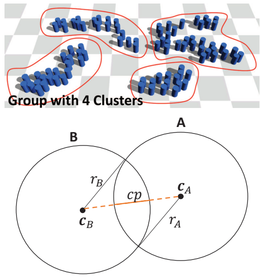
7. Definitions

We give here some definitions on various measurements we use to evaluate group behavior. **Cohesion** is defined as *"the act or state of cohering, uniting or sticking together"*. In physics, cohesion is defined as *"the molecular force between particles within a body or substance that acts to unite them"*. In this work, a cohesive group is a group of agents that are able to stay connected with each other despite any inside or outside influences. To measure cohesion, we identify clusters of agents in the simulation; at any given moment of time, cluster are constructed by identifying agents that move together similarly to [ZTW12] to which we add a distance threshold and consider agents outside the group as outside any cluster. We define **density** as the number of characters present in a given region at any given moment in time divided by the area of the region. Since groups in the experiments can split into clusters, we do not measure global density; rather we measure density on a local circular region of $2m$ radius around each agent to get an indication of how each agent perceives density. **Collision penetration** is defined as the overlapping distance cp_{AB} between two colliding agents (i.e., $cp = r_A + r_B - |\mathbf{c}_A - \mathbf{c}_B|$). **Leader attraction** is the distribution of agents near the leader at the end of the simulation; high numbers of agents in the same cluster as the leader indicate high leader attraction. Finally, **goal completion** measures how much closer to their targets the agents are at the end of simulation; this acts as an indication of agent performance in achieving goals (e.g., goal completion of 50% means that agents are halfway to their targets).



8. Sensitivity Analysis

8.1. Number of Connection Neighbors

Here we are interested to study the effect of the number of connection neighbors n^c to the group behavior. In Figure 16, we see the results for the SimSpeed scenario. As expected, increasing n^c, increases the cohesion of the group; i.e., agents tend to cluster into smaller numbers of progressively larger clusters (color indicates the number of agents in a cluster and the y-axis the number of clusters) such as the ones shown in e.g., Figure 16(a-d). This happens because agents start with different desired velocities and therefore when n^c is small, agents having similar speeds are clustered together splitting the group into small subgroups that move with different speeds. Obviously, this also affects both the density and velocity distributions as perceived by the agents. Density increases progressively since more and more agents are on the inside of bigger and bigger clusters and not on the borders Additionally, velocity starts dropping below the average requested velocity of $1m/s$ as $n^c > 10$ whereas the overall performance is high for $n^c \in [2-5]$. Finally, velocity distribution has lower variance the higher the n^c; i.e., the bigger the mass of agents in a cluster the harder it gets to move all of them together satisfying all required constraints.

