TRANSNET BW

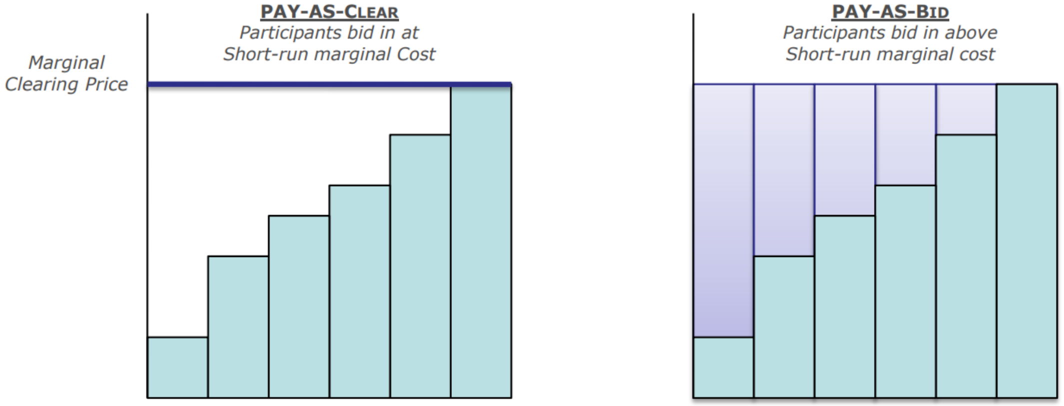
# I AM STILL LEARNING – MARGINAL PRICING IN BALANCING CAPACITY MARKETS WITH STRATEGIC AGENTS

**JORIS DEHLER-HOLLAND, ROBERT GERMESHAUSEN**

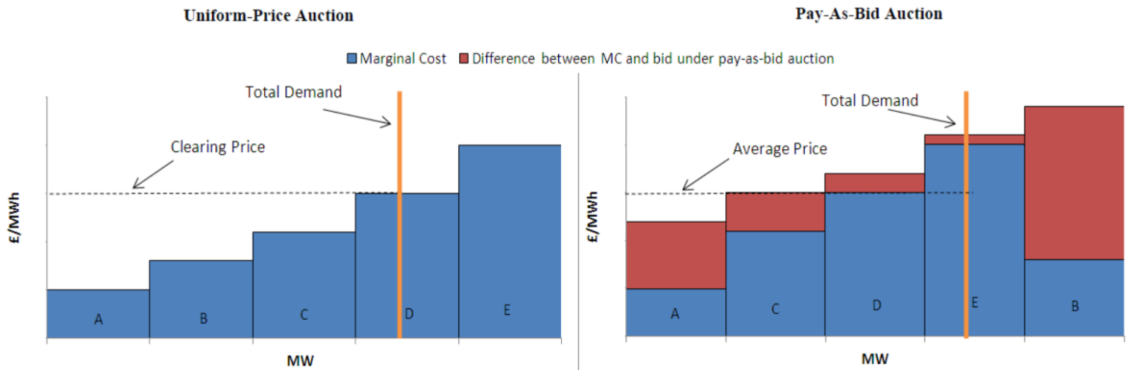
21.03.2024

**Disclaimer**: The views expressed here are those of the authors and do not necessarily represent the views of TransnetBW

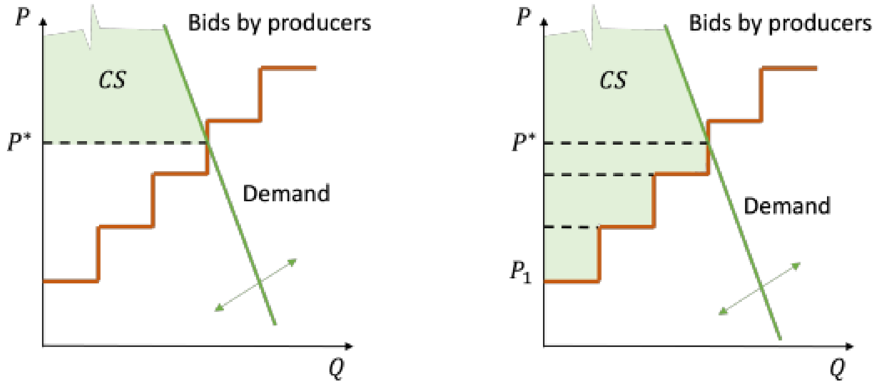
# A TRUE CLASSIC




Ofgem (2012)




Ofgem (2012)




**(a)** A uniform-price auction (UPA). **(b)** A pay-as-bid auction (PABA).

Willems & Yu (2022)

?

# BALANCING CAPACITY MARKETS

/ Balancing capacity:

“volume of reserve capacity that a balancing service provider has agreed to hold and (…) to submit bids for a corresponding volume of balancing energy (…)” [Electricity Balancing Guidelines, EBGL]

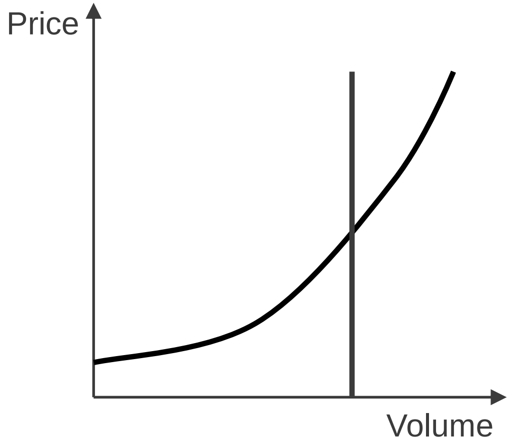
/ Procurement by TSOs:

  / Usually day-ahead before closure of day-ahead electricity market

  / Daily auctions, e.g., separately for upward / downward direction and different validity periods (e.g., 4h blocks)

**Supply**

- Pre-qualificaction of units: no free entry
- (Opportunity) costs of day-ahead market




**Demand**

- Inelastic
- Varying (known in advance)

# INTERNATIONAL COOPERATION IN BALANCING CAPACITY

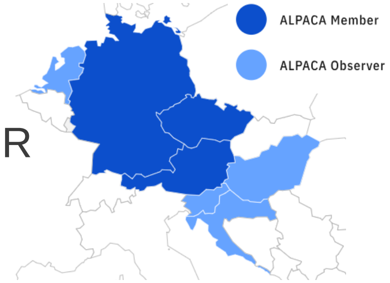
/ Predominantly national procurement, often based on pay-as-bid

/ Framework for cross-border cooperation outlined in European regulation (EBGL)

/ Regional cooperations evolving, e.g.:

FCR


aFRR

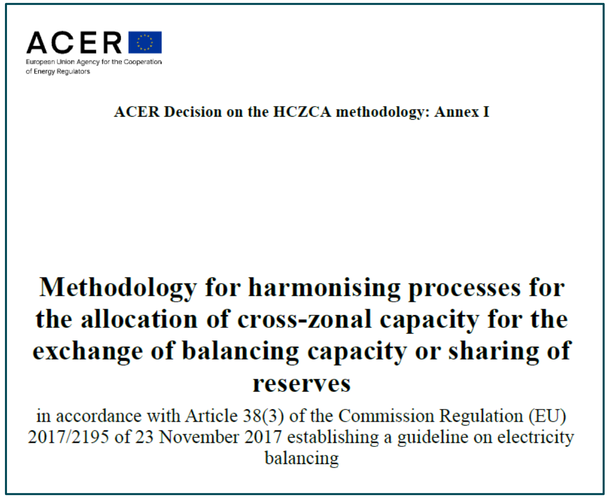



aFRR


/ Details specified in dedicated „Methodologies“, e.g., for allocation of cross-border capacity for balancing capacity

TRANSNET BW

# MARGINAL PRICING MAY BECOME THE STANDARD IN BALANCING CAPACITY COOPERATIONS


ACER

European Union Agency for the Cooperation of Energy Regulators

ACER Decision on the HCZCA methodology: Annex I

**Methodology for harmonising processes for the allocation of cross-zonal capacity for the exchange of balancing capacity or sharing of reserves**

in accordance with Article 38(3) of the Commission Regulation (EU) 2017/2195 of 23 November 2017 establishing a guideline on electricity balancing

