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

**Anonymous authors**
Paper under double-blind review

**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, Curriculum learning

## Summary

Reinforcement Learning (RL) has made significant strides but struggles with social and continuous 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 continuous learning. We evaluate MAPS across four conditions: perceptual learning (Know Thyself), SARL (MinAtar), SARL with continuous 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 continuous 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 prevents 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.

## Contribution(s)

1. This paper proposes an architecture for improved learning using a confidence (2nd order) network, which is tested in 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 uncontrolled social environments (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 & Cleeremans (2010).

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

**Anonymous authors**
Paper under double-blind review

## Abstract

Reinforcement Learning (RL) has made significant strides but struggles with social and continuous 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 continuous learning. We evaluate MAPS across four conditions: perceptual learning (Know Thyself), SARL (MinAtar), SARL with continuous 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 continuous 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 prevents 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.

## 1 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 learning process makes RL more analogous to human cognition, enabling breakthroughs in game-playing AI Silver et al. (2016), robotics Zhang & 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 for rapid generalization across environments and tasks—capabilities RL still struggles to replicate Jain et al. (2020).

One critical cognitive shortcut that humans possess—but standard AI lacks—is self-awareness, or metacognitive ability—the capacity to monitor, evaluate, and adjust one's own cognitive processes in real-time. 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 reassess 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 in RL is through a 2nd-order network—a framework that pairs a primary task network (e.g., for image recognition or gameplay) with a secondary network dedicated to evaluating its performance. Serving as a reflective mechanism, the 2nd-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 & Clune (2024) or reduce redundant trials, accelerating convergence to optimal policies Anderson et al. (2006). These mechanisms enhance exploration-exploitation balance, accelerate skill acquisition, and improve adaptability in complex environments, making metacognition a key factor in developing more intelligent and efficient 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). 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 continuous learning across tasks and environments is often overlooked (Sidra 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). Unlike humans, who integrate knowledge adaptively, RL 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 full 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 continuous 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 (metacognitive) network. To assess social learning, we integrate a 2nd-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 continuous learning within a metacognitive architecture, we implement a second-order 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 continuous learning: pattern recognition (Know Thyself), SARL (MinAtar), SARL with Continuous learning (SARL+CL, MinAtar), and MARL (MeltingPot 2.0). To investigate social learning, we compare the benefits of a 2nd-order confidence network in perceptual vs. social (SARL and MARL) tasks, examining whether metacognition enhances decision-making and interaction dynamics. For continuous learning, we implement a 2nd-order 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.

## 2 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 + continuous 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 continuous learning approach for single agent reinforcement learning following a curriculum, and study whether MAPS attenuate catastrophic forgetting.

**Know-Thyself environments**

For pattern detection, we base our baseline implementation of a 2nd order network in the work of A. Pasquali & 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 & Cleeremans (2010). We employ a contrastive loss for the main task, which provides crucial information flow for wagering Chen et al. (2020). For wagering, we used a cross-entropy loss to handle class imbalance. Both the 1st and 2nd order networks implement a cascade model that facilitates a smooth graded build-up of activation McClelland et al. (1989). We empirically selected 50 cascade iterations for pattern detection, 50 for SARL, and no cascade model variant in MARL due to computational and training time constraints.

**Single and Multi agent reinforcement learning**

For SARL, we employ a DQN van Hasselt et al. (2015) framework. We use convolutional layers which allow for reduced computational complexity, an auto-encoder, and a replay buffer for the learning stability. We then compute the comparison matrix using the inputs and outputs of the value network's auto-encoder, and connect this to 2 wagering units. For the wagering objective, we compute rewards in batches of 128 using an exponential moving average (EMA) 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. For MARL, 0.25 was used. Both were found empirically.

For MARL, 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.

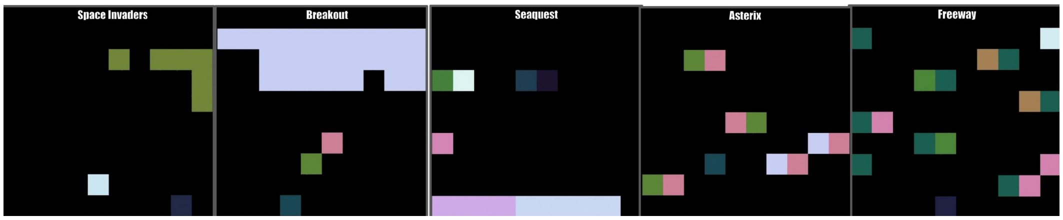


Figure 1: Visualization of trained agents of each of the MinAtar scenarios tested: Space Invaders(1st image to the left), Breakout(2nd), Seaquest(3rd), Asterix (4th), and Freeway(5th).

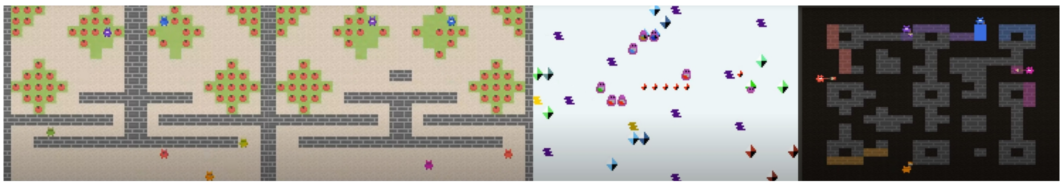

Figure 2: Visualization of trained agents of each of the Melting Pot scenarios tested: Commons Harvest Closed(1st image to the left), Commons Harvest Partnership(2nd), Chemistry Three Metabolic Cycles with Plentiful Distractors(3rd), and Territory Inside Out(4th).

**Continuous Learning**

We implement a continuous learning approach following a curriculum (curriculum learning) 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 & 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). Additionally, all loss components are normalized using the maximum individual loss observed throughout epochs to ensure comparability and facilitate summation. Our curriculum for training 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.