The effect of clustering is similar in the SimCFlow scenario; here

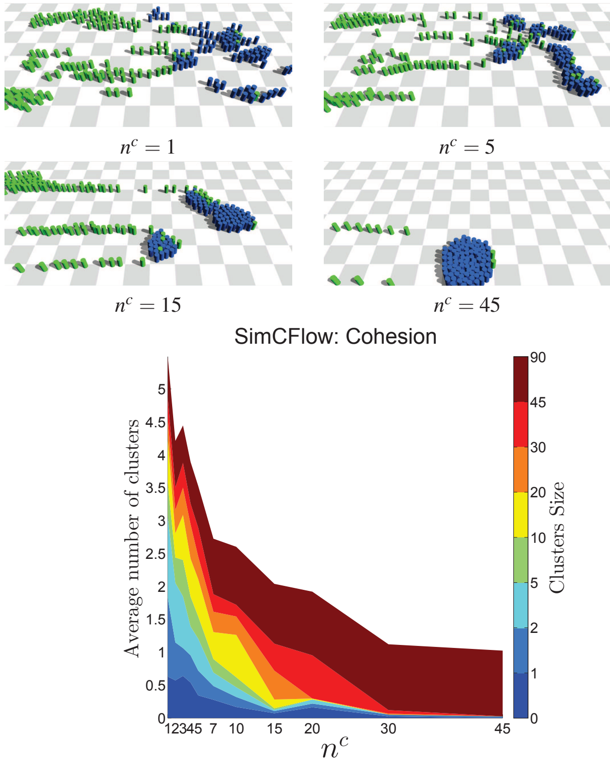
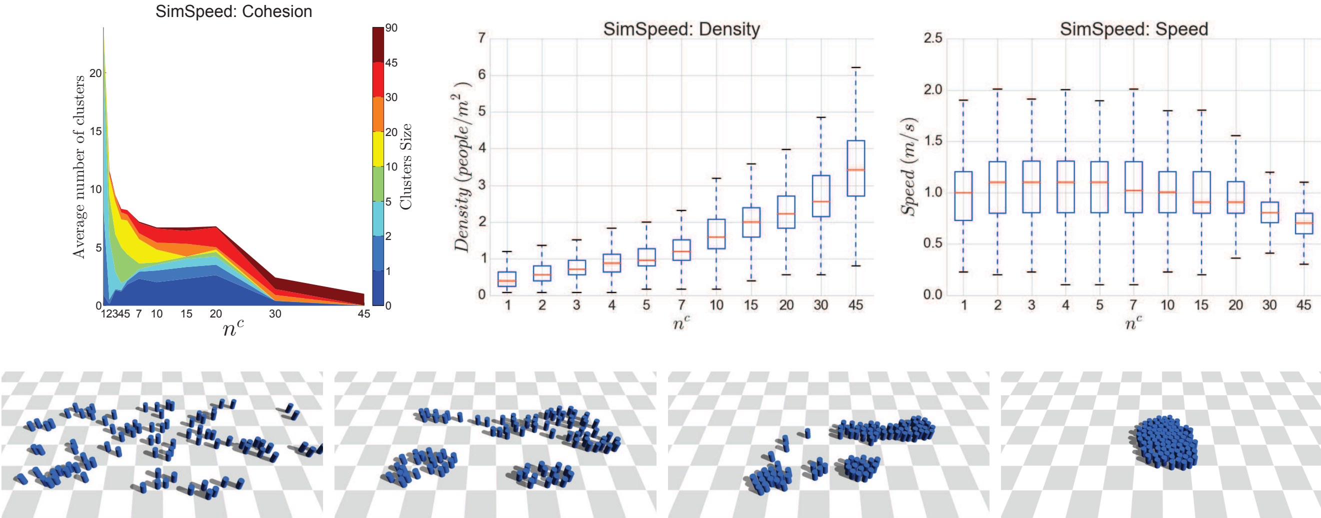


Figure 17: **SimCFlow: Connections** Increasing the number of connections increases cohesion forcing clusters to stay together when interacting with a flow of agents.

though the overall number of clusters is lower than the SimSpeed scenario since agents are initialized with the same preferred velocities and large dense clusters form early (Figure 17). We note that having $n^c \geq 15$ forces the group to break having at least one big cluster (> 45 agents); values over 30 enforce most of the times a very large rigid group.

The proposed model aims in balancing between (a) moving towards goals, (b) staying with nearby group members and (c) collision avoidance (Equation 6). Because of this, we expect that increased cohesion in a group might have a negative impact on collision avoidance and goal completion. To evaluate this impact, the number of collisions and goal completion were measured. The left column of Figure 18 shows the goal completion at the end of the simulation (i.e., how much closer to their goals agents are) for scenarios SimSpeed and SimCFlow and the collision performance for both respectively; the y-axis shows the average distance traversed by alls agents towards their goals divided by the initial distance to their goal. As expected, increasing n^c reduces the goal achieving behaviors of the agents even for the case where the agents are initialized with the same preferred velocity; this can be explained by the fact that groups get slower and denser (Figure 16) leading to less available acceptable velocities.

Additionally, as expected the denser more cohesive the group, the more collisions occur since agents stay together but in most of the cases collision penetration is very small ($< 5cm$); i.e., it is



(a) $n = 1$ (b) $n = 4$ (c) $n = 10$ (d) $n = 45$

Figure 16: **SimSpeed: Connections** (top row) Increasing the number of connections n^c increases group cohesion and density and forces agents to move slower. (bottom row) Snapshots of the simulation for different numbers of connections.

to be expected in simulations such as stampedes or people entering a stadium where density is high (Figure 16d). One solution to remove collisions is to increase the radius of agents to take into account their personal space. Finally, for $n^c = 0$, we get collisions for the original RVO implementation (i.e., no groups). Notice that here collision performance is worse than our approach; this is because for performance reasons RVO takes into account only the half plane in front of the agents and therefore a part of acceptable velocities is ignored. Our implementation does not have this issue and therefore has better collision performance than RVO for small clusters.

