

# APPENDIX

## TABLE OF CONTENT

A RL for Collective Behavior Generation	1
B Additional Materials for Feature-level Explanation	4
C Additional Materials for Policy-level Explanation for Herding Behavior	7
D Additional Materials for Policy-level Explanation for Flocking Behavior	10
E Explain the Swirling Behavior	13
F Shape Assembly of Robot Swarms	19

## A RL FOR COLLECTIVE BEHAVIOR GENERATION

### A.1 IMPLEMENTATION DETAILS

The specific parameters of environment are shown in Table 1:

Table 1: Environment parameters.

Parameters	Value	Unit
Mass of predator	1	kg
Size of predator	$6 \times 10^{-2}$	m
Mass of prey	1	kg
Size of prey	$3.5 \times 10^{-2}$	m
Max speed	$5 \times 10^{-1}$	$\text{m} \cdot \text{s}^{-1}$
Max linear acc.	1	$\text{m} \cdot \text{s}^{-2}$
Max angular vel.	$5 \times 10^{-1}$	$\text{rad} \cdot \text{s}^{-1}$
Env edge length	2	m
Contact stiffness	50	$\text{N} \cdot \text{m}^{-1}$
Drag coefficient	2	$\text{N} \cdot \text{s} \cdot \text{m}^{-1}$
Time step	$1 \times 10^{-1}$	s

The prey and predators as two distinct species of agents, where agents of the same species are assumed to be homogeneous. This assumption is reasonable, as agents within a species are considered to have identical capabilities, responsibilities, and objectives. Homogeneity has been widely adopted in swarm system modeling. For example, classical works such as Vicsek et al. (1995); Couzin et al. (2002) design unified control laws that govern the behavior of all agents in a swarm. Similarly, recent MARL-based studies like Hahn et al. (2019); Durve et al. (2020); Monter et al. (2023) implement homogeneity through parameter sharing in neural networks across agents.

Conspecifics share one critic and one actor network. The critic network is used to evaluate the quality of an action taken by the agent, by estimating the expected future reward from that action, while the actor determines the agent’s action  $a = [a_F, a_R]^T$  based on its current observation. Together, the critic and actor work to improve the agent’s behavior over time, by continuously refining its estimates of expected future rewards and adjusting its policy accordingly.

The critic is designed to be decentralized allowing agents to evaluate based solely on local observations, without the knowledge of global states and actions like centralized critics used in Lowe et al. (2017); Hüttenrauch et al. (2019). This is analogous to the situation when living organisms can only perceive nearby surroundings with limited abilities. Additionally, a decentralized critic provides better scalability as the number of agents in the system increases, making it particularly beneficial for swarm systems.

Collective behaviors emerge under various actor-critic MARL frameworks, such as MAPPO (Kölle et al., 2024; Yu et al., 2022) and MADDPG (Li et al., 2023; Lowe et al., 2017). Both critic and

actor are encoded by deep feed-forward neural networks with rectified linear unit activation with an input dimension  $d_o$  equivalent to the length of the observation vector. Each network consists of three hidden layers with 64 neurons per layer. The output dimension of the actor network is 2 and the output dimension of the critic network is 1.

Two replay buffers  $\mathcal{B}_0$  and  $\mathcal{B}_1$  are used for predators and prey, respectively. There is no need to construct replay buffers for each individual because the homogeneity allows for interchangeable use of the experiences collected by conspecifics. Therefore, experiences collected from multiple conspecifics can be congregated into a single replay buffer, resulting in more resilient and effective learning outcomes.

We designate the dimensionality of the observation and action vectors as  $d_o$  and  $d_a$ , respectively. We denote the discount factor, which determines the weight given to future rewards, as  $\gamma$ , and the soft update rate, which determines the speed at which a target network is updated towards a learning network, as  $\tau$ . The used algorithm of MADDPG is summarized in Algorithm 1:

---

**Algorithm 1** Multiagent deep deterministic policy gradient algorithm for predator–prey coevolution

---

```

1: //  $i = 0$  for predators,  $i = 1$  for prey
2: for species  $i = 0$  to 1 do
3:   Randomly initialize actor  $\mu_i$  parameterized by  $\theta_i^\mu$  and critic  $Q_i$  parameterized by  $\theta_i^Q$ ;
4:   Initialize target actor  $\mu'_i$  and target critic  $Q'_i$ ,  $\theta_i^{\prime\mu} \leftarrow \theta_i^\mu$ ,  $\theta_i^{\prime Q} \leftarrow \theta_i^Q$ ;
5: end for
6: for episode = 1 to  $M$  do
7:   Randomly spawn  $n_0$  predators and  $n_1$  prey; receive observations  $o_i \in \mathbb{R}^{n_i \times d_o}$ ;
8:   for  $t = 1$  to max-episode-length do
9:     For species  $i$ , select actions  $a_i = \mu_{\theta_i}(o_i) + \mathcal{N}_t \in \mathbb{R}^{n_i \times d_a}$ , where  $\mathcal{N}_t$  is Gaussian noise;
10:    Step environment with  $(a_0, a_1)$ , receive  $(r_0, o'_0)$  and  $(r_1, o'_1)$ ;
11:    For species  $i$ , store  $(o_i, a_i, r_i, o'_i) \in (\mathbb{R}^{n_i \times d_o}, \mathbb{R}^{n_i \times d_a}, \mathbb{R}^{n_i}, \mathbb{R}^{n_i \times d_o})$  in replay buffer  $\mathcal{B}_i$ ;
12:     $o_i \leftarrow o'_i$ 
13:    for species  $i = 0$  to 1 do
14:      Randomly sample a mini-batch of  $S \in \mathbb{N}^+$  samples  $(o_i^j, a_i^j, r_i^j, o_i^{\prime j})$  from  $\mathcal{B}_i$ ;
15:      Set  $y_i^j = r_i^j + \gamma Q'_i(o_i^{\prime j}, a_i^{\prime j})|_{a_i^{\prime j} = \mu'_i(o_i^{\prime j})}$ ;
16:      Update critics by minimizing the loss  $\mathcal{L}(\theta_i^Q) = \frac{1}{S} \sum_{j=1}^S (y_i^j - Q_i(o_i^j, a_i^j))^2$ ;
17:      Update actors using sampled policy gradient:

$$\nabla_{\theta_i^\mu} J \approx \frac{1}{S} \sum_{j=1}^S \nabla_{\theta_i^\mu} \mu_i(o_i^j) \nabla_{a_i} Q_i(o_i^j, a_i)|_{a_i = \mu_i(o_i^j)}$$

18:      Soft-update target network parameters:

$$\theta_i^{\prime\mu} \leftarrow \tau \theta_i^\mu + (1 - \tau) \theta_i^{\prime\mu}, \quad \theta_i^{\prime Q} \leftarrow \tau \theta_i^Q + (1 - \tau) \theta_i^{\prime Q}$$

19:    end for
20:  end for
21: end for

```

---

The hyperparameters for RL training are summarized in Table 2. We employed the Adam optimizer for optimization. The training was conducted on a laptop equipped with an NVIDIA GeForce RTX 4090 GPU, an Intel Core i9-14900K CPU, and 128 GB of RAM, taking approximately 40 minutes to complete RL training.

All of the code implementation is attached in the attachment.

## A.2 EFFECTS OF PARAMETERS FOR COLLECTIVE BEHAVIOR GENERATION

To evaluate the robustness of the emergent collective behavior, we investigate how variations in agents' perception configurations affect the outcome. Specifically, we vary two key perception parameters: (1) the maximum perception range, and (2) the maximum number of neighbors each



Table 2: Parameters of the algorithm.

Parameters	Value
Number of episodes	2000
Episode length	500
Number of hidden layers	3
Hidden layer size	64
Learning rate of actor	$1 \times 10^{-4}$
Learning rate of critic	$1 \times 10^{-3}$
Discount factor	0.95
Soft-update rate	0.01
Initial exploration rate	0.1
Initial noise rate	0.1
Replay buffer size	$5 \times 10^5$
Batch size	256

agent can observe. As shown in Figure 1, despite changes in these parameters, prey agents consistently exhibit emergent collective behaviors. These results suggest that the emergence of swarm behavior is not tightly dependent on precise perception settings. The policy learned through reinforcement learning generalizes well across different sensory configurations, highlighting the robustness of our approach. We further test the sensitivity of swarm formation to variations in the survival-based reward

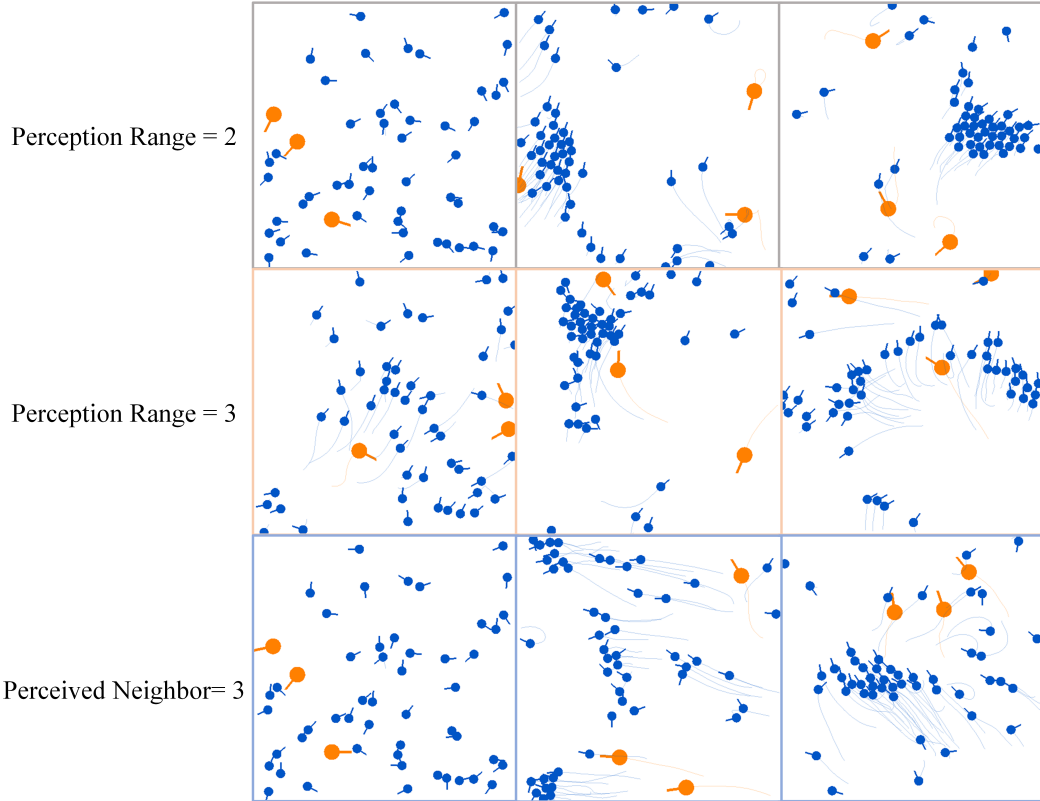


Figure 1: Robustness to perception variation: Collective behavior still emerges across different perception ranges and neighbor counts.

function. Figure 2 presents the results of experiments using different reward magnitudes. Prey agents consistently develop similar collective behaviors, indicating that the emergence of swarming is not

overly sensitive to the specific form of the survival reward. This reinforces the claim that survival pressure alone is sufficient to drive the evolution of coordinated strategies without requiring carefully tuned incentives.

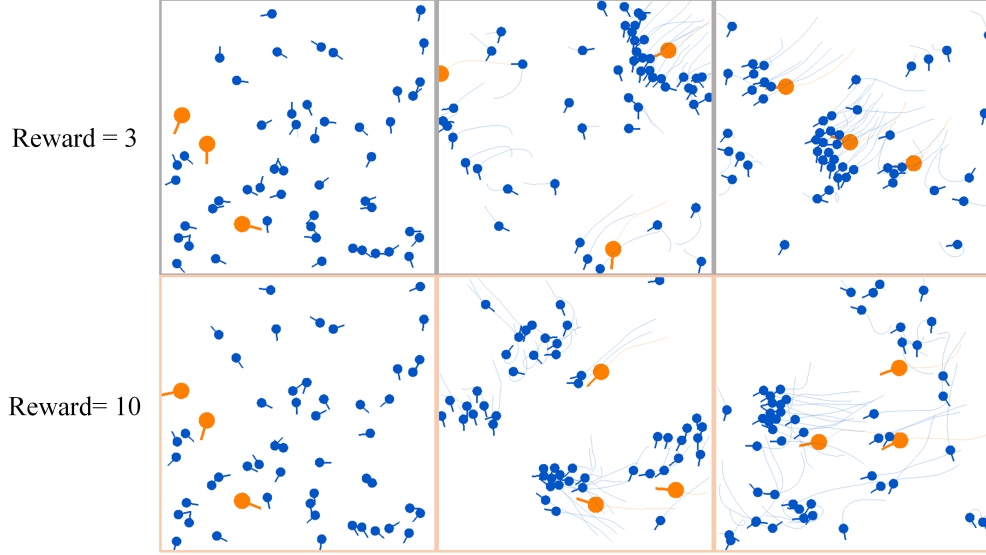


Figure 2: Robustness to reward variation: Collective behavior remains stable across different survival reward formulations.

### A.3 EFFECTS OF AGENT POPULATION SIZE

We also examine how varying the number of agents during training influences the emergence of collective behavior. Figure 3 illustrates the outcomes under different agent population sizes. When no predators are present, prey do not evolve coordinated movement, regardless of their number—highlighting the necessity of survival pressure for swarming behavior to emerge. However, once predators are introduced, even a small number of prey can learn to coordinate and exhibit herding behavior. This result indicates that our training framework exhibits strong scalability: effective swarm behavior can be learned even with a minimal agent population, and the learned policy generalizes well to larger groups during evaluation.

## B ADDITIONAL MATERIALS FOR FEATURE-LEVEL EXPLANATION

### B.1 ABLATION EXPERIMENTS: EFFECT OF REMOVING RELATIVE POSITION FEATURES

To further verify the necessity of specific observation features in driving the emergence of collective behavior, we conduct an ablation experiment in which the observation of neighboring agents’ relative positions is entirely removed from the prey’s input. As shown in Figure 4, the resulting behavior demonstrates a clear failure to exhibit any form of swarming or coordination.

