Predicting Real-World Penny Auction Durations by Integrating Game Theory and Machine Learning

Yujia Wang, Haoran Yu

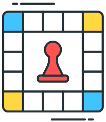
School of Computer Science & Technology, Beijing Insititute of Technology
yjwang.yoga@gmail.com, yhrhawk@gmail.com

March 10, 2024

1. BACKGROUND
2. PROBLEM
3. SOLUTION
4. EXPERIMENTS
5. CONCLUSION

Strategic Behavior Prediction: Introduction

- **Strategic behavior:** Human behavior in strategic environments:
  - Bidding in auctions.
  - Offering in bargainings.
  - Actions in card/board games.



Strategic Behavior Prediction: Conventional Approach

- **Game theory approach**
  - **Steps**: (i) equilibrium assumption; (ii) predict behavior using game theory models.

Strategic Behavior Prediction: Conventional Approach

- **Game theory approach**
  - **Steps**: (i) equilibrium assumption; (ii) predict behavior using game theory models.
  - **Cons**: (i) strong assumptions; (ii) inaccurate prediction.

Strategic Behavior Prediction: Conventional Approach

- **Game theory approach**
  - **Steps**: (i) equilibrium assumption; (ii) predict behavior using game theory models.
  - **Cons**: (i) strong assumptions; (ii) inaccurate prediction.
- **Machine learning approach**
  - **Steps**: (i) train machine learning models on historical behavior data; (ii) predict behavior with the trained models.

Strategic Behavior Prediction: Conventional Approach

- **Game theory approach**
  - **Steps**: (i) equilibrium assumption; (ii) predict behavior using game theory models.
  - **Cons**: (i) strong assumptions; (ii) inaccurate prediction.
- **Machine learning approach**
  - **Steps**: (i) train machine learning models on historical behavior data; (ii) predict behavior with the trained models.
  - **Pros**: (i) fewer assumptions (ii) accurate prediction.

Output = f(Input)

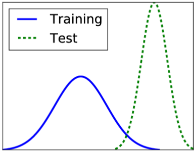
Mapping

Strategic Behavior Prediction: Conventional Approach

- **Game theory approach**
  - **Steps**: (i) equilibrium assumption; (ii) predict behavior using game theory models.
  - **Cons**: (i) strong assumptions; (ii) inaccurate prediction.
- **Machine learning approach**
  - **Steps**: (i) train machine learning models on historical behavior data; (ii) predict behavior with the trained models.
  - **Pros**: (i) fewer assumptions (ii) accurate prediction.
  - **Cons**: the domain shift problem.

Output = f(Input)

Mapping



Glauner et al. (2018)

Strategic Behavior Prediction: Motivation

- Due to bounded rationalities, a **gap** exists between the game-theoretic prediction and real human behavior.

Game theory model

Human behavior

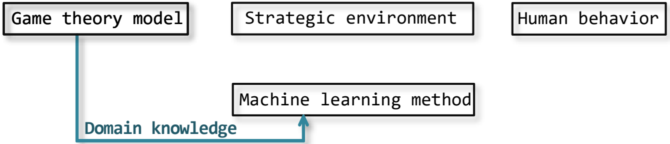
Strategic Behavior Prediction: Motivation

- Due to bounded rationalities, a **gap** exists between the game-theoretic prediction and real human behavior.
  - The gap can be highly correlated with the strategic environment.

`Game theory model` `Strategic environment` `Human behavior`

Strategic Behavior Prediction: Motivation

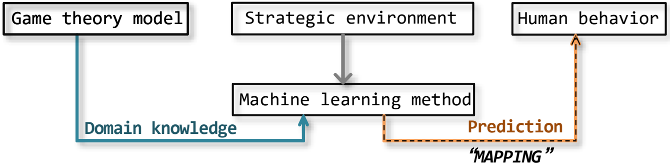
- Due to bounded rationalities, a **gap** exists between the game-theoretic prediction and real human behavior.
  - The gap can be highly correlated with the strategic environment.



- To bridge the gap:
  - Take the game theory models as source of *domain knowledge*.