8.2. Upper bound to closest approach

The upper bound to closest approach r^{max} controls how far away from other group members agents can be while moving. We measure density and speed changes for the SimSpeed scenario; as expected, low values of r^{max} tend to split the group into smaller velocity efficient groups due to varying desired velocities (Figure 19). Increasing r^{max} to $3-4m$ leads to agents of different velocities being grouped together in a more uniform unclustered way decreasing overall the density around each agent. We note here that the average velocity of the group is $1m/s$ which is exactly in the middle of the desired velocities of scenario 1. The effect of decreasing r^{max} compared to increasing n^c on all metrics is not so prominent; reducing r^{max} brings agents closer increasing density slightly, but this comes at the cost of splitting the group into multiple clusters. Increasing n^c on the other hand attracts larger numbers of agents together (as discussed in Appendix 8.1) in much smaller distances.

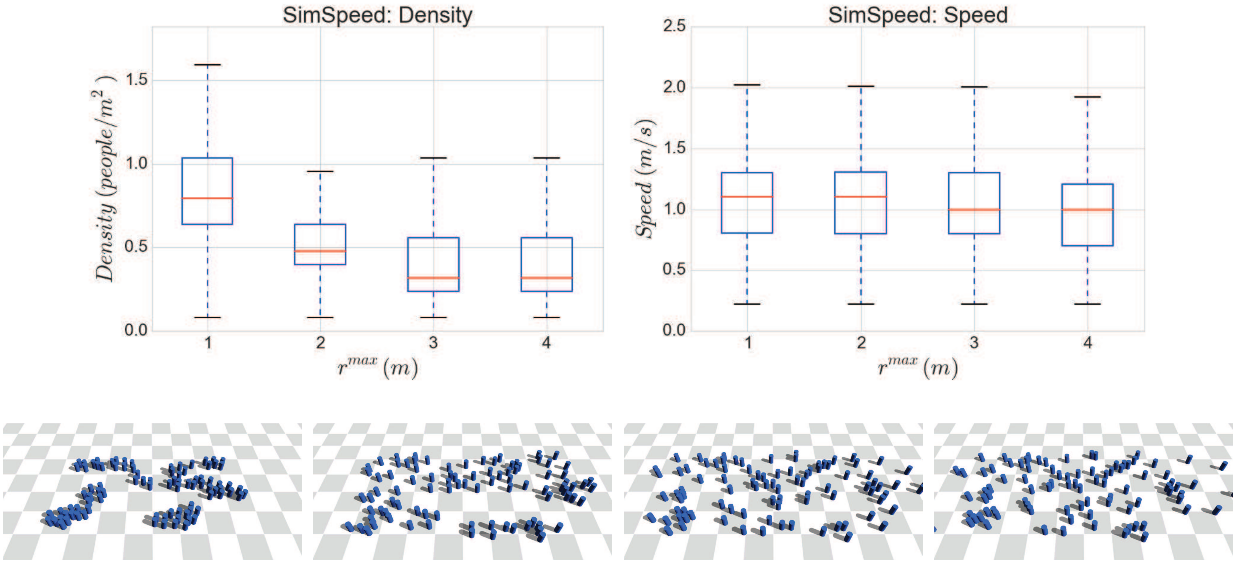
8.3. Weights for group and velocity preference

Our framework has two different weights used in *Equation* 6; w_g and w_v that control importance of group to velocity preference. There is a third implicit weight which is the weight for collision avoidance that is set to 1. We set $w_g, w_v \leq 1$ so that collision avoidance is the most important of the three. Figures 20 and 21 demonstrate the effect of the two weights on the SimSpeed and SimCFlow scenarios. The heatmap shows the average number of clusters, density and speed for different combinations of the two weights. As expected, increasing w_g forces agents to move in clusters reducing both agent centric density and speed whereas increasing w_v forces agents to distribute all over the place while moving faster and increasing density. Additionally, the effects of w_g are more dominant and as long as $w_g \geq w_v$, clustering occurs as indicated by the steepest changes in the gradient of the heatmap and the snapshots shown in Figure 20; e.g., compare the results for $w_g = .5$ and $w_g = 1$ when w_v changes. Finally, setting $w_g = 0$ and $w_v > 0$ forces agents to move as fast as possible to their targets avoiding other agents in the process imitating more aggressive behavior leading to the red bands shown in all three heatmaps of Figure 20.

