

# Generative Adversarial Network Based Heuristics for Sampling-Based Path Planning

Tianyi Zhang, Jiankun Wang, and Max Q.-H. Meng, *Fellow, IEEE*

**Abstract**—Sampling-based path planning is a popular methodology for robot path planning. With a uniform sampling strategy to explore the state space, a feasible path can be found without the complex geometric modeling of the configuration space. However, the quality of the initial solution is not guaranteed, and the convergence speed to the optimal solution is slow. In this paper, we present a novel image-based path planning algorithm to overcome these limitations. Specifically, a generative adversarial network (GAN) is designed to take the environment map (denoted as RGB image) as the input without other preprocessing works. The output is also an RGB image where the promising region (where a feasible path probably exists) is segmented. This promising region is utilized as a heuristic to achieve non-uniform sampling for the path planner. We conduct a number of simulation experiments to validate the effectiveness of the proposed method, and the results demonstrate that our method performs much better in terms of the quality of the initial solution and the convergence speed to the optimal solution. Furthermore, apart from the environments similar to the training set, our method also works well on the environments which are very different from the training set.

**Index Terms**—Generative adversarial network (GAN), optimal path planning, robot path planning, sampling-based path planning.

## I. INTRODUCTION

ROBOT path planning is to determine a collision-free path from a start state to a goal state while optimizing a performance criterion such as distance, time, or energy [1]. Many classic approaches have been proposed to solve the path planning problem in the past decades, such as potential field method [2], cell decomposition method [3], grid-based

Manuscript received January 27, 2021; revised March 30, 2021 and May 17, 2021; accepted June 22, 2021. This work was partially supported by National Key R&D Program of China (2019YFB1312400), Shenzhen Key Laboratory of Robotics Perception and Intelligence (ZDSYS20200810171800001), Hong Kong RGC GRF (14200618), Hong Kong RGC CRF (C4063-18G). Recommended by Associate Editor Zhengcai Cao. (Tianyi Zhang and Jiankun Wang contributed equally to this work. Corresponding author: Max Q.-H. Meng.)

Citation: T. Y. Zhang, J. K. Wang, and M. Q.-H. Meng, “Generative adversarial network based heuristics for sampling-based path planning,” *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 1, pp. 64–74, Jan. 2022.

T. Y. Zhang and J. K. Wang are with the Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: zhangty@mail.sustech.edu.cn; wangjk@sustech.edu.cn).

M. Q.-H. Meng is with the Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen 518055, China, on leave from the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, China, and also with the Shenzhen Research Institute, The Chinese University of Hong Kong, Hong Kong, China (e-mail: max.meng@ieee.org).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JAS.2021.1004275

methods including A\* [4] and D\* [5] algorithms. However, there are some problems existing in these approaches, such as computational difficulty and local optimality.

In recent years, sampling-based algorithms have been shown to work well in practical applications and satisfy theoretical guarantees such as probabilistic completeness. Sampling-based algorithms can avoid discretization and explicit representation of the configuration space, resulting in good scalability to high-dimensional space. Probabilistic roadmap (PRM) [6] and rapidly-exploring random tree (RRT) [7] are two representative methods of sampling-based algorithms to solve path planning problems. RRT algorithms have gained popularity for their efficient single-query mechanism and easy extension to many applications such as kinodynamic path planning [8] and socially compliant path planning [9]. However, there are also some limitations of the RRT algorithms which still remain to be solved. For example, the solutions are usually far from optimal. Although RRT\* [10] is capable of achieving asymptotic optimality by introducing near neighbor search and tree rewiring operations, it takes much time to converge to the optimal solution. The reason is that the RRT employs a random sampling mechanism. On one hand, it guarantees that one feasible solution can be found if it exists as the number of iterations goes to infinity. On the other hand, this mechanism affects the convergence speed because it uniformly searches the whole state space.

To overcome the aforementioned problems, we present a novel learning-based algorithm to leverage previous planning knowledge to generate a promising region where a feasible or optimal path probably exists. This promising region serves to generate a non-uniform sampling distribution to heuristically guide the sampling-based path planners to explore the state space more efficiently. As shown in Fig. 1, compared with RRT\*, our proposed generative adversarial network (GAN) based heuristic RRT\* can quickly find a high-quality initial path with fewer nodes.

Our main contributions include:

- 1) Propose a heuristic method for sampling-based path planning algorithm using GAN;
- 2) Design a GAN model to predict the promising region for non-uniform sampling;
- 3) Apply the GAN-based heuristic method to case studies to demonstrate the effectiveness of the proposed method.

The rest of this paper is organized as follows. We first review the previous work of the path planning and generative models in Section II. Then, the preliminaries of path planning

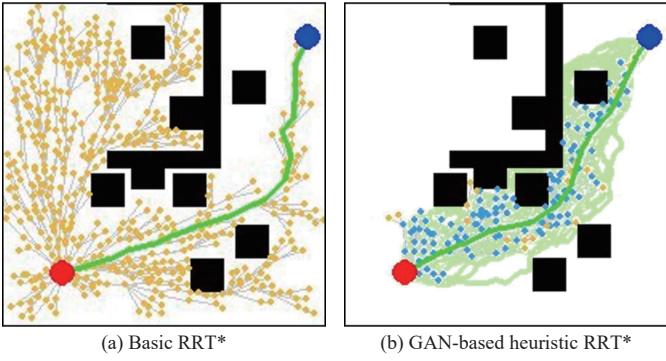


Fig. 1. A comparison of basic RRT\* and GAN-based heuristic RRT\* on finding the initial solution. The green area denotes the promising region generated by GAN. The yellow nodes are uniformly sampled from the whole state space and the blue nodes are sampled from the promising region.

problem are presented in Section III. Section IV describes the data generation method and the proposed GAN model for promising region prediction. In Section V, we conduct a series of simulation experiments to demonstrate the performance of the GAN-based heuristic path planning algorithm. At last, we draw conclusions and discuss the future work in Section VI.

## II. RELATED WORK

Recently, many researchers have proposed different types of heuristic methods to improve the performance of basic RRT algorithms.

As the grid-based algorithms can guarantee to find a *resolution optimal* path, one possible way is to combine the grid-based algorithms with sampling-based ones. In [11], the grid search process is employed to get an initial solution to guide the sampling process. Another way is to design some specific cost functions to change tree extension rules. Wang and Meng [12] utilize the generalized Voronoi graph (GVG) as the heuristic quality function to quickly extend the tree. Liu *et al.* [13] establish a set of partition heuristic rules for target and collision-free guidance. While these proposed cost functions accelerate the searching process, they tend to bias the searching towards a certain target, resulting in a local optimum easily or being stuck within complex obstacles. Moreover, the aforementioned methods are task-oriented and only demonstrate fine results on limited conditions, such as low-dimensional spaces or simple cost states. Their generality and performance cannot be guaranteed in complex environments. Controlling the sampling distribution is also a useful way to accelerate path planning. For example, Gammell *et al.* [14] shrink the sampling space into an elliptic region after a feasible path is obtained. However, its overall performance depends on the quality of the initial path.