Strategic Behavior Prediction: Motivation

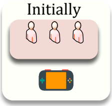
- Due to bounded rationalities, a **gap** exists between the game-theoretic prediction and real human behavior.
  - The gap can be highly correlated with the strategic environment.



- To bridge the gap:
  - Take the game theory models as source of *domain knowledge*.
  - Use machine learning to learn the **mapping** from the domain knowledge and strategic environment to real human behavior.

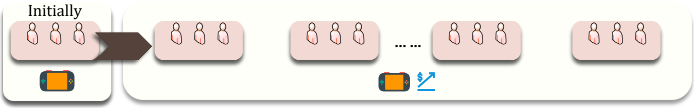
Penny Auction

- Auctions we usually see (e.g., ascending price auction):



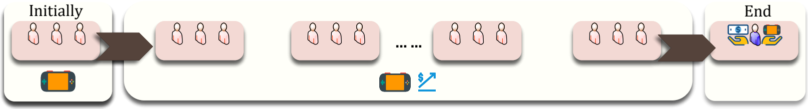
Penny Auction

- Auctions we usually see (e.g., ascending price auction):



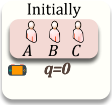
Penny Auction

- Auctions we usually see (e.g., ascending price auction):



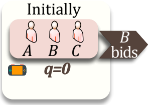
Penny Auction

- In penny auctions:
  - q: buying price



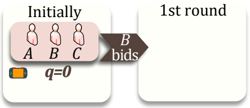
Penny Auction

- In penny auctions:
  - q: buying price



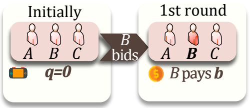
Penny Auction

- In penny auctions:
  - q: buying price



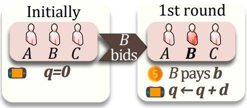
Penny Auction

- In penny auctions:
  - q: buying price
  - b: bid fee



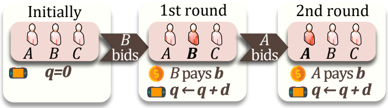
Penny Auction

- In penny auctions:
  - q: buying price
  - b: bid fee
  - d: bid increment



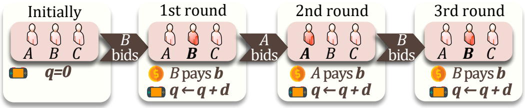
Penny Auction

- In penny auctions:
  - q: buying price
  - b: bid fee
  - d: bid increment



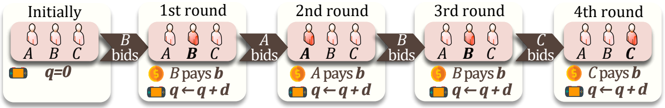
Penny Auction

- In penny auctions:
  - q: buying price
  - b: bid fee
  - d: bid increment



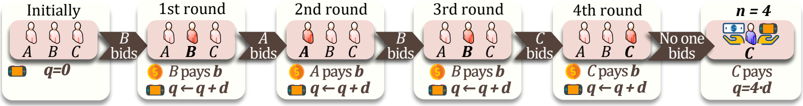
Penny Auction

- In penny auctions:
  - q: buying price
  - b: bid fee
  - d: bid increment

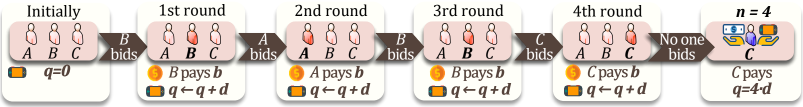


Penny Auction

- In penny auctions:
  - q: buying price
  - b: bid fee
  - d: bid increment
  - n: duration

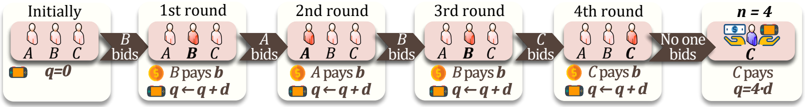


Duration in Penny Auction



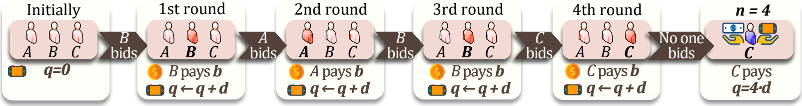
- **What we want to predict**: DURATION.
- Duration is important because it relates to:

Duration in Penny Auction



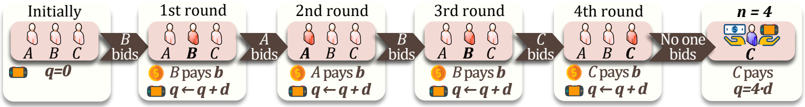
- **What we want to predict**: DURATION.
- Duration is important because it relates to:
  - buying price (e.g., $4 \cdot d$)

Duration in Penny Auction



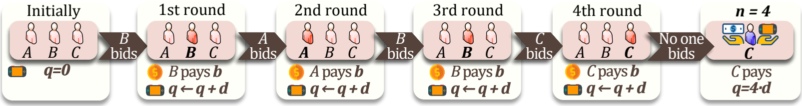
- **What we want to predict**: DURATION.
- Duration is important because it relates to:
  - buying price (e.g., $4 \cdot d$)
  - overall payment (e.g., $4 \cdot b + 4 \cdot d$)

Duration in Penny Auction



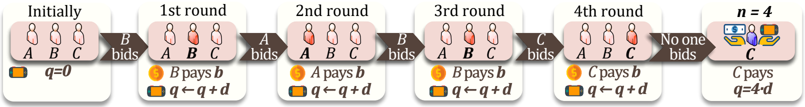
- **What we want to predict**: DURATION.
- Duration is important because it relates to:
  - buying price (e.g., $4 \cdot d$)
  - overall payment (e.g., $4 \cdot b + 4 \cdot d$)
  - **strategic behavior**

Duration in Penny Auction



- **What we want to predict**: DURATION.
- Duration is important because it relates to:
  - buying price (e.g., $4 \cdot d$)
  - overall payment (e.g., $4 \cdot b + 4 \cdot d$)
  - **strategic behavior**
    - Bidders choose between *bidding* or *not bidding*.

Duration in Penny Auction



- **What we want to predict**: DURATION.
- Duration is important because it relates to:
  - buying price (e.g., $4 \cdot d$)
  - overall payment (e.g., $4 \cdot b + 4 \cdot d$)
  - **strategic behavior**
    - Bidders choose between *bidding* or *not bidding*.
    - Duration is the result of bidders' strategic behavior.

Problem Formulation

- **Auction configuration**: Penny auctions are categorized by the configuration $s_i = \{r_i, v_i, b_i, d_i\}, s_i \in \mathcal{S}$.
    - i: data index
    - b_i: bid fee
    - d_i: bid increment

Problem Formulation

- **Auction configuration**: Penny auctions are categorized by the configuration $s_i = \{r_i, v_i, b_i, d_i\}, s_i \in \mathcal{S}$.
  - i: data index
  - b_i: bid fee
  - d_i: bid increment
  - r_i: product (e.g., "Apple iPhone 3G 16GB (White)")
  - v_i: retail price (e.g., "$699")

Problem Formulation

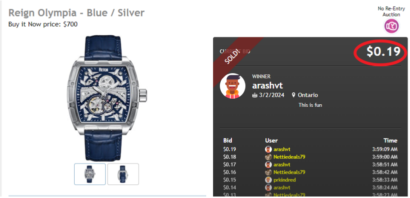
- **Auction configuration**: Penny auctions are categorized by the configuration $s_i = \{r_i, v_i, b_i, d_i\}, s_i \in \mathcal{S}$.
  - i: data index
  - b_i: bid fee
  - d_i: bid increment
  - r_i: product (e.g., “Apple iPhone 3G 16GB (White)”)
  - v_i: retail price (e.g., “$699”)
  - $\mathcal{S}$: the set of all auction configurations

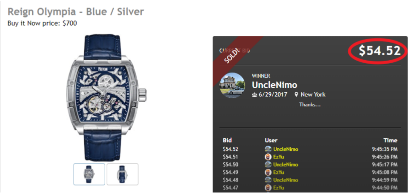
Duration in Penny Auction

- **What we want to predict**: DURATION.