Setting the appropriate parameters can affect the behavior of the group during outside interference. Take for example the SimCFlow scenario shown in Figure 21; setting both weights of the blue group to 0 makes the agents scatter and move individually, with a lot of them getting carried away by the opposing flow (circled areas). Increasing velocity weight w_v to 20 introduces more aggressiveness to the agents and move faster to the right but some of the agents still get carried away. If in addition we raise the w_g to 20, agents are more aggressive but at the same time cluster together, so it is more difficult to get carried away. If finally we increase w_g to 50, clustering is more dominant and the group has a stronger front.

8.4. Leaders/Followers

Finally, we run some experiments to evaluate the effect of a leader on a group using the SimLeader scenario. Here, a group of agents is initialized without any goal and a leader passes by to collect them.



(a) $r^{max} = 1$ (b) $r^{max} = 2$ (c) $r^{max} = 3$ (d) $r^{max} = 4$

Figure 19: r^{max} **sensitivity** Increasing the upper bound for closest approach, the group spreads more uniformly leading into less density per agent (less discomfort); low values lead to clustering. Additionally, there is a slight decrease in average speed since more agents move together. (a-d) Increasing r^{max} leads in more cohesive groups with uniform density.

We vary the parameters of the simulation to find appropriate parameters to carry the entire group. It is important to say that only the leader has a goal; all other agents have as only purpose to stay with the other members of the group . In the top row of Figure 22 we see the effect of n^c on leader attraction (violin diagram) and the effects of n^c, r^{max} and w_g, w_v on the average speed of the group. Notice that the optimal values for leader attraction and speed are for $n^c \in [3-7]$ indicating that a smaller number of connections is enough to move a group; agents follow their closest friends and not the whole group similar to what happens in real life groups (think of a parade). The leader starts to fail capturing the crowd for $n^c > 10$ where the group becomes more rigid and difficult to move; the group following the leader becomes progressively smaller and smaller (Figure 22(a-f)). Notice the correlation between speed and leader attraction for n^c and that we never get to the leaders speed ($0.6m/s$) since the group is immobile until the leader reaches it.

Additionally, setting small values for r^{max} is enough; larger values force agents to stay close to the immobile neighbors and not the leader. Similarly to n^c, increasing the weight of the group w_g reduces the desire of characters to move whereas increasing the velocity weight promotes movement. Good values for leader following are $n^c \in [3-7]$, $r^{max} \in [1,2]$ and $w_v \geq 0.3$.

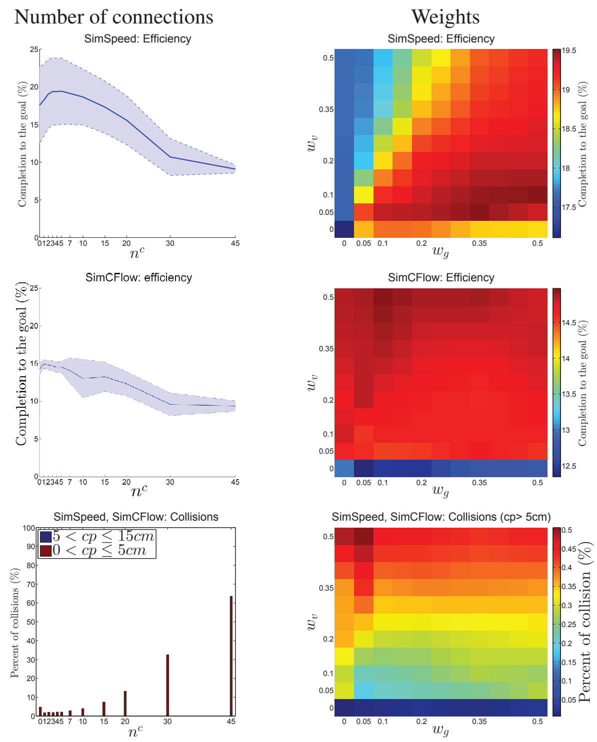


Figure 18: **Goal Completion and Collisions** These graphs show the efficiency of agents in reaching their goals for (top row) different numbers of connections and (bottom row) different weights for group and velocity preference.

