

The background of the entire image is a dark blue field filled with a pattern of red dots. These dots are arranged in a way that they form a large, faint, stylized 'H' shape, which is the primary visual element behind the text.

HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.



**ĐẠI HỌC
BÁCH KHOA HÀ NỘI**
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Neural Informed RRT*: Learning-based Path Planning with Point Cloud State Representations under Admissible Ellipsoidal Constraints

**Nguyễn Nhật Minh – 20225043
Ngô Trí Cảnh - 20220015**

ONE LOVE. ONE FUTURE.

Nội dung chính

- I. Introduction
- II. Related work
- III. Method

I. Introduction

Bài toán lập kế hoạch hiệu quả cần thoả mãn:

- tính đầy đủ và tối ưu:
 - đảm bảo tìm ra giải pháp tối ưu nếu tồn tại
 - đảm bảo đạt được giải pháp tối ưu với thời gian chạy đủ
- hiệu quả trong hội tụ tối ưu:
 - cần nhanh chóng đạt đc tới gần mức tối ưu
- linh hoạt, có thể mở rộng:
 - có thể tổng quát hoá trên các môi trng khác nhau, với các robot khác nhau
- Mục tiêu là kết hợp những ưu điểm của các phương pháp lập kế hoạch dựa trên lấy mẫu (sampling-based) tối ưu như RRT* và Informed RRT* (IRRT*) với sức mạnh của học sâu (deep learning) để tăng tốc độ hội tụ đến đường đi tối ưu, đặc biệt là trong các môi trường phức tạp

I. Introduction

Các phương pháp nền tảng trước đó

	Ưu điểm	Hạn chế
RRT	linh hoạt, phổ biến, khả năng mở rộng	không tối ưu
RRT*	cung cấp tính tối ưu tiệm cận	hội tụ chậm do khám phá đồng đều toàn bộ không gian.
Informed RRT (IRRT)	Cải thiện RRT* bằng cách giới hạn việc lấy mẫu trong một tập con hình ellipsoid được xác định bởi chi phí đường đi tốt nhất hiện tại → tìm kiếm vào vùng không gian có khả năng chứa đường đi tốt hơn.	Không ưu tiên các trạng thái quan trọng về mặt cấu trúc liên thông (topologically critical) bên trong ellipsoid (ví dụ: hành lang hẹp).
Học máy	Sử dụng mạng nơ-ron (thường là CNN trên lưới ô vuông - grid) để học cấu trúc không gian tự do và dự đoán các trạng thái gần đường đi tối ưu (trạng thái dẫn đường) để hướng dẫn việc lấy mẫu.	<ul style="list-style-type: none">• bị ảnh hưởng bởi các vùng không liên quan hoặc chướng ngại vật• không có sự cải thiện lặp đi lặp lại của trạng thái dẫn đường dựa trên chi phí đường đi hiện tại• không xem xét tính kết nối của các trạng thái dẫn đường được suy luận

I. Introduction

- Đề xuất NIRRT*

- Kết hợp điểm mạnh của **IRRT*** và **học máy**.
- Sử dụng **biểu diễn đám mây điểm** cho không gian tự do (thay vì lưới ô vuông).
- Sử dụng **PointNet++** để suy luận trạng thái dẫn đường từ đám mây điểm.
- **Neural Focus**: Giới hạn đám mây điểm đầu vào cho PointNet++ nằm trong ellipsoid của IRRT*. Điều này cho phép mạng tập trung vào vùng quan trọng và cải thiện chất lượng suy luận khi chi phí đường đi giảm.
- **Neural Connect**: Xây dựng kết nối cho tập trạng thái dẫn đường suy luận được, giải quyết vấn đề các "đốm" rời rạc.

II. Related work

- Các phương pháp kinh điển: Tìm kiếm trên lưới (A^* , D^*), Lấy mẫu (PRM, RRT, RRT*, IRRT*, BIT*).
- Đề cập các kỹ thuật tăng tốc khác: Voronoi bias, thuật toán tiến hóa, khởi tạo A^* .
- Các phương pháp dựa trên học máy:
 - Neural RRT* (NRRT*): Dùng U-Net trên ảnh 2D để dự đoán bản đồ nhiệt xác suất.
 - MPNet: Dùng 3D CNN trên voxel để tạo đường đi.
 - Các phương pháp dùng PointNet để mã hóa chướng ngại vật.
 - Các phương pháp dùng Graph Neural Network trên đồ thị hình học ngẫu nhiên.
- Hạn chế của các phương pháp: rời rạc hóa, phụ thuộc độ phân giải.
- Sự khác biệt của NIRRT*: cải thiện liên tục, biểu diễn điểm.