Duration in Penny Auction

- **What we want to predict**: DURATION.
- Penny auctions with the same configuration.





Problem Formulation

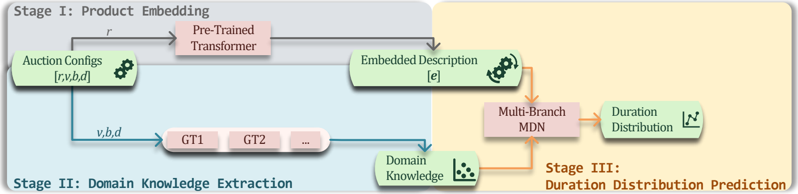
- **Auction configuration**: Penny auctions are categorized by the configuration $s_i = \{r_i, v_i, b_i, d_i\}, s_i \in \mathcal{S}$.
  - i: data index
  - b_i: bid fee
  - d_i: bid increment
  - r_i: product
  - v_i: retail price
  - $\mathcal{S}$: the set of all auction configurations
- **Duration distribution**: The probability that an auction ends after n rounds is $p_{i,n}$ for each $s_i \in \mathcal{S}$.

Auction Duration Prediction Problem

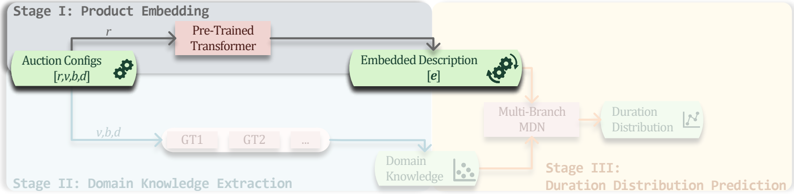
Given s_i, how to predict $p_{i,n}$.

ADAPT

- **A three-stage framework**: **A**uction **D**ur**a**tion **P**redic**t**ion (ADAPT)
- Integration of game theory and machine learning.

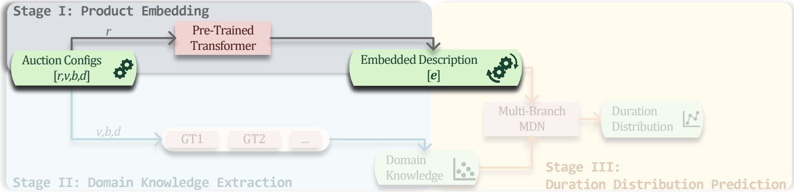


Stage I: Product Embedding



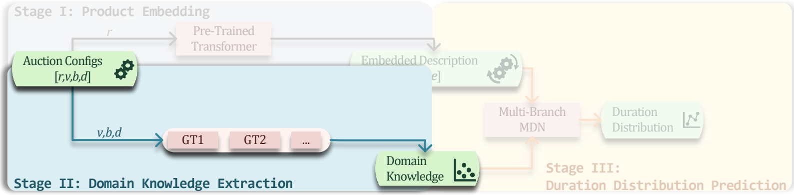
- r: product (e.g., "Apple iPhone 3G 16GB (White)").
- The pre-trained Sentence Transformer Reimers and Gurevych (2019) encodes r into a fixed-length embedding $\boldsymbol{e}$.

Stage I: Product Embedding

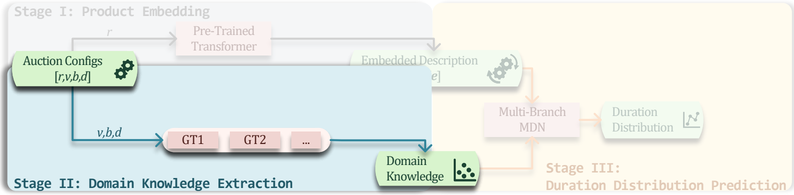


- r: product (e.g., "Apple iPhone 3G 16GB (White)").
- The pre-trained Sentence Transformer Reimers and Gurevych (2019) encodes r into a fixed-length embedding $\boldsymbol{e}$.
- "Context-aware".