In recent years, learning-based algorithms show great advantages on path planning problems. Especially, they exhibit remarkable generalization to completely unseen environments and can easily be applied to different sizes of maps. Baldwin and Newman [15] propose to learn sampling distributions by using expert data and learn an estimate of sample densities around semantic regions of interest, then incorporate them into a sampling-based planner to produce

natural plans. Wang *et al.* [16] use a reinforcement learning algorithm to improve the multi-RRT approach in narrow space. Even though the strategy enhances the local space exploration ability, the expansion of the searching trees is time-consuming. Ichter *et al.* [17] propose a general methodology for non-uniform sampling based on conditional variational autoencoder (CVAE). The weakness of this method is that sampling and planning are separated, so the planner cannot adapt to the environment during the planning cycle. Additionally, the construction of the sampling distributions requires lots of preprocessed conditional information, which demands huge effort on preliminary works. Moreover, the method is a multi-process generative model, which is time-consuming to predict the whole sampling distribution. Qureshi and Yip [18] present a neural network based adaptive sampler on generating samples for optimal paths. It contains an autoencoder to encode the environment from point cloud data and a dropout-based stochastic deep feed-forward neural network to mix the surrounding information and produce samples. However, their approach only focuses on path generation and requires some low-controllers to continuously track the generated paths, which may cause difficulty for robots to act in high-dimensional space. Wang *et al.* [19] propose a convolutional neural network (CNN) based optimal path planning algorithm to generate a promising region used to guide RRT\* sampling process. The CNN model is trained on a prior knowledge and can adapt to different clearances and step sizes. However, there are many disconnected parts in the generated regions. Mohammadi *et al.* [20] propose a GAN-based path planning algorithm for path planning. However, the model lacks universality as it is only trained and tested on very simple and small grided spaces. Choi *et al.* [21] employ GAN to predict multiple plausible paths to guide the driving intention. However, the method is used for intention generation, which can only predict simple and short-distance trajectories.

To step further in the learning-based algorithms for controlling the sampling distributions, we can first extract the promising regions of the bias sampling points. The process of obtaining promising regions can be regarded as a special problem of image semantic segmentation or domain transformation. A significant advantage of this image-based generation model is that it does not require any complicated preprocessing. In the past few years, different semantic segmentation networks are proposed, such as SegNet [22], U-Net [23], Mask R-CNN [24], and so on. In general, segmentation nets are confined to certain tasks and require labels in the input pictures. Therefore, the CNN segmentation networks may perform poorly on generating promising regions. Wang *et al.* [19] apply a segmentation network into path planning problem. However, the success rate and connectivity of the promising region are unsatisfactory.

CVAE [25] is a popular generative model. By reparameterization technique, the target images and conditions are encoded and transformed into normal distributions. Given the specified condition and random latent variables, the decoder is capable of generating various expected images. However, the images generated from CVAE models are often

blurred. More importantly, it is difficult to restore promising regions from normal distributions. Since the advent of GANs [26], it has been acknowledged as the most powerful generative model and widely used in various areas, such as image generation [27]–[29], image-to-image translations [30]–[32], and so on. Due to the flexibility of GANs, different architectures are proposed in recent years to tackle diverse problems. It is also suitable for generating promising regions of non-uniform sampling. Therefore, we design a novel GAN structure to learn promising regions for achieving heuristic non-uniform sampling to improve the performance of RRT\*. Compared with the previous methods, our image-based model avoids complicated preprocessing works of the state space and shows good adaptability to various environments.

### III. PRELIMINARIES

In this section, we first briefly introduce the path planning problem and then formulate our proposed GAN-based heuristic RRT\* algorithm.

#### A. Path Planning Problem

The basic path planning problem can be defined as follows. Let  $\mathcal{X} \in \mathbb{R}^n$  be the state space, with  $n \in \mathbb{N}^n$ ,  $n \geq 2$ . The obstacle space and the free space are denoted as  $\mathcal{X}_{\text{obs}}$  and  $\mathcal{X}_{\text{free}} = \mathcal{X}/\mathcal{X}_{\text{obs}}$ , respectively. Let  $x_{\text{init}}$  be the start state and  $x_{\text{goal}}$  be the goal state where  $x_{\text{init}} \in \mathcal{X}_{\text{free}}$ ,  $x_{\text{goal}} \in \mathcal{X}_{\text{free}}$ . Let  $\mathcal{X}_{\text{goal}}$  be the goal region where  $\mathcal{X}_{\text{goal}} = \{x \in \mathcal{X}_{\text{free}} \mid \|x - x_{\text{goal}}\| < r\}$ . The path planning algorithms aim to find a feasible path  $\sigma : [0, 1] \rightarrow \mathcal{X}_{\text{free}}$ ,  $\sigma(0) = x_{\text{init}}$  and  $\sigma(1) \in \mathcal{X}_{\text{goal}}$ .

Let  $\Sigma$  be the set of feasible paths and  $c(\sigma)$  be the cost function. The optimal path planning problem is to find a path  $\sigma^*$  with the lowest cost in  $\Sigma$

$$\begin{aligned} \sigma^* = \arg \min_{\sigma \in \Sigma} c(\sigma) \\ \text{s.t. } \sigma(0) = x_{\text{init}} \\ \sigma(1) \in \mathcal{X}_{\text{goal}} \\ \sigma(t) \in \mathcal{X}_{\text{free}}(t), \forall t \in [0, 1]. \end{aligned} \quad (1)$$

In this paper, the cost function between two states  $x_i$  and  $x_j$  is defined as

$$\text{Cost}(x_i, x_j) = \|x_i - x_j\|. \quad (2)$$

#### B. Heuristic RRT\*

Sampling-based algorithms solve path planning problems by constructing space-filling trees  $T(V, E)$  to search a path  $\sigma$  connecting start and goal states. The tree is built incrementally with samples drawn randomly from the free space  $\mathcal{X}_{\text{free}}$ .

To reduce unnecessary sampling, we construct a heuristic non-uniform sampling distribution  $\mathcal{X}_H \subset \mathcal{X}_{\text{free}}$  to improve the sampling process. A heuristic non-uniform sampling distribution refers to the state space where feasible paths exist with high probability. In order to obtain this non-uniform sampling distribution  $\mathcal{X}_H$ , we first generate a promising region  $\mathcal{S}$  with GAN model and then discretize it. The outline of our heuristic methodology is shown in Algorithm 1. The process PRGenerator() generates the promising region  $\mathcal{S}$ , and the Discretization() extracts sampling distribution  $\mathcal{X}_H$  from  $\mathcal{S}$ .

The sampling distribution  $\mathcal{X}_H$  is then fed into the heuristic sampling-based planning algorithm HeuristicSBP\*().

---

#### Algorithm 1 Outline of GAN-based heuristic RRT\*

---

```

Input:  $x_{\text{init}}, x_{\text{goal}}, \mathcal{X}$ 
Output:  $T(V, E)$ 
1  $V = x_{\text{init}}, E = \emptyset;$ 
2  $\mathcal{S} \leftarrow \text{PRGenerator}(x_{\text{init}}, x_{\text{goal}}, \mathcal{X});$ 
3  $\mathcal{X}_H \leftarrow \text{Discretization}(\mathcal{S});$ 
4  $T(V, E) \leftarrow \text{HeuristicSBP}^*(x_{\text{init}}, x_{\text{goal}}, \mathcal{X}, \mathcal{X}_H);$ 
5 Return  $T(V, E);$ 

```

---

In this paper, we formulate our model in complex 2-D environments and use 2-D environment map *Map* to represent the state space  $\mathcal{X}$ . As for the sampling-based algorithm, we employ the most general optimal path planning method RRT\* [10] as a representative. Algorithm 2 illustrates the detailed implementation of our heuristic RRT\* algorithm (HeuristicSBP\*() in Algorithm 1). To compare it with basic RRT\* algorithm, we use *UseHeuristic* to show the difference: Lines 3–7 are adopted by heuristic RRT\*, and Lines 8 to 9 are in basic RRT\*. We modify the uniform sampling strategy into a biased sampling strategy to select a random sample  $x_{\text{rand}}$ . Suppose that  $N$  samples are drawn in the sampling-based planning, and a parameter  $\mu$  is used to balance the uniform and non-uniform sampling strategies. In heuristic RRT\*,  $\mu N$  samples are randomly chosen from the non-uniform sampling distribution, while  $(1-\mu)N$  are sampled from the uniform sampling distribution (the whole state space). Once  $x_{\text{rand}}$  is determined, the following process is the same as basic RRT\*: We first use *Nearest()* to find the nearest state  $x_{\text{nearest}}$  on the current tree  $T(V, E)$  to  $x_{\text{rand}}$ , and then use *Steer()* to advance one step in the direction of the line connecting  $x_{\text{nearest}}$  and  $x_{\text{rand}}$  to find the new state  $x_{\text{new}}$ . If the connection between  $x_{\text{nearest}}$  and  $x_{\text{new}}$  is collision-free,  $x_{\text{new}}$  will be added to the tree  $T(V, E)$  and the rewire process *Rewire()* will be conducted to optimize the path. The process will be repeated until an  $x_{\text{new}}$  is located in  $\mathcal{X}_{\text{goal}}$  and the path cost converges to the optimal cost.

