ETH zürich

Deep Reinforcement Learning for Double Auction Processes

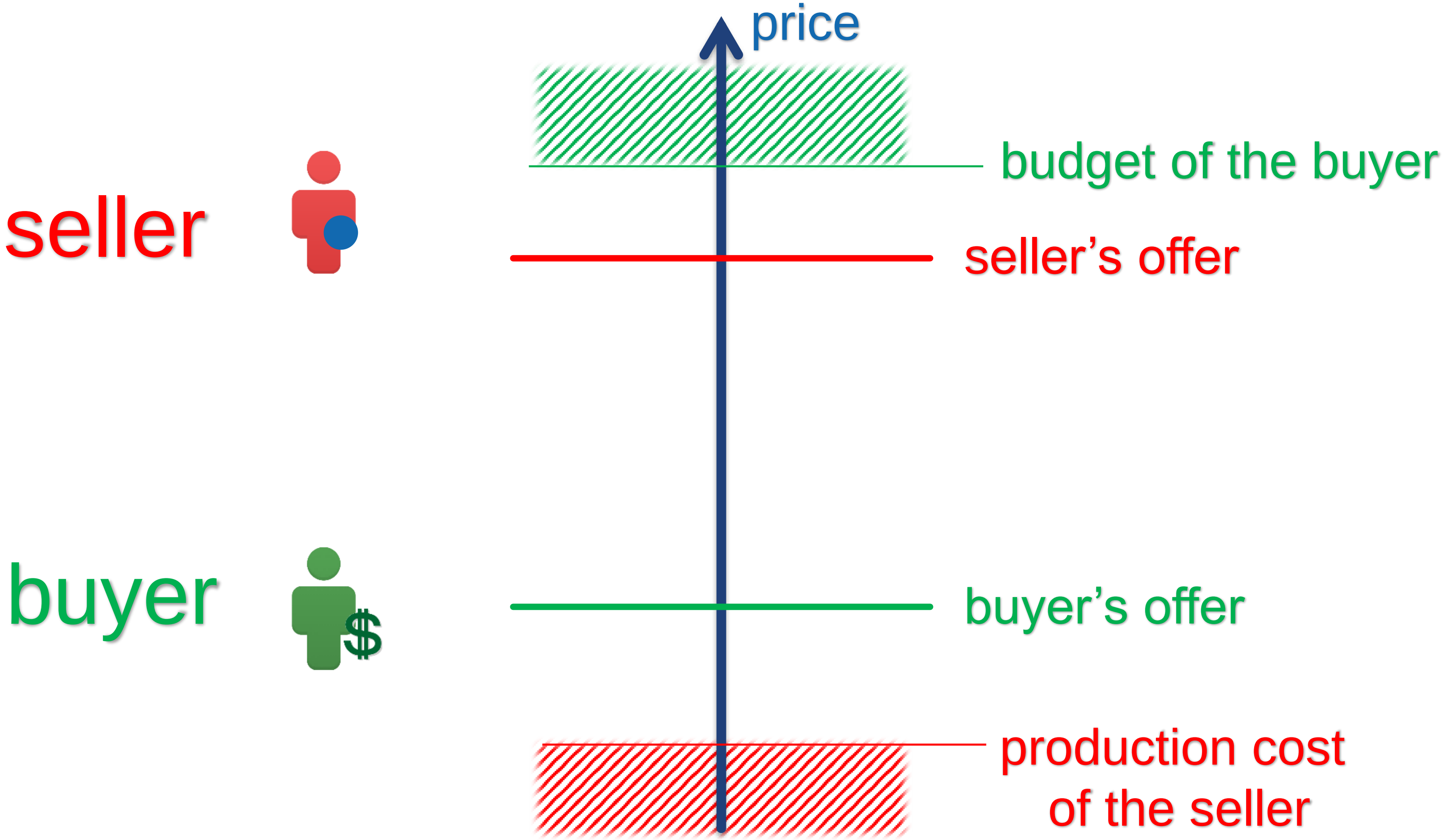
Batuhan Yardim

Aleksei Khudorozhkov

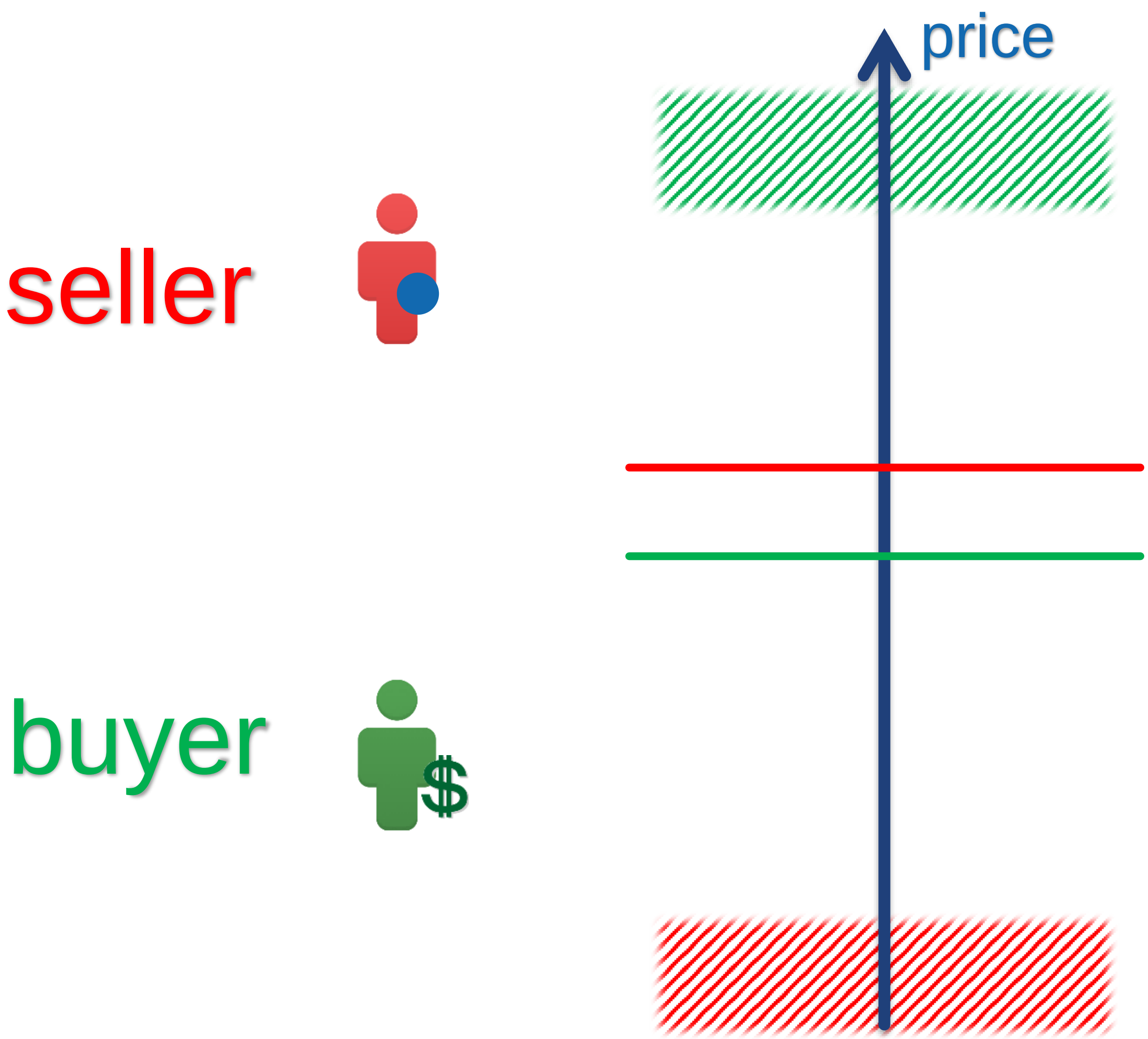
Ka Rin Sim

Neri Passaleva

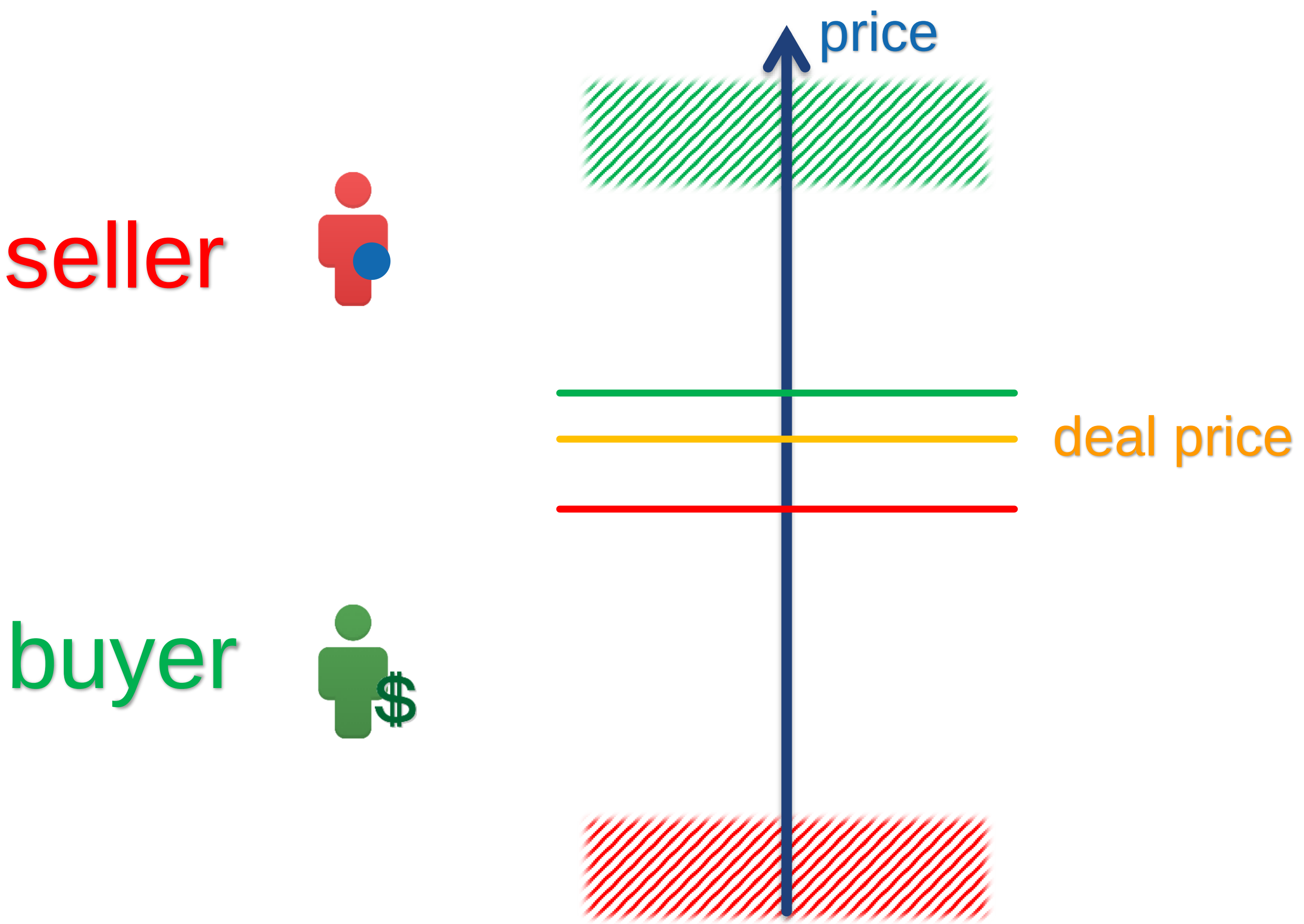
Double auction



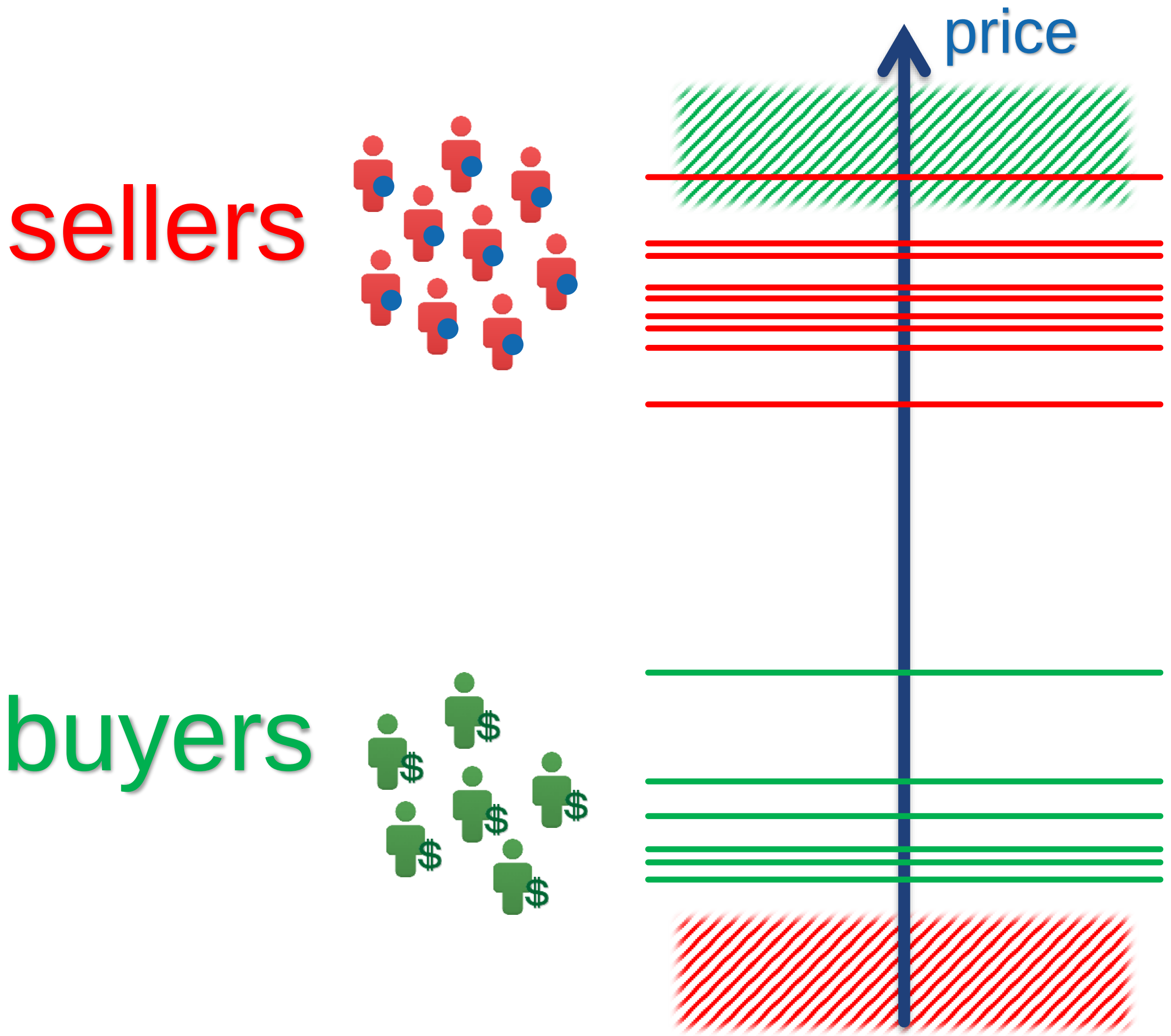
Double auction



Double auction



Double auction

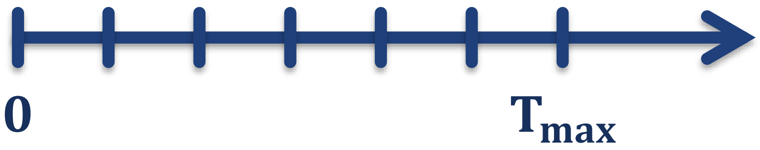


Each agent wants to maximize the reward

For this they can choose different strategies

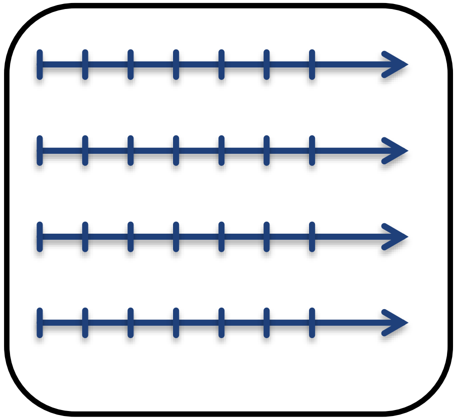
Market environment

- Each round consists of time steps



  - Round terminates when T_{max} is reached or no more deals can be made

- Each game consists of rounds



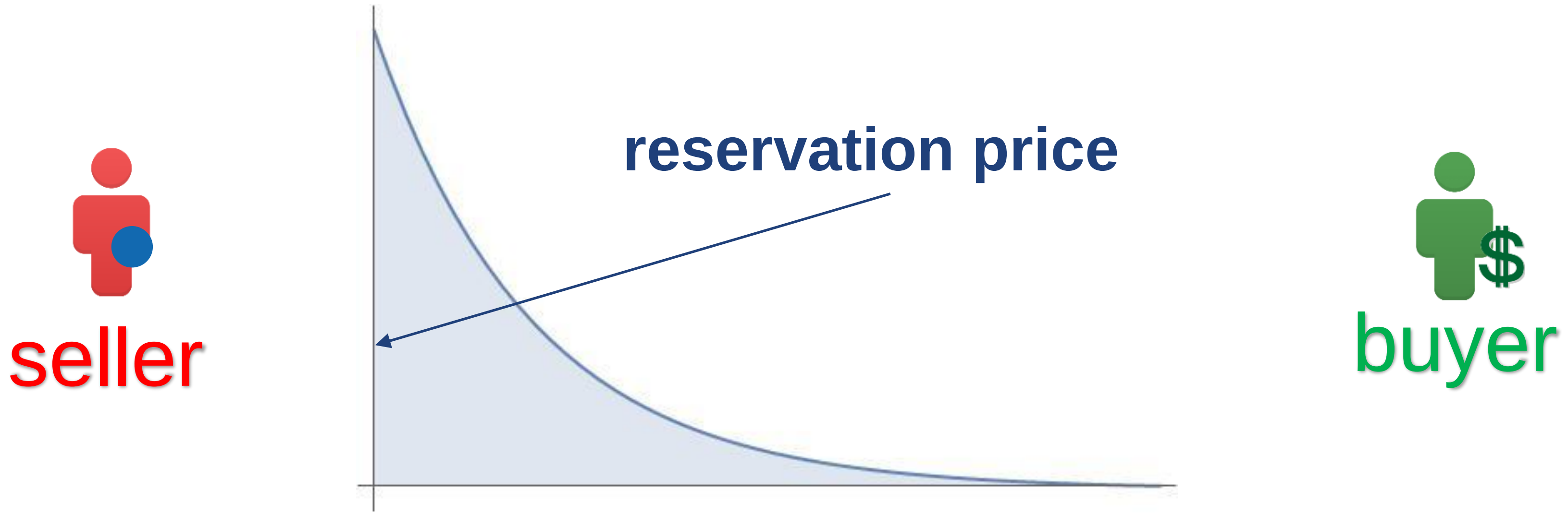
  - Agents can have memory about the previous rounds
  - Between the games agents can learn and adjust their strategy

Observations of agents

- After each time step an agent receives observations from the market environment.
