Wild Pursuit: survival of the fittest!
Dynamic population and species cooperation

Antoine Debouchage*, Guillaume Levy*, Clement Wang*
*CentraleSupélec, Université Paris-Saclay, 91190 Gif-sur-Yvette

Environment and goal

Simulating **cooperation** among agents of the same species to study **collective behavior**

- **Rectangular 2D field** with borders
- k species with predation relations, $n = \sum_{i=1}^{k} n_i$ specimens move freely and interact
- **Observation**: positional information about all other agents relative to themselves via a $2n$ dimensional vector within [1, 1]
- **Action**: continuous actions represented by 2D vectors within [1, 1], with speed varying depending on the species

Goal

- Establish an environment with realistic predation relations and interactions
- Foster collaboration among agents of the same species for the collective

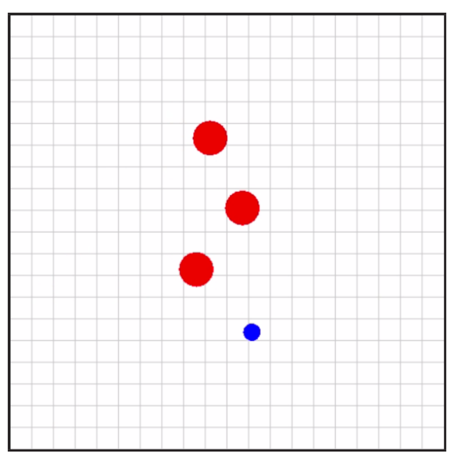


Figure 1: Prey-predator Python environment

Reward for agent a of species p

- *Caught by a predator*: -10 per collision with predators
- *Catching preys*: + 10 if an agent of the same species catch a prey
- *Evading predators*: $\alpha \sum_{b \in \text{preds}_p} \text{dist}(a, b)$
- *Hunting collaboration*: Collective reward based on the minimisation of the distance of all agents of the same species to their closest prey

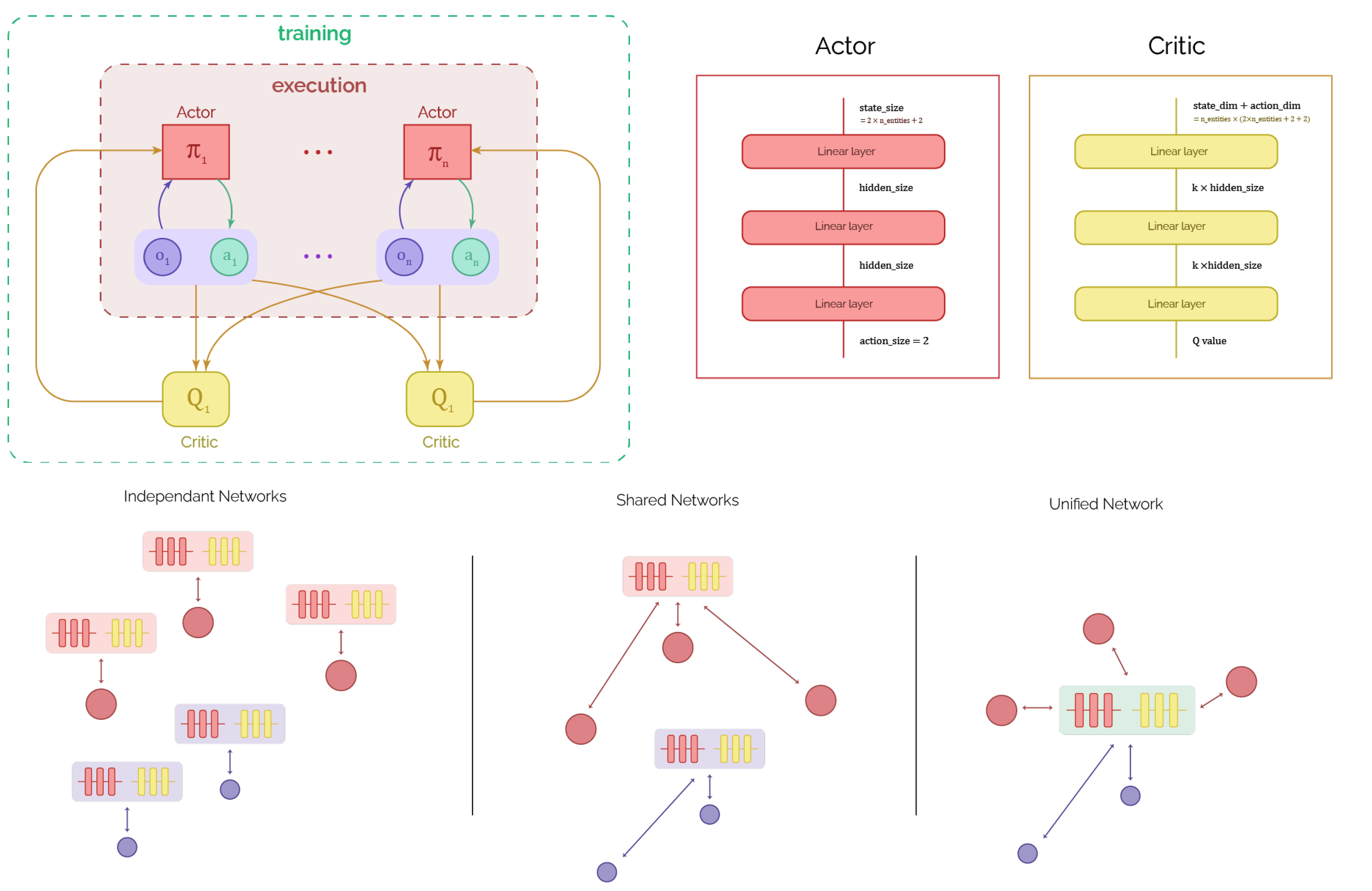
$$-\beta \sum_{a^* \in \text{species}_p} \min_{c \in \text{preys}_p} \text{dist}(a^*, c)$$