# 3 Experimental Set Up

We empirically select hyperparameters for each of our four major experiments (a complete list is provided in Appendix B). For three of the 4 major experiments (Know-Thyself environments, SARL, and SARL+CL), 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. The definition of each of these six settings is outlined below.

| Setting | Description |
|---|---|
| Setting 1 (Baseline) | No 2nd order network and no cascade model |
| Setting 2 | Cascade model, but no 2nd order network |
| Setting 3 | 2nd order network, but no cascade model |
| Setting 4 | 2nd order network, and a cascade model on the 1st order network only |
| Setting 5 | 2nd order network, and a cascade model on the 2nd order network only |
| Setting 6 (MAPS) | 2nd order network, and a cascade model on both networks |

Table 1: Description of the six settings used to analyze the components of MAPS.

Figure 1 provides a high-level depiction of the architecture used in both the SARL and SARL+CL experiments. It should be noted that for Know-Thyself environments, the equivalent of the Q-network would be a simple autoencoder, while for MARL we employ a GRU.

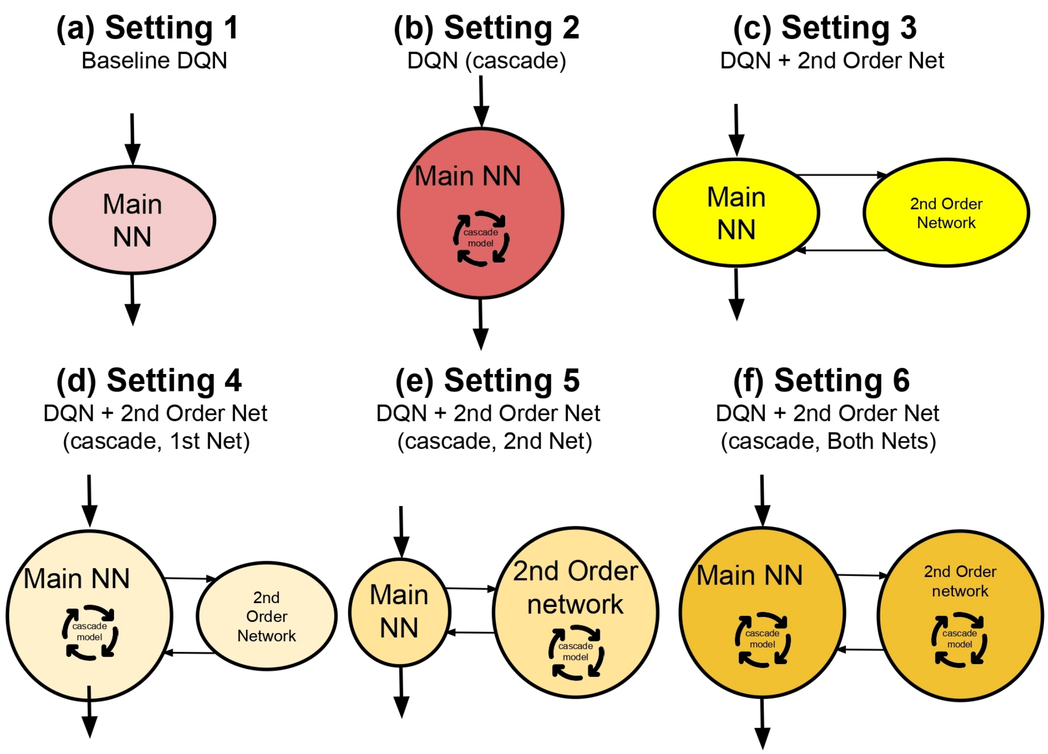


Figure 3: High level illustration of the six settings used to analyze the components of MAPS.

# 4 Results

### 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 & Cleeremans (2010), refer to Appendix A for additional information). To prevent overfitting, new patterns are generated for each epoch. Table 2 compares the proposed settings outlined in Table 1. 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 (2nd order network + cascade model in both networks). We compare our baseline (Setting-1), with MAPS (setting-6), obtaining a z-score of 8.6, meaning MAPS performance is superior and is statistically significant. However, we also see a similar overperformance in other settings (namely 2 and 4), 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 2nd-order network, and disable back-propagation 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 & 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 7.88 and 15.0 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 2nd order network and no cascade model. This supports the hypothesis that metacognition or a 2nd 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 using a 2nd 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 (141.1 for MAPS) compared to the high awareness case (41.0 for MAPS). This limitation appears to be mitigated by the improved information flow provided by the cascade model.

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

Table 2: Accuracy, Z-score, and Significant Results for Main Task and Wagering (Know Thyself environments). We use a total of 450 seeds for each setting.

## Single agent reinforcement learning (MinAtar environments)

In MinAtar, we test Space Invaders, Breakout, Seaquest, Asterix, and Freeway using the six defined settings to evaluate the effects of MAPS, as well as its main independent components (a 2nd order network and cascade model implementation). We train all settings for an equivalent of 500k steps across 3 seeds per configuration. Generally, we observe that MAPS outperforms our baseline in several cases, particularly in more complex environments. We note that using the cascade model with the 2nd order network specifically enables learning of more complex behaviors. This is evidenced by a final z-score at validation of 5.46 (MAPS) for Seaquest against the DQN baseline, and 2.89 for Space Invaders (refer to Table 3).

In Seaquest, we observe a particularly interesting behavior in the learning curves (refer to Figure 4) where DQN (baseline), DQN + cascade model, and DQN + 2nd order network all learn slowly. In contrast, when using a 2nd order network with a cascade model, effective learning occurs, which can be seen early in the training and validation curves. This suggests that a 2nd order network is indeed crucial in certain scenarios, where even though the cascade model enables the model to function, this would not work without the presence of a 2nd order network. This reinforces our belief that the cascade model, and the improved information flow it provides, is instrumental for metacognitive models in complex tasks.

Conversely, in Breakout, we observe similar learning patterns across most settings. We hypothesize this is due to the task's simplicity and lack of background obstacles or agents interacting with the main agent (except for a ball breaking walls). This reinforces our observation that MAPS can be especially useful for complex environments featuring interactions with obstacles or background populations (NPCs). Additionally, in some cases such as Space Invaders, we note that a baseline DQN + cascade model also performs well. This suggests us that the cascade model is a key element for learning complex behaviors, in some particular cases even without a 2nd order network, as also observed in perceptual tasks. However, it is likely insufficient for tasks that require a greater interaction with the environment, as previously shown with Seaquest.

