{rgt}_i(w) = \frac{1}{L}\sum_{\ell=1}^{L}\Big[\max_{v_i'\in V_i} u_i^w\big(v_i^{(\ell)};\big(v_i', v_{-i}^{(\ell)}\big)\big) - u_i^w\big(v_i^{(\ell)}; v^{(\ell)}\big)\Big], \tag{1.2}$$

and seek to minimize the empirical loss (negated revenue) subject to the empirical regret being zero for all bidders:

$$\begin{aligned} \min_{w \in \mathbb{R}^d} \quad & -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i=1}^{n} p_i^w(v^{(\ell)}) \\ \text{s.t.} \quad & \widehat{rgt}_i(w) = 0, \ \ \forall i \in N. \end{aligned} \tag{1.3}$$

We additionally require the designed auction to satisfy IR, which can be ensured by restricting the search space to a class of parametrized auctions that charge no bidder more than her valuation for an allocation.

In Section 1.3 we will model the allocation and payment rules as neural networks and incorporate the IR requirement within the architecture. In Section 1.4 we describe how the IC constraints can be incorporated into the objective using Lagrange multipliers, so that the resulting neural nets can be trained with standard pipelines. Section 1.5 and Appendix A.2 present extensive experiments.

1.2.3 Quantile-Based Regret

Our characterization-free approach will lead to mechanisms with low (typically vanishing) expected ex post regret. The bounds on the expected ex post regret also yield guarantees of the form "the probability that the ex post regret is larger than x is at most q."

Definition 1.1 (Quantile-based ex post regret)**.** *For each bidder i, and q with $0 < q < 1$, the q-quantile-based ex post regret, $rgt_i^q(w)$, induced by the probability distribution F on valuation profiles, is defined as the smallest x such that*

$$\mathbb{P}\left(\max_{v_i' \in V_i} u_i^w(v_i; (v_i', v_{-i})) - u_i^w(v_i; (v_i, v_{-i})) \geqslant x\right) \leqslant q.$$

We can bound the q-quantile based regret $rgt_i^q(w)$ by the expected ex post regret $rgt_i(w)$ as in the following lemma. The proof appears in Appendix A.3.1.

Lemma 1.1. *For any fixed q, $0 < q < 1$, and bidder i, we can bound the q-quantile-based ex post regret by*

$$rgt_i^q(w) \leqslant \frac{rgt_i(w)}{q}.$$

Using this lemma we can show, for example, that when the expected ex post regret is 0.001, then the probability that the ex post regret exceeds 0.01 is at most 10%.

1.2.4 Generalization Bound

We conclude this section with two generalization bounds. We provide a lower bound on the expected revenue and an upper bound on the expected ex post regret in terms of the empirical revenue and empirical regret during training, the complexity (or capacity) of the auction class that we optimize over, and the number of sampled valuation profiles.

We measure the capacity of an auction class $\mathcal{M}$ using a definition of *covering numbers* used in the ranking literature [RS09]. Define the $\ell_{\infty,1}$ distance between auctions $(g, p), (g', p') \in \mathcal{M}$ as

$$\max_{v \in V} \sum_{i \in N, j \in M} |g_{ij}(v) - g'_{ij}(v)| + \sum_{i \in N} |p_i(v) - p'_i(v)|.$$

For any $\epsilon > 0$, let $\mathcal{N}_\infty(\mathcal{M}, \epsilon)$ be the minimum number of balls of radius ϵ required to cover $\mathcal{M}$ under the $\ell_{\infty,1}$ distance.

Theorem 1.2. *For each bidder i, assume that the valuation function v_i satisfies $v_i(S) \leqslant 1$, $\forall S \subseteq M$. Let $\mathcal{M}$ be a class of auctions that satisfy individual rationality. Fix $\delta \in (0, 1)$. With probability at least $1 - \delta$ over draw of sample $\mathcal{S}$ of L profiles from F, for any $(g^w, p^w) \in \mathcal{M}$,*

$$\mathbf{E}_{v \sim F}\left[\sum_{i \in N} p_i^w(v) \right] \geqslant \frac{1}{L} \sum_{\ell=1}^{L} \sum_{i=1}^{n} p_i^w(v^{(\ell)}) - 2n\Delta_L - Cn\sqrt{\frac{\log(1/\delta)}{L}},$$

and

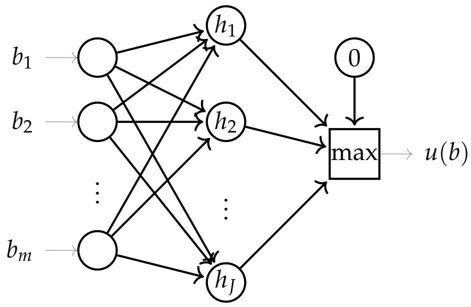
$$\frac{1}{n} \sum_{i=1}^{n} rgt_i(w) \;\leqslant\; \frac{1}{n} \sum_{i=1}^{n} \widehat{rgt}_i(w) \;+\; 2\Delta_L \;+\; C' \sqrt{\frac{\log(1/\delta)}{L}},$$

where $\Delta_L = \inf_{\epsilon > 0} \left\{ \frac{\epsilon}{n} + 2\sqrt{\frac{2 \log(\mathcal{N}_\infty(\mathcal{M}, \epsilon/2))}{L}} \right\}$ and C, C' are distribution-independent constants.

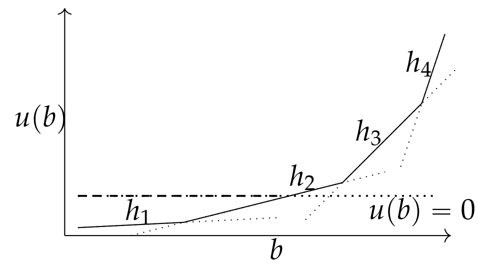
See Appendix A.3.2 for the proof. If the term Δ_L in the above bound goes to zero as the sample size L increases then the above bounds go to zero as $L \to \infty$. In Theorem 1.4 in Section 1.3, we bound Δ_L for the neural network architectures we present in this work.

1.3 Neural Network Architecture

We describe the RochetNet architecture for single-bidder multi-item settings in Section 1.3.1, and the RegretNet architecture for multi-bidder multi-item settings in Section 1.3.2. We focus on additive and unit-demand preferences. We discuss how to extend the constructions to capture combinatorial valuations for multi-bidder, multi-item settings in Appendix A.1.2.



(a)



(b)

Figure 1.1: *RochetNet: (a) Neural network representation of a non-negative, monotone, convex induced utility function; here $h_j(b) = \alpha_j \cdot b + \beta_j$ for $b \in \mathbb{R}^m$ and $\alpha_j \in [0,1]^m$. (b) An example of a utility function represented by RochetNet for one item.*

1.3.1 The RochetNet Architecture

Recall that in the single-bidder, multi-item setting we seek to encode utility functions that satisfy the requirements of Theorem 1.1. The associated auction mechanism can be deduced from the gradient of the utility function.

We first describe the construction for additive valuations. To model a non-negative, monotone, convex, Lipschitz utility function, we use the maximum of J linear functions with non-negative coefficients and zero:

$$u^{\alpha,\beta}(v) = \max\left\{\max_{j\in[J]}\{\alpha_j \cdot v + \beta_j\}, 0\right\}, \tag{1.4}$$

where parameters $w = (\alpha, \beta)$, with $\alpha_j \in [0,1]^m$ and $\beta_j \in \mathbb{R}$ for $j \in [J]$. By bounding the hyperplane coefficients to $[0,1]$, we guarantee that the function is 1-Lipschitz. The following theorem verifies that the utility modeled by RochetNet satisfies Rochet's characterization (Theorem 1.1). The proof is given in Appendix A.3.3.

Theorem 1.3. *For any $\alpha \in [0,1]^{mJ}$ and $\beta \in \mathbb{R}^J$, the function $u^{\alpha,\beta}$ is non-negative, monotonically non-decreasing, convex and 1-Lipschitz w.r.t. the ℓ_1-norm.*

The utility function, represented as a single layer neural network, is illustrated in Figure 1.1(a), where each $h_j(b) = \alpha_j \cdot b + \beta_j$ for bid $b \in \mathbb{R}^m$. Figure 1.1(b) shows an example of a utility function represented by RochetNet for $m = 1$. By using a large number of hyperplanes one can use this neural network architecture to search over a sufficiently rich class of monotone, convex 1-Lipschitz utility functions. Once trained, the mechanism (g^w, p^w), with $w = (\alpha, \beta)$, can be derived from the gradient of the utility function, with the allocation rule given by:

$$g^w(b) = \nabla u^{\alpha,\beta}(b), \tag{1.5}$$

and the payment rule is given by the difference between the expected value to the bidder from the allocation and the bidder's utility:

$$p^w(b) \;=\; \nabla u^{\alpha,\beta}(b) \cdot b \;-\; u^{\alpha,\beta}(b). \tag{1.6}$$

Here the utility gradient can be computed as: $\nabla_j u^{\alpha,\beta}(b) \;=\; \alpha_{j^*(b)}$, for $j^*(b) \in \mathrm{argmax}_{j \in [J]}\{\alpha_j \cdot b \;+\; \beta_j\}$. We seek to minimize the negated, expected revenue:

$$-\mathbf{E}_{v \sim F}\big[\nabla u^{\alpha,\beta}(v) \cdot v - u^{\alpha,\beta}(v)\big] = \mathbf{E}_{v \sim F}\big[\beta_{j^*(v)}\big]. \tag{1.7}$$

To ensure that the objective is a continuous function of the parameters α and β (so that the parameters can be optimized efficiently), the gradient term is computed approximately by using a *softmax* operation in place of the argmax. The loss function that we use is given by the negated revenue with approximate gradients:

$$\mathcal{L}(\alpha, \beta) \;=\; -\mathbf{E}_{v \sim F}\big[\sum_{j \in [J]} \beta_j \tilde{\nabla}_j(v)\big], \tag{1.8}$$

where

$$\tilde{\nabla}_j(v) \;=\; \mathrm{softmax}_j\big(\kappa \cdot (\alpha_1 \cdot v + \beta_1), \ldots, \kappa \cdot (\alpha_J \cdot v + \beta_J)\big) \tag{1.9}$$

and $\kappa > 0$ is a constant that controls the quality of the approximation.[4] We seek to optimize the parameters of the neural network $\alpha \in [0,1]^{mJ}, \beta \in \mathbb{R}^J$ to minimize loss. Given a sample $\mathcal{S} = \{v^{(1)}, \ldots, v^{(L)}\}$ drawn from F, we optimize an empirical version of the loss. We only do this approximation during training to minimize the empirical loss function. Nevertheless, we always use argmax during testingto guarantee the mechanism exact DSIC.

This approach easily extends to a single bidder with a unit-demand valuation. In this case, the sum of the allocation probabilities cannot exceed one. This is enforced by restricting

[4]The softmax function, $\mathrm{softmax}_j(\kappa x_1, \ldots, \kappa x_J) = e^{\kappa x_j} / \sum_{j'} e^{\kappa x_{j'}}$, takes as input J real numbers and returns a probability distribution consisting of J probabilities, proportional to the exponential of the inputs.

the coefficients for each hyperplane to sum up to at most one, i.e. $\sum_{k=1}^{m} \alpha_{jk} \leqslant 1, \forall j \in [J]$, and $\alpha_{jk} \geqslant 0, \forall j \in J, k \in [m]$.[5] It can be verified that even with this restriction, the induced utility function continuous to be monotone, convex and Lipschitz, ensuring that the resulting mechanism is DSIC.[6]

An interpretation of the RochetNet architecture is that the network maintains a menu of randomized allocations and prices, and chooses the option from the menu that maximizes the bidder's utility based on the bid. Each linear function $h_j(b) = \alpha_j \cdot b + \beta_j$ in RochetNet corresponds to an option on the menu, with the allocation probabilities and payments encoded through the parameters α_j and β_j respectively. Our RochetNet framework provides a novel neural network implementation for the menu-based mechanism. Indeed, combining with standard machine learning training pipeline (e.g., gradient descent method), the RochetNet can be solved very fast. Recently, [STZ19] extended RochetNet to more general settings, including non-linear utility function settings.

1.3.2 The RegretNet Architecture

We next describe the basic architecture for the characterization-free, RegretNet approach. Recall that in this case the goal is to train neural networks that explicitly encode the allocation and payment rule of the mechanism. The architectures generally consist of two logically distinct components: the allocation and payment networks. These components are trained together and the outputs of these networks are used to compute the regret and revenue of the auction.

Additive Valuations

An overview of the RegretNet architecture for additive valuations is given in Figure 1.2. The allocation network encodes a randomized allocation rule $g^w : \mathbb{R}^{nm} \to [0,1]^{nm}$ and the payment network encodes a payment rule $p^w : \mathbb{R}^{nm} \to \mathbb{R}^n_{\geqslant 0}$, both of which are modeled as feed-forward, fully-connected networks with a tanh activation function in each of the hidden nodes. The

[5] To achieve this contraint, we can re-parameterize α_{jk} as $\text{softmax}_k(\gamma_{j1}, \cdots, \gamma_{jm})$, where $\gamma_{jk} \in \mathbb{R}, \forall j \in J, k \in m$.

[6] The original characterization of Rochet [Roc87] applies to general, convex outcome spaces, as is the case here.

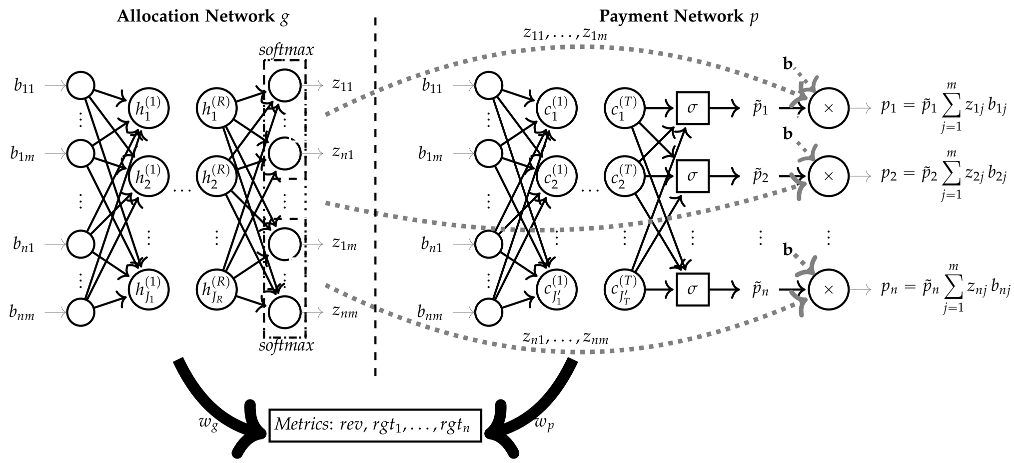


Figure 1.2: *RegretNet: The allocation and payment networks for a setting with n additive bidders and m items. The inputs are bids from each bidder for each item. The revenue rev and expected ex post rgt_i are defined as a function of the parameters of the allocation and payment networks $w = (w_g, w_p)$.*

input layer of the networks consists of bids $b_{ij} \geqslant 0$ representing the valuation of bidder i for item j.

The allocation network outputs a vector of allocation probabilities $z_{1j} = g_{1j}(b), \dots, z_{nj} = g_{nj}(b)$, for each item $j \in [m]$. To ensure feasibility, i.e., that the probability of an item being allocated is at most one, the allocations are computed using a softmax activation function, so that for all items j, we have $\sum_{i=1}^{n} z_{ij} \leqslant 1$. To accommodate the possibility of an item not being assigned, we include a dummy node in the softmax computation to hold the residual allocation probability. The payment network outputs a payment for each bidder that denotes the amount the bidder should pay in expectation for a particular bid profile.

To ensure that the auction satisfies *ex post IR*, i.e., does not charge a bidder more than her expected value for the allocation[7], the network first computes a normalized payment $\tilde{p}_i \in [0, 1]$ for each bidder i using a sigmoidal unit, and then outputs a payment $p_i = \tilde{p}_i(\sum_{j=1}^{m} z_{ij}\, b_{ij})$, where the z_{ij}'s are the outputs from the allocation network.

[7]This guarantees the ex post individual rationality, since the expectation is just from the randomization of the mechanism.

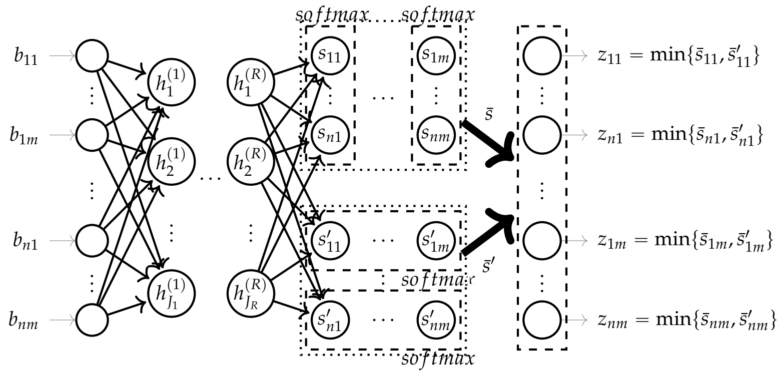


Figure 1.3: *RegretNet: The allocation network for settings with n* unit-demand *bidders and m items.*

Unit-Demand Valuations

The allocation network for unit-demand bidders is the feed-forward network shown in Figure 1.3. For revenue maximization in this setting, it is sufficient to consider allocation rules that assign at most one item to each bidder.[8] In the case of randomized allocation rules, this requires that the total allocation probability to each bidder is at most one, i.e., $\sum_j z_{ij} \leqslant 1,\ \forall i \in [n]$. We would also require that no item is over-allocated, i.e., $\sum_i z_{ij} \leqslant 1,\ \forall j \in [m]$. Hence, we design allocation networks for which the matrix of output probabilities $[z_{ij}]_{i,j=1}^{n}$ is doubly stochastic.[9]

In particular, we have the allocation network compute two sets of scores s_{ij}'s and s'_{ij}'s. Let s, $s' \in \mathbb{R}^{nm}$ denote the corresponding matrices. The first set of scores are normalized along the rows and the second set of scores normalized along the columns. Both normalizations 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:

$$z_{ij} \;=\; \varphi_{ij}^{DS}(s, s') \;=\; \min\left\{\frac{e^{s_{ij}}}{\sum_{k=1}^{n+1} e^{s_{kj}}},\ \frac{e^{s'_{ij}}}{\sum_{k=1}^{m+1} e^{s'_{ik}}}\right\},$$

where indices $n+1$ and $m+1$ denote dummy inputs that correspond to an item not being allocated to any bidder and a bidder not being allocated any item, respectively.

---

[8]This holds by a simple reduction argument: for any IC auction that allocates multiple items, one can construct an IC auction with the same revenue by retaining only the most-preferred item among those allocated to a bidder.

[9]This is a more general definition for doubly stochastic than is typical. Doubly stochastic is usually defined on a square matrix with the sum of rows and the sum of columns equal to 1.

We first show that $\varphi^{DS}(s, s')$ as constructed is doubly stochastic, and that we do not lose in generality by the constructive approach that we take. See Appendix A.3.4 for a proof.

Lemma 1.2. *The matrix $\varphi^{DS}(s, s')$ is doubly stochastic $\forall\, s, s' \in \mathbb{R}^{nm}$. For any doubly stochastic matrix $z \in [0,1]^{nm}$, $\exists\, s, s' \in \mathbb{R}^{nm}$, for which $z = \varphi^{DS}(s, s')$.*

It remains to show that doubly-stochastic matrices correspond to lotteries over one-to-one assignments. This is a special case of the *bihierarchy* structure proposed in [Bud+13] (Theorem 1), which we state in the following lemma for completeness.[10]

Lemma 1.3 ([Bud+13]). *Any doubly stochastic matrix $A \in \mathbb{R}^{n\times m}$ can be represented as a convex combination of matrices $B^1, \dots, B^k$ where each $B^\ell \in \{0,1\}^{n\times m}$ and $\sum_{j\in[m]} B_{ij} \leqslant 1$, $\forall i \in [n]$ and $\sum_{i\in[n]} B_{ij} \leqslant 1$, $\forall j \in [m]$.*

The payment network for unit-demand valuations is the same as for the case of additive valuations (see Figure 1.2).

1.3.3 Covering Number Bounds

We conclude this section by instantiating our generalization bound from Section 1.2.4 for the RegretNet architectures, where we have both a regret and revenue term. Analogous results can be derived for RochetNet, where we only have a revenue term.

Theorem 1.4. *For RegretNet with R hidden layers, K nodes per hidden layer, d_g parameters in the allocation network, d_p parameters in the payment network, m items, n bidders, a sample size of L, and the vector of all model parameters w satisfying $\|w\|_1 \leqslant W$[11] the following are valid bounds for the Δ_L term defined in Theorem 1.2, for different bidder valuation types:*

(a) additive valuations:

$\Delta_L \leqslant O\big(\sqrt{R(d_g + d_p)\log(LW\max\{K, mn\})/L}\big)$,

(b) unit-demand valuations:

$\Delta_L \leqslant O\big(\sqrt{R(d_g + d_p)\log(LW\max\{K, mn\})/L}\big)$,

[10] Budish et al. [Bud+13] also propose a polynomial algorithm to decompose the doubly stochastic matrix.

[11] Recall that $\|\cdot\|_1$ is the induced matrix norm, i.e. $\|w\|_1 = \max_j \sum_i |w_{ij}|$.

The proof is given in Appendix A.3.6. As the sample size $L \to \infty$, the term $\Delta_L \to 0$. The dependence of the above result on the number of layers, nodes, and parameters in the network is similar to standard covering number bounds for neural networks [AB09a].

1.4 Optimization and Training

We next describe how we train the neural network architectures presented in the previous section. We focus on the RegretNet architectures where we have to take care of the incentives directly. The approach that we take for RochetNet is the standard (projected) stochastic gradient descent[12] (SGD) for loss function $\mathcal{L}(\alpha, \beta)$ in Equation 1.8.

For RegretNet we use the augmented Lagrangian method to solve the constrained training problem in (1.3) over the space of neural network parameters w. We first define the Lagrangian function for the optimization problem, augmented with a quadratic penalty term for violating the constraints:

$$\mathcal{C}_\rho(w; \lambda) \;=\; -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i \in N} p_i^w(v^{(\ell)}) \;+\; \sum_{i \in N} \lambda_i \, \widehat{rgt}_i(w) \;+\; \frac{\rho}{2}\sum_{i \in N}\left(\widehat{rgt}_i(w)\right)^2$$

where $\lambda \in \mathbb{R}^n$ is a vector of Lagrange multipliers, and $\rho > 0$ is a fixed parameter that controls the weight on the quadratic penalty. The solver alternates between the following updates on the model parameters and the Lagrange multipliers: (a) $w^{new} \in \mathrm{argmin}_w\, \mathcal{C}_\rho(w^{old}; \lambda^{old})$ and (b) $\lambda_i^{new} \;=\; \lambda_i^{old} \;+\; \rho\, \widehat{rgt}_i(w^{new}),\ \forall i \in N$.

The solver is described in Algorithm 1. We divide the training sample $\mathcal{S}$ into minibatches of size B, and perform several passes over the training samples (with random shuffling of the data after each pass). We denote the minibatch received at iteration t by $\mathcal{S}_t \;=\; \{v^{(1)}, \ldots, v^{(B)}\}$. The update (a) on model parameters involves an unconstrained optimization of $\mathcal{C}_\rho$ over w and is performed using a gradient-based optimizer. Let $\widetilde{rgt}_i(w)$ denote the empirical regret in (1.2)

[12]During training for additive valuations setting in RochetNet, we project each weight α_{jk} into $[0, 1]$ to guarantee feasibility.

Algorithm 1 RegretNet Training

Input: Minibatches $\mathcal{S}_1, \ldots, \mathcal{S}_T$ of size B
Parameters: $\forall t, \rho_t > 0$, $\gamma > 0$, $\eta > 0$, $\Gamma \in \mathbb{N}$, $K \in \mathbb{N}$
Initialize: $w^0 \in \mathbb{R}^d$, $\lambda^0 \in \mathbb{R}^n$
for $t = 0$ **to** T **do**
  Receive minibatch $\mathcal{S}_t = \{v^{(1)}, \ldots, v^{(B)}\}$
  Initialize misreports ${v'}_i^{(\ell)} \in V_i, \forall \ell \in [B],\ i \in N$
  for $r = 0$ **to** Γ **do**
    $\forall \ell \in [B],\ i \in N$:
      ${v'}_i^{(\ell)} \leftarrow {v'}_i^{(\ell)} + \gamma \nabla_{v'_i} u_i^w\big(v_i^{(\ell)}; \big({v'}_i^{(\ell)}, v_{-i}^{(\ell)}\big)\big)$
  end for
  Compute regret gradient: $\forall \ell \in [B], i \in N$:
    $g_{\ell,i}^t =$
    $\nabla_w \Big[u_i^w\big(v_i^{(\ell)}; \big({v'}_i^{(\ell)}, v_{-i}^{(\ell)}\big)\big) - u_i^w\big(v_i^{(\ell)}; v^{(\ell)}\big)\Big]\Big|_{w=w^t}$
  Compute Lagrangian gradient using (1.10) and update w^t:
    $w^{t+1} \leftarrow w^t - \eta \nabla_w \mathcal{C}_{\rho_t}(w^t, \lambda^t)$
  Update Lagrange multipliers once in Q iterations:
    if t is a multiple of Q
      $\lambda_i^{t+1} \leftarrow \lambda_i^t + \rho_t \, \widetilde{rgt}_i(w^{t+1}), \quad \forall i \in N$
    else
      $\lambda^{t+1} \leftarrow \lambda^t$
end for

computed on minibatch $\mathcal{S}_t$. The gradient of $\mathcal{C}_\rho$ w.r.t. w for fixed λ^t is given by:

$$\nabla_w \mathcal{C}_\rho(w; \lambda^t) = -\frac{1}{B} \sum_{\ell=1}^{B} \sum_{i \in N} \nabla_w \, p_i^w(v^{(\ell)}) + \sum_{i \in N} \sum_{\ell=1}^{B} \lambda_i^t \, g_{\ell,i} + \rho \sum_{i \in N} \sum_{\ell=1}^{B} \widetilde{rgt}_i(w) \, g_{\ell,i} \tag{1.10}$$

where

$$g_{\ell,i} = \nabla_w \Big[\max_{v'_i \in V_i} u_i^w\big(v_i^{(\ell)}; \big(v'_i, v_{-i}^{(\ell)}\big)\big) - u_i^w\big(v_i^{(\ell)}; v^{(\ell)}\big)\Big].$$

The terms $\widetilde{rgt}_i$ and $g_{\ell,i}$ in turn involve a "max" over misreports for each bidder i and valuation profile ℓ. We solve this inner maximization over misreports using another gradient based optimizer. In particular, we maintain misreports ${v'}_i^{(\ell)}$ for each i and valuation profile ℓ. For every update on the model parameters w^t, we perform Γ gradient updates to compute the optimal misreports: ${v'}_i^{(\ell)} = {v'}_i^{(\ell)} + \gamma \nabla_{v'_i} u_i^w\big(v_i^{(\ell)}; \big({v'}_i^{(\ell)}, v_{-i}^{(\ell)}\big)\big)$, for some $\gamma > 0$. We show a visualization of these iterations in Appendix 1.5.6. In our experiments, we use the Adam optimizer [KB15] for updates on model parameters w and misreports $v'^{(\ell)}_i$.[13]

Since the optimization problem is non-convex, the solver is not guaranteed to reach a globally optimal solution. However, our method proves very effective in our experiments. The learned auctions incur very low regret and closely match the structure of optimal auctions in settings where this is known.

1.5 Experiments

We demonstrate that our approach can recover near-optimal auctions for essentially all settings for which the optimal solution is known, that it is an effective tool for confirming or refuting hypotheses about optimal designs, and that it can find new auctions for settings where there is no known analytical solution. We present a representative subset of the results here, and provide additional experimental results in Appendix A.2.

1.5.1 Setup

We implemented our framework using the TensorFlow deep learning library.[14] For RochetNet we initialized parameters α and β in Equation (1.4) using a random uniform initializer over the interval [0,1] and a zero initializer, respectively. For RegretNet we used the tanh activation function at the hidden nodes, and Glorot uniform initialization [GB10]. We performed cross validation to decide on the number of hidden layers and the number of nodes in each hidden layer. We include exemplary numbers that illustrate the tradeoffs in Section 1.5.7.

We trained RochetNet on 2^{15} valuation profiles sampled every iteration in an online manner. We used the Adam optimizer with a learning rate of 0.1 for 20,000 iterations for making the updates. The parameter κ in Equation (1.9) was set to 1,000. Unless specified otherwise we used a max network over 1,000 linear functions to model the induced utility functions, and report our results on a sample of 10,000 profiles.

[13] Adam is a variant of SGD, which involves a momentum term to update weights. Lines 9 and 15 in the pseudo-code of Algorithm 1 are for a standard SGD algorithm.

[14] All code is available through the GitHub repository at https://github.com/saisrivatsan/deep-opt-auctions.

For RegretNet we used a sample of 640,000 valuation profiles for training and a sample of 10,000 profiles for testing. The augmented Lagrangian solver was run for a maximum of 80 epochs (full passes over the training set) with a minibatch size of 128. The value of ρ in the augmented Lagrangian was set to 1.0 and incremented every two epochs. An update on w^t was performed for every minibatch using the Adam optimizer with learning rate 0.001. For each update on w^t, we ran $\Gamma = 25$ misreport updates steps with learning rate 0.1. At the end of 25 updates, the optimized misreports for the current minibatch were cached and used to initialize the misreports for the same minibatch in the next epoch. An update on λ^t was performed once every 100 minibatches (i.e., $Q = 100$).

We ran all our experiments on a compute cluster with NVDIA Graphics Processing Unit (GPU) cores.

1.5.2 Evaluation

In addition to the revenue of the learned auction on a test set, we also evaluate the regret achieved by RegretNet, averaged across all bidders and test valuation profiles, i.e., $rgt = \frac{1}{n}\sum_{i=1}^{n} \widehat{rgt}_i(g^w, p^w)$. Each $\widehat{rgt_i}$ has an inner "max" of the utility function over bidder valuations $v_i' \in V_i$ (see (1.2)). We evaluate these terms by running gradient ascent on v_i' with a step-size of 0.1 for 2,000 iterations (we test 1,000 different random initial v_i' and report the one that achieves the largest regret).

For some of the experiments we also report the total time it took to train the network. This time is incurred during offline training, while the allocation and payments can be computed in a few milliseconds once the network is trained.

1.5.3 The Manelli-Vincent and Pavlov Auctions

As a representative example of the exhaustive set of analytical results that we can recover with our approach we discuss the Manelli-Vincent and Pavlov auctions [MV06; Pav11]. We specifically consider the following single-bidder, two-item settings:

A. Single bidder with additive valuations over two items, where the item values are inde-

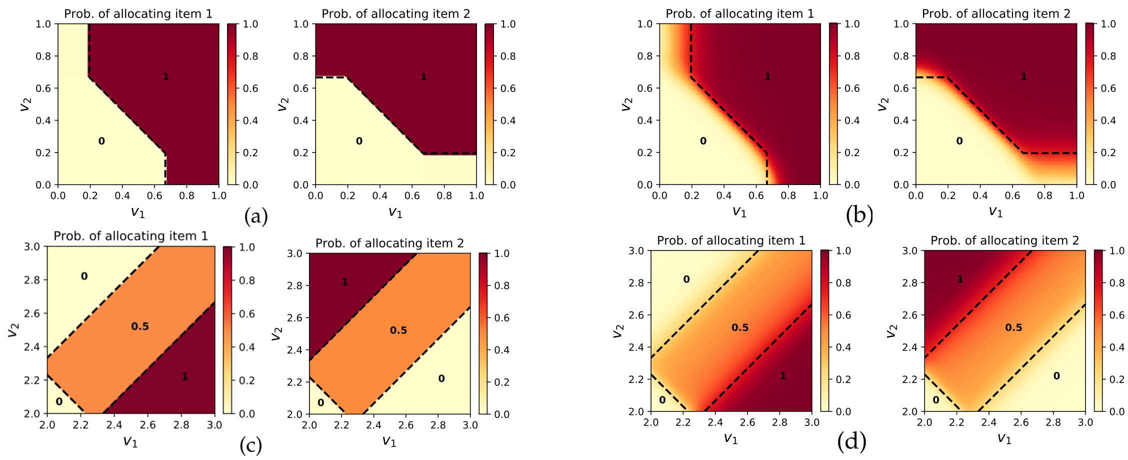


Figure 1.4: *Side-by-side comparison of allocation rules learned by RochetNet and RegretNet for single bidder, two items settings. Panels (a) and (b) are for Setting A and Panels (c) and (d) are for Setting B. The panels describe the probability that the bidder is allocated item 1 (left) and item 2 (right) for different valuation inputs. The optimal auctions are described by the regions separated by the dashed black lines, with the numbers in black the optimal probability of allocation in the region.*

pendent draws from $U[0,1]$.

B. Single bidder with unit-demand valuations over two items, where the item values are independent draws from $U[2,3]$.

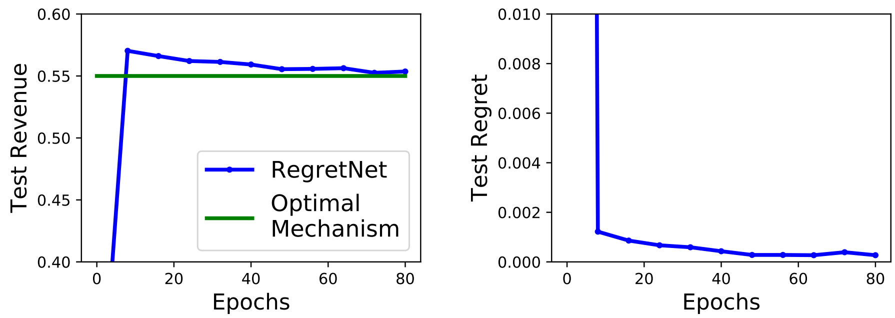
The optimal design for the first setting is given by Manelli and Vincent [MV06], who show that the optimal mechanism is deterministic and offers the bidder three options: receive both items and pay $(4-\sqrt{2})/3$, receive item 1 and pay $2/3$, or receive item 2 and pay $2/3$. For the second setting Pavlov [Pav11] shows that it is optimal to offer a fair lottery $(\frac{1}{2},\frac{1}{2})$ over the items (at a discount), or to purchase any item at a fixed price. For the parameters here the price for the lottery is $\frac{1}{6}(8+\sqrt{22}) \approx 2.115$ and the price for an individual item is $\frac{1}{6}+\frac{1}{6}(8+\sqrt{22}) \approx 2.282$.

We used two hidden layers with 100 hidden nodes in RegretNet for these settings. A visualization of the optimal allocation rule and those learned by RochetNet and RegretNet is given in Figure 1.4. Figure 1.5(a) gives the optimal revenue, the revenue and regret obtained by RegretNet, and the revenue obtained by RochetNet. Figure 1.5(b) shows how these terms evolve over time during training in RegretNet.

We find that both approaches essentially recover the optimal design, not only in terms of revenue, but also in terms of the allocation rule and transfers. The auctions learned by

Distribution	Opt	RegretNet		RochetNet
	rev	*rev*	*rgt*	*rev*
Setting A	0.550	0.554	< 0.001	0.550
Setting B	2.137	2.137	< 0.001	2.136

(a)



(b)

Figure 1.5: *(a): Test revenue and regret for RegretNet and revenue for RochetNet for Settings A and B. (b): Plot of test revenue and regret as a function of training epochs for Setting A with RegretNet.*

RochetNet are exactly DSIC and match the optimal revenue precisely, with sharp decision boundaries in the allocation and payment rule. The decision boundaries for RegretNet are smoother, but still remarkably accurate. The revenue achieved by RegretNet matches the optimal revenue up to a $< 1\%$ error term and the regret it incurs is < 0.001. The plots of the test revenue and regret show that the augmented Lagrangian method is effective in driving the test revenue and the test regret towards optimal levels.

The additional domain knowledge incorporated into the RochetNet architecture leads to exactly DSIC mechanisms that match the optimal design more accurately, and speeds up computation (the training took about 10 minutes compared to 11 hours). On the other hand, we find it surprising how well RegretNet performs given that it starts with no domain knowledge at all.

We present and discuss a host of additional experiments with single-bidder, two-item settings in Appendix A.2.

Items	SJA (*rev*)	RochetNet (*rev*)
2	0.549187	0.549175
3	0.875466	0.875464
4	1.219507	1.219505
5	1.576457	1.576455
6	1.943239	1.943216
7	2.318032	2.318032
8	2.699307	2.699305
9	3.086125	3.086125
10	3.477781	3.477722

Figure 1.6: *Revenue of the Straight-Jacket Auction (SJA) computed via the recursive formula in [GK18], and that of the auction learned by RochetNet, for various numbers of items m. The SJA is known to be optimal for up to six items, and conjectured to be optimal for any number of items.*

1.5.4 The Straight-Jacket Auction

Extending the analytical result of [MV06] to a single bidder, and an arbitrary number of items (even with additive preferences, all uniform on $[0,1]$) has proven elusive. It is not even clear whether the optimal mechanism is deterministic or requires randomization.

A recent breakthrough came with Giannakopoulos and Koutsoupias [GK18], who were able to find a pattern in the results for two items and three items. The proposed mechanism—the Straight-Jacket Auction (SJA)—offers bundles of items at fixed prices. The key to finding these prices is to view the best-response regions as a subdivision of the m-dimensional cube, and observe that there is an intrinsic relationship between the price of a bundle of items and the volume of the respective best-response region.

Giannakopoulos and Koutsoupias gave a recursive algorithm for finding the subdivision and the prices, and used LP duality to prove that the SJA is optimal for $m \leqslant 6$ items.[15] They also conjecture that the SJA remains optimal for general m, but were unable to prove it.

Figure 1.6 gives the revenue of the SJA, and that found by RochetNet for $m \leqslant 10$ items. We used a test sample of 2^{30} valuation profiles (instead of 10,000) to compute these numbers for higher precision. It shows that RochetNet finds the optimal revenue for $m \leqslant 6$ items, and that it finds DSIC auctions whose revenue matches that of the SJA for $m = 7, 8, 9$, and 10 items. Closer inspection reveals that the allocation and payment rules learned by RochetNet

[15] The duality argument developed by Giannakopoulos and Koutsoupias is similar but incomparable to the duality approach of [DDT13]. We will return to the latter in Section 1.5.5.

essentially match those predicted by Giannakopoulos and Koutsoupias for all $m \leqslant 10$. We take this as strong additional evidence that the conjecture of Giannakopoulos and Koutsoupias is correct.

For the experiments in this subsection, we used a max network over 10,000 linear functions (instead of 1,000) to increase the representation and flexibility of the neural network. This *overparameterization* trick is commonly used in deep learning and has proven to be very effective in practice, see e.g., [KSH12; AZLS19]. We followed up on the usual training phase with an additional 20 iterations of training using Adam optimizer with learning rate 0.001 and a minibatch size of 2^{30}. We also found it useful to impose item-symmetry on the learned auction, especially for $m = 9$ and 10 items, as this helped with accuracy and reduced training time. Imposing symmetry comes without loss of generality for auctions with an item-symmetric distribution [DW12]. With these modifications it took about 13 hours to train the networks.

1.5.5 Discovering New Optimal Designs

We next demonstrate the potential of RochetNet to discover new optimal designs. For this, we consider a single bidder with additive but correlated valuations for two items as follows:

C. One additive bidder and two items, where the bidder's valuation is drawn uniformly from the triangle $T = \{(v_1, v_2) | \frac{v_1}{c} + v_2 \leqslant 2, v_1 \geqslant 0, v_2 \geqslant 1\}$ where $c > 0$ is a free parameter.

There is no analytical result for the optimal auction design for this setting. We ran RochetNet for different values of c to discover the optimal auction. The mechanisms learned by RochetNet for $c = 0.5, 1, 3$, and 5 are shown in Figure 1.8. Based on this, we conjectured that the optimal mechanism contains two menu items for $c \leqslant 1$, namely $\{(0,0), 0\}$ and $\{(1,1), \frac{2+\sqrt{1+3c}}{3}\}$, and three menu items for $c > 1$, namely $\{(0,0), 0\}$, $\{(1/c, 1), 4/3\}$, and $\{(1,1), 1 + c/3\}$, giving the optimal allocation and payment in each region. In particular, as c transitions from values less than or equal to 1 to values larger than 1, the optimal mechanism transitions from being deterministic to being randomized. Figure 1.7 gives the revenue achieved by RochetNet and the conjectured optimal format for a range of parameters c computed on 10^6 valuation profiles.

c	Opt (*rev*)	RochetNet (*rev*)
0.500	1.104783	1.104777
1.000	1.185768	1.185769
3.000	1.482129	1.482147
5.000	1.778425	1.778525

Figure 1.7: *Revenue of the newly discovered optimal mechanism and that of RochetNet, for Setting C with varying parameter c.*

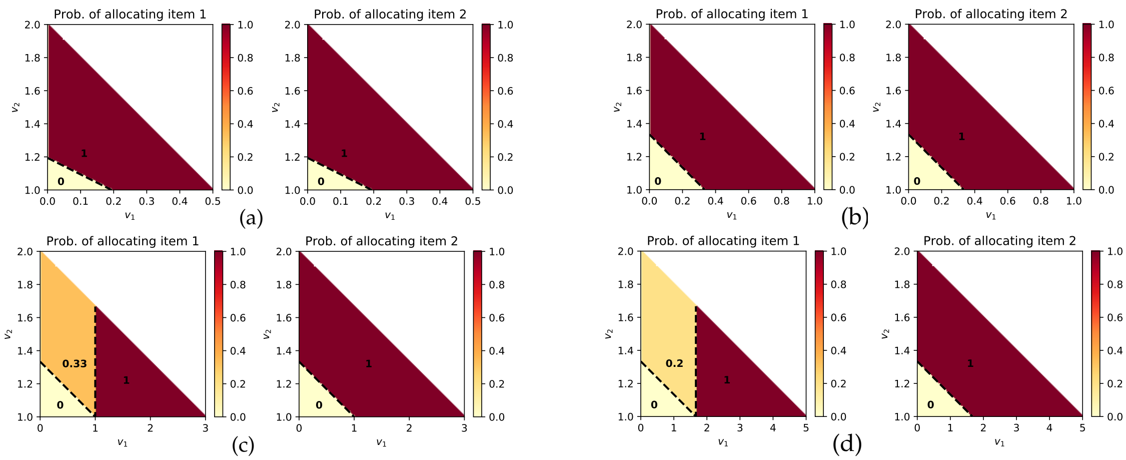


Figure 1.8: *Allocation rules learned by RochetNet for Setting C. The panels describe the probability that the bidder is allocated item 1 (left) and item 2 (right) for* $c = 0.5, 1, 3,$ *and* 5*. The auctions proposed in Theorem 1.5 are described by the regions separated by the dashed black lines, with the numbers in black the optimal probability of allocation in the region.*

We validate the optimality of this auction through duality theory [DDT13] in Theorem 1.5. The proof is given in Appendix A.3.7.

Theorem 1.5. *For any* $c > 0$*, suppose the bidder's valuation is uniformly distributed over set* $T = \{(v_1, v_2) | \frac{v_1}{c} + v_2 \leqslant 2, v_1 \geqslant 0, v_2 \geqslant 1\}$*. Then the optimal auction contains two menu items* $\{(0,0), 0\}$ *and* $\{(1,1), \frac{2+\sqrt{1+3c}}{3}\}$ *when* $c \leqslant 1$*, and three menu items* $\{(0,0),0\}$*,* $\{(1/c,1), 4/3\}$*, and* $\{(1,1), 1+c/3\}$ *otherwise.*

Shen, Tang, and Zuo [STZ19] also use neural network framework to find the optimal auction for a similar setting: single additive bidder and two items, where the bidder's valuation is drawn uniformly from the triangle $\{(v_1, v_2) | \frac{v_1}{c} + v_2 \leqslant 1, v_1 \geqslant 0, v_2 \geqslant 0\}$. Our results demonstrate that RochetNet is a powerful tool to help economists to find new theory.

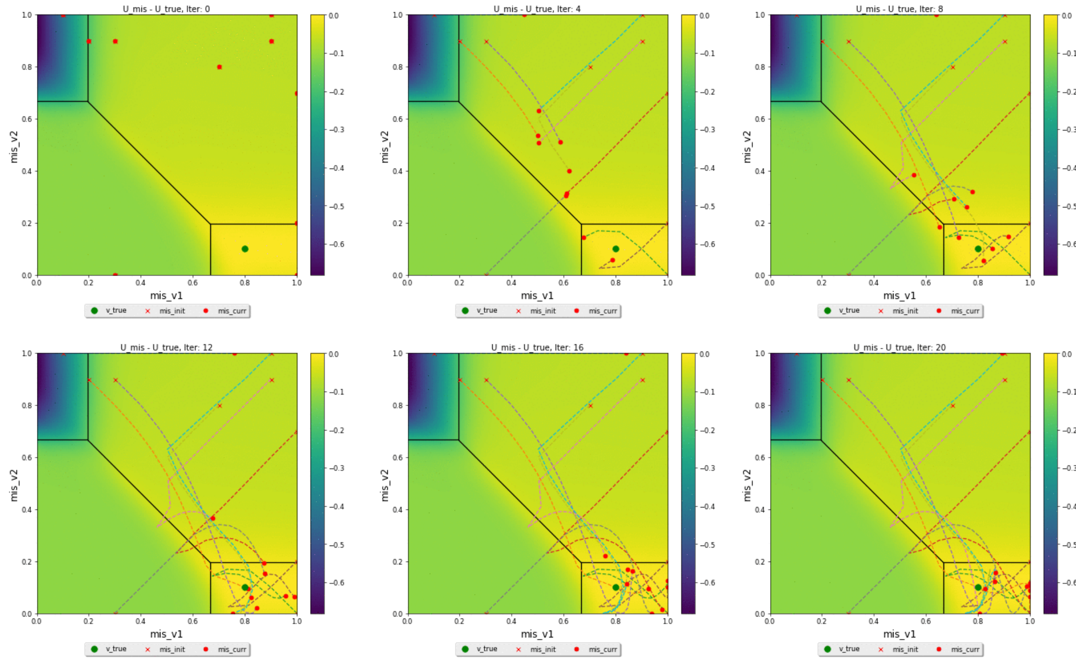


Figure 1.9: *Visualization of the gradient-based approach to regret approximation for a well-trained auction for Setting A. The top left figure shows the true valuation (green dot) and ten random initial misreports (red dots). The remaining figures display 20 steps of gradient descent, showing one in every four steps.*

1.5.6 Gradient-Based Regret Approximation

In Section 1.4, we describe a gradient-based approach to estimating a bidder's regret. We present a visualization of this approach in Figure 1.9 for a well-trained mechanism that has (almost) zero regret for Setting A. We consider a bidder with true valuation $(v_1, v_2) = (0.1, 0.8)$, represented as a green dot. The heat map represents the utility difference $u((v_1, v_2); (b_1, b_2)) - u((v_1, v_2); (v_1, v_2))$ for misreports $(b_1, b_2) \in [0, 1]^2$, with shades of yellow corresponding to low utility differences and shades of blue corresponding to high utility differences. To estimate the regret of the bidder which is essentially zero in this case, we draw 10 random initial misreports from the underlying valuation distribution, represented as red dots, and then perform a sequence of gradient-descent steps on these random misreports. The figure shows the random initial misreports and then the 20 gradient-descent steps, plotting one in every four steps.

1.5.7 Scaling Up

We next consider settings with up to five bidders and up to ten items. This is several orders of magnitude more complex than existing analytical or computational results. It is also a natural playground for RegretNet as no tractable characterizations of IC mechanisms are known for these settings.

We specifically consider the following two settings, which generalize the basic setting considered in [MV06] and [GK18] to more than one bidder:

D. Three additive bidders and ten items, where bidders draw their value for each item independently from $U[0, 1]$.

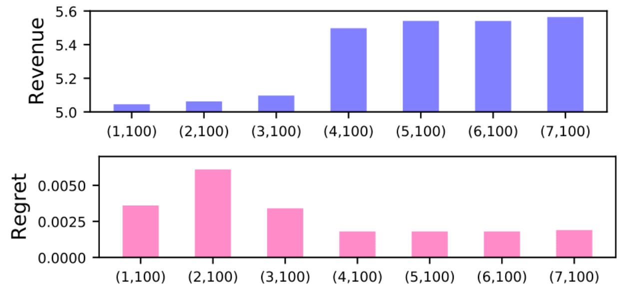
E. Five additive bidders and ten items, where bidders draw their value for each item independently from $U[0, 1]$.

The optimal auction for these settings is not known. However, running a separate Myerson auction for each item is optimal in the limit of the number of bidders [Pal83]. For a regime with a small number of bidders, this provides a strong benchmark. We also compare to selling the grand bundle via a Myerson auction.

For Setting D, we show in Figure 1.10(a) the revenue and regret of the learned auction on a validation sample of 10,000 profiles, obtained with different architectures. Here (R, K) denotes an architecture with R hidden layers and K nodes per layer. The $(5, 100)$ architecture has the lowest regret among all the 100-node networks for both Setting D and Setting E. Figure 1.10(b) shows that the learned auctions yield higher revenue compared to the baselines, and do so with tiny regret.

1.5.8 Comparison to LP

Finally, we compare the running time of RegretNet with the LP approach proposed in [CS02; CS04]. To be able to run the LP, we consider a smaller setting with two additive bidders and three items, with item values drawn independently from $U[0, 1]$. For RegretNet we used two hidden layers, and 100 nodes per hidden layer. The LP was solved with the commercial solver



(a)

Setting	RegretNet *rev*	RegretNet *rgt*	Item-wise Myerson	Bundled Myerson
D: 3×10	5.541	< 0.002	5.310	5.009
E: 5×10	6.778	< 0.005	6.716	5.453

(b)

Figure 1.10: *(a) Revenue and regret of RegretNet on the validation set for auctions learned for Setting D using different architectures, where (R, K) denotes R hidden layers and K nodes per layer. (b) Test revenue and regret for Settings D and E, for the (5, 100) architecture.*

Setting	Method	*rev*	*rgt*	*IR viol.*	Run-time
2×3	RegretNet	1.291	< 0.001	0	$\sim$9 hrs
	LP (5 bins/value)	1.53	0.019	0.027	69 hrs

Figure 1.11: *Test revenue, regret, IR violation, and running-time for RegretNet and an LP-based approach for a two bidder, three items setting with additive uniform valuations.*

Gurobi. We handled continuous valuations by discretizing the value into five bins per item (resulting in $\approx 10^5$ decision variables and $\approx 4 \times 10^6$ constraints) and rounding a continuous input valuation profile to the nearest discrete profile for evaluation.

The results are shown in Figure 1.11. We also report the violations in IR constraints incurred by the LP on the test set; for L valuation profiles, this is measured by $\frac{1}{Ln} \sum_{\ell=1}^{L} \sum_{i \in N} \max\{u_i(v^{(\ell)}), 0\}$. Due to the coarse discretization, the LP approach suffers significant IR violations (and as a result yields higher revenue). We were not able to run an LP for this setting for finer discretizations in more than one week of compute time. In contrast, RegretNet yields much lower regret and no IR violations (as the neural network satisfies IR by design), and does so in just around nine hours. In fact, even for the larger Settings D–E, the running time of RegretNet was less than 13 hours. This is 4 times faster than naive LP approach for the smaller setting.

1.6 Conclusion

Neural networks have been deployed successfully for exploration in other contexts, e.g., for the discovery of new drugs [GB+18]. We believe that there is ample opportunity for applying deep learning in the context of economic design. We have demonstrated how standard pipelines can re-discover and empirically surpass the analytical and computational progress in optimal auction design that has been made over the past 30-40 years. While our approach can easily solve problems that are orders of magnitude more complex than could previously be solved with the standard LP-based approach, a natural next step would be to scale this approach further to industry scale (e.g., through standardized benchmarking suites and innovations in network architecture). We also see promise for the framework in this work in advancing economic theory, for example in supporting or refuting conjectures and as an assistant in guiding new economic discovery.

Part II

Extensions of RegretNet

Chapter 2

Deep Learning for Revenue-Optimal Auctions with Budgets

2.1 Introduction

The design of revenue-optimal auctions in settings where bidders have private budget constraints is important yet poorly understood problem. Budget constraints arise when bidders have financial constraints that prevent them from making payments as large as their value for items. They are important in many economic settings, including spectrum auctions and land auctions, and are an integral part of the kinds of expressiveness provided to bidders in internet advertising [CB+15; Ash+10].

The design problem is not fully understood even for selling a single item. The technical challenge arises because this is a multi-dimensional mechanism design problem: a bidder's private information is her value for an item as well as her budget. This provides an obstacle to using Myerson's [Mye81] characterization results. Even for selling a single item and with two bidders, the optimal dominant-strategy incentive compatible (DSIC) design with private budget constraints is not known. No revenue-optimal designs are known for selling two or more items to even a single bidder.

In this paper, we build upon the RegretNet (Chapter 1), and use deep neural networks for the automated design of optimal auctions with budget constraints. We represent an auction as a feed-forward neural network, and optimize its parameters to maximize expected revenue. We need to include design constraints, namely *individual rationality* (IR), *budget constraints* (BC) and *incentive compatibility* (IC).[1] To the best of our knowledge, this is the first paper on automated mechanism design for settings with private budget constraints.

We design both approximately DSIC and Bayesian Incentive Compatible (BIC) auctions. In DSIC auctions, reporting truthfully is the optimal strategy for a bidder no matter what the reports of others. In a BIC auction, truth-telling is the optimal strategy for a bidder in expectation with respect to the types of others, and given that the other bidders report truthfully. The literature has also considered two additional variations in the context of budget constraints: *conditional IC* and *unconditional IC* [CG00]. We can support both of these within our framework.

2.1.1 Main Contributions

Our main contributions are summarized below:

- We extend the *RegretNet* framework (Chapter 1) to incorporate budget constraints, as well as, handle BIC and conditional IC constraints. A new aspect is that the utility of an agent can be unbounded in the presence of budgets (whenever an agent's payment exceeds her budget, her utility goes to negative infinity). To handle this, we refine the definition of regret to filter out misreports that would lead to budget violations.

- We show that our approach can be used to design new auctions with high revenue, including for the problem of selling multiple identical items to bidders with additive valuations and selling multiple distinct items to bidders with unit-demand valuations. In both cases, we consider continuous valuation distributions, which is a setting for which the problem cannot be solved through linear programming.

- We benchmark our approach in single-item settings for which analytical solutions exist, showing that neural networks can be used to learn essentially optimal auctions [CG00; MV08; PV14].

[1] We consider hard budget constraints for bidders, which means no bidder can pay more than her budget regardless of the bidder's value for the allocation. The literature also considers the case of soft budget constraints, where the bidders are allowed to gain additional funds from markets [KMR03].

2.1.2 Related Work

The high-level approach that we follow is one of *automated mechanism design* (AMD) [CS02]. Early approaches to AMD involved the use of integer programs, and did not scale up to large settings, or heuristics to search over specialized classes of mechanisms known to be IC [SL15]. In recent years, efficient algorithms have been developed for BIC design, but they do not address problems with budget constraints or problems of DSIC design [CDW12a; CDW12b; CDW13]

The use of machine learning for AMD was introduced by Dütting et al. [Düt+14], who use support vector machines for learning payment rules but not allocation rules, seeking payments that make the resulting mechanism maximally IC. Narasimhan et al. [NAP16] also use support vector machines to learn social choice and matching rules from a restricted class of mechanisms. Narasimhan and Parkes [NP16] develop a statistical framework for learning assignment mechanisms without providing a computational procedure. We first propose the use of deep neural networks for the automated design of optimal auctions in Chapter 1. This approach, which we extend in this work, is more general, does not require specialized characterization results, and uses off-the-shelf deep learning tools. RegretNet has inspired a lot of follow-up work, e.g., Golowich, Narasimhan, and Parkes [GNP18], Shen, Tang, and Zuo [STZ19], Rahme et al. [Rah+20], Rahme, Jelassi, and Weinberg [RJW20], and Curry et al. [Cur+20].

Che and Gale [CG00] design the optimal single-item auction for a single bidder. Pai and Vohra [PV14] design the optimal BIC auction for a single item and multiple bidders.[2] Malakhov and Vohra [MV08] design the optimal auction for a single-item setting with two bidders, but consider a weaker, constrained form of DSIC. Che and Gale [CG98] develop a revenue ranking of three standard single-item auctions. Maskin [Mas00] and Laffont and Robert [LR96] consider the problem of bidders with identical, known budgets.

In regard to approximation results: Borgs et al. [Bor+05] provide a multi-unit auction for private budget constraints with revenue that converges to the optimal, posted-price auction

[2] They focus on the case of independent values and budgets, but mention that they can handle positive correlation in budget and value.

in the limit of a large population of bidders. Bhattacharya et al. [Bha+12] propose a constant approximation for revenue for selling multiple items to additive bidders with private budgets (BIC) and publicly known budgets (DSIC) respectively, adopting an approach that use linear programming relaxations. Chawla et al. [CMM11] propose a multi-item auction with a constant approximation for revenue for bidders with identical, known budgets.

Budget constraints have been handled for the setting of allocative efficiency, with positive results for various multi-item settings, including for bidders with unit-demand valuations [DLN08; Ash+10; DHW11; AB09b; DHS15].[3]

2.2 Problem Setup

In this section, we describe the problem setup, starting with the simpler setting of single-item auctions.

2.2.1 Single-item auctions

There are n risk neutral bidders interested in a single indivisible good. Each bidder has a private (unknown to other bidders) *value* $v_i \in \mathbb{R}_{\geqslant 0}$ for the item, and a private *budget* $b_i \in \mathbb{R}_{\geqslant 0}$ on the amount she can pay. We let $t_i = (v_i, b_i)$ denote the *type* of bidder i and use $t = (t_1, t_2, ..., t_n)$ to denote a *type profile*. Let $\mathcal{T}_i$ denote the space of possible types for bidder i, and $\mathcal{T}$ the space of type profiles. We assume that bidder i's type is drawn from distribution F_i, and that F_i is known to both the auctioneer and, in the case of BIC, the other bidders. Let $F = \prod_{i=1}^{n} F_i$ and $F_{-i} = \prod_{j \neq i} F_j$. Further, let $v_{-i} = (v_1, ..., v_{i-1}, v_{i+1}, ..., v_n)$ denote the valuation profile without v_i, $b_{-i} = (b_1, ..., b_{i-1}, b_{i+1}, ..., b_n)$ denote the budget profile without b_i, and $t_{-i} = (v_{-i}, b_{-i})$.

Each bidder reports (perhaps untruthfully) a value and budget. An auction (a, p) consists of a *randomized allocation rule* $a : \mathcal{T} \to [0, 1]^n$ and a *payment rule* $p : \mathcal{T} \to \mathbb{R}^n_{\geqslant 0}$. Given a reported type profile $t' \in \mathcal{T}$, $a_i(t') \in [0, 1]$ denotes the probability of bidder i being allocated the item and $\sum_{i=1}^{n} a_i(t') \leqslant 1$, and $p_i(t) \in \mathbb{R}_{\geqslant 0}$ denotes the expected payment by bidder i.[4]

---

[3] The VCG mechanism is not incentive compatible for the budget-constrained case, even when modified in the natural way to truncate valuations by a bidder's budget.

[4] This is equivalent in expectation to charging each agent i a payment $p_i(t')/a_i(t')$ when she wins the auction

The utility of bidder i with type $t_i = (v_i, b_i)$ for a reported type profile $t' \in \mathcal{T}$ is the difference between the value and payment if the payment is within the budget, and $-\infty$ otherwise:

$$u_i(t_i, t') = \begin{cases} v_i \cdot a_i(t') - p_i(t') & \text{if } p_i(t') \leqslant b_i, \\ -\infty & \text{if } p_i(t') > b_i. \end{cases} \tag{2.1}$$

We consider auctions (a, p) that satisfy the *budget constraints (BC)*, i.e. charge each agent no more than her budget:

$$\forall i \in [n], t \in \mathcal{T}: \; p_i(t) \leqslant b_i \tag{BC}$$

An auction that satisfies these budget constraints is *dominant strategy incentive compatible* (DSIC) if no bidder can strictly improve her utility by misreporting her type, i.e.[5]

$$\forall i \in [n], t \in \mathcal{T}, t_i' \in \mathcal{T}_i: \; u_i(t_i, (t_i, t_{-i})) \geqslant u_i(t_i, (t_i', t_{-i})). \tag{DSIC}$$

The *revenue* from an auction is $\sum_i p_i(t)$. We are interested in designing auctions that maximize expected revenue, while satisfying BC as well as ensuring *ex post individual rationality* (IR), i.e. that each bidder receives non-zero utility for participating:

$$\forall i \in [n], t \in \mathcal{T}: \; u_i(t_i, (t_i, t_{-i})) \geqslant 0. \tag{IR}$$

We will also be interested in the design of BIC auctions because this will provide for benchmarking against some known analytical results. In practice, DSIC auctions are more preferred, at least when the effect on achievable revenue relative to BIC designs is small (and there are no other robustness concerns such as those that can arise in DSIC combinatorial auctions [AM06]) , because they are more robust— the equilibrium does not rely on common knowledge of the type distribution or common knowledge of rationality.

For *Bayesian incentive compatibility* (BIC), define the *interim allocation* for bidder i and report t_i' as $\mathcal{A}_i(t_i') = \mathbf{E}_{t_{-i} \sim F_{-i}}[a_i(t_i', t_{-i})]$ and the *interim payment* as $\mathcal{P}_i(t_i') = \mathbf{E}_{t_{-i} \sim F_{-i}}[p_i(t_i', t_{-i})]$. Given

and 0 otherwise.

[5]This inequality is well-defined for an auction that satisfies the budget constraints.

this, we can define the *interim utility function* for a bidder with type t_i and reported type t_i' as:

$$\mathcal{U}_i(t_i, t_i') = \begin{cases} v_i \mathcal{A}_i(t_i') - \mathcal{P}_i(t_i') & \text{if } \mathcal{P}_i(t_i') \leqslant b_i, \\ -\infty & \text{if } \mathcal{P}_i(t_i') > b_i. \end{cases} \tag{2.2}$$

An auction (a, p) satisfies *interim budget constraints* if

$$\forall i \in [n], t_i \in \mathcal{T}_i : \ \mathcal{P}_i(t_i) \leqslant b_i. \tag{interim BC}$$

In addition, an auction satisfying *interim* budget constraints is BIC if:

$$\forall i \in [n], t_i, t_i' \in \mathcal{T}_i : \ \mathcal{U}_i(t_i, t_i) \geqslant \mathcal{U}_i(t_i, t_i') \tag{BIC}$$

Pai and Vohra [PV14] show that, for any BIC auction that satisfies *interim* budget constraints defined here, there exists an auction with the same revenue that satisfies BIC for which the largest payment in the support of the interim payment rule is never greater than an agent's reported budget.

We will also insist that auctions that are BIC satisfy the property of *interim individual rationality*:

$$\forall i \in [n], t_i \in \mathcal{T}_i : \ \mathcal{U}_i(t_i, t_i) \geqslant 0 \tag{interim IR}$$

There is also a weaker form of both DSIC and BIC, referred to as *conditional incentive compatibility* [CG00]. Conditional IC assumes that bidders can only underreport their budgets, and thus removes one direction of the incentive constraints. DSIC and BIC become, respectively,

$$\begin{aligned} &\forall i \in [n], t \in \mathcal{T}, t_i' \in \mathcal{T}_i : \\ &\qquad u_i(t_i, (t_i, t_{-i})) \geqslant u_i(t_i, (t_i', t_{-i})) \text{ if } b_i' \leqslant b_i \end{aligned} \tag{C-DSIC}$$

$$\forall i \in [n], t_i, t_i' \in \mathcal{T}_i : \ \mathcal{U}_i(t_i, t_i) \geqslant \mathcal{U}_i(t_i, t_i') \text{ if } b_i' \leqslant b_i \tag{C-BIC}$$

Conditional IC is motivated by settings in which the auctioneer can require each bidder to post a bond that is equal to her reported budget. Where this is not practical, the more typical, unconditional IC properties are required.

2.2.2 Multi-item auctions

We also consider a multi-item setting, with both additive and unit-demand valuations on items.

In the additive setting, there are m identical units of an item for sale, and each bidder i has a private value $v_i \in \mathbb{R}_{\geqslant 0}$ for each unit of an item, and a private budget $b_i \in \mathbb{R}_{\geqslant 0}$ on the payment. Here the valuation of bidder i for k units of the item is $k \cdot v_i$.

An allocation rule $a : \mathbb{R}^{2n}_{\geqslant 0} \to [0,1]^{nm}$ maps a type profile $t' \in \mathbb{R}^{2n}_{\geqslant 0}$ to a matrix of allocation probabilities $a(t') \in [0,1]^{nm}$, where $a_{ij}(t') \in [0,1]$ denotes the probability of bidder i being allocated the j-th unit of the item, and $\sum_i a_{ij}(t') \leqslant 1,\ \forall j \in [m]$. The payment rule $p : \mathbb{R}^{2n}_{\geqslant 0} \to \mathbb{R}^{n}_{\geqslant 0}$ defines the expected payment $p_i(t')$ for each bidder.[6] The utility of a bidder is given by:

$$u_i(t_i, t') = \begin{cases} \displaystyle\sum_{j=1}^{m} v_{ij} a_{ij}(t') - p_i(\hat{t}) & \text{if } p_i(t') \leqslant b_i, \\ -\infty & \text{if } p_i(t') > b_i. \end{cases} \tag{2.3}$$

In the unit-demand setting, there are multiple distinct items $\{1, \ldots, m\}$ for sale, and each bidder i has a private value $v_{ij} \in \mathbb{R}_{\geqslant 0}$ for each item j, and a private budget b_i. A bidder's valuation for a bundle of items T is the value of the most-valued item in the bundle: $v_i(T) = \max_{j \in T} v_{ij}$. Let $t_i = (v_{i1}, \ldots, v_{im}, b_i)$ denote bidder i's type. The allocation rule $a : \mathbb{R}^{n(m+1)}_{\geqslant 0} \to [0,1]^{nm}$ maps a type profile $t' \in \mathbb{R}^{n(m+1)}_{\geqslant 0}$ to the probabilities $a_{ij}(t')$ that each bidder i is allocated item j probabilities, and the payment rule $p : \mathbb{R}^{n(m+1)}_{\geqslant 0} \to \mathbb{R}^{n}_{\geqslant 0}$ outputs the expected payments.