**Marginal pricing**
(aka Pay-As-Clear,
aka Uniform Pricing)

**Impact on procurement cost and efficiency?**

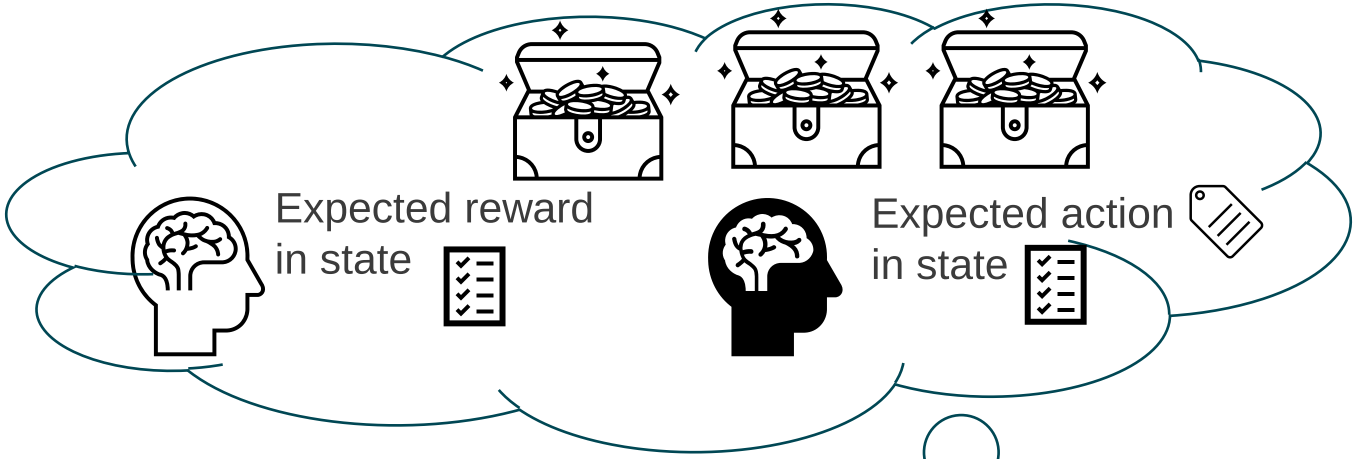
**Change in bidding behavior?**

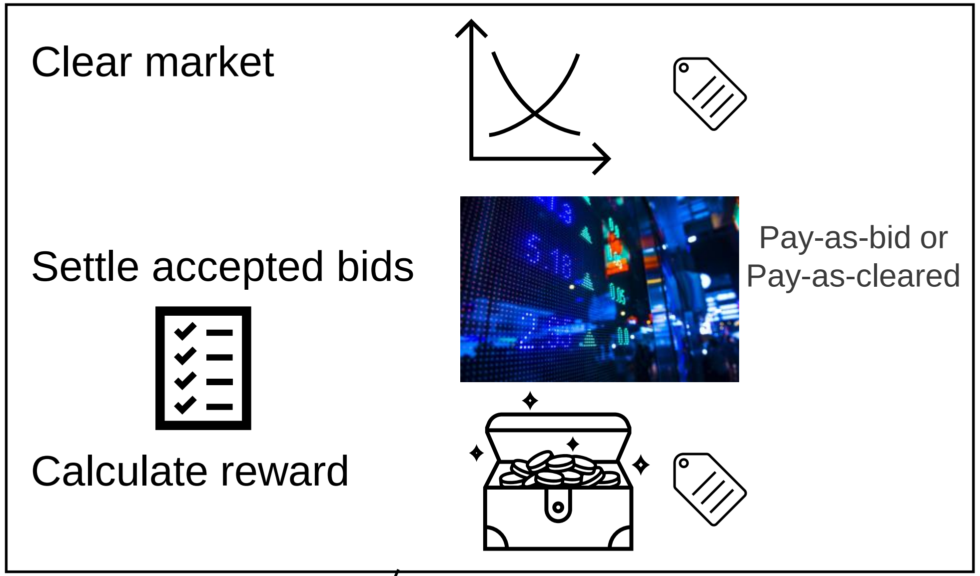
# THIS STUDY

- Research question: How would an introduction of marginal pricing impact bidding behavior and procurement costs in the balancing capacity market?
- How to model bidding behavior under changing circumstances?
  - Agent-based model for the balancing capacity market with deep reinforcement learning
  - Agents learn strategies and adjust their bids to changing market rules (PAB vs. PAC) and the market environment (electricity and fuel prices, supply and demand etc.) → no ex-ante prescription of bidding strategies
- Current status:
  - Two reinforcement learning agents with cost bidding fringe implemented in rllib
  - Balancing capacity in upward direction
  - Scenarios with different levels of competition, i.e., varying the supply of the competitive fringe
  - Perfect forecasting of day-ahead electricity prices

# THE MODEL IN A NUTSHELL

**TD3: “Twin Delayed Deep Deterministic Policy Gradient”**

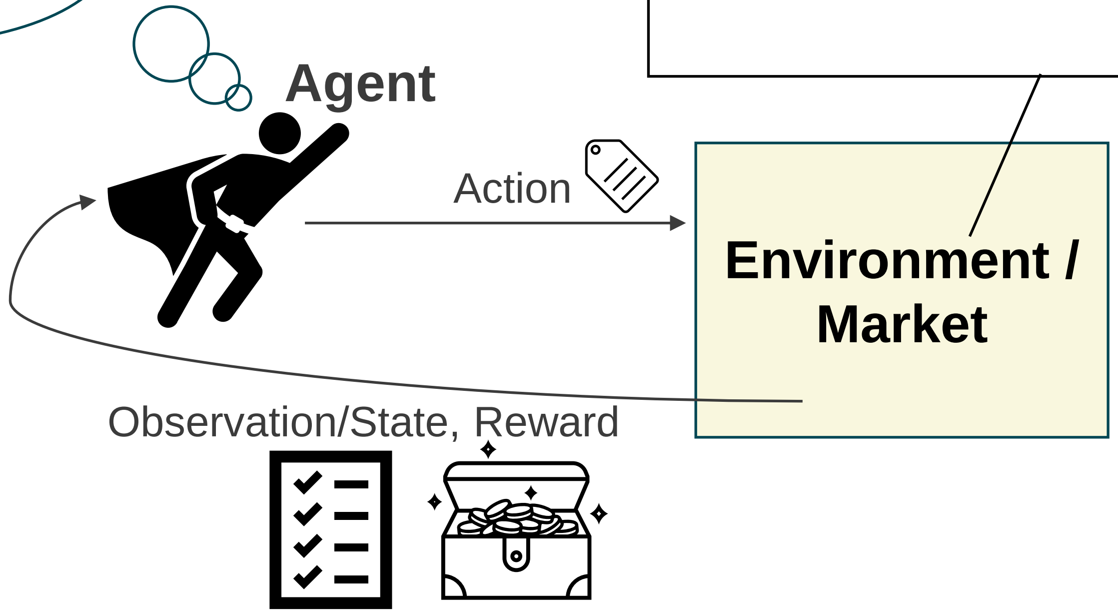



- Marginal price
- Own accepted quantity
- Own settlement price
- Demand
- Day-ahead prices
- Fuel prices

- Prices for two bids

- Profit = revenue - cost



TRANSNET BW

# SCENARIO OVERVIEW

| | Base | Low competition | High competition |
|---|---|---|---|
| 2 RL agents | ~2 GW combined in all scenarios | | |
| Fringe (cost bidding) | ~2 GW | ~1 GW | ~ 4 GW |

/ Demand is varying around ~2 GW in all scenarios

/ Fringe and RL-agents each have two technologies:

  / one oriented on day-ahead electricity market opportunities (storage)

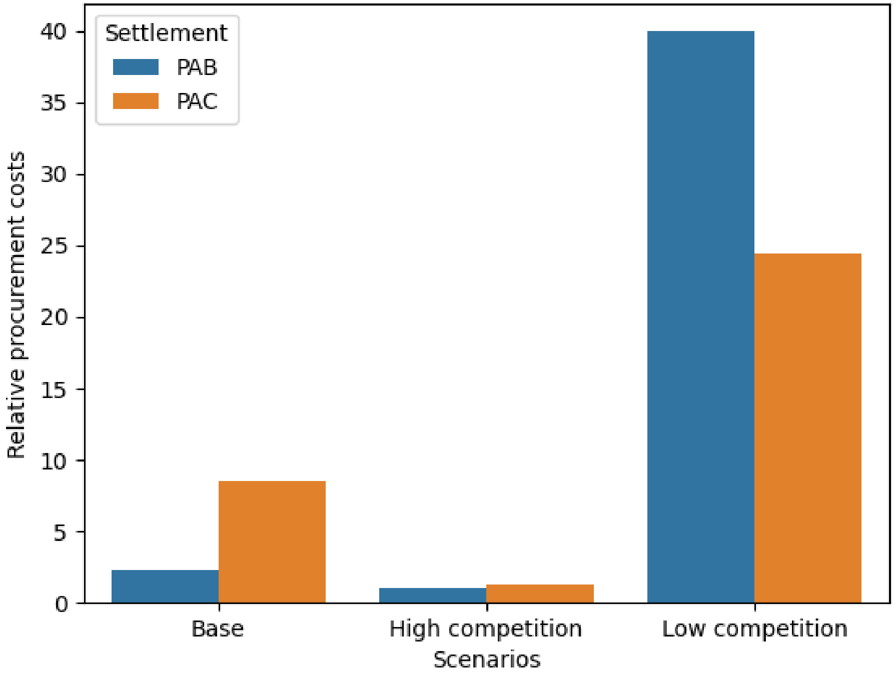
  / one based on natural gas

  / Costs for both technologies are the same for all agents and the fringe

Approach:

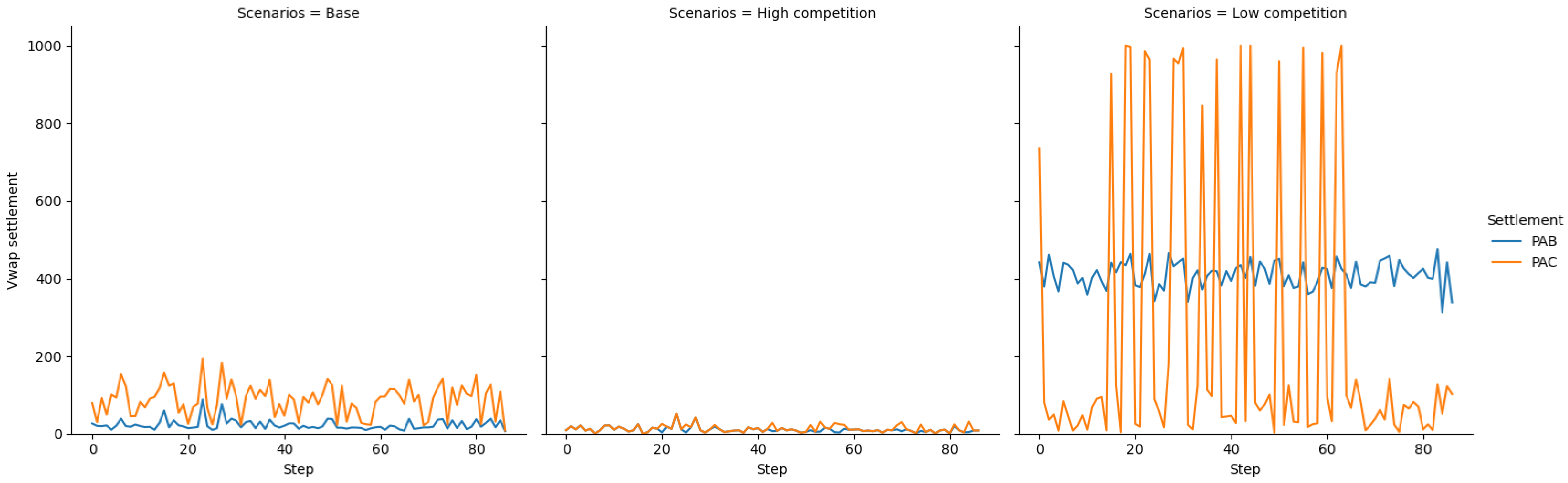
1. Split possible market situations (different demand, day-ahead electricity and fuel prices) into training and test data
2. Train agents on training data (random sampling of market situations from training data for each step)
3. Simulate 100 steps on test data (random sampling of market situations from test data for each step)

# PRELIMINARY RESULTS: PROCUREMENT COSTS RELATIVE TO HIGH COMPETITION SCENARIO (PAY-AS-BID)



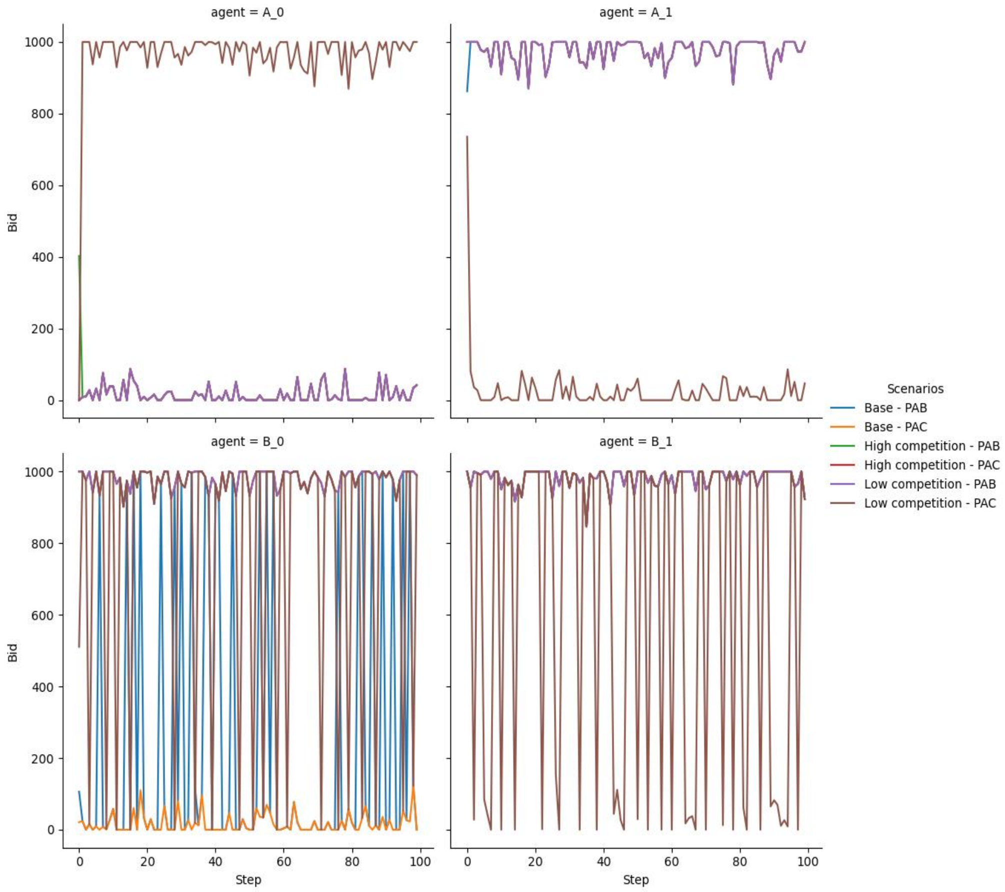

**Disclaimer**: Results are preliminary and may change due to model development.

# PRELIMINARY RESULTS: AVERAGE SETTLED PRICES




**Disclaimer**: Results are preliminary and may change due to model development.

# PRELIMINARY RESULTS: BIDDING BEHAVIOR




**Disclaimer**: Results are preliminary and may change due to model development.

# OUTLOOK

- Inclusion of balancing capacity in downward direction
- More detailed modeling of (opportunity) costs
- Switch to explicit multi-agent algorithm?
- Adjust observation space / reward formulation?
- Explore potential degree of inefficiencies:
  - Compare results to procurement cost with cost bidding
    → which scenarios offer lower margins (difference of bids to cost)?
  - Compare bid selection compared to cost bidding?

# QUESTIONS?

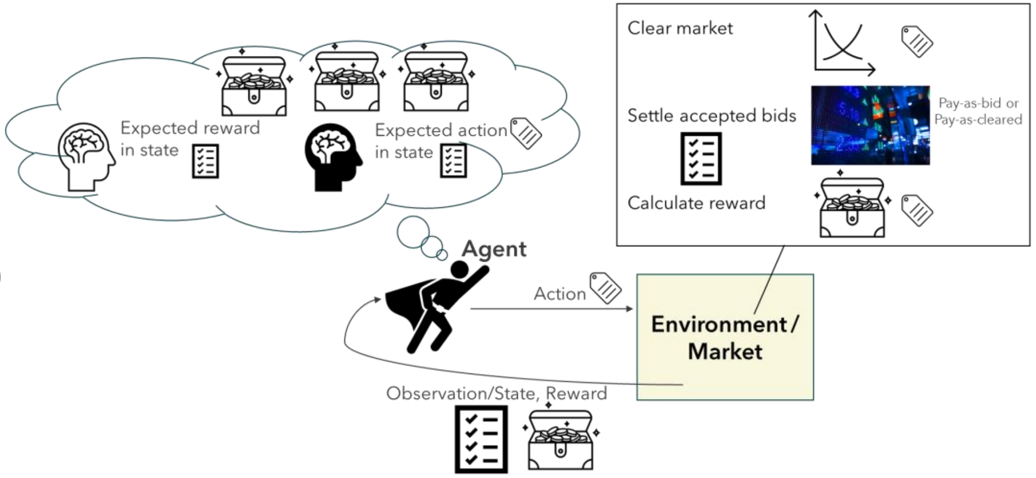
ACER
European Union Agency for the Cooperation of Energy Regulators

