Parallel Learning Theory and Its Applications in Automated Vehicles

Li Li

Associate Professor
Tsinghua University
Email: li-li@tsinghua.edu.cn

June, 2017

A.0. Outline

A.1. Application in Parking Trajectory Planning

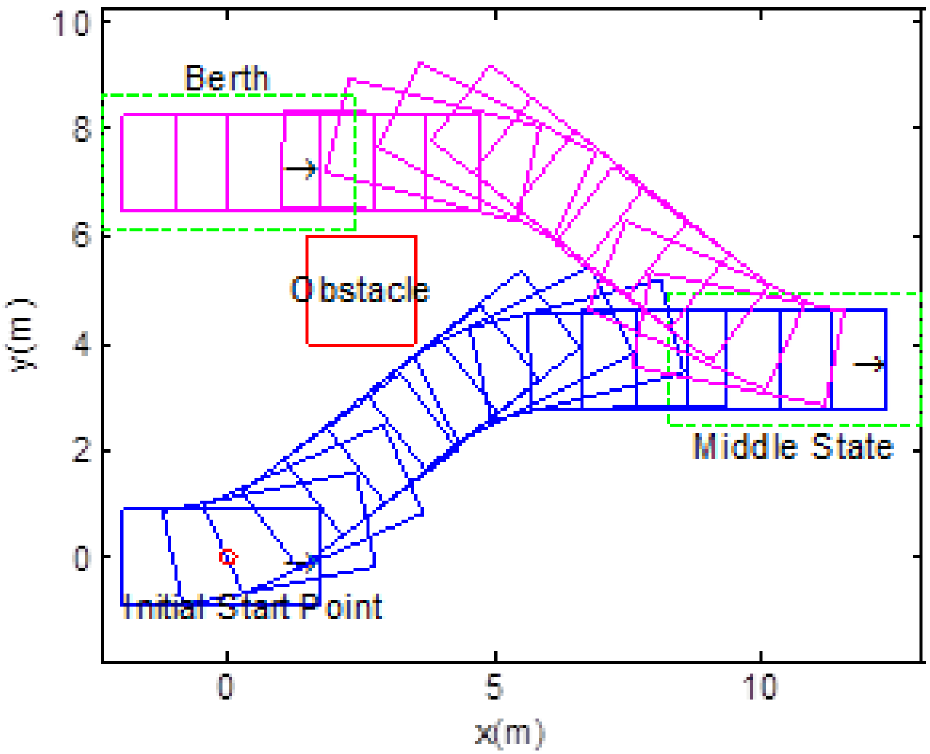
A.2. Parallel Learning Theory

A.3. Application in Virtual Test of Automated Vehicles

A.4. Conclusions

A.1. Application in Trajectory Planning

Trajectory planning: Finding a valid trajectory (between the initial and final states) along which an autonomous vehicle can track.



A.1. Application in Trajectory Planning

The major difficulties of trajectory planning include:

1. It is hard to appropriately consider the dynamic constraints of vehicles

2. It is hard to determine the immediate reward of vehicles

3. It is hard to master the general knowledge that applies for all kinds of vehicles

A.1. Application in Trajectory Planning

Most existing approaches are indirect trajectory planning in two steps:

First, design a reference parking trajectory.

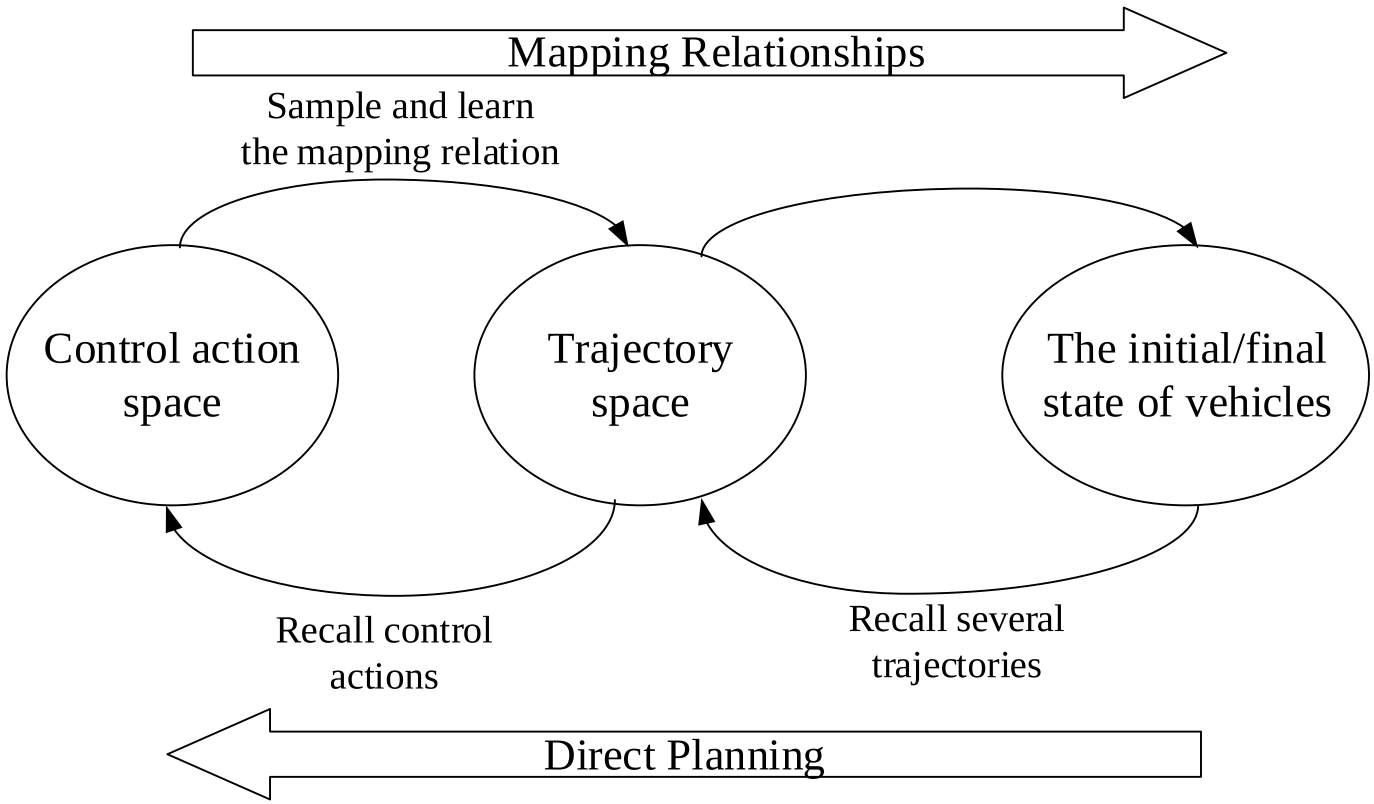
Second, design a controller to make vehicle track it.

Its shortcomings include:

- The dynamic constraints are implicit and inaccurate
- It is hard to design a proper controller
- If the first step gives a wrong solution, no chance to make up.