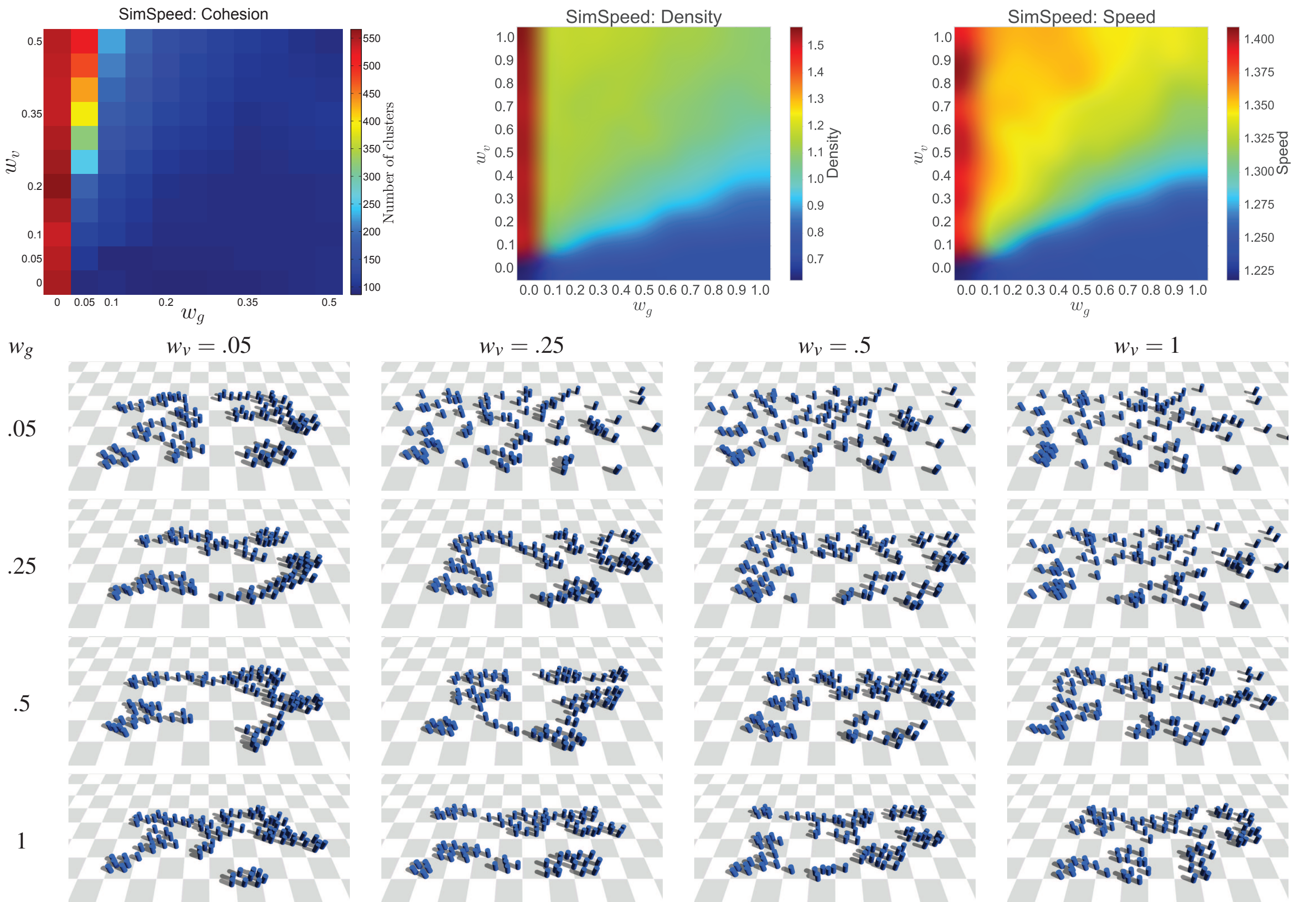


Figure 20: **SimSpeed: Weights** (a-d) Weights on the group or velocity preference mainly influence the structure of the group; more weight on velocity spreads the group whereas more weight on groups leads to more clustered and dense groups.