For revenue maximization with unit-demand bidders, it is sufficient to consider allocation rules that allocate at most one item to each bidder. Here we require the matrix of allocation probabilities to be doubly stochastic, i.e. to satisfy $\sum_j a_{ij}(t') \leqslant 1,\ \forall i \in [n]$ and $\sum_i a_{ij}(t') \leqslant 1,\ \forall j \in [m]$ for all t'. Such a randomized allocation can be decomposed into a lottery over deterministic, feasible assignments (the Birkhoff-von Neumann theorem [Bir46; Neu53]). The utility of a unit-demand bidder under a doubly stochastic allocation a is again given by (2.3).

[6] If the payment rule p is *ex post* IR, for any reported type t', there exists a set of payments $P_{ij}(t')$ on each outcome (i, j) s.t. each $P_{ij}(t') \leqslant v_{ij}$, which are equivalent in expectation to $p_i(t')$. These payments can be computed by solving a linear program.

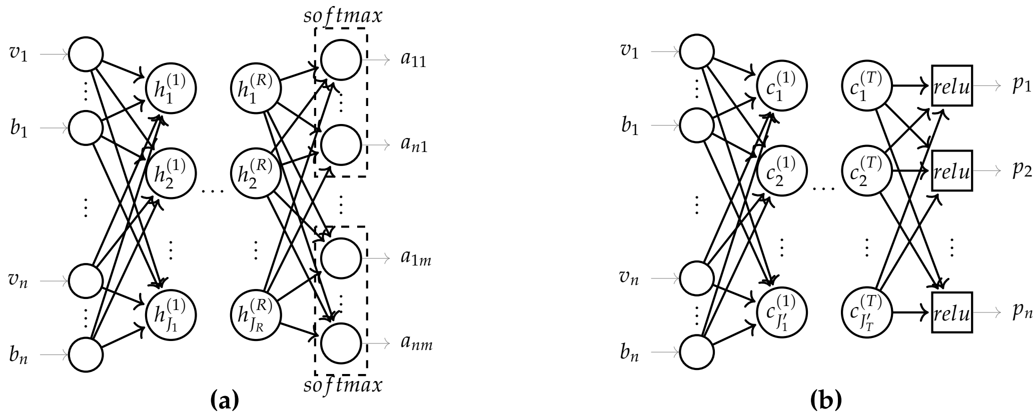


Figure 2.1: *Budgeted RegretNet: (a) Allocation rule a and (b) Payment rule p for a setting with m identical items and n additive buyers.*

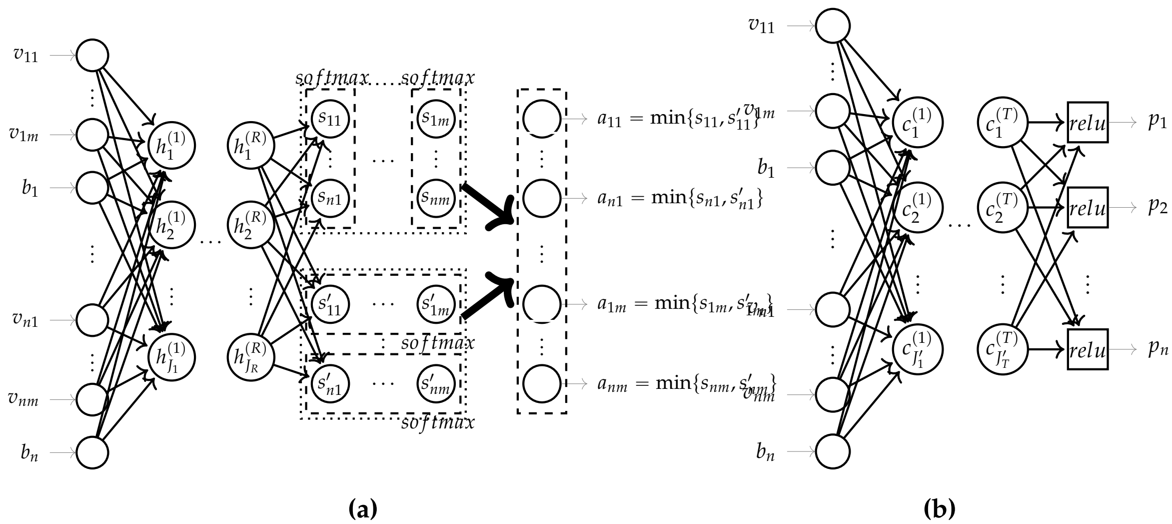


Figure 2.2: *Budgeted RegretNet: (a) Allocation rule a and (b) Payment rule p for a setting with m distinct items and n unit-demand buyers.*

2.3 The Budgeted RegretNet Framework

In this section, we explain how to extend the *RegretNet framework* of Dütting et al. [Düt+19b], which was developed and applied for settings without budget constraints, to a setting with budget constraints.

We represent an auction as a feed-forward neural network, and optimize the parameters to maximize revenue subject to regret, IR and budget constraints. While the framework of Dütting et al. enforces DSIC by requiring that the (empirical) *ex post* regret for the neural network be zero, we are able to handle more general forms of incentive compatibility by working with an appropriate notion of regret. For BIC, we constrain the (empirical) *interim* regret of the network to be zero; for conditional DSIC/BIC, we constrain the (empirical) conditional regret of the network to be zero. We additionally include budget constraints.

2.3.1 Network architecture

The allocation and payment rules are represented as separate feed-forward networks, but trained simultaneously, and connected through training loss function and constraints. The network architectures are shown in Figure 2.1 for the additive setting and in Figure 2.2 for the unit-demand setting.

Allocation network: The allocation rule for the additive setting takes a type profile t as input and outputs the probability $a_{ij}(t) \in [0, 1]$ of the j-th unit of the item being assigned to each bidder i. The neural network consists of R fully-connected hidden layers, with sigmoid activations and a fully-connected output layer. In the case of additive bidders, the output layer computes a real-valued score s_{ij} for each bidder-item pair (i, j) and converts these scores to allocation probabilities using a softmax function: $a_{ij}(t) = \frac{e^{s_{ij}}}{\sum_{k=1}^{n+1} e^{s_{kj}}}$, where $s_{n+1,j}$ is an additional "dummy score" computed for each item j. Through the inclusion of this dummy score, the softmax ensures that $\sum_{i=1}^{n} a_{ij}(t) \leqslant 1$ for each item j. The network can allocate multiple units to a single bidder.

For unit-demand bidders, we require the allocation probabilities to be doubly stochastic. For this, we modify the allocation network to generate a score s_{ij} and a score s'_{ij} for each bidder-item pair (i, j), with the first set of scores normalized along the rows, and the second set of scores normalized along the columns using softmax functions. The final allocation is an element-wise minimum of the two sets of normalized scores, $a_{ij}(t) = \min\left\{\frac{e^{s_{ij}}}{\sum_{k=1}^{n+1} e^{s_{kj}}}, \frac{e^{s'_{ij}}}{\sum_{k=1}^{m+1} e^{s'_{jk}}}\right\}$, and is guaranteed to be doubly stochastic.

Payment network: The payment rule is also defined through a feed-forward network, and consists of T fully-connected hidden layers, with sigmoid activations and a fully-connected output layer. Given an input type profile t, the neural network computes a payment $p_i(t)$ for each bidder i. In particular, the output layer computes a score $s'_i \in \mathbb{R}$ for each bidder, and applies the *ReLU activation function* to ensure that payments are non-negative: $p_i(t) = \max\{s'_i, 0\}$.

2.3.2 Training problem

We use the following metrics to measure the degree to which an auction violates the BIC, IR and BC constraints.

Regret: We define the expected *interim* regret to bidder i, for an auction with rules (a, p), as the maximum gain in *interim* utility by misreporting the bidder's type.

$$\begin{aligned} rgt_i(a,p) \ = \\ \mathbf{E}_{t_i \sim F_i}\left[\max_{t_i' \in \mathcal{T}_i} \chi_{(\mathbb{P}_i(t_i') \leqslant b_i)} \left(\mathcal{U}_i(t_i, t_i') - \mathcal{U}_i(t_i, t_i)\right)\right], \end{aligned} \tag{2.4}$$

where χ_A is an indicator function for whether predicate A is true. An auction is BIC if and only if it has zero *interim* regret. The indicator function in the above expression ensures that the first utility term does not go to $-\infty$. As long as the auction also satisfies interim BC, the second utility term is also finite for all type profiles, thus ensuring that the regret is always finite.

IR penalty: The penalty for violating IR for bidder i is given by:

$$irp_i(a,p) = \mathbf{E}_{t_i \sim F_i}\left[\max\{0, -\mathcal{U}_i(t_i, t_i)\}\right]. \tag{2.5}$$

BC penalty: The penalty for violating the budget constraint for bidder i is given by:

$$bcp_i(a,p) = \mathbf{E}_{t_i \sim F_i}\left[\max\{0, \mathbb{P}_i(t_i) - b_i\}\right]. \tag{2.6}$$

Further, we define the *loss function* as the negated expected revenue $\mathcal{L}(a,p) = -\mathbf{E}_{t\sim F}\left[\sum_{i=1}^n p_i(t)\right]$.

Let $w \in \mathbb{R}^d$ denote the parameters of the allocation network, the induced allocation rule denoted by a^w, and $w' \in \mathbb{R}^{d'}$ denote the parameters of the payment network, the induced payment rule denoted by $p^{w'}$.

The design objective is to minimize the loss function over the space of network parameters, such that the regret, IR penalty and BC penalty is zero for each bidder:

$$\begin{aligned}
&\min_{w \in \mathbb{R}^d, w' \in \mathbb{R}^{d'}} \mathcal{L}(a^w, p^{w'}) \\
s.t. \quad & rgt_i(a^w,\ p^{w'}) = 0, \forall i \in [n] \\
& irp_i(a^w,\ p^{w'}) = 0, \forall i \in [n] \\
& bcp_i(a^w,\ p^{w'}) = 0, \forall i \in [n].
\end{aligned} \tag{OP1}$$

In practice, the loss, regret, IR penalty and BC penalty can be estimated from a sample of type profiles $S = \{t^{(1)}, t^{(2)}, ..., t^{(L)}\}$ drawn i.i.d. from F. The loss for an auction with rules (a, p) can be estimated as $\widehat{\mathcal{L}}(a, p) = -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i=1}^{n} p_i\big(t^{(\ell)}\big)$.

To estimate the *interim* regret, for each type profile $t^{(\ell)}$ in S, we draw additional samples $S_\ell = \{\tilde{t}^{(1)}, \ldots, \tilde{t}^{(K)}\}$ from F, and $S'_\ell = \{\bar{t}^{(1)}, \ldots, \bar{t}^{(K')}\}$ from a uniform distribution over type space $\mathcal{T}$.[7] Using sample S_ℓ, we define the empirical *interim* utility for bidder i with type t_i and report t'_i as:

$$\widehat{\mathcal{U}}_i(t_i, t'_i) \ = \ \frac{1}{K}\sum_{k=1}^{K} u_i\left(t_i, \big(t'_i, \tilde{t}^{(k)}_{-i}\big)\right)$$

and the empirical *interim* payment as:

$$\widehat{\mathcal{P}}_i(t'_i) \ = \ \frac{1}{K}\sum_{k=1}^{K} p_i\left(t'_i, \tilde{t}^{(k)}_{-i}\right)$$

Then the empirical *interim* regret is given by:

$$\begin{aligned}
\widehat{rgt}_i(a, p) = \frac{1}{L}\sum_{\ell=1}^{L} \max_{t' \in S'_\ell} \Big\{ & \chi_{\left(\widehat{\mathcal{P}}_i(t'_i) \leq b_i^{(\ell)}\right)} \\
& \cdot \left(\widehat{\mathcal{U}}_i\big(t_i^{(\ell)}, t'_i\big) - \widehat{\mathcal{U}}_i\big(t_i^{(\ell)}, t_i^{(\ell)}\big)\right) \Big\},
\end{aligned} \tag{2.7}$$

where the sample S'_ℓ provides a set of deviating type profiles to approximate the maximum over bidder misreports.

---

[7] The deviating types need not be sampled from the distributions of true types. We adopt a uniform sampling scheme, and find this to be effective in our experiments.

The IR and BC penalties can be similarly estimated as:

$$\widehat{irp}_i(a,p) = \frac{1}{L}\sum_{\ell=1}^{L} \max\left\{0, -\widehat{\mathcal{U}}_i\big(t_i^{(\ell)}, t_i^{(\ell)}\big)\right\}$$
$$\widehat{bcp}_i(a,p) = \frac{1}{L}\sum_{\ell=1}^{L} \max\left\{0, \widehat{\mathcal{P}}_i\big(t_i^{(\ell)}\big) - b_i^{(\ell)}\right\}.$$

Following RegretNet in Chapter 1, we use the *Augmented Lagrangian method* to solve the resulting sample-based optimization problem:

$$\begin{aligned}
&\min_{w\in\mathbb{R}^d, w'\in\mathbb{R}^{d'}} \widehat{\mathcal{L}}(a^w, p^{w'}) \\
s.t.\quad & \widehat{rgt}_i(a^w,\, p^{w'}) = 0, \forall i \in [n] \\
& \widehat{irp}_i(a^w,\, p^{w'}) = 0, \forall i \in [n] \\
& \widehat{bcp}_i(a^w,\, p^{w'}) = 0, \forall i \in [n].
\end{aligned} \tag{OP2}$$

Augmented Lagrangian Solver: The solver formulates a sequence of unconstrained optimization steps that combine the revenue, regret, IR penalty, and budget penalty terms into a single objective, with the relative weights on the regret, IR and budget penalty terms adjusted across iterations. More specifically, the solver constructs the following unconstrained, augmented Lagrangian objective:

$$\begin{aligned}
&\mathcal{F}_\rho(w, w'; \lambda_{rgt}, \lambda_{irp}, \lambda_{bcp}) \\
&= \widehat{\mathcal{L}}(a^w, p^{w'}) + \sum_{i\in[n]} \lambda_{rgt,i}\, \widehat{rgt}_i(a^w, p^{w'}) + \frac{\rho}{2}\sum_{i\in[n]} \widehat{rgt}_i^2(a^w, p^{w'}) \\
&\quad + \sum_{i\in[n]} \lambda_{irp,i}\, \widehat{irp}_i(a^w, p^{w'}) + \frac{\rho}{2}\sum_{i\in[n]} \widehat{irp}_i^2(a^w, p^{w'}) \\
&\quad + \sum_{i\in[n]} \lambda_{bcp,i}\, \widehat{bcp}_i(a^w, p^{w'}) + \frac{\rho}{2}\sum_{i\in[n]} \widehat{bcp}_i^2(a^w, p^{w'})
\end{aligned}$$

where $\lambda_{rgt} \in \mathbb{R}^n$, $\lambda_{irp} \in \mathbb{R}^n$ and $\lambda_{bcp} \in \mathbb{R}^n$ are vectors of Lagrangian multipliers associated with the equality constraints in (OP2), and $\rho > 0$ is a fixed parameter that controls the weight on the augmented quadratic terms.

The solver operates across multiple iterations, and updates the Lagrange multipliers based on the violation of the constraints in each iteration t:

$$\left(w^{t+1}, w'^{t+1}\right) \in \operatorname{argmin}_{(w,w')} \mathcal{F}_\rho(w, w';\, \lambda^t_{rgt}, \lambda^t_{irp}, \lambda^t_{bcp}) \tag{2.8}$$

$$\lambda^{t+1}_{rgt,i} = \lambda^t_{rgt,i} + \rho\, \widehat{rgt}_i\left(a^{w^{t+1}}, p^{w'^{t+1}}\right), \forall i \in [n], \tag{2.9}$$

$$\lambda^{t+1}_{irp,i} = \lambda^t_{irp,i} + \rho\, \widehat{irp}_i\left(a^{w^{t+1}}, p^{w'^{t+1}}\right), \forall i \in [n], \tag{2.10}$$

$$\lambda^{t+1}_{bcp,i} = \lambda^t_{bcp,i} + \rho\, \widehat{bcp}_i\left(a^{w^{t+1}}, p^{w'^{t+1}}\right), \forall i \in [n], \tag{2.11}$$

where the inner optimization in (2.8) is approximately solved through multiple iterations of the Adam solver [KB15]. Specifically, the gradient is pushed through the loss function as well as the empirical measures of violation of IC, IR and BC.[8] In our experiments, the Lagrangian multipliers are initialized to zero.

2.3.3 Handling other kinds of IC constraints

The approach also extends to a design subject to conditional BIC, as well as DSIC and conditional DSIC. For conditional BIC, we replace the regret in (OP1) with the conditional regret, defined as:

$$crgt_i(a,p) = \mathbf{E}_{t_i \sim F_i}\left[\max_{t'_i \in \mathcal{T}_i} \chi_{\left(b'_i \leq b_i\right)} \left(\mathcal{U}_i(t_i, t'_i) - \mathcal{U}_i(t_i, t_i)\right)\right], \tag{2.12}$$

and use the following estimate of the *conditional interim regret* in (OP2):

$$\begin{aligned}\widehat{crgt}_i(a,p) \; = \\ \frac{1}{L}\sum_{\ell=1}^{L} \max_{t' \in S'_\ell} \left\{\chi_{\left(b'_i \leq b_i^{(\ell)}\right)} \cdot \left(\widehat{\mathcal{U}}_i(t_i^{(\ell)}, t'_i) - \widehat{\mathcal{U}}_i(t_i^{(\ell)}, t_i^{(\ell)})\right)\right\}.\end{aligned} \tag{2.13}$$

To handle DSIC and conditional DSIC, we replace the *interim* regret in the training problem with the *ex post* regret and a conditional version of the *ex post* regret, respectively. The expected *ex post* regret to bidder i in an auction (a, p) is defined as the maximum gain in *ex post* utility

[8] The solver handles the indicator function in the regret definition by taking its gradient to be zero.

obtained by misreporting her type:

$$eprgt_i(a,p) = \mathbf{E}_{t\sim F}\Big[\max_{t'_i\in\mathcal{T}_i} \chi_{\left(p_i(t'_i,t_{-i})\leqslant b_i\right)} \left(u_i\big(t_i,(t'_i,t_{-i})\big) - u_i\big(t_i,(t_i,t_{-i})\big)\right)\Big] \tag{2.14}$$

Similarly, the *ex post* IR penalty and *ex post* BC penalty can be defined as:

$$epirp_i(a,p) = \mathbf{E}_{t\sim F}\left[\max\{0, -u_i(t_i,(t_i,t_{-i}))]\}\right] \tag{2.15}$$

$$epbcp_i(a,p) = \mathbf{E}_{t\sim F}\left[\max\{0, p_i(t_i,(t_i,t_{-i})) - b_i\}\right] \tag{2.16}$$

To estimate the *ex post* regret, we use a set of deviating (misreport) samples $S'_\ell = \{\bar{t}^{(1)},\ldots,\bar{t}^{(K')}\}$, drawn from a uniform distribution over $\mathcal{T}$:

$$\begin{aligned}\widehat{eprgt}_i(a,p) = \frac{1}{L}\sum_{\ell=1}^{L}\max_{t'\in S'_\ell}\Big\{&\chi_{\left(p_i(t'_i,t^{(\ell)}_{-i})\leqslant b^{(\ell)}_i\right)}\\ &\cdot\Big(u_i\big(t^{(\ell)}_i,(t'_i,t^{(\ell)}_{-i})\big) - u_i\big(t^{(\ell)}_i,(t^{(\ell)}_i,t^{(\ell)}_{-i})\big)\Big)\Big\}.\end{aligned} \tag{2.17}$$

The *ex post* IR and BC penalties can be estimated as:

$$\widehat{epirp}_i(a,p) = \frac{1}{L}\sum_{\ell=1}^{L}\max\Big\{0, -u_i\big(t^{(\ell)}_i,(t^{(\ell)}_i,t^{(\ell)}_{-i})\big)\Big\} \tag{2.18}$$

$$\widehat{epbcp}_i(a,p) = \frac{1}{L}\sum_{\ell=1}^{L}\max\Big\{0, p_i\big(t^{(\ell)}_i,t^{(\ell)}_{-i}\big) - b^{(\ell)}_i\Big\} \tag{2.19}$$

For the conditional *ex post* regret, we replace $\chi_{\left(p_i(t'_i,t_{-i})\leqslant b_i\right)}$ in $eprgt_i$ by $\chi_{\left(b'_i\leqslant b_i\right)}$. Similarly, in the empirical version of this quantity $\widehat{eprgt}_i$, we replace $\chi_{\left(p_i(t'_i,t^{(\ell)}_{-i})\leqslant b^{(\ell)}_i\right)}$ by $\chi_{\left(b'_i\leqslant b^{(\ell)}_i\right)}$.

2.4 Experimental Results

We present experimental results to show that we can find new auctions for settings where the optimal design is unknown, and also recover essentially optimal DSIC and BIC auctions in a variety of simpler settings for which analytical results are available. Since DSIC is a stronger property than BIC, and preferred in practice, we give more focus to the automated design of DSIC auctions.

Experimental setup. We use the *TensorFlow* deep learning library for experiments. We solve the inner optimization in the augmented Lagrangian method using the ADAM solver [KB15], with a learning rate of 0.001 and a mini-batch size of 64. All the experiments are run on a compute cluster with NVIDIA GPU cores.

Evaluation. We generate training and test data from different type distributions, use the training set for fitting an auction network and evaluate performance of the learned auction on the test set. We use the following metrics for evaluation:

$$\begin{aligned}
\text{Regret} &= \tfrac{1}{n}\textstyle\sum_{i=1}^{n} \widehat{rgt}_i(a,p) \\
\text{Conditional Regret} &= \tfrac{1}{n}\textstyle\sum_{i=1}^{n} \widehat{crgt}_i(a,p) \\
\text{IR penalty} &= \tfrac{1}{n}\textstyle\sum_{i=1}^{n} \widehat{irp}_i(a,p) \\
\text{BC penalty} &= \tfrac{1}{n}\textstyle\sum_{i=1}^{n} \widehat{bcp}_i(a,p).
\end{aligned}$$

For experiments on DSIC auctions, the terms $\widehat{rgt}_i$, $\widehat{crgt}_i$, $\widehat{irp}_i$ and $\widehat{bcp}_i$ are *ex post* quantities. For experiments on BIC, these terms are *interim* quantities. The training and test set are large enough to avoid issues of overfitting. The specific sample sizes and network scale are provided in subsequent subsections.

2.4.1 Optimal DSIC auctions

We consider the design of DSIC auctions, adopting three different settings studied in the literature:

- Setting I: There is a single item and a single bidder, with the bidder's value $v_1 \sim \text{Unif}[0,1]$ and budget $b_1 \sim \text{Unif}[0,1]$. The optimal DSIC auction for this setting was derived by Che and Gale [CG00].

- Setting II: There is a single item amd two bidders, where $v_1, v_2 \sim Unif\{1,2,...,10\}$. The first bidder is unconstrained while the second bidder has a budget of 4. The optimal auction under conditional DSIC for this setting was derived by Malakhov and Vohra [MV08].[9]

[9] In this special case, the auctioneer knows the true budget of constrained bidder but allows her to misreport her budget. In effect, the budget of constrained bidder is publicly known.

Property	Setting	Opt	Budgeted RegretNet			
		rev	*rev*	*regret*	*irp*	*bcp*
DSIC	I	0.192	0.196	0.002 (0.003)	0.002	0.001
	II (C)	4.664	4.638	0.002	0.005	0.002
	III	–	0.709	0.002 (0.004)	0.0	0.002
	IV	–	0.287	0.002 (0.003)	0.0	0.0
BIC	II (C)	4.847	4.788	0.0	0.0	0.0
	V	0.342	0.348	0.004 (0.005)	0.001	0.0

Table 2.1: *Test metrics for Budgeted RegretNet auctions. Here (C) refers to conditional IC. For continuous valuation distributions, we also report within parenthesis the regret estimated using a larger misreport sample (i.e. with 1000 misreports for each type profile).*

Setting	Misreport sample size $|S'_\ell|$				
	100	200	400	800	1600
IV	0.0018	0.0021	0.0023	0.0026	0.0029

Table 2.2: *Test regret for Budgeted RegretNet under Setting IV with misreport samples of different sizes for each type profile.*

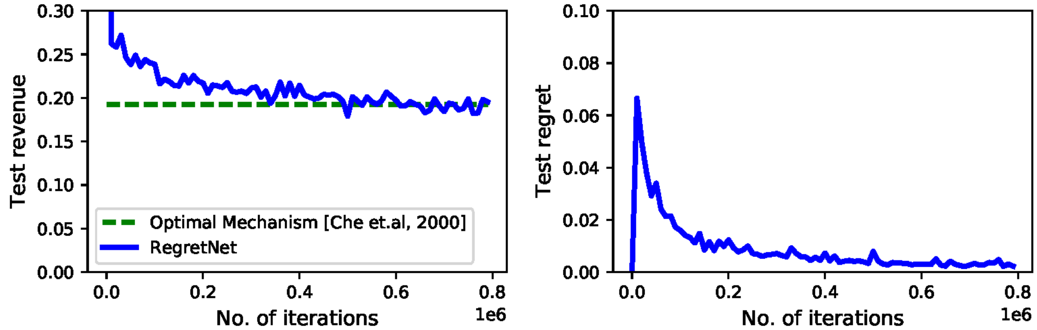
- Setting III: There are four identical items with two additive bidders where bidder i's value for each item $v_i \sim \text{Unif}[0,1]$ and the budget $b_i \sim \text{Unif}[0,1]$. There is no analytical result.

- Setting IV: There are two items with two unit-demand bidders where bidder i's value for the item j, $v_{ij} \sim \text{Unif}[0,1]$ and the budget $b_i \sim \text{Unif}[0,1]$. There is no analytical result.

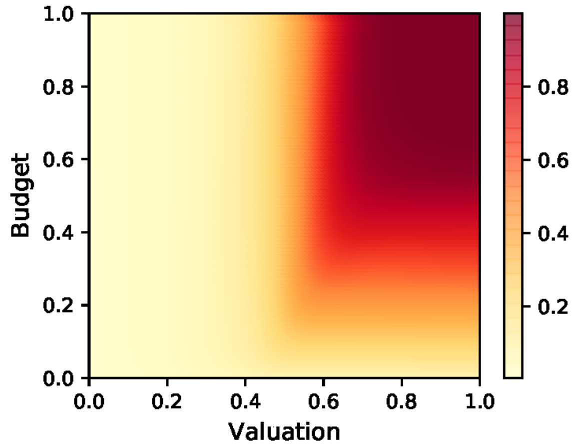
We use allocation and payment networks with two hidden layers each, and with 100 hidden nodes in each layer. For all the experiments below, for each type profile $t^{(\ell)}$, we randomly generate a sample of 100 misreports S'_ℓ to evaluate the regret. We also report the regret estimated for continuous valuation distributions using a larger misreport sample (of size 1000 or more) for each type profile.[10] A summary of our results is shown in Tables 2.1 and 2.2.

For setting I, we use a training and test sample of 5000 profiles each, with the parameter ρ in Augmented Lagrangian solver set to 0.01. Figure 2.3(a) presents plots of test revenue and test *ex post* regret for the learned auction as a function of solver iterations. Figure 2.3(b)-(c) show the allocation rule learned by the neural network, and compare this with the optimal

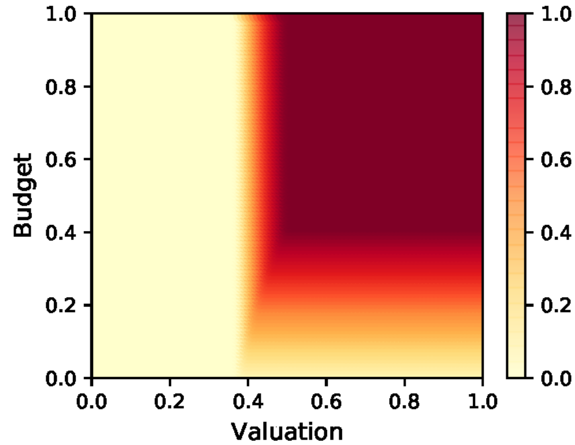
[10]For discrete valuation distributions in this paper, we find a sample of 100 misreports to be large enough to accurately estimate the regret.



(a) *Test revenue and regret*



(b) *Test allocation rule*



(c) *Optimal allocation rule*

Figure 2.3: *The auction learned under DSIC for Setting I with a single item and single bidder, where $v_1 \sim Unif[0,1]$ and $b_1 \sim Unif[0,1]$. The solid regions in (b) and (c) depict the probability of the item being allocated to the bidder.*

rule of Che and Gale [CG00]. Not only does the learned auction yields revenue close to the optimal auctions and incur negligible regret, but the learned allocation rule closely matches the optimal rule. From Table 2.1, we see that the learned auction also incurs very small IR and budget violations.

For setting II, we use a smaller training and test sample of 1000 profiles, which are large enough for the discrete distribution considered here. We set ρ to 0.001. The optimal auction for this setting is given by Malakhov and Vohra [MV08]. We trained neural network for conditional DSIC. Figure 2.4(a) shows plots of the test revenue for the learned auction, as well as plots of the test *ex post* regret for the learned auction under conditional DSIC constraints. The learned auction yields revenue very close to the optimal revenue, while yielding negligible regret, IR violations, or budget violations. Furthermore, as seen in Figure 2.4(b)-(c), the learned allocation rule for conditional DSIC closely matches the analytical result in Malakhov and Vohra [MV08].

For setting III, we use a training and test sample of 5000 profiles, with ρ set to 0.01. Since the optimal auction is not provided by the theoretical literature, we compare the learned auction rule against the optimal posted pricing auction, as well as the auction proposed by Borgs et al. [Bor+05]. Figure 2.5 shows test revenue and *ex post* regret as functions of solver iterations. In this case, the neural network is able to discover an auction with a higher revenue than the baseline, while incurring a very small regret, as well as, very small IR and budget violations.

For setting IV, we use a training and test sample of 5000 profiles, with ρ set to 0.03. Since there is no analytical result for this setting, we compare the learned auction rules against the ascending auction of Ashlagi and Braverman [AB09b]. Figure 2.6 shows the test revenue and *ex post* regret as functions of the number of solver iterations. The auction learned by RegretNet has a higher revenue than the baseline, while incurring very small regret, IR, and budget violations.

Since the regret is estimated using a sample of misreports, for this experiment, we also evaluate the regret using misreport samples S'_ℓ of different sizes. The results are summarized in Table 2.2. Figure 2.7 shows the test *ex post* regret as functions of solver iterations for different sizes of misreport samples. As seen, even with larger number of misreport samples, the regret is still very small, implying that the learned auction is indeed essentially IC.

2.4.2 Optimal BIC auctions

Next, we consider the automated design of BIC auctions. Here we focus on settings for which analytical results are available. This serves to provide a validation that we are able to use *RegretNet* to learn BIC designs. We are less interested in optimal BIC for new settings because we consider DSIC of more practical interest. We consider the following settings:

- Setting II from Section 2.4.1. The optimal BIC auction for this setting was derived by Malakhov and Vohra [MV08].
- Setting V: There is a single item and two symmetric budget constrained bidders. Each

bidder draws a value $v_i \sim \text{Unif}[0,1]$ and budget $b_i \sim \text{Unif}\{0.22, 0.42\}$. The optimal auction for this setting was derived by Pai and Vohra [PV14].

For these experiments, we use allocation and payment networks with two hidden layers with 50 nodes each.[11] A summary of the results is provided in Table 2.1. The training and test set have 1000 type profiles each and ρ was set to 0.05. To learn the BIC auctions, we need additional samples S_ℓ from known distribution F for each type profile $t^{(\ell)}$, which makes the training of RegretNet more costly than for the case of DSIC auctions.

Figure 2.8 presents the results of learning a BIC auction for setting II, providing the test revenue and test interim regret as a function of the number of solver iterations. We also illustrate the learned allocation rule, and compare it with the optimal allocation rule of Malakhov and Vohra [MV08]. Not only does the auction that is derived through machine learning achieve near-optimal revenue with essentially zero regret, IR and budget violations, but we closely recover the design of the optimal allocation rule. Figure 2.9 shows the test revenue and interim regret of the learned auction for setting V. Again, we are able to achieve almost-optimal revenue, while incurring very small regret, IR, and budget violations.

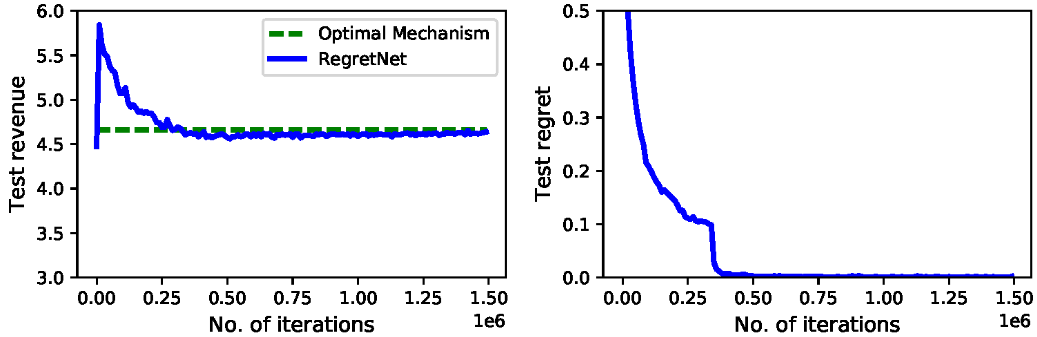
2.5 Conclusion

We have used deep learning to design essentially optimal, multi-item auctions under private budget constraints. Whereas the state-of-the-art analytical results for the design of optimal, DSIC auctions cannot handle more than two bidders, or more than one item (to even a single bidder), *RegretNet* can discover new, essentially incentive-compatible designs with high revenue in these settings (consider Setting III and Setting IV). We also validate the approach by demonstrating that *RegretNet* can recover essentially optimal designs in settings for which optimal analytical results do exist, including the case of BIC auction design.

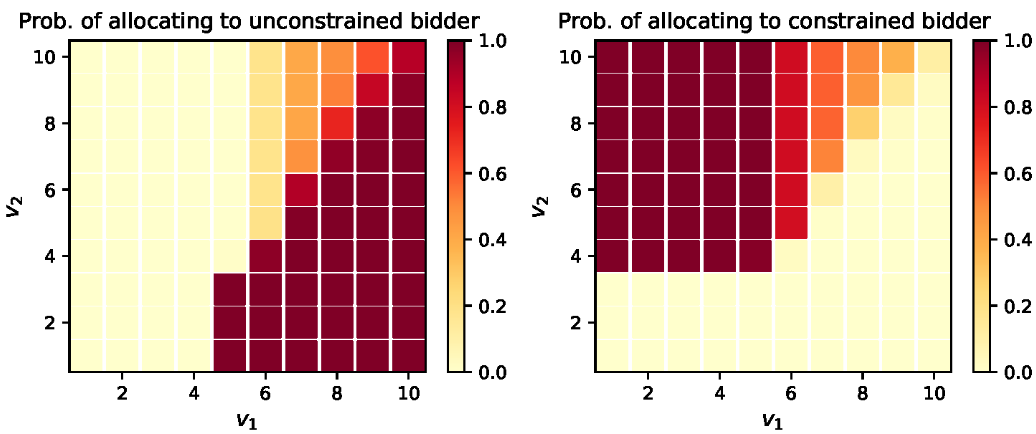
In the future, it will be interesting to study the robustness of the learned auctions to perturbations in the type distributions, develop methods that allow a single network to handle

[11] Unlike DSIC settings, we reduce the size of neural networks in BIC settings to trade-off the cost of more computation for estimating interim rules.

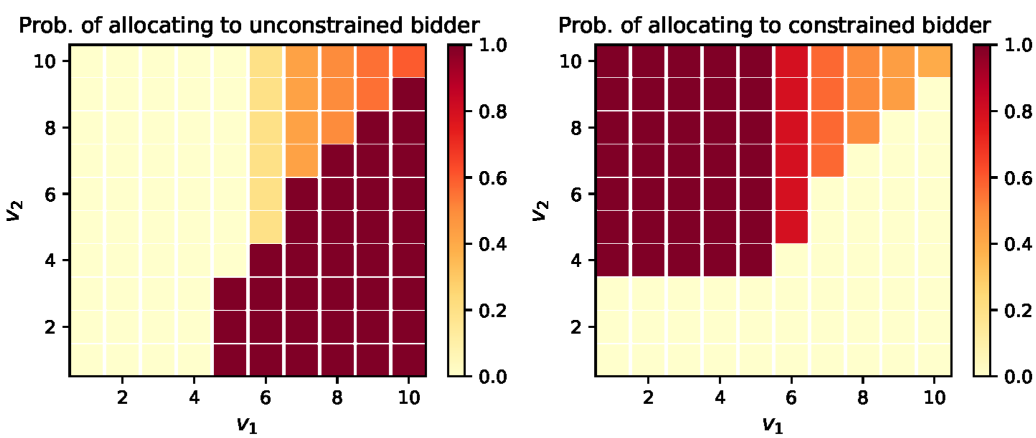
different number of bidders or items, improve the efficiency with which we can train *RegretNet* in the case of BIC design, and use our approach to estimate both upper- and lower-bounds on the revenue from exactly IC designs. It will also be interesting to explore the effect of allowing for correlation between value and budget and across bidders, soft budget constraints, and budgets that depend on a bidder's allocation. All of these seem within reach of automated methods, but are extremely challenging to handle through theoretical analysis.



(a) *Revenue and regret as a function of solver iterations*



(b) *Learned allocation rule*



(c) *Optimal allocation rule*

Figure 2.4: *The auction learned under conditional DSIC for Setting II with a single item and two bidders, where* $v_1, v_2 \sim \mathrm{Unif}\{1, 2, ..., 10\}$*, bidder 1 is unconstrained, and bidder 2 has a budget of 4.*

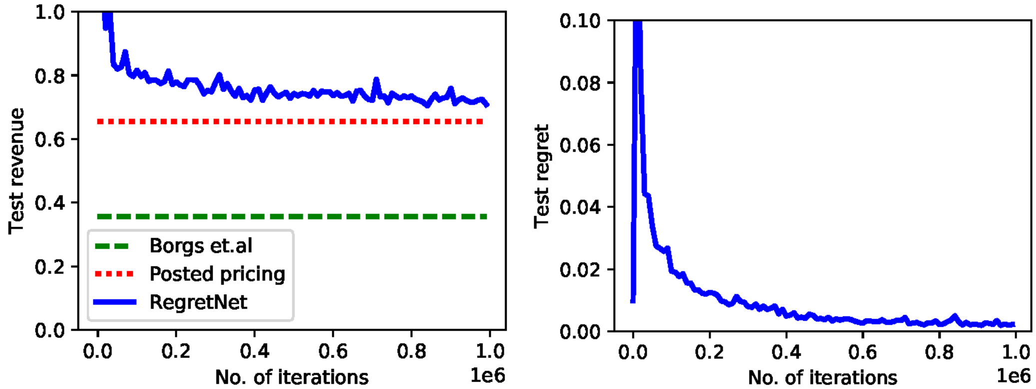


Figure 2.5: *Revenue and regret for the DSIC auction learned under Setting III with four identical items and two additive bidders, where bidder i's value for each item $v_i \sim Unif[0,1]$ and $b_i \sim Unif[0,1]$.*

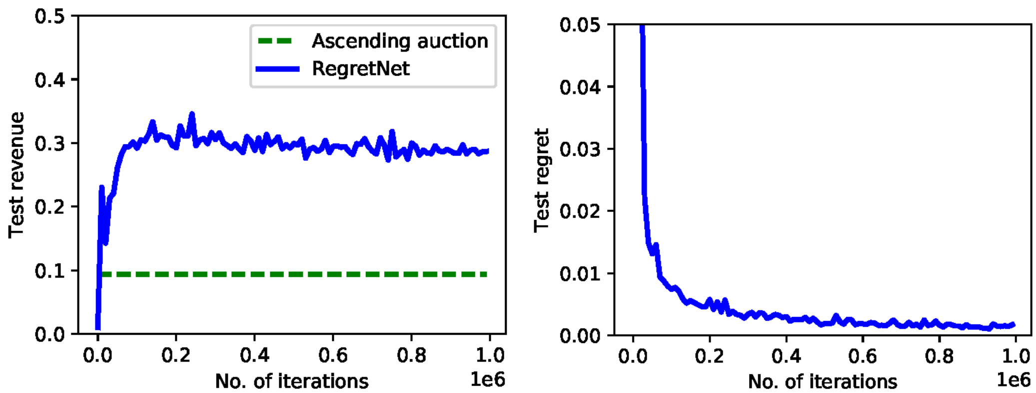


Figure 2.6: *Revenue and regret for the DSIC auction learned under Setting IV with two items and two unit-demand bidders, where bidder i's value for item j $v_{ij} \sim \mathrm{Unif}[0,1]$ and $b_i \sim \mathrm{Unif}[0,1]$.*

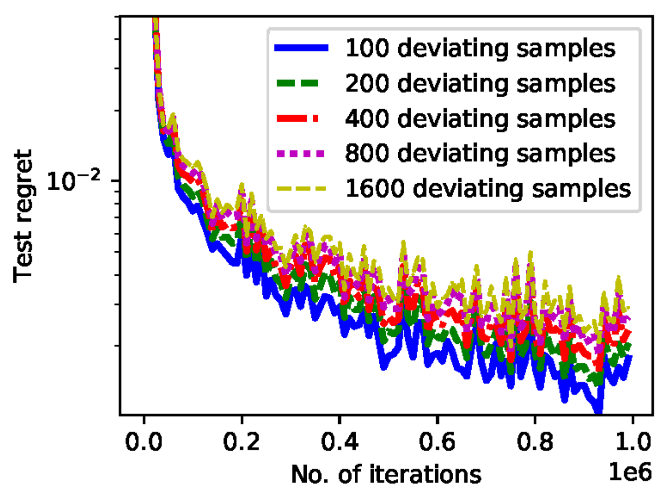
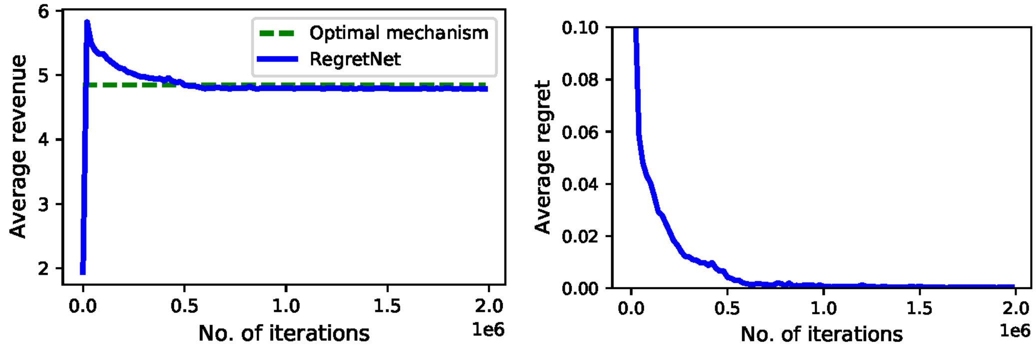
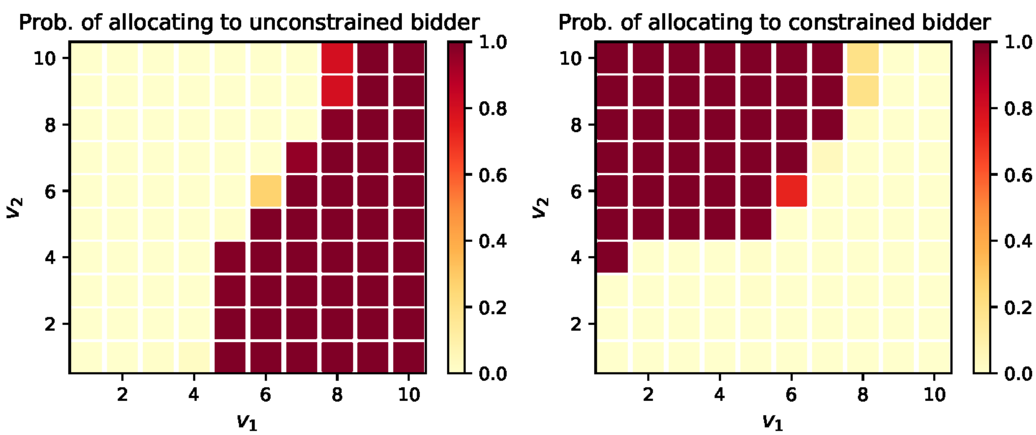


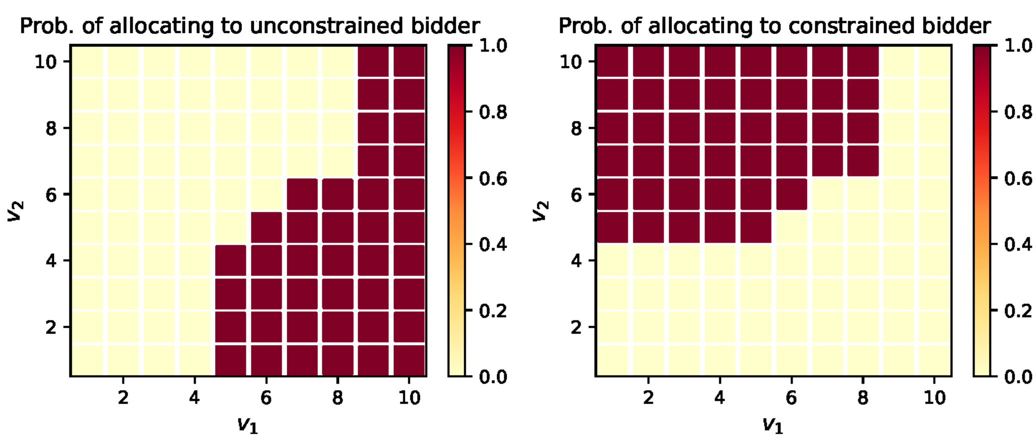
Figure 2.7: *A semi-logarithmic plot of test regret as a function of the number of iterations for different misreport sample sizes for the DSIC auction learned under Setting IV.*



(a) *Revenue and regret as a function of solver iterations*



(b) *Learned allocation rule*



(c) *Optimal allocation rule*

Figure 2.8: *Auction learned under BIC for Setting II with a single item and two bidders, where $v_1, v_2 \sim$ Unif$\{1, 2, ..., 10\}$, bidder 1 is unconstrained and bidder 2 has a budget of 4.*

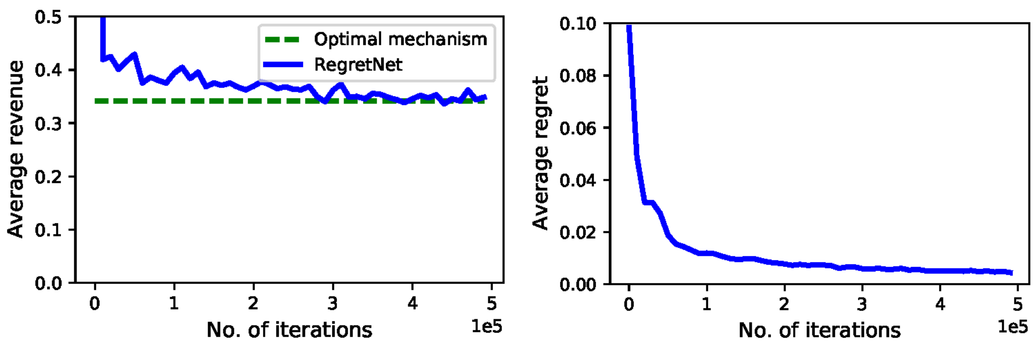


Figure 2.9: *Revenue and regret of auction learned under BIC for Setting V with a single item and two bidders, where $v_1, v_2 \sim$ Unif$[0, 1]$ and $b_1, b_2 \sim$ Unif$\{0.22, 0.42\}$.*

Chapter 3

Welfare-Preserving ε-BIC to BIC Transformation with Negligible Revenue Loss

3.1 Introduction

How should one sell a set of goods, given conflicting desiderata of maximizing revenue and welfare, and considering the strategic behavior of potential buyers? Classic results in mechanism design provide answers to some extreme points of the above question. If the seller wishes to maximize revenue and is selling a single good, then theory prescribes Myerson's optimal auction. If the seller wishes to maximize social welfare (and is selling any number of goods), then theory prescribes the Vickrey-Clarke-Groves (VCG) mechanism.

But in practical applications, one often cares about both revenue and welfare. Consider, for example, a governmental organization, which we might think of as typically trying to maximize welfare, but can also reinvest any revenue it collects from, say, a land sale, to increase welfare on a longer horizon. Similarly, a company, which we might think of as trying to maximize profit, may also care about providing value to participants for the sake of increasing future participation and, in turn, longer-term profits. Ultimately, strictly optimizing for welfare may lead to unsustainably low revenue, while strictly optimizing for revenue may lead to an unsustainably low value to participants.

Indeed, in the online advertising space, there are various works exploring this trade-off between revenue and welfare. Display advertising has focused on yield optimization (i.e., maximizing a combination of revenue and the quality of ads shown) [Bal+14], and work in sponsored search auctions has considered a squashing parameter that similarly trades off revenue and quality [LP07]. For the general mechanism design problem, however, there is a surprisingly small literature that considers both welfare and revenue together (e.g., Diakonikolas et al. [Dia+12]).

The reason for this theoretical gap is that optimal economic design is very challenging in the kinds of multi-dimensional settings where we selling multiple items, for example, such as those that arise in practice. Recognizing this there is considerable interest in adopting algorithmic approaches to economic design. These include polynomial-time black-box reductions from multi-dimensional revenue maximization to the algorithmic problem for virtual welfare optimization e.g.[CDW12a; CDW12b; CDW13], and the application of methods from linear programming [CS02; CS04] and machine learning [Düt+14; FNP18; Düt+19b] to automated mechanism design.

These approaches frequently come with a limitation: the output mechanism may only be approximately incentive compatible (IC); e.g., the black-box reductions are only approximately IC when these algorithmic problems are solved in polynomial time, the LP approach works on a coarsened space to reduce computational cost but achieves an approximately IC mechanism in the full space, and the machine learning approach trains the mechanism over finite training data that achieves approximately IC for the real type distribution.

While it is debated whether incentive compatibility may suffice, e.g., [Car12; LP12; AB19], this does add an additional layer of unpredictability to the performance of a designed mechanism. First, the fact that an agent can gain only a small amount from deviating does not preclude strategic behavior—perhaps the agent can easily identify a useful deviation, for example through repeated interactions, that reliably provides increased profit. This can be a problem when strategic responses lead to an unraveling of the desired economic properties of the mechanism (we provide such an example in this paper). The possibility of strategic reports by participants has additional consequences as well, for example making it more challenging for a designer to confidently measure ex-post welfare after outcomes are realized.

For the above reasons, there is considerable interest in methods to transform an ε-Bayesian incentive compatible (ε-BIC) mechanism to an exactly BIC mechanism [DW12; CZ17; RW18]. In this paper we alslo go beyond ε-BIC mechanisms, and also consider ε-expected ex-post IC (ε-EEIC) mechanisms [Düt+14], which is also the output of RegretNet (Chapter 1). The main question we want to answer in this paper is:

> *Given an ϵ-BIC mechanism, is there an exact BIC mechanism that maintains social welfare and achieves negligible revenue loss, compared with the original mechanism? If so, can we find the BIC mechanism efficiently?*

Model and Notation

We consider a general mechanism design setting with a set of n agents $N = \{1, \ldots, n\}$. Each agent i has a private type t_i. We denote the entire type profile as $t = (t_1, \ldots, t_n)$, which is drawn from a joint distribution $\mathcal{F}$. Let $\mathcal{F}_i$ be the marginal distribution of agent i and $\mathcal{T}_i$ be the support of $\mathcal{F}_i$. Let t_{-i} be the joint type profile of the other agents, $\mathcal{F}_{-i}$ be the associated marginal type distribution. Let $\mathcal{T} = \mathcal{T}_1 \times \cdots \times \mathcal{T}_n$ and $\mathcal{T}_{-i}$ be the support of $\mathcal{F}$ and $\mathcal{F}_{-i}$, respectively. In this setting, there is a set of feasible outcomes denoted by $\mathcal{O}$, typically an allocation of items to agents. Later in the paper, we sometimes also use "outcome" to refer to the output of the mechanism, namely the allocation together with the payments, when this is clear from the context.

We focus on the discrete type setting, i.e., $\mathcal{T}_i$ is a finite set containing m_i possible types, i.e., $|\mathcal{T}_i| = m_i$. Let $t_i^{(j)}$ denote the jth possible type of agent i, where $j \in [m_i]$. For all i and t_i, $v_i : (t_i, o) \to \mathbb{R}_{\geqslant 0}$ is a valuation that maps a type t_i and outcome o to a non-negative real number. A *direct revelation* mechanism $\mathcal{M} = (x, p)$ is a pair of *allocation rule* $x_i : \mathcal{T} \to \Delta(\mathcal{O})$, possibly randomized, and *expected payment rule* $p_i : \mathcal{T} \to \mathbb{R}_{\geqslant 0}$. We slightly abuse notation, and also use v_i to define the expected value of bidder i for mechanism $\mathcal{M}$, with the expectation taken with respect to the randomization used by the mechanism, that is

$$\forall i, \widehat{t} \in \mathcal{T}, v_i(t_i, x(\widehat{t})) = \mathbf{E}_{o \sim x(\widehat{t})}[v_i(t_i, o)], \tag{3.1}$$

for true type t_i and reported type profile $\widehat{t}$. When the reported types are $\widehat{t} = (\widehat{t}_1, \ldots, \widehat{t}_n)$, the output of mechanism $\mathcal{M}$ for agent i is denoted as $\mathcal{M}_i(\widehat{t}) = (x_i(\widehat{t}), p_i(\widehat{t}))$. We define the *utility* of agent i with true type t_i and a reported type $\widehat{t}_i$ given the reported type profile $\widehat{t}_{-i}$ of other agents as a quasilinear function,

$$u_i(t_i, \mathcal{M}(\widehat{t})) = v_i(t_i, x(\widehat{t})) - p_i(\widehat{t}). \tag{3.2}$$

For a multi-agent setting, it will be useful to also define the *interim rules*.

Definition 3.1 (Interim Rules of a Mechanism). *For a mechanism $\mathcal{M}$ with allocation rule x and payment rule p, the interim allocation rule X and payment rule P are defined as, $\forall i, t_i \in \mathcal{T}_i, X_i(t_i) = \mathbf{E}_{t_{-i} \in \mathcal{F}_{-i}}[x_i(t_i; t_{-i})], P_i(t_i) = \mathbf{E}_{t_{-i} \in \mathcal{F}_{-i}}[p_i(t_i; t_{-i})]$.*

In this paper, we assume we have oracle access to the interim quantities of mechanism $\mathcal{M}$.

Assumption 3.1 (Oracle Access to Interim Quantities). *For any mechanism $\mathcal{M}$, given any type profile $t = (t_1, \ldots, t_n)$, we receive the interim allocation rule $X_i(t_i)$ and payments $P_i(t_i)$, for all i, t_i.*

Moreover, we define the menu of a mechanism $\mathcal{M}$ in the following way.

Definition 3.2 (Menu). *For a mechanism $\mathcal{M}$, the menu of bidder i is the set $\{\mathcal{M}_i(t)\}_{t \in \mathcal{T}}$. The menu size of agent i is denoted as $|\mathcal{M}_i|$.*

In mechanism design, there is a focus on designing *incentive compatible* mechanisms, so that truthful reporting of types is an equlbrium. This is without loss of generality by the revelation principle. It has also been useful to work with approximate-IC mechanisms, and these have been studied in various papers, e.g. [DW12; CZ17; RW18; Cai+19; Düt+14; Düt+19b; FNP18; BSV19; Lah+18; FSS19], and gained a lot of attention.

In this paper, we focus on two definitions of approximate incentive compatibility, ε-BIC and ε-expected ex post incentive compatible (ε-EEIC) defined in the following. See Appendix B.4 for more different versions of approximately IC.

Definition 3.3 (ε-BIC Mechanism). *A mechanism $\mathcal{M}$ is called ε-BIC iff for all i, t_i,*

$$\mathbf{E}_{t_{-i} \sim \mathcal{F}_{-i}}[u_i(t_i, \mathcal{M}(t))] \geqslant \max_{\widehat{t}_i \in \mathcal{T}_i} \mathbf{E}_{t_{-i} \sim \mathcal{F}_{-i}}[u_i(t_i, \mathcal{M}(\widehat{t}_i; t_{-i}))] - \varepsilon$$

Definition 3.4 (ε-expected ex post IC (ε-EEIC) Mechanism [Düt+14]). *A mechanism $\mathcal{M}$ is ε-EEIC if and only if for all i, $\mathbf{E}_t\left[\max_{\widehat{t}_i \in \mathcal{T}_i}(u_i(t_i, \mathcal{M}(t)) - u_i(t_i, \mathcal{M}(\widehat{t}_i; t_{-i})))\right] \leqslant \epsilon$.*

A mechanism $\mathcal{M}$ is ε-EEIC iff no agent can gain more than ε ex post regret, in expectation over all type profiles $t \in \mathcal{T}$ (where ex post regret is the amount by which an agent's utility can be improved by misreporting to some $\widehat{t}_i$ given knowledge of t, instead of reporting its true type t_i). A 0-EEIC mechanism is strictly DSIC.[1] We can also consider an interim version of ε-EEIC, termed as ε-expected interim IC (ε-EIIC), defined as

$$\mathbf{E}_{t_i \sim \mathcal{F}_i}\left[\max_{t_i' \in \mathcal{T}_i} \mathbf{E}_{t_{-i} \sim \mathcal{F}_{-i}}\left[u_i(t_i, \mathcal{M}(t_i; t_{-i}))\right]\right] \geqslant \mathbf{E}_{t_i \sim \mathcal{F}_i}\left[\max_{t_i' \in \mathcal{T}_i} \mathbf{E}_{t_{-i} \sim \mathcal{F}_{-i}}\left[u_i(t_i, \mathcal{M}(t_i'; t_{-i}))\right]\right] - \varepsilon$$

All our results for ε-EEIC to BIC transformation hold for ε-EIIC mechanism. Indeed, we prove any ε-EEIC mechanism is ε-EIIC in Lemma B.1 in Appendix.

Another important property of the mechanism design is individual rationality (IR), where we define two standard versions of IR (ex-post/interim IR) in Appendix B.3. The transformation from ε-BIC/ε-EEIC to BIC mechanisms, proposed in this paper, preserves the individual rationality, regardless of interim or ex-post implementation. In other words, if the original ε-BIC/ε-EEIC mechanism is interim/ex-post IR, the mechanism achieved after transformation is still interim/ex-post IR, respectively.

For a mechanism $\mathcal{M}$ (even an approximate IC mechanism), let $R^{\mathcal{M}}(\mathcal{F})$ and $W^{\mathcal{M}}(\mathcal{F})$ represent the expected revenue and social welfare, respectively, of the mechanism when agents' types are sampled from $\mathcal{F}$ and they play $\mathcal{M}$ truthfully.

Definition 3.5 (Expected Social Welfare and Revenue). *For a (approximately IC) mechanism $\mathcal{M} = (x, p)$ with agents' types drawn from distribution $\mathcal{F}$, the expected revenue is defined as $R^{\mathcal{M}}(\mathcal{F}) = \mathbf{E}_{t \sim \mathcal{F}}[\sum_{i=1}^n p_i(t)]$, and the expected social welfare is defined as $W^{\mathcal{M}}(\mathcal{F}) = \mathbf{E}_{t \sim \mathcal{F}}[\sum_{i=1}^n v_i(t_i, x(t))]$.*

In this paper, we focus on welfare-preserving transform that provides negligible revenue loss, defined in the following,

[1] For discrete type settings, 0-EEIC is exactly DSIC. For the continuous type case, a 0-EEIC mechanism is strictly DSIC up to zero measure events.

Definition 3.6. *Given an ε-BIC mechanism $\mathcal{M}$ over type distribution $\mathcal{F}$, a welfare-preserving transform that provides negligible revenue loss outputs a mechanism $\mathcal{M}'$ such that, $W^{\mathcal{M}'}(\mathcal{F}) \geqslant W^{\mathcal{M}}(\mathcal{F})$ and $R^{\mathcal{M}'}(\mathcal{F}) \geqslant R^{\mathcal{M}}(\mathcal{F}) - r(\varepsilon)$, where $r(\varepsilon) \to 0$ if $\varepsilon \to 0$.*

Previous ε-BIC to BIC transformations

There are existing algorithms for transforming any ε-BIC mechanism to an exactly BIC mechanism with only negligible revenue loss [DW12; CZ17; RW18; Cai+19]. The central tools and reductions in these papers build upon the method of *replica-surrogate matching* [HL10; HKM11; BH11]. Here we briefly introduce replica-surrogate matching and its application to an ε-BIC to BIC transformation.

Replica-surrogate matching. For each agent i, construct a bipartite graph $G_i = (\mathcal{R}_i \cup \mathcal{S}_i, E)$. The vertices in $\mathcal{R}_i$ are called *replicas*, which are types sampled i.i.d. from the type distribution of agent i, $\mathcal{F}_i$. The nodes in $\mathcal{S}_i$ are called *surrogates*, and also sampled from $\mathcal{F}_i$. In particular, the true type t_i is added in $\mathcal{R}_i$. There is an edge between each replica and each surrogate. The weight of the edge between a replica $r_i^{(j)}$ and a surrogate $s_i^{(k)}$ is induced by the mechanism, and defined as

$$w_i(r_i^{(j)}, s^{(k)}) = E_{t_{-i} \in \mathcal{F}_{-i}} \left[v_i(r_i^{(j)}, x(s_i^{(k)}, t_{-i})) \right] - (1-\eta) \cdot \mathbf{E}_{t_{-i} \in \mathcal{F}_{-i}} \left[p_i(s_i^{(k)}, t_{-i}) \right]. \tag{3.3}$$

The replica-surrogate matching computes the maximum weight matching in G_i.

ε-BIC to BIC transformation by Replica-Surrogate Matching [DW12]. We briefly describe this transformation, deferring the details to Appendix B.1. Given a mechanism $\mathcal{M} = (x, p)$, this transformation constructs a bipartite graph between replicas (include the true type t_i) and surrogates, as described above. The approach then runs VCG matching to compute the maximum weighted matching for this bipartite graph, and charges each agent its VCG payment.For unmatched replicas in the VCG matching, the method randomly matches a surrogate. Let $\mathcal{M}' = (x, (1-\eta)p)$ be the modified mechanism. If the true type t_i is matched to a surrogate s_i, then agent i uses s_i to compete in $\mathcal{M}'$. The outcome of $\mathcal{M}'$ is $x(s)$, given matched surrogate profile s, and the payment of agent i (matched in VCG matching) is $(1-\eta)p_i(s)$ plus the VCG payment from the VCG matching, where η is the parameter in replica-surrogate matching . If t_i is not matched in the VCG matching, the agent gets nothing and pay zero.

The revenue loss of the replica-surrogate matching mechanism relative to the orginal mechanism $\mathcal{M}$ is at most $\eta\texttt{Rev}(\mathcal{M}) + \mathcal{O}\left(\frac{n\varepsilon}{\eta}\right)$, which has both a multiplicative and an additive loss term [DW12; CZ17; RW18; Cai+19]. Moreover, the transformation does not preserve welfare. Indeed, the replica-surrogate matching can achieve a welfare loss, which is bounded by $O\left(\frac{(1-\eta)\varepsilon}{\eta}\right)$.

The black-box reduction proposed in Bei and Huang [BH11] is a special case of this replica-surrogate matching method, where the weight of bipartite graph only depends on the valuations and not the prices ($\eta = 1$ in Eq. (3.3)), and the replicas and surrogates are both $\mathcal{T}_i$ (no sampling for replicas and surrogates). For this reason, the transformation method described there can preserve social welfare but can provide arbitrarily bad revenue (see Example 3.1).

Concurrently and independently, Cai et al. [Cai+19] propose a polynomial time algorithm to transform any ε-BIC mechanism to an exactly BIC mechanism, with only sample access to the type distribution and query access to the original ε-BIC mechanism. Their technique builds on the replica-surrogate matching mechanism [DW12], and [Dug+17][2], by extending replica-surrogate matching to handle negative weights in the graph. Their approach cannot preserve social welfare. In this work, we focus on the case that we have oracle access to the interim quantities of the original ε-BIC mechanism, following the setting proposed in e.g.,[HL10; HKM11; BH11; DW12; RW18]. How to generalize our approach to the setting that we only have sample access to the type distribution and get a polynomial-time transformation will be an interesting future work.

Our Contributions

We first state the main result of the paper, which provides a welfare-preserving transform from approximate BIC to exact BIC with negligible revenue loss.

Main Theorem 3.1 (Theorem 3.8)**.** *With $n \geqslant 1$ agents and independent private types, and an ε-BIC*

[2]Dughmi et al. [Dug+17] propose a general transformation from any black-box algorithm $\mathcal{A}$ to a BIC mechanism that only incurs negligible loss of welfare, with only polynomial number queries to $\mathcal{A}$, by using *Bernoulli factory* techniques.

and IR mechanism $\mathcal{M}$ that achieves W expected social welfare and R expected revenue, there exists a BIC and IR mechanism $\mathcal{M}'$ that achieves at least W social welfare and $R - \sum_{i=1}^{n} |\mathcal{T}_i|\varepsilon$ revenue. Given an oracle access to the interim quantities of $\mathcal{M}$, the running time of the transformation from $\mathcal{M}$ to $\mathcal{M}'$ is at most $\texttt{poly}(\sum_i |\mathcal{T}_i|)$.

