Memory based Generalization for Cognitive Robots

Shweta Singh (shweta.singh@research.iiit.ac.in)
IIIT, Hyderabad, India
Ritesh Shrivastav (shrivastavrc22.mfg@coeptech.ac.in)
COEP Technological University, India
Siddhesh Dorkulkar (dorkulkarsm22.mfg@coeptech.ac.in)
COEP Technological University, India
Vedant Ghatnekar (1032190997@mitwpu.edu.in)
MIT WPU, India

Abstract:

Reinforcement learning (RL) holds promise for training agents in complex environments, but generalization remains a key challenge. This study focuses on addressing generalization in maze navigation using Proximal Policy Optimization (PPO) with transformer-based models. We develop a custom maze environment in Unity 3D and train agents using PPO integrated with Transformer XL and Gated Transformer XL architectures. Our experiments assess the agents' ability to generalize policies to unseen maze configurations, demonstrating significant improvements in generalization performance. This research contributes to advancing RL for navigation tasks.

Keywords: Deep Reinforcement Learning (DRL), Proximal Policy Optimization (PPO), Transformer XL (TrXL) and Gated Transformer XL (GTrXL)

1. Introduction

In recent years, reinforcement learning (RL) has become instrumental in training agents for autonomous navigation [1]. However, a significant challenge remains: enabling agents to generalize their learned policies to unseen maze configurations. To address this, we propose integrating transformer-based models [2], like Transformer XL (TrXL) and Gated Transformer XL (GTrXL), with state-of-the-art RL algorithms, specifically Proximal Policy Optimization (PPO) [3]. This combination aims to equip agents with the ability to generalize effectively across diverse maze layouts. Transformer models excel in capturing long-range dependencies and abstract spatial representations, making them particularly suited for learning navigation tasks in complex maze environments.

2. Background

Reinforcement learning algorithms have shown success in maze navigation, yet struggle with high-dimensional and partially observable state spaces. Transformer-based models, initially developed for natural language processing, offer promising solutions. Transformer XL (TrXL) by Dai et al. (2019) extends the architecture with recurrent mechanisms for long-term dependency capture [4]. Gated Transformer XL (GTrXL), proposed by Child et al. (2019), enhances temporal modeling and representation learning with gated recurrent units (GRUs) within transformer blocks [5]. Integrating these models with Proximal Policy Optimization (PPO) can address these challenges effectively.

3. Problem Formulation

In maze navigation tasks, the primary aim is to train an agent to navigate complex environments to reach a goal state, optimizing a predefined reward signal. The challenge lies in the generalization problem, where agents struggle to apply learned policies effectively to unseen maze configurations.

Formally, let S denote the state space of the maze environment, A the action space, and $r(s, a, s')$ the reward function, where $s, s' \in S$ represent states and $a \in A$ actions taken by the agent. The goal is to learn a policy $\pi(a \mid s)$ that maximizes the expected cumulative reward over a trajectory:

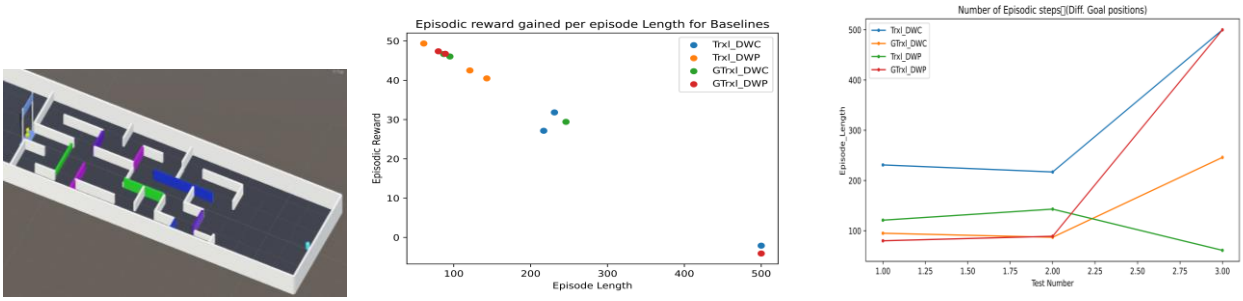
$$\max_{\pi} \mathbb{E}_{\tau \sim \pi}[R(\tau)] = \max_{\pi} \mathbb{E}_{\tau}\left[\sum_{t=0}^{T} \gamma^t r_t\right] \quad (1)$$

where $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots)$ represents a trajectory, and $R(\tau)$ is the discounted cumulative reward. The generalization challenge emerges when the learned policy fails to adapt to unseen maze configurations, hindering the agent's performance.

4. Methodology

Our approach combines DRL algorithms with transformer-based models, specifically TrXL and TrXL, to tackle the generalization problem in maze navigation tasks.

Custom Maze Environment: We design a maze environment $\mathcal{E}$ using Unity 3D, providing agents with visual observations O and rewards $r(s, a, s')$ based on their navigation performance.