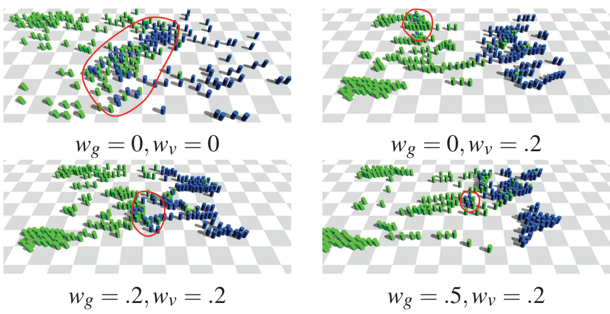
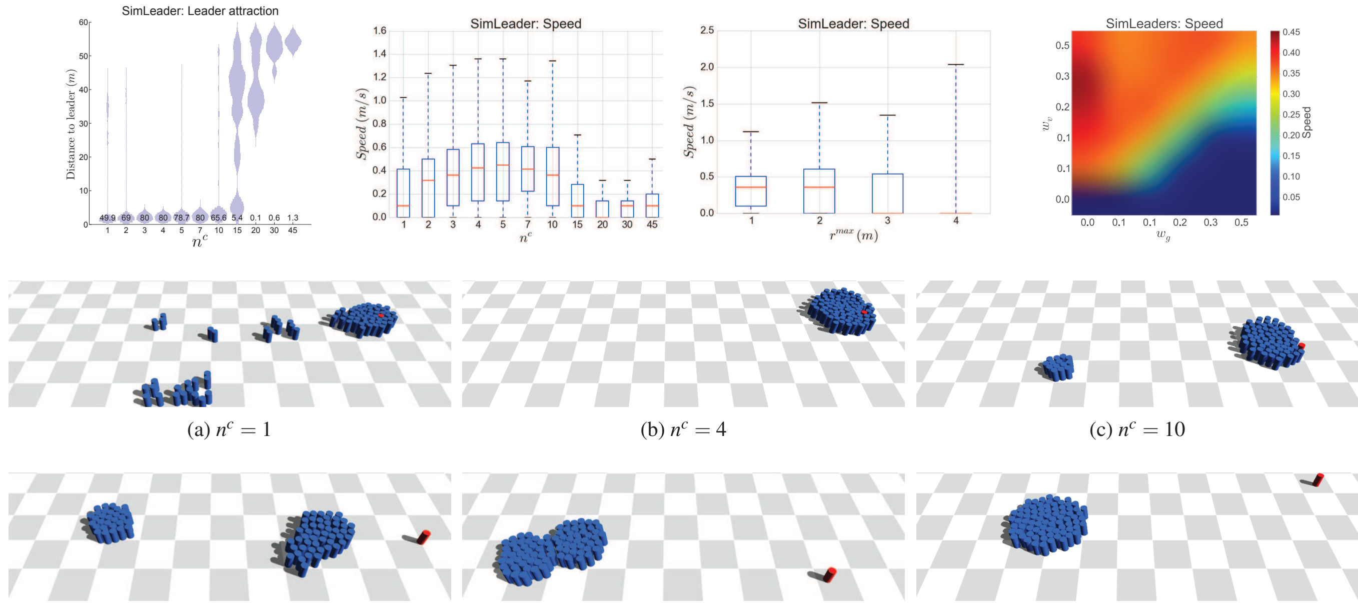


Figure 21: **SimCFlow: Weights** When the group weight is large enough, the group does not break into clusters even in large interactions with other groups.



(a) $n^c = 1$ (b) $n^c = 4$ (c) $n^c = 10$

(d) $n^c = 15$ (e) $n^c = 20$ (f) $n^c = 45$

Figure 22: **SimLeader** Analysis for the leader scenario. (a-f) Resulting groups for different n^c. Increasing the number of connections neighbors makes the standing group more rigid and difficult to move using a single leader.