Université de Montréal

Metacognitive Architecture for Perceptual and Social Systems: A Neuro-inspired Metacognition Approach

By
Juan David Vargas Mazuera

Département d'informatique et de recherche opérationnelle (DIRO)

Thesis presented in fulfillment of the requirements for the degree of Master of Science (M.Sc.) in Computer Science

April 2025

Université de Montréal
Département d'informatique et de recherche opérationnelle (DIRO)

This Thesis is titled

Metacognitive Architecture for Perceptual and Social Systems: A Neuro-inspired Metacognition Approach

Presented by
Juan David Vargas Mazuera

Was evaluated by a jury composed of the following members:

David Scott Krueger
President-rapporteur

Guillaume Dumas
Main research supervisor

Zahra Sheikhbahaee
Co-supervisor

Guillaume Lajoie
Jury member

Résumé

Cette thèse de maîtrise se concentre sur une approche pour implémenter la métacognition comme la transformation de représentations de premier ordre (réseau de 2e ordre), et l'étude de différents composants (à la fois le réseau de 2e ordre et le modèle en cascade) dans une diversité d'environnements allant de scénarios perceptuels simples à des environnements agentiques complexes avec de multiples obstacles et agents d'arrière-plan, et l'apprentissage continu dans des environnements à agent unique.

Abrégée sous le nom de MAPS (Architecture Métacognitive pour l'Apprentissage Perceptuel et Social), cette approche proposée intègre à la fois un réseau de second ordre (métacognitif) et une montée graduelle et fluide d'activation (modèle en cascade) dans les systèmes d'IA (SIA) pour améliorer à la fois l'apprentissage social et continu. Nous évaluons MAPS à travers quatre conditions : apprentissage perceptuel (Connais-toi toi-même), SARL (MinAtar), SARL avec apprentissage continu (SARL+CL, MinAtar), et MARL (MeltingPot 2.0). Pour évaluer l'apprentissage social, nous comparons un réseau de confiance de 2e ordre dans des tâches perceptuelles vs. sociales, analysant son impact sur la prise de décision et les dynamiques d'interaction. Pour l'apprentissage continu, un réseau enseignant de 2e ordre stabilise l'intégration de nouvelles connaissances, prévenant la perte de connaissances passées.

Les résultats montrent que les mécanismes métacognitifs améliorent significativement l'adaptabilité dans les SIA. Dans les tâches perceptuelles, le modèle en cascade améliore l'apprentissage structuré et le flux d'information. Dans SARL, combiner un réseau de 2e ordre avec un modèle en cascade permet l'adaptation de comportements complexes. Dans SARL+CL, cela minimise l'oubli catastrophique plus efficacement que DQN. Dans MARL, MAPS montre du potentiel dans des environnements à haute variabilité, bien que des tests supplémentaires soient nécessaires. Ces résultats suggèrent la métacognition comme un outil puissant pour améliorer l'efficacité d'apprentissage et la compétence sociale de l'IA.

Mots-clés : Métacognition, Réseau de Second Ordre, IA Neuro, Apprentissage par renforcement multi-agent, Modèle en cascade, Connais-toi toi-même, MinAtar, Meltingpot, Récompenses collectives, IA coopérative, Paramètres compétitifs, Équité, Évolutivité, IA explicable, Apprentissage par renforcement à agent unique, Apprentissage continu.

Abstract

This master's thesis focuses on an approach to implement metacognition as the transformation of first order representations (2nd order network), and the study of different components (both 2nd order network and cascade model) in a diversity of environments from simple perceptual scenarios to complex agentic environments with multiple obstacles and background agents, and continual learning in single agent environments.

Abbreviated as MAPS (Metacognitive Architecture for Perceptual and Social Learning), this proposed approach integrates both a second order (metacognitive) network and a smooth graded build-up of activation (cascade model) into AI systems (AIS) to improve both social and continual learning. We evaluate MAPS across four conditions: perceptual learning (Know Thyself), SARL (MinAtar), SARL with continual learning (SARL+CL, MinAtar), and MARL (MeltingPot 2.0). To assess social learning, we compare a 2nd-order confidence network in perceptual vs. social tasks, analyzing its impact on decision-making and interaction dynamics. For continual learning, a 2nd-order teacher network stabilizes new knowledge integration, preventing past knowledge loss.

Results show that metacognitive mechanisms significantly enhance adaptability in AIS. In perceptual tasks, the cascade model improves structured learning and information flow. In SARL, combining a 2nd-order network with a cascade model enables complex behavior adaptation. In SARL+CL, it minimizes catastrophic forgetting more effectively than DQN. In MARL, MAPS shows promise in high-variability environments, though further testing is needed. These findings suggest metacognition as a powerful tool for enhancing AI's learning efficiency and social competence.

Keywords: Metacognition, Second Order Network, Neuro AI, Multi-agent reinforcement learning, Cascade model, Know Thyself, MinAtar, Meltingpot, Collective

rewards, Cooperative AI, Competitive settings, Fairness, Scalability, Explainable AI, Single-agent reinforcement learning, Continual learning.

Contents

List of Tables

List of Figures

List of abbreviations

MAPS:	Metacognitive architecture for perceptual and social learning
RL:	Reinforcement learning
SARL:	Single agent reinforcement learning
MARL:	Multi agent reinforcement learning
DQN:	Deep Q-network
PPO:	Proximal policy optimization
MAPPO:	Multi agent proximal optimization
AIS:	Artificial intelligence systems
CL:	Continual learning

Acknowledgments

First and foremost, I would like to express my deep gratitude to my research supervisor Dr. Guillaume Dumas, who placed his trust in me from the first moment when I had the opportunity to show my dedication through replicating a paper, even when my expertise in PyTorch and deep learning was limited. Also, I deeply thank him for his support under difficult financial circumstances where he offered me an additional opportunity to help him as a research assistant to write the grant for the digital alliance of Canada. I have been supported by him in many ways, for which I am deeply grateful, from conferences to letters of recommendations for my future endeavors, to reference letters for grants, and the opportunity to be part (content co-author) of interesting initiatives as the Neuromatch course. In a similar way, I thank my co-supervisor Dr. Zahra Sheikhbahaee, who also was a big part of my addition to the PPSP team, and with whom I shared many professional growth moments, learning about her very interesting and challenging modular attention project. I also thank her for having given me great insights into this project (MAPS) when I encountered difficulties as when implementing the architecture in the reinforcement learning scenarios, as well as when implementing continual learning. In addition to this, I want to thank both of my supervisors for their help with ideas during the conceptualization process of the experiments to carry out for this project.

I would also like all the PPSP team members, whom I have had the pleasure of working for over a year and who are all very interesting people as well as talented. Special thanks to Remy Ramadour, who has been very supportive on different projects by giving me meaningful feedback, instrumental to the good results in most cases. And an even greater thanks to Natalie Kastel, who was a pivotal part in the writing delivery of the articles, and who contributed greatly by putting the content of the articles together in a way that was easier to understand and with less technical jargon (as was the case with my first drafts).

I would also like to thank Dr. Axel Cleeremans and Dr. Antoine Pasquali; each of whom influenced the article and helped shape one of the mechanisms that would be fundamental to MAPS (cascade model).

I am also grateful to the funding organizations that have supported me, directly or indirectly through the PPSP Team, including the Institute for Advanced Consciousness Studies (IACS) for their sponsorship that led to the realization of this project, CIFAR, Canada Digital Alliance / Calcul Quebec, and Dr. Irina Rish's research laboratory (additional Canada Digital Alliance allocation sponsorship other than the one from the PPSP Laboratory) for computing resources, and the UNIQUE center and the Mila Quebec AI Institute for travel funding to conferences.

Finally, I wish to express my gratitude to my family, who supported me through my life and that are at least half of the reasons of me becoming the person I am today, and possibly the person that I will become in the future. This is specially thanks to my parents and my grandparents, who have loved me and supported me through time, and that such support has allowed me to dream of doing whatever I set my eyes on, no matter if I come from relatively humble roots. I also want to thank my partner Qian Ai, who is a wonderful woman that has loved me through the realization of my MSc. I feel incredibly fortunate to have her by my side and pray for our future projects together, both professionally and personally.

To all of you, thank you. Your help, encouragement, and guidance have made this research experience an enriching journey, both scientifically and personally.

Chapter 1

General introduction

1.1 Metacognition

In cognitive science, metacognition refers to the capacity to monitor and regulate one's cognitive processes. Metacognition, originally introduced in the 70ies by Flavell, emphasizes higher-order reasoning about oneself [Flavell, 1979], involving active control over the thinking process through self-monitoring, knowledge assessment, and conscious awareness of mental activities [S. Kala and Rana, 2022].

In cognitive psychology, metacognition has become an increasingly prominent focus over the past three decades [S. Kala and Rana, 2022], particularly for learning processes [Flavell, 1979]. Metacognition is studied as an essential component of effective learning, requiring self-reflection on one's strengths, weaknesses, and methods [Murugesan et al., 2022, Jagota et al., 2021]. In this line, recent empirical work suggests that while humans may possess an inherent capacity for implicit metacognition from infancy [Goupil et al., 2016], explicit metacognitive abilities develop gradually through learning experience [Goupil and Kouider, 2019].Paralleling these human capabilities, artificial intelligence research has increasingly recognized metacognition as a crucial mechanism for developing more adaptive and efficient learning systems [Sugiyama et al., 2023]. Recent advances in AI demonstrate that embedding metacognitive processes through second-order networks—where a primary task network is paired with a reflective network that evaluates its performance—enables artificial agents to achieve faster convergence and more robust learning outcomes [Sandberg et al., 2010, Norman and Clune, 2024].

Metacognition also extends beyond individual cognitive monitoring to contribute significantly to both intrapersonal decision-making—helping individuals recognize errors and regulate ongoing thought [Rabbit, 1966]—and suprapersonal decision-making [Shea et al., 2020], allowing individuals not only to monitor their internal cognitive processes but also to externalize and share mental states with others [Shea et al., 2014, Dunstone and Caldwell, 2018, Heyes, 2016]. In artificial intelligence, this multi-level metacognitive capacity has been leveraged to develop safer and more responsible AI systems, where metacognitive frameworks enable AI agents to monitor their own decision-making processes and recognize their limitations [Walker et al., 2025]. Furthermore, computational approaches to metacognition have shown promise in addressing critical challenges such as catastrophic forgetting in neural networks, where metacognitive mechanisms help agents retain previously learned knowledge while acquiring new information [Kirkpatrick et al., 2017, Lopez-Paz and Ranzato, 2017].

Research in developmental psychology also highlights a significant link between early metacognitive monitoring skills and later ToM (the ability to attribute mental states such as beliefs, desires, and intentions to others) competencies, suggesting that metacognition lays the foundation for the development of ToM use [Feurer et al., 2015]. Artificial intelligence seeks to mimic this interplay by integrating self-awareness into multi-agent AI systems. This approach has shown to foster more robust and flexible learning algorithms, enabling artificial agents to optimize social interactions [Conway-Smith and West, 2024a, Zaroukian, 2022]. Building on this foundation, recent work in AI has explored how metacognitive capabilities can be integrated with meta-learning approaches—the ability to "learn to learn"—to enable rapid adaptation across diverse domains with minimal training data [Finn et al., 2017, Botvinick et al., 2019]. These advances demonstrate that metacognitive AI systems can dynamically adjust their learning strategies, shifting from exploration to exploitation once mastery is achieved, thereby accelerating skill acquisition in complex environments [Anderson et al., 2006]. By combining self-regulation with social reasoning, metacognition promises to enhance ToM, allowing AI systems to engage in more adaptive and human-like interactions. In fact, several researchers have independently proposed that higher cognitive functions, especially consciousness [Bengio, 2017] and metacognition [Cortese et al., 2019], serve crucial functions in processes beneficial for learning efficiently in hierarchical systems. Both consciousness and metacognition may in fact be closely correlated and share key neural mechanisms [Brown et al., 2019, Morales and Lau, 2021]. Indeed, metacognition interacts with reinforcement learning in ways that may provide key solutions for developing flexible behavioral policies in complex environments [Cortese et al., 2021, 2020, Lak et al., 2020].

In this study, we implement a Metacognitive Architecture for Perceptual and Social learning (MAPS), built on the foundation of A. Pasquali and Cleeremans [2010], who implements a 2nd-order network. As metacognition is frequently assessed through measurements of confidence (which can relate to overall self-confidence, to decisions, perceptions, or memory) [Cortese, 2020], this approach transforms first order representations through a 2nd-order (confidence) network. This framework pairs a main task network, responsible for performing primary functions such as image recognition or gameplay, with a 2nd-order network that evaluates its performance.

Acting as a reflective mechanism, the 2nd-order network assesses confidence levels, identifies knowledge gaps, and triggers adaptive adjustments to improve outcomes [Sandberg et al., 2010]. By doing so, MAPS offers a robust framework for advancing both perceptual and social learning in AIS.

Metacognitive task-specific self-regulation strategies may offer a path forward even in environments with complex social dynamics. Through planning, monitoring, evaluating, and reflecting tactics, AI systems might acquire the qualities that characterize robust human intelligence, potentially leading to next-generation artificial intelligence with enhanced metacognition, consciousness, and the ability to learn efficiently from limited samples [Dehaene et al., 2017]. However, despite the potential of metacognition to optimize ToM and improve social understanding in artificial agents, most computational implementations remain limited to basic pattern detection, or 'perceptual' tasks [Kanai et al., 2024]. These simplified approaches fail to fully harness metacognition's potential for socially relevant applications, leaving significant room for exploration in this area. In this study, we evaluate MAPS across four conditions (both perceptual and social tasks): perceptual learning (Know Thyself), SARL (MinAtar), SARL with continual learning (SARL+CL, MinAtar), and MARL (MeltingPot 2.0).

1.2 Aims

In this study, we extend the application of MAPS to investigate the behavior and performance of AI architectures when implementing a 2nd-order (metacognitive) network not only in perceptual tasks but also in SARL, MARL, and continual learning tasks. RL offers an ideal framework for evaluating social learning dynamics, as it goes beyond simple pattern detection by engaging agents in complex decision-making and interaction scenarios [Ndousse et al., 2021]. This allows us to examine, in a systematic manner, whether metacognition enhances social behavior and performance in these advanced learning settings. Pattern detection abilities are explored using the 'Blindsight' and 'AGL' tasks [A. Pasquali and Cleeremans, 2010], SARL and SARL + continual learning is explored using environments with MinAtar games [Young and Tian, 2019], and MARL scenarios are explored with the DeepMind MeltingPot 2.0

suite [Agapiou et al., 2023].

The three aims of this project are: 1) to investigate the role of metacognitive systems in distinguishing between perceptual input, memory recall and spontaneous neural activity during perceptual tasks; 2) to assess the impact of metacognitive systems on social interactions and complex agentic reinforcement learning settings (both single agent environments, and multi agent environments); and 3) to assess the impact of metacognitive systems towards minimizing catastrophic forgetting in continual learning.

1.3 Contributions

1. This paper proposes an architecture for improved learning using a confidence (2nd order) network, which is tested in a variety of environments. We test it from simple pattern detection, to single agent environments with multiple obstacles, and multi agent reinforcement learning. We show that in a variety of complex and high-variability settings, our architecture can exhibit improved performance over not using the basic elements of the architecture (2nd order network and cascade model).

   Context: Prior work established a similar concept through a different implementation, meta-autoencoders architecture. This architecture also aims to learn representations of first-order neural networks, however it used different components and wasn't tested in complex environments as single agent and multi agent reinforcement learning Kanai et al. [2024].

2. This paper introduces the use of cascade model to an existing metacognitive architecture consisting of a 2nd order confidence network. We show that the cascade model plays a central role, improving structured learning and information flow. In environments with uncontrolled social dynamics (SARL), the combination of a 2nd-order network and a cascade model is relevant for effective learning, particularly in tasks with dynamic obstacles or interactions.

   Context: Prior work introduced an architecture that used a 2nd order network for confidence judgments, but didn't include a cascade model nor tested it on complex environments A. Pasquali and Cleeremans [2010].

1.4 Co-authors' contributions

In the realization of this project, we build an interdisciplinary team with various competencies, from Neuroscience, to psychology, and computer science. The relevant article to this thesis was supervised by Dr. Guillaume Dumas, who provided both funding and ideas for the realization of the project. The co-supervisor of the project was Dr. Zahra Sheikhbahaee, who mainly supported the project when I encountered technical difficulties as when implementing the architecture in the reinforcement learning scenarios, as well as when implementing continual learning. Another member of the PPSP laboratory, Natalie Kastel, was a pivotal part writing the introduction of the article as well as supporting the writing of other sections to make the message clearer and easier to digest.

On the other hand, we also had periodical meetings with both Dr. Axel Cleeremans and Dr. Antoine Pasquali, the original authors of the Know Thyself paper were they implemented a 2nd order network in 2 environments (Blindsight, and Artificial Grammar Learning); each of whom influenced the article and helped shape one of the mechanisms that would be fundamental to MAPS (cascade model).

Chapter 2

Scientific article: MAPS- A Metacognitive Architecture for Improved Perceptual and Social Learning: from simple tasks to multi-agent reinforcement learning

MAPS- A Metacognitive Architecture for Improved Perceptual and Social Learning: from simple tasks to multi-agent reinforcement learning

Paper submitted to be submitted to TMLR (Transactions on Machine Learning Research)

Authors:

Juan David Vargas[1,2,3], Natalie Kastel[1,3,6], Antoine Pasquali[4], Axel Cleeremans[5], Zahra Sheikhbahaee[*1,3], Guillaume Dumas[*1,3,6]

Affiliations:

1. Precision Psychiatry and Social Physiology (PPSP) laboratory CHU Sainte-Justine, Montréal, Québec, Canada
2. DIRO, Université de Montréal, Montréal, Québec, Canada
3. MILA - Quebec AI Institute, Montréal, Québec, Canada
4. Université libre de Bruxelles, Bruxelles, Belgium
5. CrossLabs, Kyoto, Japan
6. Départment de Psychiatrie, Université de Montréal, Montréal, Québec, Canada

[*]: Co-Senior Authors

2.1 Abstract

Reinforcement Learning (RL) has made significant strides but struggles with social and continual learning. Cognitive neuroscience highlights metacognition as key to human self-monitoring, knowledge retention, and adaptive behavior, yet its potential in AI remains underexplored. Metacognition could mitigate RL's catastrophic forgetting and enhance social intelligence, but current implementations focus on basic perceptual tasks, overlooking broader applications. This study introduces the Metacognitive Architecture for Perceptual and Social Learning (MAPS), integrating a second-order (metacognitive) network into AI systems (AIS) to improve both social and continual learning. We present a new combination of techniques using both a second-order network and a cascade model. We hypothesize the cascade model will improve the information extracted, to be used by the second-order network. MAPS is the combination of these 2 techniques. We evaluate MAPS across four conditions: perceptual learning (Know Thyself), SARL (MinAtar), SARL with continual learning (SARL+CL, MinAtar), and MARL (MeltingPot 2.0). To assess social learning, we compare a second-order network in perceptual vs. social tasks, analyzing its impact on decision-making and interaction dynamics. For continual learning, results are limited as using a second-order network seems to stabilize new knowledge integration, preventing past knowledge loss, however, limited to one additional training environment. Results show that metacognitive mechanisms significantly enhance adaptability in AIS. In perceptual tasks, the cascade model improves structured learning and information flow. In SARL, combining a second-order network with a cascade model enables complex behavior adaptation. In SARL+CL, it prevents catastrophic forgetting more effectively than DQN. In MARL, MAPS shows improved performance in environments with relatively limited gaming and social properties, while for overly complex social environments still fails to generalize successfully. In addition, using MAPS effectively increases the convergence speed in all MARL environments to more stable agentic behaviour (measured by using the action distribution entropy). These findings suggest that metacognition is potentially a powerful tool to enhance AI learning efficiency and social competence.

