



Deep Reinforcement Learning for Double Auction Processes

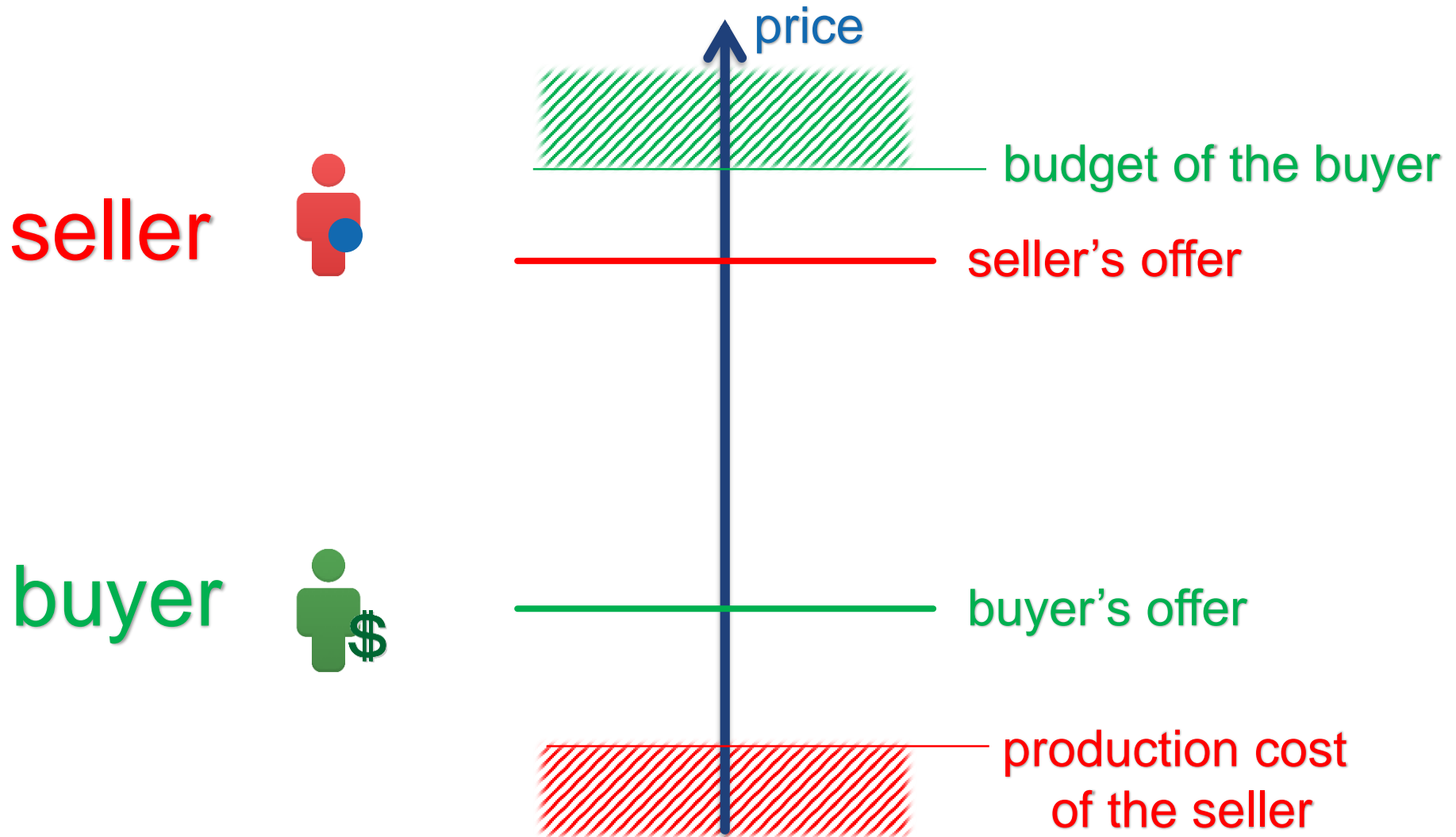
Batuhan Yardim

Aleksei Khudorozhkov

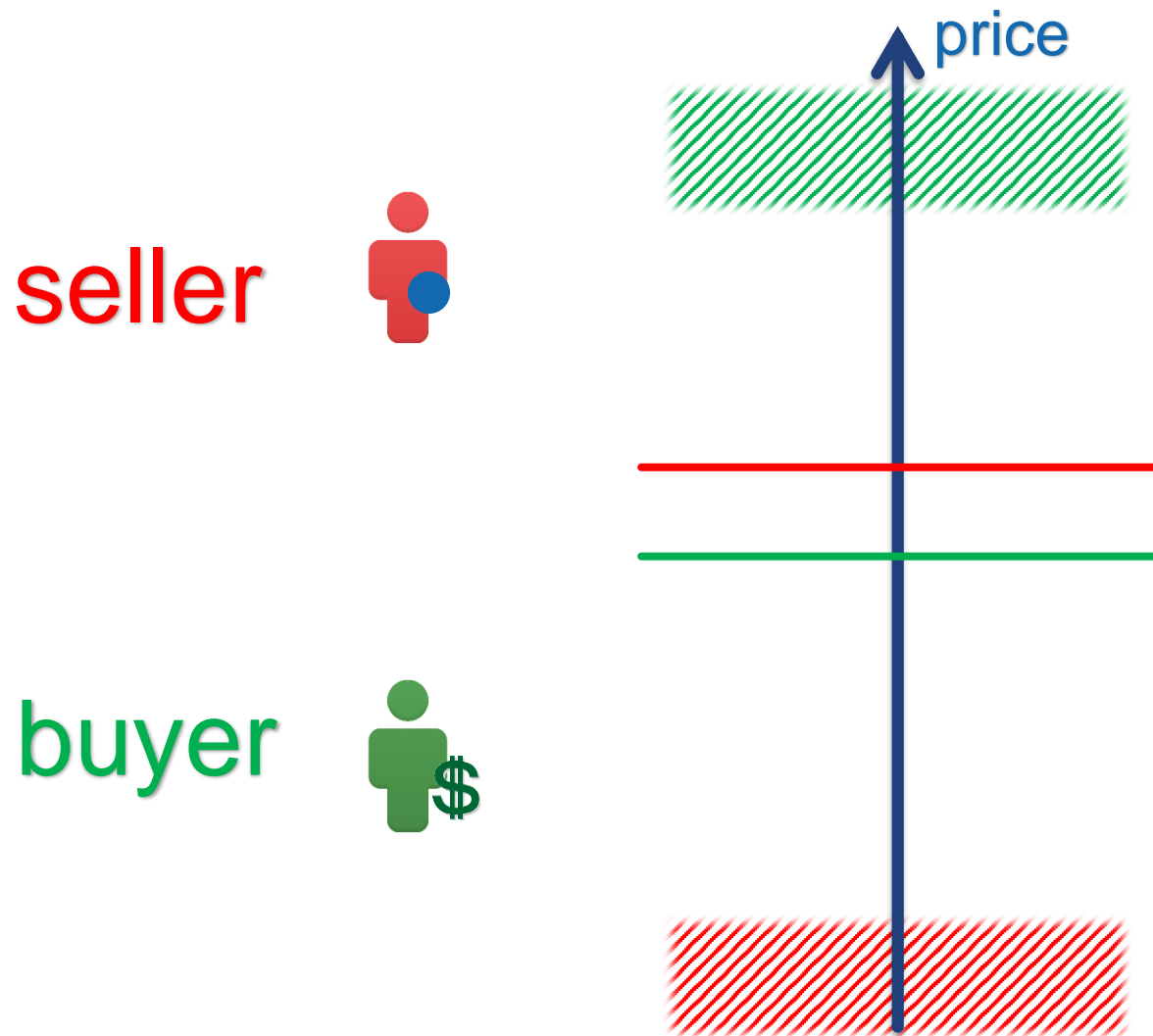
Ka Rin Sim

Neri Passaleva

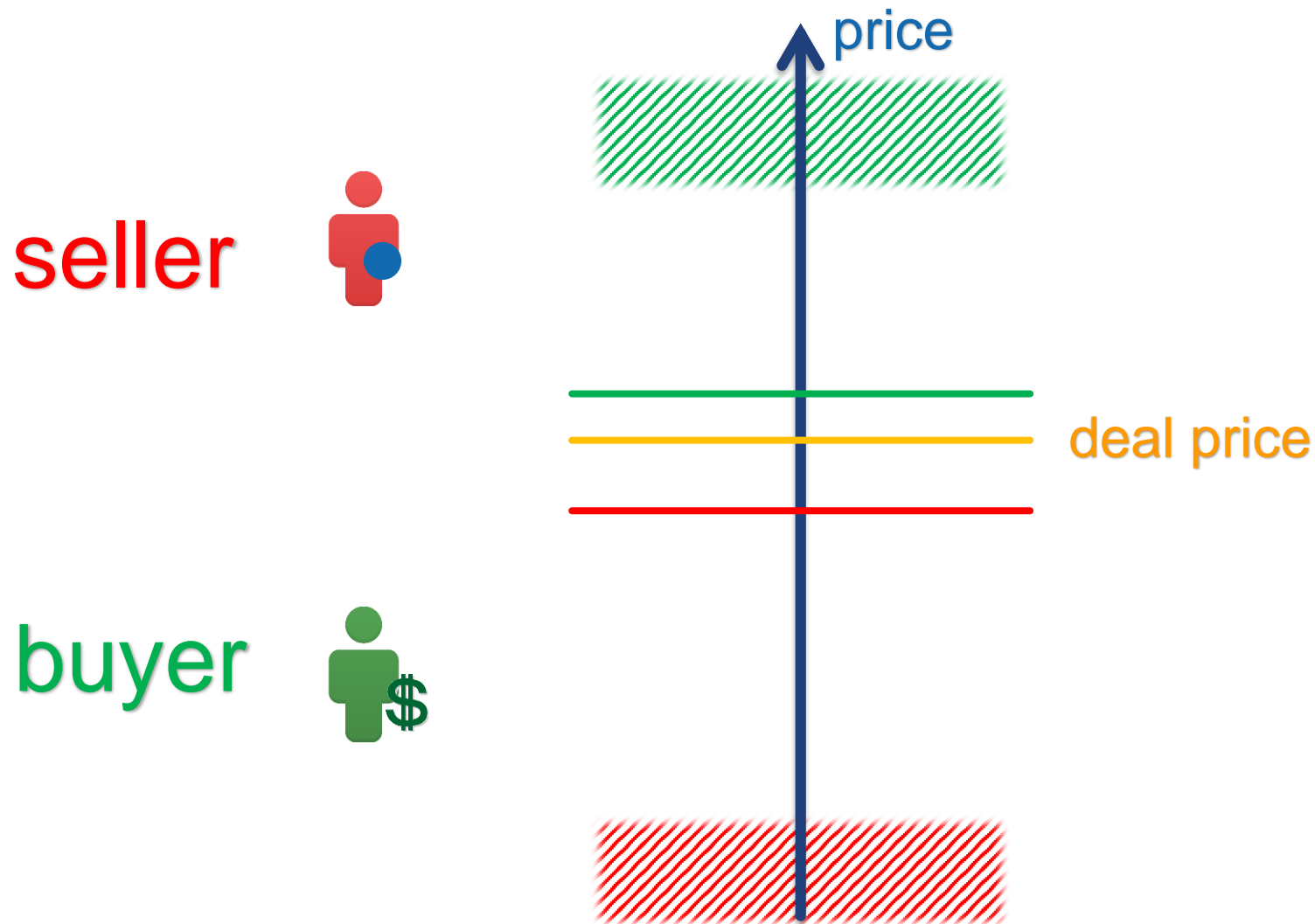
Double auction



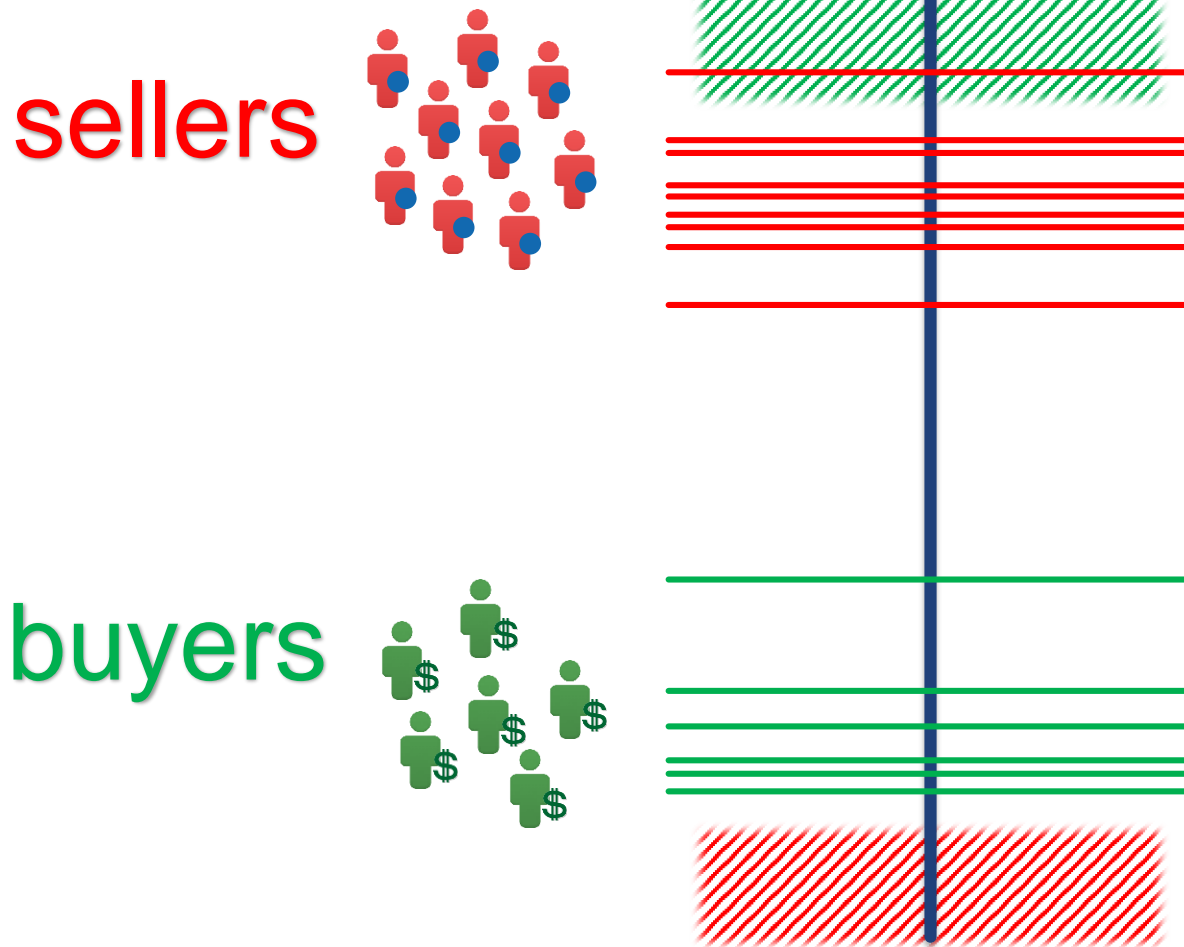
Double auction



Double auction



Double auction



Each agent wants to maximize the reward

For this they can choose different strategies

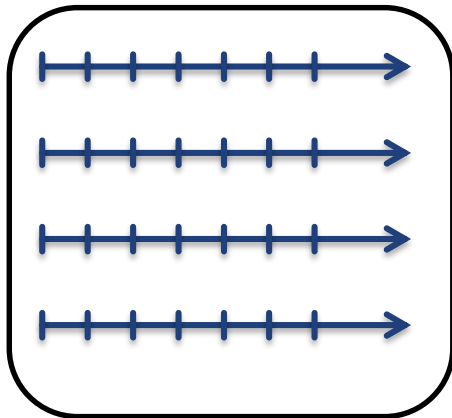