Stage II: Domain Knowledge Extraction

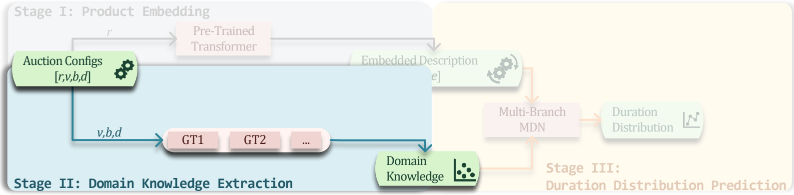


Stage II: Domain Knowledge Extraction



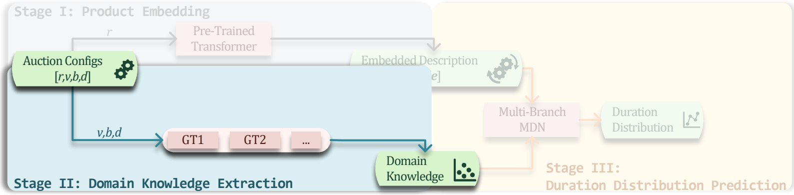
- **Domain knowledge**: Distribution prediction $p_{i,n}$ from different game theory models (e.g., GT1, GT2, . . .).
  - $p_{i,n}$: The probability that an auction ends after n rounds.

Stage II: Domain Knowledge Extraction



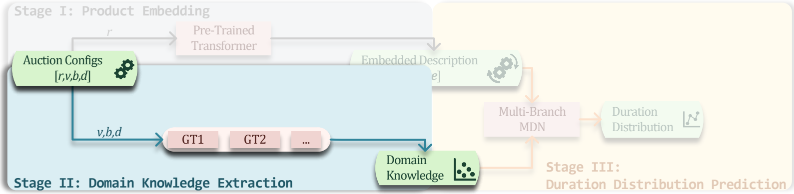
- **Domain knowledge**: Distribution prediction $p_{i,n}$ from different game theory models (e.g., GT1, GT2, ...).
  - $p_{i,n}$: The probability that an auction ends after n rounds.
- **To obtain** $p_{i,n}$,
  - **When to end**: After n rounds, no bidder would like to bid.

Stage II: Domain Knowledge Extraction



- **Domain knowledge**: Distribution prediction $p_{i,n}$ from different game theory models (e.g., GT1, GT2, . . .).
  - $p_{i,n}$: The probability that an auction ends after n rounds.
- **To obtain** $p_{i,n}$,
  - **When to end**: After n rounds, no bidder would like to bid.
  - $u_{i,t}$: The probability that at least one bidder bids after $t-1$ rounds.

Stage II: Domain Knowledge Extraction



- **Domain knowledge**: Distribution prediction $p_{i,n}$ from different game theory models (e.g., GT1, GT2, ...).
  - $p_{i,n}$: The probability that an auction ends after n rounds.
- **To obtain** $p_{i,n}$,
  - **When to end**: After n rounds, no bidder would like to bid.
  - $u_{i,t}$: The probability that at least one bidder bids after $t-1$ rounds.

$$p_{i,n} = (1 - u_{i,n+1}) \cdot \Pi_{t=1}^{n} u_{i,t} \tag{1}$$

Stage II: Domain Knowledge Extraction

- **To obtain** $p_{i,n}$, we need $u_{i,t}$:

$$p_{i,n} = (1 - u_{i,n+1}) \cdot \Pi_{t=1}^{n} u_{i,t} \quad (1)$$

  - $u_{i,t}$: The probability that at least one bidder bids after $t-1$ rounds.

Stage II: Domain Knowledge Extraction

- **To obtain** $p_{i,n}$, we need $u_{i,t}$:

$$p_{i,n} = (1 - u_{i,n+1}) \cdot \Pi_{t=1}^{n} u_{i,t} \tag{1}$$

  - $u_{i,t}$: The probability that at least one bidder bids after $t-1$ rounds.
- **To obtain** $u_{i,t}$, we turn to game theory models.