A.1. Application in Trajectory Planning

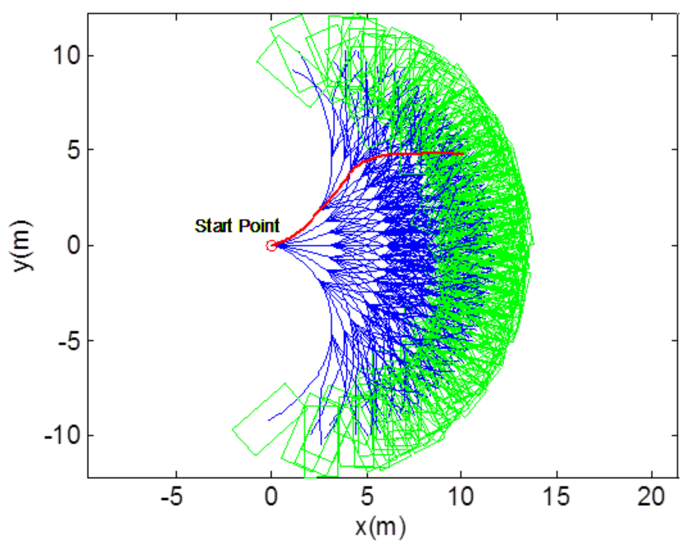
We propose direct trajectory planning method: learn the mapping relation between the final state and the corresponding trajectory via deep learning networks (Liu et al., 2017).

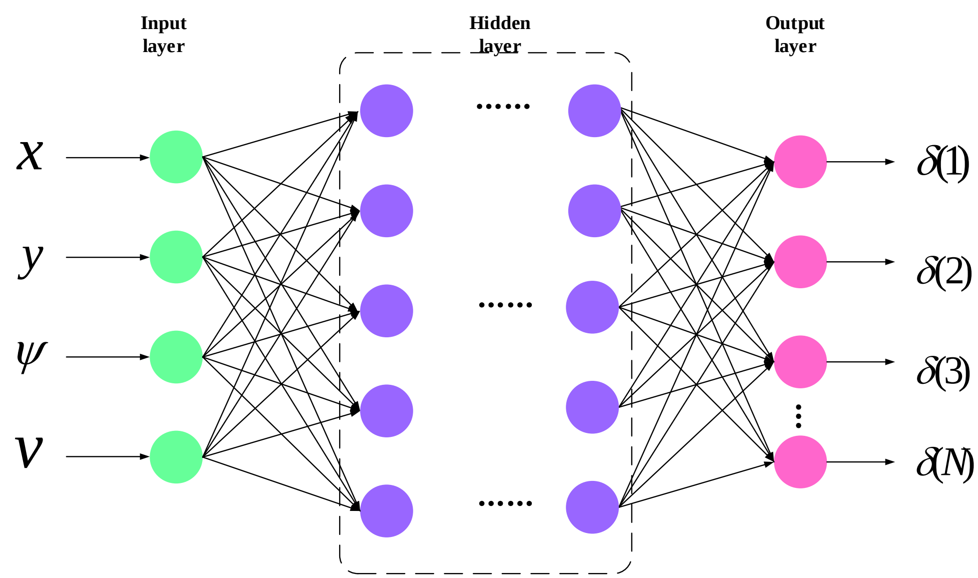


A.1. Application in Trajectory Planning

We sample all the possible solutions, use deep neural network to "remember" them, and directly recall them on demand (Li and Wang, 2003; Liu et al., 2017).

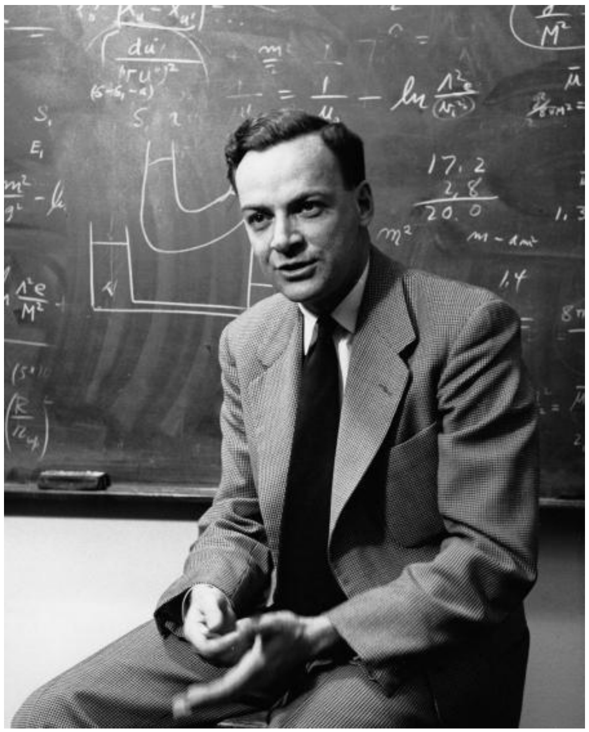
- The dynamic constraints are naturally satisfied
- No need of compelx controller
- Integrated solution in just one step