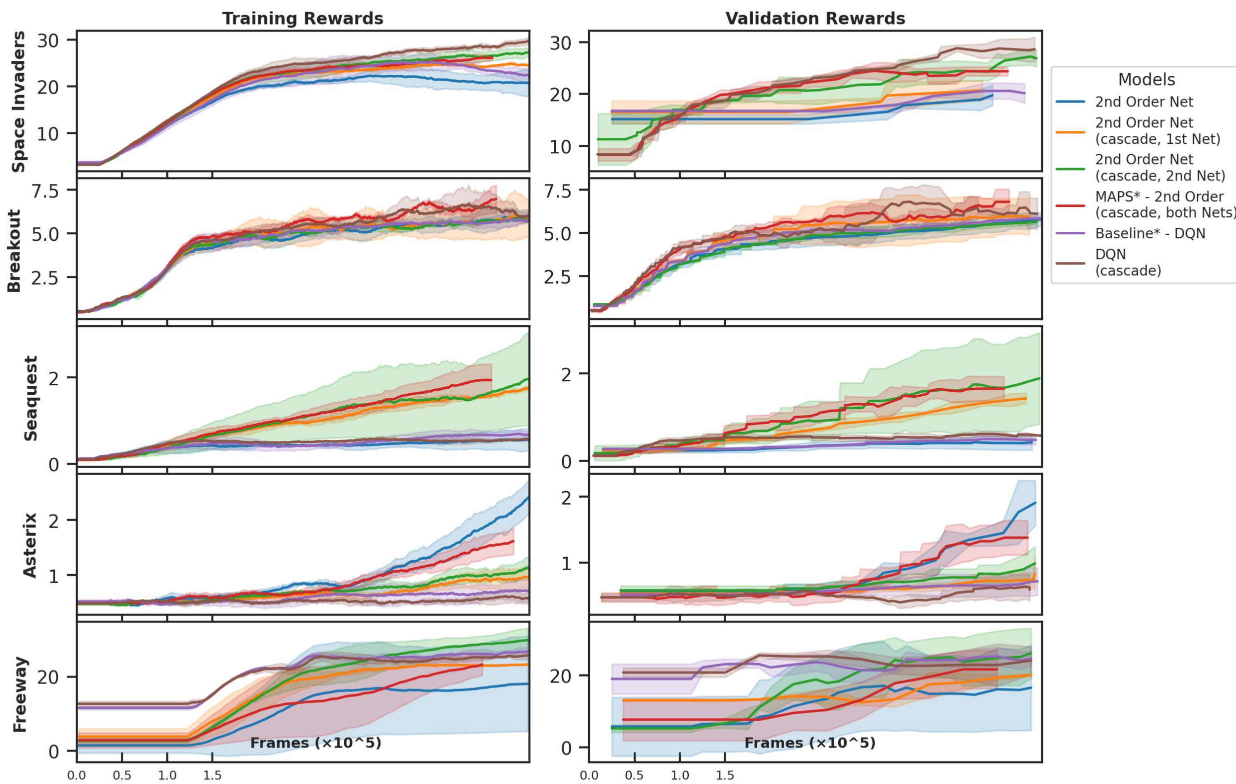


Figure 4: Training (left) and validation rewards (right) plots for SARL.

| | | | Training | | Validation | |
|---|---|---|---|---|---|---|
| **Space Invaders** | **2nd Net** | **Cascade** | **Rewards** | **Z-score (Significant)** | **Rewards** | **Z-score** |
| Setting-1 (Baseline) | No | No | $22.48 \pm 1.50$ | | $20.15 \pm 1.88$ | |
| Setting-2 | No | 1st Net | $29.72 \pm 0.85$ | **5.95(Yes)** | $28.62 \pm 2.36$ | **3.97(Yes)** |
| Setting-3 | Yes | No | $20.67 \pm 2.81$ | $-0.80$ (No) | $19.75 \pm 2.00$ | $-0.21$ (No) |
| Setting-4 | Yes | 1st Net | $24.57 \pm 0.16$ | **1.97 (Yes)** | $29.64 \pm 1.92$ | 0.26 () |
| Setting-5 | Yes | 2nd Net | $27.20 \pm 0.82$ | **3.91 (Yes)** | $26.89 \pm 1.59$ | **3.86 (Yes)** |
| Setting-6 (MAPS) | Yes | Both | $26.18 \pm 0.56$ | **3.27 (Yes)** | $24.38 \pm 0.87$ | **2.89 (Yes)** |
| **Breakout** | | | | | | |
| Setting-1 (Baseline) | No | No | $5.68 \pm 0.035$ | | $5.82 \pm 0.15$ | |
| Setting-2 | No | 1st Net | $6.08 \pm 0.34$ | 1.59 (No) | $6.1 \pm 0.89$ | 0.43 (No) |
| Setting-3 | Yes | No | $5.97 \pm 0.39$ | 1.00 (No) | $5.78 \pm 0.38$ | $-0.14$ (No) |
| Setting-4 | Yes | 1st Net | $5.81 \pm 1.00$ | 0.18 (No) | $5.96 \pm 1.06$ | 0.17 (No) |
| Setting-5 | Yes | 2nd Net | $5.75 \pm 0.12$ | 0.72 (No) | $5.63 \pm 0.12$ | $-1.47$ (No) |
| Setting-6 (MAPS) | Yes | Both | $6.98 \pm 0.80$ | **2.27 (Yes)** | $6.79 \pm 0.74$ | 1.80 (No) |
| **Seaquest** | | | | | | |
| Setting-1 (Baseline) | No | No | $0.68 \pm 0.10$ | | $0.48 \pm 0.10$ | |
| Setting-2 | No | 1st Net | $0.56 \pm 0.04$ | $-1.50$ (No) | $0.58 \pm 0.00$ | 1.29 (No) |
| Setting-3 | Yes | No | $0.55 \pm 0.26$ | $-0.66$ (No) | $0.42 \pm 0.18$ | $-0.36$ (No) |
| Setting-4 | Yes | 1st Net | $1.75 \pm 0.06$ | **12.34 (Yes)** | $1.43 \pm 0.12$ | **8.31 (Yes)** |
| Setting-5 | Yes | 2nd Net | $1.96 \pm 1.08$ | 1.67 (No) | $1.89 \pm 1.05$ | 1.89 (No) |
| Setting-6 (MAPS) | Yes | Both | $1.94 \pm 0.38$ | **4.56 (Yes)** | $1.65 \pm 0.28$ | **5.46 (Yes)** |
| **Asterix** | | | | | | |
| Setting-1 (Baseline) | No | No | $0.71 \pm 0.21$ | | $0.71 \pm 0.21$ | |
| Setting-2 | No | 1st Net | $0.58 \pm 0.11$ | $-0.79$ (No) | $0.59 \pm 0.16$ | $-0.69$ (No) |
| Setting-3 | Yes | No | $2.42 \pm 0.30$ | **6.64 (Yes)** | $1.91 \pm 0.34$ | **4.22 (Yes)** |
| Setting-4 | Yes | 1st Net | $0.96 \pm 0.20$ | 1.23 (No) | $0.83 \pm 0.24$ | 0.51 (No) |
| Setting-5 | Yes | 2nd Net | $1.14 \pm 0.19$ | **2.16 (Yes)** | $0.98 \pm 0.25$ | 1.16 (No) |
| Setting-6 (MAPS) | Yes | Both | $1.61 \pm 0.24$ | **4.09 (Yes)** | $1.38 \pm 0.27$ | **2.80 (Yes)** |
| **Freeway** | | | | | | |
| Setting-1 (Baseline) | No | No | $26.71 \pm 1.15$ | | $24.60 \pm 1.98$ | |
| Setting-2 | No | 1st Net | $25.70 \pm 1.15$ | $-0.87$ (No) | $24.03 \pm 3.85$ | $-0.18$ (No) |
| Setting-3 | Yes | No | $18.03 \pm 12.80$ | $-0.95$ (No) | $16.53 \pm 11.78$ | $-0.95$ (No) |
| Setting-4 | Yes | 1st Net | $23.23 \pm 0.18$ | $-4.23$ (Yes) | $20.0 \pm 0.29$ | $-3.24$ (Yes) |
| Setting-5 | Yes | 2nd Net | $29.78 \pm 3.26$ | 1.26 (No) | $26.10 \pm 6.93$ | 0.29 (No) |
| Setting-6 (MAPS) | Yes | Both | $23.27 \pm 2.84$ | $-1.59$ (No) | $21.60 \pm 5.27$ | $-0.75$ (No) |

