REINFORCING THE WORLD'S EDGE: A CONTINUAL LEARNING PROBLEM IN THE MULTI-AGENT-WORLD BOUNDARY

Dane Malenfant
School of Computer Science
McGill University
Mila - The Québec AI Institute
dane.malenfant@mail.mcgill.ca

ABSTRACT

Reusable decision structure survives across episodes in reinforcement learning, but this depends on how the *agent–world boundary* is drawn. In stationary, finite-horizon MDPs, an *invariant core*: the (not-necessarily contiguous) subsequences of state–action pairs shared by all successful trajectories (optionally under a simple abstraction) can be constructed. Under mild goal-conditioned assumptions, it's existence can be proven and explained by how the core captures prototypes that transfer across episodes. When the same task is embedded in a decentralized Markov game and the peer agent is folded into the world, each peer-policy update induces a new MDP; the per-episode invariant core can shrink or vanish, even with small changes to the induced world dynamics, sometimes leaving only the individual task core or just nothing. This policy-induced non-stationarity can be quantified with a variation budget over the induced kernels and rewards, linking boundary drift to loss of invariants. The view that a continual RL problem arises from instability of the agent–world boundary (rather than exogenous task switches) in decentralized MARL suggests future work on preserving, predicting, or otherwise managing boundary drift.

1 INTRODUCTION

Reinforcement learning (RL) formalizes sequential decision making as interaction between an agent and a world (Javed & Sutton, 2024). A modeling choice— the *agent–world boundary*—partitions what adapts inside the agent (state, memory, policy) from external dynamics. In the standard finite-horizon MDP, this boundary appears sharp: a policy π acts on states S and actions A, the world evolves via $P(\cdot \mid s, a)$, and rewards $R(s, a)$ provide feedback; stationarity and the Markov property render this interface time-invariant, and memoryless (Sutton & Barto, 2018).

This apparent precision is a property of the *modeling assumptions*, not of the underlying system or problem. Value functions and guarantees can change with the boundary or representation, motivating boundary-invariant/representation-robust formulations (Jiang et al., 2015) and showing that moving internal dynamics into the agent alters theoretical guarantees such as regret bounds (Jin et al., 2020). The boundary is enacted by the modeler; different framings induce different notions of agency (Abel et al., 2025; Harutyunyan, 2020).

A second subtlety is non-stationarity. In continual RL (CRL), rewards or dynamics shift over time (Khetarpal et al., 2020). In multi-agent RL (MARL), peers' evolving policies induce *effective* dynamics for a focal learner (Littman, 1994; Claus & Boutilier, 1998). Peers may be modeled as stochastic environmental features or as components of a centralized system (Busoniu et al., 2008; Shoham & Leyton-Brown, 2008; Oliehoek & Amato, 2016). In decentralized settings with unobserved peer internals, each peer update changes the induced transition kernel and thus the learning problem (Claus & Boutilier, 1998; Bowling & Veloso, 2002). Consequently, the agent–world boundary itself becomes unstable, and stationarity can fail even at short horizons.

The environment boundary as a continual learning problem was described by Khetarpal et al. who emphasized that non-stationarity should be characterized both by its *scope* (which parts of the interaction process change) and its *driver* (whether change is passive/exogenous, active/agent-influenced, or hybrid) (Khetarpal et al., 2020). Crucially, *learning in the presence of other learning agents* as a prototypical CRL regime: an (active) Markov game can be stationary at the joint level, while a single learner experiences non-stationary effective rewards and transitions as peers update their policies (Khetarpal et al., 2020; Kim et al., 2022). Equivalently, this can be viewed as partial observability, where unobserved peer policies (or learning states) act as a latent task variable that must be inferred online (Khetarpal et al., 2020).

Contributions.