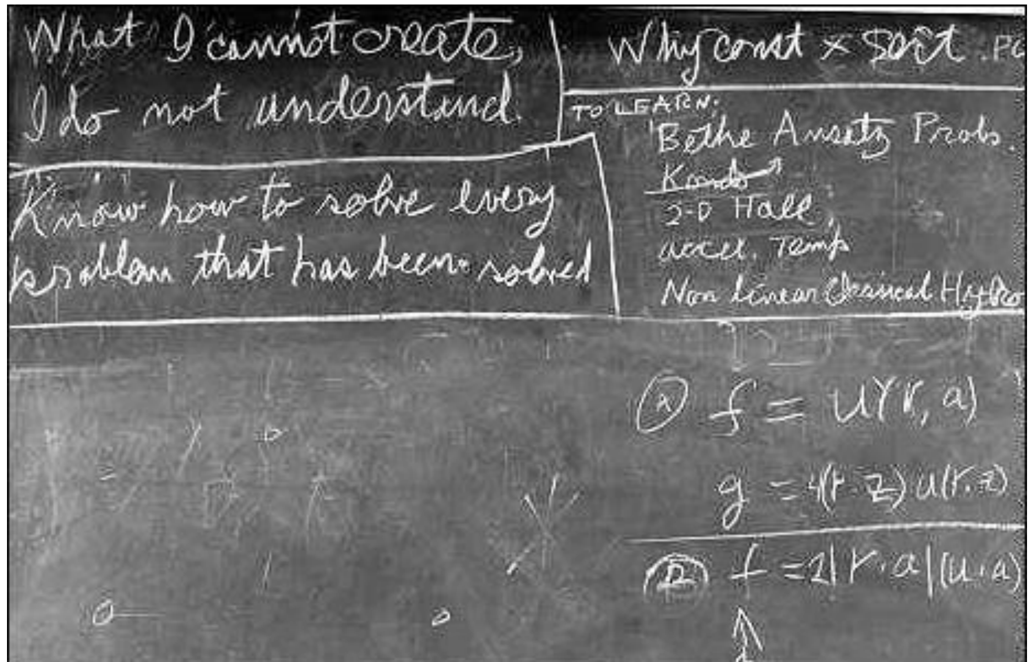




A.1. Application in Trajectory Planning

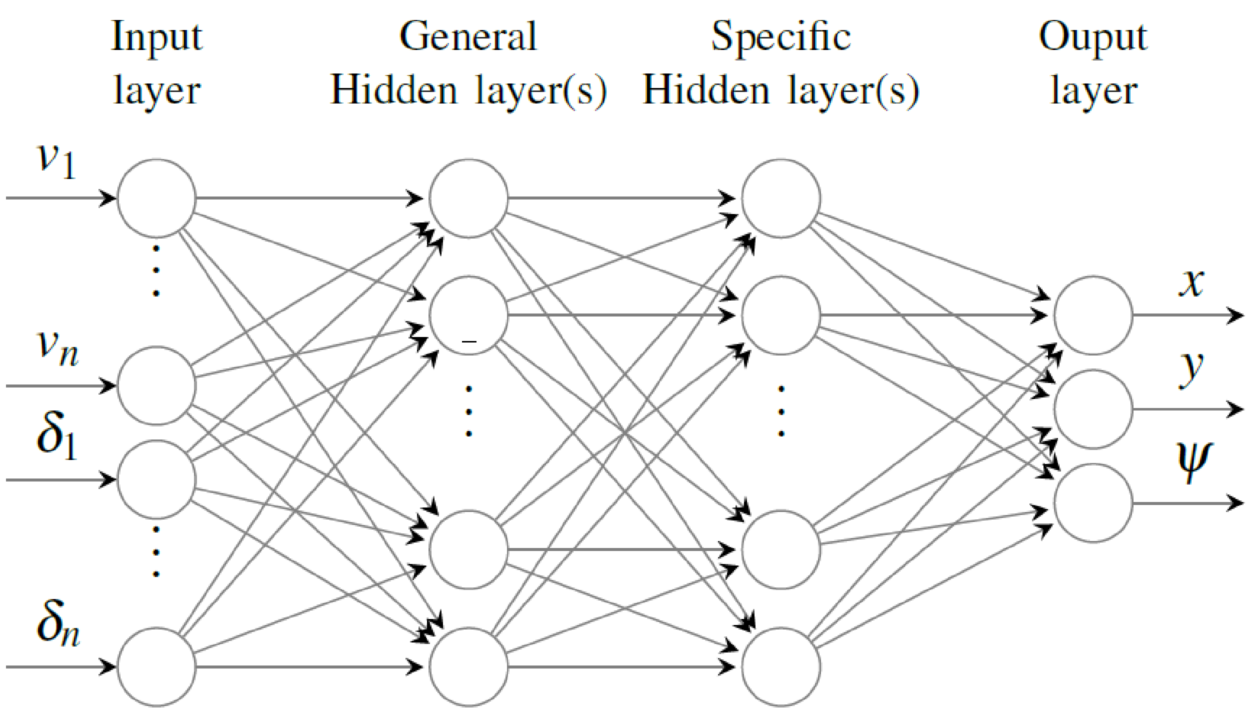
Just as Prof. Richard Feynman had said "*Know how to solve every problem that has been solved.*"





A.1. Application in Trajectory Planning

We further build a special DNN. The first few layers contain the general trajectory planning knowledge for all kinds of vehicles; while the last few layers can be quickly tuned to adapt various kinds of vehicles (Lin et al., 2017).



A.2. Parallel Learning Theory

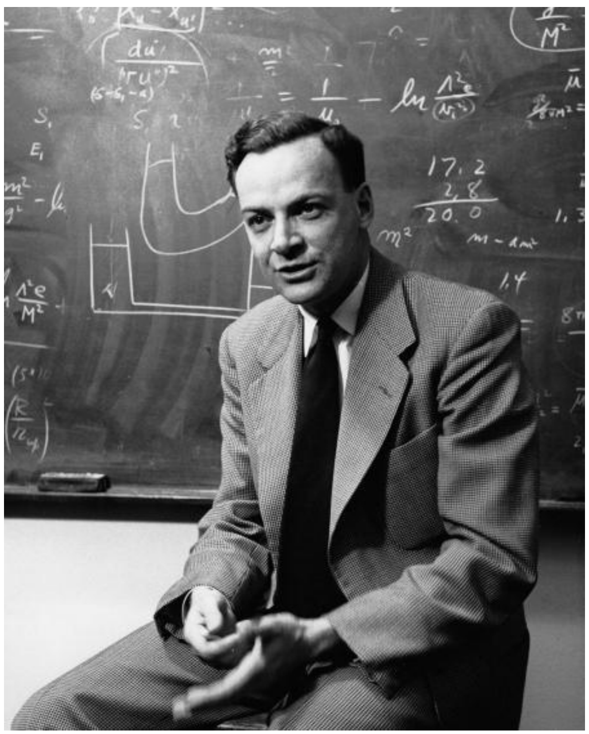
For some control systems (especially automated vehicles), we have the following problems in applying machine learning (Li et al., 2016; 2017)

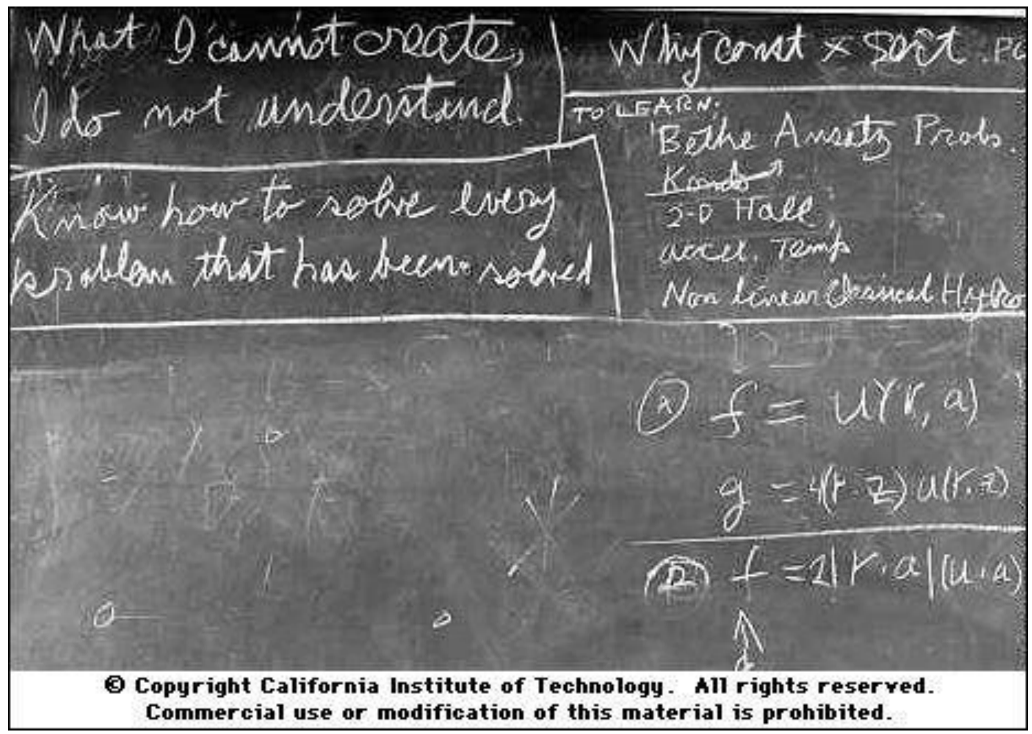
1. How to distill knowledge from data?
2. How to label data in an unsupervised way?
3. How to guide the system with growing knowledge?
4. How to master the general knowledge for similar systems?

