**On the Probabilistic, Neural, and Evolutionary Design of a Battleship Agent**

*Author: Abhinav (with assistance)*

Date: 2025-09-07

# Abstract

We present a complete research platform for the game of Battleship combining classical probabilistic placement counting, a convolutional heatmap prior, Monte Carlo posterior blending, information-theoretic scoring, and genetic optimization of meta-weights (Agent 4). We describe the system architecture, training pipelines (supervised, RL fine-tuning, GA), and evaluation. A quick benchmark demonstrates robust performance versus heuristic opponents. We release full documentation and reproducibility materials.

# 1. Introduction

The Battleship problem is a partially observable search task. Our system fuses a neural prior with probabilistic placement counting and Monte Carlo sampling to infer likely ship locations. Agent 3 integrates opponent modeling and information gain; Agent 4 loads GA-optimized weights to maximize win rate while minimizing moves.

# 2. System Overview

Environment: 10×10 board, classic ships [5,4,3,3,2]; headless engine for AI-vs-AI.

Agents: Agent 2 (probabilistic+NN+MC), Agent 3 (adds info gain + opponent modeling + graph), Agent 4 (GA-optimized blend).

Training: supervised heatmap CNN from self-play data; optional REINFORCE fine-tuning; GA evolution of meta-weights.

# 3. Methods

Density counting: enumerate legal placements under misses/sunk constraints; parity reduces early search.

Neural prior: CNN maps (miss,hit,unknown)→ per-cell ship probability.

Monte Carlo posterior: sample full-consistent placements to build occupancy frequencies.

Information gain: expected entropy reduction via MC p(hit).

Meta-weights: blend grids (density, neural, MC, info, opponent) and boost neighbours of unsunk hits.

GA: tournament selection, uniform crossover, Gaussian mutation; fitness = 100×win\_rate − avg\_moves.

# 4. Implementation

Codebase covers agents, training scripts, and a Tk dashboard. Logs, datasets, and models reside under data/models/logs.

# 5. Experiments

Quick benchmark (10 games/opponent; AIAgent4 as main).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| opponent | games | main\_ai\_wins | main\_ai\_losses | avg\_moves\_to\_win |
| UltimateBattleshipAgent | 10 | 8 | 2 | 42.0 |
| NaiveAgent6 | 10 | 10 | 0 | 45.2 |
| AIPlayer2 | 10 | 5 | 5 | 40.8 |

# 6. Discussion

Strengths: complete system; robust blend; GA tuning. Limitations: opponent prior not yet fully folded into inference; RL gains not extensively reported; evaluation vs stronger public baselines is future work.

# 7. Reproducibility

See docs/: setup, seeds, commands; scripts for dataset generation, training, and GA.

# 8. Conclusion

A practical, extensible Battleship AI stack; future work: end-to-end opponent prior integration, rigorous ablations, larger-scale evaluation.