ACER Decision on the HCZCA methodology: Annex I

**Methodology for harmonising processes for the allocation of cross-zonal capacity for the exchange of balancing capacity or sharing of reserves**

in accordance with Article 38(3) of the Commission Regulation (EU) 2017/2195 of 23 November 2017 establishing a guideline on electricity balancing

**Marginal pricing**
(aka Pay-As-Clear,
aka Uniform Pricing)

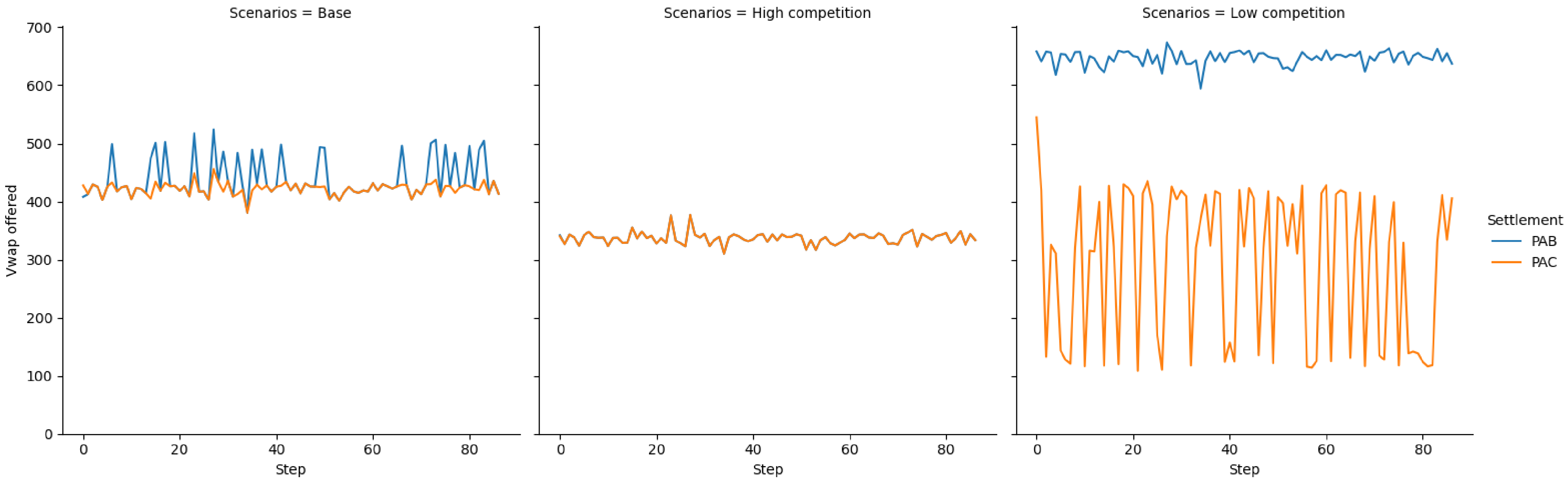



**Change in bidding behavior?**

# HINWEIS ZUR NUTZUNG VON PRÄSENTATIONEN:

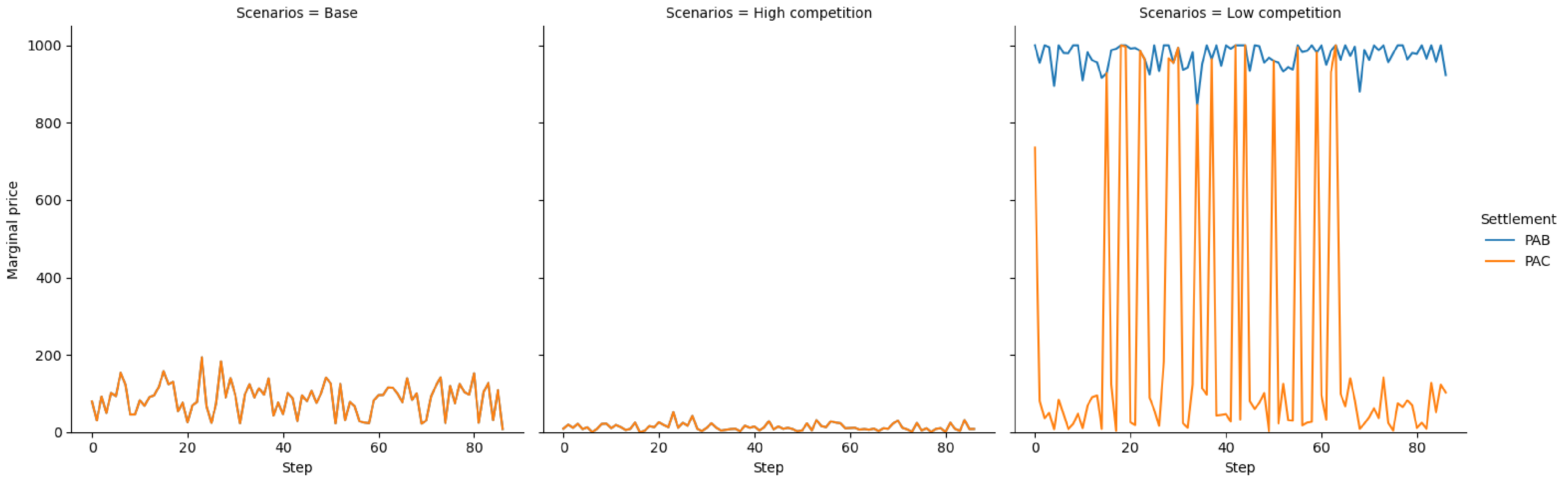
TRANSNET BW

# PRELIMINARY RESULTS: AVERAGE OFFERED PRICES



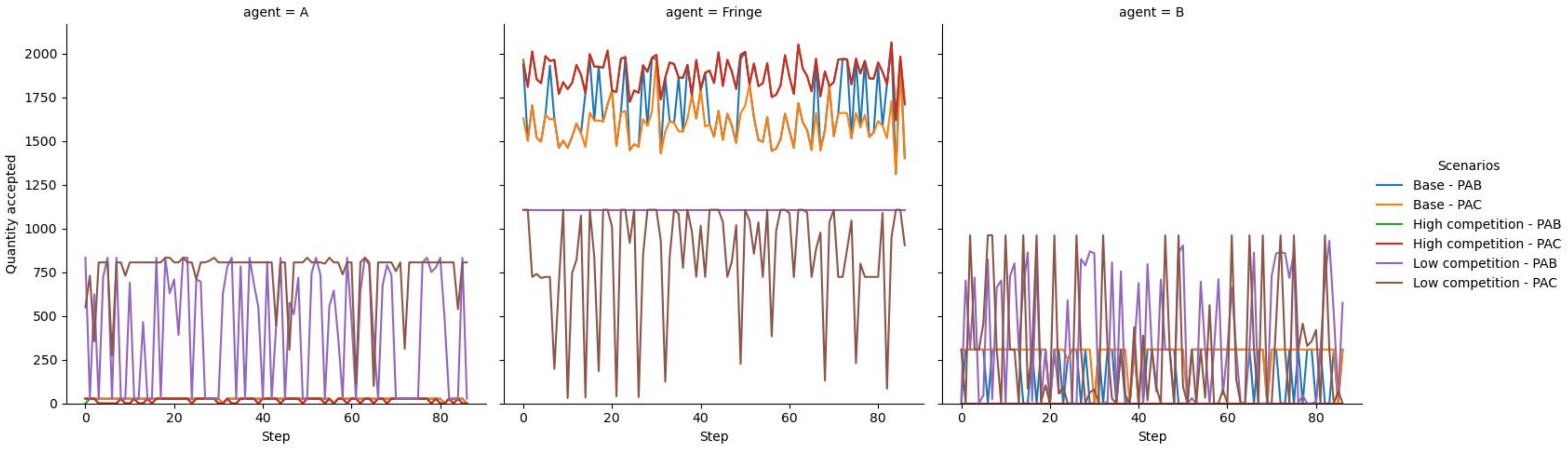

**Disclaimer**: Results are preliminary and may change due to model development.

# PRELIMINARY RESULTS: MARGINAL PRICES



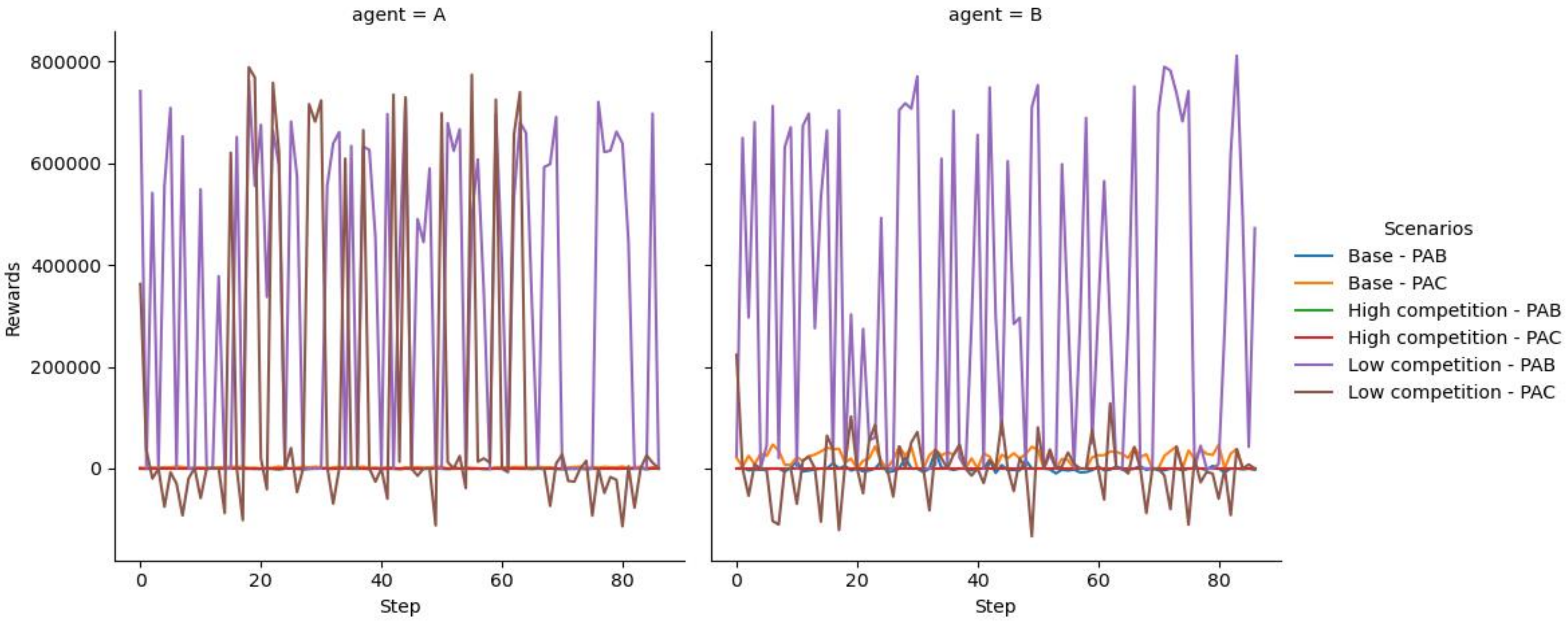

**Disclaimer**: Results are preliminary and may change due to model development.

# PRELIMINARY RESULTS: QUANTITY ACCEPTED




**Disclaimer**: Results are preliminary and may change due to model development.

# PRELIMINARY RESULTS: REWARDS




**Disclaimer**: Results are preliminary and may change due to model development.

# TD3 – DESCRIPTION

This approach is closely connected to Q-learning, and is motivated the same way: if you know the optimal action-value function $Q^*(s,a)$, then in any given state, the optimal action $a^*(s)$ can be found by solving

$$a^*(s) = \arg\max_a Q^*(s,a).$$

## Value (Critic)

$$\max_a Q(s,a) \approx Q(s,\mu(s)).$$

## Policy (Actor)

$$\max_\theta \mathop{\mathrm{E}}_{s\sim\mathcal{D}} \left[ Q_\phi(s,\mu_\theta(s)) \right].$$

**Trick One: Clipped Double-Q Learning.** TD3 learns *two* Q-functions instead of one (hence "twin"), and uses the smaller of the two Q-values to form the targets in the Bellman error loss functions.

**Trick Two: "Delayed" Policy Updates.** TD3 updates the policy (and target networks) less frequently than the Q-function. The paper recommends one policy update for every two Q-function updates.

**Trick Three: Target Policy Smoothing.** TD3 adds noise to the target action, to make it harder for the policy to exploit Q-function errors by smoothing out Q along changes in action.

**Algorithm 1** Twin Delayed DDPG

Input: initial policy parameters $\theta$, Q-function parameters $\phi_1$, $\phi_2$, empty replay buffer $\mathcal{D}$
