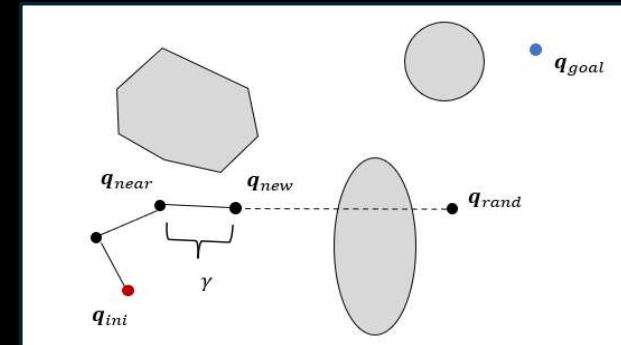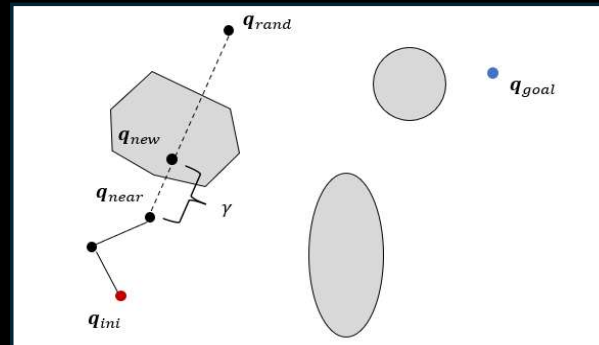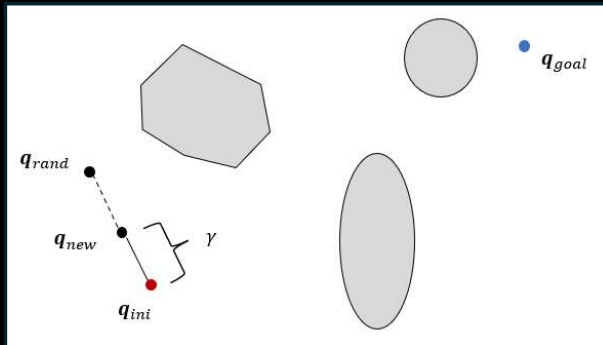# RRT(Rapidly exploring Random Tree)

Ⅰ. Definition

- **<u>Path planning</u>** based unform sampling
- **<u>random point</u>** instead of grid-separation for high dimension
- use **<u>Tree</u>** with Node, Edge

Ⅱ. Way to make Tree

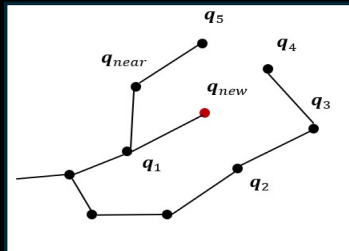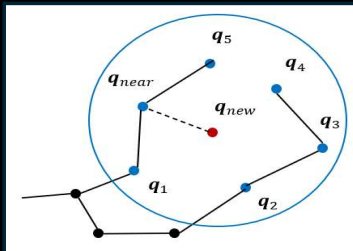# RRT(Rapidly exploring Random Tree)

Ⅲ. Limitations & Alternative Methods

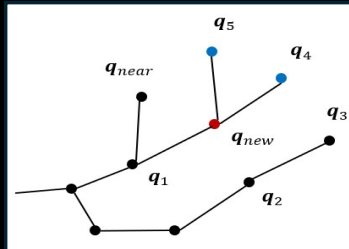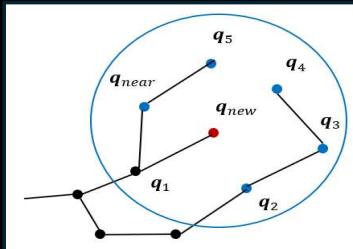- **Not optimal path planning**
  - **RRT***
    - Reparenting
      - connect with parent node to get min cost among points in the q_{new} <u>radius</u>
      - change parent node of q_{new} from q_{near} to q_1