1. Stationary, finite-horizon MDP tasks are formalized as decision tries over state–action trajectories and use this view to reason about shared structure among successes.
2. An *invariant core*: the set of $\preccurlyeq$-maximal subsequences common to all successful trajectories (optionally under a task-appropriate abstraction) is defined, and existence proven under mild goal-conditioned assumptions.
3. Decentralized MARL is shown that by folding peers into the world yields a drifting sequence of induced MDPs as peer policies change, so episode-wise invariant cores can lose prototypes or motifs across episodes.
4. This *vanishing* is argued as continual learning driven endogenously by boundary drift (not an exogenous task schedule), explaining when transfer fails between episodes.
5. Drift via a variation budget is quantified over the induced MDP sequence, connecting stability of reuse to boundary instability.

Assumptions are explicitly stated so claims about existence and stability of the core can be verifiable within standard RL theory but sketches are provided to motivate intuition for a general reader.

2 The agent–world boundary drifts as policies update over time

RL begins with a modeling choice: an agent–world boundary that determines what adapts inside the agent and what is treated as fixed dynamics. In single-agent, stationary MDPs this boundary is fixed, and successful episodes reuse common decision structure; in particular, certain subsequences of state–action pairs are shared by all successful trajectories. We formalize these shared prototypes or motifs as elements of an *invariant core* set. By contrast, in decentralized two-agent Markov games, the other agent induces world-dynamics that depend on that agent's policy; as they update, the effective MDP drifts across episodes and reusable prototypes that were reusable can disappear.

This endogenously changing agent–world boundary *poses* a continual-learning problem: stability of learned structure is not only a function of exogenous task switches but also of how the boundary is drawn because peer agents are adapting to change.

2.1 The boundary is stable in single-agent tasks

Let $M = (S, A, P, R, H, G)$ be a finite-horizon, goal-conditioned MDP with horizon H and goal set $G \subseteq S$. Episodes terminate on first visit to G. A (state–action) trajectory is $\tau = (s_1, a_1, \ldots, s_T, a_T)$ with $T \leq H$. Define the set of successful trajectories

$$\mathcal{S} \;=\; \big\{\tau :\; \exists\, t \leq H \text{ with } s_t \in G\big\}.$$

For sequences u, v over $S \times A$, write $u \preccurlyeq v$ if u is a (not-necessarily contiguous) subsequence of v.

Trajectory trie representation Let Θ be any multiset of trajectories (e.g., a dataset of rollouts). The *trajectory tree* $\mathcal{T}(\Theta)$ is the trie over the alphabet $S \times A$ whose nodes are prefixes $u \in (S \times A)^{\leq H}$ that appear in some $\tau \in \Theta$; the root is the empty prefix, and each edge appends one pair (s_t, a_t). We label a leaf (or any prefix) with a success indicator $y(u) \in \{0, 1\}$ equal to 1 if some extension of u reaches G within H and 0 otherwise. This view is purely representational; the key object for us is the set $\mathcal{S}$.

Invariant core To capture reusable prototypes, we define the core as the set of $\preccurlyeq$-maximal subsequences shared by all successful trajectories. Because exact prototypes may be semantically clearer after aggregation (e.g., options), we allow an optional task-specific abstraction $\phi : S \times A \to \Sigma$ (Dean & Givan, 1997; Li et al., 2006; Abel et al., 2016) and write

$$\mathrm{Core}_\phi(\mathcal{S}) \;=\; \max_{\preccurlyeq} \Big\{\, u \in \Sigma^{\le H} :\; \forall \tau \in \mathcal{S},\; u \preccurlyeq \phi(\tau) \,\Big\},$$

with $\mathrm{Core}(\mathcal{S})$ denoting the identity-abstraction case.

