Recent Insights in Value-based Deep Reinforcement Learning

Prabhat Nagarajan

The data distribution is an extremely important factor in off-policy value-based deep RL

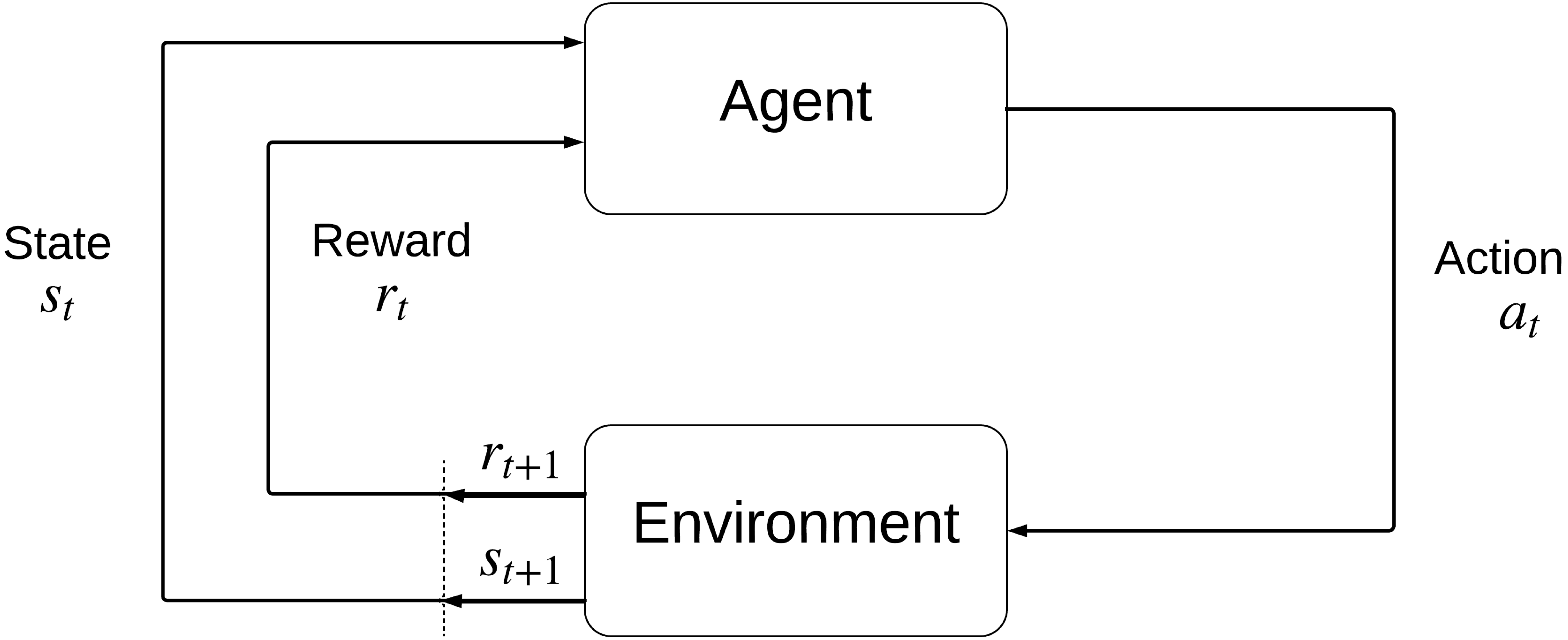
Outline

1. Background on reinforcement learning (RL) & Double deep Q-networks

2. Works

- The Tandem Effect
- Policy Churn
- The Curse of Diversity in Ensemble Exploration

Reinforcement Learning



Modeled after diagram from Sutton & Barto (2018)

Policies and Returns

- Learn policy $\pi(a \mid s)$ that yields maximum *expected discounted return:*

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid S_0 = s\right]$$ *, where* $\gamma \in [0,1)$ *is the discount factor.*

- Optimal policy is denoted π^*, a policy that maximizes expected discounted return

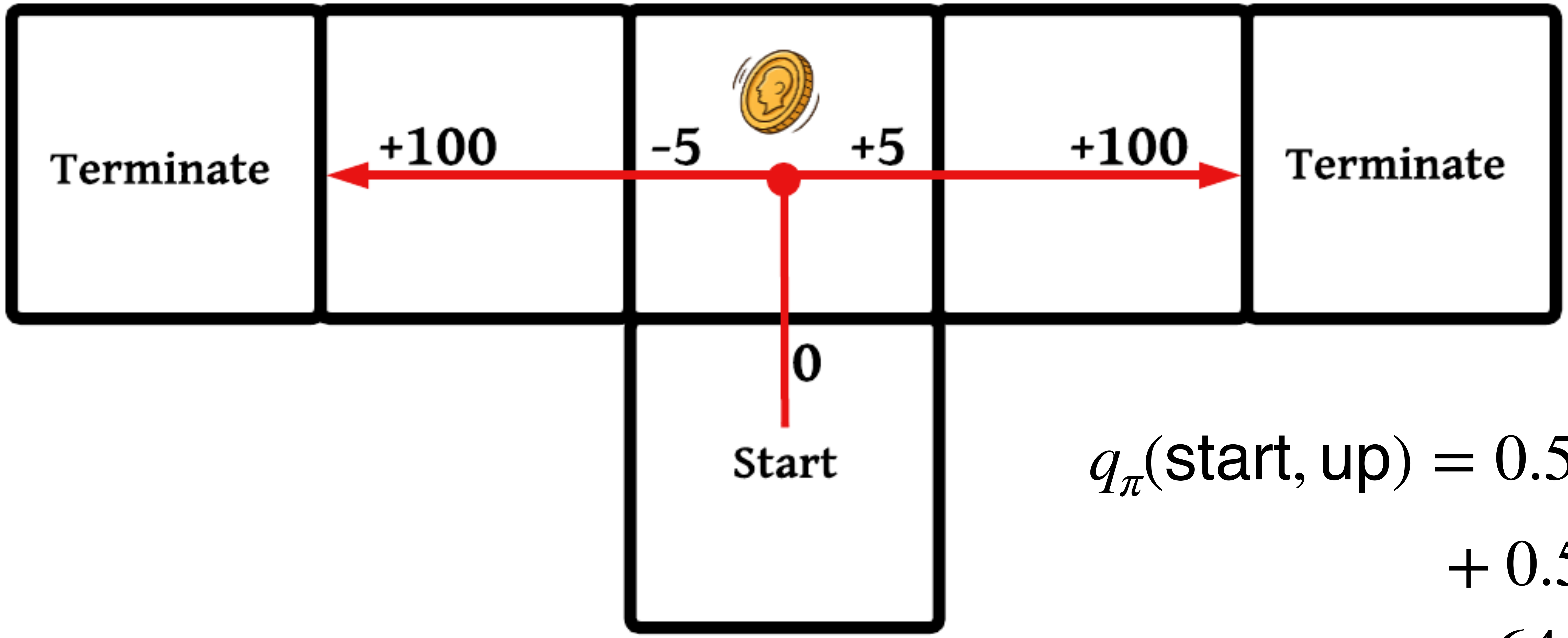
Value-based Reinforcement Learning

- The *action-value function* for a policy π is:

$$q_\pi(s,a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \,|\, S_0 = s, A_0 = a \right]$$

- Value based-control: Learn optimal policy indirectly through an optimal *value function.*
  - aim to learn $q_{\pi*}$, often denoted $q*$
  - Then in any state s can take action $\text{argmax}_a q*(s,a)$ in every state

Simple environment



$$\gamma = 0.8$$

$$\begin{aligned} q_\pi(\text{start}, \text{up}) &= 0.5(0 + 0.8 \cdot (-5) + 0.8^2(100)) \\ &\quad + 0.5(0 + 0.8 \cdot (5) + 0.8^2(100)) \\ &= 64 \end{aligned}$$

$$\begin{aligned} q^*(\text{start}, \text{up}) &= 0 + 0.8 \cdot 5 + 0.8^2(100)) \\ &= 68 \end{aligned}$$

Double Deep Q-Networks(Double DQN)[1]

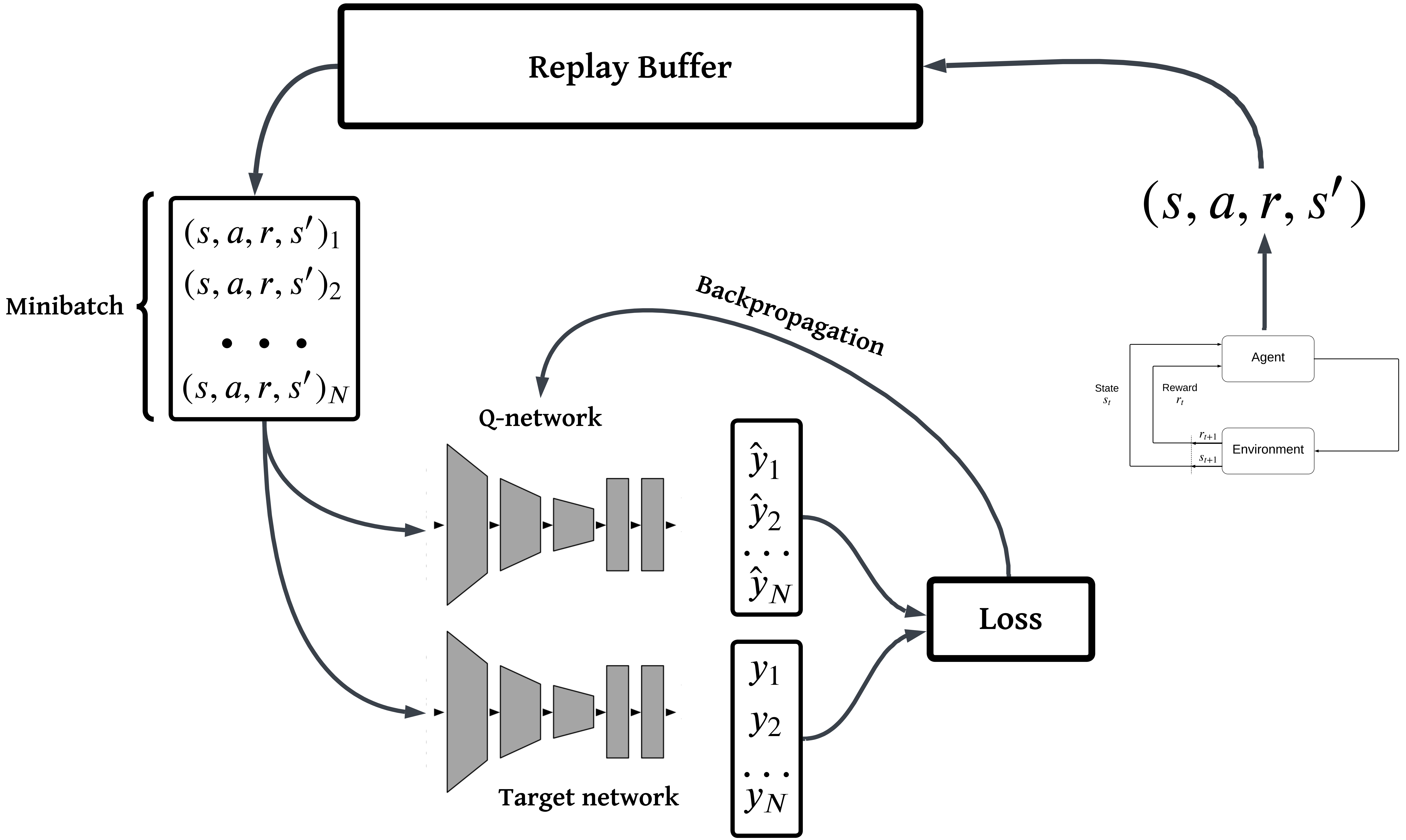
- Trains a *Q-network* θ, where $Q(s, a; \theta)$ is prediction for state-action pair (s, a)
- Acts ϵ-greedily, $\epsilon \in [0,1]$
    - With probability 1-ϵ selects a *greedy* action: **argmax**$_a Q(s, a; \theta)$
    - With probability ϵ selects a random action
- ϵ is usually annealed to a low value: $\epsilon = 0.01$
    - Acting rather greedily
- Stores (s, a, r, s') in a *replay buffer* (a large dataset of the last 1M transitions)

