北京航空航天大學
BEIHANG UNIVERSITY

LITERATURE READING

Graph Neural Networks for Decentralized Multi-Robot Path Planning

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2020.

Jinjie LI

School of Automation Science and Electrical Engineering
Beihang University

Oct. 24, 2020

Outline

Outline

- ❑ Abstract
- ❑ Introduction
- ❑ Problem Statement
- ❑ Architecture
- ❑ Performance Evaluation
- ❑ Discussion and Future Work

Abstract

Problems:

- **Decentralized** multi-robot path planning
- Effective communication

Constraints:

- Only **local** communication and **local** observations
- Constrained grid world

Methods:

- Convolutional Neural Network (CNN): **Extracts** features from local observations
- Graph Neural Network (GNN): **Communicates** features among robots locally
- The dataset is generated by an centralized expert algorithm

Results:

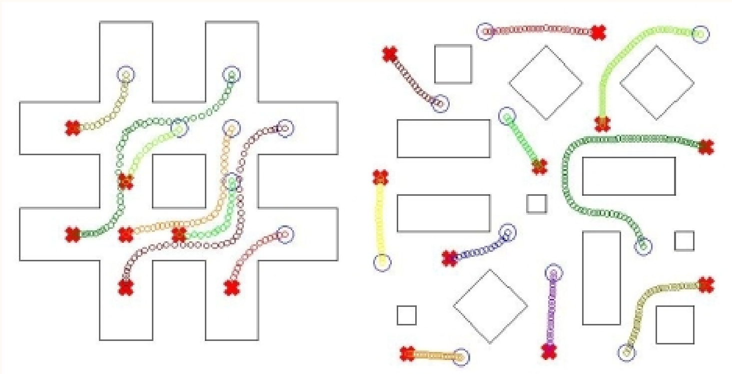
- Metrics: *Success rates* and *Flowtime Increase*
- A performance close to expert algorithm
- Generalization: larger environments and larger robot teams

Outline

- ❑ Abstract
- ❑ Introduction
- ❑ Problem Statement
- ❑ Architecture
- ❑ Performance Evaluation
- ❑ Discussion and Future Work

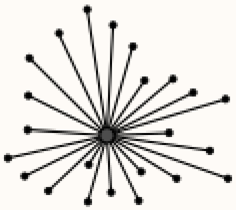
Introduction

- Multi-Robot Path Planning (MRPP)

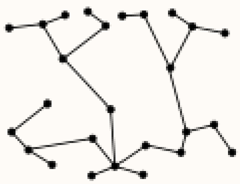


- Collision-free
- Effective

- Coupled (Centralized) or decoupled (Decentralized) method



- Ensure the optimality and completeness
- Too much calculation for large number of robots



- Sub-optimal and incomplete solutions
- Reduce the computational complexity

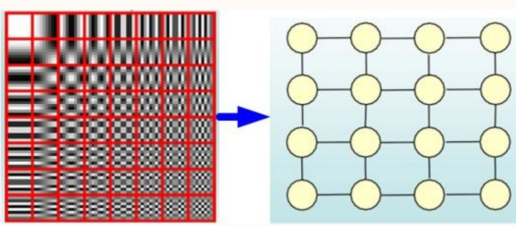
Introduction

Learning-based method

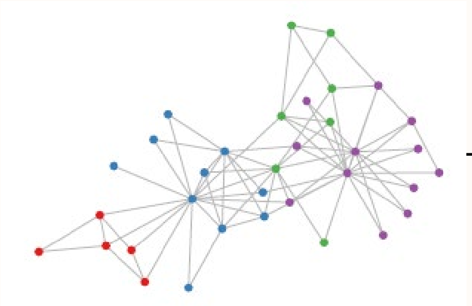
The rise of artificial intelligence：

- 计算资源的快速发展（如GPU）
- 大量训练数据的可用性
- 深度学习从欧氏空间数据中提取潜在特征的有效性

Computer Vision, Natural Language Processing



Euclidean Structure



Non-Euclidean Structure

→ GNN 图神经网络

↓

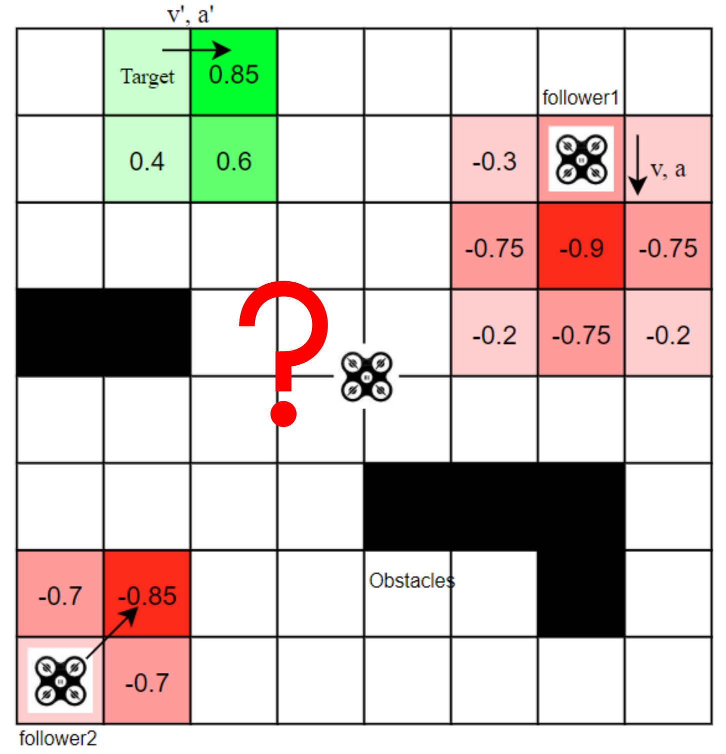
Multi-agent problem (decentralized method)

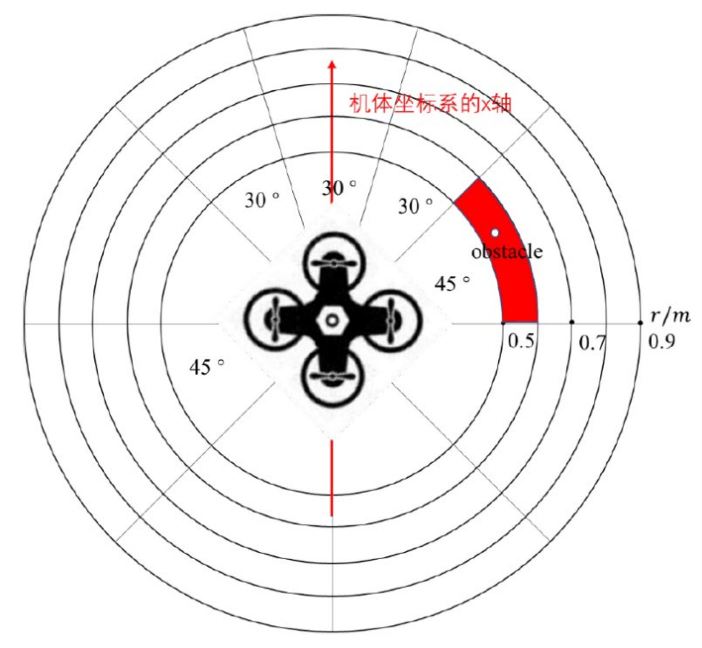
Cambridge: Li Q, Gama F, Ribeiro A, et al. Graph neural networks for decentralized multi-robot path planning[J]. arXiv preprint arXiv:1912.06095, 2019.

Upenn: Arbaaz Khan, Ekaterina Tolstaya, Alejandro Ribeiro, and Vijay Kumar. 2020. Graph policy gradients for large scale robot control.

Introduction

Problem1： It is far from obvious **what** information is crucial to the task at hand, and **how** and **when** it must be shared among robots.