A.2. Parallel Learning Theory

1. How to distill knowledge from data?

Descriptive learning: learn from data by creating the same (kind of) data. Just as Prof. Richard Feynman had said "*What I cannot create, I do not understand.*"

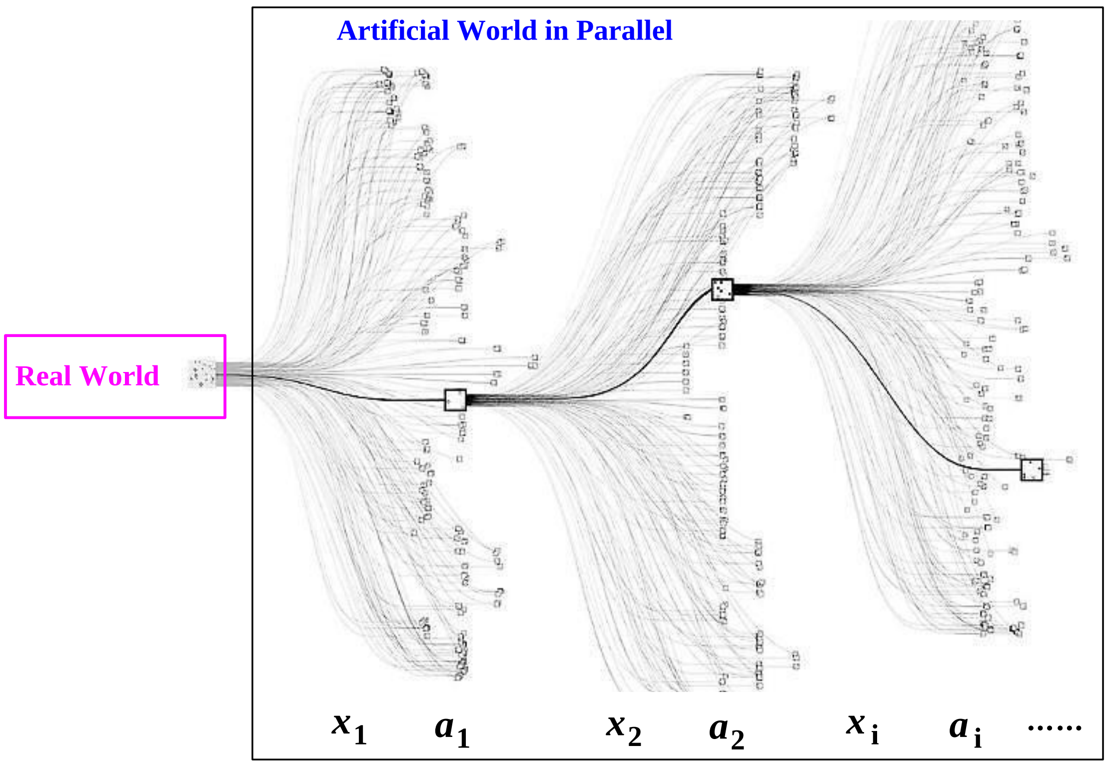




A.2. Parallel Learning Theory

2. How to label data in an unsupervised way?

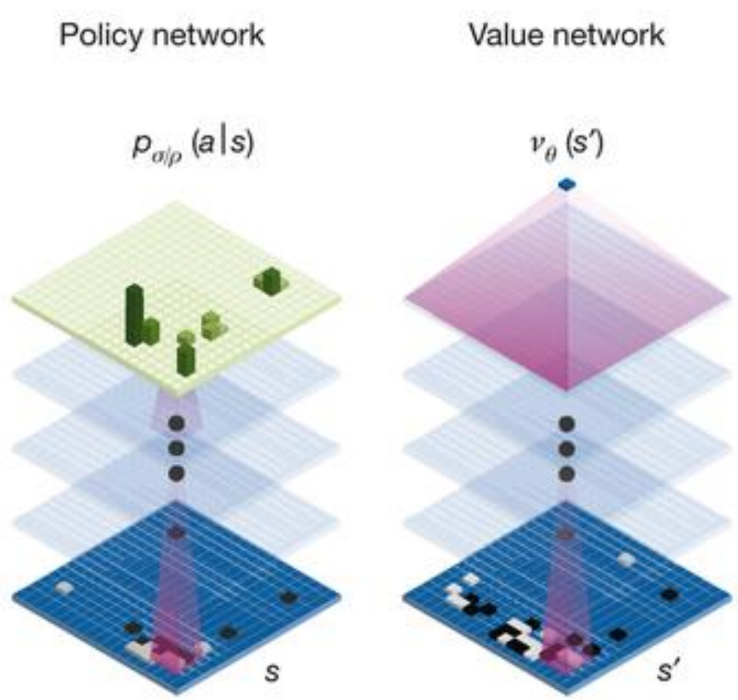
Predictive learning: label data by letting the system evolve in an unsupervised manner; just like AlphaGo

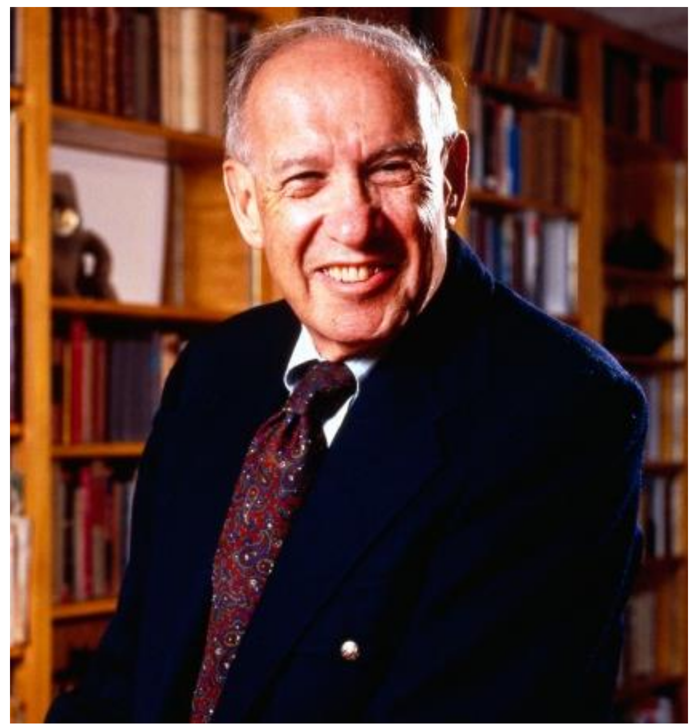


A.2. Parallel Learning Theory

3. How to guide the system with growing knowledge?

Prescriptive learning: make the system evolve appropriately by special trying-and-testing. Just like what Peter F. Drucker said: "*The best way to predict the future is to create it.*"