[1] van Hasselt et al. (2016). Deep Reinforcement Learning with Double Q-learning. AAAI.

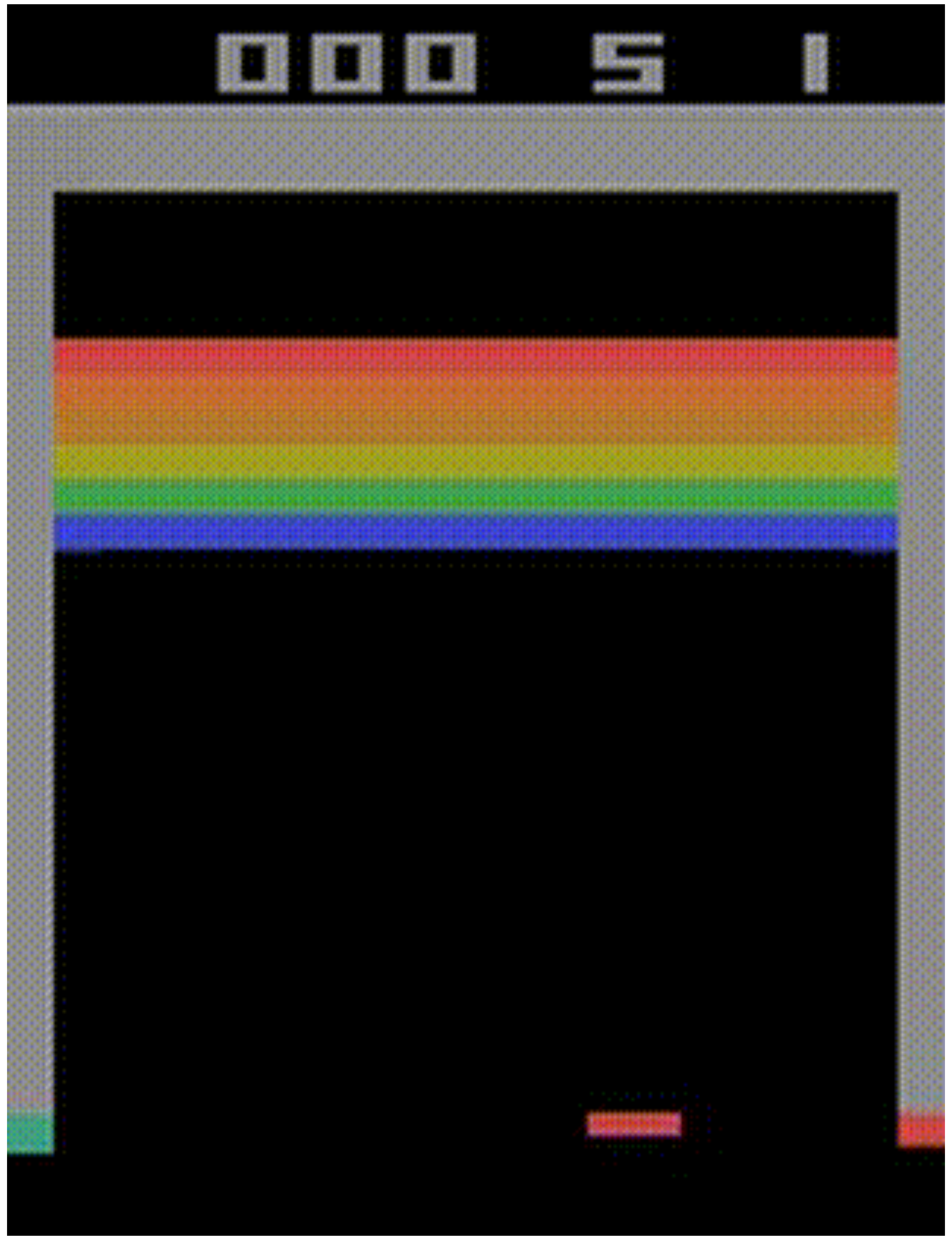
Double DQN: Update Rules

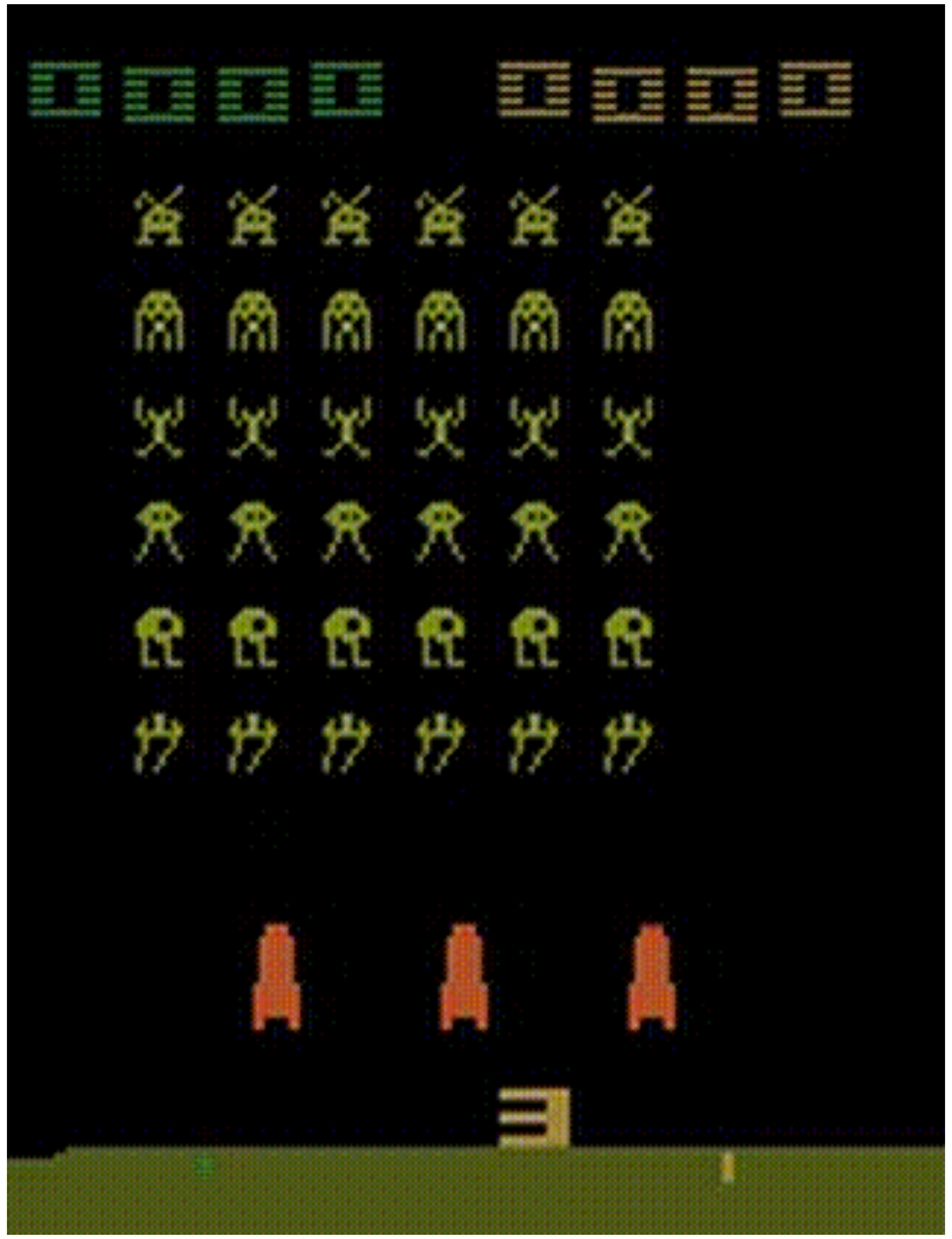
- In addition to Q-network θ, has a target network θ^{-}, a time-delayed copy of Q-network θ (periodically copied from the Q-network)
- Given transition (s, a, r, s') sampled (in minibatches) from buffer
  - $\hat{y} = Q(s, a; \theta)$ (prediction)
  - $y = r + \gamma \text{max}_{a'} Q(s', \text{argmax}_{a'} Q(s', a'; \theta); \theta^{-})$ (target)
  - Minimize $\left(y - \hat{y}\right)^2$