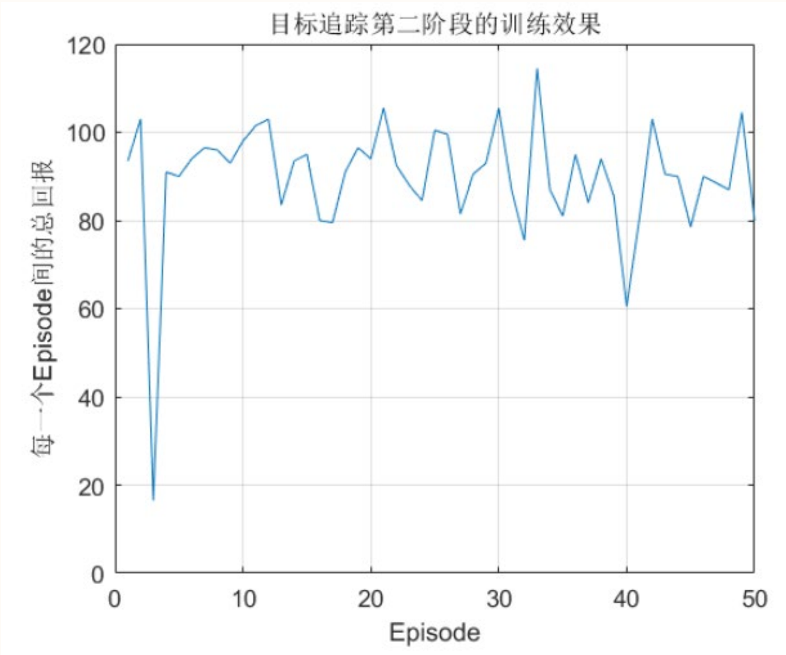


User-defined information

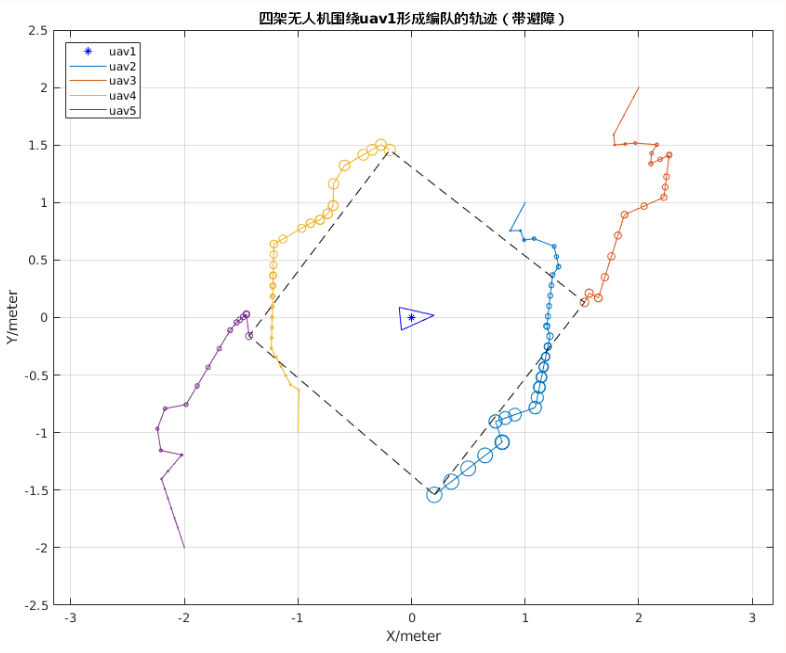
Disadvantages：(1) 传递信息有限 (2) 无法描述不规则的障碍信息

Introduction

Problem2： Reinforcement learning process is very blind and inefficient in the process of exploration.



Total Reward in Training Process

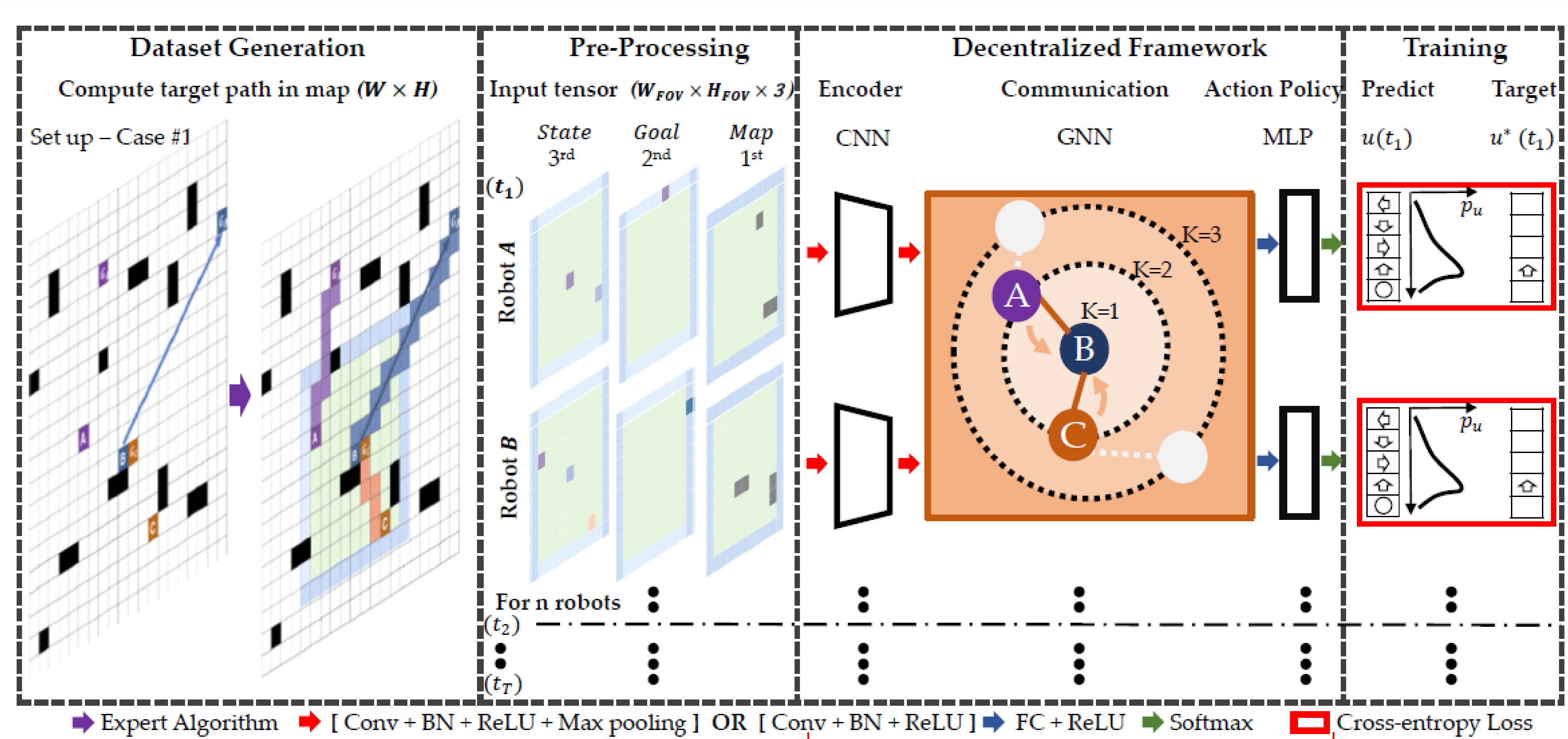


Formation Trajectories

Disadvantages：(1) 物理世界随机探索的效率太低。(2) 无法解释最终结果是不是比成熟算法更好。

Introduction

The Proposed Architecture



Sufficient Communication

Problem1: A CNN that extracts adequate features

Problem2: Supervised learning

Outline

- ❑ Abstract
- ❑ Introduction
- ❑ Problem Statement
- ❑ Architecture
- ❑ Performance Evaluation
- ❑ Discussion and Future Work

Problem Statement

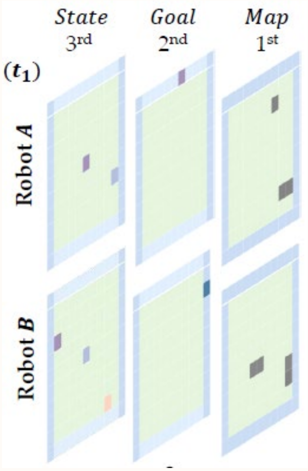
Problem Formulation:

Let $V = \{v_1, \ldots, v_N\}$ be the set of N robots.

Observation

At time t, each robot perceives its surroundings within a given field of vision.

This map perceived by robot i is denoted by $Z_t^i \in R^{W_{FOV} \times H_{FOV}}$



Each robot has access 128 observations $\tilde{x}_t^i \in R^{128}$. ⟵ CNN ⟵

Communication network: $G_t = (V, \varepsilon_t, W_t)$

V: the set of robots

$\varepsilon_t \subseteq V \times V$:the set of edges

Communication radius: r_{COMM}. If $\|p_i - p_j\| \leq r_{COMM}$, robots can communicate.

An adjacency matrix $S_t \in R^{N \times N}$, where $[S_t]_{ij} = s_t^{ij} = 1$ only if $(v_j, v_i) \in \varepsilon_t$.

Problem Statement

Problem Formulation:

Objective:

To learn a mapping $\mathcal{F}$, $u_t = \boxed{\mathcal{F}\left(\{Z_t^i\}, G_t\right).}$

For each robot:

Input: observations $\{Z_t^i\}$, and communication graph G_t

Output: an action u_t

(1) Shortest possible time, avoiding collisions with other robots and obstacles.