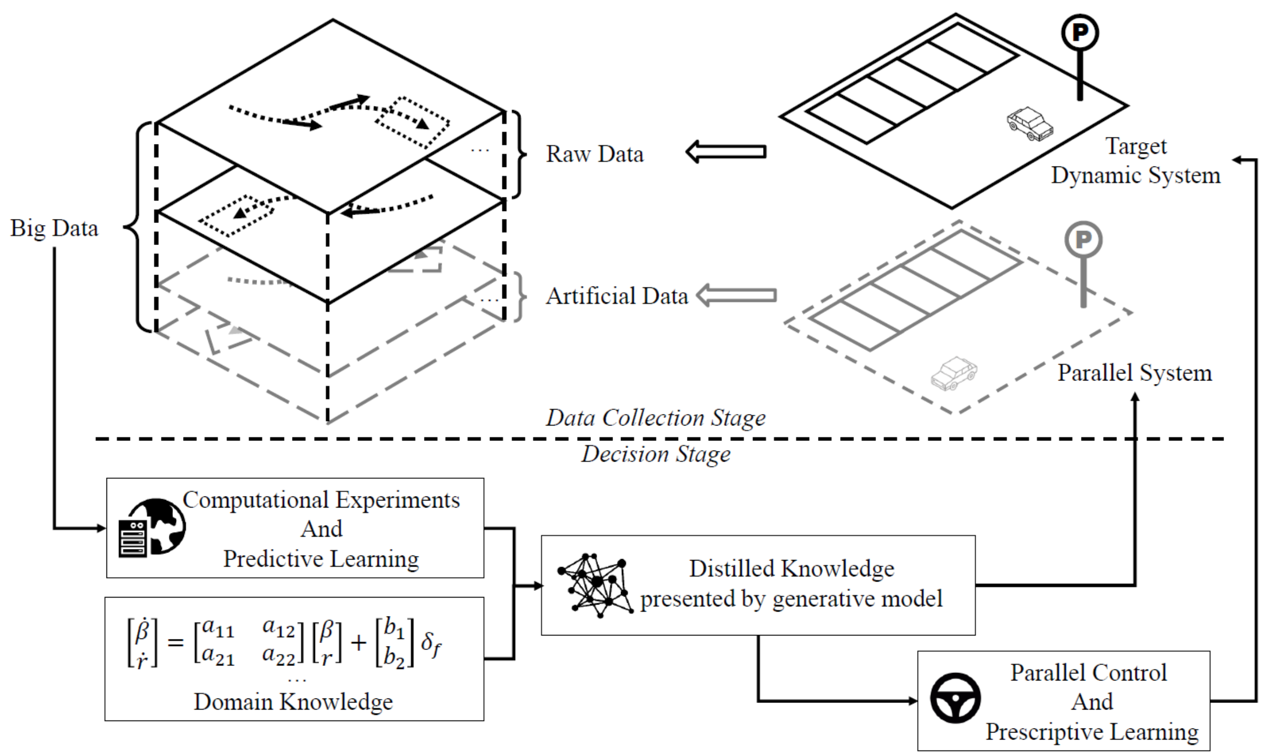




A.2. Parallel Learning Theory

4. How to master the general knowledge for similar systems?

Transfer learning: leverage the already learned knowledge of some related task or domain to deal with the new problems.



A.2. Parallel Learning Theory

Recall the trajectory planning problem, we have

- Descriptive Learning: Sample all the possible solutions (trajectories)
- Predictive Learning: Link the destination with the corresponding actions
- Prescriptive Learning: Use deep neural network to generate trajectory
- Transfer Learning: Adjust the neural network to adopt different vehicles

A.3. Virtual Test of Automated Vehicles

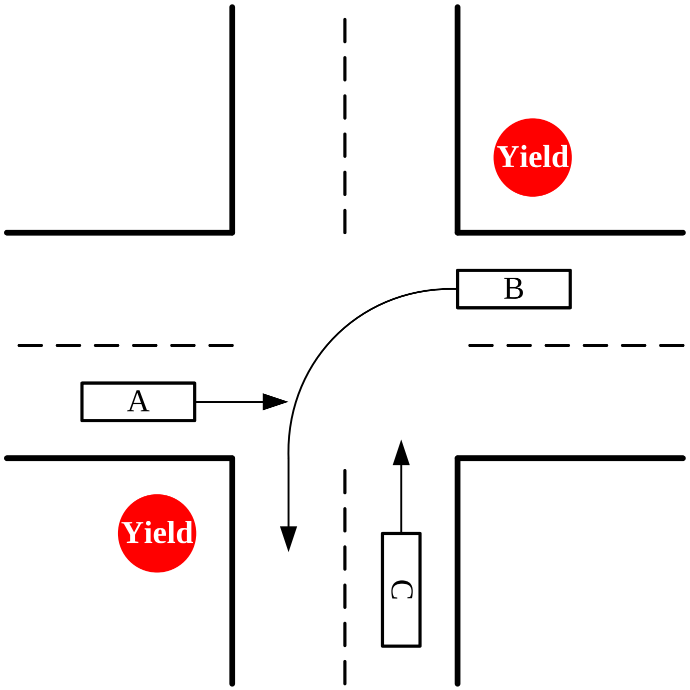
Although great achievements were obtained, there exist many difficulties in evaluating the intelligence of automated vehicles. Particularly, we care about:

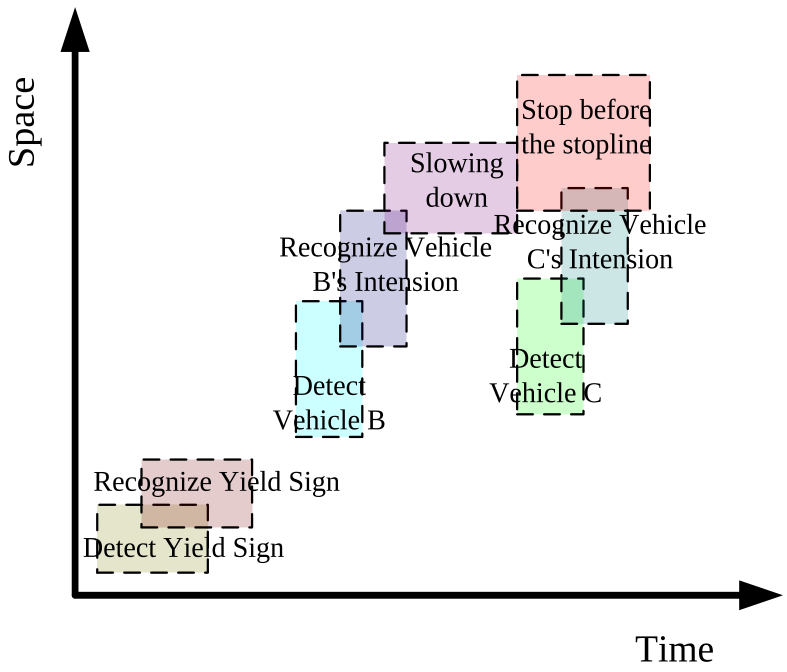
- How to describe simulated scenarios?
- How to generate simulation data?
    - Agents should behave as "real" as possible
    - Vision data should be as "rich" as possible

A.3. Virtual Test of Automated Vehicles

- How to describe simulated scenarios?

The core of scenario design is to settle a series of tasks that needs to be finished sequentially within a given temporal-spatial ranges.





A.3. Virtual Test of Automated Vehicles

- How to describe simulated scenarios?

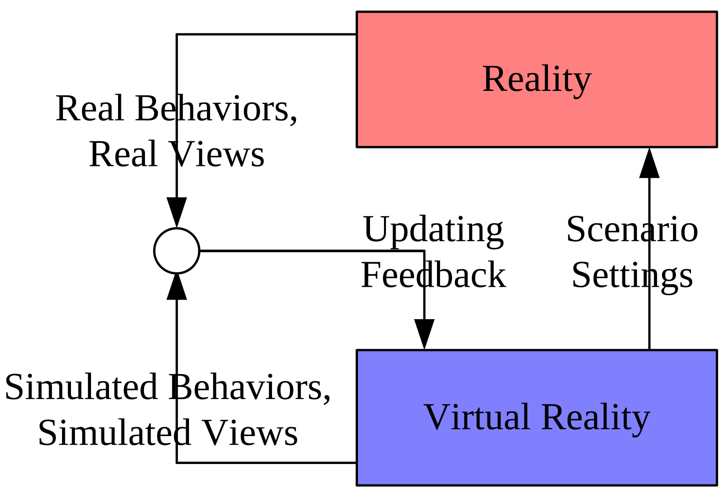
Change the combinations and parameters of tasks, we can gradually cover all the possible scenarios and quantitatively evaluate the capability of automated vehicles.