The transformation works directly on the type graph of each agent, and it is this that allows us to maintain social welfare— indeed, we may even improve social welfare in our transformation. In contrast, the transformation from Bei and Huang [BH11] can incur unbounded revenue loss (see Example 3.1, it loses all revenue), and existing approaches with negligible revenue loss can lose social welfare (see Example 3.1).

Compared with Bei and Huang [BH11], the transform described here preserves welfare as well as providing negligible revenue loss. Compared to approx-BIC to exact-BIC transformations that have focused on revenue [DW12; CZ17; RW18], these existing transformations may incur welfare loss and incur both a multiplicative and an additive-loss in revenue, while our revenue loss is additive. Choosing $\eta = \sqrt{\varepsilon}$, the revenue loss of existing transforms is at most $\sqrt{\varepsilon}\texttt{Rev}(\mathcal{M}) + O(n\sqrt{\varepsilon})$. In the case that the original revenue, $\texttt{Rev}(\mathcal{M})$, is order-wise smaller than the number of types, i.e., $\texttt{Rev}(\mathcal{M}) = o(\sum_i |\mathcal{T}_i|)$, the existing transforms provide a better revenue bound (at some cost of welfare loss). But when the revenue is relatively larger than the number of types, i.e., $\texttt{Rev}(\mathcal{M}) = \Omega(\sum_i |\mathcal{T}_i|)$, our transformation can achieve strictly better revenue than these earlier approaches, as well as preserving welfare.

Before describing our techniques, we illustrate the comparision of these properties through a simple, single agent, two outcome example in Example 3.1. We show that even for the case that $\texttt{Rev}(M) = o(\sum_i |\mathcal{T}_i|)$, our transformation strictly outperforms existing transforms w.r.t revenue loss, in some cases.

Example 3.1. *Consider a single agent with m types, $\mathcal{T} = \{t^{(1)}, \cdots, t^{(m)}\}$, where the type distribution is uniform. Suppose there are two outcomes, the agent with type $t^{(j)} (j = 1, \ldots, m-1)$ values outcome 1 at 1 and values outcome 2 at 0. The agent with type $t^{(m)}$ values outcome 1 at $1 + \varepsilon$ and outcome 2 at $\sqrt{m}$. The mechanism $\mathcal{M}$ we consider is: if the agent reports type $t^{(j)}, j \in [m-1]$, $\mathcal{M}$ gives outcome 1 to the agent with a price of 1, and if the agent reports type $t^{(m)}$, $\mathcal{M}$ gives outcome 2 to the agent with a* *price of $\sqrt{m}$. $\mathcal{M}$ is ε-BIC, because the agent with type $t^{(m)}$ has a regret ε. The expected revenue achieved by $\mathcal{M}$ is $1 + \frac{\sqrt{m}-1}{m}$. In addition, $\mathcal{M}$ maximizes social welfare, $1 + \frac{\sqrt{m}-1}{m}$.*

Our transformation decreases the payment of type $t^{(m)}$ by ε for a loss of $\frac{\varepsilon}{m}$ revenue and preserves the social welfare.

The transformation by Bei and Huang [BH11] preserves the social welfare, however, the VCG payment (envy-free prices) is 0 for each type. Therefore, Bei and Huang [BH11]'s approach loses all revenue.

Moreover, the approaches by replica-surrogate matching (with negligible revenue loss) will lose at least $\frac{\varepsilon}{m} + \frac{\varepsilon}{\sqrt{m}-1}$ revenue, which is about $(\sqrt{m}+1)$ times larger than the revenue loss of our transformation. We argue this claim by a case analysis,

- *If $\eta \geqslant \frac{\varepsilon}{\sqrt{m}-1}$, the VCG matching is the identical matching and the VCG payment is 0 for each type. In total, the agent loses at least $\eta \cdot \frac{\sqrt{m}+m-1}{m} \geqslant \frac{\varepsilon}{m} + \frac{\varepsilon}{\sqrt{m}-1}$ expected revenue.*
- *If $\eta < \frac{\varepsilon}{\sqrt{m}-1}$, the agent with type $t^{(m)}$ will be assigned outcome 1 ($t^{(m)}$ is matched to some $t^{(j)}, j \in [m-1]$, in VCG matching) and the VCG payment is η. Thus, type $t^{(m)}$ loses at least $\sqrt{m} - (1-\eta) - \eta = \sqrt{m} - 1$ revenue. For any type $t^{(j)}, j \in [m-1]$, if $t^{(j)}$ is matched in VCG matching, the VCG payment is 0, since it will be matched to another type $t^{(k)}, k \in [m-1]$. Each type $t^{(j)}, j \in [m-1]$ loses at least η revenue. Overall the agent loses at least $\frac{\sqrt{m}-1}{m}$ expected revenue. In addition, since the type $t^{(m)}$ is assigned outcome 1, we lose at least $\frac{\sqrt{m}-1-\varepsilon}{m}$ expected social welfare.*

In any case, there is a chance that the type is not matched, then it reduces the social welfare strictly.

We also work with the approximate IC concept of ε-expected ex-post IC (ε-EEIC). This is motivated by work on the use of machine learning to achieve approximately IC mechanisms for multi-dimensional settings. EEIC is a smoother metric, and can be minimized through standard machine learning pipeline, such as SVM [Düt+14] and deep learning with an SGD solver [FNP18; Düt+19b]. In particular, ε-EEIC has been leveraged within the RegretNet framework [Düt+19b; FNP18]. A concern with the ε-EEIC metric, relative to ε-BIC, is that it differs in only guaranteeing at most ε gain *in expectation over type profiles*, with no guarantee for any particular type (in general, it is incomparable in strength from ϵ-BIC because at the same time, ε-EEIC strengthens ε-BIC in working with ex post regret rather than interim regret).

Our second main result shows how to transform an approximate, ε-expected ex-post IC (ε-EEIC) mechanism to a BIC mechanism.

Main Theorem 3.2 (Informal Theorem 3.7 and Theorem 3.8)**.** *For multiple agents with independent uniform type distribution, our ε-BIC to BIC transformation can be applied for ε-EEIC mechanism and all results in Informal Main Theorem 3.1 hold here. For a non-uniform type distribution, we show an impossibility result for a ε-EEIC to BIC, welfare-preserving transformation with only negligible revenue loss, even for the single agent case.*

Moreover, we also argue that our revenue loss bounds are tight given the requirement to maintain social welfare. This holds for both ε-BIC mechanisms and ε-EEIC mechanisms for multiple agents with independent uniform type distribution, summarized in the following theorem.

Main Theorem 3.3 (Informal Theorem 3.4 and Theorem 3.9)**.** *There exists an ε-BIC/ε-EEIC and IR mechanism for $n \geqslant 1$ agents with independent uniform type distribution, for which any welfare-preserving transformation must suffer at least $\Omega(\sum_i |\mathcal{T}_i|\varepsilon)$ revenue loss.*

Finally, we show the application of our transformation to Automated Mechanism Design in Section 3.5, where we apply our transformation to linear-programming based and machine learning based approaches to maximize a linear combination of expected revenue and social welfare as follows,

$$\mu_\lambda(\mathcal{M}, \mathcal{F}) = (1-\lambda)R^{\mathcal{M}}(\mathcal{F}) + \lambda W^{\mathcal{M}}(\mathcal{F}),$$

for some $\lambda \in [0,1]$ and type distribution $\mathcal{F}$. We summarize our results for the application of our transformation to LP-based and machine learning based approaches to AMD informally in the following theorem.

Main Theorem 3.4 (Informal Theorem 3.12 and Theorem 3.13)**.** *For n agents with independent type distribution $\times_{i=1}^n \mathcal{F}_i$ on $\mathcal{T} = \mathcal{T}_1 \times \cdots \times \mathcal{T}_n$ and an α-approximation LP algorithm* `ALG` *to output* *an ε-BIC (ε-EEIC) and IR mechanism $\mathcal{M}$ on $\mathcal{F}$ with $\mu_\lambda(\mathcal{M}, \mathcal{F}) \geqslant \alpha\texttt{OPT}$, there exists a BIC and IR mechanism $\mathcal{M}'$, s.t., $\mu_\lambda(\mathcal{M}', \mathcal{F}) \geqslant \alpha\texttt{OPT} - (1-\lambda)\sum_{i=1}^{n}|\mathcal{T}_i|\varepsilon$. Given oracle access to the interim quantities of $\mathcal{M}$, the running time to output the mechanism $\mathcal{M}'$ is at most $\texttt{poly}(\sum_{i=1}|\mathcal{T}_i|, rt_{\texttt{ALG}}(x))$, where $rt_{\texttt{ALG}}(\cdot)$ is the running time of ALG and x is the bit complexity of the input. Similar results hold for a machine learning based approach, in a PAC learning manner.*

Our Techniques

Instead of constructing a bipartite replica-surrogate graph, our transformation makes use of a directed, weighted type graph, one for each agent. For simplicity of exposition, we take the single agent with uniform type distribution case as an example. Given an ε-BIC mechanism, $\mathcal{M}$, we construct a graph $G = (\mathcal{T}, E)$, where each node represents a possible type of the agent and there is an edge from node $t^{(j)}$ to $t^{(k)}$ if the output of the mechanism for type $t^{(k)}$ is weakly preferred by the agent for true type $t^{(j)}$ in $\mathcal{M}$, i.e. $u(t^{(j)}, \mathcal{M}(t^{(k)})) \geqslant u(t^{(j)}, \mathcal{M}(t^{(j)}))$. The weight w_{jk} of edge $(t^{(j)}, t^{(k)})$ is defined as the *regret* of type $t^{(j)}$ by not misreporting $t^{(k)}$, i.e.,

$$w_{jk} = u(t^{(j)}, \mathcal{M}(t^{(k)})) - u(t^{(j)}, \mathcal{M}^{\varepsilon}(t^{(j)})). \tag{3.4}$$

Our transformation then iterates over the following two steps, constructing a transformed mechanism from the original mechanism. We briefly introduce the two steps here and defer to Figure 3.2 for detailed description.

Step 1. If there is a cycle $\mathcal{C}$ in the type graph with at least one positive-weight edge, then all types in this cycle weakly prefer their descendant in the cycle and one or more strictly prefers their descendant. In this case, we "*rotate*" the allocation and payment of types against the direction of the cycle, to let each type receive a weakly better outcome compared with its current outcome. We repeat Step 1 until all cycles in the type graph are removed.

Step 2. We pick a source node, if any, with a positive-weight outgoing edge (and thus regret for truthful reporting). We decrease the payment made by this source node, as well as decreasing the payment made by each one of its ancestors by the same amount, until we create a new edge in the type graph with weight zero, such that the modification to payments is about to increase regret for some type. If we create a cycle, we move to Step 1. Otherwise, we repeat Step 2 until there are no source nodes with positive-weight, outgoing edges.

The algorithm works on the type graph induced by the original, approximately IC mechanism, $\mathcal{M}$, and directly modifies the mechanism for each type, to make the mechanism IC. This allows the transformation can preserve welfare and provides negligible revenue loss. Step 2 has no effect on welfare, since it only changes (interim) payment for each type. Step 1 is designed to remove cycles created in Step 2 so that we can run Step 2, while preserving welfare simultaneously. Both steps reduce the total weight of the type graph, which is equivalent to reduce the regret in the mechanism to make it IC. We illustrate how our transformation works in Fig. 3.1, in high level. For example, in Example 3.1, there is no cycle in the type graph. We only need to run Step 2, that is reduce the payment of type $t^{(m)}$ by ε to make the approximately IC mechanism IC.

For a single agent with non-uniform type distribution, to handle the unbalanced density probability of each type, we redefine the type graph, where the weight of the edge in type graph is weighted by the product of the probability of the two nodes that are incident to an edge. We propose a new Step 1 by introducing *fractional rotation*, such that for each cycle in the type graph, we rotate the allocation and payment with a fraction for any type $t^{(j)}$ in the cycle. By carefully choosing the fraction for each type in the cycle, we can argue that our transformation preserve welfare and provides negligible revenue loss.

For the multi-agent setting, we reduce it to the single-agent case. In particular, we build a type graph for each agent induced by the interim rules (see more details in Appendix B.4.6 for construction of the type graph). Suppose we have oracle access to the interim quantities (Assumption 3.1) of original mechanism, we can build the type graph of each agent i in $\texttt{poly}(|\mathcal{T}_i|)$ time.[3] We then apply our transformation for each type graph of agent i, induced by the interim rules. This is analogous to the spirit of ε-BIC to BIC transformation by replica-surrogate matching, as they also define the weights between replicas and surrogates by interim rules and they only need to run the replica-surrogate matching for the reported type of each agent. The existing approaches use the sampling technique in replica-surrogate

[3] If we only have oracle access to the ex-post quantities, we need at least $\texttt{poly}(\prod_{j\neq i} |\mathcal{T}_j|)$ time to build the type graph of agent i.

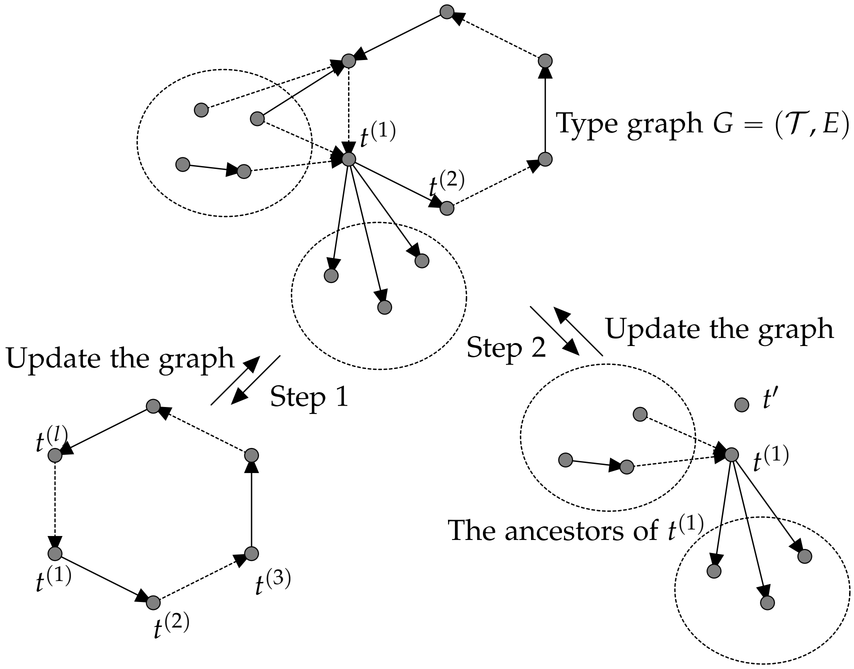


Figure 3.1: *Visualization of the transformation for a single agent with a uniform type distribution: we start from a type graph $G(\mathcal{T}, E)$, where each edge $(t^{(1)}, t^{(2)})$ represents the agent weakly prefers the allocation and payment of type $t^{(2)}$ rather than his true type $t^{(1)}$. The weight of each edge is denoted in Eq. (3.4). In the graph, we use solid lines to represent the positive-weight edges, and dashed lines to represent zero-weight edges. We first find a shortest cycle, and rotate the allocation and payment along the cycle and update the graph (Step 1). We keep doing Step 1 to remove all cycles. Then we pick a source node $t^{(1)}$, and decrease the payment of type $t^{(1)}$ and all the ancestors of $t^{(1)}$ until we reduce the weight of one outgoing edge from $t^{(1)}$ to zero or we create a new zero-weight edge from t' to $t^{(1)}$ or one of the ancestors of $t^{(1)}$ (Step 2).*

matching to make the distribution of reported type of each agent is equal to the distribution of true type. However, in our transformation, both Step 1 and Step 2 don't change the type distribution so that our transformation guarantees this property for free. Then we can apply our transformation for each type graph separately. The new challenge in our transformation is feasibility, i.e., establishing consistency of the agent-wise rotations to interim quantities. We show the transformation for each type graph guarantees the feasibility of the mechanism by appeal to Border's lemma [Bor91]. Our transformation can be directly applied to ε-EEIC mechanism, in the case that each agent has an independent uniform type distribution.

Further related work

Other work has needed to transform an infeasible, but IC mechanism into a feasible and IC mechanism. In particular, Narasimhan and Parkes [NP16] use a method from Hashimoto [Has18] to correct for feasibility violations in assignment mechanisms that result from statistical machine learning, while preserving strategy-proofness.

3.2 Warm-up: Single agent with Uniform Type Distribution

In this section, we consider the case of a single agent and a uniformly distributed type distribution $\mathcal{F}$, i.e. $\forall j \in [m], f(t^{(j)}) = \frac{1}{m}$. Even for this simple case, the proof is non-trivial. Moreover, the technique for this simple case can be extended to handle more intricate cases. The main result for a single agent and a uniform type distribution is Theorem 3.1, which makes use of a constructive proof to modify a ε-EEIC/ε-BIC mechanism to a BIC mechanism.

An interesting observation is that ε-EEIC is $m\varepsilon$-BIC for uniform type distribution, which indicates that transforming ε-EEIC may incur a worse revenue loss bound. However, Theorem 3.1 shows we can achieve the exactly same revenue loss bound for both IC definitions.

Theorem 3.1. *Consider a single agent, with m different types $\mathcal{T} = \{t^{(1)}, t^{(2)}, \cdots, t^{(m)}\}$, and a uniform type distribution $\mathcal{F}$. Given an ϵ-EEIC/ε-BIC and IR mechanism $\mathcal{M}$, which achieves W expected social welfare and R expected revenue, there exists an BIC and IR mechanism $\mathcal{M}'$ that achieves at least W expected social welfare and $R - m\varepsilon$ revenue. Given an oracle access to $\mathcal{M}$, the running time of the transformation from $\mathcal{M}$ to $\mathcal{M}'$ is at most* $\texttt{poly}(|\mathcal{T}|)$.

Proof Sketch. We construct a weighted directed graph $G = (\mathcal{T}, E)$ induced by mechanism $\mathcal{M}$, following the approach shown in Section 3.1. We apply the iterations of Step 1 and Step 2 (see Fig. 3.2), to reduce the total weight of edges in E to zero.

Firstly, we show the transformation maintains IR, since neither Step 1 nor Step 2 reduces utility. We then argue that the transformation in Fig. 3.2 will reduce the total weight of the graph to zero with no loss of social welfare, and incur at most $m\varepsilon$ revenue loss. To show this, we prove the following two auxiliary claims in Appendix B.4.1 and B.4.2, respectively.

Step 1 (Rotation step). Given the graph G induced by $\mathcal{M} = (x, p)$, find the shortest cycle $\mathcal{C}$ in G that contains at least one edge with positive weight. Without loss of generality, we represent $\mathcal{C} = \{t^{(1)}, t^{(2)}, \cdots, t^{(l)}\}$. Then rotate the allocation and payment rules for these nodes in cycle $\mathcal{C}$. Now we slightly abuse the notation of subscripts, s.t. $t^{(l+1)} = t^{(1)}$. Specifically, the allocation and payment rules for each $t^{(j)} \in \mathcal{C}$, $x'(t^{(j)}) = x(t^{(j+1)})$, $p'(t^{(j)}) = p(t^{(j+1)})$. For other nodes, we keep the allocation and payment rules, i.e. $\forall j \notin [l], x'(t^{(j)}) = x(t^{(j)}), p'(t^{(j)}) = p(t^{(j)})$. Then we update the mechanism $\mathcal{M}$ by adopting allocation and payment rules x', p' to form a new mechanism $\mathcal{M}'$, and update the graph G (We still use G to represent the updated graph for notation simplicity). If there are no cycles in G that contain at least one positive-weight-edge, move to Step 2. Otherwise, we repeat Step 1.
Step 2 (Payment reducing step). Given the current updated graph G and mechanism $\mathcal{M}'$, pick up a source node t, i.e., a node with no incoming positive-weight edges. Let outgoing edges with positive weights associated with node t be a set of E_t, and let $\bar{\varepsilon}_t$ be the minimum non-negative regret of type t, i.e.

$$\bar{\varepsilon}_t = \min_{t^{(j)}:(t,t^{(j)})\in E_t} \left[u(t, \mathcal{M}'(t^{(j)})) - u(t, \mathcal{M}'(t)) \right] \tag{3.5}$$

Consider the following set of nodes $S_t \subseteq \mathcal{T}$, such that $S_t = \{t\} \cup \{t' \big| t' \in \mathcal{T} \text{ is the ancestor of } t\}$. The weight zero edge is also counted as a directed edge. Denote ε_t as

$$\varepsilon_t = \min_{t' \notin S_t, \bar{t} \in S_t} \left[u(t', \mathcal{M}'(t')) - u(t', \mathcal{M}'(\bar{t})) \right] \tag{3.6}$$

Then we decrease the expected payment of all $\bar{t} \in S_t$ by $\min\{\varepsilon_t, \bar{\varepsilon}_t\}$. This process will only create new edges with weight zero. If we create a new cycle with at least one edge with positive weight in E, we move to Step 1. Otherwise, we repeat Step 2.

Figure 3.2: *ε-BIC/ε-EEIC to BIC transformation for single agent with uniform type distribution*

Claim 3.2. *Each Step 1 achieves the same revenue and incurs no loss of social welfare, and reduces the total weight of the graph by at least the weights of cycle $\mathcal{C}$.*

Claim 3.3. *Each Step 2 can only create new edges with zero weight, and does not decrease social welfare. Each Step 2 will reduce the weight of each positive-weight, outgoing edge associated with t by* $\min\{\varepsilon_t, \bar{\varepsilon}_t\}$, *where $\bar{\varepsilon}_t$ and ε_t are defined in Eq. (3.5) and Eq. (3.6) respectively.*

Given the above two claims, we argue our transformation incurs no loss of social welfare. The transformation only loses revenue at Step 2, for each source node t, we decrease at most $m \min\{\varepsilon_t, \bar{\varepsilon}_t\}$ payments over all the types[4]. In this transformation, after each Step 1 or Step 2, the

[4]Actually, we can get a slightly tighter bound. Since no cycle exists in the type graph after Step 1, there is at least one node is not the ancestor of t. Therefore the revenue decrease is bounded by $(m-1)\min\{\varepsilon_t, \bar{\varepsilon}_t\}$, actually.

weight of the outgoing edge of each node t is still bounded by $\max_j \left\{u(t, \mathcal{M}(t^{(j)})) - u(t, \mathcal{M}(t))\right\}$. This is because Step 1 does not create new outcome (allocation and payment) and Step 2 will not increase the weight of each edge. Therefore, in Step 2, we decrease payments by at most $m \max_j \left\{u(t, \mathcal{M}(t^{(j)})) - u(t, \mathcal{M}(t))\right\}$ in order to reduce the weights of all outgoing edges associated with t to zero. Therefore, the total revenue loss in expectation is

$$\sum_{t \in \mathcal{T}} \frac{1}{m} \cdot m \max_j \left(u(t, \mathcal{M}(t^{(j)})) - u(t, \mathcal{M}(t))\right) \leqslant m\varepsilon,$$

where the inequality is because of the definition of ε-BIC/ε-EEIC mechanism.

Running time. At each Step 1, we strictly reduce the weight of one edge with positive weight to 0 in the graph. In total, there are at most $|\mathcal{T}|^2$ edges. Thus, the total running time is $\texttt{ploy}(|\mathcal{T}|, \varepsilon)$. $\square$

3.2.1 Lower Bound on Revenue Loss

In our transformation shown in Figure 3.2, the revenue loss is bounded by $m\varepsilon$. The following theorem shows that this revenue loss bound is tight, up to a constant factor, while insisting on maintaining social welfare.

Theorem 3.4. *There exists an ε-BIC (ε-EEIC) and IR mechanism $\mathcal{M}$ for a single agent, for which any ε-BIC and IR to BIC and IR transformation (without loss of social welfare) must suffer at least $\Omega(m\varepsilon)$ revenue loss.*

Proof. Consider a single agent with m types, $\mathcal{T} = \{t^{(1)}, \cdots, t^{(m)}\}$ and $f(t^{(j)}) = 1/m, \forall j$. There are m possible outcomes. The agent with type $t^{(1)}$ values outcome 1 at ε and the other outcomes at 0. For any type $t^{(j)}, j \geqslant 2$, the agent with type $t^{(j)}$ values outcome $j-1$ at $j\varepsilon$, outcome j at $j\varepsilon$, and the other outcomes at 0. The original mechanism is: if the agent reports type $t^{(j)}$, gives the outcome j to the agent and charges $j\varepsilon$. There is a ε regret to an agent with type $t^{(j+1)}$ for not reporting type $t^{(j)}$, thus the mechanism is ε-BIC. Since this ε-BIC mechanism already maximizes social welfare, we cannot change the allocation in the transformation. Thus, we can only change the payment of each type to reduce the regret. Consider the sink node $t^{(1)}$, to reduce the regret of the agent with type $t^{(2)}$ for not reporting $t^{(1)}$, we can increase the payment of type $t^{(1)}$ or decrease the payment of type $t^{(2)}$. However, increasing the payment of type $t^{(1)}$ breaks IR, then we can only decrease the payment of $t^{(2)}$. To reduce the regret between $t^{(2)}$ to $t^{(1)}$, we need to decrease the payment of $t^{(2)}$ at least by ε. After this step, the regret of type $t^{(3)}$ for not reporting $t^{(2)}$ will be at least 2ε and $t^{(2)}$ will be the new sink node. Similarly, $t^{(3)}$ needs to decrease at least 2ε payment (if $t^{(2)}$ increase the payment, it will envy the output of $t^{(1)}$ again). So on and so forth, and in total, the revenue loss is at least $\frac{\varepsilon+2\varepsilon+\cdots+(m-1)\varepsilon}{m} = \frac{(m-1)\varepsilon}{2}$. $\square$

3.2.2 Tighter Bound of Revenue Loss for Settings with Finite Menus

In some settings, the total number of possible types of an agent may be very large, and yet the menu size can remain relatively small. In particular, suppose that a mechanism $\mathcal{M}$ has a small number of outputs, i.e., $|\mathcal{M}| = C$ and $C \ll m$, where m is the number of types and C is the menu size. Given this, we can provide a tighter bound on revenue loss for this setting. See Appendix B.4.3 for the complete proof.

Theorem 3.5. *Consider a single agent with m different types $\mathcal{T} = \{t^{(1)}, t^{(2)}, \cdots, t^{(m)}\}$, sampled from a uniform type distribution $\mathcal{F}$. Given an ϵ-BIC mechanim $\mathcal{M}$ with C different menus ($C \ll m$) that achieves S expected social welfare and R revenue, there exists an BIC mechanism $\mathcal{M}'$ that achieves at least S social welfare and $R - C\varepsilon$ revenue.*

3.3 Single Agent with General Type Distribution

In this section, we consider a setting with a single agent that has a non-uniform type distribution. A naive idea is that we can "divide" a type with a larger probability to several copies of the same type, each with equal probability, and then apply our proof of Theorem 3.1 to get a BIC mechanism. However, this would result in a weak bound on the revenue loss, since we would divide the m types into multiple, small pieces. This section is divided into two parts. First we show our transformation for an ε-BIC mechanism in this setting. Second, we show an impossibility result for an ε-EEIC mechanism, that is, without loss of welfare, no transformation can achieve negligible revenue loss.

3.3.1 ε-BIC to BIC Transformation

We propose a novel approach for a construction for the case of a single agent with a non-uniform type distribution. The proof is built upon Theorem 3.1, however, there is a technical difficulty to directly apply the same approach for this non-uniform type distribution case. Since each type has a different probability, we cannot rotate the allocation and payment in the same way as in Step 1 in the proof of Theorem 3.1.

We instead redefine the type graph $G = (\mathcal{T}, E)$, where the weight of the edge is now weighted by the product of the probability of the two nodes that are incident to an edge. We also modify the original rotation step shown in Fig. 3.2 in Appendix B.4.4: for each cycle in the type graph, we rotate the allocation and payment with the fraction of $\frac{f(t^{(k)})}{f(t^{(j)})}$ for any type $t^{(j)}$ in the cycle, where $f(t^{(k)})$ is the smallest type probability of the types in the cycle. This step is termed as "fractional rotation step." We summarize the results in Theorem 3.6 and show the proof in Appendix B.4.4.

Theorem 3.6. *Consider a single agent with m different types, $\mathcal{T} = \left\{t^{(1)}, t^{(2)}, \cdots, t^{(m)}\right\}$ drawn from a general type distribution $\mathcal{F}$. Given an ε-BIC and IR mechanim $\mathcal{M}$ that achieves W expected social welfare and R expected revenue, there exists a BIC and IR mechanism $\mathcal{M}'$ that achieves at least W social welfare and $R - m\varepsilon$ revenue.*

3.3.2 Impossibility Result for ε-EEIC Transformation

As mentioned above, given any ε-BIC for single agent with general type distribution, we can always transform to an exactly BIC mechanism, which incurs no loss of social welfare and negligible loss of revenue. However, the same claim doesn't hold for ε-EEIC, and Theorem 3.7 shows that, without loss of social welfare, no transformation can achieve negligible revenue loss. The complete proof of this result is provided in Appendix B.4.5.

Theorem 3.7. *There exists a single agent with a non-uniform type distribution, and an ε-EEIC and IR mechanism, for which any IC transformation (without loss of social welfare and IR) cannot achieve negligible revenue loss.*

3.4 Multiple Agents with Independent Private Types

First, we state our positive result for a setting with multiple agents and independent, private types (Theorem 3.8). We assume each agent i's type t_i is independent drawn from $\mathcal{F}_i$, i.e. $\mathcal{F}$ is a product distribution can be denoted as $\times_{i=1}^{n}\mathcal{F}_i$. Given any ε-BIC mechanism for n agents with independent private types (or any ε-EEIC mechanism for n agents with independent uniform type distribution), we show how to construct an exactly BIC mechanism with at least as much welfare and negligible revenue loss.

Theorem 3.8. *With n agents and independent private types, and an ε-BIC and IR mechanism $\mathcal{M}$ that achieves W expected social welfare and R expected revenue, there exists a BIC and IR mechanism $\mathcal{M}'$ that achieves at least W social welfare and $R - \sum_{i=1}^{n}|\mathcal{T}_i|\varepsilon$ revenue. The same result holds for an ε-EEIC mechanism with multiple agents, in the case that each agent has an independent uniform type distribution. Given an oracle access to the interim quantities of $\mathcal{M}$, the running time of the transformation from $\mathcal{M}$ to $\mathcal{M}'$ is at most* $\texttt{poly}(\sum_i |\mathcal{T}_i|)$.

Proof Sketch. We construct a separate type graph for each agent, based on the mechanism induced by the interim rules. We then prove the induced mechanism for each agent is still ε-BIC or ε-EEIC. Then we apply our transformation for each type graph separately. Finally, we argue that our transformation maintains feasibility by Border's lemma. The complete proof is shown in Appendix B.4.6. □

Lower bound of revenue loss. Similarly to single agent case, we can also prove the lower bound of revenue loss of any welfare-preserving transformation for multiple agents with independent private types. We summarize this result in Theorem 3.9 and show the proof in Appendix B.4.7.

Theorem 3.9. *There exists an ε-BIC/ε-EEIC and IR mechanism for $n \geqslant 1$ agents with independent uniform type distribution, for which any welfare-preserving transformation must suffer at least $\Omega(\sum_i |\mathcal{T}_i|\varepsilon)$ revenue loss.*

3.4.1 Impossibility Results

In our main positive result (Theorem 3.8), we assume each agent's type is independent and the target of transformation is BIC mechanism. In this section, we argue that these two assumptions are near-tight, in Theorem 3.10 and Theorem 3.11. See Appendix B.4.8 and Appendix B.4.9 for complete proofs.

Theorem 3.10 (Failure of interdependent type)**.** *There exists an ε-BIC mechanism $\mathcal{M}$ w.r.t an interdependent type distribution $\mathcal{F}$ (see Appendix B.3), such that no BIC mechanism over $\mathcal{F}$ can achieve negligible revenue loss compared with $\mathcal{M}$.*

Theorem 3.11 (Failure of DSIC target)**.** *There exists an ε-BIC mechanism $\mathcal{M}$ defined on a type distribution $\mathcal{F}$, such that any DSIC mechanism over $\mathcal{F}$ cannot achieve negligible revenue loss compared with $\mathcal{M}$.*

Theorem 3.10 provides a counterexample to show that if we allow for *interdependent types*, there is no way to construct a BIC mechanism without negligible revenue loss compared with the original ε-BIC mechanism, even if we ignore the social welfare loss. This leaves an open question that whether we can construct a counterexample for ε-BIC mechanism for correlated types. Note that Theorem 3.11 shows the impossibility result for the setting that we start from an ε-BIC mechanism. What if we start from an ε-EEIC mechanism $\mathcal{M}$ with independent uniform type distribution, can we get a DSIC mechanism with the similar properties to $\mathcal{M}$? We leave open the question as to whether it is possible to transform an ε-EEIC mechanism to a DSIC mechanism with zero loss of social welfare and negligible loss of revenue, for multiple agents with independent uniform type distribution.

3.5 Application to Automated Mechanism Design

In this section, we show how to apply our transformation to linear-programming based and machine-learning based approaches to automated mechanism design (AMD) [CS02], where the mechanism is automatically created for the setting and objective at hand. For this illustrative application, we take as the target of MD that of maximizing the following objective, for a given

$\lambda \in [0, 1]$ and type distribution $\mathcal{F}$,

$$\mu_\lambda(\mathcal{M}, \mathcal{F}) = (1-\lambda)R^{\mathcal{M}}(\mathcal{F}) + \lambda W^{\mathcal{M}}(\mathcal{F}). \tag{3.7}$$

Let $\texttt{OPT} = \max_{\mathcal{M}:\mathcal{M} \text{ is BIC and IR}} \mu_\lambda(\mathcal{M}, \mathcal{F})$ be the optimal objective achieved by a BIC and IR mechanism defined on $\mathcal{F}$. We consider two different AMD approaches, an LP-based approach and the RegretNet approach. We briefly introduce the above two approaches in the following.

LP-based AMD. In practice, the type space of each agent $\mathcal{T}_i$ may be very large (e.g., exponential in the number of items for multi-item auctions). To address this challenge, we can discretize $\mathcal{T}_i$ to a coarser space $\mathcal{T}_i^+$, $(|\mathcal{T}_i^+| \ll |\mathcal{T}_i|)$ and construct the coupled type distribution $\mathcal{F}_i^+$. (e.g., by rounding down to the nearest points in $\mathcal{T}_i^+$, that is, the mass of each point in T_i is associated with the nearest point in $\mathcal{T}_i^+$.) Then we can apply an LP-based AMD approach for type distribution $\mathcal{F}^+ = (\mathcal{F}_1^+, \cdots, \mathcal{F}_n^+)$. See Appendix B.5 for more details of LP-based AMD. Suppose, in particular, that we have an α-approximation LP algorithm to output an ε-BIC and IR mechanism $\mathcal{M}$ over $\mathcal{F}$,[5] such that $\mu_\lambda(\mathcal{M}, \mathcal{F}) \geqslant \alpha\texttt{OPT}$. Combining with our transformation for $\mathcal{M}$ on $\mathcal{F}$, we have the following theorem.

Theorem 3.12 (LP-based AMD)**.** *For n agents with independent type distribution $\times_{i=1}^n \mathcal{F}_i$, and an LP-based AMD approach for coarsened distribution $\mathcal{F}^+$ on coarsened type space $\mathcal{T}^+$ that gives an ε-BIC and IR mechanism $\mathcal{M}$ on $\mathcal{F}$, with $(1-\lambda)R + \lambda W \geqslant \alpha\texttt{OPT}$, for some $\lambda \in [0, 1]$, and some $\alpha \in (0, 1)$, then there exists a BIC and IR mechanism $\mathcal{M}'$ such that*

$$\mu_\lambda(\mathcal{M}', \mathcal{F}) \geqslant \alpha\texttt{OPT} - (1-\lambda)\sum_{i=1}^n |\mathcal{T}_i|\varepsilon.$$

Given oracle access to the interim quantities of $\mathcal{M}$ on $\mathcal{F}$ and an α-approximation LP solver with running time $rt_{LP}(x)$, where x is the bit complexity of the input, the running time to output the mechanism $\mathcal{M}'$ is at most $\texttt{poly}(\sum_i |\mathcal{T}_i|, rt_{LP}(\texttt{poly}(\sum_i |\mathcal{T}_i^+|, \frac{1}{\varepsilon}))$.

RegretNet AMD. RegretNet proposed in Chapter 1 is a generic data-driven, deep learning framework for multi-dimensional mechanism design. See Appendix B.5 for more details of

[5]Even though LP returns an mechanism defined only on $\mathcal{T}^+$, the mechanism $\mathcal{M}$ can be defined on $\mathcal{T}$, by coupling technique. For example, given any type profile $t \in \mathcal{T}$, there is a coupled $t^+ \in \mathcal{T}^+$ and the mechanism $\mathcal{M}$ takes t^+ as the input.

the application of RegretNet to this setting. Suppose that RegretNet is used in a setting with independent, uniform type distribution $\mathcal{F}$. To train RegretNet, we randomly draw S samples from $\mathcal{F}$ to form a training data $\mathcal{S}$ and train our model on $\mathcal{S}$. Let $\mathcal{H}$ be the functional space modeled by RegretNet. Suppose, in particular, that there is a PAC learning algorithm to train RegretNet which outputs an ε-EEIC mechanism $\mathcal{M} \in \mathcal{H}$ on $\mathcal{F}$, such that $\mu_\lambda(\mathcal{M}, \mathcal{F}) \geqslant \sup_{\widehat{\mathcal{M}} \in \mathcal{H}} \mu_\lambda(\widehat{\mathcal{M}}, \mathcal{F}) - \varepsilon$ holds with probability at least $1 - \delta$, by observing $S = S(\varepsilon, \delta)$ i.i.d samples from $\mathcal{F}$. Combining with our transformation for $\mathcal{M}$ on $\mathcal{F}$, we have the following theorem.

Theorem 3.13 (RegretNet AMD)**.** *For n agents with independent uniform type distribution $\times_{i=1}^n \mathcal{F}_i$ over $\mathcal{T} = (\mathcal{T}_1, \cdots, \mathcal{T}_n)$, and the use of RegretNet that generates an ε-EEIC and IR mechanism $\mathcal{M}$ on $\mathcal{F}$ with $\mu_\lambda(\mathcal{M}, \mathcal{F}) \geqslant \sup_{\widehat{\mathcal{M}} \in \mathcal{H}} \mu_\lambda(\widehat{\mathcal{M}}, \mathcal{F}) - \varepsilon$ holds with probability at least $1 - \delta$, for some $\lambda \in [0, 1]$, trained on $S = S(\varepsilon, \delta)$ i.i.d samples from $\mathcal{F}$, where $\mathcal{H}$ is the functional (mechanism) class modeled by RegretNet, then there exists a BIC and IR mechanism $\mathcal{M}'$, with probability at least $1 - \delta$, such that*

$$\mu_\lambda(\mathcal{M}', \mathcal{F}) \geqslant \sup_{\widehat{\mathcal{M}} \in \mathcal{H}} \mu_\lambda(\widehat{\mathcal{M}}, \mathcal{F}) - (1 - \lambda) \sum_{i=1}^n |\mathcal{T}_i| \varepsilon - \varepsilon,$$

Given oracle access to the interim quantities of $\mathcal{M}$ on $\mathcal{F}$ and an PAC learning algorithm for RegretNet with running time $rt_{RegretNet}(x)$, where x is the bit complexity of the input, the running time to output the mechanism $\mathcal{M}'$ is at most $\texttt{poly}(\sum_i |\mathcal{T}_i|, \varepsilon, rt_{RegretNet}(\texttt{poly}(S, \frac{1}{\varepsilon}))$.

3.6 Conclusion and Open Questions

In this paper, we have proposed a novel ε-BIC to BIC transformation that achieves negligible revenue loss and with no loss in social welfare. In particular, the transformation only incurs at most $\sum_i |\mathcal{T}_i| \varepsilon$ revenue loss, and no loss of social welfare. We also proved that this revenue loss bound is tight given the requirement that the transform should maintain social welfare. In addition, we investigated how to transform an ε-EEIC mechanism to an BIC mechanism, without loss of social welfare and with only negligible revenue loss.

We have demonstrated that the transformation can be applied to an ε-EEIC mechanism with multiple agents, in the case that each agent has a independent uniform type distribution.

For a non-uniform type distribution, we have established an impossibility result for ε-EEIC transforms, and even for the single-agent case.

This is the first work that contributes to approximately IC to IC transformation without loss of welfare. There remain some interesting open questions:

- Can we design a polynomial time algorithm for this ε-BIC to BIC transformation with negligible revenue loss and without loss of welfare, given only query access to the original mechanism and sample access to type distribution? (our polynomial time results assume oracle access to the interim quantities)

- Is it possible to transform an ε-EEIC mechanism to a DSIC mechanism, for multiple agents with an independent, uniform type distribution, without loss of welfare, and with only negligible revenue loss?

- If we only focus on the revenue perspective, is it possible to find a ε-EEIC to DSIC transformation, perhaps even in the non-uniform case?

Part III

Learning to Bid in Repeated Auctions

Chapter 4

Convergence Analysis of No-Regret Bidding Algorithms in Repeated Auctions

4.1 Introduction

The connection between Learning and Games has proven to be a very innovative, practical, and elegant area of research (see, e.g., [FL98; CL06; Nis+07]). Several fundamental connections have been made between Learning and Game Theory. A folklore result is that if all players play low-regret strategies in a repeated general-sum game, then the time-averaged history of the plays converges to a coarse correlated equilibrium (see, e.g., [BM07]). Similarly, low-swap-regret play leads to correlated equilibria.

In this work, we are interested in the setting of multi-agent learning in auction environments. Given the importance of auctions and bidding in the online advertising ecosystem, this is an important question from a practical point of view as well. In this setting, an auction takes input bids and determines the allocation of ad-slots and the prices for each advertiser. This is a highly repeated auction setting over a huge number of queries which arrive over time. Advertisers get feedback on the allocation and cost achieved by their chosen bidding strategy, and can respond by changing their bids, and often do so in an automated manner (we will assume throughout that advertisers are *profit-maximizing*, although there are potentially other goals as well). As each advertiser responds to the feedback, it changes the cost landscape for every other competing advertiser via the auction rules. As advertisers respond to the auction and to each other, the dynamics lead them to an equilibrium. This results in the following fundamental question: Given a fixed auction rule, do such bidding dynamics settle in an equilibrium and if so, what equilibrium do they choose.

Surprisingly, neither the Auction Theory literature from Economics nor the literature on Learning in Games provides a definitive answer. Consider the simplest setting: a repeated auction for a single item, where the allocation is either first-price or second-price. Auction Theory suggests that bidders converge to *canonical equilibria* (see, e.g., [Kri02]):

- For a second-price auction (or more generally a VCG auction), bidders will choose to be truthful (bid their true value every time) as this strategy weakly dominates every other strategy, i.e., no other strategy can yield more profit. This is a weakly dominating strategy Nash equilibirum (NE).

- For a first price auction in which each advertiser's value is picked from some commonly known distribution in an i.i.d. manner, each advertiser will underbid in a specific way to achieve a *Bayesian Nash Equilibrium* (BNE). For example, when there are two bidders and the value distribution is the uniform distribution on $[0, 1]$, then each advertiser will set its bid to be half of its true value.

While these canonical equilibria make intuitive sense, they are not the only equilibria in the respective games. For example, consider a single-item setting in which bidder 1 has a value of 1.0 and bidder 2 has a value of 0.5 for the item. While truthful bidding is an NE, any two values $b_1, b_2 \in [0.5, 1.0]^2$, with $b_1 > b_2$ also form an NE (in fact, an Envy-Free NE). Thus, there are an infinite number of NEs, with very different revenues. Similarly, in the Bayesian setting where the values are drawn from say, a uniform distribution, there are many NEs as well. For example, one player could bid 1 and the other player always bids 0 regardless of their valuations. This issue of multiple equilibria is treated in the Economics literature via various notions of equilibrium selection but it is not clear if such selection occurs naturally in settings such as ours, especially via bidding dynamics.

To take the Learning in Games approach to answering the question, we have to fix the bidding dynamics. We assume bidders use no-regret (mean-based) online learning algorithms; these are natural and powerful strategies that we may expect advertisers to use. A lot of commonly used no-regret learning algorithms, e.g. multiplicative weights update (MWU), follow the perturbed leader (FTPL), EXP3, UCB, and ε-Greedy, are all special cases of mean-based no-regret learning algorithms.

Indeed, there has been considerable work recently which studies various questions in the online advertising setting under this assumption (see, e.g., [NST15]). Folklore results in Learning imply that under low-regret dynamics the time-average of the advertisers' bids will converge to a *coarse correlated equilibrium (CCE)* of the underlying game. Hartline, Syrgkanis, and Tardos [HST15] shows that no-regret learning in Bayesian games converges to Bayesian CCE in the time-average manner. However, there could be many CCEs in a game as well. Since every NE is a CCE as well, the above examples hold for the second-price auction. For the first price setting (even when the values are drawn uniformly at random) another CCE is for the two bidders to bid $(v_1 + v_2)/2 + \epsilon$ and $(v_1 + v_2)/2$ where v_1 and v_2 are the drawn values, and $v_1 > v_2$. Further, since any convex combination of these CCEs is a valid CCE (the set of CCE forms a polytope), there is an infinite number of CCEs in both first and second price auctions. So again we are left with the prediction dilemma: it is not clear which of these CCEs a low-regret algorithm will converge to. Some of the CCEs are clearly not good for the ecosystem, for revenue or efficiency. Further, as we elaborate below, even if there is a unique CCE in the games of interest, the convergence guarantee is only for the time-average rather than point-wise.

Questions

These examples directly motivate the questions we ask in this paper:

- If the values are drawn in a bayesian setting, and the bidders follow a low-regret learning algorithm in a repeated second-price auction, do they converge to the truthful equilibrium?
- Similarly, in a first price auction with i.i.d. values with bidders values drawn from a uniform

distribution $[0,1]$, do they converge to the Bayesian Nash equilibrium (with two bidders) of bidding half of true value?

- Do the bidding dynamics converge to such an equilibrium point-wise, i.e., in the last-iterate sense, or only in the time-average sense?
- When there are multiple slots, do the bidders converge to truthful equilibrium under VCG settings?

Given the current state of the literature, we see these as fundamental questions to ask. The only guarantees we have are those known for general games: Low-regret dynamics converge to some CCE and there is no guarantee for the last-iterate convergence. If it is the case that only the time average converges, then that means that bidders may be required to keep changing their bidding strategy at every time step (see the discussion on the non-point-wise-convergence results in [BP18] below), and would achieve very different rewards over time. This would not be a satisfactory situation in the practical setting.

4.1.1 Our Results

Our main result is that when each of the bidders use a mean-based learning rule (see Definition 4.1) then all bidders converge to truthful bidding in a second-price auction and a multi-position VCG auction and to the Bayes Nash Equilibrium in a first-price auction.

Informal Main Theorem. *Suppose n players whose value are drawn from a distribution over space $\{\frac{1}{H}, \frac{2}{H}, \ldots, 1\}$ bid according to a mean-based learning algorithm in either (i) a second-price auction; (ii) a first-price auction (for $n = 2$ and uniform value distribution); or (iii) a multi-position VCG auction. Then, after an initial exploration phase, each bidder bids the canonical Bayes Nash equilibrium with high probability.*

The formal statement of this theorem appears in Theorem 4.1, Theorem 4.4, and Theorem 4.5 for second-price auctions, first-price auctions, and multi-position VCG auctions, respectively. Moreover, we show each bidder converges to bid canonical Nash Equilibrium point-wise for any time, which is in sharp contrast with previous time-average convergence analysis.

Throughout this paper, we assume the learning algorithms that the bidders use may be oblivious to the auction format that is used by the seller.

We complement these results by simulating the above model with experiments. In particular, we show that these algorithms converge and produce truthful (or the canonical) equilibria. Furthermore, we show that the algorithm converges much quicker than the theory would predict. These results indicate that these low-regret algorithms are quite robust in the context of Auctions.

4.1.2 Our Techniques

Our proof techniques involve a few key observations. For second-price and VCG auctions, we want to show that the bidding converges to truthful bidding. Firstly, note that if the other bidders bid completely randomly then the truthful arms have a slight advantage in profit. However, the other bidders themselves bid according to their own instance of the low-regret algorithms, thus the environment that a given bidder sees is not necessarily random; hence we need more insight. Fix a particular bidder, say bidder 1. In the beginning of the learning algorithm, all bidders do bid randomly. The next observation is that if the other bidders *happened* to converge to truthful bidding, then again bidder 1 will see completely random bids, because the other bidders' values are picked randomly at each stage. Hence we can say that both in the beginning and also if other bidders happen to converge to (or for some reason were restricted to) truthful bidding, then bidder 2 will see an advantage in truthful bidding and converge to that. It remains to show that in the interim period, when all bidders are learning, exploring, and exploiting, the truthful strategy builds and retains an advantage.

In a first price auction, the proofs follow the same structure. However there are some technical difficulties that one must overcome. Initially, both bidders bid uniformly at random and it is not difficult to show that bidding according to the canonical NE gives an advantage. If the bidders happen to converge to the BNE then, of course, bidding according to the BNE is the optimal strategy for either player. It is not clear, however, that when the opposing bid is *not* uniform or bidding according to the BNE that an advantage is maintained. Our technical contribution here is to show that an advantage for bidding the BNE is maintained which allows both bidders to converge to the BNE.

Our results show that we can achieve high probability results we can show that the model will bid truthfully *for all time* (assuming a modest exploratory period in the begining). This requires a new partitioning argument which enables us to apply concentrations results for all times. This technique may be of independent interest.

4.1.3 Related Works

Our work lies in a wide area of the inter-disciplinary research between mechanism design and online learning algorithms, e.g., [Aue+95; CL06; BM07], and we only point out a few lines of research which are more closely related to our work.

In online advertising, the setting where bidders may be running no-regret algorithms rather than best response, has recently been investigated and has garnered a significant amount of interest. For example, Nekipelov, Syrgkanis, and Tardos [NST15] study how to infer advertisers' values under this assumption. However, Bailey and Piliouras [BP18] show, somewhat surprisingly, that even in very simple games (e.g., a zero-sum matching pennies game), the no-regret dynamics of MWU do *not* converge, and in fact the individual iterates tend to *diverge* from the CCE. On the other hand, recent results in [DP18; Mer+19] show that certain Optimistic variants of Gradient Descent and Mirror Descent converge in the last-iterate to the NE in certain zero-sum games. Our result can be seen as a contribution in this stream of work as well, in that we show that for the (non-zero sum) games arising from auctions that we study, mean-based learning algorithms converge in the last iterate, and, in fact, to the natural (Bayes) NE. On a related note, Papadimitriou and Piliouras [PP19] shows there is a conflict between the economic solution concepts and those predicted by Learning Dynamics. In that framework, one can consider this work as suggesting that perhaps there is no such conflict between economic theory and learning, in the context of games arising from auctions, as learning converges to the solutions predicted by auction theory.

Our work is also related with *Learning to bid* literature, e.g., [WPR16; FPS18; Bal+19], where these papers focus on designing a good learning algorithm for the bidders in repeated auctions. In addition, Braverman et al. [Bra+18] considers how to design a mechanism to maximize revenue against bidders who adopt mean-based learning algorithms. In contrast, the auctions are fixed in our setting and we are interested in understanding the bidder dynamics.

Last but not least, Feldman, Lucier, and Nisan [FLN16] characterize multiple equilibria (NE, CE, and CCE) in first price auctions, under the prior-free (non-Bayesian) setting, and study the revenue and efficiency properties of these equilibria. They show there are auctions in which a CCE can have as low as $1 - 2/e \simeq 0.26$ factor of the second highest value (although not lower), and there are auctions in which a CCE can have as low as 0.81 of the optimal efficiency (but not lower). However, our results show that even though there may be "Bad" CCEs, the natural dynamics do not reach them, and instead, converge to the classic canonical Nash equilibrium.

4.2 Model and Notations

We consider the setting that there is a single seller repeatedly selling one good to n bidders per round. At each time t, each bidder i's valuation $v_{i,t}$ is i.i.d. drawn from an unknown (CDF) distribution F_i and bidder i will submit a bid $b_{i,t}$ based on $v_{i,t}$ and historical information. In this paper, we assume the value and the bid of each bidder at any time are always in a $\frac{1}{H}$-evenly-discretized space $V = \{\frac{1}{H}, \frac{2}{H}, \cdots, 1\}$, i.e, $v_{i,t}, b_{i,t} \in V, \forall i, t$. Let $v_t = (v_{1,t}, \cdots, v_{n,t})$ be the valuation profile of n bidders at time t, $v_{-i,t}$ be the valuation profile of bidders other than i, and similarly for b_t and $b_{-i,t}$. Let F be the (CDF) distribution of v_t and f_i be the probability density function (PDF) of bidder i's value. Denote $m_{i,t} = \max_{j \neq i} b_{j,t}$ as the maximum bid of the bidders other than i and $z_{i,t} = \max_{j \neq i} v_{j,t}$ be the maximum value of the bidders other than i. We denote G_i as the (CDF) distribution of $z_{i,t}$ and g_i as the associated PDF. For theoretical purpose, we propose an assumption about G_i in the following,

Assumption 4.1 (Thickness Assumption of G_i)**.** *There exists a constant $\tau > 0$ (may depend on n), s.t., $g_i(v) \geqslant \tau, \forall i \in [n], v \in V$. Without loss of generality*[1]*, we assume $\tau \leqslant \frac{1}{H^{n-1}}$.*

We assume the each bidder runs a no-regret learning algorithm to decide her bid at each time. Specifically, in this paper, we are interested in a broad class of no-regret learning

[1] It is without loss of generality, since if $\tau > \frac{1}{H^{n-1}}$, we redefine $\tau := \min\left\{\tau, \frac{1}{H^{n-1}}\right\}$.

algorithm known as mean-based (contextual) learning algorithm [Bra+18]; these include the multiplicative weights update (MWU), Exp3, and ε-Greedy algorithms as special cases.

In this paper, we focus on the contextual version of mean-based learning algorithms, which can be used to model learning algorithms of the bidders in repeated auctions, defined in the following.

Definition 4.1 (Mean-based Contextual Learning Algorithm)**.** *Let $r_{a,t}(c)$ be the reward of action a at time t when the context is c and $\sigma_{a,t}(c) = \sum_{s=1}^{t} r_{a,s}(c)$. An algorithm for the contextual bandits problem is γ_t-mean-based if it is the case that whenever $\sigma_{a,t}(c) < \sigma_{b,t}(c) - \gamma_t t$, then the probability $p_{a,t}(c)$ that the algorithm plays action a on round $t+1$, given context c, is at most γ_t. We say an algorithm is mean-based if it is γ_t-mean-based for some γ_t such that $\gamma_t t$ is increasing and $\gamma_t \to 0$ as $t \to \infty$.* [2]

In the repeated auctions setting, the context information received by each bidder i at time t is the realization of the valuation $v_{i,t}$. The reward function $r^i_{b,t}$ for bidder i can be defined as

$$\forall v \in [0,1], r^i_{b,t}(v) := u_{i,t}((b, b_{-i,t}); v), \tag{4.1}$$

where $u_{i,t}((b, b_{-i,t}); v)$ is the utility of bidder i at time t when the bidder i bids b and the others bid $b_{-i,t}$, if bidder i values the good v.

4.2.1 Learning Algorithms of Mean-Based Bidders

In this paper, we focus on the setting where each bidder i runs a γ_t-means-based contextual learning algorithm to submit the bid[3]. In addition, we assume each bidder runs several pure exploration steps in the beginning to estimate the reward of each action (bid) for each context (value). We assume each bidder runs T_0 pure exploration steps: at each pure exploration step, each bidder i uniformly generates a bid from B at random, regardless of the realization of value $v_{i,t}$. To summarize, we describe the learning algorithm of mean-based bidders in Algorithm 2.

[2]The mean-based learning algorithms proposed by [Bra+18] set γ_t be a constant, which only depends on total number of time steps T. Here we extend it to be a time-dependent variable, which is used to show our *anytime* convergence results.

[3]Indeed, our analysis can be extended to the setting where each mean-based bidder has different γ_t parameters. We assume they share parameters γ_t, for notation simplicity.

Algorithm 2 Mean-based (Contextual) Learning Algorithm of Bidder i

Input: parameters γ_t, T_0.
for $t = 1, 2, \dots, T_0$ **do**
  Choose bid $b_{i,t}$ uniformly from V at random.
end for
for $t = T_0 + 1, T_0 + 2, \dots$ **do**
  Observes value v_t.
  Choose bid $b_{i,t}$ following a γ_t-mean-based learning algorithm.
end for

In the *learning to bid* literature, there are different feedback models: full information feedback [PP19], bandit feedback [WPR16; FPS18], or cross-learning feedback [Bal+19]. However, our results hold for any feedback model, as long as each bidder uses the general mean-based learning algorithm to bid, shown in Algorithm 2.

4.3 Second Price Auctions with Mean-based Bidders

In this section, we analyze the learning dynamics of mean-based bidders in (repeated) second price auctions. In second price auctions, the utility function of each bidder i at time t can be represented as,

$$u_{i,t}((b, b_{-i,t}); v) = (v - m_{i,t}) \cdot \mathbb{I}\{b \geqslant m_{i,t}\} \tag{4.2}$$

Since the bids of each bidder are in a discrete space, we break ties randomly throughout this paper. We first show the following main theorem in this section, which proves that the mean-based learners converge to truthful reporting point-wisely, in the repeated second price auctions.

Theorem 4.1. *Suppose assumption 4.1 holds and T_0 is large enough, such that $\exp\left(-\frac{\tau^2 T_0}{32n^2H^2}\right) \leqslant \frac{1}{2}$ and $\gamma_t \leqslant \frac{\tau}{8nH}, \forall t \geqslant T_0$. Then at time $t > T_0$, each γ_t-mean-based learner i will submit $b_t = v_{i,t}$ in repeated second price auctions with probability at least $p(t) = 1 - H\gamma_t - 4\exp\left(-\frac{\tau^2 T_0}{32n^2H^2}\right)$, for any fixed $v_{i,t}$. Note $p(t) \to 1 - 4\exp\left(-\frac{\tau^2 T_0}{32n^2H^2}\right)$ when $t \to \infty$.*

Our main results for second price auctions show an *anytime* convergence for each bidder: as long as T_0 is large enough, each bidder will bid truthfully at any time t, with high probability.

The main technical contribution in this paper is the proof for Theorem 4.1.

4.3.1 Proof of Theorem 4.1

In this section, we summarize the proof of Theorem 4.1. To show that, we propose the following lemmas, in which the complete proofs are deferred to Appendix C.1.

Firstly, we characterize that in the pure exploration phase, each bidder gains significantly greater utility when bidding truthfully.

Lemma 4.1. *For any fixed value v, any bid $b \neq v$ and any time $t \leqslant T_0$, we have for each bidder i,*

$$\mathbb{P}\left(u_{i,t}((v, b_{-i,t}); v) - u_{i,t}((b, b_{-i,t}); v) \geqslant \frac{1}{H}\right) \geqslant \frac{\tau}{n}$$

Then by a standard Chernoff bound, we can argue the accumulative utility advantage obtained by bidding truthfully will be large enough, for any time t in exploration phase.

Lemma 4.2. *For any fixed v, any bid $b \neq v$ and any time $t \leqslant T_0$, we have for each bidder i,*

$$\sum_{s \leqslant t} u_{i,s}((v, b_{-i,s}); v) - u_{i,s}((b, b_{-i,s}); v) \geqslant \frac{\tau t}{2nH}$$

holds with probability at least $1 - \exp\left(-\frac{\tau^2 t}{2n^2 H^2}\right)$.

Finally, we show that if the cumulative utility advantage of truthful bidding is large enough to satisfy requirements of the mean-based learning algorithms, then truthful bidding still gains significant greater utility for each bidder at time $t > T_0$.

Lemma 4.3. *For any $t > T_0$, suppose $\sum_{s \leqslant t} u_{i,s}((v, b_{-i,s}); v) - u_{i,s}((b, b_{-i,s}); v) \geqslant \gamma_t t$ holds for any fixed $v, b \neq v$ and each bidder i, then*

$$u_{i,t+1}((v, b_{-i,t+1}); v) - u_{i,t+1}((b, b_{-i,t+1}); v) \geqslant \frac{1}{H}$$

holds with probability at least $\frac{\tau}{2n}$, for any fixed value v, bid $b \neq v$ and each bidder i.

Given the above three auxiliary lemmas, we prove Theorem 4.1 in the following.

Proof of Theorem 4.1. One of the key techniques used in this paper is the partitioning of the time steps into buckets with a geometric partitioning scheme. In particular, we divide time steps

$t > T_0$ to several episodes as follows, $\Gamma_1 = [T_0+1, T_1], \Gamma_2 = [T_1+1, T_2], ...,$ such that $\forall k \geqslant 1$, $T_k = \left\lfloor \frac{\tau T_{k-1}}{4\gamma_{T_k} nH} \right\rfloor$. We always choose the smallest T_k to satisfy this condition.[4] The total time steps of each episode $|\Gamma_k| = T_k - T_{k-1}, \forall k \geqslant 1$. Then we show the following claim, which states that in each time bucket the expected utility doesn't deviate too much.

Claim 4.2. *Let event $\mathcal{E}_k$ be $\sum_{s \leqslant T_k} u_{i,s}((v, b_{-i,s}); v) - u_{i,s}((b, b_{-i,s}); v) \geqslant \frac{\tau T_k}{4nH}$ holds for all i, given any fixed $v, b \neq v$. Then the event $\mathcal{E}_k$ holds with probability at least $1 - \sum_{\ell=0}^{k} \exp\left(-\frac{|\Gamma_\ell|\tau^2}{32n^2H^2}\right)$.*

We prove the above claim by induction. If $k = 0$, the claim holds by Lemma 4.2. We assume the claim holds for k, then we argue the claim still holds for $k+1$. We consider any time $t \in \Gamma_{k+1}$, given event $\mathcal{E}_k$ holds, we have

$$\begin{aligned} &\textstyle\sum_{s \leqslant t} u_{i,s}((v, b_{-i,s}); v) - u_{i,s}((b, b_{-i,s}); v) \\ &\geqslant \textstyle\sum_{s \leqslant T_k} u_{i,s}((v, b_{-i,s}); v) - u_{i,s}((b, b_{-i,s}); v) \\ &\qquad \geqslant \tfrac{\tau T_k}{4nH} \geqslant \gamma_t t, \end{aligned} \tag{4.3}$$

where the first inequality is based on the fact that truth-telling is the dominant strategy in second price auctions, the second inequality holds because of the induction assumption and the last inequality hold because $\forall t \in \Gamma_{k+1}$,

$$\gamma_t t \leqslant \gamma_{T_{k+1}} T_{k+1} = \gamma_{T_{k+1}} \left\lfloor \frac{\tau T_k}{4\gamma_{T_{k+1}} nH} \right\rfloor \leqslant \frac{\tau T_k}{4nH}.$$

Then by Lemma 4.3, given $\mathcal{E}_k$ holds, for any $t \in \Gamma_{k+1}$ we have,

$$\mathbb{P}\left(u_{i,t}((v, b_{-i,t}); v) - u_{i,t}((b, b_{-i,t}); v) \geqslant \frac{1}{H} \middle| \mathcal{E}_k\right) \geqslant \frac{\tau}{2n}$$

Thus, $\mathbf{E}\left[u_{i,t}((v, b_{-i,t}); v) - u_{i,t}((b, b_{-i,t}); v) \middle| \mathcal{E}_k\right] \geqslant \frac{\tau}{2nH}$ for any $t \in \Gamma_{k+1}$. Letting $\Delta_s = u_{i,s}((v, b_{-i,s}); v) -$

[4] T_k always exists since $\gamma_t t \to \infty$ as $t \to \infty$.

$u_{i,s}((b, b_{-i,s}); v)$, by Azuma's inequality (for martingales), we have

$$\begin{aligned}
&\mathbb{P}\left(\sum_{s\in\Gamma_{k+1}} \Delta_s \leqslant \frac{\tau|\Gamma_{k+1}|}{4nH}\Big|\mathcal{E}_k\right)\\
&\leqslant \mathbb{P}\left(\sum_{s\in\Gamma_{k+1}} \Delta_s \leqslant \sum_{s\in\Gamma_{k+1}} \mathbf{E}\left[\Delta_s\Big|\mathcal{E}_k\right] - \frac{\tau|\Gamma_{k+1}|}{4nH}\Big|\mathcal{E}_k\right)\\
&\leqslant \exp\left(-\frac{|\Gamma_{k+1}|\tau^2}{32n^2H^2}\right)
\end{aligned}$$

Therefore, the event $\mathcal{E}_{k+1}$ holds with probability at least

$$\left(1 - e^{-\frac{|\Gamma_{k+1}|\tau^2}{32n^2H^2}}\right) \cdot \mathbb{P}(\mathcal{E}_k) \geqslant 1 - \sum_{\ell=0}^{k+1} \exp\left(-\frac{|\Gamma_\ell|\tau^2}{32n^2H^2}\right),$$

which completes the induction and verifies the correctness of Claim 4.2. Given Claim 4.2, we have the following argument,

For any time $t > T_0$, there exists $k(t)$, s.t., $t \in \Gamma_{k(t)}$, if the event $\mathcal{E}_{k(t)}$ happens, the bidder i will report truthfully with probability at least $1 - H\gamma_t$, by the definition of γ_t-mean-based learning algorithms and the same argument as Eq. (C.1). Therefore, at any time $t > T_0$, each bidder i will report truthfully with probability at least

$$1 - H\gamma_t - \sum_{\ell=0}^{k(t)} \exp\left(-\frac{|\Gamma_\ell|\tau^2}{32n^2H^2}\right)$$

Then we bound $\sum_{\ell=0}^{k(t)} \exp\left(-\frac{|\Gamma_\ell|\tau^2}{32n^2H^2}\right)$ through the following claim, where the proof is deferred to Appendix B.