2.2 Introduction

Reinforcement Learning (RL) differs from supervised and unsupervised learning in that it acquires knowledge through direct interaction with an environment, refines decisions through trial and error, and optimizes behavior based on rewards and penalties. This dynamic enables breakthroughs in game-playing AI Silver et al. [2016], robotics Zhang and Mo [2021] , and autonomous systems Jeyaraman et al. [2024]. However, despite its adaptability, RL remains far less efficient than human learning Koedinger et al. [2023]. Over millions of years, humans have evolved cognitive shortcuts and adaptive mechanisms that allow rapid generalization in environments and tasks - the abilities RL still struggles to replicate Jain et al. [2020].

One critical cognitive shortcut that humans possess - but standard AI lacks - is self-awareness, or **metacognition**. Metacognition refers to the capacity to monitor and regulate one's cognitive processes, and emphasizes higher-order reasoning about oneself Flavell [1979], involving active control over the thinking process through self-monitoring, knowledge assessment, and conscious awareness of mental activities S. Kala and Rana [2022]. This deeply human trait enables faster learning, better decision-making, and more efficient resource use Lu et al. [2025] by allowing individuals to recognize mistakes early and adapt strategies accordingly, minimizing trial and error, cognitive load, and inefficiencies in problem-solving. Additionally, metacognition enhances confidence calibration, ensuring individuals act decisively when correct and re-assess when uncertain, leading to more effective and adaptive learning Garbayo et al. [2023].

In recent years, metacognition has been integrated into RL to replicate humans' ability to self-correct and achieve greater learning efficiency Sugiyama et al. [2023]. One method for embedding metacognitive processes is through a **second-order network**—a framework that pairs a primary task network (e.g., for image recognition or gameplay) with a second network (comparable smaller in terms of parameters) dedicated to evaluating its performance, and which input is the transformation of the main network's inputs (input - output). Serving as a reflective mechanism, the second-order network assesses confidence levels, detects knowledge gaps, and triggers adaptive adjustments to enhance learning outcomes Sandberg et al. [2010]. Research shows that, much like in humans, embedding metacognitive abilities in RL agents enables them to assess their own progress and dynamically adjust their strategies. For example, metacognitive RL agents can shift from exploration to exploitation once mastery is achieved Norman and Clune [2024] or reduce redundant trials, accelerating convergence to optimal policies Anderson et al. [2006]. Beyond reinforcement learning, recent work has explored metacognition's potential for creating safer and more responsible AI systems. Walker et al. [2025] investigate how metacognitive capabilities can be integrated into AI systems to improve safety and alignment with human values, emphasizing frameworks that enable AI systems to monitor their own decision-making processes and recognize their limitations. Similarly, Conway-Smith and West [2024b] focus on developing AI systems that can adapt and optimize their own learning strategies through metacognitive processes, drawing insights from the ACT-R cognitive architecture to inform the design of self-reflective AI systems. These mechanisms enhance exploration-exploitation balance, accelerate skill acquisition, and improve adaptability in complex environments, while also addressing critical concerns about autonomous decision-making, making metacognition a key factor in developing more intelligent, efficient, and trustworthy RL systems.

The influence of metacognition on learning extends beyond individual cognition to social learning. Evidence of this connection lies in Theory of Mind (ToM)—the human ability to understand others in a social context Feurer et al. [2015]—which is believed to be rooted in metacognitive abilities Frith [2012]. This suggests that self-reflection forms the foundation for understanding others, as the same cognitive mechanisms that allow us to evaluate our own thoughts and behaviors also help us interpret the intentions and perspectives of those around us Kastel et al. [2023]. Also connected is meta-learning, the ability to "learn to learn", and a "branch" of metacognition, which has been explored by the AI community for a variety of tasks, one of them being RL. For example, Botvinick et al. [2019] highlight how deep reinforcement learning methods that take advantage of episodic memory and meta-learning reveal fundamental connections between rapid adaptation and incremental learning, drawing parallels to human cognitive processes. Similarly, Finn et al. [2017] propose model-agnostic meta-learning approaches that enable rapid adaptation to new tasks with minimal training data, effectively training models to be easily fine-tunable across diverse domains. In essence, reflection is a fundamental and transferable human skill, facilitating both self-awareness and social cognition, as we naturally draw parallels between our own experiences and those of others Lincoln et al. [2020]. This ability is crucial for effective social interaction and cooperation, reinforcing metacognition's central role in both individual and collective intelligence.

Despite its potential to enhance both individual and social intelligence in artificial agents, the full capabilities of metacognition in AI remain largely unexplored. In individual learning, its role in enabling continual learning across tasks and environments is often overlooked Sidra and Mason [2024]. Catastrophic forgetting—where AI loses previously learned knowledge when acquiring new information—remains a major challenge, particularly in neural networks, where new learning overwrites existing representations Kemker et al. [2018]. Kirkpatrick et al. [2017] showed that this limitation can be addressed by selectively slowing down learning on weights important for previous tasks, while Lopez-Paz and Ranzato [2017] proposed Gradient Episodic Memory to alleviate forgetting while enabling beneficial knowledge transfer across tasks. However, despite these promising approaches to mitigate catastrophic forgetting, continual learning is still an open problem as, unlike humans, agents struggle to retain skills across different tasks. Similarly, in social learning, most computational implementations are limited to basic perceptual tasks Kanai et al. [2024], failing to leverage metacognition's potential for socially relevant applications. Addressing these gaps could unlock more adaptive, transferable, and socially intelligent AI systems.

This study aims to explore and evaluate the potential benefits of metacognitive abilities in AI systems (AIS), focusing on both social and continual learning. We introduce the Metacognitive Architecture for Perceptual and Social Learning (MAPS), and investigate whether AIS performs better in these domains when implementing a second-order network. To assess social learning, we integrate a second-order confidence network not only in perceptual tasks but also in single-agent (SARL) and multi-agent (MARL) reinforcement learning scenarios. RL provides an ideal framework for studying social learning dynamics, as it moves beyond basic pattern detection to engage agents in complex decision-making and interactions Ndousse et al. [2021]. This structured approach allows us to systematically examine whether metacognition enhances both social behavior and overall performance in advanced learning environments.

To examine continual learning within a metacognitive architecture, we implement a teacher network designed to help AI retain past knowledge while acquiring new skills, addressing the challenge of catastrophic forgetting. This network stores learned representations from previous tasks and serves as a reference for the main task network, which actively learns new information. As the AI adapts, it compares its outputs to those of the teacher network, ensuring that new learning does not overwrite essential prior knowledge. This balance is maintained through a hybrid loss function, which combines three key components: current task loss to focus on new learning, weight regularization loss to prevent deviation from past knowledge, and feature loss to stabilize internal representations.

Building on this framework, we test MAPS across four key conditions to evaluate its impact on both social and continual learning: pattern recognition (Know Thyself), SARL (MinAtar, illustration of envrionments in Figure 2.1), SARL with Continual learning (SARL+CL, MinAtar), and MARL (MeltingPot 2.0, illustration of envrionments in Figure 2.1). To investigate social learning, we compare the benefits of a second-order confidence network in perceptual vs. social (SARL and MARL) tasks, examining whether metacognition enhances decision-making and interaction dynamics. For continual learning, we implement a teacher network, acting as a reference for the main task network, ensuring new knowledge integrates smoothly without erasing past learning. Through these experiments, we systematically assess the effectiveness of metacognition in fostering more adaptable and socially intelligent AI systems.

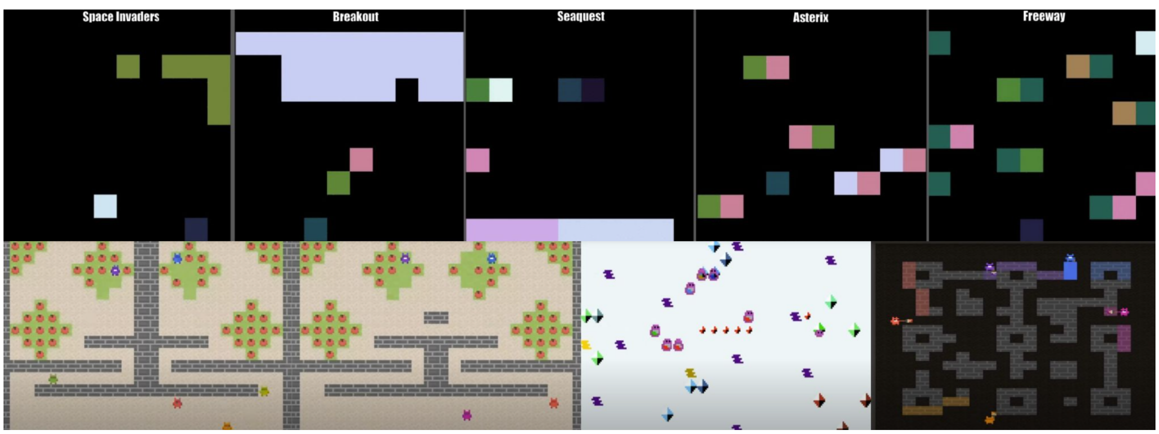

Figure 2.1: Visualization of trained agents of MinAtar(top), and Melting Pot 2.0 (bottom). The tested MinAtar scenarios are: Space Invaders(1st image to the left), Breakout(2nd), Seaquest(3rd), Asterix (4th), and Freeway(5th). For Melting Pot 2.0: Commons Harvest Closed(1st), Commons Harvest Partnership(2nd), Chemistry Three Metabolic Cycles with Plentiful Distractors(3rd), and Territory Inside Out(4th).

For easy reference, Table 2.1 show some of the most notable related work previously mentioned.

Article Title	Authors & Year	Domain	Brief Description
The metacognitive loop I: Enhancing reinforcement learning with metacognitive monitoring and control for improved perturbation tolerance	Anderson et al. [2006]	Metacognition	Demonstrates how metacognitive RL agents can reduce redundant trials and accelerate convergence to optimal policies through self-assessment
Harnessing Metacognition for Safe and Responsible AI	Walker et al. [2025]	Metacognition	Investigates integration of metacognitive capabilities in AI systems for improved safety and alignment with human values through self-monitoring
Toward Autonomy: Metacognitive Learning for Enhanced AI Performance	Conway-Smith and West [2024a]	Metacognition	Develops AI systems that adapt and optimize learning strategies through metacognitive processes, drawing from ACT-R cognitive architecture
Know Thyself: Metacognitive Networks and Measures of Consciousness	A. Pasquali and Cleeremans [2010]	Metacognition	Explores metacognitive networks and measures of consciousness, establishing foundational understanding of self-awareness in cognitive systems. Explores and evaluates perceptual tasks using a second-order network
Overcoming catastrophic forgetting in neural networks	Kirkpatrick et al. [2017]	Continual Learning	Addresses catastrophic forgetting by selectively slowing down learning on weights important for previous tasks (Elastic Weight Consolidation)
Gradient Episodic Memory for Continual Learning	Lopez-Paz and Ranzato [2017]	Continual Learning	Proposes Gradient Episodic Memory to alleviate forgetting while enabling beneficial knowledge transfer across tasks
First-Explore, then Exploit: Meta-Learning to Solve Hard Exploration-Exploitation Trade-Offs	Norman and Clune [2024]	Meta-learning	using meta-RL, shows agents' ability to shift from exploration to exploitation once mastery is achieved, improving learning efficiency. This is a step towards human-like exploration on a variety of domains
Meta-Representations as Representations of Processes	Kanai et al. [2024]	Meta-learning	Investigates meta-representations as representations of processes, contributing to understanding of how AI systems can represent and learn from their own learning processes. Explores and evaluates perceptual tasks using a meta-autoencoder

Table 2.1: Related Work in Meta-learning, Metacognition, and Continual Learning AI Implementations

2.3 Methodology

Our research over the effect of the MAPS architecture is divided into analysis over 4 environments: pattern detection (using blindsight and artificial grammar learning; from Know-Thyself), single-agent reinforcement learning (using 5 MinAtar environments), single-agent reinforcement learning + continual learning (MinAtar), and multi-agent reinforcement learning (MARL; using 4 Google Deepmind Meltingpot environments). For MARL, we present mostly preliminary results. On the other hand, we implement a continual learning approach for single agent reinforcement learning following a curriculum, and study whether MAPS attenuate catastrophic forgetting. The overview of the environments, and the expected inputs/outputs is shown in Table 2.2. In addition, a detailed description of each environment can be read in Appendix A.

Environment	Input	Test cases	Output
Blindsight	400 patterns split between random noise (0.0-0.02 activations) and designed stimulus patterns (one unit with 0.0-1.0 activation)	Three conditions simulate visual impairment levels - suprathreshold (familiar patterns), subthreshold (increased noise), and low vision (decreased intensity)	Stimulus detection under three visual impairment conditions
AGL (Artificial Grammar Learning)	Artificially generated letter strings (3-8 letters) classified as random, grammar A, or grammar B patterns	Two conditions - implicit (3 epochs, low consciousness) and explicit (12 epochs, high consciousness) learning	String reconstruction (grammatical or non-grammatical) based on implicit pattern recognition. Network performance on distinguishing grammatical vs non-grammatical strings, testing incidental learning capabilities
MinAtar	10x10 pixel game states with multiple channels (paddle, ball, trail, brick positions) converted to tensors for neural network processing	Space Invaders, Breakout, Seaquest, Asterix , and Freeway	Action selection from discrete action spaces (e.g., no-op, left, right for Breakout) with +1 rewards
Meltingpot	11x11 RGB observation window (off-center view) showing 2D game world with walls, spawn points, and various resources/objects. Additional observations include agent-specific state information	Commons Harvest Closed, Commons Harvest Partnership, Chemistry Three Metabolic Cycles with Plentiful Distractors, and Territory Inside Out	Action selection from discrete action spaces

Table 2.2: Overview of training suites used, inputs, outputs, and test cases.

2.3.1 Methods overview

Our basic setup is a baseline neural network implementation that successfully completes a task without any additional components. We then extract information from the hidden representations of our main task network by building a comparison matrix, which will be the input of a second-order network (which is comparative smaller and simpler than the main neural network). The second-order nertwork is connected to 2 **wagering** units. Wagering is a more implicit, indirect, way to assess awareness Koch and Preuschoff [2007]. Thus, we wager high when there is high confidence in the output of the main neural network, and low otherwise (each neuron represents one of the two states). We learn from both losses(of the two neural networks) in sequence through backpropagation as to adjust the weights of our main network not only to the expected outcome, but to the expected confidence state as well. See equations 2.1 to 2.3 to understand the metacognitive signal. The input of the second-order network (comparison matrix = (inputs - outputs) = $(X_t - \hat{\mathbf{Y}}_t^{(1)})$), needs to contain information from the hidden states of the main neural network (not zero sum as, e.g. using MSE loss). Thus, we use the loss described in equation 2.4.

$$\mathbf{C}_t = \mathbf{X}_t - \hat{\mathbf{Y}}_t^{(1)} \tag{2.1}$$

$$\mathbf{C}'_t = \text{Dropout}(\mathbf{C}_t) \tag{2.2}$$

$$\mathbf{W}_t = (\mathbf{W}\mathbf{C}'_t + \mathbf{b}) \tag{2.3}$$

We employ a **contrastive loss** (equation 2.4) for the main task, formulated as the summation of a mean squared error term and a weighted ℓ_2-norm of the Jacobian of hidden units with respect to inputs. This loss maintains similarity and correlation of latent representations across modalities Chen et al. [2020]. We hypothesize that this facilitates the information flow essential for wagering decisions.

$$\mathcal{L}_{\text{contrastive}} = \ell_{i,j} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)} \tag{2.4}$$

Where:
$\mathbf{z}_i, \mathbf{z}_j$ = Latent representations (hidden units $\mathbf{h}$) for samples i and j
$\text{sim}(\cdot, \cdot)$ = Similarity function
τ = Temperature parameter controlling the sharpness of the distribution
N = Batch size. $2N$ represents positive and negative pairs

For wagering, we used a binary cross-entropy loss (equation 2.5) to handle class imbalance. This is applied to the output of equation 2.3. Additionally, we implement a **cascade model** (equation 2.6) in the main task network. Using cascade model, units at each level compute activations based on the hidden states from the preceding level, but with a temporal dynamics component. Unlike standard networks that process information in a single forward pass, the cascade model incorporate a constant rate that allows activations to build up gradually over time McClelland et al. [1989]. The selection of these architectural components reflects important inductive biases that shape learning and generalization. As Goyal and Bengio [2022] argue, inductive biases inspired by higher-level cognition into deep learning architectures is crucial for achieving better out-of-distribution generalization. Similarly, Battaglia et al. [2018] demonstrate how relational inductive biases, particularly through structured representations like graph networks, can facilitate learning about entities and their relationships, enabling more sophisticated patterns of reasoning. We hypothesize that these added components can be crucial for self-evaluation, as they could allow for the assessment of internal states through the temporal evolution of activations. We empirically selected 50 cascade iterations for all test cases ($\alpha = 0.02$).

$$\mathcal{L}_{\text{BCE}} = -\left[y \cdot \log(\sigma(\text{logits})) + (1-y) \cdot \log(1-\sigma(\text{logits}))\right] \tag{2.5}$$

$$a_{ir}(t) = \alpha \sum_j w_{ij} a_{js}(t) + (1-\alpha) a_{ir}(t-1) \tag{2.6}$$

Where:

$y \in \{0, 1\}$ = Ground truth binary label

$\sigma(\cdot)$ = Sigmoid activation function: $\sigma(x) = \frac{1}{1+e^{-x}}$

$\alpha \in [0, 1]$ = Cascade rate parameter

$a_{ij}(t)$ = Activation of neuron i in layer j at time step t

w_{ij} = Weight connection from neuron i to neuron j

2.3.2 Architecture

Know-Thyself environments

For pattern detection, we base our baseline implementation of a second-order network in the work of A. Pasquali and Cleeremans [2010]. Thus, for simplicity and to allow us to more easily discern the effect of MAPS, we use an auto-encoder for the primary task, and a comparator matrix connected to 2 wagering units for the second-order network as in A. Pasquali and Cleeremans [2010] (see Figure 2.2). As in the other environments, we use the cascade model in the main task network.

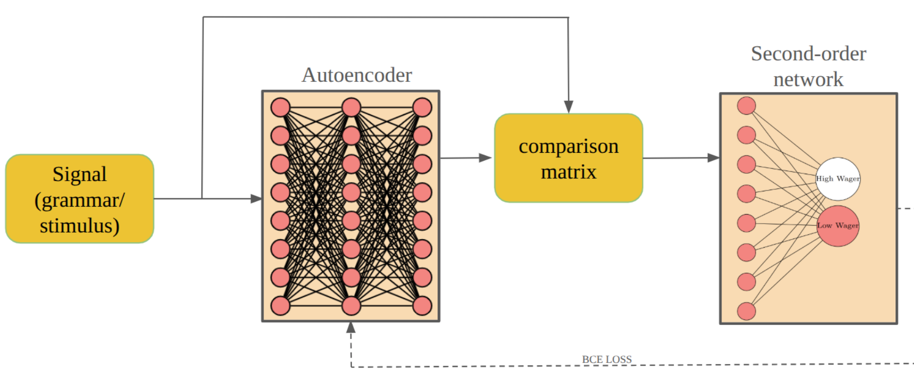

Figure 2.2: Blindsight/AGL architecture. The system implements a second-order network using an auto-encoder for the primary task and a comparator matrix connected to 2 wagering units.

