

When Does Opponent Modeling Pay Off? Spatial Inference Stabilises Reinforcement Learning under Partial Observability

Marcel Petran

Faculty of Mathematics and Physics, Charles University
Prague, Czech Republic

Stefan Edelkamp

Faculty of Mathematics and Physics, Charles University
Prague, Czech Republic

Abstract

Deep reinforcement learning agents in partially observable competitive games must act on beliefs about an unseen opponent’s intent. We study a spatial opponent model: a transformer trained with hindsight-relabelled opponent goals that outputs a subgoal heatmap from the opponent’s observed trajectory, fused early into a dueling double DQN. In symmetric constant-sum *foraging* games with partial observability, the module stabilises training. Plain TD(0) DQN suffers degenerate seeds in three of four one-step configurations, whereas the opponent-model agent never collapses and, wherever training succeeds, recovers the majority of the symmetric game value. Transfers zero-shot to a *foraging* opponent unseen by the learner, with a positive goal advantage over the baseline, and loses this edge exactly where an information argument predicts: against a full-information tracking adversary whose intent is the learner itself, and on resource-dense maps where both agents saturate. A blindfold ablation, zeroing the heatmap at deployment, decomposes the advantage into two mechanisms: training-time stabilisation, which dominates where the credit assignment is hard and survives blinding on the dense map, and deployment-time inference, whose reliance shrinks as fog thins on the sparse map. We conclude that opponent-goal inference is a stability-and-transfer mechanism whose value is governed by measurable boundary conditions: visibility (fog thickness), opponent intent, and resource density.

Keywords

opponent modeling, partial observability, multi-agent reinforcement learning, goal recognition

ACM Reference Format:

Marcel Petran and Stefan Edelkamp. 2027. When Does Opponent Modeling Pay Off? Spatial Inference Stabilises Reinforcement Learning under Partial Observability. In *Proc. of the 26th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2027)*, Hanoi, Vietnam, 3–7 May 2027, IFAAMAS, 9 pages.

1 Introduction

Competitive *foraging* [6] under fog (Figure 1) is a Partially Observable Stochastic Game (POSG) in which the pivotal latent is not where the opponent is, but where it is going. Classic DQN-style learners are hard to train in this regime: delayed sparse rewards combined with one-step temporal-difference learning (TD(0)) destabilise

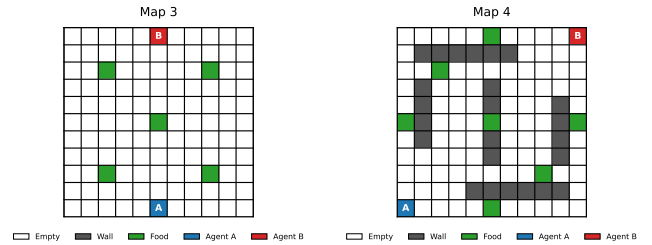


Figure 1: The two foraging maps (11×11). MAP3 (left): five densely placed central goals; MAP4 (right): seven goals dispersed behind sectioned walls. Agents and foods spawn at fixed positions and compete for the resources.

training. The ability to reason about others’ behaviour can often be a defining factor between success and failure [1, 8, 14, 15, 18, 19, 23]. Relying solely on reactive policies is often insufficient. Instead, a prominent approach is to model the adversary directly, actively inferring their underlying strategies, intentions, or goals to anticipate future states. This approach is generally known as Opponent Modeling (OM). OM is usually evaluated only as “beats the baseline,” so we ask when it helps. The spatial opponent model and its ESS formalism were introduced in prior work [16] for Fully Observable Stochastic Games (FOSG) [12]. Here we adapt it to partial observability, replacing privileged inputs with the fogged observation history and its first-order motion encodings, and fusing the inferred heatmap alongside learned belief channels. Using this adapted module, we contribute a boundary-condition analysis of when and why opponent-goal inference pays off.

Our contributions are: *POSG instantiation of the spatial OM*: The opponent model of prior work [16] re-instantiated for fogged observation histories, with motion-difference channels, hindsight supervision, and early spatial fusion in physical coordinates.

Stability: Across all training runs on two maps, the opponent-model agent never collapses, whereas plain DQN has degenerate seeds in three of four one-step configurations (every one-step configuration of the sparse map, and thick fog on the dense map).

In-distribution advantage: Where training succeeds, the opponent model recovers 95–99.9% of the symmetric game value against the training opponent (occasionally exceeding it by exploiting the suboptimal heuristic), versus 77–88% for the baseline on MAP4 and 52–107% per seed on MAP3, with degenerate TD(0) training runs finishing as low as 47%.

Generalisation across forager types: The paired advantage transfers to a *foraging* opponent unseen by the learner (+0.31–0.52) and, in well-trained scenarios, vanishes against a tracking adversary.



This work is licensed under a Creative Commons Attribution International 4.0 License.

Boundary conditions: The advantage disappears against opponents whose intent is the learner itself and saturates when resources are dense.

Mechanism: A blindfold ablation decomposes the advantage into training-time stabilisation and deployment-time inference, and shows inference reliance scales with fog.

2 Related Work

Opponent modeling has a long history, from theory-of-mind-style agent modeling [2, 17, 18] to game-specific predictors [4, 8]. Nashed and Zilberstein survey opponent-modeling methods and their design axes [14]. Closest to us, Yu et al. infer opponent *subgoals* and integrate them into policy learning [23]; their subgoal inference conditions on the full state, whereas we infer a spatial intent distribution from partial observations, and ask when the inference channel pays off rather than whether it improves a benchmark. Goal and plan recognition share our prediction target [5]; our hindsight-relabelled transformer (Section 4) is a lightweight online variant of such methods, using realised opponent outcomes as supervision in the spirit of HER [3]. Under partial observability, agent modeling has been tackled with recurrent opponent embeddings [15], Bayes-adaptive meta-learners [24], and policy-inference networks [10]; these predict opponent *policies or types*, not their spatial intent. Our games are partially observable stochastic games [7], and our contributions are orthogonal to MARL algorithmics: we modify the observation pipeline of a standard dueling DDQN [13, 21, 22] and measure the effect on training stability and transfer.

3 Problem Formulation

Definition 3.1 (Partially Observable Stochastic Game). A Partially Observable Stochastic Game (POSG) is a tuple

$$\Gamma = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{T}, R_i, \mathcal{O}, Z_i, \gamma \rangle$$

where:

- $\mathcal{N}$ is the set of N agents.
- $\mathcal{S}$ is the global state space.
- $\mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_N$ is the joint action space.
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ is the state transition function conditioned on joint actions, where $\Delta(\mathcal{S})$ denotes the probability simplex over the state space.
- $R_i : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, $i \in \mathcal{N}$, are the per-agent reward functions.
- $\mathcal{O}$ is the set of observation spaces $\mathcal{O} = \mathcal{O}_1 \times \dots \times \mathcal{O}_N$.
- $Z_i : \mathcal{S} \rightarrow \mathcal{O}_i$, $i \in \mathcal{N}$, are the per-agent observation functions mapping the global state to agent i 's observation space.
- $\gamma \in [0, 1)$ is the discount factor.