Market environment

- 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

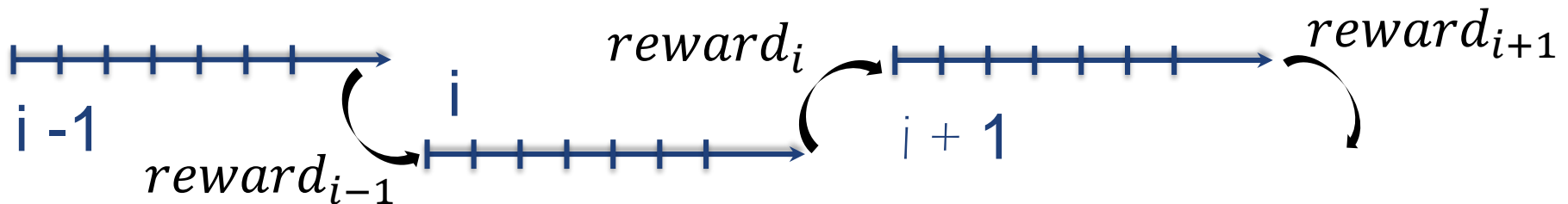
- After each time step an agent receives observations from the market environment.
- Core observations are:
 - The last offer of the agent
 - Current time step
 - bool: if the agent managed to make a deal in the previous round
- Other observations might be included:
 - Last offers of agents of the same/opposite 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