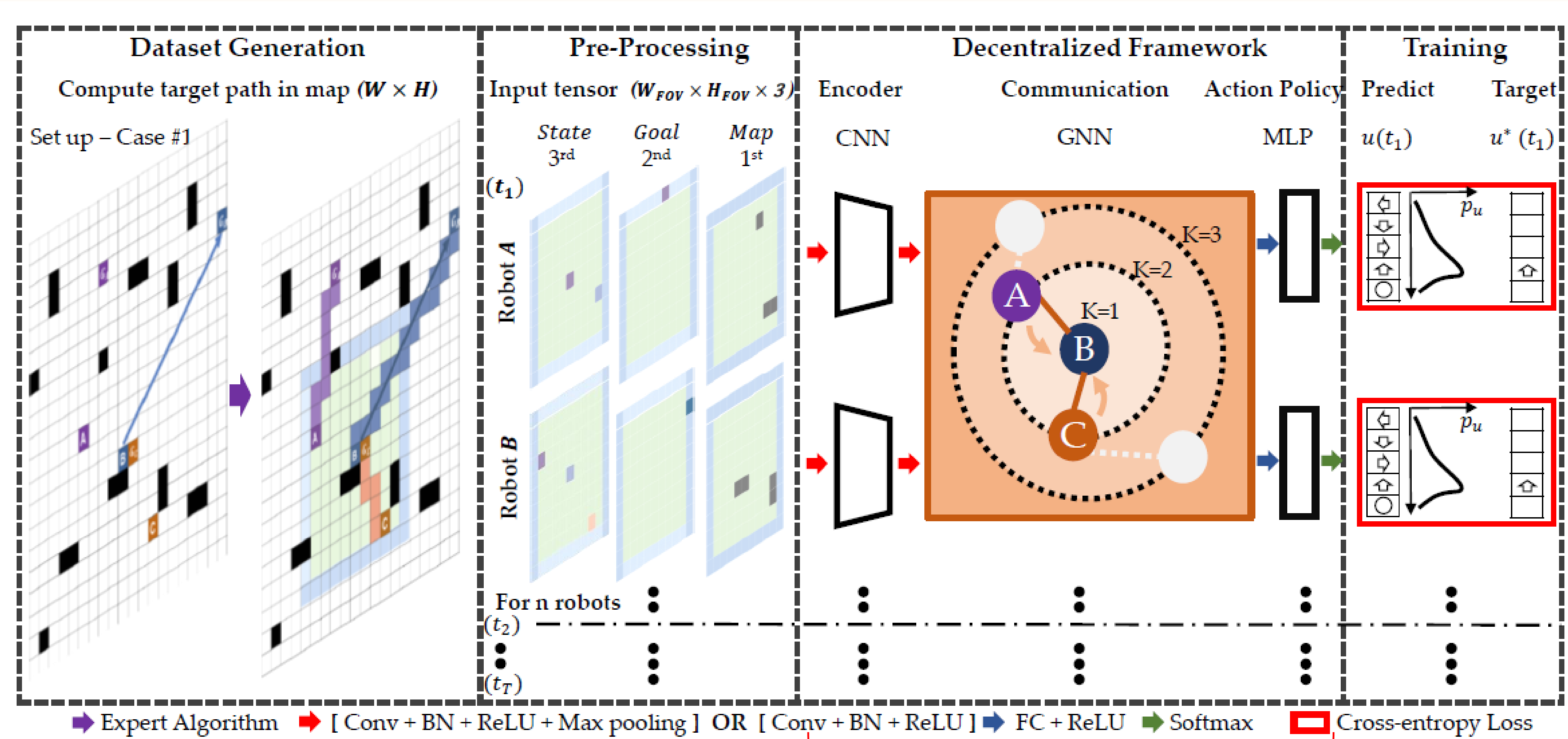
(2) To perform as well as a coupled centralized expert.

Outline

- ❑ Abstract
- ❑ Introduction
- ❑ Problem Statement
- ❑ Architecture
- ❑ Performance Evaluation
- ❑ Discussion and Future Work

Introduction

The Proposed Architecture



Sufficient Communication

Problem1: A CNN that extracts adequate features

Problem2: Supervised learning

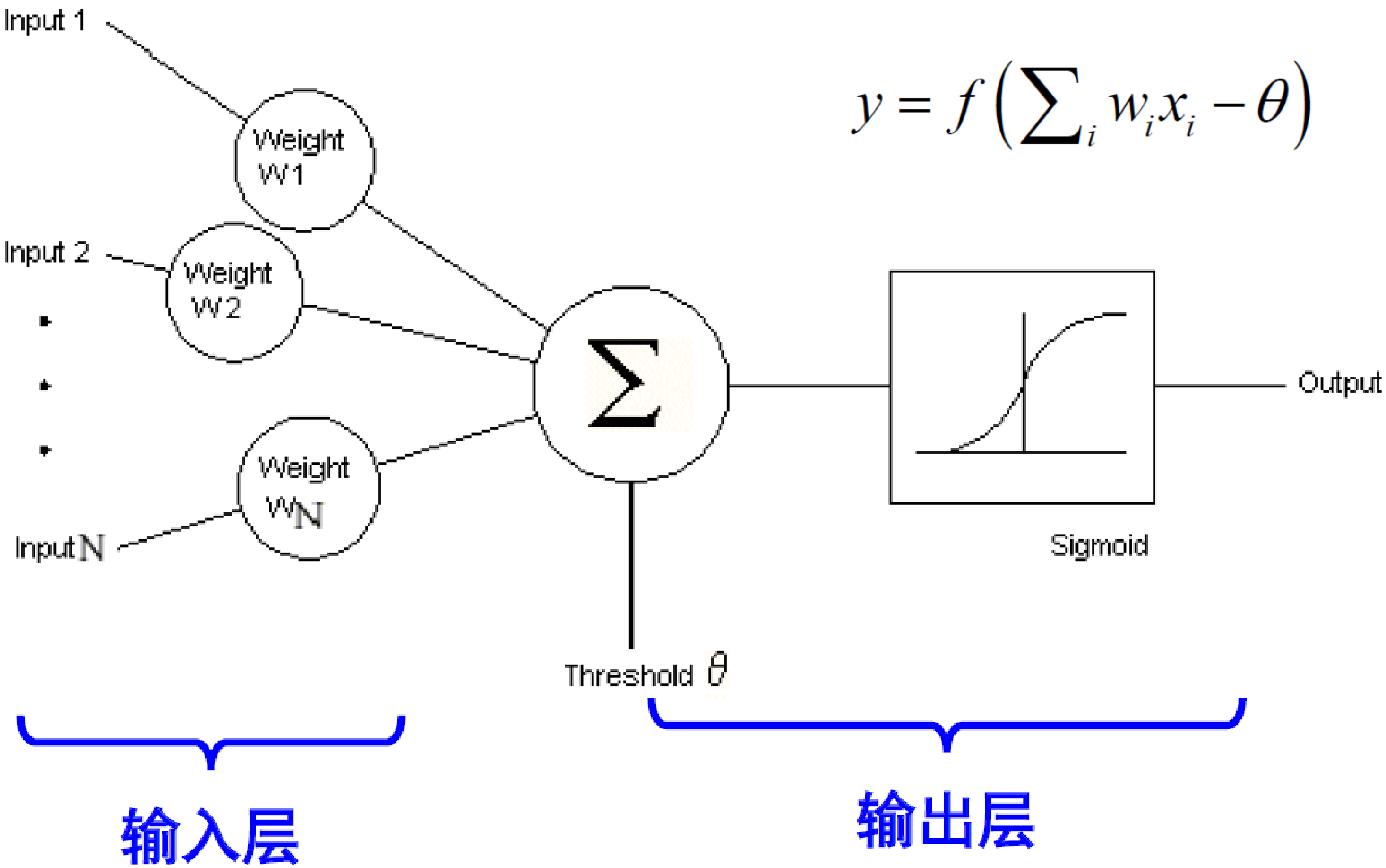
CNN

What is the CNN (Convolutional neural network)?

And what does the CNN do?

单层感知机

包含两层神经元：（1）输入层（信号传递）
（2）输出层（M-P神经元，threshold logic unit）



http://homepages.gold.ac.uk/nikolaev/311perc.htm

感知机如何学习？

对于给定的训练数据集 (x, y)

