MACHINE LEARNING

CHAPTER 0: INTRODUCTION

Contact Information

Instructor: Qi Hao

E-mail: hao.q@sustc.edu.cn

Office: Nanshan iPark A7 Room 906

Office Hours: M 2:00-4:00pm

Available other times by appointment or the open door policy

Office Phone: (0755) 8801-8537

QQ: 1132148103 机器学习2020

Web: *http://hqlab.sustc.science/teaching/CS405*

BB: CS405

Class Schedule

- **Lectures**: W 08:00 am – 09:50 am Lychee Hills 1 Room 203

- **Labs**: W 10:20 am – 12:10 pm Lychee Hills 6 Room 408
  W 16:20 am – 18:10 pm Lychee Hills 6 Room 406

- **Grading policy:**

Final Exam (in-class):	**20%**	**Midterm Exam (take-home):**	**10%**
Assignments (8~12 times):	**20%**	**Quizzes (<=10 times):**	**10%**
Lab Projects :	**20%**	**Final Projects (4 per group):**	**20%**
Bonus Credits:	**<5%**		

90~93: A-	94~97: A	98~100: A+
80~82: B-	83~86: B	87~89: B+
70~72: C-	73~76: C	77~79: C+
60~62: D-	63~66: D	67~69: D+

Textbook and Lecture Notes

Textbooks:

[1] Pattern Recognition and Machine Learning, by Christopher M. Bishop, 2006 Springer

[2] Machine Learning in Action, by Peter Harrington, 2012, Manning

Other books:

[1] 机器学习, 周志华

[2] Dive in Deep Learning, by Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola

[3] Reinforcement Learning: An Introduction, by Richard S. Sutton

[4] The Elements of Statistical Learning, by Trevor Hstie, Rober Tibshirani, Jerome Friedman

Paper reading:

[1] Ghahramani Z. Probabilistic machine learning and artificial intelligence, Nature, 2015

[2] Lecun Y, Bengio Y, Hinton G. Deep learning, Nature, 2015

[3] Littman M L. Reinforcement learning improves behavior from evaluative feedback, Nature, 2015

Lecture notes:

http://hqlab.sustc.science/teaching/CS405

Other Resources

Assignment platform: bb.sustech.edu.cn

Textbook resource: https://www.microsoft.com/en-us/research/people/cmbishop/#prml-book

Textbook Matlab codes : http://prml.github.io/

Matlab toolboxes : Machine Learning, Neural Networks

Teaching Objectives

- Fundamental knowledge about machine learning and pattern recognition, from Bayesian approaches to deep learning frameworks through lectures, quizzes and excercises
- Machine learning system development methods in Python based platforms (numpy, sciki-learn, tensorflow) through labs and projects
- Model-based and data-driven machine learning system design and integration skills through the final project, literature surveys and reports

Lecture Schedule

Section 0	Course Introduction	
Section 1	Preliminary	(HW1)
Section 2	Probability Distributions	(HW2)
Section 3	Linear Regression and Classification	(HW3)
Section 4	Neural Networks	(HW4)
Section 5	Sparse Kernel Machine	(HW5)
Section 6	Clustering and EM learning	(HW6)
-- *Midterm Exam* --		
Section 7	Dimension Reduction and Feature Selection	(HW7)
Section 8	Ensemble Learning	(HW8)
Section 9	Bayesian Networks	(HW9)
Section 10	Hidden Markov Model	(HW10)
Section 11	Markov Decision Process	(HW11)
Section 12	Reinforcement Learning	(HW12)
-- *Final Exam* --		

Lab Schedule

Section 0	Lab Introduction
Section 1	Preliminary
Section 2	Bayes
Section 3	Regression and Classification
--Final Project Proposal--	
Section 4	Decision Tree
Section 5	Random Forest (Ensemble Learning)
Section 6	KNN and Support Vector Machine
Section 7	K-Mean and EM Clustering
Section 8	Neural Network (I)
Section 9	Neural Network (II)
Section 10	Neural Network (III)
Section 11	Reinforcement Learning
--Final Project Report--	

Final Projects

[1] Reinforcement learning based planning using a self-driving car simulator

[2] Generation of annotated self-driving datasets with the CARLA simulator

[3] Segmentation of 2D/3D measurements for self-driving applications

[4] Detection and recognition of traffic signs for self-driving applications

[5] Detection and tracking of 2D/3D objects for self-driving applications

[6] Federated learning for model fusion of networked vehicle applications

Bonus Credits

- AI companies
- Survey Papers
- Attendance
- Bonus Credits

Plagiarism

From Spring 2018, the plagiarism policy applied by the Computer Science and Engineering department is the following:

* If an assignment is found to be plagiarized, the first time the score of the assignment will be 0.

- The second time the score of the course will be 0.

As **it may be difficult** when two assignments are identical or nearly identical who actually wrote it, the policy will apply to BOTH students, unless one confesses having copied without the knowledge of the other.

What is OK, and what isn't OK

It's OK

- to work on an assignment with a friend, and think together about the program structure, share ideas and even the global logic. At the time of actually writing the code, you should write it alone.
- to use in an assignment a piece of code found on the web, as long as you indicate in a comment where it was found and don't claim it as your own work.
- to help friends debug their programs (you'll probably learn a lot yourself by doing so).
- to show your code to friends to explain the logic, as long as the friends write their code on their own later.

It's NOT OK

- **to take the code of a friend, make a few cosmetic changes (comments, some variable names) and pass it as your own work.**

Make a Promise to Keep

Sign

the “Student Commitment for Assignments”

Keep

the promise during the whole semester!

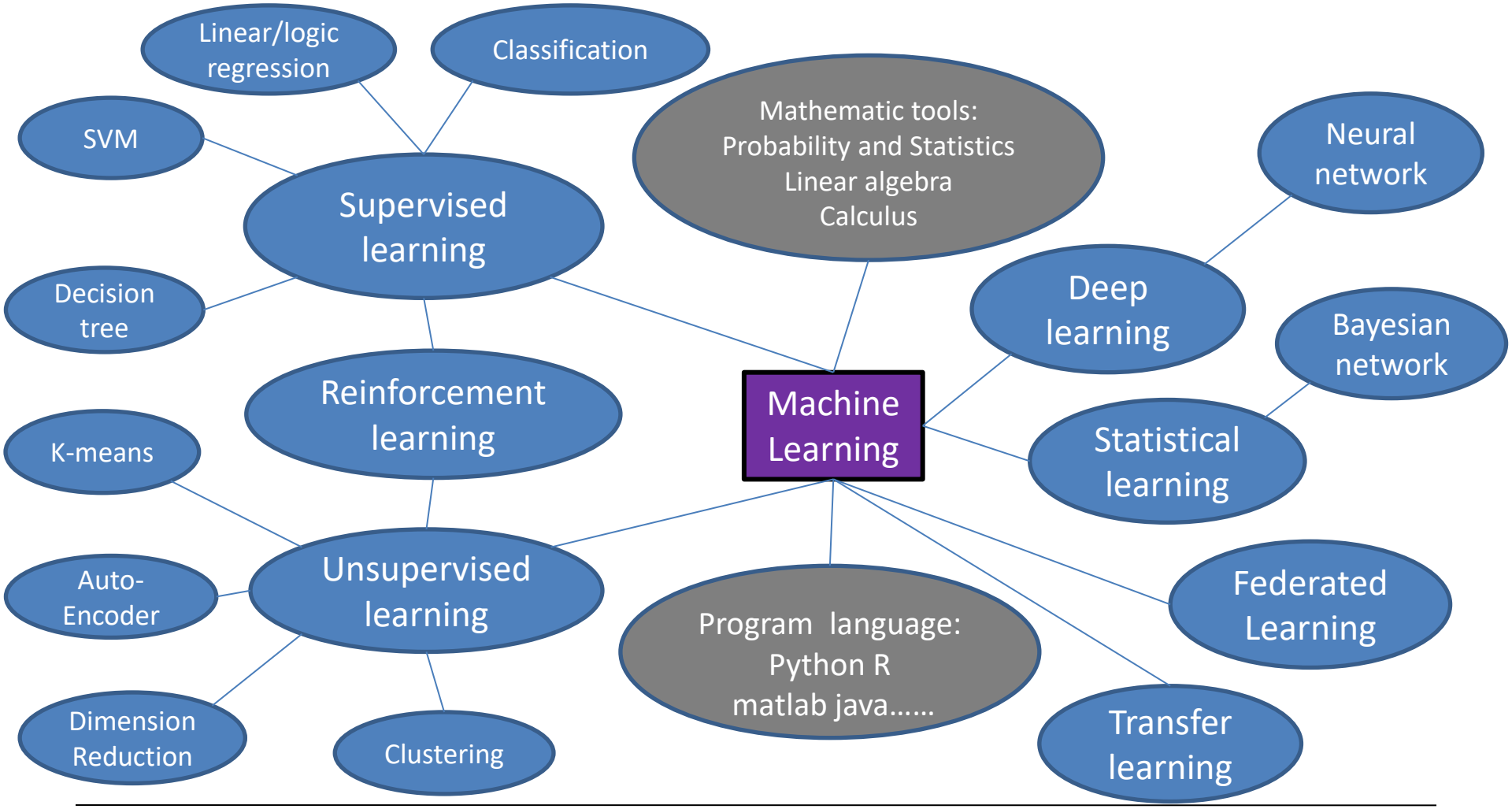
Outlines

- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

Outlines

- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

Framework

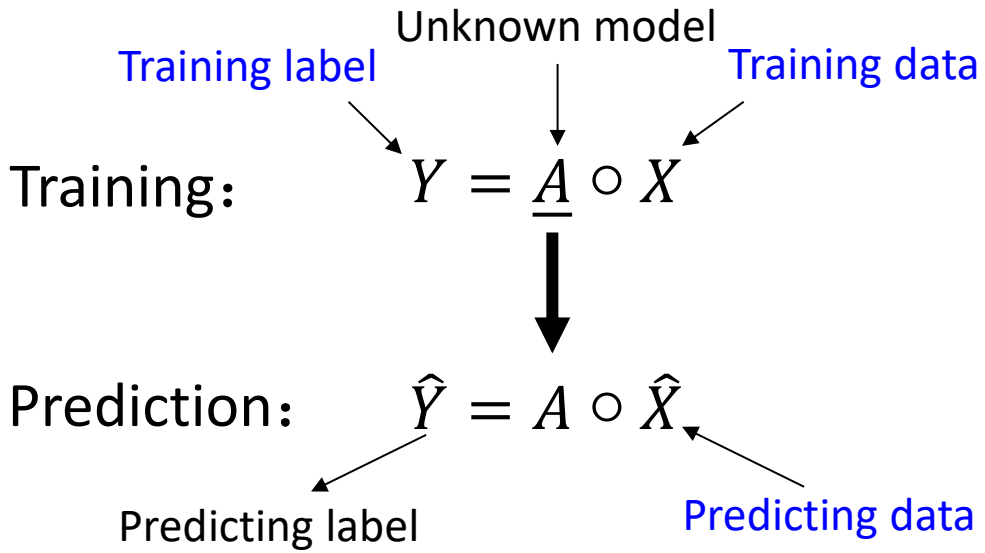

Linear/logic regression
Classification
SVM
Supervised learning
Mathematic tools:
Probability and Statistics
Linear algebra
Calculus
Neural network
Decision tree
Deep learning
Bayesian network
Reinforcement learning
Machine Learning
K-means
Statistical learning
Auto-Encoder
Unsupervised learning
Federated Learning
Program language:
Python R
matlab java……
Dimension Reduction
Clustering
Transfer learning

Outlines

- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

Problem Statement

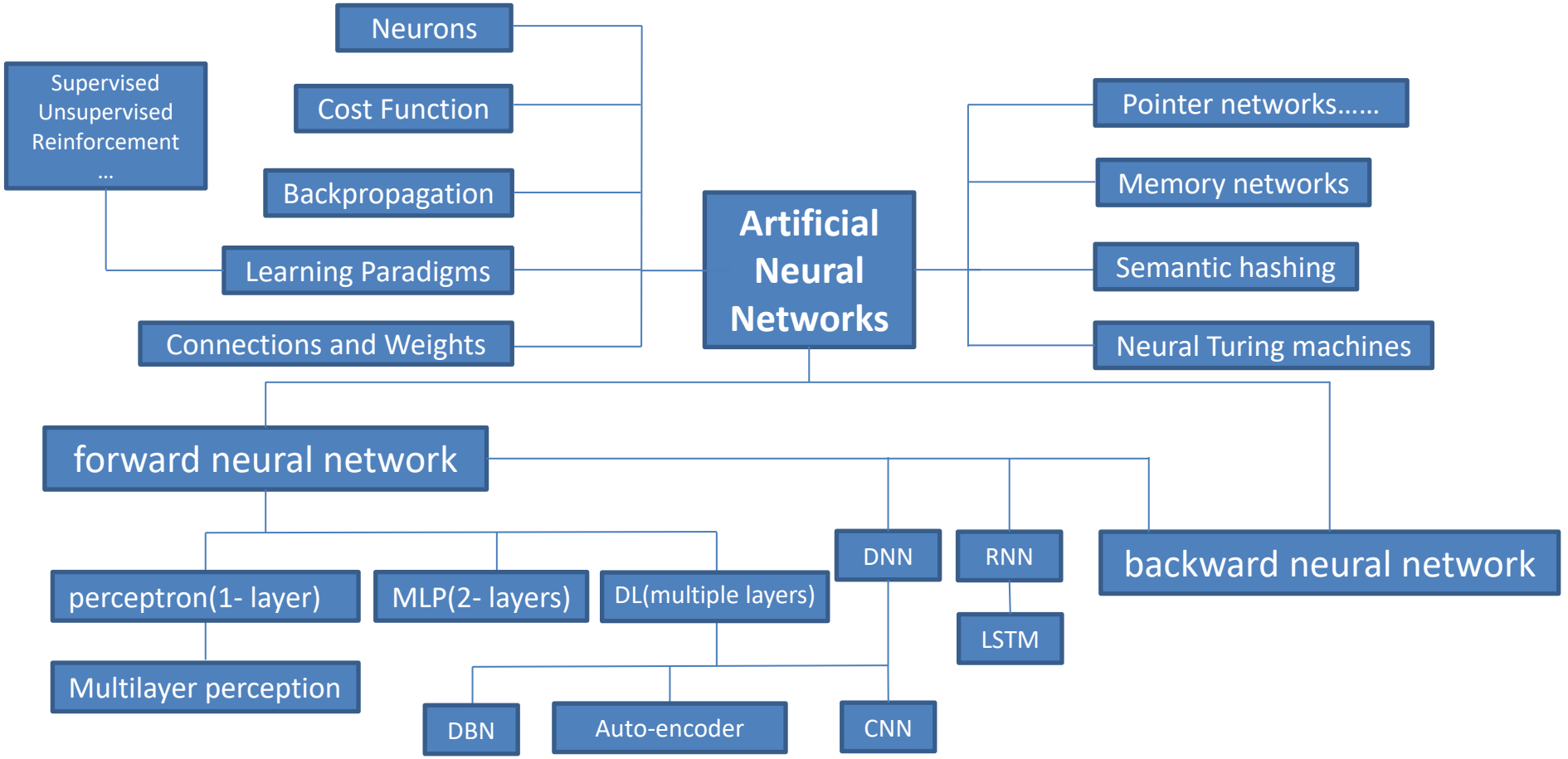
- Problem： Predict the label $\hat{Y}$ and data $\hat{X}$ with training set（X,Y）？



$\begin{cases} Y\ and\ X\ is\ known： \text{supervised learning} \\ Y\ or\ X\ is\ unknown： \text{unsupervised learning} \end{cases}$ $\begin{cases} Y、\hat{Y}\ are\ continuous： \text{Regression} \\ Y、\hat{Y}\ are\ discrete： \text{classification} \end{cases}$