Claim 4.3. *Given* $\gamma_t \leqslant \frac{\tau}{8nH}, \forall t > T_0$, $\sum_{\ell=0}^{k(t)} \exp\left(-\frac{|\Gamma_\ell|\tau^2}{32n^2H^2}\right) \leqslant 4\exp\left(-\frac{\tau^2 T_0}{32n^2H^2}\right)$, *when* T_0 *is large enough s.t.* $\exp\left(-\frac{\tau^2 T_0}{32n^2H^2}\right) \leqslant \frac{1}{2}$.

Combining the above claim, we complete the proof for Theorem 4.1. □

4.4 Generalizations to Other Auctions

In this section, we generalize our results further to first price auctions when each bidder's valuation is drawn uniformly from V, and multi-position auctions when we run the Vickrey-Clarke-Groves (VCG) mechanism. For both cases, we break ties randomly.

4.4.1 First Price Auctions

In first price auctions, the highest bidder wins and pays her bid. Therefore the utility function of each bidder i at time t can be defined as,

$$u_{i,t}((b, b_{-i,t}); v) = (v - b) \cdot \mathbb{I}\{b \geqslant m_{i,t}\} \tag{4.4}$$

It is well-known, first price auctions are not truthful and each bidder will underbid her value to manipulate the auctions. Bayesian Nash Equilibrium (BNE) bidding strategy is hard to characterize in general first price auctions. Only the BNE bidding strategy for i.i.d bidders in first price auctions, is fully understood, e.g. [Kri02].

In this paper, for simplicity, we focus on the setting that there are two i.i.d bidders with uniform value distribution over V. The BNE bidding strategy for each bidder is $b = \frac{v}{2}$ when the value is v [Kri02], if the value space $V = [0, 1]$. In this work, we assume $V = \left\{\frac{1}{H}, \frac{2}{H}, \cdots, 1\right\}$ and we break ties randomly. Under this case, we show each mean-based bidder will converge to play near-BNE bidding strategy point-wisely if the number of initial exploration steps T_0 is large enough in the following.

Theorem 4.4. *Suppose there are two bidders and each bidder's value is i.i.d drawn uniformly from V at random, H is a even positive number, and T_0 is large enough, such that $\gamma_t \leqslant \frac{1}{4H^3}, \forall t \geqslant T_0$. Then at time $t > T_0$, each γ_t-mean-based learner i in repeated first price auctions will bid b_t, s.t. $\frac{v}{2} \leqslant b_t < \frac{v}{2} + \frac{1}{H}$ with probability at least $1 - H\gamma_t - \exp\left(-\frac{(H-1)T_0}{32(4H^3+1)H^4}\right)\frac{\log t}{\log\left(\frac{4H^3+H}{4H^3+1}\right)}$, for any fixed $v_{i,t}$.*

The proof is significantly different than the proof of second price auctions. Firstly, random tie-breaking makes the analysis more difficult in first price auctions compared with the one in second price auctions. Secondly, $b = \lceil \frac{v}{2} \rceil$ [5] is not a dominant bidding strategy, we need a more

[5] $\lceil x \rceil$ means rounding x up to the nearest point in V.

complex time-splitting scheme to make induction procedure works in first price auctions. We defer the proof to Appendix C.1. We believe our proof for firs price auctions is general enough and can be extended to handle more than two bidders setting.

4.4.2 Multi-Position VCG Auctions

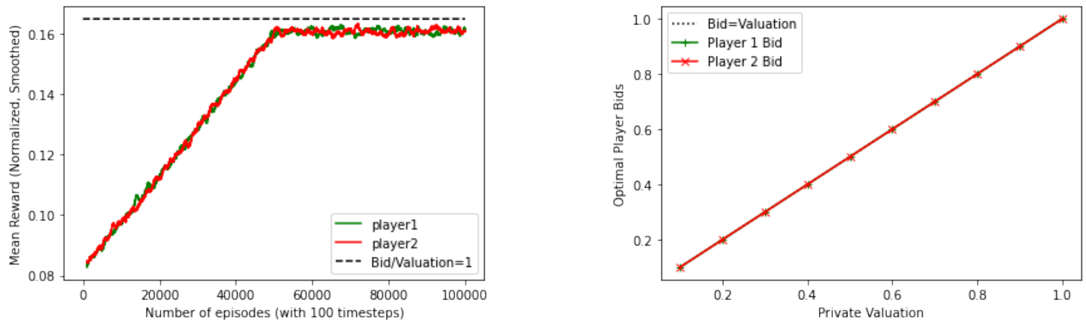
In multi-position auctions, there are k positions where $k < n$ and position multipliers $p_1 \geqslant \cdots \geqslant p_k \geqslant 0 = p_{k+1} = \cdots = p_n$ (position multipliers determine the relative values or click-through rates of the different positions). In particular, we will also say that $p_i - p_{i+1} \geqslant \rho$ for all $i \leqslant k$. In this paper, we run VCG mechanism for multi-position auctions. Without loss of generality, we assume that bidders are arranged in descending order of bids and there are no ties, i.e., $b_1 > b_2 > \cdots > b_n$. Since we run VCG auctions, bidder i gets the position i and the payment extracted from the bidder i is exactly $\sum_{j=i}^{m}(p_j - p_{j+1})b_j$. The utility that bidder i gets at being in position i is exactly, $\sum_{j=i}^{m}(p_j - p_{j+1}) \cdot (v_i - b_j)$. When there are ties, we break ties randomly.

Denote $z_i^{(k)}$ be the k-th largest value from the bidders other than bidder i, $G_i^{(k)}$ be the (CDF) distribution of $z_i^{(k)}$, and $g_i^{(k)}$ be the associated PDF. We propose the following thickness assumption for distribution $G_i^{(k)}$, for our theoretical purpose.

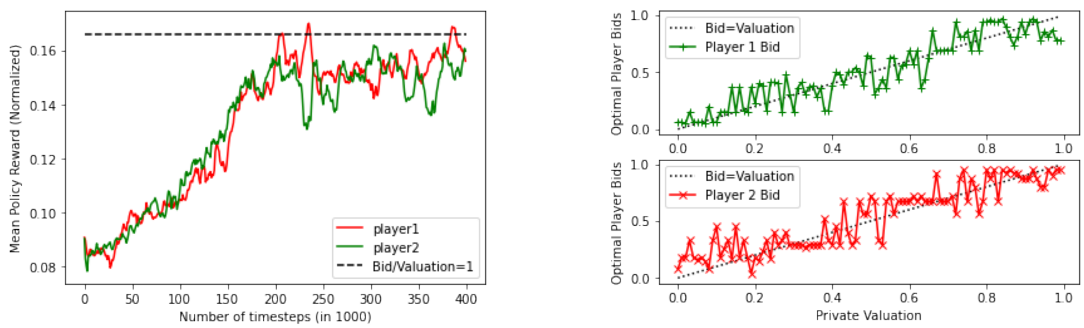
Assumption 4.2 (Thickness Assumption of $G_i^{(k)}$)**.** *There exists a constant $\tau > 0$ (may depend on n), s.t., $g_i^{(k)}(v) \geqslant \tau, \forall i \in [n], \forall v \in V$. Without loss of generality, we assume $\tau \leqslant \frac{1}{H^{n-1}}$.*

It is well-known that the VCG auction is truthful. Applying the same technique, we can show the truthful bid has an advantage compared with all the other bids in the exploration phase, as well as in the mean-based exploitation phase. Similarly, we show the following convergence results of mean-based bidders in multi-position VCG auctions.

Theorem 4.5. *Suppose assumption 4.2 holds and T_0 is large enough, such that $\exp\left(-\frac{\tau^2\rho^2 T_0}{32n^2H^2}\right) \leqslant \frac{1}{2}$ and $\gamma_t \leqslant \frac{\tau\rho}{8nH}, \forall t \geqslant T_0$. Then at time $t > T_0$, each γ_t-mean-based learner i will $b_t = v_{i,t}$ in repeated multi-position VCG auctions with probability at least $1 - H\gamma_t - 4\exp\left(-\frac{\tau^2\rho^2 T_0}{32n^2H^2}\right)$, for any fixed $v_{i,t}$.*



(a) *Training curve of mean reward of each bidder (left) and roll-out bidding strategy of each bidder (right) in the exploitation phase of contextual ε-Greedy algorithm in second price auctions.*



(b) *Training curve of mean reward of each bidder (left) and roll-out bidding strategy of each bidder (right) in the exploitation phase of Deep-Q Learning algorithm in second price auctions.*

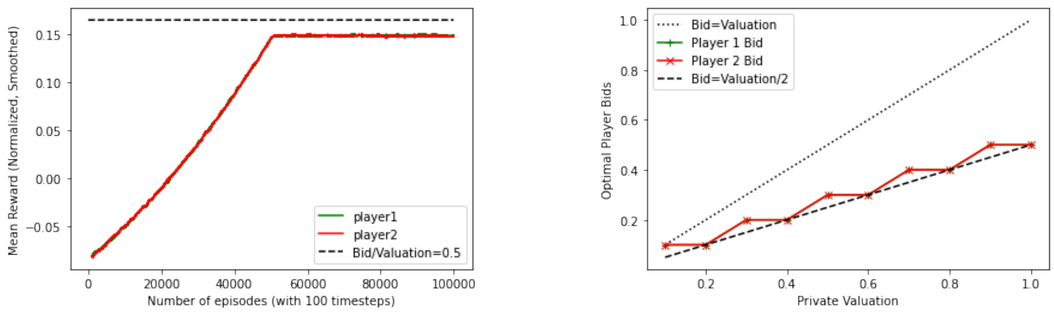
Figure 4.1: *Simulation results for second price auctions*

4.5 Experiments

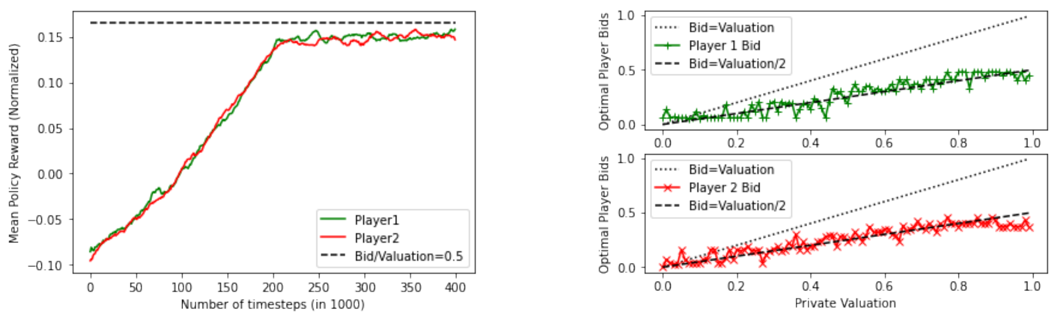
In this section, we describe the experiments using Contextual Mean-Based Algorithms and Deep-Q Learning agents participating in repeated first price and second price auctions. In these repeated auctions, the private valuations of both players are sampled independently from identical uniform distributions. The observation for each agent is defined by its private valuation and its reward by the auction outcome (with ties broken randomly).

In the first set of experiments, we study the convergence of two independent learning agents following an ε-Greedy policy in first and second price auction. We use the setting of $H = 10$ wherein both agents only observe their private valuation and the respective reward as an auction outcome.

In both cases, we observe the bidders converge to the BNE after several time steps. There is a slight gap between the (observed) mean reward and utility under BNE as the value



(a) *Training curve of mean reward of each bidder (left) and roll-out bidding strategy of each bidder (right) in the exploitation phase of contextual ε-Greedy algorithm in first price auctions.*



(b) *Training curve of mean reward of each bidder (left) and roll-out bidding strategy of each bidder (right) in the exploitation phase of Deep-Q Learning algorithm in first price auctions.*

Figure 4.2: *Simulation Results for first price auctions*

of ε (randomly exploration probability) has a floor of 0.05. We also observe that in the exploitation phase, the bidding converges completely to the BNE in the contextual bandit setting, which exactly matches our theory in Figures 4.1a and 4.2a. More experiments are shown in Appendix C.2.

Extensions to Deep-Q Learning

Contextual Mean Based Algorithms are a broad class of algorithms but can be very expensive to implement if we run a new instance for each possible value and the number of values are large. In line with modern machine learning, one way to mitigate this in practice is to augment it via Deep Q-Learning. To be more concrete, we model the learner by using a deep network with input as the private value and ask it to choose one of many bids. We model this as a reinforcement learning problem where the agents state is input into a deep neural network.

The agent's rewards are then observed over time with a chosen discount rate. The details of Deep Q-Learning model and the set of hyperparameters used to train the two Q models are outlined in Appendix C.2.

We use the setting of $H = 100$ and consider the observation of the agent as its private valuation. Again, we observe that both agents converge to BNE, shown in Figures 4.1b and 4.2b. We also study the model with a wider set of states including previously chosen bids and empirically observe the convergence of independent DQN agents to BNE for both auctions (discussed further in Appendix C.2).

Chapter 5

Learning to Bid Without Knowing your Value

5.1 Introduction

A standard assumption in the majority of the literature on auction theory and mechanism design is that participants that arrive in the market have a clear assessment of their valuation for the goods at sale. This assumption might seem acceptable in small markets with infrequent auction occurrences and amplitude of time for participants to do market research on the goods. However, it is an assumption that is severely violated in the context of the digital economy.

In settings like online advertisement auctions or eBay auctions, bidders participate very frequently in auctions that they have very little knowledge about the good at sale, e.g. the value produced by a user clicking on an ad. It is unreasonable, therefore, to believe that the participant has a clear picture of this value. However, the inability to pre-assess the value of the good before arriving to the market is alleviated by the fact that due to the large volume of auctions in the digital economy, participants can employ *learning-by-doing* approaches.

In this paper we address exactly the question of *how would you learn to bid approximately optimally in a repeated auction setting where you do not know your value for the good at sale and where that value could potentially be changing over time*. The setting of learning in auctions with an unknown value poses an interesting interplay between exploration and exploitation that is not standard in the online learning literature: in order for the bidder to get feedback on her value she has to bid high enough to win the good with higher probability and hence, receive some information about that underlying value. However, the latter requires paying a higher price. Thus, there is an inherent trade-off between value-learning and cost. The main point of this paper is to address the problem of learning how to bid in such unknown valuation settings with partial *win-only feedback*, so as to minimize regret with respect to the best fixed bid in hindsight.

On one extreme, one can treat the problem as a Multi-Armed Bandit (MAB) problem, where each possible bid that the bidder could submit (e.g. any multiple of a cent between 0 and some upper bound on her value) is treated as an arm. Then, standard MAB algorithms (see e.g. [BCB+12]) can achieve regret rates that scale *linearly* with the number of such discrete bids. The latter can be very slow and does not leverage the structure of utilities and the form of partial feedback that arises in online auction markets. Recently, the authors in [WPR16] addressed learning with such type of partial feedback in the context of repeated single-item second-price auctions. However, their approach does not address more complex auctions and is tailored to the second-price auction.

Our Contributions. Our work addresses learning with partial feedback in general mechanism design environments. Importantly, we allow for randomized auctions with probabilistic outcomes, encompassing the case of sponsored search auctions, where the outcome of the mechanism (i.e., getting a click) is inherently randomized.

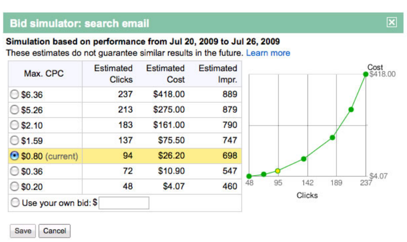
Our first main contribution is to introduce a novel online learning setting with partial feedback, which we denote *learning with outcome-based feedback* and which could be of independent interest. We show that our setting captures online learning in many repeated auction scenarios including, all types of single-item auctions, value-per-click sponsored search auctions, value-per-impression sponsored search auctions and multi-item auctions.

Our setting generalizes the setting of learning with feedback graphs [MS11; Alo+13], in a way that is crucial for applying it to the auction settings of interest. At a high level, the setting is defined as follows: The learner chooses an action $b \in B$ (e.g. a bid in an auction). The adversary chooses an *allocation function* x_t, that maps an action to a distribution over a set of potential outcomes O (e.g. the probability of getting a click) and a *reward function* r_t that maps an action-outcome pair to a reward (utility conditional on getting a click with a bid of b). Then, an outcome o_t is chosen based on distribution $x_t(b)$ and a reward $r_t(b, o_t)$ is observed. The learner also gets to observe the function x_t and the reward function $r_t(\cdot, o_t)$ for the realized outcome o_t (i.e. in our auction setting: she learns the probability of a click, the expected payment as a function of her bid and, *if she gets clicks*, her value).

Our second main contribution is an algorithm which we call WIN-EXP, which achieves regret $O\left(\sqrt{T|O|\log(|B|)}\right)$. The latter is inherently better than the generic multi-armed bandit regret of $O\left(\sqrt{T|B|}\right)$, since in most of our applications $|O|$ will be a small constant (e.g. $|O| = 2$ in sponsored search) and takes advantage of the particular feedback structure. Our algorithm is a variant of the EXP3 algorithm [Aue+02], with a carefully crafted unbiased estimate of the utility of each action, which has lower variance than the unbiased estimate used in the standard EXP3 algorithm. This result could also be of independent interest and applicable beyond learning in auction settings. Our approach is similar to the importance weighted sampling approach used in EXP3 so as to construct unbiased estimates of the utility of each possible action. Our main technical insight is how to incorporate the allocation function feedback that the bidder receives to construct unbiased estimates with small variance, leading to dependence only in the number of outcomes and not the number of actions. As we discuss in the related work, despite the several similarities, our setting has differences with existing partial feedback online learning settings, such as learning with experts [Aue+02], learning with feedback graphs [MS11; Alo+13] and contextual bandits [Aga+14].

This setting engulfs learning in many auctions of interest where bidders learn their value for a good only when they win the good and where the good which is allocated to the bidder is determined by some randomized allocation function. For instance, when applied to the case of single-item first-price, second-price or all-pay auctions, our setting corresponds to the case where the bidders observe their value for the item auctioned at each iteration only when they win the item. Moreover, after every iteration, they observe the critical bid they would have needed to submit to win (for instance, by observing the bids of others or the clearing price). The latter is typically the case in most government auctions or in settings like eBay.

Our flagship application is that of value-per-click sponsored search auctions. These are auctions were bidders repeatedly bid in an auction for a slot in a keyword impression on a search engine. The complexity of the sponsored search ecosystem and the large volume of repeated auctions has given rise to a plethora of automated bidding tools (see e.g. [Wor18]) and has made sponsored search an interesting arena for automated learning agents. Our framework captures the fact that in this setting the bidders observe their value for a click only when they get clicked. Moreover, it assumes that the bidders also observe the average probability of click and the average cost per click for any bid they could have submitted. The latter is exactly the type of feedback that the automated bidding tools can receive via the use of *bid simulators* offered by both major search engines [Goo18a; Goo18b; Mic18]. In Figure 5.1 we portray example interfaces from these tools, where we see that the bidders can observe exactly these allocation and payment curves assumed by our outcome-based-feedback formulation. Not using this information seems unreasonable and a waste of available information. Our work shows how one can utilize this partial feedback given by the auction systems to provide improved learning guarantees over what would have been achieved if one took a fully bandit approach. In the experimental section, we also show that our approach outperforms that of the bandit approach even if the allocation and payment curves provided by the system have some error that could stem from errors in the machine learning models used in the calculation of these curves by the search engines. Hence, even when these curves are not fully reliable our approach can offer improvements in the learning rate.



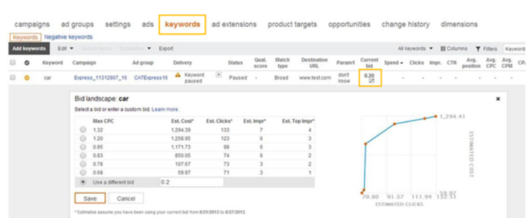


Figure 5.1: *Example interfaces of bid simulators of two major search engines, Google Adwords (left) and BingAds (right), that enables learning the allocation and the payment function. (sources [Sta14; Lan14])*

We also extend our results to cases where the space of actions is a continuum (e.g. all bids in an interval $[0, 1]$). We show that in many auction settings, under appropriate assumptions on the utility functions, a regret of $O\left(\sqrt{T\log(T)}\right)$ can be achieved by simply discretizing the action space to a sufficiently small uniform grid and running our WIN-EXP algorithm. This result encompasses the results of [WPR16] for second price auctions, learning in first-price and all-pay auctions, as well as learning in sponsored search with smoothness assumptions on the utility function. We also show how smoothness of the utility can easily arise due to the inherent randomness that exists in the mechanism run in sponsored search.

Finally, we provide two further extensions: *switching regret* and *feedback-graphs over outcomes*. The former adapts our algorithm to achieve good regret against a sequence of bids rather than a fixed bid. The latter has implications on faster convergence to approximate efficiency of the outcome (price of anarchy). Feedback graphs address the idea that in many cases the learner could be receiving information about other items other than the item he won (through correlations in the values for these items). This essentially corresponds to adding a feedback graph over outcomes and when outcome o_t is chosen, then the learner learns the reward function $r_t(\cdot, o)$ for all neighboring outcomes o in the feedback graph. We provide improved results that mainly depend on the dependence number of the graph rather than the number of possible outcomes.

Related Work. Our work lies on the intersection of two main areas: No regret learning in Game Theory and Mechanism Design and Contextual Bandits.

No regret learning in Game Theory and Mechanism Design. No regret learning has received a lot of attention in the Game Theory and Mechanism Design literature [CHN14]. Most of the existing literature, however, focuses on the problem from the side of the auctioneer, who tries to maximize revenue through repeated rounds without knowing a priori the valuations of the bidders [Ami+15; ARS14; Blu+04; BMM15; CBGM15; CR14; DRY15; KN14; MM14; OS11; MV17; Fel+16; KLM17]. These works are centered around different auction formats like the sponsored search ad auctions, the pricing of inventory and the single-item auctions. Our work is mostly related to Weed, Perchet, and Rigollet [WPR16], who adopt the point of view of the bidders in repeated second-price auctions and who also analyze the case when the true valuation of the item is revealed to the bidders only when they win the item. Their setting falls into the family of settings for which our novel and generic WIN-EXP algorithm produces good regret bounds and as a result, we are able to fully retrieve the regret that their algorithms yield, up to a tiny increase in the constants. Hence, we give an easier way to recover their results. Closely related to our work are the works of [DT13] and [BG17]. Dikkala and Tardos [DT13] analyzes a setting where bidders have to experiment in order to learn their valuations, and show that the seller can increase revenue by offering an initial credit to them, in order to give them incentives to experiment. [BG17] introduce a family of dynamic bidding strategies in repeated second-price auctions, where advertisers adjust their bids throughout the campaign. They analyze both regret minimization and market stability. There are two key differences from our setting; first, Balseiro and Gur consider the case where the goal of the bidders is the expediture rate in a way that guarantees that the available campaign budget will be spent in an optimal *pacing* way and second, because of their target being the expenditure rate at every timestep t, they assume that the bidders get information about the value of the slot being auctioned and based on this information they decide how to adjust their bid. Moreover, several works analyze the properties of auctions when bidders adopt a no-regret learning strategy [Blu+08; Car+15; Rou09]. None of these works, however, addresses the question of learning more efficiently in the unknown valuation model and either invokes generic MAB algorithms or develops tailored full information algorithms when the bidder knows his value. Another line of research takes a Bayesian approach to learn in repeated auctions and makes large market assumptions, analyzing learning to bid with an unknown value under a Mean Field Equilibrium condition [AJ13; IJS11; BBW15][1].

Learning with partial feedback. Our work is also related to the literature in *learning with partial feedback* [Aga+14; BCB+12]. To establish this connection we observe that the *policies* and the *actions* in contextual bandit terminology translate into *discrete bids* and *groups of bids for which we learn the rewards* in our work. The difference between these two is the fact that for each *action* in contextual bandits we get a single reward, whereas for our setting we observe a *group* of rewards; one for each action in the group. Moreover, the fact that we allow for randomized

[1]No-regret learning is complementary and orthogonal to the mean field approach, as it does not impose any stationarity assumption on the evolution of valuations of the bidder or the behavior of his opponents.

outcomes adds extra complication, non existent in contextual bandits. In addition, our work is closely related to the literature in *online learning with feedback graphs* [Alo+15; Alo+13; CHK16; MS11]. In fact, we propose a new setting in online learning, namely, *learning with outcome-based feedback*, which is a generalization of learning with feedback graphs and is essential when applied to a variety of auctions which include sponsored search, single-item second-price, single-item first-price and single-item all-pay auctions. Moreover, the fact that the learner only learns the probability of each outcome and not the actual realization of the randomness, is similar in nature to a feedback graph setting, but where the bidder does not observe the whole graph. Rather, he observes a distribution over feedback graphs and for each bid he learns with what probability each feedback graph would arise. For concreteness, consider the case of sponsored search and suppose for now that the bidder gets even more information than what we assume and also observes the bids of his opponents. He still does not observe whether he would get a click if he falls on the slot below but only the probability with which he would get a click in the slot below. If he could observe whether he would still get a click in the slot below, then we could in principle construct a feedback graph that would say that for all bids were the bidder gets a slot his reward is revealed, and for every bid that he does not get a click, his reward is not revealed. However, this is not the structure that we have and essentially this corresponds to the case where the feedback graph is not revealed, as analyzed in [CHK16] and for which no improvement over full bandit feedback is possible. However, we show that this impossibility is amended by the fact that the learner observes the probability of a click and hence for each possible bid, he observes the probability with which each feedback graph would have happened. This is enough for a low variance unbiased estimate.

5.2 Learning in Auctions without Knowing your Value

For simplicity of exposition, we start with a simple single-dimensional mechanism design setting, but our results extend to multi-dimensional (multi-item) mechanisms, as we will see in Section 5.4. Let n be the number of bidders. Each bidder has a value $v_i \in [0, 1]$ *per-unit of a good* and submits a bid $b_i \in B$, where B is a discrete set of bids (e.g. a uniform ϵ-grid of $[0, 1]$).

Given the bid profile of all bidders, the auction allocates a unit of the good to the bidders. The allocation rule for bidder i is given by $X_i(b_i, b_{-i})$. Moreover, the mechanism defines a per-unit payment function $P_i(b_i, b_{-i}) \in [0,1]$. The overall utility of the bidder is quasi-linear, i.e. $u_i(b_i, b_{-i}) = (v_i - P_i(b_i, b_{-i})) \cdot X_i(b_i, b_{-i})$.

Online Learning with Partial Feedback. The bidders participate in this mechanism repeatedly. At each iteration, each bidder has some value v_{it} and submits a bid b_{it}. The mechanism has some time-varying allocation function $X_{it}(\cdot,\cdot)$ and payment function $P_{it}(\cdot,\cdot)$. We assume that the bidder does *not* know his value v_{it}, nor the bids of his opponents b_{it}, nor the allocation and payment functions, *before* submitting a bid.

At the end of each iteration, he gets an item with probability $X_{it}(b_{it}, b_{-i,t})$ and observes his value v_{it} for the item only when he gets one (e.g. in sponsored search, the good allocated is the probability of getting clicked, and you only observe your value if you get clicked). Moreover, we assume that he gets to observe his allocation and payment functions for that iteration, i.e. he gets to observe two functions $x_{it}(\cdot) = X_{it}(\cdot, b_{-i,t})$ and $p_{it}(\cdot) = P_{it}(\cdot, b_{-i,t})$. Finally, he receives utility $(v_{it} - p_{it}(b_{it})) \cdot \mathbb{I}\{\text{item is allocated to him}\}$ or in other words expected utility $u_{it}(b_{it}) = (v_{it} - p_{it}(b_{it})) \cdot x_{it}(b_{it})$. Given that we focus on learning from the perspective of a single bidder we will drop the index i from all notation and instead write, $x_t(\cdot)$, $p_t(\cdot)$, $u_t(\cdot)$, v_t, etc. The goal of the bidder is to achieve small expected regret with respect to any fixed bid in hindsight: $R(T) = \sup_{b^* \in B} \mathbf{E}\left[\sum_{t=1}^{T} (u_t(b^*) - u_t(b_t))\right] \leqslant o(T)$.

5.3 Abstraction: Learning with Win-Only Feedback

We abstract a bit more the learner's problem, to a setting that could be of interest beyond auction settings.

Learning with Win-Only Feedback. Every day a learner picks an action b_t from a finite set B. The adversary chooses a reward function $r_t : B \to [-1,1]$ and an allocation function $x_t : B \to [0,1]$. The learner wins a reward $r_t(b)$ with probability $x_t(b)$. Let $u_t(b) = r_t(b)x_t(b)$ be the learner's expected utility from action b. After each iteration, if he won the reward then he learns the whole reward function $r_t(\cdot)$, while he *always* learns the allocation function $x_t(\cdot)$.

Can the learner achieve regret $O(\sqrt{T\log(|B|)})$ rather than bandit-feedback regret $O(\sqrt{T|B|})$?

In the case of the auction learning problem, the reward function $r_t(b)$ takes the parametric form $r_t(b) = v_t - p_t(b)$ and the learner needs to learn v_t and $p_t(\cdot)$ at the end of each iteration, when he wins the item. This is inline with the feedback structure we described in the previous section.

We consider the following adaptation of the EXP3 algorithm with unbiased estimates based on the information received. It is also notationally useful throughout the section to denote with A_t the event of *winning a reward at time t*. Then, we can write: $\mathbb{P}[A_t|b_t = b] = x_t(b)$ and $\mathbb{P}[A_t] = \sum_{b\in B}\pi_t(b)x_t(b)$, where with $\pi_t(\cdot)$ we denote the multinomial distribution from which bid b is drawn. With this notation we define our WIN-EXP algorithm in Algorithm 3. We note here that our generic family of the WIN-EXP algorithms can be parametrized by the step-size η, the estimate of the utility $\tilde{u}_t$ that the learner gets at each round and the feedback structure that he receives.

Algorithm 3 WIN-EXP algorithm for learning with win-only feedback

Let $\pi_1(b) = \frac{1}{|B|}$ for all $b \in B$ (i.e. the uniform distribution over bids), $\eta = \sqrt{\frac{2\log(|B|)}{5T}}$
for each iteration t **do**
  Draw a bid b_t from the multinomial distribution based on $\pi_t(\cdot)$
  Observe $x_t(\cdot)$ and if reward is won also observe $r_t(\cdot)$
  Compute estimate of utility:
  If reward is won $\tilde{u}_t(b) = \frac{(r_t(b)-1)\mathbb{P}[A_t|b_t=b]}{\mathbb{P}[A_t]}$; otherwise, $\tilde{u}_t(b) = -\frac{\mathbb{P}[\neg A_t|b_t=b]}{\mathbb{P}[\neg A_t]}$.
  Update $\pi_t(\cdot)$ as in Exponential Weights Update: $\forall b \in B : \pi_{t+1}(b) \propto \pi_t(b)\cdot\exp\{\eta\cdot\tilde{u}_t(b)\}$
end for

Bounding the Regret. We first bound the first and second moment of the unbiased estimates built at each iteration in the WIN-EXP algorithm.

Lemma 5.1. *At each iteration t, for any action $b \in B$, the random variable $\tilde{u}_t(b)$ is an unbiased estimate of the true expected utility $u_t(b)$, i.e.: $\forall b \in B : \mathbf{E}[\tilde{u}_t(b)] = u_t(b) - 1$ and has expected second moment bounded by: $\forall b \in B : \mathbf{E}\left[(\tilde{u}_t(b))^2\right] \leqslant \frac{4\mathbb{P}[A_t|b_t=b]}{\mathbb{P}[A_t]} + \frac{\mathbb{P}[\neg A_t|b_t=b]}{\mathbb{P}[\neg A_t]}$.*

Proof. Let A_t denote the event that the reward was won. We have:

$$\begin{aligned}
\mathbf{E}\left[\tilde{u}_t(b)\right] &= \mathbf{E}\left[\frac{(r_t(b)-1)\cdot\mathbb{P}[A_t|b_t=b]}{\mathbb{P}[A_t]}\mathbb{I}\{A_t\} - \frac{\mathbb{P}[\neg A_t|b_t=b]}{\mathbb{P}[\neg A_t]}\mathbb{I}\{\neg A_t\}\right]\\
&= (r_t(b)-1)\mathbb{P}[A_t|b_t=b] - \mathbb{P}[\neg A_t|b_t=b]\\
&= r_t(b)\mathbb{P}[A_t|b_t=b] - 1 = u_t(b) - 1
\end{aligned}$$

Similarly for the second moment:

$$\begin{aligned}
\mathbf{E}\left[\tilde{u}_t(b)^2\right] &= \mathbf{E}\left[\frac{(r_t(b)-1)^2\cdot\mathbb{P}[A_t|b_t=b]^2}{\mathbb{P}[A_t]^2}\mathbb{I}\{A_t\} + \frac{\mathbb{P}[\neg A_t|b_t=b]^2}{\mathbb{P}[\neg A_t]^2}\mathbb{I}\{\neg A_t\}\right]\\
&= \frac{(r_t(b)-1)^2\cdot\mathbb{P}[A_t|b_t=b]^2}{\mathbb{P}[A_t]} + \frac{\mathbb{P}[\neg A_t|b_t=b]^2}{\mathbb{P}[\neg A_t]} \leqslant \frac{4\mathbb{P}[A_t|b_t=b]}{\mathbb{P}[A_t]} + \frac{\mathbb{P}[\neg A_t|b_t=b]}{\mathbb{P}[\neg A_t]}
\end{aligned}$$

where the last inequality holds since $r_t(\cdot)\in[-1,1]$ and $x_t(\cdot)\in[0,1]$. □

We are now ready to prove our main theorem:

Theorem 5.1 (Regret of WIN-EXP)**.** *The regret of the* WIN-EXP *algorithm with the aforementioned unbiased estimates and step size* $\sqrt{\frac{2\log(|B|)}{5T}}$ *is:* $4\sqrt{T\log(|B|)}$*.*

Proof. Observe that regret with respect to utilities $u_t(\cdot)$ is equal to regret with respect to the translated utilities $u_t(\cdot)-1$. We use the fact that the exponential weights update with an unbiased estimate $\tilde{u}_t(\cdot)\leqslant 0$ of the true utilities, achieves expected regret of the form[2]:

$$R(T) \leqslant \frac{\eta}{2}\sum_{t=1}^{T}\sum_{b\in B}\pi_t(b)\cdot\mathbf{E}\left[(\tilde{u}_t(b))^2\right] + \frac{1}{\eta}\log(|B|)$$

Invoking the bound on the second moment by Lemma 5.1, we get:

$$R(T) \leqslant \frac{\eta}{2}\sum_{t=1}^{T}\sum_{b\in B}\pi_t(b)\cdot\left(\frac{4\mathbb{P}[A_t|b_t=b]}{\mathbb{P}[A_t]} + \frac{\mathbb{P}[\neg A_t|b_t=b]}{\mathbb{P}[\neg A_t]}\right) + \frac{1}{\eta}\log(|B|) \leqslant \frac{5}{2}\eta T + \frac{1}{\eta}\log(|B|)$$

Picking $\eta = \sqrt{\frac{2\log(|B|)}{5T}}$, we get the theorem. □

[2]A detailed proof of this claim can be found in Appendix D.7.

5.4 Beyond Binary Outcomes: Outcome-Based Feedback

In the win-only feedback framework there were two possible outcomes that could happen: either you win the reward or not. We now consider a more general problem, where there are more than two outcomes and you learn your reward function for the outcome you won. Moreover, the outcome that you won is also a probabilistic function of your action. The proofs for the results presented in this section can be found in Appendix D.2.

Learning with Outcome-Based Feedback. Every day a learner picks an action b_t from a finite set B. There is a set of payoff-relevant outcomes O. The adversary chooses a reward function $r_t : B \times O \to [-1, 1]$, which maps an action and an outcome to a reward and he also chooses an allocation function $x_t : B \to \Delta(O)$, which maps an action to a distribution over the outcomes. Let $x_t(b, o)$ be the probability of outcome o under action b. An outcome $o_t \in O$ is chosen based on distribution $x_t(b_t)$. The learner wins reward $r_t(b_t, o_t)$ and observes the whole outcome-specific reward function $r_t(\cdot, o_t)$. He *always* learns the allocation function $x_t(\cdot)$ after the iteration. Let $u_t(b) = \sum_{o \in O} r_t(b, o) \cdot x_t(b, o)$ be the expected utility from action b.

We consider the following adaptation of the EXP3 algorithm with unbiased estimates based on the information received. It is also notationally useful throughout the section to consider o_t as the random variable of the outcome chosen at time t. Then, we can write: $\mathbb{P}_t[o_t|b] = x_t(b, o_t)$ and $\mathbb{P}_t[o_t] = \sum_{b \in B} \pi_t(b) \mathbb{P}_t[o_t|b] = \sum_{b \in B} \pi_t(b) \cdot x_t(b, o_t)$. With this notation and based on the feedback structure, we define our WIN-EXP algorithm for learning with outcome-based feedback in Algorithm 4.

Algorithm 4 WIN-EXP algorithm for learning with outcome-based feedback

Let $\pi_1(b) = \frac{1}{|B|}$ for all $b \in B$ (i.e. the uniform distribution over bids), $\eta = \sqrt{\frac{\log(|B|)}{2T|O|}}$

for each iteration t **do**

Draw an action b_t from the multinomial distribution based on $\pi_t(\cdot)$

Observe $x_t(\cdot)$, observe chosen outcome o_t and associated reward function $r_t(\cdot, o_t)$

Compute estimate of utility:

$$\tilde{u}_t(b) = \frac{(r_t(b, o_t) - 1)\mathbb{P}_t[o_t|b]}{\mathbb{P}_t[o_t]} \tag{5.1}$$

Update $\pi_t(\cdot)$ based on the Exponential Weights Update:

$$\forall b \in B : \pi_{t+1}(b) \propto \pi_t(b) \cdot \exp\{\eta \cdot \tilde{u}_t(b)\} \tag{5.2}$$

end for

Theorem 5.2 (Regret of WIN-EXP with outcome-based feedback)**.** *The regret of Algorithm 4 with* $\tilde{u}_t(b) = \frac{(r_t(b,o_t)-1)\mathbb{P}_t[o_t|b]}{\mathbb{P}_t[o_t]}$ *and step size* $\sqrt{\frac{\log(|B|)}{2T|O|}}$ *is:* $2\sqrt{2T|O|\log(|B|)}$*.*

Applications to Learning in Auctions. We now present a series of applications of the main result of this section to several learning in auction settings, even beyond single-item or single-dimensional ones.

Example 5.1 (Second-price auction)**.** *auppose that the mechanism ran at each iteration is just the second price auction. Then, we know that the allocation function* $X_i(b_i, b_{-i})$ *is simply of the form:* $\mathbb{I}\{b_i \geqslant \max_{j \neq i} b_j\}$ *and the payment function is simply the second highest bid. In this case, observing the allocation and payment functions at the end of the auction simply boils down to observing the highest other bid. In fact, in this case we have a trivial setting where the bidder gets an allocation of either* 0 *or* 1 *and if we let* $B_t = \max_{j \neq i} b_{jt}$*, then the unbiased estimate of the utility takes the simpler form (assuming the bidder always loses in case of ties) of:* $\tilde{u}_t(b)\frac{(v_{it}-B_t-1)\mathbb{I}\{b>B_t\}}{\sum_{b'>B_t}\pi_t(b')}$ *if* $b_t > B_t$ *and* $\tilde{u}_t(b) - \frac{\mathbb{I}\{b \leqslant B_t\}}{\sum_{b' \leqslant B_t}\pi_t(b')}$ *in any other case. Our main theorem gives regret* $4\sqrt{T\log(|B|)}$*. We note that this theorem recovers exactly the results of Weed, Perchet, and Rigollet [WPR16], by simply using as* B *a uniform* $1/\Delta^o$

discretization of the bidding space, for an appropriately defined constant Δ^o (see Appendix D.2.1 for an exact comparison of the results).

Example 5.2 (Value-per-click auctions)**.** *This is a variant of the binary outcome case analyzed in Section 5.3, where $O = \{A, \neg A\}$, i.e. get clicked or not. Hence, $|O| = 2$, and $r_t(b, A) = v_t - p_t(b)$, while $r_t(b, \neg A) = 0$. Our main theorem gives regret $4\sqrt{T \log(|B|)}$.*

Example 5.3 (Unit-demand multi-item auctions)**.** *Consider the case of K items at an auction where the bidder has value v_k for only one item k. Given a bid b, the mechanism defines a probability distribution over the items that the bidder will be allocated and also defines a payment function, which depends on the bid of the bidder and the item allocated. When a bidder gets allocated an item k he gets to observe his value v_{kt} for that item. Thus, the set of outcomes is equal to $O = \{1, \ldots, K+1\}$, with outcome $K+1$ associated with not getting any item. The rewards are also of the form: $r_t(b, k) = v_{kt} - p_t(b, k)$ for some payment function $p_t(b, k)$ dependent on the auction format. Our main theorem then gives regret $2\sqrt{2(K+1)T \log(|B|)}$.*

5.4.1 Batch Rewards Per-Iteration and Sponsored Search Auctions

We now consider the case of sponsored search auctions, where the learner participates in multiple auctions per-iteration. At each of these auctions he has a chance to win and get feedback on his value. To this end, we abstract the *learning with win-only feedback* setting to a setting where multiple rewards are awarded per-iteration. The allocation function remains the same throughout the iteration but the reward functions can change.

Outcome-Based Feedback with Batch Rewards. Every iteration t is associated with a set of *reward contests* I_t. The learner picks an action b_t, which is used at *all* reward contests. For each $\tau \in I_t$ the adversary picks an outcome specific reward function $r_\tau : B \times O \to [-1, 1]$. Moreover, the adversary chooses an allocation function $x_t : B \to \Delta(O)$, which is not τ-dependent. At each τ, an outcome o_τ is chosen based on distribution $x_t(b_t)$ and independently. The learner receives reward $r_\tau(b_t, o_\tau)$ from that contest. The overall realized utility from that iteration is the average reward: $\frac{1}{|I_t|} \sum_{\tau \in I_t} r_\tau(b_t, o_\tau)$, while the expected utility from any bid b is: $u_t(b) = \frac{1}{|I_t|} \sum_{\tau \in I_t} \sum_{o \in O} r_\tau(b, o) \cdot x_t(b, o)$. We assume that at the end of each iteration the learner receives as feedback the average reward function conditional on each realized outcome, i.e. if we let $I_{to} = \{\tau \in I_t : o_\tau = o\}$, then the learner learns the function: $Q_t(b,o) = \frac{1}{|I_{to}|}\sum_{\tau \in I_{to}} r_\tau(b,o)$ (with the convention that $Q_t(b,o) = 0$ if $|I_{to}| = 0$) as well as the realized frequencies $f_t(o) = \frac{|I_{to}|}{|I_t|}$ for all outcomes o.

With this at hand we can define the *batch-analogue* of our unbiased estimates of the previous section. To avoid any confusion we define: $\mathbb{P}_t[o|b] = x_t(b,o)$ and $\mathbb{P}_t[o] = \sum_{b \in B} \pi_t(b)\mathbb{P}_t[o|b]$, to denote that these probabilities only depend on t and not on τ. The estimate of the utility will be:

$$\tilde{u}_t(b) = \sum_{o \in O} \frac{\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]} f_t(o)\left(Q_t(b,o) - 1\right) \tag{5.3}$$

We show the full algorithm with outcome-based batch-reward feedback in Algorithm 5.

Algorithm 5 WIN-EXP algorithm for learning with outcome-based batch-reward feedback

Let $\pi_1(b) = \frac{1}{|B|}$ for all $b \in B$ (i.e. the uniform distribution over bids), $\eta = \sqrt{\frac{\log(|B|)}{2T|O|}}$

for each iteration t **do**

Draw an action b_t from the multinomial distribution based on $\pi_t(\cdot)$

Observe $x_t(\cdot)$, chosen outcomes $o_\tau, \forall \tau \in I_t$, average reward function conditional on each realized outcome $Q_t(b,o)$ and the realized frequencies for each outcome $f_t(o) = \frac{|I_{to}|}{|I_t|}$.

Compute estimate of utility:

$$\tilde{u}_t(b) = \sum_{o \in O} \frac{\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]} f_t(o)\left(Q_t(b,o) - 1\right) \tag{5.4}$$

Update $\pi_t(\cdot)$ based on the Exponential Weights Update:

$$\forall b \in B : \pi_{t+1}(b) \propto \pi_t(b) \cdot \exp\{\eta \cdot \tilde{u}_t(b)\} \tag{5.5}$$

end for

Corollary 5.3. *The* WIN-EXP *algorithm with the latter unbiased utility estimates and step size* $\sqrt{\frac{\log(|B|)}{2T|O|}}$*, achieves regret in the outcome-based feedback with batch rewards setting at most:* $2\sqrt{2T|O|\log(|B|)}$.

It is also interesting to note that the same result holds if instead of using $f_t(o)$ in the expected utility (Equation (D.4)), we used its *mean value*, which is $x_t(o, b_t) = \mathbb{P}_t[o|b_t]$. This would not change any of the derivations above. The nice property of this alternative is that the learner does not need to learn the realized fraction of each outcome, but only the expected fraction of each outcome. This is already contained in the function $x_t(\cdot, \cdot)$, which we assumed was given to the learner at the end of each iteration. Thus, with these new estimates, the learner does not need to observe $f_t(o)$. In Appendix D.3 we also discuss the case where different periods can have different number of rewards and how to extend our estimate to that case. The batch rewards setting finds an interesting application in the case of learning in sponsored search, as we describe below.

Example 5.4 (Sponsored Search)**.** *In the case of sponsored search auctions, the latter boils down to learning the average value* $\widehat{v} = \frac{1}{\#clicks} \sum_{clicks} v_{click}$ *for the clicks that were generated, as well as the cost-per-click function* $p_t(b)$*, which is assumed to be constant throughout the period* t*. Given these quantities, the learner can compute:* $Q(b, A) = \widehat{v} - p_t(b)$ *and* $Q(b, \neg A) = 0$*. An advertiser can keep track of the traffic generated by a search engine ad and hence, can keep track of the number of clicks from the search engine and the value generated by each of these clicks (conversion). Thus, he can estimate* $\widehat{v}$*. Moreover, he can elicit the probability of click (aka click-through-rate or CTR) curves* $x_t(\cdot)$ *and the cost-per-click (CPC) curves* $p_t(\cdot)$ *over relatively small periods of time of about a few days. See for instance the Adwords bid simulator tools offered by Google, which exactly enable a bidder to elicit these curves [Goo18a; Goo18b; Mic18][3]. Thus, with these at hand we can apply our batch reward outcome based feedback algorithm and get regret that does not grow linearly with* $|B|$*, but only as* $4\sqrt{T \log(|B|)}$*. Our main assumption is that the expected CTR and CPC curves during this relatively small period of a few days remains approximately constant. The latter holds if the distribution of click-through-rates does not change within these days and if the bids of opponent bidders also do not significantly change. This is a reasonable assumption when feedback can be elicited relatively frequently, which is the case in practice.*

[3]One could argue that the CTRs that the bidder gets in this case are not accurate enough. Nevertheless, even if they have random perturbations, we show in our experimental results that for reasonable noise assumptions, WIN-EXP is more robust compared to EXP3.

5.5 Continuous Actions with Piecewise-Lipschitz Rewards

In this section, we extend our discussions to continuous action spaces; that is, we allow the action of each bidder to lie in a continuous action space $\mathcal{B}$ (e.g. a uniform interval in $[0,1]$). To assist us in our analysis, we are going to use the following discretization result by Kleinberg [Kle05] [4]. For what follows in this section, let $R(T,\mathcal{B}) = \sup_{b^*\in\mathcal{B}} \mathbf{E}\left[\sum_{t=1}^{T}\left(u_t(b^*) - u_t(b_t)\right)\right]$ be the regret of the bidder, after T rounds with respect to an action space $\mathcal{B}$. Moreover, for any pairs of action spaces B and $\mathcal{B}$ we let: $DE(B,\mathcal{B}) = \sup_{b\in\mathcal{B}} \sum_{t=1}^{T} u_t(b) - \sup_{b'\in B} \sum_{t=1}^{T} u_t(b')$, denote the discretization error incurred by optimizing over B instead of $\mathcal{B}$. The proofs of this section can be found in Appendix D.5.

Lemma 5.2. *([Kle05; KSU08]) Let $\mathcal{B}$ be a continuous action space and B a discretization of $\mathcal{B}$. Then:*

$$R(T,\mathcal{B}) \leqslant R(T,B) + DE(B,\mathcal{B})$$

Observe now that in the setting of Weed, Perchet, and Rigollet [WPR16] the discretization error was: $DE(B,\mathcal{B}) = 0$ if $\varepsilon < \Delta^o$, as we discussed in Section 5.4 and that was *the key* that allowed us to recover this result, without adding an extra εT in the regret of the bidder. To achieve that, we construct the following general class of utility functions:

Definition 5.1 (Δ^o-Piecewise Lipschitz Average Utilities)**.** *A learning setting with action space $\mathcal{B} = [0,1]^d$, is said to have Δ^o-Piecewise Lipschitz Cumulative Utilities if the average utility function $\frac{1}{T}\sum_{t=1}^{T} u_t(b)$ satisfies the following conditions: the bidding space $[0,1]^d$ is divided into d-dimensional cubes with edge length at least Δ^o and within each cube the utility is L-Lipschitz with respect to the ℓ_∞ norm. Moreover, for any boundary point there exists a sequence of non-boundary points whose limit cumulative utility is at least as large as the cumulative utility of the boundary point.*

Lemma 5.3 (Discretization Error for Piecewise Lipschitz)**.** *Let $\mathcal{B} = [0,1]^d$ be a continuous action space and B a uniform ε-grid of $[0,1]^d$, such that $\varepsilon < \Delta^o$ (i.e. B consists of all the points whose coordinates are multiples of a given ε). Assume that the average utility function is Δ^o-Piecewise L-Lipschitz. Then, the discretization error of B is bounded as: $DE(B,\mathcal{B}) \leqslant \varepsilon LT$.*

---

[4] In [Kle05] Kleinberg discusses the uniform discretization of continuum-armed bandits and in [KSU08] the authors extend the results for the case of Lipschitz-armed bandits.

If we know the Lipschitzness constant L mentioned above, the time horizon T and Δ^o, then our WIN-EXP algorithm for Outcome-Based Feedback with Batch Rewards yields regret as defined by the following theorem. In Appendix D.5, we also show how to deal with unknown parameters L, T and Δ^o by applying a standard doubling trick.

Theorem 5.4. *Let $\mathcal{B} = [0,1]^d$ be the action space as defined in our model and let B be a uniform ε-grid of $\mathcal{B}$. The* WIN-EXP *algorithm with unbiased estimates given by Equation D.4 on space B with step size $\sqrt{\frac{\log(|B|)}{2T|O|}}$ and $\varepsilon = \min\left\{\frac{1}{LT}, \Delta^o\right\}$ achieves expected regret at most $2\sqrt{2T|O|d\log\left(\max\left\{\frac{1}{\Delta^o}, LT\right\}\right)} + 1$ in the outcome-based feedback with batch rewards and Δ^o-Piecewise $L-$Lipschitz average utilities* [5].

Example 5.5 (First Price and All-Pay Auctions)**.** *Consider the case of learning in first price or all-pay auctions. In the former, the highest bidder wins and pays his bid, while in the latter the highest bidder wins and every player pays his bid whether he wins or loses. Let B_t be the highest other bid at time t. Then the average hindsight utility of the player in each auction is* [6]*:*

$$\frac{1}{T}\sum_{t=1}^{T} u_t(b) = \frac{1}{T}\sum_{t=1}^{T} v_t \cdot \mathbb{I}\{b > B_t\} - b \cdot \frac{1}{T}\sum_{t=1}^{T} \mathbb{I}\{b > B_t\} \qquad \text{(first price)}$$

$$\frac{1}{T}\sum_{t=1}^{T} u_t(b) = \frac{1}{T}\sum_{t=1}^{T} v_t \cdot \mathbb{I}\{b > B_t\} - b \qquad \text{(all-pay)}$$

Let Δ^o be the smallest *difference between the highest other bid at any two iterations t and t'* [7]*. Then observe that the average utilities in this setting are Δ^o-Piecewise 1-Lipschitz: Between any two highest other bids, the average allocation, $\frac{1}{T}\sum_{t=1}^{T} v_t \cdot \mathbb{I}\{b > B_t\}$, of the player remains constant and the only thing that changes is his payment which grows linearly. Hence, the derivative at any bid between any two such highest other bids is upper bounded by 1. Hence, by applying Theorem 5.4, our* WIN-EXP *algorithm with a uniform discretization on a ϵ-grid, for $\epsilon = \min\left\{\Delta^o, \frac{1}{T}\right\}$, achieves regret $4\sqrt{T\log\left(\max\left\{\frac{1}{\Delta^o}, T\right\}\right)} + 1$, where we used that $|O| = 2$ and $d = 1$ for any of these auctions.*

[5]Interestingly, the above regret bound can help to retrieve two familiar expressions for the regret. First, when $L = 0$ (i.e. when the function is constant within each cube), which is the case for the second price auction analyzed by [WPR16], $R(T) = 2\sqrt{2dT|O|\log\left(\frac{1}{\Delta^o}\right)} + 1$. Hence, we recover the bounds from the prior sections up to a tiny increase. Second, when $\Delta^o \to \infty$, then we have functions that are L-Lipschitz in the whole space $\mathcal{B}$ and the regret bound that we retrieve is: $R(T) = 2\sqrt{2dT|O|\log(LT)} + 1$, which is of the type achieved in continuous lipschitz bandit settings.

[6]For simplicity assume the player loses in case of ties, though we can handle arbitrary random tie-breaking rules.

[7]This is an analogue of the Δ^o used by [WPR16] in second price auctions.

5.5.1 Sponsored Search with Lipschitz Utilities

In this subsection, we extend our analysis of learning in the sponsored search auction model (Example 5.4) to the continuous bid space case, i.e., each bidder can submit a bid $b \in [0,1]$. As a reminder, the utility function is: $u_t(b) = x_t(b)(\widehat{v}_t - p_t(b))$, where $b \in [0,1]$, $\widehat{v}_t \in [0,1]$ is the average value for the clicks at iteration t, $x_t(\cdot)$ is the CTR curve and $p_t(\cdot)$ is the CPC curve. These curves are implicitly formed by running some form of a Generalized Second Price auction (GSP) at each iteration to determine the allocation and payment rules.

We show in this section that the form of the GSP ran in reality gives rise to Lipschitz utilities, under some minimal assumptions. Therefore, we can apply the results in Section 5.5 to get regret bounds even with respect to the continuous bid space $\mathcal{B} = [0,1]$ [8]. We begin by providing a brief description of the type of Generalized Second Price auction ran in practice.

Definition 5.2 (Weighted-GSP)**.** *Each bidder i is assigned a* quality score $s_i \in [0,1]$. *Bidders are ranked according to their score-weighted bid $s_i \cdot b_i$, typically called the* rank-score*. Every bidder whose rank-score does not pass a reserve r is discarded. Bidders are allocated slots in decreasing order of* rank-score*. Each bidder is charged per-click the lowest bid he could have submitted and maintained the same slot. Hence, if a bidder i is allocated a slot k and ρ_{k+1} is the rank-score of the bidder in slot $k+1$, then he is charged ρ_{k+1}/s_i per-click. We denote with $U_i(\mathbf{b}, \mathbf{s}, r)$, the utility of bidder i under a bid profile $\mathbf{b}$ and score profile $\mathbf{s}$.*

The quality scores are typically highly random and dependent on the features of the advertisement and the user that is currently viewing the page. Hence, a reasonable modeling assumption is that the scores s_i at each auction are drawn i.i.d. from some distribution with CDF F_i. We now show that if the CDF F_i is Lipschitz (i.e. admits a bounded density), then the utilities of the bidders are also Lipschitz.

Theorem 5.5 (Lipschitzness of the utility of Weighted GSP)**.** *Suppose that the score s_i of each bidder i in a weighted GSP is drawn independently from a distribution with an $L-$Lipschitz CDF F_i. Then, the expected utility $u_i(b_i, \mathbf{b}_{-i}, r) = \mathbf{E}_{\mathbf{s}}\left[U_i(b_i, \mathbf{b}_{-i}, \mathbf{s}, r)\right]$ is $\frac{2nL}{r}-$Lipschitz wrt b_i.*

[8] The aforementioned Lipschitzness is also reinforced by real world data sets from Microsoft's sponsored search auction system.

Thus, we see that when the quality scores in sponsored search are drawn from L-Lipschitz CDFs $F_i, \forall i \in n$ and the reserve is lower bounded by $\delta > 0$, then the utilities are $\frac{2nL}{\delta}$-Lipschitz and we can achieve good regret bounds by using the WIN-EXP algorithm with batch rewards, with action space B being a uniform ϵ-grid, $\epsilon = \frac{\delta}{2nLT}$ and unbiased estimates given by Equation (D.4) or Equation (5.3). In the case of sponsored search the second unbiased estimate takes the following simple form:

$$\tilde{u}_t(b) = \frac{x_t(b) \cdot x_t(b_t)}{\sum_{b' \in B} \pi_t(b') x_t(b')} \left(\widehat{v}_t - p_t(b) - 1\right) - \frac{(1 - x_t(b)) \cdot (1 - x_t(b_t))}{\sum_{b' \in B} \pi_t(b')(1 - x_t(b'))} \tag{5.6}$$

where $\widehat{v}_t$ is the average value from the clicks that happened during iteration t, $x_t(\cdot)$ is the CTR curve, b_t is the realized bid that the bidder submitted and $\pi_t(\cdot)$ is the distribution over discretized bids of the algorithm at that iteration. We can then apply Theorem 5.4 to get the following guarantee:

Corollary 5.6. *The* WIN-EXP *algorithm run on a uniform ϵ-grid with $\epsilon = \frac{\delta}{2nLT}$, step size $\sqrt{\frac{\log(1/\epsilon)}{4T}}$ and unbiased estimates given by Equation* (D.4) *or Equation* (5.3)*, when applied to the sponsored search auction setting with quality scores drawn independently from distributions with L-Lipschitz CDFs, achieves regret at most:* $4\sqrt{T \log\left(\frac{2nLT}{\delta}\right)} + 1$.

5.6 Further Extensions

In this section, we discuss two extensions of our setting, one is about switching regret and the implications for Price of Anarchy and the other is the extension to feedback graphs setting.

5.6.1 Switching Regret and Implications for Price of Anarchy

We show below that actually our results can be extended to capture the case where, instead of having just one optimal bid b^*, there is a sequence of $C \geqslant 1$ *switches* in the optimal bids. Using the results presented in [GLL12] and adapting them for our setting we get the following corollary (with proof in Appendix D.6).

Corollary 5.7. *Let $C \geqslant 0$ be the number of times that the optimal bid $b^* \in \mathcal{B}$ switches in a horizon of T rounds. Then, using Algorithm* 2 *in [GLL12] with $\mathcal{A} \equiv$* WIN-EXP *and any $\alpha \in (0, 1)$ we can achieve* *expected* switching regret *at most:* $O\left(\sqrt{(C+1)^2\left(2+\frac{1}{\alpha}\right)2d|O|T\log\left(\max\left\{LT,\frac{1}{\Delta^o}\right\}\right)}\right)$

This result has implications on the price of anarchy (PoA) of auctions. In the case of sponsored search where bidders' valuations are changing over time adversarially but non-adaptively, our result shows that if the valuation does not change more than C times, we can compete with any bid that is a function of the value of the bidder at each iteration, with regret rate given by the latter theorem. Therefore, by standard PoA arguments [LST16], this would imply convergence to an approximately efficient outcome at a faster rate than bandit regret rates.

5.6.2 Feedback Graphs over Outcomes

We now extend Section 5.5, by assuming that there is a directed feedback graph $G=(O,E)$ over the outcomes. When outcome o_t is chosen, the player observes not only the outcome specific reward function $r_t(\cdot,o_t)$, for that outcome, but also for any outcome o in the out-neighborhood of o_t in the feedback graph, which we denote with $N^{out}(o_t)$. Correspondingly, we denote with $N^{in}(o)$ the incoming neighborhood of o in G. Both neighborhoods include self-loops. Let $G_\epsilon=(O_\epsilon,E_\epsilon)$ be the sub-graph of G that contains only outcomes for which $\mathbb{P}_t[o_t]\geqslant\epsilon$ and subsequently, let N_ϵ^{in} and N_ϵ^{out} be the in and out neighborhoods of this sub-graph.

Based on this feedback graph we construct a WIN-EXP algorithm with step-size $\eta=\sqrt{\frac{\log(|B|)}{8T\alpha\ln\left(\frac{16|O|^2T}{\alpha}\right)}}$, utility estimate $\tilde{u}_t(b)=\mathbb{I}\{o_t\in O_\epsilon\}\sum_{o\in N_\epsilon^{out}(o_t)}\frac{(r_t(b,o)-1)\mathbb{P}_t[o|b]}{\sum_{o'\in N_\epsilon^{in}(o)}\mathbb{P}_t[o']}$ and feedback structure as described in the previous paragraph. With these changes we can show that the regret grows as a function of the *independence number of the feedback graph*, denoted with α, rather than the *number of outcomes*. The full Algorithm 6 can be found in Appendix D.1.

Theorem 5.8 (Regret of WIN-EXP-G). *The regret of the* WIN-EXP-G *algorithm with step size* $\eta=\sqrt{\frac{\log(|B|)}{8T\alpha\ln\left(\frac{16|O|^2T}{\alpha}\right)}}$ *is bounded by:* $R(T)\leqslant 2\sqrt{8\alpha T\log(|B|)\ln\left(\frac{16|O|^2T}{\alpha}\right)}+1.$

In the case of learning in auctions, the feedback graph over outcomes can encode the possibility that winning an item can help you uncover your value for other items. For instance, in a combinatorial auction for m items, the reader should think of each node in the feedback graph as a bundle of items. Then the graph encodes the fact that winning bundle o can teach you the value for all bundles $o'\in N^{out}(o)$. If the feedback graph has small dependence number then a much better regret is achieved than the dependence on $\sqrt{2^m}$, that would have been derived by our outcome-based feedback results of prior sections, if we treated each bundle of items separately as an outcome.

5.7 Experimental Results

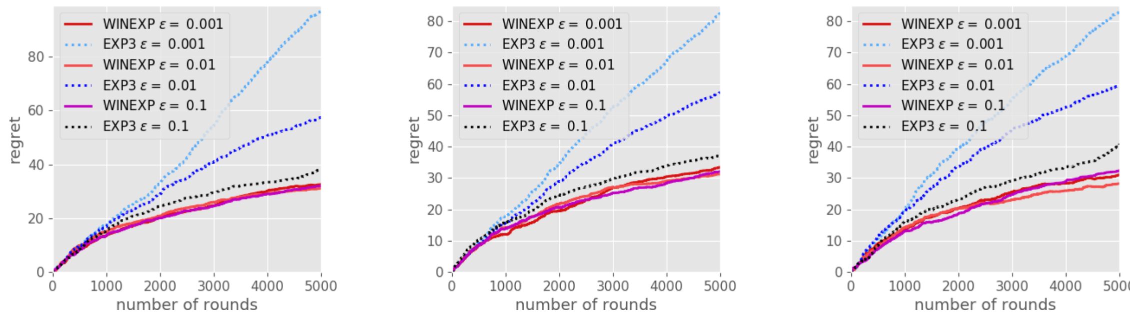
In this section, we present our results from our comparative analysis between EXP3 and WIN-EXP on a simulated sponsored search system that we built and which is a close proxy of the actual sponsored search algorithms deployed in the industry. We implemented the weighted GSP auction as described in definition 5.2. The auctioneer draws i.i.d rank scores that are bidder and timestep specific; as is the case throughout our paper, here we have assumed a stochastic auctioneer with respect to the rank scores. After bidding, the bidder will always be able to observe the allocation function. Now, if the bidder gets allocated to a slot and she gets clicked, then, she is able observe the *value* and the payment curve. Values are assumed to lie in $[0,1]$ and they are obliviously adversarial. Finally, the bidders choose bids from some ϵ-discretized grid of $[0,1]$ (in all experiments, apart from the ones comparing the regrets for different discretizations, we use $\epsilon = 0.001$) and update the probabilities of choosing each discrete bid according to EXP3 or WIN-EXP. Regret is measured with respect to the best fixed discretized bid in-hindsight.

We distinguish three cases of the bidding behavior of the rest of the bidders (apart from our learner): i) all of them are *stochastic* adversaries drawing bids at random from some distribution, ii) there is a subset of them that are bidding *adaptively*, by using an EXP3 online learning algorithm and iii) there is a subset of them that are bidding *adaptively* but using a WINEXP online learning algorithm (self play). Validating our theoretical claims, in all three cases, WIN-EXP outperforms EXP3 in terms of regret. We generate the event of whether a player gets clicked or not as follows: we draw a timestep specific threshold value in $[0,1]$ and the learner gets a click in case the CTR of the slot he got allocated (if any) is greater than this threshold value. Note here that the choice of a timestep specific threshold imposes *monotonicity*, i.e. if the learner did not get a click when allocated to a slot with CTR $x_t(b)$, she should not be able to get a click from slots with lower CTRs. We ran simulations with 3 different distributions of generating CTRs, so as to understand what is the effect of different levels of click-through-rates on the variance of our regret: i) $x_t(b) \sim U[0.1, 1]$, ii) $x_t(b) \sim U[0.3, 1]$ and iii) $x_t(b) \sim U[0.5, 1]$. Finally, we address robustness of our results to errors in CTR estimation. For this, we add random noise to the CTRs of each slot and we report to the learners the allocation and payment functions that correspond to the erroneous CTRs. The noise was generated according to a normal distribution $\mathcal{N}(0, \frac{1}{m})$, where m could be viewed as the number of training samples on which a machine learning algorithm was ran in order to output the CTR estimate ($m = 100, 1000, 10000$).

