

Project Overview

Title: "Systematic Analysis of Scalable Reinforcement Learning for Dots and Boxes: From 3x3 to Large Boards"

Research Gap: Existing work on larger boards is fragmented, lacks systematic comparison, and ignores computational efficiency **Your Contribution:** First comprehensive, systematic study of RL scalability with CPU-efficiency focus

Research Questions

Primary Research Questions:

1. **Scalability:** How do different RL algorithms perform as board size increases from 3x3 to 6x6+?
2. **Efficiency:** Which RL methods are most computationally efficient for larger boards on CPU-only systems?
3. **Comparative Analysis:** What are the relative strengths/weaknesses of different RL approaches across board sizes?

Secondary Research Questions:

1. **Breaking Points:** At what board size does each algorithm start failing?
2. **State Representation:** How does state encoding affect scalability?
3. **Training Efficiency:** Which methods require least training time for acceptable performance?
4. **Human-Level Performance:** Can we achieve human-level play on larger boards efficiently?

Literature Foundation & Baselines

Existing Work to Build On:

- **3x3 Baselines:** Pandey (2022), da Costa (2022), BoxesZero (2025)
- **Larger Board Attempts:** Miller et al. (6x6), Deakos (5x5), ChantalIMP (12x12)
- **Implementations:** Multiple GitHub projects with varying approaches

Your Systematic Approach:

- **Reproduce key results** from existing 3x3 work
- **Systematically extend** to 4x4, 5x5, 6x6 boards
- **Compare methods** that others tested individually
- **Add CPU-efficiency analysis** (novel contribution)

Methodology Framework

Board Sizes for Testing:

- **3x3:** Baseline comparison with existing work
- **4x4:** First scaling step
- **5x5:** Medium complexity (matches Deakos)
- **6x6:** High complexity (matches Miller et al.)

- **Larger if feasible:** Push boundaries of CPU capabilities

RL Algorithms to Compare:

1. Classical Methods:

- Q-Learning (tabular for small boards)
- Deep Q-Network (DQN)
- Policy Gradient (PPO/A2C)

2. Advanced Methods:

- AlphaZero-style (MCTS + NN)
- Actor-Critic variants
- N-Tuple networks (from MarkusThill)

3. Hybrid/Novel Approaches:

- Rule-based + RL combinations
- Transfer learning across board sizes
- CPU-optimized variants

State Representations to Test:

- **Binary grids:** Simple edge representation
- **Structured features:** Chains, boxes, strategic features
- **Convolutional:** 2D spatial representation
- **Graph-based:** Explicit game structure

Evaluation Metrics:

- **Performance:** Win rate vs. baselines (random, heuristic, human)
- **Efficiency:** Training time, memory usage, inference speed
- **Scalability:** How metrics degrade with board size
- **Robustness:** Performance across different opponents

Experimental Design

Phase 1: Foundation & Reproduction

Goal: Establish solid baselines and reproduce key existing results

Tasks:

- Implement/adapt existing 3x3 algorithms
- Reproduce key results from literature
- Establish evaluation protocols
- Create systematic testing framework

Deliverables:

- Working implementations of 3-4 RL algorithms

- Validated results on 3x3 boards
- Standardized evaluation pipeline

Phase 2: Systematic Scaling Analysis

Goal: Comprehensive comparison across board sizes

Experimental Matrix:

Algorithm × Board Size × State Representation × Evaluation Metric

Key Experiments:

- **Algorithm Comparison:** Same setup, different algorithms
- **Scaling Analysis:** Same algorithm, different board sizes
- **Representation Impact:** Same algorithm, different state encodings
- **Efficiency Analysis:** Training time vs. performance trade-offs

Statistical Rigor:

- Multiple random seeds for each experiment
- Proper significance testing
- Confidence intervals for all results
- Reproducibility documentation

Phase 3: CPU-Efficiency Focus

Goal: Novel contribution focused on computational constraints

Efficiency Experiments:

- **Training Efficiency:** Time to reach acceptable performance
- **Memory Usage:** RAM requirements across board sizes
- **Inference Speed:** Decision time during play
- **Scalability Limits:** Maximum feasible board size per method

CPU-Optimized Variants:

- Simplified neural network architectures
- Efficient state representations
- Approximate methods for large boards
- Hybrid approaches combining fast heuristics with RL

Phase 4: Advanced Analysis & Novel Contributions

Goal: Push beyond existing work with new insights

Advanced Experiments:

- **Transfer Learning:** Train on small boards, test on large

- **Curriculum Learning:** Progressive board size training
- **Multi-Agent Analysis:** Self-play vs. diverse opponents
- **Theoretical Analysis:** Complexity bounds, convergence analysis

Novel Algorithmic Contributions:

- CPU-efficient variants of existing methods
- Hybrid rule-based + RL approaches
- Board-size adaptive algorithms
- Computational budget allocation strategies

Expected Contributions

Primary Contributions:

1. **Systematic Scalability Analysis:** First comprehensive study of RL scaling in Dots and Boxes
2. **CPU-Efficiency Focus:** Novel analysis of computational constraints in game RL
3. **Comparative Methodology:** Standardized evaluation framework for future research
4. **Practical Insights:** Clear guidance on which methods work best for different scenarios

Secondary Contributions:

1. **Algorithmic Improvements:** CPU-optimized variants of existing methods
2. **Theoretical Insights:** Understanding of why certain methods scale better
3. **Reproducible Research:** Open-source implementations and datasets
4. **Benchmark Establishment:** Standard evaluation protocols for larger boards

Target Conferences & Positioning

Primary Targets:

- **AAMAS:** Multi-agent systems, game theory focus
- **IJCAI:** AI applications, systematic studies
- **CoG:** Conference on Games (specialized venue)

Secondary Targets:

- **AAAI:** General AI, practical applications
- **AIIDE:** Interactive entertainment, games
- **Various workshops:** At NeurIPS, ICML, etc.