- Core observations are:
  - The last offer of the agent
  - Current time step
  - bool: if the agent managed to make a deal in the previous round
- Other observations might be included:
  - Last offers of agents of the same/opposite side
  - Reservation prices of agents of the same/opposite side
  - Information about completed deals in the current round
  - The maximum time steps in a round
  - The total number of buyers/sellers
  - The number of buyers/sellers who hasn’t made a deal yet

Zero-Intelligence Agent (ZIA)

The agent randomly chooses the next offer according to the distribution and the reservation price



This means that no observation is needed in order to perform the offer

Linear Markov Decision Agents (LMDA)

- The new demand is a linear combination of the agent’s observations

Demand at current time

Demand at previous time step

$$d_i = \alpha d_{i-1} + \beta s + n_i$$

Boolean indicator of previous round outcome

Noise

$$s = \begin{cases} 0 & if\ unsuccessful \\ 1 & if\ successful \end{cases}$$

Price Aggressive Agents (PAA)

s is used as an indicator for whether the agent should be aggressive or not

If **s** is **TRUE:**

$$d_i = \alpha d_{i-1} + n_i$$

If **s** is **FALSE**

$$d_i = (\alpha + \varepsilon) d_{i-1} + n_i$$

ε is the agent's level of aggressiveness

The agent becomes aggressive after an unsuccessful round

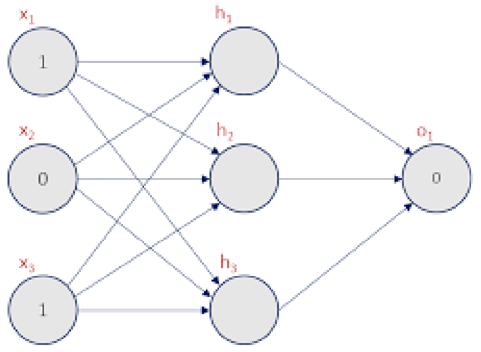
Deep RL Agents

- Learning through Deep Deterministic Policy Gradient (DDPG) framework

GAUSSIAN EXPLORATION MODEL

$$d_i = \pi_{\theta_i}(o_i) + \mathcal{N}(0, \sigma_i)$$

Parametrized Agent's Policy

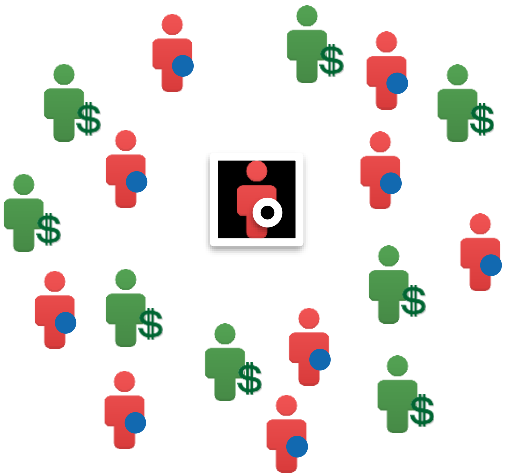


Fully-connected 3-layer neural network

Deep RL Agents

ENVIRONMENT

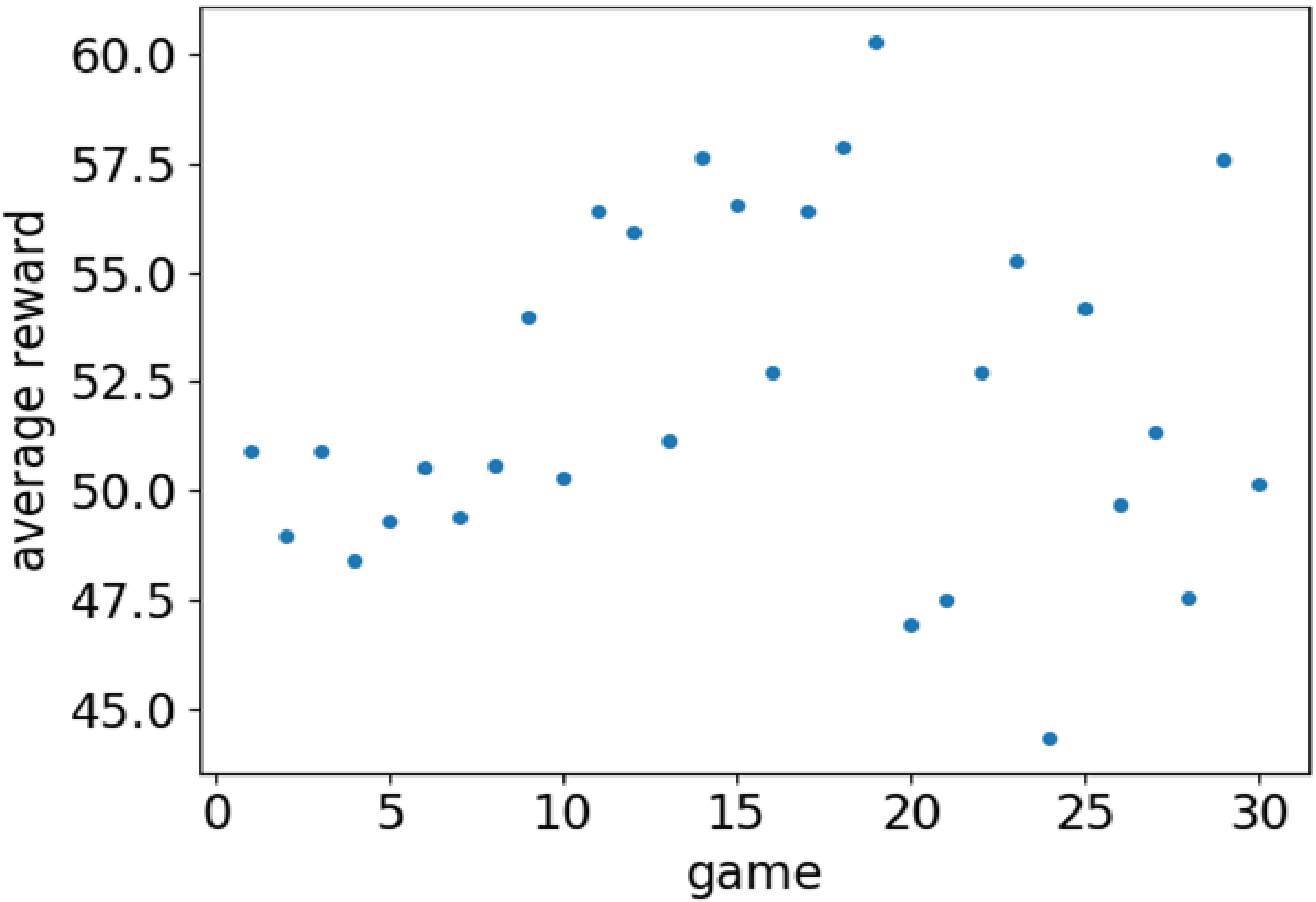
- ONLY a single RL agent is trained
- The market can be filled with:
    - ZIA
    - LMDA
    - PAA