For each of the following simulations, there are $N = 20$ bidders, $k = 3$ slots and we ran the experiment for each round for a total of 10 times. For the simulations that correspond to adaptive adversaries we used $a = 4$ adversaries. Our results for the *cumulative regret* are presented below. We measured ex-post regret with respect to the realized thresholds that determine whether a player gets clicked or not. Note that the solid plots correspond to the emprical mean of the regret, whereas the opaque bands correspond to the 10-th and 90-th percentile.

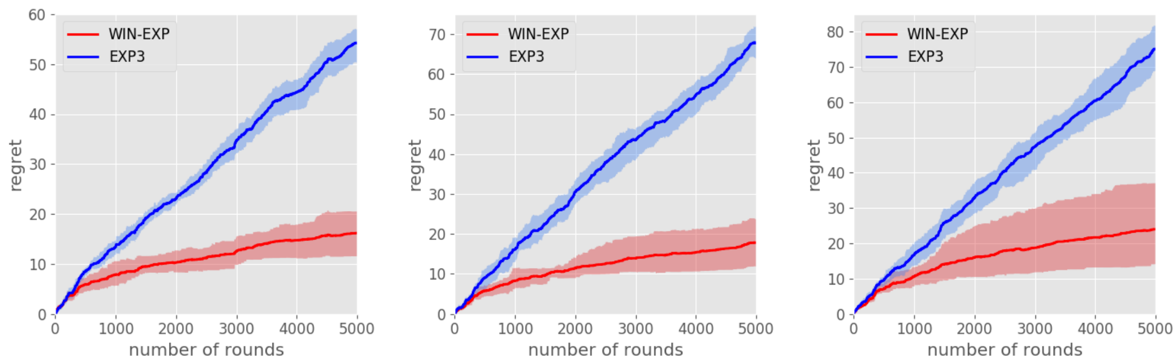
Different discretizations. In Figure 5.2 we present the comparative analysis of the estimated average regret of WIN-EXP vs EXP3 for different discretizations, ϵ, of the bidding space when the learner faces stochastic adversaries (Fig. 5.2a), adaptive ones using EXP3 (Fig. 5.2b) and adaptive ones using WINEXP (Fig. 5.2c). As it was expected from the theoretical analysis, observe that the regret of WIN-EXP, as the disretized space ($|B|$) increases exponentially, remains almost unchanged compared to the regret of EXP3. What is more, for $T = 5000$ the regret of WIN-EXP has almost been stabilized, while the regret of EXP3 has just started reaching the plateau phase yet. In summary, *finer discretization of the bid space helps our* WIN-EXP *algorithm's performance, but hurts the performance of EXP3.*

Different CTR Distributions. In Figures 5.3, 5.4 and 5.5 we present the results of the regret performance of WIN-EXP compared to EXP3, when the learner discretizes the bidding space



(a) *Stochastic Adversaries* **(b)** *Adaptive Adversaries EXP3* **(c)** *Adaptive Adversaries WINEXP*

Figure 5.2: *Regret of* WIN-EXP *vs EXP3 for different discretizations ϵ (CTR $\sim U[0.5, 1]$).*

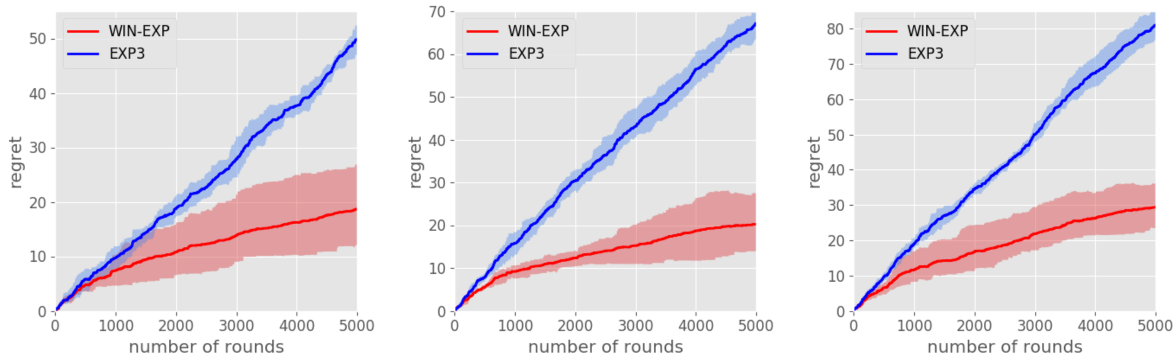


(a) *CTR* $\sim U[0.1, 1]$ **(b)** *CTR* $\sim U[0.3, 1]$ **(c)** *CTR* $\sim U[0.5, 1]$

Figure 5.3: *Regret of* WIN-EXP *vs EXP3 for different CTR distributions and stochastic adversaries,* $\epsilon = 0.001$.

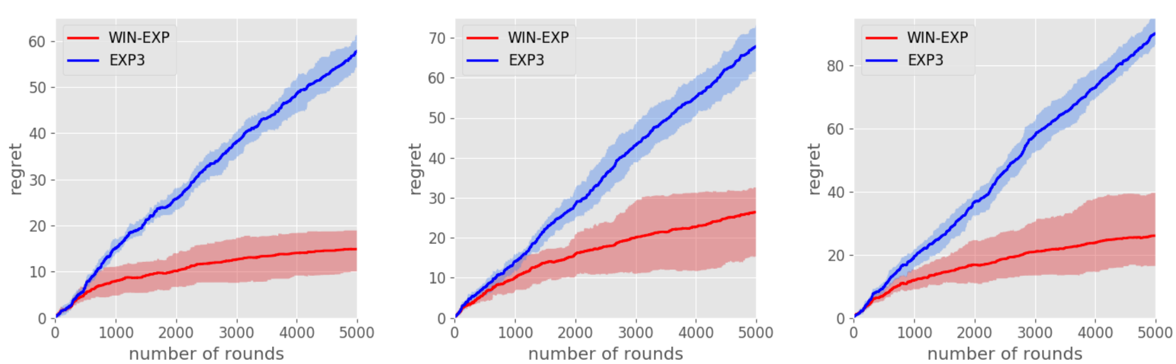
with $\epsilon = 0.001$ and when she faces stochastic, adaptive adversaries using EXP3 and adaptive adversaries using WINEXP, respectively. For all three cases, the estimated average regret of WIN-EXP is less than the estimated average regret that EXP3 yields. As the CTRs are shifted to higher values, the probability of getting clicked (based on our threshold concept) increases and thus, WIN-EXP can acquires information about the value more frequently.

Robustness to Noisy CTR Estimates. In Figures 5.6, 5.7, 5.8 we empirically tested the robustness of our algorithm to random perturbations of the allocation function that the auctioneer presents to the learner, for perturbations of the form $\mathcal{N}\left(0, \frac{1}{m}\right)$, where m could be viewed as the number of training examples used from the auctioneer in order to derive an approxima-



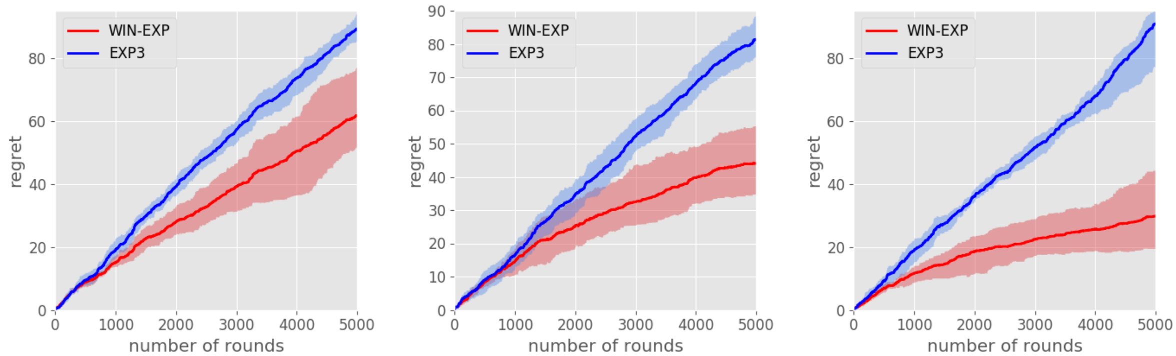
(a) $CTR \sim U[0.1, 1]$ **(b)** $CTR \sim U[0.3, 1]$ **(c)** $CTR \sim U[0.5, 1]$

Figure 5.4: *Regret of* WIN-EXP *vs EXP3 for different CTR distributions and adaptive EXP3 adversaries,* $\epsilon = 0.001$.



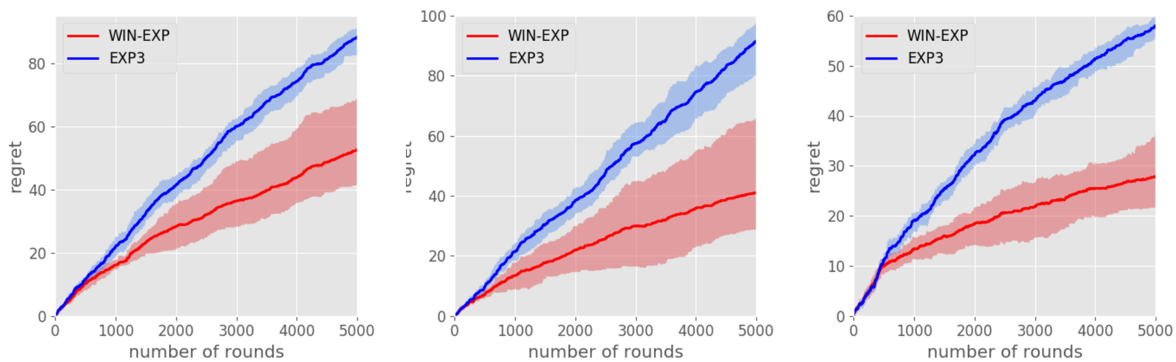
(a) $CTR \sim U[0.1, 1]$ **(b)** $CTR \sim U[0.3, 1]$ **(c)** $CTR \sim U[0.5, 1]$

Figure 5.5: *Regret of* WIN-EXP *vs EXP3 for different CTR distributions and adaptive WINEXP adversaries,* $\epsilon = 0.001$.



(a) $m = 100$ **(b)** $m = 1000$ **(c)** $m = 10000$

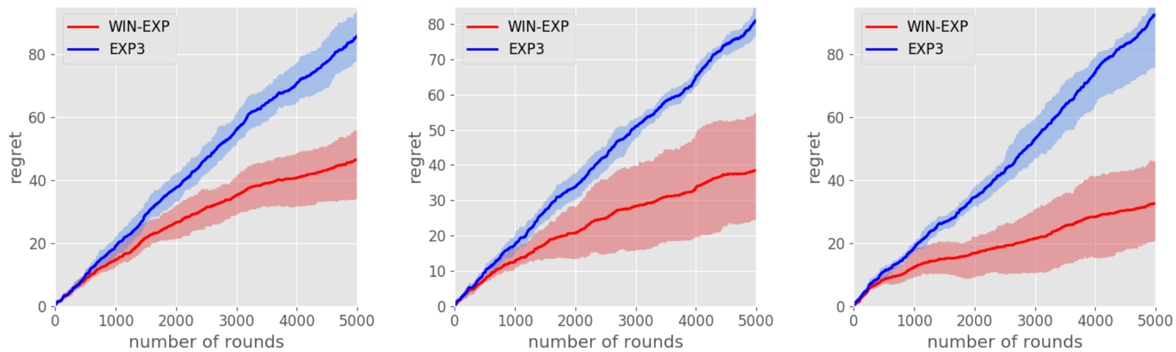
Figure 5.6: *Regret of* WIN-EXP *vs EXP3 with noise* $\sim \mathcal{N}\left(0, \frac{1}{m}\right)$ *for stochastic adversaries,* $\epsilon = 0.001$.



(a) $m = 100$ **(b)** $m = 1000$ **(c)** $m = 10000$

Figure 5.7: *Regret of* WIN-EXP *vs EXP3 with noise* $\sim \mathcal{N}\left(0, \frac{1}{m}\right)$ *for adaptive EXP3 adversaries,* $\epsilon = 0.001$.

tion of the allocation curve. Even when the number of training samples is relatively small ($m = 100$) WINEXP clearly outperforms EXP3 in terms of regret, i.e., it is more robust to such perturbations. The latter validates one of our claims throughout the paper; namely, that even though the learner might not see the exact allocation curve, but a randomly perturbed proxy WIN-EXP still performs better than the EXP3.



(a) $m = 100$ **(b)** $m = 1000$ **(c)** $m = 10000$

Figure 5.8: *Regret of* WIN-EXP *vs EXP3 with noise* $\sim \mathcal{N}\left(0, \frac{1}{m}\right)$ *for adaptive WINEXP adversaries,* $\epsilon = 0.001$.

5.8 Discussion

We addressed learning in repeated mechanism design scenarios were players do not know their valuation for the items at sale. We formulated an online learning framework with partial feedback which captures the information available to players in typical auction settings like sponsored search and provided an algorithm which achieves almost full information regret rates. Hence, we portrayed that not knowing your valuation is a benign form of incomplete information learning in auctions. Our experimental evaluation also showed that the improved learning rates are robust to violations of our assumptions and are valid even when the information assumed is corrupted. We believe that exploring further avenues of relaxing the informational assumptions or being more robust to erroneous information given by the auction system is an interesting future research direction. We believe that our outcome based learning framework can facilitate such future work.

Chapter 6

Conclusion

In this thesis, I mainly focus on two research topics: *Economic Design via Machine Learning* and *Learning in Online Markets*. These two topics form a loop between economic design and machine learning. Part I and Part II focus on the first topic and Part III is dedicated to the second topic.

In Part I, I initiate the exploration of the use of tools from deep learning for the automated design of optimal auctions. I model an auction as multi-layer neural network, frame optimal auction design as a constrained learning problem, and show how it can be solved using standard machine learning pipelines. In particular, I propose three neural network architectures: *MyersonNet*, *RochetNet*, and *RegretNet* to handle different auction settings. The first two networks use characterization results from economic theory to guarantee DSIC, but are inflexible in that they can only handle single-item auctions and single-bidder auctions. In contrast, RegretNet is a general approach that uses negated, expected revenue as the loss function. Crucially, we must achieve incentive compatibility. RegretNet is trained subject to a constraint that the expected ex post regret to bidders for bidding truthfully is zero. When attained exactly, this is equivalent to DSIC up to types with zero measure. RegretNet can handle multi-item and multi-bidder settings.

In Part II, I extend RegretNet to design multi-item auctions where each buyer has a private budget and generalize RegretNet to Bayesian incentive compatibility. This work shows the flexibility and generality of the RegretNet framework, where it can be easily extended to handle other economic constraints and different incentive constraints. Indeed, a very recent work by [Kuo+20] extends the RegretNet framework to handle fairness constraint in auctions. The transformation proposed in Chapter 3 opens a future direction to transform a RegretNet to a fully-BIC mechanism, without suffering any welfare loss and with only a negligible loss in revenue. This work is the first to discuss the transformation of an ε-EEIC mechanism (i.e. with tiny expected ex post regret) to a fully-BIC mechanism. I also show that if we want to preserve welfare it is impossible to transform an ε-EEIC mechanism to a fully-BIC mechanism with only negligible revenue loss unless the type distribution is uniform. A future direction is to understand whether we can transform an ε-EEIC mechanism into a fully-BIC mechanism while allowing negligible loss of revenue as well as welfare negligible loss of welfare.

In Part III, I consider the problem of *Learning in Online Markets*, where I focus on learning to bid in repeated auctions. In Chapter 4, I analyze the convergence results of mean-based learning algorithms in repeated single-item second price auctions, single-item first price auctions, and multi-slot VCG auctions. This work strengthens the convergence results available for repeated games, where previous results showed only that no-regret learning algorithms converge to CCE in repeated games and in a time-average manner. A natural future direction of this work is to understand when we can get stronger convergence results in general-sum repeated games. In Chapter 5, I propose an online learning algorithm for bidders in repeated auctions who don't know their own value before submitting the bid. The algorithm has exponentially faster convergence in terms of the size of the action space than generic bandit algorithms. The algorithm can be easily extended to complex auction formats, e.g., sponsored search auctions, and continuous valuation settings. Indeed, this work abstracts the real bidding system and motivates how to bid in real ads systems. A natural future direction is to understand how to incorporate into the algorithm approximate feedback about the allocation and payments and how to extend to online contextual auctions.

Further Discussion

I believe there is ample future opportunity for applying deep learning in the context of economic design. I have demonstrated how standard pipelines augmented with consideration of incentive alignment can re-discover and empirically surpass the analytical and computational progress in optimal auction design that has been made over the past 30-40 years. While RegretNet is shown to have advantages over standard LP approaches for automated mechanism design in terms of computational efficiency and representation, a natural next step would be to scale this approach further to industry scale (e.g., through standardized benchmarking suites and innovations in network architecture). I also see promise for the framework in Chapter 1 in advancing economic theory, for example in supporting or refuting conjectures and as an assistant in guiding new economic discovery, if we can continue to capture incentive constraints within the architecture of neural network. As we show, RegretNet achieves good empirical results for automated mechanism design problems, by avoiding enumerative representations and exponentially many constraints in the naive LP approaches. I believe the RegretNet framework can be extended to other non-standard mechanism design settings, when there is no succinct characterization of incentive compatibility.

The RegretNet framework can also be used in an online manner: we can train RegretNet based on the bidding data from a previous period, deploy it in the next period, and iteratively update RegretNet in this way. When using RegretNet online, we may face new challenges that the bidding data are from strategic bidders who aim to maximize long term reward, it is an open question to understand how to design robust mechanisms against these strategic bidders through deep learning. I provide generalization bounds for the RegretNet framework, however, there is no theoretical guarantee in regard to solving the optimization problem associated with training RegretNet. In future work, it will be interesting to understand the convergence and optimality guarantees of RegretNet. It is worth exploring the opportunity to extend the menu-based characterization (used in RochetNet) to multi-item settings and capture it within the architecture of neural networks in order to guarantee exact incentive compatibility. In addition, the RegretNet framework can directly handle correlated valuations since it only takes the bids profile as an input. Despite the negative sample complexity results for the correlated valuation setting in the worst case [DHN14], it will be interesting to empirically understand the sample size needed for training RegretNet for different valuation settings.

In Chapter 3, the approximately IC to BIC transformation builds upon the finite type setting and needs oracle access to interim quantities of the original mechanism. A natural next step is to generalize the transformation to continuous valuation settings with only sample access to the original approximately IC mechanism, and still in polynomial time. The negative results for ε-EEIC mechanism with non-uniform type distributions only hold for the setting that we need to preserve welfare. This leaves an open question that whether we can transform an ε-EEIC mechanism (e.g. the mechanism modeled by RegretNet) to a DSIC mechanism with only negligible loss of revenue (regardless of welfare). If such a transformation exists, then we can transform any mechanism modeled by RegretNet to an exactly DSIC mechanism with negligible loss of revenue.

To conclude, we are moving into an era that will make available a vast amount of detailed data. This creates new challenges and opportunities for the market designer to rethink how to utilize this data to design better markets and mechanisms. Moreover, with the information feedback provided by the platform, participants will need to meet the challenges of understanding how mechanisms work and learning to maximize their long term reward. These challenges motivate the work presented in this thesis, this work rethinking the problems that lie in the loop between economic design and machine learning. I believe that with continued progress, we can create better markets that serve to benefit society.

References

[AB09a] M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. 1st. Cambridge University Press, 2009 (cit. on p. 24).

[AB09b] I. Ashlagi and M. Braverman. "Ascending Unit Demand Auctions with Budget Limits". In: *Techinical Report* (2009) (cit. on pp. 41, 55).

[AB19] E. M. Azevedo and E. Budish. "Strategy-proofness in the Large". In: *Review of Economic Studies* 86.1 (2019), pp. 81–116 (cit. on p. 63).

[Aga+14] A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. Schapire. "Taming the Monster: A Fast and Simple Algorithm for Contextual Bandits". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by E. P. Xing and T. Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, 2014, pp. 1638–1646 (cit. on pp. 105, 108).

[AHK12] S. Arora, E. Hazan, and S. Kale. "The Multiplicative Weights Update Method: a Meta-Algorithm and Applications." In: *Theory of Computing* 8.1 (2012), pp. 121–164 (cit. on p. 222).

[AJ13] S. Adlakha and R. Johari. "Mean field equilibrium in dynamic games with strategic complementarities". In: *Operations Research* 61.4 (2013), pp. 971–989 (cit. on p. 108).

[Alb+21] M. Albert, V. Conitzer, G. Lopomo, and P. Stone. "Mechanism Design for Correlated Valuations: Efficient Methods for Revenue Maximization". In: *Operations Research* (2021) (cit. on p. 3).

[Alo+13] N. Alon, N. Cesa-bianchi, C. Gentile, and Y. Mansour. “From Bandits to Experts: A Tale of Domination and Independence”. In: *Advances in Neural Information Processing Systems 26*. Ed. by C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger. 2013, pp. 1610–1618 (cit. on pp. 104, 105, 109).

[Alo+15] N. Alon, N. Cesa-Bianchi, O. Dekel, and T. Koren. “Online learning with feedback graphs: Beyond bandits”. In: *Conference on Learning Theory*. 2015, pp. 23–35 (cit. on pp. 109, 221).

[AM06] L. M. Ausubel and P. Milgrom. “The Lovely but Lonely Vickrey Auction”. In: *Combinatorial Auctions, chapter 1*. MIT Press, 2006 (cit. on p. 42).

[Ami+15] K. Amin, R. Cummings, L. Dworkin, M. Kearns, and A. Roth. “Online Learning and Profit Maximization from Revealed Preferences.” In: *AAAI*. 2015, pp. 770–776 (cit. on p. 107).

[ARS14] K. Amin, A. Rostamizadeh, and U. Syed. “Repeated contextual auctions with strategic buyers”. In: *Advances in Neural Information Processing Systems*. 2014, pp. 622–630 (cit. on p. 107).

[Ash+10] I. Ashlagi, M. Braverman, A. Hassidim, R. Lavi, and M. Tennenholtz. “Position Auctions with Budgets: Existence and Uniqueness.” In: *The B.E. Journal of Theoretical Economics* 10.1 (2010) (cit. on pp. 38, 41).

[Ath18] S. Athey. “The Impact of Machine Learning on Economics”. In: *The Economics of Artificial Intelligence: An Agenda*. NBER Chapters. National Bureau of Economic Research, Inc, 2018, pp. 507–547 (cit. on p. 1).

[Aue+02] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. “The nonstochastic multiarmed bandit problem”. In: *SIAM journal on computing* 32.1 (2002), pp. 48–77 (cit. on pp. 105, 215).

[AZLS19] Z. Allen-Zhu, Y. Li, and Z. Song. “A Convergence Theory for Deep Learning via Over-Parameterization”. In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. 2019, pp. 242–252 (cit. on p. 31).

[Bab+14] M. Babaioff, N. Immorlica, B. Lucier, and S. M. Weinberg. "A Simple and Approximately Optimal Mechanism for an Additive Buyer". In: *Proceedings of the 55th IEEE Symposium on Foundations of Computer Science*. 2014, pp. 21–30 (cit. on p. 9).

[Bal+14] S. R. Balseiro, J. Feldman, V. Mirrokni, and S. Muthukrishnan. "Yield Optimization of Display Advertising with Ad Exchange". In: *Management Science* 60.12 (2014), pp. 2886–2907 (cit. on p. 63).

[Bal+19] S. Balseiro, N. Golrezaei, M. Mahdian, V. Mirrokni, and J. Schneider. "Contextual Bandits with Cross-Learning". In: *Advances in Neural Information Processing Systems 32*. 2019, pp. 9679–9688 (cit. on pp. 91, 94).

[BBW15] S. R. Balseiro, O. Besbes, and G. Y. Weintraub. "Repeated auctions with budgets in ad exchanges: Approximations and design". In: *Management Science* 61.4 (2015), pp. 864–884 (cit. on p. 108).

[BCB+12] S. Bubeck, N. Cesa-Bianchi, et al. "Regret analysis of stochastic and nonstochastic multi-armed bandit problems". In: *Foundations and Trends® in Machine Learning* 5.1 (2012), pp. 1–122 (cit. on pp. 104, 108).

[BG17] S. Balseiro and Y. Gur. "Learning in Repeated Auctions with Budgets: Regret Minimization and Equilibrium". In: (2017) (cit. on p. 108).

[BH01] C. Boutilier and H. H. Hoos. "Bidding Languages for Combinatorial Auctions". In: *Proceedings of the 17th International Joint Conference on Artificial Intelligence*. 2001, pp. 1211–1217 (cit. on p. 156).

[BH11] X. Bei and Z. Huang. "Bayesian Incentive Compatibility via Fractional Assignments". In: *Proceedings of the 2011 Annual ACM-SIAM Symposium on Discrete Algorithms* (2011), pp. 720–733 (cit. on pp. 4, 67–70).

[Bha+12] S. Bhattacharya, G. Goel, S. Gollapudi, and K. Munagala. "Budget-Constrained Auctions with Heterogeneous Items". In: *Theory of Computing* 8.20 (2012), pp. 429–460 (cit. on p. 41).

[Bir46] G. Birkhoff. "Three observations on linear algebra". In: *Univ. Nac. Tucuman. Revista A* 5 (1946), pp. 147–151 (cit. on p. 44).

[Blu+04] A. Blum, V. Kumar, A. Rudra, and F. Wu. "Online learning in online auctions". In: *Theoretical Computer Science* 324.2-3 (2004), pp. 137–146 (cit. on pp. 1, 107).

[Blu+08] A. Blum, M. Hajiaghayi, K. Ligett, and A. Roth. "Regret minimization and the price of total anarchy". In: *Proceedings of the fortieth annual ACM symposium on Theory of computing*. ACM. 2008, pp. 373–382 (cit. on p. 108).

[BM07] A. Blum and Y. Mansour. "Learning, Regret Minimization, and Equilibria". In: *Algorithmic Game Theory*. Cambridge University Press, 2007, 79–102 (cit. on pp. 2, 86, 91).

[BMM15] A. Blum, Y. Mansour, and J. Morgenstern. "Learning Valuation Distributions from Partial Observation." In: *AAAI*. 2015, pp. 798–804 (cit. on p. 107).

[Bor+05] C. Borgs, J. Chayes, N. Immorlica, M. Mahdian, and A. Saberi. "Multi-unit Auctions with Budget-constrained Bidders". In: *Proceedings of the 6th ACM Conference on Electronic Commerce*. 2005, pp. 44–51 (cit. on pp. 40, 55).

[Bor91] K. C. Border. "Implementation of Reduced Form Auctions: A Geometric Approach". In: *Econometrica* 59.4 (1991) (cit. on pp. 74, 194).

[BP18] J. P. Bailey and G. Piliouras. "Multiplicative Weights Update in Zero-Sum Games". In: *Proceedings of the 2018 ACM Conference on Economics and Computation*. ACM. 2018, pp. 321–338 (cit. on pp. 89, 91).

[Bra+18] M. Braverman, J. Mao, J. Schneider, and M. Weinberg. "Selling to a No-Regret Buyer". In: *Proceedings of the 2018 ACM Conference on Economics and Computation*. ACM. 2018, pp. 523–538 (cit. on pp. 91, 93).

[BSV16] M.-F. Balcan, T. Sandholm, and E. Vitercik. "Sample Complexity of Automated Mechanism Design". In: *Proceedings of the 29th Conference on Neural Information Processing Systems*. 2016, pp. 2083–2091 (cit. on p. 9).

[BSV18] M.-F. Balcan, T. Sandholm, and E. Vitercik. “A General Theory of Sample Complexity for Multi-Item Profit Maximization”. In: *Proceedings of the 2018 ACM Conference on Economics and Computation*. EC ’18. 2018, pp. 173–174 (cit. on p. 3).

[BSV19] M. Balcan, T. Sandholm, and E. Vitercik. “Estimating Approximate Incentive Compatibility”. In: *Proceedings of the 2019 ACM Conference on Economics and Computation*. EC’19. 2019, p. 867 (cit. on p. 65).

[Bud+13] E. Budish, Y.-K. Che, F. Kojima, and P. Milgrom. “Designing Random Allocation Mechanisms: Theory and Applications”. In: *American Economic Review* 103.2 (2013), pp. 585–623 (cit. on p. 23).

[Cai+19] Y. Cai, A. Oikonomou, G. Velegkas, and M. Zhao. “An Efficient ε-BIC to BIC Transformation and Its Application to Black-Box Reduction in Revenue Maximization”. In: (2019). arXiv: 1911.10172 [cs.GT] (cit. on pp. 65, 67, 68, 183).

[Car+15] I. Caragiannis, C. Kaklamanis, P. Kanellopoulos, M. Kyropoulou, B. Lucier, R. P. Leme, and É. Tardos. “Bounding the inefficiency of outcomes in generalized second price auctions”. In: *Journal of Economic Theory* 156 (2015), pp. 343–388 (cit. on p. 108).

[Car12] G. Carroll. “When Are Local Incentive Constraints Sufficient?” In: *Econometrica* 80.2 (2012), pp. 661–686 (cit. on p. 63).

[CB+15] R. Colini-Baldeschi, S. Leonardi, M. Henzinger, and M. Starnberger. “On Multiple Keyword Sponsored Search Auctions with Budgets”. In: *ACM Trans. Econ. Comput.* 4.1 (Dec. 2015), 2:1–2:34 (cit. on p. 38).

[CBGM15] N. Cesa-Bianchi, C. Gentile, and Y. Mansour. “Regret minimization for reserve prices in second-price auctions”. In: *IEEE Transactions on Information Theory* 61.1 (2015), pp. 549–564 (cit. on p. 107).

[CDW12a] Y. Cai, C. Daskalakis, and M. S. Weinberg. “An algorithmic characterization of multi-dimensional mechanisms”. In: *Proceedings of the 44th ACM Symposium on Theory of Computing*. 2012, pp. 459–478 (cit. on pp. 3, 9, 40, 63).

[CDW12b] Y. Cai, C. Daskalakis, and M. S. Weinberg. "Optimal Multi-dimensional Mechanism Design: Reducing Revenue to Welfare Maximization". In: *Proceedings of the 53rd IEEE Symposium on Foundations of Computer Science*. 2012, pp. 130–139 (cit. on pp. 3, 9, 40, 63).

[CDW13] Y. Cai, C. Daskalakis, and S. M. Weinberg. "Understanding Incentives: Mechanism Design Becomes Algorithm Design". In: *Proceedings of the 54th IEEE Symposium on Foundations of Computer Science*. 2013, pp. 618–627 (cit. on pp. 3, 9, 40, 63).

[CG00] Y.-K. Che and I. Gale. "The Optimal Mechanism for Selling to a Budget-Constrained Buyer". In: *Journal of Economic Theory* 92.2 (2000), pp. 198–233. ISSN: 0022-0531 (cit. on pp. 4, 39, 40, 43, 52, 54).

[CG98] Y.-K. Che and I. Gale. "Standard Auctions with Financially Constrained Bidders". In: *The Review of Economic Studies* 65.1 (1998), pp. 1–21. ISSN: 00346527, 1467937X (cit. on p. 40).

[Cha+10] S. Chawla, J. D. Hartline, D. L. Malec, and B. Sivan. "Multi-parameter mechanism design and sequential posted pricing". In: *Proceedings of the 42th ACM Symposium on Theory of Computing*. 2010, pp. 311–320 (cit. on p. 9).

[CHK16] A. Cohen, T. Hazan, and T. Koren. "Online learning with feedback graphs without the graphs". In: *International Conference on Machine Learning*. 2016, pp. 811–819 (cit. on p. 109).

[CHN14] S. Chawla, J. D. Hartline, and D. Nekipelov. "Mechanism design for data science". In: *ACM Conference on Economics and Computation, EC '14, Stanford , CA, USA, June 8-12, 2014*. 2014, pp. 711–712 (cit. on p. 107).

[CL06] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006 (cit. on pp. 5, 86, 91, 219).

[CLA15] A. Choromanska, Y. LeCun, and G. B. Arous. "The landscape of the loss surfaces of multilayer networks". In: *Proceedings of The 28th Conference on Learning Theory, Paris, France, July 3-6, 2015*. 2015, pp. 1756–1760 (cit. on p. 10).

[CMM11] S. Chawla, D. L. Malec, and A. Malekian. “Bayesian Mechanism Design for Budget-constrained Agents”. In: *Proceedings of the 12th ACM Conference on Electronic Commerce*. 2011, pp. 253–262 (cit. on p. 41).

[CR14] R. Cole and T. Roughgarden. “The sample complexity of revenue maximization”. In: *Proceedings of the 46th ACM Symposium on Theory of Computing*. 2014, pp. 243–252 (cit. on pp. 3, 9, 107).

[CS02] V. Conitzer and T. Sandholm. “Complexity of Mechanism Design”. In: *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*. 2002, pp. 103–110 (cit. on pp. 1, 3, 9, 10, 34, 40, 63, 81, 196).

[CS04] V. Conitzer and T. Sandholm. “Self-interested automated mechanism design and implications for optimal combinatorial auctions”. In: *Proceedings of the 5th ACM Conference on Electronic Commerce*. 2004, pp. 132–141 (cit. on pp. 9, 10, 34, 63).

[Cur+20] M. J. Curry, P. Chiang, T. Goldstein, and J. Dickerson. “Certifying Strategyproof Auction Networks”. In: *Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. 2020 (cit. on pp. 4, 40).

[CZ17] Y. Cai and M. Zhao. “Simple Mechanisms for Subadditive Buyers via Duality”. In: *Proceedings of the 49th ACM Symposium on Theory of Computing*. 2017, pp. 170–183 (cit. on pp. 3, 5, 9, 64, 65, 67–69, 183).

[Das15] C. Daskalakis. “Multi-item auctions defying intuition?” In: *SIGecom Exchanges* 14.1 (2015), pp. 41–75 (cit. on p. 3).

[DDT13] C. Daskalakis, A. Deckelbaum, and C. Tzamos. “Mechanism design via optimal transport”. In: *Proceedings of the 14th ACM Conference on Electronic Commerce*. 2013, pp. 269–286 (cit. on pp. 30, 32, 178, 180, 182).

[DDT17] C. Daskalakis, A. Deckelbaum, and C. Tzamos. “Strong Duality for a Multiple-Good Monopolist”. In: *Econometrica* 85 (3 2017), pp. 735–767 (cit. on pp. 8, 160, 179).

[DHN14] S. Dughmi, L. Han, and N. Nisan. “Sampling and Representation Complexity of Revenue Maximization”. In: *Web and Internet Economics*. 2014, pp. 277–291 (cit. on p. 132).

[DHS15] P. Dütting, M. Henzinger, and M. Starnberger. “Auctions for Heterogeneous Items and Budget Limits”. In: *ACM Trans. Econ. Comput.* 4.1 (2015), 4:1–4:17 (cit. on p. 41).

[DHW11] P. Dütting, M. Henzinger, and I. Weber. “An Expressive Mechanism for Auctions on the Web”. In: *Proceedings of the 20th International Conference on World Wide Web*. 2011, pp. 127–136 (cit. on p. 41).

[Dia+12] I. Diakonikolas, C. Papadimitriou, G. Pierrakos, and Y. Singer. “Efficiency-Revenue Trade-Offs in Auctions”. In: *Automata, Languages, and Programming*. 2012, pp. 488–499 (cit. on p. 63).

[DLN08] S. Dobzinski, R. Lavi, and N. Nisan. “Multi-unit Auctions with Budget Limits”. In: *Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science*. Washington, DC, USA, 2008, pp. 260–269. ISBN: 978-0-7695-3436-7 (cit. on p. 41).

[DP18] C. Daskalakis and I. Panageas. “Last-Iterate Convergence: Zero-Sum Games and Constrained Min-Max Optimization”. In: *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2018 (cit. on p. 91).

[DRY15] P. Dhangwatnotai, T. Roughgarden, and Q. Yan. “Revenue maximization with a single sample”. In: *Games and Economic Behavior* 91 (2015), pp. 318–333 (cit. on p. 107).

[DT13] N. Dikkala and É. Tardos. “Can Credit Increase Revenue?” In: *Web and Internet Economics - 9th International Conference, WINE 2013, Cambridge, MA, USA, December 11-14, 2013, Proceedings*. 2013, pp. 121–133 (cit. on p. 108).

[Dug+17] S. Dughmi, J. D. Hartline, R. Kleinberg, and R. Niazadeh. "Bernoulli Factories and Black-Box Reductions in Mechanism Design". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. 2017 (cit. on p. 68).

[DW12] C. Daskalakis and S. M. Weinberg. "Symmetries and Optimal Multi-Dimensional Mechanism Design". In: *Proceedings of the 13th ACM Conference on Electronic Commerce*. 2012, pp. 370–387 (cit. on pp. 5, 31, 64, 65, 67–69, 183).

[Düt+14] P. Dütting, F. Fischer, P. Jirapinyo, J. Lai, B. Lubin, and D. C. Parkes. "Payment Rules through Discriminant-Based Classifiers". In: *ACM Transactions on Economics and Computation* 3.1 (2014), p. 5 (cit. on pp. 5, 9, 10, 40, 63–66, 70).

[Düt+19a] P. Dütting, Z. Feng, N. Golowich, H. Narasimhan, D. C. Parkes, and S. Ravindranath. "Machine Learning for Optimal Economic Design". In: *The Future of Economic Design*. Ed. by J.-F. Laslier, H. Moulin, M. R. Sanver, and W. S. Zwicker. Vol. 70. Springer, 2019, pp. 495–515 (cit. on p. 11).

[Düt+19b] P. Dütting, Z. Feng, H. Narasimhan, D. Parkes, and S. S. Ravindranath. "Optimal Auctions through Deep Learning". In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, 2019 (cit. on pp. 45, 63, 65, 70).

[Fel+16] M. Feldman, T. Koren, R. Livni, Y. Mansour, and A. Zohar. "Online Pricing with Strategic and Patient Buyers". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc., 2016, pp. 3864–3872 (cit. on p. 107).

[FL19] D. Fudenberg and A. Liang. "Predicting and Understanding Initial Play". In: *American Economic Review* (2019). Forthcoming (cit. on p. 11).

[FL98] D. Fudenberg and D. K. Levine. *The theory of learning in games*. MIT Press, 1998 (cit. on pp. 5, 86).

[FLN16] M. Feldman, B. Lucier, and N. Nisan. "Correlated and Coarse equilibria of Single-item auctions". In: *International Conference on Web and Internet Economics*. Springer. 2016, pp. 131–144 (cit. on p. 92).

[FNP18] Z. Feng, H. Narasimhan, and D. C. Parkes. "Deep Learning for Revenue-Optimal Auctions with Budgets". In: *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*. 2018, pp. 354–362 (cit. on pp. 11, 15, 63, 65, 70).

[FPS18] Z. Feng, C. Podimata, and V. Syrgkanis. "Learning to Bid Without Knowing Your Value". In: *Proceedings of the 2018 ACM Conference on Economics and Computation*. 2018, 505–522 (cit. on pp. 91, 94).

[FSS19] Z. Feng, O. Schrijvers, and E. Sodomka. "Online Learning for Measuring Incentive Compatibility in Ad Auctions". In: *The World Wide Web Conference*. WWW '19. 2019, pp. 2729–2735 (cit. on p. 65).

[GB+18] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. "Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules". In: *ACS Central Science* 4.2 (2018), pp. 268–276 (cit. on p. 36).

[GB10] X. Glorot and Y. Bengio. "Understanding the difficulty of training deep feed-forward neural networks". In: *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*. 2010 (cit. on p. 26).

[GC10] M. Guo and V. Conitzer. "Computationally Feasible Automated Mechanism Design: General Approach and Case Studies". In: *Proceedings of the 24th AAAI Conference on Artificial Intelligence*. 2010 (cit. on p. 10).

[GK18] Y. Giannakopoulos and E. Koutsoupias. "Duality and Optimality of Auctions for Uniform Distributions". In: *SIAM Journal on Computing*. Vol. 47. 2018, pp. 121–165 (cit. on pp. 8, 30, 31, 34).

[GLL12] A. Gyorgy, T. Linder, and G. Lugosi. "Efficient tracking of large classes of experts". In: *IEEE Transactions on Information Theory* 58.11 (2012), pp. 6709–6725 (cit. on pp. 121, 217–219).

[GN17] Y. A. Gonczarowski and N. Nisan. “Efficient empirical revenue maximization in single-parameter auction environments”. In: *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*. 2017, pp. 856–868 (cit. on p. 9).

[GNP18] N. Golowich, H. Narasimhan, and D. C. Parkes. “Deep Learning for Multi-Facility Location Mechanism Design”. In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2018, pp. 261–267 (cit. on pp. 11, 40).

[Goo18a] Google. *AdWords Bid Simulator*. https://support.google.com/adwords/answer/2470105?hl=en&ref_topic=3122864. [Online; accessed 15-February-2018]. 2018 (cit. on pp. 106, 117).

[Goo18b] Google. *Bid Landscapes*. https://developers.google.com/adwords/api/docs/guides/bid-landscapes. [Online; accessed 15-February-2018]. 2018 (cit. on pp. 106, 117).

[GW18] Y. A. Gonczarowski and S. M. Weinberg. “The Sample Complexity of Up-to-ϵ Multi-Dimensional Revenue Maximization”. In: *59th IEEE Annual Symposium on Foundations of Computer Science*. 2018, pp. 416–426 (cit. on p. 9).

[Har+17] J. S. Hartford, G. Lewis, K. Leyton-Brown, and M. Taddy. “Deep IV: A Flexible Approach for Counterfactual Prediction”. In: *Proceedings of the 34th International Conference on Machine Learning*. 2017, pp. 1414–1423 (cit. on p. 11).

[Has18] T. Hashimoto. “The generalized random priority mechanism with budgets”. In: *Journal of Economic Theory* 177 (2018), pp. 708 –733 (cit. on p. 75).

[HH19] N. Haghpanah and J. Hartline. “When is Pure Bundling Optimal?” In: *Review of Economic Studies* (2019). Revise and resubmit (cit. on pp. 8, 161).

[HKM11] J. D. Hartline, R. Kleinberg, and A. Malekian. “Bayesian Incentive Compatibility via Matchings”. In: *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*. 2011 (cit. on pp. 67, 68).

[HL10] J. D. Hartline and B. Lucier. “Bayesian Algorithmic Mechanism Design”. In: *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*. 2010 (cit. on pp. 67, 68).

[HMR18] Z. Huang, Y. Mansour, and T. Roughgarden. “Making the Most of Your Samples”. In: *SIAM Journal on Computing* 47.3 (2018), pp. 651–674 (cit. on p. 9).

[HN17] S. Hart and N. Nisan. “Approximate revenue maximization with multiple items”. In: *Journal of Economic Theory* 172 (2017), pp. 313–347 (cit. on p. 9).

[HST15] J. Hartline, V. Syrgkanis, and E. Tardos. “No-Regret Learning in Bayesian Games”. In: *Advances in Neural Information Processing Systems*. Vol. 28. Curran Associates, Inc., 2015, pp. 3061–3069 (cit. on p. 88).

[HWLB16] J. S. Hartford, J. R. Wright, and K. Leyton-Brown. “Deep learning for predicting human strategic behavior”. In: *Proceedings of the 29th Conference on Neural Information Processing Systems*. 2016, pp. 2424–2432 (cit. on p. 11).

[IJS11] K. Iyer, R. Johari, and M. Sundararajan. “Mean field equilibria of dynamic auctions with learning”. In: *ACM SIGecom Exchanges* 10.3 (2011), pp. 10–14 (cit. on p. 108).

[JMM07] P. Jehiel, M. Meyer-ter-Vehn, and B. Moldovanu. “Mixed bundling auctions”. In: *Journal of Economic Theory* 134.1 (2007), pp. 494–512 (cit. on p. 162).

[Kaw16] K. Kawaguchi. “Deep Learning without Poor Local Minima”. In: *Proceedings of the 30th Conference on Neural Information Processing Systems*. 2016, pp. 586–594 (cit. on p. 10).

[KB15] D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2015 (cit. on pp. 26, 50, 52).

[Kle05] R. D. Kleinberg. “Nearly tight bounds for the continuum-armed bandit problem”. In: *Advances in Neural Information Processing Systems*. 2005, pp. 697–704 (cit. on p. 118).

[KLM17] T. Koren, R. Livni, and Y. Mansour. “Bandits with Movement Costs and Adaptive Pricing”. In: *Proceedings of the 2017 Conference on Learning Theory*. Ed. by S. Kale and O. Shamir. Vol. 65. Proceedings of Machine Learning Research. Amsterdam, Netherlands: PMLR, 2017, pp. 1242–1268 (cit. on p. 107).

[KMR03] J. Kornai, E. Maskin, and G. Roland. “Understanding the Soft Budget Constraint”. In: *Journal of Economic Literature* 41.4 (2003), pp. 1095–1136. ISSN: 00220515 (cit. on p. 39).

[KN14] Y. Kanoria and H. Nazerzadeh. “Dynamic Reserve Prices for Repeated Auctions: Learning from Bids - Working Paper”. In: *Web and Internet Economics - 10th International Conference, WINE 2014, Beijing, China, December 14-17, 2014. Proceedings*. 2014, p. 232 (cit. on p. 107).

[Kri02] V. Krishna. *Auction Theory*. Elsevier Science, 2002 (cit. on pp. 87, 98).

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems 25*. 2012, pp. 1097–1105 (cit. on p. 31).

[KSU08] R. Kleinberg, A. Slivkins, and E. Upfal. “Multi-armed bandits in metric spaces”. In: *Proceedings of the fortieth annual ACM symposium on Theory of computing*. ACM. 2008, pp. 681–690 (cit. on p. 118).

[Kuo+20] K. Kuo, A. Ostuni, E. Horishny, M. J. Curry, S. Dooley, P. yeh Chiang, T. Goldstein, and J. P. Dickerson. *ProportionNet: Balancing Fairness and Revenue for Auction Design with Deep Learning*. 2020. arXiv: `2010.06398 [cs.GT]` (cit. on p. 130).

[Lah+18] S. Lahaie, A. M. Medina, B. Sivan, and S. Vassilvitskii. “Testing Incentive Compatibility in Display Ad Auctions”. In: *Proceedings of the 27th International World Wide Web Conference (WWW)*. 2018 (cit. on p. 65).

[Lah11] S. Lahaie. “A Kernel-Based Iterative Combinatorial Auction”. In: *Proceedings of the 25th AAAI Conference on Artificial Intelligence*. 2011, pp. 695–700 (cit. on p. 9).

[Lan14] S. E. Land. *Bing Ads Launches "Bid Landscape", A Keyword Level Bid Simulator Tool*. https://searchengineland.com/bing-ads-launches-bid-landscape-keyword-level-bid-simulator-tool-187219. [Online; accessed 15-February-2018]. 2014 (cit. on p. 106).

[Lou+17] C. Louizos, U. Shalit, J. M. Mooij, D. Sontag, R. S. Zemel, and M. Welling. "Causal Effect Inference with Deep Latent-Variable Models". In: *Proceedings of the 30th Conference on Neural Information Processing Systems*. 2017, pp. 6449–6459 (cit. on p. 11).

[LP07] S. Lahaie and D. M. Pennock. "Revenue Analysis of a Family of Ranking Rules for Keyword Auctions". In: *Proceedings of the 8th ACM Conference on Electronic Commerce (EC)*. 2007, pp. 50–56 (cit. on p. 63).

[LP12] B. Lubin and D. C. Parkes. "Approximate strategyproofness". In: *Current Science* 103.9 (2012), pp. 1021–1032 (cit. on p. 63).

[LR96] J.-J. Laffont and J. Robert. "Optimal auction with financially constrained buyers". In: *Economics Letters* 52.2 (1996), pp. 181–186. ISSN: 0165-1765 (cit. on p. 40).

[LST16] T. Lykouris, V. Syrgkanis, and É. Tardos. "Learning and efficiency in games with dynamic population". In: *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics. 2016, pp. 120–129 (cit. on p. 122).

[Mas00] E. S. Maskin. "Auctions, development, and privatization: Efficient auctions with liquidity-constrained buyers". In: *European Economic Review* 44.4 (2000), pp. 667 –681. ISSN: 0014-2921 (cit. on p. 40).

[Mer+19] P. Mertikopoulos, H. Zenati, B. Lecouat, C.-S. Foo, V. Chandrasekhar, and G. Piliouras. "Optimistic Mirror Descent in Saddle-Point Problems: Going the Extra (Gradient) Mile". In: *ICLR '19 - International Conference on Learning Representations*. New Orleans, United States, May 2019, pp. 1–23 (cit. on p. 91).

[Mic18] Microsoft. *BingAds, Bid Landscapes*. https://advertise.bingads.microsoft.com/en-us/resources/training/bidding-and-traffic-estimation. [Online; accessed 15-February-2018]. 2018 (cit. on pp. 106, 117).

[MM14] A. M. Medina and M. Mohri. "Learning theory and algorithms for revenue optimization in second price auctions with reserve". In: *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 2014, pp. 262–270 (cit. on p. 107).

[MM16] M. Mohri and A. M. Medina. "Learning Algorithms for Second-Price Auctions with Reserve". In: *Journal of Machine Learning Research* 17 (2016), 74:1–74:25 (cit. on p. 9).

[MR15] J. Morgenstern and T. Roughgarden. "On the Pseudo-Dimension of Nearly Optimal Auctions". In: *Proceedings of the 28th Conference on Neural Information Processing Systems*. 2015, pp. 136–144 (cit. on p. 9).

[MR16] J. Morgenstern and T. Roughgarden. "Learning simple auctions". In: *Proceedings of the 29th Conference on Learning Theory*. 2016, pp. 1298–1318 (cit. on p. 9).

[MS11] S. Mannor and O. Shamir. "From Bandits to Experts: On the Value of Side-Observations." In: *NIPS*. Ed. by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger. 2011, pp. 684–692 (cit. on pp. 104, 105, 109).

[MV06] A. Manelli and D. Vincent. "Bundling as an optimal selling mechanism for a multiple-good monopolist". In: *Journal of Economic Theory* 127.1 (2006), pp. 1–35 (cit. on pp. 8, 27, 28, 30, 34).

[MV08] A. Malakhov and R. V. Vohra. "Optimal auctions for asymmetrically budget constrained bidders". In: *Review of Economic Design* 12.4 (2008), p. 245 (cit. on pp. 4, 39, 40, 52, 54–56).

[MV17] A. M. Medina and S. Vassilvitskii. "Revenue Optimization with Approximate Bid Predictions". In: *CoRR* abs/1706.04732 (2017). arXiv: 1706.04732. URL: http://arxiv.org/abs/1706.04732 (cit. on p. 107).

[Mye81] R. Myerson. “Optimal Auction Design”. In: *Mathematics of Operations Research* 6 (1981), pp. 58–73 (cit. on pp. 3, 8, 13, 38, 152).

[NAP16] H. Narasimhan, S. Agarwal, and D. C. Parkes. “Automated Mechanism Design without Money via Machine Learning”. In: *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. 2016, pp. 433–439 (cit. on pp. 3, 9, 40).

[Neu53] J. von Neumann. “A Certain Zero-sum Two-person Game equivalent to the Optimal Assignment Problem”. In: *Contributions to the Theory of Games (AM-28), Volume II* (1953), pp. 5–12 (cit. on p. 44).

[Nis+07] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic game theory*. Cambridge University Press, 2007 (cit. on pp. 5, 86).

[NP16] H. Narasimhan and D. C. Parkes. “A general statistical framework for designing strategy-proof assignment mechanisms”. In: *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*. 2016, pp. 527–536 (cit. on pp. 40, 75).

[NST15] D. Nekipelov, V. Syrgkanis, and E. Tardos. “Econometrics for learning agents”. In: *Proceedings of the Sixteenth ACM Conference on Economics and Computation*. ACM. 2015, pp. 1–18 (cit. on pp. 2, 88, 91).

[OS11] M. Ostrovsky and M. Schwarz. “Reserve prices in internet advertising auctions: A field experiment”. In: *Proceedings of the 12th ACM conference on Electronic commerce*. ACM. 2011, pp. 59–60 (cit. on p. 107).

[Pal83] T. Palfrey. “Bundling Decisions by a Multiproduct Monopolist with Incomplete Information”. In: *Econometrica* 51.2 (1983), pp. 463–83 (cit. on p. 34).

[Pav11] G. Pavlov. “Optimal Mechanism for Selling Two Goods”. In: *B.E. Journal of Theoretical Economics* 11 (1 2011), pp. 1–35 (cit. on pp. 8, 27, 28, 161).

[PNB16] A. Patel, T. Nguyen, and R. Baraniuk. “A Probabilistic Framework for Deep Learning”. In: *Proceedings of the 30th Conference on Neural Information Processing Systems*. 2016, pp. 2550–2558 (cit. on p. 10).

[PP19] C. Papadimitriou and G. Piliouras. “Game Dynamics As the Meaning of a Game”. In: *SIGecom Exch.* 16.2 (May 2019), pp. 53–63. ISSN: 1551-9031 (cit. on pp. 91, 94).

[PV14] M. M. Pai and R. Vohra. “Optimal auctions with financially constrained buyers”. In: *Journal of Economic Theory* 150 (2014), pp. 383 –425. ISSN: 0022-0531 (cit. on pp. 4, 39, 40, 43, 56).

[Rag+18] M. Raghu, A. Irpan, J. Andreas, R. Kleinberg, Q. V. Le, and J. M. Kleinberg. “Can Deep Reinforcement Learning Solve Erdos-Selfridge-Spencer Games?” In: *Proceedings of the 35th International Conference on Machine Learning*. 2018, pp. 4235–4243 (cit. on p. 11).

[Rah+20] J. Rahme, S. Jelassi, J. Bruna, and S. M. Weinberg. “A Permutation-Equivariant Neural Network Architecture For Auction Design”. In: *arXiv:2003.01497* (2020) (cit. on pp. 4, 40).

[RJW20] J. Rahme, S. Jelassi, and S. M. Weinberg. “Auction learning as a two-player game”. In: *arXiv:2006.05684* (2020) (cit. on pp. 4, 40).

[Roc87] J.-C. Rochet. “A Necessary and Sufficient Condition for Rationalizability in a Quasilinear Context”. In: *Journal of Mathematical Economics* 16 (1987), pp. 191–200 (cit. on pp. 9, 13, 14, 20).

[Rou09] T. Roughgarden. “Intrinsic robustness of the price of anarchy”. In: *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM. 2009, pp. 513–522 (cit. on p. 108).

[RS09] C. Rudin and R. E. Schapire. “Margin-based Ranking and an Equivalence between AdaBoost and RankBoost”. In: *Journal of Machine Learning Research* 10 (2009), pp. 2193–2232 (cit. on p. 17).

[RW18] A. Rubinstein and M. S. Weinberg. “Simple Mechanisms for a Subadditive Buyer and Applications to Revenue Monotonicity”. In: *ACM Trans. Econ. Comput.* (2018) (cit. on pp. 5, 64, 65, 67–69, 183).

[Sch+15] T. Schaul, J. Quan, I. Antonoglou, and D. Silver. "Prioritized experience replay". In: *arXiv preprint arXiv:1511.05952* (2015) (cit. on p. 207).

[Sil98] J. Sill. "Monotonic networks". In: *Proceedings of the 12th Conference on Neural Information Processing Systems*. 1998, pp. 661–667 (cit. on p. 154).

[SL15] T. Sandholm and A. Likhodedov. "Automated Design of Revenue-Maximizing Combinatorial Auctions". In: *Operations Research* 63.5 (2015), pp. 1000–1025 (cit. on pp. 9, 10, 40, 161–163).

[SSBD14] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. New York, NY, USA: Cambridge University Press, 2014 (cit. on p. 165).

[Sta14] S. M. Standard. *Google AdWords Improves The Bid Simulator Tool Feature*. `http://www.searchmarketingstandard.com/google-adwords-improves-the-bid-simulator-tool-feature`. [Online; accessed 15-February-2018]. 2014 (cit. on p. 106).

[STZ19] W. Shen, P. Tang, and S. Zuo. "Automated Mechanism Design via Neural Networks". In: *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*. Forthcoming. 2019 (cit. on pp. 4, 11, 20, 32, 40).

[Syr17] V. Syrgkanis. "A Sample Complexity Measure with Applications to Learning Optimal Auctions". In: *Proceedings of the 20th Conference on Neural Information Processing Systems*. 2017, pp. 5358–5365 (cit. on p. 9).

[Tac+19] A. Tacchetti, D. Strouse, M. Garnelo, T. Graepel, and Y. Bachrach. "A Neural Architecture for Designing Truthful and Efficient Auctions". In: *CoRR* abs/1907.05181 (2019). arXiv: `1907.05181` (cit. on p. 11).

[TNLB17] D. Thompson, N. Newman, and K. Leyton-Brown. "The Positronic Economist: A Computational System for Analyzing Economic Mechanisms". In: *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 2017, pp. 720–727 (cit. on p. 11).

[VHGS15] H. Van Hasselt, A. Guez, and D. Silver. "Deep reinforcement learning with double q-learning". In: *arXiv preprint arXiv:1509.06461* (2015) (cit. on p. 207).

[Wan+16] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas. "Dueling network architectures for deep reinforcement learning". In: *International conference on machine learning*. 2016, pp. 1995–2003 (cit. on p. 207).

[Wor18] Wordstream. *Bid Management Tools*. https://www.wordstream.com/bid-management-tools. [Online; accessed 15-February-2018]. 2018 (cit. on p. 106).

[WPR16] J. Weed, V. Perchet, and P. Rigollet. "Online learning in repeated auctions". In: *Conference on Learning Theory*. 2016, pp. 1562–1583 (cit. on pp. 91, 94, 104, 107, 114, 118, 119, 211, 212).

[Yao15] A. C.-C. Yao. "An n-to-1 Bidder Reduction for Multi-item Auctions and its Applications". In: *Proceedings of the 26th ACM-SIAM Symposium on Discrete Algorithms*. 2015, pp. 92–109 (cit. on p. 9).

[Yao17] A. C.-C. Yao. "Dominant-Strategy versus Bayesian Multi-item Auctions: Maximum Revenue Determination and Comparison". In: *Proceedings of the 18th ACM Conference on Economics and Computation*. 2017, pp. 3–20 (cit. on pp. 8, 195).

[Aue+95] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. "Gambling in a rigged casino: The adversarial multi-armed bandit problem". In: *Proceedings of IEEE 36th Annual Foundations of Computer Science*. 1995, pp. 322–331 (cit. on p. 91).

Appendix A

Appendix to Chapter 1

A.1 Additional Architectures

In this appendix we present our network architectures for multi-bidder single-item settings and for a general multi-bidder multi-item setting with combinatorial valuations.

A.1.1 The MyersonNet Approach

We start by describing an architecture that yields optimal DSIC auction for selling a single item to multiple buyers.

In the single-item setting, each bidder holds a private value $v_i \in \mathbb{R}_{\geqslant 0}$ for the item. We consider a randomized auction (g, p) that maps a reported bid profile $b \in \mathbb{R}^n_{\geqslant 0}$ to a vector of allocation probabilities $g(b) \in \mathbb{R}^n_{\geqslant 0}$, where $g_i(b) \in \mathbb{R}_{\geqslant 0}$ denotes the probability that bidder i is allocated the item and $\sum_{i=1}^n g_i(b) \leqslant 1$. We shall represent the payment rule p_i via a price conditioned on the item being allocated to bidder i, i.e. $p_i(b) = g_i(b)\, t_i(b)$ for some conditional payment function $t_i : \mathbb{R}^n_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$. The expected revenue of the auction, when bidders are truthful, is given by:

$$rev(g, p) \;=\; \mathbf{E}_{v \sim F}\Big[\sum_{i=1}^n g_i(v)\, t_i(v)\Big]. \tag{A.1}$$

The structure of the revenue-optimal auction is well understood for this setting.

Theorem A.1 (Myerson [Mye81])**.** *There exist a collection of monotonically non-decreasing functions,*

$\bar{\phi}_i : \mathbb{R}_{\geqslant 0} \to \mathbb{R}$ *called the ironed virtual valuation functions such that the optimal BIC auction for selling a single item is the DSIC auction that assigns the item to the buyer with the highest ironed virtual value* $\bar{\phi}_i(v_i)$ *provided that this value is non-negative, with ties broken in an arbitrary value-independent manner, and charges the bidders according to* $p_i(v_i) = v_i g_i(v_i) - \int_0^{v_i} g_i(t)\, dt$.

For distribution F_i with density f_i the virtual valuation function is $\psi_i(v_i) = v_i - (1 - F(v_i))/f(v_i)$. A distribution F_i with density f_i is regular if ψ_i is monotonically non-decreasing. For regular distributions $F_1, \ldots, F_n$ no ironing is required and $\bar{\phi}_i = \psi_i$ for all i.

If the virtual valuation functions $\psi_1, \ldots, \psi_n$ are furthermore monotonically increasing and not only monotonically non-decreasing, the optimal auction can be viewed as applying the monotone transformations to the input bids $\bar{b}_i = \bar{\phi}_i(b_i)$, feeding the computed virtual values to a second price auction (SPA) with zero reserve price, denoted (g^0, p^0), making an allocation according to $g^0(\bar{b})$, and charging a payment $\bar{\phi}_i^{-1}(p_i^0(\bar{b}))$ for winning bidder i. In fact, this auction is DSIC for any choice of strictly monotone transformations of the values:

Theorem A.2. *For any set of strictly monotonically increasing functions* $\bar{\phi}_1, \ldots, \bar{\phi}_n$*, an auction defined by outcome rule* $g_i = g_i^0 \circ \bar{\phi}$ *and payment rule* $p_i = \bar{\phi}_i^{-1} \circ p_i^0 \circ \bar{\phi}$ *is DSIC and IR, where* (g^0, p^0) *is the allocation and payment rule of a second price auction with zero reserve.*

For regular distributions with monotonically increasing virtual value functions designing an optimal DSIC auction thus reduces to finding the right strictly monotone transformations and corresponding inverses, and modeling a second price auction with zero reserve.

We present a high-level overview of a neural network architecture that achieves this in Figure A.1(a), and describe the components of this network in more detail in Section A.1.1 and Section A.1.1 below.

Our MyersonNet is tailored to monotonically increasing virtual value functions. For regular distributions with virtual value functions that are not strictly increasing and for irregular distributions this approach only yields approximately optimal auctions.

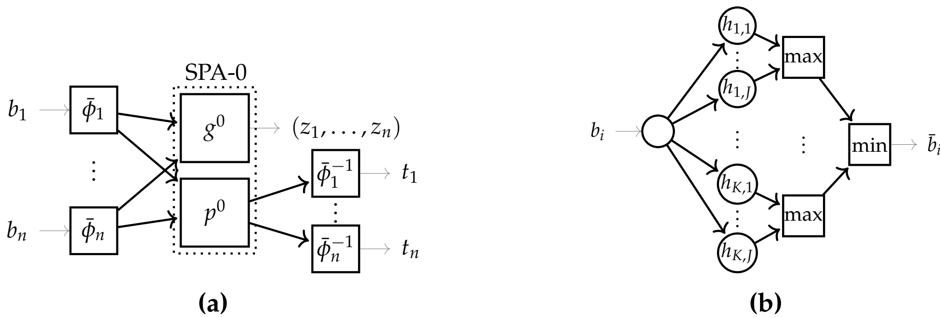


Figure A.1: *(a) MyersonNet: The network applies monotone transformations $\bar{\phi}_1, \ldots, \bar{\phi}_n$ to the input bids, passes the virtual values to the SPA-0 network in Figure A.2, and applies the inverse transformations $\bar{\phi}_1^{-1}, \ldots, \bar{\phi}_n^{-1}$ to the payment outputs. (b) Monotone virtual value function $\bar{\phi}_i$, where $h_{kj}(b_i) = e^{\alpha^i_{kj}} b_i + \beta^i_{kj}$.*

Modeling Monotone Transforms

We model each virtual value function $\bar{\phi}_i$ as a two-layer feed-forward network with min and max operations over linear functions. For K groups of J linear functions, with strictly positive slopes $w^i_{kj} \in \mathbb{R}_{>0},\ k = 1, \ldots, K,\ j = 1, \ldots, J$ and intercepts $\beta^i_{kj} \in \mathbb{R},\ k = 1, \ldots, K,\ j = 1, \ldots, J$, we define:

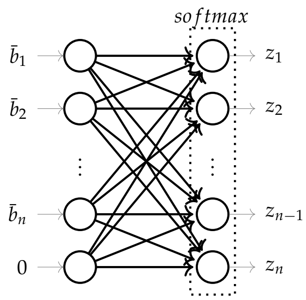
$$\bar{\phi}_i(b_i) \;=\; \min_{k \in [K]} \max_{j \in [J]} \; w^i_{kj}\, b_i + \beta^i_{kj}.$$

Since each of the above linear function is strictly non-decreasing, so is $\bar{\phi}_i$. In practice, we can set each $w^i_{kj} = e^{\alpha^i_{kj}}$ for parameters $\alpha^i_{kj} \in [-B, B]$ in a bounded range. A graphical representation of the neural network used for this transform is shown in Figure A.1(b). For sufficiently large K and J, this neural network can be used to approximate any continuous, bounded monotone function (that satisfies a mild regularity condition) to an arbitrary degree of accuracy [Sil98]. A particular advantage of this representation is that the inverse transform $\bar{\phi}^{-1}$ can be directly obtained from the parameters for the forward transform:

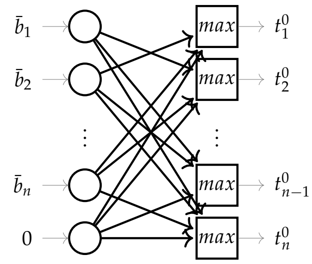
$$\bar{\phi}_i^{-1}(y) \;=\; \max_{k \in [K]} \min_{j \in [J]} \; e^{-\alpha^i_{kj}} (y - \beta^i_{kj}).$$

Modeling SPA with Zero Reserve

We also need to model a SPA with zero reserve (SPA-0) within the neural network structure. For the purpose of training, we employ a smooth approximation to the allocation rule using a neural network. Once we learn value functions using this approximate allocation rule, we use



(a) *Allocation rule g^0*



(b) *Payment rule t^0*

Figure A.2: *MyersonNet: SPA-0 network for (approximately) modeling a second price auction with zero reserve price. The inputs are (virtual) bids $\bar{b}_1, \ldots, \bar{b}_n$ and the output is a vector of assignment probabilities $z_1, \ldots, z_n$ and prices (conditioned on allocation) $t_1^0, \ldots, t_n^0$.*

them together with an exact SPA with zero reserve to construct the final auction.

