Market making with signals through deep reinforcement learning

김나영

Citations: 56
Published: 2021.04.29 in IEEE access

Rank by Journal Impact Factor

Journals within a category are sorted in descending order by Journal Impact Factor (JIF) resulting in the Category Ranking below. A separate rank is shown for each category in which the journal is listed in JCR. Beginning in 2023, ranks are calculated by category. Learn more

CATEGORY

COMPUTER SCIENCE, INFORMATION SYSTEMS

93/258

JCR YEAR	JIF RANK	JIF QUARTILE	JIF PERCENTILE
2024	93/258	Q2	64.1
2023	87/250	Q2	65.4

CATEGORY

ENGINEERING, ELECTRICAL & ELECTRONIC

128/366

JCR YEAR	JIF RANK	JIF QUARTILE	JIF PERCENTILE
2024	128/366	Q2	65.2
2023	122/353	Q2	65.6

Rank by JIF before 2023 for COMPUTER SCIENCE, INFORMATION SYSTEMS

EDITION

Science Citation Index Expanded (SCIE)

JCR YEAR	JIF RANK	JIF QUARTILE	JIF PERCENTILE
2022	73/158	Q2	54.1
2021	79/164	Q2	52.13
2020	65/161	Q2	59.94
2019	35/156	Q1	77.88

Rank by JIF before 2023 for ENGINEERING, ELECTRICAL & ELECTRONIC

EDITION

Science Citation Index Expanded (SCIE)

JCR YEAR	JIF RANK	JIF QUARTILE	JIF PERCENTILE
2022	100/275	Q2	63.8
2021	105/276	Q2	62.14
2020	94/273	Q2	65.75
2019	61/266	Q1	77.26

자료: Journal Citation Reports(Clarivate)

Authors

Bruno Gasperov

Bruno Gasperov

다른 이름 ▸

팔로우

Researcher, University of Ljubljana Faculty of Computer and Information Science
fri.uni-lj.si의 이메일 확인됨 - 홈페이지

reinforcement learning evolutionary computation quantitative finance quality-diversity

제목	인용	연도
Market making with signals through deep reinforcement learning B Gašperov, Z Kostanjčar IEEE access 9, 61611-61622	56	2021
Reinforcement learning approaches to optimal market making B Gašperov, S Begušić, P Posedel Šimović, Z Kostanjčar Mathematics 9 (21), 2689	34	2021
Deep reinforcement learning for market making under a Hawkes process-based limit order book model B Gašperov, Z Kostanjčar IEEE control systems letters 6, 2485-2490	21	2022

K. Geert Rouwenhorst

Zvonko Kostanjcar

팔로우

Associate Professor, University of Zagreb, Faculty of Electrical Engineering and Computing
fer.hr의 이메일 확인됨 - 홈페이지

Data science Quantitative finance Stochastic modeling

제목	인용	연도
Scaling properties of extreme price fluctuations in Bitcoin markets S Begušić, Z Kostanjčar, HE Stanley, B Podobnik Physica A: Statistical Mechanics and its Applications 510, 400-406	129	2018
Market making with signals through deep reinforcement learning B Gašperov, Z Kostanjčar IEEE access 9, 61611-61622	56	2021
Reinforcement learning approaches to optimal market making B Gašperov, S Begušić, P Posedel Šimović, Z Kostanjčar Mathematics 9 (21), 2689	34	2021
Deep neural networks for behavioral credit rating A Merćep, L Mrčela, M Birov, Z Kostanjčar Entropy 23 (1), 27	31	2020
Authentication approach using one-time challenge generation based on user behavior patterns captured in transactional data sets K Skračić, P Pale, Z Kostanjčar Computers & security 67, 107-121	27	2017
Information feedback in temporal networks as a predictor of market crashes S Begušić, Z Kostanjčar, D Kovač, HE Stanley, B Podobnik Complexity 2018 (1), 2834680	25	2018
Churn prediction methods based on mutual customer interdependence K Ljubičić, A Merćep, Z Kostanjčar Journal of Computational Science 67, 101940	23	2023

Keywords

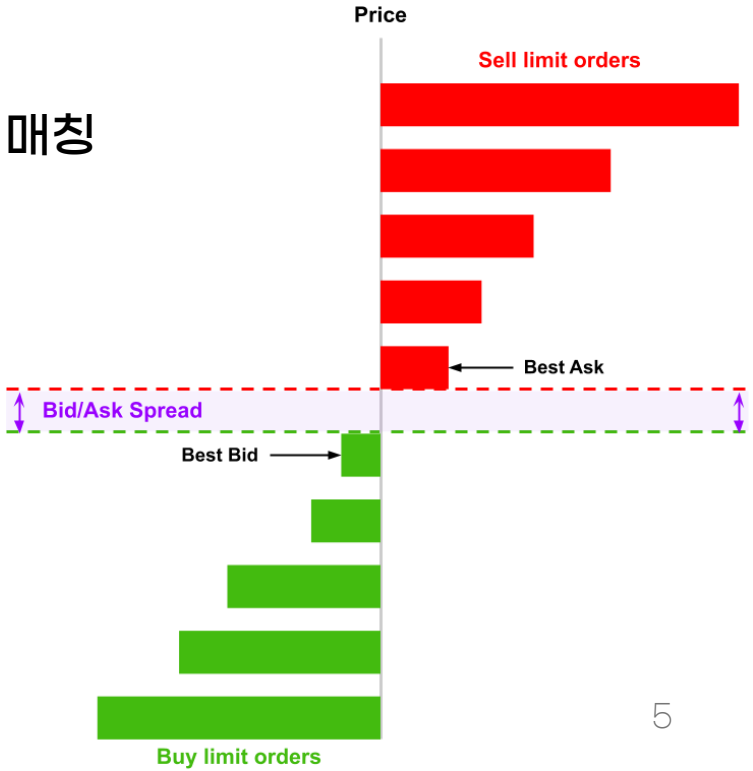
- Deep reinforcement learning
- Genetic algorithms
- High-frequency trading
- Machine learning
- Market making
- Stochastic control

Index

Keyword 1: Limit Order Book(LOB)

특정 상품의 미체결(limit) 주문을 가격 구간별로 쌓아둔 장부

- **지정가 주문(limit order): '가격'을 지정**
  - "이 가격 이하로 사겠다" 또는 "이 가격 이상으로 팔겠다"라고 미리 내는 주문
  - 체결되지 않으면 장부에 남아 있다가, 취소·수정되거나 반대쪽 시장가 주문에 걸리면 사라짐
- **시장가 주문(market order): '수량'을 지정**
  - 즉시 체결을 원하는 주문. 장부 위에 남아 있는 지정가 주문과 바로 매칭
  - 이때 거래소는 '가격-시간 우선순위'에 따라 체결
- **호가(tick):** 주문할 수 있는 최소 단위 가격 변화폭
- **최우선 매도(best ask):** 남아 있는 매도 주문 중 가장 낮은 가격
- **최우선 매수(best bid):** 남아 있는 매수 주문 중 가장 높은 가격
- **스프레드(bid-ask spread):** 최우선 매도와 매수 가격 차이
- **중간가(mid-price):** (best ask + best bid)÷2