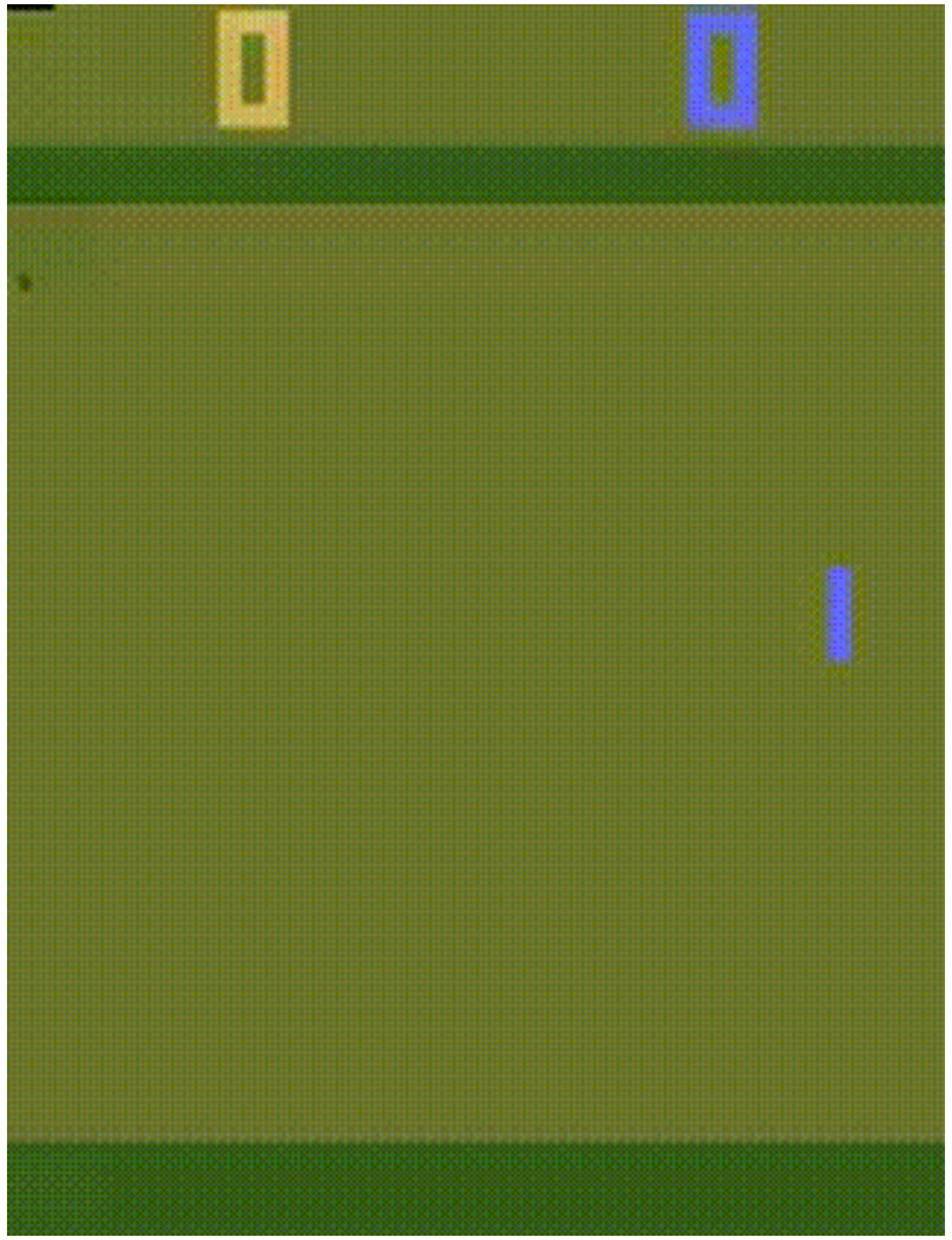
Double DQN Schematic



Testbed: Atari 2600 games







The Tandem Effect [2]

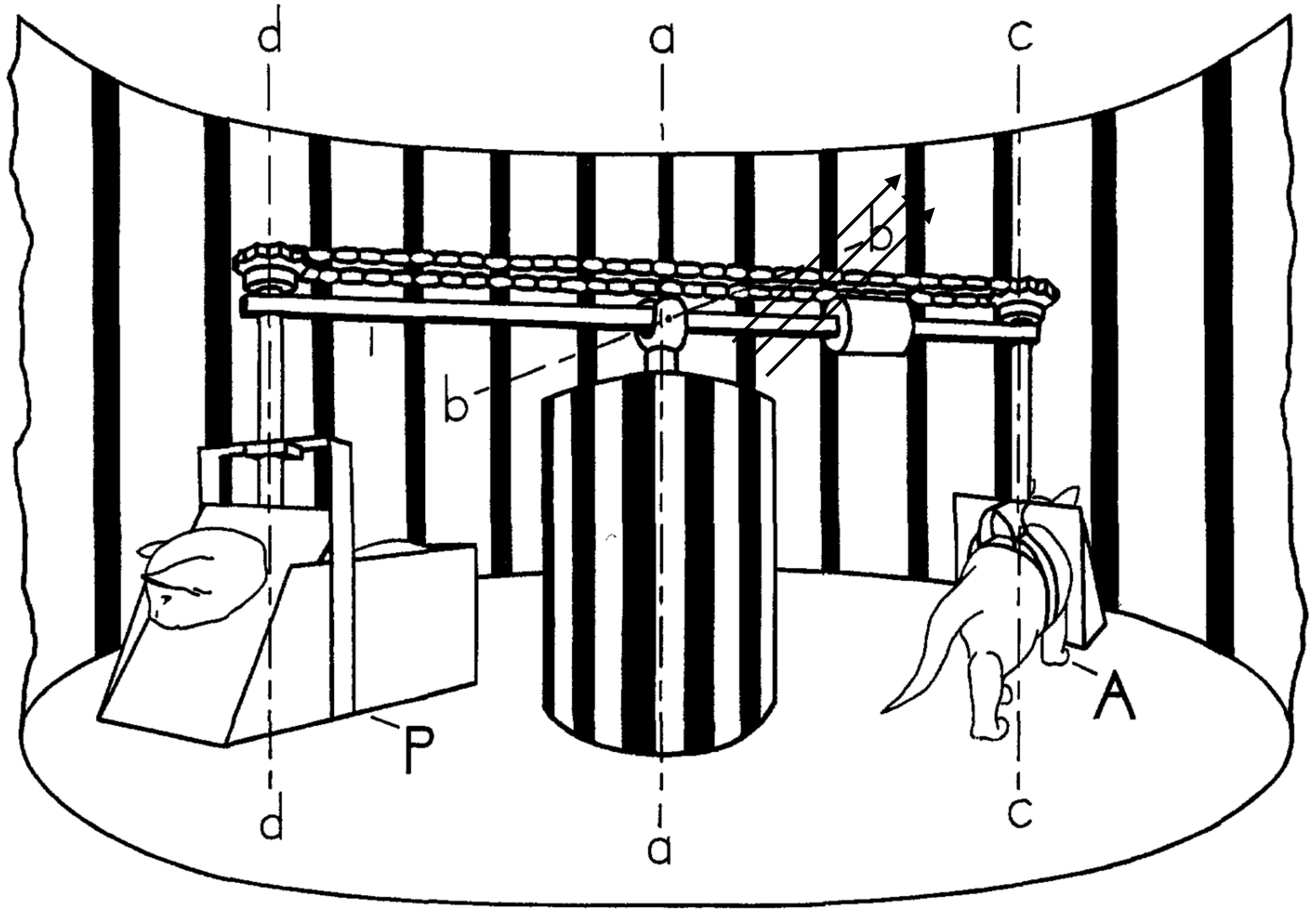
Plots & some figures in this section of the talk taken from Ostrovski et al. (2021).

[2] Ostrovski et al. (2021). The Difficulty of Passive Learning in Deep Reinforcement Learning. NeurIPS.

Biological Motivation

- Thesis: "**self-produced movement** with its concurrent visual feedback is necessary for the development of visually-guided behavior." [3]

Passive kitten



Active kitten

Figure is taken from Held & Hein (1963)

[3] Held & Hein (1963). Movement-produced stimulation in the development of visually guided behavior. Journal of Comparative and Physiological Psychology.

Tandem Effect

- Learning from offline or observational data (without interaction) is challenging (batch RL or offline RL).
- The **Tandem Effect**: Phenomenon where a "*passive learner generally fails to adequately learn from the very data stream that is demonstrably sufficient for its architecturally identical active counterpart*" [2].

[2] Ostrovski et al. (2021). The Difficulty of Passive Learning in Deep Reinforcement Learning. NeurIPS.

The Tandem Setup

- Initialize two Double DQN agents differently
- **Active learner**: interacts with environment and learns from that data
- **Passive learner**: learns from the active learner's data
- Both trained on **same minibatches**; all other details same (architecture, etc.)