$Y\ is\ known\ and\ Dim(Y) > Dim(X)$： dimensionality reduction

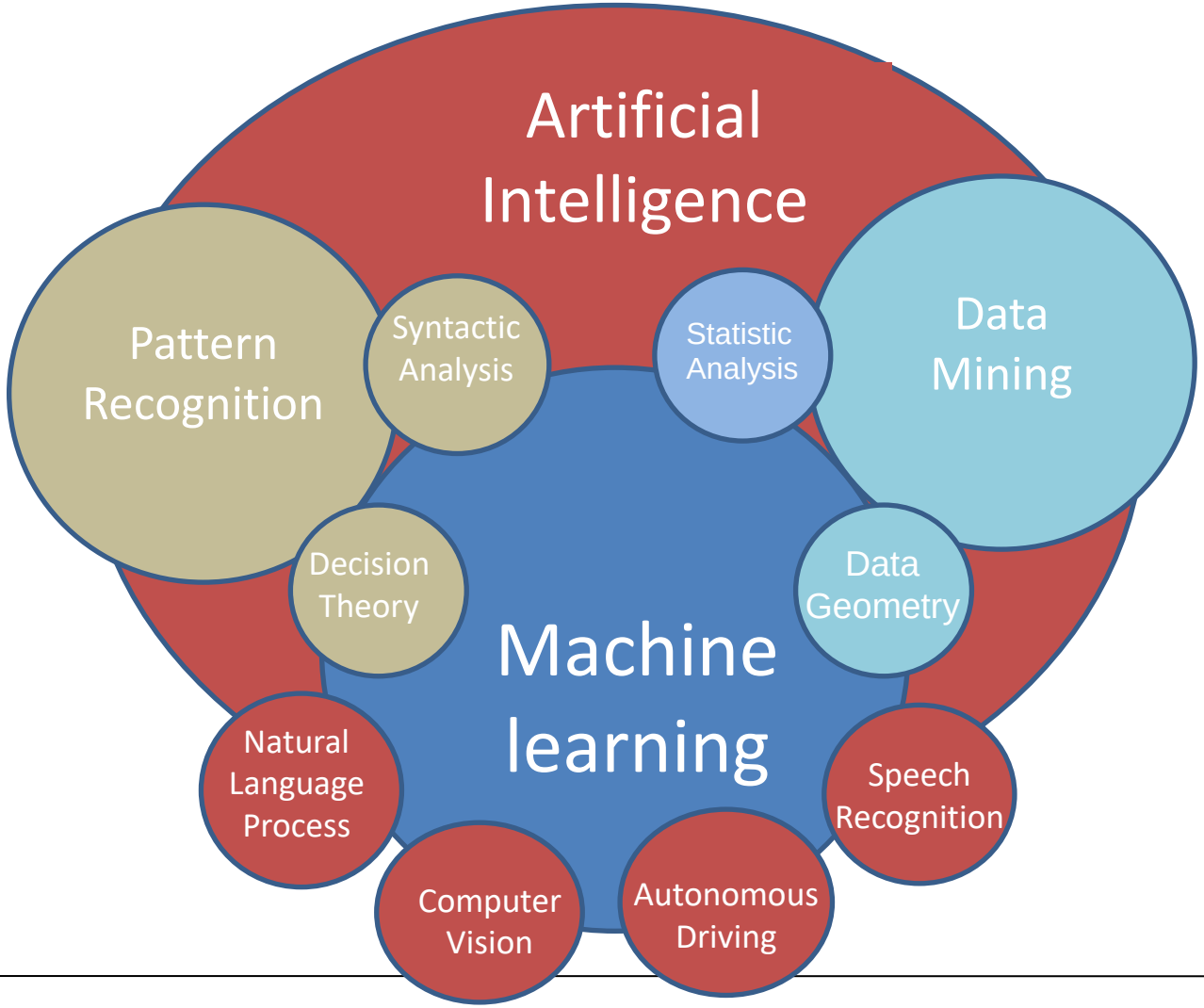
Neural Network Models



Outlines

- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

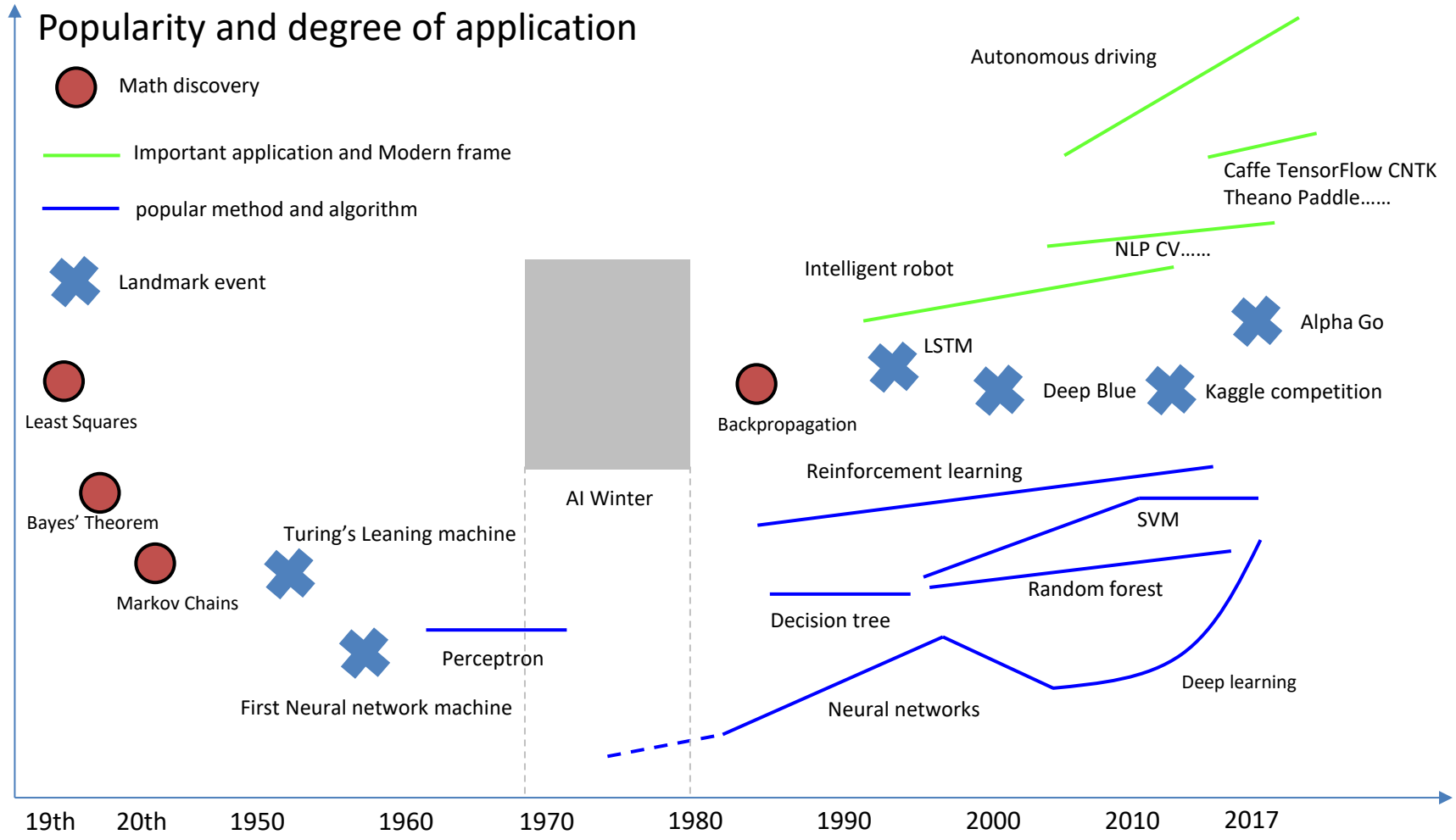
The Whole Picture



Outlines

- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

History


Popularity and degree of application
Math discovery
Important application and Modern frame
popular method and algorithm
Landmark event
Autonomous driving
Caffe TensorFlow CNTK
Theano Paddle……
NLP CV……
Intelligent robot
Alpha Go
LSTM
Deep Blue
Kaggle competition
Backpropagation
Least Squares
AI Winter
Reinforcement learning
Bayes' Theorem
Turing's Leaning machine
SVM
Markov Chains
Random forest
Decision tree
Perceptron
Deep learning
First Neural network machine
Neural networks
19th
20th
1950
1960
1970
1980
1990
2000
2010
2017

Outlines

- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

Machine learning

- **Machine Learning—minimization of some loss function for generalizing data sets with models.**
- **Datasets —annotated, indexed, organized**
- **Models —tree, distance, probabilistic, graph, bio-inspired**
- **Optimization —algorithms can minimize the loss.**

Datasets

- Collection
- Storage
- Annotation
- Indexing
- Organization
- Access

Simulators

- Data visualization
- Generate training data
- Algorithm evaluation

Benchmark Metrics

- System functionalities
- System scalability
- System robustness
- System efficiency

Models

- Tree Models
- Distance-based Models
- Probabilistic Models
- Neural Network Models
- Graph-based Models

Models

- Boosting
- Mixed Models
- Ensemble Learning

Outlines

- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

Machine learning and Optimization

- **Machine Learning—minimization of some loss function for generalizing data sets with models.**
- **Datasets —annotated, indexed, organized**
- **Models —tree, distance, probabilistic, graph, bio-inspired**
- **Optimization —algorithms can minimize the loss.**

What is optimization?

- Finding (one or more) minimizer of a function subject to constraints

$$arg\min_{x} f_0(x)$$

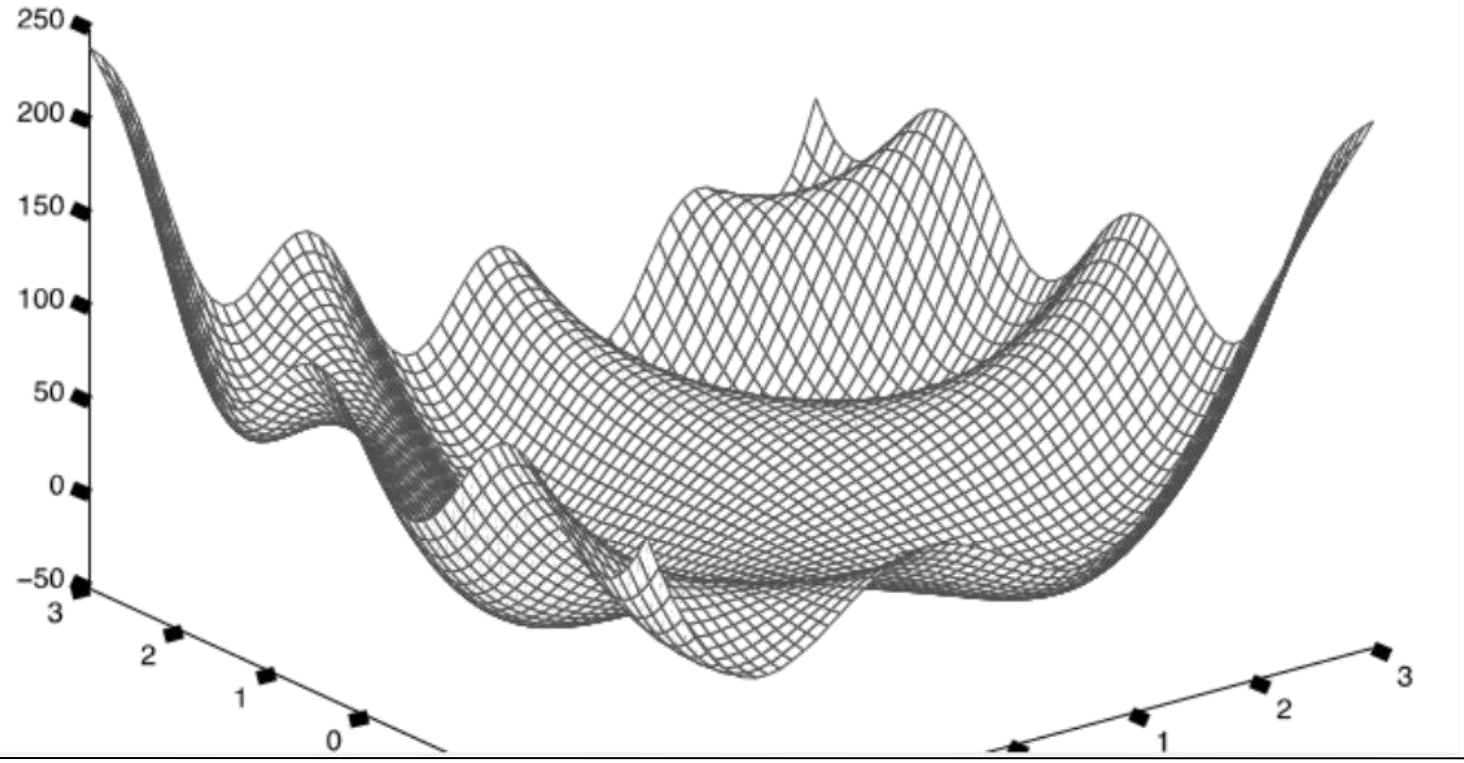
$$s.t.\, f_i(x) \leq 0, i = \{1, \ldots, k\}$$

$$s.t.\, h_j(x) = 0, j = \{1, \ldots, l\}$$

- Most of the machine learning problems are, in the end, optimization problems

General Problem

- Minimize f(x)



Linear Optimization

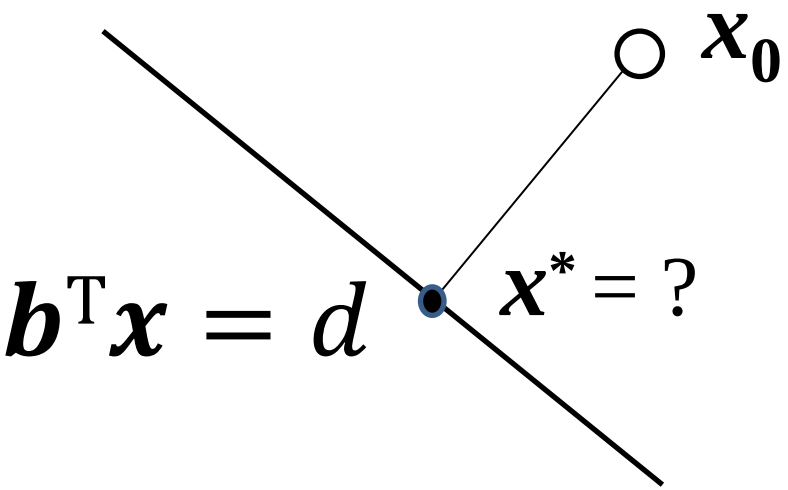
$$\boxed{Y = AX + w} \qquad w \sim \mathcal{N}(0, R)$$

$$X^{*} = \min_{X} (Y - AX)^{\mathrm{T}} R^{-1} (Y - AX)$$

$$\frac{\partial}{\partial X^{T}} (Y - AX)^{\mathrm{T}} R^{-1} (Y - AX) = 0$$

$$\Rightarrow X^{*} = (A^{T} R^{-1} A)^{-1} A^{T} R^{-1} Y$$

Linear Optimization



$$x^* = x_0 - \frac{(b^T x_0 - d)b}{b^T b}$$

$$x^* = \min_x (x - x_0)^T (x - x_0)$$

$$\text{s. t. } b^T x - d = 0$$

Nonlinear Optimization

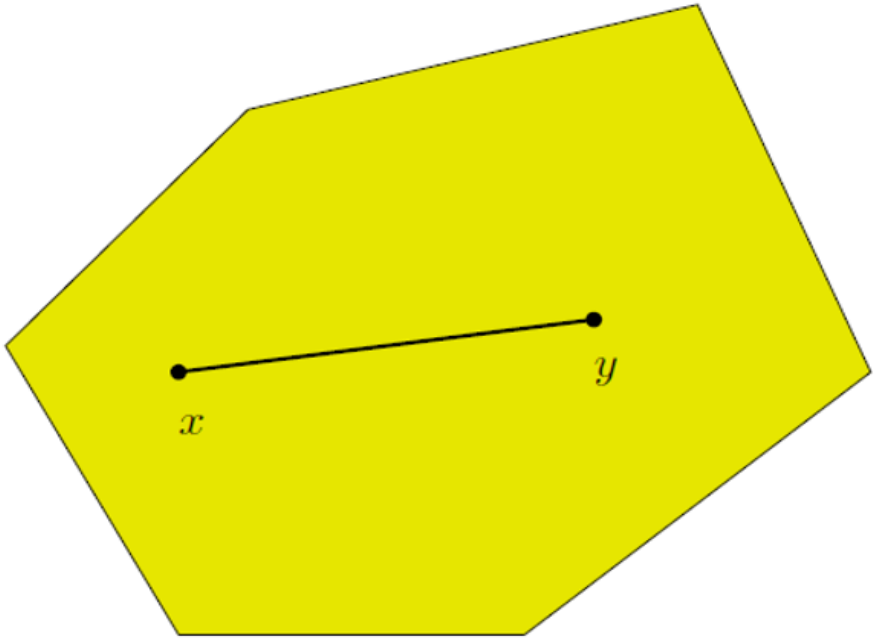
- Convex Optimization
  - Unconstrained optimization
  - Constrained optimization
  - SVMs and Bayesian models
- Non-convex Optimization
  - Heuristic algorithms
  - Random search

What is Convex?

- Convex sets

Def: A set $C \subseteq R$ is convex if for $x,y \in C$; $a \in [0,1]$

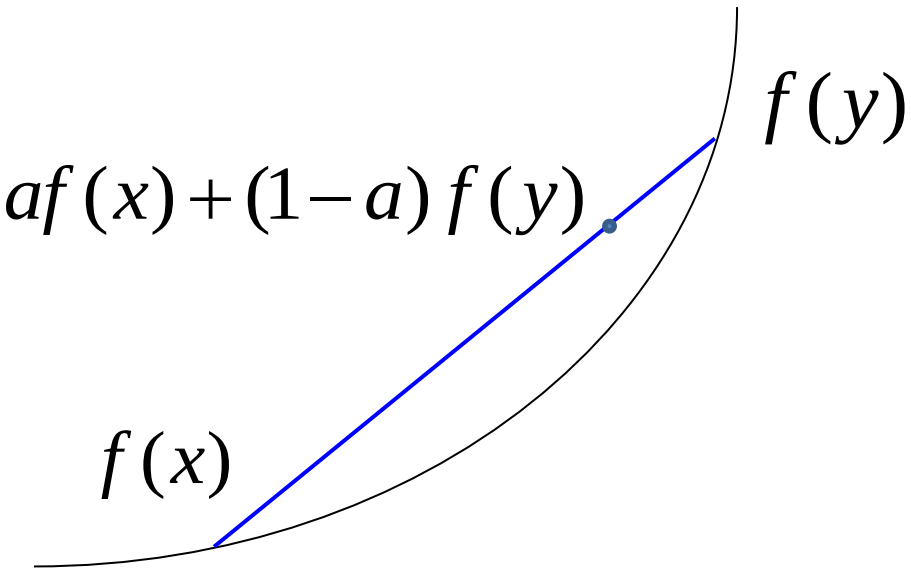
$$ax + (1-a)y \in C$$



What is Convex?