In this setting, the predator follows a fixed policy that always targets and pursues the nearest prey. Despite this consistent predation pressure, the lack of relative position information prevents prey from organizing into cohesive groups. This result highlights that relative positional awareness of nearby agents is a critical prerequisite for herding behavior to emerge. Without this information, the policy network lacks the necessary spatial context to drive alignment or aggregation, even under the influence of survival-based rewards.

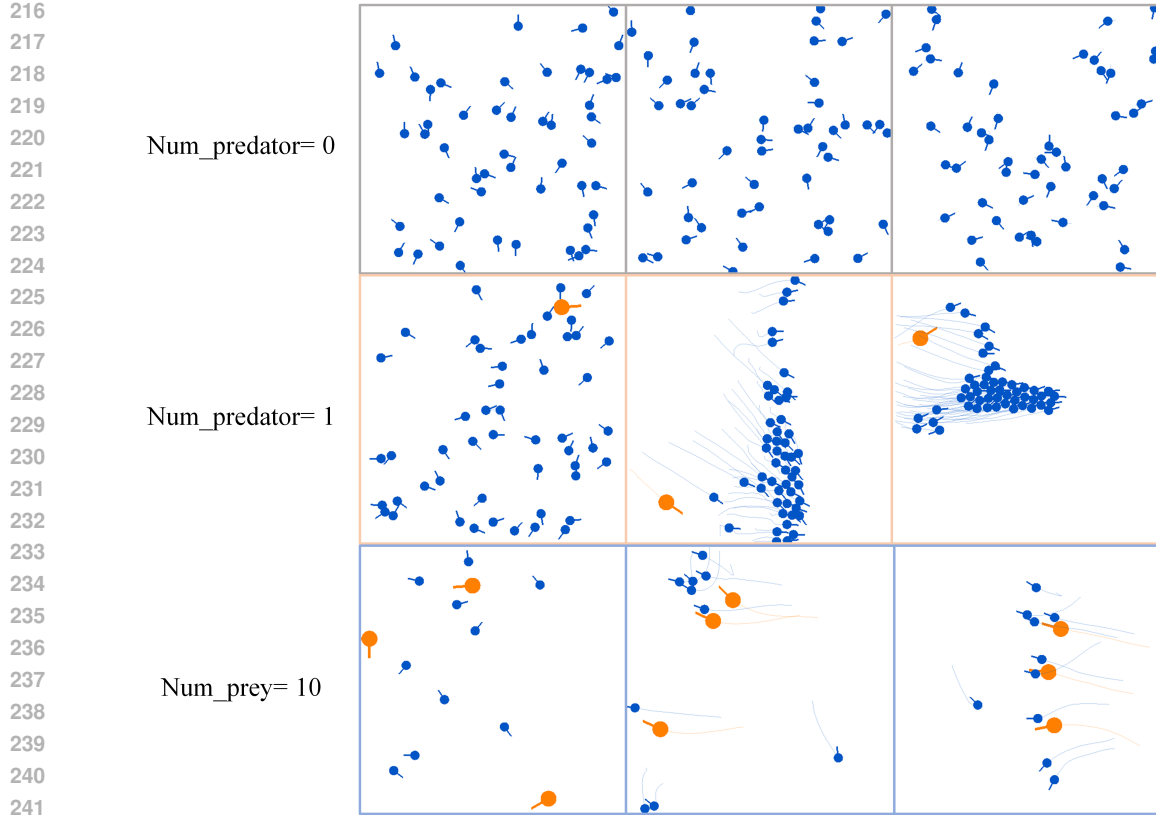


Figure 3: Robustness to agent count: With predator pressure, collective behavior emerges across different population sizes.

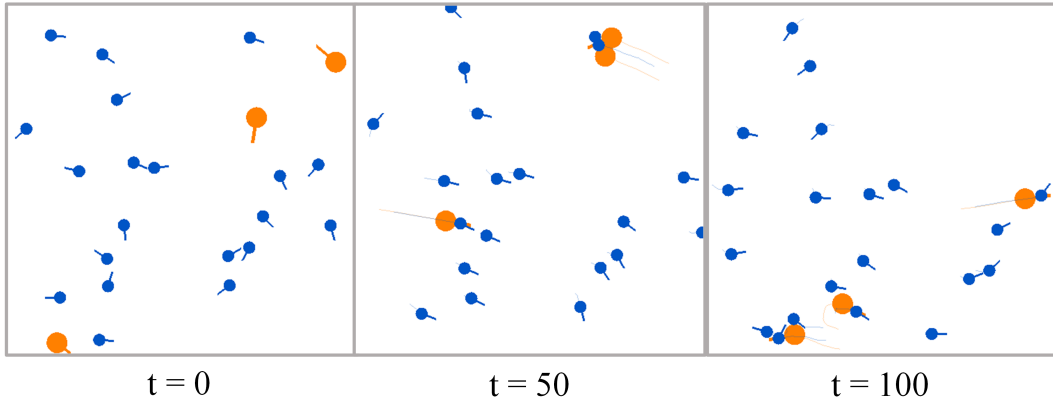


Figure 4: After removing the observation of neighbors' relative positions, prey agents no longer exhibit collective behavior. In this setting, the predator follows a policy that actively pursues the nearest prey.

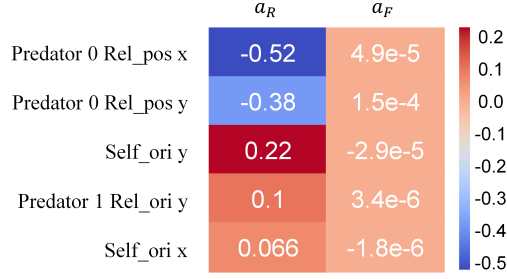


Figure 5: Integrated gradients analysis of the actor network.

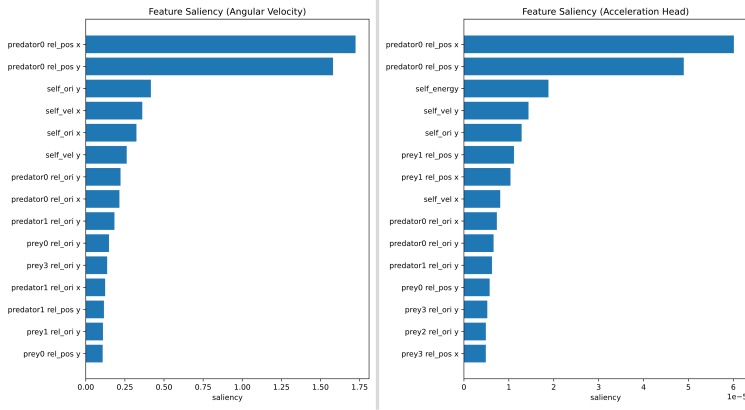


Figure 6: Saliency map analysis of the actor network.

## B.2 FEATURE ATTRIBUTION ANALYSIS

### B.2.1 COMPARISON OF DIFFERENT FEATURE ATTRIBUTION METHODS FOR POLICY NETWORK

To further validate the feature attribution results, we additionally apply the Saliency map and Integrated Gradients method Sundararajan et al. (2017) to the policy network.

Figure 5 presents the IG values of the top five observation features with the highest attributions. Each row corresponds to a selected feature, and each column represents a dimension of the policy network output. The values reflect the contribution of each feature to the output: larger absolute values indicate greater influence, while the sign denotes whether the effect is positive or negative. Figure 6 the results of the saliency map with the top 15 observation features with the highest attributions.

The results are consistent with those obtained using SHAP in the main text, reinforcing the conclusion that the relative position of the nearest predator is the most influential feature affecting the policy output  $a_R$ .

### B.2.2 FEATURE ATTRIBUTION FOR CRITIC NETWORK

We also perform feature attribution analysis on the critic network using both SHAP and Integrated Gradients. The results, shown in Figure 7, demonstrate that the relative position of the nearest predator is again the most influential observation feature for the value function estimation. Both attribution methods highlight the same dominant feature, providing further evidence of the consistency and robustness of our analysis framework.

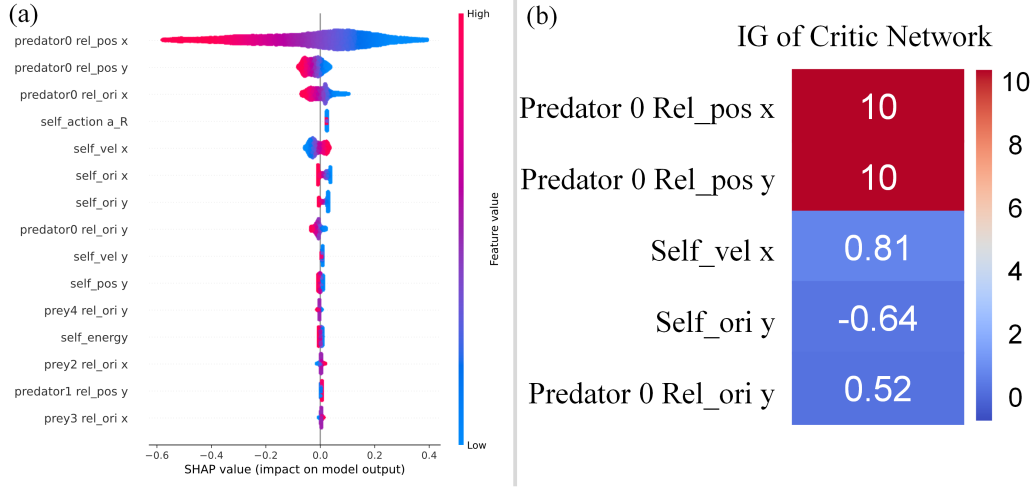


Figure 7: Feature attribution analysis for the critic network: (a) SHAP values and (b) Integrated Gradients.

## C ADDITIONAL MATERIALS FOR POLICY-LEVEL EXPLANATION FOR HERDING BEHAVIOR

### C.1 SENSITIVITY ANALYSIS OF ARM

This section investigates the sensitivity of the Agent Response Map (ARM) to variations in parameters which includes the virtual prey’s orientation and velocity, agent population size, observation noise interference, and sampling resolution:

1. **Velocity & Orientation:** Variations in velocity have minimal effect on the resulting maps. While changes in orientation alter specific action outputs, the ARM consistently identifies the Voronoi diagram boundaries.
2. **Population Size:** Altering the number of agents changes the shape of the Voronoi diagram, yet the ARM remains capable of accurately characterizing these geometric structures.
3. **Noise:** Different noise amplitudes affect the magnitude of policy outputs (shifting the color scale) and influence pursuit-evasion strategies. However, the ARM retains its ability to reveal the fundamental Voronoi structure.
4. **Sampling Resolution:** Resolution changes show limited impact on overall patterns. While lower resolutions significantly accelerate computation, they result in a minor loss of precision, particularly near the Voronoi boundaries.

Regardless of parameter variations, the ARM consistently captures stable spatial patterns, demonstrating its robustness for interpreting swarm behavior.

#### C.1.1 IMPACTS OF VELOCITY AND ORIENTATION OF VIRTUAL AGENTS

In the main text, the virtual prey is initialized with a heading angle  $\theta = 0$  (i.e.,  $h = [1, 0]^T$ ). Here, we examine an alternative case by setting the heading to  $\theta = \pi$  (i.e.,  $h = [-1, 0]^T$ ). The resulting ARM visualization under the same environment configuration is shown in Figure 8. Compared to Figure 4 in the main paper, we observe that the ARM of the critic network and  $a_F$  remain largely unchanged. In contrast, the ARM of  $a_R$  exhibits differences, which are expected since the required turning actions depend on the orientation of the virtual prey. Importantly, discontinuities in  $a_R$  still occur along the boundary of the predator’s Voronoi region, indicating that ARM retains consistent structural patterns even under heading variation. Additionally, we examine the effect of virtual prey velocity. While the main text sets a velocity of  $v = [0, 0]^T$ , we test two alternative settings:  $v = [v_{\max}, 0]^T$

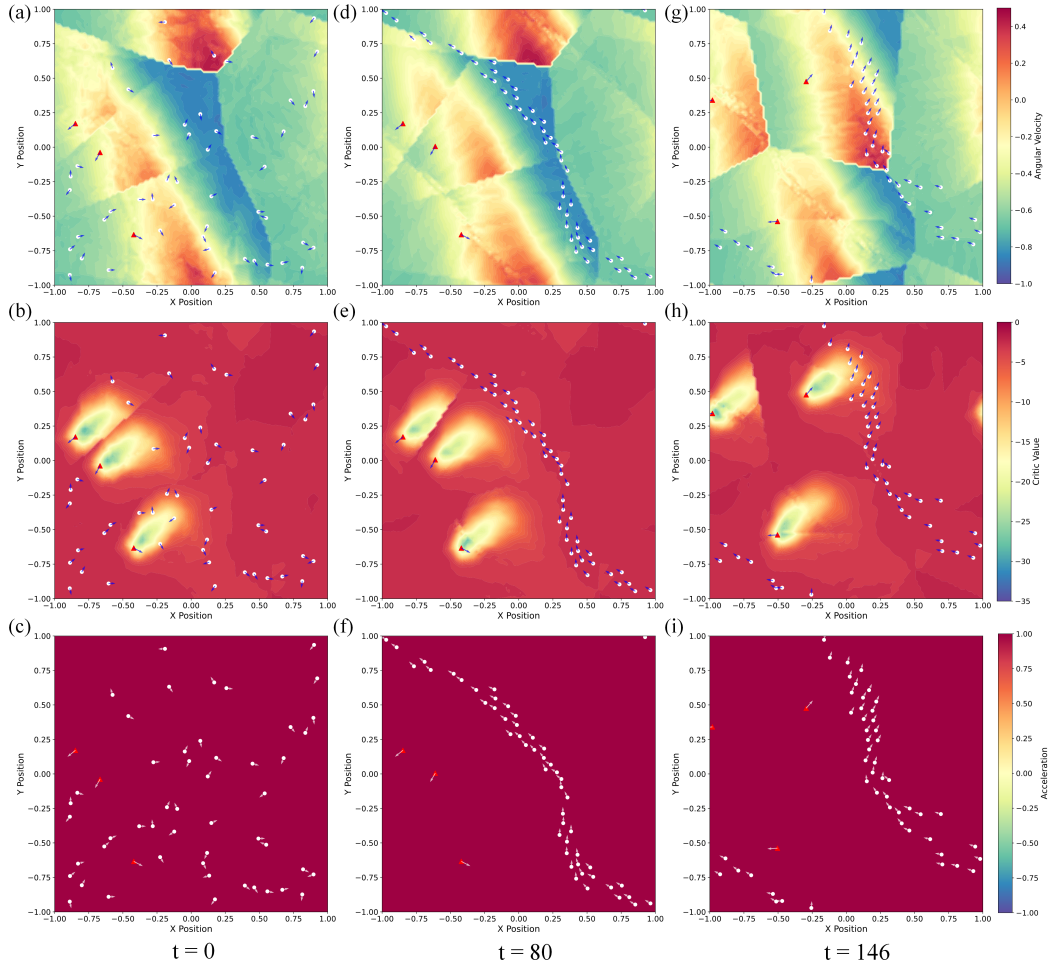


Figure 8: ARM for herding behavior with orientation  $\theta = \pi$ : Each column corresponds to a time step. The first row shows the ARM of  $a_R$ , the second row shows the ARM of the critic network, the third row shows the ARM of  $a_F$ . Color intensity indicates the output magnitude. Red triangles denote predators, white circles represent prey, and arrows indicate orientation.

and  $v = [v_{\max}, v_{\max}]^T$ . The resulting ARM visualizations are shown in Figure 9 and Figure 10, respectively. Compared with Figure 4 in the main context, we find that changes in velocity lead to only minor modifications in the critic value maps, while the ARM outputs of  $a_R$  and  $a_F$  remain almost identical.