Theorem 2.1 (Existence)**.** *If $G = \{g\}$ is a unique absorbing goal and episodes terminate on first visit to g, then $\mathrm{Core}(\mathcal{S}) \neq \emptyset$. More generally, if there exists an abstraction ϕ such that every $\tau \in \mathcal{S}$ contains a common abstract symbol (e.g., an option such as* open_door*), then $\mathrm{Core}_\phi(\mathcal{S}) \neq \emptyset$.*

Sketch. $\mathcal{S}$ is written for the set of successful state–action trajectories of length at most H. Under a unique absorbing goal g, every $\tau \in \mathcal{S}$ visits g at some time $t \le H$, so all sequences in $\mathcal{S}$ share at least one common symbol and hence admit a nonempty common subsequence. Because $H < \infty$, there are finitely many subsequences drawn from $\mathcal{S}$, so $\preccurlyeq$-maximal common subsequences exist; any longest common subsequence (LCS) of $\mathcal{S}$ is such a maximal element and therefore belongs to the core. The same argument holds in the abstract alphabet Σ whenever a common abstract symbol is guaranteed by ϕ. □

In practice one would observe a set of trajectory rollouts Θ; the trajectory trie $\mathcal{T}(\Theta)$ (a prefix tree over $S \times A$) provides a convenient way to enumerate successful leaves and search for common subsequences among them. Computing an exact LCS scales as $O(H^2)$ for two sequences and $O(H^k)$ by naive dynamic programming for k sequences, with the generalized problem NP-hard when k is part of the input. This computational profile motivates using an abstraction ϕ (e.g., options/skills) to reduce the alphabet and isolate shorter, semantically meaningful prototypes (Sutton et al., 1999; Konidaris & Barto, 2009). Classical methods could reduce complexity as well (Hunt & Szymanski, 1977). In canonical key–door tasks (Chevalier-Boisvert et al., 2018; Hung et al., 2019; Sun et al., 2023), for example, every successful trajectory contains the abstract pattern

$$\texttt{find\_key} \;\to\; \texttt{reach\_door} \;\to\; \texttt{open\_door},$$

which thus appears in $\mathrm{Core}_\phi(\mathcal{S})$ and can be implemented as reusable options across episodes while the agent–world boundary remains stationary.

Now a policy π_1 can be considered used to collect trajectories, and let Θ_1 denote the resulting trajectory set and the trajectory tree as $\mathcal{T}_1$. The $\mathrm{Core}_\phi(\mathcal{S})_1$ is the core computed from the successful leaves of $\mathcal{T}_1$. In a stationary MDP the environment (P, R) is exogenous and does not depend on the agent's policy; changing the policy may change preferences of trajectories over others but does not alter which trajectories are successful. Hence, if $\mathcal{T}_1$ is *complete* in the sense that its successful leaves enumerate all successful trajectories of M, the resulting core depends only on (M, G, ϕ) and not on the policy used to gather the trajectory. In particular, for any other policy π_2 with a complete trie $\mathcal{T}_2$ we have $\mathrm{Core}_\phi(\mathcal{S})_1 = \mathrm{Core}_\phi(\mathcal{S})_2$. Operationally, querying a complete core results in a goal-reaching behaviour that remains valid across policy updates; under the standard terminal-reward objective, executing such a process attains the optimal value. This policy-independence of (P, R) is precisely why the core is *invariant* in the single-agent, stationary setting and follows directly from a stable agent–world boundary: the policy π_1 lies on the agent side while (P, R) lie on the world side and, in the stationary single-agent case, are therefore invariant to π_1.

2.2 The agent–world boundary shifts with another agent

Now, the same task can be extended to a two-player decentralized Markov game $\mathcal{G} = (S, A_1, A_2, P, R_1, H, G)$ (Littman, 1994). In episode e, agent 2 follows a policy $\pi_2^e(\cdot \mid s)$ that is unknown to the focal agent. From the focal agent's view, the environment is a single-agent MDP

$$P_e(s' \mid s, a_1) \;=\; \sum_{a_2 \in A_2} P(s' \mid s, a_1, a_2)\, \pi_2^e(a_2 \mid s), \qquad R_e(s, a_1) \;=\; \sum_{a_2 \in A_2} R_1(s, a_1, a_2)\, \pi_2^e(a_2 \mid s),$$

so acting in the game at episode e is equivalent to acting in $M_e = (S, A_e, P_e, R_e, H, G)$ (Oliehoek & Amato, 2016; Busoniu et al., 2008). The agent–world boundary thus encloses an adaptive peer; as π_2^e changes across episodes, the induced dynamics P_e (and possibly R_e) drift.