Foraging as a POSG. We model competitive foraging under partial observability as a two-player constant-sum POSG $\Gamma = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{T}, R_i, \mathcal{O}, Z_i, \gamma \rangle$ with agents $\mathcal{N} = \{1, 2\}$, $\mathcal{S}$ that represents the global state configurations that describe the positions of all agents, walls, foods, and the food's state (present or eaten). The agents share the movement action set $\mathcal{A} = \{\text{North, South, East, West}\}$. $\mathcal{T}$ is deterministic, meaning that the movement actions update the agent's coordinates to the adjacent cell, agents automatically collect items upon entering a cell containing food and if agent $i \in \mathcal{N}$ takes an action that would result in a collision with a wall or stepping

outside the boundaries of the map, agent i stays at its current position. Each episode places both agents and a finite set of food goals on a grid. Agent i receives a reward $r^i = 1$ if and only if it moves onto a cell containing food while the opponent is not present on that exact cell. If both agents enter the same cell containing food simultaneously, the food is split, yielding a reward of $r^i = 0.5$ for each agent. In all other scenarios, including empty movements and wall collisions, the step reward is $r^i = 0$. Agent i does not observe the true state $s_t \in \mathcal{S}$: it receives a local observation $o_t^{(i)} = Z_i(s_t)$ restricted to a vision radius v around its position, with cells outside the radius masked. Agents are also unable to see through walls. All agents additionally maintain *belief channels* b_i (estimated food and opponent positions, updated on sightings), initialised from the environment's opening state. The game is a partially observable multi-agent variant of the travelling-salesman problem, making the grids minimal instances of the problem class we ultimately target.

Explicit Sequential Subgoal Games. The ESS formalism and the spatial opponent model below were introduced in prior work [16]; we summarise them for completeness and describe our partial-observability adaptation. We now characterise the class of games for which inferring a spatial subgoal is the right prediction target. Many tasks are hierarchical: before winning, an agent must complete a sequence of intermediate objectives, or *subgoals*. Predicting the opponent's full future state is unnecessarily noisy: the state carries contextual information that constrains movement but does not define the objective. We therefore decompose the state space as a Cartesian product $\mathcal{S} = \mathcal{P} \times \mathcal{Q}$ where $\mathcal{P}$ is the *subgoal feature space*, the state variables whose values complete the atomic objective (e.g. an agent's coordinates), and $\mathcal{Q}$ is the *context feature space*, variables that constrain behaviour but do not define the objective (walls, free space). Let $\rho : \mathcal{S} \rightarrow \mathcal{P}$, $\rho(s) = p$ for $s = (p, q)$, be the projection onto $\mathcal{P}$.

Let $\mathcal{S}^+ = \{s \in \mathcal{S} \mid R(s) > 0\}$ be the rewarding states. We define the set of *explicit subgoals* $\mathcal{G} = \rho[\mathcal{S}^+] \subseteq \mathcal{P}$. In the foraging POSG, $\mathcal{P}$ is the pair of agent coordinates, $\mathcal{Q}$ collects walls and remaining food, and $\mathcal{G}$ is the finite set of food coordinates.

Definition 3.2 (ESS game). A POSG is classified as an ESS game if and only if there exists a non-trivial decomposition of the state space such that $\mathcal{S} = \mathcal{P} \times \mathcal{Q}$, and the resulting set of distinct subgoals $\mathcal{G}$ is finite.

4 The Spatial Opponent Model under Partial Observability

The Spatial Opponent Model (Spatial OM) was introduced for fully observable games in prior work [16]. Relative to that instantiation, partial observability requires three changes: (i) inference over a sliding window of fogged observations rather than the full state, (ii) motion-difference channels as the primary sighting signal, and (iii) offline pre-training; in preliminary runs without it, the fusion channel is ignored by the Q-network and does not recover. The prediction target is unchanged: the adversary's physical destination as a dense heatmap over the grid, with the ESS formalism restricting predictions to meaningful destinations.

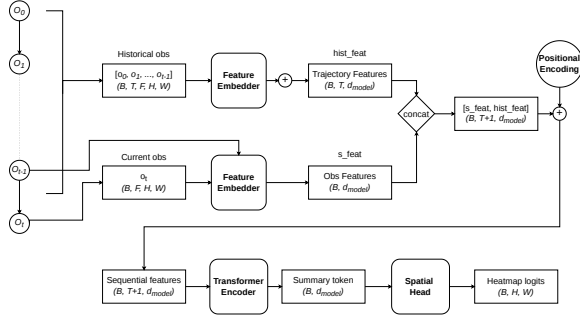


Figure 2: Overview of the Transformer-based Spatial Opponent Model. The Transformer Encoder processes the sequence of past observations, attending to the current observation to output a 2D heatmap representing the inferred opponent subgoals. The Feature Embedder is detailed in Figure 3. The diagram also shows expected tensor shapes where: B is the batch dimension, T is the trajectory dimension, F is the environment Feature dimension, H and W are the environment Height and Width dimensions, and d_{model} is the embedding output dimension.

Inference Model. The core of our module is the inference model p , parameterised by ϕ (visualised in Figure 2). It approximates the opponent’s intended spatial destination at timestep t , given the learner’s observation history and current observation, as a raw logit map $\hat{G}_{logits,t} \in \mathbb{R}^{H \times W}$ over the grid.

Let $\tau_t = (o_{t-T}, \dots, o_{t-1})$ be the learner’s observation history over a fixed-length sliding window of maximum length T (zero-padded for $t < T$). The architecture processes this sequence as follows:

- (1) **Feature Embedding:** Each observation in τ_t and the current observation o_t are processed through a shared feature embedding. As depicted in Figure 3, the spatial tensor $(2F \times H \times W)$, formed by concatenating the observation with its first-order motion encoding $o - o_{prev}$, passes through a Convolutional layer with ReLU activation to extract local spatial features. The resulting feature map is downsampled, flattened and projected via a fully connected layer into a continuous vector of dimension d_{model} . The motion channels are the key partial-observability signal: a negative difference marks a cell that just *disappeared under fog* or the opponent leaving the vision radius.
- (2) **Transformer Encoder:** A multi-layer Transformer Encoder with standard positional encodings processes the sequence; a boolean key-padding mask prevents self-attention from attending to unobserved history.
- (3) **Spatial Head:** To generate the prediction, the model extracts the “summary token” (the first output token, corresponding to the current observation o_t) and passes it through a shallow MLP that projects the d_{model} vector back into the spatial dimensions $(H \times W)$, yielding the heatmap logits $\hat{G}_{logits,t} \in \mathbb{R}^{H \times W}$.

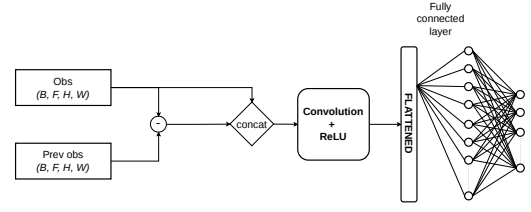


Figure 3: Feature embedding process inside the Spatial Opponent Model.