Table 3: Training and validation rewards, Z-score, and Significant Results for SARL.

### Multi agent reinforcement learning (Melting Pot 2.0 environments)

In MARL settings, we conducted preliminary tests to evaluate the potential benefits of using a second-order network in both cooperative and competitive scenarios. We focused on two specific environments and benchmarked performance against the leading model presented by Agapiou et al. (2023). Agents were trained for 1.5M steps across three seeds. Our findings revealed that the second-order network achieved marginally superior performance compared to our GRU baseline in several environments, though it still underperformed relative to the top model (ACB) presented in Agapiou et al. (2023) (see Table 4). The chemistry game proved to be an exception, probably result of this environment being the only within the group of high coefficient of variation (CV). This may suggest that metacognition, or a second-order network approach, may be particularly valuable in environments characterized by high variability or stochastic behavior in MARL settings. Another intuition that points in this direction is the high complexity of the environment, being that: the simulation goes through 3 phases each representing a metabolic cycle, and there is presence of distractors, and, as we observed in MinAtar, a 2nd order network seems to be specially useful in scenarios where there is interaction with multiple background objects or obstacles (as seaquest). This in principle could be translated to settings such as chemistry, and thus making sense of our observation. However, these results may well be attributed to a completely normal variability due to it being just a marginal increase, and thus further experimentation and analysis is required in a more extensive study focusing on MARL.

Furthermore, we observed marked superiority of the second-order network model when compared to the simple GRU baseline in the "territory inside out" environment. Further evaluation of this environment yielded a positive z-score of 2.59 relative to our baseline across 10 seeds. Additionally, we noted that MAPS consistently produced positive outliers (see Figure 5). These results are preliminary mostly due to the high computational resources required to train agents using the Melting Pot 2.0 suite, and further testing with the cascade model is necessary to study the extent to which the architecture proposed by MAPS can bring to cooperative and competitive scenarios.

| Environment | GRU | GRU + 2nd Order | ACB |
|---|---|---|---|
| Harvest Closed | $18.9 \pm 1.4$ | $20.6 \pm 2.1$ | $32.8 \pm 10.6$ |
| Harvest Partnership | $28.1 \pm 1.9$ | $28.7 \pm 3.8$ | $31.9 \pm 11$ |
| Chemistry with Distractors | $1.2 \pm 0.03$ | $1.2 \pm 0.06$ | $1.1 \pm 0.8$ |
| Territory Inside Out | $63.5 \pm 8.7$ | $76.5 \pm 8.3$ | $80.3 \pm 48.0$ |

Table 4: Training rewards in 4 multi-agent settings: Commons Harvest Closed, Commons Harvest Partnership, Chemistry Three Metabolic Cycles with Plentiful Distractors, and Territory Inside Out.

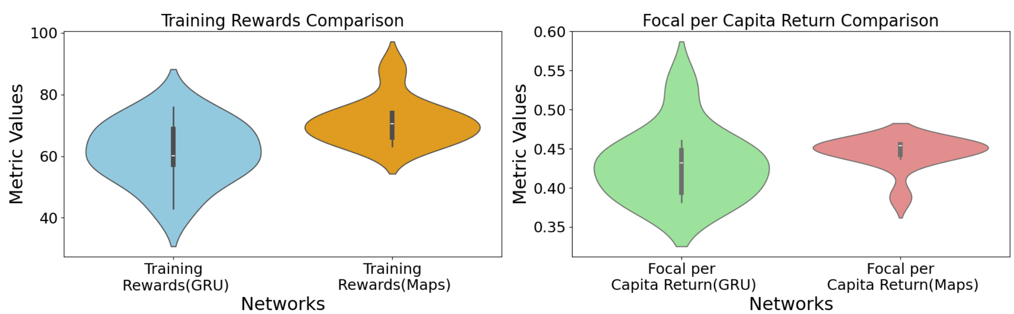


Figure 5: Territory Inside Out Results (10 seeds). Violin plot for avg. rewards (left); and Focal per Capita Return (right). Focal per capita return is a fairness measure (i.e. equal to 1.0 when all agents receive equal rewards), as defined by Agapiou et al. (2023).