Reward Mechanism

- The agent's **reward** is the difference between the deal-price and the reservation-price in absolute value

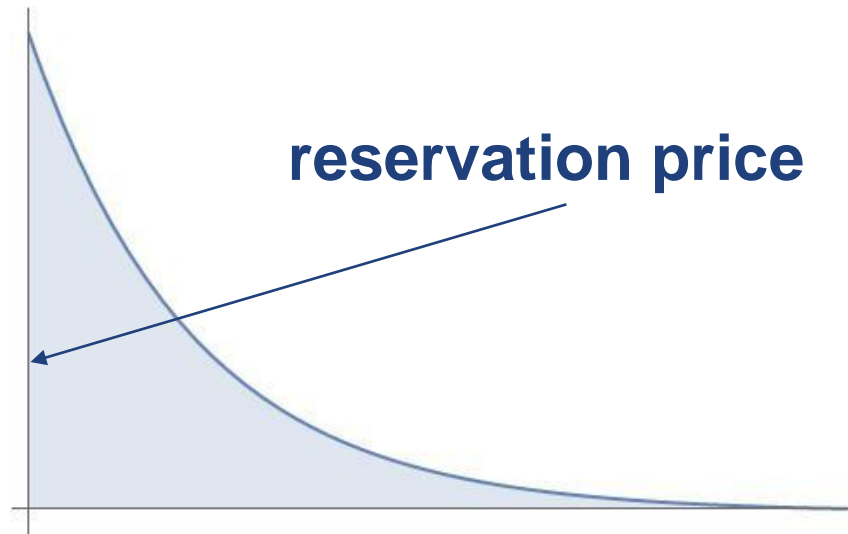
$$reward_i = |res_i - deal_i| \quad \text{reward for round } i$$

- Calculated after each round



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

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

Demand at current time

Demand at previous time step

Boolean indicator of previous round outcome

Noise

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

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

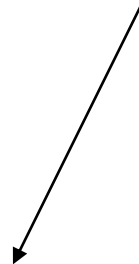
The agent becomes aggressive after an unsuccessful round

Deep RL Agents

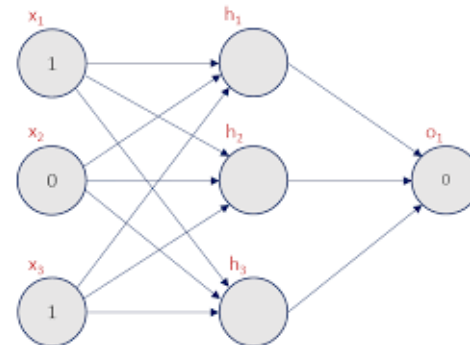
- Learning through Deep Deterministic Policy Gradient (DDPG) framework

GAUSSIAN EXPLORATION MODEL

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



Parametrized Agent's Policy

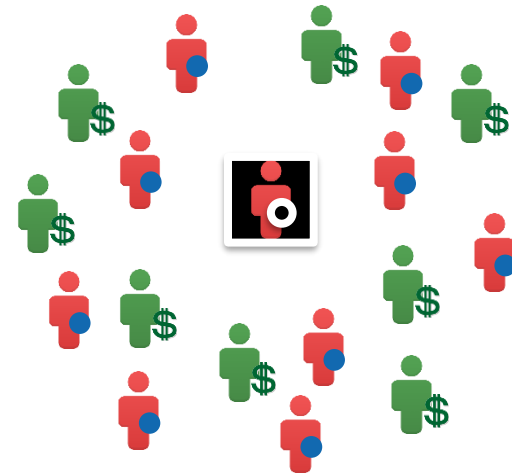


Fully-connected 3-layer
neural network

Deep RL Agents

ENVIRONMENT

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

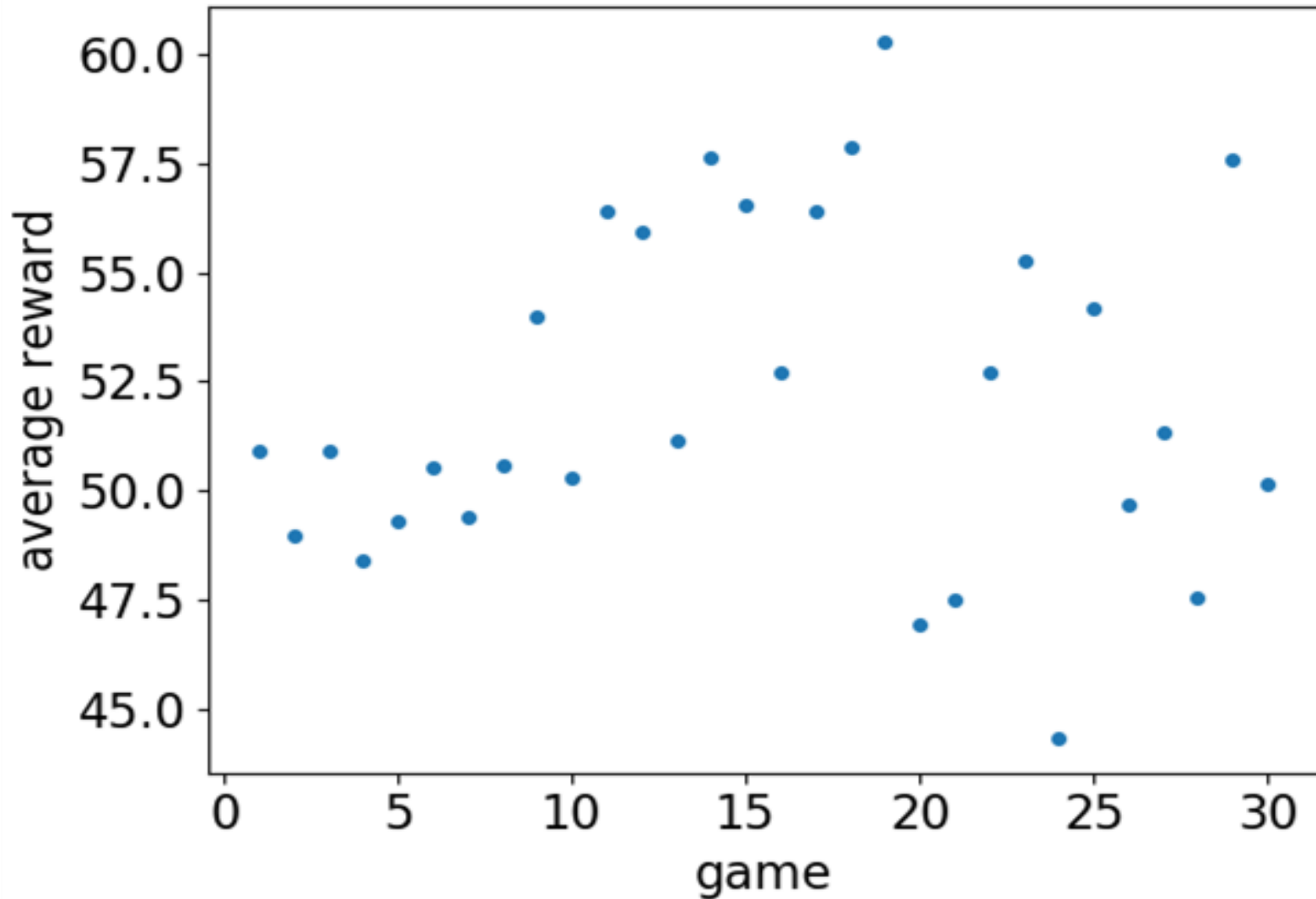


Results:

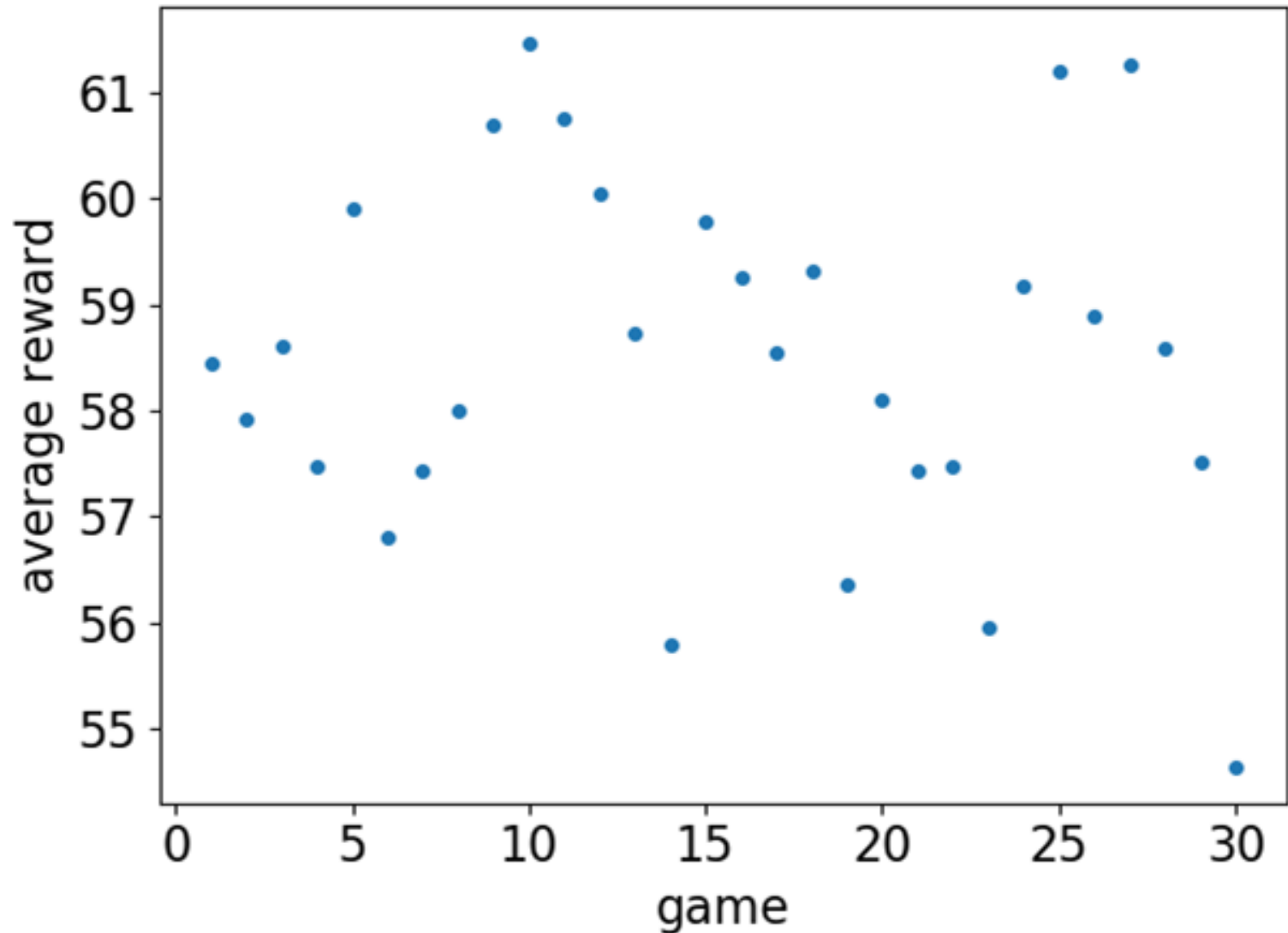
Non-Intelligent Agents

- ★ No learning
- ★ No correlation

Zero Intelligence Agents (ZIA)

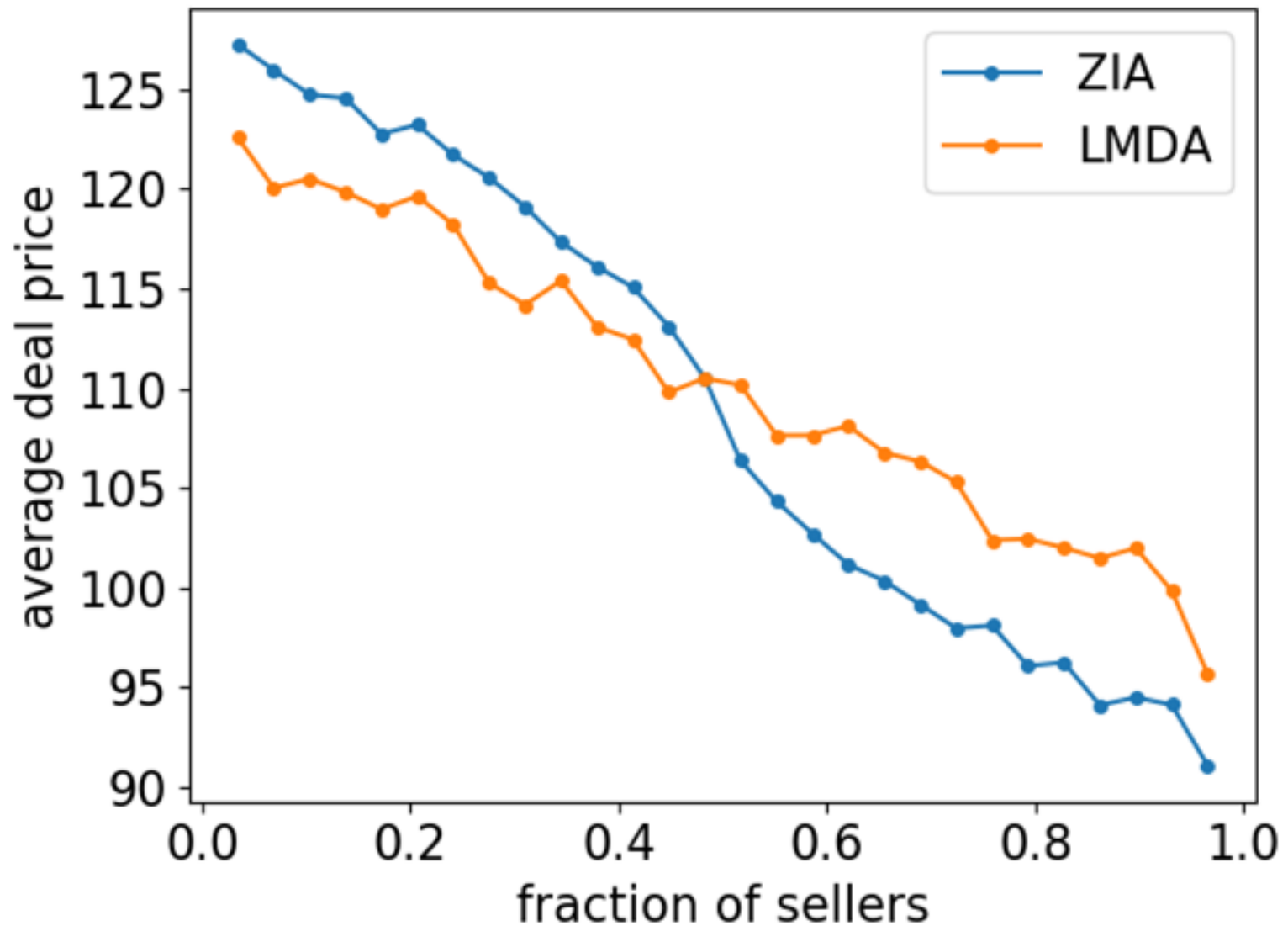


Linear Markov Decision Agents (LMDA)

