# Towards Generalizable Reinforcement Learning for Trade Execution

Chuheng Zhang, Xiaoyu Chen, Yitong Duan, Jianyu Chen, Jian Li, Li Zhao

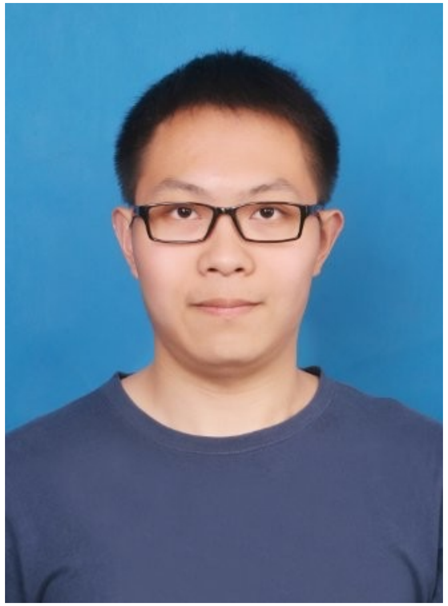
Tsinghua University, IIIS

Microsoft Research Asian

(This is a working draft. Feel free to discuss~)

# Short Bio

- Chuheng Zhang (张楚珩) is researcher at MSRA
- Ph.D. at Institute of interdisciplinary Information Sciences, Tsinghua University (Sep, 2016 - July, 2022), advised by Prof. Jian Li (李建)
- Bachelor at Physics Department at Nanjing University
- Research: reinforcement learning applications (e.g., quantitative investment, inventory management, advertising)
- Publications: AAAI, ICLR, WWW, SIGIR, CIKM, ICDM



# Outline

- RL background
- Trade execution background
- Methods
- Results
- Discussion

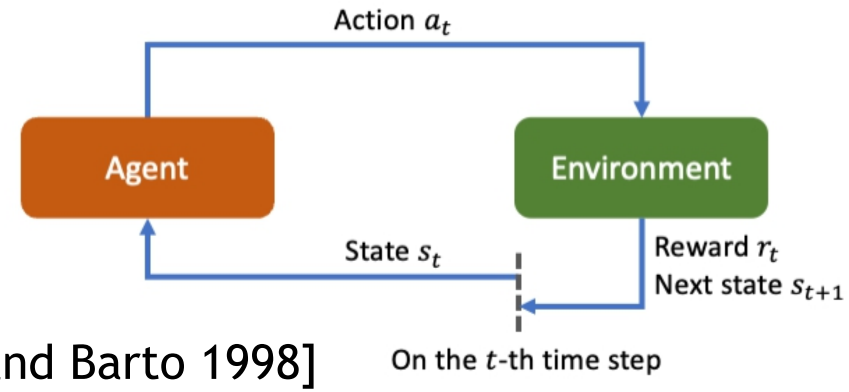
# Motivation

- RL has awesome performance on various games.
- Sequential decision is common in industry.
- Why not use RL in real applications?

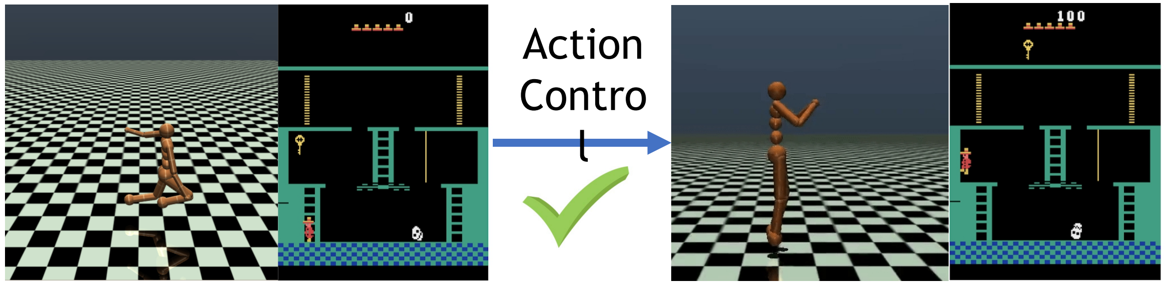
| Challenges | Solutions |
|---|---|
| Costly Interaction | RL from Offline Data |
| Sample Inefficiency | Representation Learning, Knowledge Transfer, … |
| Instability | Stable Algorithm |
| **Generalization** | **This work** |