III. Method

- A. Định nghĩa bài toán

- Không gian trạng thái:

$$X \subseteq \mathbb{R}^d$$

- Không gian chướng ngại vật: X_{obs}
- Không gian trống: X_{free}

- Đường đi:

$$\sigma: [0, 1] \rightarrow X_{\text{free}}$$

Một đường đi cụ thể, được xem như một hàm liên tục từ khoảng $[0, 1]$ vào không gian trống.

$\sigma(0)$ là điểm bắt đầu, $\sigma(1)$ là điểm đích.

- Tập tất cả các đường đi: Σ

- Hàm chi phí:

$$c: \Sigma \rightarrow \mathbb{R}_{>0}$$

- Mục tiêu bài toán:

Tìm ra đường đi σ^* sao cho:

- $\sigma^* = \operatorname{argmin}_{\sigma \in \Sigma} c(\sigma)$
- $\sigma(0) = x_{\text{start}}, \sigma(1) = x_{\text{goal}}, \forall s \in [0, 1], \sigma(s) \in X_{\text{free}}$

B. Neural Informed RRT*

- **Phần RRT* cơ bản:** phần không được highlight
- **Phần IRRT* (màu xanh):** Sử dụng hàm InformedSampling (lấy mẫu trong ellipsoid) như một phần của chiến lược lấy mẫu trong PointNetGuidedSampling (Algorithm 2).
- **Phần Đóng góp (màu đỏ):**
 - **C_{best}:** Lưu chi phí (độ dài) của đường đi tốt nhất từ start đến vùng đích tìm được cho đến nay.
 - **C_{update}:** Chi phí dùng để quyết định khi nào gọi lại mạng nơ-ron, khởi tạo bằng c_{best}.
 - **X_{guide}:** Tập hợp các điểm/trạng thái được mạng nơ-ron dự đoán là "tốt" (gần đường đi tối ưu).
 - **PointNetGuide (Algorithm 3):** Hàm gọi mạng nơ-ron (PointNet++) để tính toán/cập nhật X_{guide}. Được gọi lần đầu ở dòng 5 và có thể được gọi lại bên trong PointNetGuidedSampling (Algorithm 2).
 - **PointNetGuidedSampling (Algorithm 2):** Hàm quyết định cách lấy mẫu điểm ngẫu nhiên x_{rand} ở mỗi bước.