To sum up, the focus of our work lies in establishing an efficient generator to predict promising region  $\mathcal{S}$  given the conditions start state  $x_{\text{init}}$ , goal state  $x_{\text{goal}}$ , and environment map *Map* (shown in Algorithm 1, Line 2).

#### C. Probabilistic Completeness

Probabilistic completeness is essential for the sampling-based algorithm, which indicates whether the algorithm can find a feasible path if it exists.

The proposed heuristic RRT\* employs both uniform and non-uniform sampling. As defined in Section III-B, there will be  $\mu N$  samples drawn from the non-uniform sampling distribution and  $(1-\mu)N$  samples drawn from the uniform sampling distribution. As the number of iterations  $N$  goes to infinity, the following equation will hold:

$$\lim_{N \rightarrow \infty} \mathbb{P}(V \cup \mathcal{X}_{\text{goal}} \neq 0) = 1 \quad (3)$$

which means that the whole state space will be explored. In other words, if there is a feasible solution, it must be found as

the number of iterations  $N$  goes to infinity. Therefore, the probabilistic completeness is guaranteed.

---

**Algorithm 2** RRT\* and Heuristic RRT\*

---

```

Input:  $x_{\text{init}}, X_{\text{goal}}, X_H, \text{Map}$ , and UseHeuristic
Output:  $T(V, E)$ 

1  $V = x_{\text{init}}, E = \emptyset;$ 
2 for  $i = 1, \dots, N$  do
3   if UseHeuristic = True then
4     if Rand() <  $\mu$  then
5        $x_{\text{rand}} \leftarrow \text{Non-uniformSample}(X_H);$ 
6     else
7        $x_{\text{rand}} \leftarrow \text{UniformSample}(\text{Map});$ 
8   else
9      $x_{\text{rand}} \leftarrow \text{UniformSample}(\text{Map});$ 
10   $x_{\text{nearest}} \leftarrow \text{Nearest}(T(V, E), x_{\text{rand}});$ 
11   $x_{\text{new}} \leftarrow \text{Steer}(x_{\text{nearest}}, x_{\text{rand}});$ 
12  if ObstacleFree( $x_{\text{nearest}}, x_{\text{new}}$ ) then
13    Extend( $T(V, E), x_{\text{new}}$ );
14    Rewire();
15    if  $x_{\text{new}} \in X_{\text{goal}}$  then
16      Return  $T(V, E);$ 
17  Return failure;

```

---

#### IV. GAN-BASED PROMISING REGION GENERATION

In this section, we introduce the proposed GAN model in detail. Trained with a large amount of empirical promising region data, the GAN model is able to generate promising regions for non-uniform sampling given the conditions start state  $x_{\text{init}}$ , goal state  $x_{\text{goal}}$ , and environment map *Map*. Therefore, the inputs of the model are RGB images representing the environment map *Map*, start state  $x_{\text{init}}$ , and goal state  $x_{\text{goal}}$ . The output of the model is also an RGB image where the promising region is highlighted.

##### A. Dataset Generation

Fig. 2 illustrates the dataset. The environment maps ( $201 \times 201$  pixels) are randomly generated. In the environment maps, the obstacles are denoted as black, and the free spaces are denoted as white. On each map, 20 to 50 start and goal states are randomly chosen from the free space and denoted as red and blue, respectively. To obtain the ground truth promising region for training, we use the set of feasible paths to represent the promising region, as shown in Fig. 2. Specifically, We run 50 times RRT algorithm on each condition and draw all the feasible paths with green lines on one image.

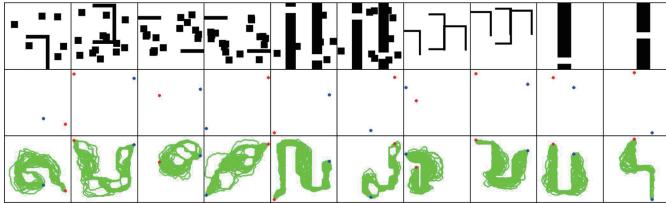


Fig. 2. An illustration of the dataset. Each row from top to bottom represents environment maps, start and goal states, and promising regions, respectively.

#### B. Model Structure

##### 1) Theory of GAN

We first briefly introduce the theory of GAN, and then discuss the connection between our model and the theory. The basic framework of GAN includes a generator  $G$  and a discriminator  $D$ . The generator accepts a sample noise  $z$  from the noise space  $\mathcal{Z} \in \mathbb{R}^n$  and outputs an image  $G(z)$ . The discriminator is fed with images  $u$  from the training data space  $\mathcal{U}$  and generates images  $G(z)$ . The generator and discriminator are in an adversarial game: to be specific, the generator aims at fooling the discriminator while the discriminator tries to distinguish the real and fake images. The objective function can be defined as

$$\min_G \max_D (\mathbb{E}_{u \sim P_{\mathcal{U}}(u)} [\log D(u)] + \mathbb{E}_{z \sim P_{\mathcal{Z}}(z)} [\log(1 - D(G(z)))]). \quad (4)$$

To control the generative process, the generator and discriminator can be conditioned on a given  $y \in \mathcal{Y}$ , where  $\mathcal{Y}$  is the conditioned space. With the conditional variable input, the generated images can be represented as  $G(z, y)$ . In this way, the objective function is modified into

$$\min_G \max_D (\mathbb{E}_{u, y \sim P_{\mathcal{U}, \mathcal{Y}}(u, y)} [\log D(u, y)] + \mathbb{E}_{z \sim P_{\mathcal{Z}}(z), y \sim P_{\mathcal{Y}}(y)} [\log(1 - D(G(z, y), y))]). \quad (5)$$

During the training process, we first label  $\mathcal{U}$  as real and  $G(z, y)$  as fake. Then the discriminator and generator are trained alternatively, which leads to the mutual growth of discernibility and generative capability.

In our model, we refer to the ground truth promising region images  $s \in \mathcal{S}$  as data space  $\mathcal{U}$ . Environment map images  $m \in \mathcal{M}$  and point images (start and goal states)  $p \in \mathcal{P}$  are two conditions in space  $\mathcal{Y}_{\text{map}}$  and  $\mathcal{Y}_{\text{point}}$ , respectively, and  $\mathcal{Z}$  is 2-D noise data. The task is to parameterize  $G$  to fit the distribution of  $G(z, m, p)$  to  $\mathcal{S}$ . To enhance the network's ability to locate start and goal states, we use two discriminators  $D_{\text{map}}$  and  $D_{\text{point}}$  to judge whether the promising regions match the environment map and point conditions, respectively. Fig. 3 shows the overall architecture of our model.

##### 2) Loss Function

Equations (4) and (5) show the object function of GAN. Herein, we introduce their implementations in the loss functions.