# RL and Context

- Popular RL Applications




[Sutton and Barto 1998]




State State

- Some Industrial RL Applications


Context (Environment State)

State (Agent State)

Action

- Distribution shift in offline RL

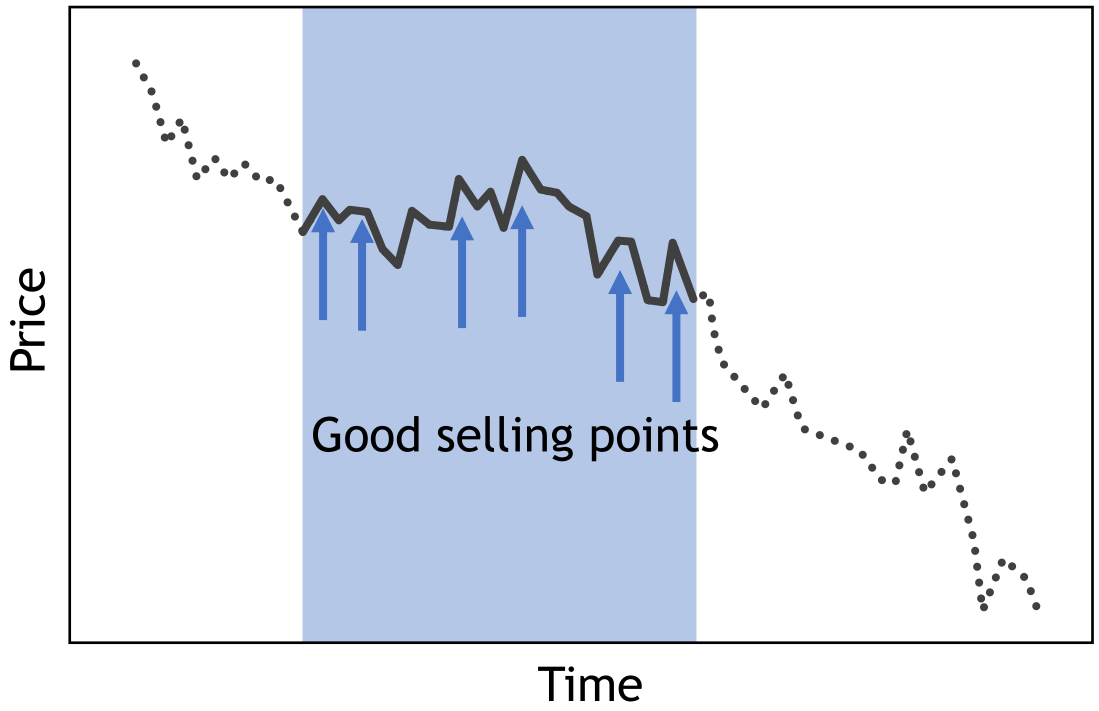
  [Fujimoto et al 2019; Kumar et al 2020]

- Control or real-time adaptation?

# Introduction for Trade Execution

- Trade execution is to sell/buy a given amount of assets in a given time with the lowest possible trading cost
- Background: Trade execution is an important task for brokerage firms

Task: sell 1000 stocks within this time period



# The Setting: Limit Order Book (LOB)

- Limit order book (LOB) is a record of outstanding limit orders
- Market order (MO) is executed immediately with large trading cost
- Limit order (LO) ensures the price at the risk of non-execution
- Raw offline data contains information for us to reconstruct the LOB dynamics