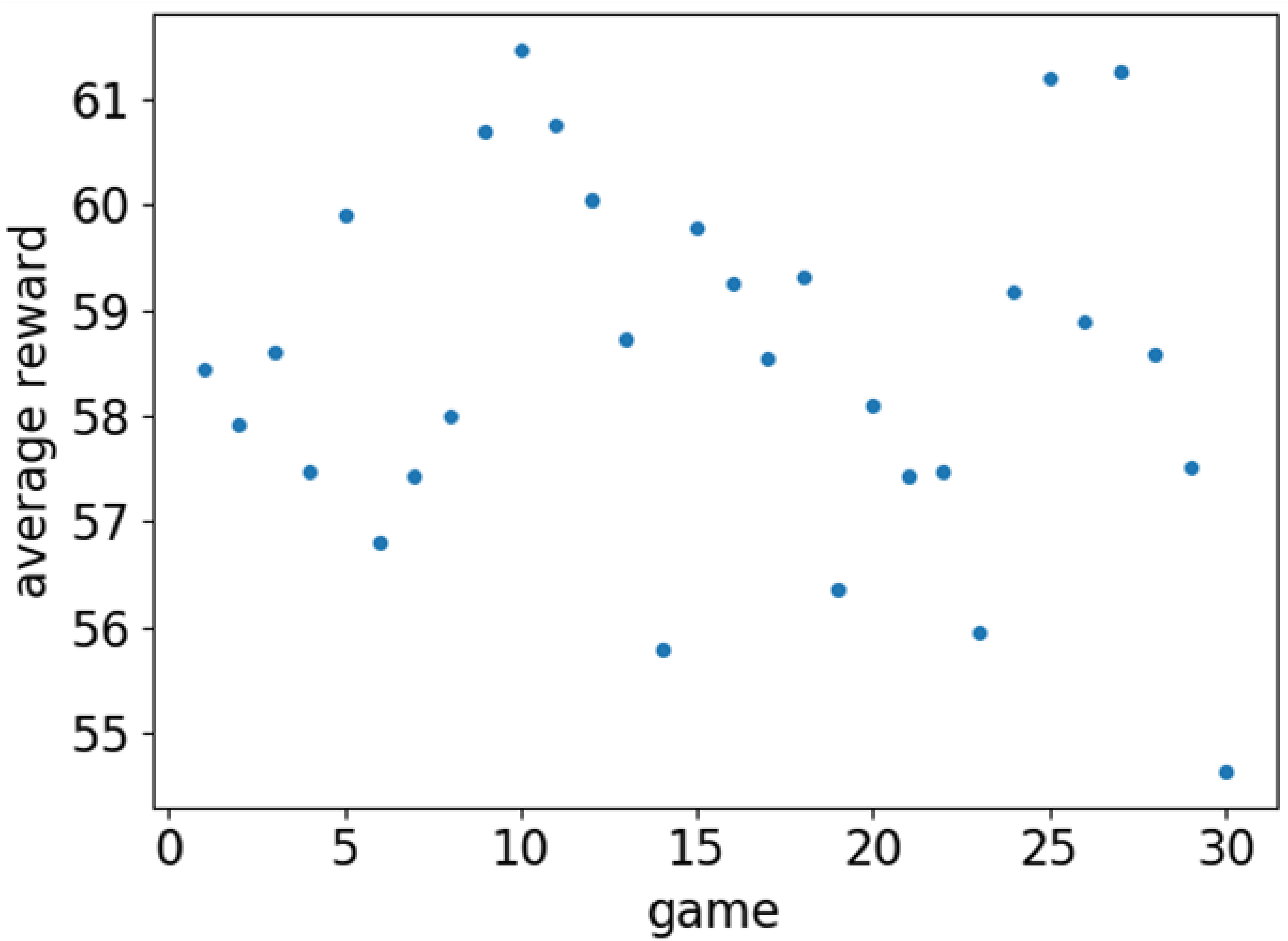
Results:
Non-Intelligent Agents

- No learning
- No correlation

Zero Intelligence Agents (ZIA)

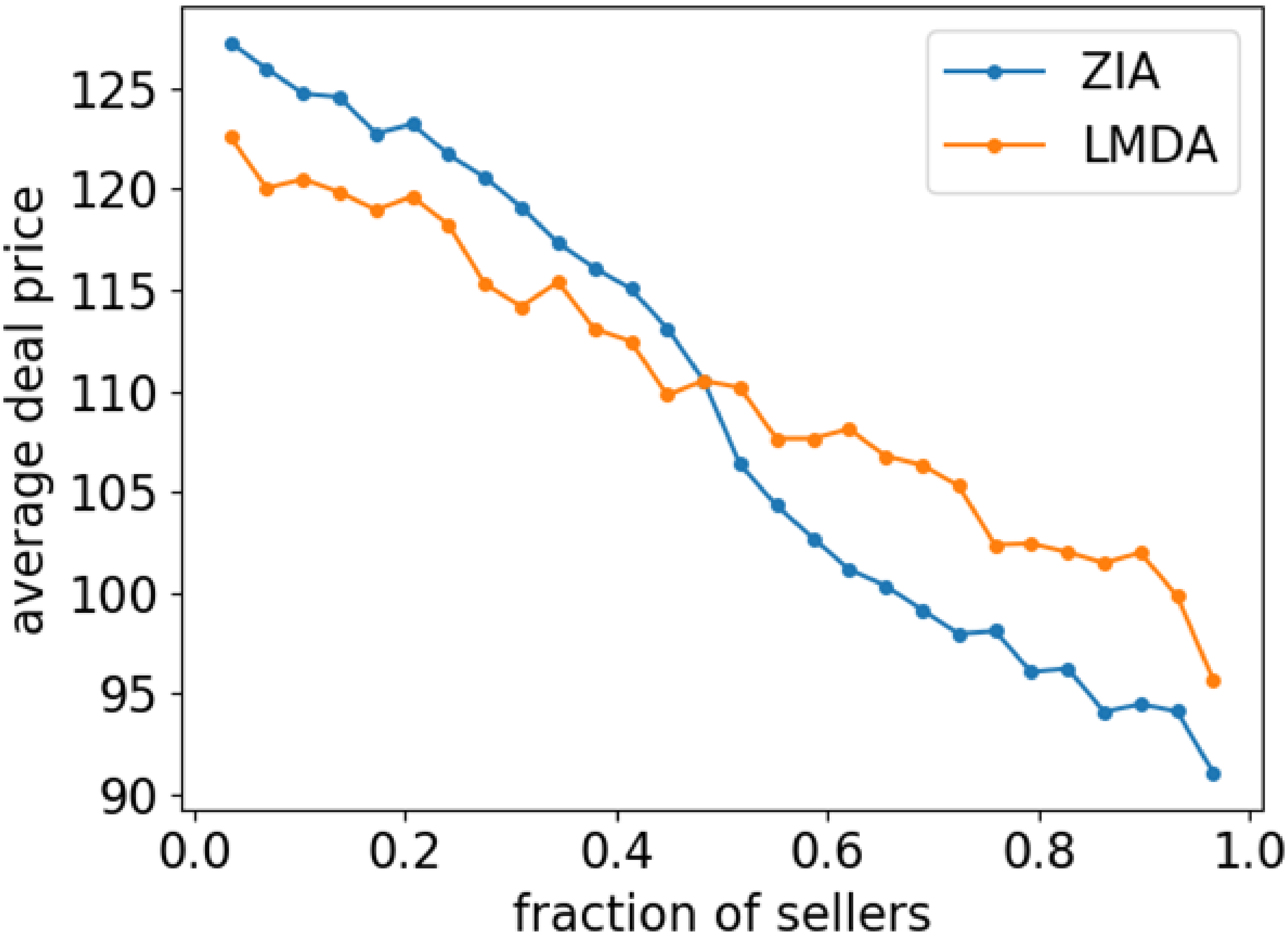


Linear Markov Decision Agents (LMDA)

