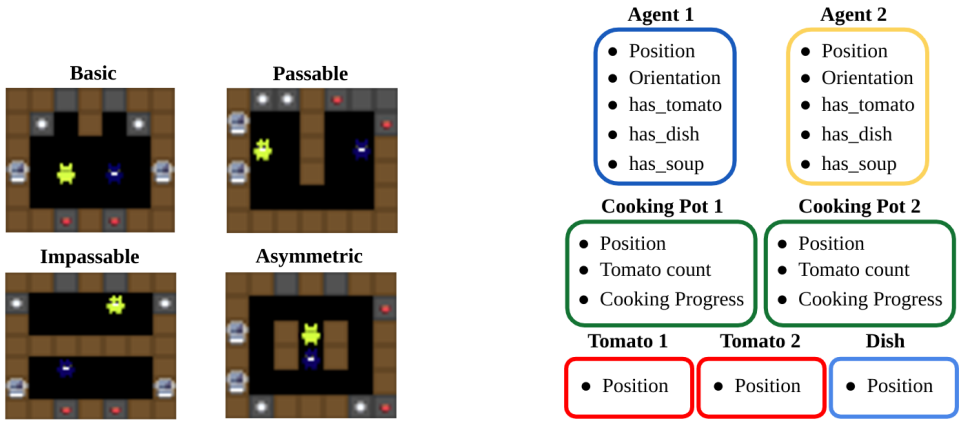

(a) Collaborative Cooking Environments. (b) Available Concepts.

Figure 7: Environment Overview.

A Additional Experimental Details

Based on the game Overcooked [52] and subsequent work that has developed Overcooked-like environments for studying multi-agent coordination [47, 48], Melting Pot's Collaborative Cooking is a game in which a group of agents inhabit a kitchen-like environment and must collaborate to find ingredients, complete recipes, and deliver finished dishes as quickly as possible. The Collaborative Cooking game for N agents is defined as follows:

- *Recipe*: (i) Bring a tomato to a cooking pot (3 times), (ii) Wait for soup to cook in the pot (20 time-steps), (iii) Bring a dish to the cooking pot, (iv) Pour soup from the pot into the dish, (v) Deliver soup to the delivery location. In practice, solving this task from scratch in its entirety is extremely difficult, as each of the aforementioned steps requires agents to execute a series of movement and interaction actions in sequence.
- *States*: Each Collaborative Cooking environment is a grid world. Grid cells can be filled by an agent or any of the following items: Floor, Counter, Cooking Pot, Dish, Tomato.
- *Observations*: Agents receive partial multi-modal observations consisting of their own position and orientation in the grid, as well as a partial RGB rendering of a 5-cell $\times$ 5-cell window centered at the agent.
- *Actions*: Agents can execute one of 8 actions: no-op, move $\{\texttt{up}, \texttt{down}, \texttt{left}, \texttt{right}\}$, turn $\{\texttt{left}, \texttt{right}\}$, and interact.
- *Reward*: By default, agents share a positive reward for completing the entire recipe outlined above. In practice, however, solving the cooking task with this sparse reward alone is infeasible (completing each of the recipe steps through random exploration is prohibitively challenging) and successful approaches in prior works either pair learning agents with helpful bot agents or introduce "densified" pseudorewards to augment the agents' learning signal [8]. We implement the latter, giving agents a small positive reward for completing steps (i) and (iii) of the recipe. More concretely, we define the following three-part reward:

$$r_t = \begin{cases} 20, & \text{if soup cooked and delivered.} \\ 1, & \text{if tomato placed in cooking pot.} \\ 1, & \text{if soup poured into dish.} \\ 0, & \text{otherwise.} \end{cases}$$

- *Concepts*: We assume that each environment supports the following concepts (and concept types): (i) agent position (scalar); (ii) agent orientation (scalar); (iii) whether or not an agent has a tomato, dish or soup (binary); (iv) cooking pot position (scalar) (v) the progress of the cooking pot (scalar); (vi) the number of tomatoes in the cooking pot (categorical); and (vii) the position of each tomato and dish (scalar).

A visualization of the four cooking environments that we used in our experiments are shown in Fig. 7(a) and a review of the supported concepts are shown in Fig. 7(b).

Table 1: Hyperparameter sweeps for training ConceptPPO and PPO. Swept values are shown in braces and highest-performing values are bolded.

Hyperparameters	
Name	Value
Training Steps	25e6
Batch Size	$\{64, 128, 256, \mathbf{512}, 1024\}$
Learning Rate	$\{1e{-}3, \mathbf{1e{-}4}, 1e{-}5\}$
Gradient Norm	$\{0.1, \mathbf{0.5}, 1.0, 5.0, 10.\}$
PPO Unroll Length	$\{4, 8, \mathbf{16}, 32\}$
PPO Clipping ϵ	$\{0.01, \mathbf{0.05}, 0.1, 0.2, 0.3\}$
PPO Entropy Cost	$\{0.001, \mathbf{0.01}, 0.05\}$
PPO Value Cost	$\{0.75, 0.9, \mathbf{1.0}\}$