**SARL + Continuous Learning (MinAtar environments)**

For continuous 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=0.5, weight regularization loss =0.3, feature loss=0.2). It's important to note that superior retention does not necessarily translate to effective training on new tasks. The results from our exploration of weights for the 3 losses can be seen in Figure 7.

We then conducted two main experiments, where 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 the right side of Figure 6, utilized the optimal retention parameters identified through exploration. This was tested with two base settings: DQN and DQN + 2nd order network. For Space Invaders, when evaluated after training through various environments, we observed reduced forgetting following the acquisition of new knowledge from one following task. However, in all cases, performance approached that of a random policy after training on two additional environments or more.

Subsequently, we empirically tested different loss combinations, including one with a higher proportion of weight regularization loss (weights: task loss = 0.3, weight regularization = 0.6, and feature loss = 0.1). In this case, this combination was found empirically after testing for several seeds with a higher proportion of the weight regularization loss. 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 2nd order network and cascade model in the 2nd 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 2nd 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 efectively retain knowledge, as the notion is that the 2nd order network could, at some point, gain independence of the main task to provide valuable confidence information regardless of the task.

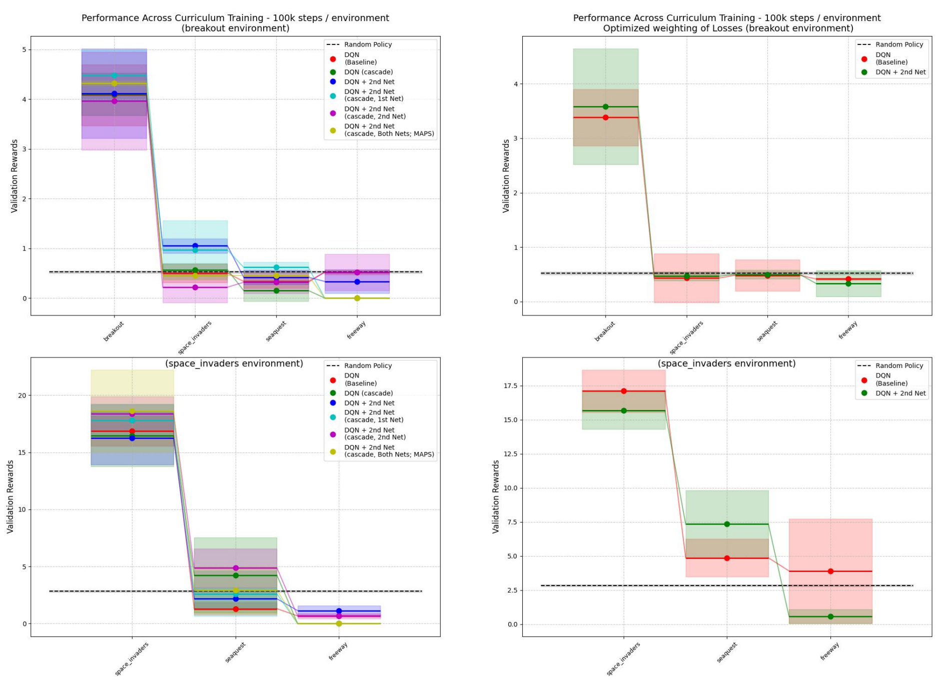


Figure 6: Continuous learning results. Left panels show validation rewards for each environment after sequential training using our continuous learning approach. The top graph displays evaluation of the Breakout environment after each scenario, while the bottom graph shows the same evaluation for Space Invaders. Right panels present preliminary results (baseline and 2nd Order Network only) using the optimal parameters identified for retention.

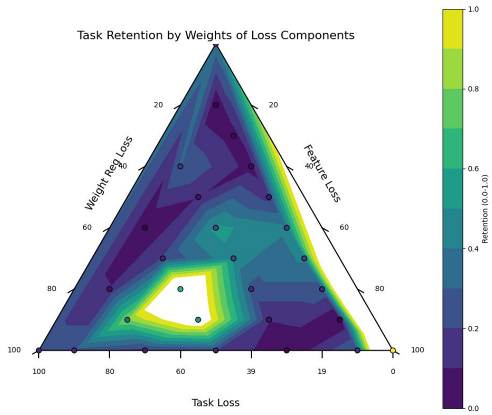


Figure 7: Ternary plot representing an extensive search of combinations of the three losses used for our continuous learning approach. Retention represents the fraction of original validation rewards effectively preserved after evaluation post-training of a new environment. For practicality, Breakout was used as baseline followed by training in Space Invaders (50,000 steps per environment).

# 5 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 2nd-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 Uncontrolled Social Environment Learning in MAPS**

In Seaquest, while DQN and DQN + cascade model struggle, models combining a 2nd-order network and a cascade model show early and effective learning, highlighting the necessity of both components in complex tasks. In Breakout, most settings perform similarly, likely due to the task's simplicity, suggesting that MAPS is least beneficial in environments with few obstacles. In Space Invaders, the DQN + cascade model alone performs well, reinforcing the cascade model's role in complex learning, as observed in perceptual tasks. However, in Seaquest, neither baseline nor partial MAPS implementations succeed—only DQN + 2nd-order network + cascade model learns effectively, confirming the necessity of both mechanisms. In Asterix, the 2nd-order network boosts early learning, though the difference diminishes over time, aligning with findings from the AGL task, where 2nd-order networks improve early-stage learning speed.