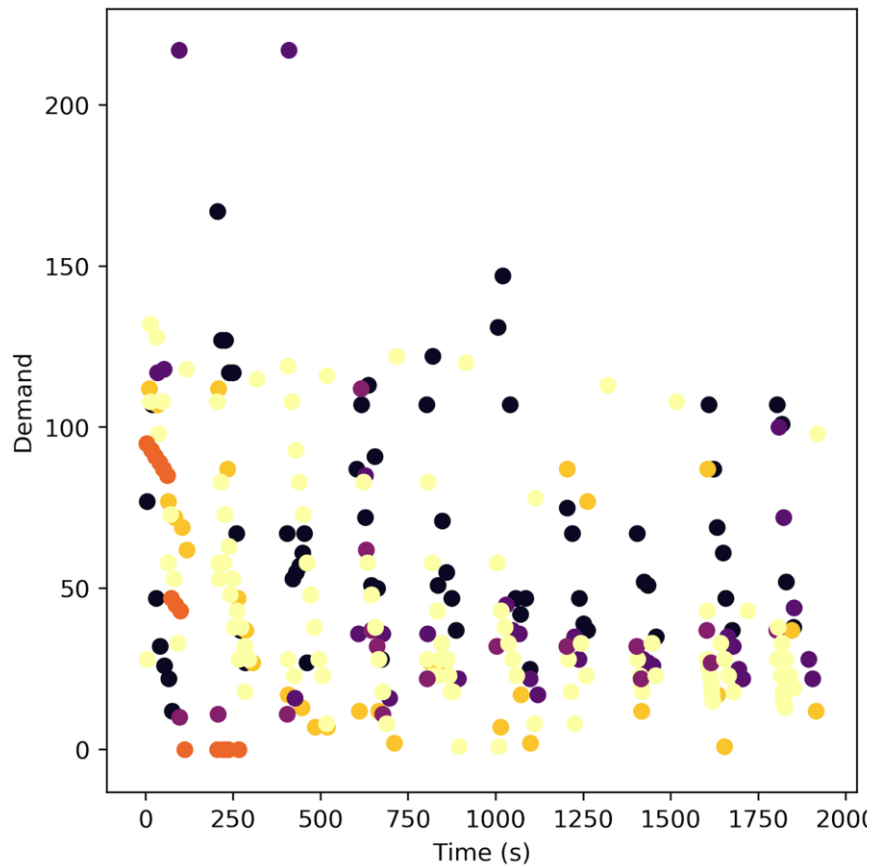


Deal Price vs Fraction of Sellers

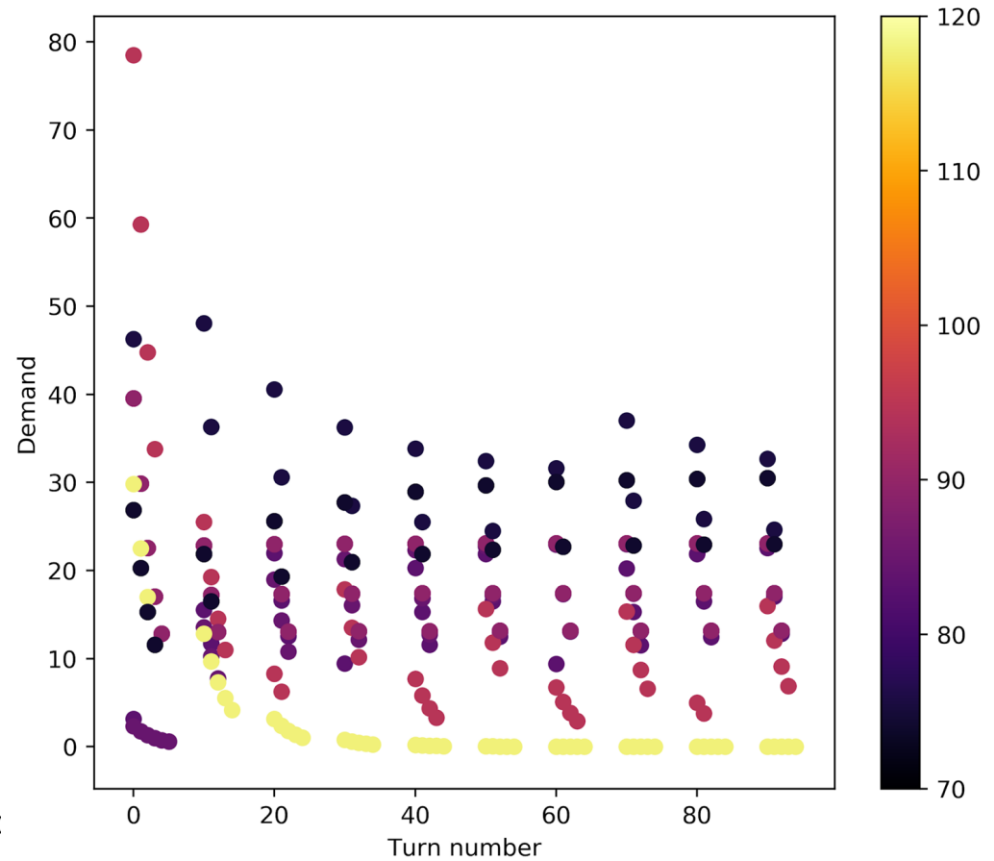
- ★ Seller fraction increases → deal price decreases
- ★ Competition
- ★ Slope → competition
 - ★ ZIA faces tougher competition than LMDA