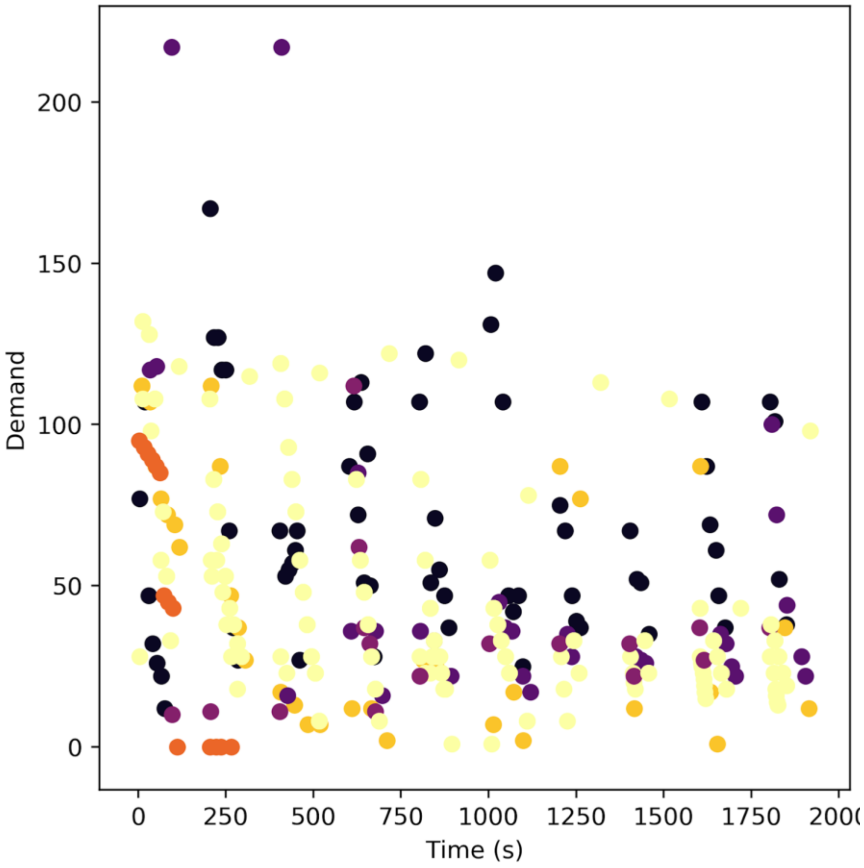


Deal Price vs Fraction of Sellers

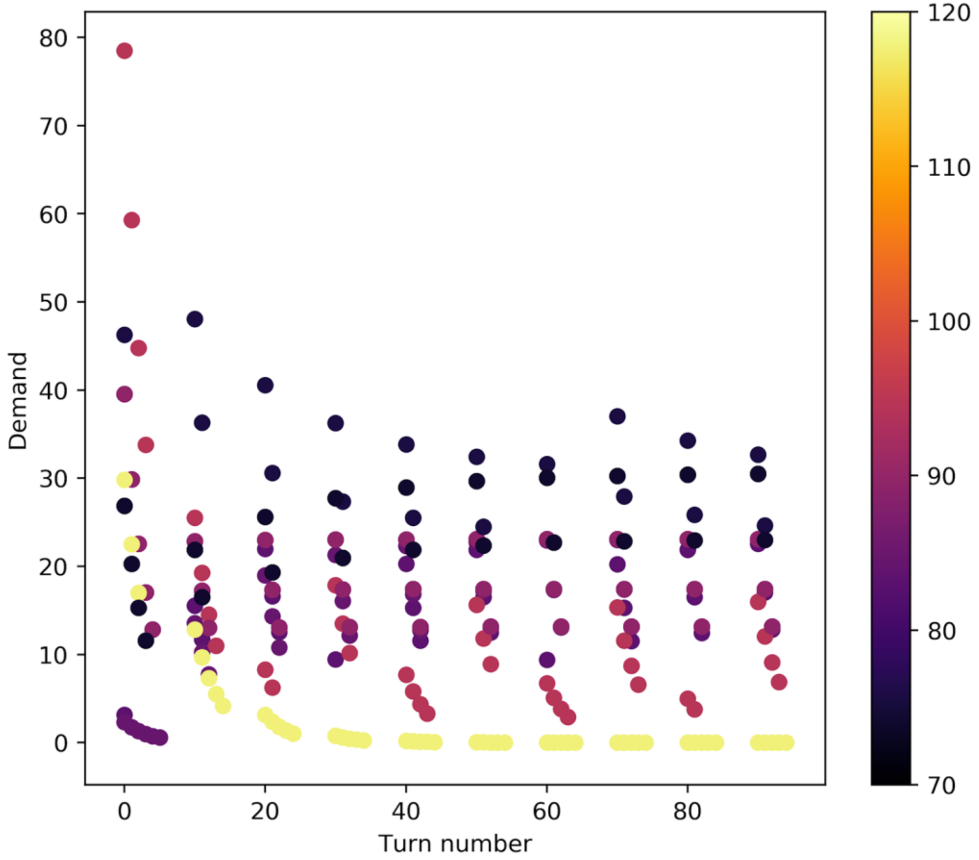
- Seller fraction increases → deal price decreases
- Competition
- Slope → competition
  - ZIA faces tougher competition than LMDA


ZIA
LMDA
average deal price
fraction of sellers
90
95
100
105
110
115
120
125
0.0
0.2
0.4
0.6
0.8
1.0

