Appendix

A Concept Definitions

Range: The range concept indicates whether an agent is within range of another agent *and* facing it, where the range is within 0.8 map units and the angle is within $\frac{\pi}{5}$ radians. This value is a one-hot encoded value (within range or not within range) for each opposing agent, such that the total number of output nodes for this concept is $2n$ where n is the number of agents on the opposing team.

Strategy: The strategy concept indicates what team-level strategy the attacking team is following: *left*, *right*, or *random*. The strategy refers to the actions that an attacking team will take; a left strategy indicates that agents will follow a trajectory along the left side of the map leading to the objective, a right strategy indicates that agents will follow a trajectory along the right side of the map, and random indicates that agents will execute random actions. This concept is one-hot encoded and requires p output nodes, where p is the number of strategies – 3 in this work.

Target: The target concept indicates which agent on the opposing team an agent is currently *targeting*. The goal of this concept is to overcome the issue of oscillating targets, e.g., if an agent is equally close to multiple agents this can lead to oscillating behavior where the ego agent is unsure which other agent to pursue, and flips between them as the distance changes. During training, the targeted agent is initially selected to be the closest opposing agent and is only updated when that agent has been tagged. As with the *range* concept, this is a one-hot encoded value with $2n$ output nodes.

Orientation: The orientation concept is a continuous value representing the relative angle offset between an agent to each other agent on the opposing team. This value consists of n nodes where n is the number of agents on the opposing team.

Position: The position concept is a continuous value representing the relative Euclidean distance between an agent and each other agent on the opposing team. This value consists of n nodes where n is the number of agents on the opposing team.

B Training Details

Models are trained using a centralized-training-decentralized-execution approach, where a single policy is trained for all agents, and then executed individually for each agent during rollouts. True concept values are provided during training via an oracle function $V(\cdot)$ and used to compute the corresponding auxiliary loss, as well as the true concept values for intervention during intervened evaluation rollouts. Attacker strategies during training are sampled with equal probability from the set {*left*, *right*, *random*}, with sampled Left and Right strategies shown in Fig. 2. The *random* strategy is utilized to encourage the defenders to develop strategies in which they are free to pursue individual attackers, as opposed to remaining stationary near the objective. All policies were trained for 10M timesteps, after which the best policy checkpoint was taken – necessary since some models experienced forgetting and instability. Extensive hyperparameter optimization was performed, with the selected hyperparamters shown in Table 2.

Reward: We have simplified the reward function of FortAttack by removing penalties to encourage policy exploration, resulting in a reward of the form

$$R(s_a, u_a) = -R_{\text{Ori.}} - R_{\text{Miss}} - R_{\text{Tagged}} - R_{\text{Lose}} + R_{\text{Tag}} + R_{\text{Win}}, \tag{1}$$

where $R_{\text{Ori.}}$ is a penalty for not facing an opponent, R_{Miss} is a penalty for missing a tag, R_{Tagged} is a penalty for being tagged, R_{Lose} is a penalty for losing, R_{Tag} is a reward for tagging, and R_{Win} is a reward for winning. Several of these are shaping terms in order to improve sample efficiency, which we found to be necessary for efficient convergence rates in the absence of expert demonstrations.

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

	Setup	Model	*Range*	*Strategy*	*Target*	*Orientation*	*Position*
Simulation	2v2	Soft	0.03 ± 0.0032	0.04 ± 0.0060	0.24 ± 0.012	-	-
		Hard	0.04 ± 0.0040	0.07 ± 0.0066	0.20 ± 0.012	0.10 ± 0.0071	0.11 ± 0.014
		Base	-	-	-	-	-
	3v3	Soft	0.03 ± 0.0029	0.10 ± 0.0085	0.17 ± 0.0109	-	-
		Hard	0.03 ± 0.0031	0.13 ± 0.098	0.23 ± 0.012	0.11 ± 0.0071	0.14 ± 0.0014
		Base	-	-	-	-	-
	5v5	Soft	0.02 ± 0.0022	0.25 ± 0.012	0.52 ± 0.013	-	-
		Hard	0.03 ± 0.0021	0.14 ± 0.012	0.13 ± 0.0097	0.11 ± 0.0063	0.21 ± 0.0142
		Base	-	-	-	-	-
Real	2v2	Soft	0.03 ± 0.0037	0.53 ± 0.015	0.13 ± 0.010	-	-
		Hard	0.04 ± 0.0040	0.53 ± 0.015	0.92 ± 0.0082	3.48 ± 0.11	0.81 ± 0.039
		Base	-	-	-	-	-

Table 1: The concept errors (mean and standard error) for our proposed models (Soft and Hard) and a baseline without concepts (Base). The Hard model is trained over all concepts, the Soft model over a subset, and the Base model with none. *Range*, *Strategy*, and *Target* are discrete concepts and as such the error shown is the error in accuracy score, while *Orientation* and *Position* are continuous and indicate mean squared error. *Orientation* is in radians and *Position* is a unit-less value in $[-1, 1]$. Errors are computed over 100 episode rollouts for each model for simulated, and 20 rollouts for real-world.