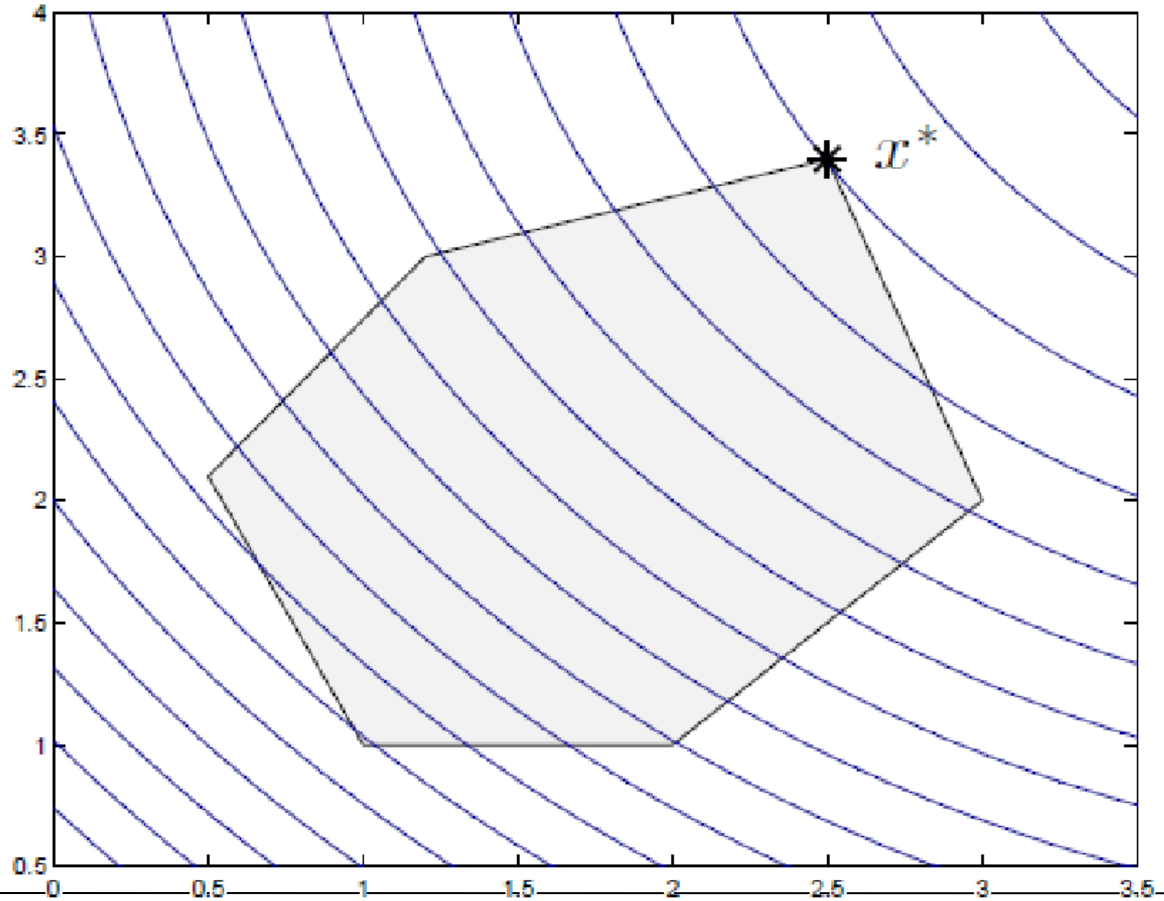
- Convex functions

$$f(ax+(1-a)y) \le af(x)+(1-a)f(y)$$



Convex Optimization

- Local minimizer = Global minimizer



Convex Optimization

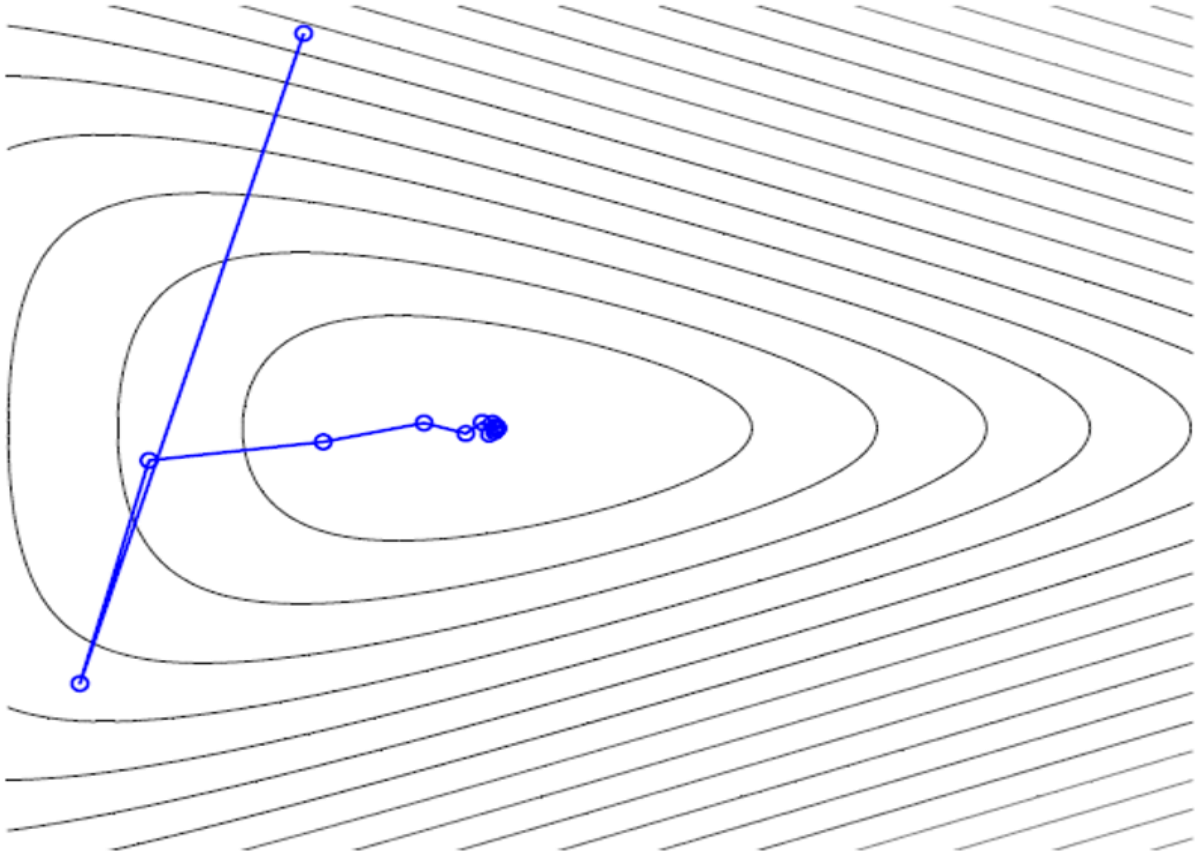
- Unconstrained optimization
  - Gradient descent
  - Gauss-Newton's method
  - Batch learning
  - Stochastic Gradient Descent
- Constrained optimization
  - Lagrange methods
  - Bayesian methods

Convex optimization

- Unconstrained optimization
  - Gradient descent
  - Gauss-Newton's method
  - Batch learning
  - Stochastic Gradient Descent
- Constrained optimization
  - Lagrange methods
  - Bayesian methods

Gradient Descent

$$f(x_{t+1}) = f(x_t) - \eta \nabla f(x_t)^T (x - x_t)$$



Gauss-Newton's Method

- Idea: use a second-order approximation to function

$$f(x+\Delta x) \approx f(x) + \nabla f(x)^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2 f(x) \Delta x$$

- Choose Δx to minimize above:

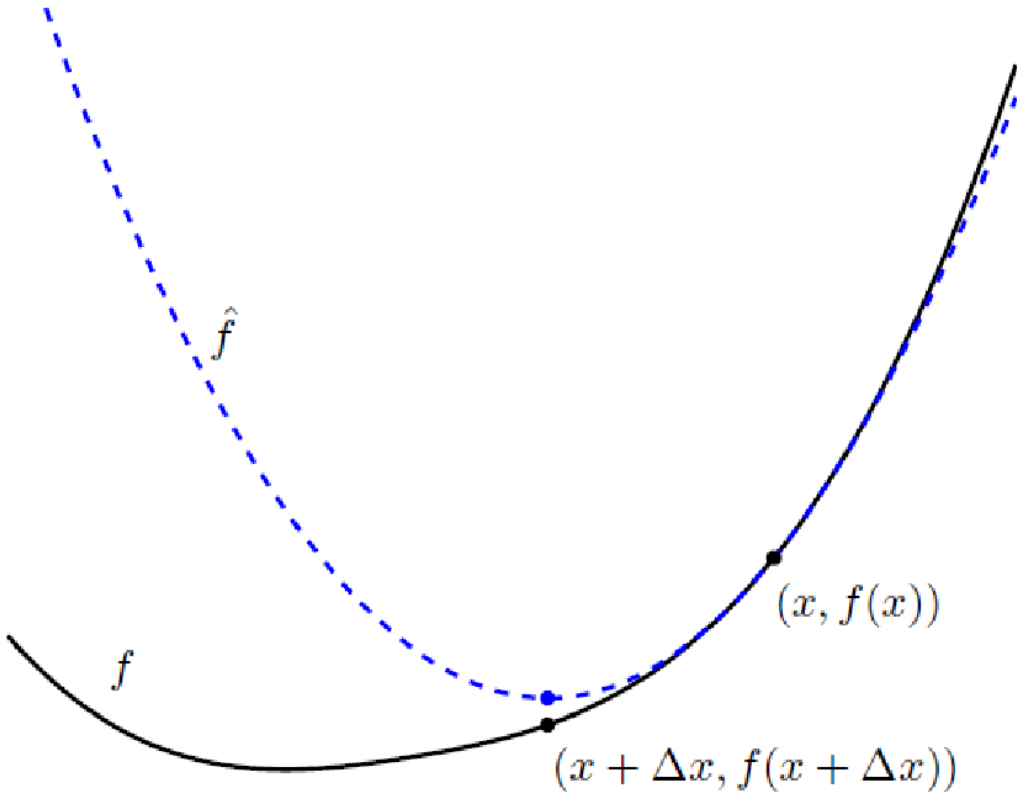
$$\Delta x = -[\nabla^2 f(x)]^{-1} \nabla f(x)$$

- This is descent direction:

$$\nabla f(x)^T \Delta x = -\nabla f(x)^T [\nabla^2 f(x)]^{-1} \nabla f(x) < 0$$

Gauss-Newton's Method

$\hat{f}$ is 2-order approximation, f is true function.



Batch Gradient Descent

- Minimize empirical loss, assuming it's convex and unconstrained
  - Gradient descent on the empirical loss
  - At each step:

$$w^{(k+1)} \leftarrow w^{(k)} - \eta_t \left(\frac{1}{n}\sum_{i=1}^{n}\frac{\partial L(w, x_i, y_i)}{\partial w}\right)$$

  - Note: ate each step, gradient is the average of the gradient for all samples (i=1,…n)
  - Very slow when n is very large

Stochastic Gradient Descent

- Alternative: compute gradient from just one (or a few samples)
- Known as stochastic gradient descent:
  - At each step,

$$w^{(k+1)} \leftarrow w^{(k)} - \eta_t \frac{\partial L(w, x_i, y_i)}{\partial w}$$

(choose one sample i and compute gradient for that sample only)

Convex Optimization

- Unconstrained optimization
  - Gradient descent
  - Gauss-Newton's method
  - Batch learning
  - Stochastic Gradient Descent
- Constrained optimization
  - Lagrange methods
  - Bayesian methods

Lagrange Methods

- Start with optimization problem:

$$\arg\min_{x} f_0(x)$$

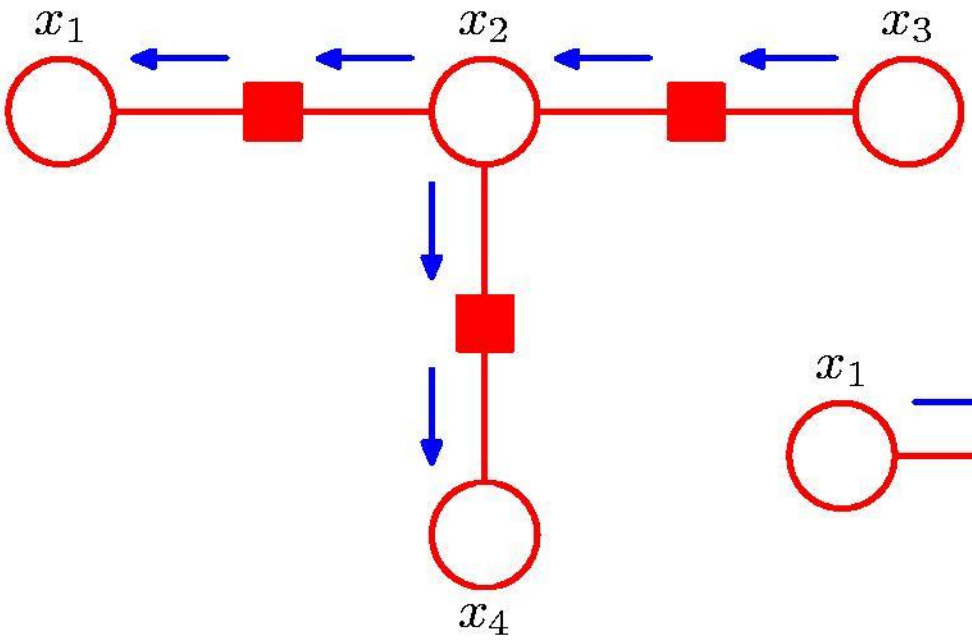
$$s.t.\, f_i(x) \leq 0, i = \{1, \dots, k\}$$

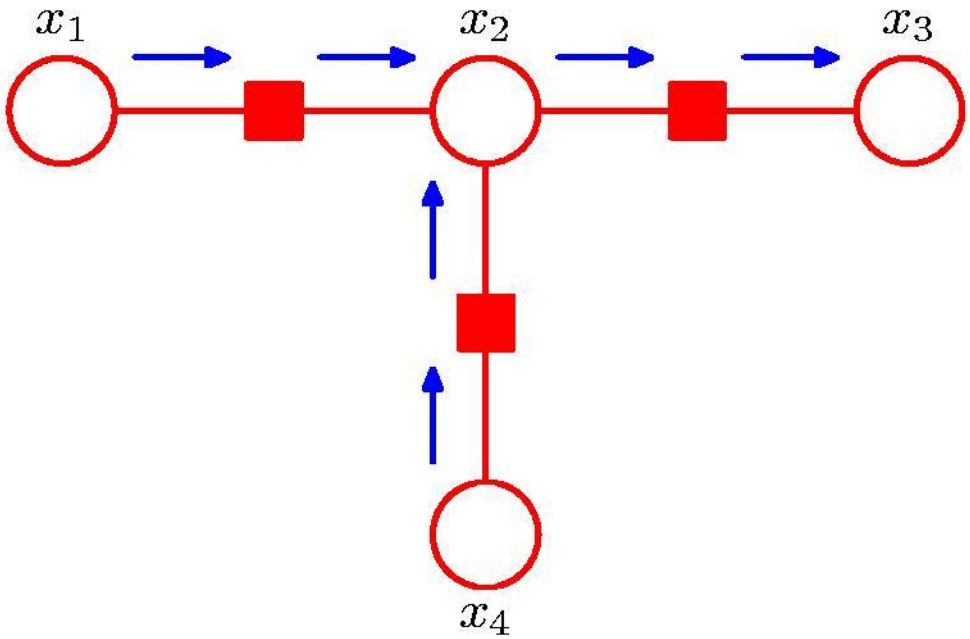
$$s.t.\, h_j(x) = 0, j = \{1, \dots, l\}$$

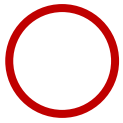
- Is equivalent to min-max optimization:

$$\arg\min_{x} \left[\max_{\lambda \geq 0, \gamma > 0} \left(f_0(x) + \sum_{i=1}^{k} \lambda_i f_i(x) + \sum_{j=1}^{l} \gamma_j h_j(x) \right) \right]$$

Bayesian Methods







Random variable

factor

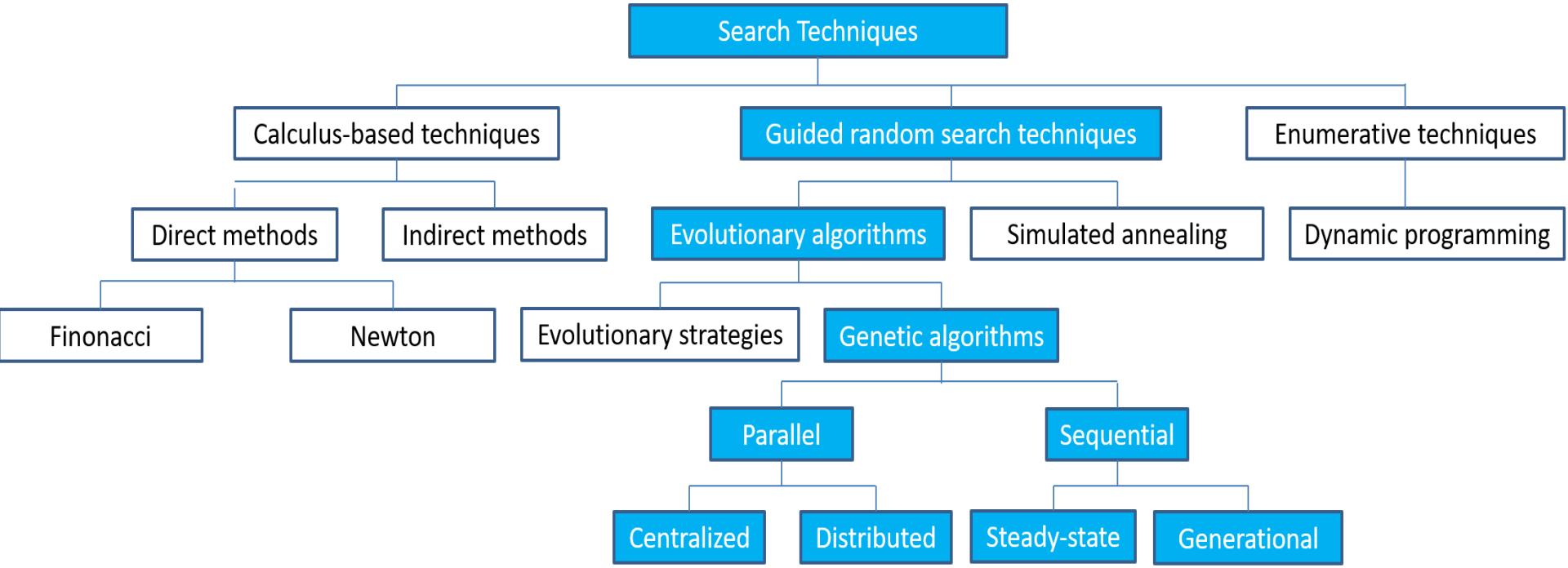
Convex Optimization for Machine Learning

- Support Vector Machine
  - Lagrange methods
- Bayesian Models:
  - Expectation-Maximization methods
  - Variational methods
  - Graph optimization

Non-convex Optimization

- Convex Optimization
  - Unconstrained optimization
  - Constrained optimization
- Non-convex Optimization
  - Heuristic algorithms
  - Random search