We define the probability of the opponent targeting a specific coordinate $(i, j) \in H \times W$ as the output of the inference model p_ϕ . The Spatial Head produces a raw logit map, $\hat{G}_{logits,t} \in \mathbb{R}^{H \times W}$. To ensure the final output represents a valid spatial probability distribution, we apply a Softmax activation function. This produces the predicted spatial intent heatmap $\hat{G}_t \in [0, 1]^{H \times W}$, where each element represents the probability of the opponent targeting a specific coordinate (i, j) conditioned on the learner’s current observation o_t and observation history τ_t :

$$\hat{G}_t^{(i,j)} = p_\phi(g_t = (i, j) \mid o_t, \tau_t) = \frac{\exp(\hat{G}_{logits,t}^{(i,j)})}{\sum_{x=1}^H \sum_{y=1}^W \exp(\hat{G}_{logits,t}^{(x,y)})} \quad (1)$$

where the resulting heatmap $\hat{G}_t$ is constrained such that the total probability mass sums up to 1:

$$\sum_{i=1}^H \sum_{j=1}^W \hat{G}_t^{(i,j)} = 1 \quad (2)$$

By treating the opponent’s observable past as a sequence of tokens, the Transformer leverages self-attention to capture long-range temporal dependencies between historical movements and current intent. Training the model, however, requires ground-truth labels that are not directly observable in a decentralised multi-agent setting.

Hindsight Target Generation. To optimise the Spatial OM, the network requires spatial ground-truth labels. Because the opponent’s internal decision-making is not always accessible to the learner during an episode, we generate labels post-hoc with Hindsight Target Generation (strongly related to Hindsight Experience Replay [3]).

At the conclusion of an episode, the trajectory is analysed backwards to determine the opponent’s achieved subgoals. If the opponent successfully reaches a food at timestep $t + k$, that coordinate (x, y) becomes the ground-truth subgoal for the preceding sequence; if the opponent fails to reach a definitive objective before the episode ends, their final occupied position becomes the fallback target. To account for spatial uncertainty and avoid overly sparse gradients, we transform this single coordinate into a continuous target intent heatmap $G_{target,t} \in [0, 1]^{H \times W}$ by applying a 2D Gaussian smoothing kernel with variance σ^2 centred at the target coordinate:

$$G_{target,t}^{(i,j)} = \exp\left(-\frac{(i-x)^2 + (j-y)^2}{2\sigma^2}\right) \quad (3)$$

Algorithm 1 Hindsight Subgoal Labelling with Gaussian Smoothing

Require:

Logged episode $E = [(o_t, r_t^{\text{opp}}, s_t^g, s_{t+1}^g)]_{t=1}^{T_{ep}}$: the learner’s observations o_t , the opponent’s rewards r_t^{opp} , and the pre-step and post-step privileged global states s_t^g and s_{t+1}^g (used for labelling only)
 Grid dimensions $H \times W$, Smoothing variance σ^2

```

1:  $g_{true} \leftarrow \text{None}$   $\triangleright$  Current true goal coordinate  $(x, y)$ 
2:  $G_{next} \leftarrow \mathbf{0}_{H \times W}$   $\triangleright$  Target map for the next state
    $\triangleright$  Fallback goal for unsuccessful or truncated episodes
3: if  $E$  is not empty then
4:    $(o_{T_{ep}}, r_{T_{ep}}^{\text{opp}}, s_{T_{ep}}^g, s_{T_{ep}+1}^g) \leftarrow E[T_{ep}]$ 
5:   if  $r_T^{\text{opp}} = 0$  then
6:      $g_{true} \leftarrow \text{ExtractOpponentPosition}(s_T^g)$ 
    $\triangleright$  Backwards Relabelling Trajectory
7: for  $t = T_{ep}$  downto 1 do
8:    $(o_t, r_t^{\text{opp}}, s_t^g, s_{t+1}^g) \leftarrow E[t]$ 
9:   if  $r_t^{\text{opp}} > 0$  then  $\triangleright$  Opponent achieved an objective (e.g., reached food)
10:     $g_{true} \leftarrow \text{ExtractOpponentPosition}(s_{t+1}^g)$ 
11:    $G_{target,t} \leftarrow \mathbf{0}_{H \times W}$ 
12:   if  $g_{true} \neq \text{None}$  then
13:      $G_{target,t}[g_{true}] \leftarrow 1.0$   $\triangleright$  Assign one-hot target
14:      $G_{target,t} \leftarrow \text{GaussianFilter}(G_{target,t}, \sigma^2)$   $\triangleright$  Generate dense soft-target map
15:   Label  $G_{target,t}$  to transition  $t$  as true_goal_map
16:   Label  $G_{next}$  to transition  $t$  as true_goal_map_next
17:    $G_{next} \leftarrow G_{target,t}$   $\triangleright$  Carry target map backward
18: return Labelled episode  $E$ 
  
```

This smoothing provides a dense learning signal, penalising “near-misses” less severely than spatially distant errors. The systematic process for extracting these labels from raw trajectories is detailed in Algorithm 1.

Training Objective. Because $G_{target,t}$ is a continuous Gaussian density, we normalise it to a valid probability distribution (summing to 1 across the grid) and optimise the network against it with the KL Divergence objective:

$$\begin{aligned} \mathcal{L}_{\text{OM}}(\phi) &= \mathcal{D}_{\text{KL}}(G_{target,t} \parallel \hat{G}_t) \\ &= \sum_{i=1}^H \sum_{j=1}^W G_{target,t}^{(i,j)} (\log G_{target,t}^{(i,j)} - \log \hat{G}_t^{(i,j)}) \end{aligned} \quad (4)$$

This encourages the Transformer to capture the regional spatial intent rather than memorising discrete coordinate points.

Feature fusion. The Q-function is conditioned on the inferred intent: $Q(o_t, a, \hat{G}_t)$.

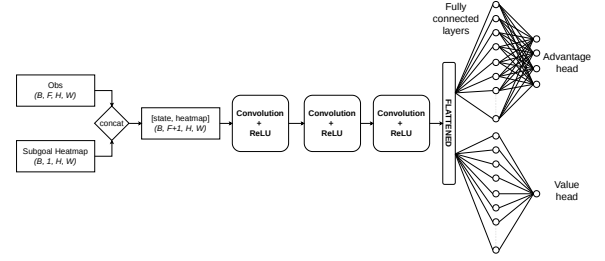


Figure 4: OM Dueling Q-Network. The observation and belief channels are concatenated with the inferred heatmap and processed by the CNN; Advantage and Value heads compute the final Q-values by Equation 6.

A low-dimensional latent summary of the opponent’s intent is easily drowned out by the far larger flattened spatial embedding of the Q-network’s CNN, so the intent signal is kept in physical coordinates: in an ESS game the opponent’s intent is tied to the grid’s physical layout. The Spatial Head therefore projects intent back into a heatmap, and the resulting $\hat{G}_t$ (Eq. 1) is concatenated as an additional channel of the network input (early spatial fusion) as depicted in Figure 4. The Q-network’s CNN thus sees the opponent’s likely destination as a heatmap overlaid on walls, food, and the agent’s own position, and learns their geometric relationships directly.

