Multi-Agent Reinforcement Learning with Selective State-Space Models

Extended Abstract

Jemma Daniel
InstaDeep
London, United Kingdom
j.daniel@instadeep.com

Ruan John de Kock
InstaDeep
Cape Town, South Africa
r.dekock@instadeep.com

Louay Ben Nessir
InstaDeep
Tunis, Tunisia
l.nessir@instadeep.com

Sasha Abramowitz
InstaDeep
Cape Town, South Africa
s.abramowitz@instadeep.com

Omayma Mahjoub
InstaDeep
Tunis, Tunisia
o.mahjoub@instadeep.com

Wiem Khlifi
InstaDeep
Tunis, Tunisia
w.khlifi@instadeep.com

Juan Claude Formanek
InstaDeep
Cape Town, South Africa
c.formanek@instadeep.com

Arnu Pretorius
InstaDeep
Kigali, Rwanda
a.pretorius@instadeep.com

ABSTRACT

Transformer-based architectures have achieved strong performance in multi-agent reinforcement learning (MARL). A notable example is the Multi-Agent Transformer (MAT), which is able to achieve state-of-the-art performance in many cooperative tasks. However, MAT's use of attention with quadratic complexity limits scalability to large agent populations. In contrast, State-Space Models (SSMs), such as Mamba, offer improved efficiency, but their potential in MARL remains unexplored. We introduce Multi-Agent Mamba (MAM), which replaces attention in MAT with causal, bi-directional, and cross-attentional Mamba blocks. Experiments show that MAM matches MAT's performance while improving computational efficiency, suggesting SSMs can replace attention-based architectures in MARL for better scalability.[1]

CCS CONCEPTS

• **Computing methodologies** → **Multi-agent planning**; **Planning under uncertainty**.

KEYWORDS

Multi-Agent Reinforcement Learning; Multi-Agent Systems; Transformer; Attention; Selective State-Space Model; Mamba

ACM Reference Format:
Jemma Daniel, Ruan John de Kock, Louay Ben Nessir, Sasha Abramowitz, Omayma Mahjoub, Wiem Khlifi, Juan Claude Formanek, and Arnu Pretorius. 2025. Multi-Agent Reinforcement Learning with Selective State-Space Models: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

[1]All experimental data and code is available at: https://sites.google.com/view/multi-agent-mamba.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) faces scalability challenges, particularly as the number of agents increases [1]. The Multi-Agent Transformer (MAT) [18] achieves state-of-the-art performance but suffers from quadratic scaling in sequence length [17], creating a computational bottleneck.

State-Space Models (SSMs) [5, 8–10, 10, 16] offer a promising alternative, scaling linearly in sequence length. Of particular interest is Mamba [7], a selective SSM with fast inference and linear complexity that matches Transformer performance in natural language tasks.

This paper explores replacing attention in MAT with Mamba blocks, including vanilla, bi-directional and a novel cross-attention variant designed to replace MAT's cross-attention. We introduce the Multi-Agent Mamba (MAM) and evaluate its performance against MAT on numerous well-known MARL benchmark environments.

2 BACKGROUND

Cooperative Multi-Agent Reinforcement Learning. MARL can be formalised as a Markov game $\langle \mathcal{N}, \boldsymbol{O}, \boldsymbol{\mathcal{A}}, R, P, \gamma \rangle$ [12], where $\mathcal{N}$ is the set of agents, $\boldsymbol{O}$ and $\boldsymbol{\mathcal{A}}$ are the joint observation and action spaces, respectively, R is the joint reward function, P is the transition probability function, and γ is the discount factor. Agents select actions using a joint policy $\boldsymbol{\pi}$ to maximise cumulative rewards.

The Multi-Agent Advantage Decomposition Theorem. HAPPO [11] decomposes the joint advantage into individual advantages $A_\pi^{i_m}$, enabling sequential policy updates:

$$A_\pi^{i_{1:n}} = \sum_{m=1}^{n} A_\pi^{i_m}\left(\mathbf{o}, \mathbf{a}^{i_{1:m-1}}, a^{m}\right). \tag{1}$$

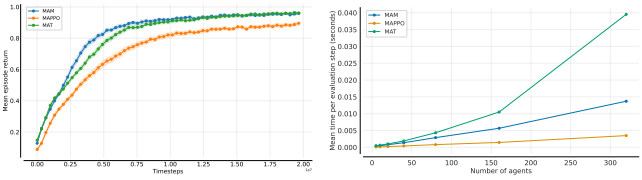
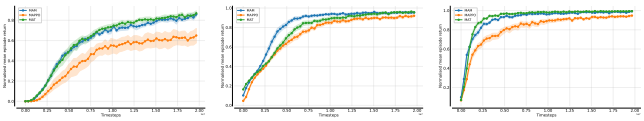


Figure 1: *Left:* Task- and environment-aggregated mean episode returns with 95% confidence intervals. *Right:* Mean evaluation step time in SMAX tasks with increasing agent counts.