The Key takeaway for MAPS in an uncontrolled social environment is that it outperforms the DQN baseline in complex tasks, with the combination of a 2nd-order network and a cascade model proving essential for learning more sophisticated behaviors.

**MARL: Evaluating Controlled Social Environment Learning in MAPS**

MAPS was tested against a GRU-only baseline in MARL settings over 1.5M steps across three seeds. While MAPS performed slightly better than GRU, it fell short of the top ACB model (Agapiou et al., 2023). However, in the chemistry game, MAPS showed promise, suggesting that 2nd-order networks are particularly useful in high-variability, high-stochasticity environments.

In Territory Inside Out, MAPS achieved a positive z-score of 2.59 over 10 seeds, showing potential for adaptive decision-making. Additionally, MAPS tended to produce positive outliers, suggesting capacity for dynamic learning (see Appendix D.4). However, these results remain preliminary, requiring further evaluation across all six experimental settings.

We learn from this that While MAPS shows promise in high-variability environments, further testing is needed to determine its full impact on multi-agent reinforcement learning.

**SARL+CL: Evaluating Continuous 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 = 0.5, weight regularization = 0.3, 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.

Testing these parameters on DQN and DQN + 2nd-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. Adjusting weight regularization loss to 0.6 improved retention in Breakout and Space Invaders, but learning still degraded with additional environments.

In summary, DQN alone struggles with retention, often performing at or below random policy levels. In contrast, 2nd-order networks, especially with a cascade model, significantly improve continuous learning by preserving prior knowledge.

# 6 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 uncontrolled social environments (SARL), the combination of a 2nd-order network and a cascade model is essential for mastering sophisticated behaviors, particularly in tasks with dynamic obstacles or interactions. In continuous learning (SARL + CL), 2nd-order networks with a cascade model significantly improve knowledge retention, preventing catastrophic forgetting better than DQN alone. In controlled social environments (MARL), MAPS shows promise in high-variability tasks, though further testing is required to fully assess its impact on multi-agent reinforcement learning. 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.

# References


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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

Kamal K Ndousse, Douglas Eck, Sergey Levine, and Natasha Jaques. Emergent social learning via multi-agent reinforcement learning. In *International conference on machine learning*, pp. 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.

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.

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.

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.

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.

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.

# A Appendix / supplemental material

## Appendix A - Additional Environment details

### Appendix 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 & 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.

### Appendix 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) A. Pasquali & Cleeremans (2010).

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.

### Appendix 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 & 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 & 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 & 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 & 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 & 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 & Tian (2019).

### Appendix 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 8). We chose the three environments with the lowest variability and the environment with the highest positive variability.

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

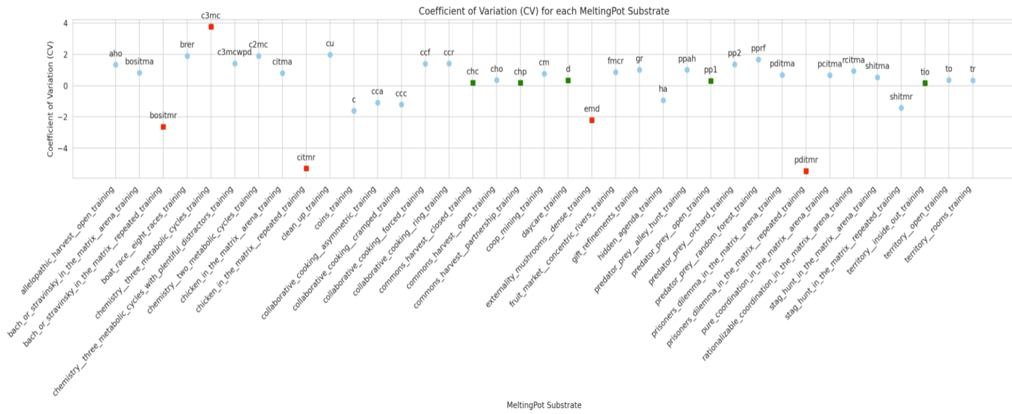


Figure 8: Variability among Melting Pot environments according to the experimentation in Agapiou et al. (2023).

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).

# Appendix B - Hyperparameter choices and Computational resources

## Appendix 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 450 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 5: Hyperparameters used for the Blindsight Task.

## Appendix 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 450 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 6: Hyperparameters used for the Artificial Grammar Learning Task.

### Appendix 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$_{i}nput_{c}hannels(Continuouslearning)$ | 10 |
| weight task loss (Continuous learning) | 0.3 |
| weight weight regularization loss (Continuous learning) | 0.6 |
| weight feature loss (Continuous learning) | 0.1 |

Table 7: Hyperparameters used for the MinAtar experiments.

### Appendix 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 8: Common hyperparameters used for the Meltingpot environments.