Let $o_t \in \mathbb{R}^{F \times H \times W}$ be the learner’s observation with F feature channels, and $b_t \in \mathbb{R}^{M \times H \times W}$ with M belief feature channels. This belief map is shared by all compared architectures. The inferred heatmap is concatenated with both along the feature dimension:

$$x_t = [o_t; b_t; \hat{G}_t] \in \mathbb{R}^{(F+M+1) \times H \times W} \quad (5)$$

The embedding $\mathbf{h}_{cnn} = f_{\text{CNN}}(x_t)$ is processed by the dueling heads [22], which separate state value ($V(\cdot)$) from per-action advantages ($A(\cdot, \cdot)$). The final Q-value is computed as:

$$Q(o_t, a, \hat{G}_t) = V(\mathbf{h}_{cnn}) + A(\mathbf{h}_{cnn}, a) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(\mathbf{h}_{cnn}, a') \quad (6)$$

The Q-network is optimised with the double-DQN update [21]: the online network selects the greedy next action and the Polyak-averaged target network Q_{θ^-} evaluates it, with target

$$y_t = r_t + \gamma Q_{\theta^-}(o_{t+1}, \arg\max_{a'} Q_{\theta}(o_{t+1}, a', \hat{G}_{t+1}), \hat{G}_{t+1}) \quad (7)$$

and loss $\text{Huber}(y_t - Q_{\theta}(o_t, a_t, \hat{G}_t))$ [11] (Algorithm 2).

Pre-training the opponent model. If the Spatial OM and the Q-network are trained concurrently from random initialisation, the early heatmaps are indistinguishable from uniform noise; the Q-network’s CNN quickly learns to assign near-zero weights to the heatmap channel, and once ignored the channel rarely recovers. We therefore pre-train p_{ϕ} offline before joint RL training. A supervised dataset of trajectories is collected by simulating games between heuristic agents (various pairings of the SIMPLE and GREEDY profiles, Section 5) and labelled via hindsight labelling (Algorithm 1). This grants the Spatial OM a foundational understanding of the grid’s topology, agent movement, and the environment’s rules, so that

Algorithm 2 DDQN with the Spatial OM (OM-DDQN) using HER

Initialise:

- 1: Replay Buffer $\mathcal{D}_{RL}$
- 2: Agent Network Q_θ , Target Network Q_{θ^-}
- 3: Spatial Opponent Model p_ϕ

Pre-train:

- 4: Train p_ϕ on offline heuristic dataset labelled via HER.
Collect data:

Collect data:

- ```

5: for episode = 1 to N_{ep} do
6: Initialise environment, observe o_0 , set history $\tau_0 = \emptyset$, and
 local episode buffer

```

**for** step  $t = 0$  to  $T_{ep}$  **do**

**Inference:**

- |    |                                                            |
|----|------------------------------------------------------------|
| 9: | Predict subgoal heatmap: $\hat{G}_t = p_\phi(o_t, \tau_t)$ |
|----|------------------------------------------------------------|

**Action:**

- |     |                                                                               |
|-----|-------------------------------------------------------------------------------|
| 11: | Select $a_t$ using Boltzmann exploration on $Q_\theta(o_t, \cdot, \hat{G}_t)$ |
|-----|-------------------------------------------------------------------------------|

**Step:**

- |     |                                                                     |
|-----|---------------------------------------------------------------------|
| 13: | Execute $a_t$ , observe reward $r_t$ and next observation $o_{t+1}$ |
|-----|---------------------------------------------------------------------|

Update history:  $\tau_{t+1} \leftarrow \text{append}(\tau_t, o_t) \triangleright \text{slide win-}$

- 15: Store step data  $(o_t, a_t, r_t, o_{t+1}, \tau_t, \hat{G}_t)$  in local episode buffer

### Insight Labelling:

- 17: Label the logged episode as described in Algorithm 1  
18: Transfer local buffer to global Replay Buffer  $\mathcal{D}_{RL}$

### Training Phase:

- 20: Sample batch of transitions and sequences from  $\mathcal{D}_{RL}$   
 ▶ Train OM Dueling Q-Network

generate next step heatmaps  $\hat{G}_{t+1}$  using the Spatial OM  
reference  $p_\phi(o_{t+1}, \tau_{t+1})$

- 22: Select next actions  $a'_{max} = \operatorname{argmax}_{a'} Q_{\theta}(o_{t+1}, a', \hat{G}_{t+1})$

Calculate target  $y = r_t + \gamma Q_{\theta^-}(o_{t+1}, a'_{max}, \hat{G}_{t+1})$

- 24: Compute Loss  $\mathcal{L}_{DDQN} = \text{Huber}(y - Q_{\theta}(o_t, a_t, \hat{G}_t))$

date  $\theta$  via gradient descent

- 26: Update target network (Polyak):
- $\theta^- \leftarrow \beta \theta + (1 - \beta) \theta^-$

▷ *Train the Spatial OM*

- 27: Compute  $\mathcal{L}_{OM}(\phi)$  using KL Divergence against normalised  
soft targets  $G_{target,t}$

date  $\phi$  via gradient descent

when joint OM-DDQN training begins (Algorithm 2) the Q-network receives structurally meaningful intent predictions from the first episode and integrates the heatmap channel during value estimation from the outset.

## 5 Experimental Setup

*Environment.* Foraging under partial observability (Section 3) played on two symmetric  $11 \times 11$  grids (Figure 1): MAP3 (dense central food, 5 goals) and MAP4 (sectioned walls, dispersed food, 7 goals), with fixed spawn and goal positions and vision radius  $v \in \{2, 5\}$  (fog thickness). The symmetry and game rules ensure that each

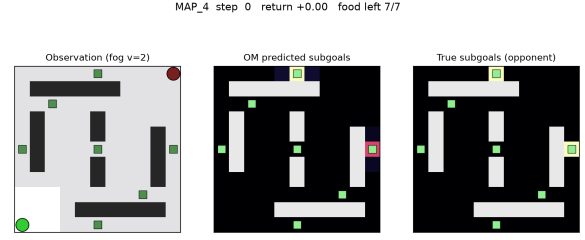


Figure 5: Example of Spatial OM predictions on MAP4 (vision radius 2). Left: the learner’s observation under fog; the learner (green dot) sees only its surroundings, while the opponent’s true position (dark red when obscured by fog) and the food (green squares) are shown for reference. Middle: the OM’s predicted subgoal distribution fed to the Q-network. Right: the opponent’s true subgoal distribution, that is not accessible during match. The OM concentrates mass on the opponent’s actual target even when the opponent is outside the learner’s vision radius.

**Table 1: Shared training hyperparameters and network sizes (both learning agents).**

|                                               |                                                                                                               |
|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| Learning rate                                 | Adam; $3 \cdot 10^{-4}$ (Q), $1 \cdot 10^{-4}$ (OM)                                                           |
| Discount $\gamma$ / soft update $\tau_{soft}$ | 0.985 / 0.001                                                                                                 |
| Batch size / train frequency                  | 128 / every 4 steps                                                                                           |
| Replay capacity                               | 450,000                                                                                                       |
| Episodes                                      | 30,000 (60 epochs $\times$ 500 episodes), max 50 steps                                                        |
| Boltzmann $\tau$                              | $1.5 \rightarrow 0.05$ over 650k steps                                                                        |
| Label smoothing $\sigma^2$                    | $2.0 \rightarrow 0.5$ over 150k steps                                                                         |
| OM pre-training                               | 20 epochs heuristic self-play                                                                                 |
| Q-net                                         | $3 \times \text{Conv}3 \times 3$ ( $C_{in} \rightarrow 32 \rightarrow 64 \rightarrow 64$ ), dueling heads 256 |
| OM transformer                                | $d_{model}=128$ , 2 layers, 4 heads, $d_{ff}=256$ , dropout 0.05, history length 50                           |

agent’s game value is half of the total food. Greedy vs greedy play shows this empirically (MAP3: 2.50/2.50; MAP4: 3.42/3.42).