These results suggest that ARM is robust to variations in both orientation and velocity of the virtual prey, and reliably captures common patterns indicative of collective behavior mechanisms.

### C.1.2 IMPACTS OF POPULATION SIZE, NOISE AND SAMPLING RESOLUTION OF ARM

We investigated the impact of noise magnitude, population size, and sampling resolution on the ARM results. Specifically, we injected Gaussian white noise of varying intensities ( $\sigma \in \{0.02, 0.05\}$ ) into the agents' observation vectors. As shown in Figure 11, the Critic value field and Voronoi structures revealed by the ARM remain clearly visible.

Altering the number of agents changes the shape of the Voronoi diagram, yet the ARM remains capable of accurately characterizing these geometric structures, which are shown in Figure 13 and Figure 14.

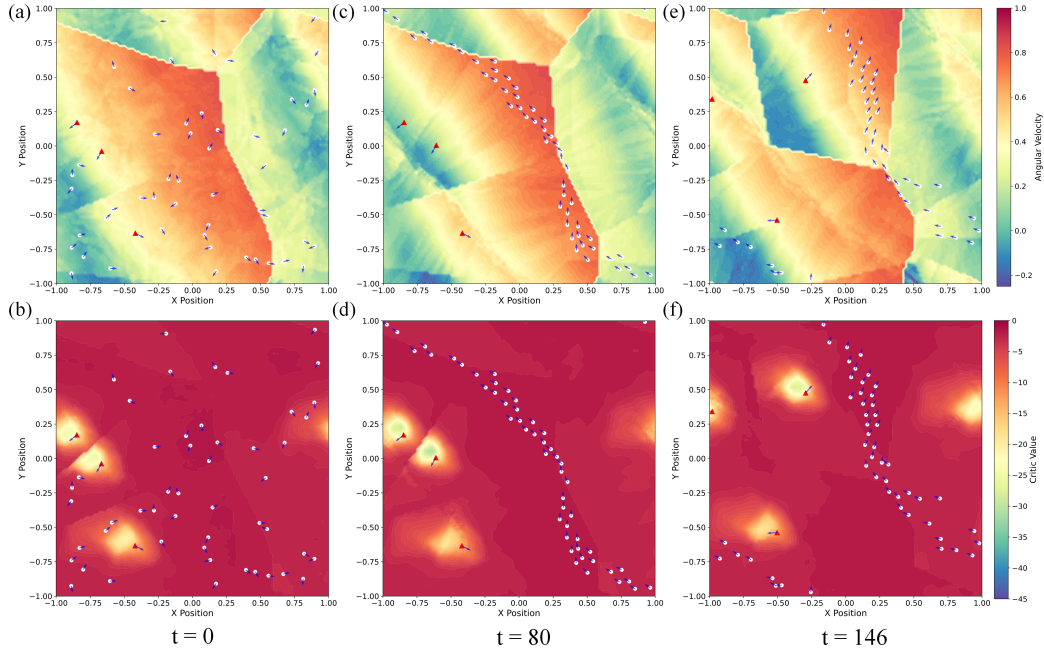


Figure 9: ARM for herding behavior with velocity  $v = [v_{\max}, 0]^T$ : Each column corresponds to a time step. The first row shows the ARM of  $a_R$ , the second row shows the ARM of the critic network. Color intensity indicates the output magnitude. Red triangles denote predators, white circles represent prey, and arrows indicate orientation.

Regarding sampling resolution, changes show limited impact on overall patterns. Regarding sampling resolution, variations have a limited impact on the overall patterns. While the original study utilized a sampling resolution of 100 (i.e., 100 virtual probes per axis), we additionally evaluated resolutions of 50 and 80. These results are presented in Figure 15 and Figure 16, respectively.

## C.2 ROBUSTNESS OF ARM ACROSS RANDOM SEEDS

To evaluate the robustness of the Agent Response Map (ARM), we examine its consistency across different random seeds. As shown in Figure 17, although the initial positions of predators and prey vary across seeds, the resulting ARMs consistently capture critical spatial patterns. Notably, the ARM of  $a_R$  in Figure 17(a), (d), and (g) reveals clear discontinuities along the boundary of the predator’s Voronoi diagram. Similarly, the ARM of the critic network—shown in Figure 17(b), (e), and (h)—indicates that the critic value is highest near this boundary. These findings demonstrate that ARM robustly identifies essential structural patterns in agent behavior regardless of the stochastic initialization, validating its general applicability.

## C.3 EVOLUTION OF ARM DURING TRAINING

We further analyze how ARM evolves throughout the MARL training process. The model is trained for 2000 epochs, and we visualize the ARM at every 200-epoch interval to capture temporal changes. The results are shown in Figure 18 and Figure 19.

In the early stages (epochs 0–600), the ARM of  $a_R$  appears nearly uniform across the environment, indicating that prey have not yet learned meaningful spatial distinctions such as the predator’s Voronoi boundary. Beginning around epoch 800, we observe emerging discontinuities in  $a_R$ , signifying that prey start to recognize spatial cues and adjust their orientation accordingly. Concurrently, the ARM of  $a_F$  reveals an expanding region where agents apply maximal acceleration, suggesting a growing ability to evade predators.



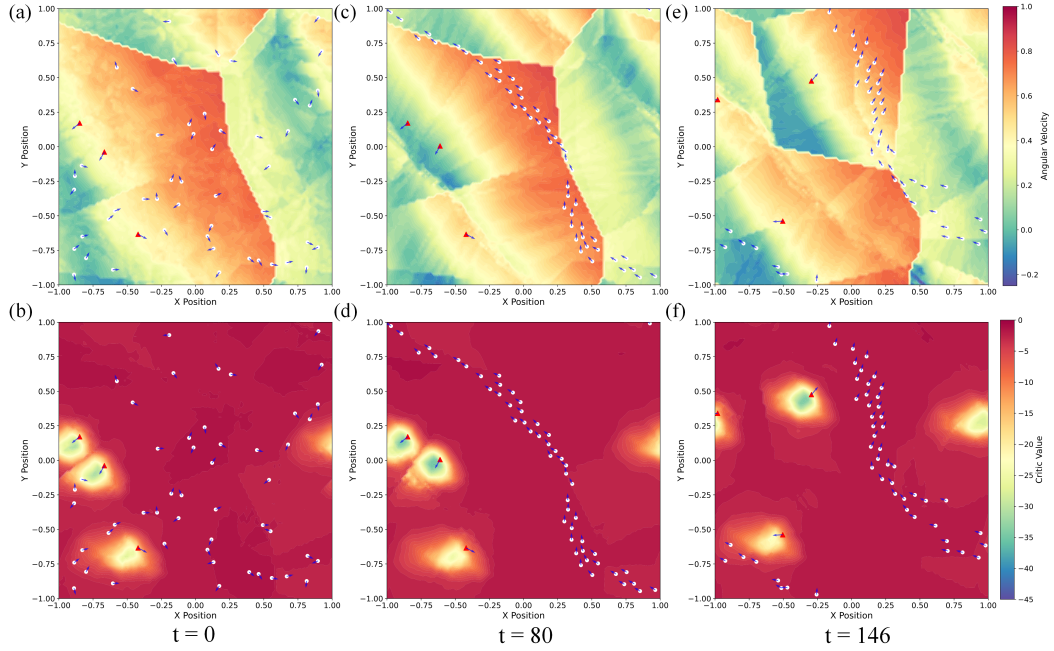


Figure 10: ARM for herding behavior with velocity  $v = [v_{\max}, v_{\max}]^T$ : Each column corresponds to a time step. The first row shows the ARM of  $a_R$ , the second row shows the ARM of the critic network. Color intensity indicates the output magnitude. Red triangles denote predators, white circles represent prey, and arrows indicate orientation.

These results collectively indicate that the safety zone—corresponding to the boundary of the predator’s Voronoi region—is not predefined or hardcoded in the environment, but instead emerges spontaneously through MARL training. Simultaneously, agents learn to maximize acceleration as an evasive strategy, highlighting the adaptive nature of the learned policy.

#### C.4 SPATIAL GRADIENT MAGNITUDE OF ARM

The Spatial Gradient Magnitude (SGM) is defined as a quantitative metric for ARM discontinuity, formulated as  $M(x, y) = \|\nabla a_R(x, y)\|$ . We evaluated this metric across different random seeds: Figure 20 corresponds to the scenario in the main text, while Figure 21 illustrates an alternative seed. The results consistently show that the regions with the highest SGM align with the Voronoi diagram boundaries, corroborating our ARM analysis.

## D ADDITIONAL MATERIALS FOR POLICY-LEVEL EXPLANATION FOR FLOCKING BEHAVIOR

The key difference between the herding and flocking behaviors lies in the presence of a predator: herding occurs when predators are present, whereas flocking emerges in their absence. Notably, despite the absence of predators during flocking, their presence remains essential during training. Without predator-induced survival pressure, prey fail to evolve swarm strategies—exhibiting neither reduced inter-agent distance nor aligned orientations, as shown in Appendix A.3. Therefore, both behaviors are shaped under identical training conditions, driven by the same underlying policy network trained on similar observation data.

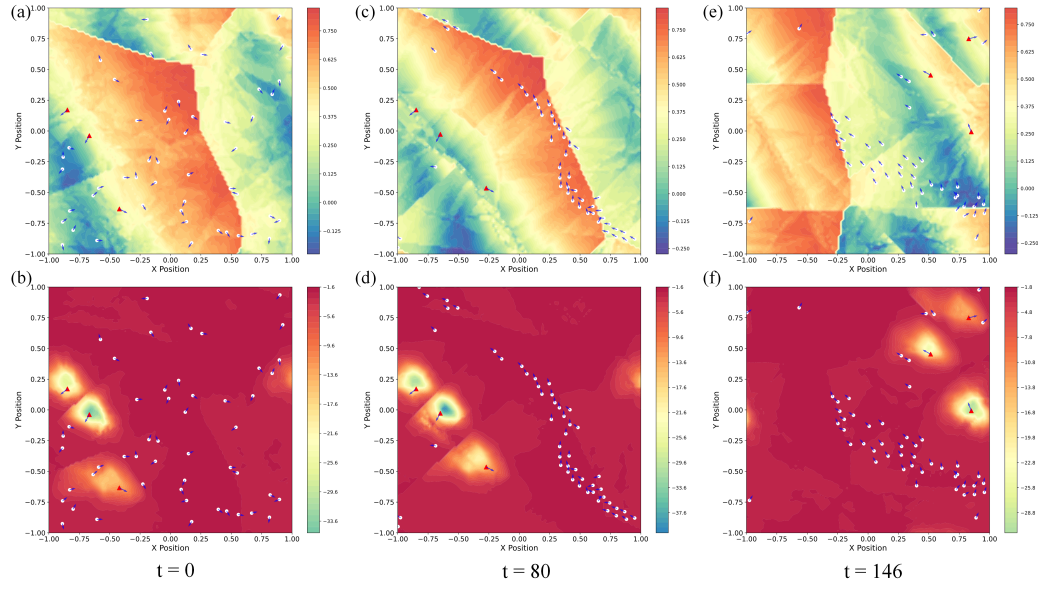


Figure 11: ARM for herding behavior with noise  $\sigma = 0.05$ : Each column corresponds to a time step. The first row shows the ARM of  $a_R$ , the second row shows the ARM of the critic network. Color intensity indicates the output magnitude. Red triangles denote predators, white circles represent prey, and arrows indicate orientation.

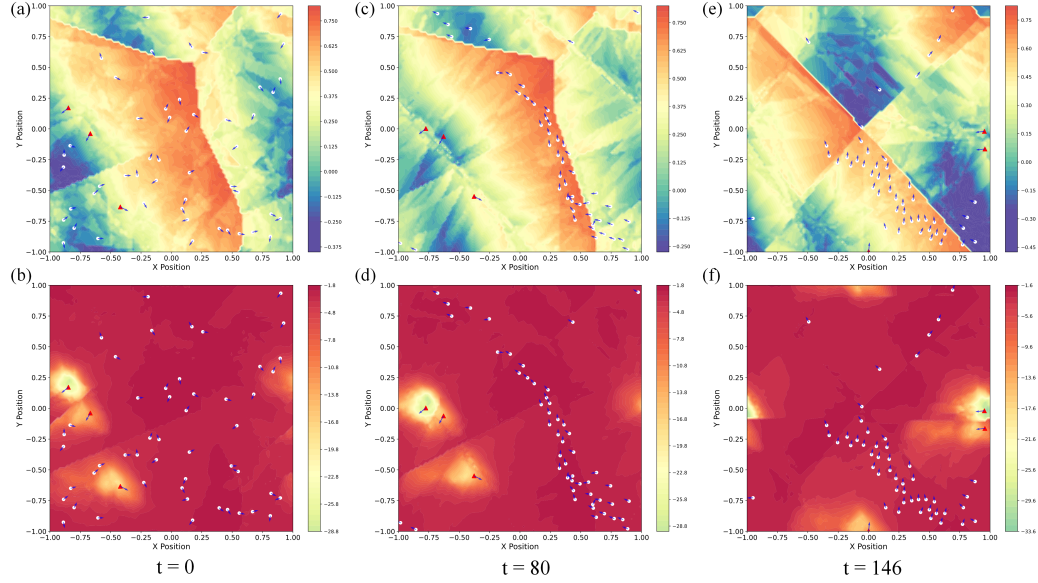


Figure 12: ARM for herding behavior with noise  $\sigma = 0.02$ : Each column corresponds to a time step. The first row shows the ARM of  $a_R$ , the second row shows the ARM of the critic network. Color intensity indicates the output magnitude. Red triangles denote predators, white circles represent prey, and arrows indicate orientation.