Demands -- Experiments vs LMDA



Experiments



LMDA

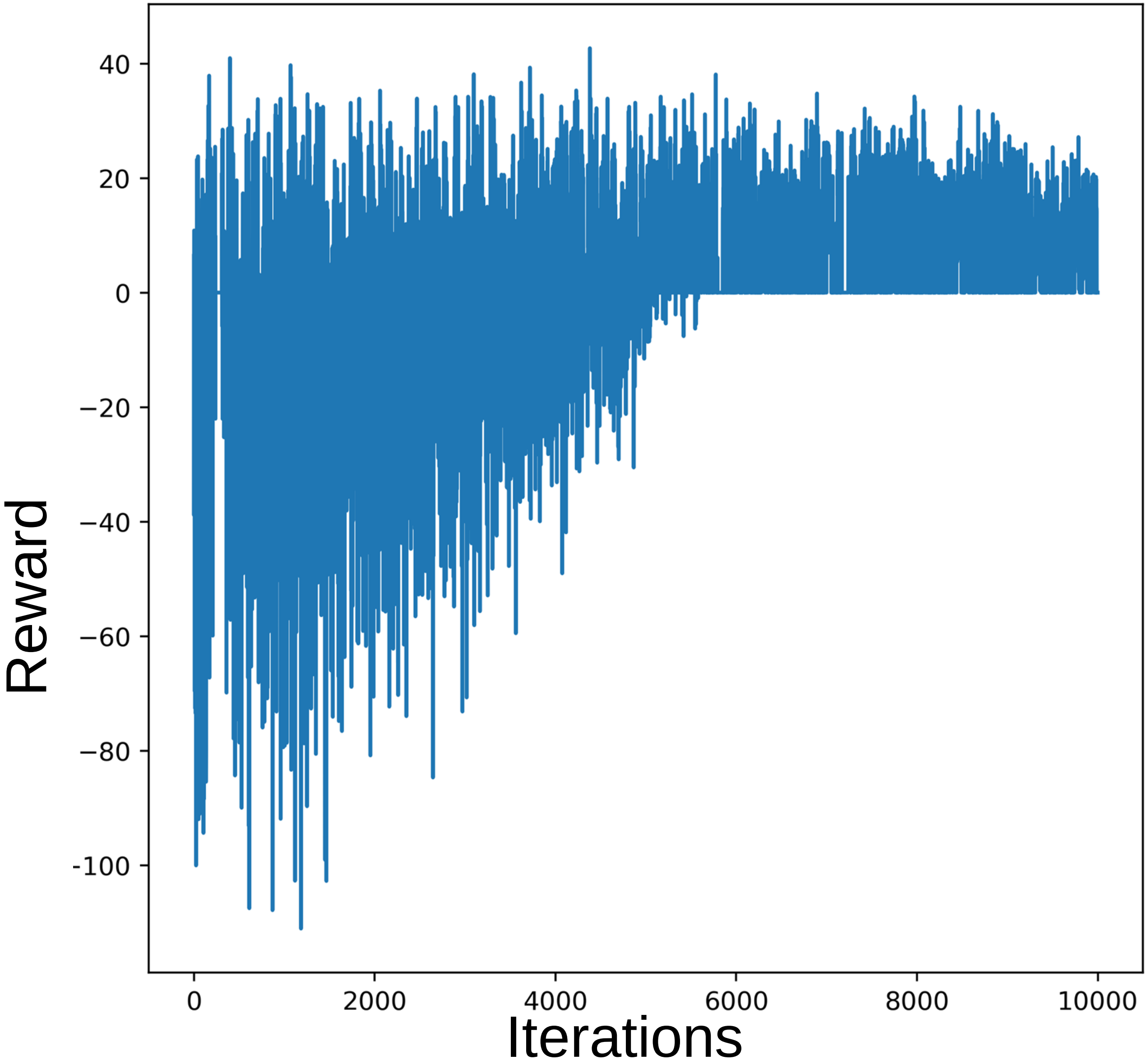
ETH zürich

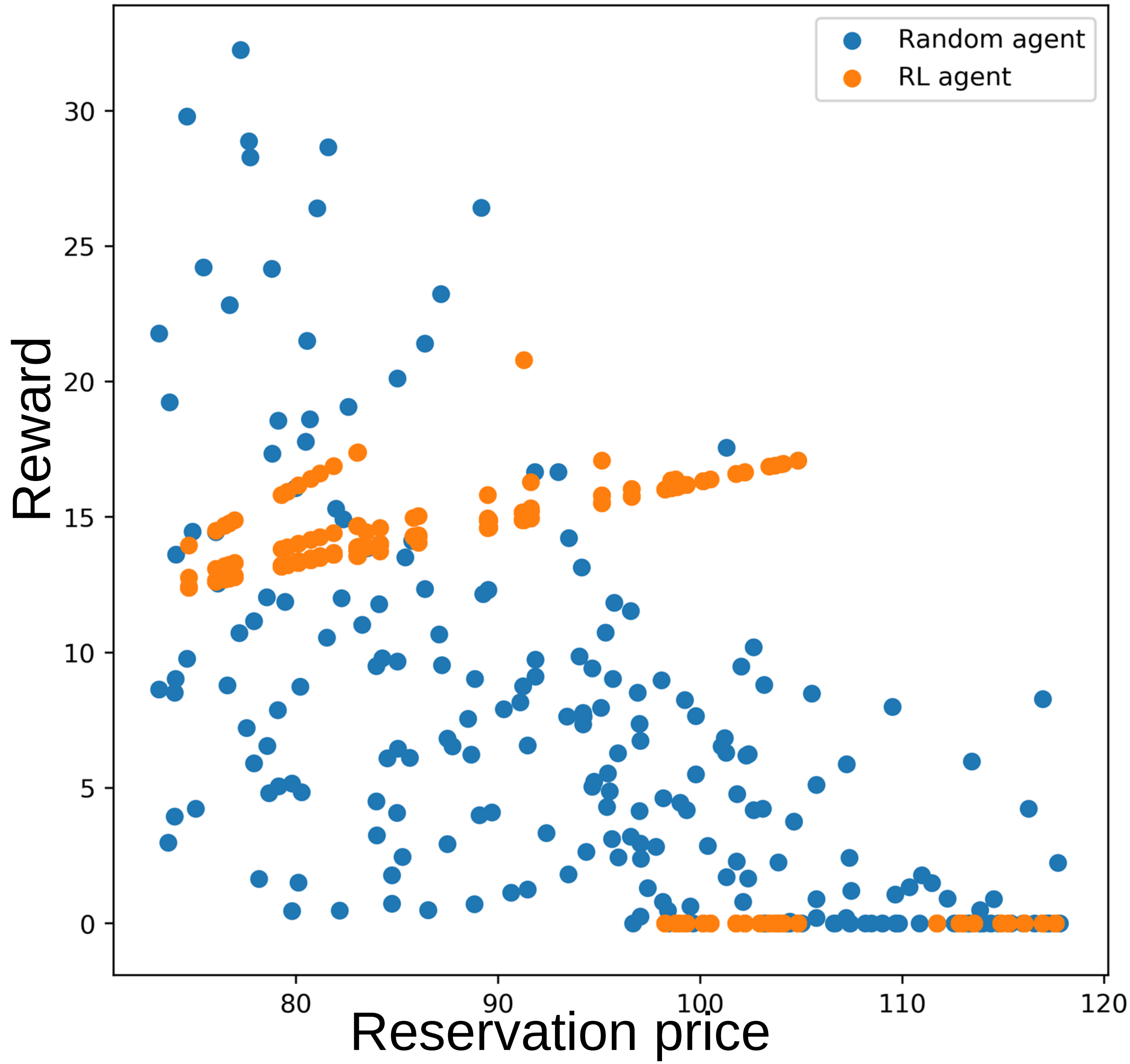
Deep Reinforcement Learning Model

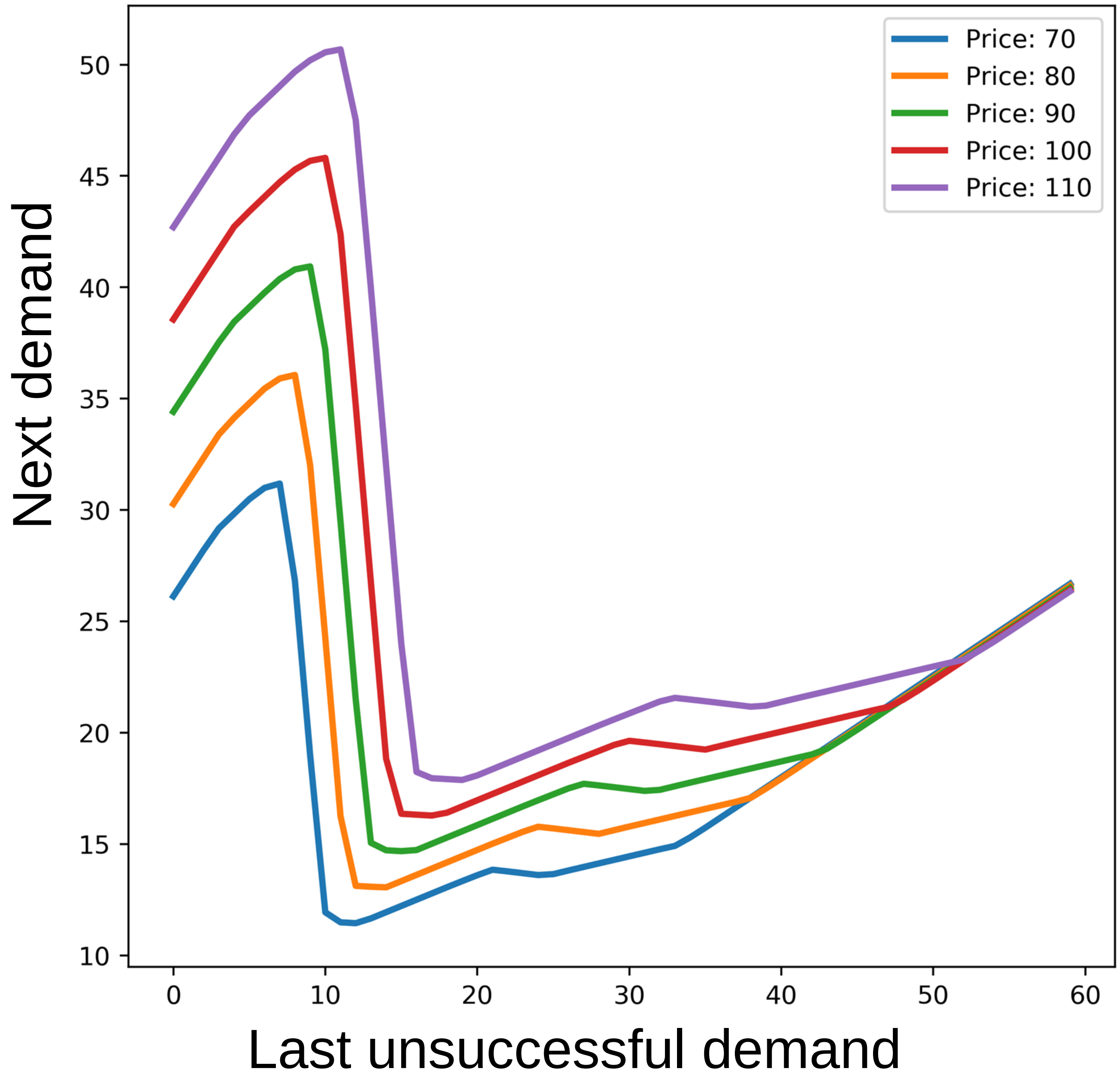
- 2 different exploration policies:
  - Gaussian
  - Ornstein-Uhlenbeck (OU)

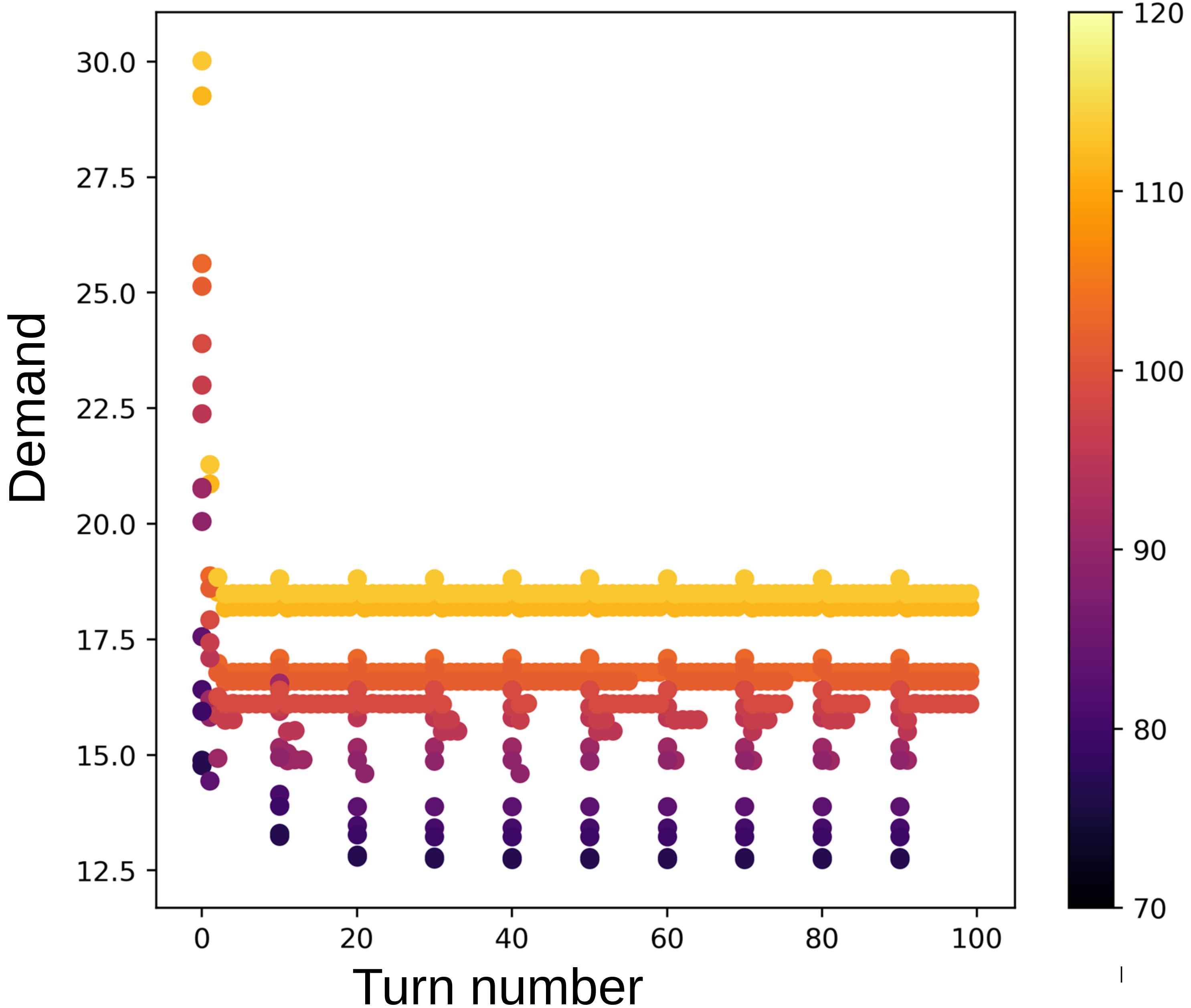
- 1 intelligent agent + a pool of non-intelligent agents (e.g. ZIA, LMDA)