Keyword 2: Market Making

특정 자산의 양쪽에 동시에 주문을 걸어두고, 스프레드를 꾸준히 챙기면서도 재고 위험을 최소화

- **방법: 양방향 호가(Quote) 제시**
  - 특정 자산에 대해 매수(bid)와 매도(ask) 호가를 **동시에** 제시
  - Ex. “100원에 10주 사고(매수), 101원에 10주 팝니다(매도)”
  - 매도-매수 간 가격 차이(스프레드, 여기서는 101원-100원=1원)를 안정적으로 취할 수 있음
- **수익원: 스프레드(Spread) 포착**
  - 매도 호가(Ask)와 매수 호가(Bid) 간 차이가 곧 시장 조성가의 잠재 이익
  - 스프레드를 **반복**해서 취득. **짧은 시간** 내에 **여러 차례** 수익을 쌓음 → High-Frequency Trading
- **리스크: 재고 위험(Inventory Risk) 관리**
  - 호가가 체결되면 시장 조성가의 장부에 보유 주식 발생 → 가격 변동 리스크 존재
  - **한 쪽만 체결된 경우, 보유 물량이 너무 많아지면** 가격 하락 시 큰 **손실 발생 가능성** ↑
  - 수시로 **반대쪽 호가를 조정***하며 재고를 중립 상태로 유지

* 재고 과다(매도호가 공격적(낮게) + 매수호가 보수적(더 낮게)) * 재고 부족(매수호가 공격적(높게) + 매도호가 보수적(더 높게))

Keyword 3: Genetic Algorithms

자연선택/유전 원리를 모방해 해답 후보들을 반복적으로 교배/돌연변이시키며 최적해를 찾는 방식

- **염색체(Chromosome) 인코딩**

  해결하고자 하는 문제의 후보 해 답을 벡터로 표현

- **초기 집단(Population) 생성**
  - 다양한 염색체를 무작위로 다수 생성해 첫 세대를 구성
- **적합도 평가(Fitness Evaluation)**
  - 각 염색체가 문제를 얼마나 잘 푸는지 평가함으로써 '좋은 해'와 '나쁜 해'를 구분
- **선택/교차/돌연변이(Selection, Crossover, Mutation)**
  - 높은 적합도의 개체를 부모로 **골라(Selection) 교차(Crossover)**로 자식을 만들고,
  - **돌연변이(Mutation)**로 일부 유전자를 바꿔 다양성을 유지
- **세대 교체 및 종료(Termination)**
  - 새로운 세대로 대체하면서 적합도가 충분히 좋아지거나 최대 세대 수에 도달하면

    알고리즘을 멈추고, 최고의 염색체를 최종 해답으로 채택

Keyword 4: Neuro-evolution algorithms

유전 알고리즘(genetic algorithm)을 신경망(DNN)에 적용한 기법

- **염색체(Chromosome) 인코딩**

  각각의 DNN 파라미터(가중치·편향)를 하나의 염색체(chromosome)처럼 인코딩

- **초기 집단(Population) 생성**
  - 랜덤한 가중치 조합을 가진 여러 개의 DNN(개체)로 구성된 첫 세대를 구성
- **적합도 평가(Fitness Evaluation)**
  - 각 DNN Agent가 내는 행동으로 얻는 누적 보상을 측정해 “적합도(fitness)”를 계산
- **선택/교차/돌연변이(Selection, Crossover, Mutation)**
  - 높은 적합도의 개체를 부모로 **골라(Selection)** **교차(Crossover)**로 자식을 만들고,
  - **돌연변이(Mutation)**로 일부 유전자를 바꿔 다양성을 유지
- **세대 교체 및 종료(Termination)**
  - 최대 세대 수 도달, 목표 적합도 달성, 성능 개선 정체 등으로 학습 종료 시점 결정

Keyword 5: Model-free DRL(1/2)

1) Model-free: 환경의 내부 규칙을 미리 수학적으로 만들지 않고, 실제 시뮬레이션으로 학습

2) DRL: 정책이나 가치를 표현하는 함수를 신경망으로 만들고 파라미터 최적화

- MDP의 구성 요소
  - 상태(S): 환경의 현재 모습(ex. 장에서의 매도·매수 물량)
  - 행동(A): Agent가 그 상태에서 선택할 수 있는 조치(ex. 얼마에, 몇 주를 사고팔지 결정)
  - 전이 확률(P): “상태 S에서 행동 A를 하면 다음에 상태 S′가 될 확률이 얼마인가?”를 나타내는 함수
  - 보상(R): 행동을 취했을 때 얻는 점수(이익)
  - 감가율(γ): 먼 미래 보상의 가치가 점점 작아지게 만드는 장치(0~1 사이).

Keyword 5: Model-free DRL(2/2)

1) Model-free: 환경의 내부 규칙을 미리 수학적으로 만들지 않고, 실제 시뮬레이션으로 학습

2) DRL: 정책이나 가치를 표현하는 함수를 신경망으로 만들어, 파라미터 최적화

- 학습 과정
  - Agent는 매 순간 상태(St)를 관찰하고, 정책(π)이라는 규칙에 따라 행동(At)을 선택
  - 행동에 따른 보상(Rt+1)을 받고, 환경은 다음 상태(St+1)로 바뀜
  - 상태—행동—보상—상태의 연쇄(궤적)를 에피소드(특정 목표 도달)까지 반복해 샘플 수집
  - Agent의 목표는 할인된 누적보상(return)을 최대화하는 최적의 정책 π^* 를 찾는 것
- 정책(policy)
  - 확률적 정책($\pi(a|s)$): 상태 s에서 행동 a를 선택할 확률 → cf. Gradient-based RL
  - 결정적 정책($\pi(s)=a$): 상태 s가 주어지면 항상 하나의 행동 a만 선택 → cf. Neuro-evolution
- Deep RL: 정책을 신경망(딥러닝)을 활용해 표현하고, 파라미터를 데이터로부터 최적화

1. Introduction

모델-프리(model-free) DRL을 활용한 새로운 마켓 메이킹 프레임워크를 제안

- 기존 방식의 한계
  - 전통적인 수리적 접근 방식: 현실을 지나치게 **단순화한 가정**을 전제로 진행
  - 머신러닝 기반 방식: **이산화된 호가**/예측에 도움이 되는 **추가 신호를 활용하지 않음**
- 새로운 프레임워크 제시
  - 외부 신호(signal) 생성 모듈의 출력을 반영한 새로운 상태 공간(state space)
  - 유연해진 행동 공간(action space)
  - 목표를 효과적으로 반영하는 보상 함수(reward function)
  - 적대적 강화학습(adversarial reinforcement learning)과 neuroevolution 기법을 결합
- 실험 결과
  - 표준 마켓 메이킹 벤치마크 대비 최종 수익은 20~30% 높음/재고 위험 노출 60% 수준

2. Related Work(1/4)

(기존 분석적 방식) MM은 확률적 재고 관리(stochastic inventory control)* 문제로 공식화

* 재고(보유 자산) 변화가 무작위적(stochastic)일 때, 언제 어떤 가격에 사고팔아야 최적 성과를 낼지를 수학적으로 다루는 분야

- **Ho와 Stoll의 연구(1981)**
  - 어느 가격에 매수·매도 주문을 걸어야 최종 시점의 기대 효용(expected terminal utility)이 최대가 될지를 연속적으로 계산. 여기서 '효용'은 PnL(손익)과 위험(risk)을 모두 고려한 값
  - 수학적으로는 **Hamilton-Jacobi-Bellman(HJB) 방정식***을 풀어 매수·매도 호가의 근사해 계산

    * 현재 상태에서 앞으로 최대한의 보상을 얻으려면 어떤 행동을 해야 하는지에 대한 최적화 문제
- **Avellaneda와 Stoikov의 연구(2008):** 처음으로 **MDP 문제**로 정의한 기념비적 연구
  - **드리프트 없는 확산 과정(driftless diffusion)***으로 중간 가격(mid-price)이 움직인다고 **가정**