若当前感知机的输出为 $\hat{y} \longleftarrow y = f\left(\sum_i w_i x_i - \theta\right)$

则感知机将根据误差对权重做如下调整：

$$w_i \leftarrow w_i + \Delta w_i$$

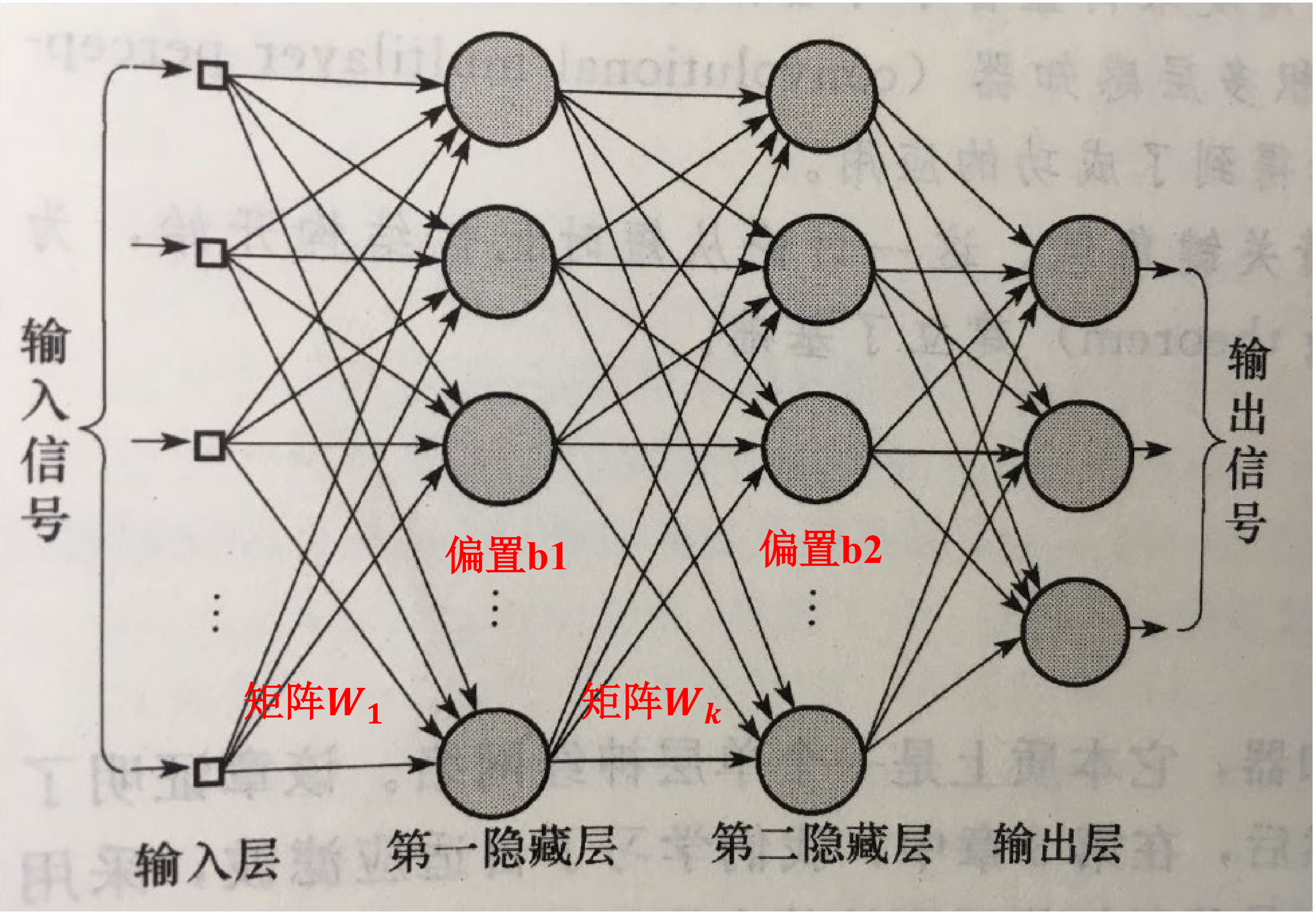
$$\Delta w_i = \eta\left(y - \hat{y}\right) x_i$$