Heuristic and Random Search



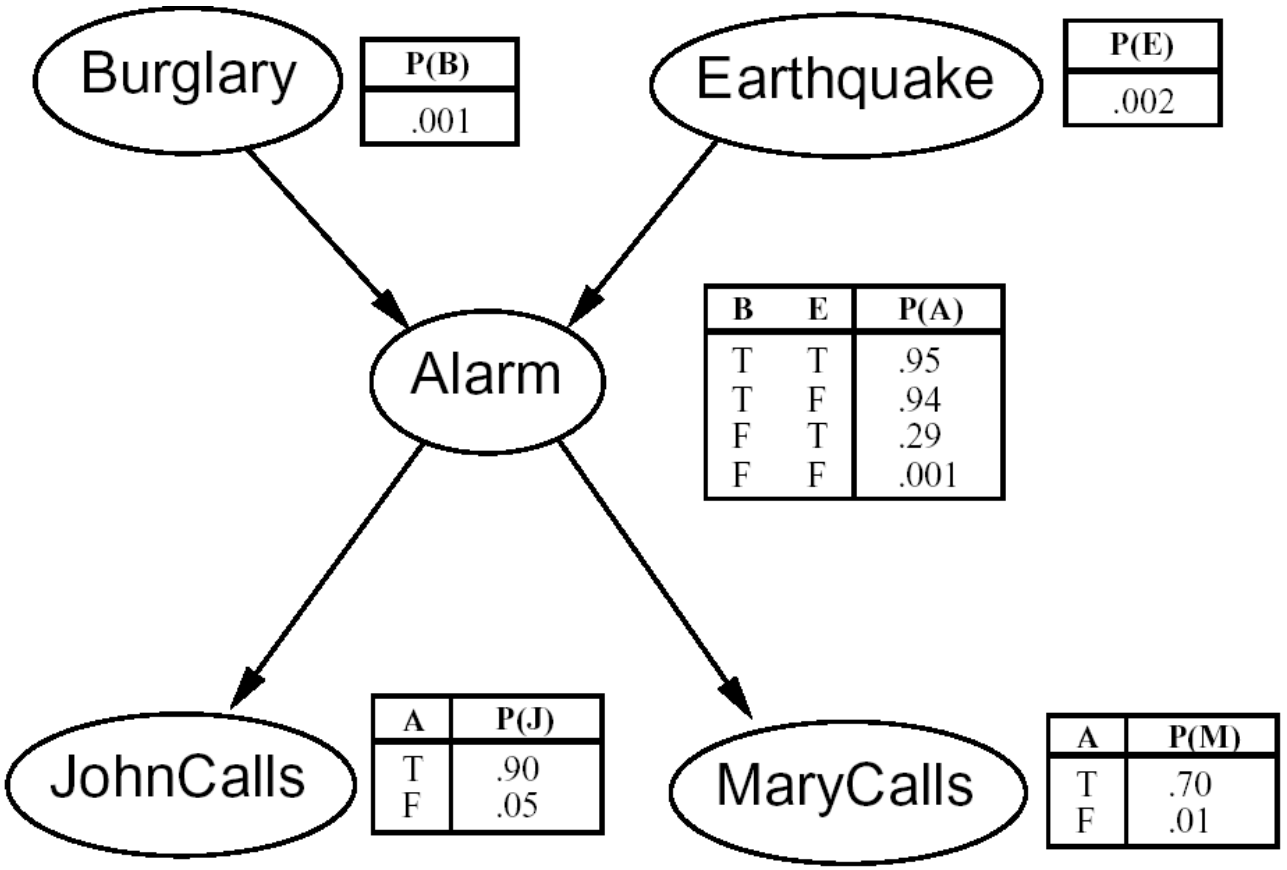
Outlines

- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

Algorithms

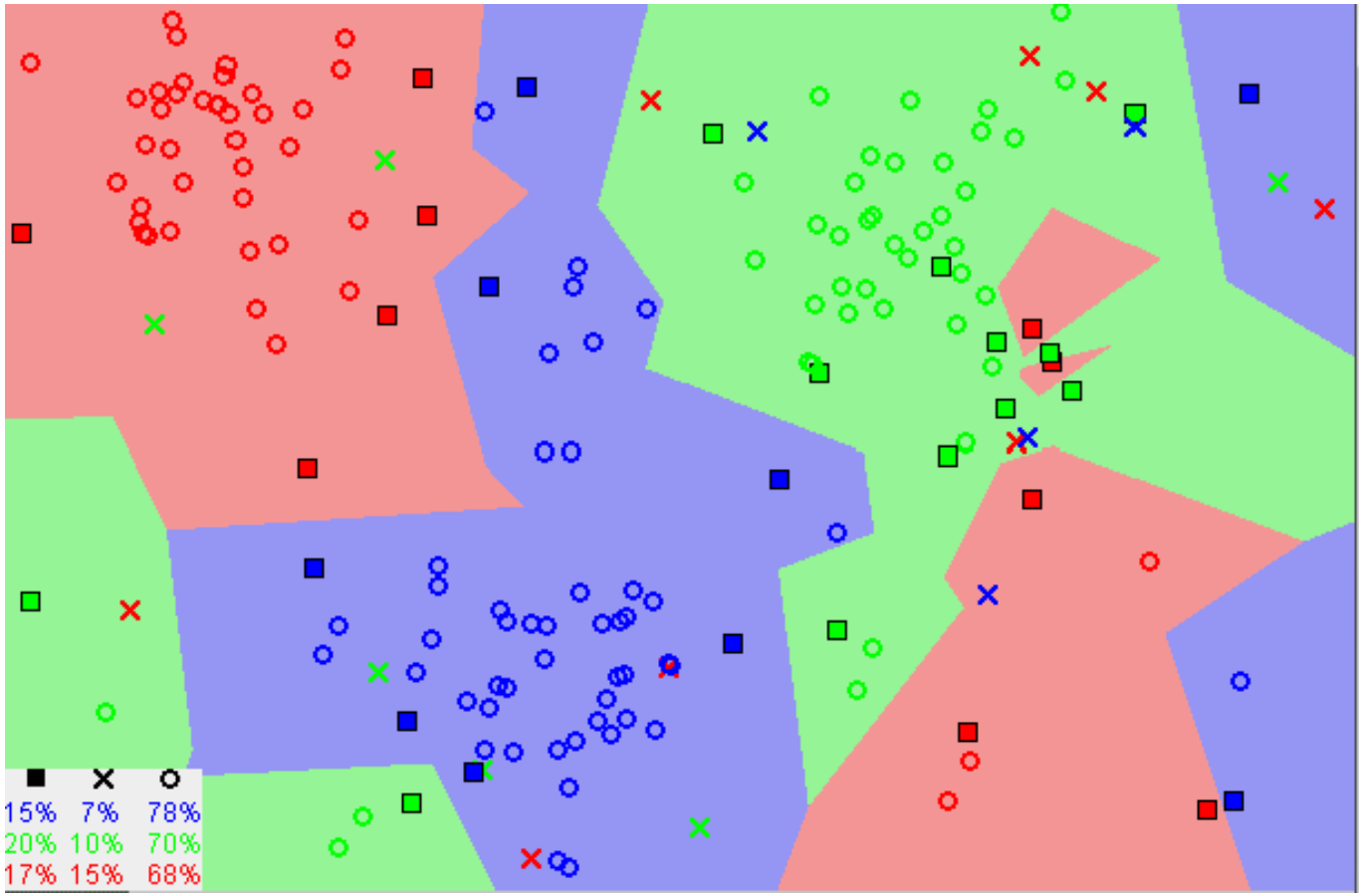
- **Bayes**
- **KNN and K-means**
- **Decision tree**
- **Support Vector Machine**
- **Boosting and Ensemble Learning**
- **Linear Statistical Learning (PCA, ICA, NMF)**
- **Nonlinear Statistical Learning (Manifold learning)**
- **Deep Neural Networks**
- **Generative Adversarial Networks**
- **Bayesian Networks**
- **Reinforcement Learning**
- **Federated Learning**

Bayes


Burglary
P(B)
.001
Earthquake
P(E)
.002
Alarm
B E P(A)
T T .95
T F .94
F T .29
F F .001
JohnCalls
A P(J)
T .90
F .05
MaryCalls
A P(M)
T .70
F .01

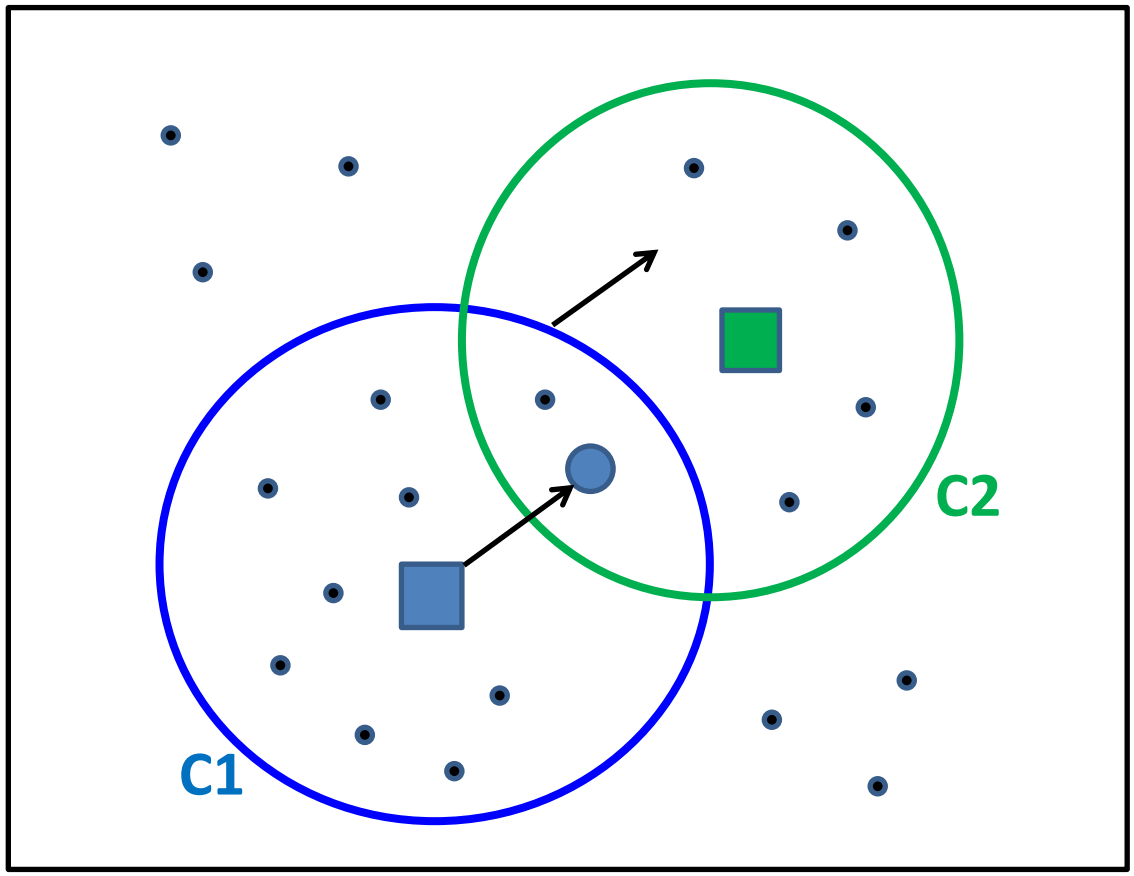
K-Nearest Neighbors

- Use training data for classification



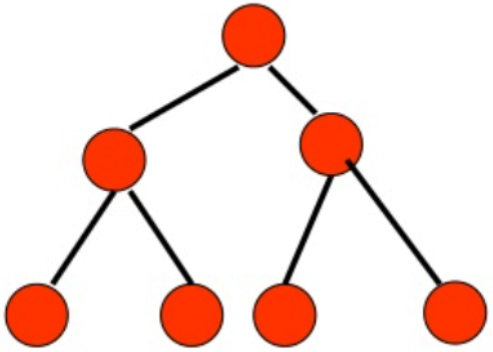
K-Means

- Mean-shift for clustering

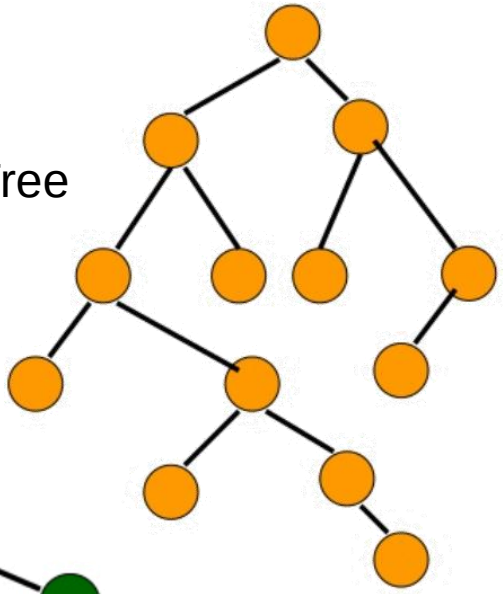


Decision Tree

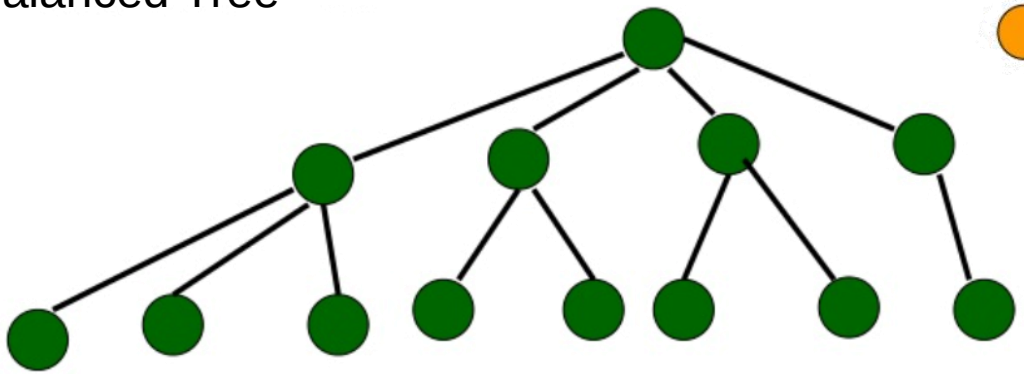
- Types of Decision Tree:



Balanced Tree



Deep Tree



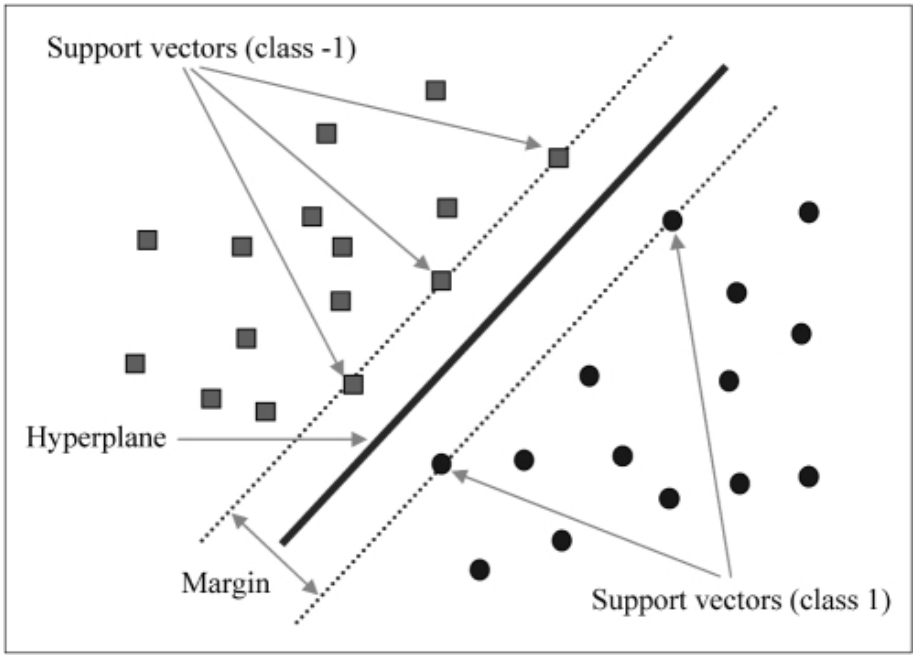
Bushy Tree

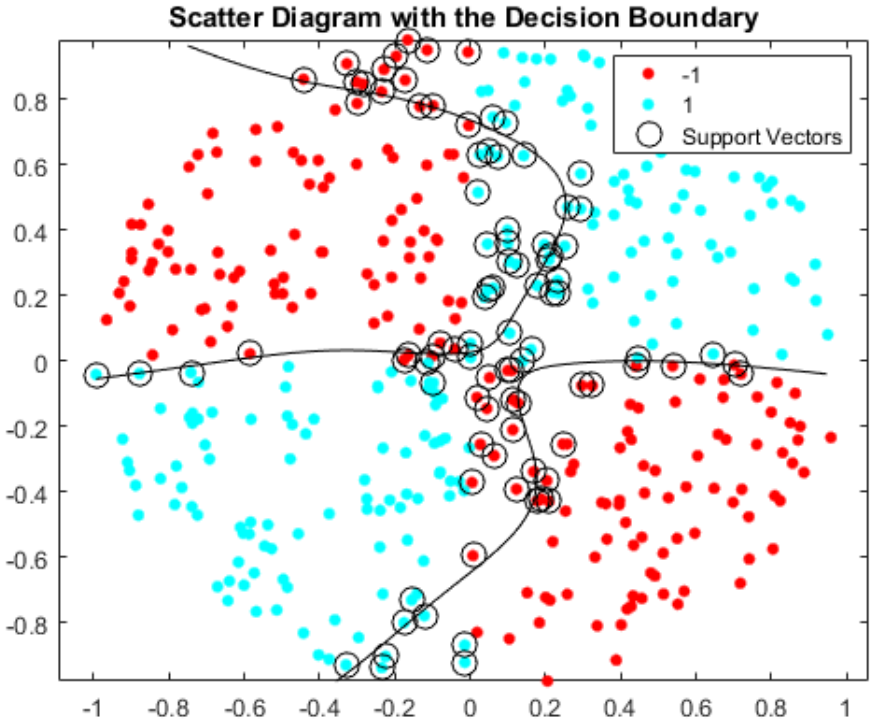
SVM

- eg. Linear SVM: $$\arg\min_{w} \sum_{i=1}^{n} ||w||^2 + C \sum_{i=1}^{n} \xi_i$$