Single and Multi agent reinforcement learning

For SARL (see Figure 2.3), we employ a DQN van Hasselt et al. [2015] framework. We use convolutional layers which allow for reduced computational complexity, a Q-network, and a replay buffer for the learning stability. We use cascade model in the main network as in equations 2.7 to 2.11.

For $i = 0, 1, 2, \ldots, N_{\text{cascade}} - 1$:

$$\mathbf{X}_{\text{flat}} = \text{Flatten}(\text{ReLU}(\text{Conv2d}(\mathbf{X}_{\text{input}}))) \tag{2.7}$$

$$\mathbf{H}^{(i)}_{\text{raw}} = \text{ReLU}(\mathbf{W}_{\text{hidden}}\mathbf{X}_{\text{flat}} + \mathbf{b}_{\text{hidden}}) \tag{2.8}$$

$$\mathbf{H}^{(i)} = \begin{cases} \alpha \cdot \mathbf{H}^{(i)}_{\text{raw}} + (1-\alpha) \cdot \mathbf{H}^{i-1} & \text{if } \mathbf{H}^{(i-1)} \neq \text{None} \\ \mathbf{H}^{(i)}_{\text{raw}} & \text{otherwise} \end{cases} \tag{2.9}$$

$$\mathbf{Q}^{(i)} = \mathbf{W}_{\text{actions}}\mathbf{H}^{(i)} + \mathbf{b}_{\text{actions}} \tag{2.10}$$

$$\tag{2.11}$$

We then compute the comparison matrix using the inputs and a reconstructed output using $\text{H}^{(i)}$ and the transpose of the weights at $\text{H}^{(i)}_{\text{raw}}$ (tied weights) as to maintain the number of parameters, and reduce complexity (see equation 2.12).

$$\hat{\mathbf{X}}_{\text{recon}} = \text{ReLU}(\mathbf{W}^T_{\mathbf{H}^{(i)}_{\text{raw}}}\mathbf{H}^{(i)} + \mathbf{b}_{\text{recon}}) \tag{2.12}$$

We then connect the comparison matrix to 2 wagering units using a linear layer. For the wagering objective, we compute rewards ($r_t^{(i)}$; $i \in 1, 2, ..., 128 =$ Batch index (128 samples per batch)) using an exponential moving average (EMA; equation 2.13) with a smoothing factor of $\alpha = 0.45$. At each step t, a low/high wager is assigned based on whether the last reward is greater than EMA (The learning objective is described in equation 2.14). α was found empirically and was used for both SARL, SARL + CL , and MARL. For MARL, the wagering signal was calculated independently for every agent.

$$\text{EMA}_t = \alpha \cdot r_t + (1-\alpha) \cdot \text{EMA}_{t-1} \tag{2.13}$$

$$y^{(i)}_{\text{wager}}(t) = \begin{cases} (1,0) & \text{if } r_t^{(i)} > \text{EMA}_t^{(i)} \text{ (high wager)} \\ (0,1) & \text{if } r_t^{(i)} \leq \text{EMA}_t^{(i)} \text{ (low wager)} \end{cases} \tag{2.14}$$

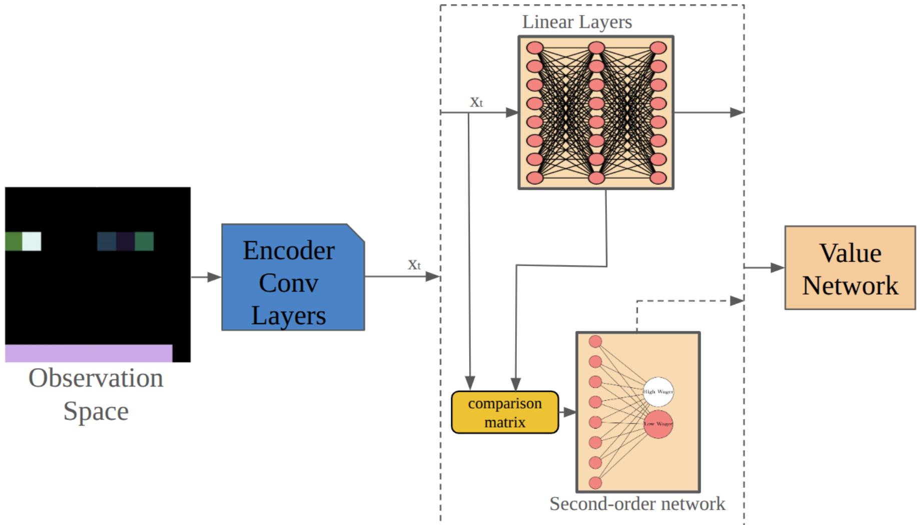

Figure 2.3: Single-agent reinforcement learning architecture. The system employs a DQN framework with convolutional layers, an auto-encoder, and a second-order network.

For MARL (see Figure 2.4), we use an MAPPO framework Yu et al. [2022], convolutional layers, sinusoidal-based relative positional encoding to add positional information, and a Gated Recurrent Unit (GRU) for stability. A second-order network is used as in SARL.

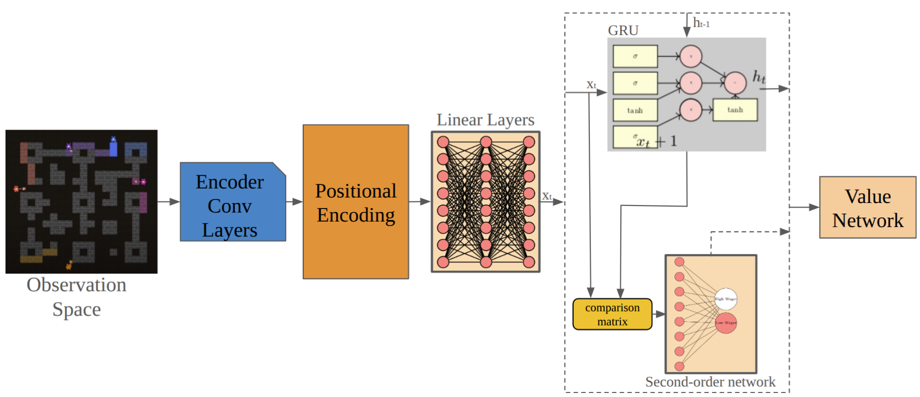

Figure 2.4: Multi-agent reinforcement learning architecture. The framework utilizes MAPPO with convolutional layers, sinusoidal-based relative positional encoding, a GRU, and a second-order network.

Continual Learning

We implement a continual learning approach following a curriculum using the SARL implementation as a baseline. As our aim is to train sequentially over the MinAtar environments, we modify the main task network (Q Network) to accommodate varying input channels across different environments. We adapt the Q network to handle multiple input channels by setting the input dimension to the maximum number of channels across all environments. For environments with fewer channels, we apply zero-padding to match the expected size, followed by a 1×1 convolution layer with ReLU activation to process inputs of different sizes while preserving spatial information. The output from this layer connects to our standard baseline Q network architecture.

Drawing inspiration from Li and Hoiem's work Li and Hoiem [2018], we implement a strategy to effectively retain information from previously encountered environments. Our approach employs a teacher network loaded with weights from the previously trained task. We calculate separate forward passes through both the current task network (main task network) and the previous task network (teacher network). We then utilize a hybrid loss function consisting of three weighted components: (1) the current task loss (using a contractive loss), (2) a weight regularization loss (inspired by elastic weight consolidation, which penalizes significant changes to model parameters from their previous state; Kirkpatrick et al. [2017]), and (3) a feature loss (the MSE loss between hidden layer outputs of both networks, using the teacher network as the target to preserve internal state behaviors of the previous model). In addition, all loss components are normalized using the maximum individual loss observed throughout epochs to ensure comparability and facilitate summation. Our training curriculum progresses through the following environments in sequence: Breakout, Space Invaders, Seaquest, and Freeway. This ordering reflects the environments that demonstrated the fastest convergence during our preliminary SARL experiments. In Figure 2.5, we can see a descriptive schematic of the architecture used for continual learning.

The weight regularization loss (Equation 2.15) constrains parameter drift by penalizing deviations of current network parameters θ_k from their teacher network counterparts. The feature preservation loss (Equation 2.16) maintains representational similarity between the hidden states $\mathbf{h}_1^{\text{student}}$ of the current network and the teacher network. The task-specific loss (Equation 2.17) varies depending on whether training the main network (contrastive loss) or second-order network (binary cross-entropy loss). These components are combined in our continual learning objective (Equation 2.18) with normalized weights that sum to one, ensuring balanced contribution from each loss component throughout the learning process.

$$\mathcal{L}_{\text{reg}} = \frac{1}{\max_t(\mathcal{L}_{\text{reg}}(t))} \sum_k \|\theta_k - \theta_k^{\text{teacher}}\|_2^2 \tag{2.15}$$

$$\mathcal{L}_{\text{feature}} = \frac{1}{\max_t(\mathcal{L}_{\text{feature}}(t))} \cdot \frac{1}{n} \sum_{i=1}^{n} \left(\mathbf{h}_1^{\text{student}}(i) - \mathbf{h}_1^{\text{teacher}}(i)\right)^2 \tag{2.16}$$

$$\mathcal{L}_{\text{task}} = \frac{1}{\max_t(\mathcal{L}_{\text{task}}(t))} \cdot \begin{cases} \mathcal{L}_{\text{contrastive}} & \text{for main network} \\ \mathcal{L}_{\text{BCE}} & \text{for second-order network} \end{cases} \tag{2.17}$$

$$\mathcal{L}_{\text{continual}} = \lambda_{\text{task}}\mathcal{L}_{\text{task}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} + \lambda_{\text{feature}}\mathcal{L}_{\text{feature}} \tag{2.18}$$

Where:

θ_k = Parameters of the current network (layer k)

$\mathbf{h}_1^{\text{network}}$ = Hidden states of the current network

$\lambda_{\text{task}} + \lambda_{\text{reg}} + \lambda_{\text{feature}} = 1$ (normalized weights)

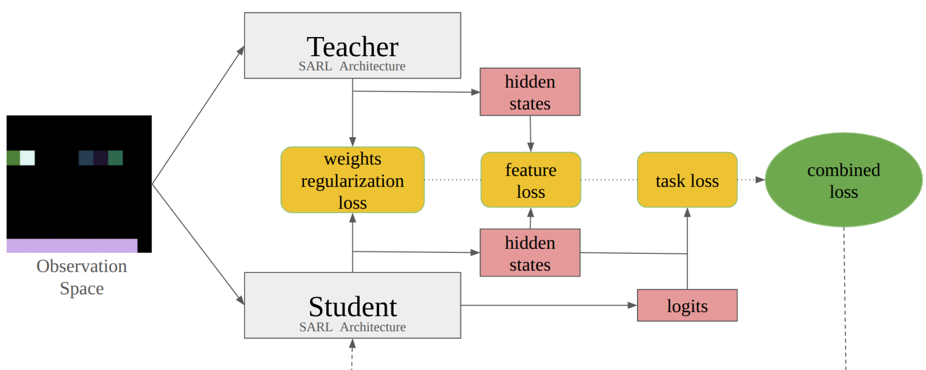

Figure 2.5: Continual learning architecture. The system consists of Teacher-Student: The Teacher network containing frozen weights from the previously trained task, and the Student network actively learning the current task. The loss is the weighted sum of 3 normalized losses: weights regularization, features, and task.

2.4 Experimental Set Up

We empirically select hyperparameters for each of each of the environments (a complete list is provided in Appendix B). We investigate the effect of MAPS using six distinct settings to better understand how each of the main components of MAPS (cascade model and second-order network) contributes to overall performance. Figure 2.6 provides a high-level representation of the baseline architecture used in all the experiments. It should be noted that for Know-Thyself environments, we employ a simple autoencoder, for SARL a Q-network, while for MARL we employ a GRU.

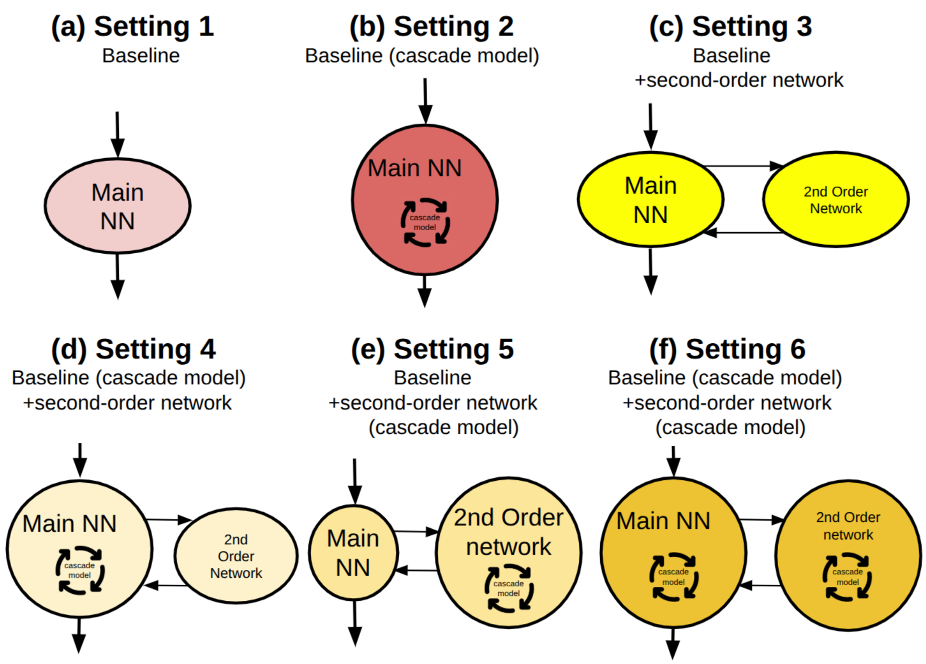

Figure 2.6: High level illustration of the six settings used to analyze the components of MAPS. Each setting shows different combinations of networks with and without cascade dynamics and second-order networks.

To validate our results we follow a set of key metrics as defined in Table 2.3. Z-scores and T-scores were strategically selected as core statistical metrics to ensure robust comparative analysis across different experimental conditions. For the Blindsight and Artificial Grammar Learning (AGL) tasks, Z-scores were employed due to large sample sizes ($n = 500$), which allow the assumption derived by the Central Limit Theorem. Z-scores provide standardized statistical significance measures using the two-sample Z-test (refer to equation 2.19), transforming all metrics to a common scale (mean=0, std=1) that enables meaningful cross-comparison regardless of original metric magnitudes Cochran [2004]. Conversely, for Single Agent Reinforcement

Learning (SARL) and Multi-Agent Reinforcement Learning (MARL) experiments, T-scores (refer to equation 2.20) were utilized due to severe computational limitations that restricted the number of experimental seeds to only 3 (these limitations are present due to the difference in complexity of the tasks). T-tests are specifically designed for small sample scenarios and/or when the population's standard deviation is unknown, accounting for additional uncertainty through the t-distribution rather than the normal distribution JMP Statistical Discovery LLC. We use both Z-scores and T-scores to determine statistical significance of our results versus our baseline model Theevamanohan [2024]. We use a 95 percent confidence level to ensure a low error rate. This methodological approach ensures statistical validity while accommodating the practical constraints of computationally intensive reinforcement learning experiments, ultimately supporting the evaluation of MAPS' effectiveness in enhancing AI adaptability across diverse learning paradigms. We include a comprehensive list of both Z-score and T-score results in Appendix C.

$$Z = \frac{\bar{X}_2 - \bar{X}_1}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{2.19}$$

$$T = \frac{\bar{X}_2 - \bar{X}_1}{s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} = \frac{\bar{X}_2 - \bar{X}_1}{\sqrt{\frac{s_1^2+s_2^2}{2}}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \tag{2.20}$$

Where:
$\bar{X}_1$ and $\bar{X}_2$ are the sample means (baseline and new model respectively)
s_1 and s_2 are the sample standard deviations
n_1 and n_2 are the sample sizes.
s_p is the pooled standard deviation

Both Blindisght and AGL were trained on a local computer with a RTX3070 GPU, while all other environments were trained on the Digital Alliance Canada Clusters (Beluga, Cedar, and Narval) with a variety of GPUs depending on availability (A100, V100, and P100). For reproducibility, it's important to note that when a seed is provided, our code creates a dedicated np.random.RandomState(seed) for deterministic NumPy operations. Without a seed, we use non-deterministic system entropy. It's also to note that all main six settings have an almost equivalent parameter budget as the second-order network is considerably more simple and less computationally expensive compared to the main neural network. On the other hand, training time, energy usage, and carbon footprint by extension, is considerably higher in the models

Environment	Metric	Description
Blindsight / AGL	Accuracy	Pattern recognition accuracy on primary task. Measures main network performance in cognitive tasks.
SARL / SARL + CL / MARL	Rewards	Cumulative episode rewards during training/validation. Evaluates agent performance in reinforcement learning environments.
SARL + CL	Retention	Percentage of previous knowledge retained after learning new tasks. Measures effectiveness against catastrophic forgetting.
MARL	Dist entropy	Action distribution entropy used to evaluate agents convergence. Lower values indicate more stable behavior patterns. This a complementary metric and not the main factor for direct evaluation, but oberservations are still presented as MARL results are preliminary due to high computational costs.
Blindsight / AGL	Z-score	Statistical significance measure using two-sample z-tests to compare means between experimental and baseline conditions. Z-tests are used when sample sizes are large and population variances are known, allowing the use of the central limit theorem. Z-scores transform all metrics to a standardized scale (mean=0, std=1), enabling meaningful cross-comparison regardless of original metric magnitudes.
SARL / MARL	T-score	Statistical significance measure using the two-sample t-test to compare means between experimental and baseline conditions when sample sizes are small or the population's standard deviations is unknown. T-tests account for additional uncertainty in small samples by using the t-distribution instead of normal distribution.
SARL / MARL	Significant	Statistical significance of results at 95% confidence vs Baseline using both z-tests and t-tests as appropriate. Boolean indicator of whether improvements are statistically meaningful based on sample characteristics.

Table 2.3: Validation metrics to evaluate MAPS in each environment

that use cascade model. This is due to the nature of this method that requires several forward passes. Thus, the number of cascade iterations could be tuned for a better performance-computational budget trade-off. These results are presented in Table 2.4. To measure energy usage, we used nvidia-smi through Python's subprocess module to query GPU power consumption. The carbon footprint was calculated using the energy usage as well as the country specific value of kg of CO_2 / kWh (Canada is 0.1 kg of CO_2 / kWh according to Canada Energy Regulator [2024]. All the scripts to replicate the implementations, measure energy usage, run the code in computing clusters, and plot results are available in the GitHub repository[1].