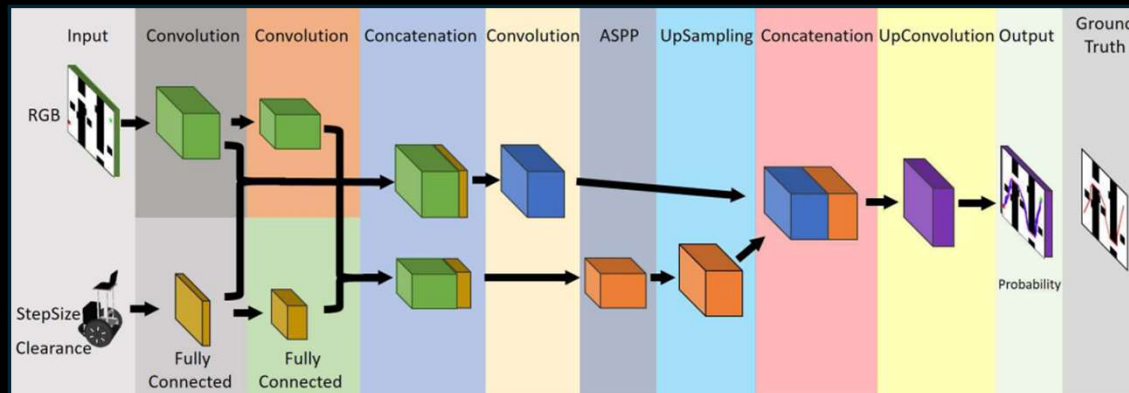

    - Rewiring (restructuring Tree)
      - Nodes in the radius of q_{new} with a new parent node also find the min cost again (q_4 and q_5)

RRT(Rapidly exploring Random Tree)

Ⅲ. Limitations & Alternative Methods

- **RRT*: a lot of time and memory**
  - **NRRT*(Neural RRT*) with CNN**
    - uniform sampling + non-uniform sampling (concentrate samples on specific areas)
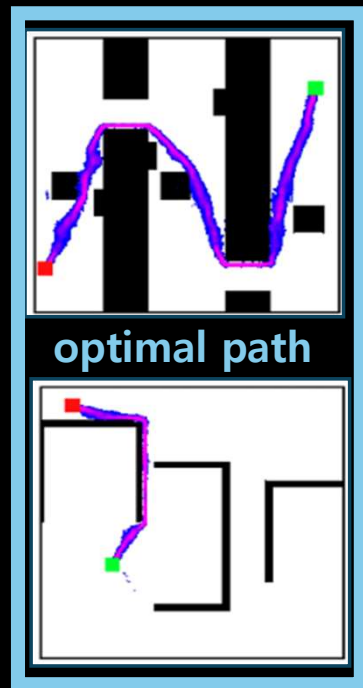
RRT(Rapidly exploring Random Tree)

IV. Implementation of NRRT* (with pseudocode)

```
Input  : x_init, G(x_goal), Map, S, C
Output: T
1 V ← x_init, E ← ∅, T = (V, E);
2 O ← NeuralModel(Map, S, C)
3 for i = 1...N do
4    if Rand() > 0.5 then
5       x_rand ← NonuniformSample(O);
6    else
7       x_rand ← UniformSample();
8    x_nearest ← Nearest(T, x_rand);
9    x_new ← Steer(x_nearest, x_rand);
10   if ObstacleFree(x_nearest, x_new) then
11      T ← Extend(T, x_new);
12      Rewire();
13      if x_new ∈ G(x_goal) then
14         Return(T);

15 Return failure;
```
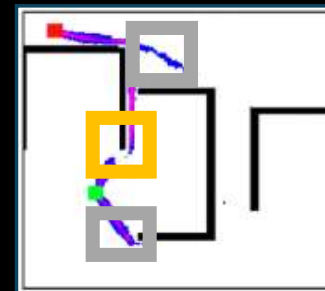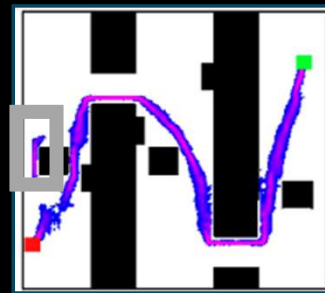
RRT(Rapidly exploring Random Tree)

V. Simulation of NRRT* with two maps and different step sizes
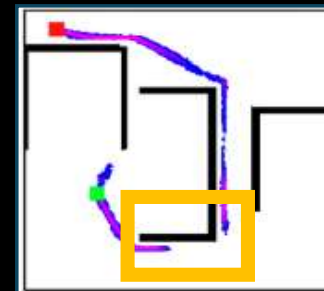- init: red, goal: green, high p to include optimal path: purple, low p: blue