Stage II: Domain Knowledge Extraction

- **To obtain** $p_{i,n}$, we need $u_{i,t}$:

$$p_{i,n} = (1 - u_{i,n+1}) \cdot \Pi_{t=1}^{n} u_{i,t} \tag{1}$$

  - $u_{i,t}$: The probability that at least one bidder bids after $t-1$ rounds.
- **To obtain** $u_{i,t}$, we turn to game theory models.
- **Equilibrium condition**: A bidder is indifferent between bidding or not bidding, when these two actions have the same utility.

Stage II: Domain Knowledge Extraction

- **To obtain** $p_{i,n}$, we need $u_{i,t}$:

$$p_{i,n} = (1 - u_{i,n+1}) \cdot \Pi_{t=1}^{n} u_{i,t} \tag{1}$$

  - $u_{i,t}$: The probability that at least one bidder bids after $t-1$ rounds.
- **To obtain** $u_{i,t}$, we turn to game theory models.
- **Equilibrium condition**: A bidder is indifferent between bidding or not bidding, when these two actions have the same utility.
- **Example: game theory model based on Expected Utility Theory (EUT)**:

Stage II: Domain Knowledge Extraction

- **To obtain** $p_{i,n}$, we need $u_{i,t}$:

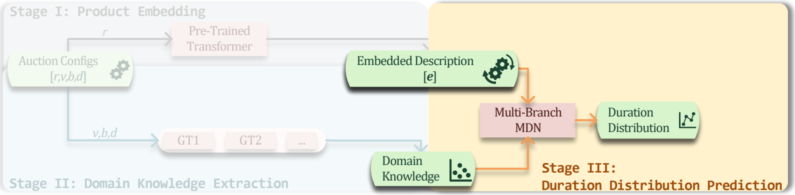
$$p_{i,n} = (1 - u_{i,n+1}) \cdot \Pi_{t=1}^{n} u_{i,t} \tag{1}$$

  - $u_{i,t}$: The probability that at least one bidder bids after $t-1$ rounds.
- **To obtain** $u_{i,t}$, we turn to game theory models.
- **Equilibrium condition**: A bidder is indifferent between bidding or not bidding, when these two actions have the same utility.
- **Example: game theory model based on Expected Utility Theory (EUT)**:

$$\underbrace{(1-u_{i,t}) \cdot (v_i-(t-1) \cdot d_i-b_i)}_{\text{Bidding and winning}} + \underbrace{u_{i,t} \cdot (-b_i)}_{\text{Bidding and losing}} = \underbrace{0}_{\text{Not bidding}} \tag{2}$$

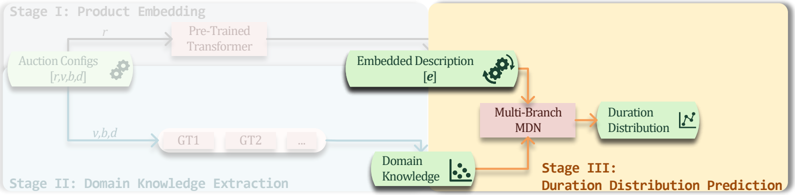
  - v_i: retail price
  - b_i: bid fee
  - d_i: bid increment

Stage III: Duration Distribution Prediction



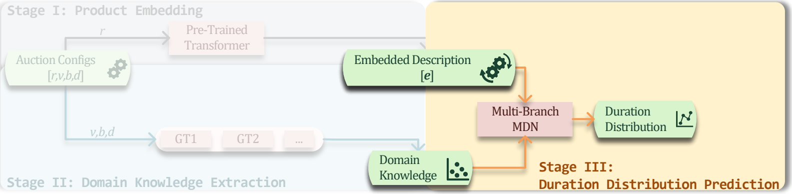
- **Multi-Branch MDN**: **M**ulti-**B**ranch **M**ixture **D**ensity **N**etwork.
  - Input: Output from Stage I and Stage II.

Stage III: Duration Distribution Prediction



- **Multi-Branch MDN**: **M**ulti-Branch **M**ixture **D**ensity **N**etwork.
  - Input: Output from Stage I and Stage II.
  - Output: Prediction of the auction duration distribution.
    - Mixture density network Bishop (1994) is able to output a distribution.