During the training process, we first fix the parameters in the generator and train the discriminators for 2 iterations. The loss functions of discriminators can be defined as

$$\begin{aligned} \mathcal{L}_{D_{\text{map}}} &= \mathbb{E}[\log D_{\text{map}}(s, m)] + \mathbb{E}[\log(1 - D_{\text{map}}(G(z, m, p), m))] \\ \mathcal{L}_{D_{\text{point}}} &= \mathbb{E}[\log D_{\text{point}}(s, p)] + \mathbb{E}[\log(1 - D_{\text{point}}(G(z, m, p), p))] \\ \text{s.t. } s, m, p &\sim P_{\mathcal{S}, \mathcal{M}, \mathcal{P}}(s, m, p), z \sim P_{\mathcal{Z}}(z). \end{aligned} \quad (6)$$

Then we train the generator with the following loss function:

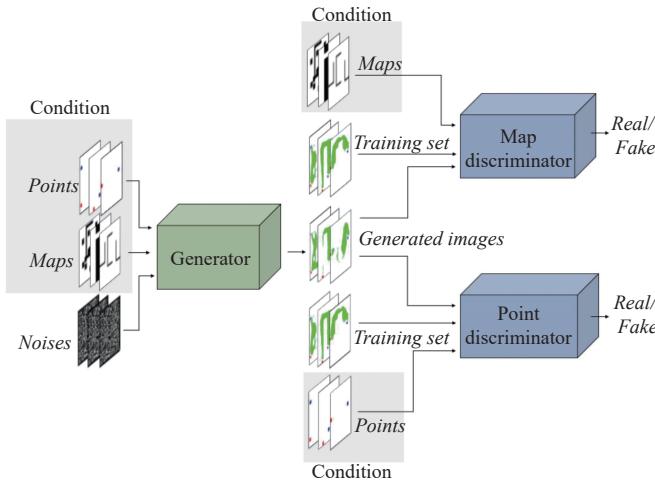


Fig. 3. The overall structure of GAN model for promising region generation.

$$\begin{aligned} \mathcal{L}_G = & \alpha_1 \mathbb{E}[\log D_{\text{map}}(G(z, m, p), m)] \\ & + \alpha_2 \mathbb{E}[\log D_{\text{point}}(G(z, m, p), p)] \\ \text{s.t. } & m, p \sim P_{\mathcal{M}, \mathcal{P}}(m, p), z \sim P_{\mathcal{Z}}(z). \end{aligned} \quad (7)$$

As the start and goal states occupy small pixels in the images, the generator may ignore the semantic information of them. To enhance the generator's attention to start and goal states, we design dynamic cross coefficients  $\alpha_1$  and  $\alpha_2$  to give a larger weight to the loss of  $D_{\text{point}}$ . The specific calculation formula is represented in (8).  $k$  is a hyperparameter. In our model, we set  $k$  to 3.

$$\begin{aligned} \mathcal{L}_G = & \frac{\mathcal{L}_{\text{point}}}{\mathcal{L}_{\text{point}} + k\mathcal{L}_{\text{map}}} \mathbb{E}[\log D_{\text{map}}(G(z, m, p), m)] \\ & + \frac{k\mathcal{L}_{\text{map}}}{\mathcal{L}_{\text{point}} + k\mathcal{L}_{\text{map}}} \mathbb{E}[\log D_{\text{point}}(G(z, m, p), p)] \\ \text{s.t. } & m, p \sim P_{\mathcal{M}, \mathcal{P}}(m, p), z \sim P_{\mathcal{Z}}(z). \end{aligned} \quad (8)$$

### 3) Detailed Model Structure

Fig. 3 shows the overall architecture of the GAN model for promising region generation and Fig. 4 illustrates the detailed structure of the generator and discriminators.

In the generator, the inputs are  $64 \times 64 \times 1$  dimensional noise image  $\mathcal{Z}$ ,  $64 \times 64 \times 3$  dimensional environment map image  $\mathcal{M}$  and  $64 \times 64 \times 3$  dimensional point image  $\mathcal{P}$ . The output is the  $64 \times 64 \times 3$  dimensional promising region image  $\mathcal{S}$ .  $\mathcal{M}$  and  $\mathcal{P}$  are separately fed into a 16-channel convolutional layer, and  $\mathcal{Z}$  is fed into a 32-channel convolutional layer, respectively. Then the feature maps are concatenated together along channel dimension and processed by the subsequent networks to encode the images into multi-channel features.

In our model, the generator obeys the encoder-decoder architecture. The encoding network contains *block1* to *block4*, and each block is composed of a Convolution-BatchNormalization-ReLU layer and two Resnet cells [33]. Herein, we store the feature maps derived from *block1* and denote it as  $I_1$ . The decoding network includes *block5* to *block7*. Each decoder block is similar to the encoder ones,

only replacing the convolutional layers with deconvolutional layers. Likewise, we denote the output feature maps of the decoding network as  $I_2$ . *Block8* is designed to improve the image quality by enhancing raw context information. The feature maps  $I_1$  and  $I_2$  are concatenated along the channel and fed into *block8*. At the end of the network, the feature maps are compressed into a 3-channel image and activated by Tanh function.

The two discriminators have similar structures except for the number of the layers. The discriminators are mainly constructed by Convolution-BatchNormalization-LeakyReLU blocks. Inspired by the self-attention mechanism [34], [35], we add self-attention blocks at the first layer and the third layer of the discriminators to help them consider global information.

As shown in Fig. 4, self-attention blocks are added to the convolutional network through residual connection

$$\mathbf{o}_a = \gamma \mathbf{q} + \mathbf{o}_c \quad (9)$$

where  $\mathbf{q}$  refers to the output of self-attention mechanism,  $\mathbf{o}_c$  denotes the output of the last layer, and  $\mathbf{o}_a$  is the final output.  $\gamma$  is a learned parameter to adjust the ratio of self-attention. The self-attention mechanism can be defined as

$$\begin{aligned} \mathbf{q} &= \text{softmax}(\theta(\mathbf{o}_c)^T \phi(\mathbf{o}_c)) g(\mathbf{o}_c) \\ &= \text{softmax}(\mathbf{o}_c^T W_\theta^T W_\phi \mathbf{o}_c) W_g \mathbf{o}_c \end{aligned} \quad (10)$$

where  $g(\mathbf{o}_c)$  is a linear embedding used to compress dimensions.  $\theta(\mathbf{o}_c)$  and  $\phi(\mathbf{o}_c)$  are used to calculate autocorrelation. Through *softmax* operation, the weight coefficients are derived and added to  $g(\mathbf{o}_c)$ . In this way, all positions are considered in the operation.

In the discriminators, the condition input layer is fed with  $64 \times 64 \times 3$  dimensional environment maps  $\mathcal{M}$  or point images  $\mathcal{P}$ , and the data input layer receives ground truth promising region images  $\mathcal{S}$  and generated promising region images  $G(z, m, p)$ . When the data is obtained from the training set  $\mathcal{S}$ , we label the output of discriminators as 1 to mark it as real. When the data is from the generator  $G(z, m, p)$ , we label the output as 0 to mark it as fake.

## V. EXPERIMENTS

### A. Training Details

Our dataset contains a total of 61 768 sets of data. We randomly choose 49 416 among them for training and 3088 for test. As shown in Fig. 2, we mainly use five types of environment maps, each of which contains 100 to 300 different images. On each environment map, we randomly choose 20 to 50 different start and goal states. The training process is conducted for 50 epochs on NVIDIA TESLA T4 with PyTorch. We use Adam optimizer with parameters  $\beta_1 = 0.5$ ,  $\beta_2 = 0.99$  for training, and the learning rates of the discriminators and generator are 0.00001 and 0.0001, respectively.