The SPA-0 allocation rule g^0 can be approximated using a 'softmax' function on the virtual values $\bar{b}_1, \ldots, \bar{b}_n$ and an additional dummy input $\bar{b}_{n+1} = 0$:

$$g_i^0(\bar{b}) \;=\; \frac{e^{\kappa \bar{b}_i}}{\sum_{j=1}^{n+1} e^{\kappa \bar{b}_j}},\; i \in N, \tag{A.2}$$

where $\kappa > 0$ is a constant fixed a priori, and determines the quality of the approximation. The higher the value of κ, the better the approximation but the less smooth the resulting allocation function.

The SPA-0 payment to bidder i, conditioned on being allocated, is the maximum of the virtual values from the other bidders and zero:

$$t_i^0(\bar{b}) \;=\; \max\big\{ \max_{j \neq i} \bar{b}_j,\, 0 \big\},\; i \in N. \tag{A.3}$$

Let $g^{\alpha,\beta}$ and $t^{\alpha,\beta}$ denote the allocation and conditional payment rules for the overall auction in Figure A.1(a), where (α, β) are the parameters of the forward monotone transform. Given a sample of valuation profiles $\mathcal{S} = \{v^{(1)}, \ldots, v^{(L)}\}$ drawn i.i.d. from F, we optimize the parameters using the negated revenue on $\mathcal{S}$ as the error function, where the revenue is approximated as:

$$\widehat{rev}(g, t) \;=\; \frac{1}{L} \sum_{\ell=1}^{L} \sum_{i=1}^{n} g_i^{\alpha,\beta}(v^{(\ell)})\, t_i^{\alpha,\beta}(v^{(\ell)}). \tag{A.4}$$

We solve this training problemma using a minibatch stochastic gradient descent solver.

A.1.2 RegretNet for Combinatorial Valuations[1]

We next show how to adjust the RegretNet architecture so that it can handle bidders with general, combinatorial valuations. In the present work, we develop this architecture only for small number of items.[2] In this case, each bidder i reports a bid $b_{i,S}$ for every bundle of items $S \subseteq M$ (except the empty bundle, for which her valuation is taken as zero). 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. To prevent the items from being over-allocated, we require that the probability that an item appears in a bundle allocated to some bidder is at most one. We also require that the total allocation to a bidder is at most one:

$$\sum_{i \in N} \sum_{S \subseteq M : j \in S} z_{i,S} \leqslant 1, \forall j \in M; \tag{A.5}$$

$$\sum_{S \subseteq M} z_{i,S} \leqslant 1, \forall i \in N. \tag{A.6}$$

We refer to an allocation that satisfies constraints (A.5)–(A.6) as being *combinatorial feasible*. To enforce these constraints, the allocation network computes a set of scores for each bidder and a set of scores for each item. Specifically, there is a group of bidder-wise scores $s_{i,S}, \forall S \subseteq M$ for each bidder $i \in N$, and a group of item-wise scores $s^{(j)}_{i,S}, \forall i \in N, S \subseteq M$ for each item $j \in M$. Let $s, s^{(1)}, \ldots, s^{(m)} \in \mathbb{R}^{n \times 2^m}$ denote these bidder scores and item scores. Each group of scores is normalized using a softmax function: $\bar{s}_{i,S} = \exp(s_{i,S})/\sum_{S'} \exp(s_{i,S'})$ and $\bar{s}^{(j)}_{i,S} = \exp(s^{(j)}_{i,S})/\sum_{i',S'} \exp(s^{(j)}_{i',S'})$. The allocation for bidder i and bundle $S \subseteq M$ is defined as the minimum of the normalized bidder-wise score $\bar{s}_{i,S}$ and the normalized item-wise scores $\bar{s}^{(j)}_{i,S}$ for each $j \in S$:

$$z_{i,S} = \varphi^{CF}_{i,S}(s, s^{(1)}, \ldots, s^{(m)}) = \min\left\{\bar{s}_{i,S}, \bar{s}^{(j)}_{i,S} : j \in S\right\}.$$

Similar to the unit-demand setting, we first show that $\varphi^{CF}(s, s^{(1)}, \ldots, s^{(m)})$ is combinatorial feasible and that our constructive approach is without loss of generality. See Appendix A.3.5

[1]We thank Zihe Wang (Shanghai University of Finance and Economics) for pointing out that the incorrect decomposition statement of combinatorial feasible allocations in the initial draft.

[2]With more items, combinatorial valuations can be succinctly represented using appropriate bidding languages; see, e.g. [BH01].

for a proof.

Lemma A.1. *The matrix* $\varphi^{CF}(s, s^{(1)}, \ldots, s^{(m)})$ *is combinatorial feasible* $\forall\, s,\, s^{(1)}, \ldots, s^{(m)} \in \mathbb{R}^{n\times 2^m}$. *For any combinatorial feasible matrix* $z \in [0,1]^{n\times 2^m}$, $\exists\, s, s^{(1)}, \ldots, s^{(m)} \in \mathbb{R}^{n\times 2^m}$, *for which* $z = \varphi^{CF}(s, s^{(1)}, \ldots, s^{(m)})$.

In addition, we want to understand whether a *combinatorial feasible* allocation z can be *implemmaentable*, defined in the following way.

Definition A.1. *A fractional combinatorial allocation* z *is implemmaentable if and only if* z *can be represented as a convex combination of combinatorial feasible, deterministic allocations.*

Unfortunately, Example A.1 shows that a combinatorial feasible allocation may not have an integer decomposition, even for the case of two bidders and two items.

Example A.1. *Consider a setting with two bidders and two items, and the following fractional, combinatorial feasible allocation:*

$$z = \begin{bmatrix} z_{1,\{1\}} & z_{1,\{2\}} & z_{1,\{1,2\}} \\ z_{2,\{1\}} & z_{2,\{2\}} & z_{2,\{1,2\}} \end{bmatrix} = \begin{bmatrix} 3/8 & 3/8 & 1/4 \\ 1/8 & 1/8 & 1/4 \end{bmatrix}$$

Any integer decomposition of this allocation z *would need to have the following structure:*

$$\begin{aligned} z \;\; = \;\; & a\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} + b\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} + c\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + d\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + e\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\ & + f\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} + g\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + h\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \end{aligned}$$

where the coefficients sum to at most 1. Firstly, it is straightforward to see that $a = b = 1/4$. *Given the construction, we must have* $c + d = 3/8, e \geqslant 0$ *and* $f + g = 3/8, h \geqslant 0$. *Thus,* $a + b + c + d + e + f + g + h \geqslant 1/2 + 3/4 = 5/4$ *for any decomposition. Hence,* z *is not implemmaentable.*

To ensure that a combinatorial feasible allocation has an integer decomposition we need to introduce additional constraints. For the two items case, we introduce the following constraint:

$$\forall i, z_{i,\{1\}} + z_{i,\{2\}} \leqslant 1 - \sum_{i'=1}^{n} z_{i',\{1,2\}}. \tag{A.7}$$

Theorem A.3. *For $m = 2$, any combinatorial feasible allocation z with additional constraints (A.7) can be represented as a convex combination of matrices $B^1, \ldots, B^k$ where each B^ℓ is a combinatorial feasible, 0-1 allocation.*

Proof. Firstly, we observe in any deterministic allocation B^ℓ, if there exists an i, s.t. $B^\ell_{i,\{1,2\}} = 1$, then $\forall j \neq i, S : B^\ell_{j,S} = 0$. Therefore, we first decompose z into the following components,

$$z = \sum_{i=1}^{n} z_{i,\{1,2\}} \cdot B^i + C,$$

and

$$B^i_{j,S} = \begin{cases} 1 & \text{if } j = i, S = \{1,2\}, \text{and} \\ 0 & \text{otherwise.} \end{cases}$$

Then we want to argue that C can be represented as $\sum_{\ell=i+1}^k p_\ell \cdot B^\ell$, where $\sum_{\ell=i+1}^k p_\ell \leqslant 1 - \sum_{i=1}^n z_{i,\{1,2\}}$ and each B^ℓ is a feasible 0-1 allocation. Matrix C has all zeros in the last (items $\{1,2\}$) column, $\sum_i C_{i,\{1\}} \leqslant 1 - \sum_{i=1}^n z_{i,\{1,2\}}$, and $\sum_i C_{i,\{2\}} \leqslant 1 - \sum_{i=1}^n z_{i,\{1,2\}}$.

In addition, based on constraint (A.7), for each bidder i,

$$C_{i,\{1\}} + C_{i,\{2\}} = z_{i,\{1\}} + z_{i,\{2\}} \leqslant 1 - \sum_{i'=1}^{n} z_{i',\{1,2\}}.$$

Thus C is a doubly stochastic matrix with scaling factor $1 - \sum_{i'=1}^n z_{i',\{1,2\}}$. Therefore, we can always decompose C into a linear combination $\sum_{\ell=i+1}^k p_\ell \cdot B^\ell$, where $\sum_{\ell=i+1}^k p_\ell \leqslant 1 - \sum_{i'=1}^n z_{i',\{1,2\}}$ and each B^ℓ is a feasible 0-1 allocation. $\square$

We leave to future work to characterize the additional constraints needed for the multi-item ($m > 2$) case.

RegretNet for Two-item Auctions with Implementable Allocations

To accommodate the additional constraint (A.7) for the two items case we add an additional softmax layer for each bidder. In addition to the original (unnormalized) bidder-wise scores $s_{i,S}, \forall i \in N, S \subseteq M$ and item-wise scores $s^{(j)}_{i,S}, \forall i \in N, S \subseteq M, j \in M$ and their normalized counterparts $\bar{s}_{i,S}, \forall i \in N, S \subseteq M$ and $\bar{s}^{(j)}_{i,S}, \forall i \in N, S \subseteq M, j \in M$, the allocation network computes an additional set of scores for each bidder i, $s'^{(i)}_{i,\{1\}}, s'^{(i)}_{i,\{2\}}, s'^{(i)}_{1,\{1,2\}}, \cdots, s'^{(i)}_{n,\{1,2\}}$. These additional scores are then normalized using a softmax function as follows,

$$\forall i, k \in N, S \subseteq M, \quad \bar{s'}^{(i)}_{k,S} = \frac{\exp\left(s'^{(i)}_{k,S}\right)}{\exp\left(s'^{(i)}_{i,\{1\}}\right) + \exp\left(s'^{(i)}_{i,\{2\}}\right) + \sum_k \exp\left(s'^{(i)}_{k,\{1,2\}}\right)}.$$

To satisfy constraint (A.7) for each bidder i, we compute the normalized score $\bar{s'}_{i,S}$ for each i, S as,

$$\bar{s'}_{i,S} = \begin{cases} \bar{s'}^{(i)}_{i,S} & \text{if } S = \{1\} \text{ or } \{2\}, \text{and} \\ \min\left\{\bar{s'}^{(k)}_{i,S} : k \in N\right\} & \text{if } S = \{1,2\}. \end{cases}$$

Then the final allocation for each bidder i is:

$$z_{i,S} = \min\left\{\bar{s}_{i,S}, \bar{s'}_{i,S}, \bar{s}^{(j)}_{i,S} : j \in S\right\}.$$

The payment network for combinatorial bidders has the same structure as the one in Figure 1.2, computing a fractional payment $\tilde{p}_i \in [0,1]$ for each bidder i using a sigmoidal unit, and outputting a payment $p_i = \tilde{p}_i \sum_{S \subseteq M} z_{i,S}\, b_{i,S}$.

A.2 Additional Experiments

We present a broad range of additional experiments for the two main architectures used in the body of the paper, and additional ones for the architectures presented in Appendix A.1

A.2.1 Experiments with MyersonNet

We first evaluate the MyersonNet architecture introduced in Appendix A.1.1 for designing single-item auctions. We focus on settings with a small number of bidders because this is where revenue-optimal auctions are meaningfully different from efficient auctions. We present experimental results for the following four settings:

F. Three bidders with independent, regular, and symmetrically distributed valuations $v_i \sim U[0,1]$.

G. Five bidders with independent, regular, and asymmetrically distributed valuations $v_i \sim U[0,i]$.

Distribution	n	Opt	SPA	MyersonNet
		rev	*rev*	*rev*
Setting F	3	0.531	0.500	0.531
Setting G	5	2.314	2.025	2.305
Setting H	3	2.749	2.500	2.747
Setting I	3	2.368	2.210	2.355

Figure A.3: *The revenue of the single-item auctions obtained with MyersonNet.*

H. Three bidders with independent, regular, and symmetrically distributed valuations $v_i \sim Exp(3)$.

I. Three bidders with independent irregular distributions $F_{\text{irregular}}$, where each v_i is drawn from $U[0,3]$ with probability 3/4 and from $U[3,8]$ with probability 1/4.

We note that the optimal auctions for the first three distributions involve virtual value functions $\bar{\phi}_i$ that are strictly monotone. For the fourth and final distribution the optimal auction uses ironed virtual value functions that are not strictly monotone.

For the training set and test set we used 1,000 valuation profiles sampled i.i.d. from the respective valuation distribution. We modeled each transform $\bar{\phi}_i$ in the MyersonNet architecture using 5 sets of 10 linear functions, and we used $\kappa = 10^3$.

The results are summarized in Figure A.3. For comparison, we also report the revenue obtained by the optimal Myerson auction and the second price auction (SPA) without reserve. The auctions learned by the neural network yield revenue close to the optimal.

A.2.2 Additional Experiments with RochetNet and RegretNet

In addition to the experiments with RochetNet and RegretNet on the single bidder, multi-item settings in Section 1.5.3 we also considered the following settings:

J. Single additive bidder with independent preferences over two non-identically distributed items, where $v_1 \sim U[4,16]$ and $v_2 \sim U[4,7]$. The optimal mechanism is given by Daskalakis, Deckelbaum, and Tzamos [DDT17].

K. Single additive bidder with preferences over two items, where (v_1, v_2) are drawn jointly and uniformly from a unit triangle with vertices $(0,0), (0,1)$ and $(1,0)$. The optimal

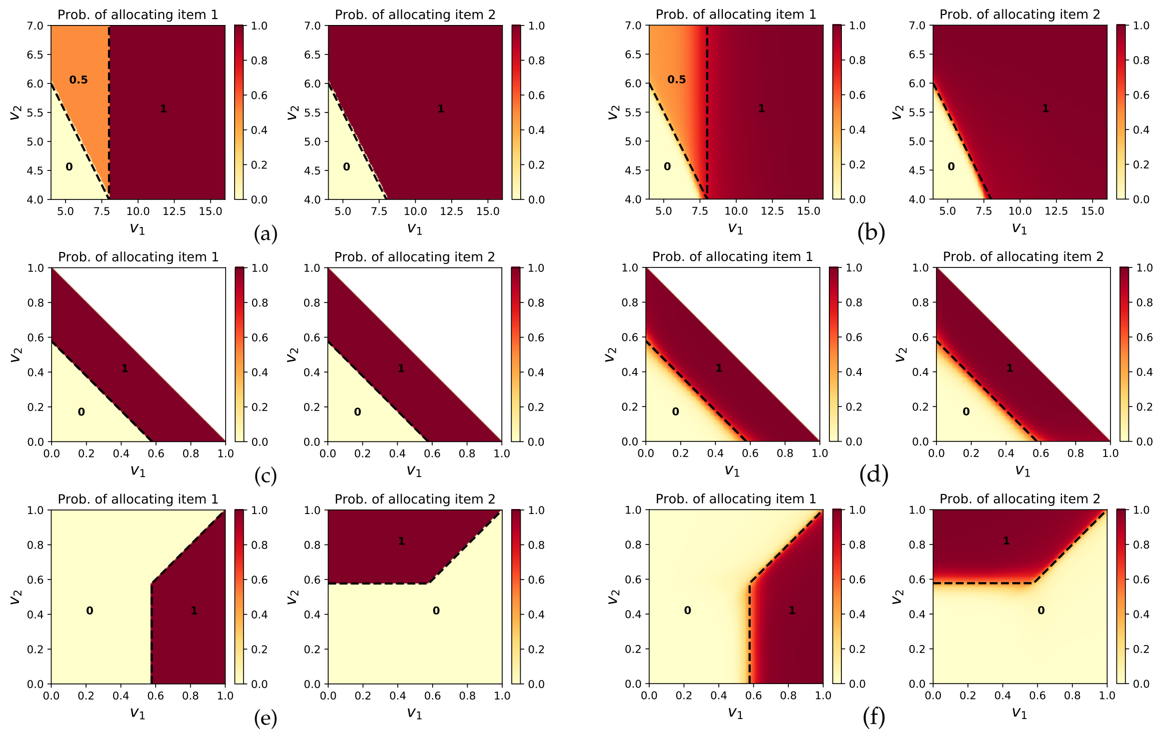


Figure A.4: *Side-by-side comparison of the allocation rules learned by RochetNet and RegretNet for single bidder, two items settings. Panels (a) and (b) are for Setting J, Panels (c) and (d) are for Setting K, and Panels (e) and (f) for Setting L. Panels describe the learned allocations for the two items (item 1 on the left, item 2 on the right). Optimal mechanisms are indicated via dashed lines and allocation probabilities in each region.*

mechanism is due to Haghpanah and Hartline [HH19].

L. Single unit-demand bidder with independent preferences over two items, where the item values $v_1, v_2 \sim U[0,1]$. See [Pav11] for the optimal mechanism.

We used RegretNet architectures with two hidden layers with 100 nodes each. The optimal allocation rules as well as a side-by-side comparison of those found by RochetNet and RegretNet are given in Figure A.4. Figure A.5 gives the revenue and regret achieved by RegretNet and the revenue achieved by RochetNet.

We find that in all three settings RochetNet recovers the optimal mechanism basically exactly, while RegretNet finds an auction that matches the optimal design to surprising accuracy.

A.2.3 Experiments with RegretNet with Combinatorial Valuations

We next compare our RegretNet architecture for combinatorial valuations described in Section A.1.2 to the computational results of Sandholm and Likhodedov [SL15] for the following

Distribution	Opt	RegretNet		RochetNet
	rev	*rev*	*rgt*	*rev*
Setting J	9.781	9.734	< 0.001	9.779
Setting K	0.388	0.392	< 0.001	0.388
Setting L	0.384	0.384	< 0.001	0.384

Figure A.5: *Test revenue and regret achieved by RegretNet and revenue achieved by RochetNet for Settings J–L.*

settings for which the optimal auction is not known:

M. Two additive bidders and two items, where bidders draw their value for each item independently from $U[0,1]$.[3]

N. Two bidders and two items, with item valuations $v_{1,1}, v_{1,2}, v_{2,1}, v_{2,2}$ drawn independently from $U[1,2]$ and set valuations $v_{1,\{1,2\}} = v_{1,1} + v_{1,2} + C_1$ and $v_{2,\{1,2\}} = v_{2,1} + v_{2,2} + C_2$, where C_1, C_2 are drawn independently from $U[-1,1]$.

O. Two bidders and two items, with item valuations $v_{1,1}, v_{1,2}$ drawn independently from $U[1,2]$, item valuations $v_{2,1}, v_{2,2}$ drawn independently from $U[1,5]$, and set valuations $v_{1,\{1,2\}} = v_{1,1} + v_{1,2} + C_1$ and $v_{2,\{1,2\}} = v_{2,1} + v_{2,2} + C_2$, where C_1, C_2 are drawn independently from $U[-1,1]$.

These settings correspond to Settings I.-III. described in Section 3.4 of [SL15]. These authors conducted extensive experiments with several different classes of incentive compatible mechanisms, and different heuristics for setting the parameters of these auctions. They observed the highest revenue for two classes of mechanisms that generalize mixed bundling auctions and λ-auctions [JMM07].

These two classes of mechanisms are the *Virtual Value Combinatorial Auctions* (VVCA) and *Affine Maximizer Auctions* (AMA). They also considered a restriction of AMA to bidder-symmetric auction (AMA_{bsym}). We use VVCA*, AMA*, and $\text{AMA}^*_{\text{bsym}}$ to denote the best mechanism in the respective class, as reported by Sandholm and Likhodedov and found using a heuristic grid search technique.

[3] This setting can be handled by the non-combinatorial RegretNet architecture and is included here for comparison to [SL15].

Distribution	RegretNet		VVCA*	$\text{AMA}^*_{\text{bsym}}$
	rev	*rgt*	*rev*	*rev*
Setting M	0.878	< 0.001	—	0.862
Setting N	2.860	< 0.001	—	2.765
Setting O	4.269	< 0.001	4.209	—

Figure A.6: *Test revenue and regret for RegretNet for Settings M–O, and comparison with the best performing VVCA and AMA_{bsym} auctions as reported by Sandholm and Likhodedov [SL15].*

For Setting M and N, Sandholm and Likhodedov observed the highest revenue for $\text{AMA}^*_{\text{bsym}}$, and for Setting O the best performing mechanism was VVCA*. Figure A.6 compares the performance of RegretNet to that of these best performing, benchmark mechanisms. To compute the revenue of the benchmark mechanisms we used the parameters reported in [SL15] (Table 2, p. 1011), and evaluated the respective mechanisms on the same test set used for RegretNet. Note that RegretNet is able to learn new auctions with improved revenue and tiny regret.

A.3 Omitted Proofs

In this appendix we present formal proofs for all theorems and lemmamas that were stated in the body of the paper or the other appendices. We first introduce some notation. We denote the inner product between vectors $a, b \in \mathbb{R}^d$ as $\langle a, b \rangle = \sum_{i=1}^d a_i b_i$. We denote the ℓ_1 norm for a vector x by $\|x\|_1$ and the induced ℓ_1 norm for a matrix $A \in \mathbb{R}^{k \times t}$ by $\|A\|_1 = \max_{1 \leqslant j \leqslant t} \sum_{i=1}^k A_{ij}$.

A.3.1 Proof of Lemma 1.1

Let $f_i(v; w) := \max_{v'_i \in V_i} u_i^w(v_i; (v'_i, v_{-i})) - u_i^w(v_i; (v_i, v_{-i}))$. Then we have $rgt_i(w) = \mathbf{E}_{v \sim F}[f_i(v; w)]$. Rewriting the expected value, we have

$$rgt_i(w) = \int_0^\infty \mathbb{P}(f_i(v; w) \geqslant x) dx \geqslant \int_0^{rgt_i^q(w)} \mathbb{P}(f_i(v; w) \geqslant x) dx \geqslant q \cdot rgt_i^q(w),$$

where the last inequality holds because for any $0 < x < rgt_i^q(w)$, $\mathbb{P}(f_i(v; w) \geqslant x) \geqslant \mathbb{P}(f_i(v; w) \geqslant rgt_i^q(w)) = q$. □

A.3.2 Proof of Theorem 1.2

We present the proof for auctions with general, randomized allocation rules. A randomized allocation rule $g_i : V \to [0,1]^{2^M}$ maps valuation profiles to a vector of allocation probabilities for bidder i, where $g_{i,S}(v) \in [0,1]$ denotes the probability that the allocation rule assigns subset of items $S \subseteq M$ to bidder i, and $\sum_{S \subseteq M} g_{i,S}(v) \leqslant 1$. This encompasses both the allocation rules for the combinatorial setting, and the allocation rules for the additive and unit-demand settings, which only output allocation probabilities for individual items. The payment function $p : V \to R^n$ maps valuation profiles to a payment for each bidder $p_i(v) \in \mathbb{R}$. For ease of exposition, we omit the superscripts "w". Recall that $\mathcal{M}$ is a class of auctions consisting of allocation and payment rules (g, p). As noted in the theorem statement, we will assume w.l.o.g. that for each bidder i, $v_i(S) \leqslant 1$, $\forall S \subseteq M$.

Definitions

Let $\mathcal{U}_i$ be the class of utility functions for bidder i defined on auctions in $\mathcal{M}$, i.e.,

$$\mathcal{U}_i = \left\{u_i : V_i \times V \to \mathbb{R} \,\middle|\, u_i(v_i, b) = v_i(g(b)) - p_i(b) \text{ for some } (g,p) \in \mathcal{M}\right\}.$$

and let $\mathcal{U}$ be the class of profile of utility functions defined on $\mathcal{M}$, i.e., the class of tuples $(u_1, \ldots, u_n)$ where each $u_i : V_i \times V \to \mathbb{R}$ and $u_i(v_i, b) = v_i(g(b)) - p_i(b), \forall i \in N$ for some $(g, p) \in \mathcal{M}$.

We will sometimes find it useful to represent the utility function as an inner product, i.e., treating v_i as a real-valued vector of length 2^M, we may write $u_i(v_i, b) = \langle v_i, g_i(b) \rangle - p_i(b)$.

Let $\mathrm{rgt} \circ \mathcal{U}_i$ be the class of all regret functions for bidder i defined on utility functions in $\mathcal{U}_i$, i.e.,

$$\mathrm{rgt} \circ \mathcal{U}_i = \left\{f_i : V \to \mathbb{R} \,\middle|\, f_i(v) = \max_{v_i'} u_i(v_i, (v_i', v_{-i})) - u_i(v_i, v) \text{ for some } u_i \in \mathcal{U}_i\right\},$$

and as before, let $\mathrm{rgt} \circ \mathcal{U}$ be defined as the class of profiles of regret functions.

Define the $\ell_{\infty,1}$ distance between two utility functions u and u' as

$$\max_{v,v'} \sum_i |u_i(v_i, (v_i', v_{-i})) - u_i(v_i, (v_i', v_{-i}))|$$

and let $\mathcal{N}_\infty(\mathcal{U}, \epsilon)$ denote the minimum number of balls of radius ϵ to cover $\mathcal{U}$ under this distance. Similarly, define the distance between u_i and u_i' as $\max_{v,v_i'} |u_i(v_i, (v_i', v_{-i})) - u_i'(v_i, (v_i', v_{-i}))|$, and let $\mathcal{N}_\infty(\mathcal{U}_i, \epsilon)$ denote the minimum number of balls of radius ϵ to cover $\mathcal{U}_i$ under this distance. Similarly, we define covering numbers $\mathcal{N}_\infty(\mathrm{rgt} \circ \mathcal{U}_i, \epsilon)$ and $\mathcal{N}_\infty(\mathrm{rgt} \circ \mathcal{U}, \epsilon)$ for the function classes $\mathrm{rgt} \circ \mathcal{U}_i$ and $\mathrm{rgt} \circ \mathcal{U}$ respectively.

Moreover, we denote the class of allocation functions as $\mathcal{G}$ and for each bidder i, $\mathcal{G}_i = \{g_i : V \to 2^M \mid g \in \mathcal{G}\}$. Similarly, we denote the class of payment functions by $\mathcal{P}$ and $\mathcal{P}_i = \{p_i : V \to \mathbb{R} \mid p \in \mathcal{P}\}$. We denote the covering number of $\mathcal{P}$ as $\mathcal{N}_\infty(\mathcal{P}, \epsilon)$ under the $\ell_{\infty,1}$ distance and the covering number for $\mathcal{P}_i$ using $\mathcal{N}_\infty(\mathcal{P}_i, \epsilon)$ under the $\ell_{\infty,1}$ distance.

Auxiliary Lemma

We will use a lemmama from [SSBD14]. Let $\mathcal{F}$ denote a class of bounded functions $f : Z \to [-c, c]$ defined on an input space Z, for some $c > 0$. Let D be a distribution over Z and $\mathcal{S} = \{z_1, \ldots, z_L\}$ be a sample drawn i.i.d. from D. We are interested in the gap between the expected value of a function f and the average value of the function on sample S, and would like to bound this gap uniformly for all functions in $\mathcal{F}$. For this, we measure the capacity of the function class $\mathcal{F}$ using the empirical Rademacher complexity on sample S, defined below:

$$\hat{\mathcal{R}}_L(\mathcal{F}) := \frac{1}{L} \mathbf{E}_\sigma \left[\sup_{f \in \mathcal{F}} \sum_{z_i \in S} \sigma_i f(z_i) \right],$$

where $\sigma \in \{-1, 1\}^L$ and each σ_i is drawn i.i.d from a uniform distribution on $\{-1, 1\}$. We then have:

Lemma A.2 ([SSBD14]). *Let $\mathcal{S} = \{z_1, \ldots, z_L\}$ be a sample drawn i.i.d. from some distribution D over Z. Then with probability of at least $1 - \delta$ over draw of $\mathcal{S}$ from D, for all $f \in \mathcal{F}$,*

$$\mathbf{E}_{z \sim D}[f(z)] \leqslant \frac{1}{L} \sum_{i=1}^{L} f(z_i) + 2\hat{\mathcal{R}}_L(\mathcal{F}) + 4C\sqrt{\frac{2\log(4/\delta)}{L}},$$

Generalization Bound for Revenue

We first prove the generalization bound for revenue. For this, we define the following auxiliary function class, where each $f : V \to \mathbb{R}_{\geqslant 0}$measures the total payments from some mechanism in $\mathcal{M}$:

$$\text{rev} \circ \mathcal{M} \;=\; \big\{f : V \to \mathbb{R}_{\geqslant 0} \,\big|\, f(v) \;=\; \textstyle\sum_{i=1}^{n} p_i(v), \text{ for some } (g, p) \in \mathcal{M}\big\}.$$

Note each function f in this class corresponds to a mechanism (g, p) in $\mathcal{M}$, and the expected value $\mathbf{E}_{v\sim D}[f(v)]$ gives the expected revenue from that mechanism. The proof then follows by an application of the uniform convergence bound in Lemma A.2 to the above function class, and by further bounding the Rademacher complexity term in this bound by the covering number of the auction class $\mathcal{M}$.

Applying Lemma A.2 to the auxiliary function class rev $\circ\, \mathcal{M}$, we get with probability of at least $1 - \delta$ over draw of L valuation profiles S from D, for any $f \in \text{rev} \circ \mathcal{M}$,

$$\begin{aligned} \mathbf{E}_{v\sim F}\Big[-\sum_{i\in N} p_i(v)\Big] \leqslant\; & -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i=1}^{n} p_i(v^{(\ell)}) \\ & + 2\hat{R}_L(\text{rev} \circ \mathcal{M}) \;+ Cn\sqrt{\frac{\log(1/\delta)}{L}}, \end{aligned} \tag{A.8}$$

All that remains is to bound the above empirical Rademacher complexity $\hat{R}_L(\text{rev} \circ \mathcal{M})$ in terms of the covering number of the payment class $\mathcal{P}$ and in turn in terms of the covering number of the auction class $\mathcal{M}$. Since we assume that the auctions in $\mathcal{M}$ satisfy individual rationality and $v(S) \leqslant 1, \forall S \subseteq M$, we have for any v, $p_i(v) \leqslant 1$.

By the definition of the covering number for the payment class, there exists a cover $\hat{\mathcal{P}}$ for $\mathcal{P}$ of size $|\hat{\mathcal{P}}| \leqslant \mathcal{N}_\infty(\mathcal{P}, \epsilon)$ such that for any $p \in \mathcal{P}$, there is a $f_p \in \hat{\mathcal{P}}$ with $\max_v \sum_i |p_i(v) - f_{p_i}(v)| \leqslant \epsilon$.

We thus have:

$$\begin{aligned}
\widehat{\mathcal{R}}_L(\mathrm{rev}\circ\mathcal{M}) &= \frac{1}{L}\mathbf{E}_\sigma\left[\sup_p \sum_{\ell=1}^{L}\sigma_\ell\cdot\sum_i p_i(v^{(\ell)})\right] \\
&= \frac{1}{L}\mathbf{E}_\sigma\left[\sup_p \sum_{\ell=1}^{L}\sigma_\ell\cdot\sum_i f_{p_i}(v^{(\ell)})\right] + \frac{1}{L}\mathbf{E}_\sigma\left[\sup_p \sum_{\ell=1}^{L}\sigma_\ell\cdot\sum_i p_i(v^{(\ell)}) - f_{p_i}(v^{(\ell)})\right] \\
&\leqslant \frac{1}{L}\mathbf{E}_\sigma\left[\sup_{\widehat{p}\in\widehat{\mathcal{P}}} \sum_{\ell=1}^{L}\sigma_\ell\cdot\sum_i \widehat{p}_i(v^{(\ell)})\right] + \frac{1}{L}\mathbf{E}_\sigma\|\sigma\|_1\epsilon \\
&\leqslant \sqrt{\sum_\ell(\sum_i \widehat{p}_i(v^\ell))^2}\sqrt{\frac{2\log(\mathcal{N}_\infty(\mathcal{P},\epsilon))}{L}} + \epsilon \\
&\leqslant 2n\sqrt{\frac{2\log(\mathcal{N}_\infty(\mathcal{P},\epsilon))}{L}} + \epsilon, \qquad \text{(A.9)}
\end{aligned}$$

where the second-last inequality follows from Massart's lemmama, and the last inequality holds because

$$\sqrt{\sum_\ell\left(\sum_i \widehat{p}_i(v^\ell)\right)^2} \leqslant \sqrt{\sum_\ell\left(\sum_i p_i(v^\ell) + n\epsilon\right)^2} \leqslant 2n\sqrt{L}.$$

We further observe that $\mathcal{N}_\infty(\mathcal{P},\epsilon) \leqslant \mathcal{N}_\infty(\mathcal{M},\epsilon)$. By the definition of the covering number for the auction class $\mathcal{M}$, there exists a cover $\widehat{\mathcal{M}}$ for $\mathcal{M}$ of size $|\widehat{\mathcal{M}}| \leqslant \mathcal{N}_\infty(\mathcal{M},\epsilon)$ such that for any $(g,p)\in\mathcal{M}$, there is a $(\widehat{g},\widehat{p})\in\widehat{\mathcal{M}}$ such that for all v,

$$\sum_{i,j}|g_{ij}(v) - \widehat{g}_{ij}(v)| + \sum_i |p_i(v) - \widehat{p}_i(v)| \leqslant \epsilon.$$

This also implies that $\sum_i |p_i(v) - \widehat{p}_i(v)| \leqslant \epsilon$, and shows the existence of a cover for $\mathcal{P}$ of size at most $\mathcal{N}_\infty(\mathcal{M},\epsilon)$.

Substituting the bound on the Radamacher complexity term in (A.9) in (A.8) and using the fact that $\mathcal{N}_\infty(\mathcal{P},\epsilon) \leqslant \mathcal{N}_\infty(\mathcal{M},\epsilon)$, we get:

$$\mathbf{E}_{v\sim F}\Big[-\sum_{i\in N} p_i(v)\Big] \leqslant -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{i=1}^{n} p_i(v^{(\ell)}) + 2\cdot\inf_{\epsilon>0}\left\{\epsilon + 2n\sqrt{\frac{2\log(\mathcal{N}_\infty(\mathcal{M},\,\epsilon))}{L}}\right\} + Cn\sqrt{\frac{\log(1/\delta)}{L}},$$

which completes the proof.

Generalization Bound for Regret

We move to the second part, namely a generalization bound for regret, which is the more challenging part of the proof. We first define the class of sum regret functions:

$$\overline{\text{rgt}} \circ \mathcal{U} = \left\{ f : V \to \mathbb{R} \,\middle|\, f(v) = \sum_{i=1}^{n} r_i(v) \text{ for some } (r_1, \ldots, r_n) \in \text{rgt} \circ \mathcal{U} \right\}.$$

The proof then proceeds in three steps:

(1) bounding the covering number for each regret class $\text{rgt} \circ \mathcal{U}_i$ in terms of the covering number for individual utility classes $\mathcal{U}_i$

(2) bounding the covering number for the combined utility class $\mathcal{U}$ in terms of the covering number for $\mathcal{M}$

(3) bounding the covering number for the sum regret class $\overline{\text{rgt}} \circ \mathcal{U}$ in terms of the covering number for the (combined) utility class $\mathcal{M}$.

An application of Lemma A.2 then completes the proof. We prove each of the above steps below.

Step 1. $\mathcal{N}_\infty(\text{rgt} \circ \mathcal{U}_i, \epsilon) \leqslant \mathcal{N}_\infty(\mathcal{U}_i, \epsilon/2)$.

Proof. By the definition of covering number $\mathcal{N}_\infty(\mathcal{U}_i, \epsilon)$, there exists a cover $\widehat{\mathcal{U}}_i$ with size at most $\mathcal{N}_\infty(\mathcal{U}_i, \epsilon/2)$ such that for any $u_i \in \mathcal{U}_i$, there is a $\widehat{u}_i \in \widehat{\mathcal{U}}_i$ with

$$\sup_{v, v_i'} |u_i(v_i, (v_i', v_{-i})) - \widehat{u}_i(v_i, (v_i', v_{-i}))| \leqslant \epsilon/2.$$

For any $u_i \in \mathcal{U}_i$, taking $\widehat{u}_i \in \widehat{\mathcal{U}}_i$ satisfying the above condition, then for any v,

$$\begin{aligned}
&\left| \max_{v_i' \in V} \big(u_i(v_i, (v_i', v_{-i})) - u_i(v_i, (v_i, v_{-i}))\big) - \max_{\bar{v}_i \in V} \big(\widehat{u}_i(v_i, (\bar{v}_i, v_{-i})) - \widehat{u}_i(v_i, (v_i, v_{-i}))\big) \right| \\
\leqslant &\left| \max_{v_i'} u_i(v_i, (v_i', v_{-i})) - \max_{\bar{v}_i} \widehat{u}_i(v_i, (\bar{v}_i, v_{-i})) + \widehat{u}_i(v_i, (v_i, v_{-i})) - u_i(v_i, (v_i, v_{-i})) \right| \\
\leqslant &\left| \max_{v_i'} u_i(v_i, (v_i', v_{-i})) - \max_{\bar{v}_i} \widehat{u}_i(v_i, (\bar{v}_i, v_{-i})) \right| + \left| \widehat{u}_i(v_i, (v_i, v_{-i})) - u_i(v_i, (v_i, v_{-i})) \right| \\
\leqslant &\left| \max_{v_i'} u_i(v_i, (v_i', v_{-i})) - \max_{\bar{v}_i} \widehat{u}_i(v_i, (\bar{v}_i, v_{-i})) \right| + \epsilon/2
\end{aligned}$$

Let $v_i^* \in \arg\max_{v_i'} u_i(v_i, (v_i', v_{-i}))$ and $\widehat{v}_i^* \in \arg\max_{\bar{v}_i} \widehat{u}_i(v_i, (\bar{v}_i, v_{-i}))$, then

$$\max_{v_i'} u_i(v_i, (v_i', v_{-i})) = u_i(v_i^*, v_{-i}) \leqslant \widehat{u}_i(v_i^*, v_{-i}) + \epsilon/2 \leqslant \widehat{u}_i(\widehat{v}_i^*, v_{-i}) + \epsilon/2 = \max_{\bar{v}_i} \widehat{u}_i(v_i, (\bar{v}_i, v_{-i})) + \epsilon,$$

$$\max_{\bar{v}_i} \widehat{u}_i(v_i, (\bar{v}_i, v_{-i})) = \widehat{u}_i(\widehat{v}_i^*, v_{-i}) \leqslant u_i(\widehat{v}_i^*, v_{-i}) + \epsilon/2 \leqslant u_i(v_i^*, v_{-i}) + \epsilon/2 = \max_{v_i'} u_i(v_i, (v_i', v_{-i})) + \epsilon/2\,.$$

Thus, for all $u_i \in \mathcal{U}_i$, there exists $\widehat{u}_i \in \widehat{\mathcal{U}}_i$ such that for any valuation profile v,

$$\left|\max_{v_i'} \left(u_i(v_i, (v_i', v_{-i})) - u_i(v_i, (v_i, v_{-i}))\right) - \max_{\bar{v}_i} \left(\widehat{u}_i(v_i, (\bar{v}_i, v_{-i})) - \widehat{u}_i(v_i, (v_i, v_{-i}))\right)\right| \leqslant \epsilon,$$

which implies $\mathcal{N}_\infty(\mathrm{rgt} \circ \mathcal{U}_i, \epsilon) \leqslant \mathcal{N}_\infty(\mathcal{U}_i, \epsilon/2)$.

This completes the proof of Step 1. □

Step 2. *For all $i \in N$, $\mathcal{N}_\infty(\mathcal{U}, \epsilon) \leqslant \mathcal{N}_\infty(\mathcal{M}, \epsilon)$.*

Proof. Recall that the utility function of bidder i is $u_i(v_i, (v_i', v_{-i})) = \langle v_i, g_i(v_i', v_{-i})\rangle - p_i(v_i', v_{-i})$. There exists a set $\widehat{\mathcal{M}}$ with $|\widehat{\mathcal{M}}| \leqslant \mathcal{N}_\infty(\mathcal{M}, \epsilon)$ such that, there exists $(\widehat{g}, \widehat{p}) \in \widehat{M}$ such that

$$\sup_{v \in V} \sum_{i,j} |g_{ij}(v) - \widehat{g}_{ij}(v)| + \|p(v) - \widehat{p}(v)\|_1 \leqslant \epsilon.$$

We denote $\widehat{u}_i(v_i, (v_i', v_{-i})) = \langle v_i, \widehat{g}_i(v_i', v_{-i})\rangle - \widehat{p}_i(v_i', v_{-i})$, where we treat v_i as a real-valued vector of length 2^M.

For all $v \in V, v_i' \in V_i$,

$$\begin{aligned}
&\left|u_i(v_i, (v_i', v_{-i})) - \widehat{u}_i(v_i, (v_i', v_{-i}))\right| \\
\leqslant\;& \left|\langle v_i, g_i(v_i', v_{-i})\rangle - \langle v_i, \widehat{g}_i(v_i', v_{-i})\rangle\right| + \left|p_i(v_i', v_{-i}) - \widehat{p}_i(v_i', v_{-i})\right| \\
\leqslant\;& \|v_i\|_\infty \cdot \|g_i(v_i', v_{-i}) - \widehat{g}_i(v_i', v_{-i})\|_1 + \left|p_i(v_i', v_{-i}) - \widehat{p}_i(v_i', v_{-i})\right| \\
\leqslant\;& \sum_j |g_{ij}(v_i', v_{-i}) - \widehat{g}_{ij}(v_i', v_{-i})| + \left|p_i(v_i', v_{-i}) - \widehat{p}_i(v_i', v_{-i})\right|
\end{aligned}$$

Therefore, for any $u \in \mathcal{U}$, take $\widehat{u} = (\widehat{g}, \widehat{p}) \in \widehat{\mathcal{M}}$, for all v, v',

$$\begin{aligned}
&\sum_i |u_i(v_i, (v_i', v_{-i})) - \widehat{u}_i(v_i, (v_i', v_{-i}))| \\
&\leqslant \sum_{ij} |g_{ij}(v_i', v_{-i}) - \widehat{g}_{ij}(v_i', v_{-i})| + \sum_i \left|p_i(v_i', v_{-i}) - \widehat{p}_i(v_i', v_{-i})\right|
\end{aligned}$$

$$\leqslant \epsilon$$

This completes the proof of Step 2. □

Step 3. $\mathcal{N}_\infty(\overline{\text{rgt}} \circ \mathcal{U}, \epsilon) \leqslant \mathcal{N}_\infty(\mathcal{M}, \epsilon/2)$.

Proof. By definition of $\mathcal{N}_\infty(\mathcal{U}, \epsilon)$, there exists $\widehat{\mathcal{U}}$ with size at most $\mathcal{N}_\infty(\mathcal{U}, \epsilon)$, such that, for any $u \in \mathcal{U}$, there exists $\hat{u}$ such that for all $v, v' \in V$,

$$\sum_i |u_i(v_i, (v_i', v_{-i})) - \hat{u}_i(v_i, (v_i', v_{-i}))| \leqslant \epsilon.$$

Therefore for all $v \in V$, $|\sum_i u_i(v_i, (v_i', v_{-i})) - \sum_i \hat{u}_i(v_i, (v_i', v_{-i}))| \leqslant \epsilon$, from which it follows that $\mathcal{N}_\infty(\overline{\text{rgt}} \circ \mathcal{U}, \epsilon) \leqslant \mathcal{N}_\infty(\text{rgt} \circ \mathcal{U}, \epsilon)$. Following Step 1, it is easy to show $\mathcal{N}_\infty(\text{rgt} \circ \mathcal{U}, \epsilon) \leqslant \mathcal{N}_\infty(\mathcal{U}, \epsilon/2)$.

Together with Step 2 this completes the proof of Step 3. □

Based on the same arguments as in Section A.3.2, we can thus bound the empirical Rademacher complexity as:

$$\begin{aligned}
\hat{\mathcal{R}}_L(\overline{\text{rgt}} \circ \mathcal{U}) &\leqslant \inf_{\epsilon > 0} \left(\epsilon + 2n \sqrt{\frac{2 \log \mathcal{N}_\infty(\overline{\text{rgt}} \circ \mathcal{U}, \epsilon)}{L}} \right) \\
&\leqslant \inf_{\epsilon > 0} \left(\epsilon + 2n \sqrt{\frac{2 \log \mathcal{N}_\infty(\mathcal{M}, \epsilon/2)}{L}} \right).
\end{aligned}$$

Applying Lemma A.2, completes the proof of the generalization bound for regret. □

A.3.3 Proof of Theorem 1.3

The convexity of $u^{\alpha,\beta}$ follows from the fact it is a "max" of linear functions. We now show that $u^{\alpha,\beta}$ is monotonically non-decreasing. Let $h_j(v) = w_j \cdot v + \beta_j$. Since w_j is non-negative in all entries, for any $v_i \leqslant v_i', \forall i \in M$, we have $h_j(v) \leqslant h_j(v')$. Then

$$u^{\alpha,\beta}(v) \;=\; \max_{j \in [J]} h_j(v) \;=\; h_{j_*}(v) \;\leqslant\; h_{j_*}(v') \;\leqslant\; \max_{j \in [J]} h_j(v') \;=\; u^{\alpha,\beta}(v'),$$

where $j_* \in \operatorname{argmin}_{j\in[J]} h_j(v)$. It remains to be shown that $u^{\alpha,\beta}$ is 1-Lipschitz. For any $v, v' \in \mathbb{R}^m_{\geqslant 0}$,

$$
\begin{aligned}
|u^{\alpha,\beta}(v) - u^{\alpha,\beta}(v')| &= |\max_{j\in[J]} h_j(v) - \max_{j\in[J]} h_j(v')| \\
&\leqslant \max_{j\in[J]} |h_j(v') - h_j(v)| \\
&= \max_{j\in[J]} |w_j \cdot (v' - v)| \\
&\leqslant \max_{j\in[J]} \|w_j\|_\infty \, |v' - v|_1 \\
&\leqslant |v'_k - v_k|_1,
\end{aligned}
$$

where the last inequality holds because each component $\alpha_{jk} = \sigma(\alpha_{jk}) \leqslant 1$. □

A.3.4 Proof of Lemma 1.2

First, given the property of the softmax function and the min operation, $\varphi^{DS}(s, s')$ ensures that the row sums and column sums for the resulting allocation matrix do not exceed 1. In fact, for any doubly stochastic allocation z, there exists scores s and s', for which the min of normalized scores recovers z (e.g. $s_{ij} = s'_{ij} = log(z_{ij}) + c$ for any $c \in \mathbb{R}$). □

A.3.5 Proof of Lemma A.1

Similar to Lemma 1.2, $\varphi^{CF}(s, s^{(1)}, \ldots, s^{(m)})$ trivially satisfies the combinatorial feasibility (constraints (A.5)–(A.6)). For any allocation z that satisfies the combinatorial feasibility, the following scores

$$\forall j = 1, \cdots, m, \quad s_{i,S} = s^{(j)}_{i,S} = \log(z_{i,S}) + c,$$

makes $\varphi^{CF}(s, s^{(1)}, \ldots, s^{(m)})$ recover z. □

A.3.6 Proof of Theorem 1.4

In Theorem 1.4, we only show the bounds on Δ_L for RegretNet with additive and unit-demand bidders. We restate this theorem so that it also bounds Δ_L for the general combinatorial valuations setting (with combinatorial feasible allocation). Recall that the ℓ_1 norm for a vector x is denoted by $\|x\|_1$ and the induced ℓ_1 norm for a matrix $A \in \mathbb{R}^{k\times t}$ is denoted by $\|A\|_1 = \max_{1\leqslant j\leqslant t}\sum_{i=1}^{k} A_{ij}$.

Theorem A.4. *For RegretNet with R hidden layers, K nodes per hidden layer, d_g parameters in the allocation network, d_p parameters in the payment network, and the vector of all model parameters $\|w\|_1 \leqslant W$, the following are the bounds on the term Δ_L for different bidder valuation types:*

(a) additive valuations:

$\Delta_L \leqslant O\big(\sqrt{R(d_g+d_p)\log(LW\max\{K, mn\})/L}\big)$,

(b) unit-demand valuations:

$\Delta_L \leqslant O\big(\sqrt{R(d_g+d_p)\log(LW\max\{K, mn\})/L}\big)$,

(c) combinatorial valuations (with combinatorial feasible allocation):

$\Delta_L \leqslant O\big(\sqrt{R(d_g+d_p)\log(LW\max\{K, n\,2^m\})/L}\big)$.

We first bound the covering number for a general feed-forward neural network and specialize it to the three architectures we present in Section 1.3 and Appendix A.1.2.

Lemma A.3. *Let $\mathcal{F}_k$ be a class of feed-forward neural networks that maps an input vector $x \in \mathbb{R}^{d_0}$ to an output vector $y \in \mathbb{R}^{d_k}$, with each layer ℓ containing T_ℓ nodes and computing $z \mapsto \phi_\ell(w^\ell z)$, where each $w^\ell \in \mathbb{R}^{T_\ell\times T_{\ell-1}}$ and $\phi_\ell : \mathbb{R}^{T_\ell} \to [-B, +B]^{T_\ell}$. Further let, for each network in $\mathcal{F}_k$, let the parameter matrices $\|w^\ell\|_1 \leqslant W$ and $\|\phi_\ell(s) - \phi_\ell(s')\|_1 \leqslant \Phi\|s - s'\|_1$ for any $s, s' \in \mathbb{R}^{T_{\ell-1}}$.*

$$\mathcal{N}_\infty(\mathcal{F}_k, \epsilon) \leqslant \left\lceil \frac{2Bd^2W(2\Phi W)^k}{\epsilon} \right\rceil^d,$$

where $T = \max_{\ell\in[k]} T_\ell$ and d is the total number of parameters in a network.

Proof. We shall construct an $\ell_{1,\infty}$ cover for $\mathcal{F}_k$ by discretizing each of the d parameters along $[-W, +W]$ at scale ϵ_0/d, where we will choose $\epsilon_0 > 0$ at the end of the proof. We will use $\widehat{\mathcal{F}}_k$ to denote the subset of neural networks in $\mathcal{F}_k$ whose parameters are in the range $\{-(\lceil Wd/\epsilon_0\rceil - 1)\,\epsilon_0/d, \ldots, -\epsilon_0/d, 0, \epsilon_0/d, \ldots, \lceil Wd/\epsilon_0\rceil\epsilon_0/d\}$. The size of $\widehat{\mathcal{F}}_k$ is at most $\lceil 2dW/\epsilon_0\rceil^d$. We shall now show that $\widehat{\mathcal{F}}_k$ is an ϵ-cover for $\mathcal{F}_k$.

We use mathematical induction on the number of layers k. We wish to show that for any

$f \in \mathcal{F}_k$ there exists a $\widehat{f} \in \widehat{\mathcal{F}}_k$ such that:

$$\|f(x) - \widehat{f}(x)\|_1 \leqslant Bd\epsilon_0(2\Phi W)^k.$$

For $k = 0$, the statement holds trivially. Assume that the statement is true for $\mathcal{F}_k$. We now show that the statement holds for $\mathcal{F}_{k+1}$.

A function $f \in \mathcal{F}_{k+1}$ can be written as $f(z) = \phi_{k+1}(w_{k+1}H(z))$ for some $H \in \mathcal{F}_k$. Similarly, a function $\widehat{f} \in \widehat{\mathcal{F}}_{k+1}$ can be written as $\widehat{f}(z) = \phi_{k+1}(\widehat{w}_{k+1}\widehat{H}(z))$ for some $\widehat{H} \in \widehat{\mathcal{F}}_k$ and $\widehat{w}_{k+1}$ is a matrix of entries in $\{-(\lceil Wd/\epsilon_0\rceil - 1)\,\epsilon_0/d, \ldots, -\epsilon_0/d, 0, \epsilon_0/d, \ldots, \lceil Wd/\epsilon_0\rceil\epsilon_0/d\}$. Also, for any parameter matrix $w^\ell \in \mathbb{R}^{T_\ell \times T_{\ell-1}}$, there is a matrix $\widehat{w}^\ell$ with discrete entries s.t.

$$\|w_\ell - \widehat{w}_\ell\|_1 = \max_{1\leqslant j\leqslant T_{\ell-1}} \sum_{i=1}^{T_\ell} |w^\ell_{\ell,i,j} - \widehat{w}_{\ell,i,j}| \leqslant T_\ell\epsilon_0/d \leqslant \epsilon_0. \tag{A.10}$$

We then have:

$$\begin{aligned}
&\|f(x) - \widehat{f}(x)\|_1 \\
&\quad = \|\phi_{k+1}(w_{k+1}H(x)) - \phi_{k+1}(\widehat{w}_{k+1}\widehat{H}(x))\|_1 \\
&\quad \leqslant \Phi\|w_{k+1}H(x) - \widehat{w}_{k+1}\widehat{H}(x)\|_1 \\
&\quad \leqslant \Phi\|w_{k+1}H(x) - w_{k+1}\widehat{H}(x)\|_1 + \Phi\|w_{k+1}\widehat{H}(x) - \widehat{w}_{k+1}\widehat{H}(x)\|_1 \\
&\quad \leqslant \Phi\|w_{k+1}\|_1 \cdot \|H(x) - \widehat{H}(x)\|_1 + \Phi\|w_{k+1} - \widehat{w}_{k+1}\|_1 \cdot \|\widehat{H}(x)\|_1 \\
&\quad \leqslant \Phi W\|H(x) - \widehat{H}(x)\|_1 + \Phi T_k B\|w_{k+1} - \widehat{w}_{k+1}\|_1 \\
&\quad \leqslant Bd\epsilon_0\Phi W(2\Phi W)^k + \Phi Bd\epsilon_0 \\
&\quad \leqslant Bd\epsilon_0(2\Phi W)^{k+1},
\end{aligned}$$

where the second line follows from our assumption on ϕ_{k+1}, and the sixth line follows from our inductive hypothesis and from (A.10). By choosing $\epsilon_0 = \frac{\epsilon}{B(2\Phi W)^k}$, we complete the proof. □

We next bound the covering number of the auction class in terms of the covering number for the class of allocation networks and for the class of payment networks. Recall that the payment networks computes a fraction $\alpha : \mathbb{R}^{m(n+1)} \to [0,1]^n$ and computes a payment $p_i(b) = \alpha_i(b) \cdot \langle v_i, g_i(b)\rangle$ for each bidder i. Let $\mathcal{G}$ be the class of allocation networks and $\mathcal{A}$ be the class of fractional payment functions used to construct auctions in $\mathcal{M}$. Let $\mathcal{N}_\infty(\mathcal{G}, \epsilon)$ and $\mathcal{N}_\infty(\mathcal{A}, \epsilon)$ be the corresponding covering numbers w.r.t. the ℓ_∞ norm. Then:

Lemma A.4. $\mathcal{N}_\infty(\mathcal{M}, \epsilon) \leqslant \mathcal{N}_\infty(\mathcal{G}, \epsilon/3) \cdot \mathcal{N}_\infty(\mathcal{A}, \epsilon/3)$

Proof. Let $\widehat{\mathcal{G}} \subseteq \mathcal{G}$, $\widehat{\mathcal{A}} \subseteq \mathcal{A}$ be ℓ_∞ covers for $\mathcal{G}$ and $\mathcal{A}$, i.e. for any $g \in \mathcal{G}$ and $\alpha \in \mathcal{A}$, there exists $\widehat{g} \in \widehat{\mathcal{G}}$ and $\widehat{\alpha} \in \widehat{\mathcal{A}}$ with

$$\sup_b \sum_{i,j} |g_{ij}(b) - \widehat{g}_{ij}(b)| \leqslant \epsilon/3 \tag{A.11}$$

$$\sup_b \sum_{i} |\alpha_i(b) - \widehat{\alpha}_i(b)| \leqslant \epsilon/3. \tag{A.12}$$

We now show that the class of mechanism $\widehat{\mathcal{M}} = \{(\widehat{g}, \widehat{\alpha}) \mid \widehat{g} \in \widehat{\mathcal{G}}, \text{ and } \widehat{p}(b) = \widehat{\alpha}_i(b) \cdot \langle v_i, \widehat{g}_i(b) \rangle\}$ is an ϵ-cover for $\mathcal{M}$ under the $\ell_{1,\infty}$ distance. For any mechanism in $(g, p) \in \mathcal{M}$, let $(\widehat{g}, \widehat{p}) \in \widehat{\mathcal{M}}$ be a mechanism in $\widehat{\mathcal{M}}$ that satisfies (A.12). We have:

$$\begin{aligned}
&\sum_{i,j} |g_{ij}(b) - \widehat{g}_{ij}(b)| + \sum_i |p_i(b) - \widehat{p}_i(b)| \\
&\quad \leqslant \epsilon/3 + \sum_i |\alpha_i(b) \cdot \langle b_i, g_{i,\cdot}(b) \rangle - \widehat{\alpha}_i(b) \cdot \langle b_i, \widehat{g}_i(b) \rangle| \\
&\quad \leqslant \epsilon/3 + \sum_i \Big(|(\alpha_i(b) - \widehat{\alpha}_i(b)) \cdot \langle b_i, g_i(b) \rangle| \\
&\qquad\qquad + |\widehat{\alpha}_i(b) \cdot (\langle b_i, g_i(b) \rangle - \langle b_i, \widehat{g}_{i,\cdot}(b) \rangle)| \Big) \\
&\quad \leqslant \epsilon/3 + \sum_i |\alpha_i(b) - \widehat{\alpha}_i(b)| + \sum_i \|b_i\|_\infty \cdot \|g_i(b) - \widehat{g}_i(b)\|_1 \\
&\quad \leqslant 2\epsilon/3 + \sum_{i,j} |g_{ij}(b) - \widehat{g}_{ij}(b)| \leqslant \epsilon,
\end{aligned}$$

where in the third inequality we use $\langle b_i, g_i(b) \rangle \leqslant 1$. The size of the cover $\widehat{\mathcal{M}}$ is $|\widehat{\mathcal{G}}||\widehat{\mathcal{A}}|$, which completes the proof. □

We are now ready to prove covering number bounds for the three architectures in Section 1.3 and Appendix A.1.2.

Proof of Theorem A.4. All three architectures use the same feed-forward architecture for computing fractional payments, consisting of K hidden layers with tanh activation functions. We also have by our assumption that the ℓ_1 norm of the vector of all model parameters is at most W, for each $\ell = 1, \ldots, R+1$, $\|w_\ell\|_1 \leqslant W$. Using that fact that the tanh activation functions are 1-Lipschitz and bounded in $[-1,1]$, and there are at most $\max\{K, n\}$ number of nodes in any layer of the payment network, we have by an application of Lemma A.3 the following bound on the covering number of the fractional payment networks $\mathcal{A}$ used in each case:

$$\mathcal{N}_\infty(\mathcal{A}, \epsilon) \leqslant \left\lceil \frac{\max(K,n)^2 (2W)^{R+1}}{\epsilon} \right\rceil^{d_p}$$

where d_p is the number of parameters in payment networks.

For the covering number of allocation networks $\mathcal{G}$, we consider each architecture separately. In each case, we bound the Lipschitz constant for the activation functions used in the layers of the allocation network and followed by an application of Lemma A.3. For ease of exposition, we omit the dummy scores used in the final layer of neural network architectures.

Additive bidders. The output layer computes n allocation probabilities for each item j using a softmax function. The activation function $\phi_{R+1} : \mathbb{R}^n \to \mathbb{R}^n$ for the final layer for input $s \in \mathbb{R}^{n\times m}$ can be described as: $\phi_{R+1}(s) = [\text{softmax}(s_{1,1}, \ldots, s_{n,1}), \ldots, \text{softmax}(s_{1,m}, \ldots, s_{n,m})]$, where softmax : $\mathbb{R}^n \to [0,1]^n$ is defined for any $u \in \mathbb{R}^n$ as $\text{softmax}_i(u) = e^{u_i} / \sum_{k=1}^n e^{u_k}$.

We then have for any $s, s' \in \mathbb{R}^{n\times m}$,

$$\begin{aligned}
&\|\phi_{R+1}(s) - \phi_{R+1}(s')\|_1 \\
&\quad \leqslant \sum_j \left\| \text{softmax}(s_{1,j}, \ldots, s_{n,j}) - \text{softmax}(s'_{1,j}, \ldots, s'_{n,j}) \right\|_1 \\
&\quad \leqslant \sqrt{n} \sum_j \left\| \text{softmax}(s_{1,j}, \ldots, s_{n,j}) - \text{softmax}(s'_{1,j}, \ldots, s'_{n,j}) \right\|_2 \\
&\quad \leqslant \sqrt{n} \frac{\sqrt{n-1}}{n} \sum_j \sqrt{\sum_i \|s_{ij} - s'_{ij}\|^2} \\
&\quad \leqslant \sum_j \sum_i |s_{ij} - s'_{ij}|
\end{aligned} \tag{A.13}$$

where the third step follows by bounding the Frobenius norm of the Jacobian of the softmax function.

The hidden layers $\ell = 1, \ldots, R$ are standard feed-forward layers with tanh activations. Since the tanh activation function is 1-Lipschitz, $\|\phi_\ell(s) - \phi_\ell(s')\|_1 \leqslant \|s - s'\|_1$. We also have by our assumption that the ℓ_1 norm of the vector of all model parameters is at most W, for each $\ell = 1, \ldots, R+1$, $\|w_\ell\|_1 \leqslant W$. Moreover, the output of each hidden layer node is in $[-1, 1]$, the output layer nodes is in $[0, 1]$, and the maximum number of nodes in any layer (including the output layer) is at most $\max\{K, mn\}$.

By an application of Lemma A.3 with $\Phi = 1$, $B = 1$, and $d = \max\{K, mn\}$ we have

$$\mathcal{N}_\infty(\mathcal{G}, \epsilon) \leqslant \left\lceil \frac{\max\{K, mn\}^2 (2W)^{R+1}}{\epsilon} \right\rceil^{d_g},$$

where d_g is the number of parameters in allocation networks.

Unit-demand bidders. The output layer n allocation probabilities for each item j as an elemmaent-wise minimum of two softmax functions. The activation function $\phi_{R+1} : \mathbb{R}^2 n \to \mathbb{R}^n$ for the final layer for two sets of scores $s, \bar{s} \in \mathbb{R}^{n \times m}$ can be described as:

$$\phi_{R+1,i,j}(s, s') = \min\{\text{softmax}_j(s_{i,1}, \ldots, s_{i,m}), \text{softmax}_i(s'_{1,j}, \ldots, s'_{n,j})\}.$$

We then have for any $s, \tilde{s}, s', \tilde{s}' \in \mathbb{R}^{n \times m}$,

$$\begin{aligned}
&\|\phi_{R+1}(s, \tilde{s}) - \phi_{R+1}(s', \tilde{s}')\|_1 \\
&\quad \leqslant \sum_{i,j} \Big| \min\{\text{softmax}_j(s_{i,1}, \ldots, s_{i,m}), \text{softmax}_i(\tilde{s}_{1,j}, \ldots, \tilde{s}_{n,j})\} \\
&\qquad\qquad - \min\{\text{softmax}_j(s'_{i,1}, \ldots, s'_{i,m}), \text{softmax}_i(\tilde{s}'_{1,j}, \ldots, \tilde{s}'_{n,j})\} \Big| \\
&\quad \leqslant \sum_{i,j} \Big| \max\{\text{softmax}_j(s_{i,1}, \ldots, s_{i,m}) - \text{softmax}_j(s'_{i,1}, \ldots, s'_{i,m}), \\
&\qquad\qquad\qquad \text{softmax}_i(\tilde{s}_{1,j}, \ldots, \tilde{s}_{n,j}) - \text{softmax}_i(\tilde{s}'_{1,j}, \ldots, \tilde{s}'_{n,j})\} \Big| \\
&\quad \leqslant \sum_{i} \big\|\text{softmax}(s_{i,1}, \ldots, s_{i,m}) - \text{softmax}(s'_{i,1}, \ldots, s'_{i,m})\big\|_1 \\
&\qquad\qquad + \sum_{j} \big\|\text{softmax}(\tilde{s}_{1,j}, \ldots, \tilde{s}_{n,j}) - \text{softmax}(\tilde{s}'_{1,j}, \ldots, \tilde{s}'_{n,j})\}\big\|_1 \\
&\quad \leqslant \sum_{i,j} |s_{ij} - s'_{ij}| + \sum_{i,j} |\tilde{s}_{ij} - \tilde{s}'_{ij}|,
\end{aligned}$$

where the last step can be derived in the same way as (A.13).