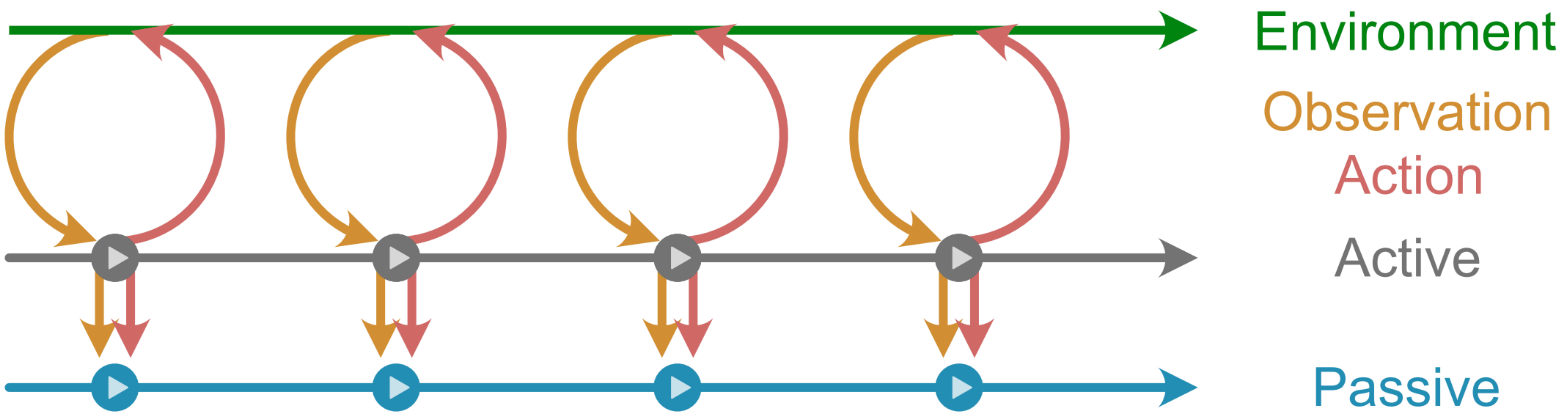
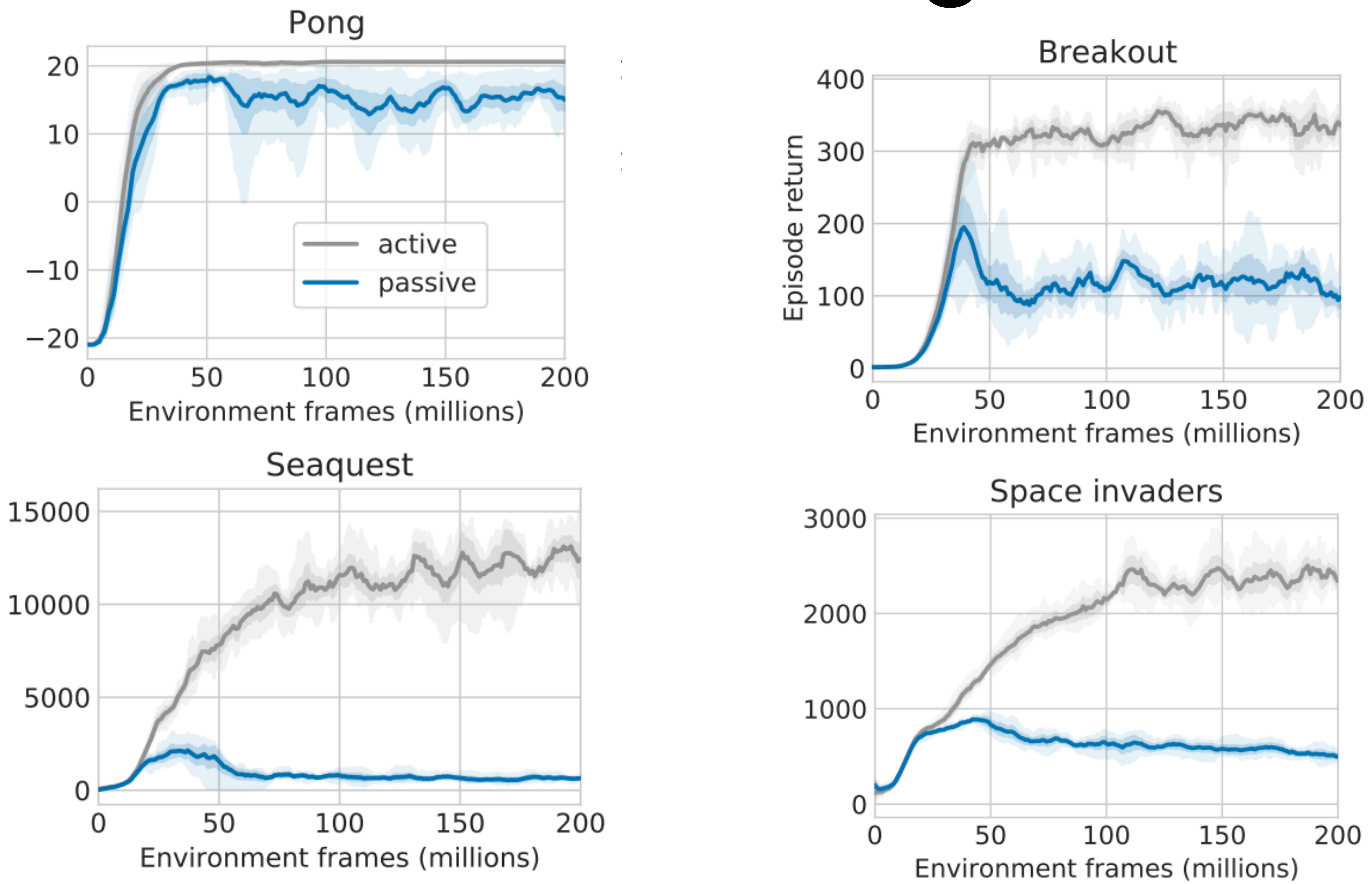


Figure taken from Ostrovski et al. (2021)

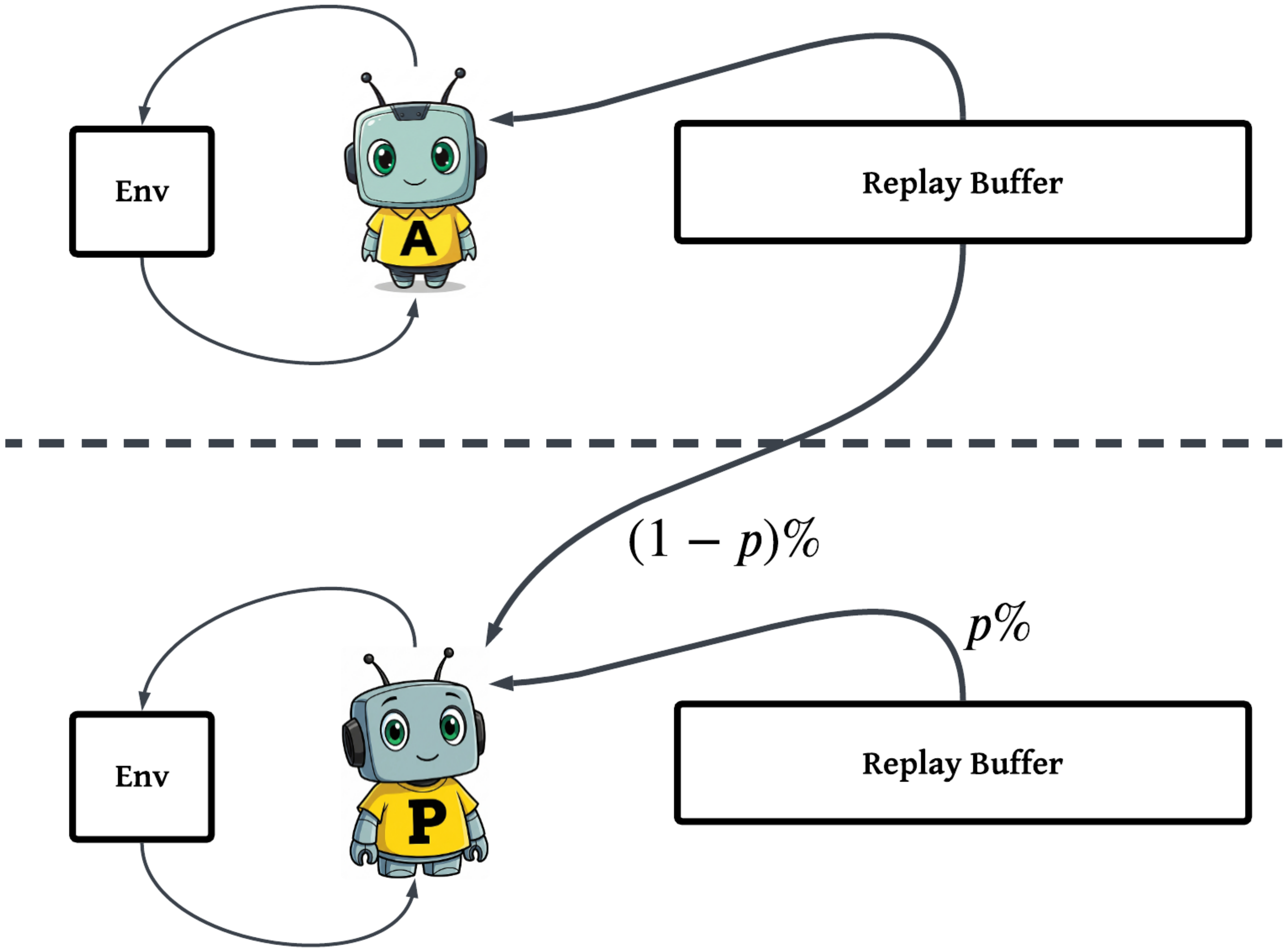
Tandem Learning: Results



Figures taken from Ostrovski et al. (2021)

Mitigating the Tandem Effect: Injecting Active Data

Train 2 agents, With both interacting and filling their own buffer.

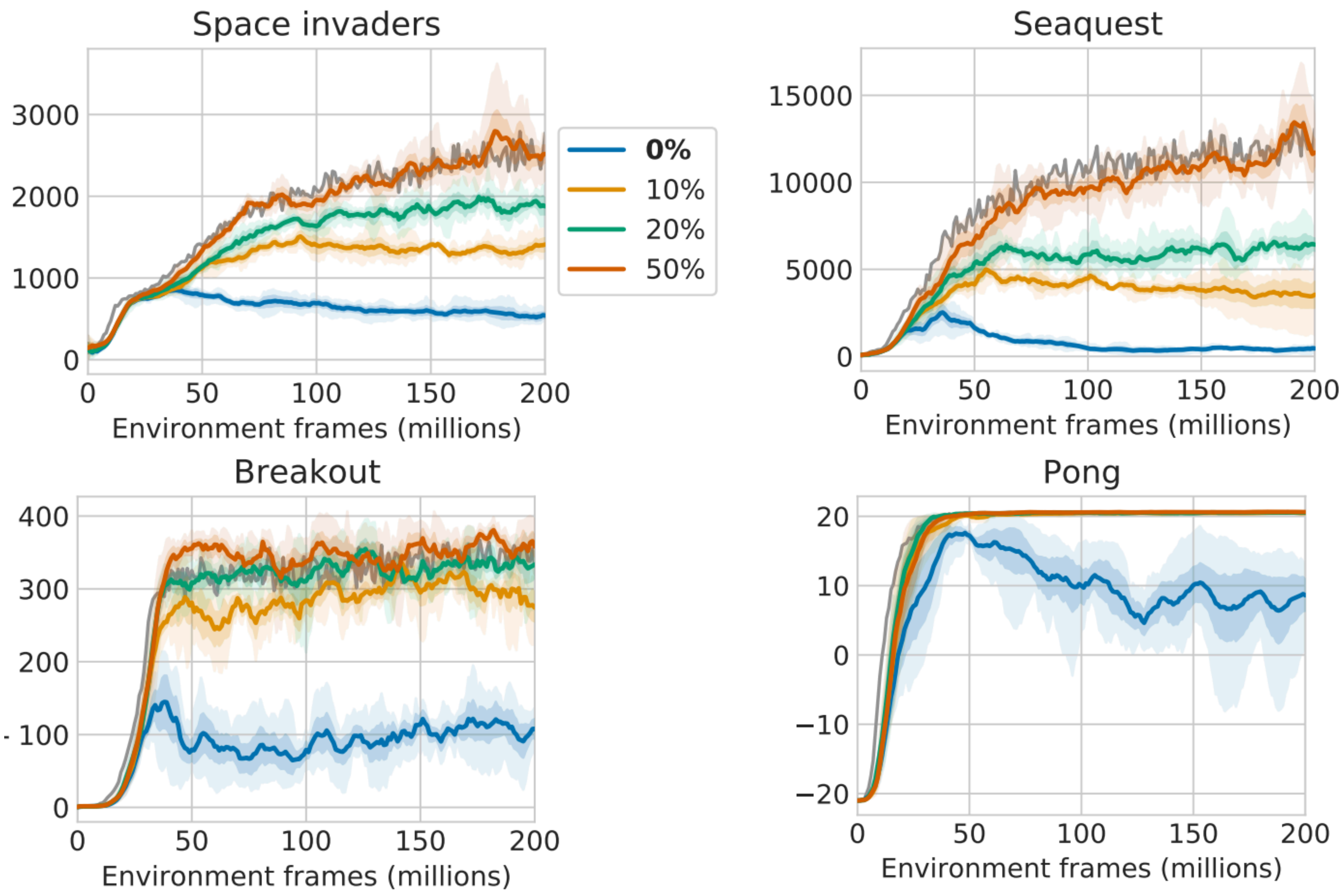


Active agent exclusively uses its own buffer.