    * 평균(추세)의 상승/하강이 없고 시계열의 noise만 존재한다고 가정. 가격의 상승/하강 기댓값이 0
  - 모든 포지션을 닫아야 하는 마감 시점(terminal time) T를 설정한 뒤,이 T 주변에서 Taylor 전개를 써서 최적 호가를 근사적으로 계산
- 한계1) 수학적으로 풀기 편하도록 강한(strong, naïve) 가정을 전제
- 한계2) historical data에서 일일이 매개변수(parameter)를 추정해야 한다는 한계

2. Related Work(2/4)

(Non-Deep RL 방식) 행동 공간을 이산화하거나/상태, 재고, 시간 외 추가 신호를 활용하지 않음

- 지금까지 진행된 Non-Deep RL/Deep RL 방식의 한계
  - 행동 공간을 몇십 개 수준으로 이산화(discretize) ↔ 연속적인 행동 공간
  - 상태, 재고, 시간 외 추가 신호를 활용하지 않음
  - 학습된 정책의 해석 가능성(interpretability)을 도입하지 않음
- 새로운 프레임워크 제안
  - 행동 공간의 정교화
  - 추가 신호를 반영하고 신호 통합
  - 정책 해석

2. Related Work(3/4)

1) 2개의 지도학습(supervised learning) 기반의 신호 생성 모듈(SGUs)

2) 위의 output으로 나오는 신호를 활용하는 DRL(deep RL) 모듈을 연결

- 새로운 프레임워크 제안
  - 1) 2개의 지도학습(supervised learning) 기반의 **신호 생성(예측)** 모듈(SGUs)
    - (1) 다음 구간의 가격 변동 범위(price range)
    - (2) 다음 구간의 추세(trend)
  - 2) 위의 output으로 나오는 신호를 활용하는 딥 강화학습(deep RL) 모듈을 연결

2. Related Work(4/4)

(새로운 프레임워크) 기여

- 1) 새로운 상태·행동·보상 설계행동 공간(action space)을 '틱(tick)' 단위로 연속적으로 정의
  - 주식 시장에서 호가가 틱 단위로 움직인다는 점을 반영
  - 기존 방식의 이산화(discretization)나 지나치게 단순한 수리 모델과 달리 현실성이 높음
- 2) 신경망 진화(neuroevolution) 알고리즘으로 DNN Agent 학습
  - 일반적인 Gradient 기반 강화학습 대신 신경망이 상태를 바로 행동으로 매핑하도록 바꿔 **'Noisy gradient'*** 문제와 **'망각(catastrophic forgetting)'***을 줄임

    * Noisy gradient: 보상 신호가 희소하거나 확률적 정책(Gradient 기반 방식에서는 파라미터의 손실함수 계산 시 샘플링 필요)의 gradient 추정 시 샘플마다 높은 noise 가 섞임. 경사 noise 가 높을수록 파라미터 업데이트 시 불안정

    * 망각(catastrophic forgetting): 이전에 잘 학습된 행동 패턴이 새로운 경험을 통해 덮어씌워지며 잊히게 됨
- 3) 적대적 강화학습(adversarial RL) 도입
  - 시장 자체를 흉내 내는 적대자 Agent를 두고 MM Agent의 행동을 직접 방해하도록 설계
  - 행동 공간 자체에 노이즈를 주어 MM Agent가 다양한 상황에서도 강건하게 학습되도록 설계
- 4) 정책 해석 가능성(interpretability) 강화: MM Agent가 어떤 원리로 호가를 결정하는지 설명

3. Formulation(1/4)

1) Market Making Procedure

- 1) 상태 관찰
  - 시각 t에서 현재 재고량(inventory level)과 보조 신호를 합친 상태 S_t를 관찰
- 2) 호가 제출
  - 시각 t에서 행동 A_t를 취함. 행동은 한 단위 크기(unit size)의 매도 주문(ask limit order)과 한 단위 크기의 매수 주문(bid limit order)을 각각 Q_t^{ask}, Q_t^{bid}가격에 거는 것을 의미
- 3) 보상과 재주문
  - 시간이 Δt만큼 흘러 t+Δt가 되면, 보상(reward)을 얻고 다음 쌍의 호가를 다시 제출
  - 단, 현재 재고가 최소 한도(I_{min})이거나 최대 한도(I_{max})인 경우에는 한쪽 주문만 내어 재고가 이 범위를 벗어나지 않도록 조절
- 4) 에피소드 종료
  - 최종 시각 T에 도달하면 한 회차(에피소드) 종료.
  - 양쪽 주문이 모두 체결되면 작은 이익을 쌓게 되고, 한쪽만 체결되면 재고 I_t과 보유 현금 X_t 변화

3. Formulation(2/4)

2) Market Making Agent

- 1) 상태 공간(State space): S_t = [I_t , RR_t , TR_t]
  - I_t (재고량): 현재 보유 중인 주식 수량
  - RR_t (가격 범위 예측): 다음 구간에 거래될 것으로 예상되는 가격 폭
  - TR_t (추세 예측): 다음 구간의 가격 흐름(오를지·내릴지)에 대한 예측
  - 상태 공간의 차원은 3차원(dim = 3)
- 2) 행동 공간(Action space): A_t = [$A_{t,1}$, $A_{t,2}$] = [$Q_t^{ask} - Q_{best,t}^{ask}$, $Q_t^{bid} - Q_{best,t}^{bid}$]
  - $A_{t,1} = Q_t^{ask} - Q_{best,t}^{ask}$: 매도 호가 위치가 최우선 매도 호가보다 몇 틱 위인지
  - $A_{t,2} = Q_t^{bid} - Q_{best,t}^{bid}$: 매수 호가 위치가 최우선 매수 호가보다 몇 틱 아래인지
  - 틱(tick)은 가격이 바뀔 수 있는 최소 단위
  - 한 번에 스프레드(두 호가 차이)와 공격성(시장 가격에 가까울수록 체결 가능성↑) 조절 가능

3. Formulation(3/4)

2) Market Making Agent

- 3) 보상 함수(Reward function)
  - 다음 시간 t+1에 얻는 보상 R_{t+1}는 다음과 같이 계산합니다
  - $R_{t+1} = (Q_t^{ask} - M_{t+1}) \cdot 1\{\text{매도호가 체결}\}$

    $+ (M_{t+1} - Q_t^{bid}) \cdot 1\{\text{매수호가 체결}\}$

    $- \lambda \cdot |I_{t+1}| M_{t+1}$
  - 중간 가격(mid-price)$1\{\cdot\}$: 해당 호가가 체결되었으면 1, 아니면 0
  - λ: 재고 보유량($|I_{t+1}|$)에 페널티를 주는 계수
  - 체결된 호가만큼 스프레드를 챙기되 재고 보유량이 커지면 벌점을 부과해 과도한 재고 축적 방지