(a) RWARE **(b) SMAX** **(c) LBF**

Figure 2: Aggregated episode returns per environment with 95% confidence intervals.

This so-called multi-agent advantage decomposition theorem, above in Equation 1, simplifies optimisation and provides monotonic policy improvement guarantees [18].

Multi-Agent Transformer (MAT). MAT [18] applies self- and cross-attention over agent observations and actions. Given key, query and value matrices $(\mathbf{K}, \mathbf{Q}, \mathbf{V})$, attention is

$$\text{attention}(\mathbf{K}, \mathbf{Q}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{d_k}\right)\mathbf{V}. \tag{2}$$

While achieving unmatched performance, MAT scales quadratically with agents, which swiftly becomes infeasible in scenarios with large agent populations.

State-Space Models (SSMs) and Mamba. SSMs model a continuous and time-invariant system with an input signal $x(t) \in \mathbb{R}^D$ via a latent state $h(t) \in \mathbb{R}^N$ evolving as

$$h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t), \quad y(t) = \mathbf{C}h(t), \tag{3}$$

for output $y(t) \in \mathbb{R}^D$. Equation 3 can also be discretised. These parameters can all be made learnable, but model dynamics are constant. Mamba [7] introduces input-dependent parameters and a hardware-aware parallel recurrent mode, achieving linear scaling with strong performance. This makes Mamba a promising candidate for replacing Transformers in MARL architectures.

3 MULTI-AGENT MAMBA (MAM)

3.1 Encoder

MAT's encoder relies on unmasked self-attention, while vanilla Mamba is causal by construction. To preserve bidirectional information flow, we adopt a bi-directional Mamba block outlined by Schiff et al. [15]. This involves applying Mamba twice, once forward and once reversed, and then merging the outputso maintain parameter efficiency, we share projection weights.

3.2 Decoder

Mamba naturally supports causal processing on a single sequence, making it a straightforward replacement for MAT's self-attention. However, adapting Mamba for cross-attention requires modification. We introduce *CrossMamba*, which extends Mamba to process two input sequences. Inspired by Mamba's attentional form in [2], we reformulate Mamba's state-space representation to incorporate cross-sequence dependencies. In our MARL setup, observations form the target sequence, while actions serve as the source, enabling effective cross-agent information flow during action selection.

4 EXPERIMENTS

4.1 Experimental Setup

We build on the JAX-based MARL library Mava [4]. We evaluate MAM on JAX-based versions of Robotic Warehouse (RWARE) [13], Level-Based Foraging (LBF) [3] and the StarCraft Multi-Agent Challenge (SMAX) [14]. Each environment includes tasks of varying difficulty and agent counts, totalling three RWARE tasks, seven LBF tasks and eleven SMAX tasks.

We train each algorithm for 20M timesteps on ten random seeds, evaluating performance 61 times throughout training. At each evaluation, we compute episode returns across 32 rollouts and aggregate results using MARL-eval with min-max normalisation [6].

4.2 Results

Performance Results. The left-hand plot in Figure 1 compares MAM, MAT, and MAPPO, aggregated over all tasks and environments. MAM achieves performance on par with MAT, the current state-of-the-art, while learning faster.

To ensure fairness, we normalise and aggregate results per environment using MARL-eval [6]. Figure 2 shows that MAM's final performance is comparable to MAT's in each environment, with superior sample efficiency in SMAX. SMAX tasks feature more agents (9.7 on average vs. 3 in LBF and 3.3 in RWARE), suggesting that MAM possesses superior scaling abilities with increasing agent counts.

Inference Speed Results. MAM operates solely in recurrent mode, leading to faster inference. The right-hand plot in Figure 1 demonstrates that MAT's evaluation time scales quadratically with agent count, while MAM and MAPPO scale linearly. This efficiency could make MAM preferable for real-world many-agent systems.

5 CONCLUSION

We introduced Multi-Agent Mamba (MAM), a sequence-based MARL architecture that replaces attention with causal, bi-directional, and cross-attentional Mamba blocks. Experiments show that MAM matches MAT's performance with improved computational efficiency, while potentially offering better sample efficiency in many-agent settings.