Algorithm 1 Neural Informed RRT*

```

1:  $V \leftarrow \{x_{\text{start}}\}; E \leftarrow \emptyset;$ 
2:  $X_{\text{soln}} \leftarrow \emptyset;$ 
3:  $c_{\text{best}}^0 \leftarrow \infty;$ 
4:  $c_{\text{update}} \leftarrow c_{\text{best}}^0;$ 
5:  $X_{\text{guide}} \leftarrow \text{PointNetGuide}(x_{\text{start}}, x_{\text{goal}}, c_{\text{best}}^0, X_{\text{free}});$ 
6: for  $i = 1$  to  $n$  do
7:    $c_{\text{best}}^i \leftarrow \min_{x_{\text{soln}} \in X_{\text{soln}}} \{\text{Cost}(x_{\text{soln}})\};$ 
8:    $x_{\text{rand}}, X_{\text{guide}}, c_{\text{update}} \leftarrow \text{PointNetGuidedSampling}(X_{\text{guide}}, x_{\text{start}}, x_{\text{goal}}, c_{\text{update}}, c_{\text{best}}^i, X_{\text{free}});$ 
9:    $x_{\text{nearest}} \leftarrow \text{Nearest}(G = (V, E), x_{\text{rand}});$ 
10:   $x_{\text{new}} \leftarrow \text{Steer}(x_{\text{nearest}}, x_{\text{rand}});$ 
11:  if  $\text{CollisionFree}(x_{\text{nearest}}, x_{\text{new}})$  then
12:     $X_{\text{near}} \leftarrow \text{Near}(G = (V, E), x_{\text{new}}, r_{\text{RRT}^*});$ 
13:     $V \leftarrow V \cup \{x_{\text{new}}\};$ 
14:     $x_{\text{min}} \leftarrow x_{\text{nearest}};$ 
15:     $c_{\text{min}} \leftarrow \text{Cost}(x_{\text{nearest}}) + c(\text{Line}(x_{\text{nearest}}, x_{\text{new}}));$ 
16:    for all  $x_{\text{near}} \in X_{\text{near}}$  do
17:      if  $\text{CollisionFree}(x_{\text{near}}, x_{\text{new}}) \wedge \text{Cost}(x_{\text{near}}) + c(\text{Line}(x_{\text{near}}, x_{\text{new}})) < c_{\text{min}}$  then
18:         $x_{\text{min}} \leftarrow x_{\text{near}};$ 
19:         $c_{\text{min}} \leftarrow \text{Cost}(x_{\text{near}}) + c(\text{Line}(x_{\text{near}}, x_{\text{new}}));$ 
20:      end if
21:    end for
22:     $E \leftarrow E \cup \{(x_{\text{min}}, x_{\text{new}})\};$ 
23:    for all  $x_{\text{near}} \in X_{\text{near}}$  do
24:      if  $\text{CollisionFree}(x_{\text{new}}, x_{\text{near}}) \wedge \text{Cost}(x_{\text{new}}) + c(\text{Line}(x_{\text{new}}, x_{\text{near}})) < \text{Cost}(x_{\text{near}})$  then
25:         $x_{\text{parent}} \leftarrow \text{Parent}(x_{\text{near}});$ 
26:         $E \leftarrow (E \setminus \{(x_{\text{parent}}, x_{\text{near}})\}) \cup \{(x_{\text{new}}, x_{\text{near}})\};$ 
27:      end if
28:    end for
29:    if  $\text{InGoalRegion}(x_{\text{new}})$  then
30:       $X_{\text{soln}} \leftarrow X_{\text{soln}} \cup \{x_{\text{new}}\};$ 
31:    end if
32:  end if
33: end for
34: return  $G=(V, E);$ 

```

C. PointNetGuidedSampling

- **Cập nhật X_{guide} :** Nếu đường đi hiện tại (C_{curr}) tốt hơn đáng kể so với lần cập nhật trước ($C_{\text{curr}} < \alpha * C_{\text{update}}$), thì gọi lại mạng nơ-ron (PointNetGuide) để cô tập X_{guide} mới. Cập nhật $C_{\text{update}} = C_{\text{curr}}$.
- **Tỷ lệ α (alpha):** Là một ngưỡng kiểm soát tần suất gọi AI, tránh gọi quá thường xuyên gây tốn kém.
- **Mixed Strategy (Chiến lược hỗn hợp):** Khi lấy mẫu x_{rand} :
 - 50% cơ hội: Lấy mẫu dùng chiến lược của IRRT* (InformedSampling - tức là lấy mẫu ngẫu nhiên bên trong ellipsoid). Điều này đảm bảo thuật toán vẫn khám phá đủ rộng và giữ được tính tối ưu tiệm cận của IRRT*.
 - 50% cơ hội: Lấy mẫu ngẫu nhiên từ tập X_{guide} (các điểm mà AI nghĩ là tốt). Điều này giúp tập trung tìm kiếm vào các vùng hứa hẹn, tăng tốc độ hội tụ.
- **Đảm bảo lý thuyết:** Vì luôn có xác suất > 0 để lấy mẫu theo kiểu IRRT*, NIRRT* kế thừa các đặc tính tốt của IRRT* (đầy đủ xác suất, tối ưu tiệm cận).

Algorithm 2 PointNetGuidedSampling($X_{\text{guide}}, x_{\text{start}}, x_{\text{goal}}, C_{\text{update}}, C_{\text{curr}}, X_{\text{free}}$)

```
1: if  $C_{\text{curr}} < \alpha C_{\text{update}}$  then
2:    $X_{\text{guide}} \leftarrow \text{PointNetGuide}(x_{\text{start}}, x_{\text{goal}}, C_{\text{curr}}, X_{\text{free}});$ 
3:    $C_{\text{update}} \leftarrow C_{\text{curr}};$ 
4: end if
5: if Rand() < 0.5 then
6:   if  $C_{\text{curr}} < \infty$  then
7:      $x_{\text{rand}} \leftarrow \text{InformedSampling}(x_{\text{start}}, x_{\text{goal}}, C_{\text{curr}});$ 
8:   else
9:      $x_{\text{rand}} \leftarrow \text{UniformSampling}(X_{\text{free}});$ 
10:  end if
11: else
12:    $x_{\text{rand}} \leftarrow \text{UniformSampling}(X_{\text{guide}});$ 
13: end if
14: return  $x_{\text{rand}}, X_{\text{guide}}, C_{\text{update}};$ 
```