# Appendix C - Architectures

### Appendix C.1 - Blindsight task and Artificial Grammar Learning Task

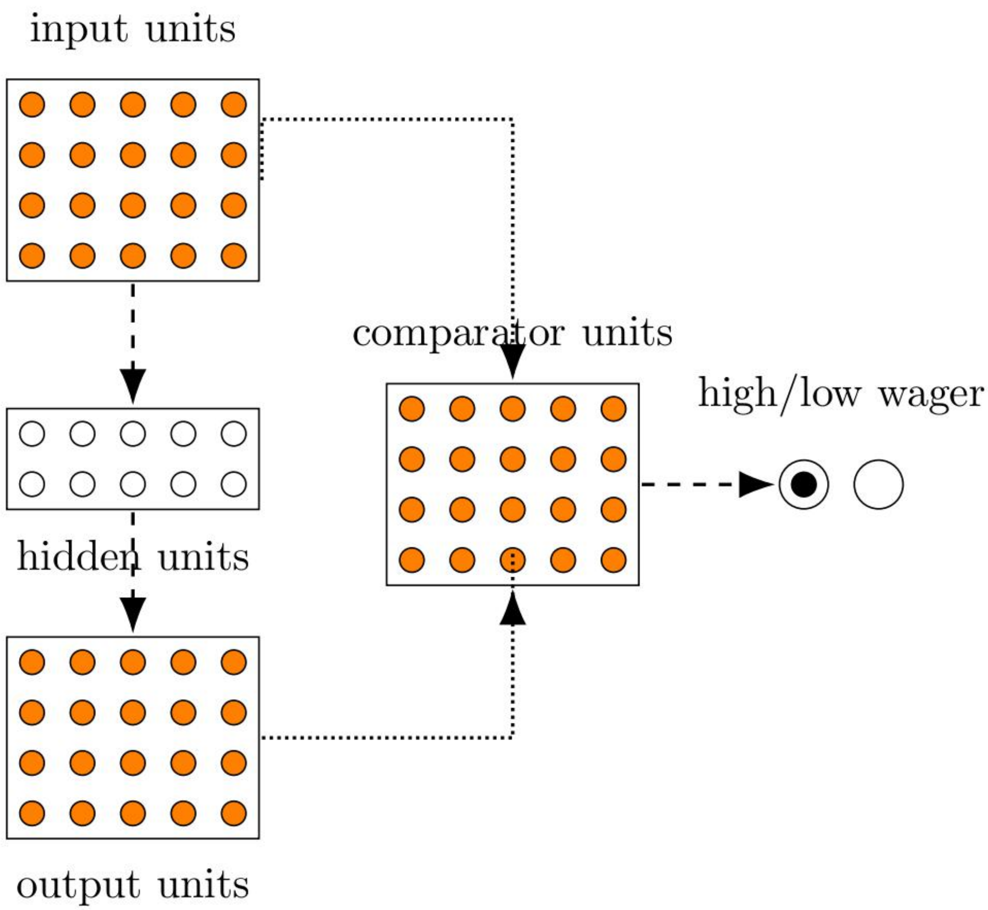


Figure 9: Illustration of the architecture used for both the Blindsight and Artificial Grammar Learning tasks.

575 **Appendix C.2 - Meltingpot**

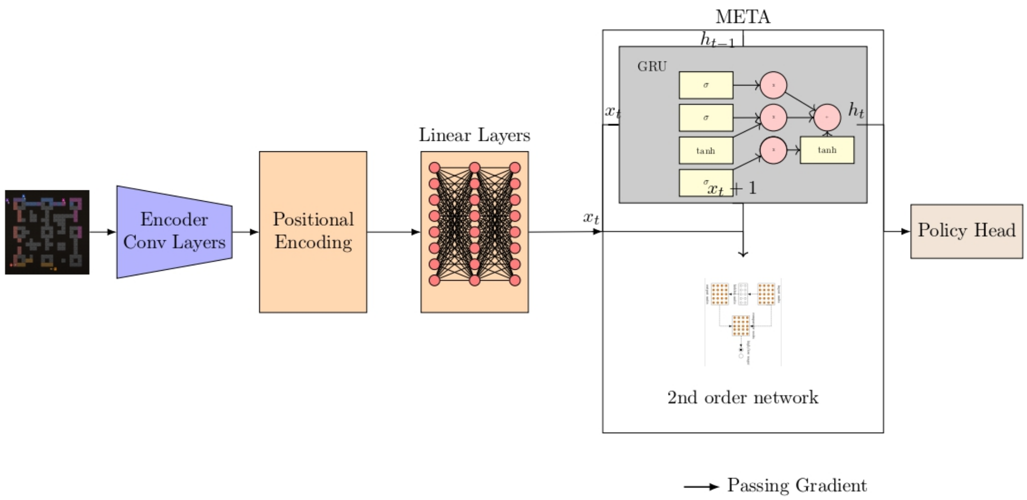


Figure 10: Illustration of the architecture used for all the Meltingpot environments

# Appendix D - Additional results

## Appendix D.1 - Meltingpot

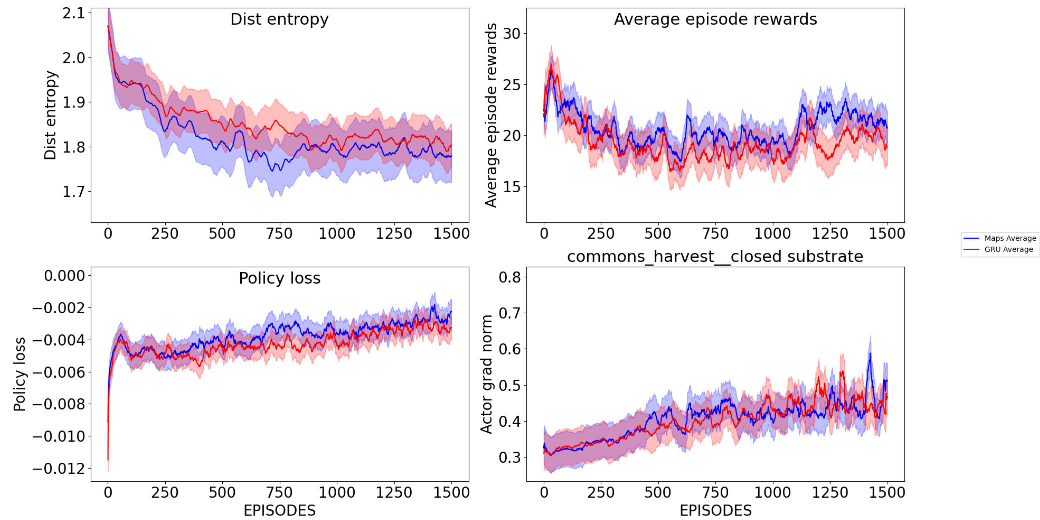


Figure 11: Results per episode over 1.5 million steps for commons harvest closed environment. To the top left the evaluation parameter is dist entropy, which represents the action distribution entropy, where a lower value points to a lower overall stochastic behaviour of the agents. The top right represents the average reward of all agents, where a higher value is desired. Bottom left is the policy loss, and bottom right is the actor gradient norm.