Gaussian exploration policy

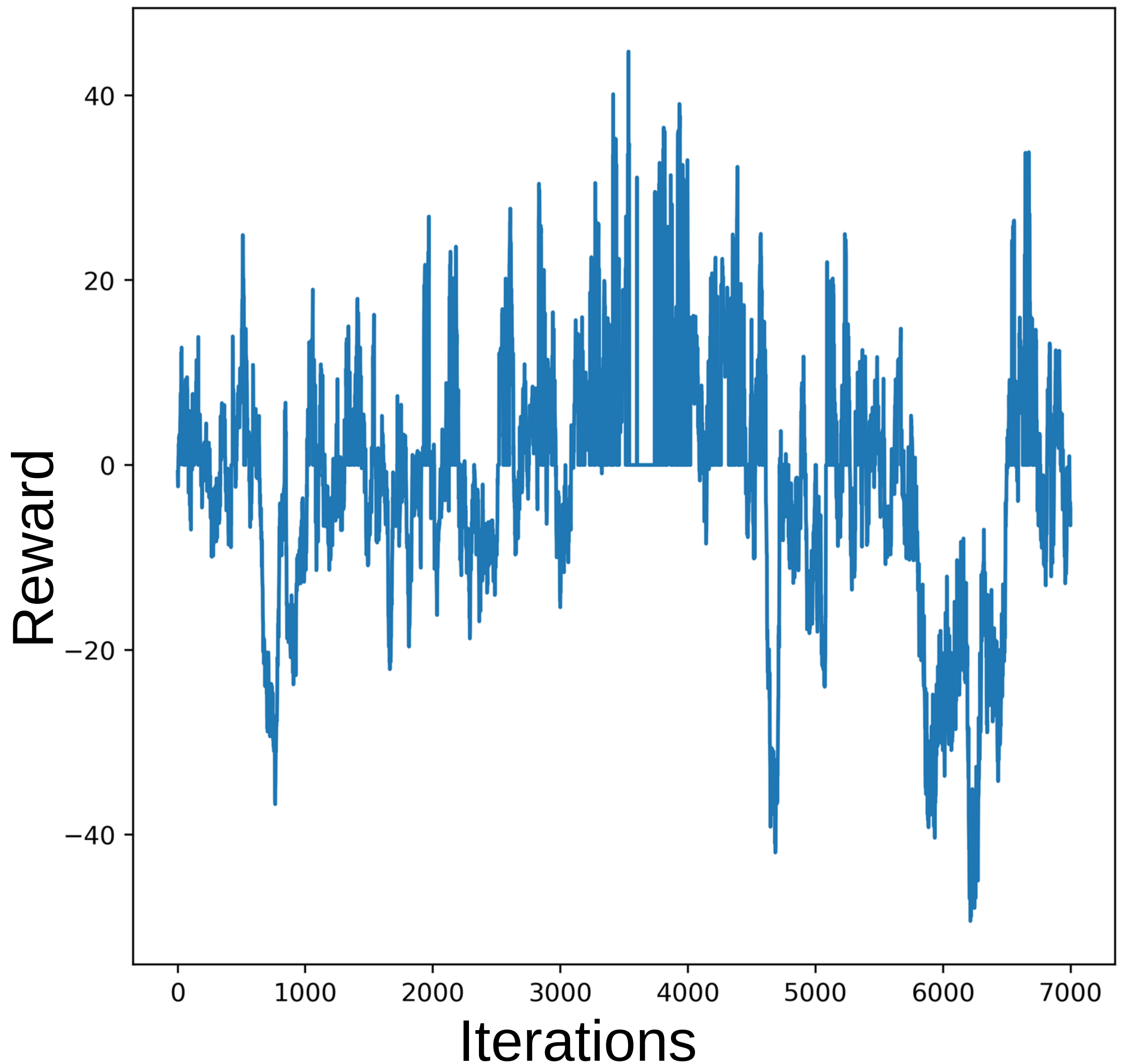


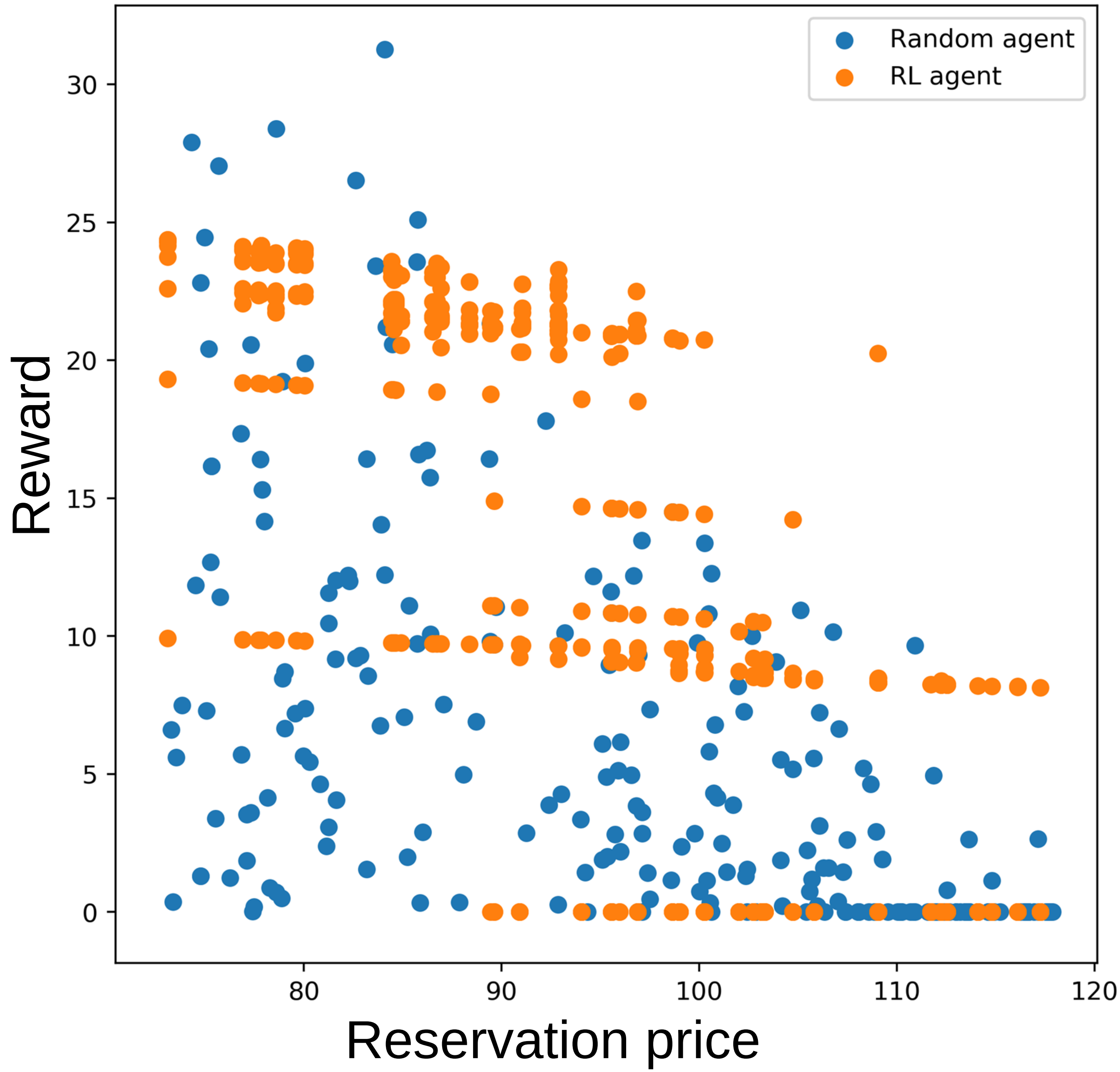

Random agent
RL agent
Reward
Reservation price
0
5
10
15
20
25
30
80
90
100
110
120