3. Formulation(4/4)

3) Adversary

- 1) 상태 공간: $S'_t = [I_t, 1\{\text{매도호가 체결}\}, 1\{\text{매수호가 체결}\}]$
  - 적대자는 재고량과 체결 여부 관찰 가능
- 2) 행동 공간: $A'_t = [A'_{t,1}, A'_{t,2}] = (A^{dist}_{ask,t} - A_{t,1}, A^{dist}_{bid,t} - A_{t,2})$
  - MM Agent 가 내려고 한 호가를 엉뚱하게 조정
    - Agent 가 '믿고 있는 행동'과 '실제 체결 결과' 사이에 불일치가 발생
  - $A'_{t,1} = A^{dist}_{ask,t} - A_{t,1}$: MM Agent의 매도 호가를 틱 단위로 얼마나 더 높이거나 낮출지
  - $A'_{t,2} = A^{dist}_{bid,t} - A_{t,2}$: MM Agent의 매수 호가를 틱 단위로 얼마나 더 높이거나 낮출지
- 3) 보상 함수: $R'_{t+1} = -R_{t+1}$
  - MM Agent의 보상의 반대. 스프레드를 작게 만들고 재고를 키워 위험을 높이는 방식으로 공격

4. Implementation(1/7)

1) Market Making Agent Details

- 1) 신경망 구조
  - Feed-forward 방식의 fully-connected NN 을 사용
  - 현재 최우선 매수·매도 호가(best bid/ask)로부터 몇 틱(tick)만큼 떨어진 가격에 주문을 낼지 계산
- 2) 출력값 조정
  - 신경망이 내는 실수 값을 틱 단위로 맞추기 위해, 먼저 스케일링(scaling) 계수를 곱하고
  - 이후, 반올림(round) 연산을 거쳐 실제 주문 가격 틱 수로 만듦
- 3) 구조
  - 2개의 hidden layer + 각 층에 32개의 뉴런 배치
  - Activation function
    - Hidden layer: ReLU
    - Fully-connected layer: linear → 호가가 스프레드 안팎 어디든 나올 수 있도록 허용
  - 얕은(shallow) 네트워크도 DRL 분야에선 성공 사례가 많았음

4. Implementation(2/7)

2) Adversary Agent Details

- 1) 신경망 구조(MM Agent와 동일)
  - Feed-forward 방식의 fully-connected NN 을 사용
  - 현재 최우선 매수·매도 호가(best bid/ask)로부터 몇 틱(tick)만큼 떨어진 가격에 주문을 낼지 계산
- 2) 출력값 조정 (MM Agent와 동일)
  - 신경망이 내는 실수 값을 **틱 단위로 맞추기 위해*** 먼저 스케일링(scaling) 계수를 곱하고

    * 호가 쪼개기 방지: 미세한 단위로 가격을 받도록 허용하면 호가 스프레드가 거의 0에 수렴하면서 주문만 쌓여 시장이 혼잡해지는 문제 발생
  - 이후, 반올림(round) 연산을 거쳐 실제 주문 가격 틱 수로 만듦
- 3) 구조
  - 1개의 hidden layer + 각 층에 12개의 뉴런 배치: 복잡한 전략을 구현하는 게 아니고 '교란'만 목적
  - Activation function
    - Hidden layer: ReLU
    - Fully-connected layer: linear → 호가가 스프레드 안팎 어디든 나올 수 있도록 허용

4. Implementation(3/7)

3) Signal Generating Units(SGUs)

- 1) 첫 번째 SGU
  - 그래디언트 부스팅(gradient boosting) 모델
  - 실제 가격 범위(realized price range)(= 변동성, volatility) 을 예측
  - 1) 레이블(Output)

$$y_i = \max\left(P_i^{\text{buy}}\right) - \min\left(P_i^{\text{sell}}\right). \qquad (20)$$

    - 수정된 실현 가격 범위 yi: 가격 범위 변동성 + 호가 사이드(매수/매도)까지 표현 가능
      - P_i^{buy}: i=t 일 때, 매수 체결된 모든 가격의 집합
      - P_i^{sell}: i=t 일 때, 매도 체결된 모든 가격의 집합
    - 한쪽 체결이 없을 때는(예: 매수 체결만 없거나 매도 체결만 없는 경우, 최우선 호가 평균으로 대체
    - yi 는 틱 크기(0.01 USD) 단위로 반올림해 소수점 둘째 자리까지 표현

4. Implementation(4/7)

3) Signal Generating Units(SGUs)

- 2) 23개 피처(Input) → 최종: 20개 피처 사용(XGBoost에서 피처 중요도 기준으로 하위 3개 피처 제거)

구분	피처 이름	개수 / 기간
거래 횟수	과거 거래 건수	p = 1,2,3,5,10 (5개)
스프레드	구간 시작 시점의 최우선 매도−매수 호가 차이	1개
거래량 불균형	(매수량−매도량)/(매수량+매도량)	1개
VWAP	과거 거래량 가중 평균 가격	r = 1,3,5 (3개)
가격 기울기	과거 가격 vs 시간 선형회귀(OLS) 기울기	s = 1,3,5 (3개)
시간대	하루 중 시간 (0~23시)	1개
총 거래량	해당 구간의 전체 체결량 합	1개
업틱 비율	가격 상승 틱 비율 (%)	1개
대형 매수 건수	기준 이상 크기 매수 주문 건수	1개
대형 매도 건수	기준 이상 크기 매도 주문 건수	1개
시차별 레이블	수정된 실현 가격 범위 레이블 $y_{i-1} \cdots y_{i-5}$	L = 1~5 (5개)
합계		**23개**

4. Implementation(5/7)

3) Signal Generating Units(SGUs)

- 2) 두 번째 SGU
  - LSTM(Long Short-Term Memory) 모델
  - 다음 구간의 추세(trend) 를 예측
    - 1) 피처(Input)
      - 시차 시계열(past pseudo-returns)
        - L=1…10 구간 전까지의 의사 수익률 레이블을 그대로 예측 입력으로 사용
        - 모든 입력 값은 Z-스코어 표준화로 변환

4. Implementation(6/7)

3) Signal Generating Units(SGUs)

- 2) 레이블(Output)
  - 직전 구간의 pseudo-중간 가격 m_{i-1} 대비 상대 수익률(단순 수익률)

$$y_i = \frac{m_i - m_{i-1}}{m_{i-1}}. \tag{23}$$

- Pseudo-중간 가격 정의
  - 1) 매수-매도 체결이 모두 있는 경우
  - 2) 매수 체결이 없는 경우
  - 3) 매도 체결이 없는 경우
  - 4) 둘 다 없는 경우
- 단, $\bar{P}$는 해당 집합 P의 평균을 의미

$$m_i = \begin{cases} \frac{1}{2}\left(\max\left(P_i^{\text{buy}}\right) + \min\left(P_i^{\text{sell}}\right)\right) & \text{if } P_i^{\text{buy}} \neq \emptyset, P_i^{\text{sell}} \neq \emptyset \\ \frac{1}{2}\left(\overline{P_i^{\text{ask}}} + \min\left(P_i^{\text{sell}}\right)\right) & \text{if } P_i^{\text{buy}} = \emptyset, P_i^{\text{sell}} \neq \emptyset \\ \frac{1}{2}\left(\max\left(P_i^{\text{buy}}\right) + \overline{P_i^{\text{bid}}}\right) & \text{if } P_i^{\text{sell}} = \emptyset, P_i^{\text{buy}} \neq \emptyset \\ \frac{1}{2}\left(\overline{P_i^{\text{ask}}} + \overline{P_i^{\text{bid}}}\right) & \text{else ,} \end{cases} \tag{22}$$