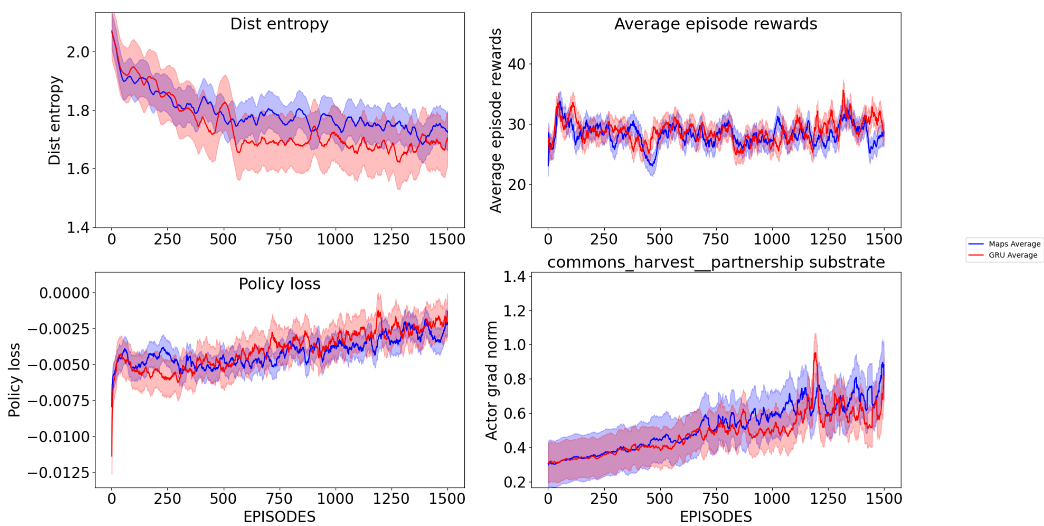


Figure 12: Results per episode over 1.5 million steps for commons harvest partnership environment. To the top left the evaluation parameter is dist entropy, which represents the action distribution entropy, where a lower value points to a lower overall stochastic behaviour of the agents. The top right represents the average reward of all agents, where a higher value is desired. Bottom left is the policy loss, and bottom right is the actor gradient norm.

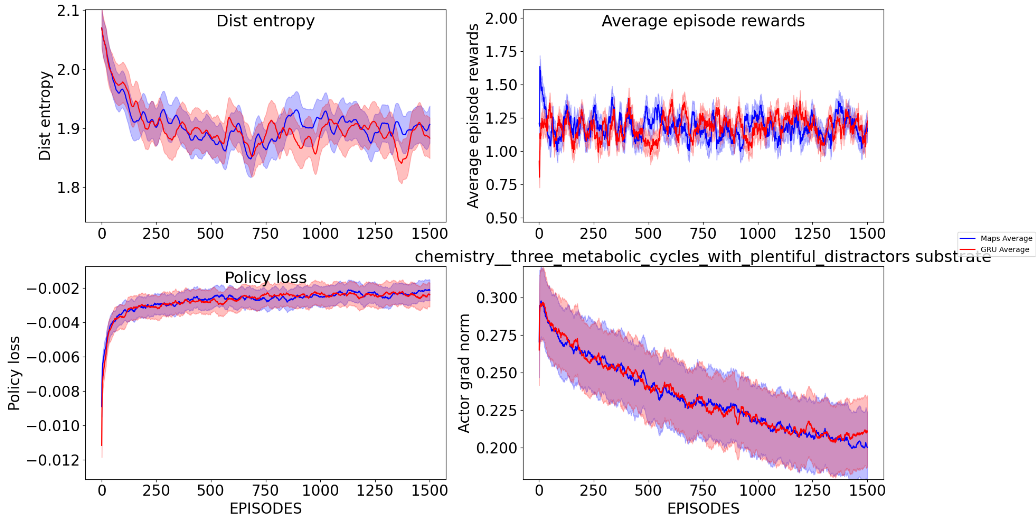


Figure 13: Results per episode over 1.5 million steps for chemistry environment. To the top left the evaluation parameter is dist entropy, which represents the action distribution entropy, where a lower value points to a lower overall stochastic behaviour of the agents. The top right represents the average reward of all agents, where a higher value is desired. Bottom left is the policy loss, and bottom right is the actor gradient norm.

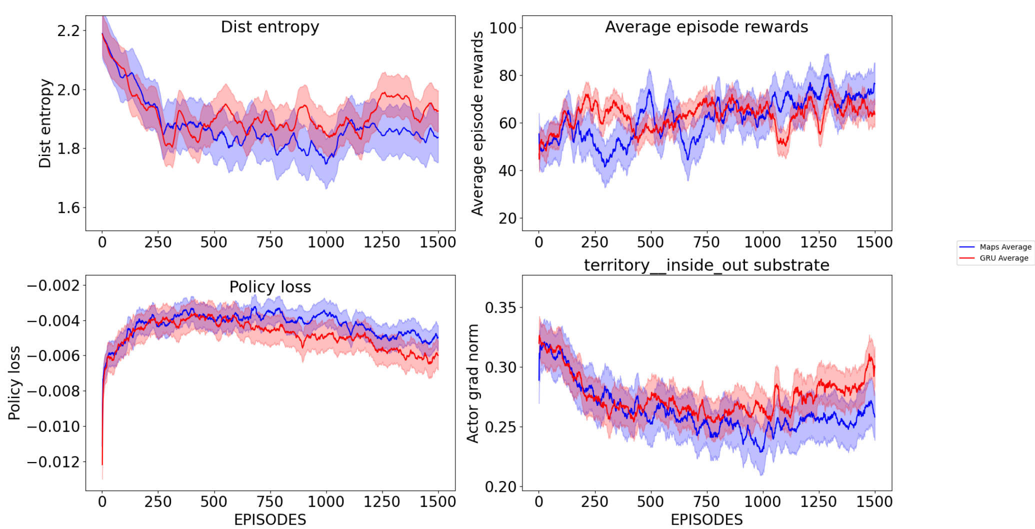


Figure 14: Results per episode over 1.5 million steps for territory inside out environment. To the top left the evaluation parameter is dist entropy, which represents the action distribution entropy, where a lower value points to a lower overall stochastic behaviour of the agents. The top right represents the average reward of all agents, where a higher value is desired. Bottom left is the policy loss, and bottom right is the actor gradient norm.