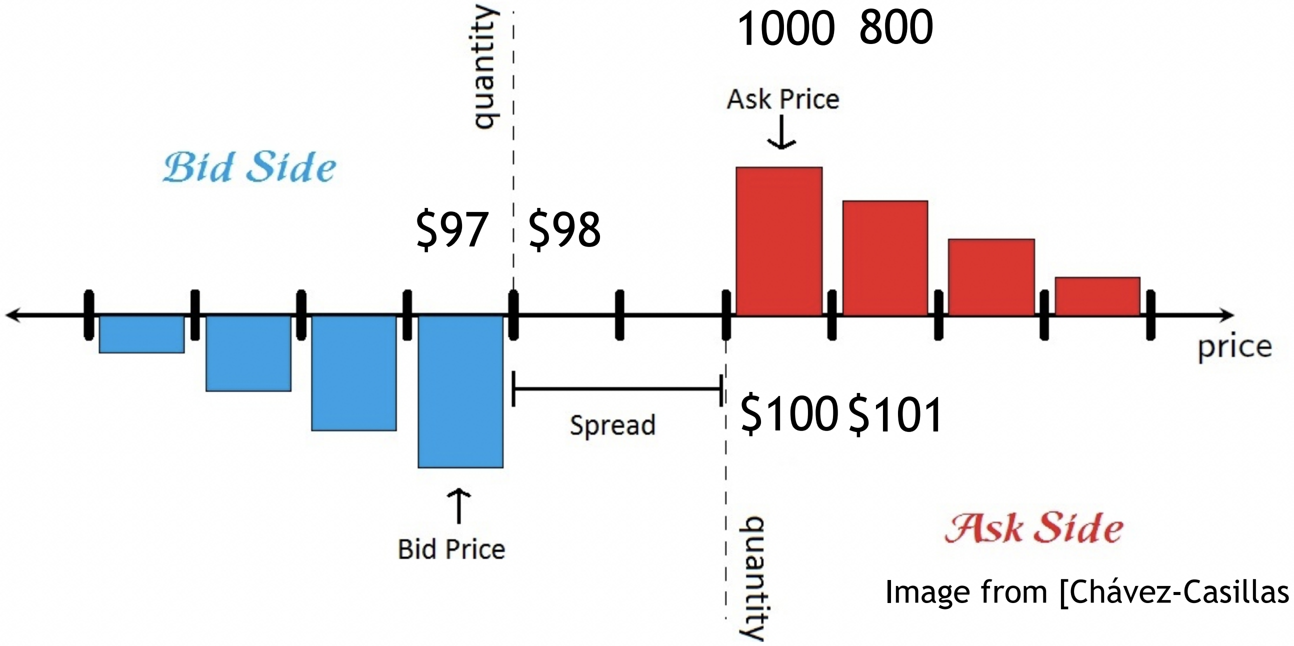
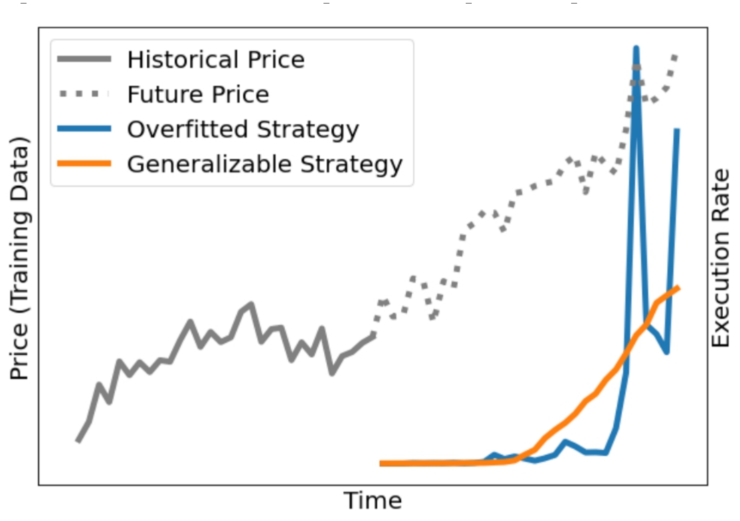



Image from [Chávez-Casillas and Figueroa-López 2017]

# Background for Trade Execution

- Traditional methods [see e.g., Almgren and Chriss 2000; Guéant et al 2012]
  - Generate static policies
  - Rely on strong assumptions
- Previous RL methods suffer from overfitting
  - Observation (market indicators) is high nd noisy
  - Agent overfits spurious noise/feature
- Objective: Developing generalizable RL ag