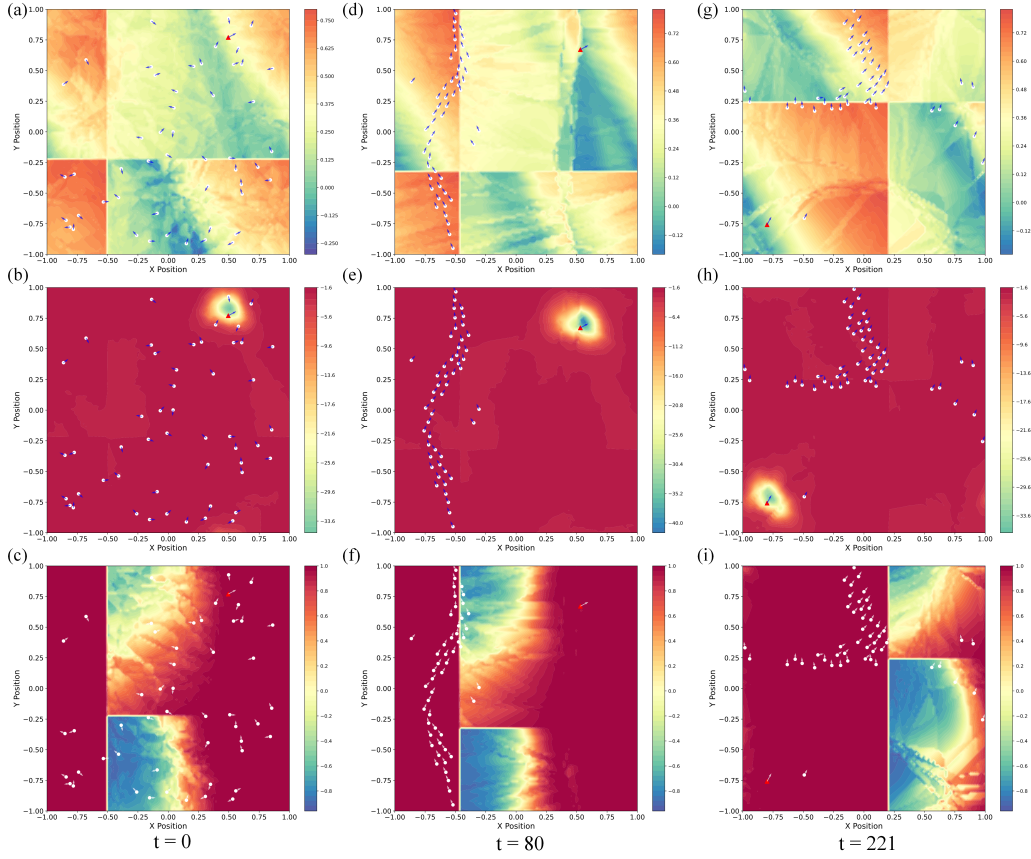


Figure 13: ARM for herding behavior with 1 predator: Each column corresponds to a time step. The first row shows the ARM of  $a_R$ , the second row shows the ARM of the critic network, the third row shows the ARM of  $a_F$ . Color intensity indicates the output magnitude. Red triangles denote predators, white circles represent prey, and arrows indicate orientation.

#### D.1 ARM OF $a_F$

Under flocking behavior, Figure 1(e) and (f) show that while the orientations of prey become aligned, their relative distances remain largely unchanged. This phenomenon can also be explained using the Agent Response Map (ARM), as illustrated in Figure 22, where (a) presents the ARM for  $a_R$ , and (b) for critic value. Figure 22(a) shows that the value of  $a_R$  is mostly distributed within the range of -0.2 to 0.2, causing prey at different locations to adopt similar orientations. Combined with the observation that the  $a_F$  (shown in Figure 23) remains near its maximum value throughout the task space, prey maintain similar velocities in both the x and y directions. Consequently, no relative velocity differences emerge, and the relative distance of prey does not decrease. Additionally, critic values in the absence of predators are lower than those with predators, as seen in Figure 22(b). While initially counterintuitive, this outcome results from the multilayer perceptron (MLP) network architecture: removing predators sets their observations to zero, implying minimal distances and thus low critic values. Future work will explore alternative architectures (e.g., invariant networks) to better handle agent removal scenarios.

To further investigate the flocking phenomenon, we present the Agent Response Map (ARM) of  $a_F$  in Figure 23. The results show that prey maintain near-maximum acceleration values across the entire environment, regardless of their spatial location. This observation is consistent with the conclusion drawn in the main text: under flocking behavior, high acceleration is universally adopted by prey, while their relative distances remain stable.

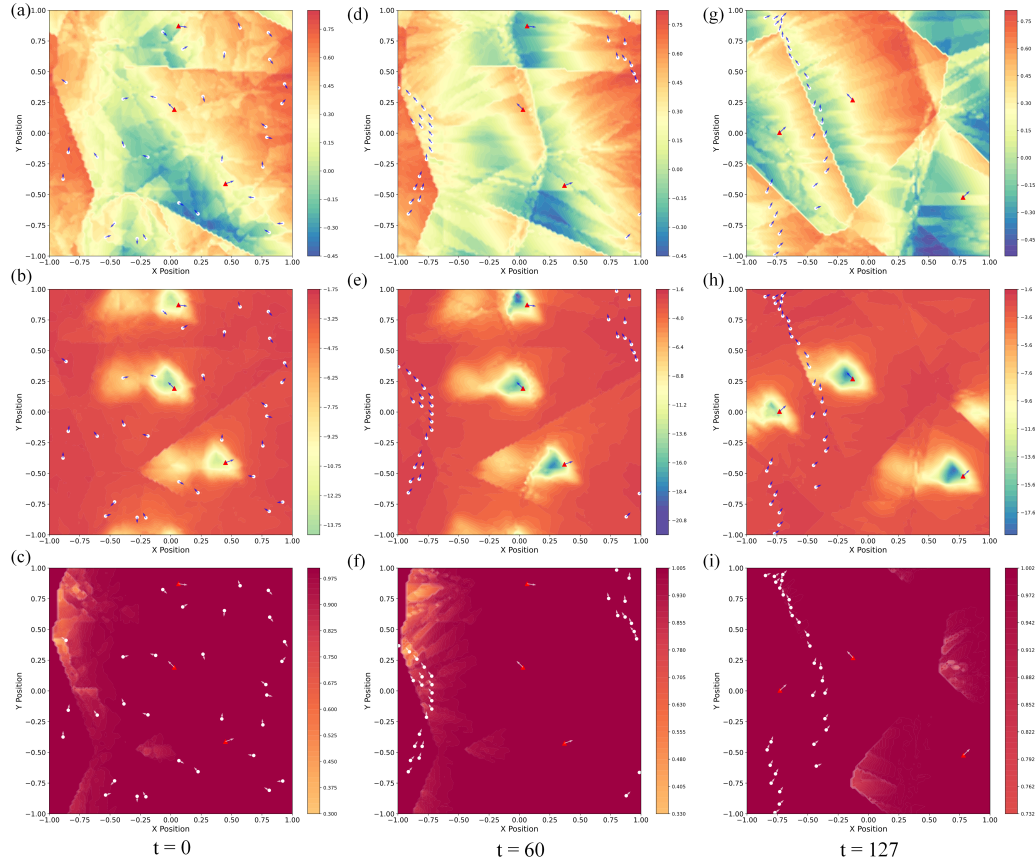


Figure 14: ARM for herding behavior with 30 prey: Each column corresponds to a time step. The first row shows the ARM of  $a_R$ , the second row shows the ARM of the critic network, the third row shows the ARM of  $a_F$ . Color intensity indicates the output magnitude. Red triangles denote predators, white circles represent prey, and arrows indicate orientation.

## D.2 ROBUSTNESS TO VIRTUAL PREY ORIENTATION AND RANDOM SEEDS

We further examine the robustness of ARM-based analyses under flocking settings by varying the random seed and virtual prey orientation. Figure 24 shows results obtained using a different random seed, while Figure 25 presents the ARM for a virtual prey oriented along  $h = [-1, 0]^T$ . In both cases, the spatial distributions of  $a_R$ , critic value, and  $a_F$  remain largely consistent with the default setting.

These findings confirm that ARM robustly captures consistent behavioral patterns in flocking behavior, regardless of initialization or virtual agent heading, supporting the generality and reliability of the method.

## E EXPLAIN THE SWIRLING BEHAVIOR

In this section, we continue to apply our two-level explanation framework to investigate the emergence of swirling behavior, which arises under confined boundary conditions. In the confined environment, prey agents develop swirling patterns through training, characterized by orientation alignment and reduced relative distances—even in the absence of predators. This distinguishes swirling from flocking behavior. Therefore, we aim to explain why prey are able to exhibit swirling behavior in confined settings regardless of whether predators are present.



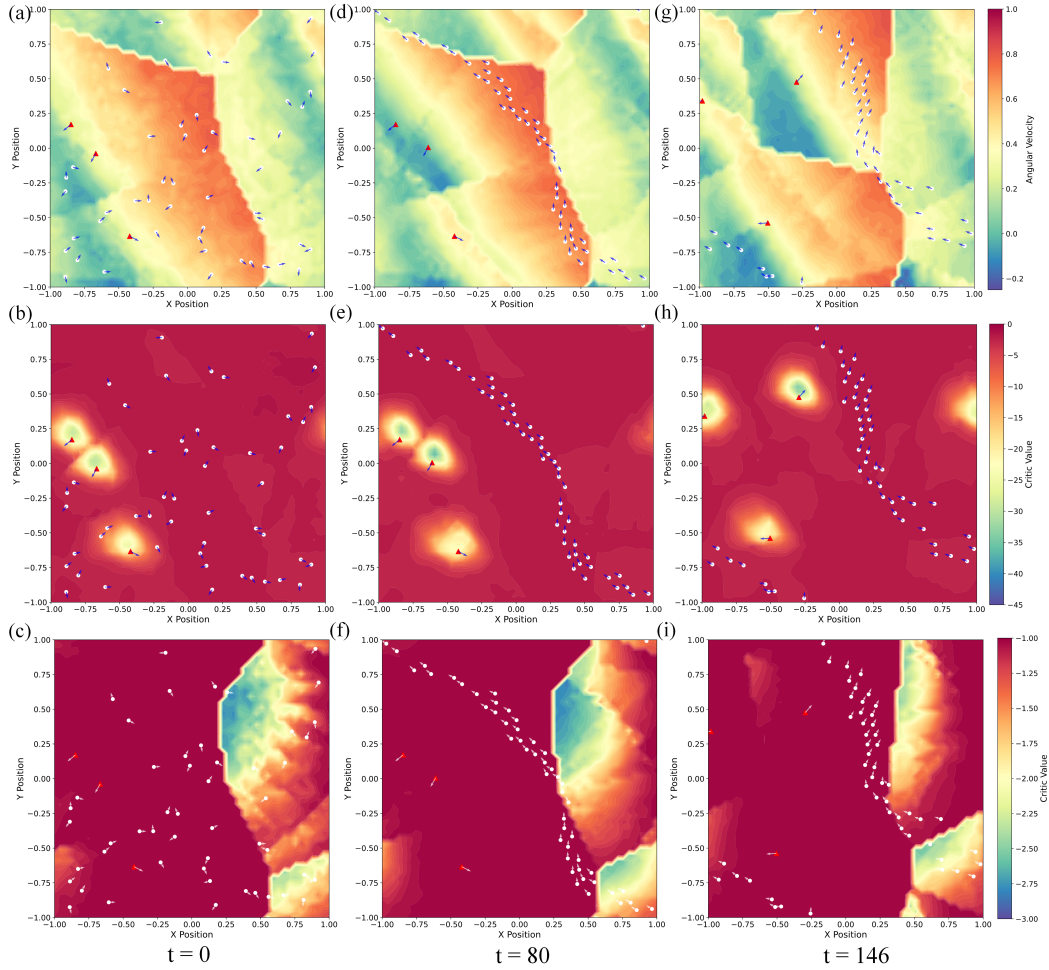


Figure 15: ARM for herding behavior with 50 sampling resolution: Each column corresponds to a time step. The first row shows the ARM of  $a_R$ , the second row shows the ARM of the critic network, the third row shows the ARM of  $a_F$ . Color intensity indicates the output magnitude. Red triangles denote predators, white circles represent prey, and arrows indicate orientation.

## E.1 FEATURE-LEVEL EXPLANATION

SHAP analyses for the swirling scenario are presented in Figure 26. As shown in Figure 26(a), the relative position of the nearest predator has the strongest influence on the angular velocity output  $a_R$ . However, unlike in periodic boundary conditions, the agent’s own positional state also has a substantial impact on  $a_R$ . This is expected, as in confined environments, proximity to the boundary affects agent behavior—agents at different distances from the wall exhibit different actions.

Figure 26(b) shows the SHAP values of observation features for  $a_F$ . All observation features yield very low SHAP values, including the relative position of the nearest predator, which has the highest (albeit still small) influence. This indicates that the magnitude of  $a_F$  is weakly correlated with the observation, consistent with findings under periodic boundary conditions.

In summary, the most influential observation feature for  $a_R$  remains the relative position of the nearest predator, while  $a_F$  is largely unaffected by observation features—consistent across both boundary conditions.