Set target parameters equal to main parameters $\theta_{\text{targ}} \leftarrow \theta$, $\phi_{\text{targ},1} \leftarrow \phi_1$, $\phi_{\text{targ},2} \leftarrow \phi_2$
**repeat**
  Observe state $s$ and select action $a = \text{clip}(\mu_\theta(s) + \epsilon, a_{Low}, a_{High})$, where $\epsilon \sim \mathcal{N}$
  Execute $a$ in the environment
  Observe next state $s'$, reward $r$, and done signal $d$ to indicate whether $s'$ is terminal
  Store $(s,a,r,s',d)$ in replay buffer $\mathcal{D}$
  If $s'$ is terminal, reset environment state.
  **if** it's time to update **then**
    **for** $j$ in range(however many updates) **do**
      Randomly sample a batch of transitions, $B = \{(s,a,r,s',d)\}$ from $\mathcal{D}$
      Compute target actions

$$a'(s') = \text{clip}\left(\mu_{\theta_{\text{targ}}}(s') + \text{clip}(\epsilon,-c,c), a_{Low}, a_{High}\right), \quad \epsilon \sim \mathcal{N}(0,\sigma)$$

      Compute targets

$$y(r,s',d) = r + \gamma(1-d)\min_{i=1,2} Q_{\phi_{\text{targ},i}}(s', a'(s'))$$

      Update Q-functions by one step of gradient descent using

$$\nabla_{\phi_i} \frac{1}{|B|} \sum_{(s,a,r,s',d)\in B} \left(Q_{\phi_i}(s,a) - y(r,s',d)\right)^2 \qquad \text{for } i = 1,2$$

      **if** $j$ mod `policy_delay` $= 0$ **then**
        Update policy by one step of gradient ascent using

$$\nabla_\theta \frac{1}{|B|} \sum_{s\in B} Q_{\phi_1}(s, \mu_\theta(s))$$

        Update target networks with

$$\begin{aligned} \phi_{\text{targ},i} &\leftarrow \rho\phi_{\text{targ},i} + (1-\rho)\phi_i \qquad \text{for } i = 1,2 \\ \theta_{\text{targ}} &\leftarrow \rho\theta_{\text{targ}} + (1-\rho)\theta \end{aligned}$$

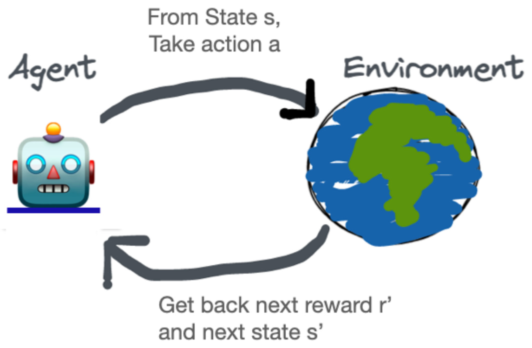
      **end if**
    **end for**
  **end if**
**until** convergence

# IMPLEMENTATION WITH RAY AND RLlib

Solving a problem in RL begins with an **environment**. In the simplest definition of RL:

> An **agent** interacts with an **environment** and receives a reward.

An environment in RL is the agent's world, it is a simulation of the problem to be solved.

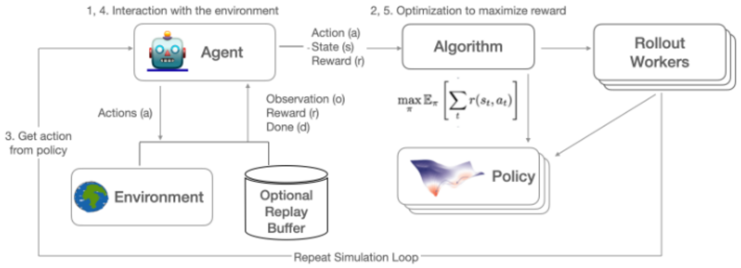



An RLlib environment consists of:

1. all possible actions (**action space**)
2. a complete description of the environment, nothing hidden (**state space**)
3. an observation by the agent of certain parts of the state (**observation space**)
4. **reward**, which is the only feedback the agent receives per action.