# RL formulation

- Simulator: based on LOB data
- Observation:
  - State: inventory level, remaining time
  - Context

| | AskPrice | Volume | MACD | Spread | … |
|---|---|---|---|---|---|
| $t-k$ | | | | | |
| … | | | | | |
| $t$ | | | | | |

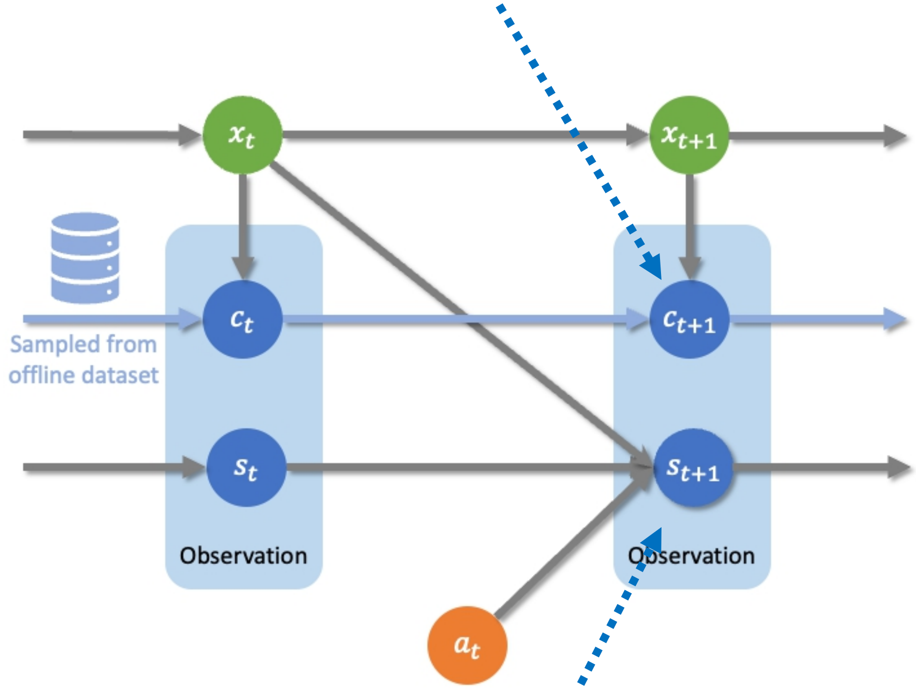
- Action: LO (price, direction, quantity), MO (direction, quantity)
- Reward/objective: Maximize the cash obtained from selling the stocks

- Challenge: Context is high-dimensional, redundant, and noisy

# Modeling

- Offline RL with Dynamic Context (ORDC)
- $\phi: \mathcal{C}$ (context) $\rightarrow \mathcal{X}$ (latent context) with $|\mathcal{C}| \gg |\mathcal{X}|$. E.g., for Brownian motion, $c$ is historical prices and $x = (\alpha, \sigma)$
- Context dynamics is highly stochastic and we assume state dyanamics is known

**Context**: Uncontrollable
E.g., market indicators
Usually high-dimensional, redundant, and stochastic




**State**: Controllable
E.g., inventory level
The transition dynamics is usually clear

# Theoretical Analysis

- Generalization is hard when data is limited, even if the context is compressible

  Theorem 1 [Sample complexity lower bound]: In a class of ORDC models, any algorithm needs at least $\Omega\left(\frac{|\mathcal{C}|\log(|\mathcal{C}|/\delta)}{(1-\gamma)^3\epsilon^2}\right)$ context samples to learn.