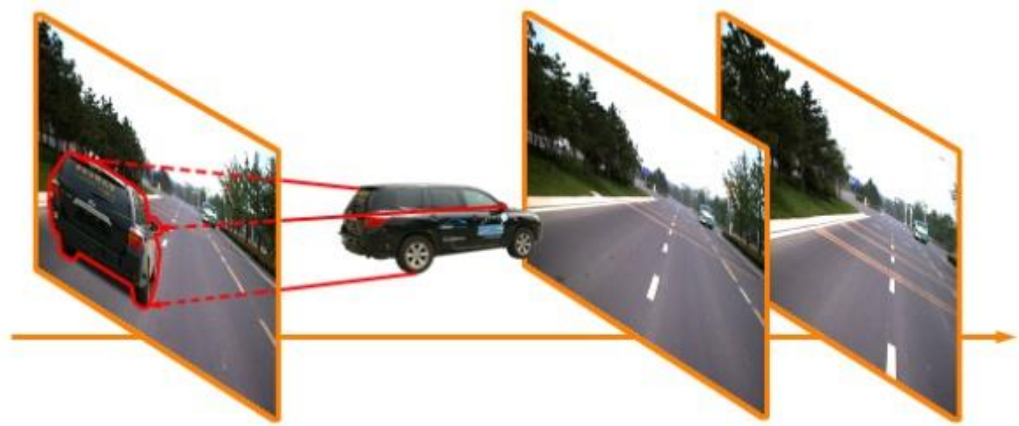
A.3. Virtual Test of Automated Vehicles

- Agents should behave as "real" as possible

Descriptive learning: Let the real world and its mirror (parallel traffic systems) co-evolve appropriately to make the simulation more and more "real".

Specially, we make agents in parallel traffic system learn how to behave from data collected from real world.





A.3. Virtual Test of Automated Vehicles

- Agents should behave as "real" as possible

We are building a V2X-adided testing base at Changshu city, which helps us to collect data to calibrate the simulation agents.

A.3. Virtual Test of Automated Vehicles

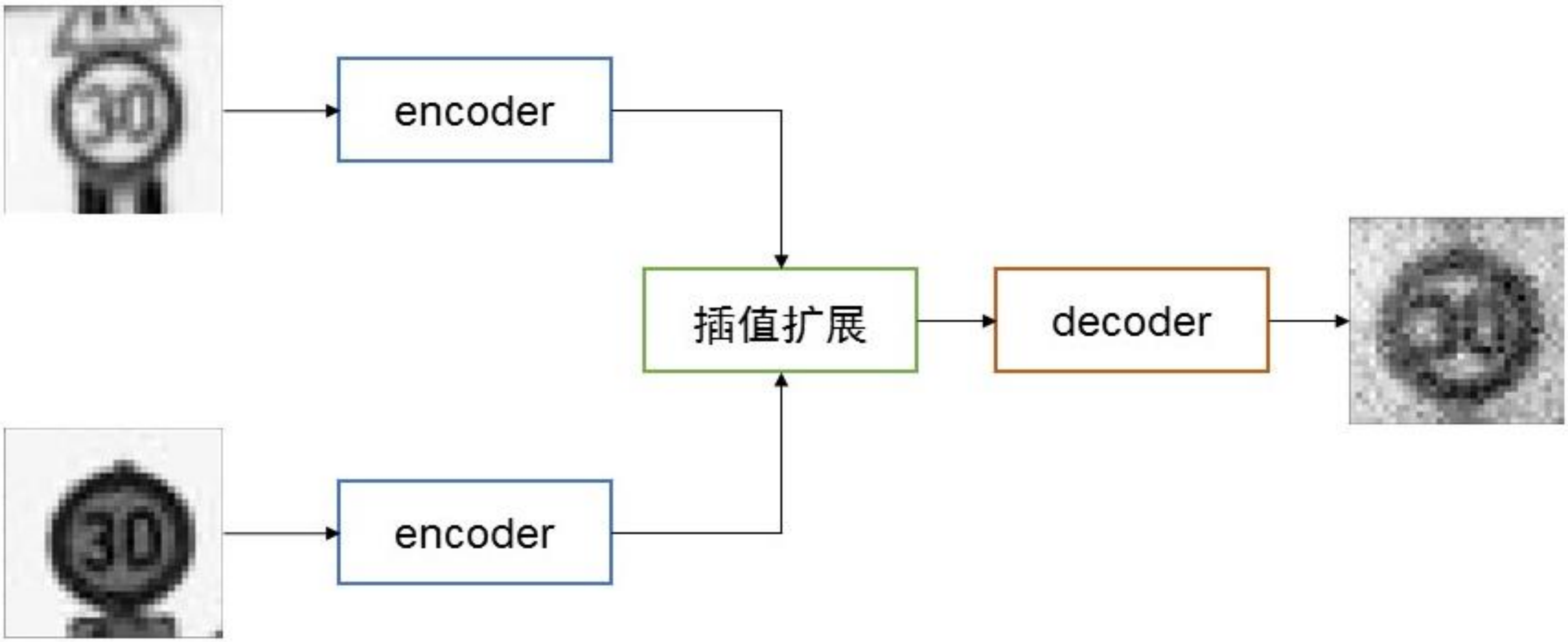
- Vision data should be as "rich" as possible

Descriptive learning + Transfer Learning: we test four ways to generate new vision data:

1. Naïve data augmentation method
2. 2D → 3D → 2D method
3. 2D Generative Adverserial Networks (GAN) method
4. 2D → 3D → 3D GAN → 2D method

A.3. Virtual Test of Automated Vehicles

1. Naïve data augmentation method: An augmented datasets for traffic sign recognition (Tsinghua Univeristy and Xi'an Jiaotong Univeristy)