C Detailed Results Analysis

Table 1 is an expanded table showing the concept errors for each model in each scenario type with the addition of the standard error, indicating fairly consistent predictions. We have additionally computed statistical significance over the win rates for each model in each scenario type (Table 1 in the main paper) using Fisher's exact test with $p < 0.05$, finding that in all simulated scenarios (2v2, 3v3, and 5v5), the Hard model wins significantly more than the Soft and Base models over 100 evaluation episodes. In the case of interventions, the only model which exhibits statistically significant improvement to the win rate after intervention is the 2v2 hard concept policy model for the real world, increasing from 25% to 95% – partially due to the small number of evaluation policy rollouts (20). We note that while the soft concept policy's intervened win rate may not be a statistically significant decrease, it is an interesting observation that interventions do not increase the win rate. We conjecture that this is because the residual layer encodes observation information which is affected by the distributional shift from simulation to the real-world, and despite being able to intervene over a subset of the concepts that are encoded, this is not enough to overcome the distribution gap. Furthermore, the intervened hard concept policy model yields a greater improvement to the win rate in the real world than in simulation. This is an interesting outcome, and we conjecture that this is due to the real-world dynamics making it easier for the defenders to win assuming correct concept predictions. For example, the attackers can move and turn faster in simulation as they are not subject to real-world physics, and this makes it harder for the defenders to tag them.

D Model Architecture

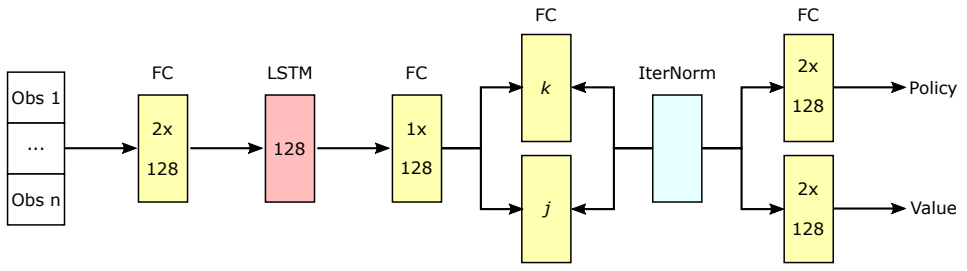

Figure 1: The common model architecture shared by all models, which vary only in j and k.

The common model architecture shared by all models is shown in Fig. 1, varying only in the size of the concept and residual layers (as described in the main paper). The observations of each agent in the opposing team are stacked and fed through a series of FC layers. This is then passed through a recurrent LSTM layer to capture temporal information, and then split into the concept and residual layers. Whitening is performed via iterative normalization over the concatenated concept and residual layers, which are then passed into the policy and value heads consisting of more fully connected layers. Each group of (2x 128) fully connected layers are followed by a ReLU activation, with the group-wise softmax following the concept layer after the IterNorm for discrete concepts only. The auxiliary loss $L^c(\theta)$ is computed over the concepts after the IterNorm layer.

E Hyperparameters

Learning Rate Schedule	1e-3 at t=0 to 1e-4 at t=10M
Entropy Schedule	0.1 at t=0 to 0.01 at t=10M
LSTM Max Sequence Length	50
Batch Size	10240
SGD Minibatch Size	1600 (Seq. Length $\times$ 32)
Concept Loss Coeff.	10
T	2
Optimizer	Adam
β_1, β_2	0.9, 0.999

Table 2: Hyperparameters used for each model type.

The MAPPO hyperparameters used for each model during training are given in Table 2. The learning rate and entropy values used a linear scheduler where the values were decreased as training progressed.

F Attacker Strategies

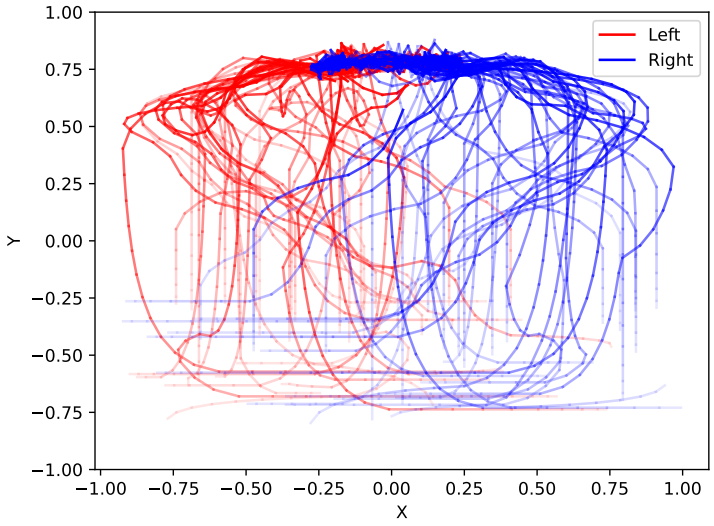

Figure 2: Sampled attacker strategies from the *left* and *right* strategies. Lighter points indicate agent positions earlier in the episode. The objective is at the top of the screen. Attackers randomly spawn in the lower half of the environment.

Figure 2 shows a set of sampled individual attacker strategies from the *left* and *right* attacker distributions (red and blue respectively). These trajectories illustrate the amount variance that is displayed by the attackers, despite sampling behavior from fixed strategy distributions.