[1] `https://github.com/juandavidvargas19/MAPS_PROJECT/tree/main`

Training Time (Relative % increment vs Baseline)

	Baseline	**Baseline (Cascade*)**	**2nd-Net**	**2nd-Net (Cascade* 1st) - MAPS**	**2nd-Net (Cascade* 2nd)**	**2nd-Net (Cascade* Both)**	**ACB**
Blindisght	0.00%	72.97%	4.81%	107.87%	45.30%	135.22%	-
AGL	0.00%	66.69%	0.29%	53.90%	10.27%	105.69%	-
SARL	0.00%	662.11%	6.95%	893.97%	138.70%	1138.93%	-37.37%
SARL + CL	0.00%	486.67%	-13.33%	420.00%	53.33%	520.00%	-
MARL	0.00%	305.80%	3.90%	328.69%	6.97%	308.82%	-

Number of Parameters (Relative % increment vs Baseline)

	Baseline	**Baseline (Cascade)**	**2nd-Net**	**2nd-Net (Cascade 1st) - MAPS**	**2nd-Net (Cascade 2nd)**	**2nd-Net (Cascade Both)**	**ACB**
Blindisght	0.00%	0.00%	1.26%	1.26%	1.26%	1.26%	-
AGL	0.00%	0.00%	1.28%	1.28%	1.28%	1.28%	-
SARL	0.00%	0.00%	1.55%	1.55%	1.55%	1.55%	0.10%
SARL + CL	0.00%	0.00%	0.76%	0.76%	0.76%	0.76%	-
MARL	0.00%	0.00%	2.17%	2.17%	2.17%	2.17%	-

Energy Usage kWh (Relative % vs Baseline)

	Baseline	**Baseline (Cascade*)**	**2nd-Net**	**2nd-Net (Cascade* 1st) - MAPS**	**2nd-Net (Cascade* 2nd)**	**2nd-Net (Cascade* Both)**	**ACB**
Blindisght	0.00%	83.08%	14.07%	142.71%	75.71%	200.00%	-
AGL	0.00%	71.87%	5.63%	65.98%	20.20%	128.90%	-
SARL	0.00%	269.26%	-13.64%	283.98%	100.00%	476.41%	377.27%
SARL + CL	0.00%	605.72%	7.37%	655.16%	114.33%	796.84%	-
MARL	0.00%	342.71%	5.24%	428.93%	15.75%	413.94%	-

Carbon Emissions - kg CO_2 / 1M Episodes

	Baseline	**Baseline (Cascade)**	**2nd-Net**	**2nd-Net (Cascade 1st) - MAPS**	**2nd-Net (Cascade 2nd)**	**2nd-Net (Cascade Both)**	**ACB**
Blindisght	0.04	0.08	0.05	0.10	0.07	0.13	-
AGL	0.04	0.06	0.04	0.06	0.04	0.08	-
SARL	0.13	0.48	0.11	0.50	0.26	0.75	0.62
SARL + CL	6.88	48.53	7.38	51.93	14.74	61.67	-
MARL	149.21	660.59	157.03	789.25	172.72	766.87	-

Table 2.4: Relative training time, number of parameters, energy usage, and carbon emissions compared to baseline across different model configurations. Values show percentage increases relative to baseline performance. Training time measured as relative percentage increment, energy usage in kWh, carbon emissions in kg CO2 per 1M episodes.

2.5 Results

2.5.1 Blindisght and Artificial Grammar Learning (Know-Thyself environments)

For blindsight, we train our networks using a combination of simple patterns that contain: 1) random noise patterns, and 2) patterns with a single stimulus representing the blindsight phenomenon (This is referred as suprathreshold patterns in A. Pasquali and Cleeremans [2010], refer to Appendix A for additional information). To prevent overfitting, new patterns are generated for each epoch. Table C.1 compares the proposed settings outlined in Figure 2.6. It's important to note that we are focusing on suprathreshold results (the results shown in the table), which is thought to be the only case for which metacognition should be beneficial Weiskrantz et al. [1974]. For blindsight, we observe superior performance on the model using MAPS (second-order network + cascade model in the main network). We compare our baseline (Setting-1), with MAPS (setting-4), obtaining a Z-score of 9.01, meaning MAPS performance is superior and is statistically significant. However, we also see a similar overperformance in other settings (namely 2 and 6), with the three of them having similar overperformance over the baseline and with the common characteristic of using cascade model in the main task network.This observation may suggest that for simple tasks as blindsight, the superior performance of MAPS is primarily driven by the benefits of the cascade model.

For AGL, we pre-train the model, save the weights of the second-order network, and disable backpropagation through it during training. Randomly generated strings are used for pre-training, grammar A for training, and a mix of grammar A and grammar B for testing. Grammar strings are defined as per Persaud et al. [2007], and we follow the data proportions outlined by Pasquali A. Pasquali and Cleeremans [2010]. We employ two training schemes: high awareness of the rules (training over 12 epochs) and low awareness (3 epochs). Our results demonstrate improvement in both scenarios when using MAPS. We observe statistically significant Z-scores of 8.20 and 15.7 for high and low consciousness respectively. Additionally, for the low awareness case, all settings show significant improvement compared to the autoencoder-only model, including the setting with a second-order network and no cascade model. This supports the hypothesis that metacognition or a second-order network may be particularly valuable in simple environments with limited training regimes. Alternatively, we hypothesize that the positive effect on the main task when

			Main Task		Wagering	
Blindsight	**2nd Net**	**Cascade**	**Accuracy**	**Z-score (Significant)**	**Accuracy**	**Z-score**
Setting-1 (Baseline)	No	No	0.95 ± 0.03		0.50 ± 0.05	
Setting-2	No	1st Net	0.97 ± 0.02	**8.50 (Yes)**	0.50 ± 0.05	0.45 (No)
Setting-3	Yes	No	0.96 ± 0.03	0.77 (No)	0.86 ± 0.03	128.1 (Yes)
Setting-4 (MAPS)	Yes	1st Net	0.97 ± 0.02	**9.01 (Yes)**	0.85 ± 0.04	121.2 (Yes)
Setting-5	Yes	2nd Net	0.96 ± 0.03	0.15 (No)	0.87 ± 0.04	126.7 (Yes)
Setting-6	Yes	Both	0.97 ± 0.02	**8.6 (Yes)**	0.86 ± 0.04	124.5 (Yes)
AGL- High Awareness	**2nd Net**	**Cascade**	**Accuracy**	**Z-score (Significant)**	**Accuracy**	**Z-score**
Setting-1 (Baseline)	No	No	0.63 ± 0.05		0.38 ± 0.07	
Setting-2	No	1st Net	0.64 ± 0.04	**6.38 (Yes)**	0.39 ± 0.09	1.10 (No)
Setting-3	Yes	No	0.64 ± 0.04	1.61 (No)	0.59 ± 0.06	45.9 (Yes)
Setting-4 (MAPS)	Yes	1st Net	0.66 ± 0.05	**8.20 (Yes)**	0.58 ± 0.06	43.3 (Yes)
Setting-5	Yes	2nd Net	0.63 ± 0.04	1.09 (No)	0.61 ± 0.06	48.7 (Yes)
Setting-6	Yes	Both	0.65 ± 0.04	**7.88 (Yes)**	0.58 ± 0.06	41.0 (Yes)
AGL- Low Awareness	**2nd Net**	**Cascade**	**Accuracy**	**Z-score (Significant)**	**Accuracy**	**Z-score**
Setting-1 (Baseline)	No	No	0.54 ± 0.08		0.14 ± 0.07	
Setting-2	No	1st Net	0.61 ± 0.07	**13.3 (Yes)**	0.17 ± 0.07	6.25 (Yes)
Setting-3	Yes	No	0.57 ± 0.07	**4.2 (Yes)**	0.83 ± 0.07	143.9 (Yes)
Setting-4 (MAPS)	Yes	1st Net	0.62 ± 0.07	**15.7 (Yes)**	0.82 ± 0.07	137.5 (Yes)
Setting-5	Yes	2nd Net	0.56 ± 0.07	**2.3 (Yes)**	0.87 ± 0.07	150.8 (Yes)
Setting-6	Yes	Both	0.62 ± 0.06	**15.0 (Yes)**	0.82 ± 0.07	141.1 (Yes)

Table 2.5: Accuracy, Z-score, and Significant Results for Main Task and Wagering (Know Thyself environments). Z-score is a statistical measure that quantifies the distance between a data point and the mean of a dataset. We use 95 percent confidence to establish statistical significance. We use a total of N = 500 seeds for each setting. 2nd Net refers to the presence (or not) of a second-order network.

using a second-order network is more pronounced when the task achieves a sufficiently high level of confidence relative to an untrained case. For instance, we observe that the Z-score is half an order of magnitude greater for the low awareness case (137.5 for MAPS) compared to the high awareness case (43.3 for MAPS). This limitation appears to be mitigated by the improved information flow provided by the cascade model.

2.5.2 Single agent reinforcement learning (MinAtar environments)

In MinAtar, we evaluate MAPS across five games: Space Invaders, Breakout, Seaquest, Asterix, and Freeway. We used six experimental configurations to assess both the complete MAPS framework and its individual components (second-order network and cascade model). Training occurs over 1 million steps across 3 seeds per configuration. Our results demonstrate that MAPS consistently outperforms baseline DQN, with particularly pronounced benefits in environments featuring complex multi-agent interactions and dynamic obstacle avoidance.

The effectiveness of MAPS correlates directly with task complexity and environmental interaction density. In Seaquest, which requires managing oxygen levels, enemy avoidance, diver rescue, and strategic surfacing decisions, MAPS achieves a validation T-score of 8.53 against the DQN baseline (refer to Table C.2). Space Invaders, involving alien pattern recognition and bullet coordination, shows a T-score improvement of 5.05, while Asterix, featuring enemy and treasure spawning dynamics, demonstrates a 1.62 improvement (2.64 when also using cascade model in the second-order network). Additionally, even in environments with simpler dynamics, as breakout, MAPS exhibit a 4.55 T-score. However, in breakout, it's to note that in actual learning curves, the training behavior looks very similar among our 6 initial settings (refer to Figure 2.7).

The learning dynamics reveal critical insights about component necessity. In Seaquest, both DQN baseline and single-component implementations (DQN + cascade model OR DQN + second-order network) exhibit ineffective learning (Figure 2.7). Only the combination of second-order network with cascade model enables sustained learning from early training stages. This behavior may suggest that complex multi-objective tasks requiring simultaneous resource management and threat assessment benefit from the enhanced information processing and metacognitive capabilities that emerge from the MAPS components integration.

In addition, we compared our results with the top-performing model presented in the original MinAtar paper (**Actor-critic baseline** with trace decay parameter = 0.8) Young and Tian [2019]. The Actor-critic baseline (ACB) combines 2 components: an actor, which decides what actions to take, and a critic, which evaluates the quality of those actions. We show these results in Table C.2, and in Figure 2.7. ACB shows improved performance against our baseline in 2 of the 5 scenarios. However, its validation T-score is still in all cases lower than MAPS. ACB has a similar concept to self-evaluation, however, these results show the potential for MAPS to produce out-of-distribution performance, likely linked to the introduction of inductive biases inspired by higher-level cognition both with the second-order network and the cascade model.

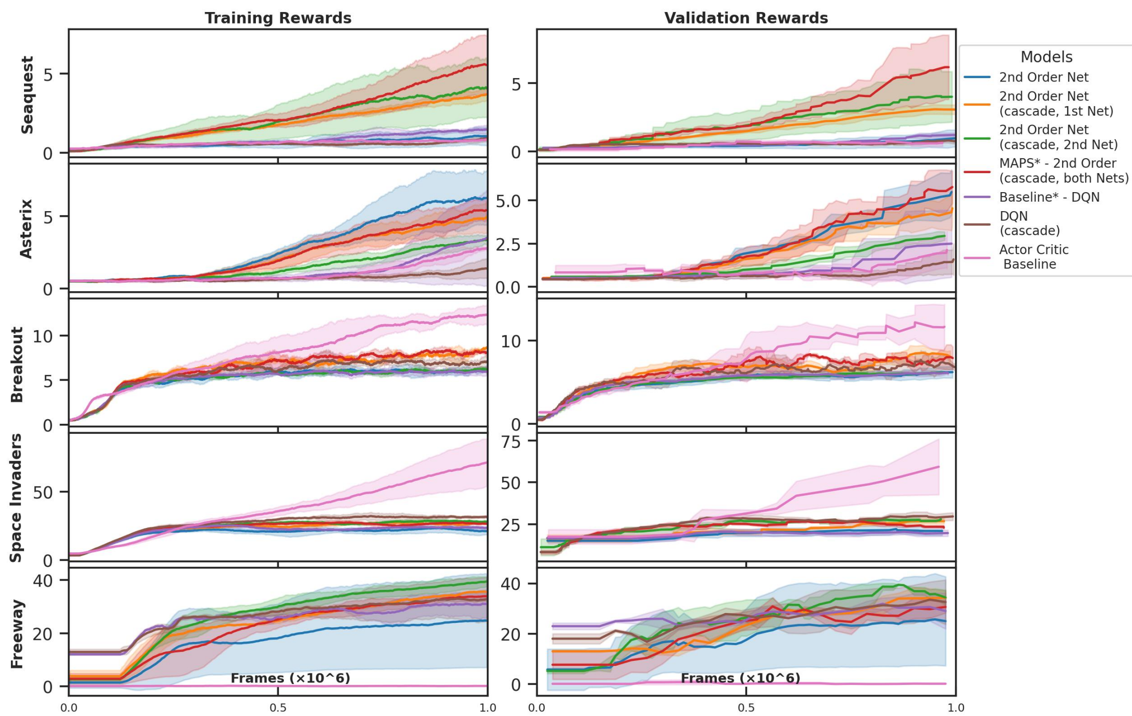

Figure 2.7: Training (left) and validation rewards (right) plots for SARL across different game environments. Y-axis shows cumulative episode rewards, X-axis shows training frames. Each subplot represents a different Atari game (Seaquest, Asterix, Breakout, Space Invaders, Freeway). Lines represent different model configurations (N = 3 seeds per setting), with shaded regions indicating standard error.

Seaquest	2nd-Net	Training Cascade	Rewards	T-score (Significant)	Validation Rewards	T-score (Significant)
Setting-1 (Baseline)	No	No	1.48 ± 0.29		1.21 ± 0.16	
Setting-2	No	1st Net	0.90 ± 0.21	-2.81 (Yes)	0.76 ± 0.19	-3.14 (Yes)
Setting-3	Yes	No	1.04 ± 0.57	-1.19 (No)	0.97 ± 0.61	-0.66 (No)
Setting-4 (MAPS)	Yes	1st Net	3.71 ± 0.40	**7.82 (Yes)**	3.06 ± 0.34	**8.53 (Yes)**
Setting-5	Yes	2nd Net	4.10 ± 1.86	**2.41 (Yes)**	3.99 ± 1.84	**2.61 (Yes)**
Setting-6	Yes	Both	5.56 ± 1.85	**3.77 (Yes)**	6.15 ± 2.34	**3.65 (Yes)**
Setting-7 (ACB)	No	No	0.81 ± 0.03	-3.98 (Yes)	0.63 ± 0.26	-3.29 (Yes)
Asterix						
Setting-1 (Baseline)	No	No	3.49 ± 3.32		2.49 ± 1.94	
Setting-2	No	1st Net	1.38 ± 0.67	-1.08 (No)	1.59 ± 0.90	-0.73 (No)
Setting-3	Yes	No	6.27 ± 1.87	1.26 (No)	5.48 ± 1.30	**2.22 (Yes)**
Setting-4 (MAPS)	Yes	1st Net	4.95 ± 1.05	0.73 (No)	4.54 ± 1.01	1.62 (No)
Setting-5	Yes	2nd Net	3.40 ± 0.30	-0.05 (No)	2.94 ± 0.25	0.40 (No)
Setting-6	Yes	Both	5.42 ± 0.69	0.99 (No)	5.77 ± 0.94	**2.64 (Yes)**
Setting-7 (ACB)	No	No	2.75 ± 0.09	-0.39 (No)	2.13 ± 1.07	-0.28 (No)
Breakout						
Setting-1 (Baseline)	No	No	5.96 ± 0.43		6.10 ± 0.21	
Setting-2	No	1st Net	7.08 ± 0.65	**2.49 (Yes)**	6.84 ± 0.41	**2.78 (Yes)**
Setting-3	Yes	No	6.12 ± 0.51	0.42 (No)	6.22 ± 0.69	0.29 (No)
Setting-4 (MAPS)	Yes	1st Net	8.54 ± 0.07	**10.26 (Yes)**	8.07 ± 0.72	**4.55 (Yes)**
Setting-5	Yes	2nd Net	6.23 ± 0.14	1.03 (No)	6.16 ± 0.05	0.48 (No)
Setting-6	Yes	Both	8.08 ± 0.24	**7.46 (Yes)**	7.91 ± 1.36	**2.28 (Yes)**
Setting-7 (ACB)	No	No	12.36 ± 1.13	**9.17 (Yes)**	11.67 ± 2.70	**3.56 (Yes)**
Space Invaders						
Setting-1 (Baseline)	No	No	23.62 ± 0.86		19.71 ± 1.84	
Setting-2	No	1st Net	31.52 ± 1.01	**10.31 (Yes)**	29.62 ± 2.28	**5.86 (Yes)**
Setting-3	Yes	No	21.14 ± 2.41	-1.68 (No)	20.95 ± 2.60	0.67 (No)
Setting-4 (MAPS)	Yes	1st Net	26.84 ± 0.11	**6.43 (Yes)**	26.80 ± 1.59	**5.05 (Yes)**
Setting-5	Yes	2nd Net	28.11 ± 0.24	**8.71 (Yes)**	27.95 ± 1.10	**6.66 (Yes)**
Setting-6	Yes	Both	26.57 ± 0.72	**4.56 (Yes)**	22.97 ± 0.89	**2.76 (Yes)**
Setting-7 (ACB)	No	No	71.50 ± 17.38	**4.77 (Yes)**	59.26 ± 16.61	**4.10 (Yes)**
Freeway						
Setting-1 (Baseline)	No	No	30.96 ± 5.35		29.03 ± 7.12	
Setting-2	No	1st Net	32.77 ± 8.23	0.32 (No)	32.57 ± 8.82	0.54 (No)
Setting-3	Yes	No	24.72 ± 17.58	-0.59 (No)	25.00 ± 17.68	-0.37 (No)
Setting-4 (MAPS)	Yes	1st Net	35.53 ± 0.08	1.48 (No)	34.20 ± 2.83	1.17 (No)
Setting-5	Yes	2nd Net	39.41 ± 1.89	**2.58 (Yes)**	34.40 ± 2.97	1.21 (No)
Setting-6	Yes	Both	33.84 ± 2.03	0.87 (No)	30.63 ± 2.36	0.37 (No)
Setting-7 (ACB)	No	No	0.19 ± 0.04	-9.96 (Yes)	0.17 ± 0.05	-7.02 (Yes)

Table 2.6: Training and validation rewards, T-score, and significant results for SARL (MinAtar). T-score is a statistical measure that quantifies the distance between a data point and the mean of a dataset. We use 95 percent confidence to establish statistical significance.