Demands -- Experiments vs LMDA



Experiments

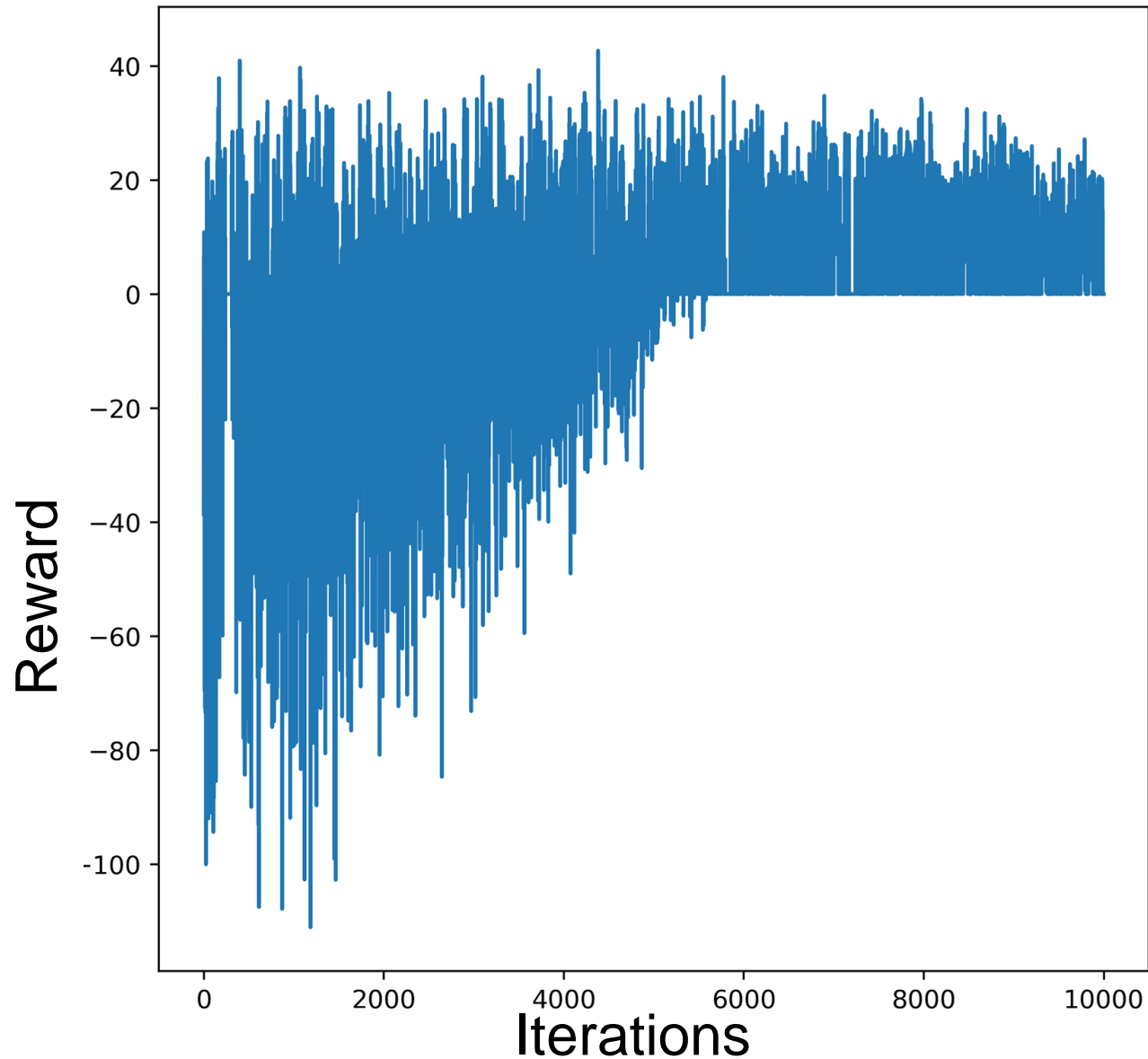


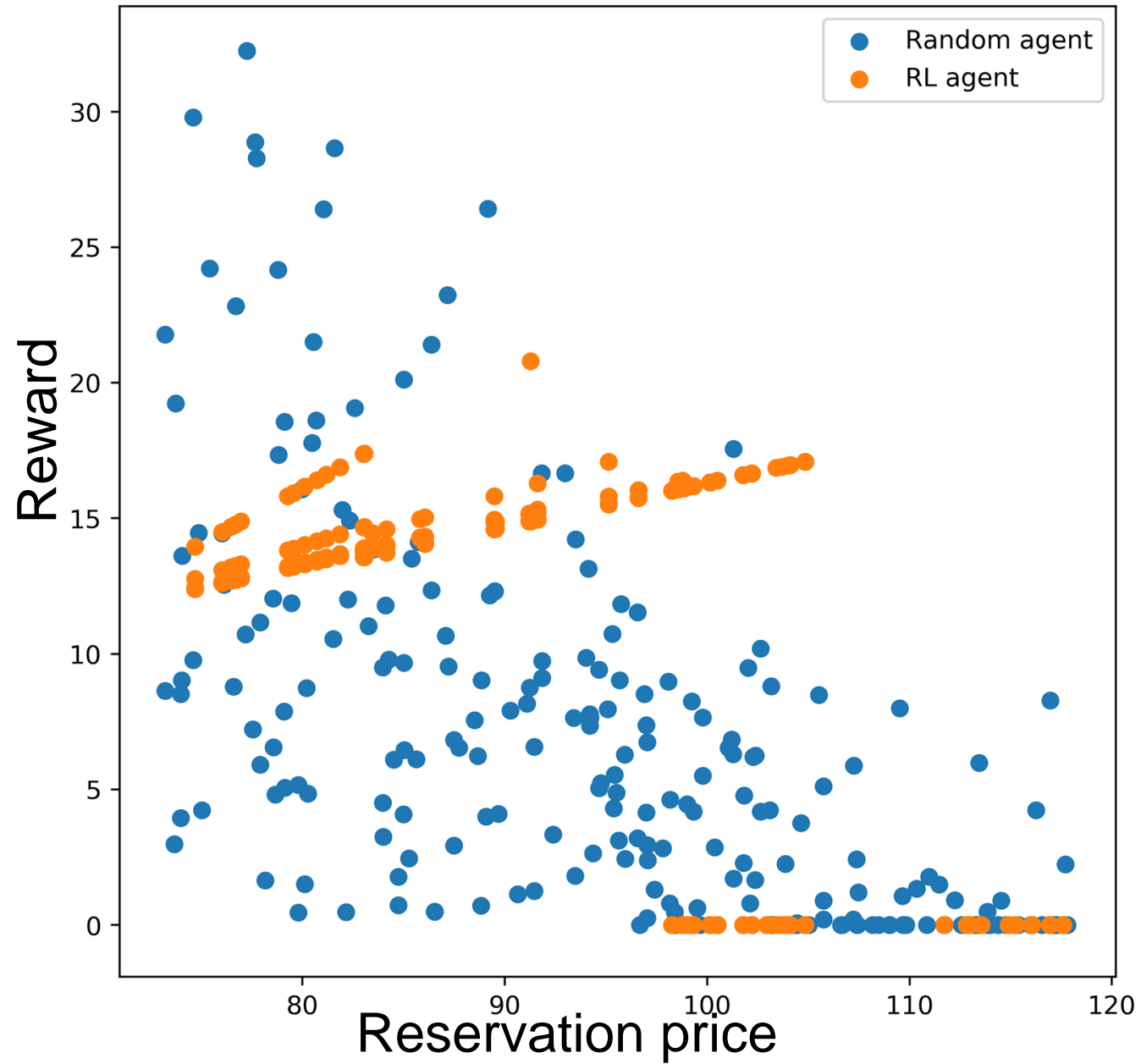
LMDA

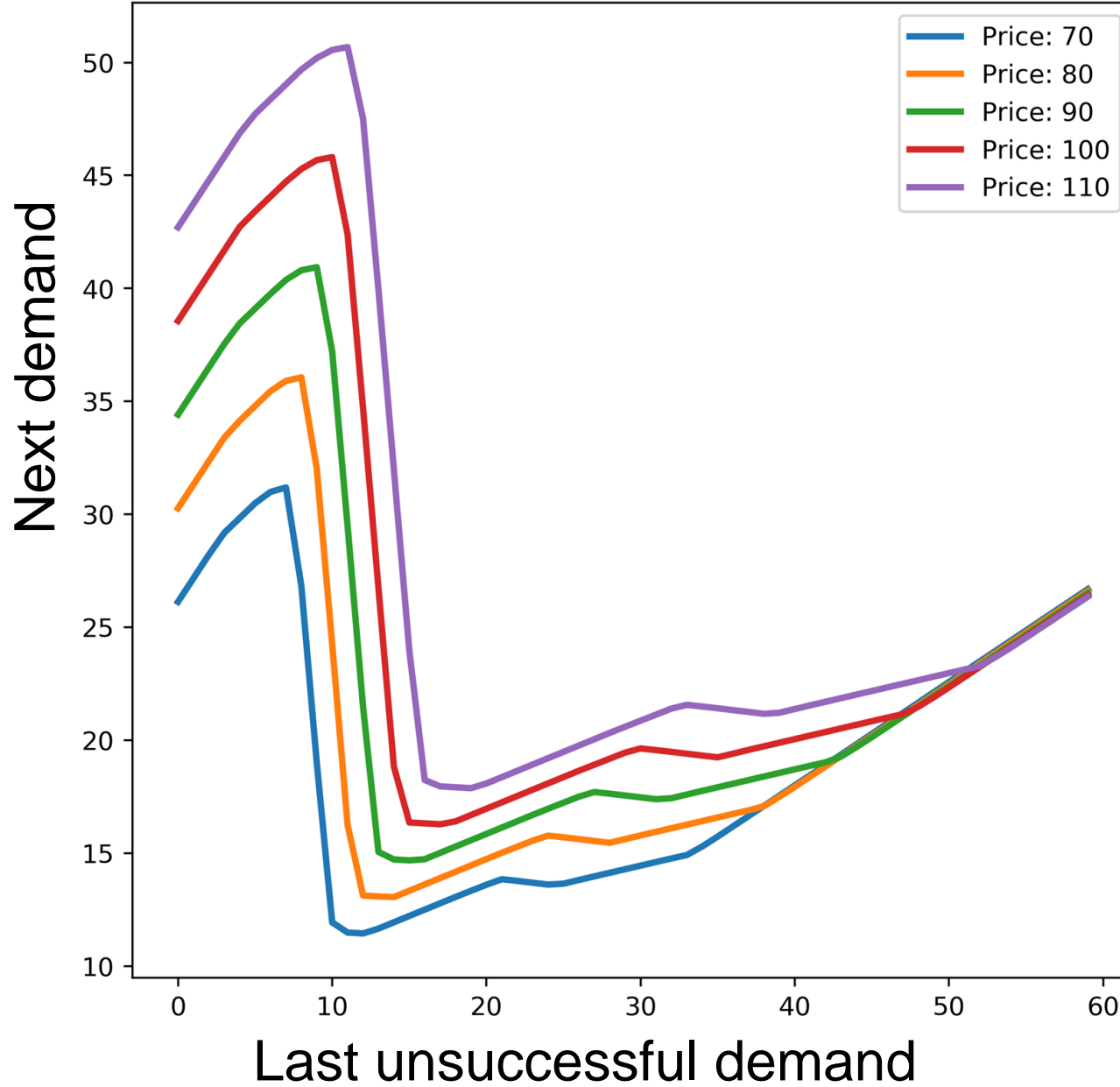
Deep Reinforcement Learning Model