其中 $\eta \in (0,1)$ 称为学习率（learning rate）

思考：如何保证学习过程的收敛与效率？

https://www.mathworks.com/help/nnet/ug/perceptron-neural-networks.html

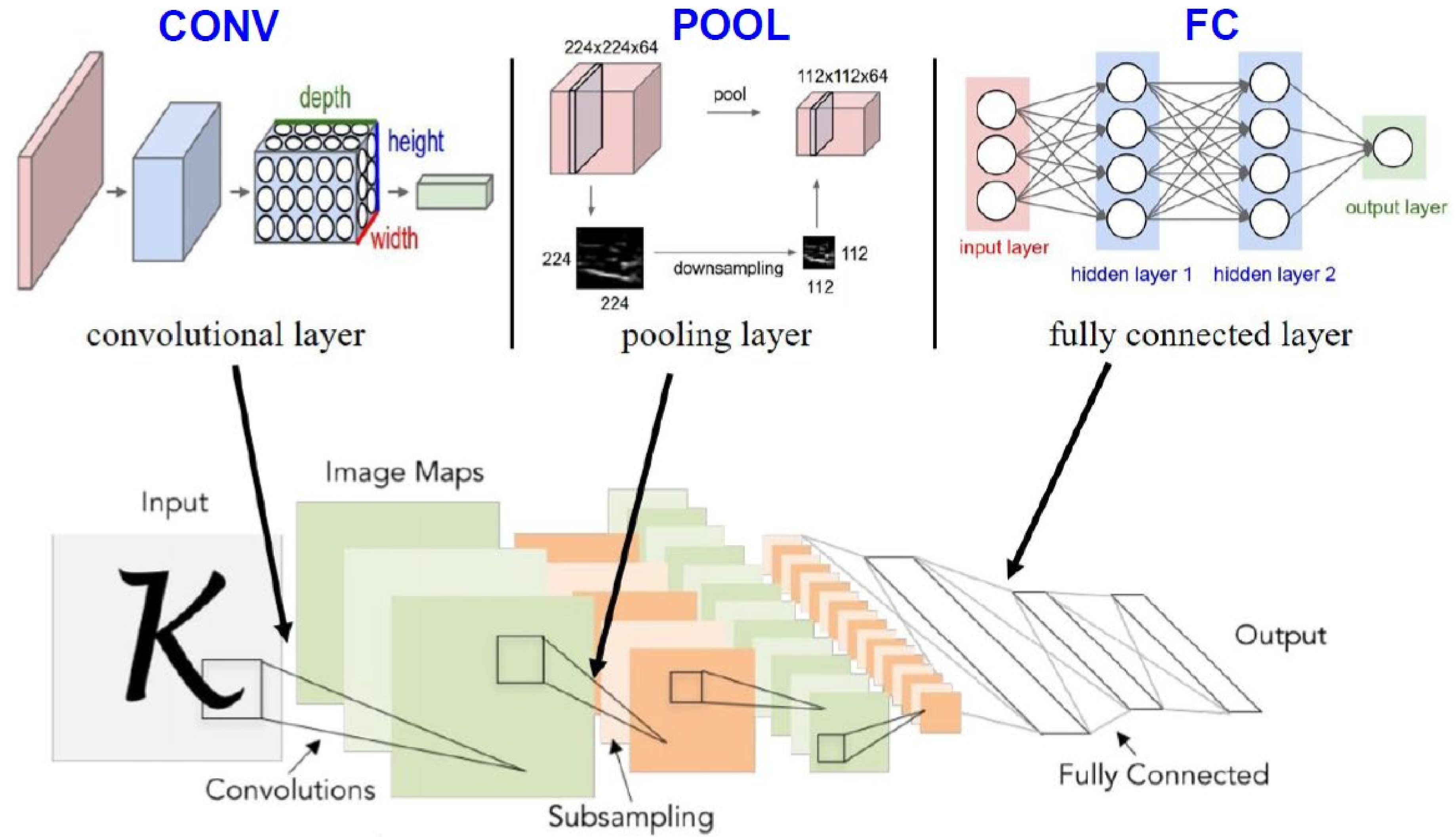
如何训练多层感知机



《神经网络与机器学习》第78页，Simon Haykin

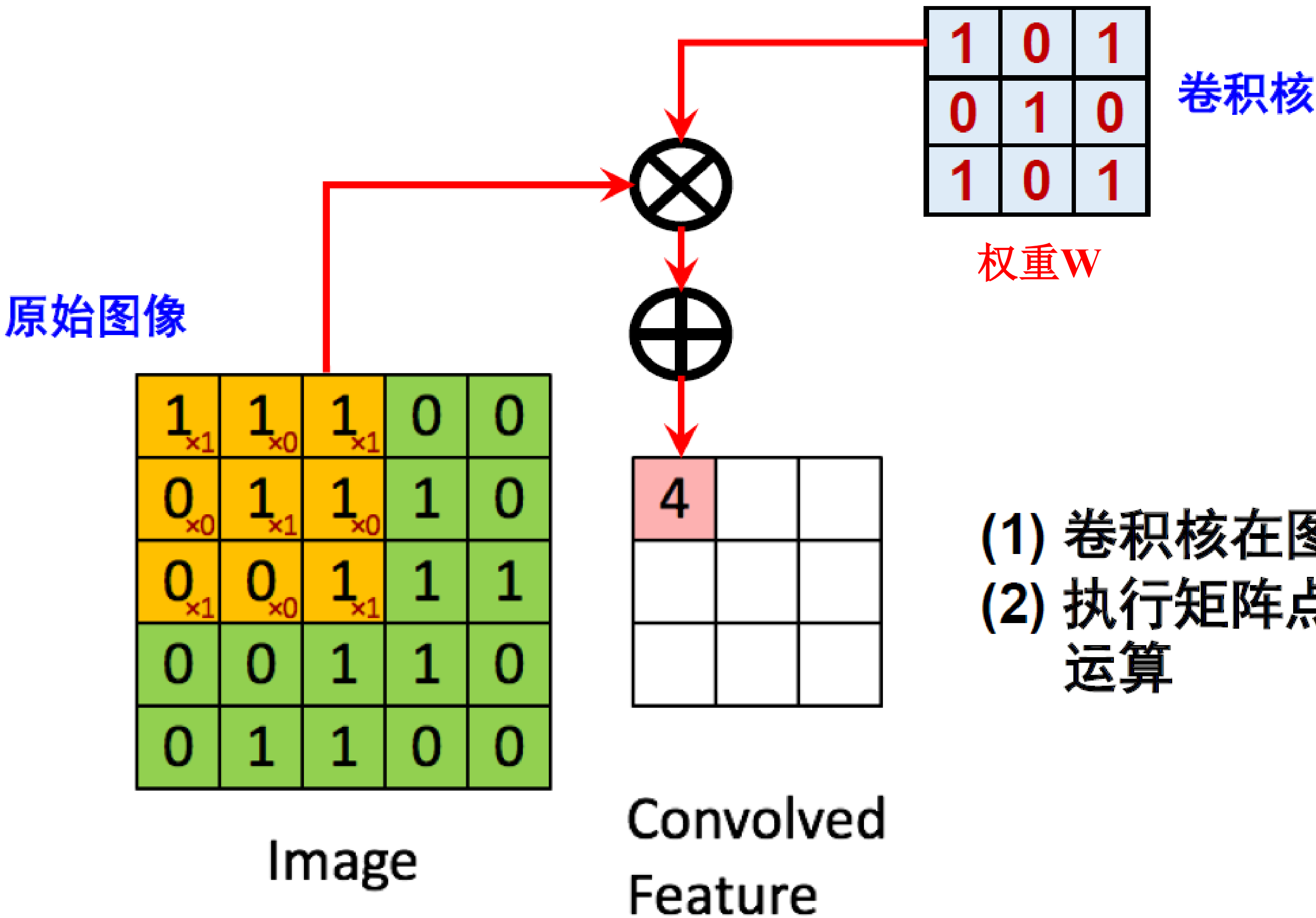
卷积神经网络
Convolutional Neural Network

初识CNN结构



The first successful applications of Convolutional Networks: **LeNet-5**

如何做卷积运算

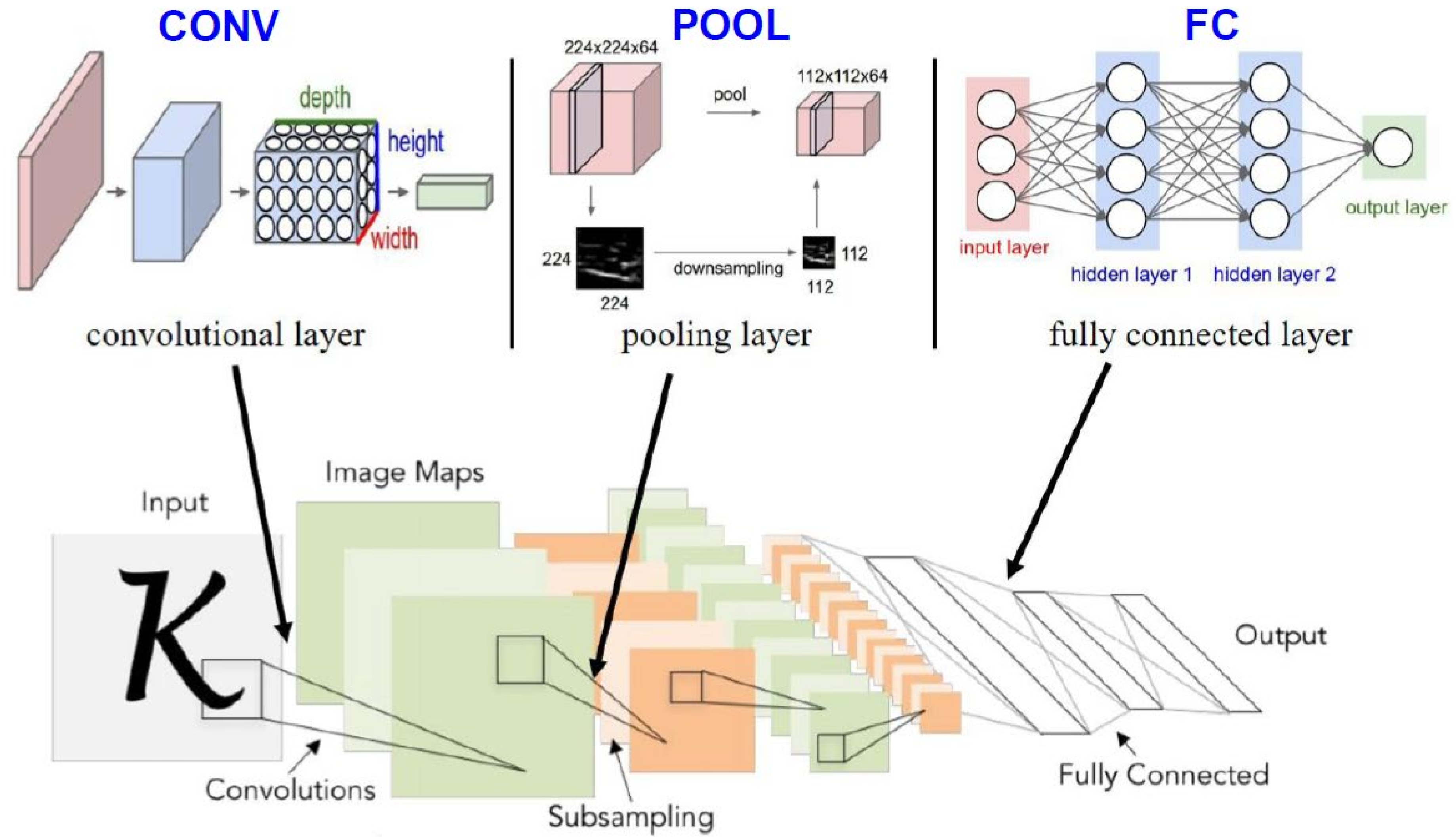


(1) 卷积核在图像上平移

(2) 执行矩阵点乘、求和运算

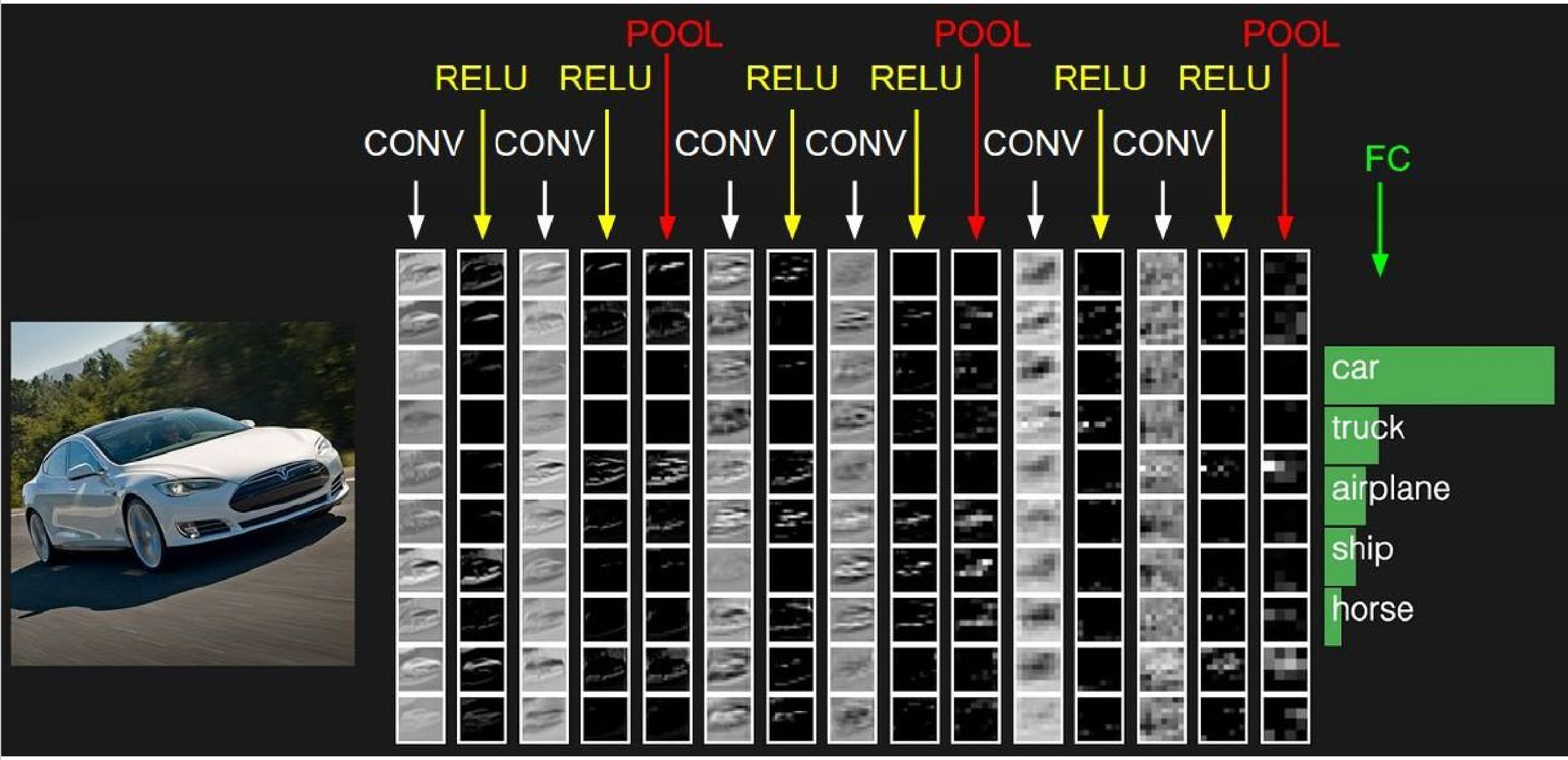
http://ufldl.stanford.edu/tutorial/supervised/FeatureExtractionUsingConvolution/

初识CNN结构



The first successful applications of Convolutional Networks: **LeNet-5**

全连接层（FC）


POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

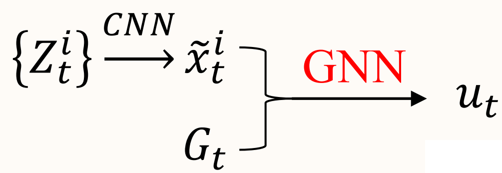
Graph Neural Network

What is the GNN (Graph Neural Network)?