4. Implementation(7/7)

4) Training: neuro-evolution 방식으로 정책 최적화

- 1) 초기화(initialization)
  - 직교 행렬(orthogonal) 초기화(이득값 gain=0.9)와 편향(bias)을 0.005로 고정. 실험으로 결정
- 2) 적대자와 번갈아 학습
  - MM Agent의 정책을 고정한 채 적대자를 학습시키고, 적대자의 정책을 고정한 채 MM Agent 학습
  - Cf. 만약 두 에이전트를 동시에 파라미터를 바꾸며 학습하면,
    - 서로가 매 스텝마다 '이전 상태의 상대'를 가정하고 업데이트한 값이 엉키면서 학습 불안정해짐
- 3) 입력 정규화(normalization)
  - 모든 입력 피처(feature)를 Z-스코어(z-score) 방식으로 표준화

5. Experiments(1/7)

1) Evaluation Dataset

- 1) 데이터 출처: 암호화폐 거래소인 비트스탬프(Bitstamp)
- 2) 기간: 2020.09.01~2020.09.30. 총 30거래일(하루 약 24시간씩, 총 720 시간)
- 3) 대상 상품: 비트코인/미국 달러(BTC/USD)
- 4) 내용: 틱 단위(tick-by-tick)의 거래 기록과 호가(주문) 기록
  - 1) 거래(trades) 데이터: 체결 당시의 ID, 타임스탬프, 거래 가격, 거래량 기록
    - 총 체결 횟수는 660,337건으로, 분당 약 1,529건
    - 전체 거래의 54.94%는 매도(sell) 체결, 거래량 기준으로는 52.93%가 매수(buy) 체결
    - 거래당 평균 크기(mean)는 0.3194 BTC, 중간값(median)은 0.0921 BTC
  - 2) 호가(quotes) 데이터: 최우선 호가(best bid/ask)가 바뀔 때마다 타임스탬프, 가격, 수량 기록
    - 총 기록 건수는 3,036,073건으로, 분당 약 7,028건이 변화. 이 중 87.41%가 가격 변동(change in price)을 의미

5. Experiments(2/7)

1) Evaluation Dataset

- 5) 비트스탬프의 틱 크기(tick size)는 0.01 USD. 시장 조성 한 주기(MM period)의 길이는 19틱으로 설정(약 1.622초)
- 6) 실험을 위해 데이터를 64 : 16 : 20 비율로 분할
  - S1(64%): SGU(신호 생성 모듈) 학습용
  - S2(16%): SGU 검증(validation)용
  - → S1과 S2로 SGU가 가격 범위(range)와 추세(trend)를 얼마나 잘 예측하는지 확인
  - 그 예측값을 이용해 DRL 유닛을 학습·검증·테스트할 데이터 S3(20%)를 다시 세분해 DRL 학습(train) : 검증(validation) : 테스트(test)" 용으로 분할
- **Figure 2**: 30일 동안의 BTC/USD 가격 움직임

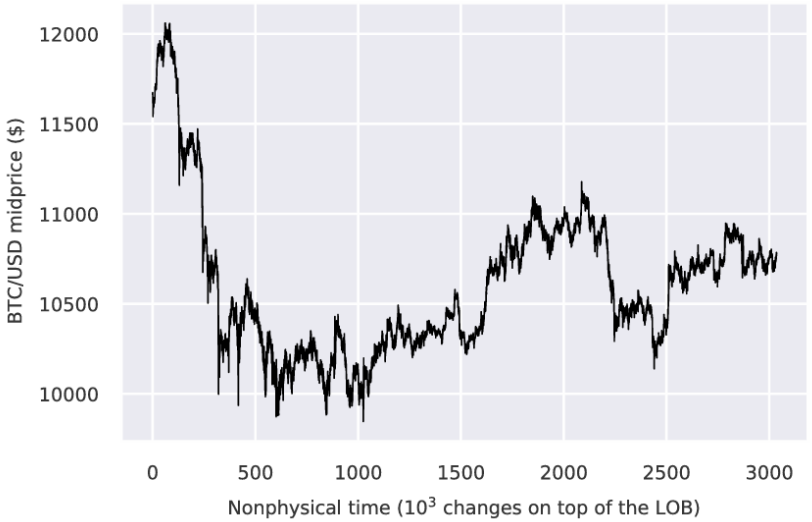


FIGURE 2. **Evolution of the BTC/USD mid-price in the full dataset.**

5. Experiments(3/7)

2) Benchmarks: 2가지 벤치마크

- 1) 고정 오프셋 + 재고 제약 전략(Fixed Offset with Inventory Constraints, FOIC)
  - FOIC 파라미터 (N,M,c)
    - N, M: 최우선 매수(bid)·매도(ask) 호가에서 떨어진 틱(offset) 수
    - c: 허용 재고 범위($-c \le I_t \le c$)
  - Ex. FOIC(1,1,5)
    - (1) 매수 호가는 최우선 매수 호가보다 1틱 낮게, 매도 호가는 최우선 매도 호가보다 1틱 높게 고정
    - (2) 재고가 +5(또는 -5)에 도달하면 반대쪽 호가만 제출 가능하여 재고 범위를 벗어나지 않음

5. Experiments(4/7)

2) Benchmarks: 2가지 벤치마크

- 2) 재고 선형 가중 + 재고 제약 전략(Linear in Inventory with Inventory Constraints, LIIC)
  - LIIC 파라미터 (a,b,c)
    - a: 스프레드 절반 폭(ask·bid 호가가 mid-price 기준 ±a)
    - b: 재고 가중치(재고 I_t가 많을수록 호가를 동적으로 조정)
    - c: 허용 재고 범위($-c < I_t < c$)
  - 호가 계산
    - $Q_t^{ask} = M_t + a + b \cdot I_t$
    - $Q_t^{bid} = M_t - a + b \cdot I_t$
    - 무관심 가격(indifference price) $M_t + b \cdot I_t$를 중심으로 ±a만큼 균일한 스프레드를 유지
      - 무관심 가격: 사거나 팔아도 기대 이익이 같아지는 가격
      - 현재 재고가 많으면(양수) 판매 유인을 높이기 위해 기준 가격을 올리고,
        재고가 적으면(음수) 구매 유인을 높이기 위해 기준 가격을 낮춰서 재고 리스크 관리

5. Experiments(5/7)

3) Performance and risk metrics

- 1) 에피소드 수익(Episode Return, G_0)
  - 식(1)에 따라 첫 상태부터 마지막 상태까지 할인 누적보상의 합

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \qquad (1)$$

- 2) 최종 자산(Terminal Wealth, W_t): t=T 시점의 포트폴리오 총 가치
- 3) 평균 절대 재고(MAP, Mean Absolute Position)
  - 시장 조성 에이전트가 보유한 재고(포지션)의 평균적인 크기
  - t 시점까지 완료된 M단계 동안의 재고 절댓값 평균. **재고 리스크**를 직접 반영

$$\mathrm{MAP}(t) = \frac{1}{M} \sum_{k=1}^{M} |I_{k\Delta t}|, \qquad (15)$$

5. Experiments(6/7)

3) Performance and risk metrics

- 4) 최대 낙폭(MDD, Maximum Drawdown)
  - 전체 기간 중 자산이 고점에서 저점으로 얼마나 크게 떨어졌는지

$$\mathrm{MDD}(T) = \max_{\tau \in (0,T)} \left[\max_{s \in (0,\tau)} W_s - W_\tau \right], \qquad (16)$$

- 5) PnL-to-MAP 비율(PnLMAP)
  - 수익성(W_t)과 재고 리스크(MAP)를 동시에 고려한 지표입니다.

