Problem Formulation

Angad Ahuja

February 3, 2026

1 Problem Setup and Modelling Rationale

1.1 Qualitative Description of the Game

Battleship is a sequential search game played on an $H \times W$ grid where $H, W \in \mathbb{N}^+$ standardly $H = W = 10$. A set of ships is placed on the grid according to fixed legality rules including but not limited to contiguity, no overlap, and within bounds. The player does not observe the ship locations directly. Instead, the player repeatedly selects a grid cell to fire at and receives immediate feedback indicating whether the shot was a miss, a hit, or whether a particular ship has been sunk which is defined as all the cells occupied by the ship have been hit. The episode ends when all ships have been sunk. The natural performance criterion is therefore efficiency defined by optimal policy: the fewer shots required to sink all ships, the better.

1.2 Board, Cells, and Hidden Layout

Let the board be an $H \times W$ grid and let the set of cells be

$$\mathcal{C} \;=\; \{1, 2, \ldots, HW\}.$$

We introduce a latent ship layout B that encodes ship occupancy and ship identity/segment structure. At minimum, occupancy is represented by

$$B_{\text{occ}} \in \{0,1\}^{H \times W}.$$

The layout is treated as latent because the player never directly observes ship positions; only indirect outcomes (hit/miss/sunk) are observed. Including ship identity/segments in B (beyond pure occupancy) is necessary because the observation alphabet includes "sunk" events, which depend on whether all segments of a specific ship have been hit.

Let $\mathcal{B}$ denote the set of legal layouts (satisfying placement constraints). Restricting to $\mathcal{B}$ formalizes game rules and ensures the latent variable ranges only over physically possible configurations.

1.3 Single-Agent Formulation as a POMDP

We model the game as a partially observable Markov decision process (POMDP)

$$\mathcal{M} \;=\; (\mathcal{S}, \mathcal{A}, \mathcal{O}, T, Z, \gamma).$$

A state is decomposed as

$$\mathcal{S} \ni s_t \;=\; (B,\; M_t,\; H_t,\; U_t),$$

where

$$M_t \in \{0,1\}^{H \times W} \quad \text{(miss indicator)}, \qquad H_t \in \{0,1\}^{H \times W} \quad \text{(hit indicator)},$$

and U_t denotes the remaining (not-yet-sunk) ship structure. The POMDP is appropriate because the game is Markovian in the underlying configuration, but the agent has incomplete information. The decomposition separates (i) the fixed hidden cause of outcomes (B) from (ii) the public record of interaction (M_t, H_t, U_t). This mirrors the actual gameplay: the board configuration does not change, but the agent's knowledge does.

Initial distribution. At the start of an episode:

$$B \sim p_0(\cdot) \text{ on } \mathcal{B}, \qquad M_0 = \mathbf{0}, \quad H_0 = \mathbf{0}, \quad U_0 = U_{\text{all}}.$$

The randomness of the game (under standard rules) comes from the ship placement. After placement, the environment is deterministic given the sequence of shots.

1.4 Action Space and Feasibility Constraints

Actions correspond to firing at a cell $a_t \in \mathcal{C}$. The feasible action set is

$$\mathcal{A}(s_t) \;=\; \{a \in \mathcal{C} : M_t(a) = 0 \;\wedge\; H_t(a) = 0\}.$$

Re-firing at the same cell provides no new information and is not permitted in standard play. Encoding feasibility at the action-space level enforces the rules and avoids degenerate policies that waste moves on already-resolved cells.

1.5 Observation Space and Observation Function

Observations are shot outcomes:

$$\mathcal{O} \;=\; \{\mathsf{miss},\ \mathsf{hit}\} \;\cup\; \{\mathsf{sunk}(k) : k \in \mathcal{K}\} \;\cup\; \{\mathsf{terminal}\},$$

where $\mathcal{K}$ indexes ship types or identities. Observations are generated by a deterministic function

$$o_t \;=\; g(B,\ M_t,\ H_t,\ U_t,\ a_t), \qquad Z(o_t \mid s_t, a_t) \;=\; \mathbf{1}\{o_t = g(\cdot)\}.$$

In the standard game, the feedback is fully determined by (i) the hidden placement and (ii) the shot history so far; there is no additional noise. Including $\mathsf{sunk}(k)$ captures the extra information provided when a ship is completed, which affects future inference.

1.6 Transition Dynamics

Conditioned on B, transitions are deterministic:

$$(M_{t+1}, H_{t+1}, U_{t+1}) \;=\; F(B,\ M_t,\ H_t,\ U_t,\ a_t).$$

The environment does not "move" ships; it only updates the public record based on the shot outcome and updates which ships remain. This makes the dynamics simple and well-specified while preserving the partial observability through B.

1.7 Termination and Objective (Shots-to-Win)

Define the termination time

$$\tau \;=\; \inf\{t \geq 0 : U_t = \varnothing\},$$

the first time all ships are sunk. The problem-level objective is

$$\min_{\pi}\ \mathbb{E}_{\pi}[\tau].$$

The primary notion of skill in Battleship is efficiency in discovering and clearing ships, which is naturally measured by shots-to-win. This objective is independent of any particular learning algorithm and matches common empirical evaluation protocols.

1.8 Adversarial Two-Player Extension (Markov Game)

To model an adversarial ship placer, we define a two-player zero-sum Markov game:

- Defender (Player D) chooses a legal layout $B \in \mathcal{B}$ at $t = 0$.
- Attacker (Player A) then plays the POMDP given that fixed B.

Let π_D be a distribution over layouts and π_A an attacker policy. Using shots-to-win as the payoff:

$$\max_{\pi_D}\ \min_{\pi_A}\ \mathbb{E}_{B\sim\pi_D}\big[\tau(\pi_A, B)\big].$$

Standard Battleship assumes a fixed placement distribution (often uniform). The adversarial framing replaces this with a strategic opponent, enabling robustness analysis: the attacker should perform well not only on typical layouts but also on "hard" layouts chosen to increase search difficulty.

1.9 Belief-State Formalisation

Let the interaction history be

$$h_t \;=\; (o_1, a_1, o_2, a_2, \ldots, o_t).$$

Define the posterior over layouts:

$$b_t(B) \;=\; \mathbb{P}(B \mid h_t), \qquad B \in \mathcal{B}.$$

The belief update after observing o_t upon firing at a_t is:

$$b_{t+1}(B) \;=\; \frac{b_t(B)\,\mathbf{1}\{o_t = g(B, M_t, H_t, U_t, a_t)\}}{\sum_{B' \in \mathcal{B}} b_t(B')\,\mathbf{1}\{o_t = g(B', M_t, H_t, U_t, a_t)\}}.$$

The belief state formalizes the agent's uncertainty about the latent layout given the public record. This is the canonical sufficient statistic for decision-making in POMDPs and cleanly separates the information structure of the problem from any particular computational approximation.