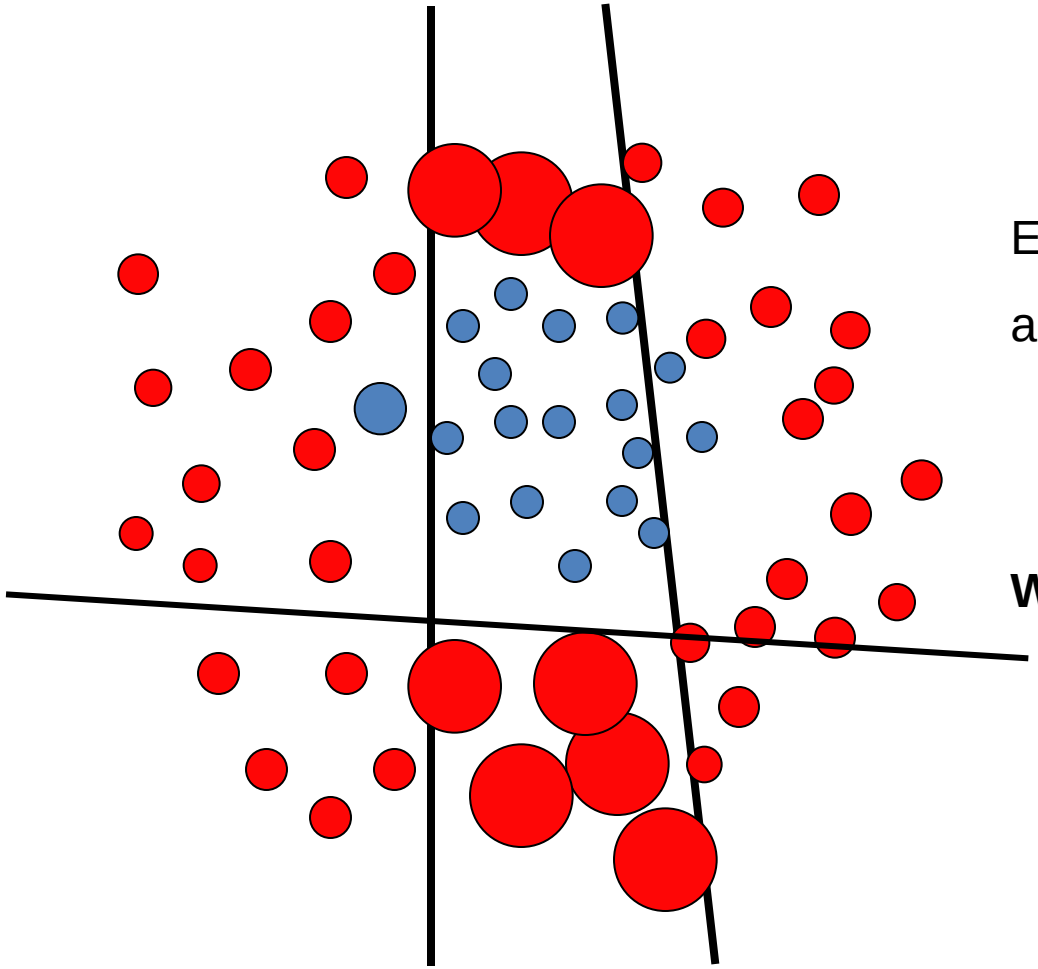
$$\text{s.t. } 1 - y_i x_i^T w \leq \xi_i$$

$$\xi_i \geq 0$$





Boosting



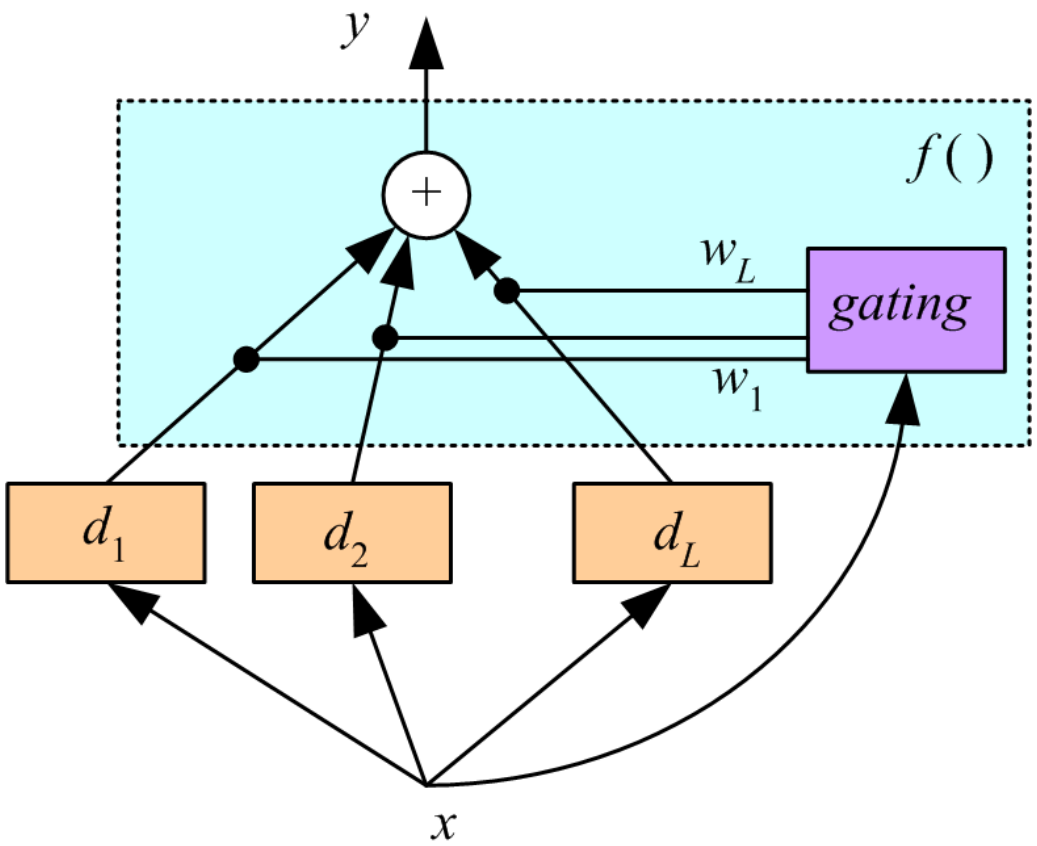
Each data point has a class label:

$$y_t = \begin{cases} +1 \text{ (●)} \\ -1 \text{ (●)} \end{cases}$$

We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

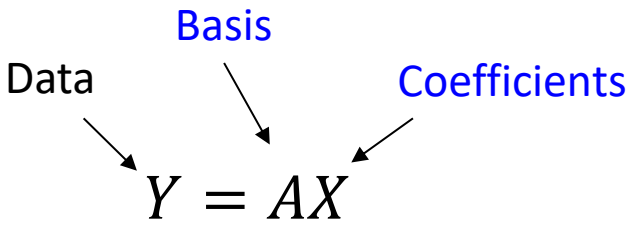
Ensemble Learning



$$y = \sum_{j=1}^{L} w_j d_j$$

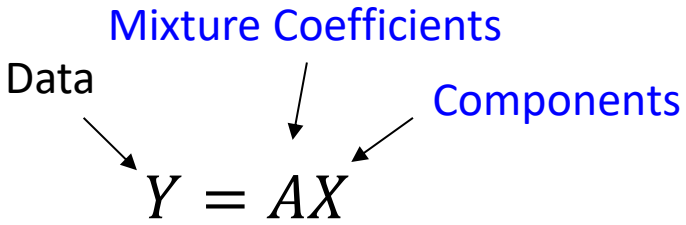
Linear Statistical Learning

- PCA



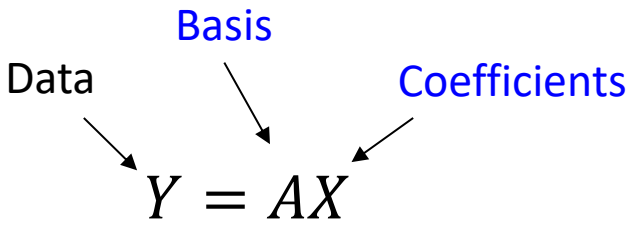
$A_i \perp A_j$

- ICA



$\max\ I(X)$

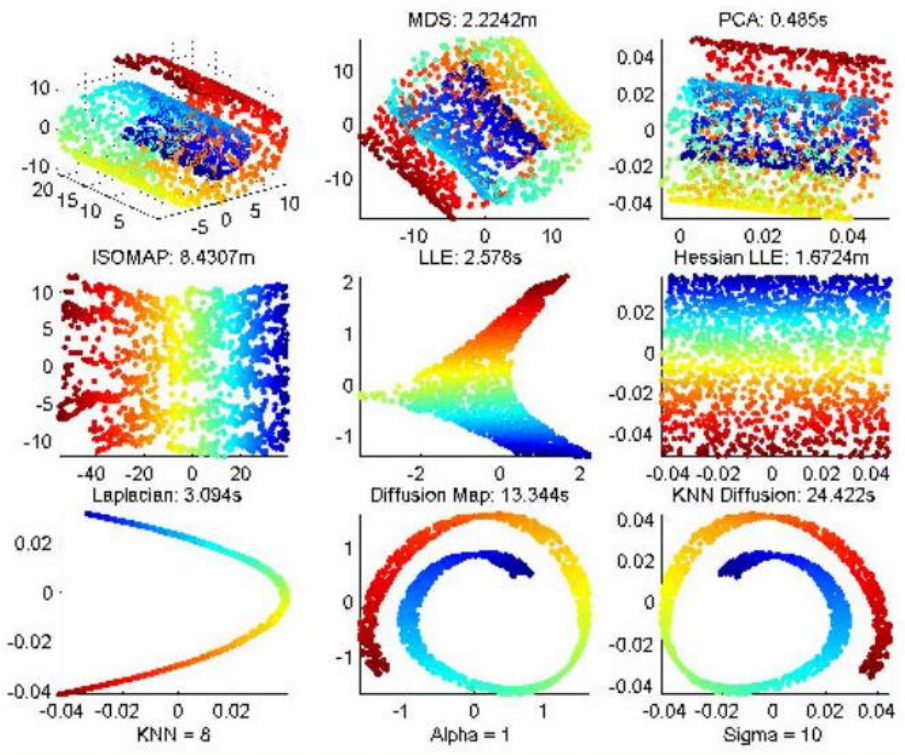
- NMF

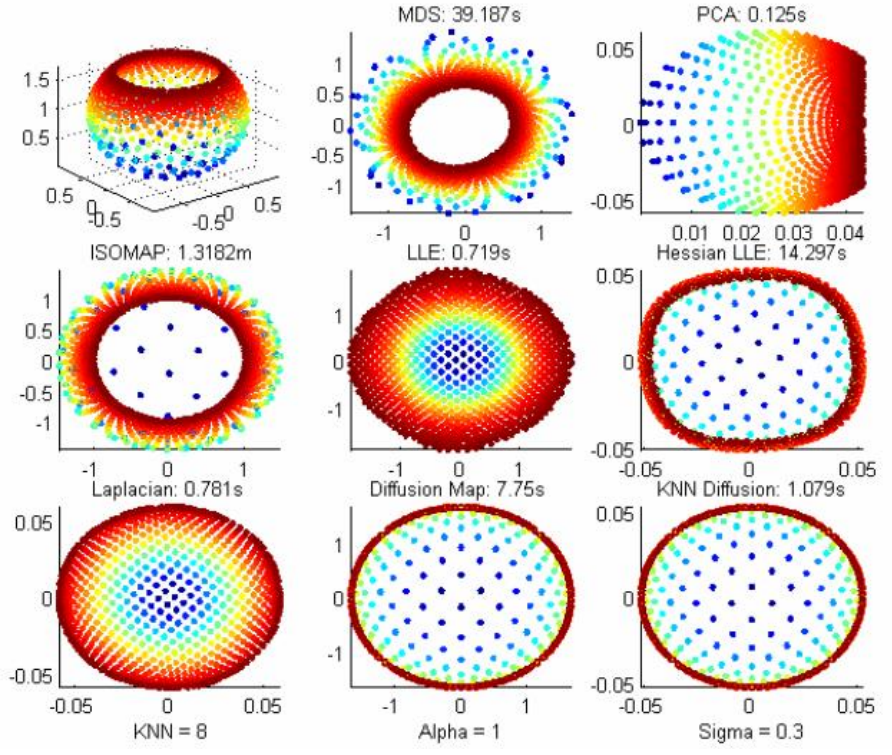


$A, X > 0$

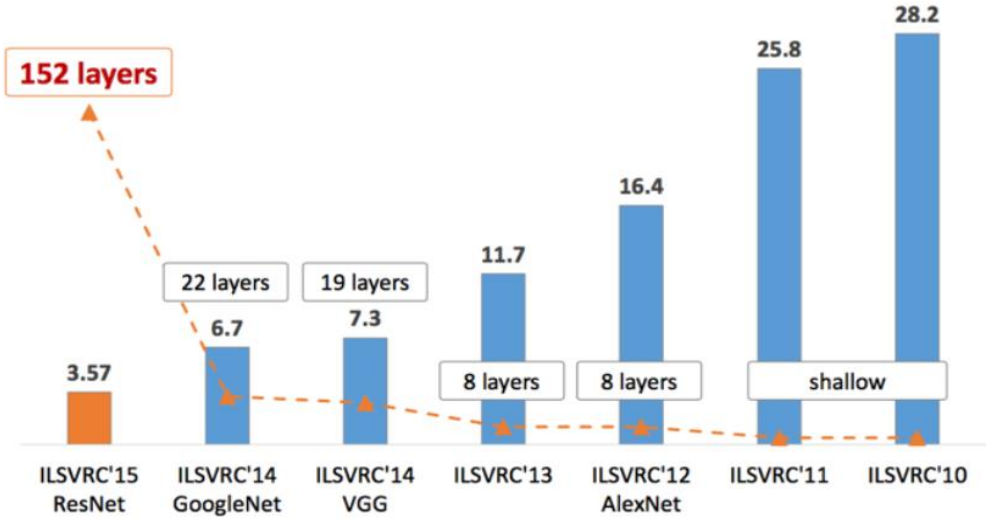
Nonlinear Statistical Learning

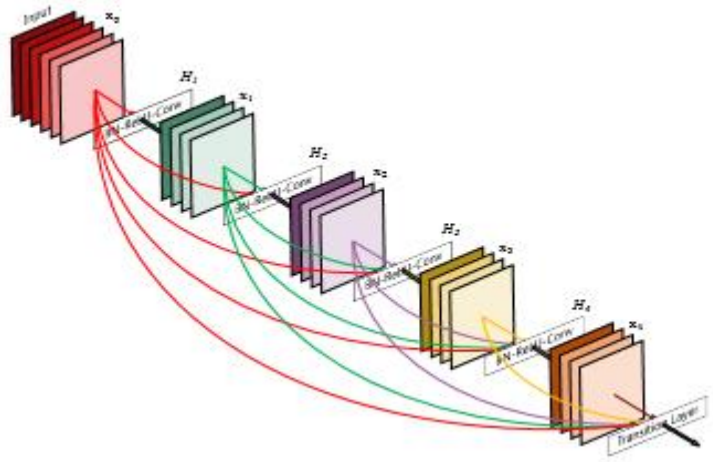
- Manifold learning





Deep Neural Networks

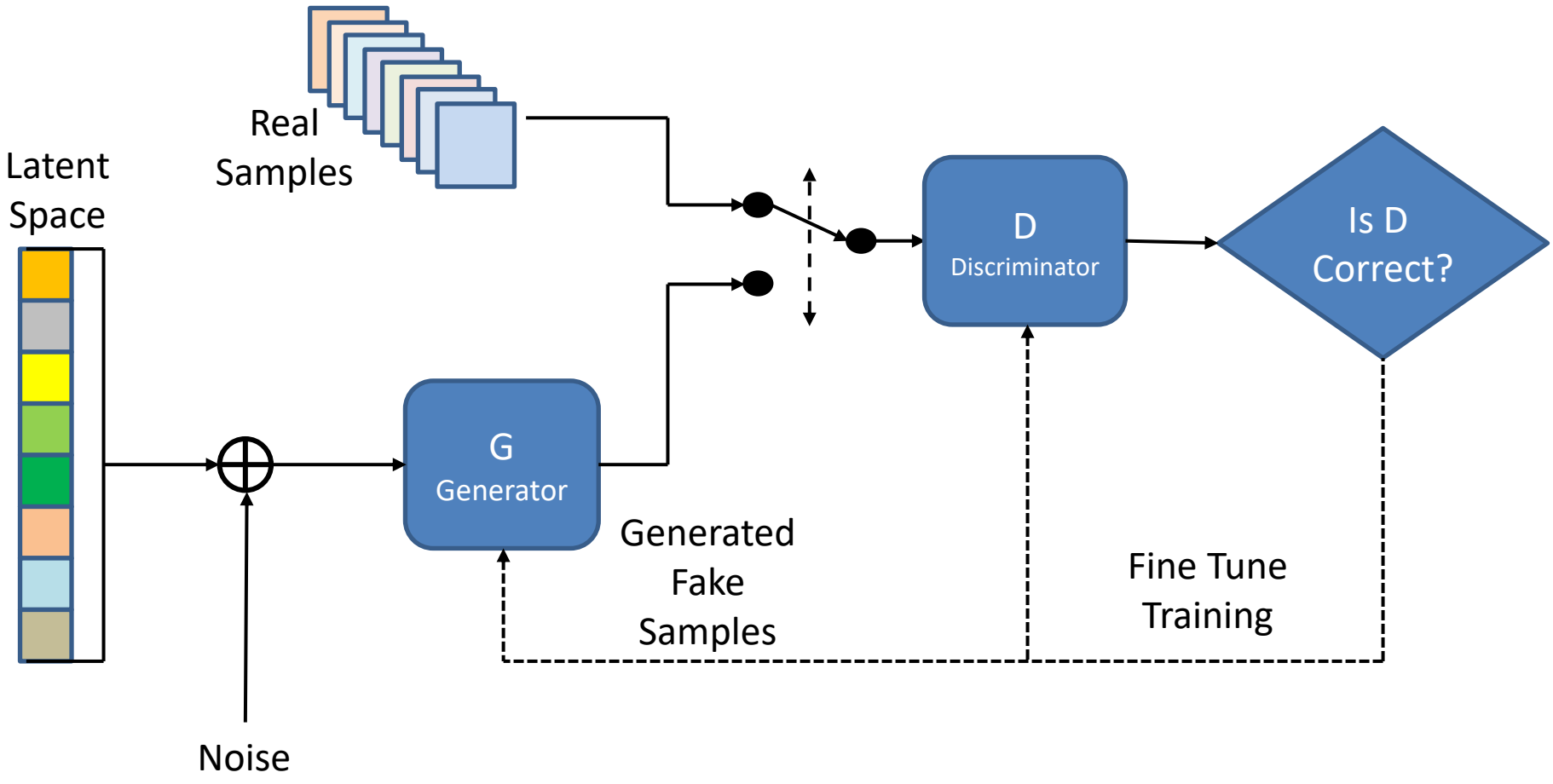




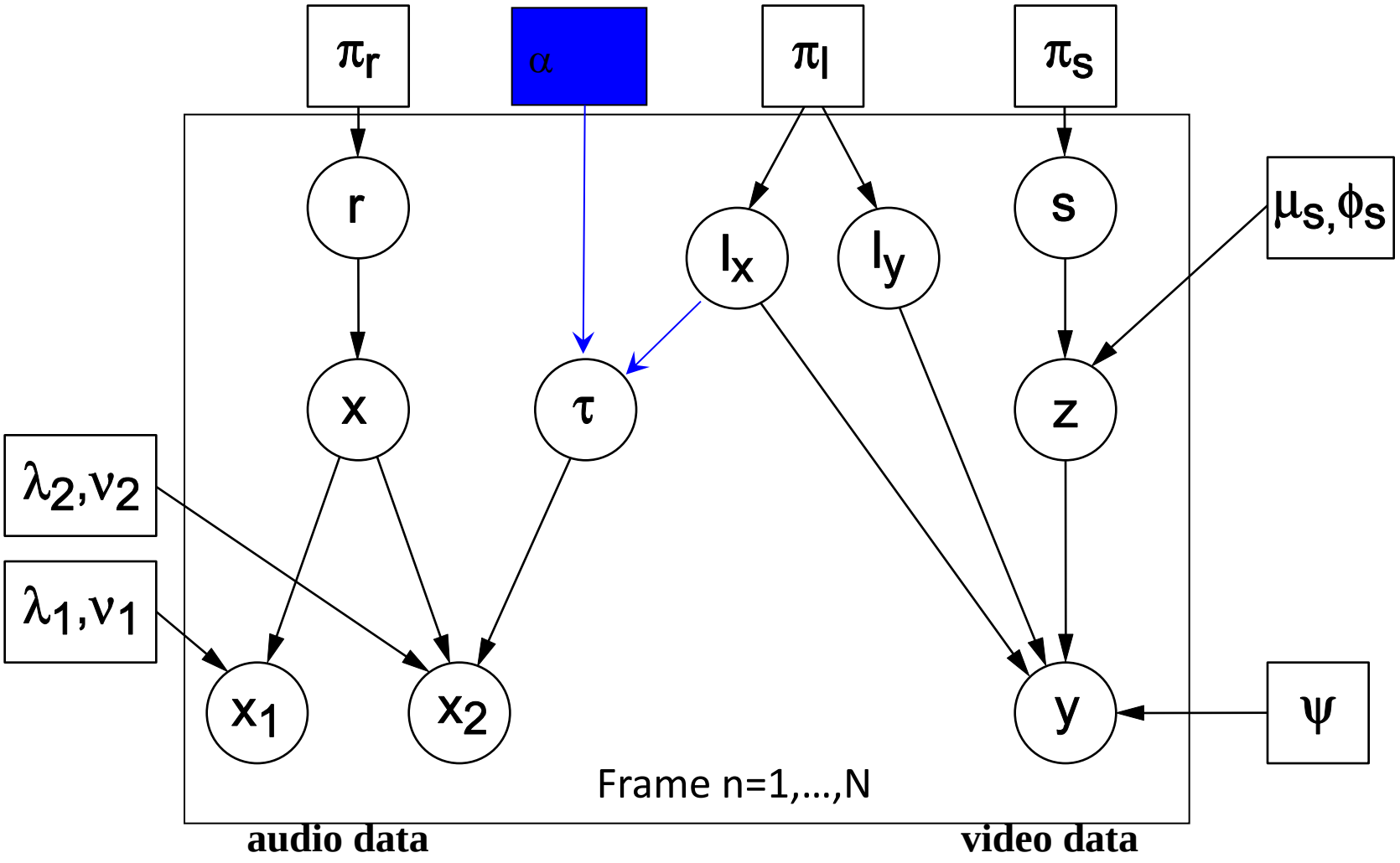
- Task：recognition
- Dataset：ILSVRC

■ Huang G, Liu Z, Weinberger K Q, et al. Densely connected convolutional networks[J]. arXiv preprint arXiv:1608.06993, 2016.