Graph Neural Network

Objective:

To learn a mapping $\mathcal{F}$, $u_t = \mathcal{F}(\{Z_t^i\}, G_t)$.

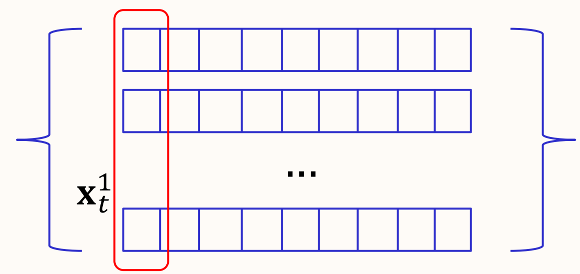


Graph Convolutions (类比于$wx + b$)

$$y = f\left(\sum_i w_i x_i - \theta\right)$$

$F(128)$ observations for robot i in time t: $\tilde{x}_t^i, i = 1, \ldots, N$

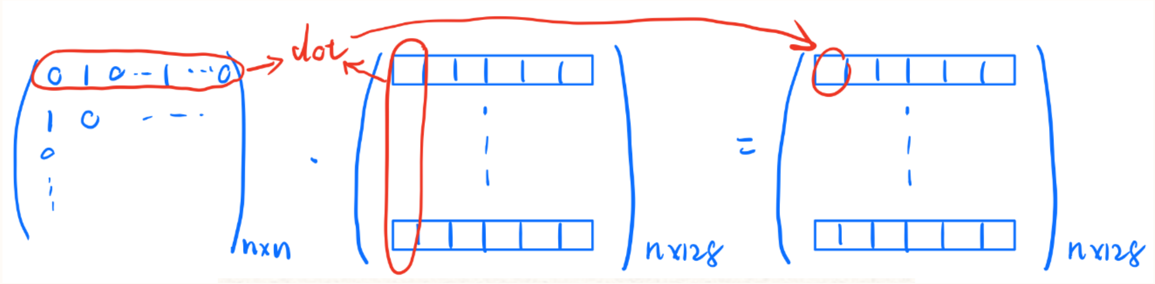
$$\mathbf{X}_t = \begin{bmatrix} (\tilde{\mathbf{x}}_t^1)^\top \\ \vdots \\ (\tilde{\mathbf{x}}_t^N)^\top \end{bmatrix} = [\mathbf{x}_t^1 \quad \cdots \quad \mathbf{x}_t^F]$$



To formally describe the communication between neighboring agents, we need adjacency matrix in time t: S_t, and the operation $S_t X_t$.

Graph Convolutions

$$[S_t X_t]_{if} = \sum_{j=1}^{N} [S_t]_{ij} \, [X_t]_{jf} = \sum_{j: v_j \in N_i} s_t^{ij} \, x_t^{jf}$$



where $N_i = \{v_j \in V : (v_j, v_i) \in \mathcal{E}_t\}$ is the set of nodes v_j that are neighbors of v_i. The linear operation $S_t X_t$ is essentially shifting the values of X_t through the nodes.

Then we can define a **graph convolution** as linear combination of shifted versions of the signal:

$$\mathcal{A}(X_t; S_t) = \sum_{k=0}^{K-1} S_t^k \, X_t A_k$$

$\{A_k\}$: is similar to the weight matrix in DNN.

K: k-hop nodes

Graph Neural Network

Graph Convolutions

$$\mathcal{A}(X_t; S_t) = \sum_{k=0}^{K-1} S_t^k X_t A_k$$

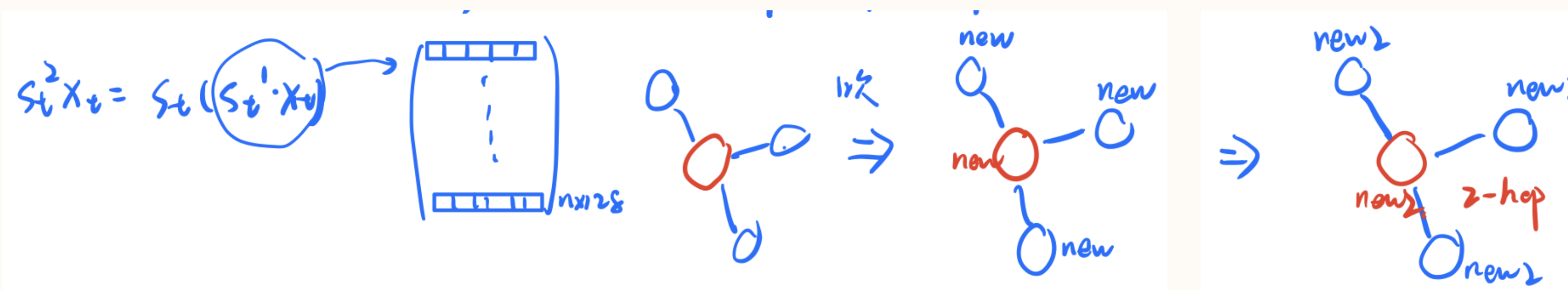
三点解释

1. 左乘 $S_t^k X_t$：与graph中的拓扑关系一致，表明不同节点之间的联系

   右乘 $X_t A_k$：值是任意的，表示同一个节点特征的线性组合，在不同节点之间构建了权值共享机制(weight sharing scheme)。

2. 如何针对*K-hop*计算$S_t^k X_t$：

$$S_t^k X_t = S_t\left(S_t^{k-1} X_t\right)$$



看似计算了k次周围邻1节点的结果，实际计算的是*k-hop*节点的结果。

Graph Neural Network

Graph Convolutions

$$\mathcal{A}(X_t; S_t) = \sum_{k=0}^{K-1} S_t^k \, X_t A_k$$

3. 分布式计算 For each robot:

(1) $[S_t X_t]_{if} = \sum_{j=1}^{N} [S_t]_{ij} \, [X_t]_{jf} = \sum_{j: v_j \in N_i} s_t^{ij} \, x_t^{jf}$

(2) $\mathcal{A}(X_t; S_t)_i = \sum_{k=0}^{K-1} [S_t^k X_t]_{if} A_k$

Graph Neural Network

$$X_\ell = \sigma[\mathcal{A}_\ell(X_{\ell-1}; S)] \quad for \quad \ell = 1, \dots, L$$