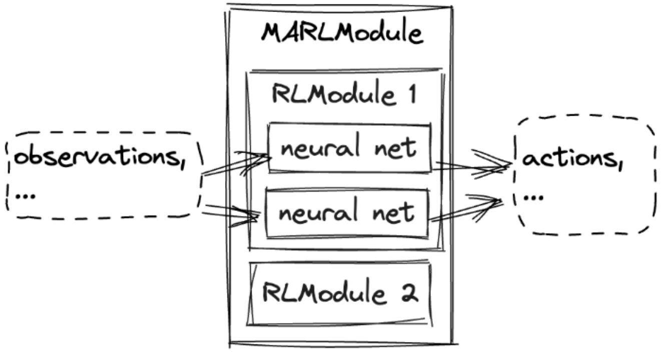
The model that tries to maximize the expected sum over all future rewards is called a **policy**.

policy is a function mapping the environment's observations to an action to take, usually written **π** (s(t)) -> a(t). Below is a diagram of the RL iterative learning process.




The RL simulation feedback loop repeatedly collects data, for one (single-agent case) or multiple (multi-agent case) policies, trains the policies on these collected data, and makes sure the policies' weights are kept in sync. Thereby, the collected environment data contains observations, taken actions, received rewards and so-called **done** flags, indicating the boundaries of different episodes the agents play through in the simulation.

The simulation iterations of action -> reward -> next state -> train -> repeat, until the end state, is called an **episode**, or in RLlib, a **rollout**. The most common API to define environments is the [Farama-Foundation Gymnasium] API, which we also use in most of our examples.



# ACTOR AND CRITIC NN PARAMETERS AND LEARNING HYPER-PARAMETERS

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| Critic NN architecture (hidden layer) | MLP, (400, 300) | Batch size | 100 |
| Actor NN architecture (hidden layer) | MLP, (400, 300) | Reward discount | 0.99 |
| Critic activation function | ReLU | Policy delay | 2 |
| Actor activation function | ReLU | Soft-update | 0.005 |
| Observation size | 15 | Target noise | 0.1 |
| Action size | 2 | Target noise clip | 0.5 |
| Optimizer, learning rate | Adam, $10^{-3}$ | Action noise | Gaussian |

TRANSNET BW

# OVERVIEW OF TESTCASES

| Testcase | Cost | Capacity | Result „pay-as-bid“ | Result „pay-as-clear“ |
|---|---|---|---|---|
| Pivotal RL-Agent (1 RL-Agent, 1 Fringe) | RL-Agent: 1<br>Fringe: 6 | RL-Agent: 10<br>Fringe: 1990 | Bids close to maximum price (~10) ✓ | |
| Competition (1 RL-Agent, 3 Fringe-Suppliers) | RL-Agent: 1<br>Fringe: 2, 4, 6 | RL-Agent: 10<br>Fringe: 995, 995, 10 | Bid just under most expensive Fringe bid (<6) ✓ | Very low bid (~0) ✓ ✗ |
| Agent Duopoly (2 RL-Agents, no Fringe) | RL-Agent A-B: 5 | RL-Agent A-B: 2000 | Both bid above their cost (~8) ✓ ✗ | Both bid above their cost (~8) ✓ ✗ |
| 4-Agents Oligopoly (4 RL-Agents, no Fringe) | RL-Agent A - B: 5 | RL-Agent A - B: 2000 | Convergence (?) of bids above costs (~7) ✓ ✗ | Convergence (?) of bids above costs (~8), except for one ✗ ✓ |
| 4-Agents Oligopoly with Fringe (4 RL-Agents, Fringe) | RL-Agent A - D: 5 | RL-Agent A - D: 1000 | Convergence of bids above own cost und below Fringe cost (~6) ✓ | Convergence of bids above own cost und below Fringe cost (~6) ✓ |

Demand in all scenarios: 2000

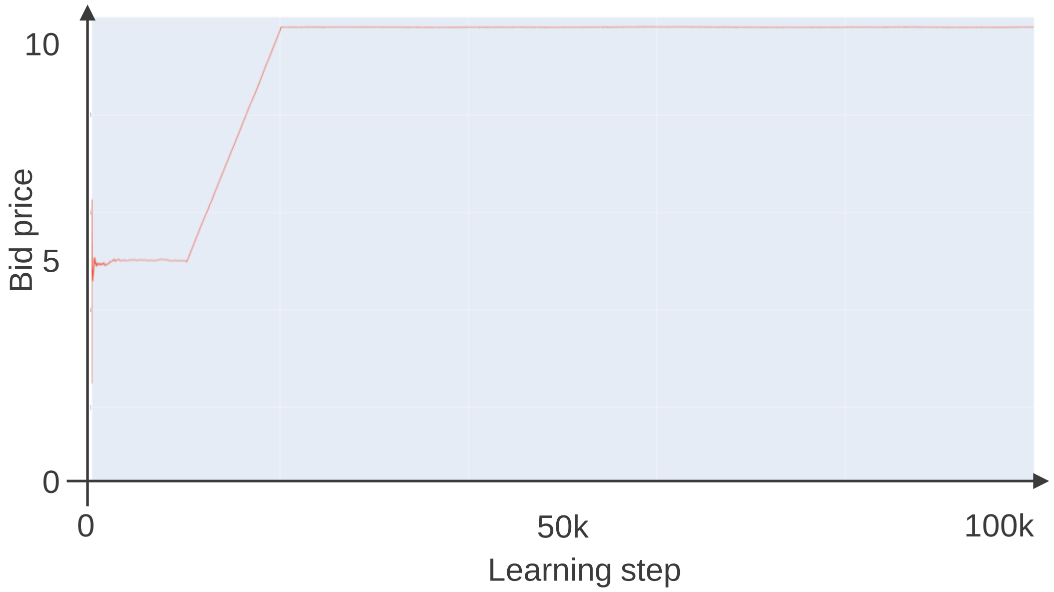
✓ As expected ✗ Not as expected

TRANSNET BW

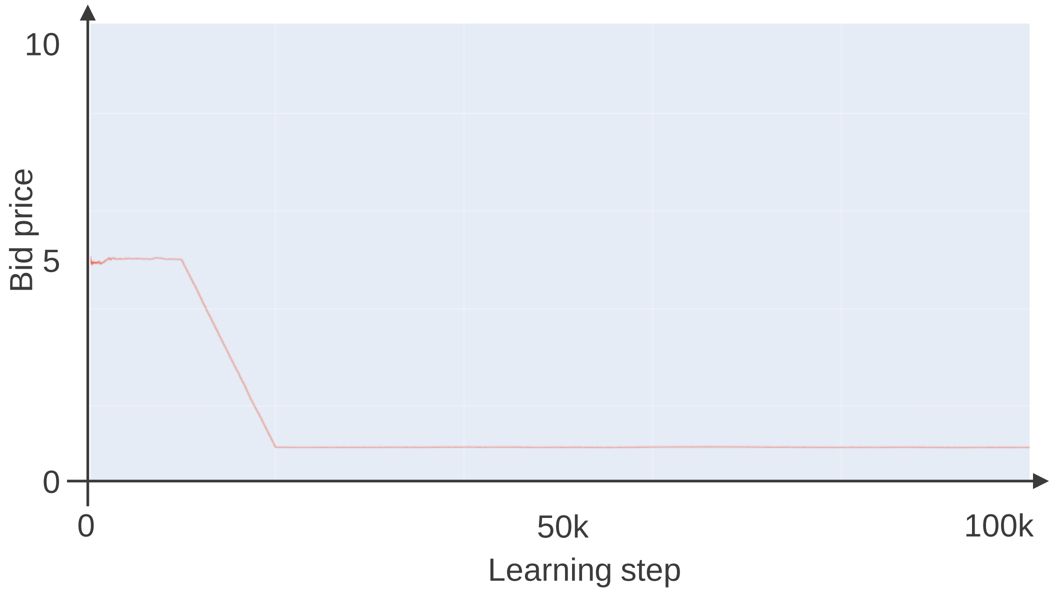
# TESTCASE „PIVOTAL RL-AGENT“

- Cost: RL-Agent (1), Fringe (6)
- Capacity: RL-Agent (10), Fringe (1990)
- Demand: 2000