- If we could properly compress the context, limited data leads to good generalization

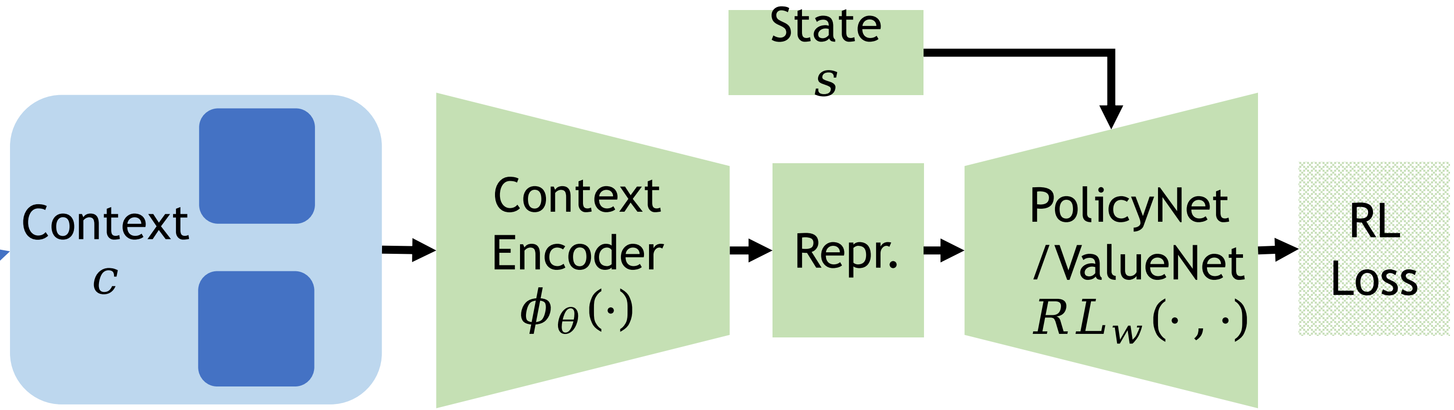
  Theorem 2 [Sample complexity upper bound] : With access to $\phi: \mathcal{C} \rightarrow \mathcal{X}$, an algorithm can learn with $O\left(\frac{|\mathcal{X}|^2\log(|\mathcal{X}|/\delta)}{(1-\gamma)^4\epsilon^2}\right)$

- Motivate: Compress context representation

| | AskPrice | Volume | MACD | RSI | … |
|---|---|---|---|---|---|
| $t-k$ | Context $c \in \mathcal{C}$ | | | | |
| … | | | | | |
| $t$ | | | | | |