$$\mathrm{PnLMAP}(t) = \frac{W_t}{\mathrm{MAP}(t)}, \qquad (17)$$

5. Experiments(7/7)

4) Reinforcement Learning Environments Details

- 1) 재고 제한: I_max = +2, I_min = -2
- 2) 재고 페널티 계수(위험회피계수): λ = 0.15
- 매우 보수적인(위험회피 성향이 강한) 시장 조성가를 모델링하기 위한 설정

6. Results(1/3)

1) Figure 3, Table 1

- 테스트용 데이터(S3)의 out-of-time에서 성능 평가
- Figure 3, Table 1: 네 가지 전략의 PnL-to-MAP 비율 (PnLMAP) 변화
  - 1) DRL 에이전트(적대자 교란 포함)
  - 2) DRL2 에이전트(적대자 없이 학습된 버전)
  - 3) FOIC(0,0,2)
  - 4) GLFT(γ=0.001, 기타 파라미터 최적화)
    - GLFT의 리스크 회피 계수 γ는 테스트 데이터에서 PnLMAP이 최대가 되도록 조정
    - 다른 벤치마크들은 DRL과 동일한 재고 한도 (I_max=2, I_min=-2)를 적용해 공정하게 비교

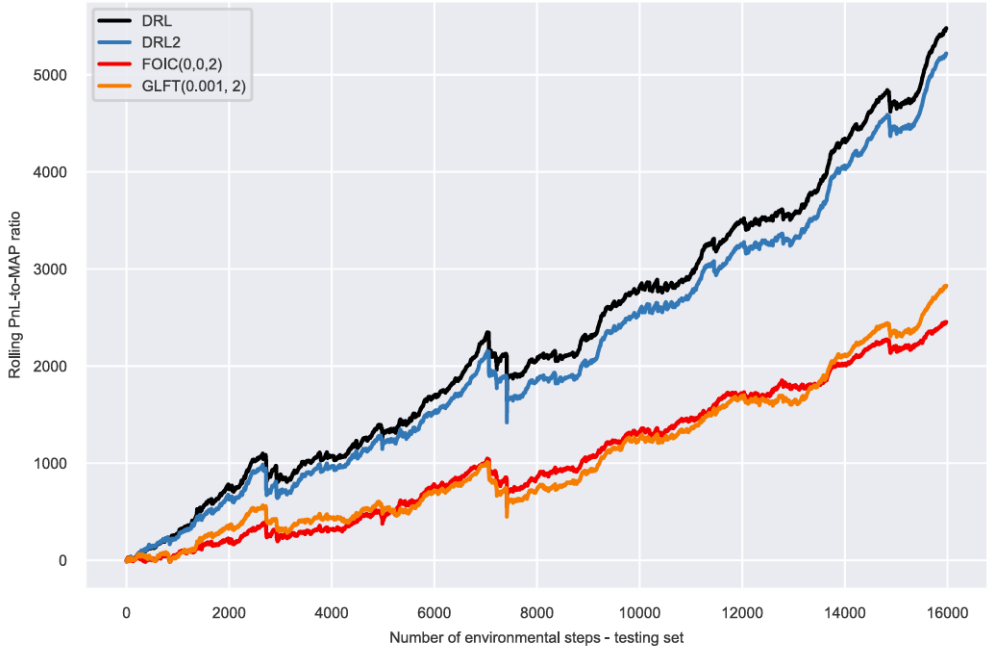


FIGURE 3. Rolling PnL-to-MAP ratio on the testing set for four of the considered strategies. The DRL variants significantly outperform the benchmarks, indicating more favorable return-to-risk performances.

TABLE 1. DRL agent vs benchmarks comparison.

Strategy	Ep. return	W_T	MAP	MDD	PnLMAP(T)
DRL agent	2909.7	4066.7	0.742	−0.289	5478.1
DRL2 agent	2809.9	3838.8	0.736	−0.331	5217.8
FOIC(0, 0, 2)	−16254.6	3072.4	1.252	−0.475	2454.8
GLFT(0.01)	−12799.9	2388.8	0.992	−0.739	2407.3
GLFT(0.001)	−15028.9	3333.4	1.178	−0.557	2828.7
GLFT(0.0001)	−16065.4	3280.9	1.247	−0.554	2630.7

6. Results(2/3)

1) Figure 3, Table 1: DRL 에이전트가 모든 지표에서 우수

- 1) FOIC(0, 0, 2)와 비교: 최종 자산(terminal wealth)은 30% 이상 높으면서, 재고 리스크(MAP)는 그 전략의 60% 미만에 불과
- 2) GLFT(0.001)과 비교: 모든 지표에서 우월한 성과
  - 시간에 따라 DRL과 벤치마크 간 격차가 꾸준히 선형으로 벌어지는 경향
- 3) DRL2와 비교: 높은 최종 자산과 더 낮은 최대 낙폭(MDD)을 기록했지만,평균 재고(MAP)는 약간 높았음. '적대자 교란(AD)을 포함한 버전'이 언제 재고를 조금 더 쌓아도 손실 없이 이익을 지킬 수 있는지를 더 잘 학습했기 때문으로 보임
- 약 7,300번째 스텝: 비트코인/달러 가격이 갑자기 크게 움직이면서 모든 전략의 성과가 일시적으로 떨어지는 모습

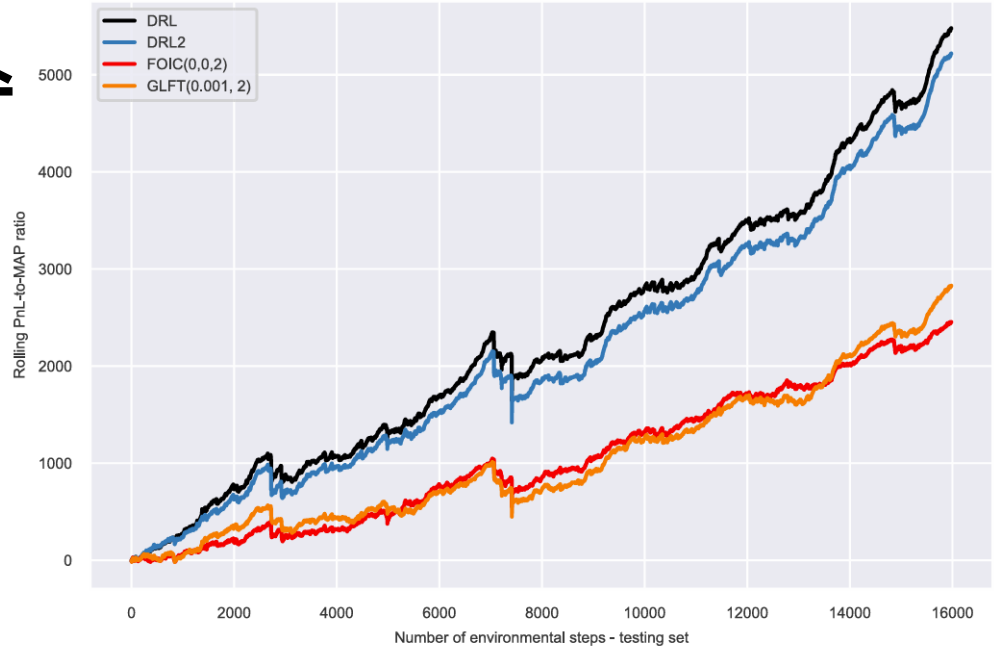


FIGURE 3. Rolling PnL-to-MAP ratio on the testing set for four of the considered strategies. The DRL variants significantly outperform the benchmarks, indicating more favorable return-to-risk performances.

TABLE 1. DRL agent vs benchmarks comparison.