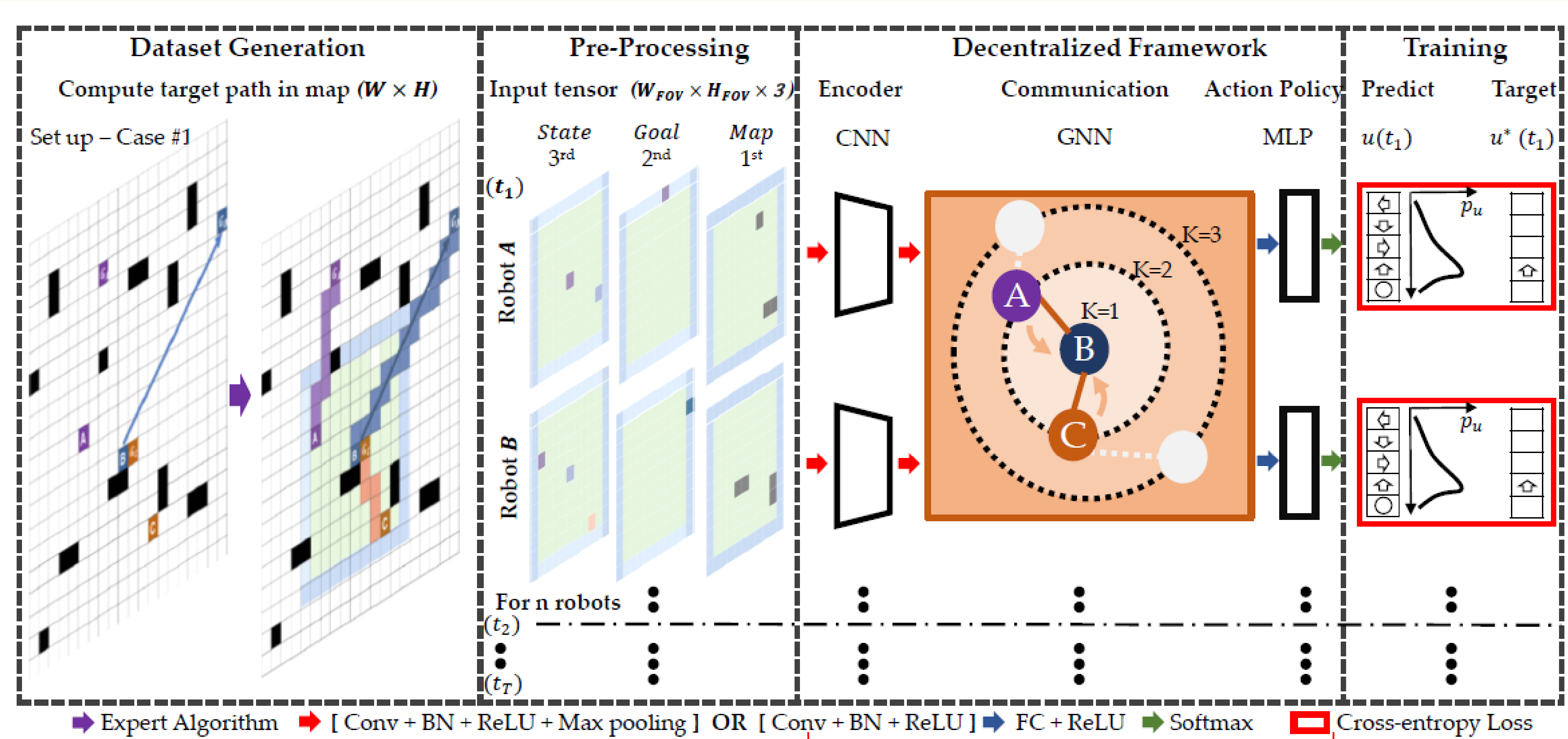
σ： activation function. σ is applied to each element of the matrix $A_\ell(X_{\ell-1};\ S)$

The final learning target： $\{A_{\ell k}\}_{k=0}^{K-1}$

The backward process is similar to CNN.

Architecture

The Proposed Architecture



CNN:
Conv2d-BatchNorm2d-ReLU-MaxPool2d and
Conv2d-BatchNorm2d-ReLU blocks 3times

GNN: 1 layer F=128
MLP:
128 input, 5 output (actions)

Cross-entropy
Loss

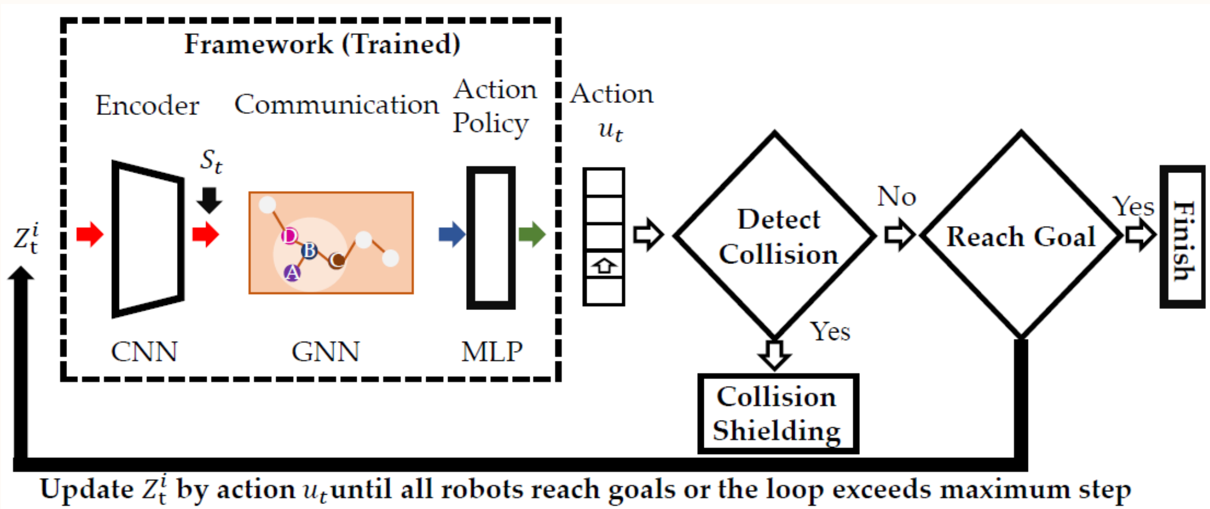
Architecture

Training: Learning from Expert Data

Training set: $\mathcal{T} = \{(\{\mathbf{Z}_t^i\}, \{\mathbf{U}_t\})\}$ $\{\mathbf{U}_t\}$: an optimal trajectory of actions

Generate random obstacles, start positions and goal positions. The optimal paths in every map are generated by an expert algorithm: Conflict-Based Search (CBS).