Let $\mathcal{S}_e$ be the set of successful trajectories in M_e and define the episode-wise core

$$\mathrm{Core}_\phi(\mathcal{S}_e) \;=\; \max_{\preccurlyeq} \Big\{ u \in \Sigma^{\le H} \,:\, \forall \tau \in \mathcal{S}_e,\; u \preccurlyeq \phi(\tau) \Big\}.$$

Under the same mild conditions as Theorem 2.1 (unique absorbing goal or a common abstract symbol), each $\mathrm{Core}_\phi(\mathcal{S}_e)$ exists; however, nothing guarantees stability *across* episodes.

Proposition 2.1 (Episode-to-episode core drift)**.** *There exist Markov games and peer policy updates $\pi_2^e \to \pi_2^{e+1}$ such that a prototype $u \in \mathrm{Core}_\phi(\mathcal{S}_e)$ is not in $\mathrm{Core}_\phi(\mathcal{S}_{e+1})$. Moreover, for suitable tasks one can have $\mathrm{Core}_\phi(\mathcal{S}_e) \cap \mathrm{Core}_\phi(\mathcal{S}_{e+1}) = \varnothing$ after removing the trivial terminal symbol.*

Sketch Consider an episode e. Let $u \in \mathrm{Core}_\phi(\mathcal{S}_e)$, so $u \preccurlyeq \phi(\tau)$ for every successful sequence $\tau \in \mathcal{S}_e$. The task $\mathcal{G} = (S, A_1, A_2, P, R_1, H, G)$ is unchanged; only the peer's policy updates from π_2^e to π_2^{e+1}, thereby changing the set of successful sequences from $\mathcal{S}_e$ to $\mathcal{S}_{e+1}$. If the update admits any success $\tilde{\tau} \in \mathcal{S}_{e+1}$ with $u \not\preccurlyeq \phi(\tilde{\tau})$ (e.g., the peer resolves an individual subgoal differently so the focal agent reaches g without executing u), then by definition $u \notin \mathrm{Core}_\phi(\mathcal{S}_{e+1})$. Thus $\mathrm{Core}_\phi(\mathcal{S}_e)$ and $\mathrm{Core}_\phi(\mathcal{S}_{e+1})$ can differ, even if the underlying task is fixed, purely due to the peer's policy change. Therefore a piece of the core can vanish between episodes, leaving only the policy-independent individual task core or, after removing the trivial terminal symbol, nothing:

$\mathrm{Core}_\phi(\mathcal{S}_e) \cap \mathrm{Core}_\phi(\mathcal{S}_{e+1}) \;\subseteq\; \mathrm{Core}_{\text{individual}}$ and possibly just $\varnothing$ □

Intuitively, because the peer is part of the world, its policy π_2^e determines which subgoals and partial plans are feasible, thereby changing the set of successful trajectories $\mathcal{S}_e$ and the prototypes shared across them. Although each per-episode core $\mathrm{Core}_\phi(\mathcal{S}_e)$ exists, prototypes that were universal at episode e need not persist at $e{+}1$. For example, in a cooperative key–door variant (Malenfant & Richards, 2025), if success at episode e requires the prototype

`drop_key_for_peer` → `peer_agent_reaches_door` → `peer_agent_opens_door`

but after the updating the peer acquires the key independently, that prototype is absent from all successes at $e{+}1$. Thus episode-wise invariant cores need not agree: the overlap $\mathrm{Core}_\phi(\mathcal{S}_e) \cap \mathrm{Core}_\phi(\mathcal{S}_{e+1})$ reduces to at most the policy-independent individual task core (or even completely empty). This is similar to multi-agent experience replay (Foerster et al., 2017). A variation budget quantifies this drift over the induced sequence $\{M_e\}$ for transfer stability across episodes.