optimal path

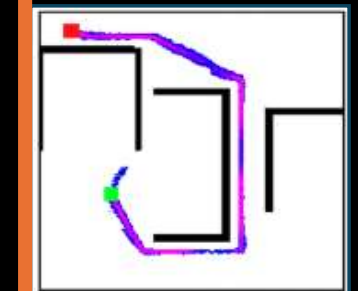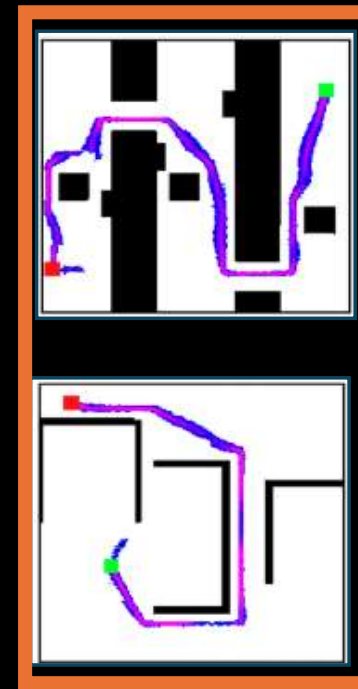incomplete preditions

step size(clearance):   1   2   4   6
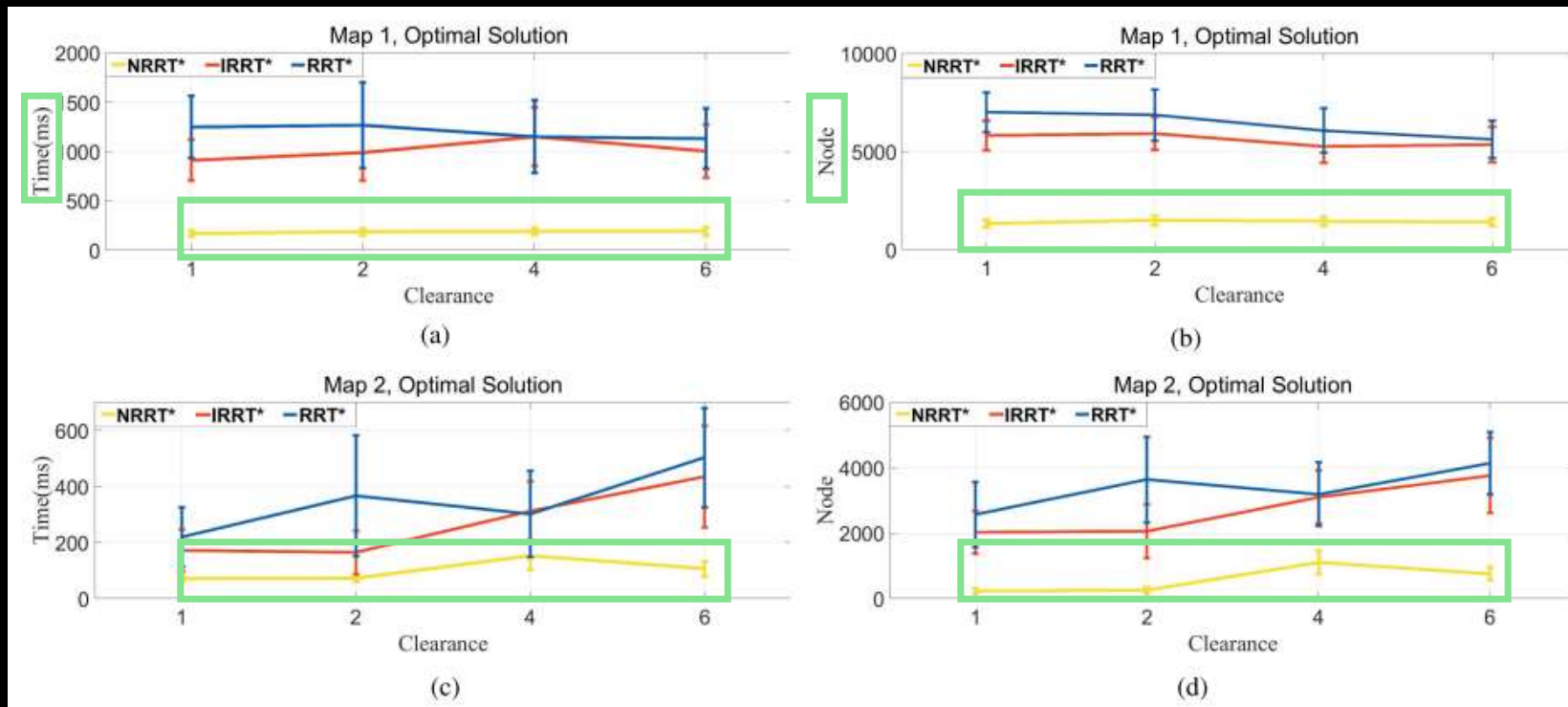
⬜ : include not optimal path   🟧 : include discontinuous path   🟧 : not optimal path

RRT(Rapidly exploring Random Tree)

VI. Performance of NRRT*
- uniform + nonuniform sampling -> no impact on performance

refer to thesis"Neural RRT*: Learning-Based Optimal Path Planning"