*Opponents.* All opponents share the learner’s action space and movement rules and act on their own partial observations (same vision radius). SIMPLE commits to a random believed food and re-plans when it is consumed or seen as collected. GREEDY targets its nearest believed food and switches away whenever it believes the learner is closer. ORACLE-STALKER (also denoted as “tracker”) sees the true state regardless of fog and races to the food nearest the learner that it can still reach first, loitering there to deny collection.

*Training.* Dueling DQN,  $n$ -step returns  $n \in \{1, 3\}$  [20], belief-map prior  $p \in \{0, 1\}$  (belief map off/on), 60 epochs, 3 seeds per configuration, against the GREEDY opponent. Training used a single Tesla V100-SXM2-32GB GPU per run, taking about 7 hours plus per-epoch evaluation (CLASSIC and OM combined). Zero-shot evaluations ran on an Apple M3 Pro (12-core CPU, 18 GB RAM). Table 1 lists the shared hyperparameters.

**Table 2: Paired zero-shot advantage  $\Delta = \text{OM} - \text{CLASSIC}$  (mean  $\pm$  std over seeds) by opponent, fog, and  $n$ -step horizon on MAP4 (prior off).  $n=3$ : well-trained cells;  $n=1$ : cells where CLASSIC training is unstable. Seen forager means that the learner has trained against the opponent; unseen forager means that the opponent is not in the learner’s training set. Dense-map (MAP3) deltas are reported in the text and Table 3.**

| Opponent                 | Fog   | $n=3$            | $n=1$                    |
|--------------------------|-------|------------------|--------------------------|
| GREEDY (seen, forager)   | $v=2$ | $+0.64 \pm 0.20$ | $+0.71 \pm 0.08^\dagger$ |
| GREEDY (seen, forager)   | $v=5$ | $+0.41 \pm 0.07$ | $+1.50 \pm 0.26$         |
| SIMPLE (unseen, forager) | $v=2$ | $+0.31 \pm 0.31$ | $+0.66 \pm 0.05^\dagger$ |
| SIMPLE (unseen, forager) | $v=5$ | $+0.52 \pm 0.10$ | $+1.16 \pm 0.05$         |
| ORACLE-STALKER (tracker) | $v=2$ | $+0.12 \pm 0.39$ | $+0.52 \pm 0.13^\dagger$ |
| ORACLE-STALKER (tracker) | $v=5$ | $+0.02 \pm 0.09$ | $+1.13 \pm 0.27$         |

<sup>†</sup>Excluding one collapsed CLASSIC seed (see Limitations); the collapse is reported as a finding.

*Zero-shot evaluation.* Checkpoints are evaluated against GREEDY (the training opponent), SIMPLE, and ORACLE-STALKER, with seat-swap probability 0.5 and 1000 episodes per seed. We report differences paired by training seed ( $\Delta = \text{OM} - \text{CLASSIC}$ ; the same seed index for both agents), split by  $n$ -step configuration (pooled only where stated, as in the belief-prior comparison), mean  $\pm$  std over seeds.

## 6 Results

*Training stability.* Plain DQN is unstable at  $n=1$  (Figure 6). On MAP4 one CLASSIC seed collapses outright in  $v=2$  and all three  $v=5$  seeds finish at 37–46% of game value (1.3–1.6 of 3.5); on MAP3 at thick fog all  $n=1$  seeds finish at 47–68% of game value (1.17–1.71 of 2.5). OM never collapses and never finishes below 80% of game value on MAP3 (floor 2.03 of 2.5). Across all zero-shot batches on both maps, seed-level standard deviation is 1.4–3.6 $\times$  smaller for OM (MAP4: 0.16–0.31 vs. 0.50–0.79).

*In-distribution performance.* Against the training opponent (Figure 7), OM recovers 95% (MAP4,  $v=2$ ) to 99.9% ( $v=5$ ) of the game value in well-trained ( $n=3$ ) cells; CLASSIC reaches 77–88%. On MAP3 both agents reach 96–101% at  $n=3$ , since the game is saturated. Under thin fog both even exceed the game value on average (104% for OM, 101% for CLASSIC; individual seeds up to 107%): a best response to the suboptimal training heuristic. Thick fog separates them: in the hard cell ( $v=2$ ,  $n=1$ ) CLASSIC degenerates to 52–76% per seed (64% on average), because exploiting an opponent you rarely see requires predicting it.

*Opponent spectrum.* The advantage transfers across forager types, seen or unseen, and in well-trained scenarios disappears against the tracker (Table 2). SIMPLE is unseen by the learner, but its profile appears in the OM’s offline pre-training corpus (Section 4), so this tests the learned policy’s transfer, not the OM’s. This fits an information argument. A forager’s intent is a fact about the environment: its trajectory reveals it, and the learner’s fogged observations do not contain it. A tracker’s intent is the learner itself, so predicting it is redundant with self-knowledge. Tracking is also costly for the tracker: total extraction drops from  $\sim 7$  goals (against foragers) to

**Table 3: Blindfold ablation: heatmap zeroed at deployment. Mean OM return over seeds;  $\Delta = \text{OM} - \text{CLASSIC}$  paired by training seed, zero-shot. MAP4: prior off,  $n$ -split; MAP3:  $n=1$ , prior on/off pooled, 6 seeds; vs. the opponent shown.**

| Map / opponent       | Cell          | OM   | OM blind | $\Delta$           | $\Delta$ blind     |
|----------------------|---------------|------|----------|--------------------|--------------------|
| MAP4, GREEDY         | $v=2$ , $n=3$ | 3.33 | 2.23     | +0.64              | −0.47              |
| MAP4, GREEDY         | $v=2$ , $n=1$ | 2.86 | 2.43     | +0.71 <sup>†</sup> | +0.28 <sup>†</sup> |
| MAP4, SIMPLE         | $v=2$ , $n=3$ | 2.77 | 2.10     | +0.31              | −0.37              |
| MAP4, SIMPLE         | $v=2$ , $n=1$ | 2.55 | 2.31     | +0.66 <sup>†</sup> | +0.42 <sup>†</sup> |
| MAP4, ORACLE-STALKER | $v=2$ , $n=3$ | 2.09 | 1.71     | +0.12              | −0.26              |
| MAP4, ORACLE-STALKER | $v=2$ , $n=1$ | 2.10 | 2.10     | +0.52 <sup>†</sup> | +0.52 <sup>†</sup> |
| MAP4, GREEDY         | $v=5$ , $n=3$ | 3.50 | 3.02     | +0.41              | −0.07              |
| MAP4, GREEDY         | $v=5$ , $n=1$ | 3.17 | 2.50     | +1.50              | +0.84              |
| MAP4, SIMPLE         | $v=5$ , $n=3$ | 2.81 | 2.57     | +0.52              | +0.28              |
| MAP4, SIMPLE         | $v=5$ , $n=1$ | 2.54 | 2.31     | +1.16              | +0.92              |
| MAP4, ORACLE-STALKER | $v=5$ , $n=3$ | 2.29 | 2.17     | +0.02              | −0.10              |
| MAP4, ORACLE-STALKER | $v=5$ , $n=1$ | 2.50 | 2.24     | +1.13              | +0.87              |
| MAP3, GREEDY         | $v=2$         | 2.18 | 2.13     | +0.57              | +0.53              |
| MAP3, SIMPLE         | $v=2$         | 1.67 | 1.64     | +0.52              | +0.48              |
| MAP3, ORACLE-STALKER | $v=2$         | 1.71 | 1.68     | +0.48              | +0.44              |
| MAP3, GREEDY         | $v=5$         | 2.60 | 2.44     | +0.07              | −0.09              |
| MAP3, SIMPLE         | $v=5$         | 2.44 | 2.29     | +0.12              | −0.03              |
| MAP3, ORACLE-STALKER | $v=5$         | 2.04 | 2.03     | +0.23              | +0.21              |