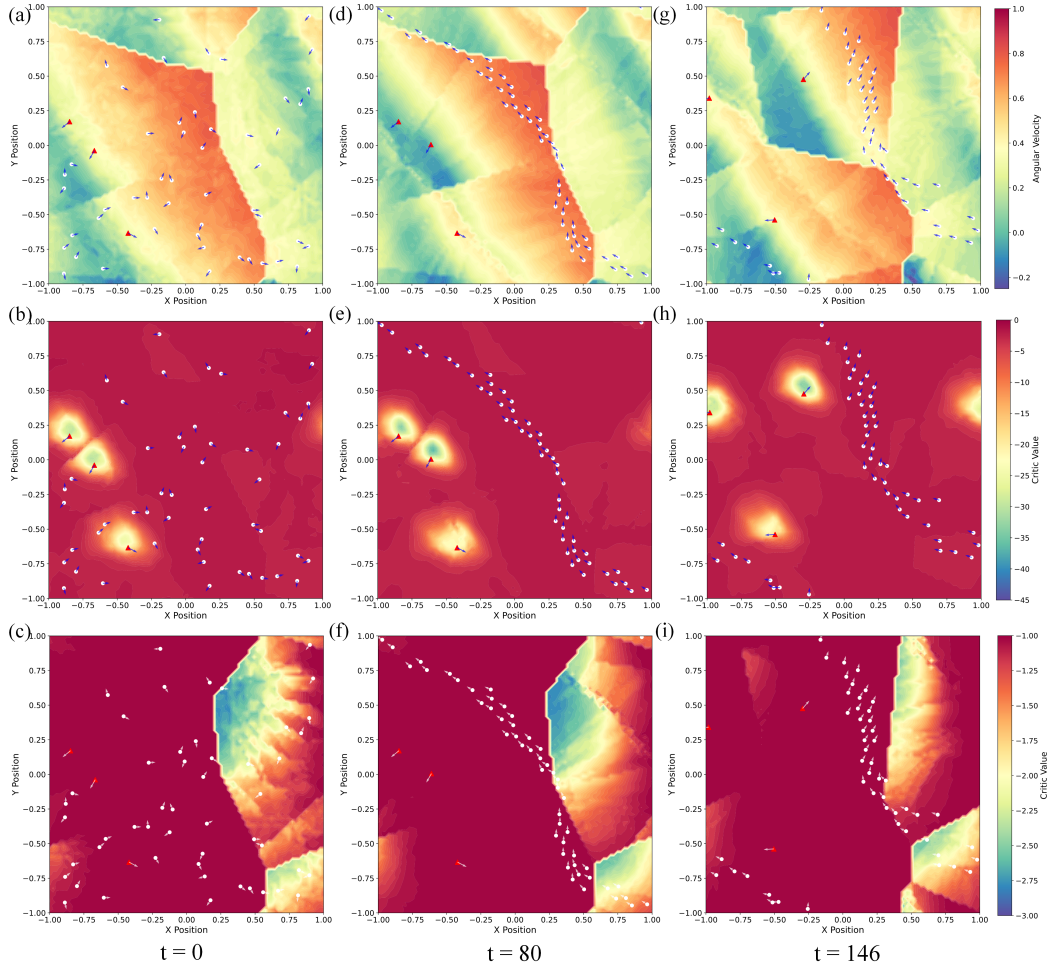


Figure 16: ARM for herding behavior with 80 sampling resolution: Each column corresponds to a time step. The first row shows the ARM of  $a_R$ , the second row shows the ARM of the critic network, the third row shows the ARM of  $a_F$ . Color intensity indicates the output magnitude. Red triangles denote predators, white circles represent prey, and arrows indicate orientation.

## E.2 POLICY-LEVEL EXPLANATION

### E.2.1 AGENT RESPONSE MAP IN THE ABSENCE OF PREDATORS

The Agent Response Map (ARM) for the swirling behavior without predators is shown in Figure 27. Initially, prey are randomly distributed across the environment with uncoordinated orientations. However, they quickly organize into a swirling formation characterized by orientation alignment and reduced relative distances. Notably, their acceleration  $a_F$  remains at or near its maximum value across all spatial locations, as illustrated in Figure 27(c) and (f). The following analysis is structured around three key questions: (1) Why does swirling emerge? (2) Why do prey align their orientations? (3) Why does the relative distance between prey decrease?

**Swirling behavior:** At  $t = 138$ , the critic value map in Figure 27(e) reveals a high-value region near the center of the environment and low-value regions near the boundaries. This pattern aligns with intuitive expectations: since collisions with the boundary incur penalties, prey are incentivized to move away from the edges and toward the center. As they do so collectively, their motion naturally results in a swirling pattern around the central region.

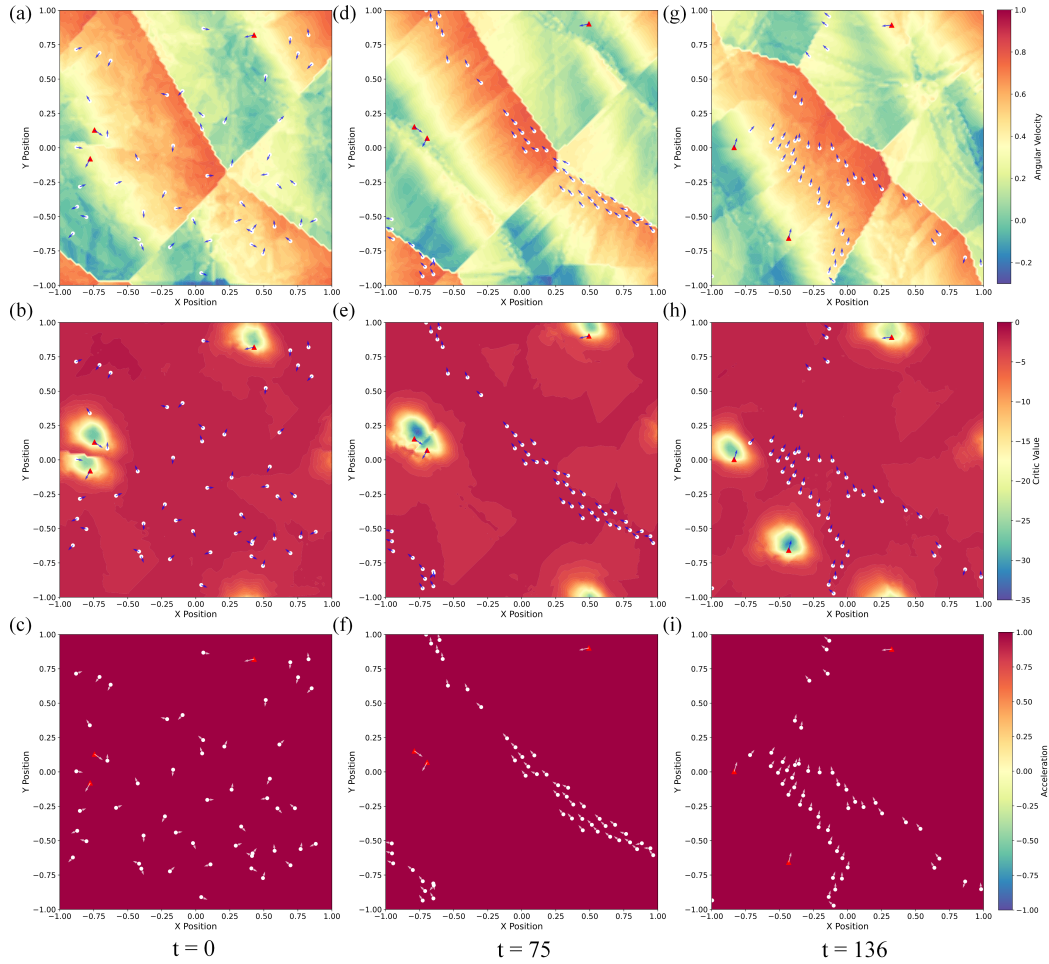


Figure 17: ARM for herding behavior under different random seeds: Each column corresponds to a time step. The first row shows the ARM of  $a_R$ , the second row shows the ARM of the critic network, the third row shows the ARM of  $a_F$ . Color intensity indicates the output magnitude. Red triangles denote predators, white circles represent prey, and arrows indicate orientation.

**Orientation alignment:** The ARM visualizations of  $a_R$  at different time steps, shown in Figure 27(a) and (d), reveal that  $a_R$  varies with prey distribution. Unlike in periodic boundary conditions, where ARM maps tend to exhibit more spatial invariance, the confined environment induces position-dependent coordination. Despite these variations, a consistent pattern emerges: in the vicinity of each prey, the surrounding spatial region displays a relatively uniform  $a_R$  value. This indicates that nearby prey tend to exhibit similar angular velocities, which results in locally consistent motion directions and thus orientation alignment.

**Distance reduction:** For prey that are initially farther apart, differences in orientation create velocity differences along the x and y axes, leading them to gradually approach one another. A more detailed illustration can be seen in the region of the white box in Figure 27(d). In this region, prey on the inner side of the swirl exhibit lower  $a_R$ , while prey on the outer side show higher  $a_R$  values. This means that outer prey orient slightly inward, creating an inward-directed velocity component that facilitates aggregation.

Importantly, this velocity difference, and hence the gathering effect, occurs even without the presence of predators. This is primarily driven by the boundary conditions: prey near the boundaries experience



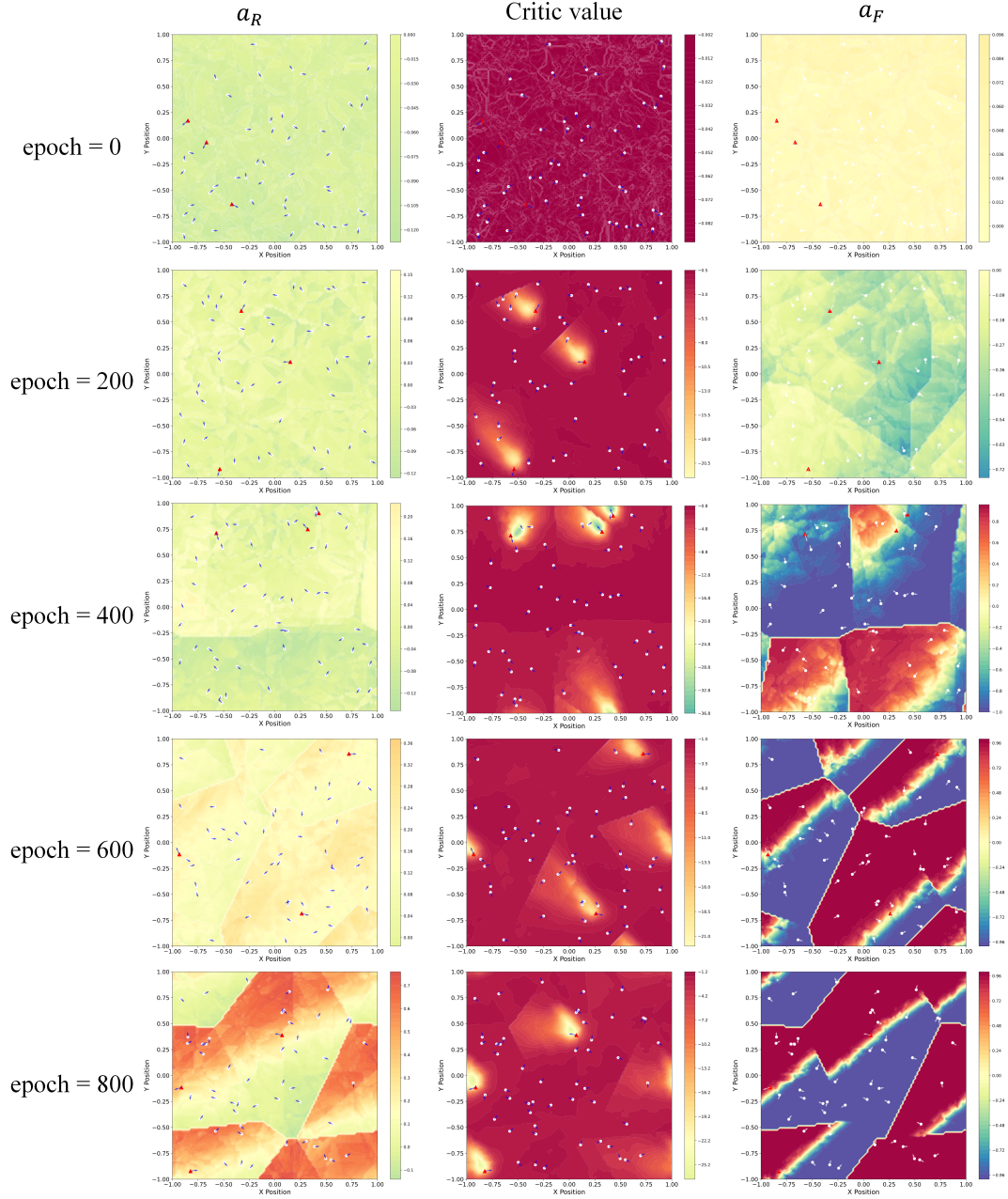


Figure 18: Evolution of ARM throughout Training Process: Each column corresponds to a training epoch. The first column shows the ARM of  $a_R$ , the second column shows the ARM of the critic network, The third column shows the ARM of  $a_F$ . (Part 1)

lower critic values and thus seek to move away from the boundaries, generating an inward bias in their motion.

## E.2.2 AGENT RESPONSE MAP IN THE PRESENCE OF PREDATORS

**Swirling behavior:** In the presence of predators, the prey's critic value and  $a_R$  exhibit notable differences from the case without predators. Figure 28(a), (d) shows the distribution of  $a_R$ , while Figure 28(b), (e) presents the distribution of the critic value. (c), (f) shows the distribution of  $a_F$ .