Generative Adversarial Networks

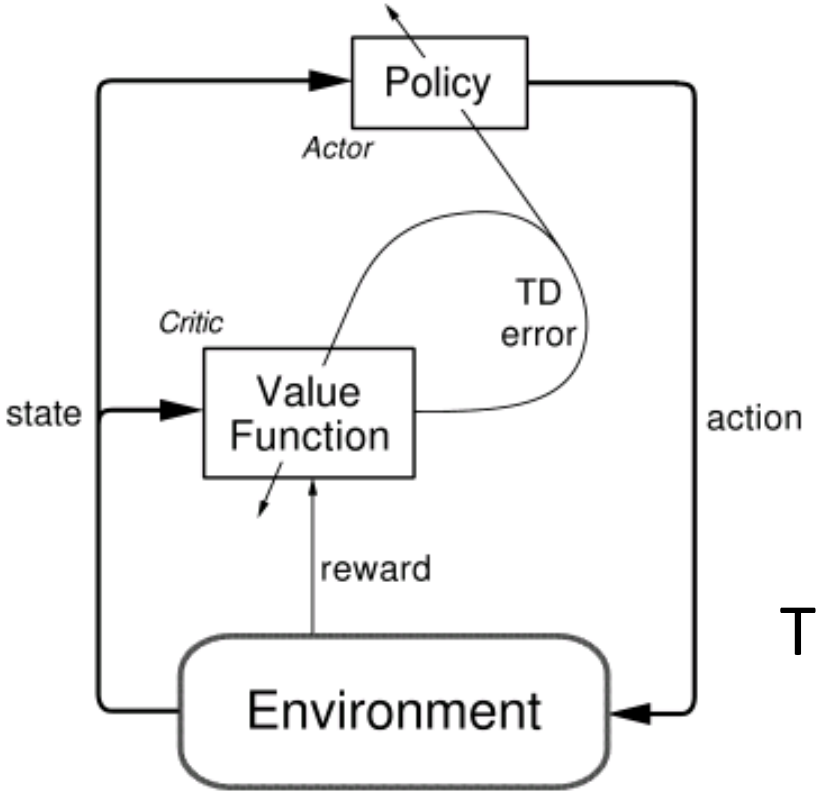


Bayesian Networks



Reinforcement Learning

- State, action, and Reward

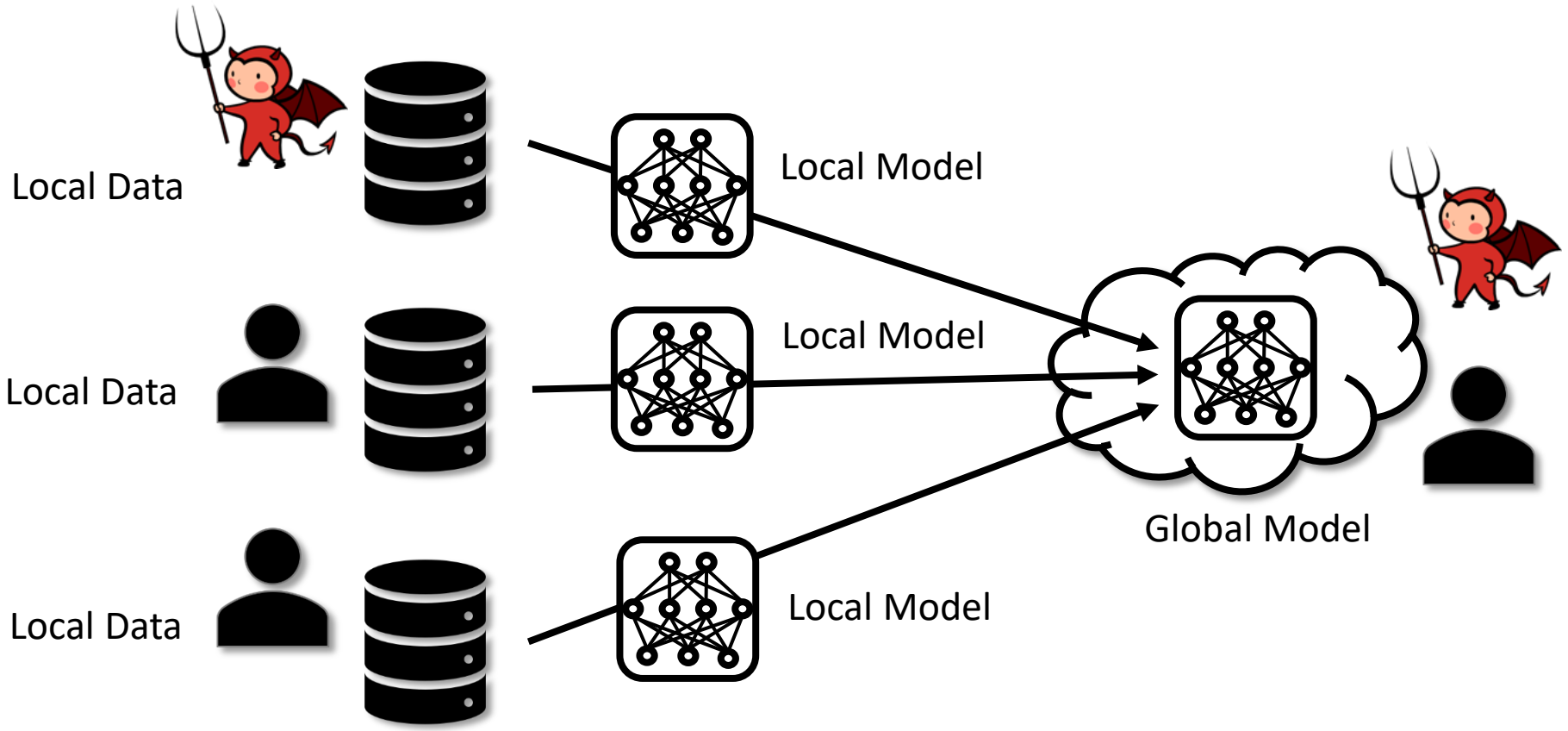


Update: Policy Function
Value Function

TD Error: Temporal Difference between Real Reward and Estimated Reward

Federated Learning

- Collaborative Learning



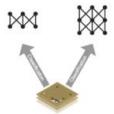
Outlines

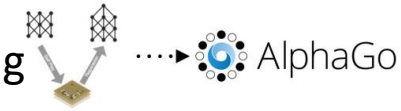
- Framework
- Problem Statement
- Related Areas
- History
- Datasets and Learning Models
- Optimization Methods
- Algorithms
- Examples

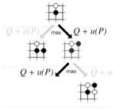
Alphago

Alphago VS. 李世石

Alphago VS. 柯洁

Supervised learning 

Reinforcement learning 

Priori knowledge 

Better network structure

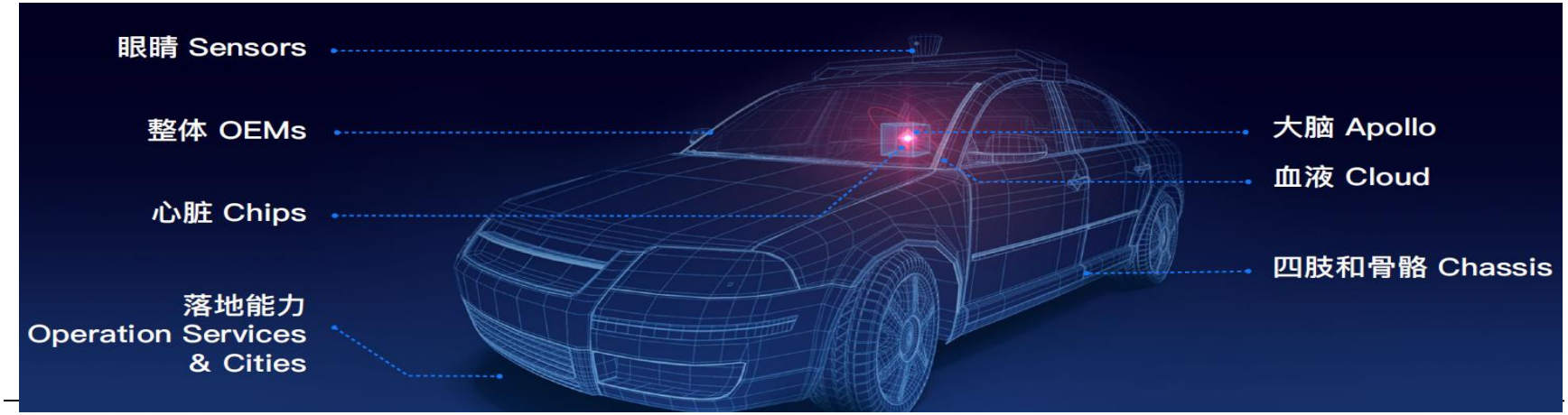
Enhance the role of Reinforcement learning