<sup>†</sup>Row excludes one collapsed CLASSIC seed (see Limitations); the collapse is reported as a finding.

$\sim 3.5$ , as the tracker spends roughly 2.2 goals of its own foraging to remove about 1 from the learner.

On the dense map, pooling prior on and off, the paired advantage concentrates in one hard cell ( $v=2$ ,  $n=1$ ): +0.57 vs. GREEDY, +0.52 vs. SIMPLE, +0.48 vs. the tracker. Wherever CLASSIC trains, the map saturates and the advantage falls to at most +0.13 against foragers and +0.23 against the tracker. The tracker at the hard cell is our mechanism probe: the advantage survives (+0.48), and blinding the heatmap leaves it nearly intact (+0.44, Table 3). Against a tracker, the gap is training-time stabilisation, not inference.

*Ablating the inference channel.* Table 3 completes the grid: all three opponents, both fog levels, and both returns configurations, with the paired advantage computed within the same runs. The picture splits along training difficulty. In the well-trained cells ( $n=3$ ), blinding erases the entire advantage in five of six cells: against the training opponent at thick fog +0.64  $\rightarrow$  −0.47, against the unseen forager +0.31  $\rightarrow$  −0.37. The one exception keeps just over half (+0.52  $\rightarrow$  +0.28, unseen forager, thin fog). Where training succeeds, the advantage is inference. In the unstable cells ( $n=1$ ), an edge survives blinding against every opponent (+0.28 to +0.92 over CLASSIC, excluding the collapsed seed at thick fog): stabilisation that does not require the heatmap at deployment.

Reliance on the channel is strongest in distribution and in fog. The largest drop is against the training opponent at thick fog (3.33  $\rightarrow$  2.23, leaving OM half a goal below CLASSIC), with a worst per-seed drop of 1.81. Across the well-trained cells the drop shrinks as fog thins (1.10  $\rightarrow$  0.48 goals against the training opponent): the policy consults the heatmap in proportion to how much its own observations leave unknown. The unstable cells show no consistent

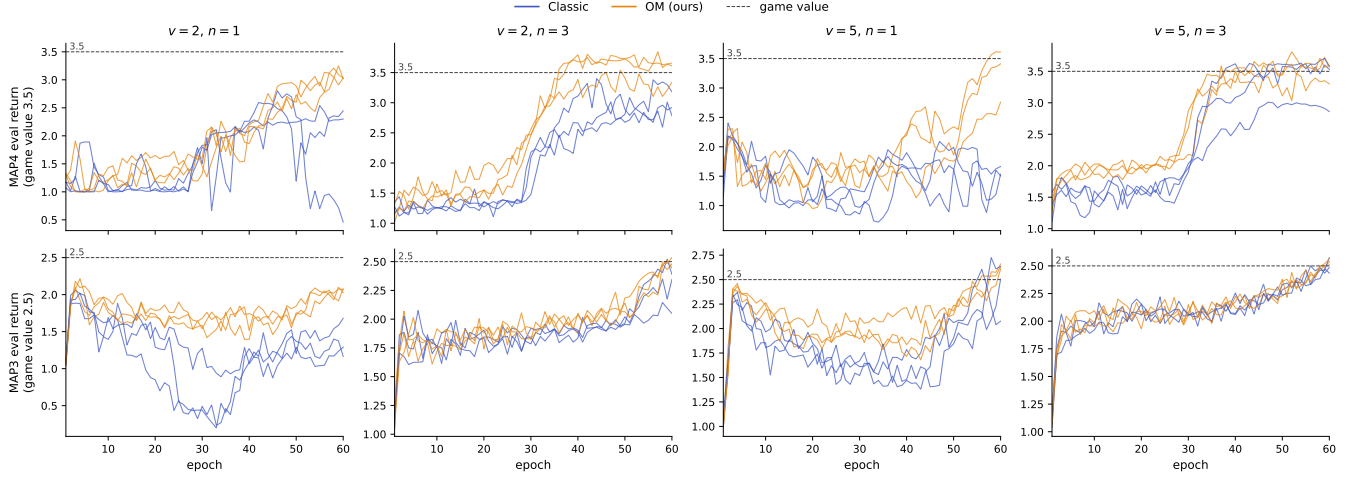


**Figure 6: Per-epoch evaluation return against the training opponent, per seed: CLASSIC (blue) vs. OM (orange), belief prior off. Top row MAP4, bottom row MAP3. CLASSIC degenerates at  $n=1$ : on MAP4 one  $v=2$  seed collapses outright, all three  $v=5$  seeds finish below 1.7 of 3.5, and on MAP3 all seeds in the  $v=2$  cell finish below 2.0 of 2.5 (thin fog recovers). OM never collapses on either map.**