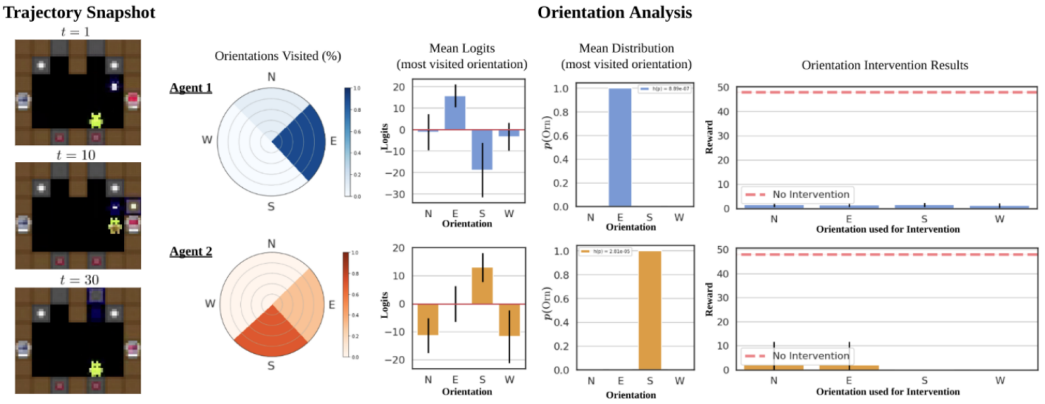

Figure 8: Overview of our method for constructing in-distribution concept masks for intervention. Using test-time trajectories, we compute an empirical distribution of the orientations experienced by each agent. Next, we compute the mean logits (and corresponding distribution) produced by each agent for for the most frequently visited orientation of its teammate. Finally, we report the results of intervening with this mean logits vector in place of each of the four cardinal orientations. The results of this intervention are consistent, regardless of whether the mask used was in- or out-of-distribution.

A.1 Training Details

We conducted a wide hyperparameter sweep to train both ConceptPPO and PPO, which is summarized in Table 1.

B Factors of Coordination Analysis (cont'd)

B.1 In vs. Out-of-Distribution Orientation Analysis

Here we further examine the impact of intervening on orientation. First, we compute an empirical distribution of orientations that each agent vists over 100 test-time trajectories (across five random seeds each). From those trajectories, we compute both the mean logits vector produced by each agent's concept bottleneck for the *most frequently visited orientation*, and the distribution represented by those logits. Crucially, the mean logits vector for the most frequently visited orientation can be used as an in-distribution value for concept interventions—it is an orientation estimate that each agent has likely seen before. Similarly, we can create an out-of-distribution concept intervention mask by permuting the mean logits vector such that the probability mass shifts a different cardinal orientation. Using these manufactured orientations, we perform the same intervention technique as before, iteratively replacing each agent's orientation concept with the mean logits vector in place of each cardinal direction, and measure performance of the multi-agent team.

To provide intuition for this technique, we ground it in the cooking task used for our experiments. Consider, for example, the tomato-picking agent in the trajectory snapshot of Fig. 8, who primarily faces south and east—the directions needed to pick tomatoes from the bottom counter and place them in the cooking pot on the right-hand side. The tomato-picking agent's teammate (the waiter agent) must learn to accuately model these orientations to satisfy its concept prediction objective, and so frequently passes low-entropy distributions for south and east to its policy network. Intervening on the waiter agent's orientation estimate with a low-entropy distribution for south and east, therefore, creates an in-distribution mask, whereas intervening with a low-entropy distribution for north or west creates an out-of-distribution mask.

The results of this intervention test are shown in Fig. 8, alongside the orientation distributions, mean logits, and the distribution represented by those logits. Interestingly, the previously observed degradation of performance as a result of intervening on teammate orientation is upheld, regardless of whether the mask value is in- or out-of-distribution. This provides further evidence that the agent's reliance on orientation is a legitimate artifact of their emergent strategy and not an adversarial or OOD example.

B.2 Intervention Without Orientation

To further investigate the surprising use of orientation as the primary signal driving the emergent behavior of our learning agents, we train a set of ConceptPPO policies in our basic environment without orientation as a concept (across each λ value as before). We then perform the same iterative intervention analysis over the concepts pertaining to each agent's teammate (as outlined in Section 5.1 (excluding orientation, of course). The results of this analysis are shown in Fig. 9. After removing each agent's orientation concept, we find that the agents latch onto a new concept–"has soup"–as the primary concept that drives their coordination.

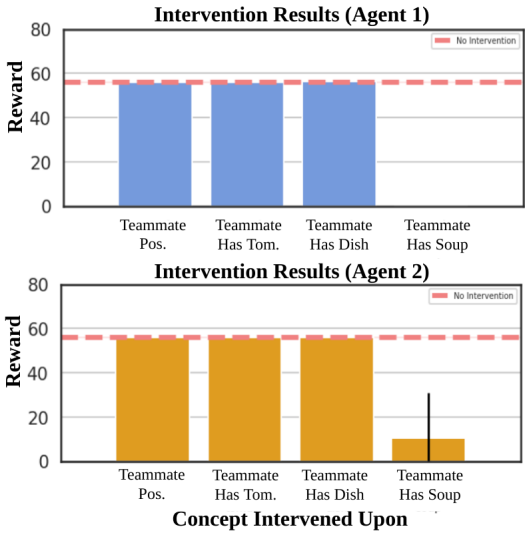

Figure 9: Results of iterative concept intervention on Concept PPO agents trained without orientation.