A.3. Virtual Test of Automated Vehicles

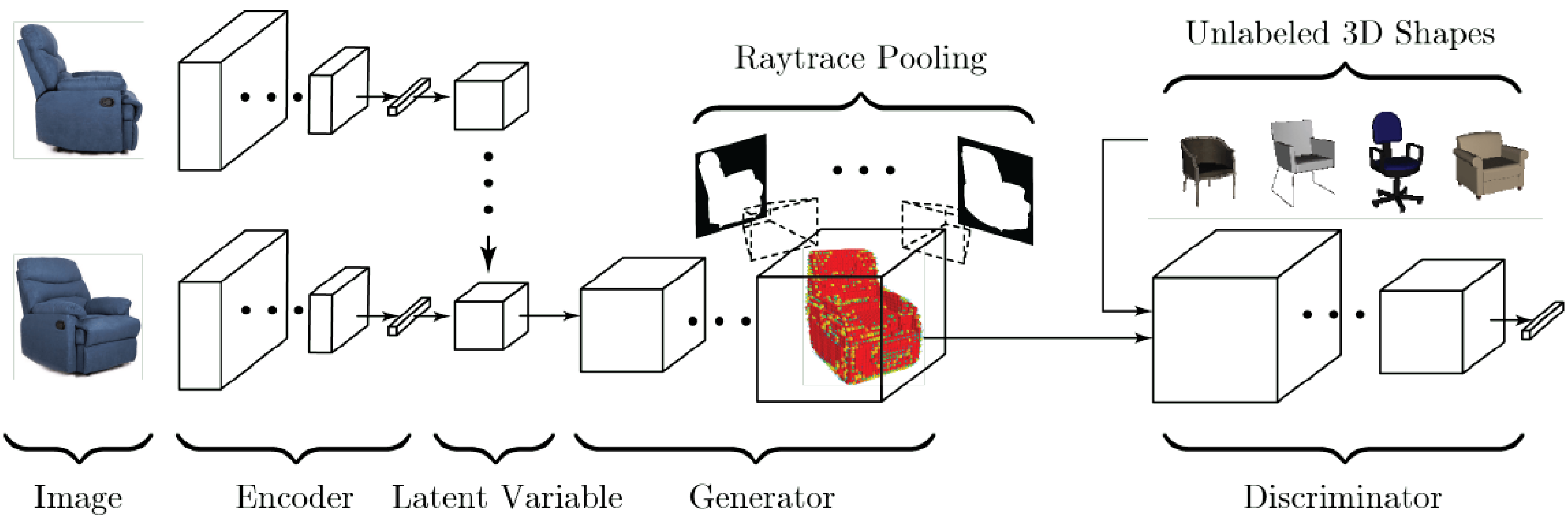
2. 2D → 3D → 2D method: Virtual worlds as proxy for multi-object tracking analysis (Xerox Research Center)

A.3. Virtual Test of Automated Vehicles

3. 2D Generative Adverserial Networks (GAN) method: Learning a Driving Simulator (comm.ai)

A.3. Virtual Test of Automated Vehicles

4. 2D → 3D → 3D GAN → 2D method: Weakly Supervised Generative Adversarial Networks for 3D Reconstruction (Stanford Univeristy)



A.4. Conclusions

We propose a new framework of machine learning theory, parallel learning, which addresses the following questions:

- How to use descriptive learning to distill knowledge
- How to use predictive learning to label data
- How to use prescriptive learning to guide actions
- How to use transfer learning to solve general problems

The obtained conclusions can be extended to other fields:

- Drone
- Unmanned boat
- Train

A.5. Reference

[1]Li Li, Yilun Lin, Dongpu Cao, Nan-Ning Zheng, Fei-Yue Wang*, "Parallel learning - A new framework for machine learning," *ACTA Automatica Sinica*, vol. 43, no. 1, pp. 1-7, 2017. (in Chinese)

[2]Li Li, Yilun Lin, Nan-Ning Zheng, Fei-Yue Wang*, "Parallel learning - A pespective and a framework," *IEEE/CAA Journal of Automatica Sinica*, submitted.

[3]Wei Liu, Zhiheng Li, Li Li*, Fei-Yue Wang, "Parking like human: A direct trajectory planning solution," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[4]Yi-Lun Lin, Li Li*, Xing-Yuan Dai, Nan-Ning Zheng, Fei-Yue Wang, "Master general parking skill via deep learning," *Proceedings of IEEE Intelligent Vehicle Symposium*, 2017.

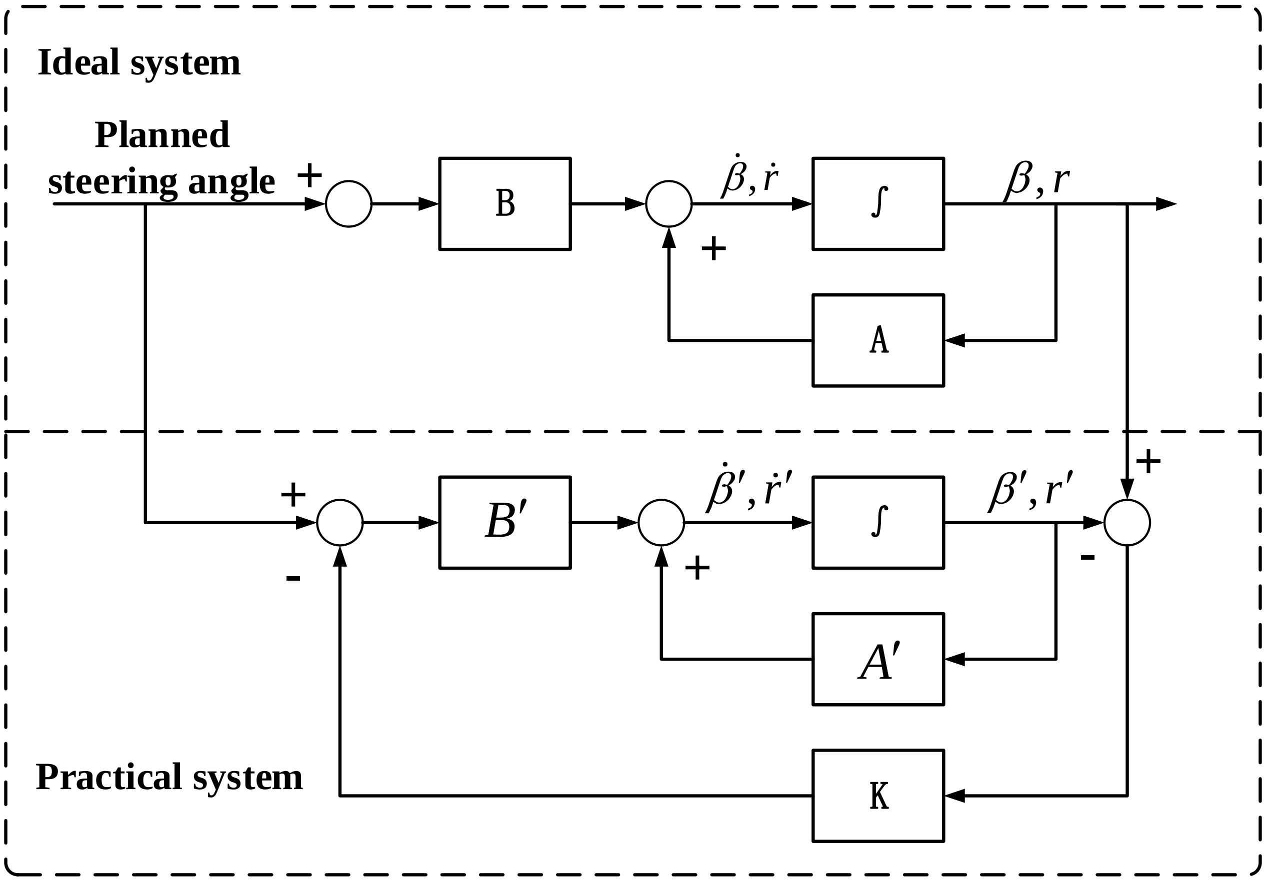
[5]Li Li, Wu-Ling Huang*, Yuehu Liu, Nan-Ning Zheng, Fei-Yue Wang, "Intelligence testing for autonomous vehicles: A new approach," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 2, pp. 158-166, 2016.

Thanks!

Questions?