- *Border penalization:* $-(\max(0, |x| - 0.9)) * 10 - \max(0, |y| - 0.9)) * 10$

Multi-Agent Deep Deterministic Policy Gradient & Extensions

MADDPG:

- MADDPG [1] extends DDPG [2] to multi-agent settings, enabling learning in cooperative or competitive environments
- It shares information during training but agents make decentralized decisions during execution
- Each agent has a DDPG architecture with a neural network for his policy and his value estimator
- The agent takes action based on his observation but learns a Q function based on everyone's observations and actions



MADDPG pseudo-code

Algorithm 1: Multi-Agent Deep Deterministic Policy Gradient for N agents

for episode $= 1$ to M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial state $\mathbf{x}$
  for $t = 1$ to max-episode-length **do**
    for each agent i, select action $a_i = \boldsymbol{\mu}_{\theta_i}(o_i) + \mathcal{N}_t$ w.r.t. the current policy and exploration
    Execute actions $a = (a_1, \ldots, a_N)$ and observe reward r and new state $\mathbf{x}'$
    Store $(\mathbf{x}, a, r, \mathbf{x}')$ in replay buffer $\mathcal{D}$
    $\mathbf{x} \leftarrow \mathbf{x}'$
    for agent $i = 1$ to N **do**
      Sample a random minibatch of S samples $(\mathbf{x}^j, a^j, r^j, \mathbf{x}'^j)$ from $\mathcal{D}$
      Set $y^j = r_i^j + \gamma Q_i^{\boldsymbol{\mu}'}(\mathbf{x}'^j, a'_1, \ldots, a'_N)|_{a'_k = \boldsymbol{\mu}'_k(o_k^j)}$
      Update critic by minimizing the loss $\mathcal{L}(\theta_i) = \frac{1}{S} \sum_j \left(y^j - Q_i^{\boldsymbol{\mu}}(\mathbf{x}^j, a_1^j, \ldots, a_N^j) \right)^2$
      Update actor using the sampled policy gradient:

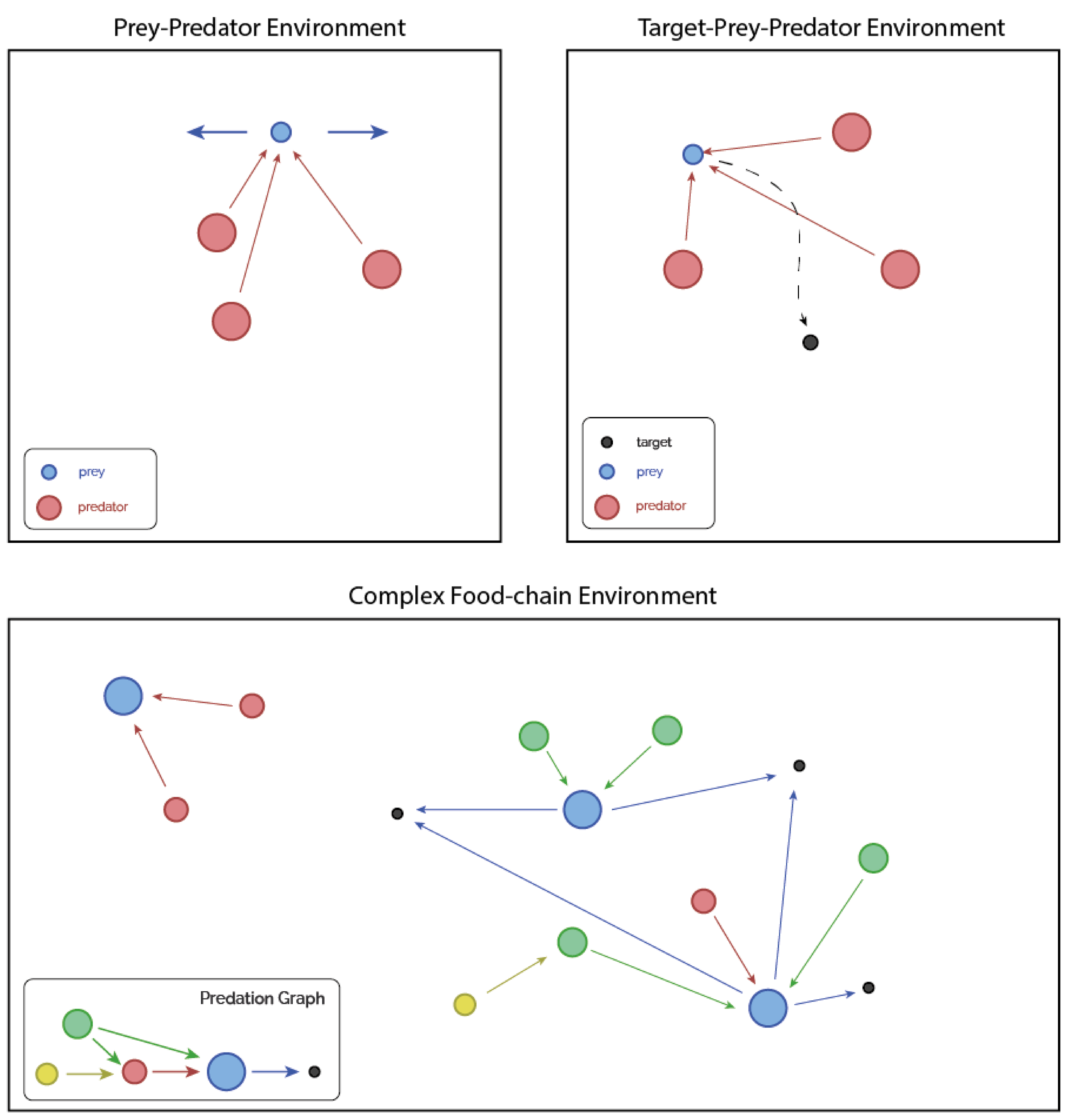
$$\nabla_{\theta_i} J \approx \frac{1}{S} \sum_j \nabla_{\theta_i} \boldsymbol{\mu}_i(o_i^j) \nabla_{a_i} Q_i^{\boldsymbol{\mu}}(\mathbf{x}^j, a_1^j, \ldots, a_i, \ldots, a_N^j)\big|_{a_i = \boldsymbol{\mu}_i(o_i^j)}$$