Passive agent samples from active buffer, but some percentage from its own buffer

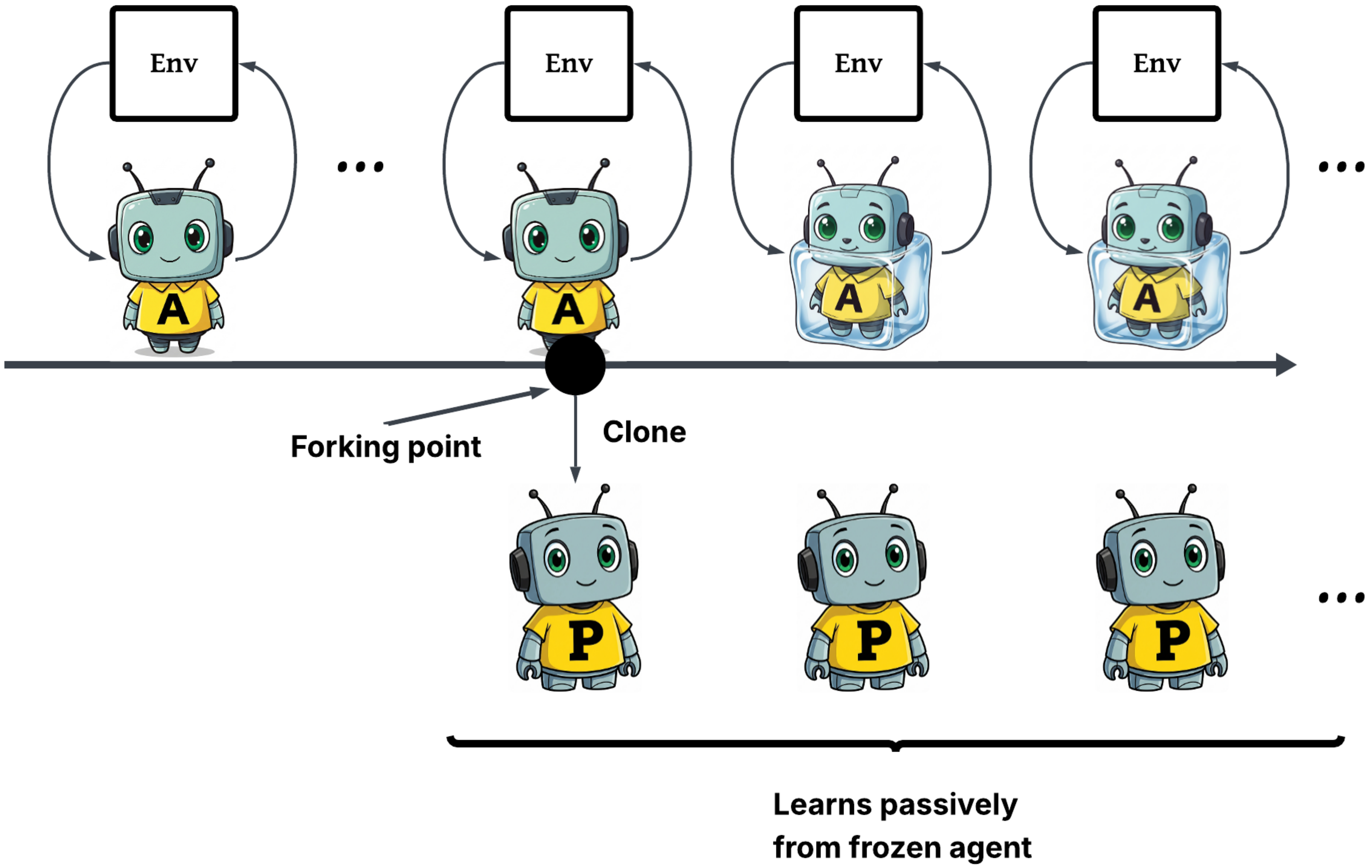
'How much data generated by the passive agent is needed to correct for the tandem effect?' (Ostrovski et al. 2021)

Injecting Active Data: Results

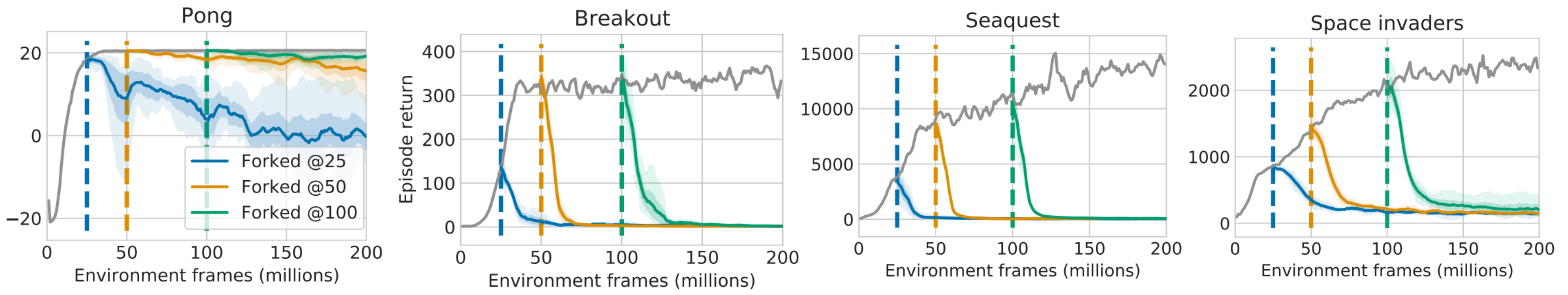


Takeaway: Allowing the passive learner to generate active data can mitigate the tandem effect. **Even a little data can help. And the effect goes away at 50% of active data.**

Forked Tandem Setup



Forked Tandem: Fixed Policy Results



- *Tandem effect in fact gets worse*! Passive learner's performance decays rapidly (except for Pong)
- **Data distribution is important**: The frozen policy fills the buffer with low diversity actions and the passive learner diverges
- Passive learning not only makes it difficult to *learn to act*, but even to *maintain performance*
- Freeze a Double DQN agent's policy and keep learning from that policy, it will diverge: self-correction is key

Tandem Effect: Summary

- The tandem effect is real: a passive learner is much worse than an active learner
    - Generally good data may not be enough
- Having some active data can significantly help a passive learner and may even be necessary
- **The data distribution is important for performance.**
    - **Continual diverse data seems important to prevent divergence**

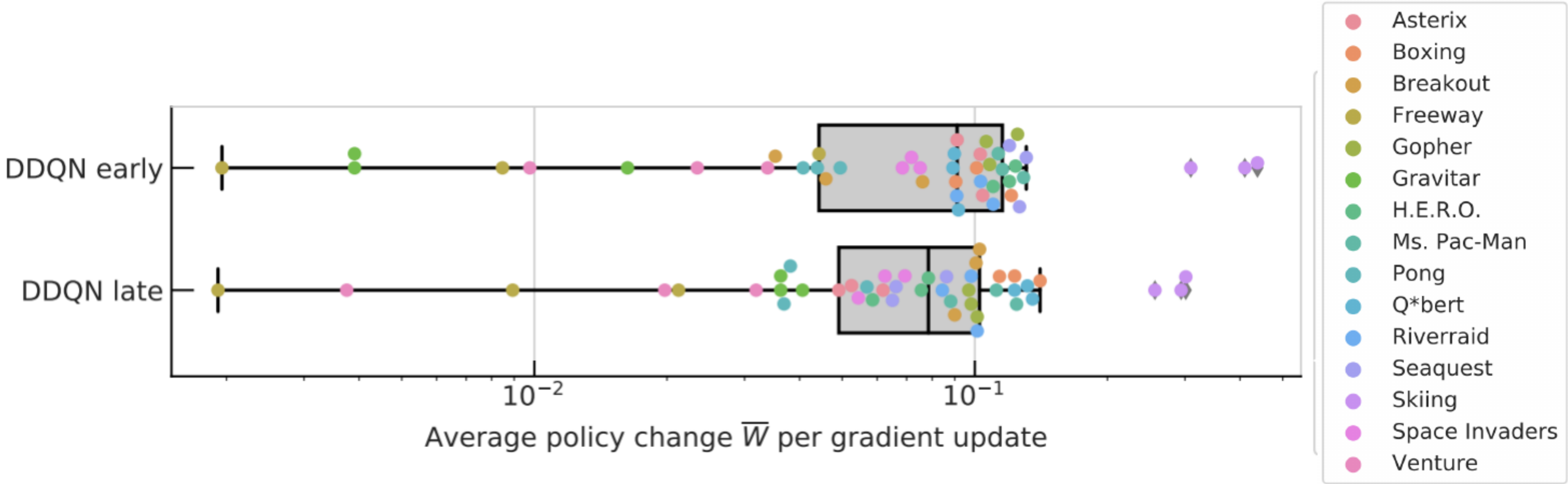
Policy Churn [4]

Plots in this section of the talk taken from Schaul et al. (2022).

[4] Schaul et al. (2022). The Phenomenon of Policy Churn. NeurIPS.

Policy Churn

- **Policy churn** is an empirical phenomenon that refers to "*the rapid change of the greedy policy in value-based reinforcement learning*" [4]
- In Double DQN on Atari 2600 games, the greedy policy changes in *approximately 9% of all states after one gradient update*.