## PAY-AS-BID




## PAY-AS-CLEAR



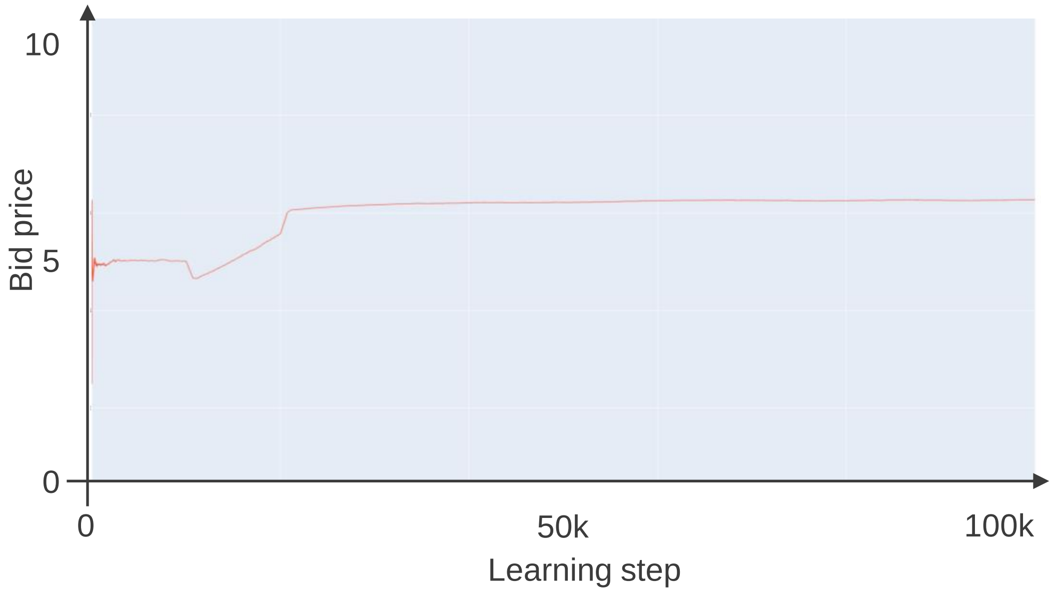

PAB: Maximum price
PAC: Maximum price

TRANSNET BW

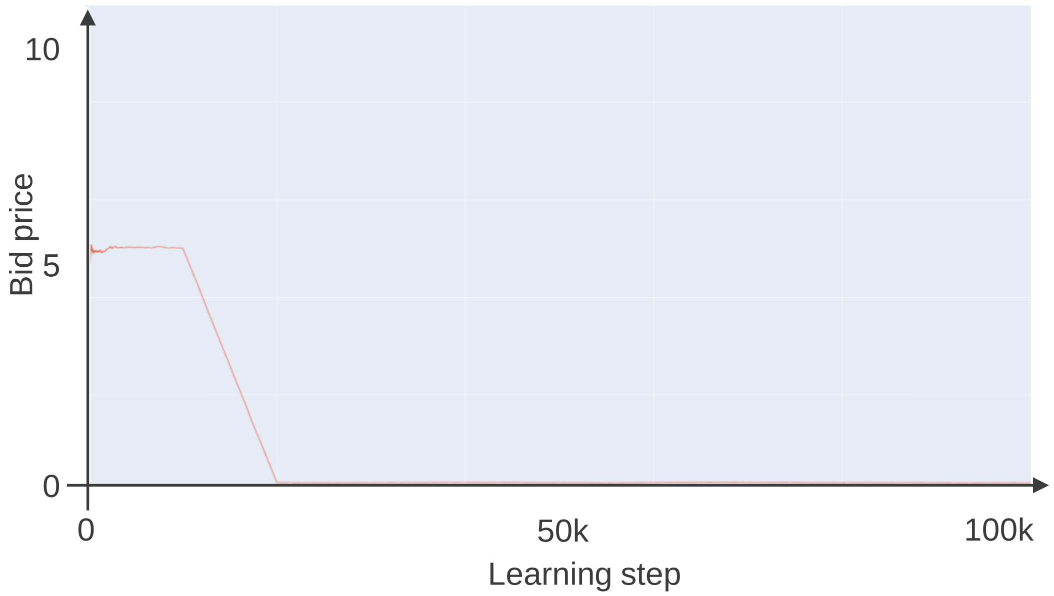
# TESTCASE "COMPETITION"

- Cost: RL-Agent (1), Fringe (2,4,6)
- Capacity: RL-Agent (10), Fringe (995,995, 10)
- Demand: 2000

## PAY-AS-BID




## PAY-AS-CLEAR

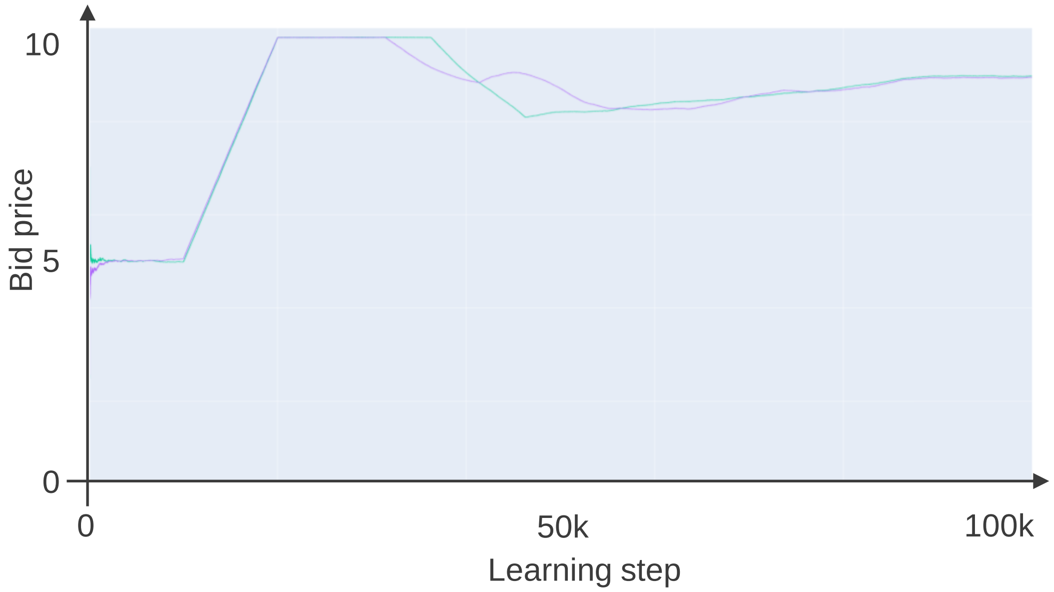



PAB: Bid below last fringe bid
PAC: Bid close to 0

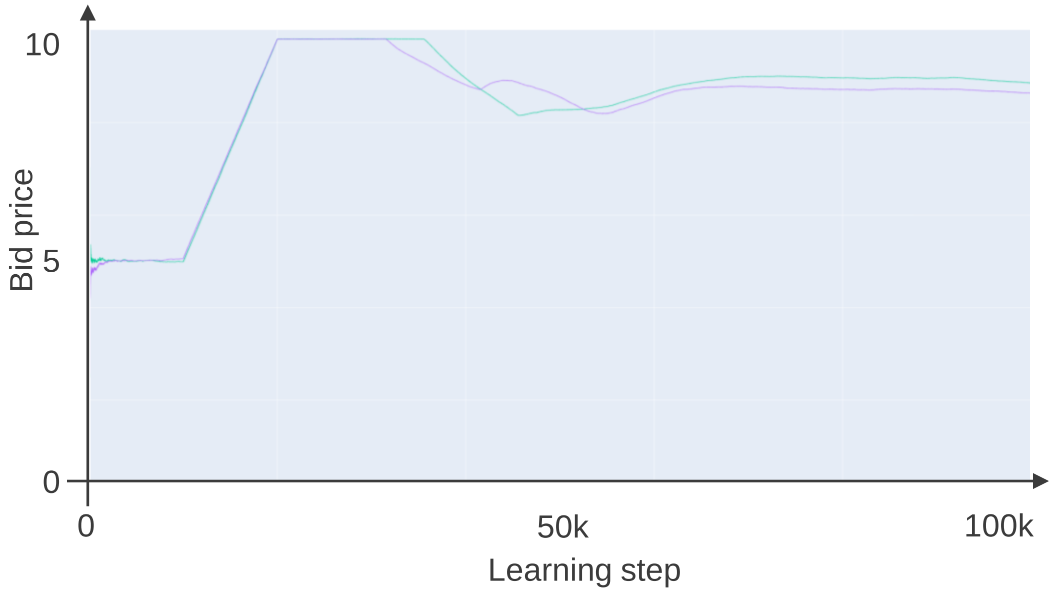
# TESTCASE “AGENT DUOPOLY”

- Cost: RL-Agent A (5), RL-Agent B (5)
- Capacity: RL-Agent A (2000), RL-Agent B (2000)
- Demand: 2000