Stage III: Duration Distribution Prediction



- **Loss function**: Negative log-likelihood.

$$\mathrm{Loss}_{\mathrm{MB-MDN}} = -\sum_{i:|\mathcal{N}_i|\neq 0} \frac{1}{|\mathcal{N}_i|} \sum_{n\in\mathcal{N}_i} \log(\hat{p}_{i,n}), \tag{3}$$

  - n: actual auction duration.
  - i: index.
  - $\hat{p}_{i,n}$: prediction from ADAPT.
  - $\mathcal{N}_i$: the set of actual auction durations under configuration s_i.

Experimental Settings: Comparison Methods

- **To evaluate auction duration prediction performance, we compare**:
    - Our method (ADAPT).

Experimental Settings: Comparison Methods

- **To evaluate auction duration prediction performance, we compare**:
  - Our method (ADAPT).
  - Game theory models (GT1, GT2, GT3).
  - Machine learning-only method (EMB+MDN).

Experimental Settings: Comparison Methods

- **To evaluate auction duration prediction performance, we compare**:
  - Our method (ADAPT).
  - Game theory models (GT1, GT2, GT3).
  - Machine learning-only method (EMB+MDN).
  - Other combinations of game theory and machine learning (GT1+MDN, GT2+MDN, GT1+EMB+MDN, GT2+EMB+MDN, GT1+GT2+MDN).

Experimental Settings: Comparison Methods

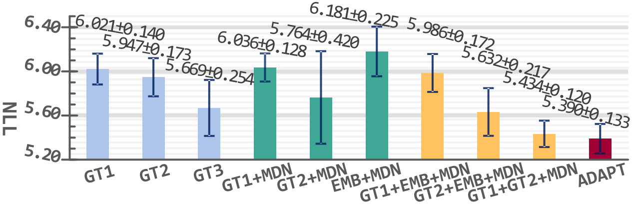
- **To evaluate auction duration prediction performance, we compare**:
  - Our method (ADAPT).
  - Game theory models (GT1, GT2, GT3).
  - Machine learning-only method (EMB+MDN).
  - Other combinations of game theory and machine learning (GT1+MDN, GT2+MDN, GT1+EMB+MDN, GT2+EMB+MDN, GT1+GT2+MDN).
    - ADAPT = GT1+GT2+EMD+MDN.

Experimental Settings: Datasets

- **Synthetic dataset**: $1,276$ auction configurations.
- **Real dataset**: $115,831$ records, $1,276$ auction configurations.
  - Collected by Byers et al. (2010); Augenblick (2016) from online penny auction websites.

Results: Auction Duration Prediction

- **Data**: Synthetic testing data (10% of all synthetic data).
- **Metric**: NLL averaged over 8 runs.



Results: Auction Duration Prediction

- **Data**: Real testing data (10% of all real data).
- **Metric**: NLL averaged over 8 runs.

		GT1	GT2	GT3	GT1+MDN	GT2+MDN	EMB+MDN	GT1+EMB+MDN	GT2+EMB+MDN	GT1+GT2+MDN	**ADAPT**
NLL	Avg	6.795	6.858	6.851	6.626	6.684	6.728	6.454	6.462	6.416	**6.344**
	Std	0.140	0.208	0.145	0.409	0.283	0.113	0.103	0.111	0.075	**0.061**

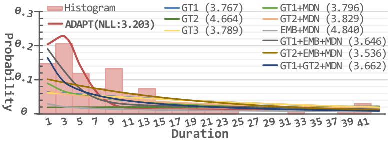
Results: Auction Duration Prediction

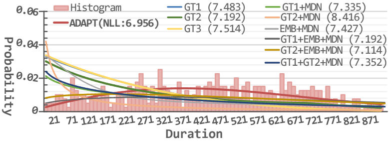
- **Data**: Real testing data (10% of all real data).
- **Metric**: NLL & KL-divergence averaged over 8 runs.