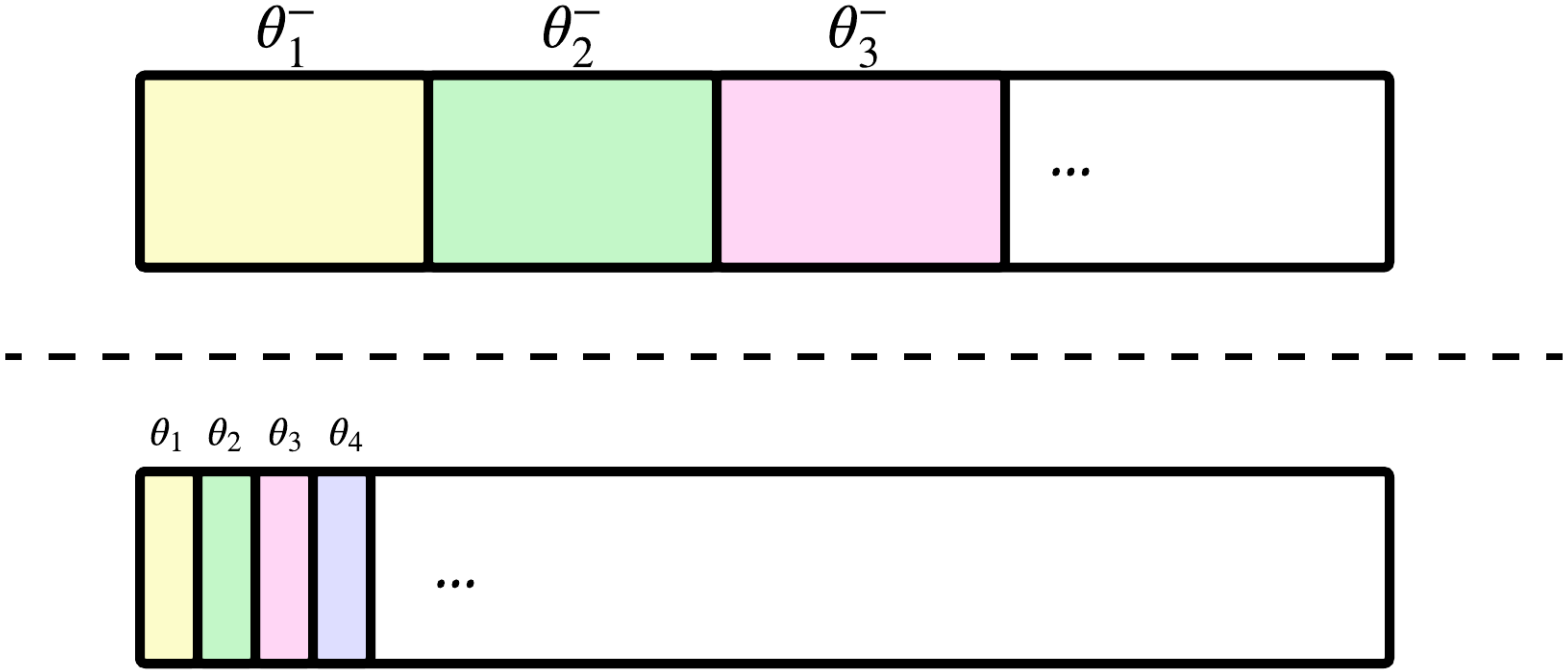


[4] Schaul et al. (2022). The Phenomenon of Policy Churn. NeurIPS.

Policy Churn: Exploration

- **Policy Churn can drive Exploration**
- **Experiment**: Reduce churn's effect on data distribution by *acting with target network*
    - The target network is copied at a slower pace
    - Greedy actions won't change as often
    - If churn helps exploration, should see reduced performance by acting with the target network

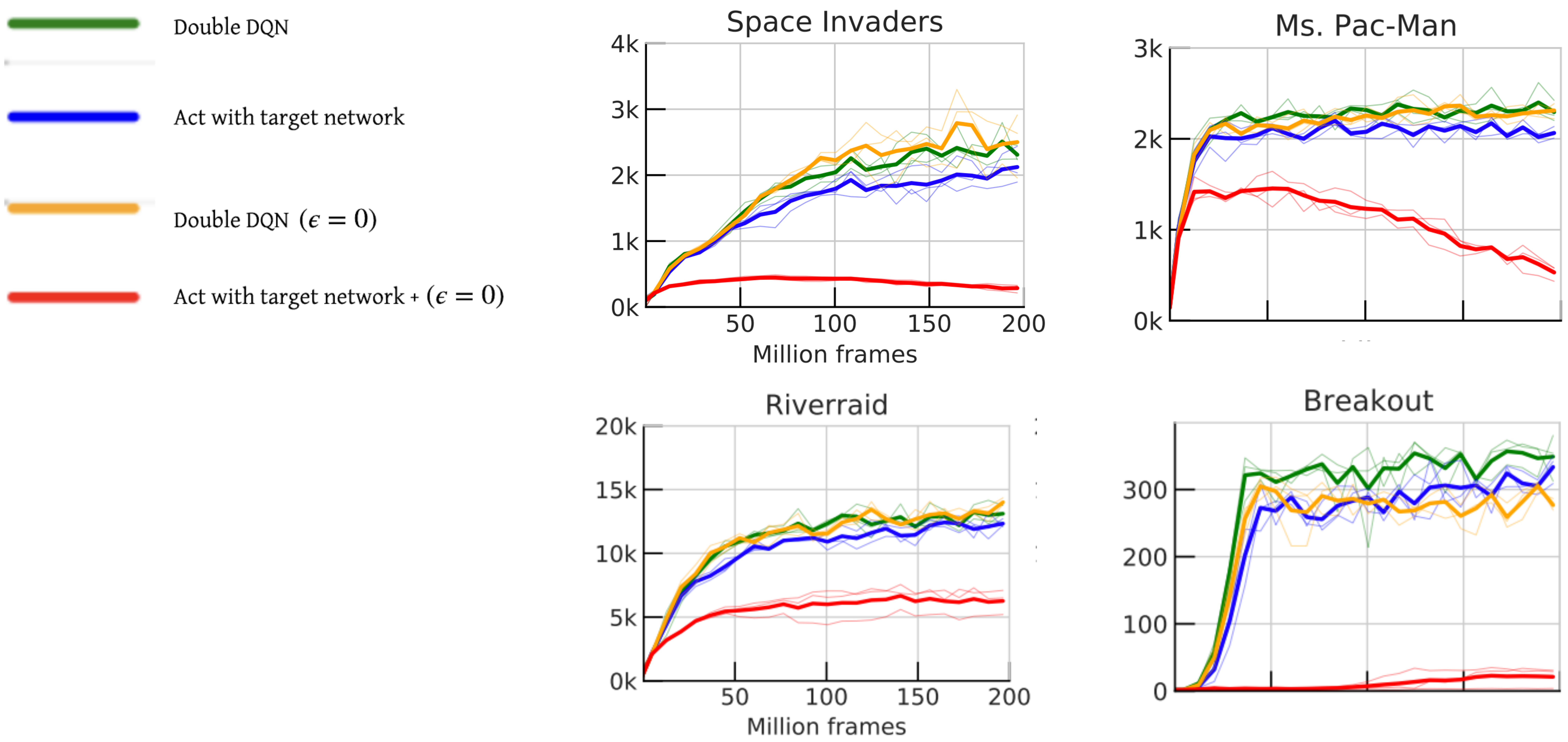
Target Network Experiment



Policy Churn: Exploration

- **Policy Churn can drive Exploration**
- **Experiment**: Reduce churn's effect on data distribution by *acting with target network*
  - The target network is copied at a slower pace
  - Greedy actions won't change as often
  - If churn helps exploration, should see reduced performance
- **Experiment**: Try greedy policy (I.e., $\epsilon = 0$) so only churn drives exploration
  - If churn helps, we should not see too much degradation

Exploration Results



Policy Churn: Causes/Influences

- Redundant actions? ✗
- Small action gaps? ✓
  - Action gap = difference between the largest and second-largest action values
  - Methods that increase the action gap reduce churn
- Non-stationary state/data distribution? ✗
- Non-stationary targets? ✗

Summary and Insights

- **Churn is omnipresent.**
- **Occurrence of churn correlated the most with the presence of function approximation**
- **Schaul et al.'s Hypothesis**: Churn is caused by two necessary components
  - Non-linear, global function approximation (e.g., DNNs)
  - Noisy learning process (e.g., SGD, large learning rate, noisy targets, non-stationary data, etc.)

The Curse of Diversity in Ensemble-Based Exploration [5]

Plots & some figures in this section of the talk are taken from Lin et al. (2024).

[5] Lin et al. (2024). The Curse of Diversity in Ensemble-Based Exploration. ICLR.

The Curse of Diversity in Ensemble Exploration

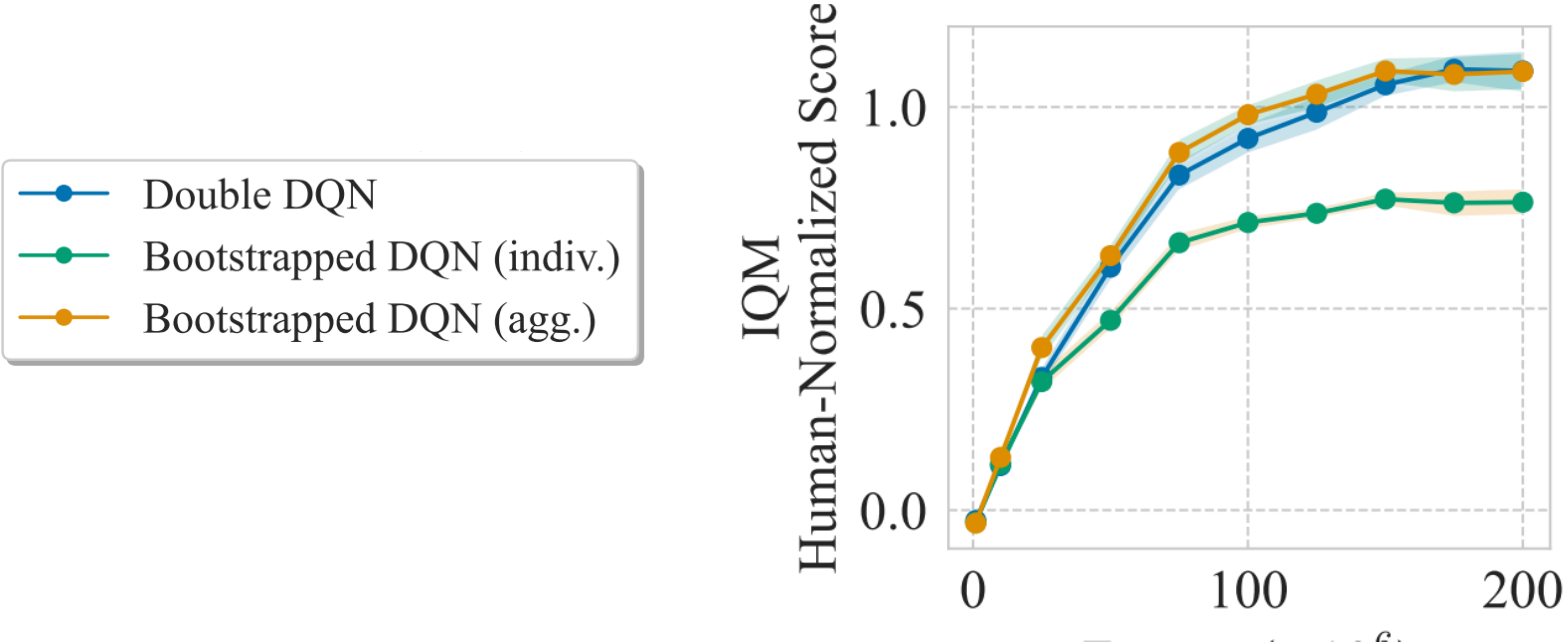
- The **Curse of Diversity** Phenomenon of Ensemble Exploration: "*individual members in a data-sharing ensemble can vastly underperform their single-agent counterparts*" [5].
- The **Tandem Effect**: Phenomenon where a "*passive learner generally fails to adequately learn from the very data stream that is demonstrably sufficient for its architecturally identical active counterpart*"(Ostrovski et al., 2021).
- Perhaps ensemble members are passive off-policy learners of their fellow ensemble members?