Paper Positioning:

- **Systematic study** (not just novel algorithm)
- **Practical focus** (CPU efficiency, scalability)
- **Reproducible research** (open source, clear methodology)
- **Bridging theory and practice** (academic rigor + practical constraints)

Technical Implementation Plan

Development Environment:

- **Python 3.8+** with standard ML libraries
- **PyTorch** for deep learning (CPU optimized)
- **OpenAI Gym** interface for environments
- **Weights & Biases** for experiment tracking
- **Git + GitHub** for version control

Code Structure:

```
dots_boxes_rl/  
├── environments/      # Game implementations  
├── agents/           # RL algorithm implementations  
├── experiments/      # Systematic experiment scripts  
├── analysis/         # Result analysis and visualization  
├── baselines/        # Existing work reproduction  
└── utils/           # Shared utilities
```

Reproducibility Requirements:

- **Fixed random seeds** for all experiments
- **Detailed logging** of hyperparameters and results
- **Docker containers** for consistent environments
- **Comprehensive documentation** of all procedures
- **Open-source release** of all code

Success Metrics

Technical Success:

- **Reproduction:** Successfully reproduce 3+ existing results
- **Scaling:** Demonstrate systematic scaling analysis up to 6x6
- **Efficiency:** Show clear computational efficiency comparisons
- **Novel Insights:** Identify at least 2 new algorithmic improvements

Publication Success:

- **Paper Acceptance:** Target tier 2-3 conference acceptance
- **Reproducibility:** All results independently verifiable
- **Impact:** Citations from follow-up work
- **Open Source:** Community adoption of code/benchmarks

Personal Success:

- **Deep RL Understanding:** Master multiple RL algorithms
- **Research Skills:** Develop systematic experimental methodology
- **Technical Skills:** Advanced Python/PyTorch proficiency
- **Academic Writing:** Produce publication-quality paper

Risk Mitigation

Technical Risks:

- **Computational Limits:** Focus on CPU-efficient methods, smaller boards if needed
- **Implementation Bugs:** Extensive testing, reproduce known results first
- **Experimental Complexity:** Start simple, add complexity gradually

Research Risks:

- **Limited Novelty:** CPU-efficiency angle provides clear differentiation
- **Negative Results:** Systematic failure analysis is still valuable
- **Scope Creep:** Well-defined research questions with clear boundaries

Timeline Risks:

- **Reproduction Takes Too Long:** Use existing implementations where possible
- **Experiments Don't Converge:** Have backup simpler algorithms
- **Writing Delays:** Start writing early, parallel to experiments

Resources & Tools

Computational Resources:

- **Your CPU:** AMD 5600G (sufficient for this project)
- **Cloud Computing:** Consider Google Colab/Kaggle for large experiments
- **Storage:** Local + cloud backup for all experimental data

Software Tools:

- **Development:** VS Code, Jupyter notebooks
- **Experiment Tracking:** Weights & Biases, TensorBoard
- **Visualization:** Matplotlib, Seaborn, Plotly
- **Writing:** LaTeX, Overleaf
- **Reference Management:** Zotero, Mendeley

Learning Resources:

- **RL Textbooks:** Sutton & Barto, Bertsekas
- **Online Courses:** Spinning Up, CS234 Stanford
- **Paper Repositories:** ArXiv, Google Scholar alerts
- **Code Examples:** GitHub repositories, research reproductions

Expected Outcomes

Academic Impact:

- **Systematic Understanding:** Clear picture of RL scalability in Dots and Boxes
- **Methodological Contribution:** Framework for systematic game RL evaluation
- **Practical Insights:** Guidance for practitioners with limited computational resources

- **Future Research:** Foundation for more advanced work in this area

Technical Outcomes:

- **Open Source Framework:** Reusable code for Dots and Boxes RL research
- **Benchmark Suite:** Standard evaluation protocols and baselines
- **Algorithmic Improvements:** CPU-efficient variants of existing methods
- **Empirical Database:** Comprehensive results across methods and board sizes

Personal Development:

- **Research Expertise:** Deep understanding of RL and systematic experimentation
- **Technical Skills:** Advanced ML engineering and experimental design
- **Academic Writing:** Publication-quality research communication
- **Domain Knowledge:** Expertise in game AI and computational efficiency

This research plan positions you to make meaningful contributions to an active research area while building on existing work systematically. The CPU-efficiency focus provides a clear novel angle that distinguishes your work from existing fragmented efforts.