As with additive bidders, using additionally hidden layers $\ell = 1, \ldots, R$ are standard feed-forward layers with tanh activations, we have from Lemma A.3 with $\Phi = 1$, $B = 1$ and

$d = \max\{K, mn\}$,

$$\mathcal{N}_\infty(\mathcal{G}, \epsilon) \leqslant \left\lceil \frac{\max\{K, mn\}^2 (2W)^{R+1}}{\epsilon} \right\rceil^{d_g}.$$

Combinatorial bidders. The output layer outputs an allocation probability for each bidder i and bundle of items $S \subseteq M$. The activation function $\phi_{R+1} : \mathbb{R}^{(m+1)n2^m} \to \mathbb{R}^{n2^m}$ for this layer for $m+1$ sets of scores $s, s^{(1)}, \ldots, s^{(m)} \in \mathbb{R}^{n \times 2^m}$ is given by:

$$\phi_{R+1,i,S}(s, s^{(1)}, \ldots, s^{(m)}) = \min\Big\{\mathrm{softmax}_S(s_{i,S'} : S' \subseteq M), \mathrm{softmax}_S(s^{(1)}_{i,S'} : S' \subseteq M), \ldots,$$
$$\mathrm{softmax}_S(s^{(m)}_{i,S'} : S' \subseteq M)\Big\},$$

where $\mathrm{softmax}_S(a_{S'} : S' \subseteq M) = e^{a_S} / \sum_{S' \subseteq M} e^{a_{S'}}$.

We then have for any $s, s^{(1)}, \ldots, s^{(m)}, s', s'^{(1)}, \ldots, s'^{(m)} \in \mathbb{R}^{n \times 2^m}$,

$$\begin{aligned}
&\|\phi_{R+1}(s, s^{(1)}, \ldots, s^{(m)}) - \phi_{R+1}(s', s'^{(1)}, \ldots, s'^{(m)})\|_1 \\
&\leqslant \sum_{i,S} \Big| \min\Big\{\mathrm{softmax}_S(s_{i,S'} : S' \subseteq M), \\
&\qquad \mathrm{softmax}_S(s^{(1)}_{i,S'} : S' \subseteq M), \ldots, \mathrm{softmax}_S(s^{(m)}_{i,S'} : S' \subseteq M)\Big\} \\
&\qquad - \min\Big\{\mathrm{softmax}_S(s'_{i,S'} : S' \subseteq M), \\
&\qquad \mathrm{softmax}_S(s'^{(1)}_{i,S'} : S' \subseteq M), \ldots, \mathrm{softmax}_S(s'^{(m)}_{i,S'} : S' \subseteq M)\Big\}\Big| \\
&\leqslant \sum_{i,S} \max\Big\{\big|\mathrm{softmax}_S(s_{i,S'} : S' \subseteq M) - \mathrm{softmax}_S(s'_{i,S'} : S' \subseteq M)\big|, \\
&\qquad \big|\mathrm{softmax}_S(s^{(1)}_{i,S'} : S' \subseteq M) - \mathrm{softmax}_S(s'^{(1)}_{i,S'} : S' \subseteq M)\big|, \ldots \\
&\qquad \big|\mathrm{softmax}_S(s^{(m)}_{i,S'} : S' \subseteq M) - \mathrm{softmax}_S(s'^{(m)}_{i,S'} : S' \subseteq M)\big|\Big\} \\
&\leqslant \sum_{i} \big\|\mathrm{softmax}(s_{i,S'} : S' \subseteq M) - \mathrm{softmax}(s'_{i,S'} : S' \subseteq M)\big\|_1 \\
&\qquad + \sum_{i,j} \big\|\mathrm{softmax}(s^{(j)}_{i,S'} : S' \subseteq M) - \mathrm{softmax}(s'^{(j)}_{i,S'} : S' \subseteq M)\big\|_1 \\
&\leqslant \sum_{i,S} |s_{i,S} - s'_{i,S}| + \sum_{i,j,S} |s^{(j)}_{i,S} - s'^{(j)}_{i,S}|,
\end{aligned}$$

where the last step can be derived in the same way as (A.13).

As with additive bidders, using additionally hidden layers $\ell = 1, \ldots, R$ are standard feed-forward layers with tanh activations, we have from Lemma A.3 with $\Phi = 1$, $B = 1$ and

$d = \max\{K, n \cdot 2^m\}$

$$\mathcal{N}_\infty(\mathcal{G}, \epsilon) \leqslant \left\lceil \frac{\max\{K, n \cdot 2^m\}^2 (2W)^{R+1}}{\epsilon} \right\rceil^{d_g}$$

where d_g is the number of parameters in allocation networks. □

We now bound Δ_L for the three architectures using the covering number bounds we derived above. In particular, we upper bound the the 'inf' over $\epsilon > 0$ by substituting a specific value of ϵ:

(a) For additive bidders, choosing $\epsilon = \frac{1}{\sqrt{L}}$, we get

$$\Delta_L \leqslant O\left(\sqrt{R(d_p + d_g)\frac{\log(W \max\{K, mn\} L)}{L}}\right)$$

(b) For unit-demand bidders, choosing $\epsilon = \frac{1}{\sqrt{L}}$, we get

$$\Delta_L \leqslant O\left(\sqrt{R(d_p + d_g)\frac{\log((W \max\{K, mn\} L)}{L}}\right)$$

(c) For combinatorial bidders, choosing $\epsilon = \frac{1}{\sqrt{L}}$, we get

$$\Delta_L \leqslant O\left(\sqrt{R(d_p + d_g)\frac{\log(W \max\{K, n \cdot 2^m\} L)}{L}}\right).$$

□

A.3.7 Proof of Theorem 1.5

We apply the duality theory of [DDT13] to verify the optimality of our proposed mechanism (motivated by empirical results of RochetNet). For the completeness of presentation, we provide a brief introduction of their approach here.

Let $f(v)$ be the joint valuation distribution of $v = (v_1, v_2, \cdots, v_m)$, V be the support of $f(v)$ and define the measure μ with the following density,

$$\mathbb{I}_{v=\bar{v}} + \mathbb{I}_{v \in \partial V} \cdot f(v)(v \cdot \hat{n}(v)) - (\nabla f(v) \cdot v + (m+1) f(v)) \qquad \text{(A.14)}$$

where $\bar{v}$ is the "base valuation", i.e. $u(\bar{v}) = 0$, ∂V denotes the boundary of V, $\hat{n}(v)$ is the outer unit normal vector at point $v \in \partial V$, and m is the number of items. Let $\Gamma_+(X)$ be the unsigned

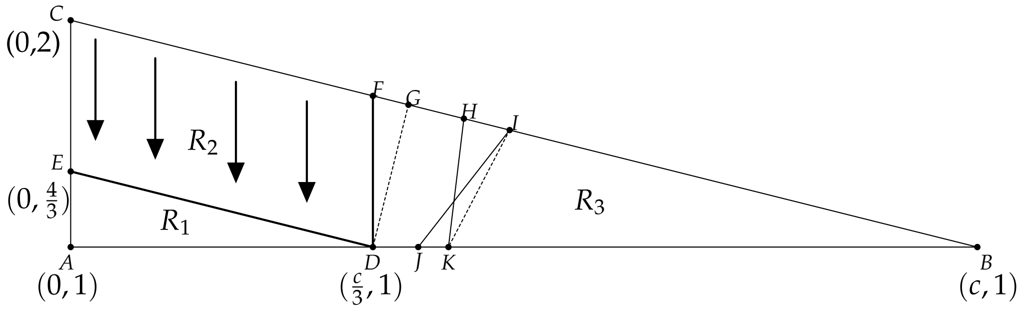


Figure A.7: *The transport of transformed measure of each region in Setting* C.

.

(Radon) measures on X. Consider an unsinged measure $\gamma \in \Gamma_+(X \times X)$, let γ_1 and γ_2 be the two marginal measures of γ, i.e. $\gamma_1(A) = \gamma(A \times X)$ and $\gamma_2(A) = \gamma(X \times A)$ for all measurable sets $A \subseteq X$. We say measure α dominates β if and only if for all (non-decreasing, convex) functions u, $\int u \, d\alpha \geqslant \int u \, d\beta$. Then by strong duality theory we have

$$\sup_u \int_V u \, d\mu = \inf_{\gamma \in \Gamma_+(V,V), \gamma_1 - \gamma_2 \succeq \mu} \int_{V \times V} \|v - v'\|_1 \, d\gamma, \tag{A.15}$$

and both the supremum and infimum are achieved. Based on "complemmaentary slackness" of linear programming, the optimal solution of Equation A.15 needs to satisfy the following conditions.

Corollary A.5 ([DDT17])**.** *Let u^* and γ^* be feasible for their respective problemmas in Equation A.15, then $\int u^* \, d\mu = \int \|v - v'\|_1 \, d\gamma^*$ if and only if the following two conditions hold:*

$$\int u^* \, d(\gamma_1^* - \gamma_2^*) = \int u^* \, d\mu$$
$$\int u^*(v) - u^*(v') = \|v - v'\|_1, \gamma^*\text{-almost surely.}$$

Then we prove the utility function u^* induced by the mechanism for setting C is optimal. Here we only focus on Settiong C with $c > 1$, for $c \leqslant 1$ the proof is analogous and we omit here[4]. The transformed measure μ of the valuation distribution is composed of:

[4]It is fairly similar to the proof for setting $c > 1$. If $c \leqslant 1$, there are only two regions to discuss, in which R_1 and

1. A point mass of $+1$ at $(0,1)$.

2. Mass -3 uniformly distributed throughout the triangle area (density $-\frac{6}{c}$).

3. Mass -2 uniformly distributed on lower edge of triangle (density $-\frac{2}{c}$).

4. Mass $+4$ uniformly distributed on right-upper edge of triangle (density $+\frac{4}{\sqrt{1+c^2}}$).

It is straightforward to verify that $\mu(R_1) = \mu(R_2) = \mu(R_3) = 0$. We will show there exists an optimal measure γ^* for the dual program of Theorem 2 (Equation 5) in [DDT13]. γ^* can be decomposed into $\gamma^* = \gamma^{R_1} + \gamma^{R_2} + \gamma^{R_3}$ with $\gamma^{R_1} \in \Gamma_+(R_1 \times R_1), \gamma^{R_2} \in \Gamma_+(R_2 \times R_2), \gamma^{R_3} \in \Gamma_+(R_3 \times R_3)$. We will show the feasibility of γ^*, such that

$$\gamma_1^{R_1} - \gamma_2^{R_1} \geq \mu|_{R_1}; \quad \gamma_1^{R_2} - \gamma_2^{R_2} \geq \mu|_{R_2}; \quad \gamma_1^{R_3} - \gamma_2^{R_3} \geq \mu|_{R_3}. \tag{A.16}$$

Then we show the conditions of Corollary 1 in [DDT13] hold for each of the measures $\gamma^{R_1}, \gamma^{R_2}, \gamma^{R_3}$ separately, such that $\int u^* d(\gamma_1^A - \gamma_2^A) = \int_A u^* d\mu$ and $u^*(v) - u^*(v') = \|v - v'\|_1$ hold γ^A-almost surely for any $A = R_1, R_2$, and R_3. We visualize the transport of measure γ^* in Figure A.7.

Construction of γ^{R_1}. $\mu_+|_{R_1}$ is concentrated on a single point $(0,1)$ and $\mu_-|_{R_1}$ is distributed throughout a region which is coordinate-wise greater than $(0,1)$, then it is obviously to show $0 \geq \mu|_{R_1}$. We set γ^{R_1} to be zero measure, and we get $\gamma_1^{R_1} - \gamma_2^{R_1} = 0 \geq \mu|_{R_1}$. In addition, $u^*(v) = 0, \forall v \in R_1$, then the conditions in Corollary 1 in [DDT13] hold trivially.

Construction of γ^{R_2}. $\mu_+|_{R_2}$ is uniformly distributed on upper edge CF of the triangle and $\mu_-|_{R_2}$ is uniformly distributed in R_2. Since we have $\mu(R_2) = 0$, we construct γ^{R_2} by "transporting" $\mu_+|_{R_2}$ into $\mu_-|_{R_2}$ downwards, that is $\gamma_1^{R_2} = \mu_+|_{R_2}, \gamma_2^{R_2} = \mu_-|_{R_2}$. Therefore, $\int u^* d(\gamma_1^{R_2} - \gamma_2^{R_2}) = \int u^* d\mu$ holds trivially. The measure γ^{R_2} is only concentrated on the pairs (v, v') such that $v_1 = v_1', v_2 \geqslant v_2'$. Thus for such pairs $(v'v')$, we have $u^*(v) - u^*(v') = (\frac{v_1}{c} + v_2 - \frac{4}{3}) - (\frac{v_1}{c} + v_2' - \frac{4}{3}) = ||v - v'||_1$.

Construction of γ^{R_3}. It is intricate to directly construct γ^{R_3} analytically, however, we will

---

R_2 are the regions correspond to allocation $(0,0)$ and $(1,1)$, respectively. Then we show the optimal $\gamma^* = \bar{\gamma}^{R_1} + \bar{\gamma}^{R_2}$ where $\bar{\gamma}^{R_1} = 0$ for region R_1 and show γ^{R_2} only "transports" mass of measure downwards and leftwards in region R_2, which is analgous to the analysis for γ^{R_3} for setting $c > 1$.

show there the optimal measure γ^{R_3} only transports mass from $\mu_+|_{R_3}$ to $\mu_-|_{R_3}$ leftwards and downwards. Let's consider a point H on edge BF with coordinates (v_1^H, v_2^H). Define the regions $R_L^H = \{v' \in R_3 | v'_1 \leqslant v_1^H\}$ and $R_U^H = \{v' \in R_3 | v'_2 \geqslant v_2^H\}$. Let $\ell(\cdot)$ represent the length of segment, then we have $\ell(FH) < \frac{2}{3\sqrt{c^2+1}}$. Thus,

$$\begin{aligned}
\mu(R_U^H) &= \frac{4\ell(FH)}{\sqrt{c^2+1}} - \frac{6}{c} \cdot \frac{\ell^2(FH)c}{2(c^2+1)} = \frac{\ell(FH)}{\sqrt{c^2+1}} \cdot \left(4 - \frac{3\ell(FH)}{\sqrt{c^2+1}}\right) > 0 \\
\mu(R_L^H) &= \frac{4\ell(FH)}{\sqrt{c^2+1}} - \frac{2}{c} \cdot \frac{\ell(FH)c}{\sqrt{c^2+1}} - \frac{6}{c} \cdot \left(\frac{2\ell(FH)c}{3\sqrt{c^2+1}} - \frac{\ell^2(FH)c}{2(c^2+1)}\right) \\
&= \frac{\ell(FH)}{\sqrt{c^2+1}} \cdot \left(\frac{3\ell(FH)}{\sqrt{c^2+1}} - 2\right) < 0
\end{aligned}$$

Thus, there exists a unique line l_H with positive slope that intersects H and separate R_3 into two parts, R_U^H (above l_H) and R_B^H (below l_H), such that $\mu_+(R_U^H) = \mu_-(R_U^H)$. We will then show for any two points on edge BF, H and I, lines l_H and l_I will not intersect inside R_3. In Figure A.7, on the contrary, we assume $l_H = HK$ and $l_I = IJ$ intersects inside R_3. Given the definition of l_H and l_I, we have

$$\mu_+(FHKD) = \mu_-(FHKD); \quad \mu_+(FIJD) = \mu_-(FIJD)$$

Since μ_+ is only distributed along the edge BF, we have

$$\mu_+(FIKD) = \mu_+(FIJD) = \mu_-(FIJD)$$

Notice μ_- is only distributed inside R_3 and edge DB, thus $\mu_-(FIKD) > \mu_-(FIJD)$. Given the above discussion, we have

$$\begin{aligned}
\mu_+(HIK) = \mu_+(FIJD) - \mu_+(FHKD) = \mu_-(FIJD) - \mu_-(FHKD) \\
< \mu_-(FIKD) - \mu_-(FHKD) = \mu_-(HIK)
\end{aligned} \tag{A.17}$$

On the other hand, let $S(HIK)$ be the area of triangle HIK, DG be the altitude of triangle DBF w.r.t BF, and h be the altitude of triangle HJK w.r.t the base HI.

$$\begin{aligned}
\mu_-(HJK) &= \frac{6}{c} \cdot S(HIK) = \frac{6}{c} \cdot \frac{1}{2}\ell(HI)h \leqslant \frac{3}{c} \cdot \ell(HI) \cdot \ell(DG) \\
&= \frac{3}{c} \cdot \frac{2c}{3\sqrt{c^2+1}} \cdot \ell(HI) = \frac{2}{\sqrt{c^2+1}} \cdot \ell(HI)
\end{aligned}$$

$$< \frac{2}{\sqrt{c^2+1}} \cdot \ell(HI) = \mu_+(HIK),$$

which is a contradiction of Equation A.17. Thus, we show l_H and l_I doesn't intersect inside R_3. Let γ^{R_3} be the measure that transport mass from $\mu_+|_{R_3}$ to $\mu_-|_{R_3}$ through lines $\{l_H | H \in BF\}$. Then we have $\gamma_1^{R_3} = \mu_+|_{R_3}, \gamma_2^{R_3} = \mu_-|_{R_3}$, which leads to $\int u^* d(\gamma_1^{R_3} - \gamma_2^{R_3}) = \int u^* d\mu$. The measure γ^{R_3} is only concentrated on the pairs (v, v'), s.t. $v_1 \geqslant v_1'$ and $v_2 \geqslant v_2'$. Therefore, for such pairs (v, v'), we have $u^*(v) - u^*(v') = (v_1 + v_2 - \frac{c}{3} - 1) - (v_1' + v_2' - \frac{c}{3} - 1) = (v_1 - v_1') + (v_2 - v_2') = ||v - v'||_1$.

Finally, we show there must exist an optimal measure γ for the dual program of Theorem 2 in [DDT13]. $\square$

Appendix B

Appendix to Chapter 3

B.1 Details of Replica-Surrogate Mechanism

We show the detailed description of Replica-Surrogate Mechanism in Fig. B.1.

B.2 Omitted Properties of Our Transformation

We state an additional property for our transformation. For a mechanism $\mathcal{M} = (x, p)$, let $\mathcal{X}$ denote the *induced allocation space*, such that $\forall a \in \mathcal{X}$, there always exists $t \in \mathcal{T}$ to satisfy $a = x(t)$. We introduce the following preserved-allocation property.

Definition B.1 (Preserved-allocation property)**.** *Let $\mathcal{X}$ and $\mathcal{X}'$ denote the induced allocation space for $\mathcal{M}$ and $\mathcal{M}'$ respectively. Mechanism $\mathcal{M}' = (x', p')$ preserves the allocation of mechanism $\mathcal{M} = (x, p)$ if, $\forall a \in \mathcal{X}$, a must be in $\mathcal{X}'$ and $\sum_{t:x(t)=a} f(t) = \sum_{t':x'(t')=a} f(t')$.*

This is a useful property, because it states that the same distribution on allocations is achieved by $\mathcal{M}'$ as the original mechanism $\mathcal{M}$. Consider, for example, a principal running the mechanism who also incurs a cost for different outcomes. With this preserved allocation property, then not only is welfare the same (or better) and revenue loss bounded, but the expected cost of the principal is preserved by the transform. By contrast, the previous transformations [DW12; RW18; CZ17; Cai+19] cannot preserve the distribution of the allocation, even for this single agent with uniform type distribution case.

Phase 1: Surrogate Sale. For each agent i,

- Modify mechanism $\mathcal{M}$ to multiply all prices it charges by a factor of $(1-\eta)$. Let $\mathcal{M}'$ be the mechanism resulting from this modification.
- Given the reported type t_i, create $r-1$ replicas sampled i.i.d from $\mathcal{F}_i$ and r surrogates sampled i.i.d from $\mathcal{F}_i$. r is the parameter of the algorithm to be decided later.
- Construct a weighted bipartite graph between replicas (including agent i's true type t_i) and surrogates. The weight of the edge between a replica $r^{(j)}$ and a surrogate $s^{(k)}$ is the interim utility of agent i when he misreports type $s^{(k)}$ rather than the true type $r^{(j)}$ in mechanism $\mathcal{M}'$, i.e.,

$$w_i(r^{(j)},s^{(k)}) = E_{t_{-i}\in\mathcal{F}_{-i}}\left[v_i(r^{(j)},x(s^{(k)},t_{-i}))\right] - (1-\eta)\cdot\mathbf{E}_{t_{-i}\in\mathcal{F}_{-i}}\left[p_i(s^{(k)},t_{-i})\right]$$

- Let $w_i((r^{(j)},s^{(k)}))$ be the value of replica $r^{(j)}$ for being matched to surrogate $s^{(k)}$. Compute the VCG matching and prices, that is, compute the maximum weighted matching w.r.t $w_i(\cdot,\cdot)$ and the corresponding VCG payments. If a replica is unmatched in the VCG matching, match it to a random unmatched surrogate.

Phase 2: Surrogate Competition.

- Let $\vec{s}_i$ denote the surrogate chosen to represent agent i in phase 1, and let $\vec{s}$ be the entire surrogate profile. We let the surrogates $\vec{s}$ play $\mathcal{M}'$.
- If agent i's true type t_i is matched to a surrogate through VCG matching, charge agent i the VCG price that he wins the surrogate and award (allocate) agent i, $x_i(s)$ (Note $\mathcal{M}'$ also charges agent i, $(1-\eta)p_i(s)$). If agent i's true type is not matched in VCG matching and matched to a random surrogate, the agent gets nothing and pays 0.

Figure B.1: *Replica-Surrogate Matching Mechanism.*

The following corollary states that, in our transformation for single agent with a uniform type distribution, the BIC mechanism $\mathcal{M}$ (achieved by transformation) has the same distribution on allocations as the original ε-BIC/ε-EEIC mechanism, $\mathcal{M}$.

Corollary B.1 (Preserved allocation for uniform type distribution)**.** *Consider a single agent with a uniform type distribution, then for any ε-BIC/ε-EEIC mechanism $\mathcal{M}$ there exists a fully IC mechanism $\mathcal{M}'$ that preserves the allocation of $\mathcal{M}$.*

For single agent with a non-uniform type distribution, the technique used in the proof of Theorem 3.6 does not satisfy this *preserved-allocation* property, since we use "fractional rotation step" to diminish weight (regret) in the type graph, which creates some new allocations.

The following theorem shows that this is not attributed to our technique: no mechanism that satisfies the preserved-allocation property can also achieve negligible revenue loss compared with the original ε-BIC mechanism.

Theorem B.2 (Non preserved-allocation for non-uniform type distribution)**.** *There exists an ε-BIC mechanism $\mathcal{M}$ with a single agent, such that no BIC mechanism can preserve the distribution of the allocation of $\mathcal{M}$.*

Proof. There are two items A, B with a single, unit-demand agent. With probability $1/6$, the agent has value $1+\varepsilon$ for item A and value 1 for item B. With probability $2/3$, the agent has value 1 for item A and value $(1+\varepsilon)$ for item B. With probability $1/6$, the agent has value 1 for each of items A and B.

There exists an ε-BIC deterministic mechanism: ask the agent which item it prefers, and allocate the other item to the agent and charge 1. This is obviously an ε-DSIC (ε-EEIC) mechanism, since the agent can only gain an additional ε by misreporting. The allocation under truth-telling is $(1,0)$ with probability at least $2/3$. For any strictly IC mechanism, by weak monotonicity, for type $(1, 1+\varepsilon)$ the allocation probability of item A must be smaller than the allocation probability of item B. Then the induced allocation space of strictly IC mechanism contains $(a, 1-a)$ with probability at least $2/3$, where $0 < a < 1$. $\square$

B.3 Omitted Definitions

Definition B.2 (DSIC/ε-DSIC/BIC/ε-BIC Mechanism)**.** *A mechanism $\mathcal{M}$ is called ε-BIC iff for all i, t_i:*

$$\mathbf{E}_{t_{-i}\sim\mathcal{F}_{-i}}[u_i(t_i, \mathcal{M}(t))] \geqslant \max_{\widehat{t}_i\in\mathcal{T}_i} \mathbf{E}_{t_{-i}\sim\mathcal{F}_{-i}}[u_i(t_i, \mathcal{M}(\widehat{t}_i; t_{-i}))] - \varepsilon$$

In other words, $\mathcal{M}$ is ε-BIC iff any agent will not gain more than ε by misreporting $\widehat{t}_i$ instead of true type t_i. Similarly, $\mathcal{M}$ is ε-DSIC iff for all $i, t_i, \widehat{t}_i, t_{-i} : u_i(t_i, \mathcal{M}(t)) \geqslant u_i(t_i, \mathcal{M}(\widehat{t}_i; t_{-i})) - \varepsilon$.

A mechanism is called BIC iff it is 0-BIC and DSIC iff it is 0-DSIC.

Definition B.3 (Individual Rationality)**.** *A BIC/ε-BIC mechanism $\mathcal{M}$ satisfies interim individual rationality (interim IR) iff for all i, v_i:*

$$\mathbf{E}_{t_{-i} \sim \mathcal{F}_{-i}}[u_i(t_i, \mathcal{M}(t))] \geqslant 0$$

This becomes ex-post individual rationality (ex-post IR) iff for all $i, t_i, t_{-i}, u_i(t_i, \mathcal{M}(t)) \geqslant 0$ with probability 1, over the randomness of the mechanism.

Definition B.4 (Interdependent private type)**.** *Each agent $i \in [n]$ has a private signal s_i, which captures her private information and the type of every agent t_i depends on the entire signal profile, $s = (s_1, \cdots, s_n)$.*

B.4 Omitted Proofs

B.4.1 Proof of Claim 3.2

Proof. First, in Step 1, since we only rotate the allocation and payment of nodes in $\mathcal{C}$, the total weight of the edges from nodes in $\mathcal{T} \backslash \mathcal{C}$ to nodes in $\mathcal{C}$ remains the same. Second, each node in $\mathcal{C}$ achieves a utility no worse than before, so that the weight of each outgoing edge from nodes in $\mathcal{C}$ to nodes in $\mathcal{T} \backslash \mathcal{C}$ will not increase. Third, since $\mathcal{C}$ is the shortest cycle, there are no other edges among nodes in $\mathcal{C}$ in addition to edges in $\mathcal{C}$, which implies we cannot create new edges among nodes in $\mathcal{C}$ by this rotation. It follows that this rotation decreases the total weights of graph G by the weights of $\mathcal{C}$. Finally, the expected revenue achieved by types $t^{(1)}, \cdots, t^{(l)}$ is still the same, since Step 1 only rotates the allocation and payment rules, and the probability of each type is the same. Combining the fact that each node gets a weakly preferred outcome, the social welfare does not decrease. □

B.4.2 Proof of Claim 3.3

Proof. In Step 2, we first prove that it can only create new edges with zero weight. A new edge created by Step 2 can only point to a node $\bar{t} \in S_t$. We show by contradiction, suppose we create a positive weight edge from $\hat{t}$ to $\bar{t} \in S_t$, then $u(\hat{t}, \mathcal{M}'(\bar{t})) - u(\hat{t}, \mathcal{M}'(\hat{t})) > 0$ for the current updated mechanism $\mathcal{M}'$, we have

$$\begin{aligned}
u(\widehat{t}, \mathcal{M}'(\widehat{t})) < u(\widehat{t}, \mathcal{M}'(\bar{t})) < u(\widehat{t}, \mathcal{M}'(\bar{t})) + \varepsilon_t \\
&\leqslant u(\widehat{t}, \mathcal{M}'(\bar{t})) + u(t'', \mathcal{M}'(t'')) - u(t'', \mathcal{M}'(\bar{t})) \\
&\leqslant u(\widehat{t}, \mathcal{M}'(\bar{t})) + u(\widehat{t}, \mathcal{M}'(\widehat{t})) - u(\widehat{t}, \mathcal{M}'(\bar{t})) \qquad \text{(By definition of } t''\text{)} \\
&= u(\widehat{t}, \mathcal{M}'(\widehat{t})),
\end{aligned}$$

which proves our claim. Second, it is straightforward to verify that Step 2 doesn't decrease social welfare since we only decrease payment in Step 2. Finally, in Step 2, we reduce the weight of every positive-weight outgoing edge associated with t by $\min\{\varepsilon_t, \bar{\varepsilon}_t\}$. This is because for any node t', s.t. there is a positive-weight edge between t and t', t' cannot be the ancestor of t, otherwise, there is already a cycle, which contradicts Step 1. □

B.4.3 Proof of Theorem 3.5

Proof. We construct the same weighted directed graph $G = (\mathcal{T}, E)$ as in the proof of Theorem 3.1. Again, the target is to reduce the total weight of G to zero, which leads to a BIC mechanism. We denote M_e as the menus and $|M_e| = C$, and we have for each type $t^{(i)}$, that there exists a menu $m_e \in M_e$, s.t. $\mathcal{M}(t^{(i)}) = m_e$. If $t^{(i)}$ and $t^{(j)}$ share a same menu, i.e., $\mathcal{M}(t^{(i)}) = \mathcal{M}(t^{(j)})$, there is an directed edge with weight zero from $t^{(i)}$ to $t^{(j)}$, and vice versa. We denote the distribution of each menu m_e as,

$$g(m_e) = \sum_{t \in T : \mathcal{M}^{\varepsilon}(t) = m_e} f(t).$$

Since $\mathcal{M}$ is ε-BIC, the weight of each edge is bounded by ε. We still apply Step 1 and Step 2 in graph G proposed in Theorem 3.1, however, we count the revenue loss over menu space.

First, in Step 1, we only rotate the allocation and payment (menu) along the cycle, it will not change the allocation and payment of each menu. In addition, it will not the distribution of menus, $g(m_e)$ is preserved for each m_e.

In Step 2, consider a source node t, and let the corresponding menu be m_e' (the output of the current mechanism with type t). Every type with m_e' is the ancestor of type t, when we decrease the payment of type t by $\min\{\varepsilon_t, \bar{\varepsilon}_t\}$, the payment for each type t' associated with menu m'_e will be decreased by the same amount. If there is a type t'' with a different menu $m''_e \neq m'_e$ and t'' is an ancestor of t, then all the types associated with menu m''_e are the ancestors of t. Thus, in Step 2, the payment of the types with the same menu must be decreased by the same amount. Therefore, Step 2 only changes the payment of each menu by the same amount, and does not change the distribution of each menu, i.e. $g(m_e)$ is the same for each $m_e \in M_e$.

Moreover, if there is an edge $(t^{(j)}, t^{(k)})$ with positive weight and if $t^{(j)}$ and t share the same menu, then (1) $t^{(k)}$ must be in different menus, and (2) $t^{(k)}$ is not the ancestor of t, otherwise, there exists a cycle, which contains a positive-weight edge. Therefore, in Step 2, if we decrease the payment of type t by $\min\{\varepsilon_t, \bar{\varepsilon}_t\}$, we also reduce the weight of edge $(t^{(j)}, t^{(k)})$ by $\min\{\varepsilon_t, \bar{\varepsilon}_t\}$. In other words, we reduce the regret of all the nodes in menu m_e by $\min\{\varepsilon_t, \bar{\varepsilon}_t\}$.

Since the weight of each edge is bounded by ε, then we may decrease the expected payment at most ε to reduce all the regret of the nodes belonging to menu m_e. In total, the revenue loss is bounded by $C\varepsilon$. $\square$

B.4.4 Proof of Theorem 3.6

Proof. We construct a weighted directed graph $G = (\mathcal{T}, E)$, different with the one in Theorem 3.1. A directed edge $e = (t^{(j)}, t^{(k)}) \in E$ is drawn from $t^{(j)}$ to $t^{(k)}$ when the outcome (allocation and payment) of $t^{(k)}$ is weakly preferred by true type $t^{(j)}$, i.e. $u(t^{(j)}, \mathcal{M}(t^{(k)})) \geqslant u(t^{(j)}, \mathcal{M}(t^{(j)}))$, and the weight of edge e is

$$w(e) = f(t^{(j)}) \cdot f(t^{(k)}) \cdot \left[u(t^{(j)}, \mathcal{M}(t^{(k)})) - u(t^{(j)}, \mathcal{M}(t^{(j)}))\right]$$

It is straightforward to see that $\mathcal{M}$ is BIC iff the total weight of all edges in G is zero.

We show the modified transformation for this setting in Fig. B.2. Firstly, it is trivial that our transformation preserves IR, since neither Step 1 nor Step 2 reduces utility. Then we show this modified Step 1 will strictly decrease the total weights of the graph G and has no negative effect on social welfare and revenue.

First, we observe each type in $\mathcal{C}$ achieves utility no worse than before, by truthful reporting. Then, the weight of each outgoing edge from a type in $\mathcal{C}$ to a type in $\mathcal{T} \backslash \mathcal{C}$ will not increase.

Second, we claim the total weight of edges from any node (type) $t \in \mathcal{T} \backslash C$ to nodes (types) in $\mathcal{C}$ does not increase. To prove this, we assume $w(t, t^{(j)}) \geqslant 0, \forall t^{(j)} \in \mathcal{C}$, i.e. there is a edge from t to any $t^{(j)} \in \mathcal{C}$ in G. This is WLOG, because if there is no edge between t to some $t^{(j)} \in C$, we can just add an edge from t to $t^{(j)}$ with weight zero, and this does not change the total weight of the graph. We denote the mechanism updated after one use of Step 1 as $\mathcal{M}'$, and denote the weight function w' for the graph G' that is constructed from $\mathcal{M}'$. Let $[\cdot]_+$ be the function $\max(\cdot, 0)$. The total weight from t to $t^{(j)} \in \mathcal{C}$ according to the mechanism $\mathcal{M}' = (x', p')$ is

$$
\begin{aligned}
& \sum_{t^{(j)} \in \mathcal{C}} w'(t, t^{(j)}) \\
= \; & \sum_{j=1}^{l} f(t^{(j)}) f(t) \big[u(t, \mathcal{M}'(t^{(j)})) - u(t, \mathcal{M}'(t)) \big]_+ \\
= \; & \sum_{j=1}^{l} f(t^{(j)}) f(t) \left[\frac{(f(t^{(j)}) - f(t^{(k)})) u(t, \mathcal{M}(t^{(j)})) + f(t^{(k)}) \cdot u(t, \mathcal{M}(t^{(j+1)}))}{f(t^{(j)})} - u(t, \mathcal{M}(t)) \right]_+ \\
& \text{(In the fractional rotation step, } \mathcal{M}'(t) = \mathcal{M}(t), \forall t \in \mathcal{T} \backslash \mathcal{C}) \\
\leqslant \; & \sum_{j=1}^{l} f(t) \cdot \Big((f(t^{(j)})) - f(t^{(k)}) \big) \big[u(t, \mathcal{M}^{\varepsilon}(t^{(j)})) - u(t, \mathcal{M}^{\varepsilon}(t)) \big]_+ \\
& + \sum_{j=1}^{l} f(t^{(k)}) \big[u(t, \mathcal{M}(t^{(j+1)})) - u(t, \mathcal{M}(t)) \big]_+ \Big) \\
& \text{(By rearranging the algebra and the fact that } [x+y]_+ \leqslant [x]_+ + [y]_+) \\
= \; & \sum_{j=1}^{l} f(t^{(j)}) \cdot f(t) \cdot \big[u(t, \mathcal{M}(t^{(j)})) - u(t, \mathcal{M}^{\varepsilon}(t)) \big]_+ \\
& \text{(By the fact that } \{t^{(1)}, \cdots, t^{(l)}\} \text{ forms a cycle and } t^{(l+1)} = t^{(1)}) \\
= \; & \sum_{t^{(j)} \in \mathcal{C}} w(t, t^{(j)})
\end{aligned}
$$

Thus, we prove our claim that the total weight of edges from any node (type) $t \in \mathcal{T} \backslash C$ to nodes (types) in $\mathcal{C}$ does not increase.

Third, by each use of modified Step 1, we remove one cycle and reduce the weight of edge $(t^{(i)}, t^{(i+1)})$ to zero, thus, we decrease the total weight at least by $f(t^{(k)}) f(t^{(k+1)}) (u(t^{(k)}, \mathcal{M}(t^{(k+1)})) - u(t^{(k)}, \mathcal{M}(t^{(k)}))$.

Finally, after one use of Step 1, the expected revenue achieved by types in $\mathcal{C}$ maintains,

Modified Step 1 (Fractional rotation step). Given a mechanism $\mathcal{M} = (x, p)$, find the shortest cycle $\mathcal{C}$ in G that contains at least one edge with positive weight in E. Without loss of generality, we represent $\mathcal{C} = \{t^{(1)}, t^{(2)}, \cdots, t^{(l)}\}$. Then we find the node $t^{(k)}, k \in [l]$, such that $f(t^{(k)}) = \min_{k \in [l]} f(t^{(k)})$. Next, we rotate the allocation and payment rules of types along $\mathcal{C}$ with fraction of $f(t^{(k)})/f(t^{(j)})$ for each type $t^{(j)}, j \in [l]$. Now we slightly abuse the notation of subscripts, s.t. $t^{(l+1)} = t^{(1)}$. Specifically, the allocation and payment rules for each $t^{(j)}$,

$$x'(t^{(j)}) = \frac{[f(t^{(j)}) - f(t^{(k)})]x(t^{(j)}) + f(t^{(k)})x(t^{(j+1)})}{f(t^{(j)})},$$
$$p'(t^{(j)}) = \frac{[f(t^{(j)}) - f(t^{(k)})]p(t^{(j)}) + f(t^{(k)})p(t^{(j+1)})}{f(t^{(j)})}.$$

Then we update mechanism $\mathcal{M}$ to adopt allocation and payment rules x', p' to form a new mechanism $\mathcal{M}'$ and reconstruct the graph G. If this has the effect of removing all cycles that contain at least one positive-weight-edge in G, then move to Step 2. Otherwise, we repeat Step 1.
Modified Step 2 (Payment reducing step). Exactly the same as Step 2 in Theorem 3.1.

Figure B.2: *ε-BIC to BIC transformation for single agent with general type distribution.*

because

$$\begin{aligned}
\sum_{j=1}^{l} f(t^{(j)}) \cdot p'(t^{(j)}) &= \sum_j (f(t^{(j)}) - f(t^{(k)})) \cdot p(t^{(j)}) + f(t^{(k)}) \cdot p(t^{(j+1)}) \\
&= \sum_j f(t^{(j)})p(t^{(j)}) + ft^{(k)} \sum_j p(t^{(j)}) - p(t^{(j+1)}) \\
&= \sum_j f(t^{(j)})p(t^{(j)}) && \text{(Because } t^{(l+1)} = t^{(1)}\text{)}
\end{aligned}$$

The modified Step 2 is the same as Step 2 in Fig. 3.2. At each step 2, we decrease the total weight of the graph by at least $\min\{\varepsilon_t, \bar{\varepsilon}_t\}$. We count the revenue loss as follows, in each Step 2, if we decrease the payment of t by $\min\{\varepsilon_t, \bar{\varepsilon}_t\}$, the expected revenue loss is bounded by

$$\sum_j f(t^{(j)}) \min\{\varepsilon_t, \bar{\varepsilon}_t\} \leqslant \min\{\varepsilon_t, \bar{\varepsilon}_t\}$$

Since the weight of each edge is bounded by ε, to reduce the weight of outgoing edges of t to zero, we may decrease the expected revenue by ε. Therefore, in total, the expected revenue loss is bounded by $m\varepsilon$. □

B.4.5 Proof of Theorem 3.7

Proof. We construct the type distribution and the ε-EEIC mechanism similar to the one in Theorem 3.4. We consider a single agent with m types $\mathcal{T} = \{t^{(1)}, \cdots, t^{(m)}\}$. The type distribution is $f(t^{(1)}) = \frac{1}{2} - \frac{\varepsilon}{2m}$, $f(t^{(2)}) = \frac{\varepsilon}{2m}$ and $f(t^{(j)}) = \frac{1}{2(m-2)}, \forall j \geqslant 3$. The agent with type $t^{(1)}$ values outcome 1 at ε and the other outcomes at 0. For any type $t^{(j)}, j \geqslant 2$, the agent with type $t^{(j)}$ values outcome $j-1$ at $m + (j-1)\varepsilon$, outcome j at $m + (j-1)\varepsilon$, and the other outcomes at 0. The mechanism we consider is: (1) if the agent reports type $t^{(1)}$, gives the outcome 1 to the agent and charges ε. (2) if the agent reports $t^{(j)}, j \geqslant 2$, gives the outcome j to the agent and charges $m + (j-1)\varepsilon$. There is a m regret to the agent for not misreporting type $t^{(1)}$ with true type $t^{(2)}$ and a regret ε for not reporting $t^{(j)}$ with true type $t^{(j+1)}$, for any $j \geqslant 2$. It is easy to verify that this mechanism is ε-EEIC (the probability of type $t^{(2)}$ is small) and already maximizes social welfare. Thus, we can only change the payment to reduce the regret of each type. Following the same argument as in Theorem 3.4, to reduce all the regret of the types, the revenue loss in total is at least

$$
\begin{aligned}
f(t^{(2)})m + \sum_{j=3}^{m} f(t^{(j)})(m + (j-2)\varepsilon) &= \frac{\varepsilon}{2} + \frac{1}{2(m-2)} \sum_{j=3}^{m} m + (j-2)\varepsilon \\
&= \frac{\varepsilon}{2} + \frac{m}{2} + \frac{(m-1)\varepsilon}{4} \geqslant \frac{m}{2}
\end{aligned}
$$

□

B.4.6 Proof of Theorem 3.8

The earlier proof approach for single agent case does not immediately extend to the multi-agent setting. However, since our target is a BIC mechanism, we can work with interim rules (see Definition 3.1), and this provides an approach to the transformation. The interim rules reduce the dimension of type space and separate the type of each agent. With this, we can construct a separate type graph for each agent, now based on the interim rules.

To simplify the presentation, we define the *induced mechanism* for each agent i of a mechanism $\mathcal{M}$ as follows.

Definition B.5 (Induced Mechanism)**.** *For a mechanism $\mathcal{M} = (x, p)$, an induced mechanism*

$\widetilde{\mathcal{M}}_i = (X_i, P_i)$ *is a pair of interum allocation rule* $X_i : \mathcal{T}_i \to \Delta(\mathcal{O})$ *and interim payment rule* $P_i : \mathcal{T}_i \to \mathbb{R}_{\geqslant 0}$. *Denote the utility function* $u_i(t_i, \widetilde{\mathcal{M}}_i(t_i)) = v_i(t_i, X_i(t_i)) - P_i(t_i)$.

The following lemma shows that given an ε-BIC/ε-EEIC mechanism, then the induced mechanism for each agent is also ε-BIC/ε-EEIC.

Lemma B.1. *For a ε-EEIC/ε-BIC mechanism $\mathcal{M}$, any induced mechanism $\widetilde{\mathcal{M}}_i$ for each agent i is ε-EEIC/ε-BIC.*

Proof. By ε-BIC definition, each induced mechanism $\widetilde{\mathcal{M}}_i$ must be ε-BIC, if the original mechanism $\mathcal{M}$ is ε-BIC. Now, we turn to consider ε-EEIC mechanism $\mathcal{M}$, for any induced mechanism $\widetilde{\mathcal{M}}_i$

$$\begin{aligned}
& \mathbf{E}_{t_i \sim \mathcal{F}_i}\left[\max_{t_i' \in \mathcal{T}_i} u_i(t_i, \widetilde{\mathcal{M}}_i(t_i')) - u_i(t_i, \widetilde{\mathcal{M}}_i(t_i))\right] \\
= \; & \mathbf{E}_{t_i \sim \mathcal{F}_i}\left[\max_{t_i' \in \mathcal{T}_i} \mathbf{E}_{t_{-i} \sim \mathcal{F}_{-i}}\left[u_i(t_i, \mathcal{M}(t_i'; t_{-i})) - u_i(t_i, \mathcal{M}(t_i; t_{-i}))\right]\right] \\
\leqslant \; & \mathbf{E}_{t_i \sim \mathcal{F}_i}\left[\mathbf{E}_{t_{-i} \sim \mathcal{F}_{-i}}\left[\max_{t_i' \in \mathcal{T}_i} u_i(t_i, \mathcal{M}(t_i'; t_{-i})) - u_i(t_i, \mathcal{M}(t_i; t_{-i}))\right]\right] \\
& \text{(By Jenson's inequality and convexity of max function)} \\
= \; & \mathbf{E}_{t \sim \mathcal{F}}\left[\max_{t_i' \in \mathcal{T}_i} u_i(t_i, \mathcal{M}(t_i'; t_{-i})) - u_i(t_i, \mathcal{M}(t_i; t_{-i}))\right] \\
& \text{(By independence of agents' types)} \\
\leqslant \; & \varepsilon.
\end{aligned}$$

□

Given Lemma B.1, we can construct a single type graph for each agent based on the induced mechanism and apply the same technique for each graph as the one in Theorem 3.6. The challenge will be to also handle feasibility of the resulting mechanism. We summarize these approaches in the following proof for Theorem 3.8.

Proof of Theorem 3.8. Here, we focus on the ε-BIC setting. The proof for ε-EEIC with independent uniform type distribution is analogous.

We construct a graph $G_i = (\mathcal{T}_i, E_i)$ for each agent $i \in [n]$, such that there is a directed edge from $t_i^{(j)}$ to $t_i^{(k)}$ if and only if $u(t_i^{(j)}, \widetilde{\mathcal{M}^{\varepsilon}}_i(t_i^{(k)})) \geqslant u(t_i^{(j)}, \widetilde{\mathcal{M}^{\varepsilon}}_i(t_i^{(j)}))$ and the weight is

$$w_i((t_i^{(j)}, t_i^{(k)})) = f_i(t^j) \cdot f_i(t_i^{(k)}) \cdot \left(u(t_i^{(j)}, \widetilde{\mathcal{M}}_i(t_i^{(k)})) - u(t_i^{(j)}, \widetilde{\mathcal{M}}_i(t_i^{(j)})\right)$$

Based on Lemma B.1, each graph is constructed by an ε-BIC induced mechanism $\widetilde{\mathcal{M}^{\varepsilon}}_i$, we can apply the same constructive proof in Theorem 3.6 to reduce the total weight of each graph G_i to be 0. An astute reader may have already realized that changing type graph G_i may affect other graphs, since we probably change the distribution of the reported type of agent i. However, in our transformation, both Step 1 and Step 2 don't change the density probability of each type (we only change the interim allocation and payment for each type), therefore when we do transformation for one type graph G_i of agent i, it has no effect on the interim rules of the other agents.

Here, if the total weight of all graphs G_i are all 0, it implies that any induced mechanism $\widetilde{\mathcal{M}^{\varepsilon}}_i$ is IC. Therefore, we make the mechanism BIC. Similarly, the new mechanism after transformation achieves at least the same social welfare and the revenue loss of each graph G_i is bounded by $m_i\varepsilon$, Hence, the total revenue loss is bounded by $\sum_{i=1}^n m_i\varepsilon = \sum_{i=1}^n |\mathcal{T}_i|\varepsilon$.

What is left to show is that using modified steps 1 and 2 on each graph G_i shown in Theorem 3.6 does not violate the feasibility of the mechanism. We only change the allocation of each type in modified Step 1 (Rotation step). Denote by X_i the interim allocation for agent i before one rotation step, and let X_i' denote the updated interim allocation for agent i after one rotation step. We then claim in the modified Step 1 in Theorem 3.6,

$$\sum_{t_i \in \mathcal{T}_i} f_i(t_i) X_i(t_i) = \sum_{t_i \in \mathcal{T}_i} f_i(t_i) X_i'(t_i).$$

To prove this claim, WLOG, we consider a l length cycle $\mathcal{C} = \{t_i^{(1)}, t_i^{(2)}, \cdots, t_i^{(l)}\}$ in modified Step 1. Let $k = \arg\min_{j \in [l]} f_i(t^{(j)})$. We observe the interim allocation of the types in $\mathcal{T}_i \backslash \mathcal{C}$ don't change in modified Step 1, i.e., $\forall t_i \in \mathcal{T}_i \backslash \mathcal{C}, X_i(t_i) = X_i'(t_i)$. We slightly abuse the notation here, and let $t^{(l+1)} = t^{(1)}$. For the types in cycle $\mathcal{C}$,

$$\sum_{j \in [l]} f_i(t^{(j)}) X_i'(t^{(j)}) \quad = \quad \sum_{j \in [l]} f(t^{(j)}) \cdot \frac{(f(t^{(j)}) - f(t^{(k)})) X_i(t^{(k)}) + f(t^{(k)}) \cdot X_i(t^{(j+1)})}{f(t^{(j)})}$$

$$= \sum_{j\in[l]} f_i(t^{(j)})X_i(t^{(j)}),$$

which validates the claim. Therefore, by Border's lemma [Bor91], the rotation step maintains the feasibility of the allocation.

Running time. Suppose we have oracle access to the interim quantities of the original mechanism, we can build each G_i in $\texttt{poly}(|\mathcal{T}_i|)$ time. Similarly, at each modified step 1 (shown in Fig. B.2), we strictly reduce the weight of at least one edge with positive weight to 0 in each type graph G_i. Then, the running time for each type graph G_i is $\texttt{poly}(|\mathcal{T}_i|)$ following the same argument for the single agent setting. In total the running time is $\texttt{ploy}(\sum_i |\mathcal{T}_i|)$. □

B.4.7 Proof of Theorem 3.9

Proof. It is straightforward to construct an example such that the type graph of each agent induced by the interim rules is the same as the type graph constructed by the mechanism shown in Theorem 3.4. For instance, agent i values outcomes $\{o_i^{(1)}, \cdots, o_i^{(m_i)}\}$ in the same way as the one constructed in Theorem 3.4. We assume the outcome $o_i^{(j)}$ are disjoint, for any i and $j \in [m_i]$. Indeed, this is also ε-DSIC mechanism. Thus, we show for this case, that the revenue loss must be at least $\Omega(\sum_i |\mathcal{T}_i|\varepsilon)$, if we want to maintain the social welfare, following the same argument in Theorem 3.4. □

B.4.8 Proof of Theorem 3.10

Proof. Consider a setting with two items A and B and two unit-demand agents 1 and 2. The two agents share the same preference order on items. Moreover, agent 1 is informed about which is better, while agent 2 has no information. Agent 1 values the better item at $1+\varepsilon$ and the other item at 1. Agent 2 values the better item at 2 and the other item at 0.

There exists an ε-IC mechanism: ask agent 1 which item is better, and give this item to agent 2 for a price of 2 and give agent 1 the other item for a price of 1. The total welfare and revenue is 3 if agent 1 reports truthfully. Bidder 1 can get ε more utility by misreporting, in which case it will get the better item for the same price. From this, we can confirm that this is an ε-IC mechanism.

For any IC mechanism, by weak monotonicity, we have $v_A(x(A)) - v_A(x(B)) \geqslant v_B(x(A)) - v_B(x(B))$, where v_A be the type that the better item is A, and similarly for v_B. $x(A)$ is the allocation if agent 1 reports A the better item and similarly for $x(B)$. This means that when agent 1 reporting A rather than B, either agent 1 is assigned item A with weakly higher probability, or agent 1 is assigned item B with weakly less probability. We only consider the former case, and the latter one holds analogously. In the former case, we have either:

(1) agent 1 is getting at least half of A when reporting A, and the total revenue and social welfare are each at most $0.5 \times 2 + 0.5 \times (1 + \epsilon) + 1 = 2.5 + \epsilon/2$, or

(2) agent 1 is getting at most half of A when reporting B, and the total revenue and social welfare are each at most $2 + 0.5 = 2.5$.

Either way, we will definitely lose at least $0.5 - \varepsilon/2$ for revenue and social welfare when making the ε-IC mechanism above BIC. □

B.4.9 Proof of Theorem 3.11

Proof. The construction of this ε-BIC mechanism is strictly generalized by the mechanism in [Yao17]. Consider a 2-agent, 2-item auction, each agent i values item j, t_{ij}. t_{ij} is i.i.d sampled from a uniform distribution over set $\{1, 2\}$, i.e. $\mathbb{P}(t_{ij} = 1) = \mathbb{P}(t_{ij} = 2) = 0.5$. The ε-BIC mechanism is shown as below,

> If $t_2 = (1, 1)$, give both items to agent 1 for a price of 3.
>
> If $t_2 = (1, 2)$ and $t_1 = (1, 2)$, give both items randomly to agent 1 or 2 for a price of 1.5.
>
> If $t_1 = (2, 1)$ and $t_2 = (1, 2)$, give item 1 to agent 1 and give item 2 to agent 2, with a price of 2 for each.
>
> If $t_2 = (1, 2)$ and $t_1 = (2, 2)$, give both items to agent 1 for a price $3.75 + \varepsilon$.
>
> If $t_1 = t_2 = (2, 2)$, give both items randomly to agent 1 or agent 2 for a price 2.
>
> For other cases, we get the mechanism by the symmetries of items and agents.

It is straightforward to verify that this is an ε-BIC mechanism and the expected revenue is $3.1875 + \varepsilon/16$. However, Yao [Yao17] characterizes that optimal DSIC mechanism achieves expected 3.125. This conclude the proof. □

B.5 Omitted Details of Applications

In this section, we give a brief introduction to LP-based AMD and RegretNet AMD.

B.5.1 LP-based Approach

The LP-based approach considered in this paper is initiated by [CS02]. We consider n agents with type distribution $\mathcal{F}$ defined on $\mathcal{T}$. For each type profile $t \in \mathcal{T}$ and each outcome $o_k \in O$, we define $x^k(t)$ as the probability of choosing o_k when the reported types are t and $p_i(t)$ as the expected payment of agent i when the reported types are t. $x^k(t)$ and $p_i(t)$ are both decision variables.

Then we can formulate the mechanism design problem as the following linear programming,

$$
\begin{aligned}
&\max_{x,p}(1-\lambda)\mathbf{E}_{t\sim\mathcal{F}}\left[\sum_i p_i(t)\right] + \lambda\mathbf{E}_{t\sim\mathcal{F}}\left[\sum_{k:o_k\in O} x^k(t)\sum_i v_i(t_i,o_k)\right]\\
s.t.\quad &\mathbf{E}_{t_{-i}}\left[\sum_{k:o_k\in O} x^k(t_i,t_{-i})v_i(t_i,o_k) - p_i(t_i,t_{-i})\right] \geqslant \mathbf{E}_{t_{-i}}\left[\sum_{k:o_k\in O} x^k(t'_i,t_{-i})v_i(t_i,o_k) - p_i(t'_i,t_{-i})\right], \forall i, t_i, t'_i\\
&\mathbf{E}_{t_{-i}}\left[\sum_{k:o_k\in O} x^k(t)v_i(t_i,o_k) - p_i(t)\right] \geqslant 0, \forall i, t
\end{aligned}
$$

where the first constraint is for BIC and the second is for interim-IR. In this case, the type space $\mathcal{T}$ is discrete, thus the expectation can be explicitly represented as the linear function with decision variables.

B.5.2 RegretNet Approach

RegretNet (Chapter 1) is a generic data-driven, deep learning framework for multi-dimensional mechanism design. We only briefly introduce the RegretNet framework here and refer the readers to Chapter 1 for more details.

RegretNet uses a deep neural network parameterized by $w \in \mathbb{R}^d$ to model the mechanism $\mathcal{M}$, as well as the valuation (through allocation function $x^w : \mathcal{T} \to \Delta(O)$) and payment functions: $v_i^w : \mathcal{T}_i \times \Delta(O) \to \mathbb{R}_{\geqslant 0}$ and $p_i^w : \mathcal{T} \to \mathbb{R}_{\geqslant 0}$. Denote utility function as,

$$u_i^w(t_i, \widehat{t}) = v_i(t_i, x^w(\widehat{t})) - p_i^w(\widehat{t}).$$

RegretNet is trained on a training data set $\mathcal{S}$ of S type profiles i.i.d sampled from $\mathcal{F}$ to maximize the empirical revenue subject to the empirical regret being zero for all agents:

$$\begin{aligned} \max_{w \in \mathbb{R}^d} \frac{1-\lambda}{S} \sum_{t \in \mathcal{S}} \sum_{i=1}^{n} p_i^w(t) + \frac{\lambda}{S} \sum_{t \in \mathcal{S}} \sum_{i=1}^{n} v_i^w(t_i, x(t)) \\ s.t. \quad \frac{1}{S} \sum_{t \in \mathcal{S}} \left[\max_{t_i' \in \mathcal{T}_i} u_i^w(t_i, (t_i', t_{-i})) - u_i^w(t_i, t) \right] = 0, \forall i \end{aligned}$$

The objective is the empirical version of learning target in 3.7. The constraint is for EEIC requirement and IR is hard coded in RegretNet to be guaranteed. Let $\mathcal{H}$ be the functional class modeled by RegretNet through parameters w. In this paper, we assume there exists an PAC learning algorithm that can produce a RegretNet to model an ε-EEIC mechanism $\mathcal{M} \in \mathcal{H}$ defined on $\mathcal{F}$, such that

$$\mu_\lambda(\mathcal{M}', \mathcal{F}) \geqslant \sup_{\widehat{\mathcal{M}} \in \mathcal{H}} \mu_\lambda(\widehat{\mathcal{M}}, \mathcal{F}) - (1-\lambda) \sum_{i=1}^{n} |\mathcal{T}_i| \varepsilon - \varepsilon,$$

holds with probability at least $1 - \delta$, by observing $S = S(\varepsilon, \delta)$ i.i.d samples from $\mathcal{F}$.

Appendix C

Appendix to Chapter 4

C.1 Missing Proofs

C.1.1 Proof of Lemma 4.1

Proof. We prove this lemma by considering the following cases,

- If $b \geqslant v + \frac{1}{H}$, then $m_{i,t} = b$ implies $u_{i,t}((v, b_{-i,t}); v) = 0$, whereas, $u_{i,t}((b, b_{-i,t}); v) = v - b \leqslant -\frac{1}{H}$ with probability at least $1/n$, because of random tie-breaking. Therefore,

$$\begin{aligned}
& \mathbb{P}\left(u_{i,t}((v, b_{-i,t}); v) - u_{i,t}((b, b_{-i,t}); v) \geqslant \frac{1}{H} \Big| b \geqslant v + \frac{1}{H}\right) \\
\geqslant \; & \mathbb{P}\left(u_{i,t}((v, b_{-i,t}); v) - u_{i,t}((b, b_{-i,t}); v) \geqslant \frac{1}{H}, m_{i,t} = b \Big| b \geqslant v + \frac{1}{H}\right) \\
\geqslant \; & \mathbb{P}\left(u_{i,t}((b, b_{-i,t}); v) \leqslant -\frac{1}{H}, m_{i,t} = b \Big| b \geqslant v + \frac{1}{H}\right) \\
\geqslant \; & \frac{1}{n}\left(\frac{1}{H}\right)^{n-1} \geqslant \frac{\tau}{n}
\end{aligned}$$

  where the second last inequality holds because $\mathbb{P}(m_{i,t} = b) \geqslant \left(\frac{1}{H}\right)^{n-1}$ and the last inequality is based on the fact that $\frac{1}{H^{n-1}} \geqslant \tau$.

- If $b \leqslant v - \frac{1}{H}$, then $m_{i,t} = b$ implies $u_{i,t}((v, b_{-i,t}); v) = v - m_{i,t} \geqslant \frac{1}{H}$, whereas, $u_{i,t}(b, b_{-i,t}); v)) = 0$ with both probability at least $1/n$ (random tie-breaking). Therefore,

$$\mathbb{P}\left(u_{i,t}((v, b_{-i,t}); v) - u_{i,t}((b, b_{-i,t}); v) \geqslant \frac{1}{H} \Big| b \leqslant v - \frac{1}{H}\right)$$

$$
\begin{aligned}
&\geqslant \quad \mathbb{P}\left(u_{i,t}((v,b_{-i,t});v) - u_{i,t}((b,b_{-i,t});v) \geqslant \frac{1}{H}, m_{i,t} = b \middle| b \leqslant v - \frac{1}{H}\right) \\
&\geqslant \quad \mathbb{P}\left(u_{i,t}((b,b_{-i,t});v) = 0, m_{i,t} = b \middle| b \leqslant v - \frac{1}{H}\right) \\
&\geqslant \quad \frac{1}{n}\left(\frac{1}{H}\right)^{n-1} \geqslant \frac{\tau}{n}
\end{aligned}
$$

□

C.1.2 Proof of Lemma 4.2

Proof. By Lemma 4.1, we have $\mathbf{E}[u_{i,s}((v,b_{-i,s});v) - u_{i,s}((b,b_{-i,s});v)] \geqslant \frac{\tau}{nH}$, for any fixed v, $b \neq v$ and $s \leqslant t$. Then by Chernoff bound, we have

$$
\begin{aligned}
&\mathbb{P}\left(\sum_{s\leqslant t} u_{i,s}((v,b_{-i,s});v) - u_{i,s}((b,b_{-i,s});v) \leqslant \frac{\tau t}{nH} - \frac{\tau t}{2nH}\right) \\
\leqslant \quad &\mathbb{P}\left(\sum_{s\leqslant t} u_{i,s}((v,b_{-i,s});v) - u_{i,s}((b,b_{-i,s});v) \leqslant t\mathbf{E}[u_{i,s}((v,b_{-i,s});v) - u_{i,s}((v,b_{-i,s});v)] - \frac{\tau t}{2nH}\right) \\
\leqslant \quad &\exp\left(-\frac{2\tau^2 t}{4n^2H^2}\right)
\end{aligned}
$$

□

C.1.3 Proof of Lemma 4.3

Proof. By definition of γ_t-mean-based learning algorithm, given a value v, each bid $b \neq v$ will be selected with probability at most γ_t for each bidder i, thus, $b_{i,t+1} \neq v_{i,t+1}$ holds with probability at most $H\gamma_t$ for any bidder i. By union bound, $m_{i,t+1} \neq z_{i,t+1}$ holds with probability at most $(n-1)H\gamma_t$, for all i.

$$
\begin{aligned}
&\mathbb{P}\left(u_{i,t+1}((v,b_{-i,t+1});v) - u_{i,t}((b,b_{-i,t+1});v) \geqslant \frac{1}{H}\right) \\
\geqslant \quad &\mathbb{P}\left(u_{i,t+1}((v,b_{-i,t+1});v) - u_{i,t+1}((b,b_{-i,t+1});v) \geqslant \frac{1}{H}, m_{i,t+1} = z_{i,t+1}\right) \\
\geqslant \quad &\mathbb{P}\left(u_{i,t+1}((v,b_{-i,t+1});v) - u_{i,t+1}((b,b_{-i,t+1});v) \geqslant \frac{1}{H}, m_{i,t+1} = z_{i,t+1}, z_{i,t+1} = b\right) \\
\geqslant \quad &(1-(n-1)H\gamma_t) \cdot \frac{\tau}{n} \geqslant \frac{\tau}{2n},
\end{aligned}
$$

where the third inequality is based on the same argument in Lemma 4.1 and the last inequality holds because $\gamma_t \leqslant \frac{1}{2nH}$. $\square$

C.1.4 Proof of Claim 4.3

Proof. Based on the construct of T_k and $\gamma_t \leqslant \frac{\tau}{8nH}, \forall t > T_0$, we have $T_k \geqslant 2T_{k-1}$. Then $|\Gamma_\ell| = T_\ell - T_{\ell-1} \geqslant 2^{\ell-1}T_0, \forall \ell \geqslant 1$, we have

$$\begin{aligned}
\sum_{\ell=0}^{k(t)} \exp\left(-\frac{|\Gamma_\ell|\tau^2}{32n^2H^2}\right) &\leqslant 2\exp\left(-\frac{\tau^2 T_0}{32n^2H^2}\right) + \sum_{\ell=2} \exp\left(-\frac{2^{\ell-1}\tau^2 T_0}{32n^2H^2}\right) \\
&\leqslant 2\exp\left(-\frac{\tau^2 T_0}{32n^2H^2}\right) + \sum_{\ell=1} \exp\left(-\frac{2^{\ell}\tau^2 T_0}{32n^2H^2}\right) \\
&= \exp\left(-\frac{\tau^2 T_0}{32n^2H^2}\right) \cdot \left(2 + \sum_{\ell=1} \exp\left(-\frac{(2^\ell - 1)\tau^2 T_0}{32n^2H^2}\right)\right) \\
&\leqslant \exp\left(-\frac{\tau^2 T_0}{32n^2H^2}\right) \cdot \left(2 + \sum_{\ell=1} \exp\left(-\frac{\ell\tau^2 T_0}{32n^2H^2}\right)\right) \\
&\quad \text{(Because } 2^\ell - 1 \geqslant \ell, \forall \ell \geqslant 1) \\
&\leqslant \exp\left(-\frac{\tau^2 T_0}{32n^2H^2}\right) \cdot \left(2 + \frac{1}{1 - \exp\left(-\frac{\tau^2 T_0}{32n^2H^2}\right)}\right) \\
&\leqslant 4\exp\left(-\frac{\tau^2 T_0}{32n^2H^2}\right),
\end{aligned}$$

where the last inequality holds, because $\exp\left(-\frac{\tau^2 T_0}{32n^2H^2}\right) \leqslant \frac{1}{2}$ if T_0 is large enough. $\square$

C.1.5 Proof of Theorem 4.4

Here we slightly abuse the notation, let $\lceil \frac{v}{2} \rceil := \{b \in V : b \geqslant \frac{v}{2}, \text{ and } b \leqslant \frac{v}{2} + \frac{1}{H}\}$. Notice, if $\frac{v}{2} \in V$, $\lceil \frac{v}{2} \rceil = \frac{v}{2}$. If $\frac{v}{2} \notin V$, $\lceil \frac{v}{2} \rceil = \frac{v}{2} + \frac{1}{2H}$. We prove this theorem based on the following claims,

Claim C.1. *For any $t \leqslant T_0$, any fixed v, any bid $b \neq \lceil \frac{v}{2} \rceil$, for each bidder i we have*

$$\mathbf{E}_{b_{-i,t}}\left[u_{i,t}((\lceil \frac{v}{2} \rceil, b_{-i,t}); v) - u_{i,t}((b, b_{-i,t}); v)\right] \geqslant \frac{1}{2H^2}$$

Proof. Let $\mathcal{U}_V$ denote the uniform distribution on V. Note, we assume the random tie-breaking in this paper, then we can rewrite the expected utility of bidder i when $t \leqslant T_0$ in the following way

$$\mathbf{E}_{b_{-i,t}\sim\mathcal{U}_V}\left[u_{i,t}((b,b_{-i,t});v)\right] = (v-b)\cdot\left(\left(b-\frac{1}{H}\right)+\frac{1}{2}\cdot\frac{1}{H}\right) = (v-b)\cdot\left(b-\frac{1}{2H}\right)$$

Then we consider two different cases in the following,

- $\frac{v}{2}\in V$, let $b=\frac{v}{2}+\alpha$, where $\alpha\geqslant\frac{1}{H}$ or $\alpha\leqslant-\frac{1}{H}$. Thus, we have

$$\begin{aligned}
& \left(v-\frac{v}{2}\right)\cdot\left(\frac{v}{2}-\frac{1}{2H}\right)-(v-b)\cdot\left(b-\frac{1}{2H}\right) \\
= & \frac{v^2}{4}-\frac{v}{4H}-\left(\frac{v^2}{4}-\alpha^2-\frac{v}{4H}+\frac{\alpha}{2H}\right) \\
= & \alpha^2-\frac{\alpha}{2H}\geqslant\frac{1}{2H^2}
\end{aligned}$$

- $\frac{v}{2}\notin V$, then $\lceil\frac{v}{2}\rceil=\frac{v}{2}+\frac{1}{2H}$. Let $b=\lceil\frac{v}{2}\rceil+\alpha$, where $\alpha\geqslant\frac{1}{H}$ or $\alpha\leqslant-\frac{1}{H}$. Thus, we have

$$\begin{aligned}
& \left(v-\left\lceil\frac{v}{2}\right\rceil\right)\cdot\left(\left\lceil\frac{v}{2}\right\rceil-\frac{1}{2H}\right)-(v-b)\cdot\left(b-\frac{1}{2H}\right) \\
= & \left(\frac{v}{2}-\frac{1}{2H}\right)\cdot\frac{v}{2}-\left(\frac{v}{2}-\frac{1}{2H}-\alpha\right)\cdot\left(\frac{v}{2}+\alpha\right) \\
= & \alpha^2+\frac{\alpha}{2H}\geqslant\frac{1}{2H^2}
\end{aligned}$$

Combining the above two cases, we complete the proof. □

Claim C.2. *For any fixed value v, any $t\leqslant T_0$, any bid $b\neq\lceil\frac{v}{2}\rceil$, we have for each bidder i,*

$$\mathbb{P}\left(\sum_{s\leqslant t}u_{i,s}((\lceil\frac{v}{2}\rceil,b_{-i,s});v)-\sum_{s\leqslant t}u_{i,s}((b,b_{-i,s});v)\leqslant\frac{\tau t}{2nH}\right)\leqslant\exp\left(-\frac{t}{8H^4}\right)$$

Proof. By Claim C.1 and Chernoff bound, we have

$$\begin{aligned}
& \mathbb{P}\left(\sum_{s\leqslant t}u_{i,s}((\lceil\frac{v}{2}\rceil,b_{-i,s});v)-\sum_{s\leqslant t}u_{i,s}((b,b_{-i,s});v)\leqslant\frac{t}{4H^2}\right) \\
\leqslant\ & \mathbb{P}\left(\sum_{s\leqslant t}u_{i,s}((\lceil\frac{v}{2}\rceil,b_{-i,s});v)-\sum_{s\leqslant t}u_{i,s}((b,b_{-i,s});v)\leqslant\sum_{s\leqslant t}\mathbf{E}_{b_{-i,s}}\left[u_{i,s}((\lceil\frac{v}{2}\rceil,b_{-i,s});v)-u_{i,s}((b,b_{-i,s});v)\right]-\frac{t}{4H^2}\right) \\
\leqslant\ & \exp\left(-\frac{2t}{16H^4}\right)=\exp\left(-\frac{t}{8H^4}\right)
\end{aligned}$$

□

Claim C.3. *For any $t > T_0$, for any fixed v, any bid $b \neq \lceil \frac{v}{2} \rceil$ and each bidder i, suppose $\sum_{s \leqslant t} u_{i,s}((\lceil \frac{v}{2} \rceil, b_{-i,s}); v) - u_{i,s}((b, b_{-i,s}); v) \geqslant \gamma_t t$ holds, then for any fixed value v, any bid $b \neq \lceil \frac{v}{2} \rceil$ and each bidder i, we have,*

$$\mathbf{E}_{b_{-i,t+1}} \left[u_{i,t+1}((\lceil \frac{v}{2} \rceil, b_{-i,t+1}); v) - u_{i,t+1}((b, b_{-i,t+1}); v) \right] \geqslant \frac{1}{2H^2}$$

Proof. We assume for each bidder i, with probability η_t^i bids $b_{i,i+1} = \lceil \frac{v_{i,t+1}}{2} \rceil$ and with probability $1 - \eta_t^i$ bids $b_{i,t+1} \neq \lceil \frac{v_{i,t+1}}{2} \rceil$. By the condition that $\sum_{s \leqslant t} u_{i,s}((\lceil \frac{v}{2} \rceil, b_{-i,s}); v) - u_{i,s}((b, b_{-i,s}); v) \geqslant \gamma_t t$ and definition of mean-based learning algorithm, we have $\eta_t^i \geqslant 1 - H\gamma_t$ for each bidder i.