[5] Lin et al. (2024). The Curse of Diversity in Ensemble-Based Exploration. ICLR.

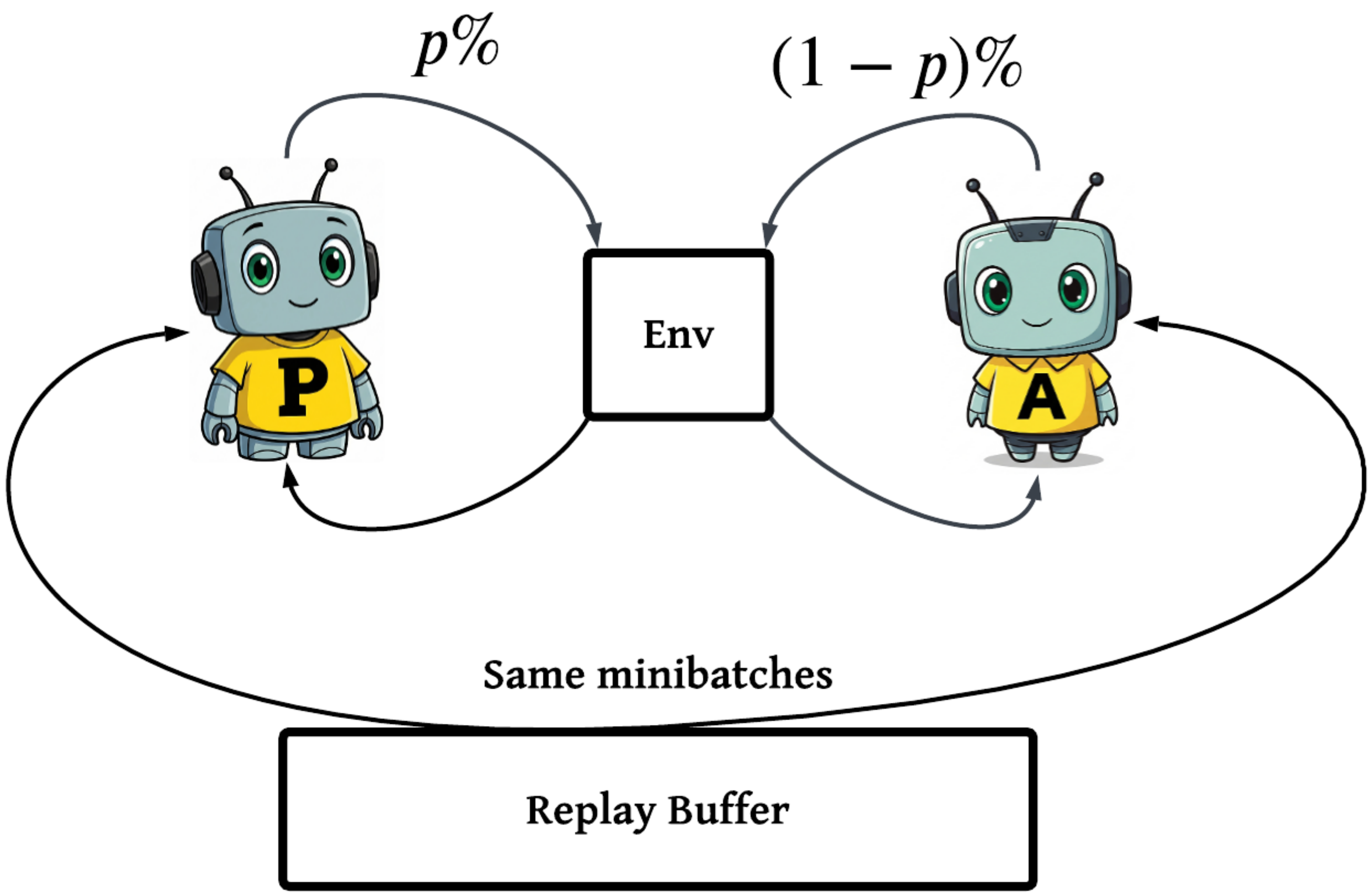
What Constitutes Ensemble-based Exploration

1. Temporally coherent exploration
2. Relative independence between ensemble members
3. Off-policy RL algorithms with a shared replay buffer