2.5.3 Multi agent reinforcement learning (Melting Pot 2.0 environments)

In MARL settings, we conducted preliminary tests to evaluate the potential benefits of using a MAPS in both cooperative and competitive scenarios. We focused on four environments and benchmarked performance against the leading model presented by Agapiou et al. [2023]. The agents were trained for 300,000 steps on three seeds, due to computational constraints (refer to Table 2.4). Our findings revealed that MAPS produced statistical significant results in 2 of the 4 tested environments: commons harvest partnership (T-score = 7.56), and commons harvest closed (T-score = 7.72)- refer to Table C.3. These 2 environments are the lower complexity ones, as the environments were designed to represent fewer multi-agents concepts (refer to Appendix A). For reference, harvest partnership represents 8 multi-agent concepts, harvest closed = 5, chemistry = 11, and territory inside out = 14 Agapiou et al. [2023]. These may point out to 2 possibilities: 1) training was insufficient to properly show MAPS benefit in environments with very complex social dynamics, or 2) MAPS' current architecture can still not harness metacognition's full potential in very complex social dynamics and may be limited to social environments with limited trade-offs. Either of these cases, it's still an open problem that needs further exploration.

MARL also presents a similar result to SARL, where is harvest partnership only when combining a second-order network and cascade model is that effective learning occurs. On the other hand, we introduced a new variable for comparison, action distribution entropy, which diminishes as the behavior of agents become more stable. We see in Figure 2.8 that in all cases MAPS has a faster convergence in terms of agents' stability. This, however, seems to be linked to the cascade model, and may also be controlled when optimizing the number of cascade iterations as to balance exploration and exploitation.

On the other hand, we compare our model against the top-performing model presented in the original Melting Pot 2.0 paper. The top-performing model is ACB Agapiou et al. [2023]. ACB's performance is statistically significant for 1 of the 4 environments we compare in (harvest closed, T-score = 3.21). However, its performance lags agains MAPS for both harvest closed and harvest partnership, and in the remaining two environments produces non-significant results against our baseline. These results highlight yet again the potential for MAPS to produce out-of-distribution performance, likely linked to the introduction of inductive biases inspired by higher-level cognition both with the second-order network and the cascade model. However, in this case, the results are mostly preliminary because of the low number of steps trained due to computational limitations, thus requiring further exploration.

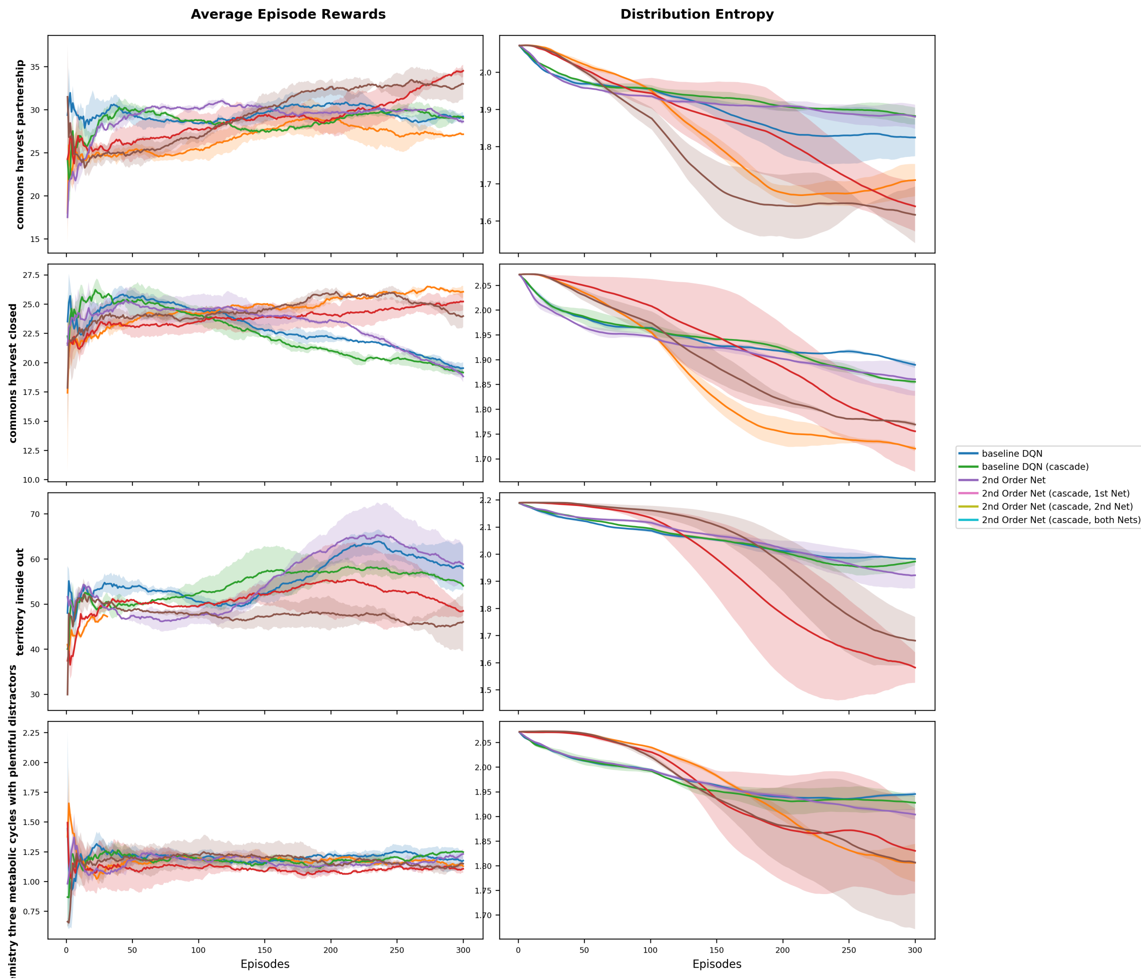

Figure 2.8: Training rewards (left) and action distribution entropy (right) plots for MARL across different environments. Left panels show average episode rewards over training episodes, right panels show action distribution entropy (lower values indicate more deterministic behavior). Lines represent different model configurations (N = 3 seeds per setting), with shaded regions indicating standard error.

		Training			Dist	Entropy
Harvest Partnership	**2nd-Net**	**Cascade**	**Rewards**	**T-score (Significant)**	**Value**	**T-score (Significant)**
Setting-1 (Baseline)	No	No	29.21 ± 0.72		1.82 ± 0.07	
Setting-2	No	1st Net	27.15 ± 0.04	-4.95 (Yes)	1.71 ± 0.06	**-2.07 (Yes)**
Setting-3	Yes	No	29.03 ± 0.29	-0.40 (No)	1.88 ± 0.03	1.36 (No)
Setting-4 (MAPS)	Yes	1st Net	34.52 ± 0.98	**7.56 (Yes)**	1.64 ± 0.10	**-2.55 (Yes)**
Setting-5	Yes	2nd Net	28.63 ± 0.36	-1.25 (No)	1.88 ± 0.05	1.21 (No)
Setting-6	Yes	Both	33.01 ± 2.57	**2.47 (Yes)**	1.62 ± 0.11	**-2.66 (Yes)**
Setting-7 (ACB)	No	No	24.58 ± 10.56	-0.76 (No)	NA	NA
Harvest Closed						
Setting-1 (Baseline)	No	No	19.52 ± 0.71		1.89 ± 0.01	
Setting-2	No	1st Net	26.05 ± 0.71	**11.26 (Yes)**	1.72 ± 0.01	**-20.82 (Yes)**
Setting-3	Yes	No	19.15 ± 0.42	-0.78 (No)	1.86 ± 0.01	**-3.67 (Yes)**
Setting-4 (MAPS)	Yes	1st Net	25.21 ± 1.06	**7.72 (Yes)**	1.76 ± 0.12	-1.87 (No)
Setting-5	Yes	2nd Net	18.81 ± 0.68	-1.25 (No)	1.86 ± 0.05	-1.02 (No)
Setting-6	Yes	Both	23.97 ± 1.30	**5.20 (Yes)**	1.77 ± 0.01	**-14.70 (Yes)**
Setting-7 (ACB)	No	No	39.39 ± 10.69	**3.21 (Yes)**	NA	NA
Chemistry						
Setting-1 (Baseline)	No	No	1.18 ± 0.09		1.95 ± 0.01	
Setting-2	No	1st Net	1.13 ± 0.01	-0.96 (No)	1.81 ± 0.05	**-4.76 (Yes)**
Setting-3	Yes	No	1.25 ± 0.01	1.34 (No)	1.93 ± 0.02	-1.55 (No)
Setting-4 (MAPS)	Yes	1st Net	1.11 ± 0.05	-1.18 (No)	1.83 ± 0.12	-1.73 (No)
Setting-5	Yes	2nd Net	1.23 ± 0.06	0.80 (No)	1.90 ± 0.01	**-6.12 (Yes)**
Setting-6	Yes	Both	1.15 ± 0.03	-0.55 (No)	1.81 ± 0.19	-1.27 (No)
Setting-7 (ACB)	No	No	1.41 ± 0.60	0.66 (No)	NA	NA
Territory Inside Out						
Setting-1 (Baseline)	No	No	57.94 ± 6.84		1.98 ± 0.00	
Setting-2	No	1st Net	41.98 ± 7.43	-2.74 (Yes)	2.01 ± 0.25	0.21 (No)
Setting-3	Yes	No	54.04 ± 0.34	-0.99 (No)	1.97 ± 0.02	-0.87 (No)
Setting-4 (MAPS)	Yes	1st Net	48.47 ± 1.45	-2.35 (Yes)	1.58 ± 0.08	**-8.66 (Yes)**
Setting-5	Yes	2nd Net	58.83 ± 6.15	0.17 (No)	1.92 ± 0.07	-1.48 (No)
Setting-6	Yes	Both	46.05 ± 9.16	-1.80 (No)	1.68 ± 0.12	**-4.33 (Yes)**
Setting-7 (ACB)	No	No	101.99 ± 38.67	1.94 (No)	NA	NA

Table 2.7: Training rewards, dist entropy results, T-score, and significant results for MARL (Melting Pot 2.0). Dist entropy is the action distribution entropy, which gets lower as the agents reduce their stochastic behavior and start to behave in a more stable manner (dist entropy should reduce as rewards increase). T-score is a statistical measure that quantifies the distance between a data point and the mean of a dataset. We use 95 percent confidence to establish statistical significance.

2.5.4 SARL + Continual Learning (MinAtar environments)

For continual learning, we conducted an extensive search of weights (summing to 1.0) for the three losses that we sum to achieve effective learning of new tasks while preserving knowledge of previous ones. For study the effectiveness of this approach of picking the weights, we did preliminary tests on the single configuration that lead to the higher retention (excluding weight regularization close to 1.0 as this wouldn't make sense for effectively learn new tasks) after training on 1 additional environment (task loss=0.4, weight regularization loss =0.4, feature loss=0.2). It's important to note that superior retention does not necessarily translate to effective training on new tasks. For transfer learning, we found that using a second-order network can manage to retain up to 84.2% of previous knowledge, or a mean of 45.1% +/- 31.1%, which is higher than what is achieved when using a random policy (see Figure 2.9). The results from our exploration of weights for the 3 losses can be seen in Figure 2.10.

For a systematic validation, we trained sequentially for 100,000 steps (due to computational limitations faced when using teacher networks) for each of the 4 environments defined in our curriculum. The primary experiment, shown on Figure 7, utilized the optimal retention parameters identified through exploration. We tested this configuration across all six settings used in previous sections, as shown in the left plot. After evaluating Breakout and Space Invaders following training across different environments, knowledge retention was evident in both cases, notoriously when using a second-order network and cascade model in the second-network. Consistent with our preliminary tests, learning effectiveness diminished substantially after training on two or more additional environments. Notably, our DQN baseline performed at or below random policy levels in most cases, contrasting with the lower forgetting observed when using a second-order network network with cascade model. It's also noteworthy that the behaviour of the tested settings seems to be highly dependent on the selected weights for each of the losses, and thus question the robustness of our approach. While it's notable that in most cases, a lower forgetting vs Baseline is evident, further research needs to be done on how to couple a metacognitive approach to be able to more effectively retain knowledge, as the notion is that the second-order network could, at some point, gain independence of the main task to provide valuable confidence information regardless of the task.

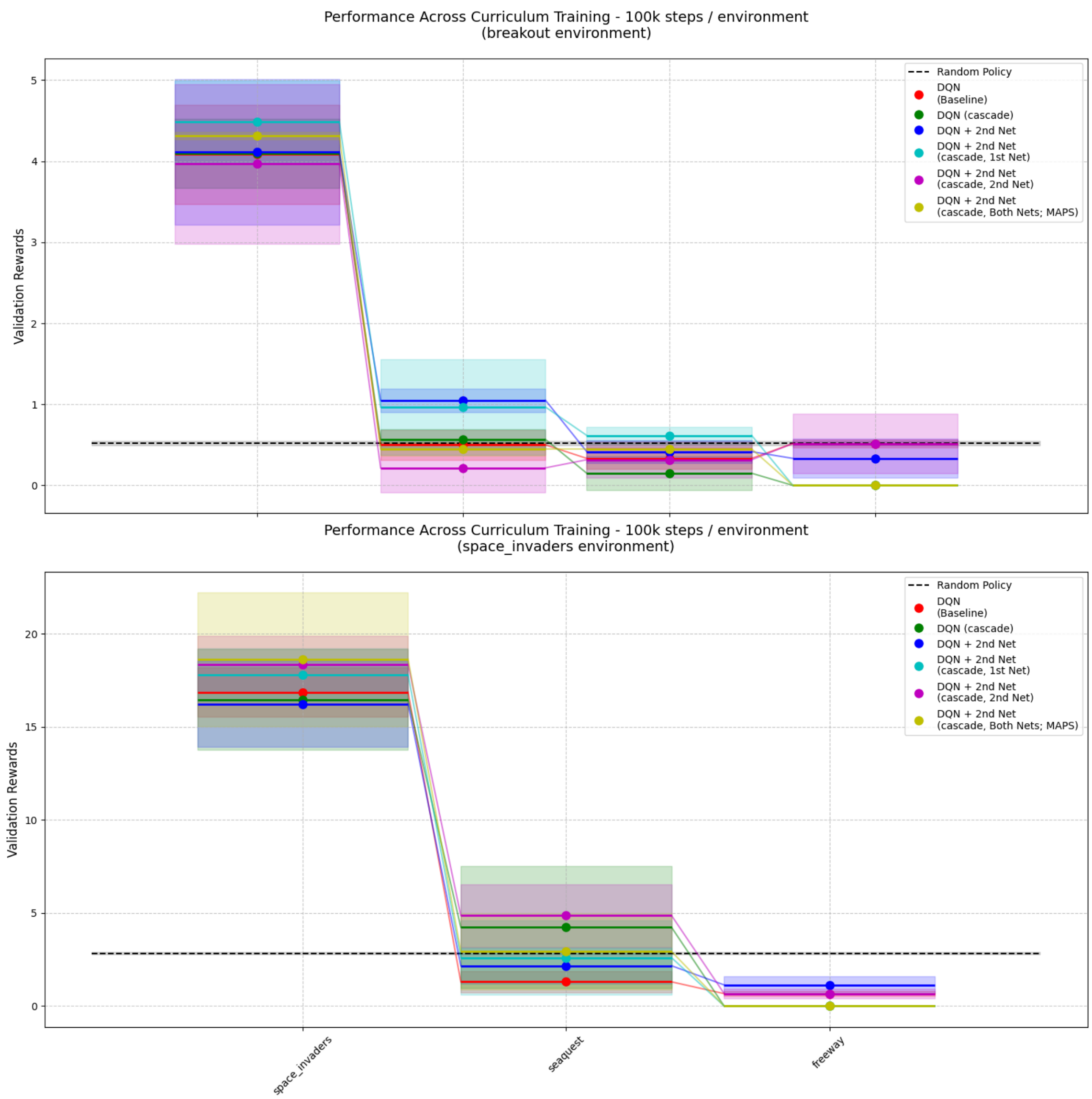

Figure 2.9: Continual learning results showing validation rewards for each environment after sequential training using the continual learning approach. Y-axis shows validation rewards (cumulative episode rewards), X-axis shows environment sequence (100k steps per environment). Top panel: Breakout environment evaluation after training on each sequential environment. Bottom panel: Space Invaders environment evaluation after training on each sequential environment. Colored lines represent different model configurations (DQN variants with cascade dynamics and second-order networks), with shaded regions indicating standard error across multiple runs (N = 3 seeds per setting). Dashed horizontal line indicates random policy performance.

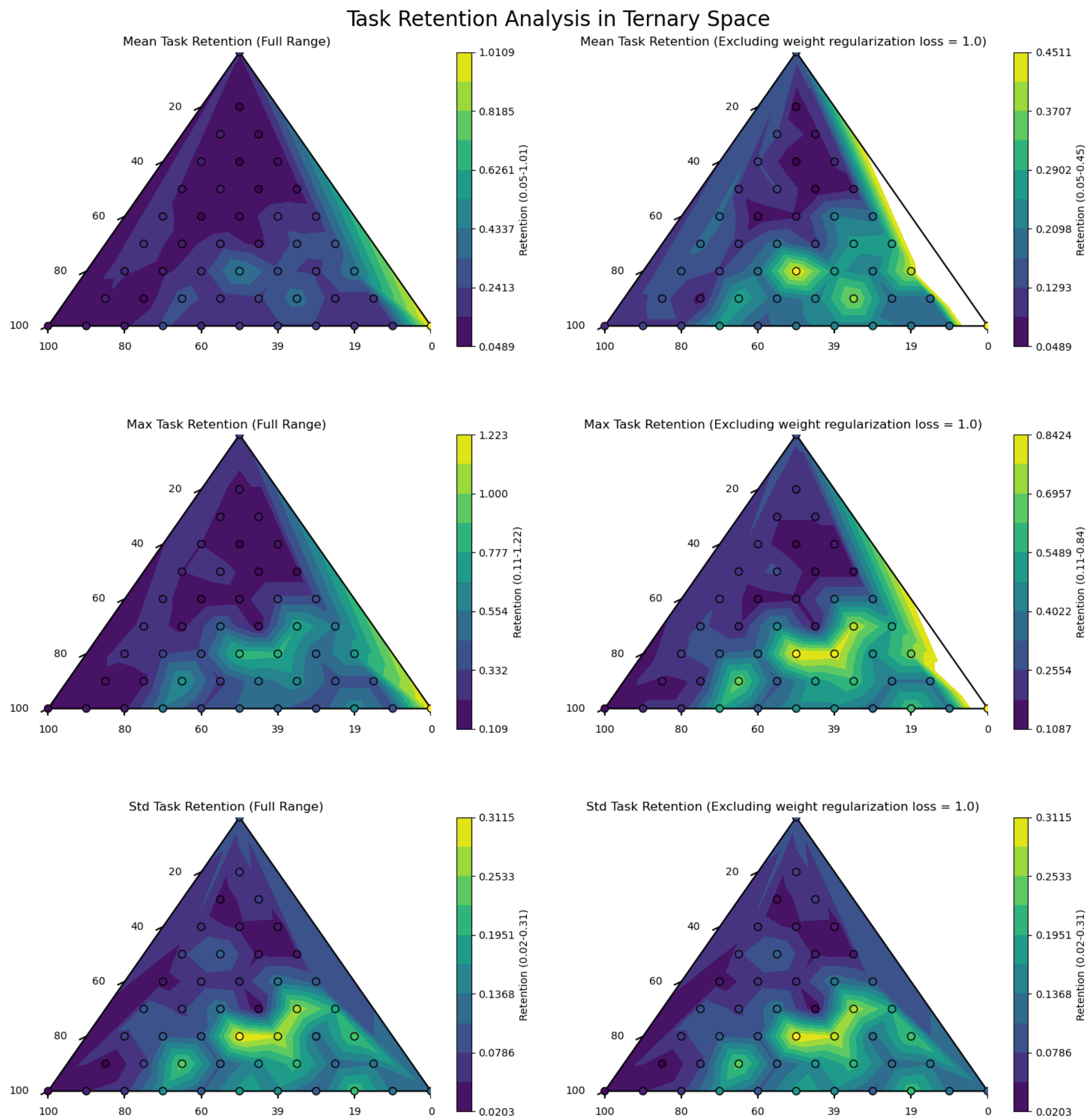

Figure 2.10: Ternary plot representing an extensive search of combinations of the three losses used for continual learning approach. Each triangle shows retention values (fraction of original validation rewards preserved after training on new environment) for different loss weight combinations. Retention represents performance preservation, with higher values (yellow) indicating better task retention. Left column shows full range results, right column excludes weight regularization loss = 1.0. Top row: mean retention, middle row: maximum retention, bottom row: standard deviation of retention. Baseline evaluation used Breakout followed by Space Invaders training (100,000 steps per environment).