Autonomous Driving

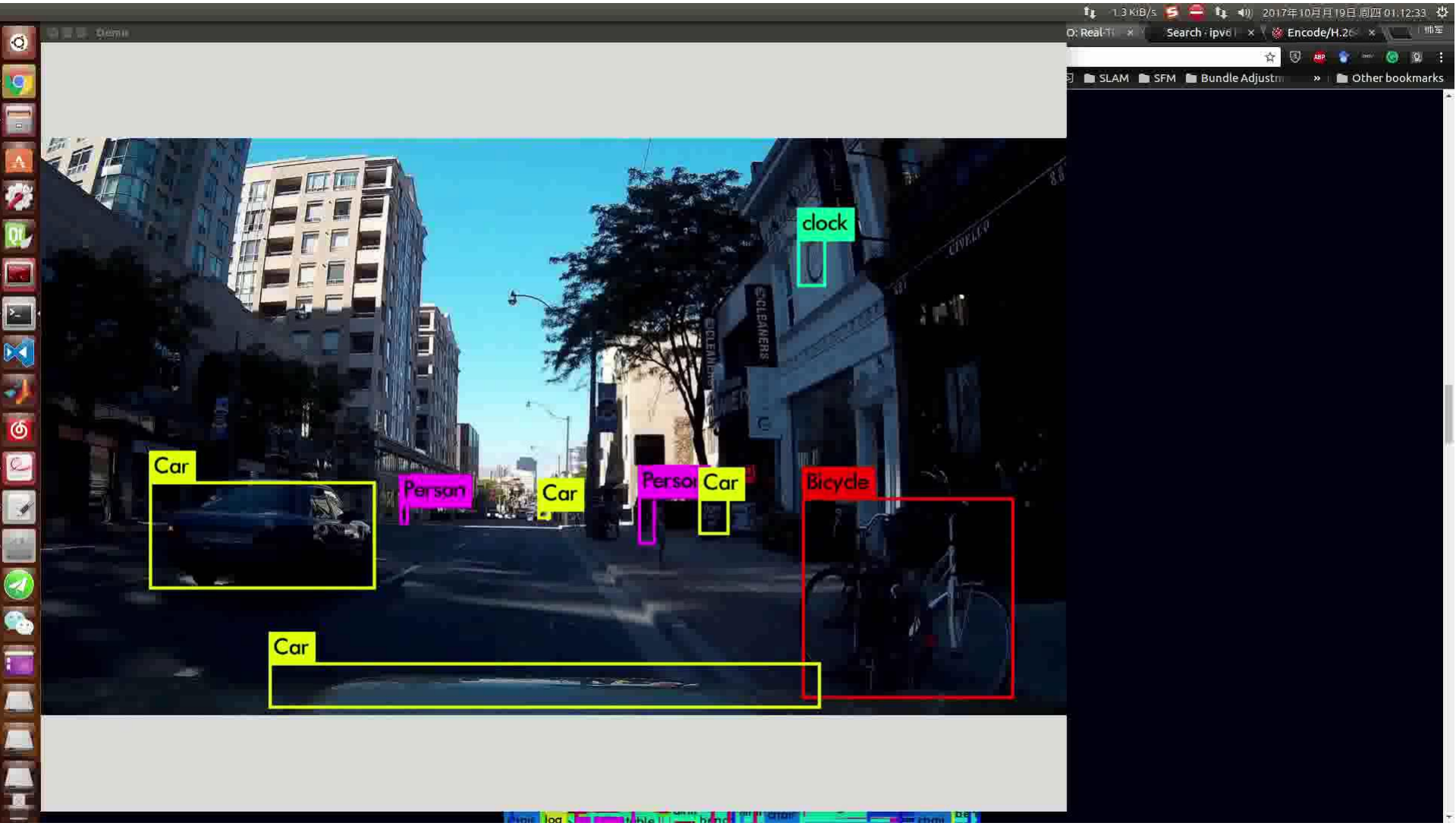
2014 Google

2016 Tesla

2017 apollo



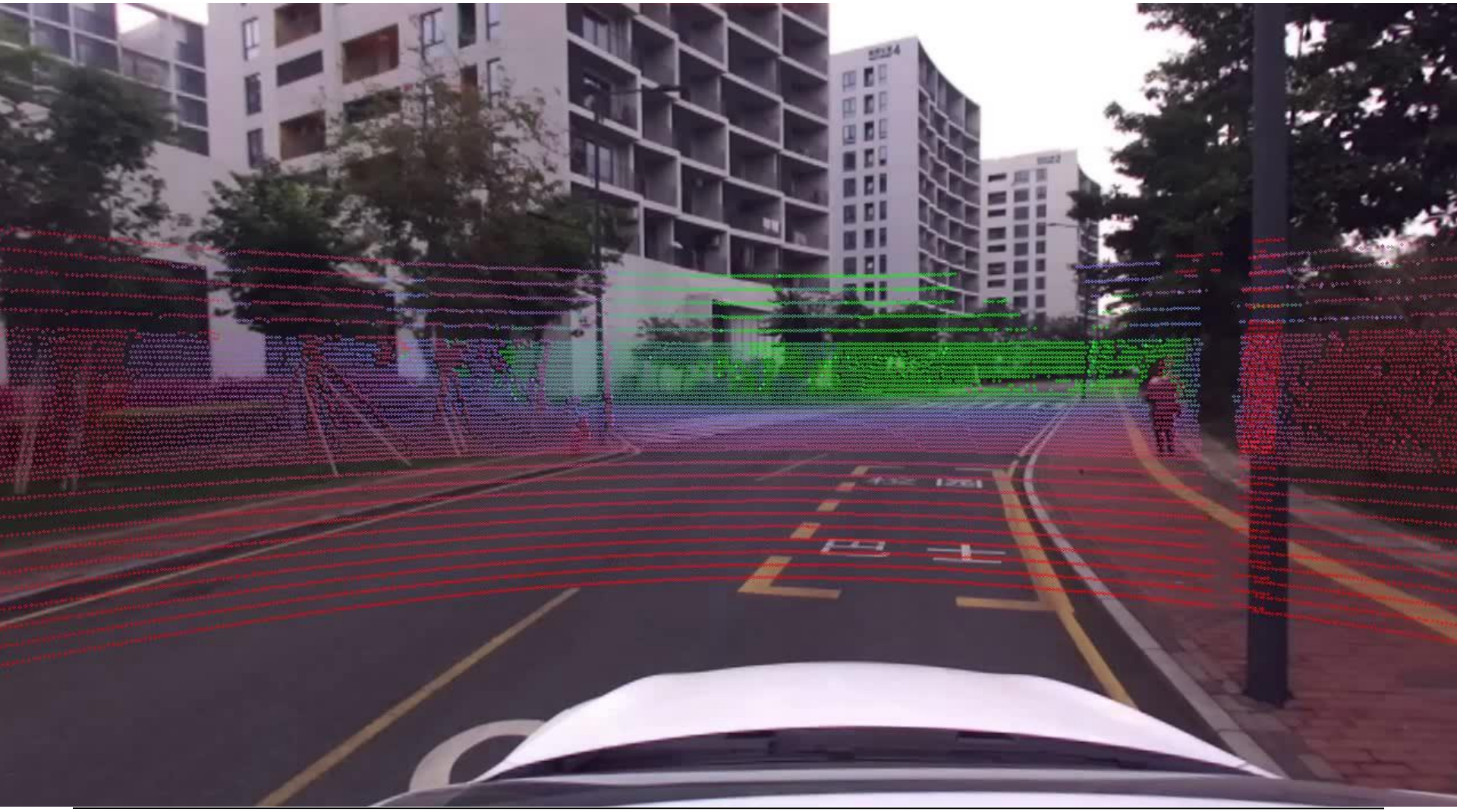
Object Detection-YOLOv3



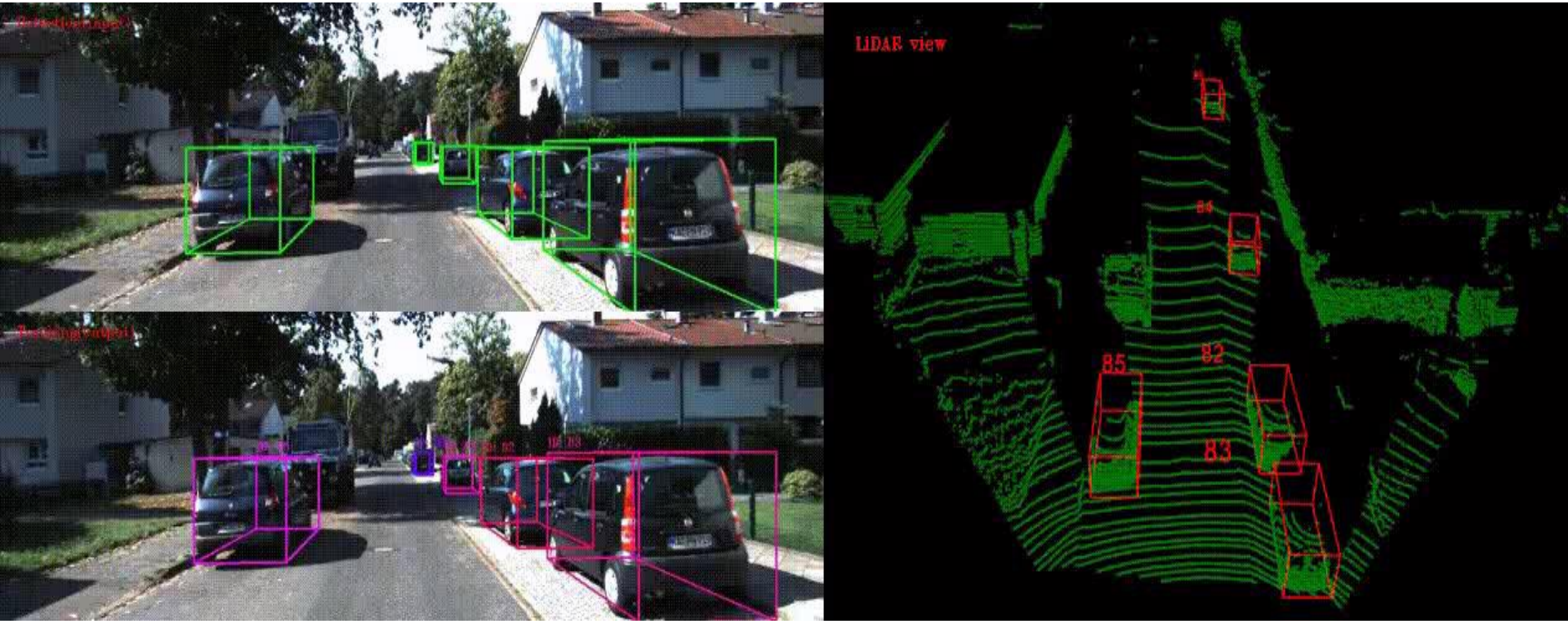
Instance Segmentation

Lane & Sign Detection

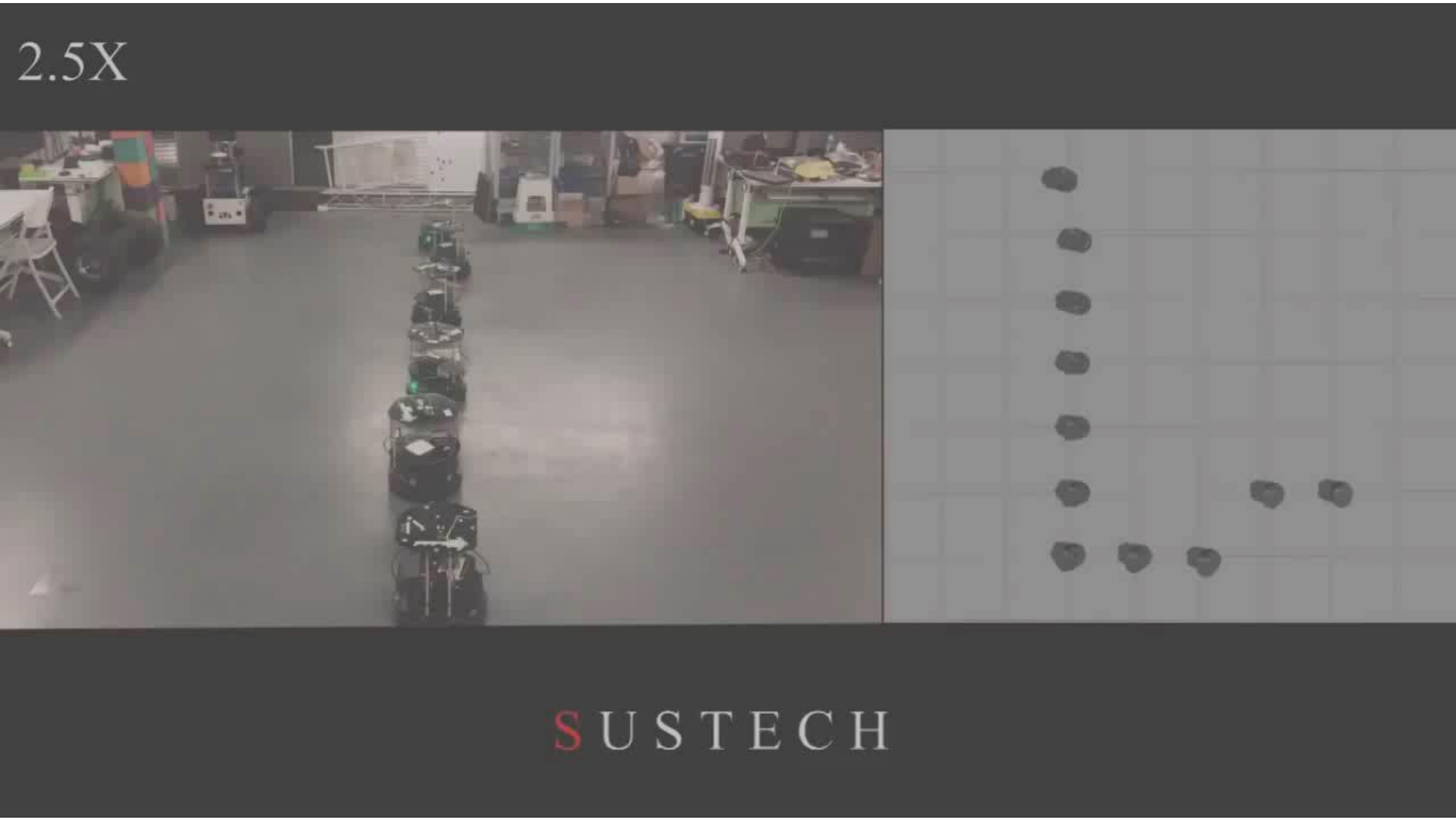
2D-3D Fusion



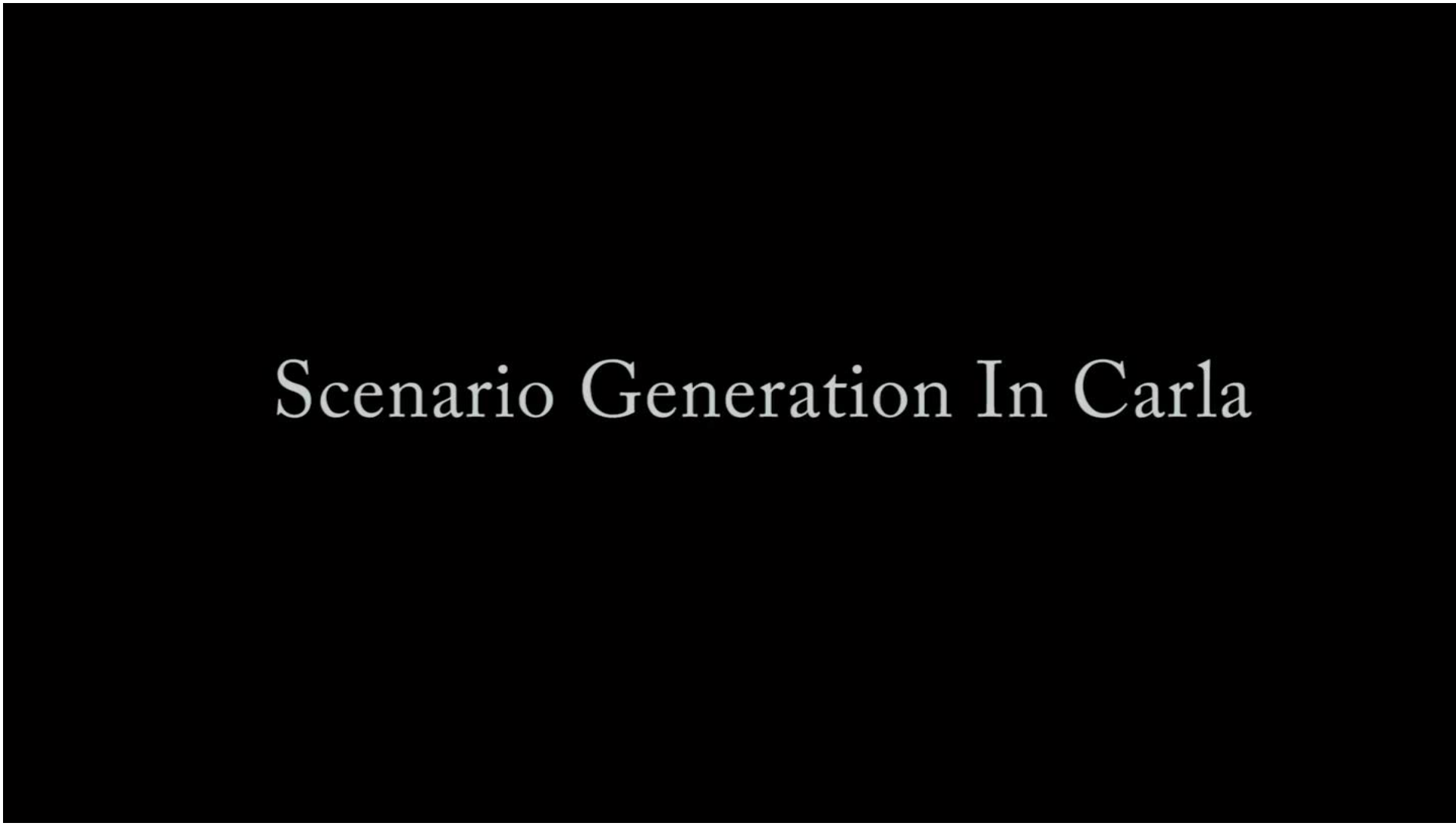
2D-3D Fusion for Tracking



Reinforcement Learning for Navigation



Federated Learning for Map Fusion



Datasets and Benchmark Metrics

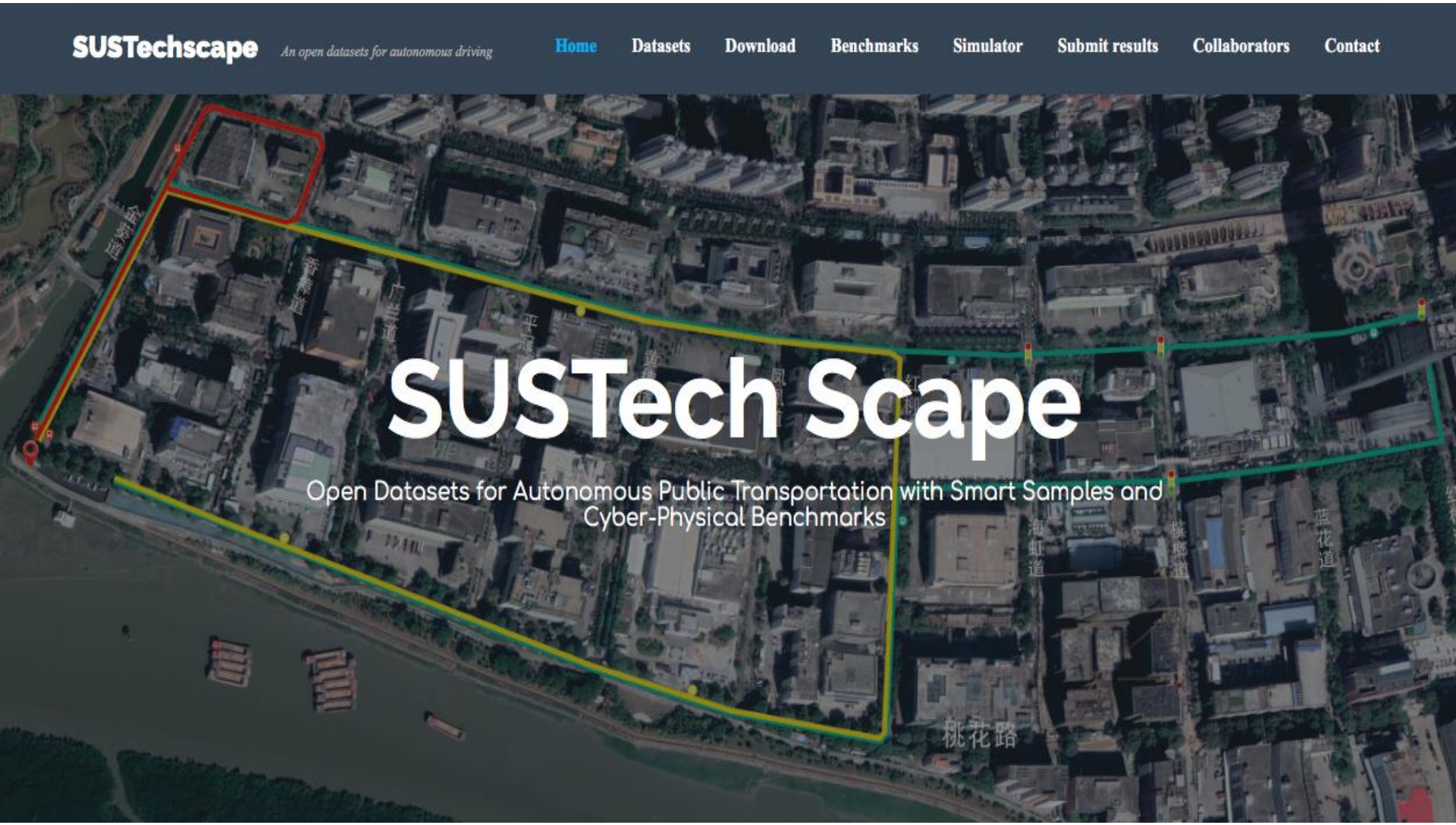
Datasets

- KITTI
- Udacity
- Waymo
- NuScences
- CityScapes
- Apolloscape
- SUSTech Scape

Metrics

- Correctness
- Robustness

Our Dataset


SUSTechscape
An open datasets for autonomous driving
Home
Datasets
Download
Benchmarks
Simulator
Submit results
Collaborators
Contact
SUSTech Scape
Open Datasets for Autonomous Public Transportation with Smart Samples and Cyber-Physical Benchmarks
桃花路

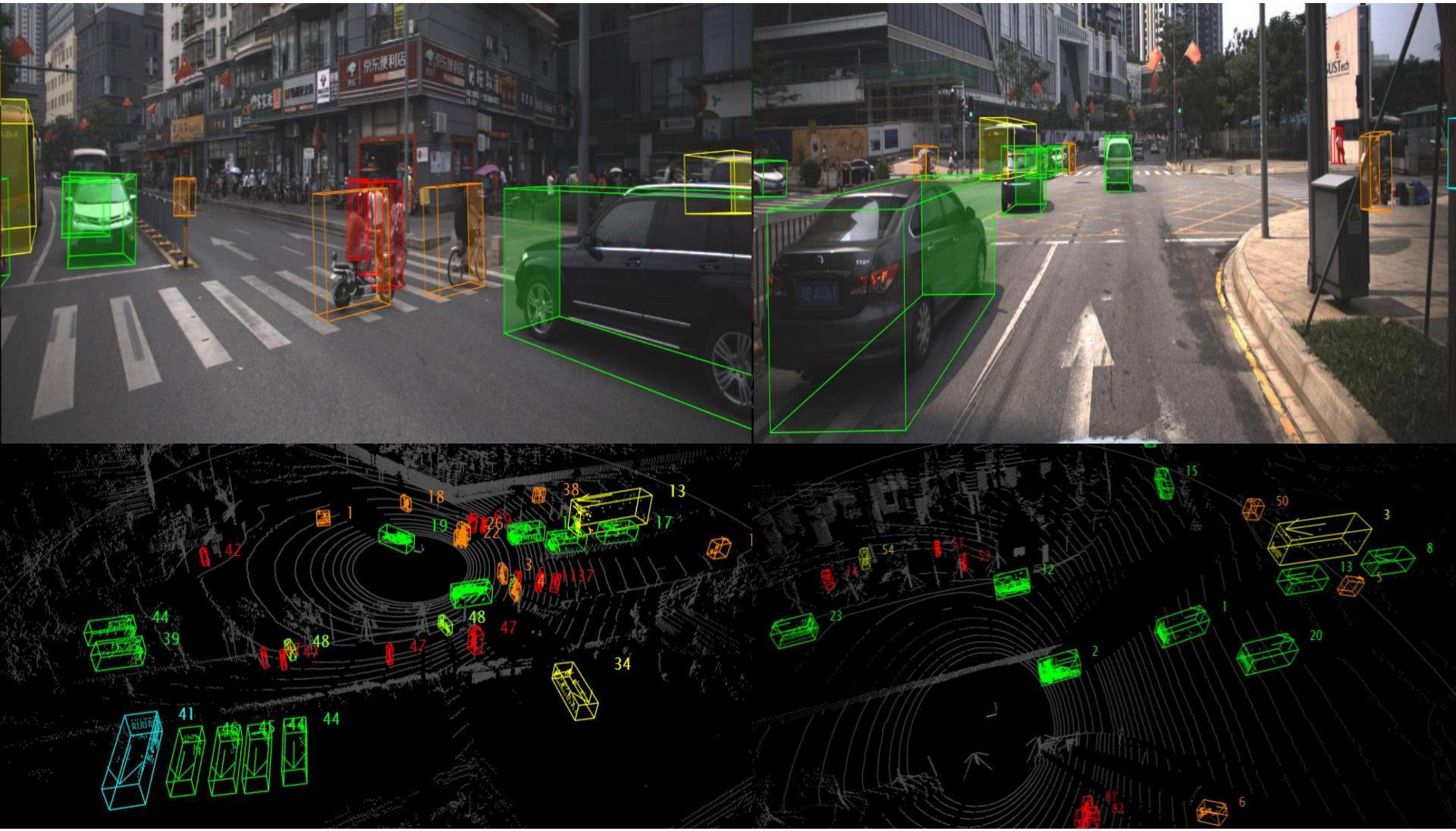
Datasets Comparison

Dataset	Sensors	Location Accuracy	Scene Diversity	Annotation				Driver	
				2D	3D	Video	Lane	Behav	Physi
KITTI	Cameras (6), Lidar (1)	cm	Regions Day time	box-15k, pixel-400	Box	no	no	no	no
Oxford RobotCar	Camera, Lidar	unknow	1 City, weather	no	no	no	no	no	no
TSD-max	Camera	m	Regions	box-10k	no	no	2D	no	no
CityScape	Camera	unknow	50 Cities, Season, weather, D&N	pixel-25k	no	no	no	no	no
Udacity	Camera, Lidar	unknow	1 City, sunny and overcast	box-25k	no	no	2D	no	no
Mapillary	Camera	m	Cities, season, weather, D&N	pixel-25k	no	no	2D, 2 classes	no	no
TorontoCity[1]	Camera, Lidar	cm	1 City	pixel	point	no	no	no	no
BDD100K	Camera, Lidar	m	4 Regions, weather, D&N	box-100k, pixel-10k	no	no	2D, 8 classes	no	no
ApolloScape	Cameras (6) Lidars (2) Cars (4)	cm	4 Regions, weather, D&N	pixel-140k resolution 2K	point	yes	2D, 3D, 28 classes	no	no
SUSTech Scape	Camera (6) Lidars (3) M-spectral Cameras (2) Buses (4) Taxis (4)	cm	1~5 Cities, weather, day and night	pixel-200k resolution <4K	point	yes	2D, 3D >28 classes	yes	yes

