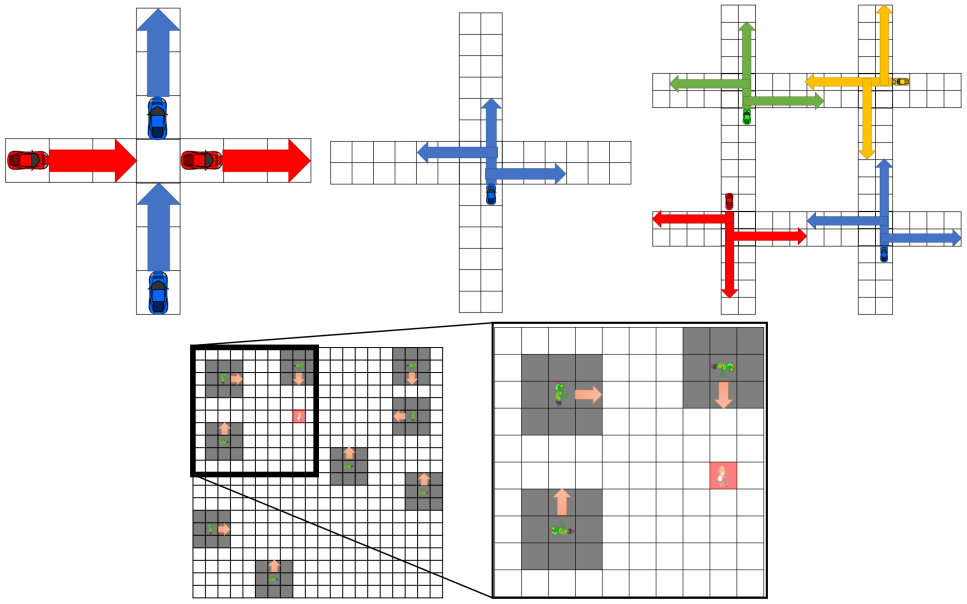

Figure 6: Above are the easy, medium, and hard traffic junction environments. Visibility is limited to the cell the car is located, so agents are effectively blind. The bottom shows a zoomed-in view of the 20×20 predator-prey environment. The predators are denoted by green aliens, while the prey is denoted by a human (in a red square).

A Experimental Setup

We train and evaluate our model in a blind traffic junction and predator-prey environment settings following prior benchmarks [19, 20, 4]. For each of these variants, we train on 10 random seeds and one epoch uses 5000 samples. We used an RMSProp optimizer with a learning rate of 0.003. See Figure 6.

The blind traffic junction scenario involves multiple agents navigating a discretized narrow intersection with no observability regarding the locations of the other agents. Clearly, this necessitates informative communication in order to avoid collisions in the environment. Note that both communication and action occur in a single time-step. We study three variants of the blind traffic junction and report results on the easiest and hardest environments which converge for continuous and discrete communication.

The predator-prey scenario involves multiple agents, where one agent is denoted as the prey and the remaining agents are denoted as predators. The predator agents move and search the environment for the prey agent. The predator agents can only observe its current cell and the adjacent cells (limited visibility to 1 cell around itself). The episode terminates when all predator agents reach the prey agent or when the maximum episode length is hit.

Predator-prey does not necessarily require communication to solve the task. However, in the fully-cooperative predator-prey environment, predators are rewarded for maximizing the number of predators who reach the discovered prey. Thus, there is no built-in incentive for fully-cooperative teams to decrease total communication. In our experiments, we show that our method, IMGS-MAC, is able to decrease messaging to a minimum sparse budget b^* with lossless performance.

Overall, our proposed method is trained ("pretraining") using the autoencoder in Eq. 3. We then analyze to determine if our model will follow a lossless sparse budget. If not, we finetune our model for a suboptimal sparse budget (Def. 3.2) using the message penalty in Eq. 4.

We use REINFORCE [26] to train both the gating function and policy network subject to the previous constraints. In order to calculate the information similarity, we compute loss, using Eq. 3, between each agent's decoded state $s_t^{i,\texttt{decoded}}$ and the concatenation of all agents' states s_t.