Strategy	Ep. return	W_T	MAP	MDD	PnLMAP(T)
DRL agent	2909.7	4066.7	0.742	−0.289	5478.1
DRL2 agent	2809.9	3838.8	0.736	−0.331	5217.8
FOIC(0, 0, 2)	−16254.6	3072.4	1.252	−0.475	2454.8
GLFT(0.01)	−12799.9	2388.8	0.992	−0.739	2407.3
GLFT(0.001)	−15028.9	3333.4	1.178	−0.557	2828.7
GLFT(0.0001)	−16065.4	3280.9	1.247	−0.554	2630.7

6. Results(3/3)

1) Figure 4: 재고 리스크 관리

- Figure 4: 테스트 기간의 작은 구간에서 DRL 에이전트가 어떻게 행동했는지를 보여줌
- 재고량(재고 리스크 관리)
  - 항상 -1~1 틱 사이를 오가며(테스트 시간의 89.06%) 재고를 0 주변에 유지하려는 전형적인 시장 조성 행태
- DRL 에이전트는 '적대자 교란'을 활용해 벤치마크뿐 아니라, 단순 DRL 학습 버전(DRL2)까지 뛰어넘는 가장 안정적이고 높은 수익-위험 비율을 달성

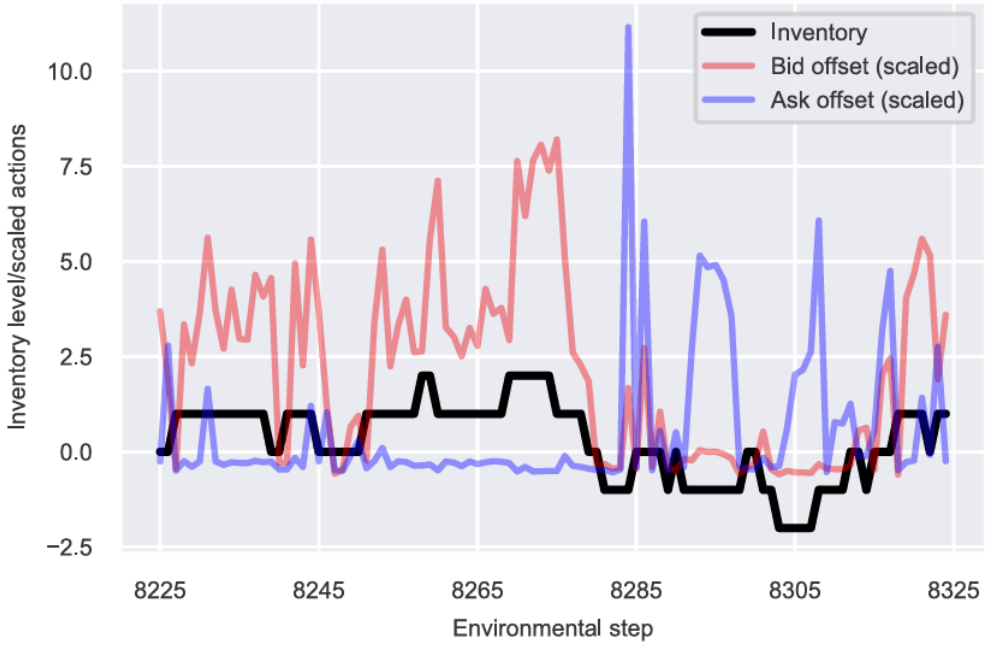


FIGURE 4. Typical DRL agent behavior. Observe that bid (ask) quotes are posted more conservatively (aggressively) when the inventory is positive, and vice versa. The actions are shown scaled (factor 0.2).

7. Interpretability(1/4)

부분 의존도 플롯(Partial Dependence Plot, PDP)

- 특정 입력 변수 x가 모델 출력 h(x)에 미치는 순수(marginal) 효과를 보여주는 그래프
  - 나머지 모든 입력 변수 집합을 Y라고 하면, 부분 의존도는 다음과 같이 정의

$$h_x = \mathrm{E}_Y\left[h\left(x, Y\right)\right] = \int h\left(x, Y\right) \mathrm{d}P\left(Y\right). \qquad (18)$$

- 학습 데이터 {(xi Yi), i=1, 2, ..., n} 를 사용해 x를 고정한 상태에서 평균을 취함으로써 간단히 추정 가능

$$\hat{h}_x(x) = \frac{1}{n}\sum_{i=1}^{n} h\left(x, Y_i\right). \qquad (19)$$

- Cf. '다른 변수는 평균적으로 이 정도일 때'라는 조건 하의 평균 효과를 보여줄 뿐 인과관계(causality)를 증명하는 것은 아님. 학습된 DRL 에이전트가 **어떤 경향으로 호가를 결정하는지** 살펴보는 데 유용한 툴

7. Interpretability(2/4)

1) Figure 5: 재고량에 따른 호가 오프셋(PDP - Inventory)

- 매수 호가 오프셋
  - 1) 재고량이 음수인 경우: '최우선 매수 호가'로부터의 오프셋(offset) 선형 증가
  - 2) 재고가 양수로 갈수록: 기울기가 훨씬 가파르게 증가
- 매도 호가 오프셋
  - 1) 재고가 양수로 갈수록: 선형에 가깝게 감소
    - 재고가 많을 때는 매도 호가를 공격적으로(높게), 매수 호가를 보수적으로(낮게) 설정해 팔도록 유도
  - 2) 재고량이 음수인 경우: 매수 호가를 공격적으로 처리해 사도록 유도
- 기울기가 매수 호가 쪽이 더 가파른 이유
  - 양쪽의 주문량(도착 강도) 비대칭성 때문

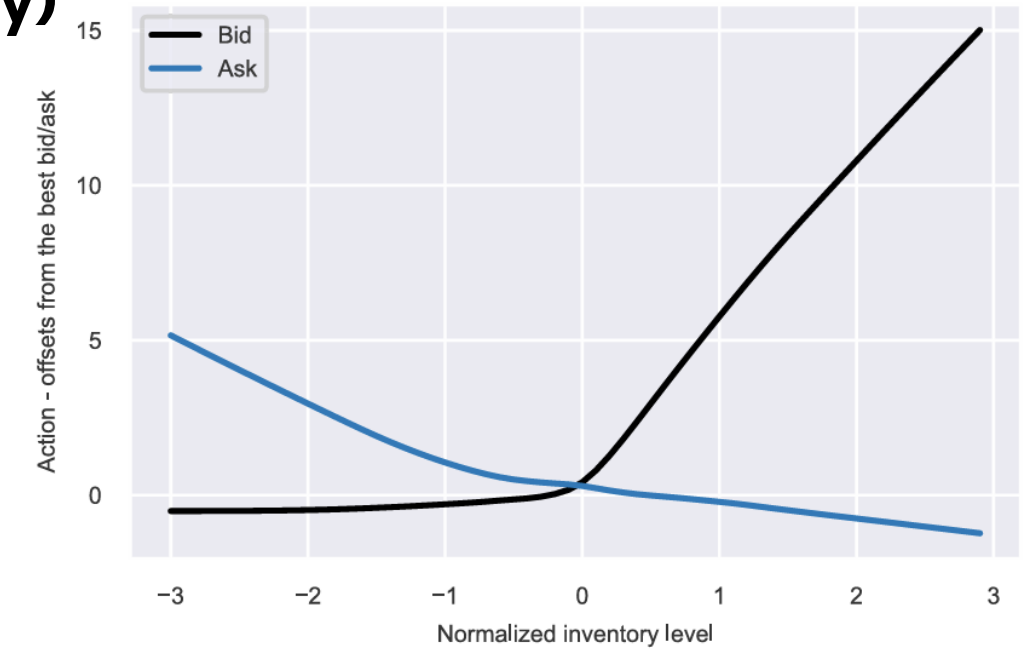


FIGURE 5. **Dependence of the learned offsets from the best bid/ask (actions) on the normalized inventory level.**