(a) (b) (c)

Figure 1: (a) Maze Environment, (b) Episode Reward vs Episode length for all variations, (c) Episode length vs Test no. for DWC and DWP, DWC – Different Wall Color, DWP – Different Wall Positions

Reinforcement Learning Algorithms: We utilize Proximal Policy Optimization (PPO) to train agents in the maze environment. The objective function for PPO is defined as:

$$\max_{\theta} \mathbb{E}_{\tau \sim \pi_\theta} \left[\frac{\pi_\theta(a \mid s)}{\pi_{\theta_{old}}(a \mid s)} A^{Clip}(\tau) \right] \quad (2)$$

Where θ are the parameters of the policy network π_θ, τ represents a trajectory, and $A^{Clip}(\tau)$ is the clipped surrogate objective.

Transformer-Based Models: To capture long-term dependencies and generalize across maze configurations, we integrate Transformer XL and Gated Transformer XL into our architecture. These models utilize self-attention mechanisms to process sequences effectively and adapt to various maze topologies.

Training Procedure: During training, agents interact with the environment, collecting experience tuples (s, a, r, s'). The policy network parameters are updated iteratively using stochastic gradient ascent to maximize the expected cumulative reward.

Evaluation Metrics: We evaluate the agents' generalization performance using metrics such as success rate S and efficiency E in navigating unseen maze configurations. Success rate is defined as the proportion of successful navigation episodes, while efficiency measures the average reward taken to reach the goal state.

5. Experiment setup

The experimental setup focused on a single, dynamically changing maze environment characterized by diverse colors and variable sizes, enabling agents to learn complex navigation behaviors, including jumping over obstacles. Figure 1(a) illustrates the dynamic nature of the maze environment. During training, agents underwent Proximal Policy Optimization (PPO) training with custom reward shaping, leveraging a neural network architecture blending convolutional and transformer layers. Distributed training techniques were employed to expedite learning, utilizing multiple CPU cores and GPUs. Training continued until convergence, with intermittent evaluations to assess generalization. In contrast, testing involved a multitude of distinct maze environments to evaluate the agent's adaptability across varied scenarios.

6. Results

The evaluation tested TrXL and GTrXL models across two scenarios and their variations to assess adaptability to maze configurations.
Scenario 1: Different Wall Colors (Variations: Color Swap, Added New Color, Removed Original Colors)
Scenario 2: Different Wall Size or Position (Variations: Wall Swap, Wall Size, Adding New Walls.)
TrXL and GTrXL were evaluated in maze environments with varied configurations (shown in Table 1). While TrXL achieved a success rate of 66.67% in scenarios with different wall colors (DWC) and 100% in scenarios involving changing wall positions (DWP), GTrXL exhibited higher success rates of 100% in DWC

Table 1: Success rate and efficiency.

Baselines	Scenarios	Success rate (SR)	Efficiency
TrXL	DWC	66.67 %	39.14 %
	DWP	100 %	90.13 %
GTrXL	DWC	100 %	83.10 %
	DWP	66.67 %	61.30 %

scenarios but slightly lower at 66.67% in DWP scenarios. In terms of efficiency, GTrXL outperformed TrXL in both scenarios, with efficiencies of 83.10% and 61.30% in DWC and DWP scenarios, respectively, compared to TrXL's 39.14% and 90.13%.

These results highlight GTrXL's superior adaptability to diverse maze layouts, demonstrating its potential for

real-world applications. Handling the complexity of unseen scenarios is enhanced by Gated Transformer XL (GTrXL) models and will be further explored through the integration of chunking and forgetting mechanisms to achieve better generalization.

7. Conclusion

Our study demonstrates the effectiveness of Transformer XL (TrXL) and Gated Transformer XL (GTrXL) models in diverse maze environments. GTrXL showed superior efficiency over TrXL, achieving high success rates across scenarios. These findings underscore the potential of transformer-based models, particularly GTrXL, in reinforcement learning generalization. Future research may explore hyperparameter optimization, meta-learning, and real-world evaluations to enhance adaptability and robustness. Additionally, investigating ensemble methods with transformer-based models could improve performance in diverse settings.

8. References

1. Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). *Human-level control through deep reinforcement learning*. Nature, 518(7540), 529-533.

2. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). *Attention is all you need*. Advances in Neural Information Processing Systems (NeurIPS), 30, 5998-6008.

3. Schulman, J., Wolski, F., Dhariwal, P., et al. (2017). *Proximal policy optimization algorithms*. arXiv preprint arXiv:1707.06347.

4. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2019). *Transformer-XL: Attentive language models beyond a fixed-length context*. arXiv preprint arXiv:1901.02860.

5. Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). *Generating long sequences with sparse transformers*. arXiv preprint arXiv:1904.10509.

6. Parisotto, E., Ba, J., Salakhutdinov, R., et al. (2019). *Stabilizing transformers for reinforcement learning*. Advances in Neural Information Processing Systems (NeurIPS), 32, 6213-6224.

7. Rueckauer, B., Lungu, I. A., Hu, Y., Pfeiffer, M., & Liu, S. C. (2017). *Deep episodic memory: Encoding, recalling, and predicting episodic experiences for robot action execution*. IEEE Transactions on Cognitive and Developmental Systems, 10(4), 1005-1018.

8. Wang, L., & Li, X. (2020). *Machine learning techniques for cognitive decision making.* Information Fusion, 64, 30-39.

9. Khan, Z., Anjum, A., & Ali, H. (2018*). Review of AI techniques and cognitive computing framework for intelligent decision support*. International Journal of Intelligent Systems and Applications in Engineering, 6(3), 25-36.

.