# Methods: RL Formulation

- Standard RL model for trade execution

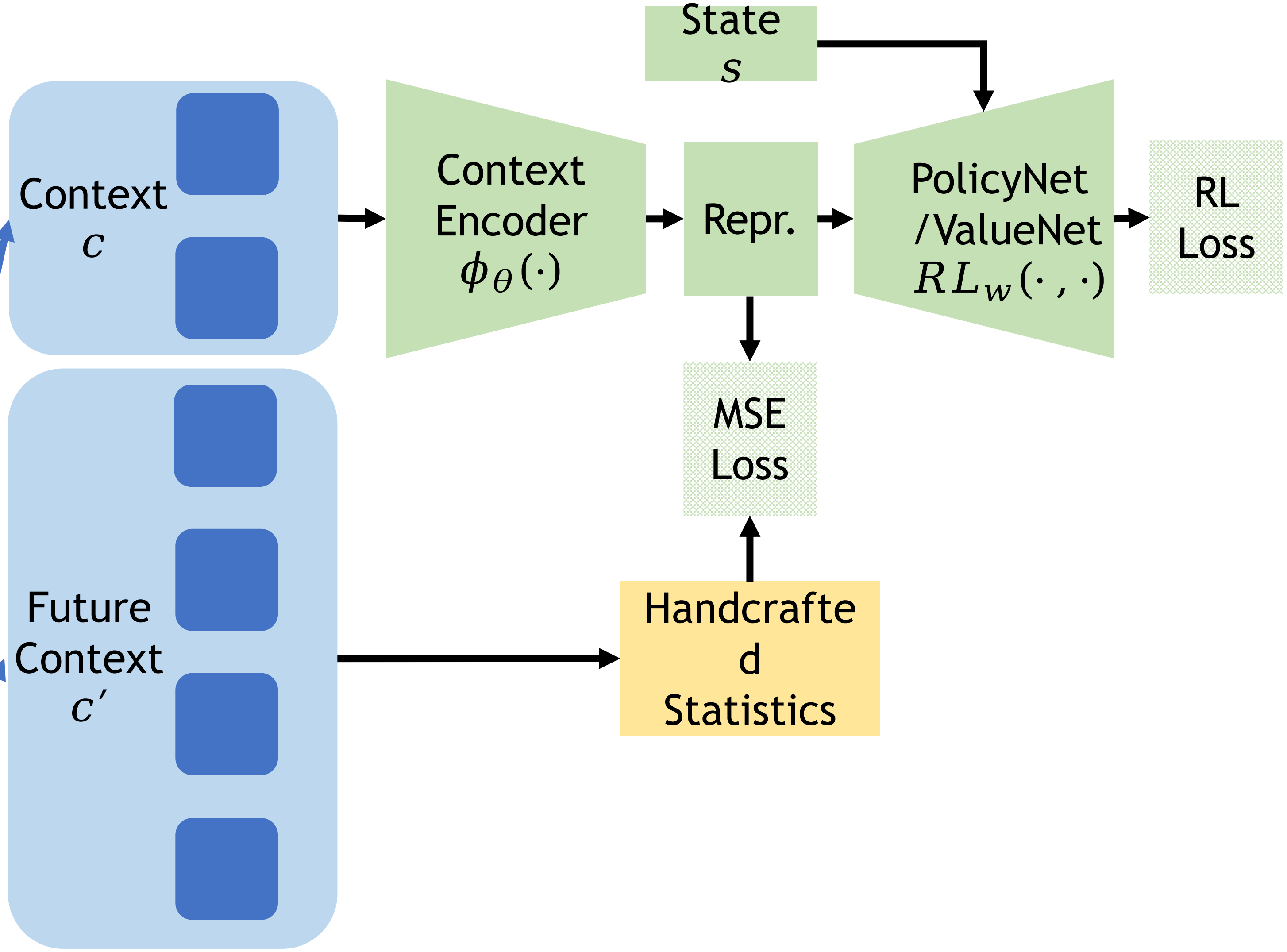



| | AskPrice | Volume | MACD | … |
|---|---|---|---|---|
| $t-k$ | | | | |
| … | | | | |
| $t$ | | | | |

# Methods: CASH

- CASH: Context Aggregation with Handcrafted Statistics
- Pre-train the context encoder to predict key future statistics
- E.g., future price trend and volatility