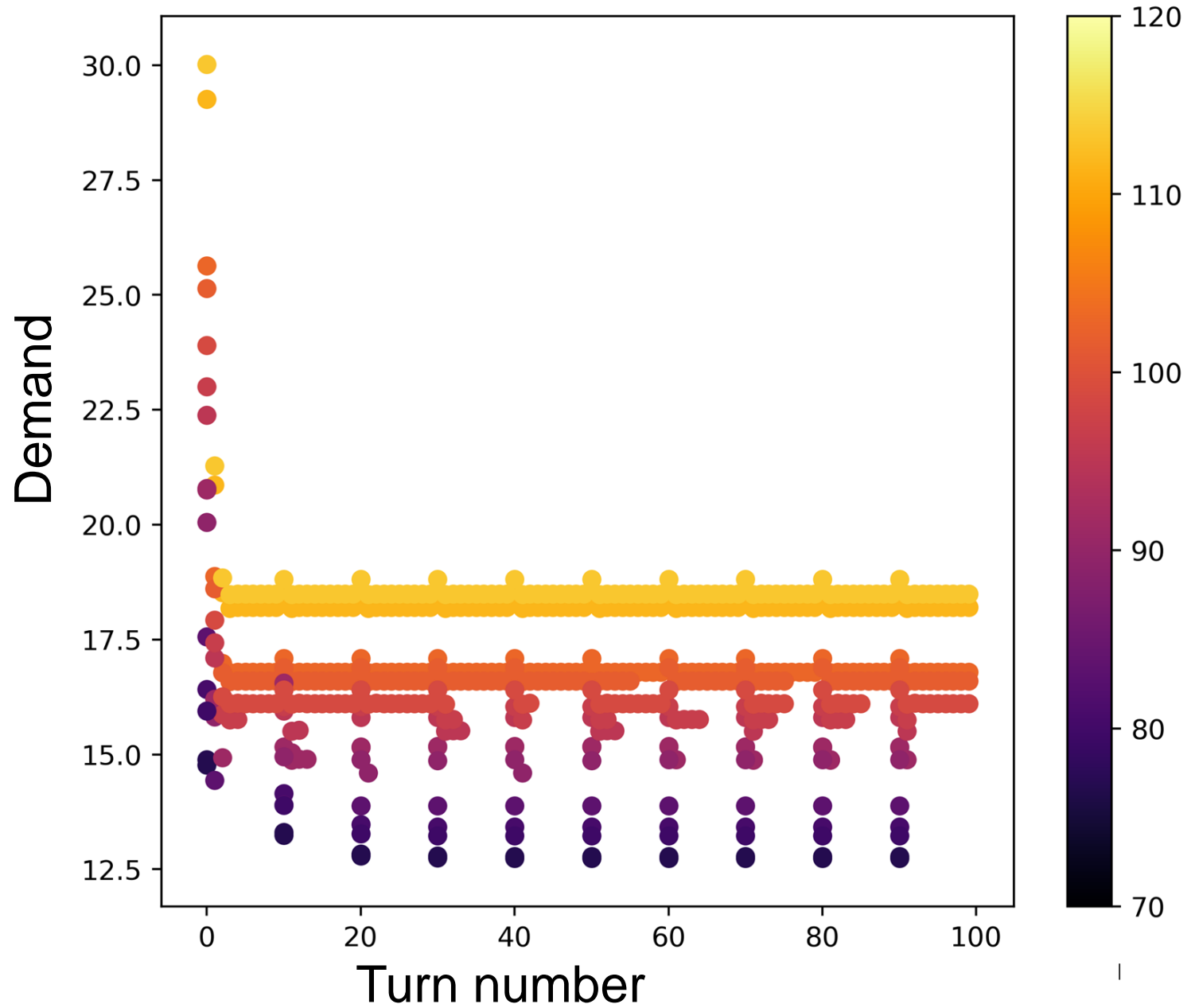
- ★ 2 different exploration policies:
 - ★ Gaussian
 - ★ Ornstein-Uhlenbeck (OU)
- ★ 1 intelligent agent + a pool of non-intelligent agents (e.g. ZIA, LMDA)