## PAY-AS-BID




## PAY-AS-CLEAR



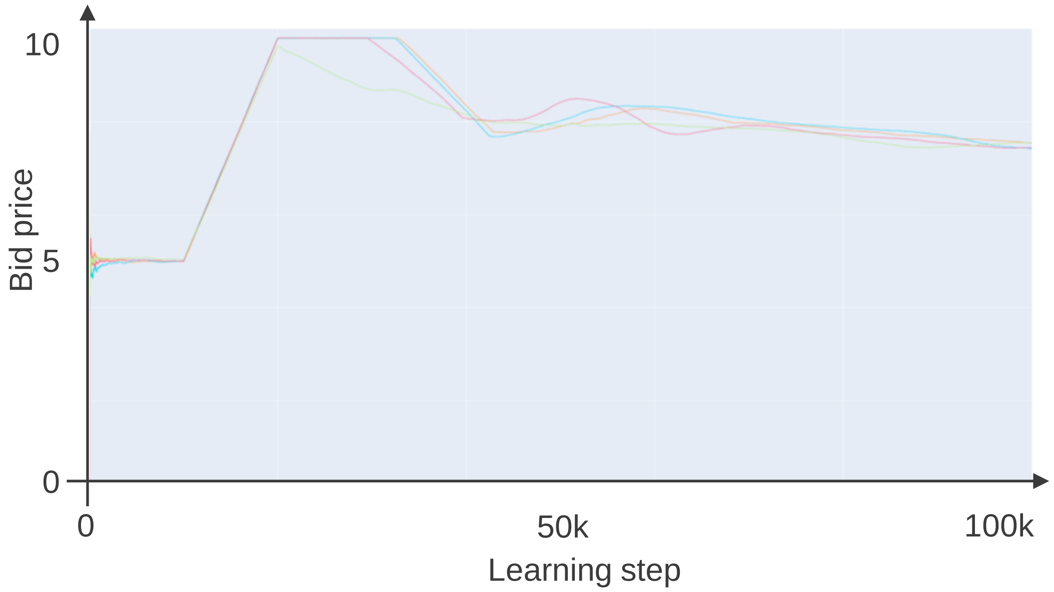

PAB and PAC: Bids close to but below maximum price (but stable?)

TRANSNET BW

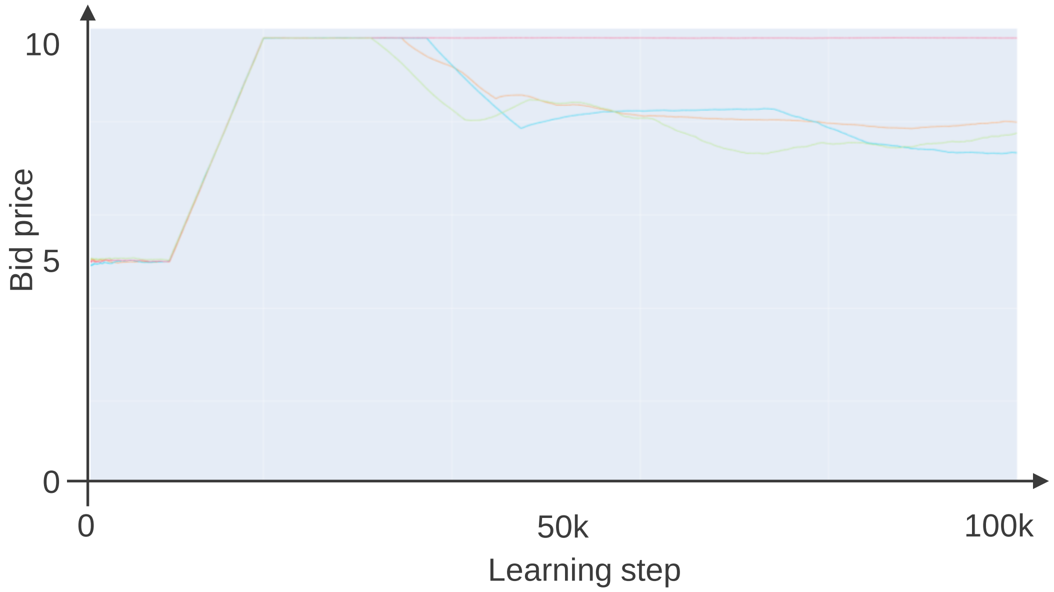
# TESTCASE “4-AGENT OLIGOPOLY“

- Cost: RL-Agent A (5), RL-Agent B (5), RL-Agent C (5), RL-Agent D (5)
- Capacity: RL-Agent A (1000), RL-Agent B (1000), RL-Agent C (1000), RL-Agent D (1000)
- Demand: 2000

## PAY-AS-BID




## PAY-AS-CLEAR

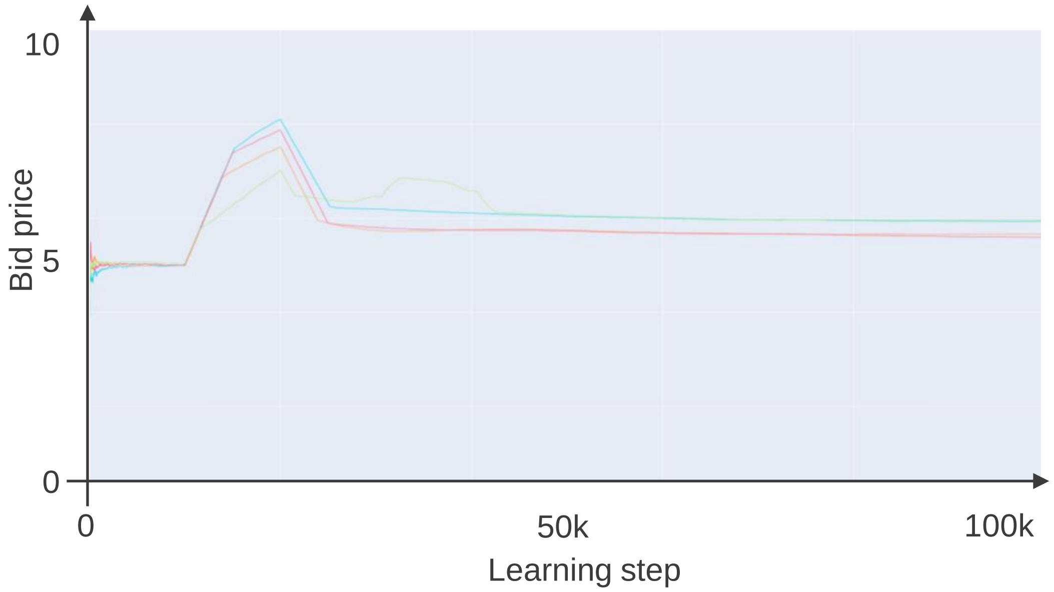



PAB: Convergence of bids (above costs)?
PAC: Convergence of 3 bids (above costs)?

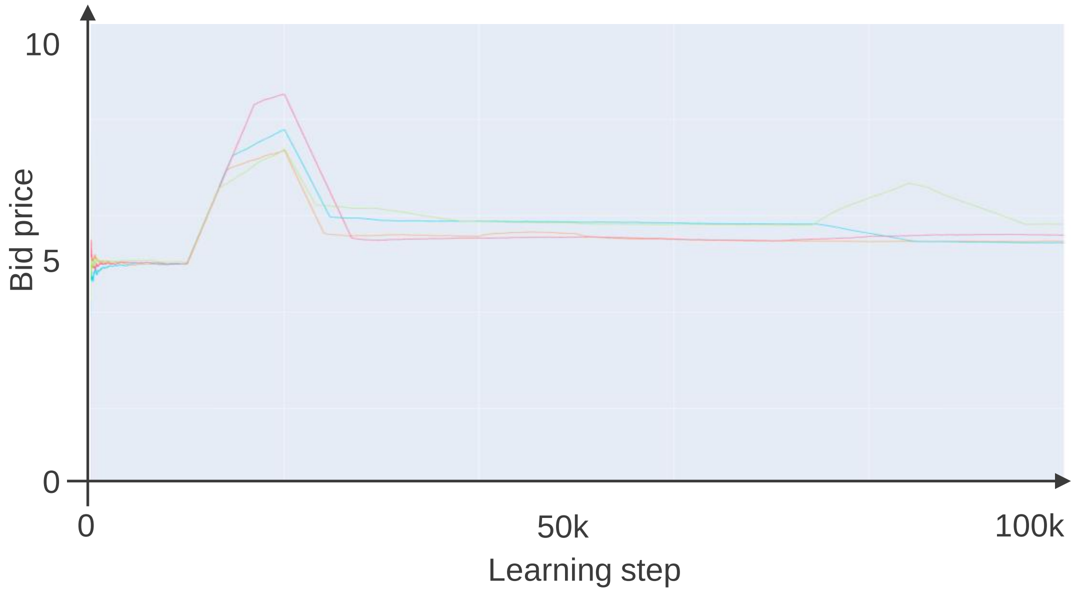
# TESTCASE “4-AGENT OLIGOPOLY WITH FRINGE”

- Cost: RL-Agent A (5), RL-Agent B (5), RL-Agent C (5), RL-Agent D (5), Fringe (7)
- Capacity: RL-Agent A (1000), RL-Agent B (1000), RL-Agent C (1000), RL-Agent D (1000), Fringe (2000)
- Demand: 2000

## PAY-AS-BID




## PAY-AS-CLEAR




PAB and PAB: Convergence of bids below fringe cost