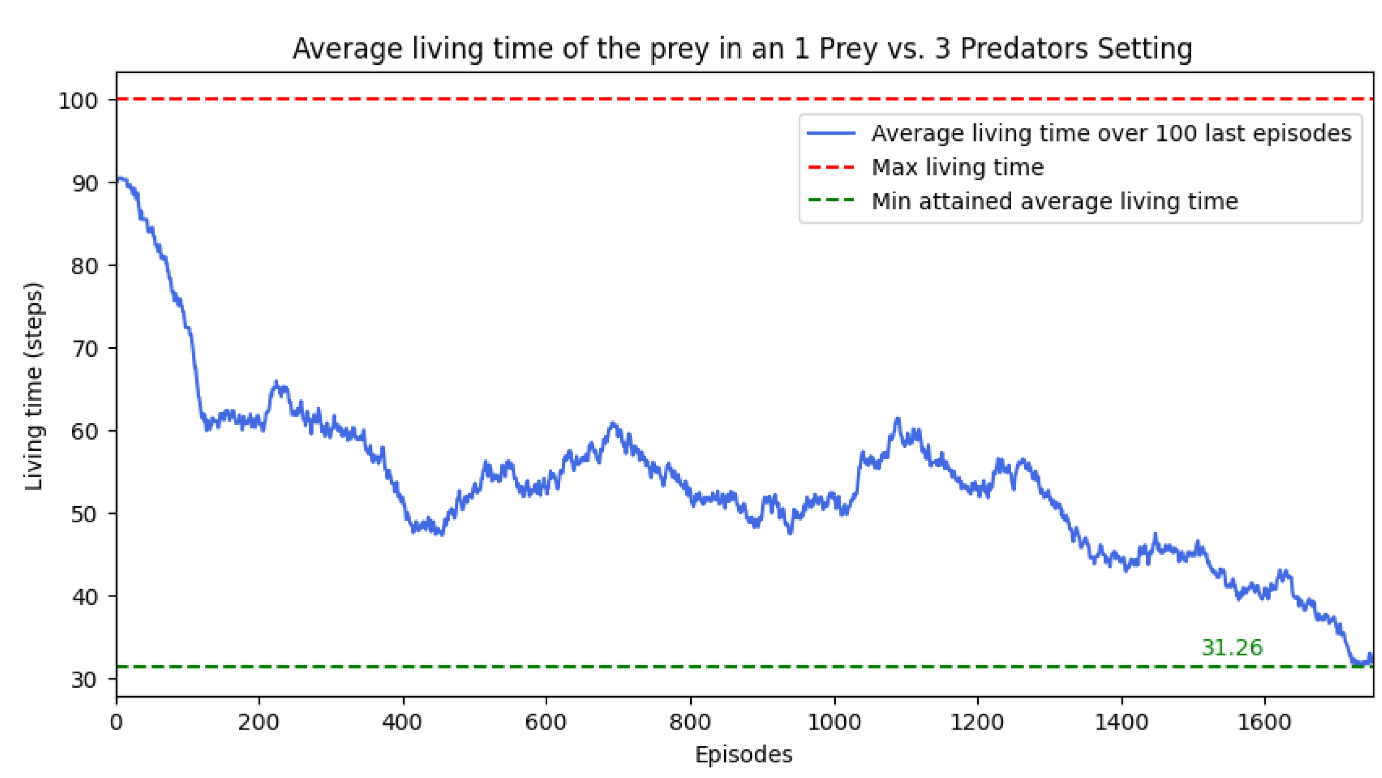
    end for
    Update target network parameters for each agent i:

$$\theta'_i \leftarrow \tau \theta_i + (1 - \tau) \theta'_i$$

  end for
end for

Experiments and results





Conclusion

- **Incite cooperative dynamics** among agents of the same species
- **Shared reward** among agents of the same species enhance collaboration
- **MADDPG & our extensions** have enabled the creation of a multi-agent system for effectively modeling complex population dynamics.

References

[1] Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Pieter Abbeel, O., Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. Advances in neural information processing systems, 30.

[2] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

Check out our Github repository!