### B. Evaluation Methods

We evaluate our model from two aspects. As an image generation problem, we evaluate the connectivity of the

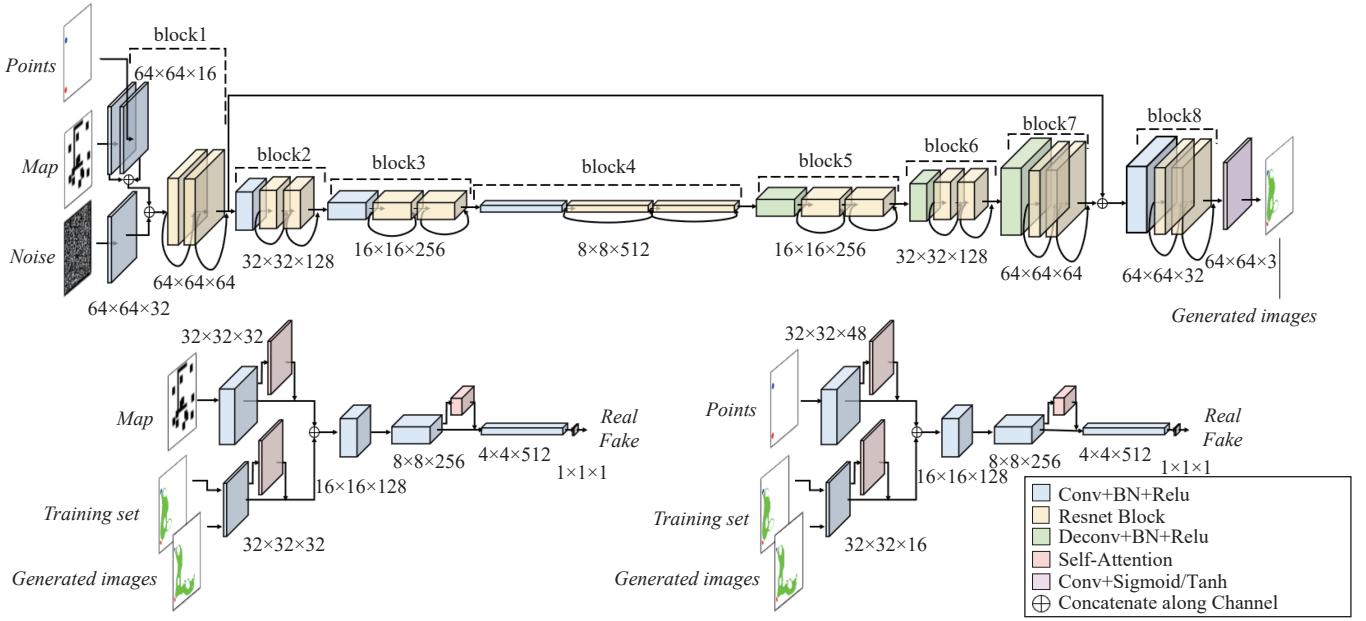


Fig. 4. The detailed structure of GAN model for promising region generation.

generated promising regions and generalization ability of the model. As a path planning problem, we apply the promising region to Algorithm 2 to compare several standard metrics between basic RRT\* and our GAN-based heuristic RRT\*.

### 1) Connectivity

Compared with the high similarity between the generated images and the ground truth images, the contribution of the generated regions to the planning tasks is more worthy of our attention. Therefore, we use connectivity to evaluate the quality of the image rather than other image similarity evaluation metrics, such as histogram-based method, SSIM, L1 loss, etc. For path planning problems, the best promising region is the unbroken area connecting the start and goal states. Therefore, we employ path planning algorithms to assist in testing the connectivity. Specifically, we set the promising region (green area) as free space  $X_{\text{free}(\text{test})}$  and denote it as white. Then we execute the RRT algorithm in  $X_{\text{free}(\text{test})}$ : if feasible paths can be found, it means that  $X_{\text{free}(\text{test})}$  is connected.

Fig. 5 displays the process of our task-based connectivity evaluation method and its advantages. In Fig. 5(a), although the shape of the generated image and the image in the training set are very similar according to the image evaluation metrics, the quality of the generated image is not good because it does not achieve our task, which is to find the promising region connecting the start and goal states. In Fig. 5(b), even though the generated image looks very different from the image in the training set, the quality of the generated image is good because the start and goal states are well connected.

### 2) Generalization Ability

Another measurement for model quality is the generalization ability in completely different environments. Herein, we use two additional datasets for test. The first additional test set includes 6 types of unseen environment maps, as shown in Fig. 6(a). As the method shown in Section IV-A, we randomly set 20 to 50 pairs of start and goal states and run RRT to

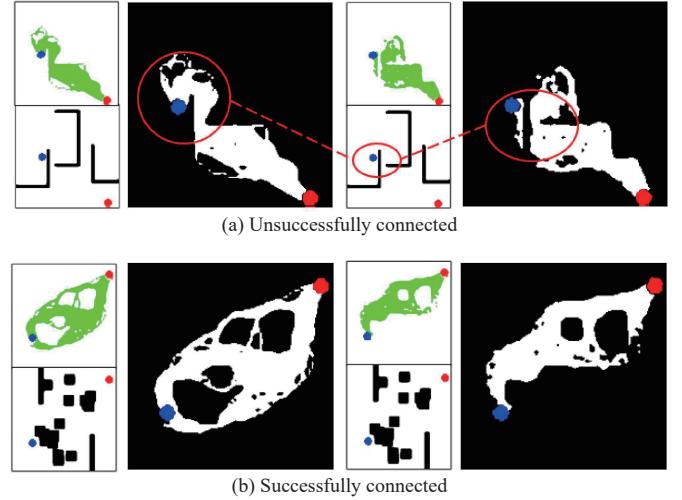


Fig. 5. An illustration of the method for connectivity evaluation. In each subfigure, the left sets of images are the ground truth promising region and its corresponding environment map, and the right images are the generated ones.

generate ground truth promising regions for comparison. In order to further test the generality of our model, we use more complex maze maps as another additional test set, which is shown in Fig. 6(b).

### 3) Improvement for RRT\*

To test the improvement of the proposed GAN-based promising region generation on sampling-based planning algorithms, we extract the heuristic non-uniform sampling distribution from the promising region and employ the heuristic RRT\* (shown in Algorithms 1 and 2). To highlight the efficiency of GAN-based heuristic method and guarantee the probabilistic completeness simultaneously, we choose the hyperparameter  $\mu$  in Algorithm 2 as 0.9, which means that 90% of nodes are randomly chosen from the heuristic non-uniform sampling distribution, and 10% are chosen uniformly

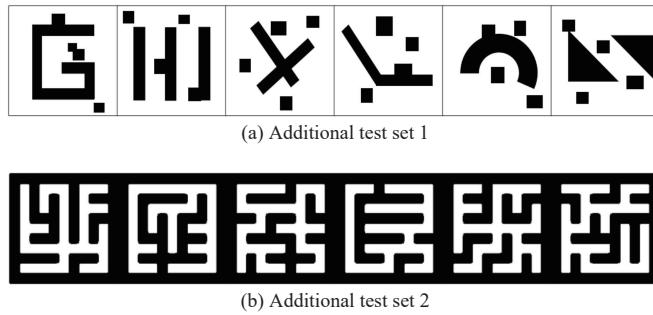


Fig. 6. Examples of different maps for generalization ability test.

from the whole state space.

We compared three indexes in the process of finding the initial path and the optimal path, including path cost, number of nodes, and computing iterations. Path cost represents the quality of the path, and we use the Euclidean distance as the cost function (shown in (2)). The other two indexes reflect the availability of the algorithm when applied to practical applications. The number of nodes indicates the requirement of memory space, and computing iteration refers to the time consumption of finding a feasible path or an optimal path. Instead of the running time, we use computing iterations to represent time consumption to avoid time differences caused by different operating platforms.

### C. Experiment Results

#### 1) Connectivity

In our experiment, the success rate reaches 89.83%. Some of the results are represented in Fig. 7. It demonstrates that our model can generate high-quality continuous promising regions on different conditions.

#### 2) Generalization Ability

As shown in Fig. 8, our model shows good adaptability in completely different environments. It predicts accurate continuous promising regions in the two additional test sets, and the success rate can achieve 81.93% and 75.56%, respectively.

#### 3) Improvement on RRT\*

To illustrate the improvement on RRT\*, we randomly choose one result from each type of the environment maps in the test set, which is represented in Fig. 9. As RRT\* has strong randomness, we run the GAN-based heuristic RRT\* and the basic RRT\* 100 times on Python 3.8 to get the statistics of evaluation results. Figs. 10–12 show the comparison results of our GAN-based heuristic RRT\* and the basic RRT\* during the initial stage and optimization stage.