Despite these strengths, current MARL benchmarks limit evaluation in large-agent scenarios, and our JAX-based implementation of MAM cannot fully leverage the CUDA optimisations of PyTorch-based Mamba. Future work includes developing many-agent environments for testing and implementing specific optimisations to make MAM more faithful to the original Mamba implementation.

REFERENCES

[1] Stefano V. Albrecht, Filippos Christianos, and Lukas Schäfer. 2024. *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches.* MIT Press. https://www.marl-book.com

[2] Ameen Ali, Itamar Zimerman, and Lior Wolf. 2024. The Hidden Attention of Mamba Models. arXiv:2403.01590 [cs.LG] https://arxiv.org/abs/2403.01590

[3] Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. 2021. Shared Experience Actor-Critic for Multi-Agent Reinforcement Learning. arXiv:2006.07169 [cs.MA] https://arxiv.org/abs/2006.07169

[4] Ruan de Kock, Omayma Mahjoub, Sasha Abramowitz, Wiem Khlifi, Callum Rhys Tilbury, Claude Formanek, Andries P. Smit, and Arnu Pretorius. 2023. Mava: a research library for distributed multi-agent reinforcement learning in JAX. https://arxiv.org/pdf/2107.01460.pdf

[5] Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. 2023. Hungry Hungry Hippos: Towards Language Modeling with State Space Models. arXiv:2212.14052 [cs.LG] https://arxiv.org/abs/2212.14052

[6] Rihab Gorsane, Omayma Mahjoub, Ruan de Kock, Roland Dubb, Siddarth Singh, and Arnu Pretorius. 2022. Towards a Standardised Performance Evaluation Protocol for Cooperative MARL. arXiv:2209.10485 [cs.LG] https://arxiv.org/abs/2209.10485

[7] Albert Gu and Tri Dao. 2024. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv:2312.00752 [cs.LG] https://arxiv.org/abs/2312.00752

[8] Albert Gu, Karan Goel, and Christopher Ré. 2022. Efficiently Modeling Long Sequences with Structured State Spaces. arXiv:2111.00396 [cs.LG] https://arxiv.org/abs/2111.00396

[9] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. 2021. Combining Recurrent, Convolutional, and Continuous-time Models with Linear State-Space Layers. arXiv:2110.13985 [cs.LG] https://arxiv.org/abs/2110.13985

[10] Ankit Gupta, Albert Gu, and Jonathan Berant. 2022. Diagonal State Spaces are as Effective as Structured State Spaces. arXiv:2203.14343 [cs.LG] https://arxiv.org/abs/2203.14343

[11] Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. 2022. Trust Region Policy Optimisation in Multi-Agent Reinforcement Learning. arXiv:2109.11251 [cs.AI] https://arxiv.org/abs/2109.11251

[12] Michael L. Littman. 1994. Markov Games as a Framework for Multi-Agent Reinforcement Learning. In *Machine Learning Proceedings 1994*, William W. Cohen and Haym Hirsh (Eds.). Morgan Kaufmann, San Francisco (CA), 157–163. https://doi.org/10.1016/B978-1-55860-335-6.50027-1

[13] Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. 2021. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks. arXiv:2006.07869 [cs.LG] https://arxiv.org/abs/2006.07869

[14] Alexander Rutherford, Benjamin Ellis, Matteo Gallici, Jonathan Cook, Andrei Lupu, Gardar Ingvarsson, Timon Willi, Akbir Khan, Christian Schroeder de Witt, Alexandra Souly, Saptarashmi Bandyopadhyay, Mikayel Samvelyan, Minqi Jiang, Robert Tjarko Lange, Shimon Whiteson, Bruno Lacerda, Nick Hawes, Tim Rocktaschel, Chris Lu, and Jakob Nicolaus Foerster. 2023. JaxMARL: Multi-Agent RL Environments in JAX.

[15] Yair Schiff, Chia-Hsiang Kao, Aaron Gokaslan, Tri Dao, Albert Gu, and Volodymyr Kuleshov. 2024. Caduceus: Bi-Directional Equivariant Long-Range DNA Sequence Modeling. arXiv:2403.03234 [q-bio.GN] https://arxiv.org/abs/2403.03234

[16] Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. 2023. Simplified State Space Layers for Sequence Modeling. arXiv:2208.04933 [cs.LG] https://arxiv.org/abs/2208.04933

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. arXiv:1706.03762 [cs.CL] https://arxiv.org/abs/1706.03762

[18] Muning Wen, Jakub Grudzien Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. 2022. Multi-Agent Reinforcement Learning is a Sequence Modeling Problem. arXiv:2205.14953 [cs.MA] https://arxiv.org/abs/2205.14953