2.6 Discussion

Know Thyself: The Role of MAPS in Perceptual Tasks

MAPS significantly improves performance in perceptual tasks, with the cascade model playing a crucial role. Settings using a cascade model show the greatest gains, suggesting that gradual activation smoothing enhances learning. The baseline + cascade model achieves a z-score just below MAPS, indicating that in simple tasks, MAPS' advantage is largely driven by the cascade model.

In the AGL task, MAPS provides statistically significant improvements over the baseline, especially under low-awareness conditions, where the second-order network aids knowledge integration. Similarly, in wagering performance, all MAPS settings outperform the baseline, particularly when confidence assessments are highly accurate. The cascade model further enhances information flow, mitigating limitations in learning.

What we learn from this condition is that MAPS enhances perceptual learning, with the cascade model playing a central role in improving structured learning and information flow.

SARL: Evaluating MAPS in Environments with Uncontrolled Social Dynamics

We evaluate MAPS in single agent reinforcement learning environments using Atari games. In these games, we control a single agent that carries out one or more tasks. However, in some of these games, even if we control a single agent, there are complex social dynamics as there is presence of uncontrollable background agents, obstacles, and enemies. Our results show that MAPS has the capability to outperform the DQN baseline in complex tasks, with the combination of a second-order network and a cascade model proving essential for learning more sophisticated behaviors. Additionally, when compared to the Actor-critic baseline (ACB) from the original MinAtar paper, MAPS demonstrates superior performance across all environments, achieving higher validation T-scores and showing potential for out-of-distribution performance, potentially linked to inductive biases inspired by higher-level cognition.

In Seaquest, MAPS demonstrates its strongest advantages due to the game's multi-layered complexity: players must simultaneously track oxygen depletion, navigate enemy submarines and fish with distinct behaviors, collect divers while managing carrying capacity, and execute strategic surfacing decisions. This environment requires continuous state monitoring, resource optimization, and long-term planning—precisely the capabilities enhanced by MAPS' metacognitive architecture. The validation T-score of 8.53 against DQN baseline reflects this complexity advantage.

Conversely, in Breakout, despite the task's relative simplicity featuring a single moving object (ball), predictable physics-based interactions, and straightforward objective (brick destruction), MAPS still exhibits substantial benefits with a T-score of 4.55. This suggests that even in environments with minimal environmental complexity, the metacognitive processing capabilities of MAPS provide measurable improvements over our baseline.

MARL: Evaluating MAPS in Environments with Controlled Social Dynamics

MAPS was tested in MARL environments with mixed results. The model showed effectiveness in simpler multi-agent scenarios but struggled in more complex social dynamics. Interestingly, MAPS' success appeared inversely correlated with environmental complexity—performing well in harvest environments (5-8 multi-agent concepts) but showing limited benefits in chemistry and territory scenarios (11-14 concepts). This suggests that while MAPS can enhance coordination in constrained social settings, its current architecture may not fully leverage metacognitive advantages in highly complex multi-agent interactions.

The comparison with established baselines reveals MAPS' potential. Against ACB, MAPS demonstrates superior performance in both harvest closed and harvest partnership environments. On other note, we observed faster convergence in agent stability, which seems to be linked to the cascade model. These resuls are still preliminary due to the high computational requirements of MARL environments, and further investigation is required to assess MAPS' role to unlock optimal behaviors in highly social and dynamic environments.

SARL+CL: Evaluating Continual Learning in MAPS

We identified an optimal loss weight distribution for maximization of knowledge retention (other than trivial values of weight regularization close to 1.0): task loss = 0.4, weight regularization = 0.4, feature loss = 0.2. While this configuration improves retention, it does not guarantee effective new learning. A key trade-off emerged—high weight regularization (1.0) preserves past knowledge but impairs adaptation, underscoring the need for balance. In transfer learning, we found that using a second-order network can manage to retain up to 84.2% of previous knowledge, or a mean of 45.1% +/- 31.1%.

Testing these parameters on DQN and DQN + second-order network, we observed lower forgetting in Space Invaders, confirming improved retention. However, after learning two additional environments, performance declined to random policy levels, indicating retention has limits when multiple tasks are introduced.

In summary, DQN alone struggles with retention, often performing at or below random policy levels. In contrast, second-order networks, especially with a cascade model, significantly improve continual learning by preserving prior knowledge.

2.7 Conclusion

This study demonstrates the potential of metacognitive architectures (MAPS) to enhance learning in both perceptual and social environments, particularly in complex and high-variability settings. In perceptual tasks, the cascade model plays a central role, improving structured learning and information flow. In environments with uncontrolled social dynamics (SARL), the combination of a second-order network and a cascade model is essential for mastering sophisticated behaviors, particularly in tasks with dynamic obstacles or interactions. In environments with controlled social dynamics(MARL), MAPS shows promise in scenarios with fewer multi-agents concepts, and shows a tendendency to produce a faster convergence towards more stable agentic behavior, though further testing is required to fully assess its impact on multi-agent reinforcement learning. In continual learning (SARL + CL), second-order networks with a cascade model significantly improve knowledge retention, preventing catastrophic forgetting better than DQN alone. In general, MAPS shows promising results to produce out-of-distribution performance, which is potentially linked to the introduction of inductive biases inspired by higher-level cognition, likely the presence of a second-order network and the cascade model. These findings suggest that metacognitive mechanisms can enhance adaptability, retention, and decision-making in AI systems, paving the way for more intelligent and socially aware reinforcement learning models.

Chapter 3

General Discussion

3.1 Results of the Study

The aim of this thesis is to better understand the effects of metacognitive systems in perceptual scenarios, social environments, and towards minimizing catastrophic forgetting in continual learning. To do so, we built several baselines for each of these environments and implemented MAPS using both a 2nd order network representing the transformation of first order representations, a contrastive loss for providing meaningful information of hidden representations for the confidence assessment of the 2nd order network, and a cascade model to facilitate additional internal dynamics to our networks that would facilitate the information flow.

Our approach builds upon the fundamental understanding of metacognition as the capacity to monitor and regulate one's cognitive processes Flavell [1979]. Metacognition is considered as a crucial component of effective learning, requiring self-reflection on strengths, weaknesses, and methods Murugesan et al. [2022], Jagota et al. [2021]. By implementing a Metacognitive Architecture for Perceptual and Social learning (MAPS), we extend the work of A. Pasquali and Cleeremans [2010] who proposed using a 2nd-order (confidence) network to transform first-order representations.

Furthermore, our exploration of metacognitive systems across both perceptual and social tasks addresses a significant gap in current computational implementations, which often remain limited to basic pattern detection or 'perceptual' tasks Kanai et al. [2024]. By evaluating MAPS across four distinct conditions—perceptual learning (Know Thyself), Single-Agent Reinforcement Learning (SARL, MinAtar), SARL with continual learning (MinAtar), and Multi-Agent Reinforcement Learning (MeltingPot 2.0)—we aim to more comprehensively assess the potential of metacognition for enhancing artificial intelligence systems across different domains.

In this study, we found 3 general benefits of our metacognition implementation: (1) Using MAPS improved the system's performance across a range of cognitive tasks. This supports the idea that self-evaluation and reflective processing can enhance learning and decision-making in AIS. (2) Specifically, using a smooth activation (cascade model) and the 2nd-Net (MAPS), helped accelerate learning by making the metacognitive process more continual and adaptive. The joint use of these components proved especially useful in environments with various obstacles/non-controllable agents. (3) Using MAPS proved to minimize catastrophic forgetting in continual learning for single agent tasks, however this effect was limited to transfer learning (1 additional environment), and the approach proved limited for training 2 or more additional environments. On the other hand, in MARL, MAPS shows promise in scenarios with fewer multi-agents concepts, and shows a tendendency to produce a faster convergence towards more stable agentic behavior.

Below, we discuss the findings for each of the 3 main points considered in this study:

1. to investigate the role of metacognitive systems in perceptual tasks
2. to assess the impact of metacognitive systems on social interactions and complex agentic reinforcement learning settings.
3. to assess the impact of metacognitive systems towards minimizing catastrophic forgetting in continual learning.

3.1.1 MAPS Outperforms a Baseline, and Shows Increased Statistical Significance When Training for Few Iterations in Perceptual Environments

We observe that using MAPS, we obtain statistically significant results for perceptual environments, both in Blindsight and in Artificial grammar learning tasks. We see this effect especially when training for few iterations as in Artificial Grammar Learning (Z-score of 15.7 vs Z-score of 8.20). We also see this behavior to be co-related to the overperformance in the confidence assessment task of each of the networks, as the wagering performance in respect to the baseline is notoriously superior when training for few epochs (low consciousness), in comparison to when we train the same task for higher number of iterations (high consciousness). This is evident with a wagering Z-score of 137.5 for the low training case, compared to 43.3 of Z-score for the high training case (artificial grammar learning task).

3.1.2 MAPS Outperforms a Baseline in Single Agent Environments, and It's Especially Significant in Complex Scenarios with Multiple Obstacles or Background Agents. Also, a Single 2nd Order Network Fails to Consistently Outperform a Baseline in Environments with Multiple Controllable Agents

We observe that using MAPS is also significant in single agent reinforcement learning. For instance, we observe that for seaquest, the most complex of the SARL scenarios, effective learning is only achieved when using both a 2nd order network and cascade model (T-score of 8.53 for MAPS). On the other hand, in environments that would theoretically need the least metacognition, breakout, we can see that the performance is very similar among all settings tested (T-score of 4.55 for MAPS and 2.78 for the cascade model only model, and non-significant T-scores for all other cases). Furthermore, the learning curves show a very similar behavior. This behavior in single agent reinforcement learning may well reflect that our architecture is effectively mimicking a potential model with metacognition capabilities to a certain extent, or at least capabilities related to confidence assessment that would provide essential information to the main task. This positive behavior can also be seen when comparing against the top-performing model in the original MinAtar paper Young and Tian [2019], for which all T-scores results are lower to MAPS.

In MARL, we train each of the tested setting for 300,000 steps due to the large computational resources required to train MAPS in multi-agent reinforcement learning settings. We observe statistically significant results for environments that were designed to represent a fewer number of multi-agent concepts (T-score of 7.56 for harvest partnership, and 7.72 for harvest closed). These results remain even when comparing the T-scores of MAPS with the top-model from the original Meltingpot 2.0 paper Agapiou et al. [2023]. On the other hand, we observed a tendency for models with cascade model to produce a faster agentic stability, thus reducing stochastic behavior, when compared to other models. However, these results are still preliminary and require further experimentation.

3.1.3 MAPS Minimizes Forgetting After Learning a Single New Environment, but Catastrophic Forgetting Is Observed When Trained for 2 or More New Environments

Finally, we evaluate the capability of metacognitive systems to minimize catastrophic forgetting in continual learning. We observe a higher knowledge retention when using a 2nd order network and cascade model in the 2nd network. Notably, our DQN baseline performed at or below random policy levels in most cases, contrasting with the lower forgetting observed when using a 2nd order network with cascade model. It's also to note that while successfully avoiding complete forgetting after training in one new environment, catastrophic forgetting would follow after training for 2 or more new environments. Thus, it's still an open question whether MAPS can't effectively minimize forgetting over an extensive number of new trained environments, or whether it's, but our current framework for continual learning fails to use the full capabilities of a 2nd order network that represents the transformation of 1st order representations.

3.2 Limitations of the Study

This study presents significant results over the use of a metacognitive architecture (MAPS). The results of this study suggest a significance of such architecture in tasks spanning from very simple perceptual tasks (blindsight and AGL) to complex single agent environment with multiple obstacles and background agents. Given the complexity of some of these tasks, this study was greatly limited by computational resources in the following scenarios: (1) studying performance over longer periods of training for complex environments (especially when training using our continual learning approach as well as multi-agent reinforcement learning), (2) increase the number of seeds to have a better generalization of the results presented in this study, (3) increase the training steps for multi-agent reinforcement learning as, when implemented with a cascade model, this configuration is substantially more expensive in computational resources, (4) not sufficient optimization of cascade iterations used as to improve the trade-off between performance and energy usage, and (5) diversity of environments as a result of the limitation in computing resources. On the other hand, this study was limited to study metacognitive architecture using a second-order network, however, there are alternatives to such an approach as the case of interpreting metacognition as the representation of the processes that generate first-order representations Kanai et al. [2024], in contrast to a transformation of first order representations. This approach is also worth to investigate and compare with MAPS over the same settings and environments present in this study.

3.3 Future Perspectives

The results of this study present significant answers about the potential effect of a metacognitive architecture for a wide range of settings and environments. However, there are still open questions that are a result of this study. To move forward, some points that are left unattended and that would be relevant to the study of metacognition in AI architectures are:

- Studying potential emerging behaviors when using a metacognitive network and training for a substantial number of epochs (grokking).
- Studying, for different environments, the scalability (number of parameters) of our metacognitive architecture in comparison to a model that doesn't implement MAPS.
- Increase the span of trained episodes of MAPS in multi-agent reinforcement learning environments.
- Extending the study of the effect of MAPS towards catastrophic forgetting in continual learning approach. Especially when increasing or decreasing the number of epochs to train in every environment.
- Extending the same experiments proposed in this study to an alternative implementation of metacognition in AI architectures, for example metacognition as a meta-representation of processes Kanai et al. [2024].
- Studying the effect of the cascade model, and the best way to optimize the number of cascade iterations as to get a reasonable trade-off of performance and computation.
- Extend the number and diversity of environments where MAPS is tested, especially in SARL, MARL, and Continual Learning. This should be done with a special focus on covering a wide range of agentic concepts.

3.4 Conclusion

In conclusion, MAPS as a potential architecture for harnessing metacognition is plausible but still farfetched, although it can bring some insights into the value of confidence judgements even in very complex environments. For instance, this study demonstrates that incorporating a metacognitive second-order network into AI systems improves performance in perceptual, single-agent and multi-agent reinforcement learning tasks, with notable learning convergence using a cascade model. We also observe very compelling evidence of both the effectiveness of MAPS for social systems with seaquest, and commons harvest partnership, and the conceptual correspondence with breakout, as it is an environment with little complexity outside the main objective of breaking as high a number of bricks as possible. While benefits in multi-agent settings were limited, MAPS shows promise in scenarios with fewer multi-agents concepts, and shows a tendency to produce a faster convergence towards more stable agentic behavior, though further testing is required to fully assess its impact on multi-agent reinforcement learning. In continual learning, MAPS demonstrates an early capability to retain knowledge from previously learned tasks, however this is still insufficient as catastrophic still is observed when training in 2 additional environments or more. In general, MAPS shows promising results to produce out-of-distribution performance, which is potentially linked to the introduction of inductive biases inspired by higher-level cognition, likely the presence of a second-order network and the cascade model. These findings underscore the value of metacognition in advancing adaptive and socially aware AI systems.

Bibliography

B. Timmermans A. Pasquali and A. Cleeremans. Know thyself: Metacognitive networks and measures of consciousness. *Cognition*, 117:182–190, 2010.

J.P. Agapiou, A.S. Vezhnevets, E.A. Duéñez-Guzmán, J. Matyas, Y. Mao, P. Sunehag, R. Köster, U. Madhushani, K. Kopparapu, R. Comanescu, D.J. Strouse, M.B. Johanson, S. Singh, J. Haas, I. Mordatch, D. Mobbs, and J.Z. Leibo. Melting pot 2.0. *arXiv preprint arXiv:2211.13746*, 2023.

M. L. Anderson, T. Oates, W. Chong, and D. Perlis. The metacognitive loop I: Enhancing reinforcement learning with metacognitive monitoring and control for improved perturbation tolerance. *Journal of Experimental & Theoretical Artificial Intelligence*, 18(3):387–411, 2006. doi: 10.1080/09528130600926066.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Joshua Bengio. The consciousness prior. Available at `https://doi.org/10.48550/arXiv.1709.08568`, 2017. URL `https://doi.org/10.48550/arXiv.1709.08568`. arXiv preprint arXiv:1709.08568 [cs.LG].

Matthew Botvinick, Sam Ritter, Jane X. Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. Reinforcement learning, fast and slow. *Trends in Cognitive Sciences*, 23(5):408–422, 2019. doi: 10.1016/j.tics.2019.02.006.

R. Brown, H. Lau, and J.E. LeDoux. Understanding the higher-order approach to consciousness. *Trends in Cognitive Sciences*, 23:754–768, 2019.

Canada Energy Regulator. Provincial and territorial energy profiles – canada, 2024. URL https://www.cer-rec.gc.ca/en/data-analysis/energy-markets/provincial-territorial-energy-profiles/provincial-territorial-energy-profiles-canada.html. Date modified: 2024-09-10.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. URL https://doi.org/10.48550/arXiv.2002.05709. ICML 2020.

S. D. Cochran. Two sample z-tests. The University of California, Los Angeles, Statistics Course Material, 2004. URL http://www.stat.ucla.edu/~cochran/stat10/winter/lectures/lect21.html. Accessed: 2025-06-08.

Brendan Conway-Smith and Robert L. West. Toward autonomy: Metacognitive learning for enhanced ai performance. *Proceedings of the AAAI 2024 Spring Symposium Series: Symposium on Human-Like Learning*, 2024a. doi: 10.1609/aaai.ss.v3i1.31270. URL https://doi.org/10.1609/aaai.ss.v3i1.31270.

Brendan Conway-Smith and Robert L. West. Toward autonomy: Metacognitive learning for enhanced ai performance. *Proceedings of the AAAI Symposium Series*, 3(1): 545–546, 2024b. doi: 10.1609/aaaiss.v3i1.31270.

A. Cortese. Metacognitive resources for adaptive learning. *Neuroscience Research*, 178:10–19, 2020.

A. Cortese, B. De Martino, and M. Kawato. The neural and cognitive architecture for learning from a small sample. *Current Opinion in Neurobiology*, 55:133–141, 2019.

A. Cortese, H. Lau, and M. Kawato. Unconscious reinforcement learning of hidden brain states supported by confidence. *Nature Communications*, pages 1–14, 2020.

A. Cortese, A. Yamamoto, M. Hashemzadeh, P. Sepulveda, M. Kawato, and B. De Martino. Value signals guide abstraction during learning. *eLife*, 10, 2021.

S. Dehaene, H. Lau, and Kouider. S. What is consciousness, and could machines have it? *Science*, 358(6362):486–492, 2017.