##### a) Initial stage

The initial stage refers to the period from the beginning to the moment when the first feasible path is found. Fig. 11 displays the initial expansion of RRT\* trees. In Fig. 11(a), the non-uniform sampling distribution guides RRT\* to search in the space with the high possibility of feasible paths, resulting in lower path cost within the initial planning stage. Fig. 11(b) represents a more difficult task and highlights the time advantage of the GAN-based heuristic RRT\*. The result shows that the basic RRT\* needs 1352 more iterations than

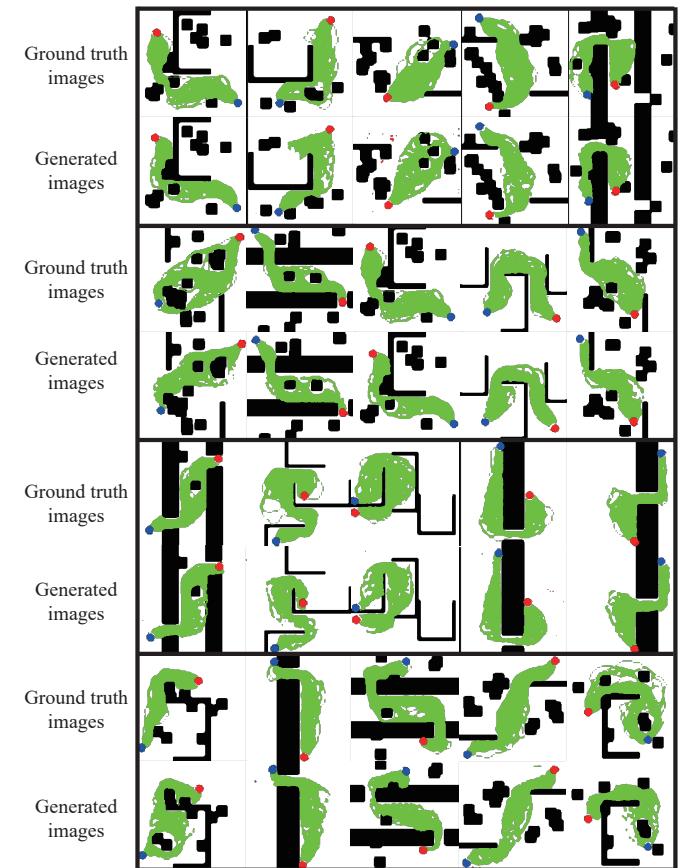


Fig. 7. Examples of the generated promising regions in the test set. In each subrow, The upper images are the ground truth promising regions, and the lower images are the generated promising regions.

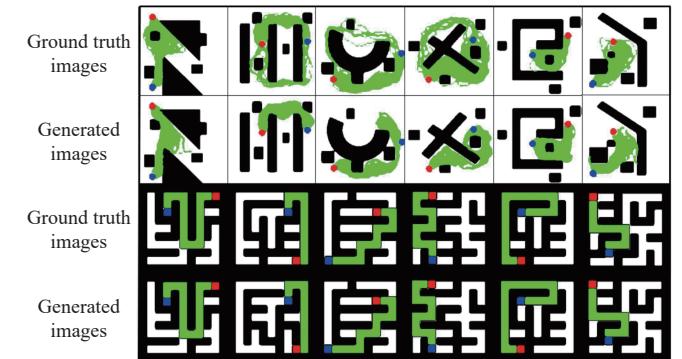


Fig. 8. Examples of generated promising region images in the additional test sets. The upper images are the ground truth promising regions, and the lower images are the generated promising regions.

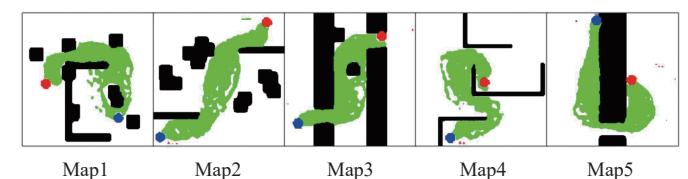


Fig. 9. Five conditions for testing the improvement of GAN-based heuristic RRT\*.

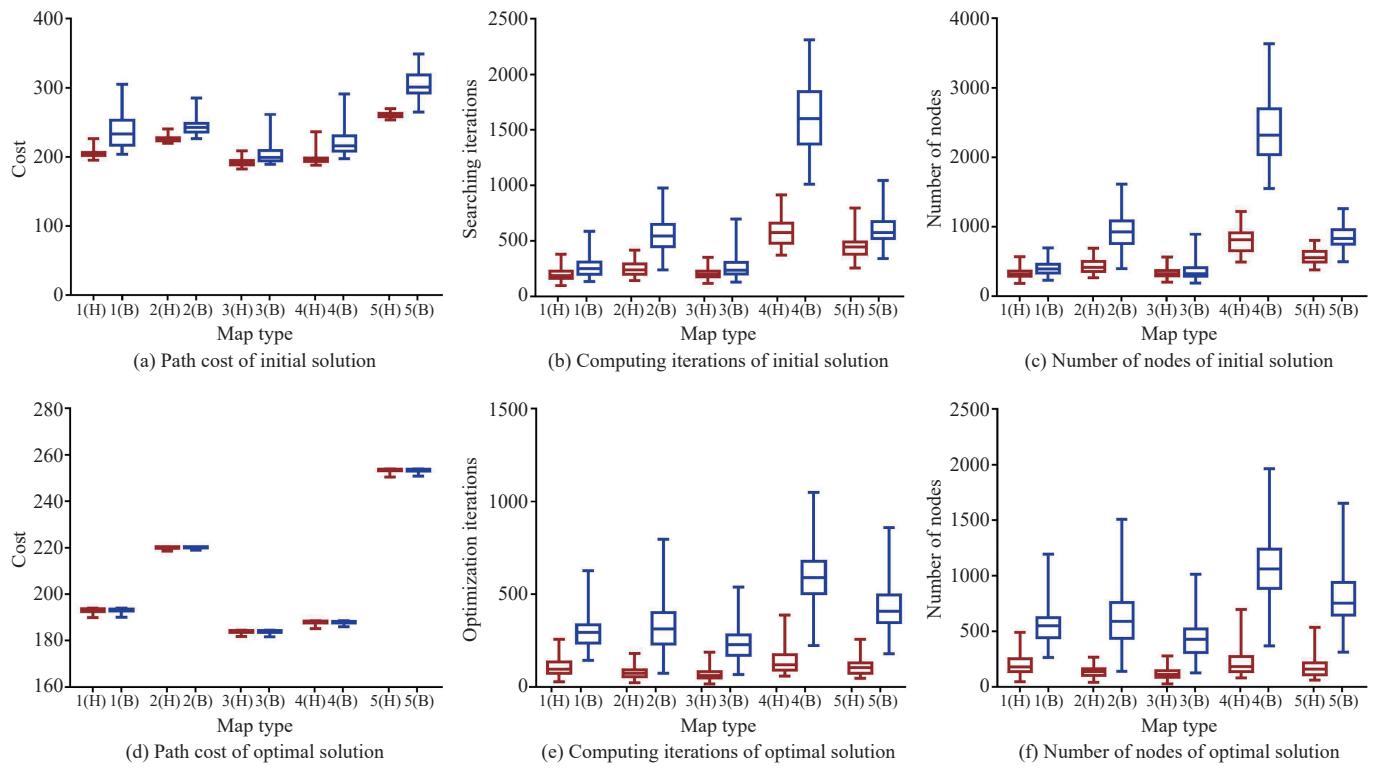


Fig. 10. Comparison of path cost, computing iterations, and the number of nodes between GAN-based heuristic RRT\* and basic RRT\* during initial stage and optimization stage. The red boxes refer to the GAN-based heuristic RRT\* and the blue boxes indicate basic RRT\*.