918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971

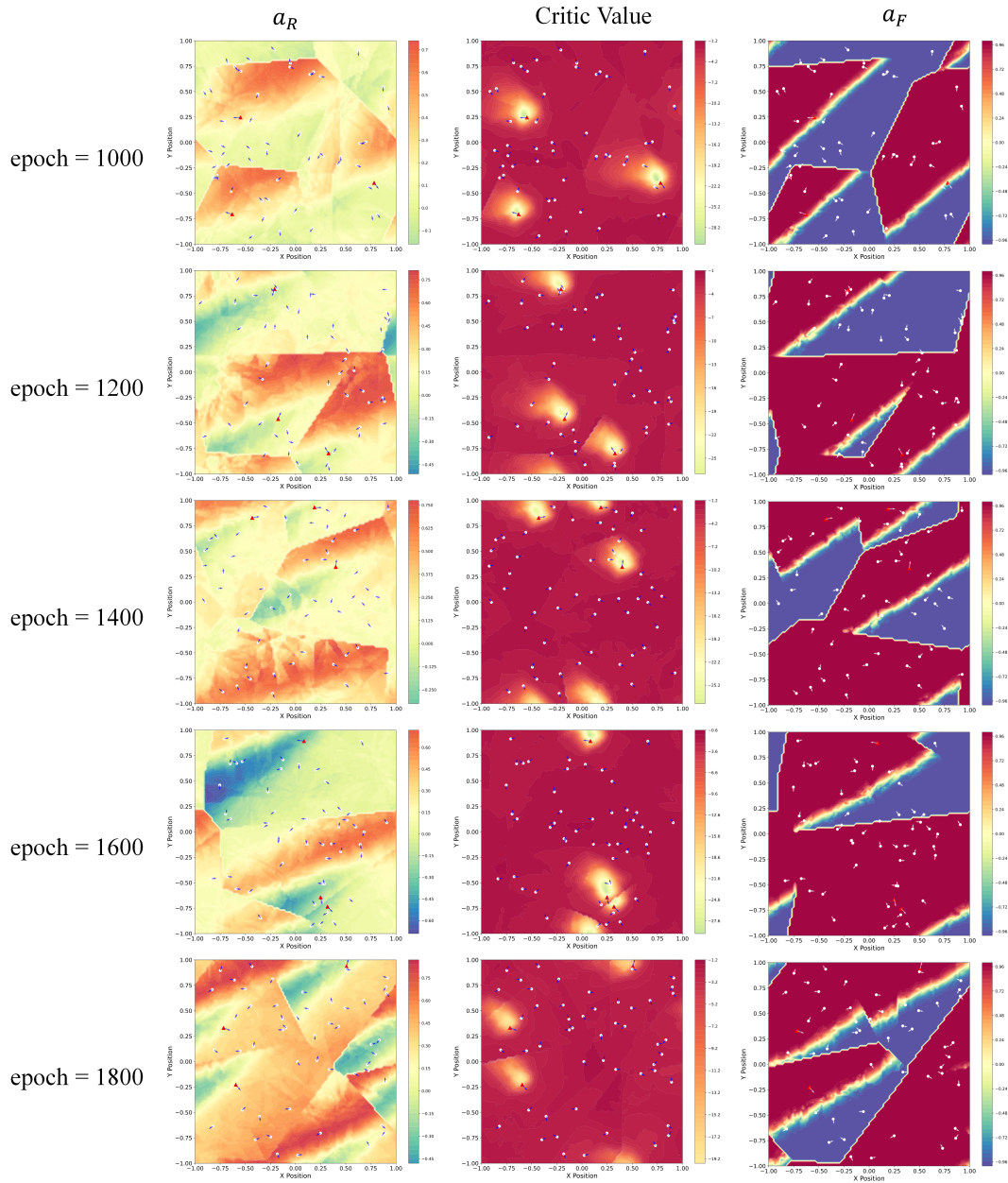


Figure 19: Evolution of ARM throughout Training Process: Each column corresponds to a training epoch. The first column shows the ARM of  $a_R$ , the second column shows the ARM of the critic network, The third column shows the ARM of  $a_F$ . (Part 2, continued)

Analysis of Figure 28(b) reveals the following pattern: the critic value is lowest when prey are close to the predator. As the distance from the predator increases, the critic value gradually rises. Beyond a certain distance, the influence of the predator becomes weaker compared to the effect of the boundary, causing the critic value to peak and then decrease toward the boundary, where a local minimum is observed.

As a result, prey tend to move along the ring where the critic value is highest. By doing so, they maintain an optimal distance from both the predator and the boundary, minimizing risk. This leads to the emergence of swirling behavior.

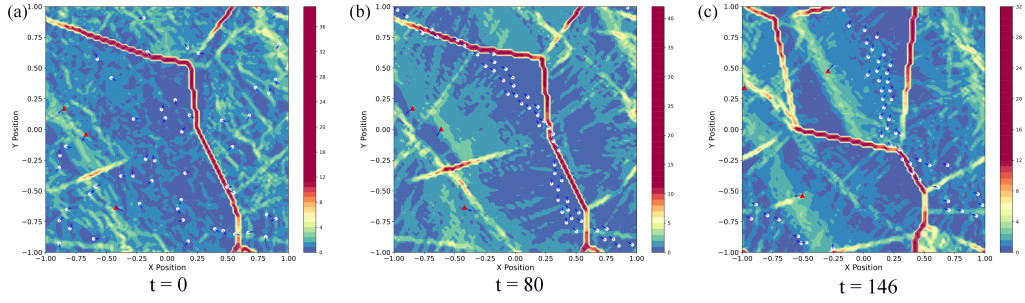


Figure 20: The Spatial Gradient Magnitude of ARM for seed in main text. Each column corresponds to a time step.

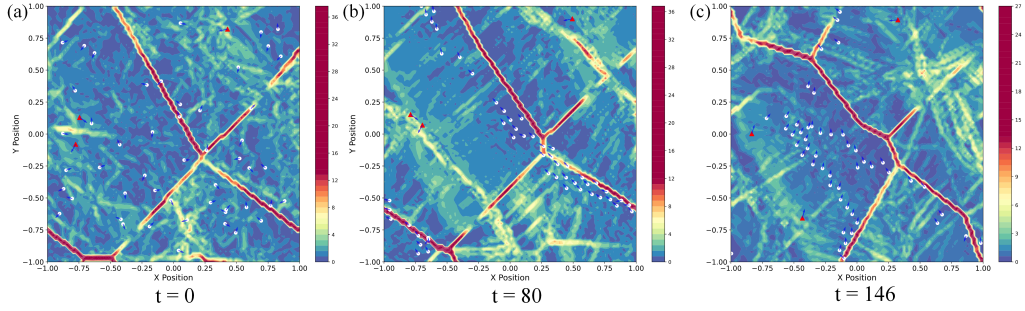


Figure 21: The Spatial Gradient Magnitude of ARM for different seed. Each column corresponds to a time step.

**Orientation alignment:** As shown in Figure 28(a) at  $t = 0$  and (d)  $t = 529$ , prey that are close to each other exhibit similar values of  $a_R$ , causing neighboring individuals to move in the same direction and thus achieve orientation alignment.

**Distance reduction:** For prey that are initially farther apart, differences in orientation angles create velocity differences along the x and y axes, facilitating their convergence. A detailed example can be seen in Figure 28(d) at  $t = 529$  (white box). In this region, prey on the inner side of the swirl have lower absolute values of  $a_R$ , while prey on the outer side have higher  $a_R$ . This implies that outer prey are oriented more inward, generating an inward-directed velocity component that drives the agents to cluster together.

## F SHAPE ASSEMBLY OF ROBOT SWARMS

### F.1 ENVIRONMENT DESCRIPTION

**Region Description.** The target region has a connected shape and is discretized into a grid composed of  $n_{\text{cell}}$  cells. The side length of each cell is denoted as  $l_{\text{cell}}$ , and the center of each cell represents its position.

**Robot Description.** The robot set is denoted as  $\mathcal{A} = \{1, \dots, n_{\text{robot}}\}$ . Each robot is represented as a disk, and its state is  $x = [p^\top, v^\top]^\top$ , where  $p$  and  $v$  are the position and velocity, respectively. The robot's movement is driven by the active and passive forces. The active force,  $f_a$ , is a self-generated force, which is the output of the actor network. The passive force,  $f_b$ , is an elastic force following

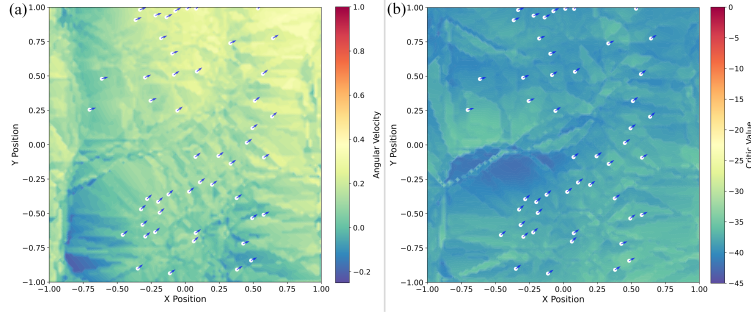


Figure 22: ARM for flocking behavior: (a) ARM of  $a_R$ . (b) ARM of the critic network.

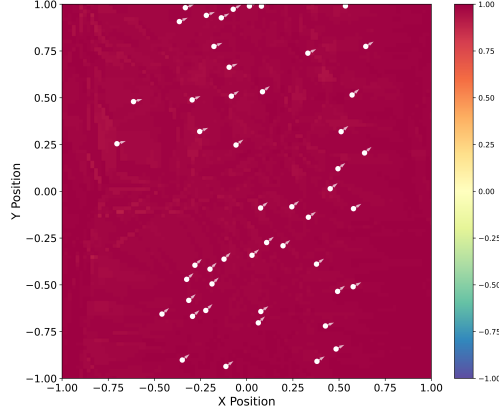


Figure 23: ARM of  $a_F$  under flocking behavior.

Hooke's Law. Thus, the robot's dynamics is

$$\dot{p}_i = v_i, \quad \dot{v}_i = \frac{f_a + f_b}{m_i}, \quad i \in \mathcal{A},$$

where  $m_i$  is the mass of robot  $i$ .

Each robot has a sensing radius,  $r_{\text{sense}}$ , within which it can perceive neighbors and cells. When a robot perceives a neighbor, it can obtain the neighbor's state; when it perceives a cell, it can obtain the cell's position. Each robot also has a collision radius,  $r_{\text{avoid}}$ . A collision occurs if the inter-robot distance is less than  $2r_{\text{avoid}}$ , and a cell is considered occupied by a robot if the robot-to-cell distance is less than  $r_{\text{avoid}}$ .

## B. Action and Observation

*Action.* Each robot's action is a two-dimensional vector with components along the  $x$  and  $y$  axes. This vector indicates the active force  $f_a$  exerted on the robot.

*Observation.* The observation vector consists of four parts: The first part is the robot's own state, the second is the relative state of its neighbors, the third is the relative position of the target cell, and the fourth is the relative positions of unoccupied/observed cells within  $r_{\text{sense}}$ . The maximum number of neighbors and observed cells are denoted as  $n_{\text{nh}}$  and  $n_{\text{hc}}$ , respectively. Hence, each robot's observation is a  $(6 + 4n_{\text{nh}} + 2n_{\text{hc}})$ -dimensional vector. For example, the robot  $i$ 's observation is

$$o_i = [x_i^\top, x_{ji,1}^\top, \dots, x_{ji,n_{\text{nh}}}^\top, p_{ti}^\top, p_{ki,1}^\top, \dots, p_{ki,n_{\text{hc}}}^\top]^\top,$$

where  $x_{ji} = x_j - x_i$ ,  $j \in \mathcal{N}_i$ ,  $p_{ki} = p_k - p_i$ ,  $k \in \mathcal{C}_i$ ,  $\mathcal{N}_i/\mathcal{C}_i$  are the sets of neighbors/observed cells of robot  $i$ , and  $p_t$  is the position of the target cell with  $p_{ti} = p_t - p_i$ .



Note that if the number of neighbors or observed cells within  $r_{\text{sense}}$  is less than  $n_{\text{nh}}$  or  $n_{\text{hc}}$ , respectively, the remaining elements of the observation vector are padded with zeros. If the number of neighbors exceeds  $n_{\text{nh}}$ , only the  $n_{\text{nh}}$  nearest neighbors are considered. If the number of observed cells exceeds  $n_{\text{hc}}$ ,  $n_{\text{hc}}$  cells are selected randomly from the set of observed cells. These adjustments ensure a fixed observation vector dimensionality.

The agents are trained by MADDPG algorithm due to its support for continuous action spaces and its off-policy nature, which ensures high sample efficiency and makes it well-suited for complex tasks.

## F.2 SHAPE ASSEMBLY RESULTS

### F.2.1 TRAINING RESULTS

We train agents for a shape-assembly task in which the target is a five-pointed star. As illustrated in Figure 29, the agents autonomously navigate to the prescribed locations and collectively assemble the desired shape.

### F.2.2 EXPLANATION RESULTS

We analyze the learned policy using SHAP and our Agent Response Map (ARM).<sup>1</sup> The SHAP analysis highlights the features most responsible for driving assembly behaviors (see Figure 30). The ARM further reveals the spatial preferences that guide agent motion (see Figure 31): at early time steps, high-value regions concentrate around the vicinity of the target shape, indicating a tendency to move there. By  $t = 193$ , as interior space becomes occupied, the highest-value regions shift toward the shape’s boundary, reflecting a preference for positions that avoid collisions while still contributing to the assembly.

To evaluate the impact of uncertainty on shape assembly, we injected Gaussian white noise ( $\sigma = 0.02$ ) into the agents’ observation vectors and retrained the policy. The results are presented in Figure 32. Panel (a) visualizes the assembly outcome, while (b) provides a quantitative analysis of the learned geometric structure. Despite increased value vibration caused by noise, internal critic values far exceed external ones. As the task progresses, the internal mean value decreases due to agent occupancy, yet the In-Shape Top 10% value remains significantly higher than that of the Out-Shape region. These findings verify the strong robustness of the shape assembly task against noise.

Furthermore, we extended our validation to complex non-convex and multi-component shapes, specifically the assembly of the letter “B” (Figure 33). Agents successfully transitioned from a random initial distribution to the target formation, demonstrating the method’s applicability to complex scenarios.

The corresponding ARM analysis is shown in Figure 34. As observed in (a) and (b), the unoccupied target center initially exhibits high critic values, drawing agents inward, while occupied regions show reduced values. As the center fills, the value peak shifts to the boundary, promoting the exploration of unoccupied areas—mirroring the intuition in Sun et al. (2023). Quantitative analysis in panel (c) supports this: while the internal mean value drops over time, the In-Shape Top 10% remains consistently higher than the Out-Shape values. This confirms that the Critic network successfully identifies the target interior as the region of highest global value. The stabilization of metrics around  $t = 50$  indicates convergence to a stable geometric attractor, validating the effectiveness of ARM in analyzing complex shape assembly.

**Large Language Model (LLM) Usage** In accordance with the ICLR policy on the use of Large Language Models (LLMs), we disclose that no LLMs were used in this work for research ideation, experimentation, analysis, or writing.

---

<sup>1</sup>SHAP quantifies the contribution of each observation feature to the policy/critic output, while ARM visualizes how value and action responses vary over spatial contexts.