Gaussian exploration policy

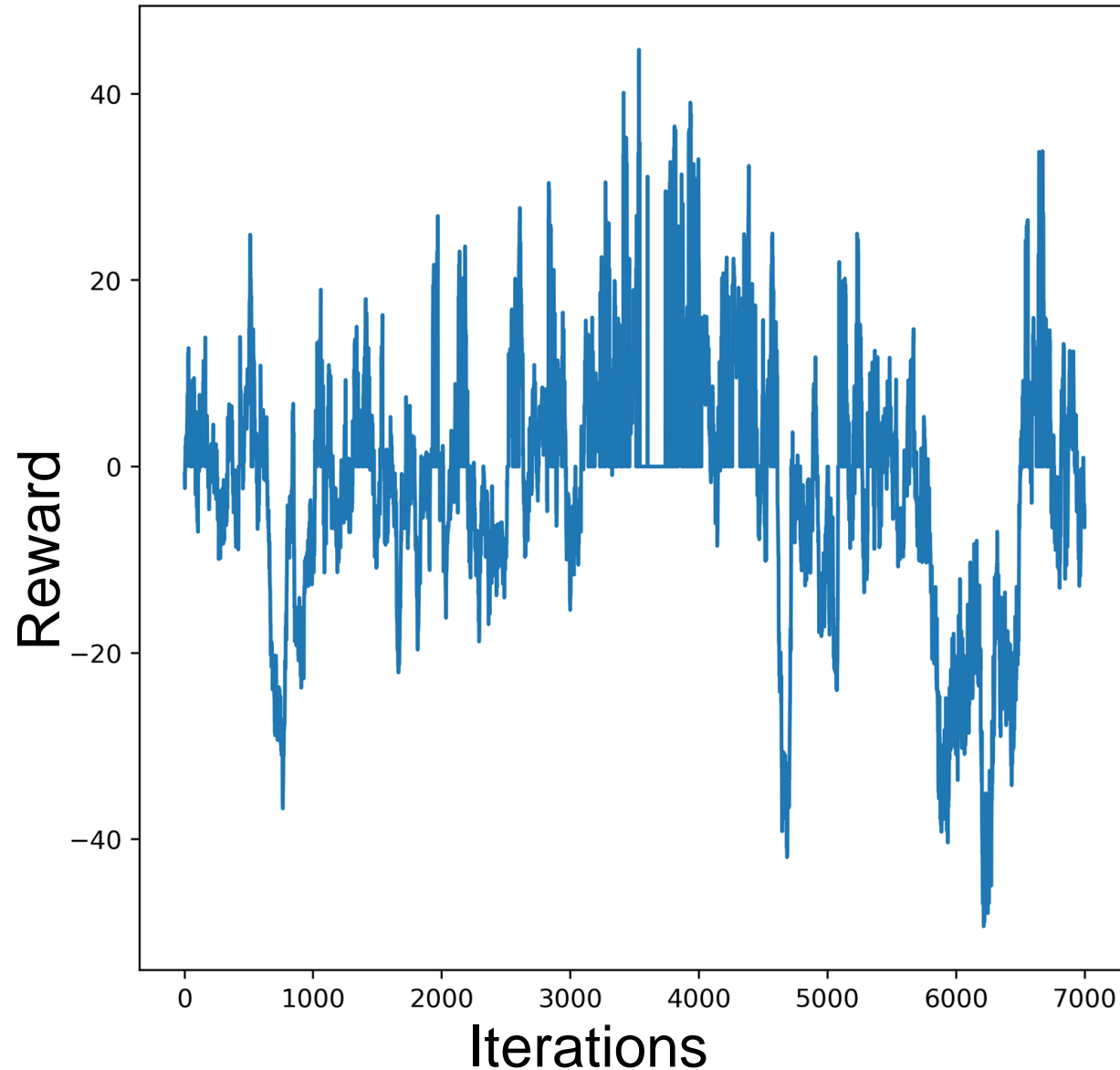


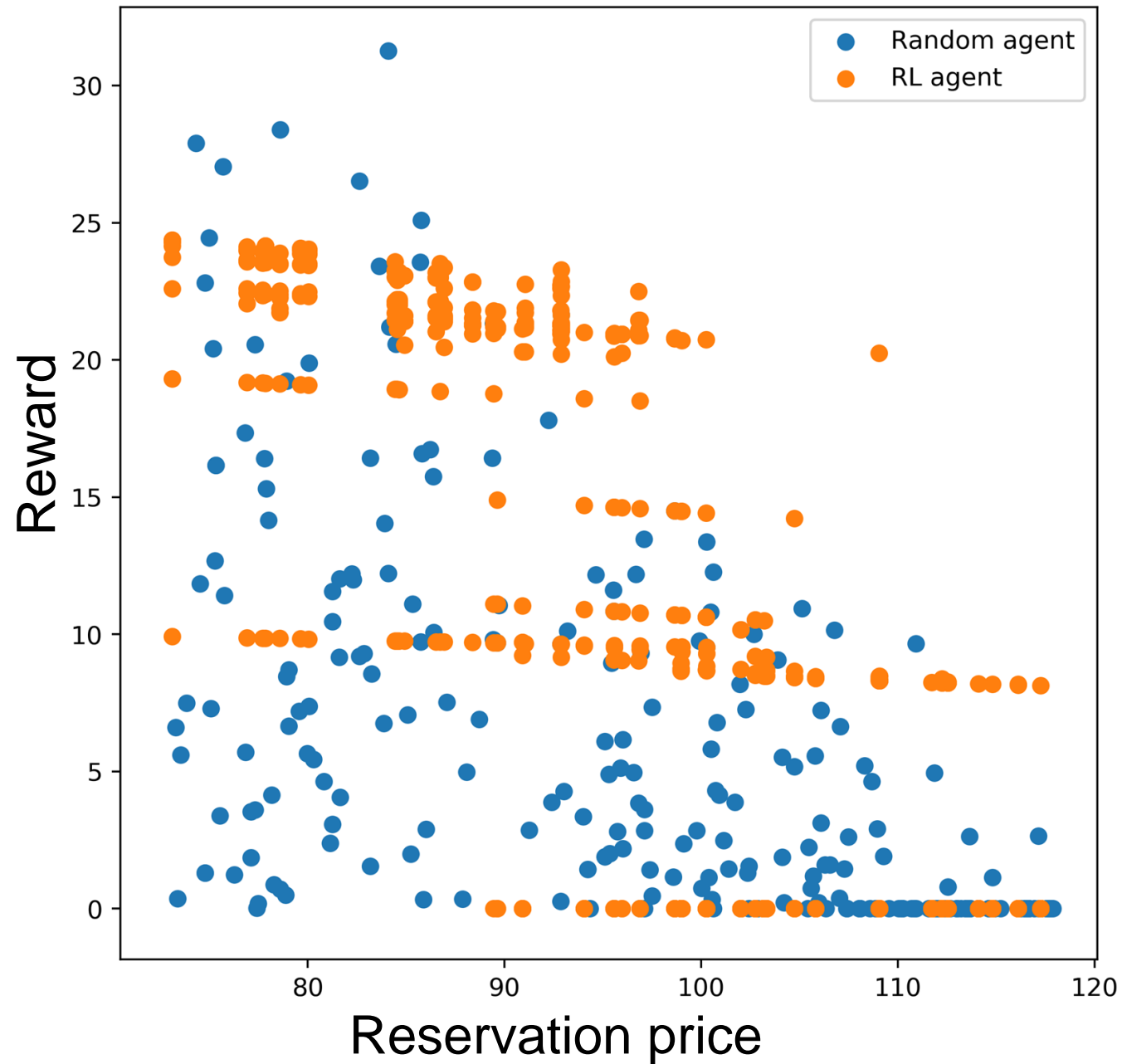


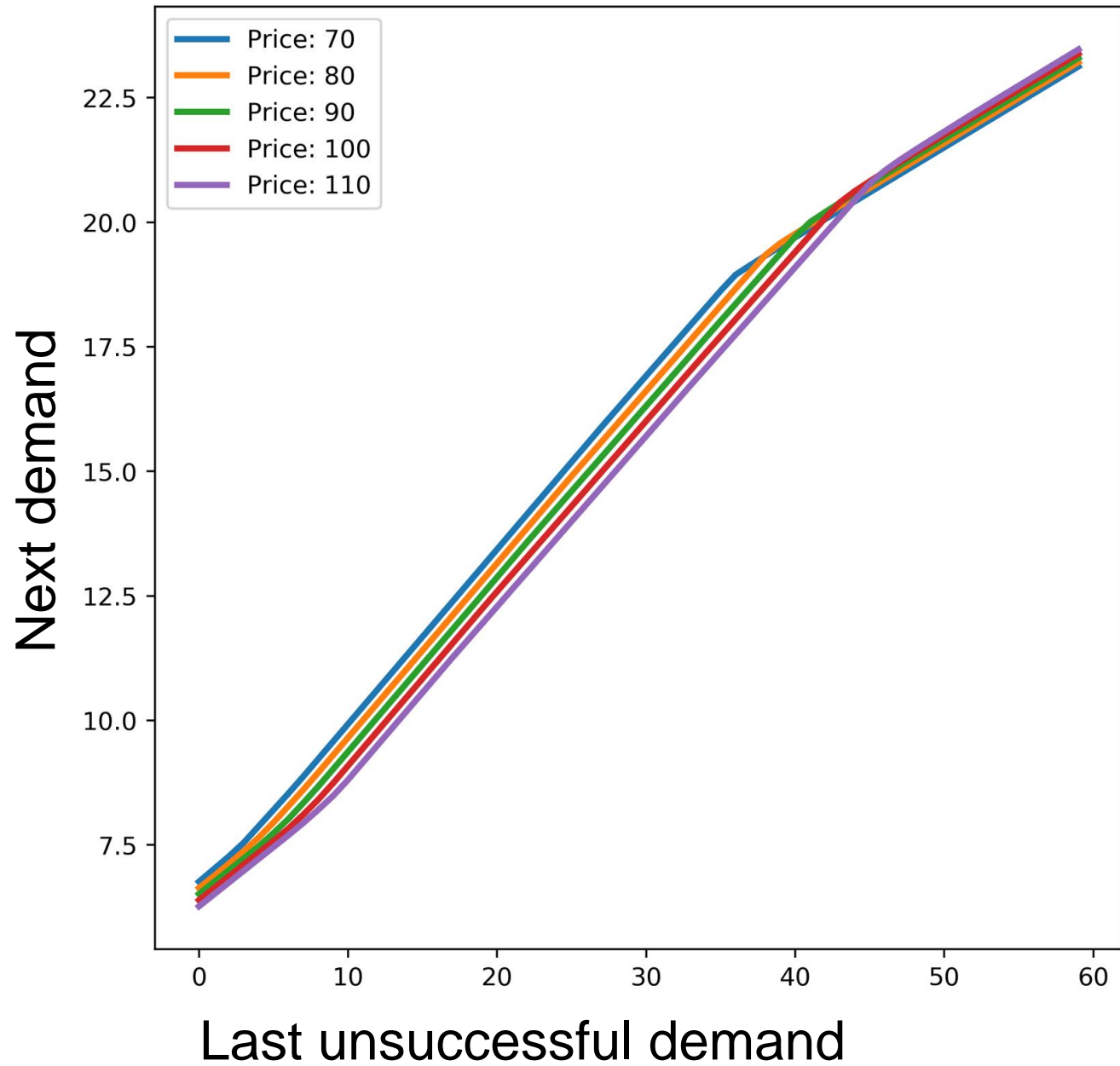


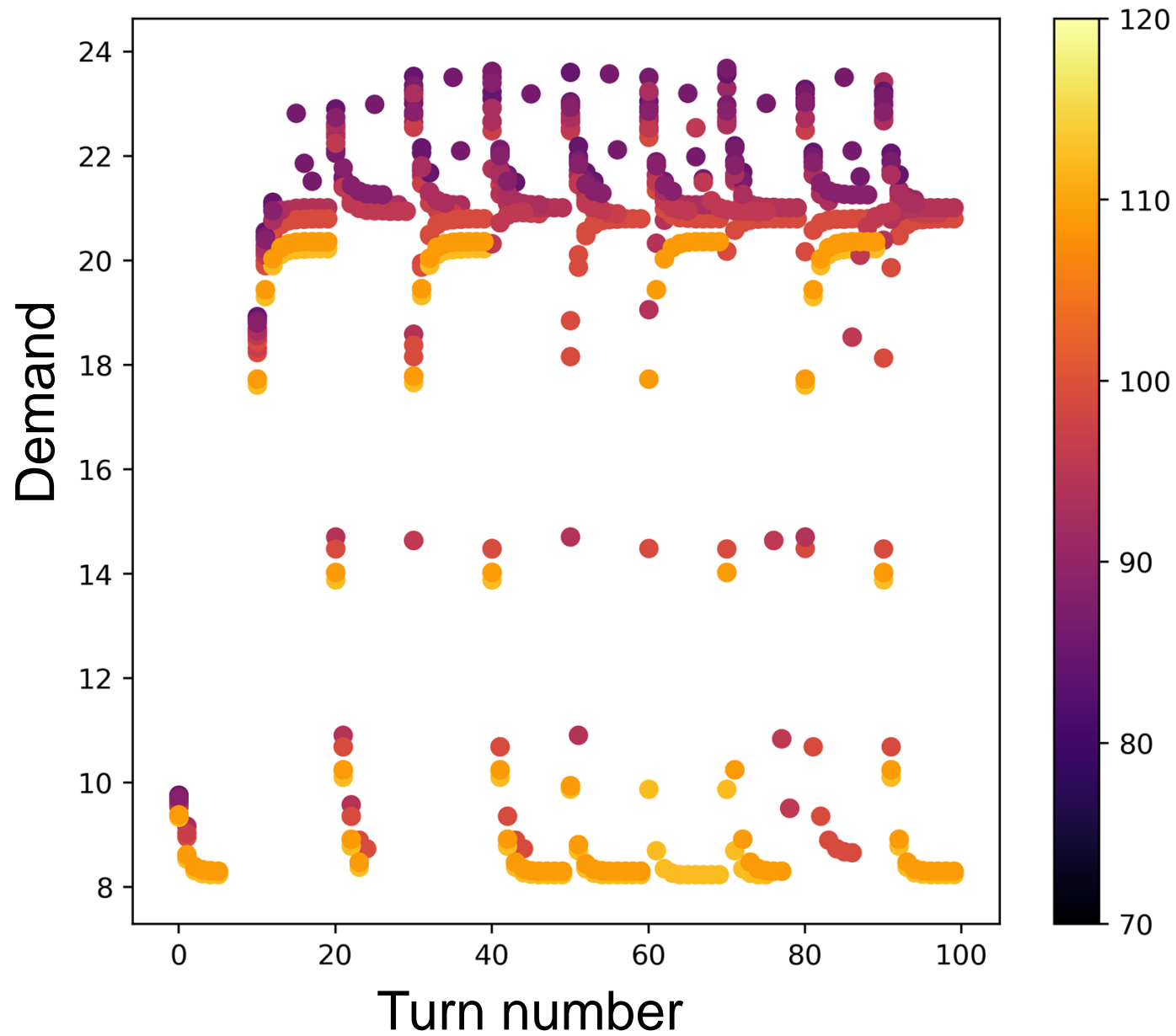


Ornstein-Uhlenbeck (OU) exploration policy









- ★ 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