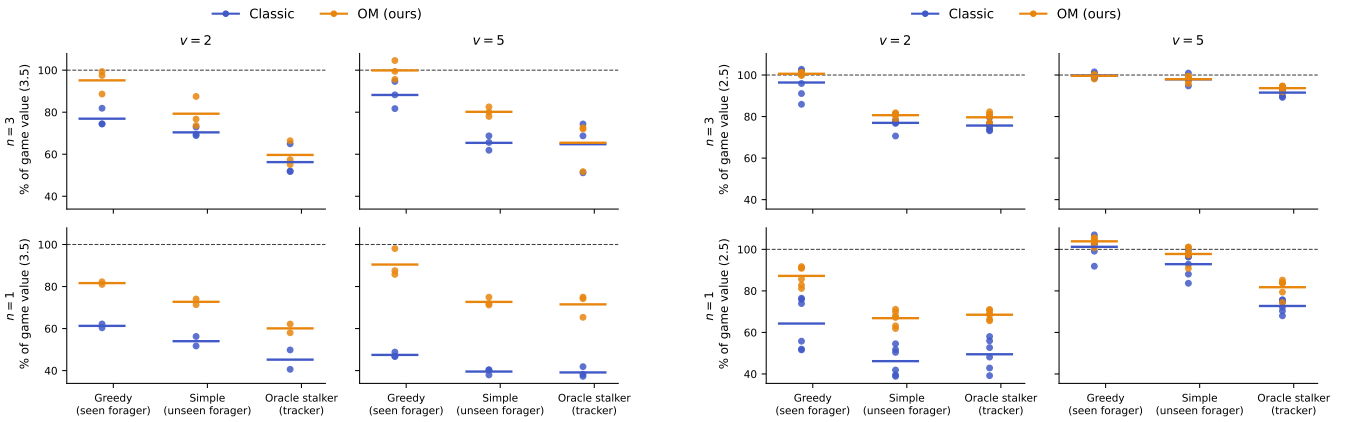


**Figure 7: Zero-shot return as a share of the symmetric game value, by opponent and cell (dots: seeds; bar: mean; dashed: game value). Left: MAP4 (prior off; collapsed CLASSIC seed excluded). Right: MAP3 (prior on/off pooled, 6 seeds per cell). Both agents exceed 100% under thin fog (best-response exploitation of the suboptimal heuristic) and both saturate on the dense map wherever training succeeds.**

fog scaling, so their stabilisation is fog-independent even where their inference is not. Blinding never collapses an OM seed (per-seed returns stay above 1.5), so the drop mixes information loss with input corruption (see Limitations). *Where training is hard, the opponent model contributes stabilisation; where training is easy, it contributes inference.*

The tracker rows tell the same story on the sparse map. In the unstable cells the gap survives blinding almost fully ( $+0.52 \rightarrow +0.52$  at thick fog,  $+0.87$  of  $+1.13$  at thin fog): stabilisation. In the well-trained cells the advantage is small, but its blindfold response still tracks fog: near zero either way at thin fog ( $+0.02 \rightarrow -0.10$ ), yet at thick fog the channel is still consulted ( $+0.12 \rightarrow -0.26$ , a 0.37-goal drop for OM itself); the heatmap there carries not the target (the

learner knows its own position) but the tracker’s approach under fog. An opponent whose intent is about the learner rather than the environment yields stabilisation, not inference.

On the dense map, blinding is nearly neutral at thick fog (at most 0.05 goals in every  $v=2$  cell), so the whole  $+0.5$  advantage there is stabilisation. The largest drops come at thin fog ( $-0.16$  against the training opponent at  $v=5$ ) and flip the paired advantage slightly negative ( $-0.09$ ). With the game saturated there is little for inference to add. The fog scaling of inference therefore needs a cell that is not saturated, which the dense map lacks: itself a density effect.

*Resource density.* Table 4 completes the picture. Wherever CLASSIC trains successfully ( $\geq 90\%$  of value), OM also sits at the ceiling

**Table 4: Dense MAP3 (game value 2.5 by symmetry): epoch-60 eval return vs. the training opponent as % of game value, mean over 3 seeds per cell.  $\Delta$ : paired OM–CLASSIC in goals.**

| Fog   | $n$ | Prior | CLASSIC | OM    | $\Delta$ |
|-------|-----|-------|---------|-------|----------|
| $v=2$ | 1   | off   | 55.1    | 82.3  | +0.68    |
| $v=2$ | 1   | on    | 65.8    | 89.4  | +0.59    |
| $v=2$ | 3   | off   | 90.5    | 99.5  | +0.23    |
| $v=2$ | 3   | on    | 98.4    | 99.8  | +0.04    |
| $v=5$ | 1   | off   | 96.2    | 105.3 | +0.23    |
| $v=5$ | 1   | on    | 104.0   | 106.9 | +0.07    |
| $v=5$ | 3   | off   | 99.9    | 101.5 | +0.04    |
| $v=5$ | 3   | on    | 97.8    | 99.6  | +0.05    |

and the advantage is at most 0.23 goals. On a dense map a lost contest costs little because another goal is nearby, so predicting the opponent’s target is worth little: the advantage vanishes by saturation, not by failure. The stability effect replicates and depends on fog. At thick fog all six CLASSIC  $n=1$  seeds fall below 2.0 of 2.5 (minimum 1.17) while OM never drops below 2.03; at thin fog even CLASSIC  $n=1$  recovers (minimum 2.08). Across both maps, the advantage shrinks with the difficulty reducers that stabilise CLASSIC training (denser map, longer  $n$ -step horizon), from +0.5–1.5 goals in the sparse map’s unstable cells to +0.04 on the dense map under thin fog. Fog is the exception on the sparse map: at  $n=1$  the gap widens as fog thins (+0.71  $\rightarrow$  +1.50 vs. the training opponent), because OM improves with visibility (2.84  $\rightarrow$  3.17) while CLASSIC does not (2.1  $\rightarrow$  1.7 goals excluding the collapsed seed).

*Belief-map prior.* On MAP3 the prior is neutral for OM: the largest cell-level return difference between  $p=1$  and  $p=0$  is 0.18 goals, and its effect on inference quality is small and inconsistent in sign (−0.28 to +0.26). It partially rescues CLASSIC where TD(0) fails: at the hard cell ( $v=2$ ,  $n=1$ ; zero-shot vs. GREEDY), CLASSIC rises from 53% to 75% of game value with the prior, while OM remains highest (91%). Table 4 shows the same direction with smaller gaps (55.1%  $\rightarrow$  65.8%, OM 82.3–89.4%).

The MAP4 replication turns this into a boundary condition. At thin fog the prior lifts the baseline (CLASSIC vs. the training opponent: 68%  $\rightarrow$  84% of game value) while OM gives up 0.3–0.6 goals. The paired advantage, pooled over the  $n$ -split for this comparison, thus shrinks from +0.90/+0.96 to +0.60/+0.04 goals (thick/thin fog) and nearly vanishes against the tracker (from +0.51/+0.58 to +0.01/+0.24). Part of the prior-free edge was an information-starved baseline: a positional belief substitutes for subgoal inference at thin fog, but not at thick fog, where inference keeps a +0.6 goal edge over the baseline with the prior. The prior is a safe default for the baseline; the opponent model buys the thick-fog remainder.

## 7 Limitations

Three seeds per configuration. We report per-seed strip plots and never pool across  $n$ -step configurations (except where stated, as in the belief-prior comparison), and several cells have seed-level standard deviations comparable to their means and should be read as non-significant. One CLASSIC seed on MAP4 collapsed during training and is excluded from paired statistics there, with the training

curve cited as justification. Collapse frequency is itself reported. All results are best responses to a heuristic. In preliminary experiments, NFSP-style self-play [9] did not converge in the FOSG domain, so we make no MARL equilibrium claims. Published opponent-modeling methods assume different observation spaces (full state, action histories, or population priors). None infers spatial opponent subgoals from partial observations in this setting, so we compare against CLASSIC, whose Q-network matches the OM Q-network on observation and belief channels and on capacity, differing only in the heatmap input, although that heatmap summarises an observation history CLASSIC does not see (see Conclusion). We fix the learner to value-based DQN methods because the claim concerns TD(0) stability under partial observability. Policy-gradient learners would change the failure mode rather than test it. Zeroing the heatmap is an input distribution shift. The drop measures information loss plus input corruption, not information loss alone. And lastly, this work focuses on grid foraging only, scale is future work.