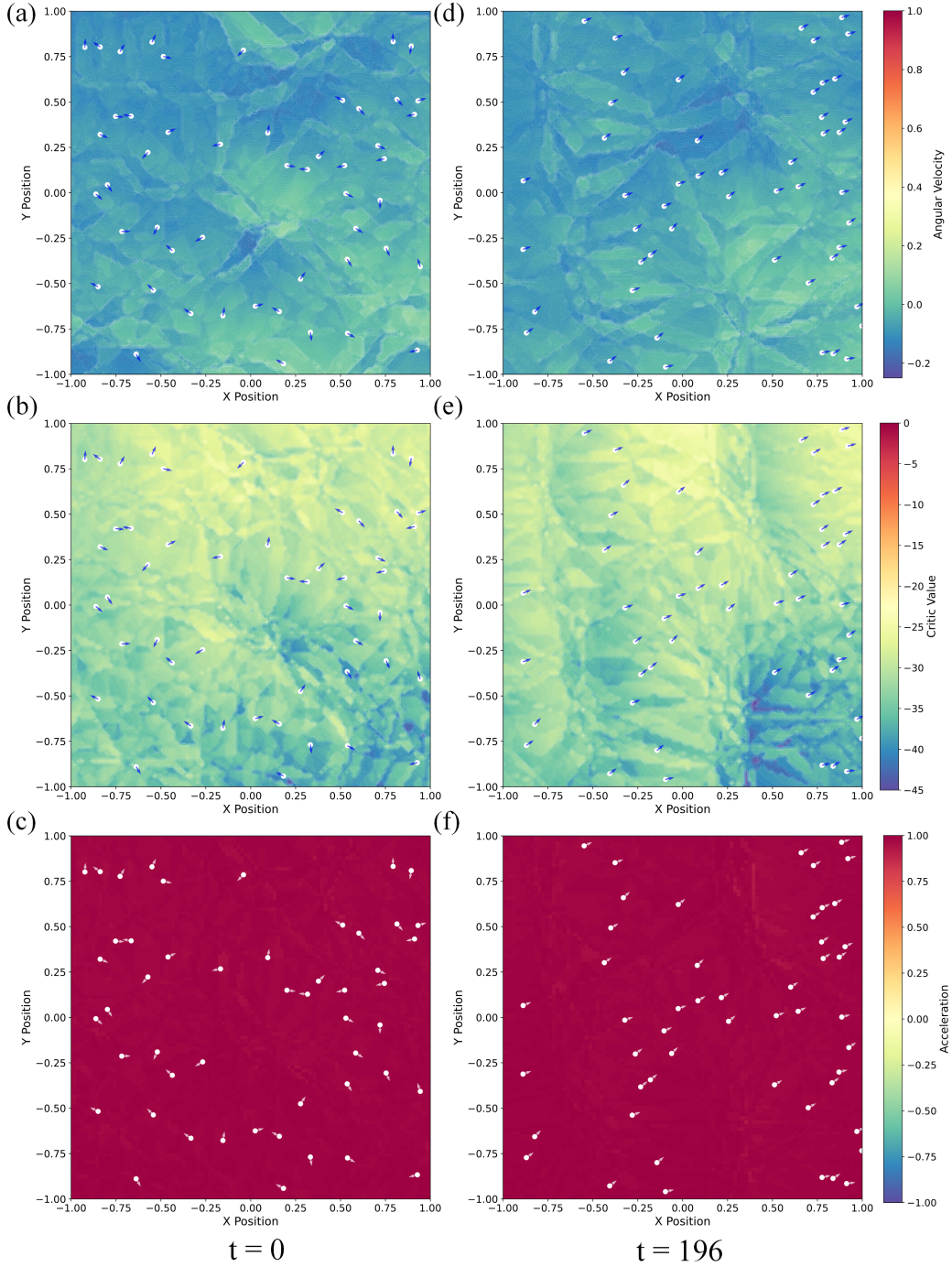


Figure 24: ARM for flocking behavior with different random seeds: Each column corresponds to a time step. The first row shows the ARM of  $a_R$ , the second row shows the ARM of the critic network, the third row shows the ARM of  $a_F$ . Color intensity indicates the output magnitude. White circles represent prey, and arrows indicate orientation.

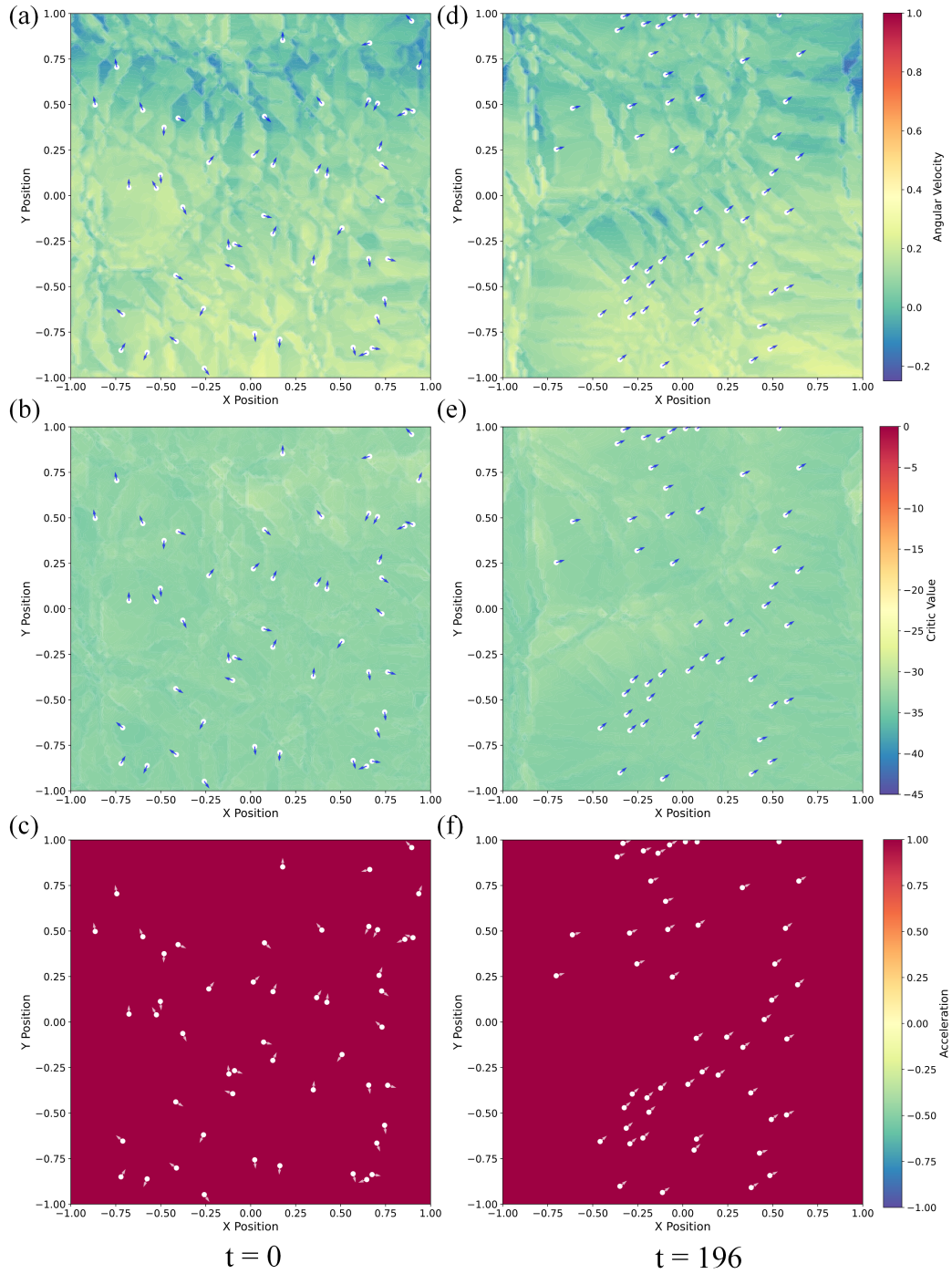


Figure 25: ARM for flocking behavior with virtual orientation  $h = [-1, 0]^T$ : Each column corresponds to a time step. The first row shows the ARM of  $a_R$ , the second row shows the ARM of the critic network, the third row shows the ARM of  $a_F$ . Color intensity indicates the output magnitude. White circles represent prey, and arrows indicate orientation.



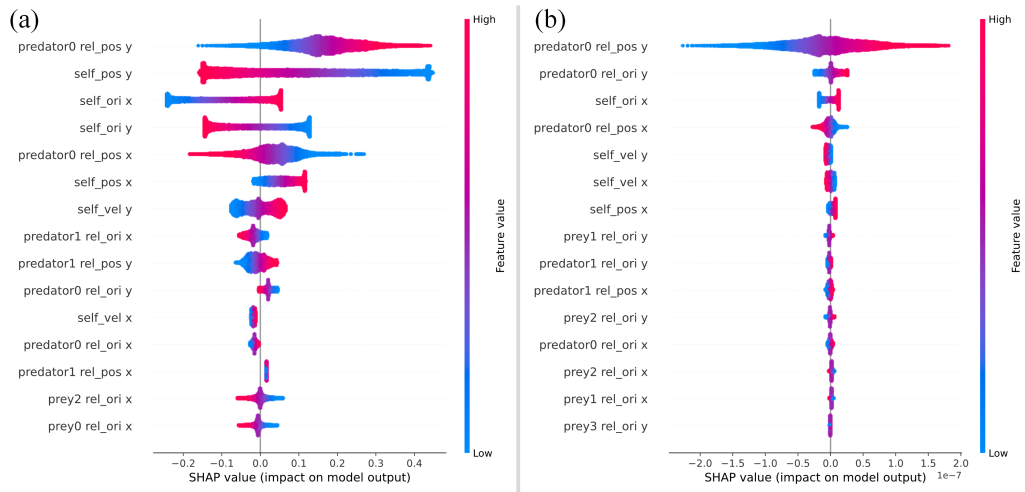


Figure 26: SHAP analysis under the confined environment: (a) features influencing angular velocity  $a_R$ ; (b) features influencing acceleration  $a_F$ .

1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349

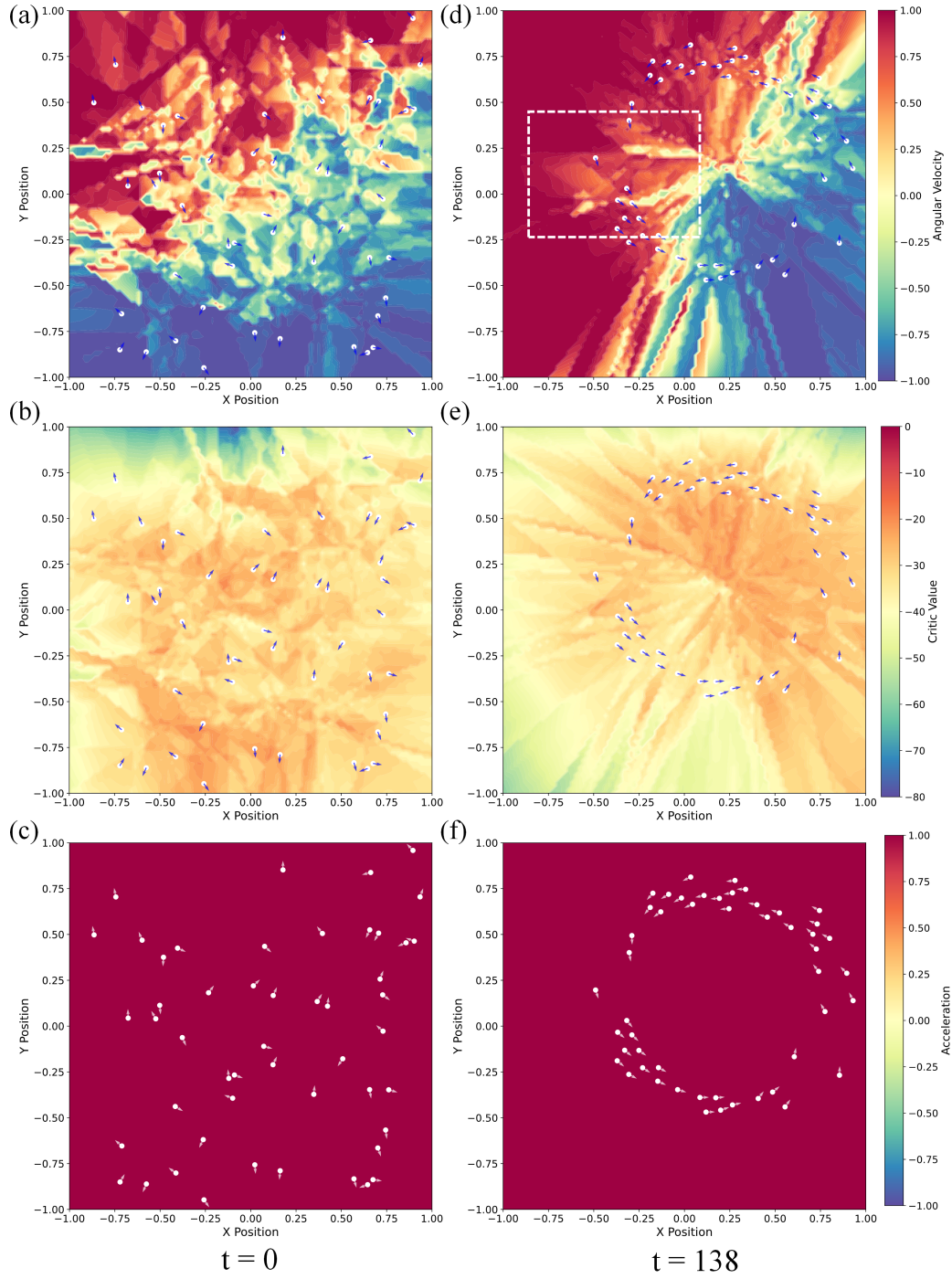


Figure 27: ARM for swirling behavior without predators: Each column corresponds to a time step. The first row shows the ARM of  $a_R$ , the second row shows the ARM of the critic network, the third row shows the ARM of  $a_F$ . Color intensity indicates the output magnitude. White circles represent prey, and arrows indicate orientation.