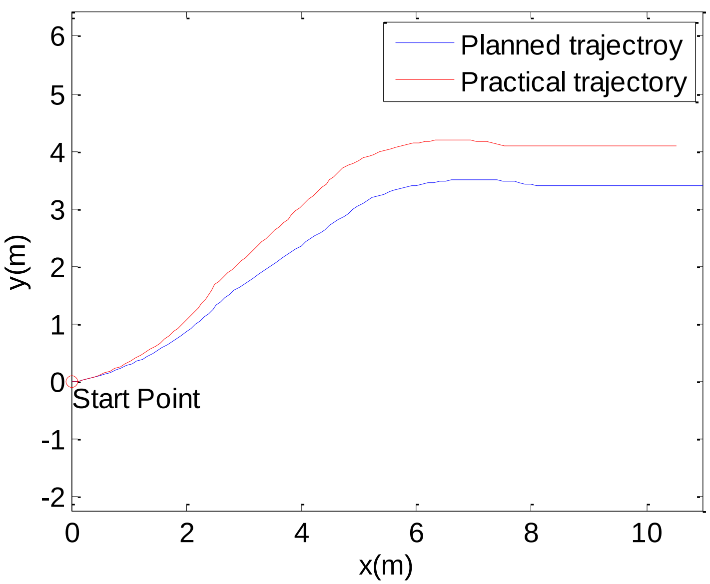
A.1. Application in Trajectory Planning

Trajectory planning belongs to feedforward control. Feedback control is still needed, in case parameters changed.

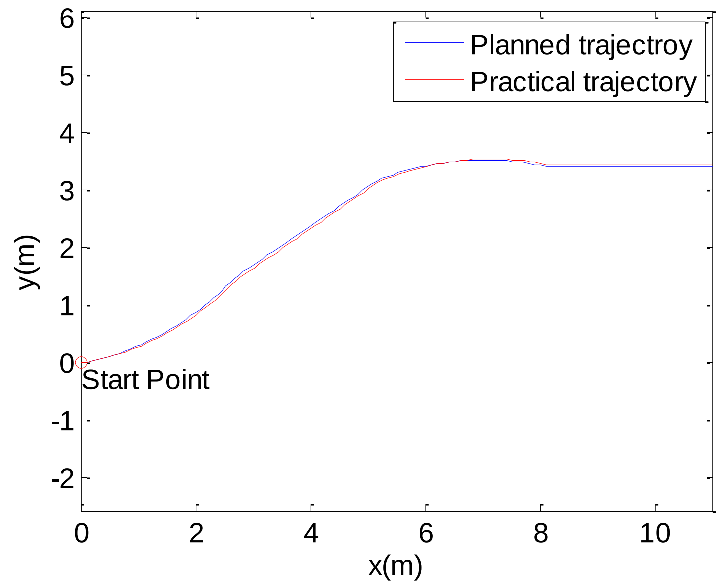


A.1. Application in Trajectory Planning

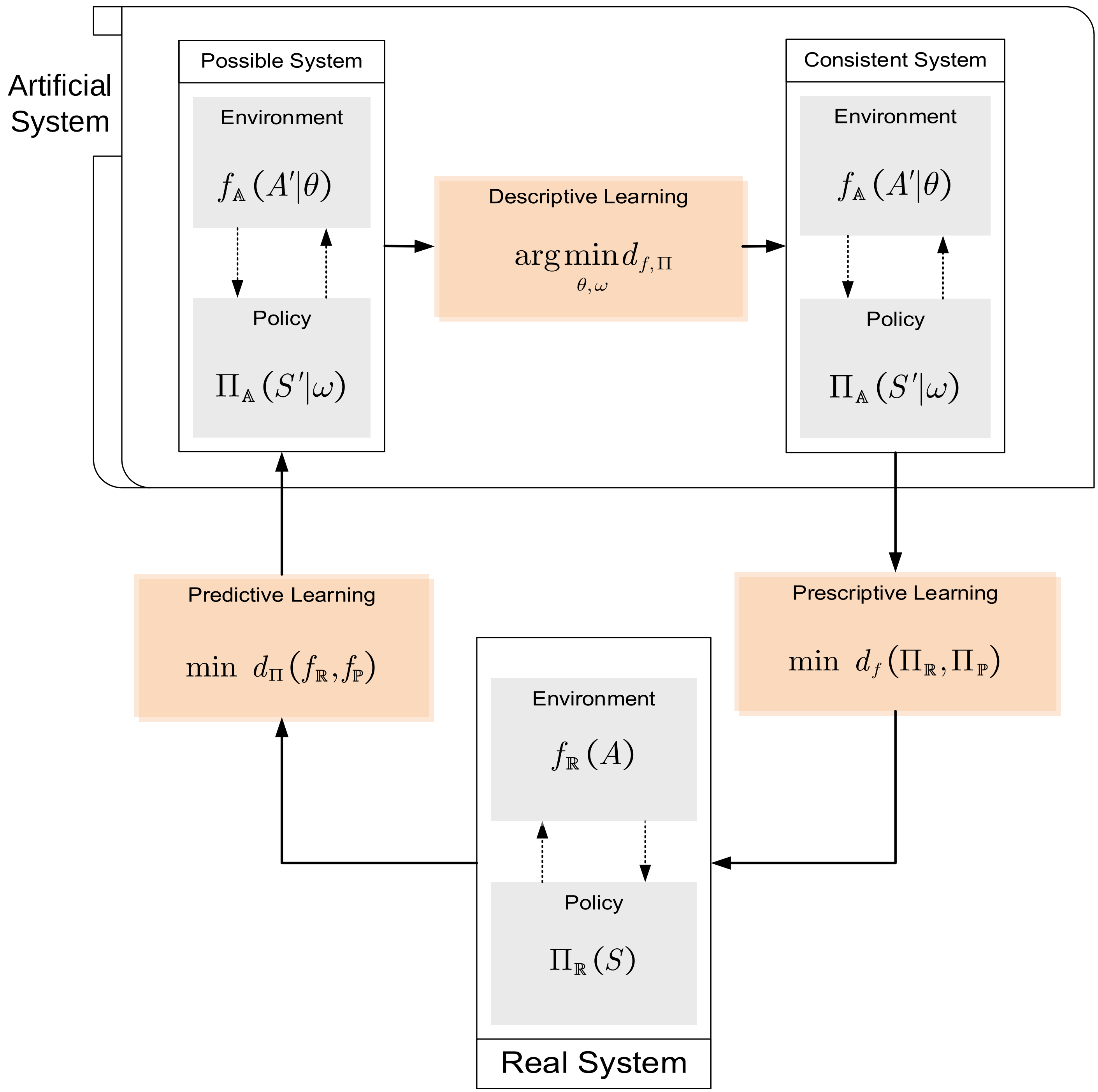
Trajectory planning belongs to feedforward control. Feedback control is still needed, in case parameters changed. For example, if tire/road friction coefficients suddenly change (by water), we need feedback to reach the goal. However, this controller can be very simple.



without feedback control



with feedback control


Artificial System
Possible System
Environment
$f_{\mathbb{A}}(A'|\theta)$
Policy
$\Pi_{\mathbb{A}}(S'|\omega)$
Descriptive Learning
$\arg\min_{\theta,\omega} d_{f,\Pi}$
Consistent System
Environment
$f_{\mathbb{A}}(A'|\theta)$
Policy
$\Pi_{\mathbb{A}}(S'|\omega)$
Predictive Learning
$\min\ d_{\Pi}(f_{\mathbb{R}}, f_{\mathbb{P}})$
Prescriptive Learning
$\min\ d_{f}(\Pi_{\mathbb{R}}, \Pi_{\mathbb{P}})$
Environment
$f_{\mathbb{R}}(A)$
Policy
$\Pi_{\mathbb{R}}(S)$
Real System