Let $b_{j,t+1}, v_{j,t+1}$ be the bid and value from the other bidder $j \neq i$ at time $t+1$, respectively. Then we can show the lower bound of the expected utility for bidder i, when bid $b \leqslant \frac{1}{2}$, in the following,

$$\begin{aligned} & E_{b_{-i,t+1}} \left[u_{i,t+1}((b, b_{-i,t+1}); v) \right] \\ = \;& (v - b) \cdot \left(\mathbb{P}(b > b_{j,t+1}) + \frac{1}{2} \cdot \mathbb{P}(b_{j,t+1} = b) \right) \\ \geqslant \;& \eta_t^j \cdot (v - b) \cdot \left(\mathbb{P}(b > \lceil \frac{v_{j,t+1}}{2} \rceil) + \frac{1}{2} \cdot \mathbb{P}(\lceil \frac{v_{j,t+1}}{2} \rceil = b) \right) \end{aligned}$$

For any $b \leqslant \frac{1}{2}$, $2b \in V$. Then if $v_{j,t+1} \leqslant 2b - \frac{2}{H}$, $\lceil \frac{v_{j,t+1}}{2} \rceil < b$. Therefore, $\mathbb{P}(b > \lceil \frac{v_{j,t+1}}{2} \rceil) = 2b - \frac{2}{H}$. Notice, when $v_{j,t+1} = 2b - \frac{1}{H}$ or $v_{j,t+1} = 2b$, $\lceil \frac{v_{j,t+1}}{2} \rceil = b$, then $\mathbb{P}\left(\left\lceil \frac{v_{j,t+1}}{2} \right\rceil = b \right) = \frac{2}{H}$. Therefore, we can lower bound the expected utility for bidder i, when bid $b \leqslant \frac{1}{2}$,

$$E_{b_{-i,t+1}} \left[u_{i,t+1}((b, b_{-i,t+1}); v) \right] \geqslant 2\eta_t^j \cdot (v - b) \cdot \left(b - \frac{1}{2H} \right)$$

Similarly, we can upper bound the expected utility for bidder i, when bid $b \leqslant \frac{1}{2}$, shown as below,

$$\begin{aligned} & E_{b_{-i,t+1}} \left[u_{i,t+1}((b, b_{-i,t+1}); v) \right] \\ \leqslant \;& \eta_t^j \cdot (v - b) \cdot \left(\mathbb{P}(b > \lceil \frac{v_{j,t+1}}{2} \rceil) + \frac{1}{2} \cdot \mathbb{P}(\lceil \frac{v_{j,t+1}}{2} \rceil = b) \right) + (1 - \eta_t^j) \\ = \;& 2\eta_t^j \cdot (v - b) \cdot \left(b - \frac{1}{2H} \right) + (1 - \eta_t^j) \end{aligned}$$

Combining the above lower bound and upper bound of the expected utility of bidder i, we have for any $b \leqslant \frac{1}{2}$,

$$\begin{aligned}
&\mathbf{E}_{b_{-i,t+1}}\left[u_{i,t+1}((\lceil\frac{v}{2}\rceil, b_{-i,t+1}); v) - u_{i,t+1}((b, b_{-i,t+1}); v)\right] \\
\geqslant\ & 2\eta_t^j \cdot \left((v - \lceil\frac{v}{2}\rceil) \cdot \left(\lceil\frac{v}{2}\rceil - \frac{1}{2H}\right) - (v-b)\cdot\left(b - \frac{1}{2H}\right)\right) - (1-\eta_t^j) \\
\geqslant\ & 2(1-H\gamma_t)\frac{1}{2H^2} - H\gamma_t = (1-H\gamma_t)\frac{1}{H^2} - H\gamma_t,
\end{aligned}$$

where the last inequality is based on the same argument as in Claim C.1. Finally, since T_0 is large enough to make $\gamma_t \leqslant \frac{1}{4H^3}$, then we have $(1-H\gamma_t)\frac{1}{H^2} - H\gamma_t \geqslant \frac{3}{4}\cdot\frac{1}{H^2} - \frac{1}{4H^2} = \frac{1}{2H^2}$. Thus, for any bid $b \leqslant \frac{1}{2}$, we have,

$$\mathbf{E}_{b_{-i,t+1}}\left[u_{i,t+1}((\lceil\frac{v}{2}\rceil, b_{-i,t+1}); v) - u_{i,t+1}((b, b_{-i,t+1}); v)\right] \geqslant \frac{1}{H}$$

For bid $b \geqslant \frac{1}{2} + \frac{1}{H}$, $\mathbb{P}(b > \lceil\frac{v_{j,t+1}}{2}\rceil) = 1$ and $\mathbb{P}(\lceil\frac{v_{j,t+1}}{2}\rceil = b) = 0$, then it is trivially to show for any $v \in V$,

$$\mathbf{E}_{b_{-i,t+1}}\left[u_{i,t+1}((\frac{1}{2}, b_{-i,t+1}); v) - u_{i,t+1}((b, b_{-i,t+1}); v)\right] \geqslant \frac{1}{H}$$

Combining the case that $b \leqslant \frac{1}{2}$, we complete the proof. □

Similarly to the proof of Theorem 4.1, we divide the time steps $t > T_0$ to several episodes as follows, $\Gamma_1 = [T_0+1, T_1], \Gamma_2 = [T_1+1, T_2], \ldots$, such that $\forall k \geqslant 1$, $T_k = \left\lfloor \frac{\left(\frac{1}{4H^2}+1\right)T_{k-1}}{\gamma_{T_k}+1} \right\rfloor$. We always choose the smallest T_k to satisfy this condition. This T_k always exists since $(\gamma_t+1)t \to \infty$ as $t \to \infty$. The total time steps of each episode $|\Gamma_k| = T_k - T_{k-1}, \forall k \geqslant 1$. Then we show the following claim holds.

Claim C.4. *Let event $\mathcal{E}_k$ be $\sum_{s \leqslant T_k} u_{i,s}((\lceil\frac{v}{2}\rceil, b_{-i,s}); v) - u_{i,s}((b, b_{-i,s}); v) \geqslant \frac{T_k}{4H^2}$ holds for all i, any fixed v, and any bid $b \neq \lceil\frac{v}{2}\rceil$. Then the event $\mathcal{E}_k$ holds with probability at least $1 - \sum_{\ell=0}^{k}\exp\left(-\frac{|\Gamma_\ell|}{32H^4}\right)$.*

We prove the above claim by induction. If $k = 0$, the claim holds by Claim C.2. We assume the claim holds for k, then we argue the claim still holds for $k+1$. We consider any time

$t \in \Gamma_{k+1}$, given event $\mathcal{E}_k$ holds, we have

$$\begin{aligned}
\sum_{s \leqslant t} u_{i,s}((\lceil \frac{v}{2} \rceil, b_{-i,s}); v) - u_{i,s}((b, b_{-i,s}); v) \geqslant & \sum_{s \leqslant T_k} u_{i,s}((\lceil \frac{v}{2} \rceil, b_{-i,s}); v) - u_{i,s}((b, b_{-i,s}); v) - (t - T_k) \\
\geqslant & \frac{T_k}{4H^2} - T_{k+1} + T_k = \left(\frac{1}{4H^2} + 1\right) T_k - T_{k+1} \geqslant \gamma_t t,
\end{aligned} \tag{C.1}$$

where the first inequality holds because $u_{i,s}((\lceil \frac{v}{2} \rceil, b_{-i,s}); v) - u_{i,s}((b, b_{-i,s}); v) \geqslant -1, \forall s > T_k$ and the final inequality holds because of the induction assumption and the last inequality hold because

$$\gamma_t t + T_{k+1} \leqslant (\gamma_{T_{k+1}} + 1) T_{k+1} = (\gamma_{T_{k+1}} + 1) \cdot \left\lfloor \frac{\left(\frac{1}{4H^2} + 1\right) T_k}{\gamma_{T_{k+1}} + 1} \right\rfloor \leqslant \left(\frac{1}{4H^2} + 1\right) T_k, \forall t \in \Gamma_{k+1}.$$

Then by Claim C.3, given $\mathcal{E}_k$ holds, for any $t \in \Gamma_{k+1}$ we have, $\mathbf{E}\left[u_{i,t}((\lceil \frac{v}{2} \rceil, b_{-i,t}); v) - u_{i,t}((b, b_{-i,t}); v) \middle| \mathcal{E}_k\right] \geqslant \frac{1}{2H^2}$ for any $t \in \Gamma_{k+1}$. By Azuma's inequality (for martingale), we have

$$\begin{aligned}
& \mathbb{P}\left(\sum_{s \in \Gamma_{k+1}} u_{i,s}((\lceil \frac{v}{2} \rceil, b_{-i,s}); v) - u_{i,s}((b, b_{-i,s}); v) \leqslant \frac{|\Gamma_{k+1}|}{4H^2} \middle| \mathcal{E}_k\right) \\
\leqslant \; & \mathbb{P}\left(\sum_{s \in \Gamma_{k+1}} u_{i,s}((\lceil \frac{v}{2} \rceil, b_{-i,s}); v) - u_{i,s}((b, b_{-i,s}); v) \leqslant \sum_{s \in \Gamma_{k+1}} \mathbf{E}\left[u_{i,s}((\lceil \frac{v}{2} \rceil, b_{-i,s}); v) - u_{i,s}((b, b_{-i,s}); v) \middle| \mathcal{E}_k\right] - \frac{|\Gamma_{k+1}|}{4H^2} \middle| \mathcal{E}_k\right) \\
\leqslant \; & \exp\left(-\frac{|\Gamma_{k+1}|}{32H^4}\right)
\end{aligned}$$

Therefore, the event $\mathcal{E}_{k+1}$ holds with probability at least

$$\left(1 - \exp\left(-\frac{|\Gamma_{k+1}|}{32H^4}\right)\right) \cdot \mathbb{P}(\mathcal{E}_k) \geqslant 1 - \sum_{\ell=0}^{k+1} \exp\left(-\frac{|\Gamma_\ell|}{32H^4}\right),$$

which completes the induction. Given Claim 4.2, we have the following argument,

For any time $t > T_0$, there exists $k(t)$, s.t., $t \in \Gamma_{k(t)}$, if the event $\mathcal{E}_{k(t)}$ happens, the bidder i will report $b_t = \lceil \frac{v_{i,t}}{2} \rceil$ at least $1 - H\gamma_t$, by the definition of γ_t-mean-based learning algorithms and the same argument as Eq. (C.1). Therefore, at any time $t > T_0$, each bidder i will report truthfully with probability at least

$$1 - H\gamma_t - \sum_{\ell=0}^{k(t)} \exp\left(-\frac{|\Gamma_\ell|}{32H^4}\right)$$

$$= \quad 1 - H\gamma_t - \sum_{\ell=0}^{k(t)} \exp\left(-\frac{|\Gamma_\ell|}{32H^4}\right)$$

We then bound $k(t)$. First, $T_k \geqslant \left(\frac{4H^3+H}{4H^3+1}\right) T_{k-1}$, since $\gamma_t \leqslant \frac{1}{4H^3}, \forall t > T_0$. Therefore, we have $\left(\frac{4H^3+H}{4H^3+1}\right)^{(k(t)-1)} T_0 \leqslant t$, which implies, $k(t) + 1 \leqslant 2 + \frac{\log(t/T_0)}{\log\left(\frac{4H^3+H}{4H^3+1}\right)} \leqslant \frac{\log t}{\log\left(\frac{4H^3+H}{4H^3+1}\right)}$. In addition, we have $|\Gamma_\ell| \geqslant \frac{H-1}{4H^3+1} T_0$. Combining the above arguments together, we complete the proof.

Remark. Let $p_i(t)$ be the probability that each bidder i bids $\lceil \frac{v_{i,t}}{2} \rceil$ at time t, for any fixed $v_{i,t}$. As long as $T_0 = \Omega(\log \log t)$, $p_i(t) \to 1$ as $t \to \infty$.

C.1.6 Proof of Theorem 4.5

This proof exactly follows the same technique in Theorem 4.1. Here we only mention the difference in multi-position VCG auctions compared with second price auctions, summarized in the following two claims.

Claim C.5. *For any fixed value v, and bid $b \neq v$ and any time $t \leqslant T_0$, for each bidder i, we have*

$$\mathbb{P}\left(u_{i,t}((v, b_{-i,t}); v) - u_{i,t}(b, b_{-i,t}); v) \geqslant \frac{\rho}{H}\right) \geqslant \frac{\tau}{n}$$

Proof. Firstly, truthful bidding is the weakly dominant strategy for each bidder i, thus, for any $b_{-i,t}$ and $b \neq v$, $u_{i,t}((v, b_{-i,t}); v) - u_{i,t}(b, b_{-i,t}); v) \geqslant 0$.

Then we focus on the case that $b \geqslant v + \frac{1}{H}$ here. It is analogous to show for the case $b \leqslant v - \frac{1}{H}$ and we omit here. Consider the other bidders all bid b, since $m < n$, bidder i wins no slot if she bids truthfully, i.e. $u_{i,t}((v, b_{-i,t}); v) = 0$ if $b_{j,t} = b, \forall j \neq i$. However, if bidder i bids b, by random tie-breaking, bidder i will win a slot with probability $1/n$ and pay b if she wins, i.e., $u_{i,t}((b; b_{-i,t}); v) \leqslant \rho(v - b) \leqslant -\frac{\rho}{H}$ if bidder i wins a slot.

Notice, the probability that the other bidders all bid b is $\frac{1}{H^{n-1}}$, therefore

$$\mathbb{P}\left(u_{i,t}((v, b_{-i,t}); v) - u_{i,t}(b, b_{-i,t}); v) \geqslant \frac{\rho}{H}\right) \geqslant \frac{1}{nH^{n-1}} \geqslant \frac{\tau}{n}$$

□

Claim C.6. *For any $t > T_0$, suppose $\sum_{s \leqslant t} u_{i,s}((v, b_{-i,s}); v) - u_{i,s}((b, b_{-i,s}); v) \geqslant \gamma_t t$ holds for any fixed $v, b \neq v$ and each bidder i, then*

$$u_{i,t+1}((v, b_{-i,t+1}); v) - u_{i,t+1}((b, b_{-i,t+1}); v) \geqslant \frac{\rho}{H}$$

holds with probability at least $\frac{\tau}{2n}$, for any fixed value v, bid $b \neq v$ and each bidder i.

Proof. Similarly to Lemma 4.3, by definition of γ_t-mean-based learning algorithm and the condition assumed in the claim, each bidder i will submit bid $b = v_{i,t+1}$ at least $1 - H\gamma_t$. Moreover, by Assumption 4.2, the probability that the k-th largest value from the other bidders is b, is at least τ. Following the same argument in Claim C.5, we have

$$u_{i,t+1}((v, b_{-i,t+1}); v) - u_{i,t+1}((b, b_{-i,t+1}); v) \geqslant \frac{\rho}{H}$$

holds with probability at least $(1 - H\gamma_t)^{n-1}\frac{\tau}{n} \geqslant (1 - H(n-1)\gamma_t)\frac{\tau}{n}$. □

Given the above two claims, following the exactly same proof steps as in Theorem 4.1, we complete the proof for Theorem 4.5.

C.2 Additional Experiments

In this section, we outline the experimental setup for the contextual bandit experiments as well as the deep Q-learning experiments. In the contextual bandit setting, we use the state to simply represent the private value of the user and reward to be the utility of the auction. In the RL setup, we define the observations of each agent as their private valuation, the state transitions as the next private valuation sampled randomly, the rewards as the utility of the auction and the actions as the discrete bid chosen at each round. Note that in the RL setup where the state is the private value and the next private valuation is sampled randomly, the proposed setting would be similar to the contextual bandit setting. Our learning algorithm is the ϵ-greedy which is a Mean-Based Algorithm studied earlier. In the experimental setup, we considered a finite horizon episode consisting of N auction rounds (where N is chosen as 100 typically in the experiments).

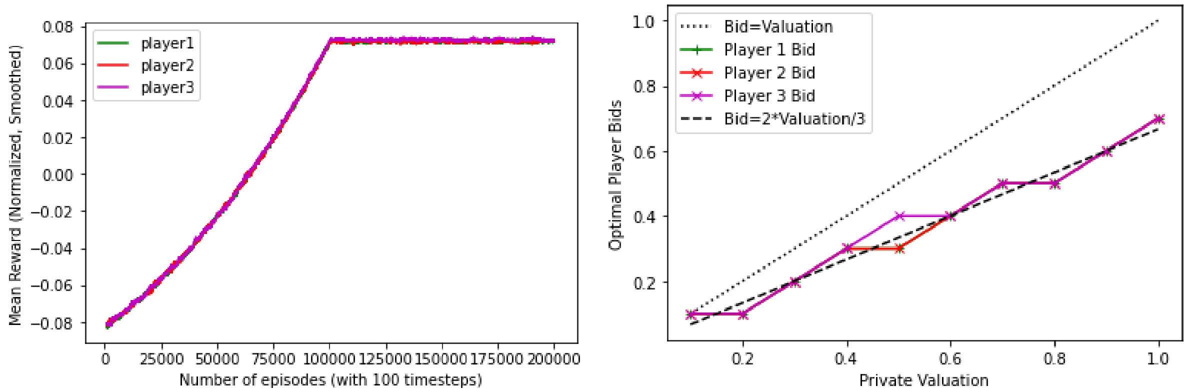


Figure C.1: *Training curve of mean reward of each bidder (left) and roll-out bidding strategy of each bidder (right) in the exploitation phase of contextual ε-Greedy algorithm in first price auctions for three bidders.*

C.2.1 Experiment Details

Contextual Bandits Setting: In first and second price auctions, we anneal the ϵ for both players from 1.0 to 0.05 over 50,000 episodes with 100 rounds each. To measure the robustness of our results for more than two players, we evaluate the equilibrium performance for three agents participating in a first price auction in Figure C.1.

Deep Q Learning Setting: In the DQN experiments mentioned earlier, we used Double Deep Q Networks [VHGS15] with Dueling [Wan+16] and Prioritized Experience Replay [Sch+15] to train the two agents with identical hyperparameters. In the experiments, the Q network is a fully-connected network with hidden dimensions [256, 256] and tanh non-linearity, the number of discrete states $H = 100$, the discount rate was set as $\gamma = 0.99$ and the learning rate $\alpha = 0.0005$. We train over 400,000 time steps with target network updates frequency of $\tau = 500$ steps. The size of the replay memory buffer is 50000 and the minibatch size is 32 (such that 1000 time steps are sampled from the buffer at each train iteration). We use an ϵ greedy policy with annealing wherein ϵ decreases linearly from 1 to 0.02 in 200,000 steps.

To capture the inter-dependence of actions chosen by the bidder, we model the observation for each agent as the current private valuation, the previous valuation, the previous bid and the auction outcome. Like before, we observe that the agents bid approximately with truthful bids for second price auctions and BNE for first price auctions in Figures C.2.

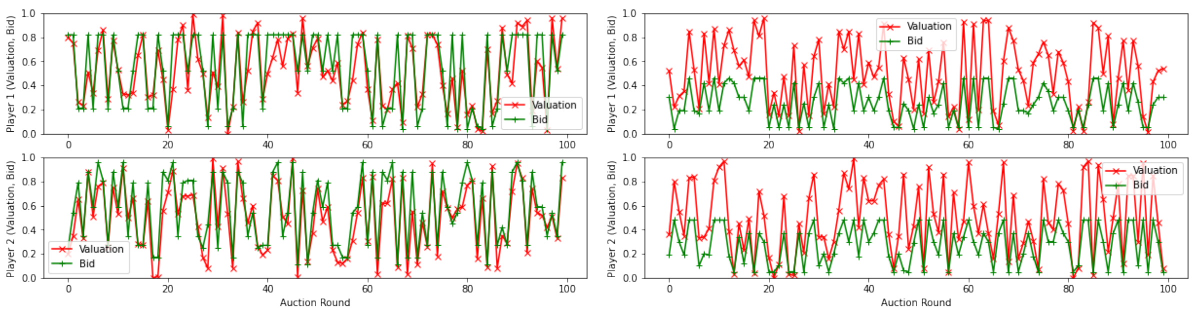


Figure C.2: *The roll-out of the optimal bidding strategy of each bidder with deep state representations in second price auction (left) and first price auction (right).*

Appendix D

Appendix to Chapter 5

D.1 Omitted Algorithms

Essentially, the family of our WIN-EXP algorithms is parametrized by the step-size η-parameter, the estimate of the utility that the learner gets at every timestep $\tilde{u}_t(b)$ and finally, the type of feedback that he receives after every timestep t. Clearly, both η and the estimate of the utility depend crucially on the particular type of feedback.

In this section, we present the specifics of the algorithms that we omitted from the main body of the text.

D.1.1 Outcome-based feedback graph over outcomes

Algorithm 6 WIN-EXP-G algorithm for learning with outcome-based feedback and a feedback graph over outcomes

Let $\pi_1(b) = \frac{1}{|B|}$ for all $b \in B$ (i.e. the uniform distribution over bids), $\eta = \sqrt{\frac{\log(|B|)}{8T\alpha \ln\left(\frac{16|O|^2 T}{\alpha}\right)}}$

for each iteration t **do**

Draw an action $b_t \sim \pi_t(\cdot)$, multinomial

Observe $x_t(\cdot)$, chosen outcome o_t and associated reward function $r_t(\cdot, o_t)$

Observe and associated reward function $r_t(\cdot,\cdot)$ for all neighbor outcomes $N_\varepsilon^{in}, N_\varepsilon^{out}$

Compute estimate of utility:

$$\tilde{u}_t(b) = \mathbb{I}\{o_t \in O_\epsilon\} \sum_{o \in N_\varepsilon^{out}(o_t)} \frac{(r_t(b,o) - 1)\mathbb{P}_t[o|b]}{\sum_{o' \in N_\varepsilon^{in}(o)} \mathbb{P}_t[o']} \tag{D.1}$$

Update $\pi_t(\cdot)$ based on the Exponential Weights Update:

(D.2)

end for

D.2 Omitted proofs from Section 5.4

We first give a lemma that bounds the moments of our utility estimate.

Lemma D.1. *At each iteration t, for any action $b \in B$, the random variable $\tilde{u}_t(b)$ is an unbiased estimate of the true expected utility $u_t(b)$, i.e.: $\forall b \in B : \mathbf{E}[\tilde{u}_t(b)] = u_t(b) - 1$ and has expected second moment bounded by: $\forall b \in B : \mathbf{E}\left[(\tilde{u}_t(b))^2\right] \leqslant 4 \sum_{o \in O} \frac{\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]}$.*

Proof of Lemma D.1. According to the notation we introduced before we have:

$$\begin{aligned}
\mathbf{E}[\tilde{u}_t(b)] &= \mathbf{E}_{o_t}\left[\frac{(r_t(b,o_t) - 1)\cdot \mathbb{P}_t[o_t|b]}{\mathbb{P}_t[o_t]}\right] = \sum_{o \in O} \frac{(r_t(b,o) - 1)\cdot \mathbb{P}_t[o|b]}{\mathbb{P}_t[o]} \mathbb{P}_t[o] \\
&= \sum_{o \in O} r_t(b,o)\mathbb{P}_t[o|b] - 1 = u_t(b) - 1
\end{aligned}$$

Similarly for the second moment:

$$\mathbf{E}\left[\tilde{u}_t(b)^2\right] \leqslant \mathbf{E}_{o_t}\left[\frac{(r_t(b,o_t)-1)^2\mathbb{P}_t[o_t|b]^2}{\mathbb{P}_t[o_t]^2}\right] = \sum_{o\in O}\frac{(r_t(b,o)-1)^2\mathbb{P}_t[o|b]^2}{\mathbb{P}_t[o]^2}\mathbb{P}_t[o]$$
$$\leqslant \sum_{o\in O}\frac{4\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]}$$

where the last inequality holds since $r_t(\cdot,\cdot)\in[-1,1]$. □

Proof of Theorem 5.2. Observe that regret with respect to utilities $u_t(\cdot)$ is equal to regret with respect to the translated utilities $u_t(\cdot)-1$. We use the fact that the exponential weight updates with an unbiased estimate $\tilde{u}_t(\cdot)\leqslant 0$ of the true utilities, achieves expected regret of the form:

$$R(T) \leqslant \frac{\eta}{2}\sum_{t=1}^{T}\sum_{b\in B}\pi_t(b)\cdot\mathbf{E}\left[(\tilde{u}_t(b))^2\right] + \frac{1}{\eta}\log(|B|)$$

For a detailed proof of the above, we refer the reader to Appendix D.7. Invoking the bound on the second moment by Lemma D.1, we get:

$$\begin{aligned} R(T) &\leqslant 2\eta\sum_{t=1}^{T}\sum_{b\in B}\pi_t(b)\cdot\sum_{o\in O}\frac{\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]} + \frac{1}{\eta}\log(|B|) \\ &= 2\eta\sum_{t=1}^{T}\sum_{o\in O}\sum_{b\in B}\pi_t(b)\cdot\frac{\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]} + \frac{1}{\eta}\log(|B|) \\ &\leqslant 2\eta T|O| + \frac{1}{\eta}\log(|B|) \end{aligned}$$

Picking $\eta=\sqrt{\frac{\log(|B|)}{2T|O|}}$, we get the theorem. □

D.2.1 Comparison with Results in [WPR16].

We note that our result in Example 5.1 also *recovers* the results of Weed, Perchet, and Rigollet [WPR16], who work in the continuous bid setting (i.e. $b\in[0,1]$). In order to describe their results, consider the grid $\mathcal{L}_T$ formed by the maximum bids from other bidders $m_t=\max_{j\neq i} b_{jt}$ for all the rounds. Let $l^o=(m_t,m_{t'})$ be the widest interval in $\mathcal{L}_T$, that contains an optimal fixed bid in hindsight and let Δ^o denote its length. Weed, Perchet, and Rigollet [WPR16] provide an algorithm for learning the valuation, which yileds regret $4\sqrt{T\log(1/\Delta^o)}$.

The same regret can be achieved, if we simply consider a partition of the bidding space

$[0,1]$ into $\frac{1}{\epsilon}$ intervals of equal length ϵ, for $\epsilon < \Delta^o$, and run our algorithm on this discretized bid space B. If l^o contains an optimal bid, then any bid $b \in l^o$ is also optimal in-hindsight, since all such bids achieve the same utility. Since $\Delta^o > \epsilon$, there must exist a discretized bid $b^*_\epsilon \in B \cap l^o$. Thus, b^*_ϵ is also optimal in hindsight. Hence, regret against the best fixed bid in $[0,1]$ is equal to regret against the best fixed discretized bid in B. By our Theorem 5.2, the latter regret is $4\sqrt{T\log(1/\epsilon)}$, which can be made arbitrarily close to the regret bound achieved by Weed, Perchet, and Rigollet [WPR16], who use a more intricate adaptive discretization. Similar to Weed, Perchet, and Rigollet [WPR16], knowledge of Δ^o can be bypassed by instead defining Δ^o as the length of the smallest interval in $\mathcal{L}_T$ and then using the standard doubling trick, i.e.: keep an estimate of Δ^o and once this estimate is violated, divide Δ^o in half and re-start your algorithm. The latter only increases the regret by a constant factor.

D.3 Notes on Subsection 5.4.1

If one is interested in optimizing the *sum* of utilities at each iteration rather than the *average*, then if all iterations have the same number of batches $|I|$, this simply amounts to rescaling everything by $|I|$, which would lead to an $|I|$ blow up in the regret.

If different periods have different number of batches and $I_{\max}$ is the maximum number of batches per iteration, then we can always pad the extra batches with all zero rewards. This would amount to again multiplying the regret by $I_{\max}$ and would change the unbiased estimates at each period to be scaled by the number of iterations in that period:

$$\tilde{u}_t(b) = \frac{|I_t|}{I_{\max}} \sum_{o \in O} \frac{\mathbb{P}_t[o|b] \cdot \mathbb{P}_t[o|b_t]}{\mathbb{P}_t[o]} \left(Q_t(b,o) - 1\right) \tag{D.3}$$

and then we would invoke the same algorithm. This essentially puts more weight on iterations with more auctions, so that the "step-size" of the algorithm depends on how many auctions were run during that period. It is easy to see that the latter modification would lead to regret $4I_{\max}\sqrt{T\log\left(|B|\right)}$ in the sponsored search auction application.

D.4 Omitted Proofs from Section 5.4.1

We first prove an upper bound on the moments of our estimates used in the case of batch rewards.

Lemma D.2. *At each iteration t, for any action $b \in B$, the random variable $\tilde{u}_t(b)$ is an unbiased estimate of $u_t(b) - 1$ and can actually be constructed based on the feedback that the learner receives:* $\forall b \in B : \tilde{u}_t(b) = \sum_{o \in O} \frac{\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]} f_t(o) \left(Q_t(b,o) - 1\right)$ *and has expected second moment bounded by:* $\forall b \in B : \mathbf{E}\left[(\tilde{u}_t(b))^2\right] \leqslant 4 \sum_{o \in O} \frac{\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]}$.

Proof of Lemma D.2. For the estimate of the utility it holds that:

$$\begin{aligned}
\tilde{u}_t(b) &= \frac{1}{|I_t|} \sum_{\tau \in I_t} \frac{(r_\tau(b, o_\tau) - 1)\mathbb{P}_t[o_\tau|b]}{\mathbb{P}_t[o_\tau]} \\
&= \frac{1}{|I_t|} \sum_{o \in O} \sum_{\tau \in I_{to}} \frac{(r_\tau(b,o) - 1)\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]} \\
&= \sum_{o \in O : |I_{to}| > 0} \frac{\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]} f_t(o) \frac{1}{|I_{to}|} \sum_{\tau \in I_{to}} (r_\tau(b,o) - 1) \\
&= \sum_{o \in O} \frac{\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]} f_t(o) \left(Q_t(b,o) - 1\right)
\end{aligned} \tag{D.4}$$

From the first equation it follows along identical lines, that this is an unbiased estimate, while from the last equation it is easy to see that this unbiased estimate can be constructed based on the feedback that the learner receives.

Moreover, we can also bound the second moment of these estimates by a similar quantity as in the previous section:

$$\begin{aligned}
\mathbf{E}[\tilde{u}_t(b)^2] &= \sum_{b_t \in B} \mathbf{E}\left[\left(\sum_{o \in O} \frac{\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]} f_t(o) \left(Q_t(b,o) - 1\right) \right)^2 \middle| b_t \right] \pi_t(b_t) \\
&\leqslant \sum_{b_t \in B} \mathbf{E}\left[\sum_{o \in O} \left(\frac{\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]} \left(Q_t(b,o) - 1\right) \right)^2 f_t(o) \middle| b_t \right] \pi_t(b_t) \qquad \text{(By Jensen's inequality)} \\
&= \sum_{b_t \in B} \sum_{o \in O} \left(\frac{\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]} \left(Q_t(b,o) - 1\right) \right)^2 \mathbf{E}[f_t(o)|b_t] \cdot \pi_t(b_t) \\
&= \sum_{o \in O} \left(\frac{\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]} \left(Q_t(b,o) - 1\right) \right)^2 \sum_{b_t \in B} \mathbf{E}[f_t(o)|b_t] \cdot \pi_t(b_t)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{o \in O} \left(\frac{\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]} \left(Q_t(b,o) - 1\right) \right)^2 \mathbb{P}_t[o] \\
&\leqslant 4 \sum_{o \in O} \frac{\mathbb{P}_t[o|b]}{\mathbb{P}_t[o]}
\end{aligned}
$$

□

Then following the same techniques in Theorem 5.2, it is straightforward to conclude the proof of the corollary.

D.5 Omitted Proofs from Section 5.5

Proof of Lemma 5.3. Let OPT $= \arg\sup_{b \in \mathcal{B}} \sum_{t=1}^{T} u_t(b)$ be the best fixed action in the continuous action space $\mathcal{B}$ in hindsight. Since $\varepsilon < \Delta^o$, then b^* must belong to some d-dimensional ε-cube, either as an interior point or as a limit of interior points, as expressed by Definition 5.1. The utility is L-Lipschitz within this ε-cube and since $\epsilon < \Delta^o$, each cube contains at least one point in the discretized space B. For the case where OPT is achieved as the limit of interior points then for every $\delta > 0$ there exist an interior point of some cube $\tilde{b}$, such that $\sum_{t=1}^{T} u_t(\tilde{b}) \geqslant \text{OPT} - \delta$. The same obviously holds when OPT is achieved by an interior point. Let $\widehat{b}$ be the closest discretized point to $\tilde{b}$ that lies in the same cube as $\tilde{b}$. Since $\|\widehat{b} - \tilde{b}\|_\infty \leqslant \epsilon$, by the Lipschitzness of the average reward function within each cube, we get:

$$
\text{OPT} \leqslant \sum_{t=1}^{T} u_t(\tilde{b}) + \delta \leqslant \sum_{t=1}^{T} u_t(\widehat{b}) + \delta + \epsilon LT \leqslant \sup_{b \in B} \sum_{t=1}^{T} u_t(\widehat{b}) + \delta + \epsilon LT
$$

Since we can take δ as close to zero as we want, we get the lemma. □

Proof of Theorem 5.4. From Lemma 5.3 we know that for $\varepsilon < \Delta^o$, the discretization error is $DE(B, \mathcal{B}) \leqslant \epsilon LT$. Combining Lemma 5.2 and Corollary 5.3, we have

$$
\begin{aligned}
R(T, \mathcal{B}) \leqslant R(T, B) + DE(B, \mathcal{B}) &= 2\sqrt{2T|O| \log(|B|)} + \varepsilon LT \\
&= 2\sqrt{2T|O| \log\left(\frac{1}{\varepsilon^d}\right)} + \varepsilon LT \\
&= 2\sqrt{2dT|O| \log\left(\frac{1}{\varepsilon}\right)} + \varepsilon LT
\end{aligned}
$$

$$= 2\sqrt{2dT|O|\log\left(\max\left\{LT, \frac{1}{\Delta^o}\right\}\right)} + \min\left\{\frac{1}{LT}, \Delta^o\right\}$$
$$\leqslant 2\sqrt{2dT|O|\log\left(\max\left\{LT, \frac{1}{\Delta^o}\right\}\right)} + 1$$

□

Unknown Lipschitzness constant. In Theorem 5.4 the discretization parameter ε depends on the prior knowledge of the Lipschitzness constant, L, the number of rounds, T and the minimum edge length of each d-dimensional cube, Δ^o. In order to address the problem that in general we do not know any of those constants a priori, we will apply a standard doubling trick ([Aue+02]) to remove this dependence. We assume that T is upper bounded by a constant T_M and similarly we also assume that $\log\left(\max\left\{LT, \frac{1}{\Delta^o}\right\}\right)$ is upper bounded by a constant.

We will then initialize two bounds: $B_T = 1$ and $B_{\Delta^o,LT} = 1$ and run the WIN-EXP algorithm with step size $\sqrt{\frac{\log(1/\varepsilon)}{2B_T|O|}}$ and $\varepsilon = \min\left\{\frac{1}{LT}, \Delta^o\right\}$ until $t \leqslant B_T$ or $\log\left(\max\left\{tL, \frac{1}{\Delta^o}\right\}\right) \leqslant B_{\Delta^o,LT}$ fails to hold. If one of these discriminants fails, then we double the bound and restart the algorithm. This modified strategy only increases the regret by a constant factor.

Corollary D.1. *The* WIN-EXP *algorithm run with the above doubling trick achieves an expected regret bound* $\mathcal{R}(T) \leqslant 25\sqrt{2dT|O|\log\left(\max\left\{LT, \frac{1}{\Delta^o}\right\}\right)} + 1$

Proof of Corollary D.1. Based on the doubling trick that we described above, we divide the algorithm into stages in which B_T and $B_{\Delta^o,LT}$ are constants. Let B_L^*, and $B^*_{\Delta^o,LT}$ be the values of B_L and $B_{\Delta,LT}$ respectively when the algorithm terminates. There is at most a total of $\log\left(B_T^*\right) + \log\left(B^*_{\Delta^o,LT}\right) + 1$ stages in this doubling process. Since the actual expected regret is bounded by the sum of the regret of each stage, following the result of Theorem 5.4, we have

$$R(T) \leqslant \sum_{i=0}^{\lceil \log(B_T^*)\rceil} \sum_{j=0}^{\left\lceil \log\left(B^*_{\Delta^o,LT}\right)\right\rceil} \left(2\sqrt{2d2^i|O|2^j}\right) + \log\left(B_T^*\right) + \log\left(B^*_{\Delta^o,LT}\right) + 1$$
$$= \sum_{i=0}^{\lceil \log(B_T^*)\rceil} \sum_{j=0}^{\left\lceil \log\left(B^*_{\Delta^o,LT}\right)\right\rceil} \left(2\sqrt{2d|O|2^i \cdot 2^j}\right) + \log\left(B_T^* B^*_{\Delta^o,LT}\right) + 1$$

$$
\begin{aligned}
&= \left[\sum_{i=0}^{\lceil \log\left(B_T^*\right)\rceil} \left(\sqrt{2}\right)^i\right] \cdot \left[\sum_{j=0}^{\left\lceil \log\left(B_{\Delta^o,LT}^*\right)\right\rceil} \left(\sqrt{2}\right)^j\right] 2\sqrt{2d|O|} + \log\left(B_T^* B_{LT,\Delta^o}^*\right) + 1 \\
&= \frac{1-\sqrt{2}^{\lceil \log(B_T^*)\rceil+1}}{1-\sqrt{2}} \cdot \frac{1-\sqrt{2}^{\lceil \log(B_{\Delta^o,LT}^*)\rceil+1}}{1-\sqrt{2}} \cdot 2\sqrt{2d|O|} + \log\left(B_T^* B_{\Delta^o,LT}^*\right) + 1 \\
&\leqslant \left(\frac{\sqrt{2}}{\sqrt{2}-1}\right)^2 \sqrt{B_T^* B_{\Delta^o,LT}^*} \cdot 2\sqrt{2d|O|} + \log\left(B_T^* B_{\Delta^o,LT}^*\right) + 1 \\
&= \left(\frac{\sqrt{2}}{\sqrt{2}-1}\right)^2 \cdot 2\sqrt{2d|O|B_T^* B_{\Delta^o,LT}^*} + \log\left(B_T^* B_{\Delta^o,LT}^*\right) + 1 \\
&\leqslant 25\sqrt{2d|O|B_T^* B_{\Delta^o,LT}^*} + 1
\end{aligned}
$$

Combining the fact that $B_T^* \leqslant T$ and $B_{\Delta^o,LT}^* \leqslant \log\left(\max\left\{LT, \frac{1}{\Delta^o}\right\}\right)$ as well as the above inequalities, we complete the proof. □

D.5.1 Omitted Proofs from Section 5.5.1

Proof of Theorem 5.5. Consider a player i. Observe that conditional on the player's score s_i, his utility remains constant if he is allocated the same slot. Moreover, when the slots are different, then the difference in utilities is at most 2, since utilities lie in $[-1, 1]$. Moreover, because the slots are allocated in decreasing order of rank scores, the slot allocation of a player is different under b_i and b_i' only if there exists a player j, who passes the rank-score reserve (i.e. $s_j \cdot b_j \geqslant r$) and whose rank-score $s_j \cdot b_j$ lies in the interval $[s_i \cdot b_i, s_i \cdot b_i']$. Hence, conditional on s_i, the absolute difference between the player's expected utility when he bids b_i and when he bids $b_i + \varepsilon$, with $\varepsilon > 0$, is upper bounded by:

$$
2 \cdot \mathbb{P}\left[\exists j \neq i \text{ s.t } s_j \cdot b_j \in [s_i \cdot b_i, s_i \cdot (b_i + \varepsilon)] \text{ and } s_j \cdot b_j \geqslant r \mid s_i\right]
$$

By a union bound the latter is at most:

$$
2 \cdot \sum_{j \neq i} \mathbb{P}\left[s_j \in \left[\frac{s_i b_i}{b_j}, \frac{s_i(b_i+\varepsilon)}{b_j}\right] \text{ and } s_j \cdot b_j \geqslant r \mid s_i\right]
$$

Since $s_j \in [0,1]$, the previous quantity is upper bounded by replacing the event $s_j \cdot b_j \geqslant r$ by $b_j \geqslant r$. This event is independent of the scores and we can then write the above bound as:

$$2 \cdot \sum_{j \neq i \text{ s.t. } b_j \geqslant r} \mathbb{P}\left[s_j \in \left[\frac{s_i b_i}{b_j}, \frac{s_i(b_i+\varepsilon)}{b_j} \right] \middle| s_i \right]$$

Since each quality score s_j is drawn independently from an L-Lipschitz CDF F_j, we can further simplify the bound by:

$$2 \cdot \sum_{j \neq i \text{ s.t. } b_j \geqslant r} \left[F_j\left(\frac{s_i(b_i+\varepsilon)}{b_j} \right) - F_j\left(\frac{s_i b_i}{b_j} \right) \right] \leqslant 2 \cdot \sum_{j \neq i \text{ s.t. } b_j \geqslant r} L\frac{s_i \varepsilon}{b_j} \leqslant 2 \cdot \sum_{j \neq i \text{ s.t. } b_j \geqslant r} L\frac{s_i \varepsilon}{r} \leqslant \frac{2nL}{r}\epsilon$$

Since the absolute difference of utilities between these two bids is upper bounded conditional on s_i, by the triangle inequality it is also upper bounded even unconditional on s_i, which leads to the Lipschitz property we want:

$$\left| u_i(b_i, \mathbf{b}_{-i}, r) - u_i(b_i+\epsilon, \mathbf{b}_{-i}, r) \right| \leqslant \frac{2nL}{r}\epsilon \tag{D.5}$$

□

D.6 Omitted proofs from section 5.6.1

D.6.1 Switching Regret and PoA

Proof of Corollary 5.7. We first observe that the results proven in [GLL12] for a prediction algorithm $\mathcal{A}$ with *regret* upper bounded by $\rho(T)$ hold also for algorithms $\mathcal{A}$ for which we know upper bound of their expected regrets. Specifically, if algorithm $\mathcal{A}$ has an upper bound of $\rho(T)$ for its expected regret, where $\rho(T)$ is a concave, non-decreasing, $[0,+\infty) \to [0,+\infty)$ function, then Lemma 1 from [GLL12] holds for *expected* regret. With that in mind, we can directly apply the *Randomized Tracking Algorithm* and get expected switching regret upper bounded by:

$$(C(TP)+1)\, L_{C(TP),T}\,\rho\left(\frac{T}{(C(TP)+1)\, L_{C(TP),T}} \right) + \sum_{t=1}^{T} \frac{\eta_t}{8} + \frac{r_T\left((C(TP)+1)\, L_{C(TP),T-1} - 1 \right)}{\eta_T} \tag{D.6}$$

where TP is the switching path of the optimal bids and $C(TP)$ is the number of switches in the optimal bid according to this path.

We proceed by making sure that the conditions for the upper bound of the expected regret of WIN-EXP satisfy the conditions required by algorithm $\mathcal{A}$ in [GLL12]. Indeed, the upper bound of the expected regret of our algorithm, $\sqrt{2dT|O|\log\left(\max\left\{LT, \frac{1}{\Delta^o}\right\}\right)} + 1$, is non decreasing in T. Also, at timestep $t = 0$, we incur no regret. We also apply the following slight modifications in Algorithm 2 in [GLL12] so as to match the nature of our problem. First, instead of computing the expected loss at each timestep t, we will now compute the expected outcome-based utility, i.e. $\bar{u}_t(\pi_t) = \sum_{b\in B}\pi_t(b)\mathbf{E}_{o_t}[\tilde{u}_t(b)]$. Second, instead of the cumulative loss of their algorithm $\mathcal{A}$ we will now use the cumulative outcome-based expected utility of WIN-EXP, i.e. $\bar{U}_t(\text{WIN-EXP}, T) = \sum_{c=0}^{C}\bar{U}_{\text{WIN-EXP}}(t_c, t_{c+1})$, where

$$\bar{U}_{\text{WIN-EXP}}(t_c, t_{c+1}) = \sum_{s=t_c}^{t_{c+1}-1}\bar{u}_s\left(\pi_{\text{WIN-EXP},s}(t_c)\right)$$

is the cumulative outcome-based expected utility gained from our WIN-EXP algorithm in the time interval $[t_c, t_{c+1})$[1] with respect to $\bar{u}_s$ for $s \in [t_c, t_{c+1})$. Now, we are computing the regret components of [GLL12] so as to achieve the desired result.

Before we show the specifics of the computation, we note here that $g > 0$ is a *parameter* of the Tracking Regret algorithm presented by [GLL12] and can be set a priori from the designer of the algorithm. The complexity of g affects the computational complexity of the algorithm and there is a tradeoff between the computational complexity and the regret of the algorithm. For our computations here, we will set

$$g + 1 = \left(\frac{T}{C(TP)+1}\right)^{\alpha} \tag{D.7}$$

where $0 < \alpha < 1$ is a constant. Now, we are ready to compute the components of the regret:

$$\begin{aligned}
A &= L_{C(TP),T}\left(C(TP)+1\right)R_{\text{WIN-EXP}}\left(\frac{T}{L_{C(TP),T}\left(C(TP)+1\right)}\right)\\
&\leqslant 25\left(\frac{\log\left(\frac{T}{C(TP)+1}\right)}{\log(g+1)}+2\right)\left(C(TP)+1\right)\left(\sqrt{2d|O|\frac{T\log(g+1)\log(m)}{\log\left(\frac{T}{C(TP)+1}\right)+2\log(g+1)}}+1\right)\\
&= 50\cdot\left(2+\frac{1}{\alpha}\right)\cdot\left(C(TP)+1\right)\sqrt{2d|O|\cdot\frac{\alpha}{1+2\alpha}\cdot T\log(m)}
\end{aligned}$$

[1]We clarify here that these time intervals are with respect to the switching bids.

$$\leqslant 50\sqrt{\frac{1+2\alpha}{\alpha}\cdot (C(TP)+1)^2\, 2d|O|T\log(m)}$$
$$\leqslant 50\sqrt{\left(2+\frac{1}{\alpha}\right)\cdot (C+1)^2\, 2d|O|T\log(m)}$$

where in the second equality we have denoted $\log(m) = \log\left(\max\left\{LT, \frac{1}{\Delta^o}\right\}\right)$ and the last inequality comes from the fact that C is the upper bound on the number of switches that the transition path TP can have. Moving on to the computation of the rest of the components of the regret:

$$B = \sum_{t=1}^{T} \frac{\eta_t}{8} \leqslant \frac{1}{8}\sqrt{\frac{T\log\left(1/\varepsilon\right)}{2|O|}} = O\left(\sqrt{\frac{T}{|O|}}\right)$$
$$D = r_T\left(L_{C(TP),T}\left(C(TP)+1\right)-1\right)$$
$$= \left(\frac{\alpha+1}{\alpha}+\varepsilon_2\right)\log T + \log\left(1+\varepsilon_2\right) - \left(\frac{\alpha+1}{\alpha}\right)\log\varepsilon_2$$

where $\varepsilon_2 \in (0,1)$ is a constant. Before we conclude, we observe that even though Corollary 1 of [GLL12] is stated as a high-probability ex post result, the proof uses a result from [CL06] (Lemma 4.1) which also holds for the expected regret. According to [GLL12] the switching regret is the sum of the aforementioned A, B, D. Thus, we get the result. □

D.6.2 Feedback Graphs over Outcomes

We first prove bounds on the moments of our unbiased estimates used in the case of a feedback graph over outcomes.

Lemma D.3. *At each iteration t, for any action $b \in B$, the random variable $\tilde{u}_t(b)$ has bias with respect to $u_t(b)-1$ bounded by: $\left|\mathbf{E}\left[\tilde{u}_t(b)\right] - (u_t(b)-1)\right| \leqslant 2\epsilon|O|$ and has expected second moment bounded by: $\forall b \in B : \mathbf{E}\left[\tilde{u}_t(b)^2\right] \leqslant 4\sum_{o\in O_\epsilon} \frac{\mathbb{P}_t[o|b]}{\sum_{o'\in N_\epsilon^{in}(o)} \mathbb{P}_t[o']}$.*

Proof of Lemma D.3. For the expected utility we have:

$$\mathbf{E}\left[\tilde{u}_t(b)\right] = \mathbf{E}_{o_t}\left[\mathbb{I}\{o_t \in O_\epsilon\} \sum_{o\in N_\epsilon^{out}(o_t)} \frac{(r_t(b,o)-1)\mathbb{P}_t[o|b]}{\sum_{o'\in N_\epsilon^{in}(o)} \mathbb{P}_t[o']}\right]$$
$$= \sum_{o_t\in O_\epsilon} \sum_{o\in N_\epsilon^{out}(o_t)} \frac{(r_t(b,o)-1)\mathbb{P}_t[o|b]}{\sum_{o'\in N_\epsilon^{in}(o)} \mathbb{P}_t[o']}\mathbb{P}_t[o_t]$$

$$
\begin{aligned}
&= \sum_{o \in O_\epsilon} \sum_{o_t \in N_\epsilon^{in}(o)} \frac{(r_t(b,o)-1)\mathbb{P}_t[o|b]}{\sum_{o' \in N_\epsilon^{in}(o)} \mathbb{P}_t[o']} \mathbb{P}_t[o_t] \\
&= \sum_{o \in O_\epsilon} \frac{(r_t(b,o)-1)\mathbb{P}_t[o|b]}{\sum_{o' \in N_\epsilon^{in}(o)} \mathbb{P}_t[o']} \sum_{o_t \in N_\epsilon^{in}(o)} \mathbb{P}_t[o_t] \\
&= \sum_{o \in O_\epsilon} (r_t(b,o)-1)\mathbb{P}_t[o|b] \\
&= \sum_{o \in O} (r_t(b,o)-1)\mathbb{P}_t[o|b] - \sum_{o \notin O_\epsilon} (r_t(b,o)-1)\mathbb{P}_t[o|b] \\
&= u_t(b) - 1 - \sum_{o \notin O_\epsilon} (r_t(b,o)-1)\mathbb{P}_t[o|b]
\end{aligned}
$$

Thus, we get that the bias of $\tilde{u}$ with respect to $u_t - 1$ is bounded by:

$$
\left| \mathbf{E}\left[\tilde{u}_t(b)\right] - (u_t(b) - 1) \right| \leqslant 2\epsilon |O| \tag{D.8}
$$

Similarly for the second moment:

$$
\begin{aligned}
\mathbf{E}\left[\tilde{u}_t(b)^2\right] &\leqslant \mathbf{E}_{o_t}\left[\left(\mathbb{I}\{o_t \in O_\epsilon\} \sum_{o \in N_\epsilon^{out}(o_t)} \frac{(r_t(b,o)-1)\mathbb{P}_t[o|b]}{\sum_{o' \in N_\epsilon^{in}(o)} \mathbb{P}_t[o']} \right)^2 \right] \\
&= \sum_{o_t \in O_\epsilon} \left(\sum_{o \in N_\epsilon^{out}(o_t)} \frac{(r_t(b,o)-1)\mathbb{P}_t[o|b]}{\sum_{o' \in N_\epsilon^{in}(o)} \mathbb{P}_t[o']} \right)^2 \mathbb{P}_t[o_t]
\end{aligned} \tag{D.9}
$$

Observe that the quantity inside the square:

$$
\sum_{o \in N_\epsilon^{out}(o_t)} \frac{(r_t(b,o)-1)}{\sum_{o' \in N_\epsilon^{in}(o)} \mathbb{P}_t[o']} \mathbb{P}_t[o|b]
$$

can be thought of as an expected value of the quantity $\frac{(r_t(b,o)-1)}{\sum_{o' \in N_\epsilon^{in}(o)} \mathbb{P}_t[o']}$, were o is the random variable and is drawn from the distribution of outcomes conditional on a bid b. Thus, by Jensen's inequality, the square of the latter expectation is at most the expectation of the square, i.e.:

$$
\left(\sum_{o \in N_\epsilon^{out}(o_t)} \frac{(r_t(b,o)-1)}{\sum_{o' \in N_\epsilon^{in}(o)} \mathbb{P}_t[o']} \mathbb{P}_t[o|b] \right)^2 \leqslant \sum_{o \in N_\epsilon^{out}(o_t)} \frac{(r_t(b,o)-1)^2}{\left(\sum_{o' \in N_\epsilon^{in}(o)} \mathbb{P}_t[o']\right)^2} \mathbb{P}_t[o|b]
$$

Combining with Equation (D.9), we get:

$$\begin{aligned}
\mathbf{E}\left[\tilde{u}_t(b)^2\right] &\leqslant \sum_{o_t\in O_\epsilon}\sum_{o\in N_\epsilon^{out}(o_t)} \frac{(r_t(b,o)-1)^2}{\left(\sum_{o'\in N_\epsilon^{in}(o)}\mathbb{P}_t[o']\right)^2}\mathbb{P}_t[o|b]\mathbb{P}_t[o_t] \\
&= \sum_{o\in O_\epsilon}\sum_{o_t\in N_\epsilon^{in}(o)} \frac{(r_t(b,o)-1)^2}{\left(\sum_{o'\in N_\epsilon^{in}(o)}\mathbb{P}_t[o']\right)^2}\mathbb{P}_t[o|b]\mathbb{P}_t[o_t] \\
&= \sum_{o\in O_\epsilon} \frac{(r_t(b,o)-1)^2}{\left(\sum_{o'\in N_\epsilon^{in}(o)}\mathbb{P}_t[o']\right)^2}\mathbb{P}_t[o|b]\sum_{o_t\in N_\epsilon^{in}(o)}\mathbb{P}_t[o_t] \\
&= \sum_{o\in O_\epsilon} \frac{(r_t(b,o)-1)^2}{\sum_{o'\in N_\epsilon^{in}(o)}\mathbb{P}_t[o']}\mathbb{P}_t[o|b] \\
&\leqslant 4\sum_{o\in O_\epsilon} \frac{\mathbb{P}_t[o|b]}{\sum_{o'\in N_\epsilon^{in}(o)}\mathbb{P}_t[o']}
\end{aligned}$$

where the last inequality holds since $r_t(\cdot,\cdot)\in[-1,1]$. □

Proof of Theorem 5.8. Observe that regret with respect to utilities $u_t(\cdot)$ is equal to regret with respect to the translated utilities $u_t(\cdot)-1$. We use the fact that the exponential weight updates with an estimate $\tilde{u}_t(\cdot)\leqslant 0$ which has bias with respect to the true utilities, bounded by κ, achieves expected regret of the form:

$$R(T)\leqslant \frac{\eta}{2}\sum_{t=1}^{T}\sum_{b\in B}\pi_t(b)\cdot\mathbf{E}\left[\tilde{u}_t(b)^2\right]+\frac{1}{\eta}\log(|B|)+2\kappa T$$

For the detailed proof of the above claim, please see Appendix D.7. Invoking the bound on the bias and the second moment by Lemma D.3, we get:

$$\begin{aligned}
R(T) &\leqslant 2\eta\sum_{t=1}^{T}\sum_{b\in B}\pi_t(b)\cdot\sum_{o\in O_\epsilon}\frac{\mathbb{P}_t[o|b]}{\sum_{o'\in N_\epsilon^{in}(o)}\mathbb{P}_t[o']}+\frac{1}{\eta}\log(|B|)+4\epsilon|O|T \\
&= 2\eta\sum_{t=1}^{T}\sum_{o\in O_\epsilon}\sum_{b\in B}\pi_t(b)\cdot\frac{\mathbb{P}_t[o|b]}{\sum_{o'\in N_\epsilon^{in}(o)}\mathbb{P}_t[o']}+\frac{1}{\eta}\log(|B|)+4\epsilon|O|T \\
&= 2\eta\sum_{t=1}^{T}\sum_{o\in O_\epsilon}\frac{\mathbb{P}[o]}{\sum_{o'\in N_\epsilon^{in}(o)}\mathbb{P}_t[o']}+\frac{1}{\eta}\log(|B|)+4\epsilon|O|T
\end{aligned}$$

We can now invoke Lemma 5 of [Alo+15], which states that:

Lemma D.4 ([Alo+15]). *Let $G=(V,E)$ be a directed graph with $|V|=K$, in which each node $i\in V$ is assigned a positive weight w_i. Assume that $\sum_{i\in V}w_i\leqslant 1$, and that $w_i\geqslant\epsilon$ for all $i\in V$ for some* *constant* $0 < \epsilon < 1/2$. *Then*

$$\sum_{i \in V} \frac{w_i}{\sum_{j \in N^{in}(i)} w_j} \leqslant 4\alpha \ln \frac{4K}{\alpha\epsilon} \tag{D.10}$$

where neighborhoods include self-loops and α *is the independence number of the graph.*

Invoking the above lemma for the feedback graph G_ϵ (and noting that the independence number cannot increase by restricting to a sub-graph), we get:

$$\sum_{o \in O_\epsilon} \frac{\mathbb{P}[o]}{\sum_{o' \in N_\epsilon^{in}(o)} \mathbb{P}_t[o']} \leqslant 4\alpha \ln \frac{4|O|}{\alpha\epsilon} \tag{D.11}$$

Thus, we get a bound on the regret of:

$$R(T) \leqslant 8\eta\alpha \ln\left(\frac{4|O|}{\alpha\epsilon}\right) T + \frac{1}{\eta}\log(|B|) + 4\epsilon|O|T$$

Picking $\epsilon = \frac{1}{4|O|T}$, we get:

$$R(T) \leqslant 8\eta\alpha \ln\left(\frac{16|O|^2 T}{\alpha}\right) T + \frac{1}{\eta}\log(|B|) + 1$$

Picking $\eta = \sqrt{\frac{\log(|B|)}{8T\alpha \ln\left(\frac{16|O|^2 T}{\alpha}\right)}}$, we get the theorem. □

D.7 Omitted proof for the regret of the exponential weights update

Lemma D.5. *The exponential weights update with an estimate* $\tilde{u}_t(\cdot) \leqslant 0$ *such that for any* $b \in B$ *and* t, $|\mathbf{E}[\tilde{u}_t(b)] - (u_t(b) - 1)| \leqslant \kappa$, *achieves expected regret on the form:*

$$R(T) \leqslant \frac{\eta}{2}\sum_{t=1}^{T}\sum_{b \in B} \pi_t(b) \cdot \mathbf{E}\left[\tilde{u}_t(b)^2\right] + \frac{1}{\eta}\log(|B|) + 2\kappa T$$

Proof. Following the standard analysis of the exponential weight updates algorithm [AHK12] and the fact that $\forall x \leqslant 0$, $e^x \leqslant 1 + x + \frac{x^2}{2}$ as well as let $b^* = \arg\max_{b \in B} \mathbf{E}\left[\sum_{t=1}^T u_t(b)\right]$, we have

$$\begin{aligned}
\mathbf{E}\left[\sum_{t=1}^{T} \tilde{u}_t(b^*)\right] &\leqslant \sum_{t=1}^{T}\sum_{b \in B} \pi_t(b)\mathbf{E}[\tilde{u}_t(b)] + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{b \in B} \pi_t(b) \cdot \mathbf{E}\left[\tilde{u}_t(b)^2\right] + \frac{1}{\eta}\log(|B|) \\
&\leqslant \sum_{t=1}^{T}\sum_{b \in B} \pi_t(b)(u_t(b) - 1 + \kappa) + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{b \in B} \pi_t(b) \cdot \mathbf{E}\left[\tilde{u}_t(b)^2\right] + \frac{1}{\eta}\log(|B|) \\
&= \mathbf{E}\left[\sum_{t=1}^{T} u_t(b_t)\right] + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{b \in B} \pi_t(b) \cdot \mathbf{E}\left[\tilde{u}_t(b)^2\right] + \frac{1}{\eta}\log(|B|) + \kappa T - T
\end{aligned}$$

which implies that

$$\begin{aligned} R(T) &= \mathbf{E}\left[\sum_{t=1}^{T} u_t(b^*)\right] - \mathbf{E}\left[\sum_{t=1}^{T} u_t(b_t)\right] \leqslant \mathbf{E}\left[\sum_{t=1}^{T} \tilde{u}_t(b^*)\right] - \mathbf{E}\left[\sum_{t=1}^{T} u_t(b_t)\right] + \kappa T + T \\ &\leqslant \frac{\eta}{2}\sum_{t=1}^{T}\sum_{b\in B} \pi_t(b)\cdot \mathbf{E}\left[\tilde{u}_t(b)^2\right] + \frac{1}{\eta}\log(|B|) + 2\kappa T \end{aligned}$$

□

Remark. Let the estimator $\tilde{u}_t(b)$ be unbiased for any t and any $b \in B$, then the expected regret is

$$R(T) \leqslant \frac{\eta}{2}\sum_{t=1}^{T}\sum_{b\in B} \pi_t(b)\cdot \mathbf{E}\left[\tilde{u}_t(b)^2\right] + \frac{1}{\eta}\log(|B|)$$