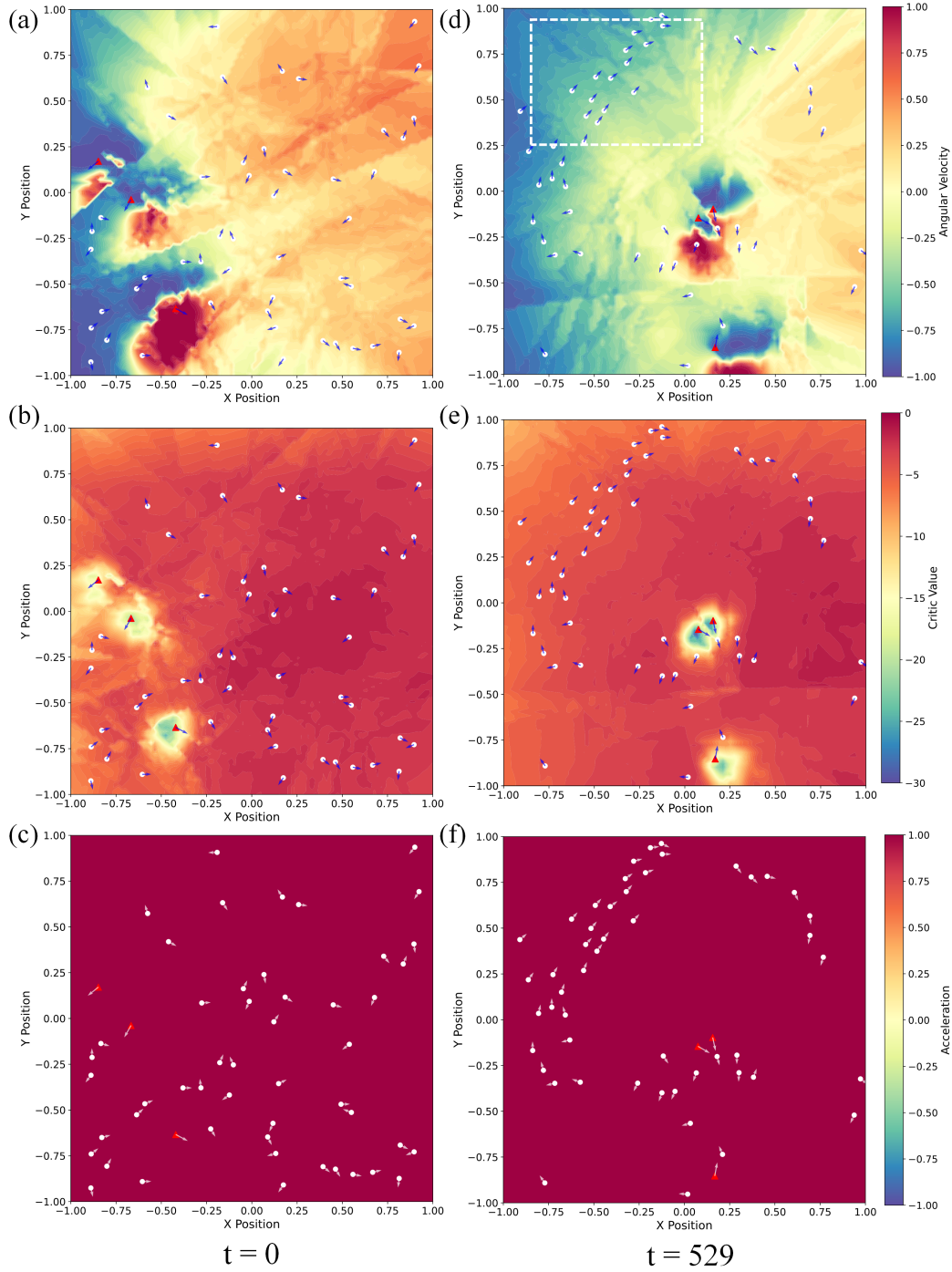


Figure 28: ARM for swirling behavior with predators: Each column corresponds to a time step. The first row shows the ARM of  $a_R$ , the second row shows the ARM of the critic network, the third row shows the ARM of  $a_F$ . Color intensity indicates the output magnitude. White circles represent prey, and arrows indicate orientation.

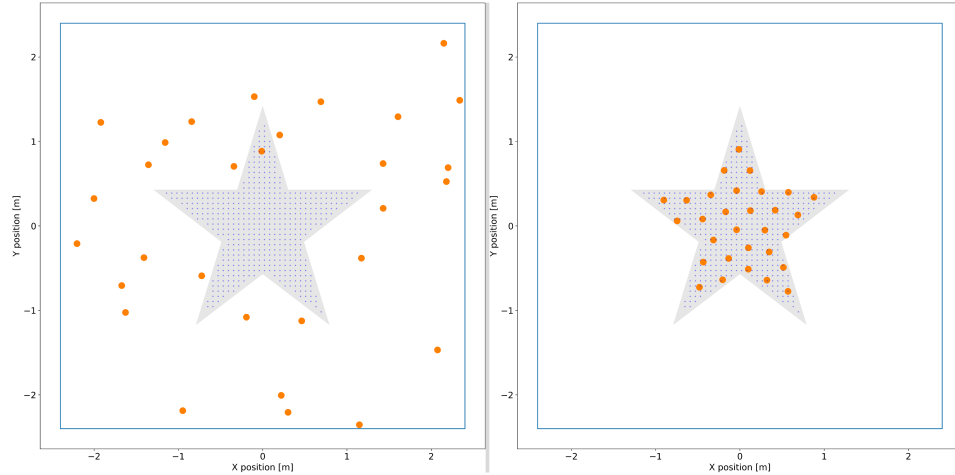


Figure 29: Shape-assembly outcome after RL training. Agents spontaneously move to the designated positions and form the target star shape.

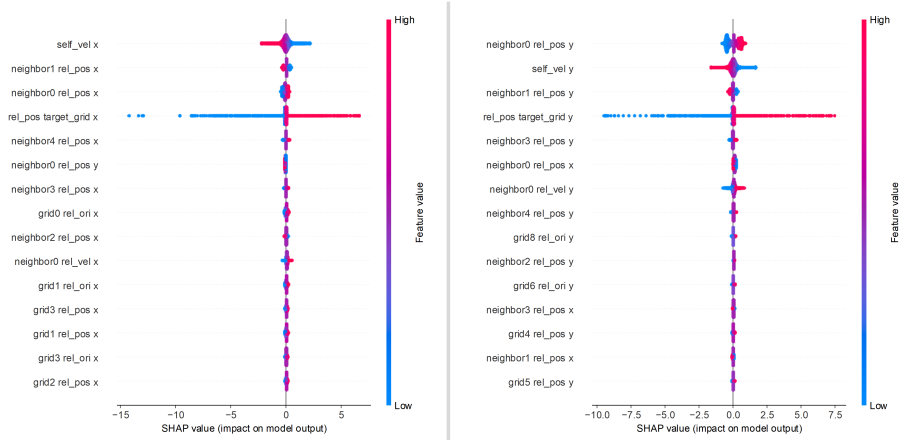


Figure 30: SHAP results for the shape-assembly task, identifying observation features that most strongly influence the learned policy and value estimates.

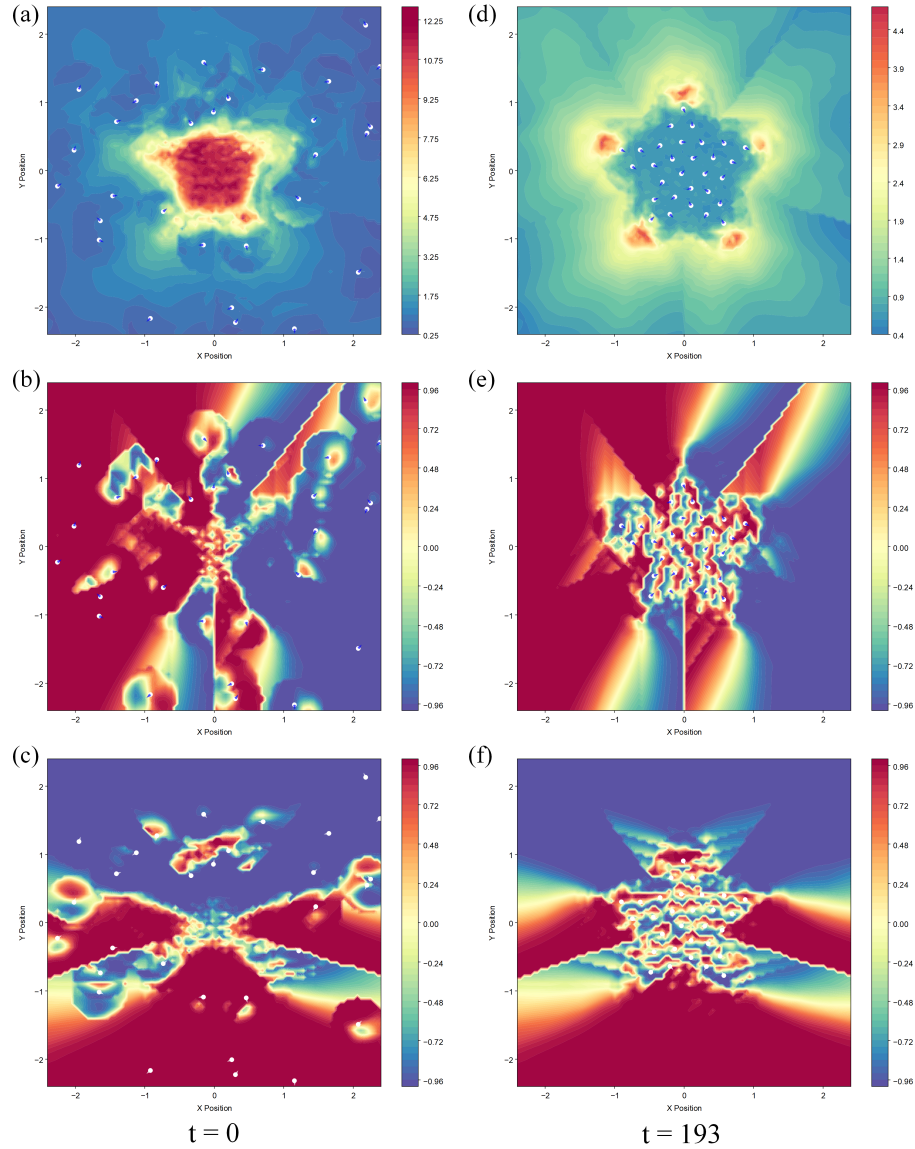


Figure 31: ARM results for the shape-assembly task. Each column corresponds to a time step; the first row shows ARM for the critic, the second for acceleration in  $x$ , and the third for acceleration in  $y$ . Early on, high-value areas lie near the target region, whereas by  $t = 193$  they concentrate near the boundary as interior cells are filled.

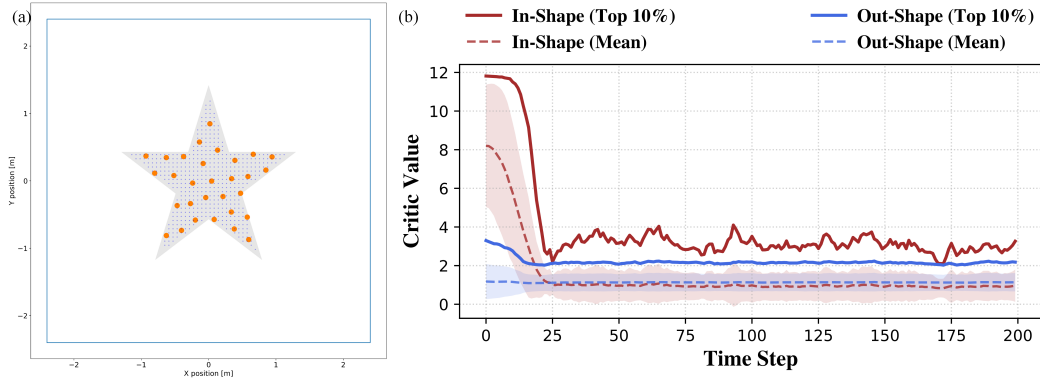


Figure 32: (a) Shape-assembly outcome under the noise interference. (b) Comparison of the overall mean and the top-10% mean of critic values within In-Shape versus Out-Shape regions under the noise interference.

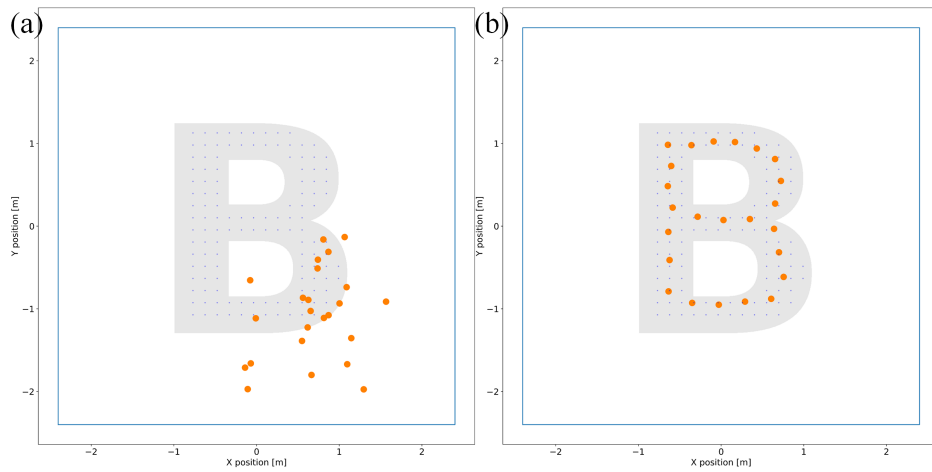


Figure 33: Shape-assembly outcome after RL training. Agents spontaneously move to the designated positions and form the target shape of letter "B".

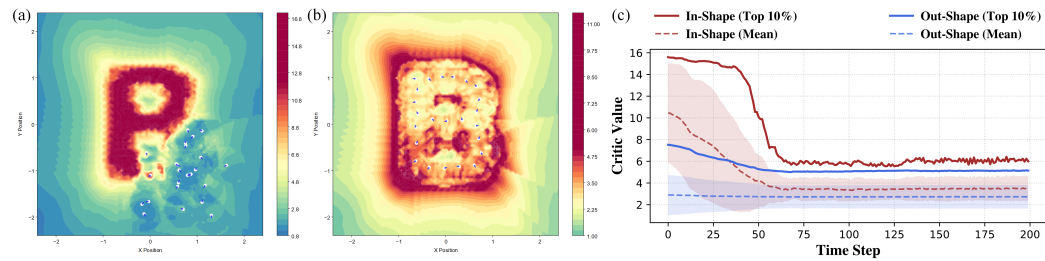


Figure 34: (a)-(b) ARM of critic network at different time steps. (c) Comparison of the overall mean and the top-10% mean of critic values within In-Shape versus Out-Shape regions.