Zoltan Dienes, Gerry Altmann, Lisa Kwan, and Andrew Goode. Unconscious knowledge of artificial grammars is applied strategically. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, 21(5):1322–1338, 1995. doi: 10.1037/0278-7393.21.5.1322.

J. Dunstone and C.A. Caldwell. Cumulative culture and explicit metacognition: a review of theories, evidence and key predictions. *Palgrave Communications*, 4:1–11, 2018.

E. Feurer, R. Sassu, P. Cimeli, and C. Roebers. Development of meta-representations: Procedural metacognition and the relationship to theory of mind. *Journal of Educational and Developmental Psychology*, 5(1):6–18, 2015.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 2017.

J.H. Flavell. Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American Psychologist*, 34(10):906–911, 1979. doi: 10.1037/0003-066X.34.10.906.

C. D. Frith. The role of metacognition in human social interactions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1599):2213–2223, 2012. doi: 10.1098/rstb.2012.0123.

L. S. Garbayo, D. M. Harris, S. M. Fiore, M. Robinson, and J. D. Kibble. A metacognitive confidence calibration (MCC) tool to help medical students scaffold diagnostic reasoning in decision-making during high-fidelity patient simulations. *Advances in Physiology Education*, 47(1):71–81, 2023. doi: 10.1152/advan.00156.2021.

L. Goupil and S. Kouider. Developing a reflective mind: from core metacognition to explicit self-reflection. *Current Directions in Psychological Science*, 2019.

L. Goupil, M. Romand-Monnier, and S. Kouider. Infants ask for help when they know they don't know. *Proceedings of the National Academy of Sciences of the United States of America*, 113:3492–3496, 2016.

Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478(2266):20210068, 2022. doi: 10.1098/rspa.2021.0068.

C. Heyes. Who knows? metacognitive social learning strategies. *Trends in Cognitive Sciences*, 20:204–213, 2016.

V. Jagota, V. Bhatia, L. Vives, and A.B. Prasad. Ml-pasd: predict autism spectrum disorder by machine learning approach. In *Artificial Intelligence for Accurate Analysis and Detection of Autism Spectrum Disorder*, page 12, 2021.

A. Jain, A. Szot, and J. J. Lim. Generalization to New Actions in Reinforcement Learning. *arXiv*, 2020. doi: 10.48550/arXiv.2011.01928.

J. Jeyaraman, J. N. A. Malaiyappan, R. Ranjan, and S. M. K. Sistla. Advancements in Reinforcement Learning Algorithms for Autonomous Systems. 9(3):1941, 2024. doi: 10.17613/sd0v-he77.

JMP Statistical Discovery LLC. The two-sample t-test. Statistics Knowledge Portal. URL `https://www.jmp.com/en_us/statistics-knowledge-portal/t-test/two-sample-t-test.html`. Accessed: 2025-06-08.

Ryota Kanai, Ryota Takatsuki, and Ippei Fujisawa. Meta-representations as representations of processes. *PsyArXiv*, 2024. doi: 10.31234/osf.io/zg27u. Preprint.

N. Kastel, C. Hesp, K. R. Ridderinkhof, and K. J. Friston. Small steps for mankind: Modeling the emergence of cumulative culture from joint active inference communication. *Frontiers in Neurorobotics*, 16, 2023. doi: 10.3389/fnbot.2022.944986.

R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan. Measuring Catastrophic Forgetting in Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.11651.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

C. Koch and K. Preuschoff. Betting the house on consciousness. *Nature Neuroscience*, 10:140–141, 2007. doi: 10.1038/nn0207-140.

K. R. Koedinger, P. F. Carvalho, R. Liu, and E. A. McLaughlin. An astonishing regularity in student learning rate. *Proceedings of the National Academy of Sciences*, 120(13):e2221311120, 2023. doi: 10.1073/pnas.2221311120.

A. Lak, E. Hueske, J. Hirokawa, P. Masset, T. Ott, A. E, T.H. Donner, M. Carandini, S. Tonegawa, N. Uchida, and A. Kepecs. Reinforcement biases subsequent perceptual decisions when confidence is low: a widespread behavioral phenomenon. *eLife*, 9:e49834, 2020.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018. doi: 10.1109/TPAMI.2017.2782686.

S. H. Lincoln, L. T. Germine, P. Mair, and C. I. Hooker. Simulation and social behavior: An fMRI study of neural processing during simulation in individuals with and without risk for psychosis. *Social Cognitive and Affective Neuroscience*, 15(2):165–174, 2020. doi: 10.1093/scan/nsaa047.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems*, 30, 2017.

X. Lu, C. Murawski, P. Bossaerts, and S. Suzuki. Estimating self-performance when making complex decisions. *Scientific Reports*, 15(1):3203, 2025. doi: 10.1038/s41598-025-87601-8.

James L. McClelland, David E. Rumelhart, Jerome Feldman, and Patrick Hayes. *Explorations in Parallel Distributed Processing - Macintosh version: A Handbook of Models, Programs, and Exercises*. Bradford Books. The MIT Press, 1989. ISBN 9780262291279. doi: 10.7551/mitpress/5617.001.0001. URL `https://doi.org/10.7551/mitpress/5617.001.0001`. Includes 2 diskettes for the Macintosh, In Special Collection: CogNet.

J. Morales and H. Lau. Confidence tracks consciousness. In J. Weisberg, editor, *Qualitative consciousness: themes from the philosophy of David Rosenthal*. Cambridge University Press, 2021.

G. Murugesan, T.I. Ahmed, M. Shabaz, J. Bhola, B. Omarov, R. Swaminathan, F. Sammy, and S.A. Sumi. Assessment of mental workload by visual motor activity among control group and patient suffering from depressive disorder. *Computational Intelligence and Neuroscience*, 2022:8555489, 2022.

Kamal K Ndousse, Douglas Eck, Sergey Levine, and Natasha Jaques. Emergent social learning via multi-agent reinforcement learning. In *International conference on machine learning*, pages 7991–8004. PMLR, 2021.

B. Norman and J. Clune. First-Explore, then Exploit: Meta-Learning to Solve Hard Exploration-Exploitation Trade-Offs. *arXiv*, 2024. doi: 10.48550/arXiv.2307.02276.

N. Persaud, P. McLeod, and A. Cowey. Post-decision wagering objectively measures awareness. *Nature Neuroscience*, 10:257–261, 2007. doi: 10.1038/nn1840.

P.M.A. Rabbit. Error correction time without external error signals. *Nature*, 212: 438, 1966.

Y. Rajkumari S. Kala and M. Rana. A deep dive into metacognition: Insightful tool for moral reasoning and emotional maturity. *Neuroscience Informatics*, 2, 2022.

Kristian Sandberg, Bert Timmermans, Morten Overgaard, and Axel Cleeremans. Measuring consciousness: Is one measure better than the other? *Consciousness and Cognition*, 2010. doi: 10.1016/j.concog.2009.12.013.

N. Shea, A. Boldt, D. Bang, N. Yeung, C. Heyes, and C.D. Frith. Supra-personal cognitive control and metacognition. *Trends in Cognitive Sciences*, 18:186–193, 2014.

N. Shea, C. Frith, C. Heyes, D. Bang, and S. Fleming. Knowing ourselves together: The cultural origins of metacognition. *Trends in Cognitive Sciences*, 24:349–362, 2020. Opinion Piece.

S. Sidra and C. Mason. Reconceptualizing AI Literacy: The Importance of Metacognitive Thinking in an Artificial Intelligence (AI)-Enabled Workforce. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pages 1181–1186, 2024. doi: 10.1109/CAI59869.2024.00211.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. doi: 10.1038/nature16961.

T. Sugiyama, N. Schweighofer, and J. Izawa. Reinforcement learning establishes a minimal metacognitive process to monitor and control motor learning performance. *Nature Communications*, 14(1):3988, 2023. doi: 10.1038/s41467-023-39536-9.

Arjun Theevamanohan. T-test vs. z-test: When to use each. DataCamp Tutorial, August 2024. URL `https://www.datacamp.com/tutorial/t-test-vs-z-test`. Accessed: 2025-06-08.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *arXiv preprint arXiv:1509.06461*, 2015. URL `https://doi.org/10.48550/arXiv.1509.06461`. AAAI 2016.

Peter B. Walker, Jonathan J. Haase, Melissa L. Mehalick, Christopher T. Steele, Dale W. Russell, and Ian N. Davidson. Harnessing metacognition for safe and responsible ai. *Technologies*, 13(3):107, 2025. doi: 10.3390/technologies13030107.

L. Weiskrantz, E. K. Warrington, M. D. Sanders, and J. Marshall. Visual capacity in hemianopic field following a restricted occipital ablation. *Brain*, 97:709–728, 1974.

Kenny Young and Tian Tian. Minatar: An atari-inspired testbed for thorough and reproducible reinforcement learning experiments. *arXiv preprint arXiv:1903.03176*, 2019.

C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu. The surprising effectiveness of ppo in cooperative multi-agent games. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

E. Zaroukian. Theory of mind and metareasoning for artificial intelligence: A review. 2022. Submitted for review.

T. Zhang and H. Mo. Reinforcement learning for robot research: A comprehensive review and open issues. *International Journal of Advanced Robotic Systems*, 18(3): 17298814211007305, 2021. doi: 10.1177/17298814211007305.

Appendix A

Additional Environment details

A.1 Blindsight task

Blindsight is a neurological phenomenon where individuals with damage to their primary visual cortex can still respond to visual stimuli without consciously perceiving them.

To study this, we use a simulated dataset that mimics the conditions of blindsight according to A. Pasquali and Cleeremans [2010]. This dataset contains 400 patterns, equally split between two types:

- **Random noise patterns:** These consist of low activations ranging between 0.0 and 0.02.
- **Designed stimulus patterns:** Each pattern includes one unit that shows a higher activation level, varying between 0.0 and 1.0.

This dataset allows us to test hypotheses concerning how sensory processing and network responses adapt under different conditions of visual impairment.
We have three main testing scenarios, each designed to alter the signal-to-noise ratio to simulate different levels of visual impairment:

- **Suprathreshold stimulus condition:** Here, the network is tested against familiar patterns used during training to assess its response to known stimuli.

- **Subthreshold stimulus condition:** This condition slightly increases the noise level, akin to actual blindsight conditions, testing the network's capability to discern subtle signals.

- **Low vision condition:** The intensity of stimuli is decreased to evaluate how well the network performs with significantly reduced sensory input.

A.2 Artificial Grammar Learning Task

In the AGL experiment, Persaud et al. Persaud et al. [2007] demonstrate that participants exposed incidentally to letter strings generated by an artificial grammar perform better than chance on a subsequent, unexpected test where they distinguish between new grammatical and non-grammatical strings. However, they fail to optimize their earnings through wagering. Once participants were informed about the grammar rules, they began to place advantageous wagers (explicit condition).
To simulate this, we utilize artificially generated strings ranging from 3 to 8 letters, classified into three types: randomly generated, grammar A, and grammar B, as defined by Persaud et al.
During training, the networks are exposed to two conditions: explicit and implicit, reflecting the results of implicit learning Dienes et al. [1995]. For the implicit condition (low consciousness), networks are trained for 3 epochs, while for the explicit condition (high consciousness), they are trained for 12 epochs.

A.3 MinAtar

MinAtar provides simplified versions of classic Atari 2600 games, designed specifically for AI agent testing and development. MinAtar offers more accessible and computationally efficient environments for AI research and experimentation Young and Tian [2019]. There are 5 Atari games implemented:

- **Space Invaders:** The player controls a cannon to shoot at aliens that move across and down the screen, with each destroyed alien providing +1 reward and causing the remaining aliens to speed up. Aliens also shoot back at the player, new waves spawn at increased speeds after clearing a wave, and termination occurs when the player is hit by an alien or bullet Young and Tian [2019].

- **Breakout:** The player controls a paddle at the bottom of the screen to bounce a diagonally-traveling ball toward three rows of bricks at the top, earning +1 reward for each brick broken and getting new rows when all are cleared. The ball's direction changes based on which side of the paddle it hits or when it contacts walls and bricks, with game termination occurring when the ball reaches the bottom of the screen Young and Tian [2019].

- **Seaquest:** The player controls a submarine that can fire bullets at enemy submarines and fish, earning +1 reward for each hit while also rescuing divers to fill a progress bar and maintaining oxygen that depletes over time. Oxygen replenishes when surfacing with at least one rescued diver, surfacing with six divers provides additional rewards based on remaining oxygen, and the game ends when hit by enemies, running out of oxygen, or surfacing without divers Young and Tian [2019].

- **Asterix:** The player moves freely in four cardinal directions to collect treasure while avoiding enemies that spawn from the sides, with each treasure providing a +1 reward and enemy contact causing termination. Enemy and treasure movements are indicated by trail channels, and the game's difficulty increases periodically by enhancing the speed and spawn rate of both enemies and treasures Young and Tian [2019].

- **Freeway:** The player moves vertically up and down at a restricted pace (once every 3 frames) to cross a road filled with horizontally-moving cars, earning +1 reward upon reaching the top before being returned to the bottom. When hit by a car, the player returns to the bottom without penalty, car speeds randomize after each successful crossing, and the game terminates after 2500 frames have elapsed Young and Tian [2019].

A.4 Meltingpot

The Melting Pot Suite provides a comprehensive framework for generating test scenarios that assess an agent population's ability to generalize cooperative behavior in new situations. It offers up to 50 distinct training and testing environments. The test scenarios combine novel background populations of agents and include a variety of substrates, such as classic social dilemmas like the Prisoner's Dilemma, as well as complex mixed-motive coordination games. In our experiments, we selected four environments based on the coefficient of variation among the models tested in Agapiou et al. [2023]. This value was calculated for the 37 non-zero-sum environments out of the 50 available (see Figure A.1. We chose the three environments with the lowest variability and the environment with the highest positive variability.

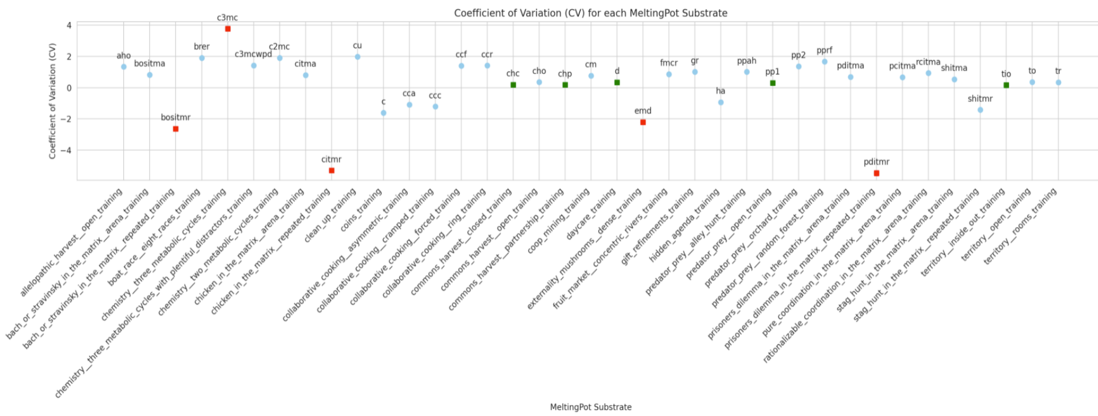

Figure A.1: Variability among Melting Pot environments according to the experimentation in Agapiou et al. [2023].

Our tested environments are: Commons Harvest Closed, Commons Harvest Partnership, Chemistry Three Metabolic Cycles with Plentiful Distractors, and Territory Inside Out. A short description is provided below:

- **Commons Harvest Closed:** Apples are dispersed and can be consumed by agents. Additionally, apples have a probability at every step to regrow, which depends on the number of nearby apples: 0.0025 when there are three or more apples, 0.005 for two, 0.001 if there is one, and 0 otherwise. Thus, agents need to exercise restraint in consuming all apples in a batch to ensure the long-term regrowth of apples. Even though it is not beneficial to consume the last apple, agents are incentivized to do so to prevent other agents from consuming it. In this closed variant, there are rooms full of apples, promoting agents to defend them and minimize the probability of other agents harvesting the full patch of apples Agapiou et al. [2023].
- **Commons Harvest Partnership:** Similar to the Commons Harvest Closed environment, this variant still has rooms filled with apples. However, it requires two agents to protect a room, thus promoting the development of cooperative behavior and a mutually sustainable situationAgapiou et al. [2023].
- **Chemistry Three Metabolic Cycles with Plentiful Distractors:** In this

setting, a set of agents work to generate mutual benefits from metabolic reactions defined by a predefined graph. These reactions occur stochastically when reactants are in close proximity to one another. Agents can carry molecules and are rewarded when the molecule in their inventory is part of a reaction, either as a reactant or a product. In the three metabolic cycles variant, agents benefit from three different cycles, which continue as long as the minimum energy requirements are fulfilled. Agents must learn to facilitate the right reactions to generate enough energy to sustain the cycles. The environment also contains distractors, which are molecules that do not provoke reactions but provide a small constant reward to encourage agents to pursue less rewarding strategiesAgapiou et al. [2023].

- **Territory Inside Out:** Each agent is assigned a unique color and seeks to claim territory by painting walls in that color. Wet paint does not yield rewards. After 25 steps following the application of paint, if no further paint has been added, the paint dries and turns into a brighter shade of the agent's color. Once dry, the painted wall rewards the claiming player at a consistent rate. The more walls a player claims, the higher their expected rewards per timestep. In the Inside Out variant, agents are generated in a maze and must move inward toward the center of the map to claim territory. In this scenario, agents can zap each other, immobilizing the other agent for a set number of steps. An agent that is zapped twice is eliminatedAgapiou et al. [2023].

Each of the environments were designed with different properties. A summary of the properties of the 4 environments can be seen in Table A.1

Environment	**Properties**	**Total Properties**
Commons Harvest Partnership	Corresponding/Conflicting Interests, Dyadic (1:1) interactions cause immediate rewards, 1:many interactions cause immediate rewards, Social dilemma, Some physical state changes are irreversible, property/Ownership, resource sharing, teaching and sanctioning	8
Commons Harvest Closed	Conflicting/Corresponding Interests, Social dilemma, Some physical state changes are irreversible, property/Ownership, teaching and sanctioning	5
Chemistry Three Metabolic Cycles with Plentiful Distractors	Corresponding/Conflicting Interests, Near-perfect Information, many:many interactions cause immediate rewards, Equilibrium selection, Puzzle-like, Some physical state changes are irreversible, Stochastic rewards (variable magnitude or timing), Resource sharing, Task partitioning, Flexibility, Possible competitive exclusion (multiple niches)	11
Territory Inside Out	Conflicting/Corresponding Interests, Far-from-perfect information, many:many interactions cause immediate rewards, Social dilemma, Bargaining, Equilibrium selection, Some physical state changes are irreversible, Dynamic intra-episode learning, Delayed rewards (long after their proximal cause), Stochastic rewards (variable magnitude or timing), Property/ownership, Resource sharing, Teaching and sanctioning, Dynamic coalition formation	14

Table A.1: Overview of properties of meltingpot 2.0 environments Agapiou et al. [2023].

Appendix B

Hyperparameter choices and Computational resources

B.1 Blindsight task

For the blindisight task, we used a Nvidia RTX3070 gpu for training, with 8GB of RAM. The training time was maximum for MAPS (2nd order network and cascade model in both 1st and 2nd order network). For this setting, training over the 500 seeds took roughly 12 hours.