## 8 Conclusion and Future Work

Opponent-goal inference under partial observability is not a uniform advantage: its value is set by visibility (fog thickness), by whether the opponent’s intent is about the environment or about the learner, and by resource density. Its most robust benefit is making DQN training stable where plain TD(0) fails. We attribute the stabilisation to the module as a whole. Whether it stems from subgoal inference or from history-derived structured input is not isolated here. The blindfold ablation offers partial evidence: zeroing the heatmap, the only input beyond the channels shared with CLASSIC, costs the most deployment performance in the cells where the information argument says inference matters most (in distribution and at thick fog on the sparse map), consistent with the inference carrying information rather than acting as generic structured input. Future work scales these mechanisms to roadmap graphs of real-world maps, where foraging under fog becomes a multi-agent travelling-salesman problem under partial observability, in both competitive and cooperative settings, and replaces self-play with policy-space or population methods for equilibrium approximation.

## 9 Acknowledgements

I would like to thank my supervisor, Ing. Rostislav Horčík, Ph.D., for his invaluable guidance and support throughout this research. The authors acknowledge the support of the OP VVV “Research Center for Informatics”. The access to the computational infrastructure of the OP VVV “Research Center for Informatics” is also gratefully acknowledged.

## References

- [1] Maruan Al-Shedivat, Trapit Bansal, Yura Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. 2018. Continuous Adaptation via Meta-Learning in Non-stationary and Competitive Environments. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. <https://openreview.net/forum?id=Sk2u1g-0>
- [2] Stefano V. Albrecht and Peter Stone. 2018. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artif. Intell.* 258 (2018), 66–95. doi:10.1016/J.ARTINT.2018.01.002
- [3] Marcin Andrychowicz, Dwight Crow, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. 2017. Hindsight Experience Replay. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5048–5058. <https://proceedings.neurips.cc/paper/2017/hash/453fadb8a1a3af50a9df4df899537b5-Abstract.html>
- [4] Sander C. J. Bakkes, Pieter H. M. Spronck, and H. Jaap van den Herik. 2009. Opponent modelling for case-based adaptive game AI. *Entertain. Comput.* 1, 1 (2009), 27–37. doi:10.1016/J.ENTCOM.2009.09.001
- [5] Eugene Charniak and Robert P. Goldman. 1993. A Bayesian Model of Plan Recognition. *Artif. Intell.* 64, 1 (1993), 53–79. doi:10.1016/0004-3702(93)90060-O
- [6] Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. 2020. Shared Experience Actor-Critic for Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). <https://proceedings.neurips.cc/paper/2020/hash/7967cc8e3ab559e68cc944c44b1cf3e8-Abstract.html>
- [7] Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. 2004. Dynamic Programming for Partially Observable Stochastic Games. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, 2004, San Jose, California, USA*, Deborah L. McGuinness and George Ferguson (Eds.). AAAI Press / The MIT Press, 709–715. <http://www.aaai.org/Library/AAAI/2004/aaai04-112.php>
- [8] He He and Jordan L. Boyd-Graber. 2016. Opponent Modeling in Deep Reinforcement Learning. In *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016 (JMLR Workshop and Conference Proceedings, Vol. 48)*, Maria-Florina Balcan and Kilian Q. Weinberger (Eds.). JMLR.org, 1804–1813. <http://proceedings.mlr.press/v48/he16.html>
- [9] Johannes Heinrich and David Silver. 2016. Deep Reinforcement Learning from Self-Play in Imperfect-Information Games. *CoRR* abs/1603.01121 (2016). arXiv:1603.01121 <http://arxiv.org/abs/1603.01121>
- [10] Zhang-Wei Hong, Shih-Yang Su, Tzu-Yun Shann, Yi-Hsiang Chang, and Chun-Yi Lee. 2017. A Deep Policy Inference Q-Network for Multi-Agent Systems. *CoRR* abs/1712.07893 (2017). arXiv:1712.07893 <http://arxiv.org/abs/1712.07893>
- [11] Peter J. Huber. 1964. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics* 35, 1 (1964), 73 – 101. doi:10.1214/aoms/1177703732
- [12] Michael L. Littman. 1994. Markov Games as a Framework for Multi-Agent Reinforcement Learning. In *Machine Learning, Proceedings of the Eleventh International Conference, Rutgers University, New Brunswick, NJ, USA, July 10-13, 1994*, William W. Cohen and Haym Hirsh (Eds.). Morgan Kaufmann, 157–163. doi:10.1016/B978-1-55860-335-6.50027-1
- [13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharmashan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nat.* 518, 7540 (2015), 529–533. doi:10.1038/NATURE14236
- [14] Samer B. Nashed and Shlomo Zilberstein. 2022. A Survey of Opponent Modeling in Adversarial Domains. *J. Artif. Intell. Res.* 73 (2022), 277–327. doi:10.1613/JAIR.1.12889
- [15] Georgios Papoudakis, Filippos Christianos, and Stefano V. Albrecht. 2021. Agent Modelling under Partial Observability for Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc’Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 19210–19222. <https://proceedings.neurips.cc/paper/2021/hash/a03caec56cd82478bf197475b48c05f9-Abstract.html>
- [16] Marcel Petran. 2026. *Opponent Modeling in Zero-Sum Games*. Master’s thesis. Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic.
- [17] David Premack and Guy Woodruff. 1978. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences* 1, 4 (1978), 515–526. doi:10.1017/S0140525X00076512
- [18] Neil C. Rabinowitz, Frank Perbet, H. Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew M. Botvinick. 2018. Machine Theory of Mind. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 4215–4224. <http://proceedings.mlr.press/v80/rabinowitz18a.html>
- [19] Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. 2018. Modeling Others using Oneself in Multi-Agent Reinforcement Learning. *CoRR* abs/1802.09640 (2018). arXiv:1802.09640 <http://arxiv.org/abs/1802.09640>
- [20] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement learning - an introduction, 2nd Edition*. MIT Press. <http://www.incompleteideas.net/book/the-book-2nd.html>
- [21] Hado van Hasselt, Arthur Guez, and David Silver. 2016. Deep Reinforcement Learning with Double Q-Learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, Dale Schuurmans and Michael P. Wellman (Eds.). AAAI Press, 2094–2100. doi:10.1609/AAAILV30I1.10295
- [22] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. 2016. Dueling Network Architectures for Deep Reinforcement Learning. In *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016 (JMLR Workshop and Conference Proceedings, Vol. 48)*, Maria-Florina Balcan and Kilian Q. Weinberger (Eds.). JMLR.org, 1995–2003. <http://proceedings.mlr.press/v48/wangf16.html>
- [23] Xiaopeng Yu, Jiechuan Jiang, and Zongqing Lu. 2024. Opponent Modeling based on Subgoal Inference. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (Eds.). [http://papers.nips.cc/paper\\_files/paper/2024/hash/6fb9ea5197c0b8ece8a64220fb82cdf-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2024/hash/6fb9ea5197c0b8ece8a64220fb82cdf-Abstract-Conference.html)
- [24] Luisa M. Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. 2020. VariBAD: A Very Good Method for Bayes-Adaptive Deep RL via Meta-Learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. <https://openreview.net/forum?id=Hkl9JlBYr>