2.3 A VARIATION BUDGET FROM SHIFTING MDPS CAN MEASURE THIS CHANGE

To quantify drift across episodes, define

$$V_E \;=\; \sum_{e=2}^{E} \Big(\sup_{s,a_1} \sum_{s'} \big| P_e(s' \mid s, a_1) - P_{e-1}(s' \mid s, a_1) \big| \;+\; \sup_{s,a_1} \big| R_e(s, a_1) - R_{e-1}(s, a_1) \big| \Big).$$

Equivalently, $V_E = \sum_{e=2}^{E} \big(\|P_e - P_{e-1}\|_{1,\infty} + \|R_e - R_{e-1}\|_\infty \big)$, where $\|P\|_{1,\infty} := \sup_{s,a_1} \sum_{s'} |P(s' \mid s, a_1)|$. By construction, $V_E = 0$ iff (P_e, R_e) are stationary, implying $\mathcal{S}_e = \mathcal{S}_{e-1}$ and hence $\mathrm{Core}_\phi(\mathcal{S}_e) = \mathrm{Core}_\phi(\mathcal{S}_{e-1})$. Any peer-policy update that changes (P, R) on some (s, a_1) contributes positively to V_E; when this change adds or removes successful sequences, a prototype can vanish, leaving at most the policy-independent individual task core (or even $\varnothing$). This is the standard drifting-MDP measure (Even-Dar et al., 2009; Cheung et al., 2020; Mao et al., 2021) and each episodic instance of the peer agent's policy can be viewed as analogous to a new MDP.

3 CONCLUSION

When and why reusable structure in RL survives across episodes was attempted to be formalized, and to show how the decentralization of agents destabilizes it through the agent-world boundary which was perceived as a continual learning problem (Khetarpal et al., 2020). Our analysis introduced an *invariant core* (i.e common subsequences of successful trajectories), proved its existence in stationary single-agent settings under mild assumptions (unique absorbing goal or an appropriate abstraction), and showed that embedding the task in a decentralized Markov game induces policy-driven drift

that can remove previously shared prototypes. A quantification of this drift was then shown with a variation budget V_E, linking agent-world boundary movement to the loss of invariants and explaining why transfer can fail even when the underlying task is unchanged.

This boundary-centered view matters because it reframes decentralized MARL as continual RL. Not only as adaptation to non-stationarity, but as robustness to *agent–world boundary* instability. Further work should consider: **1.** *preserve* invariants via options or deviation mechanisms that remain valid under small V_E (Elelimy et al., 2025; Sutton et al., 1999; Konidaris & Barto, 2009) and **2.** *predict* or *influence* boundary shifts to be predictable via opponent modeling or recursive reasoning so cores remain exploitable (He et al., 2016; Raileanu et al., 2018; Foerster et al., 2018; Jaques et al., 2019). Possible next steps include algorithms with guarantees that scale in V_E, online estimation of V_E from rollouts, and benchmarks that vary the boundary in controlled ways. Altogether, these considerations frame decentralized MARL as a continual-RL problem grounded in the agent–world boundary.

REFERENCES

David Abel, David Hershkowitz, and Michael Littman. Near optimal behavior via approximate state abstraction. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2915–2923, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/abel16.html.

David Abel, André Barreto, Michael Bowling, Will Dabney, Shi Dong, Steven Hansen, Anna Harutyunyan, Khimya Khetarpal, Clare Lyle, Razvan Pascanu, et al. Agency is frame-dependent. *arXiv preprint arXiv:2502.04403*, 2025.

Michael Bowling and Manuela Veloso. Multiagent learning using a variable learning rate. *Artificial intelligence*, 136(2):215–250, 2002.

Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 38(2): 156–172, 2008.

Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Reinforcement learning for non-stationary Markov decision processes: The blessing of (More) optimism. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1843–1854. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/cheung20a.html.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minigrid: A minimalistic gridworld environment for openai gym. https://github.com/Farama-Foundation/Minigrid, 2018.

Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. *Proceedings of the National Conference on Artificial Intelligence*, 15(2):746–752, 1998.

Thomas Dean and Robert Givan. Model minimization in markov decision processes. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, AAAI'97/IAAI'97, pp. 106–111. AAAI Press, 1997. ISBN 0262510952.

Esraa Elelimy, David Szepesvari, Martha White, and Michael Bowling. Rethinking the foundations for continual reinforcement learning. In *Reinforcement Learning Conference*, 2025.

Eyal Even-Dar, Sham Kakade, and Yishay Mansour. Online markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

Jakob Foerster, Nantas Nardelli, Gregory Farquhar, Triantafyllos Afouras, Philip HS Torr, Pushmeet Kohli, and Shimon Whiteson. Stabilising experience replay for deep multi-agent reinforcement learning. In *International conference on machine learning*, pp. 1146–1155. PMLR, 2017.

Jakob Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 122–130, 2018.

Anna Harutyunyan. What is an agent? Online, 2020. Available at: https://annaharutyunyan.github.io/blog/what-is-an-agent (accessed 2025-09-02).

He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. Opponent modeling in deep reinforcement learning. In *International conference on machine learning*, pp. 1804–1813. PMLR, 2016.

Chia-Chun Hung, Timothy Lillicrap, Josh Abramson, Yori Wu, Adam Marblestone, and Greg Wayne. Optimizing agent behavior over long time scales by transporting value. *Nature Communications*, 10(1):1362, 2019.

James W. Hunt and Thomas G. Szymanski. A fast algorithm for computing longest common subsequences. *Commun. ACM*, 20(5):350–353, May 1977. ISSN 0001-0782. doi: 10.1145/359581.359603. URL https://doi.org/10.1145/359581.359603.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*, pp. 3040–3049. PMLR, 2019.

Khurram Javed and Richard S Sutton. The big world hypothesis and its ramifications for artificial intelligence. In *Finding the Frame: An RLC Workshop for Examining Conceptual Frameworks*, 2024.

Nan Jiang, Alex Kulesza, Satinder Singh, and Richard Lewis. The dependence of effective planning horizon on model accuracy. In *Proceedings of the 2015 international conference on autonomous agents and multiagent systems*, pp. 1181–1189, 2015.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on learning theory*, pp. 2137–2143. PMLR, 2020.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020.

Dong-Ki Kim, Matthew Riemer, Miao Liu, Jakob Foerster, Michael Everett, Chuangchuang Sun, Gerald Tesauro, and Jonathan P How. Influencing long-term behavior in multiagent reinforcement learning. *Advances in Neural Information Processing Systems*, 35:18808–18821, 2022.

George Konidaris and Andrew Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. *Advances in neural information processing systems*, 22, 2009.

Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. *AI&M*, 1(2):3, 2006.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pp. 157–163. Elsevier, 1994.

Dane Malenfant and Blake A. Richards. The challenge of hidden gifts in multi-agent reinforcement learning, 2025. URL `https://arxiv.org/abs/2505.20579`.

Weichao Mao, Kaiqing Zhang, Ruihao Zhu, David Simchi-Levi, and Tamer Basar. Near-optimal model-free reinforcement learning in non-stationary episodic mdps. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7447–7458. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/mao21b.html`.

Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. Springer, 2016.

Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. Modeling others using oneself in multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008.

Chen Sun, Wannan Yang, Thomas Jiralerspong, Dane Malenfant, Benjamin Alsbury-Nealy, Yoshua Bengio, and Blake Aaron Richards. Contrastive retrospection: Honing in on critical steps for rapid learning and generalization in reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.