D. Point-based Network Guidance

- **Khởi tạo:** Đặt điểm start/goal cho Neural Connect, khởi tạo X_{guide} rỗng.
- **Neural Focus & Tạo Input:**
 - Nếu đã có giải pháp ($c_{\text{curr}} < \infty$), chỉ lấy mẫu điểm (X_{input}) từ vùng giao của ellipsoid (X_{focus}) và không gian trống (X_{free}).
 - Nếu chưa có giải pháp, lấy mẫu điểm từ toàn bộ không gian trống.
- **Vòng lặp Neural Connect (tối đa n_{guide} lần):**

- **Chuẩn bị Input cho Mạng:** ADD One heat và chuẩn hóa Input
- **Gọi PointNet++:** Đưa X_{input} vào mạng, lấy các điểm dự đoán là guidance và thêm vào X_{guide} .
- **Kiểm tra Kết nối (BFS xuôi và ngược):** Chạy BFS trong X_{guide} từ x_{start}^i đến x_{goal}^i .
 - Nếu kết nối -> **Xong!** Trả về X_{guide} .
 - Nếu không, lấy điểm biên x_{start}^{i+1} .
- (Cập nhật x_{start}^i và x_{goal}^i cho vòng lặp tiếp theo).

Algorithm 3 PointNetGuide($x_{\text{start}}, x_{\text{goal}}, c_{\text{curr}}, X_{\text{free}}$)

```
1:  $x_{\text{start}}^1 \leftarrow x_{\text{start}};$ 
2:  $x_{\text{goal}}^1 \leftarrow x_{\text{goal}};$ 
3:  $X_{\text{guide}} \leftarrow \emptyset;$ 
4: if  $c_{\text{curr}} < \infty$  then
5:    $X_{\text{focus}} \leftarrow \text{InformedSubset}(x_{\text{start}}, x_{\text{goal}}, c_{\text{curr}});$ 
6:    $X_{\text{input}} \leftarrow \text{PointCloudSampling}(X_{\text{focus}} \cap X_{\text{free}});$ 
7: else
8:    $X_{\text{input}} \leftarrow \text{PointCloudSampling}(X_{\text{free}});$ 
9: end if
10: for  $j = 1$  to  $n_{\text{guide}}$  do
11:    $\tilde{X}_{\text{input}} \leftarrow \text{AddOneHotFeatures}(X_{\text{input}}, x_{\text{start}}^j, x_{\text{goal}}^j);$ 
12:    $\tilde{X}_{\text{input}} \leftarrow \text{NormalizeCoordinates}(\tilde{X}_{\text{input}});$ 
13:    $X_{\text{guide}} \leftarrow X_{\text{guide}} \cup \text{PointBasedNetwork}(\tilde{X}_{\text{input}});$ 
14:    $\text{connectivity}, x_{\text{start}}^{j+1} \leftarrow \text{BFS}(X_{\text{guide}}, x_{\text{start}}, x_{\text{goal}}, \eta);$ 
15:   if  $\text{connectivity}$  then
16:     return  $X_{\text{guide}};$ 
17:   end if
18:    $\text{connectivity}, x_{\text{goal}}^{j+1} \leftarrow \text{BFS}(X_{\text{guide}}, x_{\text{goal}}, x_{\text{start}}, \eta);$ 
19:   if  $\text{connectivity}$  then
20:     return  $X_{\text{guide}};$ 
21:   end if
22: end for
23: return  $X_{\text{guide}};$ 
```

