Behavior:

Currently, our behavior module is based on a mixture of behavior tree and finite state machine, implemented in Python. It reads information from the localization module and gamecontroller module, and plans action command in each main loop. As the localization is not satisfying enough for complicate moving strategies, we choose decentralized decision making. Thus, our robots can use relative positions to make a decision which can avoid the impact of their wrong self-localization.

Last year, our attacking robots all using a same strategy and no matter where other teammates are. As a result, they often collide when two robots walking toward the ball and finally both of them lost the ball. Thus, we design cooperation skills to avoid collision this year.

The most important rule is that one robot will control the ball(which means go to kick the ball) until it kicks or falls to avoid continuous status changing caused by uncertain information. When one robot is controlling the ball(we call it kicker), the other attacker will assist it(we call it assistant). More specifically, it will stand and wait at a suitable point. When the kicker kicks or falls, the assistant will soon go ahead to control the ball and become the kicker.

Also, there is some exception for example, when the assistant is much closer to the ball than the kicker, they will switch their roles. Such dynamic role changing greatly improves offense efficiency.

As mentioned above, we are using relative positions for decision making. We used to decide the kicking direction based on the self-localization result and real goal position. However, as we can accurately see the goal, a better way is deciding the target according to the goal position seen by the robot. It ensures that the robot is kicking to a goal rather than a fixed point.

As the rule changed a lot on RoboCup 2018, we also enhanced the strategies for the added penalty rules.