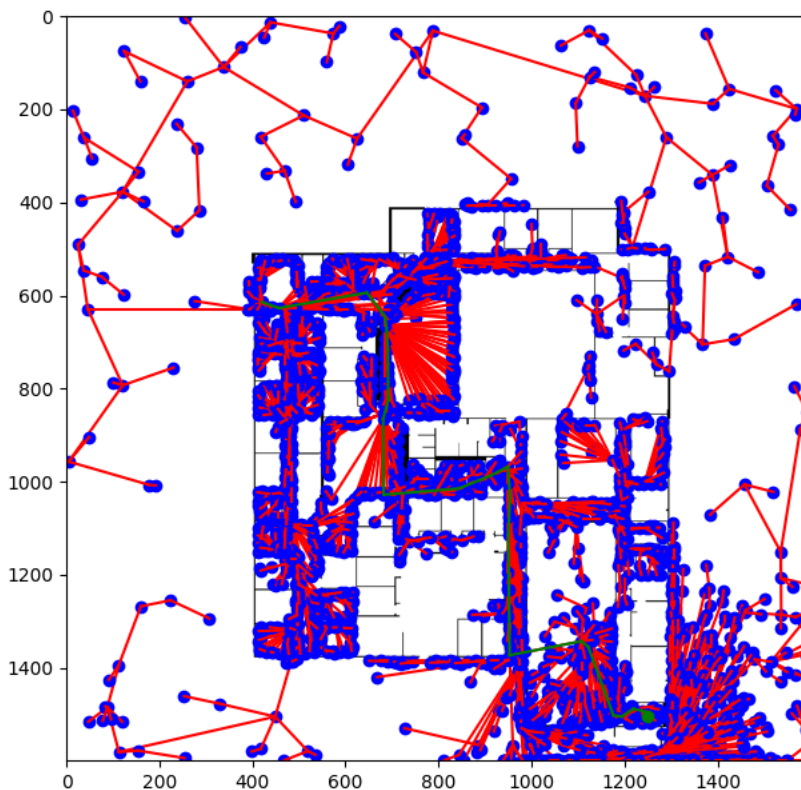


# RRT

RRT is a probabilistic planning algorithm that builds up a tree of feasible paths by sampling nodes at random and connecting them to the nearest neighbor if that neighbor has a feasible path to the new node.

In order to enable the large-scale planning necessary to generate the path seen below, we utilize a few tricks to bias sampling toward promising areas.



First, we bias sampling near walls. To do this, we use sequential dilations to get the areas of the map that are between a distance of 1 times the robot radius and 2.5 times the robot radius away from a wall.

We also bias away from sampling areas we have already sampled and toward areas on the frontier of where we have sampled. We maintain a probability distribution over the promising near-wall pixels and update it every time a new node is added.

This is visible in `update_distance_distribution`

Finally, we bias sampling near the goal.

## RRT\*

This implementation is mostly vanilla. When we add a new node to the tree, we attach it to the feasible neighbor that gives the lowest cost to come instead of the closest one. Then for each neighbor in a ball around the new node, we recursively rewire each of those. We fully recurse so that each of those rewired nodes further generates a ball in which we rewire all nodes. We could make this more efficient by only trying to reconnect to nodes who have had their cost-to-come changed, but we do not do this for simplicity of implementation. We also set the theta value such

that each node points from its parent to itself. This is important for our controller to work properly.

