Beating the World Record with AI: PPO Experiments in Super Mario Bros

Joel Saji Varghese
School of Computer Science, Algoma University
Brampton, Ontario, Canada
jovarghese@algomau.ca

ABSTRACT

This project explores the application of two reinforcement learning methods, Proximal Policy Optimization (PPO), for training agents on several levels of Super Mario Bros. One approach trains a separate agent for each level, while the other trains a general agent using dynamic level sampling. We compare performance between both approaches with customized reward shaping and measure completion rates, overall scores, and generalization to novel challenges. Results show that even though per-level agents achieve rapid mastery, generalized agents provide greater versatility.

1 INTRODUCTION

Deep reinforcement learning (RL) has been capable of solving hard video game issues, specifically in heterogeneous challenges and sparse reward settings. In this paper, we train AI agents on Super Mario Bros using PPO and try two training methods:

- **Approach 1:** Train a separate model for each level (1-1 to 1-4).
- **Approach 2:** Train one generalized model with dynamic sampling.

Generalization in this case is the ability of the agent to learn policies that can be reused and are effective in various level layouts, enemy configurations, platforming setups, and reward schemes—without needing to be trained anew.

2 RELATED WORK

RL has been applied to platformers with DQN, A3C, and PPO. Much early work concentrated on single-task training. Transfer learning, curriculum learning, and multi-task RL were explored more recently to promote generalization. Our extension builds upon this by introducing adaptive level sampling and comparing it to isolated training.

3 METHODOLOGY

3.1 Environment and Preprocessing

We use `gym-super-mario-bros`, an NES emulator interface built on OpenAI Gym. Observations are grayscale, resized to 84x84, and frame-stacked. The action space includes 7 discrete combinations of right movement and jump controls.

Observation: $84 \times 84 \times 4$ frames
Actions: Right, Right+A, Right+B, Right+A+B, A, Down, Down+Right

3.2 Reward Shaping

The sparsity of the game in the reward structure was addressed with a shaped reward:

- +0.1× horizontal progress
- +300 on level completion
- +0.05× remaining time
- −50 penalty on failure without flag

3.3 Approach 1: Per-Level PPO Agents

Here, we have four independent PPO models, one for each level from 1-1 to 1-4. Each of these models has 12 parallel workers with early stopping when the completion is above 95%.

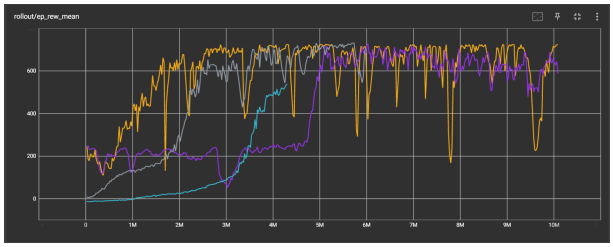

Figure 1: Approach 1: Separate Per-Level PPO Agents Reward Graph

3.4 Approach 2: Generalized PPO with Dynamic Sampling

This approach trains a single PPO agent across all levels. Worker environments are assigned levels based on sampling weights:

$$w = \max(0.2, 1 - \text{completion\_rate}^2)$$

The agent updates its sampling probabilities after every training chunk (e.g., 1M steps). At least one worker remains on every level so as not to forget.

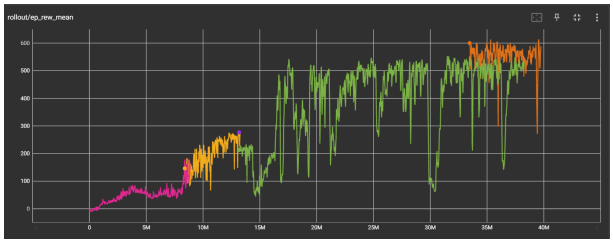

Figure 2: Approach 2: Generalized PPO with Dynamic Sampling Reward Graph

4 IMPLEMENTATION DETAILS

- PPO (from Stable-Baselines3) with `CnnPolicy`
- Learning rate: 1×10^{-4}, entropy coefficient: 0.01 (raised to 0.05 during resumed training)
- Steps per update: 2048, batch size: 256
- Frame stack: 4, Max steps per episode: 8000
- Hardware: 4060 NVIDIA RTX GPU, 12 CPU workers

5 RESULTS

5.1 Score Comparison

Table 1: World Record vs AI Performance

Level	WR	4-Model PPO	Dynamic PPO	Time (s)
1-1	370	357	**366**	4
1-2	350	333	**338**	12
1-3	260	247	–	–
1-4	263	259	**261**	2

5.2 Findings

- All per-level agents converged after 5M steps per level.
- The generalized model outperformed on 3 out of 4 levels.
- Level 1-3 lacked sufficient training samples under the dynamic strategy.
- Increased entropy improved exploration and suppressed local optima.

6 DISCUSSION

6.1 Generalization Defined

Generalization refers to the capability of the agent to act well on levels it has not been well-trained on by capitalizing on common patterns, behaviors, and dynamics learned on other levels.

6.2 Key Takeaways

- Per-level training is efficient for specific mastery.
- Generalized PPO improves policy reuse and robustness.
- Dynamic sampling mitigates overfitting and forgetting.

7 LIMITATIONS

- The generalized model failed to fully master level 1-3, possibly due to under-sampling or complexity.
- PPO's sample inefficiency led to high compute costs.
- No testing was done beyond World 1—future work should evaluate scalability.
- We didn't apply replay memory or curriculum progression which could aid generalization further.
- Training was limited to 4 levels—real generalization would require testing across dozens.

8 CONCLUSION

We compared two PPO training paradigms on Super Mario Bros. Standalone models provide quick results but less scalable and generalizable agents. Generalized PPO with dynamic sampling showed promising generalization capabilities and efficient reuse of learned policies. Future work includes curriculum learning, meta-RL, and attention-based memory for storing cross-level knowledge.

ACKNOWLEDGMENTS

Thanks to OpenAI Gym, NES-Py, and the Stable-Baselines3 team. Special thanks to the Mario speedrunning community for WR benchmarks.

REFERENCES

[1] Schulman, J., et al., "Proximal Policy Optimization Algorithms," *arXiv:1707.06347*, 2017.
[2] Kauten, J., "gym-super-mario-bros." GitHub. https://github.com/Kautenja/gym-super-mario-bros
[3] Raffin, A., et al., "Stable-Baselines3," *JMLR*, 2021.