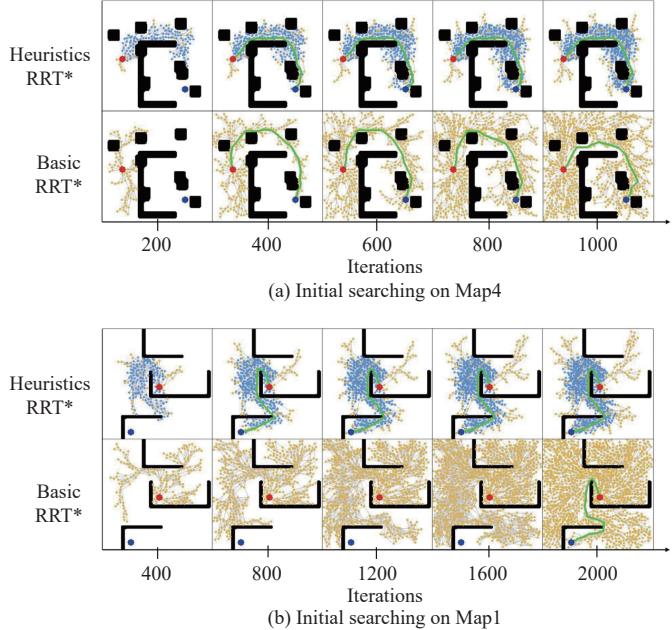


Fig. 11. Comparison of the searching process between GAN-based heuristic RRT\* and basic RRT\* at the initial stage of planning. The X-axis refers to the number of searching iteration in RRT\*. The yellow nodes are uniformly sampled from the whole state space, and the blue nodes are sampled from the non-uniform sampling distribution, respectively. The initial paths are marked as green.

the GAN-based heuristic RRT\* to find the goal.

Figs. 10(a)–10(c) display the box-plot of the comparison on initial path cost, computing iterations, and the number of

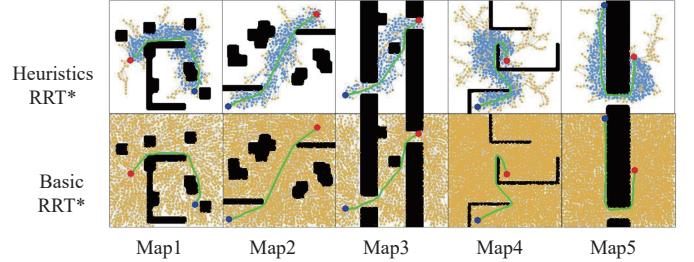


Fig. 12. Comparison of RRT\* expansion. The figures are the final moments of RRT\*. The yellow nodes are uniformly sampled from the whole state space and the blue nodes are from the non-uniform sampling distribution. Start states, goal states and optimal paths are marked as red, blue, green, respectively.

nodes. The midpoint of the boxes refers to average value, and the height of boxes refers to variance. According to the results, the GAN-based heuristic RRT\* has better performance on the three indexes. Additionally, the effectiveness is more obvious on difficult tasks (such as Map4). Furthermore, the heuristic non-uniform sampling distribution reduces the variance of different iterations, which means that it increases the robustness of RRT\* to find a feasible path in a short time.

#### b) Optimization stage

The optimization stage refers to the period from the beginning to the time when the optimal path is found. Fig. 10(d)–10(f) compare the statistical results of the three metrics in the optimization process. As shown in Fig. 10(d), the cost of optimal path remains the same between the basic RRT\* and GAN-based heuristic RRT\*, which implies that taking most of the sampling nodes from our non-uniform sampling distribu-

tion will not affect the cost of the optimal path. Figs. 10(e) and 10(f) illustrate the significant decrease in time consumption and the number of nodes: our GAN-based heuristic RRT\* achieves 4 to 12 times faster, and the number of nodes is reduced by about 30%.

The advantage of our GAN-based heuristic RRT\* is more obvious during the optimization stage, for the non-uniform sampling distribution guides RRT\* to avoid sampling in low possibility region, resulting in fewer nodes to search during the optimization process of RRT\*.

Figs. 12 and 13 demonstrate the above-mentioned benefits more intuitively. Fig. 12 displays the final expansion of RRT\* trees when optimal paths are found. The nodes of basic RRT\* flood the whole state space. In comparison, the number of the nodes of GAN-based heuristic RRT\* is much less than the basic RRT\*, and the nodes distribute in the regions of high possibility. Fig. 13 shows the convergence process of the basic RRT\* and GAN-based heuristic RRT\*. Figs. 13(a) and 13(c) exhibit the convergence of path cost over time, and Figs. 13(b) and 13(d) display the increment of nodes. The comparison demonstrates that our GAN-based heuristic RRT\* can significantly reduce the planning time and quantity of sampling nodes.

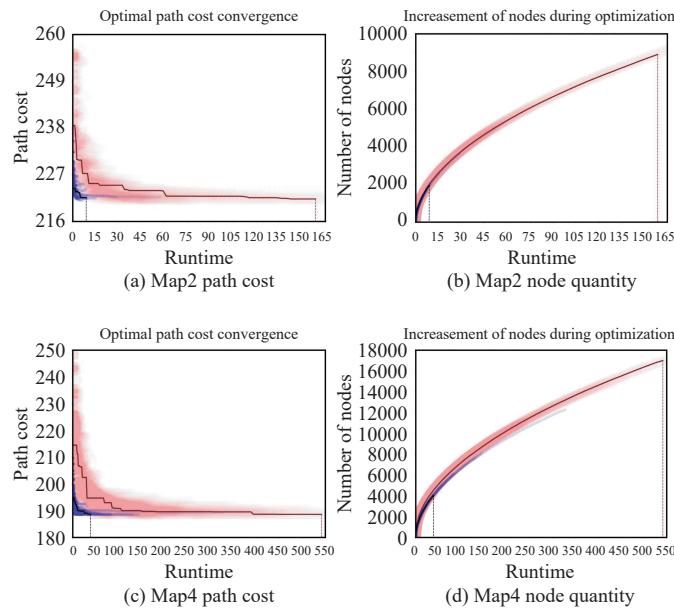


Fig. 13. Convergence process of path optimization. The blue translucent scatters refers to the multiple experiments of GAN-based heuristic RRT\*, and the red scatters refer to basic RRT\*'s. The solid line of each color is a demo of the searching process. The dotted line connecting the solid line and the  $X$ -axis marks the end moment of the search.

#### D. Discussions

Herein, we make some extended discussions on the current model and propose some promising improvements.

##### 1) Extensions

Our model is applicable to large, complex, and dynamic situations. When the environment maps are much larger than the current input size, we can resize the large maps into

smaller ones and use the same network to generate promising regions, and then interpolate the maps into the original size. On the other hand, our model can also directly process larger-scale environment maps by increasing the scale of the network. Furthermore, our method can also be extended to three-dimensional space.

In dynamic environments, each frame of the dynamic process can be seen as a static map, so we can use the same method to process each frame (or take a frame at intervals), and generate the sampling distribution in real time. The generator of our model takes 20 ms to generate the promising region of a map in Python 3.8, which meets the real-time requirement. When facing very different environment maps, the accuracy of the trained GAN model will drop, but we can add this type of maps to the training set, and the accuracy will get improved.

##### 2) Improvements

As for GAN model, its architecture can be further improved in a few aspects to increase the success rate of prediction. For example, the current model performs imperfectly on generating promising regions that cross long distances. This is mainly because the long-distance paths account for a small ratio in the whole randomly-generated dataset, resulting in difficulties for the model to learn distant connections. In addition, there is enough room for improvement, such as simplifying structure for lightweight network, repeating cycles for more accurate predictions, and so on.

In terms of the implementation of sampling-based planning algorithms, heuristic rate  $\mu$  may have better value to balance randomness and guidance. Furthermore, more intelligent sampling strategies can be applied to the heuristic sampling-based algorithm to avoid redundant node selection when moving towards the goal.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we present an image-based heuristic methodology to guide non-uniform sampling-based path planning algorithms. In particular, we design a GAN model to predict the promising region and apply it to generate heuristic non-uniform sampling distribution for the heuristic RRT\* algorithm. We evaluate the GAN model from the image generation aspect and path planning aspect. As an image generation problem, the model shows a high success rate and strong adaptability. As a path planning problem, our method shows significant improvement in the quality of initial paths and greatly accelerates the convergence speed to the optimum. Our heuristic method is more human-intuitive, which can avoid complicated preprocessing on the state space, thus not confining to a specific environment.