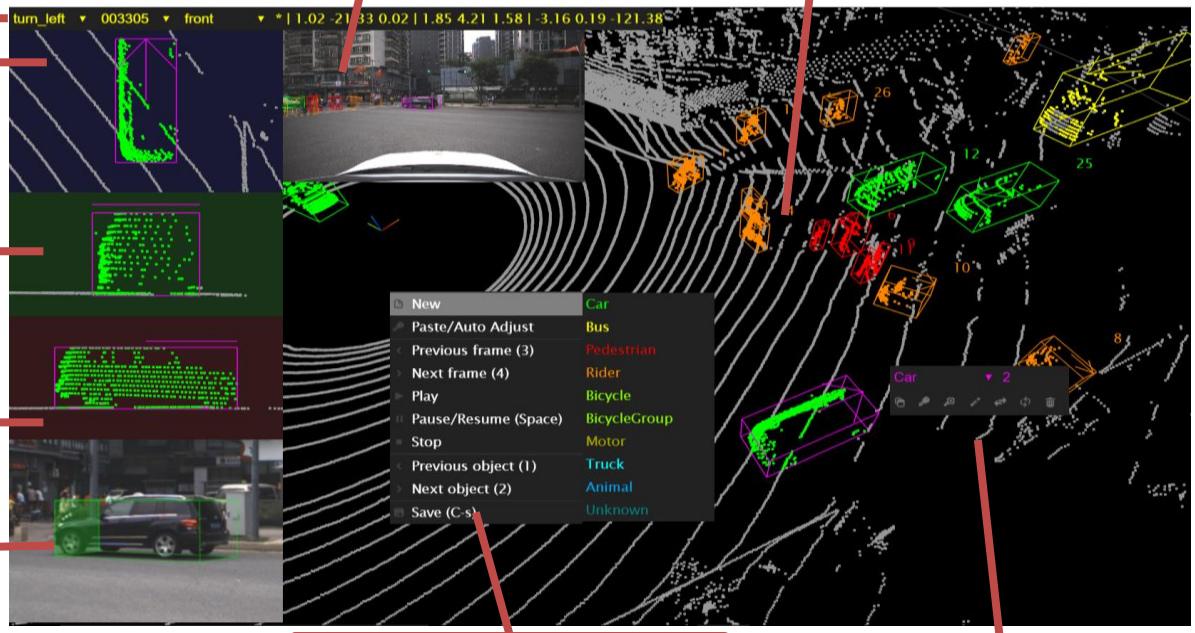
Benchmark Metrics

Evaluation Metrics			Methods	Measures	Subjects
Scene Processing	Scene flow	stereo disparity outliers [%] optical flow outliers [%] scene flow outlier [%]	Image, 3D Points Cloud Based Experiments	Optical and scene flow estimation errors, 3D reconstruction errors compared with ground truth	**Optical flow estimation, scene flow estimation, 2D-3D reconstruction algorithms**
	Depth	Scale invariant error [%] Relative squared error [%] Absolute squared error [%] Root mean squared error [%]			
Mapping & Localization Accuracy	Mapping	iRMSE [1/km] iMAE [1/km] RMSE [mm] MAE [mm]	Odometry (IMU, Camera) Readings and 3D Points Cloud Based Experiments	Environment mapping and vehicle localization under various scenarios and conditions	**SLAM, key frame selection, sensor fusion, optimization algorithms**
	Odometry	Translation errors [%] Rotation errors [deg/m]			
Energy Efficiency	Urban Area	Battery capacity [kW/h] Range [kM] Complete weight [t]	Testing Experiments	Current and power consumption of devices	**Power modes, battery management and Scheduling algorithms**
	Suburb Area				
	Hybrid Area				
System Scalability	Task completion time [h] Task completion distance [km] MTCA [km] MTDA [h]		Simulation Experiments in Different Scales	Predicted system performances in different scales of the vehicle fleet	**Communication protocols, group formation control, collaborative planning Algorithms**
System Reliability & Robustness	**Scenarios Overlap Rate[%] One Pass Rate[%] Defect resolution rate[%] Pressure running time[%]**		**Simulation Experiments under Complex Scenarios**	**Sharp turns, degraded marks, faulty signals, reckless driving, sensor failures**	**Sensor perception, motion planning, decision & control algorithms**

Dataset Annotation



Dataset Annotation Tools



Scene & frame navigation
Box information
Camera selection

Photo context
Freely resizable
Auto-follow target obj.

Colored objects

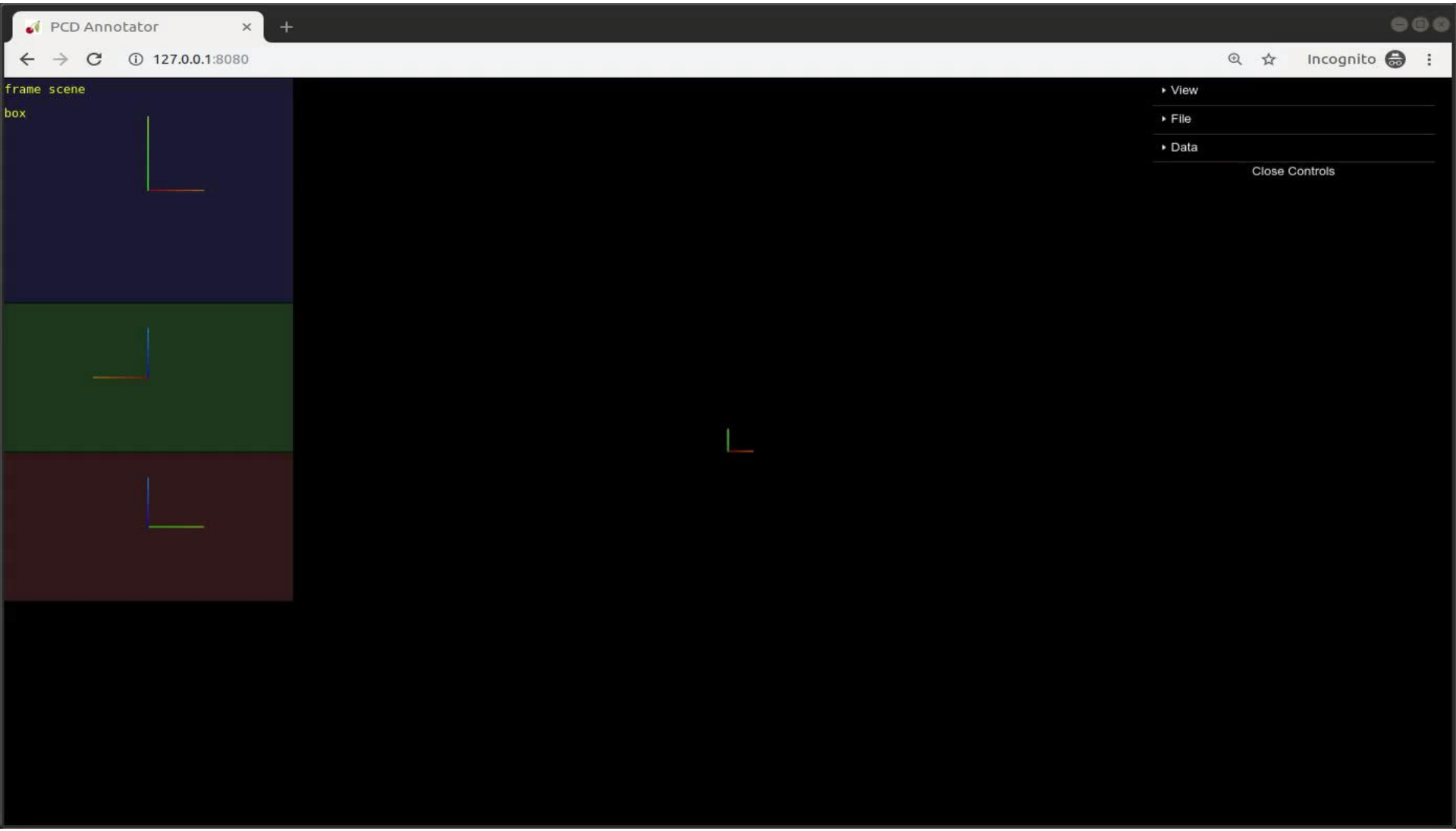
Projective view
Auto-shrink
Rotation
Resize
Move
Keyboard fine-tune

Auto-focused target object

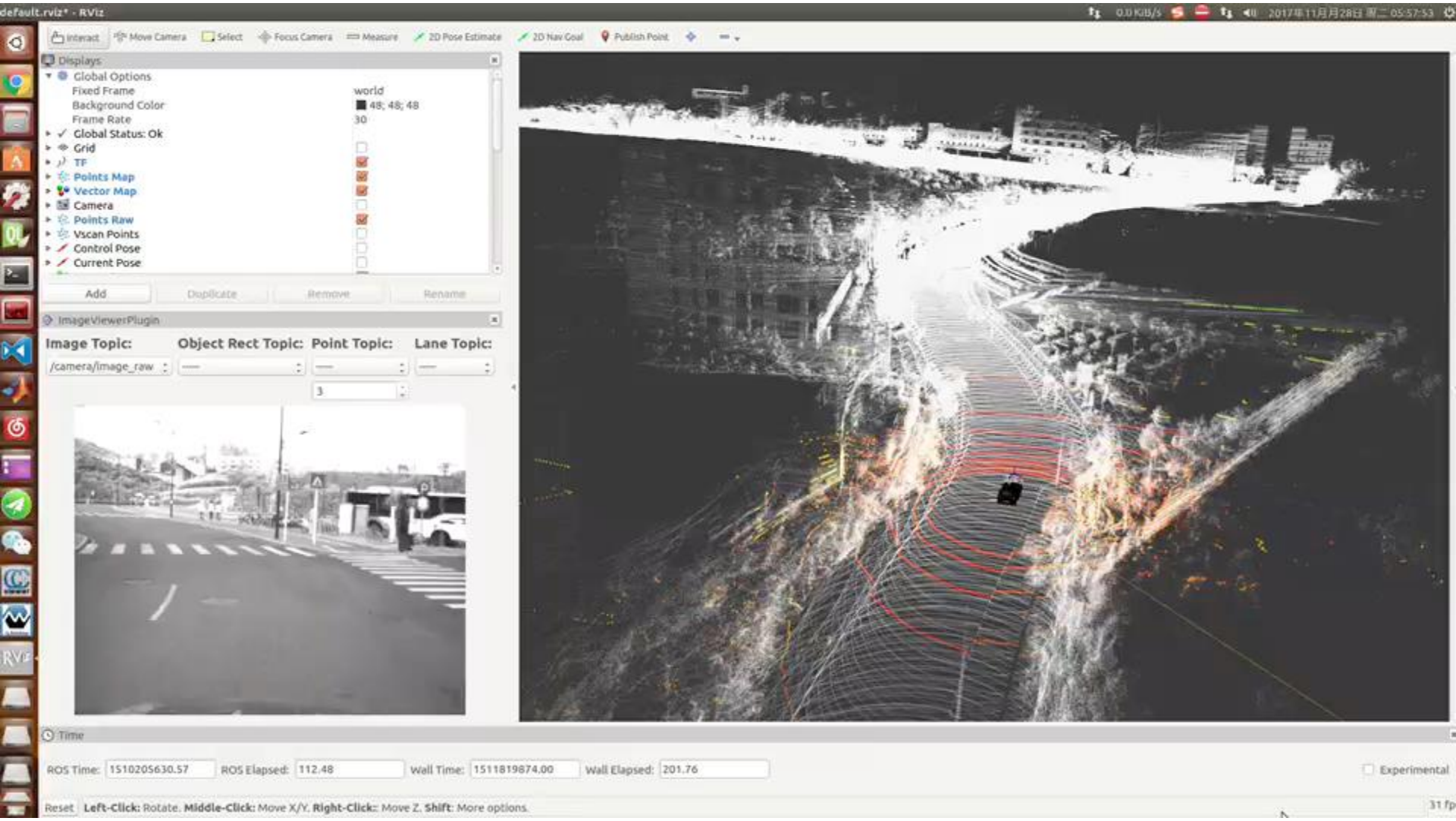
Context menu
Frame/obj. navigation
Stream play/stop
New/paste/save

Floating fast toolbox

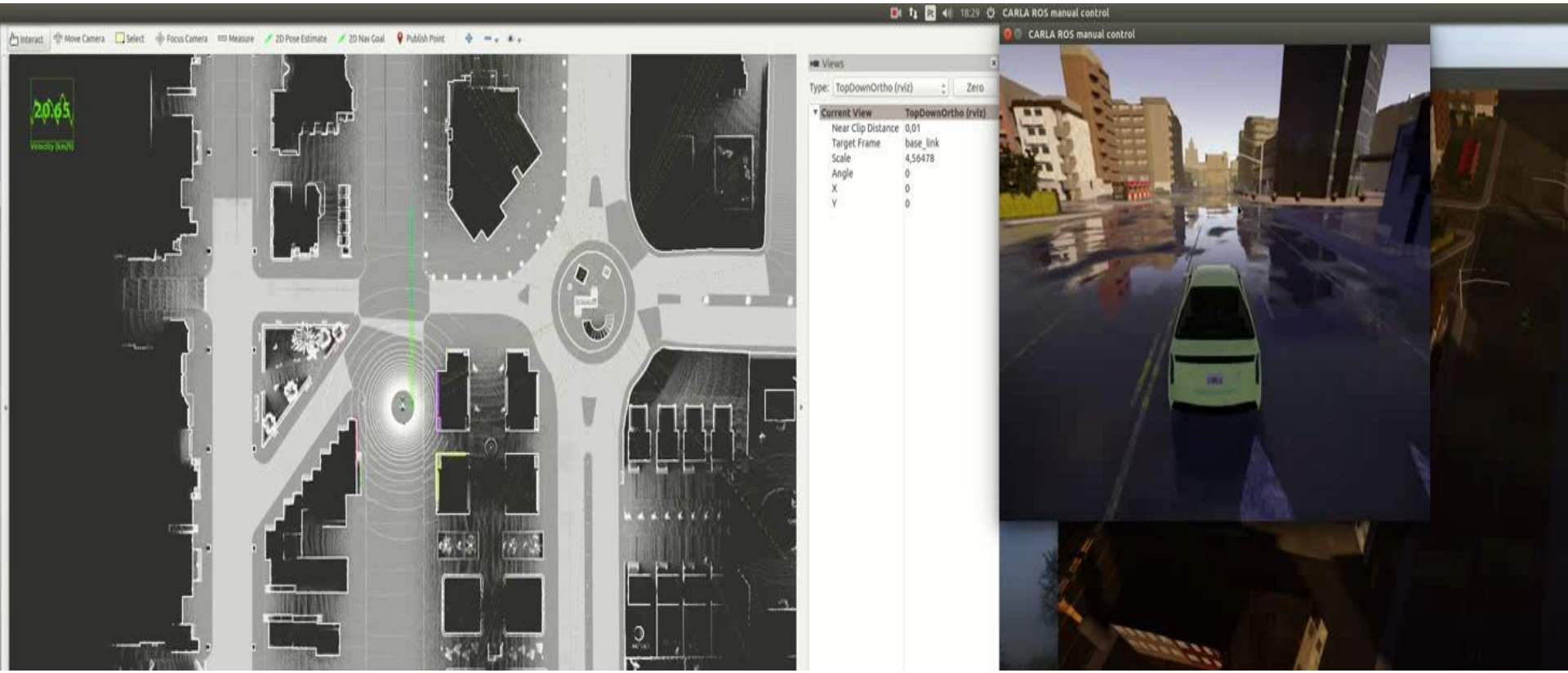
SUSTech POINTS



Play-Back: Autoware



Simulators: CARLA+Autoware



More Reading and Multimedia Materials

Books： 《人类简史》《奇点将近》《终极算法》《人工智能时代》《2050》《情感机器》《数学之美》

Movies： “Blade Runner” “AI” “Prometheus” “Covenant” “Ex Machina” “She” “2001：Space Odyssey” “The Matrix” “I，Robot” “Bicentennial Man” “Terminator”

TV Series： “West World” “Humans” “Black Mirrors”

More Course Links

Stanford Machine Learning：
https://see.stanford.edu/Course/CS229/47

MIT Machine Learning： https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-867-machine-learning-fall-2006/index.htm

Stanford CNN for Vision： http://cs231n.stanford.edu

Stanford Deep Learning: http://cs230.stanford.edu/syllabus.html

MIT Deep Learning: http://introtodeeplearning.com/