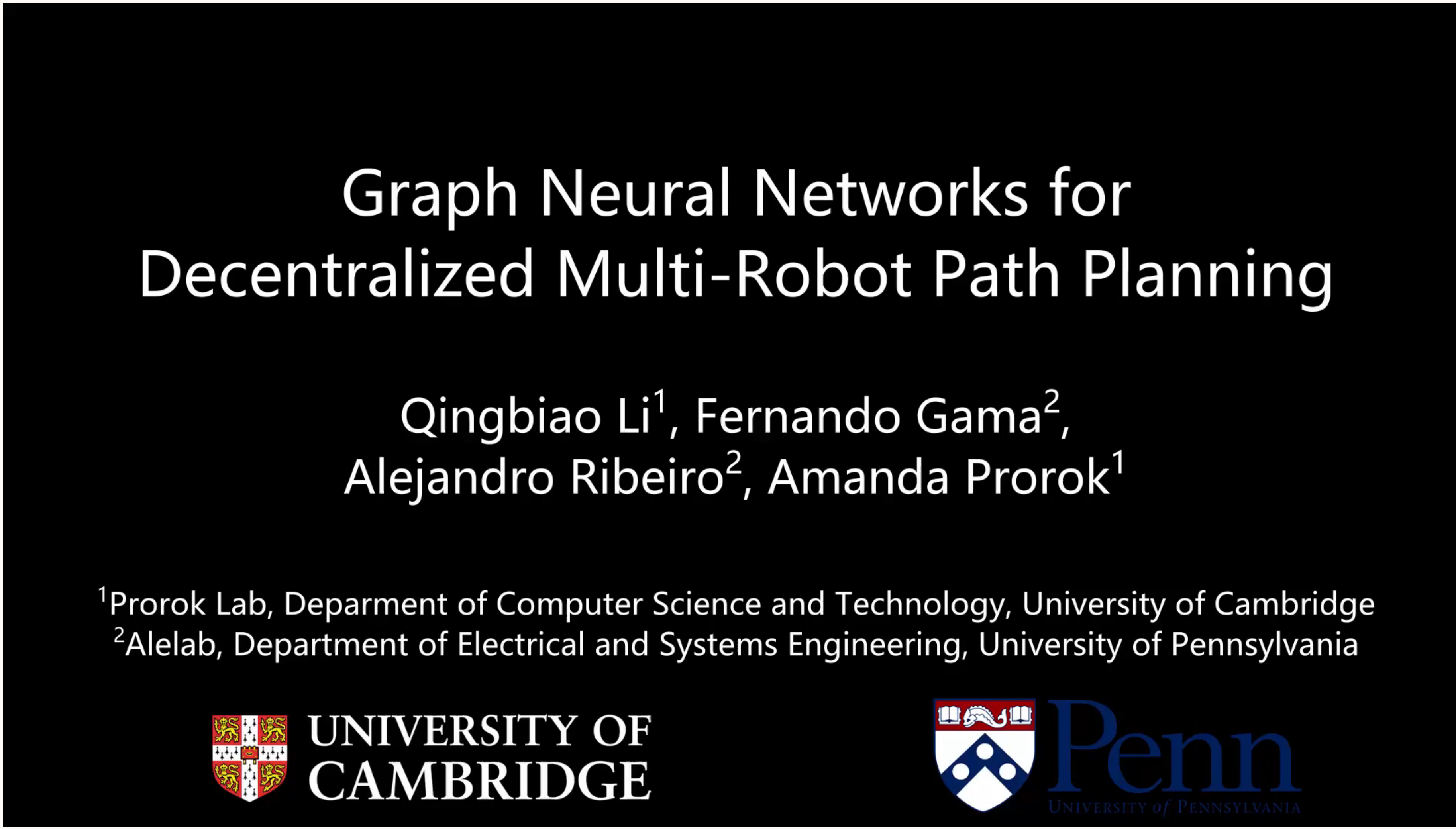
Inference stage



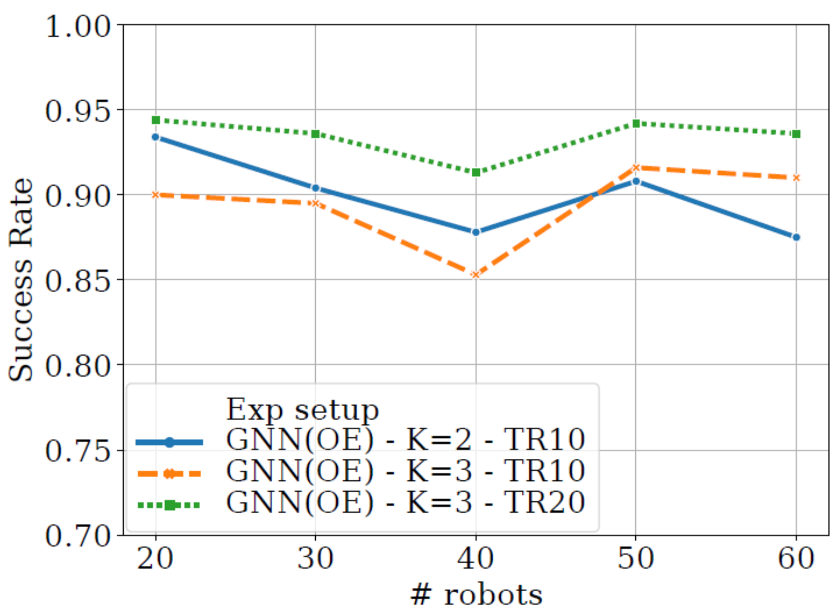
Outline

- ❑ Abstract
- ❑ Introduction
- ❑ Problem Statement
- ❑ Architecture
- ❑ Performance Evaluation
- ❑ Discussion and Future Work

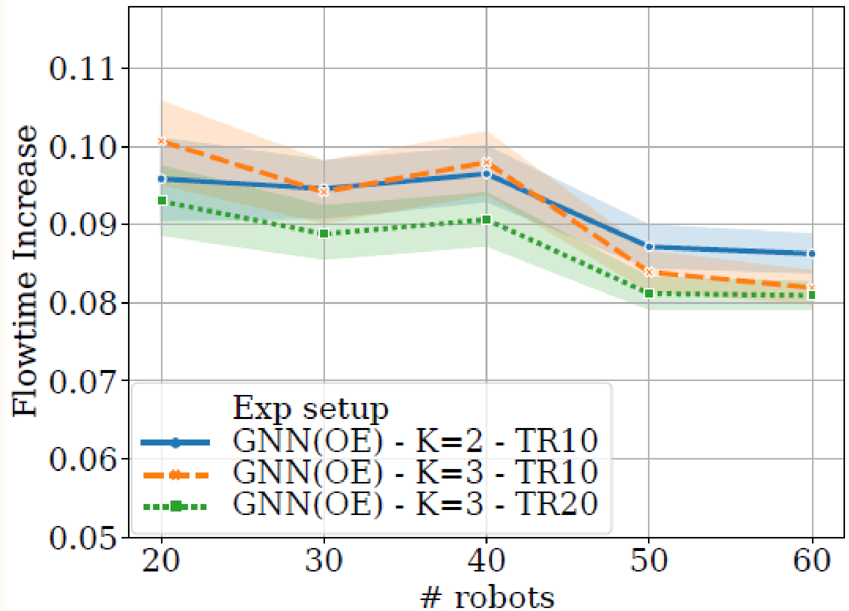
Performance Evaluation



Performance Evaluation



a: Success rate (α)



b: Flowtime increase (δ_{FT})

Outline

- ❑ Abstract
- ❑ Introduction
- ❑ Problem Statement
- ❑ Architecture
- ❑ Performance Evaluation
- ❑ Discussion and Future Work

Discussion and Future Work

Discussion

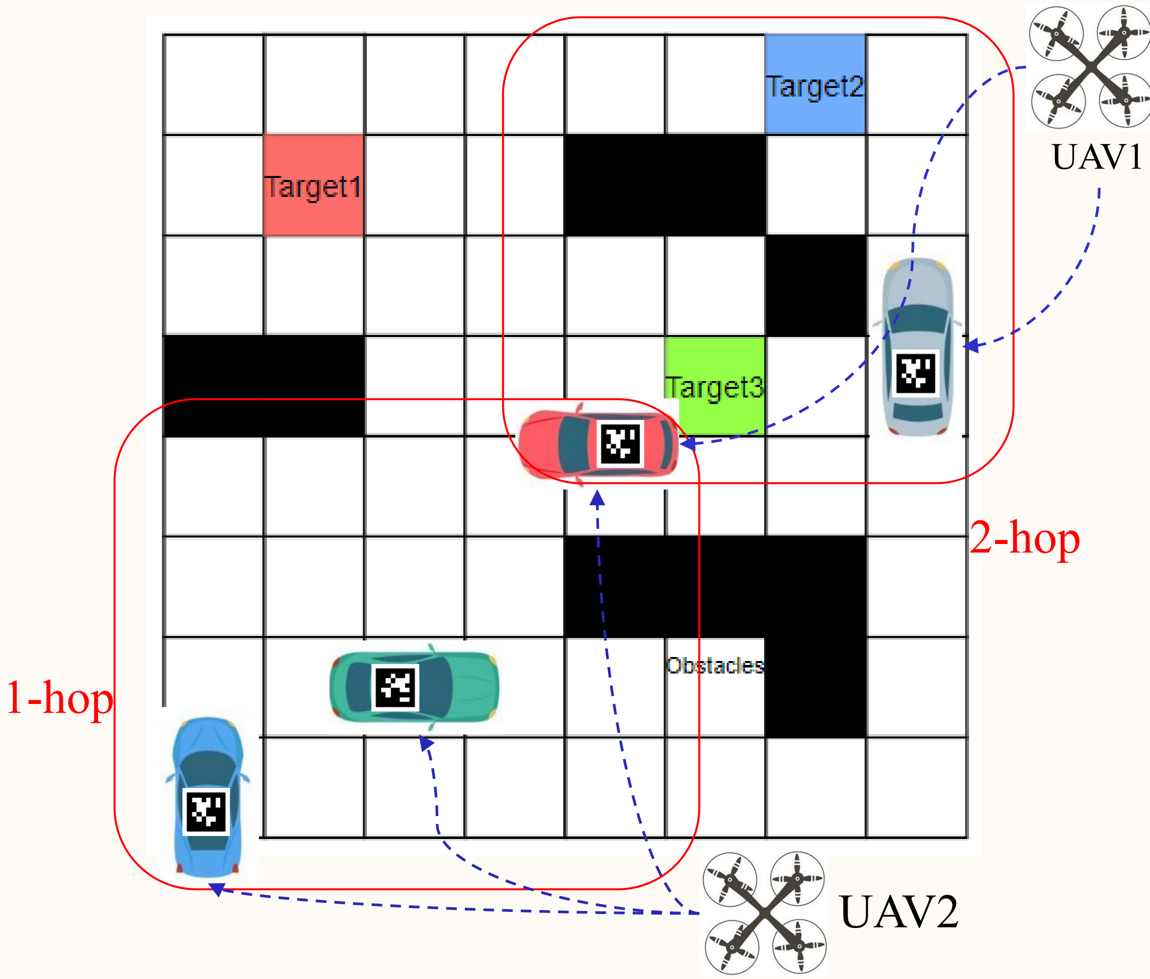
1. 与专家算法相比，时间上有优势。
2. 较好的泛化性。
3. 在更多数量的机器人群中训练的算法更好。

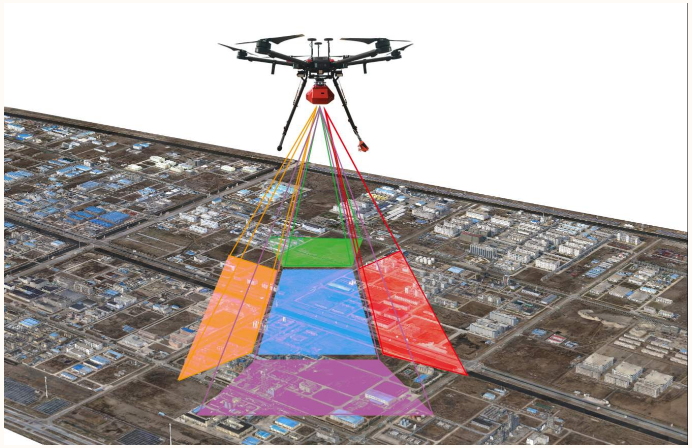
Future Work

1. Time-delayed aggregation GNNs, inter-robot live-locks and position swaps
2. 实物应用：空地协同，无人机提供视野，小车作为节点进行动作。

Future Work

实物应用：空地协同，无人机提供视野，小车作为节点进行计算与动作。





Other Resources

- Code： *https://github.com/proroklab/gnn_pathplanning*
- Simulation demo: *https://www.youtube.com/watch?v=AGDk2RozpMQ&featu re=youtu.be*
- A website about Multi-Agent Path Finding (MAPF) problem: *http://mapf.info/*

Thanks for your attention!

Q&A