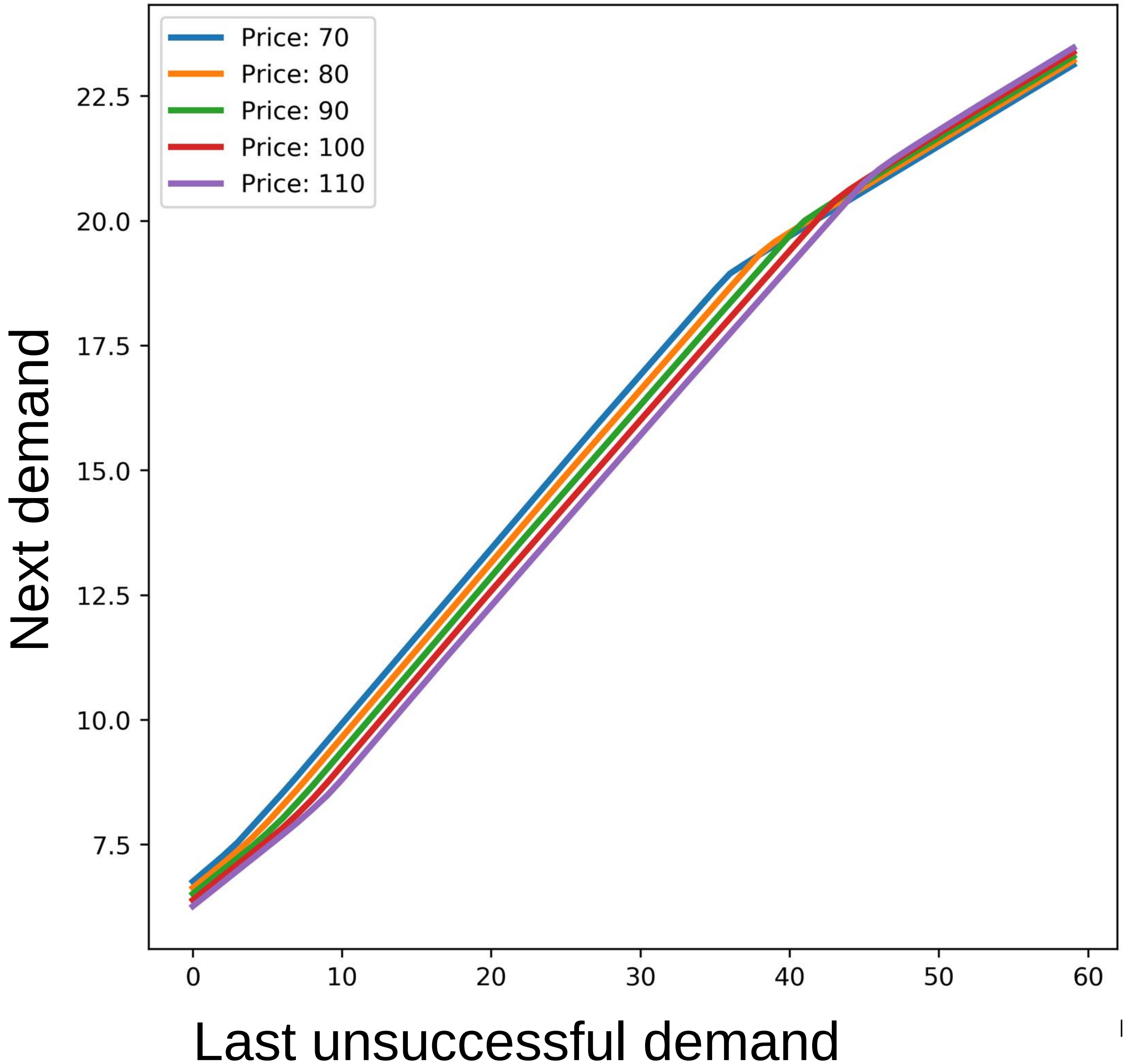

Price: 70
Price: 80
Price: 90
Price: 100
Price: 110
Next demand
Last unsuccessful demand
10
15
20
25
30
35
40
45
50
0
10
20
30
40
50
60

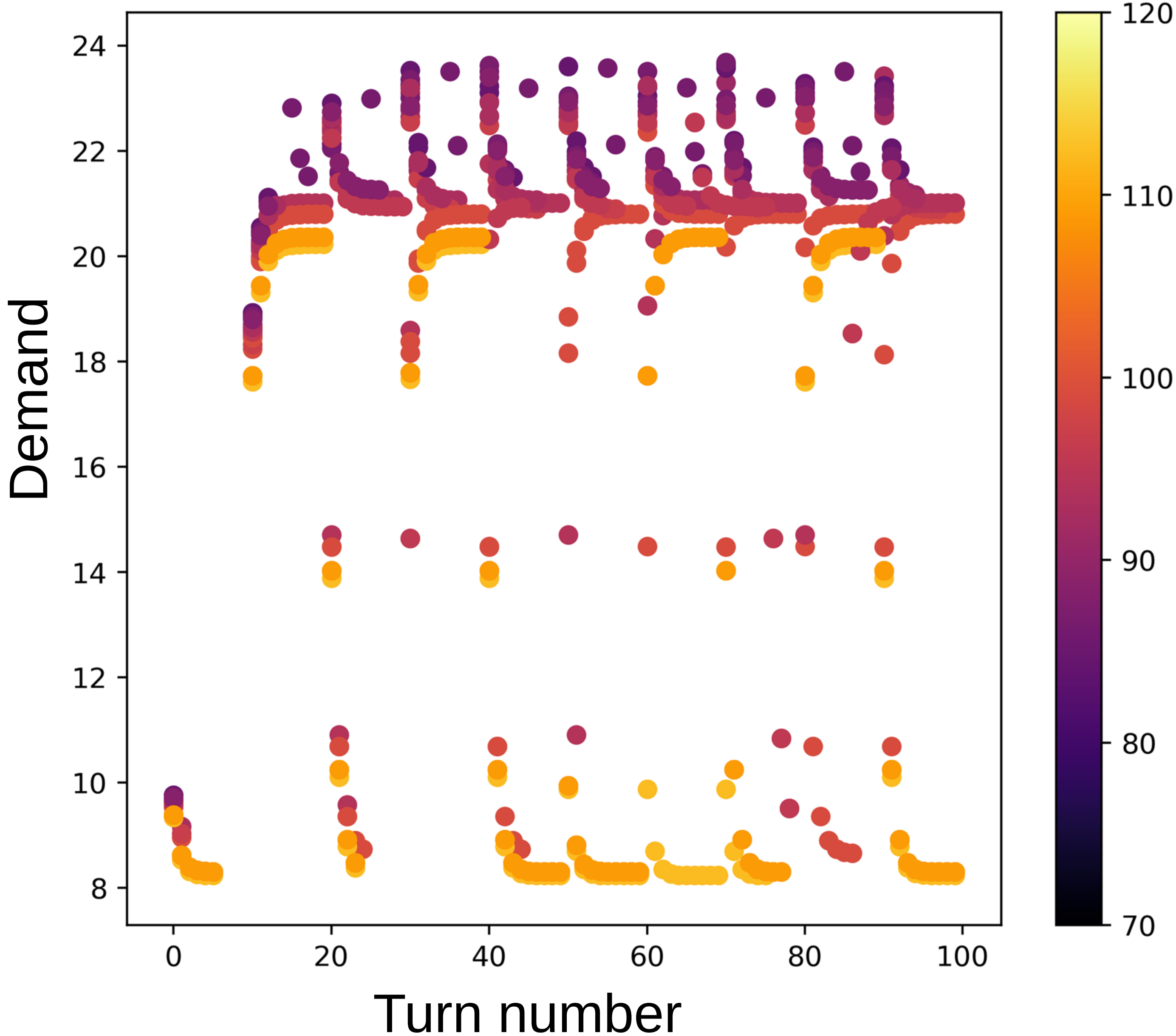

Demand
30.0
27.5
25.0
22.5
20.0
17.5
15.0
12.5
0
20
40
60
80
100
Turn number
120
110
100
90
80
70

Ornstein-Uhlenbeck (OU) exploration policy




Random agent
RL agent
30
25
20
15
10
5
0
80
90
100
110
120
Reward
Reservation price


Price: 70
Price: 80
Price: 90
Price: 100
Price: 110
22.5
20.0
17.5
15.0
12.5
10.0
7.5
0
10
20
30
40
50
60
Next demand
Last unsuccessful demand


Demand
24
22
20
18
16
14
12
10
8
0
20
40
60
80
100
Turn number
120
110
100
90
80
70

- reward is negative at early stages
- agent learns quickly to avoid negative reward
- higher reservation price
  - more difficult to sell
  - market performs worse
- Gaussian agent performs better than OU agent

Agent Pool	Pool Earnings	RLA Earnings	Learning Agent
ZIA	5.23	9.27	RLA+Gaussian
ZIA	7.01	8.39	RLA+OU
ZIA	5.27	12.32	RLA+OU+anneal
LMDA	7.19	-	RLA+Gaussian
LMDA	-	-	RLA+OU
PAA	-	-	RLA+OU