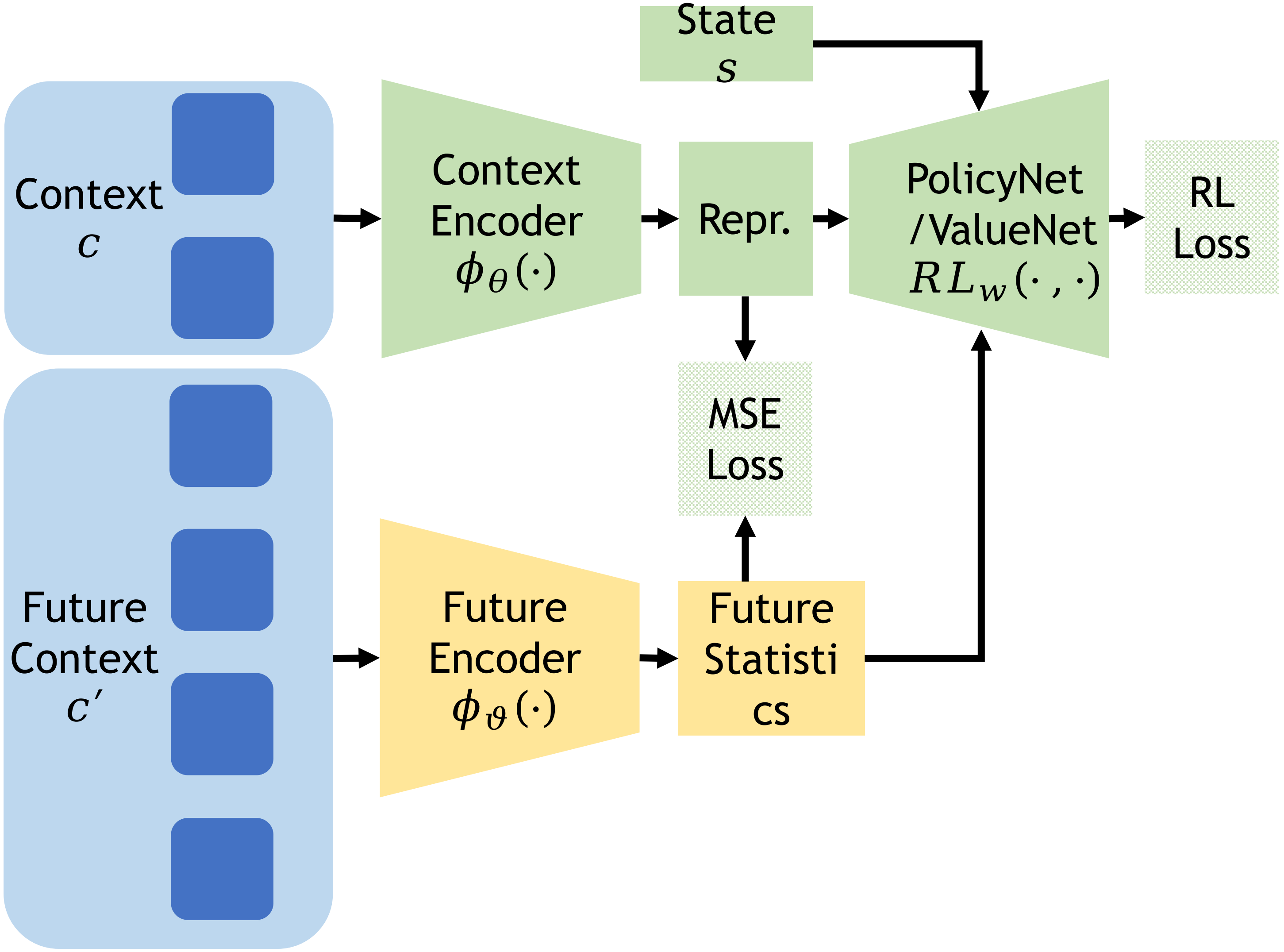
| | AskPrice | Volume | MACD | … |
|---|---|---|---|---|
| $t-k$ | | | | |
| … | | | | |
| $t$ | | | | |
| $t+1$ | | | | |
| … | | | | |
| $t+k'$ | | | | |



# Methods: CATE

- CATE: Context Aggregation with End-to-end Training
- Future encoder is trained to

1. Extract useful information from future
2. Be predictable based on current context



# Methods

- Why learn a compressed context representation to predict the statistics of the future context?

1. Convenient: The context representation depending on the future context is independent of actions
2. Informative: Information on the future context (e.g., prices) is useful for the agent to make decisions
3. Feasible: The statistics of future context is more predictable than the full future context

# Experiment Results: Trading Cost

| Algorithm | Training | Validation | Testing | Generalization Gap |
|---|---|---|---|---|
| TWAP Strategy | - | - | 13.1217 (2.0858) | - |
| Tuned DQN | 1.6078 (2.1974) | 4.9241 (2.2437) | 6.7199 (2.7776) | 5.1121 |
| Nevmyvaka et al. (2006) | 2.8726 (5.6576) | 8.8261 (1.6331) | 9.7045 (1.2583) | 6.8319 |
| Ning et al. (2018) | 7.2350 (5.5857) | 10.5678 (1.8090) | 10.0825 (2.1145) | 2.8475 |
| Lin et al. (2020) | 6.0305 (6.7465) | 11.1709 (1.1422) | 12.4054 (1.3553) | 6.3749 |
| Tuned DQN + CASH | 1.9351 (1.8016) | 3.2507 (1.2944) | 3.9184 (1.2287) | **1.9833** |
| Tuned DQN + CATE | -2.7526 (1.3319) | -1.7554 (1.1277) | **0.0749** (1.5233) | 2.8275 |
| Tuned PPO | -1.7340 (1.7653) | 1.9763 (0.5393) | 3.2686 (0.5302) | 5.0026 |
| Dabérius et al. (2019) | 6.7671 (11.0498) | 10.2517 (2.2711) | 12.9987 (2.5579) | 6.2316 |
| Lin et al. (2021) | 5.5028 (7.4558) | 7.8800 (1.0740) | 10.2599 (1.5293) | 4.4571 |
| Fang et al. (2021) | -5.4925 (15.8669) | 9.9096 (4.2206) | 11.5612 (5.3556) | 17.0537 |
| Tuned PPO + CASH | -4.7082 (0.8733) | -4.1690 (0.2108) | -3.8524 (0.2234) | **0.8558** |
| Tuned PPO + CATE | -5.1725 (0.6705) | -4.7672 (0.1717) | **-4.3000** (0.4813) | 0.8725 |