Much research can be continued based on this work. One extension is to add kinodynamic conditions (such as step size, rotation angle, etc.) as inputs to improve regional accuracy and enable the network to adjust the promising region according to these constraints. Another promising extension is to apply the model to dynamic environments. Specifically, the small black blocks in our environment maps can be considered as moving pedestrians and vehicles. Through computing our model on each frame (or take a frame at

intervals), it is feasible to guide sampling-based planning in dynamic environments. The third promising work is to expand the methodology into high-dimensional configuration space to guide sampling on more complicated conditions.

## REFERENCES

- [1] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Springer, 2016.
- [2] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *Int. Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [3] F. Lingelbach, “Path planning using probabilistic cell decomposition,” in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, 2004, pp. 467–472.
- [4] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE Trans. Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [5] A. Stentz, “Optimal and efficient path planning for partially known environments,” in *Intelligent Unmanned Ground Vehicles*. Springer, 1997, pp. 203–220.
- [6] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Trans. Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [7] S. M. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” 1998.
- [8] D. J. Webb and J. Van Den Berg, “Kinodynamic RRT\*: Asymptotically optimal motion planning for robots with linear dynamics,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 2013, pp. 5054–5061.
- [9] J. Wang and M. Q.-H. Meng, “Socially compliant path planning for robotic autonomous luggage trolley collection at airports,” *Sensors*, vol. 19, no. 12, p. 2759, 2019.
- [10] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [11] A. Hentout, A. Maoudj, D. Yahiaoui, and M. Aouache, “RRT-A\*-BT approach for optimal collision-free path planning for mobile robots,” 2019.
- [12] J. Wang and M. Q.-H. Meng, “Optimal path planning using generalized voronoi graph and multiple potential functions,” *IEEE Trans. Industrial Electronics*, vol. 67, no. 12, pp. 10621–10630, 2020.
- [13] Z. Liu, F. Lan, and H. Yang, “Partition heuristic RRT algorithm of path planning based on Q-learning,” in *Proc. IEEE 4th Advanced Information Technology, Electronic and Automation Control Conf.*, vol. 1, 2019, pp. 386–392.
- [14] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, “Informed RRT\*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2014, pp. 2997–3004.
- [15] I. Baldwin and P. Newman, “Non-parametric learning for natural plan generation,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2010, pp. 4311–4317.
- [16] W. Wang, L. Zuo, and X. Xu, “A learning-based multi-RRT approach for robot path planning in narrow passages,” *Journal of Intelligent & Robotic Systems*, vol. 90, no. 1–2, pp. 81–100, 2018.
- [17] B. Ichter, J. Harrison, and M. Pavone, “Learning sampling distributions for robot motion planning,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 2018, pp. 7087–7094.
- [18] A. H. Qureshi and M. C. Yip, “Deeply informed neural sampling for robot motion planning,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2018, pp. 6582–6588.
- [19] J. Wang, W. Chi, C. Li, C. Wang, and M. Q.-H. Meng, “Neural RRT\*: Learning-based optimal path planning,” *IEEE Trans. Automation Science and Engineering*, vol. 17, no. 4, pp. 1748–1758, 2020.
- [20] M. Mohammadi, A. Al-Fuqaha, and J.-S. Oh, “Path planning in support of smart mobility applications using generative adversarial networks,” in *Proc. IEEE Int. Conf. Internet of Things and IEEE Green Computing and Communications and IEEE Cyber, Physical and Social Computing and IEEE Smart Data*, 2018, pp. 878–885.
- [21] D. Choi, S.-j. Han, K. Min, and J. Choi, “Pathgan: Local path planning with generative adversarial networks,” arXiv preprint arXiv: 2007.03877, 2020.
- [22] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [23] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.
- [24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proc. IEEE Int. Conf. Computer Vision*, 2017, pp. 2961–2969.
- [25] K. Sohn, H. Lee, and X. Yan, “Learning structured output representation using deep conditional generative models,” in *Advances in Neural Information Processing Systems*, 2015, pp. 3483–3491.
- [26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.
- [27] Y. Lu, S. Wu, Y.-W. Tai, and C.-K. Tang, “Image generation from sketch constraint using contextual gan,” in *Proc. European Conf. Computer Vision*, 2018, pp. 205–220.
- [28] C. Han, H. Hayashi, L. Rundo, R. Araki, W. Shimoda, S. Muramatsu, Y. Furukawa, G. Mauri, and H. Nakayama, “GAN-based synthetic brain MR image generation,” in *Proc. IEEE 15th Int. Symp. Biomedical Imaging*, 2018, pp. 734–738.
- [29] M. Zhai, L. Chen, F. Tung, J. He, M. Nawhal, and G. Mori, “Lifelong GAN: Continual learning for conditional image generation,” in *Proc. IEEE Int. Conf. Computer Vision*, 2019, pp. 2759–2768.
- [30] A. Cherian and A. Sullivan, “Sem-GAN: Semantically-consistent image-to-image translation,” in *Proc. IEEE Winter Conf. Applications of Computer Vision*, 2019, pp. 1797–1806.
- [31] Y. Li, S. Tang, R. Zhang, Y. Zhang, J. Li, and S. Yan, “Asymmetric GAN for unpaired image-to-image translation,” *IEEE Trans. Image Processing*, vol. 28, no. 12, pp. 5881–5896, 2019.
- [32] X. Guo, Z. Wang, Q. Yang, W. Lv, X. Liu, Q. Wu, and J. Huang, “GAN-based virtual-to-real image translation for urban scene semantic segmentation,” *Neurocomputing*, vol. 394, pp. 127–135, 2020.
- [33] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [34] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

- [35] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” in *Proc. Int. Conf. Machine Learning*. PMLR, 2019, pp. 7354–7363.



**Tianyi Zhang** received the B.E. degree in automation from Harbin Institute of Technology in 2020. She is currently a Research Assistant with the Department of Electronic and Electrical Engineering of the Southern University of Science and Technology. Her current research interests include machine learning in robotics and computer vision.



**Jiankun Wang** received the B.E. degree in automation from Shandong University in 2015, and the Ph.D. degree in Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, China, in 2019. He is currently a Research Assistant Professor with the Department of Electronic and Electrical Engineering of the Southern University of Science and Technology. During the Ph.D. study, he spent six months at Stanford University, USA, as a Visiting Student Scholar supervised by Prof. Oussama Khatib. His current research interests include motion planning and control, human robot interaction, and machine learning in robotics.



**Max Q.-H. Meng** (Fellow, IEEE) is currently a Chair Professor and the Head of the Department of Electronic and Electrical Engineering at the Southern University of Science and Technology, on leave from the Department of Electronic Engineering at The Chinese University of Hong Kong, China. He received the Ph.D. degree in electrical and computer engineering from the University of Victoria, Canada, in 1992. He joined The Chinese University of Hong Kong, China, in 2001 as a Professor and later the Chairman of Department of Electronic Engineering. He was with the Department of Electrical and Computer Engineering at the University of Alberta in Canada, where he served as the Director of the ART (Advanced Robotics and Teleoperation) Lab and held the positions of Assistant Professor (1994), Associate Professor (1998), and Professor (2000), respectively. He is an Honorary Chair Professor at Harbin Institute of Technology and Zhejiang University, and also the Honorary Dean of the School of Control Science and Engineering at Shandong University. His research interests include robotics, perception, and intelligence. He has published more than 750 journal and conference papers and book chapters and led more than 60 funded research projects to completion as Principal Investigator. He has been serving as the Editor-in-Chief and editorial board of a number of international journals and as the General Chair or Program Chair of many international conferences, including the General Chair of IROS 2005 and General Chair of ICRA 2021 held in Xi'an in May 2021. He served as an Associate VP for Conferences of the IEEE Robotics and Automation Society (2004–2007), Co-Chair of the Fellow Evaluation Committee and an elected Member of the AdCom of IEEE RAS. He is a Recipient of the IEEE Millennium Medal, a Fellow of IEEE, a Fellow of Hong Kong Institution of Engineers, and an Academician of the Canadian Academy of Engineering.