Hyperparameter	**Value**
Input size	100
Output size	100
Hidden size	60
lr first order	0.5
lr second order	0.1
Temperature	1.0
Step size	25
Gamma	0.98
Epochs number for training	200
Optimizer	*Adamax*
Cascade iterations	50

Table B.1: Hyperparameters used for the Blindsight Task.

B.2 Artificial Grammar Learning Task

For the AGL task, we used a Nvidia RTX3070 gpu for training, with 8GB of RAM. The training time was maximum for MAPS (2nd order network and cascade model in both 1st and 2nd order network). For this setting, training over the 500 seeds took roughly 12 hours.

Hyperparameter	**Value**
Input size	48
Output size	48
Hidden size	40
lr first order	0.4
lr second order	0.1
Temperature	1.0
Step size	1
Gamma	0.999
Epochs number for pre-training	60
Epochs number for training(high consciousness)	12
Epochs number for training(low consciousness)	3
Optimizer	$RangerVA$
Cascade iterations	50

Table B.2: Hyperparameters used for the Artificial Grammar Learning Task.

B.3 MinAtar

For the MinAtar environments, we used a GPU V100 for training. The training time was maximum for MAPS (2nd order network and cascade model in both 1st and 2nd order network). For this setting, training took roughly 6 days per million steps per seed, and double when training with our curriculum learning approach.

Hyperparameter	**Value**
Batch size	128
Replay buffer size	$100,000$
Target network update frequency	$1,000$
Training frequency	1
Number of frames	$500,000$
First N frames	$100,000$
Replay start size	$5,000$
End epsilon	0.1
Step size	0.0003
Step size (second order)	0.0002
Gradient momentum	0.95
Squared gradient momentum	0.95
Minimum squared gradient	0.01
Gamma	0.999
Step Size	1
Epsilon	1.0
Alpha	0.45
Cascade iterations	50
Optimizer	$Adam$
Max$_i nput_c hannels(Continuallearning)$	10
weight task loss (Continual learning)	0.3
weight weight regularization loss (Continual learning)	0.6
weight feature loss (Continual learning)	0.1

Table B.3: Hyperparameters used for the MinAtar experiments.

B.4 Meltingpot

For the meltingpot tasks, we used a Nvidia A100 gpu for training. The average training time was roughly 16 hours per seed(baseline, MAPS not implemented fully, only with simple 2nd order network with no cascade model due to limitations with computational resources). Every run required roughly 4-6 GB of RAM, mainly depending on the number of agents.

Hyperparameter	**Value**
Num agents (harvest closed)	6
Num agents (harvest partnership)	4
Num agents (chemistry)	8
Num agents (territory)	5
Hidden size	100
Actor lr	$7e-5$
Critic lr	100
Num env steps	$15e6$
Entropy coef	0.01
Clip param	0.2
Weight decay	$1e-5$
PPO epoch	15
Optimizer	$Adam$

Table B.4: Common hyperparameters used for the Meltingpot environments.

Appendix C

Additional Results

C.1 Blindsight and Artificial Grammar Learning Tasks

			Main Task			Wagering		
Blindsight	**2nd Net**	**Cascade**	**Accuracy**	**Z-score (Significant)**	**T-score (Significant)**	**Accuracy**	**Z-score**	**T-score (Significant)**
Setting-1 (Baseline)	No	No	0.95 ± 0.03			0.50 ± 0.05		
Setting-2	No	1st Net	0.97 ± 0.02	**8.50 (Yes)**	**12.40 (Yes)**	0.50 ± 0.05	0.45 (No)	0.00 (No)
Setting-3	Yes	No	0.96 ± 0.03	0.77 (No)	**5.27 (Yes)**	0.86 ± 0.03	128.1 (Yes)	**138.05 (Yes)**
Setting-4 (MAPS)	Yes	1st Net	0.97 ± 0.02	**9.01 (Yes)**	**12.40 (Yes)**	0.85 ± 0.04	121.2 (Yes)	**122.23 (Yes)**
Setting-5	Yes	2nd Net	0.96 ± 0.03	0.15 (No)	**5.27 (Yes)**	0.87 ± 0.04	126.7 (Yes)	**129.21 (Yes)**
Setting-6	Yes	Both	0.97 ± 0.02	**8.6 (Yes)**	**12.40 (Yes)**	0.86 ± 0.04	124.5 (Yes)	**125.72 (Yes)**
AGL-High Awareness	**2nd Net**	**Cascade**	**Accuracy**	**Z-score (Significant)**	**T-score (Significant)**	**Accuracy**	**Z-score**	**T-score (Significant)**
Setting-1 (Baseline)	No	No	0.63 ± 0.05			0.38 ± 0.07		
Setting-2	No	1st Net	0.64 ± 0.04	**6.38 (Yes)**	**3.49 (Yes)**	0.39 ± 0.09	1.10 (No)	1.96 (No)
Setting-3	Yes	No	0.64 ± 0.04	1.61 (No)	**3.49 (Yes)**	0.59 ± 0.06	45.9 (Yes)	**50.93 (Yes)**
Setting-4 (MAPS)	Yes	1st Net	0.66 ± 0.05	**8.20 (Yes)**	**9.49 (Yes)**	0.58 ± 0.06	43.3 (Yes)	**48.51 (Yes)**
Setting-5	Yes	2nd Net	0.63 ± 0.04	1.09 (No)	0.00 (No)	0.61 ± 0.06	48.7 (Yes)	**55.78 (Yes)**
Setting-6	Yes	Both	0.65 ± 0.04	**7.88 (Yes)**	**6.98 (Yes)**	0.58 ± 0.06	41.0 (Yes)	**48.51 (Yes)**
AGL-Low Awareness	**2nd Net**	**Cascade**	**Accuracy**	**Z-score (Significant)**	**T-score (Significant)**	**Accuracy**	**Z-score**	**T-score (Significant)**
Setting-1 (Baseline)	No	No	0.54 ± 0.08			0.14 ± 0.07		
Setting-2	No	1st Net	0.61 ± 0.07	**13.3 (Yes)**	**14.72 (Yes)**	0.17 ± 0.07	6.25 (Yes)	**6.78 (Yes)**
Setting-3	Yes	No	0.57 ± 0.07	**4.2 (Yes)**	**6.31 (Yes)**	0.83 ± 0.07	143.9 (Yes)	**155.86 (Yes)**
Setting-4 (MAPS)	Yes	1st Net	0.62 ± 0.07	**15.7 (Yes)**	**16.83 (Yes)**	0.82 ± 0.07	137.5 (Yes)	**153.60 (Yes)**
Setting-5	Yes	2nd Net	0.56 ± 0.07	**2.3 (Yes)**	**4.21 (Yes)**	0.87 ± 0.07	150.8 (Yes)	**164.89 (Yes)**
Setting-6	Yes	Both	0.62 ± 0.06	**15.0 (Yes)**	**17.89 (Yes)**	0.82 ± 0.07	141.1 (Yes)	**153.60 (Yes)**

Table C.1: Accuracy, Z-score, T-score, and Significant Results for Main Task and Wagering (Know Thyself environments). Z-score and T-score are statistical measures that quantify the distance between a data point and the mean of a dataset. We use 95 percent confidence to establish statistical significance. We use a total of N = 500 seeds for each setting. 2nd Net refers to the presence (or not) of a second-order network.

C.2 MinAtar

Seaquest	2nd-Net	Cascade	Training Rewards	Training Z-score (Significant)	Training T-score (Significant)	Validation Rewards	Validation Z-score	Validation T-score (Significant)
Setting-1 (Baseline)	No	No	1.48 ± 0.29			1.21 ± 0.16		
Setting-2	No	1st Net	0.90 ± 0.21	-2.32 (Yes)	-2.81 (Yes)	0.76 ± 0.19	-2.59 (Yes)	-3.14 (Yes)
Setting-3	Yes	No	1.04 ± 0.57	-0.97 (No)	-1.19 (No)	0.97 ± 0.61	-0.53 (No)	-0.66 (No)
Setting-4 (MAPS)	Yes	1st Net	3.71 ± 0.40	**6.46 (Yes)**	**7.82 (Yes)**	3.06 ± 0.34	**7.03 (Yes)**	**8.53 (Yes)**
Setting-5	Yes	2nd Net	4.10 ± 1.86	**1.97 (Yes)**	**2.41 (Yes)**	3.99 ± 1.84	**2.14 (Yes)**	**2.61 (Yes)**
Setting-6	Yes	Both	5.56 ± 1.85	**3.09 (Yes)**	**3.77 (Yes)**	6.15 ± 2.34	**2.98 (Yes)**	**3.65 (Yes)**
Setting-7 (ACB)	No	No	0.81 ± 0.03	-3.26 (Yes)	-3.98 (Yes)	0.63 ± 0.26	-2.65 (Yes)	-3.29 (Yes)
Asterix								
Setting-1 (Baseline)	No	No	3.49 ± 3.32			2.49 ± 1.94		
Setting-2	No	1st Net	1.38 ± 0.67	-0.88 (No)	-1.08 (No)	1.59 ± 0.90	-0.60 (No)	-0.73 (No)
Setting-3	Yes	No	6.27 ± 1.87	1.03 (No)	1.26 (No)	5.48 ± 1.30	1.81 (No)	**2.22 (Yes)**
Setting-4 (MAPS)	Yes	1st Net	4.95 ± 1.05	0.59 (No)	0.73 (No)	4.54 ± 1.01	1.32 (No)	1.62 (No)
Setting-5	Yes	2nd Net	3.40 ± 0.30	-0.04 (No)	-0.05 (No)	2.94 ± 0.25	0.32 (No)	0.40 (No)
Setting-6	Yes	Both	5.42 ± 0.69	0.81 (No)	0.99 (No)	5.77 ± 0.94	**2.15 (Yes)**	**2.64 (Yes)**
Setting-7 (ACB)	No	No	2.75 ± 0.09	-0.31 (No)	-0.39 (No)	2.13 ± 1.07	-0.23 (No)	-0.28 (No)
Breakout								
Setting-1 (Baseline)	No	No	5.96 ± 0.43			6.10 ± 0.21		
Setting-2	No	1st Net	7.08 ± 0.65	**2.05 (Yes)**	**2.49 (Yes)**	6.84 ± 0.41	**2.25 (Yes)**	**2.78 (Yes)**
Setting-3	Yes	No	6.12 ± 0.51	0.36 (No)	0.42 (No)	6.22 ± 0.69	0.24 (No)	0.29 (No)
Setting-4 (MAPS)	Yes	1st Net	8.54 ± 0.07	**8.36 (Yes)**	**10.26 (Yes)**	8.07 ± 0.72	**3.70 (Yes)**	**4.55 (Yes)**
Setting-5	Yes	2nd Net	6.23 ± 0.14	0.85 (No)	1.03 (No)	6.16 ± 0.05	0.39 (No)	0.48 (No)
Setting-6	Yes	Both	8.08 ± 0.24	**6.06 (Yes)**	**7.46 (Yes)**	7.91 ± 1.36	1.87 (No)	**2.28 (Yes)**
Setting-7 (ACB)	No	No	12.36 ± 1.13	**7.49 (Yes)**	**9.17 (Yes)**	11.67 ± 2.70	**2.90 (Yes)**	**3.56 (Yes)**
Space Invaders								
Setting-1 (Baseline)	No	No	23.62 ± 0.86			19.71 ± 1.84		
Setting-2	No	1st Net	31.52 ± 1.01	**8.46 (Yes)**	**10.31 (Yes)**	29.62 ± 2.28	**4.79 (Yes)**	**5.86 (Yes)**
Setting-3	Yes	No	21.14 ± 2.41	-1.37 (No)	-1.68 (No)	20.95 ± 2.60	0.55 (No)	0.67 (No)
Setting-4 (MAPS)	Yes	1st Net	26.84 ± 0.11	**5.28 (Yes)**	**6.43 (Yes)**	26.80 ± 1.59	**4.13 (Yes)**	**5.05 (Yes)**
Setting-5	Yes	2nd Net	28.11 ± 0.24	**7.14 (Yes)**	**8.71 (Yes)**	27.95 ± 1.10	**5.45 (Yes)**	**6.66 (Yes)**
Setting-6	Yes	Both	26.57 ± 0.72	**3.74 (Yes)**	**4.56 (Yes)**	22.97 ± 0.89	**2.26 (Yes)**	**2.76 (Yes)**
Setting-7 (ACB)	No	No	71.50 ± 17.38	**3.89 (Yes)**	**4.77 (Yes)**	59.26 ± 16.61	**3.34 (Yes)**	**4.10 (Yes)**
Freeway								
Setting-1 (Baseline)	No	No	30.96 ± 5.35			29.03 ± 7.12		
Setting-2	No	1st Net	32.77 ± 8.23	0.26 (No)	0.32 (No)	32.57 ± 8.82	0.44 (No)	0.54 (No)
Setting-3	Yes	No	24.72 ± 17.58	-0.48 (No)	-0.59 (No)	25.00 ± 17.68	-0.30 (No)	-0.37 (No)
Setting-4 (MAPS)	Yes	1st Net	35.53 ± 0.08	1.21 (No)	1.48 (No)	34.20 ± 2.83	0.95 (No)	1.17 (No)
Setting-5	Yes	2nd Net	39.41 ± 1.89	**2.11 (Yes)**	**2.58 (Yes)**	34.40 ± 2.97	0.98 (No)	1.21 (No)
Setting-6	Yes	Both	33.84 ± 2.03	0.71 (No)	0.87 (No)	30.63 ± 2.36	0.30 (No)	0.37 (No)
Setting-7 (ACB)	No	No	0.19 ± 0.04	-8.14 (Yes)	-9.96 (Yes)	0.17 ± 0.05	-5.73 (Yes)	-7.02 (Yes)

Table C.2: Training and validation rewards, Z-score, T-score, and significant results for SARL (MinAtar). Z-score and T-score are statistical measures that quantify the distance between a data point and the mean of a dataset. We use 95 percent confidence to establish statistical significance.

C.3 Meltingpot

			Training			Dist	Entropy	
Harvest Partnership	**2nd-Net**	**Cascade**	**Rewards**	**Z-score (Significant)**	**T-score (Significant)**	**Value**	**Z-score**	**T-score (Significant)**
Setting-1 (Baseline)	No	No	29.21 ± 0.72			1.82 ± 0.07		
Setting-2	No	1st Net	27.15 ± 0.04	-4.07 (Yes)	-4.95 (Yes)	1.71 ± 0.06	-1.73 (No)	**-2.07 (Yes)**
Setting-3	Yes	No	29.03 ± 0.29	-0.34 (No)	-0.40 (No)	1.88 ± 0.03	1.04 (No)	1.36 (No)
Setting-4 (MAPS)	Yes	1st Net	34.52 ± 0.98	**6.20 (Yes)**	**7.56 (Yes)**	1.64 ± 0.10	**-2.20 (Yes)**	**-2.55 (Yes)**
Setting-5	Yes	2nd Net	28.63 ± 0.36	-1.03 (No)	-1.25 (No)	1.88 ± 0.05	0.93 (No)	1.21 (No)
Setting-6	Yes	Both	33.01 ± 2.57	**2.01 (Yes)**	**2.47 (Yes)**	1.62 ± 0.11	**-2.28 (Yes)**	**-2.66 (Yes)**
Setting-7 (ACB)	No	No	24.58 ± 10.56	-0.75 (No)	-0.76 (No)	NA	NA	NA
Harvest Closed								
Setting-1 (Baseline)	No	No	19.52 ± 0.71			1.89 ± 0.01		
Setting-2	No	1st Net	26.05 ± 0.71	**9.18 (Yes)**	**11.26 (Yes)**	1.72 ± 0.01	**-20.93 (Yes)**	**-20.82 (Yes)**
Setting-3	Yes	No	19.15 ± 0.42	-0.64 (No)	-0.78 (No)	1.86 ± 0.01	**-4.51 (Yes)**	**-3.67 (Yes)**
Setting-4 (MAPS)	Yes	1st Net	25.21 ± 1.06	**6.31 (Yes)**	**7.72 (Yes)**	1.76 ± 0.12	-1.64 (No)	-1.87 (No)
Setting-5	Yes	2nd Net	18.81 ± 0.68	-1.02 (No)	-1.25 (No)	1.86 ± 0.05	-0.86 (No)	-1.02 (No)
Setting-6	Yes	Both	23.97 ± 1.30	**4.25 (Yes)**	**5.20 (Yes)**	1.77 ± 0.01	**-15.96 (Yes)**	**-14.70 (Yes)**
Setting-7 (ACB)	No	No	39.39 ± 10.69	**3.19 (Yes)**	**3.21 (Yes)**	NA	NA	NA
Chemistry								
Setting-1 (Baseline)	No	No	1.18 ± 0.09			1.95 ± 0.01		
Setting-2	No	1st Net	1.13 ± 0.01	-0.70 (No)	-0.96 (No)	1.81 ± 0.05	**-3.61 (Yes)**	**-4.76 (Yes)**
Setting-3	Yes	No	1.25 ± 0.01	1.07 (No)	1.34 (No)	1.93 ± 0.02	-1.15 (No)	-1.55 (No)
Setting-4 (MAPS)	Yes	1st Net	1.11 ± 0.05	-0.91 (No)	-1.18 (No)	1.83 ± 0.12	-1.33 (No)	-1.73 (No)
Setting-5	Yes	2nd Net	1.23 ± 0.06	0.72 (No)	0.80 (No)	1.90 ± 0.01	**-3.66 (Yes)**	**-6.12 (Yes)**
Setting-6	Yes	Both	1.15 ± 0.03	-0.37 (No)	-0.55 (No)	1.81 ± 0.19	-1.03 (No)	-1.27 (No)
Setting-7 (ACB)	No	No	1.41 ± 0.60	0.64 (No)	0.66 (No)	NA	NA	NA
Territory Inside Out								
Setting-1 (Baseline)	No	No	57.94 ± 6.84			1.98 ± 0.00		
Setting-2	No	1st Net	41.98 ± 7.43	-2.24 (Yes)	-2.74 (Yes)	2.01 ± 0.25	0.18 (No)	0.21 (No)
Setting-3	Yes	No	54.04 ± 0.34	-0.81 (No)	-0.99 (No)	1.97 ± 0.02	-0.66 (No)	-0.87 (No)
Setting-4 (MAPS)	Yes	1st Net	48.47 ± 1.45	-1.92 (No)	-2.35 (Yes)	1.58 ± 0.08	**-7.16 (Yes)**	**-8.66 (Yes)**
Setting-5	Yes	2nd Net	58.83 ± 6.15	0.14 (No)	0.17 (No)	1.92 ± 0.07	-1.29 (No)	-1.48 (No)
Setting-6	Yes	Both	46.05 ± 9.16	-1.47 (No)	-1.80 (No)	1.68 ± 0.12	**-3.46 (Yes)**	**-4.33 (Yes)**
Setting-7 (ACB)	No	No	101.99 ± 38.67	1.89 (No)	1.94 (No)	NA	NA	NA

Table C.3: Training rewards, dist entropy results, Z-score, T-score, and significant results for MARL (Melting Pot 2.0). Dist entropy is the action distribution entropy, which gets lower as the agents reduce their stochastic behavior and start to behave in a more stable manner (dist entropy should reduce as rewards increase). Z-score and T-score are statistical measures that quantify the distance between a data point and the mean of a dataset. We use 95 percent confidence to establish statistical significance.