Table The trading cost (bp=$10^{-4}$) of different algorithms. The validation set is used for hyperparameter tuning. The numbers are the average mean (std.) trading cost in the last 10% iterations over five different random seeds.

# Experiment Results: Learned Policy

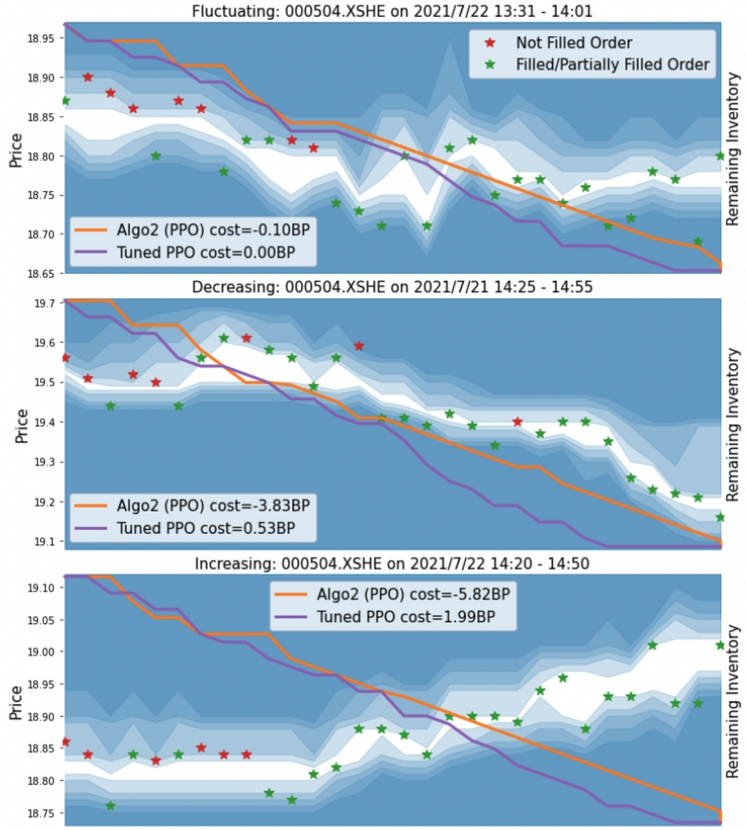



Figure 3: The policy learned by CATE (based on PPO) and the corresponding baseline. The background shaded areas indicate the 5-level ask/bid prices, and the stars indicate the orders placed by CATE. The lines indicate the remaining inventory of CATE and the baseline algorithm.

| Predicted Statistics | Price | Volume |
|---|---|---|
| Twap volatility | 1.2871 | -0.4335 |
| Avg future twap - current twap | 4.5296 | -0.0420 |
| Max future twap - current twap | 1.3932 | -0.1761 |
| Min future twap - current twap | -1.0587 | 0.2978 |
| Sprd volatility | 6.2039 | -0.6810 |
| Avg future sprd - current sprd | -0.0944 | 0.0879 |
| Max future sprd - current sprd | 1.8254 | -0.2629 |
| Min future sprd - current sprd | -1.2859 | 0.1579 |

Table 3: The impact of predicted statistics on the agent's action in CASH (based on PPO).

# Discussion

- Generalization in Finance
    - Data augmentation
    - Ensemble
    - Market style detection and adaptation
    - …