Curse of Diversity Demonstrated

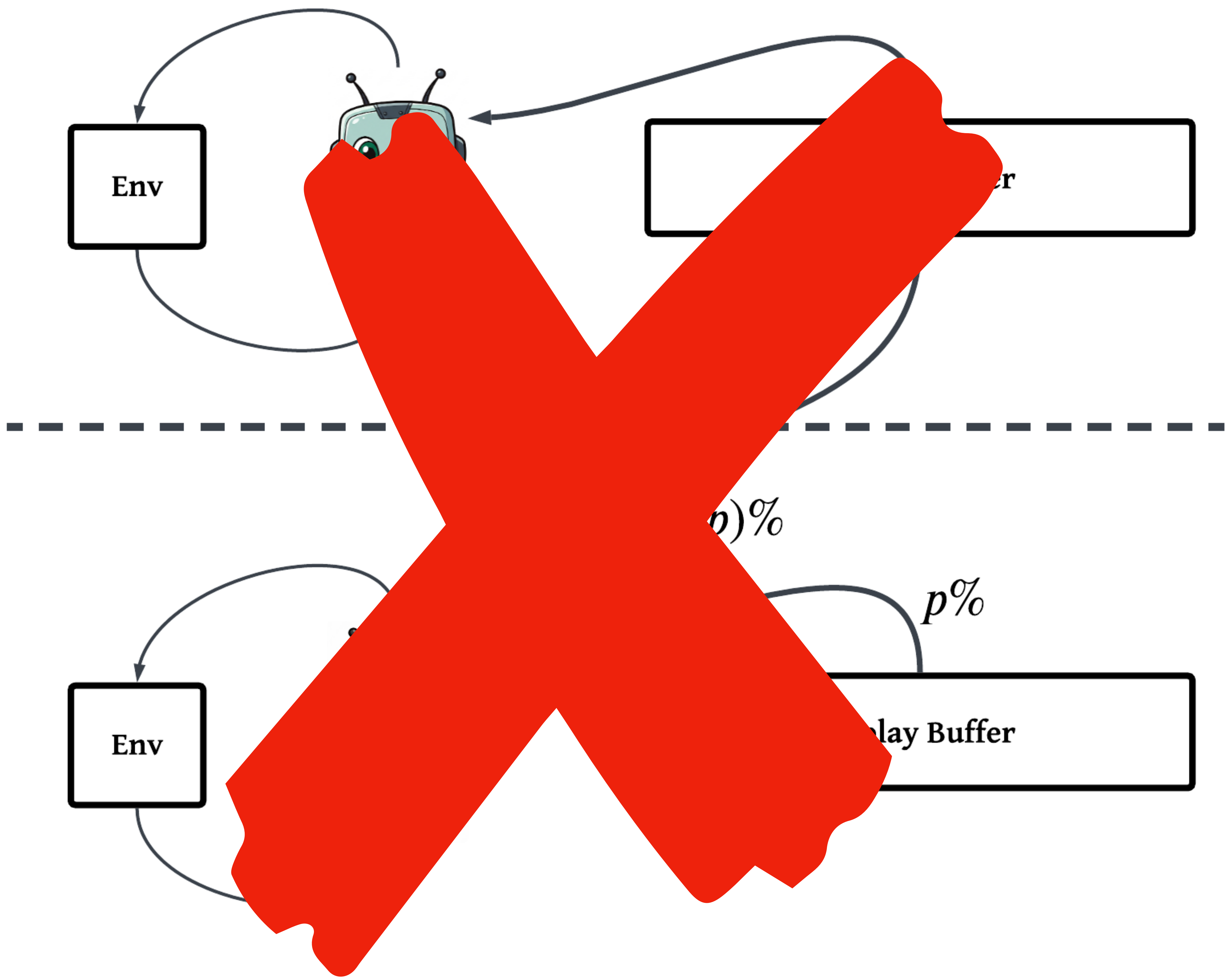


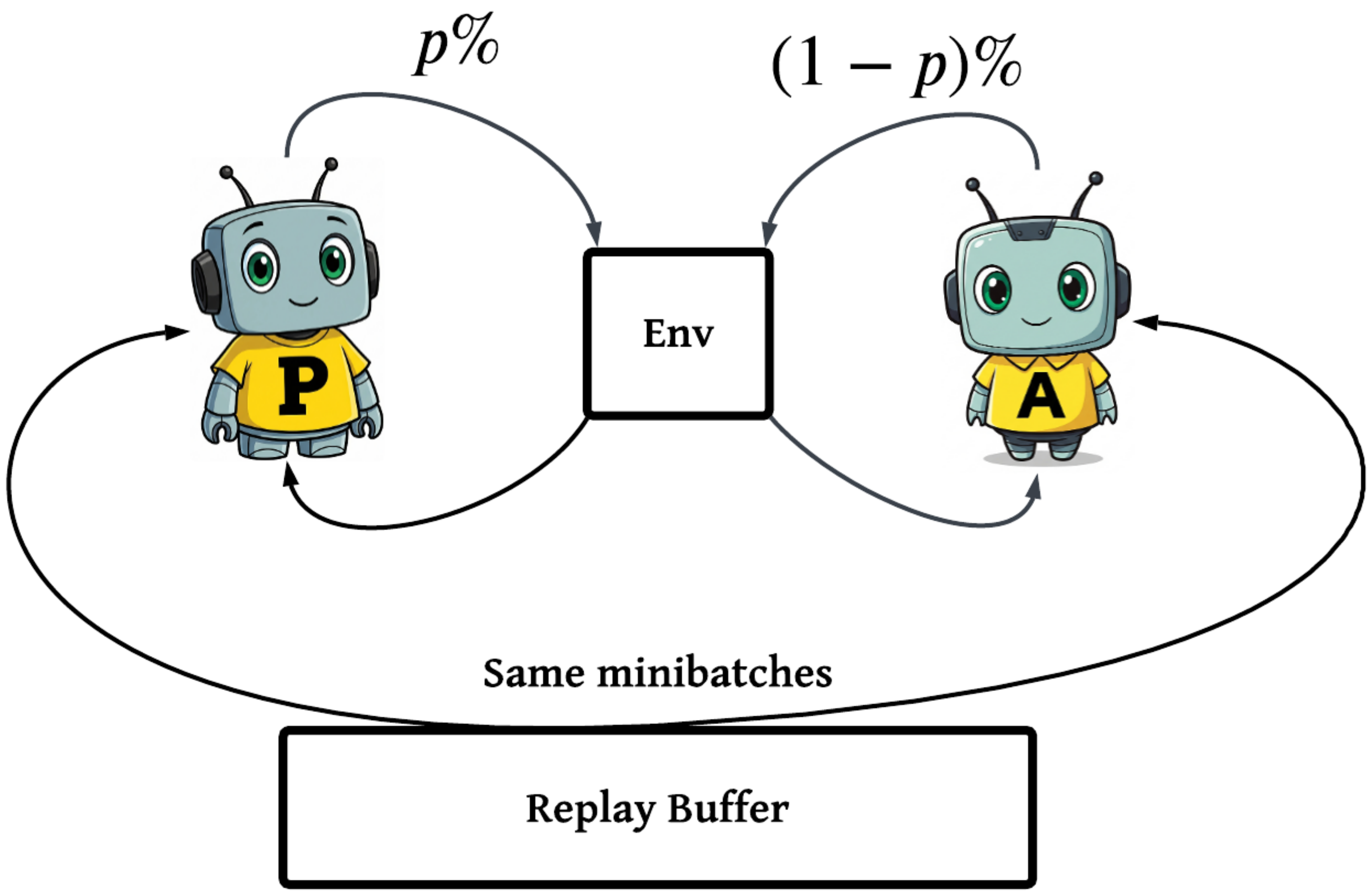
p%-tandem setup



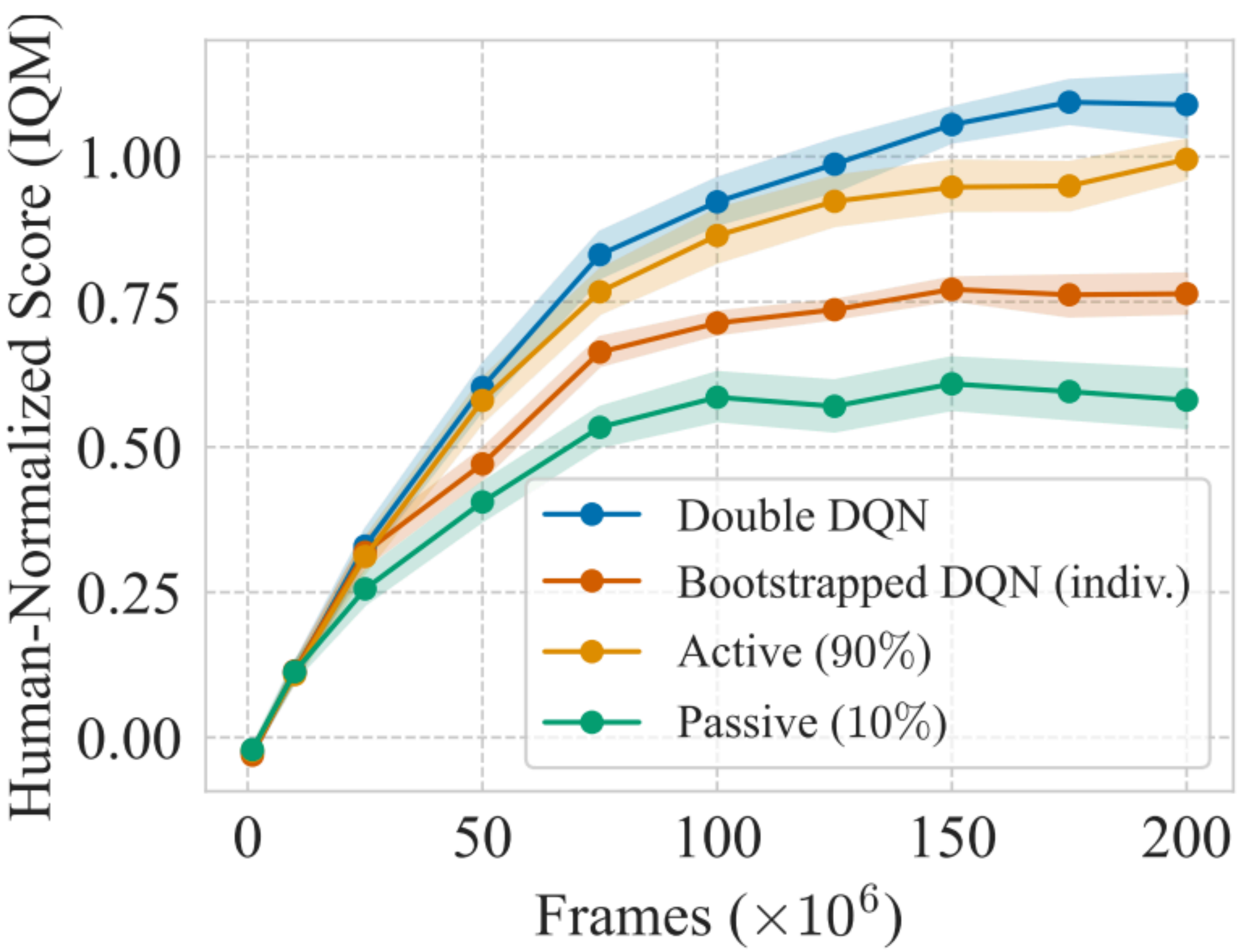
"any performance gap between the active and passive agents can only be due to the difference in the proportions of the two agents' self-generated data and the inefficiency of the passive agent to learn from the shared data." (Lin et al., 2024)

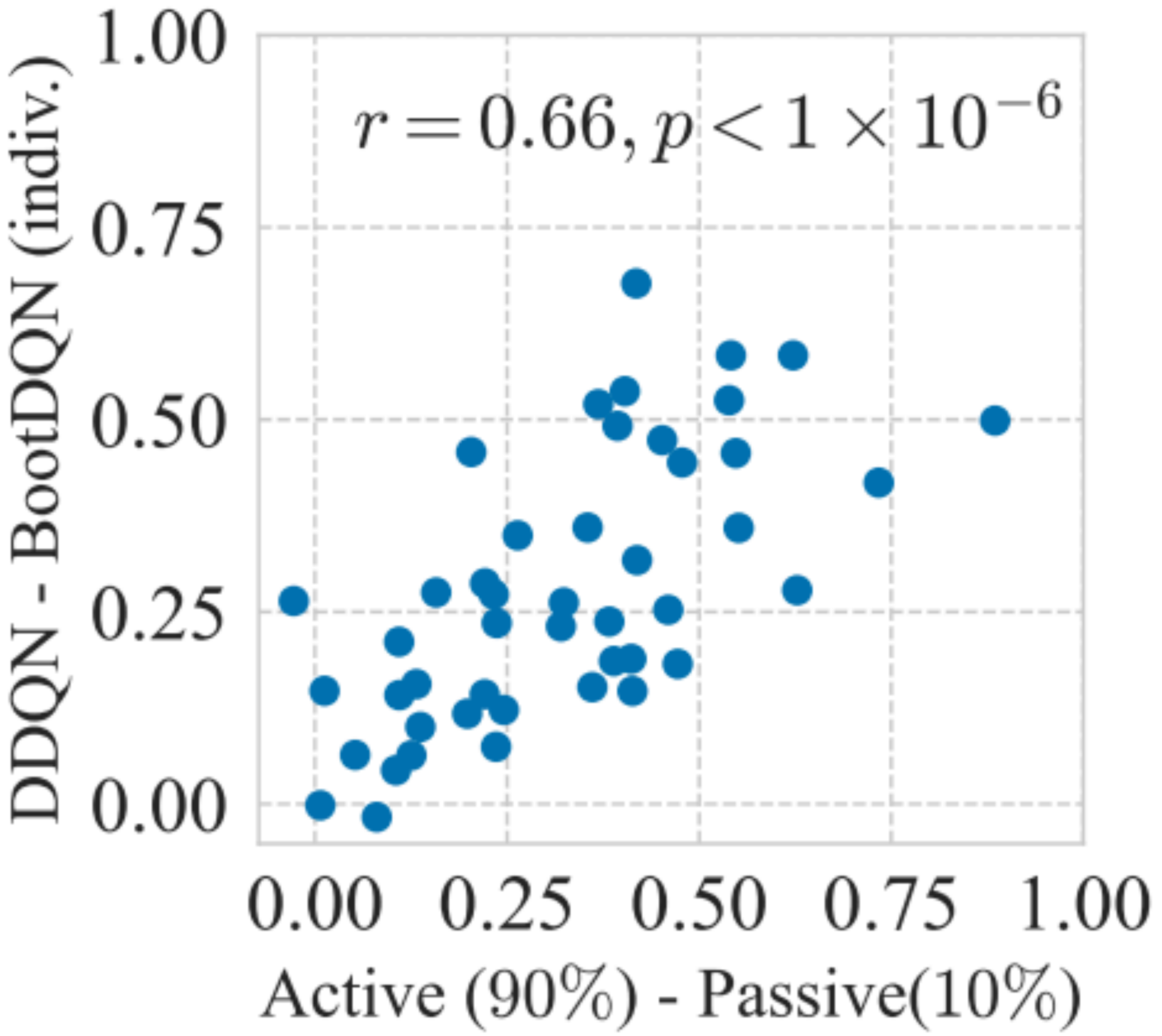
Contrast





p% tandem results





- Conclusions: Curse of Diversity is due to
  - "*The low proportion of self-generated data in the shared training data for each ensemble member*" (Lin et al., 2024)
  - *"The inefficiency of the individual ensemble members to learn from such highly off-policy data" (Lin et al., 2024)*

Conclusion

- Data distribution is important for off-policy value-based RL
- Improved intuition and analysis
- Some light on potential ways to resolve these problems
- PSA: Check out the papers themselves!