- Neural Focus - Informed Subnet

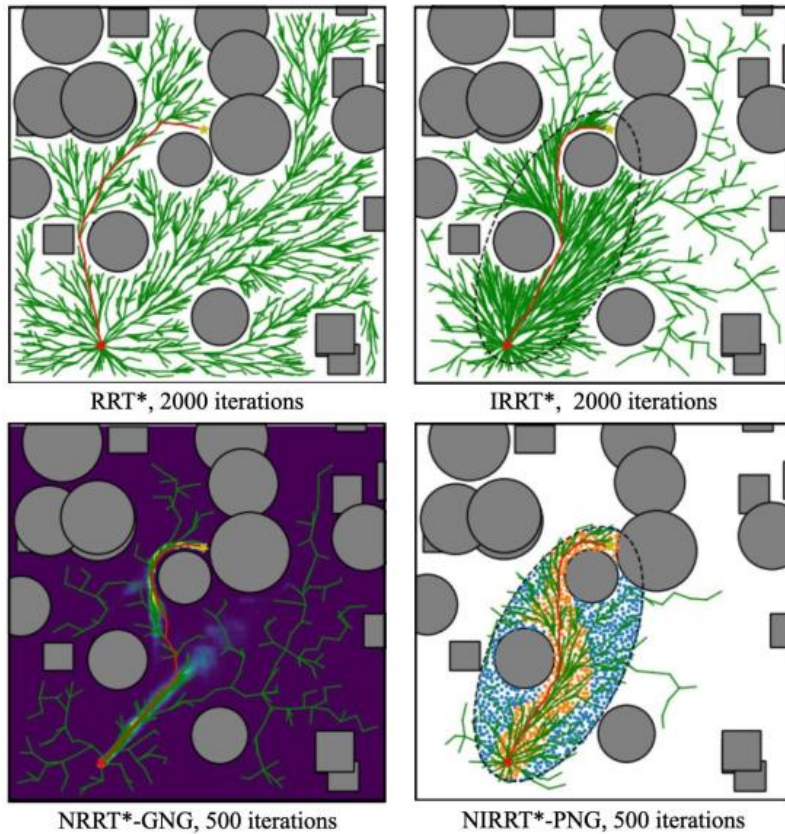
$$X_{\text{focus}} = \{x \in X \mid \|x - x_{\text{start}}\|_2 + \|x - x_{\text{goal}}\|_2 \leq c_{\text{curr}}\} \quad (3)$$

- Pointnet++

- PointNet++ nhận X_{input} (tập N điểm, mỗi điểm có tọa độ chuẩn hóa + one-hot vector).
- Nó xử lý cấu trúc không gian của đám mây điểm và mối quan hệ giữa các điểm.
- Đầu ra là N xác suất p_1, \dots, p_N , mỗi p_i ứng với điểm x_i . p_i thể hiện "niềm tin" của mạng rằng điểm x_i nằm trên hoặc gần đường đi tối ưu.
- **X_guide**: Tập hợp các điểm x_i có xác suất $p_i > 0.5$. Đây là những điểm được AI "đề cử" để dẫn đường cho việc lấy mẫu của RRT*.

IV. Experiments

2D Environment



3D Environment

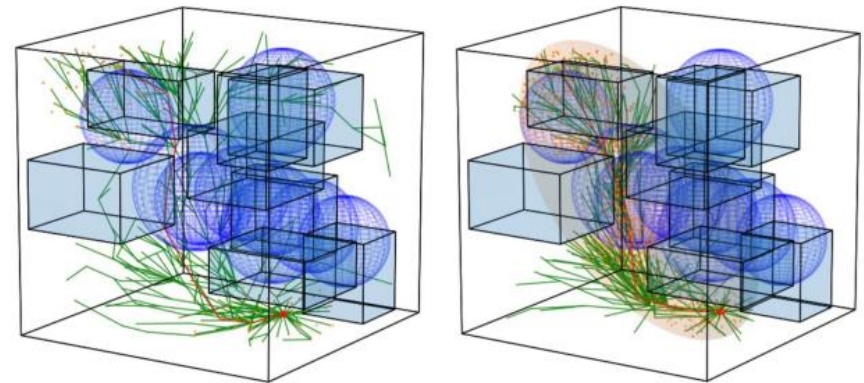


Fig. 6: Visualization on planning in 3D random world at 500 iterations. Left: NRRT*-PNG. Right: NIRRT*-PNG(FC).

A decorative graphic on the left side of the slide. It features a dark blue background with a large, stylized circular pattern composed of many small red dots. The dots are arranged in concentric, slightly irregular rings, creating a sense of depth and movement. The word "HUST" is centered within this pattern.

HUST

THANK YOU !