		GT1	GT2	GT3	GT1+MDN	GT2+MDN	EMB+MDN	GT1+EMB+MDN	GT2+EMB+MDN	GT1+GT2+MDN	**ADAPT**
NLL	Avg	6.795	6.858	6.851	6.626	6.684	6.728	6.454	6.462	6.416	**6.344**
	Std	0.140	0.208	0.145	0.409	0.283	0.113	0.103	0.111	0.075	**0.061**
KL-D	Avg	3.254	3.564	3.541	2.928	3.224	3.203	2.848	2.967	2.911	**2.836**
	Std	0.128	0.209	0.463	0.084	0.259	0.086	0.076	0.143	0.123	**0.075**

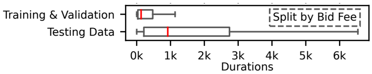
Results: Auction Duration Prediction

- **Example in real testing data**:
  - Histogram: Actual auction duration (normalized).
  - **ADAPT**: Prediction with the lowest NLL.



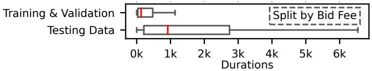


Results: Prediction Under Large Domain Shifts



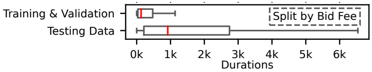
Results: Prediction Under Large Domain Shifts

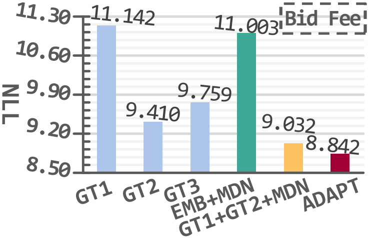
- **Split by bid fee**:
  - Real testing data: auctions with the bid fee of 0.01.
  - Real training & validation data: the remaining.



Results: Prediction Under Large Domain Shifts

- **Split by bid fee**:
  - Real testing data: auctions with the bid fee of 0.01.
  - Real training & validation data: the remaining.



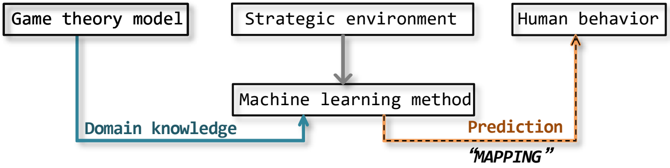


Contribution

- Design a framework to predict the auction duration under different auction configurations.

Contribution

- Design a framework to predict the auction duration under different auction configurations.
- Propose a approach to predict strategic behavior combining the strengths of game theory and machine learning.



References

Augenblick, N. (2016). The sunk-cost fallacy in penny auctions. *The Review of Economic Studies*, 83(1):58–86.

Bishop, C. (1994). Mixture density networks. Workingpaper, Aston University.

Byers, J. W., Mitzenmacher, M., and Zervas, G. (2010). Information asymmetries in pay-per-bid auctions. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 1–12.

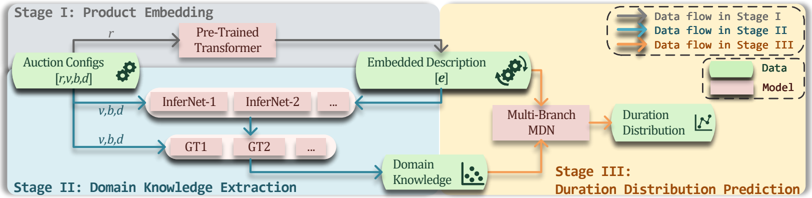
Glauner, P., State, R., Valtchev, P., and Duarte, D. (2018). On the reduction of biases in big data sets for the detection of irregular power usage. In *Data Science and Knowledge Engineering for Sensing Decision Support: Proceedings of the 13th International FLINS Conference (FLINS 2018)*, pages 439–445. World Scientific.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Thanks!

Supplement: InferNet

- **A three-stage framework:** Auction **D**ur**a**tion **P**redic**t**ion (ADAPT)



Supplement: Model Parameter Inference

- **Data**: Synthetic testing data and real testing data (10% of all).
- **Model**: GT2 with parameters inferred by different methods.
- **Metric**: NLLs averaged over 8 runs.

		SA-Avg	SA-Unified	**InferNet**
Synthetic Data	Avg	7.938	6.588	**5.947**
	Std	0.262	0.565	**0.173**
Real Data	Avg	7.492	7.421	**6.858**
	Std	**0.133**	0.355	0.208