- 재고 I=0에서 교차
  - 더 팔거나 사고 싶은 동기가 사라짐
  - Skew=0

7. Interpretability(3/4)

2) Figure 6: 가격 범위 예측에 따른 호가 오프셋(PDP - Price Range)

- 정규화된 가격 범위 신호(realized price range)가 커질 때
  - 양쪽 호가 오프셋이 모두 증가하는 모습
  - 가격 변동 폭이 클 것으로 예측되면,
    - 더 큰 스프레드를 노리기 위해 어느 쪽이든 보수적으로(스프레드 확대) 호가를 내는 것이 유리

    * 1) 스프레드(ask-bid)가 넓어져서 한 번 체결시 얻는 수익 증가

    * 2) 체결 확률이 낮아지며 추가 리스크(불리한 체결) 방지 가능
- 양쪽 오프셋 합계(현재 스프레드에 Agent가 더 추가한 폭)
  - 역시 예측 값이 커질수록(= 변동성이 커질수록) 스프레드 단조 증가(convex)

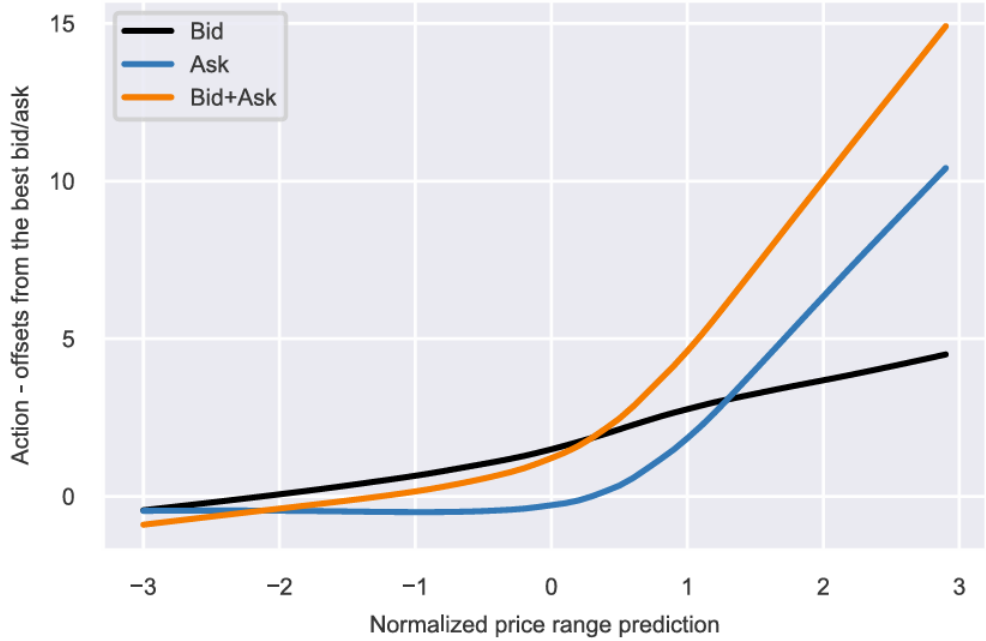


FIGURE 6. Dependence of the learned offsets from the best bid/ask (actions) on the realized price range predictions (the signal from the SGU1).

7. Interpretability(4/4)

2) Figure 7: 추세 예측에 따른 호가 오프셋(PDP - Trend)

- 추세 신호는 예측력이 약해서 **y축 범위가 더 작게*** 나타남

  * Y축 범위가 크게(= 스프레드를 크게 넓힘)할 만큼 확실한 방향성이 없음을 의미

- 추세가 마이너스(하락 예상)일 때는
  - 매도 호가를 더 공격적으로(매도 유도) 설정하고
    매수 호가는 보수적으로 설정
- 추세가 플러스(상승 예상)일 때는
  - 그 차이가 줄어들지만, 여전히 **매도·매수 공격성 차이*** 존재

  * 스프레드 차이가 언제나 양수(ask>bid): '스프레드 수익' 보다 공격적으로 재고 늘리지 않음

- LOB 구조나 미시적 주문 분포 비대칭 때문일 가능성이 높음
- 재고량이 작을수록(위험회피 성향이 강할수록) 양쪽 체결 확률이 비슷하도록 호가를 조정해, 한쪽이 과도하게 채워지는 것을 막는 전략을 학습했음을 알 수 있음

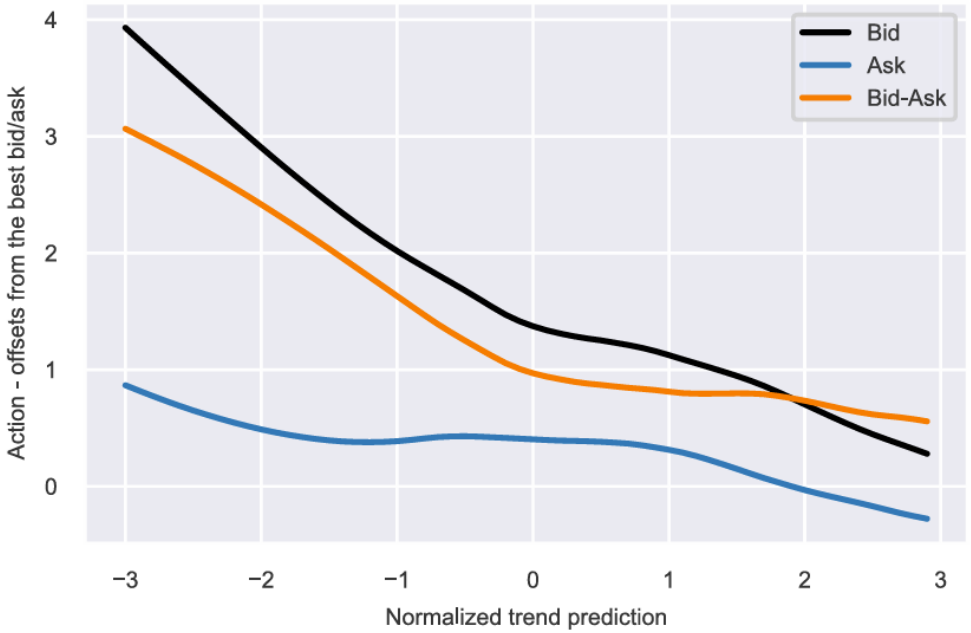


FIGURE 7. Dependence of the learned offsets from the best bid/ask (actions) on the trend predictions (the signal from the SGU2).

8. Conclusion

결론

- 1) 새로운 프레임워크 제시
  - 독립적으로 학습된 두 개의 신호 생성 모듈(SGUs)에서 나온 예측 신호를 상태 공간에 통합하고,
  - 틱 단위(tick-based)이면서도 연속적인 행동 공간을 사용하는 DRL 기반 시장 조성 프레임워크
  - 신경망 진화(neuroevolutionary) 기법과 적대적 강화학습(adversarial RL) 아이디어 결합
- 2) 모델 해석 가능성
  - 부분 의존도 플롯(PDP)을 활용해 학습된 정책을 설명